

# Efficient Point Cloud Rasterization for Real Time Volumetric Integration in Mixed Reality Applications

Autores : Christian Kunert, Tobias Schwandt, Wolfgang Broll  
Ilmenau University of Technology

# Introdução

- Contexto

- Técnicas de **SLAM** juntamente com a reconstrução volumétrica em **TSDF** têm sido amplamente empregadas para gerar representações tridimensionais do ambiente em tempo real.

- Problemas

- Desafios de desempenho em reconstruções 3D em tempo real pois não tem uma seleção inteligente de regiões a serem atualizadas, evitando processamento desnecessário.

- Objetivo

- Reconstrução em tempo real utilizando uma etapa simples de pré-processamento nos dados de entrada para considerar adequadamente a representação **TSDF**.

# TSDF

- Representação volumétrica que armazena distâncias truncadas entre pontos no espaço e superfícies próximas.
- Facilita a fusão de dados para reconstruções 3D pois diferentemente do SDF limita as distâncias entre 1 e -1.

-0.9	-0.4	-0.1	0.2	0.9	1	1	1	1	1
-1	-0.9	-0.2	1	0.5	0.9	1	1	1	1
-1	-0.9	-0.3	0.1	0.2	0.8	1	1	1	1
-1	-0.9	-0.4	0.2	0.2	0.8	1	1	1	1
-1	-1	-0.8	-0.1	0.2	0.6	0.8	1	1	1
-1	-0.9	-0.3	-0.6	0.3	0.7	0.9	1	1	1
-1	-0.9	-0.4	-0.1	0.3	0.8	1	1	1	1
-0.9	-0.7	-0.5	0	0.4	0.9	1	1	1	1
-0.1	-0.9	-0.9	-0.1	0.4	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

# KinectFusion

- Introduziu um pipeline que se tornou padrão para muitas abordagens de reconstrução e SLAM.
- Não escala bem devido à dependência de uma única grade fixa (que contém principalmente espaço vazio). Reconstruções maiores exigem grande quantidade de memória ou resultam em baixa qualidade.

## Pipeline de reconstrução, para cada frame novo de profundidade

- Conversão do mapa de profundidade, gerando um normal map na GPU
- Estimativa de Pose da câmera (ICP)
- Atualização dos pesos do TSDF
- Raycasting, gera a visualização do modelo a partir da pose calculada

- Moving Volume KinectFusion
- Kintinuous
- Sparse Voxel Octrees (SVOs)
- Flat tree
- MonoFusion
- Voxel Hashing

## Voxel Hashing

- Eficiência de memória
- Escalabilidade
- Paralelismo pela GPU

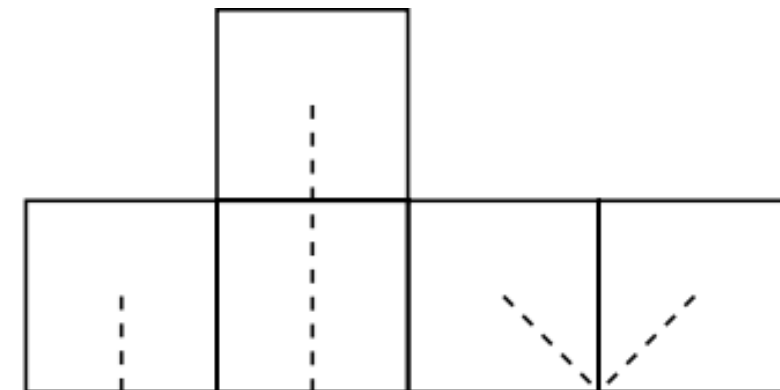
## Sparse Voxel Octrees & Flat Tree

- Apresentação Hierárquica
- Baixo desempenho em GPUs
- Multi resolução
- Sobrecarga de ponteiros

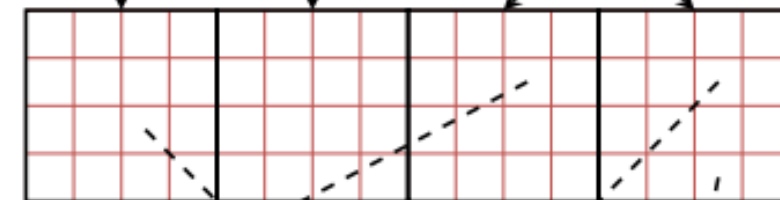
## Data Structure

- Comparado com outras estruturas em árvore não tem o sobrecarregamento de ponteiros
- Alocação de memória que apontam para os grids filhos
- Cada cor e peso são armazenados nos “leaf grids”

