ParseMVS: Learning Primitive-aware Surface Representations for Sparse Multi-view Stereopsis

Haiyang Ying*
Department of Electronic Engineering, Tsinghua University
Tsinghua Shenzhen International Graduate School

Jinzhi Zhang*
Department of Electronic Engineering, Tsinghua University
Tsinghua Shenzhen International Graduate School

Yuzhe Chen
Department of Electronic Engineering, Tsinghua University
Tsinghua Shenzhen International Graduate School

Zheng Cao
BirenTech Research

Jing Xiao
Pingan Group

Ruqi Huang†
Tsinghua Shenzhen International Graduate School

Lu Fang†
Department of Electronic Engineering, Tsinghua University

ABSTRACT

Multi-view stereopsis (MVS) recovers 3D surfaces by finding dense photo-consistent correspondences from densely sampled images. In this paper, we tackle the challenging MVS task from sparsely sampled views (up to an order of magnitude fewer images), which is more practical and cost-efficient in applications. The major challenge comes from the significant correspondence ambiguity introduced by the severe occlusions and the highly skewed patches. On the other hand, such ambiguity can be resolved by incorporating geometric cues from the global structure. In light of this, we propose **ParseMVS**, boosting sparse MVS by learning the **P**rimitive-**A**wa**R**e **S**urface r**E**presentation. In particular, on top of being aware of global structure, our novel representation further allows for the preservation of fine details including geometry, texture, and visibility. More specifically, the whole scene is parsed into multiple geometric primitives. On each of them, the geometry is defined as the displacement along the primitives' normal directions, together with the texture and visibility along each view direction. An unsupervised neural network is trained to learn these factors by progressively increasing the photo-consistency and render-consistency among all input images. Since the surface properties are changed locally in the 2D space of each primitive, ParseMVS can preserve global primitive structures while optimizing local details, handling the 'incompleteness' and the 'inaccuracy' problems. We experimentally demonstrate that ParseMVS constantly outperforms the state-of-the-art surface reconstruction method in both completeness and the overall score under varying sampling sparsity, especially under the extreme sparse-MVS settings. Beyond that, ParseMVS also shows great potential in compression, robustness, and efficiency.

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**.

KEYWORDS

multi-view stereopsis, sparse views, primitive

ACM Reference Format:
Haiyang Ying, Jinzhi Zhang, Yuzhe Chen, Zheng Cao, Jing Xiao, Ruqi Huang, and Lu Fang. 2022. ParseMVS: Learning Primitive-aware Surface Representations for Sparse Multi-view Stereopsis . In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3503161.3547920

1 INTRODUCTION

Multi-View Stereopsis (MVS) aims to recover accurate and dense 3D surfaces from a set of 2D images with known camera parameters. The classical MVS algorithms, including patch matching [9, 10] and learning-based methods [24, 72], find dense photo-consistent correspondence among densely sampled images. However, as the observation becomes sparser, the uncertainty of the correspondence rises due to the significantly skewed matching patches and severe occlusions. Such dependence on densely sampled images hinders the utility of MVS in more challenging tasks [33, 60, 79]. In this paper, we aim to establish an MVS framework under sparse multi-view sensation from an order of magnitude fewer images.

To overcome the challenges of sparse MVS, primitive-based methods [8, 12, 43, 63, 70] introduce *global* structural prior explicitly to resolve the ambiguity in the correspondence matching. For instance, lines [63] and planes [8, 43] in global-scale 3D space are used as photo-consistency cues. More advanced priors, mixing non-planar regions [12] or polygons [70], are used to fit more complex scenes. However, the purely primitive-based methods usually suffer from low geometric accuracy in the absence of point-wise geometric modeling. On the other end of the spectrum, the recent advances of

*Both authors contributed equally to this research.
†Corresponding Author: ruqihuang@sz.tsinghua.edu.cn, fanglu@tsinghua.edu.cn, http://luvision.net.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

MM '22, October 10–14, 2022, Lisboa, Portugal

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9203-7/22/10.
https://doi.org/10.1145/3503161.3547920

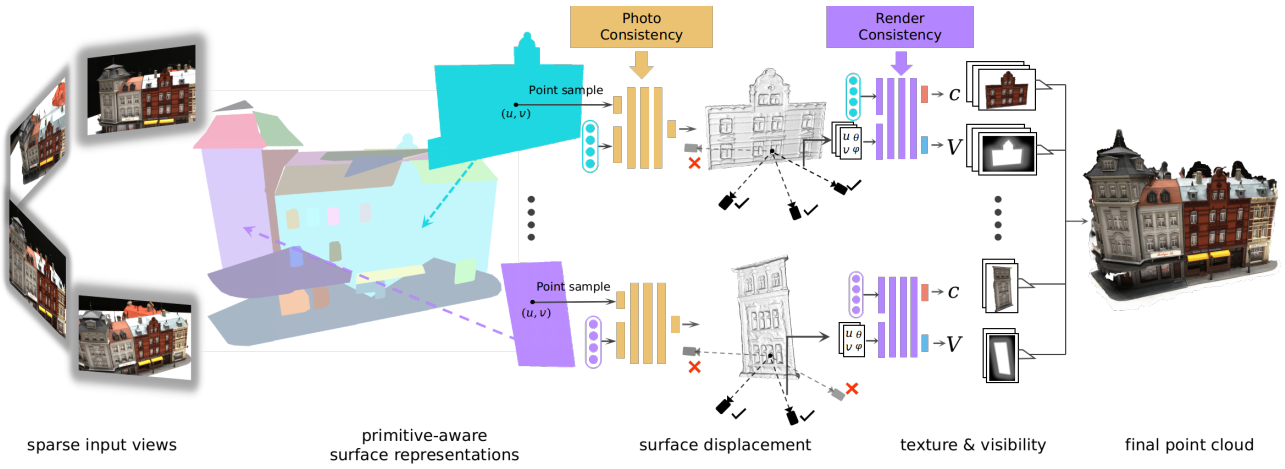

Figure 1: An overview of ParseMVS (Primitive-AwaRe Surface rEpresentation) and the learning framework for sparse multi-view stereopsis. The local point-level geometry, texture and the visibility are learned from two multi-layer perceptrons (MLP) defined in the global primitive-level spaces.

implicit representation [25, 44, 50] allow for high-quality preservation of fine *local* geometric details. For instance, SurRF [78] builds radiance fields on continuous and explicit 2D surfaces that can be gradually deformed along view-dependent camera rays by differentiable rendering. Unfortunately, the accuracy of the detailed geometric contents and the scale of the global geometric priors are severely limited by the high memory consumption of such methods. Meanwhile, the large baseline of sparse observation also introduces severe artifacts at geometric edges and occlusion areas.

Our key insight is that, under the sparse sensation, finding correspondence requires strong global primitive-level regularization while modeling the local point-level geometry. In light of this, we propose **ParseMVS**, boosting sparse MVS by learning the **P**rimitive-**A**wa**R**e **S**urface r**E**presentation, which encodes geometry, texture, and visibility in an integrated manner. Our novel representation takes advantage of the rich geometric cues from both levels, giving rise to a complete, accurate and efficient 3D surface reconstruction framework under the sparse MVS setting.

