Check for updates

DOI: 10.1111/cgf.15000

COMPUTER GRAPHICS *forum*
Volume 0 (2024), number 0, e15000

PPSurf: Combining Patches and Point Convolutions for Detailed Surface Reconstruction

Philipp Erler,[1] Lizeth Fuentes-Perez,[2] Pedro Hermosilla,[1] Paul Guerrero,[3] Renato Pajarola[2] and Michael Wimmer[1]

[1]TU Wien, Vienna, Austria
perler@cg.tuwien.ac.at, phermosilla@cvl.tuwien.ac.at, wimmer@cg.tuwien.ac.at
[2]University of Zürich, Zurich, Switzerland
{fuentes, pajarola}@ifi.uzh.ch
[3]Adobe Research, London, UK
guerrero@adobe.com

Abstract

3D surface reconstruction from point clouds is a key step in areas such as content creation, archaeology, digital cultural heritage and engineering. Current approaches either try to optimize a non-data-driven surface representation to fit the points, or learn a data-driven prior over the distribution of commonly occurring surfaces and how they correlate with potentially noisy point clouds. Data-driven methods enable robust handling of noise and typically either focus on a global *or a* local *prior, which trade-off between robustness to noise on the global end and surface detail preservation on the local end. We propose PPSurf as a method that combines a global prior based on point convolutions and a local prior based on processing local point cloud patches. We show that this approach is robust to noise while recovering surface details more accurately than the current state-of-the-art. Our source code, pre-trained model and dataset are available at https://github.com/cg-tuwien/ppsurf.*

Keywords: modeling, surface reconstruction

CCS Concepts: • Computing methodologies → Shape modelling; Machine learning

1. Introduction

3D surface reconstruction from point clouds is a key step for workflows in areas such as content creation, archaeology, digital cultural heritage and engineering, to convert raw 3D point scan data, like casual RGBD (colour and depth) mobile phone images or more accurate range scans (*e.g.* from laser range scanner), to surface-based 3D object representations that can be used in downstream applications (see Figure 1).

Given the large practical interest, surface reconstruction has become a central problem in computer graphics and vision research. The problem is generally ill-defined, as different surfaces may correspond to similar point clouds. However, several approaches have been proposed to tackle this ambiguity. One research direction attempts to optimize surface representations with strong *non-data-driven* inductive biases to fit the point cloud [KBH06, WSS*19, PJL*21, BZYSM21]. This resolves the ambiguity, but is susceptible to deteriorating conditions of the input points, such as scan noise or regions with missing points, which cannot easily be corrected using a fixed inductive bias. Another line of research focuses on learning *data-driven* priors, usually over the distribution of commonly occurring surfaces and how they correlate with potentially noisy point clouds [PFS*19, EGO*20, PJL*21, BM22]. The surface reconstruction ambiguity can then be resolved by finding a surface that has a high probability for the given point cloud under the learned prior. The prior in these data-driven methods can range from *global*, where the prior captures a distribution over full 3D object surfaces, to *local*, where the prior captures the distribution over local surface patches. Global priors are the least susceptible to noise and missing points, but have limited capability to capture fine local details. Local priors, on the other hand, can capture such fine details accurately, but are more susceptible to strong noise and missing points. Existing methods mostly focus their prior on a small range in this global–local spectrum. For example, DeepSDF [PFS*19] uses a global prior, Points2Surf [EGO*20] mostly focuses on a local prior, while POCO's point convolutions [BM22] learn a prior in the medium range that is reasonably robust to deteriorating conditions, but still struggles to accurately capture local detail.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

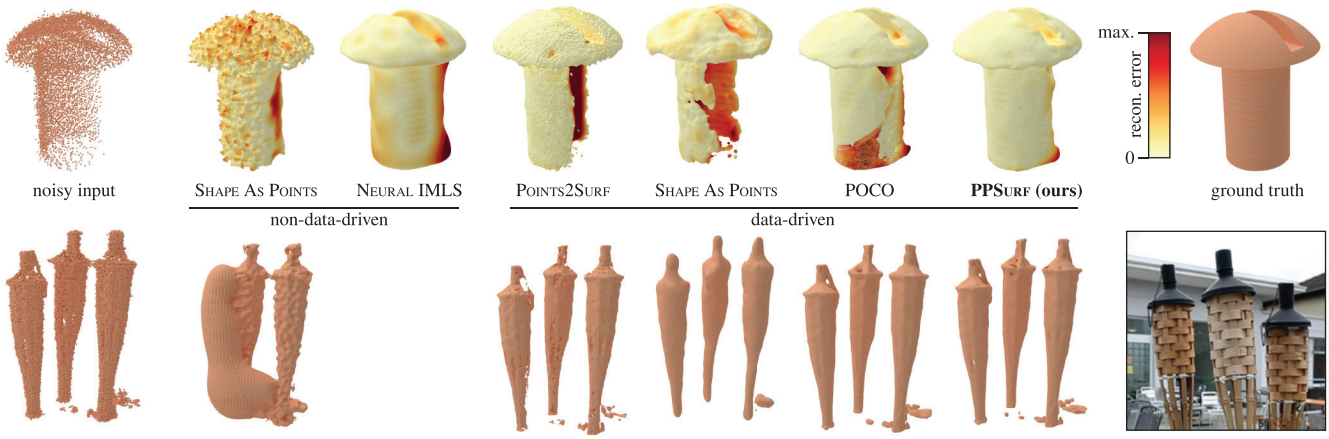

Figure 1: *We present PPSurf, a method to reconstruct surfaces from noisy point clouds. Unlike previous methods, our approach combines two strong data-driven priors, one prior over local surface details, and a second prior over the coarse shape of larger surface regions. This makes PPSurf robust to noise, while reconstructing surface detail better than current methods.*

We propose PPSurf as a method that covers a wider range in the global–local spectrum of priors, by combining the local prior of a patch-based method like Points2Surf with a more global prior of a point convolution-based method like POCO. For this purpose, we design an architecture that has two branches: the first branch is based on POCO [BM22] and provides a global prior by applying several layers of point convolutions to a sparse set of support points. To reconstruct geometric details more accurately, we merge features from this first branch with features from a second branch, which processes a local patch of points with PointNet [QSMG16]. We additionally discovered that modifying the architecture of PointNet by replacing the sum aggregation with an attention-based aggregation improves performance. This results in a method that is robust to noise and missing points, while preserving details more accurately than previous methods.

In our experiments, we compare PPSurf to several previous state-of-the-art methods, both data-driven and non-data-driven, on synthetic as well as real-world data, and demonstrate improved performance on both in-distribution, and out-of-distribution surface reconstruction tasks.

2. Related Work

Surface reconstruction from point clouds is an active area of research. We distinguish between *data-driven* methods that train on a large dataset, and *non-data-driven* methods that do not use machine learning or overfit to a single shape.

