Vis2Mesh: Efficient Mesh Reconstruction from Unstructured Point Clouds of Large Scenes with Learned Virtual View Visibility

Shuang Song
The Ohio State University

Zhaopeng Cui*
Zhejiang University

Rongjun Qin†
The Ohio State University

Abstract

We present a novel framework for mesh reconstruction from unstructured point clouds by taking advantage of the learned visibility of the 3D points in the virtual views and traditional graph-cut based mesh generation. Specifically, we first propose a three-step network that explicitly employs depth completion for visibility prediction. Then the visibility information of multiple views is aggregated to generate a 3D mesh model by solving an optimization problem considering visibility in which a novel adaptive visibility weighting in surface determination is also introduced to suppress line of sight with a large incident angle. Compared to other learning-based approaches, our pipeline only exercises the learning on a 2D binary classification task, i.e., points visible or not in a view, which is much more generalizable and practically more efficient and capable to deal with a large number of points. Experiments demonstrate that our method with favorable transferability and robustness, and achieve competing performances w.r.t. state-of-the-art learning-based approaches on small complex objects and outperforms on large indoor and outdoor scenes. Code is available at https://github.com/GDAOSU/vis2mesh.

1. Introduction

3D surface reconstruction is essential to drive many computer vision and VR/AR applications, such as vision-based localization, view rendering, animation, and autonomous navigation. With the development of 3D sensors, *e.g.*, LiDAR or depth sensors, we can directly capture accurate 3D point clouds. However, determining surfaces from unstructured point clouds remains a challenging problem, especially for point clouds of objects of complex shapes, varying point density, completeness, and volumes. A favorable mesh reconstruction method should be 1) capable of recovering geometric details captured by the point clouds; 2) robust and generalizable to different scene contexts; 3) scalable to large scenes and tractable in terms of memory and
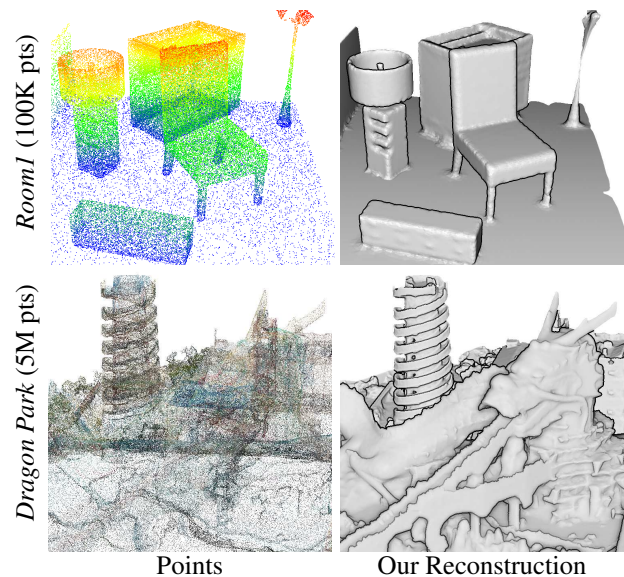

Figure 1. Examples of reconstructed surfaces with our approach on both indoor and large outdoor scenes. The number of points ranges from thousands to millions.

computation.

Typical reconstruction methods either explicitly explore the local surface recovery through connecting neighboring points (*e.g.* Delaunay Triangulation [5]), or sort solutions globally through implicit surface determination (*e.g.* Screened Poisson Surface Reconstruction [21, 22]). However, both of them favor high-density point clouds. With the development of 3D deep learning, many learning-based methods have demonstrated their strong capability when reconstructing mesh surfaces from moderately dense or even comparatively sparse point clouds of complex objects. However, these end-to-end deep architectures, on one hand, encode the contextual/scene information which often runs into generalization issues; on the other hand, the heavy networks challenge the memory and computations when scaling up to large scenes.

The point visibility in views has been shown as a good source of information, which can greatly benefit surface reconstruction [47]. For example, a 3D point visible on a physical image view alludes to the fact the line of sight from

---

*Contribution initiated and performed at ETH Zürich.
†Corresponding author. Email: qin.324@osu.edu

the 3D point to the view perspective center, travels through free space (no objects or parts occludes), which as a result, can serve as a strong constraint to guide the mesh reconstruction. Intuitively, the more visible a point is in more views, the more information about the free space between points and views that one can explore [17]. This principle has been practiced in multi-view stereo (MVS) and range images and shows promising results [25, 26, 51], in which the visibility of each 3D point to each physical is recorded through dense matching (in MVS) or decoded directly from the sensors. However, in both cases, the visibility is limited by the number of physical views and accounts for only limited knowledge of the free space, thus leading to the lack of reconstruction details.

To address it, we propose a novel solution by generating a large number of virtual views around the point clouds and utilize the learned visibility through a graph-cut based surface reconstruction approach. This will literally give us unlimited visibility information for high-quality mesh generation through traditional approaches based on visibility. However, it is non-trivial to design such a system. First, visibility prediction in virtual views is not easy as the projected pixels from the point clouds to different views can have varying sparsity. Second, unlike the MVS views which normally assume a good incidence angle to the generated point clouds, the generated virtual views may present a large variation in terms of their incident angle and distance to the surface. To solve these problems, we design a network for visibility prediction, which utilizes partial convolution to model the sparsity of the input in the neural network. To further improve the accuracy, we propose a cascade network that explicitly employs depth completion as an intermediate task. Secondly, to suppress the side effect of unfavored virtual rays (large-incidence angle to the surface), we propose a novel adaptive visibility weighting term that adaptively weights the virtual rays. Compared to the end-to-end learning-based methods, our method has better generalization across different scene contexts because a very simple and learning-friendly task, *i.e.*, visibility prediction, is involved in our pipeline. Moreover, our method also maintains the capability to process an extremely large volume of outdoor point clouds with high quality.

Our **contributions** can be summarized as follows: Firstly, we present a surface reconstruction framework from unstructured point clouds, that exploits the power of both traditional and learning-based methods; Secondly, we propose a three-step network that explicitly employs the depth completion for the visibility prediction of 3D points; Thirdly, we propose a novel adaptive visibility weighting term into the traditional graph-cut based meshing pipeline, to increase the geometric details; Lastly, we demonstrate through our experiments that our proposed method yields better generalization capability, and can be scaled up to process point clouds of a large scene. It is also robust against different types of noises and incomplete data and has better performance than the state-of-the-art methods.

