Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction

Rohan Chabra[1,3,*], Jan E. Lenssen[2,3,*], Eddy Ilg[3], Tanner Schmidt[3], Julian Straub[3], Steven Lovegrove[3], and Richard Newcombe[3]

[1] University of North Carolina at Chapel Hill
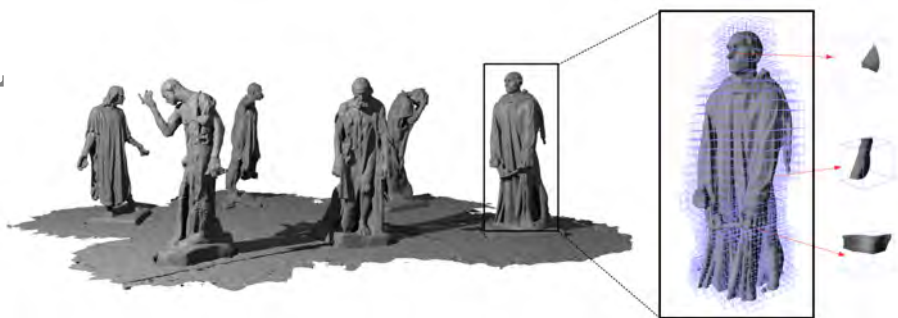[2] TU Dortmund University
[3] Facebook Reality Labs

Fig. 1: Reconstruction performed by our Deep Local Shapes (DeepLS) of the Burghers of Calais scene [59]. DeepLS represents surface geometry as a sparse set of local latent codes in a voxel grid, as shown on the right. Each code compresses a local volumetric SDF function, which is reconstructed by an implicit neural network decoder.

Abstract. Efficiently reconstructing complex and intricate surfaces at scale is a long-standing goal in machine perception. To address this problem we introduce Deep Local Shapes (DeepLS), a deep shape representation that enables encoding and reconstruction of high-quality 3D shapes without prohibitive memory requirements. DeepLS replaces the dense volumetric signed distance function (SDF) representation used in traditional surface reconstruction systems with a set of locally learned continuous SDFs defined by a neural network, inspired by recent work such as DeepSDF. Unlike DeepSDF, which represents an object-level SDF with a neural network and a single latent code, we store a grid of independent latent codes, each responsible for storing information about surfaces in a small local neighborhood. This decomposition of scenes into local shapes simplifies the prior distribution that the network must learn, and also enables efficient inference. We demonstrate the effectiveness and generalization power of DeepLS by showing object shape encoding and reconstructions of full scenes, where DeepLS delivers high compression, accuracy, and local shape completion.

---

1   Introduction

A signed distance function (SDF) represents three-dimensional surfaces as the zero-level set of a continuous scalar field. This representation has been used by many classical methods to represent and optimize geometry based on raw sensor observations [12,31,48,37,38]. In a typical use case, an SDF is approximated by storing values on a regularly-spaced voxel grid and computing intermediate values using linear interpolation. Depth observations can then be used to infer these values and a series of such observations are combined to infer the most likely SDF using a process called fusion.

Voxelized SDFs have been widely adopted and used successfully in a number of applications, but they have some fundamental limitations. First, the dense voxel representation requires significant amounts of memory (typically on a resource-constrained parallel computing device), which imposes constraints on resolution and the spatial extent that can be represented. These limits on resolution, as well as sensor limitations, typically lead to surface estimates that are missing thin structures and fine surface details. Second, as a non-parametric representation, SDF fusion can only infer surfaces that have been directly observed. Some surfaces are difficult or impossible for a typical range sensor to capture, and observing every surface in a typical environment is a challenging task. As a result, reconstructions produced by SDF fusion are often incomplete.

Recently, deep neural networks have been explored as an alternative representation for signed distance functions. According to the universal approximation theorem [26], a neural network can be used to approximate any continuous function, including signed distance functions [34,40,35,10]. With such models, the level of detail that can be represented is limited only by the capacity and architecture of the network. In addition, a neural network can be made to represent not a single surface but a family of surfaces by, for example, conditioning the function on a latent code. Such a network can then be used as a parametric model capable of estimating the most likely surface given only partial noisy observations. Incorporating shape priors in this way allows us to move from the maximum likelihood (ML) estimation of classical reconstruction techniques to potentially more robust reconstruction via maximum a posteriori (MAP) inference.

These neural network representations have their own limitations, however. Most of the prior work on learning SDFs is object-centric and does not trivially scale to the detail required for scene-level representations. This is likely due to the global co-dependence of the SDF values at any two locations in space, which are computed using a shared network and a shared parameterization. Furthermore, while the ability of these networks to learn distributions over classes of shapes allows for robust completion of novel instances from known classes, it does not easily generalize to novel classes or objects, which would be necessary for applications in scene reconstruction. In scanned real-world scenes, the diversity of objects and object setups is usually too high to be covered by an object-centric training data distribution.

Contribution. In this work, we introduce Deep Local Shapes (DeepLS) to combine the benefits of both worlds, exposing a trade-off between the prior-based MAP inference of memory efficient deep global representations (e.g., DeepSDF), and the detail preservation of computationally efficient, explicit volumetric SDFs. We divide space into a regular grid of voxels, each with a small latent code representing signed distance functions in local coordinate frames and making use of learned local shape priors. These voxels can be larger than is typical in fusion systems without sacrificing on the level of surface detail that can be represented (c.f. Sec. 5.2), increasing memory efficiency. The proposed representation has several favorable properties, which are verified in our evaluations on several types of input data:

1. It relies on readily available local shape patches as training data and generalizes to a large variety of shapes,
2. provides significantly finer reconstruction and orders of magnitude faster inference than global, object-centric methods like DeepSDF, and
3. outperforms existing approaches in dense 3D reconstruction from partial observations, showing thin details with significantly better surface completion and high compression.

2   Related Work

The key contribution of this paper is the application of learned local shape priors for reconstruction of 3D surfaces. This section will therefore discuss related work on traditional representations for surface reconstruction, learned shape representations, and local shape priors.

2.1   Traditional Shape Representations

Traditionally, scene representation methods can broadly be categorized into two categories, namely local and global approaches.

Local approaches. Most implicit surface representations from unorganized point sets are based on Blinn's idea of blending local implicit primitives [6]. Hope et al. [25] explicitly defined implicit surfaces by the tangent of the normals of the input points. Ohtake et al. [39] established more control over the local shape functions using quadratic surface fitting and blended these in a multi-scale partition of unity scheme. Curless and Levoy [12] introduced volumetric integration of scalar approximations of implicit SDFs in regular grids. This technique was further extended into real-time systems [31,48,37,38]. Surfaces are also shown to be represented by surfels, i.e. oriented planar surface patches [41,30,53].

Global approaches. Global implicit function approximation methods aim to approximate single continuous signed distance functions using, for example, kernel-based techniques [8,28,51,17]. Visibility or free space methods estimate which subset of 3D space is occupied, often by subdividing space into distinct tetrahedra [32,27,4]. These methods aim to solve for a globally view consistent surface representation.

Our work falls into the local surface representation category. It is related to the partition of unity approach [39], however, instead of using quadratic functions as local shapes, we use data-driven local priors to approximate implicit SDFs, which are robust to noise and can locally complete supported surfaces. While we also experimented with partition of unity blending of neighboring local shapes, we found it to be not required in practice, since our training formulation already includes border consistency (c.f. Sec 4.1), thus saving function evaluations during decoding. In comparison to volumetric SDF integration methods, such as SDF Fusion [38], our approach provides better shape completion and denoising, while at the same time uses less memory to store the representation. Unlike point- or surfel-based methods, our method leads to smooth and connected surfaces.

2.2   Learned Shape Representations

Recently there has been lot of work on 3D shape learning using deep neural networks. This class of work can also be classified into four categories: point-based methods, mesh-based methods, voxel-based methods and continuous implicit function-based methods.

Points. The methods use generative point cloud models for scene representation [3,57,58]. Typically, a neural network is trained to directly regress 3D coordinates of points in the point cloud.

Voxels. These methods provide non-parametric shape representation using 3D voxel grids which store either occupancy [55,11] or SDF information [14,49,33], similarly to the traditional techniques discussed above. These methods thus inherit the limitations of traditional voxel representations with respect to high memory requirements. Octree-based methods [50,43,24] relax the compute and memory limitations of dense voxel methods to some degree and have been shown on voxel resolutions of up to 512^3.

