Alice

Projektdokumentation
Interactive Visual Computing
WiSe 2012/13

Sascha Graeff, Max Menzel, Mario Mohr

30. Januar 2013

Zusammenfassung

In diesem Paper beschreiben wir die in unserem Kurzfilm 'Alice' verwendeten Konzepte und Werkzeuge.

1 Einführung

2 Allgemeines

2.1 Das Rendern (auf dem Cluster)

Um den POV-Ray-Code schnell rendern zu können haben wir das *ccblade*-Rechencluster der Informatikums verwendet, dieses umfasst 11 Blade-Systemen, von denen allerdings meist nur 10 verfügbar waren. Seit *Povray 3.7* ist der Code multithreadfähig, wodurch ein Rendervorgang eine Blade füllen kann. Dies gilt allerdings nur für den Rendervorgang und nicht für das Parsen der Codedatei, denn das ist immer noch singlethreaded. Da wir den ganzen Cluster zur Verfügung hatten, konnten wir nicht nur die Parallelität einer Blade ausnutzen, sondern auch gleichzeitig auf den anderen die nächsten Bilder rendern.

Um POV-Ray auf den Blades zum Laufen zu bringen, mussten wir uns erstmal mit den Kommandozeilenparametern auseinandersetzten. Auf der POV-Ray Dokumentationsseite gibt es leider keine aggregierte Übersicht über alle Parameter sondern sind nur in den Unterkapiteln der entsprechenden Themen. Eine gut aber nicht vollständig Liste kann unter [1] gefunden werden. Die wichtigsten sind:

+Hn +Wm $n \times m$-Auflösung

+A -A Antialiasing an/ausschalten

+Qn Qualität auf $n = (0..9)$ setzen

+KFIn +KFFm Frame n bis m definieren

+SFn +EFm von Frame n bis m rendern

Somit sah ein Aufruf so aus: povray foo.pov [options]

Wollten wir nun z.B. eine Szene mit 240 Frames rendern, mussten wir einfach *+KFI1* und *+KFF240* setzten und bei 10 Knoten mit *+SFn* und *+EFm* in 24er Teile zerlegen.

Um den letzten Rest Performance herauszuholen, haben wir bei einigen Szenen mit langen Parse-Zeiten (mehrere Minuten) auf einem Knoten zweimal POV-Ray so ausgeführt, dass während der Parse-Zeit des einen (ein Thread) der andere gerendert hat (die restlichen Threads).

2.2 Das colorize_intersection-Makro

Das colorize_intersection-Makro ist ein sehr simples Makro, dass es uns erlaubt hat, Objekte über csg-Intersections einzufärben. Man übergibt zwei Objekte und eine Textur. Das Makro bildet damit den Merge"der Differenz $A - B$ und der Überschneidung $A \cap B$ der beiden Objekte. Die Überschneidung allerdings wird mit der Übergebenen Textur eingefärbt, sodass als Resultat das Objekt A entsteht, das jedoch am Schnittvolumen mit Objekt B eine neue Textur erhalten hat.

2.3 Tiefenunschärfe

Tiefenunschärfe ist ein in POV-Ray sehr einfach zu verwendendes Mittel, um den Fokus

[1] `http://library.thinkquest.org/3285/language/cmdln.html`

des Betrachters zu lenken. Die Möglichkeit, auf dem Rechencluster des Informatikums rendern zu können, hat es uns erlaubt, jede Szene trotz des gesteigerten Renderaufwands in endlicher Zeit mit Tiefenunschärfe in einer akzeptablen Qualität zu rendern. Wichtig war hierbei neben dem kalkulierten Setzen des Fokuspunktes die Anpassung der Linsengröße. In der Mitte des Films findet ein Szenenwechsel vom Wald in die Taschenuhr statt. Die beiden Charaktere haben danach nur noch einen Bruchteil ihrer ursprünglichen Größe. Damit diese Änderung deutlich wird, muss die Linse der Kamera ebenfalls kleiner werden. Lediglich in der Szene, in der man über die Schulter des Gentleman hinweg Alice erblickt, wurde die Linsengröße zwecks Verstärkung des Effekts wieder stark vergrößert.

2.4 Die gentle.inc-Datei

Die gentle.inc-Datei beinhaltet den Gentleman und alles, was ihn ausmacht. Dazu gehören auch Objekte wie der Schnurrbart, der für die Pilze erneut verwendet wurde. Zu finden sind Definitionen für den Schnurrbart, das Monokel, den Zylinder sowie Kopf und Körper des Gentleman.

