Self-supervised Learning of Hybrid Part-aware 3D Representations of 2D Gaussians and Superquadrics

Supplementary Material

At first, this supplementary material provides detailed experimental settings, including data processing procedures, implementation details for comparison baseline methods, and evaluation metrics (Sec.1). Subsequently, we provide further implementation specifics, including the training configurations and rendering process (Sec.3).

In Fig. 1 and Fig. 2, we provide additional visual comparisons of our method against state-of-the-art baselines on DTU [3] and ShapeNet [1] datasets. Moreover, we include visual results on several scenes from the BlendedMVS [16] and self-captured data in Fig. 3. It can be observed that the decomposed reconstructions by our method are more reasonable and precise, which is believed to be beneficial for downstream tasks. Our code and data are available at https://github.com/zhirui-gao/PartGS.

1. More Details on Experiments

1.1. Datasets

We conduct evaluations on two public datasets: DTU [3] and ShapeNet [1]. DTU is a multi-view stereo (MVS) dataset comprising 80 forward-facing scenes, each captured with 49 to 64 images. In experiments, we use 15 publicly recognized scenes commonly adopted in previous studies [2, 7]. We downsample the images to a resolution of 400×300 for computational efficiency.

To validate the effectiveness of our method in the decomposition of man-made objects, we construct a subset of the ShapeNet dataset comprising four categories: *Chair*, *Table*, *Gun*, and *Airplane*. Each category includes 15 distinct objects, providing diverse instances. For each object, we randomly sample 100 camera poses on a sphere and render images at a resolution of 400×400. The rendered images are equally split into training and testing sets.

Additionally, to explore the potential of PartGS in handling real-life data, we present qualitative results on the BlendedMVS dataset [16] and self-captured scenes. For BlendedMVS, we use official camera poses, while for self-captured scenes, camera poses are estimated by COLMAP [12] and normalized using IDR [17]. All images are resized to 400×300 pixels. For the DTU, ShapeNet, and BlendedMVS datasets, we utilize the ground-truth foreground masks provided within the datasets. For self-captured real-world scenes, we employ the Segment Anything Model (SAM) [6] to segment foreground objects.

1.2. Implementation Details on Baselines

We compare our method with four state-of-the-art works on 3D shape decomposition: EMS [8], MonteboxFinder (MBF) [11], PartNeRF [13] and DBW [9]. MBF and EMS are applied to point clouds, utilizing cuboids and superquadrics, respectively, to fit the 3D points. The input point clouds are sampled from either ground truth 3D shapes (GT) or meshes reconstructed by state-of-the-art MVS method Neus [14]. We adhere to the testing procedure outlined by DBW [9] and randomly sample 5K and 200K points from the GT meshes as the input for EMS and MBF, respectively. Similar to our problem setting, PartNeRF and DBW use images as input to construct structural 3D representations. To ensure fair comparisons, we discard the pre-defined ground plane in DBW, as the test scenario consists exclusively of foreground objects. For PartNeRF, we perform instance-specific training and set the same number of parts as in our method, while retaining all other default parameter settings. Additionally, we compare with SOTA Gaussian Splatting reconstruction method 2DGS [2], and surface reconstruction approach Neuralangelo [7], to provide an intuitive evaluation of the reconstruction quality achieved by our method.

1.3. Evaluation Metrics

We evaluate from 3D reconstruction quality, view synthesis, shape parsimony, and reconstruction time.

- The 3D reconstruction quality is measured by the official Chamfer distance (CD) evaluation [3] between the recovered geometry and GT, reflecting the accuracy of 3D reconstructions.
- View synthesis uses three standard metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [15], and Learned Perceptual Image Patch Similarity (LPIPS) [19].
- Shape parsimony is quantified by the number of parts, while reconstruction time refers to the running time measured on the same device. While shape parsimony does not directly indicate the quality of the decomposition, a smaller number of decomposed components, given consistent reconstruction quality, suggests a more concise and reasonable decomposition. This is because fitting different parts of an object with fewer blocks is inherently more challenging, whereas utilizing more blocks simplifies the fitting process.

