

Praktikum Dynamic Fur Rendering

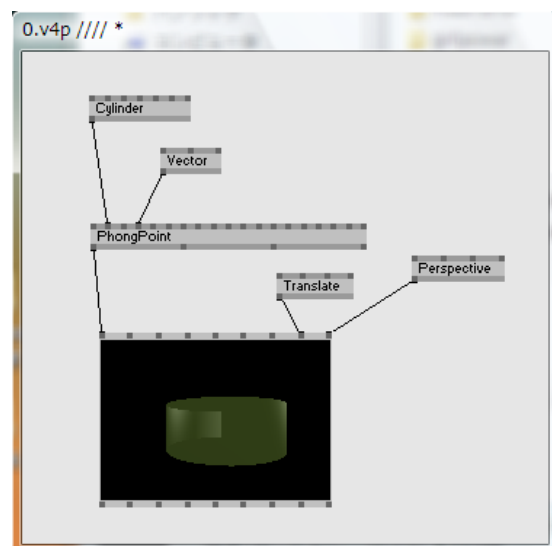
Rene Fürst
0627916
033 532, Medieninformatik

Aufgabenstellung

Die Aufgabenstellung dieses Praktikums beinhaltete die Implementierung einer Fellrendering-Methode in vvvv, welche in Echtzeit ausführbar sein musste. Das Fell sollte an beliebigen Objekten angebracht werden können und sich automatisch an die Gegebenheiten des Objekts anpassen. Zusätzlich zur Darstellung der Haare mussten sich diese auch physikalisch plausibel verhalten. Das beinhaltete die Verformung der Haare, wenn sich zum Beispiel der Körper, zu dem die Haare gehören, dreht oder schnell fortbewegt. Als Referenz wurde die Haar-Implementierung von Tomohide Kane bereitgestellt. Diese Implementierung beinhaltete eine grundlegende Methode, wie Fell in Echtzeit gerendert werden kann und eine Möglichkeit, das Fell physikalisch plausibel zu bewegen. Zusätzlich wurden zwei wissenschaftliche Arbeiten bereitgestellt, welche erweiterte Methoden zur Felldarstellung in Echtzeit beinhalteten [1] [2].

Entwicklungstools

Wie in der Aufgabenstellung verlangt, wurde die Implementation in vvvv durchgeführt. vvvv ist ein Werkzeug, mit dem verschiedene Anwendungen realisiert werden können. Unter anderem unterstützt vvvv auch Direct3D9 als Grafikschnittstelle, mit welcher die 3D-Darstellungen von Objekten erfolgen. Wie in Abbildung 1 zu sehen ist, arbeitet vvvv mit Knoten und Kanten, wobei jeder Knoten eine bestimmte Funktion darstellt. Jeder Knoten hat Eingangspins, welche sich an der oberen Seite des Knotens befinden und Ausgangspins, welche sich an der unteren Seite des Knotens befinden. Es kann immer nur eine Verbindung zu einem Eingangspin bestehen, jedoch können mehrere Verbindungen von einem Ausgangspin ausgehen. In der Figur wurde ein Zylinder-Knoten erstellt und mit einem Phong-Beleuchtungs-Knoten verbunden. Der Vektor Knoten, welcher auch mit einem Eingangspin vom Phong-Beleuchtungs-Knoten verbunden ist stellt die Position eines Punktlichtes dar. Die restlichen nicht-verbundenen Eingangspins des Phong-Beleuchtungs-Knotens wurden ohne Verbindung auf die



