

# Projektdokumentation

---

## MARc

Mixed Reality **A**rchitecture **C**omposer

---

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

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Anwendungskontext . . . . .	4
1.2	Projektziel . . . . .	4
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
<b>3</b>	<b>Projektmanagement</b>	<b>4</b>
<b>4</b>	<b>Materialien</b>	<b>4</b>
4.1	Hardware . . . . .	4
4.1.1	Computer zur Ausführung der Unity-Simulation . . . . .	4
4.1.2	Computer zur Ausführung der Tracking-Anwendung . . . . .	5
4.1.3	HTC Vive . . . . .	5
4.1.4	Spielfeldkamera . . . . .	5
4.1.5	Leap Motion . . . . .	5
4.2	Marker . . . . .	6
4.2.1	ArUco Marker . . . . .	6
4.2.2	Trackingmarker . . . . .	6
4.3	Obsolete Hardware . . . . .	6
4.3.1	Ovrvision Pro . . . . .	6
4.3.2	Webcam . . . . .	7
4.4	Software . . . . .	7
4.4.1	Unity . . . . .	7
4.4.2	Visual Studio . . . . .	7
4.4.3	OpenCV . . . . .	8
4.4.4	Ovrvision Pro SDK . . . . .	8
4.4.5	Leap Motion SDK . . . . .	8
4.4.6	Orion Beta Software . . . . .	8
4.4.7	uEye SDK . . . . .	8
4.4.8	Steam VR . . . . .	8
<b>5</b>	<b>System</b>	<b>9</b>
5.1	Menüführung . . . . .	9

---

---

5.2	Kalibrierung . . . . .	9
5.2.1	Kamerakalibrierung . . . . .	9
5.2.2	Kalibrierung des Arbeitsbereichs . . . . .	9
<b>6</b>	<b>Auswertung</b>	<b>9</b>
6.1	Testing durch unerfahrene Benutzer . . . . .	9
6.2	Testing durch Architekten . . . . .	9
<b>7</b>	<b>Zusammenfassung</b>	<b>9</b>

---

# 1 Einleitung

Virtuelle und erweiterte Realität ist bereits seit einiger Zeit in aller Munde. Mit dem Erscheinen von Oculus Rift, HTC Vive und anderen Virtual-Reality-Headsets rückt eine neue Art der Immersion beim Genuss von Videospielen in greifbare Nähe.

Wenn es jedoch um die Nutzung dieser Technologien zur Effizienzsteigerung in professionellen Umgebungen geht, so sind verfügbare Anwendungen bisher noch Mangelware. Die Entwicklung von MARC, einem Mixed-Reality-System für die architektonische Planung bei Siedlungsbauten, soll dies ändern.

## 1.1 Anwendungskontext

Die frühe Konzeptionierung zur Erschließung von Wohngebieten findet heute in Architekturbüros meist noch so statt wie vor [??] Jahren. Dazu werden simple Modelle aus leicht zu verarbeitenden Materialien – wie etwa Styropor – erstellt und als Platzhalter für die zu planenden Gebäude bei dem Entwurf verwendet.

Diese Herangehensweise macht Änderungen an den Gebäuden aufwendig und führt zu dem Umstand, dass ein bestimmter Zustand der Planung nur umständlich wiederhergestellt werden kann – zum Beispiel durch Fotografieren und späteren manuellen Wiederaufbau.

## 1.2 Projektziel

# 2 Grundlagen

- Wissenschaftlicher Stand

# 3 Projektmanagement

# 4 Materialien

## 4.1 Hardware

Zur Ausführung der MARC-Software sind diverse Hardware-Komponenten Voraussetzung. Diese Komponenten werden nachfolgend beschrieben und deren Kontext im System näher erläutert.