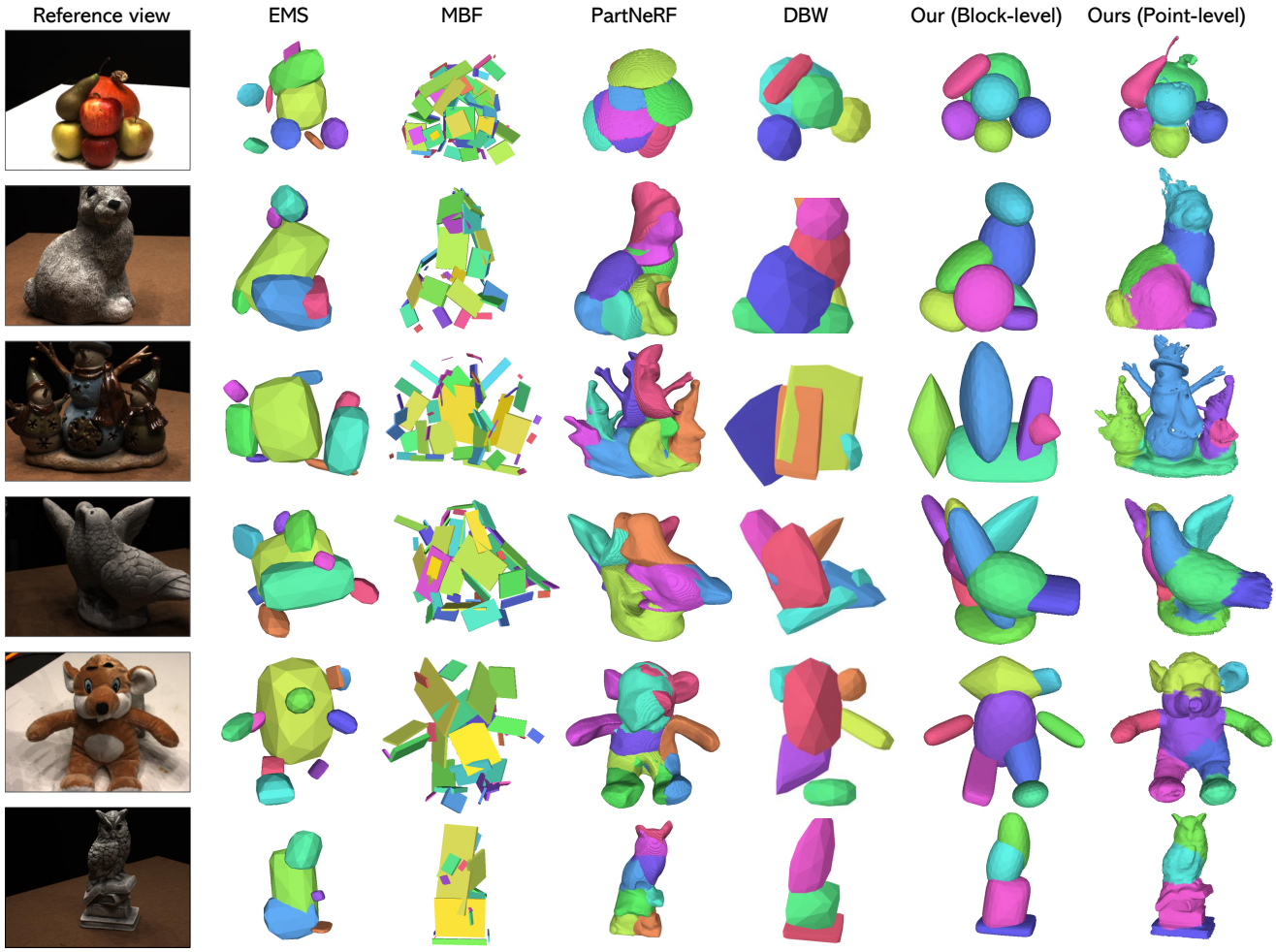

Figure 1. **Qualitative comparisons on DTU [3].** We compare our approach with state-of-the-art baselines on the DTU dataset with the background removed. The last two columns show our block-level and point-level reconstructions, respectively. Our method is the only one that provides reasonable 3D part decomposition while capturing detailed geometry.

