Bachelorprojekt

Planung, Integration und Kalibration eines Richtungssensors und GPS-Gerätes in die kommerzielle VR-Software "Virtools"

Projektarbeit in Computer Science

Eingereicht bei: Prof. Dr. Helmar Burkhart

Oliver Koch

Universität Basel

Basel, August 2007

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Abstract

Inhaltlich handelt die vorliegende Arbeit von der Integration der Orientierung über GPS und einem Richtungssensor in ein Augmented Reality System. Diese werden über ein Virtual Reality (VR) Pack in Virtools, eine 3D-Visualisierungs-Software, eingebunden. Beim Richtungssensor muss zuerst das Koordinatensystem mit demjenigen des in Virtools verwendeten übereinstimmen, damit Kopfbewegungen richtig interpretiert werden, was mit Quaternionenmultiplikation erreicht werden kann. Anschliessend muss das Modell mit der Realität in Übereinstimmung gebracht werden. Das GPS-Gerät gibt die Koordinaten im Schweizer x,y-System aus, so dass dabei nur noch ein Offset dazugerechnet werden muss. Für das Videobild verwendet man in Virtools ein 3D-Sprite und vergrössert dieses so, dass das Bild mit der Realität übereinstimmt.

2 Ausgangslage

Lifeclipper2[1], ein vom Institute for Research and Design der Fachhochschule Nordwestschweiz zusammen mit der Universität Basel (Gruppe Helmar Burkhart) und weiteren Partnern realisiertes Projekt, versucht den öffentlichen Raum durch Augmented Reality zu inszenieren. Die Realität wird mit virtuellen Inhalten ergänzt. Art und Inhalt der Ergänzungen können durch den Benutzer mitbestimmt werden, indem dieser sich im Raum bewegt. Je nachdem wo sich ein Benutzer befindet und in welche Richtung er blickt, werden Inhalte hinzugefügt oder wieder ausgeblendet. Dazu benötigt der Anwender ein mobiles Computersystem (Computer auf

Abbildung 1: Realität ergänzt mit virtuellem Inhalt

[1] Vgl. LIFECLIPPER [2007].

speziellem Traggestell), ein Head Mounted Display (HMD) mit Videokamera, einen Richtungssensor (Tracker), der die Kopfbewegungen registriert, und ein GPS-Gerät, das die Position des Anwenders erfasst. Eine sich so bewegende Person erlebt eine Mischung von projektierter Realität und virtueller Welt.

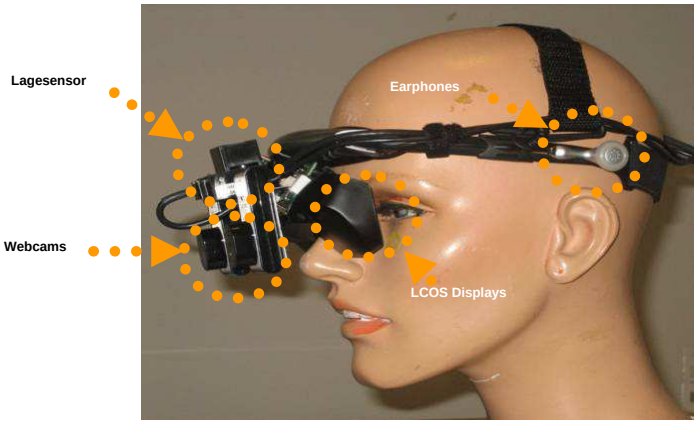


Abbildung 3: Head Mounted Display HMD

Die Realisierung des Projekts beschränkt sich auf einen Raum im Bereich Novartis Campus und Dreirosenbrücke in Basel und teilt dieses Gebiet in verschiedene Zonen ein mit unterschiedlichen Szenarien.

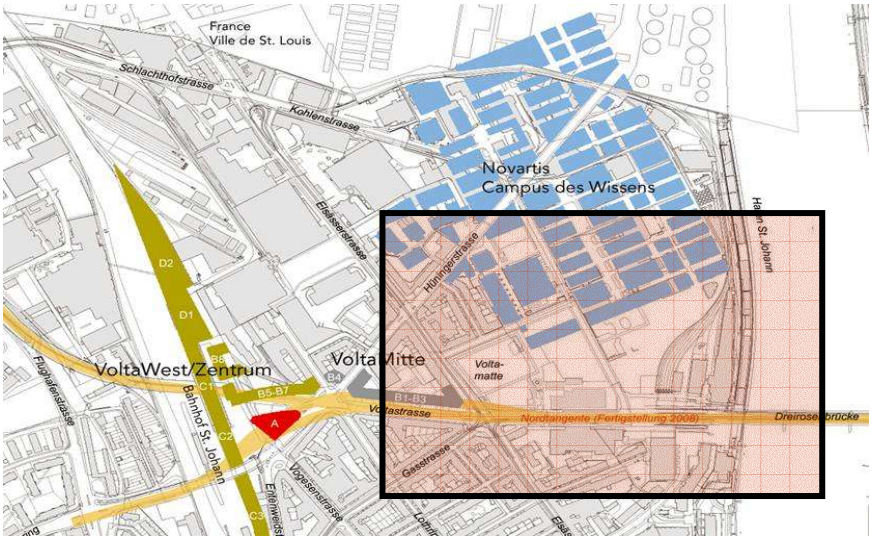


Abbildung 2: Gebiet des Projekts

Die vorliegende Arbeit handelt von der Einbindung der Orientierung über ein GPS-Gerät sowie der Einbindung eines Richtungssensors in das System.

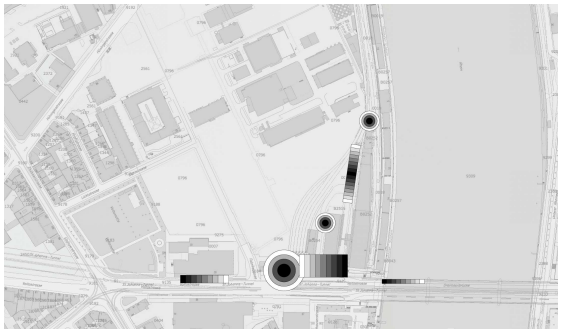


Abbildung 4: Zonen mit unterschiedlichem Inhalt

3 Problemstellung

Entsprechend den zu erledigenden Aufgaben ergeben sich fünf Milestones, die im Folgenden genauer ausgeführt werden:

- Milestone 1: Einarbeitung in die Softwareumgebung von Virtools
- Milestone 2: Alle 6 Koordinaten (Richtung und Ort) sollen via VRPN-Protokoll in Virtools eingelesen werden
- Milestone 3: Das virtuelle Modell soll mit der Realität in Übereinstimmung gebracht werden. (Markante Objekte erscheinen unter dem gleichen Richtungswinkel im Virtuellen (Modell) und in der Realität (ohne 3D-Helm) an verschiedenen Positionen im Gelände)
- Milestone 4: Einpassung des Kamerabildes
- Milestone 5: Stabilitätstest, Update-Abläufe, individuelle Kalibration bezüglich der Richtung

4 Ergebnis

4. 1 Milestone 1: Einarbeitung in die Softwareumgebung von Virtools

Virtools ist eine französische Software für 3D-Visualisierung, welche aus einem Kompositionswerkzeug (authoring application), Behavioral Engine (CK2), Render Engine, Web Player und einem Software Development Kit (SDK) besteht. Für unser Projekt ist die Version 3.5 verwendet worden.

