

Informatik im Fokus

Herausgeber:

Prof. Dr. O. P. Günther

Prof. Dr. W. Karl

Prof. Dr. R. Lienhart

Prof. Dr. K. Zeppenfeld

Informatik im Fokus

Weitere Titel der Reihe Informatik im Fokus:
<http://www.springer.com/series/7871>

Marcus Tönnis

Augmented Reality

Einblicke in die Erweiterte Realität

 Springer

Marcus Tönnis
TU München
Institut für Informatik (I-16)
Boltzmannstrasse 3
85748 Garching b. München
Deutschland
toennis@in.tum.de

Herausgeber

Prof. Dr. O. P. Günther
Humboldt-Universität zu Berlin

Prof. Dr. W. Karl
Universität Karlsruhe (TH)

Prof. Dr. R. Lienhart
Universität Augsburg

Prof. Dr. K. Zeppenfeld
Hochschule Hamm-Lippstadt

ISSN 1865-4452

e-ISSN 1865-4460

ISBN 978-3-642-14178-2

e-ISBN 978-3-642-14179-9

DOI 10.1007/978-3-642-14179-9

Springer Heidelberg Dordrecht London New York

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer-Verlag Berlin Heidelberg 2010

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Einbandentwurf: KuenkelLopka GmbH

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media (www.springer.com)

Vorwort

Erweiterte Realität (oder „Augmented Reality“, AR) stellt eine logische Weiterentwicklung der Virtuellen Realität (VR) dar. Im Gegensatz zur VR wird in der AR die den Benutzer umgebende tatsächliche Realität durch dreidimensionale virtuelle Elemente erweitert. Diese betten sich nahtlos in die Umgebung ein und stellen neue Gegenstände dar oder geben, im direkten räumlichen Bezug, weitere Informationen zu existierenden Objekten. Die breite Masse kann AR momentan allerdings höchstens in Sportsendungen erleben, wenn z.B. im Fußball Entfernungslinien zum Abseits eingeblendet werden.

Erste AR-Anwendungen können allerdings selbst mit geringer Programmiererfahrung erstellt werden. Genau hier platziert sich auch dieses Buch. Mit seinem Umfang kann es selbstverständlich niemals alle Möglichkeiten, Probleme und Lösungen der AR umfassen, vielmehr soll es eine Fibel darstellen, die aufzeigt, wie erste Probleme bei der Einbettung virtueller Objekte in der Realität gemeistert werden. Das im Zuge dieses Buches entwickelte System stellt kein vollkommenes AR-System dar. Hierzu wären z.B. Da-

tenbrillen und andere Sensorsysteme notwendig, die aber schnell größere Kosten erreichen. Beispiele aus Anwendungen der AR zeigen aber, dass das hier gegebene Handwerkszeug die essenzielle Basis für jede Art von AR-Systemen ist. Je nach weitergehendem Interesse erlauben im Buch angegebene Referenzen dann das Vertiefen in einzelne Bereiche.

An dieser Stelle möchte ich mich auch bei den Personen bedanken, die zum Entstehen dieses Buches beigetragen haben. Der Dank geht an Herrn Heine, der überhaupt die Idee eines Buches dieser Art aufbrachte. Hervorheben möchte ich auch die wertvolle Zusammenarbeit mit den Mitarbeitern des Springer Verlages. Spezieller Dank geht an Manuel Huber, der mir mit wirklich großem Einsatz unter die Arme gegriffen hat, damit ich dieses Buch und besonders die Übungen umsetzen konnte! Ohne Professor Gudrun Klinker, Ph.D. hätte dieses Buch auch nicht entstehen können. Als Mentor meiner Doktorarbeit hat sie mich in das Thema Augmented Reality eingeführt und mir viele wertvolle Erfahrungen mitgegeben. Schließlich danke ich meinen Kollegen und Freunden, die mich viele Stunden entbehren mussten, weil ich unzählige Abende hinter meinem Schreibtisch verbracht habe.

