



Institut für Rechnerentwurf
und Fehlertoleranz (IRF)
Prof. Dr.-Ing. R. Dillmann
Fakultät für Informatik
Universität Karlsruhe

Spielerische Evaluierung eines Augmented Reality Systems

Studienarbeit
von

Christopher S. Özbek
oezbek@ira.uka.de

15.04.2003 – 15.07.2003

Referent : Prof. Dr.-Ing. Rüdiger Dillmann
Betreuer : Dipl.-Inform. Björn Giesler

Ich versichere hiermit, die vorliegende Studienarbeit selbstständig und ohne fremde Hilfe angefertigt zu haben. Die verwendeten Hilfsmittel und Quellen sind im Literaturverzeichnis vollständig aufgeführt.

Karlsruhe, den 15.07.2003

Christopher S. Özbek

Zusammenfassung

Zum Vergleich eines Video-See-Through (VST) Augmented Reality Systems mit einer Optical-See-Through-Lösung (OST) wurde eine Spielumgebung in Anlehnung an die „Star Wars“-Filme programmiert. Das Spiel wurde in das Augmented Reality Framework PorkiII eingebettet und dieses für Stereo-Sehen erweitert. Eine Studie an 30 Probanden verglich die zwei Systeme in Verbindung mit unterschiedlichen Tracking-Mechanismen und ergab eine Präferenz für die aus optischer Durchsichtbrille und Infrarot-Tracking bestehende Konfiguration.

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
1 Einführung	2
1.1 Aufgaben- und Fragestellung	4
1.2 Gliederung	4
2 Implementierung	5
2.1 Die Hardwareumgebung	5
2.1.1 Video-See-Through	5
2.1.2 Video-Grabbing	6
2.1.3 Optical-See-Through	7
2.1.4 Infrarot-Tracker	8
2.2 Die Programmierumgebung	8
2.2.1 C++	8
2.2.2 Open Inventor	9
2.2.3 GLUT	10
2.2.4 ARToolkit	11
2.2.5 PorkiII	11
2.2.6 Die Settings-Datei	15
2.3 Die Spielumgebung	17
2.3.1 Szenario	17
2.3.2 Drohne	18
2.3.3 Laserschwert	19
2.3.4 Spielverlauf	19

3 Studie	22
3.1 Vorüberlegungen	22
3.2 Testpersonen und Auswahl	23
3.3 Durchführung	24
3.4 Statistik	25
3.5 Ergebnisse und Diskussion	25
3.5.1 Vorwissen	25
3.5.2 Erreichte Punkte	26
3.5.3 Fragebogen	28
3.5.4 Sonstiges	33
4 Diskussion und Schluss	34
4.1 Aktionsplan	34
4.2 Dank	37
A Vorlagen zur Studie	38
A.1 Terminplan	38
A.2 Fragebogen	39
A.3 FAQ	40
B Rohdaten	41

Abbildungsverzeichnis

1.1	Ein Proband beim Spielen. Das Bild ist insoweit gestellt, als dass der Spieler selbst zwar die Szene so sah, der Photograph Laserstrahl und Laserschwert aber nur auf den Monitoren verfolgen konnte.	3
2.1	Verwendete Marker-Typen	13
2.2	Die verschiedene Tracker-Renderer Zuordnungen. Blaue Pfeile stehen für fest vorgegeben Transformationen und rote Pfeile für vom Tracker bestimmte.	15
2.3	Das Entstehen der Drohne: Vom Foto über die perspektivische Textur-extraktion hin zum computergenerierten Abbild	18
2.4	Das Zeigegerät „Zauberstab“ mit aufgeklebten ARToolkit-Markern und retroreflektiven Targets zur Positionsbestimmung. In Kopf des Stabes verbirgt sich ein Bluetooth-Modul und ein Vibrationsmotor.	20
2.5	Übersicht über die Anzeige während des Spiels	21
3.1	Durchschnittliche Punktezahlen der Spieler aus Gruppe V (rot) und Gruppe O (blau) in Abhängigkeit zur Nummer des Durchlaufes. Dreiecke sind Spiele am optischen, Vierecke solche am Video-System. Beim vierten Spiel wechselten die Gruppen die Geräte („Systemwechsel“). . .	27
3.2	Übersicht über die Durchschnittswerte der Antworten für Fragen 5-14 in rot beim Video- und blau beim optischen-System. Es lässt sich gut die Präferenz für das optische System erkennen.	28

Kapitel 1

Einführung

Augmented Reality (AR), als Technik zur Verknüpfung von computergenerierten Sinneseindrücken mit unseren herkömmlichen Wahrnehmungen, steckt noch in den Kinderschuhen [2]. So sehr unsere immer komplizierter werdende Welt es nötig macht, Informationen auf einfache und intuitive Weise in unsere Sinne zu integrieren, so sehr hinkt die Realisierung solcher Systeme den Anforderungen hinterher.

Noch gibt es kein Kochbuch, das angenehm über dem Herd zu schweben scheint und von selbst umblättert, wenn sich unser Blick dem Ende der Seite nähert. Noch gibt es keine Anleitung, die den Knopf am Videorecorder hervorhebt, mit dem wir die Aufnahme starten können. Noch müssen wir auf den Computerbildschirm starren, wenn wir Informationen über etwas erhalten wollen, was eigentlich direkt vor uns liegt.

Augmented Reality ist angetreten, um diese Kluft zu überwinden. Die Wirklichkeit wird hierzu um Sinneseindrücke ergänzt - man kann sagen „augmentiert“, die rein virtuellen Ursprungs sind und im Computer entstanden.

Was der Fantasie jedoch Ketten anlegt, ist die Technologie, welche diese Sinneseindrücke erzeugen und dem Anwender übermitteln muss. Die Freude, keine Straßenkarte zu brauchen, weil der richtige Weg im Blickfeld des Anwenders durch den Computer vorgezeichnet wird, ist nur von kurzer Dauer, wenn die Daten-Brille nach 5 Minuten unsere Augen ermüdet hat.

Versucht wurde, die momentanen Möglichkeiten der in unserem Institut verfügbaren Ressourcen im Bereich Augmented Reality, zu evaluieren und hierbei den Probanden und Lesern darzulegen, welche Kombinationen aus Realität und Virtualität schon jetzt in greifbarer Reichweite sind.

Dazu wurde eine Testumgebung im Augmented-Reality-Framework PorkiII realisiert,

welche eine Szene aus dem ersten „Star Wars“-Film zur Grundlage hat. Der Spieler wird hierbei zum Laserschwert schwingenden Jedi-Ritter, welcher seine spielerischen Fähigkeiten nicht mehr, wie bei herkömmlichen Computerspielen, nur über die Maus in die virtuelle Welt überträgt, statt dessen steht er selbst direkt in ihr. Sie wird für ihn mit der realen Welt verschmolzen.

In dieser Arbeit werden, die vorangehenden Tätigkeiten, die sich um Eingabegeräte, Framework, Kalibrierung und Tracking kümmern, integriert und eine lauffähige Anwendung mit Stereosicht erstellt. Ich hoffe, dass hier der Auftakt einer ganzen Reihe von konkreten Anwendungen des Augmented-Reality-Frameworks gesetzt werden konnte und die Evaluation genug Arbeitsfelder und Probleme für die nächsten Runden an studentischem Eifer liefert.



Abb. 1.1: Ein Proband beim Spielen. Das Bild ist insoweit gestellt, als dass der Spieler selbst zwar die Szene so sah, der Photograph Laserstrahl und Laserschwert aber nur auf den Monitoren verfolgen konnte.

1.1 Aufgaben- und Fragestellung

Es galt ein „Augmented Reality“-Computerspiel zu programmieren und, anhand einer Untersuchung an Probanden, deren Spiel-Leistung und Meinung jeweils zu einem Video-See-Through-System und einem Optical-See-Through-System zu erheben und hiermit die beiden Systeme zu bewerten.

1.2 Gliederung

Beschrieben wird im ersten Teil der Arbeit die Implementierung des Spiels und die Integration in das Augmented-Reality-Framework. Dieser Teil ist gegliedert nach den verwendeten Hardware- und Softwarekomponenten. Er beschäftigt sich individuell mit diesen Bestandteilen und beschreibt die mit ihnen verbundenen Probleme und Verbesserungsvorschläge. Hauptaufgabe dieses Kapitels ist es, das während der Arbeit gesammelte Wissen kompakt und übersichtlich für nachfolgende Studenten zum Nachschlagen darzustellen. Daher werden einzelne wichtige Gesichtspunkte und Folgerungen der Methodik bereits in den zugehörigen Abschnitten diskutiert. Der zweite Teil legt sein Augenmerk auf die Studie, ihre Durchführung und Auswertung. Im Schlussteil werden die gesammelten Probleme und Hinweise noch einmal übersichtlich priorisiert, sowie die Hauptideen zusammengefasst. Die verwendeten Vorlagen für z.B. Fragebögen und die Rohdaten der Studie können im Anhang eingesehen werden.

Kapitel 2

Implementierung

Im folgenden Kapitel werden die einzelnen Hard- und Softwarekomponenten vorgestellt, die bei der Implementierung des LaserTrainings verwendet wurden.

2.1 Die Hardwareumgebung

In diesem Abschnitt werden die verwendeten Geräte und vorgenommenen Einstellungen beschrieben.

2.1.1 Video-See-Through

Zum Einsatz kam die Video-See-Through-Brille „ARVision-3D HMD“ der Firma Trivisio. Sie besteht aus je zwei 6 mm interlaced 640 * 480-Pixel Video-Kameras und zwei 800 * 600-Pixel LCD Bildschirmen in einer kompakten, auf dem Kopf tragbaren Einheit. Die Kamerabilder werden um AR-Informationen erweitert und über zwei Bildschirmausgänge einer Radeon 9500 der Firma ATI an die Brille zurückgeführt. Der Versuch, eine G450 der Firma Matrox ebenfalls in den benötigten Modus eines DRI-beschleunigten 3D-X-Servers auf beiden Bildschirmen zu versetzen, scheiterte an den angebotenen Treibern, welche nur den primären Bildschirm beschleunigen.

Die *Xinerama*-Erweiterung unterstützt normalerweise den Desktop-Manager (KDE, Gnome) bei der Verwaltung von Dual-Head-Bildschirmkonfigurationen. Hierdurch ergeben sich höhere Benutzerfreundlichkeit im Hinblick auf Fenstererstellung, -maximierung und -positionierung. Beide Karten unterstützen zwar die Erweiterung, sind aber dann generell unbeschleunigt, was die Verwendung in diesem Projekt unmöglich mach-

te.

Auch die Umstellung auf Kantenglättung mit mehreren Renderläufen misslang. Dies wäre besonders im Hinblick auf die niedrigen Auflösungen der Brillen unterhalb von XVGA sehr nützlich gewesen, da es dem Benutzer eine feinere Darstellung durch weichere Kanten suggeriert. Neue Treiber könnten hier Abhilfe schaffen [6].

Das Interlaced-Bild der Kameras erreicht zufriedenstellende Frameraten von 25 fps. Die Displaybilder können in ihrer Farbstärke und Schärfe überzeugen, das Sichtfeld ist allerdings sehr klein und erfordert unangenehmes Zurechtrücken.

Ungelöst bleibt das Problem der Verkabelung. Obwohl die AR-Brille an zwei handliche Akku- und Verarbeitungsstufen angeschlossen ist, welche autark jeweils eine der Kameras und Displays mit Strom versorgen, sind für die Übertragung des Video- und Bildsignals noch insgesamt vier Kabel nötig. Dies schränkt die Bewegungsfreiheit stark ein, zumal es für mobile Computer keine Lösungen mit zwei Monitorausgängen gibt und man damit an einen ungleich schwereren Desktop-Computer und eine passende Stromversorgung gebunden ist. Die zu übertragende Datenmenge von den beiden Kameras bei $640 * 480$, 20 Frames und 3 Byte Farbinformation pro Pixel, sowie dem gleichen Informationsstrom mit $800 * 600$ Pixeln zurück, entspricht ≈ 750 MBit/s. Dies bleibt selbst mit Kompression wohl noch auf längere Zeit außerhalb der per Wireless-LAN erreichbaren Größenordnung (11 MBit/s). USB 2.0 (480 MBit/s) oder auch Firewire (400 MBit/s) könnten bei entsprechender Komprimierung oder Interlacing jedoch den Datenstrom bewältigen.

Eines der größten praktischen Probleme, die im Betrieb der Brille auftraten, war die Parallaxverzerrung, da das Spiel einen ständigen Wechsel der Fokulentfernung erfordert. Die starr angebrachten Kameras können den Fokusbewegungen der Augen nicht folgen. Dadurch ist es für den Benutzer nicht möglich das linke und rechte Bild in Kongruenz zu bringen und einen Tiefeneindruck zu gewinnen. Ob Ansätze wie in [20] wirklich eine Verbesserung in dieser Hinsicht bringen, bleibt abzuwarten.

