GScream: Learning 3D Geometry and Feature Consistent Gaussian Splatting for Object Removal

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

- Authors: Yuxin Wang, Qianyi Wu, Guofeng Zhang, Dan Xu

- Venue: **ECCV 2024**
- TA – Hrishikesh Hemke
- Guided by - Prof. C. Krishna Mohan

Presented By-

CS24MTECH14006 Gulshan Hatzade

Presentation Outline

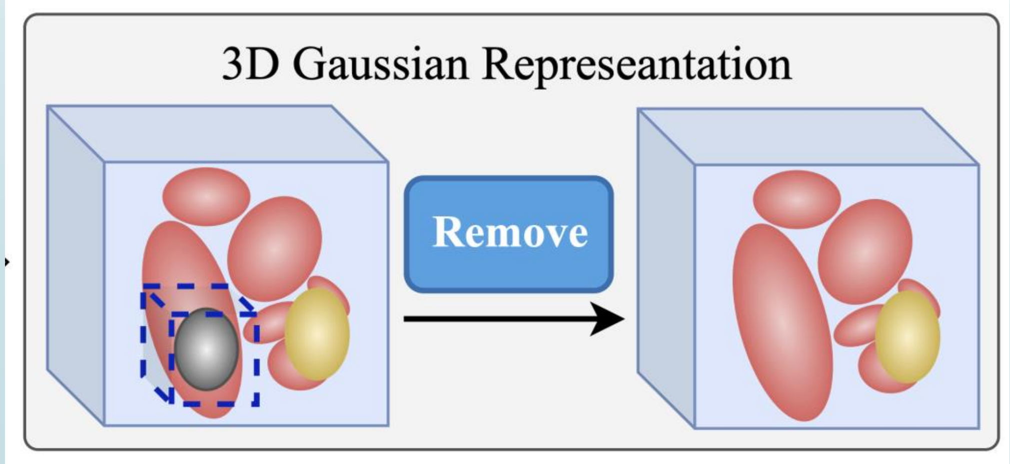
Motivation

- Need for realistic 3D scene editing
- Gap: 2D in-painting vs. 3D object removal
- Demand for real-time, high-quality view synthesis
- Benefits of explicit representations (Gaussian Splatting)

Problem Statement

- Objective: Remove objects from 3D scenes by updating radiance field without the object
- Preserve:
  - Geometric structure
  - Texture consistency

Ilustration of the Object Removal using 3D Gaussian Representations



Challenges

- Discrete Gaussian primitives can introduce geometric noise
- Maintaining consistent textures across multiple views
- 2D in-painting inadequate when extended to 3D
- Limitations of NeRF (slow training/rendering)

Existing Methods & Limitations

- **NeRF (Neural Radiance Field) based Approaches**
  - *Examples:*
    - **SPIn-NeRF [1]:** Combines multi-view segmentation and inpainting within a NeRF pipeline for realistic object removal.
    - **OR-NeRF [2]:** Uses multi-view segmentation cues to remove objects in 3D scenes, but with speed and geometry limitations
    - **View-Sub [3]:** Applies a reference-guided approach for inpainting in NeRF, though it may not achieve uniform reconstruction across views.
  - *Strength:* Excellent visual quality
  - *Weakness:* Computationally expensive and inconsistent geometric reconstruction

References mentioned-

[1] Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: CVPR (2023)

[2] Or-nerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields. arXiv preprint arXiv:2305.10503 (2023)

[3] Reference-guided controllable inpainting of neural radiance fields. In: ICCV (2023)