2. Additional Comparative Experiments

Very recently, DPA-Net [18] and GaussianBlock [4] have achieved advanced 3D shape abstraction. DPA-Net enables part-aware reconstruction in a feedforward manner, requiring approximately 3 days of GPU training on a pre-collected dataset and 2 hours per object for inference. GaussianBlock utilizes SAM [6] to guide superquadric splitting and fusion for 3D decomposition, with a processing time of 6 hours per object. We compare our method on DTU and BlendedMVS scenes, with results presented in Fig. 4. The proposed approach achieves comparable performance while demonstrating superior efficiency. It is important to clarify the supervision requirements: DPA-Net is a supervised method that relies on SAM-generated segmentation masks for training, whereas GaussianBlock, while not strictly 3D supervised, depends on high-quality pre-trained datasets that typically require days of training. Our method distinguishes itself by being completely self-supervised, requiring neither 3D supervision nor pre-trained components, yet achieving significantly better computational efficiency.

3. Implementation Details

3.1. Training Configurations

To ensure uniform distributions of 2D Gaussians across surfaces of the mesh, random barycentric coordinates are generated directly within each triangular face. Specifically, barycentric weights are computed as $u = \sqrt{rand}$ and $v = rand$. These weights are transformed to obtain $\alpha = [1-u, u(1-v), uv]$, $rand \in (0,1)$. The sampled position on triangular face is calculated as $o = \alpha_0 v0 + \alpha_1 v1 + \alpha_2 v2$, where v_o, v_1, v_2 are vertices of the triangle. This method compensates for the non-uniformity caused by the triangle's

