Semantic & Neural Rendering & SLAM

Research Notes & Literature Review

Shuqi XIAO

July 9, 2024

Outline

Semantic 3DGS

Semantic 3DGS

Overview

Research Background

11/2023 - 12/2024	01/2024 - 04/2024	05/2024 - 06/2024
LEGS [1]	CoSSegGaussians [6]	GOI [10]
LangSplat [2]	Semantic Gaussians [7]	RT-GS2 [11]
Feature 3DGS [3]	Feature Splating [8]	SA-GS [12]
Gaussian Grouping [4]	CLIP-GS [9]	Fast-LGS [13]
SAGA [5]		

Figure 1: A timeline of Semantic 3DGS papers

Consensus

- What do we care about?
  - Accuracy; Consistency[1]; Efficiency; Interactivity[2].
- How can we achieve it?

Taxonomy of Methodology for SLAM

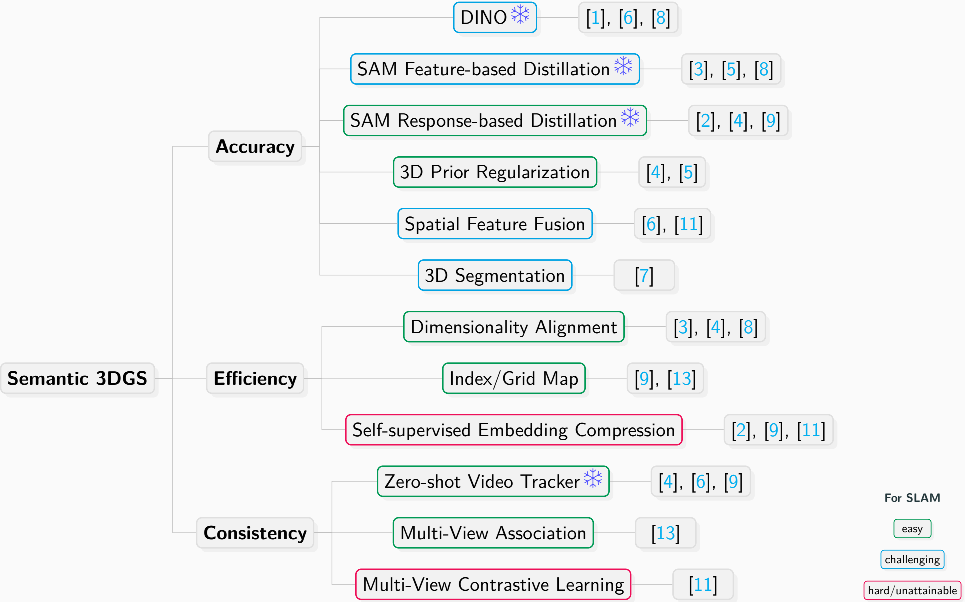


Figure 2: A taxonomy of Semantic 3DGS

Semantic 3DGS

Feature-3DGS

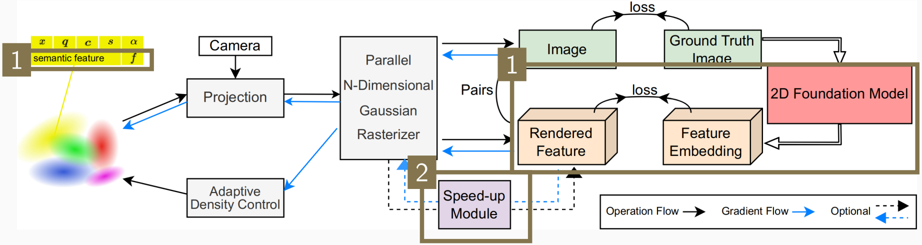


Figure 3: Overview of Feature 3DGS

1 Semantic Rendering Pipeline

Differentiable rendering of Gaussian-wise latent semantic features.

2 Speed-up module

Dimensionality Alignment.

