

Johann Wolfgang Goethe-Universität
Frankfurt am Main



Diplomarbeit

**Konzept und Implementierung
eines Systems zur
Visualisierung von Zelldifferenzierungssimulationen**

von Paul Izquierdo Rojas

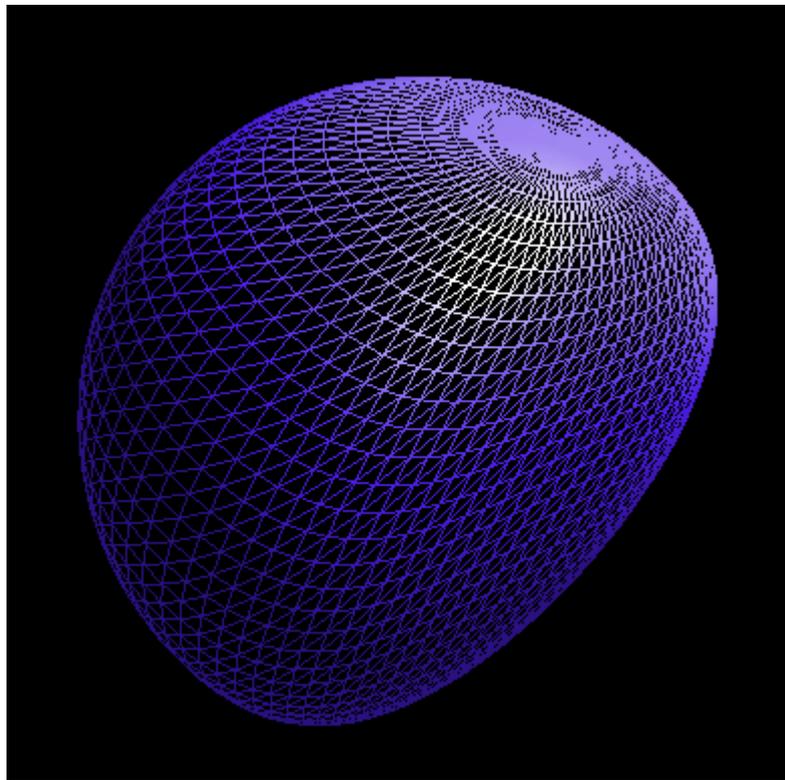
Fachbereich Biologie und Informatik

Prüfer: Prof. Dr.-Ing. Detlef Krömker
Professur für Graphische Datenverarbeitung

Betreuer: Dipl.-Inform. Christian Seiler
Dipl.-Biol. Jens Barthelmes

vorgelegt am 13. April 2004

**Konzept und Implementierung
eines Systems
zur Visualisierung
von
Zelldifferenzierungssimulationen**



D-VISION

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass die vorliegende Diplomarbeit ohne unzulässige Hilfe und nur unter Verwendung der angegebenen Literatur angefertigt wurde.

Frankfurt am Main, den 13. April 2004

Paul Izquierdo Rojas

Aufgabenstellung

Die Tatsache, dass sich komplexe vielzellige Organismen aus einer einzigen Zelle mit beschränktem Informationsgehalt entwickeln können, stellt eines der faszinierendsten Phänomene in den Lebenswissenschaften dar. Obwohl jede Zelle des menschlichen Körpers die gleiche genetische Information beinhaltet, gibt es trotzdem sehr viele verschiedene, auf bestimmte Aufgaben spezialisierte Zelltypen (z.B. Nervenzellen, Leberzellen, Hauptzellen etc.). Diese Zelltypen entstehen aus gemeinsamen Vorläufern, den sog. Stammzellen, welche noch das Potential haben, sich in verschiedene spezialisierte Zellen zu differenzieren.

Einige der für die Differenzierung von Stammzellen verantwortlichen Mechanismen sind bereits aufgeklärt und entsprechende Modelle wurden aufgestellt. Die Grundlage hierfür liefert der enorme technologische Fortschritt auf dem Gebiet der Molekularbiologie und die immensen Datenmengen, die durch die modernen Techniken generiert werden. Trotz dieser riesigen Datenbasis zu Modellorganismen wie *C.elegans*, *Drosophila* und Maus konnte die Entwicklung von bioinformatischen Methoden zur Analyse und Integration dieser Daten nicht schritthalten. Dies ist nicht zuletzt in der Komplexität und Nonlinearität der betrachteten Systeme begründet. Zudem ist die Übertragung von Ergebnissen aus der Forschung an Modellorganismen auf den Menschen nur begrenzt möglich. Direkte Forschungen an menschlichen embryonalen Stammzellen sind wiederum aus ethischen Gründen umstritten – zum Zweck der Gewinnung dieser Stammzellen müssten entsprechende menschliche Embryonen gezüchtet und gezielt abgetötet werden.

Einen wichtigen Beitrag zum Verständnis von Differenzierungsvorgängen kann die Simulation der beschriebenen komplexen entwicklungsbiologischen Modelle liefern. Um die Ergebnisse der Simulation mit den experimentellen Beobachtungen vergleichen zu können, bedarf es einer dedizierten Visualisierung.

Aufgabe dieser Arbeit ist es zunächst, den Stand der Forschung im Bereich der Simulation und Visualisierung der Zelldifferenzierung zu analysieren. Aus dieser Analyse ist eine Anforderungsbeschreibung an ein Visualisierungssystem zu formulieren. Dabei sind auch die Anforderungen an eine Schnittstelle zu Simulationswerkzeugen zu erarbeiten. Es ist ein eigenes Konzept eines Systems zur Visualisierung von Zelldifferenzierungssimulationen zu erstellen. Hierbei soll die Schnittstelle zu existierenden Simulatoren beachtet werden. Es ist weiterhin zu prüfen, welche Voraussetzungen der Einsatz von Grid Computing an das System stellt. Das Konzept soll die dreidimensionale Visualisierung der Entwicklungsprozesse und der dynamischen Zellzustände beinhalten. Dabei soll die Visualisierung möglichst so konzipiert sein, dass eine Vergleichbarkeit mit realen Mikroskopieaufnahmen gegeben ist, ohne dabei auf die erweiterten Möglichkeiten der Computergraphischen Visualisierung zu verzichten.

Das Konzept soll in einer Prototypischen Implementierung validiert werden. Hierzu kann statt einer Anbindung an ein existierendes Simulationssystem eine eignen rudimentäre Simulation zugrunde gelegt werden. Als Programmiersystem soll Java/Java3D Verwendung finden, Zielsystem ist der PC

Inhalt

Eidesstattliche Erklärung	v
Aufgabenstellung	vii
Abbildungsverzeichnis	xi
1 Einleitung	1
1.1 Motivation	1
1.2 Ziele	3
1.3 Aufbau der Arbeit	4
2 Grundlagen	5
2.1 Biologische Grundlagen	5
2.1.1 Bausteine der Zellen	5
2.1.2 Zellarten: Prokaryonten und Eukaryonten	9
2.1.3 Vorgänge in der Zelle	10
2.1.4 Zellwachstum und Zellteilung	12
2.1.5 Gen-Kontrolle	14
2.1.6 Entwicklungsbiologie	15
2.2 Simulationen	17
2.3 Visualisierung	19
2.3.1 Klassifikation von Visualisierungen	20
2.3.2 Ziele und Güte einer Visualisierung	20
2.3.3 Visualisierungsprozess	21
2.3.4 Methoden der Visualisierung	22
2.3.5 Visualisierungstechniken	23
3 Stand der Zellsimulations-Forschung	27
3.1 Forschungsgrundlagen für Zellsimulatoren	27
3.2 Vorstellung existierender Zelldifferenzierungssimulatoren	29
3.3 Bewertung der vorgestellten Simulatoren	35
4 Konzept	39
4.1 Daten	39
4.1.1 Das Datenproblem	39
4.1.2 Von den biologischen Grundlagen zum Datenmodell	40
4.1.3 Datenmodell	44
4.1.4 Anforderungen an einen Simulator	45

4.2	Datenfluss	47
4.2.1	Von der Simulation zur Visualisierung	47
4.2.2	Aufbau	47
4.3	Datenvisualisierung	48
4.3.1	Darstellung der Daten	49
4.3.2	Exploration der Daten	58
4.3.3	Zeitkomponente	62
4.3.4	Definition von Zellklassen	64
4.4	nicht in das Konzept aufgenommene Ideen	66
5	Implementierung	71
5.1	Eingesetzte Technologien	71
5.2	Beschreibung der Datenstruktur	72
5.3	Beschreibung der Objektstruktur	76
5.4	Beschreibung ausgewählter Realisierungen von Konzepten	72
5.4.1	Deformation von Kugeln	77
5.4.2	Konzentration mit Tiny Cubes	84
5.4.3	2D-Darstellung	86
5.5	Beschreibung der GUI	88
6	Beurteilung und Ausblick	91
	Anhang A: XML Dokument Definition	95
	Anhang B: Volumenberechnung deformierter Kugeln	97
	Anhang C: Glossar	99
	Anhang D: Literaturverzeichnis	103

Abbildungsverzeichnis

Abb. 2.1: Aufbau der Zellmembran aus Lipiden [ALB95, S. 564, 565]	6
Abb. 2.2: Struktureller Aufbau der DNS [1][2]	7
Abb. 2.3: Proteinstruktur [ALB95, S. 136]	7
Abb. 2.4: Hexokinase [ALB95, S. 226]	8
Abb. 2.5: Transkription und Translation [ALB95 S. 121]	11
Abb. 2.6: Proteinbiosynthese am Ribosom [1]	12
Abb. 2.7: Zellzyklus [MAX02]	13
Abb. 2.8: Bindung eines Proteins an DNS [ALB95 S. 493, 495]	14
Abb. 2.9: Ausbildung der Blastula[3]	16
Abb. 2.10: menschliche Embryogenese [4]	16
Abb. 2.11: Informationskreislauf bei Simulationen	18
Abb. 2.12: Visualisierungspipeline [SCH00]	21
Abb. 2.13: Autor und Betrachter [SCH00]	22
Abb. 2.14: Diagramme	22
Abb. 2.15: Farbskalen [SCH00]	23
Abb. 2.16: Diagramme mit Farben bzw. Texturen	23
Abb. 2.17: Konnektivität	23
Abb. 2.18: Darstellung der Zeit [5]	25
Abb. 3.1: Pflanzenmorphologie durch L-Systeme [PRU97]	29
Abb. 3.2: Ausbildung von Zellanhäufungen [AGA93]	30
Abb. 3.3: Zellhaufenbildung mit zwei verschiedenen Zelltypen [FLE96] ...	31
Abb. 3.4: Zelldarstellung durch Rechtecke [DEL94]	31
Abb. 3.5: Konzentrationsdarstellungen an einem Chromosom [BAK02]	32
Abb. 3.6: Zellhaufen als Graph mit Knoten und Kanten [SHA01]	32
Abb. 3.7: embryonale Entwicklung von <i>C. elegans</i> [KIT98]	34
Abb. 3.8: Ausbildung von Extremitäten [IZA04]	34

Abb. 3.9: Beispiele verschieden dichter Zellhaufen [KUN03]	35
Abb. 4.1: konzeptionelles Datenmodell	44
Abb. 4.2: Systemaufbau	48
Abb. 4.3: zweizelliger Embryo [ALB95, S.1251]	50
Abb. 4.4: Zwillingsidotter	50
Abb. 4.5: Schnittbild	52
Abb. 4.6: Substanzverteilung in vier Zellen	54
Abb. 5.1: Datenstruktur	72
Abb. 5.2: sich überschneidende Kugeln	77
Abb. 5.3: verschiedene Lagen der Trennebene	79
Abb. 5.4: Deformation durch Verschiebung der Punkte	79
Abb. 5.5: lineare, quadratische und kubische Deformation	80
Abb. 5.6: Veranschaulichung der Kugeldeformation	81
Abb. 5.7: Tiny Cubes Verteilung	85

- [1] http://www.agen.ufl.edu/~chyn/age2062/lect/lect_07/lect_07.htm
[2] http://www.biology.arizona.edu/cell_bio/tutorials/cell_cycles/cells1.html
[3] <http://home.uni-one.nl/walembryo/dkwakbol.htm>
[4] http://www.mercksource.com/pp/us/cns/cns_hl_dorlands.js
pZQzpgzEzzSzppdocszSzuszSzcommonzSzdorlandzSzdorlandzSzdmd_e_06zPzhtm
[5] entnommen aus: R. Dörner: Vorlesungsskript Visualisierung, WS 2002/2003

1 Einleitung

Der Mensch besteht aus Zellen. Es sind es sehr viele Zellen, und es sind sehr viele verschiedene Zellen. Sie erfüllen ihre Aufgabe in den inneren Organen, in jedem Muskel, im Gehirn, etc. Und obwohl sie in Form, Größe und Funktion verschiedener nicht sein könnten, bringen sie alle dieselben Voraussetzungen mit. Jede Zelle eines Körpers enthält dieselben Gene. Alle Zellen entstehen aus einer einzigen, der befruchteten Eizelle. Wie ist es möglich, dass aus dieser einzelnen Zelle ein so komplexer Organismus wie der menschliche Körper entsteht?

“Das Erbgut von Mensch und Schimpanse ist zu 98,7 Prozent identisch.” [MAX02] Wie kann es dann sein, dass sie sich dennoch so sehr unterscheiden? Was bewirkt die Unterschiede? Und welchen Nutzen kann die Antwort auf diese Frage haben?

1.1 Motivation

Der Schlüssel zu den Fragen liegt in den Zellen selbst. Tatsächlich sind nicht allein die Gene dafür verantwortlich, was aus der Eizelle einmal wird. Eine entscheidende Rolle spielt, was die Zelle mit ihren Genen tut¹. Nur ein Bruchteil des gesamten Erbgutes wird von den Zellen unseres Körpers überhaupt benutzt. Und all die verschiedenen Zellen setzen verschiedene Teile des Erbgutes ein, um zu funktionieren.

Am Anfang, wenn ein Embryo zu wachsen beginnt, sind alle seine Zellen noch gleich. Sie sind noch nicht darin festgelegt, zu welchem Teil des Körpers sie einmal werden. Diese sog. Stammzellen beginnen sich beim Menschen innerhalb des ersten Tages nach der Befruchtung zu unterscheiden [MAX02]. Die Entwicklung einer Stammzelle zu einer spezialisierten Körperzelle nennt man Zelldifferenzierung.

Zellen, die sich differenziert haben, besitzen immer noch dieselbe genetische Information. Aber sie haben begonnen, unterschiedliche Teile davon zu nutzen. Sie stellen verschiedene Substanzen her. Sie beginnen sich in ihrer Stoffzusammensetzung zu unterscheiden. Sie verändern ihre Form und Größe. Sie bilden Gewebestrukturen aus, werden zu Muskeln, Organen und Knochen.

Das Wissen darüber, warum sich Zellen in eine bestimmte Richtung weiterentwickeln, ist bisher unvollständig. Zwar kennt man einige der

¹ ihre sog. Genexpression

Mechanismen der Zelldifferenzierung, aber dieses Wissen reicht nicht aus, um die Differenzierung einzelner Zellen aktiv zu lenken.

Der Nutzen einer gesteuerten Zelldifferenzierung liegt in ihrer medizinischen Anwendung. Könnte man einer Zelle "mitteilen", zu welcher Körperzelle sie werden soll, so könnte man im Labor Ersatzgewebe züchten und damit viele Krankheiten, die auf Gewebedefekten beruhen, behandeln. Aktuell wird in Deutschland an der Nachzüchtung von Nerven- und Herzmuskelzellen geforscht.²

Das Rohmaterial für diese Forschungen sind embryonale, menschliche Stammzellen. Deren Herstellung ist in Deutschland jedoch gesetzlich untersagt. Um Stammzellen zu gewinnen, müssten Embryonen gezielt abgetötet werden, was in unserer Gesellschaft als ethisch verwerflich gilt. Die Alternative ist der Import von Stammzellen aus dem Ausland, und der ist teuer. Welche Möglichkeiten bleiben den Wissenschaftlern noch? Eine Methode ist der Einsatz von Simulationen.

Um 1967 bezeichnete der Informatiker J.C.R. Licklider die Entstehung rechnergestützter Simulationen als das wichtigste Ereignis für Wissenschaft und Technologie seit der Erfindung des Schreibens. [LIC67] Computersimulationen finden als neue Formen der Wissensproduktion in unterschiedlichsten Einsatzgebieten z.B. in der akademischen Forschung wie auch in der Industrie Verwendung. Heutzutage werden Simulationen zunehmend durch Computer realisiert, weil diese ein flexibles Umfeld für diverse Arten der Simulation bieten.

"Mit der Etablierung der Computational Science und ihres Forschungsinstruments der Computersimulation entsteht nicht nur, neben Theorie und Experiment, ein dritter Weg der Wissenschaften [...], sondern mit der Visualisierung der enormen Datenmengen erhält eine Bildlichkeit Einzug in die Wissenschaften wie nie zuvor." [GRA01]

Simulationen werden benutzt, wenn die Vorgänge, die simuliert werden sollen, in der Realität zu teuer oder zu gefährlich sind. Eine Simulation kann nicht nur häufig mit geringerem Materialaufwand erstellt werden als ein realer Test, sie ist auch leicht zu variieren und kann solange optimiert werden, bis sie alle nötigen Anforderungen erfüllt.

Neben den wirtschaftlichen Gründen sprechen oft auch zeitliche oder ethische Probleme gegen die praktische Umsetzung in der Forschung. So beschäftigt sich der Earth Simulator in Yokohama [STI04] unter anderem mit der Simulation von

² Oliver Brüstle von der Bonner Universität bekam die Erlaubnis, Stammzellen zur Forschung an neuronalem Ersatzgewebe aus Israel zu importieren, ebenso Jürgen Hescheler von der Kölner Universität, der Herzmuskelzellen erforscht. [SPR95]

Klimamodellen, deren Beobachtung in der Natur schlicht einen zu großen Zeitrahmen in Anspruch nehmen würde. Experimente wie Tierversuche [GRO98] oder die bereits angesprochene Stammzellenforschung sind gesellschaftlich und wissenschaftlich umstritten. Der Einsatz von Simulationen könnte solche Spannungsfelder deutlich entschärfen.

Um eine Simulation zu betreiben, müssen Modelle entwickelt werden. In der Stammzellenforschung beschreiben die Modelle die Vorgänge in Zellen. Mithilfe anderer Spezies werden Erkenntnisse über die Entwicklung von Stammzellen gesammelt. Das gesammelte Wissen wird in Modelle verpackt, die als Grundlage für die Entwicklung von Zellsimulationen dienen. Da die Zellen speziell von Säugetieren besonders zu Beginn der Embryogenese in ihrer Entwicklung den menschlichen Zellen ähneln, können die Schlussfolgerungen zum Teil auch auf menschliche Stammzellen übertragen werden. Wenn sich also herausstellt, dass die Simulationsergebnisse mit der Natur übereinstimmen, so kann der nächste Schritt ein Modellentwurf für die menschliche Stammzelle sein.

Simulationen produzieren Daten. Je komplexer das Modellsystem ist, umso größer ist die Datenmenge und umso dringender wird eine gute Visualisierung der Daten benötigt. Die Daten, die eine Simulation für die Wettervorhersage liefert, sind äußerst umfangreich. Um die Wettervorhersage auch ungeschulten Zielpersonen verständlich zu machen, muss eine Visualisierung der Daten die Informationen auf ein geeignetes Maß reduzieren und dann ein anschauliches Bild erzeugen. Das Ergebnis kennen wir aus den täglichen Fernsehnachrichten.

Um eine gute Visualisierung zu erzeugen, müssen die zugrunde liegenden Daten analysiert werden. Die Visualisierung muss in der Lage sein, den springenden Punkt in das Auge des Betrachters zu werfen. Der Simulationshintergrund muss deshalb dahingehend untersucht werden, welche Rohdaten entscheidend sind, um den Zweck der Simulation zu erfüllen.

1.2 Ziele

Im Rahmen dieser Arbeit soll ein Konzept für ein Visualisierungssystem für Zelldifferenzierungssimulationen entworfen werden. Dazu soll eine Anforderungsbeschreibung durch die Analyse von existierenden Simulatoren an ein solches Visualisierungssystem formuliert werden. Das Konzept soll eine dreidimensionale Darstellung von Daten, die Voraussetzung für den Einsatz von Grid Computing und die Vergleichbarkeit zu realen Mikroskopieaufnahmen beinhalten.

Durch die Erstellung eines Prototypen soll das Konzept realisiert werden.

1.3 Aufbau der Arbeit

Im zweiten Kapitel dieser Arbeit werden die Grundlagen der Zelldifferenzierung, Simulation und Visualisierung vorgestellt.

Kapitel drei beschäftigt sich damit, den Stand der Forschung im Bereich der Simulation von Zelldifferenzierung und der Visualisierung, der aus den Simulationen gewonnenen Daten, wiederzugeben. Schließlich wird überprüft, ob die aktuell verfügbaren Simulatoren als Datenquelle für die geplante Visualisierung dienen können.

Im vierten Kapitel wird das Konzept, auf dem der entwickelte Prototyp basiert, vorgestellt. Es soll aufgezeigt werden, wie die Erkenntnisse der vorangegangenen Kapitel zur Lösung der gestellten Aufgabe geführt haben. Das beinhaltet die Konzeption des Datenmodells, den grundlegenden Aufbau des Prototypen sowie die Beschreibung der in das Konzept aufgenommenen Darstellungsmethoden.

Das fünfte Kapitel beschreibt schließlich die Implementierung des Prototypen. Es werden die Datenstruktur, die Objektstruktur sowie interessante Aspekte der Umsetzung des Konzeptes in die prototypische Implementierung beschrieben.

Den Abschluss bildet das sechste Kapitel mit einer Zusammenfassung und Bewertung der Arbeit.

2 Grundlagen

In diesem Kapitel wird der Begriff der Zelldifferenzierung und die zugrunde liegenden biologischen Vorgänge vorgestellt. Weiterhin wird der Begriff der Simulation in Abschnitt 2.2, der Begriff der Visualisierung in Abschnitt 2.3 sowie die Methoden, die sich dahinter verbergen, kurz erläutert.

2.1 Biologische Grundlagen

Alle Lebewesen auf unseren Planeten, von Bakterien und Einzellern über Pflanzen und Tiere bis zum Menschen bestehen aus Zellen. Das Leben, wie es in [NET98] definiert wird, findet in Zellen statt. Der prinzipielle Aufbau und die grundlegenden Funktionen von Zellen sind allen Lebewesen gleich.

Eine Zelle stellt ein in sich geschlossenes Kompartiment dar, in dem die Prozesse des Lebens stattfinden. Eine Membran, die aus einer Doppellipidschicht aufgebaut ist, trennt das Innere der Zelle von ihrer Umgebung ab.

Die für das Leben essentiellen Prozesse sind laut [NET98] Stoffwechsel sowie Wachstum und Reproduktion.

Im Folgenden wird der Aufbau und die Funktionsweise von Zellen näher vorgestellt. Es werden Zellarten unterschieden und die prinzipiellen Abläufe in den Zellen erklärt. Weiterhin wird der Ablauf des Zellwachstums und der Zellteilung beschrieben. Die Funktionsweise der Gen-Kontrolle, die die Grundlage für die Differenzierung von Zellen darstellt, wird im Unterabschnitt 2.1.5 näher beleuchtet. Als Letztes wird die Entwicklung eines mehrzelligen Organismus aus einer einzelnen Ursprungzelle beschrieben.

2.1.1 Bausteine der Zellen

Zellen sind aus einigen wenigen Arten von Bausteinen, die allerdings biochemisch sehr komplex sind, aufgebaut. Sie werden im Folgenden kurz vorgestellt.

Lipide

Lipide sind Moleküle, die einen hydrophilen und einen hydrophoben Teil besitzen. Sie bilden eine Doppelschicht, die die Zelle als Membran umschließt. Dadurch wird die wässrige Lösung im Inneren der Zelle, das sog. Zytoplasma, von der Umgebung getrennt. Wasser kann durch die Membran diffundieren, die enthaltenen Substanzen und Bestandteile der Zelle werden jedoch in der Zelle gehalten. Allerdings verfügt die Zelle in der Zellmembran über aktive

Mechanismen, mit deren Hilfe sie gezielt Substanzen mit ihrer Umgebung austauschen kann.

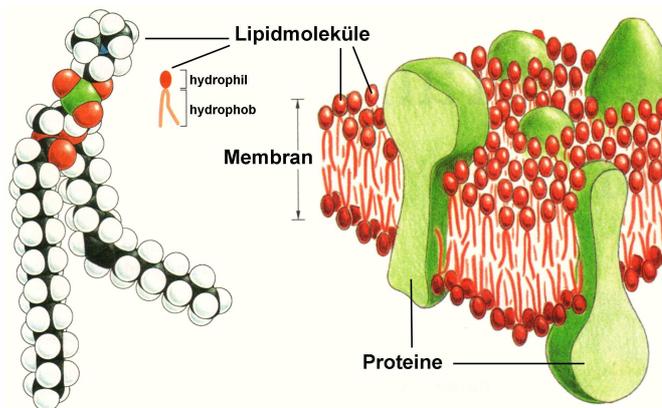


Abb. 2.1: Aufbau der Zellmembran aus Lipiden [ALB95, S. 564, 565]

DNS und RNS

Der Bauplan für alle Substanzen, die von der Zelle hergestellt werden können, ist im Erbmaterial codiert. Die Desoxyribonukleinsäure (DNS) bzw. die Ribonukleinsäure (RNS) ist der Träger dieser Information. Die DNS ist ein Polymer, das aus Nukleotiden aufgebaut ist. Ein Nukleotid besteht aus einem Zuckermolekül (Ribose), an das eine Base sowie ein Phosphat gebunden sind. Das Phosphatmolekül stellt die Brückenverbindung zwischen den Zuckermolekülen der langkettigen DNS-Stränge her. Es gibt vier Basen, die in der DNS enthalten sein können: Adenin (A), Guanin (G), Cytosin (C) und Thymin (T). Das genetische Alphabet enthält also nur vier Buchstaben. Jede Base bindet sich durch eine Wasserstoffbrücke an eine Base eines zweiten, komplementären DNS-Stranges. Dabei bindet sich Adenin an Thymin (A-T) und Guanin an Cytosin (G-C). Die DNS wird durch diesen zweiten Strang stabilisiert, und es kann sich eine α -Helix bilden. Die doppelsträngigen DNS-Ketten (die Chromosomen) winden sich um Histon-Proteine, die dann erneut eine Helix bilden. So ist eine kompakte Lagerung möglich, die die DNS reaktionsträge macht und dadurch die enthaltene genetische Information besser erhält.

Der Unterschied zwischen RNS und DNS besteht darin, dass die Ribose in der RNS eine freie OH-Gruppe besitzt, an deren Stelle bei der Ribose der DNS nur ein Wasserstoffatom sitzt. Ein weiterer Unterschied besteht darin, dass für den Aufbau der RNS anstelle der Base Thymin die Base Uracil (U) verwendet wird. Die RNS ist im Gegensatz zur DNS in den Zellen abbaubar.

DNS dient in den Zellen als Informationslager. Durch einen Kopiervorgang wird diese Information auf RNS übertragen, die dann der Zelle als Arbeitsmaterial dient. Unter anderem dient die RNS zur Synthese von Proteinen.

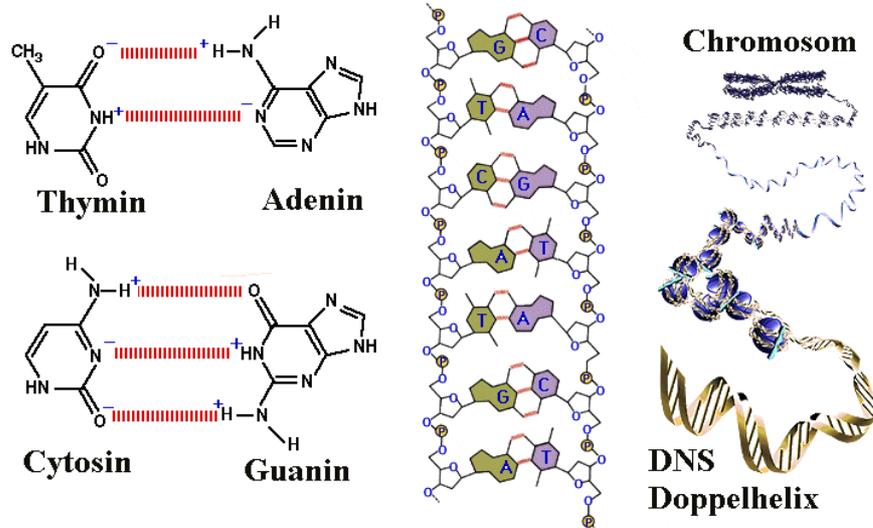


Abb. 2.2: Struktureller Aufbau der DNS [1][2]

Proteine

Proteine sind Polypeptide. Sie sind Polymere aus Aminosäuren. Zwei Aminosäuren sind durch eine Peptidbindung miteinander verbunden. Es gibt 20 verschiedene Aminosäuren, die zum Aufbau von Proteinen verwendet werden³. Eine Sequenz von Aminosäuren stellt die Primärstruktur der Proteine dar. Bedingt durch die verschiedenen Molekülreste, die an den Aminosäuren hängen und von der eigentlichen Polypeptidkette weg zeigen, bilden die Ketten eine übergeordnete Sekundärstruktur⁴ aus. Das äußere Erscheinungsbild einer Proteinkette, das durch

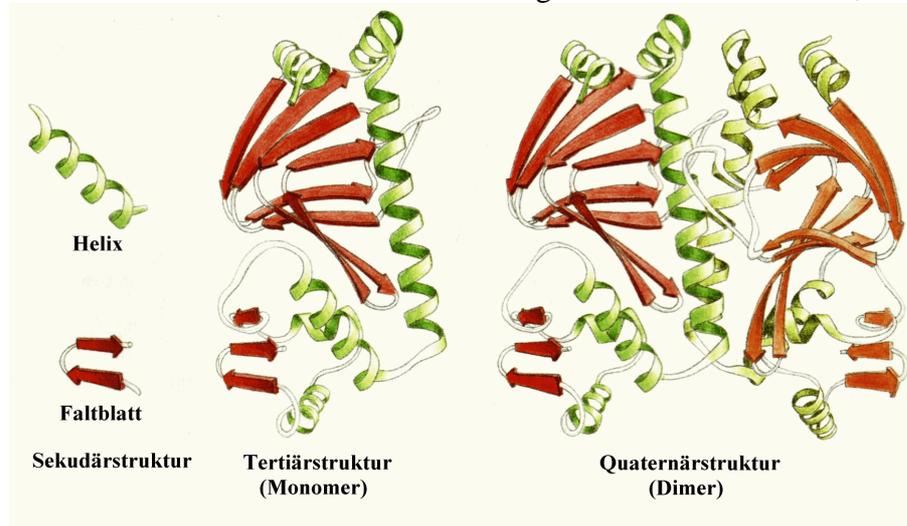


Abb. 2.3: Proteinstruktur [ALB95, S. 136]

³ Von diesen 20 Aminosäuren können 9 sog. essentielle Aminosäuren nicht vom Menschen in seinen Zellen hergestellt werden, und müssen mit der Nahrung aufgenommen werden.