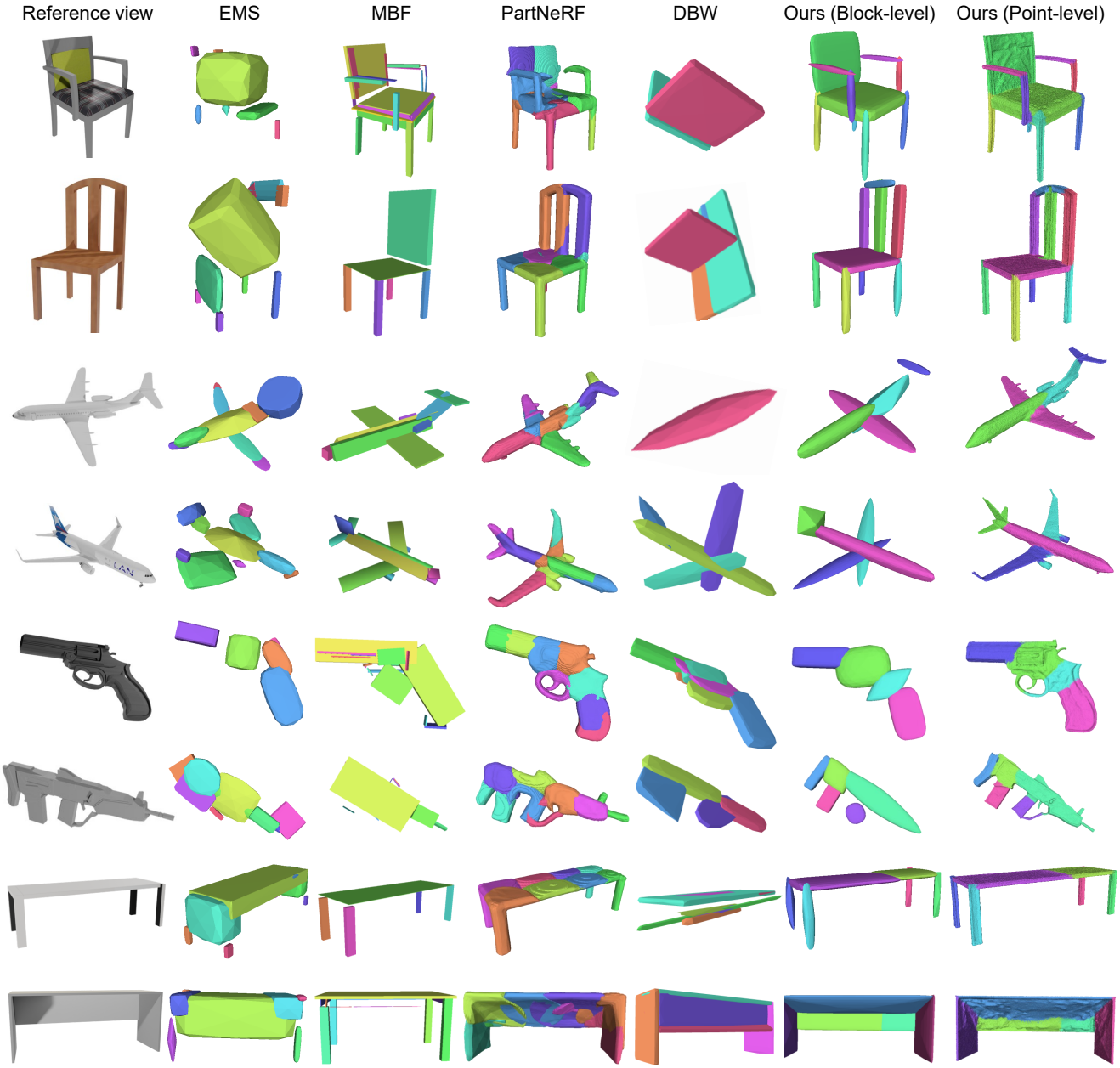

Figure 2. **Qualitative comparisons on ShapeNet [3].** We compared our approach with state-of-the-art baselines across four categories. The last two columns display our block-level and point-level reconstructions, respectively. Our method uniquely provides reasonable 3D part decomposition while simultaneously capturing detailed geometry.

geometry. Unlike naive random sampling which results in uneven distributions, it ensures that the sampled points are evenly distributed across mesh surfaces.

The same hyperparameters are used for all experiments. We set the initial number of primitives M to 8. In the hybrid representation, each superquadric mesh is a level-2 icosphere (320 triangular face). Each triangular face contains 100 Gaussians, with a scaling parameter c of 0.1. The number of sampled points in each ray is 2048.

During refinement, we employ regularization in 2DGS to achieve better geometric reconstruction, including depth distortion maps, depth maps, and normal maps. Additionally, we introduce a mask cross-entropy to filter out extra noise Gaussians. To extract the meshes from 2D Gaussians, we use truncated signed distance fusion (TSDF) to fuse rendered depth maps, utilizing Open3D [20].

