

# **Morphing mit Beier-Neely**

## **Digitale Bildverarbeitung WS2023/24**

Michael Eggers, Johann Rittenschober, Kevin

26. Dezember 2023



## Erklärung

Hiermit erklären wir, dass die vorliegenden Arbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet wurden.

Michael Eggers, Johann Rittenschober, Kevin

München, 26. Dezember 2023

Matrikelnummer: 00322614

Studiengruppe: Master Informatik VZ/TZ

## **Zusammenfassung**

schreiben

## **Abstract**

write

# Inhaltsverzeichnis

<b>1. Grundlagen</b>	<b>7</b>
1.1. Das Vulkan API . . . . .	7
1.1.1. Koordinatensysteme . . . . .	7
1.1.2. Heaps und Memory Types . . . . .	7
1.1.3. Descriptor Sets . . . . .	7
1.1.4. Grafik-Pipeline . . . . .	7
1.2. Ray Tracing . . . . .	7
1.2.1. Whitted Ray Tracing . . . . .	7
1.2.2. Distributed (Stochastic) Ray Tracing . . . . .	7
1.2.3. Kajiya-Style Ray Tracing (Path Tracing) . . . . .	7
<b>2. Fazit und Ausblick</b>	<b>8</b>
<b>A. Erstellen einer Visual Studio Solution des Vulkan Rahmenprogramms</b>	<b>9</b>
A.1. Download und Installation des Vulkan SDKs . . . . .	9
A.2. Download und Verwendung von CMake . . . . .	9
A.2.1. Aktualisieren des Vulkan SDK . . . . .	9
A.3. GeForce-Treiber für RTX Extensions . . . . .	9
<b>Literaturverzeichnis</b>	<b>10</b>

# Abbildungsverzeichnis

# **Tabellenverzeichnis**

# **1. Grundlagen**

## **1.1. Das Vulkan API**

### **1.1.1. Koordinatensysteme**

### **1.1.2. Heaps und Memory Types**

#### **Staging Buffer**

### **1.1.3. Descriptor Sets**

### **1.1.4. Grafik-Pipeline**

## **1.2. Ray Tracing**

### **1.2.1. Whitted Ray Tracing**

### **1.2.2. Distributed (Stochastic) Ray Tracing**

### **1.2.3. Kajiya-Style Ray Tracing (Path Tracing)**

## **2. Fazit und Ausblick**



## **A. Erstellen einer Visual Studio Solution des Vulkan Rahmenprogramms**

**A.1. Download und Installation des Vulkan SDKs**

**A.2. Download und Verwendung von CMake**

**A.2.1. Aktualisieren des Vulkan SDK**

**A.3. GeForce-Treiber für RTX Extensions**

# Literaturverzeichnis

- [1] Tomas Akenine-Möller, Eric Haines, Naty Hoffman, Angelo Pesce, Michał Iwanicki, and Sébastien Hillaire. *Real-Time Rendering 4th Edition*. A K Peters/CRC Press, Boca Raton, FL, USA, 2018.
- [2] Nischwitz Et al. *Computergrafik, 4. Auflage*. Springer Vieweg, 2019.
- [3] Alexey Panteleev Christoph Schied. *Real-Time Path Tracing and Denoising in 'Quake 2' (Presented by NVIDIA)*, 2019. <https://www.gdcvault.com/play/1026185/Real-Time-Path-Tracing-and>.
- [4] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *SIGGRAPH Comput. Graph.*, 18(3):137–145, January 1984.
- [5] Koch Daniel. *Ray Tracing In Vulkan*, 2020. <https://www.khronos.org/blog/ray-tracing-in-vulkan>.
- [6] Mark. Deloura. *Game Programming Gems*. Charles River Media, Inc., 2000.
- [7] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2002.
- [8] Zandro Fagnoli. *Synchronizing the CPU and GPU*, 2020. [https://github.com/KhronosGroup/Vulkan-Samples/blob/master/samples/performance/wait\\_idle/wait\\_idle\\_tutorial.md](https://github.com/KhronosGroup/Vulkan-Samples/blob/master/samples/performance/wait_idle/wait_idle_tutorial.md).
- [9] Randima Fernando. *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education, 2004.
- [10] Randy Fernando. Percentage closer soft shadow. *ACM SIGGRAPH 2005 Sketches*, 01 2005.
- [11] Khronos Group. *OpenGL® and OpenGL® ES Reference Pages*, 2014. <https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/glClipControl.xhtml>.
- [12] Khronos Group. *Vulkan® 1.2.141 - A Specification*, 2020. <https://www.khronos.org/registry/vulkan/specs/1.2-khr-extensions/html/chap23.html#vertexpostproc-clipping>.
- [13] Eric Haines and Tomas Akenine-Möller, editors. *Ray Tracing Gems*. Apress, 2019. <http://raytracinggems.com>.
- [14] Robert Hallock. *One of Mantle's Futures: Vulkan*, 2015. <https://community.amd.com/community/gaming/blog/2015/05/12/one-of-mantles-futures-vulkan>.
- [15] Khronos Inc. *Depth Buffer Precision*, 2012. [https://www.khronos.org/opengl/wiki/Depth\\_Buffer\\_Precision](https://www.khronos.org/opengl/wiki/Depth_Buffer_Precision).
- [16] Khronos Inc. *GLSL Nvidia Raytracing Extension*, 2019. [https://github.com/KhronosGroup/GLSL/blob/master/extensions/nv/GLSL\\_NV\\_ray\\_tracing.txt](https://github.com/KhronosGroup/GLSL/blob/master/extensions/nv/GLSL_NV_ray_tracing.txt).