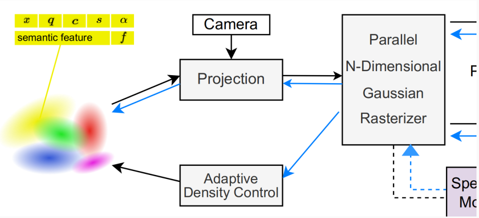
(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

To render semantic embeddings, i.e.

$$\mathbb{R}^{\overbrace{D}^{\text{dimension of semantic embedding}}} \times \{0,1,\cdots,\overbrace{N}^{\text{number of 3D Gaussians}}\} \times \underbrace{\{\mathcal{F}\}}_{\text{camera frames}} \mapsto \{0,1,\cdots,\underbrace{H}_{\text{image height}}\} \times \{0,1,\cdots,\underbrace{W}_{\text{image width}}\} \times \mathbb{R}^D \tag{1}$$

5 things,

1. representation
2. projection
3. blending
4. rasterization
5. inverse rendering



(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

1. Representation: 3D Gaussian augmented with a latent embedding.

$$\underbrace{\mathcal{G}}_{\text{optimizable attributes of a 3D Gaussian}}{}_{\underbrace{i}_{\in \mathbb{N},\ \text{index of 3D Gaussian}}} = \{\underbrace{\mathbf{x}}_{\in \mathbb{R}^3,\ \text{position}}, \underbrace{\mathbf{q}}_{\in \mathrm{SO}(3),\ \text{rotation}}, \underbrace{\mathbf{s}}_{\in \mathbb{R}^3,\ \text{scale}}, \underbrace{\alpha}_{\in [0,1],\ \text{opacity}}, \underbrace{\mathbf{c}}_{\in \mathbb{R}^{3n},\ \text{color}}, \underbrace{\mathbf{f}}_{\in \mathbb{R}^3,\ \text{semantic feature}}, \} \tag{2}$$

---

n: the maximal order of spherical harmonics to represent a color channel. In practice, $n = 4$.

(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

2. Projection: from 3D ellipsoids to 2D ellipses.

$$\underbrace{\mu_i}_{\in \mathbb{P}^2,\ \text{2D(image) mean}} = \underbrace{\pi}_{\text{projection}} \left(\underbrace{\mathbf{T}_{cw}}_{\in \mathrm{SE}(3),\ \text{camera pose}} \cdot \underbrace{\mu_w}_{\in \mathbb{P}^3,\ \text{3D(world) mean}} \right) \tag{3}$$

$$\underbrace{\Sigma_i}_{\in \mathbb{R}^{2\times 2},\ \text{2D(image) covariance}} = \underbrace{\mathbf{J}_\pi}_{\in \mathbb{R}^{2\times 3},\ \text{Jacobian of the linear approximation of } \pi} \underbrace{\mathbf{R}_{cw}}_{\in \mathrm{SO}(3),\ \text{rotation component of } \mathbf{T}_{cw}} \underbrace{\Sigma_w}_{\in \mathbb{R}^{3\times 3},\ \text{3D(world) covariance}} \mathbf{R}_{cw}^{\mathrm{T}} \mathbf{J}_\pi^{\mathrm{T}} \tag{4}$$

(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

Semantic Rendering Pipeline iii

3. Blending: α-blending of semantic embeddings.

number of the **sorted & visible** subset of 3D Gaussians

$$\mathbf{f}(h, w) = \sum_{i=1}^{N} T_i \, \alpha_i \, \mathbf{f}_i(h, w), \quad T_i = \prod_{j=1}^{i-1}(1 - \alpha_j) \tag{5}$$

semantic feature on pixel (h, w)

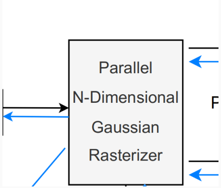
semantic feature of i-th Gaussian on pixel (h, w)

opacity of i-th Gaussian

background opacity for i-th Gaussian

(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

4. Rasterization: tiled and implemented in CUDA.



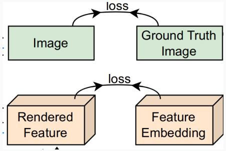
- Divide the screen space into tiles (CUDA thread blocks).
- Group the Gaussians by view frustum and tile index.
- Sort the Gaussians by front-to-back depth order.
- Blend each pixel within a tile in parallel (CUDA threads).

