

OpenGL-basiertes Software Occlusion Culling zur Beschleunigung des 3D-Renderings großer Datenmengen und komplexer Szenen

Projekt-INF

Dominik Sellenthin

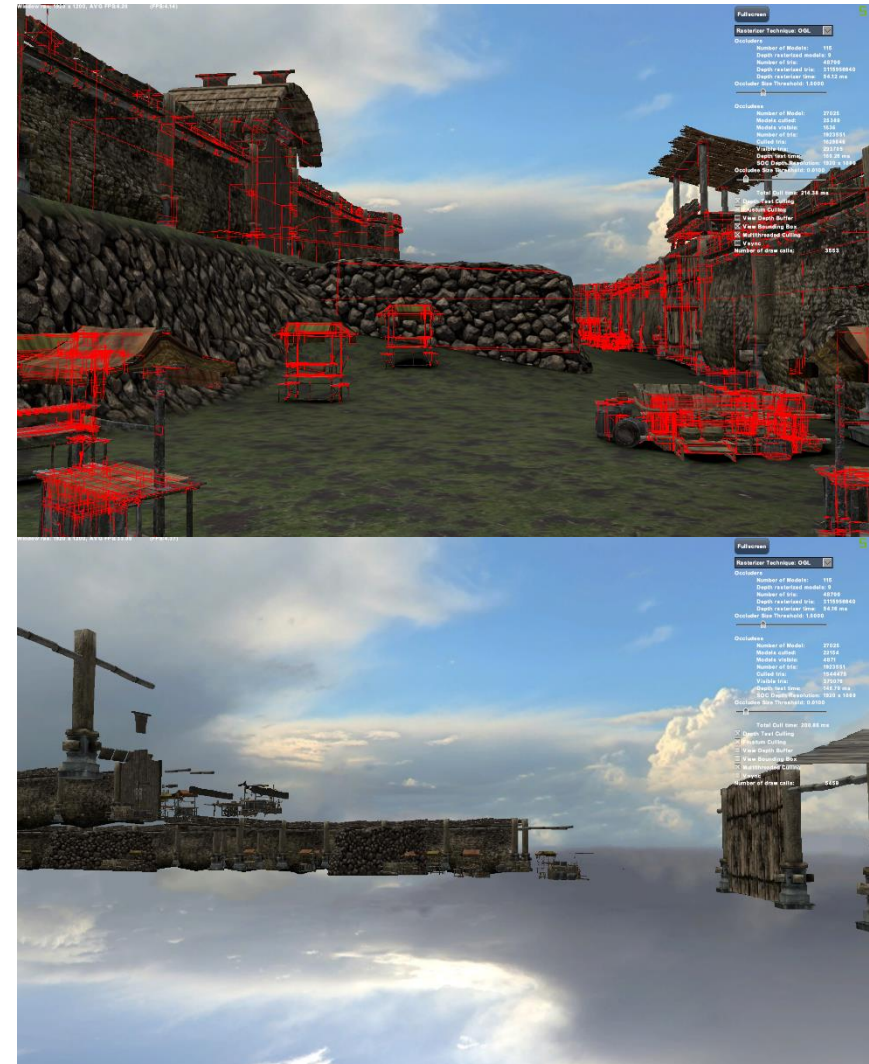
Christian Stegmaier

Betreuer: Dr. Guido Reina

Michael Becher, M. Sc.

Motivation

- Mehr dynamische Szenen
- PSM schwierig zu verwenden
- → Occlusion Culling
- Finde Objekte, die nicht gerendert werden sollen
- Ungenutzte CPU

