Local Implicit Grid Representations for 3D Scenes

Chiyu “Max” Jiang\textsuperscript{1,2} Avneesh Sud\textsuperscript{2} Ameesh Makadia\textsuperscript{2} Jingwei Huang\textsuperscript{2,3} Matthias Nießner\textsuperscript{4} Thomas Funkhouser\textsuperscript{2}

\textsuperscript{1}UC Berkeley \textsuperscript{2}Google Research \textsuperscript{3}Stanford University \textsuperscript{4}Technical University of Munich

![Diagram of 3D scene with labeled objects]

(a) Training parts from ShapeNet. (b) t-SNE plot of part embeddings. (c) Reconstructing entire scenes with Local Implicit Grids

Figure 1: We learn an embedding of parts from objects in ShapeNet \cite{3} using a part autoencoder with an implicit decoder. We show that this representation of parts is generalizable across object categories, and easily scalable to large scenes. By localizing implicit functions in a grid, we are able to reconstruct entire scenes from points via optimization of the latent grid.

Abstract

Shape priors learned from data are commonly used to reconstruct 3D objects from partial or noisy data. Yet no such shape priors are available for indoor scenes, since typical 3D autoencoders cannot handle their scale, complexity, or diversity. In this paper, we introduce Local Implicit Grid Representations, a new 3D shape representation designed for scalability and generality. The motivating idea is that most 3D surfaces share geometric details at some scale – i.e., at a scale smaller than an entire object and larger than a small patch. We train an autoencoder to learn an embedding of local crops of 3D shapes at that size. Then, we use the decoder as a component in a shape optimization that solves for a set of latent codes on a regular grid of overlapping crops such that an interpolation of the decoded local shapes matches a partial or noisy observation. We demonstrate the value of this proposed approach for 3D surface reconstruction from sparse point observations, showing significantly better results than alternative approaches.

1. Introduction

Geometric representation for scenes has been central to various tasks in computer vision and graphics, including geometric reconstruction, compression, and higher-level tasks such as scene understanding, object detection and segmentation. An effective representation should generalize well across a wide range of semantic categories, scale efficiently to large scenes, exhibit a rich expressive capacity for representing sharp features and complex topologies, and at the same time leverage learned geometric priors acquired from data.

In the last years, several works have proposed new network architectures to allow conventional geometric representations such as point clouds \cite{31, 13, 43}, meshes \cite{37, 15}, and voxel grids \cite{9, 40} to leverage data priors. More recently, a neural implicit representation \cite{4, 28, 29} has been proposed as an alternative to these approaches for its expressive capacity for representing fine geometric details. However, the aforementioned works focus on learning representations for whole objects within one or a few categories, and they have not been studied in the context of generalizing to other categories, or scaling to large scenes.
In this paper we propose a learned 3D shape representation that generalizes and scales to arbitrary scenes. Our key observation is that although different shapes across different categories and scenes have vastly different geometric forms and topologies on a global scale, they share similar features at a certain local scale. For instance, sofa seats and car windshields have a similar curved parts, tabletops and airplane wings both have thin sharp edges, etc.. While no two shapes are the same at the macro scale, and all shapes on a micro-scale can be locally approximated by an angled plane, there exists an intermediate scale (a “part scale”), where a meaningful shared abstraction for all geometries can be learned by a single deep neural network. We aim to learn shape priors at that scale and then leverage them in a scalable and general 3D reconstruction algorithm.

To this end, we propose the Local Implicit Grid (LIG) representation, a regular grid of overlapping part-sized local regions, each encoded with an implicit feature vector. We learn to encode/decode geometric parts of objects at a part scale by training an implicit function autoencoder on 13 object categories from ShapeNet [3]. Then, armed with the pretrained decoder, we propose a mechanism to optimize for the Latent Implicit Grid representation that matches a partial or noisy scene observation. Our representation includes a novel overlapping latent grid mechanism for confidence-weighted interpolation of learned local features for seamlessly representing large scenes. We illustrate the effectiveness of this approach by targeting the challenging application of scene reconstruction from sparse point samples, where we are able to faithfully reconstruct entire scenes given only sparse point samples and shape features learned from ShapeNet objects. Such an approach requires no training on scene level data, where data is costly to acquire. We achieve significant improvement both visually and quantitatively in comparison to state-of-the-art reconstruction algorithms for the scene reconstruction from point samples task (Poisson Surface Reconstruction [23, 24], or PSR, among other methods).