Non-data-driven methods. Poisson reconstruction [KBH06, KH13] has for many years been the gold standard of non-data-driven approaches. Recent works have suggested optimizing the parameters of a neural network to predict the signed distance to the surface [AL20, SMB*20, AL21] directly from a *single* point cloud. In particular, Atzmon and Lipman [AL20] introduced this concept for un-oriented point clouds. They optimized the parameters of the neural network with a sign-agnostic loss and a geometric initialization of its parameters. Gropp *et al.* [GYH*20] and Atzmon and Lipman [AL21] followed up on this work and included a gradient regularization in the loss. Later, Ma *et al.* [BZYSM21] introduced Neural-Pull, an optimization objective that uses directly the gradient of the optimized SDF to move the query points to the closest point in the input point cloud. In follow-up work, this approach was extended by incorporating a network to classify a point being on the surface or not [CHL23], and an additional loss that aligns the gradient direction between different level sets of the SDF [MZLH23]. In order to improve the quality of the final SDF, Yifan *et al.* [YWOSH20] and Zhou *et al.* [ZML*22] proposed to iteratively increase the input point cloud with points sampled from the optimized SDF in the previous iteration. A different approach was proposed by Peng *et al.* [PJL*21] (also used in LION [ZVW*22]), based on a differentiable Poisson surface reconstruction operation that could be used for optimization-based or learned reconstructions. Differently from previous methods, the set of points in the surface is optimized through the differentiable reconstruction instead of a neural network representing the SDF. Lin *et al.* proposed a parametric Gauss formula for reconstruction [LXSW22], which has quadratic complexity in memory leading to prohibitive costs for larger point clouds. VIPSS by Huang *et al.* [HCJ19] formulates reconstruction as a constrained quadratic optimization problem. iPSR by Hou *et al.* [HWW*22] uses an iterative approach to Poisson reconstruction that improves the surface more and more, while removing the need to be given point normals. IsoPoisson by Xiao *et al.* [XSL*23] incorporate an isovalue constraint to the Poisson equation, which helps with consistent normal orientation and consequently improved reconstruction.

Non-data-driven methods are sensitive to noise, which is usually present in real 3D scans. In order to address this limitation to some extent, a recent pre-print from Wang *et al.* [WWW*22] proposed Neural-IMLS, a non-data-driven method that regularizes the smoothness of surface normals using an MLP with limited capacity. While this produces smooth surfaces, it also loses some geometric detail due to this non-data-driven regularization. Noise-to-noise mapping by Baorui *et al.* [MLH23] focuses on the reconstruction of noisy point clouds in an unsupervised overfitting scheme.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Additionally, these methods require significant reconstruction times due to the optimization being performed for each shape individually, which can be a limiting factor for large scans.

Data-driven methods. A recent line of research has approached the problem of shape reconstruction in a data-driven manner by using a large dataset to learn a prior over the distribution of commonly occurring surfaces and how they correlate with the input point cloud. These approaches are typically fast and robust to noisy inputs compared to non-data-driven approaches. However, in such methods, the resulting reconstruction highly depends on the quality of such priors.

Several works have proposed to use a *global* prior to capturing the distribution over full 3D object surfaces [CZ19, MON*19, PFS*19]. These methods define such a prior as a single latent vector representing the shape, which is then used as a condition in a fully connected network to decode the SDF of a given query point. Usually, the decoder is trained on large data sets with a point-cloud encoder [MON*19, CZ19]. However, Park *et al.* [PFS*19] proposed to train the decoder directly on such data sets and then optimize the latent vector to match the noisy point cloud during inference. Recently, Zhang *et al.* [ZTNW23] proposed to use richer global priors. They introduced an encoder–decoder network that encodes the input point cloud using attention modules into a set of latent vectors representing the shape, which are then used to predict the SDF for a set of query points using cross-attention modules.

Other works have opted to condition their models with *local* priors. Siddiqui *et al.* [STM*21] encoded the input point clouds in a set of latent scene patches. These latent vectors are used to query a database of latent vectors from patches obtained from the training set. The obtained patches are then blended together using an attention mechanism. Ma *et al.* [BYSZ22] incorporated *local* priors by including a network pre-trained on a large number of surface patches which classifies a point as being on the surface or not. This network is used to guide an optimization process that learns the shape's SDF using another neural network. Jiang *et al.* [JSM*20] pre-trained an SDF encoder–decoder on a large data set of object parts. Then, during the optimization process, only the latent codes of the different parts of the object are optimized. Chen *et al.* [CTFZ22] propose a dual contouring method learned on a small local prior.

Since *global* and *local* priors provide complementary information about the shape, a common approach is to use a prior in the *medium* range using a hierarchical encoder–decoder network. These approaches reduce the input point cloud to a simplified representation, *e.g.* voxelization or sub-sampled point cloud, which is then enriched by the global information provided by the bottleneck of the encoder–decoder architecture. Chibane *et al.* [CAPM20] and Peng *et al.* [PNM*20] proposed a 3DCNN encoder–decoder network to encode the sparse or noisy point cloud to later predict the SDF for an arbitrary point around the surface. Chibane *et al.* [CMPM20] extended this work to predict an unsigned distance field, which allowed them to represent complex open surfaces. Tang *et al.* [TLX*21] extended the work of Peng *et al.* [PNM*20] to include test-time optimization to improve out-of-distribution point clouds. Ummenhofer and Koltun [UK21] proposed a CNN that works directly on an Octree, from which the model was able to predict the SDF. Wang *et al.* [WLT22] also represented the input point cloud with an octree, from which they constructed a graph. This graph was further processed by a GCN encoder–decoder to generate an embedding for each octree node, from where the final SDF is predicted. Dai *et al.* [DDN20] instead used a 3D sparse encoder–decoder network to complete partial 3D scans and predict a complete SDF. Lionar *et al.* [LESP21] also developed an encoder–decoder network but used instead the projection of the input point cloud to a set of arbitrary 2D planes, from which the final SDF was predicted. Boulch and Marlet [BM22] recently proposed to use an encoder–decoder network that directly worked with points, avoiding discretization artifacts from voxel-based representations. Although all these methods work relatively well when compared with methods that use *global* or *local* priors alone, they struggle to accurately capture fine local details of the shapes.

Erler *et al.* [EGO*20] proposed to explicitly model *global* and *local* priors directly from point clouds using two different branches. Each branch used a PointNet [QSMG16] architecture, to process the local patch around the query point in the local branch, and a point cloud representing the complete shape in the global branch. While the local branch was able to capture high-frequency details relatively well, they used a weak global prior due to the small subset of points selected to represent the shape. Our approach addresses the limitations of all these methods by incorporating strong *global* and *local* priors.

3. Method

The goal of our method is to take as input an unoriented point cloud $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ that was sampled from an unknown watertight surface $\mathcal{S}^{\text{gt}}$ with a noisy sampling process, and output a surface $\mathcal{S}$ that approximates $\mathcal{S}^{\text{gt}}$ as closely as possible. Similar to several previous approaches, we define the surface $\mathcal{S}$ using an implicit representation, since this guarantees watertightness and naturally handles arbitrary surface topology in a smooth and differentiable way. More specifically, $\mathcal{S}$ is defined as the 0.5-level set of an occupancy field $o(\mathbf{x})$: $\mathcal{S} \coloneqq \{\mathbf{x} \| \; o(\mathbf{x}) = 0.5\}$.