2. Related Work

Classic surface reconstruction from unstructured point clouds exploits both local and global surface smoothness, global regularity, visibility, *etc*. [3]. The *local surface smoothness* based methods [12, 16, 1] seek for smooth surface only in close proximity to the data. The *global surface smoothness* based methods [20, 7, 21, 22, 24] tend to find a field function (signed distance function, indicator function) approximating the point cloud. The most common global smoothness algorithm is Screened Poisson Surface Reconstruction (SPSR) [21, 22] which can generate smooth, void-free surfaces with high-quality oriented normals, while it can be oversensitive to incorrect normals, as well as non-watertight shapes. The *global regularity* based methods [33, 56, 27] deal with the man-made objects or architectural shapes that feature symmetry, orthogonality, repetition, and parallelism. The *visibility* based methods assume the information of the sensors from which the point clouds are collected, and this yields additional cues as rays and visibility of each point at sensor locations in which they are observable. Varying forms of visibility (includes lines of sight [11, 25], exterior visibility [45, 19], parity [32]) have been explored. The most widely used methods utilize lines of sight [25, 26, 17, 18, 59] that determine the surface between interior and exterior tetrahedron of the Delaunay triangulation of a point set by solving a *s-t* graph which is weighted by lines of sight. Nonetheless, the quality of results relies on the number and quality of rays and it may fall short in concave areas on the reconstructed surface if insufficient rays are identified [59].

Learning-based surface reconstruction takes a data-driven approach that takes advantage of the available examples and the high-capability network structures [40, 41, 9, 52] to build surfaces for challenging and complex point clouds that classic methods have poor performance on. Several methods [8, 31, 35] encode the entire object to an embedding vector and decode it with a neural network, as a general problem in such deep networks, the representation often cannot be generalized to objects of unseen categories or scenes. CONet [36] proposed alternative volumetric representations and enabled convolution in the implicit field making the representations invariant to translation, and the sliding-window nature of this approach makes CONet capable of scaling up to large scenes, while the method being local does not capture global shape information and the volumetric representations are still memory demanding for 3D data generation. Points2Surf [13] proposed a patch-based learning framework combining outputs of local-scale and global-scale neural networks to construct a signed distance
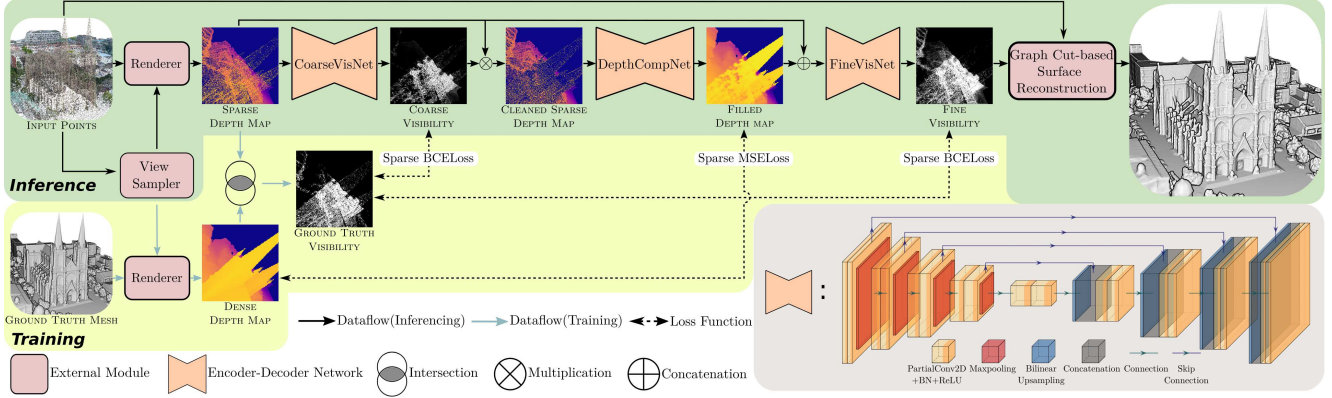
Figure 2. The proposed Vis2Mesh framework. Our framework reconstructs surfaces in *four* steps shown in inference region from left to right follow the flow of arrows: 1) Virtual view sampling 2) Rendering 3) Visibility determination and 4) Delaunay and graph-cut based surface reconstruction. Visibility determination composed by three networks: **CoarseVisNet**, predicting visibility from sparse depth input, **DepthCompNet** completing dense depth map based on coarse predicted visible points and **FineVisNet** refining visibility prediction with sparse depth and completed dense depth map, detailed in Section 4.2.

function (SDF) and shows a certain level of generalization capability. Yet the query on the per-voxel level and the inference over the voxels field are computationally intensive (4x slower than SPSR), which limits it to only process small scenes. The MIER method [30] uses the explicit representation of input points and cast the connectivity to a surrogate comparing geodesic/euclidean distances. A neural network-based classifier is trained to determine surface triangles, which has shown to be capable of preserving fine-grained details and works well on ambiguous structures. However, the per-triangle classification overlooks the connectivity of adjacent triangles, causing holes on the reconstructed surfaces. Point2Mesh [15] proposed a self-prior to detect self-similarity, which contributes to reconstructing surface as well as removing noises and completing missing parts. Nonetheless, Point2Mesh only works with watertight and simple models, requires an initial mesh model, and comparably needs a long running time to converge. As compared to the classic methods, the learning-based methods, in general, demand a high volume of memory and can be scene-specific.

