Karlsruhe Institute of Technology

Paper Out of Core Processes

Proseminar Paper by

Jonathan Beller

At the Faculty for Computer Science
Institute for Visualization and Data Analysis,
Computer Graphics Chair

14. Mai 2018

Inhaltsverzeichnis

1. Abstract

With the ever increasing amount of data to be rendered for a single given scene, experimenting with new approaches for handling and structuring that data becomes more and more important. This paper covers the approach to enhance response times and quality using out-of-core techniques.

With the help of out-of-core techniques, it is possible to achieve real-time rendering performance on current-generation graphics cards by overcoming limits posed by the graphics cards' internal memory [CNLE09]. Thus, this topic is essential for everyone hoping to achieve such performance for scenes that cannot be efficiently rendered using conventional methods.

I will discuss different strategies used to load necessary data chunks into working memory taking into account its finiteness using intelligent data streaming. These include visibility culling and virtual texturing. Also, this paper will give an overview of the possibility to use volumetric texture units called "Voxels" instead of regular texels and the advantages and challenges that come with it. Lastly, the topic will be further explored using the examples [CNLE09] and [VW09].

2. Motivation

Over the course of the last years, the complexity of scenes to be rendered has risen further and further. Nowadays, performance and complexity demands cannot be satisfied by solely utilizing brute force methods and trying to fit all necessary data into the scarce graphics memory. There are a number of specific use cases where another approach is needed. For example, [CGG$^+$03] look for a way to interactively render very large landscapes in the context of virtual reality using commmodity graphics hardware. Further, [GM05] name the fields of 3D scanning, computer-aided design and numerical simulation.

In essence, all of these use cases require a technique able to expand available memory by making use of larger but slower storage. While doing this, the searched technique should also minimize response times which are bound to increase due to longer memory access times. A technique that finds a suitable compromise between memory extension and response times is called an Out-of-Core technique.

3. Basics

This chapter aims to introduce key terminology that will be used in this paper. The chapter gives a basic understanding of the terms “Texture Mapping”, “Memory Virtualization”, “Virtual Texture” and “Level of Detail”.

3.1 Texture Mapping

Texture mapping provides a way of adding detail to rendered surfaces without increasing geometric complexity. In its simplest form, it takes a 2-dimensional image called *texture map* and a 3-dimensional model and paints the image onto the model. *Texture coordinates* are used to determine which point on the texture map corresponds to a point on the model.

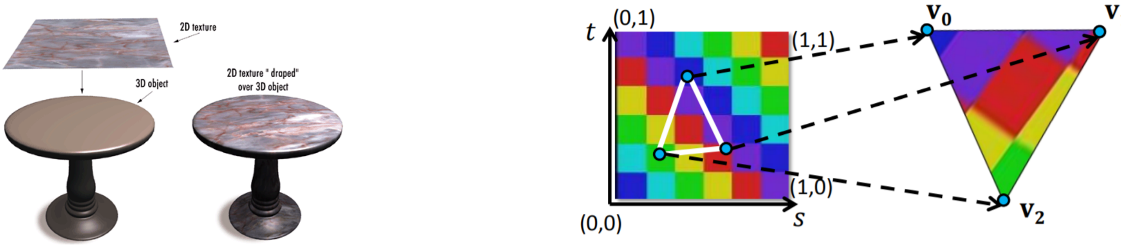

Abbildung 3.1: A texture is mapped onto the table model, making it more realistic. Images from [Tex] and [Dac17]

The first notable paper about texture mapping was published by [Cat74] in 1974. After that, texture mapping became particularly popular in video games of the 90’s and is now a crucial part of the vast majority of computer-rendered scenes. Because of its usefulness, the concept of texture mapping was extended from affecting the diffuseness to also altering other aspects like specularity, lighting and normals.

3.2 Memory Virtualization

Memory virtualization is the process of abstracting from the physical memory available. The initial motivation for this concept was to enable the uniform usage of heterogeneous storage devices [Eul08]. However, it also has more advantages. Firstly, memory fragmentation is reduced, decreasing average memory access times. Secondly, virtual memory can combine (graphics) main memory and slower storage and thus increase the total space available to load data into. Other aspects of memory virtualization are beyond the scope of this paper. The most important aspect of memory virtualization regarding rendering large amounts of data is its ability to extend available memory as this solves one of the issues mentioned in 2.

The Windows Display Driver Model v2.0 supports graphics memory virtualization under Windows 10 [Mic17]. However, page faults cause rendering to halt until the requested page is loaded. This makes this solution unsuitable on its own as it reduces responsiveness.

3.3 Virtual Texture

A virtual texture is, in general, a texture too large to fit into the physical graphics memory that is available in virtual memory.

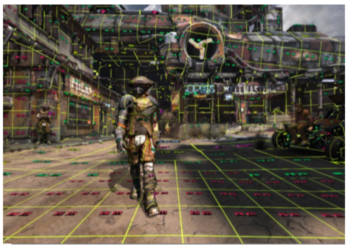

Abbildung 3.2: Image only shows a tiny portion of the actual virtual texture. Different levels of detail can be seen. Image from [VW09]

The advantage of a virtual texture over a conventional physical texture is its ability to hold significantly larger amounts of rendering data. For example, id Tech 5 uses virtual textures that consist of approximately 16.3 billion texels [VW09]. Without special handling, virtual textures face the same responsiveness issues as virtual memory.

