SPAR3D: Stable Point-Aware Reconstruction of 3D Objects from Single Images

Supplementary Material

Zixuan Huang[1,2*] Mark Boss[1] Aaryaman Vasishta[1] James M. Rehg[2] Varun Jampani[1]
[1]Stability AI, [2]UIUC

A. Limitations

The main limitations of SPAR3D are twofold. First, the point clouds generated during the point sampling stage occasionally exhibit artifacts, such as small surface spikes or detached parts. While these imperfections can typically be remedied through SPAR3D's editing capabilities with minimal effort (see Fig. 7 in the main paper), exploring more principled solutions (e.g. improving the denoiser design or diffusion samplers) could further enhance the utility and robustness of our method.

Second, although SPAR3D learns material decomposition during training, the accuracy of these decompositions can sometimes be suboptimal. This limitation is primarily due to the inherent ambiguity of inverse rendering from a single image, especially when learned in an unsupervised manner. Unsupervised decomposition learning is useful given the scarcity of 3D assets containing high-quality Physically Based Rendering (PBR) materials and is scalable to real-world multi-view datasets. However, investigating semi-supervised learning techniques may offer a pathway to more plausible material estimations in future work.

B. Additional Implementation Details

Additional Training Details. We use a batch size of 128 for point diffusion, and batch sizes of 168/96 for initial/later phases of the meshing stage. Our learning rates are 3e-5 (point) and 5e-5 (meshing) under a cosine decay. We use a linear learning rate warm-up for 1000 steps, and an AdamW optimizer with weight decay of 0.05. Training our 2B model takes 10 days on three 8-H100 nodes. Our training data is the same as TripoSR [56] with additional point cloud curation steps. Our point clouds are generated by rendering and unprojecting depth maps, which remove the internal surfaces that can be challenging to learn.

Additional Illustrations of our Architecture. We show additional illustrations of our point cloud denoiser and our meshing model in Fig. 9 and Fig. 10. We hope these illustrations facilitate a better understanding of our architecture.

C. More Results

Comparison to More Baselines. In Fig. 12, we show additional qualitative comparison with Era3D and MeshLRM [64]. We include surface normal renderings for a better surface visualization. Beyond better texture, our geometry also demonstrates crisp details despite having significantly fewer mesh faces than other methods.

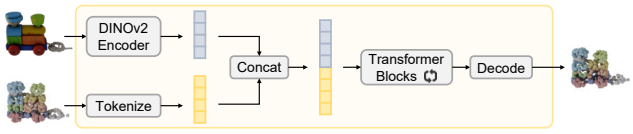

Figure 9. **Point Cloud Denoiser Architecture.** We illustrate the architecture of our point cloud denoiser. The point cloud denoiser takes the noisy point cloud and the image as input, and produces a denoised point cloud. The image and the noisy point cloud are encoded as latent vectors and concatenated together. The concatenated latent vectors are processed by a set of transformer blocks and decoded as the denoised point cloud.

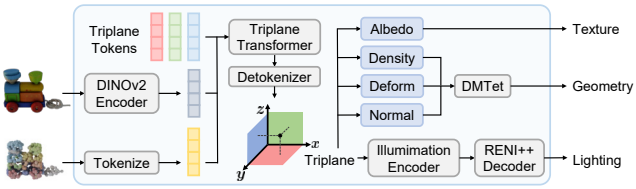

Figure 10. **Meshing Model Architecture.** We illustrate the architecture of our meshing model, which takes the point cloud and the image as input, and produces a textured mesh and an environment map as output. Specifically, the meshing model first encodes the image and the point cloud as latent vectors. The learnable triplane tokens are then processed by the triplane transformer conditioned on the latent vectors. We query the triplane with MLPs to obtain albedo, density, vertex deformation and surface normal, which are converted to a textured mesh using DMTet. The triplane also produces an environment map using the illumination prior from RENI++. The metallic and roughness values are estimated from the image directly and are omitted here for simplicity.