Figure 4: A brief summary of the tiled rasterization in 3DGS

(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

In practice, 16×16 blocks are used in rasterization.

5. Inverse rendering: guided by image-wise photometric loss.



$$\mathcal{L} = \mathcal{L}_{appearance} + \gamma \mathcal{L}_{semantics} \tag{6}$$

captured RGB image (GT)

rendered RGB image

$$\mathcal{L}_{appearance} = (1-\lambda)\mathcal{L}_1\left(\mathbf{C}, \hat{\mathbf{C}}\right) + \lambda \mathcal{L}_{D-SSIM}\left(\mathbf{C}, \hat{\mathbf{C}}\right) \tag{7}$$

$$\mathcal{L}_{semantics} = \mathcal{L}_1\left(\mathbf{F}, \hat{\mathbf{F}}\right) = \sum_{h=1}^{H}\sum_{w=1}^{W} \|\mathbf{f}(h,w) - \hat{\mathbf{f}}(h,w)\|_1 \tag{8}$$

inferred semantic feature map

rendered semantic feature map

(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

Motivation

Too **inefficient** to embed naively,

1. High dimension: latent features in large foundation models.
2. Large quantities: millions of Gaussians in a scene.

Solution

1. Compactness: to embed Gaussians with more compact vectors, $\dim = D' < D$.
2. Alignment: to align the dimensionalities using a lightweight decoder.

$D = 512$ in CLIP; $D = 256$ in SAM.

In practice, $D' = 128$.

Lightweight decoder: In practice, a 1×1 convolutional layer or a fully-connected layer.

(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

Summary

Limitations

1. Inefficiency
   - "Speed-up module" is not enough,

     $\mathrm{dim} = 128$ embedding for millions of Gaussians.
2. 3D Inconsistency & Inaccuracy
   - 2D foundation models are still 2D.

(CVPR Highlight, 2024) Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields

Semantic 3DGS

LangSplat

Overview

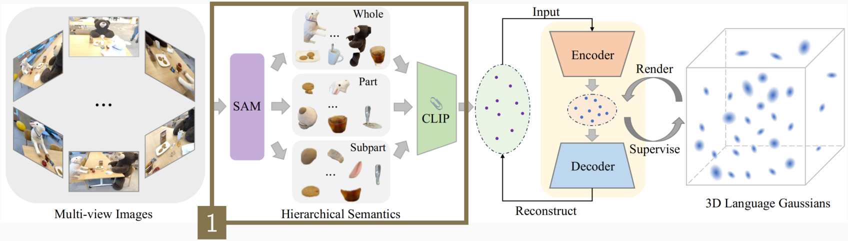


Figure 5: Overview of LangSplat

1 Accuracy: SAM outputs to enhance CLIP features.

- CLIP: image-aligned training leads to “point-ambiguity”.
- SAM: pixel-aligned & object-centered & multi-granularity.

(CVPR Highlight, 2024) LangSplat: 3D Language Gaussian Splatting

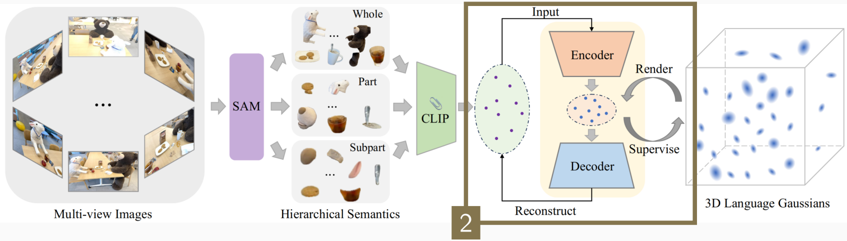


Figure 5: Overview of LangSplat

2 Efficiency: an auto-encoder to compress latent features.

- More complexity and better compression, compared with "speed-up module" in Feature 3DGS [3].

