

# Spotlights: Probing Shapes from Spherical Viewpoints

Jiixin Wei<sup>1</sup>[0000–0002–7645–6093], Lige Liu<sup>2</sup>, Ran Cheng<sup>2</sup>, Wenqing Jiang<sup>1</sup>[0000–0002–4514–0424], Minghao Xu<sup>2</sup>, Xinyu Jiang<sup>2</sup>, Tao Sun<sup>2\*</sup>, Sören Schwertfeger<sup>1</sup>[0000–0003–2879–1636], and Laurent Kneip<sup>1</sup>[0000–0001–6727–6608]

<sup>1</sup> ShanghaiTech University

{weijx,jiangwq,soerensch,lkneip}@shanghaitech.edu.cn

<sup>2</sup> RoboZone, Midea Inc

{liulig12,chengran1,xumh33,jxy77,tsun}@midea.com

**Abstract.** Recent years have witnessed the surge of learned representations that directly build upon point clouds. Inspired by spherical multi-view scanners, we propose a novel sampling model called *Spotlights* to represent a 3D shape as a compact 1D array of depth values. It simulates the configuration of cameras evenly distributed on a sphere, where each virtual camera casts light rays from its principal point to probe for possible intersections with the object surrounded by the sphere. The structured point cloud is hence given implicitly as a function of depths. We provide a detailed geometric analysis of this new sampling scheme and prove its effectiveness in the context of the point cloud completion task. Experimental results on both synthetic and real dataset demonstrate that our method achieves competitive accuracy and consistency while at a lower computational cost. The code and dataset will be released at <https://github.com/goldoak/Spotlights>.

**Keywords:** Shape Representation · Point Cloud Completion.

## 1 Introduction

Over the past decade, the community has put major efforts into the recovery of explicit 3D object models, which is of eminent importance in many computer vision and robotics applications [13, 2, 10, 23, 22, 33, 21, 36]. Traditionally, the recovery of complete object geometries from sensor readings relies on scanning systems that employ spherical high-resolution camera arrangements [11, 26]. If additionally equipped with light sources, such scanners are turned into light stages. However, without such expensive equipment, shape reconstruction often turns out to be erroneous and incomplete owing to occlusions, limited viewpoints, and measurement imperfections.

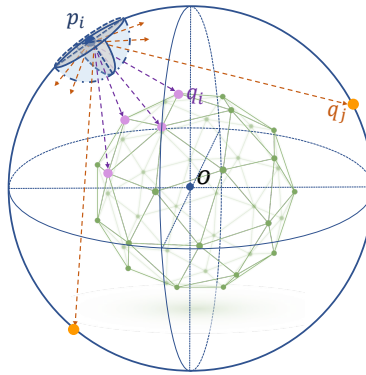
Previous work has investigated many different representations of objects to deal with those problems, such as volumetric occupancy grids [41], distance

---

\* Corresponding author.

fields [24], and more recently, point cloud-based representations [30, 31]. The latter are preferred in terms of efficiency and capacity of maintaining fine-grained details. Yet, each point in a point cloud needs to be represented by a 3D coordinate, and the large number of points generated in reality can strain computational resources, bandwidth, and storage.

To this end, we propose a novel structured point cloud representation inspired by spherical multi-view 3D object scanners [11, 26]. Sparse point clouds are generated by a simulated configuration of multiple views evenly distributed on a sphere around the object where each view casts a preset bundle of rays from its center towards the object. Due to its similarity to stage lighting instruments, we name it as *Spotlights*. It represents point clouds in a more compact way since it only needs to store a 1D array of scalars, i.e. the depths along the rays, and thus serves as a basis for efficient point-based networks. Moreover, Spotlights is a structured representation that produces ordered point clouds. This order-preserving property is of great benefit to spatial-temporal estimation problems such as Simultaneous Localization And Mapping (SLAM) [36] and 3D object tracking [17].



**Fig. 1.** Illustration of Spotlights with only one view. Dashed arrows indicate the ray bundle cast from  $p_i$ . Some rays intersect with the bounded object (purple) while others spread in void space (orange).

To demonstrate the superiority of Spotlights, we focus on the 3D shape completion task which aims to predict a complete point cloud with respect to a partial observation. Previous efforts are devoted to inferring fine-grained details [5, 14, 44] and designing delicate point-based decoders [43][37]. In this work, we present a lightweight neural network for fast point cloud prediction based on our proposed shape representation. It achieves competitive results in accuracy without substantially sabotaging completeness. Furthermore, our completed point clouds can easily find correspondences across temporal states, therefore, are useful for some subsequent tasks. Our main contributions are as follows:

- We introduce a novel point representation denoted *Spotlights*. It is a unified and compact representation. The effectiveness of the proposed representation is shown by the theoretical analysis of local point density.
- We apply our proposed representation to the task of 3D shape completion and achieve remarkable results on par with existing supervised state-of-the-art approaches on both synthetic and real data. Notably, our network consumes less memory and is fast at inference time.
- We build a new synthetic dataset for 3D shape completion, comprising 128000 training samples and 30000 testing samples over 79 unique car models (i.e. 2000 different observations per object). The ground truth point clouds are extracted by our Spotlights.

## 2 Related Work

Nowadays, shape prediction is mostly studied in the context of shape completion and reconstruction from 3D data (e.g. depth images or point clouds). Though this problem can be solved using lower-level geometric principles (e.g. volumetric diffusion [6], symmetry [38]) or shape retrieval methods [19, 25], the focus of our work lies on learning-based 3D point cloud completion.