2.1.2 Video-Grabbing

Die Videosignale wurden von zwei Pinnacle-TV-Karten aufgenommen und über den Video4Linux-Treiber Version 1 [10] von der Hardware abstrahiert. Im Rahmen einer notwendigen Performance-Evaluation wurden die parallel den Bus belastenden PCI-Karten als Flaschenhals ermittelt. Auf dem eingesetzten 1-GHz Pentium-III-System ließ sich die maximale Leistung erreichen, indem in beiden vom Treiber angebotenen Capture-Puffern mittels `ioctl` parallel Bilder aufgenommen wurden. Bei jedem neuen Durchlauf der Hauptschleife wurde mit dem älteren der beiden Aufrufe mittels

`ioctl(VIDIOSYNC...)` synchronisiert, der Pufferinhalt in einen weiteren Speicherbereich kopiert und unmittelbar ein weiterer Capture-Vorgang gestartet, so dass die Auslastung der zwei Capture-Puffer optimiert werden konnte. Dieser Ansatz entspricht nicht der Vorgabe aus der API-Dokumentation, brachte jedoch eine 50 prozentige Erhöhung der Framerate mit sich (Alg. 2.1). In wieweit dieser Ansatz eine höhere Verzögerung der Bilder (Lag) mit sich bringt, wurde nicht evaluiert. Eine genauere Analyse der beschränkenden Faktoren und ihres Zusammenwirkens sollte unbedingt in späteren Arbeiten erfolgen.

Der Einsatz einer Nebenläufigkeit zur Performance-Steigerung brachte nicht den erwünschten Effekt und wurde verworfen. Da das Tracking beim vorliegenden Stand des Systems aus dem Videobild selbst ermittelt wird, macht eine Entkopplung zwischen Video-Grabbing und dargestellter 3D-Sicht wenig Sinn. Dies wäre erst dann interessant, wenn das Tracking verteilt auf verschiedenen Rechner lief. Hier könnte die 3D-Sicht durch externe Daten aktualisiert werden, obwohl das Frame, welches die jeweilige Änderung in der Welt reflektiert, noch nicht `captured` worden ist.

<pre> 1: /* Setup */ 2: VIDIOCMCAPTURE(0) 3: while true do 4: VIDIOCMCAPTURE(1) 5: VIDIOCSYNC(0) 6: /* Verarbeite Frame 0 während die Hard- hardware Frame 1 captured */ 7: VIDIOCMCAPTURE(0) 8: VIDIOCSYNC(1) 9: /* Verarbeite Frame 1 während die Hard- hardware Frame 0 captured */ 10: end while </pre>	<pre> 1: /* Setup */ 2: VIDIOCMCAPTURE(0) 3: VIDIOCMCAPTURE(1) 4: i ← 0 5: while true do 6: VIDIOCSYNC(i) 7: memBuffer ← videoBuffer(i) 8: VIDIOCMCAPTURE(i) 9: i ← 1 - i 10: /* Verarbeite memBuffer während Frame 0 und 1 aufgezeichnet werden */ 11: end while </pre>
---	---

Alg. 2.1: Bildaufnahme in zwei Bildpuffer: Links der Vorschlag der API-Dokumentation[10] und rechts die beschleunigte Variante

2.1.3 Optical-See-Through

Das verwendete optische Durchsichtssystem ist das von der Firma Sony hergestellte Glasstron PLM-700E, welches mittlerweile nicht mehr produziert wird, jedoch eine weite Verbreitung in der Augmented Reality Forschung besitzt. Die mitgelieferte Kopfhalterung wurde durch eine in einer vorangehenden Arbeit [8] entstandene Version ersetzt. Angesteuert wurde die Brille über den stereoskopischen Shuttermode bei 800*600 bei 85 Hz, der je ein Bild auf das rechte und auf das linke Auge umleitet. Die benötigte

Quad-Buffer-Extension der OpenGL-Implementierung stand mit einer „NVIDIA Quadro 4“-Karte zur Verfügung.

Die Länge der Kabel ist wie auch beim Video-System sehr einschränkend, es entsteht aber nicht der gleiche Kabelwust, da die Sony Brille durch den Shuttermode nur einen Monitoreingang und eine Stromversorgung benötigt. Die optische Durchsicht durch das halbtransparente Display bei hellster Einstellung ergibt eine gewisse Verdunklung der Realität.

2.1.4 Infrarot-Tracker

Als Hardwarelösung kam das von der Firma NDI hergestellte Polaris-Trackingsystem zum Einsatz. Es verwendet Infrarotlicht zum stereoskopischen Verfolgen von kleinen retroreflektiven Kügelchen. Sein Wirkungsfeld ist auf einen ca $(2m)^3$ großen Würfel beschränkt. Das für die Medizin entworfene System besitzt eine hohe translatorische Genauigkeit und erkennt handgefertigte Markerkonfigurationen.

2.2 Die Programmierumgebung

Für die Entwicklung der Lasertrainings musste auf eine Vielzahl von bestehenden Softwarepaketen und Entwicklungswerkzeugen aufgesetzt werden, welche in den folgenden Abschnitten vorgestellt werden.

2.2.1 C++

Die Wahl von C++ [19] zur Programmiersprache war unumgänglich, da die bisherigen Teile des Frameworks PorkiII und alle gebräuchlichen Grafikpakete in dieser Sprache geschrieben sind. Auch schränken die hohen Anforderungen an Reaktionszeiten und Durchsatz die Auswahl der verfügbaren Sprachen stark ein. C++ bringt keine eleganten Möglichkeiten für verteilte Funktionsaufrufe (RPC) mit und auch Reflektion gehört nicht zur starken Seite der Sprache. Wenn man von Java absieht, welche selbst eine Reihe an neuen Komplikationen mit sich bringt, gibt es aber wohl keine ernsthafte Alternative.

2.2.2 Open Inventor

Open Inventor [18] von der Firma SGI als Szenengraphensprache zur Darstellung und Animation von 3D-Objekten auf hohem Abstraktionsniveau wurde wegen des direkten Aufsatzes auf OpenGL [17] gewählt. So lässt Open Inventor trotz des Komforts nicht den Low-Level-Zugriff vermissen lässt, der durch den ungewöhnlichen Systemaufbau mit zwei Bildschirmen und Framegrabbern notwendig wurde. Open Inventor ist jedoch mittlerweile mehr als zehn Jahre alt und man merkt vor allem der Version von SGI im Vergleich zu den kommerziellen Weiterentwicklungen von TGS dieses Alter an. Es traten in vieler Hinsicht Probleme mit Open Inventor auf, welche die Evaluation von Alternativen zu Open Inventor für sinnvoll erscheinen lassen.

Licht Wohl durch die Verwendung von GLUT in Verbindung mit Open Inventor wurden Lichtquellen von nachfolgenden Transformationen im Inventor-Scenegraph mitgedreht. Es war also nicht möglich, ein Objekt von einer anderen Seite anzuleuchten als initial für das Licht als Richtung angegeben wurde. Dies entspricht nicht der sonstigen Wirkung von Transformationen im Scenegraph. Da der Effekt in den Open Inventor Beispielapplikationen nicht auftritt, liegt der Schluss nahe, dass Lichter in Inventor anders gehandhabt werden als sonstige Geometrielemente und dies, in Verbindung mit der GLUT Version des AR-Toolkits, nicht berücksichtigt wurde.

Caching Bei der Implementierung des Render-Moduls für die Video-Brille, welches zwei unabhängige Fenster erzeugen muss, die getrennt auf das linke und das rechte Auge projiziert werden können, treten mit Open Inventor Caching-Probleme auf. Open Inventor legt bei der Traversierung des Scenegraphs einen Cache an, welcher die aufwendigen Transformationen von hochsprachigen Szenelementen in grafikartenkompatible Primitive durchführt und diese in OpenGL-DisplayLists speichert. Da der Scenegraph jetzt aber in zwei verschiedenen OpenGL-Renderkontexten - einen für das jeweilige Fenster - dargestellt werden muss, gilt es Open Inventor darauf hinzuweisen, mehrere Caches anzulegen. Dies gelingt mittels Aufruf von `SoGLRenderAction::setCacheContext(int)`. Performance-Tests ergaben jedoch, dass hier noch ein entscheidender Flaschenhals vorliegt. Profiling und Tests anderer Herangehensweisen, z.B. durch Verwendung eines einzelnen GL-Kontextes, welcher in zwei separate Viewports aufgeteilt wird, könnten die Anwendung beschleunigen.

Video-Capturing PorkiII beinhaltet die Möglichkeit den gesamten Bildschirmbereich pro Frame auf die Festplatte zu schreiben, um anschließend daraus Präsentati-

onsvideos und dergleichen zu erstellen. Leider bietet OpenGL selbst keinen direkten Zugriff auf den Framebuffer an und Aufrufe mittels `glReadPixels` bewirkten einen dramatischen Einbruch der Performance auf 10 % der Ausgangsframerate. Durch Reduzierung der Fenstergröße ließen sich dann wieder knapp fünf Frames erreichen. Weitere Arbeiten am Framework sollten hierfür eine akzeptable Lösung finden, beispielsweise ein Videocapturing auf einem anderen Rechner.

Font-Rendering Eine großer Teil der Verbesserung der visuellen Qualität heutiger Desktop-Umgebungen beruht auf der Verwendung von geglätteten Schriftarten. Leider bleibt diese Weiterentwicklung den Inventor-Anwendern vorerst versperrt. Versuche in die Richtung scheiterten stets, so dass am Ende doch die Rückkehr zu der Standard-X-Fontbibliothek stand. Eine Verbesserung dieser Situation hätte ähnliche Auswirkungen wie das bereits angesprochene Fullscreen-Antialiasing (S. 6). Stellt man das Font-Rendering auf 3D-Primitive um, wie dies durch Verwendung der Open Inventor-Klasse `SoText3` geschehen kann, so könnte man mit Kantenglättung beide Probleme auf einmal lösen.

2.2.3 GLUT

Das *OpenGL Utility Toolkit* [12] diene als plattformunabhängige Softwareschicht zwischen Fenstermanager und OpenGL-Anwendungsschicht. Die Einfachheit, gute Portabilität und bestehende Integration in das ARToolkit ließen GLUT als erste Wahl erscheinen, die sich erst im Nachhinein als problematisch erwies. Gravierenderweise steckt ein entscheidender Designfehler im Callback-Mechanismus von GLUT, welcher die Übergabe eines vom Programmierer übergebenen Identifikationstokens vermissen lässt. Dieser üblicherweise als „*user data*“ bezeichnete Parameter dient der Wiedererlangung von Kontextinformationen nach Aufruf der statischen C-Funktion. Eine effiziente, objektorientierte Kapselung bleibt damit untersagt. Obwohl ein Work-Around gelang, bleibt die Lösung unbefriedigend und es ist anzuraten GLUT bei Gelegenheit durch eine modernere Fassung zu ersetzen, welche graphische Benutzerschnittflächen, Dialogmöglichkeiten und deren komplette Abschaltung für den reinen Bildschirmbetrieb im Augmented-Reality-Modus mitbringen sollte. Zu evaluieren wären die Inventor-kompatiblen So-Bibliotheken wie `SoQT` oder `SoXT`.

2.2.4 ARToolkit

Das ARToolkit von Mark Billinghurst und Hirokazu Kato [3] wird zur Positionsbestimmung aus Kamerabildern verwendet. Bei gleichzeitiger Verwendung mehrerer Marker kommt eine verbesserte Kalibrierung von Jan Minar zum Einsatz [15], welche in PorkiII als Plugin `PARToolkitTracker` integriert ist. Leider ist das ARToolkit in C geschrieben und nicht auf die Anforderungen für den Dual-Head-Betrieb ausgelegt. Dieser bedauerliche Effekt entsteht durch die Verwendung von globalen und `static`-Variablen innerhalb der einzelnen Routinen. Die Entwicklung des „DualWindowRenderer“-Plugins für die Anzeige der Szene in zwei getrennten Fenstern erforderte den Umstieg auf C++. So konnten über den objektorientierten Kontext Informationen bzgl. des verwendeten Fensters gespeichert werden. Ein Umschalten zwischen den Fenstern kann jetzt mittels Aufruf von `glutSetWindow` der Klasse `ARG` erfolgen.

Während der Studienarbeit wurde die aufwendige Konvertierung nur in den OpenGL und GLUT betreffenden, „gsub/arg“-genannten Codeteilen vorgenommen, da die entsprechende Anpassung des ganzen Toolkits den Rahmen der Studienarbeit gesprengt hätte. Eine solche Umstellung bleibt jedoch letztlich unumgänglich, da auch der Einsatz mehrerer Tracker von der jetzigen Implementierung versperrt bleibt. Nur eines der beiden Kamerabilder kann daher momentan für die Positionsbestimmung mit Hilfe des ARToolkits herangezogen werden. Wie bereits in den zwei vorangehenden Abschnitten erwähnt, wäre die vollständige Trennung von GLUT sehr wünschenswert. Der mit 100 Stunden von mir eingeschätzte Arbeitsaufwand würde sich hauptsächlich auf die Portierung der ARToolkit-Kalibrierungs- und Entzerrungsroutinen verteilen.

Das ARToolkit verbraucht einen beträchtlichen Teil der Prozessorressourcen, so dass ein schneller Prozessor (≥ 1 GHz) dringend empfohlen wird.