Meshes. These methods use existing [46] or learned [22,5] parameterization techniques to describe 3D surfaces by morphing 2D planes. When using mesh representations, there is a tradeoff between the ability to support arbitrary topology and the ability to reconstruct smooth and connected surfaces. Works such as [46,5] are variations on deforming a sphere into more complex 3D shape, which produces smooth and connected shapes but limits the topology to shapes that are homeomorphic to the sphere. AtlasNet, on the other hand, warps multiple 2D planes into 3D which together form a shape of arbitrary topology, but this results in disconnected surfaces. Other works, such as Scan2Mesh [13] and Mesh-RCNN[21], use deep networks to predict meshes corresponding to range scans or RGB images, respectively.

Implicit Functions. Very recently, there has been significant work on learning continuous implicit functions for shape representations. Occupancy Networks [34] and PiFU [44] represent shapes using continuous indicator functions which specify which subset of 3D space the shapes occupy. Similarly, DeepSDF [40] approximates shapes using Signed Distance Fields. We adopt the DeepSDF model as the backbone architecture for our local shape network.

Much of the work in this area has focused on learning object-level representations. This is especially useful when given partial observations of a known class, as the learned priors can often complete the shape with surprising accuracy. However, this also introduces two key difficulties. First, the object-level context means that generalization will be limited by the extent of the training set – objects outside of the training distribution may not be well reconstructed. Second, object-level methods do not trivially scale to full scenes composed of many objects as well as surfaces (e.g. walls and floors). In contrast, DeepLS maintains separate representations for small, distinct regions of space, which allows it to scale easily. Furthermore, the local representation makes it easier to compile a representative training set; at a small enough scale most surfaces have similar structure.

2.3   Local Shape Priors

In early work on using local shape priors, Gal et al. [18] used a database of local surface patches to match partial shape observations. However, the ability to match general observations was limited by the size of the database as the patches could not be interpolated. Ricao et al. [42] used both PCA and a learned autoencoder to map SDF subvolumes to lower-dimensional representations, approaching local shape priors from the perspective of compression. With this approach the SDF must be computed by fusion first, which serves as an information bottleneck limiting the ability to develop priors over fine-grained structures. In another work, Xu et al. [56] developed an object-level learned shape representation using a network that maps from images to SDFs . This representation is conditioned on and therefore not independent of the observed image. Williams et al. [54] showed recently that a deep network can be used to fit a representation of a surface by training and evaluating on the same point cloud, using a local chart for each point which is then combined to form a surface atlas. Their results are on complete point clouds in which the task is simply to densify and denoise, whereas we also show that our priors can locally complete surfaces that were not observed. Other work on object-level shape representation has explored representations in which shapes are composed of smaller parts. Structured implicit functions used anisotropic Gaussian kernels to compose global implicit shape representations [20]. Similarly, CvxNets compose shapes using a collection of convex subshapes [15]. Like ours, both of these methods show the promise of compositional shape modelling, but surface detail was limited by the models used. Last, concurrent work of Genova et al. [19] combines a set of irregularly positioned implicit functions to improve details in full object reconstruction.

3   Review of DeepSDF

We will briefly review DeepSDF [40]. Let $f_\theta(\mathbf{x}, \mathbf{z})$ be a signed surface distance function modeled as a fully-connected neural network with trainable parameters
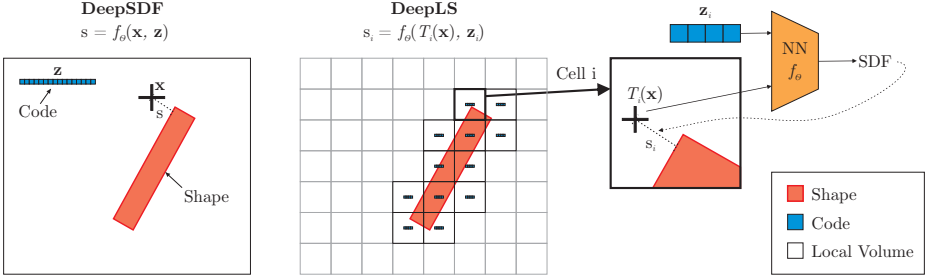
Fig. 2: 2D example of DeepSDF [40] and DeepLS (ours). DeepSDF provides global shape codes (left). We use the DeepSDF idea for local shape codes (center). Our approach requires a matrix of low-dimensional code vectors which in total require less storage than the global version. The gray codes are an indicator for empty space. The SDF to the surface is predicted using a fully-connected network that receives the local code and coordinates as input.

θ and shape code $\mathbf{z}$. Then a shape $\mathcal{S}$ is defined as the zero level set of $f_\theta(\mathbf{x}, \mathbf{z})$:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 \mid f_\theta(\mathbf{x}, \mathbf{z}) = 0\}. \tag{1}$$

In order to simultaneously train for a variety of shapes, a $\mathbf{z}$ is optimized for each shape while network parameters θ are shared for the whole set of shapes.

4   Deep Local Shapes

The key idea of DeepLS is to compose complex general shapes and scenes from a collection of simpler local shapes as depicted in Fig. 2. Scenes and shapes of arbitrary complexity cannot be described with a compact fixed length shape code such as used by DeepSDF. Instead it is more efficient and flexible to encode the space of smaller local shapes and to compose the global shape from an adaptable amount of local codes.

To describe a surface $\mathcal{S}$ in $\mathbb{R}^3$ using DeepLS, we first define a partition of the space into local volumes $V_i \subseteq \mathbb{R}^3$ with associated local coordinate systems. Like in DeepSDF, but at a local level, we describe the surface in each local volume using a code $\mathbf{z}_i$. With the transformation $T_i(\mathbf{x})$ of the global location $\mathbf{x}$ into the local coordinate system, the global surface $\mathcal{S}$ is described as the zero level set

$$\mathcal{S} = \left\{\mathbf{x} \in \mathbb{R}^3 \mid \bigoplus_i w(\mathbf{x}, V_i) f_\theta\left(T_i(\mathbf{x}), \mathbf{z}_i\right) = 0\right\}, \tag{2}$$

where $w(\mathbf{x}, V_i)$ weighs the contribution of the ith local shape to the global shape $\mathcal{S}$, $\bigoplus$ combines the contributions of local shapes, and f_θ is a shared autodecoder network for local shapes with trainable parameters θ. Various ways of designing the combination operation and weighting function can be explored. From voxel-based tesselations of the space to more RBF-like point-based sampling to – in the limit – collapsing the volume of a local code into a point and thus making $\mathbf{z}_i$ a continuous function of the global space.
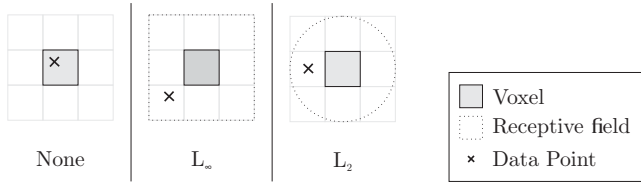
Fig. 3: Square (L_∞ norm) and spherical (L_2 norm) for the extended receptive fields for training local codes.

Here we focus on exploring the straight forward way of defining local shape codes over a sparsely allocated voxels V_i of the 3D space as illustrated in Fig. 2. We define $T_i(\mathbf{x}) := \mathbf{x} - \mathbf{x}_i$, transforming a global point $\mathbf{x}$ into the local coordinate system of voxel V_i by subtracting its center $\mathbf{x}_i$. The weighting function becomes the indicator function over the volume of voxel V_i. Thus, DeepLS describes the global surface $\mathcal{S}$ as:

$$\mathcal{S} = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid \textstyle\sum_i \mathbb{1}_{\mathbf{x} \in V_i} f_\theta \left(T_i(\mathbf{x}), \mathbf{z}_i \right) = 0 \right\} . \tag{3}$$

4.1 Shape Border Consistency

We found that with the proposed division of space (i.e. disjoint voxels for local shapes) leads to inconsistent surface estimates at the voxel boundaries. One possible solution is to choose w as partition of unity [39] basis functions with local support to combine the decoded SDF values. We experimented with trilinear interpolation as an instance of this. However, this method increases the number of required decoder evaluations to query an SDF value by a factor of eight.

Instead, we keep the indicator function and train decoder weights and codes such that a local shape is correct beyond the bounds of one voxel, by using training pairs from neighboring voxels. Then, the SDF values on the voxel boundaries are accurately computable from any of the abutting local shapes. We experimented with spheres (i.e. L_2 norm) and voxels (i.e. L_∞ norm) (c.f. Fig. 3) for the definition range of extended local shapes and found that using an L_∞ norm with a radius of 1.5 times the voxel side-length provides a good trade-off between accuracy (fighting border artifacts) and efficiency (c.f. Sec. 5).

