

Interaktive und stereoskopische Visualisierung in der speziellen Relativitätstheorie

DISSERTATION

zur Erlangung des Grades eines Doktors
der Naturwissenschaften

der Fakultät für Mathematik und Physik
der Eberhard-Karls-Universität Tübingen

vorgelegt von

Marc P Borchers

aus Münster

2005

Tag der mündlichen Prüfung: 13.05.2005
Dekan: Prof. Dr. Peter Schmid
1. Berichterstatter: Prof. Dr. Hanns Ruder
2. Berichterstatter: Prof. Dr. Thomas Ertl

Zusammenfassung

In der vorliegenden Arbeit wird ein erweitertes Verfahren zur bildbasierten relativistischen Visualisierung vorgestellt. Die gegenüber bisherigen Implementierungen wesentlich gesteigerte Geschwindigkeit wird durch die Verwendung der programmierbaren Vertex- und Pixeleinheiten der Grafikkhardware erreicht. Alle Berechnungen, von der Erstellung der Tetraeder Map oder Cubemap bis hin zur relativistischen Umrechnung mit korrekter Darstellung des Doppler- und Searchlighteffekts, laufen vollständig auf der Grafikkhardware ab. Auf einer einzelnen GPU lassen sich so für Umgebungen mit über 100000 Polygonen Bildraten von 60 FPS erreichen, für die Darstellung komplexerer Modelle lässt sich das Verfahren leicht auf einem Rendercluster parallelisieren.

Neben der relativistischen Umrechnung ist eine pixelgenaue Geometriekorrektur und Helligkeitsanpassung, wie sie beispielsweise bei der Darstellung mit mehreren Projektoren oder bei Verwendung einer gekrümmten Projektionsfläche benötigt wird, in das Verfahren integriert.

Das vorgestellte Verfahren eignet sich damit auch zur interaktiven stereoskopischen Visualisierung, bei der jeweils zwei Bilder für die beiden Augen erzeugt werden müssen. Die Laufzeitdifferenzen des Lichts zwischen den beiden Augen beeinflussen die Entfernungswahrnehmung bei binokularer Beobachtung. Schnell bewegte Punkte werden nicht in der wirklichen Entfernung wahrgenommen. Die bei den Arbeiten zum Aussehen schnell bewegter Körper angegebene Verdrehung lässt sich bei stereoskopischer Beobachtung quer zur Bewegungsrichtung als Kombination aus einer dreidimensionalen Drehung und einer Stauchung des Körpers sehen. Beim Blick in Bewegungsrichtung ist die binokular wahrgenommene Entfernung konsistent mit der durch die Aberration scheinbar vergrößerten Entfernung eines Körpers.

Inhaltsverzeichnis

1	Einführung	1
2	Visualisierung speziell-relativistischer Effekte	6
2.1	Visuelle Information	7
2.2	Transformation der Strahldichte	9
2.3	Strahlungsbewertung	10
2.4	Modelle für Spektralverteilungen	14
2.5	Visualisierung der spektralen Strahldichte	16
2.6	Techniken der relativistischen Visualisierung	18
2.6.1	Geometriebasierte Verfahren	18
2.6.2	Bildbasierte Verfahren	20
2.6.3	Raytracing	22
2.7	Bewegungssimulation	23
3	Dreidimensionale Visualisierung	25
3.1	Entfernungswahrnehmung	25
3.2	Stereogeometrie	27
3.3	Technik der stereoskopischen Visualisierung	30
3.3.1	Verfahren zur Bildkanaltrennung	30
3.3.2	Erzeugung der Teilbilder	31
3.4	Entfernung und Aussehen schnell bewegter Körper	33
3.4.1	Bewegung parallel zu den Augen	36
3.4.2	Bewegung senkrecht zur Verbindungslinie der Augen	41
3.4.3	Beobachtung mit mitbewegtem Kopf	44
3.4.4	Beschleunigte Bewegungen	45
3.5	Entfernungsbestimmung bei rel. Geschwindigkeiten	46
3.6	Beispiele	48
4	Projektion auf gekrümmte Flächen	52
4.1	Entzerrung	53
4.2	Anpassung der Leuchtdichte	56
4.3	Anwendungen	58
4.3.1	Kuppelprojektion	58
4.3.2	Interaktive Verzerrung	61
4.3.3	Stereoskopische Visualisierung	62

5	Implementierung und Ergebnisse	63
5.1	Teilschritte	64
5.1.1	Erzeugung der Rundumsicht	64
5.1.2	Geometrieanpassung	68
5.1.3	Aberration und Sampling der Rundumsicht	68
5.1.4	Doppler- und Searchlighteffekt	69
5.1.5	Tone Mapping	69
5.1.6	Anpassung der Leuchtdichte und Farbe bei Projektion	70
5.1.7	Materialabhängige Spektralverteilungen	70
5.2	Stereoskopische Darstellung	71
5.3	Bildqualität	72
5.4	Geschwindigkeit	76
5.5	Optimierungen	78
5.6	Beispiele	80
5.6.1	Virtual Tübingen	80
5.6.2	Relativistischer Flugsimulator	84
5.6.3	Interaktive Verzerrung von Panoramaaufnahmen	86
6	Ausblick	88
	Anhang	90
A.1	Invarianz des Phasenraumvolumens	90
A.2	Projektion auf ebene Flächen	92
A.3	Projektion für Tetraeder Maps	94
A.4	Dokumentation Vertex- und Pixelshader	95
A.4.1	Aberration	95
A.4.2	Geometrieanpassung	95
A.4.3	Vollständige relativistische Umrechnung	96
A.4.4	Geometriebasierte Verzerrung	98
A.5	Interaktives schwarzes Loch	100
	Literaturverzeichnis	103

Kapitel 1

Einführung

Visualisierung ist die grafische Aufbereitung und Darstellung von Informationen auf eine Art und Weise, die dem Benutzer die Information leichter oder überhaupt erst zugänglich macht. Dies trifft in besonderem Maß auf die Visualisierung im Rahmen der Relativitätstheorie zu, deren Effekte erst bei sehr hohen Geschwindigkeiten und Gravitationsfeldern sichtbar werden und deshalb der menschlichen Wahrnehmung bei den erreichbaren Geschwindigkeiten nicht direkt zugänglich sind. Hier ist die Simulation in Kombination mit der Visualisierung der relativistischen Effekte die einzige Möglichkeit, die vom alltäglichen dreidimensionalen Raum abweichende Struktur der vierdimensionalen Raumzeit der Wahrnehmung zugänglich zu machen.

Die Fortschritte auf dem Gebiet der Visualisierung mit Hilfe schneller Computer und Grafikhardware erlauben es heute, die relativistischen Effekte grafisch ansprechend und korrekt darzustellen und so schnell zu erzeugen, dass der Benutzer die Raumzeit interaktiv erkunden kann. Dabei können einzelne Aspekte der Raumzeit wie beispielsweise die Längenkontraktion oder die Zeitdilatation verdeutlicht werden, oder es wird versucht, das Aussehen von Körpern bei relativistischen Geschwindigkeiten möglichst realistisch wiederzugeben.

Das Thema dieser Arbeit ist die interaktive Simulation und schnelle Darstellung des optischen Erscheinungsbildes mit hoher Bildqualität für einen schnell bewegten Beobachter beim Flug durch virtuelle Szenerien unter Ausnutzung der programmierbaren Vertex- und Pixeleinheiten der Grafikhardware.

Bisherige Ansätze

Das wahrnehmbare oder fotografierbare Aussehen schnell bewegter Körper wird erstmals 1924 von Lampa[25] anhand des visuellen Erscheinungsbildes eines schnell bewegten Stabes untersucht. Penrose[42] und Terrell[56] zeigen 1959 die Unsichtbarkeit der

Lorentzkontraktion für die Kugel und Körper mit kleiner Winkelausdehnung. Weitere Arbeiten[7, 48, 49, 65] beschäftigen sich mit dem Aussehen ausgedehnter Körper bei relativistischen Bewegungen. Der Begriff der Sichtbarkeit der Lorentzkontraktion stiftet auch in jüngster Zeit noch Verwirrung[30]. Den Einfluss des relativistischen Dopplereffekts und des Searchlighteffekts auf das visuelle Erscheinungsbild untersuchen McKinley[33], Chang, Lai und Chen[9] und Weiskopf, Kraus und Ruder[59].

Den Arbeiten zum Aussehen schnell bewegter Körper ist die Interpretation des wahrgenommenen Bildes als das Bild eines entsprechend gedrehten, aber unverzerrten Körpers gemeinsam. Bei der Beobachtung mit einem Auge oder einer Kamera kann dieses Bild nicht vom Bild eines kontrahierten und gescherten Körpers unterschieden werden. Die stereoskopische Beobachtung mit zwei Augen sollte die Entfernungsbestimmung ermöglichen, so dass die scheinbare dreidimensionale Form des Körpers bestimmt werden kann. Terrell[56] sowie Scott und van Driel[48] vermuten, dass so der kontrahierte und entlang der Beobachtungsrichtung gescherte Photokörper dreidimensional sichtbar sein sollte. Durch die unterschiedlichen Laufzeiten des Lichts zu den beiden Augen wird jedoch das von einem Punkt ausgesandte Licht, das die Augen für den Beobachter gleichzeitig erreicht, an unterschiedlichen Punkten ausgestrahlt. Die wahrgenommene Entfernung als Schnittpunkt der Sichtlinien stimmt dann im Allgemeinen nicht mit der wirklichen Entfernung überein. McGill[32] zeigt 1968, dass ein Stab bei stereoskopischer Beobachtung gedreht erscheint. Boas, Calhoun und Horan[8] untersuchen 1971 den Einfluss schneller Bewegungen auf die binokulare Beobachtung für einzelne Punkte und deren Trajektorien, sowie für ausgedehnte Körper bei einer Bewegung orthogonal zur Verbindungslinie der Augen oder Kameras.

Der Einsatz von Computern als Hilfsmittel zur Visualisierung der relativistischen Effekte wird 1989 von Hsiung und Dunn[22] anhand eines relativistischen Raytracings beschrieben. Hsiung[23] und Gekelman[19] führen 1991 das geometriebasierte Verfahren auf Polygonbasis ein. Eine schnelle Implementierung wird 1998 von Betts[6] vorgestellt und von Weiskopf, Kraus und Ruder[61] korrigiert. Das texturbasierte Verfahren zur Darstellung der relativistischen Verzerrungen wird 1999 von Weiskopf[60] vorgestellt. Die Technik der bildbasierten relativistischen Umrechnung am Ende der Renderpipeline wird 2000 von Weiskopf[63, 62] eingeführt.

Virtual Reality Visualisierungssysteme wie die CAVE[10], PowerWall[57] oder blue-c[21] und die dabei eingesetzten Verfahren zur Anpassung der Bilddaten an die Geometrie der Projektionen sind mittlerweile weit verbreitet. Eine Anpassung der Daten für nicht feinjustierte Projektoren und gekrümmte Projektionsflächen mit Hilfe der programmierbaren Grafikhardware wird 2004 von Raskar, van Baar und Willwacher[45] vorgestellt.

Zielsetzung

Die enorm erweiterte Funktionalität der aktuellen Generation von Grafikkarten mit frei programmierbaren Vertex- und Pixeleinheiten eröffnet der Visualisierung des Erschei-

nungsbildes bei relativistischen Bewegungen neue Möglichkeiten. Ziel dieser Arbeit ist die Entwicklung von schnellen Algorithmen zur interaktiven und stereoskopischen Visualisierung bei speziell-relativistischen Bewegungen und deren Implementierung auf aktueller PC-Hardware. Das Verfahren soll insbesondere auf bestehende 3D-Modelle einer virtuellen Umgebung mit einer Komplexität im Bereich von 20000 bis 100000 Polygonen angewendet werden können und für diese Modelle interaktive Bildraten von mehr als 30 Bildern pro Sekunde bei hoher Bildqualität liefern, wobei neben der reinen Verzerrung auch der relativistische Doppler- und Searchlighteffekt berücksichtigt werden soll.

Für die Darstellung mit Hilfe mehrerer Projektoren oder auf einer gekrümmten Projektionsfläche z.B. bei einer Kuppelprojektion soll die dafür notwendige Anpassung der Bilder an die Geometrie der Projektionsflächen und die Aufstellung der Projektoren modular in das Verfahren integriert sein.

Schließlich soll der Einfluss der relativistischen Effekte auf die Entfernungsbestimmung bei stereoskopischer Wahrnehmung für beliebige Relativbewegungen untersucht werden und mit Hilfe der entwickelten interaktiven Visualisierung für ausgedehnte Körper dargestellt werden.

Was ist neu ?

Diese Arbeit stellt ein erweitertes Verfahren zur bildbasierten Visualisierung speziell-relativistischer Effekte vor. Die gegenüber bisherigen Verfahren wesentlich erhöhte Geschwindigkeit wird durch die Verwendung der programmierbaren Vertex- und Pixel-einheiten der Grafikkarte erreicht, so dass die gesamte relativistische Umrechnung für jeden einzelnen Bildpunkt vollständig auf der Grafikkarte erfolgen kann.

Das Verfahren kann als Aufsatz für bestehende klassische Renderengines benutzt werden, wobei diese lediglich die blickrichtungsunabhängige Erzeugung einzelner Bilder mit definierbarem Bildausschnitt erlauben müssen. Je nach Renderengine brauchen während der Laufzeit weder Vertex- noch Texturdaten auf die Grafikkarte übertragen werden. Von der CPU werden nur die aktuellen Beobachtertransformationen und Sichtbarkeitsinformationen für ganze Objekte oder Objektgruppen sowie die Parameter für die programmierbaren Pixeleinheiten übertragen. Die relativistische Umrechnung kann mit dieser Technik auf aktueller Hardware auf einem einzelnen Rechner in Abhängigkeit der berücksichtigten Effekte in 2 ms bis maximal 16 ms erfolgen. Damit kann die Umrechnung auch in Echtzeit auf Videoquellen, die eine Rundumsicht liefern, angewendet werden.

Darüber hinaus besteht die Möglichkeit, den Doppler- und Searchlighteffekt korrekt darzustellen, wenn speziell für die relativistische Visualisierung erstellte Modelle verwendet werden. Die verschiedenen Materialeigenschaften werden bei diesen Modellen durch die Angabe der spektralen Eigenschaften in einem weiten Bereich charakterisiert

und nicht nur durch die Angabe der RGB-Farbwerte. Erst mit dieser Information ist eine realistische Berechnung des Aussehens bei schnellen Bewegungen möglich.

Die Anpassung der einzelnen Bilder für die Darstellung mit mehreren Projektoren und bei der Projektion auf gekrümmte Flächen ist zusammen mit der bildbasierten relativistischen Umrechnung im Pixelshader integriert. Die Umrechnung erfolgt zusammen mit der relativistischen Verzerrung auf Pixelebene, so dass auch für die Geometrie-anpassung keine feine Unterteilung des zugrunde liegenden 3D-Modells notwendig ist. Die zusätzlich benötigte Zeit für die Korrektur liegt unter 1.5 ms . Spezielle Hardware, die die Geometrieverzerrung auf das Videosignal anwendet, wird damit überflüssig. Die Verzerrungen sind nicht auf polynomiale Funktionen beschränkt und können während der Laufzeit beliebig vorgegeben werden, so dass sich die Korrektur bei Bedarf leicht interaktiv oder automatisch einstellen und mit der Benutzerposition verändern lässt.

Die schnelle Visualisierung ermöglicht die interaktive Darstellung der relativistischen Effekte auch bei binokularer Betrachtung, indem die Bilder für die Positionen der beiden Augen getrennt erzeugt und den Augen einzeln zugeführt werden. Damit lässt sich der Einfluss der bei schnellen Bewegungen auftretenden Verzerrungen auf die Tiefenwahrnehmung quasi experimentell untersuchen. Die theoretisch gewonnenen Ergebnisse für das dreidimensionale Erscheinungsbild schnell bewegter Körper bei verschiedenen Relativbewegungen, wie beispielsweise die dreidimensional wahrnehmbare Verdrehung, können so direkt in der Simulation erfahren werden.

Überblick über die vorliegende Arbeit

Kapitel 2 fasst die Grundlagen der relativistischen Visualisierung zusammen und stellt die verschiedenen Verfahren zur Berechnung der Verzerrungen vor. Die Umrechnung der spektralen Strahldichte zwischen zwei Systemen als Grundlage des bildbasierten Verfahrens wird beschrieben, ebenso wie verschiedene Modelle für Spektralverteilungen und die Darstellung einer berechneten Spektralverteilung mit Hilfe eines Monitors oder Projektors. Ein Überblick über die verschiedenen Techniken zur relativistischen Visualisierung und Simulation bildet den Abschluss des Kapitels.

Kapitel 3 gibt einen Überblick über die verschiedenen Mechanismen der Entfernungswahrnehmung und ihre Unterstützung bei der Visualisierung. Besonderes Augenmerk liegt dabei auf der stereoskopischen Tiefenwahrnehmung, die den stärksten Eindruck räumlicher Tiefe liefert, und der 3D-Stereo-Visualisierung. Der Einfluss der relativistischen Effekte auf die stereoskopische Entfernungsbestimmung bei verschiedenen Relativbewegungen von Beobachter und Objekt wird untersucht. Durch die, je nach betrachtetem System, unterschiedliche Laufzeit beziehungsweise Aberration erscheint ein bewegter Punkt nicht in der wirklichen Entfernung. In Abhängigkeit von der Relativgeschwindigkeit lässt sich eine Abbildung des dreidimensionalen Raumes auf sich selbst definieren, die jedem Punkt die scheinbare Position bei stereoskopischer Beobachtung zuordnet. Die lokalen Eigenschaften dieser Abbildung beschreiben dann die dreidimen-

sionale Wahrnehmung schnell bewegter Körper bei stereoskopischer Betrachtung.

Bei der Darstellung der berechneten Bilder auf einer Projektionsfläche soll diese dem Benutzer als Fenster in die virtuelle Welt erscheinen, und selbst möglichst wenig als Fläche wahrgenommen werden. Dies gelingt am besten, wenn dem Benutzer die Objekte in den gleichen Richtungen präsentiert werden, unter denen sie einem Beobachter in der virtuellen Welt erschienen. Kapitel 4 beschreibt die notwendige Anpassung der Geometrie und der Helligkeit von Bildern bei der Projektion auf gekrümmte Flächen und stellt Beispiele für die schnelle Verzerrung von Bildern vor.

Kapitel 5 beschreibt die Implementierung des bildbasierten Verfahrens für die interaktive relativistische Visualisierung inklusive Searchlight- und Dopplereffekt sowie der Anpassung der Geometrie und Helligkeit für die Projektion auf gekrümmte Flächen. Neben der Diskussion der Bildqualität des implementierten Verfahrens und der möglichen Erweiterungen für komplexere Szenerien und Beleuchtungsmodelle werden aktuelle Anwendungen des Verfahrens vorgestellt.

Kapitel 6 gibt einen kurzen Ausblick auf geplante Erweiterungen des schnellen bildbasierten Verfahrens und zukünftige Anwendungen mit mehreren GPUs und Rechnern, sowie auf mögliche weitere Arbeiten bei der Simulation relativistischer Bewegungen und bei der Verallgemeinerung auf gekrümmte Raumzeiten.

Kapitel 2

Visualisierung speziell-relativistischer Effekte

Die Struktur der vierdimensionalen flachen Raumzeit und die sich daraus ergebenden Effekte der speziellen Relativitätstheorie auf die Physik können auf vielfältige Art visualisiert werden. Unter *Visualisierung speziell-relativistischer Effekte* soll in dieser Arbeit immer die Visualisierung des Erscheinungsbildes einer Umgebung für einen Beobachter bei Bewegungen mit relativistischen Geschwindigkeiten verstanden werden, also die Simulation des optischen Strahlungsfeldes, die ein Beobachter mit dem Auge oder einer Kamera messen kann.

Die rasante Entwicklung der Computer und Grafikhardware in den letzten Jahren hat es ermöglicht, diese visuellen Effekte der speziellen Relativitätstheorie interaktiv zu berechnen und darzustellen. In diesem Kapitel werden die Grundlagen und die wichtigsten Verfahren der Visualisierung der relativistischen Effekte vorgestellt. Einführungen in die spezielle Relativitätstheorie und die Computervisualisierung finden sich beispielsweise in [5, 16, 35, 46, 47].

Modellierung

Als Modell für das Strahlungsfeld wird durchgehend die geradlinige Bewegung einer Menge von Photonen im Vakuum mit jeweils definierter Energie und lichtartigem Impulsvektor angenommen. Die Polarisation und Welleneigenschaften des Lichts werden vollständig vernachlässigt, auch wenn bei der Diskussion der durch die Energie der Photonen bestimmten Farbwahrnehmung und der Transformation der spektralen Strahlendichte die übliche Darstellung in Abhängigkeit von der Wellenlänge verwendet wird.

Die räumlichen Ausdehnungen des Beobachters oder einer Kamera als abbildende Systeme sowie die Integrationszeit für die Bildaufnahme werden im Vergleich zu allen

anderen auftretenden Größen als so klein angenommen, dass sie vernachlässigt werden können. Augen und Kameras werden in den meisten Fällen als einfache Lochkameras modelliert, die Begriffe Auge und Kamera werden im Folgenden austauschbar verwendet.

Zunächst soll die Erzeugung eines einzelnen Bildes für den Beobachter an einem bestimmten Punkt der Raumzeit betrachtet werden. Eine interaktive Simulation oder ein Film ist dann nichts als eine Folge von einzelnen Bildern, die für verschiedene Punkte der Raumzeit generiert werden. Um den korrekten zeitlichen Eindruck für einen schnell bewegten Beobachter wiederzugeben, müssen die Punkte auf der Weltlinie des Beobachters so gewählt werden, dass sie einen konstanten Abstand in der Eigenzeit des Beobachters aufweisen, beziehungsweise dass die simulierte Eigenzeit der wirklichen Zeit des Benutzers entspricht.

2.1 Visuelle Information

Bei der Erzeugung des Bildes für einen bestimmten Punkt E in der Raumzeit stellt sich zunächst die Frage, welche visuelle Information ein Beobachter an diesem Punkt prinzipiell erhalten kann. Für die Beobachtung verfügbar sind diejenigen Photonen, deren Weltlinie E enthält und die von Ereignissen auf dem in die Vergangenheit von E gerichteten Lichtkegel \mathcal{L}_E^- stammen.

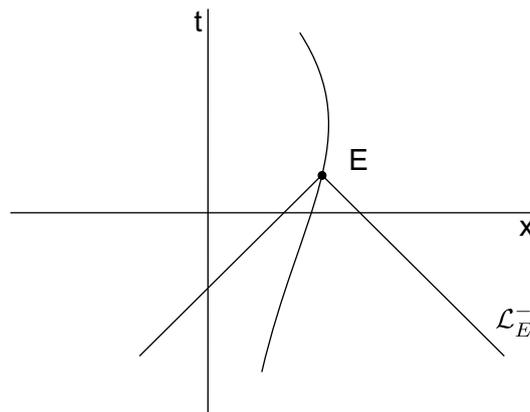


Abbildung 2.1: Die für einen Beobachter in E sichtbaren Photonen stammen von Ereignissen auf dem in die Vergangenheit von E gerichteten Lichtkegel \mathcal{L}_E^- .

Da die Geschwindigkeit und die Beschleunigung des Beobachters keinen Einfluss auf den Lichtkegel \mathcal{L}_E^- haben, ist die prinzipiell messbare optische Information in E unabhängig vom Bewegungszustand des Beobachters. Ist sie für einen Beobachter vollständig¹ bekannt, so ist sie auch für alle weiteren Beobachter, die sich mit einer anderen Geschwindigkeit bewegen, festgelegt. In diesem und den folgenden Abschnitten wird dargestellt, wie die visuelle Information beschrieben, berechnet und dargestellt werden kann.

Das Strahlungsfeld lässt sich als Dichte der Photonen im Phasenraum beschreiben, also als Anzahl der Photonen in einem kleinen Volumen um einen Punkt in der Raumzeit mit einem Impuls in einem Impulsvolumen um den Impuls p , dividiert durch das Volumen im Phasenraum:

$$\mathcal{N}(E, p) = \frac{dN}{dV dp} \quad (2.1)$$

Beschränkt man sich auf ein Ereignis E , so ist die gesamte optische Information als Photonendichte in Abhängigkeit des Photonenimpulses darstellbar. In der Visualisierung, Optik und Astronomie wird statt der Photonendichte üblicherweise die äquivalente Beschreibung des Strahlungsfeldes durch die spektrale Strahldichte verwendet, die den Energiefluss in einen bestimmten Raumwinkel pro Flächeneinheit und Wellenlängenintervall angibt. Der Zusammenhang zwischen der spektralen Strahldichte I_λ und der Photonendichte für einen Beobachter ist durch

$$I_\lambda(\lambda, \mathbf{n}) = \frac{h^4 c^2}{\lambda^5} \mathcal{N}(p) \quad (2.2)$$

gegeben, wobei \mathbf{n} ein Einheitsvektor in Richtung des Photonenimpulses im Bezugssystem des Beobachters B ist. λ sei die vom Beobachter gemessene Wellenlänge, so dass

$$p = \frac{h}{\lambda}(1, \mathbf{n}) \quad (2.3)$$

die Darstellung des Impulses im Bezugssystem von B ist.

Die optische Information in einem Punkt der Raumzeit ist vollständig bestimmt, wenn die spektrale Strahldichte für alle Richtungen und alle Wellenlängen bekannt ist. Die Funktion, die für einen Beobachter jeder Richtung eine spektrale Verteilung oder eine vereinfachte Beschreibung der spektralen Verteilung durch mehrere Zahlenwerte zuordnet, soll im Folgenden als *Rundumsicht* bezeichnet werden.

¹Vollständig heißt im relativistischen Fall, dass die spektrale Strahldichte zunächst für den gesamten Raumwinkel und alle Wellenlängen bekannt sein muss. Erst unter bestimmten Voraussetzungen ist ein kleinerer Bereich ausreichend (vgl. 2.3)

Lässt man die Beschränkung auf einen Punkt der Raumzeit fallen, und liegt die spektrale Strahldichte für alle Punkte, Richtungen und Wellenlängen vor, so wird die Funktion auch als plenoptische Funktion bezeichnet [2].

2.2 Transformation der Strahldichte

Ist die spektrale Strahldichte für einen Beobachter B in einem Ereignis der Raumzeit bekannt, so kann die Strahldichte für einen weiteren Beobachter B' , der sich mit der Geschwindigkeit \mathbf{v} relativ zum ersten Beobachter bewegt, umgerechnet werden. Es kann gezeigt werden, dass die Phasenraumdicke der Photonen unabhängig vom Bezugssystem ist [35, 27]:

$$\mathcal{N}'(p) = \mathcal{N}(p) \quad (2.4)$$

Ein Beweis ist in Anhang A.1 angegeben. Mit 2.2 ergibt sich für die Umrechnung der spektralen Strahldichte

$$I'_\lambda(\lambda', \mathbf{n}') = D(\boldsymbol{\beta}, \mathbf{n})^5 I_\lambda(\lambda, \mathbf{n}) \quad (2.5)$$

wobei der Zusammenhang zwischen λ und λ' sowie \mathbf{n} und \mathbf{n}' mit 2.3 durch die Darstellung von p in den beiden Bezugssystemen bestimmt wird

$$\begin{aligned} \lambda' &= \gamma(1 - \boldsymbol{\beta} \cdot \mathbf{n}) \lambda \\ \mathbf{n}' &= \mathbf{n} + \frac{\gamma^2}{\gamma + 1} \boldsymbol{\beta} \cdot \mathbf{n} \boldsymbol{\beta} - \gamma \boldsymbol{\beta} \end{aligned} \quad (2.6)$$

und D der von der Richtung abhängige Dopplerfaktor ist

$$D(\boldsymbol{\beta}, \mathbf{n}) = \frac{\lambda}{\lambda'} = \gamma(1 - \boldsymbol{\beta} \cdot \mathbf{n}) \quad (2.7)$$

$\boldsymbol{\beta}$ und γ stehen wie üblich für \mathbf{v}/c und $1/\sqrt{1 - v^2/c^2}$.

Gleichung 2.5 stellt den gesuchten Zusammenhang zwischen den in den beiden Bezugssystemen messbaren Strahldichten her. Die Strahldichte für den Beobachter B' aus einer bestimmten Richtung \mathbf{n}' bei einer bestimmten Wellenlänge λ' wird durch die für B berechnete oder gemessene Strahldichte aus der Richtung \mathbf{n} bei der Wellenlänge λ

vollständig bestimmt. Zusammen mit 2.6 und 2.7 enthält die Umrechnung der Strahldichte nach 2.5 den relativistischen Searchlight- und Dopplereffekt sowie die Aberration und bildet die Grundlage der bildbasierten relativistischen Visualisierung.

Wählt man die Relativbewegung entlang der x -Achse, so erhält man mit $D = \gamma(1 + \beta n_x)$ die für die schnelle Berechnung geeignete Form

$$\begin{pmatrix} n'_x \\ n'_y \\ n'_z \end{pmatrix} = \frac{1}{D} \begin{pmatrix} \gamma(\beta + n_x) \\ n_y \\ n_z \end{pmatrix} \quad (2.8)$$

Bezeichnet α den Winkel zwischen der Bewegungsrichtung und der Beobachtungsrichtung, so ergeben sich mit $n_x = \cos \alpha$ die bekannten Formeln für die relativistische Aberration und den Dopplereffekt:

$$\begin{aligned} \cos \alpha' &= \frac{\beta + \cos \alpha}{1 + \beta \cos \alpha} \\ \lambda' &= \gamma(1 + \beta \cos \alpha)\lambda \end{aligned} \quad (2.9)$$

2.3 Strahlungsbewertung durch eine Kamera oder das Auge

Mit den im letzten Abschnitt angegebenen Formeln kann das Strahlungsfeld an einem Punkt in der Raumzeit für Beobachter mit beliebigem Bewegungszustand berechnet werden. Um die Frage zu beantworten, was ein solcher Beobachter bei visueller Beobachtung oder mit Hilfe einer Kamera sehen kann, muss die Strahlungsbewertung durch das Auge oder durch die Kamera untersucht werden.

Optische Systeme, sei es das menschliche Auge oder eine Kamera, bilden die aus einem bestimmten Raumwinkel stammende Strahlung auf einen Sensor ab, der die einfallende Strahlungsenergie in elektrische Signale umwandelt. Jeder Sensor ist dabei nur für Licht in einem bestimmten Wellenlängenbereich empfindlich, die einfallende Strahlung wird mit einer von der Wellenlänge abhängigen Funktion gewichtet. Durch den Einsatz von Filtern kann die Empfindlichkeitskurve modifiziert werden und die Strahlungsenergie in verschiedenen Wellenlängenbereichen gemessen werden.

Abbildung 2.2 zeigt die relative Empfindlichkeit für zwei Sensoren, die Stäbchen im menschlichen Auge [54] und einen CCD Chip [40]. Das Auge ist für Strahlung im Wellenlängenbereich von 380nm bis 780nm empfindlich, die drei unterschiedlichen Sorten von Rezeptoren erlauben eine grobe Unterscheidung verschiedener spektraler Verteilun-

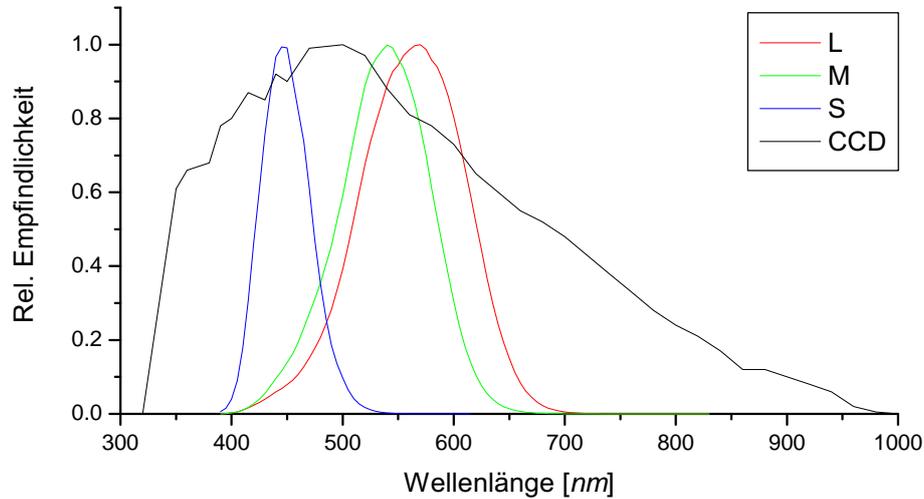


Abbildung 2.2: *Spektrale Empfindlichkeit der menschlichen Rezeptoren und eines CCD Chips (nach [54] und [40])*

gen. Erst der Vergleich der Intensitäten in verschiedenen Wellenlängenbereichen liefert den Eindruck *Farbe*. Die letztendlich wahrgenommene Farbe wird darüber hinaus nicht nur durch die spektrale Verteilung des Reizes bestimmt, sie wird auch durch die Helligkeiten und Farben im Gesichtsfeld sowie den Adaptionszustand des visuellen System beeinflusst.

Die Empfindlichkeit des CCD Chips geht bis in den infraroten Bereich, mit geeigneten Filtern lassen sich die Empfindlichkeiten der menschlichen Rezeptoren nachbilden und Farbaufnahmen erstellen.

Zu einer gegebenen spektralen Verteilung erhält man die Antwort X_i eines Sensors für einen bestimmten Spektralbereich als mit der Empfindlichkeitskurve gewichtetes Integral

$$X_i(\mathbf{n}) = \int d\lambda x_i(\lambda) L(\lambda, \mathbf{n}) \quad (2.10)$$

wobei sich die Integration hier und im Folgenden über alle Wellenlängen beziehungsweise mindestens über den Träger der Bewertungsfunktion $x_i(\lambda)$ erstrecken soll. Um die Antwort der Netzhaut oder einer Kamera zu berechnen, muss also die spektrale Strahldichte mindestens in dem Wellenlängenbereich bekannt sein, in dem die Empfindlichkeitskurve von Null verschieden ist.

Die vom Sensor gelieferten Signale X_i für die verschiedenen Spektralbereiche können für

die weitere Verarbeitung beliebig kombiniert werden. Festzuhalten ist jedoch, dass die gesamte in der spektralen Verteilung enthaltene Information durch den Sensor zunächst auf wenige Zahlenwerte reduziert wird. Die Abbildung einer spektralen Verteilung auf diese Zahlenwerte ist nicht injektiv, es wird also immer voneinander verschiedene Spektralverteilungen geben, die die gleichen Werte ergeben. In der Farbwahrnehmung werden solche Spektralverteilungen *Metamere* genannt.

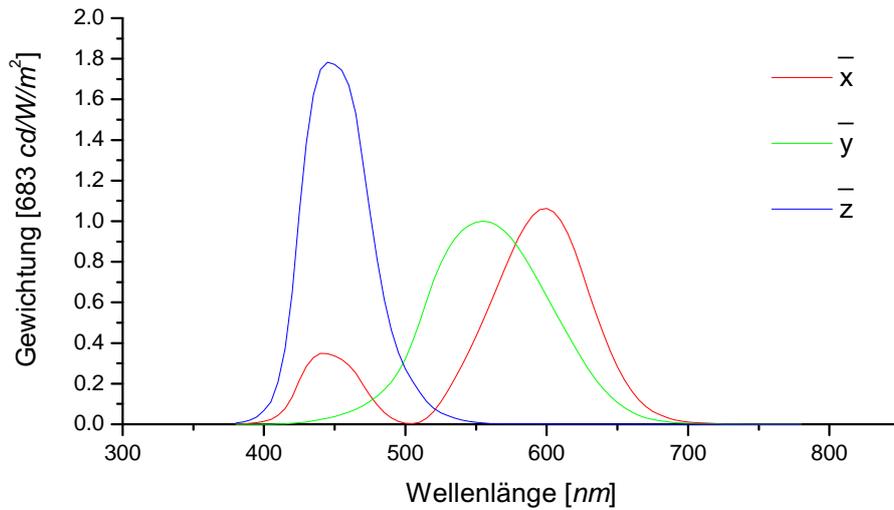


Abbildung 2.3: *Color Matching Functions*

Bewertungsfunktionen, die der gemittelten menschlichen Wahrnehmung angepasst sind, sind die *Color Matching Functions* [67]. Die Leuchtdichte $L = Y$ ist als das Integral über die Strahldichte mit der Gewichtungsfunktion $x_i(\lambda) = \bar{y}(\lambda)$ definiert. Zusammen mit den Werten X und Z , die sich durch Gewichtung mit den beiden weiteren Color Matching Functions ergeben, erhält man die Normfarbwerte für eine Spektralverteilung

$$\begin{aligned}
 X(\mathbf{n}) &= \int d\lambda \bar{x}(\lambda) I_\lambda(\lambda, \mathbf{n}) \\
 Y(\mathbf{n}) &= \int d\lambda \bar{y}(\lambda) I_\lambda(\lambda, \mathbf{n}) \\
 Z(\mathbf{n}) &= \int d\lambda \bar{z}(\lambda) I_\lambda(\lambda, \mathbf{n})
 \end{aligned} \tag{2.11}$$

mit $\bar{x}, \bar{y}, \bar{z}$ wie in Abbildung 2.3.