Der Schnurrbart besteht aus zwei gespiegelten Hälften, die mit jeweils einem Blob erstellt werden. Der "Blob-Inhalt"wiederum wird durch das iterative Platzieren von Kugeln auf einem Spline erzeugt. Die Größe der Kugeln wird nach außen hin immer kleiner.

Der Zylinder ist eine simple Kombination aus zwei (geometrischen) Zylindern, dessen Kolorierung durch Verwendung des colorize_intersection-Makros Gebrauch macht.

2.5 Alice

Alice ist eine prinzipiell sehr einfache Figur. Sie hat eine einfarbige Kugel als Kopf, der auf einem zweifarbigen Kegel sitzt, der ihr Kleid darstellt. Ihre Frisur ist ein aus drei deformierten Kugeln bestehender Blob, der den Eindruck einer halboffenen Steckfrisur vermittelt. Als Detail trägt sie eine Schleife auf dem Hinterkopf. Hier haben wir erneute Verwendung für die in gentle.inc definierte Fliege gefunden. Das Kleid erhält seine Zweifarbigkeit durch die Verwendung des colorize_intersection-Makros.

Die größte Herausforderung der Figur waren die Arme. Eine einfache Aneinandersetzung von Unter- und Oberarm ist in POV-Ray sehr einfach zu realisieren, erzeugt aber den Eindruck eines Roboters oder Strichmännchens. Bei echten Armen findet eine Verformung des ellenbogens statt, wann immer der Unterarm sich relativ zum Oberarm dreht. Diesen Umstand wollten wir zum Wohle einer realistischeren Animation bei der Erstellung der Arme implementieren.

Der erste Schritt war die Erstellung eines Makros, das drei Vektoren als Parameter erwartet: Die Position der Schulter, des Ellenbogens und der Hand. Das Makro erstellt einen Spline mit effektiv sieben Punkten. Die primären Punkte sind dabei die drei übergebenen Positionen, zwischen zweien davon liegen jeweils zwei Punkte als Tangenten, deren Position sich durch lineare Interpolation zwischen den Primärpunkten ergibt. Es liegt jeweils ein Punkt nah an der einen Seite und einer nah ander anderen. Gerade die beiden Tangenten nahe des Ellenbogens sorgen so dafür, dass die Arme nicht gebogen sind. Dass diese Punkte allerdings einen gewissen Abstand zum Primärpunkt haben, bewirkt in Kombination mit der natural_spline-Einstellung, dass eine kleine Kurve an der Stelle des Ellenbogens entsteht.

Zwei weitere Punkte am Anfang und am Ende haben die gleichen Positionen wir ihre Nachbarn, da natural_spline die beiden äußeren Punkte der Liste als Tangenten beansprucht.

Ein weiteres Makro nun nutzt das erste und gibt dem Benutzer die Restriktion, anstatt der drei Punkte nur noch die Schulterpositiom und zwei Rotationen zu übergeben. Diese werden auf den Ober- und den Unterarm angewandt. Die Länge des Armes wird allerdings durch das Makro gesetzt. Bei der Berechung der drei Punkte aus den zwei Winkeln nutzten wir die vrotate-Funktion.

Die beiden Makros zusammen ermöglichten es uns, einen Arm über einen Spline zu realisieren, der einen vernünftig anmutenden Ellenbogen darstellt, sich aber genau wie die Lösung mit z.B. zwei Zylindern als Arm durch Verwendung zweier Rotationen steuern lässt. Über diesen Spline iteriert das erste Makro mit einer Schleife, die in sehr kleinen Abständen Kugeln auf dem Spline platziert, die dem Arm seine Form geben. Die Größe der Kugeln wird in Abhängigkeit von der Position auf dem Spline gesetzt, damit z.B.

der Unterarm zum Handgelenk hin kontinuierlich dünner wird.

3 Szenen 1-10: Der Pilzwald

3.1 Die Pilzwald-Umgebung

Die Pilze in unserem Film sind als Makro realisiert, das einen Pilz mit variabler Höhe, Kopfradius, Drehung, Kopftextur und Stielscherung erzeugt. Zusätzlich erlaubt es der Parameter "gentle", anzugeben, ob der Pilz über einen enormen Schnurrbart zu verfügen hat. Eine spätere Variante des Makros ermöglicht es zudem, eine explizite Drehung des Kopfes anzugeben. Dies haben wir für die Alice ansehenden Pilze in den Wanderszenen verwendet.

