Relatório do artigo pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction

Revisor Alberto Arkader Kopiler
IMPA
alberto.kopiler@impa.br

Arqueólog@
Institution2
secondauthor@i2.org

Hacker Vitor Pereira Matias
ICMC - USP
vitorpmatias@usp.br

Fernando Pereira Gonçalves de Sá
IC - UFF
fernandosa@id.uff.br

1. Revisão

1.1. Resumo

- Problema abordado: pixelSplat [1] é um modelo feed-forward que aprende a reconstruir campos de radiância 3D parametrizados por primitivas Gaussianas 3D a partir de pares de imagens de entrada.
- Motivação: A reconstrução 3D é uma das mais populares áreas de pesquisa na *visão computacional*. NeRF (Neural Radiace Fields) [4] introduziu redes neurais para gerar renderizadores 3D. pixelNeRF [5] usa apenas algumas imagens como entrada e um Codificador baseado em CNN no topo de um NeRF para gerar melhores renderizadores 3D. Gaussian Splatting [3] usa Gaussianas 3D e gradiente descedente para gerar melhores renderizadores 3D do que os pré-existentes. pixelSplat se baseia nesses trabalhos para inovar combinando de certa forma Gaussian Splatting com NeRF.
- Resumo do método: pixelSplat combina 3D Gaussian splatting com um truque de reparametrização e redes neurais. Recebe como entrada duas imagens de um objeto de dois pontos de vista diferentes e gera uma renderização 3D com um tempo de inferência reduzido.
- Lista de contribuições: Isso resulta em uma representação 3D explícita que pode ser renderizada em tempo-real, pode ser editada e é barata para treinar (do ponto de vista de processamento e memória, comparada aos métodos existentes).

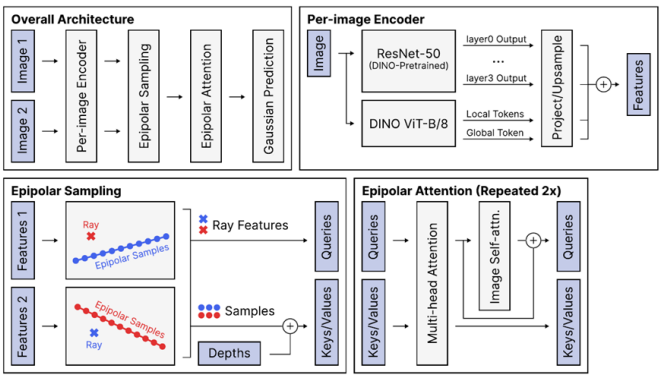
  - Se beneficia de uma representação 3D primitiva para ser rápida e usar memória de forma eficiente.

  - Além de gerar uma renderização, gera também uma estrutura 3D interpretável

  - Síntese de vistas generalizáveis

Podemos ver o diagrama em blocos com a arquitetura do modelo na Figure 1.

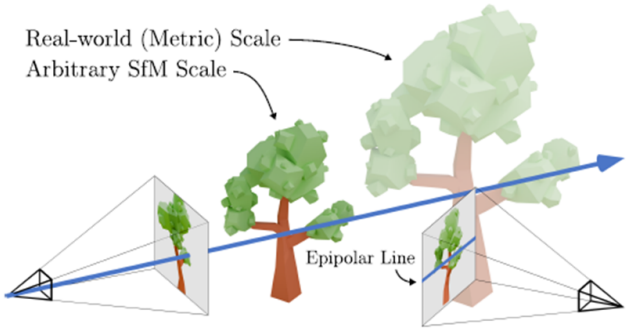
Figure 1. Arquitetura do Modelo.



Fonte: [1].

- Codificador de imagem com duas vistas: pixelSplat começa processando um par de imagens de entrada através de uma rede de extração de características que gera uma representação de alta dimensão de cada imagem. Essa rede neural é frequentemente estruturada similarmente àquelas usadas em arquiteturas NeRF, extraindo características visuais e espaciais cruciais das imagens, estabelecendo o estágio para entender a geometria da cena.
- Geometria Epipolar e Resolução da Ambiguidade de Escala: As características extraídas são então processadas usando um trasformer epipolar, um componente que facilita descobrir o relação entre essas duas vistas para resolver a ambiguidade de escala, um desafio inerente à reconstrução de cenas 3D a partir de imagens 2D Figure 2. Esse passo garante que as posições 3D inferidas a partir de diferentes imagens são consistentes em relação à

Figure 2. Resolução da Ambiguidade de Escala.

