Kapitel 6

Demonstration von Anwendungen

Die Eignung des in der vorliegenden Arbeit entworfenen Systemes für die Simulation von dynamischem Verhalten in virtuellen Umgebungen soll nun anhand von Anwendungen dargestellt werden.

6.1 Astronauten Trainingssimulator

In Zusammenarbeit mit dem Johnson Space Center in Houston und der University of Houston Downtown entstand 1995 eine virtuelle Umgebung zum Training von Astronauten [FDFM96, Bowe96]. Hier konnte das im Rahmen dieser Arbeit entwickelte Simulationssystem zum ersten mal eingesetzt werden.

6.1.1 Trainingsszenario

Am Johnson Space Center (JSC) werden virtuelle Umgebungen seit Jahren für das Training von Bodencrews eingesetzt. Die Aufgabe der Bodencrew ist es, die Astronauten bei der Durchführung ihrer Aufgaben über Sprechfunk zu unterstützen. Dazu ist es notwendig, daß die Mitarbeiter der Bodencrews den geplanten Ablauf der Mission genau kennen und die Astronauten bei deren Erfüllung anleiten.

Bei Arbeiten außerhalb des Raumschiffes (Extra Vehicular Activity, kurz EVA) ist es besonders wichtig, daß bei den Bodencrews eine genaue räumliche Vorstellung der Situation vorhanden ist, um den Astronauten genaue Anweisungen geben zu können. Hierzu kann die Virtuelle Realität beitragen.

Im Rahmen dieses Projektes sollte nun untersucht werden, ob die Anwendung von Virtueller Realität auch für das Training der Astronauten selbst eingesetzt werden kann. Von besonderem Interesse war die Möglichkeit eine Aufgabe zu simulieren, die von zwei Astronauten in Zusammenarbeit bewältigt werden muß. Solche Aufgaben werden normalerweise gemeinsam am JSC trainiert. Für internationale Missionen entsteht dabei ein hoher Kosten- und Zeitaufwand durch die notwendigen Reisen. Diese kann durch den Einsatz einer verteilten virtuellen Umgebung zum Astronautentraining reduziert werden.

Die zu simulierende Mission bestand in der Aufgabe, ein defektes Aggregat des Hubble Space Telescopes auszutauschen. Diese Mission war bereits erfolgreich abgeschlossen worden und es lag umfangreiches Material vor. Die Aufgabe erfordert die Zusammenarbeit von zwei Astronauten. Während einer das defekte Aggregat aus dem Teleskop ausbaut, holt der andere das Ersatzaggreat aus der Ladebucht. Dann treffen sich beide Astronauten und tauschen die Teile. Schließlich wird das neue Aggregat eingesetzt und das defekte Teil wieder in der Ladebucht des Space Shuttle verstaut.

6.1.2 Technische Installation

Das virtuelle Modell für diese Mission wurde durch Virtual Reality Laboratory (VETL) am Johnson Space Center entwickelt. Es stellte das Space Shuttle mit geöffneter Ladebucht und das angedockte Hubble Teleskop dar. Zusätzlich war ein einfaches Modell eines Astronauten im Raumanzug vorhanden.

Das VR-Trainingssystem bestand aus zwei gleichzeitig laufenden VR-Systemen. Am JSC lief das bereits erprobte VR-Tool, welches dort für das Training der Bodencrews entwickelt worden war.

Am Fraunhofer-IGD wurde ein System eingesetzt, welches aus dem Echtzeit-Renderer Y [Rein94], dem Interaktions-Toolkit INTO (siehe Abschnitt 2.2.2)[1], und dem Simulationssystem für Autonome Objekte bestand.

Auf beiden Seiten stand dasselbe grafische Modell in Form von Inventor-Dateien zur Verfügung.

Zwischen beiden Systemen bestand eine TCP/IP Verbindung über eine ISDN-Standleitung mit 2x64 KBit/Sekunde. Über diese Leitung wurden ständig Informationen ausgetauscht, um beide Systeme synchron zu halten, und so eine gemeinsame virtuelle Umgebung zu schaffen. Zusätzlich bestand eine Sprechverbindung zwischen beiden Astronauten über analoges Telefon.

[1]IDEAL stand damals noch nicht zur Verfügung, dessen Entwicklung begann erst 1997

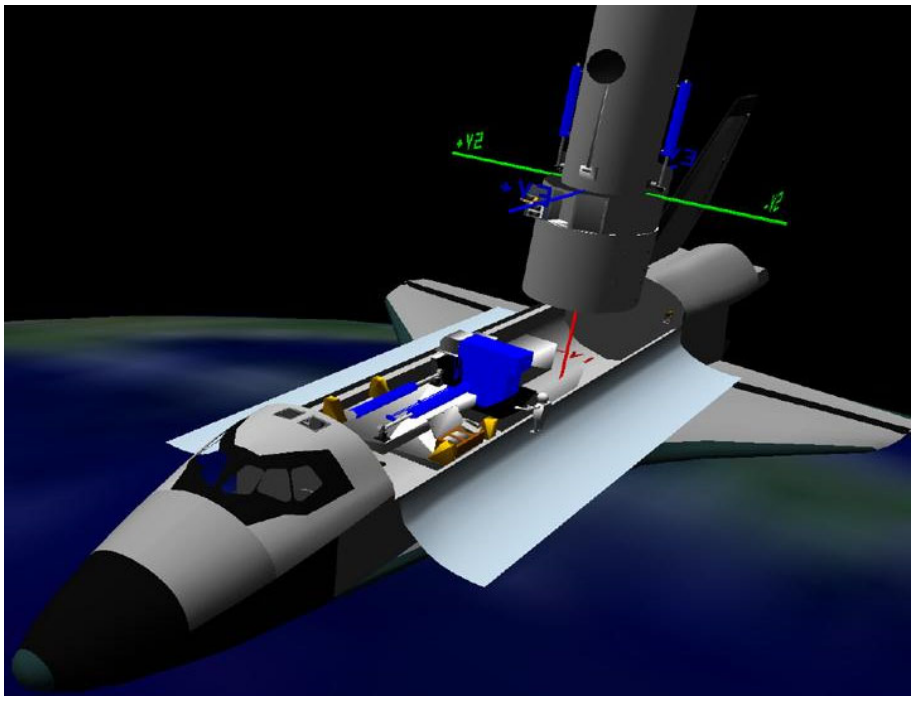

Abbildung 6.1: Modell des Space Shuttle mit angedocktem Hubble Teleskop

Für die grafische Darstellung wurde auf beiden Seiten ein Head Mounted Display (Monitorbrille) mit Headtracker eingesetzt, zur Interaktion wurde ein Datenhandschuh verwendet. Für die Navigation stand am JSC ein Steuerknüppel zur Verfügung, wie er in der Raumfahrt eingesetzt wird, am Fraunhofer-IGD wurde stattdessen eine Spacemouse verwendet.

Jeder der beiden Astronauten wurde durch ein Modell repräsentiert, welches aus einem Raumanzug mit beweglichem rechten Arm bestand. Die Position der rechten Hand wurde durch den am Datenhandschuh angebrachten Tracker gesteuert, die Finger über den Datenhandschuh bewegt.

6.1.3 Einsatz Autonomer Objekte

Das Simulationssystem für Autonome Objekte wurde eingesetzt, um die Figurinen, die die Astronauten darstellten zu steuern. Hierzu konnte das in Kapitel 4 entwickelte Benutzermodell herangezogen werden. Den Zusammenhang zwischen Szenengraphen und Autonomen Objekten stellt Abbildung 6.2 dar.

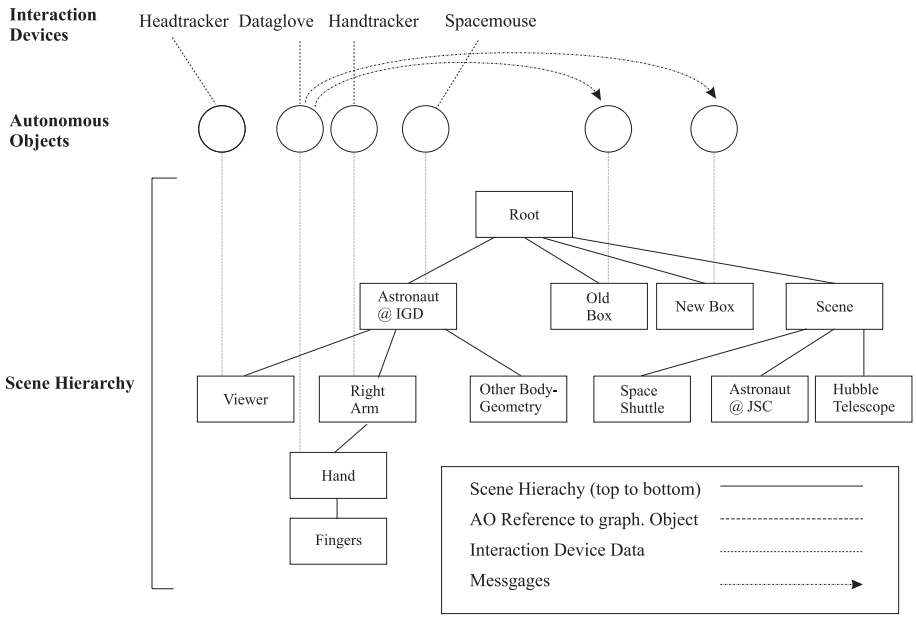

Abbildung 6.2: Autonome Objekte im Astronauten Trainingssimulator

6.1.4 Nachrichten zum Greifen der Aggregate

Die Aufgabe der Astronauten war es, das defekte und das neue Aggregat in der Szene zu bewegen und beide einander zu übergeben. Dazu mußten beide Aggregate gegriffen und losgelassen werden können. Die Greifoperation selbst wurde über Kollisions- und Gestenerkennung innerhalb des Autonomen Objektes, welches die Hand kontrolliert, realisiert. Diese Funktionen stellten der Renderer bzw. das Interaktionstoolkit zur Verfügung.

Ein Objekt wird gegriffen, wenn eine Kollision zwischen der Hand und dem Objekt vorliegt und gleichzeitig eine Faust-Geste erkannt wird. Losgelassen wird ein Objekt, wenn eine andere Geste als die Faust erkannt wurde.

Sobald eines der Aggregate gegriffen wurde, muß es der Hand, die es gegriffen hat solange folgen, bis es wieder losgelassen wird.

Dazu sendet das Hand-AO an das AO, welches das zu greifende Aggregat kontrolliert eine entsprechende Nachricht. Es existieren zwei Arten von Nachrichten, für das Greifen (*Grab*) und das Loslassen (*Release*). Das empfangende AO reagiert auf den Empfang einer Nachricht, in dem es sein assoziiertes grafisches Objekt im Szenengraphen unter die Hand hängt (gegriffen) oder unter die Wurzel der Szene (losgelassen). Da grafische Objekte in hierarchischen Szenengraphen u.A. die Transformation ihren Eltern-Knoten erben, bewegt sich das gegriffene Objekt mit der Hand mit.

6.1.5 Synchronisation der Szenengraphen

Damit beide Astronauten eine gemeinsame virtuelle Umgebung erleben, in der sie interagieren und sich gegenseitig sehen können, müssen beide VR-Systeme miteinander Nachrichten austauschen um ihre Szenengraphen miteinander zu synchronisieren. Wie bereits erwähnt wurde, stand dazu eine Verbindung über TCP/IP-Sockets zur Verfügung. Da beide Systeme zu Beginn über identische Szenen verfügten und die TCP/IP Sockets die sichere Übertragung von Nachrichten garantieren, konnte das Protokoll sehr einfach gehalten werden. Es waren nur zwei Arten von Nachrichten nötig:

Zur Aktualisierung der Transformation eine grafischen Objektes, das sich bewegt hat und zur Information darüber, daß ein Objekt im Szenengraphen umgehängt wurde, weil es gegriffen oder losgelassen wurde.

Um Objekte im Szenengraphen zu identifizieren, wurde jedem Objekt eine eindeutige Identifikationsnummer zugeordnet. Da diese Nummern zunächst auf beiden Seiten (bei der NASA und am Fraunhofer-IGD) unterschiedlich waren, wurde eine Umsetztabelle realisiert, um die Nummern hin und her zu übersetzen.

Die beschriebene Funktionalität zur Synchronisierung der Szenengraphen wurde in den Y-Renderer integriert und lief so transparent für das Simulationssystem ab.

6.1.6 Das Experiment

Am 10. September 1995 wurde das System durch zwei Astronauten erprobt. Am JSC nahm daran der amerikanische Astronaut Bernard Harris, auf deutscher Seite der europäische Astronaut Ulf Merbold teil. Die beschriebene Aufgabe wurde mehrmals simuliert und erfolgreich abgeschlossen. Beide Astronauten beurteilten das System als brauchbar und vielversprechend [FDFM96].

Das Experiment wurde als gemeinsamer Beitrag von NASA und IGD anläßlich Ausstellung "Digital Bayou" auf der SIGGRAPH 1996 wiederholt und dort Besuchern präsentiert.

Abbildung 6.3 stellt die Szene aus Sicht des europäischen Astronauten dar. Links wird das defekte Aggregat übergeben. Die rechte Seite zeigt einen virtuellen Handschlag nach erfolgreich beendetem Experiment.

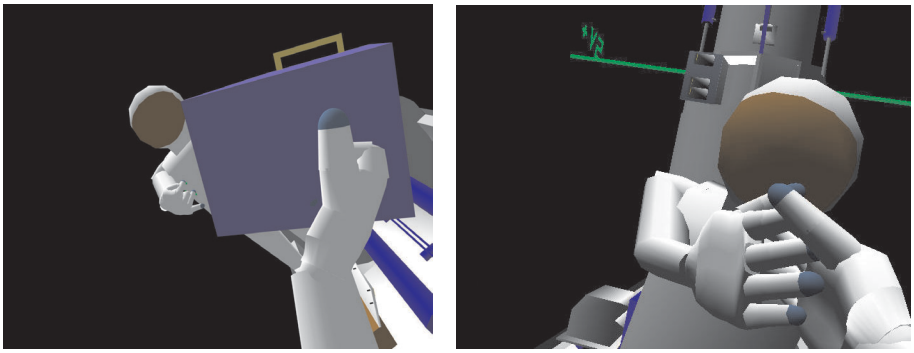

Abbildung 6.3: Szene aus Sicht des europäischen Astronauten

6.2 Das Virtuelle Ozeanarium

6.2.1 Einleitung

Das Virtuelle Ozeanarium wurde von 1996 bis 1998 am Fraunhofer Institut für grafische Datenverarbeitung für die Weltausstellung in Lissabon, EXPO'98 entwickelt [Merk97, Klam97, Froh97, Froh00]. Koordiniert wurde es vom Centro de Computaçao Graphica in Coimbra, Portugal.

Aufgabe war es, die Hauptattrakation der EXPO, das Ozeanarium in Form einer Virtuellen Umgebung darzustellen. Diese wurde in einem eigenen Auditorium auf der EXPO'98 einem großen Publikum präsentiert. Ein wichtiges Ziel war es dabei, das reale Ozeanarium so exakt wie möglich darzustellen. Dies galt nicht nur für das charakteristische Gebäude selbst und das umgebende Ausstellungsgelände, sondern auch für 5 Aquarien und das marine Leben, das sie beherbergen.

Anhand des Virtuellen Ozeanariums wird die Anwendung des von autonomen Objekten für die Simulation einer komplexen Unterwasserwelt mit hunderten von Lebewesen dargestellt. Die Basisfunktionalität des in Kapitel 3 eingeführten Systems wird für die Simulation des Verhaltens von Tieren und Pflanzen erweitert.

Zunächst werden verwandte Arbeiten im Bereich künstliches Leben diskutiert.

Als weitere Aspekte werden die Simulation von Lichteffekten unter Wasser, sowie die Anwendung von Level Of Detail Techniken zur Beschleunigung der grafischen Darstellung beschrieben.

Schließlich wird gezeigt, wie das Simulationsmodell mit dem quantitativen Lotka-Volterra Modell erweitert werden kann. Dadurch entsteht ein System, mit welchem Ergebnisse ökologischer oder wirtschaftswissenschaftlicher Simulationen grafisch-interaktiv dargestellt werden können.

Das Kapitel schließt mit einer Zusammenfassung.

6.2.2 Verwandte Arbeiten

Ein sehr umfassendes System zur Simulation von Fortbewegung, Wahrnehmung und Verhalten von Fischen führen Terzopoulos et. al [TeTG94, TuTe94, Tu96] ein. Dem dort beschriebenen Modell von Fischen liegt ein physikalisch basierter Ansatz zugrunde.

Der Körper jedes Fisches wird dort durch ein einheitliches Feder-Masse Modell beschrieben, welches für verschiedene Spezies parametriert werden kann. Einige der Federn können von einer übergeordneten motorischen Ebene aus zusammengezogen oder gedehnt werden und simulieren auf diese Weise Muskeln.

Ein einfaches hydrodynamisches Modell simuliert die Wechselwirkung der Fischoberfläche mit dem umgebenden Wasser. Die Wahrnehmung der Fische wird durch einen visuellen Sinn dargestellt, außerdem können die Fische die Wassertemperatur wahrnehmen.