Specifically, the scene of interest is parsed into multiple primitives under our representation. The geometry of each primitive is encoded by the displacement along the primitive's normal direction, while color and visibility are defined as functions of each view direction. All these factors are differentiable and therefore can be learned by a neural network and the corresponding embeddings. The initialization of our method comes from a sparse point cloud reconstruction, primitive detection, and the segmentation of each image. Then, the detailed geometry is refined on each primitive by progressively increasing the photo-consistency among all input images. Under each view, the rendered images and the primitive segmentations are predicted as the weighted summations of texture and visibility from all primitives, using visibility values as their weights. By minimizing the rendering error, the visibility learns to identify non-occluded views, leading to more reliable photo-consistent correspondence. Different from traditional point cloud completion methods [69, 80] which basically rely on smoothness regularization and therefore suffer from low geometric accuracy, the detailed geometry and texture in ParseMVS can be recovered by progressively increasing the photo-consistency among all images.

Our proposed ParseMVS outperforms state-of-the-art MVS methods under all sparse sampling strategies in the DTU dataset. Under the extreme sparse-MVS settings, where existing methods only return very few points, ParseMVS produces far more complete surfaces with higher fidelity textures, demonstrating the effectiveness of both the global primitive-level regularization and the local geometry and texture refinement. Moreover, we show the potential ability of our ParseMVS in compression, robustness, and efficiency.

To summarize, our main contributions are as follows:

- Primitive-aware representation: We propose a novel scene representation by modeling geometry, visibility, and texture on a set of primitives, which are parameterized by MLP networks and embedding grids. Such a representation introduces both strong global structural prior and detailed local geometry of the scene.
- Sparse MVS: Based on the primitive-aware representation, we establish an unsupervised learning pipeline for the sparse MVS task, where the surface geometry, visibility, and texture are gradually optimized by enhancing photo-consistency and non-occluded view identification, guaranteeing both completeness and accuracy.

2 RELATED WORK

2.1 Multi-view Stereopsis (MVS)

Works in the multi-view stereopsis (MVS) field can be roughly categorized into three categories.

Depth map fusion algorithms[2, 9, 10, 59, 65] decouple the complex MVS problem into view-wise depth map estimation and multi-view depth map fusion[14, 22, 48, 77]. Along this line of works, 2D convolutional neural networks (2D-CNNs)[11, 17, 35, 61] are applied in various stages in MVS. However, for these methods, skewed

patches and uneven samples on the 3D surface can lead to poor photo-consistency agreements, causing incomplete fusion models.
Volumetric-based methods divide the 3D space into regular grids. They use either implicit representation[7, 38, 57, 81] or explicit surface properties[18, 23, 24, 29, 36, 67] to represent and optimize in a global framework. However, the regular voxels do not directly carry neighborhood information with respect to shapes and colors, making it hard to generate high fidelity surfaces.Currently, learning methods based on 3D cost volume regularization have been proposed. By projecting 2D images or features into 3D volume, SurfaceNet[24, 25] optimizes the 3D geometry by 3D CNN, and the similar 3D cost volume regularization methods are proposed in [5, 6, 16, 24, 25, 29, 31, 72, 73, 75]. However, these supervised methods crucially rely on ground-truth 3D data annotation.
Point-cloud-based methods operate directly on 3D points, which progressively densify the reconstruction in a propagation manner [9, 39]. Recently, Chen *et al.* [4, 5] implement the graph neural network into the propagation step, which helps achieve larger reception fields and makes the learning-based framework more parallelizable. Wang *et al.* [66] further introduce a patch-match-inspired point-wise attention among multi-view features to predict the visibility of each point. However, these supervised methods suffer from occlusion problem under sparse observation.

2.2 Neural Surface Representations

Some recent works also investigate the reconstruction of surface meshes [20, 28, 30, 37, 53, 62, 68], deformable shapes [27, 28] and implicit representations [15, 42, 52, 64] (e.g., distance functions[3, 26, 55] and occupancy fields[13, 49]). To facilitate the reliance over the difficult and tedious 3D annotation, several works [44, 47, 50, 52, 64, 74] propose to formulate differentiable rendering functions using only 2D images as supervision. However, these methods are either limited to indoor scenes with over-smoothed results estimated by the implicit surface or generate rugged and mistaken surfaces from the transmittance fields. In order to extend to large-scale reconstruction, Zhang *et al.* [78] propose the surface radiance field defined on a continuous and explicit 2D surface that can be gradually deformed along view-dependent camera rays by differentiable rendering. Similar gradual learning methods are proposed in [44, 71]. However, these methods require dense viewpoints due to the lack of global shape prior.

2.3 Primitive Learning

Primitive fitting is an important step toward holistic 3D reconstruction. There have been extensive works in computer graphics and computer-aided geometric design on primitive fitting, such as RANSAC [41, 58] and region growing [46, 54, 56]. Though aware of the global geometric structures, these methods usually suffer from low accuracy in the presence of noises or complex surface structures. Recently, learning-based algorithms have been used to detect and fit primitives on images [19] or point clouds [21, 40, 76]. However, all these methods lack clear boundary definition, texture representation, and fine-detailed identification, which are hard to generalize to image-based reconstructions.

The success of primitive detection attracts researchers in MVS. They try to introduce structural regularities such as planarity and orthogonality defined by primitives, especially in the textureless and/or highly non-Lambertian regions where photometric consistency is unreliable. Sinha *et al.* [63] compute 3D planes based on sparse point cloud and 3D lines from multiple views and predict a piecewise planar depth map for each image by graph-cuts. Liu *et al.* [43] employ a convolutional neural network to detect planes with parameters and segmentation masks, then jointly refine all the segmentation masks enforcing the cross-view consistency. To break through the limitations of the plane assumption, Gallup *et al.* [12] present hybrid modeling of planar and non-planar regions, utilizing planar region segmentation from images, capable of handling more general scenes. However, these primitive-based methods suffer from low geometric and texture accuracy due to the clumsy representation, especially in fine-detailed regions.

3 PRIMITIVE-AWARE SURFACE REPRESENTATION

In this section, we introduce our novel representation in ParseMVS, which is illustrated in Fig. 2. The scene is first parsed into multiple geometric primitives, including plane, sphere, cylinder, and cone. Instead of naively fitting the primitives onto the surface, our method recovers the scene by gradually 'embossing' the primitives with the local surface information, including geometry, texture, and visibility. The locally-changed surface properties in the 2D parametric space preserve global primitive structures while optimizing local details, which is able to handle the 'incompleteness' and the 'inaccuracy' problems. We first introduce how we encode geometry, color, and visibility in a differentiable way. Then we describe how to deal with primitive boundaries, and finally put everything together to form a complete scene.

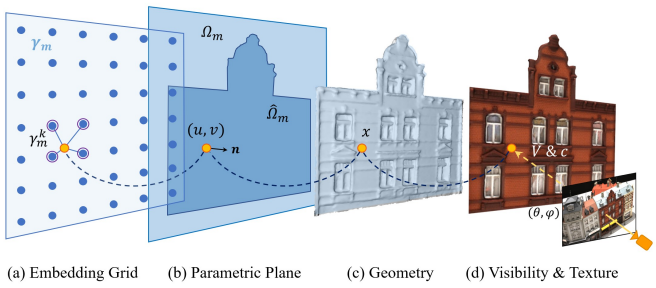

Figure 2: Illustration of our primitive-aware surface representation.

