AutoRecon: Automated 3D Object Discovery and Reconstruction
Supplemental Material

Yuang Wang Xingyi He Sida Peng Haotong Lin Hujun Bao Xiaowei Zhou[†]
State Key Lab of CAD&CG, Zhejiang University

A. Unsupervised Point Cloud Segmentation and Detection Pipeline

Given an SfM point cloud with point-wisely aggregated DINO [2] features, we segment the foreground object and generate its compact 3D bounding box with the following pipeline based on Normalized Cut (NCut).

A.1. Preprocessing

The SfM point clouds built with LoFTR [12] matches are dense but noisy and usually contain too many points for NCut to process. Thus, we preprocess the SfM point clouds to filter out noisy 3D points and simplify the point clouds.

First, we filter out points with too short SfM feature tracks or large reprojection errors. We found that most of the SfM track lengths follow Gaussian distributions and the reprojection errors follow Pareto distributions, thus we set the minimum track length as the mean track length plus 3 standard deviations, and set the maximum reprojection error as the median. Then, we apply statistical outlier removal upon the point cloud. The global statistic of local distances is computed with 30 nearest neighbors, and the maximum distance is set to the mean plus 0.5 standard deviations.

To simplify the point cloud, we run voxel-based downsampling with a resolution of 128. If the videos are object-centric, we can optionally filter out 3D points lying behind more than 10% of all camera planes, since such points are mostly background points. We do not filter 3D points based on camera frustums since objects are only partially visible in most of the frames. The foreground object usually only occupies a small region of the entire scene covered by an SfM point cloud; the ratio between the number of foreground and background points is very small and NCut might fail to find a balanced bipartition in such an extreme case. To increase the robustness, we segment the ground plane from an SfM point cloud with RANSAC-based plane fitting. All 3D planes are recursively segmented from the SfM point cloud, and we take the largest plane vertical to the gravity direction as the ground plane. The segmented ground plane is rejected if there are non-negligible numbers of points distributed on both sides of it. If a ground plane is segmented successfully, we downsample ground plane points with farthest point sampling (FPS) such that the numbers of ground points and non-ground points are similar, which facilitates NCut solving. Finally, to limit the time and memory cost of solving NCut, we downsample the preprocessed point cloud with FPS to keep a maximum of 10,000 points.

A.2. Graph construction and cue combination

Applying NCut on the DINO point cloud facilitates global reasoning in 3D for salient object decomposition. Moreover, there is another advantage to lifting the problem to 3D, as we can explicitly utilize the spatial distances between 3D points, to further disambiguate points sharing similar DINO features with their spatial proximity. We define the final weight matrix as the multiplication of the grouped cosine similarity matrix as described in Sec. 4.1 and a spatial proximity matrix:

$$\mathrm{S}_{\text{euc}}\left(v_i, v_j\right) = \max\left(\exp\left(-\gamma \frac{\left\|p_i - p_j\right\|^2}{\sigma_i \sigma_j}\right)^{-1}, \theta_{euc}\right), \tag{1}$$

where v_i and v_j denote two 3D points, p_i and p_j denote their spatial positions, σ_i and σ_j are point-wise local scaling factors [16] which are standard deviations computed within the K-nearest neighbors of each point, the computed spatial affinities are further thresholded with θ_{euc} to avoid the structure of the graph dominated by the spatial affinity term.

Following TokenCut [13], the grouped cosine similarities are binarized:

$$\mathrm{S}^{*\prime}\left(v_i, v_j\right) = \begin{cases} 1 & \text{if } \mathrm{S}^{*}\left(v_i, v_j\right) \geq \tau \\ \epsilon & \text{otherwise,} \end{cases} \tag{2}$$

where S^{*} is grouped cosine similarity, τ is set to 0.2 and ϵ equals 1e-5. The final weight matrix is defined as

$$w\left(v_i, v_j\right) = \mathrm{S}^{*\prime}\left(v_i, v_j\right) \cdot \mathrm{S}_{\text{euc}}\left(v_i, v_j\right). \tag{3}$$

A.3. Solving Normalized Cut

We perform NCut upon the above-defined graph by solving a generalized eigenvalue problem. The second-smallest eigenvector is found using a matrix-free LOBPCG