2.2.5 PorkiII

Das Laserschwert-Training wurde als „Welt“ im AR-Framework PorkiII [7] erstellt. PorkiII ist für die Verwaltung der als Plugins realisierten Systemkomponenten zuständig. Werden solche über die Settings-Datei spezifiziert, so initialisiert und lädt das Framework die entsprechende Shared-Library. Dieser Plugin-Mechanismus vereinfacht v.a. den Kompiliervorgang, da bei Änderungen nur einzelne Plugins neu kompiliert werden müssen und nicht die ganze Anwendung. Ein gravierender Nachteil wurde sichtbar, als ein Performance-Engpass auftrat: Profiling über Shared-Libraries ist unter x86-Linux nicht möglich. Das GNU-Tool `gprof` [11] kann nur sinnvoll eingesetzt werden, wenn das ganze Projekt in eine einzige, statisch gelinkte Anwendung kompiliert wird. Die

den `bgtools` [9] entstammenden Makefiles bieten diese Option momentan nicht.

Die verschiedenen Plugintypen:

- **Renderer** sind Ausgabemodule, welche eine Welt und ein zugehöriges Menü visualisieren. Unterschieden werden kann zwischen verwendeter Fensterschnittstelle und angezeigtem Stereo-Modus:

ATKSingleRenderer Basierend auf der `arg`-Bibliothek des ARToolkits und deshalb mit GLUT als Fensterschnittstelle, rendert dieses Plugin nur in ein Fenster ohne Stereo.

ATKRedGreenRenderer Dieses Anzeigemodul erweitert den `ATKSingleRenderer` um einen Zwei-Farben-Stereo-Modus, bei welchem die Szene zweimal mit perspektivischem Offset auf verschiedene Farbkanäle gerendert wird. Der Benutzer kann durch Tragen einer entsprechend gefärbten 3D-Brille einen Stereo-Eindruck gewinnen. Dieses Modul wurde in der Anfangsphase der Entwicklung verwendet.

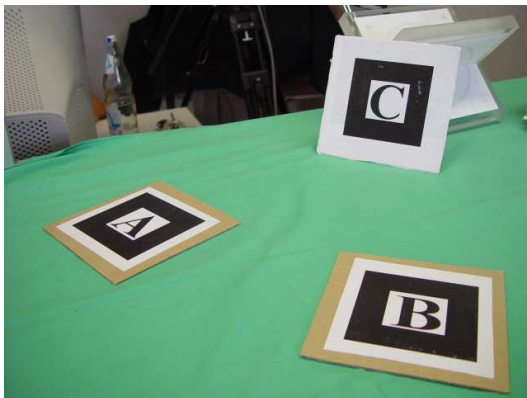
ATKDualWindowRenderer Dieser Renderer wurde für die Video-See-Through-Brille im Rahmen der Studienarbeit komplett neu erstellt. Durch Einsatz der auf C++ portierten `arg`-Bibliothek werden zwei separate Fenster geöffnet, in welchen die Szene aus zwei versetzten Blickwinkeln gerendert werden kann. Wird jeweils eins der zwei Fenster auf einen anderen Monitor einer Dual-Bildschirm-Konfiguration verschoben, so kann der Benutzer die zwei Monitor-signale auf das jeweilige LCD-Display in seiner 3D-Brille gespeist bekommen. Die in den Abschnitten 2.2.2 und 2.2.3 erläuterten Probleme gelten leider.

OpenInventorRenderer Da das Polaris-Tracking-System in Verbindung mit der Sony Glasstron-Brille ohne ein Video-Signal auskommt, konnte hierfür ein separates, rein auf Inventor basierendes Rendermodul im Rahmen der Studienarbeit geschrieben werden. Der Code wurde hierdurch signifikant einfacher und es besteht weniger Gefahr, dass unerwünschte Nebeneffekte, wie die in Abschnitt 2.2.2 beschriebene Lichtabnormalität, auftreten.

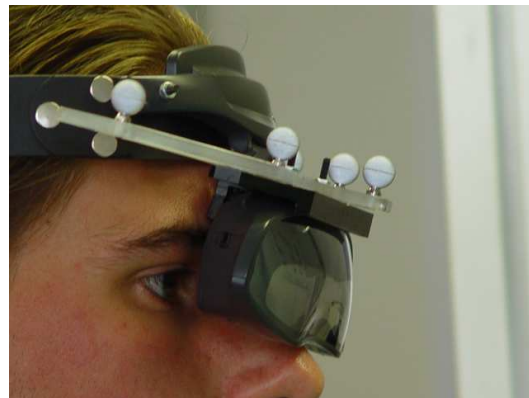
Weitere Renderer-Module werden wohl unausweichlich folgen müssen, zumal eine Trennung vom AR-Toolkit in diesem Bereich und eine damit verbundene Konversion der ATK-Renderer dringend anzuraten ist. Ein bestehendes Problem bleiben die unterschiedlichen Koordinatenformate der verschiedenen Renderer. Eine

Festlegung auf ein einheitliches Koordinatensystem oder eine komfortable Koordinatentransformation als Zwischenfilter sollte in einer der nächsten Versionen von PorkiII folgen. In der Settings-Datei verankert, wäre dies sicher eine elegante Lösung.

- **Cameras** kapseln die Schnittstelle zu Bildaufnahmegegeräten wie z.B. Webcams, Framegrabbern und in 3D-Brillen eingebauten Kameras. Das einzige bisher existierende Modul dient als Kapselung zu Video4Linux-Geräten. Die im Rahmen der Studienarbeit vorgenommenen Änderungen sind in Abschnitt 2.1.2 (S. 6) beschrieben.
- **Tracker** sind positionsbestimmende Geräte oder Bibliotheken und versorgen die Anwendung und die Renderer mit 3D-Koordinaten von als „TrackedFeatures“ bezeichneten Merkmalen. So kann der PARToolkitTracker aus einem Kamerabild die Positionen von ARToolkit-Markern (Abb. 2.1(a)) bestimmen, und der PPolarisTracker erkennt Marker-Konfigurationen aus reflektierenden Silberkugeln in seinem Wirkungsfeld (Abb. 2.1(b)). Das PPolarisTracker-Modul entstand parallel zur Studienarbeit als Portierungsarbeit von Johannes Ahlmann. Das Problem der Koordinatensysteme, wie bereits bei den Renderern erwähnt, gilt natürlich auch für die Tracker und sollte in einem Schritt gelöst werden.



(a) Drei ARToolkit-Marken



(b) Eine Polaris-Marker-Konfiguration befestigt an der Glasstron-Brille

Abb. 2.1: Verwendete Marker-Typen

Im Zuge der Ausdehnung der Tracking-Optionen, welche ursprünglich nur die Option „ARToolkit“ erlaubten, wurde es nötig, die fest in den Code einprogrammierte Verknüpfung von Tracker und Renderer-Position aufzubrechen:

- **fix** (Abb. 2.2(a)): Eine „fixe“ Zuordnung von Tracker (B) und Renderer (C) entspricht dem vor dieser Studienarbeit verwendeten Ansatz, bei welchem die virtuelle Szene aus der gleichen Position aufgenommen wird, an der sich auch der positionsbestimmende Tracker in der Welt (A) befindet. Dieser Ansatz entspricht dem ARToolkit, das aus dem gleichen Kamerabild seine 3D-Position bestimmt, das dem Benutzer auch als Sicht der Welt präsentiert wird.
- **generic** (Abb. 2.2(b)): Bei diesem Zuordnungstyp wird die Position des Renderers aus einer Menge von Tracker-Merkmalen bestimmt. Dies entspricht dem Vorgehen des fest im Raum (A) installierten Polaristrackers (B), welcher die Position des Renderers (D) durch eine bestimmte Marker-Konfiguration (C) ermittelt. Die Szene wird also für den Benutzer hier nicht wie bei **fix** aus der Position des Trackers gezeichnet.
- **untracked** (Abb. 2.2(c)): Dieser dritte Zuordnungstyp setzt den Renderer (B) ohne Anbindung an einen Tracker in die Welt (A). Hiermit könnte z.B. der Szenengraph zusätzlich aus einer festen Übersichtsposition gefilmt werden, oder ein virtueller Durchflug durch die Welt ohne Tracking durchgeführt werden.

Welcher Zuordnungsmechanismus zum Zuge kommt und wie die vorgegebenen Transformationen aussehen, wird in der Settings-Datei (S. 15) als Parameter eingestellt. Der Vorteil dieser Zuordnungsart liegt in ihrer Variabilität, die auch neue Tracking-Mechanismen leicht in das Framework integrierbar machen sollte. Es werden auch „unsinnige“ Kombinationen möglich: So kann man aus der Videokamera der Brille die Positions des Zauberstabs mit dem ARToolkit bestimmen, die virtuelle Szene dem Benutzer aber aus dem Blickwinkel des Zauberstabs präsentieren. Der Benutzer kann damit in der virtuellen Welt mit dem Zauberstab „um die Ecke“ sehen. Eine Demonstration für diese Kombinationsmöglichkeit findet sich voreingestellt in der Settings-Datei unter der Bezeichnung **GenericDemo**.

Die Namenswahl für die Zuordnung zwischen Markern und Renderern ist leider unglücklich in Analogie zu einem weiteren Mechanismus innerhalb des Frameworks erfolgt. Da **generic** allerdings dort nur eine Provisorium sein sollte, muss man jetzt auch hier mit einer baldigen Änderung rechnen.

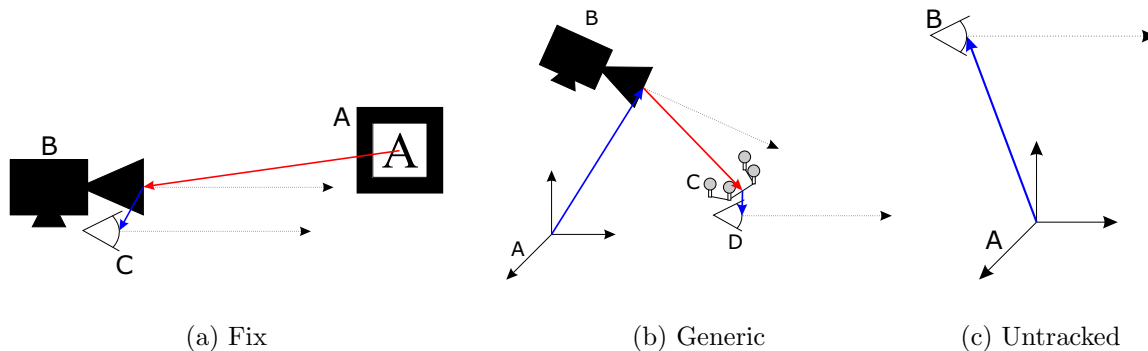


Abb. 2.2: Die verschiedene Tracker-Renderer Zuordnungen. Blaue Pfeile stehen für fest vorgegeben Transformationen und rote Pfeile für vom Tracker bestimmte.

- **Welten.** Welten sind ebenfalls als Plugins realisiert. Eine Welt bietet Schnittstellen, um zugehörige Welt- und OnScreenDisplay-Szenegraphen zu erhalten, sie über Tastatur-Ereignisse zu informieren und zur Aktualisierung ihres Zustandes aufzufordern. Im Rahmen der Studienarbeit entstanden die kanonische Nullwelt, welche nichts tut, die Cube-Welt, welche grundlegende Mechanismen wie Szenegraph für Welt und OnScreenDisplay vorstellt und die LaserTraining-Welt. Ein Mechanismus zum Weltenwechsel innerhalb eines Applikationslaufes wird bald verfügbar sein.

2.2.6 Die Settings-Datei

PorkiIII verwendet zur Konfiguration eine komfortable Einstellungsdatei, welche dem Entwickler eine Vielzahl von Neukompilierungen erspart. Die genaue Syntax kann der Datei selbst entnommen werden. An dieser Stelle werden nur die während der Studienarbeit entstandenen Teile beschrieben.

Includes Durch das Anwachsen der Anzahl an Welten und Plugins über den direkt überschaubaren Bereich wurde es notwendig, die Settings-Datei in geeigneter Weise zu modularisieren. Der hierfür eingebrachte Mechanismus erweitert die Klasse `PUserSettings` um die Möglichkeit, Unterkategorien zu definieren und diese in bestehende Abschnitte zu integrieren. Die erstellte Implementierung liest hierfür zuerst die gesamte Settingsdatei ein, ermöglicht hiermit auch Includes von erst später definierten

Kategorien und expandiert dann rekursiv alle Kategorien um die von ihnen jeweils eingeladenen Abschnitte. Um Selbstrekursion oder andere Zyklen zu verhindern wird vor dem Rekursionsschritt die Include-Anweisung aus der Kategorie selbst entfernt. Dies kann zu Abhängigkeiten des Ergebnisses von der Reihenfolge der Definition führen. Es kann deshalb nicht dazu geraten werden Zyklen zu verwenden.

Die Syntax zur Definition einer Kategorie bleibt unverändert:

```
[KategorieName]
Key = Value
```

Zum Einfügen einer Kategorie mit Namen *KategorieName* dient der Schlüssel **Include**:

```
[Einfügende Kategorie]
Include = KategorieName
Key = Value
```

Besteht in der einfügenden Kategorie bereits ein Schlüssel mit gleichem Namen, wie in dem eingefügten Abschnitt, so werden die beiden Werte mit Komma getrennt konkateniert. Da die Datei zuerst in Schlüssel-Wert-Paare zerlegt wird, stehen die Werte der Kategorie selbst immer am Anfang der kommaseparierten Liste. Eine Erweiterung um ersetzende Includes ist einfach realisierbar und sollte wie auch das Einfügen von externen Dateien ohne Mühe in der nächsten Version von PorkiII verfügbar sein.

