

Towards Realistic Example-based Modeling via 3D Gaussian Stitching

– Supplemental Material –

1 IMPLEMENTATION DETAILS

All experiments are carried out on a single NVIDIA RTX 3090 GPU. We use the Adam optimizer for 3D Gaussian feature attributes, with learning rates of 0.02 and 0.001 for the SH feature’s zero-frequency and high-frequency components, respectively. Each composition pair’s optimization takes less than 5 minutes in total. During the S-phase, we sample 5,000 points as a batch from the 3D Gaussians’ point cloud for each iteration rather than using the entire set; otherwise, the training speed will be slow. Throughout the KNN and palette collection, customized CUDA kernels are used to accelerate the process in less than three seconds. The entire optimization takes 6,000 iterations, consistently maintaining the loss in the S-phase and boundary conditions, with the T-phase beginning at 4,500 and continuing until completion.

2 FAIRNESS OF COMPARISON ON REAL-WORLD DATA

We compared our method with SeamlessNeRF [Gong et al. 2023], but we encountered a disparity when conducting our experiment on real-world data, prompting us to enhance the baseline performance using our approach. The discrepancy arises from the fact that SeamlessNeRF, built upon TensorRF [Chen et al. 2022], was not implemented for editing scene geometry, such as segmentation and cropping. In real-world scenarios, precise masks for target objects are often unavailable, thus making the SeamlessNeRF hardly directly applied to real-world data. To compare with SeamlessNeRF on the real-world data, we utilized the interactive editing capability of our framework to generate alpha channels rendered by 3DGS [Kerbl et al. 2023] to crop the target object from the background. Additionally, to ensure editing effects are based on clean density fields, we introduced a random background argumentation to mitigate artifacts during the SeamlessNeRF training process:

$$\mathcal{L}_{\text{alphacolor}} = \|w_q(c_q - \delta_q) - \hat{\alpha}_q(\hat{c}_q - \delta_q)\|_2^2 \quad (1)$$

where w_q is the accumulated weights along ray q in NeRF’s render equation, and $\hat{\alpha}_q$ is the alpha channels generated for supervision. In the equation, c_q is the color computed by our model, and \hat{c}_q is the corresponding ground-truth color. The black and white background colors δ_q are randomly selected for each ray q with equal possibility in our implementation. Fig. 1, shows that without this loss, too many artifacts prevent SeamlessNeRF from performing seamless editing effects. Therefore, the fairness of comparison between ours and the baseline’s effects is contributed by the strength of our approach and some additional efforts, which, in turn, gives proof of our superiority.

2.1 Choice of Benchmark

Given the interactive nature of our method, the outcomes in all cases hinge on users’ selections of compelling examples and their efforts to craft semantically meaningful results. Finding an existing



Fig. 1. Improvement for SeamlessNeRF. With the help of mask loss and the mask provided by our method, artifacts are significantly suppressed, resulting in a fair comparison.

dataset tailored to this specific task proved challenging. Consequently, we opted to utilize datasets such as BlendedMVS [Yao et al. 2020], Mip360 [Barron et al. 2022], and the synthetic data employed in SeamlessNeRF [Gong et al. 2023]. It is worth mentioning that while the latter dataset is not derived from real-world sources, we have included it to underscore the discernible disparities between our approach and the baseline.

3 MORE QUALITATIVE COMPARISON

We present a more extensive qualitative comparison, encompassing all cases in our benchmark. Direct visualization is considered to be more comprehensive than a user study. In Fig. 2, the rows (from top to bottom) represent cases numbered from 1 to 21. Cases 1-13 are derived from real-world data obtained from BlendedMVS and Mip360, while cases 14-21 originate from synthesis data used in SeamlessNeRF. The columns (from left to right) depict part models, raw composites, and two views of our method and the baseline, respectively.

4 MORE QUANTITATIVE COMPARISON

4.1 Evaluating with VQA

The VQA (Video Quality Assessment) method acts as a tool to assess video quality, which has become increasingly essential due to the rapid increase of 2D user-generated content. Therefore, instead of evaluating the 3D models directly, we utilize VQA [Wu et al. 2023] to assess the quality of the videos generated from our models. To produce coherent video sequences, we configure the camera orbit to showcase the models and ensure that the camera remains focused on the models at all times. Specifically, for results where the target field occupies a substantial space, circular camera orbits are employed to provide panoramic views, while for those occupying

specific angles, spiral camera orbits are utilized (refer to our videos for visual demonstration).

Statistic. Table 2 provides detailed information from the table presented in the main text. In Tab. 2, a positive number indicates that our method outperforms the baseline. The column Δt represents the difference in the technical score, which typically relates to distortions or artifacts, while the column Δa represents the difference in the aesthetic score, which typically reflects preferences and recommendations regarding content. It is important to note that the Δa metric for certain cases (e.g., case 10, case 12) may not accurately reflect the true performance. This is because the VQA model struggles to comprehend seamless editing effects and instead favors situations with more diverse colors present.