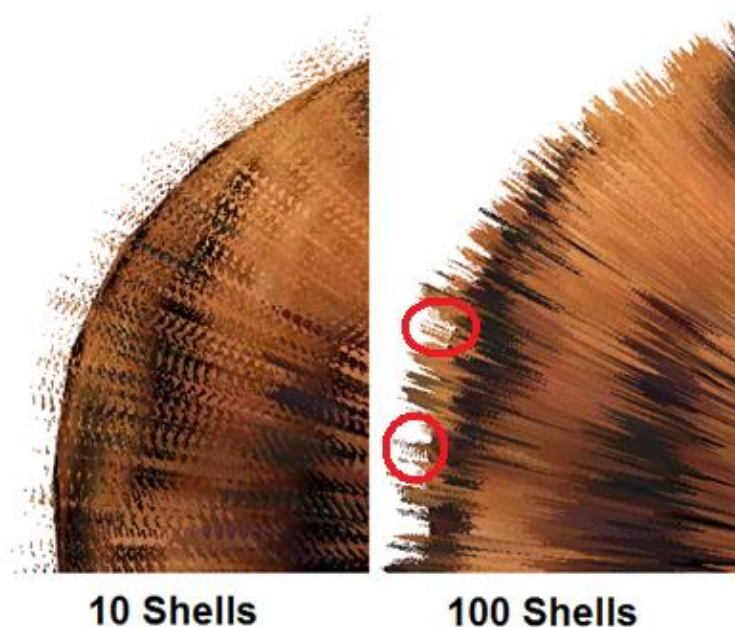
Figur 1: vvvv Beispiel

entsprechenden Werte gesetzt. Der Darstellungsausgang des Phong-Beleuchtungs-Knotens wurde mit einem Direct3D9-Rendering-Knoten verbunden, welcher dann die eigentliche Darstellung der Objekte durchführt. Die zwei restlichen Knoten dienen der Kameratransformation und der perspektivischen Darstellung der Objekte.

Neben den vordefinierten Meshes unterstützt vvvv auch das Laden von *.x Dateien, welches das Importieren von beliebigen Objekten ermöglicht. Zusätzlich können beliebige Direct3D9 Shader in Form von Effekt-Dateien frei programmiert werden und als eigenständiger Knoten implementiert werden. Es werden dabei alle DirectX9 Shadermodelle (Version 1.0 bis 3.0) unterstützt.

Renderingtechnik

Die implementierte Renderingtechnik von Haaren besteht aus zwei Teilen. Der erste Teil beinhaltet die sogenannten *Shells*, welche durch mehrmaliges, schrittweises Vergrößern der Geometrie des Objekts gerendert werden. Dabei wird aber nicht die originale Textur auf die einzelnen Hüllen aufgebracht, sondern für jeden Punkt auf dem Objekt anhand einer Haarhöhentextur festgestellt, ob an diesem Punkt ein Haar vorhanden ist, oder nicht. Ist ein Haar vorhanden und das Haar mindestens so groß, wie die aktuelle Hülle hoch ist, so wird der Punkt in der Farbe der darunterliegenden Objekttextur eingefärbt, andernfalls wird der Punkt durchsichtig gemacht. Wird dieses für viele Hüllen durchgeführt, so erkennt man ab einer gewissen Anzahl die Haare recht gut. Figur 2 zeigt einen Vergleich zwischen 10 Shells und 100 Shells.



Figur 2: Vergleich: 10 Shells und 100 Shells

Bei dem Bild mit 10 Shells kann man sehr gut erkennen, dass die Haare durch mehrere Hülle dargestellt werden. Bei dem Bild mit 100 Shells sind die Haare schon sehr gut dargestellt, allerdings sind trotz dieser

hohen Shellanzahl noch immer Artefakte am Rand der Haare sichtbar, welche dadurch entstehen, dass der Normalvektor der vergrößerten Hüllen an diesen Stellen im normalen Winkel zur Kamerarichtung steht. Das wiederum wirkt sich auf die Größe der gezeichneten Shell Texel am Bildschirm aus und der Zwischenraum zwischen den einzelnen Shells wird sichtbar.

Um dieses Problem zu beheben wurde eine zusätzliche Technik entwickelt, welche darin besteht, dass auf jede Kante im Modell eine normal auf der Objektoberfläche stehende Fläche gestellt wird, auf die die Haare an der jeweiligen Stelle gezeichnet werden. Diese Flächen werden als *Fins* bezeichnet und werden zusätzlich zu den Shells gerendert. In einem Artikel von Lengyel et al. [2] wird erwähnt, dass es genügt einen beliebigen Schnitt durch die Shells als einzelne, sich wiederholende Haartextur für die Fins zu verwenden, allerdings hat sich im Laufe des Praktikums herausgestellt, dass diese Technik bei etwas längeren Haaren doch erkennbar ist. Daher werden in der Endfassung für die Fins, analog zu den Shells, die Haare zur Laufzeit aus der Haarhöhentextur generiert. Da die Fins aber nur am Randbereich des Objekts gebraucht werden, werden diese in der Mitte ausgeblendet um Artefakte mit den Shells zu vermeiden. Die Ausblendung der Fins erfolgt mit einem sanften Übergang, damit dieser nicht bemerkt wird. Die folgende Figur zeigt die Fins an einem Objekt.



Figur 3: Fins alleine

Die Beleuchtung der Haare erfolgt mittels des Kajiya-Kay Modells von Kajiya et al. [3].