Ich wünsche allen Lesern viel Spaß beim Lesen und beim Ausprobieren der einzelnen Übungen. Sein eigenes AR-System wachsen zu sehen, neue Probleme zu erkennen und diese dann zu lösen, ist eine ganz besondere Erfahrung, die nicht nur zwischen Bits und Bytes, sondern auch in der realen Welt liegt!

München, im April 2010
Marcus Tönnis

Inhaltsverzeichnis

1	Einführung in die Augmented Reality	1
1.1	Paradigmen der Augmented Reality	1
1.2	Geschichte	3
1.3	Komponenten von AR-Systemen	4
1.4	Empfohlene Voraussetzungen	6
2	Darstellung virtueller Objekte	7
2.1	3-D-Rendering	8
2.1.1	Virtuelle Kamera	8
2.1.2	Räumliche Transformationen	13
2.1.3	Hierarchien und Szenengraphen	19
2.2	Displays	21
2.2.1	Head-Mounted Displays	23
2.2.2	Raum- und umgebungsfixierte Displays	25
2.2.3	Bewegliche Displays	28
2.2.4	Handheld Displays	28
2.3	Übung - Video-See-Through auf normalen TFT-Displays	31
2.4	Andere Arten der Darstellung	36

3	Tracking	43
3.1	Technologien und Prinzipien	44
3.1.1	Optisches Tracking	44
3.1.2	Inertialtracking	52
3.1.3	Magnetisches Tracking	53
3.1.4	Laufzeitbasiertes Tracking	54
3.1.5	Mechanisches Tracking	58
3.1.6	Prinzipien des Trackings	58
3.2	Kalibrierung und Registrierung	60
3.2.1	Korrektur der Linsenverzerrung	60
3.2.2	Spitzenkalibrierung	63
3.2.3	Absolute Orientierung	66
3.2.4	Hand-Auge-Kalibrierung	70
3.2.5	Displaykalibrierung	71
3.2.6	Sensorfusion	80
3.3	Visuelle Störeffekte und andere Probleme ...	82
3.3.1	Schwimmeffekte	83
3.3.2	Verdeckung	84
3.3.3	Jitter	87
3.3.4	Fehlerfortpflanzung	89
3.4	Übung - AR Toolkit	90
4	Eingabe und Interaktion	95
4.1	Eingabe	96
4.1.1	Marker-Basierte Eingaben	96
4.1.2	Tangible User Interfaces	98
4.1.3	Motion-Capturing	100
4.1.4	Eingaben mit dem Blick	104
4.1.5	Spracheingabe	105
4.2	Interaktion	106
4.2.1	Selektion	108
4.2.2	Manipulation	111
4.2.3	Bewegung	113
4.2.4	Signaletik	115

4.2.5	Systemeinstellungen	118
4.2.6	Symbolische Eingaben	120
4.3	Übung - Kontrolle und Steuerung von Objekten mit Markern	123
5	Anwendungen und Erfahrungen	127
5.1	Industrie	127
5.2	Medizin	132
5.2.1	Überlagerung des Körpers mit klinischen Daten	133
5.2.2	Zielführung bei minimalinvasiver Chirurgie	136
5.2.3	Virtual-Mirror	137
5.3	Automotive AR	140
5.3.1	Nightvision und Großdistanz HUDs ..	141
5.3.2	Kontaktanaloge Navigationssysteme .	144
5.3.3	Anzeige eines Bremsbalkens.....	145
5.4	Forschung und Entwicklung	148
5.4.1	Darstellung des Sichtfeldes.....	148
5.4.2	Visualisierung von Sensordaten.....	149
5.4.3	Multimodale AR	152
5.5	Edutainment.....	153
5.5.1	Museumsführer	154
5.5.2	Interaktive Pflanzenklassifikation ...	156
5.6	Spiele	158
5.6.1	AR-Quake	158
5.6.2	Eye of Judgement	159
6	Ausblicke	161
6.1	Ubiquitous Augmented Reality	162
6.2	Zukunftstrends	165
7	Fazit	173