⁴ Zur Sekundärstruktur zählen die α -Helix und die β -Faltblattstruktur.

die Sekundärstruktur von Proteinsequenzteilen sowie durch Verknüpfungen zwischen den Aminosäureresten entsteht, wird Tertiärstruktur genannt. Mehrere solche „zusammengeknüllte“ Proteine können sich zu einem Protein-Komplex zusammenfügen, der dann das eigentliche, wirksame Protein in seiner Quaternärstruktur darstellt.

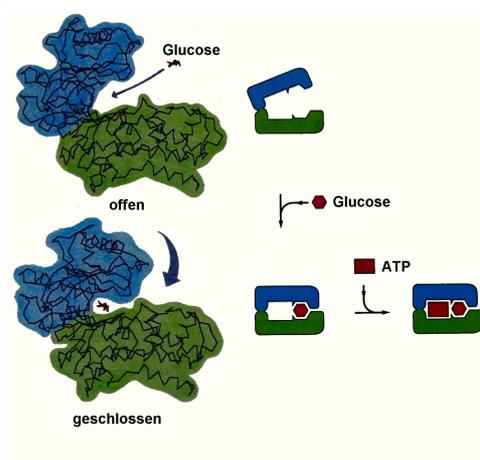


Abb. 2.4: Hexokinase [ALB95, S. 226]

An bestimmten Stellen der äußeren Proteinoberfläche sind polare, unpolare, saure oder basische Reste von Aminosäuren zu finden, die eine spezifische dreidimensionale Struktur von Bindungsstellen formen. Sie ermöglicht es dem Protein, sich an einer entsprechenden, komplementären Oberfläche anzulagern.

Dies ist die Grundlage für das „Schlüssel-Schloss-Prinzip“, mit dem die meisten biochemischen Vorgänge kontrolliert werden.

andere Substanzen

Weitere Substanzen, die in Zellen vorkommen und insbesondere für die Ernährung von Zellen eine Rolle spielen, sind Kohlenhydrate und Fette.

Kohlenhydrate sind Moleküle, deren chemische Summenformel dem Schema $C_n(H_2O)_m$ entspricht. Größere Kohlenhydrate sind Polymere kleiner Zuckermoleküle. Die Zellulose ist z.B. ein in Pflanzen vorkommendes langkettiges Kohlenhydrat. Die meisten höheren Lebewesen ernähren sich von Kohlenhydraten, indem diese vollständig abgebaut d.h. aerob zu Kohlendioxid (CO_2) und Wasser oxidiert werden.

Fette bestehen aus Fettsäuren, die wiederum lange Kohlenwasserstoff-Ketten mit einer Karbonsäure-Gruppe sind. Auch Fette können von Zellen abgebaut werden um Energie zu gewinnen.

Fettsäuren und Zucker sind aber auch Bestandteile von anderen Bausteinen der Zelle.

Abgesehen von den bereits erwähnten Substanzen, gibt es noch viele weitere chemische Substanzen in Zellen, deren Vorstellung den Rahmen dieser Arbeit jedoch sprengen würde.

2.1.2 Zellarten: Prokaryonten und Eukaryonten

Im Laufe der Evolution haben sich zwei grundsätzlich verschiedene Arten von Zellen gebildet: Prokaryonten und Eukaryonten.

Die Prokaryonten sind in ihrer Struktur einfach. Sie bestehen aus einer einzelnen Zelle und besitzen keinen Zellkern. Das Erbgut liegt frei im Zytoplasma. Bakterien sind typische Vertreter von Prokaryonten. Die Evolution hat eine Vielzahl verschiedenartiger Bakterien hervorgebracht, die sich vor allem in ihrem Stoffwechsel und damit in der Nutzbarmachung von chemischer Energie unterscheiden (z.B. Schwefelbakterien, Halophile Bakterien, Methanogene, Cyanobakterien, aerobe und anaerobe Bakterien).

Die zweite Art von Zellen sind die Eukaryonten, die im Gegensatz zu den Prokaryonten ein sehr differenziertes Zellinneres aufweisen: Entscheidendes Merkmal der Eukaryonten ist der Zellkern, in dem das Erbgut vom Rest des Zellinhaltes getrennt ist. Dabei ist das Genom, also sämtliche DNS-Stränge der Zelle, hoch strukturiert zusammengepackt.

Weiterhin enthält das Zytoplasma mehrere Arten von Zellorganellen, die durch eigene Lipidmembranen abgeschlossene Systeme bilden. In ihnen finden spezifische Stoffwechselreaktionen statt. Zu den Organellen gehören unter anderem die Mitochondrien, die Chloroplasten, das Endoplasmatische Retikulum und der Golgi-Apparat. Organellen wie die Mitochondrien und die Chloroplasten entsprechen in ihrem Aufbau Prokaryonten und enthalten eigene DNS. Es wird vermutet, dass sie im Verlauf der Evolution in die Eukaryonten eingewandert sind. Abgesehen von den Organellen enthalten vor allem tierische eukaryontische Zellen ein Cytoskelett, das aus hochstrukturierten Protein-Filamenten besteht und einerseits der mechanischen Stabilität dient, andererseits aber auch an einer Vielzahl intrazellulärer Prozesse beteiligt ist (z.B. Zellteilung, Zellbewegung, Vesikeltransport). Pflanzliche Zellen bilden zusätzlich eine Zellwand aus, die ihnen äußeren Halt bietet.

Im Folgenden werden kurz einige eukaryontische Organellen vorgestellt:

Mitochondrien

Die Mitochondrien sind die Kraftwerke der Zelle. In ihnen wird die Energie, die beim Abbau von Nährstoffen umgesetzt wird, in Adenosin-Tri-Phosphat (ATP) gespeichert.

Chloroplasten

Die Chloroplasten enthalten Chlorophyll, das für die Photosynthese notwendig ist. Chloroplasten kommen in allen höheren Pflanzen vor und ermöglichen es ihnen, mit Hilfe der Lichtenergie der Sonne aus CO₂ und Wasser Kohlenhydrate zu synthetisieren. Im Inneren der Chloroplasten befindet sich ein hoch entwickeltes inneres Membransystem, das den Photosyntheseapparat enthält.

Endoplasmatisches Retikulum

Das Endoplasmatische Retikulum (ER) besteht aus einem ausgedehnten Membransystem, das das Zytoplasma der Zelle durchzieht. Man kann das glatte und das raue ER unterscheiden. Das glatte ER ist eher schlauchförmig und ist für den Lipidstoffwechsel zuständig. Das raue ER besteht aus ausgedehnten Membrantaschen und ist mit Ribosomen bedeckt, die für die Proteinsynthese notwendig sind.

Golgi-Apparat

Der Golgi-Apparat, der aus gestapelten, flachen Membransäckchen besteht, ist für das Sammeln, Sortieren, Verpacken und Transportieren von Makromolekülen zuständig. Diese Substanzen können zu anderen Organellen oder zur Zellmembran, wo sie dann aus der Zelle ausgeschüttet werden, transportiert werden.

Pilze, Hefen, einzellige Urtierchen, alle Pflanzen und Tiere sind aus eukaryontischen Zellen aufgebaut. Dabei bestehen alle höheren Organismen wie Pflanzen und Tiere aus vielen Zellen.⁵ Der komplexe Aufbau dieser Organismen wird durch spezialisierte Zellen erreicht. Die Spezialisierung der Zellen wird Zelldifferenzierung genannt. Beispiele für sehr unterschiedliche Zelltypen sind Nervenzellen, Muskelzellen, Blutzellen usw. Sie unterscheiden sich nicht nur in ihrem äußeren Erscheinungsbild, sondern vor allem in ihren Funktionen. Nervenzellen haben ein komplexes, lang gezogenes Äußeres. Sie können sich nicht teilen und sich damit nicht vermehren. Dafür haben sie einen Mechanismus, um mit anderen Nervenzellen zu kommunizieren. Rote Blutkörperchen sind Zellen die sich ebenfalls nicht teilen können, da sie über keinen Zellkern verfügen. Muskelzellen sind Verschmelzungen vieler Zellen und besitzen ein ausgeprägtes Cytoskelett, das zu kontraktile Bewegungen fähig ist.

Im nachfolgenden Teil der Arbeit sind mit Zellen immer eukaryontische Zellen gemeint, da nur diese mehrzellige Organismen formen und damit Zelldifferenzierung betreiben.

2.1.3 Vorgänge in der Zelle

Von den Vorgängen, die in einer Zelle stattfinden, werden im nächsten Abschnitt der (Ernährungs-) Stoffwechsel und die Biosynthese näher beschrieben.

Stoffwechsel findet in der Zelle statt, indem von der Zelle aufgenommene Substanzen biochemisch umgewandelt werden. Die dabei ablaufenden chemischen Reaktionen werden durch Katalysatoren ermöglicht. Diese

⁵ Der 1mm große Fadenwurm *Caenorhabditis elegans* besteht aus ca. 3000 Zellen.

Katalysatoren sind in den Zellen enthaltene Enzyme. Beim Abbau von Nährstoffen wird Energie gewonnen, die in biochemischen Energiespeichern wie ATP für andere, in den Zellen ablaufende Prozesse, zur Verfügung gestellt wird. Die zum Abbau der Nährstoffe benötigten Katalysatoren werden in der Zelle synthetisiert, d.h. die Zelle erzeugt die zur Verdauung der Nahrung notwendigen Substanzen selbst. Außerdem erzeugt die Zelle die für ihren Aufbau und ihre Funktionen erforderlichen Stoffe.

Den meisten nicht pflanzlichen Zellen dienen Fette oder Kohlenhydrate (insbesondere Glucose) als Energiequelle. Diese Substanzen werden über viele biochemische Reaktionen zu CO_2 oxidiert. Die Zelle benötigt für diesen Vorgang Sauerstoff. Für jede Teilreaktion ist ein spezieller Katalysator notwendig.

Als erstes werden große Makromoleküle, wie Proteine, Kohlenhydrate und Fette in kleinere Untereinheiten wie Aminosäuren, einfache Zucker und Fettsäuren gespalten. Dieser Vorgang der Verdauung findet nicht in Zellen, sondern im Verdauungstrakt der höheren Organismen, durch hinzufügen von Enzymen über die Zellen des Verdauungstraktes statt.

Die entstandenen, kleineren Moleküle werden nun von den Zellen aufgenommen. Im Zytoplasma der Zellen werden diese zum Grossteil zu Pyruvat oder zu Acetyl, das an Coenzym A gebunden ist, umgewandelt. Diese Reaktionen sind anaerob. Das Pyruvat wird in die Mitochondrien transportiert, wo es ebenfalls zu einer Acetylgruppe umgewandelt wird. In den Mitochondrien wird das Acetyl (Essigsäurerest) im sog. Zitronensäurezyklus unter Zugabe von Sauerstoff zu CO_2 oxidiert.

Chemische Energie, die für Teilreaktionen bei diesen Stoffwechselprozessen benötigt oder frei wird, wird in den Zellen in ATP gespeichert, bzw. bei Bedarf daraus zurück gewonnen.

Proteinbiosynthese

Bei der Proteinbiosynthese werden, nach dem in der DNS kodierten Bauplan, die Proteine in den Zellen synthetisiert. Dies läuft folgendermaßen ab: im Zellkern wird die DNS an bestimmten Stellen⁶ in RNS umgeschrieben (transkribiert). Dieser Vorgang wird von einem hoch spezialisierten Protein-Komplex der Transcriptase ermöglicht. Die DNS wirkt wie eine Schablone, an der sich RNS-Bausteine anlagern und zu einem m-RNS-Strang (messenger RNS) zusammengefügt werden.

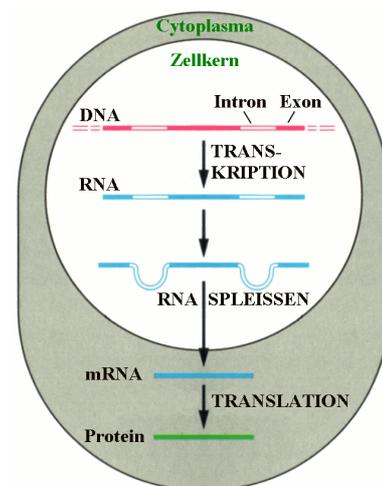


Abb. 2.5: Transkription und Translation [ALB95 S. 121]

⁶ Eine bestimmte DNS-Sequenz, der sog. Promotor, veranlasst die Transkription der DNS.

Der m-RNS-Strang reift anschließend im Zellkern, d.h. es werden durch Spleißung Teile des RNS-Stranges wieder herausgeschnitten, so dass die endgültige m-RNS, die ein Protein codiert, entsteht. Ein gereifter m-RNS-Strang wird aus dem Zellkern über die Zellkernporen heraustransportiert. Die sich jetzt im Zytoplasma befindende m-RNS wird an den Ribosomen, die sich unter anderem an der Oberfläche des Endoplasmatischen Retikulums befinden, in Proteine umgeschrieben (Translation).

Bei der Translation wird eine Folge von drei Basen auf der m-RNS in eine Aminosäure übersetzt.⁷ Dazu sind Bausteine nötig, die diese Verknüpfung ermöglichen. Sie bestehen aus geschlungener RNS, die ein exponiertes Triplet von Basen (den sog. Anticodon) am einen Ende und eine Bindung zu einer speziellen Aminosäure am anderen Ende haben. Diese Bausteine werden t-RNS (translation RNS) genannt. Die Translation an den Ribosomen erfolgt dadurch, dass sich t-RNS an den m-RNS-Strang anlagert und aus der sich daraus gebildeten Aminosäuresequenz die Peptidbindungen des Proteins gebildet werden.

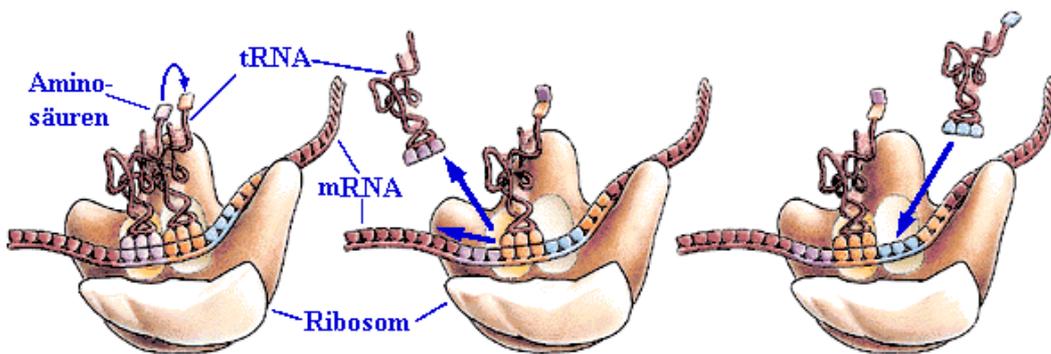


Abb. 2.6: Proteinbiosynthese am Ribosom [1]

2.1.4 Zellwachstum und Zellteilung

Zellen vermehren sich, indem sie ihre Bestandteile vervielfachen und sich dann zweiteilen. Die meisten Zellbestandteile entstehen nicht exakt in doppelter Anzahl, ihre Zahl nimmt während der Wachstumsphase lediglich zu. Die DNS allerdings wird exakt dupliziert.

Das Zellwachstum und die Zellteilung stellen einen Zyklus dar, der in unterschiedliche Phasen eingeteilt wird. Die unter einem Mikroskop deutlich

⁷ Der genetische Code wurde in den frühen 1960er Jahren entschlüsselt. Drei aufeinander folgende Basen auf der RNS stellen einen Codon dar, der eine Aminosäure codiert. Es sind $4^3=64$ Codon-Tripletts möglich. Da nur 20 Aminosäuren in Proteinen gefunden werden, codieren mehrere Codons dieselben Aminosäuren. Weiterhin codieren einige Codons Start und Stopp Signale für die Translation.

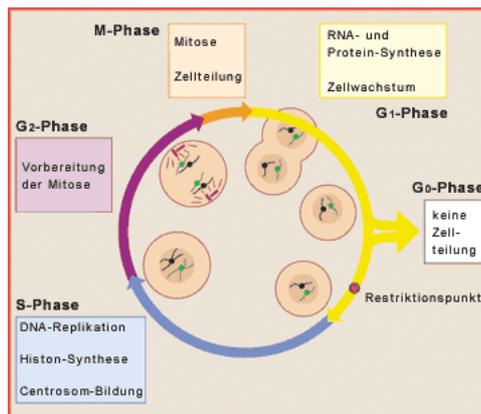


Abb. 2.7: Zellzyklus [MAX02]

sichtbare Phase, wenn die Zelle sich teilt, wird M-Phase (M = Mitose) genannt. Die Phase, die zwischen zwei M-Phasen liegt, heißt Interphase.

Die Interphase gliedert sich in drei Abschnitte, G1 (G = gap), S-Phase (S = Synthese) und G2. Während der S-Phase wird die DNS im Zellkern verdoppelt. Die Zellaktivitäten während G1 und G2 beziehen sich nicht auf die Zellteilung.

Die verschiedenen Zellen eines Organismus haben unterschiedlich lange Zellzyklen. Einige Zelltypen (wie beispielsweise Nervenzellen) teilen sich gar nicht mehr. Entsprechend muss jede Zelle ihren Zellzyklus selbständig regeln. Diese Regelung wird durch bestimmte Faktoren (im Zytoplasma enthaltene Substanzen) erreicht.

In der G1-Phase kann ein Zeitpunkt (Restriktionspunkt) beobachtet werden, nach dem die Zellteilung innerhalb eines festgelegten Zeitrahmens stattfindet. Dieser Zeitrahmen ist bei fast allen Zellen gleich, während die Längen der G1-Phasen je nach Zelltyp stark voneinander abweichen können.

Abgesehen vom Zellzyklus werden auch das Wachstum und die Zellteilung durch bestimmte Wachstumsfaktoren geregelt.

In der frühen Embryogenese ist der Zellzyklus verkürzt. Die G1 und G2-Phasen entfallen und es wechseln sich nur Synthese und Mitose Phasen ab. So kann eine schnelle Teilung und Vermehrung der Zellen stattfinden.

Ablauf der Zellteilung

Während der Interphase des Zellzyklus vermehren sich alle Zellbestandteile (Organellen). In der S-Phase wird die DNS im Zellkern genau verdoppelt, und im Centrosom der Zelle bildet sich ein zweites Centriolenpaar. Während der Mitose wandern diese Centriolenpaare auseinander und bilden die Spindelpole des Spindelapparates. Nachdem sich die Kernmembran aufgelöst hat, wird die DNS in Form von Chromosomen in der Mitte der Spindelpole aufgereiht, und anschließend über den Spindelapparat genau auf die zwei Pole aufgetrennt. Nach der Trennung der DNS bilden sich neue Kernmembranen, es entstehen zwei neue Zellkerne. Anschließend findet die Cytokinese statt, bei der das Zytoplasma durch Zellfurchung aufgeteilt wird. Dabei schnürt sich die Zellmembran zusammen und es entstehen zwei Zellen. Die Spindelpole sind nun die Centrosomen der neuen Zellen und bauen ein Cytoskelett auf.

2.1.5 Genregulation

Vielzellige Lebewesen stammen von einer einzigen Ursprungszelle (der befruchteten Eizelle) ab. Die Zellen eines Körpers sind entsprechend genetisch identisch. Also müssen in jeder Zelle alle genetischen Informationen für die Funktion einer jeden Zelle liegen. Da es in Vielzellern aber phänotypisch verschiedene Zellen gibt, werden diese Zelltypen durch Gen-Kontrolle erzeugt. Die Funktionen bzw. biochemischen Reaktionen, die in einer Zelle stattfinden, werden durch die in der Zelle vorhandenen Substanzen bestimmt. Die wiederum werden durch die Gene in der Zelle erzeugt. Also sind nicht zu jeder Zeit alle Gene in einer Zelle aktiv. Es bedarf eines Regulationsmechanismus, der bestimmt, welche Gene in einer Zelle aktiv sein sollen. Die Differenzierung der Zellen wird durch die Kontrolle der Gen-Expression realisiert.

Die Gen-Expression wird durch Gen-Regulator-Proteine kontrolliert. Diese Regulatoren können positive oder negative Kontrolle ausüben, d.h. bei Anwesenheit des Regulators wird ein Gen aktiv oder inaktiv. Weiterhin gibt es sog. Meister-Gen-Regulatoren, die gleichzeitig eine Vielzahl von Genen steuern.

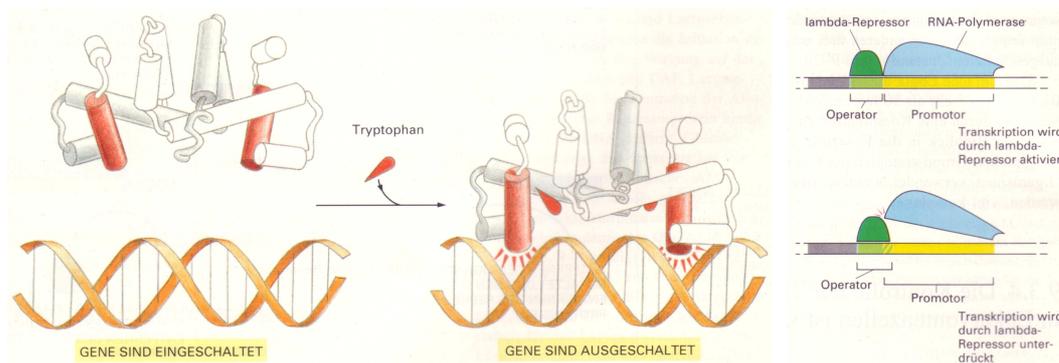


Abb. 2.8: Bindung eines Proteins an DNS [ALB95 S. 493, 495]

Die Kontrolle der Regulatoren kann auf verschiedenen Niveaus der Biosynthese stattfinden:

- Transkriptionskontrolle:

Ein Regulator (Transkriptionsfaktor) kann die Transkription kontrollieren, indem es sich an die DNS vor dem Gen anlagert und so die Arbeit der Transcriptase ermöglicht oder verhindert.

- Processing Kontrolle:

Das „Reifen“ der m-RNS im Zellkern kann auch aktiv reguliert werden. Vor allem kann alternatives Spleißen eines m-RNS-Stranges durch einen Regulator kontrolliert werden. (Ein Gen kann im transkribierten m-RNS-Strang mehrere Introns enthalten, die alternativ entfernt werden können, wodurch verschiedenen Aminosäuresequenzen kodiert werden.)

- Transportkontrolle:

Da der Transport der m-RNS durch die Zellkernmembran ein aktiver Prozess ist, ist auch hier eine Kontrolle möglich, bei der ein „Erkennungsprotein“ den spezifischen Transport erlaubt oder verhindert.

- Translationskontrolle:

Es gibt Translationsrepressor-Proteine, die die Translation eines m-RNS-Stranges in ein Protein verhindern, indem sie sich an die m-RNS heften und damit den Start der Translation an den Ribosomen verhindern.

Ausserdem gibt es Translationsverstärker, die die Translation eines Proteins ermöglichen.

Diese Regulation kann nicht nur durch Proteine, sondern auch durch RNS-Stränge, die sich an m-RNS anlagern, stattfinden.

Die Gen-Regulatoren sind diffusible Faktoren, die in nur geringen Mengen in den Zellen enthalten sein müssen. Weiterhin können sie aus der Zelle ausgeschüttet werden und somit benachbarte Zellen beeinflussen. Das Zellgedächtnis, d.h. die Erinnerung einer Zelle an ihren Zelltypen bzw. welche Gene für die Zellfunktionen relevant oder irrelevant sind, wird durch diffusible Faktoren im Zytoplasma der Zelle reguliert.

2.1.6 Entwicklungsbiologie

Obwohl alle Zellen eines vielzelligen Organismus aus ein und derselben Ursprungszelle entstehen, unterscheiden sie sich phänotypisch. Die Zellen sind zu verschiedenen Zelltypen spezialisiert (z.B. Muskelzellen, Nervenzellen, Blutzellen, Hautzellen etc.). Diese spezialisierten Zellen ordnen sich wohl definiert an. Alle ihre Eigenschaften (auch die Anordnung der Zellen) werden durch das Genom des Lebewesens bestimmt.

Der komplexe Aufbau und die Funktionalitäten der spezialisierten Regionen in multizellulären Organismen werden durch folgende Aktivitäten bestimmt: Zellen wachsen, teilen sich und sterben; sie differenzieren sich, d.h. sie schalten die Produktion bestimmter Proteine, die in ihren Genen codiert sind, an oder ab; sie sondern an ihrer Oberfläche Substanzen ab bzw. nehmen Substanzen auf, und beeinflussen so ihre Nachbarzellen.

Die Embryogenese, die Entwicklung eines Lebewesens aus einer befruchteten Eizelle, kann in drei Phasen unterteilt werden. In der ersten Phase (der Blastogenese) teilt sich das Ei in viele kleine Zellen und bildet einen Hohlkörper (Bauchhöhlenanlage oder Blastula) und das Neuralrohr aus.

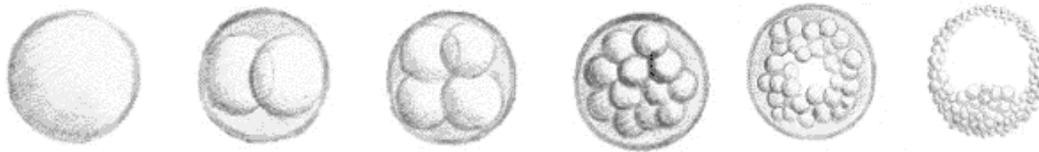


Abb. 2.9: Ausbildung der Blastula[3]

Während der Gastrulation, einer Wanderung von Zellen in die Blastula, bilden sich drei verschiedene übereinander liegende Schichten von Zellen aus. Außen wird das Ektoderm gebildet, aus dem in der späteren Entwicklung die Haut und das Nervensystem entstehen. Innen wird das Entoderm gebildet, aus dem sich anschließend die inneren Organe wie der Darm, die Lunge oder die Leber herausbilden. Und dazwischen bildet sich das Mesoderm, das zu Bindegewebe, Muskulatur und Urogenitalsystem wird.

Die zweite Phase ist die Organogenese, in der sich aus den Anlagen die Organe des Körpers bilden. In der dritten Phase, der Fetogenese, wächst der Fötus bis zur Geburt heran.

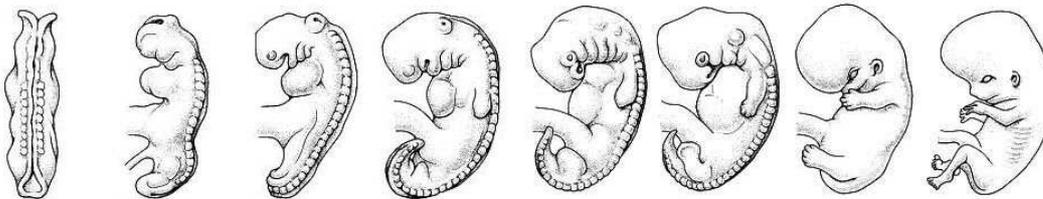


Abb. 2.10: menschliche Embryogenese [4]

Schon bei den ersten Zellteilungen der frühen Embryogenese gibt es eine Ausrichtung der Zellen (die drei Hauptachsen: anterior/posterior, dorsal/ventral, und zentral/lateral). Diese Polarisierung findet durch lokale Ausschüttung von Substanzen statt, die eine Regulation der regionalen Entwicklung bewirken.

Diese Signalsubstanzen (Morphogene) wirken als Netzwerk von Gen-Regulatoren, die eine positionsabhängige Determinierung der Zellen bewirken. Eine determinierte Zelle ist in ihrer weiteren Entwicklung bereits festgelegt, ohne dass sie dafür äußere Anzeichen trägt. Der Konzentrationsgradient eines Morphogens, der durch Diffusion von einer Morphogenquelle entsteht, wirkt als Parameter, damit die Zelle „weiß“ wo sie sich befindet. Diese Positionsinformationen unterscheiden sich zu verschiedenen Zeiten der Entwicklung. Zu Beginn sind die Positionen nur sehr allgemein definiert, im Verlauf der Entwicklung werden Informationen für immer feinere Strukturen zu den bereits vorhandenen, gröberen hinzugefügt.

Beispielsweise wurde bei Experimenten mit der Entwicklung von Extremitäten von Hühnerembryos herausgefunden, dass die verfeinerten Positionsinformationen homolog sind. Aus einem in die Flügelknospe transplantierten Stück Gewebe einer Beinknospe entstand ein Flügel mit Zehen statt Fingern, weil sich die älteren Positionsinformationen, die von der Vorgeschichte der Zellen

abhängen, unterschieden. Die Homologie der feineren Informationen erlaubt dennoch die Ausbildung der Extremität.

Durch Zell-Zell-Wechselwirkung wird der komplexe Bauplan eines Lebewesens in der Entwicklung realisiert. Dazu gehört auch das Zellgedächtnis, durch das Zellen (mit Hilfe von Substanzen in ihren Inneren) wissen, in welchem Zustand der Entwicklung sie sich befinden. Der Determinierung folgt eine Differenzierung zu den verschiedenen Zelltypen.

