CoherentGS: Sparse Novel View Synthesis with Coherent 3D Gaussians

Diana Aldana - Revisora
Horácio Macêdo - Arqueólogo
Mohara Nascimento - Hacker
Victor Ferrari - Estudante de Doutorado

impa

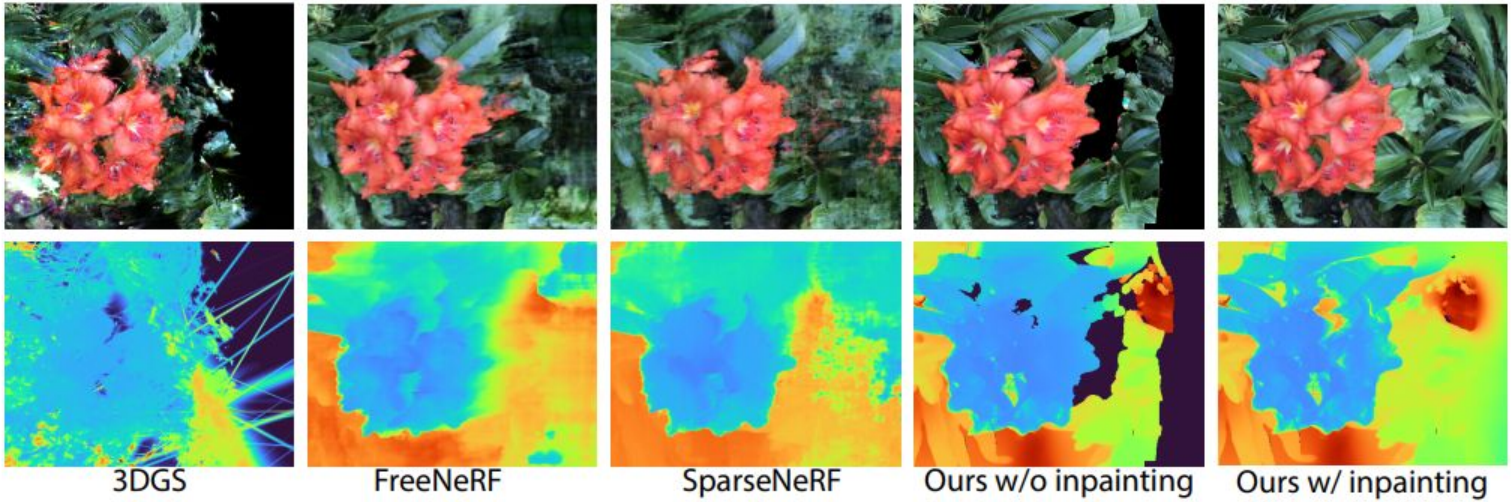
CoherentGS

Revisora - Diana Aldana

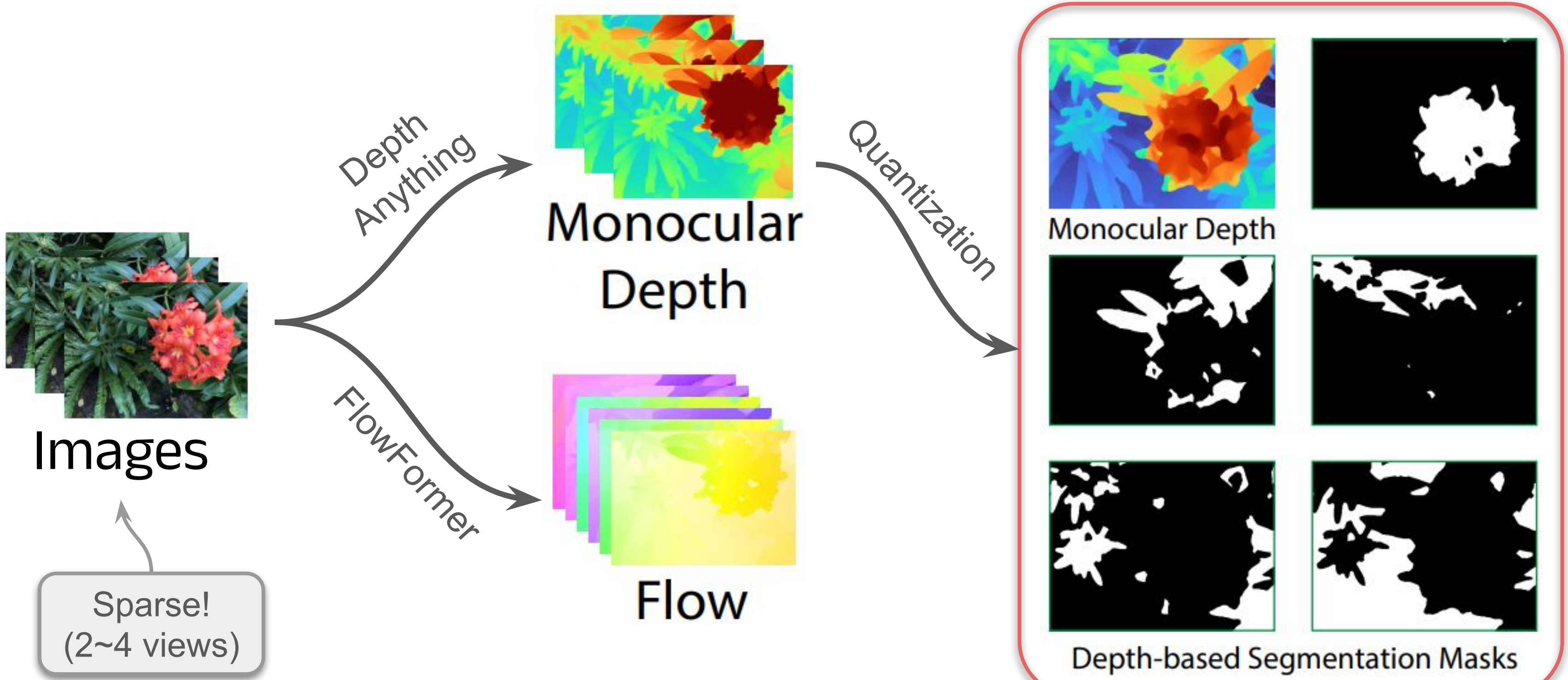
Method: Problem

Reconstruction of scene from sparse views (2~4 images)

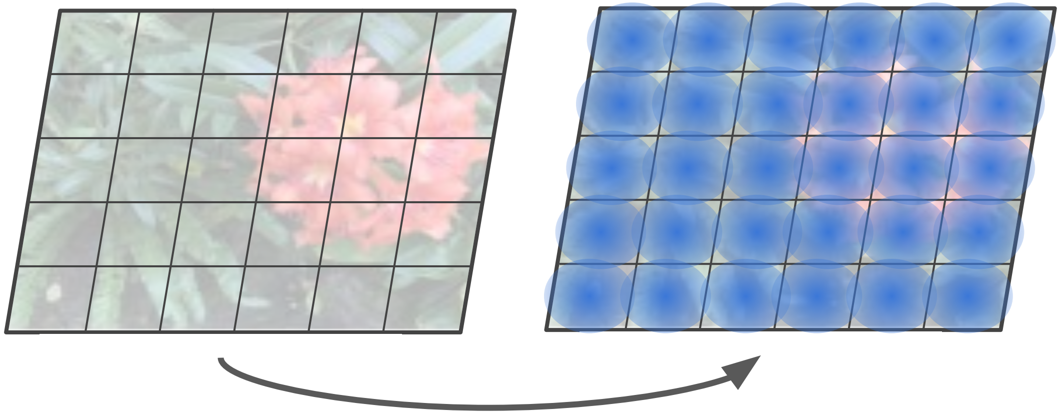
- For applications where it's hard to obtain multiple views (such as medical images).