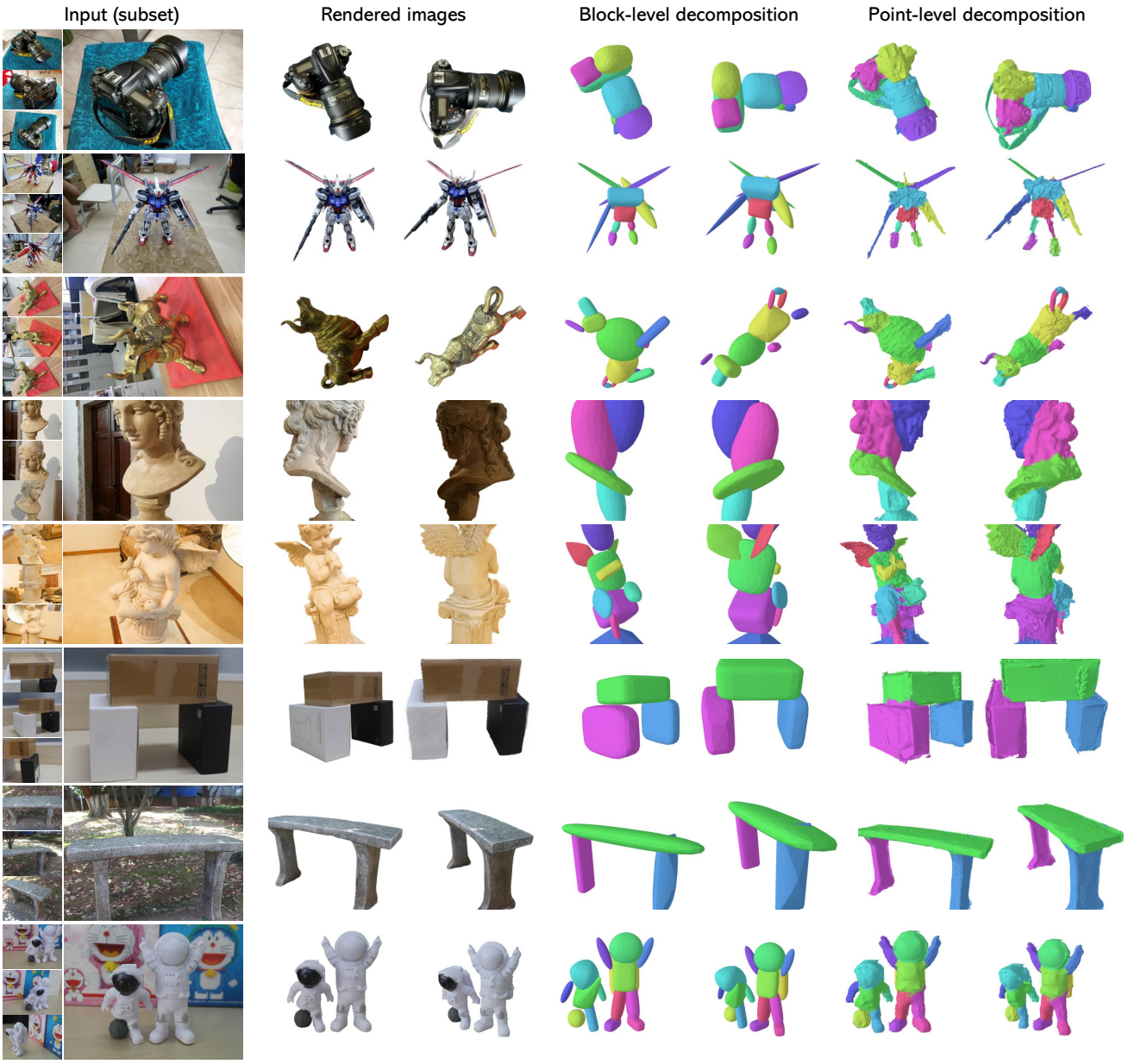

Figure 3. **Qualitative results on BlendedMVS [16] and self-capatured data.** We demonstrate the RGB renderings and decomposed parts from novel views across a variety of objects. The first five examples are from the BlendedMVS dataset, and the remaining examples are from our own captured scenes.

3.2. Rendering

With Gaussians attached to the surface of each block, we achieve view-dependent rendering through tile-based rasterization as Gaussian Splatting [2, 5]. Given a view, Gaussians on the superquadric surface are projected onto the image space, forming an RGB image. Initially, the screen space determines the bounding box for each Gaussian. Subsequently, these Gaussian ellipses are sorted according to their depths of center to the image plane. Finally, volumetric alpha compositing [10] is utilized to integrate the alpha-weighted RGB values for each pixel.

