

Medienprojekt - Projection Space

Nadja Rutsch, Philipp Hermann, Philipp Schardt, Thomas Rupp

Projektbeschreibung

Ziel der Arbeit ist es, auf recht einfache Weise ein Mehrbenutzersystem mithilfe einer Kinect aufzubauen, das Projektionen ausgehend von bis zu zwei teilnehmenden Personen verzerrungsfrei erscheinen lässt. Die angesprochenen Projektionen werden hierbei von einem mobilen Projektor erzeugt, welcher in ein Samsung Smartphone integriert ist.

Die Vorgehensweise gestaltet sich dabei folgendermaßen: Die Kinect ermöglicht die Erkennung der Hand des Nutzers, wodurch die Position des Projektors im Raum ermittelt werden kann. Davon ausgehend und mit Hilfe der Sensordaten, welche durch das Smartphone bereitgestellt werden, wird ein entzerrtes Bild berechnet. In unserer Anwendung beschränken wir uns hierbei lediglich auf Projektionen auf die vor dem Nutzer befindliche Wand. Erfasst die Projektion an Randbereichen teilweise (oder an den Seiten vollständig) die angrenzenden Raumumfassungen, so wird in diesen Bereichen kein Inhalt projiziert.

Über die Entzerrung hinaus stellt die bereits angesprochene Anwendung einen Teil unserer Arbeit dar. Es handelt sich hierbei um ein Zeichenprogramm, mit welchem die zu Beginn in weiß projizierte Fläche der Wand verändert werden kann. Je nach Position des Nutzers, d.h. Entfernung zur Wand und Blickrichtung, ist jeweils nur ein Ausschnitt sichtbar, da entsprechend nur ein Teil der Wand von der Projektion erfasst wird. Um die Veränderungen der Wandprojektion in einem Mehrbenutzerkontext verwenden zu können, wird eine Verbindung zu einer Server-Applikation aufgebaut.

Im Folgenden werden die einzelnen Bestandteile nun detailliert erläutert. Hierbei ist ebenfalls eine Beschreibung enthalten, die den gegangenen Lösungsweg aufzeigt und dort aufgetretene Schwierigkeiten anspricht.

Raummodell

Um eine Entzerrung zu berechnen, verwenden wir ein virtuelles Raummodell. Dieses erzeugen wir mittels OpenGL ES, einer Programmierschnittstelle zur Entwicklung von 3D-Graphiken. Die Funktionsweise ist dabei, die Situation aus der realen Welt, d.h. Position des Mobilprojektors in Bezug zur Wand, entsprechend im Programm realgetreu zu modellieren. Hierzu ist es notwendig, die Ausmaße des Raumes zu kennen. Bei Programmstart unserer Anwendung werden diese daher als erstes abgefragt. Angenommen wird hierbei, dass sich

die Kinect genau in der Mitte unmittelbar vor der Wand befindet, an welcher die Projektion statt finden soll. Tatsächlich programmiert wird lediglich eine Fläche, welche die Projektionswand darstellt. Darüber liegt eine Textur, welche es schließlich erlaubt, die Beschaffenheit im Sinne der Anwendung an gewissen Stellen zu beeinflussen.

Projektion

Bei der Berechnung einer entzerrten Projektion gibt es verschiedene Aspekte, die zu beachten sind. Zunächst einmal ist es von fundamentaler Bedeutung, dass wir verlässliche Sensordaten durch das Mobiltelefon erhalten. Ein schwerwiegendes Problem in unserer Situation besteht nun gerade darin, dass eben diese Daten zum einen recht starken Schwankungen unterliegen, sowie zum anderen sich gegenseitig beeinflussen. Konkret wird beispielsweise der Azimut allein dadurch verändert, dass das Mobiltelefon um die eigene Achse (welche durch den Projektor verläuft) gedreht wird. Eine Lösung zu diesem Problem konnten wir leider nicht finden. Die Schwankungen werden durch einen recht primitiven Gewichtungsfilter ausgeglichen.