Method: Inputs

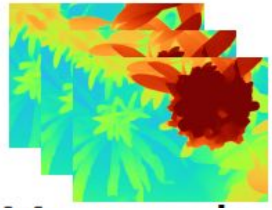


Method: Depth initialization

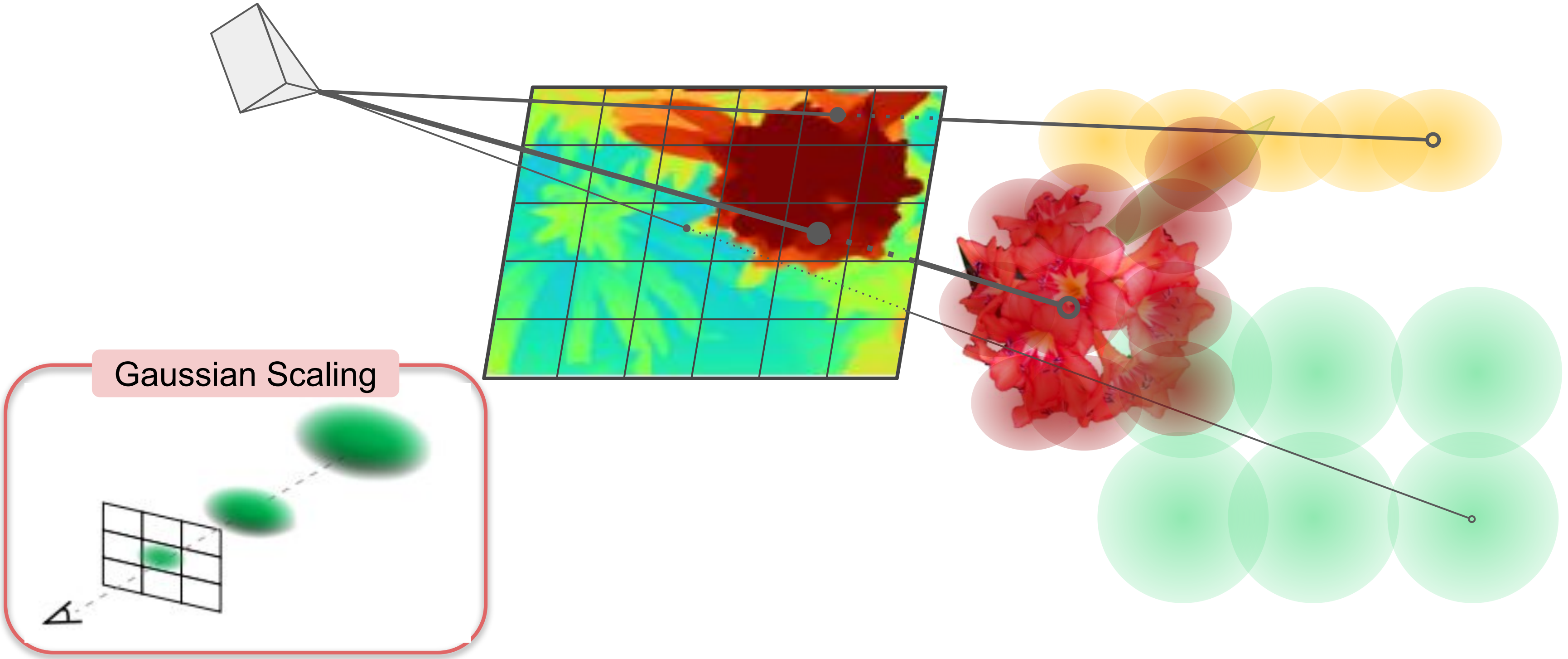


Initializes a 3D gaussian
per pixel per image

Uses monocular
depth estimations to
throw the 3D
gaussians to the
space



Method: Initialization D^{init} for a single view



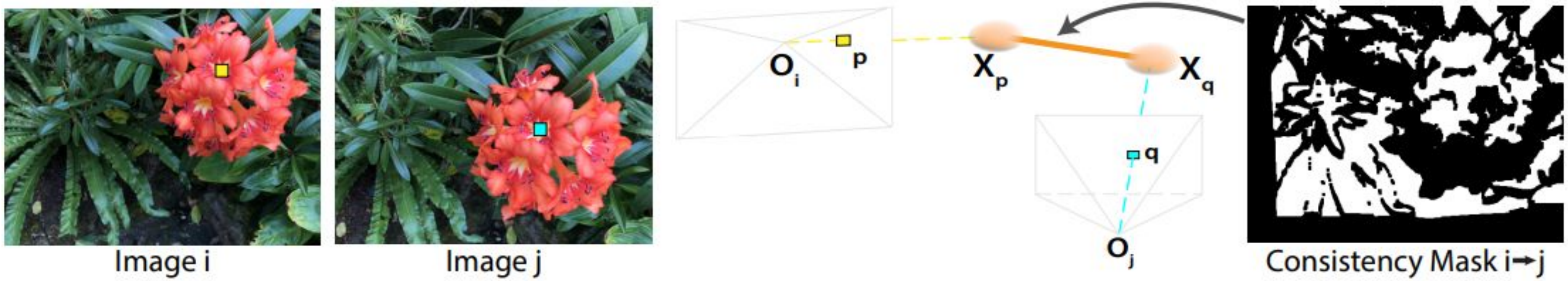
Method: Initialization D^{init} for multiple views

Misalignment due to view-dependency of monocular depth maps.

Initialization using Monocular Depth

Coarse Alignment with Optical Flow

Method: Initialization D^{init} for multiple views



$g(d, p)$ projects pixel p into 3D according to depth d

$$\mathbf{s}^*, \mathbf{o}^* = \arg\min_{\mathbf{s},\mathbf{o}} \sum_{(i,j)} \sum_{\mathbf{p}} \left\| M_{i\to j} \odot \left(g(s_i \cdot D_i^{\mathrm{m}}[\mathbf{p}] + o_i, \mathbf{p}) - g(s_j \cdot D_j^{\mathrm{m}}[\mathbf{q}] + o_j, \mathbf{q}) \right) \right\|_1$$

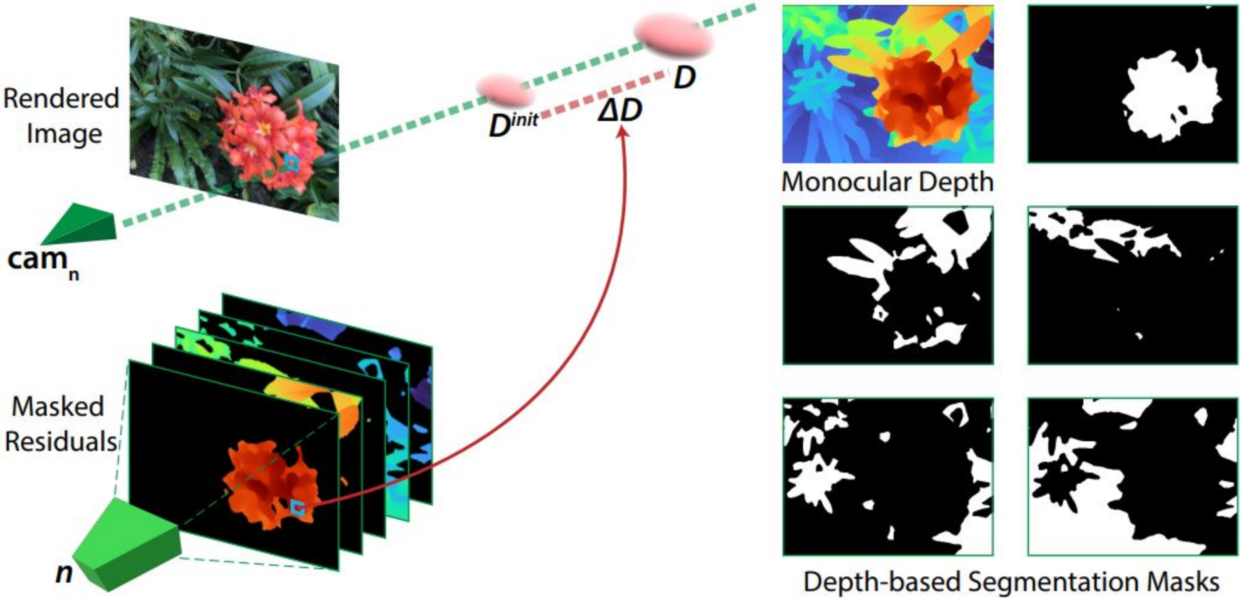
scale

monocular depth

offset

$$\boxed{D^{\text{init}} = s \cdot D^{\mathrm{m}} + o}$$

Method: Training constraint (single view)