4.2 Deep Local Shapes Training and Inference

Given a set of SDF pairs $\{(\mathbf{x}_j, s_j)\}_{j=1}^{N}$, sampled from a set of training shapes, we aim to optimize both the parameters θ of the shared shape decoder $f_\theta(\cdot)$ and all local shape codes $\{\mathbf{z}_i\}$ during training and only the codes during inference.

Let $\mathcal{X}_i = \{\mathbf{x}_j \mid L(T_i(\mathbf{x}_j)) < r\}$ denote the set of all training samples $\mathbf{x}_j$, falling within a radius r of voxel i with local code $\mathbf{z}_i$ under the distance metric L. We train DeepLS by minimizing the negative log posterior over the training

data $\mathcal{X}_i$:

$$\arg\min_{\theta, \{\mathbf{z}_i\}} \sum_i \sum_{\mathbf{x}_j \in \mathcal{X}_i} ||f_\theta (T_i(\mathbf{x}_j), \mathbf{z}_i) - s_j||_1 + \frac{1}{\sigma^2}||\mathbf{z}_i||_2^2.$$

In order to encode a new scene or shape into a set of local codes, we fix decoder weights θ and find the maximum a-posteriori codes $\mathbf{z}_i$ as

$$\arg\min_{\mathbf{z}_i} \sum_{\mathbf{x}_j \in \mathcal{X}_i} ||f_\theta (T_i(\mathbf{x}_j), \mathbf{z}_i) - s_j||_1 + \frac{1}{\sigma^2}||\mathbf{z}_i||_2^2, \qquad (4)$$

given partial observation samples $\{(\mathbf{x}_j, s_j)\}_{j=1}^M$ with $\mathcal{X}_i$ defined as above.

4.3 Point Sampling

For sampling data pairs $(\mathbf{x}_j, s_j)$, we distinguish between sampling from meshes and depth observations. For meshes, the method proposed by Park et al. [40] is used. For depth observations, we estimate normals from the depth map and sample points in 3D that are displaced slightly along the normal direction, where the SDF value is assumed to be the magnitude of displacement. In addition to those samples, we obtain free space samples along the observation rays. The process is described formally in the supplemental materials.

5 Experiments

The experiment section is structured as follows. First, we compare DeepLS against recent deep learning methods (e.g. DeepSDF, AtlasNet) in Sec. 5.1. Then, we present results for scene reconstruction and compare them against related approaches on both synthetic and real scenes in Sec. 5.2.

Experiment setup. The models used in the following experiments were trained on a set of local shape patches, obtained from 200 primitive shapes (e.g. cuboids and ellipsoids) and a total of 1000 shapes from the 3D Warehouse [1] dataset (200 each for the airplane, table, chair, lamp, and sofa classes). Our decoder is a four layer MLP, mapping from latent codes of size 128 to the SDF value. We present examples from the training set, several additional results, comparisons and further details about the experimental setup in the supplemental materials.

5.1 Object Reconstruction

3D Warehouse [1] We quantitatively evaluate surface reconstruction accuracy of DeepLS and other shape learning methods on various classes from the 3D Warehouse dataset. Quantitative results for the chamfer distance error are shown in Table 1. As can be seen DeepLS improves over related approaches by approximately one order of magnitude. It should be noted that this is not a comparison between equal methods since the other methods infer a global,
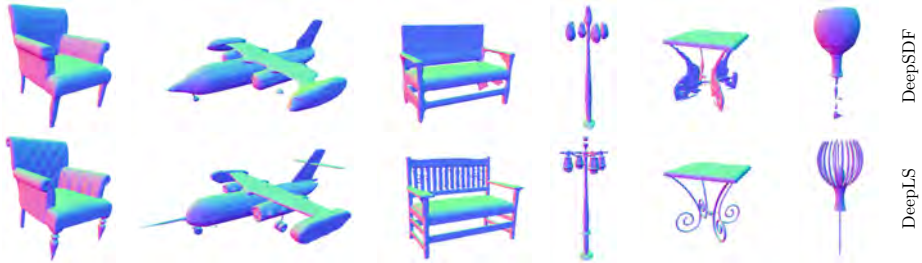
Fig. 4: Qualitative comparison of DeepLS with DeepSDF on some shapes from the 3D Warehouse [1] dataset.

Method	Shaoe Category					Decoder Params	Represent. Params
	chair	plane	table	lamp	sofa		
AtlasNet-Sph. [22]	0.752	0.188	0.725	2.381	0.445	3.6 M	1.0 K
AtlasNet-25 [22]	0.368	0.216	0.328	1.182	0.411	43.5 M	1.0 K
DeepSDF [40]	0.204	0.143	0.553	0.832	0.132	1.8 M	**0.3 K**
DeepLS	**0.030**	**0.018**	**0.032**	**0.078**	**0.044**	**0.05 M**	312 K

Table 1: Comparison for reconstructing shapes from the 3D Warehouse [1] test set, using the Chamfer distance. Results with additional metrics are similar as detailed in the supplemental materials. Note that due to the much smaller decoder, DeepLS is also orders of magnitudes faster in decoding (querying SDF values).

object-level representation that comes with other advantages. Also, the parameter distribution varies significantly (c.f. Tab. 1). Nonetheless, it proves that local shapes lead to superior reconstruction quality and that implicit functions modeled by a deep neural network are capable of representing fine details. Qualitatively, DeepLS encodes and reconstructs much finer surface details as can be seen in Fig. 4.

Efficiency Evaluation on Stanford Bunny [2] Further, we show the superior inference efficiency of DeepLS with a simple experiment, illustrated in Figure 5. A DeepLS model was trained on a dataset composed only of randomly oriented primitive shapes. It is used to infer local codes that pose an implicit representation of the Stanford Bunny. Training and inference together took just one minute on a single GPU. The result is an RMSE of only 0.03% relative to the length of the diagonal of the minimal ground truth bounding box, highlighting the ability of DeepLS to generalize to novel shapes. For comparison, we also trained a DeepSDF model to represent only the Stanford Bunny (jointly training latent code and decoder). In order to achieve the same surface error, this model required over 8 days of GPU time, showing that the high compression rates and object-level completion capabilities of DeepSDF and related techniques comes at the cost of long training and inference times. This is likely caused at least in
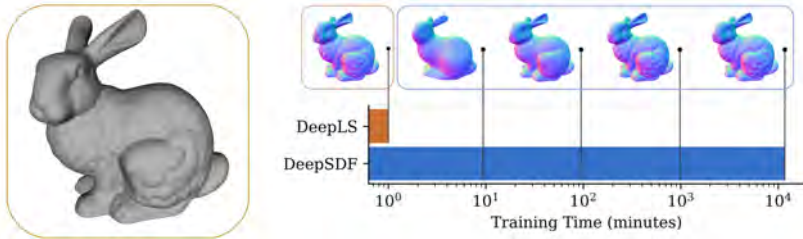
Fig. 5: A comparison of the efficiency of DeepLS and DeepSDF. With DeepLS, a model trained for one minute is capable of reconstructing the Stanford Bunny [2] in full detail. We then trained a DeepSDF model to represent the same signed distance function corresponding to the Stanford Bunny until it reaches the same accuracy. This took over 8 days of GPU time (note the log scale of the plot).

Method	mean	kt0	kt1	kt2	kt3
TSDF Fusion	5.42 mm	5.35 mm	5.88 mm	5.17 mm	5.27 mm
DeepLS	**4.92 mm**	**5.15 mm**	**5.48 mm**	**4.32 mm**	**4.71 mm**

Table 2: Surface reconstruction accuracy of DeepLS and TSDF Fusion [12] on the synthetic ICL-NUIM dataset [23] benchmark

part by gradient computation amongst all training samples, which we avoid by subdividing physical space and optimizing local representations in parallel.

5.2   Scene Reconstruction

We evaluate the ability of DeepLS to reconstruct at scene scale using synthetic (in order to provide quantitative comparisons) and real depth scans. For synthetic scans, we use the ICL-NUIM RGBD benchmark dataset [23]. The evaluation on real scans is done using the 3D Scene Dataset [59]. For quantitative evaluation, the asymmetric Chamfer distance metric provided by the benchmark [23] is used.

