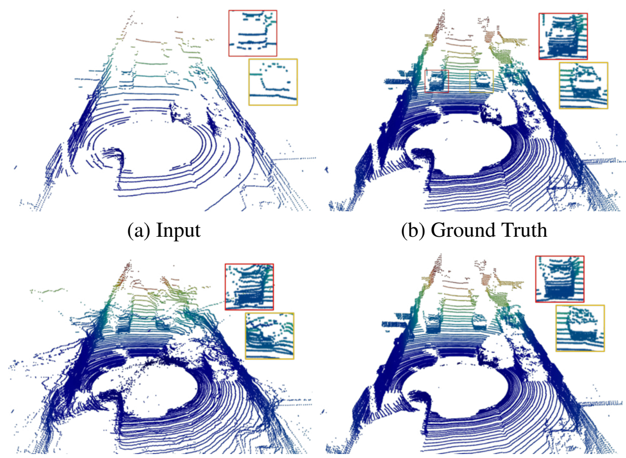
TULIP: Transformer for Upsampling of LiDAR Point Clouds

Bin Yang[1], Patrick Pfreundschuh[1], Roland Siegwart[1], Marco Hutter[2], Peyman Moghadam[3,4†], Vaishakh Patil[2†]
[1]Autonomous Systems Lab, ETH Zurich, Switzerland, [2]Robotic Systems Lab, ETH Zurich, Switzerland
[3]School of EER, Queensland University of Technology (QUT), Australia, [4]Data61, CSIRO, Brisbane, Australia
{biyang, patripfr, rolandsi, mahutter, patilv}@ethz.ch, peyman.moghadam@csiro.au

Abstract

LiDAR Upsampling is a challenging task for the perception systems of robots and autonomous vehicles, due to the sparse and irregular structure of large-scale scene contexts. Recent works propose to solve this problem by converting LiDAR data from 3D Euclidean space into an image super-resolution problem in 2D image space. Although their methods can generate high-resolution range images with fine-grained details, the resulting 3D point clouds often blur out details and predict invalid points. In this paper, we propose TULIP, a new method to reconstruct high-resolution LiDAR point clouds from low-resolution LiDAR input. We also follow a range image-based approach but specifically modify the patch and window geometries of a Swin-Transformer-based network to better fit the characteristics of range images. We conducted several experiments on three public real-world and simulated datasets. TULIP outperforms state-of-the-art methods in all relevant metrics and generates robust and more realistic point clouds than prior works. The code is available at https://github.com/ethz-asl/TULIP.git.



(a) Input (b) Ground Truth

(c) SwinIR [29] (d) *TULIP* (Ours)

Figure 1. *TULIP* generates more realistic LiDAR point clouds from low-resolution LiDAR input (a) and outperforms the state-of-the-art image super-resolution approach [29] in upsampling with fine-grained details of different objects in the scene.

1. Introduction

Light Detection And Ranging (LiDAR) is one of the most common sensors for perception in various fields of autonomy, such as autonomous driving, and unmanned aerial vehicles (UAV). LiDARs are used to generate 3D point clouds of the scene. These point clouds are essential for mapping, localization, and object detection tasks. However, the accuracy of these tasks often depends on the resolution of the point cloud [53]. Furthermore, the resolution of a LiDAR is inherently associated with increased energy consumption and cost, making its use impractical for various applications. LiDARs also have different vertical and horizontal resolutions. The vertical resolution of rotating 3D LiDARs is typically much lower than the horizontal resolution. These limitations make an upsampling technique necessary to increase the resolution of the LiDAR data, especially in the vertical direction. Moreover, upsampling LiDAR data has the potential to counter domain shift problems in LiDAR-based learning methods. Notably, the use of lower resolution LiDAR data in a system that has been trained on higher resolution data leads to a significant drop in performance [3]. Therefore, such upsampling techniques can not only help to mitigate the domain gap by creating a virtual high-resolution sensor that matches the target domain [42, 61] but also reduce the high cost of collecting new LiDAR data, annotating and retraining the methods.

Deep learning has led to remarkable advancements in LiDAR upsampling techniques in recent years. Several recent methods [42, 56, 58] have focused on learning the upsampling process within the 3D Euclidean space. However, the processing of 3D data in deep learning can be quite resource-intensive. One way to address this challenge is the representation of the point cloud as a range image [18, 26, 43, 49]. This approach facilitates the adoption of well-established image-based super-resolution methods.

†Authors share last authorship.

Such methods mostly employ autoencoder-style networks with convolutional and deconvolutional layers [14, 43, 62]. However, these methods are not directly transferable to range images. Range images generated by the projection of sparse 3D point clouds have distinct sharp edges at the object boundaries. Convolutional neural network-based methods often induce edge smoothing due to the regularization effect, which limits their usability for LiDAR range image processing.

As an alternative, Transformer [48] has shown great success in RGB image super-resolution and produces less blurry outputs [29]. Vision transformers, however, face challenges such as the need for large amounts of training data [9, 29, 35] and the global computation of self-attention, limiting the capture of local information in the range images. To address this, Swin Transformer [33] introduced tokenizing the input image and applying the self-attention mechanism locally in independent windows. While the modifications to Swin Transformer have led to robust training and promising results compared to Vision Transformer, its default structure and settings are not ideally suited for processing LiDAR range images. These range images consist of single-channel data, representing 3D spatial information, as opposed to three-channel color images that depict visual appearance. Furthermore, range images feature large, smooth areas within 3D objects, separated by distinct sharp edges between objects. As a result, minor inaccuracies in the 2D prediction (*e.g.* blurry output) can lead to significant differences in the projected 3D occupancy. This non-uniform distribution of image details further complicates network training. The challenging details in the range image are concentrated in a few specific pixels, contrasting with RGB images where relevant details are evenly distributed. This makes range images fundamentally different from RGB images. Inspired by the limitations of state-of-the-art range image upsampling methods and the potential of Swin-Transformer [33], we propose a novel network named *TULIP*. Our proposed geometry-aware architecture is tailored to accommodate LiDAR range image data. The proposed method takes a low-resolution range image as input and produces a high-resolution range image, which can then be projected into a 3D point cloud. To accommodate LiDAR range image geometry, we utilize one-line row patches to tokenize the input range image, as opposed to the square patches used for RGB images. This approach helps to preserve vertical information for the upsampling process while capturing boundary discontinuities effectively. This also creates a trade-off between local detail capture and model complexity in the training process. Additionally, we incorporate a non-square window for local self-attention computation. This enhances learning of spatial contexts at different scales in the range image. Further, we extensively train and evaluate *TULIP* on two real-world and one simulated autonomous driving datasets: KITTI [21], DurLAR [28] and CARLA [26], respectively. Comparisons to existing approaches demonstrate that *TULIP* outperforms the state-of-the-art methods on all three datasets.

