Gaussian Opacity Fields

Leonardo Mendonça (Scientific peer reviewer)
Mateus Barbosa (Archaeologist)
Esteban Wirth (Hacker)
Diana Aldana (PhD Researcher)

Institute for Pure and Applied Mathematics (IMPA)

Outline

Scientific peer reviewer

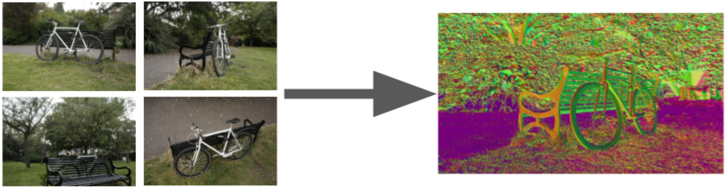
Leonardo Mendonça

Introduction

- The paper proposes an improvement on the method of 3D Gaussian Splatting for surface reconstruction and novel-view synthesis;
- Its central contribution is the development of the *Opacity field*, derived from the same images used for training the model;
- Improvements are also made to the 3DGS training process;
- The contributions are modular, which allows for their incorporation in different Gaussian Splatting pipelines.

Objectives

- Surface reconstruction from multiple views of a static scene;
- Creating a mesh that accurately describes the surface of the observed objects;
- Keeping a manageable mesh size and computing time compared to the state of the art;
- Extract accurate background meshes in unbounded scenes, a challenge which existing methods such as Neuralangelo [6] and 2DGS [3] still struggle with.



Methodology: Gaussian Primitives

- Similar to 3DGS [4], GOF models the scene through a set of K 3D Gaussian primitives $\mathcal{G}_1, \ldots, \mathcal{G}_K$;

- For a point $\mathbf{x}$ in world space, a Gaussian $\mathcal{G}_k$ is defined as follows:

$$\mathcal{G}_{\mathbf{k}}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{p_k})^T \cdot \mathbf{\Sigma_k}^{-1} \cdot (\mathbf{x}-\mathbf{p_k})} \tag{1}$$

  where $\mathbf{p_k}$ is the Gaussian's center and $\mathbf{\Sigma_k} \coloneqq \mathbf{R_k} \cdot \mathbf{S_k} \cdot \mathbf{S_k}^T \cdot \mathbf{R_k}^T$ is the variance matrix;

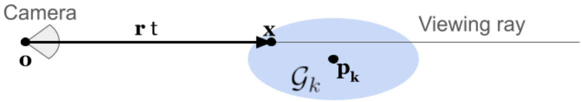
- $\mathcal{G}_{\mathbf{k}}$ are evaluated on a set of rays departing from each camera position;

- Unlike 3DGS, in GOF the Gaussians are not "splatted" on screen space.

Methodology: Ray-Splat Intersection

- GOF parameterizes a ray with origin at a point $\mathbf{o}$ with unit direction $\mathbf{r}$ as

$$\mathbf{x}(t) = \mathbf{o} + t\,\mathbf{r}; \tag{2}$$



- To find the ray-Gaussian intersection, GOF expresses the ray in the Gaussian $\mathcal{G}_k$'s coordinate system:

$$\mathbf{o_g} = \mathbf{S_k}^{-1} \cdot \mathbf{R_k}^T \cdot (\mathbf{o} - \mathbf{p_k}) \tag{3}$$

$$\mathbf{r_g} = \mathbf{S_k}^{-1} \cdot \mathbf{R_k}^T \cdot \mathbf{r} \tag{4}$$

$$\mathbf{x_g} = \mathbf{o_g} + t\,\mathbf{r_g} \tag{5}$$

* In the paper, the definitions of $\mathbf{o_g}$ and $\mathbf{r_g}$ have a minor mistake: the authors write $\mathbf{R_k}$ instead of $\mathbf{R_k}^T$.

Ray-Splat Intersection (cont.)