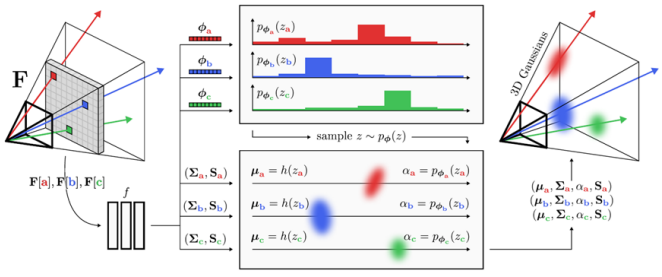


Fonte: [1].

outra, refletindo as variações na posição e orientação das câmeras.

- Amostragem Probabilística de Parâmetros Gaussianos: Tendo a escala e geometria calibradas, o próximo passo envolve uma nova aplicação de 3D Gaussian splatting, onde o modelo prevê uma densa distribuição de densidade para a potencial localização de primitivas Gaussianas. Essa abordagem é facilitada pelo truque de reparametrização que permite a rede amostrar essas localizações de forma diferenciada. Aqui, cada posição Gaussiana (média), forma (covariância) e visibilidade (opacidade) são determinadas, possibilitando que os gradientes sejam retro propagados através da rede durante o treinamento, dessa forma otimizando o posicionamento eficiente das Gaussianas Figure 3 e Figure 4.
- Renderização e Geração da Saída: Finalmente, a cena 3D parametrizada, agora representada como uma coleção de Gaussian splats, é renderizada para produzir novas vistas. Esse processo de renderização é otimizado para redução do tempo de processamento (aumento da velocidade de processamento) e eficiência de uso de memória, usando técnicas de Gaussian splatting como a pegada computacional leve (light computational footprint). A saída é um

Figure 3. Predição Probabilística Proposta de Gaussianas alinhadas por pixel.



Fonte: [1].

Figure 4. Algoritmo de Predição Probabilística de Gaussianas alinhadas por pixel.

Algorithm 1 Probabilistic Prediction of a Pixel-Aligned Gaussian.

Require: Depth buckets $\mathbf{b} \in \mathbb{R}^Z$, feature $\mathbf{F}[\mathbf{u}]$ at pixel coordinate $\mathbf{u}$, camera origin of reference view $\mathbf{o}$, ray direction $\mathbf{d_u}$.

$(\phi, \delta, \Sigma, \mathbf{S}) = f(\mathbf{F}[\mathbf{u}])$ ▷ predict depth probabilities ϕ and offsets δ, covariance Σ, spherical harmonics coefficients $\mathbf{S}$

$z \sim p_\phi(z)$ ▷ Sample depth bucket index z from discrete probability distribution parameterized by ϕ

$\mu = \mathbf{o} + (\mathbf{b}_z + \delta_z)\mathbf{d_u}$ ▷ Compute Gaussian mean μ by unprojecting with depth $\mathbf{b}_z$ adjusted by bucket offset δ_z

$\alpha = \phi_z$ ▷ Set Gaussian opacity α according to probability of sampled depth (Sec. 4.2).

return $(\mu, \Sigma, \alpha, \mathbf{S})$

Fonte: [1].

conjunto de novas imagens, ou novas vistas, geradas a partir de perspectivas que não foram capturadas originalmente pelas imagens de entrada, demonstrando a habilidade do modelo de interpolar e extrapolar o espaço 3D com a restrição de uso de dados limitados.

1.2. Pontos positivos

- Foi feita uma abordagem original para o problema de com somente duas imagens como entrada tiradas de dois diferentes pontos de vista, sintetizar novas vistas.
- O trabalho apresentado utiliza uma pipeline formada por : um codificador de pré-imagem, seguido por um amostrador epipolar, um bloco de atenção epipolar e um bloco de predição gaussiana.
- Os autores afirmam que seu trabalho na fase de inferência é significativamente mais rápido que os trabalhos anteriores em síntese generalizável de novas vistas, ao mesmo tempo em que produz representação explícita de cenas 3D.
- Os autores afirmam que para resolver o problema de mínimo local que ocorre na regressão de função baseada em primitivas, eles introduziram uma nova forma de parametrização de localização de primitivas via uma densa distribuição de probabilidades e também introduziram um novo truque de reparametrização para propagar gradientes de forma retrógrada (backpropagate) nos parâmetros dessa distribuição.
- Os autores afirmam que seu framework é geral e eles espera que seu trabalho inspire novos trabalhos em inferência baseada em ocorrências anteriores (prior-based inference) de representação baseada em primitivas em muitas aplicações.
- Eles sugerem como trabalhos futuros aproveitar seu modelo para modelagem generativa, através da combinação dele com modelos de difusão ou eliminar a necessidade de poses de câmera para permitir o treinamento em larga escala.
- Seu modelo resolve o problema de ambiguidade de escala.
- Foram feitas validações através de experimentos e comparação com trabalhos correlatos, além de ablação.

