Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Marcus Rödiger

Entwicklung und Aufbau eines multimedialen Augmented Reality Systems mit Hilfe visueller Techniken

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Marcus Rödiger

Entwicklung und Aufbau eines multimedialen Augmented Reality Systems mit Hilfe visueller Techniken

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Gunter Klemke
Zweitgutachter : Prof. Dr. Kai von Luck

Abgegeben am 26. August 2006

Marcus Rödiger

Thema der Bachelorarbeit

Entwicklung und Aufbau eines multimedialen Augmented Reality Systems mit Hilfe visueller Techniken

Stichworte

Augmented Reality, Virtual Reality, Informationssystem, Bildmustererkennung, Positionsbestimmung, Ambient Intelligence, Head-Mounted-Display

Kurzzusammenfassung

Ziel dieser Arbeit ist die Entwicklung eines Augmented Reality Systems, zur positionsabhängigen Anzeige von in die Realität eingebetteten virtuellen Informationen. Diese Lösung wird am Beispiel eines Museumsinformationssystems anschaulich dargestellt.

Marcus Rödiger

Title of the paper

Development and Building-Up of a multimedial Augmented Reality System with the assistance of visual technics.

Keywords

Augmented Reality, Virtual Reality, information System, pattern detection, position determination, ambient intelligence, Head-Mounted-Display

Abstract

Object of this paper is the development of an Augmented Reality System for position dependet presentation of in the reality embedded virtual informations. This solution is vividly shown by the example of a museum-information-system.

Damit das Mögliche entsteht,
muß immer wieder das Unmögliche
versucht werden.

Danksagung

Ich möchte an dieser Stelle meiner Familie und meinen Freunden danken. Ohne Sie und ihre Geduld wäre diese Arbeit nicht möglich gewesen.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

1.1 Motivation

Die Welt, wie wir sie kennen, wird es in der Zukunft nicht mehr geben. Gesellschaft und Technologie schreiten immer mehr voran. Insbesondere im Bereich der Informatik ist dieser Umstand ständig zu spüren. Wo früher Mainframecomputer in großen Hallen standen, stehen heute kleine Personalcomputer unter fast jedem Schreibtisch, welche einen Großrechner von damals mit ihren enormen Möglichkeiten locker abhängen. Doch diese Entwicklung ist noch nicht zu Ende. Anstatt einen Computer pro Mensch wird es dann hunderte kleiner Computerhelfer geben, welche uns den Tag erleichtern. Das Stichwort heißt Ambiente Intelligenz - Intelligente Umgebung. Die Umgebung versucht unser Leben besser und einfacher zu machen. Sei es nun der intelligente Kühlschrank, welcher sich die Verfallsdaten seiner Lebensmittel merkt und gegebenenfalls Alarm schlägt oder sei es das intelligente Auto, welches beim Parken den Abstand zum Wagen hinter einem laut vorsagt. Das sind alles Anwendungen, welche bereits heute das Leben der Menschen erleichtern oder in naher Zukunft erleichtern werden.

Ein Bereich innerhalb der Ambienten Intelligenz ist Augmented Reality. Augmented Reality ist die Anreicherung der Realität mit erweiterten virtuellen Informationen, welche uns von uns umgebenden Computern zur Verfügung gestellt werden. Die folgende Arbeit beschäftigt sich mit dem Thema Augmented Reality und zwar am Beispiel eines Museums. Es geht um ein Museumsinformationssystem, welches den Besucher mit erweiterten Informationen zu einzelnen Ausstellungsstücken versorgen kann.

Vor einigen Jahren war die technische Unterstützung durch ein Museumsinformationssystem praktisch nicht vorhanden. Die einzige Unterstützung erhielt man durch Museumsführer in Buchform oder durch Führungen eines Mitarbeiters. Erste Anfänge den Museumsbesucher durch Technik zu unterstützen

machten damals Tonbandgeräte mit Kopfhörern, mit welchen man einen vorgegebenen Pfad von Exponat zu Exponat ging und zeitlich abhängige Informationen vom Band vorgelesen bekam. Später gab es dann Kopfhörersysteme, welche - wenn man einen bestimmten Raum oder Bereich betrat - über Funk gesprochene Texte empfingen und in einer Endlosschleife abspielten. Der nächste Schritt waren Zahlen oder Strichcodes an einzelnen Ausstellungsobjekten. Mit Hilfe dieser Codes konnte jeder auf einem mitgeführten Handheldcomputer per Hand gezielt Informationen zum jeweiligen Ausstellungsstück abfragen.

Diese Arbeit soll sich mit einer alternativen Technik beschäftigen, welche einen Besucher frei im Museum herumgehen lässt und ihm individuell zu einem Exponat, vor dem er gerade steht, Informationen einblenden. Diese Informationen werden so dargestellt, dass der Besucher den Eindruck bekommt, die Informationen seien ein Teil der realen Umgebung. Diese Anzeige ist nur für den einzelnen Besucher sicht- bzw. hörbar. Dadurch kann jeder Besucher Informationen abrufen und sich jederzeit einem anderen Gemälde oder Kunstwerk zuwenden, ohne andere Besucher zu stören. Dieses System kann auf den Besucher abgestimmt werden und ihn so bei seinem Weg durch das Museum individuell unterstützen. Das Beispiel soll die Anwendungsgebiete und Möglichkeiten von Augmented Reality Systemen in der heutigen Gesellschaft zeigen.

1.2 Problemstellung und Zielsetzung

Schwerpunkt dieser Arbeit ist die Entwicklung eines Systems zur eingebetteten Darstellung von virtuellen Objekten im Realen Raum anhand des Beispiels eines Museumsinformationssystems. Das Szenario soll in einem fiktiven Kunstmuseum stattfinden. In dem Museum werden vornehmlich große Gemälde mit klassischen Motiven gezeigt. Es soll ein System entwickelt werden, welches dem einzelnen Benutzer ermöglicht individuelle Zusatzinformationen zu den Gemälden abzurufen, und eingebettet in die Realität angezeigt zu bekommen. Um andere Besucher nicht zu stören, sollen Bildbeiträge nur von den jeweiligen Besuchern wahrgenommen werden.

Bei den dargestellten virtuellen Informationen soll es sich, zur Vereinfachung, um Bilder von zweidimensionalen oder dreidimensionalen Figuren handeln. Es soll nur die optische Wahrnehmung manipuliert werden, nicht Teil dieser Arbeit ist die Anreicherung anderer Sinneswahrnehmungen wie Hören oder Fühlen.

Zur Bestimmung der Position und Ausrichtung vom Besucher gegenüber den Gemälden soll eine Lösung mittels visueller Techniken entwickelt werden. Ziel - aber nicht Schwerpunkt - ist die einfache Benutzereingabe mittels Handbewegungen.

Die Effizienz und Performanz des Systems spielt nur eine untergeordnete Rolle. Die Betrachtung der Wirtschaftlichkeit des Systems ist nicht Teil der Arbeit, jedoch sollen möglichst kosten- und lizenzfreie Komponenten verwandt werden.

Nicht Teil dieser Arbeit sind Betrachtungen zur Ergonomie und Akzeptanz eines solchen Systems.

1.3 Gliederung der Arbeit

Im zweiten Kapitel werden die Grundlagen und Technologien beschrieben, welche für die späteren Anforderungen benötigt werden. Diese Grundlagen werden im dritten Kapitel genutzt, um mittels eines Anwendungsscenarios die Anforderungen genauer zu spezifizieren und mit bereits vorhandenen Lösungen zu vergleichen. Diese Anforderungen werden dann im vierten Kapitel für die Erstellung einer grundsätzlichen Systemarchitektur genutzt. Insbesondere geht es dabei um Verfahren zur Bildmustererkennung und zur Darstellung von dreidimensionalen Objekten. Die Überlegungen und Konzepte aus den vorherigen Kapiteln werden im fünften Kapitel verwandt, um eine prototypische Anwendung zu realisieren. Anschließend wird im sechsten Kapitel das Erarbeitete nochmals zusammengefasst, bewertet und ein Ausblick in die zukünftige Entwicklung gewährt.

2 Technologien

Ein Ansatz zur Lösung der Aufgabenstellung kann die Verwendung eines Augmented Reality (kurz AR) Systems sein. In diesem Kapitel wird ein kurzer Überblick über die verwendete Technologie gegeben. Es wird erläutert, was Augmented Reality ist, welche Besonderheiten dabei optische Systeme aufweisen und welche Fehler in optischen Augmented Reality Systemen kompensiert werden müssen.

2.1 Was ist Augmented Reality?

Augmented Reality heißt - frei übersetzt - vermehrte oder auch angereicherte Realität. Es beschreibt das Verfahren Sinneswahrnehmungen (Hören, Fühlen, Sehen usw.) vom Benutzer kontextabhängig mit virtuellen Informationen anzureichern. Die Wahrnehmung des Benutzers wird so manipuliert, dass für ihn der Eindruck entsteht, als wären die virtuell vorhandenen Informationen ein Teil der realen Welt. Diese Manipulation sollte dabei in Echtzeit durchgeführt werden, um den realen Teil und virtuellen Teil zeitsynchron darstellen zu können.

Ein Objekt, zum Beispiel ein virtueller Stuhl, wird nicht wie bei einem Head-Up-Display vor das reale Bild gelegt, sondern perspektivisch korrekt in die reale Umgebung eingepasst.

Die noch in ihrem Anfangsstadium befindliche Augmented Reality Technik und ihre Anwendungen finden in der heutigen Zeit meist nur in Randbereichen und in der Forschung Anwendung.

Die ersten Anzeichen für eine erste kommerzielle Nutzung, sehen wir in der Einbettung von virtuellen Informationen in zum Beispiel Fussballliveübertragungen oder in der Einblendung von Nachrichtensprechern in virtuelle Studios. Bilder 2.1 und 2.2 [Beispiele unter ORAD]

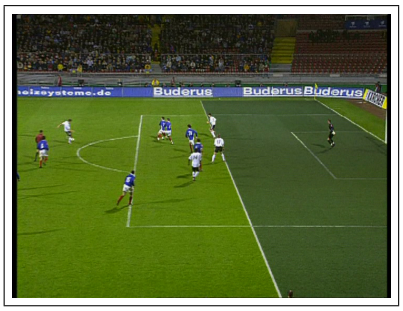

Abbildung 2.1: Abseitslinie beim Fussball

Abbildung 2.2: Nachrichtensprecher im virtuellen Studio

Jedes Augmented Reality System besteht aus folgenden Komponenten:

- Eine Komponente zur Bestimmung der Position und Ausrichtung des Benutzers und/oder der realen Objekte.
- Eine Komponente zur Positions- und Ausrichtungsabhängigen Transformation der virtuellen Information in eine für den Benutzer erfassbare Form.
- Eine Komponente zur Integration und Einblendung der transformierten Information in die reale Welt.

2.1.1 Bestimmung der Position und Ausrichtung von Objekten

Um die virtuellen Informationen in die reale Welt exakt integrieren zu können, ist es nötig, eine möglichst exakte Bestimmung der Position und Ausrichtung des Benutzers und der realen Objekte durchzuführen.

Dieser Bereich splittert sich in zwei Unterpunkte auf

Kalibrierung und Parametrisierung Es werden Werte zur Kalibrierung bestimmt, um Messfehler zu korrigieren. (Siehe auch Abschnitt 2.3)

Tracking Bestimmung der Position und Ausrichtung absolut oder relativ.

Die Kalibrierungsdaten und die Trackingdaten werden kombiniert, um eine möglichst exakte Position und Ausrichtung des Beobachters zu erhalten.

Die Pose der virtuellen Objekte wird statisch festgelegt, relativ zu den realen Objekten oder einem gemeinsamen Fixpunkt. Mit anderen Worten sie werden in einem virtuellen Raum plaziert, welcher später in das Koordinatensystem der realen Welt überführt wird.

Tracking

Das Tracking und das eigentliche Gerät - der so genannte Tracker - dienen zur eigentlichen Bestimmung der Position und der Ausrichtung eines realen Objekts im realen dreidimensionalen Raum.

Die Position und die Ausrichtung eines Objektes wird auch als **Pose** bezeichnet und kann durch einen sechs dimensionalen Vektor beschrieben werden.