Während der Entwicklung in der Organogenese wird zusammenhängendes Gewebe gebildet. Der Zusammenhalt wird durch adhäsive Eigenschaften auf der Oberfläche der Zellen realisiert. Dies hilft bei der räumlichen Musterbildung. Auch Zellwanderungen werden durch adhäsive Eigenschaften von Zellen kontrolliert. Besonders bei der Entwicklung der Gliedmaßen kommt es zu Einwanderung von Zellen (Muskelgewebe und Nervenzellen wandern in die Gliedmaßen ein), die richtig positioniert werden müssen.

2.2 Simulationen

Eine in der Wissenschaft erprobte Methode, Zusammenhänge zu analysieren und besser zu verstehen, ist es, Modelle zu bilden und Simulationen zu betreiben.

Modell wird in [BRO04] folgendermaßen definiert:

Modell das,

1) Vorbild, Muster, Gebrauchsmuster.

2) Aufbau, Form, nach der das eigentliche Werk geschaffen wird.

3) [...]

4) vereinfachende bildliche oder mathematische Darstellungen von Strukturen, Funktionsweisen, Verlaufsformen. modellieren, formen, bilden.

Ein Modell wird in einem bestimmten Abstraktionsgrad beschrieben. Beispielsweise reicht die Zeichnung eines LKW aus, um ein stark abstrahiertes Modell eines LKW zu erzeugen. Ist es allerdings das Ziel, mit Hilfe eines LKW-Modells ein Abrechnungssystem für ein funktionierendes LKW-Maut-System zu analysieren, so muss das LKW-Modell weniger abstrakt sein.

Ein nützliches Modell erleichtert das Verständnis der Funktionsweise der Vorgänge oder den Aufbau der Struktur, die analysiert werden sollen. Bildliche Modelle helfen, unsichtbare oder komplexe Objekte einfacher darzustellen und ihren Aufbau zu verstehen (z.B. Atommodell). Mathematische Modelle beschreiben mit Gleichungen oder Regeln Abläufe. Bei nicht vollständig verstandenen Abläufen können mit Hilfe eines Modells Hypothesen über die zugrunde liegenden Vorgänge aufgestellt werden.

Die Abstraktion eines Modells kann idealisierend auf das erzeugte Bild wirken. Das erleichtert unter Umständen die Bewertung der Simulationsergebnisse.

Modelle können beschreibend oder vorhersagend sein. Ein vorhersagendes Modell versucht Aussagen über die Ergebnisse eines Vorganges zu machen, indem es Informationen über dessen Ablauf verarbeitet. Die Vorhersagen werden durch Simulation der Vorgänge mit dem Modell erzeugt.

In [BRO04] wird Simulation folgendermaßen definiert:

„Darstellung technischer, biologischer, ökonomischer u. a. Prozesse oder Systeme durch (mathematische) Modelle. Simulationen erlauben Untersuchungen oder Manipulationen, deren Durchführung am eigentlichen System zu gefährlich, zu teuer oder anderweitig nicht möglich ist.“

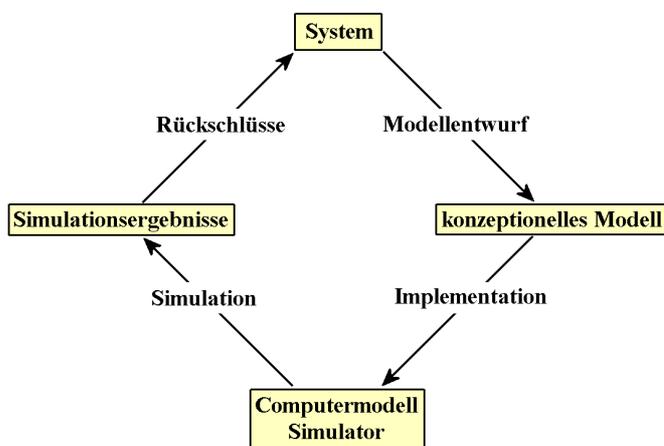


Abb. 2.11: Informationskreislauf bei Simulationen

Eine Simulation benutzt ein Modellsystem, um die Abläufe eines realen Systems nachzubilden. Durch Vergleich der Ergebnisse der Simulation mit dem nachgebildeten System, werden Aussagen über die Güte von Hypothesen möglich, die beim Modellieren vorausgesetzt wurden.

Ziel einer Simulation ist es, entweder hypothetische Modellvorstellungen durch Vergleich mit der Realität zu bestätigen, oder mit einem ratifizierten Modell Vorhersagen zu machen. In jedem Fall ist für eine Simulation ein zugrunde liegendes Modell notwendig.

Simulatoren, die die Simulation eines Modells ausführen, werden meistens speziell für ein konkretes, vorliegendes Modell erzeugt.

Man unterscheidet statische und dynamische Simulationen, deterministische und stochastische, exogene und endogene, diskrete und kontinuierliche. [BIL03]

Eine statische Simulation gibt als Ergebnis nur einmal Werte aus, eine dynamische Simulation funktioniert dagegen mit der Zeit als Komponente und gibt mehrmals Daten aus. Beispielsweise kann die statische Simulation von chemischen Reaktionen die Konzentrationen der Stoffe am Ende der simulierten Reaktion ausgeben. Eine dynamische Simulation gibt die Konzentrationen der beteiligten Stoffe über den Zeitraum, während dem die Reaktion stattfindet, mehrmals an.

Stochastische Simulationen benutzen im Gegensatz zu deterministischen Simulationen zufallsgesteuerte Variablen. Ihre Modelle können bei gleicher Eingabe unterschiedliche Ausgaben liefern. Das Beispiel der chemischen Reaktion kann z.B. die Reaktionsgeschwindigkeit als Zufallsvariable beinhalten [ROD02]. Dies ist insbesondere dann sinnvoll, wenn der tatsächliche Wert eines beliebigen Reaktionsparameters unbekannt ist. Durch zufälliges Testen verschiedener Parameterwerte kann der am besten passende Wert ermittelt werden.

Exogene Simulationen können, während die Simulation aktiv ist, von außen beeinflusst werden, endogene nicht. Bei chemischen Reaktionen entspricht dies etwa dem Zufügen weiterer Substanzen oder einer aktiven Veränderung der Temperatur, während die Reaktion bereits im Gange ist.

Bei diskreten Simulationen sind alle Werte, Variablen und Ereignisse diskret. Kontinuierliche Simulationen arbeiten mit Differentialgleichungen und stetigen Werten. Simulationen von chemischen Abläufen entsprechen, da sie natürliche, analoge Vorgänge sind, kontinuierlichen Simulationen. Ein gängiges Beispiel für diskrete Simulationen sind simulierte Fabrikationen, bei denen die Modelle auf Stückzahlen basieren.

2.3 Visualisierung

Im Volksmund heißt es: „Bilder sagen mehr als tausend Worte“. Da der Mensch im visuellen Kortex über 4 Milliarden Neuronen besitzt, werden die über die Augen aufgenommenen Informationen sehr schnell und effizient verarbeitet. „Was man sieht, versteht man besser.“ Ein anschauliches Bild vermittelt Sachverhalte verständlicher als verbale Beschreibungen.

Visualisierung bedeutet, etwas sichtbar zu machen. Visualisierungen werden gebraucht, um Dinge zu sehen, die sich dem bloßen Auge verschließen. Manches ist einfach nur zu klein, anderes zu weit weg. Die Visualisierung soll dem Betrachter Informationen vermitteln, gleichgültig ob die visualisierten Daten auf optischem Wege gewonnen werden können oder z.B. Teil einer abstrakten Theorie sind. Eine gute Visualisierung vermittelt ein konkretes mentales Bild, stellt optisch erfassbare Zusammenhänge her und hilft dem Betrachter die Hintergründe zu erkennen.

Man kann zwischen wissenschaftlicher Visualisierung und Informations-Visualisierung unterscheiden:

Bei der wissenschaftlichen Visualisierung werden Daten, die naturwissenschaftlichen Ursprungs sind, also meist aus konkreten Messungen oder Berechnungen hervorgehen, visualisiert. Durch lang erprobte Konventionen sowie

direktes Abbilden aus der Natur liegt häufig eine entsprechende, optische Repräsentation nahe.

Bei der Informations-Visualisierung hingegen sind die Daten oft abstrakt, und ein direkter Übertrag in eine optische Repräsentation ist komplizierter.

2.3.1 Klassifikation von Visualisierungen

Visualisierungen können nach folgenden Kriterien klassifiziert werden:

- Der Abstraktionsgrad der visuellen Elemente kann von fotorealistischen über stilisierte bis zu symbolischen Abbildungen variieren.
- Die Daten können durch unterschiedliche Strukturen visualisiert werden. Dazu gehören: Texte, Tabellen, Diagramme, Zeichnungen oder Bilder. Für komplexe Daten kann die Darstellung mit Hilfe einer dreidimensionalen, virtuellen Welt sinnvoll sein.
- Das Zeitverhalten der Darstellung ist ein weiteres Kriterium für die Klassifikation. Es ist zwischen Standbildern und Bewegtbildern zu unterscheiden.
Das Zeitverhalten der Daten ist davon unabhängig zu beurteilen. Die Daten können statisch oder dynamisch in ihrem Zeitverhalten sein. Dynamische Daten sind diskret oder kontinuierlich.
- Visualisierungssysteme können unterschiedliche Grade der Interaktion zulassen. Es sind folgende Interaktionsmöglichkeiten vorstellbar:
 - Navigation in der Präsentation, d.h. Betrachtung der Darstellung aus verschiedenen Blickwinkeln.
 - Interaktion mit dem Simulationsmodell, d.h. Eingriff in die Erzeugung der Daten.
 - Interaktion mit dem graphischen Modell, d.h. die Repräsentation der Daten wird verändert, es werden andere Methoden der Visualisierung benutzt.

2.3.2 Ziele und Güte einer Visualisierung

Die Visualisierung „*soll die Analyse, das Verständnis und die Kommunikation von Modellen, Konzepten und Daten erleichter[n]*“. [SCH00]

Visualisierungen werden zur Erkenntnisgewinnung oder zur Kenntnisvermittlung eingesetzt.

„*Die Visualisierung einer Datenmenge muss prinzipiell expressiv, effektiv und angemessen sein.*“ [SCH00]

Expressiv bedeutet, dass die Daten unverfälscht wiedergegeben werden.

Effektivität wird durch die eingesetzten Visualisierungstechniken bestimmt.

Eine Visualisierung ist *angemessen*, wenn die Relation zwischen Aufwand und Nutzen günstig ist.

2.3.3 Visualisierungsprozess

Der Vorgang der Visualisierung kann mit einer Pipeline beschrieben werden.



Abb. 2.12: Visualisierungspipeline [SCH00]

Der erste Punkt der Visualisierungspipeline ist die Erfassung der Rohdaten, also die Daten, die aus Messungen, Berechnungen, von einem Simulator oder durch manuelle Eingabe aus Tabellen, Listen oder Datenbanken resultieren.

Als nächstes folgt das Filtering. Bei diesem Vorgang werden die Rohdaten aufbereitet. Die Daten können durch Interpolation ergänzt werden, allerdings kann darunter die Expressivität leiden. Die Daten können reduziert oder geglättet werden. Weiterhin können aus den Rohdaten bestimmte interessante Teildaten extrahiert werden.

Nachdem die Daten aufbereitet wurden, müssen sie beim Mapping auf eine visuelle Repräsentation abgebildet werden.

Das Rendering ist der Vorgang, der aus graphischen Beschreibungen von visuellen Repräsentationen der Daten ein digitales Bild erzeugt.

Zum Schluss des Visualisierungsprozesses löst das erzeugte Bild einen optischen Reiz im Auge des Betrachters aus.

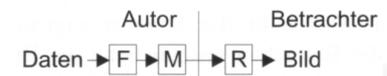
Ein Visualisierungssystem ist lediglich für das Filtering (insofern es notwendig ist) und das Mapping verantwortlich und gibt als letztes ein Bild oder Bewegtbild aus.

Bei einer Visualisierung kann zwischen dem Autor und dem Betrachter unterschieden werden. Der Autor ist der Experte, der die Rohdaten zur Verfügung stellt. Außerdem erzeugt er die Visualisierung. Der Betrachter ist der Leser der visuell aufbereiteten Daten. Er schaut zu und versucht aus der Visualisierung zu lernen. Die Trennung der Rollen des Autors und des Betrachters sind in der Visualisierungspipeline fließend.

Beispielsweise kann ein Autor den gesamten Prozess der Visualisierung übernehmen und dem Betrachter ein fertiges Bild präsentieren. Der Betrachter hat keinen Einfluss auf das Bild.



Eine andere Art der Visualisierung entsteht, wenn der Autor die Visualisierungspipeline nur bis zum Mapping bearbeitet. Der Betrachter kann dann das Erzeugte Bild beeinflussen und damit in der Visualisierung navigieren.



Eine dritte Variante entsteht, wenn der Autor nur die Daten liefert und der Betrachter mit einem zur Verfügung gestellten Visualisierungssystem das



Mapping sowie die Darstellung selbst übernehmen kann.

Abb. 2.13: Autor und Betrachter [SCH00]

Es ergeben sich damit verschiedene Interaktionsmöglichkeiten, die eine Visualisierung bzw. ein Visualisierungssystem zur Verfügung stellt.

2.3.4 Methoden der Visualisierung

Wie oben erwähnt, ist, abgesehen vom Filtering und der damit verbundenen Selektion der Daten, vor allen das Mapping der entscheidende Schritt beim Erstellen einer Visualisierung. Die visuellen Variablen, auf die die Daten gemappt werden, sind dabei ausschlaggebend. Unter anderem ist ein Mapping auf Geometrie und auf Farbe möglich.

Beim Mapping auf Geometrie können Daten vor allem auf die Variablen Position und Größe korreliert werden. Beispielsweise sind Diagramme wie Balken-, Kreis- oder Flächendiagramme typische Anwendungen des Mapping auf Geometrie. Bei einem Balkendiagramm werden verschiedene Daten (ordinale Daten) auf verschieden große und entsprechend positionierte Rechtecke abgebildet. Die Auswahl der Geometrie, auf die die Daten abgebildet werden sollen, ist für die Expressivität und die Effektivität der Visualisierung entscheidend. Oft liegt es in der Natur der Daten, dass eine bestimmte Art der geometrischen Abbildung nahe liegt.

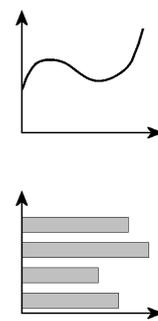


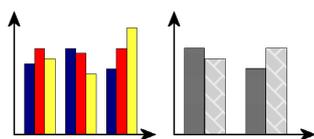
Abb. 2.14: Diagramme

Beim Mapping auf Farbe werden Farben verwendet um Informationen darzustellen. Farbe kann beispielsweise zur Unterscheidung von qualitativ verschiedenen Daten oder zur Betonung von bestimmten Daten verwendet werden. Weiterhin können mit Farben Assoziationen zwischen Daten hergestellt werden. Sollen Farben zur Unterscheidung (nominaler Daten) verwendet werden,

so müssen optisch gut unterscheidbare Farben benutzt werden. Weiterhin sollte die Zahl der verwendeten Farben nicht zu groß sein, da sonst der Betrachter keine Zuordnung der Daten mehr vornehmen kann. Farben können auch benutzt werden um eine Übersicht über ordinale Daten zu geben. Dazu werden Farbskalen verwendet, die Quantität auf Farbe abbildet. Die Temperatur-Farbskala scheint dafür besonders gut geeignet zu sein, da sie als natürlich empfunden wird und somit eine intuitiv richtige Interpretation beim Betrachter bewirkt.



Abb. 2.15: a) Temperatur Farbskala b) Magenta Farbskala [SCH00]



Unterscheidungen können statt mit Farbe auch mit Schraffuren (allgemein durch Texturen) vorgenommen werden.

Abb. 2.16: Diagramme mit Farben bzw. Texturen

Häufig werden mehrere visuelle Variablen benutzt um Daten darzustellen. Selbst einfache Diagramme bedienen sich neben Geometrien auch des Mapping auf Farbe, um Unterscheidungen vorzunehmen.

Ein visuelles Objekt kann aus mehreren, verschiedenfarbigen geometrischen Primitiven aufgebaut sein, und kann damit komplexe Daten repräsentieren.

2.3.5 Visualisierungstechniken

Abgesehen vom Abbilden der Werte oder der Art der Daten auf die visuellen Variablen, wie Position, Größe und Farbe können noch weitere Eigenschaften der Daten visualisiert werden.

Konnektivität

Bei der Visualisierung von Konnektivität werden Beziehungen oder Zusammenhänge der Daten dargestellt. Dabei reicht eine einfache Linie um zwei Punkte visuell miteinander zu verbinden. Die Grundlage zur Darstellung der Konnektivität ist der Graph. Ein Graph besteht aus Knoten, die ein Datum mit den oben beschriebenen Methoden repräsentieren. Die Knoten sind über Kanten miteinander verbunden. Kanten können wiederum Attribute und eine Richtung haben, die über Farbe, Textur, Beschriftung und Form visuell dargestellt werden können.

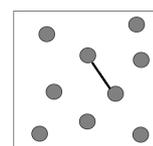


Abb. 2.17: Konnektivität

Raumbezogene Daten

Wenn Daten aus Messungen oder durch Simulationen berechnete Werte einen Raumbezug haben, so werden diese Daten an konkreten Orten gewonnen. Das räumliche Bezugssystem kann punktuell, lokal oder global sein, entsprechend der Natur der Daten. Der Wirkungskreis der Daten muss bei der Darstellung der raumbezogenen Daten mit berücksichtigt werden. Zur Darstellung raumbezogener Daten wird der Messraum (meist zwei- oder dreidimensional) auf das, durch die Visualisierung zu erzeugende Bild, übertragen. Dabei werden bei der direkten Darstellung die Daten an den entsprechenden Punkten visualisiert. Bei der indirekten Darstellung werden nur die Messpunkte angezeigt. Die Selektion eines Messpunktes hat eine separate Auflistung der Messwerte zur Folge. Komplexere multivariate Daten können auch als Kombination von direkter und indirekter Darstellung des Raumbezuges visualisiert werden. In diesem Fall sind z. B. die Informationen mit der stärksten Aussagekraft direkt visualisiert, während zusätzliche Daten nur auf Abruf verfügbar werden.

In jedem Fall ist es notwendig dem Betrachter eine gute Navigation durch den visualisierten Messraum zu ermöglichen.

Zeitbezogene Daten

Häufig haben Daten die visualisiert werden sollen einen Zeitbezug. Da Zeit keine beliebige, unabhängige Variable ist, kommt ihr eine spezielle Rolle bei der Darstellung zu. Als Erstes muss zwischen dem Zeitbezug der Daten und dem Zeitbezug der Darstellung unterschieden werden. Die Daten können statisch zu einem bestimmten Zeitpunkt gewonnen worden sein. Sind Daten, die von verschiedenen Zeitpunkten stammen, vorhanden, so ist der Zeitbezug der Daten dynamisch. Es wird unterschieden ob die Daten kontinuierlich also für jeden Zeitpunkt des betrachteten Zeitraumes vorliegen, oder zu diskreten Zeitpunkten gehören. Wobei kontinuierliche Daten nur aus Berechnungen oder Simulationen hervorgehen können. Allerdings haben diskrete Abtastungen der Daten mit hoher Abtastrate einen ähnlichen Charakter wie kontinuierliche Daten.

Die Darstellung der Daten kann durch eine statische oder eine dynamische Repräsentation erfolgen. Die dynamische Repräsentation kann linear oder nichtlinear sein. Bei Linearität läuft die Zeit stetig ab. Die Geschwindigkeit kann allerdings variieren von Zeitlupe über Echtzeit zu Zeitraffer. Eine nichtlineare Darstellung springt in der Zeit, was ein Springen der Daten zur Folge haben kann. Das kann unbeabsichtigt die Aufmerksamkeit des Betrachters auf sich ziehen.

Dadurch ergeben sich mehrere Möglichkeiten der Kombination von Darstellungen, abhängig vom Zeitbezug der Daten, wie sie in Tabelle 2.1 angegeben sind.

Zeitbezug der Darstellung \ Zeitbezug der Daten	Statisch	diskret abgetastet	dynamisch
statische Repräsentation	Zeitangabe	spezielle Ikonen; Mehrfenster Technik; Zeitangabe	Zeitdiagramme
dynamische Repräsentation	Kamerafahrten; Objektbewegung; Deformationen	sequenzielle Abfolge von Darstellungen;	sequenzielle Abfolge von Darstellungen;

Abb. 2.18: Darstellung der Zeit [5]

Beispielsweise können zu diskreten Zeitpunkten abgetastete Messwerte statisch repräsentiert werden, indem spezielle Ikonen wie z.B. stilisierte Uhren dargestellt werden. Eine dynamische Repräsentation von dynamischen Daten kann durch eine sequenzielle Abfolge von Visualisierungen der einzelnen Zeitpunkte dargestellt werden.

Sollen die Werte der Daten abgelesen werden, eignet sich eher eine statische Repräsentation, oder ein Standbild der dynamischen Repräsentation.

3 Stand der Zellsimulations-Forschung

In diesem Kapitel wird noch einmal kurz zusammengefasst, welche biologischen und biochemischen Grundlagen ein Simulator, der speziell für die Simulation von Zellen vorgesehen ist, berücksichtigen kann oder muss.

Im Anschluss werden einige Konzepte und Realisierungen von Zellsimulationen und Zellvisualisierungen kurz beschrieben. Ziel dieses Kapitels ist es, die aktuell verfügbaren Methoden dahin gehend zu analysieren, inwiefern sie für die Weiterverwendung in dem zu konzipierenden Visualisierungssystem infrage kommen.

3.1 Forschungsgrundlagen für Zellsimulatoren

Es sollen Zelldifferenzierungsvorgänge, wie sie bei der Entwicklung von Lebewesen stattfinden, visualisiert werden. Sie sind bisher jedoch noch nicht vollständig erforscht. Die zugrunde liegenden Mechanismen, die zu einer Differenzierung von Zellen führen, sind bekannt. Das Zusammenspiel von Genen, ihren Verwendungszwecken und den dazugehörigen Regulatoren wurde jedoch nur bei wenigen Arten und speziellen Genen untersucht. Da die Wirkungsweise von Proteinen, die als Regulator wirken können, von ihrem äußerem Erscheinungsbild (Tertiär-, Quaternärstruktur von Proteinen) und ihrer Bindungsfähigkeit an andere Substanzen (z.B. andere Proteine, RNS, DNS) abhängt, reicht das Wissen der Sequenz eines Proteins nicht aus, um seine Wirkung zu verstehen. Vielmehr ist sein räumliches Aussehen und seine im Raum angeordneten wirksamen Interaktionszentren (Aminosäurereste, die mit anderen Molekülbestandteilen Bindungen eingehen können) von Bedeutung. In Zellen werden mehr als zehntausend verschiedene Proteine synthetisiert, die für die meisten Vorgänge verantwortlich sind. Weiterhin sind viele Millionen von Nukleotiden in der Sequenz der DNS vorhanden. Ein Transkriptionsregulator kann sich mit Hilfe einer entsprechenden dreidimensionalen Bindungsstruktur an eine bestimmte Nukleotidsequenz, die an beliebig vielen Stellen innerhalb des gesamten Genoms vorkommen kann, anlagern.

Das Genom von einigen Lebewesen (z.B. *Arabidopsis thaliana*, *C.elegans*, *Drosophila*, Maus, Mensch) [MAX02] wurde zwar „vollständig“ sequenziert, allerdings reicht das Wissen über die Sequenz nicht aus, um zu verstehen, wie diese Lebewesen im Detail funktionieren. Nicht die gesamte DNS kodiert Proteine. Ein Grossteil der DNS in höheren Lebewesen hat keine bekannte

kodierende Funktion. Mit dem Wissen, welche DNS-Sequenzen Gene⁸ sind, kann man die insgesamt mögliche Gen-Expression herausfinden. In Zellen werden aber bei weitem nicht alle möglichen Proteine auch synthetisiert.

Mehr Erkenntnisse über den genauen Ablauf in Zellen zu gewinnen, ist das Ziel aktueller Forschungen im Bereich der Biologie (Zellbiologie, Entwicklungsbiologie). Prognosen zufolge wird es noch mindestens einige Jahrzehnte dauern, bis man die Funktion und den Aufbau eines Lebewesens nur anhand seiner DNS erklären kann, wenn überhaupt.

Von Interesse ist die Analyse der Entwicklung von mehrzelligen Organismen, da bei ihrer Entwicklung regional spezialisierte Zellen entstehen, also Zellen, die durch Gen-Regulation ihre Gen-Expression verändert und sich somit differenziert haben.

Dabei sind Modelle unerlässlich, die eine etwas abstraktere Sicht auf das Problem zulassen. Die Kenntnis von biochemischen Mechanismen in Zellen könnte zur Formulierung von klar strukturierten Regeln führen, wie z.B.:

Wenn Regulator X vorhanden ist, wird Gen A aktiviert. Ist Gen A aktiv und Regulator Y vorhanden, so wird Substanz K erzeugt. Substanz K reagiert mit Substanz L zu Regulator Z. Eine bestimmte Menge von Regulator Z, kombiniert mit Regulator Y, deaktiviert Gen B.

Anhand solcher Regeln und mit Hilfe von Reaktionsgleichungen können die Vorgänge in Zellen simuliert werden.

Soll zusätzlich noch die Entwicklung (Embryogenese) eines ganzen Lebewesens betrachtet werden, so ist auch eine Simulation der Zellen, ihrer Lage und der Beeinflussung von Zellen untereinander zu simulieren.

Nach der Modellierung und Simulation der Entwicklungsvorgänge ist es nötig, die Ergebnisse zu visualisieren. Das Ziel einer Visualisierung ist es, Erkenntnisse zu vermitteln oder neue Erkenntnisse aus gesammelten Daten zu gewinnen. Weiterhin können Hypothesen, die beim Modellieren angenommen wurden, durch Vergleich der resultierenden Daten mit der Wirklichkeit, validiert werden.

⁸ Als Gen wird eine DNS-Sequenz bezeichnet, die über Transkription und Translation zur Synthetisierung eines Proteins benutzt werden kann.

3.2 Vorstellung existierender Zelldifferenzierungs-simulatoren

In diesem Abschnitt werden einige Arbeiten vorgestellt, die sich mit der Simulation von Zellvorgängen beschäftigen. Es wird vor allem die Repräsentation der Zellen und die Modellierung der Vorgänge beschrieben. Weiterhin wird falls vorhanden auf die eingesetzte Visualisierung der Simulationsergebnisse eingegangen.

Eine grundlegende, sehr frühe Arbeit wurde 1952 von Turing [TUR52] vorgestellt. Er lässt die Repräsentation und Mechanik der Zellen zum größten Teil außen vor und beschäftigt sich mit den chemischen Aspekten der Zellentwicklung. Dazu stellt er die mathematischen Grundlagen zur Berechnung eines Modells chemischer Substanzen, den sog. Morphogenen, vor. Der Zustand des Systems von kontinuierlich verteilten Morphogenen wird über deren Konzentration, die sich durch Diffusion und durch chemische Reaktionen ändern, beschrieben. Die Änderung der Konzentration wird über Differentialgleichungen mathematisch formuliert und berechnet. In seinen Ausführungen geht er von einer ringförmigen Verteilung der Zellen bzw. der Morphogenlokalisierung aus. Auch eine dreidimensionale Betrachtung in Form einer Kugeloberfläche, die er mit einer Blastula vergleicht, wird vorgenommen. Er kommt zu dem Resultat, dass es in solchen chemischen Systemen zu stabilen Mustern von Morphogenverteilungen kommt, die die Grundlage für die Morphogenese sein könnten. Seine Arbeit, vor allem die mathematischen Überlegungen, dienen vielen nachfolgenden Arbeiten als Grundlage.

1968 wurde von Aristid Lindenmayer das sog. L-System zur Modellierung von pflanzlicher Morphologie vorgestellt. Ein L-System wird als eine Formale Sprache durch ein Alphabet von Symbolen, ein Startsymbol und eine Menge von Produktionen definiert. Die Symbole repräsentieren die verschiedenen Entwicklungszustände der biologischen Komponenten. Die Produktionen sind Regeln, die die Entwicklung der

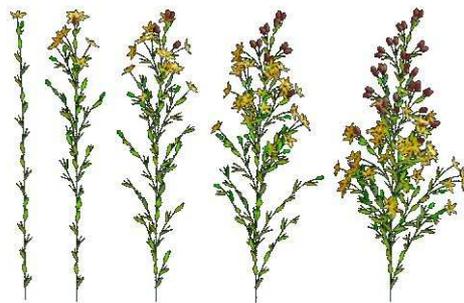


Abb. 3.1: Pflanzenmorphologie durch L-Systeme [PRU97]

Komponenten beschreiben. Das Startsymbol gibt den Anfangszustand an. Über die Anwendung der Entwicklungsregeln werden die einzelnen Stadien der Morphogenese erzeugt. L-Systeme können zur Simulation von Zelldifferenzierung benutzt werden. Die Symbole können verschiedene Zellen sein. Mit einer Produktion können sie sich teilen bzw. differenzieren. Kontextsensitive Produktionen machen komplexere Szenarien möglich. Es ist

auch möglich Zellstrukturen, Zellwachstum und Zellteilung mit L-Systemen zu beschreiben [LAN97]. Allerdings werden L-Systeme hauptsächlich für die Simulation von Pflanzenwachstum eingesetzt. Beispiele für die dadurch möglich Berechnungen von Pflanzenmorphologien werden in [PRU96] und [PUR97] beschrieben.

Die von Agarwal in [AGA93] und [AGA94] vorgestellte *Cell Programming Language* (CPL) ist eine Beschreibungssprache für eine Zelldifferenzierungs-simulation. Die Zellen in dieser Simulation werden zweidimensional als Menge von zusammenhängenden Sechsecken⁹ modelliert. Jeder Zelltyp hat ein *Programm*, das die Eigenschaften und Abläufe in den Zellen steuert. Die Simulation berechnet die Konzentrationen der in den Zellen enthaltenen biochemischen Substanzen. Benachbarte Zellen können Substanzen austauschen. Weiterhin wird die Mechanik, d.h. die Lage, die Größe und die auf die Zelle wirkenden Kräfte, mit berücksichtigt. Die Simulationsergebnisse werden in Form von 2D Bildern, in denen Anhäufungen von Zellen eines Zelltyps erkennbar sind, visualisiert.