$$\mathbf{x} = g(D_n^{\text{init}}[\mathbf{p}] + \Delta D_n[\mathbf{p}], \mathbf{p})$$

gaussian position

Uses a decoder to obtain the residual depth:

$$\Delta D_n = f_\phi(n)$$

Gaussian opacity is trained in a similar way

Method: Training constraint (multi-view)

Total variation constraints: Smoothness of reconstructed geometry

$$\mathcal{L}_{\mathrm{TV}} = \left\| \nabla \left(\frac{1}{1 + R_{\Sigma,\alpha,\mathbf{x},d}} \right) \right\|_1, \quad \mathcal{L}_{\mathrm{MTV}} = \left\| \nabla \left(\mathbf{S} \odot \left(\underbrace{\frac{1}{1 + R_{\Sigma,\alpha,\mathbf{x},d}}}_{\text{rendered depth}} \right) \right) \right\|_1$$

$$\boxed{\mathcal{L}_{\mathrm{multi}} = (1 - \lambda_s)\mathcal{L}_{\mathrm{TV}} + \lambda_s \mathcal{L}_{\mathrm{MTV}}}$$

Flow-based constraint: Force the position of the Gaussians of the corresponding pixels in two images to be similar

$$\boxed{\mathcal{L}_{\mathrm{flow}} = \sum_{(i,j)} \sum_{\mathbf{p}} \left\| M_{i \to j} \odot \left(g(\underbrace{D_i[\mathbf{p}]}_{D_i^{\mathrm{init}}[\mathbf{p}] + \Delta D_i[\mathbf{p}]}, \mathbf{p}) - g(D_j[\mathbf{q}], \mathbf{q}) \right) \right\|_1}$$

Method: Optimization

$$\boxed{\Sigma^*, \phi^*, \mathbf{c}^* = \arg\min_{\Sigma,\phi,\mathbf{c}} \sum_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(R_{\Sigma,\alpha,\mathbf{x},\mathbf{c}}(\mathbf{p}), R(\mathbf{p})) + \beta_m \mathcal{L}_{\text{multi}} + \beta_f \mathcal{L}_{\text{flow}}}$$

Other details:

- Position and opacity are implicitly optimized by ϕ.
- Optimization at multiple samples within each pixel. (Didn't understand well)
- Training schedule:
  - 1000 epochs: Coarse alignment (initialization)
  - 8000 epochs: Gaussian splatting with rotation matrix $\boldsymbol{R} = \boldsymbol{I}_{3x3}$ and scale $\boldsymbol{S}$ such that gaussians are isotropic with radius $r = f \cdot D^{\text{init}} / H$
  - 5000 epochs: Gaussian splatting without constraints over $\boldsymbol{R}$ or $\boldsymbol{S}$.

Advantages

+ Very well written, with good illustrative figures for their method.
+ Interesting use of masks to fill the voids using generative models.
+ Complete ablations and experiments.

To improve

- The decoder structure could be better described.
- Some notation was not clearly explained (e.g., S on the total variation loss).
- Lack of a previous reference: (30/11/2023: SparseGS: Real-Time 360° Sparse View Synthesis using Gaussian Splatting)

Decision and justification

- The method is clearly explained and innovative.
- The points to improve were mostly to polish the article.
- Using generative models to complete the scene is interesting, but under-explored.
- Points of concern:
  - How this method compares to the unreferenced GS sparse-view work?
  - Computational cost / time compared to classical 3DGS.

Considering the previous points, I believe the article should be **accepted** 😇.

CoherentGS

Arqueólogo - Horácio Macêdo

O passado: 3D Gaussian Splatting

- Trabalhos anteriores (tanto de NeRF quanto de 3DGS) tendem a sofrer para reconstruir cenas com poucas visadas
- 3DGS reconstrói cenas a partir de gaussianas não-estruturadas que restringem fracamente a reconstrução e leva a *overfitting* do modelo
- CoherentGS tem a intenção de introduzir coerência na representação não-estruturada através de:
  - inicialização de splats de acordo com a profundidade de visadas monoculares
  - forçar deslocamento de splats de forma coerente de acordo com suas profundidades

O passado: NeRFs com imagens esparsas

Reg-NeRF (Niemeyer et al. 2022)
- Regularização de geometria e cor a partir de pontos de vista não observados

DS-NeRF (Deng et al. 2022)
- Pontos esparsos 3D do COLMAP para supervisionar profundidade

ViP-NeRF (Somraj & Soundararajan, 2023)
- Pré-computação de um *prior* de visibilidade para restringir volume

Sparse-NeRF (Wang et al., 2023)
- Uso de profundidade monocular para produção de um rank de profundidade local e regularização de continuidade espacial

FreeNeRF (Yang, 2023)
- Refinamento do modelo através de aumento gradual de frequências de codificação de posicionamento (*positional encoding*)

FlipNeRF (Seo, 2023)
- Criação de raios de reflexão para compensar por poucas visadas conhecidas

