FlashSplat: 2D to 3D Gaussian Splatting Segmentation Solved Optimally

Supplementary Material

6 More Implementation details

The implementation of FlashSplat unfolds in two main parts. Initially, the focus is on deriving the contribution set $\{A_e\}$ for every Gaussian across objects e. This step involves projecting 3D Gaussians onto each mask M_v, capturing the product $\alpha_i T_i$ within the alpha blending formula into the buffer A_e where a pixel $M_{ij}^v = e$. This procedure compiles the contributions from every object across all viewpoints into a matrix $\mathcal{A} \in \mathbb{R}^{E \times |\{G_i\}|}$, where E is total number of objects in the 3D scene, and $|\{G_i\}|$ represents the total number of 3D Gaussians. Following this, we allocate labels P_i to each 3D Gaussian G_i based on this contribution matrix, as delineated in equations Eq. 7 and Eq. 8. For binary segmentation, the assignment process simplifies to an arg max operation for optimal assignment. Scene segmentation is resolved through dynamic programming to manage the complexity of multiple assignments, with the specific implementation details provided in list 1.1. The segmentation results for scenes are represented within a matrix $\mathcal{S} \in \mathbb{R}^{E \times |\{G_i\}|}$, where each entry $\mathcal{S}_{m,n} \in \{0, 1\}$ specifies whether the n-th Gaussian belongs to object m.

6.1 Mask association details

Our work primarily concentrates on lifting 2D masks into 3D space, with less emphasis placed on mask association within the core sections of main paper. In this context, we provide additional insights into the methodology used to associate 2D masks in the 3D segmentation experiments presented.

Binary Mask Association. Within binary segmentation scenarios, association among 2D view masks is achieved through the propagation of point prompts across different views. Specifically, for a point prompt $p_i^{2D} \in \mathbb{R}^2$ identified on an object in a single view, this point is back-projected to the 3D space, acquiring a world coordinate $p_i^{3D} \in \mathbb{R}^3$, to locate its corresponding 3D Gaussian G_i. However, due to the prevalence of numerous Gaussians surrounding this 3D point prompt, relying solely on distance for Gaussian correspondence can lead to incorrect outcomes. To mitigate this, we initially identify the $\text{Top} - 10$ closest 3D Gaussian centers using the $\mathcal{L}_2$ distance. Subsequently, the specific Gaussian is determined by selecting the one with the least depth when projected onto the reference view. The center positions of these 3D Gaussians are then projected onto other views to link point prompts associated with the same object across different views. By utilizing SAM [20] to produce masks for each view based on these aligned point prompts, we inherently associate these predicted 2D masks. This method of point prompt propagation is implemented via CUDA kernels, enabling the association of point prompts across all views in under one second.

Scene Mask Association. Beyond binary segmentation, the segmentation of entire 3D scenes without specific point prompts is essential. Therefore, we introduce an alternative approach for associating scene masks across all views. To assign each 2D object with a unique ID in the 3D scene, multiple views are treated akin to a video sequence. Utilizing a zero-shot video tracker [7, 53], we ensure the consistent association and propagation of objects across viewpoints.

7 More qualitative evaluation

To validate the effectiveness of our FlashSplat, we conduct qualitative comparisons on the object removal task with prior works in 3D Gaussian Splatting segmentation, specifically Gaussian-Grouping [53]. As outlined in Sec.4.4, 3D object removal involves entirely eliminating the 3D Gaussians subset of selected objects from the scene, which is a fundamental application of 3D segmentation. For fair comparison, we use identical 2D scene mask set $\{M^v\}$ for both methods. Our process begins with conducting 3D segmentation using these scene masks, followed by the specification of object IDs for removal. We present the multiple object removal results in Fig. 8 and single object removal results in Fig. 9. We render 4 distinct views of the removal results, showing that our FlashSplat can cleanly remove these 3D objects with imperceptible artifacts, while the results of Gaussian-grouping show severe artifacts near the removed 3D objects. These comparisons underscore our method is not only superior for the efficiency of 3D segmentation, but also excels at 3D scene segmentation quality.