We train a network $f_\theta(\mathbf{x}, P)$ with parameters θ to model the field o given a point cloud P:

$$o(\mathbf{x}) \coloneqq f_\theta(\mathbf{x}, P) \tag{1}$$

The network f uses two branches: (i) a *global* branch $f^g(\mathbf{x}, P')$ that performs point convolutions [BPM20] on a sparse random subset of points $P' \subseteq P$ and effectively learns a global prior over the coarse shape of $\mathcal{S}$ given the input points P, and (ii) a *local* branch $f^l(\mathbf{x}, P_\mathbf{x})$ that processes a small local patch $P_\mathbf{x} \subset P$ around $\mathbf{x}$ and effectively learns a local prior over the detailed shape of local surface patches. Each branch outputs a feature vector for a given query point $\mathbf{x}$ that is combined into a single feature vector before being processed by a small MLP f^o that outputs the occupancy probability $o(\mathbf{x})$:

$$f_\theta(\mathbf{x}, P) \coloneqq f^o\big(f^g(\mathbf{x}, P') \oplus f^l(\mathbf{x}, P_\mathbf{x})\big), \tag{2}$$

where $\oplus$ is the operation used to combine the two feature vectors, a sum in our experiments. Here, we omit the parameters of the networks f^o, f^g and f^l to avoid a cluttered notation. Figure 2 illustrates our architecture.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

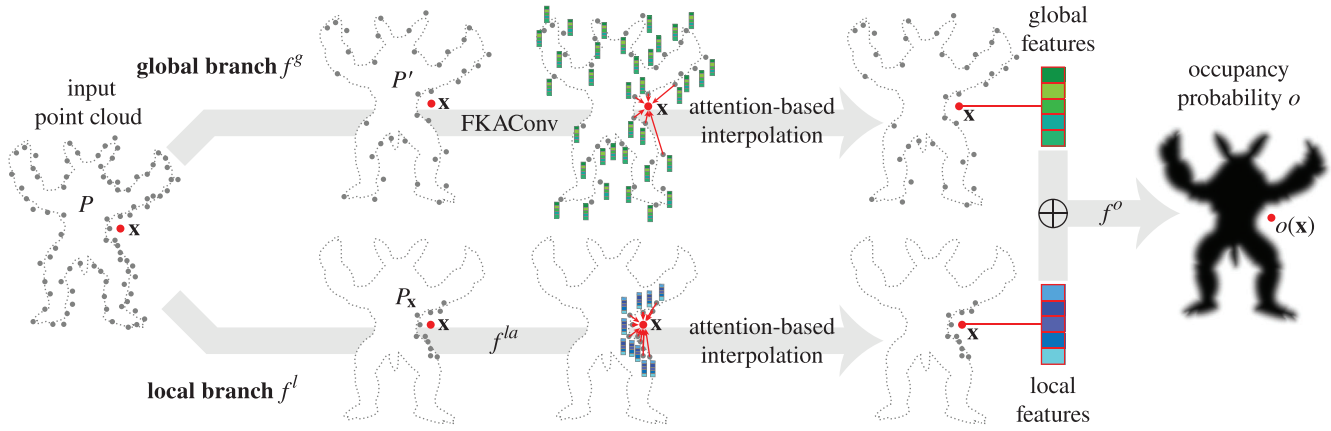

Figure 2: *PPSurf computes the occupancy probability at a query point* $\mathbf{x}$ *given a noisy point cloud* P*. A* global *branch processes a sparse subset* $P' \subseteq P$ *using point convolutions, followed by an attention-based interpolation to get features at* $\mathbf{x}$ *that capture the coarse shape of the point cloud. A* local *branch processes a local patch* $P_\mathbf{x} \subset P$ *using a PointNet [QSMG16] with attention-based aggregation to get features at* $\mathbf{x}$ *that capture the detailed shape of the point cloud near* $\mathbf{x}$*. Global and local features are aggregated to compute the occupancy probability at* $\mathbf{x}$*.*

In the following, we describe the architecture of PPSurf, including the global and local branches in Section 3.1, followed by a description of the training and inference setups in Sections 3.2 and 3.3, respectively.

3.1. Architecture

Global branch. The global branch $f^g(\mathbf{x}, P')$ takes as input a random subset $P' \subseteq P$ and a 3D query point $\mathbf{x}$ and outputs a *global* feature vector for the point $\mathbf{x}$, which encodes information about the coarse shape of the point cloud. We implement the global branch using POCO [BM22], which consists of two main components: (i) a point convolution module that computes a feature vector $\mathbf{z}'_i$ for each sparse point $\mathbf{p}'_i \in P'$, followed by (ii) an interpolation module that interpolates the feature vectors $\mathbf{z}'_i$ to get the global feature vector at point $\mathbf{x}$.

The point convolution module uses FKAConv [BPM20] to process the sparse point cloud P' into a feature vector for each point:

$$Z' = \text{FKAConv}(P'), \tag{3}$$

where $Z' = \{\mathbf{z}'_1, \mathbf{z}'_2, \dots, \mathbf{z}'_{|P'|}\}$ is the set of feature vectors at each sparse point. Due to limitations both in performance and network capacity, convolutions can only be performed on the sparse subset P' instead of the full point cloud P, with $|P'| = 10\text{k}$ in our experiments. This module consists of 10 layers of convolutions. Each layer uses a convolution kernel that operates over the 16 nearest neighbours of each point.

Given a query point $\mathbf{x}$, the interpolation module interpolates the feature vectors $\mathbf{z}'_i$ at the nearest neighbours $\mathcal{N}'_\mathbf{x}$ of the query point to get the global feature vector using an attention-based weighting:

$$f^g(\mathbf{x}, P') := f^{gb}\left(\sum_{j \in \mathcal{N}'_\mathbf{x}} w_{\mathbf{x},j}\, f^{ga}\big((\mathbf{x} - \mathbf{p}'_j) \,\|\, \mathbf{z}'_j\big)\right) \tag{4}$$

$$\text{with } w_{\mathbf{x},j} := \frac{1}{k}\sum_{k=1}^{64} \text{softmax}_j\, f_k^{gw}\big((\mathbf{x} - \mathbf{p}'_j) \,\|\, \mathbf{z}'_j\big), \tag{5}$$

where $\|$ denotes concatenation, f^{ga}, f^{gb} are two MLPs that transform the feature vectors before and after the weighted sum, and f_k^{gw} are learned weighting functions, each implemented as a single linear layer. Analogous to the attention heads in multi-head attention, multiple different weighting functions are used as a form of ensemble learning, 64 in our experiments. Note that when evaluating multiple query points $\mathbf{x}$ for a point cloud, the point convolution module only needs to be evaluated once, while the interpolation module needs to be evaluated once per query point.