Reg-NeRF

Regularização de geometria e cor a partir de pontos de vista não observados

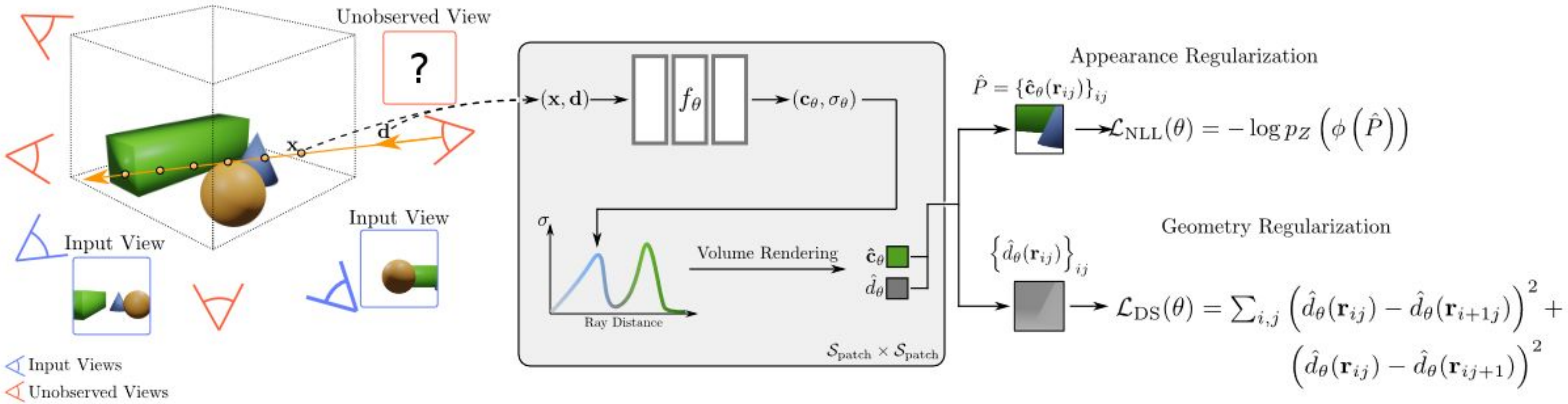
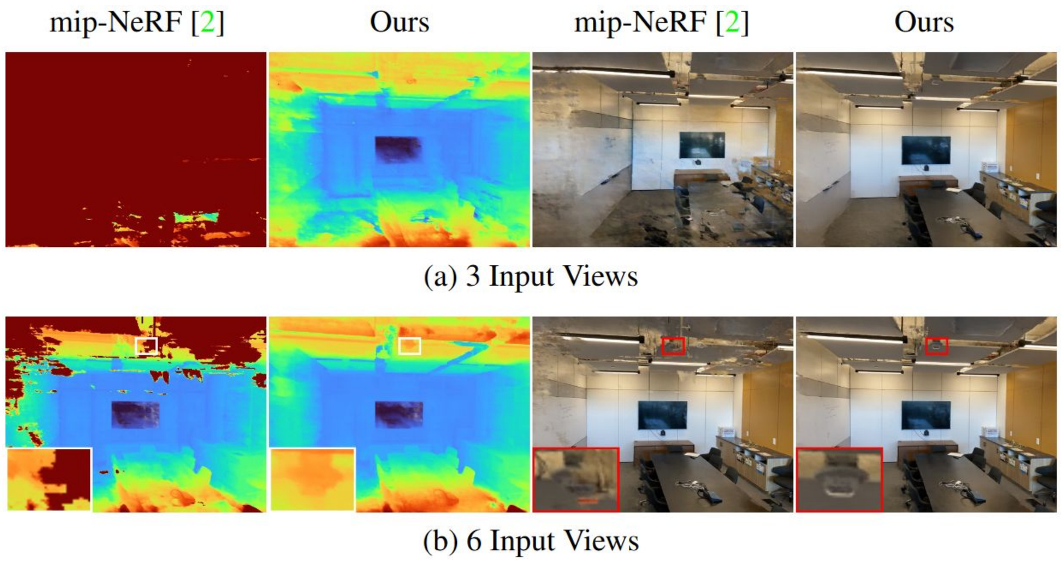


Figure 2. **Overview.** NeRF optimizes the reconstruction loss for a given set of input images (blue cameras). For sparse inputs, however, this leads to degenerate solutions. In this work, we propose to sample unobserved views (red cameras) and regularize the geometry and appearance of patches rendered from those views. More specifically, we cast rays through the scene and render patches from unobserved viewpoints for a given radiance field f_θ. We then regularize appearance by feeding the predicted RGB patches through a trained normalizing flow model ϕ and maximizing predicted log-likelihood. We regularize geometry by enforcing a smoothness loss on the rendered depth patches. Our approach leads to 3D-consistent representations even for sparse inputs from which realistic novel views can be rendered.

Reg-NeRF

- Regularização de cor e geometria em visadas desconhecidas que residem entre visadas conhecidas
- Estratégia de *annealing* no início do treino para redução de densidade no início dos raios



DS-NeRF

Uso de pontos esparsos do COLMAP para supervisão de treino de depth

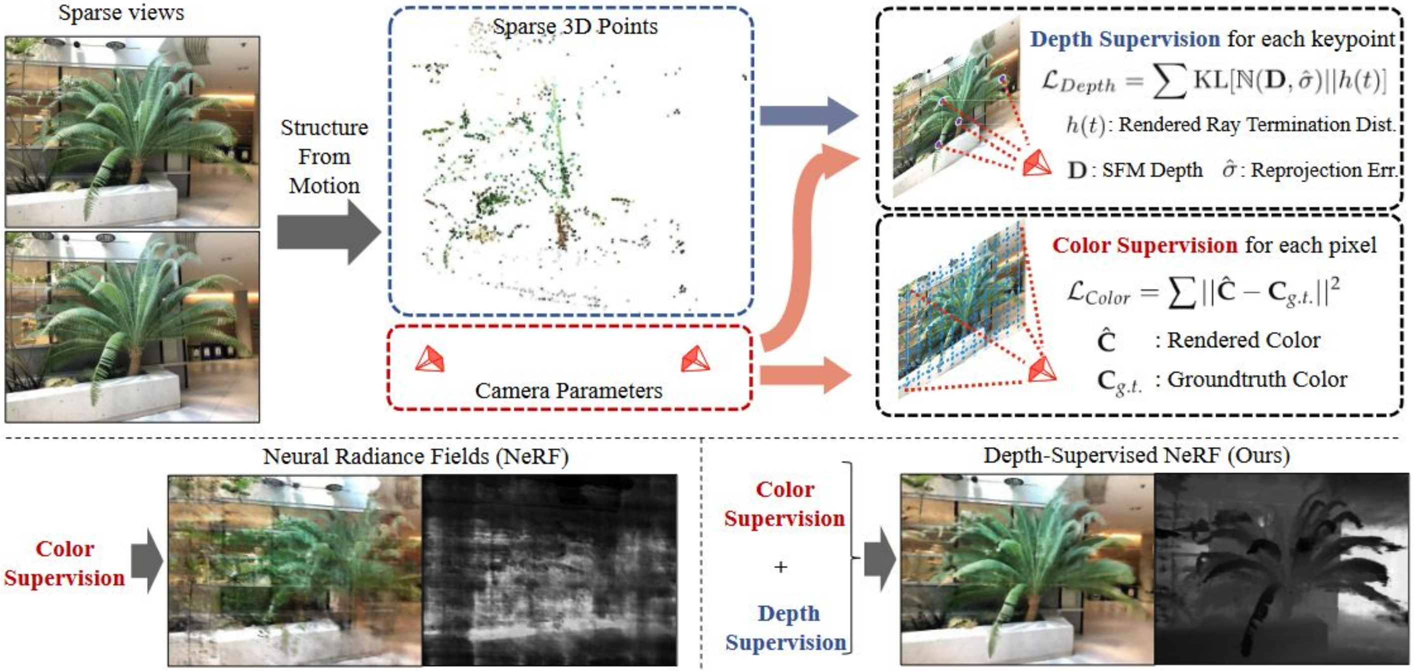


Figure 1. Training NeRFs can be difficult when given insufficient input images. We utilize additional supervision from depth recovered from 3D point clouds estimated from running structure-from-motion and impose a loss to ensure the rendered ray's termination distribution respects the surface priors given by the each keypoint. Because our supervision is complementary to NeRF, it can be combined with any such approach to reduce overfitting and speed up training.