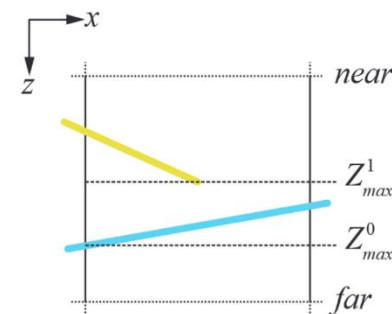
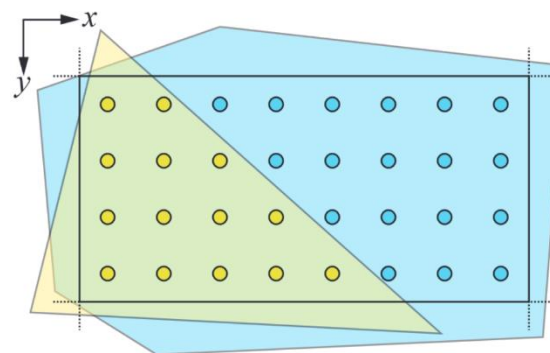


Related Work

- Intels Software Occlusion Culling Framework
- Scalar
- SSE - Streaming SIMD Extension
- AVX – Advanced Vector Extensions
- Masked AVX
- (Alle sowohl Single als auch Multithreaded)

Masked AVX Occlusion Culling

- Arbeitet mit abgewandeltem hierarchischem Z-Buffer (HiZ)
- Arbeitsebene und Referenzebene
- 8 SIMD-lanes formen 32x8 Kachel
- SIMD-lane als 8x4 Kachel



Masked Software Occlusion Culling [1]

Occlusion Culling

- Voraussetzung: Unterteilung Occluder/Occludee
- Ablauf:
 - Occluder:
 - Frustum Culling
 - Tiefenpuffer-Rasterisierung
 - Occludee:
 - Frustum Culling
 - Occlusion Queries
 - Ergebnisweiterleitung an Renderroutine

Occlusion Culling

Occluder, große Objekte

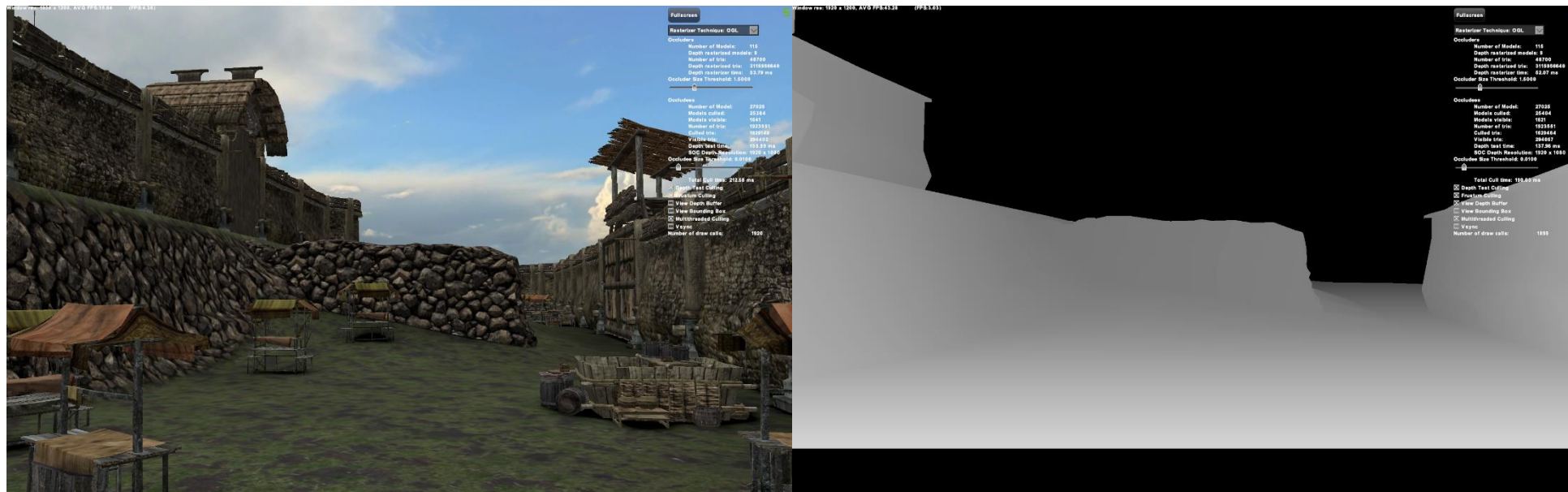


Occludee, kleine Objekte



Tiefenpuffer-Rasterisierung

- Renderingdurchlauf mit Occludermenge
- Tiefenpuffer wird beschrieben
- Farbpuffer wird nicht beschrieben