2. Related Works

Image super-resolution has seen significant progress in recent years. The technique aims to construct high-resolution (HR) images from low-resolution (LR) observations, often leveraging advances in Convolutional Neural Networks (CNNs) to enhance the fidelity and detail of images for better visualization and information extraction. Subsequently, several enhanced frameworks have been developed [14, 15, 24, 25, 59].

Driven by their success in the field of natural language processing (NLP) [4, 23], Transformers [48] have been extended to solve a variety of vision-related tasks such as object detection [6] and semantic segmentation [2, 7]. Such Vision Transformers (ViT) [17] excel at learning to focus on relevant image regions by exploring global interactions among different regions. Their impressive performance has led to their adoption in image restoration tasks [9, 11, 29] as well. While they brought significant improvements in RGB image super-resolution, Vision Transformer [17] comes with the drawback of the quadratic computational complexity of self-attention. Additionally, they capture mostly global dependencies within the data and require large amounts of data for training.

To address these challenges, Swin Transformer [33] has been proposed. Unlike the Vision Transformer, which relies on global self-attention across the entire image, the Swin Transformer employs a local window-based attention mechanism and establishes a pyramid-like architecture. This approach processes images hierarchically by gradually merging smaller image patches into larger ones. This strategy enables more effective handling of different scales and facilitates the processing of multi-scale features, making it particularly suitable for range image processing. Furthermore, due to Swin Transformer's hierarchical structure, the increase in the number of model parameters scales linearly with the input image size, which mitigates the challenges associated with handling large range images. Additionally, they require less training data than the classical Vision Transformer, which is beneficial for LiDAR upsampling as LiDAR datasets are typically magnitudes smaller than RGB datasets used for Vision Transformer. The Swin and Vision Transformers have also been used for image super-resolution on omnidirectional panoramic camera images [45, 55]. However, these approaches are specifically designed for the unique characteristics of these cameras (*e.g.* strong distortions, spherical continuity) that are vastly different from the LiDAR sensing model, which renders them unsuitable for application to range images.

Other related works have focussed on upsampling high-density point clouds from low-density point clouds [38, 54, 56, 60, 61]. While those approaches also provide qualitative results for LiDAR point clouds, they focus on increasing the general 3D point density of point clouds of single objects instead of a large scene. Another area closely related to LiDAR upsampling is depth completion [12, 22, 30, 36, 37, 39, 46, 52]. The primary objective of depth completion is to improve sparse depth estimates acquired from LiDAR by integrating data from multiple sensors, predominantly RGB cameras. The result of depth completion methods is a dense depth map that provides depth values to every pixel in the input depth map. Nonetheless, this field slightly diverges from LiDAR upsampling due to its reliance on multimodal sensors and its common deployment within a restricted field of view, making it not directly comparable within the scope of this paper. LiDAR point clouds exhibit specific characteristics, such as a distinct point pattern (due to the stacked lasers in rotating 3D LiDARs) and a decrease in point density with increasing distance from the sensor. Differently from arbitrary point cloud upsampling, LiDAR upsampling tries to mimic a realistic point cloud of a high-resolution LiDAR given the point cloud of low-resolution LiDAR and thus targets a different result than the approaches above. Due to the difficulty and high computational cost of upsampling LiDAR point clouds in 3D space [1, 8, 42], most works [26, 43, 47] represent the point cloud as a range image and perform the LiDAR upsampling task in 2D image space. LiDAR-CNN [47] introduces a CNN-based architecture with semantic and perceptual guidance. The method specifies two further loss functions besides a per-point reconstruction loss to acquire a better synthesis of the high-resolution range image. However, this approach is limited to data with semantic annotations, which drastically reduces its applicability. LiDAR-SR [43] deploys a CNN-based network with U-Net [40] architecture. It additionally uses Monte-Carlo dropout post-processing to reduce the amount of noisy points in the prediction. Approaches based on convolutional operations tend to fail to reconstruct the sharpness in the range image effectively. While the previous approaches directly predict the range image, Implicit LiDAR Network (ILN) [26] uses an implicit neural architecture that learns interpolation weights to fill in new pixels instead of their values directly. Although the method outperforms CNN-based approaches in training speed and preserving the geometrical details in the input, it can still suffer from limited neighboring information in the LiDAR data, especially in areas distant from the sensor due to the extremely sparse input data. Differently from the previous works, our approach builds on Swin Transformer as a backbone and is specifically customized to effectively process range images.