In summary, the main contributions of this work are:

- We propose the Local Implicit Grid representation for geometry, where we learn and leverage geometric features on a part level, and associated methods such as the overlapping latent grid mechanism and latent grid optimization methods for representing and reconstructing scenes at high fidelity.

- We illustrate the significantly improved generalizability of our part-based approach in comparison to related methods that learn priors for entire objects — i.e., we can reconstruct shapes from novel object classes after training only on chairs, or construct entire scenes after training only on ShapeNet parts.

- We apply our novel shape representation approach towards the challenging task of scene reconstruction from sparse point samples, and show significant improvement over the state-of-the-art approach (For Matterport reconstruction from 100/m² input points, an F-Score of 0.889 versus 0.455.

2. Related Work

2.1. Geometric representation for objects

In computer vision and graphics, geometric representations such as simplicial complexes (point clouds, line meshes, triangular meshes, tetrahedral meshes) have long been used for representing geometries for its flexibility and compactness. In recent years, various neural architectures have been proposed for analyzing or generating such representations. For instance for [31, 38] have been proposed for analyzing point cloud representations, and [13, 43] for generating point clouds. [27, 17, 20, 19] have been proposed for analyzing signals on meshes, and [37, 15, 7] for generating mesh representations. [21] proposed a general framework for analyzing arbitrary simplicial complex based geometric signals. Naturally paired with 3D Convolutional Neural Networks (CNNs), voxel grids have also been extensively used as a 3D representation [41, 8, 5].

More recently, alternative representations have been proposed in the context of shape generation. Most related to our method are [28, 29, 4], where the implicit surfaces of geometries are represented as spatial functions using fully-connected neural networks. Continuous spatial coordinates are fed as input features to the network which directly produces the values of the implicit functions, however these methods encode the entire shape using a global latent code. [33] used such implicit networks to represent neural features instead of occupancies that can be combined with a differentiable ray marching algorithm to produce neural renderings of objects. Rather than learning a single global implicit network to represent the entire shape, [32] learns a continuous per-pixel occupancy and color representation using implicit networks. Other novel geometric representations in the context of shape reconstruction include Structured Implicit Functions that serves as learned local shape templates [14], and CvxNet [10] which represents space as a convex combination of half-planes that are localized in space. These methods represent entire shapes using a single global latent vector, which can be decoded into continuous outputs with the associated implicit networks.

2.2. Localized geometric representations

Though using a single global latent code to represent entire geometries and scenes is appealing for its simplicity, it fails to capture localized details, and scales poorly to large scenes with increased complexities. [42] proposes to address the localization problem in the context of image to 3D

6002
reconstruction by first estimating a camera pose for the images followed by the projection of local 2D features to be concatenated with global latents for decoding. However, the scalability of such hybrid representations beyond single objects has yet to be shown. Similar to our approach, [39] uses a local patch based representation. However it is not trained on any data, hence is not able to leverage any shape priors from 3d datasets. [30] combines shape patches extracted directly from a set of examples, which limits the shape expressibility. Similar to our spatial partitioning of geometries into part grids, [36] uses PCA-based decomposition to learn a reduced representation of geometric parts within TSDF grids of a fixed scale for the application of real-time geometry compression. These methods do not support scalable reconstruction with learned deep implicit functions.

2.3. Scene-level geometry reconstruction

Most deep learning studies have investigated object reconstruction, with input either as an RGB/D image [5, 37, 28, 4, 13, 10, 14] or 3D points [29, 26, 22], and yet few have considered learning to reconstruct full scenes. Scene level geometry reconstruction is a much more challenging task in comparison to single objects. [34] performs semantic scene completion within the frustum of a single depth image. [8] uses a 3D convolutional network with a coarse-to-fine inference strategy to directly regress gridded Truncated Signed Distance Function (TSDF) outputs from incomplete input TSDF. [1] tackles the scene reconstruction problem by CAD model retrieval, which produces attractive surfaces, at the expense of geometric inaccuracies. However, all of the methods require training on reliable and high-quality scene data. Though several real and synthetic scene datasets exist, such as SunCG [35], SceneNet [16], Matterport3D [2], and ScanNet [6], they are domain-specific and acquiring data for new scenes can be costly. In contrast to methods above that require training on scene dataset, our method naturally generalizes shape priors learned from object datasets and does not require additional training on scenes.