ViP-NeRF

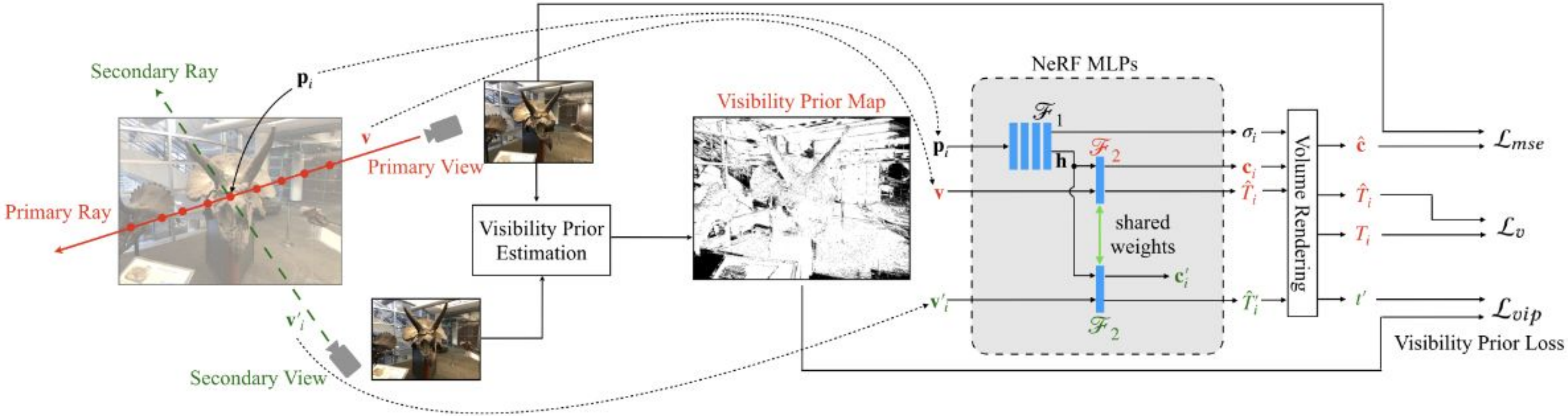


Fig. 1. Overview of ViP-NeRF architecture. Given the images from primary and secondary views, we estimate a visibility prior map in the primary view and use it to supervise the visibility of pixels as predicted by the NeRF. Specifically, we cast a ray through a randomly selected pixel in the primary view and sample 3D points along the ray. For every point p_i, we use the NeRF MLPs to obtain its visibility in primary and secondary views, along with volume density σ_i and color c_i. Volume rendering outputs visibility t' of the chosen pixel in the secondary view which is supervised by the visibility prior. $\mathcal{L}_v$ constrains the visibilities $\hat{T}_i$ output by network and T_i computed using volume rendering to be consistent with each other.

ViP-NeRF

- Regularização do NeRF com *priores* densos de visibilidade a partir da estimação esparsa de pontos
- Priores obtidos através de *plane sweep volumes* (PSVs) e mapas binários de visibilidade

Fig. 3. Qualitative examples on RealEstate-10K dataset with two input views. We observe that the predictions of ViP-NeRF are close to the ground truth, while those of other models suffer from various distortions. In particular, DDP-NeRF blurs regions of the frame near the left door and contains black floater artifacts.

SparseNeRF

- Usa quatro componentes:
  - o *backbone* do MipNeRF
  - um módulo de reconstrução de cor
  - um módulo de distilação de *ranking* de profundidade
  - um módulo de distilação de continuidade espacial

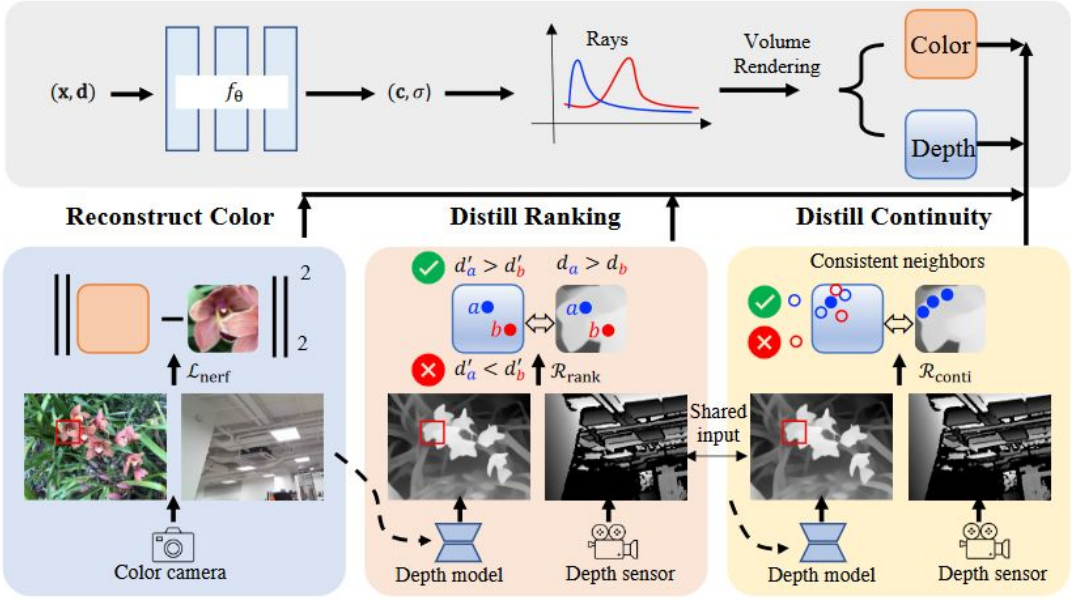
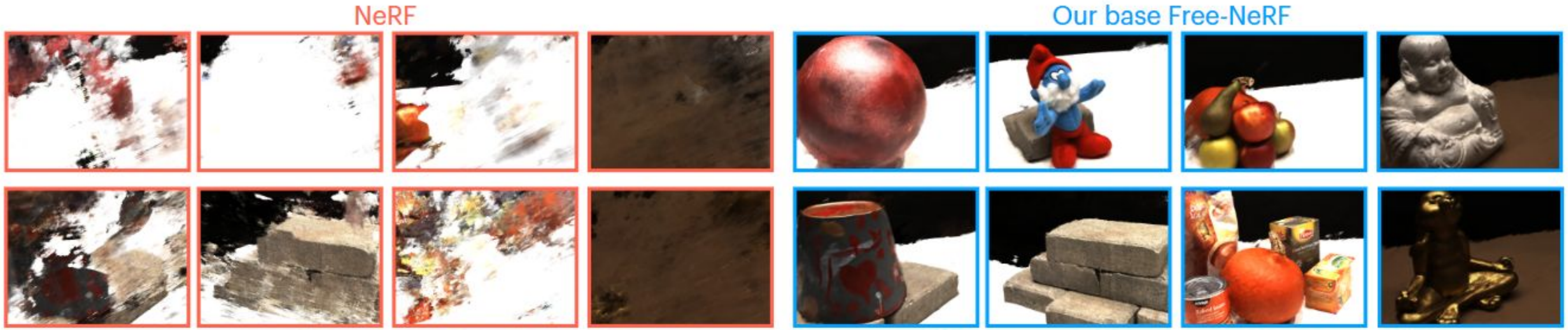


Figure 2: Framework Overview. SparseNeRF consists of two streams, i.e., NeRF and depth prior distillation. As for NeRF, we use Mip-NeRF as the backbone. we use a NeRF reconstruction loss $\mathcal{L}_{nerf}$. As for depth prior distillation, we distill depth priors from a pre-trained depth model. Specifically, we propose a local depth ranking regularization and a spatial continuity regularization to distill robust depth priors from coarse depth maps.