Existing Methods & Limitations

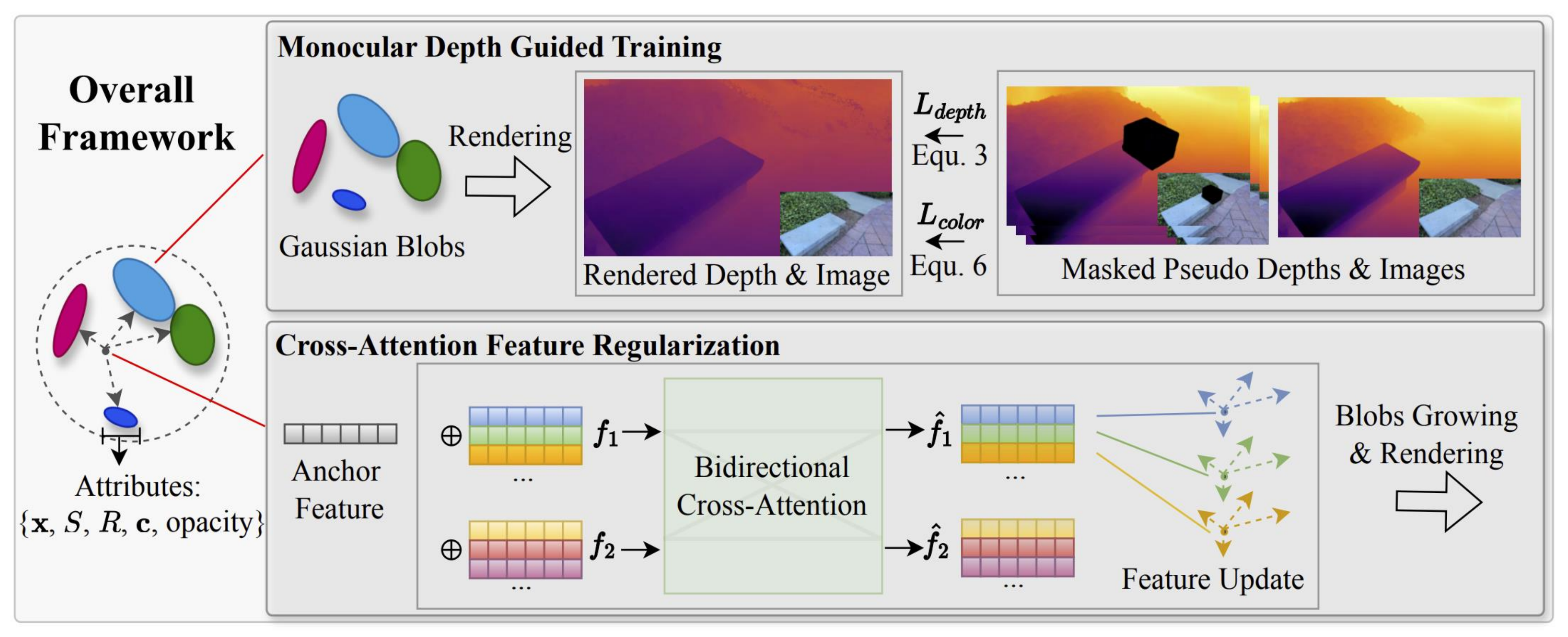
- **2D in-painting** methods **[4]**:
  - Effective for single images.
  - Lack multi-view consistency
- The gap: No integrated approach that ensures both geometric and texture restoration efficiently.

Reference mentioned-

[4] Nerf-in: Free-form nerf inpainting with rgb - d priors IEEE (CG&A)- 2023

Overview of GScream Methodology

- Utilizes 3D Gaussian Splatting for explicit scene representation.
- Two core components:
  - Monocular Depth-Guided Training
  - Cross-Attention Feature Regularization



Monocular Depth-Guided Training

- **Goal:**

  Use a depth map from one image to place blobs correctly in 3D.

- **Process:**

  Generate a depth map using a depth estimator.

  Align the 3D blobs so that their distances match the depth map.

- **Outcome:**

  Ensures the scene has accurate 3D **shapes** and **distances** even after object removal.

3D Gaussian Representation

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

- μ: Center of the Gaussian blob
- Σ: Covariance matrix (captures scale & orientation)

Volume Rendering Equations

- **Color Rendering:** $$\hat{C} = \sum_{k=1}^{K} c_k \, \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j)$$

- **Depth Rendering:** $$\hat{D} = \sum_{k=1}^{K} t_k \, \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j)$$

Where,

K : Total number of Gaussians sampled along that ray.

