Relatório do artigo:
SplaTAM: Splat, Track Map 3D Gaussians for Dense RGB-D SLAM

Revisor Davi Guimarães
UFF
firstauthor@i1.org

Arqueólogo Thiago Baldivieso
IME
thiagojmb@ime.eb.br

Hacker Diana Aldana
IMPA
diana.aldana@impa.br

Doutorando Leonardo Nanci
UFF
leonardonanci@id.uff.br

1. Revisão

Sugestão para o Revis@r: ler as orientações do CVPR. Tentem manter o relatório dentro de 8 páginas.

1.1. Resumo

Descreva brevemente o método e sua contribuição para *visão computacional* ou *computação gráfica*. Forneça sua avaliação sobre o escopo/magnitude da contribuição do artigo. Segue uma sugestão de estrutura para o resumo.

- Problema abordado;
- Motivação;
- Resumo do método;
- Lista de contribuições.

Descreva o método e faça uma ánalise justificando cada formula utilizada. A exposição está clara? Como poderia ser melhorada?

1.2. Pontos positivos

- Ideias interessantes validadas através de experimentalmente e de forma teórica, novas ferramentas, resultados impressionantes, ...
- O que alguém da área aprenderia lendo o paper?

1.3. Pontos negativos

- Falta de experimentos (quais?)
- Alegações enganosas e erros
- Difícil de reproduzir (Participação do Hacker)

1.4. Avaliação

Dê uma classificação geral do trabalho e do artigo em uma escala contínua de 1 a 5, onde 1 é o pior e 5 é o melhor. Especificamente: 1 = Rejeitar, 2 = Possivelmente rejeitar, 3 = Duvidoso, 4 = Possivelmente aceitar, 5 = Aceitar.

Deve ficar claro quais dos pontos positivos e negativos foram mais considerados.

2. Arqueólogo

Trabalhos Anteriores: O SplaTAM é uma evolução das abordagens tradicionais de SLAM, como MonoSLAM e LSD-SLAM, que utilizam representações de pontos e superfícies. Esses métodos têm limitações em termos de eficiência e qualidade de reconstrução, o que motiva a pesquisa em representações volumétricas explícitas, como as 3D Gaussians. O artigo SplaTAM possui atualmente 134 citações de artigos posteriores o consolidando como referência na área de SLAM. A seguir artigos em ordem cronologica que são citados pelo artigo com contribuições importantes que culminaram na base para a criação do SplaTAM.

2.1. Cronologia dos Artigos Relacionados ao SplaTAM

- 1992: Davison et al. - "A theory of active vision."
  Este artigo estabeleceu os fundamentos sobre como sistemas de visão podem interagir ativamente com o ambiente, criando as bases para o desenvolvimento de SLAM.
- 1994: Reid et al. - "Real-time visual tracking of a moving object."

Introduziu técnicas de rastreamento visual em tempo real, essenciais para a localização em SLAM, onde o rastreamento de objetos em movimento é crucial.