- It can be shown that the value of $\mathcal{G}_k$ along the ray becomes a 1D Gaussian:

$$\mathcal{G}_k(\mathbf{o}+\mathbf{r}t) = \mathcal{G}_k^{1D}(t) = e^{-\frac{1}{2}(\mathbf{x_g}^T \cdot \mathbf{x_g})} = e^{-\frac{1}{2}(\mathbf{r_g}^T \cdot \mathbf{r_g} t^2 + 2\mathbf{o_g}^T \cdot \mathbf{r_g} t + \mathbf{r_g}^T \cdot \mathbf{r_g} t^2)} \quad (6)$$

- (6) enables the computation of the depth t^* at which $\mathcal{G}_k^{1D}$ is maximized:

$$\mathcal{E}(\mathcal{G}_k, \mathbf{o}, \mathbf{r}) = \sup_{t \in \mathbb{R}} \mathcal{G}_k^{1D}(t) = \mathcal{G}_k^{1D}(t^*) \quad (7)$$

$$t^* = -\frac{\mathbf{o_g}^T \cdot \mathbf{r_g}}{\mathbf{r_g}^T \cdot \mathbf{r_g}}. \quad (8)$$

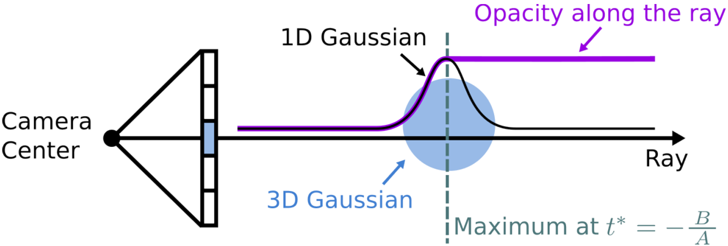
Rendering

- As in 3DGS, a ray's color is calculated through alpha-blending:

$$\mathbf{c}(\mathbf{o}, \mathbf{r}) = \sum_{k=1}^{K} \mathbf{c}_k \alpha_k \mathcal{E}(\mathcal{G}_k, \mathbf{o}, \mathbf{r}) \prod_{j=1}^{k-1} (1 - \alpha_j \mathcal{E}(\mathcal{G}_j, \mathbf{o}, \mathbf{r})), \tag{9}$$

  where $\mathbf{c}_k$ is an RGB color parameterized with spherical harmonics, and $\alpha_k \in [0, 1]$ is the weight, or opacity, of each Gaussian primitive;
- The K Gaussians are sorted using the depth.

Opacity Field



- For each Gaussian and viewing ray, one can define the *opacity along the ray*:

$$O_k(\mathcal{G}_k, \mathbf{o}, \mathbf{r}, t) := \begin{cases} \mathcal{G}_k^{1D}(t) & \text{, if } t \leq t^* \\ \mathcal{G}_k^{1D}(t^*) & \text{, if } t > t^* \end{cases} \tag{10}$$

Gaussian Opacity Field

- The definition extends to multiple Gaussians through alpha-blending:

$$O'(\mathbf{o},\mathbf{r},t) \coloneqq \sum_{k=1}^{K} \alpha_k O_k(\mathcal{G}_k,\mathbf{o},\mathbf{r},t) \prod_{j=1}^{k-1} \Big(1 - \alpha_j O_j(\mathcal{G}_j,\mathbf{o},\mathbf{r},t)\Big); \qquad (11)$$

- Finally, the *Gaussian Opacity Field* $O : \mathbb{R}^3 \mapsto [0,1]$ is defined as the minimum opacity of a point $\mathbf{x}$ along all viewing rays:

$$O(\mathbf{x}) \coloneqq \min_{(\mathbf{o},\mathbf{r})} \Big(\big\{O'(\mathbf{o},\mathbf{r},t)\colon \mathbf{x}(t) = \mathbf{o} + t\,\mathbf{r}\big\} \cup \{1\}\Big)\,; \qquad (12)$$

- This field measures how visible a given point is in relation to the training camera views: a point of opacity 1 is understood to be on the interior an object or outside the bounds of the scene;

* The paper writes $O(\mathbf{x}) \coloneqq \min_{(\mathbf{o},\mathbf{r})} O'(\mathbf{o},\mathbf{r},t)$, which is not rigorous.