Normalerweise geben die ersten drei Vekoren die Position relativ zum Uhrsprung und die letzten drei Vektoren die Ausrichtung des Objektes an.

Um die Pose zu beschreiben, gibt zwei Arten von Trackingtypen:

- Relative Tracker, welche die Positions- und Ausrichtungsveränderung in einem bestimmten Zeitintervall bestimmen. Dieses wird zum Beispiel mittels Beschleunigungsmesser erreicht. Die daraus gewonnene Information ist abhängig von der vorherigen Pose des Objektes.
- Absolute Tracker, welche die absolute Position und Ausrichtung gegenüber dem Koordinatenuhrsprung eines Objektes bestimmen.
  - Mechanische Tracker
  - GPS (Global Positioning System)
  - Tag Tracker (RFID)
  - Visuelle Tracker (Kameratracking mit oder ohne Marker)

Siehe auch [Kutak (2006)]

Ein Trackingsystem bestimmt die relative oder absolute Position des Benutzers im Weltkoordinatensystem und übergibt sie an das Augmented Reality System.

2.1.2 Transformation der virtuellen Information

Relativ zu der Pose des Benutzers muss das System die virtuelle Information in eine für den Benutzer wahrnehmbare Form transformieren.

Zum Beispiel wird ein als 3D-Gitternetz im RAM des Computers abgespeicherter Würfel in eine zweidimensionale Projektion transformiert, um sie anschließend über die reale Szene zu legen.

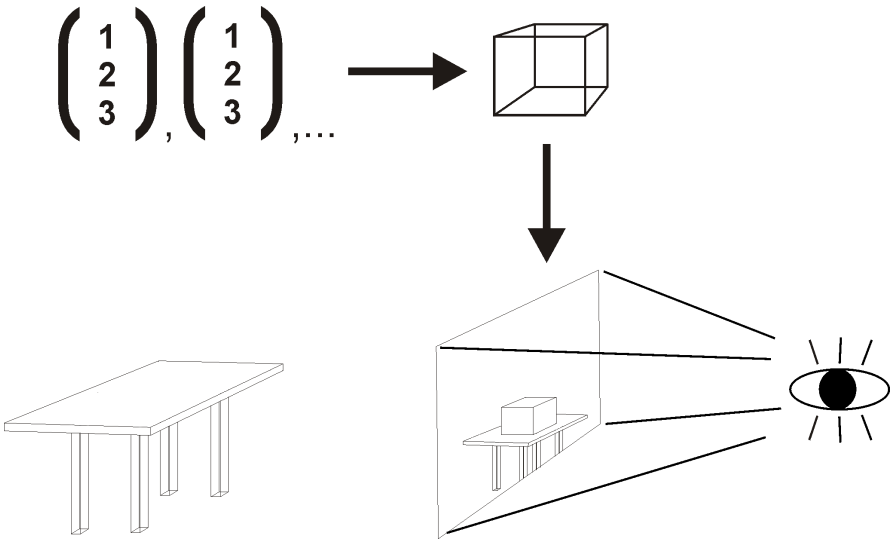

Abbildung 2.3: Datentransformation

2.1.3 Einblendung der transformierten Information in die reale Welt

Zwei unterschiedliche Varianten finden bei der Einblendung und Integration der transformierten Information in die reale Szene Anwendung.

Bei der ersten Variante wird eine komplette Sinneswahrnehmung des Benutzers aufgezeichnet. Diese Aufzeichnungen werden anschließend mit der transformierten Information angereichert oder auch überblendet. Dem Benutzer wird nun diese veränderte Wahrnehmung als seine eigene präsentiert und die Ursprungswahrnehmung gänzlich unterdrückt. Ein Beispiel soll die Integration einer Stimme aus dem Computer sein, welche sich so anhören soll, als würde sie aus einer bestimmten Richtung kommen. Dafür wird mittels zweier

Mikrofone die umliegende Geräuschkulisse aufgenommen und mit der in der Lautstärke abhängig von der Entfernung eingestellten digitalen Stimme angereichert. Diese veränderte Realität wird dann dem Benutzer auf zwei Kopfhörer ausgegeben. Er hört nur die veränderte Realität aus den Kopfhörern, die Originalgeräusche werden komplett abgeschirmt.

Die zweite Variante blendet die virtuelle Information in die reale Welt nur zusätzlich ein. Im Gegensatz zum vorherigen Beispiel, würde hier die virtuelle Stimme nur über Lautsprecher ausgegeben und so die Wahrnehmung der realen Welt nicht ersetzt, sondern nur überlagert.

Vorteil der ersten Variante ist, dass Veränderungen wesentlich einfacher durchgeführt werden können. Es können leicht Elemente hinzugefügt oder auch entfernt werden. Nachteil ist, dass die Aufzeichnung und Wiedergabe der Realität nur mit Qualitätsverlust durchgeführt werden kann und so die Bandbreite der Sinnesorgane nicht vollständig ausgeschöpft werden. Die zweite Variante hat den Vorteil, dass die real vorhandenen Elemente unverfälscht und ohne Qualitätsverlust beim Benutzer ankommen. Der Nachteil ist, dass die Realität mit den virtuellen Elementen nur überlagert und nicht ersetzt werden kann.

Ein weiteres Beispiel und weitere Informationen zu diesem Thema finden sie im folgenden Abschnitt 2.2 Besonderheiten von optischen Augmented Reality Systemen

2.2 Besonderheiten von optischen Augmented Reality Systemen

Der folgende Abschnitt soll näher auf optische Augmented Reality Systeme eingehen. Insbesondere auf die beiden Verfahren zur Einblendung der virtuellen Information in die reale Szene.

Der Bereich der Augmented Reality, welcher am meisten Anwendung findet, ist dass visuelle oder auch optische Augmented Reality, welches nur die Sehwahrnehmung mit virtuellen Informationen anreichert und/oder verändert.

Bei der Augmentierung optischer Informationen gibt es grundsätzlich zwei Ansätze:

Video-based see-through

1. Eine Kamera nimmt ein Bild auf
2. Das Bild wird positions- und ausrichtungsabhängig mit virtuellen Informationen angereichert.
3. Das Bild wird auf einem Monitor oder auf einem Head-Mounted-Display komplett ausgegeben.

Der Benutzer bekommt nur das Videobild zu sehen, alle realen Sehwahrnehmungen werden abgeschirmt.

Optical see-through

1. Zur Position und Ausrichtung passend, wird ein die virtuellen Informationen enthaltenes zweidimensionales Bild erstellt. Alle nicht veränderten Bereiche sind dabei transparent.
2. Das berechnete Bild wird beispielsweise mittels eines halbdurchsichtigen Head-Mounted-Displays über die reale Szene gelegt.

Die Vorteile des Video-based see-through Verfahrens liegen in der zeitlichen Übereinstimmung von realen und virtuellen Objekten, der leichten Positionierung und Integrierung der virtuellen Objekte in das reale Bild. Nachteile sind die begrenzte Auflösung der realen Welt, die zeitliche Verzögerung zwischen Erfassen und Darstellen des realen Bildes und der daraus resultierenden Simulationskrankheit.

Beim Optical see-through Verfahren liegen die Vorteile bei der Anzeige der realen Welt in Echtzeit und der unbegrenzten Auflösung der Realwelt. Nachteilig ist, dass die virtuellen Objekte immer noch zeitlich verzögert dargestellt werden und so keine Garantie für eine zeitliche Übereinstimmung realer und virtueller Objekte gegeben werden kann. Das halbdurchsichtige Display sorgt zusätzlich dafür, dass die reale Welt nur abgedunkelt und die virtuellen Objekte nur transparent dargestellt werden.

Siehe auch [Azuma (1997)]

2.3 Fehler und Fehlerquellen bei optischen Augmented Reality Systemen

Der folgende Abschnitt beschreibt die Fehler und Fehlerquellen welche speziell bei optischen Augmented Reality Systemen entstehen und die durch Ein-

satz von Verfahren und Techniken kompensiert werden müssen. Die Fehler werden hier anhand von optischen Augmented Reality Systeme (kurz ARS) dargelegt, können aber zum größten Teil auf alle anderen ARS übertragen werden.

Um virtuelle Objekte in die reale Landschaft zu integrieren, benötigt das AR - System exakte Position und Ausrichtung (genannt Pose, siehe Kapitel 2.1.1) der realen bzw. virtuellen Objekte und des Betrachters relativ zueinander oder zu einem gemeinsamen Fixpunkt. Bei der Erfassung der Pose können eine Reihe von Fehlern entstehen. Ronald T. Azuma beschreibt dieses grundlegende Problem sehr genau. [Azuma (1997)] Er unterteilt die bei der Erfassung (Registation) der Pose entstehenden Fehler in zwei Kategorien. In Statische und in dynamische Fehler. Statische Fehler sind optische Verzerrungen/Täuschungen, Fehler im Tracking System, mechanische Kalibrierungsfehler oder auch falsche Parametrisierung (z.B. Abstand Monitor-Auge oder Abstand Kamera-Auge).

Diese Fehler werden meist durch gute Kalibrierung kompensiert. Zum Beispiel können die richtigen Parameter durch zusätzliche Sensoren oder durch Referenzmessungen bestimmt oder angenähert werden.

Dynamische Fehler entstehen durch Verzögerungen bei der Verarbeitung. Azuma spricht von einem Ende-zu-Ende Fehler. [Azuma (1997)] Welchen er definiert als die Zeitdifferenz zwischen dem Erfassen der Pose und dem Anzeigen des errechneten Resultates. Durch einen hohen Ende-zu-Ende Fehler kann es zu einer fehlerhaften Anzeige der virtuellen Inhalte kommen. Wenn beispielsweise ein Benutzer ein optisches Augmented Reality System mit Optical-see-through Head-Mounted-Displays (kurz HMD) verwendet und die Verarbeitungszeit zu hoch ist, zieht bei Bewegung des HMD das virtuelle Objekt scheinbar nach. Obwohl bereits ein neues reales Bild vorliegt, wird das virtuelle Objekt noch an der alten Position angezeigt. Bild 2.4

Abbildung 2.4: Ende-zu-Ende Fehler

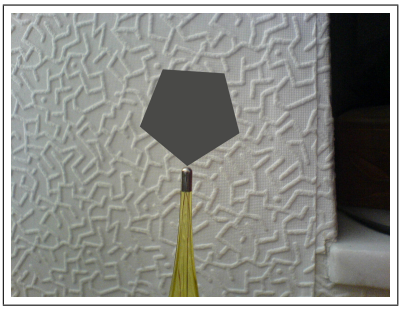

Abbildung 2.5: Ruhezustand

Maßnahmen, diesem Fehler entgegen zu wirken, sind die Reduzierung der benötigten Verarbeitungszeit und/oder das Vorausberechnen der zukünftigen Position und Ausrichtung, um die Darstellung des virtuellen Objekts bereits für die zukünftige Pose zu berechnen.

Dynamische Fehler werden nur wahrgenommen, wenn sich die Pose des Betrachters relativ zur Pose der virtuellen Objekte verändert, er zum Beispiel den Kopf bewegt. Ansonsten sind das alte und das neu berechnete Bild identisch. Da keine Veränderung stattfand, wird der Fehler vom Benutzer dann auch nicht wahr genommen. Bild 2.5

3 Analyse

Wie in der Einführung bereits beschrieben, geht es um die Entwicklung eines Augmented Reality Systems mittels visueller Techniken. Das ganze soll am Beispiel eines Museumsinformationssystems spezifiziert werden. Im folgenden Kapitel wird eine erste Analyse der Problemstellung durchgeführt. Es wird das Anwendungsszenario näher spezifiziert und die Anforderungen des Systems festgelegt. Zuletzt wird ein kurzer Überblick über bereits vorhandene System- oder Komponentenlösungen gegeben und mit den Anforderungen der Aufgabe verglichen.

3.1 Anwendungsszenario

Anwendungszenario ist ein fiktives Kunstmuseum. In diesem Museum werden Gemälde ausgestellt. Diese Bilder hängen, fest montiert, an den Wänden und zeigen beispielhaft klassische Motive wie Portraits oder Landschaften. Der Wandhintergrund besitzt eine durchgängige neutrale Farbe - wie weiß oder dunkelbraun.