In recent years, the community has demonstrated the feasibility of learning highly efficient representations for geometric problems by relying directly on 3D point clouds [30, 31, 34]. Fan et al. [9] and Achlioptas et al. [1] introduce auto-encoder CNNs for direct point cloud completion, showing that point clouds encode only points on the surface, naturally permit direct manipulation and put fewer restrictions on the topology or fine-grainedness of a shape. Yuan et al. [44] extend the idea by permutation invariant and convolution-free feature extraction, as well as a coarse-to-fine feature generation process. Xie et al. [42] internally revert to a volumetric representation by adding differentiable layers to transform back and forth between point clouds and grids. Huang et al. [14] furthermore propose to limit the prediction to the missing part of a point cloud, thus preserving details in the original input measurement. Wen et al. [40] mimic an earth mover and complete point clouds by moving points along the surface.

While direct 3D point-based representations have lead to significant advances in terms of efficiency, they suffer from disorder and the inability to focus on inherent local structures. Yang et al. [43] propose *FoldingNet*, a deep auto-encoder that generates ordered point clouds by a folding-based decoder that deforms a regular 2D grid of points onto the underlying 3D object surface of a point cloud. Although they claim that their representation can learn “cuts” on the 2D grid and represent surfaces that are topologically different from a single plane, they at the same time admit that the representation fails on more complex geometries. Li et al. [18] propose to use GAN to generate realistic shapes from unit spheres, establishing dense correspondences between spheres and generated shapes. Wen et al. [39] propose an extension with hierarchical folding by adding skip-attention connections. Groueix et al. [12] introduce *AtlasNet*. Similar to the structure of FoldingNet, this work deforms multiple 2D grids. Nonetheless, the modelling as

a finite collection of manifolds still imposes constraints on the overall topology of an object’s shape. Tchapmi et al. [37] propose *TopNet*. Their decoder generates structured point clouds using a hierarchical rooted tree structure. The root node embeds the entire point cloud, child nodes subsets of the points, and leaf nodes individual points. Rather than sampling a finite set of manifolds, this architecture generates structured point clouds without any assumptions on the topology of an object’s shape.

### 3 Method

In this section, we first review relevant background knowledge on Fibonacci spheres and solid angles (see Section 3.1). Then, we describe the detailed construction of our newly proposed Spotlights representation (see Section 3.2), and further prove its theoretical validity (see Section 3.3).

#### 3.1 Background

**Spherical Fibonacci Point Set** One of the most common approximation algorithms to generate uniformly distributed points on a unit sphere  $\mathcal{S}^2 \subset \mathbb{R}^3$  is mapping a 2D Fibonacci lattice onto the spherical surface. The Fibonacci lattice is defined as a unit square  $[0, 1)^2$  with an arbitrary number of  $n$  points evenly distributed inside it. The 2D point with index  $i$  is expressed as

$$\mathbf{p}_i = (x_i, y_i) = \left( \left[ \frac{i}{\Phi} \right], \frac{i}{n} \right), 0 \leq i < n.$$

Note that  $[\cdot]$  takes the fractional part of the argument and  $\Phi = \frac{1+\sqrt{5}}{2}$  is the golden ratio. The 2D points are then mapped to 3D points on the unit sphere using cylindrical equal-area projection. The 3D points are easily parameterized using the spherical coordinates

$$\mathbf{P}_i = (\varphi_i, \theta_i) = (2\pi x_i, \arccos(1 - 2y_i)), \varphi_i \in [0, 2\pi], \theta_i \in [0, \pi].$$

We refer the readers to [15] for more details.

**Solid Angle** Similar to the planar angle in 2D, the solid angle for a spherical cap is a 3D angle subtended by the surface at its center. Suppose the area of the cap is  $A$  and the radius is  $r$ , then the solid angle is given as  $\Omega = \frac{A}{r^2}$ . Thus, the solid angle of a sphere is  $4\pi$ . Extended to arbitrary surfaces, the solid angle of a differential area is

$$d\Omega = \frac{dA}{r^2} = \frac{r \sin\theta d\varphi \cdot r d\theta}{r^2} = \sin\theta d\theta d\varphi,$$

where  $\theta \in [0, \pi]$  and  $\varphi \in [0, 2\pi]$ . By calculating the surface integral, we can obtain

$$\Omega = \iint \sin\theta d\theta d\varphi.$$

### 3.2 Spotlights Model Construction

In this subsection, we show how to construct the Spotlights model (see Fig. 1 for an illustration). There are three consecutive steps described in the following.

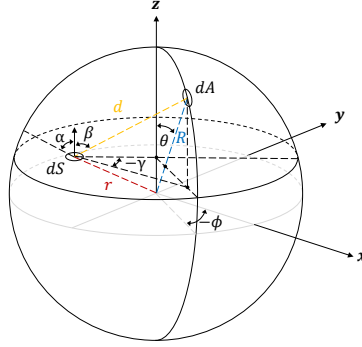
**Bounding Sphere Extraction** To find the bounding sphere of an object of interest, we first compute its bounding box by determining the difference between the extrema of the surface point coordinates along each dimension. Practically, we can also easily get the bounding box parameters (i.e. center, size and orientation) for real-world objects using state-of-the-art 3D object detection methods [28, 27, 29, 16, 7]. The bounding sphere is then given as the circumscribed sphere of the bounding box. The radius of the sphere is used as a scaling factor to normalize the real-world object into a unit sphere.