Depth Distortion Regularization

- The Gaussian positions $\mathbf{p}_k$, rotations $\mathbf{R}_k$, scales $\mathbf{S}_k$ and weights α_k are optimized simultaneously through gradient descent;
- The paper adapts the depth distortion regularization from 2DGS [3] and Mip-NeRF [1]. For each Gaussian intersecting a ray $(\mathbf{o}, \mathbf{r})$, one defines:

$$\omega_i := \alpha_k \mathcal{E}(\mathcal{G}_i, \mathbf{o}, \mathbf{r}) \prod_{j=1}^{i-1} \Big(1 - \alpha_j \mathcal{E}(\mathcal{G}_j, \mathbf{o}, \mathbf{r})\Big), \tag{13}$$

  where $\mathcal{G}_i$ are the Gaussians sorted along the ray;
- Summing over every pair of Gaussians i, j along the ray results in the depth distortion loss:

$$\mathcal{L}_d := \sum_{i=1}^{K} \sum_{j=1}^{K} \omega_i \omega_j |t_i^* - t_j^*|; \tag{14}$$

- The gradient of (14) is only applied to t_i^*, which leads to grouping primitives closer together without affecting their α values;

* The paper might benefit from a more clear definition of (14), as well as a thorough explanation of its benefits.

Normal Consistency Regularization

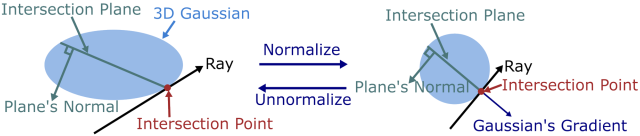
- 2DGS [3] introduced a successful approach for aligning planar Gaussians to the object's actual surface, in the form of an additional regularization term for each viewing ray:

$$\mathcal{L}_n := \sum_i \omega_i (1 - \mathbf{n_i}^T \cdot \mathbf{N}), \tag{15}$$

  where $\mathbf{n_i}$ is the normal of the i-th Gaussian along the ray and $\mathbf{N}$ is the normal estimated from the depth map;
- To adapt it to GOF, one must define the normal vector of a 3D Gaussian for a given viewing ray;

* The definition used for the normal (next slide) is not sufficiently explained or justified in paper.

Gaussian Normal Vector



- The text of the paper defines the normal as being $\mathbf{n_g} = -\mathbf{r_g}$ in the Gaussian coordinate system. Then, to get the normal in world coordinates, one "reverses" normalization and scaling: $\mathbf{n} = -\mathbf{R_k}^T \cdot \mathbf{S_k}^{-1} \cdot \mathbf{r_g}$;
- Following our corrected definition of $\mathbf{r_g}$, the normal direction becomes

$$\mathbf{n_g} = -\mathbf{R_k}^T \cdot \mathbf{S_k}^{-2} \cdot \mathbf{R_k}^T \cdot \mathbf{r}; \tag{16}$$

* Despite what is stated in the text, this is *not* equivalent to normalizing and denormalizing;
* The ablation section shows good quantitative results for this approach, but a better mathematical explanation is required.

Final Loss Function

- The loss function for optimizing the Gaussian primitives is

$$\mathcal{L} := \mathcal{L}_c + \alpha\mathcal{L}_d + \beta\mathcal{L}_n; \tag{17}$$

- $\mathcal{L}_c$ is an RGB reconstruction loss modified by a decoupled appearance approach taken from Vastgaussian [7];
- α and β are hyperparameters given in 2DGS [3].

Densification

- In 3DGS [4], Gaussian primitives are cloned or split when their view-space gradient reaches a certain threshold τ;
- This paper proposes replacing the gradient with a sum of norms of the individual pixel gradients:

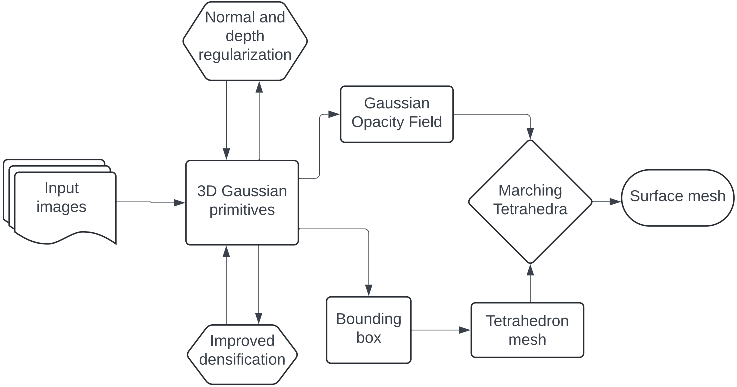
$$M := \sum_i \left\| \frac{dL}{d\mathbf{p_i}} \frac{d\mathbf{p_i}}{d\mathbf{x}} \right\|; \tag{18}$$

- This prevents the gradient from "cancelling out" between different views;
- The ablation shows this metric yields better results than the one used in 3DGS.