Von der Kinect erhalten wir Informationen darüber, an welcher Stelle im Raum sich der mobile Projektor befindet. Wie bereits angesprochen, möchten wir ein Modell des Raumes verwenden, um die Entzerrung zu berechnen und die Beschaffenheit der Wandfläche zu verändern. Möglich wird dies dadurch, dass über die Programmierschnittstelle eine virtuelle Kamera gesteuert werden kann, welches sich im modellierten Raum befindet und die Szenerie einfängt. Das Bild dieser Kamera projizieren wir auf die Wand und können dadurch eine entsprechende Entzerrung realisieren. Mit Hilfe der Standortbestimmung durch die Kinect und den Sensordaten kann nun die exakte Blickrichtung und damit ein Blickpunkt ermittelt werden. Um die Kamera analog zum Projektor in der realen Umgebung im Programm auszurichten, muss dieser bekannt sein. Zudem ist es erforderlich einen Vektor zu bestimmen, der in die Richtung weist, in welche die Oberseite der virtuellen Kamera, bzw. das Display des Mobiltelefons weist.

Eine Herausforderung angesichts der nicht sehr verlässlichen Sensordaten stellt die Berechnung der Blickrichtung, sowie des Ausrichtungsvektors dar. Da wir zwangsläufig auf die Richtigkeit dieser Daten angewiesen sind, um über Drehungen im dreidimensionalen Raum die Ausrichtung zu erfassen, ergeben sich bei zu großen Abweichungen von der Ausgangslage nicht erwünschte Effekte, wie beispielsweise Verschiebungen der Objekte in der virtuellen Welt. Drehungen werden durch Drehmatrizen ausgedrückt, welche auf die entsprechenden Ausgangspunkte angewandt werden.

Die zu projizierende Darstellung ist abhängig vom Bereich, welcher auf der entsprechenden Wand ausgeleuchtet ist. Dies bedeutet, es existiert ein Raummodell, in welchem sich an der virtuellen Wand die jeweiligen Inhalte befinden. An den Stellen, die vom Beamer an der realen Wand ausgeleuchtet werden, wird nun der Inhalt aus dem Raummodell angezeigt.

Kinecteinbindung

Die Kinect wird in der Mitte der Projektionswand auf dem Boden platziert und blickt in einem 18° Winkel den Spielern entgegen. Sobald sie mit dem Server verbunden ist, wird kontinuierlich die Position (x, y, z) der rechten Hand, welche das Smartphone im Raum bewegt, von maximal zwei Spielern an das jeweilige Handy gesendet. Da sich das Koordinatensystem mit dem Neigungswinkel der Kinect mit dreht und somit die Z-Achse (Blickrichtung der Kinect) nicht mehr parallel zum Boden und die Y-Achse nicht mehr parallel zur Projektionswand ausgerichtet ist, muss der 18° Winkel herausgerechnet werden, um den exakten Abstand zur Wand (Z-Achse) und die exakte Höhe vom Boden (Y-Achse) zu bekommen. Nach der Berechnung erhält man die korrekten x-, y-, z- Raumkoordinaten der Hand des Spielers.

Server

Der Server, geschrieben in C, wird auf einem Windows-Pc ausgeführt und dient als Schnittstelle sowohl zwischen Kinect und Handy, als auch zur Kommunikation zwischen den beteiligten Handys selbst. Kern des Servers ist ein Thread der fortlaufend auf neue Verbindungsanfragen reagiert. Zur Verbindung mit dem Server wird zunächst ein Handshake durchgeführt um eine Verbindung erfolgreich aufzubauen.

Im ersten Schritt wird die Ip des Handys sowie die vom Nutzer eingegebenen Raummaße übertragen. Nach Empfang dieser ersten Nachricht richtet der Server die Repräsentation des Clients und alle weiteren Netzwerkverbindungen mit diesem ein und sendet eine Antwort in Form einer Socket-Nr und der zugeordneten ClientID. Die Socket-Nr dient als gemeinsame Basis für weitere Verbindungen mit dem Client.