Bei Virtools handelt es sich um eine visuelle Programmiersprache. Schiffer definiert die visuelle Sprache folgendermassen: „Eine visuelle Sprache ist eine formale Sprache mit visueller Syntax oder visueller Semantik und dynamischer oder statischer Zeichengebung.“[2] Virtools erfüllt Schiffers Vorgaben. Es existiert zwar auch ein Texteditor,

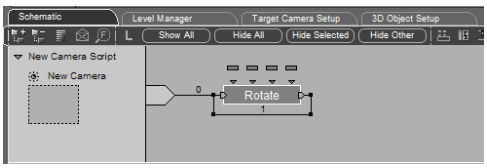


Abbildung 5: Building Block Rotate

[2] SCHIFFER [2007].

in dem zusätzliche Programmteile geschrieben werden können, aber grösstenteils erfolgt die Programmierung auf graphischer Ebene. Dabei kann auch der Ablauf eines Programms graphisch verfolgt werden, was vor allem für das Debugging von grossem Nutzen ist. Um ein Programm zu erstellen, kann aus einer Vielzahl von Building Blocks (BB) ausgewählt werden. Diese Building Blocks erfüllen jeweils eine bestimmte Funktion und lassen sich mit zusätzlichen Parametern detailliert einstellen. Mehrere BBs können zusammen-gestellt und nach Bedarf mit Linien verbunden werden, was den Programmablauf steuert.

Wenn zum Beispiel ein Objekt rotieren soll, wird zu diesem Objekt ein Script erstellt und an entsprechender Stelle der Building Block „3D Transformations/Basic/Rotation“ eingefügt, danach müssen die Parameter „Axis of Rotation“, „Angle of Rotation“, „Referential“ und „Hierarchy“ dem gewünschten Verhalten entsprechend eingestellt werden. In der Folge entsteht ein Objekt, beispielsweise eine Kamera, welche mit den angegebenen Parametern rotiert.

Mit Hilfe der Virtools Scripting Language ist es auch möglich, selber Building Blocks zu programmieren. Dies ist allerdings nur selten nötig, da durch die umfangreiche BB-Bibliothek und deren Kombinationen praktisch alles realisierbar ist.

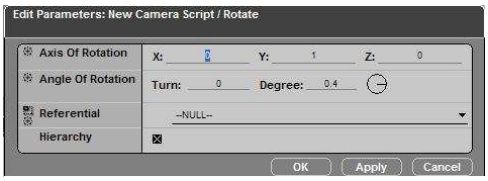


Abbildung 6: Die Parameter für BB Rotate

Ein Vorteil der visuellen Programmierung liegt ganz klar darin, dass man sich relativ schnell einen Überblick über das, was geschieht, verschaffen kann. In Virtools lassen sich mehrere Building Blocks zusammenfassen und beliebig benennen, was die Übersichtlichkeit weiter erhöht. Wie bei jeder anderen Programmiersprache muss der Benutzer zuerst das Instrumentarium zum Arbeiten kennen lernen. Beim vorliegenden Projekt sind es vor allem die verschiedenen Building Blocks und die Parameterbedeutung. Zum Debuggen verfügt Virtools über Tools, welche die Fehlersuche erleichtern. Als aufwändig oder etwas umständlich habe ich persönlich die Umstrukturierung von Programmen empfunden, bei denen graphisch teilweise viel verschoben werden musste, bis die Übersichtlichkeit wieder vorhanden war.

Für das Projekt Lifeclipper hatten die Beteiligten ein Stadtmodell zur Verfügung,

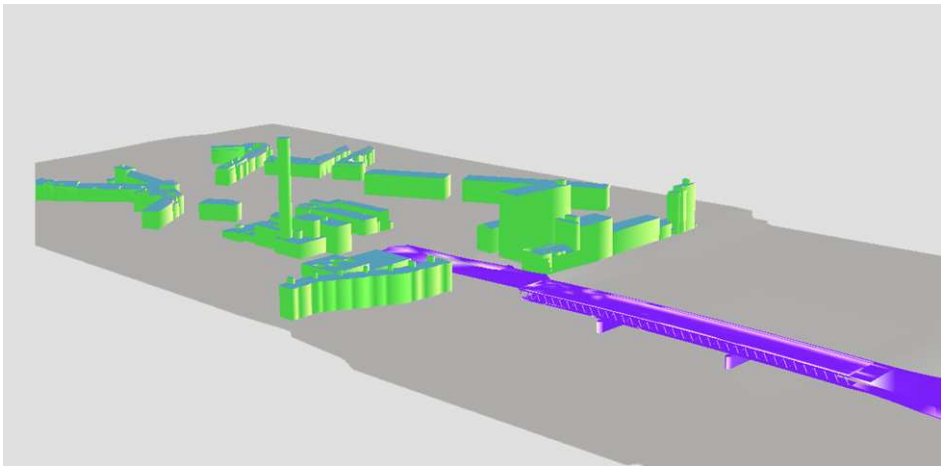
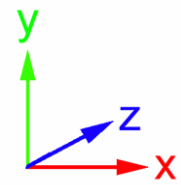


Abbildung 7: Das Stadtmodell in Virtools

welches bereits in Virtools realisiert war. Und auch die Szenerien von Lifeclipper wurden extern programmiert.



Zu beachten ist das in Virtools verwendete „linkshändige“ und für die Computergraphik übliche Koordinatensystem, bei dem die y-Achse für die Höhe steht. Die verwendeten Einheiten sind Meter.

Abbildung 8: Koordinatensystem von Virtools

4. 2 Milestone 2: Alle 6 Koordinaten (Richtung und Ort) sollen via VRPN-Protokoll in Virtools eingelesen werden

Damit die für Augmented Reality notwendige Orientierung in das Virtools-System eingebunden werden kann, wird von Virtools ein VR Pack zur Verfügung gestellt. Dieses beinhaltet zahlreiche Konfigurationsfiles, mit denen unterschiedliche Tracker-systeme eingebaut und das HMD konfiguriert werden können.

4. 2. 1 Tracker

Für die Kopforientierung kommt der InertiaCube3 von Intersense zum Einsatz. Dabei handelt es sich um einen dr kleinsten Präzisions-Orientierungs-Tracker auf dem

Markt.[3] InertiaCube3 misst simultan neun physikalische Eigenschaften: Winkelraten, lineare Beschleunigung und magnetische Felder über allen drei Achsen. Durch die Kombination von Gyrometer, Beschleunigungsmesser und Magnetometer (Messung magnetischer Feldstärke) wird ein sehr genaues Tracking erreicht. Dabei werden die Beschleunigungs- und Magnetometer-Daten vor allem für die Stabilisierung gegenüber den Erdgravitations- und Magnetfeldern benötigt. Das System verfügt über einen eingebauten Kompass, so dass die x-Achse des Trackers nach Norden zeigt.

Kalibrierung des Richtungssensors

Abhängig von der Montage des Richtungssensors muss das Koordinatensystem korrigiert, respektive in Übereinstimmung mit Virtools gebracht werden. In unserem Setup funktionierte das nicht auf Anhieb: Eine Bewegung um die x-Achse (entspricht in Virtools der Kopfbewegung nach oben/unten) erschien als eine Mischung aus mehreren Achsen.
Um präzise Resultate zu erhalten ist es wichtig, den Tracker auf den Tisch legen und ihn dann exakt um eine beliebige Achse drehen zu können. Dies ist eine Grundvoraussetzung für eine genaue Kalibration. Mit dem Konfigurationsfile VRDevice.cfg lassen sich „neutralPosition“, „neutralQuaternion“, „axisPermutation“ und „axisSign“ einstellen.