3. Preliminary

This section introduces the formula of Delaunay triangulation and graph-cut optimization based surface reconstruction [26]. The surface reconstruction problem is considered as the outer space/inner space labeling problem of the connected graph built based on Delaunay triangulation, to be solved by minimum s-t cut algorithm [6]. The reconstructed surface, denoted by S in the following, is a union of oriented Delaunay triangles which are guaranteed to be watertight and intersection free as it bounds a volume consist of tetrahedra. The energy function of the reconstruction problem is formulated by the basic method [26] as Equa-

tion 1.

$$
\begin{aligned}
E(S) &= E_{vis}(S, \mathcal{V}) + \lambda_{ql} E_{ql}(S) \\
&= \sum_{v \in \mathcal{V}} [E_s(v) + E_t(v) + E_{ij}(v)] + \lambda_{ql} E_{ql}(S),
\end{aligned} \quad (1)
$$

where $E_{vis}(S, \mathcal{V})$ is a sum of penalties for conflicts and wrong orientations of the surface S concerning the constraints imposed by all lines of sight $\mathcal{V}$, $E_{ql}(S)$ is an additional smoothness term to suppress skinny triangles [26]. In graph-cut, E_s and E_t are data terms for outer and inner space respectively, and E_{ij} is a smoothness term. $E_{ql}(S)$ is another smoothness term independent of visibility.

$$
E_s(v) = \infty \cdot \delta[T_1 \in t], \quad (2a)
$$

$$
E_t(v) = \alpha_{vis} \cdot \delta[T_{M+1} \in s], \quad (2b)
$$

$$
E_{ij}(v) = \sum_{i=1}^{M-1} \alpha_{vis} \cdot \delta[T_i \in s \wedge T_{i+1} \in t], \quad (2c)
$$

where T_i is the i-th tetrahedron intersects with the line of sight v, M is the size of the intersection set. So that T_1 and T_{M+1} denote the first and the one right behind the last tetrahedra along v. The conditional operator $\delta[\cdot]$ is 1 if the condition is true, otherwise is 0. α_{vis} is the confidence of the visibility which is proportional to the distance to the endpoint, which is called soft visibility constraint. Please refer to our supp. material for more details about graph weighting.

4. Method

In this section, we introduce our framework of surface reconstruction from unstructured point clouds (Figure 2). Firstly, we sample feasible 6 DoF (Degree of freedom) poses according to input point clouds. Then, a 3D renderer will take poses as viewing parameters and project points onto that virtual image plane based on its projection matrix. Next, a three-step network that explicitly employs
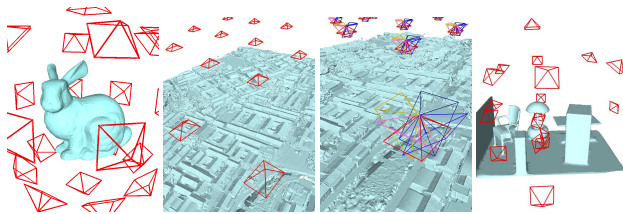
(a) Spherical   (b) Grid+Nadir   (c) Grid+Oblique (d) Customized

Figure 3. Virtual view generators. **Spherical Pattern Generator**: samples around the object while targeting it, always used for small objects. **Grid+Nadir Generator**: samples in grid fashion that parallel to the ground plane for the terrain-like large-scale scenario, uses above ground height and overlap rate to control the density of sampled views. **Grid+Oblique generator**: beside nadir views, samples concentric oblique views to capture facades in the urban area. **Customized**: we create an interactive tool to let users pick views. Customized views can improve the quality of ambiguous structures of complex objects.

depth completion to assist visibility estimation, and predicts the visible and occluded points to build visibility information. Finally, the graph-cut based solver processes the input points associated with predicted visibility to reconstruct the surface. Our overall framework is shown in Figure 2, and the individual components will be discussed in the following subsections.

4.1. Virtual View Sampling

Given the input unstructured point cloud, we first generate the images under virtual views that simulate the input images in the MVS method. Virtual view sampling given a presumed object or scene is a challenging problem, and automatic view selection for tasks such as stereo reconstruction or surveillance is an active topic in computer vision and robotics community [37, 46, 54, 55], and the performance of these methods are task and scene-specific. In our work, we utilize several general generators as shown in Figure 3 and empirically find that they work very well with our method when dealing with different objects and scene contexts. In practice, our method shows a certain level of robustness regarding varieties of virtual view sampling, and the number of virtual views can affect surface quality, the analysis of which can be found in Section 5.2.

4.2. Visibility Estimation Network

Given the projected image with sparse depth information, we exploit the deep neural network to predict the visibility of 3D points in the virtual views. The biggest challenge of the visibility estimation task is that it requires pixel-wise accuracy rather than region-wise. To this end, the standard CNNs (Convolutional Neural Networks) with small kernel size is preferred for the task because it is sensitive to local structure. However, for data with varying sparsity levels, the standard CNNs have poor performance since

invalid pixels or voids produced by projecting sparse points make it difficult to learn robust representations [49].

Partial Convolution (PConv) [29, 28] is proposed for image inpainting from an incomplete input image. The proposed re-weighted convolution operation is particularly suitable for processing sparse data since it models the validity of each pixel and propagates it along with the convolutional operation. A similar idea has been used in depth completion [49] from sparse laser scan data. Visibility prediction exploits the local sensitivity of neural networks, while depth completion requires local smoothness. Therefore, we design an intermediate task with a dedicated network module for depth completion and proposed a cascade network VDVNet (Visibility-Depth-Visibility).

The input of our visibility estimation network is a $H \times W \times 2$ feature map in which the first channel is a normalized depth image, and the second is a binary mask that indicates whether the pixel is valid. Our frame shown in Figure 2 includes an example input and intermediate results between different network components (which will be detailed in the following sections). The specifications of the network architecture can be found in our supplementary material.

CoarseVisNet. We train the sub-network to predict the visibility with supervised learning. As shown in Figure 2, we apply Sparse BCELoss (binary cross-entropy loss with mask) between the predicted coarse visibility map and the ground truth visibility map, where the mask indicates the valid pixels.

DepthCompNet. The depth completion sub-network is designed to convert a sparse depth map to a dense one. We filter raw depth map with coarse visibility predicted by CoarseVisNet before feeding it to DepthCompNet. The supervision of DepthCompNet is depth maps of ground truth surface. Mean squared error loss function is applied between them (Sparse MSELoss in Figure 2).

FineVisNet. This module takes the input of CoarseVisNet and the output of DepthCompNet to predict the fine visibility. The loss function is applied as same as CoarseVisNet.