**Ray Bundle Organization** We leverage the Fibonacci lattice mapping method presented in Section 3.1 to evenly sample the unit sphere obtained from step 1 and place a bundle of virtual viewing rays at each position to mimic the behavior of a camera. The motivation of using a sphere instead of a cuboid arrangement of views is given by the fact that a spherical arrangement has more diversity in the distribution of principal viewing directions, thereby alleviating self-occlusion in the data. Also, spherical multi-view configurations are commonly adopted for 2.5D multi-surface rendering [35, 4].

To define structured ray bundles emitted from each viewpoint, we can imagine placing a smaller sphere at each primary sampling point on the unit sphere. The intersection plane between the unit sphere and each small sphere forms an internal spherical cap, on which we again use Fibonacci sampling to define a fixed arrangement of sampling points. Thus, the rays originate from the primary sampling points and traverse the secondary sampling points on the spherical cap. It is easy to adjust the opening angle of the ray bundle. Suppose the radii of the outer sphere and the small sphere are  $R$  and  $r$ , respectively. The maximum polar angle of the spherical cap is then given by  $\omega = \arccos \frac{r}{2R}$ . The opening angle  $\omega$  as well as the number of rays serve as design hyper-parameters. We refer to Section 5.3 for a detailed discussion.

**Ray Casting** We finally cast the rays onto the object and retrieve the distance between the view center and the first intersection point with the object’s surface. For rays that miss the surface, we manually set the distances to a sentinel value of zero. With known origins and viewing directions, the 3D point cloud can be easily recovered from the depths along the rays. Compared to 3D point coordinates, 1D distances are more efficient to store and generate. To validate this property, we apply our representation to the 3D point cloud completion task. While obtaining competitive results in terms of accuracy, our network is fast at inference time and has fewer parameters than previous methods (see Section 5.1). Besides, the fixed ray arrangement enables the reproduction of ordered point clouds, thus providing one-to-one correspondences across temporal states.

### 3.3 Analysis of Spotlights



**Fig. 2.** Mathematical model of Spotlights.

The quality of the sampling is measured from two aspects, namely the uniformity of the ray density for surfels at different positions and orientations within the bounding sphere when 1) there is no occlusion, or 2) there is occlusion. The first measure analyses how much the sampling density of the object surface depends on its location. The second measure provides a means to analyse the impact of occlusion that typically occurs in practical scenarios. To simplify the theoretical analysis, we make the following assumptions:

1. The object is bounded by a unit sphere, that is,  $R = 1$ .
2. An arbitrary surfel  $dS$  on the object can be parameterized by the distance of the surfel to the center of the sphere  $r$  and the orientation of the surfel  $\alpha$ , where  $\alpha$  is defined as the angle between the surface normal and the vector connecting the centers of the sphere and the surfel (see Fig. 2). The two parameters are uniformly distributed in their respective ranges, i.e.  $r \sim U(0, 1)$  and  $\alpha \sim U(0, \pi)$ .
3. There are  $N$  ray sources uniformly distributed on the sphere and each of them emits  $M$  rays within a certain solid angle  $\Omega$  oriented towards the center of the sphere. For convenience, we set  $\Omega = 2\pi$ . Note that we also assume  $N, M \gg 1$ . The source density  $\rho_s$  and the ray density  $\rho_r$  are then given by

$$\rho_s = \frac{N}{4\pi R^2} = \frac{N}{4\pi}, \quad \text{and} \quad \rho_r = \frac{M}{2\pi}.$$

In order to calculate the ray density of  $dS$ , we take another surfel  $dA$  on the sphere, calculate the ray density of  $dS$  caused by  $dA$ , and then do the integration over  $dA$ . The area of surfel  $dA$  with polar angle  $\theta$  and azimuth angle  $\phi$  is  $dA =$

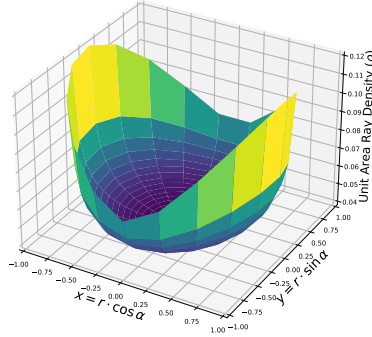
$\sin \theta d\theta d\phi$ . Using trigonometry function, we can calculate the distance between  $dS$  and  $dA$ :

$$d^2 = 1 + r^2 - 2r (\cos \alpha \cos \theta - \sin \alpha \sin \theta \cos \phi).$$

The rays that can be received by the surfel  $dS$  from  $dA$  are the solid angle occupied by the view of  $dS$  at  $dA$  multiplied by the ray density  $\rho_r$ . Then, the ray density  $\rho$  of surfel  $dS$  is the integration over  $dA$ , i.e.

$$\begin{aligned} \rho(r, \alpha) &= \frac{1}{dS} \int dA \cdot \rho_s \cdot \rho_r \cdot \frac{dS \cos \beta}{d^2} \\ &= \frac{NM}{8\pi^2} \int d\theta \int d\phi \frac{\sin \theta (\cos \theta - r \cos \alpha)}{(r^2 - 2r(\cos \alpha \cos \theta - \sin \alpha \sin \theta \cos \phi) + 1)^{3/2}}. \end{aligned}$$

The ray density  $\rho$  as a function of  $r$  and  $\alpha$  is shown in Fig. 3. We can observe that  $\rho$  has no divergence and is a fairly smooth function of  $r$  and  $\alpha$  when the surfel  $dS$  is not too close to the sphere surface. Quantitatively, when  $0 \leq r \leq 0.8$ ,  $\rho$  takes values between 0.039 and 0.057. The proposed sampling strategy therefore ensures that the object surfaces at arbitrary positions and orientations in the sphere have comparable sampling density.