Surface Extraction

- Finally, the objects' surface is defined as a level set of opacity: $\{\mathbf{x} \colon O(\mathbf{x}) = 0.5\}$;
- Some points of interest are created on a bounding box of size 3σ around each Gaussian;
- A sparse mesh of tetrahedrons is created through Delaunay triangulation from these points;
- A Marching Tetrahedra algorithm [5] modified with a binary search is then used to extract a triangular mesh corresponding to this surface ;
- Creating the tetrahedrons only in the Gaussian's bounding boxes limits the computation time and the complexity of the final triangular mesh.

Surface Reconstruction Pipeline



Contributions

- The paper's main contributions to the state-of-the-art are:
  1. Applying the depth distortion and normal consistency regularizations from 2DGS [3] to 3D Gaussian optimization;
  2. Defining the *Gaussian Opacity Field* for delimiting the surface of an object or scene;
  3. Proposing an efficient method for extracting a mesh from a 3D Gaussian cloud, by adapting the Marching Tetrahedra [5] algorithm.
- These contributions can be implemented together, as proposed in the paper, or separately, as part of a different Gaussian Splatting pipeline;
- As an example, the authors apply (2) and (3) to the output of Mip-Splatting in order to extract a surface mesh.

Results: Surface Reconstruction

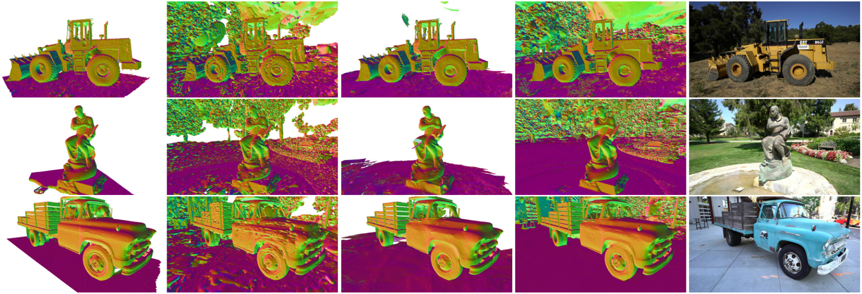


Figure: Normal maps from meshes generated for the *Tanks and Temples* dataset

Results: Surface Reconstruction

	Implicit			Explicit			
	NeuS	Geo-Neus	Neuralangelo	SuGaR	3DGS	2DGS	Ours
Barn	0.29	0.33	0.70	0.14	0.13	0.41	0.51
Caterpillar	0.29	0.26	0.36	0.16	0.08	0.23	0.41
Courthouse	0.17	0.12	0.28	0.08	0.09	0.16	0.28
Ignatius	0.83	0.72	0.89	0.33	0.04	0.51	0.68
Meetingroom	0.24	0.20	0.32	0.15	0.01	0.17	0.28
Truck	0.45	0.45	0.48	0.26	0.19	0.45	0.59
Mean	0.38	0.35	0.50	0.19	0.09	0.32	0.46
Time	>24h	>24h	>24h	>1h	14.3 m	15.5 m	24.2 m

Figure: F1-score and computational time for surface reconstruction on the *Tanks and Temples* scenes (only foreground objects)

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
implicit	NeRF [Mildenhall et al. 2021]	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49	> 12h
	VolSDF [Yariv et al. 2021]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86	>12h
	NeuS [Wang et al. 2021]	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84	>12h
	Neuralangelo [Li et al. 2023]	0.37	0.72	0.35	0.35	0.87	0.54	0.53	1.29	0.97	0.73	0.47	0.74	0.32	0.41	0.43	0.61	> 12h
explicit	3DGS [Kerbl et al. 2023]	2.14	1.53	2.08	1.68	3.49	2.21	1.43	2.07	2.22	1.75	1.79	2.55	1.53	1.52	1.50	1.96	11.2 m
	SuGaR [Guédon and Lepetit 2023]	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33	~ 1h
	GaussianSurfels [Dai et al. 2024]	0.66	0.93	0.54	0.41	1.06	1.14	0.85	1.29	1.53	0.79	0.82	1.58	0.45	0.66	0.53	0.88	6.7 m
	2DGS [Huang et al. 2024]	0.48	0.91	0.39	0.39	1.01	0.83	0.81	1.36	1.27	0.76	0.70	1.40	0.40	0.76	0.52	0.80	10.9 m
	Ours	0.50	0.82	0.37	0.37	1.12	0.74	0.73	1.18	1.29	0.68	0.77	0.90	0.42	0.66	0.49	0.74	18.4 m