Abbildung 9: Verwendeter Tracker IntertiaCube3

Trotz stundenlangem Experimentieren mit verschiedenen Parametern gelang keine Kalibration. Erst nach Anpassungen des VRPN-Treibers und mit der Verwendung von Quaternionenmultiplikation direkt in Virtools gelang schliesslich die richtige Kalibration. Im Treiberfile vrpn_Tracker_isense.C musste die Reihenfolge für die Quaternionen geändert werden. In Virtools werden nämlich bei Quaternionen zuerst die Achsenwerte (x, y und z) und erst danach der s-Wert (vgl. Quaternionen) angegeben. Anstelle von

```
if(m_StationInfo[station].AngleFormat == ISD_QUATERNION) {
     d_quat[0] = data.Station[station].Orientation[0];
     d_quat[1] = data.Station[station].Orientation[1];
     d_quat[2] = data.Station[station].Orientation[2];
     d_quat[3] = data.Station[station].Orientation[3];}
```

[3] INTERSENSE 2005, S. 6.

muss die Reihenfolge folgendermassen lauten:

```
if(m_StationInfo[station].AngleFormat == ISD_QUATERNION) {
  d_quat[0] = data.Station[station].Orientation[1];
  d_quat[1] = data.Station[station].Orientation[2];
  d_quat[2] = data.Station[station].Orientation[3];
  d_quat[3] = -data.Station[station].Orientation[0];
}.
```

Des Weiteren verhielt sich der Tracker nicht konstant. Plötzlich wurden Rotationen wieder. anders gedeutet. Dies hing mit den Zeilen

```
if (m_reset_at_start)
     ISD_Boresight(m_Handle, station+1, true);
```

zusammen, welche Boresight-Einstellungen, die dem Winkelausgleich der Trackermontage dienen, periodisch wieder auf null stellten. Diese Zeilen wurden folglich auskommentiert.

4. 2. 2 Quaternionen[4]

Quaternionen eignen sich sehr gut zur Beschreibung von Rotationen und spielen somit im vorliegenden Projekt eine grundlegende Rolle. Deshalb soll an dieser Stelle eine kurze Einführung dazu gegeben werden: Quaternionen wurden im 19. Jahrhundert von William Rowan Hamilton als Erweiterung der komplexen Zahl eingeführt. Der Vorteil von Quaternionen gegenüber Eulerwinkeln, welche im Allgemeinen beliebt sind aufgrund der Nachvollziehbarkeit einer Rotation, liegt darin, dass sie eine Rotation eindeutig zu beschreiben vermögen und ein Gimbal Lock[6] nicht auftreten kann. Quaternionen können sich so vorgestellt werden, dass eine neue Achse definiert und in der Folge das System um diese Achse unter einem bestimmten Winkel gedreht wird. Bei den Eulerwinkeln erfolgen die Rotationen nacheinander um die verschiedenen vorhandenen Achsen: dabei steht zum Beispiel R_{zyx} für eine Rotation um die z-Achse, danach folgt eine Rotation um die y-Achse und am Schluss wird um die x-Achse

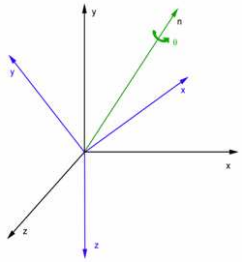


Abbildung 10: Rotation mit Quaternionen[5]

---

4 Mathematische Angaben aus: MATHEMATICS FOR COMPUTER GRAPHICS 2006, S. 90.

5 DAS EINHEITSQUATERNION [2007].

6 Gimbal Lock wird die Situation genannt, wenn bei einer Rotation mit Eulerwinkeln die erste und die dritte Achse zusammenfallen.

gedreht. Dadurch sind auch andere Reihenfolgen möglich wie R_{zxz} oder R_{zyz}.

Einzelne Drehungen im Raum kommutieren nicht, deshalb muss jeweils die Drehreihenfolge mitangegeben werden. Bei einer Quaternionenrotation entfällt diese Angabe. Ein Quaternion $q = [s, \mathbf{v}]$ besteht aus einem Skalar s und einem 3D-Vektor $\mathbf{v} = [x, y, z]$. In algebraischer Form lässt sich das schreiben als

$$q = [s + x\,\mathrm{i} + y\,\mathrm{j} + z\,\mathrm{k}]$$

wobei s, x, y und z für reelle Zahlen und i, j und k für imaginäre Einheiten stehen. Dabei gelten folgende Rechenregeln:

Zwei Quaternionen q_1 und q_2:

$$q_1 = [s_1 + x_1\,\mathrm{i} + y_1\,\mathrm{j} + z_1\,\mathrm{k}]$$

$$q_2 = [s_2 + x_2\,\mathrm{i} + y_2\,\mathrm{j} + z_2\,\mathrm{k}]$$

sind genau dann gleich, wenn die korrespondierenden Terme gleich sind. Addiert und subtrahiert werden Quaternionen folgendermassen:

$$q_1 \pm q_2 = [(s_1 + s_2) + (x_1 + x_2)\,\mathrm{i} + (y_1 + y_2)\,\mathrm{j} + (z_1 + z_2)\,\mathrm{k}].$$

Für die Multiplikation, welche nicht kommutativ ist, gibt es zusätzliche Regeln:

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1$$
$$\mathrm{ij} = \mathrm{k}, \mathrm{jk} = \mathrm{i}, \mathrm{ki} = \mathrm{j}$$
$$\mathrm{ji} = -\mathrm{k}, \mathrm{kj} = -\mathrm{i}, \mathrm{ik} = -\mathrm{j}$$

Das Produkt von q_1 und q_2 ist

$$q_1 q_2 = [(s_1 s_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + (s_1 x_2 + s_2 x_1 + y_1 z_2 - y_2 z_1)\,\mathrm{i} + (s_1 y_2 + s_2 y_1 + z_1 x_2 - z_2 x_1)\,\mathrm{j} + (s_1 z_2 + s_2 z_1 + x_1 y_2 - x_2 y_1)\,\mathrm{k}]$$

was man einfacher mit Skalar- und Vektorprodukt schreiben kann:

$$q_1 q_2 = [(s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2), s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2].$$

Dabei sind $\mathbf{v}_1 = [x_{1,} y_1, z_1]$ und $\mathbf{v}_2 = [x_{2,} y_2, z_2]$.

Das Inverse von $q = [s + x\,\mathrm{i} + y\,\mathrm{j} + z\,\mathrm{k}]$ beträgt

$$q^{-1} = \frac{[s - x\,\mathrm{i} - y\,\mathrm{j} - z\,\mathrm{k}]}{|q|^2}$$

wobei $|q|$ das gleiche ist wie

$$\|q\| = \sqrt{s^2 + x^2 + y^2 + z^2}$$

und:

$qq^{-1} = q^{-1}q = 1$.

Ein Einheitsquaternion ist ein Quaternion mit dem Betrag 1:

$\|q\| = 1$.

Rotation eines Punktes

Einheitsquaternionen können für Rotationen verwendet werden.

