Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Implementation of a Real-time Streaming-based Terrain Level of Detail System

Bachelor Thesis by

Amar Tabakovic

Bern University of Applied Sciences
Engineering and Information Technology
Computer Perception & Virtual Reality Lab

Supervisor
Prof. Marcus Hudritsch

External expert
Dr. Eric Dubuis

March 19, 2024

Abstract

Terrains are an important part of various practical applications of computer graphics, such as video games, flight simulators and geographical information systems (GIS). Since terrains are expensive to render, various terrain level of detail (LOD) approaches have been developed over the last three decades. Terrain LOD algorithms improve the performance of terrain rendering by applying various optimizations, such as rendering distant areas in a lower resolution, not rendering areas out of view (view-frustum culling), and more. Besides the rendering performance, another important aspect to terrain rendering is the size of terrain data. Terrain data too large to fit entirely in memory must be streamed from the disk or over the network.

This thesis describes the implementation of a real-time streaming-based terrain LOD system, which is capable of streaming in data from various providers based on the XYZ map tiling scheme. The implemented terrain LOD algorithm is based on Chunked LOD by Ulrich TODO cite.

Contents

List of Tables

List of Figures

Listings

Chapter 1

Introduction

Terrains are an important part of many practical applications of 3D computer graphics. They can be found in video games, simulation software and geographical information systems (GIS). Terrains are, however, due to their constant visibility and sheer size expensive to render. Over the last three decades, efficient terrain rendering has been the topic of research and numerous terrain rendering algorithms and approaches spawned.

Another important problematic aspect of terrain rendering is the efficient and real-time storage, organisation and usage of terrain data. In order to support terrains which cannot fit entirely in memory, various terrain paging and streaming approaches were developed and published over the last few years as well.

1.1 Goals of this Thesis

1.2 Intended Readership

The reader is assumed to be familiar with the basics of computer science, C++ programming and 3D computer graphics.

1.3 Structure of this Thesis

This thesis is structured as follows:

- Chapter 2 introduces the reader to various topics covered in this thesis, such as terrain LOD rendering. In addition, basic terminology and notation is defined.
- Chapter 3 gives an overview of previous work conducted in the area of real-time terrain streaming and a short recap of the preceeding project "3D Terrain with Level of Detail".
- Chapter 4

- Chapter 5
- Chapter 6
- Chapter 7
- Chapter 8
- Chapter 9
- Appendix A contains various artifacts related to the project management of this thesis.
- Appendix B contains the usage guide for Streaming-ATLOD with installation and building steps.

TODO dynamic links

Chapter 2

Theoretical Background

This chapter describes some background information on computer graphics, terrain LOD and network programming.

2.1 Notation and Terminology

2.1.1 Mathematical Notation

2.1.2 Terminology

2.2 Terrain Rendering

Terrain rendering refers to the process of displaying the terrain on the screen.

2.2.1 Data Representation

Terrain data can be represented in a number of ways.

2.2.2 Level of Detail

Level of Detail (LOD) is the concept of reducing the complexity of a mesh using various metrics to optimize rendering performance. Such metrics include the distance of the mesh to the camera, the dimensions of the mesh in screen-space and more.

Terrain LOD algorithms can be categorized as follows.

Some of the most important terrain LOD algorithms are summarized in chapter TODO.

2.3 Terrain Streaming

For visualizing very large terrains, the main memory is often not large enough to hold the entire terrain data. In this case, terrain rendering systems often deploy

terrain streaming (also called *terrain paging*), which refers to the concept of dynamically loading and offloading terrain data from the disk or from a network, such as the internet.

Chapter 3

Previous Work

This chapter aims to present ideas from already existing approaches. First, the author's predecessor project on basic terrain LOD rendering is summarized. Afterwards, an overview of literature and software systems for terrain streaming/paging is given, in which their central ideas are outlined.

3.1 Review of Project 2 – 3D Terrain with Level of Detail

This thesis is the logical continuation of the author's preceeding project *3D Terrain with Level of Detail* [Tab24] as part of the course "Project 2" at the Bern University of Applied Sciences. In said project, the basics of terrain rendering and existing approaches were studied and evaluated and a simple terrain renderer named *ATLOD* was developed. A treatment of terrain rendering with streaming/paging was explicitly not part of the preceeding project. This section aims to summarize the most important parts of the preceeding project.

3.1.1 Implemented Terrain LOD Algorithm

The implemented LOD algorithm is mainly based on GeoMipMapping and certain elements from GPU-based Geometry Clipmaps TODO cite. The terrain gets split up into logical blocks of size $blockSize = 2^n + 1$ for some $n \in \mathbb{N}$, where each block contains information such as current LOD level, its world-space center point, the extent points of its AABB, current border permutation, and more. A single grid-like flat mesh of side length $blockSize$ is stored on the vertex and index buffer of the terrain. The indices are stored such that the flat mesh is split into its center and border area. The first part of the index buffer contains indices of the center area at every LOD resolution, from highest to lowest, and the second part of the index buffer contains the indices of the border area at every LOD resolution and for every of the $2^4 = 16$ possible border permutations. A border permutation is a 4-tuple (l, r, t, b) (the variables corresponding to left, right, top and bottom) representing a border area, set up such that each element of the 4-tuple is set to 1 if the neighboring block on

the corresponding side has a lower LOD level, otherwise 0. The arrangement of indices with border permutations is for preventing cracks from occurring in the terrain. The heightmap is stored as a texture image on the GPU.