Lösungen	175
Literaturverzeichnis	181
Sachverzeichnis	191

Einführung in die Augmented Reality

Wem zum ersten Mal der Begriff Erweiterte Realität (Augmented Reality, AR) und die zugehörigen Prinzipien erklärt werden, antwortet meistens folgendermaßen: „Das gibt es doch schon in den Kinofilmen, die im Computer nachbearbeitet werden“. Der Vergleich zu Kinofilmen ist allerdings nur teilweise richtig. Um die Unterschiede genauer zu erklären, beginnt dieses Buch mit der Aufschlüsselung der grundlegenden Paradigmen der Augmented Reality. Die nachfolgenden Kapitel beziehen schrittweise einzelne Technologien und Komponenten von AR-Systemen ein und entwickeln dann ein exemplarisches AR-System.

1.1 Paradigmen der Augmented Reality

Das am häufigsten verwendete Paradigma, das AR definiert, wurde 1997 von Ron Azuma [3] aufgestellt. Azuma definiert folgende Charakteristika: Wie auch bei Kinofilmen wird die *Realität mit der Virtualität kombiniert*. Im Kinofilm allerdings basiert dies auf einem zuvor aufgezeichneten

Video. In der AR wird ein Live-Videobild verwendet. Die eingefügten Gegenstände oder Charaktere sind in *3D registriert*. Diese Überlagerung muss in *Echtzeit* erfolgen. Das steht im Gegensatz zur Nachbearbeitung von Filmen. Dort steht zur Bearbeitung jedes einzelnen Bildes ein größerer Zeitraum zur Verfügung, um das Bild korrekt zu überlagern. Bei AR muss die Erzeugung des computergenerierten, perspektivisch korrekten Bildes in einem Zeitraum erfolgen, der geringer ist als der zeitliche Abstand zum nächsten Bild. Schließlich ist AR im Allgemeinen *interaktiv*, Nutzer können mit den zusätzlich eingefügten Objekten interagieren.

Desweiteren wurde von Milgram und Kishino [41] ein Kontinuum zwischen Realität und Virtualität definiert. An einem Ende des Kontinuums steht die *vollkommene Realität*, am anderen Ende die vollkommene Virtualität. In der vollkommenen Virtualität hält sich der Benutzer in einer *vollständig modellierten virtuellen Welt*, z.B. in einer CAVE [15], auf. Der Bereich dazwischen, die sog. *Mixed Reality*, ist charakterisiert durch den Grad der Virtualität. Die AR liegt näher bei der Realität, nur einzelne virtuelle Objekte werden in die reale Umgebung eingebettet. In Richtung zur Virtualität findet der Begriff *Augmented Virtuality* Verwendung. Hier ist die umgebende Welt überwiegend virtuell, nur einzelne Elemente entstammen der realen Welt oder Umgebung.

Zusätzlich sollte hier der Begriff *Ubiquitous Computing* nach Weiser [69] erwähnt werden, der später genauere Betrachtung finden wird. Weiser hat eine Welt skizziert, in der nicht mehr einzelne monolithische Computersysteme von Menschen bedient und genutzt werden, sondern stattdessen viele kleine Einheiten miteinander vernetzt sind. Jeder Gegenstand in dieser Welt ist mit Sensoren oder Aktuatoren ausgestattet, die miteinander kommunizieren können. Das ermöglicht eine neue Form von Interaktion. Der Benut-

zer kann überall auf die vorhandene Infrastruktur zugreifen und sie nutzen. Interessant wird dies für AR durch den Bedarf nach Sensoren zur genauen Positionsbestimmung in weitläufigen Anwendungen. Da viele verschiedene Sensoren, auch solche für die Positionsbestimmung des Nutzers, vorhanden sind, kann die Position des Benutzers auch über weite Strecken von verschiedenen Sensoren aufgenommen und an das AR-System weitergeleitet werden.