3.1 Definition

The scene is first decomposed into M geometric primitives, each encoding a basic parametric shape such as a plane, sphere, cylinder, or cone. Each primitive m admits a parametric mapping function $P_m : (u, v) \in R^2 \rightarrow R^3$ where each 2D coordinate (u, v) on the primitive is mapped to a point in 3D space. Since primitive m itself is unbounded, we set an initial boundary Ω_m as the 2D bounding

box of the projected primitive's support point set[1]. The whole scene is further represented by M primitives and a corresponding set of embedding grid $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_M\}$ encoding detailed geometry, texture, and visibility information. For each primitive m, the embedding gird γ_m is composed of k_m embeddings (with code length l) distributed evenly inside the boundary Ω_m.

Geometry. We use a multilayer perceptron network (MLP) $r_{\Theta_s} : R^2 \times R^l \rightarrow R$ to decode the pointwise surface displacement along the normal direction of the primitive. The final surface is represented by the mapping:

$$x = P_m(u, v) + r_{\Theta_s}(u, v; \gamma_m^k)\mathbf{n}, \tag{1}$$

where γ_m^k is the bilinear interpolated feature embedding on γ_m (Fig. 2(a)) and the normal is defined as:

$$\mathbf{n} = \frac{\partial_u P_m \times \partial_v P_m}{\|\partial_u P_m \times \partial_v P_m\|}. \tag{2}$$

Texture. The color of each point on the primitive is predicted by a texture MLP, whose input is the combination of the coordinate (u, v), view direction (θ, ϕ) and the corresponding embedding γ_m^k:

$$\begin{aligned} \mathbf{C}_{\Theta_c} &: R^2 \times R^2 \times R^l \rightarrow R^3 \\ &(u, v; \theta, \phi; \gamma_m^k) \rightarrow (c) \end{aligned} \tag{3}$$

Visibility. We tackle the severe occlusion under sparse view sampling by applying an additional visibility representation along each view direction. At each point on the primitive, the visibility value along each ray indicates the probability of whether the point is visible or not. This enables the ParseMVS to detect both the occlusion relations and the boundary of each primitive.

Similar to the definition of the color (Eq. 3), the visibility is predicted by an MLP with the same inputs:

$$\begin{aligned} \mathbf{V}_{\Theta_v} &: R^2 \times R^2 \times R^l \rightarrow R^1 \\ &(u, v; \theta, \phi; \gamma_m^k) \rightarrow (V), V \in [0, 1]. \end{aligned} \tag{4}$$

3.2 Surface Formation

We have defined the texture and the visibility of each primitive along each view direction, which can be used to learn the accurate surface attributes. However, both of them are defined on a radiance field, which should be fused together to generate a view-independent surface.

To generate the precise boundary of each primitive, we fuse the visibility of all views altogether. The key idea is that a point is valid if and only if it is visible from at least N_v input views. Specifically, we define the visible view set $\mathbf{B_m}(u, v)$ of each point by the collection of all input view directions that the visibility is larger than a threshold τ:

$$\mathbf{B_m}(u, v) = \left\{ (\theta_n, \phi_n) \middle| \mathbf{V}_{\Theta_v}(u, v; \theta_n, \phi_n; \gamma_m^k) \geqslant \tau, n \in N \right\}. \tag{5}$$

The final boundary of primitive m is defined as:

$$\hat{\Omega}_m = \left\{ (u, v) \middle| (u, v) \in \Omega_m, |\mathbf{B_m}(u, v)| \geqslant N_v \right\}. \tag{6}$$

[1]Each primitive has a corresponding support point set from RANSAC [58] (as will be discussed in Section 4). We project the point set onto each primitive's 2D parametric space and calculate the 2D bounding box as the initial boundary.

We assign the final color of the point with the weighted average of colors from all views in the visible view sets $\mathbf{B_m}(u, v)$:

$$c_m(u, v) = \frac{\sum_{(\theta,\phi)\in \mathbf{B_m}(u,v)} \mathbf{V}_{\Theta_v}(u, v; \theta, \phi; \gamma_m^k)\mathbf{C}_{\Theta_c}(u, v; \theta, \phi; \gamma_m^k)}{\sum_{(\theta,\phi)\in \mathbf{B_m}(u,v)} \mathbf{V}_{\Theta_v}(u, v; \theta, \phi; \gamma_m^k)}. \tag{7}$$

Therefore, the surface $\mathbf{S_m}$ on each primitive m is:

$$\mathbf{S_m} = \left\{ (x, c) \middle| \begin{aligned} x &= P_m(u, v) + r_{\Theta_s}(u, v; \gamma_m^k)\mathbf{n}, \\ c &= c_m(u, v), (u, v) \in \hat{\Omega}_m \end{aligned} \right\}, \tag{8}$$

and the final result is the union of M surfaces: $\mathbf{S} = \bigcup_M \mathbf{S}_m$.

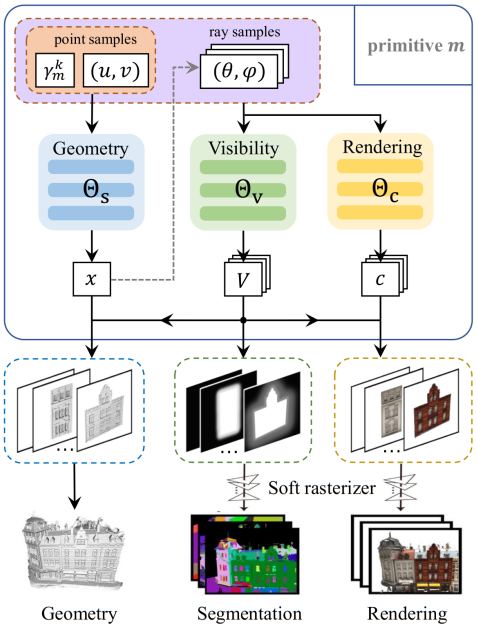

Figure 3: The architecture of ParseMVS. The geometry network predicts the surface displacement along the normal direction. The color and visibility are then estimated by two additional networks to give the segmentation and rendering results from multiple views.

4 OPTIMIZATION

In this section, we describe our overall pipeline in detail.Given a set of sparsely sampled images, we first get an incomplete point cloud with normal using PMVS [9]. Then we extract rough primitives (including geometric parameters and the corresponding support point sets) from the initial point cloud using efficient RANSAC [58].

During training, we sample points (u, v) and embeddings γ_m^k on each primitive m as the input of the geometry network Θ_s and get the displaced points x (Eq. 1), which will be optimized by enhancing photometric consistency (Sec. 4.1). For each posed image, the view direction (θ, ϕ) towards x together with (u, v) and γ_m^k are sent into network $\mathbf{\Theta_c}$ and $\mathbf{\Theta_V}$ to obtain pointwise texture c and visibility V, which will be supervised with render consistency loss via a soft

rendering operation (Sec. 4.2). The network training settings are described in Sec. 4.3.