Nach einem Erfolgreichen Handshake werden zwischen Client und Server zwei unterschiedliche Netzwerkverbindungen, jeweils eine UDP und eine TCP Verbindung, über den übermittelten Socket und dessen Folge-Socket aufgebaut. Beide Verbindungen dienen unterschiedlichen Zwecken und basieren auf dem entsprechend besseren Protokoll für ihre Aufgabe.

Die "SensorConnection", basierend auf dem UDP-Protocol, dient zum kontinuierlichen Streamen von Sensordaten. Im anfänglichen Allocation-Modus streamt das Handy die zugehörigen Accelerometer Daten zum Server damit das Handy einem entsprechenden Skelettpunkt der Kinect zugeordnet werden kann. Hat diese Zuordnung stattgefunden, wechselt die Übertragungsrichtung und es werden nun kontinuierlich die Positionsdaten des zugeordneten Skelettpunktes über die SensorConnection an das Handy gestreamt und dort zur Entzerrung genutzt.

Die "GameConnection" hingegen, basierend auf dem TCP- Protokoll, dient zur Übertragung von Events. Diese Events sind entweder Allgemeiner Natur (AllocationBegin/End) oder gehören zu den Applikationen (PointDrawnEvent). Die Bytestruktur der übertragenen Events ist festgelegt auf 32 Bytes, mit dem ersten Byte als Eventtyp und den restlichen Bytes gefüllt zugehörigen Zusatz-

daten. Die genaue Struktur für einzelne Events ist zusätzlich in einem Excel Dokument festgehalten und einsehbar. Innerhalb der Applikationen liegen die Events in Objektform vor und werden für die Übertragung in eine entsprechende Byte-Repräsentation beziehungsweise nach Empfang durch eine Eventfactory wieder in Objektform überführt.

Eventstruktur

Für die Realisation von weiterführenden Funktionen und Games besitzt die Applikation eine Eventstruktur um applikationsbezogene Datensätze einfach anzulegen und über den Server an andere Clients zu senden. Alle Events erben dabei von der Klasse SimpleEvent und stellen eine Methode bereit um entsprechende Objekte in eine Byte-Array Repräsentation zu überführen, welche dann über die "GameConnection" übertragen werden kann.

Ein SimpleEvent besitzt nur einen Eventtyp welche in der Klasse Staticstuff festgelegt sind und jeweils durch einen Byte-Wert repräsentiert werden. Dieser Wert befindet sich auch immer an erster Stelle im Byte-Array des Events. Komplexere Events in der Vererbungshierarchie können nun einfach die vorhandene ByteStruktur der Oberklasse nutzen und einfach um zusätzliche Daten erweitern. Die Länge der Event-Datensätze ist dabei aus netzwerktechnischen Gründen auf insgesamt 32 Byte begrenzt.

Desweiteren gibt es eine Eventfactory Klasse, welche in beide Richtungen funktioniert. Zum Einen werden auf dem Handy auftretende Events direkt erstellt und zudem können ankommende Events anhand ihrer ByteStruktur wieder in Objektform überführt werden. Die erstellten Events können dann je nach Applikation an das entsprechende Modell weitergeleitet und dort verarbeitet werden.

Zeichenprogramm

Das Zeichenprogramm ist eine App, die als Grundgerüst die Entzerrung verwendet. Man kann über das Smartphone auf die komplette Projektionswand zeichnen.

Interface und Interaktion

Das Interface bietet in der Mitte des Bildschirms einen Punkt, der angibt an welche Stelle gemalt wird. Durch Berühren des Displays wird die Wand an dieser Stelle gefärbt. Zusätzlich besteht die Möglichkeit die Farbe und Größe des Zeichenpunktes zu verändern. Dazu streift man mit dem Finger über die obere linke oder rechte Ecke und zieht weiter nach unten, um entweder die Größe oder die Farbe zu verändern. Beim Berühren der Ecke wird gleichzeitig das passende Interface eingeblendet (siehe Bild). Die Größe wird von oben nach unten größer, die Farbe passt sich dem Farbenstrahl an.