

Projektdokumentation

AR² Composer

von

Laura Anger	(Matrikelnr. 11086356)
Vera Brockmeyer	(Matrikelnr. xxxxxxxxxxxx)
Paul Berning	(Matrikelnr. xxxxxxxxxxxx)
Lukas Kolhagen	(Matrikelnr. 11084355)

Durchgeführt im
Master Medientechnologie
im **SS 2016** und **WS 2016/17**

Betreuer:

Prof. Dr. Stefan Michael Grünvogel
Institut für Medien- und Phototechnik

Inhaltsverzeichnis

1	Einleitung	3
1.1	Anwendungskontext	3
1.2	Projektziel	3
2	Grundlagen	3
3	Projekt Management	3
4	Materialien	3
4.1	Hardware	3
4.1.1	Rechner 1	3
4.1.2	Rechner 2	3
4.1.3	HTC Vive	3
4.1.4	Ovrvision Pro	4
4.1.5	Ueye 164xLE	4
4.1.6	Leap Motion	4
4.2	Marker	5
4.2.1	ArUco Marker	5
4.2.2	Trackingmarker	5
4.2.3	Spielfeldkalibrierung	6
4.2.4	Marker uEye Kalibrierung	6
4.3	Software	6
4.3.1	Unity	6
4.3.2	Visual Studio	6
4.3.3	OpenCV	6
4.3.4	Ovrvision Pro sdk	6
4.3.5	Leap Motion sdk	6
4.3.6	Orion Beta Software	7
4.3.7	Ueye sdk	7
4.3.8	steam vr	7
5	System	8
6	Auswertung	8
6.1	Testing durch unerfahrene Benutzer	8
6.2	Testing durch Architekten	8
7	Zusammenfassung	8

1 Einleitung

1.1 Anwendungskontext

1.2 Projektziel

2 Grundlagen

- Wissenschaftlicher Stand

3 Projekt Management

4 Materialien

4.1 Hardware

4.1.1 Rechner 1

Noch nicht klar.XMG?

alle

4.1.2 Rechner 2

Noch nicht klar.

alle

4.1.3 HTC Vive

Bei der HTC Vive handelt es sich um ein Head-Mounted Display, welches von HTC in Kooperation mit Valve [6] produziert wird. Vorgestellt wurde dieses am 1. März 2015 im Vorfeld der Mobile World Congress [4].

Laura

Die Auflösung des Displays beträgt insgesamt 2160x1200 Pixel, was 1080x1200 Pixeln pro Auge entspricht. Die Brille bietet ein Sichtfeld von bis zu 110° bei einer Bildwiederholrate von 90 Hz [1]. Zur Positionsbestimmung im Raum wird die Lighthousetechnologie von Valve genutzt. Zusätzlich sind neben einem Gyro-sensor auch ein Beschleunigungsmesser und ein Laser-Positionsmesser verbaut. Mittels speziellen Game-Controllern wird eine Interaktion mit virtuellen Objekten ermöglicht. Die eingebaute Frontkamera wird für dieses Projekt nicht verwendet. Stattdessen wird auf die Ovrvision Pro zurückgegriffen, die im Folgenden in 4.1.4 beschrieben wird.

4.1.4 Ovrvision Pro

Bei der Ovrvision Pro handelt es sich um eine open-source Stereokamera, welche über USB 3.0 mit dem Rechner verbunden werden kann [5]. Sie ist kompatibel mit Programmen wie Unity, welches für das Projekt benutzt wurde und in 4.3.1 beschrieben wird. **Nicht fertig. Brauchen wir das noch??** 1280x960x2 45fps H115 V105

Laura

4.1.5 Ueye 164xLE

Vera

4.1.6 Leap Motion

Laura



Abbildung 1: Explosionszeichnung der Leap Motion. Quelle [3].

Bei der Leap Motion [2] handelt sich um ein $7,6 \times 3 \times 1,3$ cm großes Gerät, welches es mit Hilfe von Sensoren möglich macht, Hand- und Fingerbewegungen als Eingabemöglichkeit zu nutzen. Die Idee dahinter ist, eine Eingabegerät analog zu Maus zu schaffen, welches keinen direkten Kontakt bzw. keine Berührung benötigt. Hergestellt wird die Leap Motion von der Firma Leap Motion, Inc., die ihren Hauptsitz in Amerika hat. Gegründet wurde die Firma am 1. November 2010.

Wie auf Abbildung 4.1.6 erahnt werden kann, besteht das Gerät im wesentlichen aus zwei integrierten weitwinkel Kameras und 3 einfachen Infrarot LEDs. Die LEDs haben jeweils eine Wellenlänge von 850 nm. Der durch die beiden Kameras aufgespannte Interaktionsraum der Leap Motion ähnelt einer umgedrehten Pyramide, mit einem Flächeninhalt von knapp 243 cm^2 .

Für das Projekt wurde die Orion beta software, die in 4.3.6 näher beschrieben wird. Diese Software ermöglicht unter anderem eine Erweiterung der Reichweite

der Leap Motion von 60cm auf 80cm. Diese Reichweite ist durch die Ausbreitung der LED Lichter räumlich begrenzt. Die Lichtintensität der LEDs ist wiederum durch den maximalen Strom, der über die USB-Verbindung fließt beschränkt.