### 4.1.1 Computer zur Ausführung der Unity-Simulation

Die Anwendung, welche aus Unity<sup>[7]</sup> heraus erstellt wurde, benötigt einen Host-Computer, welcher sowohl mit dem HTC Vive Head-Mounted Display kompatibel,

als auch leistungsstark genug sein muss, um das Rendering der Simulation mit ausreichend hoher Bildrate ausführen zu können.

#### 4.1.2 Computer zur Ausführung der Tracking-Anwendung

##### 4.1.3 HTC Vive

Bei der HTC Vive handelt es sich um ein Head-Mounted Display, welches von HTC in Kooperation mit Valve<sup>[8]</sup> produziert wird. Vorgestellt wurde dieses am 1. März 2015 im Vorfeld der Mobile World Congress<sup>[4]</sup>.

Laura

Die Auflösung des Displays beträgt insgesamt  $2160 \times 1200$  Pixel, was  $1080 \times 1200$  Pixeln pro Auge entspricht. Die Brille bietet ein Sichtfeld von bis zu  $110^\circ$  bei einer Bildwiederholrate von  $90\text{ Hz}$ <sup>[1]</sup>. Zur Positionsbestimmung im Raum wird die Lighthousetechnologie von Valve genutzt. Zusätzlich sind neben einem Gyrosensor 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 4.3.1 beschrieben wird.

##### 4.1.4 Spielfeldkamera

IDS uEye 164xLE

Vera

##### 4.1.5 Leap Motion

Bei der Leap Motion<sup>[2]</sup> handelt es sich um ein  $7,6 \times 3 \times 1,3\text{ cm}$  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.

Laura

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

Für das Projekt wurde die Orion beta software, die in 4.4.6 näher beschrieben wird. Diese Software ermöglicht unter anderem eine Erweiterung der Reichweite der Leap Motion von  $60\text{ cm}$  auf  $80\text{ cm}$ . 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. ArUco Marker bestehen ähnlich wie QR-Codes aus einer zweidimensionalen Matrix, aus schwarzen und weißen Quadraten, die die kodierten Daten binär darstellen. Die ArUco Bibliothek kann für Augmented Reality Anwendungen genutzt werden und basiert ausschließlich auf der OpenCV Bibliothek.

Laura

### 4.2.2 Trackingmarker

Das System umfasst zwölf 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  $46\text{ mm}$ . Die Kanten sind in einem Winkel von  $45^\circ$  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  $40\text{ mm}$  aufgebracht. Auf diesem ist, ebenfalls mittig, ein  $35\text{ mm}$  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

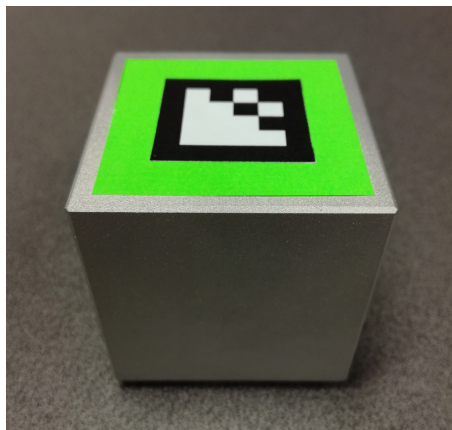


Abbildung 1: Trackingmarker mit der ID ??.

## 4.3 Obsolete Hardware

Im Laufe eines Projekts nach Art von MArC ist es kaum vermeidbar, dass die gesetzten Projektziele reevaluiert werden müssen. Die Gründe hierfür können vielfältig sein. Beispielsweise könnte die Fertigstellung eines bestimmten Teils des Projekts

Örtliche Auflösung pro Auge	Zeitliche Auflösung	Bildwinkel	
		Horizontal	Vertikal
$2560 \times 1920$	15 fps	$115^\circ$	$105^\circ$
$1920 \times 1080$	30 fps	$87^\circ$	$60^\circ$
$1280 \times 960$	45 fps	$115^\circ$	$105^\circ$
$1280 \times 800$	60 fps	$115^\circ$	$90^\circ$
$960 \times 950$	60 fps	$100^\circ$	$98^\circ$
$640 \times 480$	90 fps	$115^\circ$	$105^\circ$
$320 \times 240$	120 fps	$115^\circ$	$105^\circ$

Tabelle 1: Bildmodi der Ovrvision Pro Stereokamera.<sup>[6]</sup>

deutlich länger gedauert haben als geplant, oder es könnte sich herausgestellt haben, dass bestimmte Komponenten zueinander nicht kompatibel sind.