**Fig. 3.** Ray density of surfel  $dS$ .

Next, we consider the case where certain directions seen from  $dS$  are occluded, therefore no rays can be received from the related sources on the surface of the sphere. The blocked direction is defined by the polar and azimuth angles  $(\beta, \gamma)$  (see Fig. 2). We assume that the blocking is random and uniform, i.e.,  $\beta \sim U(0, \pi)$  and  $\gamma \sim U(0, 2\pi)$ . The unit area ray density  $(\rho)$  as a function of  $(r, \alpha, \beta, \gamma)$  is then given by

$$\begin{aligned} \rho(r, \alpha, \beta, \gamma) &= \frac{1}{dS} \cdot dA \cdot \rho_s \cdot \rho_r \cdot \frac{dS \cos \beta}{d^2} \\ &= \frac{NM}{8\pi^2} \frac{\cos \beta}{(\xi r^2 + 1)^{1/2}}, \quad \xi = (\sin \alpha \sin \beta \cos \gamma - \cos \alpha \cos \beta)^2 - 1. \end{aligned}$$

As can be observed,  $\rho$  is a continuous function of  $\beta$  and  $\gamma$ , which ensures that when part of the directions are occluded, there are still rays that can hit  $dS$  from other angles, resulting in better sampling coverage. A simplified version of our point cloud sampling strategy only consists of casting a single ray towards the center of the sphere from each viewpoint. However, it is intuitively clear that this would limit the local inclination of rays sampling  $dS$  to a single direction, and thereby lead to an increased likelihood of occlusions.

Note that in practice the solid angle covered by each bundle of viewing rays should be chosen smaller such that more rays intersect with the object surface, and fewer rays are wasted. We experiment with different solid angles and find a balance between completeness and hit ratio in Section 5.3.

## 4 Shape Completion with Spotlights

Next we introduce our 3D shape completion network based on the Spotlights representation, named *Spotlights Array Network (SA-Net)*.

By leveraging the Spotlights model, we reformulate the 3D shape generation into a 1D array regression problem. The ground truth depth values  $\mathbf{d}_{gt}$  satisfies that

$$\mathbf{P}_{gt} = \mathbf{P}_1 + \mathbf{r} \cdot \mathbf{d}_{gt}, \text{ and } \mathbf{r} = \frac{\mathbf{P}_2 - \mathbf{P}_1}{\|\mathbf{P}_2 - \mathbf{P}_1\|}$$

where  $\mathbf{P}_{gt}$ ,  $\mathbf{P}_1$  and  $\mathbf{P}_2$  are ground truth point cloud, primary sampling points and secondary sampling points, respectively. The ray directions  $\mathbf{r}$  are computed using the primary sampling points and secondary sampling points as discussed in Section 3.2. Actually, the Spotlights model is formed by  $\mathbf{P}_1$ ,  $\mathbf{P}_2$  and  $\mathbf{r}$ .

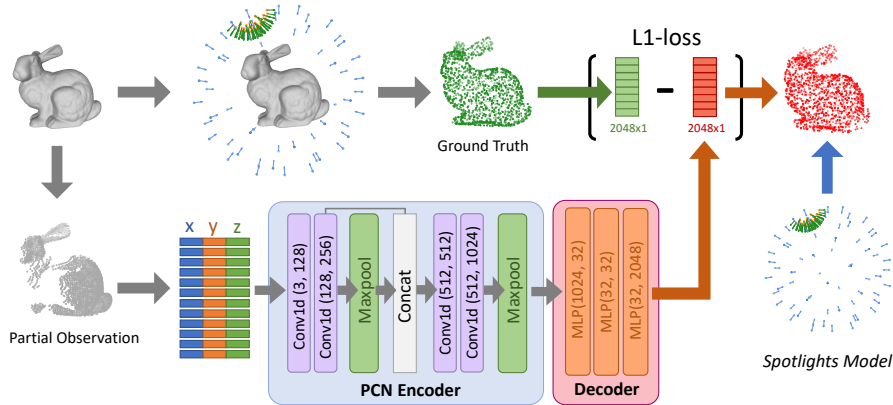
To speed up convergence, we normalize the ground truth depths into the range  $[0, 1]$  using a factor  $2R$  where  $R$  is the radius of the bounding sphere. Missing rays in ground truth are masked with a zero value. Note that the existence of missing rays requires post-processing on the predicted array to filter out the outliers in the recovered point cloud. This is achieved by simply clipping the array using a pre-defined threshold. Here, we empirically set the threshold value to be 0.2.

The pipeline of SA-Net is shown in Fig. 4. It consists of a point-based encoder and a decoder purely made up of multi-layer perceptrons (MLPs). Similar to previous work, we feed the partial observation into the encoder to obtain a 1024-dim latent vector. The difference is that we regress an array of scalars instead of 3D coordinates, which significantly reduces the computational cost. We calculate a L1 training loss between the predicted values and ground truth. At inference time, a complete 3D point cloud of the object can be recovered from the predicted 1D scalars using the Spotlights sampling model, that is,

$$\mathbf{P}_{pred} = \mathbf{P}_1 + \mathbf{r} \cdot \mathbf{d}_{pred}.$$

In addition, 3D points in  $\mathbf{P}_{pred}$  are ordered due to the structure in the Spotlights model.