Stimmen die Normfarbwerte für zwei Spektralverteilungen überein, so wird ein Benutzer mit normaler Farbwahrnehmung unter ansonsten gleichen Bedingungen die gleiche

Farbe und Helligkeit wahrnehmen. Auf den Einfluss der Leuchtdichte und Farbe in anderen Richtungen des Gesichtsfeldes und der unterschiedlichen Adaptationszustände auf die Farbwahrnehmung soll an dieser Stelle nicht weiter eingegangen werden, dazu sei auf [14, 67] verwiesen.

Transformation der Leuchtdichte

Die Leuchtdichte und die Farbwerte für einen Beobachter mit beliebigem Bewegungszustand lassen sich nun einfach berechnen. Die Strahldichte wird nach 2.5 transformiert und anschließend entsprechend 2.11 gewichtet. Für die Leuchtdichte ergibt sich

$$\begin{aligned}
 L'(\mathbf{n}') &= \int d\lambda' \bar{y}(\lambda') I'(\lambda', \mathbf{n}') \\
 &= D(\mathbf{n}')^5 \int d\lambda' \bar{y}(\lambda') I(D(\mathbf{n}')\lambda', \mathbf{n}') \\
 &= D(\mathbf{n}')^5 L_I(D(\mathbf{n}'), \mathbf{n}')
 \end{aligned} \tag{2.12}$$

mit

$$L_I(D, \mathbf{n}) = \int d\lambda \bar{y}(\lambda) I(D\lambda, \mathbf{n}) \tag{2.13}$$

Die Farbwerte X und Z lassen sich analog aus der transformierten Strahldichte berechnen.

Um die wahrnehmbare Leuchtdichte oder die Antwort eines Sensors bei schneller Bewegung berechnen zu können, muss die ursprüngliche spektrale Verteilung für den gesamten um den Dopplerfaktor verschobenen Empfindlichkeitsbereich bekannt sein. Der Dopplerfaktor liegt für eine bestimmte Geschwindigkeit im Bereich zwischen $\gamma(1 - \beta)$ und $\gamma(1 + \beta)$. Sind λ_{min} und λ_{max} die Grenzen des Empfindlichkeitsbereiches, so muss die spektrale Verteilung für alle Wellenlängen

$$\gamma(1 - \beta)\lambda_{min} < \lambda < \gamma(1 + \beta)\lambda_{max} \tag{2.14}$$

durchgehend bekannt sein. Wird nur ein Ausschnitt des kompletten Raumwinkels verwendet, so kann der benötigte Wellenlängenbereich entsprechend verkleinert werden.

2.4 Modelle für Spektralverteilungen

Bei der relativistischen Umrechnung ist die spektrale Strahldichte in dem nach 2.14 benötigten Wellenlängenbereich und Raumwinkel im Allgemeinen nicht durch Messungen bekannt. Farbkameras decken nur den visuellen Bereich und eventuell das nahe Ultraviolett und Infrarot ab, spezielle Spektalkameras arbeiten meistens mit verschiedenen Bändern, die der Anwendung angepasst sind und nicht den kompletten Wellenlängenbereich abdecken. Für eine korrekte Darstellung der Leuchtdichte und Farbe ist es jedoch notwendig, die spektralen Eigenschaften der Strahlung im gesamten Bereich mit ausreichender Auflösung (einige nm) zu kennen.

Die relativistische Visualisierung ist daher auf Modelle für die Spektralverteilungen angewiesen. Häufig werden einfach darstellbare Verteilungen wie Linienspektren, energiegleiche Verteilungen oder Gaussverteilungen und Kombinationen aus diesen Verteilungen verwendet, oder physikalisch motivierte Verteilungen wie das oft verwendete Spektrum des schwarzen Körpers. Mit dem Planckschen Strahlungsgesetz lassen sich die Werte für alle Wellenlängen einfach berechnen, die entsprechende Farbe liegt zudem für alle Temperaturen im darstellbaren Bereich typischer Wiedergabegeräte. Darüber hinaus ist die transformierte Verteilung für einen Beobachter mit einer beliebigen Relativbewegung wiederum die spektrale Verteilung eines schwarzen Körpers bei einer anderen Temperatur. Die Temperaturen sind dabei einfach durch $T' = DT$ mit dem Dopplerfaktor D für die entsprechende Richtung miteinander verknüpft [41].

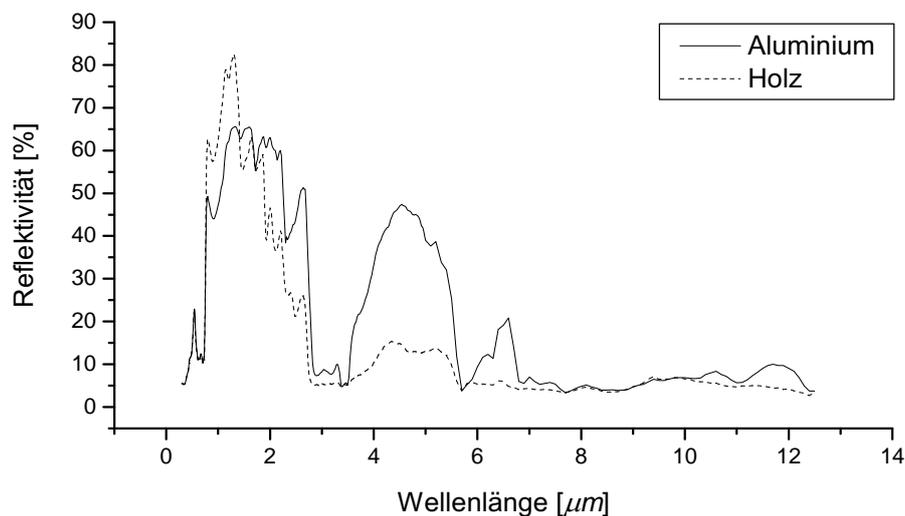


Abbildung 2.4: *Spektrale Reflektivität einer grünen Farbe auf Aluminium und Kiefernholz im Bereich von $0.3 \mu m$ bis $12 \mu m$ (nach [4])*

Bei der relativistischen Umrechnung von Bildern oder Texturen tritt das Problem auf,

zu einer Farbe in der Textur eine Spektralverteilung zu finden, die dieser Farbe entspricht. Da dies für unendlich viele Verteilungen möglich ist, können weitere Bedingungen an die Verteilung gestellt werden. Verschiedene Ansätze, ein Spektrum aus gegebenen RGB Werten zu rekonstruieren, finden sich in [20]. Eine korrekte Darstellung im relativistischen Fall ist allerdings ohne weitere Information nicht möglich. Eine Bildfarbe wird dann immer durch dieselbe Verteilung dargestellt, damit ist die Farbe schon für alle Dopplerverschiebungen festgelegt. Unterschiedliche Verteilungen insbesondere im Infraroten können mit dieser Technik nicht dargestellt werden.

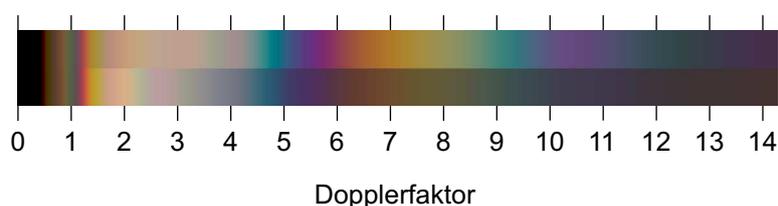


Abbildung 2.5: *Berechnete Farben¹ für die Reflektivitäten aus 2.4 in Abhängigkeit von der Dopplerverschiebung bei Beleuchtung mit einem gleichverteilten Spektrum ohne den Faktor D^5 . Grüne Farbe auf Aluminium (oben) und Kiefernholz (unten).*

Abbildung 2.4 stellt die gemessene spektrale Reflektivität einer grünen Farbe auf Holz und Aluminium dar. Unverschoben erscheinen beide Verteilungen grün. Abbildung 2.5 zeigt den Farbeindruck der beiden Materialien bei Beleuchtung mit einem gleichverteilten Spektrum für verschiedene Werte des Dopplerfaktors. Die unterschiedliche Reflektivität im infraroten Bereich verursacht deutliche Unterschiede in der Farbe und Helligkeit für die transformierten Verteilungen. Abbildung 2.5 zeigt auch, dass die Farben nicht nur wie bei der spektralen Verteilung des schwarzen Körpers oder bei einer dominanten Wellenlänge mit steigendem Dopplerfaktor ins Bläuliche verschoben werden, es können vielmehr alle Farbeindrücke auftreten.

Bei der relativistischen Visualisierung muss man also gewisse Ungenauigkeiten akzeptieren, wenn normales Bildmaterial in Form von Texturen und eine generische Verteilung für Wellenlängen außerhalb des visuellen Bereichs verwendet wird. Die Ungenauigkeit liegt dabei nicht im Verfahren der relativistischen Visualisierung, sondern in der unvollständigen Information über die spektrale Verteilung der dargestellten Körper.

Wird ein 3D-Modell speziell für die relativistische Visualisierung erstellt, so kann das Problem gelöst werden, indem für die Oberflächen nicht die Farbe, sondern verschiedene Materialien in der Textur spezifiziert werden. Jedem Material kann dann eine spektrale Verteilung zugeordnet werden. Die Farbe für die Darstellung wird erst zur Laufzeit

¹Im Druck und auf dem Monitor können nicht alle Farben dargestellt werden. In dieser Arbeit berechnete Farbwerte können also nur näherungsweise wiedergegeben werden.

nach der relativistischen Umrechnung der Verteilung ermittelt. Diese Einbindung der Spektralverteilungen in die interaktive Visualisierung ist in Abschnitt 5.1.7 beschrieben.

2.5 Visualisierung der spektralen Strahldichte

Die letzte Stufe der Visualisierung ist die physikalische Darstellung der aufbereiteten Daten für den Benutzer mit Hilfe von Bildschirmen, Projektionen oder VR-Brillen. Um den visuellen Eindruck für einen Beobachter, der sich relativ zu den betrachteten Körpern mit hoher Geschwindigkeit bewegt, exakt wiederzugeben, müsste die spektrale Strahldichteverteilung nach 2.5 reproduziert werden. Eine identische Reproduktion der spektralen Strahldichte ist im Allgemeinen jedoch weder technisch möglich, noch aus der Sicht des Benutzers wegen der teilweise extrem hohen Strahldichte wünschenswert.

In Abschnitt 2.2 wurde gezeigt, dass die Leuchtdichte für einen schnell bewegten Beobachter extrem unterschiedliche Werte für verschiedene Blickrichtungen annehmen kann. Die mit üblichen Geräten erreichbaren Leuchtdichten liegen bei einigen hundert cd/m^2 und haben einen Kontrastumfang von etwa 500:1. Es ist offensichtlich, dass die bei relativistischer Bewegung auftretenden Leuchtdichten nicht einmal annäherungsweise wiedergegeben werden können. Für die Darstellung müssen die Leuchtdichten daher so skaliert werden, dass sie im darstellbaren Bereich des Monitors oder Projektors liegen, und dass die Helligkeits- und Kontrastverhältnisse der Szene trotz dieser Einschränkungen möglichst gut wiedergegeben werden. Die entsprechende Abbildung der spektralen Verteilungen beziehungsweise Leuchtdichten auf darstellbare Werte wird als *Tone Mapping* bezeichnet.

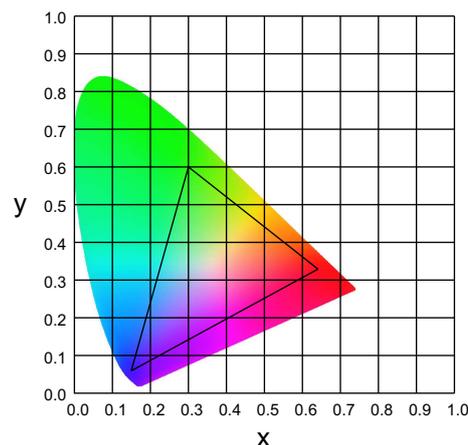


Abbildung 2.6: CIE XY Farbdiaagramm mit dem von einem Projektor darstellbaren Farbbereich.

Abbildung 2.6 zeigt den mit handelsüblichen Monitoren darstellbaren Bereich im CIE-XY Farbdiaagramm[67]. Um aus einer spektralen Verteilung die Ausgabewerte (in den

meisten Fällen werden dies RGB-Werte sein) für ein bestimmtes Gerät zu berechnen, werden zunächst die Normfarbwerte berechnet. Die rechenaufwendige Integration der gewichteten Verteilung für jeden Bildpunkt kann entfallen, wenn man sich auf eine endliche Zahl von spektralen Verteilungen oder wenige, durch einen kontinuierlichen Parameter gekennzeichnete Familien von Verteilungen I_j beschränkt. Die Strahldichte lässt sich dann in Abhängigkeit von der Richtung und Wellenlänge als

$$I(\lambda, \mathbf{n}) = \sum_j a_j(\mathbf{n}) I_j(\lambda, \mathbf{n}) \quad (2.15)$$

darstellen. Die Funktionen a_j beschreiben die Anteile der jeweiligen Verteilung in Abhängigkeit von der Richtung. Die gewichteten Werte für einen bewegten Beobachter lassen sich dann mit 2.5 und 2.10 berechnen:

$$\begin{aligned} X_i(\mathbf{n}') &= \int d\lambda' x_i(\lambda') L'(\lambda', \mathbf{n}') \\ &= D(\mathbf{n}')^5 \sum_j a_j(\mathbf{n}) \int d\lambda' x_i(\lambda') I_j(D(\mathbf{n}')\lambda', \mathbf{n}) \\ &= \sum_j a_j(\mathbf{n}) D(\mathbf{n}')^5 X_{ij}(D(\mathbf{n}')) \end{aligned} \quad (2.16)$$

mit

$$X_{ij}(D) = \int d\lambda' x_i(\lambda') I_j(D\lambda') \quad (2.17)$$

Die X_{ij} lassen sich aus den spektralen Verteilungen I_j beim Programmstart für alle auftretenden Dopplerfaktoren berechnen und in einer Textur ablegen. Der Faktor D^5 könnte prinzipiell mit in die Funktion $X_{ij}(D)$ genommen werden. Für die praktische Anwendung und Darstellung der Funktion $X_{ij}(D)$ als Textur ist die gewählte Form jedoch numerisch besser geeignet, siehe Abschnitt 5.1.7 im Kapitel Implementierung.

Während der Laufzeit werden die Normfarbwerte für jeden Bildpunkt nach 2.16 berechnet, anschließend werden mit Hilfe des Tone Mappings die endgültigen Ausgabewerte ermittelt. Die einfachste Realisierung des Tone Mappings ist eine lineare Skalierung der berechneten Leuchtdichte auf den darstellbaren Leuchtdichtebereich, allerdings gehen in diesem Fall meistens Details des Bildes verloren und es wirkt insgesamt zu dunkel. Für eine der menschlichen Wahrnehmung besser angepasste Darstellung von Bildinhalten mit hohem dynamischen Umfang sind verschiedene Verfahren entwickelt worden, [14] und [12] geben einen Überblick über die Probleme und Techniken. Abschnitt 5.1.5 stellt die Implementierung des Tone Mappings im Rahmen dieser Arbeit dar.

2.6 Überblick über die Techniken zur relativistischen Visualisierung

Die Verfahren zur relativistischen Visualisierung lassen sich im Wesentlichen in drei Gruppen gliedern: Geometriebasierte Verfahren, bildbasierte Verfahren und Raytracing. Im letzten Abschnitt dieses Kapitels werden die unterschiedlichen Techniken für die Erzeugung der Bildinformation bei schneller Bewegung kurz skizziert, für eine sehr gute weitergehende Darstellung der verschiedenen Verfahren sei auf [64] verwiesen.

2.6.1 Geometriebasierte Verfahren

Bei den geometriebasierten Verfahren wird die Geometrie der darzustellenden Körper während der Laufzeit so verändert, dass sie der relativistischen Verzerrung entspricht. Ausgangspunkt für das geometriebasierte Verfahren ist wieder die Tatsache, dass ein Beobachter an einem bestimmten Ereignis in der Raumzeit nur diejenigen Emissionsereignisse beobachten kann, die einen lichtartigen Abstand zu seiner Position in der Raumzeit haben. Das Licht, das den Beobachter von einem Punkt auf einem Körper erreicht, stammt also vom Schnittpunkt der Weltlinie des Punktes mit dem rückwärts gewandten Lichtkegel des Beobachters. Durch diesen Schnittpunkt und die aktuelle Position des Beobachters ist die Richtung, in der der Punkt gesehen wird, eindeutig festgelegt. Da eine eventuelle Beschleunigung keinen Einfluss auf den Schnittpunkt hat, kann das Verfahren auch für nicht gleichförmig bewegte Beobachter verwendet werden.

Die Berechnung des Schnittpunktes erfolgt im momentanen Ruhesystem des Beobachters. Die Position des Beobachters sei durch (ct_0, \mathbf{r}_0) gegeben, der betrachtete Punkt bewege sich mit der Geschwindigkeit \mathbf{v} relativ zum Beobachter und habe die Koordinaten \mathbf{r}' in seinem Ruhesystem. Wie üblich sei angenommen, dass die Koordinatensysteme im Ereignis $(0, 0)$ zusammenfallen und die gleiche Orientierung haben. Die Bewegung des Punktes wird dann durch

$$\mathbf{r} = \mathbf{r}' - \frac{\gamma}{\gamma + 1} \boldsymbol{\beta} \cdot \mathbf{r}' \boldsymbol{\beta} + ct \boldsymbol{\beta} \quad (2.18)$$

beschrieben, das Emissionsereignis und die Beobachtung sind durch einen lichtartigen Vektor verbunden:

$$(ct - ct_0)^2 - (\mathbf{r} - \mathbf{r}_0)^2 = 0 \quad (2.19)$$

Auflösen nach ct ergibt für das Emissionsereignis

$$ct = \gamma^2 \left(\boldsymbol{\beta} \cdot \mathbf{d} + ct_0 \pm \sqrt{(\boldsymbol{\beta} \cdot \mathbf{d} + ct_0)^2 + \frac{1}{\gamma^2} (\mathbf{d}^2 - (ct_0)^2)} \right) \quad (2.20)$$

mit

$$\mathbf{d} = \mathbf{r}' - \frac{\gamma}{\gamma + 1} \boldsymbol{\beta} \cdot \mathbf{r}' \boldsymbol{\beta} - \mathbf{r}_0 \quad (2.21)$$

Bei der Visualisierung wird nur die retardierte Lösung benötigt, der gesuchte Schnittpunkt ist

$$\mathbf{r}_r = \gamma^2 \left(\boldsymbol{\beta} \cdot \mathbf{d} + ct_0 - \sqrt{(\boldsymbol{\beta} \cdot \mathbf{d} + ct_0)^2 + \frac{1}{\gamma^2} (\mathbf{d}^2 - (ct_0)^2)} \right) \boldsymbol{\beta} + \mathbf{d} + \mathbf{r}_0 \quad (2.22)$$

Die so gewonnenen Punkte bilden die Oberfläche eines Körpers, des so genannten Photokörpers, der anschließend normal dargestellt wird, um das Aussehen für den Beobachter zu erhalten. Die nichtlineare Transformation wird dabei nicht auf alle einzelnen Punkte der Oberfläche angewendet, sondern auf die Gitterpunkte des zugrunde liegenden polygonalen Modells des Körpers, meistens ist dies ein einfaches Dreiecksgitter. Wegen der Nichtlinearität ist es für das Verfahren wichtig, dass das Dreiecksgitter fein genug unterteilt ist, andernfalls treten Artefakte durch die lineare Interpolation zwischen den einzelnen Gitterpunkten auf, siehe auch Abschnitt 5.3. Texturierte Modelle müssen nicht nur an den Kanten fein unterteilt sein, um sichtbare Knicke in den Mustern und ein ‘Schwimmen’ der Texturen bei Richtungsänderungen zu vermeiden. Abbildung 2.7 zeigt den Photokörper für einen schnell bewegten Zug aus der Sicht des Beobachters und aus einer weiteren Perspektive zur Verdeutlichung der Geometrie.

Searchlight- und Dopplereffekt können in das Verfahren eingebunden werden, indem die entsprechenden Werte für die Gitterpunkte berechnet werden und linear interpoliert in die Darstellung der Textur einfließen [61].

Die Berechnung der retardierten Position kann auf aktueller Grafikhardware vollständig in der programmierbaren Vertexeinheit erfolgen, Modelle mit 25000 Polygonen lassen sich so mit 450 Bildern pro Sekunde rendern.

Das geometriebasierte Verfahren eignet sich besonders gut für Umgebungen, bei denen Körper, die sich mit unterschiedlichen Geschwindigkeiten bewegen, darzustellen sind. Auch beschleunigte Körper lassen sich mit diesem Verfahren visualisieren, wenn die Weltlinien der einzelnen Punkte aus einem Modell für die Verformung bei relativistischen Beschleunigungen bekannt sind. Die retardierten Positionen sind dann im allgemeinen nicht mehr durch einen geschlossenen Ausdruck wie in 2.22 darstellbar.

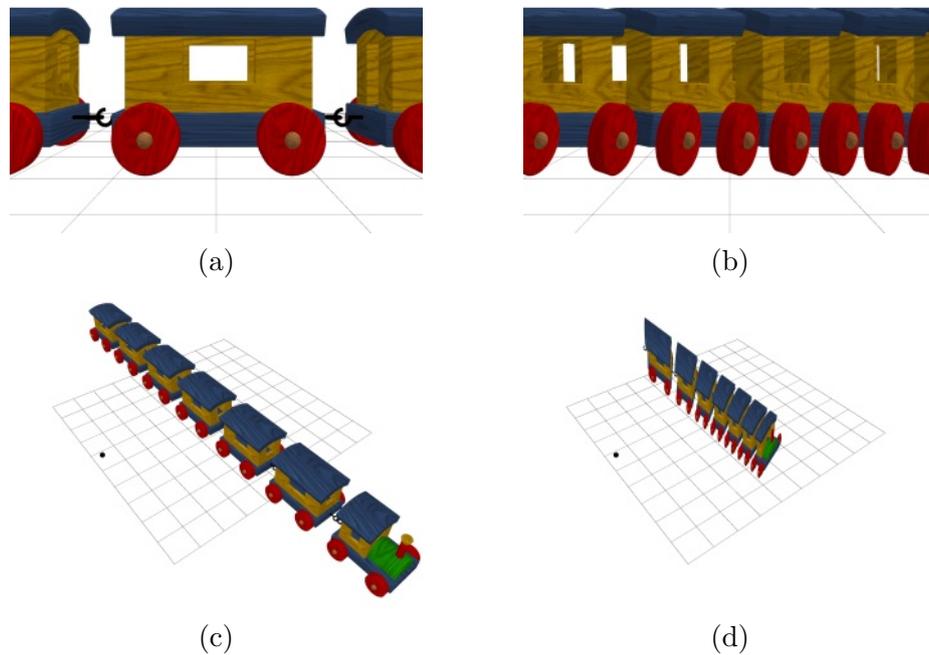


Abbildung 2.7: Geometriebasierte Visualisierung: Photokörper eines ruhenden (a) und eines mit 90% der Lichtgeschwindigkeit (b) bewegten Zuges aus der Sicht des Beobachters und aus einer anderen Perspektive (c),(d). Die Position des Beobachters ist schwarz markiert.

2.6.2 Bildbasierte Verfahren

Ausgangspunkt der bildbasierten Verfahren ist die in Abschnitt 2.1 gemachte Feststellung, dass die optische Information in einem Punkt der Raumzeit vollständig bekannt ist, wenn sie für einen – meistens nur gedachten – Beobachter mit einer beliebigen Geschwindigkeit in diesem Punkt bekannt ist. Für Beobachter mit einer anderen Geschwindigkeit kann die Leuchtdichte dann mittels 2.5 bestimmt werden.

Die Rundumsicht wird praktischerweise im Ruhesystem der Körper erstellt. Dort kann die Rundumsicht aus einzelnen Aufnahmen der Umgebung zusammengesetzt werden, oder aber für die interaktive Simulation erst während der Laufzeit für den jeweiligen Punkt in der Raumzeit erzeugt werden. Im Ruhesystem der Körper können die einzelnen Teilbilder klassisch gerendert werden, es können alle bekannten Rendermethoden und Beleuchtungsmodelle verwendet werden.

Ist die Rundumsicht erstellt, so wird diese in einem zweiten Schritt für jeden Bildpunkt oder durch Verzerrung eines Gitters in das System des Beobachters umgerechnet, um die relativistisch korrekte Sicht für den Beobachter zu erhalten. Je nach Anwendung

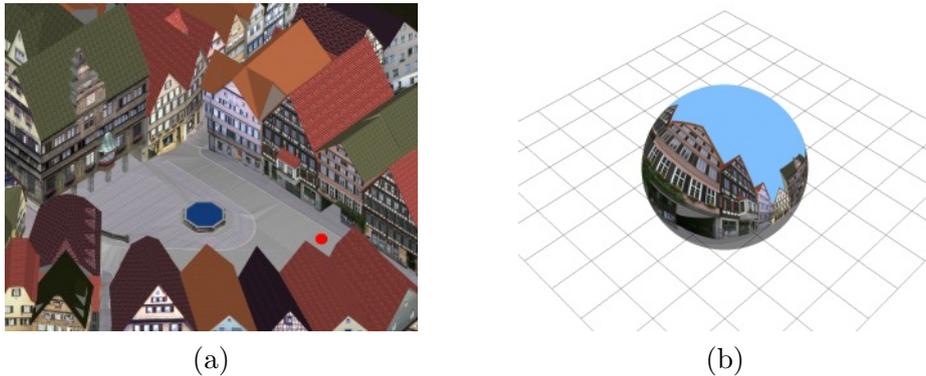


Abbildung 2.8: *Bildbasierte Visualisierung: Rundumsicht für einen Beobachter an der in (a) rot markierten Position. Der Beobachter ruht im Ruhesystem der Häuser.*

kann nur die geometrische Verzerrung erfolgen, oder es wird komplett mit Doppler- und Searchlighteffekt umgerechnet, falls die spektrale Information in der Rundumsicht vorliegt oder ein Modell für die Rekonstruktion der spektralen Verteilungen verwendet wird. Sollen realistische spektrale Verteilungen verwendet werden, so muss der Renderalgorithmus der ersten Stufe die entsprechende Materialinformation und Beleuchtung generieren und in der Rundumsicht ablegen. Andernfalls kann die spektrale Verteilung bei der Umrechnung aus den Farbwerten nur grob modelliert werden, vergleiche Abschnitt 2.4.

Das bildbasierte Verfahren eignet sich am besten zur Visualisierung eines Fluges durch eine statische Umgebung. Mit anderer Geschwindigkeit bewegte Körper können zwar visualisiert werden, indem sie entweder mit Hilfe des geometriebasierten Verfahrens in die Rundumsicht gerendert werden, oder indem für den Körper eine zweite Rundumsicht in seinem Ruhesystem erzeugt wird, die anschließend mit der vorhandenen Rundumsicht gemischt wird. Im zweiten Fall sind die gegenseitigen Verdeckungen der Körper zu beachten, die Rundumsicht muss daher zusätzlich eine Entfernungsinformation enthalten. In beiden Fällen geht allerdings die Eleganz des bildbasierten Verfahrens verloren.

Durch die Aberration können die Raumwinkel stark vergrößert oder verkleinert werden, so dass beim zweiten Schritt des Verfahrens Abtastfehler (oversampling/undersampling) die Bildqualität verschlechtern können. Abhilfe kann hier eine adaptive Texturauflösung schaffen. Für die Implementierung vergleiche Abschnitt 5.3.

Die gesamte relativistische Umrechnung kann im zweiten Schritt zusammengefasst werden. Dies erlaubt es, die relativistische Umrechnung als Aufsatz für bestehende Simulationen und Renderengines zu verwenden. Die Simulation muss lediglich erlauben, an der aktuellen Position die Rundumsicht zu erstellen, es müssen also die Blickrichtung und der Blickwinkel für die einzelnen Teilbilder vorgegeben werden können. Zu beachten ist dabei, dass der darzustellende Bildinhalt nicht von der Blickrichtung abhängig

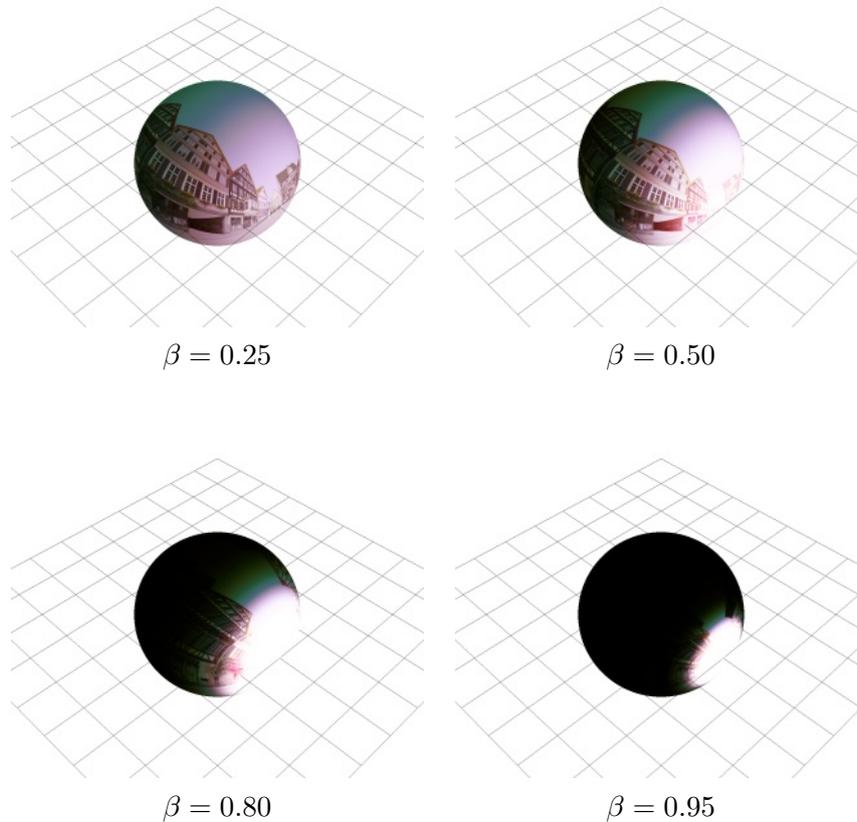


Abbildung 2.9: *Bildbasierte Visualisierung: Die Rundumsicht aus Abbildung 2.8 für einen mit verschiedenen Geschwindigkeiten bewegten Beobachter. Die Bewegung erfolgt entlang der Gitterachse nach rechts unten, zur Wiedergabe der Farben vergleiche die Bemerkungen im Text.*

sein darf, ansonsten können Bildfehler in der Rundumsicht an den Grenzen zwischen den einzelnen Teilbildern auftreten. Ein Beispiel für eine solche Abhängigkeit von der Blickrichtung sind Billboards, die orthogonal zur Blickrichtung dargestellt werden.

2.6.3 Raytracing

Relativistisches Raytracing existiert in einer Vielzahl von Varianten, vom klassischen dreidimensionalen Raytracing, bei dem nur die Richtung der für einen Bildpunkt generierten Primärstrahlen entsprechend der Abberation verändert wird, bis hin zum Raytracing in vierdimensionalen gekrümmten Raumzeiten[64]. Bei allen Verfahren werden zur Bestimmung der Farbe eines Bildpunktes ein oder mehrere Strahlen erzeugt und entlang einer Nullgeodäte rückwärts in der Raumzeit verfolgt. An den Schnittpunkten mit den Objekten der modellierten Umgebung wird die Wechselwirkung mit dem entsprechenden Material in einem lokalen Ruhesystem berechnet, dazu werden eventuell weitere Strahlen rekursiv ausgesandt.

Das Raytracing war das erste für die Visualisierung der relativistischen Effekte vorgeschlagene Verfahren [22]. Es ist die Methode der Wahl, wenn höchste Bildqualität gefordert wird, und wenn viele voneinander unabhängig bewegte und beschleunigte Körper dargestellt werden sollen. Variable und bewegte Lichtquellen, Schatten, sowie Reflexion und Brechung an bewegten Körpern können nur mit dem Raytracing korrekt dargestellt werden. Für die interaktive Bilderzeugung ist das Verfahren jedoch noch zu langsam.

2.7 Bewegungssimulation

Wegen der im Verhältnis zur Lichtgeschwindigkeit kleinen Geschwindigkeiten im Alltag oder auch von Raumsonden sind die visuellen Effekte der speziellen Relativitätstheorie nur durch Visualisierung und Simulation zugänglich. Um die Effekte in einer Simulation interaktiv zugänglich zu machen, muss die Lichtgeschwindigkeit künstlich verringert werden, oder es müssen hypothetische Antriebe verwendet werden, um in kurzer Zeit eine mit der Lichtgeschwindigkeit vergleichbare Geschwindigkeit zu erreichen.

Die Simulation des Beobachters oder eines Raumschiffes wird meistens auf die Simulation eines Punktes beschränkt, für ausgedehnte Körper ist bei den notwendigen Beschleunigungen kein physikalisch vernünftiges Modell bekannt. Das gleiche gilt bei einer Herabsetzung der Lichtgeschwindigkeit, die auch die Materialeigenschaften beeinflussen würde. Weiter wird angenommen, dass die Beschleunigung keinen Einfluss auf die Kamera oder die Wahrnehmung hat (das Photonenfeld ist natürlich durch den Punkt in der Raumzeit vollständig festgelegt und unabhängig von der Geschwindigkeit und Beschleunigung des Beobachters [61]. Reale abbildende Systeme werden aber Effekte bei Beschleunigung zeigen).

Um dem Benutzer ein Gefühl für die Zeitdilatation und Längenkontraktion zu vermitteln, kann die Integration der Bewegung mit der Eigenzeit des simulierten Punktes erfolgen, bei ausreichend großer Geschwindigkeit kann der Benutzer dann trotz der endlichen Lichtgeschwindigkeit in einem festen Zeitintervall beliebig große Strecken in der simulierten Welt zurücklegen. Sollen sich allerdings mehrere Benutzer gleichzeitig durch eine virtuelle Welt bewegen können, so kann die simulierte Eigenzeit nicht mehr proportional zur echten Zeit des Benutzers sein, wenn keine Inkonsistenzen zwischen den Benutzern auftreten sollen. Fliegt beispielsweise ein Beobachter auf einer Kreisbahn, so ist für ihn beim erneuten Zusammentreffen mit einem Beobachter, der einfach in einem Inertialsystem ruht, weniger Eigenzeit vergangen. Das Zwillingsparadoxon lässt grüßen.

Die Integration der Bewegung erfolgt zweckmäßigerweise in einem Inertialsystem mit einem Runge-Kutta- oder ähnlichem Verfahren. Der Benutzer steuert die Bewegung meistens durch direkte Vorgabe einer Richtung und Geschwindigkeit. Effekte wie die Thomas-Rotation eines mitgeführten Systems bei beschleunigten Bewegungen werden meist nicht dargestellt.

Die durch die Aberration verursachten Verzerrungen machen die Navigation in der virtuellen Welt gewöhnungsbedürftig. Bei beschleunigter Bewegung entfernt sich ein Objekt in Bewegungsrichtung zunächst scheinbar, da es wegen der zunehmenden Aberration einen immer kleineren Winkelbereich im Gesichtsfeld ausfüllt. Die alternative Interpretation eines kleiner werdenden Objekts in gleichbleibender Entfernung wird nicht wahrgenommen, siehe auch Abschnitt 3.4.2 für den Eindruck bei stereoskopischer Visualisierung. Die Erfahrungen der Benutzer der Simulationen zeigen, dass die Bestimmung der eigenen Position im Raum erschwert ist. Kurven mit einem bestimmten Winkel lassen sich ohne Bezugspunkt nur ungenau fliegen, Entfernungen sind schwer einzuschätzen. Trotzdem gelingt den meisten Benutzern nach kurzer Gewöhnungsphase die Navigation in der virtuellen Welt recht gut. Die eigene Erfahrung zeigt, dass zur Positionsbestimmung der enge Vorbeiflug an Landmarken und die Verwendung von Peillinien über zwei Objekte hinweg an Bedeutung gewinnen.