FreeNeRF

- **Fre**quency regularized **NeRF**
- Modifica NeRF para regularizar frequências nos dados de entrada
- NeRF tende a *overfitting* em cenários com poucas imagens e altas frequências
- Dois termos de regularização:
    - regularização do *range* de frequências nos dados de entrada;
    - penalização de oclusão de objetos próximos o suficiente da câmera



FlipNeRF

Criação de raios de reflexão com base nos raios traçados por visadas conhecidas para reconstruções com visadas esparsas

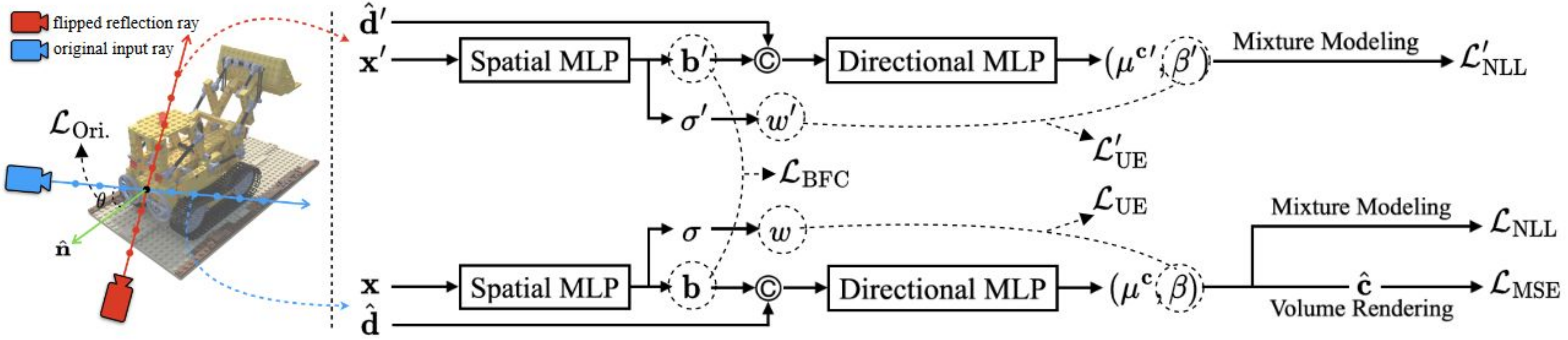


Figure 2: **Overall framework of FlipNeRF.** Our FlipNeRF utilize the newly generated flipped reflection rays with our proposed UE Loss and BFC Loss as well as existing MSE, NLL and Orientation losses. See Sec. 3 and Fig. 3 for more details about generation process of flipped reflection rays and the loss terms.

3DGS para síntese de visadas esparsas

FSGS (ECCV 2024)

- Gera priores de profundidade monoculares a partir de visadas desconhecidas para geração de uma geometria razoável

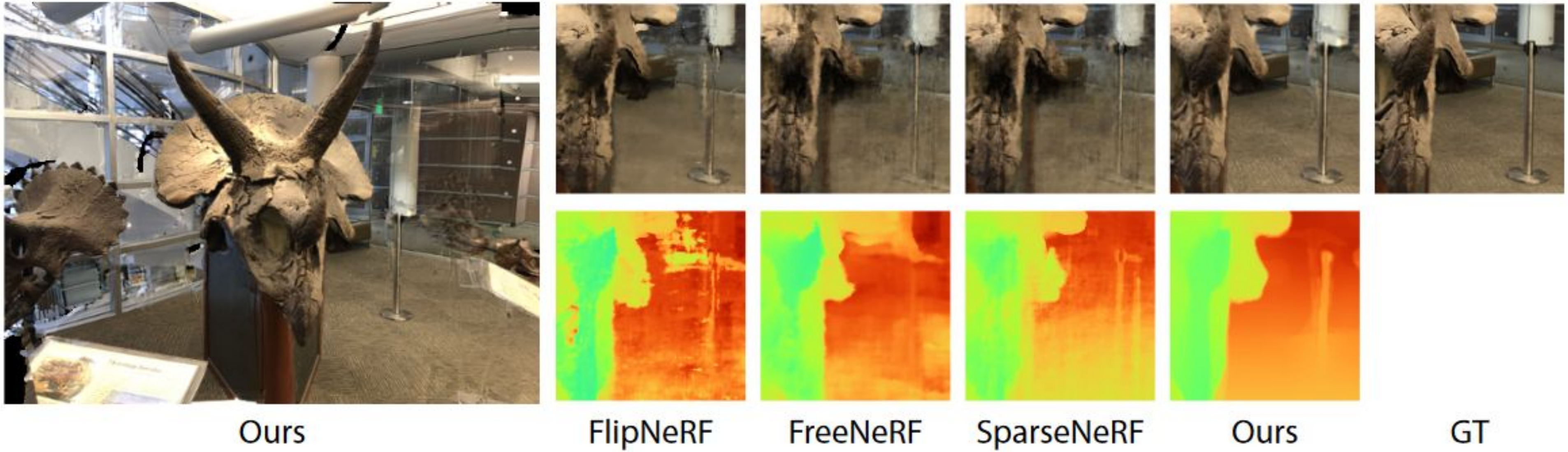
SparseGS (2023)

- Estimação de profundidade a partir de uma técnica própria (*softmax depth* + Loss de correlação de retalhos de profundidade + Loss de amostragem de destilação + Loss de reprojeção)
- Poda de artefatos indesejáveis através de um operador adaptativo

DNGaussian (CVPR 2024)

- Regularização em dois passos (*hard* e *soft*) para restringir a geometria dos splats
- Normalização de profundidade Global-Local

3DGS para síntese de visadas esparsas



CoherentGS: Sparse Novel View Synthesis with Coherent 3D Gaussians (ECCV - 2024)

CoherentGS foi citado por:

- GeoRGS: Geometric Regularization for Real-Time Novel View Synthesis from Sparse Inputs **(IEEE Transactions on Circuits and Systems for Video Technology, 2024)**
- GGS: Generalizable Gaussian Splatting
- Next Best Sense: Guiding Vision and Touch with FisherRF for 3D Gaussian Splatting
- Leveraging Depth Maps and 3D Gaussian Splatting for Camera Pose Recovery and 3D Scene Reconstruction
- Self-Ensembling Gaussian Splatting for Few-Shot Novel View Synthesis
- FewViewGS: Gaussian Splatting with Few View Matching and Multi-stage Training **(NeurIPS 2024)**
- Aquatic-GS: A Hybrid 3D Representation for Underwater Scenes

GeoRGS: Geometric Regularization for Real-Time Novel View Synthesis from Sparse Inputs

Publisher: IEEE Cite This PDF

Zhaoliang Liu ; Jinhe Su ; Guorong Cai ; Yidong Chen ; Binghui Zeng ; Zongyue Wang **All Authors**

- Early access na IEEE Transactions on Circuits and Systems for Video Technology
- Mitigação de *overfitting* do treino do 3DGS através de regularização geométrica
- Propõe duas técnicas para correção da geometria:
  - Seleção de *seed patches* da 3D Gaussian na cena para geração de geometria correta
  - Regularização da similaridade de profundidade entre superfícies e bordas

FewViewGS: Gaussian Splatting with Few View Matching and Multi-stage Training

- Publicado no NeurIPS, 2024
- Usa o CoherentGS como inspiração (citado apenas uma vez nos trabalhos relacionados)
- Método em três partes para reconstrução de cenas com poucas imagens
    - Treinar 3DGS por poucas eras
    - Sintetizar novas visadas para servir de supervisão para o treino de mais 3DGS
    - Retificação de 3DGS usando visadas conhecidas

CoherentGS

Hacker - Mohara Nascimento

Paper and code

- Gaussian shape using initial depth

Paper

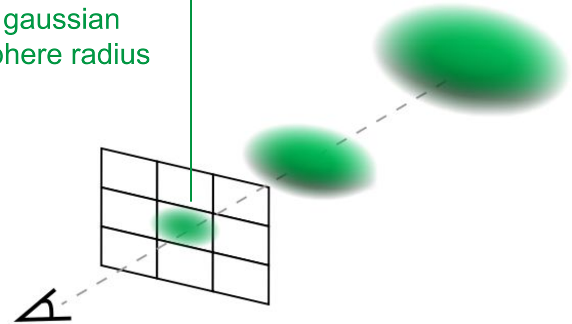
$$r = f \cdot D^{init} / H$$

image height

initial depth

vertical focal length

gaussian
sphere radius



Code

coherentgs/scene/gaussian_model.py

```
    # Reset gaussian shape using depth
    def set_scaling(self, use_decoder=False, radius_mult=None):
        [...]
         with torch.no_grad():
             depth = self._z

             if use_decoder:
                 depth = depth + self.get_residual()

             radii = np.tan(0.5 * float(self.ref_camera.FoVy))  *
depth / self.height
             radii2 = radii**2

             scales = torch.log(torch.sqrt(radii2) *
radius_mult).repeat(1, 3) #/ 1.1
             self._scaling.data = scales.contiguous()go(f, seed, [])
}
```