3. Methodology

LiDAR upsampling is achieved by upsampling a range image, effectively transforming it from a 3D upsampling problem into a 2D image super-resolution problem [18, 26, 43, 47]. It is, therefore, an evident approach to build on RGB image super-resolution. However, image super-resolution aims to solve a different problem than LiDAR upsampling. Image super-resolution enhances the visual appearance of a low-resolution RGB image by recreating a high-resolution image. While quantitative metrics for evaluating these works exist, there is no one specific correct solution. Differently, LiDAR upsampling tries to recreate a high-resolution, 3D LiDAR sensor output from a low-resolution LiDAR input point cloud. Range images contain more geometrical and spatial contexts rather than visual information, and the scene's geometry strictly dictates the correct solution. The underlying sensing model is also drastically different, as LiDARs are active sensors. The respective range pixel values purely depend on distance, while pixels in an RGB camera depend on many factors, such as scene appearance, lighting conditions, exposure time, and white balance. Furthermore, range images have a highly asymmetric aspect ratio, typically between 1:8 to 1:64, which differs drastically from regular camera images. Accordingly, most works [18, 43, 47] only perform vertical upsampling, in contrast to image super-resolution, which operates in both dimensions. A naive application of state-of-the-art image super-resolution networks [29] to range images does not result in adequate performance (Fig. 1). To this end, we developed *TULIP*, a novel range image-based LiDAR Upsampling method. Based on the observations above, we specifically adapt a network based on SwinUnet [5] to incorporate range images better. In the following sections, we will describe the technical details of our method, with a special focus on how our approach accommodates range images.

3.1. Problem Definition

A LiDAR point cloud consists of points captured during one revolution $\mathcal{P} = \{\mathbf{p}_1, ..., \mathbf{p}_n\}$. Each measurement represent a 3D point $\mathbf{p}_i = \{x_i, y_i, z_i\}$. As input, we assume a low-resolution point cloud $\mathcal{P}_l$ with $n_l = H_l \times W_l$ points, where H_l and W_l correspond to vertical and horizontal resolution. We aim to predict a high-resolution point cloud $\mathcal{P}_h$, that has the same field-of-view(FoV) as $\mathcal{P}_l$ but contains $n_h = \beta * n_l$ points. In our experiments, we set $\beta = 4$. We only increase vertical resolution (amount of LiDAR beams), *i.e.*, $n_h = H_h \times W_h$, where $H_h = \beta * H_l$ and $W_h = W_l$. We project the point cloud into a 2D range image. In this image, each row and column coordinate v, u correspond to the respective elevation and azimuth angles of the LiDAR points, while the pixel value contains the range of the point $r = \sqrt{x^2 + y^2 + z^2}$. The image coordinate of a 3D point can be calculated using a spherical projection model:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{W}{2} - \frac{W}{2\pi} arctan(\frac{y}{x}) \\ \frac{H}{\Theta_{max} - \Theta_{min}} * (\Theta_{max} - arctan(\frac{z}{\sqrt{x^2+y^2}}) \end{bmatrix} \quad (1)$$

Θ_{max} and Θ_{min} correspond to the limits of the vertical field of view of the LiDAR. We formulate the LiDAR upsampling problem as an image upsampling problem, *i.e.* we aim to predict a high-resolution range image $I_h \in \mathbf{R}^{1 \times H_h \times W}$ given a low-resolution range image $I_l \in \mathbf{R}^{1 \times H_l \times W}$. We can then calculate the high-resolution point cloud $\mathcal{P}_h$ by inverting the projection in Eq. 1.

3.2. Architecture Design and Overview

Our network builds upon Swin-Unet [5] which was originally designed for image segmentation. We deploy a U-shaped network structure featuring skip connections that link the encoder and decoder modules. The fundamental building block of *TULIP* is the Swin Transformer [33]. *TULIP* first patchifies and maps the input to a high-dimensional feature space. The tokenized low-level feature maps are passed through a multi-stage encoder. Each stage contains a pair of Swin Transformer [33] blocks and a patch merging layer to downsample the resolution by a factor of two and increase the dimension by a factor of four. This step realizes the hierarchic computation and extraction of multi-scale features through encoding. Within each stage, multiple instances of the self-attention mechanism are performed locally in parallel, which is the so-called Window Multi-Head Self Attention (W-MSA). After passing through a two-layer MLP and residual connection with the input, the feature vector is further passed to the second part of the Swin Transformer block, utilizing a Shifted Window Multi-Head Self Attention (SW-MSA) which extends W-MSA by employing a shifted window operation. This enhances the model capacity as it compensates for the lack of interaction between the local windows in W-MSA. The decoder, which realizes the upsampling has a symmetric design to the encoder and operates in a reverse way of the encoder. The resolution of the feature maps is first expanded, and the dimension is reduced by a factor of two accordingly. We feed existing geometrical information via skip connections. The feature maps are transformed into a single-channel, high-resolution range image in the last layer. The reconstruction head comprises a 1×1 convolutional layer for feature expansion, followed by a Leaky ReLU activation, a pixel shuffle layer [44], and another 1×1 convolutional layer for final projection. As the loss function, we select the pixel-wise L1 loss. Details of the network can be found in the Appendix.

3.2.1 Tokenization

Tokenization is done by a patch partition layer that creates an initial feature embedding from the input range image. This initial feature representation significantly influences the network's performance. Specifically, the selection of this layer determines how the inherent characteristics, patterns, and relevant information of the LiDAR range image data are encoded into a format that the network can effectively learn from. Most transformer-based RGB image super-resolution approaches utilize a relatively large patch size to tokenize the input [9, 29, 35]. These larger patches enable the construction of a global spatial context, facilitating an understanding of the overall structure and high-level features of the RGB image. These networks employ square-shaped patches and implement upsampling in both vertical and horizontal directions. In contrast, LiDAR upsampling primarily aims to enhance the vertical resolution of the input, given that range images possess properties that are vastly different from RGB images. Furthermore, attention across more pixels in range images is less useful than in RGB images, since there is almost no geometrical relation for the spatial contexts that are far from each other in the 2D range image space. Motivated by this, we propose two adjustments to effectively process range images within a Swin-Unet:

Row-Based Patch Partition: Our model builds row patches with a dimension of 1×4 for range image tokenization. The new patch geometry is designed to retain full vertical information while compressing horizontally. This aligns with our objective of conserving and extending vertical details in the range image. Besides that, row patches excel at capturing boundary discontinuities between distinct objects and the background within the scene. Moreover, the selection of the patch size strikes a balance between enhancing model performance and limiting model capacity. Smaller patches can capture more detailed local information, but they also increase the number of model parameters, which can slow down the model's training process.