At runtime, in a first step the LOD level and current border permutation of each block is updated. The new LOD level is calculated using the 3-dimensional distance between the camera and the block's center point. In a second step, each block is intersected with the view-frustum and if they intersect, the block gets rendered with two draw-calls. In the vertex shader, the flat mesh gets translated by the block's world-space center point and the heightmap is sampled for a y-value, which is then used to displace the y-coordinate of the vertex. In the fragment shader, the overlay texture is applied and Phong shading is calculated. The normal vectors for Phong shading are calculated using the four orthogonally neighboring points in the heightmap. Finally, a simple distance fog is applied.

3.1.2 Strengths and Limitations

The main strengths of the implemented algorithm are its low GPU memory usage.

On the other hand, the main limitations are that it lacks vertex morphing for preventing pops during LOD level changes, the organization of blocks into a quadtree for more efficient view-frustum culling and some other optimizations such as instanced rendering.

3.2 Literature

Real-time terrain data streaming has been the topic of research, albeit with a lower focus in comparison to terrain LOD algorithms [Pro].

3.2.1 Level of Detail for 3D Graphics

Subchapter "7.2.5 Paging, Streaming, and Out of Core" in the the book *Level of Detail for 3D Graphics* by Luebke et al. describes some basic background, existing approaches and relevant publications related to terrain streaming. Although the book was published in the year 2003 and can be considered somewhat outdated today, certain basic ideas might still hold today.

3.2.2 Geometry Clipmaps

GPU-based Geometry Clipmaps by Asirvatham and Hoppe is a terrain LOD algorithm based on nested grid-like rings centered around the camera.

3.2.3 Adaptive Streaming and Rendering of Large Terrains: A Generic Solution

https://dspace5.zcu.cz/bitstream/11025/10885/1/Lerbour.pdf

3.2.4 CDLOD

Strugar describes some basic information on how the CDLOD algorithm can be extended to work with streaming [Str09]. Additionally, his public implementation of the CDLOD algorithm contains a streaming-based variant (*StreamingCDLOD*), which dynamically loads and offloads terrain tiles based on the user's movement TODO cite.

3.3 Video Games and Game Engines

3.3.1 Unreal Engine

TODO check out documentation https://docs.unrealengine.com/5.3/en-US/level-streaming-in-unreal-engine

3.3.2 Far Cry 5

TODO check out https://www.gdcvault.com/play/1025480/Terrain-Rendering-in-Far-Cry

3.3.3 Microsoft Flight Simulator

Microsoft Flight Simulator is a video game developed by Asobo Studio and released in 2020. The video game allows the player to fly on the entire earth and uses various data sources for representing the earth as accurately as possible. It received high praise for its TODO.

At the GDC 2022, Fuentes presented the terrain system of Microsoft Flight Simulator, in which he described the data organization, system architecture, rendering process and many more aspects.

The terrain system uses a number of sources for height data, overlay texturing, vegetation, bodies of water, cities and more.

TODO check out https://www.gdcvault.com/play/1027581/Advanced-Graphics-Summit-Designing-the

3.3.4 Battlefield 3

TODO check out https://media.contentapi.ea.com/content/dam/eacom/frostbite/files/gdc12-terrain-in-battlefield3.pdf

3.3.5 Red Engine 3

TODO Check out https://www.gdcvault.com/play/1020197/Landscape-Creation-and-Rendering-in

3.4 Geographic Information Systems

3.4.1 OpenWebGlobe

OpenWebGlobe is a project and software development kit written in JavaScript and WebGL for rendering 3D globes on web browsers. It was developed in TODO by TODO.

3.4.2 Google Earth

Google Earth is a 3D globe viewer by Google which is capable of rendering the earth, utilizing a number of data sources for its imagery and height data, which include Sentinel, TODO.

Chapter 4

StreamingATLOD

4.1 Data Streaming

4.2 Terrain Tile Caching

4.3 Terrain LOD Algorithm

Chapter 5

Results

5.1 StreamingATLOD

5.2 Terrain Streaming Server

Chapter 6

Discussion

Chapter 7

Conclusion

Bibliography

[Pro] Virtual Terrain Project. Terrain lod: Runtime regular-grid approaches. `http://vterrain.org/LOD/other.html`.

[Str09] Filip Strugar. Continuous distance-dependent level of detail for rendering heightmaps. *J. Graphics, GPU, & Game Tools*, 14:57–74, 01 2009.

[Tab24] Amar Tabakovic. 3d terrain with level of detail, 2024. Report for semester project "Project 2".

Appendix A

Project Management

A.1 Time Chart

A.2 Requirements

A.2.1 Must-have Features

A.2.2 Should-have Features

A.2.3 Optional Features

Appendix B

Usage Guide

B.1 Height and Overlay Data Preprocessing

B.2 Terrain Streaming Server

B.2.1 Installation

B.2.2 Building

B.2.3 Running

B.3 StreamingATLOD

B.3.1 Installation

B.3.2 Building

B.3.3 Running