Training Data. The training data used for visibility prediction can be easily simulated from any type of 3D model. We select a few well-reconstructed textured mesh models of large-scale scenes from a public MVS reconstruction dataset [53] as the *ground truth surfaces* to generate synthetic visibility. Points are uniformly and sparsely sampled from surfaces. Then, the sampled points are further augmented with additional Gaussian white noises and outliers. In the next step, we sample virtual views from the point clouds with the combination of patterns shown in Figure 3. For each view, a depth map of both point clouds and surface are rendered and combined to generate the *ground truth visibility*. If the depth value of the point rendered image and the surface rendered image are matched, the point is labeled as visible, otherwise, as occluded. Depth maps of the sur-

(a) Sharp angle

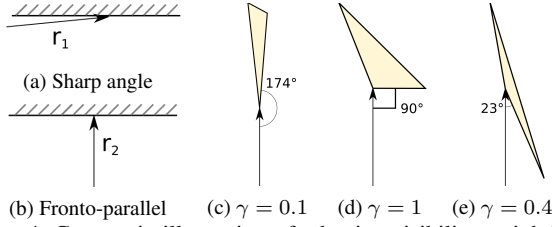(b) Fronto-parallel   (c) $\gamma = 0.1$ (d) $\gamma = 1$ (e) $\gamma = 0.4$

Figure 4. Geometric illustration of adaptive visibility weighting, where $\lambda_{avw} = 1$.

faces serve as *ground truth depth* for the training of Depth-CompNet.

4.3. Adaptive Visibility Weighting of Graph-Cut based Surface Reconstruction

Once the visibility of points in all views is predicted, we utilize the graph-cut based mesh generation [26] to reconstruct the final mesh model. Due to the different characteristics of virtual visibility, the basic method [26] suffers from the over-smoothing issue of creases on the surface, which motivates us to propose adaptive visibility weighting. The formulas of the basic method can be found in Section 3.

Compared with the visibility from instruments, the virtual view visibility generated by our VDVNet from unordered points and sampled views has very different characteristics in terms of density, direction, and distance. As Equation 2b indicates, the basic method has an issue that the strong constraint imposed on the tetrahedra right behind the point along the line of sight and might overlook details of the surface especially when the lines of sight have a large incident angle. Since the large incident angle is rare in traditional applications, the issue of the baseline method was not discovered before.

Since our virtual visibility is predicted by the neural network from unordered points directly, the angle of the incidence can range from 0 to 90 degrees, because there is no implicit filter of the multi-stereo view method. The visibility with a large incident angle (Figure 4a) is equivalent to a sharp surface in the depth map of the view, which is always considered as a sign of self-occlusion [4].

The key idea of our adaptive visibility weighting (AVW) is illustrated in Figure 4. Apparently, for surface reconstruction, a measurement for a sharp angle surface (Figure 4a) is considered as the lowest confidence, while for the fronto-parallel surface (Figure 4b) is considered the best confidence. We use cosine similarity between the direction of the line of sight and the surface normal to construct a smooth weighting function. The weight is regarded as the confidence of visibility, its definition overlaps with α_{vis}, and it changes terms E_t and E_{ij}. We use the product of two factors γ and α_{vis} as the new confidence of line of sight. In order to apply the weighting function to a tetrahedron with 3 triangles where the vertex and the line of sight intersect, we consider the one most likely to be on the surface, that is, the triangle whose normal is closest to the direction of the

ray. We took the maximum value of the cosine similarity of the three angles, as shown in Equation 3, where the cosine similarity is equivalent to the dot product because the normal is of unit length. Figure 4c, 4d, 4e show examples of our adaptive visibility weighting term.

$$E_t^{OURS}(v) = \gamma \cdot \alpha_{vis} \cdot \delta[T_{M+1} \in s], \tag{3a}$$

$$E_{ij}^{OURS}(v) = \sum_{i=1}^{M-1} \gamma \cdot \alpha_{vis} \cdot \delta[T_i \in s \wedge T_{i+1} \in t], \tag{3b}$$

$$\gamma = (1 - \lambda_{avw}) + \lambda_{avw} \max(N_v^T \cdot [N_{f1}, N_{f2}, N_{f3}]), \tag{3c}$$

where N_v denotes direction of the incident line of sight v, $N_{f\#}$ are normal of faces of tetrahedron T_{M+1} incident to the endpoint of v, λ_{avw} is the damping factor to adjust the adaptive visibility weighting γ. Our method is equivalent to Labatut's method [26] when $\lambda_{avw} = 0$.

5. Experiments

We evaluate our proposed Vis2Mesh method through a series of quantitative and qualitative experiments covering a variety of small objects, indoor scenarios, large-scale outdoor scenarios either acquired by multi-view stereo, laser scanning, or simulated images. We compare our method with several state-of-the-art learning-based reconstruction approaches including Point2Mesh (P2M) [15], Points2Surf (P2S) [13], Meshing Point Cloud with IER (MIER) [30] Convolutional Occupancy Networks(CONet) [36], and a classical approach Screened Poisson Surface Reconstruction (SPSR) [22]. To demonstrate the generalization capacity, we widely collect data from multiple datasets with varying sensors and platforms, including BlendedMVS [53], COSEG [50], Thingi10K [57], P2M [15], CONet [36], senseFly [44], KITTI [14], Mai City Dataset [48], Airborne LiDAR data of Columbus, Ohio [34] and Airborne LiDAR data of Toronto [43]. Details of these datasets including their source, volume, as well as virtual view generators used in our experiments can be found in the supp. material.

Our method does not assume surface normal information, while it is required for some of the comparing methods (SPSR, P2M, *etc*.). High-quality surface normals are an important source of information to guide the reconstruction method but are not easily obtainable or directly measured by most of the sensors. In the experiment, the surface normal used by these comparing methods is estimated by a Minimal Spanning Tree (MST) based local plane fitting method, implemented by the open-source software Cloud-Compare [10]. The software also integrated SPSR [22] implementation which we use in our comparative study. We report the performance of the comparing methods quantitatively in Section 5.1, and qualitatively in Section 5.3.