Physikalisch plausible Bewegung

Um die physikalische Bewegung der Haare zu simulieren wird eine zusätzliche Textur angelegt, welche die aktuellen Bewegungsvektoren im Tangentialraum für jede Position des Objektes speichert. Um dies zu bewerkstelligen muss die Annahme getroffen werden, dass am Objekt Texturkoordinaten existieren, welche überall im Intervall $[0; 1]$ sind und über das ganze Objekt verteilt sind. Diese Textur wird dann von Frame zu Frame upgedated und anhand der aktuellen Bewegung des Objekts verändert. Die Berechnung dieser Bewegungsvektoren erfolgt mittels folgender Formel:

$$F_{gesamt} = F_{extern} + V_{position} \times \Omega$$

F_{gesamt} gibt dabei den schlussendlichen Bewegungsvektor, F_{extern} mögliche extern wirkende Kräfte wie zum Beispiel die Gravitation an. $V_{position}$ ist die Position des aktuellen Vertexes und Ω ist die Winkelbeschleunigung des Objekts. Um an die Vertexposition des Objektes im PixelShader zu gelangen werden die gesamten Weltkoordinaten des Objekts auf eine Textur abgebildet, indem die Texturkoordinate im Vertex Shader als endgültige Vertexposition und die Vertexposition des Objekts in die Bildpixel geschrieben wird. Zusätzlich mit der zuvor erwähnten Annahme kann diese Technik für jedes Objekt angewendet werden. Um einen sanften Übergang der einzelnen Haarrichtungen zwischen zwei Frames zu ermöglichen wird der aktuelle Bewegungsvektor mit dem Bewegungsvektor vom vorherigen Frame interpoliert.

Nachdem nun die Textur mit der Bewegungsinformation fertiggestellt ist, müssen ihre Informationen nur mehr auf die Haare angewendet werden. Für jeden Punkt auf den Shells erfolgt dies in drei Schritten:

- Textur mit den Bewegungsinformationen an der aktuellen Position sampeln
- Bewegungsvektor von der aktuellen Position abziehen um die Position der „Haarwurzel“ zu erhalten
- Haarahöhentextur an dieser Position sampeln und mit dem Wert weiterrechnen

Dieses „back-tracing“ der Haare ist einfach möglich, da der Bewegungsvektor bereits im Tangentialraum ist.

Bei den Fins ist das allerdings schwerer, da ein direktes Verschieben wie bei den Shells nur einen Schnitt durch die Haare an der Fin-Ebene ergeben würde. Für stark gebogene Haare würde man dann auf den Fins nur mehr Punkte sehen. Daher müssten die Vertexpositionen im Vertex Shader verschoben werden. Allerdings erfolgt die Verschiebung der Haare bei den Shells im Tangentialraum, welcher eine beliebige Krümmung aufweisen kann und somit könnte ein gerader Vektor im Tangentialraum in Weltkoordinaten auch eine Krümmung aufweisen. Das eigentliche Problem liegt aber darin, dass diese Transformation vom Tangentialraum in den world-space nicht effizient durchgeführt werden kann. Eine Möglichkeit wäre sich in einer Textur alle Vertexposition des Objekts abzuspeichern und nach der Verschiebung im Tangentialraum die Position des Vertexes im world-space zu holen. Das funktioniert aber nur dann, wenn in floating-point Texturen gerendert werden kann, was aber bei vvvv zurzeit nicht der Fall ist.

Implementierung

Insgesamt konnten mit den von vvvv zur Verfügung gestellten Möglichkeiten die meisten Features implementiert werden. Abstriche mussten aber bei der Generierung der Fins gemacht werden, da vvvv nur Direct3D9 unterstützt und somit keine GeometryShader verfügbar sind. Daher werden die Fins mittels eines vvvv Plug-Ins in Software generiert. Weiters konnte die dynamische Bewegung der Fins auch nicht implementiert werden, da vvvv kein Rendering in floating-point Texturen unterstützt.

Resultate

Folgende Resultate wurden mit der Implementierung erzielt:



Figur 4: Rechtsdrehung mit Gravitation



Figur 5: Rechtsdrehung ohne Gravitation



Figur 7: Dichtere Haare



Figur 6: Kleinere Haare

Literaturverzeichnis

- [1] Gang Yang, H. S. (2006). *Interactive Fur Modeling Based on Hierarchical Texture Layers*.
- [2] Jerome Lengyel, E. P. (2001). *Real-Time Fur over Arbitrary Surfaces*.
- [3] Kajiya J. T., K. T. (1989). Rendering fur with three dimensional textures. *Proceedings of SIGGRAPH 89*, (S. 271-280).