1.2 Geschichte

1965, fünf Jahre vor der Entwicklung des Personal Computers, postulierte Ivan Sutherland unter dem Titel *The Ultimate Display* [57], dass „Fortschritte in der Computertechnik es eventuell möglich machen können, die menschlichen Sinne mit virtuellen Erfahrungen zu überzeugen“. Anschließend entwarf und entwickelte er ein Display, das dem Nutzer das immersive Erfahren einer computergenerierten virtuellen 3D Welt ermöglicht. Bis 1968 entstand so das wohl erste Head-Mounted Display (HMD). Es war damals noch so schwer, dass es an der Decke des Raumes befestigt werden musste.

Diese Entwicklung kann als die Geburtsstunde der AR angesehen werden. Ein Display stand zur Verfügung, das dem Benutzer ermöglicht, abhängig zu seiner eigenen Position im Raum platzierte 3-D-Computergrafik aus seiner eigenen Perspektive zu betrachten. Über die Jahre entwickelte sich zunächst die Forschung im Bereich der Virtuellen Realität. Als logische Konsequenz sollte die reale Umwelt mit virtuellen Elementen angereichert werden können, die Erweiterte Realität entstand.

In den späten neunziger Jahren des 20. Jahrhunderts verstärkte die Forschung ihre Bemühungen um die AR. In

Japan wurde ein mehrjähriges Projekt zum Thema der Mixed Reality gestartet, in dem unter anderem zwei Symposien (International Symposium on Mixed Reality, ISMR) abgehalten wurden. Ungefähr zur selben Zeit wurde in den Vereinigten Staaten ein Workshop zum Thema Augmented Reality (IEEE Workshop on Augmented Reality, IWAR) ins Leben gerufen. Durch starke Beteiligung von europäischer, vor allem von deutscher Seite, fanden die jährlichen Konferenzen auch in Deutschland statt. Unter anderem sei hier das ARVIKA Projekt [1] zu nennen. Aus dem Workshop entwickelte sich ein Symposium, die ISAR. In Anlehnung an den Namen ISAR fand das Symposium zum ersten Mal im Jahr 2000 in München an der Isar statt. 2002 erfolgte der Zusammenschluss der beiden bisher getrennten Communities. So entstand das *International Symposium on Mixed and Augmented Reality* (ISMAR), welches seit 2002 jährlich abwechselnd auf einem der drei ursprünglich beteiligten Kontinente stattfindet. Informationen über alle Konferenzen [30] sind im Internet¹ verfügbar. Die Tagungsbände sind Quellen zahlreicher in diesem Buch genannter Referenzen.

1.3 Komponenten von AR-Systemen

Zu Realisierung von AR-Systemen werden verschiedene Komponenten benötigt, die sich in drei Bereiche einteilen lassen: *Darstellung*, *Tracking* und *Interaktion*.

Um virtuelle Objekte darzustellen, wird Software zur Erzeugung von 3-D-Computergrafik benötigt. Diese Grafik wird auf einem geeigneten Display angezeigt. Displays können Computerbrillen sein, konventionelle Monitore oder Videoprojektionswände, die allesamt als Fenster in die erweiterte Welt dienen. Zur Erzeugung von 3-D-Computergrafik

¹ ISMAR Konferenzen: <http://www.ismar-conf.org/>

gibt es eine Vielzahl von 3-D-Rendering-Bibliotheken, die meist auf OpenGL oder DirectX basieren. Kapitel 2 vertieft dieses Thema. Es enthält eine Übung, die illustriert, wie 3-D-Objekte in realen, aber noch statischen, Szenen korrekt mit virtuellen 3-D-Objekten überlagert werden.

Nun ist keine Szenerie statisch – der Betrachter und andere Objekte können sich bewegen. In Kapitel 3 werden deshalb Prinzipien und Technologien des Trackings erläutert und grundlegende Methoden gezeigt, die notwendig sind, um Trackingsysteme für AR zu verwenden. Wahrnehmungsprobleme, die in Zusammenhang mit der Überlagerung der Realität auftreten, werden angesprochen und diskutiert. Die zugehörige Übung integriert ein allgemein verfügbares Trackingsystem in das zuvor verwendete Darstellungssystem. Als Ergebnis können dynamische Szenen mit AR überlagert werden, der Nutzer kann mit einer Webcam den Blickwinkel auf Szenen verändern und überlagerte Objekte in der Szene verschieben.