Occlusion Queries

- Pro Occludee eine Occlusion Query (glQuery)
 - Ablauf:
 - Starte alle Queries (glBeginQuery, glDrawArrays, glEndQuery)
 - Warte auf Ergebnis (GL_QUERY_RESULT_AVAILABLE)
 - Wartezeiten!
 - Hole Ergebnisse aller Queries
 - Doppelte Anzahl Queries
 - Einmal für aktuellen Frame
 - Einmal für vorherigen Frame
- keine Warteroutine → keine Wartezeiten

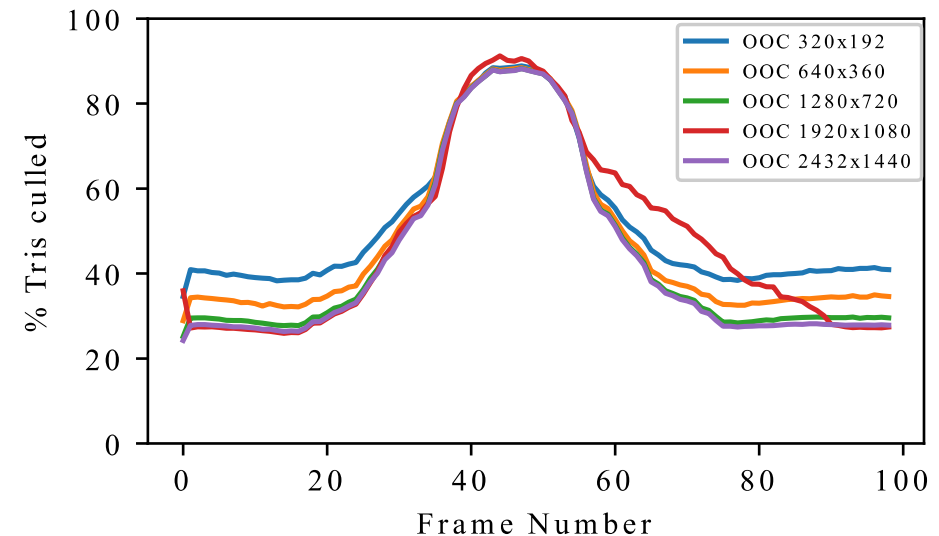
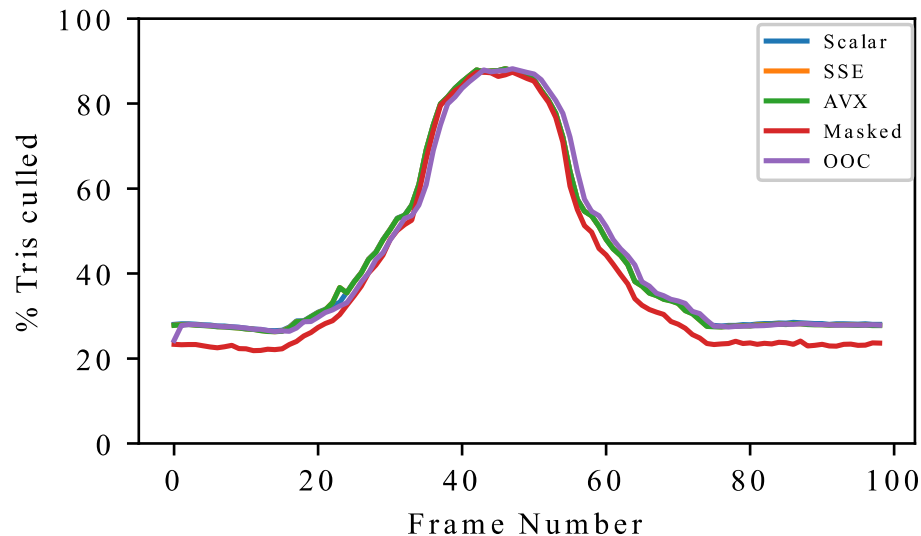
Ergebnisse: SOC-Framework