Synthetic ICL-NUIM Dataset Evaluation We provide quantitative measurements of surface reconstruction quality on all four ICL-NUIM sequences in Table 2, where each system has been tuned for lowest surface error. We also show results qualitatively in Fig. 6 and show additional results, e.g. on data with artificial noise, in the supplemental materials. Most surface reconstruction techniques involve a tradeoff between surface accuracy and completeness. For TSDF fusion systems such as KinectFusion [38], this tradeoff is driven by choosing a truncation distance and the minimum confidence at which surfaces are extracted by marching cubes. With DeepLS, we only extract surfaces up to some fixed distance from the nearest observed depth point, and this threshold is what trades off accuracy and completion of our system. For a full and fair comparison, we derived a pareto-optimal curve by varying these parameters for the two methods on the 'kt0' sequence of the ICL-NUIM benchmark and plot the results in Figure 7. We measure completion by computing the fraction of ground truth points for which there is a reconstructed point within 7 mm. Generally,
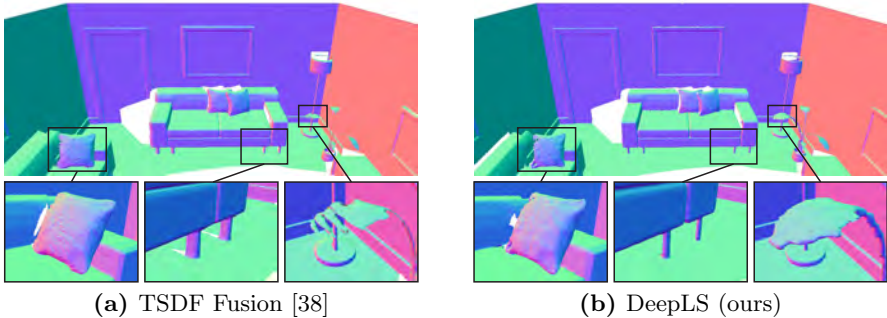
(a) TSDF Fusion [38]   **(b)** DeepLS (ours)

Fig. 6: Qualitative results of TSDF Fusion [38] (left) and DeepLS (right) for scene reconstruction on a synthetic ICL-NUIM [23] (CC BY 3.0, Handa, A., Whelan, T., McDonald, J., Davison) scene. The highlighted areas indicate the ability of DeepLS to handle oblique viewing angles, partial observation, and thin structures.
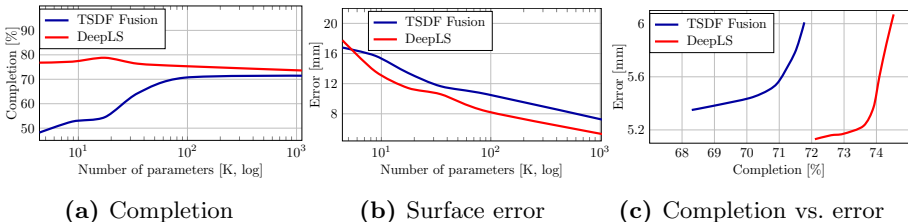

(a) Completion   **(b)** Surface error   **(c)** Completion vs. error

Fig. 7: Comparison of completion (a) and surface error (b) as a function of representation parameters on a synthetic scene from the ICL-NUIM [23] dataset. In contrast to TSDF Fusion, DeepLS maintains reconstruction completeness almost independent of compression rate. On the reconstructed surfaces (which is 50% less for TSDF Fusion) the surface error decreases for both methods (c.f. Fig. 8). Plot (c) shows the trend of surface error vs. mesh completion. DeepLS consistently shows higher completion at the same surface error. It scores less error than TSDF Fusion in all but the highest compression setting but it produces nearly two times more complete reconstruction than TSDF Fusion at this compression rate.

DeepLS can reconstruct more complete surfaces at the same level of accuracy as SDF Fusion.

The number of representation parameters used by DeepLS is theoretically independent of the rendering resolution and only depends on the resolution of the local shapes. In contrast, traditional volumetric scene reconstruction methods such as TSDF Fusion have a tight coupling between number of parameters and the desired rendering resolution. We investigate the relationship between representation size per unit volume of DeepLS and TSDF Fusion by evaluating the surface error and completeness as a function of the number of parameters. As a starting point we choose a representation that uses 8^3 parameters per $5.6 \text{cm} \times 5.6 \text{cm} \times 5.6 \text{cm}$ volume (7 mm voxel resolution). To increase compression we increase the voxel size for TSDF Fusion and the local shape code volume size for DeepLS. We provide the quantitative and qualitative analysis of the scene reconstruction results with varying representation size in Fig. 7 (a and b)
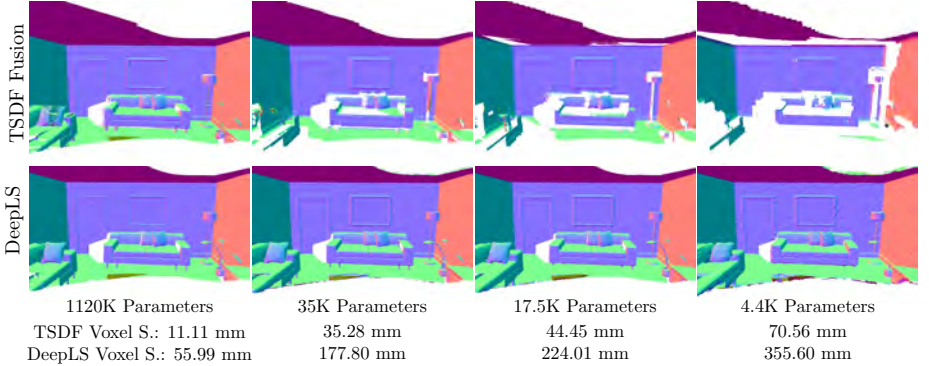
1120K Parameters | 35K Parameters | 17.5K Parameters | 4.4K Parameters
TSDF Voxel S.: 11.11 mm | 35.28 mm | 44.45 mm | 70.56 mm
DeepLS Voxel S.: 55.99 mm | 177.80 mm | 224.01 mm | 355.60 mm

Fig. 8: Qualitative analysis of representation size with DeepLS and TSDF Fusion [12] on a synthetic scene in the ICL-NUIM [23] dataset. DeepLS is able to retain details at higher compression rates (lower number of parameters). It achieves these compression rates by using bigger local shape voxels, leading to a stronger influence of the priors.

Method	Burghers		Lounge		CopyRoom		StoneWall		TotemPole	
	Error	Comp	Error	Comp	Error	Comp	Error	Comp	Error	Comp
TSDF F. [12]	10.11	85.46	11.71	85.17	12.35	83.99	14.23	91.02	13.03	83.73
DeepLS	**5.74**	**95.78**	**7.38**	**96.00**	**10.09**	**99.70**	**6.45**	**91.37**	**8.97**	**87.23**

Table 3: Quantitative evaluation of DeepLS with TSDF Fusion on 3D Scene Dataset [59]. The error is measured in mm and *Comp* (completion) corresponds to the percentage of ground truth surfaces that have reconstructed surfaces within 7 mm. Results suggest that DeepLS produces more accurate and complete 3D reconstruction in comparison to volumetric fusion methods on real depth acquisition datasets.

and Fig. 8 respectively. The plots in Fig. 7 show conclusively that TSDF Fusion drops to about 50% less complete reconstructions while DeepLS maintains completeness even at the highest compression rate, using only 4.4K parameters for the full scene. Quantitatively, TSDF Fusion also achieves low surface error for high compression. However, this can be contributed to the used ICL-NUIM benchmark metric, which does not strongly punish missing surfaces.

Evaluation on Real Scans We evaluate DeepLS on the 3D Scene Dataset [59], which contains several scenes captured by commodity structured light sensors, and a challenging scan of thin objects. In order to also provide quantitative errors we assume the reconstruction performed by volumetric fusion [12] of all depth frames to be the *ground truth*. We then apply DeepLS and TSDF fusion on a small subset of depth frames, taking every 10th frame in the capture sequence. The quantitative results of this comparison are detailed in Table 3 for various scenes. It is shown that DeepLS produces both more accurate and more complete 3D reconstructions. Furthermore, we provide qualitative examples of this experiment in Fig. 9 for the outdoor scene "Burghers of Calais" and for the indoor scene "Lounge". Notice, that DeepLS preserves more details on the faces of the statues in "Burghers of Calais" scene and reconstructs thin details such