Kategorien-Nomenklatur Mit Einführung des Include-Schlüsselwortes entstanden schnell eine ganze Anzahl von neuen Kategorien, welche einer einheitlichen Nomenklatur bedurften. Hier eine Übersicht über die Namen, die zum Projektende verwendet wurden:

- **Setup** Ein Setup stellt eine Konfiguration von Parametern da, der nur noch eine Welt zugeordnet werden muss, um lauffähig zu sein.
- **Demo** Eine Demonstration ist eine vollständige Konfiguration, welche den Benutzer sofort in die Lage versetzt etwas zu sehen. Sie sollte niemals aufwändige Konfigurationen, Kalibrationen oder dergleichen nach dem Start erfordern.
- **World** Eine Welt lädt momentan ein Welt-Plugin und startet dieses dann. Es sollte in Zukunft jedoch möglich sein, dass die Welt sich selbst in diesem Abschnitt konfiguriert und eine vom Setup bedingte, aber nicht festgelegte Menge an TrackedFeatures lädt und sich zuordnet. Diese wären dann bei einer Weltumschaltung ihr allein zugeordnet und nicht für alle Welten sichtbar.

Plugin-Schlüsselwort Zu dem bereits existierenden Schlüsselwort `Plugins`, für das Einladen von Plugins aus einem bestimmten Verzeichnispfad, wurde eine Variante für das gezielte Einladen von einzelnen Plugins erstellt. Dies wurde nötig, da die Initialisierung von manchen Welten nur in bestimmten Konfigurationen von Rechnern und laufenden Servern möglich ist und ein Start von PorkiII ohne eben diese nicht möglich ist. Um dem Laden des gesamten Welt-Plugin-Verzeichnisses aus dem Wege zu gehen, entstand das neue Schlüsselwort. `Plugin` stellt somit nur eine vorläufigen Workaround für eine nötige Änderung in der Plugin-Struktur da, welche Welten erst dann instanziiert, wenn sie auch wirklich benötigt werden, wie dies bei allen anderen Plugins auch der Fall ist. Wenn es gelingen sollte, Segmentation-Faults in Shared-Libraries abzufangen, könnte das Schlüsselwort ebenfalls wieder entfernt werden.

Mathematische Ausdrücke Obwohl die Möglichkeiten der Matrixeingabe um einen Skalierungsfaktor erweitert wurden, gehören mathematische Ausdrücke allgemein momentan noch auf die Wunschliste für Erweiterungen der Settings-Datei. Die Anforderungen an ein solches System lassen sich so umreißen:

- Mathematische Konstanten sollten im gebräuchlichen Umfang vorhanden sein, dies gilt v.a. für die Konstante des trigonometrischen Alltagsgebrauchs π . Benutzerdefinierte Konstanten sollen einfach selbst definiert und verwendet werden können.
- Termauswertung in sinnvollem Umfang mit den Grundrechenarten und Potenzen.
- Wählbare Eingabe von Winkelgraden in Grad oder Radiant.

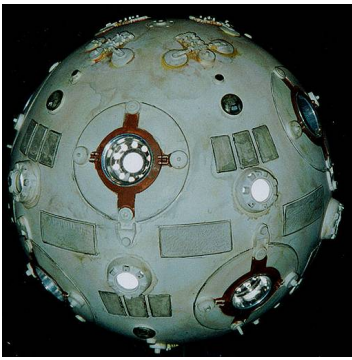
2.3 Die Spielumgebung

2.3.1 Szenario

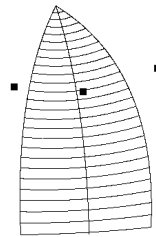
Eine kugelförmige Drohne schwebt innerhalb des Sichtfeldes des Spielers und feuert hierbei in unregelmäßigen Abständen Laser-Strahlen auf diesen ab. Die Aufgabe des Spielers ist es, mit einem „Laser-Schwert“ die Strahlen abzuwehren und am Vorbeiflug zu hindern. Das Spiel wurde aus dem Science-Fiction Film „Star Wars“ [13] entnommen und ist dort Teil eines Testes auf Reaktionsfähigkeit und Konzentration. Das Szenario kann nur in einem „Augmented Reality“-System glaubhaft dargestellt werden, da ein Großteil der Komponenten in der Wirklichkeit nicht existieren können (Laserschwert, schwebende Drohne).

2.3.2 Drohne

Die Drohne ist eine im Durchmesser ca. 20 cm große Kugel mit acht weißen Laseröffnungen, welche jeweils im Abstand von 90° auf zwei Ringen angeordnet sind. Diese Ringe befinden sich auf dem 30. Längengrad nördlicher und südlicher Breite und sind zu einander um 45° verschoben. Zur graphischen Darstellung wurde aus der Photographie eines Fan-Nachbaus der Drohne [5] mit Hilfe von Perspektivenkorrektur eine 2D-Textur (Abb. 2.3) erstellt.



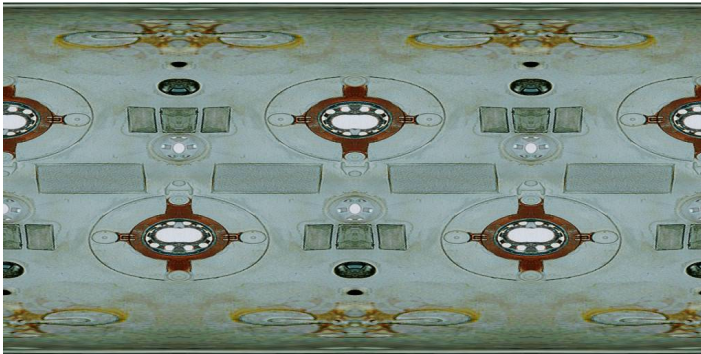
(a) Photo der Drohne[5]



(b) Bezierpatch



(c) Extraktionsergebniss



(d) Textur (eine Hälfte)



(e) 3D Resultat

Abb. 2.3: Das Entstehen der Drohne: Vom Foto über die perspektivische Texturextraktion hin zum computergenerierten Abbild

2.3.3 Laserschwert

Der reale Teil des Laserschwerts, im Gegensatz zur virtuellen Laserkomponente, entstand im Rahmen einer Studienarbeit von Stefan Burger als AR-Zeigergerät mit Bluetooth-Anbindung, vibratorischer Krafrückkopplung und drei Funktionstasten. Das in seiner Grundfunktionalität als „Zauberstab“ bezeichnete Eingabegerät ist mit vier AR-Toolkit-Marker zur Positionsbestimmung ausgestattet. Ein Aufsatz für das Polaris-Tracking-System wurde provisorisch angebracht (Abb. 2.4). Da beide Studienarbeiten zeitüberlappend entstanden und Bluetooth bis zuletzt Probleme bereitete, konnte es keinen Eingang in die für die Studie verwendete Version des Spiels finden. Auch stand der Titel der Ausarbeitung kurz vor Abschluss meiner Arbeit nicht fest und konnte deshalb nicht ins Literaturverzeichnis eingetragen werden.

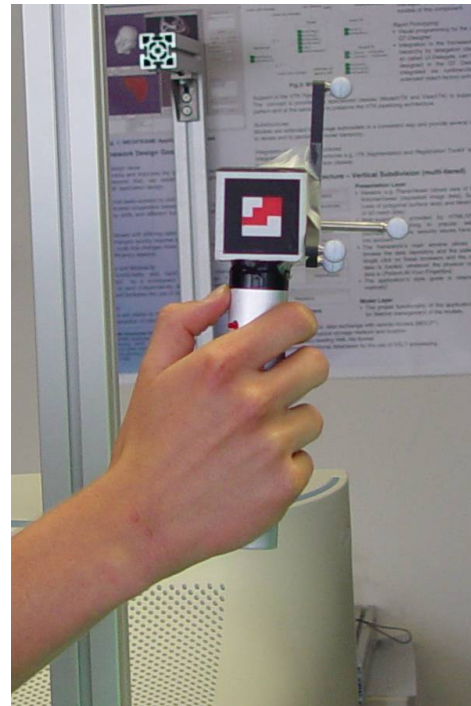
2.3.4 Spielverlauf

Beim Start der Anwendung befindet sich das Spiel im Wartezustand und kann durch Druck auf Taste ‘n’ für „New Game“ aktiviert werden. Der Punktestand wird hierbei auf null und die Lebenspunkte auf fünf zurückgesetzt. Die Drohne beginnt dann ihre Position in einem rechteckigen 3D-Bereich innerhalb des Sichtfeldes des Spielers zu verändern und feuert mit gewisser Wahrscheinlichkeit einen Laserstrahl auf den Spieler ab. Der Spieler muss versuchen, mit der virtuellen Laserkomponente diesen Strahl zu parieren. Gelingt dem Spieler die Abwehr des Laserstrahls, so ist dieser von nun an ungefährlich und wird in den Raum zurück reflektiert (Code hierzu aus [4] S. 370ff) und der Spieler bekommt seinen Punktestand um eins erhöht. Gelingt die Parade nicht und erreicht der Strahl den Spieler (genauer: die Near-Plane des Sichtbereichs), so verliert der Spieler einen Lebenspunkt.

Alle zehn Punkte wird der Schwierigkeitsgrad gesteigert und der Spieler erhält einen Lebenspunkt zurück. Der Spieler sieht sich dann mit steigenden Geschwindigkeiten bei Drohne und Laserstrahlen und einer größeren Varianz dieser konfrontiert. Es kann somit in höheren Schwierigkeitsgraden vorkommen, dass ein früher abgeschossener Schuss vor einem ihm nachfolgenden den Spieler erreicht. Sinkt die Anzahl der Lebenspunkte des Spielers auf Null, so endet das Spiel und der Spieler wird aufgefordert seinen Namen einzutippen, welcher dann mit Punktestand vermerkt wird. Abbildung 2.5 zeigt die 3D-Darstellung und die OnScreenDisplay-Elemente aus Sicht des Spielers.



(a) Das Laserschwert



(b) Im Einsatz

Abb. 2.4: Das Zeigegerät „Zauberstab“ mit aufgeklebten ARToolkit-Markern und retroreflektiven Targets zur Positionsbestimmung. In Kopf des Stabes verbirgt sich ein Bluetooth-Modul und ein Vibrationsmotor.

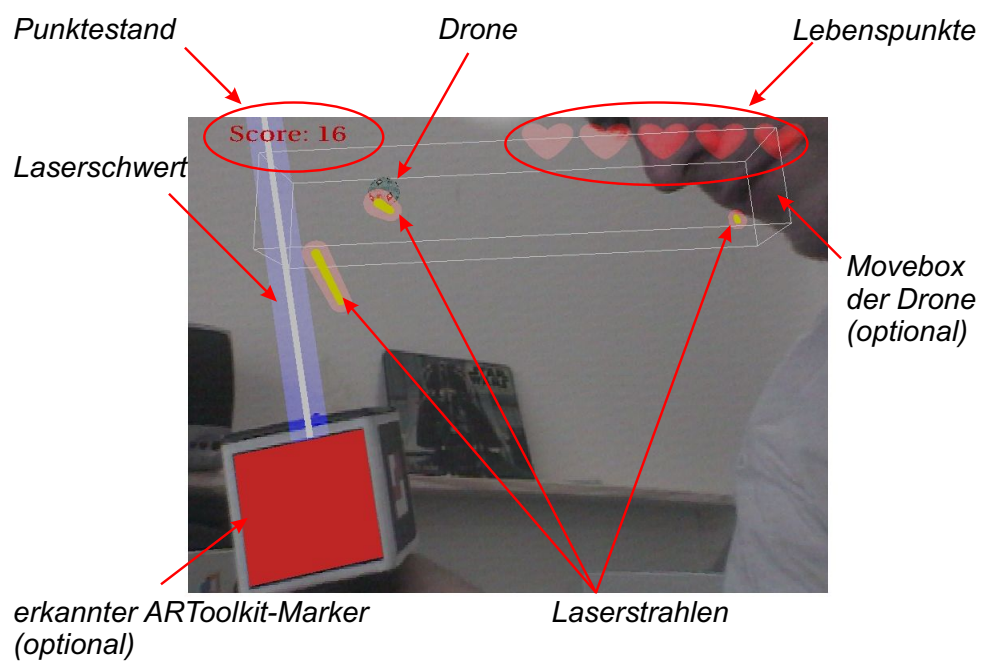


Abb. 2.5: Übersicht über die Anzeige während des Spiels

Kapitel 3

Studie

3.1 Vorüberlegungen

Die anschließende Studie sollte v.a. einen Überblick über die im Rahmen von Nutzeranwendungen entstehenden Probleme mit „Augmented Reality“-Systemen geben. Deshalb wurden auch nicht gezielt zwei verschiedene Komponenten miteinander verglichen, sondern gleich zwei gänzlich verschiedenen Systeme:

Komponente	System 1	System 2
Realität	Video-See-Through (VST)	Optical-See-Through (OST)
Tracking	AR-Toolkit	Polaris
Rendering	Dual-Window-Renderer	Inventor-Renderer mit Shutter-Stereo
Brille	Triviso ARVision-3D HMD	Sony Glasstron

Dies macht es natürlich schwierig einzelne Kausalzusammenhänge aus der Datenmenge zu extrahieren, ist jedoch im Rahmen dieser kleinen Forschungsarbeit sinnvoller als eine ausgedehnte Studie mit einem Vielfachen an Probanden.