Diese Makros werden in einer zweidimensionalen Schleife verwendet, um einen flächendeckenden Pilzwald zu erzeugen. In der shroomforest.inc-Datei gibt es ein Makro zum Erzeugen eines 60x60-Feldes, welches für Szenen innerhalb des Waldes ausreicht, und eines zum Aufstocken dieses Waldes auf 110x110 Pilze. Letzteres findet z.B. in der Anfangsszene, in der Alice den Wald überblickt, Verwendung. Die Unterteilung in den großen und den kleinen Wald haben wir aufgrund des enormen Parsing-Aufwands einer großen Pilzmenge ersonnen, denn das Parsen der Objekte functioniert in POV-Ray im Gegensatz zum Rendern leider nicht nebenläufig, sodass das Rechenkluster des Informatikums keine Vorteile beim Parsen generieren kann.

3.2 Die ChesSir-Cat

MARIO KNOWS ALL ABOUT IT!

4 Szenen 11-20: Die Taschenuhr

4.1 Die Taschenuhr

Die Taschenuhr lässt sich in Rahmen, Ziffernblatt, Glas und Zeiger zerlegen.

Der Rahmen besteht lediglich auch einem Torus und einem Zylinder; der Griff aus 4 Tori und einem abgerundeten Zylinder. Das Innerer der Uhr wird durch einen Zylinder beschrieben, der beim Zusammensetzten der Uhr aus dem Rahmen herausgeschnitten wird.

Das Glas ist eine simple Intersection aus einem Zylinder und einer Sphäre, wodurch eine (mono) konvexe Linse erzeugt wird. Als Material wird *M_Glass* aus *glass.inc* verwendet.

Die Zeiger bestehen 2 Zylindern in einem Blob, wobei der Sekundenzeiger noch einen Torus besitzt.

Das Ziffernblatt selbst ist nur ein Zylinder. Die Zeichen auf diesem werden über ein Makro (*writeNumbersToFace*) gesetzt: Es nimmt eine Array aus (UTF8-)Zeichen, die Schriftart sowie den Radius von Zeichenmitte bis zu Null an. Des weiteren wird die Skalierung, das Material, eine initiale Drehung und ein Flag, der bestimmt, ob die Untere Kante jedes Zeichens auf die Null oder zur xz-Ebene ausgerichtet ist, angenommen. Um den Abstand von der Mitte des Zeichens zur Null zu bestimmen, wird *min_extent* und *max_extent* verwendet. Auf dem Kreis werden nun die Zeichen gleichmäßig verteilt (dabei ist es egal wie viele Zeichen in dem Array enthalten sind). Da es sich um UTF8-Zeichen handelt kann man beliebige Zeichen einfüge: Seien es die arabischen Zahlen 1 bis 12 oder "O", "S", "W", N" oder römische Zahlen. Auf dem Ziffernblatt wurde das Makro zweimal verwendet einmal für die Zahlen "III", "VI", "IX", "XIII"(Kante zur xz-Ebene) und einmal für die kurzen Striche (Kante zur Null).

Beim Zusammenfügen der Uhr kann man auch die Uhrzeit angeben, dabei werde Stunden, Minuten und Sekunden nicht getrennt von einander eingestellt, sondern die Minuten und Stunden haben einen glatten Übergang.

4.2 Die Teaparty

Die Teaparty ist eine sehr einfache Szene, die durch die Taschenuhr als Umgebung ihre Einzigartigkeit erhält. Gerade der spiegelnde Boden ist wichtig dafür, dass die Szene nicht als leer empfunden wird. Denn effektiv besteht die Szene nur aus ihren Darstellern, einem Tisch, zwei sehr simplen Stühlen, zwei Teetassen und einer Teekanne.

Die Teekanne ist eine aus dem Netz übernommene POV-Ray-Variante des berühmten Ütah Teapot", dessen Bedeutung in der 3D-Welt in Kombination mit dem äußerst passenden Kontext uns gerade dazu nötigte, sie an dieser Stelle zu verwenden. Die Teetassen dagegen sind eine eigene Kreation, bei der wir vom Surface Of Revolution gebraucht

machten. Mit diesem war es neben der eleganten Darstellung einer Teetasse noch möglich, den darin enthaltenen Tee beliebig anzupassen, sodass z.B. die Oberfläche der Flüssigkeit in Abhängigkeit des Winkels der Tasse neigbar gewesen wäre. Von dieser Möglichkeit haben wir aufgrund von verschiedenen Erwägungen keinen Gebraucht gemacht.

4.3 Die Riesen-Pilzwald-Umgebung

Sascha versteht den Unterschied zwischen dieser Sektion und 3.1 nicht.