Im vorliegenden Projekt trat eine Kombination der beiden oben genannten Gründe auf. Das Betreiben der Ovrvision Pro am USB-Bus verschiedener während der Entwicklung verwendeter Rechner stellte sich als unberechenbar und damit leider unbenutzbar heraus. Die Kamera sorgte während der Ausführung von Unity dafür, dass mit allen anderen Geräten, die ebenfalls per USB angeschlossen waren, unterschiedlichste Probleme auftraten. Als die Situation nach dem Verbinden der Kamera in der teilweisen Zerstörung eines Mainboards gipfelte, wurde die Entscheidung getroffen, die Ovrvision nicht länger als Gerät in der Entwicklung zu verwenden.

Stattdessen war zu diesem Zeitpunkt die Idee, eine gewöhnliche Webcam zu verwenden, um die Realisierung von Augmented Reality dennoch zu ermöglichen, wenn auch ohne den Stereo-3D-Effekt, welchen die Ovrvision nativ bereitgestellt hätte.

Im weiteren Verlauf des Projekts führte eine lange Zeit ungeklärte, starke Abweichung der Positionen der realen und virtuellen Marker zur Neuordnung der Projektprioritäten. Dies hatte zur Folge, dass letztendlich auch die Webcam als Plattform für die Umsetzung der AR-Fähigkeiten von MARC aufgegeben wurde.

Nachfolgend werden die Eigenschaften und technischen Daten beider Geräte kurz beschrieben.

#### 4.3.1 Ovrvision Pro

Die Ovrvision Pro ist eine kompakte Stereokamera, welche über USB 3.0 mit dem Rechner verbunden werden kann<sup>[5]</sup>. Sie ist kompatibel mit Programmen wie Unity, welches für das Projekt benutzt wurde und in 4.4.1 beschrieben wird.

Die Bildmodi der Kamera sind in Tabelle 1 aufgeführt.