Local branch. The local branch $f^l(\mathbf{x}, P_\mathbf{x})$ processes a local patch $P_\mathbf{x}$ around the query point $\mathbf{x}$ and outputs a *local* feature vector for the point $\mathbf{x}$, which encodes information about the detailed shape of the point cloud near $\mathbf{x}$. We base the local branch on the popular PointNet [QSMG16] architecture, which has been successfully applied in various methods that process local point cloud patches [GKOM18, RLBG*19]. We modify the architecture with an attention-based aggregation, instead of the original max- or sum-based aggregation, which we found to improve performance.

We define the local patch $P_\mathbf{x}$ as the 50 nearest neighbours of the query point $\mathbf{x}$. We normalize the patch by centering it at the origin and scaling it to fit into a unit sphere, obtaining the normalized patch $\bar{P}_\mathbf{x}$. Subsequently, we apply PointNet with attention-based aggregation similar to Equations (4) and (5), but without using multiple attention heads:

$$f^l(\mathbf{x}, P_\mathbf{x}) := f^{lb}\left(\sum_{\bar{\mathbf{p}}_j \in \bar{P}_\mathbf{x}} v_j\, f^{la}(\bar{\mathbf{p}}_j)\right) \tag{6}$$

$$\text{with } v_j := \text{softmax}_j\, f^{lv}\big(f^{la}(\bar{\mathbf{p}}_j)\big), \tag{7}$$

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where f^{lv} is a learned weighting function implemented as linear layer, and f^{la}, f^{lb} are two MLPs that transform the feature vectors before and after the weighted aggregation.

3.2. Training setup

We train our network with a binary cross-entropy loss $\mathrm{BCE}(o(\mathbf{x}), o^{\mathrm{gt}}(\mathbf{x}))$ supervised by the ground-truth occupancy $o^{\mathrm{gt}}(\mathbf{x})$ on query points defined by the Points2Surf ABC var-noise training set [EGO*20]. We train with AdamW (lr = 0.001, betas = (0.9, 0.999), eps = 1e-5, weight_decay = 1e-2, amsgrad = False) for 150 epochs with scheduler steps at 75 and 125 epochs. On our training machine, we can fully utilize all four NVIDIA A40 GPUs with distributed data-parallel training using a total batch size of 50 and 48 workers. The other hyperparameters are mostly based on POCO, namely $10k$ manifold points, a network decoder k of 64 and two output classes. One change is the increased latent size of 128, which was 32 in POCO. The additional hyperparameters for the local branch are a PointNet latent size of 256 and a patch size of 50. The training takes about 5 h.

3.3. Inference setup

We use the inference setup from POCO [BM22], which differs from the training setup in two main aspects: First, we perform test-time augmentation in our global branch to obtain more reliable results. Second, we sample query points in a grid and use a variant of marching cubes to reconstruct a mesh. We describe both in more detail below.

Test-time augmentation. The sparse sub-sample $P' \subseteq P$ used for the global branch may miss important geometric detail. To improve robustness, we compute the per-point feature vectors $\mathbf{z}'_i$ for multiple different random sub-samples $P'_1, P'_2, \ldots$, until each point in P is included in at least 10 sub-samples. The $\geq$10 different feature vectors for each point in P are then averaged before performing the interpolation step.

Mesh reconstruction. We place query points in a 257^3 grid and use a variant of marching cubes [LC87] proposed in POCO to obtain a mesh from the occupancy field $o(\mathbf{x})$. That marching cubes variant uses a region-growing strategy starting from the input points to avoid the costly evaluation at all grid points, and super-samples marching-cube edges that intersect a surface to get a more accurate estimate of the intersection point.

4. Results

We evaluate PPSurf by comparing our surface reconstruction performance to several state-of-the-art methods, both data-driven and non-data-driven. We show both quantitative and qualitative comparisons in Section 4.1. Additionally, we provide an ablation to empirically validate our main design choices in Section 4.2.

Metrics. We use three well-known metrics to evaluate the error of our reconstructed surfaces: the Chamfer distance, the F1-score and the normal error. We evaluate each metric at $100k$ random surface samples for the Chamfer distance and normal error, or volume samples for the IoU. This results in roughly $\pm 0.5\%$ variance between different runs.

The *Chamfer distance* [BTBW77, FSG17] measures the distance between two point sets. We use it to measure the distance between reconstructed and GT surface samples. It is defined as

$$\frac{1}{|A|}\sum_{\mathbf{p}_i \in A} \min_{\mathbf{p}_j \in B} \|\mathbf{p}_i - \mathbf{p}_j\|_2^2 + \frac{1}{|B|}\sum_{\mathbf{p}_j \in B} \min_{\mathbf{p}_i \in A} \|\mathbf{p}_j - \mathbf{p}_i\|_2^2, \tag{8}$$

where A and B are point sets of size $100k$ sampled on the surface of the GT object and the reconstructed object.

The *F1 Score* [TH15] measures the overlap between the ground truth surface and the region enclosed by the reconstructed surface, similar to the IoU. It weights precision and recall equally.

The *normal error* measures the difference between the normals of the reconstructed surface and the ground truth normals. We sample $100k$ points uniformly on the ground truth mesh A and the reconstructed mesh B, storing the normals of their originating faces. Then, we find the closest neighbour of each point $b \in B$ in A. We report the average angle between the normals of these point pairs: $\frac{1}{n_s}\sum_{i=1}^{n_s}(\arccos(\mathbf{n}_i^A \cdot \mathbf{n}_i^B))$, where $\mathbf{n}_i^A$ and $\mathbf{n}_i^B$ are ground truth and reconstructed normals, respectively.

Datasets. We evaluate our method on the set of dataset variants introduced in P2S [EGO*20]:

- The ABC variant of P2S [EGO*20] is a subset of the ABC dataset by Koch *et al.* [KMJ*19] and contains 4950 points clouds from high-quality CAD meshes in the training set and 100 point clouds in the test set.
- The Famous [EGO*20] dataset consists of 22 diverse well-known meshes, including the Stanford Bunny, the Utah Teapot and the Armadillo. We use this dataset for testing only.
- A subset of 100 shapes from the Thingi10k [ZJ16] dataset is used as additional test set. The Thingi10k dataset contains a variety of CAD shapes, but also more organic shapes like statues.
- The Real [EGO*20] dataset consists of three real-world point clouds.