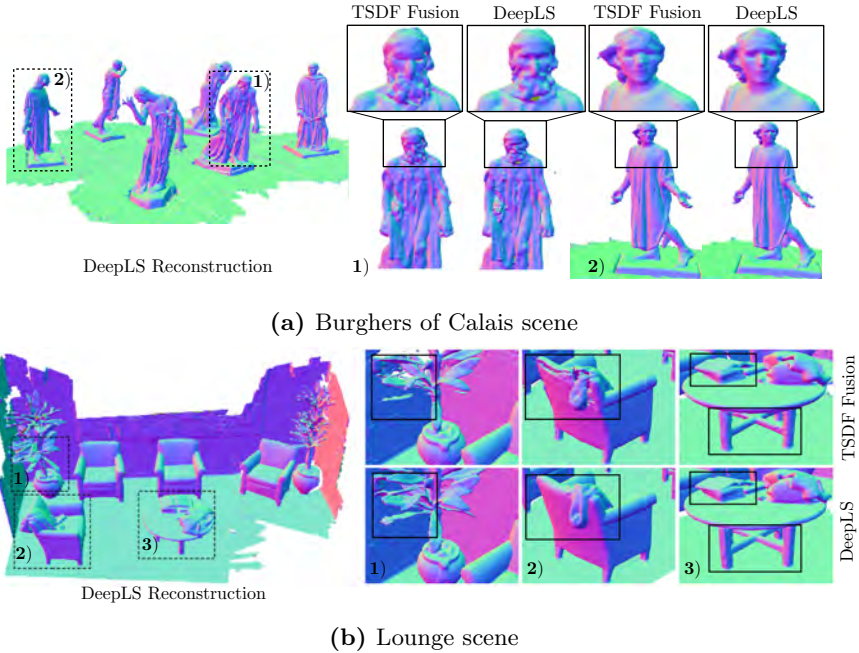(a) Burghers of Calais scene

(b) Lounge scene

Fig. 9: Qualitative results for DeepLS and TSDF Fusion [12] on two scenes of the 3D Scene Dataset [59]. Best viewed with zoom in the digital version of the paper.

as leaves of the plants in "Lounge" scene.

Further, we specifically analyse the strength of DeepLS in representing and completing thin local geometry. We collected a scan from an object consisting of two thin circles kept on a stool with long but thin cylindrical legs (see Fig. 10). The 3D points were generated by a structured light sensor [52,9,47]. It was scanned from limited directions leading to very sparse set of points on the stool's surface and legs. We compared our results on this dataset to several 3D methods including TSDF Fusion [38], Multi-level Partition of Unity (MPU) [39], Smooth Signed Distance Function [7], Poisson Surface Reconstruction [29], PFS [51] and TSR [4]. We found that due to lack of points and thin surfaces most of the methods failed to either represent details or complete the model. MPU [39], which fits quadratic functions in local grids and is very related to our work, fails in this experiment (see Fig. 10b). This indicates that our learned shape priors are more robust than fixed parameterized functions. Methods such as PSR [29], SSD [7] and PFS [51] fit global implicit function to represent shapes. These methods made the thin shapes thicker than they should be. Moreover, they also had issues completely reconstructing the thin rings on top of the stool. TSR [4] was able to fit to the available points but is unable to complete structures such as bottom surface of the stool and it's cylindrical legs, where no observations exist. This shows how our method utilizes local shape priors to complete partially
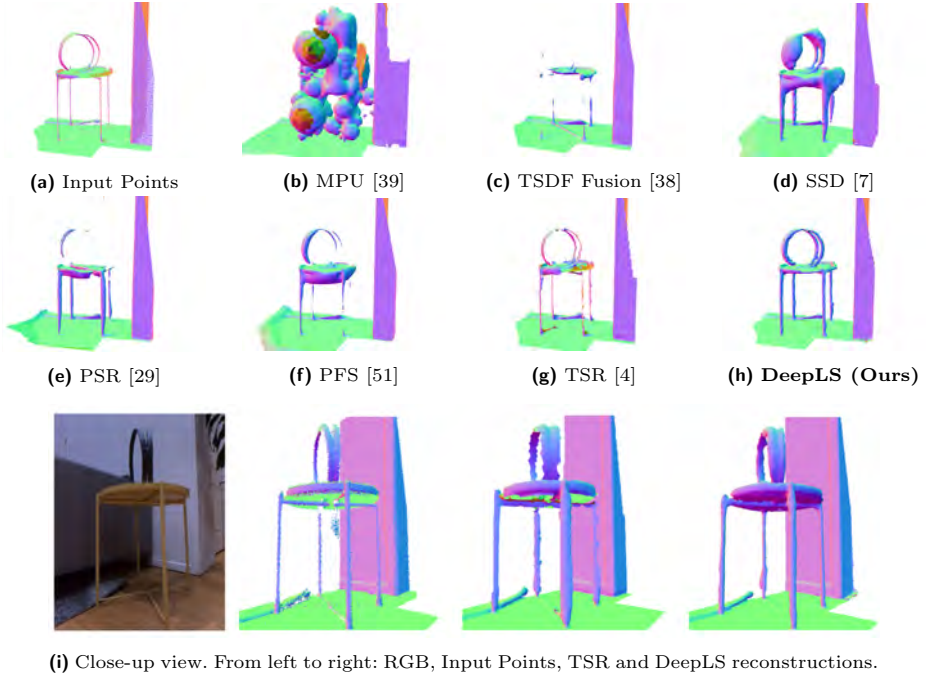
(a) Input Points   **(b)** MPU [39]   **(c)** TSDF Fusion [38]   **(d)** SSD [7]

(e) PSR [29]   **(f)** PFS [51]   **(g)** TSR [4]   **(h) DeepLS (Ours)**

(i) Close-up view. From left to right: RGB, Input Points, TSR and DeepLS reconstructions.

Fig. 10: Qualitative comparison of DeepLS against other 3D Reconstruction techniques on a very challenging thin and incomplete scan. Most of the methods fail to build thin surfaces in this dataset. TSR fits to the thin parts but is unable to complete structures such as the bottom and cylindrical legs of the stool. In contrast, DeepLS reconstructs thin structures and also completes them.

scanned shapes. Please refer to the supplemental materials for further qualitative comparisons and results.

6   Conclusion

In this work we presented DeepLS, a method to combine the benefits of volumetric fusion and deep shape priors for 3D surface reconstruction from depth observations. A key to the success of this approach is the decomposition of large surfaces into local shapes. This decomposition allowed us to reconstruct surfaces with higher accuracy and finer detail than traditional SDF fusion techniques, while simultaneously completing unobserved surfaces, all using less memory than storing the full SDF volume would require. Compared to recent object-centric shape learning approaches, our local shape decomposition leads to greater efficiency for both training and inference while improving surface reconstruction accuracy by an order of magnitude.

References

1. 3D Warehouse. `https://3dwarehouse.sketchup.com/`

2. Stanford 3D Scanning Repository. `http://graphics.stanford.edu/data/3Dscanrep/`

3. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. arXiv preprint arXiv:1707.02392 (2017)

4. Aroudj, S., Seemann, P., Langguth, F., Guthe, S., Goesele, M.: Visibility-consistent thin surface reconstruction using multi-scale kernels. ACM Transactions on Graphics (TOG) **36**(6), 187 (2017)

5. Ben-Hamu, H., Maron, H., Kezurer, I., Avineri, G., Lipman, Y.: Multi-chart generative surface modeling. In: SIGGRAPH Asia 2018 Technical Papers. p. 215. ACM (2018)

6. Blinn, J.F.: A generalization of algebraic surface drawing. ACM transactions on graphics (TOG) **1**(3), 235–256 (1982)

7. Calakli, F., Taubin, G.: Ssd: Smooth signed distance surface reconstruction. In: Computer Graphics Forum. vol. 30, pp. 1993–2002. Wiley Online Library (2011)

8. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3d objects with radial basis functions. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 67–76. ACM (2001)

9. Chabra, R., Straub, J., Sweeney, C., Newcombe, R., Fuchs, H.: Stereodrnet: Dilated residual stereonet. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11786–11795 (2019)

10. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

11. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)

12. Curless, B., Levoy, M.: A volumetric method for building complex models from range images (1996)

13. Dai, A., Nießner, M.: Scan2mesh: From unstructured range scans to 3d meshes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5574–5583 (2019)

14. Dai, A., Ruizhongtai Qi, C., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5868–5877 (2017)

15. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., Tagliasacchi, A.: Cvxnets: Learnable convex decomposition (2019)

16. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence **40**(3), 611–625 (2017)

17. Fuhrmann, S., Goesele, M.: Floating scale surface reconstruction. ACM Transactions on Graphics (ToG) **33**(4), 46 (2014)

18. Gal, R., Shamir, A., Hassner, T., Pauly, M., Cohen-Or, D.: Surface reconstruction using local shape priors. In: Symposium on Geometry Processing. pp. 253–262. No. CONF (2007)

19. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Deep structured implicit functions. arXiv preprint arXiv:1912.06126 (2019)

20. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. arXiv preprint arXiv:1904.06447 (2019)