Benutzer des Systems ist Max Jedermann, ein aufgeschlossener Mensch, welcher sich auch nicht scheut, futuristisch aussehende Geräte aufzusetzen.

Herr Max Mustermann möchte sich gerne vor ein Gemälde stellen und erweiterte Informationen über dieses Erhalten. Die Informationen sind optische Objekte wie Texte oder Figuren. Dargestellt werden sollen die Informationen so, als wären sie ein Teil der realen Welt.

3.2 Anforderung

Das System soll die Position und Blickrichtung des Benutzers relativ zu den Gemälden bestimmen. Anschließend soll es ein vorher definiertes virtuelles Objekt - eingebettet in den realen Raum - in das Sichtfeld des Anwenders projizieren. Das virtuelle Objekt soll möglichst von keinem anderen Benutzer wahrgenommen werden. Der Benutzer soll nun die Möglichkeit bekommen, mittels Handbewegung das virtuelle Objekt zu aktivieren und so mit dem System zu interagieren um beispielsweise weitere Informationen angezeigt zu bekommen.

Alle verwendeten Komponenten sollen möglichst kosten- und lizenzfrei erhältlich sein. Die Effizienz und Performanz des Systems spielt nur eine untergeordnete Rolle.

Wichtigste Anforderung ist, dass das Entwickeln einer Lösung den vorgegebenen Zeitraum nicht überschreitet.

3.3 Vergleich mit vorhandenen Lösungen

Im Bereich visueller Augmented Reality gibt es bereits eine Vielzahl von Lösungen und Frameworks zu den unterschiedlichsten Aufgabenbereichen.

Im Folgenden ist ein Auszug der gängigen Systeme aufgelistet.

Mobile Augmented Reality Quest [MARQ] ist ein auf einem PDA laufender elektronischer Museumsführer, welcher abhängig von der Position des Benutzers auf dem PDA in die Realität eingebettete Informationen anzeigt. Jedoch wird das Bild auf einem PDA angezeigt und besitzt so nur ein eingeschränktes Sichtfeld.

Studierstube [StbAPI] ist ein Framework basierend auf C++. Sie enthält einen großen Klassensatz zum Tracking und Anzeigen von 3D-Objekten. Das Framework wird ständig von Professoren bzw. Studenten weiterentwickelt und ist deshalb schon relativ komplex.

ARToolKit [ARToolKit] ein einfaches OpenSource Framework auf C++ Basis zur visuellen Positionsbestimmung mittels Marker und Anzeige einfacher dreidimensionaler Objekte in Realzeit.

Unifeye [metaioGmbH] ist ein kommerzielles Marker- oder Szenenbasiertes Komplettsystem für den professionellen Einsatz der Firma Metaio GmbH. Die Software wird in eigene Programme als Windows COM-Objekt eingebunden und mittels VRML werden virtuelle Objekte definiert und angezeigt.

Tinmith [Tinmith] ein System von der Universität von Süd Australien mit dem Focus auf mobile outdoor Augmented Reality. Laut Martin Wagner [Wagner (2005)] eines der effizientesten Systeme. Viele Funktionen zur bidirektionalen Interaktion zwischen Mensch und Maschine mittels Markerhandschuh. Die Software ist nicht frei erhältlich.

Das Forschungsprojekt Distributed Wearable Augmented Reality Framework [Navab] ein auf CORBA basiertes Framework der Universität München. Teile basieren auf dem OpenSource Framework ARToolKit. Sehr komplex und sehr flexibel gehaltenes System. Nicht frei erhältlich. Alle hier vorgestellten

Name	Framework	Sprache	Outdoor	Lizenz	Erhältlich
MARQ	Nein		Nein	Nein	Ja
Studierstube	Ja	C++	Ja	Nein	Nein
ARToolKit	Ja	C++	Nein	OpenSource	Ja
Unifeye	Ja	COM-Obj.	Ja	Kommerziell	Ja
Tinmith	Ja	C	Ja	GPL	Nein
DWARF	Ja	CORBA	Nein	OpenSource	Ja

Tabelle 3.1: Vergleich vorhandener Systeme

Produkte und Frameworks erfüllen alle technischen Anforderungen. Sie können mittels unterschiedlicher Techniken wie Marker- oder Szenenerkennung die Position des Benutzers bestimmen und frei definierbare virtuelle Objekte auf ein Head-Mounted-Display projizieren. Die Softwaresysteme sind jedoch teilweise in ihrer Einarbeitung zu komplex, nicht erhältlich oder an ein freies oder kommerzielles Lizenzmodell gebunden, wodurch sie für diese Aufgabenstellung nicht geeignet sind.

Aus diesem Grund muss ein eigenes Softwaresystem entwickelt werden, welches schnell zu realisieren ist und alle für diese Aufgaben gestellten Anforderungen erfüllt.

4 Design

In diesem Kapitel geht es um den grundsätzlichen Aufbau des Augmented Reality Systems. Anhand der Hard- und Softwarearchitektur werden die Anforderungen weiter spezifiziert. Dann wird gezeigt, wie eine effiziente Bildmustererkennung und Koordinatenbestimmung durchgeführt wird. Im vorletzten Abschnitt werden Verfahren vorgestellt, wie dreidimensionale virtuelle Objekte abhängig von den vorher bestimmten Koordinaten auf ein zweidimensionales Bild projiziert werden. Zum Schluß wird die Grundlage zur Erkennung von Handbewegungen mittels einer Kamera geschaffen und die Anforderungen weiter spezifiziert.

4.1 Architektur

In diesem Abschnitt geht es um die Architektur des Augmented Reality Systems. Hier werden die in der Analyse aufgestellten Anforderungen näher spezifiziert.

4.1.1 Hardwarearchitektur

Die Hardwarearchitektur wird einfach gehalten, eine Farbkamera zur Aufnahme, ein Display zur Ausgabe und ein Standard-Laptop zur Verarbeitung. Wichtig hierbei ist, dass alle Komponenten möglichst den Standards für wearable Computing entsprechen sollten. Siehe Arbeit von Mirco Gerling [Gerling (2006)]

- Sie müssen leicht sein.
- Sie müssen gut am Körper befestigt werden können und bequem sitzen.

- Alle Komponenten müssen kabellos mit Energie versorgt werden und einen geringen Stromverbrauch haben, um hohe Akkulaufzeiten gewährleisten zu können.
- Abwärme muss leicht abgeführt werden können.
- Sie müssen robust sein und leichte Erschütterungen aushalten.
- Alle Komponenten sollten möglichst klein sein und sich in die normale Kleidung integrieren lassen.

Das Display und die Kamera sollten räumlich möglichst nah bei einander liegen, um Anzeigefehler zu kompensieren.

Aufgrund des hohen Rechenaufwandes bei Bildverarbeitung sollten alle Komponenten, insbesondere die Rechnerleistung des Laptops, eine ausreichende Performance zur Verfügung stellen.

Nach Möglichkeit sollte die Hardware den Betrieb von Echtzeitanwendungen unterstützen, um eine konstante Verarbeitung und dadurch eine konstante Bildwiederholrate zu erreichen.

4.1.2 Softwarearchitektur

In diesem Abschnitt geht es um die Softwarearchitektur unseres Systems.

Bei Augmented Reality Systemen ist eine schnelle Verarbeitung eines der wichtigsten Kriterien, während bei der Manipulation der Realität bei zum Beispiel Filmen mehrere Monate für die Postproduktion zur Verfügung stehen, (Bild 4.1) haben AR - Systeme nur wenige hundertstel Millisekunden Zeit in der sie das aufgenommene Bild mit den virtuellen Inhalten anreichern und anzeigen müssen.

Das menschliche Auge nimmt Bilderfolgen mit einer Geschwindigkeit über 16 Bilder pro Sekunde als Bewegung war. Um den Ende-zu-Ende Fehler (siehe Abschnitt 2.3) auf einen einigermaßen guten Wert zu verringern ist eine Bildfolge von 24 oder mehr Bildern pro Sekunde erforderlich. Daraus ergibt sich, dass ein optimales AR - System für die Erfassung des Bildes, die Anreicherung und die Darstellung des neuen Bildes zusammen ungefähr 42 Millisekunden zur Verfügung hat. Bei der Software und ihrer Architektur sollte es sich also um ein Echtzeitsystem handeln, welches für die nötigen Arbeitsabschnitte eine vorgegebene Zeit nicht überschreitet. Wobei mit einfachen Mitteln das

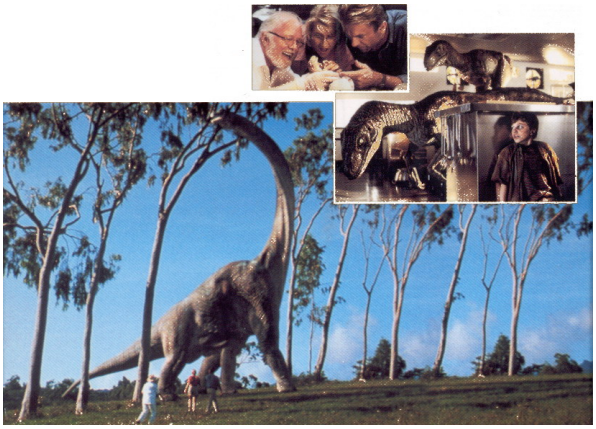

Abbildung 4.1: Ausschnitt aus dem Film Jurassic Park, Universal Studios 1993

Erreichen einer Zeit von 42 Millisekunden fast unmöglich ist und deshalb bei den meisten Systemen ein Wert von minimal 100 Millisekunden als akzeptabel gilt.

Bei dem benötigten Betriebssystem sollte es sich deshalb um ein Betriebssystem für Echtzeitanwendungen handeln. (z.B. QNX [QNX])

Auch die nötige Softwareumgebung sollte die Anforderungen an ein Echtzeitsystem erfüllen, um eine konstante Bildverarbeitung gewährleisten zu können.

Die für die Lösung der Aufgabe benötigte Software besteht aus drei Komponenten:

- Eine Komponente zur Steuerung der Kamera
- Eine Komponente zur Speicherung der Markerbeschreibung und der virtuellen Elemente.
- Eine Komponente zur Verarbeitung und Steuerung.

Die dritte Komponente unterteilt sich wiederum in eine Komponente zur Bildanalyse und eine zur Anreicherung und Darstellung des Bildes.

4.2 Bildmustererkennung

Um in Erfahrung zu bringen, ob der Benutzer vor einem Gemälde steht und wie seine Position relativ zu dem Bild ist, bedienen wir uns der Technik der Bildmustererkennung (Auch RobotVision genannt).

Für das Augmented Reality System nutzen wir ein optisches Bilderkennungsverfahren mit zweifarbigen Markern. Marker sind vorher fest definierte geometrische Formen, welche eindeutig erkannt und zugeordnet werden können.

Eine Alternative wäre ein objektorientiertes Bilderkennungssystem, welches die eigentlichen Objekte (in diesem Fall die Gemälde) anhand markanter Punkte erkennt. Dieses Verfahren ist recht komplex und aufwendig zu realisieren.