(CVPR Highlight, 2024) LangSplat: 3D Language Gaussian Splatting

In practice, the latent dimension is 3 in the auto-encoder.

Semantic 3DGS

CLIP-GS

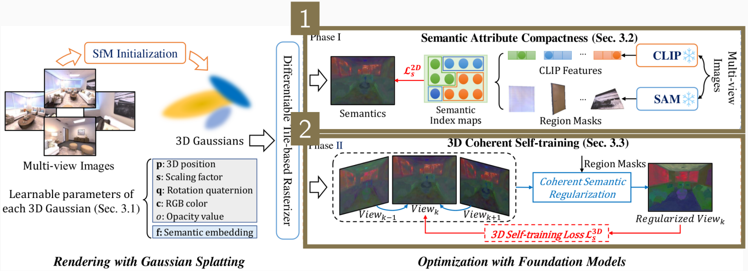


Figure 6: Overview of CLIP-GS

Key Insights

1. **Efficiency:** unify semantic features within an object by leveraging SAM.
2. **Consistency:** supervise consecutive frames by video segmentation.

(arXiv 04/2024) CLIP-GS: CLIP-informed Gaussian Splatting for Real-time and View-consistent 3D Semantic Understanding

3DGS SLAM

3DGS SLAM

Overview

Research Background

11/2023 - 12/2024

SplaTAM [14]
GS-SLAM [15]
Gaussian-SLAM [16]
MonoGS [17]

01/2024 - 04/2024

RTG-SLAM [18]

05/2024 - 06/2024

accepted to top conferences

References

References i

[1] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, *Language embedded 3d gaussians for open-vocabulary scene understanding*, Nov. 30, 2023. arXiv: 2311.18482[cs]. [Online]. Available: http://arxiv.org/abs/2311.18482 (visited on 06/08/2024) (cit. on pp. v, vi).

[2] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, *LangSplat: 3d language gaussian splatting*, Dec. 26, 2023. arXiv: 2312.16084[cs]. [Online]. Available: http://arxiv.org/abs/2312.16084 (visited on 02/23/2024) (cit. on pp. v, vi).

[3] S. Zhou, H. Chang, S. Jiang, *et al.*, *Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields*, Apr. 8, 2024. arXiv: 2312.03203[cs]. [Online]. Available: http://arxiv.org/abs/2312.03203 (visited on 05/22/2024) (cit. on pp. v, vi, xix).

[4] M. Ye, M. Danelljan, F. Yu, and L. Ke, *Gaussian grouping: Segment and edit anything in 3d scenes*, Dec. 1, 2023. arXiv: 2312.00732[cs]. [Online]. Available: http://arxiv.org/abs/2312.00732 (visited on 01/02/2024) (cit. on pp. v, vi).

[5] J. Cen, J. Fang, C. Yang, *et al.*, *Segment any 3d gaussians*, Dec. 1, 2023. arXiv: 2312.00860[cs]. [Online]. Available: http://arxiv.org/abs/2312.00860 (visited on 03/12/2024) (cit. on pp. v, vi).

[6] B. Dou, T. Zhang, Y. Ma, Z. Wang, and Z. Yuan, *CoSSegGaussians: Compact and swift scene segmenting 3d gaussians with dual feature fusion*, Jan. 30, 2024. arXiv: 2401.05925[cs]. [Online]. Available: http://arxiv.org/abs/2401.05925 (visited on 06/08/2024) (cit. on pp. v, vi).

[7] J. Guo, X. Ma, Y. Fan, H. Liu, and Q. Li, *Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting*, Mar. 22, 2024. arXiv: 2403.15624[cs]. [Online]. Available: http://arxiv.org/abs/2403.15624 (visited on 05/20/2024) (cit. on pp. v, vi).

