

Projekt 7: Displacement Mapping

Displacement Mapping ist eine Technik in der Computergrafik, um Texturen dreidimensional aussehen zu lassen, ohne ein dreidimensionales Mesh zu verwenden. Dazu wird eine Textur durch eine zusätzliche Textur ergänzt, welche Höhendaten enthält (meist in Grauwert). In den Shadern der Grafikpipeline wird dann aus einer einfachen Fläche, je nach Bedarf viele kleine Dreiecke gebildet, die anschließend an deren Ecken durch die Höhendaten verschoben werden. Es ist so nicht notwendig vorher ein dreidimensionales Mesh an die GPU zu übergeben, sondern dies kann zur Renderzeit automatisch generiert werden. Eine ähnliche Technik ist Bump Mapping.



Ziel

Das Ziel dieses Projekts ist die Umsetzung des Displacement Mappings auf der GPU. Der Nachteil der Generierung des Meshes in der Rendering-Pipeline ist, dass die Höhendaten nicht für die Applikation zur Verfügung stehen, z.B. für Kollisionen. Um dies zu vereinfachen, kann nun mit CUDA oder OpenCL aus einer bestehenden Textur eine Displacement Map erzeugt werden. Das Mesh kann danach an den Host zurückgegeben werden. Der Ansatz ist dabei, Bildmaterial zu verwenden, welches leicht in Displacement Maps umzuwandeln sind. Siehe als Beispiel die Abbildung oben. Dazu kann das Bild durch vorherige Bildverarbeitung in ein Grauwertbild mit Höhendaten umgewandelt werden (z.B. Gimp).

Die Grafikpipeline wird dann zur Ausgabe des so generierten Meshes verwendet. Als Abschluss kann die Originaltextur dann auf das Mesh aufgetragen werden. NVidia selbst hat ein Artikel für die Bildung einer Displacement Map veröffentlicht [1].

Geschätzter Aufwand

- CUDA / OpenCL: 70 %
- OpenGL: 30 %

Themen

- Aus Eingabebild eine Displacement Map erstellen.
- Grauwertbild als Displacement anwenden
- Granularität des Mesh per Parameter bestimmen
(z.B. über Tastendruck)
- Tiefe/Kontur der Displacement Map per Parameter bestimmen
(z.B. über Tastendruck)
- Triangulierung des Bildes als Mesh
- Rendering des Mesh mit OpenGL
- Optional: Parameter (Kontur und Granularität) live änderbar
- Optional: Texture auf Mesh rendern

Evaluation

- Verschiedene Farbstrukturen um das Mesh eindeutig zu rendern z.B. Heightmap oder Heatmap
- Verschiedene Beispielbilder verwenden z.B. Maschendrahtzaun, Steinmauer oder Schriftzüge
- Verschiedene Auflösungen der Bilder
- Verschiedene Granularitäten für das Mesh (Anzahl Dreiecke)
- Optional: Texture auf Mesh rendern

References

- [1] William Donnelly. Per-pixel displacement mapping with distance functions. *GPU gems*, 2(22):3, 2005.