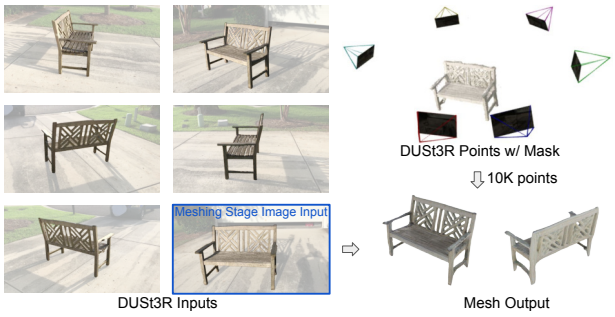

Figure 11. **Ours w/ DUSt3R.** Our robust meshing model can use point clouds from different sources at higher resolution.

Figure 12. **Additional Visual Comparison.** The images are from 3D-TopiaXL and MeshLRM demo pages. SPAR3D reconstructions exhibit significantly better details in both geometry and texture than other methods at a much faster speed.

Figure 13. **Decomposition and Relighting Results.** We show decomposed albedo and relighting results of SPAR3D in comparison with SF3D. The albedo estimated by SPAR3D has less baked-in lighting compared with SF3D and results in better relighting outcomes.

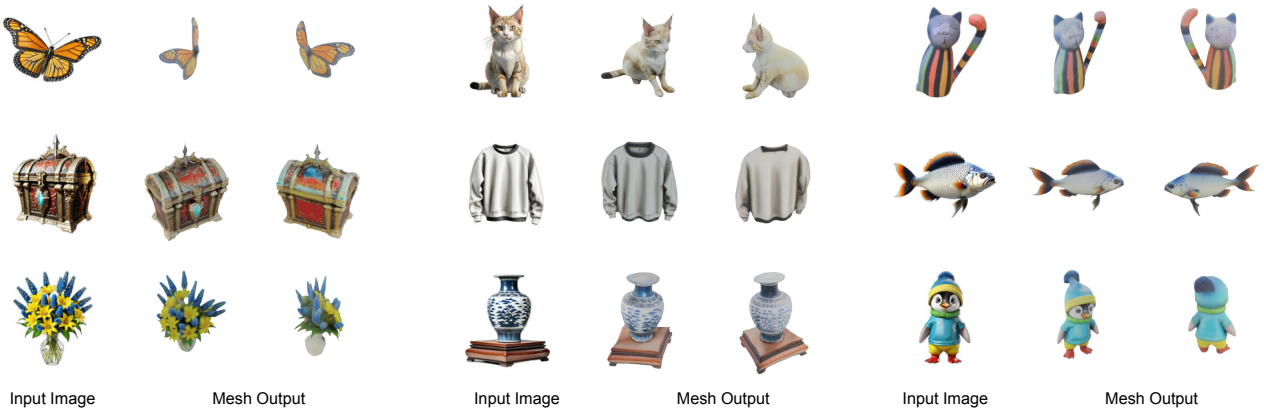

Figure 14. **Additional In-the-wild Results.** We show additional results of SPAR3D on in-the-wild images. The reconstructed meshes achieve high fidelity and exhibit great surface details.

Inference with Other Point Clouds. SPAR3D is not restricted to its own generated points, and we find our point conditioner relatively robust to different resolutions. In Fig. 11, we show a qualitative example of feeding a DUSt3R [62] point cloud (10K sampled points) to our meshing model. We observe our model still produces high-quality reconstructions.

Decomposition Results. We show decomposition and relighting results of SPAR3D in comparison with SF3D, which is a full regressive method. As shown in Fig. 13, our estimated albedo often has less baked-in lighting artifacts compared with SF3D, which improves the quality of relighting under different illumination conditions.

Meshing Stage Ablation. To better understand the effect of point cloud conditioning, we evaluate the meshing stage with groundtruth point clouds on GSO. This leads to a CD of 0.070 (vs. 0.120 original) and a PSNR of 20.4 (vs. 18.6 original). The performance improvement with oracle further verifies our hypothesis that even sparse point clouds effectively reduce reconstruction ambiguity.

Additional In-the-wild Results. We present additional reconstruction results on in-the-wild images. In Fig. 14, we show the reconstructions of SPAR3D on images from 3D-Arena (Ebert, 2024). On this data source, SPAR3D also achieves high reconstruction quality. This further validates the strong generalization ability of SPAR3D.