Circular Padding (CP) in Horizontal Dimension: Instead of adding zero pixels padding, our model uses circular padding in the horizontal dimension. This avoids introducing artificial features and naturally matches the sensor model of a rotating 3D LiDAR. This preserves accurate neighbor relations along the edges, which would otherwise be compromised during the projection to the 2D range image. The Circular Padding (CP) also allows us to fit flexible input sizes without changing the patch geometry.

3.2.2 Non-Square window for local self-attention

Inside Swin Transformer, input feature maps are window-partitioned along width and height to compute the self-attention locally in each window. In addition to square-shaped RGB images, some recent work applies a square window for panoramic data as well [31, 57]. However, using square windows on range images is unfavorable. On one

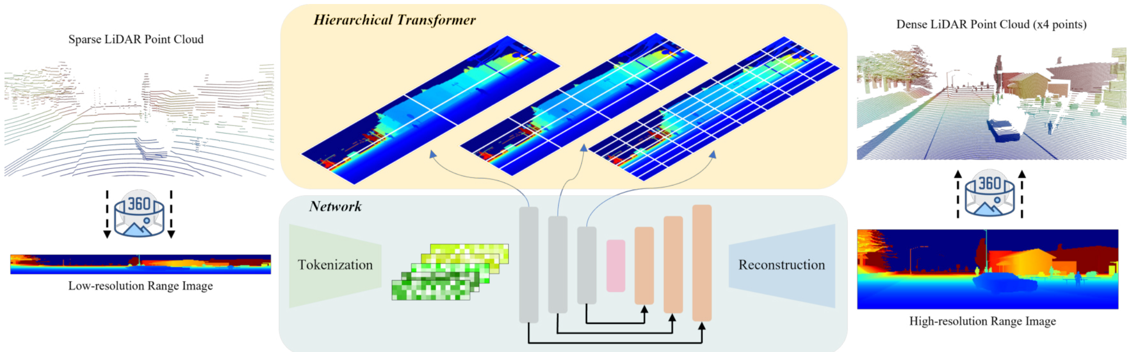


Figure 2. Overview of *TULIP*: The low-resolution point cloud is spherically projected into the range image. In the training and inference, the range image is tokenized into feature embeddings without compressing the spatial resolution along the vertical axis. We adopt the U-Net-based network presented in [5] for range image upsampling. We follow the hierarchical structure of Swin Transformer [33] for feature extraction and use the non-square window geometry for computing the local self-attention. The reconstruction head generates the high-resolution range image from the latent features and then, the point cloud can be obtained by projecting pixels back into the 3D space.

hand, small windows can focus more on attention within near areas but lead to increasing computational complexity and can struggle with the reconstruction of objects at a further distance in the scene. On the other hand, a larger square window can blend scene-related contexts at different scales, which can potentially degrade the network performance. In a prior study done by Eskandar *et al*. [18], they demonstrated a significant enhancement in range image super-resolution by separately upsampling the upper and lower portions of the data. In contrast to their approach of utilizing CNN-based shallower and deeper network branches for distinct feature extraction, our proposal involves the direct partitioning of the input using a rectangular window. This window geometry facilitates robust attention interaction among distant points through W-MSA and subtle cross-attention between objects at multiple scales via SW-MSA and hierarchical processing. In addition, range images, as typical panorama data, inherently contain abundant information along the horizontal direction. Therefore, directing attention more toward image width rather than height can aid the network in capturing a greater amount of geometrical information. At the same time, the computational complexity remains the same as using a large square window.

3.2.3 Further Adaptation and Refinement

Patch Unmerging (PU) and Pixel Shuffle (PS): Similar to the patch merging layer that is responsible for downsampling at the encoder stage, a mechanism is necessary within the decoder to upsample the feature maps. Conventional CNN-based approaches [43, 50] achieve upsampling through the transposed convolutional operation. As they tend to smooth out image sharpness [29], we reversed the operation of patch merging, re-arranging the channels of patches into their respective positions in a grid to upsample the spatial resolution of input feature maps, which we denote as patch unmerging. Furthermore, to reconstruct the final range image, we built the reconstruction head upon the upsampled feature maps. The implementation of the reconstruction module is based on the pixel shuffle layer [44].

Monte Carlo Dropout: We additionally use Monte Carlo Dropout [43] to filter unreliable points by thresholding the uncertainty. MC-Dropout executes several feed-forward passes during inference with different active dropouts, which yields a distribution of outputs. We refine the results by removing points with high output variance, as they often come down to noisy, invalid points.

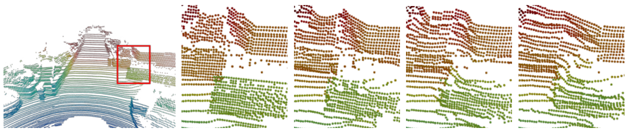
4. Experiments

4.1. Experimental Settings

Datasets: We conduct experiments on three different datasets, that include the two large-scale real-world datasets DurLAR [28] and KITTI [21] (Sec. 4.3.2), as well as on a dataset [26] that was generated using the CARLA simulator [16] (Sec. 4.3.1). We select test sequences that are recorded in different locations than the train set to avoid spatial overlap. We vertically downsample the high-resolution range images with a factor of four by skipping the respective lines to generate low-resolution input images. For ablation studies in Sec. 4.2, we select the KITTI dataset.