All synthetic point clouds were created with the simulated scanner BlenSor [GKUP11] with a scanner resolution of 176×144, using a random number of scans between 5 and 30. Each dataset comes in up to five variants:

- *no noise*: A version without noise
- *med. noise*: A version with noise using a standard deviation of $0.01L$, where L is the largest side of the object's bounding box.
- *high noise*: A version with noise using a standard deviation of $0.05L$.
- *var. noise*: A version with variable noise, where the amount of noise used for a given shape is sampled uniformly in $[0, 0.05L]$ and the number of scans in [5, 30].
- *sparse*: A version with medium noise where all shapes only uses five scans, resulting in point clouds between $2k$ and $22k$ points.
- *dense*: A version with medium noise where all shapes use 30 scans, resulting in point clouds between $5k$ and $112k$ points.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

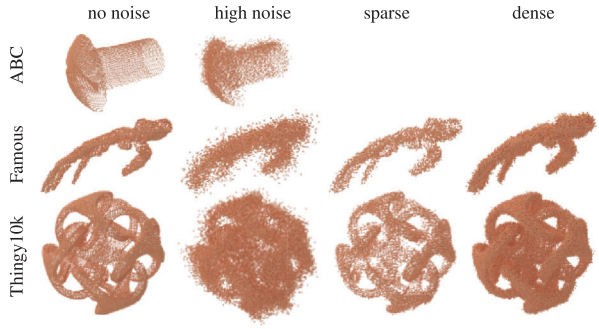

Figure 3: *Point cloud examples of the data sets used in our evaluation.*

For a fair comparison, we train all data-driven methods on the ABC var. noise dataset and evaluate them with each test set. Some point cloud examples of these data sets are illustrated in Figure 3.

4.1. Comparisons

We compare PPSurf to several recent data-driven and non-data-driven reconstruction methods. PGR [LXSW22], Neural-IMLS (IMLS) [WWW*22] and Shape as Points (SAP-O) are non-data-driven methods that do not train on a large dataset and instead directly fit a surface to the input point cloud. Shape as Points also has a data-driven variant (SAP) that uses a trained network. Additionally, we use Points2Surf (P2S) [EGO*20] and POCO [BM22] as data-driven methods. We took the best available variants and settings for each method: For PGR, we use the default parameters *wmin = 0.0015, alpha = 1.05* for no noise, med noise and var. noise. We use the following adapted parameters for the other datasets: *wmin = 0.03, alpha = 2.0* for high noise, *wmin = 0.03, alpha = 1.5* for dense and sparse. We use *thingi-noisy* for SAP-O, *vanilla* for P2S and *10k-FKAConv-InterpAttentionKHeadsNet* for POCO. We used the provided *noise-large* configuration for SAP. For IMLS, we used the results provided by the authors (high noise datasets were not provided by the authors). Note that IMLS was developed concurrently with our work.

Qualitative comparison. Figure 4 shows comparisons for one example of each dataset variant. While non-data-driven methods give competitive results on low-noise results, PPSurf has a clear advantage with sparse and noisy point clouds.

We show examples on real-world point clouds in Figure 5, where PPSurf produces clearer edges and finer details.

Quantitative comparison. Table 1 shows the performance of PPSurf on all dataset variants. We report the average over all shapes in the test set. Similar to the qualitative results, POCO, PPSurf and the non-data-driven methods share the first place in most low-noise dataset variants, but PPSurf 50NN takes the lead in almost all other dataset variants. This confirms that adding the local branch does indeed improve the local reconstruction.

Computation time and memory consumption. Training PPSurf on the ABC var-noise training set was done in 5 h on 4 NVIDIA A40 GPUs and 48 AMD EPYC-Milan cores. We reconstruct all shapes in our test sets on a single A40 and 48 CPU cores. See the timings and memory consumption in Table 2. While non-data-driven methods tend to be faster than data-driven ones, SAP is a lightning-fast exception. PPSurf with small patch sizes has a negligible impact on resources compared to POCO. Neural IMLS does not report timings. As it is concurrent work, we could not do our own measurements. While it is fast, PGR's memory usage varies a lot with point cloud size, between a few GB to going out-of-memory with >46GB on 21 shapes.

Discussion. For dense and noise-free point clouds, non-data-driven methods such as PGR, SAP-O and especially IMLS are a good option. However, their performance is limited in the presence of typical point-cloud artifacts, due to missing data-driven priors. Data-driven methods such as SAP, P2S, POCO and PPSurf can better deal with such artifacts. SAP is the fastest method but lacks accuracy, possibly due to its very small network. A bigger version could perhaps produce competitive results but would require non-trivial changes to the method.

P2S employs a relatively simple PointNet for global shape encoding, which results in a weak global prior that can not reach the quality of a more efficient encoder such as FKAConv. Furthermore, it reconstructs noisy surfaces, which is reflected in the relatively high normal error, even with noise-free inputs.

Apart from some noise-free datasets, only POCO is close to PPSurf's quality. PPSurf achieves similar results on low-noise point clouds, but significantly better reconstructions for noisy point clouds. When predicting the occupancy at the query points, POCO has no direct access to the full point cloud, only to a coarse latent representation. This inability to accurately represent local information is likely the reason why POCO tends to produce blobby structures and over-smooth the reconstructed surfaces. We avoid this by providing a latent code that captures local detail more accurately by adding a local branch that directly encodes dense local patches of the point cloud.

4.2. Ablation

We investigate several design choices in an ablation study on the ABC var-noise test set. Most importantly, Table 3 shows that having both global and local branches gains a major advantage. Referring to Table 4, the optimal local patch size lies in the range of 25NN to 100NN. Further, attention is a better symmetric operation than max, and concatenating features is similar to summing them. This can be seen in Table 5. Please see the supplementary for an evaluation of the most relevant variants on all datasets. We compare the following variants of our method:

- *Full* is the full method as described in Section 3.
- For *Only Local*, we set the global features to zeros, disabling this branch. Based on the results of this experiment, we conclude that this model can not reliably encode any surface since it lacks global knowledge of the surface to reconstruct.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Figure 4: *Qualitative comparison to all baselines. We evaluate one example from each dataset variant (except for the no-noise variants, where we only show one example due to space constraints). Colours show the distance of the reconstructed surface to the ground-truth surface. Due to our combined local and global branches, PPSurf reconstructs details more accurately than the baselines, especially in the presence of strong input noise. Note that results for Neural IMLS are not provided by the authors for the high-noise dataset variants. See the supplementary material for a qualitative comparison on all shapes in our test sets.*

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 1: *Comparison of reconstruction errors. We show the Chamfer distance, F1 Score and normal error between reconstructed and ground-truth surfaces averaged over all shapes in a dataset. Apart from a few noise-free datasets, PPSURF consistently performs similar or better than the baselines. Note that the mean performance of neural IMLS does not include results of the high noise datasets, which are likely to favour PPSURF. Due to out-of-memory errors, PGR could not reconstruct all shapes, which are ignored here. Best results per row are marked in bold and the second-best results are underlined*