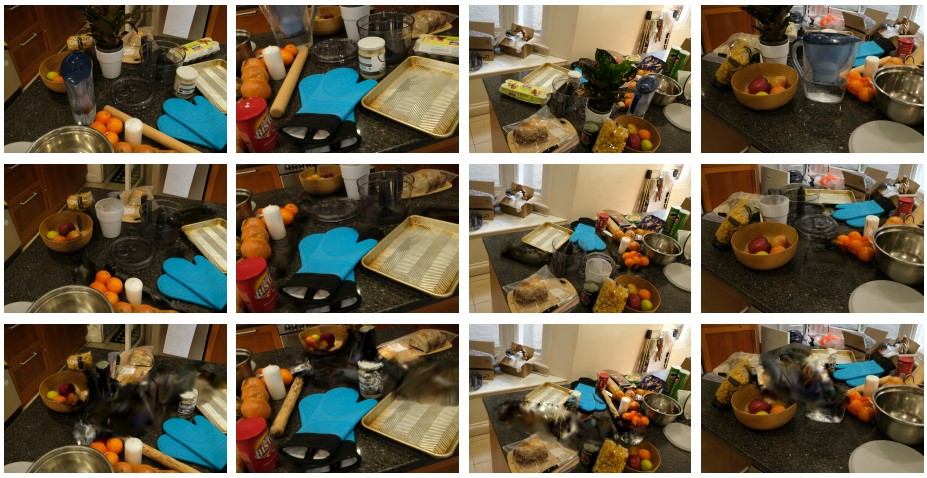

Fig. 8: Multiple Object Removal Comparison. Here we show a qualitative comparison by removing multiple objects from the Counter scene in the MIP-360 [1] dataset. The first row is the ground truth, the second row shows our FlashSplat, and the third row displays the results from Gaussian-Grouping. A total of 5 objects are removed in this scene, the same as in Fig. 4 row 2.

Fig. 9: Single object removal comparison. Here we show a qualitative comparison by removing a single object from the Bear scene in the Instruct-NeRF2NeRF [15] dataset. The first row is the ground truth, the second row is our FlashSplat, and the third row shows the results from Gaussian-Grouping.

8 More discussions

8.1 The effect of background bias γ

Table 3 presents our ablation study on the truck scenes from the T&T dataset [21]. We annotate 5 views 2D mask as target views, and other view masks predicted by SAM [20] are used as reference view masks. With the background bias γ ranging from $[-1, 1]$, we get the 3D segmentation of the truck and then render it to 2D masks to compute the mean IoU. Among these γ values, a setting of $\gamma = 0.4$ produces the optimal mean IoU of 94.2%. This is caused by the noise in masks predicted by SAM (as visualized in Fig. 6), the assignment with $\gamma = 0$ is prone to take background Gaussians as foreground, while this softened refinement helps to reduce such noises.

γ	-0.8	-0.4	0	0.4	0.8
mIoU	82.4	89.6	92.3	94.2	93.8

Table 3: Effect of the background bias γ on the truck scene.

8.2 Quantization in novel view mask

In Sec. 3.4, we outline projecting masks from 3D segmentation results onto novel views using simple quantization and depth-guidance. Here we take mask rendering in binary segmentation as an example to claim why this quantization is

necessary. Without this quantization, novel view masks are produced by first projecting subsets of 3D Gaussians $\{G_i\}_e$ for instance e, and then the mask value for each pixel $\hat{M}^v_{jk}$ is determined by $\hat{M}^v_{jk} = \arg\max_e \rho^e_{jk}$. The absence of quantization leads to masks, displayed in the 2-nd column and 4-th row of Fig. 10, riddled with numerous unintended holes. This phenomenon stems from the nature of 3D Gaussian Splatting, where each Gaussian is a semi-transparent ellipse with opacity o_i. As such, when a pixel (i, j) is rendered, the background 3D Gaussians also affect the alpha blending outlined in Eq. 1, at times more significantly than the foreground Gaussians. Implementing quantization and depth guidance ameliorates these discrepancies, as evident in the first and third columns of Fig. 4. However, it's noteworthy that despite the employment of depth guidance, mask rendering can still produce vague outcomes in scene mask rendering due to the absence of geometric supervision in 3D Gaussian Splatting reconstruction. The geometry that is learned does not conform precisely to the underlying geometry, occasionally impairing the effectiveness of depth guidance.

Fig. 10: Quantization in mask rendering.

8.3 Scene segmentation extension

In Sec. 3.3, we extend our optimal assignment for binary segmentation to scene segmentation. This formulation, shown in Eq. 8, is chosen over a straightforward approach that would simply perform arg max among the E instances. This choice is driven by the non-exclusive nature of Gaussian Splatting, where a Gaussian can be shared between objects. For instance, we quantitatively analyze the Counter scene in the MIP360 [1] dataset under different numbers of given masks. As illustrated in Fig. 11, approximately 20% of the Gaussians in this scene are shared between more than two objects. This phenomenon occurs because, in the 3D reconstruction of 3DGS, supervision is limited to view space, with no additional geometric or semantic constraints to enforce mutual exclusivity.