4.1 Photometric Consistency

The goal of the multi-view stereopsis in this work is to learn a surface deformation r_{Θ_s} defined in Eq. 1, which is the optimal offset along the normal direction of the primitive point (u, v) that maximizes the photometric consistency score:

$$\Theta_s = \text{argmax}_{\Theta_s} \mathcal{NCC}(x, \hat{\mathbf{n}}), \tag{9}$$

where

$$x = P_m(u, v) + r_{\Theta_s}(u, v; \gamma_m^k)\mathbf{n}, \quad \hat{\mathbf{n}} = \frac{\partial_u x \times \partial_v x}{\|\partial_u x \times \partial_v x\|}, \tag{10}$$

and $\mathcal{NCC}(x, \hat{\mathbf{n}})$ is the photometric consistency score [9] on the tangent plane of point x with normal $\hat{\mathbf{n}}$. Given two images, the photometric consistency score is defined as the normalized cross correlation (NCC) of its projections into the two images. We only select the views in the visible view set $\mathbf{B_m}(u, v)$ (Eq. 5). Since the photometric function in Eq. 9 is difficult to optimize via gradient descent, we relax the problem by decomposing it into the photo-consistency optimization and the offset fitting. Specifically, we first use a non-linear optimization in PMVS [9] to find all points $\mathcal{P}$ that maximize the photo-consistency. Then the optimization objective comes to the squared distance between the points $\tilde{x}$ in $\mathcal{P}$ and the location of the optimal offset:

$$\Theta_s = \text{argmin}_{\Theta_s} \left\{ \sum_{(\tilde{u},\tilde{v})\in\Omega_m} \left\| \tilde{x} - \left(P_m(\tilde{u}, \tilde{v}) + r_{\Theta_s}(\tilde{u}, \tilde{v}; \gamma_m^k)\mathbf{n} \right) \right\|^2 \right\}, \tag{11}$$

where $(\tilde{u}, \tilde{v})$ is the projection of the point $\tilde{x}$ onto the primitive.

4.2 Render Consistency

To train the implicit parameters $\mathbf{\Theta_c}$ and $\mathbf{\Theta_V}$ defined in Sec. 3, a proper sampling strategy is required to compose the scene appearance from a collection of primitives. The visibility along each view direction (Sec. 3.1) provides a natural way to combine the information of each primitive. A pixel in one image may path through several different primitives[2]. Therefore, the visibility defined on each point of each primitive acts as a soft boundary (just like the soft rasterizer described in [45, 78]). Specifically, for each pixel p from one of the N images, we only select N_f visible points (marked as $D_{n,p}$), getting rid of the occluded ones. The rendered pixel color is the weighted average of these points:

$$\bar{c}_{n,p} = \frac{\sum_{i\in D_{n,p}} V_i \cdot \mathbf{C}_{\mathbf{\Theta_c}}(u_i, v_i; \theta_i, \phi_i; \gamma_i^{k_i})}{\sum_{i\in D_{n,p}} V_i}, \tag{12}$$

where V_i is the predicted visibility. The network can learn the texture and visibility by minimizing the rendering differences between $\bar{c}_{n,p}$ and the pixel color $\tilde{c}_{n,p}$. We also average the instance label of the primitives:

$$\bar{S}_{n,p} = \frac{\sum_{i\in D_{n,p}} V_i \cdot S_{k_i}}{\sum_{i\in D_{n,p}} V_i}, \tag{13}$$

[2]We apply (u_i, v_i) sampling at high density on each primitive to ensure that for each pixel, there are enough sampled points projected onto it.

where S_{k_i} is the one-hot encoding vector of the primitive label m. We use compact watershed [51], an unsupervised semantic segmentation, to obtain the primitive segmentation $\tilde{S}_{n,p}$ on each image.

The final optimization objective comes to the sum of the error:

$$(\Theta_c, \Theta_v) = \text{argmin}_{\Theta} \left\{ \sum_{n\in N,p} \left(\beta \cdot E(\bar{S}_{n,p}, \tilde{S}_{n,p}) + \|\bar{c}_{n,p} - \tilde{c}_{n,p}\|^2 \right) \right\}, \tag{14}$$

where $E(\cdot, \cdot)$ is the cross entropy.

4.3 Network Training

We train the network by randomly sampling points from primitives and rays from camera views for each scene separately. Each of the prediction function in $(r_{\Theta_s}, \mathbf{C}_{\mathbf{\Theta_c}}, \mathbf{V}_{\mathbf{\Theta_v}})$ is an MLP with 4 fully-connected layers. The positional encoding [50] is implemented on the input coordinate (u, v) and (θ, ϕ). All activation functions are set to ReLU. The length of embedding γ_m^k and the channels per layer are all set to 128. We use the Adam optimizer [32] with an initial learning rate of 1.0×10^{-4} which is decreased by 0.9 for every 100 epochs. During training, we use the downsampled images resolution (600×800) for geometry, and the original image scale (1200×1600) for texture and visibility. In our experiments, $\beta = 0.1$ in Eq. 14. To prevent the network from overfitting the noise in geometry, we stop the training r_{Θ_s} after only 100,000 iterations. All the training is conducted on a single GTX 1080Ti graphics card.

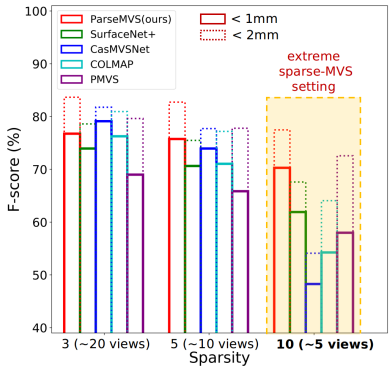

Figure 4: Comparison with the existing methods in the DTU Dataset [1] with different input sparsity. Under the extreme sparse-MVS setting (sparsity = 10), only one-tenth of the images (~5 views) are selected from the original DTU dataset.

5 EXPERIMENT

5.1 Dataset and Evaluation

We evaluate the performances of our pipeline and several baselines on the DTU dataset [1] under different sparse MVS settings. The DTU dataset is a large-scale MVS benchmark, which features a variety of objects and materials, and contains 80 different scenes seen from 49 camera positions under seven different lighting conditions. Our test set includes 22 challenging scans, with both complex primitives and detailed surfaces. The scan indexes are: 1, 5, 6, 8, 9, 10, 11, 12, 15, 17, 18, 19, 23, 24, 34, 35, 38, 39, 40, 43, 46, 59.