- Zur Messung aller Ergebnisse dient das Intel SOC-Framework
- Testszene:
 - 115 Occluder => ~48.700 Dreiecke
 - 27.025 Occludees => ~1.923.000 Dreiecke
- Tiefenpuffer Auflösung: 1920x1080
- Kamerafahrt über 100 Frames

Ergebnisse: Kamerafahrt

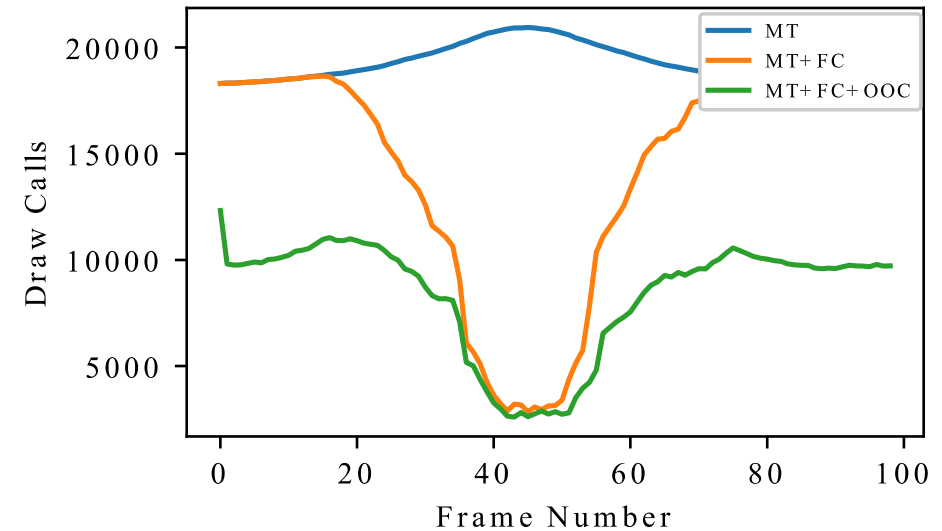
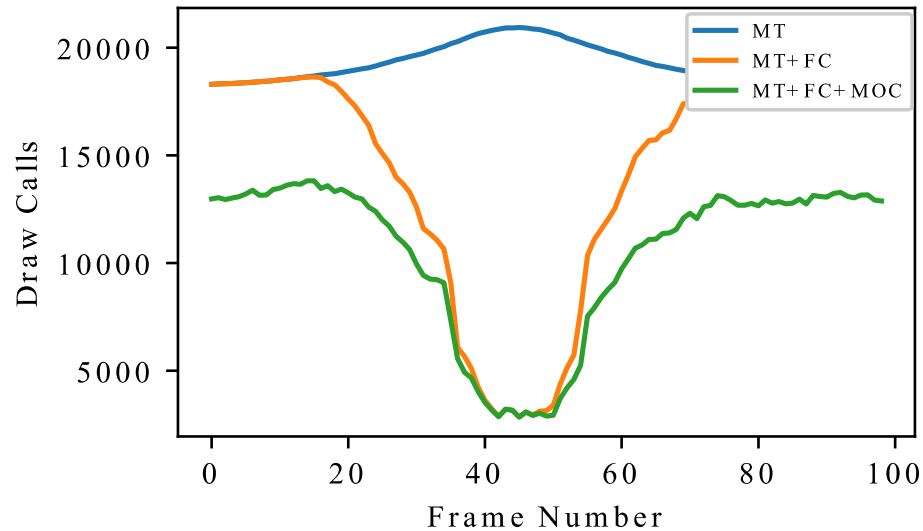