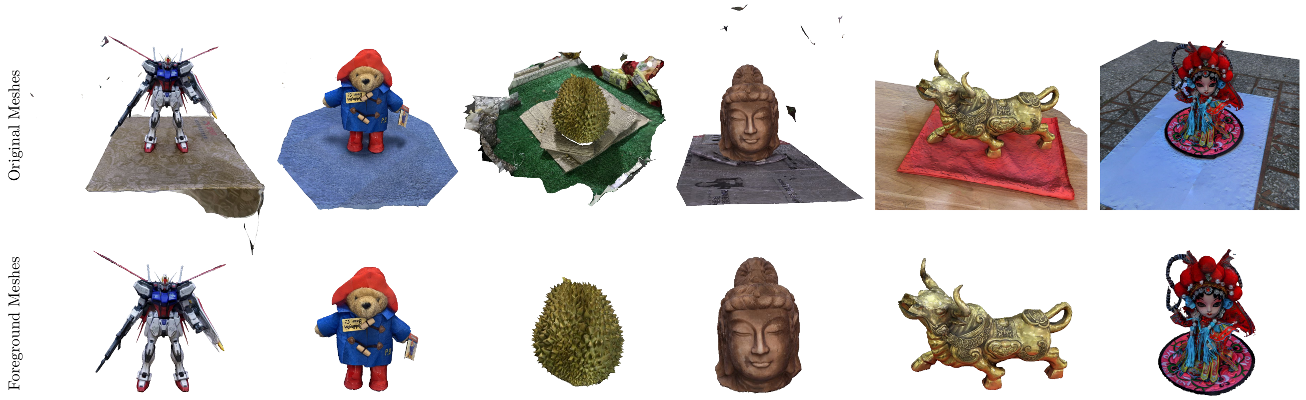

Figure 1. Manually segmented ground truth meshes of BlendedMVS [14]. We manipulate original meshes with background and only keep the foreground models for evaluating 3D reconstruction and multi-view segmentation.

method [4, 11]. We extract the discrete partition from the continuous one indicated by the second-smallest eigenvector with an evenly spaced search over possible splitting points, and the one minimizing the NCut criterion is taken as the final splitting point of bipartition.

A.4. Salient object segmentation and detection

Given a bipartition of the graph, we need to determine which partition corresponds to the foreground. TokenCut [13] assumes that the foreground is less connected to the entire graph and thus can be identified as the partition containing the largest absolute value of the eigenvector. We find this strategy applies to SfM point clouds as well.

The extracted foreground point cloud sometimes still contains several floating 3D points far away from the salient object. We optionally perform euclidean clustering and take the largest connected component as the final foreground point cloud. We then generate a compact 3D bounding box for the foreground object, which is achieved by taking the axis-aligned bounding box (AABB) in an object-centric space. First, we transform the point cloud to an object-centric space. If the gravity direction is not known, we align the up direction of the world space with the normal direction of the estimated ground plane. Then, we perform principal component analysis (PCA) on the foreground point cloud and reorient the world frame to the principal directions of the foreground point cloud with the up direction fixed. Then, we extract the AABB of the foreground point cloud in the object-centric space and optionally extend the AABB to the estimated ground plane. The above-described pipeline leads to satisfactory segmentation and bounding boxes for most of the videos in the CO3D [10] dataset, combined with the noise-aware pseudo-ground-truth definition in Fig. 3 of the main text, is sufficient for automatically generating massive training data for our salient object segmentation Transformer.

B. Unbounded Scene Modeling

We use the scene contraction function proposed in mip-NeRF 360 [1] to represent unbounded scenes:

$$\operatorname{contract}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\| \leq 1 \\ \left(2 - \frac{1}{\|\mathbf{x}\|}\right) \frac{\mathbf{x}}{\|\mathbf{x}\|} & \text{if } \|\mathbf{x}\| > 1. \end{cases} \tag{4}$$

We use the above function with L_∞ norm when using grid-based scene representations to accelerate training and use L_2 norm when using MLP-based fields. However, different from mip-NeRF 360, which focuses on the modeling of the entire unbounded scene, we pay more attention to the reconstruction quality of foreground objects. Thus we leverage different feature grids and MLPs to model the foreground and background regions separately. We keep the foreground region uncontracted and apply linear contraction to the interior of the background region and non-linear contraction to the exterior of the background region:

$$\operatorname{contract}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\| \leq 1 \\ g(\mathbf{x}) & \text{if } 1 < \|\mathbf{x}\| \leq \alpha \\ \left(2 - \frac{1}{\|g(\mathbf{x})\|}\right) \frac{g(\mathbf{x})}{\|g(\mathbf{x})\|} & \text{if } \|\mathbf{x}\| > \alpha, \end{cases} \tag{5}$$