c_k : Color of the kth Gaussian

a_k : Opacity of the k-th Gaussian

t_k : The depth of the k-th Gaussian

Loss Functions

- Depth Loss: $$\mathcal{L}_{\text{depth}} = \frac{1}{HW}\sum M_i' \|(w\hat{D}_i + q) - D_i\|$$
- Total Variation Loss: $$\mathcal{L}_{\text{tv}} = \frac{1}{N}\sum M_i' \|\nabla((w\hat{D}_i + q) - D_i))\|$$
- Color Loss: $$\mathcal{L}_{\text{color}} = \frac{1}{HW}\sum M_i' \Big[(1-\lambda_{\text{ssim}})\|\hat{C}_i - I_i\| + \lambda_{\text{ssim}}\, \text{SSIM}(\hat{C}_i, I_i)\Big]$$

- Optimization Objective: $$\mathcal{L}_{\text{total}} = \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{tv}}\mathcal{L}_{\text{tv}} + \mathcal{L}_{\text{color}}$$

where, **H,W**: The height and width of the image

i : Index over pixels.

M\`$_i$: A weight applied to pixel i

D^$_i$: The rendered depth at pixel i computed via Gaussian splatting.

D$_i$: The depth value estimated by an external (monocular) depth estimator.

w: A scaling factor to align the rendered depth with the estimated depth.

q: A shift (offset) factor to align the two depths.

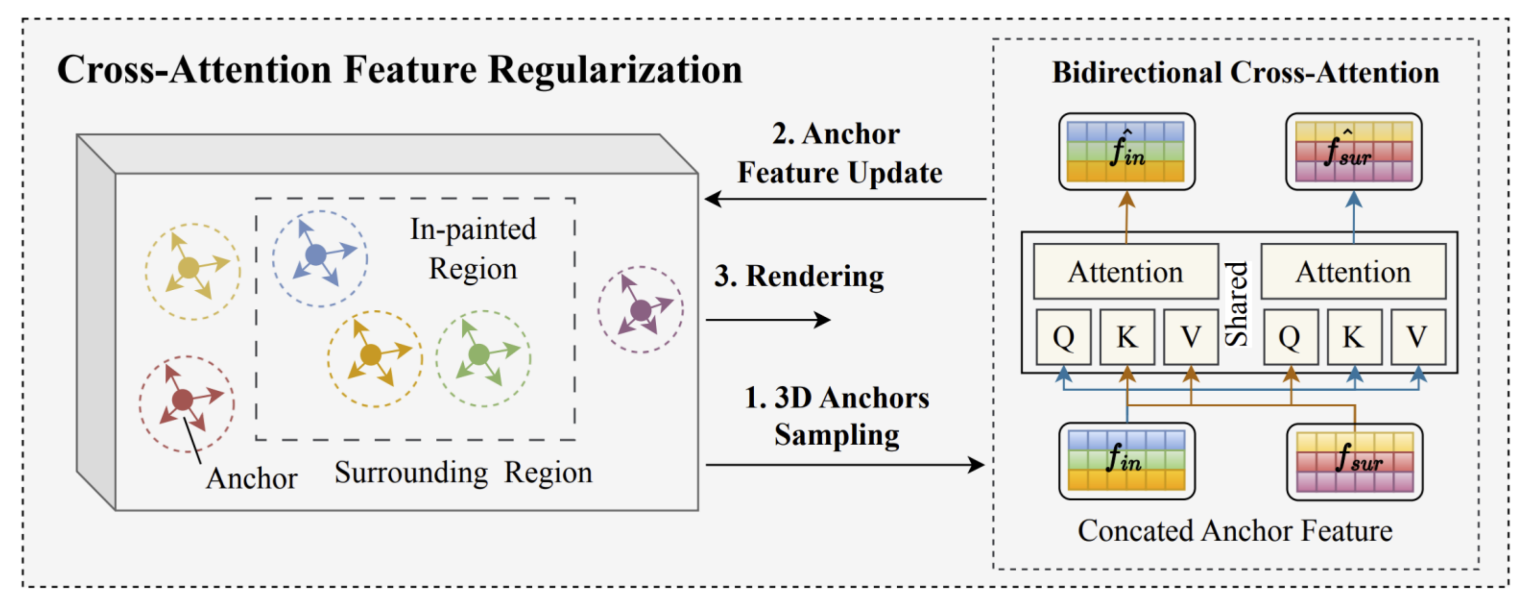
N, ∇ : A normalization factor & The gradient operator