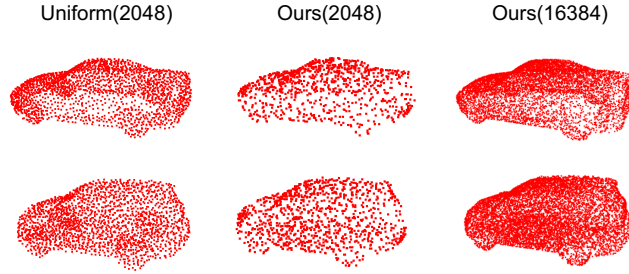
**Fig. 4. Architecture of SA-Net.** The network takes incomplete point cloud observations as input and predicts a 1D array, from which the complete point cloud is recovered by an inverse application of the Spotlights Model. Note that the PCN backbone can be replaced with other point-based encoders.

## 5 Experiments

We first describe how to build a synthetic dataset using Spotlights. Next, we perform 3D shape completion on both the newly made dataset and KITTI-360 [20], and compare our results against existing state-of-the-art methods. Moreover, we provide ablation studies on ShapeNet [3] to figure out the importance of the hyper-parameters involved in the Spotlights model.

### 5.1 Shape Completion on Synthetic Dataset

**Data Generation** We export 79 different vehicle assets of 3 categories from CARLA[8] and randomly select 64 vehicles for training, 8 vehicles for validation and the rest for testing. We use Blender to simulate a HDL-64 lidar sensor to further generate 2000 partial observations for each object and downsample those point clouds to 256 points. As for the ground truth, we uniformly sample 2048 points from the mesh surface for other methods while using Spotlights to get a 1D array containing 2048 values for SA-Net. Specifically, we evenly sample 32 primary points and 64 secondary points with  $60^\circ$  opening angle to construct the Spotlights model. For a fair comparison, we also densely sample 16384 points for each object to calculate the evaluation metrics introduced in the following. Examples of point clouds generated using different sampling methods are shown in Fig. 5. Despite the sparsity of the point cloud sampled by Spotlights, it still captures the overall shape of the target object. Besides, increasing the number of rays can notably improve the point density.



**Fig. 5.** Examples of point clouds generated using different sampling methods. The first column shows the point clouds uniformly sampled from the mesh surface. The second and the third columns show the point clouds sampled by our proposed Spotlights model using different numbers of rays.

**Comparison methods** We compare all methods on the Completion3D benchmark [37], an evaluation platform designed for 3D point cloud completion, and replace the original dataset with our generated synthetic one outlined above. In our setting, TopNet [37], PCN [44], FoldingNet [43] and AtlasNet [12] directly predict a complete point cloud containing 2048 points, whereas our SA-Net outputs a 1D array with 2048 values from which we further decode the sparse point cloud. According to the analysis in [44], we use the PCN encoder for all methods.

**Evaluation Metrics** The Chamfer Distance (CD) introduced by [9] is one of the most widely used metrics to measure the discrepancy between a predicted and the ground truth point cloud. Suppose the two point clouds are denoted as  $P_1, P_2 \subseteq \mathbb{R}^3$ . The CD is given by

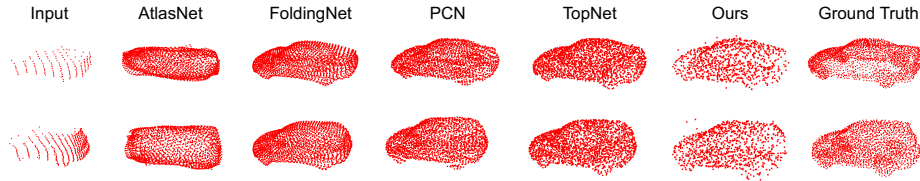
$$d_{CD}(P_1, P_2) = \frac{1}{|P_1|} \sum_{x \in P_1} \min_{y \in P_2} \|x - y\| + \frac{1}{|P_2|} \sum_{y \in P_2} \min_{x \in P_1} \|x - y\|, \quad (1)$$

and it computes the average nearest neighbor distance between  $P_1$  and  $P_2$ . Here we use CD in a single direction (i.e. we only employ one of the terms in (1)) to calculate the distance from the prediction to ground truth. As in [32], this metric indicates the accuracy of each of the predicted points with respect to the ground truth point cloud. To prevent large distances caused by too sparse reference point clouds, we furthermore choose dense ground truth point clouds with 16384 points. In addition, the L1-norm is used to improve robustness against outliers.

**Performance** The quantitative results of SA-Net compared with alternative methods [44, 43, 37, 12] is shown in Table 1. As can be observed, our SA-Net achieves competitive accuracy, while being faster and using less memory. Note that SA-Net performs better after applying an additional clip operation for statistical outlier removal. The qualitative results are shown in Fig. 6. The sparsity of our predicted point clouds is caused by missing rays.