Figure: Chamfer distance and average computational time for Surface Reconstruction on the *DTU* dataset

Results: Ablation

	Precision ↑	Recall ↑	F-score ↑
A. Mip-Splatting w/ TSDF	0.15	0.25	0.16
B. Mip-Splatting w/ GOF	0.40	0.33	0.36
C. Ours w/o GOF	0.37	0.45	0.39
D. Ours w/o normal consistency	0.41	0.35	0.37
E. Ours w/o decoupled appearance	0.49	0.39	0.43
F. Ours w/ minimal axis's normal	0.46	0.36	0.40
G. Ours w/o improved densification	0.52	0.39	0.44
H. Ours	0.54	0.42	0.46

Figure: Comparative performance on the *Tanks and Temples* dataset, with and without each of the strategies proposed in the paper for surface reconstruction

Strengths

- Innovative contributions that can be implemented to existing and future Gaussian Splatting methodologies;
- Best surface reconstruction among state-of-the-art explicit methods with computational cost significantly smaller than implicit methods;
- Unprecedented capacity for background mesh reconstruction in unbounded scenes;
- The definition of the Opacity field elegantly associates surface reconstruction with available view information.

Weaknesses

- Some equations are not written in a rigorous manner or have minor mistakes;
- The definition of the Gaussian normal is unclear in the paper, and the reason it works is only justified empirically;
- The desired properties of the Opacity field that align its levelsets with object surfaces are not mathematically proved;
- There is no quantitative experiment backing the paper's claim that the meshes it generates are compact;
- We have found it difficult to run the source coded provided by the authors in order to replicate the paper's results.

Conclusion

- The paper introduces a novel strategy for surface reconstruction employing the Opacity field and Marching Tetrahedra;
- There is extensive experimentation on well-established datasets showing an improvement in surface reconstruction over the state-of-the-art;
- More attention should be paid to the mathematical aspects of the methodology, towards clarity and rigor;
- All things considered, we believe this paper should be **accepted**.

Archaeologist

Mateus Barbosa

Context of GOF

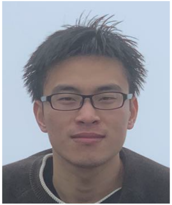
Gaussian Opacity Fields

Efficient Adaptive Surface Reconstruction in Unbounded Scenes

SIGGRAPH ASIA 2024 (Journal Track)

Zehao Yu[1,2] Torsten Sattler[3] Andreas Geiger[1,2]

[1]University of Tübingen [2]Tübingen AI Center [3]Czech Technical University in Prague



Context of GOF

Some of the authors' previous collaborations include:

- **MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction.**

  Zehao Yu, Songyou Peng, Michael Niemeyer, **Torsten Sattler**, **Andreas Geiger**. 2022

- **Mip-Splatting: Alias-free 3D Gaussian Splatting.**

  Zehao Yu, Anpei Chen, Binbin Huang, **Torsten Sattler** and **Andreas Geiger**. 2024

- **2D Gaussian Splatting for Geometrically Accurate Radiance Fields.**

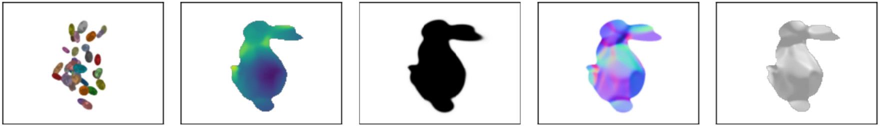
  Binbin Huang, **Zehao Yu**, Anpei Chen, **Andreas Geiger** and Shenghua Gao. 2024

Context of GOF

- Other attempts on surface reconstruction with Gaussians had limited results.
- SuGaR used Poisson reconstruction. 2DGS used 2D disks and TSDF fusion.
- What sets GOF apart from other surface extraction methods is that it constructs opacity fields using a ray-tracing approach that relies on a method of intersecting the ray and the Gaussian.
- They attribute the intersection definition to Keselman and Hebert 2022.

Previous paper

Leonid Keselman and Martial Hebert. 2022. Approximate Differentiable Rendering with Algebraic Surfaces. In European Conference on Computer Vision (ECCV).