I$_i$: The ground-truth (or target) color at pixel I

λ_{ssim} : A weighting factor

Cross-Attention Feature Regularization

- Sample 3D Gaussian anchors from:
  - In-painted (masked) regions
  - Surrounding visible regions
- Apply bidirectional cross-attention:
  - Feature exchange between regions
- Outcome: Improved texture consistency



Cross-Attention Mechanism

- Equation: $$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$
- Update equations: $$\hat{f}_{in} = \text{Attention}(f_{in}, f_{sur}, f_{sur}),$$

$$\hat{f}_{sur} = \text{Attention}(f_{sur}, f_{in}, f_{in})$$

where,

Q: Query matrix- features that need to be updated

K: Key matrix - reliable features that the query can attend.

V: Value matrix - features that will be used to update the queries.

d_k : The dimensionality of the key vectors

Softmax : Normalizes the scores into a probability distribution, which weighs the contribution of each key.

Experiments & Results

Datasets: SPIN-NeRF

- Contains 10 forward-facing scenes.
- Approximately 100 multi-view images per scene.
  - Around 60 with the target object
  - Around 40 object is physically removed
- Includes annotated object masks for segmentation and inpainting

IBRNet (LLFF subset)

- Comprises a six-scene subset of the LLFF dataset(appox. 35 images per scene).
- Scenes are captured with significant parallax, offering a diverse range of viewpoints.
- Does not include object-level annotations and used for novel view synthesis.

Methods	PSNR ↑	masked PSNR ↑	SSIM ↑	masked SSIM ↑	LPIPS↓	masked LPIPS ↓	FID ↓	Training Time ↓
SPIn-NeRF [23]	20.18	15.80	0.46	**0.21**	0.47	0.58	58.78	~ 3.0h
OR-NeRF [41]	20.32	15.74	0.54	**0.21**	0.35	0.56	38.69	~ 6.0h
View-Sub [22]	-	-	-	-	-	**0.45***	-	-
GScream (Ours)	**20.49**	**15.84**	**0.58**	0.21	**0.28**	0.54	**36.72**	~ **1.2h**