Folgende Fragekomplexe werden von der Studie erörtert:

Hand-Augen-Koordination Beeinflusst das Tragen einer AR-Brille und Benutzen eines AR-System die Hand-Augen-Koordination, und wenn ja: Wie stark macht sich dies bemerkbar?

3D-Gefühl Sind die Probanden in der Lage in der augmentierten Sicht ein 3D-Gefühl aufrecht zu erhalten und Tiefen-Entscheidungen richtig zu treffen?

Tracking Welche Vor- und Nachteile besitzen die Tracking-Mechanismen, und wie gut kommen die Probanden mit den jeweiligen Einschränkungen klar?

Komfort Wie unterscheiden sich die beiden Systeme im Bezug auf den erfahrenen Komfort über einen mittelkurzen Zeitraum von jeweils zehn Minuten. Schwerpunkte lagen auf der Kopfhaltung, visuellen Qualität, Wahrnehmung der realen Umgebung, Anstrengung für die Augen und Verträglichkeit für Brillenträger.

Akzeptanz Wie groß ist die alltags- und berufsspezifische Akzeptanz der Systeme?

Vorwissen Wie wirken sich Vorwissen und Kenntnissen über AR, Fähigkeiten im Umgang mit besonderen 3D-Anforderungen (durch Sport, ähnliche Spiele) und technische Anpassungsfähigkeit auf die Leistung im System aus?

Endnutzertauglichkeit Nicht zuletzt sollte die Studie auch den Einsatz des Frameworks, Brillen und Tracking-Verfahrens in einer Endnutzerkonfiguration testen, um somit Einblicke an erweiterte Anforderungen jenseits des Entwicklungsumfeldes zu gewinnen.

3.2 Testpersonen und Auswahl

Geplant war, die Studie an 30 freiwilligen Probanden durchzuführen, welche sich im Vorhinein auf direkte Ansprache oder nach Aufruf in der Vorlesung „Kognitive Systeme“ melden würden. Leider blieb der Aufruf in der Vorlesung nahezu unbeantwortet, was sicherlich daran lag, die Terminliste nicht im Saal herumzugeben, sondern vorne auf dem Pult zu hinterlegen. Nach anstrengenden anderthalb Stunden Vorlesung hatten die meisten Studenten die vorangehende Präsentation wohl vollständig vergessen.

Dass dann aber doch 40 Probanden gefunden werden konnten, lag wohl letztlich an der interessanten Themenwahl, die es ermöglichte nahezu jede angesprochene Person zu dem knapp halbstündigen Spieltest mit anschließender Fragebogenbeantwortung zu bewegen. Der Überschuss von zehn Probanden kompensierte hierbei das Ausscheiden von 9 Datensätzen, welche durch fehlerhafte Verkabelung (5), temporären LCD-Defekt auf dem linken Auge der Video-Brille (3) und fehlender Zeit für einen vollständigen Durchgang (1) verloren gingen.

3.3 Durchführung

Die einzelnen Probanden bekamen zuerst einen Übersichtsbogen mit den am meisten gestellten Fragen (Anhang A.3) und die ersten vier allgemeinen Fragen des Fragebogens (Anhang A.2) zur Beantwortung. Anschließend begann die eine Hälfte der Probanden mit dem **Video-System (Gruppe V)**, während die andere zuerst mit dem optischen Durchsichtsystem (Abb. 3.1(b)) anfang (Gruppe O). Nach Aufziehen der Brillen, unter Anleitung und Überprüfung des korrekten Sitzes, mussten die Teilnehmer jeweils 3 Spiele am jeweiligen System durchführen. Nach jedem korrekt durchgeführten Spieldurchlauf wurden die Punktestände der Spieler am Fuß des Fragebogens vermerkt und jeder Spieler aufgefordert ohne die Brille abzusetzen, auf der bereitgestellten Tastatur seinen Namen einzugeben. Hierbei wurde explizit darauf geachtet, dass die Probanden nicht unter oder seitlich an dem Sichtmechanismus der Brille vorbeischaute und so zur ungewohnten Sicht durch das Augmented Reality System gezwungen waren. Wir testeten hierdurch die Fähigkeiten der Probanden im Bezug auf dem, neben der Maus wichtigsten Eingabemechanismus des Computers.

Nach den drei durchgeführten Durchläufen auf dem ersten System wechselten die Probanden nach einer möglichen Erholungspause, die allerdings von keinem Teilnehmer wahrgenommen wurde, das System und führten das gleiche Spiel auf dem anderen System ebenfalls in drei Iterationen und mit jeweiliger Tastatureingabe durch.



(a) Probanden beim Ausfüllen der Fragebögen



(b) Ein Teilnehmer beim Spiel am OST-System

Anschließend wurden die Probanden aufgefordert, die restlichen Fragen zu bewerten und ihre Verbesserungsvorschläge und Kritikpunkte niederzuschreiben (Abb. 3.1(a)). Alle Fragen bis auf die letzten zwei erforderten hierbei eine Kategorisierung von 0 bis 10, wobei null für „wenig, nein, schlecht, Ablehnung“ und 10 für „viel, ja, sehr gut, Zustimmung“ stand.

3.4 Statistik

Bei Antworten und Punkteständen wurden jeweils in dieser Reihenfolge Mittelwert (\bar{X}), Standardabweichung (s), Minimum (Min) und Maximum (Max) angegeben. Die Systeme wurden mit Index **v** für das Video-System und mit **o** für das optische System gekennzeichnet.

Zur Bestimmung von linearen Korrelationen wurde der Pearson-Test verwendet.

3.5 Ergebnisse und Diskussion

Unter den Probanden fanden sich 36 Männer und 4 Frauen, darunter insgesamt 15 BrillenträgerInnen. 38 Probanden waren Studenten der Universität Karlsruhe und die zwei übrigen Teilnehmer Mitarbeiter des Instituts.

3.5.1 Vorwissen

Die ersten vier Fragen zum Vorwissen und die Ergebnisse finden sich in folgender Tabelle:

Frage	\bar{X}	s	Min	Max
1. Wieviel Erfahrungen hattest Du bereits mit Augmented Reality vor Deiner Teilnahme?	0,52	1,03	0	4
2. Wie groß ist Deine Erfahrung mit Sportarten, die ähnliche Anforderungen an räumliches Sehen und Hand-Augen-Koordination wie das LaserTraining haben (z.B. Tennis)?	4,55	2,79	0	10
3. Wie sieht es mit Deiner Computerspiele-Erfahrung (v.a. 3D) aus?	5,19	2,89	0	9
4. Wie gut schätzt Du Dich und Deine Anpassungsgabe im Umgang mit neuen technischen Geräten und Umgebungen ein?	7,16	1,46	5	10

Die selbsteingeschätzte allgemeine Kenntnis über AR-Systeme war damit insgesamt sehr gering, während die Einschätzung der eigenen technischen Anpassungsfähigkeit bei beiden Gruppen sehr hoch war. Sowohl Erfahrung mit 3D-Spielen, als auch Erfahrung mit Sportarten, welche räumliches Sehen und Koordination verlangen (Tennis, Jonglieren, Badminton) waren in beiden Gruppen recht gleichmäßig über das Spektrum verteilt. Der Durchschnitt in der Gruppe „Optical-See-Through“ lag allerdings bei Sport mit $\bar{X}_o = 5.2$ um 1.26 höher als bei der Video-Gruppe ($\bar{X}_v = 3.94$) und bei 3D-Spielen mit $\bar{X}_o = 5.53$ um 0.66 höher ($\bar{X}_v = 4.88$).

Untersucht man die Parameter Sport, Computerspieleerfahrung und technische Anpassungsfähigkeit auf eine Korrelation mit der in den Spielen erreichten Leistung, so ergaben sich keine Pearson-Faktoren von über $|0.3|$. Damit ist keine starke lineare Korrelation zwischen Antworten auf die Fragen und der erreichten Leistung gegeben. Werte nahe -1 bzw +1 würden diese Aussage zulassen. Trennt man die Probanden in eine starke und eine schwache Gruppe bezüglich eines Attributes, so sind nur bei der Erfahrung mit Sport geringe Unterschiede bei den Punktezahlen erkennbar (Durchschnitt unспортlich = 42,68, Durchschnitt sportlich = 50,77).

3.5.2 Erreichte Punkte

Die Probanden spielten auf beiden Systemen drei Spiele, um ihre Leistungen mit der Video- und optischen Durchsichtbrille zu evaluieren. Bei den ersten drei Läufen, war ein deutlicher Lerneffekt zu spüren: Die Probanden steigerten ihre Punktzahlen im Verlauf auf das Doppelte. Dies lässt sich sehr schön auf Gewöhnung bzw. „Einspielen“ zurückführen. Die anschließenden Läufe auf dem anderen System verliefen dann relativ

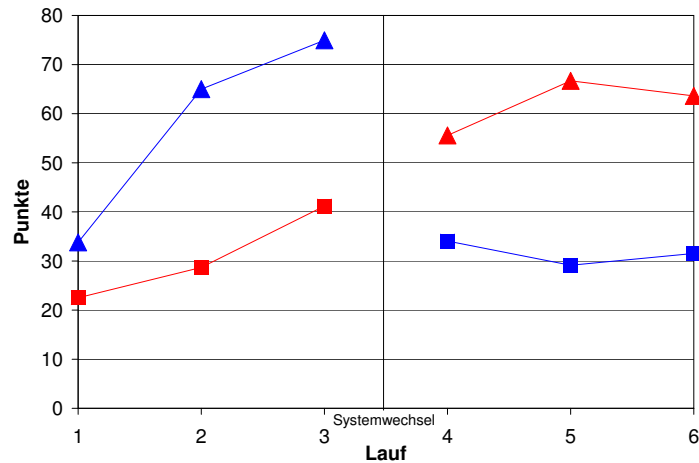


Abb. 3.1: Durchschnittliche Punktezahlen der Spieler aus Gruppe V (rot) und Gruppe O (blau) in Abhängigkeit zur Nummer des Durchlaufes. Dreiecke sind Spiele am optischen, Vierecke solche am Video-System. Beim vierten Spiel wechselten die Gruppen die Geräte („Systemwechsel“).

konstant ($\pm 10\%$). Nach dem Systemwechsel erreichten beide Gruppen mit den Läufen 4 bis 6 jedoch nicht das Niveau, welches die jeweils andere Gruppe am gleichen System mit dem 3. Lauf aufgestellt hatte. Die sinkende Motivation nach knapp 20 Minuten Spiel ist hierfür wohl ausschlaggebend. Punktezahlen in Abhängigkeit von der Nummer des Laufs finden sich in Abbildung 3.1. Kästchen stellen Spiele am Video-System und Dreiecke solche am Optical-See-Through-System dar, wobei der blaue Graph zur Gruppe O gehört und der rote zur Gruppe V.

Der Vergleich zwischen den System gewinnt insgesamt die optische Durchsicht-Konfiguration. Die Probanden erreichen mit letzterem System, egal ob sie dieses zuerst oder zuletzt spielten, nahezu doppelte Punktzahlen im Durchschnitt pro Lauf, welche in folgender Tabelle dargestellt sind. Die Durchschnittswerte liegen hierbei durch den obig erwähnten Eingewöhnungseffekt bei dem zuerst verwendeten System etwas niedriger.

	Gruppe VST zuerst	Gruppe OST zuerst
VST	30,80	31,58
OST	61,96	57,91

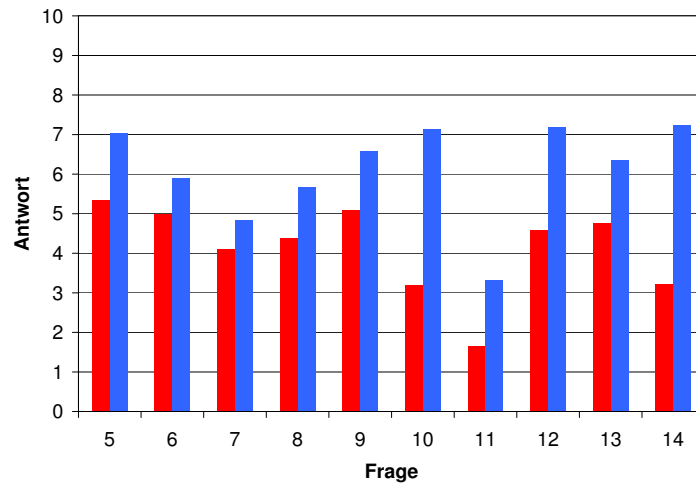


Abb. 3.2: Übersicht über die Durchschnittswerte der Antworten für Fragen 5-14 in rot beim Video- und blau beim optischen-System. Es lässt sich gut die Präferenz für das optische System erkennen.

3.5.3 Fragebogen

Bei der Beantwortung der systemspezifischen Fragen 5 bis 14 wurde die Polaris/OST-Kombination entsprechend den höheren Punktezahlen besser bewertet. Um die zwei Gruppen zu einer Population zusammenzufassen, wurden die Unterschiede zwischen den Durchschnittswerten der Antworten ausgewertet. Da sich alle Mittelwerte hierbei mit einer Ausnahme um nur bis zu 0.8 unterscheiden ist die Zusammenlegung wohl ungefährlich.

