FAST GPU GENERATION OF DISTANCE FIELDS FROM A VOXEL GRID

by

NICOLAS FEDOR
URN: 6683787

A dissertation submitted in partial fulfilment of the requirements for the award of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

May 2025

Department of Computer Science
University of Surrey
Guildford GU2 7XH

Supervised by: Joey Sik Chun Lam

I declare that this dissertation is my own work and that the work of others is acknowledged and indicated by explicit references.

Nicolas Fedor
May 2025

Abstract

Write a summary of the work presented in your dissertation. Introduce the topic and highlight your main contributions and results. The abstract should be comprehensible on its own, and should no contain any references. As far as possible, limit the use of jargon and abbreviations, to make the abstract readable by non-specialists in your area. Do not exceed 300 words.

Acknowledgements

Write any personal words of thanks here. Typically, this space is used to thank your supervisor for their guidance, as well as anyone else who has supported the completion of this dissertation, for example by discussing results and their interpretation or reviewing write ups. It is also usual to acknowledge any financial support received in relation to this work.

Contents

List of Figures

List of Tables

Glossary

P	Placeholder

Abbreviations

SVO	Sparse Voxel Octree
JFA	Jump Flooding Algorithm
CPU	Central Processing Unit
GPU	Graphics Processing Unit

Chapter 1

Introduction

In recent years, real-time computer graphics applications have increasingly adopted distance fields as a fundamental representation for rendering and physics simulations (Jones & Satherley 2001). Distance fields, which encode the minimum distance from any point to the nearest surface, provide an elegant solution for various graphics operations including collision detection (Fuhrmann, Sobotka & Groß 2003), soft shadows (Tan, Chua, Koh & Bhojan 2022), and ambient occlusion (Wright 2015). While techniques exist for generating distance fields in real-time from a triangle mesh (Kramer 2015), techniques covering distance field generation from discrete voxel data are uncommon.

1.1 Problem Statement

Current approaches to distance field generation from voxel grids present various tradeoffs that limit their effectiveness in dynamic scenes. The Jump Flooding Algorithm (JFA) (Rong & Tan 2006, Rong & Tan 2007, Wang, Ino & Ke 2023), while efficient for parallel computation, introduces accuracy issues particularly at larger distances from surfaces and near feature edges. Scan, or prefix sum-based, approaches (Erleben & Dohlmann 2008) provide accurate results but suffer from inherent sequential dependencies that limit GPU parallelization. Wavefront propagation methods can efficiently update local regions but may struggle with concurrent updates in complex scenes (Teodoro, Pan, Kurc, Kong, Cooper & Saltz 2013).

Common optimization strategies, such as spatial partitioning into smaller chunks for localized updates (Naylor 1992), introduce their own challenges including boundary artifacts and increased memory management overhead. While these techniques work well in isolation for specific use cases, there remains a fundamental gap in solutions that can handle arbitrary dynamic scene modifications while maintaining both accuracy and performance. This research investigates whether a novel hybrid approach—combining elements of existing techniques or developing new algorithmic patterns—could better address these challenges.

1.2 Aims and Objectives

This research aims to develop and evaluate novel GPU-based techniques for rapid distance field generation from voxel grid representations. The primary objectives are:

- To analyze and classify existing approaches to distance field generation, with particular focus on GPU-accelerated methods.
- To develop new algorithms that optimize the conversion process from voxel grids to distance fields.
- To implement and validate these algorithms on modern GPU architectures.
- To establish a comprehensive comparison framework for evaluating different distance field generation techniques.

1.2.1 Performance Metrics

The evaluation of the proposed methods will be conducted against sparse voxel octree implementations, which currently represent the state-of-the-art in many graphics applications. Key performance metrics include:

1. Computation time for initial distance field generation.
2. Memory consumption during generation and storage.
3. Update latency for localized geometric changes.
4. Scalability with increasing voxel grid resolution.
5. Accuracy of distance field values compared to analytical solutions.
6. GPU resource utilization, including memory bandwidth and compute occupancy.

1.3 Scope and Limitations

While this research addresses the core challenge of distance field generation, several related aspects fall outside its scope:

- The optimization of ray marching techniques for distance field rendering.
- The development of new compression methods for distance field storage.
- The optimization of the underlying renderer, which will include features like:
  - Memory management between the CPU and GPU.
  - Sychronization with the windowing framework.
  - Optimizing presentation of ray marching output image.

The focus remains specifically on the GPU-based generation process and its performance characteristics in dynamic scenarios. The research assumes access to modern GPU hardware and primarily targets real-time graphics applications where frequent distance field updates are required.

A sample "Falling Sand" graphics application will be implemented that will serve as a demo and benchmark for real-time performance of the distance field regeneration. The simulation itself will not be optimized and include a bare minimum of sand and water voxels that can fall and spread out in the world.

Bibliography

Erleben, K. & Dohlmann, H. (2008), 'Signed distance fields using single-pass gpu scan conversion of tetrahedra', *Gpu Gems* **3**, 741–763.

Fuhrmann, A., Sobotka, G. & Groß, C. (2003), Distance fields for rapid collision detection in physically based modeling, *in* 'Proceedings of GraphiCon', Vol. 2003, Citeseer, pp. 58–65.

Jones, M. W. & Satherley, R. (2001), Using distance fields for object representation and rendering, *in* 'Proc. 19th Ann. Conf. of Eurographics (UK Chapter)', London, pp. 37–44.

Kramer, L. (2015), Real-time sparse distance fields for games, *in* 'Game Developers Conference'.

Naylor, B. F. (1992), Interactive solid geometry via partitioning trees, *in* 'Proc. Graphics Interface', Vol. 92, pp. 11–18.

Rong, G. & Tan, T.-S. (2006), Jump flooding in gpu with applications to voronoi diagram and distance transform, *in* 'Proceedings of the 2006 symposium on Interactive 3D graphics and games', pp. 109–116.

Rong, G. & Tan, T.-S. (2007), Variants of jump flooding algorithm for computing discrete voronoi diagrams, *in* '4th international symposium on voronoi diagrams in science and engineering (ISVD 2007)', IEEE, pp. 176–181.

Tan, Y. W., Chua, N., Koh, C. & Bhojan, A. (2022), 'Rtsdf: Real-time signed distance fields for soft shadow approximation in games', *arXiv preprint arXiv:2210.06160* .

Teodoro, G., Pan, T., Kurc, T. M., Kong, J., Cooper, L. A. & Saltz, J. H. (2013), 'Efficient irregular wavefront propagation algorithms on hybrid cpu–gpu machines', *Parallel computing* **39**(4-5), 189–211.

Wang, J., Ino, F. & Ke, J. (2023), Prf: A fast parallel relaxed flooding algorithm for voronoi diagram generation on gpu, *in* '2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)', IEEE, pp. 713–723.

Wright, D. (2015), Dynamic occlusion with signed distance fields, *in* 'ACM SIGGRAPH', Vol. 3.