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

- 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

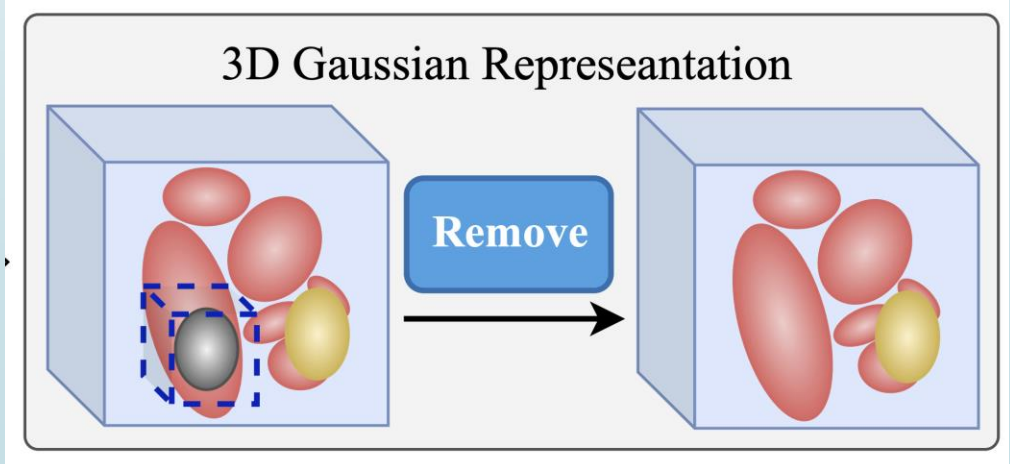
Presentation Outline

- Problem Statement
- Related Work & Limitations
- Methodology (Depth Guidance & Cross-Attention)
- Experiments & Results
- Limitations of the Proposed Method
- Reproduced Results
- Novelties
- Conclusion
- References

Problem Statement

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

Illustration of the Object Removal using 3D Gaussian Representations



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-

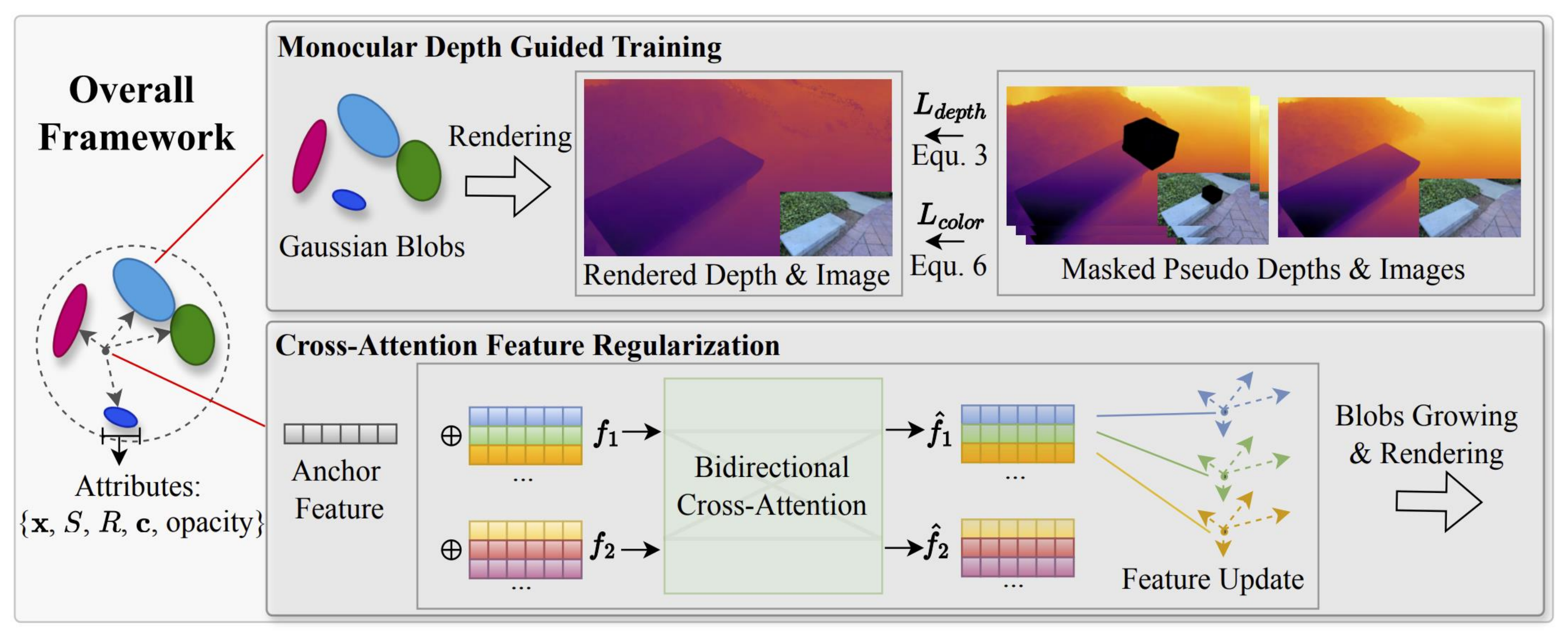
[1] Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In: CVPR (2023)