Das Verhalten der Fische wird durch drei Kriterien bestimmt: A-priori festgelegtes Verhalten (*Habits*), den mentalen Zustand (Drei Variable H, L und F für Hunger, Libido und Furcht) und die wahrgenommenen Stimuli. Das Verhalten der höchsten Priorität wird aus eine Liste von möglichen Intentionen ausgewählt: Kollisionsvermeidung, Flucht vor Räubern, Schwarmbildung, Fressen, Paarung, Suche von Gebieten mit angenehmer Wassertemperatur oder Helligkeit. Die Prioritäten sind teilweise vorgegeben, so hat die Kollisionsvermeidung immer höchste Priorität. Die Wichtigkeit von Flucht, Fressen oder Paarung dagegen schwankt entsprechend den Werten der Variablen F, H bzw. L.

Wegen seiner grafischen Qualität, besonders der simulierten Schwimmbewegungen und der akkuraten Darstellung von Verhalten hat das Modell sowohl in der Computer Grafik als auch auf dem Gebiet der künstlichen Intelligenz Anerkennung gefunden. Auch für die vorliegende Arbeit konnten viele Anregungen gewonnen werden. Daher wird darauf immer wieder Bezug genommen werden.

Nachteilig an diesem Modell ist jedoch der hohe Rechenaufwand, insbesondere zur Simulation der zugrunde liegenden Feder-Masse Modelle. In [TuTe94] wird berichtet, daß auf einer Grafikworkstation SGI Indigo2 ein System aus 10 Fischen, 15 Futterpartikeln und 5 statischen Hindernissen mit 4 Frames pro Sekunde (inklusive grafischer Darstellung als Drahtgittermodell) darstellen lies. Auch wenn heutige Rechner mehr Rechen- und Grafikleistung zur Verfügung stellen, ist der dort beschriebene Ansatz für die Simulation von Systemen mit hunderten von Tieren in einer komplexen Umgebung und

grafisch aufwendigen Darstellung ungeeignet.

Von Allison et al. [AWHW96] wurde ein System zur visuellen Simulation von Gorillas in einem Zoogehege entwickelt. Das System wurde im Zoo von Atlanta installiert und war den Besuchern des Zoos zugänglich. Das simulierte Gorilla Habitat war dem des Zoos nachempfunden. Insofern besteht ein interessanter Zusammenhang zum Virtuellen Ozeanarium, welches ebenfalls ein reales Habitat nachbildet und in der Nähe seines realen Vorbildes ausgestellt und der Öffentlichkeit zugänglich war.

Das von Allison et. al beschriebene System kennt 5 Typen von Gorillas unterschiedlichen Alters und Geschlechts, welche als hierarchische polygonale Modelle realisiert wurden. Bewegung wird aus einzelnen vorgegebenen Posen zusammengesetzt, zwischen denen linear interpoliert wird. Ein Terrain-Verfolgungs Algorithmus erlaubt die Fortbewegung auf dem unebenen Boden des Geheges.

Die Installation im Zoo von Atlanta bestand aus einer kleinen Plattform, die von einem Geländer umgeben war. Für die grafische Darstellung und die Interaktion mit dem System stand ein Head Mounted Display (HMD) und ein Joystick zur Verfügung. Zielgruppe der Präsentation waren Schüler, die sich im Rahmen eines längerfristigen Programmes mit Gorillas und deren Verhalten beschäftigten. Einzelne Schüler konnten zunächst das Modell des Geheges erforschen und mit zwei virtuellen Gorillas interagieren.

Zwei Lernziele sollten erreicht werden: Die Schüler sollten auf experimentellem Wege über die soziale Interaktion zwischen Gorillas lernen. Diese basiert wesentlich auf der Stellung des Individuums in der Hierarchie. Zweitens sollten sie etwas über die Gestaltung von Gorilla Gehegen in Zoos lernen. Um diese Ziele zu erreichen, spielten die Schüler die Rolle eines heranwachsenden Gorillas, in der sie zwei erwachsenen Gorillas im Gehege begegneten. Je nach dem wie sie sich den Erwachsenen näherten reagierten diese mit Zuwendung, Ärger oder Angriff. Durch unterwürfige Posen oder Flucht konnte die Situation geklärt werden.

6.2.3 Systemarchitektur des Virtuellen Ozeanariums

Nachdem ein Simulationssystem für Verhalten von Objekten und ein System zur Anbindung von Interaktionsgeräten zur Verfügung steht, fehlt als dritte Komponente ein Renderer zur grafischen Darstellung der Szene. Hierzu wurde der Y-Renderer (kurz Y) [Rein94] herangezogen, der seit 1996 Teil des VR-Systems Virtual Design II ist. Basierend auf OpenGL, erlaubt Y die Darstellung von polygonalen Modellen und Szenen in Echtzeit und mit hoher grafischer Qualität. Diese Modelle können in verschiedenen Datenformaten vorliegen, u.A. im Fraunhofer-eigenen Format FHS[FHS98], VRML97 [Iso97].

Y ist in der Lage, die verschiedensten grafischen Displays anzusteuern, neben dem Bildschirm auch Großbild-Stereoprojektionen und Mehrseitenprojektionen wie die CAVE. Y hält die Szene intern als hierarchischen Szenengraphen und bietet eine Palette von Möglichkeiten auf Attribute von Objekten oder die Struktur des Szenengraphen zuzugreifen.

Als vierte und letzte Komponente wurde wurde eine einfache Ablaufsteuerung entwickelt. Diese Ablaufsteuerung ist Teil des Hauptprozesses des Systems (vergl. Abschnitt 3.11 ab Seite 53) und steuert das Gesamtsystem. Ihre Aufgabe ist es, beim Start des Systems die Szene und die benötigten Konfigurationsdateien zu laden. Sie wird ausführlicher in Abschnitt 6.2.6 dargestellt.

Damit ergibt sich ein Gesamtsystem welches Abbildung 6.4 skizziert.

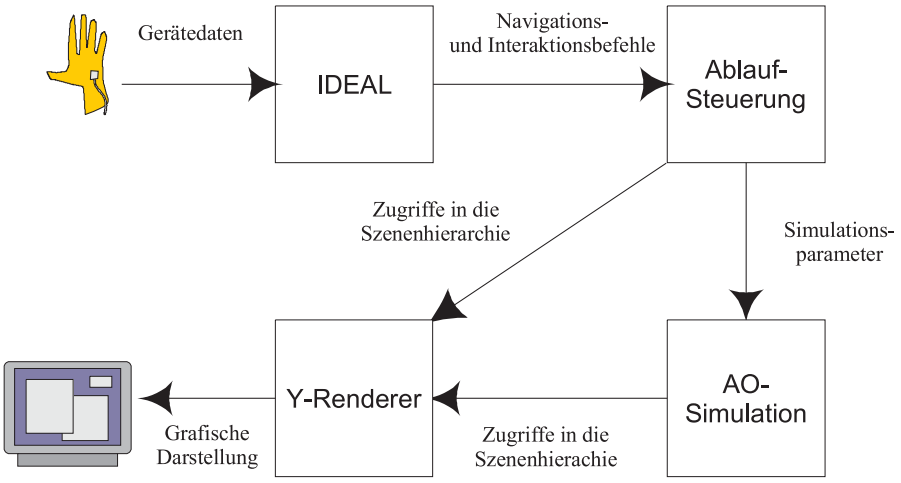

Abbildung 6.4: Softwarearchitektur des Gesamtsystems

Das IDEAL -System übernimmt die Anbindung der Interaktionsgeräte. Es verarbeitet die Gerätedaten die bei der Benutzerinteraktion entstehen und sendet diese an die Ablaufsteuerung.

Diese wiederum steuert die Simulation und die Darstellung der Szene. Das Simulationssystem realisiert das Verhalten der simulierten Objekte und aktualisiert zyklisch deren grafische Darstellung durch den Renderer.

Der Renderer schließlich stellt die Szene grafisch dar.

6.2.4 Aufbau und Inhalte des Virtuellen Ozeanariums

Wie sein reales Vorbild besteht das Virtuelle Ozeanarium aus vier Biotopen, die um einen zentralen Haupttank herum gruppiert sind. Diese sogenannten Habitats zeigen die Küstengebiete des nördlichen Atlantiks, der Antarktis,

des indischen Ozeans und des Pazifiks in Kalifornien. Jedes der Habitats gliedert sich in eine Unterwasser- und eine Küstenebene.

Der zentrale Haupttank, der auch Global Ocean Tank genannt wird, verbindet die vier Habitats und stellt die globalen Weltmeere dar. Mit der Höhe von zwei Stockwerken und einem Fassungsvermögen von $5000m^3$ Wasser stellt er eines der größten Aquarien der Welt dar. Abbildung 6.5 gibt einen Überblick.

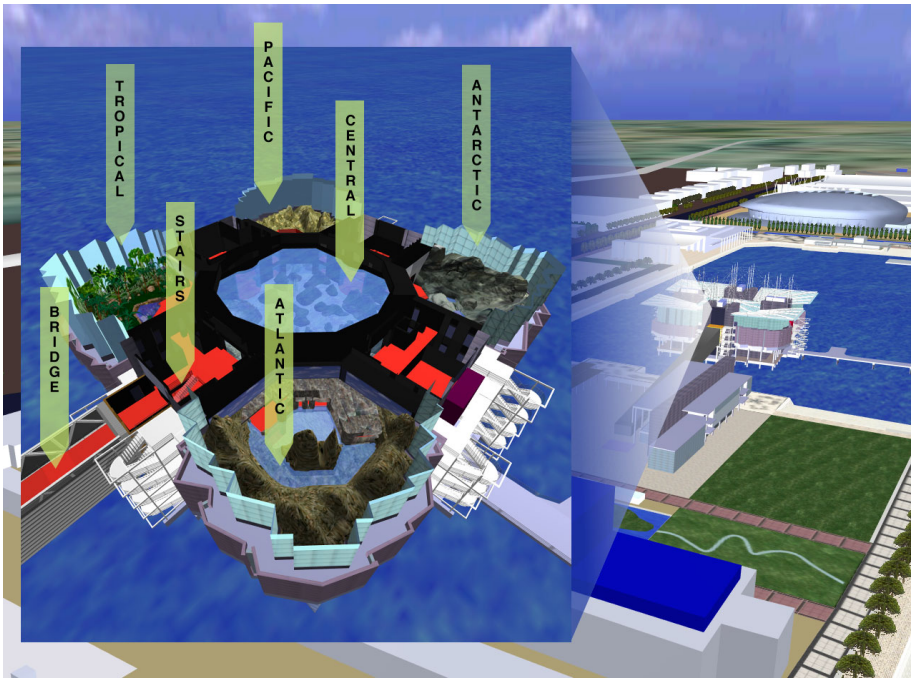

Abbildung 6.5: Übersicht über das Virtuelle Ozeanarium

Das Modell des Virtuellen Ozeanariums umfaßt neben dem Inneren des Gebäudes auch dessen Außenansicht sowie das umgebende Expo-Gelände. Die Erzeugung dieser aufwendigen Modelle wurde am Centro Computatcção Graphica durchgeführt und dauerte 2 Jahre. Insgesamt umfaßt das Modell ca. 100000 Polygone. Damit diese auf dem zur Verfügung stehenden Grafikrechner[2] in Echtzeit dargestellt werden kann, wurde das Geländemodell teilweise mit sogenannten Level-Of-Detail Knoten modelliert. Dabei werden Gebäude- und Geländeteile in verschiedenen Auflösungen modelliert und diese dann zur Laufzeit abhängig von der Entfernung des Betrachters umgeschaltet. Mit der der Level-of-Detail Technik und ihrer Übertragung in die Simulation befaßt sich Abschnitt 6.2.11.

[2]SGI Onyx2 Infinite Reality, 4xR10000, 1G Ram

6.2.5 Modellstruktur

Das Modell wurde unter Verwendung verschiedener CAD- und Modellierungssysteme, u.A. FormZ, Softimage und Alias Wavefront erstellt. Durch Konverter wurde es in ein einheitliches Datenformat, das sogenannte Fraunhofer-Standard Format (FHS [FHS98]) übertragen. FHS ist ein Format für polygonale Geometrie und unterstützt texturierte Materialien, Lichtquellen und Level-Of-Detail. Die Szene wird dabei in Form eines hierarchischen Szenengraphen dargestellt. Den Szenengraphen des Virtuellen Ozeanariums stellt Abbildung 6.6 dar.

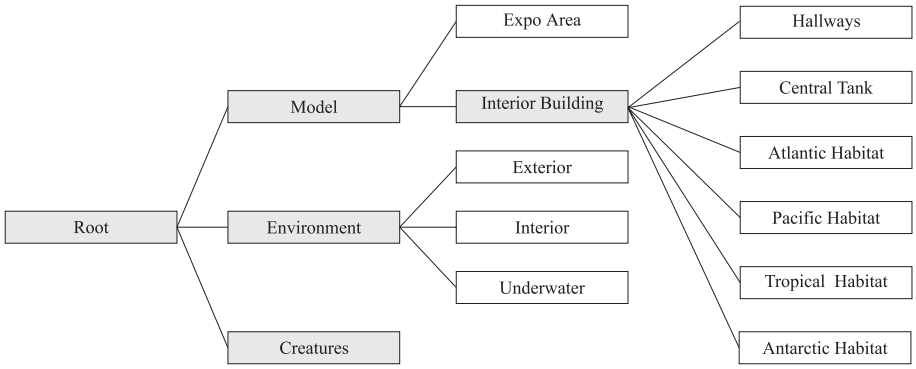

Abbildung 6.6: Szenengraph des Virtuellen Ozeanariums

Die Hierarchie läuft von links nach rechts, beginnend mit dem Wurzelknoten*Root*. Bei den grau unterlegten Knoten handelt es sich um reine Strukturelemente des Szenengraphen, die selbst keine Geometrie enthalten. Die weißen Knoten symbolisieren dagegen Teilgraphen mit weiteren Hierarchien und Geometrie.

Der Knoten *Modell* faßt das eigentliche grafische Modell zusammen. Der *Creatures* benannte Knoten dient zur Integration der virtuellen Kreaturen in den Szenengraphen. Unter ihn werden zur Laufzeit die Geometriemodelle der Tiere und Pflanzen des Virtuellen Ozeanariums angehängt, die nicht Teil der statischen Szene sind.

Environment faßt die verschiedenen Beleuchtungsmodelle zusammen, die je nach Position des Betrachters außerhalb oder innerhalb des Gebäudes sowie innerhalb der Aquarien umgeschaltet werden. Jeder der drei Knoten faßt dabei Lichtquellen sowie einen Knoten, der Nebel simuliert, zusammen. Mit Hilfe von Nebel mit blauer Farbe wird die Trübheit des Wassers dargestellt. Auf die Darstellung der Lichteffekte unter Wasser wird in Abschnitt 6.2.7 näher eingegangen. Während außerhalb des Gebäudes die Sonne scheint, ist es in den Gängen des Ozenariums relativ dunkel. Auf künstliche Beleuchtung

wird fast ganz verzichtet, Licht dringt nur durch die Scheiben der Aquarien ein. Unter Wasser kommt das Licht hauptsächlich von oben (durch das Glasdach und zusätzliche Scheinwerfer, die auch bei bedecktem Himmel Sonnenstrahlen simulieren). Im Wasser wird es gestreut und vielfältig reflektiert.

Der Rest der Szene ist in Außen- und Innenmodell aufgeteilt. Letzteres zerfällt in die 4 Habitats, den Haupttank sowie die Gänge und Treppenhäuser im Gebäude (*Hallways*).

6.2.6 Ablaufsteuerung

Eine Reise durch das Virtuelle Ozeanarium beginnt typischerweise mit einem Helikopterflug über das Gelände der Expo. Dazu steuert ein professioneller Moderator im Auditorium die virtuelle Kamera mit einer Spacemouse (siehe Abschnitt 5.2.1. Auch andere Interaktionstechniken wurden implementiert, wie beispielsweise die *Zeige-und-Fliege* genannte Technik, bei der die Kamera mit durch Zeigegesten mit einem Datenhandschuh oder einem speziellen Zeigegerät gesteuert wird (vergl. [Felg95].

Während des Fluges bekommen die Besucher einen Überblick über das Expo-Gelände, die wichtigsten Pavilions sowie das Gebäude des Ozeanariums.

Anschließend geht die Reise über die Brücke, die das im Wasser gebaute Hauptgebäude des Ozeanariums mit dem Festland verbindet in das innere des Gebäudes. Beim Eintritt in das Gebäudeinnere verändert sich die Beleuchtung drastisch. Die große Helligkeit des Tages verwandelt sich in die schummerige Beleuchtung in den Gängen des Ozeanariums. Auch die Klangkulisse ändert sich: Statt den Schreien der Möwen und der Hörner der vorbeifahrenden Schiffe herrscht nun Stille. Nach ein paar Schritten befindet sich das Auditorium vor einer gigantischen Akrylscheibe und blickt in den Global Ocean Tank (Abbildung 6.7).