Anschließend werden weitere Möglichkeiten für Eingabe und Interaktion in Kapitel 4 vorgestellt. Nach einem Überblick über Prinzipien der Interaktion entwickelt die zugehörige Übung eine Anwendung, die den Benutzer mit einer Teekanne Tee in eine Tasse einschenken lässt.

Im weiteren Verlauf des Buches werden verschiedene existierende AR-Anwendungen vorgestellt, die einen Überblick über die Möglichkeiten geben, die durch AR geschaffen werden. Das Kapitel soll inspirieren und neue Ideen für eigene Anwendungen geben.

Das Buch schließt mit einem Ausblick, der unter anderem auf das Konzept der Ubiquitären Erweiterten Realität eingeht. Dabei wird angesprochen, wie in mobilen Anwendungen unterschiedliche Trackingsysteme verbunden werden können, um weitläufige AR zu ermöglichen.

1.4 Empfohlene Voraussetzungen

Zum Verständnis der in diesem Buch beschriebenen mathematischen Hintergründe sollten Kenntnisse in Linearer Algebra und Analytischer Geometrie vorhanden sein. Ein allgemeines Verständnis im Umgang mit Vektoren und Matrizen ist empfehlenswert.

Die technische Umsetzung in diesem Buch basiert auf OpenGL. Um die Übungen selbst zu bearbeiten, muss in C/C++ programmiert werden. Der Leser sollte das Programmieren in dieser Sprache beherrschen und Sourcecode selbstständig kompilieren können. Benötigt werden dazu die Bibliotheken zu OpenGL und Glut. Falls diese nicht vorhanden sind, können sie kostenfrei aus dem Internet heruntergeladen werden.

Um die Übungen zu realisieren, wird zusätzlich eine Webcam benötigt. Themen, die über den Stoff der Übungen hinausgehen, benötigen oft weitere Soft- und Hardware. Falls dem so ist, wird in Fußnoten, wenn möglich, auf Open Source-Implementierungen verwiesen.

Darstellung virtueller Objekte

Dieses Kapitel behandelt die zwei für die Darstellung von virtuellen Objekten notwendigen Hauptelemente, die softwaretechnischen Grundlagen für den Umgang mit räumlichen Strukturen und die Hardware, auf der die Ausgaben dargestellt werden. Zunächst einmal werden fundamentale Grundlagen der Computergrafik vorgestellt, anschließend werden weiterführende Konzepte wie Szenengraphen angesprochen. Aufbauend auf der Erstellung von 3-D-Objekten werden verschiedene Anzeigesysteme und -displays vorgestellt. Daran anschließend vertieft eine Übung den bearbeiteten Stoff. Dabei wird der Monitor des eigenen Rechners zu einem AR-Display gemacht. Die Übung überlagert ein statisches Bild mit einer AR-Darstellung. Das Kapitel schließt mit einem Überblick über nicht-visuelle Anzeigen, wobei vor allem auf akustische und haptische Anzeigen eingegangen wird.

2.1 3-D-Rendering

Renderingsysteme beschreiben einen 3-D-Raum, der durch eine Projektion auf einen zweidimensionalen Bildschirm übertragen wird. Diese Aufgabe übernimmt eine *virtuelle Kamera*.

2.1.1 Virtuelle Kamera

Eine virtuelle Kamera wird durch eine räumliche Transformation, die ihren Ort und ihre Lage (extrinsische Kameraparameter) im Raum bestimmt, und durch ihre Abbildungseigenschaften (intrinsische Kameraparameter) beschrieben.