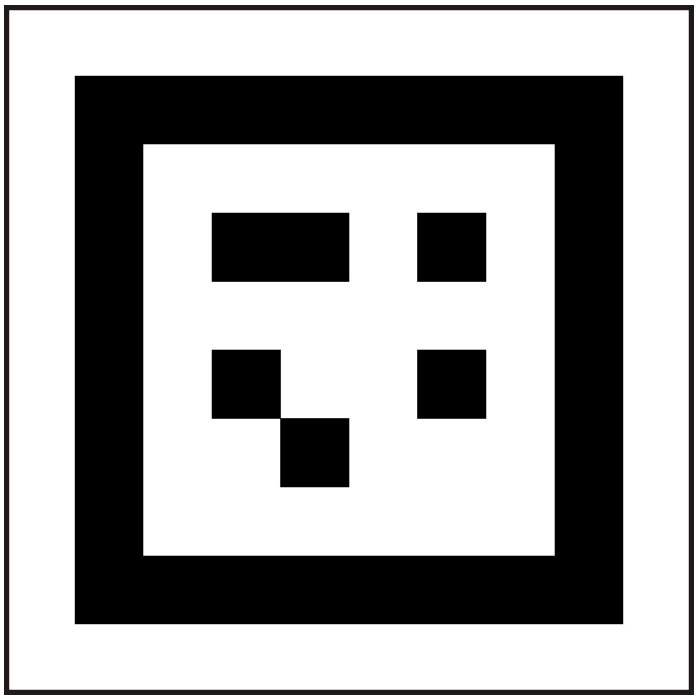

Abbildung 4.2: Beispiel für einen Marker

Auf dem Bild 4.2 sehen sie ein Beispiel für einen solchen Marker. Er besteht aus einer schwarzen, quadratischen Umrandung. Diese dient dazu, dass der Marker sich von anderen Objekten einfach abgrenzt. Innerhalb liegt auf weißem Grund ein sechs mal sechs Raster von kleinen schwarzen und weißen Quadraten, welche den Marker eindeutig identifizieren. Um den Marker

und seine Position auf einem Bild mit anderen Objekten erkennen zu können, werden verschiedene Verfahren aus dem Bereich der Bildmustererkennung eingesetzt.

Bildmustererkennung ist laut Metagis [Metagis], die flexible Interpretation und Verarbeitung von Bildern im Computer mit Grau- und/oder Multispektral - (Farb-) Werten.

Eine Bildmustererkennung wird meistens als Kettenverarbeitung realisiert (Siehe Bild 4.3) und unterteilt sich in folgende Abschnitte:

- Digitalisierung
- Vorverarbeitung
- Segmentierung
- Merkmalsextraktion
- Klassifikation

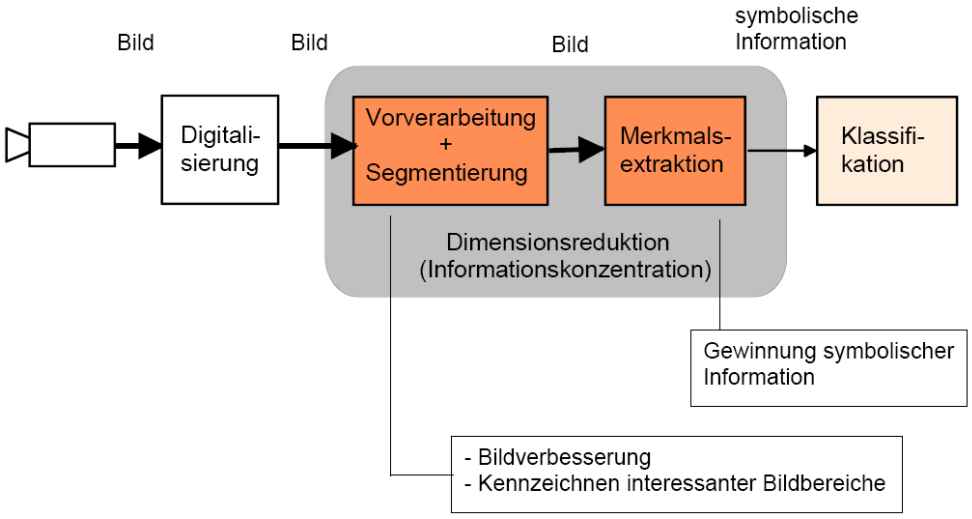

Abbildung 4.3: Verarbeitungskette

4.2.1 Digitalisierung

Das reale Bild wird mittels eines Analog-Digitalwandlers (z.B. einer Farbkamera) in ein digitales Raster umgewandelt. Dabei kann schon, z.B. durch Auswahl der Brennweite oder Beleuchtungsstärke eine Vorverarbeitung stattfinden.

4.2.2 Vorverarbeitung

Bei der Vorverarbeitung wird mittels mehrerer digitaler Filter versucht, die Qualität des Bildes zu verbessern. Zusätzlich werden die Eingangsdaten auf relevante Informationen reduziert, um eine schnellere Verarbeitung zu gewährleisten. Wichtige Informationen ,wie zum Beispiel Kanten, werden hervorgehoben und weniger wichtige Informationen oder Bildfehler werden unterdrückt.

Abbildung 4.4: Orginalaufnahme

Im ersten Schritt wird das Bild mit einer Farbtiefe von 256 auf ein Bild mit einer Farbtiefe von 3 Farben reduziert. Wobei alle Farben oberhalb eines Grenzwerts auf Weiß, unterhalb eines Grenzwertes auf Schwarz und dazwischen auf einen transparenten Wert gesetzt werden. Transparent bedeutet in diesem Fall, dass diese Pixel nicht weiter betrachtet werden müssen. Zur besseren Darstellung wurden die Pixel auf den folgenden Bildern anders eingefärbt Weiß=Schwarz, Schwarz=Hellgrau, Transparent=Dunkelgrau (Bild 4.5)

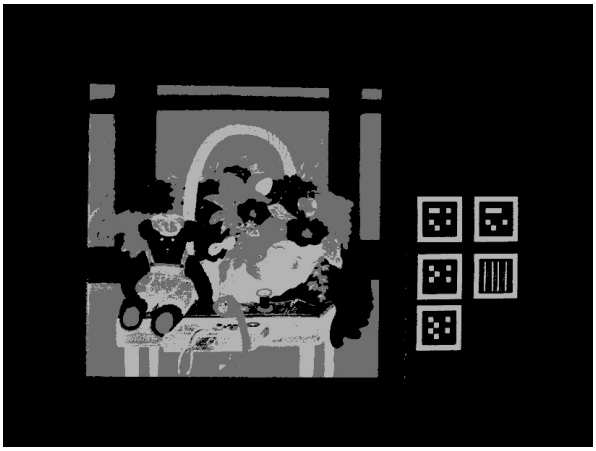

Abbildung 4.5: Binarisiertes Bild

Im zweiten Schritt werden alle schwarzen Flächen welche an den Bildrand angrenzen, in die Transparenzfarbe eingefärbt. Dadurch kann sichergestellt werden, dass nur Marker, welche vollständig zu sehen sind, ausgewertet werden.

In der dritten Stufe wird das bei einfachen Kameras entstehende Bildrauschen entfernt. Dafür werden alle schwarzen und weißen Pixel transparent gefärbt, welche im Bild alleine stehen. Dadurch werden nur Objekte erkannt, welche größer als ein Pixel sind. (Bild 4.6)

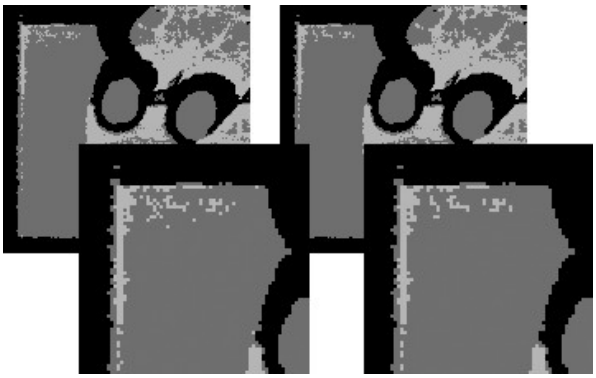

Abbildung 4.6: Anwendung der Rauschunterdrückung

4.2.3 Segmentierung

Da sich mehrere Objekte auf einem Bild befinden können, müssen die einzelnen Objekte von einander getrennt, also segmentiert werden. Für die Objekterkennung sind in diesem Fall nur die durch schwarze Flächen eingeschlossenen Bereiche interessant. Um ein Bild zu segmentieren, gibt es mehrere Verfahren.

Connected Components Labeling

Connected Components Labeling ist ein Verfahren zur Bestimmung einzelner zusammenhängender Flächen in einem Binärbild. Von oben links beginnend wird zeilenweise der nächste schwarze Pixel gesucht. Wenn der direkt über diesem liegenden Pixel bereits einer Klasse zugeordnet wurde, bekommt dieser Pixel die selbe Klassennummer, wenn der direkt über ihm liegende Pixel keiner Klasse zugeordnet wurde bekommt der Pixel eine eigene neue Klassennummer. Danach wird mit dem nächsten schwarzen Pixel weitergemacht. Wenn zwei Klassen direkt nebeneinander liegen, werden sie als Äquivalent eingetragen. Die einzelnen Klassen und ihre Äquivalenz werden in eine Matrix eingetragen und minimiert. Das daraus entstandene Bild besteht nun aus einzelnen Flächensegmenten, welches jeweils einer Klasse zugeordnet wurde.

Geradenextraktion durch Houghtransformation

Das Houghverfahren ist eine effektive Methode um Geraden oder Kreise aus einem Bild zu extrahieren. Dabei wird mittels zum Beispiel des Sobel - Kantenfilters (Siehe Meisel [Meisel (2005)]) alle Objektkanten im Bild hervorgehoben. Anschließend werden für jeden Kantenpunkt im Bild alle möglichen Geraden, welchen durch diesen Punkt laufen in einen quantisierten Houghraum geschrieben bzw. addiert. Alle Felder im Houghraum mit einem hohen Wert sind Geraden, welche durch eine große Anzahl von Kantenpunkten im Bild laufen. Solche gefundenen Geraden sind die Geraden von Objektkanten.

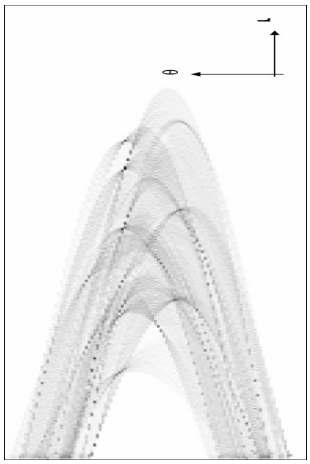

Abbildung 4.7: Houghraum

Konturextraktion mittels Pavlidis Algorithmus

Der Pavlidis Algorithmus ist ein Verfahren zur Bestimmung von Objektaußenkanten in einem Binärbild. Der Algorithmus fährt von links oben beginnend das Bild zeilenweise ab, bis er auf ein Objekt trifft. Von da an merkt sich der Algorithmus jeden weiteren Bildpunkt und läuft weiter an der Objektkante entlang, bis er seinen Ursprungsauftreffpunkt wieder erreicht hat. Er umkreist quasi das Objekt und markiert anschließend den von ihm umschlossenen Bereich. Dadurch wird nicht nur die schwarze Fläche markiert, sondern auch gegebenenfalls der von der schwarzen Fläche eingeschlossene Bereich. Zusätzlich bekommt man durch dieses Verfahren eine Liste mit allen Pixeln, welche am Rand der Segmentkontur liegen. Alle von diesen Pixeln umschlossenen Bildpunkte sind Teil des jeweiligen Segmentes.(Siehe Bild 4.8)

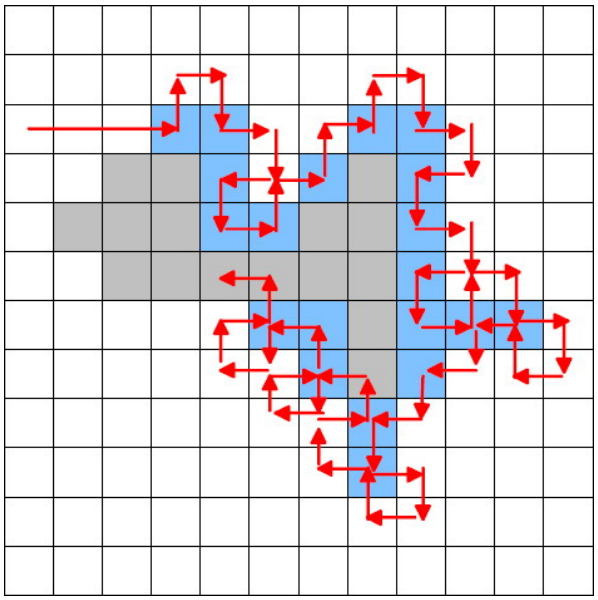

Abbildung 4.8: Beispiel für Kantenfindung mittels Pavlidis - Algorithmus

Der Pavlidis Algorithmus ist für unsere Aufgabe am besten geeignet, da er auch den von der schwarzen Fläche eingeschlossenen Bereich mit als Segment markiert. Dieser Bereich kann dann anschließend wieder Segmentiert werden. Zusätzlich liefert uns der Algorithmus eine Liste mit allen Pixeln, welche an der Außenkante des Objekts liegen. Mit dieser Liste kann bei der Merkmalsextraktion weitergearbeitet werden.