[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)

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

$D^\wedge_i$: 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.

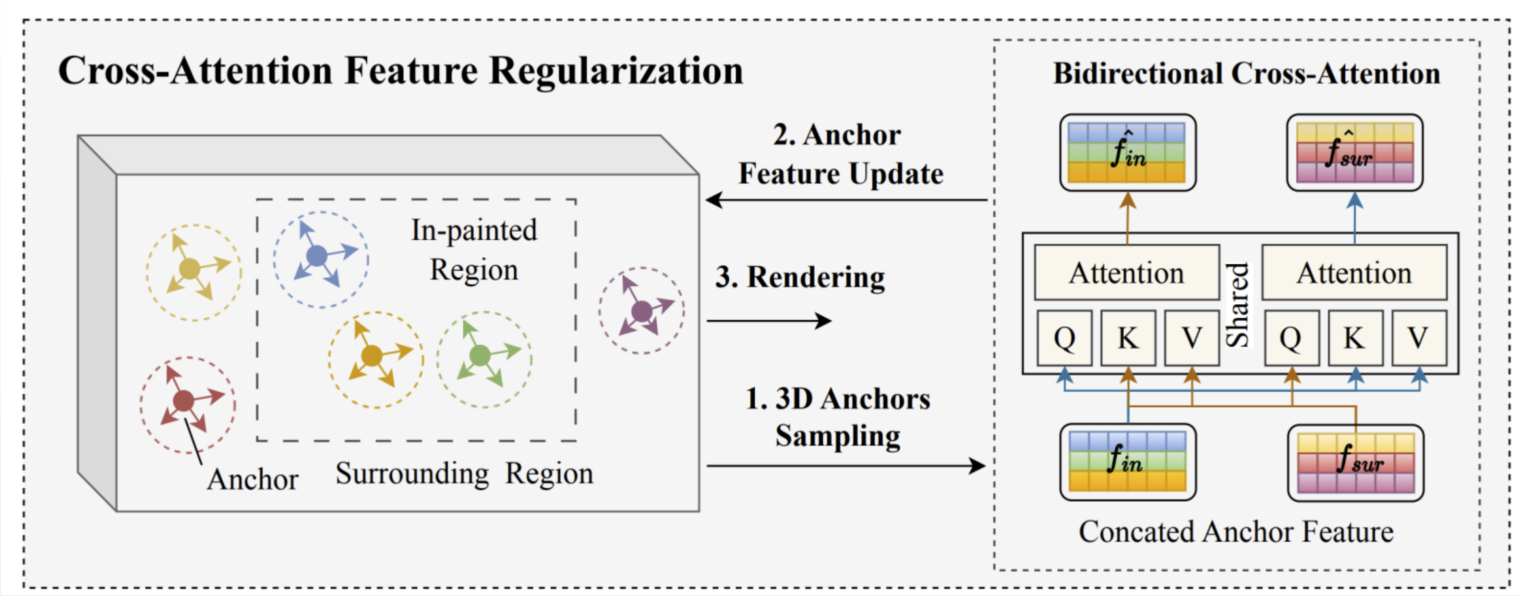
N, ∇ : A normalization factor & The gradient operator