- [17] Khronos Inc. *The SPIR-V Open Source Ecosystem*, 2020. <https://www.khronos.org/spir/>.
- [18] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986.
- [19] Dustin Land. *First Light - Bringing DOOM to Stadia*, 2019. <https://www.youtube.com/watch?v=qdz4b5psrhE&t=2322>.
- [20] Eric Lengyel. *Foundations of Game Engine Development, Volume 2, Rendering*. Terathon Software LLC, 2019.
- [21] Juan Liniestky. *Porting Godot to Vulkan*, 2020. <https://youtu.be/sh7Sr4-RIfM>.
- [22] Juan Liniestky. *Vulkan progress report*, 2020. <https://godotengine.org/article/vulkan-progress-report-7>.
- [23] MultiMedia LLC. *NVIDIA Vulkan Ray Tracing Tutorial*, 2020. [https://nvpro-samples.github.io/vk\\_raytracing\\_tutorial/](https://nvpro-samples.github.io/vk_raytracing_tutorial/).
- [24] Peter Meyberg, Kurt und Vachenauer. *Höhere Mathematik 1*. Springer-Verlag Berlin Heidelberg, 6. edition, 2001.
- [25] Christoph Kubisch (Nvidia). *OpenGL like Vulkan*, 2016. <https://developer.nvidia.com/opengl-vulkan>.
- [26] Christoph Peters. *Moments in Graphics - Blue Noise Textures*, 2016. <http://momentsingraphics.de/BlueNoise.html>.
- [27] Matt Pharr and Randima Fernando. *Gpu Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Addison-Wesley Professional, first edition, 2005.
- [28] William T. Reeves, David Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. In *SIGGRAPH '87*, 1987.
- [29] Peter Rubin. *What to Expect From Sony's Next-Gen PlayStation*, 2019. <https://www.wired.com/story/exclusive-sony-next-gen-console/>.
- [30] Christoph Schied. *Q2VKPT - Quake 2 Real-Time Path Tracing using RTX*, 2019. <http://brechpunkt.de/q2vkpt/>.
- [31] Christoph Schied, Christoph Peters, and Carsten Dachsbacher. Gradient estimation for real-time adaptive temporal filtering. *Proc. ACM Comput. Graph. Interact. Tech.*, 1(2), August 2018.
- [32] G. Sellers and J.M. Kessenich. *Vulkan Programming Guide: The Official Guide to Learning Vulkan*. Always learning. Addison-Wesley, 2016.
- [33] Peter Shirley. Ray tracing in one weekend, July 2020.
- [34] Peter Shirley. Ray tracing: The rest of your life, July 2020.
- [35] Martin Stich. *Real-Time Raytracing with Nvidia RTX*, 2018. <http://on-demand.gputechconf.com/gtc-eu/2018/pdf/e8527-real-time-ray-tracing-with-nvidia-rtx.pdf>.
- [36] R. A. Ulichney. Dithering with blue noise. *Proceedings of the IEEE*, 76(1):56–79, 1988.

- [37] Will Usher. *The RTX Shader Binding Table Three Ways*, 2019. <https://www.willusher.io/graphics/2019/11/20/the-sbt-three-ways>.
- [38] James M. Van Verth and Lars M. Bishop. *Essential Mathematics for Game and Interactive Applications, Third Edition*. AK Peters/CRC Press, 2015.
- [39] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, June 1980.
- [40] Alan Wolfe. *Animating Noise For Integration Over Time*, 2017. <https://blog.demofox.org/2017/10/31/animating-noise-for-integration-over-time/>.