About reproducibility

- Easy;

Code changes necessary to run:

coherentgs/scene/decoder.py

```
def forward(self, x):
    """
    ...
    """
    # Initial up+conv
    x = self.upconv1(x)

    [...]

    # Padding + last layer
    x = (F.pad(x, self.last_padding, mode=self.last_padding_mode))
    # Convolution + Leaky ReLU
    x = (self.last_act(self.last_layer(x)))
    x.to(torch.float32)
    return x
```

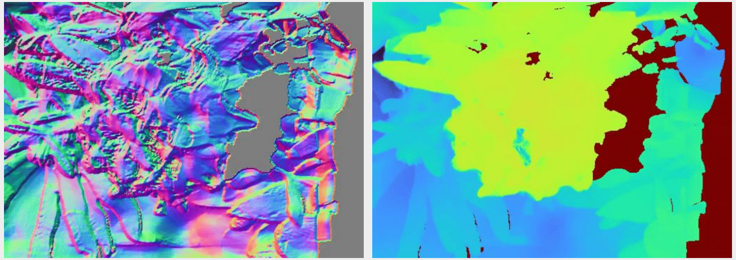
About reproducibility (Demo)

Input

- images_8
  - depth
  - depth_rel
  - flow
  - 34 images.png

Output

3D normal and depth maps

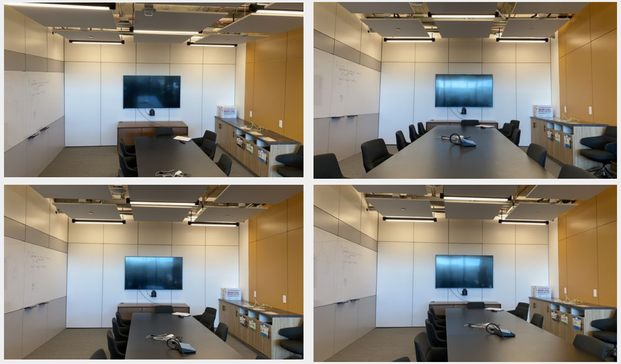


3D reconstruction

About reproducibility (Demo)

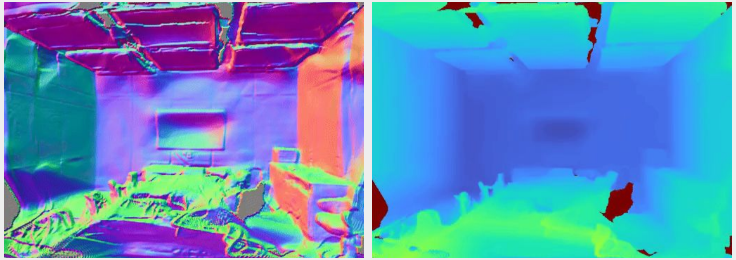
Input

- images_8
  - depth
  - depth_rel
  - flow
  - 41 images.png



Output

3D normal and depth maps



3D reconstruction

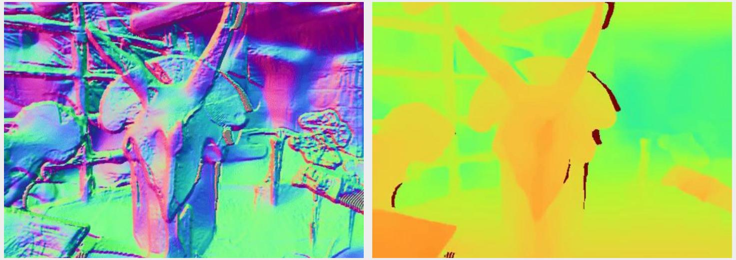
About reproducibility (Demo)

Input

- images_8
  - depth
  - depth_rel
  - flow
  - 62 images.png

Output

3D normal and depth maps



3D reconstruction

Comparison with 3DGS

CoherentGS

3D Gaussian Splatting with Polycam

https://poly.cam

Occluded areas: empty blurry and repetitive structure

Experiments

Change in the number of cameras (from 3 to 4)

3D reconstruction with 3 cameras (default)

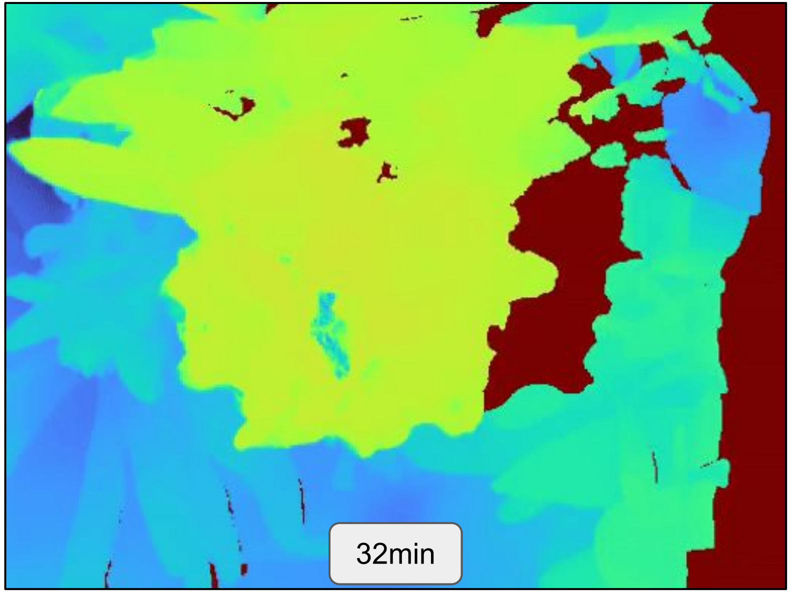
3D reconstruction with 4 cameras

- python train.py --source_path path/nerf_llff_data/flower --eval --model_path output/flower --num_cameras 3

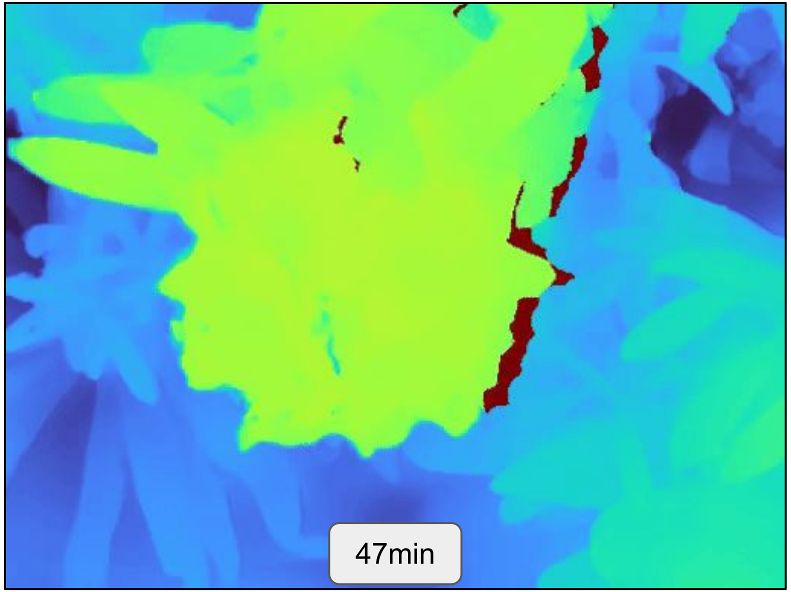
Experiments

Change in the number of cameras (from 3 to 4)