4.2.4 Merkmalsextraktion

Die einzelnen Segmente sind immer noch zu komplex, um entscheiden zu können, ob es sich hierbei um ein gesuchtes Objekt handelt. Deshalb werden nun aus den Segmenten typische Merkmale extrahiert.

In einigen Fällen ist das Segment selbst schon das zur Klassifikation nötige Merkmal. Meist jedoch müssen die nötigen Merkmale wie Objektgeraden, Schnittpunkte oder Fixpunkte noch genauer bestimmt werden.

Eine einfache Art ist die Zerlegung der Objektkontur in ein Polygonobjekt mittels Polygonapproximation. Dabei wird ein Polygon zwischen zwei Konturpunkten aufgespannt. Für jede Polygonkante wird jetzt der am weitesten entfernte Konturpunkt P zwischen Startkonturpunkt A und Endkonturpunkt B der Kante gesucht. Anschließend wird die ursprüngliche Polygonkante durch zwei neue Kanten ersetzt. Die erste Kante hat ihren Ursprung im vorherigen Startpunkt A und ihr Ziel im gefundenen Konturpunkt P. Die zweite Kante hat ihren Ursprung im Konturpunkt P und ihren Zielpunkt im Konturpunkt B. Dieses Verfahren wird mit allen Polygonkanten solange wiederholt, bis der einzelne Fehler einen bestimmten Schwellwert nicht mehr überschreitet. Das daraus entstandene Polygon ist eine Annäherung an die Objektkontur und kann gegebenenfalls mit Referenzpolygonen verglichen werden.

Ein weiteres sehr einfaches und für unseren Marker geeignetes Verfahren ist, die vier Konturpunkte mit dem geringsten Abstand zu den einzelnen Eckpunkten zu finden, von diesen Punkten aus ein perspektivisch angepasstes Netz über das Objekt zu legen und die Farbe der Pixel an den einzelnen Messpunkten in einem Messraster abzuspeichern. Zu beachten ist hierbei, dass das Segment eine gewisse Mindestgröße haben muss, um Rundungs- und Quantisierungsfehler zu vermeiden.

Andere Verfahren zur Bestimmung relevanter Merkmale sind die Fourierdeskriptoren oder die Extraktion der geometrischen Momente. (Siehe [Meisel (2005)])

Das für unsere Aufgabe beste Verfahren - welches auch in der Realisierung Anwendung findet - ist die Merkmalsextraktion mittels Messnetz. Dieses Verfahren ist schnell, einfach zu realisieren und robust gegenüber einfachen Bildstörungen.

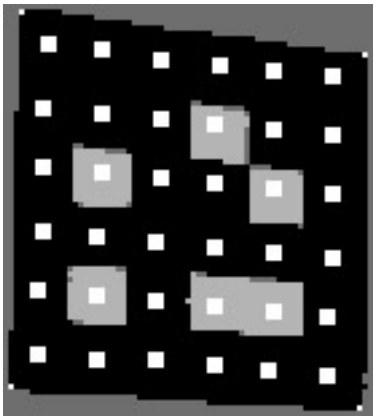

Abbildung 4.9: Merkmalextraktion mittels Messnetz

4.2.5 Klassifikation

Die extrahierten Merkmale der einzelnen Segmente müssen nun einer vorgegebenen Klasse zugeordnet werden.

Ein einfaches Beispiel ist die Klassifikation von Mengen.

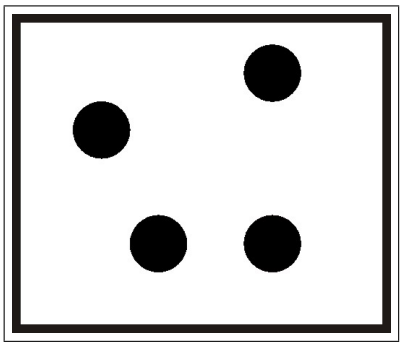

Abbildung 4.10: Kasten Nr.1 mit vier Kreisen

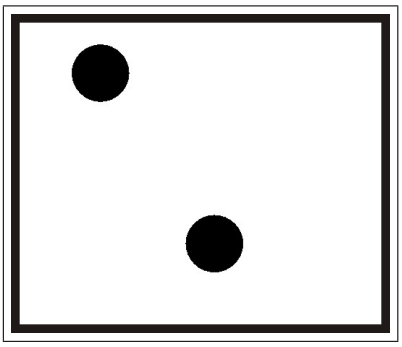

Abbildung 4.11: Kasten Nr.2 mit zwei Kreisen

In einem Kasten sind eine bestimmte Anzahl von Kreisen. Jeder Kasten ist ein Segment. Das zu extrahierende Merkmal ist die Anzahl von Kreisen innerhalb eines Kastens.

Der so genannte Klassifikator enthält nun die einzelnen möglichen Klassen und ihre dazugehörige Bedingung.

Kasten 1 wird also der Klasse 3 und Kasten 2 der Klasse 2 zugeordnet.

Klasse	Merkmale
Klasse 1	Ein Kreis
Klasse 2	Zwei Kreise
Klasse 3	Vier Kreise
Klasse 4	Fünf Kreise
Klasse 5	Alles Andere

Tabelle 4.1: Klassifikatortabelle

Dadurch wurden die beiden Objekte eindeutig zugeordnet.

Dasselbe kann auch mit den Markern gemacht werden, ein Segment wird einer Markerklasse zugeordnet wenn das extrahierte Messraster mit dem Messraster des Markers übereinstimmt.

Eine alternative Technik der Klassifikation ist die Arbeit mit neuronalen Netzen. Dabei werden die gesamten erfassten Merkmale in ein vorher trainiertes neuronales Netz gespeist, Das neuronale Netz führt dann selbstständig die Klassifizierung durch. Das Verfahren ist sehr flexibel und in Bereichen mit vielen Merkmalen sehr effizient. Jedoch ist die Verarbeitung mittels neuronaler Netze teilweise sehr rechen aufwendig und erfordert vorgegebene Trainingsmuster.

Wir beschränken uns bei der Lösung der Aufgabe auf die Klassifikation mittels Markerklassen und Messraster. Die Arbeit mit neuronalen Netzen wäre für unsere Aufgabe zu aufwendig.

4.3 Darstellen von Objekten

Die virtuellen Objekte müssen dem Anwender in irgendeiner Form angezeigt werden. Im folgenden Abschnitt geht es um die eingebettete Darstellung von virtuellen Objekten in die Realität. Es werden verschiedene technische Geräte zur Darstellung vorgestellt und Verfahren zur Transformation von dreidimensionalen Objekten auf eine zweidimensionale Projektionsfläche präsentiert.

4.3.1 Monitore und Displays

Stationäre Displays zur Darstellung gibt es in vielen verschiedenen Varianten. Vom einfachen TFT-Bildschirm, über Rückprojektionsbildschirme bis zu leistungsstarken Beamern. Jedoch sind diese nur eingeschränkt für diese Problemstellung geeignet.
Zum einen, weil sie sich bedingt durch ihre Größe nur schlecht vom Benutzer transportieren lassen. Und zum anderen weil die Darstellung von jedem in der Umgebung wahrgenommen werden kann. Jeder sieht die auf dem Monitor aufgerufenen Informationen. Dieses könnte zum Beispiel in einem Museum für andere Besucher als sehr störend empfunden werden. Daher benötigt man eine kleinere und unauffälligere Lösung. Eine Alternative sind Head Mounted Displays (HMD). Dieses sind Monitore welche unmittelbar vor den Augen sitzen oder kleine TFT-Displays die ihr Bild auf im 45 Grad Winkel vor dem Auge angeordnete Spiegel projizieren. Das Display bzw. die Spiegel können dabei undurchsichtig oder halb transparent sein. (Siehe Abschnitt 2.2) Zum Zubehör von Modernen HMD's gehören eine bereits in die Brille eingebaute Kamera und passende Kopfhörer. Der Nachteil der HMD's ist, dass sie das Sichtfeld des Benutzers zum Teil stark einschränken, die Auflösung der TFT-Bildschirme noch sehr gering ist und sie bei guter Qualität einen sehr hohen Preis haben. Jedoch sind die Vorteile dieser Geräte die geringe Größe und das relativ geringe Gewicht. Dadurch können Moderne HMD's fast wie normale Brillen aufgesetzt und einfach mitgenommen werden ohne den Benutzer in seiner Bewegungsfreiheit einzuschränken. Zusätzlich kommt hinzu, dass das Bild, welches dargestellt wird, nur vom jeweiligen Benutzer wahrgenommen wird. So werden zum Beispiel andere Museumsbesucher bei ihrem Rundgang nicht gestört während man selber weitere Informationen zu einem Exponat angezeigt bekommt. Die einzelnen HMD's unterscheiden sich im Wesentlichen durch ihre Baugröße, Netzanbindung und durch die darzustellende Auflösung.

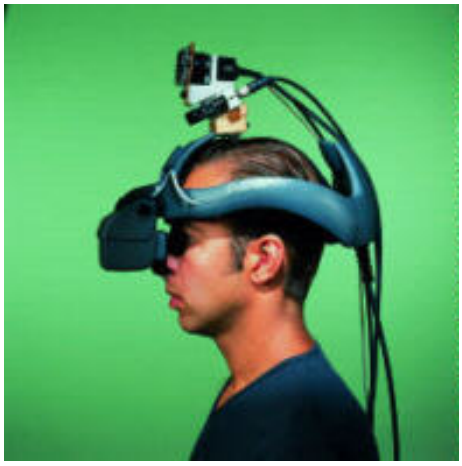

Abbildung 4.12: Beispiel eines HMD's mit Kamera

Abbildung 4.13: See - through Display beim US Militär

Um die Abweichung zwischen Bilderfassung und Darstellung des Bildes möglichst gering zu halten, wird die Kamera bei modernen Geräten möglichst nah neben oder vor das HMD montiert.

Eine zurzeit sich in der Entwicklung befindliche Technik ist ein Anzeigeverfahren mittels Laser, dabei wird das Bild mittels eines schwachen Lasers direkt auf die Netzhaut des Benutzers projiziert.

4.3.2 Bildtransformation

In diesem Abschnitt wird auf die Projektion von dreidimensionalen Objekten auf zweidimensionalen Anzeigeebenen eingegangen. Es wird erklärt wie diese Projektion durchgeführt wird und welche Vorraussetzungen dafür erfüllt sein müssen.

Um die dreidimensionalen Objekte auf dem Monitor oder einem HMD, also einer zweidimensionalen Fläche anzeigen zu können, müssen die Objekte erst vom dreidimensionalen Raum der virtuellen Welt auf den zweidimensionalen Raum des Monitordisplays umgerechnet werden. Bei solch einem Verfahren spricht man von einer Projektion. Es gibt mehrere Projektionsarten wie Parallelprojektion, Isometrische Projektion oder auch Dreitafelprojektion. Diese Projektionsarten werden meist in der Technik eingesetzt, um Gebäude- und Maschinenpläne zu erstellen und sind für diesen Fall nicht geeignet. Die Projektion, welche an die Sichtweise des Menschen am nähsten herankommt, ist die Zentralprojektion. Hierbei wird - von einem Augenpunkt ausgehend - ein Strahl zu jedem Punkt des 3D-Objektes gezogen. Dazwischen wird eine 2D-Bildebene gelegt. Die Schnittpunkte der einzelnen Strahlen mit der Bildebene sind dann die projizierten Punkte auf der Bildebene. (Siehe 4.14)

Wenn P1 = (x,y,z) der zu projizierende Punkt im Raum ist und α bei $\alpha ! = 0$ die kürzeste Entfernung vom Augpunkt zur Bildebene darstellt, ergibt sich

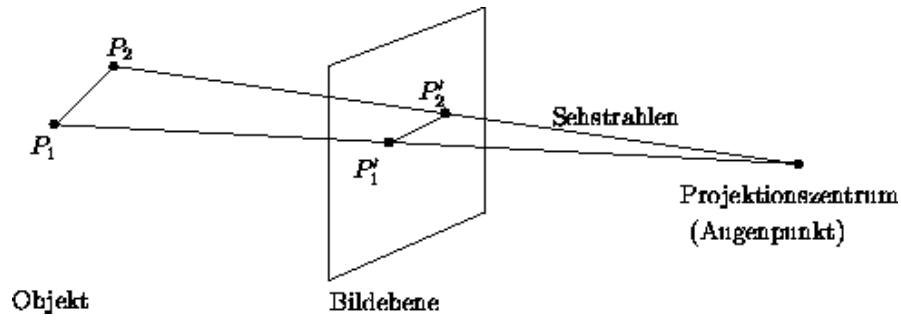

Abbildung 4.14: Beispiel Zentralprojektion

aufgrund des Strahlensatzes folgende Beziehung:

$$xneu = \frac{x}{1 + \frac{z}{\alpha}}$$

$$yneu = \frac{y}{1 + \frac{z}{\alpha}}$$

Um diese Berechnung durchführen zu können, muss also der minimale Abstand des Augpunktes zur Bildebene bekannt sein.