21. Gkioxari, G., Malik, J., Johnson, J.: Mesh r-cnn. arXiv preprint arXiv:1906.02739 (2019)

22. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: Atlasnet: A papier-m\^ ach\'e approach to learning 3d surface generation. arXiv preprint arXiv:1802.05384 (2018)
23. Handa, A., Whelan, T., McDonald, J., Davison, A.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: IEEE Intl. Conf. on Robotics and Automation, ICRA. Hong Kong, China (May 2014)
24. Häne, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: 2017 International Conference on 3D Vision (3DV). pp. 412–420. IEEE (2017)
25. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on Computer graphics and interactive techniques. pp. 71–78 (1992)
26. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural networks **2**(5), 359–366 (1989)
27. Jancosek, M., Pajdla, T.: Multi-view reconstruction preserving weakly-supported surfaces. In: CVPR 2011. pp. 3121–3128. IEEE (2011)
28. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. vol. 7 (2006)
29. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG) **32**(3), 1–13 (2013)
30. Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., Kolb, A.: Real-time 3d reconstruction in dynamic scenes using point-based fusion. In: 2013 International Conference on 3D Vision-3DV 2013. pp. 1–8. IEEE (2013)
31. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. pp. 1–10. IEEE Computer Society (2007)
32. Labatut, P., Pons, J.P., Keriven, R.: Robust and efficient surface reconstruction from range data. In: Computer graphics forum. vol. 28, pp. 2275–2290. Wiley Online Library (2009)
33. Liao, Y., Donne, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2916–2925 (2018)
34. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)
35. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.: Deep level sets: Implicit surface representations for 3d shape inference. arXiv preprint arXiv:1901.06802 (2019)
36. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
37. Newcombe, R.A., Davison, A.J.: Live dense reconstruction with a single moving camera. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1498–1505. IEEE (2010)
38. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. pp. 127–136. IEEE (2011)
39. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicits. In: Acm Siggraph 2005 Courses, pp. 173–es (2005)

40. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. arXiv preprint arXiv:1901.05103 (2019)
41. Pfister, H., Zwicker, M., Van Baar, J., Gross, M.: Surfels: Surface elements as rendering primitives. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 335–342. ACM Press/Addison-Wesley Publishing Co. (2000)
42. Ricao Canelhas, D., Schaffernicht, E., Stoyanov, T., Lilienthal, A., Davison, A.: Compressed voxel-based mapping using unsupervised learning. Robotics **6**(3), 15 (2017)
43. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3577–3586 (2017)
44. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. arXiv preprint arXiv:1905.05172 (2019)
45. 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)
46. Sinha, A., Bai, J., Ramani, K.: Deep learning 3d shape surfaces using geometry images. In: European Conference on Computer Vision. pp. 223–240. Springer (2016)
47. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
48. Stühmer, J., Gumhold, S., Cremers, D.: Real-time dense geometry from a handheld camera. In: Joint Pattern Recognition Symposium. pp. 11–20. Springer (2010)
49. Stutz, D., Geiger, A.: Learning 3d shape completion from laser scan data with weak supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1955–1964 (2018)
50. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2088–2096 (2017)
51. Ummenhofer, B., Brox, T.: Global, dense multiscale reconstruction for a billion points. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1341–1349 (2015)
52. Whelan, T., Goesele, M., Lovegrove, S.J., Straub, J., Green, S., Szeliski, R., Butterfield, S., Verma, S., Newcombe, R.: Reconstructing scenes with mirror and glass surfaces. ACM Transactions on Graphics (TOG) **37**(4), 102 (2018)
53. Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., Davison, A.: Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems
54. Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., Panozzo, D.: Deep geometric prior for surface reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10130–10139 (2019)
55. 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. pp. 1912–1920 (2015)

56. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. arXiv preprint arXiv:1905.10711 (2019)
57. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Interpretable unsupervised learning on 3d point clouds. arXiv preprint arXiv:1712.07262 (2017)
58. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV). pp. 728–737. IEEE (2018)
59. Zhou, Q.Y., Koltun, V.: Dense scene reconstruction with points of interest. ACM Transactions on Graphics (ToG) **32**(4), 1–8 (2013)

Supplementary Material

A   Overview

The supplementary material consists of detailed information about the experimental setup in Sec. B, a detailed view on local shapes in Sec. C, additional metrics for the 3D Warehouse dataset comparison in Sec. E, and more results, comparisons and details about experiments in Sec. F. Additionally, we provide a video alongside this document, showing reconstructions in motion.
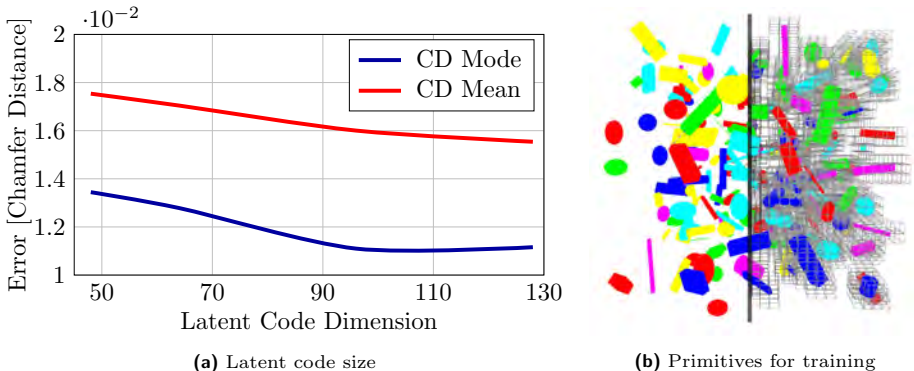

(a) Latent code size   **(b)** Primitives for training

Fig. 11: Fig. (a) shows the effect of changing the latent code dimensions on the Chamfer distance test error on airplanes class of 3D Warehouse [1]. Fig. (b) shows an example for a scene containing 200 primitives shapes as used for training the local shape priors. On the right side, the instantiated local shape blocks are shown.

B   Experimental Setup

Autodecoder Network The DeepLS autodecoder network is a lighter version of the network proposed for DeepSDF [40]. It consists of four fully-connected layers, separated by leaky ReLUs and a tanh at the end, producing values in $[-1, 1]$ that are then scaled by the chosen SDF truncation value. Each layer has 128 output neurons. Fig. 11a shows the result of a small study to find the best latent code size in a trade-off between accuracy and compression. We chose a latent size of 125, leaving us with 128 input neurons for the first network layer.

Training The output of the network is trained to produce truncated SDF values. To this end, tanh is also applied on the appropriately scaled ground truth SDF before computing the loss against the network output. We chose the scale so that the interval $[-0.9, 0.9]$ after tanh covers approximately two blocks. We
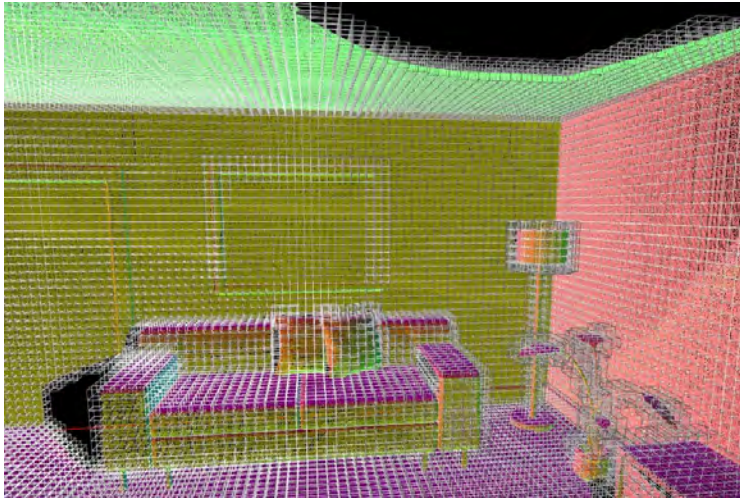
Fig. 12: Instantiated local shape blocks in a scene. The blocks are allocated sparsely, based on available depth data, which makes the approach scale well to real world inputs.

Fig. 13: Interpolation in latent space of a local shape code between a flat surface and a pole.

optimize codes and network parameters using the Adam optimizer with initial learning rate of 0.01, which we decrease twice over the course of training.

Training Data The training data to learn local shape priors consists of three different categories of shapes. The first category contains simple primitive shapes, as shown in Fig. 11b, with random 6-DOF pose in space. The second category consists of 3D Warehouse [1] training meshes: We sampled a subset of 200 models from each training set of the classes *airplane*, *chair*, *lamp*, *sofa*, and *table*. Each model was split into $32 \times 32 \times 32$ local shape blocks. The last category consists of models from the Stanford 3D scanning repository [2], namely *bunny* and *dragon*.