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:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Updated 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

Training Iterations	30,000
GPU	NVDIA RTX 3050

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

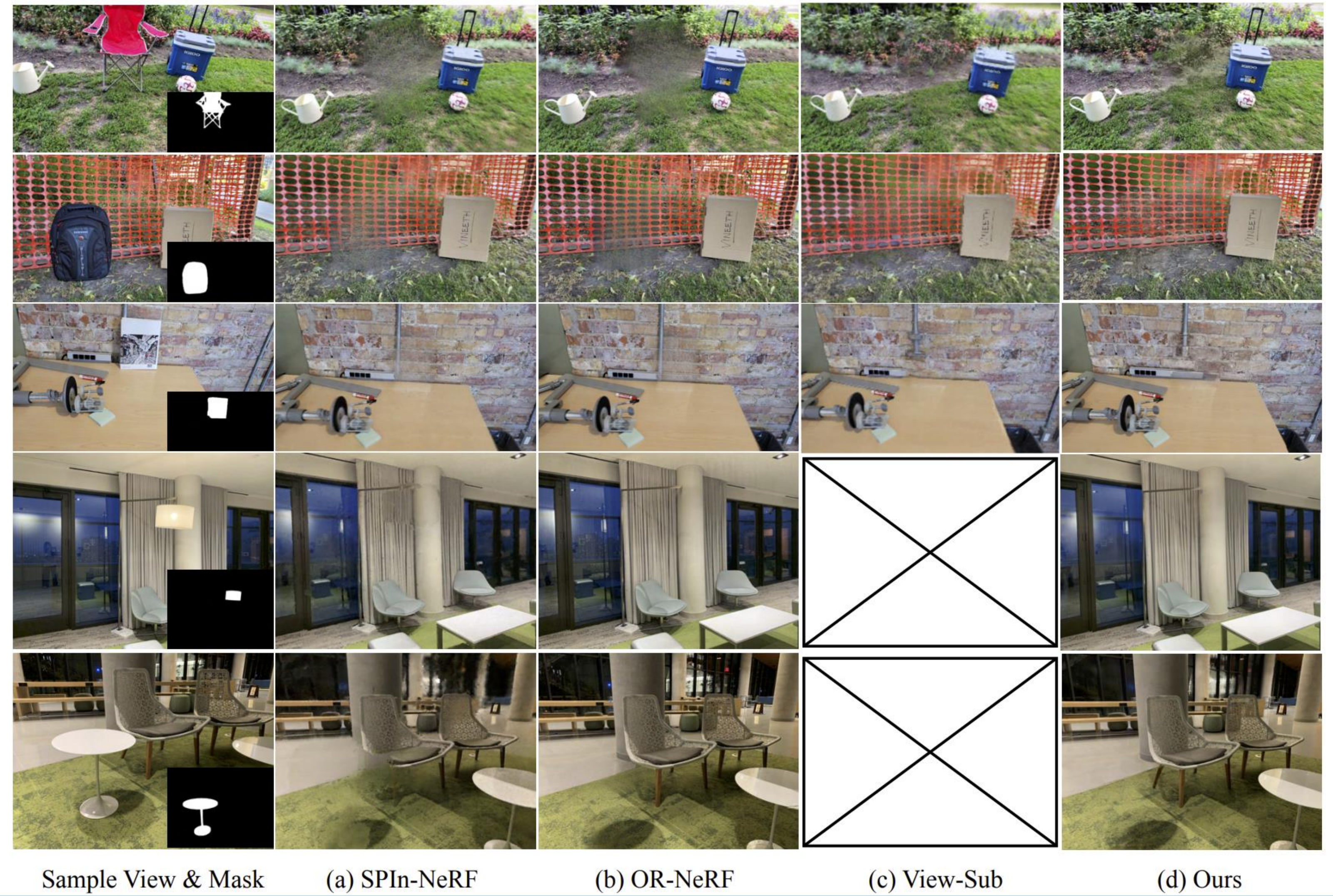


Fig. : Qualitative results compared with the most representative object removal approaches