Abb. 3.2: Ausbildung von Zellanhäufungen [AGA93]

In [FLE93] wird eine Simulationsumgebung vorgestellt, die die Umgebung der Zellen mit berücksichtigt. Folgende Vorgänge finden in der Umgebung statt: Diffusion von Substanzen, Kollision von Zellen, Adhäsion von Zellen, Zellbewegung und Interaktion benachbarter Zellen über Membranproteine. Eine Zelle bzw. ihr Zustand¹⁰ wird von ihrer Umgebung beeinflusst. Dies kann zu Zustandsänderungen führen. Der Zustand bestimmt das Zellverhalten, was wiederum die Umgebung beeinflussen kann. Die Diffusion und Reaktionen der Substanzen werden über Differentialgleichungen berechnet. Kollisionen, Bewegung und Adhäsion sowie Zellwachstum, Zellteilung und Zelltod bestimmen die Lage der Zellen. Die Simulationsergebnisse werden zweidimensional mit Kreisen für die Repräsentation der Zellen visualisiert. Die Substanzverteilungen werden durch Farbverläufe dargestellt. In [FLE96] wurde die Simulation

⁹ Es sind auch andere Grundstrukturen möglich.

¹⁰ Der Zustand der Zelle wird hier als Genom interpretiert.

erweitert, um auch dreidimensionale Zellen zu simulieren. Die Darstellung erfolgt als Kugeln, die je nach Zustand der Zellen verschieden eingefärbt sind.

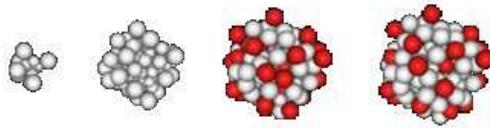


Abb. 3.3: Zellhaufenbildung mit zwei verschiedenen Zelltypen [FLE96]

Die Verteilung von biochemischen Substanzen in der Umgebung wird auch in „The biological Toolbox“ [RAS97] berücksichtigt. Die Umgebung wird in eine 100 x 100 Matrix eingeteilt und die Konzentration der Substanzen wird durch lineare Interpolation bestimmt. Auch in dieser Arbeit wird eine Beschreibungssprache¹¹ vorgestellt, die die Zellen, ihr Verhalten und die Umgebung spezifiziert. Die zweidimensional repräsentierten Zellen werden durch ihre Position, ihre Größe, ihren Typ und die auf sie wirkenden physikalischen Kräfte beschrieben. Neben der Berücksichtigung von Zellteilung, Zelltod, Zellbewegung, wird zwischen einer genotypischen und einer phänotypischen Differenzierung unterschieden, wobei nur die phänotypischen Änderungen durch Einfärbung angezeigt werden. Über Kontakte zwischen Zellen können sich Zellen gegenseitig beeinflussen. Die Visualisierung der Simulation wird in Form einer Video-Datei ausgegeben, die eine Sequenz von 2D-Zustandsbildern ist.

In der Arbeit von Dellaert und Beer [DEL94] wird der Zustand von Zellen, der die Differenzierung ausmacht, nur anhand des Genoms spezifiziert. Die Autoren argumentieren damit, dass die aktiven Gene die Genexpression der Zelle und damit ihren Zustand bestimmen. Die Zustandsänderungen der Zellen werden mit einem genetischen Regulations-Netzwerk

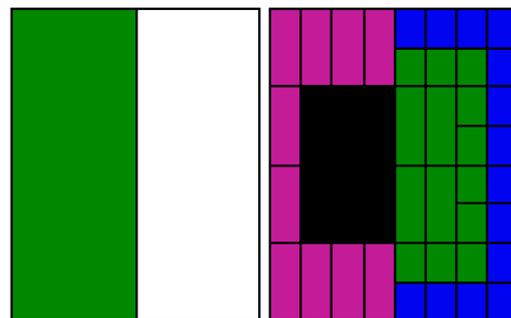


Abb. 3.4: Zelldarstellung durch Rechtecke [DEL94]

berechnet, das in Form eines booleschen Netzwerkes implementiert wurde. Der Zustand der Nachbarzellen fließt in die Berechnungen der Zellen ein. Das zweidimensionale Modell für die Zellen wird bewusst sehr einfach gehalten. Zellen werden als Rechtecke, die sich bei der Zellteilung horizontal oder vertikal teilen können, dargestellt. Der Zustand der Zellen wird durch Farben repräsentiert.

In [BAK02] wird ein Visualisierungssystem für genetische Regulations-Netzwerke präsentiert. Es werden keine Zellen, sondern nur Gene und Proteine

¹¹ Cell Description Language - CDL

dargestellt. Proteine regeln die Aktivität der Gene. Aktive Gene produzieren Proteine. An einem Kreis, das ein Chromosom repräsentiert, werden die Gene punktuell dargestellt. Die Proteine können punktuell oder als Farbverläufe, die die Konzentrationsverteilung visualisieren, dargestellt werden.

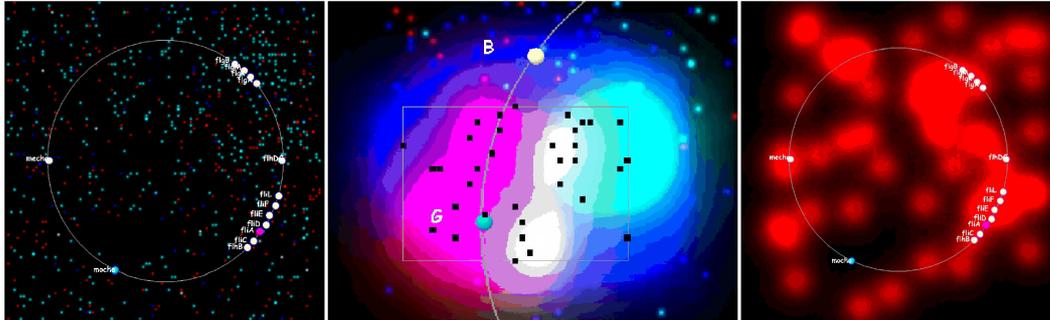


Abb. 3.5: Konzentrationsdarstellungen an einem Chromosom [BAK02]

Ein graphentheoretischer Ansatz für die Modellierung von Entwicklungsvorgängen wird mit Cellerator [SHA01] verfolgt. Zellen werden durch Knoten repräsentiert, die als Information die Lage im dreidimensionalen Raum und ein *Modell* enthalten. Die Kanten stellen die Zellverbindungen der miteinander interagierenden Zellen dar. Auch die Zelllinien also Zellabstammungen werden durch Kanten repräsentiert. Cellerator ist ein Mathematica package, das die biologische Modellierung erleichtert. Biochemische Reaktionen werden in einer Pfeil Notation (wie $\blacksquare \rightarrow \square$) angegeben¹², bei der die Variablen (\blacksquare oder \square) Substanzen darstellen. Sie werden von Cellerator in Differentialgleichungen (DGL) umgewandelt. Interaktionen von Zellen werden gleichfalls als Reaktionen und somit mit DGL angegeben. Auch andere Vorgänge wie z.B. Zellbewegung werden in Form von DGL ausgedrückt. Ereignisse wie Zellteilung oder Zelltod werden durch Schwellenwerte von Variablen gesteuert, was ebenso durch DGL ausgedrückt wird. Solche DGL werden zu *Modellen* zusammengefasst, die den

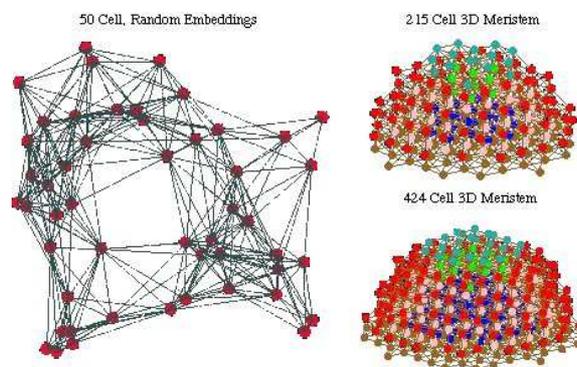


Abb. 3.6: Zellhaufen als Graph mit Knoten und Kanten [SHA01]

¹² Es müssen auch immer die Reaktionsgeschwindigkeiten mit angegeben werden, denn sie sind notwendig, um mit ihnen und den Konzentrationen der Produkte und Edukte die Differentialgleichung zu formulieren. Bei einer Gleichgewichtsreaktion (also Hin- und Rückreaktion und entsprechenden Reaktionsgeschwindigkeiten) entspricht der Quotient der Reaktionsgeschwindigkeiten der Gleichgewichtskonstante des Massenwirkungsgesetzes.

Zellen zugeordnet werden. Bei der Simulation werden die DGL numerisch (von Mathematica) gelöst, und damit die Lage und der Zustand der Zellen berechnet. Die Lösungen der DGL werden über Zeitdiagramme angezeigt, weiterhin ist eine dreidimensionale Ansicht der punktförmigen Zellen und ihrer Verbindungen möglich.

Ein weiteres Werkzeug zum Simulieren von biochemischen Systemen ist Gepasi [MEN96]. Die Kinetik eines Systems von biochemischen Reaktionen wird simuliert. Auch hier werden die angegebenen Reaktionen mit ihren Reaktionsgeschwindigkeiten in Differentialgleichungen umgeformt und mit Angabe der Anfangskonzentrationen numerisch gelöst. Es wird allerdings nicht Bezug auf Zellen, sondern auf ein spezifiziertes Volumen genommen. Die Resultate können anderen Systemen zur Verfügung gestellt oder mit internen Tools in Form von 2D oder 3D Plots dargestellt werden.

Die „System Biology Markup Language“ (SBML)¹³ ist eine standardisierte Beschreibungssprache für Modelle von biochemischen Reaktions-Netzwerken, wie sie in diesem Kapitel bereits beschrieben wurden. Eine Vielzahl von Projekten unterstützen SBML-Modelle. Einige berechnen die Modelle, andere verwenden sie, um Zellvorgänge zu simulieren, und wieder andere stellen die Modelle visuell dar. Unter anderem unterstützen Cellerator, Gepasi und E-Cell das Einlesen von SBML-Modellen.

Mit der „System Biology Workbench“ (SBW) wurde ein Framework erstellt, das Anwendungen als Schnittstelle dient, um miteinander zu kommunizieren und u.a. SBML-Modelle auszutauschen. So ist es beispielsweise möglich, dass eine Simulationsanwendung ihre biochemischen Reaktionen über die SBW-Schnittstelle in Form von SBML-Modellen von einer anderen, extern berechnenden Anwendung simulieren lässt und deren Resultate verwendet. Die SBW Programmbibliotheken sind für verschiedene Programmierumgebungen verfügbar.

Neben den Simulationen von Differenzierungsvorgängen werden auch Simulationen der tatsächlich in Zellen ablaufenden Prozesse gemacht. Dabei wird versucht, die Vorgänge auf einem niedrigen Abstraktionsgrad zu modellieren, um die Funktionsweise einer Zelle bis ins Detail nachahmen zu können. Das „E-Cell Project“ [IAB95] hat als Ziel, sämtliche Vorgänge einer Zelle „in silicio“ [IAB95] zu simulieren. E-Cell ist eine Simulationsumgebung, in der verschiedenartige Modelle integriert werden können.

Ein anderes Ziel, nämlich die Visualisierung und nicht die Simulation von Entwicklungsvorgängen, wird in „The perfect C. elegans Project“ [KIT98]

¹³ Nähere Informationen unter <http://sbml.org/>

verfolgt. Die Arbeit stellt ein in Java/Java3D programmiertes Visualisierungstool vor, das dreidimensionale Bilder der Embryogenese von *C. elegans*¹⁴ erzeugt. Die dafür benötigten Daten werden einer Datenbank (wormbase.org) entnommen, die durch Messung bzw. Beobachtungen an *C. elegans* Embryonen gewonnen wurden. Visualisiert werden die Zellen als Kugeln und der Zelltyp als Farbe, die einer Farbtabelle für die verschiedenen Zelltypen entnommen wird. Um auch Bilder zwischen den in der Datenbank festgehaltenen Zeitpunkten der Embryogenese zu erzeugen, werden die Zellbewegungen durch Interpolation simuliert. Weiterhin werden auch die neuronalen Verbindungen, deren Topologie identifiziert wurde und die entsprechend in der Datenbank vorliegen, durch Linien visualisiert. Der Benutzer kann mit dem Visualisierungssystem interagieren und sich bestimmte Zellen, Zelltypen oder neuronale Verbindungen hervorheben lassen.

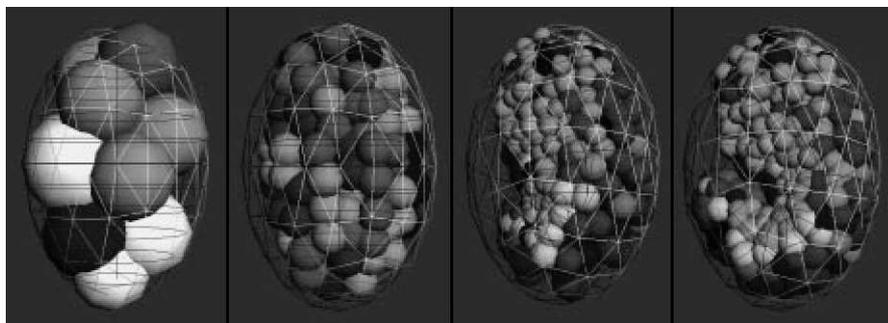


Abb. 3.7: embryonale Entwicklung von *C. elegans* [KIT98]

Neben den bisher vorgestellten Arbeiten, die sich mit der Simulation oder Visualisierung der Zellvorgänge im Allgemeinen oder bezüglich der Embryogenese oder Morphogenese beschäftigt haben, gibt es Arbeiten die sich mit der Simulation von ganz bestimmten Entwicklungsvorgängen befassen. Beispielsweise wird in [IZA04] die Ausbildung von Gliedmaßen simuliert. Es werden Gen Regulations-Netzwerke und die CPL ähnliche Beschreibungssprache *BioLogo* benutzt, um die Zellen und ihre Vorgänge zu beschreiben. Diese Arbeit

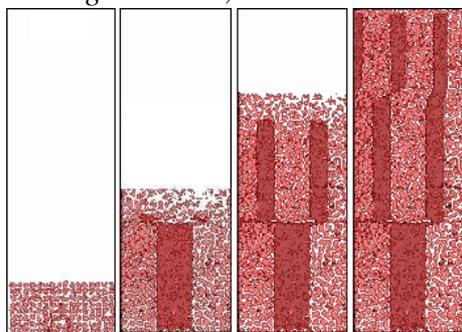


Abb. 3.8: Ausbildung von Extremitäten [IZA04]

wurde in [CHA04] erweitert, um die Morphogenese der Entwicklung einer Extremität auch dreidimensional zu simulieren. Ein weiteres Beispiel ist SimCortex [RYD99], ein Simulator, der die frühe Entwicklung des Zerebralen Kortex simuliert, indem er das Wachstum und die Spezialisierung von Neuronen nachbildet.

¹⁴ *Caenorhabditis elegans* ist ein durchsichtiger Fadenwurm.

Eine weitere interessante Arbeit beschäftigt sich ausschließlich mit der Richtung der Zellteilungen [KUN03]. Dafür wird ein Modell, in dem dreidimensionale Zellen, Proteine und Gene enthalten sind, benutzt, um verschiedene Arten von Zellteilungsmechanismen zu analysieren. Ein Spindelapparat in den immer gleichgroßen Zellen gibt die Richtung der Zellteilung vor und bewirkt im dichtesten Fall eine Kugelanordnung von 12 äquidistanten Kugeln. Es werden Proteine, die die Teilung oder den Zelltod fördern oder hemmen, unterschieden. Weiterhin gibt es Proteine, die den Spindelapparat drehen. Gene werden über Transkriptionsfaktoren gesteuert. Aktive Gene werden zu Proteinen und Transkriptionsfaktoren transkribiert. Die Konzentrationen der Substanzen lösen durch Schwellenwert die Zellteilung, den Zelltod oder die Aktivierung bzw. Deaktivierung von Genen aus. Mit Hilfe eines genetischen Algorithmus werden die jeweils besten Einstellungen ermittelt, um ein kugelförmiges Wachstum der Zellen zu erreichen.

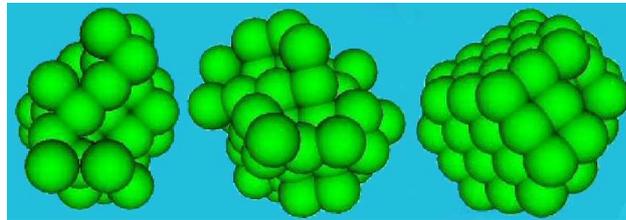


Abb. 3.9: Beispiele verschieden dichter Zellhaufen [KUN03]

3.3 Bewertung der vorgestellten Simulatoren

Ziel dieser Arbeit ist es ein Konzept für ein Visualisierungssystem für Zelldifferenzierungssimulatoren zu erstellen. Nachdem im vorhergehenden Abschnitt existierende Simulatoren vorgestellt wurden, soll in diesem Abschnitt eine Bewertung bzw. eine Analyse der Simulatoren folgen.

Simulatoren / Modelle

Die Simulatoren unterscheiden sich hauptsächlich, durch die in ihren Modellen verwendeten Konzepte: die Repräsentation der Zellen, die Auslöser für die Differenzierung ob Gene oder Substanzen und wie all dies miteinander interagiert.

In den Simulatoren, die eine zweidimensionale Repräsentation der Zellen benutzen¹⁵, werden die Zellen als Kreise, Rechtecke, Punkte oder zusammenhängende Punkte dargestellt. Die dreidimensionalen Simulatoren verwenden Punkte oder Kugeln.

Es werden bei den Simulatoren folgende Zellvorgänge in unterschiedlichem Ausmaß berücksichtigt: die Zellteilung, der Zelltod, das Zellwachstum, die

¹⁵ wie [AGA93], [FLE93], [RAS97], [DEL94], [IZA04] und [RYD99]

Zellbewegungen, Kollisionen von Zellen, Zelladhäsion und Zellinteraktionen. Sie führen zu einer Berechenbarkeit der Mechanik der Zellen.

Die biochemischen Vorgänge, wie sie schon von Turing [TUR52] beschrieben wurden, stellen bei fast allen Simulatoren die Grundlage für die Vorgänge in den Zellen dar. Substanzen, egal ob sie Morphogene, Proteine oder Faktoren genannt werden, sind die Auslöser von Ereignissen, die zur Zelldifferenzierung führen. Die Diffusion, die Erzeugung und die Umwandlung von Substanzen sind die Vorgänge, die die Menge der Substanzen beeinflussen, und damit die Verteilung verändern. Die Unterscheidung von Substanzen in den Zellen und in der Umgebung der Zellen erweitert die möglichen Vorgänge. Diffusion durch die Zellmembran und aktiver Transport durch die Membran können modelliert werden. Weiterhin können in der Umgebung der Zelle vorhandene Substanzen als Signale wirken, um in den Zellen Aktivitäten auszulösen. Diese ganzen Vorgänge werden in Form von biochemischen Reaktions-Netzwerken formuliert und berechnet.

Die Gene werden bei den meisten Simulatoren nicht explizit mit berücksichtigt, da sie über die biochemische Reaktion mit modelliert werden. Ein Gen, das durch einen bestimmter Transkriptionsfaktor aktiviert wird und damit die Produktion eines Proteins ermöglicht, kann auch durch eine Reaktion, mit dem Transkriptionsfaktor als Katalysator, bei der das Protein entsteht, ausgedrückt werden. In [DEL94] wird der umgekehrte Ansatz, nämlich keine Substanzen, sondern nur Gene und deren Aktivität zu simulieren, verfolgt. Auch in [KUN03] werden Gene explizit mit modelliert.

Diese Konzepte, die zur Modellierung der Zelldifferenzierung verwendet werden, führen nur durch entsprechend genaue Modelle zu guten, d.h. die Realität nachbildenden Simulationsergebnissen. Die Hauptschwierigkeit besteht in der Formulierung der Modelle.

Visualisierungen

Das zu erstellende Visualisierungssystem soll möglichst realistische, dreidimensionale Bilder der Zellentwicklung zeigen. Weiterhin sollen die der Differenzierung zugrunde liegenden Auslöser sichtbar gemacht werden, um die Vorgänge der Differenzierung zu visualisieren. Eine einfache Anzeige des Zelltyps, ermöglicht zwar das Wahrnehmen der Differenzierung, nicht aber des Grundes für die Entwicklung.

Die Simulatoren, die eine zweidimensionale Repräsentation der Zellen benutzen, kommen zwar nicht infrage, dennoch gibt es bei ihnen einige interessante Aspekte die Visualisierung betreffend. In [FLE93] werden neben den kreisförmigen Zellen die Verteilungen der Substanzen in der Umgebung als Farbverläufe angezeigt. Solche Farbverläufe werden auch in [DEL04] verwendet, um

Konzentrationsverläufe anzuzeigen. Weiterhin ist Farbe zum Markieren verschiedener Zelltypen ein häufig angewendetes Prinzip. Zweidimensionale Bilder von Zellen können durchaus sinnvoll sein, da sie leicht mit Lichtmikroskopieaufnahmen verglichen werden können.

Bei den dreidimensionalen Darstellungen wie in [FLE96], [KIT98] oder [KUN03] werden Zellen hauptsächlich durch Kugeln repräsentiert. In diesen Darstellungen wird keine Verteilung der Konzentration von Substanzen angezeigt, sondern nur die Einfärbung zum Verdeutlichen von Merkmalen benutzt. In Cellerator [SHA01] werden die punktförmigen Zellen mit Linien verbunden, um die Nachbarschaft bzw. Interaktion kenntlich zu machen. Linien zur Anzeige von neuronalen Verbindungen sind auch in [FLE96] und in [KIT98] vorgesehen. Nur bei „The perfect C. elegans Project“ [KIT98] handelt es sich um ein Visualisierungssystem, das den Benutzer auch interagieren lässt. Allerdings werden keine Simulationsdaten, sondern nur gemessene Werte aus einer Datenbank und vor allem nur die entsprechend positionierten Zellen und deren Zelltyp angezeigt.

Grundlage für das Visualisierungskonzept

Das im nächsten Kapitel erarbeitete Konzept für die Daten, die visualisiert werden sollen, gründet sich auf die biologischen Grundlagen. Nichtsdestotrotz soll die Analyse der existierenden Simulatoren eine Einschätzung, was machbar ist, liefern.

Zellen sollten dreidimensional im einfachsten Fall als Kugeln repräsentiert werden. Der Simulator soll die Zellpositionen berechnen. Dafür sollte er die mechanischen Aspekte, die die Positionen beeinflussen, berücksichtigen. Die Zellteilung und vor allem die Richtung der Zellteilungen, das Zellwachstum, die Zellbewegung sind unmittelbare Faktoren für die Berechnung der Zellpositionen. Indirekt bestimmen auch Kollisionen und Zelladhäsion die Lage der Zellen. Der Tod bestimmter Zellen gibt Platz für die benachbarten Zellen frei.

Neben den Zellen sollten die biochemischen Vorgänge, wie sie in fast allen untersuchten Simulatoren berücksichtigt werden, simuliert werden. Substanzen sollten in den Zellen und in der Umgebung simuliert werden.

Weiterhin ist es sinnvoll, dass auch die Gene vom Simulationsmodell berücksichtigt werden. Denn aktive oder inaktive Gene sind ein weiteres Unterscheidungskriterium für die Visualisierung von differenzierten Zellen, das einfacher zu handhaben ist als die Substanzen, die die Genexpression ausmachen.

4 Konzept

Aufgrund der in Kapitel 2 vorgestellten Grundlagen wird in diesem Kapitel ein Konzept für ein Visualisierungssystem von Zelldifferenzierungsvorgängen entwickelt. Der Aufbau dieses Kapitels folgt der Pipeline des Visualisierungsprozesses¹⁶. Es wird beschrieben, wie die Daten beschaffen sein müssen, welchen Weg sie vom Simulator zum Visualisierungssystem nehmen und schließlich, in welcher Form sie visuell umgesetzt werden.

Das im Rahmen dieser Diplomarbeit vorgeschlagene Konzept wurde durch eine, im Folgenden *D-VISION* genannte, prototypische Implementierung umgesetzt.

4.1 Daten

Üblicherweise wird eine Visualisierung aufbauend auf den zu visualisierenden Daten erstellt. Da bei einer Visualisierung, Daten in Bilder umgewandelt werden, muss für die Erstellung eines Konzeptes für ein Visualisierungssystem als erstes festgestellt werden, was visualisiert werden soll und durch welche Daten es repräsentiert wird. Das soll in den nun folgenden Abschnitten geschehen.

4.1.1 Das Datenproblem

Da keine Modellierung oder Simulation von Zelldifferenzierungsvorgängen existieren, die als Grundlage für das zu erstellende Visualisierungssystem dienen können, müssen im Vorfeld erst einmal die Daten, die zur Visualisierung dienen sollen, spezifiziert werden. Gleichzeitig wird dadurch auch eine Anforderung an die von einem Simulator zu liefernden Daten formuliert. Die Datenspezifikation ist als Vorschlag für künftige Simulatoren zu verstehen.

Ein Visualisierungssystem, das nicht auf einem speziellen Simulator sondern auf einer allgemeinen Datenspezifikation beruht, kann mit mehreren verschiedenen Simulatoren zusammenarbeiten, wenn diese die notwendigen Daten liefern. Es entsteht ein flexibles Visualisierungssystem, das unabhängig von einem bestimmten Simulator ist.

Die Auswahl, welche Daten visualisiert werden sollen, ist in diesem Fall entscheidend. Die Daten sollen in ihrer Struktur so einfach wie möglich sein, um eine reelle Chance zu haben, einen Simulator entwickeln zu können. Denn wenn die vom Visualisierungssystem benötigten Daten in ihrer Struktur zu komplex sind, als dass in absehbarer Zeit ein entsprechender Simulator entwickelt werden

¹⁶ Pipeline: Daten → Filtering → Mapping → Rendering → Bild [SCH00]

könnte, wird es ungenutzt bleiben. Werden jedoch nicht genug Daten berücksichtigt, die dem Anspruch, Zelldifferenzierung von Zellen zu visualisieren, gerecht werden, so ist das Visualisierungssystem für künftige Simulatoren ungeeignet, da nicht dargestellt werden kann, wozu der Simulator eigentlich entwickelt wurde.

Als nächstes folgt eine Analyse, welche Daten aus Sicht der biologischen Abläufe notwendig sind, wobei das Hauptaugenmerk auf der Darstellung der Zellen und ihrem Zustand liegt.

4.1.2 Von den biologischen Grundlagen zum Datenmodell

Die Frage, welche Daten nötig sind, um Zelldifferenzierungsvorgänge zu visualisieren, soll im Folgenden erarbeitet werden. Da Zelldifferenzierung von Zellen vollzogen wird, sind es die Zellen, die visualisiert werden müssen.

Repräsentation von Zellen

Als erstes stellt sich dementsprechend die Frage nach der Repräsentation von Zellen. Das äußere Erscheinungsbild von Zellen kann sehr stark in Form und Größe variieren. Dennoch sind vor allem in der frühen embryonalen Entwicklung die Zellen annähernd kugelförmig. Entsprechend können Zellen der Einfachheit halber als Kugeln repräsentiert werden, für deren Darstellung nur ein Mittelpunkt (also die Position) und eine Größe (Radius oder Volumen) notwendig sind. Sofern ein komplexeres Äußeres für die Repräsentation der Zellen angenommen wird, müssen die Daten dafür vom Simulator geliefert werden, d.h. der Simulator muss abgesehen von der Lage und der Größe auch die äußere Form der Zelle berechnen. Für die Darstellung von Differenzierungsvorgängen, bei denen es vor allem auf die Unterscheidung der Zelltypen ankommt, scheint diese ungleich schwierigere Berechnung unnötig zu sein.

Deshalb soll hier die Darstellung der Zelle mit Hilfe einer Kugel genügen.

Zellinneres

Das Innere einer eukaryontischen Zelle ist durch die Organellen stark strukturiert. Sieht man eine Zelle durch ein Mikroskop an, so kann man vor allem den Zellkern erkennen. Auch die anderen Organellen können gesehen werden. Es stellt sich die Frage, ob es nötig ist, in einem Visualisierungssystem, das die Zelldifferenzierung visualisieren soll, das Innere der Zellen ebenfalls darzustellen. Da die Positionen und die Zahl der Organellen in verschiedenen Zellen unterschiedlich ist, müssen diese Daten von einem Simulator geliefert werden. Beispielsweise die Position des Zellkerns, die Lage und das Aussehen des Endoplasmatischen Retikulums usw. Natürlich könnte auch eine Abstraktion der Organellen für eine Visualisierung ausreichen, also nur die Information „da ist ein Zellkern“ und „da ist ein Mitochondrium“ usw. Allerdings wird bei der Modellierung von Zelldifferenzierung häufig davon ausgegangen, dass die Bestandteile in den

Zellen unbedeutend genug sind, um eine hohe Abstraktion bei der Simulation gerechtfertigt erscheinen zu lassen.

Die Zellorganellen sind zwar in den einzelnen Vorgängen in der Zelle involviert¹⁷. Allerdings sind die Vorgänge an sich und nicht der exakte Ort innerhalb der Zelle, an dem sie ablaufen, relevant.

Das Innere einer Zelle muss folglich nicht dargestellt werden.

Zellvorgänge

Es gibt Vorgänge in Zellen, die beobachtet werden können. Zu diesen von außen sichtbaren Vorgängen zählen das Zellwachstum, die Zellteilung, Zellbewegungen und der Zelltod. Auch die Änderung der äußeren Form der Zelle ist beobachtbar. In der Zelle finden Vorgänge statt, die natürlicherweise nicht beobachtet werden können, wie die Synthese von Proteinen, die Verdoppelung der DNS oder der Abbau von Nährstoffen.

