

# Frei im 3-D-Raum

Autor(en): **Settgast, Volker**

Objektyp: **Article**

Zeitschrift: **Tec21**

Band (Jahr): **141 (2015)**

Heft 15: **Spiel mit dem virtuellen Raum**

PDF erstellt am: **21.07.2024**

Persistenter Link: <https://doi.org/10.5169/seals-514972>

## **Nutzungsbedingungen**

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

## **Haftungsausschluss**

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

VON GAMES ZUR ARCHITEKTUR

# Frei im 3-D-Raum

Vom digitalen Architekturmodell zur virtuellen Umgebung: Was man tun muss, um in einem Entwurf spazieren gehen zu können. Unser Tipp: Entwicklungen für die Gameindustrie sind hier (zweckentfremdet) nützlich.

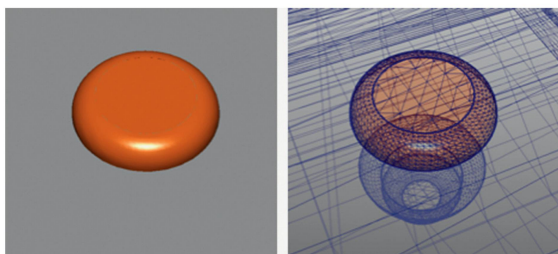
Text: Volker Settgast

In der virtuellen Realität bezeichnet Immersion das Gefühl, in eine künstliche Welt einzutauchen, und diese dann als real zu empfinden. Computerspiele erzeugen Immersion durch eine mitreissende Geschichte, glaubwürdige Charaktere und eine in sich plausible Spielwelt. Der wichtigste Faktor ist jedoch immer noch die reibungslose Darstellung der künstlichen Welt selbst.

Es können alle möglichen 3-D-Daten in einer immersiven Umgebung angezeigt werden. Da aber der Benutzer selbst entscheiden soll, wo er hinschaut, ist ein interaktiver Bildaufbau notwendig. 3-D-Modelle, die nicht für die Verwendung in einer interaktiven Anwendung erzeugt wurden, müssen angepasst werden. Game-Engines (Framework für Computerspiele: steuert den Spielverlauf, zuständig für visuelle Darstellung) wie Unity und CryEngine sind für einen schnellen Bildaufbau bei hervorragender visueller Qualität bestens geeignet. Der Weg vom 3-D-Ausgangsmaterial über die Game-Engine zum interaktiven Bild wird in seiner Komplexität oft unterschätzt. Denn die Optimierung der Geometrie kann nur zu einem gewissen Teil automatisch erfolgen. Für die besten Ergebnisse ist das menschliche Urteilsvermögen unverzichtbar.

## Von Präzision zu Performance

Die in der Architektur oft verwendeten CAD-Werkzeuge enthalten eine detaillierte Beschreibung, die der Konstruktion von Objekten dienen soll. In Game-Engines ergeben sich dadurch folgende Probleme:



Ein winziger Niet wurde automatisch in ein hoch aufgelöstes Dreiecksnetz gewandelt. Die Rückseite ist niemals sichtbar und würde nur unnötig Ressourcen verbrauchen. «Culling» heisst das Prinzip, nicht Sichtbares zu verwerfen, um die Performance zu erhöhen.

- Das Modell enthält sehr viel Geometrie, die eigentlich nicht benötigt wird, zum Beispiel Schrauben und innere Strukturen (Abb. unten). Dies führt zu einer unnötig grossen Datenmenge, die die Darstellung verlangsamt. Gerade diese Verlangsamung kann beim Betrachter zu einem Unwohlsein führen.
- Die Materialdefinitionen von CAD-Werkzeugen beschreiben oft nur physikalische, aber keine visuellen Eigenschaften. Für die Konstruktion eines Objekts reicht das aus, aber um das Material interaktiv anzuzeigen, ist eine andere Beschreibungsart nötig.
- Freiformflächen werden von ihrer platzsparenden Darstellung in eine Dreiecksgruppe gewandelt, weil möglicherweise die Game-Engine bestimmte Flächen nicht unterstützt. Die Approximation der Freiformfläche kann durch Rundungsfehler und Vereinfachung sehr eckig wirken.
- Das automatische Umwandeln von Freiformflächen kann zu unnötig vielen Dreiecken führen, die für die Darstellung möglicherweise nicht relevant sind.

## Einfach, aber nicht simpel

Die Reduktion der Datenmenge ist oft der einzig wirksame Schritt, um komplexe Geometrie in Echtzeit anzeigen zu können. So ist es ratsam, fokussiert zu vereinfachen. Soll zum Beispiel eine Küche präsentiert werden, muss diese möglichst hochauflösend gezeigt werden. Alles, was nur aus einer Distanz sichtbar ist, wie der Ausblick aus dem Küchenfenster oder die Möbel im Flur, kann stark vereinfacht werden, um den Bildfluss nicht zu stören. Es ist also eine Auswahl zu treffen, welche Teile in welchem Detaillierungsgrad benötigt werden.