Qualitative Comparison of Object Removal Approaches

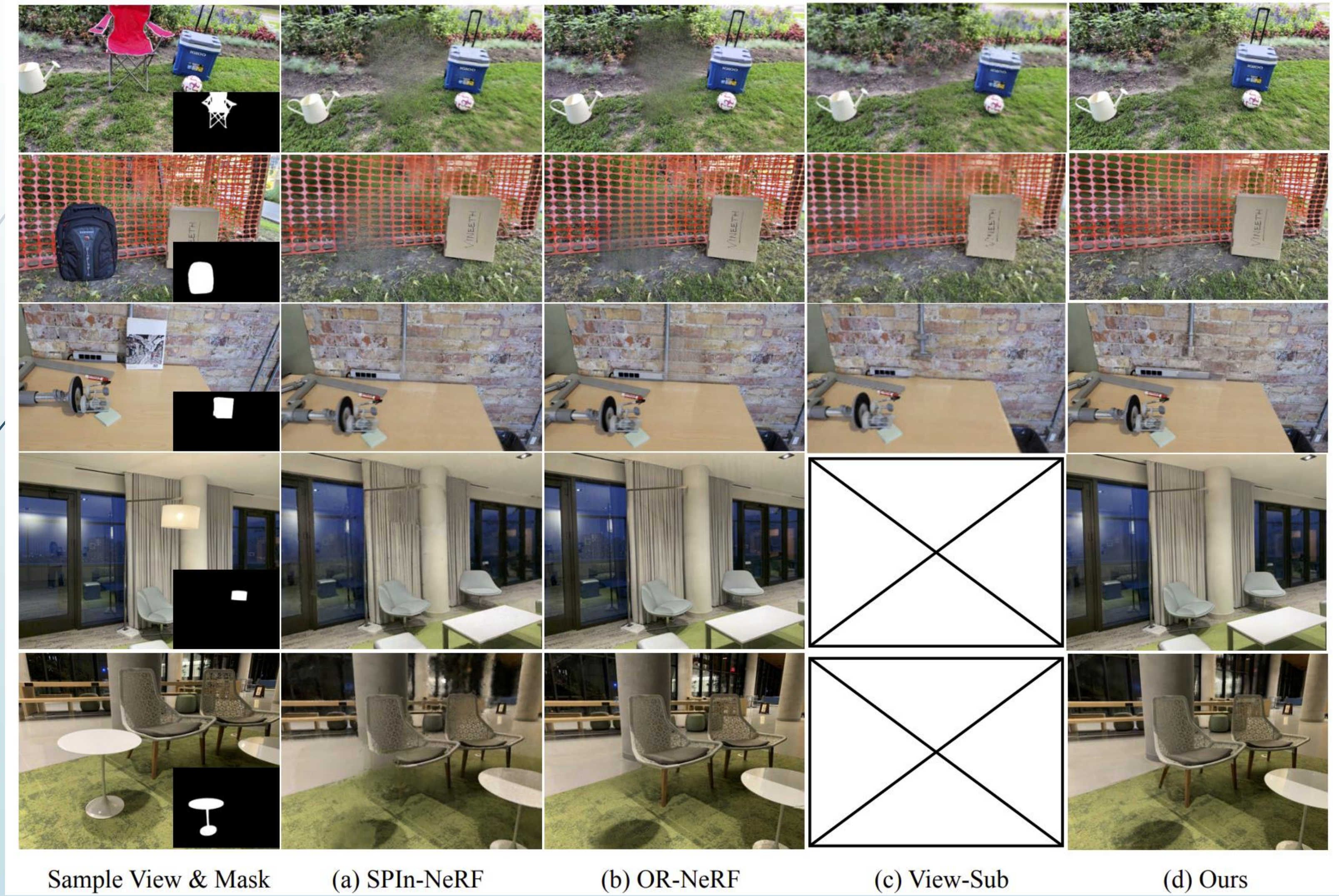


Fig. : Qualitative results compared with the most representative object removal approaches

Quantitative Comparison of Different Variants of Proposed Method

Variants	PSNR ↑	masked-PSNR ↑	SSIM ↑	masked-SSIM ↑	LPIPS↓	masked-LPIPS ↓
GScream w/o Cross-Attn & Mono-Depth	20.12	14.87	0.58	0.19	**0.26**	0.56
GScream w/o Cross-Attn	20.47	15.63	0.58	0.20	**0.26**	**0.50**
GScream (Our Full Model)	**20.49**	**15.84**	**0.58**	**0.21**	0.28	0.54

Conclusion

- GScream delivers robust object removal in 3D scenes
- Overcomes shortcomings of NeRF & 2D in-painting methods
- Opens up possibilities for real-time 3D editing in various applications.

Limitations of the Proposed Method

- **Dependence on Accurate Object Masks:** Relies on multi-view object masks, which may require manual intervention or lead to errors if automatically generated.
- **External Module Dependencies:** Uses separate monocular depth estimation and 2D in-painting models; errors in these can propagate into the final reconstruction.
- **Computational Considerations:** Although more efficient than implicit methods (e.g., NeRF), managing and optimizing a large number of Gaussian primitives still presents challenges.

Future Work

- **Joint Learning for Mask and Depth Estimation:** Integrate object mask generation and depth estimation into a unified, end-to-end framework to reduce external dependency.
- **Extension to Dynamic Scenes**
- **Application in Autonomous Vehicles**
  - Improve **3D scene understanding** by handling **occlusions** in real-time for enhanced perception (tree branch covering part of pedestrian)
  - Enhance **depth estimation** by integrating LiDAR and camera data.
  - Enable **efficient, real-time 3D scene reconstruction** for safe navigation in complex driving environments.
  - Extend the method for **removing transient obstacles** (e.g rain, reflections, sensor noise) while preserving critical dynamic objects for better decision-making.

References

1. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: CVPR (2023)
2. Or-nerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields. arXiv preprint arXiv:2305.10503 (2023)
3. Reference-guided controllable inpainting of neural radiance fields. In: ICCV (2023)
4. Nerf-in: Free-form nerf inpainting with rgb - d priors IEEE (CG&A)- 2023
5. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022)
6. Swift and controllable 3d editing with gaussian splatting. In: CVPR (2024)

Thank You