Der einzige größere Unterschied trat bei Frage 6 „Wie gut war Deine Wahrnehmung der realen Umwelt (Schärfe, Kontrast)?“ auf. Aufgefordert diese Frage im Bezug auf das Videosystem zu beantworten, erhielt es von der Video-Gruppe wesentlich bessere Bewertungen ($\bar{X}_v = 6,06$, $s_v = 2,17$, $min = 3$, $max = 10$), als von den Probanden aus Gruppe O ($\bar{X}_o = 3,87$, $s_o = 2,13$, $min_o = 2$, $max_o = 9$). Im umgekehrten Fall, bei der Bewertung des optischen See-Through-Systems, kam es zu keiner solchen Abweichung. In der folgenden Betrachtung werden die zwei Gruppen zu einer Population zusammengefasst, deren Stichproben-Durchschnitte an den zwei unterschiedlichen Systemen in Abbildung 3.2 dargestellt werden.

Bildqualität und Umgebungswahrnehmung Fragen 5 bis 7 beschäftigen sich mit der Bildqualität und Umgebungswahrnehmung der zwei unterschiedlichen Brillensysteme:

Frage	Video-System				Optical-System			
	\bar{X}	s	Min	Max	\bar{X}	s	Min	Max
5. Wie beurteilst Du die Bildqualität der künstlichen 3D Umgebung (Drohne, Laser)?	5,35	2,42	2	10	7,03	1,47	4	10
6. Wie gut war Deine Wahrnehmung der realen Umwelt (Schärfe, Konstrast)?	5,00	2,39	2	10	5,90	1,89	2	10
7. Wie echt erschienen Dir die Simulationselemente Vergleich zur Realität?	4,10	2,30	0	10	4,84	1,98	1	9

Bei keiner der Fragen konnte die Triviso-Brille in Konkurrenz zur Brille aus Fernost treten. Die Sony Glasstron gewinnt selbst auf dem Gebiet „Sicht auf die reale Welt“ (Frage 6), welche durch die halbreflektiven Gläser abgedunkelt wird. Die Qualität der Verschmelzung von Realität und Virtualität (Frage 7) wird mit Werten zwischen 4 und 5 als nicht sehr weit fortgeschritten eingeschätzt.

3D-Gefühl Auf die Frage 8 „Wie beurteilst Du das erlebte 3D-Gefühl?“ antworten die Probanden mit mittelmäßigen Werten für beide Systeme (Video: $\bar{X}_v = 4,39$, $s_v = 2,35$, $Min_v = 1$, $Max_v = 10$; Optical: $\bar{X}_o = 5,68$, $s_o = 2,34$, $Min_o = 0$, $Max_o = 10$). Hier liegt ein kritischer Punkt der Untersuchung: Wenn es nicht gelingt mit AR das gleiche 3D-Gefühl zu erzeugen, welches unsere herkömmliche Sicht auf die Welt erzeugt, wird Augmented Reality nicht an Popularität gewinnen können.

Trage- und Sehkomfort Für den längerfristigen Einsatz von „Augmented Reality“-Systemen sind Komfort von Kopfhaltung (Frage 9) und Sicht (Frage 10) besonders wichtig. Während die Triviso-Brille bei ersterer Fragestellung noch mithalten kann, wird sie bei der Frage zum Sehkomfort klar distanziert. Die Anstrengung für die Augen sind so groß, dass sieben Probanden über Augenschmerzen und zwei gar über Schwindelgefühle klagten. Beim optischen System beklagte nach längerem Spiel ein Proband die Ermüdung seiner Augen.

Frage	Video-System				Optical-System			
	\bar{X}	s	Min	Max	\bar{X}	s	Min	Max
9. Wie angenehm fandest Du das Tragen der Brille von der Art der Kopfhaltung?	5,10	2,20	0	9	6,58	1,80	3	10
10. Wie angenehm fandest Du das Sehen durch die Brille?	3,19	1,94	0	7	7,13	1,41	4	9

Akzeptanz Die nächsten zwei Fragen beschäftigen sich mit der Akzeptanz der Systeme in Alltag (11) und Beruf (12).

Frage	Video-System				Optical-System			
	\bar{X}	s	Min	Max	\bar{X}	s	Min	Max
11. Kannst Du Dir vorstellen, ein solches System im Alltag zu tragen?	1,65	1,64	0	7	3,32	2,31	0	8
12. In bestimmten Berufssituationen?	4,58	2,73	0	9	7,19	1,76	2	10

Die Ergebnisse lassen keinen baldigen Einsatz im vernetzten Heim erwarten, zumindest aber das optische System könnte für zeitlich begrenzte Tätigkeiten in speziellen Berufsfelder tauglich sein.

Brillenträger Unter den gültigen Probanden befanden sich 13 Teilnehmer mit und 18 ohne Brille. Die Frage „Wie gut verträglich war das System im Bezug auf Deine Brille?“ ergab bei der Triviso-Brille des Video-Systems mittlere Werte mit hoher Streuung ($\bar{X} = 4,77$, $s = 3,65$, $Min = 0$, $Max = 10$) und etwas höhere bei der Sony-Brille beim optischen Durchsichtsystem ($\bar{X} = 6,85$, $s = 2,38$, $Min = 3$, $Max = 10$).

Die Brillenträger der Studie sind nach den von ihnen abgegebenen Antworten weniger mit Sport in Berührung ($\bar{X}_{ohne} = 5,06$, $\bar{X}_{mit} = 3,85$), dafür aber mehr vor dem Computer zu finden ($\bar{X}_{ohne} = 4,55$, $\bar{X}_{mit} = 6,07$). Sie bewerteten beide Systeme im Durchschnitt schlechter, nur bei der Video-Brille fielen jedoch die folgenden Unterschiede der Mittelwerte um mehr als einen Punkt schlechter aus: Die Wahrnehmung der Realität im Hinblick auf Schärfe und Kontrast (Frage 6: $\bar{X}_{vohne} = 5,67$, $\bar{X}_{vmit} = 4,08$), die Sicht allgemein (Frage 10: $\bar{X}_{vohne} = 3,78$, $\bar{X}_{vmit} = 2,38$) und die Tauglichkeit in gewissen beruflichen Umfeldern (Frage 12: $\bar{X}_{vohne} = 5,33$, $\bar{X}_{vmit} = 3,54$) werden mit Abschlügen zwischen 1,39 und 1,79 im Durchschnitt versehen. Die Brillenträger schnitten auch deutlich schlechter ab, wenn man die jeweils im Spiel erzielten Punkte betrachtet:

Spiel	System 1 (VST)		System 2 (OST)	
	Brillenträger	Nicht-Brillenträger	Brillenträger	Nicht-Brillenträger
1.	21,79	32,67	32,00	54,44
2.	22,00	33,89	55,46	73,39
3.	24,07	45,50	65,08	72,00
Ø	22,61	37,35	50,85	66,61

Tastatur-Eingabe Die Probanden kamen sehr schlecht mit dem Video-See-Through-System zurecht, als es daran ging den Namen über die Tastatur einzugeben. Bis auf zwei Probanden gaben alle anderen dem Optical-See-Through-System auf die Frage „Wie gut

kamst Du mit der Tastatureingabe zurecht?“ gleiche oder bessere Punkte (Video: $\bar{X} = 3,23$, $s = 2,17$, $Min = 0$, $Max = 8$; Optical: $\bar{X} = 7,23$, $s = 2,35$, $Min = 2$, $Max = 10$).

Da dieses Nicht-Zurechtkommen jedoch sowohl der Bildwiederholrate als auch der veränderten Perspektive angelastet werden kann, bleibt es schwierig ein generelles Urteil auszusprechen. Der experimentelle Setup gibt hierüber keine direkte Auskunft; es beklagten jedoch mehr Probanden die geringe Framerate, als dass sie sich über einen ungewohnten Blickwinkel mokierten.

Die Tastatureingabe machte deutlich, dass ein Video-System wie unseres im momentanen Zustand nicht für Produktionsumgebungen oder gar lebenskritische Umgebungen, wie z.B. invasive Chirurgie geeignet sind. Die Koordination zwischen Sicht, virtueller und realer Umgebung ist bei momentanen Frameraten zu schwierig.

Freiform-Felder Bei den zwei Freiform-Fragen Nr. 15 und Nr. 16 konnten die Probanden mit eigenen Worten ihrer Meinung freien Ausdruck geben.

Aus den Antworten zu Frage 15 „Welcher Teil des System Bedarf der größten Verbesserung für eine bessere Nutzbarkeit (z.B. Auflösung, Kontrast, Sichtfeld, Framerate, Tracking, 3D-Gefühl)?“ wurde die folgende Liste nach Häufigkeit der Antworten für beide Systeme erstellt:

VST	Frage 15 sortiert nach VST	OST
21	Framerate bzw. Lag	2
16	Tracking	7
13	Sichtfeld	3
11	3D Gefühl schlecht	9
7	Augen schmerzen	1
6	Am Rand schlecht sichtbar	0
4	Auflösung	1
3	Kontrast schlecht	5
3	Einstellung der Brillen	0
3	Kontaktlinsen schlecht	0
2	Reale Welt irritiert nur	1
2	Wirres Kopfgefühl, Schwindel	0
2	Flimmern im Video	0
1	Kalibrierung	3
1	Brille verschob sich	0

OST	Frage 15 sortiert nach OST	VST
9	3D Gefühl schlecht	11
7	Tracking	16
5	Kontrast schlecht	3
4	Welt zu dunkel	0
3	Sichtfeld	13
3	Kalibrierung	1
2	Framerate bzw. Lag	21
1	Augen schmerzen	7
1	Auflösung	4
1	Reale Welt irritiert nur	2
1	Nur durch Schielen 3D-Sicht	0
1	Fehler mit Blickwinkel	0

Frage 16 erfragte die „Sonstigen Verbesserungsvorschläge und Anmerkungen“ der Teilnehmer, welche diese gerne in Zukunft realisiert sehen würden. In der folgenden Auflistung finden sich auch die Einträge, die bereits auf dem Merkzettel mit den am häufigsten gestellten Fragen (FAQ A.3, S. 40) zu finden sind:

- **Zum Zauberstab/Laserschwert und 3D-Brillen:**
 - Zauberstab schwer über die Zeit / Zeigegerät zu schwer
 - Bluetooth im Zauberstab zum Laufen bekommen.
 - Seitenabschattung fehlt / Brillen um die Augen anliegend
- **Zum LaserTraining-Spiel:**
 - Typische Laserschwertsounds integrieren
 - Möglichkeit, die Drohne zu zerstören
 - Strahl und Kugel nicht die idealen Objekte für 3D
 - Spielgeschwindigkeit für Studie zu langsam
 - (Wohl nicht ganz ernst gemeint:) Mehr Monster und Blut; anstatt Herzchen → Rebellen-Logo aus Star Wars
 - Selbstverletzung mit dem Laserschwert
 - Beim Beginn des Spiels feuert die Drohne teilweise direkt
 - Spiel war eher VR als AR, keine Kollision mit realen Objekten
 - Spiel ist mehr 2D-Pong, nicht 3D-Aufgabe
- **Zum Augmented-Reality-System:**

- 3D-Eindruck nicht von Umgebung unterstützt, da keine Interaktion zw. Realität und Virtualität stattfindet
 - Trackingbereich bei beiden Systemen erweitern
 - Schatten der 3D-Objekte in der echten Welt
 - Tracking bei ARToolkit auch auf dem rechten Auge möglich
- **Zur Studie:**
 - Einheitlichen Hintergrund für beide Systeme
 - FAQ-Verbesserungsvorschläge umsetzen
 - Bessere Einführung
 - Testumgebung für alle gleichartiger gestalten (Licht, Reden anderer Leute)

3.5.4 Sonstiges

Tracking Ein Hauptkritikpunkt, nahezu von jedem Probanden während des Spiels mündlich geäußert, ist die wenig intuitive Verwendung des ARToolkits zum Tracken des Laserschwertes. Dieses ständig im Blickfeld des linken Auges halten zu müssen, wurde von nahezu allen Probanden (nicht nur den Rechtshändern) als ermüdend und irritierend bezeichnet.

Kapitel 4

Diskussion und Schluss

In der durchgeführten Studie gelang es dem optischen Durchsichtssystem aus Sony Glasstron-Brille und NDI Polaris-Tracker sich klar vom Video-System aus Trivisio ARVision-3D HMD und ARToolkit zu distanzieren. Sowohl bei der gemessenen Spieleleistung als auch bei der Beantwortung sämtlicher Fragen schnitt ersteres System besser ab. Es offenbarte sich eine Fülle von Arbeitsschwerpunkten für das weitere Vorgehen, die in folgendem Aktionsplan zusammengefasst werden:

4.1 Aktionsplan

1. Die **Bildwiederholrate bzw. Framerate** beim Videosystem ist auf akzeptable Werte zu steigern. Anzustreben sind Frameraten > 25 ; besser wäre ≈ 50 . Ob dies überhaupt mit bisheriger Hardware möglich ist, muss getrennt untersucht werden. Da jedoch beim Rendern in nur ein Fenster 25 fps souverän erreicht werden, kann jedoch mittels gezieltem Profiling sicherlich der verursachende Flaschenhals gefunden werden.
2. Das **Tracking** des AR-Toolkits muss auf vielfältige Weise verbessert werden. Zu minimieren sind Zittern und Beleuchtungsabhängigkeit, wie sie das Spiel vieler Probanden erschwerten. Ein Wackeln in der zurückgegebenen 3D-Koordinate könnte z.B. durch einen Trägheitsfilter unterbunden werden. Das Problem der Lichtverhältnisse und deren starke Einflussnahme auf die Trackingqualität des Toolkits wird wohl einen tiefergehenden Eingriff in den Source-Code selbst benötigen. Der resultierende Zugewinn an Handhabungskomfort wäre auf jeden Fall die Mühe wert.