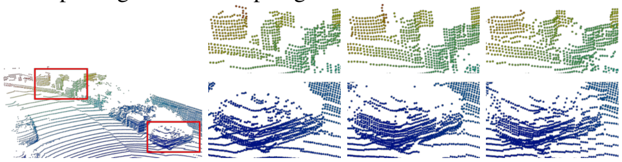
Implementation Details: In all experiments, we use distributed processing with $4\times$ GeForce RTX 2080 Ti. For training, we use a fixed batch size of eight per GPU for all datasets. We use the AdamW [34] optimizer with a weight decay factor of 0.01 and a base learning rate of 5e-4.

Evaluation metrics: We evaluate the Mean Absolute Error (MAE) for all the pixels in the generated 2-D range images. Additionally, we assess the performance by considering the 3D points reconstructed by our neural networks. The Chamfer Distance (CD) measures the Euclidean distance between



(a) Scene (b) GT (c) 1×4 (d) 2×2 (e) 4×4

Figure 3. Upsampling results with different patch sizes: We observe that square patches (4×4, 2×2) blur out edges and create invalid connections between separate walls, while the proposed 1×4 patch generates sharp edges and correct discontinuities.



(a) Scene (b) GT (c) Rectangular (d) Square

Figure 4. Upsampling results with different window geometries: Rectangular windows improve reconstruction of objects at different ranges, indicated by more distinct discontinuities (top) and a better reconstruction of the car (bottom).

two point clouds. We also follow the approach in [26] to evaluate the volumetric occupancy similarity. To do so, the point clouds are voxelized using a voxel size of $0.1m$. A voxel is classified as occupied for each point cloud if it contains at least one point. We then calculate the Intersection-over-Unit (IoU) based on the occupancy.

4.2. Ablation Studies

Patch size: In Tab. 1 Baseline and Model 1-2, we compare the upsampling results with different patch sizes. We observe that the smaller patch (2×2) outperforms the larger one (4×4) while the proposed row patch (1×4) can further refine the network performance significantly. In particular, the row patch improves 32.6% in MAE, 12.8% in IoU, and 28.3% in CD, compared to a 4×4 patch. As visualized in Fig. 3, the row patch better preserves sharpness around edges and corners compared to the larger patch sizes.

Window geometry: To verify the effectiveness of modifying window geometry for range images, we conduct experiments with square and rectangular windows (Tab. 1 Model 2-3. Based on the quantitative results, we can observe the

Model	Patch	Window	CP/PU/PS	MC	MAE ↓	IoU ↑	CD ↓
Baseline [5]	4×4	Square	×	×	0.7138	0.3250	0.1940
Model 1	2×2	Square	×	×	0.6251	0.3337	0.1661
Model 2	1×4	Square	×	×	0.4814	0.3667	0.1391
Model 3	1×4	Rectangular	×	×	0.4248	0.4040	0.1218
Model 4	1×4	Rectangular	✓	×	0.4227	0.4084	0.1207
TULIP	1×4	Rectangular	✓	✓	0.4185	0.4174	0.1207
TULIP-L	1×4	Rectangular	✓	✓	0.3708	0.4329	0.0992
ViT-Unet*	1×4	-	✓	✓	1.4484	0.2948	0.2665

Table 1. Ablation study results for $4\times$ upsampling on KITTI. Input Resolution: 16×1024, Output Resolution: 64×1024. *We replaced Swin Transformer blocks with ViT blocks and followed the same training procedure as for *TULIP*.

following: reconstruction quality produced by local attention within large square windows is inferior to the rectangular window concerning both 2D and 3D evaluation metrics, while they result in the same computational complexity. The underlying reason for this performance difference lies in the perceptive field of the windows. A wider window is more likely to include discontinuities between objects at different scales. Accordingly, the network performs better at separating objects from each other, which can visually be validated by clearer surface boundaries in Fig. 4.

Further Adaptation and Refinement: Circular Padding (CP) helps maintain continuity and consistency at the edges of the panoramic images, but it has minor contributions to other areas. Patch Unmerging (PU), and Pixel Shuffle (PS) aim at upscaling the spatial resolution with rearrangement, avoiding information loss and parameter increase led by using additional de-convolutional layers. Although they are designed to improve efficiency rather than efficacy of the model, tested with those components (Model 4), a slight improvement in upsampling is still observable. Monte Carlo Dropout [19], as a post-processing step, helps to further reduce ghost points in between objects and we present more details in the Appendix. For *TULIP*-L, we increase the model capacity, using four Swin-Transformer layers instead of three in both encoder and decoder, which demonstrates an additional performance boost on all metrics.

Transformer Block: From the previous discussion, we infer that Swin Transformer [33] is more advantageous than ViT (Vision Transfromer) [17] in LiDAR upsampling. On one hand, prior works [13, 20, 27, 32] have pointed out the inferiority of ViT on smaller datasets due to lack of locality learning and non-overlapping attention. Relevant LiDAR datasets [21, 28] are generally small compared to RGB dataset [41]. On the other hand, by narrowing the patch size, the exponential increase of model capacity hinders sufficient training with ViT on range images, while Swin Transformer expands only linearly. To numerically show the distinction, we trained the network on the same dataset by replacing ViT blocks and applied the same patch size and additional components to the network for a fair comparison. In the last row of Tab. 1, it shows that *TULIP* significantly outperforms the one with ViT as the backbone.