Reproduced results

Metric	Paper Results	Reproduced Results
Training Iterations	30,000	30,000
GPU Specs	NVIDIA RTX 3050	Dual T4 GPUs (Kaggle)
Learning Rate Schedule	Adaptive	Same as paper
Total Training Time	~12h	~29h

Reproduced results

Scene	PSNR ↑	SSIM ↑	LPIPS ↓	masked PSNR ↑	masked SSIM ↑	masked LPIPS ↓
1	19.58	0.47	0.33	13.05	0.12	0.58
2	18.91	0.49	0.35	15.43	0.15	0.57
3	17.98	0.53	0.25	15.12	0.23	0.41
4	22.08	0.64	0.31	21.13	0.41	0.71
7	21.24	0.61	0.22	16.22	0.11	0.50
9	22.23	0.65	0.20	17.02	0.09	0.48
10	19.61	0.63	0.25	14.92	0.16	0.54
12	16.37	0.41	0.35	12.28	0.06	0.63
book	23.75	0.79	0.22	16.28	0.24	0.55
trash	23.51	0.72	0.23	16.21	0.24	0.54

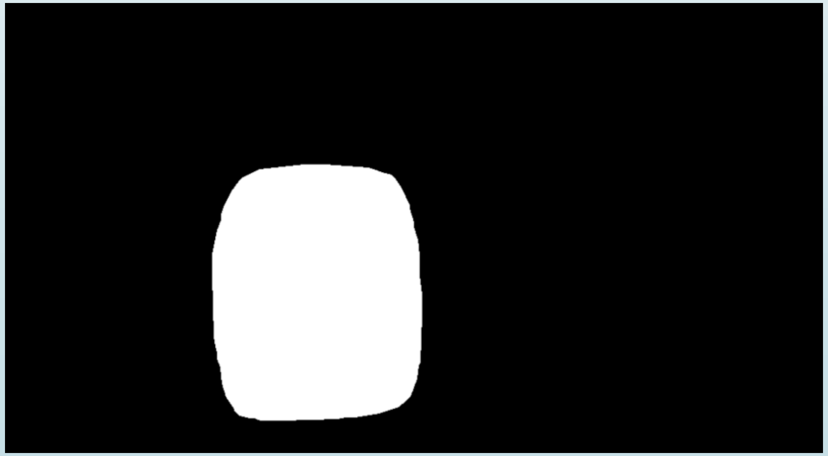
Table 1: Per-scene metric values

	PSNR ↑	SSIM ↑	LPIPS ↓	masked PSNR ↑	masked SSIM ↑	masked LPIPS ↓
Original Results	20.49	0.58	0.28	15.69	0.21	0.54
Reproduced Results	**20.53**	**0.59**	**0.27**	**15.81**	0.21	0.54

Table 2: Comparison between Original Results and Reproduced Results

Reproduced results

Original Image



Object Mask

Result with object removed

Limitations of the Proposed Method

- Texture inconsistency across novel viewpoints **(Novelty 1)**
- Boundary artifacts and blurry transitions in removal zones **(Novelty 1)**
- Equal loss weighting causes error propagation in uncertain regions **(Novelty 2)**
- **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.

Novelties

- **Implemented Novelties**
  1. Perceptual Loss Integration and Gradient Domain Consistency Loss
  2. Adaptive Depth Confidence Weighting

- **Proposed Novelty**

  Autonomous vehicle domain

Novelty 1- Perceptual Loss Integration and Gradient Domain Consistency Loss

- **Why Novelty 1**
  - Boundary artifacts in object removal regions
  - Texture inconsistency across novel viewpoints
  - Limited perceptual quality in complex regions
  - Geometric discontinuities at removal boundaries
- **VGG-Based Perceptual Loss - (For improving LPIPS)**
  - Multi-layer feature matching for enhanced texture fidelity
  - Regional application with mask-awareness
- **Gradient Domain Consistency Loss - (For improving PSNR,SSIM)**
  - Sobel-based boundary preservation
  - Adaptive boundary emphasis for seamless transitions