Ein Punkt $P = (x, y, z)$ kann in Quaternionenform als

$P = [0 + x\,\mathrm{i} + y\,\mathrm{j} + z\,\mathrm{k}]$

geschrieben werden. Man definiert eine Rotationsachse $\mathbf{u}$:

$\mathbf{u} = [x_{\mathbf{u}}\,\mathrm{i} + y_{\mathbf{u}}\,\mathrm{j} + z_{\mathbf{u}}\,\mathrm{k}]$

und daraus ein Transformationsquaternion q mit dem Rotationswinkel θ:

$q = [\cos(\theta/2), \sin(\theta/2)\mathbf{u}]$.

Das Inverse ergibt dann

$q^{-1} = [\cos(\theta/2), -\sin(\theta/2)\mathbf{u}]$

und der rotierte Punkt P' berechnet sich aus

$P' = qPq^{-1}$.

Euler-Rotation in Quaternionen überführen

Bei Euler-Rotationen spricht man bei einer Rotation um die x-Achse auch von pitch, um die y-Achse von yaw und um die z-Achse von roll. Die Bezeichnungen stammen aus der Flugsteuerung[7]. Dabei ist zu beachten, dass wenn roll, pitch und yaw kombiniert werden, sich die einzelnen Rotationen immer auf die originalen Achsen beziehen. Einzelne Rotationen können folgendermassen als Quaternionen geschrieben werden:

$q_{roll} = [\cos(\theta/2), \sin(\theta/2)[0,0,1]]$
$q_{pitch} = [\cos(\theta/2), \sin(\theta/2)[1,0,0]]$
$q_{yaw} = [\cos(\theta/2), \sin(\theta/2)[0,1,0]]$

Diese einzelnen Rotationen können kombiniert werden zu einer Rotation

$q = q_{yaw}q_{pitch}q_{roll} = [s + x\,\mathrm{i} + y\,\mathrm{j} + z\,\mathrm{k}]$

wobei

[7] ROLL-PITCH-YAW-WINKEL [2007].

$$s = \cos(\tfrac{yaw}{2})\cos(\tfrac{pitch}{2})\cos(\tfrac{roll}{2}) + \sin(\tfrac{yaw}{2})\sin(\tfrac{pitch}{2})\sin(\tfrac{roll}{2})$$
$$x = \cos(\tfrac{yaw}{2})\sin(\tfrac{pitch}{2})\cos(\tfrac{roll}{2}) + \sin(\tfrac{yaw}{2})\cos(\tfrac{pitch}{2})\sin(\tfrac{roll}{2})$$
$$y = \sin(\tfrac{yaw}{2})\cos(\tfrac{pitch}{2})\cos(\tfrac{roll}{2}) - \cos(\tfrac{yaw}{2})\sin(\tfrac{pitch}{2})\sin(\tfrac{roll}{2})$$
$$z = \cos(\tfrac{yaw}{2})\cos(\tfrac{pitch}{2})\sin(\tfrac{roll}{2}) - \sin(\tfrac{yaw}{2})\sin(\tfrac{pitch}{2})\cos(\tfrac{roll}{2})$$

Wenn man zum Beispiel mit der Euler-Rotation folgende Kombination hat:

$$roll = 90°$$
$$pitch = 180°$$
$$yaw = 0°$$

dann errechnen sich die Werte für das Quaternion zu:

$$s = 0$$
$$x = \cos(45°)$$
$$y = -\sin(45°)$$
$$z = 0$$

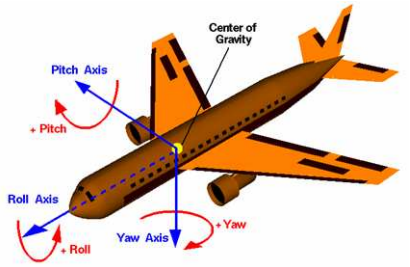


Abbildung 11: Roll-Pitch-Yaw-Winkel

Zu beachten gilt, dass Euler-Rotationen auch als Kombination von Achsen wie zum Beispiel mit zy'x'' angeben werden können, was bedeutet, dass zuerst um die originale z-Achse gedreht wird, dann um die neu entstandene y-Achse (entspricht y') und als Letztes um die daraus entstandene x-Achse (entspricht x'').

Quaternionenmultiplikation

Wie bereits erwähnt ist die Multiplikation von Quaternionen nicht kommutativ. Zum Berechnen wird die oben angegebene Formel verwendet:

$$q_1 q_2 = \left[(s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2), s_1\mathbf{v}_2 + s_2\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2\right]$$

oder eine hermitesche Matrix[8]:

$$q_1 q_2 = \begin{pmatrix} s_1 & -x_1 & -y_1 & -z_1 \\ x_1 & s_1 & -z_1 & y_1 \\ y_1 & z_1 & s_1 & -x_1 \\ z_1 & -y_1 & x_1 & s_1 \end{pmatrix} \cdot \begin{pmatrix} s_2 \\ x_2 \\ y_2 \\ z_2 \end{pmatrix}$$

Abbildung 12: Ohne Korrektur

Durch Quaternionenmultiplikation besteht eine gute Möglichkeit, zahlreiche notwendige Korrekturen vorzunehmen. Wenn zum Beispiel der Tracker nicht genau horizon-

8 ORIENTIERUNG VON LASERSCANNER-PUNKTWOLKEN [2007], S. 7.

tal, sondern leicht nach oben zeigend auf dem HMD montiert ist, kann das mit Quaternionenmultiplikation korrigiert werden. Dies wird im folgenden Beispiel ersichtlich: Die Abbildung 12 zeigt als Annahme die originale Orientierung eines Kopfes ohne jegliche Korrektur, dessen Stellung deutlich nach oben gerichtet ist (um die x-Achse verdreht). In Abbildung 13 sieht man den Kopf in gerader Stellung aus vier verschiedenen Perspektiven. Um dies zu erreichen, wird die Originalorientierung zuerst mit der Richtung multipliziert:

$$q_1 q_2 = [s_1, x_1, y_1, z_1] \cdot [s_2, x_2, y_2, z_2] = \begin{pmatrix} s_1 & -x_1 & -y_1 & -z_1 \\ x_1 & s_1 & -z_1 & y_1 \\ y_1 & z_1 & s_1 & -x_1 \\ z_1 & -y_1 & x_1 & s_1 \end{pmatrix} \cdot \begin{pmatrix} s_2 \\ x_2 \\ y_2 \\ z_2 \end{pmatrix} = [1,0,0,0] \cdot [0.707,0,0.707,0]$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0.707 \\ 0 \\ 0.707 \\ 0 \end{pmatrix}$$

$$= [0.707,0,0.707,0] = [\cos(90/2), \sin(90/2)[0,1,0]]$$

Das Resultat entspricht einer Drehung um 90° um die y-Achse.