Alpha und damit die perspektivische Verzerrung der virtuellen Objekte soll bei unserem Augmented Reality System identisch mit der Perspektive des Kamerabildes sein, deshalb muss mit Hilfe von Messungen der aufgenommenen Marker die Entfernung α erst bestimmt werden.

Eine alternative Möglichkeit ist, sich auf die Anzeige zweidimensionaler Objekte zu beschränken. Dabei bildet der Marker eine Ebene durch den Raum, auf welchem die zweidimensionalen virtuellen Objekte plaziert werden.

Die Berechnung des jeweiligen Punktes erfolgt mit Hilfe der vier Eckkoordinaten des Markers auf dem zweidimensionalen Bild und der vorher definierten realen Seitenlänge des Markers.

Die dafür benötigten Formeln lauten wie folgt:

$MarkerX_{ol}$ ist die X-Markerkoordinate oben links.

$MarkerX_{or}$ ist die X-Markerkoordinate oben rechts.

usw.

$Seite$ ist die definierte reale Seitengröße des Markerquadrates.

x und y sind die zu transformierenden Koordinaten aus dem virtuellen Raum.

$xneu$ und $yneu$ sind die neuberechneten Koordinaten für x und y auf dem zweidimensionalen Bild.

$$xneu = MarkerX_{ol} * \frac{Seite - x}{Seite} * \frac{Seite - y}{Seite} + MarkerX_{or} * \frac{x}{Seite} * \frac{Seite - y}{Seite} + MarkerX_{ur} * \frac{x}{Seite} * \frac{y}{Seite} + MarkerX_{ul} * \frac{Seite - x}{Seite} * \frac{y}{Seite}$$

$$yneu = MarkerY_{ol} * \frac{Seite - x}{Seite} * \frac{Seite - y}{Seite} + MarkerY_{or} * \frac{x}{Seite} * \frac{Seite - y}{Seite} + MarkerY_{ur} * \frac{x}{Seite} * \frac{y}{Seite} + MarkerY_{ul} * \frac{Seite - x}{Seite} * \frac{y}{Seite}$$

4.4 Erkennung von Handbewegungen zur Eingabe

Anforderung an das System ist eine einfache visuelle Erkennung von Handbewegungen zur Eingabe in das System. Da dieses nicht Kern der Arbeit ist, soll sich die Erkennung auf ein einfaches hin- und herschwenken der Hand vor der Kamera beschränken.

Ein mehrmaliges Schwenken der Hand vor der Kamera wird als Aktivierung erkannt, ein nochmaliges Schwenken als Deaktivierung.

Die Farbe der menschlichen Hand wird aus dem Videobild extrahiert und es wird die Bildveränderung, relativ zu einem vorher abgespeicherten Bild ge-

messen. Wenn die Bildveränderung groß genug ist, wurde die Hand bewegt und eine Aktivierung bzw. Deaktivierung erkannt.

5 Realisierung

Aus der Analyse geht hervor, dass ein eigenes Softwaresystem erstellt werden muss. Im folgenden Kapitel geht es um die Erstellung einer prototypischen Anwendung, welche die wichtigsten Anforderungen erfüllt. Dabei wird auf die verwendeten Hard- und Softwarekomponenten eingegangen und deren Aufbau und Funktionsweise erklärt.

5.1 Verwendete Hardwarekomponenten

Für den Prototyp wurde ein Mininotebook der Firma JVC eingesetzt. Seine hohe Akkulaufzeit und seine kleine, kompakte Bauform ermöglichen es, den Rechner in einer einfachen leichten Jacke am Körper zu tragen. Zusätzlich verfügt das Gerät über mehrere USB Anschlüsse und einen extra Monitorausgang um Kamera und HMD anzuschließen.

Die eingesetzte Kamera ist aus Kostengründen eine einfache USB-Webcam der Firma Philips mit einer maximalen Videoauflösung von 640x480. Aus Performanzgründen wurde die Auflösung auf 320x240 heruntergestellt. Bild Webkamera 5.1

Abbildung 5.1: Webkamera

Zur Ausgabe wurde ein Head-Mounted-Display (HMD) von der Firma MicroOptical mit einer maximalen Auflösung von 640x480 verwendet. Das HMD besitzt nur ein Display für ein Auge und kann nur im Video-based Modus arbeiten. Das Display kann an jede gängige Brille mit dünnem Gestell geklemmt werden und besitzt mehrere Gelenke zur richtigen Positionierung. Das Display kann mit Netzstrom oder auch mit ortsungebundenen Akkus betrieben werden. (Siehe Bilder 5.2 und 5.3)

Abbildung 5.2: HMD Seitenansicht

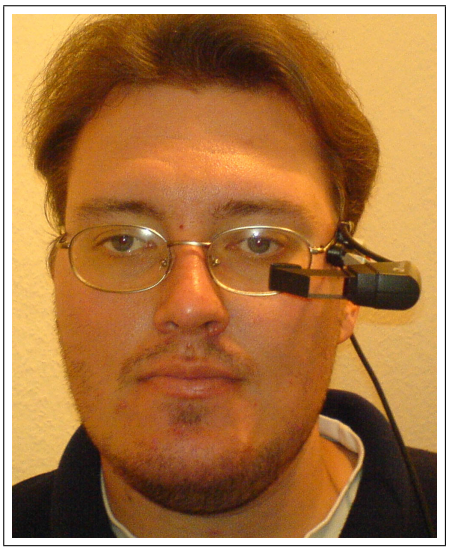

Abbildung 5.3: HMD Vorderansicht

5.2 Verwendete Softwarekomponenten

Aus Kosten- und Zeitgründen wurde beim Betriebssystem auf ein Echtzeitbetriebssystem verzichtet und das auf dem JVC Notebook vorinstallierte Microsoft Windows XP zurückgegriffen. Die bessere Alternative wäre ein Echtzeitbetriebssystem wie QNX gewesen, welches jedoch zu einer zu hohen Einarbeitungszeit geführt hätte.

Als Programmiersprache wurde JAVA [Java] gewählt, da JAVA mit einer umfangreichen Bibliothek zur Videoaufnahme und Bildverarbeitung ausgestattet ist und keine lange Einarbeitungszeit des Programmierers benötigt wurde. Außerdem erfüllt JAVA alle wichtigen Grundregeln für gutes Programmieren.

Jede andere Programmiersprache, welche die Grundregeln für gutes Programmieren und Softwaredesign erfüllt, (Zum Beispiel Erweiterbarkeit, Wartbarkeit, Übertragbarkeit, Wiederverwendbarkeit, e.t.c.) hätte für diese Aufgabe auch verwendet werden können.

Als Entwicklungsumgebung wurde die JAVA Standardentwicklungsumgebung Eclipse 3.1 für Windows eingesetzt.

5.2.1 Softwarekomponenten

Die eigentliche programmierte Software besteht aus drei Komponenten:

- Komponente zur Steuerung der Kamera
- Komponente zur Speicherung der Markerbeschreibung und der Beschreibung der virtuellen Elemente.
- Komponente zur Verarbeitung und Steuerung.

Die Komponente zur Steuerung der Kamera wird repräsentiert durch die Klasse **CameraController**. Diese Klasse ist für die Initialisierung und Ansteuerung der Webkamera zuständig. Sie nutzt dazu die Klasse Player aus dem Java Media Framework [JMF] von Sun. Nach der Instanzierung und Initialisierung, kann über die Methode getFrame ein aktuelles Bild von der Webkamera gelesen werden.

Die Komponente zur Speicherung der Markerbeschreibung und der virtuellen Elemente wird durch die Klassen **Marker**, **VirtualWorld** und der abstrakten Klasse **VirtualObjekt** verkörpert.

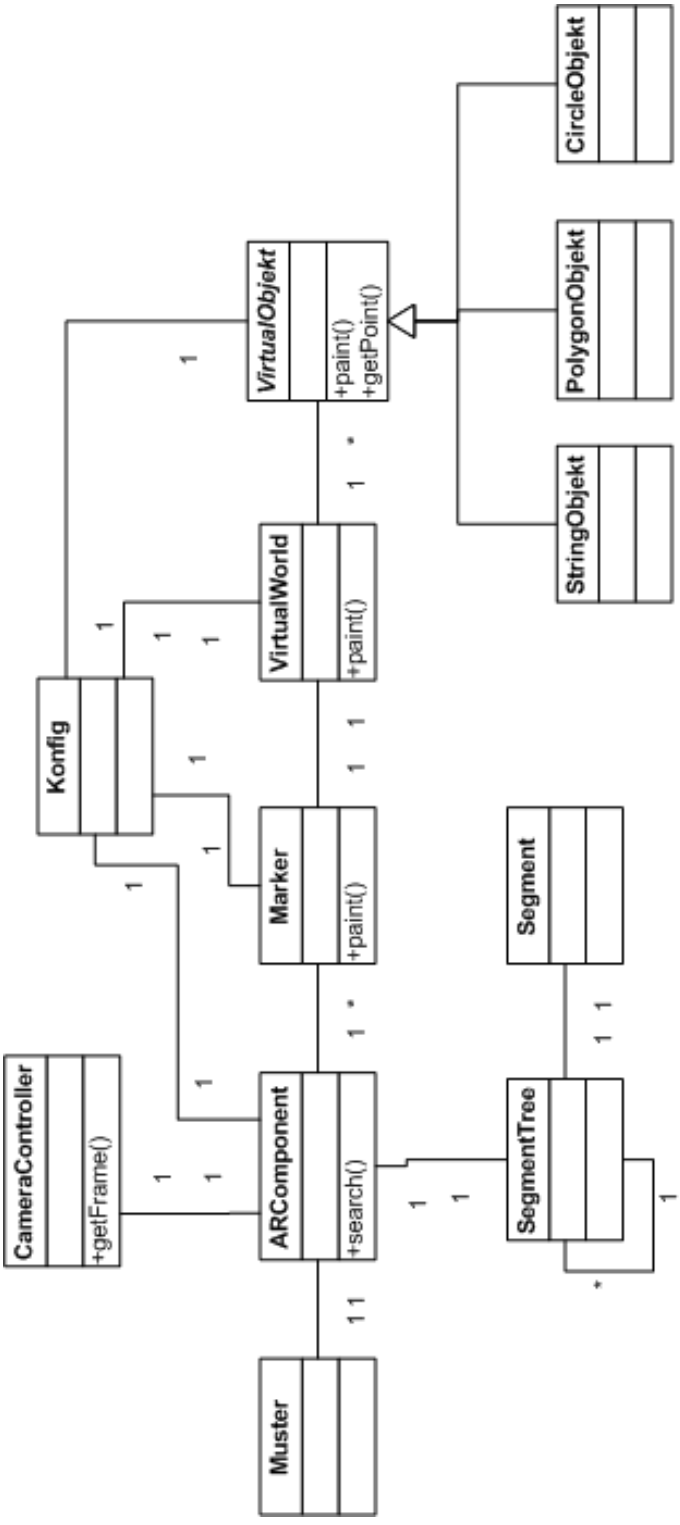

Abbildung 5.4: Zentrales Klassenmodel

Die Klasse **Marker** enthält die Beschreibung über den Aufbau und die reale Größe der einzelnen Marker. Zusätzlich sorgt die Klasse in der Methode paint dafür, dass gegebenenfalls die Markerumrandung und die Objekte aus der Klasse **VirtualWorld** gezeichnet werden.