4.2 Why Not FID.

To compare using FID, we collected training data from the benchmark to serve as the ground truth set, enabling the identification of the distribution of realistic objects. However, the FID scores for both methods exceeded 300, far beyond the normal range of previous generation tasks. This suggests that comparing with the FID metric makes no sense. The main reason is that the created composites themselves did not appear in any dataset. Additionally, in some cases, the backgrounds were missing, further complicating the FID algorithm’s assessment.

	ours	SeamlessNeRF
average optimizing time ↓	< 4 min	> 1 h
real-time adjustment	YES	NO

Table 1. Speed Comparison between ours and the baseline.

5 SPEED COMPARISON

Table 1 presents a concise comparison of speed, demonstrating that our method also surpasses the baseline in terms of optimization efficiency. In addition to the advantage of our method in terms of user time consumption during interactive adjustments, particularly noteworthy is the optimization speed: SeamlessNeRF requires over one hour, whereas ours takes less than 5 minutes. For visualizing the optimization process, please refer to our video.

REFERENCES

- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5470–5479.
- Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *European Conference on Computer Vision (ECCV)*.
- Bingchen Gong, Yuehao Wang, Xiaoguang Han, and Qi Dou. 2023. SeamlessNeRF: Stitching Part NeRFs with Gradient Propagation. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. 2023. Exploring Video Quality Assessment on User Generated Contents from Aesthetic and Technical Perspectives. In *International Conference on Computer Vision (ICCV)*.
- Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. 2020. BlendedMVS: A Large-Scale Dataset for Generalized Multi-View