where $g(\mathbf{x}) = \frac{\mathbf{x}}{\alpha}$ which linearly contract the space. We rescale the scene such that the foreground bounding box fits into a unit sphere (for L_2 norm) or a unit box (for L_∞ norm), which left the foreground modeling in Euclidean space. The $\|\mathbf{x}\| > 1$ region is recognized as background and is parameterized by separate fields. We find that $\alpha = 5.0$ works well across object instances and one can further tune α based on the relative size of the foreground and background regions. We name the contraction function in Eq. (5) foreground-object-aware scene contraction. This contraction function facilitates independent scene parameterization settings for the foreground and background regions.

C. Evaluation Details

C.1. Coarse decomposition

Our coarse decomposition aims to segment the salient foreground object and generate its compact 3D bounding box from an SfM point cloud. We evaluate all methods based on the produced 3D bounding boxes. We do not evaluate the 3D segmentation directly since SfM point clouds are noisy, containing points not lying on any surface and thus whose ground truth segmentation is hard to define and generate.

We label the ground truth 3D bounding boxes for CO3D [10], BlendedMVS [14] and DTU [5] in a semi-automated pipeline. First, we preprocess the dataset to get geometric models of the foreground object. For the CO3D dataset, we filter the provided MVS point cloud and only keep the foreground points. For the BlendedMVS dataset, we manually manipulate the ground truth mesh to remove all background vertices and faces as shown in Fig. 1. For the DTU dataset, we filter the ground truth point cloud with Visual Hull based on object masks annotated in [15]. Then, the bounding boxes are inferred based on a plane-aligned principal component analysis of foreground geometries, in the same way as described in Appendix A. Finally, we manually inspect all generated bounding boxes in their corresponding SfM point clouds and discard incorrect ones, which are mostly caused by partial or noisy foreground geometries.

To evaluate all methods fairly, a shared 3D bounding boxes generation pipeline is applied to all SfM point cloud segmentation results. We transform all world spaces into object-centric spaces defined by the ground truth bounding boxes. Then we apply euclidean clustering and take the largest connected component as the foreground point cloud. This prevents the bounding box from being affected by occasional floating segmentation noises. Finally, the axis-aligned bounding boxes are taken upon the foreground point clouds. Since SfM point clouds produced with LoFTR matches contain complete foreground geometries, we can also generate the bounding boxes in arbitrary world frames by reorienting the world frame to the principal directions of the segmented foreground point cloud with the up direction aligned with gravity. We found these two schemes led to similar results for our methods. We report average precisions (AP) of 3D IoU with thresholds of 0.5 and 0.7. We use the 3D IoU implemented in PyTorch3D [9].

C.2. Fine decomposition

Multi-view segmentation. Qualities of multi-view segmentation reflect both the performance of 3D reconstruction and foreground object decomposition. We evaluate segmentation masks with Mask IoUs and Boundary IoUs [3] averaged over multiple views. The Mask IoU is defined as

$$\text{Mask-IoU}\,(G, P) = \frac{|G \cap P|}{|G \cup P|}, \tag{6}$$

where G is the ground truth binary mask and P is the predicted binary mask. The Boundary IoU is defined as

$$\text{Boundary-IoU}\,(G, P, G_d, P_d) = \frac{|(G_d \cap G) \cap (P_d \cap P)|}{|(G_d \cap G) \cup (P_d \cap P)|}, \tag{7}$$

where G_d and P_d are the set of values in the boundary region of the binary masks. The Boundary IoU focuses more on boundary quality, which better reflects the meticulousness of our produced masks. Please refer to [3] for more details.

3D reconstruction. We evaluate the quality of 3D reconstruction with Chamfer l_2 distance. BlendedMVS [14] provides high-fidelity ground truth meshes reconstructed with a commercial 3D reconstruction tool. However, most of the meshes contain both the foreground object and its surroundings. We manually segment the foreground mesh of 5 objects for evaluation of 3D reconstruction; the original and foreground-only meshes are shown in Fig. 1. We render multi-view object masks with foreground-only meshes. When evaluating baselines that jointly reconstruct the foreground object and its surroundings, we remove all points from their meshes lying outside the visual hull defined by the rendered multi-view masks. We compute Chamfer distance by sampling 100,000 points from the ground truth and the predicted meshes.