Die freie Geometrie-Verarbeitungssoftware MeshLab bietet einige hilfreiche Funktionen, um den Detaillierungsgrad von Objekten zu reduzieren. Die Reduktion findet dabei unter Berücksichtigung der Darstellungsqualität statt, wobei der «quadric edge collapse decimation»-Algorithmus eingesetzt wird. Die kommerzielle Software Simplygon zielt auf die Verarbeitungskette der Spieleindustrie: Sie ermöglicht es, automatisch dynamisch zwischen unterschiedlichen Detailstufen zu wechseln, um Polygonmodelle zu erstellen.

Des Weiteren lassen sich detaillierte Modelle auch durch vereinfachte Geometrien ersetzen, die mit

einer speziellen Oberflächeneigenschaft (Textur) versehen werden. Hoch aufgelöste Bilder, die als Textur verwendet werden, sollten genau wie Geometrie kritisch ausgewählt werden: Alle «unwichtigen» Texturen verschwinden nur unnötig Platz und sind in der Grösse zu reduzieren. Texturen sollten ausserdem in ein komprimiertes Format umgewandelt werden. Eine optimierte Organisation der 3-D-Szene hilft, die Darstellung weiter zu beschleunigen. Diese Optimierungen werden von der Engine automatisch vorgenommen. Jedoch ist es notwendig, die 3-D-Daten vor dem Importieren in sinnvolle Teilstrukturen aufzuteilen. Objektteile, die mehrfach verwendet werden, sollten auch als solche identifizierbar sein. Dies ist gewährleistet, solange die 3-D-Szene mit Unterobjekten importiert wird.

Funktionale Objekte, z. B. Türen, müssen in der Game-Engine gesondert behandelt werden. Es ist zwar möglich, Animationen über Dateiformate zu importieren, aber logische Abfolgen und Programme müssen in der Engine festgelegt werden.

### Der Trick mit dem Licht

Ähnlich bedeutend wie die strukturierte Vereinfachung der Daten ist die gezielte Täuschung des Betrachters. Dinge werden nicht so dargestellt, wie sie in der Realität aussehen, sondern wie sie dem Betrachter erscheinen.

Für eine real wirkende Darstellung ist Beleuchtung wichtig. Ohne Licht und Schatten ist die Wahrnehmung von Tiefe und räumlichen Zusammenhängen auch in stereoskopischen Anzeigen unmöglich.

Eine gute Lichtberechnung ist allerdings sehr zeitaufwändig und ohne ein paar Tricks nicht schnell genug darstellbar. Um eine virtuelle Welt mit realistischer Beleuchtung interaktiv anzuzeigen, wird ein Teil der Berechnung bereits vorher erledigt. Die Abbildung oben rechts zeigt eine 3-D-Szene einmal ohne Beleuchtung und einmal mit einer vorberechneten Lichtausbreitung. Die Schatten wurden dabei in das Material der Oberflächen gemalt. Diese Lichtvorbereitung kann von gängigen Grafikwerkzeugen und sogar von Game-Engines erledigt werden.

Eine realistische Materialdarstellung führt zu einem hohen Level an Immersion. Beim interaktiven Anzeigen werden abstrakte Programme (Shader) verwendet, um die Materialeigenschaften realistisch zu imitieren. Glänzende Oberflächen werden zum Beispiel mit einem Phong-Material realisiert. Die Reflexionen der Lichtquelle wandern damit sehr realistisch über die Oberfläche, während man das Objekt aus unterschiedlichen Positionen betrachtet. Spiegelnde Materialien werden mit Umgebungsbildern (cube maps) erweitert. In vielen Fällen reicht eine vorberechnete Spiegelumgebung aus, um eine realistische Darstellung zu erzeugen.

Materialien werden in Game-Engines auch für animierte Oberflächen verwendet. Eine Wasseroberfläche zum Beispiel kann mit einem Shader-Programm bewegt werden und sieht damit fast fotorealistisch aus. Und auch die Grashalme, die sich im Wind bewegen,



Unbeleuchtete Geometrie ist schwer zu erkennen (oben). Mit vorberechneter Beleuchtung kann eine **komplexe, globale Lichtausbreitung in Echtzeit** angezeigt werden (unten).

Bäume und Laub werden mit Shadern zum Leben erweckt. Game-Engines setzen verstärkt auf eine sogenannte Metasprache für die Definition von Shadern. Ausgefallene Oberflächengestaltung muss für die jeweilige Game-Engine angepasst werden.

Die Kombination aus moderner Spiele-Engine und immersiver Umgebungen wie der Oculus Rift ermöglichen eine realitätsnahe Betrachtung von Räumen, Gebäuden und Landschaften. Der Aufwand für die Umwandlung vorhandener Geometrie zur interaktiven Darstellung ist nicht unerheblich. Bei frühzeitiger Planung lassen sich 3-D-Szenen aus der Konstruktion und aus Konzeptvisualisierungen für diese Zwecke verwenden. •

*Dr. techn. Volker Settgast*, Senior Researcher bei Fraunhofer Austria, Abteilung Visual Computing, Graz, volker.settgast@vc.fraunhofer.at

#### Literatur

Lancelle, M., Settgast, V., und Fellner, W.-D. (2008). Definitely Affordable Virtual Environment. Virtual Reality Conference.



Trailer zu den im Heft erwähnten Spielen finden Sie auf [espazium.ch](http://espazium.ch).