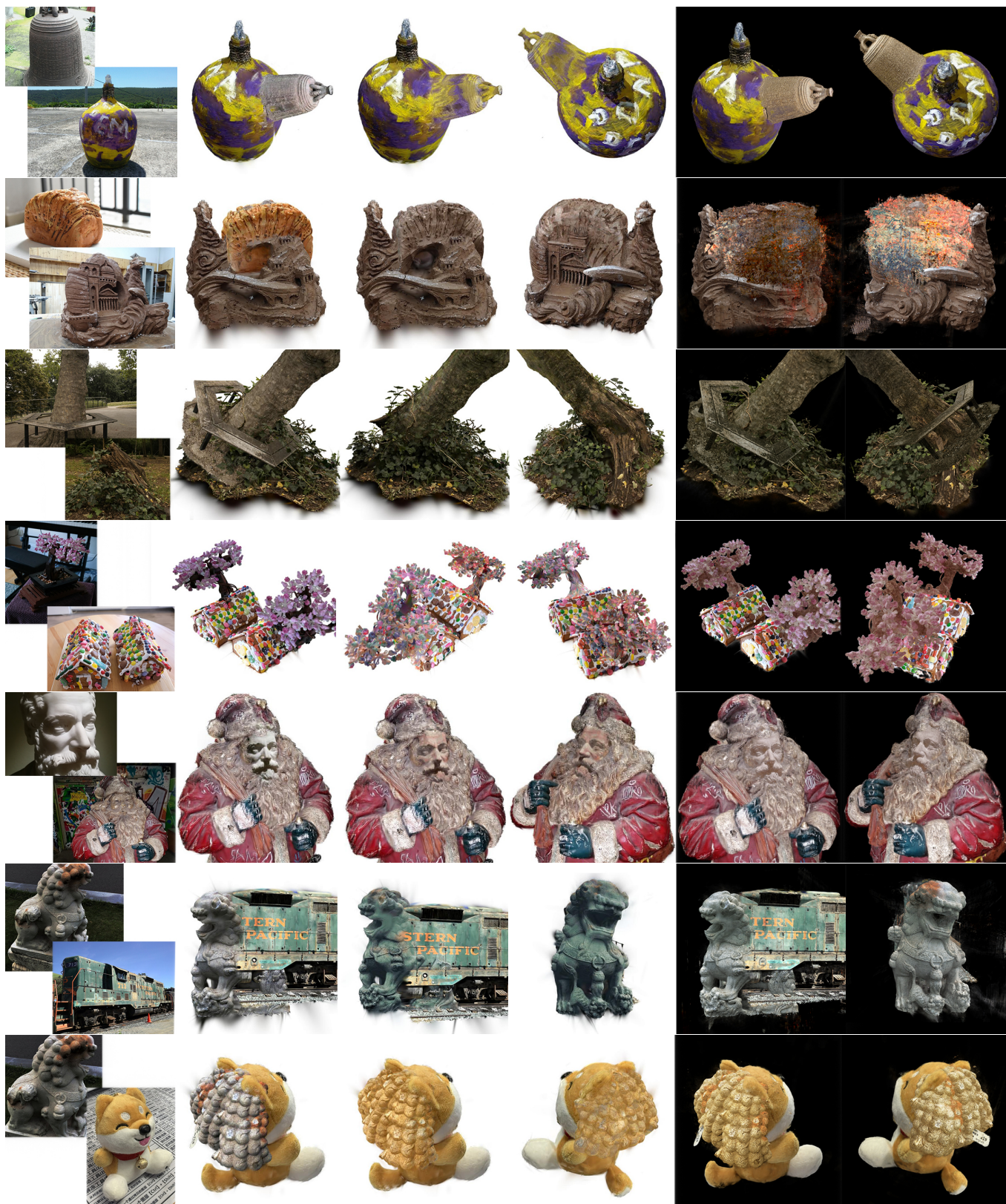


Fig. 2. Case 1-7 are displayed in rows from top to bottom. The rightmost two columns present the baseline results for comparison.



Fig. 2. Case 8-14 are displayed in rows from top to bottom. The rightmost two columns present the baseline results for comparison.



Fig. 2. Case 15-21 are displayed in rows from top to bottom. The rightmost two columns present the baseline results for comparison.

	LIVE_VQC		KoNViD-1k		LSVQ_Test		LSVQ_1080P		YouTube_UGC	
	$\Delta t \uparrow$	$\Delta a \uparrow$	$\Delta t \uparrow$	$\Delta a \uparrow$	$\Delta t \uparrow$	$\Delta a \uparrow$	$\Delta t \uparrow$	$\Delta a \uparrow$	$\Delta t \uparrow$	$\Delta a \uparrow$
case1	-0.075	+0.149	-0.058	+0.142	-0.049	+0.140	-0.059	+0.148	-0.066	+0.094
case2	+0.887	+0.453	+0.804	+0.394	+0.760	+0.376	+0.808	+0.442	+0.841	+0.413
case3	+0.057	+0.326	+0.052	+0.284	+0.049	+0.270	+0.052	+0.318	+0.054	+0.298
case4	+0.706	-0.337	+0.640	-0.293	+0.605	-0.278	+0.642	-0.327	+0.669	-0.307
case5	+0.077	+0.078	+0.070	+0.067	+0.066	+0.064	+0.070	+0.075	+0.073	+0.070
case6	+0.370	+0.051	+0.335	+0.044	+0.317	+0.043	+0.337	+0.050	+0.351	+0.047
case7	+0.528	+0.132	+0.478	+0.115	+0.454	+0.109	+0.482	+0.129	+0.501	+0.121
case8	+0.018	-0.179	+0.016	-0.156	+0.015	-0.148	+0.016	-0.174	+0.017	-0.163
case9	+1.053	+0.426	+0.953	+0.372	+0.902	+0.355	+0.957	+0.416	+0.997	+0.390
case10	+0.887	-0.039	+0.804	-0.033	+0.761	-0.032	+0.807	-0.037	+0.841	-0.035
case11	+0.284	+0.018	+0.257	-0.022	+0.244	+0.012	+0.259	-0.027	+0.269	-0.014
case12	+0.072	-0.293	+0.065	-0.256	+0.062	-0.242	+0.065	-0.285	+0.068	-0.268
case13	+0.349	-0.120	+0.317	-0.104	+0.299	-0.099	+0.318	-0.116	+0.331	-0.109
case14	+0.459	+0.014	+0.416	-0.012	+0.392	+0.011	+0.417	+0.014	+0.435	-0.013
case15	+0.101	-0.426	+0.092	-0.371	+0.087	-0.353	+0.092	-0.415	+0.097	-0.390
case16	+0.040	+0.091	+0.036	+0.079	+0.034	+0.076	+0.036	+0.089	+0.038	+0.082
case17	+0.053	-0.014	+0.048	+0.012	-0.015	+0.012	-0.028	+0.014	-0.050	-0.013
case18	+0.386	+0.117	+0.349	+0.101	+0.330	+0.097	+0.350	+0.114	+0.366	+0.107
case19	+0.019	-0.205	+0.017	-0.178	+0.016	-0.170	+0.017	-0.199	+0.018	-0.187
case20	-0.088	-0.091	-0.079	-0.079	-0.075	-0.075	-0.081	-0.089	-0.083	-0.083
case21	+0.022	-0.031	+0.019	-0.027	+0.018	-0.015	+0.020	-0.030	+0.021	-0.018
average	+0.295	+0.006	+0.268	+0.004	+0.251	+0.007	+0.266	+0.005	+0.276	+0.001

Table 2. Per-case Quantitative Results. We color each cell as **better** and **worse**.