5.1. Quantitative Evaluation

We evaluate our method on points sampled from ground truth mesh from COSEG, CONet, and BlendedMVS

		F-score							Chamfer distance						
		P2M	P2S	MIER	CONet	SPSR (Trimmed)	Ours ($\gamma_{av}=0$)	Ours ($\gamma_{av}=1$)	P2M	P2S	MIER	CONet	SPSR (Trimmed)	Ours ($\gamma_{av}=0$)	Ours ($\gamma_{av}=1$)
Object 25K	DSLR	0.9790	0.9014	**0.9935**	0.9041	0.9517	0.9844	0.9841	0.1593	0.3558	0.1440	0.2570	0.1288	**0.0702**	0.0707
	bull	0.9990	0.9946	0.9851	0.8203	**0.9993**	0.9929	0.9991	0.3328	0.3576	0.3421	1.3881	0.2914	0.3037	**0.2893**
	giraffe	0.9969	0.9525	0.9938	0.9776	**0.9999**	0.9989	0.9996	0.2227	0.4325	0.3276	0.3831	**0.1893**	0.2136	0.1939
	Average	*0.9916*	*0.9495*	*0.9908*	*0.9007*	*0.9836*	*0.9921*	*0.9943*	*0.2383*	*0.3819*	*0.2712*	*0.6761*	*0.2032*	*0.1959*	*0.1846*
Indoor 100K	room0	/	0.7334	0.8885	0.9112	0.9049	**0.9345**	0.9340	/	2.0151	0.3403	0.6476	0.8272	**0.3296**	0.3571
	room1	/	0.5379	0.8453	0.8695	0.8065	0.8747	**0.8775**	/	2.8026	0.6714	0.7814	1.2546	0.5971	**0.5457**
	Average	*/*	*0.6356*	*0.8669*	*0.8903*	*0.8557*	*0.9046*	*0.9058*	*/*	*2.4088*	*0.5059*	*0.7145*	*1.0409*	*0.4634*	*0.4514*
Outdoor 500K	Church	/	/	/	0.4086	0.8292	0.9206	**0.9221**	/	/	/	19.8253	1.7784	0.6811	**0.6657**
	Archway	/	/	/	0.9133	0.9815	**0.9840**	0.9772	/	/	/	0.2753	0.0869	**0.0677**	0.0756
	Pedestrian street	/	/	/	0.7494	0.9401	**0.9889**	0.9860	/	/	/	0.8008	0.3561	**0.1590**	0.1620
	Eco Park	/	/	/	0.7416	0.8470	0.9431	**0.9553**	/	/	/	0.8813	0.9463	0.3273	**0.2961**
	Dragon Park	/	/	/	0.7700	0.8702	0.9804	**0.9866**	/	/	/	0.8696	0.7398	0.2096	**0.1859**
	Average	*/*	*/*	*/*	*0.7166*	*0.8936*	*0.9634*	*0.9654*	*/*	*/*	*/*	*4.4809*	*0.7815*	*0.2890*	*0.2770*
MVS	Hotel	/	/	/	0.3577	0.7732	0.8229	**0.8350**	/	/	/	1.3053	0.9369	0.3660	**0.3479**
	GSM Tower	/	/	/	0.1863	0.6261	0.6321	**0.6354**	/	/	/	6.6513	0.8051	0.6935	**0.6888**
	UThammasat	/	/	/	0.4656	**0.8046**	0.7094	0.7259	/	/	/	1.4951	0.3741	0.3982	**0.3737**
	Average	*/*	*/*	*/*	*0.3365*	*0.7346*	*0.7215*	*0.7321*	*/*	*/*	*/*	*1.7845*	*0.7054*	*0.4859*	*0.4701*

Table 1. Quantitative comparison of reconstruction on COSEG, CONet, and BlendedMVS dataset, grouped by the size of the scene and number of points. ("F-score": Higher is better. "Chamfer distance": Lower is better. "/": Not applicable.)

datasets. Since the existing methods have certain limitations on the amount of data, we divide the data set into object/indoor/outdoor scenes, and the complexity and scene scale are gradually increasing. We uniformly sample 25K points from the ground truth surface for objects, 100K points for indoor data, and 500K points for outdoor data. In order to evaluated with actual data, we generated ground truth for the sensefly [44] dataset by using a commercial MVS pipeline [39] that employs SPSR for surface reconstruction. Table 1 shows F-score [23] and Chamfer distance (CD) [2] metrics on each category. To be noted that, we use the sliding window manner of CONet for all scales, and we trim SPSR output by the density value, and the threshold is set to $max\_density/2$.

Compared with several state-of-the-art learning-based reconstruction approaches and the SPSR method, our virtual view visibility-based methods present outstanding performance in terms of both metrics on each level of datasets, which also proves the insensitivity of our method to the type of scene. Furthermore, although our VDVNet is trained from the synthetic dataset collected from outdoor-level models, and the quantitative evaluation shows our method has achieved competitive results on object-level and indoor-level datasets. Regarding efficiency, our method has similar speed and memory as SPSR when reaching the same level of detail. Detailed statistics on running time can be found in the supp. material.

5.2. Ablation Studies

We first demonstrate the advantage of virtual view visibility in surface reconstruction even if the actual visibility is available (Figure 6), and show the quality improvement by the proposed adaptive visibility weighting (Figure 7). Then, we analyze the network architecture with a quantitative evaluation (Table 2).

Importance of virtual view visibility. In this experiment, we compare surfaces reconstructed by real visibility (obtained from physical views) and virtual view visibility to

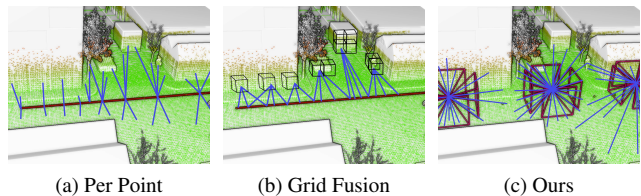(a) Per Point   (b) Grid Fusion   (c) Ours

Figure 5. Methods of modeling visibility from mobile LiDAR scans. (Per Point: each LiDAR point only connects to the trajectory once. Grid Fusion: points within the voxel of the grid fuse into a centroid and connect it to the trajectory. Ours: pure virtual visibility generated from sampled views and VDVNet.)

prove the importance of virtual visibility. The data in this experiment is LiDAR scans associated with 6DoF pose trajectory from the Mai City Dataset [14]. We present three methods as shown in Figure 5. *Per Point* is generated by associating the corresponding scan pose with each point, and each point has only one line of sight (or visibility) collected from the sensor. For *Grid Fusion*, we use a voxel grid filter to aggregate nearby points' visibility to obtain points with multiple actual visibility. Finally, we use the proposed network to generate virtual view visibility, where the virtual views are sampled along the recorded trajectory.