Die Reise kann nun durch die Gänge in die benachbarten Habitats fortgesetzt werden. Die größte Attraktion ist aber, durch die Scheibe hindurch in den Haupttank hineinzutauchen. Das Auditorium ist von hunderten Fischen und anderen Meerestieren umgeben. Der Eindruck, sich unter Wasser zu befinden wird durch die simulierte Trübheit des Wassers und das Atemgeräusch eines Lungenautomaten verstärkt. In den terrestrischen Ebenen der Habitats wird tropische Feuchtigkeit oder arktische Kälte ebenfalls mit Nebel verschiedener Farben und Stärken simuliert. Schreie von typischen Vögeln, Brandung, Wind oder das Plätschern eines Baches verstärken den Immersionseindruck.

Um visuelle und akustische Effekte zu steuern, wurde ein Steuerungssystem entwickelt, welches über eine einfache Skriptsprache gesteuert wird.

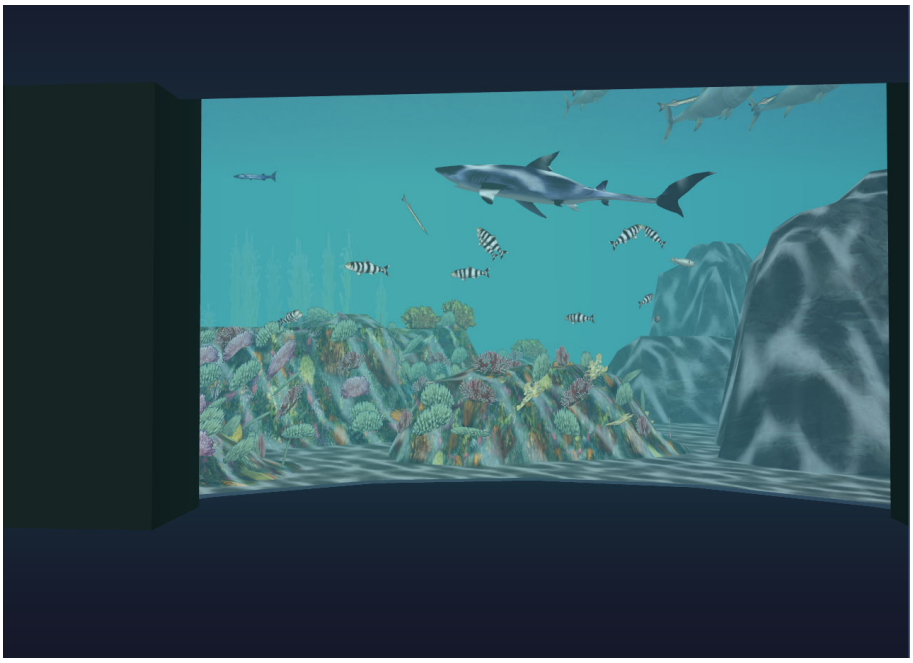

Abbildung 6.7: Blick in den Global Ocean Tank

Ein Skript besteht aus einer Reihe von Regeln die der folgenden Grammatik genügen:

```
<rule> ::= <condition> '->' <list-of-commands> '.'
<condition> ::= <term> [ '&' <term> ]
<list-of-commands> ::= <command> [ ',' <command> ]
<term> ::= [ '!' ] ( <variable> | <function> )
<variable> ::= <string-of-capital-letters>
<function> ::= <string-of-lowercase-letters> <quoted-string>
<command> ::= <string-of-lowercase-letters> <quoted-string>
<quoted-string> ::= '"' [ <any-char-except-double-qoute> ] '"'
```

Die Skriptsprache kennt ausschließlich binäre Variable und Funktionen mit binärem Rückgabewert. Zum Setzen und Löschen von Variablen existieren die Kommandos `set` und `clear`, die beide als Parameter den Namen der zu ändernden Variablen erwarten. Ein Satz weiterer Kommandos bildet eine Schnittstelle zum System und erlaubt:

- An- und Abschalten von Knoten und Lichtquellen im Szenengraphen,
- An- und Abschalten sowie Lautstärkeregelung von Sounds,

- Setzen von Parametern für Rendering und Stereoprojektion,
- An- und Abschalten der Simulation für virtuelle Kreaturen sowie
- die Ausgabe von Meldungen.

Weiterhin existiert eine Reihe von Funktionen, mit denen Regeln aufgebaut werden können. Sie erlauben die folgenden Tests:

- Frame Nummer: Vergleicht die im Parameter-String übergebene Nummer Zähler, der pro generiertes Bild weiter gezählt wird. Erlaubt damit die Ausführung von Regeln zur Initialisierung des Systems.
- Abstand: Prüft, ob der Abstand zwischen Betrachter und einem 3-Punkt in der Szene kleiner ist als der angegebene Wert. Koordinate und Abstand werden als Parameter übergeben.
- Sichtlinie: Zieht eine gedachte Linie zwischen aktueller Kameraposition und einem 3D-Punkt innerhalb der Szene. Liefert *true*, falls die Linie die in einem Teilgraphen der Szene nicht schneidet. Erwartet Namen des Teilgraphen und Koordinate als Parameter.

40 Regeln dieser Art genügen, um die Ablaufsteuerung für das Virtuelle Ozeanarium aufzubauen.

6.2.7 Simulation von Lichteffekten unter Wasser

Die Simulation von visuellen Effekten unter Wasser ist wichtig für die Glaubhaftigkeit der Darstellung. Wegen des Echtzeit Charakters der Simulation verbieten sich aufwendige physikalisch basierte Lichtsimulationen. Dennoch können mit Mitteln der Grafiksprache OpenGL realistische Lichteffekte erzielt werden.

Gerichtetes Licht

Zwei gerichtete Lichtquellen werden eingesetzt, um die Geometrie unter Wasser zu beleuchten. Eine Lichtquelle simuliert das von oben in das Becken einfallende Licht. Ein reduzierter Rotanteil simuliert dabei die Absorption von rotem Licht durch Wasser. Das vom Boden reflektierte Licht wird durch die zweite Lichtquelle dargestellt. Ihr fehlt der Rotanteil völlig, da das reflektierte Licht einen weiteren Weg zurücklegen muß und die Absorption dadurch größer wird.

Trübheit des Wassers

OpenGL erlaubt die Darstellung von Nebel. Dazu wird die Farbe jedes Pixels mit der Farbe des Nebels gemischt. Der Anteil des Nebels steigt dabei linear oder exponentiell abhängig vom Tiefenwert (Z-Buffer) des Pixels. Auf diese Weise wird erreicht, daß Objekte mit wachsender Entfernung nach und nach im Nebel verschwinden. Auf diese Weise können atmosphärische Effekte sowie die Trübheit von Wasser simuliert werden. Im realen Ozeanarium ist das Wasser so klar, daß man von einer Seite des Haupttanks bis zur anderen Seite schauen kann (Entfernung ca. 30 Meter), ohne eine starke Trübung wahrzunehmen. Für eine physikalisch korrekte Simulation wäre also der Einsatz von Nebel überflüssig.

Der Eindruck, sich unter Wasser zu befinden, wird aber durch den Einsatz von Nebel so wesentlich verstärkt, daß diese Ungenauigkeit in Kauf genommen wurde um den gewünschten Effekt zu erreichen. Eine Analogie hierzu ist der Sound, den Raumschiffe in Science-Fiction Filmen machen. Da sich im Vakuum kein Schall ausbreiten kann, werden im Weltall keinerlei Geräusche übertragen. Dennoch findet sich dieser "Fehler" in jedem praktisch Film.

Lichtbrechung an der Wasseroberfläche

Ein wichtiger Lichteffekt, der im flachen Wasser zu beobachten ist, werden durch die Brechung von oben einfallenden Lichtes in Wellen an der Wasseroberfläche verursacht. Es entstehen helle, sich kontinuierlich bewegende Muster auf dem Meeresgrund und auf der Oberfläche von im Wasser befindlichen Objekten.

Man spricht in diesen Zusammenhang von sogenannten *Caustics*. Solche Muster können mit verfügbaren Werkzeugen leicht berechnet werden. Dazu wird eine Reihe von Einzelbildern generiert, die hintereinander abgespielt, den gewünschten Effekt darstellen. Wird die Sequenz zyklisch angelegt, kann sie beliebig lange wiederholt werden. Abbildung 6.8 stellt eine Sequenz mit 8 Bildern dar, im Virtuellen Ozeanarium wurden 32 Bilder verwendet.

Ist eine solche Sequenz erstellt, kann sie als animierte Textur auf die Oberfläche von Boden, Steinen, Pflanzen und Tieren aufgebracht werden. Da allerdings alle genannten Objekte bereits eine Oberflächentextur besitzen, muß die Lichttextur als zweite Schicht über der ursprünglichen aufgebracht werden.

Abbildung 6.9 illustriert noch einmal die Wirkung der verschiedenen Lichteffekte im Haupttank des Ozeanariums.

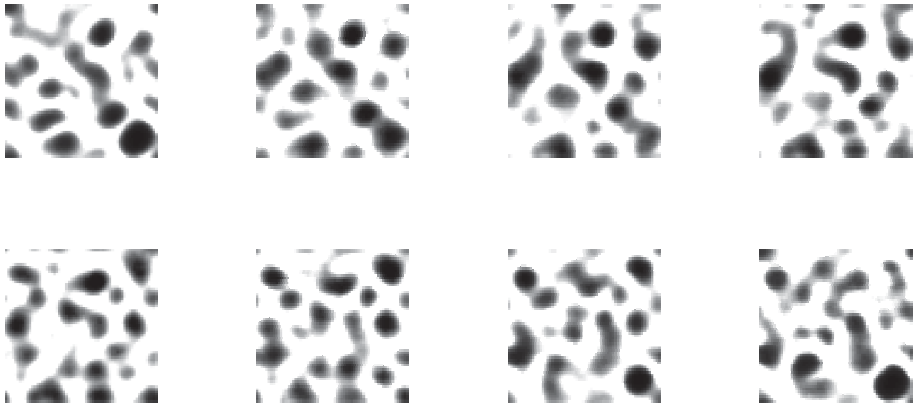

Abbildung 6.8: Sequenz von 8 Caustic-Texturen

6.2.8 Virtuelle Kreaturen

Für die Simulation von Fischen werden eine Reihe weiterer Feature-Klassen (siehe Kapitel simdyn) eingeführt.

Das Modell für die Lebewesen im Virtuellen Ozeanarium besteht aus folgenden Feature-Klassen:

Kinematic Sie erweitert Autonome Objekte um ein kinematisches Modell welches die Fortbewegung der Kreatur definiert.

Animation Führt die Schwimmbewegungen von Fischen und die Bewegung von Pflanzen in der Strömung durch Geometrie Animationen durch.

Behavior Bestimmt das Verhalten eines Fisches, wie etwa Kollisionsvermeidung, Schwarm- und Fluchtverhalten[3].

6.2.9 Simulationsmodell virtueller Fische

Bevor das Simulationsmodell im einzelnen vorgestellt wird, soll zunächst auf verwandte Ansätze eingegangen werden und die eigene Entwurfsstrategie begründet werden.

Den Stand der Wissenschaft stellen die Arbeiten von Terzopoulos und Xiaoyuan Tu dar. Hier wird ein realistisches physikalisch und biologisch fundiertes Simulationsmodell entwickelt und realisiert. Die Ergebnisse sind extrem realistisch. Faszinierend an diesem Ansatz ist neben der hohen Qualität der Simulation auch die Tatsache, daß die dort simulierten Fische ihr Verhalten in einer Lernphase erwerben.

[3]Tatsächlich werden auch können auch Tiere simuliert werden, die nicht zur Gruppe der Fische gehören, etwa Delphine und Meeresschildkröten.

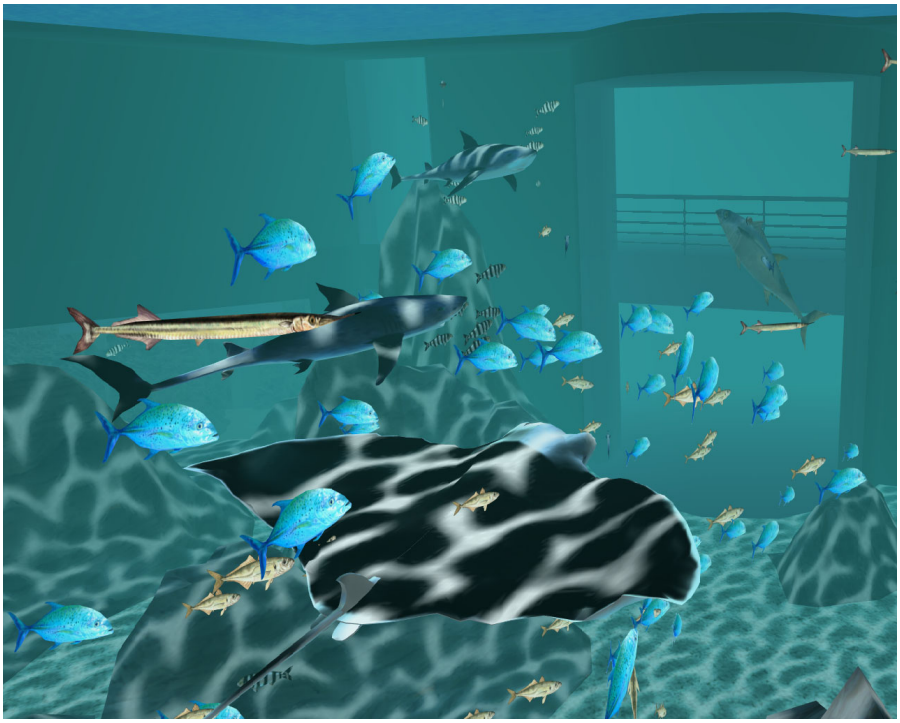

Abbildung 6.9: Lichteffekte im Haupttank des Virtuellen Ozeanariums

Der Körper jedes Fisches wird durch ein Feder-Masse-Modell und eine polygonale Hülle dargestellt. Das Feder-Masse-Modell besteht aus ca. 20 Massepunkten und Federn. Einige Federn können dabei von einem übergeordneten Motorik-Modul analog zur Funktion von Muskeln zur Kontraktion gebracht werden. Auf den Körper des simulierten Fisches wirkt das umgebende Wasser durch eine ebenfalls in der Simulation angelegte Hydrodynamik. Die bei der Bewegung verbrauchte Energie und die resultierenden Kräfte werden berechnet und der Fisch so realistisch bewegt. Das polygonale Modell des Fisches wird verformt und dessen Bewegung im Raum berechnet. In der Lernphase wird eine erste Generation von Fischen mit zufällig programmierten Motorik-Modulen generiert und ihre Schwimmbewegungen simuliert. Die Ergebnisse sind wie zu erwarten zunächst unkoordiniert und dementsprechend ineffizient. Ineffizient meint in diesem Zusammenhang energieaufwendig bei geringer Fortbewegung. Für die nächste Generation werden die Fische mit der effizientesten Bewegung ausgesucht und "vermehrt". Zusätzlich wird wird die Programmierung der Motorik entsprechend einer simulierten Mutation zufällig verändert. Nach und nach "lernen" die Fische auf diese Weise sich bei möglichst geringem Energieaufwand durch das Wasser zu bewegen.

Die Wahrnehmung der Fische wird, wie in der vorliegenden Arbeit, durch einen einzelnen simulierten Sinn abgebildet. Dieser entspricht weitgehend dem Sehsinn und wird durch Raycasting realisiert.

Die Resultate sind überzeugend simulierte virtuelle Fische in einer hochqualitativen grafischen Darstellung. In Echtzeit (also mit Simulationsraten von mindestens 15 Schritten pro Sekunde) können allerdings wegen dem hohen Rechenaufwand nur wenige Fische gleichzeitig dargestellt werden.

Für die vorliegende Arbeit stand aber nicht eine naturwissenschaftlich fundierte Simulation im Vordergrund, sondern die Darstellung einer reichhaltigen Unterwasserwelt mit hunderten von Fischen. Dies ist mit den genannten Verfahren bei dem aktuellen Stand der Technik nicht realisierbar. Der Ansatz der vorliegenden Arbeit besteht daher in stark vereinfachten Simulationsmodellen. Diese werden so gestaltet, daß sie möglichst realistisch *aussehen*, sich dabei aber so schnell durchrechnen lassen, daß eine Darstellung in Echtzeit möglich wird.