1.3. Pontos negativos

- Em vez de fundir ou duplicar gaussianas observadas em ambas as vistas de referência, o pixelSplat gera como saída a união de gaussianas previstas para cada vista.
- O pixelSplat não usa modelagem generativa para gerar partes não vistas da cena.
- Quando extendido para muitas vistas de referência, seu mecanismo de atenção epipolar fica muito caro em termos de uso de memória se tornando proibitivo.
- Difícil de reproduzir como pode ser visto na parte de hacker deste documento.
- No texto os autores não explicitam a necessidade de entrar também com a pose das câmeras, além das duas imagens.

1.4. Avaliação

O artigo está bem estruturado com descrição clara do método, experimentos, análise qualitativa e quantitativa, ablação, limitações e código, sendo que seus pontos positivos prevalecem em muito seus pontos fracos, recebendo avaliação 5, ou seja, aceitação.

2. Arqueólog@

Determinar onde este artigo se encaixa no contexto de trabalhos anteriores e posteriores. Você encontrou esse artigo e deve apresentar a ordem cronólogica que o trabalho se encaixa. Sugestão: leia a seção de trabalhos relacionados.

Encontrar e relatar sobre um artigo mais antigo citado pelo artigo atual e um artigo mais recente que cita o artigo atual. Claro, explicar como eles se relacionam.

Além disso avalie se as referências estão adequadas? Liste referências que estão faltando.

Note que quase todos os trabalhos da nossa lista cita o 3D Gaussian Splatting (3DGS) [3] e EWA volume splatting [6]. Logo, o Arqueólog@ precisa ir além dessas referências.

3. Código e experimentos

A arquitetura definida pelo pixelSplat leva inicialmente em consideração duas imagens como entradas e caso seja necessário é possível realizar experimentos com mais, porém, como o próprio autor cita, o custo computacional é acrescido demasiadamente. Além disso, as imagens devem conter suas respectivas matrizes intrínsecas e as poses das câmeras. No entanto isso é extremamente difícil visto que esses parâmetros são normalmente obtidos por métodos como o COLMAP, e para isso é necessário uma sequência grande de imagens.

Desse modo, rodar o código utilizando duas imagens de uma cena que nunca foi visto pelo modelo é desafiador, até por que tudo tem que corroborar com a estruturação dos dados de entrada. Nota-se que no github existem issues perguntado como faz para rodar em duas novas imagens ([issue 100], [issue 86] e outros) e como visualizar as gaussianas do arquivo .ply ([issue 96], [issue 81] e outros). Além disso, não é especificado no artigo que após a geração das nuvens de gaussianas deve-se realizar um corte no eixo Z de modo a remover gaussianas repetitivas que aparecem devido a aprendizagem da distribuição de gaussianas em um dado raio Figure 5. Ademais, o README do github não especifica como gerar a nuvem de gaussianas, porém a Issue 43 nos dá a resposta que é a partir do código presente no arquivo `generate_point_cloud_figure`, para utilizá-lo no dataset re10k precisamos também ter um modelo treinado e escolher a elevação e um azimuth.

Portanto, para ser possível aplicar o método em duas novas imagens de uma cena que não está presente no dataset re10k ou ACID, devemos

1. Obter a matriz intrínsica
2. Obter as poses das câmeras.
3. Transformar os metadados no formato especificado por eles (OPENCV).
4. Gerar os metadados para serem utilizados como entrada.
5. Escolher a elevação e o azimuth
6. Eliminação de gaussianas no plano Z.

Veja na Figure 5 que temos a necessidade de eliminar gaussianas repetitivas, note as repetição do fogão diversas vezes. Esse efeito pode ser visto em diversas outras cenas, os autores do MVSplat [2] disponibilizaram um armazenamento de exemplos que mostra diversas cenas onde isso ocorre.