Kapitel 3

Dreidimensionale Visualisierung

Das Auge und die meisten abbildenden Systeme liefern ein zweidimensionales Bild der dreidimensionalen Umwelt. In jedem Bildpunkt ist die Helligkeit und Farbe für eine Richtung abgebildet. Die Entfernung der Objekte geht bei der Abbildung des dreidimensionalen Raumes auf die zweidimensionale Bildebene verloren.¹

Das menschliche Gehirn ist in der Lage, aus den zweidimensionalen Bildern der beiden Augen die Entfernung zu rekonstruieren, so dass die Umgebung als dreidimensional wahrgenommen wird. Einen wichtigen Beitrag zur räumlichen Wahrnehmung liefert die Stereopsis, die Extraktion von Tiefenhinweisen aus den durch die Parallaxe bedingten Unterschieden zwischen den Bildern der beiden Augen. In diesem Kapitel werden Verfahren vorgestellt, die die stereoskopische Wahrnehmung bei der Visualisierung relativistischer Bewegungen ermöglichen. In Abschnitt 3.4 wird der Einfluss der relativistischen Effekte auf die wahrgenommene Entfernung schnell bewegter Körper untersucht.

3.1 Entfernungswahrnehmung

An der Rekonstruktion der räumlichen Tiefe aus einer Helligkeitsverteilung in der Bildebene sind bei der menschlichen Wahrnehmung eine Vielzahl von Mechanismen beteiligt[29]. In Abhängigkeit von der Anzahl der verwendeten Bilder sowie der zeitlichen und räumlichen Unterschiede in der Position der Augen können die Mechanismen grob in drei Gruppen eingeteilt werden:

¹Streng genommen gilt dies nur für die idealisierte Abbildung. Bei realen Systemen treten auch entfernungsabhängige Effekte wie z.B. Defokussierung auf, die sich zur Rekonstruktion der Tiefe verwenden lassen.

Tiefenhinweise aus einem Bild

- **Verdeckung**
Weiter entfernte Objekte können teilweise von näher liegenden Objekten verdeckt werden. Die Verdeckungen liefern damit einen Hinweis auf die Tiefenstaffelung der Objekte untereinander.
- **Größe von Körpern mit bekannter Ausdehnung**
Ist die Größe eines Körpers bekannt, so kann aus seiner Bildgröße und den Abbildungsparametern die Entfernung bestimmt werden.
- **Beleuchtung und Schatten**
Die scheinbare Helligkeit hängt von der Richtung zur Lichtquelle ab. Umgekehrt kann aus der Helligkeitsverteilung auf die Orientierung geschlossen werden. Aus dem Schattenwurf kann die Lage der Körper untereinander bestimmt werden.
- **Texturgradient**
Abhängig von der Orientierung einer Oberfläche wird ein Muster perspektivisch verzerrt erscheinen. Mit bestimmten Annahmen über das Muster kann die Neigung und Form der Fläche bestimmt werden.
- **Perspektive**
Parallele Geraden und bekannte Winkel können zur Rekonstruktion der Lage im Raum und damit zur Entfernungsbestimmung verwendet werden.

Tiefenhinweise aus einem Bildpaar

- **Disparität / Parallaxe**
Ein Punkt im Raum erscheint, von zwei verschiedenen Positionen aus gesehen, in unterschiedlichen Richtungen. Aus diesen Richtungen und der Geometrie der bilderzeugenden Systeme kann die Tiefe bestimmt werden.

Tiefenhinweise aus einer Bildfolge / Bewegung

- **Bewegungsparallaxe**
Ändert sich die Position der Kamera oder des Auges zwischen den Aufnahmen, so erscheint ein Punkt im Raum wieder in verschiedenen Richtungen. Aus den Richtungen kann, analog zum obigen Fall der Parallaxe, die Entfernung bestimmt werden.
- **Kinetischer Tiefeneffekt**
Bewegt sich der Beobachter, so ist die Geschwindigkeit des Bildes eines Punktes in der Bildebene auch von dessen Entfernung abhängig. Dieser Effekt kann wiederum zur Entfernungsbestimmung genutzt werden.

Bei der menschlichen Wahrnehmung werden alle diese Tiefenhinweise genutzt, um eine Tiefenwahrnehmung zu erzeugen. Auf die Mechanismen der Zusammenfassung der einzelnen Tiefenhinweise zu einer einheitlichen Tiefenwahrnehmung, die *Integration von Tiefenhinweisen*, kann an dieser Stelle nicht eingegangen werden. Eine Übersicht findet sich in [31].

Welche der Mechanismen können bei der dreidimensionalen Visualisierung unterstützt werden? Die Erzeugung der einzelnen Bilder (und damit auch der Bildfolgen) kann auf aktueller Grafikhardware bei entsprechender Komplexität des dreidimensionalen Modells und des Beleuchtungsmodells so erfolgen, dass fast Foto- beziehungsweise Videoqualität erreicht wird. Bei entsprechender Präsentation der Bilder für das Auge werden so alle Mechanismen, die die Tiefenhinweise aus einem Bild oder einer Bildfolge ermitteln, unterstützt.

Einen wesentlich stärkeren räumlichen Eindruck erhält der Benutzer, wenn die Visualisierung zusätzlich die stereoskopische Tiefenwahrnehmung unterstützt. In den folgenden Abschnitten wird daher besonders auf die stereoskopische Visualisierung und den Einfluss der relativistischen Effekte auf die stereoskopische Wahrnehmung eingegangen.

Der Searchlight- und Dopplereffekt sowie insbesondere die Aberration beeinflussen die Tiefenwahrnehmung über die meisten der genannten Tiefenhinweise. So wird beispielsweise bei der Entfernungsbestimmung über den Winkel, unter dem ein Körper mit bekannter Größe erscheint, eine größere Entfernung wahrgenommen, wenn die Aberration den Körper bei schneller Bewegung unter einem kleineren Winkel erscheinen lässt. Inwiefern diese Verfälschung der Tiefenhinweise konsistent ist, ob also die aus den einzelnen Tiefenhinweisen ermittelte Entfernung für alle Mechanismen um den gleichen Faktor verändert wird, soll an dieser Stelle nicht weiter untersucht werden (vgl. aber die Bemerkungen am Ende von Abschnitt 3.4.2).

3.2 Stereogeometrie

Das stereoskopische Tiefensehen verwendet die durch die Parallaxe hervorgerufenen Unterschiede zwischen den Bildern der beiden Augen. Die Unterschiede zwischen den beiden Bildern bei binokularer Beobachtung werden durch die unterschiedlichen Positionen und Orientierungen der beiden Augen oder Kameras, sowie durch eventuell unterschiedliche Abbildungsparameter hervorgerufen. Die Entfernung eines Objekts ergibt sich dann als Schnittpunkt von zwei aus korrespondierenden Bildmerkmalen ermittelten Sichtlinien.

Im Folgenden sei ein abbildendes System durch die Position des Nodalpunktes R_l , eine Orientierungsmatrix M_l und eine Abbildungsfunktion φ vollständig beschrieben. Die Abbildungsfunktion $\varphi : R^2 \rightarrow S^2$ ordnet dabei jedem Bildpunkt diejenige Richtung im Bezugssystem des Auges oder der Kamera zu, aus der das Licht auf den Bildpunkt

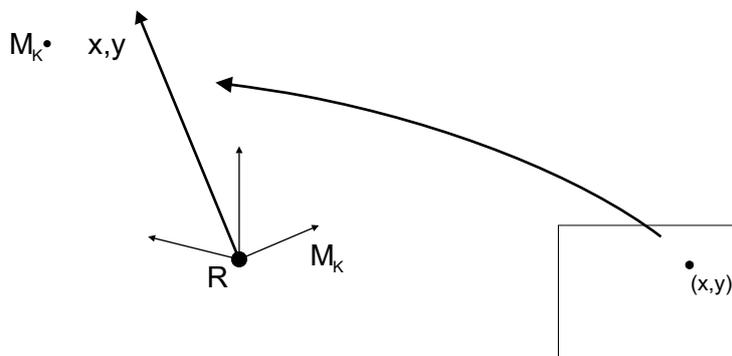


Abbildung 3.1: Kameramodell: Die Abbildung φ ordnet jedem Bildpunkt die Sichtlinie bezüglich der Kameraorientierung M zu.

abgebildet wird. Ein Punkt P im Raum wird damit auf den Punkt

$$b_l = \varphi_l^{-1}(M_l^{-1}(P - R_l)) \quad (3.1)$$

in der Bildebene abgebildet. Offensichtlich werden aber auch alle Punkte, die auf der Geraden durch P und R liegen, auf denselben Bildpunkt abgebildet. Für ein zweites abbildendes System wird der Punkt P entsprechend auf einen weiteren Bildpunkt

$$\varphi_r^{-1}(M_r^{-1}(P - R_r)) \quad (3.2)$$

abgebildet. Sind die Positionen, Orientierungen und Abbildungsfunktionen der beiden Systeme vollständig bekannt, so lässt sich die Lage von P im Raum als Schnittpunkt der beiden Halbgeraden

$$\begin{aligned} R_l + \lambda_l M_l \varphi(b_l) & \quad \lambda_l, \lambda_r > 0 \\ R_r + \lambda_r M_r \varphi(b_r) & \end{aligned} \quad (3.3)$$

bestimmen. In der Praxis werden sich die Geraden nicht exakt bestimmen lassen und sich nicht schneiden, da die Auflösung der abbildenden Systeme endlich ist und die Parameter der Abbildung nicht beliebig genau bekannt sind, man wählt dann den Mittelpunkt der kürzesten Verbindung zwischen den Geraden als beste Schätzung für die Lage von P im Raum.

Die größte Schwierigkeit bei der Bestimmung der Tiefe aus den beiden Bildern stellt das so genannte Korrespondenzproblem dar, also das Auffinden von Bildpunkten im

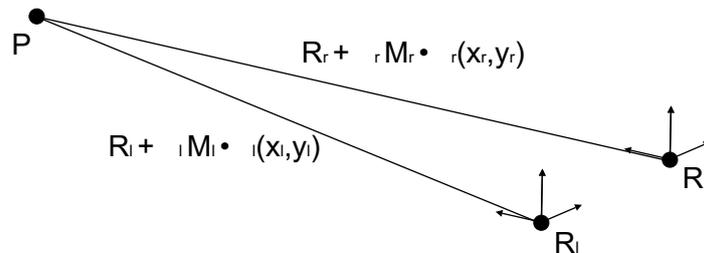


Abbildung 3.2: Geometrie der stereoskopischen Entfernungsbestimmung: Die Lage eines Punktes P im Raum ergibt sich als Schnittpunkt der Sichtlinien.

rechten und linken Bild, die das Bild ein und desselben Punktes der Umwelt sind. Erst wenn diese korrespondierenden Punkte gefunden sind, kann die Entfernung als Schnittpunkt der Sichtlinien bestimmt werden. Dabei werden meistens nicht nur einzelne korrespondierende Punkte bestimmt, es werden vor allem weitere Bildmerkmale wie Kanten und Ecken, deren Lage in den beiden Bildern mit größerer Genauigkeit bestimmt werden kann, zur Bestimmung der Entfernung der Körper herangezogen. Auf die Details der Extraktion von Bildmerkmalen aus den beiden Bildern und die entsprechende Zuordnung soll an dieser Stelle nicht weiter eingegangen werden, ebenso wenig wie auf die Bestimmung der Kameraorientierungen und Abbildungsfunktionen aus Aufnahmen von Testbildern und Testobjekten. Eine Einführung findet sich in [69]. Die Leistung des menschlichen Gehirns bei der Lösung des Korrespondenzproblems ist im allgemeinen Fall immer noch unübertroffen.

Der Zusammenhang zwischen der Winkelauflösung $\Delta\alpha$ des abbildenden Stereosystems und der Entfernungsauflösung Δd in einer Entfernung d ist durch

$$\frac{\Delta d}{d} = \frac{d}{a} \Delta\alpha \quad (3.4)$$

gegeben, a ist der halbe Abstand der Aufnahmepositionen. Für die menschliche Tiefenwahrnehmung liegt $\Delta\alpha$ im Bereich von einigen Bogensekunden [53], mit dem Astrometriesatelliten *Hipparcos* lassen sich Auflösungen im Bereich von Millibogensekunden erreichen.

3.3 Technik der stereoskopischen Visualisierung

Während die meisten anderen Tiefenhinweise bei der Visualisierung durch die gerenderten Bildinhalte eines einzelnen Bildes oder einer Bildfolge übermittelt werden, so sind für die stereoskopische Tiefenwahrnehmung spezielle Techniken erforderlich. Die Tiefenhinweise werden aus den Unterschieden in den Bildern der beiden Augen extrahiert, die durch die voneinander verschiedenen Augenpositionen bedingt sind. Bei der Visualisierung müssen also die jeweils perspektivisch richtigen Bilder für jedes Auge einzeln erstellt und so dargestellt werden, dass ein Auge jeweils nur die gewünschte Bildinformation empfängt.

3.3.1 Verfahren zur Bildkanaltrennung

Grundsätzlich beruhen alle Verfahren zur stereoskopischen Visualisierung auf einem gemeinsamen Prinzip. Die beiden Teilbilder für das linke und das rechte Auge müssen den Augen durch eine so genannte Bildkanaltrennung zugeführt werden, jedes Auge darf nur das entsprechende Teilbild sehen. Die Teilbilder können dabei räumlich, zeitlich oder durch Benutzung unterschiedlicher Spektralbereiche oder Polarisationsrichtungen getrennt werden. Für die interaktive Visualisierung sind die folgenden Verfahren am weitesten verbreitet:

- **Head Mounted Displays / VR Brillen**

Die beiden Teilbilder werden jeweils auf einem kleinen LCD dargestellt und den Augen räumlich getrennt zugeführt. Bei ausreichend großem Blickfeld kann eine vollständige virtuelle Umgebung erzeugt werden. Kann die Kopfposition genau genug bestimmt werden (Headtracking), so kann sich der Benutzer interaktiv in der virtuellen Welt bewegen. Wichtig ist dabei eine genaue Bestimmung der Kopfposition und eine Darstellung mit kurzer Latenz, ansonsten können durch die unterschiedlichen visuellen und vestibulären Eindrücke Schwindelgefühle entstehen.

- **Anaglyph-Brillen**

Anaglyph-Brillen sind die preiswerteste Lösung für die stereoskopische Visualisierung. Die Trennung erfolgt durch die Darstellung der beiden Teilbilder mit unterschiedlichen Spektralanteilen. Die Brille mit einem roten und einem grünen Filter (auch rot-cyan) trennt die beiden Teilbilder. Ein Nachteil ist die Beeinflussung der Farbwahrnehmung.

- **Shutterbrillen**

Bei diesem Verfahren wird eine zeitliche Trennung verwendet, die Teilbilder werden abwechselnd dargestellt. Die Brille besteht aus zwei LCD-Panels, die synchron zur Darstellung der Teilbilder wechselseitig schließen. Der Wechsel der Bilder kann

bei einem Monitor maximal mit der Bildwiederholffrequenz erfolgen, die effektive Bildrate für ein Auge wird somit halbiert. Bei der Projektion kann dasselbe Verfahren verwendet werden, hier besteht zusätzlich die Möglichkeit, zwei Projektoren zu verwenden, die gleichzeitig die Bilder für das rechte und linke Auge darstellen. Durch zusätzliche Shutter vor den Projektoren und eine Shutterbrille können die Bilder auf die beiden Augen aufgeteilt werden, wobei auch höhere Shutterfrequenzen als die Bildwiederholffrequenz verwendet werden können.

- **Polarisationsbrillen**

Die beiden Teilbilder werden durch Polarisationsfilter, die um 90 Grad zueinander gedreht sind, projiziert. Die Brille enthält zwei Polarisationsfilter mit entsprechender Orientierung, so dass jedes Auge nur das richtige Teilbild sieht. Die Projektionsfläche muss bei diesem Verfahren die Polarisationsrichtung des Lichts erhalten, hierzu werden metallisch beschichtete Leinwände verwendet.

- **3D-TFT-Bildschirme**

Die räumliche Trennung der Teilbilder erfolgt durch eine in das Display eingebaute winkelabhängige Spezialmaske, so dass für bestimmte Positionen vor dem Display nur das rechte oder linke Teilbild zu sehen sind [50, 68].

Den besten Eindruck einer dreidimensionalen Welt erhält man bei der großflächigen Projektion in Kombination mit Polarisations- oder Shutterbrillen.

3.3.2 Erzeugung der Teilbilder

Die beiden unterschiedlichen Teilbilder für die beiden Augen müssen bestimmten Anforderungen genügen, damit sie zu einer dreidimensionalen Wahrnehmung verschmolzen werden können. Jedes Teilbild muss die virtuelle Welt aus der Sicht eines virtuellen Auges darstellen. Anschließend müssen die Teilbilder so dargestellt werden, dass die entsprechenden Richtungen korrekt wiedergegeben werden.

Einige Grafikkarten bieten in Kombination mit einem geeigneten Treiber die Möglichkeit, die beiden Teilbilder automatisch zu erzeugen. Für eine Anwendung, die mit einer Standardschnittstelle wie OpenGL auf die Grafikkhardware zugreift, ist die stereoskopische Darstellung dann völlig transparent. Die dreidimensionalen Objekte werden im Wechsel für das linke und das rechte Auge dargestellt, wobei intern zwei Bildspeicher jeweils mit Front- und Backbuffer für die beiden Augen eingesetzt werden. Zusätzlich erzeugt die Grafikkhardware die Signale zur Synchronisation der Shutterbrille. Als Parameter ist meist nur ein 'Augenabstand' einstellbar, der sowohl den Augenabstand in der virtuellen Welt, als auch den Versatz für die Darstellung auf dem Monitor bestimmt.

Die stereoskopische Visualisierung für den relativistischen Fall benutzt grundsätzlich das gleiche Verfahren wie die klassische stereoskopische Visualisierung. Die Teilbilder für die beiden Augen werden entsprechend der aktuellen Augenposition erstellt, diesmal

allerdings mit den relativistischen Techniken aus Abschnitt 2.6. Mit der Bildkanaltrennung werden die Teilbilder den Augen getrennt zugeführt, so dass eine stereoskopische Wahrnehmung relativistisch bewegter Körper simuliert werden kann. Die Verfahren zur Erzeugung der relativistisch verzerrten Teilbilder und die verschiedenen Arten der Bildkanaltrennung sind dabei beliebig kombinierbar.

Wird das bildbasierte Verfahren zur Erstellung der Teilbilder verwendet, so ist zu beachten, dass die Rundumsichten für die beiden Augen für Ereignisse berechnet werden müssen, die im Ruhesystem der Augen gleichzeitig sind und den korrekten Augenabstand haben. Vom Ruhesystem der dargestellten Körper aus betrachtet werden die Rundumsichten zeitlich nacheinander und in einem größeren räumlichen Abstand erstellt.

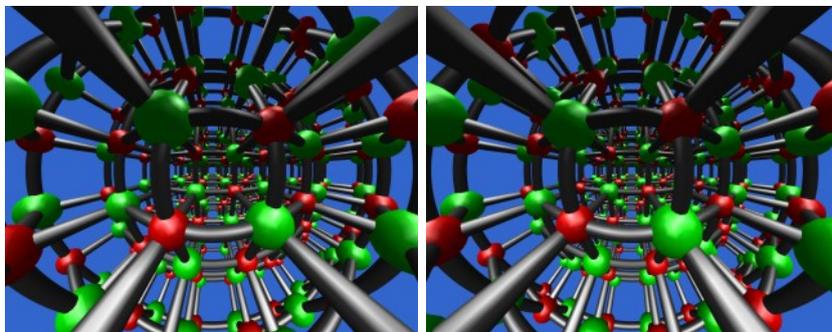


Abbildung 3.3: *Räumliches Gitter bei der Bewegung mit 80% der Lichtgeschwindigkeit für das linke und rechte Auge.*

Die automatische Erzeugung der Teilbilder kann im relativistischen Fall nicht mehr verwendet werden. Beim bildbasierten Verfahren enthält die Rundumsicht nur noch die Information für ein Auge, bei der verzerrten Darstellung im zweiten Schritt wird keine oder eine völlig andere Tiefeninformation an die Grafikhardware geschickt. Beim geometriebasierten Verfahren muss die unterschiedliche Retardierung für die beiden Augen berücksichtigt werden. Die automatische Erzeugung der Teilbilder kann somit auch bei diesem Verfahren nicht verwendet werden.

Die wahrgenommene Entfernung kann - wie im Folgenden gezeigt wird - im relativistischen Fall deutlich verschieden von der wirklichen Entfernung sein, sofern die beiden Bilder eines Körpers überhaupt eine Interpretation als ein Objekt in einer bestimmten Entfernung zulassen. Darüber hinaus können sich die Verdeckungsverhältnisse für Körper mit unterschiedlichen Geschwindigkeiten für den Beobachter umkehren. Ein Körper, der vor einer Wand entlangläuft, kann also hinter der Wand erscheinen, natürlich ohne dass er verdeckt wird. Umgekehrt bleibt ein hinter einer Wand laufender Körper unsichtbar, obwohl die scheinbare Entfernung (sichtbar beispielsweise in Löchern in der Wand) geringer als die der Wand ist (vgl. Abbildung 3.14). Daher ist auch die Technik, die beobachtbare Entfernung zu berechnen und die Körper anschließend in dieser Entfernung dreidimensional darzustellen, wenigen Spezialfällen vorbehalten.

Mit Grafikhardware, die zwar die stereoskopische Darstellung auf die oben genannte Art unterstützt, aber keine *stereo buffer* für unabhängige Teilbilder bietet, kann der Augenabstand auf Null gesetzt werden und die Erzeugung der Teilbilder in die Applikation verlagert werden. Die Synchronisation mit der Shutterbrille muss dann ebenfalls in der Applikation erfolgen.

Die erreichbare Tiefenauflösung für einen normalen Monitor mit 0.26 mm Pixelabstand und einer Beobachterentfernung von 50 cm liegt nach 3.4 bei wenigen Millimetern und kann durch Disparitäten im Subpixelbereich noch verringert werden.

3.4 Entfernung und Aussehen schnell bewegter Körper

Im Folgenden soll untersucht werden, wie schnell bewegte Körper bei binokularer Beobachtung gesehen werden, genauer: In welcher Entfernung werden Objekte wahrgenommen, wenn die beiden Teilbilder des Objekts entsprechend der Relativgeschwindigkeit von Beobachter und Objekt für die beiden Augen relativistisch verzerrt werden?

Hierbei wird die Verwendung eines klassischen Stereoalgorithmus vorausgesetzt, bei dem Effekte wie die Lichtlaufzeit, Kontraktion und Aberration nicht korrigiert werden. Die Entfernung wird also als Schnittpunkt der Sichtlinien bestimmt, die dem von einem Punkt ausgestrahlten Licht entsprechen, das für den Beobachter gleichzeitig an beiden Augen ankommt. Dies entspricht der stereoskopischen Entfernungswahrnehmung des Menschen. Bei der Beobachtung mit optischen Instrumenten können die relativistischen Effekte natürlich korrigiert werden, um die korrekte Entfernung eines Körpers zu erhalten. In Abschnitt 3.5 wird allerdings gezeigt, dass bei schneller Bewegung die Entfernung im allgemeinen Fall nicht mehr aus einem Stereobildpaar allein ohne weitere Information gewonnen werden kann.

Modell für die stereoskopische Entfernungsbestimmung

Zunächst soll die stereoskopisch bestimmbare Entfernung für eine punktförmige Lichtquelle P , die sich relativ zum Beobachter gleichförmig bewegt, berechnet werden. Das Erscheinungsbild eines bewegten Körpers ergibt sich dann aus den Entfernungen und scheinbaren Positionen für alle Punkte auf der Oberfläche des Körpers.

Licht von der Lichtquelle P , das für den Beobachter gleichzeitig an den Augen ankommt, komme aus den Richtungen \mathbf{n}_0 und \mathbf{n}_1 . Die scheinbare Position der Lichtquelle für den Beobachter ergibt sich dann als Schnittpunkt der Geraden durch die Punkte \mathbf{a}_0 und \mathbf{a}_1 mit den Richtungen n_0 und n_1 .

Die Berechnungen der Richtungen und des Schnittpunktes können im Ruhesystem des Beobachters oder im Ruhesystem des Körpers erfolgen. Im zweiten Fall trifft das Licht

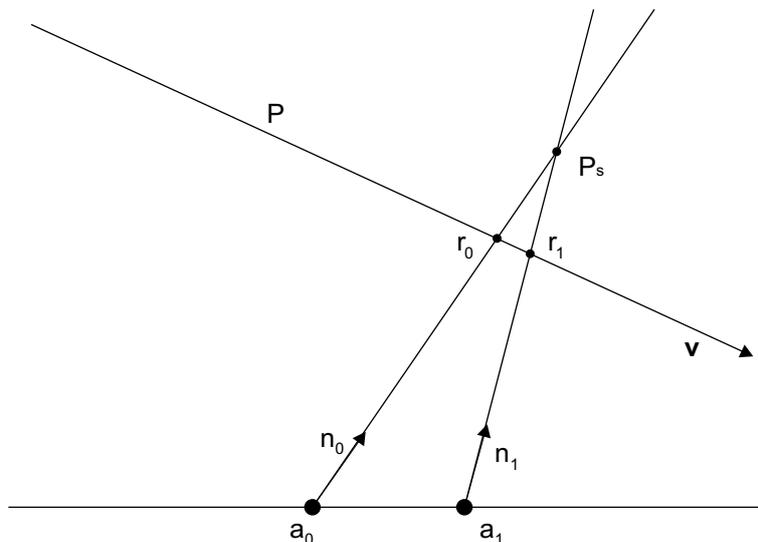


Abbildung 3.4: Stereoskopische Entfernungsbestimmung: Das von P in r_0 und r_1 emittierte Licht erreicht a_0 und a_1 gleichzeitig. P wird an der Position P_s wahrgenommen.

zu verschiedenen Zeitpunkten bei den beiden Augen ein, die Richtungen müssen entsprechend der Aberration umgerechnet werden. Auch wenn die Berechnung im Ruhesystem des Körpers dem in dieser Arbeit favorisierten Verfahren der bildbasierten Visualisierung eher entspricht, so ist die Berechnung im Ruhesystem des Beobachters anschaulicher und wird deshalb im Folgenden verwendet. Die Richtung, aus der das Licht von P auf ein Auge trifft, ergibt sich in diesem Fall als Richtungsvektor von der Position des Auges zu der Position, an der sich die Lichtquelle befand, als das Licht ausgesandt wurde.

Die Augen beziehungsweise die Beobachtungsinstrumente werden im Folgenden wieder als vernachlässigbar klein gegenüber allen auftretenden Größen angenähert, insbesondere wird ihre Ausdehnung als klein gegenüber ihrem Abstand vorausgesetzt. Alle Effekte, die durch die Lichtlaufzeit im Instrument auftreten, können damit vernachlässigt werden. Die genaue Abbildungsfunktion des Auges oder der Kamera spielt hier keine Rolle, da der Mechanismus zur Entfernungsbestimmung nur mit der eigentlichen Sichtlinie arbeitet, die Abbildungsfunktion fällt damit heraus. Im Ruhesystem des Beobachters befinden sich die Augen im Abstand von $2a$ an den Positionen $(a, 0, 0)$ und $(-a, 0, 0)$.

Abbildung 3.4 zeigt die Position der Augen und die beiden Positionen der Lichtquelle P , an denen das Licht emittiert wurde, das die Augen zum Zeitpunkt t_0 erreicht. Mit Gleichung 2.22 für die Retardierung ergibt sich für diese Positionen

$$\mathbf{r}_i = \gamma^2 \left(\boldsymbol{\beta} \cdot \mathbf{d}_i + ct_0 - \sqrt{(\boldsymbol{\beta} \cdot \mathbf{d}_i + ct_0)^2 + \frac{1}{\gamma^2} (\mathbf{d}_i^2 - (ct_0)^2)} \right) \boldsymbol{\beta} + \mathbf{d}_i + \mathbf{a}_i \quad (3.5)$$

wobei

$$\mathbf{d}_i = \mathbf{r}' + \frac{1}{\beta^2} \left(\frac{1}{\gamma} - 1 \right) \boldsymbol{\beta} \mathbf{r}' \boldsymbol{\beta} - \mathbf{a}_i \quad (3.6)$$

für $i = 0, 1$ gesetzt wurde.

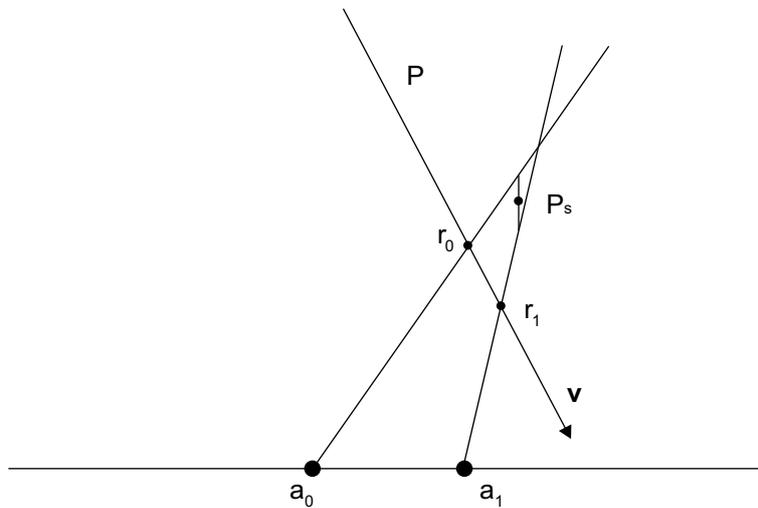


Abbildung 3.5: *Stereoskopische Entfernungsbestimmung: Im allgemeinen Fall schneiden sich die Sichtlinien nicht, bei kleinem Abstand der Sichtlinien wird die Lichtquelle in der Mitte der kürzesten Verbindungsstrecke wahrgenommen.*

Die Geraden durch die Punkte \mathbf{r}_0 und \mathbf{a}_0 sowie durch \mathbf{r}_1 und \mathbf{a}_1 sind die Sichtlinien, unter denen die Lichtquelle zum Zeitpunkt t_0 gesehen wird. Schneiden sich die Geraden in einem Punkt P_s , so erscheint die Lichtquelle dem Beobachter an dieser Position. Im allgemeinen Fall werden die Geraden keinen gemeinsamen Schnittpunkt haben, ist der Abstand der Geraden zueinander klein, so wird die Lichtquelle in der Mitte der kürzesten Verbindung beider Geraden gesehen, bei größerem Abstand werden statt einer dreidimensionalen Wahrnehmung zwei Bilder der Lichtquelle gesehen.

Damit die Geraden einen Schnittpunkt besitzen, muss die Lichtquelle im System des Beobachters ruhen (dies ist der klassische Fall ohne Bewegung beziehungsweise mit unendlicher schneller Lichtausbreitung), oder die Bewegung muss auf eine Ebene, die die Position der beiden Augen enthält, beschränkt sein. In diesen Fällen ist eine stereoskopische Wahrnehmung möglich. Die wichtigsten Fälle werden in den folgenden Abschnitten diskutiert.

3.4.1 Bewegung parallel zu den Augen

Ein besonderer Fall liegt vor, wenn die Relativbewegung parallel zur Verbindungslinie der beiden Augen ist. Die Bewegung ist dann für alle Punkte, die nicht auf der Geraden durch die Augen liegen, auf die durch die Augen und den jeweiligen Punkt definierte Ebene beschränkt. Die beiden Geraden durch die Augen und die retardierten Positionen haben in diesem Fall immer einen Schnittpunkt.

Wegen der Rotationssymmetrie um die Gerade durch die beiden Augen ist es ausreichend, die Entfernungsbestimmungen für Punkte einer Ebene zu betrachten. Das Koordinatensystem S des Beobachters sei so gewählt, dass die Augen auf der x-Achse bei $(-a, 0)$ und $(a, 0)$ liegen. Der betrachtete Punkt habe die Koordinaten (x', y') in Bezug auf ein zweites System S' , dessen Achsen parallel zu den Achsen von S sind, und das sich mit der Geschwindigkeit v entlang der x-Achse bewegt, so dass die Koordinatenursprünge zum Zeitpunkt $t = t' = 0$ zusammenfallen.

Für die Bewegung des Punktes gilt dann

$$\begin{aligned} x &= x'/\gamma + vt \\ y &= y' \end{aligned} \quad (3.7)$$

Die retardierten Positionen sind nach 3.5 durch

$$\begin{aligned} x_{0/1} &= \gamma x' + \beta \gamma^2 \left(\mp \beta a + ct_0 - \sqrt{\left(\frac{x'}{\gamma} \mp a + \beta ct_0 \right)^2 + \frac{y'^2}{\gamma^2}} \right) \\ y_{0/1} &= y' \end{aligned} \quad (3.8)$$

gegeben. Der Schnittpunkt der Geraden durch die Augen und die retardierten Positionen ist dann

$$\begin{aligned} x_s &= \frac{(x_0 + x_1)/2}{1 + (x_0 - x_1)/2a} = \frac{x'/\gamma + \beta ct_0 - \frac{\beta}{2}(q_0 + q_1)}{1 + \frac{\beta}{2a}(q_0 - q_1)} \\ y_s &= \frac{y'}{1 + (x_0 - x_1)/2a} = \frac{y'/\gamma^2}{1 + \frac{\beta}{2a}(q_0 - q_1)} \end{aligned} \quad (3.9)$$

wobei

$$q_{0/1} = \sqrt{\left(\frac{x'}{\gamma} \mp a + \beta ct_0\right)^2 + \frac{y'^2}{\gamma^2}} \quad (3.10)$$

gesetzt wurde. Mit diesen Gleichungen kann für jeden Punkt (x', y') die scheinbare Position (x_s, y_s) bei stereoskopischer Beobachtung bestimmt werden. Die Abbildung

$$T_{\parallel} : (x', y') \rightarrow (x_s, y_s) \quad (3.11)$$

ordnet jedem Punkt seine scheinbare Position zu. Die Form eines ausgedehnten Körpers bei binokularer Beobachtung erhält man, indem man alle Punkte auf der Oberfläche des Körpers entsprechend der Abbildung T_{\parallel} transformiert. Abbildung 3.6 zeigt das Bild eines quadratischen Gitters in der Halbebene $x' > 0$ unter der Abbildung T_{\parallel} für verschiedene Geschwindigkeiten.

Die Entfernung y_s ist für eine bestimmte Geschwindigkeit auf den Bereich

$$y'(1 - \beta) < y_s < y'(1 + \beta) \quad (3.12)$$

beschränkt. Für Punkte mit $x_s < 0$ ist die Entfernung geringer als die wirkliche Entfernung, für Punkte mit $x_s > 0$ ist $y_s > y'$. Punkte, für die die retardierten Positionen zusammenfallen, erscheinen in der wirklichen Entfernung. Die retardierten Positionen haben den gleichen Abstand von beiden Augen, liegen also in der Ebene $x = 0$.

Ausgedehnte Körper

Für einen ausgedehnten Körper werden alle Punkte mittels T_{\parallel} transformiert. Die graue Fläche in Abbildung 3.6 ist das Bild eines Quadrats um den Punkt $(0, y' = 10a)$. Der Beobachtungszeitpunkt ist mit $ct_0 = \sqrt{a^2 + y'^2}$ jeweils so gewählt, dass das Bild des Mittelpunktes in der Ebene zwischen den beiden Augen liegt, der Mittelpunkt erscheint zu diesem Zeitpunkt also immer in der wirklichen Entfernung $y_s = y'$. Mit zunehmender Geschwindigkeit wird das Quadrat immer stärker deformiert.