**Table 1.** Accuracy and efficiency of comparisons on the synthetic dataset. Accuracy indicates the quality of completed point clouds, which is measured by the single direction CD. Efficiency is evaluated in terms of inference time and the number of network parameters.

| Method          | Accuracy       |                |                |                | Network     |               |
|-----------------|----------------|----------------|----------------|----------------|-------------|---------------|
|                 | Hatchback      | Sedan          | SUV            | Average        | #Params     | Runtime       |
| AtalsNet        | 0.06586        | 0.07325        | 0.06051        | 0.06462        | 8.3M        | 14.1ms        |
| PCN             | 0.02528        | 0.02842        | 0.02221        | 0.02428        | 7.6M        | 34.1ms        |
| FoldingNet      | 0.02437        | 0.02701        | 0.02166        | 0.02345        | 8.5M        | 34.6ms        |
| TopNet          | 0.02396        | <b>0.02631</b> | 0.02051        | 0.02252        | 7.5M        | 37.9ms        |
| Ours (w/o clip) | 0.02356        | 0.03095        | 0.02171        | 0.02442        | <b>0.9M</b> | <b>0.80ms</b> |
| Ours (w/ clip)  | <b>0.02124</b> | 0.02775        | <b>0.01850</b> | <b>0.02133</b> | <b>0.9M</b> | 1.05ms        |



**Fig. 6.** Qualitative Results of shape completion on the synthetic dataset.

## 5.2 Shape Completion on Real Dataset

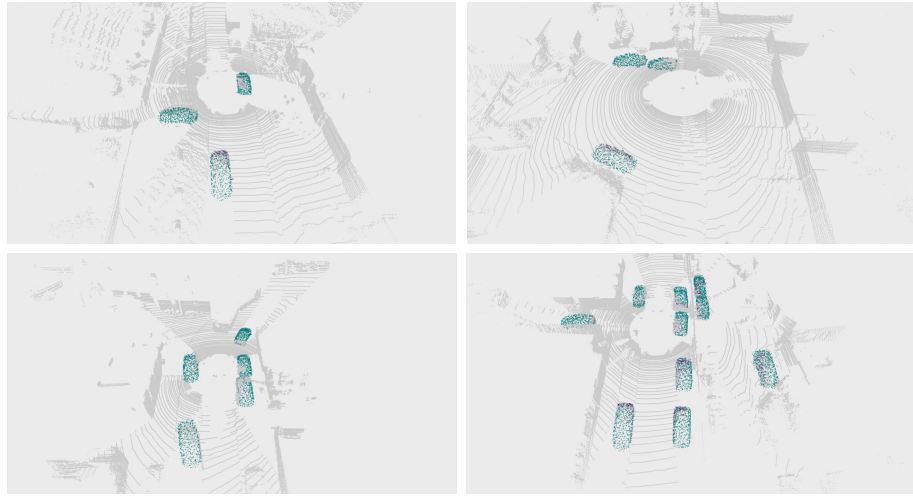
We transfer all models trained on our synthetic dataset to KITTI-360 [20], a large-scale outdoor dataset, to further evaluate generalization ability. Specifically, we take real-world LiDAR scans and segment out points within the bounding boxes of each frame. This preprocessing generates a total of 4382 partial observations spreading over the classes of *car*, *truck*, *caravan*, and *trailer*. All point clouds are represented in object coordinates and normalized into a unit sphere using the bounding box parameters. Since there is no ground truth point clouds on KITTI-360, we adopt the consistency metric introduced in [44]. It averages the CD results over multiple predictions of the same object in different frames. The results are listed in Table 2. It is obvious that our SA-Net generates more consistent point clouds, indicating that it can better deal with large variations in the input point clouds. The qualitative results on real scans are shown in Fig. 7.

## 5.3 Ablation Study

In this subsection, we further evaluate the influence of the number of rays, polar angles used in Spotlights. The controlled experiments are conducted on a subset of the ShapeNet dataset [3] containing 6 different categories. Here, we uniformly sample 16384 points on the mesh surface of each object as our ground truth. In addition, the impact of different sampling resolutions is measured through shape completion on our synthetic dataset.

**Table 2.** Consistency of comparisons on KITTI-360. We report the average CD of completed point clouds of the same target to evaluate the consistency of predictions.

| Model      | Consistency   |               |               |               |               |
|------------|---------------|---------------|---------------|---------------|---------------|
|            | Car           | Truck         | Caravan       | Trailer       | Average       |
| AtalsNet   | 0.0649        | 0.0674        | 0.0651        | 0.0710        | 0.0671        |
| PCN        | 0.0493        | 0.0612        | 0.0644        | 0.0580        | 0.0582        |
| FoldingNet | 0.0514        | 0.0611        | 0.0629        | 0.0595        | 0.0587        |
| TopNet     | 0.0513        | 0.0644        | 0.0648        | 0.0593        | 0.0599        |
| Ours       | <b>0.0410</b> | <b>0.0576</b> | <b>0.0613</b> | <b>0.0498</b> | <b>0.0524</b> |

**Fig. 7.** Qualitative results of point cloud completion on real scans in KITTI-360. The gray points are from raw scans. The purple points and the green points are partial inputs and completed results, respectively.

**Completeness of Sampled 3D Point Clouds** Following [32], we measure the completeness of the 3D point cloud sampled from Spotlights, which indicates how much of the object surface is covered by sample points. The metric again relies on the single-direction CD, however this time iterating through the ground truth point cloud finding the nearest point in the Spotlights representation.

For Spotlights, completeness is closely related to both the number of rays and the polar angle. Hence, we vary the number of rays from 2048 to 8192 and test 3 different polar angles under each setting. Note that the maximum polar angle in our case is about 83 degrees ( $r = \frac{1}{4}R$ ). The average score on each shape category is reported in Table 3. The results coincide with the intuition that more rays usually produce more points on the surface, whereas polar angles that are too large or too small can lead to degraded point cloud quality since most of the rays will either scatter into the void or concentrate on small local regions.