Viele Spieler forderten, das Tracking des Zauberstabes nicht durch die Kamera für den Sichtbereich, sondern durch ein separates bildgebendes Gerät, z.B. eine festinstallierte Webcam oder eine Panoramakamera durchzuführen.

Hierfür wäre die Entwicklung einer verteilten Version (siehe auch Punkt 8 „Verteilte Augmented Reality“) von PorkiII sinnvoll, welche den Austausch von Daten über die Grenzen von Rechnern ermöglicht, damit aufwendige Berechnungen und bildbasierte Tracking-Algorithmen nicht den Gesamteindruck des Nutzers trüben.

3. Das erlebte **3D-Gefühl** wurde als minderwertig beschrieben und bedarf gezielter Verbesserung. Zumindest der software-abhängige Teil sollte in den nächsten Versionen von PorkiII bzw. der entsprechenden Rendermodule in Angriff genommen werden. Eine weitere Studie hierzu wäre anzuraten, die ermittelt, welche Kombinationen von Parametern (Augenabstand, Augenwinkel) beste Resultate beim Greifen und Abschätzen von virtuellen Objekten und Entfernungen, Größen etc. ermöglichen. Hardwareabhängige Entwicklungen in diesem Gebiet gilt es parallel zu evaluieren (z.B. [20]).
4. Das kleine **Sichtfeld** v.a. der Videobrille wurde bemängelt. Leider wurde nicht ganz klar, ob hiermit das „Field of View“, d.h. der durch die Videokameras aufgenommene Bereich, oder die Größe der LCD-Bildschirme gemeint war. Beide wären verbesserungswürdig, sind aber durch die Hardware vorgegeben.
5. Obwohl die **Kalibrierung** zwischen realen und virtuellen Objekten bei keinem der beiden Systemen ein wirkliches Hindernis auf Grund der Art des Spiels darstellte, ist diese von der wissenschaftlichen und programmiertechnischen Seite garantiert eine der Hauptangriffspunkte für weitere Entwicklungen. Eine Kalibrierung für das Polarissystem entstand bereits hier am Institut und muss jetzt nur noch in PorkiIII integriert werden [8]. Das Verfahren ist leider ein aktiv vom Benutzer durchzuführendes und daher auf Grund des Aufwands nicht gut verwendbar. Ein ausschließlich rechnergestütztes Verfahren wäre auf längere Sicht garantiert wünschenswerter, wenn dies auch zusätzliche Hardware erfordern würde. Leider gab es keine explizite Frage hierzu auf dem Fragebogen, aber die Anzahl der Anmerkungen im Bezug auf die Kalibrierung lag mehrheitlich beim Optical-See-Through-System, was durch das 1 zu 3 Verhältnis bei Frage 14 angedeutet wird.
6. Die **Settings-Datei** enthält kleinere Schwachstellen, die gezielt durch eine studentische Hilfskraft in weniger als 30 Stunden ausgeschaltet werden können. Das Einfügen von externen Dateien und Ersetzen von bestehenden Optionen wird sicherlich in unter 10 Zeilen Code erfolgen können. Die mathematischen Ausdrücke könnten z.B. durch die Verwendung einer Skriptsprache im Vorhinein geregelt werden. Für evaluierende Vorverarbeitungen sind diese schließlich da. Eine Warnfunktion an den Benutzer beim Auftreten von zyklischen Abhängigkeiten würde die Settings-Datei abrunden.
7. Die saubere **Verknüpfung der verschiedenen Koordinatensystem** ist mit der steigenden Zahl an Trackern und Renderern nötig geworden und sollte entweder durch

einen allgemeinen Filtermechanismus oder ein einheitliches Koordinatensystem gelöst werden.

8. **Verteilte Augmented Reality** als Möglichkeit, eine Welt gleichzeitig mehreren Benutzern zugänglich zu machen und die anstehenden Aufgaben wie Tracking oder komplexe Berechnungen auf mehreren Rechner zu verteilen, gilt es aufsetzend auf ein **Nachrichten- und Ereignis Framework** zu implementieren.
9. Die Steigerung der **Integration von Virtualität in die Realität** gehört zu den wohl aufwendigsten Aufgaben. Für die Erkennung von realen Okklusionen, d.h. Hindernissen, und deren Einflussnahme auf die 3D-Darstellung wurde bereits hier am Institut im Rahmen der Studienarbeit von Johannes Ahlmann [1] ein lauffähige Version erstellt, welche aber noch nicht in PorkiII integriert ist. Das Framework besitzt momentan auch noch keine Schnittstelle für eine solche Verarbeitungsstufe. Die umgekehrte Rückkopplung der Virtualität auf die Realität ist ebenfalls für einen allumfassenden realen Eindruck. So erhöhen Schatten von virtuellen Objekten auf realen den 3D-Eindruck und die Kohäsion der verschiedenen Sinnesebenen [16].
10. Effizientes **Videocapturing** für die Erstellung qualitativ hochwertiger Präsentationen und Anschauungsmaterialien erfordert eine hardwarenahe Arbeit, Suche nach bereits bestehenden allgemeinen Mechanismen (z.B. Grafikkarten-integriert) oder die Verwendung von Softwaretools [14].
11. Zur **Verbesserung der visuellen Qualität** bei den niedrigen Auflösungen der Brillen sollte der Umstieg auf Kantenglättung (Antialiasing) und die Verwendung von Freetype als Fontsystem erwogen werden.
12. Die **Loslösung von ARToolkit/GLUT** als Rendraufsatz scheint notwendig und würde das System in erheblichen Maße vereinfachen. Ein Teil der Koordinatenproblematik wäre damit bereits behoben, auch würden eine Reihe von unerwünschten Nebeneffekten zwischen GLUT und OpenInventor wegfallen.
13. Die bestehenden **Makefiles** zur Erstellung des PorkiII-Projekts, der Plugins und Welten kranken an verschiedenen Stellen. So müssen Bibliothekspfade momentan noch in jedem Unterverzeichnis erneut spezifiziert werden, ohne dass die Möglichkeit bestände, diese an untergeordnete Make-Prozesse weiterzureichen. Dies macht umfangreiche Anpassungen an den jeweiligen zugrunde liegenden Rechner und seine Installationsbasis notwendig. Als zweiter Punkt ist die Dokumentationsgenerierung ohne Änderungsmöglichkeit stets angeschaltet, dies verzögert den Kompilierungsprozess um gut das Doppelte. Die zwei weiteren Schwachpunkte resultieren aus dem schlechten Zusammenspiel zwischen Makefiles und Plugin-Struktur über die Verwendung von Shared-Libraries: Die mehrfache Verwendung von Bibliotheken mit globalem Datensegment (betroffen

waren die `bgtools`) in unterschiedlichen Plugins führt zu stets wiederholter Initialisierung, welche bereits getätigte Einstellungen stets überschreibt. Auch ist ein ein Profilen mittels `gprof` unmöglich, da dieses nicht über Shared-Library-Grenzen hinaus den Programmfluss verfolgen kann.

14. Eine **Verbesserung und Vervollständigung des LaserTrainings** hin zu einem technischen Exponat zum Anfassen und Benutzen im Sinne beispielsweise der Installationen des Zentrums für Kunst und Medientechnologien (ZKM) oder anderer medialer Erlebnismuseen könnte die wichtigen Funktion der Steigerung der Benutzerakzeptanz erfüllen. Es gilt hierfür die Grafik zu verbessern, Sound und Musik zu integrieren und die Intuitivität der Benutzung zu erweitern.

4.2 Dank

Großer Dank geht an meine Freundin Majbritt, die wohl am meisten unter der Arbeitslast der Studienarbeit zu leiden hatte, und meine Familie, die sich schon daran gewöhnt zu haben scheint, dass ich immer im Stress bin. An Johannes Ahlmann gehen zwei erhobene Daumen für unermüdlichen (im wahrsten Sinne) Einsatz und hervorragende Arbeit. Ohne ihn wäre LaserTraining/Polaris nicht mehr möglich gewesen! Meinem Betreuer Björn Giesler gilt Dank für die gute Betreuung und die geniale Idee. Schöne Grüße und vielen Dank geht natürlich auch an alle Probanden, die ihre Zeit für die Wissenschaft opferten. Ich hoffe es hat Spaß gemacht. Viel Freude allen im Medizinlabor bei Leber und Dr. Best. Gruß an die WG Beuschlein und Dank an Joris fürs Korrekturlesen.

Anhang A

Vorlagen zur Studie

A.1 Terminplan

**Teilnahmeliste für „Spielerische Evaluation
eines Augmented Reality Systems“**

Uhrzeit	Montag, 30.06.	Dienstag, 01.07.	Mittwoch, 02.07.	Donnerstag, 03.07.	Freitag, 04.07.	Uhrzeit
10:00						10:00
10:20						10:20
10:40						10:40
11:00						11:00
11:20						11:20
11:40						11:40
12:00						12:00
12:20						12:20
12:40						12:40
13:00						13:00
13:20						13:20
13:40						13:40
14:00						14:00
14:20						14:20
14:40						14:40
15:00						15:00
15:20						15:20
15:40						15:40
16:00						16:00
16:20						16:20
16:40						16:40

A.2 Fragebogen

Nr.		
<p>Fragebogen zur „Spielerischen Evaluierung eines Augmented Reality Systems“</p> <p>(10 sehr gut, sehr viel, absolute Zustimmung, 5 befriedigend, etwas, unentschlossen, 0 mangelhaft, gar keine, nein Danke)</p>		
Vorwissen und Kenntnisse		
	10-0	
1. Wieviel Erfahrungen hattest Du bereits mit Augmented Reality vor Deiner Teilnahme?		
2. Wie groß ist Deine Erfahrung mit Sportarten, die ähnliche Anforderungen and räumliches Sehen und Handaugenkoordinaten wie das LaserTraining haben (z.B. Tennis)?		
3. Wie sieht es mit Deiner Computerspiele-Erfahrung (v.a. 3D) aus?		
4. Wie gut schätzt Du Dich und Deine Anpassungsgabe im Umgang mit neuen technischen Geräten und Umgebungen ein?		
Vergleich der Systeme		
	Video-System mit Marker-Tracking	See-Through- System mit Polaris-Tracker
5. Wie beurteilst Du die Bildqualität der künstlichen 3D Umgebung (Drone, Laser)?		
6. Wie gut war Deine Wahrnehmung der realen Umwelt (Schärfe, Konstrast)?		
7. Wie echt erschienen Dir die Simulationselemente Vergleich zur Realität?		
8. Wie beurteilst Du das erlebte 3D-Gefühl?		
9. Wie angenehm fandest Du das Tragen der Brille von der Art der Kopfhalterung?		
10. Wie angenehm fandest Du das Sehen durch die Brille?		
11. Kannst Du Dir vorstellen, ein solches System im Alltag zu tragen?		
12. In bestimmten Berufssituationen?		
(13.) Für Brillenträger: Wie gut verträglich war das System im Bezug auf Deine Brille?		
14. Wie gut kamst Du mit der Tastatureingabe zurecht?		
<p>15. Welcher Teil des System Bedarf der größten Verbesserung für eine bessere Nutzbarkeit (z.B. Auflösung, Kontrast, Sichtfeld, Framerate, Tracking, 3D-Gefühl)?</p> <p>Video: _____ See-Through: _____</p>		
<p>16. Sonstige Anmerkungen oder Verbesserungsvorschläge:</p>		
<hr/> <p>V1: V2: V3: S1: S2: S3:</p>		

A.3 FAQ

Frequently Asked Questions

(V0.3 01.07.2003 – Christopher Özbek)

1. Wie ist der Ablauf des Versuchs?

Zuerst FAQ lesen. Dann je nachdem, ob Du ein gerader oder ungerader Teilnehmer bist, wirst Du zuerst am Video-System oder am See-Through-System jeweils 3 Partien LaserTraining absolvieren. Behalte die 3D Brille jeweils während der ganzen Session auf, auch wenn Du deinen Namen über die Tastatur eingibst. Dies ist Teil des Versuchs. Anschließend bekommst Du noch einen Fragebogen, den Du ausfüllen und abgeben solltest.

2. Im Videobild sehe ich alles doppelt!

Leider haben wir keine gute Kalibrierung für das Video-System entwickeln können (das wäre wahrscheinlich eine eigene Arbeit), welches Realität und Simulation in Überdeckung bringt und dazu auch noch ein komfortables Stereosehen ermöglicht. Die Einstellung des Video-System auf 3D-Sehen der Simulation wird am Anfang mit Dir vorgenommen.