Die Deformation kann bestimmt werden, indem die Abbildung T_{\parallel} in der Umgebung des Mittelpunktes untersucht wird. Dazu wird die Funktion in der Umgebung von $(0, y')$ durch eine lineare Abbildung angenähert, die Jacobimatrix von T_{\parallel} ist

$$DT_{\parallel}(0, y') = \frac{\sqrt{y'^2 + a^2}}{y'^2 + (a/\gamma)^2} \begin{pmatrix} \frac{1}{\gamma} \sqrt{y'^2 + a^2} & -\beta y' \\ \frac{\beta}{\gamma} y' & \frac{1}{\gamma^2} \sqrt{y'^2 + a^2} \end{pmatrix} \quad (3.13)$$

Für $a \ll y'$ ergibt sich die Matrix

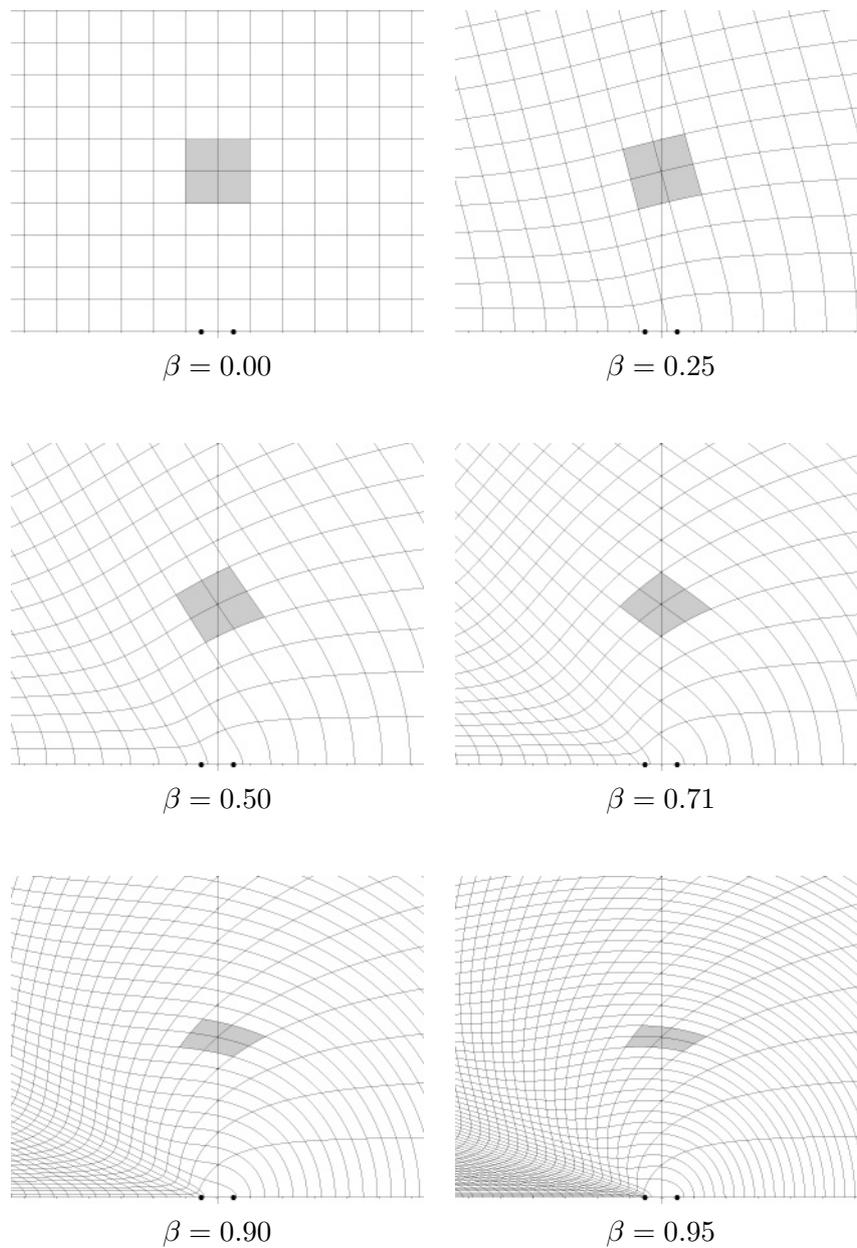


Abbildung 3.6: *Bewegung parallel zur Verbindungslinie der Augen: Die Abbildung T_{\parallel} der Ebene in Abhängigkeit von β . Die Position der Augen ist wieder durch schwarze Punkte markiert.*

$$\begin{pmatrix} 1/\gamma & -\beta \\ \beta/\gamma & 1/\gamma^2 \end{pmatrix} \quad (3.14)$$

Diese Matrix beschreibt die Transformation für Punkte in der Umgebung von M . Mit zunehmender Geschwindigkeit erscheint ein bewegter Körper bei stereoskopischer Be-

obachtung zunächst wie starr rotiert. Die Drehung nähert sich 45 Grad bei $\beta \approx 0.7$, ab dann erscheint der Körper nicht weiter gedreht, sondern wird mit zunehmender Geschwindigkeit in Richtung der y' -Achse gestaucht. Für β fast gleich eins erscheint der Körper schließlich als flache Scheibe bei $y = y'$. Dies ist unabhängig von der Entfernung, sofern nur $a \ll y'$ ist, und die stereoskopische Auflösung groß genug ist, um den Körper noch dreidimensional wahrnehmen zu können.

Die Matrix kann als Produkt aus einer reinen Drehung und einer Stauchung um den Faktor $1/\gamma$ in y -Richtung(!) geschrieben werden:

$$\begin{pmatrix} 1/\gamma & -\beta \\ \beta/\gamma & 1/\gamma^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1/\gamma \end{pmatrix} \begin{pmatrix} 1/\gamma & -\beta \\ \beta & 1/\gamma \end{pmatrix} \quad (3.15)$$

Die Drehung um den Winkel $\sin^{-1} \beta$ entspricht genau der Drehung, die üblicherweise bei den Ausführungen zur Unsichtbarkeit der Lorentzkontraktion angeführt wird. Bei der stereoskopischen Beobachtung ist die Drehung allerdings als echte Drehung im dreidimensionalen Raum zu sehen, zusammen mit der Stauchung in Beobachtungsrichtung. Die Drehung, die zur Interpretation der relativistischen Verzerrung eines Körpers bei monokularer Beobachtung verwendet wird, entspricht damit der stereoskopischen Beobachtung besser als die Kombination von Lorentzkontraktion und Scherung entlang der Bewegungsrichtung.

Da sich alle Punkte bei der Bewegung parallel zu den Augen auf den Linien $y = y' = \text{const}$ bewegen, beschreiben die Bilder dieser Linien unter der Abbildung T_{\parallel} auch die scheinbare Bewegung des Körpers für verschiedene Beobachtungszeitpunkte. Seine Bewegung folgt den Linien $y' = \text{const}$ in Abbildung 3.6. Ein vorbeifliegender Körper erscheint zunächst in y -Richtung näher als es seiner Flugbahn entspricht. Da er für den Beobachter unter einem Blickwinkel erscheint, der seiner Ausdehnung in der wirklichen Tiefe y' entspricht, ist der Körper scheinbar verkleinert. Wenn sich der Körper für den Beobachter direkt voraus befindet, erscheint er in der wirklichen Entfernung und in seiner korrekten Größe, natürlich entsprechend gedreht und gestaucht. Wie aus Abbildung 3.6 zu sehen ist, ist die momentane Bewegungsrichtung nicht parallel zu den Augen, sondern verläuft schräg nach hinten. Mit zunehmender scheinbarer Entfernung wird der Körper bei gleicher Winkelausdehnung als immer größer werdend wahrgenommen.

Nichtrelativistischer Fall

Die vorangegangenen Überlegungen haben gezeigt, dass die Entfernung, Größe und Form eines schnell bewegten Körpers bei stereoskopischer Beobachtung verfälscht wahrgenommen werden. Die endliche Lichtgeschwindigkeit und die dadurch bedingte Laufzeitdifferenz verfälschen die Tiefenwahrnehmung. Es ist also zu erwarten, dass ähnliche Effekte auch im nichtrelativistischen Fall auftreten, wenn die Geschwindigkeiten in der

Größe der Signalgeschwindigkeit auftreten.

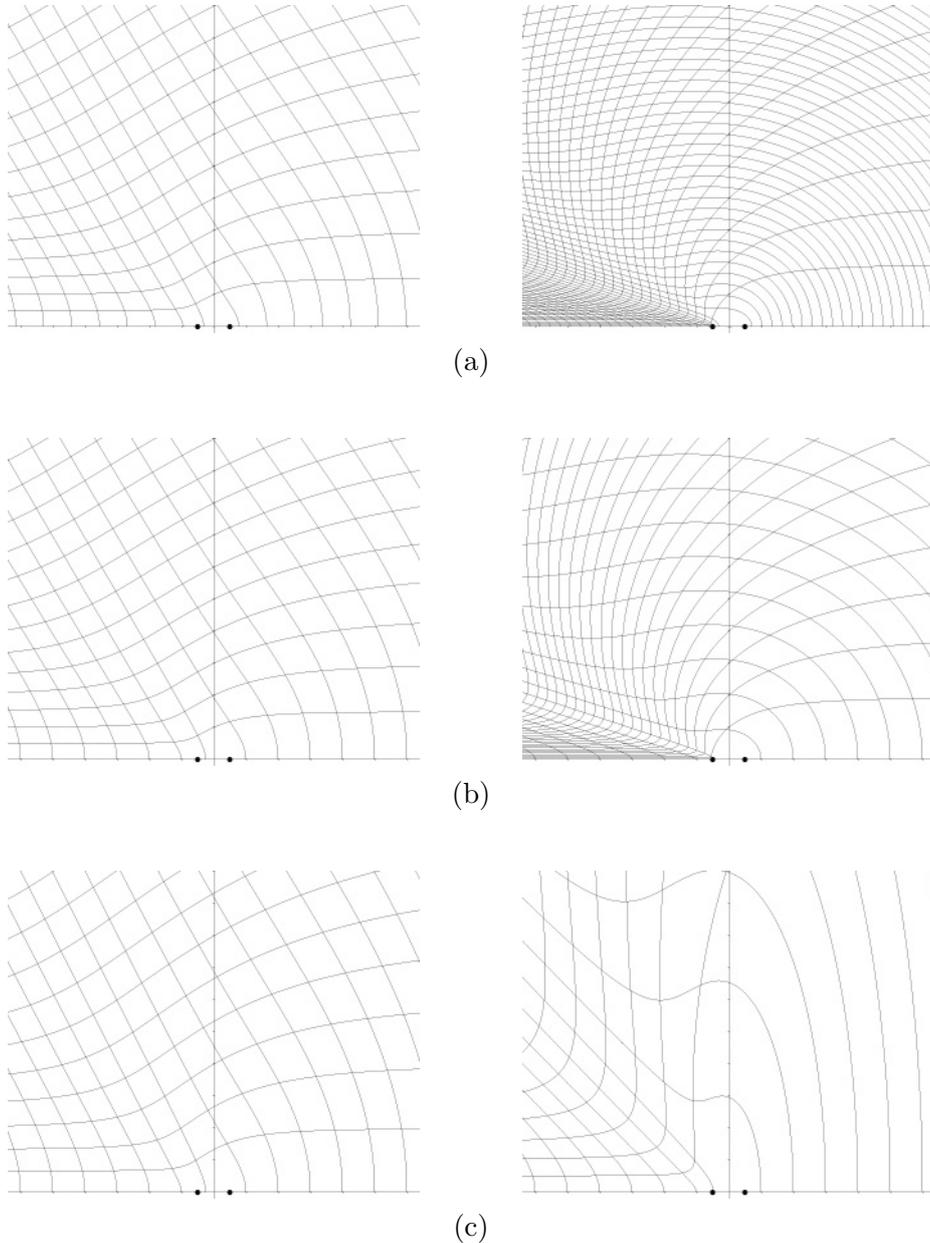


Abbildung 3.7: Vergleich mit nichtrelativistischer Bewegung: Die Abbildung T_{\parallel} der Ebene für die relativistische Bewegung (a), für die Bewegung des Körpers (b) und für einen bewegten Beobachter (c) jeweils für 50% und 95% der Signalgeschwindigkeit.

Anders als im relativistischen Fall ist beim klassischen Fall immer ein (durch ein Medium) ausgezeichnetes Bezugssystem vorhanden, in dem die Signalausbreitung mit fester Signalgeschwindigkeit erfolgt. Hier ist die visuelle Erscheinung eines Körpers nicht mehr allein von der Relativgeschwindigkeit von Körper und Beobachter abhängig, die

jeweilige Geschwindigkeit in Bezug auf das ausgezeichnete System beeinflusst das Erscheinungsbild zusätzlich. Für den Vergleich mit dem relativistischen Ergebnis werden die beiden Fälle untersucht, bei denen entweder der Körper oder der Beobachter im ausgezeichneten System ruht. Abbildung 3.7 zeigt die auftretenden Verzerrungen für die beiden Fälle im Vergleich zur relativistischen Verzerrung.

Bewegt sich der Körper gegenüber dem ausgezeichneten System, so wird sein visuelles Erscheinungsbild durch die Lichtlaufzeiteffekte bestimmt. Auch hier tritt eine Drehung auf, die Kontraktion in der scheinbaren Bewegungsrichtung fehlt. Die Matrix für die scheinbare Deformation eines kleinen Körpers ist durch

$$\begin{pmatrix} 1/\gamma & -\beta \\ \beta/\gamma & 1/\gamma^2 \end{pmatrix} \quad (3.16)$$

gegeben.

Bewegt sich der Beobachter gegenüber dem ausgezeichneten System, so kann die wahrgenommene Position eines Punktes als Schnittpunkt der entsprechend aberrierten Sichtlinien bestimmt werden. Abbildung 3.7(c) stellt wieder die Verzerrung des Gitters dar, neben der Drehung tritt nun eine starke Vergrößerung des Abstands in y -Richtung auf.

Der Vergleich für den bewegten Körper (b) mit dem relativistischen Fall zeigt, dass die Lorentzkontraktion bei stereoskopischer Beobachtung schnell bewegter Körper durchaus beobachtbar ist und als Kontraktion entlang der gedrehten scheinbaren Bewegungsrichtung auftritt. Wie bei anderen relativistischen Effekten ist man auch hier auf die Simulation beschränkt, bei den im Alltag und der Raumfahrt auftretenden Geschwindigkeiten sind die Effekte bei der visuellen Beobachtung vernachlässigbar. Für Bewegungen im Planetensystem ist $\beta \approx 10^{-4}$, ein weit entfernter Himmelskörper erscheint, die entsprechende Auflösung und Genauigkeit der stereoskopischen Beobachtung vorausgesetzt, also um einen Winkel von lediglich 20 Bogensekunden gedreht.²

3.4.2 Bewegung senkrecht zur Verbindungslinie der Augen

Bei der Bewegung durch reale und virtuelle Welten ist die Blickrichtung in Richtung der Bewegung für die Navigation und Kollisionsvermeidung von großer Bedeutung. In diesem Abschnitt soll daher das Aussehen von Körpern untersucht werden, wenn die Relativgeschwindigkeit senkrecht zur Verbindungslinie der Augen ist.

In diesem Fall werden die retardierten Positionen eines Punktes im allgemeinen nicht mehr in einer Ebene mit den beiden Augen liegen. Die Sichtlinien haben eine vertikale Disparität und schneiden sich nicht. Abhängig von der Größe der vertikalen Disparität werden ein oder zwei Bilder des Punktes wahrgenommen. Eine Ausnahme bilden

²Dieser Drehwinkel entspricht natürlich genau der üblichen astronomischen Aberration.

diejenigen Punkte, die in der Mittelebene der beiden Augen liegen (diese Punkte haben immer den gleichen Abstand zu beiden Augen, die retardierten Positionen fallen zusammen), sowie alle Punkte in der durch die Augen und den Geschwindigkeitsvektor definierten Ebene. Die retardierten Positionen dieser Punkte liegen ebenfalls in der Ebene, allerdings schneiden sich die Sichtlinien nicht für alle Punkte, oder sie schneiden sich auf der entgegengesetzten Seite der Augen, so dass keine Fusion der Bilder zu einem räumlichen Eindruck möglich ist, vgl. Abbildung 3.8.

Zunächst soll daher die durch die Augen und den Geschwindigkeitsvektor definierte Ebene betrachtet werden, für die die scheinbaren Positionen bei stereoskopischer Beobachtung analog zum Fall der Bewegung parallel zu den Augen berechnet werden können. Die Bewegung eines Punktes verläuft diesmal entlang der Linien $x = \text{const}$ mit der Geschwindigkeit v . Die retardierten Positionen sind wieder nach 3.5 durch

$$\begin{aligned} x_{0/1} &= x' \\ y_{0/1} &= \gamma y' - \beta \gamma^2 \left(ct_0 + \sqrt{\left(\beta \frac{y'}{\gamma} - ct_0 \right)^2 + \frac{1}{\gamma^2} \left((x' \mp a)^2 + \frac{y'^2}{\gamma^2} \right)} \right) \end{aligned} \quad (3.17)$$

gegeben, der Schnittpunkt (x_s, y_s) der beiden Geraden durch die Augen und die retardierten Positionen ist

$$\begin{aligned} x_s &= \frac{(y_0 + y_1)x' + (y_0 - y_1)a}{(y_0 - y_1)x' + (y_0 + y_1)a} a \\ y_s &= \frac{2y_0y_1}{(y_0 - y_1)x' + (y_0 + y_1)a} a \end{aligned} \quad (3.18)$$

Die entsprechenden Positionen für den Fall, dass sich der Beobachter und der Körper voneinander entfernen, erhält man einfach durch Einsetzen eines negativen Wertes für v . Abbildung 3.8 zeigt wieder die scheinbaren Positionen für Punkte eines Gitters, das sich auf den Beobachter zu bewegt, beziehungsweise sich von ihm entfernt. Körper, die sich auf den Beobachter zu bewegen, erscheinen auch bei stereoskopischer Beobachtung entlang der Bewegungsrichtung gestreckt. Entfernen sich Beobachter und Körper voneinander, so erscheint der Körper in Bewegungsrichtung kontrahiert. Für Punkte in der Umgebung des Punktes $(0, y')$ kann wieder die Matrix der linearisierten Abbildung berechnet werden, um die Deformation eines kleinen Körpers zu beschreiben:

$$\begin{pmatrix} 1 & 0 \\ 0 & \gamma(1 + \beta) \end{pmatrix} \quad (3.19)$$

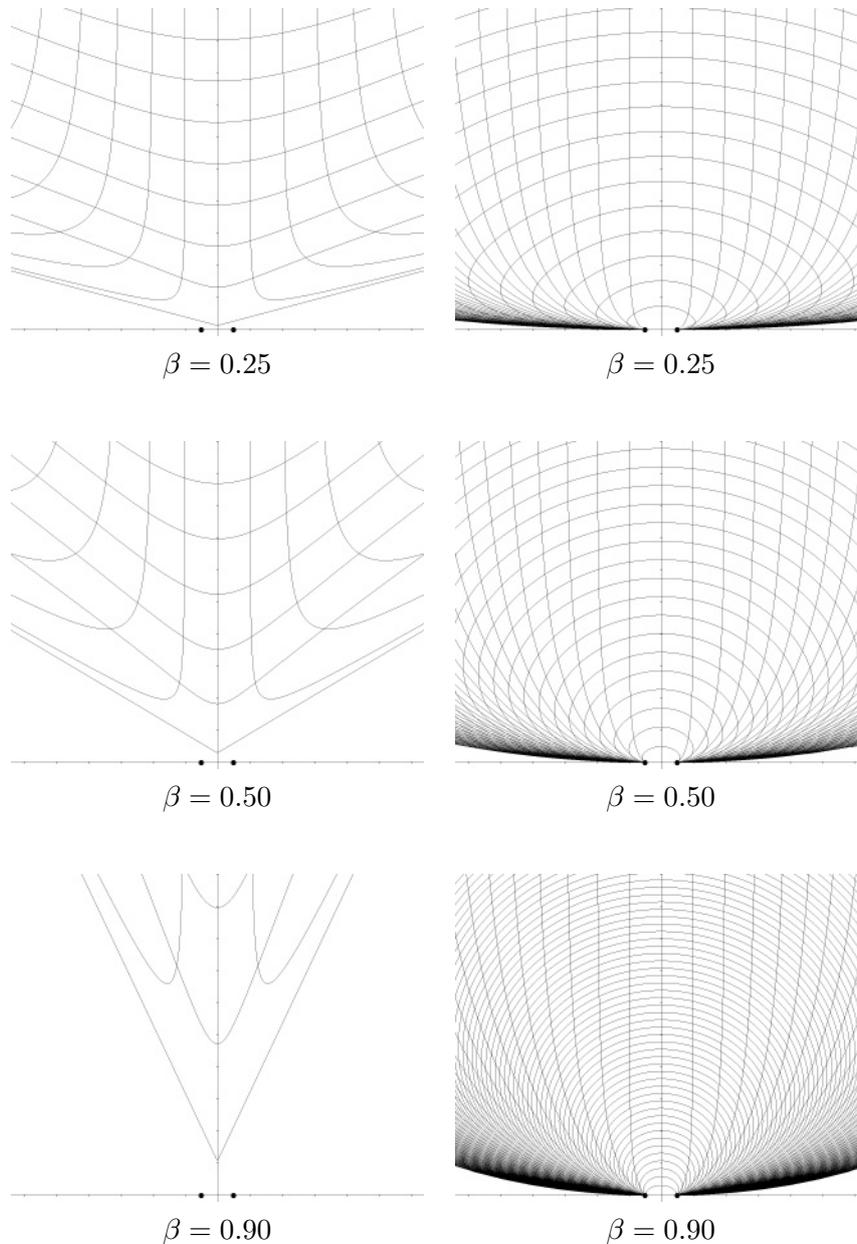


Abbildung 3.8: *Bewegung orthogonal zur Verbindungslinie der Augen: Abbildung T_{\perp} der Ebene für unterschiedliche β . In den Bereichen, in denen kein Gitter dargestellt ist, sind die Sichtlinien divergent*

Die Sichtlinien von Punkten, die sich außerhalb der beiden erwähnten Ebenen befinden, haben im allgemeinen keinen Schnittpunkt mehr. Je nach Abstand zu den Ebenen und der Relativgeschwindigkeit können die beiden Bilder jedoch noch zum Eindruck eines Punktes im dreidimensionalen Raum verschmolzen werden. Die räumliche Wahrnehmung eines Körpers entspricht dabei genau der Verzerrung, die bei der Beobachtung mit nur einem Auge gesehen wird, insbesondere sind die Tiefenhinweise aus der schein-

baren Größe eines Körpers und aus der stereoskopischen Beobachtung beim Blick in Bewegungsrichtung konsistent. Ein Körper in Bewegungsrichtung erscheint also weiter entfernt und unter einem kleineren Winkel, so dass sich die wahrgenommene absolute Größe des Körpers im Gegensatz zum Fall der Bewegung parallel zu den Augen nicht verändert. Abbildung 3.15 zeigt ein räumliches Gitter bei unterschiedlichen Geschwindigkeiten als Stereobild. Bei Annäherung an die Lichtgeschwindigkeit wird der Bereich, in dem die Fusion zu einem räumlichen Eindruck gelingt, zunehmend kleiner.

3.4.3 Beobachtung mit mitbewegtem Kopf

In den vorangegangenen Abschnitten ist deutlich geworden, dass die stereoskopische Entfernungswahrnehmung durch die trotz gleicher Bewegungsrichtung unterschiedliche Aberration für die beiden Augen beeinflusst wird. Gegenläufige Bewegungen, wie sie zum Beispiel bei der Drehung des Kopfes auftreten können, führen schon bei Bewegungen mit einem Bruchteil der Lichtgeschwindigkeit zu deutlichen Verzerrungen. In diesem Abschnitt soll der Einfluss einer Rotation der beiden Augen um den Mittelpunkt ihrer Verbindungslinie kurz angeschnitten werden. Die Beobachtungen sollen dabei so erfolgen, dass sie im System des Mittelpunkts gleichzeitig stattfinden, dies entspricht der Verarbeitung gleichzeitig im Mittelpunkt ankommender Bilder bei symmetrischer Übertragung. An dieser Stelle sei nochmals darauf hingewiesen, dass die Augen selbst als punktförmig angenommen werden, und dass eventuelle Einflüsse der Beschleunigung bei der kreisförmigen Bewegung vernachlässigt werden.

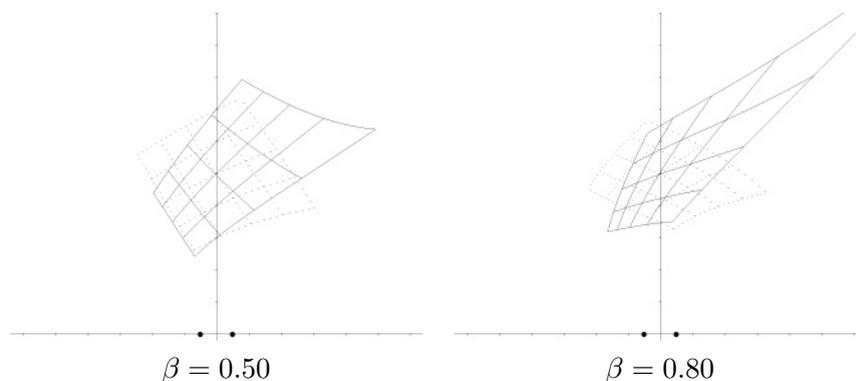


Abbildung 3.9: *Stereoskopische Wahrnehmung bei rotierendem Beobachter. Im Vergleich zu 3.6 (gestrichelt) tritt eine zusätzliche Verzerrung auf.*

Abbildung 3.9 zeigt die stereoskopische Wahrnehmung eines Gitters, wobei die Augen so um den Mittelpunkt gedreht werden, dass ihre Verbindungslinie zu jedem Zeitpunkt orthogonal zur retardierten Position des Mittelpunktes des Gitters ist. Dies entspricht der visuellen Verfolgung eines Körpers, wobei allein der Kopf bewegt wird. Die Geschwindigkeit der Augen erreicht dabei $1/10$ der Geschwindigkeit des verfolgten Körpers, die

zusätzlich auftretenden Verzerrungen sind deutlich sichtbar, darüber hinaus schneiden sich die Sichtlinien nicht mehr für alle Punkte der Ebene.

3.4.4 Beschleunigte Bewegungen

Die stereoskopische Beobachtung schnell bewegter Körper ist nicht auf Bewegungen mit gleichbleibender Relativgeschwindigkeit beschränkt. Abschließend soll hier noch das Erscheinungsbild beschleunigter Punkte untersucht werden.

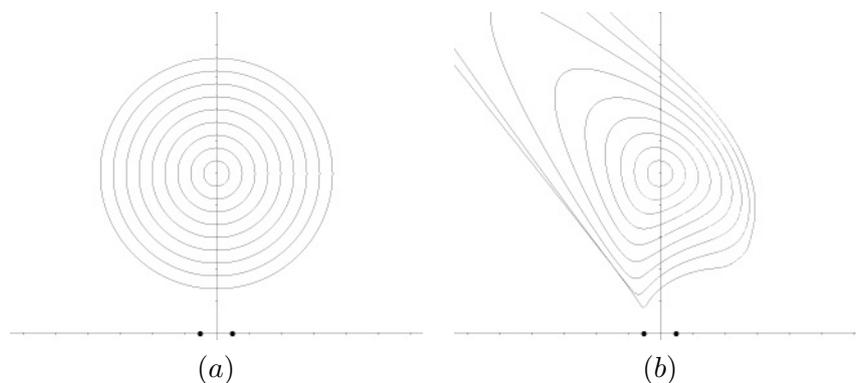


Abbildung 3.10: *Stereoskopische Wahrnehmung (b) von Punkten, die sich auf verschiedenen Kreisbahnen (a) bewegen. Alle Punkte bewegen sich mit der gleichen Winkelgeschwindigkeit, die Punkte auf dem äußersten Kreis erreichen $\beta = 0.9$*

Der Beobachter selbst soll dabei keine Beschleunigung erfahren, also in einem Inertialsystem ruhen. Die retardierten Positionen des beschleunigten Punktes sind wieder die Schnittpunkte seiner Weltlinie mit den Lichtkegeln für die beiden Augen. Anders als im Fall der gleichförmigen Bewegung kann die retardierte Position in den meisten Fällen nur numerisch bestimmt werden, aus den so gewonnenen Sichtlinien für die beiden Augen lassen sich dann wieder die stereoskopisch wahrgenommene Entfernung und Position bestimmen. Für die Bewegung auf einer Kreisbahn ergibt sich folgendes Bild (Abbildung 3.10): Wie zu erwarten entspricht die wahrgenommene Entfernung nicht der wirklichen Entfernung, die Bahn des Punktes wird mit zunehmender Geschwindigkeit scheinbar immer stärker deformiert. Kann die Bewegung während der Zeit, die der Laufzeitdifferenz zwischen den beiden Augen entspricht, durch eine geradlinige Bewegung angenähert werden kann, so kann die scheinbare Position auch mittels der Abbildung T_β für die momentane Geschwindigkeit bestimmt werden.

Das Aussehen eines Körpers bei beschleunigter Bewegung lässt sich dann aus den scheinbaren Positionen der einzelnen Punkte seiner Oberfläche bestimmen, hierbei ist allerdings der Einfluss der Beschleunigung auf die Form des Körpers und die Weltlinien der Punkte zu beachten.

3.5 Entfernungsbestimmung bei relativistischen Geschwindigkeiten

In den letzten Abschnitten wurde gezeigt, dass die Entfernung und die Form eines Körpers anders als im klassischen Fall mit quasi unendlicher Lichtgeschwindigkeit unter Verwendung eines Stereoalgorithmus, der die Effekte der Lichtlaufzeit und der Lorentzkontraktion nicht berücksichtigt, falsch wiedergegeben werden. Dies gilt allgemein für die stereoskopische Entfernungsbestimmung, wenn die auftretenden Geschwindigkeiten vergleichbar mit der Signalgeschwindigkeit sind. Hier stellt sich nun die Frage, ob es für einen intelligenten Beobachter, der diese Effekte berücksichtigt, prinzipiell möglich ist, die korrekten Entfernungen aus einem Stereobildpaar zu bestimmen.

Ist die Relativgeschwindigkeit bekannt, so lassen sich die Sichtlinien der Geschwindigkeit entsprechend umrechnen, aus den Schnittpunkten erhält man wie im statischen Fall die Entfernung der einzelnen Punkte und damit die Form des Körpers. Alternativ kann der Beobachter auch aus der scheinbaren Position und Form des Körpers mit Hilfe der Inversen der Abbildung T_β für die jeweilige Geschwindigkeit die wirkliche Entfernung und Form des Körpers bestimmen. Die größte Schwierigkeit besteht wie im statischen Fall darin, die schon in Abschnitt 3.2 erwähnte Zuordnung von Punkten oder Bildmerkmalen in den beiden Bildern vorzunehmen.

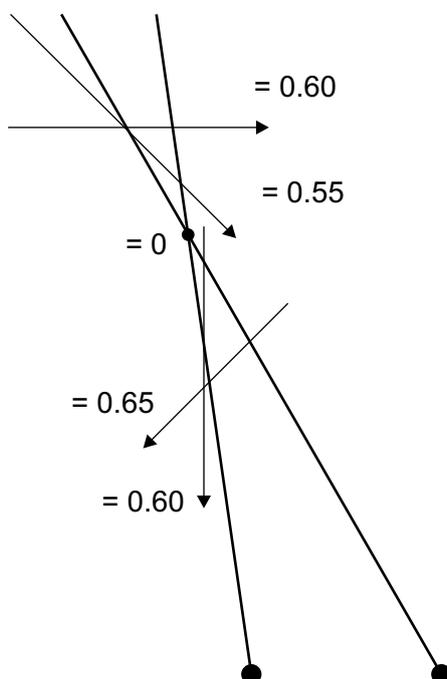


Abbildung 3.11: Verschiedene Bewegungen können den gleichen stereoskopischen Eindruck erzeugen. Der bewegte Punkt wird bei allen dargestellten Bewegungen an der Position für $\beta = 0$ wahrgenommen.

Anders stellt sich die Situation dar, wenn die Relativgeschwindigkeit nicht bekannt ist. Liegen außer dem Stereobildpaar keine weiteren Informationen vor, so lassen sich die Entfernungen im Allgemeinen nicht eindeutig bestimmen. Ist die Bewegung beispielsweise parallel zur Verbindungslinie der Augen (vgl. Abschnitt 3.4.1), so kann der Beobachter nicht zwischen einem ruhenden Punkt an einer bestimmten Position und einem bewegten Punkt mit der selben scheinbaren Position unterscheiden. Abbildung 3.11 zeigt verschiedene Kombinationen für die Entfernung und Geschwindigkeit eines Punktes, die alle für einen Beobachter die gleichen Sichtlinien erzeugen. Für den Beobachter ist damit nicht möglich, anhand der beiden Bilder die Position des Punktes im Raum zu bestimmen.

Erst wenn die Form und Größe des beobachteten Körpers schon bekannt sind, kann der Beobachter die Entfernung (und damit auch die Geschwindigkeit) bestimmen.

Im allgemeinen Fall werden die Sichtlinien eines Punktes für die beiden Kameras windschief sein. Dieser Unterschied der Bildpositionen in vertikaler Richtung kann nur durch eine Bewegung des Punktes entstanden sein und gibt somit eine Untergrenze für die Relativgeschwindigkeit. Ohne weitere Informationen lassen sich allerdings auch in diesem Fall die Entfernung und Form nicht eindeutig bestimmen.

Bei schnellen Bewegungen ist es also für einen Beobachter nicht möglich, die Entfernung und Form eines Körpers allein aus einem Stereobildpaar eindeutig zu bestimmen, dies gelingt erst mit weiteren Information über die Relativgeschwindigkeit, beispielsweise aus einer Bildfolge.