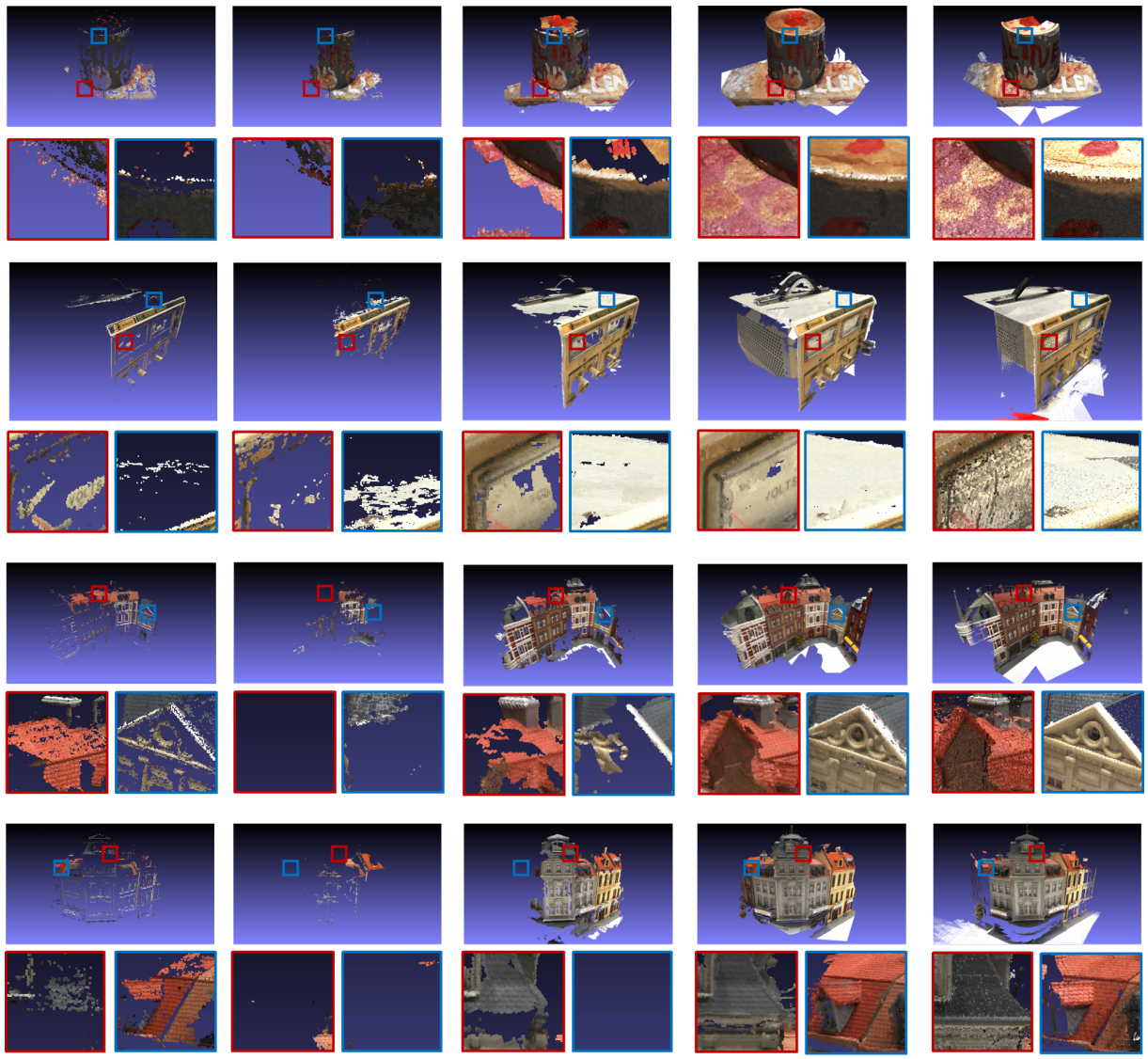

Figure 5: Qualitative results for *sparsity* = 10, i.e., from an order of magnitude fewer images. We choose scan 1, 11, 17, and 18 in the DTU dataset[1], compared with COLMAP[59], CasMVSNet[16], SurfaceNet+[25] and ground truth point cloud. Our method, ParseMVS, generates much more complete and denser point clouds with higher fidelity textures.

The evaluation metrics quantify how closely the reconstructed point cloud lie to the ground-truth laser scans. The distance metric [1] and the percentage metric [33] are used. The former is the same as the standard Chamfer distance in the DTU dataset [1], and the latter counts the percentage of points under a certain threshold [34]. The *overall* score for the distance metric is the average of the mean precision and mean recall, and the *overall* score for the percentage metric is measured as the F-score.

5.2 Benchmark

We qualify the performances on the DTU dataset [1] under different sparse MVS settings. In our experiments, the sparsity of sampled view points is defined as follows: we select a small proportion of the camera views by consecutively sampling views from every $sparsity = n$ image index, i.e., $\{1, n+1, 2n+1, \cdots\}$. Under the extreme sparse-MVS setting ($sparsity = 10$), only one-tenth of the images (~5 views) are selected from the original DTU dataset. Besides PMVS [9], We compare our method with one state-of-the-art sparse MVS algorithm: SurfaceNet+ [25], and two state-of-the-art MVS algorithm: COLMAP [59] and CasMVSNet [16] in terms

Table 1: Quantitative results of reconstruction quality on the DTU dataset in terms of the distance metric [1] (lower is better) and the percentage metric [33] (higher is better) with $1mm$ and $2mm$ as thresholds. Our method constantly outperforms state-of-the-arts in terms of recall and F-score in all the sparse-MVS settings with $n = 3, 5, 10$.

Sparsity	Method	Mean Distance(mm)			Percentage(<1mm)			Percentage(<2mm)		
		Precision	Recall	**Overall**	Precision	Recall	**F-score**	Precision	Recall	**F-score**
3 (~20 views)	**ParseMVS (ours)**	0.450	**0.589**	**0.520**	84.01	**71.65**	76.76	92.78	**77.08**	**83.64**
	SurfaceNet+[25]	0.457	0.692	0.575	86.85	71.44	77.93	93.04	75.03	82.59
	CasMVSNet[16]	0.445	1.014	0.730	**94.81**	68.96	**79.12**	**96.61**	71.94	81.76
	COLMAP[59]	**0.438**	1.125	0.782	90.49	66.79	76.25	93.25	72.45	80.94
	PMVS[9]	0.455	1.210	0.832	82.46	60.18	68.99	88.89	73.19	79.63
5 (~10 views)	**ParseMVS (ours)**	0.529	**0.634**	**0.582**	85.73	**68.66**	**75.74**	94.07	**74.65**	**82.73**
	SurfaceNet+	0.543	0.688	0.616	88.67	65.33	74.63	94.31	69.57	79.48
	CasMVSNet	**0.343**	1.423	0.883	**95.63**	61.65	73.94	**97.51**	65.95	77.71
	COLMAP	0.356	1.985	1.171	93.94	57.97	71.04	95.91	65.46	77.18
	PMVS	0.413	1.540	0.977	84.19	54.78	65.85	90.32	69.30	77.78
10 (~5 views)	**ParseMVS (ours)**	0.497	**0.754**	**0.626**	79.48	**64.11**	**70.29**	87.56	**70.64**	**77.48**
	SurfaceNet+	0.511	1.108	0.810	88.72	53.31	65.90	94.50	58.47	71.59
	CasMVSNet	0.329	2.985	1.657	96.99	34.45	48.28	**98.54**	39.79	54.10
	COLMAP	**0.315**	3.342	1.829	**97.43**	38.52	54.25	98.48	48.39	64.06
	PMVS	0.405	1.985	1.195	85.72	44.33	57.98	92.08	60.66	72.53

of *F-score* with respect to three different sparsity levels in Fig. 4. It is evident that ParseMVS constantly outperforms the baselines in all the sparse settings when the distance threshold is set to $2mm$. Especially for the extremely sparse case of $sparsity = 10$ (~5 views), our method outperforms the competing methods under all the distance thresholds by a significant margin.

In Table 1, more detailed quantitative results are presented in terms of three different metrics. ParseMVS consistently outperforms the state-of-the-art methods in both recall and the F-score under all sparsity settings. Besides, unlike other methods whose recall dramatically decay as the sparsity increases, ParseMVS has almost consistent recall quality with high precision. The reconstruction accuracy of PMVS is always lower than COLMAP and CasMVSNet. While for the recall, even though PMVS performs a little better than COLMAP and CasMVSNet when the threshold is 2mm, there still exists a big gap between PMVS and our proposed ParseMVS.

We show the qualitative comparison in the setting of $sparsity = 10$ in Fig. 5. Compared to the baselines, ParseMVS precisely reconstructs the scenes while maintaining high recall. Remarkably, ParseMVS is able to generate much more complete and denser point clouds with higher fidelity textures, indicating the correct occlusion detection and visibility prediction.

Table 2: Ablation study on the effectiveness of the surface displacement (Disp.), visibility (Vis.) and segmentation (Seg.).