Ergebnisse: Qualität + unterschiedliche TB Auflösung



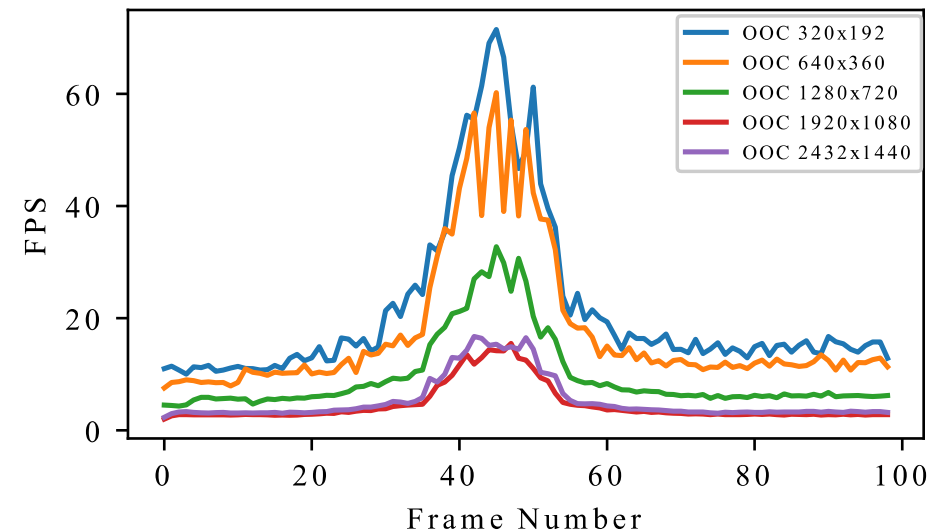
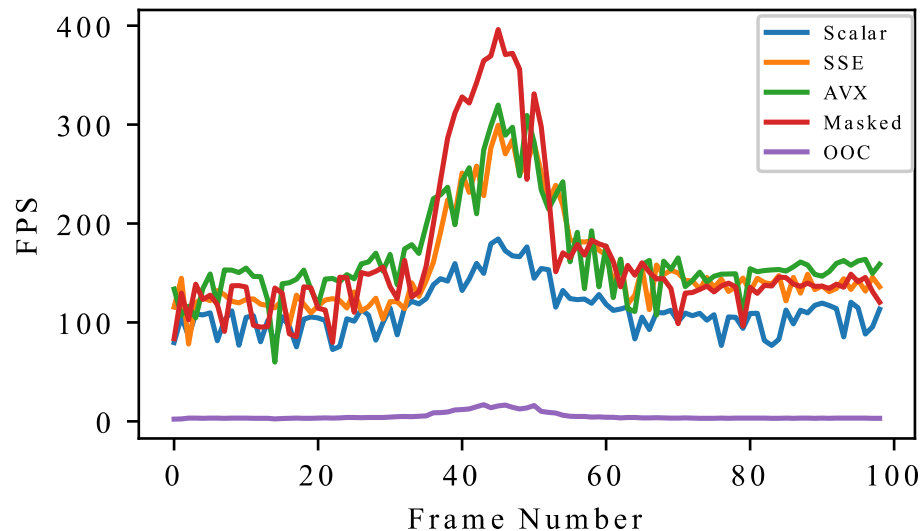
	OpenGL Occlusion Culling				
	320x192	640x360	1280x720	1920x1080	2432x1440
Tris culled (%)	51.75	47.59	44.65	48.18	43.77
Draw Calls	3551	5696	7644	7896	8423
Models Culled	23819.61	21970.67	20320.78	20135.84	19660.16
Models Visible	3205.39	5054.33	6704.22	6889.16	7364.84

