

OMeGa: Joint Optimization of Explicit Meshes and Gaussian Splats for Robust Scene-Level Surface Reconstruction

Supplementary Material

1. Implementation Details

1.1. Mesh Extraction Methods

We outline the mesh extraction methods used for different baselines in Tab. 5.

Method	Mesh Extraction	Post hoc Extraction
2DGS	TSDF	✓
3DGS	TSDF	✓
GSRec	Poisson	✓
SuGaR	Poisson	✓
GOF	Marching Tetrahedra	✓
PGSR	TSDF	✓
GaMeS	Poisson	×
OMeGa	Poisson	×

Table 5. Mesh extraction methods used in the experiments

1.2. Hyperparameters

Parameters. Regarding the coefficient vector of scale, s_{rel} , we set its range to $[0.5, 1.5]$, which guarantees that the splats stay on their corresponding surfaces while maintaining flexible representational capacity.

Weights of losses. We follow the common practice in prior works and set rendering loss coefficient $\lambda_1 = 0.8$. The coefficients for Laplacian smoothing loss and normal consistency loss are inherently dependent on the scale of the scene. In indoor scenes reconstructed by COLMAP, we empirically observe that setting $\lambda_{lap} = 40$ and $\lambda_{nc} = 1$, respectively, achieves a good balance between stability and efficiency during training. We further argue that in other scenarios, these hyperparameters can be straightforwardly rescaled according to the scene scale and the granularity of geometric details. Finally, for the monocular normal loss coefficient λ_n , we adopt an empirical setting of 0.5.

Mesh Refinement. During mesh refinement, we postpone subdivision and removal until after 1000 iterations to ensure stable training in the early stage. Before 15,000 iterations, subdivision and removal are performed every 500 iterations: the top 2% of triangles ranked by reconstruction error are subdivided, while the removal threshold τ_{remove} is fixed at 0.01. This strategy is primarily determined by the granularity of the initial mesh and the desired final resolution. In indoor scenes, we find that such default settings are sufficient to achieve satisfactory performance.

2. Additional Evaluation Metrics

We report training time, peak GPU memory, splat count, and the final model’s disk storage in Tab. 6. While OMeGa trains longer than most baselines—expected given its joint optimization of an explicit mesh, it uses less GPU memory and yields a smaller on-disk model than most MAST3R-initialized baselines.

Method	Training time [min]	Memory [GB]	# Splats [$\times 10^5$]	Disk storage [MB]
3DGS	9	1.60	7.86	185.45
2DGS	16	1.86	8.44	212.60
2DGS [†]	33	9.85	17.85	421.33
2DGS [†] + N	50	9.86	18.36	433.34
2DGS [†] + N + E	71	13.42	18.05	332.17
GSRec	105	2.61	2.11	64.05
SuGaR	182	15.39	11.00	272.87
GOF [†]	65	19.38	14.08	354.76
PGSR [†]	51	15.55	9.77	50.76
GaMeS [†]	23	9.36	12.86	318.90
OMeGa [†]	109	5.94	13.93	287.75

Table 6. Additional metrics for the main experiments on ScanNet. All values are averaged across 8 scenes. [†] indicates initialized with MAST3R results.

3. More Discussion on Illumination and Geometry Reconstruction

Figure 7 shows a representative indoor scene from ScanNet++ scene #8b5caf3398 where illumination differs noticeably between two frames. Such variations are common in real-world indoor captures and are known to challenge Gaussian Splatting-based algorithms.

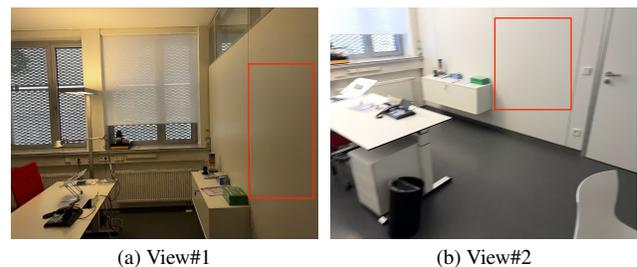


Figure 7. An example of intense illumination change in the ScanNet++ dataset.

As shown in Figure 8, previous Gaussian Splatting algorithms often overfit the illumination of per-view images by increasing the number of Gaussian Splats in low-texture regions, which leads to multiple layers of Gaussian primitives while the ground truth is a surface (left). This fitting can cause the learned surface to deviate from the true

scene, producing artifacts under novel views and inaccurate geometry when extracting the mesh. We identify the previous work’s process as *shaping geometry to overfit appearance*. By contrast, OMeGa anchors splat means and orientations on mesh faces, so *appearance is learned on the correct geometry* rather than reshaping geometry to overfit appearance. Based on the correct geometry, OMeGa captures light variation through feature embeddings (right). As a result, OMeGa achieves both robust novel-view renderings when lighting differs from training views and accurate geometry reconstruction.

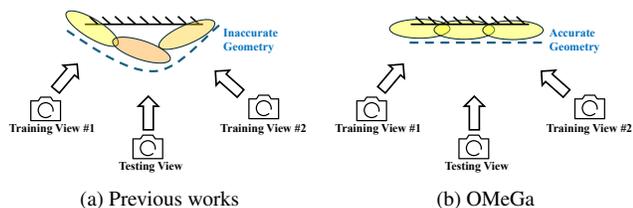


Figure 8. Illustration of the difference between OMeGa and previous Gaussian Splatting algorithms in handling illumination changes.

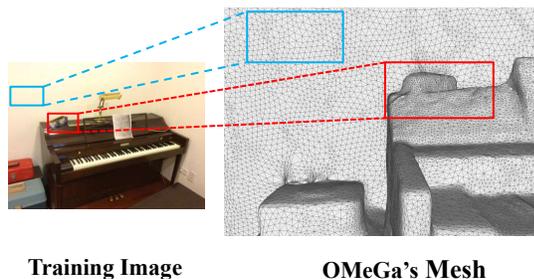


Figure 9. Illustration of the effectiveness of our proposed error-based mesh subdivision strategy.

4. Effectiveness of Error-based Mesh Subdivision Strategy

Figure 9 qualitatively demonstrates the effectiveness of our error-based subdivision strategy in adapting to both low-frequency and high-frequency geometry. In textureless, low-frequency regions, the triangle density remains sparse, preserving computational efficiency (highlighted in the blue box). In contrast, around high-frequency regions (highlighted in the red box), our method iteratively subdivides mesh faces to better fit geometric complexity, leading to a denser and more detailed mesh representation.