- 1995: M. Pollefeys et al. - "A system for real-time 3D reconstruction."
  Descreveu um sistema que permite a reconstrução 3D em tempo real, um passo importante para a criação de mapas em SLAM.
- 1999: D. Fox et al. - "Simultaneous localization and mapping for mobile robots."
  Introduziu o conceito de SLAM, onde um robô pode mapear um ambiente enquanto localiza sua posição dentro dele.
- 2000: K. Konolige et al. - "Real-time 3D reconstruction and tracking."
  Focou na reconstrução 3D e no rastreamento em tempo real, abordando desafios que ainda são relevantes nas técnicas modernas de SLAM.
- 2001: S. Thrun et al. - "A survey of visual SLAM."
  Revisão abrangente das técnicas de SLAM visual, discutindo métodos e desafios, ajudando a consolidar o campo.
- 2007: Davison, A. J., Reid, I. D., Molton, N. D., Stasse, O. - "Monoslam: Real-time single camera slam."
  Introduziu o conceito de SLAM em tempo real usando uma única câmera.
- 2014: Engel, J., Schöps, T., Cremers, D. - "Lsd-slam: Large-scale direct monocular slam."
  Apresentou uma abordagem de SLAM monocular que lida com grandes escalas, ampliando as capacidades do SLAM tradicional.
- 2015: Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., Davison, A. - "Elasticfusion: Dense slam without a pose graph."
  Introduziu um método de SLAM denso que não depende de um grafo de pose, melhorando a robustez do sistema.
- 2015: Whelan, T., Kaess, M., Johannsson, H., Fallon, M., Leonard, J. J., McDonald, J. - "Real-time large-scale dense rgb-d slam with volumetric fusion."
  Focou na fusão volumétrica para SLAM denso em tempo real, contribuindo para a eficiência do mapeamento.
- 2017: Tateno, K., Tombari, F., Laina, I., Navab, N. - "Cnn-slam: Real-time dense monocular slam with learned depth prediction."
  Introduziu o uso de redes neurais convolucionais para prever profundidade em SLAM monocular, melhorando a precisão do mapeamento.
- 2021: Sucar, E., Liu, S., Ortiz, J., Davison, A. - "imap: Implicit mapping and positioning in real-time."
  Apresenta um sistema de mapeamento implícito que melhora a eficiência do SLAM em tempo real.
- 2021: Teed, Z., Deng, J. - "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras."
  Explorou o uso de técnicas de aprendizado profundo para melhorar o desempenho do SLAM em diferentes tipos de câmeras.
- 2023: Goel, K., Tabib, W. - "Incremental multimodal surface mapping via self-organizing gaussian mixture models."
  Introduziu modelos com gaussianas para mapeamento de superfícies, enfatizando a importância de representações probabilísticas.
- 2023: Goel, K., Michael, N., Tabib, W. - "Probabilistic point cloud modeling via self-organizing gaussian mixture models."
  Abordou a modelagem de nuvens de pontos, destacando a importância de representações probabilísticas.
- 2023: Han, X., Liu, H., Ding, Y., Yang, L. - "Romap: Real-time multi-object mapping with neural radiance fields."
  Introduziu o uso de NERFs para mapeamento em tempo real, expandindo as técnicas de SLAM para cenários mais complexos.
- 2023: Johari, M. M., Carta, C., and Fleuret, F. - "Eslam: Efficient dense slam system based on hybrid representation of signed distance fields."
  Apresentou um sistema SLAM denso eficiente, utilizando uma representação híbrida, contribuindo para a eficiência e precisão do mapeamento.
- 2023: Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G. - "3d gaussian splatting for real-time radiance field rendering."
  Descreveu a renderização de campos de radiação em tempo real usando gaussian splatting 3D, que é uma das bases para o SPLATAM.
- 2023: Wang, H., Wang, J., Agapito, L. - "Coslam: Joint coordinate and sparse parametric encodings for neural real-time slam."
  Abordou uma abordagem para SLAM em tempo real, combinando codificações paramétricas esparsas.

Os métodos de SLAM evoluíram significativamente ao longo dos anos, com avanços notáveis nos cálculos fundamentais e tecnologias aplicadas. A base matemática do SLAM envolve a resolução de problemas de otimização, onde o objetivo é minimizar o erro entre as previsões do modelo e as observações reais. Técnicas como o Filtro de Kalman e o Filtro de Partículas são amplamente utilizadas para estimar a posição e a orientação de robôs em ambientes desconhecidos. Diferentes metodologias foram desenvolvidas para abordar os desafios do SLAM, incluindo abordagens baseadas em gráfos, métodos de otimização não linear e técnicas de aprendizado de máquina. Cada metodologia aplicado nos artigos de cronologia contribuiu para melhorar a precisão e a eficiência dos sistemas de SLAM. A técnica de Gaussian Splatting permite a criação