Reference-

[4] VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION- ICLR 2015

Perceptual Loss Formulation

$$\mathcal{L}_{\text{perceptual}}(I, \hat{I}, M) = \sum_{l=1}^{L} w_l \cdot \frac{1}{C_l H_l W_l} \sum_{c,h,w} M_l \cdot \left(F_l(I)_{c,h,w} - F_l(\hat{I})_{c,h,w}\right)^2$$

- Where,

  I : Ground truth image

  I^ : Predicted (rendered) image

  F_l (.) : Feature map extracted from the l^{th} layer of the VGG16 network

  w_l: Weight for the l^{th} layer [0-1/32, 3-1/16, 8-1/8, 15-¼ , 22-1]

  M_l : Spatial mask for focusing on specific regions in layer *l*

  C_l, H_l, W_l : Number of channels, height, and width of the feature map at layer l

Gradient Domain Consistency Loss

$$\mathcal{L}_{\text{grad}}(I, \hat{I}, M) = \frac{1}{CHW} \sum_{c,h,w} W_{h,w} \cdot M_{c,h,w} \cdot \left(\left| G_x(\hat{I})_{c,h,w} - G_x(I)_{c,h,w} \right| + \left| G_y(\hat{I})_{c,h,w} - G_y(I)_{c,h,w} \right| \right)$$

- Where,

  I : Ground truth image

  I^ : Predicted (rendered) image

  G$_x$,G$_y$: Sobel gradient operators applied in the **x** and **y** directions

  C, H, W - The dimensions of the images

  M$_{c,h,w}$: Mask at each pixel (c, h, w)

  W$_{h,w}$: **Boundary emphasis weight**, calculated as-

$$W_{h,w} = 1 + 5 \cdot \left(|G_x(M)|_{h,w} + |G_y(M)|_{h,w} \right)$$

Total Loss Function

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{RGB}} + 0.1 \cdot \mathcal{L}_{\text{perceptual}} + 0.5 \cdot \mathcal{L}_{\text{grad}}$$

where,

L_{RGB} : The original GScream RGB loss, measuring direct pixel-wise difference between the predicted and ground truth images.

$L_{perceptual}$: Perceptual loss using VGG16 features

L_{grad} : Gradient domain loss for edge sharpness

Result comparison

Metric	Original Results*	Novelty1 Results*	Improvement
PSNR ↑	20.1680551	**20.3065662**	+0.1385111
masked PSNR ↑	14.9271436	**15.0328143**	+0.1056707
SSIM ↑	0.5878702	**0.5880707**	+0.0002005
masked SSIM ↑	0.2638252	**0.2693104**	+0.0054852
LPIPS ↓	0.2693806	**0.2614031**	+0.0079775
masked LPIPS ↓	0.4770665	**0.4677608**	+0.0093057

* - Results on scenes 1,3,10,trash

Novelty 2 -Adaptive Depth Confidence Weighting

- **Why Novelty 2**
  - **Depth maps unreliable at boundaries** - Edge regions have inaccurate values
  - **Equal weighting is problematic** - Errors propagate into reconstruction
  - **Visual artifacts** appear in removed object regions

- **Key Insight:** Not all depth values are equally reliable
- **Solution:** Weight depth loss based on estimated confidence
- **Confidence Estimation:**
  - Higher confidence in smooth regions
  - Lower confidence at depth discontinuities (edges)

Implementation

```
# Before: Standard depth loss
loss += depth_lr * l1_loss(aligned_depth, midas_depth)
```

```
# After: Confidence-weighted depth loss
confidence_map = generate_depth_confidence_map(depth)
weighted_mask = valid_mask * confidence_map
loss += depth_lr * l1_loss_masked(aligned_depth, midas_depth, weighted_mask)
```