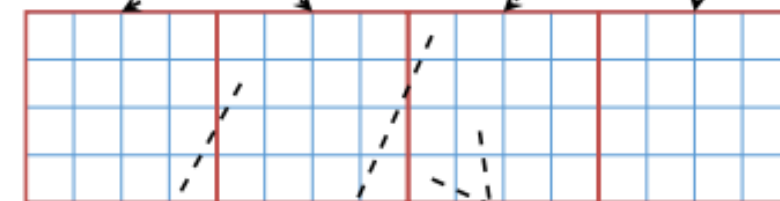
Root volumes



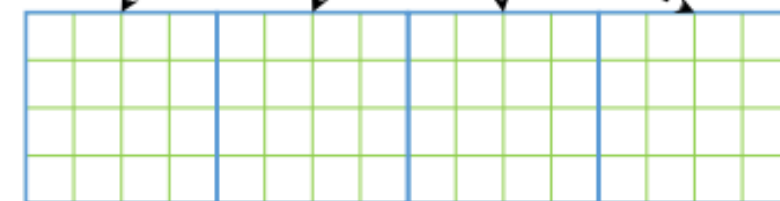
Root grids



Internal grids



Leaf grids

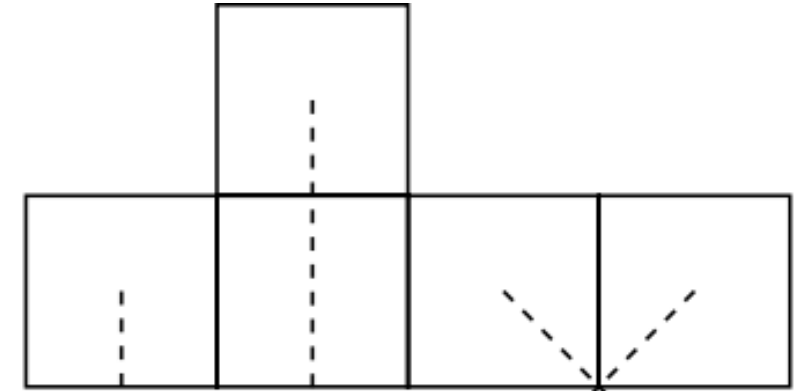




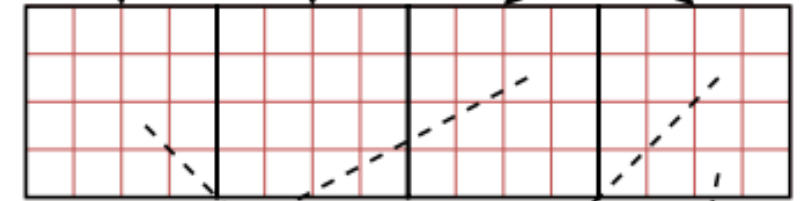
## Data Structure

- A GPU processa apenas grids relevantes (folhas e internos), evitando regiões vazias.
- Permite reconstruções em ambientes extensos sem sobrecarregar a memória.

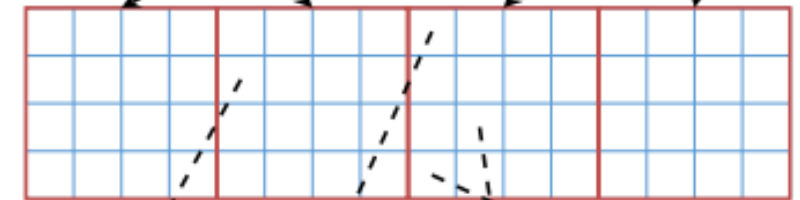
Root volumes



Root grids

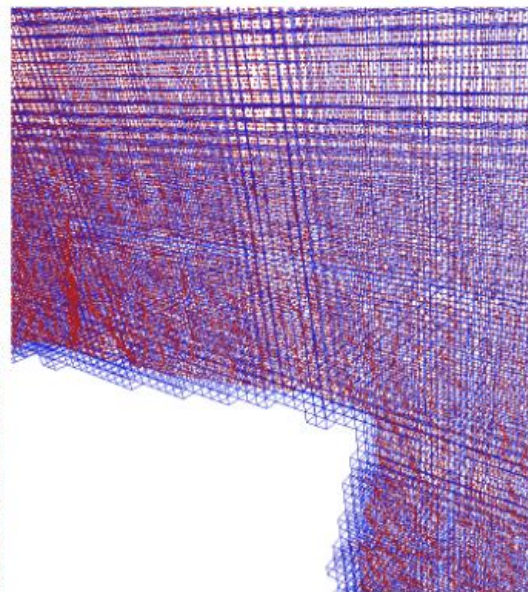
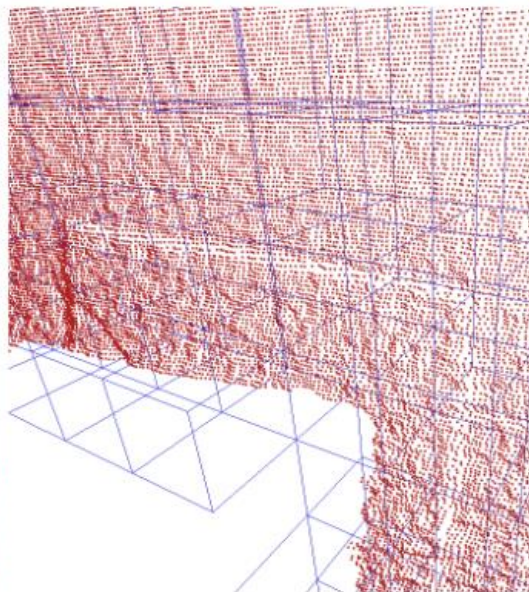
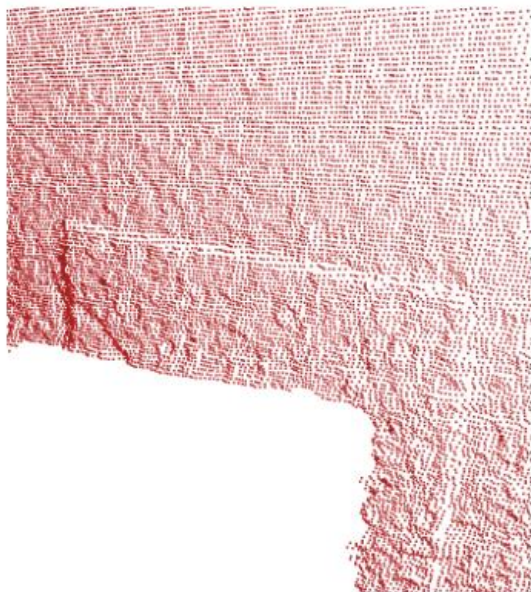