9 Limitations

Despite the advancements presented by our method in 3D-GS segmentation, we acknowledge several limitations for future exploration. The linear programming approach, although effective, may encounter scalability challenges with

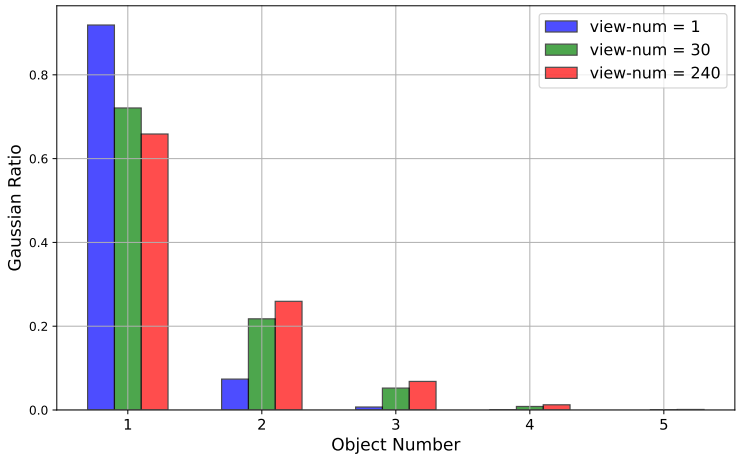

Fig. 11: Visualization of each Gaussian's contributed object number.

significantly larger 3D scenes with substantial spatial resolutions, as we need to traverse all mask pixels. Moreover, due to the inherent property of 3D-GS, rendering 3D segmentation results onto novel view with depth guidance may currently yield ambiguous mask. To address this limitation, incorporating explicit geometry supervision into the 3D-GS reconstruction is essential for more accurately representing the underlying geometry. Additionally, the investigation of adaptive strategies aimed at reducing computational demands and enhancing the adaptability of our method to handle a broader array of scene complexities presents a promising future work.

```python
def multi_instance_opt(all_contrib, gamma=0.):
    """
    Input:
    all_contrib: A_{e} with shape (obj_num, gs_num)
    gamma: background bias range from [-1, 1]

    Output:
    all_obj_labels: results S with shape (obj_num, gs_num)
    where S_{i,j} = 1 denotes j-th gaussian belong i-th object
    """
    all_contrib_sum = all_contrib.sum(dim=0)
    all_obj_labels = torch.zeros_like(all_contrib)
    for obj_idx, obj_contrib in enumerate(all_contrib):
        other_contrib = all_contrib_sum - obj_contrib
        obj_contrib = torch.stack([other_contrib, obj_contrib])
        obj_contrib = F.normalize(obj_contrib, dim=0, p=1)
        obj_contrib[0, :] += gamma
        obj_label = torch.argmax(obj_contrib, dim=0)
        all_obj_labels[obj_idx] = obj_label
    return all_obj_labels
```

Listing 1.1: Multi-instance optimization in PyTorch

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022)
2. Bing, W., Chen, L., Yang, B.: Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. arXiv preprint arXiv:2208.07227 (2022)
3. Cen, J., Fang, J., Yang, C., Xie, L., Zhang, X., Shen, W., Tian, Q.: Segment any 3d gaussians. arXiv preprint arXiv:2312.00860 (2023)
4. Cen, J., Zhou, Z., Fang, J., Yang, C., Shen, W., Xie, L., Jiang, D., Zhang, X., Tian, Q.: Segment anything in 3d with nerfs. In: NeurIPS (2023)
5. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV (2022)
6. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. arXiv preprint arXiv:2309.16585 (2023)
7. Cheng, H.K., Oh, S.W., Price, B., Schwing, A., Lee, J.Y.: Tracking anything with decoupled video segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1316–1326 (2023)
8. Fang, J., Yi, T., Wang, X., Xie, L., Zhang, X., Liu, W., Nießner, M., Tian, Q.: Fast dynamic radiance fields with time-aware neural voxels. In: SIGGRAPH Asia (2022)
9. Fei, B., Xu, J., Zhang, R., Zhou, Q., Yang, W., He, Y.: 3d gaussian as a new vision era: A survey. arXiv preprint arXiv:2402.07181 (2024)
10. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022)
11. Fu, X., Zhang, S., Chen, T., Lu, Y., Zhu, L., Zhou, X., Geiger, A., Liao, Y.: Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In: 3DV (2022)
12. Goel, R., Sirikonda, D., Saini, S., Narayanan, P.: Interactive segmentation of radiance fields. arXiv preprint arXiv:2212.13545 (2022)
13. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: ICCV (2023)
14. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.E.: Baking neural radiance fields for real-time view synthesis. In: ICCV (2021)
15. Hu, B., Huang, J., Liu, Y., Tai, Y.W., Tang, C.K.: Instance neural radiance field. arXiv preprint arXiv:2304.04395 (2023)
16. Hu, X., Wang, Y., Fan, L., Fan, J., Peng, J., Lei, Z., Li, Q., Zhang, Z.: Semantic anything in 3d gaussians. arXiv preprint arXiv:2401.17857 (2024)
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (ToG) (2023)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM TOG **42**(4), 1–14 (2023)
19. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. arXiv preprint arXiv:2303.09553 (2023)
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
21. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Trans. Graph. (2017)
22. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. In: NeurIPS (2022)