The visual results shown in Figure 6 demonstrates that the virtual views visibility place a significant improvement in the mesh quality. Per Point actual visibility failed to reconstruct a complete scene because only one line of sight associated with each point is not geometrical stable, and is not sufficient to create a reliable weighted graph for the optimization. Grid Fusion shows better reconstruction quality since more lines of sight are included (4/points on average), but the details of reconstruction are poor, and noises seriously affect the surface. In contrast, our pure virtual view visibility method creates smooth, detailed, and noise-free surfaces (thanks to the denoising capability of VDVNet).

Importance of adaptive visibility weighting. Figure 7 demonstrates the impact of proposed adaptive visibility weighting. To be noted that when setting $\lambda_{avw} = 0$, the
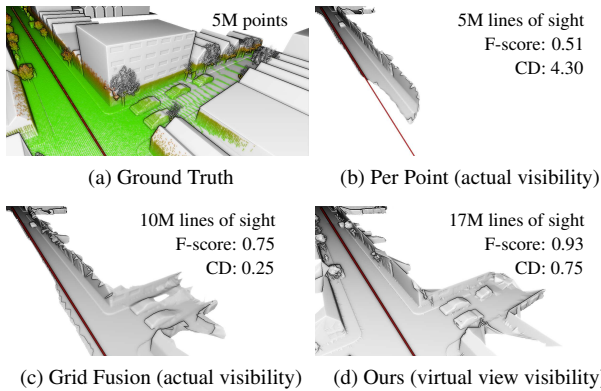
(a) Ground Truth   (b) Per Point (actual visibility)

(c) Grid Fusion (actual visibility)   (d) Ours (virtual view visibility)

Figure 6. Importance of virtual view visibility. Reconstruction using our method for each visibility dataset in Figure 5.

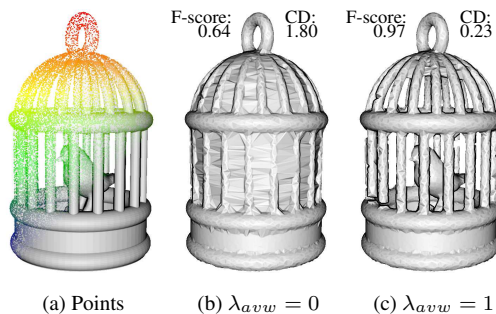(a) Points   (b) $\lambda_{avw} = 0$ (c) $\lambda_{avw} = 1$

Figure 7. Impact of AVW on synthetic data with complex shape. The data is *Birdcage*.

visibility weighted graph is exactly equivalent to the basic method [26]. We observe the method $\lambda_{avw} = 1$ works better than the basic method for objects with pole-like and notches on the surface structure, since such a complex structure may introduce large incidence angles for our virtual lines of sight. The adaptive visibility weighting works as we expected in Section 4.3. The quantitative results in Table 1 show that in most cases, adaptive visibility weighting works better than the basic method.

Impact of the number of virtual views. To evaluate the effect of the view configuration, we demonstrate the reconstruction results w.r.t. different numbers of virtual views in Figure 8. Firstly, the single view on the top center of the scene generates a very coarse surface. Secondly, 7 virtual views on the spherical orbit around the target carved out objects of the scene, the base of the lamp, legs of the chair starting emerging. With more virtual views, the reconstructed surface yields more geometric details. The last configuration contains virtual views selected interactively by the user to reconstruct the complex structure of the chair.

Visibility estimation networks. We independently evaluate the performance of our VDVNet in the task of visibility prediction comparing with geometric-based method Hidden Point Removal (HPR) [19] implemented by Open3D [58] and the baseline learning method UNet [42]. The VISIB-NET proposed by InvSFM [38] is designed for the same
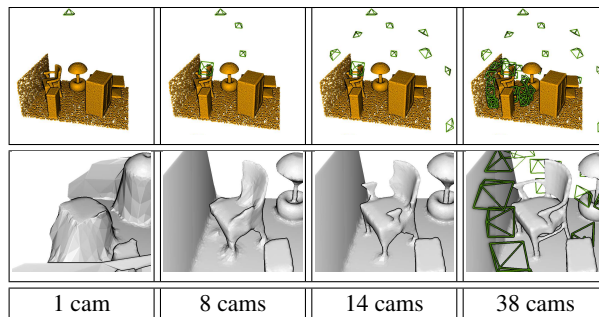

Figure 8. Impact of the number of virtual views.

Visibility Estimator	#Param	%P	%R	%F1	%AUC
HPR	/	85.79	85.01	85.24	82.22
UNet	51M	90.42	87.10	88.68	88.52
UNet + PConv	51M	90.73	88.03	89.32	89.59
VISIBNET	52M	90.15	91.50	90.78	90.22
Ours w/o DepthComp or PConv	153M	91.84	84.08	87.71	89.29
Ours w/o DepthComp	153M	91.38	93.10	92.21	92.62
Ours w/o PConv	153M	91.86	94.24	93.01	93.14
Ours	153M	**92.37**	**94.95**	**93.63**	**94.17**

Table 2. Quantitative analysis of methods on binary visibility classification task. #Param is the number of parameters, %P is precision, %R is recall, %F1 is F1 score, and %AUC is Area Under The Curve of ROC curves. DepthComp represents the depth completion module and PConv denotes t partial convolutional layers.

task as us, and its architecture is UNet with convolutional layers at the end of the decoder.