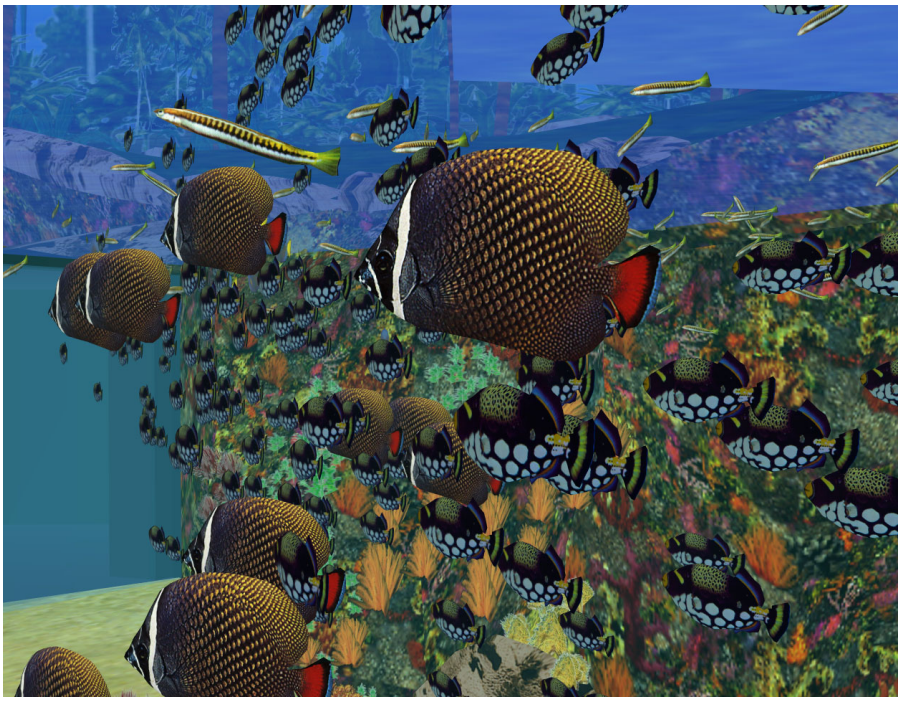

Abbildung 6.10: Farbenfrohe Fischschwärme im tropischen Habitat

Das Diagramm in Abbildung 6.11 stellt das Simulationsmodell eines virtuellen Fisches dar. Ein virtueller Fisch besteht aus einer Reihe von Modulen (im Diagramm als Scheiben dargestellt), die einzelne Komponenten des Organismus darstellen. Zwischen diesen fließen Daten, durch Pfeile symbolisiert.

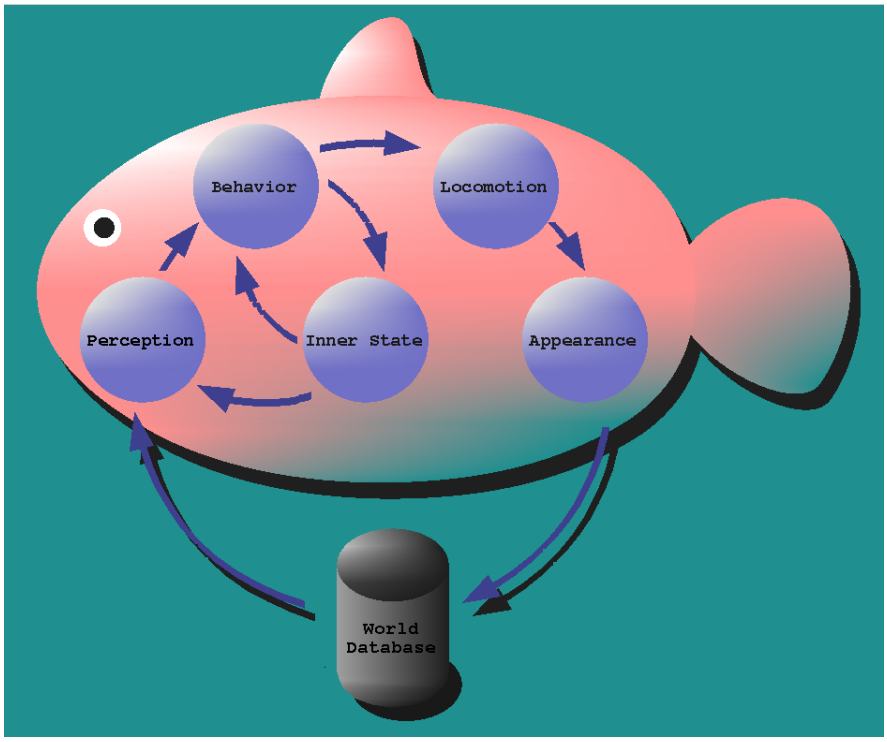

Abbildung 6.11: Modell einer virtuellen Kreatur

Die Simulation des Verhaltens stellt einen Regelkreis dar. In jedem Simulationsschritt wird dieser Kreislauf einmal durchlaufen. Nacheinander wird zunächst die Wahrnehmung (*Perception*), das Verhalten (*Behavior*), die Änderung der inneren Zustände (*Inner State), die Fortbewegung (*Locomotion*) und schließlich die grafische Darstellung (*Appearance) berechnet. Schließlich wird die Welt-Datenbasis aktualisiert.

Wahrnehmung

Reale Fische nehmen ihre Umwelt auf vielfältige Weise war. Nach Q. Bone und N.B. Marshall [BoMa85] nehmen sie die folgenden Reize wahr:

- optische,
- akustische und
- elektrische Reize,
- Vibrationen,

- Geruch und Geschmack sowie
- Pheromone (Hormonelle Reize)

Wie man sehen kann, ist die Vielfalt der wahrgenommenen Reize sehr groß. Für ein realistisches Modell müßte die Ausbreitung von Licht, Schall und elektromagnetischen Feldern im Wasser simuliert werden. Für die Nachbildung von Geruchs- und Geschmackssinn sowie der Wahrnehmung hormoneller Reize wäre zusätzlich die Ausbreitung von Substanzen im Wasser zu simulieren. Die Arbeit von Xiaoyuan Tu [Tu96] berücksichtigt ausschließlich den visuellen Sinn, dies allerdings sehr aufwendig. Für jeden Fisch wird der Sichtbereich in ein Bild gerendert und dies anschließend durch die simulierte Wahrnehmung analysiert. Für die vorliegende Arbeit kommt ein solcher Ansatz jedoch nicht in Frage, da er für eine große Zahl zu simulierender Fische und eine realistische grafische Darstellung für eine Simulation in Echtzeit um Größenordnungen zu aufwendig ist.

Daher wurde ein stark vereinfachtes Modell entwickelt, welches sich ebenfalls im Wesentlichen an der optischen Wahrnehmung orientiert. Da die Bereiche, in denen Fische Vibrationen und elektrische Reize wahrnehmen sich mit dem Bereich der optischen Wahrnehmung überschneiden, können diese in grober Näherung mit abgedeckt werden[4].

Wie Abbildung 6.12 skizziert, besitzt das Modell zwei Parameter, den Sichtradius und den Sichtwinkel. Diese können für verschiedene Spezies angepaßt werden. Nur Objekte innerhalb des so definierten Bereiches werden wahrgenommen.

Bei der Simulation der Wahrnehmung wird zwischen statischen und dynamischen Objekten unterschieden. Dynamische Objekte sind alle Fische im Virtuellen Ozeanarium, die gesamte restliche Szenengeometrie (Boden, Steine, Korallen und Pflanzen, etc.) wird als statisch betrachtet.

Aus technischer Sicht kann der Vorgang der Wahrnehmung als Anfrage an die Welt-Datenbasis betrachtet werden. Die Anfrage wird mit mit der Position und Richtung der Kreatur sowie mit deren Sichtradius und -Winkel parametriert. Als Antwort liefert die Welt-Datenbasis eine Liste aller im Sichtradius befindlichen dynamischen Objekte zurück. Diese Liste wird an das Verhaltens-Modul weitergereicht und bestimmt dort die Auswahl des adäquaten Verhaltens mit.

Statische Objekte stellen potentielle Hindernisse für die Navigation dar. Die gewünschte Information besteht darin, ob ein bestimmter Punkt für die

[4]Bei der Schwarmbildung spielt die Wahrnehmung von Vibrationen durch die Seitenlinienorgane eine wichtige Rolle. Durch diese kann die Entfernung und Richtung benachbarter Fische wahrgenommen werden.

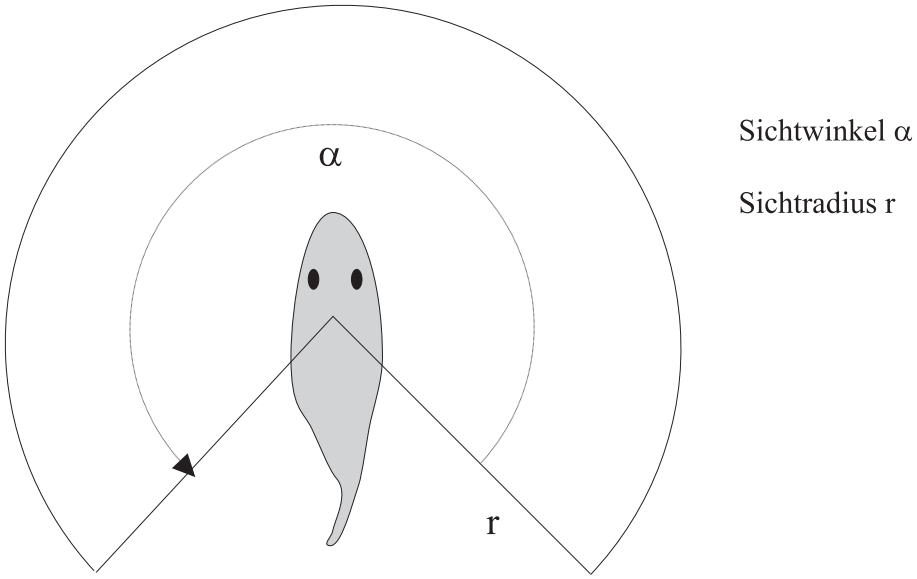

Abbildung 6.12: Modell der Wahrnehmung eine Virtuellen Fisches

Navigation frei oder durch ein Hindernis blockiert ist. Die Anfrage wird also mit der zu untersuchenden Raumkoordinate parametriert und liefert als binäre Information, ob diese Stelle frei oder blockiert ist.

Es liegt auf der Hand, daß die Implementierung der Welt-Datenbasis und der Funktionen, die das Wahrnehmungsvermögen der virtuellen Kreaturen realisieren, eine zentrale Bedeutung für die Performanz des Systems besitzen, da diese extrem häufig ausgeführt werden. Diese Aspekte werden in Abschnitt 6.2.9 diskutiert.

Verhalten und innere Zustände

Der Wahrnehmung nachgeschaltet ist das Verhaltensmodul. Es verarbeitet die von der Wahrnehmung generierten Stimuli. Dabei handelt es sich um Listen von Positionen anderer Fische, sowie Informationen darüber, ob sich an einer bestimmten Position ein Hindernis oder freies Wasser befindet. Auch hier wird wieder ein stark vereinfachtes Modell gewählt, welches einerseits das Verhalten der Fische glaubhaft darstellt, andererseits aber die Simulation in Echtzeit zuläßt.

Die wichtigste Aufgabe des Verhaltensmoduls ist die Navigation. Diese wird von folgenden Faktoren bestimmt, die nach ihrer Priorität geordnet aufgezählt werden:

1. Kollisionsvermeidung

2. Schwarmverhalten

3. Jagd- und Fluchtverhalten

4. Neugier

Vermeidung von Kollisionen mit statischen und dynamischen Hindernissen ist das wichtigste Ziel der Navigation. Können Kollisionen nicht vermieden werden, wird der Eindruck der grafischen Simulation für den Betrachter sofort unglaubwürdig. Ein zunächst berechneter Kurs muß auf seine Kollisionsfreiheit hin untersucht werden. Liegt eine Kollision vor, wird der Kurs zunächst in einer Reihe von Schritten zunächst leicht und dann immer stärker variiert. Kann auf diese Weise die Kollision nicht vermieden werden, wird der Kurs verworfen und ein Ausweichmanöver eingeleitet.

Schwarmverhalten ist bei vielen Tierarten zu beobachten. Dies gilt nicht nur für Fische, sondern auch für viele Vogel- und Landtierarten (Herden). Schwarmverhalten ist seit Jahren Gegenstand der Forschung sowohl im Bereich der Computergrafik und Künstlichen Intelligenz [Reyn87],[Tu96], [TuTe94] als auch in der Biologie [Part82].

Jagd- und Fluchtverhalten ist ebenfalls bei praktisch allen Tierarten zu beobachten. Im Falle der Fische in einem Aquarium spielt es aber eine untergeordnete Rolle. Zum einen werden die Fische ständig ausreichend gefüttert, sodaß selbst Raubfische praktisch nie jagen. Daher kommt auch Fluchtverhalten kaum vor. Das in dieser Arbeit vorgestellte System ist jedoch nicht darauf beschränkt, die Situation in einem Aquarium zu simulieren, sondern soll sich auch für Biotope in Seen oder im Meer eignen. Auch spielt der Taucher eine besondere Rolle. Seine Anwesenheit kann Fische zumindest zu Fluchtverhalten veranlassen, wenn er sich diesen zu sehr nähert oder sich hektisch bewegt. Das der Taucher Opfer von Raubfischen wird, wurde aus naheliegenden Gründen (Präsentation des Systems auf der EXPO) ausgeschlossen.

Fortbewegung

Wurde durch das Modul für die Simulation von Verhalten ein Ziel für die Fortbewegung festgelegt, muß der Fisch nun ausgehend von seiner aktuellen Position und Orientierung schrittweise darauf hin bewegt werden. Die Bewegung sollte möglichst natürlich wirken. Dazu sind zwei Aufgaben sind zu lösen:

- Bewegung des Fisches entlang einer Kurve im dreidimensionalen Raum (Pfadanimation).

- Simulation der Schwimmbewegungen durch Verformung des polygonalen Modells (Geometrieanimation)

Die Aufgabe, virtuelle Kreaturen Ozeanarium entlang von Pfaden zu animieren übernimmt das Feature *Kinematic*, die Durchführung der Geometrieanimation wird durch das Feature *Animation* realisiert.

Für die Pfadanimation muß eine geeignete Kurvenform gewählt werden. Für die vorliegende Arbeit wurden hierzu Hermite-Kurven herangezogen. Sie besitzen die folgenden günstigen Eigenschaften, die sie für eine Verwendung prädestinieren:

1. Die Kurve wird durch Start- und Endpunkte definiert, zwei Kontrollvektoren bestimmen dort die Tangenten.

2. Die Kurve interpoliert zwischen Start- und Endpunkten, verläuft also exakt durch diese.

3. Hermite-Kurven sind als kubische Funktionen über einen Laufparameter t gegeben und lassen sich mit geringem Aufwand berechnen.

4. Die Kurve ist C^2-stetig, besitzt also die gewünschte organische Form.

Hermite Kurven werden in der Parameterdarstellung

$$p(t) = G * H * T$$

geschrieben. Die Geometriematrix $G = (P_1, P_2, D_1, D_2)$ enthält die Kontrollpunkte der Kurve, nämlich Start- und Endpunkt sowie die Tangenten am Start- und Endpunkt. Die Basismatrix H enthält die Polynominalkoeffizienten und der Vektor T $T = (t^3 t^2 t 1)^T$ die Potenzen des Kurvenparameters t.

Abbildung 6.13 stellt eine typische Bewegungskurve dar, die aus zwei Hermite-Kurvensegmenten besteht. Der Fisch bewegt sich entlang dieser Kurve, indem der Parameter t das Intervall $[0, 1]$ durchläuft. Anschließend erzeugt das Verhaltensmodul ein neues Ziel, und der Bewegungszyklus wird erneut durchlaufen.

Bevor jedoch die Bewegung tatsächlich ausgeführt werden kann, muß die Kurve zunächst auf Kollisionsfreiheit überprüft werden. Hierzu kann die Gitterdarstellung der Welt-Datenbasis herangezogen werden. Die Kurve wird zunächst rekursiv tesseliert. Die Unterteilung der Kurve kann abgebrochen werden, wenn die Abstände der Stützpunkte kleiner sind als die Abstände der Gitterpunkte. Befindet sich mindestens ein Stützpunkt innerhalb einer

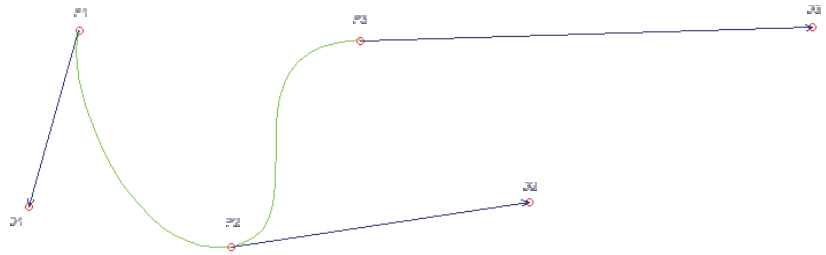

Abbildung 6.13: Hermitekurve aus zwei Segmenten

Zelle, die als Hindernis markiert ist, muß die Kurve so modifiziert werden, daß sie nur durch freie Zellen verläuft.

Hierzu wurde ein einfaches stochastisches Verfahren herangezogen. Zielpunkt und Zielrichtung werden solange zufällig modifiziert, bis die Kurve kollisionsfrei ist. Dabei wird der Wertebereich der Modifikation in jedem Schritt vergrößert. Das Verfahren versucht also zunächst durch kleine Modifikationen das Ziel noch zu erreichen, ist dies nicht möglich werden nach und nach auch größere Abweichungen in Kauf genommen.