Da das Visualisierungssystem die Zelldifferenzierung darstellen soll, macht es keinen Sinn die inneren Vorgänge zu visualisieren, da sie nur indirekt zur Zelldifferenzierung beitragen. Zwar ist die Anwesenheit eines Genregulators in einer Zelle mitentscheidend für die Gen-Expression der Zelle und damit für ihren Zelltyp, aber eine Visualisierung der Herstellung des Regulators scheint nicht erforderlich zu sein.

Die äußeren Vorgänge hingegen beeinflussen das Aussehen des von den Zellen gebildeten Gewebes. Da die Zellen in ihrer räumlichen Anordnung dargestellt werden sollen, müssen diese Vorgänge berücksichtigt werden. Im Folgenden wird nun analysiert, ob die bisherige Repräsentation einer Zelle durch eine Kugel (Position und Größe) dafür ausreicht.

Beim Zellwachstum, kann beobachtet werden, wie eine Zelle an Umfang zunimmt und manchmal ihre äußere Form verändert. Im Inneren werden durch Stoffwechselprozesse neue Substanzen hergestellt, die in die Zelle eingebaut werden. Beispielsweise werden Lipide erzeugt, die die Membranen der Organellen in der Zelle und die Zellmembran erweitern. Organellen wie die Mitochondrien vermehren sich durch Teilung. Es werden viele verschiedene Proteine synthetisiert, von denen einige das Cytoskelett der Zelle erweitern, andere sich in die Membranen einlagern usw. Da die Darstellung der inneren Prozesse, sowie ein komplexes Äußeres für Zellen oben schon ausgeschlossen wurden, bleibt nur die Zunahme des Umfanges der Zelle, die visualisiert werden

¹⁷ beispielsweise die Mitochondrien beim Stoffwechsel, der Golgi-Apparat zum Ausschütten von Substanzen oder die Ribosomen am Endoplasmatischen Retikulum beim Synthetisieren von bestimmten Proteinen

muss. Dazu reicht die Repräsentation durch eine Kugel aus, denn eine größere Kugel entspricht einer größer gewordenen Zelle.

Die Zellteilung bzw. der Zellzyklus einer Zelle besteht, wie in Abschnitt 2.1.4. dargestellt, aus vielen Vorgängen, die als Phasen verstanden werden können. Es scheint klar zu sein, dass ein Simulator einige dieser Vorgänge berücksichtigen muss, um Zellteilungen und vor allem die Lage der neu entstehenden Zellen zu berechnen. Weiterhin müssen auch benachbarte Zellen und der Platz, der für eine Zellteilung vorhanden ist, berücksichtigt werden. Soll der Vorgang der Zellteilung visualisiert werden, so sind einige Daten notwendig. Da das Zellinnere nicht dargestellt werden soll, ist von der M-Phase nur die Cytokinese für die Visualisierung von Relevanz. Die Zellfurchung (Einschnürung der Membran) und die Entstehung der beiden neuen Zellen resp. Kugeln müssen visualisiert werden. Dieser Aufwand scheint aber für ein Visualisierungssystem, das als Prämisse hat, die Zellzustände darzustellen, unnötig zu sein. Es reicht, die Zelle, die sich teilt, solange als eine Kugel darzustellen, bis die Teilung vollzogen ist, und dann die beiden entstandenen Zellen durch zwei Kugeln an entsprechenden Positionen zu visualisieren.

Ein weiterer, interessanter Aspekt der Zellteilung ist die Betrachtung der Verwandtschaftsverhältnisse von Zellen. Aus einer Zelle gehen bei der Zellteilung zwei Tochterzellen hervor. Dieses exponentielle¹⁸ Wachstum lässt sich als binärer Baum verstehen, in dem die Abstammung einer Zelle, die sog. Zelllinie, zurückverfolgt werden kann.

Die Zellbewegung kann einfach als Änderung der Position der Kugel, die die Zelle repräsentiert, dargestellt werden.

Auch der Zelltod lässt sich mit der bisherigen Repräsentation einfach realisieren. Eine Zelle die stirbt, löst sich auf, d.h. die Zellmembran bricht auf und ihr Inneres erschüttert sich in den extrazellulären Raum. Da das Zellinnere vom Visualisierungssystem nicht dargestellt werden soll, kann eine sterbende Zelle einfach durch Wegnahme der sie repräsentierenden Kugel erfolgen.

Eine Darstellung der Zelle als Kugel reicht aus, um auch äußere Zellvorgänge darzustellen. Die Information, in welche zwei Zellen sich eine Zelle teilt, ist für die Visualisierung ein interessanter Zusatz.

Zelltyp und Zellzustand

Der Zelltyp ist eine Benennung von bestimmten Zellen, die sich durch ihr Äußeres oder durch ihre Funktionsweise von anderen Zellen unterscheiden.

¹⁸ Das Wachstum der Zellanzahl ist insofern exponentiell, als der Zelltod außer Acht gelassen wird.

Begründet ist diese Unterscheidung in der verschiedenen Gen-Expression der Zellen. Da alle Vorgänge in Zellen durch bestimmte Substanzen ausgelöst oder mit bestimmten Substanzen durchgeführt werden, definiert die Summe aller in der Zelle vorhandenen Substanzen den Zustand einer Zelle. Diese Substanzen werden entweder in der Zelle erzeugt, oder von außen aus dem extrazellulären Raum über Transportmechanismen der Membran in die Zelle aufgenommen. Außerdem können Zellen sowohl über die Membran als auch mithilfe des Golgi-Apparates Substanzen aus dem Zellinneren in den extrazellulären Raum ausschütten.

Gene kodieren den Bauplan für Proteine. Ist ein Gen aktiv, so wird das entsprechende Protein in der Zelle synthetisiert. Proteine können Bausteine für die Zelle und ihre Bestandteile sein oder sie können Katalysatoren für Reaktionen sein, die Substanzen in der Zelle umwandeln. Weiterhin können Proteine als Regulatoren fungieren, die bestimmte Vorgänge aktivieren, deaktivieren, fördern oder hemmen. Entsprechend sind Gene für die Funktionalität einer Zelle mit verantwortlich.

Insgesamt scheint der Zustand einer Zelle von den in ihr vorhandenen Substanzen, den sie umgebenden Substanzen und ihrer aktiven bzw. inaktiven Gene abzuhängen.

Ein einfaches Visualisierungssystem könnte den von einem Zelldifferenzierungssimulator berechneten Zelltyp einer Zelle visualisieren. Wobei alle möglichen Zelltypen eine wohl definierte Menge bilden. Eine solche Darstellung ist für eine Analyse der Zelldifferenzierung ungeeignet, da die dem Simulator zugrunde liegenden Entscheidungen über die Klassifikation der Zellen in Zelltypen nicht sichtbar gemacht werden können. Eine reine Auswertung des Zelltyps durch ein solches Visualisierungssystem würde dem Aufwand beim Modellieren und Simulieren nicht gerecht werden. Es scheint vielmehr sinnvoll, den Zellzustand zu visualisieren. D.h. abgesehen von der Zelle selbst soll auch der Zustand der Zelle visualisiert werden. Dazu müssen dem Visualisierungssystem folgende Daten vorliegen:

- die in einer Zelle aktiven Gene
- die Substanzen, die sich in einer Zelle befinden sowie deren Menge (Konzentration)
- die Substanzen, die sich auf der Außenseite der Zelle befinden sowie deren Menge
- der vom Simulator berechnete Zelltyp

Als Substanzen werden hier nicht nur Proteine, sondern auch RNA, Co-Enzyme, Glucose oder alle anderen in Zellen vorkommende Stoffe bezeichnet.

Differenzierung

Es sollen Zelldifferenzierungsvorgänge visualisiert werden. Der eigentliche Vorgang der Zelldifferenzierung ist eine Zustandsänderung der Zelle. Dadurch dass z.B. bisher inaktive Gene aktiv werden, ändert sich die Funktionsweise der Zelle. Sie wird Proteine herstellen, die bisher nicht in ihr vorhanden waren. Diese Proteine können wiederum bisher nicht mögliche Reaktionen möglich machen oder sich an Stellen anlagern, an die bisher kein Protein gepasst hat. Hier spielt vor allem der zeitliche Ablauf eine Rolle. Die Differenzierung einer Zelle darzustellen, bedeutet, ihren sich ändernden Zustand über eine Zeit hinweg aufzuzeigen. Ein Simulator berechnet die Zustände einer Zelle über die Zeit. Die Zeit als Komponente ist dabei wesentlich.

Die Daten der Zelle, d.h. ihre Position, ihre Größe und alle Daten die ihren Zustand definieren, müssen sich auf einen konkreten Zeitpunkt beziehen. Differenzierung wird durch die fortlaufende Betrachtung über einen Zeitraum sichtbar.

4.1.3 Datenmodell

Diese Überlegungen führen zu folgendem Datenmodell:

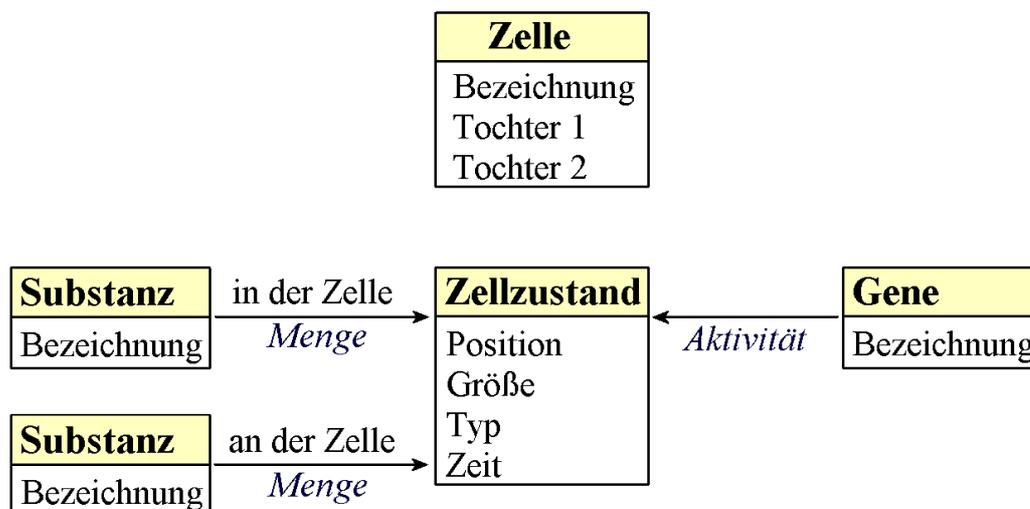


Abb. 4.1: konzeptionelles Datenmodell

Die Entität *Zelle* enthält als Attribut eine eindeutige Bezeichnung einer Zelle. Weiterhin enthält jede Zelle eine Referenz auf die beiden Tochterzellen. Aus diesen Informationen kann eine Baumstruktur der Verwandtschaften von Zellen extrahiert werden.

Die Entität *Zellzustand* stellt den Zustand einer Zelle zu einem bestimmten Zeitpunkt dar. Die Attribute sind: Zellposition, Zellgröße, Zelltyp sowie eine Zeitangabe. Weiterhin gehören zum Zustand einer Zelle die in ihr enthaltenen sowie die an ihrer Oberfläche vorhandenen Substanzen und die aktiven oder

inaktiven Gene der Zelle. Diese Informationen werden über Verknüpfungen mit den Entitäten *Gen* und *Substanz* bereitgestellt. Die Entität *Gen* und die Entität *Substanz* enthalten jeweils als Attribut nur eine Bezeichnung. Durch die Verknüpfungen werden die Informationen des Zellzustandes, welche Gene aktiv oder inaktiv und welche Substanzen in welchen Mengen in oder an der Zelle vorkommen, abgebildet. Jeder Datensatz der Entität *Zellzustand* gehört zu einem konkreten Zeitpunkt.

4.1.4 Anforderungen an einen Simulator

Die im vorhergehenden Abschnitt definierten Daten, die für das Visualisieren von Zelldifferenzierungen verwendet werden sollen, müssen von einem Simulator geliefert werden.

Die hauptsächlich zu visualisierenden Daten sind die Zellen, die durch ihre Position und ihre Größe repräsentiert werden. Ein Simulator muss entsprechend in erster Linie die Lage und Größe von Zellen berechnen. Weiterhin sollen die Zelltypen bzw. die Zellzustände visualisiert werden. Entsprechend muss auch dies von einem Simulator geliefert werden. Um die vom Visualisierungssystem benötigten Daten berechnen zu können, muss ein Simulator Modelle verwenden, die eine entsprechende Berechnung ermöglichen. Ein Modell soll die biologische Wirklichkeit nachahmen.

Für *D-VISION* müssen Modelle die biologischen Vorgänge der Zelldifferenzierung auf einen dreidimensionalen Raum abbilden. Dieser Raum enthält neben den Zellen das extrazelluläre Medium. Die Zellen sind in sich geschlossene, lokale Systeme, die untereinander und mit dem sie umgebenden, extrazellulären Raum interagieren. Jede Zelle nimmt einen eigenen, sich nicht mit anderen Zellen überschneidenden, Raum ein.

Zellen sind keine statischen Objekte, es sind aktive Einheiten, die sich mechanisch und biochemisch verändern. Um eine Zelle zu modellieren müssen ihre Aktivitäten in das Modell einbezogen werden.

Zu den mechanischen Aktivitäten gehören Wachstum, Bewegung und Zellteilung. Neben diesen durch die Zelle selbst verursachten Veränderungen wirken auch von außen, bedingt durch den Druck, der durch die veränderlichen Platzbedürfnisse benachbarter Zellen entsteht, sowie durch Zelladhäsion, physikalische Kräfte auf sie ein.

Zu den biochemischen Aktivitäten gehört der Stoffwechsel der Zelle, der ein kontinuierlicher Prozess ist. Auch die Proteinbiosynthese, bei der aufgrund aktiver Gene Proteine synthetisiert werden, findet stetig¹⁹ in den Zellen statt.

¹⁹ Während der Mitose ist die Transkription für die Proteinbiosynthese unterbrochen.

Stoffwechsel besteht aus Kaskaden chemischer Reaktionen. Mit Hilfe eines mathematischen Modells einer Reaktion können die Stoffkonzentrationen der beteiligten Produkte und Edukte berechnet werden.

Interaktion zwischen Zellen findet durch Übergabe von Stoffen statt. Zellen können Substanzen direkt mit Nachbarzellen oder mit dem sie umgebenden Medium austauschen.

Neben den kontinuierlich in Zellen ablaufenden Prozessen müssen auch Ereignisse modelliert werden. Beispiele für Ereignisse sind die Aktivierung oder Deaktivierung eines Gens, die Zellteilung oder der Zelltod. Sie werden durch spezifische Auslöser hervorgerufen.

All diese unterschiedlichen Aspekte müssen durch Regeln in einem Gesamtmodell verknüpft werden, da die Mechanik und die Biochemie der Zellen miteinander in Wechselwirkung stehen.

Ein solches Modell stellt die Basis für eine Simulation von multizellulärer Entwicklung dar.

Das zeitliche Attribut der Daten, die *D-VISION* benötigt, ist diskret, d.h. die Position, die Größe und der Zustand der Zellen müssen sich immer auf einen konkreten Zeitpunkt beziehen. *D-VISION* generiert ein Standbild, eine Momentaufnahme vom simulierten Modell. Jede darzustellende Zelle muss dem Visualisierungssystem in einem Datensatz mit Position, Größe und Zustand vorliegen.

Die Arbeitsweise des Simulators ist davon unabhängig. Selbst wenn kontinuierliche Prozesse berechnet werden, werden nur die Daten von diskreten Zeitpunkten verwendet. Durch diese Vorgehensweise bleibt es dem Simulator freigestellt, mit Hilfe welcher methodischer Mittel er die Daten berechnet. Beispielsweise können alle Zellen als Gesamtheit oder jede Zelle einzeln nacheinander berechnet werden. Außerdem kann ein Simulator als verteiltes System konstruiert sein, in dem jede Zelle als Simulationseinheit behandelt wird.

Selbst Simulatoren, die nicht alle in Abschnitt 4.2 definierten Daten liefern, können mit *D-VISION* zusammenarbeiten. Die Minimalanforderung, die der Simulator erfüllen muss, damit noch ein brauchbares Bild erstellt werden kann, ist die Berechnung von Position und Größe der Zellen. Ohne diese Information können keine Zellen dargestellt werden.

Liegt dem Visualisierungssystem nur die Information zu Position und Größe der Zellen vor, wird es ein Abbild der Zellen ohne Unterscheidbarkeit der Zelltypen generieren. Jede weitere Information wie Zelltyp, aktive Gene, enthaltene oder umgebende Substanzen ermöglicht eine Klassifikation der Zellen.

4.2 Datenfluss

Da das Visualisierungssystem unabhängig von Simulatoren einsetzbar sein soll, ist ein verteilter Aufbau des Gesamtsystems notwendig. Der gesamte Vorgang vom Erzeugen der Daten bis zum fertigen Bild wird von mehreren Programmen ausgeführt.

4.2.1 Von der Simulation zur Visualisierung

Ein Simulator benutzt Modelle, die ein System nachbilden, um Daten zu berechnen. Ein Visualisierungssystem stellt Daten optisch dar, um daraus Erkenntnisse zu gewinnen. Visualisierung wird eingesetzt, um die Daten eines Simulators zu analysieren. Man kann zwischen einer direkten Kopplung zwischen Simulator und Visualisierung und eigenständigen Systemen unterscheiden. Bei einer direkten Kopplung werden die Daten, die vom Simulator berechnet werden, direkt (in Echtzeit) dargestellt. Die Simulation und die Visualisierung sind eine Einheit, wird die Simulation über Parameter beeinflusst, wirkt sich das auf die Visualisierung aus. Wird sie angehalten, so erhält man ein Standbild der Simulation. Sind allerdings Simulator und Visualisierungssystem unabhängig, so müssen die durch den Simulator berechneten Daten zwischengelagert werden. Zu einem anderen Zeitpunkt können sie dann vom Visualisierungssystem dargestellt werden. Allerdings ist dann keine Beeinflussung der Daten mehr möglich. Um andere Daten zu erhalten, ist ein erneuter, veränderter Simulationsdurchlauf erforderlich. Das Visualisierungssystem kann nur die Art und Weise der Darstellung (Mapping) verändern. Im ersten Fall wird eine Visualisierung für den Simulator auf Maß gefertigt. Im zweiten Fall ist es denkbar, ein vom Simulator unabhängiges Visualisierungssystem zu benutzen. Die Entscheidung, ob eine Visualisierung direkt oder indirekt gemacht wird, wird meist anhand der Dauer der Simulationsberechnung gefällt. Wenn ein Simulator sehr lange für seine Berechnungen braucht, so ist eine direkte Darstellung nicht sinnvoll. Es wird eine Zwischenspeicherung der Daten notwendig. Weiterhin ist es von Vorteil, dass auch ältere, gespeicherte Simulationsdurchläufe erneut visualisiert werden können. So wird auch ein Vergleich verschiedener Simulationen möglich.

4.2.2 Aufbau

Im vorliegenden Fall ist kein Simulator existent. Außerdem ist zu erwarten, dass ein Simulator, der die erforderlichen Daten liefert, aufgrund der Komplexität der Modelle zur Simulation von Zelldifferenzierungsvorgängen, eine lange Berechnungszeit benötigt. Deshalb müssen die erzeugten Daten abgespeichert werden. Die Speicherung der Daten soll vor allem dem Visualisierungssystem dienen. Die Daten werden entsprechend in einer auf die Visualisierung zugeschnittenen, aufbereiteten Form gespeichert. Zwischen Visualisierung und Simulation vermittelt ein System, das für die Speicherung und Bereitstellung der

Daten verantwortlich ist. Dieses, im Folgenden *Aufbereiter* genannte, System verfügt über eine Datenbank, in der die aufbereiteten Daten der Simulationen zwischengespeichert werden. Der *Aufbereiter* stellt eine Schnittstelle, mit der Daten in Form von XML-Dateien entgegengenommen werden, zur Verfügung. Soll vom *Aufbereiter* ein neuer Simulator unterstützt werden, der die Daten nicht über diese Schnittstelle liefert, so kann der *Aufbereiter* dahingehend erweitert werden (durch Programmierung), dass es ihm möglich ist, die benötigten Daten aus den Resultaten des Simulators selbst zu extrahieren.

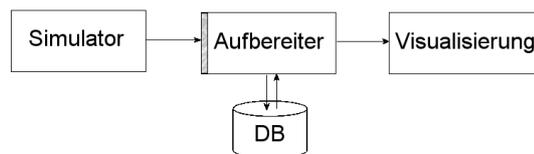


Abb. 4.2: Systemaufbau

Der Aufbau des Gesamtsystems ist in Abb. 4.2 dargestellt. Neben den Komponenten *Simulator*, *Aufbereiter* und *Visualisierungssystem* ist auch der Datenfluss angegeben.

Der Datenfluss zwischen *Aufbereiter* und *Visualisierungssystem* wird durch Methodenaufrufe realisiert. Das *Visualisierungssystem* ruft dabei vom *Aufbereiter* zur Verfügung gestellte Methoden auf, die die gewünschten Daten zurückgeben. Es sind auch entfernte Methodenaufrufe (Client/Server, RMI, CORBA) denkbar, mit denen eine physikalische Trennung von *Aufbereiter* und *Visualisierungssystem* möglich wird. Der *Aufbereiter* könnte über das Internet verfügbar sein, und somit mehreren Instanzen von Visualisierungssystemen als Datenquelle dienen.

Das so aufgebaute Gesamtsystem lässt verschiedene Architekturen für den Simulator zu. Er kann auch als verteiltes System oder als Agenten-System realisiert werden und somit die Vorteile von Grid Computing nutzen. Einem auf mehreren Rechnern laufenden Simulator steht mehr Rechenkapazität zur Verfügung und es können somit auch komplizierte Modelle berechnet werden. In so einem System besteht die Notwendigkeit einer Zentrale, die die Resultate der Simulation sammelt bzw. zur Verfügung stellt. Vorstellbar ist ein Computercluster, in dem jeder Computer einige wenige Zellen berechnet. Ein Zentralrechner verwaltet den Rechner-Pool, vergibt Ressourcen und koordiniert die Kommunikation zwischen den Einheiten.

4.3 Datenvisualisierung

Die Aufgabe des Visualisierungssystems ist es, die Daten, die von einem Simulator geliefert werden, zu visualisieren. Dazu ist es notwendig, die Daten auf optische Repräsentationen abzubilden.

Weiterhin soll ein Benutzer die dargestellten Daten nicht nur betrachten, sondern sie auch genauer untersuchen können. Eine Exploration der Daten muss vom Visualisierungssystem zugelassen werden.

Das Visualisierungssystem soll dem Benutzer Interaktionsmittel zur Hand geben, damit dieser in den Daten navigieren und sie analysieren kann.

4.3.1 Darstellung der Daten

Die Visualisierung soll ein dreidimensionales Bild der darzustellenden Zellen generieren, das die zu simulierenden Zellen räumlich nachbilden soll. Dazu wird ein virtueller dreidimensionaler Raum benutzt, in dem die virtuellen dreidimensionalen Objekte positioniert werden und der beim Rendering (Erzeugen des Bildes) auf den zweidimensionalen Bildschirm des Betrachters abgebildet wird. Hierzu wird eine virtuelle Kamera positioniert und das von ihr aufgefangene Bild erzeugt.

a) Zellen

Wie oben schon öfters erwähnt, liegt auf den Zellen das Hauptaugenmerk der Visualisierung. Die Daten, die eine Zelle repräsentieren, nämlich die Position und die Größe, müssen verwendet werden um die Zellen darzustellen.

Geometrie

Es sollen Kugeln zur Darstellung der Zellen verwendet werden. Die Position der Zellen, die mit drei Koordinaten in den Daten angegeben sind, bestimmt den Mittelpunkt der Kugel.

Um verschiedene Simulatoren zu unterstützen, kann die Größe der Zelle, die in den Daten als ein Zahlenwert angegeben ist, als Radius oder als Volumen angegeben sein. Um die Größenverhältnisse der Zellen korrekt wiederzugeben²⁰ ist es notwendig zu wissen, ob es sich bei der Größe um den Radius oder um das Volumen einer Zelle handelt. Die Zahlenwerte können entsprechend umgerechnet werden.

Die Zellen werden als Kugel-Objekte im virtuellen dreidimensionalen Raum dargestellt, wobei sie nicht als massive Körper, sondern nur durch ihre Oberfläche, also als Hohlkörper repräsentiert werden.

Ein Zellhaufen wird als Kugelhaufen dargestellt. Es obliegt dem Simulator korrekte Werte für die Positionen und die Größe der Zellen zu liefern.

Da die Zellen, je nach Simulator verschieden positioniert und skaliert sein können, ist es notwendig, vor der Darstellung der Zellen den insgesamt benötigten

²⁰ Das Volumen wächst in Relation zum Radius mit der dritten Potenz, der doppelte Radius entspricht dem achtfachen Volumen.

Raum zu bestimmen, um die Abbildungen in den vorhandenen, virtuellen Raum einpassen zu können. Somit muss einem Simulator keine Vorgabe zu den Größenverhältnissen gemacht werden. Allerdings sollten die vom Simulator berechneten Größen und Positionen der Zellen zueinander passen.

Die Kugeln, die die Zellen repräsentieren, sollten sich möglichst nicht überschneiden, da sich Zellen in einem Organismus ja auch nicht überschneiden.

Die so entstehenden Bilder ähneln nur bedingt tatsächlichen Aufnahmen von Zellhaufen der frühen Embryogenese.

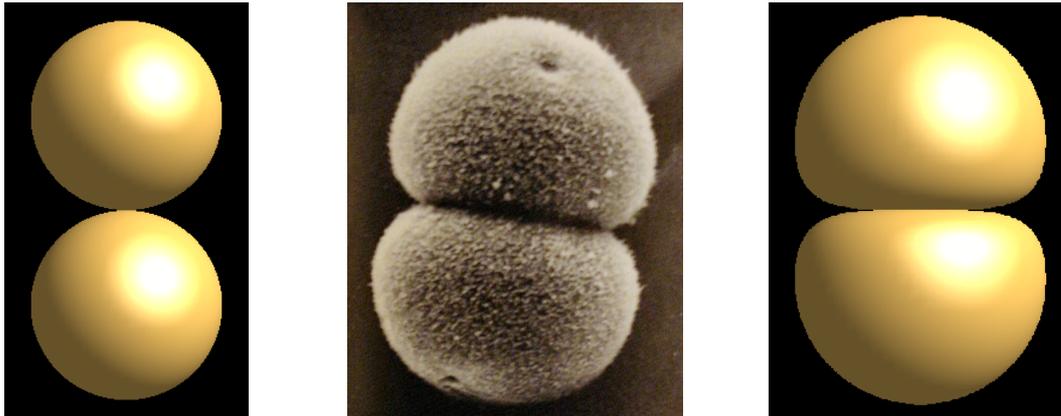


Abb. 4.3: zweizelliger Embryo [ALB95, S.1251]



Abb. 4.4: Zwillingseidotter

Die Zellen sehen tatsächlich eher wie sich gegenseitig deformierende Kugeln aus, wie z.B. mehrere Eidotter in einer runden Schüssel.

Es liegt nahe, das räumliche Aussehen der Zellen auch für die Visualisierung zu benutzen, statt Kugeln also sich gegenseitig deformierende Kugeln zu verwenden.

Die Grundlage für eine Deformation zweier Kugeln ist, dass sie sich überschneiden.

Eine einfache Realisierung dieser Deformation kann durch die Verschiebung der Punkte an der Oberfläche der Kugel im Überschneidungsbereich senkrecht zur Schnittebene erfolgen. Näheres siehe Kapitel 5.4.1. Die so entstehenden Bilder entsprechen in ihrem Aussehen schon viel besser dem tatsächlichen Aussehen der Zellen.

Mit der Darstellung der Zellen als deformierte Kugeln entstehen zwei Probleme:

Erstens müssen sich die Kugeln überschneiden, damit sie sich gegenseitig deformieren können. Wenn ein Simulator allerdings darauf achtet, dass sich die Kugeln nicht überschneiden, so wird er nur entsprechende Daten liefern. Eine Möglichkeit, trotzdem deformierte Kugeln darzustellen, entsteht, wenn die Größe

aller Zellen mit einem benutzerdefinierten Faktor vergrößert wird, da sie sich dann überschneiden werden.

Das zweite Problem entsteht dadurch, dass sich die Größe also das Volumen der Kugeln bei der Deformation verkleinert. Um die Größe der Zellen korrekt wiederzugeben, muss diese Verkleinerung kompensiert werden. Es ist denkbar dies z.B. durch eine Vergrößerung der Kugel vor der Deformation zu erreichen. Allerdings wird eine größere Kugel stärker deformiert und deformiert auch die angrenzenden Kugeln stärker. Dies führt dazu, ebenso wie mehrfache Deformationen einer Kugel, dass die zur Kompensation benötigte Vergrößerung der Kugel nicht exakt berechnet werden kann. Durch mehrere Iterationen von Vergrößerung und anschließender Deformation kann eine Lösung für die Verkleinerung bei der Deformation gefunden werden. Voraussetzung dafür ist natürlich, dass die Daten des Simulators genug Platz für die Zellen lassen.

Mit der Darstellung der Zellen als deformierte Kugel stellt sich das Problem, dass die dargestellten Daten verändert werden könnten und damit die Expressivität der Visualisierung leidet. Allerdings ist der Nutzen durch eine Darstellung der Zellen als deformierte Kugel groß, da ein viel realistischeres Bild der Zellen erreicht werden kann.

Auch ein Simulator könnte mit dieser Form der Darstellung der Zellen rechnen und sich bei der Größe der Zellen auf das Volumen konzentrieren. Damit die Zellen nebeneinander genug Platz finden, muss der Simulator die Platzverhältnisse berücksichtigen.