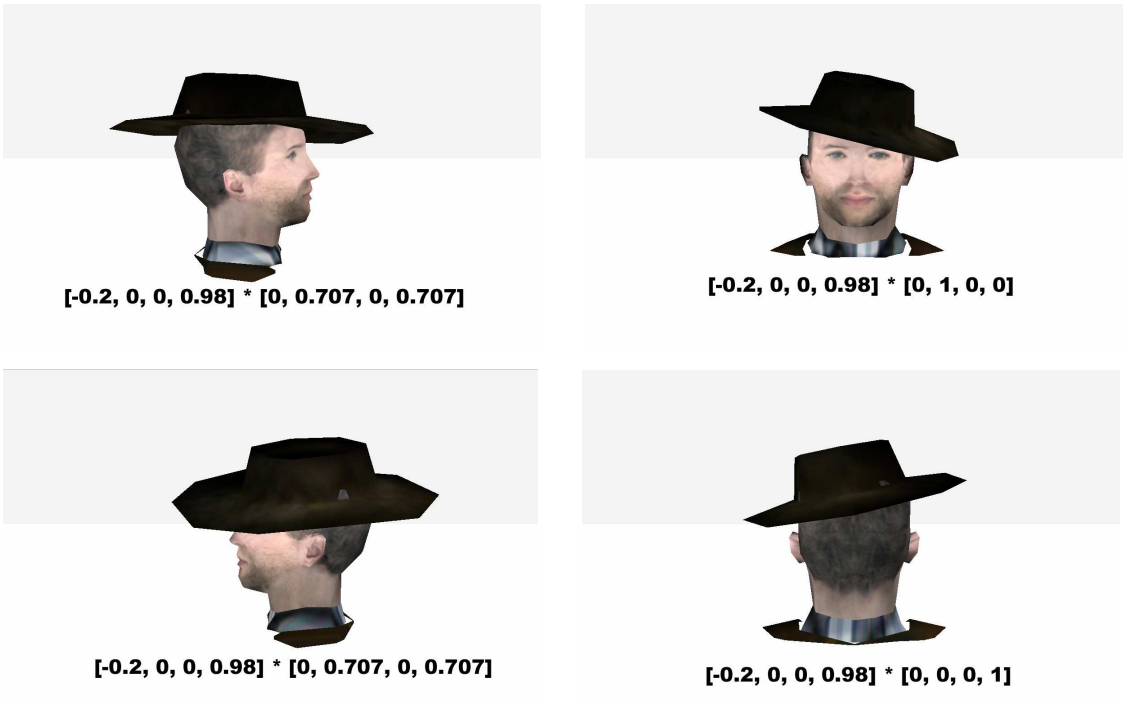


Abbildung 13: Mit Korrektur

In einem zweiten Schritt wird die Neigungskorrektur, in unserem Beispiel $[0.98,-0.2,0,0]$, mit dem vorherigen Resultat multipliziert:

$[0.98,-0.2,0,0]\cdot[0.707,0,0.707,0]=[0.69286,-0.1414,0.69286,-0.1414]$.

Daraus erfolgt insgesamt eine Drehung um 90° um die y-Achse und um 23.0692° um die x-Achse, womit das Gesicht in gerader Stellung erscheint. Je nach Orientierung der y-Achse, also der Blickrichtung, muss das eine Quaternion angepasst werden.

Als Vergleich zum vorangegangenen Beispiel ist in Abbildung 14 die zweite Multiplikation vertauscht, also:

$[0,0.707,0,0.707]\cdot[0,0.2,0,0.98]$,

was die Auswirkungen der Nichtkommutativität sehr schön erkennen lässt.

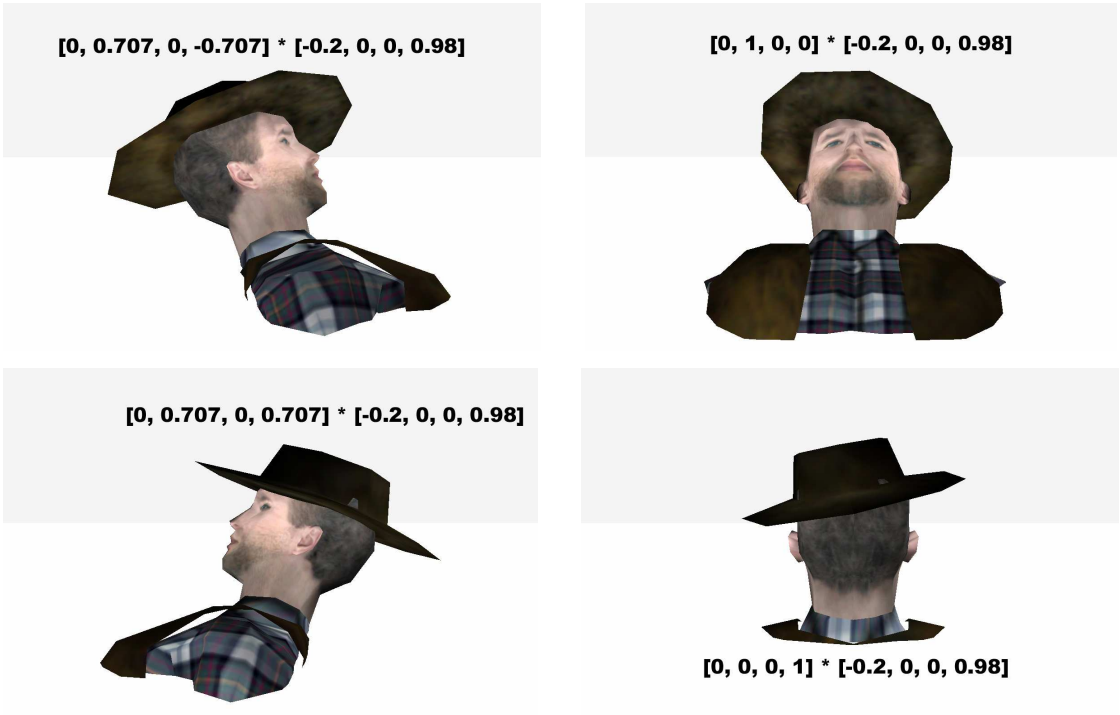


Abbildung 14: Mit Korrektur, Multiplikation vertauscht

Damit bei Lifeclipper2 die Werte q, welche vom Tracker kommen und in Virtools über die BB VRPN Init und VRPN Tracker ausgegeben werden, richtig interpretiert werden, ist q folgendermassen zu verarbeiten:

$[-1,0,-1,0]\cdot([0,0,-0.707,-0.707]\cdot(q\cdot[0.707,0.707,0,0]))$.

Dabei gilt zu beachten, dass Quaternionen in Virtools in der Form $[x, y, z, s]$ verwendet werden, beim vorliegenden Projektbericht aber die allgemein übliche Schreibweise $[s, x, y, z]$ übernommen wird. In Virtools steht folglich das hier angewandte jeweils erste Element am Schluss.

4. 2. 3 GPS[9]

Für die GPS-Ortung wird das Leica-Gerät GPS1200 GX1230 GG eingesetzt. Es handelt sich um ein professionelles Navigationsgerät, welches mit einer Genauigkeit im Zentimeterbereich arbeitet. Diese Genauigkeit wird erreicht, indem das Gerät mit einem GNSS (Global Navigation Satellite System) Measurement Engine ausgestattet ist, das neben GPS auch noch GLONASS[10] unterstützt und somit im Vergleich zu GPS mit fast 100% mehr Satelliten arbeiten kann. Zusätzlich wird die RTK-Technologie (Real Time Kinematik) eingesetzt. Dabei werden die Daten, ähnlich wie beim DGPS (Differential GPS), mit mehreren Referenzpunkten verglichen, was zwar eine kostenpflichtige Leitung für Datenkommunikation voraussetzt, aber die Genauigkeit enorm erhöht.

Abbildung 15: GPS1200 GX1230

Da die Datenausgabe des Gerätes verschiedenste Formate beherrscht, können Daten direkt im Schweizer Koordinatensystem CH-1903 mit x- und y-Wert und einer Höhe h erhalten werden.