3D depth map with 3 cameras (default)



3D depth map with 4 cameras



- python train.py --source_path path/nerf_llff_data/flower --eval --model_path output/flower --num_cameras 3

Experiments

Change in num of camera and interactions

3D reconstruction with 4 cameras and 20k interactions (default)

3D reconstruction with 4 cameras and 10k interactions

- python train.py [...] --num_cameras 4

coherentgs/arguments/__init__.py

Experiments

Change in num of camera and interactions

3D reconstruction with 3 cameras (default)

3D reconstruction with 4 cameras and 10k interactions

- python train.py [...] --num_cameras 4

coherentgs/arguments/__init__.py

Experiments

Change in num of camera and interactions

3D reconstruction with 4 cameras and 20k interactions (default)

3D reconstruction with 4 cameras and 30k interactions

- python train.py [...] --num_cameras 4

coherentgs/arguments/__init__.py

Experiments

Change in num of camera, interactions and radius scale

3D reconstruction with 4 cameras, 10k interactions

3D reconstruction with 4 cam., 10k int. + radius scale (4)

- python train.py [...] --num_cameras 4

coherentgs/arguments/__init__.py

coherentgs/scene/gaussian_model.py

CoherentGS

Aluno de Doutorado - Victor Ferrari

Proposta de projeto de pesquisa

$$\Sigma^*, \phi^*, \mathbf{c}^* = \arg\min_{\Sigma, \phi, \mathbf{c}} \sum_{\mathbf{p} \in \mathcal{P}} \mathcal{L}(R_{\Sigma, \alpha, \mathbf{x}, \mathbf{c}}(\mathbf{p}), R(\mathbf{p})) + \beta_m \mathcal{L}_{\text{multi}} + \beta_f \mathcal{L}_{\text{flow}}, \tag{7}$$

5

0.1

Proposta de projeto de pesquisa

$$\Sigma^*, \phi^*, \mathbf{c}^* = \arg\min_{\Sigma,\phi,\mathbf{c}} \sum_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(R_{\Sigma,\alpha,\mathbf{x},\mathbf{c}}(\mathbf{p}), R(\mathbf{p})) + \boxed{\underbrace{\beta_m}_{5}\mathcal{L}_{\text{multi}} + \underbrace{\beta_f}_{0.1}\mathcal{L}_{\text{flow}},} \tag{7}$$

Proposta de projeto de pesquisa

Depth-Regularized Optimization for 3D Gaussian Splatting in Few-Shot Images

Jaeyoung Chung[1] Jeongtaek Oh[2] Kyoung Mu Lee[1,2]
[1]ASRI, Department of ECE, [2]IPAI, Seoul National University, Seoul, Korea
{robot0321,ohjtgood,kyoungmu}@snu.ac.kr

https://openaccess.thecvf.com/content/CVPR2024W/3DMV/papers/Chung_Depth-Regularized_Optimization_for_3D_Gaussian_Splatting_in_Few-Shot_Images_CVPRW_2024_paper.pdf

$$\mathcal{L}_{depth} = \left\| D - D^*_{\text{den}} \right\|_1 .$$

$$\mathcal{L}_{smooth} = \sum_{d_j \in \text{adj}(d_i)} \mathbb{1}_{ne}(d_i, d_j) \cdot \left\| d_i - d_j \right\|^2$$

$$\begin{aligned} \mathcal{L} = (1-\lambda_{ssim})\mathcal{L}_{color} + \lambda_{ssim}\mathcal{L}_{D-SSIM} \\ +\lambda_{depth}\mathcal{L}_{depth} + \lambda_{smooth}\mathcal{L}_{smooth} \end{aligned}$$

CoR-GS: Sparse-View 3D Gaussian Splatting via Co-Regularization

Jiawei Zhang[1], Jiahe Li[1], Xiaohan Yu[3], Lei Huang[2], Lin Gu[4,5], Jin Zheng[1⋆], and Xiao Bai[1⋆]

https://arxiv.org/pdf/2405.12110

$$\mathcal{R}_{pcolor} = (1-\lambda)\mathcal{L}_1(I'^1, I'^2) + \lambda\mathcal{L}_{\text{D-SSIM}}(I'^1, I'^2).$$

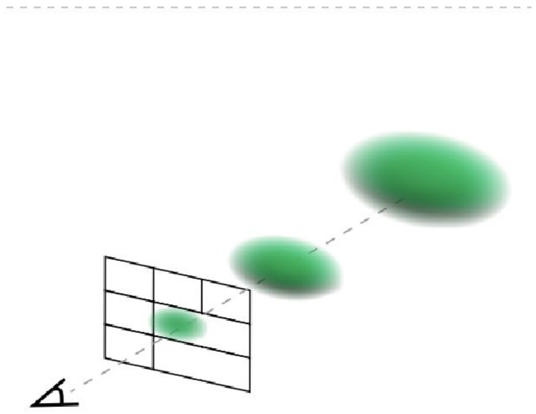
$$\mathcal{L}_{color} = (1-\lambda)\mathcal{L}_1(I^1, I^*) + \lambda\mathcal{L}_{\text{D-SSIM}}(I^1, I^*).$$

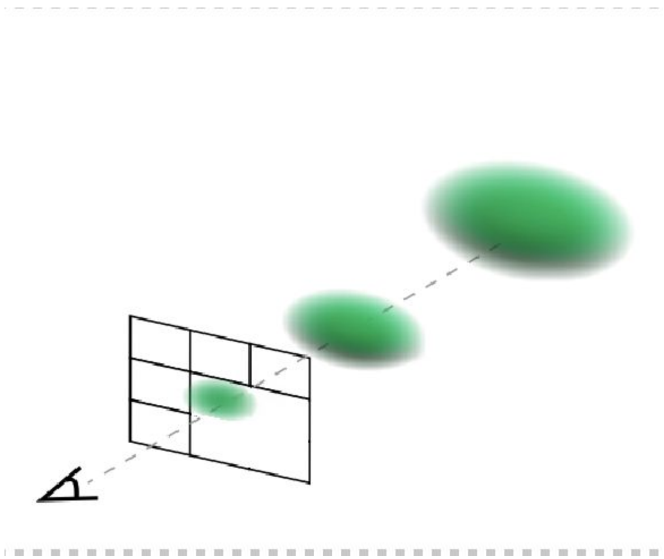
$$\mathcal{L} = \mathcal{L}_{color} + \lambda_p \mathcal{R}_{pcolor},$$

Proposta de projeto de pesquisa

Limitations Since we assign a single Gaussian to each pixel, our approach has difficulty handling scenes with transparent objects. An example of such a case is shown in Fig. 10 where our technique is not able to properly reconstruct both the reflections on the glass and the hand rails behind it. Nevertheless, our results are still significantly better than the competing methods. Additionally, our approach relies on the monocular depth and may not be able to produce reasonable results if the depth is highly inaccurate.

Proposta de projeto de pesquisa





CoherentGS: Sparse Novel View Synthesis with Coherent 3D Gaussians (ECCV - 2024)

Avaliação do projeto em situações práticas

Aplicações em Engenharia Naval e Submarina:

- **Equipamentos em profundidade**: Criação de modelos 3D de equipamentos submarinos em grandes profundidades com imagens limitadas.
- **Ecossistemas Marinhos**: Modelagem automatizada de recifes e corais para monitorar invasores.

Reconstrução de Monumentos Históricos:

- **Reconstrução com imagens escassas**: Recriação de monumentos e estruturas históricas com fotos antigas, mesmo com dados limitados.

Obrigado!

Dúvidas?