To formulate the process, considering a viewing transformation W, the covariance matrix $\Sigma_i^{'}$ of i-th Gaussian in the camera coordinate system is calculated by:

$$\Sigma_i^{'} = JW\Sigma W^T J^T, \tag{1}$$

where J is the Jacobian of the affine approximation of the projective transformation, and Σ is the covariance matrix

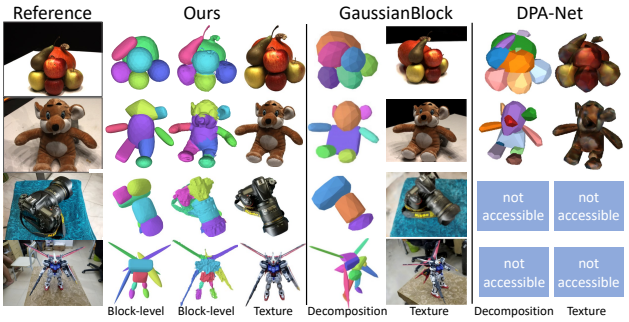

Figure 4. **Qualitative comparisons to DPA-Net and GaussianBlock.** The first two examples are from the DTU dataset, and the last two examples are from the BlendedMVS dataset.

of the Gaussian ellipse. Note that the last row and column of Σ are omitted since we adopt 2D Gaussians. Following alpha compositing, we first calculate an alpha value for each Gaussian ellipse:

$$\alpha_i = \tau_i \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{u}_i)^T \Sigma'^{-1}_{i-1}(\mathbf{x}-\mathbf{u}_i))). \quad (2)$$

Here, $\mathbf{u}_i$ is the center coordinate of the projected Gaussian ellipse, and τ_i is the opacity of the block where the i-th Gaussian is located. The calculated alphas are sorted according to their depths from the image plane. Meanwhile, we can acquire the color value c_i from the spherical harmonics of $\mathcal{N}$ ordered points, thus obtaining the rendering RGB value:

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_i). \quad (3)$$

This process is differentiable and can optimize the hybrid representation through gradient descent.

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv:1512.03012 [cs.CV]*, 2015. 1

[2] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. *SIGGRAPH*, 2024. 1, 4

[3] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanaes. Large Scale Multi-view Stereopsis Evaluation. In *CVPR*, 2014. 1, 2, 3

[4] Shuyi Jiang, Qihao Zhao, Hossein Rahmani, De Wen Soh, Jun Liu, and Na Zhao. Gaussianblock: Building part-aware compositional and editable 3d scene by primitives and gaussians. *arXiv preprint arXiv:2410.01535*, 2024. 2

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 4

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 1, 2

[7] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 1

[8] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and Accurate Superquadric Recovery: a Probabilistic Approach. In *CVPR*, 2022. 1

[9] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei A. Efros, and Mathieu Aubry. Differentiable Blocks World: Qualitative 3D Decomposition by Rendering Primitives. In *NeurIPS*, 2023. 1

[10] Thomas Porter and Tom Duff. Compositing Digital Images. In *SIGGRAPH*, 1984. 4

[11] Michaël Ramamonjisoa, Sinisa Stekovic, and Vincent Lepetit. MonteBoxFinder: Detecting and Filtering Primitives to Fit a Noisy Point Cloud. In *ECCV*, 2022. 1

[12] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[13] Konstantinos Tertikas, Despoina Paschalidou, Boxiao Pan, Jeong Joon Park, Mikaela Angelina Uy, Ioannis Emiris, Yannis Avrithis, and Leonidas Guibas. PartNeRF: Generating Part-Aware Editable 3D Shapes without 3D Supervision. In *CVPR*, 2023. 1

[14] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *NeurIPS*, 2021. 1

[15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 1

[16] 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. In *CVPR*, 2020. 1, 4

[17] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *NeurIPS*, 2020. 1

[18] Fenggen Yu, Yimin Qian, Xu Zhang, Francisca Gil-Ureta, Brian Jackson, Eric Bennett, and Hao Zhang. Dpa-net: Structured 3d abstraction from sparse views via differentiable primitive assembly. *arXiv preprint arXiv:2404.00875*, 2024. 2

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1

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