Material

Da die Kugeln, die die Zellen repräsentieren, in einem virtuellen, dreidimensionalen Raum abgebildet werden sollen, stellt sich die Frage nach ihrer Oberflächengestaltung. Die Oberfläche der Kugeln soll matt und einfarbig sein. Durch die Beleuchtungsrechnung beim Rendering wird ein Eindruck von Räumlichkeit in der zweidimensionalen Abbildung durch Schattierungen und Lichtreflexionen erweckt. Texturen, Mehrfarbigkeit oder sonstige Oberflächenmuster sind nicht zweckmäßig, da sie eine differenzierte Zelloberfläche suggerieren könnten. Allerdings können solche Oberflächengestaltungen eine Identifikation der Zelle ermöglichen. Dies ist auch durch eine Einfärbung der Oberfläche mit einer anderen Farbe möglich. Da Farbe eine preattentive optische Eigenschaft ist, die eine sehr schnelle Identifikation erlaubt [HEA99], soll diese einfache Variante der Markierung von bestimmten Zellen verwendet werden. Durch Einfärbung einiger Zellen können diese im Zellhaufen aufgespürt und identifiziert werden. Eine Markierung von bestimmten Zellen wird möglich, da sich die andersfarbigen Zellen vom Rest abheben.

Da ein Zellhaufen innen liegende Zellen enthält, die von äußeren verdeckt sein können, ist es sinnvoll, eine Möglichkeit bei der Darstellung von Zellen einzubauen, die einen Einblick ins Innere ermöglichen. Dies kann durch eine transparente Gestaltung der Oberfläche erreicht werden. Durch einen transparenten Zellhaufen kann hindurch geblickt werden und es wird erkennbar, wie die Zellen im Inneren angeordnet sind.

Auch 2D:

Eine andere Möglichkeit, den Blick ins Innere eines Zellhaufens freizugeben, ist es, einen Schnitt durch den Zellhaufen als 2D Bild darzustellen. Das so entstehende Bild ähnelt in seinem Aussehen dem Anblick von Zellen durch ein Mikroskop.

Die kugelförmigen Zellen werden in einem solchen Schnittbild als Kreise angezeigt. Eine durchschnittene, deformierte Kugel wird in der zweidimensionalen Darstellung ein deformierter Kreis sein.

Diese Methode ermöglicht ein sehr viel exakteres Bild vom Inneren eines Zellhaufens, da die vor oder hinter der Schnittebene liegenden Zellen nicht angezeigt werden.

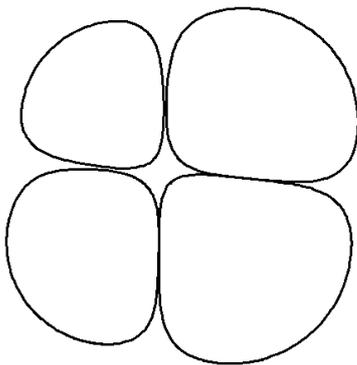


Abb. 4.5: Schnittbild

Das zweidimensionale Bild ergibt sich durch einen Schnitt durch die dreidimensionale Darstellung. Die zur Darstellung des zweidimensionalen Bildes notwendigen Daten werden aus den dreidimensionalen Daten der Zellen und der Lage der Schnittebene berechnet.

Genauso wie in der dreidimensionalen Darstellung werden auch in der zweidimensionalen Darstellung die Zellen durch ihre Hüllen repräsentiert. Im 2D Bild entsprechen die Hüllen den geschlossenen Linien (Kreise oder deformierte Kreise). Auch im Schnittbild können Zellen über eine Einfärbung der Hülle markiert werden. Dies stellt ein Unterscheidungskriterium der Zellen dar. Darüber hinaus kann auch das Innere einer Zelle eingefärbt werden. Diese Einfärbung des Inneren der Zelle kann für die Visualisierung anderer Daten (siehe unten) verwendet werden.

b) Substanzen und Gene

Im Datenmodell sind Substanzen und Gene in den Zellen enthalten. Ihre Menge bzw. ihr Status gehören zu den Daten der Zelle. An der Oberfläche der Zellen befinden sich ebenfalls Substanzen, deren Menge ebenso zu den Daten von Zellen gehören. Sie repräsentieren die Konzentration einer Substanz im extrazellulären Raum, die in direktem Kontakt mit der Zelloberfläche stehen. Die Substanzen bzw. ihre Menge in oder an der Zelle sowie die Gene, die in der Zelle aktiv oder

inaktiv sind, sind raumbezogene Daten, die direkt oder auch indirekt angezeigt werden können. Eine direkte, visuelle Repräsentation all dieser Daten in der Abbildung der Zellen würde das Bild überladen. Die Daten werden stattdessen indirekt dargestellt, indem nach der Selektion einer Zelle, die zu dieser Zelle gehörenden Daten in einer separaten Anzeige aufgelistet werden. Dabei werden die Substanzen, die sich in dieser Zelle befinden, mit ihrer Bezeichnung und ihrer Menge in einer Liste aufgeführt. Ebenso wird eine Liste für die Substanzen und deren Konzentrationen, die sich an der Oberfläche der selektierten Zelle befinden, angegeben. In einer dritten Liste werden die Gene, die sich in der markierten Zelle befinden, sowie die Information ob dieses Gen in der Zelle aktiv oder inaktiv ist, aufgeschlüsselt.

Abgesehen von den Listen, die den Zellinhalt einer Zelle anzeigen, sind weitere Listen für Substanzen und Gene sinnvoll: eine Liste, in der alle in der darzustellenden Simulation vorkommenden Substanzen mit ihren Bezeichnungen aufgezählt sind, und eine Liste, in der alle Gene, die von der Simulation berücksichtigt werden, enthalten sind. Diese beiden Listen ermöglichen einen Überblick über die Substanzen und Gene, die insgesamt als Zellbestandteil in Frage kommen.

Eine Substanz in der Zelle

Ein interessanter Aspekt der Substanzen ist ihre räumliche Verteilung, d.h. ihre Konzentration über die Zellen hinweg. In diesem Zusammenhang ist es sinnvoll, den Konzentrationsgradienten einer Substanz in der räumlichen Anordnung der Zellen sichtbar zu machen. Es wird dadurch möglich, einen Zusammenhang zwischen der Konzentration einer Substanz und dem Zustand einer Zelle zu erkennen. Hierzu wird zunächst eine Substanz ausgewählt, deren Konzentrationsgefälle angezeigt werden soll. In den Zellen wird nun die Konzentration dieser Substanz visuell dargestellt. Die Darstellung der Substanz in den Zellen soll nicht als Visualisierung des Inhaltes einer Zelle, sondern als Darstellung der Konzentration einer bestimmten Substanz über alle Zellen²¹ hinweg verstanden werden. Die Repräsentation der Konzentration einer Substanz wird über Objekte realisiert, die sich im Innern der Zellen befinden.

Eine in [NIE90] vorgestellte Methode²² zur Darstellung von Volumendaten stand für die Idee der Darstellung des Konzentrationsgefälles einer Substanz mithilfe kleiner Objekte Pate. Die Daten werden bei [NIE90] über regelmäßig im Raum

²¹ Die Konzentration einer Substanz wird bei der Darstellung des Konzentrationsgradienten mit der maximalen Konzentration über die Zeit der gesamten Simulation hinweg skaliert, um auch zeitlich unterschiedliche Bilder des Konzentrationsgefälles miteinander vergleichen zu können.

²² die sog. „Tiny Cubes Method“

platzierte kleine Objekte, wie z.B. Würfel – Tiny Cubes (TC) - oder Kugeln, repräsentiert. Sie werden entsprechend des Konzentrationswertes an ihrer Position eingefärbt. Diese Darstellung bietet den Vorteil, dass durch die Zwischenräume der Objekte hindurch weiter hinten liegende Objekte betrachtet werden können.

Anders als bei der in [NIE90] beschriebenen Methode sollen die TC jedoch nicht regelmäßig platziert und nicht entsprechend des darzustellenden Wertes eingefärbt werden. Die TC sind gleichfarbig und sollen allein durch die Anzahl, mit der sie in einer Zelle präsent sind, die Konzentration einer Substanz in dieser Zelle darstellen. Der Vorteil der Methode, durch die Zwischenräume einen Einblick auf die Konzentration in allen Zellen zu erhalten, bleibt bestehen. Damit kann ein Eindruck von der Verteilung dieser Substanz im gesamten Raum gewonnen werden. An Orten, an denen sich TC häufen, ist die Konzentration größer als in Bereichen mit einer weniger dichten TC Verteilung.

Da im Datenmodell die Menge einer Substanz durch einen Zahlenwert angegeben ist, kann dieser Wert je nach Simulator die Konzentration oder die Stoffmenge der Substanz in der Zelle sein. Es ist notwendig zu wissen, ob es sich um Konzentration oder um Stoffmenge handelt, denn für die Darstellung mit der oben beschriebenen Methode ist die Stoffmenge relevant. Eine Konzentration kann unter Berücksichtigung des Zellvolumens in Stoffmenge umgerechnet werden.²³

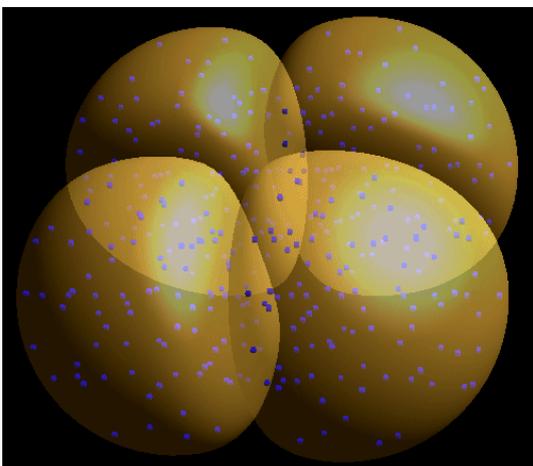


Abb. 4.6: Substanzverteilung in vier Zellen

Es wird also die Stoffmenge einer Substanz in den Zellen dargestellt. Dazu werden entsprechend viele TC in einer Zelle platziert. Die Zellen müssen transparent sein, um die TC sehen zu können. Die Anzahl der TC, die eine bestimmte Stoffmenge einer Substanz in einer Zelle repräsentieren, wird durch eine Skalierung mithilfe der maximal vorkommenden Stoffkonzentration und den maximal darstellbaren TC in einer Zelle errechnet (vgl. Kapitel 5.4.2). Die

Konzentration der Substanz ergibt sich optisch, denn viele TC auf kleinem Raum entsprechen einer hohen Konzentration.

Die Darstellung des Konzentrationsgefälles auf diese Art und Weise ist nicht als exakte Datenvisualisierung zu verstehen. Da nur eine begrenzte Anzahl an TC in einer Zelle sinnvoll sind, weil zu viele Objekte den Blick ins Innere des

²³ Konzentration = Stoffmenge / Volumen

Zellhaufens und damit den Vorteil dieser Darstellung zunichte machen würden, wird die Konzentration stark gerundet wiedergegeben. Kleine Unterschiede in der Konzentration sind unter Umständen nicht visuell erkennbar. Um dies zu verbessern, könnte die Skalierung der darzustellenden Stoffmengen vom Betrachter eingestellt werden. Allerdings kann es dann vorkommen, dass zur Darstellung der Stoffmenge in einer bestimmten Zelle nicht genügend TC in dieser Zelle Platz finden, also die Menge der Substanz in dieser Zelle nicht richtig wiedergegeben werden kann. Werden solche überladenen Zellen vom Betrachter in Kauf genommen, kann er durch Änderung der Skalierung auch noch kleine Unterschiede der Stoffmenge in anderen Zellen sichtbar machen.

Insgesamt ist die Darstellung des Konzentrationsgefälles einer Substanz, als Methode gedacht, um beim Betrachter optisch einen Eindruck der Substanzverteilung hervorzurufen, und damit Zusammenhänge erkennbar werden zu lassen. Zur genauen Datenexploration werden die Mengen der Substanzen wie oben beschrieben für jede Zelle numerisch in einer Liste wiedergegeben.

Sinnvoll kann es auch sein, das Konzentrationsgefälle von mehr als einer Substanz sichtbar zu machen. Dazu wird jede Substanz durch verschiedenfarbige TC repräsentiert. Um tatsächlich noch Zusammenhänge erkennen zu können, scheinen allerdings mehr als drei Substanzen nicht sinnvoll, da sonst die Zellen überladen sind und zusätzlich das Unterscheiden der Farben erschwert wird. Für die unterschiedlichen Substanzen sollten möglichst verschiedene Farben wie Rot, Blau und Grün verwendet werden, damit eine Unterscheidung der unter Umständen sehr kleinen TC möglich ist.

Wird mehr als ein Konzentrationsgefälle dargestellt, stehen in den Zellen auch weniger Positionen für die TC zur Verfügung. Dies hat eine stärkere Quantisierung der Werte zur Folge.

Eine Substanz außerhalb der Zelle

Genauso wie die Konzentration der in den Zellen enthaltenen Substanzen im Zusammenhang mit dem Zelltyp steht, so können auch die an der Oberfläche der Zelle vorhandenen Substanzen den Zelltyp beeinflussen. Um auch diese Zusammenhänge optisch erkennbar werden zu lassen, ist es ebenso sinnvoll den Konzentrationsgradienten der Substanzen im extrazellulären Raum sichtbar zu machen. Da der extrazelluläre Raum nicht direkt, sondern nur indirekt über die Oberfläche der Zellen in den Daten repräsentiert wird, muss die Konzentration einer Substanz im extrazellulären Raum an den Oberflächen der Zellen dargestellt werden.

Die Konzentration einer Substanz wird über das Einfärben der Zelloberfläche realisiert, da bei dem vorgeschlagenen Datenmodell für die gesamte Oberfläche einer Zelle dieselbe Konzentration einer Substanz angegeben ist. Die Farbe der

Zelle ergibt sich über einen Farbgradienten, mit dem die Konzentration auf eine bestimmte Farbe einer Farbskala abgebildet werden kann. Bei verschiedenfarbigen Zellen liegen an den Oberflächen verschiedene Konzentrationen der Substanz, deren Konzentrationsgefälle angezeigt wird, vor.

Damit auch die Einfärbungen der innen liegenden Zellen sichtbar werden können, müssen die Zellen transparent dargestellt werden.

Als Farbgradient können die im Allgemeinen als natürlich empfundene Farb-Temperatur-Skala oder die Magenta-Skala, die die menschliche Farbempfindlichkeit für Violetttönungen ausnutzt, dienen. Eine Abbildung der Skala, um eine optische Einordnung der dargestellten Konzentration vornehmen zu können, ist notwendig.

Mit dieser Art der Darstellung kann nicht mehr als das Konzentrationsgefälle einer einzelnen Substanz im extrazellulären Raum dargestellt werden. Die Anzeige der Konzentrationen mehrerer Substanzen ist somit nicht möglich.

Durch die Art der Darstellung der Konzentrationen von Substanzen in einer Zelle und an deren Oberfläche ist es möglich, beides zu kombinieren. In *D-VISION* können also die Konzentrationsgradienten von maximal drei Substanzen in den Zellen und einer Substanz im extrazellulären Raum dargestellt werden. Die Darstellung der Konzentrationsgefälle mehrerer Substanzen ermöglicht es nicht nur Zusammenhänge zwischen Substanz und Zellzustand, sondern auch eventuelle Zusammenhänge zwischen den Substanzen sichtbar zu machen.

Gene

Neben den Substanzen liefern die Daten vom Simulator auch Informationen über die in den Zellen enthaltenen Gene und deren Aktivität. Neben der oben beschriebenen Möglichkeit sich die Gene einer selektierten Zelle anzeigen zu lassen, kann es auch sinnvoll sein, alle Zellen anzuzeigen, in denen dieses Gen aktiv oder inaktiv ist.

Nach der Wahl eines Genes und der Entscheidung ob dieses Gen aktiv oder inaktiv sein soll, können alle Zellen, für die dieses Kriterium zutrifft, eingefärbt werden. Um die Aktivität des Genes auch bei innen liegenden Zellen zu sehen, müssen die Zellen bei dieser Anzeige ebenfalls transparent sein.

2D Darstellung

In der oben beschriebenen 2D-Darstellung der Zellen ist auch eine Visualisierung des Konzentrationsgradienten einer Substanz möglich. Die Konzentration einer Substanz in einer Zelle wird dazu einer bestimmten Farbe einer Farbskala zugeordnet. Sollen die Konzentrationen einer Substanz in der Zelle dargestellt werden, so werden die Flächen der Zellabbilder entsprechend eingefärbt. Soll das Konzentrationsgefälle einer Substanz im extrazellulären Raum angezeigt werden, so werden die kreisförmigen Zellhüllen in der entsprechenden Farbe dargestellt.

Auch ein Aufzeigen der Zellen, in denen ein bestimmtes Gen aktiv oder inaktiv ist, kann über das Kolorieren der kreisförmigen Zellhüllen realisiert werden.

Die Zellhülle kann in der zweidimensionalen ebenso wie in der dreidimensionalen Darstellung nur eine Farbe gleichzeitig annehmen. Deshalb kann höchstens die Konzentration einer Substanz oder eine beliebige Markierung durch die Einfärbung der Zellhülle dargestellt werden. Zwei Farben könnten zwar als Mischfarbe realisiert werden, der Informationsgehalt würde dabei jedoch verfälscht. Die Mischung erlaubt es nicht mehr, eine optische Zuordnung der Farbe zu den entsprechenden Daten vorzunehmen.

c) Zelltyp und Zellklasse

Zu den Daten der Zelle, wie sie im Datenmodell definiert wurden, gehört auch ein Zelltyp. Bei der Simulation kann ein Simulator eine Zelle zu einem bestimmten Zeitpunkt als zu einem Zelltyp gehörig klassifizieren. Diese Zelltypen sind Benennungen für Zellen. Sie sind, was die Daten für die Visualisierung betrifft, von dem Zustand der Zelle, also den enthaltenen Substanzen, den extrazellulären Substanzen und den Genen, unabhängig. Alle in einer Simulation vorkommenden Zelltypen bilden eine wohl definierte Menge, die in einer Liste angezeigt werden kann.

Auch bei diesem Attribut können, ähnlich wie bei den Genen, alle Zellen, die dem selektierten Zelltyp entsprechen, markiert werden.

Abgesehen von den vom Simulator vergebenen Zelltypen, kann eine Klassifikation der Zellen durch den Betrachter über deren Zustand erfolgen. (näheres siehe Abschnitt 4.3.4) Dazu können über Regeln benutzerdefinierte Zellklassen, die auf dem Zustand der Zellen basieren, verwendet werden. Der Zustand einer Zelle ist durch die in ihr enthaltenen Substanzen, sowie durch die sich im extrazellulären Raum an ihrer Oberfläche befindenden Substanzen und durch die in ihr aktiven bzw. inaktiven Gene definiert. Über einige dieser Merkmale kann eine Regel definiert werden, die eine Zellklasse beschreibt, wobei eine Zelle zur Klasse gehört, wenn sie die Regel erfüllt. Ein Beispiel für eine Regel könnte folgendermaßen lauten: die Zellen, in denen Gen A aktiv aber Gen B inaktiv, in denen Substanz K vorhanden, d.h. die Konzentration größer als 0 ist aber von Substanz L weniger als 2 mol/l enthalten ist und an deren Oberfläche die Substanz X in Konzentration zwischen 1 und 3 mol/l vorkommt, können als dorsale Mesoderm-Zelle²⁴ klassifiziert werden, egal wie viele andere Gene oder Substanzen sonst noch vorhanden sind.

Solche benutzerdefinierten Regeln zur Klassifikation von Zellen werden in einer Zellklassen-Liste angegeben. Nach der Wahl einer Klasse, können die zu dieser

²⁴ vgl. Abschnitt 2.1.6

Klasse gehörenden Zellen in der optischen Repräsentation der Zellen durch Einfärbung sichtbar gemacht werden.

Die Visualisierung von Zellklassen kann auch in der zweidimensionalen Darstellung durch Einfärbung erfolgen.

Eine Kombination der Visualisierung des Zelltyps oder einer Zellklasse mit der Darstellung eines Substanzenkonzentrationsgefälles ist ebenfalls denkbar.

4.3.2 Exploration der Daten

Ziel des Visualisierungssystems ist es, die Daten, die von einem Zelldifferenzierungssimulator berechnet werden, darzustellen und damit einem Betrachter die Möglichkeit zu geben die Resultate der Simulation zu analysieren. Die reine Darstellung aller Daten, vor allem wenn eine große Menge von Daten untersucht werden soll, reicht unter Umständen nicht aus, um Zusammenhänge zu erkennen und Erkenntnisse zu gewinnen. Der Betrachter braucht Mittel, um die Darstellung der Daten zu beeinflussen, damit er die Daten aus verschiedenen Blickwinkeln sehen bzw. einige bestimmte, für ihn interessante Daten fokussieren kann.

In *D-VISION* sollen dem Betrachter mehrere Möglichkeiten der Interaktion mit dem Visualisierungssystem zur Verfügung stehen, die die Form der Darstellung beeinflussen, so dass verschiedene Aspekte der Daten visualisiert werden. Dazu werden die oben beschriebenen Darstellungsmethoden verwendet.

Navigation

Bei der Darstellung der Daten liegt das Hauptaugenmerk auf der dreidimensionalen Visualisierung der Zellen, die als Haufen von (deformierten) Kugeln repräsentiert werden. Der Betrachter muss sich in diesem dreidimensionalen Haufen zurechtfinden, um bestimmte Zellen lokalisieren oder allgemein um sich bestimmte Bereiche näher ansehen zu können.

Hierzu wird eine Abwandlung der in der 3D-Computergrafik üblichen Virtual-Sphere-Metapher eingesetzt, die Michael Chen 1988 erstmals vorstellte²⁵. Dabei ist die Kamera auf einer das zu untersuchende Objekt umgebenden Kugel mit Blickrichtung zum Kugel-Mittelpunkt positioniert und kann auf der Kugel bewegt werden. „Bewegungen der Maus werden als Bewegungen entlang der Längen- und Breitengrade einer virtuellen Kugel interpretiert [...]. Diese Metapher ist mittlerweile stark verbreitet, da sie von Benutzern als intuitiv empfunden wird“ [STO00] und dem menschlichen Verhalten der Untersuchung eines realen Objektes entspricht.

²⁵ in: M. Chen, S.J. Mountford, A. Sellen: *A study in interactive 3-D rotation using 2-D control devices*, Computer Graphics (Siggraph'88 proceedings), 1988

Der Zellhaufen stellt das zu untersuchende Objekt dar. Er kann entsprechend rotiert werden, um ihn von allen Seiten betrachten zu können. Da der Zellhaufen groß sein kann, ist zusätzlich die Möglichkeit, das dargestellte Bild zoomen zu können, von Nutzen, um bestimmte Bereiche näher betrachten zu können. Dies kann durch ein Heranfahren oder Wegfahren der Kamera realisiert werden.

Da die Kamera auf den Mittelpunkt des Zellhaufens gerichtet ist, kann beim Zoomen nur der in der Mitte liegende Bereich stark vergrößert werden. Die Randbereiche liegen beim starken Heranzoomen außerhalb des dargestellten Bildes. Entsprechend ist es notwendig auch eine Bewegung der Kamera zuzulassen, so dass auch die Ränder des Zellhaufens aus größerer Nähe betrachtet werden können.

Ein Betrachter hat für diese dreidimensionale Navigation fünf Freiheitsgrade (DOF²⁶), mit denen er das generierte Bild des Zellhaufens beeinflussen kann. Die Navigation kann folgendermaßen beschrieben werden:

- Das zu untersuchende Objekt liegt im Zentrum einer sie umschließenden Kugel (virtual sphere)
- Die Größe der Kugel kann variiert werden (Zoom → 1 DOF).
- Die Kameraposition entspricht einem beliebigen Punkt auf der Kugeloberfläche (Rotation → 2 DOF)
- Der Blick der Kamera ist auf den Kugelmittelpunkt gerichtet
- Außerdem kann die Kamera entlang einer gedachten Ebene, die die Kugel im Punkt der Kameraposition auf der Kugeloberfläche tangiert, verschoben werden. Die Blickrichtung wird dabei beibehalten, so dass der Fokus nicht mehr auf dem Kugelmittelpunkt liegt. (Bewegung → 2 DOF)

Diese Navigation lässt nur einen Blick von außen auf die Zellen zu. Zwar kann man ganz nah heranzoomen, so dass man die äußersten Zellen mit der Kamera passiert, allerdings werden dann alle Zellen so groß dargestellt, dass man keinen deutlichen Eindruck von der inneren Struktur des Zellhaufens bekommen kann. Eine Möglichkeit, dennoch die inneren Zellen zu sehen, ist es, wie oben schon erwähnt, die Zellen transparent zu machen. Der Betrachter kann entweder alle Zellen oder nur ausgewählte Zellen transparent machen. Ebenso ist es denkbar, dass einige Zellen auch unsichtbar gemacht werden. Werden dann äußere Zellen transparent oder unsichtbar gemacht, so ist ein Blick auf die dahinter liegenden Zellen möglich.

²⁶ DOF = degree of freedom

Eine andere Methode, ins Innere des Zellhaufens zu blicken, ist es, ein zweidimensionales Schnittbild zu generieren, denn durch den Schnitt werden alle Zellen in der Schnittebene sichtbar. Das Schnittbild wird durch eine Ebene erzeugt, die senkrecht zur Blickrichtung positioniert ist. Grundlage für die Schnittebene und damit für das Schnittbild ist die davor eingestellte Ansicht auf das dreidimensionale Modell der Zellen. Die Lage der Schnittebene wird also durch die Blickrichtung bestimmt und enthält zu Beginn den Ursprung. Sie kann aber noch durch Verschiebung entlang der Blickrichtung auf andere Zellen ausgerichtet werden.

Das Schnittbild ähnelt den Bildern, die bei der Lichtmikroskopie von Gewebeproben entstehen. Sind die Proben mehrere Zellschichten dick, so können durch Verlagerung des Fokus verschiedene Schichten untersucht werden. Diese Funktionalität wird durch die Verschiebbarkeit der Schnittebene nachgebildet. Der Betrachter hat die Möglichkeit, den Zellhaufen schichtweise zu durchwandern und sich somit ein Bild des inneren Aufbaus des Zellhaufens zu machen.

Selektion von Zellen

Da die Zustände von Zellen nicht direkt in der dreidimensionalen Darstellung repräsentiert werden, sondern nur als indirekt raumbezogene Daten in einer separaten Anzeige in Form von Listen angezeigt werden, besteht die Notwendigkeit, Zellen selektieren zu können. Um Zellen selektiv transparent oder unsichtbar machen zu können, ist ebenso eine Selektion notwendig.

Die Auswahl einer Zelle kann durch zwei Methoden realisiert werden. Durch das direkte Anklicken einer Zelle im angezeigten Bild wird diese Zelle selektiert. Allerdings ist diese Methode, besonders in der dreidimensionalen Ansicht, nicht allzu effektiv, da nur äußere, gerade sichtbare Zellen selektiert werden können. Auch wenn die außen liegenden Zellen transparent sind, können die inneren Zellen nicht angeklickt werden. Das ist nur möglich, wenn die äußeren Zellschichten unsichtbar geschaltet sind. In der zweidimensionalen Ansicht können nur Zellen, die in der Schnittebene liegen, selektiert werden.

Die zweite Methode ergibt sich über eine Auflistung aller Zellen im dargestellten Zellhaufen. Eine Zelle kann durch direkte Auswahl aus der Liste selektiert werden. Zwar kann somit jede auch noch so weit innen liegende Zelle gewählt werden, allerdings ist keine gezielte Selektion aufgrund der geometrischen Position möglich, da die Zuordnung, welche Zelle der Liste welcher im Bild entspricht, nicht erkennbar ist.

Selektierte Zellen werden durch Einfärbung ihrer Hülle im dargestellten Bild visualisiert.

Wenn eine Zelle selektiert ist, so können ihre Daten, also ihr Zellinhalt, ihre Position und ihre Größe, ihr Zelltyp sowie ihre Zugehörigkeit zu einer Zellklasse (alphanumerisch) eingesehen werden.

Wenn eine Zelle selektiert ist, ist es möglich, über ihre Entfernung zu anderen Zellen diejenigen zu ermitteln, die als Nachbarzellen in Frage kommen. Mithilfe eines entsprechenden Kontrollwerkzeuges, das alle Nachbarzellen anzeigt, können diese dann direkt ausgewählt werden. Dadurch kann der Betrachter von Zelle zu Zelle durch den Zellhaufen navigieren.

Weiterhin ist es sinnvoll auch mehr als eine Zelle selektieren zu können, um beispielsweise mehrere Zellen transparent oder unsichtbar zu machen. Die Selektion mehrerer Zellen kann wiederum durch die Liste oder durch direktes Anwählen im Bild geschehen.

Eine Inversion der Selektion der Zellen, d.h. alle bisher nicht selektierten Zellen werden selektiert und umgekehrt, kann auch sehr brauchbar sein. Beispielsweise können alle Zellen eines bestimmten Zelltyps markiert werden (siehe nächster Abschnitt). Die Auswahl wird invertiert und die nun selektierten Zellen unsichtbar gemacht. Dadurch wird die geometrische Anordnung der Zellen dieses Zelltyps sichtbar.

Eine Selektion mehrerer Zellen ist also sinnvoll, um mit diesen Zellen weitere visuelle Aktionen durchzuführen zu können.

Zellen markieren

Durch die Einfärbung ihrer Hülle in der dreidimensionalen wie in der zweidimensionalen Darstellung können Zellen visuell von anderen Zellen abgehoben und somit markiert werden. Die Markierung und damit gezielte Selektion von Zellen kann nach mehreren Kriterien erfolgen:

- Über eine Substanz

Nach der Wahl einer Substanz aus der Liste aller in der Simulation vorkommenden Substanzen und der anschließenden Entscheidung, ob diese Substanz in der Zelle oder im extrazellulären Raum an der Oberfläche der Zelle vorkommen soll, kann eine Bedingung über die Menge formuliert werden. Die Bedingung wird durch ein mathematisches Vergleichssymbol ($<$, \leq , $=$, \geq , $>$) sowie einen numerischen Wert definiert. Alle Zellen, die den Angaben entsprechen, also beispielsweise Substanz F im Inneren mit einer Menge ≥ 3 enthalten, werden markiert.

- Über ein Gen

Die Auswahl eines Genes und die Angabe, ob es in der Zelle aktiv oder inaktiv sein soll, bestimmen das Kriterium, nach dem die Zellen markiert werden.

- Über einen Zelltyp oder einer Zellklasse

Alle Zellen eines Zelltyps oder einer Zellklasse können markiert werden. Dazu ist nur eine Auswahl des Zelltyps oder der Zellklasse zu treffen.