- Creating a confidence map
  - Values close to 0 = low confidence (edges, complex geometry) → Low influence
  - Values close to 1 = high confidence (smooth depth regions) → High influence
- Combining the original valid mask (0 for object to remove, 1 for background) with the confidence map
- This multiplication creates a new mask where:
  - Areas outside the valid region remain 0
  - Valid areas now have values between 0-1 based on confidence
  - Result: A spatially-varying weighting map based on depth reliability

Modified confidence weighting depth loss

$$\mathcal{L}_{\text{depth}}^{\text{weighted}} = \lambda_d \cdot \|D_{\text{pred}} - D_{\text{gt}}\|_1 \cdot (M \odot C)$$

where,

D_{pred} : Predicted depth

D_{gt} : Ground truth depth

λ_d : depth loss weight

M : Binary mask

C : Confidence map with values in [0, 1] indicating the reliability of each depth value

$\odot$: Element-wise multiplication

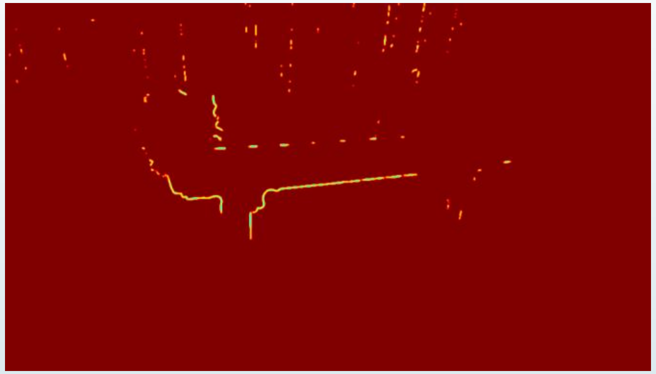
Result comparison

Metric	Original Results*	Novelty2 Results*	Improvement
PSNR ↑	20.4270301	**20.5403814**	+0.1133513
masked PSNR ↑	15.1102653	**15.2583745**	+0.1481092
SSIM ↑	0.6023686	**0.6034779**	+0.0011093
masked SSIM ↑	0.2109824	**0.2137069**	+0.0027245
LPIPS ↓	0.2633046	**0.2625838**	+0.0007208
masked LPIPS ↓	0.5248348	**0.5244400**	+0.0003948

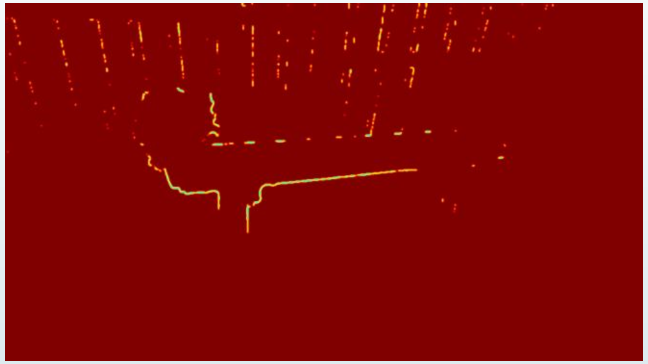
* - Results on scenes – 9,10,12, trash

Confidence map

Original image



Confidence Map at 5000 it.



Confidence Map at 24000 it.

Proposed Novelty-Application in Autonomous vehicle domain

- **Extension to Dynamic Scenes**
- 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.

Challenges

- **12-Hour Session Limit**: Kaggle's maximum runtime per session forced breaking training into multiple sessions
- **Limited GPU Hours**: Kaggle's free tier restricts GPU usage to 30 hours per week
- **Limited GPU Memory**: T4 GPUs have less VRAM than modern GPUs
- **CUDA Memory Errors**
- **Higher Training Time** : Working within Kaggle's runtime limits for free tier usage
- **Debugging Overhead**: Significant time spent on debugging

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.

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. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION-ICLR 2015
5. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric CVPR (2018)
6. Advances in 3D Generation: A Survey - https://3dvar.com/Li2024Advances.pdf
7. Neural RGB-D Surface Reconstruction CVPR 2022

Thank You