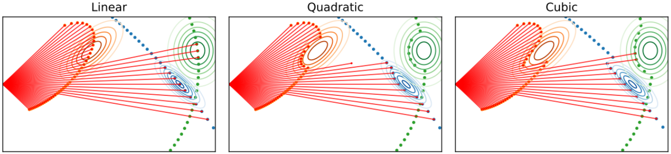


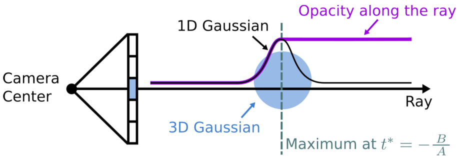
Components   Depth   Alpha   Normals   Phong

- Develops a ray-tracing formulation for Gaussians which implicitly defines the surface.
- To do this they develop a way of defining intersections between Gaussians and rays, and a way of combining intersections across all Gaussians.

Previous paper

- Keselman and Hebert 2022 proposes to approximate the intersection of ray with 3D Gaussian as a point where the 3D Gaussian's contribution peaks.
- GOF adopts this suggestion.





Reception of the paper

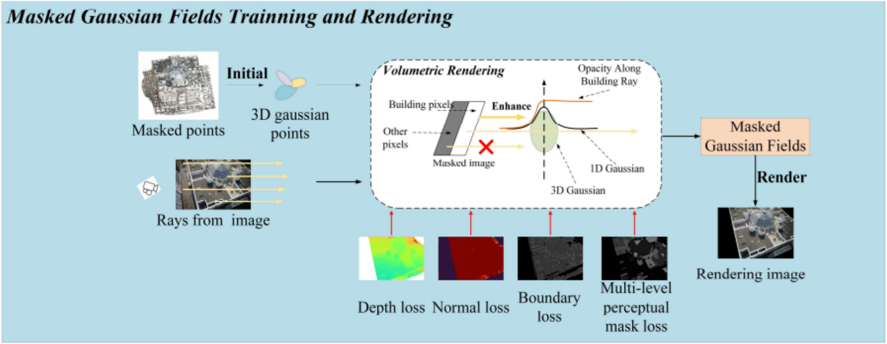
- Currently has 34 citations.
- 10/34 actually incorporate elements of the paper.
- 6/10 adopt only the improved densification.

Next paper

MGFs: Masked Gaussian Fields for Meshing Building based on Multi-View Images

Tengfei Wang, Zongqian Zhan*, Rui Xia, Linxia Ji, Xin Wang*

School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, Wuhan 430072, People's Republic of China



Next paper

Human 3Diffusion: Realistic Avatar Creation via Explicit 3D Consistent Diffusion Models

Yuxuan Xue[1,2] **Xianghui Xie**[1,2,3] **Riccardo Marin**[1,2] **Gerard Pons-Moll**[1,2,3]

[1]University of Tübingen [2] Tübingen AI Center
[3]Max Planck Institute for Informatics, Saarland Informatics Campus
https://yuxuan-xue.com/human-3diffusion/

- Creates 3D Gaussian Splats of realistic avatars with high-fidelity geometry and texture.
- Uses GOF to generate depth map.
- But performs volumetric TSDF fusion to extract the meshes.

Hacker

Esteban Wirth

Overall Structure of Code

- Initializes points in 3 dimensions with previous runs of COLMAP
- Trains RGB and depth model using 3DGS and 2DGS loss functions and model. Also uses densification. (Note here Mip-Splatting is also used with the 3D-filter)
- Finally the code extracts the mesh from the depth calculations using a binary search over the tetrahedra vertices and renders the scene.

Training

- Loss functions are taken from 3DGS and 2DGS. Hyperparamters for depth distortion and normal consistency are hard coded as 100 and 0.05. Applies them after 15000 iterations.
- Cameras are taken randomly from the list of training cameras. It goes through every camera before repeating itself.
- It has hard-coded to use around 30% of the iterations a high-resolution camera.
- It densifies every 100 iterations. Begining at iteration 500 up until iteration 15000.
- Reset the opacity every 3000 iterations.
- Increases the Spherical Harmonics maximum degree every 1000 iterations