de representações tridimensionais densas, utilizando pontos gaussianos para modelar a distribuição de radiância em uma cena. A renderização diferenciável, por sua vez, facilita a otimização dos parâmetros do modelo, permitindo ajustes finos para melhorar a qualidade das reconstruções. Dentre os artigos fundamentais para geração do SplaTAM temos: "Differentiable Rendering" (Mildenhall et al., 2020), que explora a renderização diferenciável e suas aplicações em representações 3D; "MonoSLAM: Real-time single camera slam" de Andrew J. Davison et al. (2007), que introduziu um sistema de SLAM em tempo real usando uma única câmera; "ORB-SLAM: a Versatile and Accurate Monocular SLAM System" de Raul Mur-Artal et al. (2015), que é um sistema de SLAM monocular que combina características de pontos de interesse com otimização de pose do veículo; "Vox-Fusion: Real-time 3D Reconstruction and Tracking" (Liu et al., 2019), que explora a fusão de voxels para reconstrução em tempo real; "NICE-SLAM: Neural Implicit Continuous Environment SLAM" (Liu et al., 2021), que investiga representações implícitas para SLAM; e "Neural 3D Mesh Renderer" (Kato et al., 2018), que introduz conceitos de renderização diferenciável aplicados a malhas 3D. O SplaTAM é comparado a métodos como Point-SLAM, ORB-SLAM3, NICE-SLAM e Vox-Fusion, destacando suas vantagens em termos de desempenho, precisão na estimativa de pose e qualidade de renderização. Um artigo mais antigo citado é o de D. Fox et al. (1999), que introduz o conceito de SLAM, onde um robô pode mapear um ambiente enquanto localiza sua posição dentro dele. O SplaTAM é mencionado como um avanço significativo na reconstrução de alta fidelidade a partir de uma única câmera RGB-D, além de oferecer renderização rápida e otimização densa. Artigos posteriores que citaram e utilizaram o método SplaTAM como comparação: "Splat-SLAM: Globally Optimized RGB-only SLAM with 3D Gaussians" e "Gaussian-LIC: Real-Time Photo-Realistic SLAM with Gaussian Splatting and LiDAR-Inertial-Camera Fusion" apresentam melhorias significativas em relação ao SplaTAM, integrando dados de LiDAR, IMU e câmeras, e operando em tempo real com qualidade visual superior e otimizações em CUDA para processamento eficiente. Conseguindo melhorias com a redução de ruídos de iluminação e rapidez durante o processamento.

3. Código e experimentos

Os autores compartilharam seu repositório junto com o artigo no seguinte link `https://github.com/spla-tam/SplaTAM`. O README associado ao repositório era um pouco confuso, e havia algumas etapas omitidas que, apesar de óbvias, podem ser difíceis para um novato no tópico seguir. Por outro lado, o repositório é bem particionado, pois cada conjunto de dados e modo de execução tem seus próprios arquivos de configuração, scripts de execução e pastas. Além disso, havia arquivos para baixar e processar os conjuntos de dados de forma organizada. Agora, ao executar os scripts do método, apareceu um erro simples para corrigir e um erro que dificultou o registro online dos resultados. O segundo erro em particular foi bastante demorado, ao contrário de sua solução fácil.

A demonstração para executar o código online foi hospedada no Colab, com todas as instruções e etapas adicionais explicadas no notebook. Esse código pode ser encontrado aqui`https://colab.research.google.com/drive/1whkrKZOmaiZvSR2sP6cDh6rJB2XQ2YAk?usp=sharing`.

A principal perda utilizada pelo SplaTAM é definida pela seguinte expressão,

$$L_t = \sum_{\mathbf{p}} \left(S(\mathbf{p}) > 0.99\right) \left(L_1(D(\mathbf{p})) + 0.5 L_1(C(\mathbf{p}))\right).$$

```
# Depth loss
if use_l1:
    mask = mask.detach()
    if tracking:
        losses['depth'] = torch.abs(curr_data['depth'] - depth)[mask].sum()
    else:
        losses['depth'] = torch.abs(curr_data['depth'] - depth)[mask].mean()

# RGB Loss
if tracking and (use_sil_for_loss or ignore_outlier_depth_loss):
    color_mask = torch.tile(mask, (3, 1, 1))
    color_mask = color_mask.detach()
    losses['im'] = torch.abs(curr_data['im'] - im)[color_mask].sum()
elif tracking:
    losses['im'] = torch.abs(curr_data['im'] - im).sum()
else:
    losses['im'] = 0.8 * l1_loss_v1(im, curr_data['im']) + 0.2 * (1.0 - calc_ssim(im, curr_data['im']))
weighted_losses = {k: v * loss_weights[k] for k, v in losses.items()}
loss = sum(weighted_losses.values())
```

Figure 1. Pedaço de código implementando o termo de perda L_t.