3. Methods

3.1. Method overview

We present a schematic overview of our method in Figure 1. We first learn an embedding of shape parts at a fixed scale from objects in a synthetic dataset using part autoencoders (see Sec. 3.2). We show two interesting properties of such a latent embedding: (1) objects that originated from different categories share similar part geometries, validating the generalizability of such learned representations, and (2) parts that are similar in shape are close in the latent space. In order to scale to scenes of arbitrary sizes, we introduce an overlapping gridded representation that can layout these local representations in a scene (Sec. 3.3). Using such part embeddings that can be continuously decoded spatially using a local implicit network, we are able to faithfully reconstruct geometries from only sparse oriented point samples by searching for a corresponding latent code using gradient descent-based optimization to match given observations (Sec. 3.4), thus efficiently leveraging geometric priors learned from parts of the ShapeNet dataset.

3.2. Learning a latent embedding for parts

Data Our part embedding model is learned from a collection of 20 million object parts culled from 3D-R2N2 [5], a 13-class subset of ShapeNet. As preprocessing, we normalize watertight meshes (generated with tools from [28]) into a [0, 1] unit cube, leaving a margin of 0.1 at each side. To maintain the fidelity of the parts, we compute a signed distance function (SDF) at a grid resolution of 256^3. Starting from the origin and with a stride of 16, all 32^3 patches that have at least one point within 3/255 of the shape surface are extracted as parts for training.

Part Autoencoder We use a 3D CNN decorated with residual blocks for encoding such local TSDF grids, and a reduced IM-NET [4] decoder for reconstructing the part (See Fig. 2). An IM-NET decoder is a simple fully connected neural network with internal skip connections that takes in a latent code concatenated with a 3D point coordinate, and outputs the corresponding implicit function value at the point. We train the network using point samples with binary in/out labels so that the network learns a continuous decision boundary of the binary classifier as the encoded surface. Since decoding a part is a much more simplified task than decoding an entire shape, we reduce the number

Figure 2: A schematic of the part autoencoder. At training, crops of the TSDF grid from the ShapeNet dataset are used to train a part autoencoder, with a 3D CNN encoder and implicit network decoder. Interior and exterior points are sampled to supervise the network during training. At inference time, the pre-trained implicit network is attached to a Local Implicit Grid, and the corresponding latent values are optimized via gradient descent on observed interior/exterior points.

6003
of feature channels in each hidden layer of IM-NET by 4 fold, obtaining a leaner and more efficient decoder. To acquire a compact latent representation of parts, we further reduce the number of latent channels for each part to 32. We train the part autoencoder with 2048 random point samples that we sample from the SDF grid on-the-fly during training, where we sample points farther from the boundary with Gaussian-decaying probabilities. The sign of the sample points is interpolated from the sign of the original SDF grid. Furthermore, we truncate the input SDF grids to a value of 3/255 and renormalize the grid to [0, 1] for stronger gradients near the boundary.

We train the part autoencoder with binary cross entropy loss on the point samples, with an additional latent regularization loss to constrain the latent space of the learned embeddings. The loss is given as:

\[
\mathcal{L}(\theta_c, \theta_d) = \frac{1}{|\mathcal{P}||\mathcal{B}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{B}} \mathcal{L}_c(D_{\theta_c}(x_{i,j}, E_{\theta_e}(g_i)), \text{sign}(x_{i,j})) + \lambda ||E_{\theta_e}(g_i)||_2
\]

(1)

where \(\mathcal{P}\) is the set of all training parts in a given mini-batch, \(\mathcal{B}\) is the set of point samples sampled per part, \(\mathcal{L}_c(\cdot, \cdot)\) is the binary cross-entropy loss with logits, \(E_{\theta_e}\) is the convolutional encoder parameterized by trainable parameters \(\theta_e\), \(D_{\theta_d}\) is the implicit decoder parameterized by trainable parameters \(\theta_d\), and \(g_i\) is the input tsdf grid for the \(i\)-th part, \(\text{sign}(\cdot)\) takes the sign of the corresponding point \(x_{i,j}\).