3.6 Beispiele

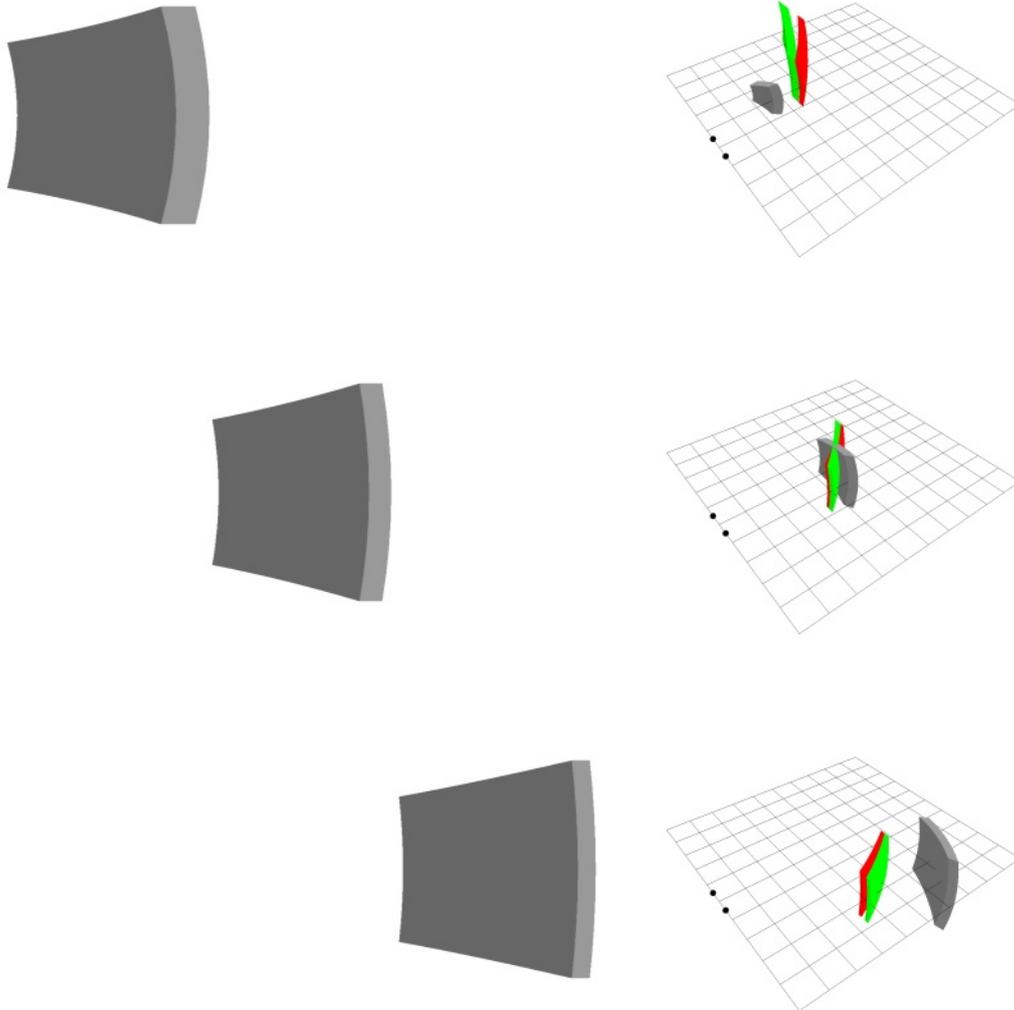


Abbildung 3.12: *Entstehung des dreidimensionalen Eindrucks bei stereoskopischer Beobachtung: Der Würfel bewegt sich mit 99% der Lichtgeschwindigkeit für den Beobachter von links nach rechts. Die linke Spalte zeigt den Würfel aus der Sicht des Beobachters zu verschiedenen Zeitpunkten, rechts sind die retardierten Bilder (Photokörper) für die beiden Augen (links:rot / rechts:grün) und der sich daraus ergebende dreidimensionale Eindruck (grau) dargestellt.*

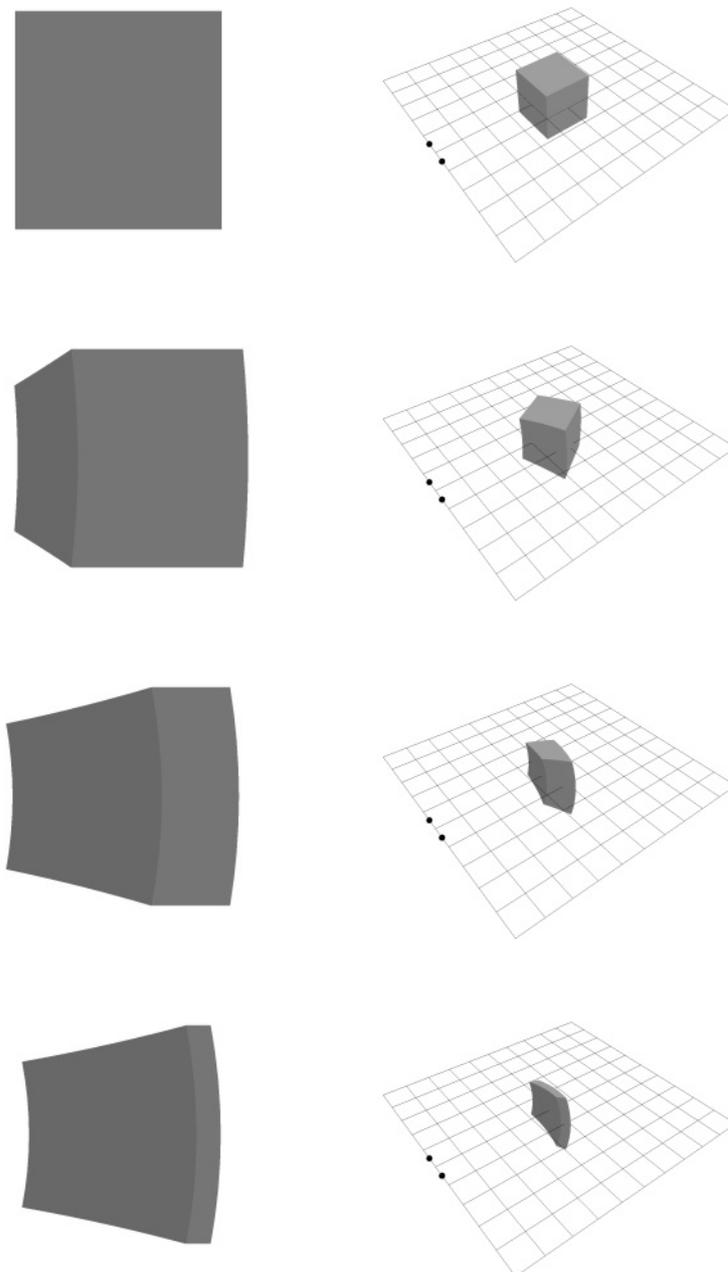


Abbildung 3.13: *Räumliche Wahrnehmung eines schnell bewegten Körpers: Links der Würfel aus der Sicht des Beobachters, rechts die dreidimensionale Wahrnehmung wie in 3.12 für verschiedene Geschwindigkeiten. Die Punkte markieren wieder die Position der Augen im Abstand $2a$, der Würfel bewegt sich in einer Entfernung von $5a$ von links nach rechts. Die starke perspektivische Verzerrung wird durch den kleinen Abstand von Beobachter und Würfel im Verhältnis zur Größe des Würfels verursacht. Mit zunehmender Geschwindigkeit erscheint der Körper zunächst gedreht und später entlang der Bewegungsrichtung gestaucht.*

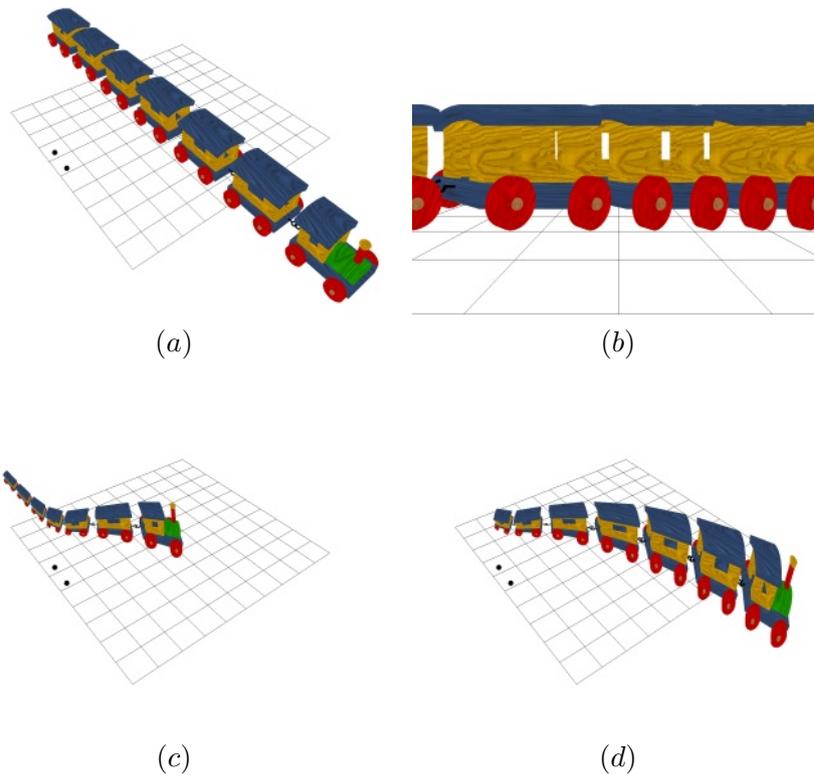
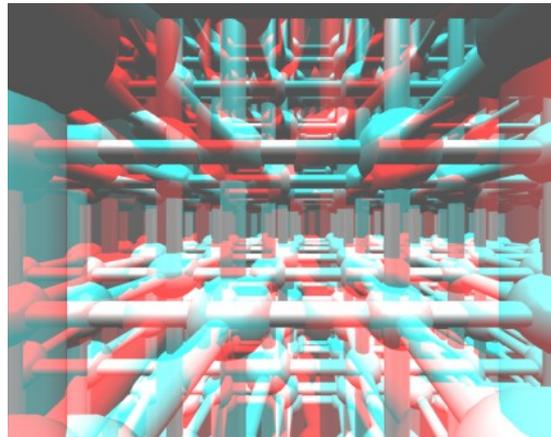
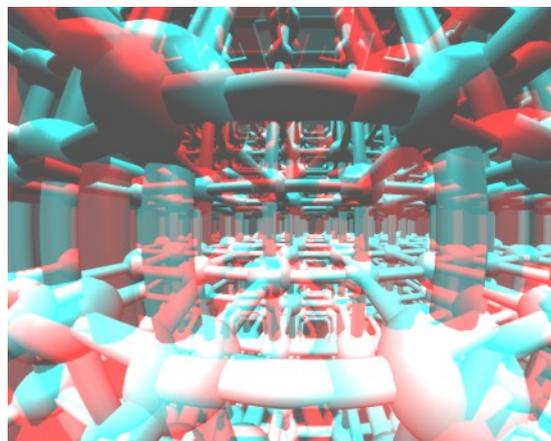


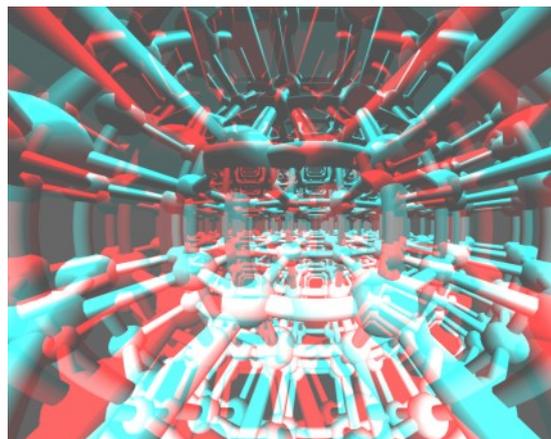
Abbildung 3.14: *Wahrgenommene Bewegungsrichtung bei stereoskopischer Beobachtung: Der Zug aus Abbildung 2.7 (a) bewegt sich mit 80% der Lichtgeschwindigkeit für den Beobachter von links nach rechts. (b) zeigt den Zug, wie ihn der Beobachter sieht, (c) und (d) stellen den vom Beobachter wahrgenommenen dreidimensionalen Eindruck dar. Die scheinbare Bewegungsrichtung entspricht immer der momentan gesehenen Verdrehung, für den Beobachter bewegt sich der Zug von links vorne nach rechts hinten. Da sich der Winkel, unter dem der Zug gesehen wird, nicht verändert, scheint der Zug dabei größer zu werden.*



$$\beta = 0$$



$$\beta = 0.5$$



$$\beta = 0.8$$

Abbildung 3.15: Räumliches Gitter in stereoskopischer Darstellung für verschiedene Geschwindigkeiten. Die gesehene Streckung in Bewegungsrichtung und die Krümmung der Gitterebenen wird auch stereoskopisch als Streckung und Krümmung im dreidimensionalen Raum wahrgenommen.

Kapitel 4

Projektion auf gekrümmte Flächen

Die letzte Stufe der Visualisierung ist die physikalische Darstellung der Information für das Auge. Bei Simulationen und Visualisierungen virtueller Welten ist das Ziel, die spektrale Leuchtdichte der Umgebung für die aktuelle Position so gut wie möglich abzubilden. Bei der meistens verwendeten Darstellung auf Monitoren wird nur ein kleiner Raumwinkel der virtuellen Welt dargestellt, mit allen Einschränkungen und Nachteilen, die das begrenzte Sichtfeld bedingt. Um wenigstens den dargestellten Ausschnitt der virtuellen Welt zu vergrößern, wird ein größerer Ausschnitt dargestellt, als es dem Raumwinkel des Monitors aus der Position des Benutzers entspricht. Die menschliche Wahrnehmung ist robust genug, so dass auch in diesem Fall die Position und Bewegung in der virtuellen Welt genau genug abgeschätzt werden können, erst bei Darstellung sehr großer Blickwinkel (Fischauge) auf einem Monitor treten sichtbare Bewegungsartefakte auf.

Im Gegensatz zu den in Abschnitt 2.5 beschriebenen grundlegenden Einschränkungen, die für die Wiedergabe der Leuchtdichte und der spektralen Verteilung in Kauf genommen werden müssen, kann mit entsprechender Technik der gesamte Raumwinkel um die aktuelle Beobachterposition in der virtuellen Welt dargestellt werden. Mit mehreren Projektoren und einer entsprechenden Projektionsfläche kann der Ausschnitt auf das gesamte Gesichtsfeld erweitert werden, die Auflösung ist besser als die des Auges.

Bei größeren Gesichtsfeldern muss die virtuelle Welt für die Benutzerposition möglichst unverzerrt dargestellt werden, ansonsten treten insbesondere bei Bewegungen in der virtuellen Welt wahrnehmbare Verzerrungen auf. Statt der dreidimensionalen Welt wird dann nur noch ein flaches und verzerrtes Bild gesehen.

Besonders genau müssen die Richtungen bei der stereoskopischen Visualisierung und bei der Projektion mehrerer zusammengesetzter Bilder übereinstimmen. Die Projektionen müssen dabei nicht nur am Rand oder im Überblendbereich übereinstimmen,

ansonsten können gerade Linien einen Knick aufweisen oder gebogen erscheinen. Störender sind die Effekte, die bei Bewegung auftreten. Die menschliche Wahrnehmung ist sehr empfindlich für Relativbewegungen, kleine Abweichungen werden bei bewegten Objekten sofort wahrgenommen, die Welt erscheint wie durch eine ungleichmäßig dicke Glasscheibe betrachtet.

Bei der Projektion auf eine gekrümmte Projektionsfläche wie beispielsweise eine Kuppel ist es nicht mehr ausreichend, den gerenderten Bildausschnitt anzupassen und den Projektor mit Hilfe der eingebauten Korrekturmechanismen wie Keystonekorrektur einzurichten. Die Korrektur muss in diesen Fällen auf Pixelebene erfolgen und kann auch eine Helligkeitsanpassung beinhalten, so dass möglichst viele Tiefenhinweise der Projektionsfläche selbst unterdrückt werden. Die folgenden Überlegungen gelten auch für die Darstellung auf einem Monitor, wenn der Bildschirm als Projektionsfläche aufgefasst wird und eine korrekte Darstellung des Gesichtsfeldes, beispielsweise bei stereoskopischer Visualisierung, gewünscht wird.

Für die notwendigen Anpassungen der Geometrie und Helligkeiten bei der Projektion mit mehreren Projektoren auf eine gekrümmte Fläche ist eine Vielzahl von Verfahren beschrieben, vergleiche beispielsweise [10, 28, 45]. In diesem Kapitel wird ein Verfahren vorgestellt, das die notwendigen Korrekturen auf Pixelebene durchführt, und das sich nahtlos in das Verfahren zur relativistischen Verzerrung integrieren lässt.

4.1 Entzerrung

Das Bild auf der Projektionsfläche soll nicht als flaches Bild wahrgenommen werden, vielmehr soll die Umgebung dreidimensional wahrgenommen werden. Im Idealfall hat jedes kleine Flächenelement auf der Projektionsfläche für den Benutzer die gleiche spektrale Leuchtdichte wie das entsprechende Raumwinkelement für einen Beobachter an einer bestimmten Position in der virtuellen Welt. Abbildung 4.1 zeigt eine typische Anordnung mit einem Projektor P , der Projektionsfläche C und einem Beobachter B .

Der Projektor befindet sich an der Position \mathbf{R}_p , seine Orientierung wird durch die Matrix M_p beschrieben. Die gesamte Abbildungsfunktion der Optik sei wie bei den abbildenden Systemen in Abschnitt 3.2 in der Funktion

$$\varphi_p : \mathbb{R}^2 \rightarrow \mathbb{S}^2 \subset \mathbb{R}^3 \quad (4.1)$$

zusammengefasst, die jedem Bildpunkt die Projektionsrichtung in Bezug auf die Projektorachsen zuordnet. $S_C(\mathbf{r}, \mathbf{n})$ sei der Schnittpunkt eines Strahls mit der Richtung \mathbf{n} durch den Punkt \mathbf{r} mit den Projektionsfläche C . Für einen Bildpunkt (x, y) ist der Schnittpunkt damit durch

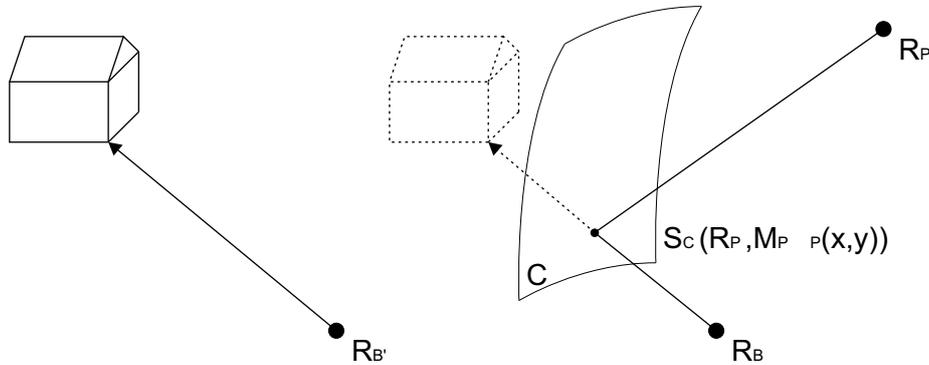


Abbildung 4.1: Für einen Betrachter B sollen die Objekte in der selben Richtung erscheinen wie für einen Beobachter B' in der virtuellen Umgebung

$$S_C(\mathbf{R}_p, M_p \varphi_p(x, y)) \quad (4.2)$$

gegeben. Für den Benutzer an der Position \mathbf{R}_B entspricht das der Richtung

$$S_C(\mathbf{R}_p, M_p \varphi_p(x, y)) - \mathbf{R}_B \quad (4.3)$$

Mit diesen Gleichungen lässt sich für jeden Bildpunkt (x, y) aus der Geometrie der Anordnung von Projektor, Projektionsfläche und Benutzer die entsprechende Richtung berechnen, deren Leuchtdichte er darstellen muss.

Die Korrektur erfolgt in zwei Schritten. Im ersten Schritt wird die Rundumsicht erstellt, im zweiten Schritt erfolgt die korrigierte Darstellung. Für jeden Bildpunkt wird die entsprechende Richtung berechnet und die entsprechende Farbe und Helligkeit aus der Rundumsicht dargestellt. Für eine feste Anordnung von Projektor, Projektionsfläche und Beobachter können die entsprechenden Richtungen beim Programmstart berechnet und für die Umrechnungen zur Laufzeit gespeichert werden. Soll sich die Beobachterposition während der Laufzeit verändern können, zum Beispiel mit Headtracking in einer CAVE oder Kuppelprojektion, so können bei der Initialisierung die Schnittpunkte mit der Projektionsfläche für alle Bildpunkte berechnet und gespeichert werden. Die eigentlichen Richtungen sind dann abhängig von der Beobachterposition und werden erst während der Laufzeit auf der Grafikhardware für jeden Bildpunkt be-

rechnet. Wird die Bewegung auch auf den Beobachter in der virtuellen Welt übertragen, so ergibt sich für den Benutzer eine korrekte Bewegungsparallaxe bei Bewegungen des Kopfes.

Im ersten Schritt ist es dabei nicht notwendig, die gesamte Rundumsicht zu erstellen. Der gerenderte Ausschnitt der virtuellen Welt muss von der aktuellen Position aus gesehen lediglich alle Richtungen enthalten, die den projizierten Bildpunkten entsprechen. Um eine hohe Auflösung zu erreichen und um möglichst wenig Bildpunkte zu rendern, die im zweiten Schritt nicht dargestellt werden, wählt man ein Frustum, das die notwendigen Richtungen so eng wie möglich umschließt. Ist $P_B : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ die beim Rendern verwendete Projektion (gegebenenfalls einschließlich einer Rotation und Translation, die die Beobachterposition in der virtuellen Welt mit der Benutzerposition verbindet), so muss der Bildpunkt (x, y) des Projektors das gerenderte Bild an der Position $V(x, y)$ sampeln, wobei V mit 4.3 durch

$$V(x, y) = P_B S_C(\mathbf{R}_p, M_p \varphi_p(x, y)) \quad (4.4)$$

gegeben ist. Der Einfluss der Korrektur auf die Bildqualität ist in Kapitel 5.3 beschrieben.

Die Ähnlichkeit des Verfahrens mit dem Verfahren der bildbasierten relativistischen Visualisierung ist auffallend, in beiden Fällen wird zunächst ein Teil der Rundumsicht gerendert, um in einem zweiten Schritt verzerrt dargestellt zu werden. Aus diesem Grund lassen sich beide Verzerrungen nahtlos zusammenfassen. Im Fall der festen Beobachterposition sind während der Laufzeit keine zusätzlichen Rechenschritte im Vergleich zur relativistischen Visualisierung notwendig. Die Details der Implementierung sind in Abschnitt 5.1.2 beschrieben.

Projektion auf ebene Flächen

Im speziellen Fall der Projektion auf ebene Flächen oder der Darstellung auf einem Monitor kann auf den zweiten Schritt verzichtet werden. Beim Projektor darf die Funktion φ_p keine nichtlinearen Verzerrungen beinhalten, was für die meisten aktuellen Projektoren in guter Näherung erfüllt ist. Alle auftretenden Abbildungen in Gleichung 4.4 sind dann projektive Abbildungen. Durch geeignete Wahl des Frustums und der Projektionsmatrix beim Rendern kann die Abbildung V zur identischen Abbildung gemacht werden. Je nach Anordnung ist es in manchen Fällen sogar ausreichend, mit Hilfe der Keystonekorrektur am Projektor, die die Funktion φ_P verändert, die Abbildung V zur Identität zu machen. In diesem Fall kann der zweite Schritt wegfallen. Die Details sind in Anhang A.2 beschrieben.

4.2 Anpassung der Leuchtdichte

Eine vollständige Reproduktion der Intensitätsverteilung für den Benutzer muss neben der geometrischen Umrechnung auch eine Umrechnung der Leuchtdichte beinhalten. Die Leuchtdichte auf der Projektionsfläche soll der berechneten Leuchtdichte aus derselben Richtung entsprechen, oder, falls eine Skalierung der Leuchtdichte vorgenommen wird, so sollen gleiche berechnete Leuchtdichten auch für verschiedene Richtungen für den Benutzer durch die gleiche Leuchtdichte dargestellt werden.

Damit kann erreicht werden, dass das Gehirn weniger Information über die Projektionsfläche aus der Krümmung von eigentlich geraden Linien bei Bewegung oder über die Winkelneigung zur Projektionsquelle aus der Schattierung bekommt. Die projizierten Körper werden dann weniger als flaches Bild auf der Projektionsfläche wahrgenommen, sondern als Körper im Raum.

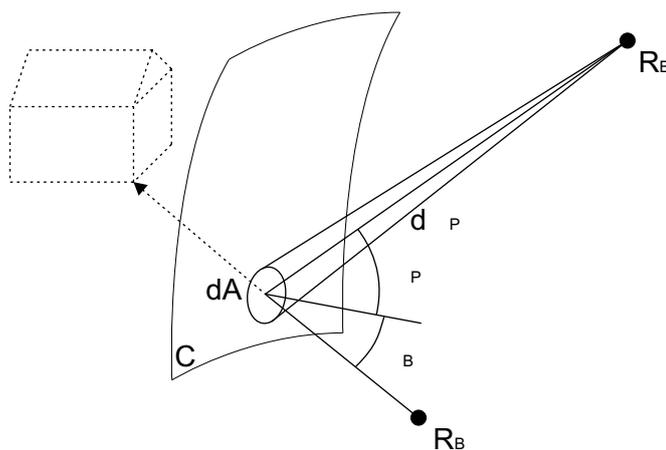


Abbildung 4.2: *Anpassung der Leuchtdichte: Geometrie der Anordnung von Projektor, Projektionsfläche und Benutzer*

Bei bekannter und fester Geometrie der Projektoren und Projektionsfläche lässt sich die Leuchtdichte auf der Projektionsfläche berechnen und bei der Erzeugung des Bildes so anpassen, dass eine konstante Leuchtdichte auf der gesamten Projektionsfläche erreicht wird. Ist die Reflektion der Projektionsfläche ideal diffus, gehorcht also dem Lambertischen $\cos(\theta)$ Gesetz, so ist die Leuchtdichte für den Beobachter unabhängig von seiner Position vor der Projektionsfläche.

Im allgemeinen Fall wird die Reflektion nicht ideal diffus sein. Mit der Reflektionsfunktion B kann die an der Position des Beobachters auf der Fläche erzeugte Leuchtdichte

berechnet werden (Abbildung 4.2). Die vom Projektor ausgestrahlte Leistung trifft ein Flächenelement dA auf der Projektionsfläche C . Vom Projektor aus gesehen erscheint das Flächenelement dA unter dem Raumwinkel $d\Omega_P$. Die Leuchtdichte der Projektionsfläche in Richtung des Beobachters ist dann

$$L = B(\theta_P, \theta_B) \frac{\cos \theta_P dA}{r_P^2} I_P \quad (4.5)$$

Für die Implementierung der Leuchtdichteanpassung muss schließlich noch die Abhängigkeit der Strahlstärke des Projektors in Abhängigkeit der Eingangswerte (Gamma-Funktion) bekannt sein. Der entsprechende Korrekturfaktor wird für jeden Bildpunkt beim Programmstart aus der Geometrie der Anordnung berechnet und in einer Textur abgelegt. Eine weitergehende Korrektur der unterschiedlichen spektralen Reflektanz in verschiedenen Bereichen der Projektionsfläche kann prinzipiell in die Leuchtdichteumrechnung mit einbezogen werden, ist aber in den meisten Fällen nicht erforderlich.

Besonders wichtig ist die korrekte Anpassung der Leuchtdichte beim Einsatz von mehreren Projektoren, die gemeinsam ein Bild erzeugen sollen. Mit der Korrektur kann in diesem Fall nicht nur die Helligkeit der einzelnen Projektoren untereinander angepasst werden, sondern auch die Überblendung mit vollständiger Gammakorrektur im Überlappungsbereich.

Die gesamte Korrektur greift erst in der letzten Verarbeitungsstufe der Visualisierung ein und ist daher für die eigentliche Anwendung völlig transparent. Die Integration der beschriebenen Leuchtdichte-, Farb- und Geometrie-Anpassung in das Verfahren zur relativistischen Visualisierung wird in Abschnitt 5.1 dargestellt.

Automatische Feineinstellung

Die beschriebene Anpassung an die Geometrie der Anordnung kann vollständig auf der Grafikhardware durchgeführt werden, und damit auch jederzeit durch die Angabe neuer Parameter verändert werden. Dies lässt sich dazu verwenden, die Feineinstellung der Ausrichtung und Leuchtdichteanpassung interaktiv oder automatisch mit einem Verfahren wie beispielsweise [44] durchzuführen. Mit Hilfe einer Kamera, die die projizierten Testbilder aufnimmt, kann die Korrektur bei der Aufstellung und im laufenden Betrieb eines Mehrprojektorsystems aus den Abweichungen der Markierungen für die verschiedenen Projektoren mit Ausgleichsverfahren berechnet werden.

Der Farbgleich der Projektoren untereinander kann bei gegebenem Feedback durch eine Kamera ebenfalls automatisch vorgenommen werden (vergleiche beispielsweise [28]), alle Korrekturen können wieder in die Umrechnung für die relativistische Verzerrung integriert werden.

4.3 Anwendungen

Zur Verdeutlichung der Konzepte bei der Projektion auf gekrümmte Flächen seien an dieser Stelle drei Anwendungen – eine Laserkuppelprojektion, die interaktive Visualisierung der durch ein schwarzes Loch hervorgerufenen Verzerrungen und die stereoskopische Visualisierung – kurz skizziert.

4.3.1 Kuppelprojektion

Die Kuppelprojektion [3] arbeitet mit sechs Laserprojektoren. Die Projektionsfläche ist eine Halbkugel mit 14 m Durchmesser. Als Projektoren werden sechs Laserprojektoren verwendet, die Optiken befinden sich am unteren Rand der Kuppel, die eigentlichen Laserquellen befinden sich in einem abgetrennten Raum außerhalb der Kuppel. Die Laserquellen liefern Licht bei den Wellenlängen $465nm$, $547nm$ und $608nm$.

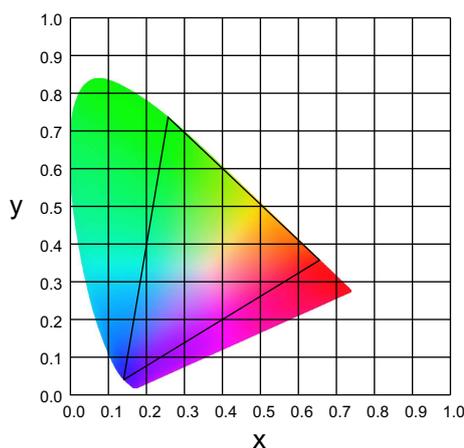


Abbildung 4.3: *CIE XY Diagramm mit dem darstellbaren Farbbereich der Laserprojektoren*

Abbildung 4.3 zeigt die damit darstellbaren Farben im CIE XY Diagramm. Der Farbbereich der Laserprojektoren geht über den Bereich hinaus, der normalerweise mit Bildschirmen darstellbar ist.

Mit einer Leistung von 10 W erreicht ein einzelner Projektor eine Lichtstärke von etwa 2000 ANSI Lumen.

Der angegebene Kontrastumfang der Laserquelle von 50000:1 ist für die Praxis nicht von Bedeutung (außer bei der Darstellung weniger heller Bildelemente wie bei der Darstellung des Sternenhimmels), da dunkle Bereiche durch reflektiertes Licht von anderen Stellen der Projektionsfläche aufgehellt werden.

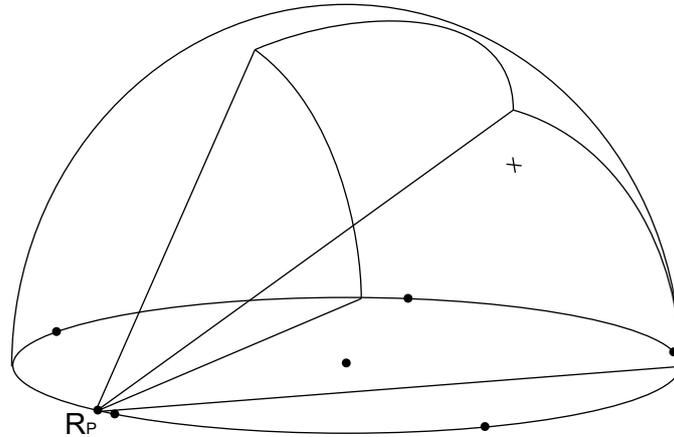


Abbildung 4.4: Aufbau der Kuppelprojektion. Die Punkte markieren die Positionen der Projektoren, fünf Projektoren erzeugen ein Panorama bis zu einer Höhe von etwa 60 Grad, der sechste Projektor stellt den Abschluss um den Zenit dar.

Für die Bilderzeugung wird der Strahl über rotierende Spiegel zeilenweise über die Projektionsfläche geführt, die Helligkeit der Laserquelle wird dem Bildinhalt entsprechend moduliert. Der Bildpunkt in der Mitte wird auf die optische Achse abgebildet, für alle anderen Bildpunkte ist die Ablenkung $1/40$ Grad pro Pixel. Durch eine spezielle Optik wird der Winkel zur optischen Achse um den Faktor 1.8 vergrößert. Die Abbildungsfunktion φ_p für den Laserprojektor ist damit

$$\varphi_p(x, y) = \begin{pmatrix} \cos(1.8\alpha) \\ -\sin(1.8\alpha) \sin(\kappa x)/a \\ \sin(1.8\alpha) \cos(\kappa x) \sin(\kappa y)/a \end{pmatrix} \quad (4.6)$$

mit

$$\begin{aligned} a &= \sqrt{1 - \cos^2(\kappa x) \cos^2(\kappa y)} \\ \alpha &= \sin^{-1} a \\ \kappa &= \frac{\pi}{7200} / \text{Pixel} \end{aligned} \quad (4.7)$$

Die Projektoren sind am unteren Rand der Halbkugel angebracht, die Schnittpunkte mit der Projektionsfläche sind dann einfach durch

$$S_{Kuppel}(\mathbf{R}_p, \varphi_p) = \mathbf{R}_p - 2(\varphi_p \cdot \mathbf{R}_p) \varphi_p \quad (4.8)$$

gegeben.

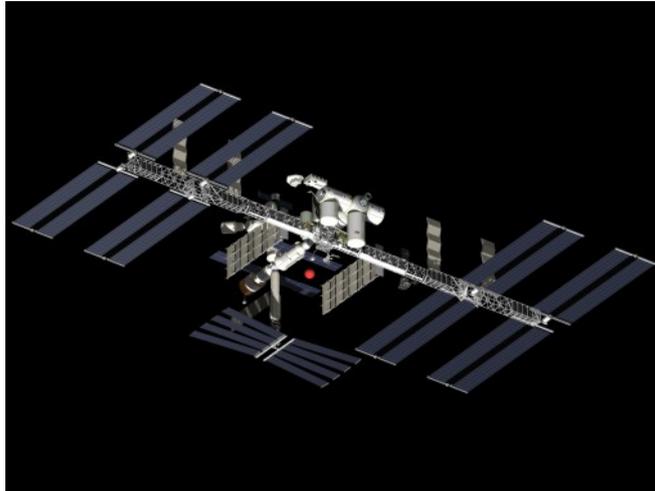
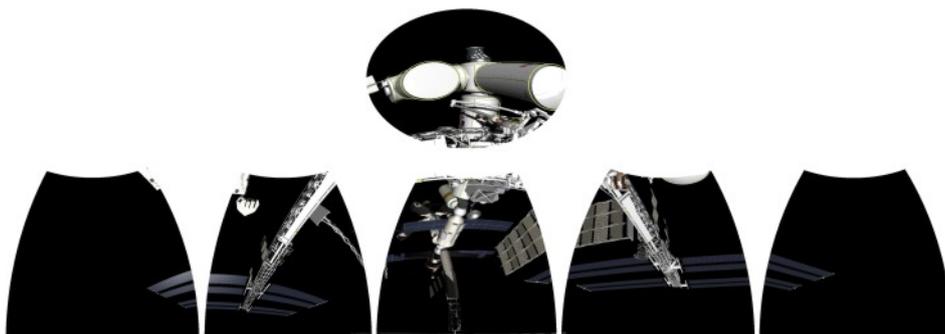


Abbildung 4.5: 3D-Modell der internationalen Raumstation ISS. Die Beobachterposition in der virtuellen 3D-Welt ist rot markiert.



(a)



(b)

Abbildung 4.6: Für einen Benutzer in der Mitte der Kuppel (a) und einen Benutzer bei $R/2$ (b) umgerechnete Bilder für die sechs Projektoren. Die Unterschiede sind allein durch die geänderte Benutzerposition in der Kuppel bedingt, die Position des virtuellen Beobachters relativ zur Raumstation wurde nicht verändert.

Zur Verdeutlichung der Abhängigkeit von der Benutzerposition zeigt Abbildung 4.6 die umgerechneten Bilder eines 3D Modells der internationalen Raumstation ISS für die sechs Projektoren für zwei verschiedene Benutzerpositionen in der Kuppel. Die Position des virtuellen Beobachters in Bezug auf die Raumstation ist in beiden Fällen gleich, die Unterschiede sind allein durch die Umrechnung auf die Kuppel als Projektionsfläche für die verschiedenen Benutzerpositionen bedingt. Der Benutzer sieht unabhängig von seiner Position in der Kuppel das gleiche Bild der Raumstation.

Wird die Position in der virtuellen Welt entsprechend der Bewegung in der Kuppel variiert, so kann die virtuelle Welt interaktiv durchwandert werden. Dazu ist eine genaue Bestimmung der Position des Benutzers in der Kuppel via Headtracking notwendig.

4.3.2 Interaktive Verzerrung

Das beschriebene Verfahren der interaktiven Geometrie- und Helligkeitsanpassung erlaubt beliebige Verzerrungen sowie Farb- und Helligkeitsänderungen. Es kann nicht nur dazu verwendet werden, die Bilder bei der Projektion auf gekrümmte Flächen zu korrigieren, sondern auch um beliebige Abbildungen und Verzerrungen sowie Helligkeitsänderungen für einzelne Bildpunkte interaktiv auf Bilder anzuwenden. Die Umrechnung für ein Bild mit 1280×1024 Bildpunkten nimmt auf der Grafikhardware lediglich 2 ms in Anspruch, so dass sich die Umrechnungen problemlos auch in Echtzeit auf Videoquellen anwenden lassen.



Abbildung 4.7: *Interaktives schwarzes Loch*

Ein Beispiel für die schnelle Umrechnung eines Videostreams ist das Projekt *interaktives schwarzes Loch*, bei dem die von einer Kamera aufgenommenen Bilder des Benutzers so verzerrt werden, als ob sich ein kleines schwarzes Loch zwischen Kamera und Benutzer befände. Die komplette Verzerrung und gleichzeitige Anpassung des Kontrastes und der

Farbsättigung der Kamerabilder für die zwei Ausgabegeräte, einen Plasmabildschirm mit 1280×1024 Bildpunkten und einen Touchscreen mit einer Auflösung von 1024×768 Bildpunkten, erfolgt in nur 7 ms . Anhang A.5 beschreibt die technischen Details der Implementierung.

4.3.3 Stereoskopische Visualisierung

Mit der schnellen Umrechnung hat man auch bei Projektion auf ebene Flächen eine größere Flexibilität bei der Aufstellung der Projektoren und ist nicht auf die alleinige Verwendung der eingebauten Keystonekorrektur angewiesen. Dies lässt sich bei der stereoskopischen Darstellung nutzen, das Einsetzen der Augenpositionen als Benutzerposition in Gleichung 4.3 ergibt automatisch die richtigen Bildausschnitte für die beiden Augen.