	Chamfer distance (×100) ⇓							F1 ⇑							Normal Error ⇑						
Dataset	IMLS	PGR	SAP-O	SAP	P2S	POCO	PPSURF	IMLS	PGR	SAP-O	SAP	P2S	POCO	PPSURF	IMLS	PGR	SAP-O	SAP	P2S	POCO	PPSURF
ABC var. noise	1.08	1.60	1.18	1.18	0.84	0.70	**0.66**	0.78	0.50	0.67	0.79	0.83	0.89	**0.90**	0.55	1.29	1.11	0.52	0.65	0.32	**0.30**
ABC no noise	**0.48**	0.53	0.63	1.08	0.61	0.50	**0.48**	0.92	0.92	0.88	0.80	0.88	**0.94**	**0.94**	0.20	0.26	0.30	0.51	0.31	**0.19**	**0.19**
Famous no noise	**0.35**	0.36	**0.35**	0.99	0.46	0.39	0.37	0.95	0.95	0.95	0.84	0.93	0.95	**0.96**	**0.44**	0.48	**0.44**	0.75	0.57	0.46	0.46
Thingi10k no noise	0.40	**0.30**	0.43	0.89	0.39	0.33	0.33	0.93	0.95	0.92	0.85	0.93	0.95	**0.96**	0.25	0.19	0.23	0.49	0.32	**0.18**	0.19
Mean no noise	0.41	0.40	0.47	0.99	0.49	0.41	**0.39**	0.93	0.94	0.92	0.83	0.91	**0.95**	**0.95**	0.30	0.31	0.33	0.58	0.40	**0.28**	**0.28**
Famous med. noise	0.54	0.95	0.58	1.06	0.52	0.49	**0.48**	0.91	0.60	0.90	0.83	0.92	**0.94**	0.93	0.57	1.35	0.91	0.78	0.63	0.53	**0.54**
Thingi10k med. noise	0.58	0.93	0.56	0.93	0.44	0.39	**0.38**	0.90	0.57	0.89	0.85	0.92	**0.94**	**0.94**	0.37	1.32	0.78	0.50	0.38	**0.24**	0.25
Mean med. noise	0.56	0.94	0.57	0.99	0.48	0.44	**0.43**	0.91	0.59	0.89	0.84	0.92	**0.94**	**0.94**	0.47	1.33	0.85	0.64	0.50	**0.38**	0.39
ABC high noise	–	1.90	1.96	1.51	1.24	1.00	**0.97**	–	0.42	0.49	0.75	0.78	0.84	**0.85**	–	1.35	1.42	0.65	0.99	0.43	**0.41**
Famous high noise	–	1.86	1.80	1.62	1.14	1.11	**1.01**	–	0.50	0.59	0.78	0.84	0.84	**0.85**	–	1.35	1.39	0.91	1.04	0.76	**0.72**
Thingi10k high noise	–	1.94	1.89	1.45	1.08	0.92	**0.83**	–	0.51	0.60	0.80	0.84	0.87	**0.88**	–	1.31	1.36	0.64	0.90	0.47	**0.43**
Mean high noise	–	1.90	1.88	1.53	1.16	1.01	**0.94**	–	0.32	0.56	0.78	0.82	0.85	**0.86**	–	1.34	1.39	0.73	0.98	0.55	**0.52**
Famous sparse	0.90	0.88	0.71	1.24	0.77	0.67	**0.64**	0.86	0.88	0.88	0.74	0.89	**0.92**	**0.92**	0.68	0.75	0.86	0.89	0.71	0.60	**0.61**
Thingi10k sparse	0.82	0.89	0.86	1.35	0.78	**0.63**	**0.63**	0.85	0.86	0.84	0.73	0.87	**0.90**	**0.90**	0.48	0.53	0.76	0.73	0.51	**0.37**	0.39
Mean sparse	0.86	0.88	0.79	1.29	0.77	0.65	**0.63**	0.86	0.87	0.86	0.73	0.88	**0.91**	**0.91**	0.58	0.64	0.81	0.81	0.61	**0.48**	0.50
Famous dense	0.45	0.70	0.53	0.96	0.41	0.42	**0.40**	0.93	0.43	0.90	0.86	0.94	**0.95**	**0.95**	0.52	1.33	1.00	0.74	0.59	0.49	**0.48**
Thingi10k dense	0.49	0.67	0.54	0.88	0.36	0.35	**0.33**	0.91	0.47	0.89	0.87	0.94	0.95	**0.96**	0.30	1.23	0.84	0.47	0.33	**0.21**	**0.21**
Mean dense	0.47	0.69	0.53	0.92	0.39	0.39	**0.37**	0.92	0.45	0.89	0.86	0.94	0.95	**0.96**	0.41	1.28	0.92	0.60	0.46	0.35	**0.34**
Mean overall	0.61	1.04	0.93	1.16	0.70	0.61	**0.58**	0.89	0.66	0.80	0.81	0.89	0.91	**0.92**	0.43	0.98	0.88	0.66	0.61	**0.40**	**0.40**

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

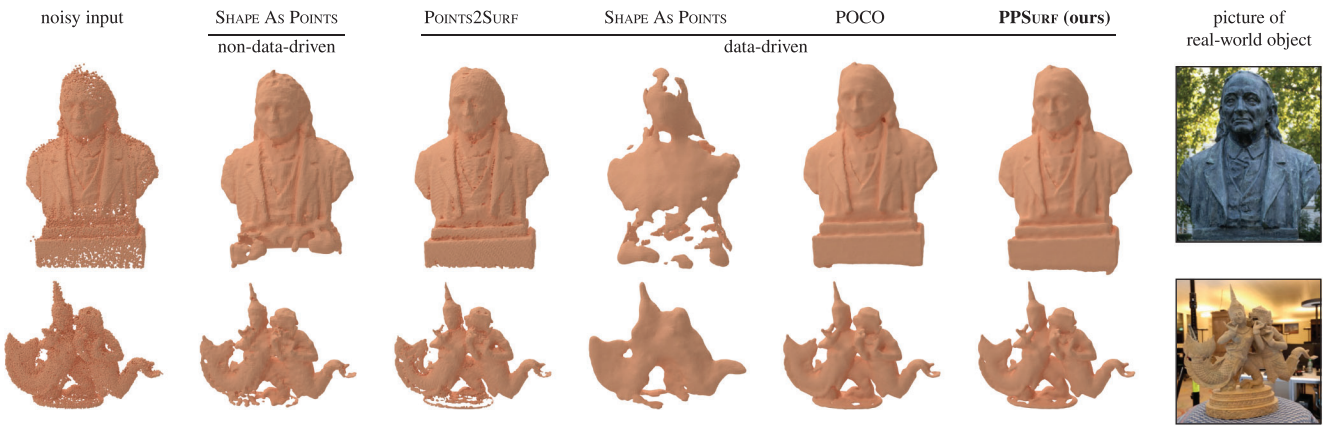

Figure 5: *Real-world reconstructions. We compare to all baselines on the two point clouds that were obtained from real-world objects.*

Table 2: *Comparison of reconstruction times and memory usage. We show the mean reconstruction time per shape and the maximum GPU-memory consumption for each method on the ABC var noise dataset. 200NN uses reconstruction batch size 25k instead of 50k. PGR went out of memory on 21 shapes*