Internal grids



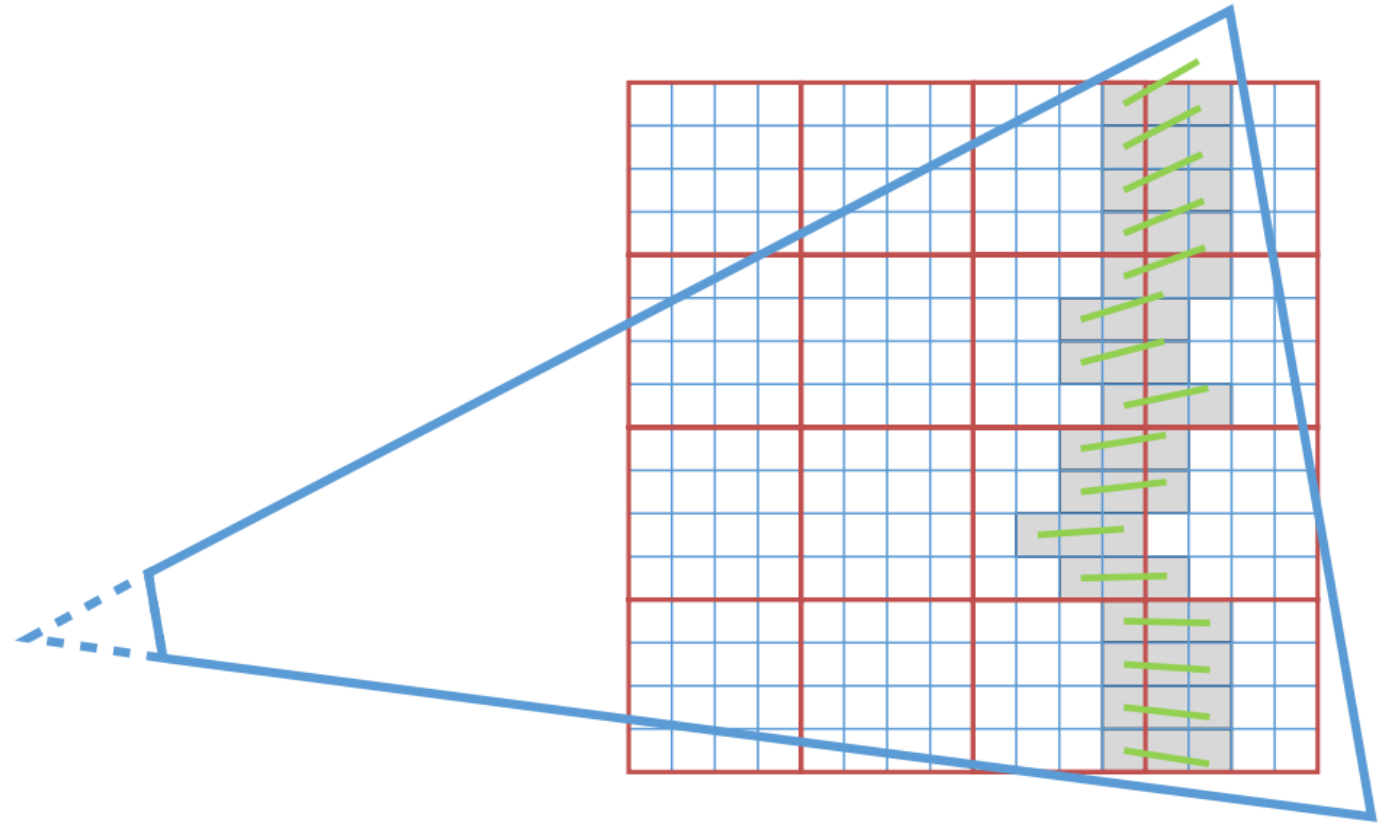
Leaf grids





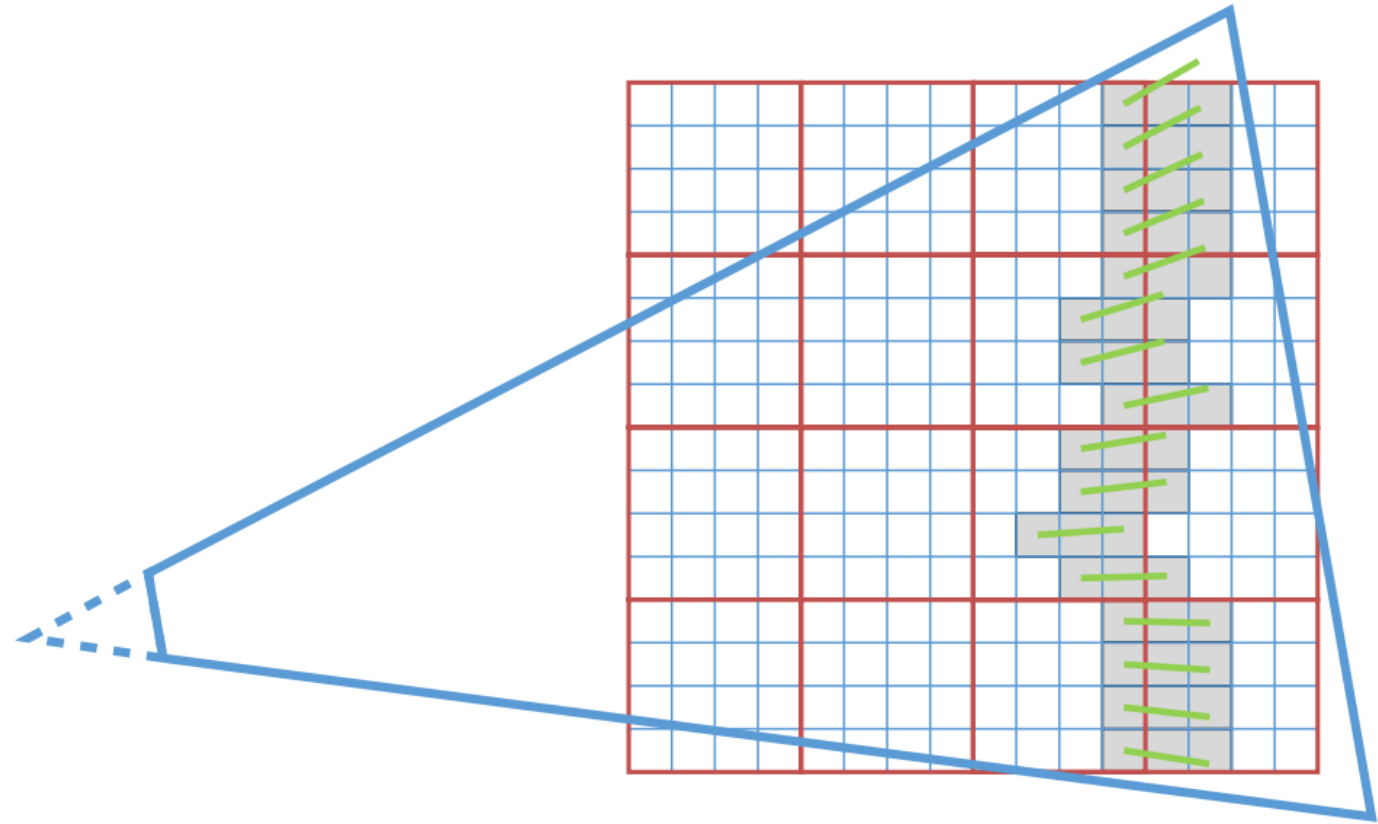
# Rasterization

- Identificar Voxels afetados que intersectam qualquer ponto da nuvem de pontos
- Criar uma fila com os voxels afetados.
- Atualizar o TSDF das leaf grids



# Rasterization

- Pré-processamento da Nuvem de Pontos
- Projeção em Grades Intermediárias:
- Rasterização Conservadora:
- Identificação de Voxels Relevantes:

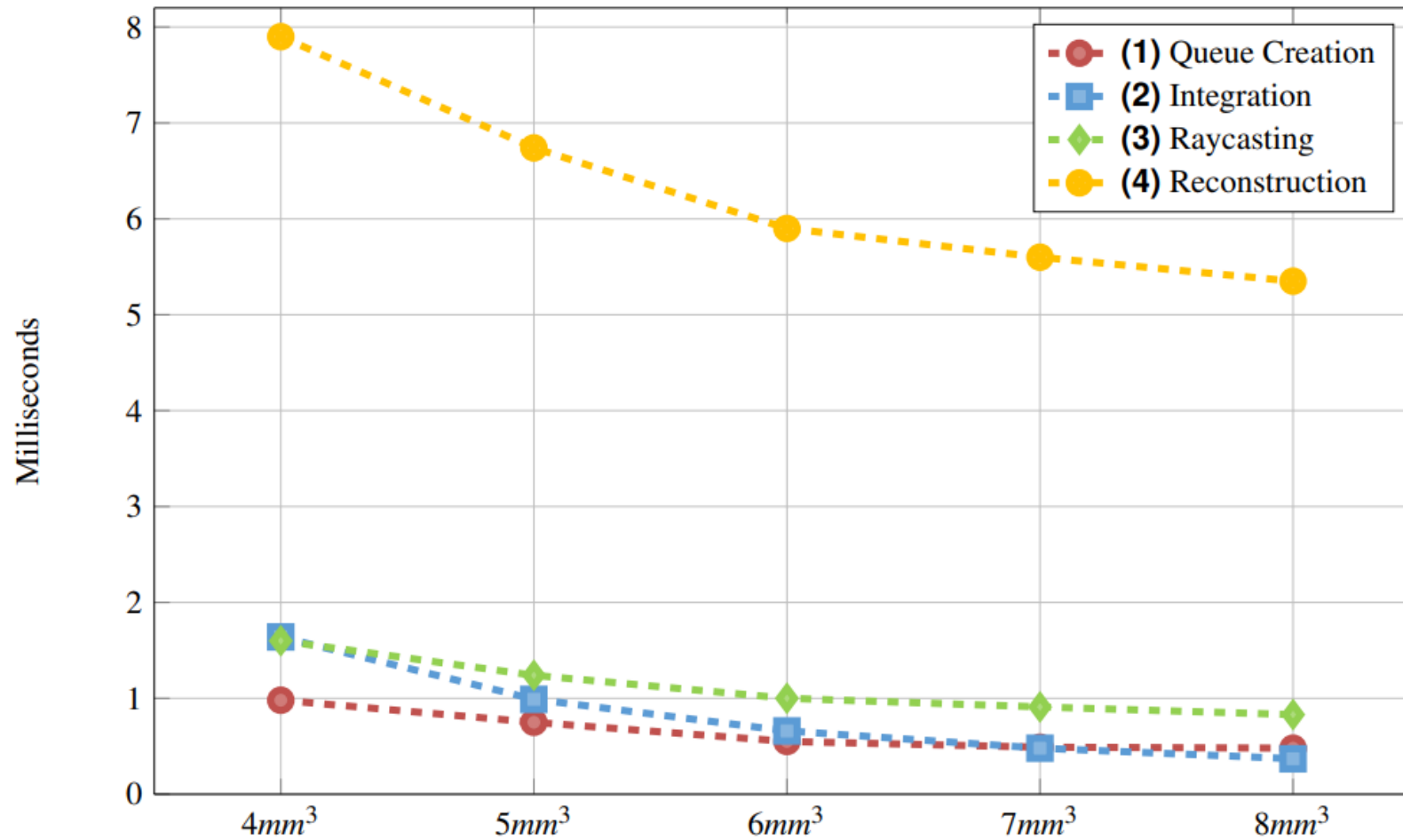


# Raycasting

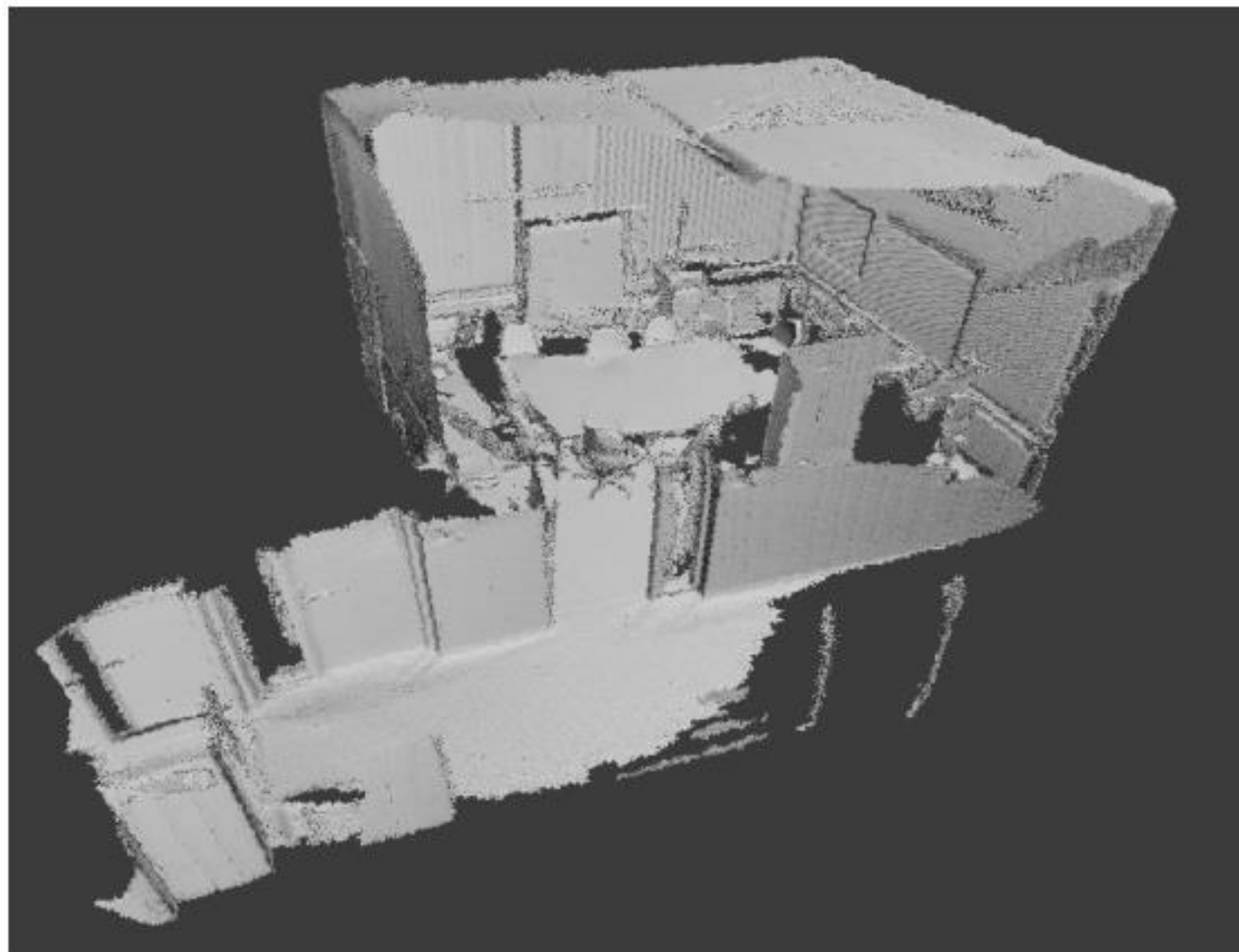
- Um raio é traçado a partir da posição da câmera e percorre o volume até sair da **AABB** da reconstrução ou encontrar uma superfície (detectada por uma mudança de sinal na **TSDF**).
- Feedback para ICP: A imagem gerada pelo raycasting é usada no algoritmo **ICP** para ajustar a pose da câmera.
- Grids com peso baixo (ruído ou dados recentes) são temporariamente ignorados, melhorando a eficiência.