23. Lindell, D.B., Martel, J.N.P., Wetzstein, G.: Autoint: Automatic integration for fast neural volume rendering. In: CVPR (2021)
24. Ling, H., Kim, S.W., Torralba, A., Fidler, S., Kreis, K.: Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. arXiv preprint arXiv:2312.13763 (2023)
25. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)
26. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
27. Liu, X., Chen, J., Yu, H., Tai, Y., Tang, C.: Unsupervised multi-view object segmentation using radiance field propagation. In: NeurIPS (2022)
28. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. arXiv preprint arXiv:2308.09713 (2023)
29. Mildenhall, B., Srinivasan, P.P., Cayon, R.O., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: practical view synthesis with prescriptive sampling guidelines. ACM Trans. Graph. (2019)
30. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
31. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
32. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. (2022)
33. Niemeyer, M., Geiger, A.: GIRAFFE: representing scenes as compositional generative neural feature fields. In: CVPR (2021)
34. Qiu, J., Yang, Y., Wang, X., Tao, D.: Scene essence. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8322–8333 (2021)
35. Ren, J., Pan, L., Tang, J., Zhang, C., Cao, A., Zeng, G., Liu, Z.: Dreamgaussian4d: Generative 4d gaussian splatting. arXiv preprint arXiv:2312.17142 (2023)
36. Ren, J., Xie, K., Mirzaei, A., Liang, H., Zeng, X., Kreis, K., Liu, Z., Torralba, A., Fidler, S., Kim, S.W., et al.: L4gm: Large 4d gaussian reconstruction model. arXiv preprint arXiv:2406.10324 (2024)
37. Ren, Z., Agarwala, A., Russell, B.C., Schwing, A.G., Wang, O.: Neural volumetric object selection. In: CVPR (2022)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
39. Shen, Q., Yang, X., Wang, X.: Anything-3d: Towards single-view anything reconstruction in the wild. arXiv preprint arXiv:2304.10261 (2023)
40. Shen, Q., Yi, X., Wu, Z., Zhou, P., Zhang, H., Yan, S., Wang, X.: Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. arXiv preprint arXiv:2403.18795 (2024)
41. Stelzner, K., Kersting, K., Kosiorek, A.R.: Decomposing 3d scenes into objects via unsupervised volume segmentation. arXiv preprint arXiv:2104.01148 (2021)
42. Sun, C., Sun, M., Chen, H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: CVPR (2022)

43. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: WACV (2022)
44. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)
45. Tang, S., Pei, W., Tao, X., Jia, T., Lu, G., Tai, Y.W.: Scene-generalizable interactive segmentation of radiance fields. In: ACMMM (2023)
46. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In: 3DV (2022)
47. Vora, S., Radwan, N., Greff, K., Meyer, H., Genova, K., Sajjadi, M.S., Pot, E., Tagliasacchi, A., Duckworth, D.: Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. arXiv preprint arXiv:2111.13260 (2021)
48. Wizadwongsa, S., Phongthawee, P., Yenphraphai, J., Suwajanakorn, S.: Nex: Real-time view synthesis with neural basis expansion. In: CVPR (2021)
49. Wu, Z., Zhou, P., Yi, X., Yuan, X., Zhang, H.: Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9892–9902 (2024)
50. Yang, X., Wang, X.: Hash3d: Training-free acceleration for 3d generation. arXiv preprint arXiv:2404.06091 (2024)
51. Yang, Z., Yang, H., Pan, Z., Zhu, X., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv:2310.10642 (2023)
52. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. arXiv preprint arXiv:2309.13101 (2023)
53. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023)
54. Yi, T., Fang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. arXiv preprint arXiv:2310.08529 (2023)
55. Yi, X., Wu, Z., Shen, Q., Xu, Q., Zhou, P., Lim, J.H., Yan, S., Wang, X., Zhang, H.: Mvgamba: Unify 3d content generation as state space sequence modeling. arXiv preprint arXiv:2406.06367 (2024)
56. Yi, X., Wu, Z., Xu, Q., Zhou, P., Lim, J.H., Zhang, H.: Diffusion time-step curriculum for one image to 3d generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9948–9958 (2024)
57. Yin, Y., Xu, D., Wang, Z., Zhao, Y., Wei, Y.: 4dgen: Grounded 4d content generation with spatial-temporal consistency. arXiv preprint arXiv:2312.17225 (2023)
58. Yu, H., Guibas, L.J., Wu, J.: Unsupervised discovery of object radiance fields. In: ICLR (2022)
59. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
60. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation. In: ICCV (2021)
61. Zou, Z.X., Yu, Z., Guo, Y.C., Li, Y., Liang, D., Cao, Y.P., Zhang, S.H.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10324–10335 (2024)