4.3. Benchmark Results

Besides bilinear interpolation, which computes the interpolation weights from the four nearest neighbors, we evaluate against the state-of-the-art LiDAR upsampling approaches Implicit LiDAR Network (ILN) [26] and LiDAR-SR [43]. We also compare against several image super-resolution works: SRNO [51], and LIIF [10] learn implicit features for interpolation instead of 2D coordinates of new pixels while HAT [9] and Swin-IR [29] are pixel-based approaches that directly reconstruct high-resolution images from low-

Model	MAE ↓	IoU ↑	CD ↓
CARLA 4x Output Resolution: 128×2048			
Bilinear	1.8128	0.1382	0.7262
SRNO [51]	2.4640	0.1343	2.0230
HAT [9]	1.6032	0.2698	0.6337
SWIN-IR [29]	1.9560	0.2718	0.4840
LIIF [10]	0.8064	0.3502	0.174
LIDAR-SR [43]	0.8216	0.2581	0.2044
ILN [26]	0.8592	0.5006	0.1855
TULIP (Ours)	0.7699	0.5152	0.1028
TULIP-L (Ours)	**0.7539**	**0.5301**	**0.1001**
KITTI 4x Output Resolution: 64×1024			
Bilinear	2.0892	0.1063	0.6000
SRNO [51]	0.8350	0.2035	0.4417
HAT [9]	0.6856	0.2035	0.2516
SWIN-IR [29]	1.2972	0.2774	0.7347
LIIF [10]	0.6143	0.3226	0.1916
LIDAR-SR [43]	0.5674	0.1005	0.2165
ILN [26]	1.0528	0.3342	0.2787
TULIP (Ours)	0.4185	0.4174	0.1207
TULIP-L (Ours)	**0.3708**	**0.4329**	**0.0992**
DurLAR 4x Output Resolution: 128×2048			
Bilinear	2.4384	0.1266	0.6346
SRNO [51]	1.5396	0.1507	0.5108
HAT [9]	1.7820	0.2353	0.1973
SWIN-IR [29]	1.9416	0.2157	0.2279
LIIF [10]	1.5672	0.2469	0.1548
LIDAR-SR [43]	**1.5312**	0.1370	0.1128
ILN [26]	1.5720	0.3430	0.0893
TULIP (Ours)	1.5432	0.3562	0.06484
TULIP-L (Ours)	1.5592	**0.3654**	**0.06346**

Table 2. Quantitative comparison against state-of-the-art LiDAR and image super-resolution methods on different datasets. All methods are trained and evaluated on the same splits.

resolution feature embeddings. Similar to our work, they are based on Swin Transformer [33]. We train all methods on the specific datasets using their default parameters.

4.3.1 Simulation Results

Similar to prior works, we first evaluate the performance on noise-free, simulated data. We train and test all approaches on the dataset presented in [26] that was recorded using the CARLA simulator and select the ground-truth range images of size 128×2048 that capture a vertical FoV of $30°$. We use a (20699/2618) train/test split. We observe that the performance between image super-resolution approaches varies greatly, and LIIF performs the best among them. While LIIF [10] achieves a comparable MAE to the LiDAR upsampling methods, the 3D metrics are significantly worse, indicating that it is unsuitable for the underlying geometric input data. Nevertheless, LIIF [10] still outperforms the LiDAR upsampling method in LIDAR-SR. The interpolation-based approach in ILN shows decent results for the noise-free data, as indicated by their high IoU. Nevertheless, our approaches achieve the best performance in all evaluated metrics, with the strongest improvement in Chamfer Distance, which validates that our design is well suited for the geometric reconstruction in range images.

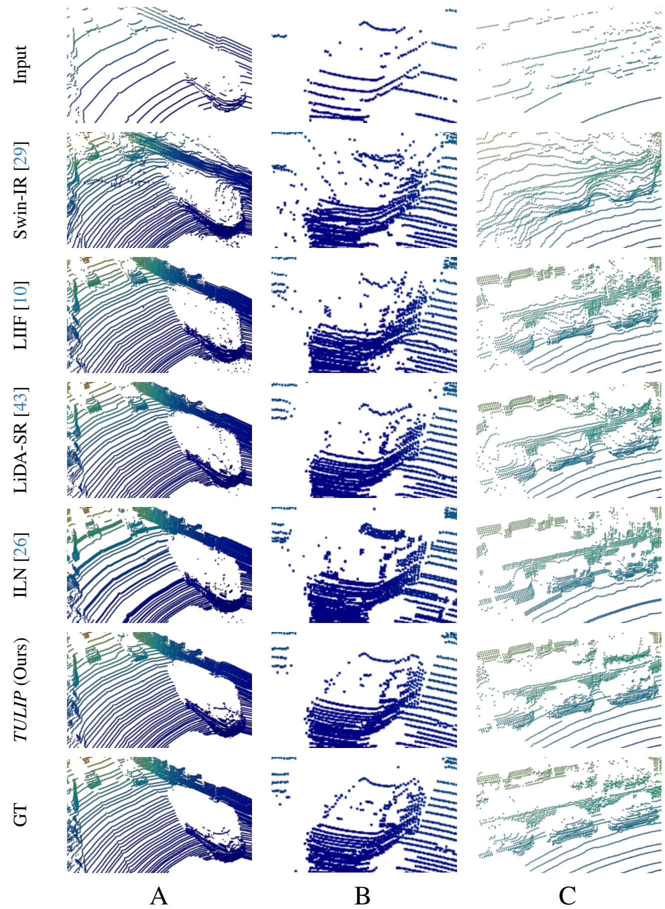


Figure 5. Qualitative results on KITTI: We provide visualizations for the top-4 results from Tab. 2 and Swin-IR [29]. Our approach outperforms other state-of-the-art methods in upsampling realistic point clouds. **A**: *TULIP* is the only approach that does not generate invalid ghost points in between the car (right bottom) and the wall behind. It can also be observed that all approaches except for ILN match the characteristic line pattern of the LiDAR on the ground (middle). **B**: Swin-IR and LIIF generate a blurry reconstruction. *TULIP* achieves the clearest point cloud and even reconstructs small details such as the car's side mirrors. **C**: Swin-IR, LIIF, and LiDAR-SR do not clearly separate the car from the wall behind. ILN generates a clear point cloud but does not properly reconstruct the shape of the cars. *TULIP* achieves the clearest point cloud, with distinct discontinuities between objects, and closely resembles the shape of the three cars.

