

3D Sensordatenverarbeitung - Übung 5

Sommersemester 2021

Polygonalisierung mit Marching Cubes

Aufgabe 1: Erstellung eines Octree-Voxelgrids (25 Punkte)

Schreiben Sie eine Klasse `Octree`, die eine Punktwolke aus einer Datei einliest und daraus einen Octree mit vorgegebener maximaler Voxelgröße erstellt, d.h. die Voxel werden so lange weiter halbiert, bis die vorgegebene Voxelgröße unterschritten wird. Die Blätter des Baumes sollen für die Rekonstruktion mit Marching Cubes traversiert werden können.

Aufgabe 2: Implementierung der Distanzfunktion nach Hoppe (25 Punkte)

Schreiben Sie eine Klasse `DistanceFunction`, die eine Berechnung der Distanzfunktion nach Hoppe implementiert. Schätzen Sie dafür zunächst mit einer der Methoden vom letzten Aufgabenblatt die Normalen für die Datenpunkte. Die Anzahl der Nachbarn zur Normalenschätzung und Interpolation soll parametrierbar sein. Die Klasse `DistanceFunction` soll zur Berechnung der Abstände der Voxelcken der Blätter des Octrees eine Methode

```
float distance(float x, float y, float z, int k = 1)
```

zur Verfügung stellen, die für einen Punkt (x, y, z) den relativen Abstand zur geschätzten Isofläche ermittelt. Falls $k > 1$ ist, soll der Mittelwert zu den k nächsten Punkten der Isofläche zurückgegeben werden. Nutzen Sie `flann` zur Nächste-Nachbar-Suche. In StudIP finden Sie eine `.ply`-Datei einer Punktwolke mit zugehörigen Normalen, die Sie während der Implementierung zu Testzwecken verwenden können.

Aufgabe 3: Marching Cubes (25 Punkte)

Implementieren Sie im Octree eine Methode `reconstruct` mit folgender Signatur:

```
void reconstruct(float* vertices, float* indices, int& nVert, int& nFaces)
```

Diese Methode soll über alle Blätter iterieren und mit Hilfe der Hoppe-Funktion und der in StudIP zur Verfügung gestellten Tabelle die Polygonmuster des Marching-Cubes-Algorithmus erstellen. Die Arrays `vertices` und `indices` sollen einen indizierten Buffer des erzeugten Polygonnetzes enthalten, die Variablen `nVert` und `nFaces` enthalten nach dem Aufruf die Anzahl der erzeugten Vertices und Dreiecke. Sie müssen bei dieser Übung nicht auf redundante Vertices prüfen.

Aufgabe 4: Programm `reconstruct` (25 Punkte)

Schreiben Sie ein Programm `reconstruct` zur Polygonrekonstruktion. Es soll folgende Parameter bekommen:

```
./reconstruct <filename> -kd -ki -kn -v
```

Die Parameter `kn`, `ki` und `kd` geben an, wie viele Nächste-Nachbar-Punkte für Normalenschätzung, Normaleninterpolation und Abstandsberechnung verwendet werden. Die Voxelgröße des Octrees wird mit `-v` vorgegeben. Sinnvolle Testwerte für die in StudIP gegebenen Datensätzen bewegen sich zwischen 3 und 10 cm. Exportieren Sie das erzeugte Polygonnetz in einem `.ply`-File (ASCII-Format), das Sie mit Meshlab öffnen können.

Abgabe:

Checken Sie Ihre Programme bis zum 30.06.2021 in das git-Repository Ihrer Gruppe ein.