Ansonsten gibt es für die Umrechnung eine Näherungsformel[11], welche bezüglich Genauigkeit im 1-Meter-Bereich liegt und die zu Beginn des Projekts für die Berechnung von Fixpunkten eingesetzt worden ist (siehe Anhang).

---

9 Vgl. GPS1200 BROCHURES 2006.

10 GLONASS ist das Satellitennavigationssystem des russischen Verteidigungsministeriums und ähnlich aufgebaut wie GPS.

11 NÄHERUNGSLÖSUNG FÜR DIREKTE TRANSFORMATION 2005, S. 1.

4. 3 Milestone 3: Das virtuelle Modell soll mit der Realität in Übereinstimmung gebracht werden.

Augmented Reality beinhaltet zwei verschiedene „Welten“. Reelle Bilder, aufgenommen mit einer Videokamera, werden kombiniert mit vom Computer berechneten Modellen. Dieser Vorgang macht verschiedene Kalibrierungsschritte notwendig. Am Anfang besteht die Gefahr, diese verschiedenen Aspekte nicht wirklich getrennt zu betrachten. Wenn aber zuverlässige Resultate erwünscht sind, ist dieses Vorgehen nicht optimal, sondern die Durchführung der Kalibrierung ist Schritt für Schritt erforderlich. Für die Modellkalibrierung ist es von Vorteil, das Videobild ganz auszublenden. Zu Beginn kann man noch ohne GPS-Gerät arbeiten und sich auf Fixpunkte im Modell beschränken. Dabei werden markante Punkte im Modell gesucht und diese mit der Realität abzugleichen versucht. Es muss darauf geachtet werden, dass die Fixpunkte sehr genau markiert sind, zumal bereits kleine örtliche Abweichungen beträchtliche Unterschiede im Modell bewirken können. Um das Modell mit der Realität in Übereinstimmung zu bringen, fixiert man durch das HMD einen markanten Punkt im Modell, klappt anschliessend die Brille nach oben und vergleicht, ob sich der Punkt in der Realität am gleichen Ort befindet. Allenfalls kann die Positionierung des Modells, welche ja durch den Tracker bestimmt wird, mit folgenden Parametern korrigiert werden: Korrekturen der x-, y- und z-Achse und die Perspektiveneinstellungen der Kamera (Grösse). Dazu ist in der Virtoolsumgebung eine spezielle Tastaturbelegung programmiert worden, so dass jemand mit aufgesetztem HMD die Eindrücke übermittelt und gleichzeitig die Werte in der gewünschten Grössenordnung über das Keyboard abgeändert werden könnnen. Dadurch lässt sich das Modell möglichst genau mit der Realität in Übereinstimmung bringen.
Als nächstes erfolgt die Einbindung des GPS-Gerätes, was in unserem Projekt keine Probleme bereitet hat. Da sich aus dem Gerät direkt die x-,y- und z-Werte übernehmen lassen, muss die Umrechnung nicht mehr durchgeführt werden. So bleibt lediglich das Errechnen einer Translation, damit die Nullpunkte übereinstimmen. Dazu können die Koordinaten von markanten Punkten im Modell mit den erhaltenen GPS-Daten am gleichen Ort verglichen werden. Die Translation lässt sich in Virtools mit einer Additionsbox korrigieren. Für die Anwendung in unserem Projekt ergeben sich die Werte:

$x = -610837, y = 0, z = -268935$.

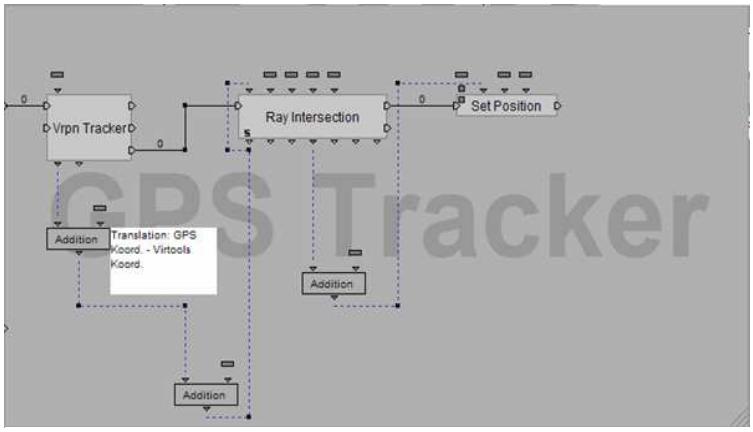


Abbildung 16: Einbindung GPS in Virtools

Die Höhenwerte stammen aus zwei Gründen direkt vom Modell und nicht vom GPS. Erstens ist das GPS im Bereich Höhenmessung weniger genau als in den anderen Dimensionen und zweitens würden sich allfällige Modellfehler stärker negativ auswirken als in den anderen Dimensionen. In Virtools gibt es eine BB „Ray Intersection“, mit der es möglich ist, die Entfernung zum Boden zu messen. Dazu werden alle Objekte, welche den Boden bilden, zu einer Gruppe zusammengefasst, folglich kann zu den korrigierten x- und z-Werten aus dem GPS jeweils der y-Wert des Bodens berechnet werden. Zusätzlich wird noch die Körpergrösse der beteiligten Person dazu addiert und an dieser Stelle die Kamera positioniert.

Im Anschluss daran wird die Kombination von GPS und Tracker überprüft. Im Gelände zeigt es sich oft als schwierig, genügend markante Punkte mit dem für diesen Zweck doch eher rudimentären Modell zu erhalten. Da keine Brückengeländer vorhanden sind, ist es zum Beispiel kaum auszumachen, wie weit das Trottoir wirklich geht und wo sich der Brückenrand befindet. Aus diesem Grund ist das Modell mit zusätzlichen Markierungen ergänzt worden. Diese bestehen aus vier Würfeln mit Kantenlänge 20 cm, aufgestellt auf einer geraden Linie in einem Abstand von jeweils 3 Metern. So kann der Anwender versuchen, sich genau auf einen Würfel zu stellen, die Würfel anzufassen und die Dimension abzuschätzen oder den Würfeln entlang zu gehen. Dieses Vorgehen vermittelt ein viel besseres Gefühl für die Distanzen als das alleinige Vorhandensein grosser Gebäude. Das Modell kann somit noch einmal ganz genau an die Realität angepasst werden.