Abbildung 4.8: *Aufbau für die stereoskopische Visualisierung. Zur Funktion des Fahrrades siehe Abschnitt 5.6.1*

Abbildung 4.8 zeigt einen Aufbau für die stereoskopische Visualisierung mit zwei Projektoren ohne jegliche Korrektur. Deutlich erkennbar ist die notwendige Anpassung der Geometrie und der Leuchtdichte der beiden Projektoren. Bei Verwendung des beschriebenen Verfahrens brauchen die Projektoren nicht mehr möglichst nah beieinander stehen, auch Anordnungen wie die oben gezeigte sind möglich. Steht eine Kamera zur Verfügung, so lassen sich die Verzerrungen für die exakte Stereodarstellung ohne Verwendung der Keystonekorrektur der Projektoren automatisch justieren.

Kapitel 5

Implementierung und Ergebnisse

Dieses Kapitel beschreibt die Implementierung des kompletten bildbasierten Algorithmus mit Berücksichtigung des Dopplereffekts, des Searchlighteffekts und der Geometrie- und Leuchtdichtekorrektur sowie Anwendungsbeispiele des Algorithmus in verschiedenen Applikationen. Die bildbasierte Technik wurde gewählt, weil sie für die vorhandenen 3D-Modelle einen ausgewogenen Kompromiss zwischen der besten Bildqualität und hohen Bildraten auf aktueller PC-Hardware ermöglicht. Darüber hinaus lässt sich die Anpassung für die Projektion auf gekrümmte Flächen auf natürliche Weise in das Verfahren einbinden. Als unabhängige Verarbeitungsstufe am Ende des Rendervorgangs kann es auch für 3D-Modelle verwendet werden, die mit wenigen großen Polygonen für die nichtrelativistische Visualisierung optimiert sind. Ein geometriebasiertes Verfahren ist für diese Modelle meistens nicht brauchbar, siehe dazu Abschnitt 5.3 (Bildqualität).

Die Software wurde in der für die systemnahe Programmierung weit verbreiteten Programmiersprache C++ [55] geschrieben und mit dem Intel C++ Compiler (Version 7.1) [24] übersetzt. Als Grafikschnittstelle wurde OpenGL 1.5 [52] mit programmierbaren Vertex- und Pixelshadern entsprechend den Erweiterungen `GL_ARB_VERTEX_PROGRAM` und `GL_ARB_FRAGMENT_PROGRAM` in der Version 1.1 verwendet. OpenGL ist weit verbreitet und auf einer Vielzahl von Plattformen verfügbar. Alle vom verwendeten Betriebssystem (Microsoft Windows XP) abhängigen Funktionen sind in einem Modul gekapselt, so dass sich die Portierung auf ein anderes Betriebssystem wie Linux leicht durch Austausch dieses Moduls bewerkstelligen lässt.

3D-Modelle können vom Programm aus einer Datei mit einem einfachen Textformat geladen werden. Unterstützt werden verschiedene Objekte mit unterschiedlichen Materialien und Texturen. Die Datei kann mit Hilfe eines Exporter-Plugins direkt aus 3ds max [1] erstellt werden. In Kombination mit einer Software wie Polytrans [43] können so 3D-Modelle in einer Vielzahl von Formaten verwendet werden. Das Plugin erzeugt Triangle Strips für die einzelnen Objekte und berechnet alle Mipmabeenen für die Texturen.

Die Aussagen zur Geschwindigkeit und zur Implementierung eines Details aus Performancegründen, wie beispielsweise die Erstellung der Rundumsicht im Bildspeicher und nicht direkt in einer Textur mit der `WGL_ARB_RENDER_TEXTURE` Erweiterung, beziehen sich immer auf die vorhandene Grafikkhardware (nVidia GeForce FX5950 Ultra und ATI Radeon 9800XT), bei anderen Architekturen und sogar bei neuen Treiberversionen können andere Techniken schneller sein. Die Erwähnung einer verworfenen, langsameren Möglichkeit ist daher auch als Alternative für die Optimierung des Verfahrens auf anderer Grafikkhardware zu verstehen.

5.1 Teilschritte

Während der Laufzeit gliedert sich das Verfahren in drei Stufen: Zunächst werden die Position, Orientierung und Geschwindigkeit des Beobachters in der virtuellen Welt aus der Simulation bestimmt. Für die so gewonnene Position wird in der zweiten Stufe die Rundumsicht erstellt, anschließend wird für jeden Bildpunkt mit Hilfe der eben erstellten Rundumsicht der entsprechende Farbwert ermittelt. Die einzelnen Teilschritte lassen sich somit in Operationen pro Gesamtbild, pro Teilbild der Rundumsicht und pro Bildpunkt unterteilen.

Abbildung 5.1 stellt den Ablauf mit den Teilschritten der einzelnen Stufen schematisch dar. Je nach Anwendung können einzelne Teilschritte, wie die Geometrieanpassung oder das Tone Mapping, wegfallen.

5.1.1 Erzeugung der Rundumsicht

Die Rundumsicht wird aus einzelnen Teilbildern zusammengesetzt, die im Ruhesystem der darzustellenden Umgebung gerendert werden. Das Ergebnis sind eine oder mehrere Texturen, mit deren Hilfe sich in den folgenden Teilschritten für jede Richtung die Farbe oder der Materialindex bestimmen lassen.

Für jedes Teilbild wird dazu die Projektionsrichtung und der Bildausschnitt (Frustum) anhand der aktuellen Beobachterposition und Orientierung so festgelegt, dass alle Teilbilder zusammen den gesamten Raumwinkel abdecken. Bei der üblichen hardwarebeschleunigten Projektion auf eine Ebene sind also mindestens vier Teilbilder nötig.

Da bei der Erstellung der Teilbilder noch keine relativistischen Effekte einfließen, können alle bekannten Methoden zur Beschleunigung der Bilderzeugung, wie z.B. Frustum- und Occlusion-Culling bei hierarchisch unterteilten Modellen, verwendet werden. Beim Rendern der einzelnen Teilbilder muss lediglich darauf geachtet werden, dass das Ergebnis nicht von der Richtung abhängig ist, andernfalls können an der Grenze der einzelnen Seiten störende Übergänge sichtbar werden. Da meistens nur wenige Teilbilder erzeugt werden sollen, müssen die Blickwinkel recht groß sein, die Verwendung der Projek-

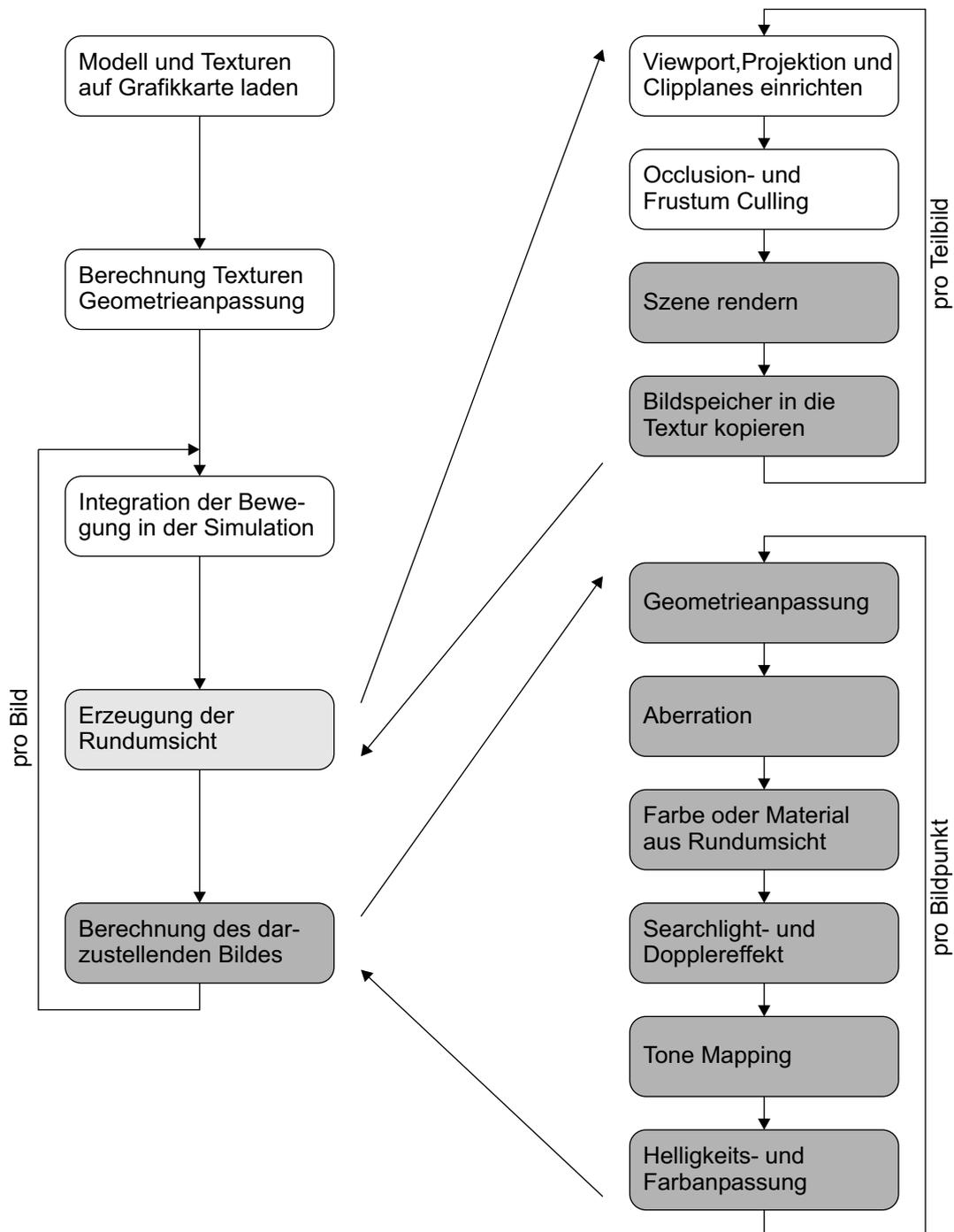


Abbildung 5.1: Teilschritte des kompletten bildbasierten Verfahrens zur relativistischen Visualisierung. Die grau hinterlegten Schritte werden auf der Grafikkarte ausgeführt.

tionsrichtung anstelle des Vektors vom Beobachter zum Objekt bei der Darstellung von Billboards, Nebel und der Beleuchtungsberechnung kann dann zu den genannten Bildfehlern führen.

Prinzipiell können beliebig viele Teilbilder zur Erzeugung der Rundumsicht verwendet werden, von praktischer Bedeutung sind jedoch nur Verfahren mit vier beziehungsweise sechs Seiten.

Tetraeder Map

Um den gesamten Raumwinkel durch die Projektion auf einzelne Flächen abzudecken, sind mindestens vier Teilbilder notwendig. Die einzelnen Teilbilder entsprechen dabei den Seitenflächen eines Tetraeders, dessen Schwerpunkt an der Beobachterposition liegt. Die Bilder können so erzeugt werden, dass sie sich in einer Textur, der *Tetraeder Map*[17], zusammenfassen lassen. Dabei werden jeweils zwei Seitenflächen zusammen in den Bildspeicher gerendert, so dass der Bildspeicher nur zweimal in die Textur kopiert werden muss.

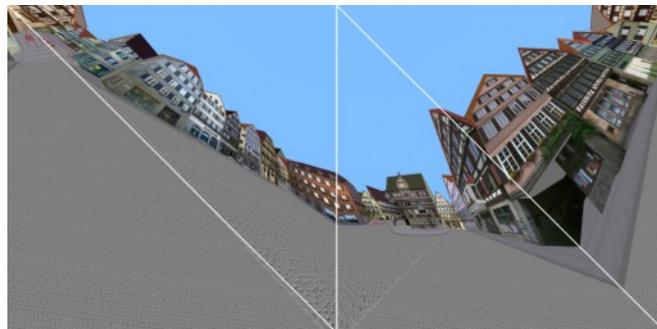


Abbildung 5.2: *Tetraeder Map* für die Rundumsicht aus Abbildung 2.8

Abbildung 5.2 zeigt die Tetraeder Map für die Rundumsicht aus Abbildung 2.8, die Unterteilung in die einzelnen Teilbilder ist markiert. Die Tetraeder Map ist besonders dann vorteilhaft, wenn die Erzeugung des einzelnen Teilbildes aufwendig ist, da man hier mit nur vier Bildern auskommt. Anhang A.3 beschreibt die Projektionen und Clip-Planes zur Erzeugung der Teilbilder.

Das Sampling der Tetraeder Map, also die Bestimmung der Position in der Textur für eine bestimmte Richtung der Rundumsicht, ist auf aktuellen Grafikkarten nicht hardwarebeschleunigt, hier wird einfach eine weitere Textur als Lookup-Table verwendet, siehe auch Abschnitt 5.1.3.

Cubemap

Die Zusammenfassung von sechs einzelnen Texturen zu einer Cubemap ist in der Computergrafik weit verbreitet. Jede einzelne quadratische Textur entspricht dabei der Seite eines Würfels, entsprechend einfach lassen sich die Projektion und das Frustum bei der interaktiven Erzeugung einrichten.

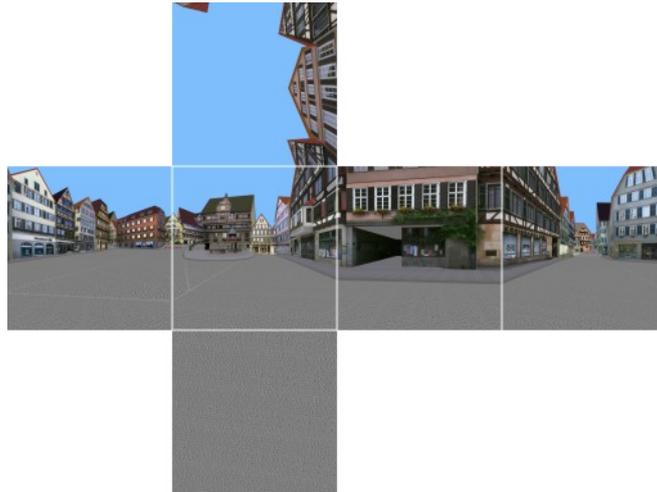


Abbildung 5.3: *Cubemap für die Beobachterposition in Abbildung 2.8*

Abbildung 5.3 zeigt die Cubemap wieder für die Rundumsicht aus Abbildung 2.8 zum Vergleich mit der Tetraeder Map. OpenGL unterstützt Cubemaps als eigenes Texturformat, dadurch ist die Verwendung der Cubemap der Verwendung der Tetraeder Map vorzuziehen, wenn die Applikation eher durch die Füllrate begrenzt ist, und die Erstellung von zwei zusätzlichen Teilbildern nicht ins Gewicht fällt.

Auf den verwendeten Grafikkarten ist das Rendern in den Bildspeicher und das anschließende Kopieren in die Cubemap oder Tetraeder Map schneller als das direkte Rendern in die Textur mit Hilfe eines P-Buffers.

Spherical Map

Liegt die Rundumsicht schon in Form einer einzelnen Textur oder als Folge von Texturen vor, so entfällt die Erzeugung der Teilbilder in diesem Schritt. Die jeweils zu verwendende Textur muss dann lediglich auf die Grafikkarte übertragen werden. Aus Aufnahmen gewonnene Rundumsichten liegen oft als Spherical Map vor, deren Abbildung der äquidistanten zylindrischen Projektion einer Kugel entspricht.

Abbildung 5.4 zeigt eine solche Spherical Map für die Position aus Abbildung 2.8. Das

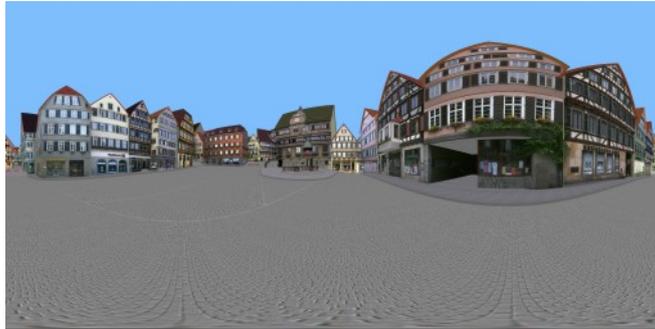


Abbildung 5.4: *Spherical Map für die Rundumsicht aus Abbildung 2.8*

Sampling ist wie bei Tetraeder Map nicht hardwareunterstützt, für die Darstellung wird wieder eine weitere Textur als Lookup Table verwendet.

5.1.2 Geometrieangepassung

Die Geometrieangepassung ist der erste Teilschritt der zweiten Stufe des bildbasierten Verfahrens. In der zweiten Stufe wird einfach ein Rechteck gerendert, das den gesamten Viewport ausfüllt. Die eigentlichen Berechnungen der folgenden Teilschritte erfolgen dann für jeden einzelnen Bildpunkt vollständig im Pixelshader. Die verwendeten Pixelshader sind in Anhang A.4 wiedergegeben.

Durch die Geometrie der Anordnung von Projektor und Projektionsfläche definiert jeder Bildpunkt für den Benutzer nach Gleichung 4.3 eine Richtung im Raum. Aus einer vorberechneten Textur, die diese Richtungen für alle Bildpunkte enthält, wird die Richtung für die folgenden Schritte bestimmt. Die üblicherweise verwendeten Texturen mit 8 bit sind hier nicht verwendbar, da bei diesen nur 256 Stufen für die Darstellungen einer Komponente des Richtungsvektors möglich sind und deutliche Streifen im Bild auftreten. Wahlweise kann entweder ein 16 bit Texturformat wie `GL_LUMINANCE16_ALPHA16` verwendet werden, oder es werden jeweils zwei 8 bit Werte mit entsprechenden Faktoren zu einem 16 bit Wert kombiniert.

Die zusätzlich benötigte Zeit für die Geometrieangepassung beträgt etwa 1.2 ms für eine Bildgröße von 1280×1024 Bildpunkten.

5.1.3 Aberration und Sampling der Rundumsicht

Die durch den Bildpunkt direkt oder aus der Geometrieangepassung bestimmte Richtung wird nun entsprechend der Aberrationsgleichung 2.8 in das System transformiert, in dem die Rundumsicht erstellt wurde. Die dazu notwendigen Informationen über die Be-

wegungsrichtung und Geschwindigkeit werden einmal pro Bild als Parameter für die Pixeleinheit gesetzt. Anschließend wird die berechnete Richtung benutzt, um die entsprechende Farbe oder den Materialparameter aus der Rundumsicht zu bestimmen. Am einfachsten ist dies für die Cubemap, die Information kann mit einem Texturbefehl erhalten werden, wobei die Richtung als Argument nicht normiert sein muss, die Division durch D in 2.8 kann damit entfallen. Wird eine Tetraeder Map oder Spherical Map als Rundumsicht verwendet, so wird eine zusätzliche Cubemap benutzt, die jeder Richtung die entsprechenden Texturkoordinaten in der Rundumsicht zuordnet. Wie bei der Geometrieangepassung muss auch diese Cubemap eine Auflösung von 16 bit besitzen.

Auf Grafikkhardware, die keine Pixelshader unterstützt, kann die Geometrieverzerrung implementiert werden, indem die Rundumsicht auf ein Dreiecksgitter in Form einer Kugel, eines Würfels oder eines Tetraeders gelegt wird. Dieses Gitter wird anschließend für die relativistische Darstellung per Vertex statt per Pixel verzerrt. Je nach Hardware kann diese Verzerrung noch auf der Hardware in einer programmierbaren Vertexeinheit erfolgen, oder auf der CPU.

5.1.4 Doppler- und Searchlighteffekt

Mit dem aus der Rundumsicht bestimmten Farbwert oder Materialindex und dem für den Bildpunkt nach Gleichung 2.7 berechneten Dopplerfaktor erhält man die Leuchtdichte für den bewegten Beobachter aus 2.16. Die einzelnen Werte für die $X_{ij}(D)$ sind dabei in einer Textur als Funktion des Dopplerfaktors für die drei Farbkanäle abgelegt. Enthält die Rundumsicht keine Materialinformation, so können nur drei vom Dopplerfaktor abhängige Farben mit den RGB-Werten gewichtet werden. Die korrekte Farbe lässt sich ohne weitere Informationen nicht bestimmen.

5.1.5 Tone Mapping

Die Berücksichtigung des Searchlighteffekts bei der Leuchtdichteberechnung für einen Bildpunkt führt durch den Faktor D^5 zu einem großen Bereich für die Leuchtdichte bei unterschiedlichen Geschwindigkeiten. Für die Darstellung müssen die Werte letztendlich jedoch auf den Bereich $[0, 1]$ beschränkt werden, entsprechend dem darstellbaren Leuchtdichtebereich des Ausgabegerätes. Ohne weitere Skalierung schneidet die Grafikkhardware Werte größer als eins ab. Dadurch treten für Farben mit einzelnen RGB-Komponenten größer eins Farbfehler auf, und größere Leuchtdichten werden allesamt als Weiß dargestellt.

Durch das Tone Mapping lassen sich diese Artefakte vermindern, und es kann ein größerer berechneter Leuchtdichtebereich mit dem begrenzten Leuchtdichtebereich eines Monitors oder Projektors dargestellt werden.

Eine einfache und schnelle Möglichkeit, den gesamten Bereich von null bis unendlich

auf das Intervall $[0, 1]$ abzubilden, liefert die Funktion

$$L \rightarrow 1 - e^{-kL} \quad (5.1)$$

Für kleine Leuchtdichten ist der Wert etwa proportional zu L , für höhere Leuchtdichten tritt eine Sättigung ähnlich wie bei einer photographischen Schwärzungskurve mit Sättigung auf. Der Parameter k erlaubt eine Steuerung der Helligkeit. Die Funktion lässt sich mit drei Befehlen im Pixelshader implementieren (vgl. Anhang A.4).

Andere Kurven für das Tone Mapping sind analog im Pixelshader realisierbar. Für komplexere Kurven ist es sinnvoll, statt der Berechnung im Pixelshader die vorberechneten Werte als Tabelle in einer eindimensionalen Textur abzulegen und die Berechnung wieder durch ein Texturlookup zu ersetzen. Auch hier muss der Wert für die Leuchtdichte zunächst durch eine einfache Skalierung oder eine Funktion wie 5.1 auf den Bereich $[0, 1]$ abgebildet werden.

Die Textur kann für jedes Bild neu gesetzt werden, so dass mit dieser Technik auch interaktive oder iterative Verfahren wie das aus [26] verwendet werden können. Für ein iteratives Verfahren können einzelne Kontrollpunkte aus dem fertigen Bild dazu verwendet werden, die Tonemappingfunktion für das nächste Bild jeweils neu festzulegen. Weitere Beispiele für Implementierungen des Tone Mappings auf der Grafikhardware sind in [15] zu finden.

Wird für den Projektor oder Monitor eine Gamma-Korrektur gewünscht, so lässt sich diese an die Funktion 5.1 anschließen beziehungsweise in die vorberechneten Werte der Lookuptextur integrieren.

5.1.6 Anpassung der Leuchtdichte und Farbe bei Projektion

Bevor der berechnete Farbwert ausgegeben wird, kann noch eine Anpassung der Leuchtdichte und Farbe erfolgen, so dass bei der Projektion auf gekrümmte Flächen und insbesondere bei der Verwendung mehrerer Projektoren eine gleichmäßige Ausleuchtung der Projektionsfläche erreicht wird. Die Korrekturfaktoren werden entsprechend der Geometrie der Anordnung von Projektoren und Projektionsfläche nach Gleichung 4.5 berechnet und für jeden Projektor in einer Textur abgelegt. Bei der Darstellung wird der ermittelte Farbwert dann für jeden Bildpunkt mit dem Korrekturfaktor aus der Textur multipliziert.

5.1.7 Materialabhängige Spektralverteilungen

Eine korrekte Darstellung des relativistischen Searchlight- und Dopplereffekts ist nur möglich, wenn für alle Körper in der Szene die notwendigen spektralen Informatio-

nen vorliegen. Bei der Erstellung der Rundumsicht werden dann anstelle der RGB-Farbwerte ein Materialindex j und ein Gewichtungsfaktor a_j für das Material gerendert, so dass in den folgenden Schritten die zur Berechnung der Leuchtdichte nach 2.16 benötigte Information bereitsteht.

Die $X_{ij}(D)$ werden beim Programmstart für alle in der Szene vorkommenden spektralen Verteilungen und Dopplerfaktoren berechnet und in einer zweidimensionalen Textur als XYZ-Farbwerte abgelegt. Die erste Texturkoordinate ist durch den Dopplerfaktor bestimmt, der mit der Funktion $1 - 2^{-D}$ auf den Bereich $[0, 1]$ abgebildet wird. Die zweite Koordinate wird durch das Material bestimmt. Für jedes Material wird ein Streifen reserviert, der in der Textur vier bis acht Pixel umfasst, um das Auftreten von Interpolationsfehlern zu vermeiden. Entspricht der Materialindex einem kontinuierlichen physikalischen Parameter, wie beispielsweise der Temperatur eines schwarzen Körpers oder einer dominanten Wellenlänge, so lassen sich die Farbwerte für die verschiedenen Materialien dichter in die Textur packen.

In der zweiten Stufe wird nun für jeden Bildpunkt die entsprechende Richtung \mathbf{n} berechnet und damit der Materialindex und der Faktor a_j aus der Rundumsicht bestimmt. Zusammen mit dem Dopplerfaktor lassen die Farbwerte aus der Textur $X_{ij}(D)$ ermitteln, diese werden mit dem Faktor a_j und dem Searchlightfaktor D^5 multipliziert, um die Leuchtdichte zu erhalten. Anschließend wird die Leuchtdichte mit Hilfe des Tone Mapping wieder auf den darstellbaren Bereich von null bis eins transformiert. Der gesamte Ablauf als Pixelshader ist in Anhang A.4 zu finden.

5.2 Stereoskopische Darstellung

Bei der stereoskopischen Visualisierung muss die relativistische Sicht für jedes Auge mit dem oben beschriebenen Verfahren einzeln erstellt werden. Die beiden Positionen, an denen die Rundumsichten erzeugt werden, sollen für den virtuellen Beobachter gleichzeitig sein. Im Ruhesystem der Objekte sind die Ereignisse dadurch zeitlich versetzt und ihr Abstand ist größer als der Augenabstand.

Die Auswahl des jeweiligen Bildausschnittes für die beiden Augen kann einfach in die Verzerrungskorrektur integriert werden. Setzt man die Position der Augen in die Korrektur nach Gleichung 4.4 ein, so werden die Bildausschnitte automatisch richtig gewählt.

Neben der Bildkanaltrennung über die Polarisation (vgl. Abbildung 4.8) und mit Shutterbrille wurde auch die Bildkanaltrennung mit Hilfe der Anaglyph-Brille implementiert. Dabei werden zuerst die Rundumsichten für beide Augen wie beschrieben erzeugt. Die Kombination zu einem Bild erfolgt dann jedoch für jeden Bildpunkt in einem modifizierten Pixelshader. Der Shader führt die einzelnen Schritte aus Abbildung 5.1 für beide Rundumsichten aus. Die Mischung der Farbwerte aus den beiden Rundumsichten

erfolgt im letzten Schritt zusammen mit der Farbkorrektur. Gegenüber der Technik, die Bilder für die beiden Augen zunächst vollständig zu erzeugen, als Textur zwischenspeichern und erst in einem dritten Schritt wieder zu kombinieren, ist die Kombination im Pixelshader wesentlich schneller. Abbildungen 3.15 und 5.12 zeigen mit dieser Technik erzeugte Bilder.

5.3 Bildqualität

In diesem Abschnitt soll die Bildqualität des schnellen bildbasierten Verfahrens mit der Bildqualität eines vergleichbar schnellen geometriebasierten Verfahrens verglichen werden, und es wird beschrieben, wie sich die Bildqualität des Verfahrens durch Einbindung von Antialiasing und Mipmapping verbessern lässt.

Vergleich mit dem geometriebasierten Verfahren

Bei den geometriebasierten Verfahren wird die relativistische Verzerrung für jeden Vertex berechnet, für alle anderen Punkte des Polygons wird von der Hardware linear interpoliert. Beim bildbasierten Verfahren hingegen wird die Verzerrung wie oben beschrieben pro Bildpunkt berechnet. Es eignet sich daher besonders für Szenen, die mit geringer Polygonanzahl modelliert sind, und bei denen sich der Beobachter im Verhältnis zur Ausdehnung einzelner Polygone nah an den Körpern bewegt. In diesem Fall erstreckt sich ein Polygon über einen größeren Bereich im Bild, so dass sich die Nichtlinearität der relativistischen Verzerrung besonders deutlich bemerkbar macht.

Abbildung 5.5 zeigt die relativistisch verzerrten Ansichten für eine Bewegung mit 80% der Lichtgeschwindigkeit durch *Virtual Tübingen* (5.6.1) für das bildbasierte und das geometriebasierte Verfahren in drei Verfeinerungsstufen. In (b) wurde das unveränderte Gitter benutzt, in (c) und (d) wurde das Gitter jeweils durch Halbierung der Kantenlänge verfeinert. Auch in der zweiten Verfeinerungsstufe mit der 16-fachen Polygonanzahl sind noch deutliche Artefakte zu erkennen, insbesondere am Haus oben links im Bild und am Kopfsteinpflaster. Um eine dem bildbasierten Verfahren ähnliche Qualität zu erreichen, muss das Gitter zumindest lokal adaptiv so verfeinert werden, dass sich eine um den Faktor 100 höhere Polygonanzahl ergibt. Dabei muss das gesamte Polygon feiner unterteilt werden, eine feinere Unterteilung nur am Rand und an Kanten zu anderen Objekten ist für texturierte Polygone nicht ausreichend. Enthält die Textur gerade Linien, so wird die gröbere Unterteilung in Form von Knicken in den Linien sichtbar. In Abbildung 5.5 ist dieser Effekt deutlich am Kopfsteinpflaster zu sehen. Stärker noch als bei einzelnen Bildern macht sich der Effekt bei Bewegung durch die Szene bemerkbar, hier tritt insbesondere bei Richtungsänderungen ein störendes ‘Fließen’ der Texturen auf den Körpern auf.

Durch die notwendige hohe lokale Verfeinerung beim geometriebasierten Verfahren ist

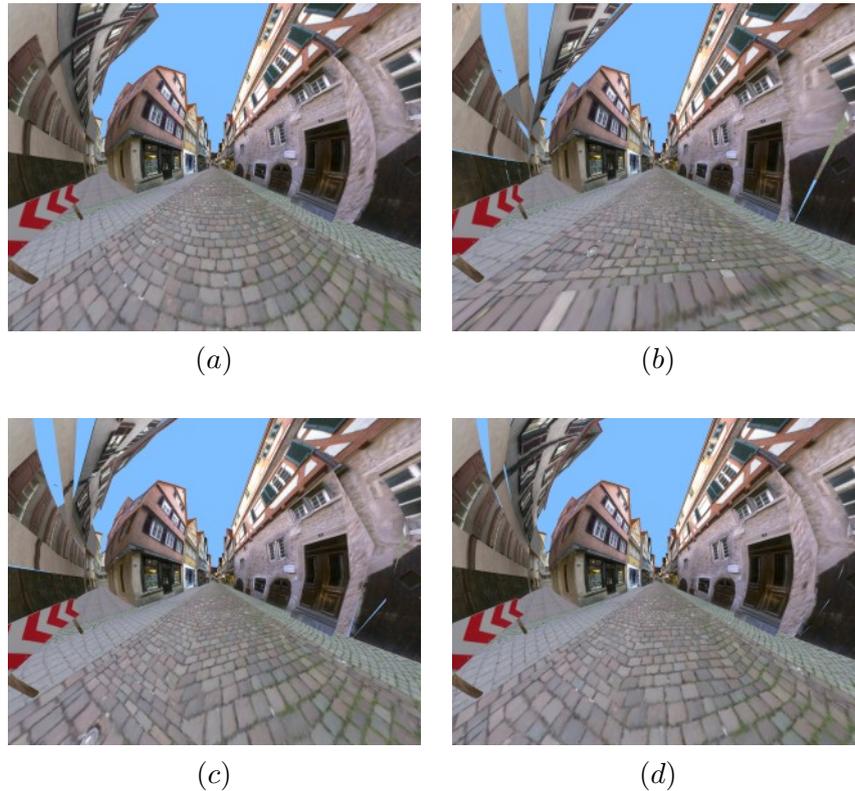


Abbildung 5.5: *Relativistische Verzerrung beim bildbasierten Verfahren (a) und beim geometriebasierten Verfahren (b-d). In (c) und (d) wurde das Gittermodell jeweils um den Faktor 2 verfeinert.*

das bildbasierte Verfahren auf aktueller Grafikhardware bei gleicher Bildqualität überlegen.

Antialiasing und Mipmapping

Die Qualität des bildbasierten Verfahrens wird im Wesentlichen von zwei Faktoren bestimmt: erstens von der Qualität der Rundumsicht und zweitens von der Qualität der Umrechnung in der zweiten Stufe.

Die Bildqualität in der klassischen Renderstufe bei der Erzeugung der Rundumsicht kann durch Antialiasing und anisotrope Texturfilterung verbessert werden. Beide Verfahren werden von der Grafikhardware unterstützt und können mittels Treibereinstellungen verändert werden.

Durch die relativistische Verzerrung werden Teile der Rundumsicht im finalen Bild stark vergrößert beziehungsweise stark verkleinert dargestellt. Abbildung 5.6 stellt das Verhältnis von Bildpunkten in der Rundumsicht zu den Bildpunkten im fertigen Bild in

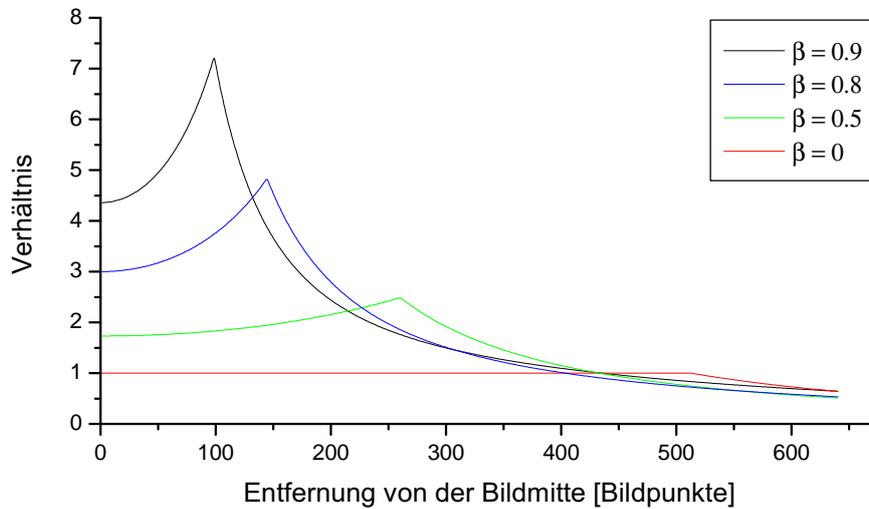


Abbildung 5.6: Verhältnis von Bildpunkten in der Rundumsicht zu dargestellten Bildpunkten bei verschiedenen Geschwindigkeiten. Rundumsicht: Cubemap 1024×1024 , Bildschirm: Auflösung 1280×1024

Abhängigkeit vom Winkel zur Bewegungsrichtung für verschiedene Geschwindigkeiten dar. Bei höheren Geschwindigkeiten sind insbesondere in Bewegungsrichtung Artefakte durch Abtastfehler zu erwarten.

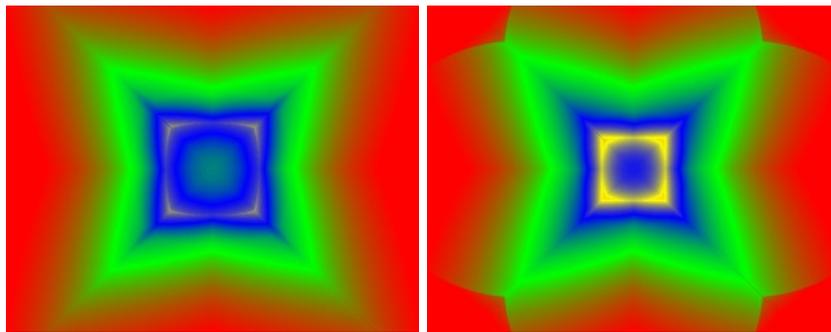


Abbildung 5.7: Mipmabebenen bei Verwendung einer Cubemap für $\beta = 0.8$ und $\beta = 0.9$. Die Gewichtung der einzelnen Mipmabebenen ist folgendermaßen kodiert: Ebene 0 rot, 1 grün, 2 blau, 3 gelb

Diese Abtastfehler lassen sich durch Mipmapping [66] verringern. Je nach erforderlicher Verkleinerung wird dabei zwischen verschiedenen, vorgefilterten Texturen der Rundumsicht umgeschaltet. Die Auswahl der erforderlichen Mipmabebene erfolgt automatisch in der Hardware. Bei der interaktiven Visualisierung müssen die verschiedenen Mipmabebenen während der Laufzeit erzeugt werden, was mit der OpenGL Exten-

sion `GL_SGIS_GENERATE_MIPMAP` auf der Grafikhardware erfolgen kann. Abbildung 5.7 stellt die verschiedenen Mipmabeenen für die Cubemap bei einer Geschwindigkeit von $\beta = 0.5$ farbkodiert dar.