We report the size and the performance of classifiers in Table 2 on our pixel-wise visibility dataset sampled from BlendedMVS. We observe the networks with partial convolutional layers outperform standard convolutional networks without parameter increasing. The variants of our networks present the contribution of the depth completion module and partial convolutional layers w.r.t. visibility estimation task. Please refer to the supp. material for the evaluation ofdatasets with various modalities as well as the qualitative results of visibility estimation.

Impact of noises and missing data. Since the real 3D point clouds normally have noises and incomplete data, it is very important for mesh generation to deal with these problems. We designed two experiments to evaluate the robustness of different methods against the additional noises and incompleteness.

We evaluated the impact of additional noises (random noise and outliers) on CONet, SPSR, and our method on indoor scenes, as shown in Figure 9. The added noises may affect both the position of the points and the normals of the points. The middle column of Figure 9 shows outliers introduced undesired objects to CONet, but it has much less impact on other methods and ours. However, noises can be problematic for SPSR since it relies on high-quality normal estimation, which can be easily affected by noises. CONet and our method have demonstrated a certain level of noises robustness in the reconstructed meshes. Incomplete data or missing data is a common problem in practice, such as those from laser scanning. Figure 10 compares the results
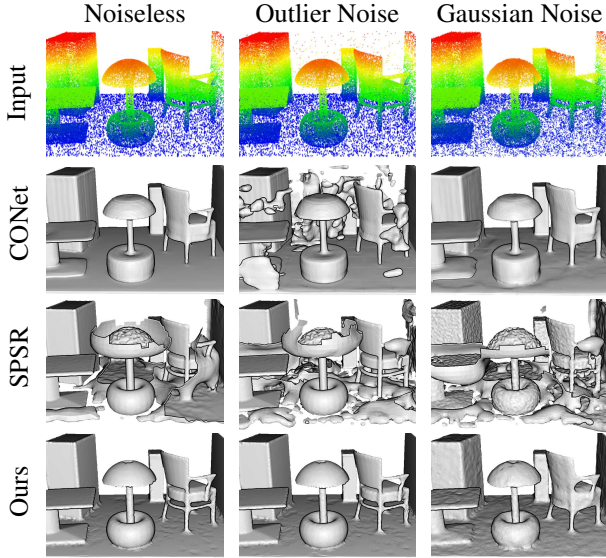
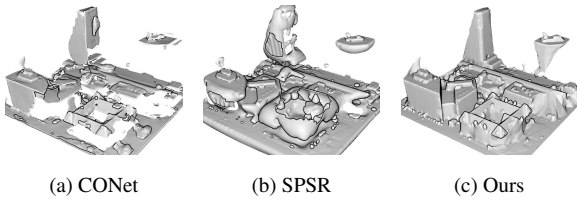Figure 9. Evaluation of the methods subject to additional noises on the point clouds. The data is *Room0*.

(a) CONet   (b) SPSR   (c) Ours

Figure 10. Evaluation of the methods on incomplete point clouds. The data is *Toronto downtown*.

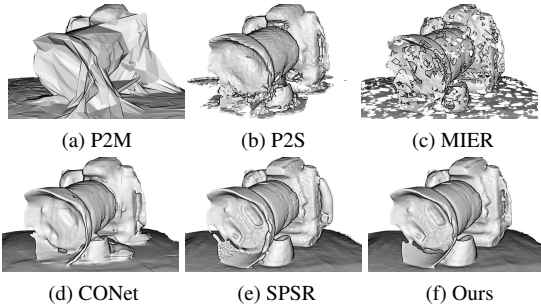(a) P2M   (b) P2S   (c) MIER

(d) CONet   (e) SPSR   (f) Ours

Figure 11. Qualitative comparison on small object level datasets. The data is *DSLR*.

of CONet and ours in large-scale scenes, and we note that our approach is able to complete the missing facades with triangle faces in the incomplete region. Thanks to Delaunay triangulation and tetrahedron labeling, our method is able to close gaps and holes caused by missing data.

5.3. Qualitative Evaluation

Our qualitative comparison is divided into three scales as well, shown in Figure 11, the first column of Figure 9, and Figure 12 respectively. P2M [15] and P2S [13] have poor reconstruction quality in terms of open surface (DSLR). Triangles generated by MIER [30] are close to the ground truth
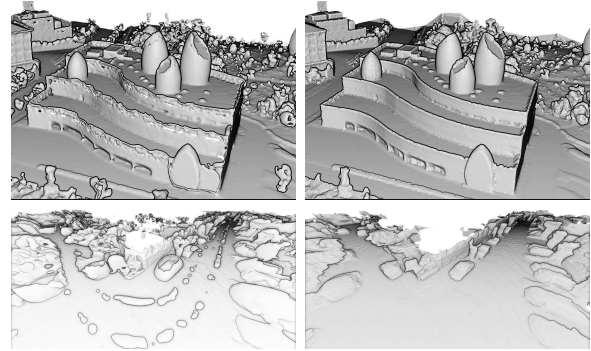
(a) SPSR   (b) Ours ($\lambda_{avw} = 1$)

Figure 12. Qualitative comparison on the large-scale outdoor dataset. The data are *Eco Park* (top) and *Crossroad* (bottom).

surface, but the completeness is poor. CONet usually produces a flat surface, which is a good feature for specific scenes (*i.e.* indoors), but it is not good for curved surfaces. While SPSR [22] also produces reasonable reconstructions, it tends to close the resulting meshes and present bubbles event after trimming with the threshold $max\_density/2$. A carefully chosen trimming parameter could reduce such artifacts. Moreover, instead of uniform sampling from the ground truth surface, we use real-world datasets derived from multi-view stereo [44] or laser scan data [14] to demonstrate the generality of the proposed method. More results can be found in the supp. material.

6. Conclusion

We presented a novel surface reconstruction framework that combines the traditional graph-cut based method and learning-based method. In our method, the neural network only focuses on a very simple task - learning to predict the visibility of the point clouds given a view which thus present a much better generalization capability over existing learning-based approaches, while maintaining the efficiency of the traditional method. Specifically, we proposed a three-step network that explicitly employs depth completion for the visibility prediction of 3D points. Furthermore, we have improved graph-cut based formulation by proposing a novel adaptive visibility weighting term. Experiments show that our proposed method can be generalized to point clouds in different scene contexts, and can be extended to point clouds in large scenes. On datasets of varying scales, our framework presents better performance than the comparing classic reconstruction methods and state-of-the-art learning-based approaches.

