DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Thesis type (Bachelor's Thesis in Informatics, Master's Thesis in Informatics: Games Engineering, ...)

Thesis title

Author

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Thesis type (Bachelor's Thesis in Informatics, Master's Thesis in Informatics: Games Engineering, ...)

Thesis title

Titel der Abschlussarbeit

Author:	Author
Supervisor:	Supervisor
Advisor:	Advisor
Submission Date:	Submission date

I confirm that this thesis type (bachelor's thesis in informatics, master's thesis in informatics: games engineering, ...) is my own work and I have documented all sources and material used.

Munich, Submission date Author

Acknowledgments

Abstract

Kurzfassung

Contents

1. Introduction

Use with pdfLaTeX and Biber.

1.1. Section

1. Experiments in GPU-based occlusion culling (Anagnostou, Interplay of Light) [1]:
tldr CPU occlusion culling is very coarse as the CPU isn't great at rasterising. GPU can rasterise efficiently and supports hardware occlusion queries. Problem: "drawcall level" granularity, needs query and fence around every drawcall, thus can't handle instancing very well. Other problem: readback to CPU necessary, to avoid stalling use previous frame occlusion info for current frame. Thus popping likely for fast moving objects or camera. Other solution: render low res occlusion buffer on GPU, produce mip chain, determine screen-space size of each prop and compare vs appropriate mip-level of buffer. Still needs CPU readback though, unless... compute shader in modern API: "In this case I will be using a compute shader to perform the occlusion tests, producing a list of visible props that I will then be consuming on the GPU, avoiding the CPU roundtrip."

2. Why Frustum Culling Matters, and Why It's Not Important [2]:
tldr disregardm, oversimplified explanation of frustum culling.

3. Overview on popular occlusion culling techniques [3]:
tldr temporal culling methods are prone to artifacting with fast object or camera movement, CPU culling "is the most efficient, forward-looking approach", then again the article is written by someone from Umbra3D, a CPU occlusion culling middleware. Still, precomputed visibility sets for weak devices are brought up, low res HI-Z depth buffer rasterised on the CPU with SIMD units for dynamic culling, see Intel's 2015/2016 occlusion culling article.

4. Frustum Culling [4]:
tldr code samples and brief performance numbers for bounding sphere, AABB, OBB based culling, SSE support, MT support, somewhat simple GPU culling. Sphere or AABB with SSE + MT support quite fast, 0.1-0.2ms for 100k objects on unspecified i5 quadcore.

5. Dual-Cone View Culling for Virtual Reality Applications [5]:
tldr not discernibly faster than frustum pyramid culling plus stencil mesh. Needs careful tuning for each HMD to avoid performing worse when a larger image is rendered and warped to view. With SIMD, more accurate cones etc may be more efficient.

6. Culling Techniques [6]:
tldr general explanation. Of interest maybe hierarchical bounding volumes for frustum culling. As the goal is a fast rendering of extreme number of objects, hierarchical ordering seems obviously necessary.

7. A Survey of Visibility for Walkthrough Applications [7]:
tldr .

8. Occlusion Culling Methods [8]:
tldr .

9. Math for Game Developers - Frustum Culling[9]:
tldr as expected, brief mathematical explanation on how to compute whether a prop is inside the view frustum (based on position and prop radius).

10. Wie Sie einen Einstieg in das Frustum Culling finden [10]:
tldr D3D sample code for view frustum culling. Very brief, not very useful. But raises the questions how to compute (and store) bounding sphere radius for each prop, and how to compute and update frustum planes each frame (aka reconstruct from view + proj matrices or set up at start and then move all planes according to camera movement).

11. Superfrustum culling [11]:
tldr instead of one frustum per eye, construct one superfrustum covering both. On paper means somewhat fewer objects are culled, but the reduced overhead still comes out on top. In their testing, shaved off 1ms on lower end CPUs. Also monoscopic far-field for lower end devices and far draw distances: past a certain distance, stereo separation becomes very small, nigh indistinguishable, so it may be faster to render the far field monoscopic and then composit with the near-field stereo image.

12. OpenGL sample for shader-based occlusion culling [12]:
tldr shader-based batched occlusion culling system: leverages multi_draw_indirect and works well with setups where all geo is stored in one big buffer.

Thus, things up in the air: frustum culling, occlusion culling, detail culling, distance culling, hierarchical culling, cpu culling vs gpu culling, dual-view frustum culling vs superfrustum culling, stencil meshing, ...center-distance culling for foveated rendering, SIMD and MT optimization, ...

1.1.1. Subsection

See Table 1.1, Figure 1.1, Figure 1.2, Figure 1.3, Figure 1.4, Figure 1.5.

This is how the glossary will be used.

Table 1.1.: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

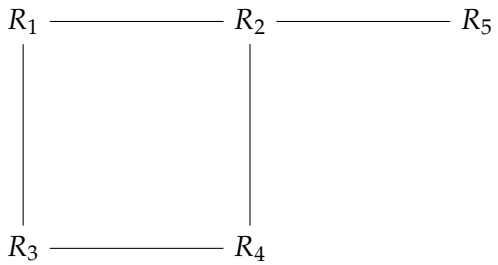

Figure 1.1.: An example for a simple drawing.