3. Das Laserschwert zittert!

Das verwendete ARToolkit für das Video-System-Tracking hat leider einen großen Toleranzradius, was erkannte Marker angeht. Einen Trägheitsfilter für die ermittelten Positionen gibt es noch nicht.

4. Wie zieht man die Brillen an?

Die See-Through-Brille ist sehr unproblematisch: Du ziehst sie einfach über und passt gegebenenfalls den Umfang der Halterung durch Drehen an den Stellschrauben an. Die meisten Leute haben keine Probleme, alles klar und deutlich über die Displays zu sehen.

Die Video-Brille ist problematischer: Da die Display-Öffnungen kleiner sind, musst Du wahrscheinlich an der Innenseite der Brillen den Augenabstand für Dich anpassen. Schließe anschließend abwechselnd das linke und das rechte Auge. Das Bild sollte bei beiden Augen klar und hell sein, siehst Du nur einen Teil oder viel schwarzen Rand, dann musst Du nachkorrigieren.

5. Wäre es nicht besser, wenn die Laserstrahlen öfter quer durchs Bild fliegen würden?

Ich habe 5 Tage vor Beginn der Studie einen kategorischen Feature-Freeze begonnen. Leider können deshalb erstmal keine Änderungen mehr bis zum Ende der Studie eingepflegt werden. Bis zum Beginn der Studie sind nur noch Bugfixes möglich.

6. Wäre es nicht besser, wenn man die Drone abschießen könnte?

Die Drone kann von Dir abgeschossen werden! Sie wird nur ohne sichtbare Verzögerung unmittelbar durch eine neue Drone ersetzt! *G*

7. Wieso kann man das Laserschwert nicht einfahren? Wieso funktionieren die Knöpfe auf dem Laserschwert nicht? Was machen die lustigen Lichter auf dem Schwert? Wozu ist das Licht oben auf dem Schwert?

Leider ist das Laserschwert noch nicht voll einsatzfähig. Ein Bluetooth-Kern schlummert schon in seinem Innern, verweigert aber die Funkarbeit.

8. Muss man beim Video-System das Laserschwert immer in Sichtfenster haben?

Ja. Da wir das ARToolkit zum Tracking verwenden ist nur bei vollständiger Sicht mindestens eines Markers auf dem Schwert eine Verfolgung möglich.

9. Wo bleibt der Laserschwert-Sound?

Sicher. Ein entsprechendes Modul geschrieben in SDL existiert bei mir zu Hause auf der Festplatte. Wer Lust hat es in unser Framework einzupflegen sei herzlich eingeladen (e-Mail an oezbek@ira.uka.de)

10. Wieso funktionieren weder Enter noch Backspace beim See-Through?

Keine Ahnung wieso. Aber ein Workaround gibt es: 3 = Backspace und 4 = Enter

11. Cool! Was wollt ihr als nächstes damit machen?

Folgende Projekte stehen an und suchen noch tatenhungrige Hackernaturen:

- 3D Gestenerkennung. Das Laserschwert als Zauberstab.
- ARQuake
- Diminished Reality. Anstatt Informationen zur Realität hinzunehmen, wird hier die unliebsame Schwiegermutter einfach direkt komplett in Bild und Ton ausgeblendet.
- Wer sonst noch Ideen hat, die er umsetzen will oder bei einem der obigen Punkte loslegen will, meldet sich am besten bei Björn Giesler (giesler@ira.uka.de)

Anhang B

Rohdaten

Nr.	Geschlecht	Start-Konfiguration	Daten gültig	Video-See-Through-Konfiguration														Optical-See-Through-Konfiguration															
				Lauf Nr.														Lauf Nr.															
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	5	6	7	8	9	10	11	12	13	14			
1	m	O	J	1	9	4	8	7	2	4	6	5	2	0	3	0	2	38	15	17	8	8	2	0	5	9	0	10	0	10	28	28	6
2	m	V	J	0	3	0	5	5	8	5	3	5	4	1	4	0	4	7	8	17	7	5	5	5	7	7	4	6	0	2	7	7	25
3	m	O	J	4	6	7	5	7	2	4	6	6	4	0	4	0	2	10	4	9	7	6	5	7	8	7	4	7	0	9	9	121	8
4	m	O	J	0	2	3	8	5	4	7	8	6	5	2	5	0	5	25	15	31	7	6	6	6	6	7	3	6	0	8	1	6	18
5	m	O	J	0	3	3	5	3	2	3	2	3	1	2	2	1	3	1	12	4	6	5	5	6	6	7	5	8	5	6	8	45	37
6	m	V	J	0	5	7	8	3	5	2	2	3	5	2	2	0	3	15	16	39	6	5	6	4	5	5	2	7	0	5	18	70	103
7	m	O	J	3	4	9	8	8	7	5	5	6	3	1	8	10	5	73	23	49	5	4	5	3	4	5	1	8	10	5	15	50	86
8	m	V	J	0	6	8	7	9	8	5	4	4	6	1	8	0	4	21	37	14	5	6	9	9	4	6	3	9	0	10	8	66	95
9	m	O	J	0	8	8	7	6	4	1	1	9	5	3	9	0	5	105	71	77	8	8	1	4	9	9	3	9	0	8	75	95	123
10	m	V	J	0	3	9	8	7	5	7	7	6	2	2	3	1	8	3	9	35	8	7	7	7	8	9	7	8	3	5	55	3	7
11	m	O	N	0	5	10	7	4	3	3	3	4	3	0	3	0	2	17	18	12	6	7	7	6	5	7	5	8	0	6	69	13	86
12	m	O	N	0	1	2	9	6	7	3	3	2	1	0	1	10	5	6	1	12	6	4	3	5	5	7	7	7	10	9	66	91	99
13	m	V	N	0	8	8	7	7	5	4	6	4	3	2	4	5	5	7	6	7	6	7	4	7	6	7	8	8	7	4	33	52	24
14	m	V	N	6	7	9	6	6	5	4	5	3	3	2	4	0	0	9	5	16	8	7	8	7	8	8	5	8	0	7	1	22	14
15	m	O	N	0	3	3	8	5	3	3	1	7	6	0	5	0	8	8	5	21	6	5	3	3	8	6	0	5	0	8	6	4	113
16	m	V	J	0	1	1	7	6	7	5	2	5	6	2	9	0	1	4	65	111	8	4	5	4	6	8	2	9	0	4	108	139	109
17	m	V	J	0	0	2	10	5	3	1	3	7	3	3	8	0	1	5	16	34	8	10	5	8	5	7	8	8	0	10	84	86	126
18	m	V	J	2	1	6	8	4	3	2	2	4	1	0	2	0	0	42	7	37	6	5	4	6	3	4	2	5	5	4	25	61	4
19	m	O	J	0	3	7	8	2	2	0	2	0	0	0	1	6	3	6	23	4	8	8	5	9	7	8	3	7	6	9	68	43	98
20	m	O	J	0	3	7	8	3	3	5	5	4	2	0	2	3	18	21	38	8	5	5	5	5	6	0	5	6	5	68	64	119	
21	m	O	J	0	10	5	8	7	5	6	8	9	3	1	7	0	4	81	80	88	6	6	5	7	8	6	2	8	0	4	82	44	97
22	m	V	J	0	2	6	5	4	3	2	3	8	6	2	4	4	8	14	11	7	4	2	3	3	9	8	2	4	4	8	5	6	6
23	m	O	N	0	5	10	8	5	4	4	4	3	3	5	5	0	8	31	46	0	5	4	6	6	3	4	7	7	0	8	137	26	66
24	m	V	N	0	2	9	7	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	3	4	8	9	5	8	0	10	97	128	97
25	m	O	J	0	8	3	7	8	4	10	10	7	5	2	4	8	3	28	26	23	10	6	8	10	10	8	4	9	10	6	42	114	35
26	m	O	N	0	7	9	10	0	0	0	0	0	0	0	0	0	0	0	0	0	3	5	2	0	9	8	7	8	10	10	26	100	99
27	m	O	J	0	2	0	9	10	9	1	2	3	2	0	3	9	1	15	5	19	10	2	1	4	8	9	0	7	9	10	73	97	125
28	m	V	J	0	5	3	5	10	10	8	8	8	4	0	0	0	1	38	46	27	7	7	8	8	4	8	2	2	0	10	98	128	119
29	w	V	J	0	2	0	5	2	5	4	4	2	2	0	4	0	1	17	28	48	6	6	5	8	9	8	0	8	0	9	87	103	106
30	m	O	J	1	2	9	5	3	2	3	4	5	0	0	3	1	1	24	79	36	7	5	5	7	6	6	5	8	6	5	83	85	
31	w	V	J	0	5	6	8	2	5	2	1	2	1	2	6	0	1	8	24	51	6	6	5	6	7	6	3	6	0	4	87	88	126
32	m	V	J	2	4	9	10	8	9	5	3	5	4	2	9	0	6	57	57	103	6	4	8	4	9	9	4	9	0	10	137	137	1
33	m	O	J	2	10	4	7	2	2	5	4	2	2	2	5	6	2	29	3	15	4	6	3	1	7	6	5	9	10	7	24	29	109
34	m	V	J	1	5	2	6	7	6	3	7	3	0	0	0	0	3	46	27	59	9	8	3	7	4	6	0	5	0	8	101	65	85
35	m	O	J	0	2	7	9	4	4	5	4	7	3	2	4	4	4	25	22	18	7	8	4	5	7	8	4	6	7	7	9	64	68
36	m	V	J	0	5	5	7	5	5	5	7	5	7	3	5	0	4	16	22	13	8	7	4	7	7	7	6	7	0	10	34	37	24
37	m	V	J	0	7	9	7	7	7	2	3	7	4	5	7	10	6	5	45	28	8	9	2	5	7	8	8	9	8	9	19	62	67
38	m	N	3	10	8	10	5	8	3	3	5	2	0	2	0	0	0	0	0	0	8	5	3	5	5	6	0	5	0	0	0	0	0
39	w	O	J	0	6	7	7	4	6	7	5	5	5	7	7	0	0	33	38	45	8	3	7	8	8	9	7	7	0	7	0	92	110
40	w	V	J	0	9	5	7	3	8	3	5	7	2	4	6	0	6	62	41	36	7	6	4	3	6	5	4	7	0	9	16	9	15

Literaturverzeichnis

- [1] Johannes Ahlmann. Erkennen von Verdeckungen durch Fremdkörper in der Intraoperativen Visualisierung. Studienarbeit, Universität Karlsruhe, 2003.
- [2] Ronald Azuma, Yohan Baillet, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE CG&A*, Nov 2001.
- [3] Mark Billinghurst and Hirokazu Kato. Collaborative Mixed Reality. In *Proceedings of the International Symposium on Mixed Reality (ISMR '99)*, pages 261–284, March 1999.
- [4] Peter Burger and Duncan Gillies. *Interactive Computer Graphics*. Addison Wesley, 1989.
- [5] Frank Cerney. The Training Remote. <http://www.hobbymatrix.com/models/remote/remote.html>.
- [6] Schneider Digital. ATI-Treiber Downloads. http://www.schneider-digital.de/html/download_ati.html.
- [7] Björn Giesler et al. Porkiii. <http://www.iam.ira.uka.de/users/giesler/>.
- [8] Sassan Ghanai. Kalibrierung einer Durchsichtbrille und Aufbereitung medizinischer Bilddaten für ein System der Erweiterten Realität. Diplomarbeit, Universität Karlsruhe, 2003.
- [9] Björn Giesler. The bgtools Manual, Version 1.0. <http://www.iam.ira.uka.de/users/giesler/Download/manual.pdf>, Jan 2003.
- [10] Fred Gleason. The Video4Linux API. <http://www.informatics.ed.ac.uk/teaching/modules/sdp/creative/v4lapi.html>.
- [11] Susan L. Graham, Peter B. Kessler, and Marshall K. McKusick. gprof: a Call Graph Execution Profiler. In *SIGPLAN Symposium on Compiler Construction*, pages 120–126, 1982.

- [12] Mark J. Kilgard. The OpenGL Utility Toolkit (GLUT) Programming Interface. <http://www.opengl.org/developers/documentation/glut/glut-3.spec.pdf>, Nov 1996.
- [13] George Lucas. Star Wars, 1977.
- [14] Roderick Maher. FRAPS - fps, video capture, screen capture for DirectX & OpenGL! www.fraps.com, 2003.
- [15] Jan Minar. Positionskalibrierung von passiven Landmarken mit dem ARToolKit. Studienarbeit, Universität Karlsruhe, 2003.
- [16] Toshikazu Ohshima, Kiyohide Satoh, Hiroyuki Yamamoto, and Hideyuki Tamura. RV-Border Guards: A Multi-player Entertainment in Mixed Reality Space. Poster, presented at IWAR'99, San Francisco, Oct. 1999.
- [17] Inc. Silicon Graphics. *OpenGL Reference Manual*. Addison-Wesley Publishing Company, Nov 1992.
- [18] Paul Strauss and Rikk Carey. An object-oriented 3D graphics toolkit. *Computer Graphics (SIGGRAPH 1992 Proceedings)*, pages 341–349, 1992.
- [19] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, Reading, Mass., 3 edition, 1997.
- [20] A. Takagi, S. Yamazaki, Y. Saito, and N. Taniguchi. Development of a stereo video seethrough hmd for ar systems. In *Proceedings ISAR*, 2000.