Grafische Darstellung

Die grafische Darstellung virtueller Kreaturen erfolgt über polygonale Modelle. Deren Visualisierung übernimmt der dem System zugrunde liegende Y-Renderer. Für jede Spezies existiert eine Reihe von Varianten mit unterschiedlicher Anzahl von Polygonen (Level-Of-Detail). Je nach Abstand vom Betrachter, wird eine dieser Varianten ausgewählt und vom Renderer visualisiert. Das Verfahren wird in Abschnitt 6.2.11 weiter diskutiert.

Um die Schwimmbewegungen von Fischen darzustellen, werden Geometrieanimationen verwendet. Dazu wird die Verformung des Tierkörpers zunächst in einem Animationssystem modelliert. Anschließend werden einzelne "eingefrorene" Stellungen (die sogenannten "Keyframes") in Form von polygonalen Modellen als Dateien abgelegt. Beim Start des Systems werden diese geladen und als *Appearance Patterns* den Spezies zugeordnet. Zur Laufzeit wird die Position jedes Scheitelpunktes durch lineare Interpolation zwischen zwei Keyframes neu berechnet.

Die Ablaufgeschwindigkeit der Animation ist linear abhängig von der aktuellen Geschwindigkeit und Beschleunigung der individuellen Kreatur. Die entsprechenden Gewichtungsfaktoren werden in der Beschreibung der jewei-

ligen Spezies festgelegt.

Für einige Fische (etwa Hai und Thunfisch) kann die Tatsache ausgenutzt werden, daß die Bewegungen nicht nur zyklisch, sondern spiegelsymmetrisch zur gestreckten Lage des Fischkörpers sind. Auf diese Weise muß lediglich die Hälfte der Keyframes im Speicher gehalten werden.

Zwar ist die Geometrieanimation im Vergleich zu physikalisch basierten Bewegungssimulation relativ performant, dennoch übersteigt deren gleichzeitige Durchführung für hunderte von Fischen die Leistungsfähigkeit der zur Verfügung stehenden Grafikrechner bei weitem. Daher wird diese im Virtuellen Ozeanarium nur für große, besonders wichtige Spezies durchgeführt, deren Population vergleichsweise klein ist. Dies sind Hai, Thunfisch, Mantarochen, Meeresschildkröte und Igelfisch. Auch muß die Geometrieanimation nur dann tatsächlich ausgeführt werden, wenn der Fisch tatsächlich sichtbar ist. Zwar wird die Berechnung der Animationsparameter in jedem Fall ausgeführt, die Neuberechnung der Scheitelpunkte kann aber ohne Verlust an Realismus entfallen. Auch werden die Schwimmbewegungen nur dann simuliert, wenn sich der Fisch nahe genug am Betrachter befindet, sodaß diese überhaupt wahrgenommen werden.

Die Welt-Datenbasis

Die Welt-Datenbasis steht am Anfang und am Ende des in Abbildung 6.11 dargestellten Regelkreises.

Am Anfang steht dabei die Wahrnehmung, am Ende deren Aktualisierung. Die Aktualisierung besteht im wesentlichen darin, die virtuellen Kreaturen zu bewegen und die Geometrieanimation für die Schwimmbewegungen durchzuführen.

Um die Wahrnehmung dynamischer Objekte zu simulieren, muß zunächst festgestellt werden, ob sie sich im definierten Wahrnehmungsbereich befinden. Abstand und Winkel müssen berechnet und mit den Parametern verglichen werden. Betrachtet man die Gesamtzahl der Fische n ergeben sich $n^2/2$ Berechnungen. Der Aufwand wächst quadratisch mit der Zahl der simulierten Fische und ist sofort als Flaschenhals zu erkennen. Für die statischen Objekte muß die Szenengeometrie mit dem Sichtbereich jedes Fisches geschnitten werden. Näherungsweise kann stattdessen ein Raycastingverfahren eingesetzt werden. Dabei wird der Wahrnehmungsbereich mit einem Fächer von "Sehstrahlen" abgetastet. In jedem Fall wächst der Aufwand linear mit der Zahl der Fische und der Komplexität der Szenengeometrie. Für die aufwendige grafische Darstellung der Szene im virtuellen Ozeanarium sind beide Verfahren auf dem heutigen Stand der Technik nicht in Echtzeit praktikabel.

Um die oben diskutierten Probleme zu lösen, wird die Welt-Datenbasis in

würfelförmige Zellen eingeteilt, wie dies in Abschnitt 3.13 bereits beschrieben wurde.

Für die Simulation der Wahrnehmung dynamischer Objekte bildet das Voxelgitter ebenfalls die Grundlage. Jedes dynamische Objekt wird zur Laufzeit des Systems einer Zelle in diesem Gitter eindeutig zugeordnet. Ausschlaggebend ist dabei die Position des Mittelpunktes der Bounding Box[5] des Objektes. Jede Gitterzelle enthält eine Liste aller ihr aktuell zugeordneten dynamischen Objekte. Verläßt ein Objekt den Bereich einer Zelle, wird die Referenz auf dieses Objekt aus der Liste entfernt und in die der neuen Zelle eingetragen. Abbildung 6.14 stellt die beschriebene Datenstruktur noch einmal grafisch dar. Zu sehen ist der Haupttank des Virtuellen Ozeanariums und mehrere Fischschwärme. Rot dargestellt sind die Gitterzellen, blau die Bounding Boxen der einzelnen Fische. Man sieht, daß die statische Szenengeometrie vollständig innerhalb der Zellenstruktur liegt.

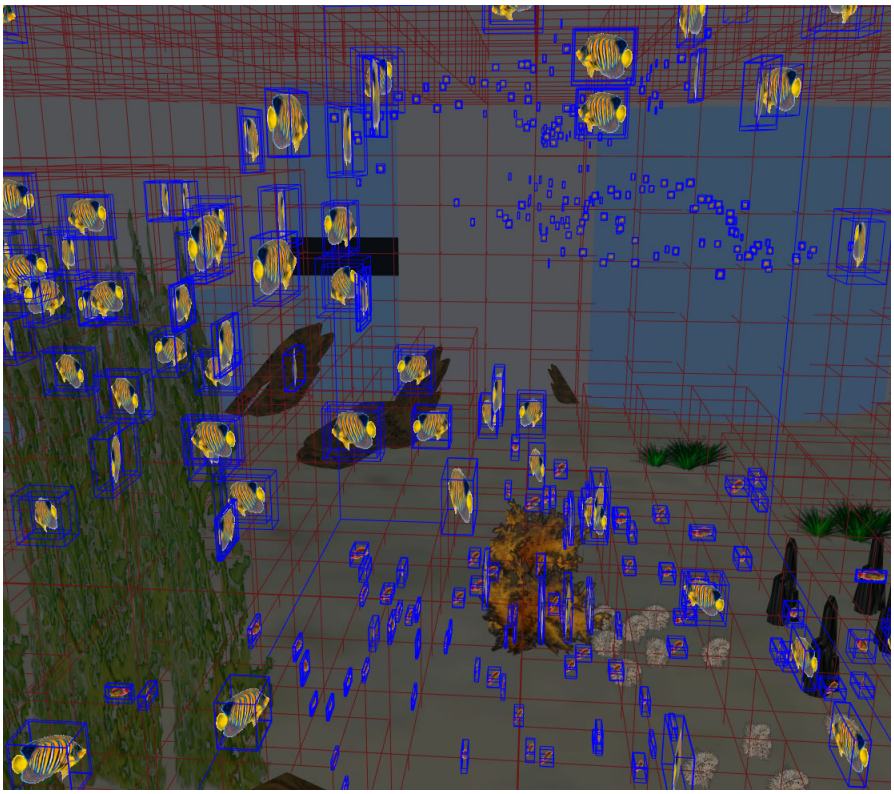

Abbildung 6.14: Einteilung der Datenbasis in ein regelmäßiges Gitter

[5] Quaderförmiges Hüllvolumen an den Koordinatenachsen des Weltkoordinatensystems ausgerichtet.

Potentiell wahrnehmbar sind nun nicht mehr alle dynamischen Objekte sondern lediglich die, die sich in den umgebenden Zellen eines Fisches befinden. Der Rechenaufwand wächst dadurch nicht mehr quadratisch mit der Gesamtzahl der Fische im Virtuellen Ozeanarium sondern nur noch linear mit der mittleren Populationsdichte N/V in den Fischtanks. N ist wiederum die Gesamtzahl der Fische, V ist das (wassergefüllte) Volumen des Aquariums. Auf diese Weise Weise werden Simulationsraten von 15 fps (Frames Per Second) und mehr für mehrere hundert Fische erreicht[6]. Durch Parallelisierung der Simulation und das Aura-Konzept konnte diese bis auf über tausend gesteigert werden (siehe Abschnitte 6.2.10 und 6.16).

6.2.10 Parallelisierung der Simulation

Da für die Realisierung des Virtuellen Ozeanariums eine Grafikworkstation mit mehreren Prozessoren und Grafik-Subsystemen zur Verfügung stand[7], lag es nahe das System zu parallelisieren, um die zur Verfügung stehende Grafik- und Rechenleistung optimal auszunutzen und so die Geschwindigkeit der Simulation resp. die Zahl der simulierten Fische zu erhöhen.

Hierzu konnte die Funktionalität des Simulationssystems ausgenutzt werden (siehe Abschnitt 3.11). Das System wird so konfiguriert, daß die Simulation auf drei parallele Prozesse verteilt abläuft. Die weiteren Prozessoren führen die Ablaufsteuerung, die Geräteanbindung (IDEAL) und die grafische Darstellung aus.

Durch der Parallelisierung der Simulation ist das System nun in der Lage, mehr als 10000 Fische pro Sekunde zu simulieren und grafisch darzustellen. Ein System von über 1000 Fischen ist somit in Echtzeit darstellbar.

6.2.11 LOD in der Simulation

Level-Of-Detail (LOD) Techniken sind in der Computergrafik seit einigen Jahren Stand der Technik. Der Grundgedanke dabei ist, daß weiter vom Betrachter entfernte Objekte mit geringerer polygonaler Auflösung (Level Of Detail) dargestellt werden können, da sie bei der grafischen Darstellung auf eine kleiner Fläche abgebildet werden und feine Details kaum noch sichtbar sind.

Man unterscheidet zwischen statischen und dynamischen LOD-Verfahren.

[6]Haupttank des Virtuellen Ozeanariums, SGI Onyx R10000/190Mhz.

[7]SGI Onyx 6xR10000, 3 Infinite Reality (IR) Pipes und Onyx2 4xR10000, 2 IR Pipes

Statische LOD-Verfahren

Bei statischen LOD-Verfahren wird eine Reihe von vorberechneten Varianten desselben Objektes mit abnehmender polygonaler Auflösung vom Renderer vorgehalten. Abhängig von der aktuellen Entfernung zwischen Objekt und Betrachter wird bei der grafischen Darstellung die passende Variante ausgewählt. Bei visuellen Simulationen spielt bei der Auswahl des LOD zusätzlich die einzuhaltende Bildrate bei der grafischen Darstellung eine Rolle. Der Renderer mißt hierzu laufend die aktuell erreichte Bildrate (R_i) und vergleicht diese mit der vorgegebenen Rate (R_s). Der Quotient $S = R_s/R_i$ bildet den sogenannten Streßfaktor. Für $S > 1$ läuft das System zu langsam, die Zahl der Polygone muß durch Auswahl eines geringeren LOD reduziert werden. Für $S < 1$ kann die Auflösung erhöht werden. Auf diese Weise wird die Abwägung zwischen Geschwindigkeit und Genauigkeit der Darstellung von der Applikation bzw. dem Anwender in die Hand gegeben. Leistungsfähigere Grafikhardware wird ohne Umstellung automatisch optimal ausgenutzt.

Nachteilig an diesem Verfahren ist der sogenannte "Pop" Effekt. Dieser Effekt kann vom Betrachter wahrgenommen werden, wenn zwischen zwei LOD umgeschaltet wird, und plötzlich Details des Objektes hinzukommen oder verschwinden. Dieser kann weitgehend verhindert werden, wenn die Umschaltung weit genug entfernt vom Betrachter durchgeführt wird, was aber andererseits zu Lasten der Performanz der grafischen Darstellung geht. Weiter muß verhindert werden, daß in schneller Folge zwischen zwei LOD umgeschaltet wird, wenn sich der Betrachter in der Nähe der Umschaltentfernung hin und her bewegt. Dies kann mit einem sogenannten Hystereseverhalten (Verzögerung des Umschaltvorgangs in beide Richtungen) bei der Umschaltung umgangen werden.

Dynamische LOD-Verfahren

Dynamische LOD-Verfahren sind in Hinblick auf die genannten Artefakte weniger problematisch, bei der grafischen Darstellung aber aufwendiger. Man unterscheidet zwischen dynamischer Tesselierung und Progressive-Mesh Verfahren.

Bei der dynamischen Tesselierung wird das grafische Objekt als Menge gekrümmter Flächen (beispielsweise NURBS) dargestellt. Für die grafische Darstellung wird es zur Laufzeit in eine polygonale Repräsentierung umgewandelt (tesseliert). Die Genauigkeit der Tesselierung (d.h. die Zahl der generierten Polygone) wird wieder von der Entfernung des Objektes zum Betrachter abhängig gemacht. Auf diese Weise ist theoretisch immer die optimale Darstellung gewährleistet. Allerdings ist der Tesselierungsvorgang viel zu

aufwendig, um pro Objekt und Bild zur Laufzeit durchgeführt zu werden. Die Tesselierung wird daher seltener ausgeführt, etwa wenn sich die Entfernung signifikant verändert hat. Einmal berechnete LOD können vom Renderer zur späteren Verwendung aufbewahrt werden (Caching).

Bei den Progressive Meshes liegt das grafische Objekt in einer hierarchischen polygonalen Darstellung vor. Abhängig von einem dynamischen Gütefaktor wird diese mehr oder weniger weit durchlaufen und entsprechend mehr oder weniger Polygone für die grafische Darstellung erzeugt. Der Gütefaktor ist wiederum eine Funktion der Entfernung zum Betrachter.

LOD für die Darstellung virtueller Kreaturen

Für die grafische Darstellung der virtuellen Kreaturen wurde das statisches LOD-Verfahren verwendet, da dies der dem Virtuellen Ozeanarium zugrunde liegende Y-Renderer zur Verfügung stellt.

Die Körper der Fische wurden zunächst als NURBS-Modelle modelliert und daraus mehrere verschieden detaillierte polygonale Darstellungen generiert und in getrennten Dateien abgelegt. Die Konfigurationsdateien des Virtuellen Ozeanariums enthalten für jedes Spezies Anzahl, Dateinamen und die Entfernung für die Umschaltung der LOD. Zusätzlich kann ein Hysteresefaktor angegeben werden. Geometrieanimationen werden nur für den höchsten LOD durchgeführt. Um Sprünge bei der Umschaltung eines animierten LOD in einen niedrigeren statischen LOD zu vermeiden, wird die Umschaltung solange verzögert, bis die Animation eine neutrale Phase durchläuft. Wird zurück in den höchsten LOD geschaltet, beginnt die Animation wieder dort.

Dies soll Abbildung 6.15 noch einmal am Beispiel eines Mantarochens illustrieren. In der oberen Zeile sind die LOD in absteigender Reihenfolge dargestellt. Die linke spalte zeigt drei der insgesamt 15 Animationsphasen (Keyframes). Das Bild links oben bildet als neutrales Keyframe den Übergang zwischen dem nächst niedrigeren LOD und der Animation.