D. More Ablation Studies

Ablations on the grouped cosine similarity. When segmenting the foreground object with our NCut-based pipeline, we find that using the proposed grouped cosine similarity (S^* in Eq. (3)) can better segment objects with complex structures and appearances than using cosine similarity on flattened features (S_C). We show quantitative results in Table 2.

	CO3D	
	AP@0.5	AP@0.7
Ours NCut + Grouped Cosine Similarity	**0.867**	**0.306**
Ours NCut + Cosine Similarity	0.673	0.235

Table 2. **Ablation results of the grouped cosine similarity.**

Ablations on regularizations enforcing fine decomposition. We propose several regularization terms to enforce the decomposition of foreground and background when training our implicit neural scene representation, including additional constraints directly applied on the SDF with the segmented foreground point cloud and the estimated ground plane, and a $\mathrm{Beta}(0.5, 0.5)$ prior on object masks rendered with our foreground reconstruction. Depending on the modeling difficulties of the foreground object and its surroundings, we find different video sequences rely more or less

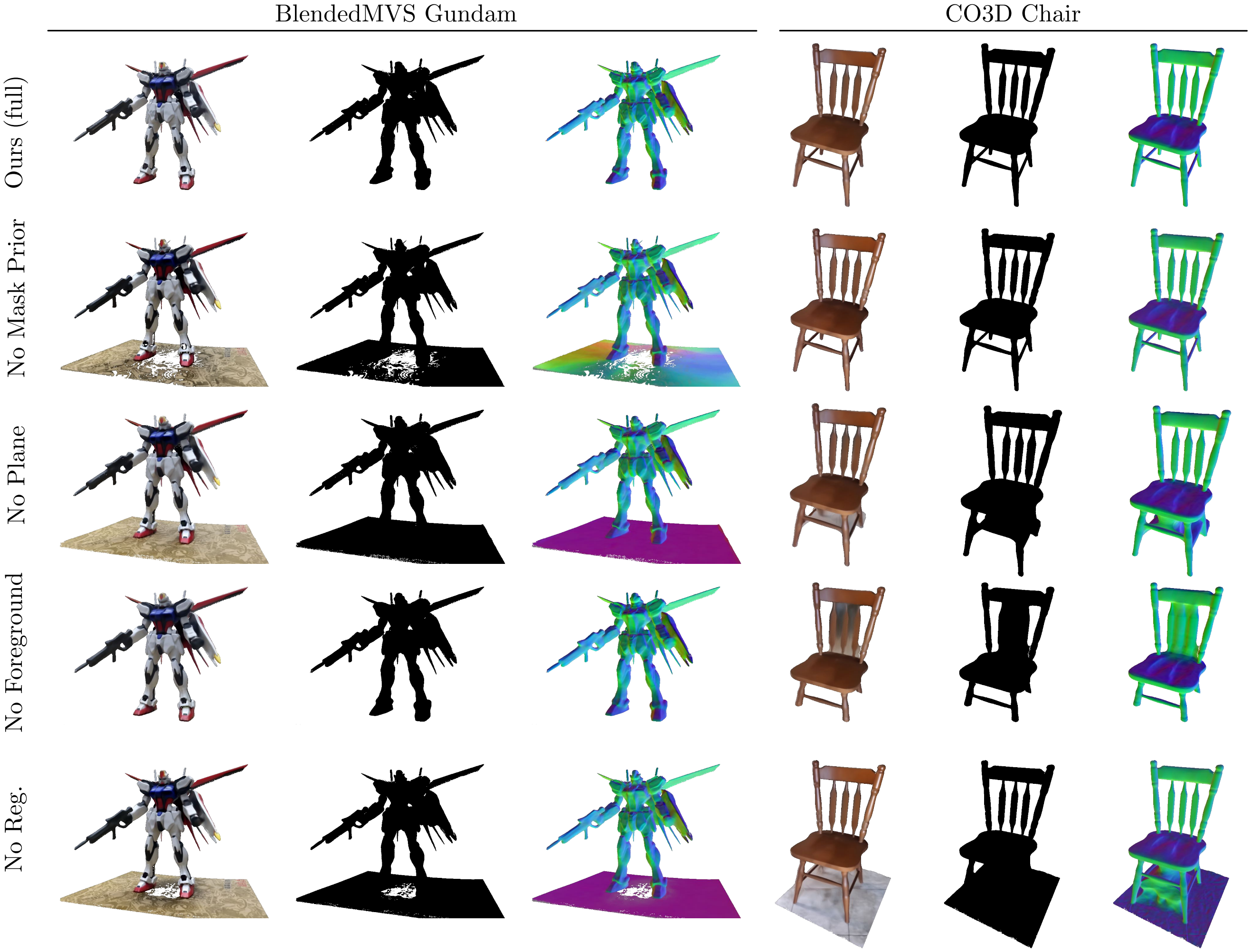

Figure 2. Ablation studies on different regularization terms and their effects on final decomposition results. We visualize renderings of the foreground SDF-based radiance field including colors, segmentation and normals. We use MLP-based fields without feature grids in all experiments.