References ii

[8] R.-Z. Qiu, G. Yang, W. Zeng, and X. Wang, *Feature splatting: Language-driven physics-based scene synthesis and editing*, Apr. 1, 2024. arXiv: 2404.01223[cs]. [Online]. Available: http://arxiv.org/abs/2404.01223 (visited on 06/08/2024) (cit. on pp. v, vi).

[9] G. Liao, J. Li, Z. Bao, *et al.*, *CLIP-GS: CLIP-informed gaussian splatting for real-time and view-consistent 3d semantic understanding*, Apr. 22, 2024. arXiv: 2404.14249[cs]. [Online]. Available: http://arxiv.org/abs/2404.14249 (visited on 05/20/2024) (cit. on pp. v, vi).

[10] Y. Qu, S. Dai, X. Li, *et al.*, *GOI: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane*, May 27, 2024. arXiv: 2405.17596[cs]. [Online]. Available: http://arxiv.org/abs/2405.17596 (visited on 06/08/2024) (cit. on p. v).

[11] M.-B. Jurca, R. Royen, I. Giosan, and A. Munteanu, *RT-GS2: Real-time generalizable semantic segmentation for 3d gaussian representations of radiance fields*, May 28, 2024. arXiv: 2405.18033[cs]. [Online]. Available: http://arxiv.org/abs/2405.18033 (visited on 06/08/2024) (cit. on pp. v, vi).

[12] B. Xiong, X. Ye, T. H. E. Tse, K. Han, S. Cui, and Z. Li, *SA-GS: Semantic-aware gaussian splatting for large scene reconstruction with geometry constrain*, May 28, 2024. arXiv: 2405.16923[cs]. [Online]. Available: http://arxiv.org/abs/2405.16923 (visited on 06/08/2024) (cit. on p. v).

[13] Y. Ji, H. Zhu, J. Tang, *et al.*, *FastLGS: Speeding up language embedded gaussians with feature grid mapping*, Jun. 3, 2024. arXiv: 2406.01916[cs]. [Online]. Available: http://arxiv.org/abs/2406.01916 (visited on 06/08/2024) (cit. on pp. v, vi).

[14] N. Keetha, J. Karhade, K. M. Jatavallabhula, *et al.*, *SplaTAM: Splat, track & map 3d gaussians for dense RGB-d SLAM*, Apr. 16, 2024. arXiv: 2312.02126[cs]. [Online]. Available: http://arxiv.org/abs/2312.02126 (visited on 05/20/2024) (cit. on p. xxiv).

[15] C. Yan, D. Qu, D. Wang, *et al.*, *GS-SLAM: Dense visual SLAM with 3d gaussian splatting*, Nov. 21, 2023. arXiv: 2311.11700[cs]. [Online]. Available: http://arxiv.org/abs/2311.11700 (visited on 12/26/2023) (cit. on p. xxiv).

[16] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, *Gaussian-SLAM: Photo-realistic dense SLAM with gaussian splatting*, Mar. 22, 2024. arXiv: 2312.10070[cs]. [Online]. Available: http://arxiv.org/abs/2312.10070 (visited on 03/27/2024) (cit. on p. xxiv).

[17] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, *Gaussian splatting SLAM*, Apr. 14, 2024. arXiv: 2312.06741[cs]. [Online]. Available: http://arxiv.org/abs/2312.06741 (visited on 05/20/2024) (cit. on p. xxiv).

[18] Z. Peng, T. Shao, Y. Liu, *et al.*, *RTG-SLAM: Real-time 3d reconstruction at scale using gaussian splatting*, May 8, 2024. DOI: 10.1145/3658233. arXiv: 2404.19706[cs]. [Online]. Available: http://arxiv.org/abs/2404.19706 (visited on 05/18/2024) (cit. on p. xxiv).