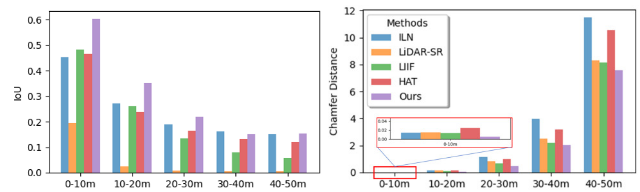
4.3.2 Real-World Results

To evaluate our approach on real-world data, we downsample the DurLAR and KITTI datasets to a comparable number of frames to the CARLA dataset by temporally skipping frames in the dataset sequences. In particular, we have the following (train/test) split: DurLAR (24372/2508), and KITTI (20000/2500).

KITTI: The KITTI dataset [21] was collected using a Velodyne HDL-64E LiDAR with a vertical FoV of 26.8° and a resolution of 64×1024. For the image super-resolution approaches, we observe comparable results and trends on KITTI as on the simulated data in Sec.4.3.1. Even though LIDAR-SR [43] is designed for LIDAR upsampling and outperforms image super-resolution approaches in terms of MAE, it achieves the lowest IoU among all approaches, which can be explained by a large amount of invalid floating points between real objects. LIIF [10] achieves better CD and comparable IoU compared to ILN. We observe that the IoU of ILN [26] is drastically lower than on CARLA. The interpolation scheme in ILN is thus not suitable to properly handle the lower resolution data with noise from the real-world sensor. Our approaches significantly outperform all competing methods, especially in terms of IoU as it can better handle discontinuities between objects. As indicated in the visualizations in Fig.5, our approach predicts fewer invalid float points between objects, as well as fewer blurry edges, which results in much clearer point clouds.

DurLAR: The DurLAR dataset [28] was recorded using an Ouster OS1-128 LiDAR, that captures point clouds with a resolution of 128×2048 with a 45° vertical FoV. Compared to CARLA and KITTI, we see a large performance drop for all approaches in MAE and IoU, which results from the larger maximum range and vertical FoV of the sensor as well as stronger sensor noise. We observe that all approaches designed for LiDAR upsampling clearly improve in Chamfer Distance compared to image super-resolution approaches. ILN and *TULIP* outperform all methods by a large margin in IoU. *TULIP*-L achieves the best 3D metrics, which validates that it can also handle the challenging DurLAR data better than prior works.

Range Analysis: As the density of LiDAR point clouds decreases with increasing distance from the sensor, we additionally compare the performance at different ranges for a more fine-grained evaluation. We observe that our approach achieves better or similar performance compared to all approaches in all metrics at all ranges. Up to 30m, *TULIP* clearly outperforms the baseline approaches. *TULIP* handles discontinuities between objects better than other approaches, which can be verified visually through sharper edges and fewer "floating" points between objects. This is especially reflected in the high IoU. At ranges above 30m, our approach performs comparable to other methods.



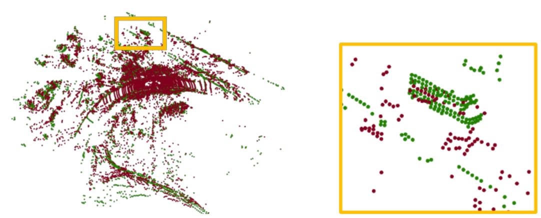
(a) IoU ↑ (b) Chamfer distance ↓

Figure 6. 3D error metrics visualized at different ranges for KITTI.

Above this range, LiDAR data is extremely sparse, meaning that points neighboring in the low-resolution range image are actually distant in the Euclidean space. It is, therefore, not feasible to infer meaningful information about the scene from such neighboring points at a high range, which is also represented in the poor performance of all approaches.

4.4. Failure Cases

Although *TULIP* outperforms other state-of-the-art methods both qualitatively and quantitatively, the upsampling quality can still be limited in certain cases. For instance, as shown in Fig 7, our approach shows inferior upsampling results for the specific scene. The irregularity of the scene leads to high uncertainty, which makes the reconstruction noisy. Observing the details in Fig. 7a, the network fails to reconstruct the car at the top of the scene.



(a) Full scene (b) Details

Figure 7. Failure Case: noisy point cloud generated by *TULIP* and missing reconstruction of the object. (Ours and GT)

5. Conclusion

This work presents *TULIP*, a novel method for LiDAR Upsampling that achieves incredible performance in upsampling the range image. Our approach transforms 3D point clouds into 2D range images and performs the upsampling in 2D space. We build upon a Swin-Transformer-based network and specifically modify the patch partition and attention windows to better accommodate the characteristics of range images. Throughout various experiments, testing on three different benchmarks, it shows that *TULIP* outperforms state-of-the-art methods quantitatively in all evaluation metrics, and quantitatively, *TULIP* demonstrates more promising results in upsampling the realistic LiDAR data.
Acknowledgement: This work was supported by Swiss National Science Foundation's NCCR DFab P3.

References

[1] Anique Akhtar, Zhu Li, Geert Van der Auwera, Li Li, and Jianle Chen. Pu-dense: Sparse tensor-based point cloud geometry upsampling. *IEEE Transactions on Image Processing*, 31:4133–4148, 2022. 3

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022. 2

[3] Borna Bešić, Nikhil Gosala, Daniele Cattaneo, and Abhinav Valada. Unsupervised domain adaptation for lidar panoptic segmentation. *IEEE Robotics and Automation Letters*, 7(2): 3404–3411, 2022. 1

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[5] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022. 3, 4, 5, 6

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2

[8] Tso Yuan Chen, Ching Chun Hsiao, and Ching-Chun Huang. Density-imbalance-eased lidar point cloud upsampling via feature consistency learning. *IEEE Transactions on Intelligent Vehicles*, 2022. 3

[9] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22367–22377, 2023. 2, 4, 6, 7