on different regularizations. But overall, these three regularizations combined lead to robust results among most of the sequences.

We show the effectiveness of each term with additional ablation studies on two objects, the Gundam action figure from BlendedMVS and a chair from CO3D. Both of them contain complex foreground geometries and cluttered backgrounds. We use MLP-based scene representations without multiresolution hash encodings [8] in this ablation. As shown in Fig. 2, when different networks are used for modeling different spatial partitions and no regularization is used (*No Reg.*), the ground planes are modeled by the foreground SDF network in both cases. When the $\mathrm{Beta}(0.5, 0.5)$ prior regularization on rendered object masks is disabled (*No Mask Prior*), the Gundam sequence has part of its ground plane modeled by the foreground SDF network while the chair is decomposed successfully. We attribute this to the complex textures on the ground plane in the Gundam sequence, which is hard to model with the NeRF covering the region around the ground plane. In contrast, the ground plane in the chair sequence has simpler textures and is easily modeled by a tiny NeRF. When the ground plane constraint is disabled (*No Plane*), both of the two sequences have degenerate decompositions. The Gundam sequence is less sensitive to the foreground point cloud constraint (*No Foreground*), but the legs of the chair sequence fail to be modeled by the foreground SDF and the thin structures of its back are blurry and noisy. When using scene representations with multiresolution hash encodings, we find the decomposition results more robust to the combination of different regularization terms.