### 3.3. Local implicit grids

In order to use the learned part representations for representing entire objects and scenes, we lay out a sparse latent grid structure, where within each local grid cell the surface is continuously decoded from the local latent codes within the cell. In world coordinates, when querying for the implicit function value at location \(x\) against a single voxel grid cell centered at \(x_i\), the implicit value is decoded as:

\[
f(x, c_i) = D_{\theta_d}(c_i, \frac{2}{s}(x - x_i))
\]

(2)

where \(c_i\) is the latent code corresponding to the part in cell \(i\), and \(s\) is the part scale. The coordinates are first being transformed into normalized local coordinates within the cell to \([-1, 1]\), before being queried against the decoder.

Though directly partitioning space into a voxel grid with latent channels within each cell gives decent performance, there will be discontinuities across voxel boundaries. Hence we propose the overlapping latent grid scheme, where each grid cell for a part overlaps with its neighboring cells by half the part scale (see Fig. 3). When querying for the implicit function value at an arbitrary position \(x\) against overlapping latent grids, the value is computed as a trilinear interpolation of independent queries to all cells that overlap at this position, which is 4 in 2 dimensions and 8 in 3 dimensions:

\[
f(x, \{c_j | j \in \mathcal{N}_j\}) = \sum_{j \in \mathcal{N}} w_j D_{\theta_d}(c_j, \frac{2}{s}(x - x_j))
\]

(3)

where \(\mathcal{N}_j\) is the set of all neighboring cells of point \(x\), and \(w_j\) is the trilinear interpolation weight corresponding to cell \(j\). Under such an interpolation scheme, the overall function represented by the implicit grid is guaranteed to be \(C^0\) continuous. Higher-order continuity could be similarly acquired with higher degrees of polynomial interpolations, though we do not explore it in the scope of this study. For additional efficiency, since most grid cells do not have any points that fall into them, we use a sparse data structure for storing latent grid values, optimization, and decoding for the reconstructed surface, where empty space is assumed to be exterior space.

### 3.4. Geometric encoding via latent optimization

At inference time, when presented with a sparse point cloud of interior/exterior samples as input, we decompose space into a coarse grid and then perform optimization for the latent vectors associated with the grid cells in order to minimize the cost function for classifying sampled interior/exterior points. The initial values within the latent grid is initialized as random normal with a standard deviation of \(10^{-2}\). If we denote the set of effective latent grid cells as \(\mathcal{G}\), the corresponding latent code in each grid cell \(c_j\), and the set of all sampled interior/exterior input points as \(\mathcal{B}\), we optimize the latent codes for the minimal classification loss on the sampled points:

\[
\arg \min_{c_i \in \mathcal{G}} \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}_i} \mathcal{L}_c(f(x_i, \{c_j | j \in \mathcal{N}_j\}), \text{sign}(x_i)) + \lambda ||c_j||_2
\]

(4)

How do we acquire the signed point samples for performing this latent grid optimization? For autoencoding a geometry with a latent grid, the signed point samples are densely sampled near the surface of the given shape to be
encoded. However, for the application of recovering surface geometry from sparse oriented point samples, we randomly sample interior and exterior points for each point sample along the given normal direction, with a Gaussian falloff probability parameterized by a standard deviation $\sigma$. The latent codes within the overlapping latent grids are updated via optimization for minimizing classification loss as in Eqn. 4. The surface of the shape is reconstructed by densely querying the latent grid and extracting the zero-contour of the output logit.

As our method requires optimizing over the learned latent space, it is reasonable to wonder if alternate models such as a variational autoencoder [25] or autodecoder [29] would be a more appropriate choice, as both formulations incorporate a latent distribution prior. However, [29] observed the stochastic nature of the VAE made training difficult. Also, the autodecoder is fundamentally unable to scale to large numbers of parts at training as it requires fast storage and random access to all latent embeddings during training. These concerns motivated our decision to adopt an autoencoder formulation with a regularization loss to constrain the latent space.