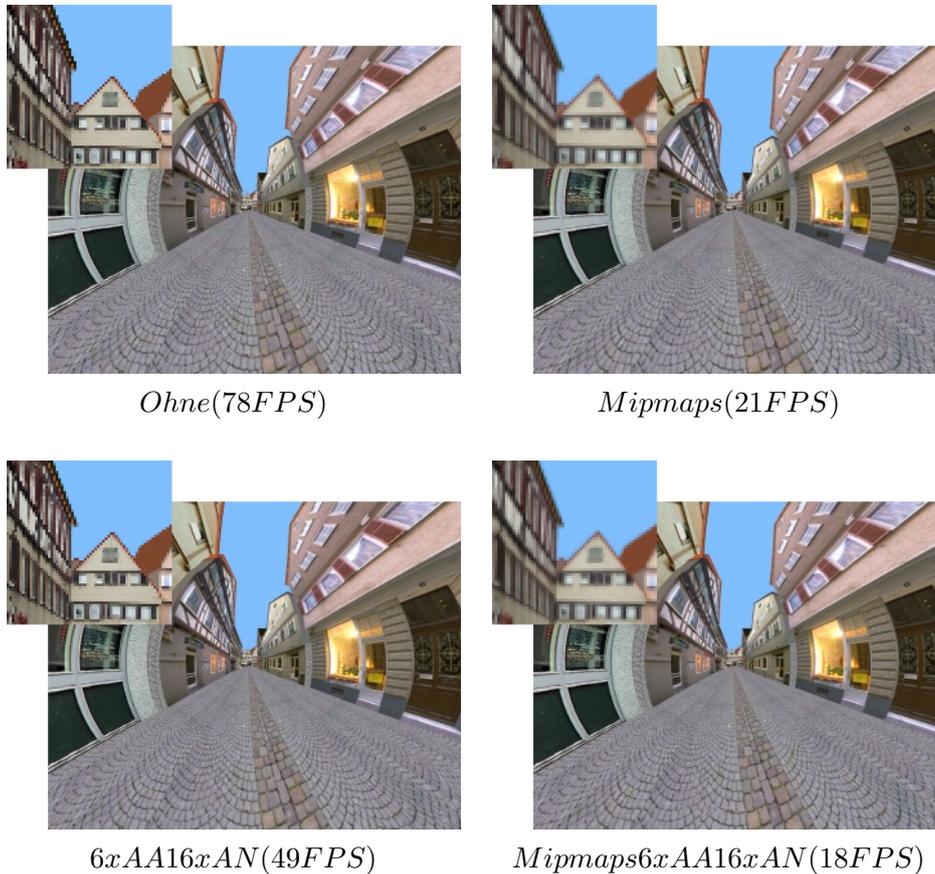


Abbildung 5.8: *Verschiedene Einstellungen für Mipmapping, Antialiasing und Anisotrope Filterung, in allen Fällen ist $\beta = 0.8$ (6xAA: 6-faches Antialiasing, 16xAN 16-fache anisotrope Filterung, in Klammern die Bildrate für die ATI Radeon 9800XT)*

Abbildung 5.8 zeigt den unterschiedlichen Einfluss der Qualitätseinstellungen bei der Erzeugung der Rundumsicht und bei der relativistischen Umrechnung auf das finale Bild. Die Verwendung des Mipmapping verbessert die Bildqualität wesentlich, was sich vor allem bei Bewegung durch die virtuelle Welt bemerkbar macht. Die Abtastfehler an den Häuserfronten und das Moirémuster auf dem Kopfsteinpflaster werden durch das Mipmapping vollständig unterdrückt. Die Verwendung von Antialiasing und anisotroper Filterung bei der Erzeugung der Rundumsicht hat bei höheren Geschwindigkeiten keinen merklichen Einfluss, da die Kantenglättung und die bessere Texturfilterung durch die Verkleinerung bei der relativistischen Verzerrung von den Abtastfehlern beziehungsweise von der Filterung der Textur bei der Erstellung der Mipmabeenen nicht zum Tragen kommen.

Die bessere Qualität bei der Verwendung von Mipmaps wird durch eine wesentlich verlängerte Renderzeit erkauft, was durch die in Abbildung 5.8 angegebenen Bildraten deutlich wird. Die Situation kann verbessert werden, indem für jede Seite der Cubemap oder Tetraeder Map nur die jeweils benötigten Mipmapebenen erzeugt werden. Dabei kann die Gesamtperformance sogar zunehmen, vergleiche Abschnitt 5.5.

Anstelle des Mipmappings kann die Bildqualität bei der relativistischen Umrechnung auch durch Überabtastung verbessert werden. Die Farbe eines Bildpunktes wird durch mehrfaches Sampling der Rundumsicht im Pixelshader ermittelt. Eine vierfache Überabtastung kann die Bildqualität ohne großen Performanceverlust im unteren Geschwindigkeitsbereich verbessern. Bei höheren Geschwindigkeiten und der damit verbundenen starken Verkleinerung der Rundumsichtstextur in Vorwärtsrichtung treten auch bei Überabtastung wieder die Abtastfehler auf. In diesem Bereich liefert auch die sechs- bis achtfache Überabtastung nicht die mit dem Mipmapping erreichbare Bildqualität, so dass dem optimierten Mipmapping (5.5) mit variablen Texturgrößen der einzelnen Seiten der Cubemap beziehungsweise Tetraeder Map insgesamt der Vorzug zu geben ist.

5.4 Geschwindigkeit

Das beschriebene bildbasierte Verfahren wurde auf einem Standard-PC mit einem Intel 845P Chipsatz und einem Pentium 4 Prozessor bei 3.0 GHz getestet. Als Grafikkhardware standen Karten mit den Chipsätzen ATI Radeon 9800XT sowie nVidia GeForce FX5950 Ultra zur Verfügung. Beide Grafikkarten verfügen über 256MB RAM und 8 Pixeleinheiten für die parallele Ausführung von Pixelshadern.

	9800XT	9800XT	FX5950U	FX5950U
Bildspeicher löschen	2.3	2.3	1.6	1.6
Rendern	11.0	6.1	6.2	5.8
Kopieren	2.5	2.5	6.6	6.6
Pixelshader	2.0	2.0	5.2	5.2
Gesamt	17.8	12.9	19.6	19.2
Bildrate	56.1	77.3	51.0	51.9

Tabelle 5.1: *Benötigte Zeiten für die einzelnen Stufen des bildbasierten Verfahrens in ms und Bildrate in FPS für zwei verschiedene Position in der Umgebung. (9800XT: ATI Radeon 9800XT / FX5950U: nVidia GeForce FX 5950 Ultra)*

Die Ergebnisse für die Szenerie *Virtual Tübingen* (5.6.1) bei einer Auflösung von 1280×1024 Bildpunkten, einer Cubemap mit der Auflösung von 1024×1024 Punkten und dem einfachsten Pixelshader ohne Geometriekorrektur und Doppler- bzw. Tone-Mapping seien exemplarisch dargestellt. Tabelle 5.1 zeigt die durchschnittlich benötigten Zeiten

für die einzelnen Stufen des Rendervorgangs für die beiden Grafikkarten und jeweils zwei verschiedene Position in der Szenerie. Der Unterschied zwischen den Architekturen der beiden Grafikkarten ist deutlich zu erkennen, die benötigten Zeiten für die einzelnen Teilschritte unterscheiden sich zum Teil um mehr als den Faktor zwei. Die Zeiten für das Löschen des Bildspeichers, das Kopieren aus dem Bildspeicher und die Ausführung des Pixelshaders sind für eine Grafikkarte jeweils als konstant anzusehen. Wird das bildbasierte Verfahren als Aufsatz auf eine klassische Renderengine verwendet, so lassen sich mit den oben angegebenen Werten die erreichbaren Bildraten in Abhängigkeit von der Bildrate der Renderengine abschätzen.

Bildrate klassisch [FPS]	100	200	500	1000
Relativistisch (1)	15	30	60	95
Relativistisch (2)	10	20	30	40

Tabelle 5.2: *Erreichbare Bildraten beim bildbasierten Verfahren mit relativistischer Verzerrung allein (1) sowie Geometriekorrektur, Dopplereffekt, Searchlighteffekt und Tone Mapping (2) in Abhängigkeit von der Bildrate, mit der die Szenerie klassisch gerendert werden kann. (ATI Radeon 9800XT, 1280×1024 Bildpunkte, Cubemap 1024×1024)*

Tabelle 5.2 zeigt deutlich, wie stark die Leistung der klassischen Renderengine bei der Erstellung der Rundumsicht die Gesamtleistung beeinflusst.

Die Zeit für die Ausführung des Pixelshaders steigt mit der Komplexität der Berechnungen auf maximal 16 ms für der Shader mit Geometriekorrektur, materialabhängigen Spektralverteilungen, Dopplereffekt, Searchlighteffekt und Tone Mapping an, womit noch Bildraten von über 30 FPS erreichbar sind.

Die Leistungen der Workstation-Grafikkarten mit den Chipsätzen nVidia Quadro FX3000 und 3dLabs Wildcat liegen bei der Szenerie *Virtual Tübingen* im gleichen Bereich (50-60 FPS) wie für die oben angegebenen Grafikkarten, wobei die Quadro FX 3000 erst nach einer Reduzierung der Cubemapgröße auf 512×512 Bildraten über 50 FPS erreicht.

Je nach Organisation des Dreiecksgitters und der Texturkomplexität der Szene lassen sich auch wesentlich mehr Polygone mit Bildraten von über 50 FPS darstellen, die Grenze liegt etwa bei 400000 Dreiecken.

Der Vollständigkeit halber seien an dieser Stelle noch die Werte für die geometriebasierte Verzerrung eines Dreiecksgitters angegeben. Die relativistische Umrechnung erfolgt wie beim bildbasierten Verfahren vollständig in der Grafikhardware, in diesem Fall jedoch für jeden Vertex im Vertexshader mit der `GL_VERTEX_PROGRAM` Erweiterung. Die Umrechnung von 20000 Polygonen beansprucht etwa 1 ms und ist damit um Faktor 20 schneller als die Umrechnung in der CPU mit anschließendem Transfer auf die Grafikkarte. Damit lassen sich bei *Virtual Tübingen* in der zweiten Verfeinerungsstufe nach

Abbildung 5.5 Bildraten von 60-70 FPS erreichen. Anders organisierte 3D-Modelle, wie beispielsweise das Gitter aus Abbildung 3.3, bei denen längere Triangle Strips verwendet werden können, lassen sich auch bei etwa 400000 Polygonen noch mit 50-60 FPS geometriebasiert darstellen.

5.5 Optimierungen

Die in Abschnitt 5.4 angegebenen Zeiten für die einzelnen Schritte der Bilderzeugung zeigen deutlich, dass die Erstellung der Rundumsicht einen großen Teil der gesamten Zeit beansprucht. Je nach verwendeter Technik müssen vier oder sechs Bilder gerendert werden. Der erste Ansatzpunkt ist daher, die Erzeugung dieser einzelnen Bilder so schnell wie möglich zu machen. Der Rendervorgang dieser Bilder ist völlig unabhängig von der relativistischen Stufe, es können daher alle üblichen Techniken zur Beschleunigung verwendet werden, von denen die wichtigsten kurz erwähnt werden sollen. Die gesamten Geometrie- und Texturdaten werden beim Programmstart auf die Grafikkarte übertragen (in Form von OpenGL *Display Lists* und *Triangle Strips*). Während der Laufzeit wird durch Frustum Culling und je nach 3D-Modell auch durch Occlusion Culling die Anzahl der darzustellenden Dreiecke und die Füllrate verringert.

Im Zusammenspiel mit der relativistischen Darstellung werden zusätzliche Clip Planes eingesetzt, die in Abhängigkeit von der simulierten Geschwindigkeit die Bildbereiche so begrenzen, dass nur derjenige Teil der Rundumsicht gerendert wird, der auch später bei der Verzerrung benötigt wird. Bei niedrigen Geschwindigkeiten kann die benötigte Füllrate so auf weniger als die Hälfte verringert werden. Wird eine Cubemap für die Rundumsicht benutzt, so ist es ausreichend, für Geschwindigkeiten kleiner als $\beta \approx 0.86$ nur fünf statt der sechs Bilder zu rendern.

Das Kopieren vom Bildspeicher in die Textur benötigt ebenfalls einen nicht zu vernachlässigenden Teil der gesamten Zeit. Die verwendeten Grafikkarten unterstützen die WGL_ARB_RENDER_TEXTURE Erweiterung, die das direkte Rendern in eine Textur ermöglicht. Mit den aktuellen Treiberversionen ergibt sich allerdings kein Geschwindigkeitsvorteil. Der Kontextwechsel beim Umschalten zwischen dem Bildspeicher und den Texturen als Ziel des Rendervorgangs benötigt die gleiche Zeit wie das Kopieren aus dem Bildspeicher in die Textur.

Bei der Diskussion der Bildqualität ist gezeigt worden, dass die Textur für die Rundumsicht mit zunehmender Geschwindigkeit im Vorwärtsrichtung immer stärker verkleinert wird, vergleiche Abbildungen 5.6 und 5.7. Wird die Textur mit gleichbleibender Auflösung erzeugt, so treten Abtastfehler auf. Die Situation kann verbessert werden, indem die einzelnen Texturen der Cubemap oder Tetraeder Map jeweils nur in der maximal benötigten Größe gerendert werden. Wird dabei kein Mipmapping verwendet, so werden die Abtastfehler durch die kleinere Texturauflösung verringert. Andernfalls werden nur die benötigten Mipmapebenen erzeugt, bei der Verwendung der automati-

sehen Mipmapgenerierung auch alle kleineren Ebenen. In allen Fällen ist die benötigte Füllrate geringer als bei der Verwendung von festen Texturauflösungen. Zusammen mit dem oben genannten Einsatz von Clip Planes kann die Füllrate bei der Erstellung der Rundumsicht so über den gesamten Geschwindigkeitsbereich annähernd konstant gehalten werden, wodurch eine etwa gleichbleibende Bildrate während der Simulation erreicht wird.

Parallelisierbarkeit

Die Berechnung der sechs Seiten der Cubemap drängt sich für eine Parallelisierung geradezu auf. Problematisch ist hier jedoch die Integration der einzelnen Texturen zu einer Cubemap, wenn die Seiten auf verschiedenen GPUs erzeugt werden. Die GPU, auf der die Cubemap für den letzten Rendschritt benötigt wird, muss bei einer Texturgröße von 1024×1024 , 24 bit Farbtiefe und einer Bildrate von 60 Hz in der Lage sein, die Texturen mit einer Datenrate von 1.1 GB/s zu empfangen. Die Übertragung der gerenderten Texturen erweist sich damit bei dieser Konfiguration als Flaschenhals. Durch Kaskadierung mehrerer Rechner oder GPUs kann die nötige Bandbreite auf etwa 190 MB/s verringert werden.

Alternativ kann die relativistische Verzerrung auch für jedes Teilbild der Rundumsicht zunächst einzeln auf der jeweiligen GPU erfolgen. Im letzten Schritt werden die so erstellten Bildausschnitte dann zu einem Gesamtbild zusammengefügt. Entsprechende Hardware, die die Videosignale (DVI) von bis zu acht Quellen zu einem Bild zusammenfassen und als ein Videosignal ausgeben kann, ist als Prototyp vorhanden [39].

5.6 Beispiele

Der letzte Abschnitt dieses Kapitels gibt einen Überblick und technische Details zu den im Rahmen dieser Arbeit entstandenen Visualisierungen und Simulationen.

5.6.1 Virtual Tübingen

In Zusammenarbeit mit dem Max-Planck-Institut für biologische Kybernetik ist die Simulation einer Fahrradfahrt auf einem Rundkurs in der Tübinger Altstadt entstanden. Die Lichtgeschwindigkeit ist, wie bei Gamows *City Speed Limit*[18], in der Simulation auf 30 km/h herabgesetzt, so dass die relativistischen Effekte für den Benutzer schon bei den mit dem Fahrrad erreichbaren Geschwindigkeiten sichtbar werden.

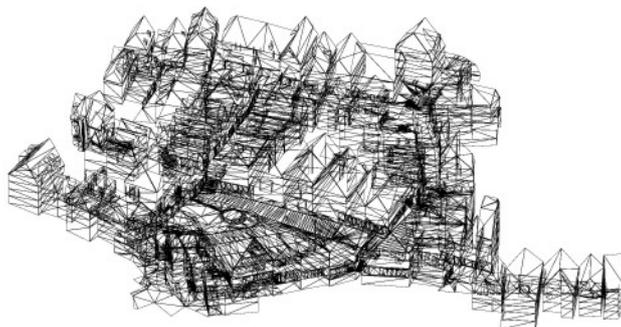


Abbildung 5.9: *Virtual Tübingen: Gittermodell*

Als 3D-Modell sollte ein am Max-Planck-Institut für biologische Kybernetik erstelltes Modell der Tübinger Innenstadt[58] verwendet werden, wobei der Rundkurs nur einen Ausschnitt des gesamten Modells darstellt. Abbildung 5.9 zeigt das Gittermodell des verwendeten Ausschnitts mit etwa 20000 Polygonen und 200 MB Texturdaten. Das Modell ist – für die klassische Darstellung optimiert – grob trianguliert, einzelne Dreiecke erstrecken sich in den meisten Fällen über die gesamte Breite eines Hauses oder der Straße. Der Beobachter befindet sich deshalb oft in einer Entfernung von weniger als einer Kantenlänge zu einzelnen Dreiecken, was die relativistische Verzerrung mit einem geometriebasierten Verfahren auch bei lokaler Verfeinerung (vgl. Abbildung 5.5) verhindert, daher wurde für die Realisierung das vorgestellte bildbasierte Verfahren entwickelt.

Die einzelnen Häuser oder Gebäudegruppen werden beim Programmstart als OpenGL Display Lists auf der Grafikkarte abgelegt und können bei der Erstellung der Ein-



Abbildung 5.10: *Virtual Tübingen: Interaktiv dargestellter Ausschnitt aus dem größeren Modell* [58]

zelbilder der Rundumsicht schnell ein- und ausgeschaltet werden. Die Ausblendung erfolgt durch Frustum Culling mit der Bounding Box des Gebäudes und durch einfache Sichtbarkeitsregeln in Abhängigkeit von der Beobachterposition. Eine aufwendigere Unterteilung mit hierarchischen Strukturen zur Sichtbarkeitsbestimmung bringt bei dem dargestellten Ausschnitt des Modells keine Geschwindigkeitsvorteile.

Als Rundumsicht wird eine 1024×1024 Cubemap verwendet, die Auflösung des relativistisch umgerechneten Bildes ist 1280×1024 .

Als Rechner kommen wahlweise ein System mit einem Intel Pentium 4 bei 3.06 GHz und 2GB RAM beziehungsweise ein AMD Athlon 64 3200+ bei 2.0 GHz mit 1 GB RAM zum Einsatz, die Grafikkhardware ist in beiden Fällen eine Karte mit der ATI Radeon 9800XT GPU.

Die Steuerung erfolgt mit Hilfe eines vom Max-Planck-Institut umgebauten Fahrrades, das dort ebenfalls für psychophysikalische Experimente entwickelt wurde. Der Benutzer kann die Geschwindigkeit durch die Trittfrequenz vorgeben, sowie lenken und eine zweistufige Bremse betätigen. Die Bremsleistung der Wirbelstrombremse wird aus dem Simulationszustand und aus der Stellung des Bremshebels vom Rechner ermittelt. Die Kommunikation mit dem Rechner erfolgt über eine USB-Verbindung. Um den Eindruck der Zeitdilatation beziehungsweise der Längenkontraktion zu vermitteln, wird die Bewegung so integriert, dass die simulierte Eigenzeit mit der realen Zeit übereinstimmt.

Abbildung 5.13 zeigt den Aufbau mit Fahrrad und gekrümmter Projektionsfläche. Der Projektor mit 3000 ANSI Lumen befindet sich einer Höhe von etwa 5 m knapp hinter dem Fahrrad und kann mit seiner 1:1 Optik so die gesamte Projektionsfläche von 4.5×3 m ausleuchten. Durch die gewählte Anordnung wird ein großes Gesichtsfeld für

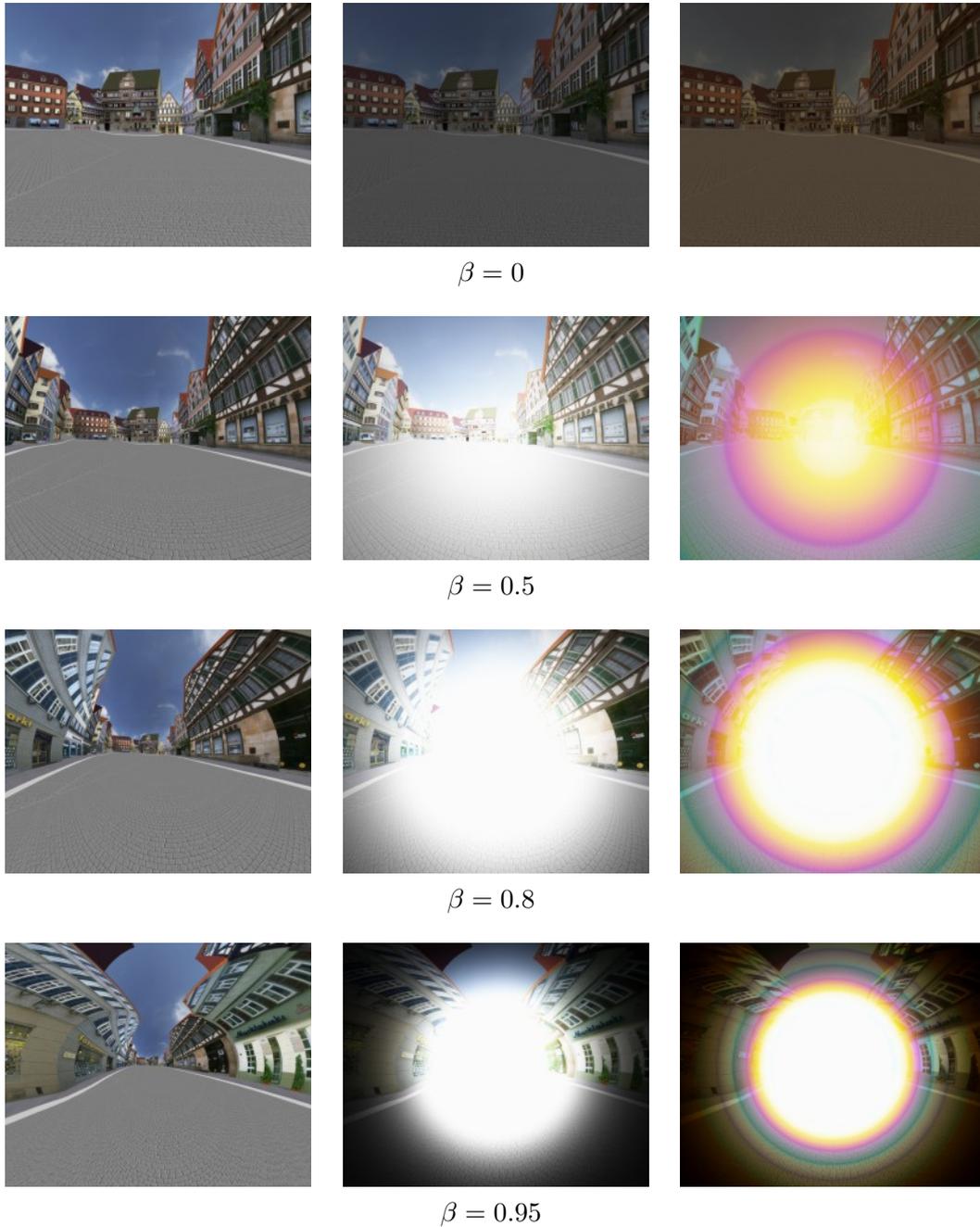


Abbildung 5.11: *Virtual Tübingen: Screenshots. Linke Spalte: Geometrieverzerrung Mitte: mit Searchlighteffekt Rechte Spalte: mit Doppler- und Searchlighteffekt. Die Farben und Helligkeiten können wegen der fehlenden spektralen Information nur näherungsweise wiedergegeben werden. In der Mitte und rechts wird die Helligkeit beim Tone Mapping etwa um den Faktor 2 verringert. Zur Darstellung der durch den Dopplereffekt verschobenen Farben vergleiche Abschnitt 2.4.*

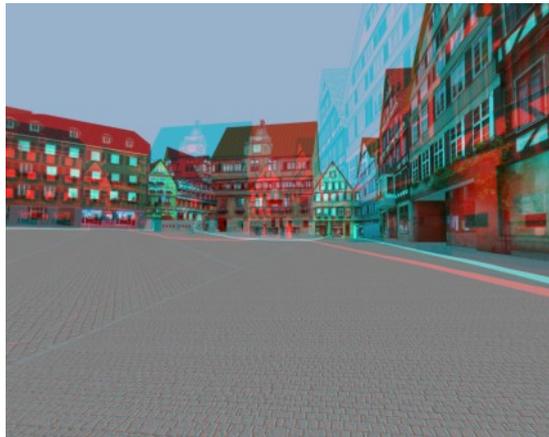
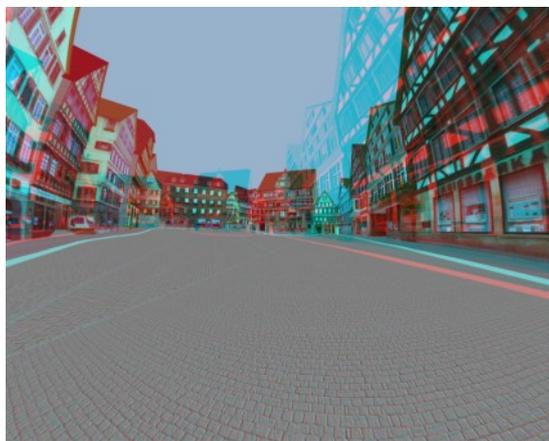
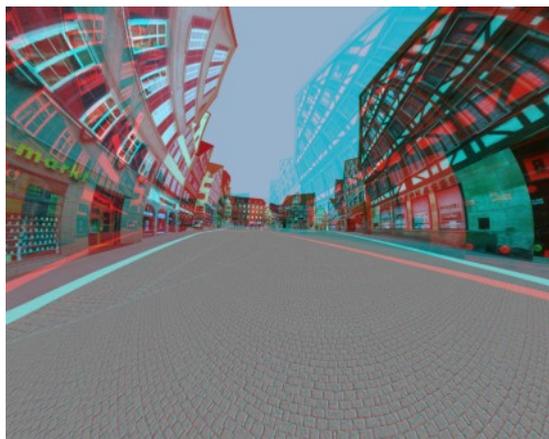
 $\beta = 0$  $\beta = 0.5$  $\beta = 0.8$

Abbildung 5.12: *Virtual Tübingen* bei 3D-Stereo-Darstellung für verschiedene Geschwindigkeiten. Die gesehene Streckung in Bewegungsrichtung und die Krümmung der Häuserfronten wird auch stereoskopisch als Streckung und Krümmung im dreidimensionalen Raum wahrgenommen.

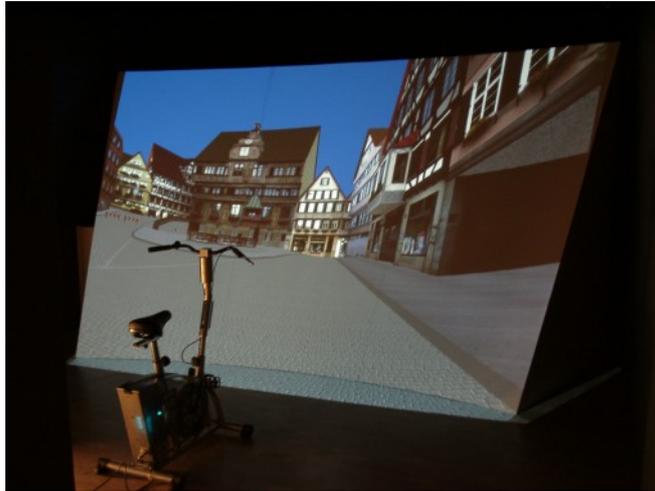


Abbildung 5.13: *Virtual Tübingen: Aufbau mit gekrümmter Projektionsfläche und Fahrrad*

den Benutzer möglich, ohne dass er selbst einen Schatten auf die Projektionsfläche wirft. Die implementierte Geometrieverzerrung für die Projektion konnte für die spezielle Anordnung zur Performancesteigerung abgeschaltet werden, da nur ein Projektor verwendet wird, und da im Bereich des Knicks in der Projektionsfläche am Boden die Projektionsrichtung und Blickrichtung des Benutzers fast übereinstimmen, somit wären auch in diesem Bereich nur minimale Korrekturen erforderlich.

5.6.2 Relativistischer Flugsimulator

Der *relativistische Flugsimulator* ist ein Beispiel für die Verwendung des bildbasierten Verfahrens als Aufsatz für eine bestehende Software, einen Flugsimulator für Modellflugzeuge [13]. Das Programm erlaubt es dem Benutzer, die Bewegungen eines Modellflugzeuges mit Hilfe eines 4-Kanal-Joysticks zu steuern und unter anderem aus der Sicht des Modellflugzeuges zu verfolgen. Zusammen mit der relativistischen Verzerrung können so die auftretenden visuellen Effekte bei dreidimensionalen Bewegungen verdeutlicht werden.

Das Flugzeug wird als Mehrkörpersystem simuliert, wobei der Rumpf, die Tragflächen, Ruderflächen und weitere Bauelemente als starre Körper modelliert sind. Ein Modell besteht aus etwa 20-30 Körpern, die untereinander durch Feder-Dämpfer-Gelenke verbunden sind. Für jeden Körper werden die Kräfte und Momente aus Kollisionen mit der Umgebung, aus den Gelenken sowie aerodynamische Kräfte und Momente berücksichtigt. Die aerodynamischen Kräfte ergeben sich aus der Anströmung und vereinfachten Modellen für die Winkelabhängigkeit der Auftriebs-, Widerstands- und Momentenbeiwerte des Körpers. Die Integration der Bewegungsgleichungen erfolgt mit einem expliziten Verfahren.

Die Simulation der Flugphysik wurde unverändert beibehalten, so dass die Bewegungen den normalen Flugbewegungen eines Modellflugzeuges in einer Atmosphäre entsprechen. Die simulierte Geschwindigkeit wird herangezogen, um einem Wert für β und damit für die Größe der relativistischen Verzerrungen zu bestimmen. Eine Fluggeschwindigkeit von etwa 180 km/h entspricht dabei $\beta = 0.9$, für höhere Geschwindigkeiten geht β gegen eins. Wie beim vorangegangenen Beispiel wird die daraus resultierende Eigenzeit als Zeitbasis für die Simulation verwendet, um die Zeitdilatation beziehungsweise Lorentzkontraktion zu verdeutlichen.

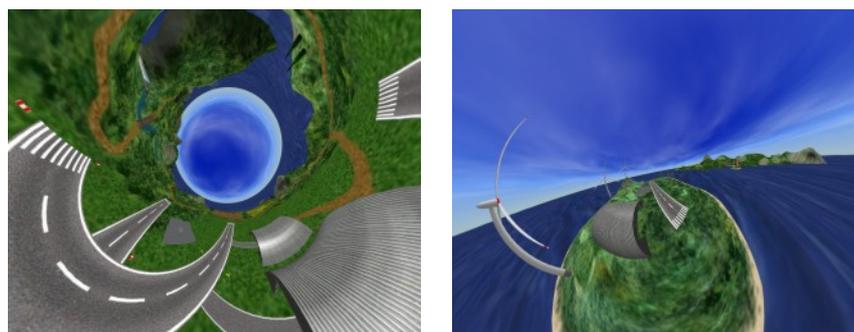


Abbildung 5.14: *Relativistischer Flugsimulator: Screenshots Insel*

Die Renderstufe des Simulators wird zur Erzeugung der einzelnen Seiten der Rundumsicht in Form einer Tetraeder Map verwendet. Neben der jeweils anderen Projektion für die vier Seiten musste nur die Orientierung der Billboards geändert werden, so dass diese unabhängig von der Blickrichtung ist. Für die Darstellung wird die Rundumsicht wieder entsprechend relativistisch verzerrt gerendert.



Abbildung 5.15: *Relativistischer Flugsimulator: Screenshots Mars*

Eine hypothetische Marsstation stellt eine alternative Szenerie mit 80000 Polygonen und Multitexturing dar. Hier kann der Benutzer die Flugrichtung und Geschwindigkeit frei vorgeben und sich in einem beschränkten Gebiet um die Station bewegen.

5.6.3 Interaktive Verzerrung von Panoramaaufnahmen

Zur Demonstration der relativistischen Verzerrung kann diese auch auf Panoramaphotos oder vorher erstellte statische Rundumsichten angewendet werden, der Benutzer kann dann lediglich die Geschwindigkeit als freien Parameter variieren. Die Rundumsicht kann entweder als Cubemap oder als Spherical Map vorliegen. Im zweiten Fall wird die Position in der Spherical Map für jeden Bildpunkt wieder wie bei der Tetraeder Map durch eine Lookuptextur bestimmt.

Die Panoramen können aus einzelnen Aufnahmen mit einer Software wie Panorama Tools [11] zusammengesetzt werden. Da jedoch meistens keine weitergehende spektrale Information vorliegt, kann nur die relativistische Verzerrung korrekt wiedergegeben werden.

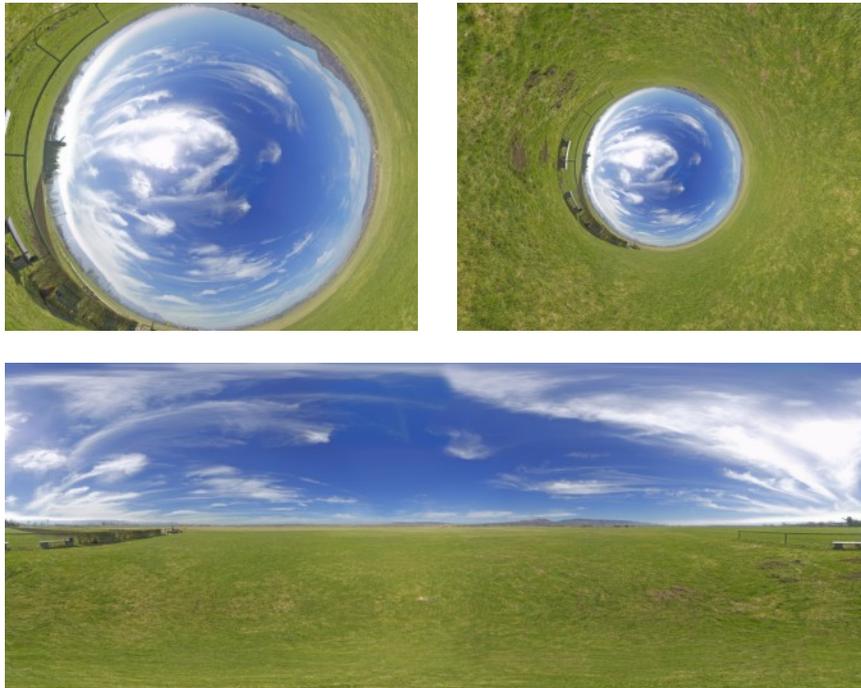


Abbildung 5.16: *Interaktive Verzerrung von Panoramaaufnahmen: Flug senkrecht nach oben mit 70% und 90% der Lichtgeschwindigkeit und die zugrundeliegende Panoramaaufnahme.*

Die Mipmabeenen werden für das Panoramabild beim Programmstart vorberechnet, die erreichbaren Bildraten mit Mipmapping für die Auflösung 1280×1024 liegen bei über 200 FPS und sind unabhängig von der Auflösung der Cubemap. Die benötigte Zeit für die Umrechnung hängt leicht von der Geschwindigkeit β ab, da je nach Geschwindigkeit eine unterschiedliche Anzahl von einzelnen Texturen und Mipmabeenen der Cubemap abgetastet werden muss.

