

Lab Report

Master Projekt System Entwicklung, SS 2013
(*Prof. Dr. J. Wietzke, Prof. Dr. E. Hergenröther*)

„Was Sandkastenspiele mit der Frischluftzufuhr in Städten zu tun
haben“

vorgelegt von

T. Sturm (709794)

A. Holike (724986)

S. Arthur (715720)

M. Djakow (718531)

01.05.2013

Inhaltsverzeichnis

1	Einleitung	2
2	Bestehende Arbeiten	3
3	Konzept	4
4	Grundlagen	5
4.1	Mathematische Verfahren	5
5	Realisierung	6
5.1	Kinect Integration	6
5.2	XNA Renderer	7
5.3	Partikelsystem	8
5.4	GUI	9
6	Zusammenfassung	10
7	Probleme	11
7.1	Echtzeitfähigkeit	11
7.2	Darstellung	11
8	Ausblick	11

1 Einleitung

hier kommt einleitendes gequatsche

2 Bestehende Arbeiten

wie wurde das in den usa gemacht

[**Kreylos2010**]

was gibt es für ähnliche ansätze

beispiele für kinect

beispiele für xny

beispiele für partikelsystem

3 Konzept

wie sieht unser konzept aus

4 Grundlagen

Hier kommt immer die Kapitelüberschrift hin, ein kleines Vorgeplänkel was im Kapitel behandelt wird.

4.1 Mathematische Verfahren

4.1.1 Verzerrung von Bildern

4.1.2 Billboarding

Um unsere Vorgabe der Echtzeitfähigkeit zu erfüllen benötigt es ein paar Tricks, die es erlauben die Komplexität unseres Renderers zu minimieren, gleichzeitig jedoch darf dem Zuschauer diese Manipulation nicht bemerken. Eine beliebte Technik hierfür ist das Billboarding. Die Idee des Billboardings basiert darauf, komplexe geometrische 3D-Objekte auf ein zweidimensionales Rechteck das sogenannte Billboard runterzubrechen. Bei dem Billboard handelt es sich meist um ein vorher berechnetes Bild von dem ursprünglich darzustellenden 3D-Objekts. Anschließend wird dieses Billboard zur Kamera ausgerichtet, dem Zuschauer fällt es somit sehr schwer zu erkennen, das es sich bei dem gezeigten Objekt um eine zweidimensionale Kopie des 3D-Objektes handelt. Diese Technik wird hauptsächlich dazu verwendet die benötigten Rechenoperationen für Objekte welche in der Ferne liegen zu minimieren. Kommt die Kamera dem tatsächlichen Objekten sehr nahe, wird meist mit einer Interpolation zwischen dem Billboard und dem tatsächlichen 3D-Objekt umgeschaltet.

5 Realisierung

Es folgt die Beschreibung unserer Realisierung.

5.1 Kinect Integration

hier wird die dll erklärt und wie sie eingebunden wird
kinect baut metrik vom bild um veraenderungen wahrzunehmen
sendet event nur wenn neues Tiefenbild vorhanden
tiefenbild blur

5.2 XNA Renderer

wie tut der renderer

warum haben wir den genommen

vorteile

5.3 Partikelsystem

wie ist das partikelsystem gebaut
was kann das ding

5.4 GUI

anbindung der elemente
darstellungsart bla blub
integration xna in gui etc
kallibration der sandkiste

6 Zusammenfassung

hier schreiben wir unsere erfahrungen rein und was wir genau hinbekommen haben. zudem sollen probleme die während der arbeit aufgetreten sind erwähnt / erläutert werden.

7 Probleme

7.1 Echtzeitfähigkeit

Leider besitzt die derzeitige Ausarbeitung diverse kleinere Probleme, welche die Echtzeitfähigkeit des Systems gefährden. Diverse teile von Berechnungen werden noch wie in ?? beschrieben auf der CPU ausgeführt, während der Teil der Visualisierung bereits auf die GPU portiert wurde. Dies führt zu erheblichen Performanceproblemen, denn es muss bei jeder Physikberechnung (jeden Frame), die Partikeldaten zwischen GPU und CPU kopiert und synchronisiert werden.

7.2 Darstellung

Die Darstellung stellte sich im Laufe des Projektes als schwieriger heraus als vorher angedacht. Hierbei kann man die Probleme auf welche wir gestoßen sind grob in Hard- und Softwareprobleme unterscheiden.

7.2.1 Hardware

Trotz das wir einen Beamer von einem Grafiklabor der Hochschule zur Verfügung gestellt bekommen haben, bemerkten wir bereits bei ersten Tests, das ein großer Farbunterschied zwischen Beamer und Monitor vorhanden ist. Leider scheint das Spektrum unseres Beamers sehr begrenzt zu sein, so das wir einen Farbunterschied zwischen weiß und gelb kaum wahrnehmen können.

7.2.2 Software

Durch die physikalische Gegebenheit das Kinect und Beamer sich an unterschiedlichen Orten befinden, entsteht bei der Projektion zusätzlich zur Verzerrung auch noch das Problem der Verschiebung. Die Kalibrierung stellte sich somit schwieriger heraus als bisher gedacht, deshalb wurden aus zeitlichen Gründen der Fokus auf Aufgaben gesetzt um schnellstmöglich eine lauffähige Version zu erstellen.

8 Ausblick

Trotz das auf uns allerlei Probleme zukamen, entstand im Laufe eines Semesters eine Echtzeit Sandkastensimulation, die bereits grundlegende Funktionalität bietet. Im Laufe des nächsten Semesters werden wir dann Aufgaben, welche in diesem Semester ein wenig vernachlässigt wurden wie z. B. die Kalibrierung nachbessern. Des Weiteren werden wir die bisherigen Physikberechnungen auf die GPU portieren um so hoffentlich wieder die Echtzeitfähigkeit des Sy-

stems zu erlangen. Auch neue Funktionalitäten sind geplant, welche notwendig sind um unser eigentliches ?? zu erreichen.