Ergebnisse: Qualität MOC vs. OOC



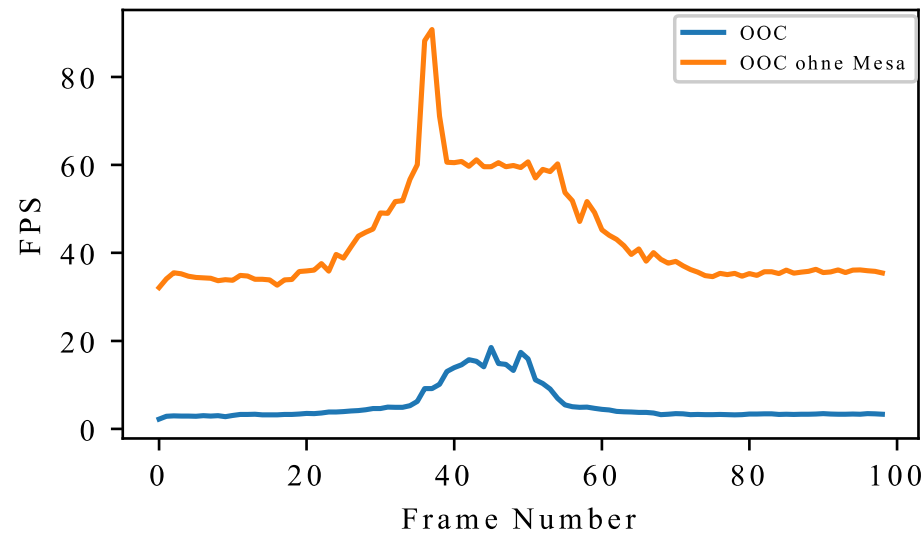
			MOC	OOC
	MT	MT+Frustum	MT + Frustum + OC	
Tris culled (%)	11.13	11.13	43.77	40.41
Draw Calls	19219	14318	8423	10528
Models Culled	10180.34	14532.39	19660.16	17903.78
Models Visible	16844.66	12492.61	7364.84	9121.22

Ergebnisse: Performance + unterschiedliche TB Auflösung



	OpenGL Occlusion Culling				
	320x192	640x360	1280x720	1920x1080	2432x1440
FPS	21.48	17.53	9.55	4.58	5.31
DepthTest time (ms)	50.4	63.22	122.21	265.67	228.85
Rasterize time (ms)	6.64	6.85	6.87	7.07	8.02

Ergebnisse: Performance CPU vs. GPU



Fazit

- Mesa 3D ermöglicht eine zur GPU parallele OC Methode auf der CPU
- Qualität von OOC ist gut
- Performance ist sehr schlecht
 - Hauptgrund: Occlusion Queries

Future Work: Occludee Pakete

- Bisher: Je Objekt eine Occlusion Query
 - schlechte Performance
- Ansatz: Eine Occlusion Query für mehrere Objekte
 - Anzahl der Queries wird verringert

1) Occludees zusammenfassen

2) Ist ein Occludee sichtbar, wird das ganze Paket gerendert

- Für gute Ergebnisse müssen die Pakete sinnvoll gewählt werden
- Sinnvoll = nah beieinander

Literatur

- [1] J. Bittner, M. Wimmer, H. Piringer, and W. Purgathofer. Coherent hierarchical culling: Hardware occlusion queries made useful. In Computer Graphics Forum, volume 23, pages 615–624. Wiley Online Library, 2004.
- [2] C. Chandrasekaran, D. McNabb, K. Kiefer, M. Fauconneau, and F. Giesen. Software occlusion culling. <https://software.intel.com/en-us/articles/software-occlusion-culling>, 2013-2016.
- [3] D. Cohen-Or, Y. L. Chrysanthou, C. T. Silva, and F. Durand. A survey of visibility for walkthrough applications. IEEE Transactions on Visualization and Computer Graphics, 9(3):412–431, 2003.
- [4] N. Greene, M. Kass, and G. Miller. Hierarchical z-buffer visibility. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pages 231–238. ACM, 1993.
- [5] J. Hasselgren, M. Andersson, and T. Akenine-Möller. Masked Software Occlusion Culling. In U. Assarsson and W. Hunt, editors, Eurographics/ACM SIGGRAPH Symposium on High Performance Graphics. The Eurographics Association, 2016.

TODO

- Doppeltes set an queries so gut wie keine performanceverbesserung
- Mesa3d erwähnen