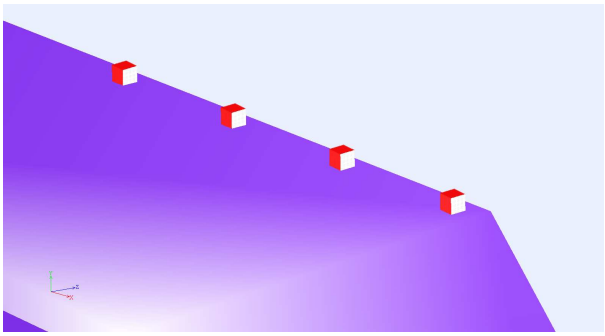


Abbildung 17: Bodenmarkierungen für die genaue Kalibrierung

4. 4 Milestone 4: Einpassung des Kamerabildes

Am Anfang hat sich die Frage gestellt, wie das Kamerabild mit dem Modell kombiniert werden kann. Dafür gibt es in Virtools die Möglichkeit, ein 3D-Sprite zu erstellen. Es handelt sich dabei um ein zweidimensionales Bild – oder im vorliegenden Fall um eine zweidimensionale Videoprojektionsfläche, welche wir Backdrop nannten – das eine Position und eine Orientierung hat und worauf das vom Video aufgenommene Bild projiziert wird. Durch den Sprite Type „Billboard“ wird erreicht, dass die Projektionsfläche immer als Normale zur Blickrichtung, respektive zu der aktiven Kamera liegt. In einer späteren Phase könnte noch versucht werden, das Video auf eine dreidimensionale Fläche mit angewinkelten Rändern zu projizieren und dadurch eventuell eine bessere Wirkung zu erzielen.
Die Dimension des Videobildes lässt sich durch verschiedene Parameter beeinflussen: Bei der „Camera“ kann das „Field of View“ eingestellt und bei der Backdrop können die Dimension und/oder der Abstand geändert werden. Durch das Einstellen der „Render Priority“ im Hierarchy Manager kann verhindert werden, dass, wenn die Backdrop-Position vor den hintersten Gebäuden läge, diese Gebäude überblendet werden. Ähnlich wie bei der Modelleinpassung ist es auch hier von Vorteil, wenn das Modell ausgeblendet und nur noch das Kamerabild mit der Realität verglichen wird.
Obwohl wir im Projekt während der gesamten Kalibrierungsphase immer wieder mit der Kameraeinpassung arbeiteten und verschiedene Varianten ausprobierten, wurde am Schluss die Backdrop-Grösse auf den gesamten Bildschirm gelegt und das Kamerabild über die Parameter „Fild of View“ und „Focal Length“ einigermassen der Originalgrösse angepasst. Dabei konnten wir feststellen, dass eine gewisse Abweichung zwischen Realität und Video nicht als störend wirkte, zumal nur selten ein direkter Vergleich zwischen Videobild und Modell gemacht werden kann. Das Videobild soll eher einen korrekten Eindruck bezüglich der Distanz zu einem Objekt verglichen mit derjenigen in der Realität hinterlassen, was für das realistische Empfinden eine grössere Rolle spielt.
Bis zu diesem Zeitpunkt ist beim Projekt nur eine Kamera im HMD aktiv und auf beiden Displays erscheint das gleiche Videobild. In einer weiteren Phase des Projektes soll dann jedoch noch die zweite, auf dem HMD montierte Kamera miteinbezogen werden, womit das rechte Display das Bild der rechten Kamera erhält und umgekehrt.

4. 5 Milestone 5: Stabilitätstest, Update-Abläufe, individuelle Kalibration bezüglich der Richtung

Die Schlussphase gestaltete sich ziemlich hektisch. Da das GPS-Gerät erst einige Tage vor Präsentationstermin wieder verfügbar war, mussten innerhalb kurzer Zeit zahlreiche Arbeiten erledigt werden. So wurden zum Teil mehrere neue Versionen für die verschiedenen Szenen pro Tag geliefert, welche jedes Mal angepasst werden mussten[12]. Das gesamte Traggestell wurde neu gebaut und war dadurch angenehmer zu tragen, zum Arbeiten und Korrigieren auf dem Laptop jedoch umständlicher.

Zahlreiche Bugs galt es zu korrigieren. So zum Beispiel wechselte das HMD nach einer bestimmten Zeit in den Sleepmodus und die Bildschirme wurden schwarz, weil nicht der originale Tracker verwendet wurde, dies war durch einen Tastendruck auf die Einheit jeweils wieder zu beheben. Folglich montierten wir zusätzlich auch noch den originalen Tracker. Diese Veränderung beeinträchtigte ihrerseits die Performance des verwendeten InterSense Trackers und dadurch musste der originale Tracker mit einem längeren Kabel bestückt respektive ummontiert werden.

Wie lange hält ein Akku? Wie oft stürzt das System ab? Wie ist ein Systemabsturz am kundenfreundlichsten zu beheben? So lauteten die Fragen, auf welche wir ein paar Tage vor der Präsentation noch keine exakten Antworten bereit hatten. Die Installation erwies sich aber als sehr stabil und die Akkus ermöglichten eine angenehm lange Betriebsdauer. Mit einer externen Tastatur und einer Maus mit USB-Anschluss konnten, falls notwendig, direkt über die Bildschirme des HMDs notwendige Eingriffe durchgeführt werden, ohne dass der Computer ausgebaut werden musste.

Insgesamt konnten Lösungen gefunden werden, welche alle Beteiligten positiv stimmten und bei der Präsentation die Möglichkeiten des Gerätes aufzuzeigen vermochten.

---

[12] Vgl. Anhang 8. 3: Anpassung an eine neue Virtools Composition.

5 Ausblick

Obwohl bis zur Präsentation nur eine Kamera verwendet worden ist, sich das Modell manchmal noch etwas ruckartig bewegt und die exakte Einstellung des Gerätes auf ein Individuum vernachlässigt werden musste, vermag das Projekt zu begeistern. Die Entwicklung des Systems wird weitergeführt und kann zukünftig noch optimiert werden. Dabei gilt es zu überlegen, wie gross der Einfluss von Verkehr (LKWs mit eigenen GPS-Geräten), Metallkonstruktionen, Lichtampeln und hohen Häusern auf das Tracking- und GPS-System ist und wie Ungenauigkeiten vermieden werden können. Auch die Beeinträchtigung der Performance durch die grossen Datenmengen muss untersucht werden.
Die Orientierung kann noch verfeinert und weitere Korrekturen, wie zum Beispiel die Distanz des GPS-Gerätes zum Richtungssensor, können berücksichtigt werden. Zudem muss eine Möglichkeit gefunden werden, das System schnell und einfach an die verschiedenen Personen anzupassen. Eventuell lässt sich dazu eine spezielle Kalibrierungssoftware erstellen. Und wie bereits erwähnt, wird das reelle Bild in Zukunft von zwei Kameras aufgenommen, dies gilt es einzurichten. Schliesslich soll mit einer Änderung der Backdrop-Form, welche rechteckig ist, das Resultat zu optimieren versucht werden.

6 Schlussbemerkung

Das Projekt Liveclipper2 war sehr anspruchsvoll und spannend und eröffnete mir Einblicke in die Gebiete Augmented Reality und visuelle Programmierung. Projektartiges Arbeiten und die Zusammenarbeit mit Personen aus verschiedenen Bereichen gestalteten sich interessant und herausfordernd. Als grösste Herausforderung erwies sich der relativ knappe Zeitrahmen mit einem fixen Präsentationstermin insofern, als die Aufgabe komplett neu war und somit Umwege in Kauf genommen werden mussten.
Ich möchte mich für die sehr gute Zusammenarbeit vor allem bei Martin Guggisberg und Jan Torpus und bei allen weiteren Beteiligten herzlich bedanken.