# Resultados

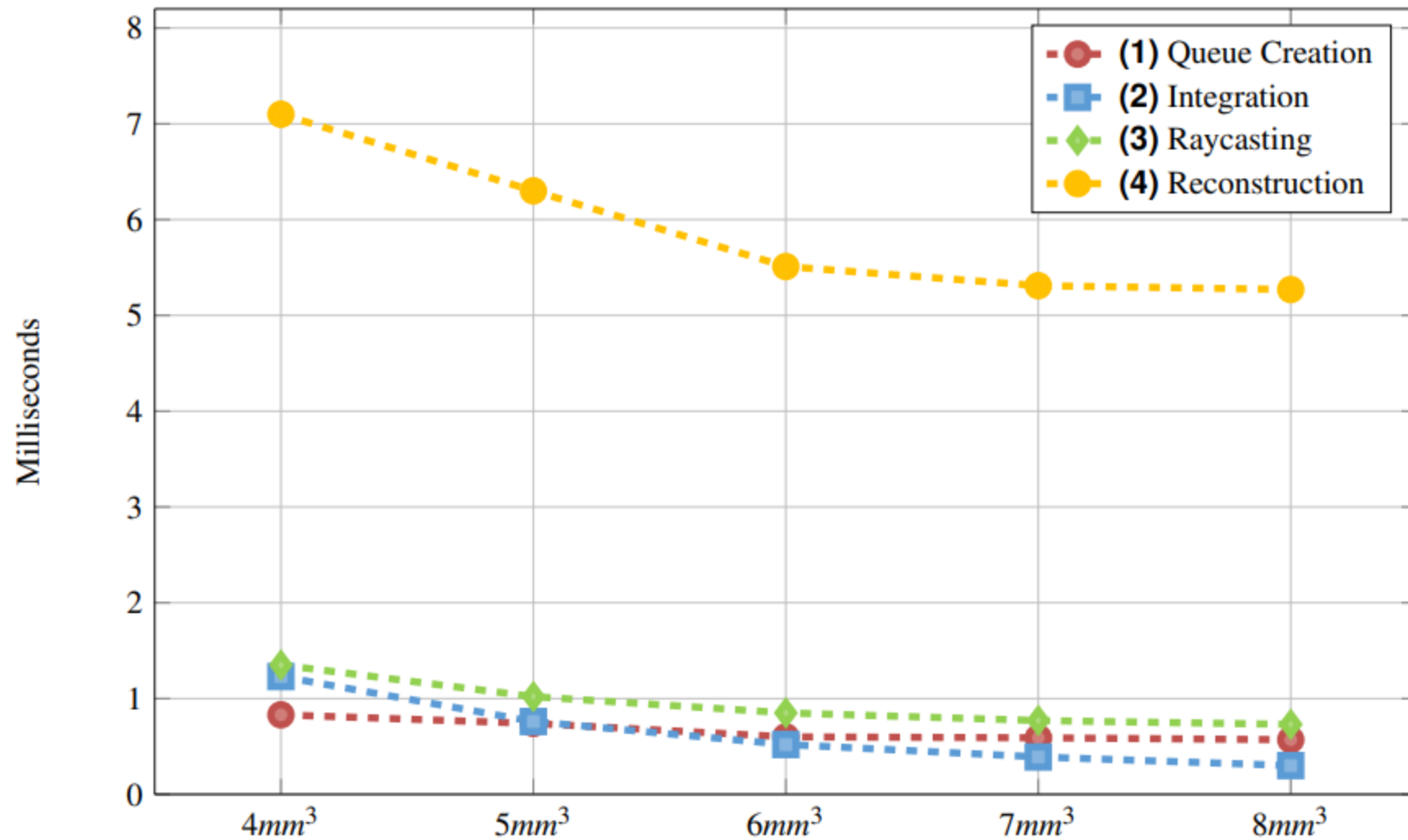
Average performance of different voxel sizes over 2000 frames. (N=2000)