	Time per shape	Max GPU memory
PGR	1.9 min	>46GB
SAP-O	1.1 min	3.8GB
SAP	0.8 s	3.1GB
P2S	13.5 min	14.3GB
POCO	1.6 min	9.0GB
PPSurf 10NN	1.6 min	9.1GB
PPSurf 25NN	1.7 min	9.1GB
PPSurf 50NN	1.9 min	9.3GB
PPSurf 100NN	2.6 min	13.7GB
PPSurf 200NN	3.5 min	13.2GB

Table 3: *Branch Ablation Study. Using the ABC var-noise test set, we compare PPSurf* Full *to variants with disabled branches. The only-local variant failed to produce some meshes, which are ignored in the metrics. The best results per column are marked in bold*

Model	Chamfer (×100) ⇓	F1 score ⇑	Normal error ⇓
Only local	2.69	0.36	1.56
Only global	0.70	0.89	0.33
PPSurf Full	**0.66**	**0.90**	**0.30**

- *Only Global* is similar to POCO as it omits the local branch. The results show that a global prior can help to obtain reliable reconstructions but with lower performance due to the missing fine details.
- For *Sym Max*, we replace the attention-based interpolation used in the local branch with the max, effectively making this branch a PointNet [QSMG16]. The results show an advantage for attention.

Table 4: *Patch Size Ablation Study. Using the ABC var-noise test set, we compare PPSurf* Full *(which is 50NN) to variants with different patch sizes. The best results per column are marked in bold*

Model	Chamfer (×100) ⇓	F1 score ⇑	Normal error ⇓
PPSurf 10NN	1.10	**0.90**	0.40
PPSurf 25NN	**0.66**	**0.90**	0.31
PPSurf Full	**0.66**	**0.90**	**0.30**
PPSurf 100NN	**0.66**	**0.90**	**0.30**
PPSurf 200NN	0.67	0.89	0.31

Table 5: *Miscellanous Ablation Study. Using the ABC var-noise test set, we compare PPSurf* Full *(which uses Merge Sum and Sym Att) to more variants. The best results per column are marked in bold*

Model	Chamfer (×100) ⇓	F1 score ⇑	Normal error ⇓
PPSurf Sym Max	1.11	**0.90**	0.40
PPSurf QPoints	0.67	0.89	0.31
PPSurf Merge Cat	**0.66**	**0.90**	**0.30**
PPSurf Full	**0.66**	**0.90**	**0.30**

- In *Merge Cat*, we concatenate the features of both branches instead of summing them, which leads to twice the input size for the final MLP. Results show that this is slightly worse than *Full*.
- The *QPoints* variant is the same as *Merge Cat*, but additionally, we concatenate query point coordinates to the input of the learned weighting function f^{lv}. However, this results in a slightly worse performance than *Full* and even *Merge Cat*.
- For the *xNN* variants, we take the x nearest neighbours for local sub-sample. *Full* is equal to 50NN.

4.3. Limitations

Reconstruction times are still non-interactive, due to the need to evaluate the occupancy at a large number of samples. Possibilities for speed-ups include more efficient sampling strategies to use fewer query points.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

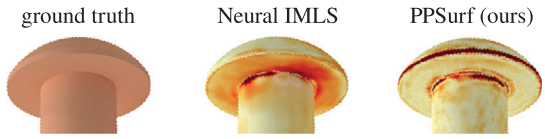

Figure 6: *Limitations. Our method has difficulties to recover the edges of clean point clouds due to training with noisy point clouds.*

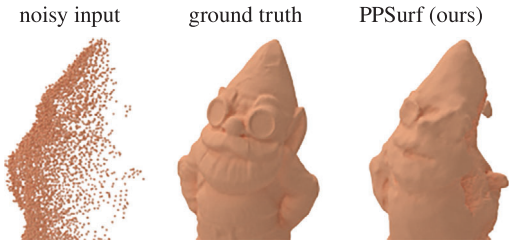

Figure 7: *Limitations. Our method struggles with reconstructions of large missing areas in the input point cloud since we did not incorporate any generative model capabilities.*

As our learned priors were trained on noisy data to make PPSurf more robust to noise, they also bias the reconstructed surface to some extent towards the distributions learned by the priors. This results in some loss of accuracy when applied to noise-free point clouds compared to some of the non-data-driven methods (see Figure 6). Learning a prior that is specialized to noise-free point clouds, or including more noise-free point clouds in our training set would alleviate this issue.

While PPSurf is better than the baselines in filling scan shadows, it is not a generative method and cannot generate new geometric detail in large missing regions. This limits the size of missing regions that can be filled with plausible geometry. Combining PPSurf with a generative model would be an interesting direction for future work. See Figure 7 for an example of inaccurately filled scan shadows.

5. Conclusion

In this paper, we have introduced PPSurf as a method for surface reconstruction from raw, unoriented point clouds. In contrast to previous methods, PPSurf incorporates strong local and global priors learned from data. Whilst our global prior is based on a point convolutional neural network that processes the point cloud as a whole, fine details are preserved through the local prior based on dense local point cloud patches. We have shown in extensive studies that PPSurf is able to achieve better surface reconstructions than previous data-driven and non-data-driven methods, being more robust to noise in the input point cloud and preserving fine details at the same time.

In the future, we would like to investigate how modern techniques borrowed from generative models could improve the obtained reconstruction from sparse point clouds where large parts of the shape are missing.

Acknowledgements

This work has been supported by the FWF projects P24600-N23 and P32418-N31, the WWTF project ICT19-009 and the EU MSCA-ITN project EVOCATION (Grant Agreement 813170).

References

[AL20] Atzmon M., Lipman Y.: Sal: Sign agnostic learning of shapes from raw data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[AL21] Atzmon M., Lipman Y.: SALD: sign agnostic learning with derivatives. In *International Conference on Learning Representations, ICLR* (2021).

[BM22] Boulch A., Marlet R.: POCO: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 6302–6314.

[BPM20] Boulch A., Puy G., Marlet R.: FKAConv: Feature-kernel alignment for point cloud convolution. In *15th Asian Conference on Computer Vision (ACCV 2020)* (2020).

[BTBW77] Barrow H. G., Tenenbaum J. M., Bolles R. C., Wolf H. C.: Parametric correspondence and chamfer matching: Two new techniques for image matching. In *IJCAI'77: Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2* (San Francisco, CA, USA, 1977), Morgan Kaufmann Publishers Inc., pp. 659–663. http://dl.acm.org/citation.cfm?id=1622943.1622971

[BYSZ22] Baorui M., Yu-Shen L., Zhizhong H.: Reconstructing surfaces for sparse point clouds with on-surface priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[BZYSM21] Baorui M., Zhizhong H., Yu-Shen L., Matthias Z.: Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *International Conference on Machine Learning (ICML)* (2021).