7 Literaturverzeichnis

DAS EINHEITSQUATERNION: Andreas Nüchter, Das Einheitsquaternion, aus: http://www.ais.fraunhofer.de/ARC/3D/download/diplom/node27.html, 2002 (Zugriff: 16. Juli 2007)

GPS1200 BROCHURES: GPS1200 Brochures, aus: http://www.leica-geosystems.com/corporate/de/products/total_stations/lgs_4521.htm, unter „Kundendienst“/„Downloads“/„GPS 1200“/„Brochures“, 2006 (Zugriff: 16. Juli 2007)

INTERSENSE: o.V., Product Manual for use with IntertiaCube 3 and the InertiaCube3 Processor, Bedford MA, 2005

LIFECLIPPER: Lifeclipper, www.lifeclipper.net, Informationsseite über das Projekt Lifeclipper2 (Zugriff: 16. Juli 2007)

LIFECLIPPER2-KONZEPT: Jan Torpus, lifeClipper2-Konzept, aus: http://www.lifeclipper.net/download/LC2_concept_NT.pdf, Basel 2007 (Zugriff: 29. August 2007)

MATHEMATICS FOR COMPUTER GRAPHICS: John Vince, Mathematics for Computer Graphics, London 2006

NÄHERUNGSLÖSUNG FÜR DIREKTE TRANSFORMATION: Bundesamt für Landestopographie (Hrsg.), Näherungslösungen für die direkte Transformation CH1903 – WGS84, aus: http://www.swisstopo.ch/pub/down/basics/geo/system/ch1903_wgs84_de.pdf, Wabern 2005

ORIENTIERUNG VON LASERSCANNER-PUNKTWOLKEN: Michael Hofer und Helmut Pottmann, Orientierung von Laserscanner-Punktwolken, Technische Universität Wien, aus: http://www.geometrie.tuwien.ac.at/geom/ig/papers/tr113.pdf (Zugriff: 16. Juli 2007)

ROLL-PITCH-YAW-WINKEL: o.V., Roll-Pitch-Yaw-Winkel, aus: http://de.wikipedia.org/wiki/Roll-Pitch-Yaw-Winkel, (Zugriff: 23. Juli 2007)

SCHIFFER: Stefan Schiffer, Visuelle Programmierung, aus: http://de.wikipedia.org/wiki/Visuelle_Programmierung, (Zugriff: 20. Juli 2007)

8 Anhang

8. 1 Verwendete Hard- und Software

Laptop: Dell Precision M90 mit einem Intel Core™2 Prozessor T7400 mit 2.16 GHz und 2 GB RAM
Graphikkarte: NVIDIA Quadro FX 2500M

Microsoft Windwos XP Professional

HMD: EMagin Z800 3DVisor, mit zwei SVGA 3D OLED Mikrodisplays mit einer Auflösung von 800x600

Kamera: Zwei QuickCam for Notebooks Pro von Logitech

Tracker: InertiaCube3 von InterSense

GPS: Leica GPS1200 GX1230 GG

Software: Virtools Dev. Version 3.5

8. 2 Näherungsformel[13]

„nach: [H. Dupraz, Transformation approchée de coordonnées WGS84 en coordonnées nationales suisses, IGEO-TOPO, EPFL, 1992]
Die Parameter wurden von U. Marti (Mai 1999) neu berechnet. Zudem wurden die Einheiten so angepasst, dass sie mit den Formeln aus [Bolliger 1967] vergleichbar werden.

1. Breite φ und Länge λ sind in Sexagesimalsekunden ["] umzuwandeln
2. Hilfsgrössen (Breiten- und Längendifferenz gegenüber Bern in der Einheit [10000"]) berechnen:

$$\varphi' = (\varphi - 169028.66'') / 10000$$
$$\lambda' = (\lambda - 26782.5'') / 10000$$

3.

$$y[m] = 600072.37 + 211455.93 \cdot \lambda' - 10938.51 \cdot \lambda' \cdot \varphi' - 0.36 \cdot \lambda' \cdot \varphi'^2 - 44.54 \cdot \lambda'^3$$

$$x[m] = 200147.07 + 308807.95 \cdot \varphi' + 3745.25 \cdot \lambda'^2 + 76.63 \cdot \varphi'^2 - 194.56 \cdot \lambda'^2 \cdot \varphi' + 119.79 \cdot \varphi'^3$$

$$h'[m] = h - 49.55 + 2.73 \cdot \lambda' + 6.94 \cdot \varphi'$$

4. Zahlenbeispiel

$$gegeben: \quad \varphi = 46°2'38.87'' \quad \lambda = 8°43'49.79'' \quad h = 650.60m$$
$$\Rightarrow \quad \varphi' = -0.326979 \quad \lambda' = 0.464729$$
$$\Rightarrow \quad y = 699999.76m \quad x = 99999.97m \quad h' = 600.05m$$

[13] Zitiert aus: NÄHERUNGSLÖSUNG FÜR DIREKTE TRANSFORMATION 2005, S. 1.

Diese Näherungen sind für die ganze Schweiz besser als 1 Meter in der Lage und 0.5 Meter in der Höhe.

Bemerkung zu den Höhen:
In diesen Formeln wird davon ausgegangen, dass mit ellipsoidischen Höhen gearbeitet wird, wie sie z.B. mit GPS-Messungen erhalten werden. Wird mit 'Höhen über Meer' gearbeitet, so sind die Höhen im Meterbereich in beiden Systemen gleich. Sie müssen also in diesem Fall nicht umgerechnet werden.“

8. 3 Anpassung an eine neue Virtools Composition

Hierbei handelt es sich um vorzunehmende Anpassungen an neue Composition Modelle, um die Orientierung und die Kamera einzubauen.

- Modell öffnen
- Bei *Schematic Init*: re Maustaste, *Import Behaviour Graph*, dann „Tracker.nms“ und „Video.nms“ einfügen.
- Beide ans *Init* "anschliessen".

- Box *Tracker* öffnen: BB *Set Quaternion Orientation* doppelklicken, *Target POSITION _ORIENTATION* einstellen.

- Box *GPS Tracker* öffnen: BB *Set Position* doppelklicken, Target *POSITION_ORIENTATION* einstellen.
- Kontrollieren: im *GPS Tracker*, erste *Addition* Box (nach *VRPN Tracker*) muss bei *y* Wert „0“ sein, bei *Ray Intersection*: *Depth* muss Wert „1000“ sein und *Filter* muss *INTERSECTION* sein.

- Box *Video* öffnen: In jeder BB darin (*Video Input Properties*, *Video Input Control*, *Video Player*)Target auf *New Video* setzen.

- *Init* schliessen (oder einfach wieder klein machen).

- Im Hauptmenü oben *Resources* auswählen, *Import File*.
- „Backrop.nmo“ auswählen.
- *Setup* von *Backdrop* öffnen (*LevelManager*: Doppelklick auf *Backdrop*).
  Kontrollieren: *Material*: „New Material.0213“
  Scale: *x* ist „800“ und *y* ist „600“

- *Setup* von New Material.0213 kontrollieren: unter *Material Setup* „New Material.0213“ suchen, kontrollieren: bei *Textur* muss „New Texture.0007“ drin sein.

- *Texture Setup* öffnen: bei „New Texture.0007“ kontrollieren, ob es da ist.

- Im *Level Manager Cameras* auswählen, *POSITION_ORIENTATION* doppelklicken, nun ist man im *Setup*, dort kontrollieren, ob bei *Field of View* wirklich „56.2294“ drinsteht, sonst ändern und mit *SetIC* bestätigen.

- Beim 3d Fenster: links *Create Video* auswählen, drücken, *New Video* ist kreiert: *Type*: „Live“ auswählen.
- *Video ID*: „Logitech“ auswählen (geht nur, wenn an Laptop mit Kamera angeschlossen).
- *Audio*: „SigmaTel Audio“ auswählen.
- *Output Options*: *Type*: „Texture“ auswählen, *Type*: „New Texture.0007“ auswählen.

- *Hierarchy Manager* öffnen (Fenster oben rechts): *Backdrop* in *POSITION_ORIENTATION* reinziehen.

- Bei *3D Sprite Setup*: *Backdrop* auswählen und *Remove IC* (linke Seite) und danach *Set IC*.