4.2 Marker

Laura

4.2.1 ArUco Marker

Da sowohl für die Marker, die für den eigentlichen Trackingalgorithmus verwendet werden, als auch für sämtliche Marker, die zur Kalibrierung des Systems zum Einsatz kommen ArUco Marker verwendet werden, werden diese im Folgenden kurz erläutert.

Laura

ArUco Marker bestehen ähnlich wie QR-Codes aus einer zweidimensionalen Matrix, aus schwarzen und weißen Quadraten, die die kodierte Daten binär darstellen. Die ArUco Bibliothek kann für Augmented Reality Anwendungen genutzt werden und basiert ausschließlich auf der OpenCV Bibliothek.

4.2.2 Trackingmarker

Das System umfasst 12 Marker über die das Tracking realisiert wird. Alle Marker stimmen in Form und Farbe, sowie Material und Oberflächenbeschaffenheit überein. Sie sind würfelförmig und haben eine Kantenlänge von 46mm. Die Kanten sind in einem Winkel von 45° angefast. Die Marker bestehen aus Aluminium, welches glasperlgestrahlt ist um eine matte Oberfläche zu erzeugen. Auf die Oberseite des Markers ist mittig ein grünes Quadrat mit einer Kantenlänge von 40mm aufgebracht. Auf diesem ist, ebenfalls mittig, ein 35 mm großer ARUCO-Marker, welcher aus dem *DICT_4X4_50* generiert wurde und einen Rand von einem bit hat. Jeder Marker hat einen einzigartigen ARUCO-Marker, der einer Id von 1-12 entspricht.

Laura

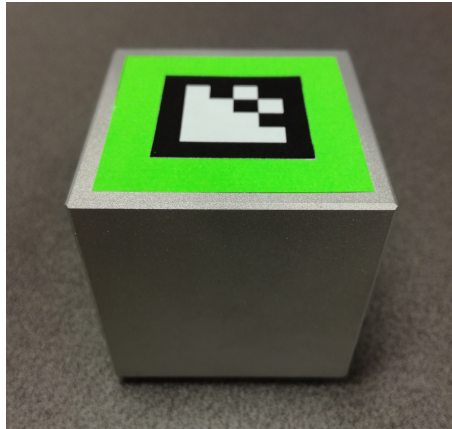


Abbildung 2: Trackingmarker mit der ID ??.

4.2.3 Spielfeldkalibrierung

Marker auf schwarzen Holzplatten von Luke

Lukas

4.2.4 Marker uEye Kalibrierung

Poster von Vera

Vera

4.3 Software

4.3.1 Unity

4.3.2 Visual Studio

Paul
oder
Lu-
kas

Laura

4.3.3 OpenCV

Wo führen wir die dlls auf die du erzeugt hast??

Vera

4.3.4 Ovrvision Pro sdk

4.3.5 Leap Motion sdk

Paul
oder
Lu-
kas

Once the image data is streamed to your computer, it's time for some heavy mathematical lifting. Despite popular misconceptions, the Leap Motion Controller

Laura

doesn't generate a depth map – instead it applies advanced algorithms to the raw sensor data.

The Leap Motion Service is the software on your computer that processes the images. After compensating for background objects (such as heads) and ambient environmental lighting, the images are analyzed to reconstruct a 3D representation of what the device sees.

Next, the tracking layer matches the data to extract tracking information such as fingers and tools. Our tracking algorithms interpret the 3D data and infer the positions of occluded objects. Filtering techniques are applied to ensure smooth temporal coherence of the data. The Leap Motion Service then feeds the results – expressed as a series of frames, or snapshots, containing all of the tracking data – into a transport protocol.

Through this protocol, the service communicates with the Leap Motion Control Panel, as well as native and web client libraries, through a local socket connection (TCP for native, WebSocket for web). The client library organizes the data into an object-oriented API structure, manages frame history, and provides helper functions and classes.

From there, the application logic ties into the Leap Motion input, allowing a motion-controlled interactive experience. Next week, we'll take a closer look at our SDK and getting started with our API. [3]

4.3.6 Orion Beta Software

4.3.7 Ueye sdk

Vera

4.3.8 steam vr

Paul
oder
Lu-
kas

5 System

6 Auswertung

6.1 Testing durch unerfahrene Benutzer

6.2 Testing durch Architekten

7 Zusammenfassung

- Wissenschaftlicher Stand

Literatur

- [1] HTC Vive. <https://www.vive.com/>. Aufgerufen: 30. November 2016.
 - [2] Leap Motion. <https://www.leapmotion.com/>. Aufgerufen: 30. November 2016.
 - [3] Leap Motion Blog. <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>. Aufgerufen: 2. Januar 2017.
 - [4] Mobile World Congress. <https://www.mobileworldcongress.com/>. Aufgerufen: 30. November 2016.
 - [5] Ovrvision Pro. <http://ovrvision.com/>. Aufgerufen: 30. November 2016.
 - [6] Valve Software. <http://www.valvesoftware.com/>. Aufgerufen: 30. November 2016.
-