3.4 Level of Detail

Level of Detail refers to altering an object's complexity between rendered frames. An example of this technique applied to textures is mip-mapping.

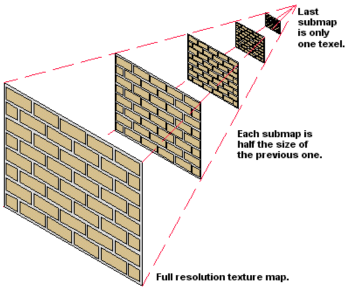
Last submap is only one texel.
Each submap is half the size of the previous one.
Full resolution texture map.

Abbildung 3.3: A texture is saved at different resolutions. As the viewer moves closer, resolution is increased and vice versa. Image from [PCM18]

Level of Detail addresses the issue of increasing scene complexity as the viewpoint moves away from a limited clip and its environment moves into view. Decreasing the complexity of the objects furthest away from the camera increases rendering performance while not significantly affecting quality [Uni17]. Hence, Level of Detail can be used to address the aspect of responsiveness mentioned in 2.

4. Main Content

4.1 Memory Management

Here I will talk about ways to utilize slow external memory to be able to store large datasets without losing wanted performance. I will refer to voxels, virtual paging and other techniques.

4.2 Voxels

Here I will give a detailed overview of the concept of voxels. I will talk about their differences to texels, advantages, challenges and their suitability for out of core processes.

4.3 Example 1: GigaVoxels

Here I will use the discussed concepts to explain how GigaVoxels can be used to render large volumetric datasets with high performance. I will explain how this is related to both voxels and out of core processes.

4.4 Example 2: ID Tech 5 Challenges

Here I will discuss in detail how out of core rendering is compatible with parallelization using the ID Tech 5 Challenges example. I will talk about the faced challenges and how they were solved.

4.5 Example 3: Visualization of very large Landscapes

Here I will summarize the findings from [CGG$^+$03]. I will refer to multi-resolution algorithms and how they are related to the topic.

5. Epilogue

Here I will outline how out of core techniques can be used to develop new ways of data provision and rendering scenes in the future.

Literaturverzeichnis

[Cat74] Edwin Catmull: *A subdivision algorithm for computer display of curved surfaces.* Technischer Bericht, UTAH UNIV SALT LAKE CITY SCHOOL OF COMPUTING, 1974.

[CGG$^+$03] Paolo Cignoni, Fabio Ganovelli, Enrico Gobbetti, Fabio Marton, Federico Ponchio und Roberto Scopigno: *Interactive Out-of-Core Visualisation of Very Large Landscapes on Commodity Graphics Platform.* In: Olivier Balet, Gérard Subsol und Patrice Torguet (Herausgeber): *Virtual Storytelling. Using Virtual RealityTechnologies for Storytelling*, Seiten 21–29, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg, ISBN 978-3-540-40014-1.

[CNLE09] Cyril Crassin, Fabrice Neyret, Sylvain Lefebvre und Elmar Eisemann: *GigaVoxels: Ray-guided Streaming for Efficient and Detailed Voxel Rendering.* In: *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, I3D '09, Seiten 15–22, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-429-4. `http://doi.acm.org/10.1145/1507149.1507152`.

[Dac17] Prof. Dr. Ing. Carsten Dachsbacher: *Kapitel 4: Texture Mapping.* 2017.

[Eul08] *Some History about Virtual Memory.* `http://euler.vcsu.edu:7000/14315/`, 2008. Accessed: 2018-05-14.

[GM05] Enrico Gobbetti und Fabio Marton: *Far Voxels: A Multiresolution Framework for Interactive Rendering of Huge Complex 3D Models on Commodity Graphics Platforms.* ACM Trans. Graph., 24(3):878–885, Juli 2005, ISSN 0730-0301. `http://doi.acm.org/10.1145/1073204.1073277`.

[Mic17] *GPU virtual memory in WDDM 2.0.* `https://docs.microsoft.com/en-us/windows-hardware/drivers/display/gpu-virtual-memory-in-wddm-2-0`, 2017. Accessed: 2018-05-14.

[PCM18] *Level of Detail (LOD).* `https://www.pcmag.com/encyclopedia/term/47078/mip-mapping`, 2018. Accessed: 2018-05-14.

[Tex] *texture map - Computer Definition.* `http://www.yourdictionary.com/texture-map`. Accessed: 2018-05-14.

[Uni17] *Level of Detail (LOD).* `https://docs.unity3d.com/Manual/LevelOfDetail.html`, 2017. Accessed: 2018-05-14.

[VW09] JMP Van Waveren: *id tech 5 challenges-from texture virtualization to massive parallelization.* Talk in Beyond Programmable Shading course, SIGGRAPH, 9:5, 2009.

Erklärung

Ich versichere, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt und von dieser als Teil einer Prüfungsleistung angenommen.

Karlsruhe, den 14. Mai 2018

(Jonathan Beller)