Mesh Reconstruction

- Using binarysearch over the mesh and the Gaussian Opacity Field they extract the surface

```
def evaluage_alpha(points, views, gaussians, pipeline, background, kernel_size, return_color=False):
    final_alpha = torch.ones((points.shape[0]), dtype=torch.float32, device="cuda")
    if return_color:
        final_color = torch.ones((points.shape[0], 3), dtype=torch.float32, device="cuda")

    with torch.no_grad():
        for _, view in enumerate(tqdm(views, desc="Rendering progress")):
            ret = integrate(points, view, gaussians, pipeline, background, kernel_size=kernel_size)
            alpha_integrated = ret["alpha_integrated"]
            if return_color:
                color_integrated = ret["color_integrated"]
                final_color = torch.where((alpha_integrated < final_alpha).reshape(-1, 1), color_integrated, final_color)
            final_alpha = torch.min(final_alpha, alpha_integrated)

        alpha = 1 - final_alpha
    if return_color:
        return alpha, final_color
    return alpha
```

Figure: Code to construct GOF

Results of the Experiment: Original model vs No Regularization 7000 steps

Figure: Original hyperparameters

Figure: Hyperparameters set to 0

Results of the Experiment: Original model vs No Regularization 30000 steps

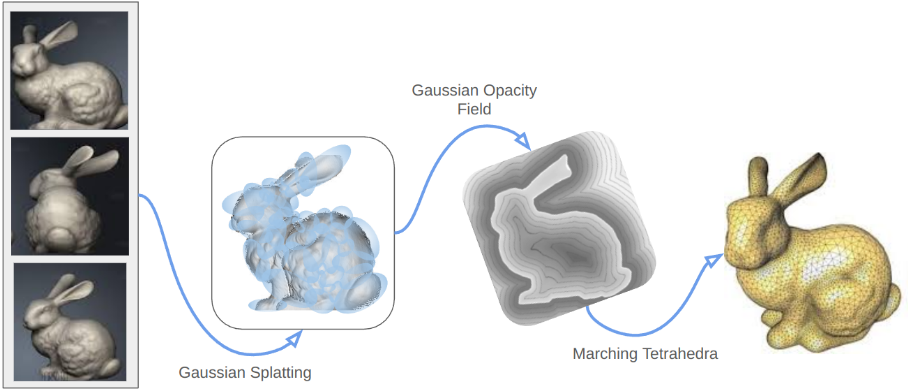
Figure: Original hyperparameters

Figure: Hyperparameters set to 0

Phd Researcher
Diana Aldana

Summary

GOF prospective title: Gaussian SDF for multiview scenes



Characteristics

Advantages

- Fast runtime.
- Background reconstruction.
- Accuracy against explicit methods.

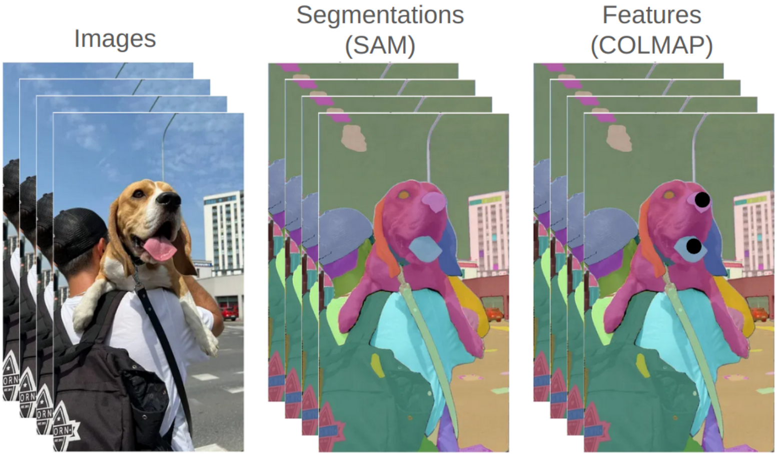
Limitations

- Accuracy against implicit methods.
- Delaunay Triangulation Efficiently
- Opacity evaluation optimization.
- View dependant appearance modeling.

Neuralangelo [Li et al. 2023] SuGar [Guédon and Lepetit 2023] 2DGS [Huang et al. 2024] Ours GT

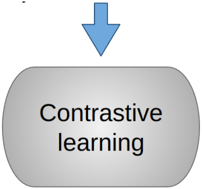
Title: Segmentation of multiview unbounded scenes via Gaussians

Previous to training



Gaussian grouping

Clusterize the gaussians such that each group corresponds to a different object.