Eine Kombination von verschiedenen Kriterien zur Auswahl von Zellen wird über benutzerdefinierte Zellklassen ermöglicht; mehr dazu in Abschnitt 4.3.4.

Konzentrationen anzeigen

Eine weitere Visualisierungstechnik, mit der die Daten der Zellen genauer in Augenschein genommen werden können, ist die Darstellung des Konzentrationsgradienten einer Substanz. Die Zusammenhänge zwischen Zelltyp oder besser der Differenzierung einer Zelle und ihrem Zellzustand können damit optisch wiedergegeben werden.

In der dreidimensionalen Ansicht sind bis zu drei verschiedene Substanzen wählbar, dessen Konzentrationsgefälle über den Zellhaufen hinweg in den Zellen durch TC angezeigt werden. Die TC sind je nach Substanz verschieden eingefärbt. Damit alle TC zu sehen sind, werden alle Zellen transparent dargestellt.

In der zweidimensionalen Ansicht kann das Konzentrationsgefälle von nur einer Substanz über die Einfärbung des Zellinneren realisiert werden.

Die Anzeige der Konzentrationsverteilung im extrazellulären Raum wird über Einfärbung der Hüllen realisiert. Eine gleichzeitige Markierung von Zellen ist entsprechend nicht möglich, da dies ebenfalls über die Einfärbung der Hülle visualisiert wird.

Die Zuordnung der Menge einer Substanz auf eine Farbe wird durch Farbskalen ermöglicht. Dies wird in der dreidimensionalen Ansicht für die Substanzen an der Oberfläche der Zellen und in der zweidimensionalen Ansicht für die Substanz innen und die Substanz außen angewendet. In der zweidimensionalen Ansicht sollten zwei verschiedene Farbskalen verwendet werden um die Verteilungen besser abschätzen zu können. Wenn allerdings innen und außen die Konzentrationen der gleichen Substanz dargestellt werden sollen, kann auch nur eine Farbskala Verwendung finden. Bei Zellen, die innen und außen die gleiche Konzentration einer Substanz haben, werden die Hülle und das Zellinnere in derselben Farbe dargestellt.

4.3.3 Zeitkomponente

Bisher wurde bei der Beschreibung der Darstellung und der Interaktionsmöglichkeiten des zu konzipierenden Visualisierungssystems der Zeitaspekt der Daten außen vor gelassen. Zelldifferenzierung ist ein zeitlicher Vorgang. Eine Zelle ändert ihren Zustand über die Zeit. Entsprechend werden Zelldifferenzierungssimulationen die Vorgänge in den Zellen über die Zeit nachbilden.

Der zeitliche Aspekt der Daten wird im Datenmodell über ein Zeitattribut der Zustandsdaten der Zellen berücksichtigt. Alle Zustandsdaten mit der gleichen Zeitangabe stellen die Daten des Zellhaufens zu diesem Zeitpunkt dar. Die Zeitangaben in den Daten beziehen sich immer auf einen konkreten Zeitpunkt, d.h. alle Daten, die die gleiche Zeitangabe haben, stellen eine Momentaufnahme des Zustands aller Zellen im Zellhaufen dar. Die gesamten Daten ergeben sich durch aufeinander folgende Momentaufnahmen. Je zeitlich dichter diese Momentaufnahmen sind, desto kontinuierlicher sind die Zustände der Zellen in den Daten wiedergegeben.

Das Visualisierungssystem stellt solche in den Daten gespeicherten Momentaufnahmen der simulierten Zellen optisch dar. Es werden Standbilder eines kontinuierlichen Prozesses visualisiert. In diesen Standbildern sind die oben beschriebenen Methoden der Exploration der Daten möglich. Um nun auch den zeitlichen Aspekt der Daten zu berücksichtigen, ist es nötig, sich von Standbild zu Standbild zu bewegen. Ein solcher Zeitschritt bewirkt die Darstellung der nächsten Momentaufnahme der Daten.

Abgesehen vom natürlichen Zeitablauf können Rückwärtsschritte oder Zeitsprünge gemacht werden. Durch einen Zeit-Schieberegler kann der Betrachter der Visualisierung zu jedem beliebigen Zeitpunkt innerhalb der Simulation springen. Er kann einen Zeitschritt nach vorne oder nach hinten ausführen. Somit kann er den zeitlichen Ablauf des Zustandes der Zellen verfolgen.

Bei Zeitschritten oder Zeitsprüngen bleiben die Darstellungsparameter erhalten. Beispielsweise ist eine selektierte Zelle auch nach einem Zeitschritt selektiert. Ist eine Zelle im nächsten Zeitschritt allerdings nicht mehr vorhanden, weil sie abgestorben ist oder sich geteilt hat, so kann sie natürlich nicht mehr selektiert sein. Liegen hingegen die Informationen über die Zelllinien und damit das Wissen, in welche zwei Zellen sich die selektierte Zelle geteilt hat, vor, so werden bei einer Zellteilung beide Tochterzellen selektiert. Somit kann man nach einem Zeitsprung erkennen, welche Zellen aus einer bestimmten, selektierten Zelle hervorgegangen sind.

Wurden z.B. die Zellen einer Zellklasse durch Auswahl dieser Zellklasse markiert, so werden auch beim Ablauf der Zeit zu jedem Zeitpunkt alle Zellen, die zu dieser Klasse gehören, markiert sein.

Wird das Konzentrationsgefälle einer Substanz angezeigt, so kann man die zeitliche Änderung des Gradienten verfolgen.

Weiterhin ist es vorstellbar die Standbilder zeitlich ablaufen zu lassen. Es entsteht eine Animation. Für eine Wiedergabe der Einzelbilder als Animation ist ein zeitlicher Faktor nötig, der die Geschwindigkeit der Wiedergabe der Einzelbilder festlegt.

Die simulierten Zellvorgänge brauchen auch in der Natur ihre Zeit. Gemessen wird üblicherweise die Dauer eines Zellzyklus, also die Zeit, die zwischen zwei Zellteilungen vergeht. Der Zellzyklus ist sehr unterschiedlich lang. Er kann bei Leberzellen eines ausgewachsenen Säugers über ein Jahr dauern. In der frühen Embryogenese dauert der Zellzyklus je nach Spezies zwischen 8 und 60 Minuten [ALB95, Kap. 17.1]. Ein Simulationsmodell sollte entsprechend auch diesen zeitlichen Ablauf nachbilden. Zwischen zwei vom Simulator gelieferten Momentaufnahmen liegt ein ganz bestimmter Zeitraum. Werden die Standbilder dieser Momentaufnahmen in genau diesem zeitlichen Abstand angezeigt, so wird die Visualisierung in Echtzeit erfolgen. Da die Vorgänge etliche Zellzyklen dauern, ist eine schnellere Wiedergabe im Zeitraffer sinnvoll.

Ein begrenzender Faktor für die Wiedergabe einer Animation ist die Leistungsfähigkeit des Rechners, der die Visualisierung darstellt. Da für jedes Einzelbild etliche Berechnungen für die dreidimensionale Darstellung vorgenommen werden müssen, kann die Wiedergabe von Einzelbildern sehr stockend sein.

Das Datenmodell ermöglicht es, dass die zeitlichen Abstände zwischen den Momentaufnahmen unterschiedlich groß sind. Diese Eigenschaft erschwert jedoch eine flüssige Wiedergabe der Animation.

Eine Änderung der Darstellungsparameter während der Wiedergabe ist zwar denkbar, kann aber unter Umständen zu Fehlinterpretationen der dargestellten Daten durch den Betrachter führen.

Neben der direkten Navigation durch die Zeit mit Hilfe eines Zeit-Schiebereglers ist auch die Navigation mit Hilfe einer selektierten Zelle möglich. Mit Hilfe eines entsprechenden Navigationselementes kann zur Mutterzelle oder zu einer der Tochterzellen gesprungen werden. Beim Sprung zurück zur Mutterzelle, wird zu dem Zeitpunkt gesprungen, an dem die Mutterzelle sich gerade noch nicht geteilt hat. Die Auswahl einer der Tochterzellen verursacht einen Sprung vorwärts in der Zeit zu dem Zeitpunkt, an dem sich die zuvor selektierte Zelle eben geteilt hat. Somit ist es möglich sich zeitlich durch den Zellhaufen entlang der Zelllinien zu bewegen. Voraussetzung dafür ist natürlich, dass die entsprechenden Daten vom Simulator geliefert werden.

4.3.4 Definition von Zellklassen

Da *D-VISION* die Visualisierung von Zelldifferenzierung zum Ziel hat, müssen neben der Darstellung der Zellen vor allem die verschiedenartigen Zellen hervorgehoben werden. Abgesehen vom Zelltyp, ein im Datenmodell für jede Zelle enthaltenes Attribut, das vom Simulator vergeben wird, stehen für das Kennzeichnen verschiedener Zellen die Daten, die den Zustand der Zelle repräsentieren, zur Verfügung. Der Zustand der Zellen wird von drei

verschiedenen Merkmalsarten bestimmt, nämlich den Substanzen in der Zelle, den Substanzen im extrazellulären Raum, die sich an der Oberfläche der Zellen befinden, und den Genen in der Zelle. Jedes Merkmal, also die Konzentration einer Substanz in oder an der Zelle oder die Aktivität eines Genes, ist Teil des Zustandes der Zelle. Allerdings wird es einige Merkmale geben, die insbesondere für eine bestimmte Funktionalität der Zelle zuständig sind und damit die Differenzierung dieser Zelle mit verantworten. Es ist aber davon auszugehen, dass nicht nur ein Merkmal die Differenzierung einer Zelle auslöst. Vielmehr sind es Kombinationen von mehreren Merkmalen, die die Gesamtfunktionalität der Zelle ausmachen. Um dem gerecht zu werden, enthält das Visualisierungssystem das Konzept der Zellklassifikation.

Zellklassen werden durch Regeln, die die Merkmale von Zellen betreffen, definiert. Die verwendeten Regeln sind boolesche Ausdrücke und aus einzelnen booleschen Argumenten aufgebaut. Die Argumente werden mit logischen Verknüpfungen wie *UND* und *ODER* sowie Klammern zu einer Regel zusammengesetzt. Außerdem kann der boolesche Wert eines Argumentes durch Negation invertiert werden. Als Argumente werden in ihrer Aktivität bzw. in ihrer Menge exakt spezifizierte Gene und Substanzen benutzt. Das boolesche Argument für ein Gen überprüft, ob dieses Gen aktiv oder inaktiv ist. Durch einen mathematischen Vergleich mit einem Zahlenwert ($<$, \leq , $=$, \geq , $>$) wird ein boolesches Argument für Substanzen formuliert. Erfüllen die Merkmale einer Zelle die Regel einer Klasse, so gehört sie zu dieser Klasse.

Das Konzept der Zellklassen ist ein mächtiges Werkzeug in der Hand des Benutzers des Visualisierungssystems. Mithilfe einer entsprechend umfangreichen Regel können Zellen klassifiziert und lokalisiert werden, die sich zu einem bestimmten Typ differenziert haben. Genauso können durch eine einfache Regel Zellen mit einem bestimmten Merkmal gefunden werden. Die Zellklassen dienen der Exploration und Analyse der Daten.

Die Zellklassen sind Bestandteil des Visualisierungssystems und damit unabhängig vom Simulator. Zellklassen werden nicht vom Simulator vorgegeben, sondern vom Benutzer des Visualisierungssystems erstellt. Um sie nicht bei jeder Benutzung neu erstellen zu müssen, ist es möglich die Klassendefinitionen abzuspeichern. Dabei sind drei verschiedene Arten von Klassendefinitionen möglich. Globale Klassendefinitionen sind in allen Simulationen verwendbar. Lokale Klassendefinitionen sind nur für die aktuelle Simulation, allerdings auch in späteren Aufrufen, verfügbar. Temporäre Klassendefinitionen können nur in der aktuellen Simulation und auch nur bis zum Laden einer anderen Simulation oder dem Beenden des Visualisierungssystems benutzt werden. Entsprechend müssen die globalen und die lokalen Klassendefinitionen persistent gemacht werden. Sie werden in der Datenbank des Aufbereiters gespeichert. Folgendes

Verhalten ergibt sich daraus. Beim Öffnen des Visualisierungssystems werden die globalen Klassendefinitionen eingelesen. Wird eine Simulation geladen, werden zusätzlich die entsprechenden lokalen Klassendefinitionen der Datenbank entnommen. Sie stehen somit dem Benutzer zur Verfügung. Abgesehen vom Erstellen neuer Klassendefinitionen können vorhandene eingesehen, verändert und gelöscht werden. Bei der Verwendung eines zentralen Aufbereiters, auf den mehrere Instanzen des Visualisierungssystem zugreifen, sollte für das Editieren und Löschen von Klassendefinitionen eine Rechteverwaltung implementiert werden, damit nicht die abgespeicherten Klassen anderer Benutzer willkürlich verändert oder gelöscht werden können.

4.4 nicht in das Konzept aufgenommene Ideen

Dieser Abschnitt stellt einige weitere Ideen vor, die nicht in das Konzept aufgenommen wurden.

Als erstes wird die Verwendung anderer oder erweiterter Daten beschrieben. Anschließend wird auf die Visualisierung eingegangen. Es werden Ideen zur Erweiterung und alternative Visualisierungstechniken besprochen.

Daten

Die vorhandenen Daten bzw. ihre Natur sind für den Visualisierungsprozess entscheidend. Entsprechend ist im vorliegenden Fall, bei dem es keine vorgegebenen Daten gab und sie erst spezifiziert werden mussten, die Wahl welche Daten dargestellt werden sollen, ausschlaggebend für den Rest des Konzeptes. Ziel war es, basierend auf den biologischen Grundlagen eine möglichst einfache Struktur der Daten vorzugeben, damit möglichst viele Simulatoren in der Lage sein werden die Daten zu liefern, damit das konzipierte Visualisierungssystem zum Einsatz kommt.

Es ist natürlich denkbar, dass in künftigen Forschungen herausgefunden wird, dass ganz bestimmte, mit dem vorgestellten Datenmodell nicht abbildbare Vorgänge in Zellen für die Zelldifferenzierung verantwortlich sind, die entsprechend visualisiert werden sollten.

Ein Simulator könnte nicht nur mit der abstrakten Repräsentation der Zellen in Form von Kugeln, sondern mit komplexeren Formen der Zellen rechnen. Um dies zu visualisieren, könnten die Zellen z.B. in ein Raster aus kleinen Würfeln eingetragen werden. Eine Zelle würde durch eine zusammenhängende Menge von lokalisierten Würfeln definiert. Dies ermöglicht es, die Zelle nicht nur als geschlossene Einheit zu betrachten, sondern das Innere der Zelle in funktionale Bereiche (z.B. Organellen) aufzuteilen, die äußere Form der Zellen, Zellteilung, Zellwachstum und die mechanischen Einflüsse der Nachbarzellen sehr viel detaillierter darzustellen.

Allerdings erfordert eine solche Darstellung neben den entsprechenden Kenntnissen auch mehr Berechnungen.

Im vorliegenden Konzept wird die Zelle als Blackbox betrachtet, in der die Vorgänge unabhängig von der genauen Lokalisierung im Zellinneren stattfinden. Sollte allerdings herausgefunden werden, dass z.B. ein bestimmtes Organell und die darin ablaufenden Prozesse für die Zelldifferenzierung entscheidend sind, könnte ein entsprechendes Zellmodell die Darstellung des Zellinneren erforderlich machen. So könnte es notwendig werden, das entsprechende Organell in allen seinen Funktionen zu visualisieren.

Die Zelladhäsion, die Zellverbindungen wie Tight- und Gap-Junktions und die Mechanik der Zellen werden in *D-VISION* nur durch die Lage der Zellen sichtbar. Sie sind unter anderem für die Morphogenese in der Entwicklung von Lebewesen verantwortlich [ALB95, Kap. 16, 19, 21]. Sollen diese Verbindungen oder die mechanischen Kräfte, die auf die Zellen einwirken, in der Visualisierung dargestellt werden, so muss die Information in das Datenmodell eingebettet werden, damit sie visualisiert werden kann.

Das gleiche gilt für einen Simulator, der mehrere Zellen und das Wachstum dieser Zellen simuliert. Er wird den Zellzyklus der einzelnen Zellen berechnen und entsprechend notwendige Daten verarbeiten. Auch wird die Zellteilung wesentlich detaillierter, als sie in *D-VISION* dargestellt wird, im Simulationsmodell ablaufen. Die einzelnen Phasen des Zellzyklus insbesondere die M-Phase könnten entsprechend visualisiert werden. Dazu sind aber insbesondere für den Vorgang der Cytokinese detaillierte Daten notwendig, die stark von den vom Simulator verwendeten Modellen abhängig sind.

Ein Zelldifferenzierungssimulator verarbeitet eine große Menge von Informationen, die über die Modelle spezifiziert werden. Modelle der Zellen bzw. der Zellmechanik, Modelle der Reaktionsabläufe, genetische Netzwerke, Signal-Vermittlungs-Netzwerke usw., sind notwendig, um die Vorgänge in Zellen simulieren zu können, die die Zelldifferenzierung ausmachen. Die Informationen, die die Modelle enthalten, können nicht nur durch die Erstellung eines dreidimensionalen Bildes eines simulierten Zellhaufens visualisiert werden. Beispielsweise könnten Ablaufdiagramme der simulierten Reaktionen erstellt und angezeigt werden. Allerdings sollen mit *D-VISION* die Resultate der Simulation und nicht die zugrunde liegenden Modelle visualisiert werden. Zur Visualisierung von Modellen gibt es bereits einige Visualisierungstools²⁷, die natürlich auch zusätzlich zu *D-VISION* eingesetzt werden können.

²⁷ Beispielsweise werden auf der Site von SBML (<http://sbml.org>) unter anderem Visualisierungstools für die Darstellung von SBML-Modellen zur Verfügung gestellt.

Visualisierung

Durch eine zusätzliche Oberflächengestaltung der Zellen ist eine kombinierte Markierung von Zellen möglich. Die Oberflächen könnten durch Texturen, Muster oder 3D-Texturen wie z.B. Noppen verändert werden und so als Markierung von Zellen dienen. Zusätzlich können die Zellen auch noch eingefärbt werden. Somit wäre beispielsweise eine gleichzeitig Darstellung der Konzentration einer Substanz an der Oberfläche durch Farbe und eine zusätzliche Markierung mit einer Schraffur möglich.

Statt die Zellen, in denen ein bestimmtes Gen aktiv ist, durch die Einfärbung der Oberfläche zu markieren, kann die Aktivität eines Genes auch durch Platzieren eines Objektes im Inneren der Zelle realisiert werden. Durch verschieden geformte oder verschiedenfarbige Objekte kann die Aktivität mehrerer Gene gleichzeitig angezeigt werden. Allerdings werden bei einem großen Zellhaufen, der mit großem Abstand betrachtet wird, diese Objekte im angezeigten Bild recht klein ausfallen und nur schwer zu unterscheiden sein.

Eine nützliche Erweiterung der Selektion von Zellen ist die Gruppierung von bestimmten Zellen. Sind einige Zellen markiert bzw. selektiert könnten sie zu einer Gruppe zusammengefasst werden, die im Anschluss über eine Gruppenbezeichnung direkt selektiert werden kann.

Wenn Zellen unsichtbar gemacht werden, so werden bei der Konzentrationsanzeige nicht die in ihnen vorhandenen TC angezeigt. Es könnte aber auch das Konzentrationsgefälle ohne die Zellen angezeigt werden. Somit wäre nur die Substanzverteilung im Raum ohne Zellzusammenhang zu sehen.

Genauso wie die Konzentration der Substanzen in den Zellen mit kleinen Objekten in ihnen angezeigt werden, könnte auch die Konzentration der Substanzen an der Oberfläche der Zellen mit kleinen Objekten statt durch Einfärbung der Hülle der Zellen realisiert werden. Die Objekte müssten auf der Hülle platziert werden. Durch verschiedenfarbige Objekte könnte somit auch mehr als eine Substanz an der Zelloberfläche visualisiert werden. Problematisch dabei könnte das Auseinanderhalten der inneren und äußeren Objekte sein.

Statt die Objekte für verschiedene Substanzen bei der Konzentrationsanzeige, nur durch verschiedene Farben zu unterscheiden, könnten auch unterschiedliche Formen Verwendung finden. Da die kleinen Objekte unter Umständen so klein sind, dass die Form nicht unterschieden werden kann, ist es sinnvoll die Form lediglich als Zusatz zur Verschiedenfarbigkeit zu verwenden.

In der zweidimensionalen Anzeige wird das Konzentrationsgefälle durch Einfärbung des Zellinneren realisiert. Stattdessen könnten auch in der zweidimensionalen Darstellung kleine Objekte Verwendung finden. Dies würde auch die Darstellung mehrerer Substanzen gleichzeitig zulassen. Problematisch ist

dabei, dass beim Schnitt durch den Zellhaufen eine große Zelle unter Umständen nur angeschnitten wird. Im Schnittbild wird eine solche Zelle entsprechend durch einen kleinen Kreis dargestellt, in dem allerdings die gesamte Substanzmenge repräsentiert werden muss, um die Konzentrationen im Vergleich zu den anderen Zellen richtig darzustellen.

Bei der Definition der Zellklassen werden für die Argumente mit den Substanzen nur Vergleiche mit einem Zahlenwert zugelassen. Das könnte durch Vergleiche zwischen Substanzen erweitert werden, die Ausdrücke wie „mehr von Substanz A als von Substanz B“ oder „mehr von Substanz außen an der Zelle als innen“ zulassen würde. Weiterhin könnten auch schon existierende Regeln als Argument für das Erstellen neuer Regeln dienen.

Eine Idee für eine Erweiterung der Navigation ist eine automatische Kamerafahrt, die den Zellhaufen auf Anfrage so bewegt, dass eine selektierte Zelle im Mittelpunkt des Bildschirms platziert wird. Somit könnten auch in großen Zellhaufen bestimmte, aus einer Liste ausgewählte Zellen in der dreidimensionalen Darstellung schnell und einfach aufgefunden werden.

Um Verbindungen oder Verwandtschaften von Zellen darzustellen, könnten die Mittelpunkte von zwei Zellen mit einer Linie bzw. einem schmalen Zylinder verbunden werden.

Noch eine Erweiterung des Visualisierungssystems wäre die Möglichkeit, Animationen als Movie-Clips abzuspeichern. Beispielsweise könnte die zeitliche Entwicklung des Zellhaufens oder ein Zustand zu einem Zeitpunkt mit Kamerafahrten und sich ändernden Darstellungsparametern als Animation festgehalten werden. Solche Filme könnten als Anschauungsmaterial dienen.

5 Implementierung

Das Kapitel fünf enthält Beschreibungen der umgesetzten Konzepte zu den wichtigsten Teilen des Prototypen. Zunächst wird die Entscheidung für die eingesetzte Software-Grundlage erläutert, danach die verwendete Datenstruktur beschrieben. Im Anschluss werden Erklärungen zu interessanten Aspekten der Programmierung gegeben. Dazu gehört die visuelle Umsetzung der Zellen, der Substanzen und des zweidimensionalen Schnittbildes.

5.1 Eingesetzte Technologien

Die Zielplattform für die prototypische Implementierung des entwickelten Konzeptes sind Java und Java3D.

Java ist eine Objektorientierte Programmiersprache, die oft als Plattformübergreifend bezeichnet wird. Java-Programme werden durch Kompilierung zu Bytecode und anschließender Interpretation durch eine Virtuelle Maschine, die auf vielen Rechnerplattformen verfügbar ist, ausgeführt.

Es sind sehr viele Programmbibliotheken für Java verfügbar, die einen universellen Einsatz der Programmiersprache ermöglichen.

Die Java3D-API von Sun [SUN03] ist eine Programmierschnittstelle, die eine 3D Grafikprogrammierung auf hohem Abstraktionsniveau ermöglicht. Sie wird als 3D-Grafik-API der 4. Generation²⁸ bezeichnet, die auf dem Szenegraph-Konzept basiert. Der Szenegraph ist eine Baumstruktur, die alle in der Szene dargestellten Objekte (wie 3D-Objekte, Transformationskomponenten und Licht) verwaltet und eine Manipulation dieser Objekte zulässt. Das Rendering ist für den Programmierer transparent, ohne jedoch die Flexibilität zu beeinträchtigen [MOH98]. Der objektorientierte Ansatz von Java wird auch in Java3D verfolgt.

Ein Vorteil von Java / Java3D ist die Verfügbarkeit auf unterschiedlichen Plattformen. Die Laufzeitumgebungen müssen auf dem Rechner, auf dem *D-VISION* laufen soll, installiert werden.

Als Datenbank Management System (DBMS) wird die McKoi SQL Database [DOW03] eingesetzt. Es handelt sich um eine relationale Datenbank, die komplett in Java geschrieben ist und deswegen einige Vorteile bietet. Allem voran bedarf es keiner Installation des DBMS. Es sind nur die Programm-Bibliotheken

²⁸ Nach den vorangegangenen 3D-API Generationen Siggraph CORE, PHIGS und OPEN GL wurden die 3D-APIs basierend auf der Szenegraph-Struktur entwickelt.

notwendig, die in den Classpath eingebunden werden müssen, um zur Laufzeit von *D-VISION* zur Verfügung zu stehen. Die Daten werden in proprietären Dateien im Dateisystem des Rechners abgelegt. Der Zugriff auf die Daten wird über die in Java vordefinierte Datenbank-Zugriffs-API JDBC realisiert. Das DBMS arbeitet mit Standard SQL, wobei nicht alle in Standard SQL enthaltenen Features unterstützt werden.

Die für *D-VISION* verwendete Datenbank kann prinzipiell ohne großen Aufwand ausgetauscht werden, wenn ein JDBC-Treiber für das einzusetzende DBMS vorhanden ist und die benötigten Tabellen angelegt worden sind. Im Javaprogramm müssen nur der veränderte Treiber und die Zugangsdaten verändert werden.

Für die Erstellung der Tabellen und sonstige Manipulationen der Daten bzw. Datenbank wurde während der Entwicklungsphase DbVisualizer von Minq Software²⁹ eingesetzt. Es handelt sich um ein in Java programmiertes Datenbank-Tool, mit dem relationale Datenbanken bearbeitet werden können.

5.2 Beschreibung der Datenstruktur

In diesem Abschnitt wird die tatsächliche Datenstruktur, wie sie in der Datenbank realisiert wurde, vorgestellt. Weiterhin wird das Dateiformat für den Import von Daten präsentiert.

Tatsächliche Datenstruktur

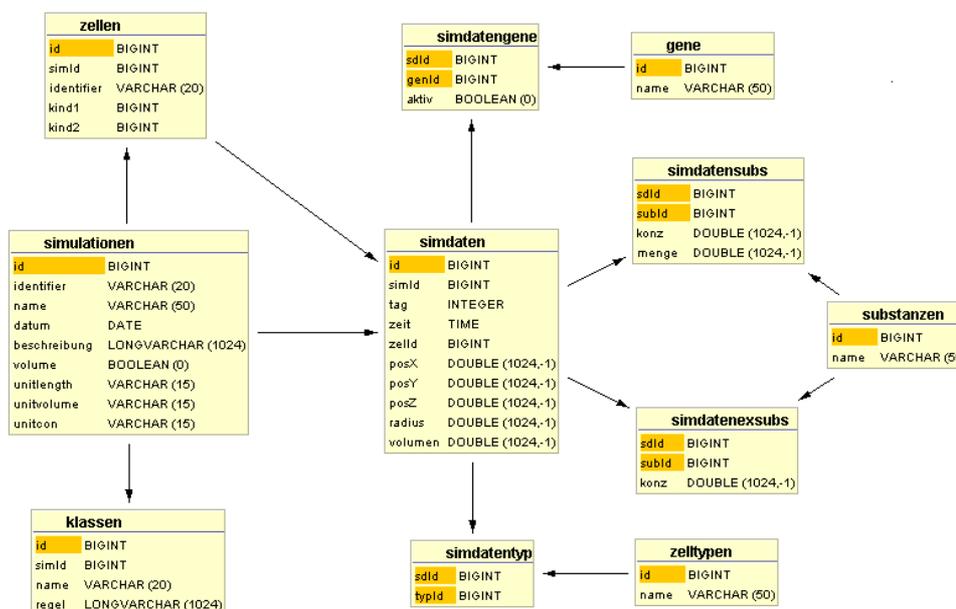


Abb. 5.1: Datenstruktur

Die Tabelle **simulationen** enthält die verschiedenen in der Datenbank gespeicherten Simulationen mit den zugehörigen Daten. Neben Name, eindeutigem Kürzel, Datum und Beschreibung sind auch Informationen für die Visualisierung enthalten, wie Maßeinheiten die zur Anzeige der Daten dienen und ob die Größe der Zellen vom Simulator in Form von Radius oder Volumen geliefert wird.

Die Tabellen **substanzen**, **gene**, **zelltypen** und **zellen** enthalten nur die Namen dieser Objekte. Die entsprechenden Daten werden über die Relationen abgebildet. Eine Ausnahme bildet die Tabelle **zellen**, denn dort werden zusätzlich die Verwandtschaften der Zellen hinterlegt. Weiterhin enthält sie eine Referenz auf die entsprechende Simulation.

Das Kernstück der Datenbank ist die Tabelle **simdaten**. In ihr werden die eigentlichen Daten einer Simulation abgelegt. Jeder Datensatz repräsentiert eine Zelle und ihren Zustand zu einem bestimmten Zeitpunkt der Simulation. Er enthält den Zeitpunkt in Form von Tag und Zeit³⁰, die Position der Zelle durch die drei Raumkoordinaten und die Größe der Zelle. Wobei die Größe sowohl durch den Radius als auch das Volumen gespeichert wird³¹.