# Resultados

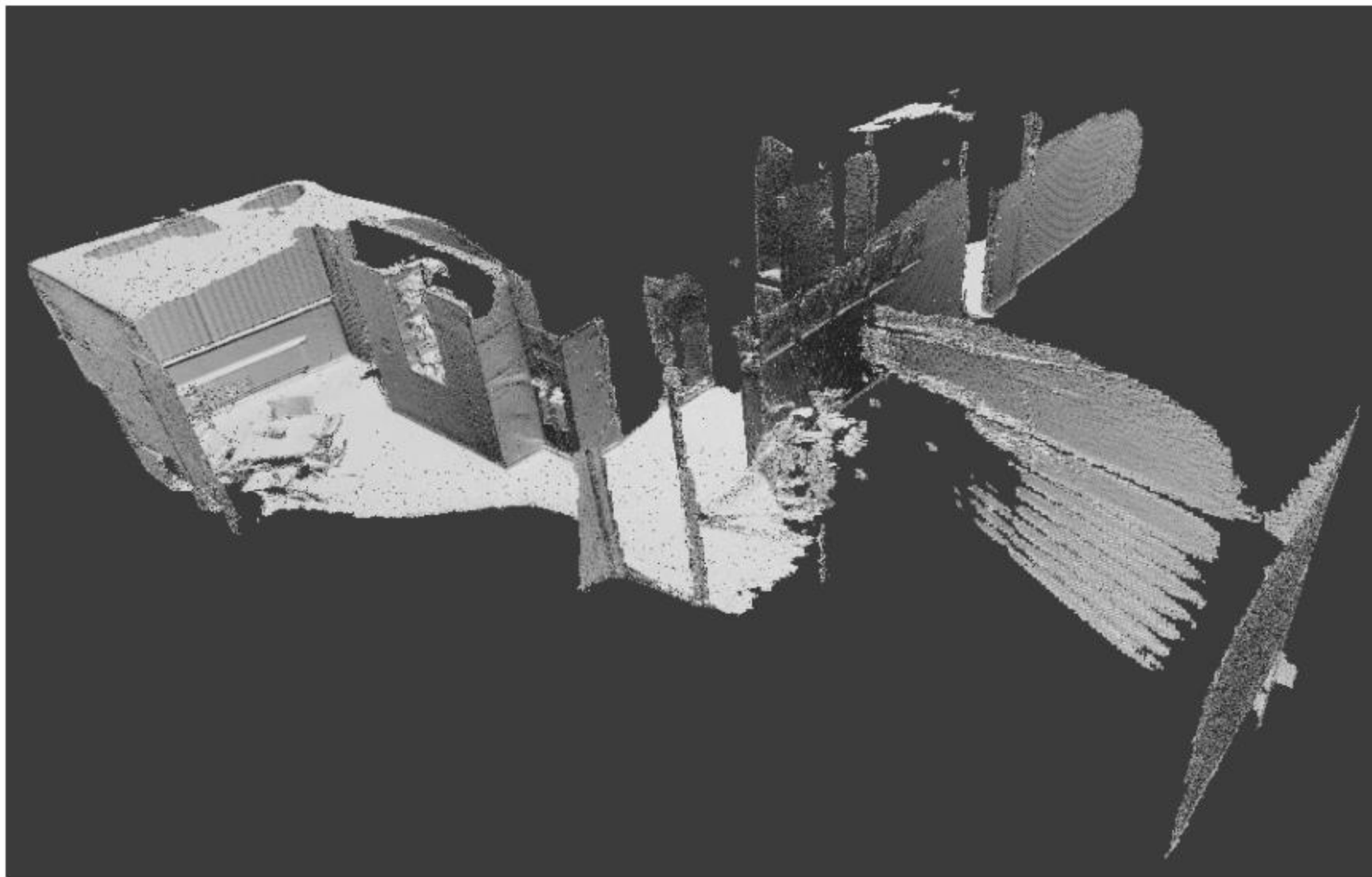


# Resultados





# Resultados



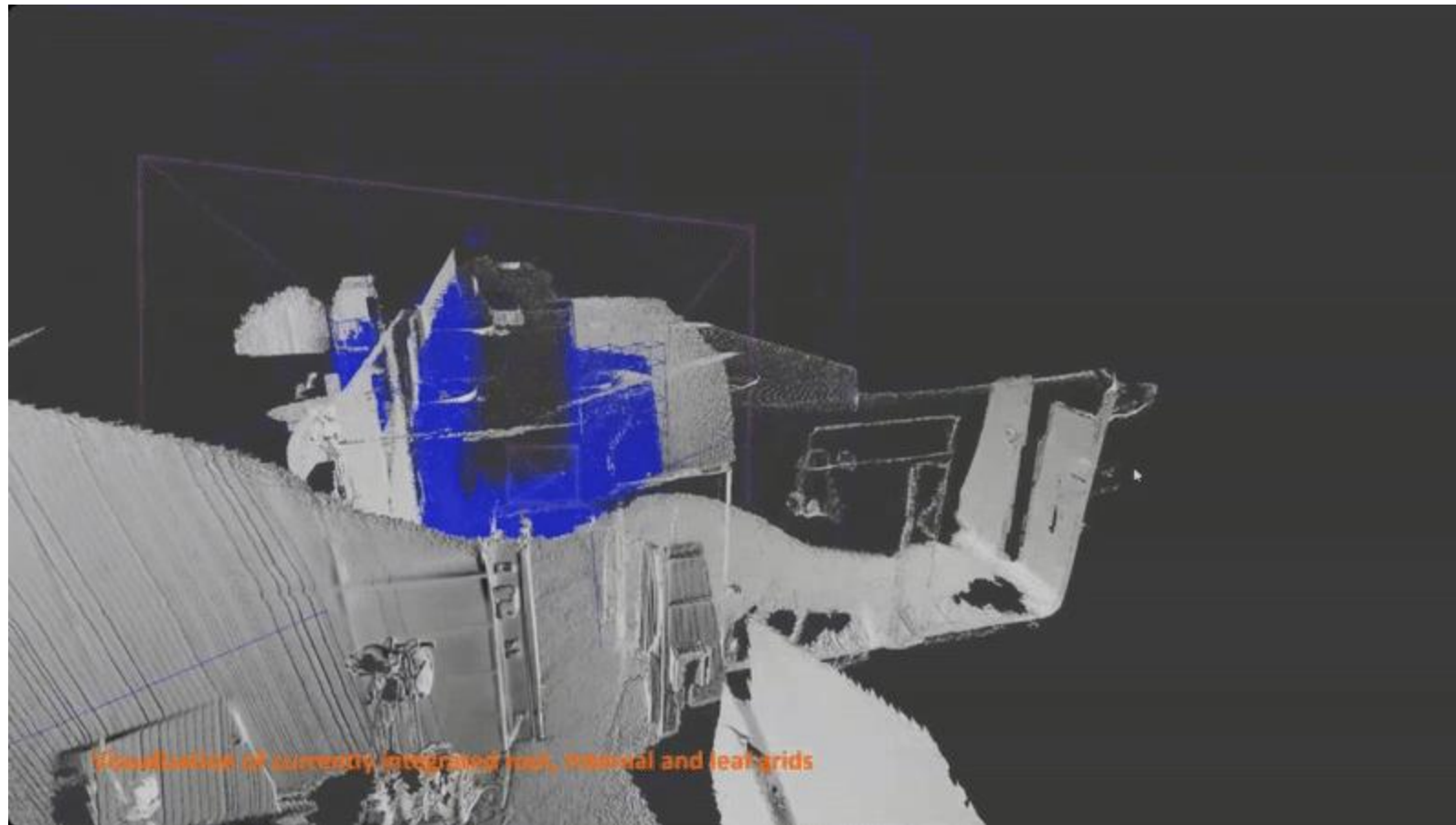
# Resultados

- OpenGL 4.5
- Kinect V2
- Intel i7-6820HK, 16 GB de RAM e Nvidia GeForce GTX 1070
- Grids com 16, 8, 8 =  $(1024^3)$  root grid
- 4 mm<sup>3</sup>: Alta precisão, mas maior consumo de memória e tempo de processamento.
- 8 mm<sup>3</sup>: Qualidade aceitável, menor uso de memória e alta velocidade.

## Conclusões

- O artigo apresenta uma abordagem eficiente para integração de nuvens de pontos em estruturas hierárquicas de voxels, utilizando rasterização rápida e técnicas de reconstrução volumétrica em tempo real.
- A técnica proposta demonstrou ser capaz de lidar com grandes volumes de dados, mantendo alta taxa de quadros (~30 FPS) e consumo moderado de memória.
- A hierarquia em árvore usada na estrutura permite ignorar subárvores irrelevantes, otimizando o desempenho geral e reduzindo o número de cálculos necessários.
- O método mostrou-se adequado para aplicações em Mixed Reality (MR), oferecendo reconstruções precisas e escaláveis, mesmo com hardware consumidor.

<https://youtu.be/0fYI8eQq6YM?t=110>



# Referências

- Kunert, C., Schwandt, T., and Broll, W. (2018). Efficient point cloud rasterization for real time volumetric integration in mixed reality applications, in 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (Munich), 1–9. doi: 10.1109/ISMAR.2018.00023
- PCL. **Using Kinfu Large Scale to generate a textured mesh.** Disponível em: [https://pcl.readthedocs.io/projects/tutorials/en/master/using\\_kinfu\\_large\\_scale.html](https://pcl.readthedocs.io/projects/tutorials/en/master/using_kinfu_large_scale.html). Acesso em: 11 abr. 2025.
- VWDS. **Efficient Point Cloud Rasterization for Real Time Volumetric Integration in Mixed Reality Applications.** YouTube, 16 out. 2018. Disponível em: <https://www.youtube.com/watch?v=0fYl8eQq6YM>. Acesso em: 11 abr. 2025.