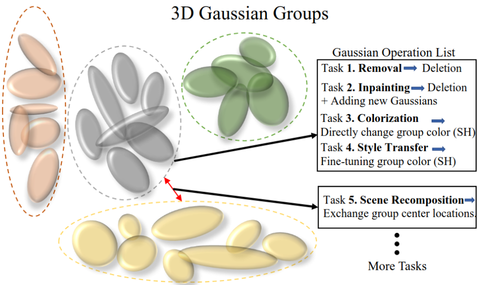


Implementation ideas

Possible contrastive learning loss:

$$\mathcal{L}_{contra} = -\frac{1}{|\Omega|} \sum_{\Omega_j \in \Omega} \sum_{u \in \Omega_j} \log \frac{\exp\left(\mathtt{sim}(\mathbf{F}_u, \bar{\mathbf{F}}_j)\right)}{\sum_{\Omega_l \in \Omega} \exp\left(\mathtt{sim}(\mathbf{F}_u, \bar{\mathbf{F}}_j)\right)}$$

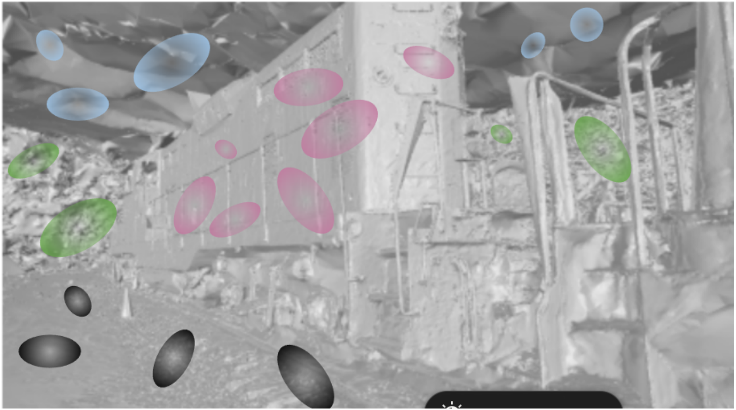
where Ω is the set of pixels, Ω_j is the set of pixels of the j-th group, $\mathtt{sim}$ is a similarity kernel and $\bar{\mathbf{F}}_j$ are the features of the centroid gaussian of group j.



Operations over the gaussians:

- Change of groups.
- Densify on different groups.

After training



Advantages:

- Implicit/explicit object representation.
- Speed up mesh reconstruction.
- Operations over a single scene's object.

Related references

- Object-Aware Lifting For 3d Scene Segmentation In Gaussian Splatting:
  - Doesn't show results for unbounded scenes
- Gaussian grouping: Segment and edit anything in 3d scenes [8]:
  - Uses gaussians to modify objects on the scene
- CoSSegGaussians: Compact and Swift Scene Segmenting 3D Gaussians with Dual Feature Fusion [2]:
  - 3D segmentation.
  - Erasure/translation of objects. Their work presents many artefacts.
- Hugs: Holistic urban 3d scene understanding via gaussian splatting (CVPR) [9]:
  - Dynamic scene decomposition on traffic.
  - Use optical flow to translate/rotate objects with precision.
  - Don't show the reconstruction of their geometry.

Bibliography I

[1] Jonathan T Barron et al. "Mip-nerf 360: Unbounded anti-aliased neural radiance fields". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5470–5479.

[2] Bin Dou et al. "CoSSegGaussians: Compact and Swift Scene Segmenting 3D Gaussians with Dual Feature Fusion". In: *arXiv preprint arXiv:2401.05925* (2024).

[3] Binbin Huang et al. "2d gaussian splatting for geometrically accurate radiance fields". In: *ACM SIGGRAPH 2024 Conference Papers*. 2024, pp. 1–11.

[4] Bernhard Kerbl et al. "3D Gaussian Splatting for Real-Time Radiance Field Rendering.". In: *ACM Trans. Graph.* 42.4 (2023), pp. 139–1.

[5] Jonas Kulhanek and Torsten Sattler. "Tetra-nerf: Representing neural radiance fields using tetrahedra". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 18458–18469.

[6] Zhaoshuo Li et al. "Neuralangelo: High-fidelity neural surface reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 8456–8465.

Bibliography II

[7] Jiaqi Lin et al. "Vastgaussian: Vast 3d gaussians for large scene reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 5166–5175.

[8] Mingqiao Ye et al. "Gaussian grouping: Segment and edit anything in 3d scenes". In: *arXiv preprint arXiv:2312.00732* (2023).

[9] Hongyu Zhou et al. "Hugs: Holistic urban 3d scene understanding via gaussian splatting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 21336–21345.

Thank you!