### Table 1: Shape autoencoding for autoencoders trained on only chairs and evaluated on all 13 categories. The mean corresponds to class-averaged mean of all out-of-training object categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>IM-NET CD ($\downarrow$)</th>
<th>IM-NET Normal ($\uparrow$)</th>
<th>IM-NET F-Score ($\uparrow$)</th>
<th>Ours CD ($\downarrow$)</th>
<th>Ours Normal ($\uparrow$)</th>
<th>Ours F-Score ($\uparrow$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>chair</td>
<td>0.181</td>
<td>0.820</td>
<td>0.505</td>
<td>0.150</td>
<td>0.817</td>
<td>0.564</td>
</tr>
<tr>
<td>airplane</td>
<td>0.698</td>
<td>0.550</td>
<td>0.151</td>
<td>0.150</td>
<td>0.817</td>
<td>0.564</td>
</tr>
<tr>
<td>bench</td>
<td>0.229</td>
<td>0.719</td>
<td>0.433</td>
<td>0.054</td>
<td>0.905</td>
<td>0.857</td>
</tr>
<tr>
<td>cabinet</td>
<td>0.343</td>
<td>0.700</td>
<td>0.250</td>
<td>0.118</td>
<td>0.948</td>
<td>0.733</td>
</tr>
<tr>
<td>car</td>
<td>0.354</td>
<td>0.646</td>
<td>0.240</td>
<td>0.152</td>
<td>0.825</td>
<td>0.472</td>
</tr>
<tr>
<td>display</td>
<td>0.601</td>
<td>0.574</td>
<td>0.130</td>
<td>0.170</td>
<td>0.926</td>
<td>0.551</td>
</tr>
<tr>
<td>lamp</td>
<td>0.836</td>
<td>0.592</td>
<td>0.120</td>
<td>0.114</td>
<td>0.882</td>
<td>0.624</td>
</tr>
<tr>
<td>loudspeaker</td>
<td>0.377</td>
<td>0.702</td>
<td>0.246</td>
<td>0.139</td>
<td>0.937</td>
<td>0.711</td>
</tr>
<tr>
<td>rifle</td>
<td>0.902</td>
<td>0.400</td>
<td>0.080</td>
<td>0.113</td>
<td>0.824</td>
<td>0.693</td>
</tr>
<tr>
<td>sofa</td>
<td>0.199</td>
<td>0.812</td>
<td>0.484</td>
<td>0.077</td>
<td>0.944</td>
<td>0.822</td>
</tr>
<tr>
<td>table</td>
<td>0.425</td>
<td>0.681</td>
<td>0.242</td>
<td>0.066</td>
<td>0.936</td>
<td>0.844</td>
</tr>
<tr>
<td>telephone</td>
<td>0.623</td>
<td>0.547</td>
<td>0.120</td>
<td>0.037</td>
<td>0.984</td>
<td>0.962</td>
</tr>
<tr>
<td>vessel</td>
<td>0.591</td>
<td>0.574</td>
<td>0.147</td>
<td>0.178</td>
<td>0.847</td>
<td>0.467</td>
</tr>
</tbody>
</table>

mean*: 0.435, 0.666, 0.274  Ours: 0.114, 0.898, 0.692

### 4. Experiments

We ran a series of experiments to test the proposed LIG method. We focus on two properties of our method: the generalization of our learned part representation, and the scalability of our learned shape representation to large scenes. Our target application is reconstructing scenes from a sparse set of oriented point samples, a challenging task that requires learned part priors for detailed and accurate reconstruction.

**Metrics** In all of our experiments, we evaluate geometric reconstruction quality with Chamfer Distance (CD), Normal Alignment (Normal), and F-Score. For Chamfer Distance and Normal Alignment, we base our implementation on [28] with small differences. For object-level autoencoding experiments, we follow [13, 28] and normalize the unit distance to be 1/10 of the maximal edge length of the current object’s bounding box. We estimate CD and Normal Alignment using 100,000 randomly sampled points on the ground truth and reconstructed meshes. For the two scene-level experiments, we randomly sample 2 million points on each mesh when estimating CD and Normal Alignment. When evaluating scene reconstructions, we use world coordinate scales (meters) for computing CD, since data is provided in a physically-meaningful scale. Additionally, in all experiments, we compute the F-Score at a threshold of $\tau$, as F-Score is a metric less sensitive to outliers. F-Score is the harmonic mean of recall (percentage of reconstruction to target distances under $\tau$) and precision (vice versa). For object reconstruction (Sec. 3.2) we use $\tau = 0.1$ and for scene reconstruction, we use $\tau = 0.025$ (i.e., 2.5cm).
### Table 2: Qualitative comparison of scene representational performance for IM-NET versus our method.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>CD(↑)</th>
<th>Normal(↑)</th>
<th>F-Score(↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM-NET</td>
<td>0.183</td>
<td>0.827</td>
<td>0.647</td>
</tr>
<tr>
<td>Ours</td>
<td>0.007</td>
<td>0.945</td>
<td>0.985</td>
</tr>
</tbody>
</table>

#### 4.1. Generalization of learned part representation

**Task** In order to investigate the generalization of the learned embedding by reducing the scale of the learned shape from object scale to part scale, we construct an investigative experiment of training the models to learn a shape autoencoder on a single category of objects (in this case, chairs in the training set of ShapeNet), and reconstructing examples from the all 13 object categories, including the other 12 unseen categories.