Für besonders kleine Fische, die in großen Schwärmen im Virtuellen Ozeanarium auftreten (Makrelen, Pilotfische, etc.), wurde für den niedrigsten LOD die einfachstmögliche Darstellung durch ein einziges texturiertes Polygon gewählt. Um die Textur zu gewinnen, wurde der Fisch zunächst in höchster Auflösung von der Seite gerendert und der Bildausschnitt als Bilddatei abgelegt. Anschließend wurden mit einem Bildbearbeitungsprogramm alle Hintergrundpixel gelöscht und als transparent markiert. Es liegt auf der Hand, daß solcherart dargestellte Objekte verschwinden, wenn sie der Betrachter genau von vorn oder oben betrachtet. In der Praxis ist dieser Effekt aber aus zwei Gründen kaum wahrnehmbar:

Erstens sind weit entfernte Fische durch die Trübheit des Wassers nur

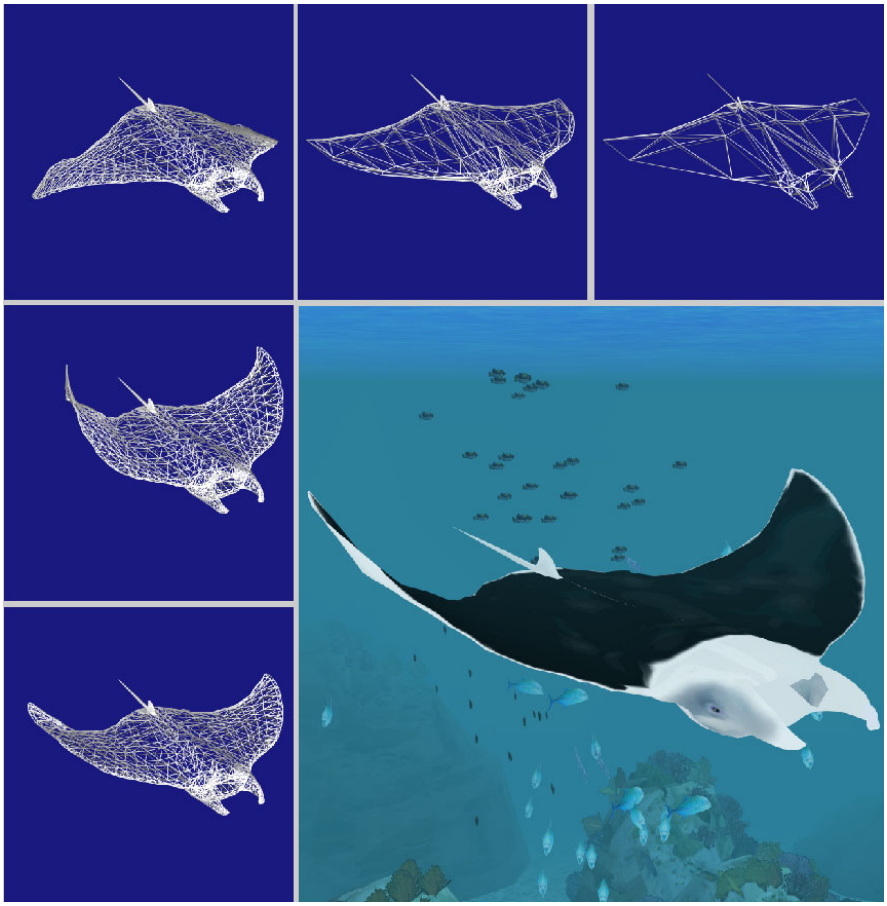

Abbildung 6.15: Verschiedene Level Of Detail und Keyframes eines Mantarochens

schemenhaft zu erkennen (und nur diese werden als texturierte Polygone dargestellt) und zweitens ist die Körperform der Fische von vorn betrachtet hoch und schmal (siehe beispielsweise Abbildung 6.10 auf Seite 127).

Die Auflösung der Modelle liegt insgesamt zwischen 12000 (Mantarochen, höchster LOD und 2 Dreiecken.

LOD in der Simulation virtueller Kreaturen

Wie bereits dargelegt wurde, hängt die die Performanz des des Gesamtsystems wesentlich davon ab, daß grafische Darstellung und Simulation ungefähr gleich schnell ablaufen. Nachdem nun die grafische Darstellung durch LOD-Techniken beschleunigt werden konnte, sollen diese nun auch auf die Simulation des Verhaltens angewendet werden, um daß System optimal aus-

zubalancieren.

Wie bereits in Abschnitt 6.2.9 beschrieben, wird das Verhalten der virtuellen Kreaturen in zwei Ebenen simuliert. Auf der oberen Ebene findet die Simulation des biologischen Verhaltens statt, welches durch Wahrnehmung und innere Zustände der Kreatur determiniert wird. Die visuelle Simulation durch Pfadanimationen und Geometrieinterpolation bildet die untere Ebene. Auf beiden Ebenen können LOD-Verfahren angewendet werden, um die Simulation zu beschleunigen. In beiden Fällen wird hierbei das Zeitraster der Simulation mit dem Abstand der virtuellen Kreatur gröber.

6.2.12 Quantitative Simulation

Das in den vorangegangenen Abschnitten entwickelte Simulationssystem behandelte jedes individuelle Lebewesen als Software-Objekt, dessen Wahrnehmung und Verhalten diskret simuliert wurde. Dieses Vorgehen ist typisch für visuelle Simulationen wie Verkehrs- und Flugsimulatoren oder eben das Virtuelle Ozeanarium. Der Aufwand für die Berechnungen steigt dabei mindestens linear mit der Zahl der simulierten Einheiten. Für die Simulation von komplexen natürlichen Systemen, wie etwa einem ganzen Küstenabschitt mit all den dort vorkommenden Lebewesen ist ein solches System daher ungeeignet, da der zu betreibende Rechenaufwand schnell über alles technisch machbare hinaus wächst.

Um sich solchen Systemen dennoch zu nähern, kann die visuelle Simulation durch ein anderes Verfahren, die sogenannte quantitative Simulation ergänzt werden. Dieser Begriff stammt aus dem Bereich der Wirtschaftswissenschaften und der Biologie. Gegenstand der quantitativen Simulation sind Mengen von gleichartigen Einheiten, etwa die Anzahl von Waren eines bestimmten Typs oder von Tieren einer Spezies. Nicht individuelle Tiere, sondern Populationen von Spezies werden dargestellt.

Dementsprechend ist der Aufwand unabhängig von der Zahl der Tiere und wird stattdessen von der Komplexität des zugrunde liegenden Regelwerkes bestimmt.

Im nächsten Abschnitt wird zunächst ein einfaches quantitatives Modell vorgestellt und anschließend seine Integration in die visuelle Simulation diskutiert.

Das Lotka-Volterra Modell

Das Lotka-Volterra Modell wurde unabhängig voneinander von dem Amerikaner Lotka [Lotk25]und dem Italiener Volterra [Volt26] eingeführt. Es wird seitdem in der Biologie und den Wirtschaftswissenschaften eingesetzt.

Mit Hilfe des Lotka-Volterra Modells können sogenannte Räuber-Beute Beziehungen zwischen zwei oder mehr Spezies simuliert werden. Die Veränderung der Population über die Zeit wird in Form der folgenden Differentialgleichungen angegeben:

$$\frac{dB}{dt} = a \cdot B - b \cdot B \cdot R \tag{6.1}$$

$$\frac{dR}{dt} = d \cdot b \cdot B \cdot F - c \cdot F \tag{6.2}$$

Gleichung 6.1 beschreibt die Veränderung der Beute-Population B, 6.2 die der Räuber-Population. Die Konstanten haben die folgenden Bedeutungen:

- a ist die natürliche Wachstumsrate von B in Abwesenheit von Räubern.
- b definiert die Sterberate von B durch die Räuber.
- c ist die Sterberate der Räuberspezies B falls keine Beute vorhanden ist.
- d ist die Effizienz, mit der Beutetiere in Raubtiere "umgewandelt" werden.

Wie man leicht sieht, handelt es sich um ein sehr einfaches Modell. Seine Gültigkeit wird durch die folgenden vereinfachenden Annahmen beschränkt.

1. Populationen können sich beliebig stark vermehren. Begrenzter Lebensraum oder andere Randbedingungen, die die Anzahl der Lebewesen nach oben begrenzen, existieren nicht.
2. Als einzige Interaktion zwischen den Spezies wird die Räuber-Beute Beziehung dargestellt. Andere Zusammenhänge, wie beispielsweise Symbiose oder Parasitentum werden nicht berücksichtigt.
3. Es wird implizit angenommen, daß alle Tiere gleichmäßig über den Lebensraum verteilt sind. Dies ist in der Natur natürlich in den meisten Fällen nicht der Fall.

Trotz dieser Einschränkungen wird das Lotka-Volterra Modell seit seiner Einführung 1925 in naturwissenschaftlichen Simulationen verwendet. Im Rahmen dieser Arbeit wird es beispielhaft für quantitative Simulationen und deren Kombination mit visueller Simulation verwendet. Andere Modelle, die die oben dargestellen Einschränkungen nicht aufweisen, könnten es in Zukunft ersetzen.

6.2.13 Integration von visueller und quantitativer Simulation

Synchronisierung der Simulationszeiten

Grundvoraussetzung für die Integration unterschiedlicher Simulationsmodelle ist ein gemeinsames Zeitraster. Stellt die quantitative Simulation Vorgänge dar, die in Echtzeit[8] ablaufen, ist ein gemeinsames Zeitraster gegeben. In einer Verkehrssimulation etwa kann die quantitative Simulation die Veränderung des Vekehrsaufkommens auf den Straßen einer Stadt im Verlauf eines Tages darstellen. Die Simulation beginnt zu einer bestimmten Tageszeit und der Autofahrer erlebt beispielsweise die Zunahme des Verkehrs nach Geschäftsschluß.

Veränderungen in Ökosystemen, wie sie durch das Lotka-Volterra Modell simuliert werden, laufen in Zeiträumen von Monaten und Jahren ab. Die Dauer der visuellen Simulation liegt aber typischerweise im Bereich von Minuten. Sie kann also nur einen vergleichsweise kleinen Ausschnitt des Geschehens darstellen. Dennoch können beide Modelle sinnvoll miteinander kombiniert werden.

Meist werden Simulationsergebnisse in Form von Tabellen oder Diagrammen dargestellt. Während diese Darstellung für Fachleute eine hohe Aussagekraft besitzt, ist sie für Laien eher schwer verständlich. Durch die visuelle Simulation steht nun ein ganz neues Mittel zur Verfügung, abstrakte Daten, wie die Veränderung von Populationen in einem Ökosystem direkt erfahrbar und verständlich zu machen.

In Was-wäre-wenn? Szenarien könnte das interessierte Publikum zunächst in das Simulationszenario eingreifen. Etwa könnte die Population einer Spezies verändert oder bestimmte Umwelteinflüsse verstärkt oder reduziert werden.

Anschließend läuft die Simulation nun einige (simulierte) Jahre und das Ergebnis kann durch die visuelle Simulation dargestellt werden. Auch eine Simulation im Zeitraffer ist möglich, in der die Simulationszeit der quantitativen Simulation auf die Echtzeit abgebildet wird. So vergeht etwa ein Monat pro Minute. Veränderungen im Ökosystem wie das Anwachsen einer Population oder das Aussterben einer anderen können so "live" erfahren werden.

Integration der Simulationsräume

Um die visuelle und quantitative Simulation räumlich zu integrieren wird das Konzept der Aura eingeführt, welches Abbildung 6.16 eingeführt.

[8]Echtzeit bedeutet, daß die Simulationszeit gleich schnell wie die reale Zeit abläuft.

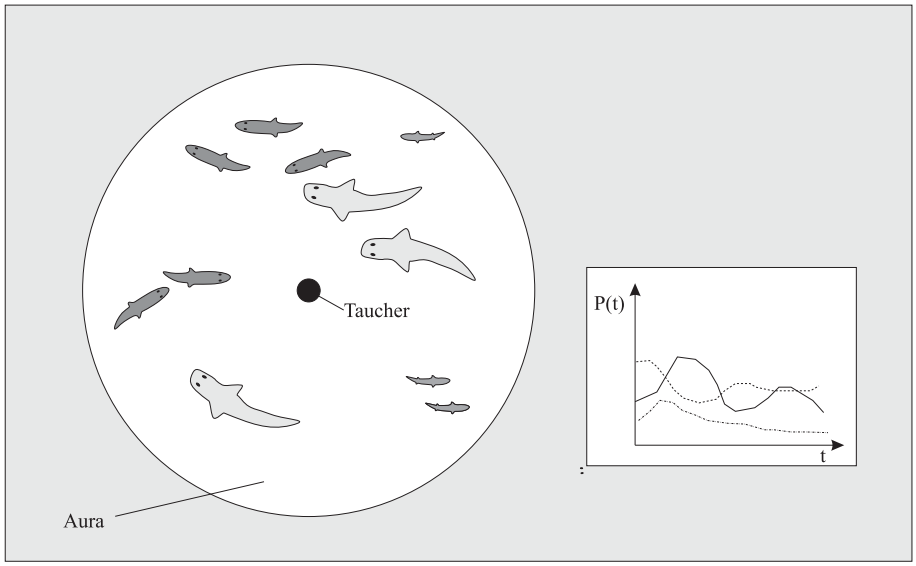

Abbildung 6.16: Aura Konzept

Bei der Aura handelt es sich um ein kugelförmiges Gebiet, in dessen Mitte sich der Beobachter (Taucher) befindet. Innerhalb der Aura findet die visuelle Simulation von einzelnen Kreaturen statt, außerhalb werden die Populationen quantitativ simuliert. Der Radius der Aura entspricht der Sichtweite des Betrachters. Günstig wirkt sich bei einem Szenario unter Wasser die Trübheit des Wassers aus, die die Sichtweite begrenzt.

Die quantitative Simulation gibt für jeden Zeitpunkt die Populationen der einzelnen Spezies vor. Diese ist in die visuelle Simulation abzubilden. Dazu muß zunächst berechnet werden, wieviele Tiere jeder Spezies ist der Aura vorhanden sein müssen, um die Simulationsergebnisse widerzuspiegeln.

Sei $P(t)$ die Population einer Spezies P zum Zeitpunkt t. Sei weiterhin V das Gesamtvolumen des Biotops und A das Volumen der Aura. Dann berechnet sich zu die erwartende Anzahl der Tiere der Spezies innerhalb der Aura p(t) wie folgt[9]:

$$p(t) = P(t) \cdot \frac{A}{V} \tag{6.3}$$

Für die Simulation unter Wasser muß berücksichtigt werden, daß es sich bei den Volumina A und V um das Volumen von Wasser handelt. Meeresboden, Steine, und die Teile des Modells, die außerhalb der Aquarien liegen, müssen abgezogen werden. Hierzu kann kann das bereits eingeführte Voxel-

[9]Hier wird implizit angenommen, daß die Tiere gleichmäßig im Biotop verteilt sind. Dies ist wird aber auch im Lotka-Volterra Modell angenommen.

gitter (siehe Abschnitt 3.13) herangezogen werden. V ergibt sich aus dem Produkt der Gesamtzahl freier Zellen, für P müssen die freien Zellen innerhalb des Aura-Volumens gezählt werden.

Sind nun innerhalb der Aura weniger als p(t) Fische der Spezies P vorhanden, müssen neue Fische "erzeugt" werden, sind es mehr, müssen die überzähligen Exemplare "gelöscht" werden. Wie kann dieses geschehen, ohne daß Fische plötzlich vor dem Betrachter auftauchen oder verschwinden? Dies muß natürlich auf jeden Fall verhindert werden.

Der einfachere Fall ist das Löschen überzähliger Fische. Da sich diese im Wasser bewegen, passiert es irgendwann, daß sie die Aura verlassen und gelöscht werden können, ohne daß der Beobachter dies wahrnimmt. Der Radius der Aura entspricht ja der Sichtweite unter Wasser. Bewegt sich der Taucher nun durch das Wasser, wird auch die Aura mitbewegt. Wie Abbildung 6.17 illustriert, entsteht dabei eine Zone hinter dem Taucher, in der Fische sowieso gelöscht werden müssen da sie durch die Bewegung der Aura, plötzlich außerhalb liegen.

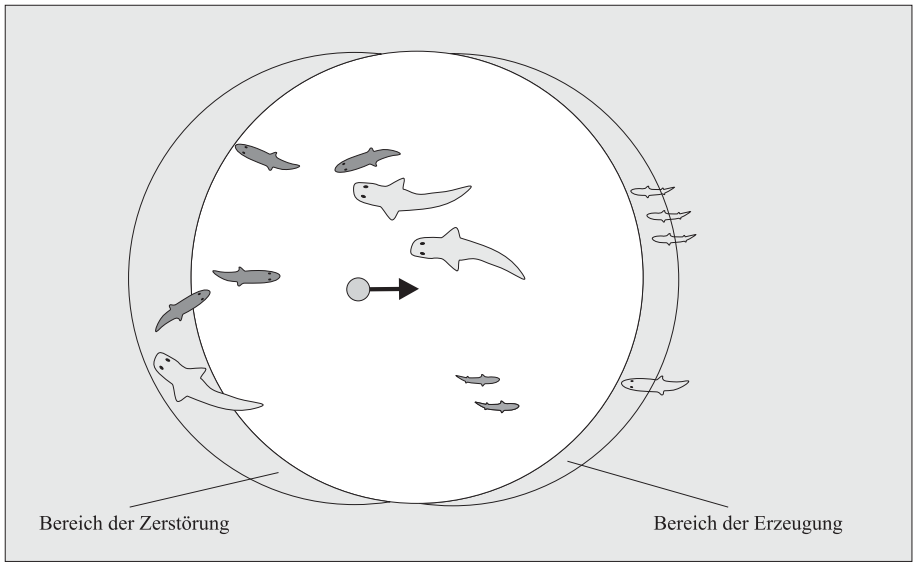

Abbildung 6.17: Erzeugung und Zerstörung von Fischen

Das Erzeugen neuer Fische ist etwas aufwendiger. Betrachten wir zunächst den statischen Fall, in dem sich die Aura nicht bewegt. Klar ist, daß neue Fische am Rand der Aura positioniert werden müssen. Die Richtung sollte dabei ungefähr auf das Zentrum der Aura weisen, damit sie nicht sofort wieder herausschwimmen. Fische die zufällig genau über dem Taucher plaziert wurden, würden nun aber genau senkrecht stehen, was in der Natur eher selten vorkommt. Um das zu verhindern, wird die senkrechte Koordinate des

Richtungsvektors auf einen Maximalwert beschnitten. Außerdem bevorzugt der Algorithmus, der die Positionen der neuen Fische berechnet, Positionen in der Höhe des Tauchers.