[CAPM20] Chibane J., Alldieck T., Pons-Moll G.: Implicit functions in feature space for 3D shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[CHL23] Chen C., Han Z., Liu Y.-S.: Unsupervised inference of signed distance functions from single sparse point clouds without learning priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[CMPM20] Chibane J., Mir A., Pons-Moll G.: Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems (NeurIPS)* (Dec. 2020).

[CTFZ22] Chen Z., Tagliasacchi A., Funkhouser T., Zhang H.: Neural dual contouring. *ACM Transactions on Graphics (Special Issue of SIGGRAPH) 41*, 4 (2022), 1–13.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[CZ19] Chen Z., Zhang H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[DDN20] Dai A., Diller C., Nießner M.: SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)* (2020), IEEE.

[EGO*20] Erler P., Guerrero P., Ohrhallinger S., Mitra N. J., Wimmer M.: Points2Surf: Learning implicit surfaces from point clouds. In *European Conference on Computer Vision (ECCV)* (2020).

[FSG17] Fan H., Su H., Guibas L. J.: A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 605–613.

[GKOM18] Guerrero P., Kleiman Y., Ovsjanikov M., Mitra N. J.: PCPNet: Learning local shape properties from raw point clouds. *Computer Graphics Forum 37*, 2 (2018), 75–85.

[GKUP11] Gschwandtner M., Kwitt R., Uhl A., Pree W.: Blensor: Blender sensor simulation toolbox. In *International Symposium on Visual Computing* (2011), Springer, pp. 199–208.

[GYH*20] Gropp A., Yariv L., Haim N., Atzmon M., Lipman Y.: Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)* (2020).

[HCJ19] Huang Z., Carr N., Ju T.: Variational implicit point set surfaces. *ACM Transactions on Graphics 38*, 4 (July 2019). https://doi.org/10.1145/3306346.3322994

[HWW*22] Hou F., Wang C., Wang W., Qin H., Qian C., He Y.: Iterative Poisson surface reconstruction (iPSR) for unoriented points. *ACM Transactions on Graphics 41*, 4 (July 2022). https://doi.org/10.1145/3528223.3530096

[JSM*20] Jiang C., Sud A., Makadia A., Huang J., Nießner M., Funkhouser T.: Local implicit grid representations for 3D scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2020).

[KBH06] Kazhdan M., Bolitho M., Hoppe H.: Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing* (2006).

[KH13] Kazhdan M., Hoppe H.: Screened Poisson surface reconstruction. *ACM Transactions on Graphics (ToG) 32*, 3 (2013), 29.

[KMJ*19] Koch S., Matveev A., Jiang Z., Williams F., Artemov A., Burnaev E., Alexa M., Zorin D., Panozzo D.: ABC: A big CAD model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

[LC87] Lorensen W. E., Cline H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *ACM SIGGRAPH Computer Graphics* (1987), vol. *21*, ACM, pp. 163–169.

[LESP21] Lionar S., Emtsev D., Svilarkovic D., Peng S.: Dynamic plane convolutional occupancy networks. In *Winter Conference on Applications of Computer Vision (WACV)* (2021).

[LXSW22] Lin S., Xiao D., Shi Z., Wang B.: Surface reconstruction from point clouds without normals by parametrizing the Gauss formula. *ACM Transactions on Graphics 42*, 2 (Oct. 2022). https://doi.org/10.1145/3554730

[MLH23] Ma B., Liu Y.-S., Han Z.: Learning signed distance functions from noisy 3D point clouds via noise to noise mapping. In *International Conference on Machine Learning (ICML)* (2023).

[MON*19] 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 (CVPR)* (2019).

[MZLH23] Ma B., Zhou J., Liu Y.-S., Han Z.: Towards better gradient consistency for neural signed distance functions via level set alignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[PFS*19] Park J. J., Florence P., Straub J., Newcombe R., Lovegrove S.: DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 165–174.

[PJL*21] Peng S., Jiang C. M., Liao Y., Niemeyer M., Pollefeys M., Geiger A.: Shape as points: A differentiable poisson solver. In *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[PNM*20] Peng S., Niemeyer M., Mescheder L., Pollefeys M., Geiger A.: Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)* (2020).

[QSMG16] Qi C. R., Su H., Mo K., Guibas L. J.: PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 652–660.

[RLBG*19] Rakotosaona M.-J., La Barbera V., Guerrero P., Mitra N. J., Ovsjanikov M.: PointCleanNet: Learning to denoise and remove outliers from dense point clouds. *Computer Graphics Forum 39* (2019), 185–203.

[SMB*20] Sitzmann V., Martel J. N., Bergman A. W., Lindell D. B., Wetzstein G.: Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[STM*21] Siddiqui Y., Thies J., Ma F., Shan Q., Nießner M., Dai A.: RetrievalFuse: Neural 3D scene reconstruction with a database. In *International Conference on Computer Vision (ICCV)* (2021).

[TH15] Taha A. A., Hanbury A.: Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging 15*, 1 (2015), 1–28.

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[TLX*21] Tang J., Lei J., Xu D., Ma F., Jia K., Zhang L.: SA-ConvONet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021).

[UK21] Ummenhofer B., Koltun V.: Adaptive surface reconstruction with multiscale convolutional kernels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

[WLT22] Wang P.-S., Liu Y., Tong X.: Dual octree graph networks for learning adaptive volumetric shape representations. *ACM Transactions on Graphics 41* (2022), 1–15.

[WSS*19] Williams F., Schneider T., Silva C. T., Zorin D., Bruna J., Panozzo D.: Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 10130–10139.

[WWW*22] Wang Z., Wang P., Wang P., Dong Q., Gao J., Chen S., Xin S., Tu C., Wang W.: Neural-IMLS: Self-supervised implicit moving least-squares network for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* (2023), 1–16. https://doi.org/10.1109/TVCG.2023.3284233

[XSL*23] Xiao D., Shi Z., Li S., Deng B., Wang B.: Point normal orientation and surface reconstruction by incorporating isovalue constraints to Poisson equation. *Computer Aided Geometric Design 103* (2023), 102195. https://www.sciencedirect.com/science/article/pii/S0167839623000274

[YWOSH20] Yifan W., Wu S., Oztireli C., Sorkine-Hornung O.: Iso-Points: Optimizing neural implicit surfaces with hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[ZJ16] Zhou Q., Jacobson A.: Thingi10K: A dataset of 10,000 3D-printing models. *arXiv preprint arXiv:1605.04797* (2016). https://ten-thousand-models.appspot.com/

[ZML*22] Zhou J., Ma B., Liu Y.-S., Fang Y., Han Z.: Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

[ZTNW23] Zhang B., Tang J., Nießner M., Wonka P.: 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics 42*, 4 (2023), 92. https://doi.org/10.1145/3592442

[ZVW*22] Zeng X., Vahdat A., Williams F., Gojcic Z., Litany O., Fidler S., Kreis K.: Lion: Latent point diffusion models for 3D shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information

© 2024 The Authors. *Computer Graphics Forum* published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.15000 by Readcube (Labtiva Inc.), Wiley Online Library on [17/01/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License