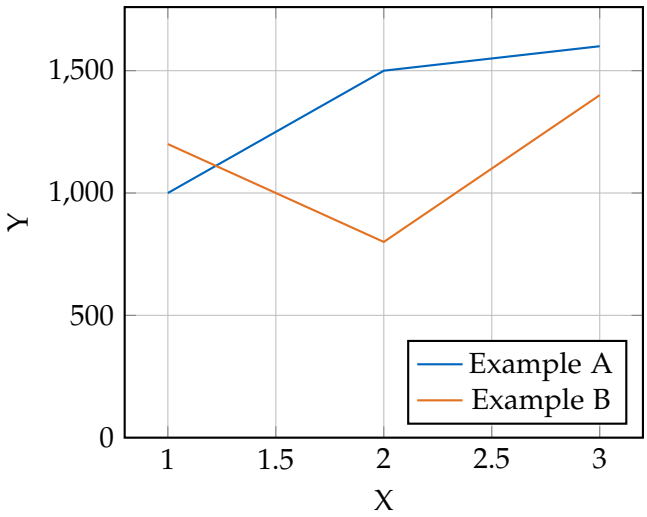

Figure 1.2.: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 1.3.: An example for a source code listing.

Figure 1.4.: Includegraphics searches for the filename without extension first in logos, then in figures.

Figure 1.5.: For pictures with the same name, the direct folder needs to be chosen.

(a) The logo.

(b) The famous slide.

Figure 1.6.: Two TUM pictures side by side.

Donor dye, ex. Alexa 488 (D_{dye}), Förster distance, Förster distance (R_0), and k_{DEAC}. Also, the TUM has many computers, not only one Computer. Subsequent acronym usage will only print the short version of Technical University of Munich (TUM) (take care of plural, if needed!), like here with TUM, too. It can also be –> hidden[1] <–.

[(TODO: Now it is your turn to write your thesis.
This will be a few tough weeks.)]
[(Done: Nevertheless, celebrate it when it is done!)]

[1]Example for a hidden TUM glossary entry.

A. General Addenda

If there are several additions you want to add, but they do not fit into the thesis itself, they belong here.

A.1. Detailed Addition

Even sections are possible, but usually only used for several elements in, e.g. tables, images, etc.

B. Figures

B.1. Example 1

✓

B.2. Example 2

✗

List of Figures

List of Tables

Bibliography

[1] K. Anagnostou. *EXPERIMENTS IN GPU-BASED OCCLUSION CULLING*. 2017. URL: https://interplayoflight.wordpress.com/2017/11/15/experiments-in-gpu-based-occlusion-culling/.

[2] S. Barrett. *Why Frustum Culling Matters, and Why It's Not Important*. 2017. URL: https://gist.github.com/nothings/913056601b56e5719cc987684a16544e.

[3] *Overview on popular occlusion culling techniques: The experts at Umbra share their knowledge on solving the 3D visibility problem*. 2016. URL: https://www.gamesindustry.biz/articles/2016-12-07-overview-on-popular-occlusion-culling-techniques.

[4] A. Gerlits. *Frustum Culling*. 2017. URL: https://www.gamedev.net/articles/programming/general-and-gameplay-programming/frustum-culling-r4613/.

[5] J. Hale. *Dual-Cone View Culling for Virtual Reality Applications*. 2018. URL: https://blog.squareys.de/dual-cone-view-culling-for-vr/.

[6] Dr. K.R. Subramanian. *Culling Techniques*. Charlotte, NC. URL: https://webpages.uncc.edu/krs/courses/5010/ged/lectures/cull_lod2.pdf.

[7] D. Cohen-Or, Y. L. Chrysanthou, C. T. Silva, and F. Durand. "A Survey of Visibility for Walkthrough Applications". In: *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS* 9.3 (2003), pp. 412–431. URL: https://www.researchgate.net/publication/2440562_A_Survey_of_Visibility_for_Walkthrough_Applications.

[8] H. Hey and W. Purgathofer. "Occlusion Culling Methods". In: *EUROGRAPHICS STAR* (2001). URL: https://pdfs.semanticscholar.org/c1a0/aa9000f62bb9b1a60f27182e23872c375e1e.pdf.

[9] J. Rodriguez. *Math for Game Developers - Frustum Culling*. 2013. URL: https://www.youtube.com/watch?v=4p-E_31XOPM.

[10] *Wie Sie einen Einstieg in das Frustum Culling finden*. URL: https://viscircle.de/wie-sie-einen-einstieg-in-das-frustum-culling-finden/.

[11] N. Whiting. *Oculus Connect 4 | The Road to Shipping: Technical Postmortem for Robo Recall: Superfrustum culling*. 2017. URL: https://www.youtube.com/watch?v=BZhOUGG45_o&feature=youtu.be&t=46m12s.

[12] C. Kubisch. *OpenGL sample for shader-based occlusion culling*. 2014. URL: https://github.com/nvpro-samples/gl_occlusion_culling.