Die Klasse **VirtualWorld** enthält für die einzelnen Marker die Szenenbeschreibung der virtuellen Welt. Jede virtuelle Welt besteht aus einem oder mehren Objekten der Klasse **VirtualObjekt**.

Die Klasse **VirtualObjekt** ist eine abstrakte Oberklasse zur Beschreibung der eigentlichen virtuellen Objekte. Die abstrakte Methode paint ist zum Zeichnen des jeweiligen Objektes. Die Methode getPoint transformiert den Punkt im virtuellen Raum auf einen Punkt in der zweidimensionalen Bildebene. Abgeleitete Klassen der Klasse sind **VirtualObjekt** sind **PolygonObjekt**, **StringObjekt** und **CircleObjekt**.

In diesen Klassen wird das jeweilige virtuelle Objekt gezeichnet, alle diese Klassen bedienen sich der Methode getPoint aus der Mutterklasse **VirtualObjekt** um die Raumkoordinaten abhängig von der Position des Markers in die richtigen Koordinaten der Bildebene zu transformieren.

Die Klasse **ARComponent** bildet die Komponente zur Verarbeitung und Steuerung. Sie ist abgeleitet von der Java AWT Klasse Component. In der Methode findObjekts wird ein aktuelles Bild von der Kamera geholt, das Bild segmentiert und auf vorhandene Marker überprüft. Anschließend werden die gefundenen Marker zusammen mit ihrer Bildposition gespeichert und ein neu zeichnen des Fensters veranlasst.

Die Segmentierung des Bildes in einzelne Unterbereiche erfolgt dabei als rekursive Instanzierung der Klasse **SegmentTree**, welche einen einfachen Baum aus SegmentTree-Objekten aufbaut. Jedes SegmentTree Objekt enthält dabei sein eigenes Segment und alle aus diesem Segment extrahierten SegmentTree Objekte. Zur Beschleunigung ist eine maximale Tiefe von zwei Ebenen eingestellt. (Siehe Bild 5.5)

Beim Neuzeichnen des Fensters werden für alle Marker, die auf dem Videobild erkannt wurden, deren zugehörige virtuellen Objekte in das Fenster gemalt.

Aus Zeitgründen wurde auf eine Anzeige von dreidimensionalen Objekten verzichtet, stattdessen wurde die vereinfachte zweidimensionale Alternative gewählt, Bei dieser Variante werden zweidimensionale Objekte wie Polygone oder Kreise auf der Ebene des Markers im Raum angezeigt. (Siehe Abschnitt 4.3.2)

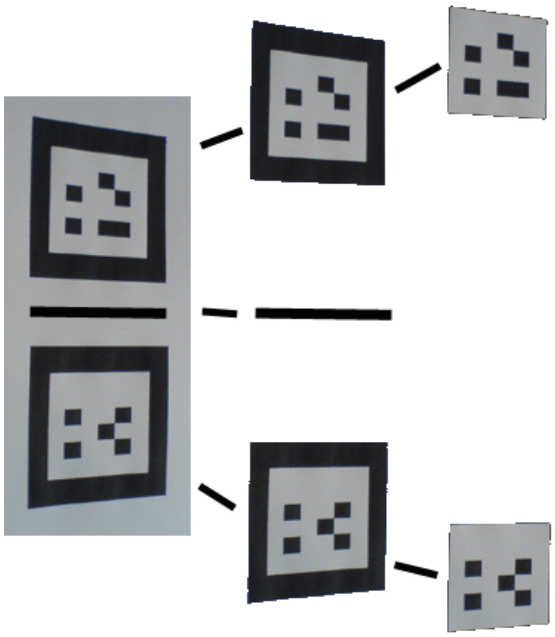

Abbildung 5.5: Segmentierungsbaum der Klasse SegmentTree

Die Klassen **ARComponent**, **Marker**, **VirtualWorld** und **VirtualObjekt** besitzen jeweils die Referenz auf ein Objekt der Klasse **Konfig**. Die Klasse **Konfig** ist ein reines Datenobjekt, welche die jeweiligen Einstellungen für die Anzeige der Objekte am Bildschirm enthält.

5.3 Verfahren zur Erkennung von Handbewegungen

Um eine simple Eingabe zu ermöglichen, wird ein Verfahren zur Erkennung der Handbewegung eingesetzt.

Die Verarbeitung erfolgt in der Klasse ButtonController.

Als erstes wird das bereits verwendete Orginalbild der Kamera wieder binarisiert. Dieses mal werden jedoch nur die Pixel betrachtet, welche eine typische rosa Hautfarbe plus/minus eine fixe Toleranz besitzen. Die Reverenzhautfarbe und die Toleranz werden vorher festgelegt.

Beim ersten Durchlauf wird dieses Binärbild intern abgespeichert und die Verarbeitung beendet. Bei jedem weiteren Durchlauf wird die Anzahl der ungleichen Rasterpunkte gegenüber dem älteren intern abgespeicherten Binärbild gezählt.

Wenn diese Bildänderung $\frac{1}{3}$ der Anzahl der Rasterpunkte übersteigt, wird ein weiterer Zähler hochgezählt.

Sollte dieser Zähler sein Maximum von drei erreichen, wechselt der Button seinen Aktivierungsstatus von Deaktiviert $\Rightarrow$ Aktiviert oder von Aktiviert $\Rightarrow$ Deaktiviert.

Dieses wird für eine einfache Erweiterung genutzt, die Klasse ARComponent fügt bei aktiviertem Button ein StringObjekt mit dem Wort „Selected“ zu den virtuellen Objekten jedes Markers hinzu.

5.4 Fazit

Der realisierte Prototyp erfüllt den Großteil der in der Einführung und in der Analyse spezifizierten Anforderungen.

Das System nimmt mittels einer Kamera ein Bild auf und erkennt auf diesem vorher definierte und im Raum angebrachte Marker. Es plaziert - relativ zur Perspektive des Markers - virtuelle Objekte wie Kreise, Polygone oder Text in das Bild und zeigt diese dem Benutzer an, so als würden diese Objekte sich im realen Raum befinden. Die Anzeige kann in der Orginalauflösung der Kamera oder skaliert auf die Größe des Ausgabefensters erfolgen.

Für den Prototyp stand nur ein Head-mounted-Display mit reinem Video-based-see-through-Modus zur Verfügung. Der Optical-see-through-Modus konnte mangels eines diesen Modus unterstützenden Displays nicht vollständig implementiert werden. Das System würde auch auf einem Optical-see-through-Display arbeiten, müsste dann jedoch noch mit einem Kalibrierungssystem erweitert werden.

Durch das Fehlen eines Kalibrierungssystems werden Fehler und Fehlerquellen wie der Abstand zwischen Kamera und dem Display oder auch der Abstand zwischen Auge und Display nicht kompensiert. Das angezeigte Bild wird dadurch nur relativ zur Videokamera, jedoch nicht relativ zum Auge perspektivisch korrekt dargestellt.

Die Anforderung ein Echtzeitsystem zu entwickeln, konnte aus Zeitgründen im Prototyp nicht realisiert werden. Die derzeitige Architektur der Software ist in ihrer Arbeitszeit stark abhängig von den von der Kamera gelieferten Bildern. Dieses schließt im Moment eine Anwendung in Echtzeit aus. Der Prototyp kann jedoch im nächsten Schritt zu einem Echtzeitsystem ausgebaut bzw. erweitert werden.

Das benutzte Head-mounted-Display ersetzt durch das zu kleine Display und den zu großen Abstand zum Auge nur einen kleinen Teil des menschlichen Sichtfelds. Deshalb ist es für Augmented Reality Anwendungen, welche das gesamte menschliche Sichtfeld ersetzen bzw. manipulieren wollen, nicht geeignet.

Die Entwicklung von Computern und Anzeigegeräten entwickelt sich zur Zeit so rasant, dass in den nächsten Jahren bestimmt hochauflösende Displays mit eingebauten Kameras zu erschwinglichen Preisen zur Verfügung stehen, wodurch diese Probleme schnell gelöst werden. Ein neues Head-Mounted-Display muss in der Lage sein viel des Sichtfeldes des Benutzers abdecken und die Kamera sollte sich nah zwischen beiden Augen befinden, um Fehler zu kompensieren.

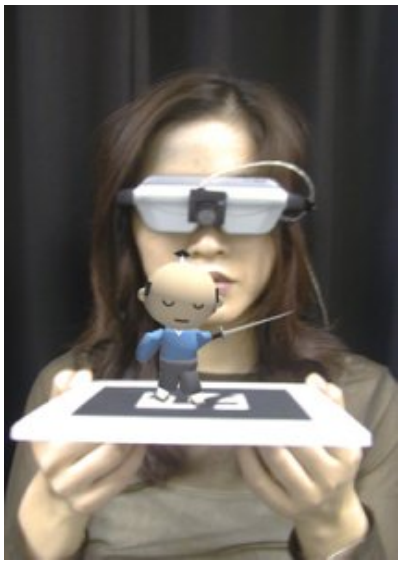

Abbildung 5.6: Modernes Head-Mounted-Display mit Kamera

6 Zusammenfassung

Im folgenden Kapitel werden die Arbeiten und die daraus erworbenen Erkenntnisse abschließend zusammengefasst. Es werden Erfolge und Misserfolge der Arbeit aufgezeigt und Verbesserungsmöglichkeiten vorgeschlagen. Anschließend werden Möglichkeiten zur zukünftigen Entwicklung und Weiterführung des erstellten Prototypen dargelegt. Als letztes kommt ein persönlicher Kommentar über die persönlichen gewonnenen Erfahrungen des Autors beim erstellen der Arbeit.stellen der Arbeit.

6.1 Ausgangslage

Ausgangslage war die Aufgabe, ein Augmented Reality System zu erstellen, welches mittels einer Kamera in die Realität platzierte Objekte erkennt. Anschließend sollten auf einem Head-Mounted-Display, in Abhängigkeit der Position der platzierten Objekte, virtuelle Objekte eingeblendet werden. Dabei sollten die eingeblendeten/eingebetteten virtuellen Objekte so wirken, als seien sie ein Teil der realen Welt. Es sollte dabei nur auf bereits vorhandene Hardware zurückgegriffen werden und als Software nur bereits vorhandene und/oder lizenzfreie Softwarekomponenten verwendet werden.

6.2 Resumee

Lösungsansatz war die Entwicklung einer eigenen Software, welche Bilder der realen Welt mittels einer Webkamera aufnimmt, spezielle Marker auf diesen Bildern erkennt, die Position der Marker im Kamerabild berechnet und auf einem Head-Mounted-Display simple virtuelle Figuren in Abhängigkeit der Markerposition perspektivisch korrekt darstellt.

Es wurde eine prototypische Anwendung entwickelt, um die Machbarkeit der in der Analyse aufgestellten Anforderungen zu beweisen.

Die entwickelte Software erkennt auf mehrere Meter Entfernung platzierte Marker und zeigt auf einem Video-see-through-Head-Mounted-Display das Kamerabild mit den eingebetteten virtuellen Objekten an.

Es wurde auf die Darstellung von dreidimensionalen Objekten verzichtet und stattdessen nur zweidimensionale Objekte auf der Ebene des jeweiligen Markers im Raum platziert. Diese werden, sobald ein Marker erkannt wird, in dem vorhandenen Videobild perspektivisch korrekt angezeigt.

Eine weitere Anforderung war die simple Eingabe mittels Handbewegung. Dieses wurde mit einem einfachen Verfahren zur Erkennung von Bildveränderungen realisiert. Zur Veranschaulichung und als Beispiel werden im Prototyp bei Aktivierung weitere Informationen in Form von virtuellen Objekten in die reale Welt ein- bzw. ausgeblendet.

Der entwickelte Prototyp erfüllt alle wichtigen an ihn gestellten Anforderungen: Ein System welches Informationen in die Reale Welt - abhängig von der Position des Benutzers - einblendet. Jedoch sollten einige Bereiche und Komponenten des Prototyps noch verbessert und weiterentwickelt werden.