References

[1] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1):3–15, 2003. 2

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'77, page 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc. 6

[3] Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library, 2017. 2

[4] András Bódis-Szomorú, Hayko Riemenschneider, and Luc Van Gool. Superpixel meshes for fast edge-preserving surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2020, 2015. 5

[5] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry, chapter*, chapter Voronoi diagrams: Euclidian metric, Delaunay complexes. Cambridge University Press, 1998. 1

[6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 3

[7] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, 2001. 2

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[9] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 2

[10] CloudCompare. CloudCompare - Open Source project. Available: http://www.cloudcompare.org/. 5

[11] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 303–312, New York, NY, USA, 1996. Association for Computing Machinery. 2

[12] Herbert Edelsbrunner and Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994. 2

[13] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J. Mitra, and Michael Wimmer. Points2Surf: Learning implicit surfaces from point clouds. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 108–124, Cham, 2020. Springer International Publishing. 2, 5, 8

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5, 6, 8

[15] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. *ACM Trans. Graph.*, 39(4), 2020. 3, 5, 8

[16] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, 1992. 2

[17] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR 2011*, pages 3121–3128, 2011. 2

[18] Michal Jancosek and Tomas Pajdla. Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces. *International Scholarly Research Notices*, 2014:1–20, 08 2014. 2

[19] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. *ACM Trans. Graph.*, 26(3):24–es, July 2007. 2, 7

[20] Michael Kazhdan. Reconstruction of solid models from oriented point sets. In *Proceedings of the third Eurographics symposium on Geometry processing*, pages 73–es, 2005. 2

[21] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 1, 2

[22] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 1, 2, 5, 8

[23] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 6

[24] Ravikrishna Kolluri, Jonathan Richard Shewchuk, and James F O'Brien. Spectral surface reconstruction from noisy point clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 11–21, 2004. 2

[25] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007. 2

[26] Patrick Labatut, J-P Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. In *Computer graphics forum*, volume 28, pages 2275–2290. Wiley Online Library, 2009. 2, 3, 5, 7

[27] Yangyan Li, Qian Zheng, Andrei Sharf, Daniel Cohen-Or, Baoquan Chen, and Niloy J Mitra. 2d-3d fusion for layer

decomposition of urban facades. In *2011 International Conference on Computer Vision*, pages 882–889. IEEE, 2011. 2

[28] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*, 2018. 4

[29] Guilin Liu, Kevin J. Shih, Ting-Chun Wang, Fitsum A. Reda, Karan Sapra, Zhiding Yu, Andrew Tao, and Bryan Catanzaro. Partial convolution based padding. In *arXiv preprint arXiv:1811.11718*, 2018. 4

[30] Minghua Liu, Xiaoshuai Zhang, and Hao Su. Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In *The European Conference on Computer Vision (ECCV)*, 2020. 3, 5, 8

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[32] Yukie Nagai, Yutaka Ohtake, and Hiromasa Suzuki. Tomographic surface reconstruction from point cloud. *Computers & Graphics*, 46:55–63, 2015. 2

[33] Sven Oesau, Florent Lafarge, and Pierre Alliez. Planar shape detection and regularization in tandem. In *Computer Graphics Forum*, volume 35, pages 203–215. Wiley Online Library, 2016. 2

[34] Office of Information Technology Ohio Department of Administrative Services. Ohio geographical referenced information program. Available: https://ogrip.oit. ohio.gov/Home.aspx. 5

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[36] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 5

[37] R. Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 1999. 4

[38] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019. 7

[39] Pix4D. Professional photogrammetry and drone mapping software. Avaliable: https://www.pix4d.com/. 6

[40] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2

[42] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 7

[43] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sebastien Benitez, and Uwe Breitkopf. The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, 1(1):293–298, 2012. 5

[44] senseFly Inc. Datasets. Available: https://www. sensefly.com/education/datasets/. 5, 6, 8

[45] Shy Shalom, Ariel Shamir, Hao Zhang, and Daniel Cohen-Or. Cone carving for surface reconstruction. In *ACM SIGGRAPH Asia 2010 Papers*, SIGGRAPH ASIA '10, New York, NY, USA, 2010. Association for Computing Machinery. 2

[46] Neil Smith, Nils Moehrle, Michael Goesele, and Wolfgang Heidrich. Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark. *ACM Trans. Graph.*, 37(6), Dec. 2018. 4

[47] Shuang Song and Rongjun Qin. Optimizing mesh reconstruction and texture mapping generated from a combined side-view and over-view imagery. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 5(2), 2020. 1

[48] StachnissLab. Mai city dataset, 2021. Available: http://www.ipb.uni-bonn.de/data/ mai-city-dataset/. 5

[49] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20, 2017. 4

[50] Oliver Van Kaick, Andrea Tagliasacchi, Oana Sidi, Hao Zhang, Daniel Cohen-Or, Lior Wolf, and Ghassan Hamarneh. Prior knowledge for part correspondence. In *Computer Graphics Forum*, volume 30, pages 553–562. Wiley Online Library, 2011. 5

[51] H. Vu, P. Labatut, J. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):889–901, 2012. 2

[52] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[53] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 4, 5

[54] Rui Zeng, Yuhui Wen, Wang Zhao, and Yong-Jin Liu. View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, pages 1–21, 2020. 4

[55] Rui Zeng, Wang Zhao, and Yong-Jin Liu. Pc-nbv: A point cloud based deep network for efficient next best view planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020*. Institute of Electrical and Electronics Engineers Inc., 2020. 4

[56] Qian Zheng, Andrei Sharf, Guowei Wan, Yangyan Li, Niloy J Mitra, Daniel Cohen-Or, and Baoquan Chen. Nonlocal scan consolidation for 3d urban scenes. *ACM Trans. Graph.*, 29(4):94–1, 2010. 2

[57] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016. 5

[58] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 7

[59] Yang Zhou, Shuhan Shen, and Zhanyi Hu. Detail preserved surface reconstruction from point cloud. *Sensors*, 19(6):1278, 2019. 2