Disp.	Vis.	Seg.	Precision	Recall	F-score
✓	✓	✓	82.19	68.99	**75.01**
×	✓	✓	70.15	54.05	61.06
✓	×	✓	40.11	**73.07**	51.79
✓	✓	×	**86.36**	65.00	74.17

5.3 Ablation Studies

In this section, we provide ablation studies to analyze the strengths of the key components in ParseMVS. Experiments are conducted on scan 9 in the DTU dataset, which contains complex surface topology, cross-scale primitives, and various illuminations. We train different networks and embeddings with the same initial point cloud and the primitive detection results. The qualitative and quantitative comparison are respectively shown in Fig. 6 and Table 2.

Surface displacement. We disable the surface displacement $\mathbf{r}_{\Theta_s}$ in Eq. 1. The algorithm now only predicts the texture and visibility of each primitive. As shown in Table 2, both the precision and the overall F-score drop significantly since the point cloud reconstruction only leverages global geometric structures without any local details. As shown in Fig. 6, (b) shows poor geometric accuracy in detailed regions, further demonstrating the importance of the displacement for accuracy.

Visibility. In this part, we disable visibility (Sec. 3.1) on each ray. The final boundary $\hat{\Omega}_m$ in Eq. 6 and Eq. 8 is simply replaced to the initial coarse boundary Ω_m. As shown in Table 2, the precision drops dramatically and Fig. 6 (c) also depicts that rough geometry and inconsistent texture occurred. The noisy prediction in the reconstruction comes from the unclear definition of the boundary on the primitives. This demonstrates the effectiveness of visibility for occlusion detection, view selection, and boundary generation.

Segmentation. We train our network without the segmentation loss, i.e., $\beta = 0$ in Eq. 14. Table 2 shows that the recall drops with a close precision value. Qualitative results are shown in Fig. 6 (d), where discontinuous and incomplete surfaces occur in the area with repetitive textures and reflectance surfaces. Therefore, using the primitive prior during the segmentation provides a strong regularization in the area with highly ambiguous correspondence, leading to more complete reconstructions.

5.4 Discussion

Robustness to the initialization. We perform experiments with different initial primitive detections to evaluate the robustness of ParseMVS on initialization. The results show that ParseMVS is robust to noisy initialization in a certain noise level range. Given

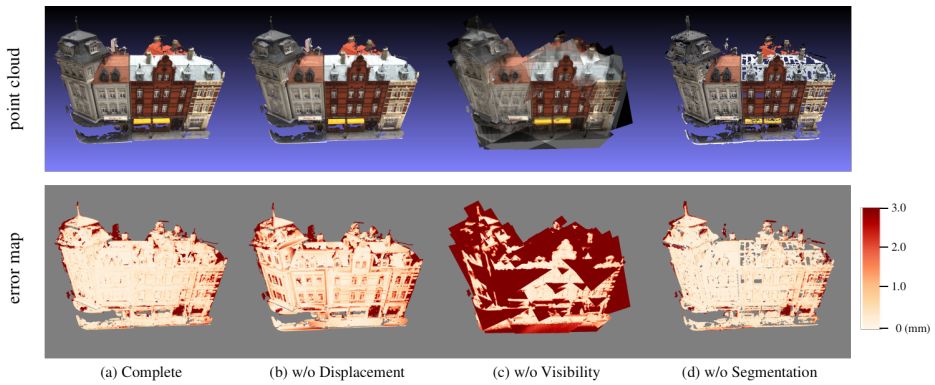

Figure 6: Ablation study on different components. Results without surface displacement (b), visibility (c) and segmentation (d) are shown respectively.

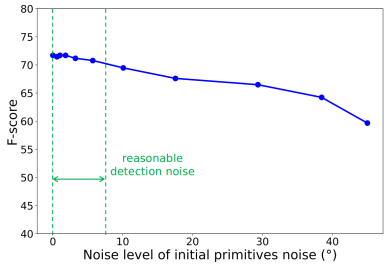

Figure 7: Performance w.r.t. noise level of different primitives parameters initialization.

the initial primitive detection results, we add Gaussian perturbation with different scales to the primitive parameters. The normal vector of each plane is further normalized to length 1. The noise level is defined as the changed angle of this random perturbation added to the original primitive's normal.

As shown in Fig. 7, when the noise level is relatively small (less than 10 degrees), the performance of the F-score remains stable. The reason is that the learned displacement field on the primitives provides a relatively high tolerance to noisy primitive parameters. When the increasing noise level becomes no longer reasonable (larger than 20 degrees), the performance of the F-score drops as expected.

Table 3: Comparison with Nerf [50], IDR [74] and SurRF [78]. The table reports the average results on 15 models used in the IDR evaluation set.

Method	Training time (h)	No. of parameters	GPU Mem. (MB) (~100,000 rays)
Nerf [50]	25.5	162,959	182,324
IDR [74]	17.3	2,915,258	42,465
SurRF [78]	10.0	369,078	10,324
ParseMVS (ours)	**1.0**	**131,072**	**2048**

Training efficiency. Since the surface properties in ParseMVS are modeled locally in the 2D space of each primitive, this representation naturally enjoys the advantage in training speed. In Table 3, we report the quantitative time and memory efficiency comparisons among the per-scene optimization methods (Nerf [50], IDR [74], and SurRF [78]). The time efficiency for training is evaluated as the total hours optimizing all views with the size of 600×800 each. As shown in Table 3, the proposed ParseMVS is over ten-time faster during training. Besides, ParseMVS shows around 10x GPU efficiency. Therefore, we demonstrate that the efficient primitive-aware representation boosts the network training.

5.5 Limitation and Future work

Since our proposed ParseMVS belongs to the per-scene optimization methods, a long time optimization is still required for each scene. One further direction is to take advantage of the great amount of data to understand scenes on 2D images. That is, learning the segmentation and parsing from 2D multi-view images, which may contain part of the primitive and geometry information, can further speed up the reconstruction and improve accuracy.

6 CONCLUSION

In this paper, we tackle the challenging sparse MVS task by learning the **P**rimitive-**A**wa**R**e **S**urface r**E**presentation (ParseMVS), which models attributes of the scene based on global primitive parsing and local optimization. Neural networks are trained to learn local surface geometry, together with the texture and visibility along each view direction. We found our method produces state-of-the-art surface geometry and texture with high completeness and fidelity under the sparse-MVS settings. Beyond that, ParseMVS also shows great potential in scene compression and efficient optimization.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under contract No. 62125106, No. 61860206003, No. 62088102 and No. 62171256, in part by Ministry of Science and Technology of China under contract No. 2021ZD0109901, in part by Shenzhen Key Laboratory of next generation interactive media innovative technology (No. ZDSYS20210623092001004), in part by Shanghai Biren Technology Co., Ltd. through the Collaboration on GPGPU Innovation Research between Tsinghua University and Shanghai Biren Technology Co., Ltd.

REFERENCES

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. 2016. Large-Scale Data for Multiple-View Stereopsis. *International Journal of Computer Vision* (2016), 1–16.

[2] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. 2008. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision*. 766–779.

[3] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. 2020. Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction. *arXiv preprint arXiv:2003.10983* (2020).

[4] Rui Chen, Songfang Han, Jing Xu, et al. 2020. Visibility-Aware Point-Based Multi-View Stereo Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[5] Rui Chen, Songfang Han, Jing Xu, and Hao Su. 2019. Point-Based Multi-View Stereo Network. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[6] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. 2020. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2524–2534.

[7] Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 303–312.

[8] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. 2009. Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1422–1429.

[9] Yasutaka Furukawa and Jean Ponce. 2010. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 8 (2010), 1362–1376.