6.3 Verbesserungsmöglichkeiten

Durch die gewonnene Erfahrung bei der Entwicklung des ersten Prototypen, sollten einige Teile im Design und der Realisierung abgeändert werden.

Das Softwaresystem des Prototypen sollte um eine Komponente zur Kalibrierung erweitert werden, dadurch können die perspektivischen Fehler, welche durch den physikalischen Abstand zwischen Kamera und Display entstehen, verringert werden.

Derzeit werden nur auf einer Ebene liegende zweidimensionale Objekte angezeigt. Das System sollte dahingehend erweitert werden, dass zwei- und dreidimensionale virtuelle Objekte frei im Raum platziert und perspektivisch korrekt angezeigt werden können. Das Anzeigen von dreidimensionalen Objekten ist eine Grunddisziplin der virtuellen Realität und sollte deshalb kein großes Problem darstellen.

Um eine flüssigere Anzeige und einen geringeren Ende-zu-Ende Fehler zu erreichen, sollte die ganze Architektur des Systems auf ein Echtzeitsystem umgestellt werden. Der Prototyp zeigt, dass die derzeit verwendete Technik zur Objekterkennung zeitweilig zu massiven Rucklern führt.

6.4 Aussichten, Weiterführung

Mit dem Ziel ein Museumsinformationssystem zu entwickeln sollte als nächstes die Anzeige von virtuellen Objekten um multimediale Inhalte erweitert werden. Zum Beispiel dreidimensionaler Sound , Video und animierte dreidimensionale Objekte. Dadurch könnte zum Beispiel das eingeblendete Videobild eines Sprechers einen Vortrag über die Beleuchtungseffekte eines Gemäldes halten, während auf dem Gemälde selber markante Punkte hervorgehoben oder sogar herausgelöst werden.

Der nächste Schritt wäre der Einbau von virtuellen Klippingobjekten. Diese Objekte dienen als Platzhalter für Objekte aus der realen Welt. Dadurch wird zum Beispiel ein virtueller Ball, welcher sich teilweise hinter einem realen Tisch befindet, nur da angezeigt wo er über den Tisch hinaus ragt.

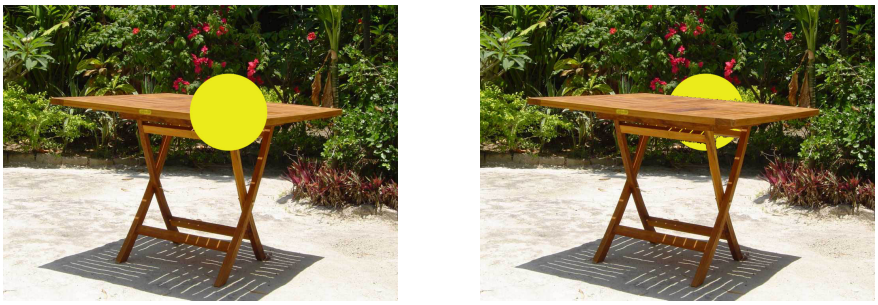

Abbildung 6.1: Tisch mit und ohne Klippingobjekt

Das System sollte um eine Komponente zur komfortablen Eingabe erweitert werden. Diese Eingabe kann zum Beispiel über Gestenerkennung, Sprachsteuerung oder virtuelle Buttons erfolgen.

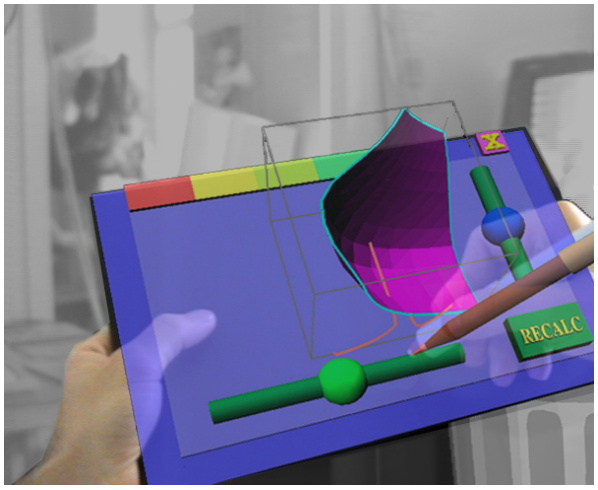

Abbildung 6.2: Virtuelles Eingabefeld

Zuletzt sollte sich das System von der markerbasierten Objekterkennung hin zur szenenbasierten Objekterkennung hin entwickeln. Dadurch ist es zum Beispiel möglich anhand der Konturen eines Bildes zu erkennen, um welches Bild es sich handelt und mittels von Schlüsselpunkten im Bild die Position des Objektes zu errechnen.

Alle diese Vorschläge wurden in dieser oder abgewandelter Form bereits in anderen Systemen implementiert, sind also grundsätzlich realisierbar.

6.5 Kommentar

Hier möchte ich zum Schluss meine persönlichen Eindrücke und gewonnenen Erfahrungen beim Erstellen dieser Arbeit wiedergeben.

Als ich mich, in Absprache mit meinem Professor, für die Bearbeitung dieses Themas entschied, war mir noch gar nicht klar wie spannend das Thema und das Umfeld Augmented Reality und Mixed Reality sind.

Auch wenn die Technik und die Forschung im Bereich Augmented Reality noch in den Kinderschuhen stecken, so könnte es doch in Zukunft wie der MP3-Player ein ganz normaler Teil unseres Lebens werden.

Während heutzutage Augmented Reality nur in kleinen Bereichen wie Sportübertragungen im Fernsehen Anwendung findet, so beginnen bereits die ers-

ten Firmen und Forschungseinrichtungen weitergehende Produkte zu entwickeln. Beispiele sind da Unterstützungssysteme bei der Reparatur von Kraftfahrzeugen oder auch Lernsysteme in der industriellen Ausbildung.

Das Potenzial der Technik, einem Benutzer Informationsobjekte zur Verfügung zu stellen, als wären diese physikalisch vorhanden, ist riesengroß.

Wenn die Preise für die heutzutage noch viel zu teure Hardware in Zukunft sinken und die Entwicklung im Bereich wearable Computing voranschreitet, wird auch der Bereich Augmented Reality ein ständiger Begleiter in unserem Leben sein.

Ich war überrascht, wie relativ einfach es war, ein simples Augmented Reality System zu entwickeln. Die Grundfunktion ein Objekt mittels Kamera zu erkennen und entsprechend Objekte auf dem Bildschirm anzuzeigen, wurde schnell realisiert. Jedoch konnten aus Zeitgründen einige Anforderungen nur teilweise oder auch gar nicht umgesetzt werden. Auch die verwendete Hardware wie die einfache Webkamera oder das Head-Mounted-Display mit viel zu kleinem Sichtfeld führte zu einem Augmented Reality System, welches vorerst keinen Platz in einem Museum finden wird. Die Probleme und Herausforderungen stecken nun im nächsten Schritt, aus einem simplen Augmented Reality System ein umfangreicheres und effizienteres System mit allen Anforderungen zu machen. Realzeitverarbeitung, hochauflösende Positionsbestimmung, multimediale Darstellung und vor allem Robustheit gegenüber dem Benutzer sind alles Ziele, welche noch eine Menge Arbeit erfordern, die investiert werden muss, um aus dem entwickelten Prototyp eine akzeptable Anwendung zu machen.

Bei meiner Materialsuche habe ich festgestellt, wie viele Firmen und Bildungseinrichtungen sich bereits mit dem Thema Augmented Reality und Ambiente Intelligence beschäftigen. Die präsentierten Arbeiten sind unter Laborbedingungen bereits sehr fortgeschritten und scheitern oft nur an der noch unzureichenden Hardwaretechnologie. Auch diese Erfahrung hat gezeigt, dass der Bereich Augmented Reality ein großes Potenzial für die Zukunft hat. Im Hardwarebereich ist es nicht mehr eine Frage ob, sondern vielmehr wann die geeignete Hardware zur Verfügung steht. Im Bereich Software und Ergonomie muss jedoch noch viel Arbeit investiert werden.

Zusätzlich habe ich beim Erstellen dieser Arbeit erkannt, dass optische Augmented Reality Wissen in vielen verschiedenen Diszipinen wie zum Beispiel Bildmustererkennung, Virtual Reality, Mixed Reality oder Positionsbestimmung

Abbildung 6.3: Filmausschnitt Jurassic Park, Universal Studios 1993

(Tracking) erfordert. Nach der Einarbeitung in diese Themen ist es möglich, in kurzer Zeit ein einfaches Augmented Reality System zu entwickeln.

Die Technik und Technologie schreiten stetig voran und das Stichwort der Zukunft heißt dann nicht mehr nur Augmented Reality, sondern Ambient Intelligence. Viele kleine Computer in unserer Umgebung reichern unsere Wahrnehmung mit wichtigen Informationen an. Sei es nun ein vibrierendes Lenkrad, welches uns mitteilt, dass wir von der Straße abkommen oder ein zyklischer Piepton, welcher uns beim Einparken mitteilt, wie weit wir von der Stoßstange des Wagens hinter uns noch entfernt sind. Tausende Anwendungen sind für die Zukunft denkbar. Jedoch fängt da die eigentliche Arbeit eines Informatikers erst an. Er muss die neuen Ideen umsetzen und für die Gesellschaft brauchbar und benutzbar machen.

Ambient Intelligence und Augmented Reality werden allgegenwärtig und vielleicht sind auch wir eines Tages in der Lage, virtuelle Dinos in unserem Garten betrachten zu können. Einfach so, als wären sie ein Teil der realen Welt.

Doch bis dahin wird bestimmt noch oft versucht werden, das Unmögliche möglich zu machen.

Literaturverzeichnis

[ARToolKit] ARToolKit, Human Interface Technology Laboratory. – URL http://www.hitl.washington.edu/artoolkit/

[Metagis] Definition Bildmustererkennung, Metagis Verlag. – URL http://www.metagis.de/metagis/glossar/glossar-text1.asp?Text=Bildmustererkennung

[QNX] Echtzeitbetriebssystem QNX, URL www.qnx.com

[Java] *Java Hompage*, Sun Microsystems. – URL http://www.java.com

[JMF] *Java Media Framework Webpage*, Sun Microsystems. – URL http://java.sun.com/products/java-media/jmf/

[MARQ] Mobile Augmented Reality Quest, Technische Universität Wien. – URL http://studierstube.icg.tu-graz.ac.at/handheld_ar/marq.php

[StbAPI] Studierstube, Technische Universität Graz. – URL http://studierstube.icg.tu-graz.ac.at/doc/stb.orig/intro.html

[Tinmith] Tinmith augmented reality project, URL http://www.tinmith.net/

[Azuma 1997] AZUMA, Ronald T.: A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments, URL http://www.cs.unc.edu/~azuma/ARpresence.pdf, 1997

[Gerling 2006] GERLING, Mirco: *Wearable Computing & Mobile Augmented Reality im Ferienclub*, Dissertation, 2006. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master05-06-proj/gerling/paper.pdf

[Kutak 2006] KUTAK, Edyta: *Entwicklung eines Location Tracking Systems für die Indoor Navigation*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2006. – URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/kutak/folien.pdf

[Meisel 2005] MEISEL, Prof. D.: *Vorlesungsfolien "Robot Vision"*, URL http://www.informatik.haw-hamburg.de/432.html, 2005

[metaioGmbH] METAIOGMBH: Unifeye SDK, URL http://www.metaio.com

[Navab] NAVAB, Prof. Dr. N.: Distributed Wearable Augmented Reality Framework, URL http://ar.in.tum.de/Chair/ProjectDwarf

[ORAD] ORAD: *ORAD Mastering Graphics*, URL http://www.orad.co.il/

[Wagner 2005] WAGNER, Martin: Vorlesungsfolien Augmented Reality Sommersemester, URL https://wiki.medien.ifi.lmu.de/pub/Main/AugmentedRealityVorlesungSoSe2005/AR-S05-12-Authoring6side.pdf, 2005

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 26. August 2006	
Ort, Datum	Unterschrift