Dieser Abschnitt vertieft zunächst die intrinsischen Kameraparameter. Die Kamera bleibt dabei am Ursprung des Koordinatensystems, wir betrachten ausschließlich, wie der gezeigte Raum abgebildet wird. Abbildungseigenschaften können orthogonale (parallele) Projektionen oder perspektivische Projektionen auf die 2-D-Bildebene sein. Für AR ist vorwiegend die perspektivische Abbildung von Interesse. Bei der perspektivischen Abbildung treffen sich alle Strahlen von Punkten im Raum in einem Punkt, der Lochkamera (pinhole camera). Eine virtuelle Lochkamera hat keine Verzerrungen, wie sie üblicherweise bei echten Kameras durch die Optik entstehen.

Da das Sichtfeld einer virtuellen Kamera auf einen rechteckigen Bereich auf dem Bildschirm projiziert werden soll, gleicht der Sichtbereich einer Pyramide, an deren Spitze die Lochkamera liegt. Abbildung 2.1 zeigt die pyramidenartige Form des Sichtbereiches mit Standort der Kamera am Ursprung. Üblicherweise blickt die Kamera in vielen Renderingsystemen, auch in OpenGL, entlang der negativen Z-Achse.

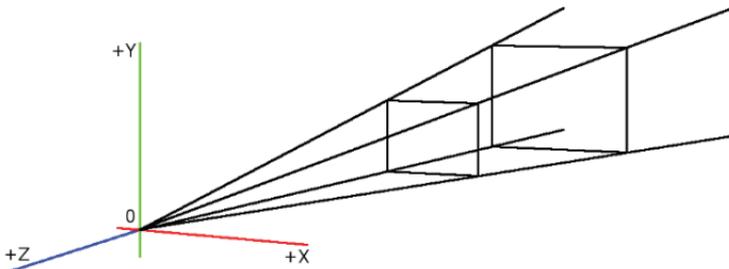


Abb. 2.1. Darstellung des Sichtbereiches und des *Frustums* einer Kamera, die am Ursprung eines Koordinatensystemes platziert ist

In der Computergrafik wird der pyramidenförmige Sichtbereich meist zusätzlich eingeschränkt. Alle Objekte, die näher als die sog. *near clipping plane* oder weiter weg als die *far clipping plane* liegen, werden nicht dargestellt, diese Ebenen beschneiden (Engl. clip) den Darstellungsraum. Dies hat einerseits Vorteile in der Performanz, weit entfernte Objekte sind nicht sichtbar und würden unnötigen Rechenaufwand erzeugen. Andererseits kann durch die Einschränkung der Darstellungstiefe die Genauigkeit der Tiefendarstellung beeinflusst werden. Je kleiner der Abstand der beiden clipping planes, desto höher auflösend ist die Tiefendarstellung. Wie Abb. 2.1 zeigt, wird der Sichtbereich damit zu einem Pyramidenstumpf, dem Frustum.

Darstellen lässt sich die Projektion dieses 3-D-Bereiches mittels einer sog. *Projektionsmatrix*. Die Projektionsmatrix zur Darstellung eines Frustums ist

$$\begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

wobei

- n, f Abstand zu near and far clipping plane
(beide positiv)
- l, r X-Koordinate der linken bzw. rechten
vertikalen Kante der near clipping plane
- b, t Y-Koordinate der unteren bzw. oberen
horizontalen Kante der near clipping
plane

sind.

Die Blickachse eines Frustum liegt parallel zur Z-Achse. Ist das Frustum symmetrisch um diese Achse, werden die beiden oberen Einträge in der dritten Spalte zu 0. Gesetzt wird ein perspektivisches Frustum in OpenGL durch die Funktionsaufrufe

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum(left, right, bottom, top, zNear,
          zFar : glDouble) };
glMatrixMode(GL_MODELVIEW);
```

Der erste Funktionsaufruf gibt an, dass alle weiteren Operationen mit Matrizen auf die Projektionsmatrix ausgeführt werden. Im nächsten Funktionsaufruf wird die Einheitsmatrix gesetzt. Der Aufruf von `glFrustum` setzt die Projektionsparameter entsprechend der übergebenen Werte. Abschließend wird die zu bearbeitende Matrix auf die Model-View Matrix zurückgestellt. Mit der Model-View-