Statt einzelner Panoramen können prinzipiell auch Panoramavideos beim Abspielen relativistisch verzerrt werden, so dass sich der Benutzer interaktiv entlang des durch die Aufnahmen vorgegebenen Pfades bewegen kann. Je nach Anzahl und Abstand der Aufnahmepositionen können eventuell nur bestimmte Geschwindigkeiten zugelassen werden, wenn zeitliches Aliasing vermieden werden soll.

Pro Bild werden 4.4 ms für die Umrechnung benötigt. Entscheidend für die Geschwindigkeit ist in diesem Fall die Datenrate, mit der die Rundumsicht auf die Grafikkarte übertragen werden kann. Bei einer Cubemap mit einer Seitenlänge von 1024 Texeln und einer Bildfrequenz von 30 Hz werden für unkomprimierte Texturen 566 MB/s benötigt, bei der Verwendung der von der Grafikkarte unterstützten Texturkomprimierung lässt sich dieser Wert auf etwa 150 MB/s verringern. Die benötigten Datenraten lassen sich mit dem AGP bzw. PCIe 16x Bus leicht erreichen. Beim Abspielen von Festplatten müssen für diese Datenraten allerdings spezielle Techniken verwendet werden.

Kapitel 6

Ausblick

In der vorliegenden Arbeit wurde ein verbessertes Verfahren zur bildbasierten relativistischen Visualisierung vorgestellt. Die hohe Geschwindigkeit wird durch die Verwendung der programmierbaren Vertex- und Pixeleinheiten moderner Grafikkhardware erreicht. Erst dadurch lassen sich alle relativistischen Effekte pixelgenau interaktiv berechnen. Die Geometrieverzerrung und Leuchtdichteanpassung bei der Projektion auf gekrümmte Flächen sowie die stereoskopische Darstellung sind nahtlos in das Verfahren integriert.

Um in Zukunft immer komplexere 3D-Modelle mit aufwendigeren Beleuchtungs- und Materialmodellen interaktiv darstellen zu können, kann die gute Parallelisierbarkeit des Verfahrens ausgenutzt werden. Die neue Generation von Grafikkarten mit PCI express Bus ermöglicht prinzipiell den Einsatz mehrerer GPUs in einem Rechner. Entscheidend ist dabei eine hohe Bandbreite bei der Übertragung der Daten zwischen zwei GPUs. Aktuell lassen sich zwei Grafikkarten direkt verbinden, wobei jede GPU eine Hälfte des Bildes rendert[37]. Es bleibt abzuwarten, ob mehr als zwei Grafikkarten verwendet werden können, und ob mit entsprechenden Treibern die GPUs einzeln angesprochen werden können, so dass beispielsweise die einzelnen Teilbilder der Rundumsicht parallel gerendert werden können.

Alternativ können die Teilbilder der Rundumsicht auch auf verschiedenen Rechnern gerendert und dort relativistisch verzerrt werden. In diesem Fall müssen die Teilbilder in Form von digitalen Videosignalen mit spezieller Hardware zusammengesetzt werden. Mit der am Max-Planck-Institut entwickelten Grafikkhardware [39] lassen sich so die auf sechs verschiedenen Rechnern erzeugten verzerrten Teilbilder zu einem Videosignal für einen Projektor zusammenführen.

Neue Displays oder Projektoren mit hohem Dynamikbereich [51] sowie mit direkter Unterstützung der 3D-Stereo-Visualisierung [50, 68] werden eine bessere Darstellung der mit dem vorgestellten Verfahren erzeugten Bilder erlauben. Die zu erwartende Steigerung der Geschwindigkeit und Flexibilität bei der Ausführung der Berechnungen pro

Bildpunkt wird komplexere Modelle für Materialien und das Tone Mapping bei hohem Dynamikbereich ermöglichen. Die Lücke zum Raytracing schließt sich mit zunehmender Programmierbarkeit der Pixeleinheiten. In Zukunft scheint auch hardwarebeschleunigtes vierdimensionales Raytracing oder die Erweiterung auf gekrümmte Raumzeiten möglich. Die Effekte der Lichtablenkung auf die stereoskopische Entfernungswahrnehmung in gekrümmten Raumzeiten werden von Thomas Müller untersucht [36].

Das Hauptaugenmerk der vorliegenden Arbeit liegt auf der Visualisierung der relativistischen Effekte. Die Simulation der relativistischen Bewegung ist einfach durch die Bewegung eines einzelnen Punktes modelliert, wobei die Blick- und Bewegungsrichtung direkt vom Benutzer vorgegeben werden. Hier sind realistischere Simulationen durch eine komplett relativistische Modellierung eines ‘starren’ Körpers als System von gekoppelten Massenpunkten oder mit finiten Elementen denkbar, um dem Benutzer ein besseres Gefühl für beschleunigte Bewegungen zu vermitteln.

Anhang

A.1 Invarianz des Phasenraumvolumens

In diesem Anhang wird gezeigt, dass das Phasenraumvolumen einer definierten Menge von Photonen nicht vom Beobachter abhängt. Dazu seien zwei Beobachter B und B' betrachtet, die sich mit der Relativgeschwindigkeit β zueinander bewegen. Die Koordinaten seien so gewählt, dass die Bewegung entlang der x-Achsen erfolgt.

Die Bewegung eines einzelnen Photons werde durch seinen Impuls, einen Punkt in der Raumzeit und einen Parameter λ von B durch

$$\begin{pmatrix} 0 \\ r_x \\ r_y \\ r_z \end{pmatrix} + \lambda \begin{pmatrix} |p| \\ p_x \\ p_y \\ p_z \end{pmatrix} \quad (\text{A.1})$$

beschrieben. B' beschreibt die Weltlinie entsprechend durch

$$\begin{pmatrix} -\beta\gamma r_x \\ \gamma r_x \\ r_y \\ r_z \end{pmatrix} + \lambda \begin{pmatrix} \gamma(|p| - \beta p_x) \\ \gamma(p_x - \beta|p|) \\ p_y \\ p_z \end{pmatrix} \quad (\text{A.2})$$

Die Beobachter bestimmen das von den Photonen eingenommene räumliche Volumen aus der Position der Photonen für $x^0 = 0$ bzw. für $x^{0'} = 0$. Für B bedeutet dies $\lambda = 0$, für B' ist

$$\lambda = \frac{\beta\gamma r_x}{\gamma(|p| - \beta p_x)} = \beta\gamma|p| D r_x \quad (\text{A.3})$$

zu setzen. Die Positionen sind dann durch

$$\begin{pmatrix} 0 \\ Dr_x \\ r_y + \beta\gamma p_y Dr_x \\ r_z + \beta\gamma p_z Dr_x \end{pmatrix} \quad (\text{A.4})$$

gegeben. Bildet man die Determinanten der Abbildungen von (r_x, r_y, r_z) auf die räumlichen Anteile der Positionen, so ergibt sich

$$V'_r = DV_r \quad (\text{A.5})$$

für die von den beiden Beobachtern bestimmten räumlichen Volumina der Photonen. Analog erhält man mit den Determinanten der Abbildungen von (p_x, p_y, p_z) auf die räumlichen Anteile der Impulse in A.1 und A.2 für das Volumen im Impulsraum

$$V'_p = \gamma(1 - \beta \frac{p_x}{|p|})V_p = \frac{1}{D}V_p \quad (\text{A.6})$$

Das Volumen im Phasenraum $V = V_r V_p = V'_r V'_p$ ist also unabhängig vom Beobachter, ebenso wie die Anzahl der betrachteten Photonen. Damit ist auch die Phasenraumdichte \mathcal{N} unabhängig vom Bezugssystem.

A.2 Projektion auf ebene Flächen

Bei der Projektion auf ebene Flächen kann die Abbildung V aus Gleichung 4.4 zur identischen Abbildung gemacht werden, so dass der zweite Schritt der Geometrieanpassung entfallen kann. Der Projektor darf dazu keine nichtlinearen Verzerrungen aufweisen.

Gesucht ist eine projektive Abbildung, die die vier Eckpunkte des projizierten Bildes auf die Eckpunkte des Viewports abbildet. Das Projektionszentrum ist die aktuelle Benutzerposition.

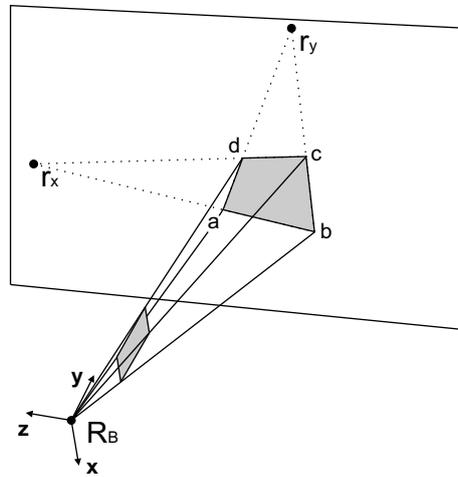


Abbildung A.1: Projektion auf ebene Flächen

Die Geraden durch jeweils zwei benachbarte Eckpunkte des projizierten Bildes schneiden sich in zwei Punkten \mathbf{r}_x und \mathbf{r}_y . Zusammen mit der Benutzerposition \mathbf{R}_B definieren sie die Orientierung für die Projektion (sind die Geraden parallel, so wird der entsprechende Richtungsvektor benutzt):

$$\begin{aligned}\mathbf{x} &= \frac{\mathbf{R}_B - \mathbf{r}_x}{\|\mathbf{R}_B - \mathbf{r}_x\|} \\ \mathbf{z} &= \frac{\mathbf{x} \times (\mathbf{r}_y - \mathbf{R}_B)}{\|\mathbf{x} \times (\mathbf{r}_y - \mathbf{R}_B)\|} \\ \mathbf{y} &= \mathbf{z} \times \mathbf{x}\end{aligned}\tag{A.7}$$

Die Projektionen der beiden Geraden sind nun parallel. Durch eine Scherung mit der Matrix

$$\begin{pmatrix} 1 & s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{A.8})$$

$$s = \frac{\mathbf{x} \cdot (\mathbf{r}_y - \mathbf{R}_B)}{\mathbf{y} \cdot (\mathbf{r}_y - \mathbf{R}_B)} \quad (\text{A.9})$$

können die Geraden orthogonal gemacht werden. Setzt man das Frustum mit

$$\begin{aligned} a_x &= -\frac{(\mathbf{a} - \mathbf{R}_B) \cdot \mathbf{x}}{(\mathbf{a} - \mathbf{R}_B) \cdot \mathbf{z}} & a_y &= -\frac{(\mathbf{a} - \mathbf{R}_B) \cdot \mathbf{y}}{(\mathbf{a} - \mathbf{R}_B) \cdot \mathbf{z}} \\ c_x &= -\frac{(\mathbf{c} - \mathbf{R}_B) \cdot \mathbf{x}}{(\mathbf{c} - \mathbf{R}_B) \cdot \mathbf{z}} & c_y &= -\frac{(\mathbf{c} - \mathbf{R}_B) \cdot \mathbf{y}}{(\mathbf{c} - \mathbf{R}_B) \cdot \mathbf{z}} \end{aligned} \quad (\text{A.10})$$

als

`glFrustum($a_x + sa_y, c_x + sc_y, a_y, c_y, n, f$),`

so wird genau der in Abbildung A.1 grau markierte Bereich auf den Viewport abgebildet.

A.3 Projektion für Tetraeder Maps

Bei der Erzeugung der Tetraeder Map muss ein Ausschnitt der Umgebung, der einer Seitenfläche des Tetraeders von seinem Schwerpunkt aus gesehen entspricht, auf ein Dreieck im Bildspeicher projiziert werden. Dies kann mit einer geeigneten Projektionsmatrix und einer zusätzlichen Clip Plane erreicht werden. Die Projektionsmatrix ist das Produkt aus einer Scherung und einer Projektion, die das umschließende Rechteck auf den Viewport abbildet:

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{6}} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{3} & \frac{1}{3} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad (\text{A.11})$$

f und n sind die Entfernungen zur far und near Clip Plane.

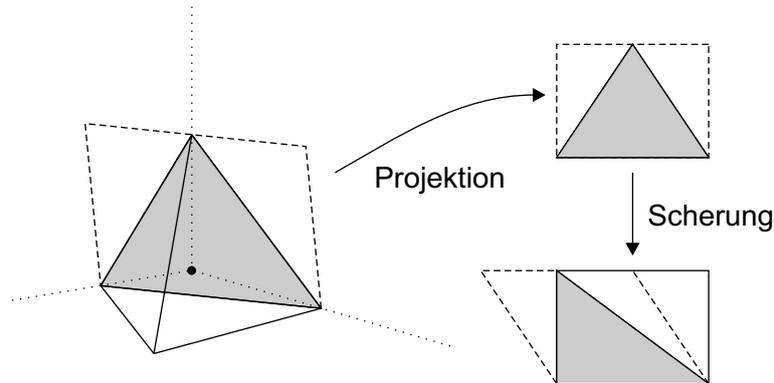


Abbildung A.2: *Projektion bei der Erstellung der Tetraeder Map.*

Eine zusätzliche Clip Plane mit den Parametern $(-\sqrt{3}/2, -1/2, -\sqrt{2}, 0)$ sorgt dafür, dass nur der grau markierte Bereich des Bildspeichers beschrieben wird. Alternativ kann dazu auch der Stencil Buffer verwendet werden.

Die weiteren drei Seiten werden analog gerendert, um die komplette Tetraeder Map zu erhalten.

A.4 Dokumentation Vertex- und Pixelshader

Dieser Abschnitt beschreibt die verwendeten Vertex- und Pixelshader in der durch die OpenGL `GL_ARB_VERTEX_PROGRAM` und `GL_ARB_FRAGMENT_PROGRAM` Erweiterungen [38] definierten Sprache.

A.4.1 Aberration

Das kürzeste und schnellste Programm berechnet lediglich die Aberration und bestimmt die Farbe aus der Rundumsicht für jeden Bildpunkt. Dazu wird ein Rechteck gezeichnet, das den gesamten Viewport ausfüllt. Die Richtung wird für jeden Bildpunkt aus den Richtungen der Eckpunkte in der Form $(x, y, 1)$ von der Textureinheit 0 linear interpoliert. `program.env[0]` und `program.env[1]` enthalten die Parameter β und γ als $(\beta, 0, 0, 0)$ und $(\gamma, 1, 1, 1)$. Die Rundumsicht ist als Cubemap an die Textureinheit 0 gebunden.

```
!!ARBfp1.0

ATTRIB direction = fragment.texcoord[0];
PARAM  beta      = program.env[0];
PARAM  gamma     = program.env[1];
TEMP   newdirection;
TEMP   norm;

# normalize direction and subtract beta
DP3 norm.w, direction, direction;
RSQ norm.w, norm.w;
MAD newdirection, direction, norm.w, -beta;

# multiply by gamma and sample texture
MUL newdirection, newdirection, gamma;
TEX result.color, newdirection, texture[0], CUBE;

END
```

A.4.2 Geometrieangepassung

Die Geometrieangepassung lässt sich mit drei zusätzlichen Befehlen in das Programm integrieren. Die Verzerrungstextur enthält für jeden Bildpunkt die darzustellende Richtung nach 4.3 und ist an die Textureinheit 2 gebunden.

```

!!ARBfp1.0

ATTRIB pixel = fragment.texcoord[0];
PARAM  beta = program.env[0];
PARAM  gamma = program.env[1];
PARAM  scale = { 2.6666, 2, 2, 2 };
PARAM  shift = { 1.3333, 1, 1, 1 };
TEMP   direction, newdirection, norm;

# get the direction from the distortion texture
TEX direction, pixel, texture[2], 2D;
MAD direction, direction, scale, -shift;
SWZ direction, direction, x, w, 1, 0;

# normalize direction and subtract beta
DP3 norm.w, direction, direction;
RSQ norm.w, norm.w;
MAD newdirection, direction, norm.w, -beta;

# multiply by gamma and sample texture
MUL newdirection, newdirection, gamma;
TEX result.color, newdirection, texture[0], CUBE;

END

```

A.4.3 Vollständige relativistische Umrechnung

Das Programm für die komplette Umrechnung berechnet für jeden Bildpunkt die relativistisch gedrehte Richtung und den Dopplerfaktor. Mit der neuen Richtung wird die Intensität von drei Kanälen bzw. Materialien aus der Rundumsicht bestimmt. Zusammen mit dem Dopplerfaktor werden die verschobenen Farbwerte aus der Dopplermap ermittelt. Die so erhaltenen Farbwerte werden mit D^5 multipliziert und durch eine einfache Abbildung auf den darstellbaren Bereich zwischen null und eins gebracht.

```

!!ARBfp1.0

ATTRIB direction = fragment.texcoord[0];
PARAM beta = program.env[0];
PARAM gamma = program.env[1];
PARAM attenuation = program.env[2];
PARAM one = { 1, 1, 1, 1 };
PARAM mone = { -1, 0, 0, 0 };

```

```
PARAM ch0          = { 1, 0.25, 0, 0 };
PARAM ch1          = { 1, 0.50, 0, 0 };
PARAM ch2          = { 1, 0.75, 0, 0 };

TEMP norm, newdirection, intensities, color;
TEMP doppler, doppler5, dopplermap;
TEMP channel0, channel1, channel2;

# normalize direction
DP3 norm.w, direction, direction;
RSQ norm.w, norm.w;
MUL newdirection, direction, norm.w;

# compute doppler factor
MAD doppler, -beta, newdirection.z, { 1, 1, 1, 1 };
MUL doppler, doppler, gamma;
RCP doppler, doppler.z;

# get position in doppler map
MUL dopplermap, doppler, -one;
EX2 dopplermap, dopplermap.x;

# compute d^5 and combine with attenuation
MUL doppler5, doppler, doppler;
MUL doppler5, doppler5, doppler5;
MUL doppler5, doppler5, doppler;
MUL doppler5, doppler5, attenuation;

# get new direction and intensities
SUB newdirection, newdirection, beta;
MUL newdirection, newdirection, gamma;
TEX intensities, newdirection, texture[0], CUBE;

# get shifted colors from the dopplermap
MAD channel0, dopplermap, mone, ch0;
MAD channel1, dopplermap, mone, ch1;
MAD channel2, dopplermap, mone, ch2;
TEX channel0, channel0, texture[1], 2D;
TEX channel1, channel1, texture[1], 2D;
TEX channel2, channel2, texture[1], 2D;

# sum up colors from all channels according to their intensity
MUL color, channel0, intensities.r;
MAD color, channel1, intensities.g, color;
```

```

MAD color, channel2, intensities.b, color;

# multiply color with doppler/attenuation factor
# and apply a simple hdr mapping
MUL color, color, doppler5.x;
EX2 color.r, color.r;
EX2 color.g, color.g;
EX2 color.b, color.b;
SUB result.color, { 1, 1, 1, 2 }, color;

END

```

A.4.4 Geometriebasierte Verzerrung

Der Vertexshader für die geometriebasierte Verzerrung berechnet die retardierte Position nach Gleichung 2.22. Als Parameter werden die vorberechneten Werte für

$$\beta, \gamma^2, \frac{\gamma}{\gamma+1}, \frac{1}{\gamma^2}, \mathbf{r}_0, ct_0 \text{ und } \frac{1}{\gamma^2}(ct_0)^2$$

an das Programm übergeben.

```

!!ARBvp1.0

PARAM projection[4] = { state.matrix.mvp };
PARAM beta         = program.env[0];
PARAM gamma2      = program.env[1];
PARAM gamma_over_gp1 = program.env[2];
PARAM one_over_gamma2 = program.env[3];
PARAM r0          = program.env[4];
PARAM ct0         = program.env[5];
PARAM ct02        = program.env[6];

TEMP b, c, d, r, temp;

# d = r' - gamma/(gamma+1) beta*r' beta - r0
DP3 d, beta, vertex.position;
MUL d, d.x, beta;
MAD d, gamma_over_gp1, d, r0;
SUB d, vertex.position, d;

# b = beta * d + ct0

```

```
DP3 b, beta, d;
ADD b, b, ct0;

# c = ( d * d - ct0 * ct0 ) / gamma / gamma
DP3 c, d, d;
MAD c, one_over_gamma2, c, -ct02;

# compute square root of b * b + c
MAD temp, b, b, c;
RCP temp, temp.x;
RSQ temp, temp.x;

# retarded = gamma * gamma * ( b - sqrt ) * beta + d + r0
SUB temp, b, temp;
MUL temp, temp, gamma2;
MAD r, temp.x, beta, d;
ADD r, r, r0;

# modelview and projection transformation
DP4 temp.x, projection[0], r;
DP4 temp.y, projection[1], r;
DP4 temp.z, projection[2], r;
DP4 temp.w, projection[3], r;

# keep color and texture coordinates
MOV result.position, temp;
MOV result.color, vertex.color;
MOV result.texcoord, vertex.texcoord;

END
```

A.5 Interaktives schwarzes Loch

Am Beispiel des interaktiven schwarzen Loches kann die Verzerrung und Farb-/ Helligkeitskorrektur eines Videostreams demonstriert werden.

Die Bilder werden von der Kamera Dalsa DS-22-2M30[40] mit einer Auflösung von 1600×1200 Bildpunkten und einer Bildrate von 30 Bildern pro Sekunde aufgenommen und über die Camera-Link Schnittstelle digital an den Framegrabber microEnable III[34] übertragen.

Die Farbe wird von der Kamera mit Hilfe von drei Filtern bestimmt, die jeweils einen Pixel bedecken und in einem Bayer-Muster angeordnet sind. Die Farbinformation eines Bildpunktes wird aus dem Wert des Pixels und den interpolierten Werten für die beiden verbleibenden Kanäle berechnet, wobei die Umrechnung direkt auf der Framegrabberhardware erfolgen kann. Bei der aktuellen Implementierung wird allerdings das Originalbild übertragen, da auf dem System nur ein 32bit / 33MHz PCI Bus zur Verfügung steht, der statt der erforderlichen 180 MB/s nur eine Übertragungsrate von etwa 100 MB/s bereitstellen kann. Die Umrechnung wird von der CPU erledigt, bevor die Daten als Textur an die Grafikkarte übertragen werden.



Abbildung A.3: *Interaktives schwarzes Loch: Benutzeroberfläche auf dem Touchscreen. Neben dem Kamerabild können vier weitere Aufnahmen ausgewählt werden, das schwarze Loch kann verschoben und in der Masse verändert werden.*

Die Verzerrung und Anpassung der Farbe erfolgt wieder für jeden Bildpunkt auf der Grafikkarte. Dazu sind die Verzerrungen, die von dem simulierten schwarzen Loch hervorgerufen werden, in Form von verschiedenen Texturen für die unterschiedlichen Massen abgelegt. Die Texturen werden beim Programmstart durch die numerische Lösung

der Geodätengleichung in der Schwarzschildmetrik berechnet.¹ Während der Simulation kann der Benutzer das schwarze Loch verschieben und die Masse verändern, was durch die Auswahl der entsprechenden Verzerrungstextur und die Verschiebung der Texturkoordinaten interaktiv erfolgen kann.



Abbildung A.4: *Interaktives schwarzes Loch: Aufbau mit Touchscreen, Plasmascreen und der auf den Benutzer gerichteten Kamera (oben Mitte)*

Die Farb- und Helligkeitsanpassung wird zur Korrektur der Vignettierung und Farbvignettierung verwendet. Die Vignettierung wird durch das große Gesichtsfeld und das dafür notwendige Objektiv mit kurzer Brennweite verursacht, die Farbvignettierung durch den Infrarotsperfilter, der vor dem Objektiv angebracht ist. Der Filter ist als Interferenzfilter ausgeführt, dadurch ist die Kante des Filters für Strahlen, die nicht achsenparallel sind, verschoben, und die entsprechenden Bildpunkte erscheinen grünlicher. Beide Fehler lassen sich im Pixelshader korrigieren. Die Textur mit den Korrekturfaktoren für die einzelnen Bildpunkte kann durch Anpassung weniger Parameter oder direkt aus einer Flatfield-Aufnahme erzeugt werden.

Der Infrarotsperfilter ist notwendig, weil der Sensor der Kamera über den sichtbaren Bereich hinaus im Infrarot empfindlich ist (vgl. Abbildung 2.2. Ohne den Filter erscheinen die Farben verfälscht. Besonders deutlich ist dies an ‘schwarzen’ Körpern (entsprechend unserer Wahrnehmung bzw. mit Filter) zu sehen. Je nach Material und damit Infrarotreflektivität wird der Körper ohne Filter schwarz oder braun wiedergegeben. Der Effekt ist gleichzeitig ein gutes Beispiel für die in Abschnitt 2.4 angesprochene Problematik, dass sich die spektrale Verteilung keineswegs aus den RGB-Daten normaler Aufnahmen ermitteln lässt.

¹Der Quelltext für die Berechnung wurde von Thomas Müller zur Verfügung gestellt.

Literaturverzeichnis

- [1] *3ds max*. Autodesk, Inc., 2001. www.discreet.com
- [2] Adelson EH, Bergen JR. The Plenoptic Function and the Elements of Early Vision. In Landy M, Movshon JA (Hrsg.) *Computational Models of Visual Processing*, 3-20, MIT Press, Cambridge, 1991.
- [3] *ADLIP All Dome Laser Image Projektor*. Mitteilung Hasse R, Carl Zeiss GmbH, Jena. 2003. www.zeiss.de
- [4] *ASTER Spectral Library*. Jet Propulsion Laboratory, California Institute of Technology, Pasadena, 1999. asterweb.jpl.nasa.gov/asterhome
- [5] Bender M, Brill M. *Computergrafik*. Carl Hanser Verlag, München, 2003
- [6] Betts C. Fast Rendering of Relativistic Objects. *Journal of Visualization and Computer Animation*, **9**: 17-31, 1998.
- [7] Boas ML. Apparent shape of large objects at relativistic speeds. *American Journal of Physics*, **29**(5):283-286, 1961.
- [8] Boas ML, Calhoun RC, Horan O. Binocular Observation of Moving Objects. *American Journal of Physics*, **39**:782-790, 1971.
- [9] Chang MC, Lai F, Chen WC. Image shading taking into account relativistic effects. *ACM Transactions on Graphics*, **15**(4):265–300, 1996.
- [10] Cruz-Neira C, Sandin DJ, DeFanti TA. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. *SIGGRAPH 93 Conference Proceedings*, **27**:135-142, 1993.
- [11] Dersch, H. *Panorama Tools*, 2000. www.ptgui.com/download2.html
- [12] Devlin K, Chalmers A, Wilkie A, Purgathofer W. Tone Reproduction and Physically Based Spectral Rendering. In Fellner D, Scopigno R (Hrsg.) *State of The Art Reports, Eurographics 2002*, 101-123, 2002.
- [13] *easyFLY Modellflugsimulator*. IPACS Hans-Borchers-Grüntjens GbR. 2002. www.ipacs.de

- [14] Fairchild MD. *Color appearance models*. Addison-Wesley, Reading, 1997.
- [15] Fernando R (Hrsg.). *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*. Addison-Wesley, Boston, 2004.
- [16] Foley JD, van Dam A, Feiner SK, Hughes JF. *Computer Graphics*. Addison-Wesley, Reading, 1990.
- [17] Fortes T. *Tetrahedron Environment Maps*. Master's Thesis, Chalmers University of Technology, Göteborg, 2000.
- [18] Gamow G. *Mr Tompkins in Paperback*. Cambridge University Press, Cambridge, 1993.
- [19] Gekelmann W, Maggs J, Xu L. Real-Time relativity. *Computers in Physics*, **5**(4):372-385, 1991.
- [20] Glassner AS. How to Derive a Spectrum from an RGB Triplet. *IEEE Computer Graphics & Applications*, **9**(4):95-99, 1989.
- [21] Gross M, Staadt O, Kunz A, Meier M. blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence. *Proceedings of ACM SIGGRAPH 2003*, **22**(3):819-827, 2003.
- [22] Hsiung PK, Dunn RHP. Visualizing relativistic effects in spacetime. In *Proceedings of Supercomputing '89 Conference*, 597-606, 1989.
- [23] Hsiung PK, Thibadeau RH, Wu M. T-buffer: fast visualization of relativistic effects in spacetime. *Computer Graphics*, **24**:83-88, 1990.
- [24] *Intel C++ Compiler 7.1 for Windows*. Intel Corporation, Santa Clara, 2003. www.intel.com/software/products/compilers/
- [25] Lampa A. Wie erscheint nach der Relativitätstheorie ein bewegter Stab einem ruhenden Beobachter? *Zeitschrift für Physik*, **27**:138-148, 1924.
- [26] Larson GW, Rushmeier H, Piatko C. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Transactions on Visualization and Computer Graphics*, **3**(4):291-306, 1997.
- [27] Lindquist RW. Relativistic Transport Theory. *Annals of Physics*, **37**:487-518, 1966.
- [28] Majumder A, He Z, Towles H, Welch G. Achieving Color Uniformity Across Multi-Projector Displays. In Ertl T, Hamann B, Varshney A (Hrsg.) *Proceedings of the IEEE Conference on Visualization 2000*, 117-124, 2000.
- [29] Mallot HA. *Sehen und die Verarbeitung visueller Information*. Vieweg, Braunschweig, 2000.

- [30] Manoukian EB, Sukkhasena S. Projection of relativistically moving objects on a two-dimensional plane, the ‘train’ paradox and the visibility of the Lorentz contraction. *European Journal of Physics*, **23**:103-110, 2002.
- [31] Marr D. *Vision*. Freeman, San Francisco, 1982.
- [32] McGill NC. The apparent shape of rapidly moving objects in special relativity. *Contemporary Physics*, **9**:33-48, 1968.
- [33] McKinley JM. Relativistic transformations of light power. *American Journal of Physics*, **47**(7):602-605, 1979.
- [34] *microEnable III Framegrabber*. SILICONSOFTWARE GmbH, Mannheim, 2003. www.silicon-software.de/microenable3.html
- [35] Misner CW, Thorne KS, Wheeler JA. *Gravitation*. W. H. Freeman and Company, New York, 1973.
- [36] Müller T. *Visualisierung der Relativitätstheorie*. Dissertation, Universität Tübingen, 2005.
- [37] *NVIDIA SLI (Scalable Link Interface)*. nVidia Corporation, Santa Clara, 2004. www.nvidia.com/page/sli.html
- [38] *OpenGL Extension Registry*. oss.sgi.com/projects/ogl-sample/registry/
- [39] *openWARP combiner*. Bühlhoff H, Max-Planck-Institut für biologische Kybernetik, Tübingen, 2004. www.openwarp.com
- [40] *Pantera SA 2M30* Dalsa, Waterloo, 2003. www.dalsa.com
- [41] Peebles PJE, Wilkinson DT. Comment on the anisotropy of the primeval fireball. *Physical Review*, **174**(5):2168, 1968.
- [42] Penrose R. The apparent shape of a relativistically moving sphere. *Proceedings of the Cambridge Philosophical Society*, **55**:137-139, 1959.
- [43] *PolyTrans*. Okino Computer Graphics, Inc., Toronto, 2004. www.okino.com
- [44] Raskar R, van Baar J, Chai JX. A Low-Cost Projector Mosaic with Fast Registration. In *Proceedings of The 5th Asian Conference on Computer Vision*, 2002. www.merl.com/publications/TR2002-014/
- [45] Raskar R, van Baar J, Willwacher T. Quadric Transfer for Immersive Curved Screen Displays. *EUROGRAPHICS 2004 Proceedings*, **23**(3):451-460, 2004.
- [46] Rindler W. *Relativity: special, general and cosmological*. Oxford University Press, Oxford, 2001.
- [47] Schwarz PM, Schwarz JH. *Special relativity: from Einstein to strings*. Cambridge University Press, Cambridge, 2004.

- [48] Scott GD, Viner RR. The geometrical appearance of large objects moving at relativistic speeds. *American Journal of Physics*, **33**(7):534-536, 1965.
- [49] Scott GD, van Driel HJ. Geometrical appearances at relativistic speeds. *American Journal of Physics*, **38**(8):971-977, 1970.
- [50] *SeeReal C-nt Display*. SeeReal Technologies GmbH, Dresden, 2004. www.seereal.com/EN/products_cnt.en.htm
- [51] Seetzen H, Heidrich W, Stuerzlinger W, Ward G, Whitehead L, Trentacoste M, Ghosh A, Vorozcovs A. High Dynamic Range Display Systems. *ACM Transactions on Graphics*, **23**(3):760-768, 2004.
- [52] Segal M, Akeley K. *The OpenGL Graphics System: A Specification (Version 1.5)*. Silicon Graphics, Inc., Mountain View, 2003. www.opengl.org
- [53] Stevenson SB, Cormack LK, Schor CM. Hyperacuity, superresolution and gap resolution in human stereopsis. *Vision Research*, **29**(11):1597-1605, 1989.
- [54] Stockman A, Sharpe LS. Cone spectral sensitivities and color matching. In Gegenfurtner K, Sharpe LS (Hrsg.) *Color Vision*, 53-87, Cambridge University Press, Cambridge, 1999. cvision.ucsd.edu/index.htm
- [55] Stroustrup B. *The C++ Programming Language*. Addison-Wesley, Reading, 1997.
- [56] Terrell J. Invisibility of the Lorentz Contraction. *Physical Review*, **116**(4): 1041-1045, 1959.
- [57] University of Minnesota, Silicon Graphics, Inc., Ciprio, Inc., IBM, Inc. PowerWall. www.lcse.umn.edu/research/powerwall/powerwall.html
- [58] *Virtual Tübingen*. Bühlhoff H, Max-Planck-Institut für biologische Kybernetik, Tübingen, 2001. vr.tuebingen.mpg.de
- [59] Weiskopf D, Kraus U, Ruder H. Searchlight and Doppler Effects in the Visualization of Special Relativity: A Corrected Derivation of the Transformation of Radiance. *ACM Transactions on Graphics*, **18**(3):278-292, 1999.
- [60] Weiskopf D. A texture mapping approach for the visualization of special relativity. In *IEEE Visualization 1999 Late Breaking Hot Topics Proceedings*, 41-44, 1999.
- [61] Weiskopf D, Kraus U, Ruder H. Illumination and acceleration in the visualization of special relativity: a comment on fast rendering of relativistic objects. *Journal of Visualization and Computer Animation*, **11**:185-195, 2000.
- [62] Weiskopf D. Fast Visualization of Special Relativistic Effects on Geometry and Illumination. In de Leeuw W, van Liere R (Hrsg.) *Data Visualization 2000 (Proceedings of the EG/IEEE TCVG Symposium on Visualization)*, 219-228, Springer, Wien, 2000.

- [63] Weiskopf D, Kobras D, Ruder H. Real-World Relativity: Image-Based Special Relativistic Visualization. In Ertl T, Hamann B, Varshney A. (Hrsg.) *IEEE Visualization 2000 Proceedings*, 303-310, ACM Press, 2000.
- [64] Weiskopf D. *Visualization of Four-Dimensional Spacetimes*. Dissertation, Universität Tübingen, 2001.
- [65] Weisskopf VF. The visual appearance of rapidly moving objects. *Physics Today*, **13**(9):24-27, 1960.
- [66] Williams L. Pyramidal Parametrics. *Computer Graphics (SIGGRAPH '83 Proceedings)*, **17**(3):1-11, 1983.
- [67] Wyszecki G, Stiles WS. *Color Science*. Wiley, New York, 2000.
- [68] *X3D-17 Display AD*. X3D Technologies GmbH, Jena, 2004. www.x3dworld.de
- [69] Zhang Z, Faugeras O. *3D Dynamic Scene Analysis*. Springer, Berlin, 1992.

Lebenslauf

Marc P Borchers

geboren am 07.02.1972 in Münster

Schule

1978 - 1982	Grundschule Berkheim
1982 - 1985	Gymnasium Ochsenhausen
1985 - 1991	Gymnasium Cochem

Studium

1992 - 1998	Studium der Physik an der Universität Tübingen
1994	Vordiplom Physik (Universität Tübingen)
1995 - 1996	Auslandsstudium an der University of Arizona, Tucson, USA
1998	Diplom Physik (Universität Tübingen)

Promotion

1999 - 2004	Anfertigung der Dissertation "Interaktive und stereoskopische Visualisierung in der speziellen Relativitätstheorie" bei Prof. Dr. Hanns Ruder am Institut für Theoretische Astrophysik und Computational Physics
-------------	--

Akademische Lehrer

Universität Tübingen	M. Baumann, W. Dittrich, H. Fischer, K.-H. Gaukler, R. Hübener, H. Kaul, D. Kern, B. Kümmerer, O. Lutz, E. Plies, H. Reinhardt, H. Ruder, R. Schief, R. Staubert, G. Staudt, H.-P. Thier, G. Wagner, M. Wolff
University of Arizona	H. Flaschka, P. Jessen, D. Pickrell, M. Scadron, M. Shupe