**Baseline** As our main objective is to explore the gain in generalizability from learning an embedding of part scales, we benchmark our method against the original IM-NET decoder with a similar 3D convolution based encoder as the encoder part of our part autoencoder. To implement autoencoding for our method, we train our autoencoder on all the parts we extract from the training split of the chair category in ShapeNet. We then “encode” the geometries of the unseen shapes using the latent optimization method that is described in Sec. 3.4.

**Results Discussion** We quantitatively and qualitatively compare reconstruction performances in Table 1 and Figure 5, respectively. Given an IM-NET that is trained to learn a latent representation of objects (in this scenario, chairs), the learned representation does not generalize to classes beyond the source class. Visually, IM-NET achieves good reconstructions on the source class as well as related classes (e.g., sofa), but performs poorly on semantically different classes (e.g., airplane). In contrast, the part representation learned by our local implicit networks is transferable across drastically different object categories.

#### 4.2. Scalability of scene representational power

**Task** As a second experiment, we investigate the increased representational power and scalability that we gain from learning a part-based shape embedding. The definition of the task is: given one scene, what is the best reconstruction performance we can get from either representation for memorizing and overfitting to the scene.

**Baseline** Similar to the previous experiment, we compare directly with IM-NET for representational capacity towards a scene, as it is the decoder backbone that our method is based on, to investigate the improvement in scalability that we are able to gain by distributing geometric information in

Figure 5: Qualitative comparison of autoencoded shape from in-category (chair) and out-of-category shapes. IM-NET trained to learn embeddings of one object category does not transfer well to unseen categories, while the part embedding learned by our local implicit networks is much more transferable across unseen categories.

Figure 6: Qualitative comparison of the scene representational performance: Left to right: Ground truth scene, our reconstruction using sampling density 500 points/m², and IM-NET. First two rows from Matterport, last row from SceneNet.
spatially localized grid cells versus a single global representation. For this task, as the objective is to encode one scene, we use the encoderless version of IM-NET, where during training time, the decoder only receives spatial coordinates of point samples (not concatenated with a latent code) that are paired with the signs of these points. For our method, we use latent optimization against the pretrained decoder for encoding the scenes, using 100k surface point samples from the scene, with a sampling factor of $k = 10$ per point along the normal direction.

Data We evaluate the representational qualities of the two methods on the meshes from the validation set of the Matterport3D scene dataset. We perform the evaluations at the region level of the dataset, requiring the models to encode one region at a time. Additionally, we provide one example from SceneNet for visual comparison in Fig. 6.

Results Discussion The quantitative (Table 2) and qualitative (Fig. 6) results are presented. While IM-NET is able to reconstruct the general structure of indoor scenes such as smooth walls and floors, it fails to capture fine details of objects due to the difficulty of scaling a single implicit network to an entire scene. Our Local Implicit Grids are able to capture global structures as well as local details.

4.3. Scene reconstruction from sparse points

Task As a final task and our main application, we apply our reconstruction method to the classic task in computer graphics to reconstruct geometries from sparse points. This is an important application since surface reconstruction from points is a crucial step in the process of digitizing the 3-dimensional world. The input to the reconstruction pipeline is the sparse point samples that we randomly sample from the surface mesh of the scene datasets. We study reconstruction performances with a varied number of input point samples and point densities.

Baseline We mainly compare our method to the traditional Poisson Surface Reconstruction (PSR) method [23, 24] with a high octree depth value (depth=10) for the scene reconstruction experiment, which remains the state-of-the-art method for surface reconstruction tasks of scenes. We also compare with other classic (PSR at depth 8 and 9, Alpha Complex [11], Ball Pivoting [12]) and deep (Deep Geometric Prior [39]) reconstruction methods on one representative scenario (see 100pts/m² in Table 3) due to the high computational cost of evaluating all methods on all scenes. While various other deep learning-based methods [29, 26, 22] have been proposed for surface reconstruction from points in a similar setting, all of the deep learning-based methods are object-specific, trained and tested on specific object categories in ShapeNet, with no anticipated transferability to unseen categories or scenes, as we have shown in the experiment in Sec. 4.1. Furthermore, as both PSR and our method require no training/finetuning on the scene level datasets, the task is based on the premise that high-quality 3D training data is costly to acquire or unavailable for scenes. For our method, we adaptively use different part sizes for different point densities. We use 25cm (1000 pts/m²), 35cm (500 pts/m²), 50cm (100 pts/m²) and 75cm (20 pts/m²) corresponding to different point densities for optimal performance.