Esta perda considera a máscara S, bem como dois termos de perda sobre a profundidade e a cor. No entanto, a equação não é muito clara quanto ao que exatamente está sendo computado. A Figura 1 mostra o código correspondente à perda L_t. Em particular, o primeiro `if` indica o primeiro termo de perda $L_1(D(\mathbf{p}))$ enquanto o segundo `if` mostra o termo de perda $L_1(C(\mathbf{p}))$. Aqui, fica claro que L_1 significa a norma $l1$ e $D(p)/C(p)$ indica a diferença entre a profundidade/cor do *ground truth* e a profundidade/cor computada.

Finalmente, a Figura 2 apresenta um dos experimentos executados que testou o efeito de escolher cada quadro para otimizar os parâmetros extrínsecos contra escolher a cada cinco quadros. O resultado qualitativo mostra que parece ser melhor escolher a cada cinco quadros, no entanto, a média da precisão mostra que ainda é melhor selecionar cada quadro.

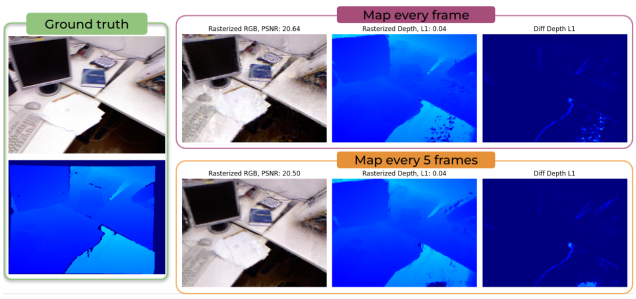

Figure 2. Comparação entre a escolha de cada 1 ou 5 quadros para calcular os extrínsecos da câmera.

4. Projeto de doutorado

No método de SplaTAM, a renderização diferencial das gaussianas é explorada para ajustar os parâmetros da câmera, entre outras variáveis. No artigo, os autores argumentam que técnicas de SLAM podem ser aplicadas no contexto de realidade extendida, para realizar o tracking do dispositivo de visualização em 3D. Nesse cenário, seria interessante explorar a renderização da cena aprendida para exibí-la ao usuário.

Uma característica comum em aplicações de realidade aumentada é a inserção de objetos digitais na visualização do mundo real. Porém, usualmente tais modificações no cenário as alterações da iluminação da cena apresentada são nulas ou demasiadamente simplistas, gerando resultados de aspecto pouco realista.

Como proposta de linha de pesquisa, sugiro estimar a reiluminação de cenas estáticas após a introdução de objetos digitais, tirando proveito da capacidade do método de renderização da cena real, associada à representação por gaussianas cujos parâmetros — como cor — podem ser alterados. A proposta é similar à de Valença et al. [1], que estima o sombreamento originado por objetos digitais em uma imagem. A a proposta aqui sugerida difere do trabalho de Valença et al. devido à introdução de informações do espaço tridimensional representado por primitivas gaussianas. Acredito que, assim, poderia-se melhor estimar as fontes de luz e sua interação com o cenário, possivelmente gerando resultados realistas em 3D — idealmente, em tempo real.

5. Conclusões

O SplaTAM usa um esquema multiestágio que permite construir uma cena e calcular os extrínsecos via SLAM. Ele considera uma nova maneira de usar a equação de renderização para otimizar os extrínsecos da cena, então adiciona gaussianos em regiões com baixa opacidade e finalmente mascara as áreas com muitos gaussianos para treinar apenas aquelas novas regiões da cena. Este algoritmo permite acelerar a renderização enquanto mapeia o movimento da câmera. Como resultado disso, SplaTAM se posiciona como uma contribuição significativa no campo do SLAM, construindo sobre os fundamentos estabelecidos por trabalhos anteriores e influenciando pesquisas futuras.

References

[1] Lucas Valença, Jinsong Zhang, Michaël Gharbi, Yannick Hold-Geoffroy, and Jean-François Lalonde. Shadow harmonization for realistic compositing. In *ACM SIGGRAPH Asia 2023 Conference Proceedings*, 2023. 4