**Table 3.** Completeness of point clouds sampled from Spotlights. We report the average distance from the dense ground truth to the sampled point cloud with varying numbers of rays and polar angles.

| #Rays | Polar angle | Completeness  |               |               |               |               |               |               | Average |
|-------|-------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------|
|       |             | Bathtub       | Bottle        | Cabinet       | Chair         | Lamp          | Sofa          |               |         |
| 2048  | 30°         | 0.0474        | 0.0547        | 0.0678        | 0.0444        | 0.0426        | 0.0619        | 0.0506        |         |
|       | 60°         | 0.0349        | 0.0380        | 0.0450        | 0.0339        | 0.0352        | 0.0447        | 0.0376        |         |
|       | 83°         | 0.0382        | 0.0397        | 0.0457        | 0.0389        | 0.0407        | 0.0469        | 0.0415        |         |
| 4096  | 30°         | 0.0326        | 0.0367        | 0.0476        | 0.0302        | 0.0298        | 0.0450        | 0.0353        |         |
|       | 60°         | 0.0260        | 0.0278        | 0.0349        | 0.0253        | 0.0263        | 0.0346        | 0.0285        |         |
|       | 83°         | 0.0290        | 0.0297        | 0.0361        | 0.0293        | 0.0312        | 0.0367        | 0.0318        |         |
| 8192  | 30°         | 0.0233        | 0.0259        | 0.0345        | 0.0216        | 0.0212        | 0.0331        | 0.0255        |         |
|       | 60°         | <b>0.0197</b> | <b>0.0208</b> | <b>0.0279</b> | <b>0.0192</b> | <b>0.0200</b> | <b>0.0274</b> | <b>0.0220</b> |         |
|       | 83°         | 0.0222        | 0.0225        | 0.0294        | 0.0223        | 0.0244        | 0.0295        | 0.0248        |         |

**Table 4.** Hit ratio of rays with different polar angles. It evaluates how many rays actually hit the object.

| Polar angle | Hit ratio (%) |             |             |             |             |             | Average     |
|-------------|---------------|-------------|-------------|-------------|-------------|-------------|-------------|
|             | Bathtub       | Bottle      | Cabinet     | Chair       | Lamp        | Sofa        |             |
| 30°         | <b>93.8</b>   | <b>81.6</b> | <b>93.9</b> | <b>79.1</b> | <b>47.2</b> | <b>85.0</b> | <b>78.6</b> |
| 60°         | 63.4          | 45.2        | 67.0        | 49.1        | 31.5        | 57.1        | 50.9        |
| 83°         | 43.2          | 29.9        | 47.7        | 32.6        | 22.6        | 39.8        | 34.8        |

**Hit Ratio of Virtual Rays** In most cases, objects cannot fill the entire space bounded by the outer sphere of Spotlights, which results in wasted rays that have no intersection with the object. To measure the utilization of rays, we compute the hit ratio defined as the number of surface intersecting rays over the total number of casted rays. As shown in Table 4, the hit ratio is inversely proportional to the polar angle and much lower for thin objects such as lamps. Although small polar angles waste fewer rays, we strongly suggest not to use those values as completeness also needs to be considered.

**Sampling Resolution** As discussed in Section 3.2, we sample twice in Spotlights model to better control the ray directions. The total number of rays is then the multiplication of the number of primary and secondary sampling points. To evaluate the performance of Spotlights model with different sampling resolutions, we try several resolution combinations on the task of shape completion and report the accuracy as in Section 5.1 in Table 5. By comparing the first and the second row, we can find that changing the sampling resolutions has no big difference on performance as long as the total number of rays are the same. Also, the third row shows that adding more rays can increase the accuracy, which confirms our intuition.

**Table 5.** Accuracy of completed point clouds with different sampling resolutions on the synthetic dataset. For A\*B, A and B are the number of primary and secondary sampling points, respectively. Accuracy measures the single direction CD from the prediction to ground truth.

| Resolution | Accuracy       |                |                |                |
|------------|----------------|----------------|----------------|----------------|
|            | Hatchback      | Sedan          | SUV            | Average        |
| 32*64      | 0.02124        | 0.02775        | 0.01850        | 0.02133        |
| 64*32      | 0.02219        | 0.02771        | 0.01907        | 0.02179        |
| 256*64     | <b>0.02058</b> | <b>0.02700</b> | <b>0.01839</b> | <b>0.02098</b> |

## 5.4 Limitations

Although Spotlights representation is compact compared to 3D coordinate-based point cloud, it still has several limitations.

First, the ray bundle needs to be carefully organized to reach the balance between efficiency and accuracy. Since the rays spread across the space bounded by the sphere, around 50% to 60% of the rays fail to actually hit the object surface, which is a big waste of model capacity and also results in the sparsity of sampled point clouds.

Second, adjacent rays do not assure intersections in a local surface, which makes it hard to learn depth variation along rays, especially for complex concave objects. To address these issues, in future extensions, we will try a differentiable variant of our Spotlights model to actively learn the orientation of each ray and dynamically adjust it based on the observations.

