

Homogeneous coordinates

Die homogene Notation ist eine in der Graphik häufig verwendete Repräsentation von Positionen und Richtungen, die eine einheitliche Realisierung vielfältiger Transformationen von Positionen und Richtungen durch ein einfaches Matrix-Vektor-Produkt ermöglicht.

Quellen:

https://cg.informatik.uni-freiburg.de/course_notes/graphics_02_transformations.pdf

https://de.wikipedia.org/wiki/Homogene_Koordinaten

<http://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/>

Rendering pipeline

Die Rendering-Pipeline umfasst alle Prozesse, die während des Rasterisierungsprozesses zur Bildgenerierung ablaufen. Als Eingabe erhält die Pipeline Positionen und Primitiven, z. B. Dreiecke. Das Ergebnis der Pipeline sind für Bildpunkte berechnete Farben.

Quellen:

https://cg.informatik.uni-freiburg.de/course_notes/graphics_01_pipeline.pdf

<https://de.wikipedia.org/wiki/Grafikpipeline>

Bresenham line algorithm

Der Bresenham-Linienalgorithmus erhält als Eingabe eine Strecke, d.h. die Positionen der beiden Endpunkte. Daraus berechnet der Bresenham-Algorithmus eine Menge von Bildpunktpositionen, welche diese Strecke approximativ repräsentieren.

Quellen:

https://cg.informatik.uni-freiburg.de/course_notes/graphics_05_rasterization.pdf

<https://de.wikipedia.org/wiki/Bresenham-Algorithmus>

Phong illumination model

Das Phong-Beleuchtungsmodell ist eine Möglichkeit, die Farbe eines Punktes auf einer Objektoberfläche zu berechnen. Das Modell ist physikalisch motiviert. Es berücksichtigt Parameter, wie z. B. die Oberflächennormale, Betrachterrichtung, Beleuchtungsrichtung sowie Materialeigenschaften und Beleuchtungsstärke.

Quellen:

https://cg.informatik.uni-freiburg.de/course_notes/graphics_04_lighting.pdf

<https://de.wikipedia.org/wiki/Phong-Beleuchtungsmodell>

Williams shadow mapping

Shadow mapping ist eine Möglichkeit, nicht beleuchtete Teile einer Szene zu ermitteln. Dazu wird die Szene zunächst aus Richtung der Lichtquelle gerendert. Diese Information wird im zweiten Rendering-Schritt aus Sicht des Betrachters verwendet, um schattierte und beleuchtete Szenenteile zu unterscheiden.

Quellen:

https://cg.informatik.uni-freiburg.de/course_notes/graphics_07_shadows.pdf

https://de.wikipedia.org/wiki/Shadow_Mapping

http://cg.informatik.uni-freiburg.de/intern/seminar/shadowMapping_1978_Williams.pdf

Appel raytracing

Raytracing verfolgt Sichtstrahlen in eine Szene hinein. Der erste Schnittpunkt eines Strahls mit einem Objekt bestimmt, welcher Objektpunkt einem Bildpunkt zugeordnet wird. Der Objektpunkt wird beleuchtet und die entsprechende Farbe dem Bildpunkt zugeordnet.

Quellen:

https://cg.informatik.uni-freiburg.de/course_notes/graphics2_00_introduction.pdf

<https://de.wikipedia.org/wiki/Raytracing>

http://cg.informatik.uni-freiburg.de/intern/seminar/raytracing_1968_Appel.pdf

Marching Cubes

Der Marching-Cubes-Algorithmus berechnet triangulierte Isoflächen aus Volumendatensätzen. Eine populäre Anwendung des Algorithmus ist die Rekonstruktion von Knochenoberflächen aus Computertomographiedatensätzen.

Quellen:

https://de.wikipedia.org/wiki/Marching_Cubes

https://cg.informatik.uni-freiburg.de/intern/seminar/marchingCubes_1987_Lorensen.pdf

Mesh simplification

Mesh simplification beschreibt eine Klasse von Algorithmen, mit denen die Zahl von Dreiecken in hochaufgelösten Dreiecksnetzen reduziert wird, wobei die repräsentierte Fläche sich möglichst wenig verändert.

Quellen:

https://cg.informatik.uni-freiburg.de/intern/seminar/meshSimplification_2001_Luebke.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/meshSimplification_2004_Talton.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/meshSimplification_1997_Garland.pdf

Particle systems

Partikelsysteme bestehen aus einer großen Anzahl kleiner Partikel, deren Bewegung über die Zeit berechnet wird. Solche Systeme werden in Graphik für die Animation von Feuer, Rauch, Schnee oder auch Staub verwendet.

Quellen:

https://cg.informatik.uni-freiburg.de/intern/seminar/particleSystems_1983_Reeves.pdf

https://cg.informatik.uni-freiburg.de/intern/seminar/particleSystems_2001_Witkin.pdf

https://cg.informatik.uni-freiburg.de/course_notes/sim_02_particles.pdf