C   Local Shape Space

In order to give a better intuition about the space of learned local priors, inter-polation sequences between local surfaces are provided in Fig. 13. It should be noted that, in general, the space of possible functions in a voxel is much larger.

Therefore, training local priors heavily restricts the space of solutions to those producing local SDF functions that describe reasonable surfaces. The behavior of local shapes over the course of optimization is shown in the accompanying video. Additionally, Fig. 12 show all allocated blocks in a scene, which together reconstruct the whole surface.

D   Shape border consistency

In order to better understand the border consistency among the borders of local shapes we used simple 2D scenes often composed of simple primitive shapes such as triangles, rectangle and circles. In training and testing session we sample points around these shapes and extract SDF measurements as described in DeepSDF [40]. Note, we color code these sample points with red for positive, blue for negative and green for zero SDF measurements. In all the 2D experiments we use roughly 1000 samples inside a grid cell (local shape spatial size) in training session and 100 samples in test session. We report testing error as the SDF prediction error in 2d (pixels).

In the following experiment, in order to study shape border consistency we increased the receptive field of local shapes as shown in Figure 14a. By receptive field we mean the physical space of input samples for a particular local shapes. In general, we observe improvement in SDF prediction on the boundaries of local shapes with increasing receptive field as shown in Figure 14b. Although, we observe a critical point in the receptive field after which the performance drops as shown in Figure 14c. As increasing receptive field makes the local shapes bigger and more complicated so more parameters in the network F_θ are required to express the space of shapes. Hence, each F_θ has a critical point in the receptive field. We also observe the early convergence in optimization for optimal receptive field as shown in Figure 14d.

## E   3D Warehouse Comparison - Additional Metric

In Tab 4 we extend the comparison on shapes from 3D Warehouse dataset on other metrics. In addition to the Chamfer distance we show mesh accuracy, which is defined as the maximum distance d such that 90% of generated points are within d if the ground truth mesh. All metrics show the similar trend that DeepLS achieves way higher accuracy than the related object-level representations.

F   Scene Experiments

Here, we explain the process from depth maps to SDF samples for real scenes in more detail and provide qualitative results. See also the provided video for further results.
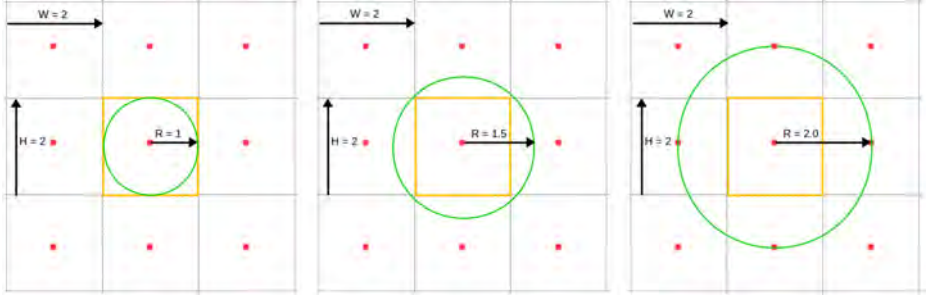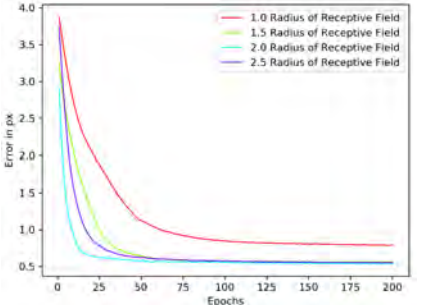
(a) This figure demonstrates the receptive field of the reference local shape inside yellow block with area inside green circle. R represents the radius of receptive field.
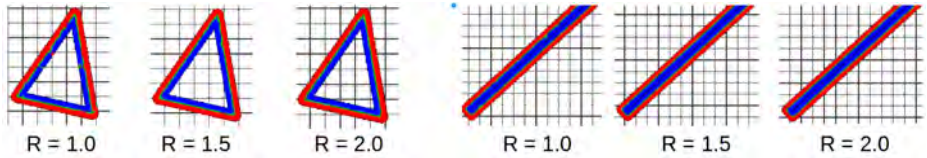
(b) This figure demonstrates the qualitative difference in the SDF prediction with varying receptive fields.
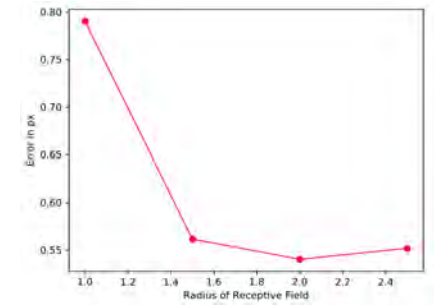
(c) Error in SDF prediction with increasing receptive field. Critical point in receptive field is observed.

(d) Plot shows the test error with increasing iterations in optimization. Optimal receptive field shows early convergence.

Fig. 14: This figure demonstrates the effect of receptive field on the quality of reconstruction of SDF.

CD, mean	chair	plane	table	lamp	sofa
AtlasNet-Sph.	0.752	0.188	0.725	2.381	0.445
AtlasNet-25	0.368	0.216	0.328	1.182	0.411
DeepSDF	0.204	0.143	0.553	0.832	0.132
DeepLS	**0.030**	**0.018**	**0.032**	**0.078**	**0.044**
CD, median					
AtlasNet-Sph.	0.511	0.079	0.389	2.180	0.330
AtlasNet-25	0.276	0.065	0.195	0.993	0.311
DeepSDF	0.072	0.036	0.068	0.219	0.088
DeepLS	**0.023**	**0.011**	**0.026**	**0.019**	**0.039**
Mesh acc., mean					
AtlasNet-Sph.	0.0330	0.0130	0.0320	0.0540	0.0170
AtlasNet-25	0.0180	0.0130	0.0140	0.0420	0.0170
DeepSDF	0.0090	0.0040	0.0120	0.0130	0.0040
DeepLS	**0.0009**	**0.0008**	**0.001**	**0.0012**	**0.0011**

Table 4: Representing unknown shapes from the 3D Warehouse [1] test set. In addition to the Chamfer distance, we provide mesh accuracy [45]. Lower is better for all metrics. It can be seen that all metrics show a similar trend.

F.1   Sample Generation

Sample generation from depth scans consists of the following steps: (1) For a given scene, we generate a collection of 3D points from depth maps. (2) For each depth point, we create one sample with zero SDF, and several positive and negative SDF samples by moving the sample along the pre-computed surface normal by 1.5 cm and -1.5 cm, respectively. The accompanying SDF value is chosen as the moved distance.

(3) We generate additional free space samples along the observation rays. Further, we weight each set of points inversely based on the depth of the initial scan point, to ensure that accurate points closer to the scanning device are weighted higher. This procedure is described in detail in TSDF Fuison [12]. Similar to traditional SDF fusion approaches [38], DeepLS exposes a parameter which controls the size of the region around actual depth samples in which marching cubes is performed. Varying this parameter leads to the mesh accuracy vs. completion trade-off, discussed in the main paper.