Para gerar a imagem da Figure 5 utilizamos a base de dados re10k com o modelo treiando nela, seguindo o seguinte comando:

```
python3 -m
src.paper.generate_point_cloud_figure.py
+experiment=re10k
checkpointing.load=ckpt/re10k.ckpt
```

Na Figure 5, utilizamos a cena 4cfefe4588b687a9 que está presente no dataset re10k.

Além disso, não foi possível rodar a avaliação, e os ex-

perimentos de ablation, pois é necessária muita memória gráfica.

O Hacker precisa fornecer um DEMO do paper o mais rápido possível. Logo, deve avaliar a reprodutibilidade do método e implementar uma pequena parte do artigo ou uma versão bem simplificada (eg. 2D em vez de 3D).

- O trabalho poderia ser reproduzido por um ou mais estudantes de pós-graduação? Execute o código do github associado ao paper escolhido e teste em outros datasets;
- Compare as formulas implementadas no código com as equações do paper. Todos os detalhes importantes de algoritmos ou sistemas são discutidos adequadamente?
- "Rode" os experimentos apresentados no paper;
- (Adicional) Pensar em outros experimentos. Discutir a possibilidade no discord primeiro (Participação do Doutorand@).

4. Projeto de doutorado

PixelSplat adota, na representação de cenas tridimensionais, gaussianas tridimensionais primitivas $\{\boldsymbol{g}_k = (\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, \boldsymbol{\alpha}_k, \boldsymbol{S}_k)\}_k^k$, cada uma com seu parâmetro de média $\boldsymbol{\mu}_k$, covariância $\boldsymbol{\sigma}_k$, opacidade $\boldsymbol{\alpha}_k$ e coeficiente harmônico esférico $\boldsymbol{S}_k)\}_k^k$. Essa técnica, explorada no trabalho "3D Gaussian Splatting", do qual o PixelSplat é derivado, apresentou importante otimização na representação de cena tridimensional quando comparado a outras técnicas, como NERF e *voxel grids*.

5. Conclusões

Apresente as conclusões, sugestões de título e um resultado ausente que o artigo poderia ter incluído.

References

[1] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024. 1, 2

[2] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. *MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-view Images*, page 370–386. Springer Nature Switzerland, 2024. 3

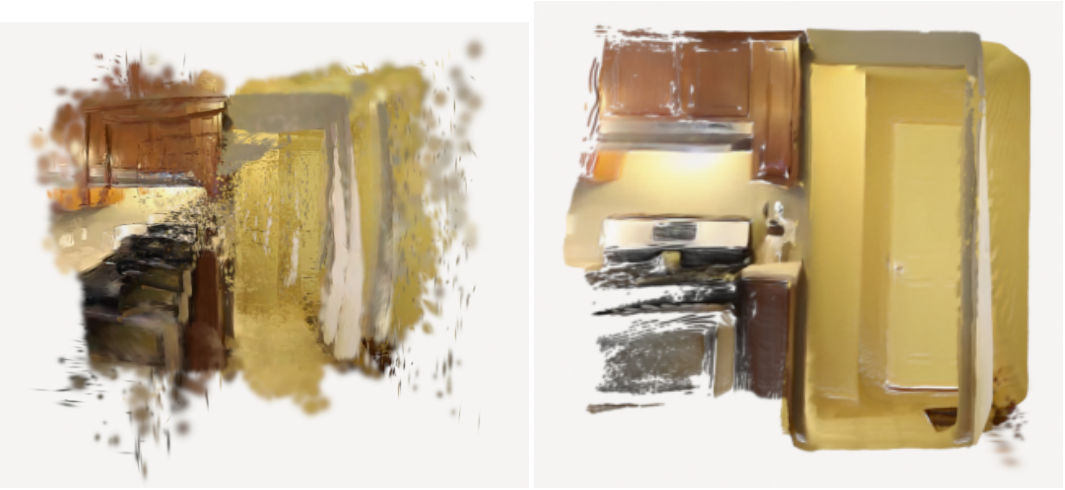
[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 3

[4] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[5] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 1

[6] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001. 3

Figure 5. Cena da cozinha. Na esquerda temos o pixelsSplat com Gaussianas repetitivas e na direita temos o MVSplat.



Fonte: O autor.