Für eine bewegte Aura wird eine weitere Heuristik herangezogen. Während im statischen Fall Fische überall um den Taucher herum erzeugt und wieder gelöscht werden, liegt der Fall hier anders. Fische werden hauptsächlich in Bewegungsrichtung des Tauchers generiert. Andernfalls würden sie bei fortgesetzter Bewegung gleich wieder verschwinden. Dieser Fall ist ebenfalls in Abbildung 6.17 dargestellt.

Zusätzlich muß beim Erzeugen berücksichtigt werden, daß Fische in der Realität eben nicht homogen im Wasser verteilt vorkommen. Einige Fische bilden Schwärme, viele bevorzugen bestimmte Lebensräume, etwa Korallenriffe oder den Meeresboden. Die Schwarmbildung kann beim Erzeugen der Fische berücksichtigt werden. Stehen mehrere Exemplare einer Spezies zur Erzeugung an, werden diese nahe beieinander positioniert. Der bevorzugte Lebensraum einzelner Gattungen kann im Voxelgitter vermerkt werden.

6.3 Der Dom von Siena

Eine weitere Anwendung des Simulationssystem für Autonome Objekte stellt die VR-Darstellung des Domes von Siena dar. Dieses Projekt wurde im Auftrag des Bundesministeriums für Bildung und Forschung (BMBF) durchgeführt und auf der EXPO 2000 präsentiert [LuFr00, RiSe99]. Ähnlich wie das Virtuelle Ozeanarium soll diese Installation einem breiten Publikum auf unterhaltsame Weise Wissen vermitteln.

6.3.1 Installation

Die Installation im Themenpark der Weltausstellung bestand aus einer Großbildprojektion mit zwei Metern Breite und vier Metern Höhe. Dieses Format wurde gewählt, um dem Modell des Domes gerecht zu werden, in dessen Darstellung Vertikale dominieren. Das Modell des Domes besteht aus dem Äußeren des Gebäudes und dem umgebenden Vorplatz sowie dessen Innerem inklusive einiger Kapellen und der Bibliothek. Wie beim Ozeanarium wurde auf die akkurate Wiedergabe des realen Vorbildes geachtet.

Einige Meter vor der Projektionsleinwand befindet sich ein Lesepult, in den ein berührungssensitiver Bildschirm eingelassen ist, das einen mittelalterlichen Folianten darstellt. Durch Berührung dieses Bildschirms können die Besucher darin blättern und mit dem Fremdenführer interagieren.

Das Publikum wird von einem virtuellen Fremdenführer mit namens Luigi, der in traditioneller Tracht bekleidet ist, durch den Dom geführt und mit den Sehenswürdigkeiten vertraut gemacht. Weitere Charaktere können die Szene beleben und in gespielten Episoden historisches Hintergrundwissen präsentieren. Der Platz vor dem Eingang des Domes wird von Tauben belebt.

Animierte Körperbewegungen und sprachsynchrone Mimik geben der Darstellung ein realistisches Aussehen.

Abbildung 6.18: Modell des Fremdenführers Luigi

Den Ablauf der Führung stellt Abbildung 6.19 grafisch dar. Insgesamt existieren 10 Stationen (in der Abbildung als numerierte rote Kreise dargestellt), an denen das System weitere Informationen bereithält. Zwischen diesen Stationen kann entlang der blauen Pfeile navigiert werden. Die Wahl der nächsten Station geschieht durch Berührung des Buches. Sobald eine Wahl getroffen wird, setzt sich Luigi in Bewegung und die Kamera folgt ihm automatisch. Diese einfache Form der Navigation wurde gewählt, um den Besuchern, die ständig wechseln ein schnelles Verstehen zu ermöglichen.

6.3.2 Autonome Objekte für virtuelle Charaktere

Der Einsatz von Autonomen Objekten für die Steuerung virtueller Charaktere soll an folgendem Szenario erläutert werden:

Akteure sind – neben dem Buch, den Besuchern und Luigi dem Fremdenführer – der Kirchenvorsteher der Kathedrale und Nicola Pisano, ein Bild-

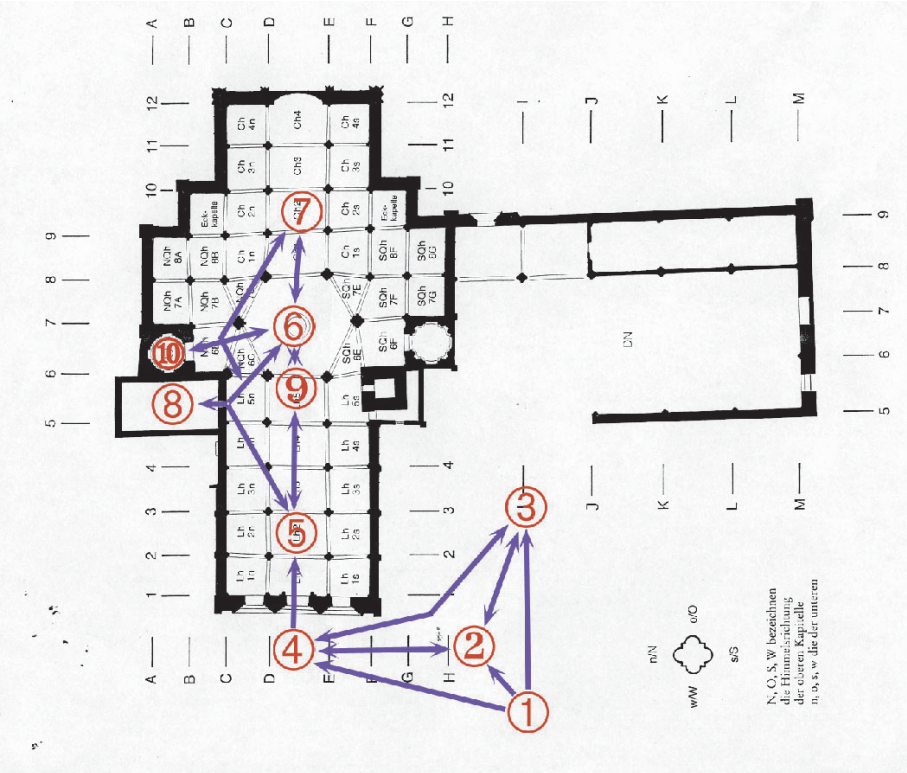

Abbildung 6.19: Grundriß des Modells und Ablaufplan

hauer aus der Zeit des Mittelalters, der einige der berühmtesten Kunstwerke im Dom geschaffen hat, sowie ein Ingenieur, der an der (gescheiterten) Erweiterung des Domes beteiligt war. Schließlich werden die Tauben auf dem Vorplatz ebenfalls durch eine eigene Klasse von Autonomen Objekten repräsentiert.

Für jeden Akteur existiert ein Autonomes Objekt, welches dessen Verhalten realisiert. Diese wird durch die folgenden Feature-Klassen realisiert:

1. Menschliche Figur
2. Taube
3. Buch
4. Kamera

Die Autonomen Objekte und deren Kommunikationsstruktur illustriert Abbildung 6.20.

Im Folgenden sollen die Rollen der einzelnen Autonomen Objekte und die Art der Nachrichten die sie untereinander austauschen skizziert werden.

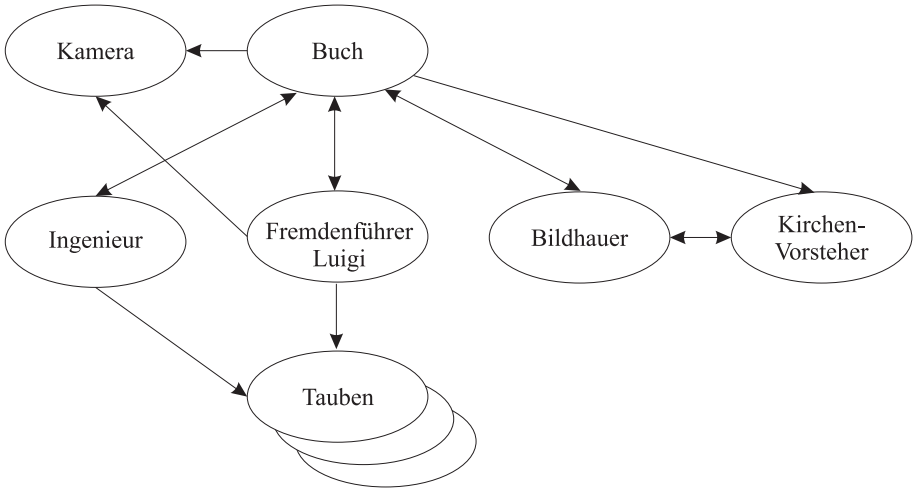

Abbildung 6.20: Kommunikation Autonomer Objekte

Das Buch

Wie bereits dargelegt, dient das Buch, realisiert durch einen berührungssensitiven Bildschirm, der Benutzerinteraktion. Es erlaubt die Navigation zwischen den verschiedenen Stationen der Führung, sowie an einigen Plätzen die Kontrolle über die Kamera um ein Umschauen zu ermöglichen. Daher sendet das Buch Nachrichten an Luigi und an die Kamera. Ist Luigi an Punkt angekommen, an dem neue Informationen bereitliegen, teilt er dies dem Buch durch eine entsprechende Nachricht mit. Daraufhin werden den Besuchern neue Wahlmöglichkeiten präsentiert.

Sollen den menschlichen Figuren Fragen gestellt werden, geschieht dies durch Berührung des Bildschirms. Dies löst eine entsprechende Nachricht an Luigi oder einen der anderen Akteure aus.

Die Kamera

Die Besucher werden in der virtuellen Umgebung ausschließlich durch die Kamera repräsentiert, welche dem Fremdenführer folgt oder von den Besuchern gesteuert werden kann. Sie empfängt also ausschließlich Nachrichten vom Buch und von Luigi.

Der Fremdenführer Luigi

Die Hauptaufgabe des AO, wie bei allen menschlichen Figuren, die grafische Darstellung der Person sowie die lippensynchrone Wiedergabe von gesprochenem Text. Über Nachrichten, die das Buch generiert, wird der Fremdenführer

gesteuert. Dieser teilt im Gegenzug mit, an welcher Stelle im Ablaufplan man sich gerade befindet und welche Informationen daher abrufbereit sind.

Kirchenvorsteher und Bildhauer

Diese beiden Figuren stehen im Dialog miteinander. Diese bestehen aus kurzen Sequenzen von gesprochenem Text und den passenden Animationen. Um den Dialog zu synchronisieren, werden Nachrichten ausgetauscht. Beide Figuren können auch auf Fragen der Besucher reagieren.

Ingenieur

Aufgabe dieser Figur ist es, das Bauwerk vom Vorplatz aus zu erklären. Auch ihm können Fragen gestellt werden. Da der Ingenieur die einzige Figur ist (neben dem Fremdenführer), die sich außerhalb des Domes aufhält, haben nur diese beiden die Möglichkeit Nachrichten an die Tauben zu senden. Um die Figuren menschlicher erscheinen zu lassen, können diese nämlich die Tauben füttern oder durch ärgerliche Gesten aufscheuchen und vertreiben.

Die Tauben

Von dieser Klasse autonomer Objekte existiert eine größere Anzahl von Instanzen. Tauben haben zwei Zustände, laufen und fliegen. Während sie auf dem Boden herumlaufen, vermeiden sie Kollisionen miteinander und suchen nach Futter (vergleiche Kapitel 6.2. Sobald sie aufgescheucht werden, fliegen sie in Schwärmen davon. Allerdings lassen sie sich nach einer Weile, oder wenn die gefüttert werden, wieder auf dem Domplatz nieder.

Kapitel 7

Zusammenfassung und Ausblick

Das Anliegen der vorliegenden Arbeit ist es, virtuelle Umgebungen mit mehr Leben zu füllen und deren Handhabbarkeit zu erleichtern. Sie konzentrierte sich daher auf zwei Schwerpunkte: Die Simulation von Verhalten in virtuellen Umgebungen und die komfortable und sichere Anbindung von multidimensionalen Interaktionsgeräten in VR-Systeme.

Zunächst wurde in Kapitel 2 der Stand von Wissenschaft und Technik evaluiert. Als Ergebnis konnte festgestellt werden, daß mit heutigen Systemen die Simulation von hoch dynamischen virtuellen Umgebungen nur begrenzt möglich ist. Als wichtigste Ursache wurde die Verwendung der hierarchischen Szenenbeschreibung als Datenbasis für die Simulation und die statische Verknüpfung zwischen Objekten mit Verhalten ausgemacht. Für die Modifikation von Verhalten müssen die VR-Systeme meist angehalten werden oder sind durch die Verwendung von Skriptsprachen nicht skalierbar auf komplexe Anwendungen. Diese Problem wird noch verstärkt durch die mangelnde Unterstützung von Multiprozessor Systemen.

In Bezug auf die Integration multidimensionaler Interaktionsgeräte wurde festgestellt, das die Integration und Handhabung dieser Geräte bei den meisten heutigen Systemen noch zu wünschen übrig lässt.

Aufgrund dieser Analyse wurde ein eigener, neuartiger Ansatz entwickelt, der die Simulation von hoch dynamischen virtuellen Umgebungen zuläßt und gleichzeitig Interaktionsgeräte flexibel und sicher handhabbar integriert.

Verhalten wird auf der Grundlage einer Welt-Datenbasis simuliert, die neben dem Szenengraphen, weitere Datenstrukturen enthält, die die effiziente Simulation von komplexen, dynamischen Szenen unterstützt.

Für die Beschreibung von Verhalten wurde ein innovatives Konzept aus Autonomen Objekten und sogenannten Features eingeführt. Während Autonome Objekte eigenständige Einheiten der Simulation darstellen und deren inneren Zustand kapseln, bestimmen Features die Eigenschaften dieser Ob-

jekte. Da Features zur Laufzeit zugeordnet werden können ist das System sehr flexibel einsetzbar und leicht zu erweitern.

Die Statische Verknüpfung zwischen Objekten wurde aufgelöst und durch ein durch dynamische Kommunikation über Nachrichten ersetzt. Für die Vermittlung der Kommunikation dient wiederum die Welt-Datenbasis.

Das System ist so gestaltet, daß es sich leicht parallelisieren lässt und so moderne Multiprozessor-Hardware ausnutzt. Damit sind die Vorbedingungen für die Entwicklung von reichhaltigen dynamischen Virtuellen Umgebungen gegeben.

Kapitel 5 wendete sich anschließend dem Problem der Integration multidimensionaler Interaktionsgeräte zu. Mit IDEAL wurde ein System entwickelt, welches die Anforderungen an ein modernes System, was die leichte und sichere Handhabbarkeit von Interaktionsgeräten betrifft, erfüllen kann.

Dieses System befindet sich seit 1998 im operativen Einsatz und ist Teil des kommerziellen VR-Systems Virtual Design II.

Die Leistungsfähigkeit des Systemes konnte anhand dreier unterschiedlicher Anwendungen demonstriert werden.

Autonome Objekte bildeten die Grundlage für den Astronauten-Trainingssimulator, eine verteilte Virtuelle Umgebung zwischen dem Johnson Space Center der NASA und dem Fraunhofer-IGD .

Die Einsatzmöglichkeiten von Autonomen Objekten für die Simulation von virtuellen Charakteren wurde anhand des Domes von Siena demonstriert. Ein Konzept für die Steuerung eines virtuellen Fremdenführers der durch eine gotische Kathedrale führt sowie weiterer Nebenfiguren wurde präsentiert. Ein Verfahren für die Pfadsuche von virtuellen Charakteren wurde ebenfalls eingeführt.

Am Beispiel des Virtuellen Ozeanariums, einer visuellen Simulation von Europas größtem Aquarium konnte die Leistungsfähigkeit besonders deutlich aufgezeigt werden. Über 1000 Fische konnten gleichzeitig mit ihrem Verhalten, ihrer Wahrnehmung und ihren typischen Schwimmbewegungen simuliert werden.

Durch eine Ergänzung der individuellen Simulation durch ein quantitatives Modell kann die Anzahl simulierter Fische nahezu unbegrenzt gesteigert werden.

Einige wichtige Aspekte konnten in der vorliegenden Arbeit nicht berücksichtigt werden und könnten Gegenstand weiterer Forschung und Entwicklung sein.

Der Import von VRML-Szenen und die Abbildung des Routing Konzepts auf Netze von Autonomen Objekten wäre ein wichtiger Entwicklungsschritt. In Form von VRML-Szenen existiert bereits eine Unzahl von teilweise animierten Modellen. Ziel einer solchen Entwicklung müsste es sein, VRML-

Szenen für die Simulation von Autonomem Verhalten verfügbar zu machen.