## 6 Conclusions

In this paper, we propose a novel compact point cloud representation denoted *Spotlights*, which is useful for applications demanding for high efficiency. In contrast to prior art, the structure in our representation is not imposed in an implicit way using complex network architectures, but in an explicit way through a hand-crafted object surface sampling model, which is highly inspired by the spherical view arrangement in 3D object scanners. Also, no strong assumptions about the enclosed objects are being made, and a relatively homogeneous sampling distribution is achieved.

We demonstrate its potential in the context of shape completion where experiments on both synthetic and real dataset show that it can achieve competitive accuracy and consistency while being orders of magnitude faster. The fact that we directly predict depths along rays suggests that our representation has the ability to predict any shape of similar complexity. In future work, we will try to minimize the invalid rays in Spotlights model and figure out a better way to learn the depth variation along the ray.

## References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.J.: Learning representations and generative models for 3d point clouds. In: *Proceedings of the IEEE International Conference on Machine Learning (ICML)*. pp. 40–49 (2018)
2. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map slam. *IEEE Transactions on Robotics (T-RO)* **37**(6), 1874–1890 (2021)
3. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015)
4. Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On visual similarity based 3d model retrieval. In: *Computer graphics forum*. vol. 22, pp. 223–232. Wiley Online Library (2003)
5. Dai, A., Qi, C.R., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 5868–5877 (2017)
6. Davis, J., Marschner, S.R., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: *Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission* (2002)
7. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv preprint arXiv:2012.15712* **1**(2), 4 (2020)
8. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: *Proceedings of the 1st Annual Conference on Robot Learning*. pp. 1–16 (2017)
9. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 605–613 (2017)
10. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **32**, 1362–1376 (2010)
11. Garsthagen, R.: An open source, low-cost, multi camera full-body 3d scanner. In: *Proc. of 5th Int. Conf. on 3D Body Scanning Technologies*. pp. 174–183 (2014)
12. Groueix, T., Fisher, M., Kim, V., Russell, B., Aubry, M.: AtlasNet: A Papier-Mache Approach to Learning 3D Surface Generation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
13. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edn. (2003)
14. Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X.: PF-Net: Point fractal network for 3d point cloud completion. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 7662–7670 (2020)
15. Keinert, B., Innmann, M., Sängler, M., Stamminger, M.: Spherical fibonacci mapping. *ACM Transactions on Graphics (TOG)* **34**(6), 1–7 (2015)
16. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 12697–12705 (2019)
17. Li, P., Qin, T., Shen, S.: Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 646–661 (2018)

18. Li, R., Li, X., Hui, K.H., Fu, C.W.: Sp-gan:sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (Proc. SIGGRAPH)* **40**(4) (2021)
19. Li, Y., Dai, A., Guibas, L., Nießner, M.: Database-assisted object retrieval for real-time 3d reconstruction. In *Computer Graphics Forum* **34**, 435–446 (2015)
20. Liao, Y., Xie, J., Geiger, A.: KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv preprint arXiv:2109.13410* (2021)
21. McCormac, J., Clark, R., Bloesch, M., Davison, A., Leutenegger, S.: Fusion++: Volumetric object-level slam. In: *Proceedings of the International Conference on 3D Vision (3DV)*. pp. 32–41 (2018)
22. Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)* (2011)
23. Newcombe, R., Lovegrove, S., Davison, A.: Dtam: Dense tracking and mapping in real-time. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. pp. 2320–2327 (2011)
24. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
25. Pauly, M., Mitra, N.J., Giesen, J., Gross, M.H., Guibas, L.J.: Example-based 3d scan completion (EPFL-CONF-149337), 23–32 (2005)
26. Pesce, M., Galantucci, L., Percoco, G., Lavecchia, F.: A low-cost multi camera 3d scanning system for quality measurement of non-static subjects. *Procedia Cirp* **28**, 88–93 (2015)
27. Qi, C.R., Chen, X., Litany, O., Guibas, L.J.: Imvotenet: Boosting 3d object detection in point clouds with image votes. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 4404–4413 (2020)
28. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: *proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 9277–9286 (2019)
29. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 918–927 (2018)
30. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 652–660 (2017)
31. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* **30** (2017)
32. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. vol. 1, pp. 519–528. IEEE (2006)
33. Shan, T., Englot, B., Meyers, D.: LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In: *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)* (2020)
34. Shin, D., Fowlkes, C.C., Hoiem, D.: Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)



35. Shin, D., Fowlkes, C.C., Hoiem, D.: Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 3061–3069 (2018)
36. Sucar, E., Wada, K., Davison, A.: NodeSLAM: Neural object descriptors for multi-view shape reconstruction. In: *Proceedings of the International Conference on 3D Vision (3DV)* (2020)
37. Tchapmi, L.P., Kosaraju, V., Rezaatofghi, H., Reid, I., Savarese, S.: Topnet: Structural point cloud decoder. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 383–392 (2019)
38. Thrun, S., Wegbreit, B.: Shape from symmetry. In: *Proceedings of the International Conference on Computer Vision (ICCV)* (2005)
39. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point cloud completion by skip-attention network with hierarchical folding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
40. Wen, X., Xiang, P., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: PMP-Net: Point Cloud Completion by Learning Multi-step Point Moving Paths. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
41. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A Deep Representation for Volumetric Shapes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015)
42. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: GRNet: Gridding Residual Network for Dense Point Cloud Completion. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2020)
43. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 206–215 (2018)
44. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: PCN: Point completion network. In: *Proceedings of the International Conference on 3D Vision (3DV)*. pp. 728–737. IEEE (2018)