F.2   Comparisons for Synthetic Noise

Fig. 15 shows results of DeepLS and TSDF Fusion on an ICL-NUIM benchmark scene with artificial noise of $\sigma = 0.015$. The learned local shape priors of DeepLS effectively are able to find plausible surfaces given the noisy observations, which results in smoother surfaces in comparison to TSDF Fusion.
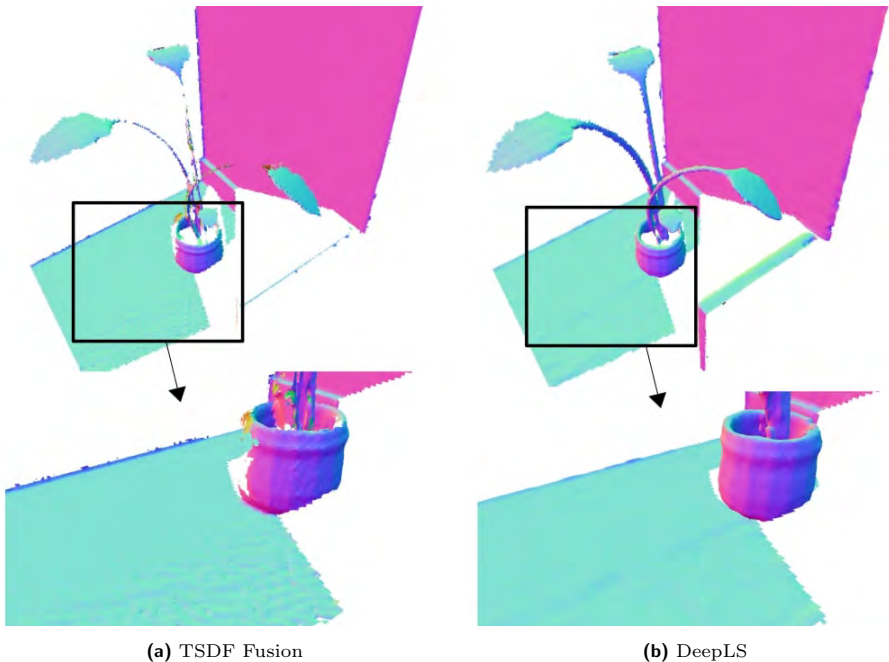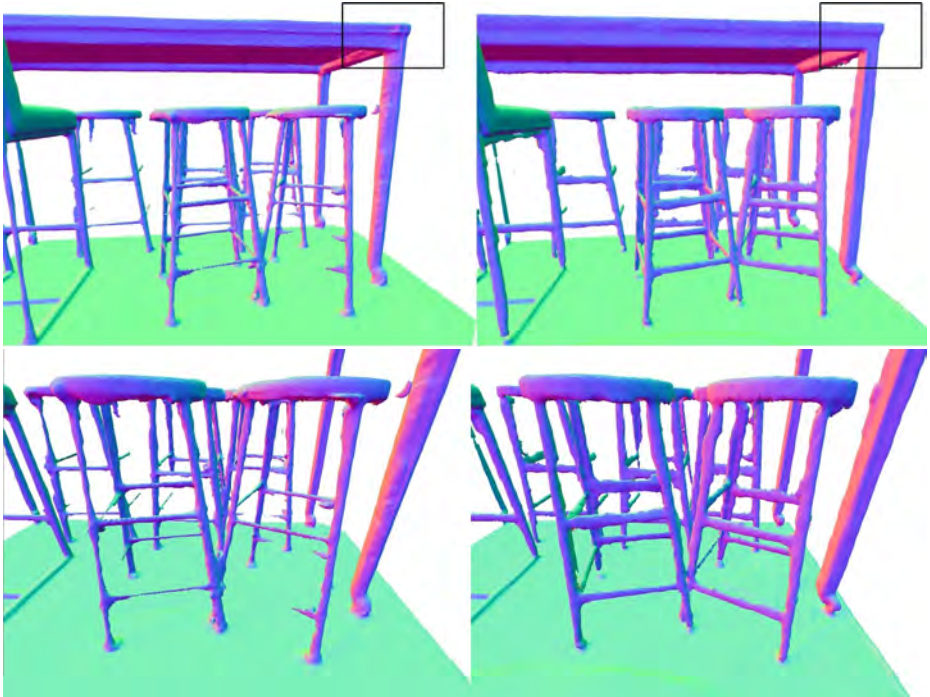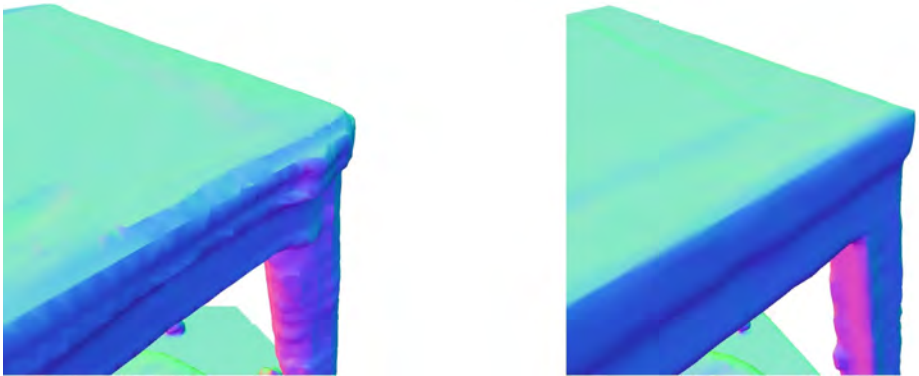
(a) TSDF Fusion   (b) DeepLS

Fig. 15: The figure shows a part of the ICL-NUIM kt0 scene [23], reconstructed from samples with artitificial noise of $\sigma = 0.015$. DeepLS shows better denoising properties than TSDF Fusion. For the whole ICL-NUIM benchmark scene, DeepLS achieves a surface error of **6.41** mm with **71.04** % completion while TSDF Fusion has an error of 7.29 mm and 68.53 % completion.

F.3   Qualitative Results

We show additional qualitative results on real scanned data in Fig. 16, Fig. 17 and in the supplemented video. Both scenes showed in the figures were captured using a handheld structured light sensor system as was used for capturing the Replica dataset [47] and in related work [52,9]. An in-house SLAM system, similar to state-of-the-art systems [16,36], was used to provide 6 degree of freedom (DoF) poses for individual depth frames from the sensor. It can be seen that DeepLS succeeds in representing small details like the bars of chairs while TSDF Fusion tends to loose these details. Also, we observe sharper corners (c.f. 16b) and more complete surfaces (c.f. 17b) with DeepLS.

(a) DeepLS (right) captures thin chair legs better than TSDF Fusion (left) which tends to loose those details.

(b) Zoomed view of region marked with black box in (a). DeepLS (right) represents sharper corners and smoother planes than TSDF Fusion (left).

Fig. 16: Qualitative comparison of TSDF Fusion (left) with DeepLS (right) on real scanned data prepared using a structured light sensor system [47]. The figure (b) is the magnified region marked with black box in figure (a).
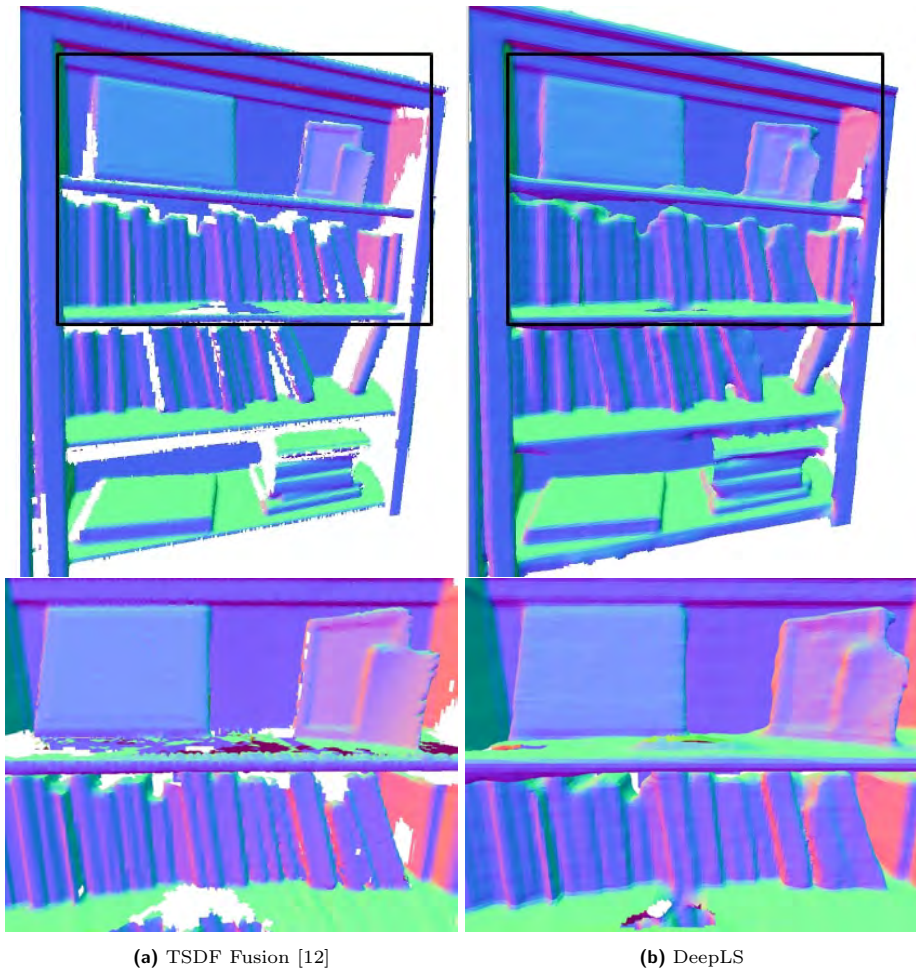
(a) TSDF Fusion [12]   (b) DeepLS

Fig. 17: We show the scene reconstruction quality of DeepLS vs TSDF Fusion [12] on a partially scanned real scene dataset using a structured light sensor system [47]. This figure shows that DeepLS provides better local shape completion than TSDF Fusion. The bottom row represents the zoomed in view marked with black box in the top row.