[10] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. 2015. Massively parallel multiview stereopsis by surface normal diffusion. In *IEEE International Conference on Computer Vision*. 873–881.

[11] S. Galliani and K. Schindler. 2016. Just Look at the Image: Viewpoint-Specific Surface Normal Prediction for Improved Multi-View Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5479–5487.

[12] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. 2010. Piecewise planar and non-planar stereo for urban scene reconstruction. In *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 1418–1425.

[13] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. 2020. Local Deep Implicit Functions for 3D Shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4857–4866.

[14] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven Seitz. 2007. Multi-View Stereo for Community Photo Collections. 1–8. https://doi.org/10.1109/ICCV.2007.4408933

[15] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 216–224.

[16] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. 2019. Cascade Cost Volume for High-Resolution Multi-View Stereo and Stereo Matching. *arXiv preprint arXiv:1912.06378* (2019).

[17] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander. C. Berg. 2015. MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching. In *CVPR*.

[18] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. 2020. ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1018–1027.

[19] Qian He, Desen Zhou, Bo Wan, and Xuming He. 2021. Single Image 3D Object Estimation with Primitive Graph Networks. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2353–2361.

[20] Paul Henderson and Vittorio Ferrari. 2019. Learning single-image 3D reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision* (2019), 1–20.

[21] Jingwei Huang, Yanfeng Zhang, and Mingwei Sun. 2021. PrimitiveNet: Primitive Instance Segmentation with Local Primitive Embedding under Adversarial Metric. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15343–15353.

[22] Michal Jancosek and Tomas Pajdla. 2011. Multi-View Reconstruction Preserving Weakly-Supported Surfaces. 3121 – 3128. https://doi.org/10.1109/CVPR.2011.5995693

[23] Michal Jancosek and Tomás Pajdla. 2011. Multi-view reconstruction preserving weakly-supported surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3121–3128.

[24] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. 2017. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*. 2307–2315.

[25] Mengqi Ji, Jinzhi Zhang, Qionghai Dai, and Lu Fang. 2020. SurfaceNet+: An End-to-end 3D Neural Network for Very Sparse Multi-view Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[26] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. 2020. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6001–6010.

[27] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.

[28] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. 2018. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 371–386.

[29] Abhishek Kar, Christian Häne, and Jitendra Malik. 2017. Learning a multi-view stereo machine. *Advances in neural information processing systems* 30 (2017).

[30] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3907–3916.

[31] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. 2017. End-to-End Learning of Geometry and Context for Deep Stereo Regression. *CoRR* abs/1703.04309 (2017).

[32] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[33] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics* 36, 4 (2017).

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

[35] Patrick Knöbelreiter, Christian Reinbacher, Alexander Shekhovtsov, and Thomas Pock. 2017. End-to-End Training of Hybrid CNN-CRF Models for Stereo. In *2017 Computer Vision and Pattern Recognition (CVPR)*.

[36] K. N. Kutulakos and S. M. Seitz. 1999. A theory of shape by space carving. In *IEEE International Conference on Computer Vision*, Vol. 1. 307–314.

[37] Lubor Ladicky, Olivier Saurer, SoHyeon Jeong, Fabio Maninchedda, and Marc Pollefeys. 2017. From point clouds to mesh using regression. In *Proceedings of the IEEE International Conference on Computer Vision*. 3893–3902.

[38] Victor Lempitsky and Yuri Boykov. 2007. Global optimization for shape fitting. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

[39] M. Lhuillier and L. Quan. 2005. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 3 (March 2005), 418–433. https://doi.org/10.1109/TPAMI.2005.44

[40] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. 2019. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2652–2660.

[41] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. 2011. Globfit: Consistently fitting primitives by discovering global relations. In *ACM SIGGRAPH 2011 papers*. 1–12.

[42] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. 2019. Photometric Mesh Optimization for Video-Aligned 3D Object Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[43] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. 2019. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4450–4459.

[44] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems* 33 (2020).

[45] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. 2019. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*. 7708–7717.

[46] David Marshall, Gabor Lukacs, and Ralph Martin. 2001. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Transactions on pattern analysis and machine intelligence* 23, 3 (2001), 304–314.

[47] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2020. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. *arXiv preprint arXiv:2008.02268* (2020).

[48] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. 2007. Real-Time Visibility-Based Fusion of Depth Maps. In *IEEE International Conference on Computer Vision*. 1–8.

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

[50] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance

fields for view synthesis. *arXiv preprint arXiv:2003.08934* (2020).
[51] Peer Neubert and Peter Protzel. 2014. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In *2014 22nd international conference on pattern recognition*. IEEE, 996–1001.
[52] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3504–3515.
[53] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. 2019. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE International Conference on Computer Vision*. 4531–4540.
[54] Sven Oesau, Florent Lafarge, and Pierre Alliez. 2016. Planar shape detection and regularization in tandem. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 203–215.
[55] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 165–174.
[56] Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann. 2006. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences* 36, 5 (2006), 248–253.
[57] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. 2017. Octnetfusion: Learning depth fusion from data. In *2017 International Conference on 3D Vision (3DV)*. IEEE, 57–66.
[58] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. 2007. Efficient RANSAC for point-cloud shape detection. In *Computer graphics forum*, Vol. 26. Wiley Online Library, 214–226.
[59] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*.
[60] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. 2017. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3260–3269.
[61] Akihito Seki and Marc Pollefeys. 2017. SGM-Nets: Semi-Global Matching With Neural Networks. In *CVPR*.
[62] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. 2017. Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6040–6049.
[63] Sudipta Sinha, Drew Steedly, and Rick Szeliski. 2009. Piecewise planar stereo for image-based rendering. In *2009 International Conference on Computer Vision*. 1881–1888.
[64] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*. 1121–1132.
[65] Engin Tola, Christoph Strecha, and Pascal Fua. 2012. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications* (2012), 1–18.
[66] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. 2021. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14194–14203.
[67] Meng Wang, Lingjing Wang, and Yi Fang. 2017. 3densinet: A robust neural network architecture towards 3d volumetric object prediction from 2d image. In *Proceedings of the 25th ACM international conference on Multimedia*. 961–969.
[68] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 52–67.
[69] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. 2021. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5499–5509.
[70] Qingshan Xu and Wenbing Tao. 2020. Planar Prior Assisted PatchMatch Multi-View Stereo. In *AAAI*.
[71] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. 2022. Point-NeRF: Point-based Neural Radiance Fields. *arXiv preprint arXiv:2201.08845* (2022).
[72] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. 2018. MVSnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 767–783.
[73] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. 2019. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5525–5534.
[74] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. 2020. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *Advances in Neural Information Processing Systems* 33 (2020).
[75] Hongwei Yi, Zizhuang Wei, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. 2019. Pyramid Multi-view Stereo Net with Self-adaptive View Aggregation. *arxiv preprint arXiv:1912.03001* (2019).
[76] Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. 2021. CAPRI-Net: Learning Compact CAD Shapes with Adaptive Primitive Assembly. *arXiv preprint arXiv:2104.05652* (2021).
[77] Christopher Zach. 2008. Fast and High Quality Fusion of Depth Maps. (01 2008).
[78] Jinzhi Zhang, Mengqi Ji, Guangyu Wang, Xue Zhiwei, Shengjin Wang, and Lu Fang. 2021. SurRF: Unsupervised Multi-view Stereopsis by Learning Surface Radiance Field. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).
[79] Jianing Zhang, Jinzhi Zhang, Shi Mao, Mengqi Ji, Guangyu Wang, Zequn Chen, Tian Zhang, Xiaoyun Yuan, Qionghai Dai, and Lu Fang. 2021. GigaMVS: A Benchmark for Ultra-large-scale Gigapixel-level 3D Reconstruction. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 01 (2021), 1–1.
[80] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. 2021. View-guided point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15890–15899.
[81] Fan Zhu, Li Liu, Jin Xie, Fumin Shen, Ling Shao, and Yi Fang. 2018. Learning to synthesize 3d indoor scenes from monocular images. In *Proceedings of the 26th ACM international conference on Multimedia*. 501–509.