[10] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 6, 7, 8

[11] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12312–12321, 2023. 2

[12] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 103–119, 2018. 3

[13] Yin Dai, Fayu Liu, Weibing Chen, Yue Liu, Lifu Shi, Sheng Liu, Yuhang Zhou, et al. Swin mae: Masked autoencoders for small datasets. *Computers in Biology and Medicine*, 161: 107037, 2023. 6

[14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 2

[15] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 391–407. Springer, 2016. 2

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 5

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 6

[18] George Eskandar, Sanjeev Sudarsan, Karim Guirguis, Janaranjani Palaniswamy, Bharath Somashekar, and Bin Yang. Hals: A height-aware lidar super-resolution framework for autonomous driving. *arXiv preprint arXiv:2202.03901*, 2022. 1, 3, 5

[19] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 6

[20] Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. How to train vision transformer on small-scale datasets? In *33rd British Machine Vision Conference*, 2022. 6

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2, 5, 6, 8

[22] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13656–13662. IEEE, 2021. 3

[23] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, page 2, 2019. 2

[24] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 2

[25] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016. 2

[26] Youngsun Kwon, Minhyuk Sung, and Sung-Eui Yoon. Implicit lidar network: Lidar super-resolution via interpolation weight prediction. In *2022 International Conference on*

Robotics and Automation (ICRA), pages 8424–8430. IEEE, 2022. 1, 2, 3, 5, 6, 7, 8

[27] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021. 6

[28] L. Li, K.N. Ismail, H.P.H. Shum, and T.P. Breckon. Durlar: A high-fidelity 128-channel lidar dataset with panoramic ambient and reflectivity imagery for multi-modal autonomous driving applications. In *Proc. Int. Conf. on 3D Vision*. IEEE, 2021. 2, 5, 6, 8

[29] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 1, 2, 3, 4, 5, 6, 7

[30] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1638–1646, 2022. 3

[31] Zhixin Ling, Zhen Xing, Xiangdong Zhou, Manliang Cao, and Guichun Zhou. Panoswin: A pano-style swin transformer for panorama understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17755–17764, 2023. 4

[32] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small datasets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 6

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2, 4, 5, 6, 7

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 5

[35] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 457–466, 2022. 2, 4

[36] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 120–136. Springer, 2020. 3

[37] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019. 3

[38] Shi Qiu, Saeed Anwar, and Nick Barnes. Pu-transformer: Point cloud upsampling transformer. In *Proceedings of the Asian Conference on Computer Vision*, pages 2475–2493, 2022. 3

[39] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6250–6259, 2022. 3

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 6

[42] Artem Savkin, Yida Wang, Sebastian Wirkert, Nassir Navab, and Federico Tombari. Lidar upsampling with sliced wasserstein distance. *IEEE Robotics and Automation Letters*, 8(1): 392–399, 2022. 1, 3

[43] Tixiao Shan, Jinkun Wang, Fanfei Chen, Paul Szenher, and Brendan Englot. Simulation-based lidar super-resolution for ground vehicles. *Robotics and Autonomous Systems*, 134: 103647, 2020. 1, 2, 3, 5, 6, 7, 8

[44] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 4, 5

[45] Xiaopeng Sun, Weiqi Li, Zhenyu Zhang, Qiufang Ma, Xuhan Sheng, Ming Cheng, Haoyu Ma, Shijie Zhao, Jian Zhang, Junlin Li, et al. Opdn: Omnidirectional position-aware deformable network for omnidirectional image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1293–1301, 2023. 2

[46] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020. 3

[47] Larissa T Triess, David Peter, Christoph B Rist, Markus Enzweiler, and J Marius Zöllner. Cnn-based synthesis of realistic high-resolution lidar data. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1512–1519. IEEE, 2019. 3

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[49] Suhong Wang, Hongqing Wang, Shufeng She, Yanping Zhang, Qingju Qiu, and Zhifeng Xiao. Swin-t-nfc crfs: An encoder–decoder neural model for high-precision uav positioning via point cloud super resolution and image semantic segmentation. *Computer Communications*, 197:52–60, 2023. 1

[50] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022. 5

[51] Min Wei and Xuesong Zhang. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18247–18256, 2023. 6, 7

[52] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Baobei Xu, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *European Conference on Computer Vision*, 2021. 3

[53] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1951–1960, 2019. 1

[54] Shuquan Ye, Dongdong Chen, Songfang Han, Ziyu Wan, and Jing Liao. Meta-pu: An arbitrary-scale upsampling network for point cloud. *IEEE transactions on visualization and computer graphics*, 28(9):3206–3218, 2021. 3

[55] Fanghua Yu, Xintao Wang, Mingdeng Cao, Gen Li, Ying Shan, and Chao Dong. Osrt: Omnidirectional image super-resolution with distortion-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13283–13292, 2023. 2

[56] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2790–2799, 2018. 1, 3

[57] Heeseung Yun, Sehun Lee, and Gunhee Kim. Panoramic vision transformer for saliency detection in 360 videos. In *ECCV*, 2022. 4

[58] Wentai Zhang, Haoliang Jiang, Zhangsihao Yang, Soji Yamakawa, Kenji Shimada, and Levent Burak Kara. Data-driven upsampling of point clouds. *Computer-Aided Design*, 112:1–13, 2019. 1

[59] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *TPAMI*, 2020. 2

[60] Wenbo Zhao, Xianming Liu, Zhiwei Zhong, Junjun Jiang, Wei Gao, Ge Li, and Xiangyang Ji. Self-supervised arbitrary-scale point clouds upsampling via implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2007, 2022. 3

[61] Yifan Zhao, Le Hui, and Jin Xie. Sspu-net: Self-supervised point cloud upsampling via differentiable rendering. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2214–2223, 2021. 1, 3

[62] Örkény Zováthi, Balázs Pálffy, Zsolt Jankó, and Csaba Benedek. St-depthnet: A spatio-temporal deep network for depth completion using a single non-repetitive circular scanning lidar. *IEEE Robotics and Automation Letters*, 2023. 2