Um das Authoring von lebendigen Virtuellen Welten zu vereinfachen, wird eine komfortable grafische Benutzungsschnittstelle benötigt, die die Möglichkeit, dynamisches Verhalten zur Laufzeit des Systems zu konfigurieren ausschöpft. Eine solche Oberfläche sollte die vorhandenen Eigenschaften verwalten und Möglichkeiten bieten, diese Autonomen Objekten zuzuordnen. Protokolle zur Kommunikation durch Nachrichten sollten mit einem grafischen Editor visualisiert und manipuliert werden können. Schließlich sollte die Herstellung von Verbindungen zwischen grafischen und Autonomen Objekten erleichtert werden, indem grafische Objekte im Szenengraphen und in der dreidimensionalen Darstellung gepickt und Autonomen Objekten zugeordnet werden können.

Schließlich bietet es sich an, das System für verteilte Multi-User Anwendungen auszubauen. Die Kommunikation über Nachrichten bietet bereits eine Grundlage für den Transport von Informationen über das Netzwerk. Ansätze dafür wurden bereits anhand des Astronauten Trainingssimulators aufgezeigt.

Zukünftige Virtuelle Welten werden wesentlich lebendiger sein, als die, die wir heute kennen. Objekte, die tatsächlich simuliertes Verhalten aufweisen, werden die heute üblichen vorgerechneten Animationen ablösen. Nur so wird echte Interaktion möglich. Interaktionsgeräte werden leichter handhabbar seien, so wie heute Maus oder Tastatur. Zu beidem wollte die vorliegende Arbeit einen Beitrag leisten.

Literaturverzeichnis

[AiFr98] Ai, Z.; Fröhlich T: "Molecular Dynamics Simulation in Virtual Environments", Computer Graphics Forum (Eurographics'98 Proc.), 15(3): 267-273, 1998

[AWHW96] Allision, Don; Wills, Brian; Hodges, Larry, Wineman, Jean: "Gorillas in the Bits", Georgia Institute of Technology, Internal Report GIT-GVU-96-16, 1996

[AFFG96] Dai, F.; Felger, W.; Fruehauf, Th.; Goebel M.; Reiners, D.; Zachmann G.; "Virtual Prototyping Examples For Automotive Industries", Virtual Reality World 1996, Stuttgart, February 13-15, 1996

[BaTa97] Bandi, S.; Thalmann, D.:"A Configuration Space Approach for Efficient Animation of Human Firures", IEEE Non-Rigid and Articulated Motion Workshop, Puerto Rico, 1997

[BaTa98] Bandi, S.; Thalmann, D.:"Space Disretization for Efficient Human Navigation", Computer Graphics Forum, 17(3), 1998

[BBHL90] Blanchard, C.; Burgess, S.; Harvill, Y.; Lanier, J.; Lasko, A.; Obermann, M.; Teitel, M. : "Reality Built for two: A virtual reality tool", ACM SIGGRAPH Special Issue on the 1990 Symposium on Interactive 3D Graphcis, S. 35-36, 1990

[BEFK02] Behr, J.; Eschler, P.; Fröhlich, T.; Knöpfle, C.; Lutz, B.; Müller, S.; Roth, M.: "Cybernarium Days 2002 – Proving Virtual and Augmented Reality In A Public Exhibition", erscheint in Proceedings of Cyberworlds 2002, Tokyo, Japan, 2002

[BFKK01] Behr, J.; Fröhlich, T.; Knöpfle, C.; Kresse, W.; Lutz B.; Reiners, D.; Schöffel F.: "The Digital Cathedral of Siena – Innovative Concepts for Interactive and Immersive Presentation of Cultural He-

ritage Sites", Proceedings of ICHIM 2001 Conference, Mailand, Italien, 2001

[BoMa85] Q.Bone und N.B. Marshall: "Biologie der Fische", Gustav Fischer Verlag, 1985

[Bowe96] Bowen-Loftin: "A Shared Virtual Environment for Astronaut Training", Realtime Graphics, Vol 4, 1996

[Brys94] Bryson, S.: "Virtual Environments in Scientific Visualization", Virtual Reality Software and Technology, Proceedings of the VRST '94 Conference, 23. - 26. August, 1994

[CaHa93] Carlsson, C; Hagsand, O.:"DIVE – A platform for multi-user virtual environments", Computers & Graphics 17(6), S. 663-669, 1993

[Conw00] Conway, M. et al: "Alice: Lessons Learned from Building a 3D System for Novices", Proceedings of CHI 2000 Conference on Human Factors in Computing Systems, Den Haag, Niederlande, 2000

[DiIn99] Microsoft Corporation "DirectX Online Library", msdn.microsoft.com/library/sdkdoc/directx, 25.9.1999

[FDFH97] Foley, D.; van Dam, A.; Feiner, S.; Hughes, J.: "Computer Graphics Principles And Practice", 2nd Edition, Addision-Wesley Publishing, 1997

[Felg95] Felger, Wolfgang: "Innovative Interaktionstechniken in der Visualisierung", Dissertationsschrift, Technische Universität Darmstadt, Fachgebiet Grafisch-Interaktive Systeme, Springer Verlag Berlin Heidelberg 1995

[FHS98] "Syntax Description of the Fraunhofer VR-Data Exchange Format (FHS)", Fraunhofer-IGD , Interner Bericht, 1998

[FSZ96] Felger, Wolgang; Schäfer Reiner; Zachmann, Gabriel: "Interaktions-Toolkit INTO", Fraunhofer-IGD , Interner Bericht FIGD94i002, Darmstadt, 1996

[FDFM96] Fröhlich, T.; Damian, K.; Felger, W; Merbold, U: "Vom globalen Dorf zur virtuellen Raumstation", Spektrum der Wissenschaft, 10/1996, 1996.

[Froh97] Fröhlich, Torsten: "Das Virtuelle Ozeanarium", Thema Forschung 2/97, Technische Universität Darmstadt, 1997

[Froh00] Fröhlich, Torsten: "The Virtual Oceanarium - A Visual Computer Simulation of Europe's Largest Aquarium", Communications Of ACM, July 2000

[FrRo00] Fröhlich, T.; Roth, M.: "Integration of Multidimensional Interaction Devices in Real-Time Computer Graphics Applications", Computer Graphics Forum (Eurographics'00 Proc.), 15(3): 267-273, 2000

[FrKu02] Fröhlich, T.; Kullmann, D.: "Autonomous and Robust Navigation for Simulated Humanoid Characters in Virtual Envoronments", erscheint in Proceedings of Cyberworlds 2002, Tokyo, Japan, 2002

[Ghee95] Ghee, S: "dVS: A distributed VR systems infrastructure", ACM SIGGRAPH95 Course Notes, 1995

[GHJV94] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John: "Design Patterns - Elements of Reusable Object-Oriented Software", Addison Wesley, ISBN 0-201-63361-2, 1994

[Gree01] Green, M.: "MR Objects Programming Tutorial", University of Alberta, Dept. of Computer Science, Edmonton, Alberta, Canada, 2001

[GrWh95] Green, M.; White, L.:"Minimal Reality Toolkit Version 1.4 Programmer's Manual", University of Alberta, Dept. of Computer Science, Edmonton, Alberta, Canada, 1995

[Hags96] Hagsand, O.:"Interactive multiuser VEs in the DIVE system", IEEE Multimedia, S. 30-39, 1996

[HoUl88] Hopcroft, John E., Ullmann, Jeffrey D.: "Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie", ISBN 3-295118-82-9 Addison-Wesley Verlag, 1988

[Iso86] International Organization for Standardization: "Information Processing Systems - Computer Graphics - Graphics Kernel System for Three Dimensions (GKS-3D) - Functional Description",ISO DIS 8805, 1986

[Iso89] International Organization for Standardization: "Information Processing Systems - Computer Graphics - Programmer's Hierarchy Interactive Graphics System (PHIGS)", ISO DIS 9592, 1989

[Iso97] International Organization for Standardization: "Information technology - Computer graphics and image processing - The Virtual Reality Modeling Language (VRML)", 1997

[KeBH00] Kessler, G. Drew; Bowman, Doug A.; Hodges, Larry F.:"The Simple Virtual Environment Library: An Extensible Framework for Building VR Applications", Presence, Vol. 9, No. 2, April 2000, S. 187-208, 2000

[Klam97] Klambauer, Roger: "Verhalten künstlicher Lebewesen im Virtuellen Ozeanarium", Fachhochschule Darmstadt, Fachbereich Informatik, 1997

[KTMS98] Kalra, P.; Thalmann, N.; Moccozet, Sannier, L.: "Real-Time Animation of Realistic Virtual Humans", IEEE Computer Graphics and Applications Sept./Oct. 98, 1998

[Kuff98] Kuffner, J:"Goal-Directed Navigation for Animated Characters Using Real-Time Path Planning and Control", Proceedings of CAPTECH 98, 1998

[KuLV00] Kuffner, J; LaValle, S.:"RRT-Connect: An Efficient Approach to Single-Query Path Planning", Proceedings of IEEE Conference on Robotics and Automation, 2000

[Lotk25] Lotka, A. J.: "Elements of physical biology", Baltimore, Williams and Wilkins Co., 1925

[LuFr00] Lutz, B.; Fröhlich, T.:"Virtuelles Geschichtenerzählen - Der Dom von Siena", in U. Spierling (Hrsg.) Tagungsband zum 1. Workshop Digital Storytelling, Fraunhofer IRB Verlag, Darmstadt, 2000

[LYDS01] Lin, Fuhua; Ye, Lan; Duffy, Vincent G.; Su, Chuan-Jun: "Developing virtual environments for industrial training", Information Sciences 140 (2002), S. 153-170, Elsevier Science Inc, 2002

[Muel02] Müller, Stefan; persönliche Kommunikation, Darmstadt, 2002

[LuWe99] Lutz, B.; Weintke M.: "Die Dunhuang Höhlen"; Computer Graphics Forum (Eurographics'99 Proc.), 1999

[MaSa94] Massie, Thomas H.; Sailsbury, J. Kenneth: "The PHANToM Haptic Interface: A Device For Probing Virtual Objects", Symposium on Haptic Interfaces for Virtual Environments For Virtual Environments And Teleoperator Systems (ASME Winter Annual Meeting, Chicago, November 1994), 1994 Zitiert in [Zieg98])

[Merk97] Merkel, Michael: "Level Of Detail in der Simulation", Diplomarbeit, Fachhochschule Mannheim, 1997

[Muel96] Müller, Stefan: "Simulation von Kunst- und Tageslicht mit dem Radiocity Verfahren", Dissertationsschrift, Technische Universität Darmstadt, Fachbereich Informatik, Fachgebiet Grafisch-Interaktive Systeme, 1996

[PaBu95] Pausch, R. et al:"A Brief Architectural Overview of Alice, a Rapid Prototyping System for Virtual Reality", IEEE Computer Graphics and Applications, May 95, 1995

[Part82] Partgridge, Brian L.: "Wie Fische zusammenhalten", Spektrum der Wissenschaft, Spektrum (Akademischer Verlag), 1982

[Prop96] "Hardwaretreiber System für Superscape 4.00 (DIS) / Bedienermanual", VR Technologies GmbH, Juli 1996

[Pint01] Pinter, M.:"Toward More Realistic Pathfinding", http://www.gamasutra.com/features/20010314/pinter_01.htm, 2001

[Beaz01] Beazley, David:"Python Essential Reference", New Riders, ISBN 0735710910, Juni 2001

[RiSe99] Riedel, Peter A.; Seidel, Max (Hrsg.): "Die Kirchen von Siena", Band 3.1.3, "Der Dom von S. Maria Assunta", München, 1999

[Roth97] Roth, Marcus: "Entwicklung eines Client-Server Systems zur Handhabung multidimensionaler Interaktionsgeräte in 3D Echtzeit-Grafiksystemen", Diplomarbeit an der Fachhochschule Mannheim, Fachgebiet Informatik, 1997

[Rein94] Reiners, Dirk: "High Quality Real-Time Rendering for Virtual Environments", Diplomarbeit, Technische Universität Darmstadt, 1994

[Reyn87] Reynolds, Craig W.: "Flocks, Herds and Schools: A Distributed Behavior Model", Proceedings of ACM SIGGRAPH '87, S. 25-43, 1987

[RuNo95] Russel, S.; Norvig, P.:"Artificial Intelligence: A Modern Approach." Prentice Hall, Upper Saddle River, New Jersey, 1995

[SFGW00] Sharlin, E.; Figueroa, P; Green, M; Watson, B.: "A Wireless, Inexpensive Optical Tracker for the CAVE", In IEEE Virtual Reality 2000 Conference (IEEE-VR), March 22, 2000

[SGLS93] Shaw, C.; Green M.; Liang J; Sun, Y.:"Decoupled Simulation in Virtual Reality with The MR Toolkit", ACM Transactions on Information Systems, Volume 11 Number 3, pages 287-317, July 1993

[SLGS92] Shaw, C.; Liang, J., Green M., Sun, Y.:"The Decoupled Simulation Model for Virtual Reality Systems", in Proceedings of ACM CHI'92, S. 321-328, 1992

[Snow94] Snowdon, D.N.; West A. J.:"AVIARY: Design Issues for future large-scale virtual environments", Presence: Teleoperators and Virtual Environments, 3(4), S. 288-308

[Star96] Starck, Volker: "Entwicklung eines Prototypen für multimediale Virtual-Reality-Applikationen", Fachhochschule Furtwangen, 1996

[SFSE00] Stricker, D.; Fröhlich, T.; Söller-Eckert C.: "The Augmented Man", Proceedings of the International Symposium On Augmented Reality (ISAR00), München, 2000

[Stou99] Stout, B.:"Smart Moves: Intelligent Path-Finding", http://www.gamasutra.com/features/19990212/sm01.htm, 1999

[SVE02] "The Simple Virtual Environment (SVE) Toolkit", Quick start guide, File format definition, API reference manual" http://www.cse.lehigh.edu/GIVE/Software/softwover.html, Juni 2002

[Tamm84] Tamminen, Markku: "Efficient Octree Conversion by Connectivity Labeling", Computer Graphics, Vol. 18, No. 3, July 1984, 1984

[TeTG94] Terzopoulos, D; Tu, X.; Grzeszczuk, R.: "Artificial Fishes: Autonomous Locomotion, Perception, Behavior and Learning in a Simulated Physical World", Artificial Life 1, 1994

[ThSC96] Thalmann, D.; Shen j.; Chauvineau E.: "Fast Realistic Human Body Deformations for Animations and VR Applications", IEEE Computer Graphics, June 96, 1996

[ThSh95] Thalmann, D.; Shen, J.: "Interactive Shape Design Using Metaballs And Splines", Eurographics Workshop on Implicit Surfaces, 1995

[Tram99] Tramberend, H.: "Avocado: A distributed virtual reality framework", Proceedings of IEEE VR'99, S. 14-21, 1999

[TuTe94] Tu, Xiaoyuan; Demetri Terzopoulos: "Artificial Fishes: Physics, Locomotion, Perception, Behavior", Proceedings of ACM SIGGRAPH '94, S. 44-50, 1994

[Tu96] Tu, Xiaoyuan; "Biomechanics, Locomotion, Perception and Behavior", PhD Thesis, Department of Computer Science, University of Toronto, 1996

[Volt26] Volterra, V.: "Variazioni e fluttuazioni del numero d'individui in specie animali conviventi", Mem. R. Accad. Naz; dei Lincei. Ser. VI; Vol. 2, 1926

[VRT96] "VRT for Windows", Superscape Inc., 1997

[WTK94] "World Tool Kit / Technical overview" Sense8, 1994

[WTK99] "World Toolkit Reference Manual Release 9", Sense 8, April, 1999

[WUP00] "World Up User's Guide Release 5", Sense 8, 2000

[YaKF84] Yamaguchi, T. L.; Kunii, L; Fujimura, K.: "Octree Related Data Structures and Algorithms", Computer Graphics & Applications, January 1984, S. 53–59, 1984

[Zach96] Zachmann, Gabriel: "A Language for Describing Behavior of and Interaction With Virtual Worlds", Proceedings of ACM Conf. VRTS '96, Hongkong, 1996

[Zach97] Zachmann, Gabriel: "Distortion Correction of Magnetic Fields for Position Tracking", Proc. Computer Graphics International (CGI'97), June 23-27 1997, Hasselt and Diepenbeek, Belgien, 1997

[Zieg98] Ziegler, Rolf: "System zum integrierten Einsatz von haptischen Displays in virtuellen Umgebungen",Dissertationsschrift, Technische Universität Darmstadt, Fachgebiet Grafisch-Interaktive Systeme, 1998

Abbildungsverzeichnis

Curiculum Vitae

Name:	Torsten Fröhlich
Geburtsdatum:	31.1.1964
Geburtsort:	Bonn

1970 - 1974	Grundschule “Stiftsschule”
1974 - 1979	altsprachl. Beethovengymnasium
1979 - 1984	naturwissensch. Helmholtzgymnasium
1984 - 1986	Zivildienst
1986 - 1995	Studium der Informatik an der Technischen Universität Darmstadt
1993	Auszeichnung mit dem 2. internationalen Preis für graphische Datenverarbeitung
1995	Studienaufenthalt am Johnson Space Center der NASA in Houston, USA
seit 1996	Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für graphische Datenverarbeitung