A RESEARCH METHODS

A.1 Details of Primitive Definition

Four typical types of primitive are used in our work: plane, sphere, cylinder, and cone. We take the same mathematical formulation as defined in [58]. A plane is represented by a normal vector $\mathbf{n_p} \in R^3$ and the distance $d_p \in R^1$ from origin point to the plane. Each point x on plane $(\mathbf{n_p}, d_p)$ satisfies the following equation:

$$\mathbf{n_p} \cdot x + d_p = 0 \tag{15}$$

A sphere is defined as (c_p, r_p), where $c_p \in R^3$ is the center point and $r_p \in R^1$ is the radius.

$$||x - c_p||^2 = r_p^2 \tag{16}$$

A cylinder $(c_p, \mathbf{n_p}, r_p)$ is represented by a reference center point $c_p \in R^3$, a central axis $\mathbf{n_p} \in R^3$, and a radius $r_p \in R^1$.

$$r_p^2 + [(x - c_p) \cdot \mathbf{n_p}]^2 = ||x - c_p||^2 \tag{17}$$

A cone $(c_p, \mathbf{n_p}, \alpha_p)$ is represented by its apex $c_p \in R^3$, central axis $\mathbf{n_p} \in R^3$, and an opening angle $\alpha_p \in (0, \pi/2)$.

$$\left| \frac{x - c_p}{||x - c_p||} \cdot \mathbf{n_p} \right| = \cos \alpha_p \tag{18}$$

A.2 Details of Optimization

To train the network for a specific scene, an appropriate training strategy should be adopted since each scene may contain many primitives of different scales. In light that each primitive has a set of supportive points, we assume that the more supportive points a primitive has, the richer the geometric details the primitive carries, and the more training epochs should be assigned to the primitive. More specifically, the number of epochs during training is distributed proportionally to the number of supportive points the primitive includes.

This strategy can reasonably allocate the computational resources to primitives of different scales and speed up the convergence of the optimization process for each scene.

B EXPERIMENTS

B.1 Comparison with initial point cloud

The initial point clouds generated by PMVS[9] are shown in Fig. 8 and compared with the results of our method. We use the same parameters in PMVS for all initial point cloud generation (i.e., $csize = 2, level = 1, ncc\_threshold = 0.7, iteration = 4, minImagenum = 2$). As shown in Fig. 8, the initial point cloud is much sparser and less complete due to sparse observation and large illumination variances. However, the initial point cloud can offer valuable structural cues (e.g., edges and corner points) for primitive detection and local displacement measurement, which are utilized to optimize the scene representation by networks. Quantitative comparison in Table 1 also depicts the huge improvement in completeness with the help of our primitive-based scene representation and optimization.

B.2 Scene Compression

In this section, we apply different compression ratios and measure corresponding performances on geometry. Scan 9 of the DTU dataset is used to conduct this experiment. For primitive grid generation, we place the embeddings evenly on parameterized primitive space with resolution $R_{prim} = 5mm$ as the baseline (equal to the resolution used in the benchmark part of the main text). The resolution of point cloud sampling is set to $R_{point} = 0.2mm$ on each primitive's parametric space. In the point cloud, each point contains 3 doubles for coordinate (x, y, z) and 3 unsigned chars for RGB color, so the total bit number is $B_{point} = 216$. Each embedding vector in embedding gird contains 32 doubles ($B_{prim} = 2048$). Note that in this section, both MLP networks have 4 layers with the hidden size of 32. The compression ratio C_{ratio} is defined as the ratio of the bit

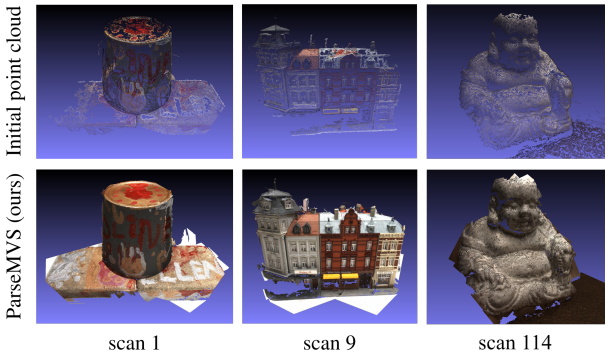

Figure 8: Comparison between the initial point clouds generated by PMVS [9] and results of ParseMVS for $sparsity = 10$ (~ 5 views). The results of our algorithm are far more complete and denser than initial point clouds.

Figure 9: The point cloud reconstructed with our primitive-based reconstruction under different compression ratios.

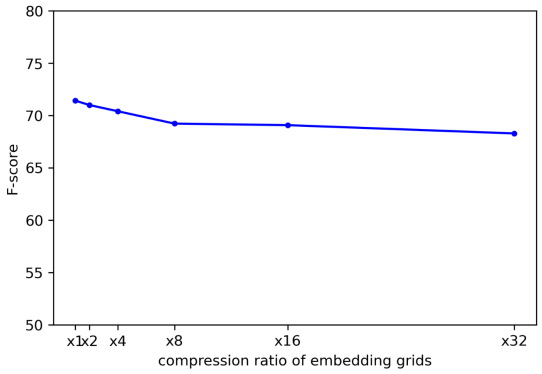

Figure 10: The quantitative comparison of different compression ratios of the primitive-based representation.

(a) COLMAP (b) CasMVSNet (c) SurfaceNet+ (d) **ParseMVS (ours)** (e) Ground Truth

Figure 11: More qualitative results for $sparsity = 10$ (~ 5 views). We choose scan 9, 15, 19, 24 in the DTU dataset[1], compared with COLMAP[59], CasMVSNet[16], SurfaceNet+[25], and ground truth point cloud. Our method, ParseMVS, generates much more complete and denser point clouds with higher fidelity textures.

number consumed to store the same scene using point cloud to the bit number used by our primitive-based representation.

$$C_{ratio} = (\frac{1}{R_{point}^2} * B_{point})/(\frac{1}{R_{prim}^2} * B_{prim}) \tag{19}$$

The compression ratio of the baseline setting (x1) is $C_{ratio} = 66$ (where $R_{prim} = 5mm$). The ratio is changed later by adjusting the resolution of embedding sampling step $R_{prim} = \{5, 10, 20, 40, 80, 160\}$ mm on the parametric plane.

As shown in Fig. 10, where the F-score threshold is 1mm, the quantitative performance of ParseMVS does not drop significantly for compression ratios ranging from x2 to x8 and even x32. There is also no apparent qualitative difference under these compression ratios according to the results in Fig. 9, which means our embedding grid-based primitive representation has great potential in scene compression and allows a relatively high compression ratio.

B.3 More Benchmark Results

More qualitative results of the DTU dataset are shown in Fig. 11 under the setting of $sparsity = 10$ (~5 views).