Data We evaluate the reconstruction performance of the methods on a synthetic dataset: SceneNet [16], and a
Table 3: Reconstruction performance on SceneNet dataset.

<table>
<thead>
<tr>
<th>points/m²</th>
<th>Method</th>
<th>CD(↓)</th>
<th>Normal(↑)</th>
<th>F-Score(↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>PSR10</td>
<td>0.167</td>
<td>0.655</td>
<td>0.276</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.028</td>
<td>0.813</td>
<td>0.691</td>
</tr>
<tr>
<td>100</td>
<td>PSR10</td>
<td>0.106</td>
<td>0.757</td>
<td>0.455</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.013</td>
<td>0.883</td>
<td>0.889</td>
</tr>
<tr>
<td>500</td>
<td>PSR10</td>
<td>0.103</td>
<td>0.871</td>
<td>0.778</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.008</td>
<td>0.928</td>
<td>0.970</td>
</tr>
<tr>
<td>1000</td>
<td>PSR10</td>
<td>0.102</td>
<td>0.910</td>
<td>0.862</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.007</td>
<td>0.945</td>
<td>0.985</td>
</tr>
</tbody>
</table>

Table 4: Reconstruction performance on Matterport dataset.

<table>
<thead>
<tr>
<th>CL</th>
<th>PS</th>
<th>Overlap</th>
<th>CD(↓)</th>
<th>Normal(↑)</th>
<th>F-Score(↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>25cm</td>
<td>Yes</td>
<td>0.013</td>
<td>0.948</td>
<td>0.921</td>
</tr>
<tr>
<td>32</td>
<td>50cm</td>
<td>Yes</td>
<td>0.012</td>
<td>0.961</td>
<td>0.957</td>
</tr>
<tr>
<td>32</td>
<td>75cm</td>
<td>Yes</td>
<td>0.013</td>
<td>0.945</td>
<td>0.929</td>
</tr>
<tr>
<td>32</td>
<td>50cm</td>
<td>No</td>
<td>0.023</td>
<td>0.886</td>
<td>0.857</td>
</tr>
</tbody>
</table>

Table 5: Ablation study on the effects of the choice of latent code length (CL), part scale (PS), and overlapping latent grid design on the reconstruction performance for scenes.

5. Ablation Study

Additionally, we study the effects of two important aspects of our method: the part scale that we choose for reconstructing each scene, and overlapping latent grids. We choose SceneNet reconstruction from 100 point samples / m² as a representative case for the ablation study. See Table 5 for a comparison. As seen from the results, the reconstruction results are affected by the choice of part scale, albeit not very heavily influenced. Overlapping latent grids significantly improves the quality of the overall reconstruction. With a smaller latent code size of 8, the performance is slightly deteriorated due to more limited expressivity for part geometries.

6. Discussion and Future Work

The Local Implicit Grid (LIG) representation for 3D scenes is a regular grid of overlapping part-sized local regions, each encoded with an implicit feature vector. Experiments show that LIG is capable of reconstructing 3D surfaces of objects from classes unseen in training. Furthermore, to our knowledge, it is the first learned 3D representation for reconstructing scenes from sparse point sets in a scalable manner. Topics for future work include ways to constrain the LIG optimization to produce latent codes near training examples, explore alternate implicit function representations (e.g., OccNet), and to investigate the best ways to use LIG for 3D reconstruction from image(s).

Acknowledgements

We would like to thank Kyle Genova, Fang Yin Wei, Abhijit Kundu, Alireza Fathi, Caroline Pantofaru, David Ross, Yue Wang, Mahyar Najibi and Chris Bregler for helpful discussions, Angela Dai for help with supplemental video, JP Lewis for offering help in paper review, as well as anonymous reviewers for helpful feedback. This work was supported by the ERC Starting Grant Scan2CAD (804724).
References