Abbildung 2: Explosionszeichnung der Leap Motion.<sup>[3]</sup>

#### 4.3.2 Webcam

### 4.4 Software

#### 4.4.1 Unity

#### 4.4.2 Visual Studio

Paul  
oder  
Lu-  
kas

Laura

#### 4.4.3 OpenCV

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

Vera

#### 4.4.4 Ovrvision Pro SDK

#### 4.4.5 Leap Motion SDK

Paul  
oder  
Lu-  
kas

Laura

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



mental 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.<sup>[3]</sup>

#### 4.4.6 Orion Beta Software

#### 4.4.7 uEye SDK

---

Vera

#### 4.4.8 Steam VR

---

Paul  
oder  
Lu-  
kas

---

## 5 System

### 5.1 Menüführung

### 5.2 Kalibrierung

#### 5.2.1 Kamerakalibrierung

#### 5.2.2 Kalibrierung des Arbeitsbereichs

## 6 Auswertung

### 6.1 Testing durch unerfahrene Benutzer

### 6.2 Testing durch Architekten

## 7 Zusammenfassung

- Wissenschaftlicher Stand

---

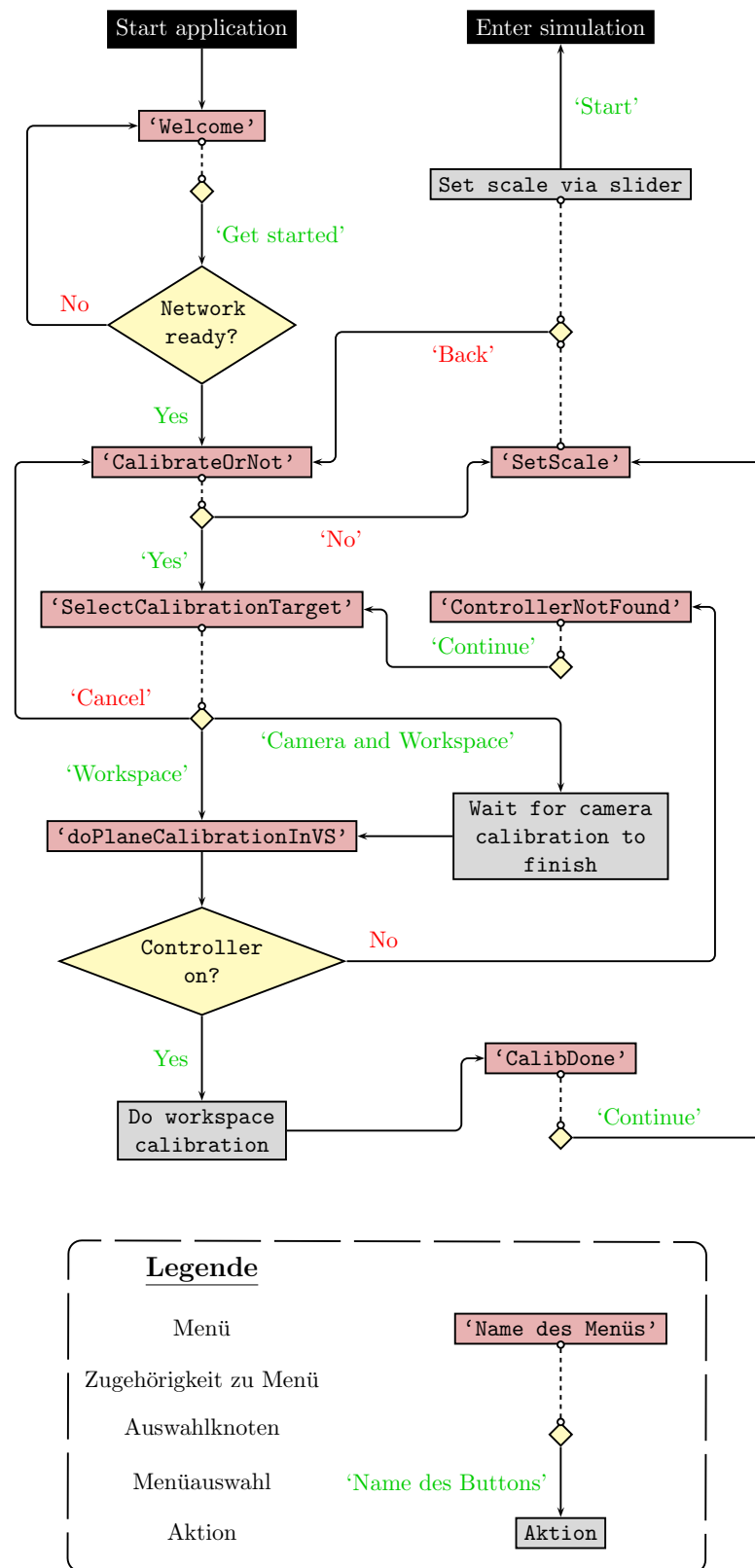


Abbildung 3: Flussdiagramm der Menüführung.

## 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] Ovrvision Pro Produktdetails. <http://ovrvision.com/product-en/>. Aufgerufen: 9. März 2017.
  - [7] Unity. <https://unity3d.com/de>. Aufgerufen: 8. März 2017.
  - [8] Valve Software. <http://www.valvesoftware.com/>. Aufgerufen: 30. November 2016.
-