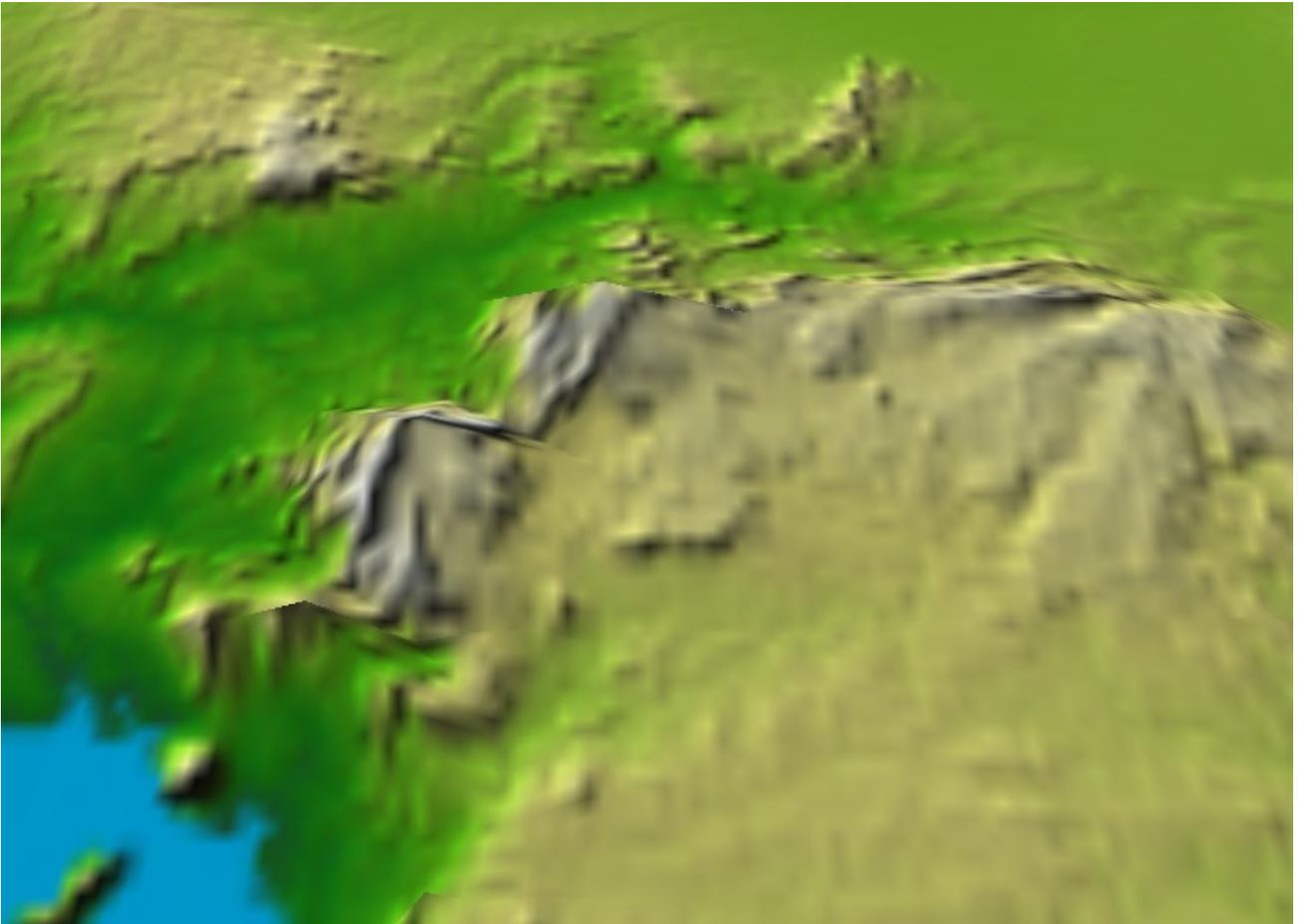


Hippie



VU Echtzeitgraphik WS07/08

Name	Kennzahl	Matrikelnummer	Email
Marc Dietrichstein	532	0327606	marc.dietrichstein@gmail.com
Phillipp Kogelnig	932	0225185	kogelnig_philipp@gmx.at

Arbeitstitel Hippie

Demo

Eine Heightmap wird aus Nasa strm30 Daten geladen. Die Daten werden über Clipmaps mit einem Shader in Abhängigkeit zur Kameraposition gerendert. Bewegt sich die Kamera werden die Daten in den jeweiligen Levels upgedatet und die Shader Parameter werden neu eingestellt. Um ein schnelles Rendern der einzelnen Lod zu ermöglichen werden die Daten in static VBO Patches mit Instance werten gespeichert. Dadurch sind für jedes LOD nur 4 Drawcalls für das feinste Level nur 3 Drawcalls notwendig.

Effekte

- Heightmap ([1] Kapitel 2)
Visualisierung der Heightmap über einen Vertexshader.
- Clipmaps ([1] Kapitel 2)
Update der Texturedaten in Abhängigkeit der Position.
- Instancing ([1] Kapitel 3)
Mehrere Vertexpatches werden nebeneinander mit Instancewerten in die Graphikkarte übertragen um später keine Vertexdaten mehr übertragen zu müssen.
- LOD ([1] Kapitel 2)
Statisches LOD wird über die vorgegebene Gittergröße erreicht, in Abhängigkeit zur Entfernung der einzelnen Vertices wird ausserdem ein Alpha Factor berechnet der dazu dient dynamisches LOD zu implementieren. Vertexhöhe und Normalen werden interpoliert.
- Culling
Culling der einzelnen Patches gegen ein AABB des Viewfrustum um nur Daten zu zeichnen die im sichtbaren Bereich liegen, verändern der Instancearray Daten und Drawcall Länge. Dadurch müssen als einzige Daten Variablen Arrays zu der Grafikkarte geschickt werden.
- Sampeln
Upsampeln der Nasadaten durch Subdivision in einem Shader. Die Daten werden

Technik

Realisiert mit C++, OpenGL und Cg.

Kommentare Sourcecode

Main	/src/zort/apps/apps_terrain.cpp
clipmaps	/src/plugins/resources/clipmap.h /src/plugins/resources/clipmap.h
terrain	/src/plugins/visuals/terrain.h /src/plugins/visuals/terrain.cpp

Das Projekt ist Plattform kompatibel zu WindowsXp und Linux

Abhängigkeiten

Devil, Cg 1.5, Python

References

- [1] Matt Pharr. *GPU Gems 2*, Upper Saddle River, NJ, 2005. Addison-Wesley.
- [2] David H. Eberly. *3D Game Engine Design, Second Edition: A Practical Approach to Real-Time Computer Graphics*, San Francisco, CA, USA, 2006. Morgan Kaufmann Publishers Inc.