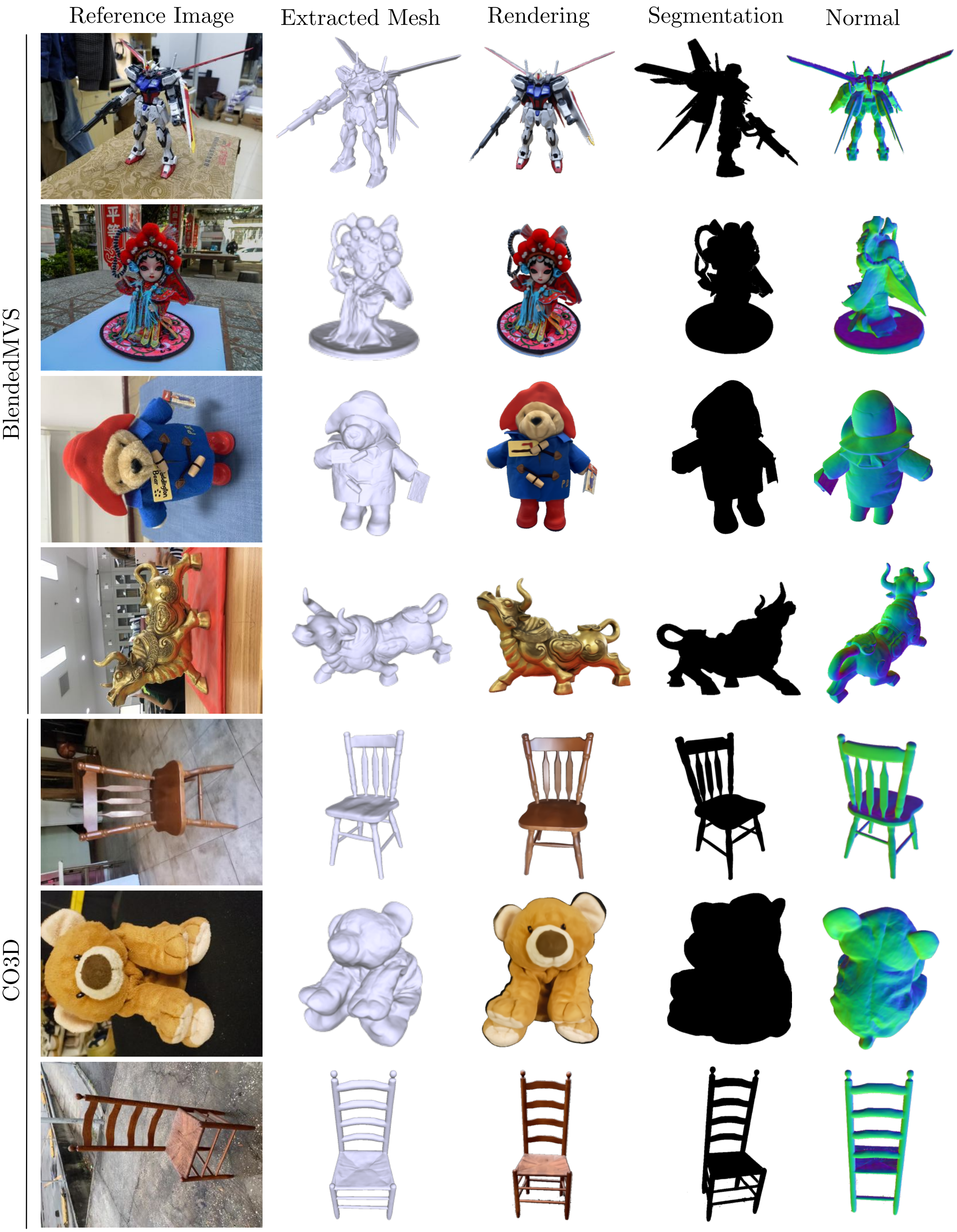

Figure 3. More qualitative results on 3D reconstruction and multi-view foreground-only renderings. We show the reconstruction results of MLP-based fields without utilizing feature grids.

E. More Implementation Details

Point cloud Transformer. Our SfM point cloud segmentation Transformer consists of 2 Transformer encoder layers with linear attentions [6]. The 384-d point-wise DINO features are added with positional encodings as inputs. We use discrete sinusoidal positional encodings defined on a 128^3 voxel grid and interpolate it with continuous 3D positions. We found discrete learned positional encoding and continuous sinusoidal encoding lead to similar results, we use discrete sinusoidal encoding for its simplicity. Following the architecture of ViT-Small, we use 6 attention heads. In contrast, we use a smaller hidden dimension of only 96 instead of 384. We train our Transformer with Adam optimizer [7] and a learning rate of 1e-3 for 20 epochs. We use gradient clipping with a maximum norm of 0.5 to avoid the training being affected by extreme outliers caused by incorrect pseudo-ground-truths.

Timing of coarse decomposition. We provide timings of our unsupervised point cloud segmentation pipeline with NCut and our Transformer-based segmentation in Table 3. We only report the timings of the segmentation, excluding preprocessing of building DINO point clouds and postprocessing of generating object boundings boxes.

Method	Timing (ms)	
	Downsampled Point Cloud (10k points)	Original Point Cloud (170k points)
NCut	10647.7	—
Transformer	5.8	83.9

Table 3. Timing of NCut and Transformer based segmentation on the downsampled and original SfM point clouds. All results are averaged over 100 runs. It is infeasible to apply NCut on the original point cloud due to memory limitations.

F. More Qualitative Results

We show more qualitative results on 3D reconstruction and multi-view segmentation in Fig. 3. The reconstructed meshes and multi-view renderings of foreground objects are free of backgrounds, illustrating the effectiveness of our coarse-to-fine salient object decomposition and reconstruction pipeline.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, 2021. 1

[3] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, 2021. 3

[4] Jed A. Duersch, Meiyue Shao, Chao Yang, and Ming Gu. A robust and efficient implementation of LOBPCG. *SIAM J. Sci. Comput.*, 40, 2018. 2

[5] Rasmus Ramsbøl Jensen, Anders Lindbjerg Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 3

[6] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 6

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[8] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM ToG*, 2022. 4

[9] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 3

[10] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021. 2, 3

[11] Andreas Stathopoulos and Kesheng Wu. A Block Orthogonalization Procedure with Constant Synchronization Requirements. *SIAM J. Sci. Comput.*, 2002. 2

[12] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1

[13] Yangtao Wang, Xi Shen, Shell Hu, Yuan Yuan, James Crowley, and Dominique Vaufreydaz. Self-Supervised Transformers for Unsupervised Object Discovery using Normalized Cut. In *CVPR*, 2022. 1, 2

[14] 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. 2, 3

[15] 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 *NeuIPS*, 2020. 3

[16] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *NeurIPS*, 2004. 1