Die Relation zur Tabelle **simulationen** bildet die Zugehörigkeit der Datensätze zu einer Simulation ab. Die m:n Relationen über die Zwischentabellen **simdatensubs**, **simdatenexsubs**, **simdatengene** und **simdatentyp** legen den Zustand der Zelle fest. Dabei werden in den Substanzen-Zwischentabellen, die Konzentration der Substanz in und außerhalb der Zelle gespeichert. Die Zwischentabelle **simdatensubs** enthält zusätzlich auch die Stoffmenge, die benötigt wird um die Skalierungen der Tiny Cubes vorzunehmen. Der Wert wird beim Einlesen der Daten durch den Aufbereiter zusammen mit dem Volumen der Zelle aus der Konzentration berechnet. Die Relation zu den Genen enthält das Attribut, ob das Gen aktiv bzw. inaktiv ist. Die Relation **simdatentyp** ermöglicht das Zuordnen von Zelltypen zu bestimmten Zellen.

In der Tabelle **klassen** werden die Regeln für die Klassifizierung der Zellen abgelegt. Eine Referenz auf eine Simulation kennzeichnet eine lokale Klassendefinition, ansonsten handelt es sich um eine globale Klassendefinition³².

²⁹ <http://www.minq.se/products/dbvis/>

³⁰ Die Simulation fängt am Tag 0 und zum Zeitpunkt 00:00:00 an. Tag und Zeit repräsentieren entsprechend den vergangenen Zeitraum seit Beginn der Simulation.

³¹ Die entsprechend nicht vorhandene Größe wird vom Aufbereiter beim Einlesen der Daten berechnet.

³² vgl. Kapitel 4.3.4

Es wurde ein SQL-Skript erzeugt, das die Tabellen und die Relationen anlegt. Damit lässt sich die Datenstruktur auch auf anderen relationalen Datenbanken erzeugen. Zusätzlich muss ein entsprechender Datenbanktreiber in die Klasse DBConnector eingebunden werden.

XML - Dateiformat

Die Schnittstelle des Aufbereiters, an der Daten entgegengenommen werden können, wurde durch das Einlesen von XML-Dateien realisiert. Gültige XML-Dateien werden von der Klasse *Preperator* eingelesen und die Daten in der Datenbank gespeichert.

In Anhang A ist die Dokument Definition Datei *dvision.dtd* aufgelistet. Dennoch soll hier kurz auf die Struktur der XML-Dateien eingegangen werden.

Das Wurzelement der XML-Datei ist das `<dvision>` Element. Es kann maximal ein `<simulation>` Element und beliebig viele `<simulationdata>` Elemente beinhalten.

Das `<simulation>` Element beinhaltet die Informationen über eine Simulation. Es kann beispielsweise so aussehen:

```
<simulation>
  <name>Test-Simulation 3</name>
  <identifier>TS03</identifier>
  <date>03.04.2004</date>
  <description>
    Dies ist eine TestSimulation. Die Daten wurden von Hand
    eingegeben.
  </description>
  <cellsize by="radius" />
  <units length="micrometer" volume="fl" concentration="mmol/pl"
/>
</simulation>
```

Das `<simulationdata>` Element enthält das Attribut `simulation`, mit dem angegeben wird, zu welcher Simulation die folgenden Daten gehören.

Das `<simulationdata>` Element kann beliebig viele `<cellheapsnapshot>` oder `<celllines>` Elemente enthalten.

In den `<cellheapsnapshot>` Elementen werden die Daten, die einen momentanen Zustand der Simulation beschreiben, gekapselt. Es enthält Zeitattribute mit denen der entsprechende Zeitpunkt innerhalb der Simulation festgehalten wird. Darüber hinaus enthält es eine Menge von `<cell>` Elementen, die jeweils eine Zelle mit ihrem momentanen Zustand beschreiben. Dies kann z.B. folgendermaßen aussehen:

```

<cell identifier="Aa">
  <position x="-2.0" y="2.0" z="0.0" />
  <radius value="4.0" />
  <type>blastula</type>
  <type>dorsal</type>
  <innersubstances>
    <substance name="Protein01" concentration="3.0" />
    <substance name="Protein02" concentration="2.0" />
  </innersubstances>
  <outersubstances>
    <substance name="Hormon01" concentration="12.0" />
  </outersubstances>
  <gens>
    <gen identifier="Gen01" aktive="true"/>
  </gens>
</cell>

```

Das Element `<celllines>` hat keinen Zeitbezug. Es gibt die Verwandtschaftsverhältnisse der existierenden Zellen an:

```

<celllines>
  <cellrelation identifier="egg" child1="A" child2="B" />
  <cellrelation identifier="A" child1="Aa" child2="Ab" />
  <cellrelation identifier="B" child1="Ba" child2="Bb" />
</celllines>

```

Die in der Simulation verwendeten Bezeichnungen für Substanzen, Gene oder Zelltypen müssen nicht separat aufgelistet werden. Der Aufbereiter legt beim Einlesen der XML-Datei die entsprechenden Objekte in der Datenbank ab.

Die Struktur der XML-Dateien bietet den Vorteil, dass die Daten auch auf mehrere Dokumente aufgeteilt werden können. So können z.B. die gesamten Daten einer Simulation in einer einzigen XML-Datei zusammengefasst werden. Ebenso gut kann eine XML-Datei auch nur Teile einer Simulation enthalten. Beispielsweise kann der Zustand einer einzigen Zelle zu einem beliebigen Zeitpunkt mit einer XML-Datei beschrieben werden. Der Aufbereiter liest die XML-Dateien einzeln ein und vervollständigt so die Gesamtdaten einer Simulation.

Somit ist es möglich, auch verteilte Simulatoren zu bedienen, die ihre Daten einzeln an den Aufbereiter schicken. In der Datenbank des Aufbereiters sammeln sich die Daten und ergeben die gesamten Simulationsdaten.

5.3 Beschreibung der Objektstruktur

In diesem Abschnitt wird eine kurze Übersicht über die Objekte bzw. Klassen und deren Einordnung in Packages gegeben.

com.pir.dvision.data

enthält die Basisklassen für die Daten, die aus der Datenbank gelesen werden und der Visualisierung dienen.

com.pir.dvision.prep

enthält die Klassen für den Aufbereiter. Die Klasse *DBConnector* baut die Verbindung zur Datenbank auf. Die Klasse *Preparator* stellt die Methoden zum Lesen der Daten aus der Datenbank für die Visualisierung und zur Speicherung der Daten für die XML-Schnittstelle zur Verfügung.

com.pir.dvision.vis

enthält nur die Klasse *DVision*, die die Visualisierungsanwendung startet.

com.pir.dvision.vis.gui

enthält sämtliche Klassen für die grafische Oberfläche der Anwendung.

Die Klasse *MainFrame* stellt das Hauptfenster der Anwendung dar. Die Klassen **Panel*, *ViewBar* und *StatusBar* sind für die jeweiligen Anzeigen der Daten und Benutzerkontrollen wie Listen und Buttons usw. zuständig. Die Klasse *Controller* stellt eine Schnittstelle zwischen den einzelnen Komponenten bereit und kapselt die für die anderen Komponenten notwendigen Methoden der Klasse *MainFrame*. Die Klassen *ActionController* und *CursorController* sind für die Ausführung der vom Benutzer ausgelösten Ereignisse verantwortlich. Die Klasse *Schnittbild* ist für die Berechnung und die Darstellung der Schnittbilder verantwortlich. Die übrigen Klassen sind Dialog-Fenster und selbst definierte Kontrollen.

com.pir.dvision.vis.j3d

Die Klassen im Unterpaket *j3d* sind für die Darstellung der dreidimensionalen Objekte zuständig. Weiterhin befindet sich im Package die Klasse *TCController*, die für die Berechnung und Positionierung der Tiny Cubes verantwortlich ist. Die Klasse *DeformationsKugel* ist die dreidimensionale Repräsentation einer Zelle. Sie berechnet die Kugelpunkte und Tiny Cubes Positionen innerhalb einer Kugel und führt die Deformation aus. Die Klasse *Schnittkreis* repräsentiert die zweidimensionale Darstellung einer Zelle.

5.4 Beschreibung ausgewählter Realisierungen von Konzepten

In diesem Abschnitt sollen einige, ausgesuchte Umsetzungen des Konzeptes vorgestellt werden. Es wurden dafür die Implementierungen bzw. die Umsetzungen von der konzeptionellen Idee bis zum fertigen Algorithmus ausgewählt, die komplex aufgebaut sind und bei der Implementierung aufwändig waren.

5.4.1 Deformation von Kugeln

Nachdem die Idee, die Zellen als deformierte Kugel darzustellen, geboren war, begann eine Suche nach bereits realisierten Lösungen. Die Suche führte unter anderem über Blobs³³, zur Erkenntnis, dass eine selbst entwickelte Lösung, die die geometrischen Objekte deformiert, weniger rechenintensiv und einfacher zu implementieren sein würde.

Mathematische Beschreibung

Nur zwei sich überschneidende Kugeln können sich auch gegenseitig deformieren.

Die beiden Kugeln K_1 und K_2 seien durch ihre Mittelpunkte M_1 und M_2 sowie ihre Radien r_1 und r_2 gegeben.

Die Kugeln schneiden sich wenn $r_1 + r_2 > |M_1M_2|$ ist.

Wenn sich die beiden Kugeln überschneiden, so gibt es einen Überschneidungsbereich B .

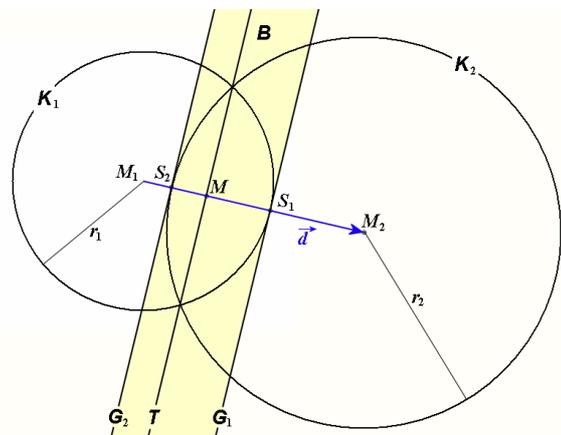


Abb. 5.2: sich überschneidende Kugeln

Mit dem Verbindungsvektor $\vec{d} = \overrightarrow{M_1M_2}$ kann der Abstand $d = |\vec{d}|$ zwischen den Kugelmittelpunkten berechnet werden.

³³ Blobs stellen eine Lösung für das Problem von sich vereinigenden Objekten dar. Um ihre Funktionalität für deformierte Objekte zu nutzen, wäre eine Veränderung der Dichtefunktion erforderlich gewesen.

Mit dem normierten Abstandsvektor \vec{d}_0 können die beiden Punkte S_1 und S_2 auf den Kugeloberflächen berechnet werden, deren Tangentialebenen G_1 und G_2 die Grenzen des Überschneidungsbereiches B definieren.

$$S_1 = M_1 + r_1 \cdot \vec{d}_0 \qquad S_2 = M_2 - r_2 \cdot \vec{d}_0$$

Da die Geometrie der Kugeln durch Punkte an ihrer Oberfläche beschrieben wird, wird die Deformation der Kugel realisiert, indem die Punkte der beiden sich überschneidenden Kugeln im Überschneidungsbereich entsprechend verschoben werden. Dazu ist eine Trennebene T , die parallel zu den Begrenzungsebenen des Überschneidungsbereiches ist, erforderlich. Ihre Lage wird durch den frei wählbaren Punkt M , der jedoch auf der Verbindungslinie $\overline{M_1M_2}$ zwischen S_1 und S_2 liegen muss, bestimmt. Der Punkt M kann mit der Geradengleichung $M = M_1 + \alpha \cdot \vec{d}_0$ und einem entsprechenden Skalierungsfaktor α berechnet werden. Die Lage der Trennebene bestimmt, wie weit die jeweiligen Punkte der Kugeln verschoben werden und damit wie stark die Kugeln deformiert werden.

Ein mögliche Lage für die Trennebene T wäre die Schnittebene der beiden Kugeln.

Der Punkt M kann über die Gleichung $\alpha = \frac{d^2 + r_1^2 - r_2^2}{2d}$, die durch Anwendung

des Satzes von Pythagoras hergeleitet werden kann, berechnet werden³⁴.

Eine kleine und eine große Kugel würden sich allerdings gegenseitig so deformieren, dass die kleine Kugel sehr viel stärker deformiert würde als die Große. Das Bild, das sich daraus ergibt, wirkt unnatürlich.

Um ein natürlicheres Aussehen der deformierten Kugeln zu erhalten, wurden andere Lagen für die Trennebene getestet. Das beste Ergebnis wurde unter Verwendung des Verhältnisses der Radien der beiden Kugeln gewonnen:

$$\alpha = \frac{r_1}{r_1 + r_2} \cdot d \qquad \Rightarrow \qquad M = M_1 + \frac{r_1}{r_1 + r_2} \cdot \vec{d}$$

Bei der daraus entstehenden Deformation wird die größere Kugel stärker deformiert als die kleinere. Dies suggeriert, dass kleine Kugeln eine größere innere Stabilität haben und deswegen bei gleicher Krafteinwirkung weniger stark deformiert werden. Allerdings ist die Einschätzung, dass ein solches Bild natürlicher aussieht, subjektiv. Beide alternativen Berechnungen wurden im Prototypen umgesetzt und können entsprechend eingestellt werden.

³⁴ vgl [BOU88]/geometry/spheresphere/

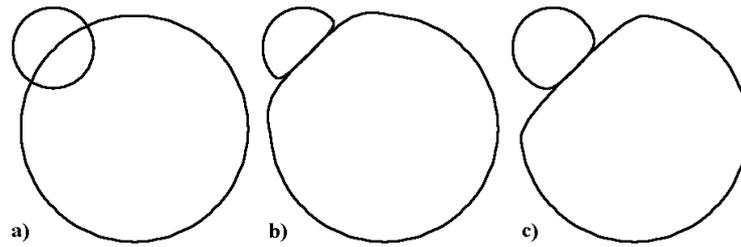


Abb. 5.3: verschiedene Lagen der Trennebene

Die eigentliche Deformation wird nun für eine Kugel beschrieben. Sie muss natürlich für beide ausgeführt werden. Für die Berechnungen der Deformation müssen die oben beschriebenen Punkte S_1 , S_2 und M vorliegen.

Sämtliche Punkte der Kugeloberfläche, die im Überschneidungsbereich B liegen, werden so verschoben, dass sie anschließend im Bereich N liegen. Dieser Bereich wird durch die Trennebene T und die in der Kugel liegende Grenzebene G_2 des Überschneidungsbereiches B definiert.

Ein Punkt P der Kugeloberfläche im Überschneidungsbereich B wird senkrecht zu

den Ebenen in Richtung des Vektors $\vec{d} = \overrightarrow{S_1M}$ verschoben, so dass das Verhältnis des Punktes P zu den Bereichsgrenzen von B in den Bereich N übertragen wird.

Mit dem Abstand³⁵ a des Punktes P zur äußeren Bereichsgrenze G_1 kann entschieden werden, ob ein Punkt P im Überschneidungsbereich B liegt. Ist a kleiner

als die Breite b des Bereiches B (entspricht dem Abstand der Punkte S_1 und S_2 : $b = |S_1S_2|$), so liegt P im Überschneidungsbereich und muss entsprechend verschoben werden.

Das Verhältnis der Lage des Punktes P zu den Bereichsgrenzen von B wird über den Abstand a und die Breite b des Bereiches B bestimmt. Mit der Breite n des Bereiches N wird die Lage des verschobenen Punktes P_n über den Abstand x , den er zur Ebene T haben muss, bestimmt.

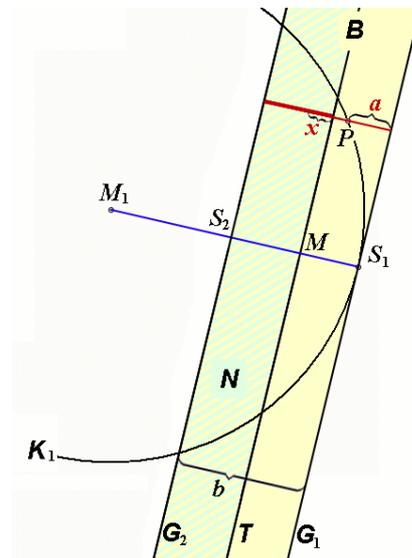


Abb. 5.4: Deformation durch Verschiebung der Punkte

³⁵ Zum Berechnen des Abstandes eines Punktes zu einer Ebene wird die Ebenengleichung benötigt. Sie ergibt sich im Fall von G_1 durch den Punkt S_1 und den Normalenvektor der Ebene $\vec{d} = \overrightarrow{S_1M}$.

$$(1) \quad a : b = x : n \quad \Rightarrow \quad x = \frac{a}{b} n$$

Dadurch ergibt sich die Lage des verschobenen Punktes mit:

$$P_n = P + \left(x + \left| \overline{S_1 M} \right| - a \right) \cdot \vec{d}_0 \quad \text{wobei } \vec{d}_0 \text{ der normierte Vektor } \vec{d} = \overline{S_1 M} \text{ ist.}$$

Durch Anwendung dieser Berechnungen auf alle Kugelpunkte wird die Kugel deformiert.

Implementierung in 2D

Um diese mathematischen Überlegungen zu testen, wurde eine prototypische Anwendung³⁶ implementiert, die die Berechnungen allerdings im Zweidimensionalen also mit Kreisen statt Kugeln und Geraden statt Ebenen ausführt. Bei der Betrachtung der entstehenden deformierten Kreise fiel auf, dass die Deformation eckig aussah.

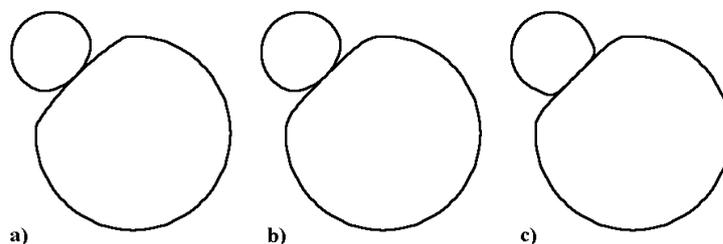


Abb. 5.5: lineare, quadratische und kubische Deformation

Eine Lösung für dieses unnatürliche Aussehen wurde darin gefunden, die Verschiebung der Punkte im Überschneidungsbereich zu verändern. Das Verhältnis der Lage des Punktes P zur Breite des Bereiches B wird durch den Term $\frac{a}{b}$ ausgedrückt. Eine Quadrierung dieses Terms hat zur Folge, dass Punkte, die nahe an der äußeren Bereichsgrenze G_1 liegen, nach der Verschiebung noch näher an der Trennebene T liegen. Dadurch sehen deformierte Kugeln weicher aus.

Statt zu quadrieren, kann das Verhältnis auch kubisch in die Berechnung einfließen, was eine noch stärkere Rundung der Deformation zur Folge hat.

Gleichung (1) muss dementsprechend modifiziert werden:

$$\text{quadratisch: } x = \left(\frac{a}{b}\right)^2 n \quad \text{kubisch: } x = \left(\frac{a}{b}\right)^3 n$$

Alle drei Deformationsstärken wurden als Parameter in die Implementierung aufgenommen, allerdings wurde die quadratische Deformation als Standard definiert, da sie am natürlichsten aussieht.

³⁶ In der Klasse *Deform2D* realisiert.

Implementierung in 3D

Die Java3D API enthält zwar geometrische Primitive insbesondere eine Kugel, allerdings lassen sich die Punkte, die die Oberfläche der Primitive bestimmen, nicht ohne weiteres verändern. Für die Implementierung wurde deswegen eine eigene Kugel programmiert. Dazu wurden die Punkte an der Oberfläche der Kugel berechnet und mit einem *TriangleArray* die Geometrie aufgebaut. Die Punkte werden nur einmalig statisch für eine Kugel mit einem Radius $r = 1$ berechnet. Sie dienen für die weiteren, zu erstellenden Kugeln als Grundlage und gehen durch Skalierung hervor. Außerdem dienen sie als Normalen für die jeweiligen Punkte der Oberfläche.

Die Koordinaten der Kugeloberflächenpunkte werden durch parametrisierte Gleichungen mit trigonometrischen Berechnungen über zwei Winkel, die die Parameter sind, berechnet³⁷. Die Anzahl der Segmente, in die diese Winkel die Kugel oder genauer 180° unterteilen, bestimmt, aus wie vielen Punkten bzw. Dreiecken die Kugeloberfläche aufgebaut ist. Es werden standardmäßig 36 Segmente (das entspricht einer Rasterung der Kugeloberfläche in 5° Schritten) verwendet. Daraus ergeben sich 2522 Punkte und 5040 Dreiecke³⁸, mit denen eine Kugel dargestellt wird. Durch die Beleuchtungsrechnung und die daraus resultierende Schattierung der Oberfläche ergibt sich eine gute optische Repräsentation einer Kugel. Damit das Programm auch auf Rechnern mit vergleichsweise geringer Leistung lauffähig ist, kann die Anzahl der Segmente eingestellt werden.

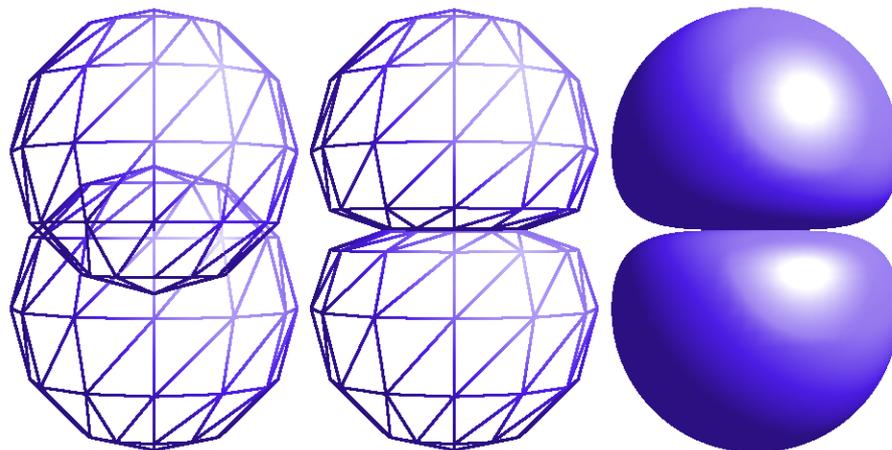


Abb. 5.6: Veranschaulichung der Kugeldeformation

³⁷ vgl. [BOU88]/geometry/sphere/

³⁸ Bei der verwendeten Implementierung ergeben n Segmente $(2n^2 - 2n + 2)$ Punkte und $(4n^2 - 4n)$ Dreiecke.

Mit den selbst erzeugten Kugeln wurde die Deformation implementiert. Die Methode *intersects()* eines Objektes der Klasse *DeformationsKugel* überprüft, ob sich diese mit einer übergebenen Instanz einer anderen Kugel überschneidet. Ist dies der Fall, so wird die Kugel über den Aufruf der Methode *deform()* deformiert. Diese Prozedur muss auch mit der anderen Kugel durchgeführt werden. Die gesamten Berechnungen zur Deformation der Kugeln werden vor dem Anzeigen, d.h. vor dem Einhängen in den Szenegraphen, ausgeführt.

Volumenbetrachtung

In Kapitel 4.3.1.a) wurde bei der Beschreibung der Geometrie für die Darstellung der Zellen auf das Problem der Volumenreduktion bei der Deformation von Kugeln eingegangen.

Mit der in Anhang A hergeleiteten Formel kann mathematisch das Volumen für ein deformiertes Kugelsegment berechnet werden. Durch Subtraktion des nicht deformierten Kugelsegmentes und anschließender Addition des deformierten Kugelsegmentes zum Volumen der Kugel, lässt sich das Volumen einer deformierten Kugel mathematisch berechnen. Es ist sogar möglich, das Volumen einer mehrfach deformierten Kugel zu berechnen, wenn sich die Deformationsbereiche nicht überlappen. Wenn sich die Deformationen einer Kugel allerdings überschneiden, so ist keine exakte mathematische Berechnung des Volumens mehr möglich.

Abgesehen von der direkten mathematischen Berechnung über Formeln, ist auch eine numerische Berechnung des Volumens über die geometrischen Angaben der Kugeln möglich. Die Oberfläche der Kugeln wird durch Dreiecke beschrieben, die zusammen mit dem Mittelpunkt der Kugel Pyramiden mit dreieckiger Grundfläche bilden. Die Summe über das Volumen³⁹ all dieser Pyramiden ergibt annähernd das Volumen für die Kugel. Dies gilt ebenso für deformierten Kugeln, da bei der Deformation ja die Oberflächenpunkte verschoben werden, die die Dreiecke formen. Je mehr Punkte eine Kugel beschreiben, desto genauer wird das auf diese Weise berechnete Volumen.

Wie schon beschrieben, stellt die Kompensation der Volumenverringerung ein Optimierungsproblem dar, das mit mehreren Iterationen von Vergrößerung der nicht-deformierten Kugel, anschließender Deformation und Bewertung des resultierenden Volumens gelöst werden kann. Abgesehen vom Vergleich mit dem

³⁹ Mit den drei Punkten des Dreieckes lässt sich über das Kreuzprodukt der Verbindungsvektoren die Fläche des Dreieckes berechnen. Weiterhin kann mit den drei Punkten eine Ebene beschrieben werden, deren Abstand zum Mittelpunkt der Kugel der Höhe der Pyramide entspricht. Über die Formel $\text{Volumen} = 1/3 * \text{Grundseite} * \text{Höhe}$ lässt sich schließlich das Volumen der Pyramiden berechnen.

Sollvolumen kann mit dem tatsächlichen Volumen einer deformierten Kugel berechnet werden, um wie viel diese Kugel durch die Deformation verkleinert wurde. Dies kann als Maßstab für die Vergrößerung in der nächsten Iteration dienen.

Zwar wurde die numerische Volumenberechnung implementiert, es wurde allerdings in *D-VISION* davon abgesehen, einen Algorithmus für das beschriebene Optimierungsproblem einzubauen, da dies mit einigem Rechenaufwand verbunden ist. Für jedes Standbild eines Zellhaufens, das angezeigt wird, müsste die Optimierung neu durchgeführt werden. Um die Volumenreduktion einer Kugel zu kompensieren, sind mehrere Iterationen von Deformationen notwendig. Für die Deformation einer einzigen Kugel⁴⁰ fallen pro infrage kommenden Deformationspartner 2522 Abstandsberechnungen für die Auswahl, welche Punkte verschoben werden sollen, an. Weiterhin werden pro Kugel 5040 Pyramidenvolumina berechnet.

Wenn eine mathematische Berechnung des Volumens einer deformierten Kugel möglich wäre, so würde sich der Rechenaufwand wesentlich verringern. Beim Ausführen einer Iteration müssten die Kugeln nicht tatsächlich deformiert werden, denn dies muss nur gemacht werden, um anschließend das Volumen berechnen zu können.

Neben dem Nachteil des hohen Rechenaufwandes ist der einzige Vorteil die genauere Darstellung der Kugeln. D.h. die deformierten Kugeln werden mit dem richtigen Volumen dargestellt, vorausgesetzt es wurde eine Lösung für die Optimierung jeder einzelnen Zelle gefunden. Allerdings wird auch eine deformierte Kugel mit kompensiertem Volumenverlust nur annähernd das exakte Volumen besitzen. Neben der ungenauen, numerischen Berechnung des Volumens wird auch das Optimierungsverfahren eine unscharfe Abbruchbedingung haben und somit eine Kugel als kompensiert ansehen, selbst wenn das Volumen nicht exakt stimmt. Das Ausführen der Kompensation der Volumenreduktion ist wegen des hohen Rechenaufwandes nur dann gerechtfertigt, wenn das ansonsten dargestellte Bild so verfälscht würde, dass es zu Fehlinterpretationen durch den Betrachter käme.

Im Fall der Darstellung eines Zellhaufens zur Analyse der Zelldifferenzierung, ist die exakte geometrische Repräsentation nicht wirklich entscheidend. Ziel ist es ja nicht, die Größe der Zellen visuell gegeneinander abzuschätzen, sondern den Zelltyp im geometrischen Kontext zu sehen. Außerdem ist davon auszugehen, dass in einem Zellhaufen alle Kugeln, außer denjenigen, die am äußeren Rand liegen, ungefähr gleichmäßig deformiert werden. Weiterhin wird nach Selektion

⁴⁰ Es wird von Kugeln mit den oben beschriebenen 36 Segmenten ausgegangen.

einer Zelle unter anderem auch die exakte vom Simulator gelieferte Größe separat angezeigt.

In *D-VISION* sind die abgebildeten, deformierten Zellen demnach als geometrisch richtig angeordnete, visuelle Referenz und nicht als exakte und im Bild nachmessbare Wiedergabe der Informationen des Simulators zu verstehen.

5.4.2 Konzentration mit Tiny Cubes

Um die in einer Zelle enthaltenen Substanzen zu visualisieren, wurde eine Methode verwendet, die sich von der Tiny Cubes Methode [NIE90] ableitet. Allerdings werden die darzustellenden Werte nicht durch den Farbewert, sondern allein durch die Anzahl der Tiny Cubes ausgedrückt. Außerdem wird ihre Position nicht in ein einheitliches Raster eingeordnet, sondern abhängig von der Zelle, zu der sie gehören, positioniert. Es wird entsprechend der in einer Zelle enthaltenen Stoffmenge eine bestimmte Anzahl an kleinen Würfeln positioniert. Die Anzahl der in einer Zelle enthaltenen Objekte repräsentiert visuell die Konzentration einer Substanz. Damit alle Bilder einer Simulation vergleichbar sind, müssen alle Würfel dieselbe Größe haben.

Da die Zellen eines Zellhaufens sehr unterschiedlich groß sein können, ist es erforderlich, eine einheitliche Größe für die Würfel zu wählen, die auch in der kleinsten Zelle Platz findet. Um dies zu erreichen, wird die kleinste Zelle innerhalb der gesamten, visualisierten Simulation gesucht. Ihr Radius wird auf eine Längeneinheit des virtuellen Raums skaliert und gibt die Größe für alle Tiny Cubes in allen Zellen vor. Die Kantenlänge eines Würfels beträgt dann 0,05 Längeneinheiten.

Die Positionen der einzelnen Würfel werden durch konzentrische Kugelschalen bestimmt. Sie werden bei der Deformation der Zellen mit deformiert. Jede Zelle erhält fünf ineinander liegende Schalen. Sie sind in ihrer Struktur genau wie die Zellhülle aus Punkten aufgebaut. Dabei enthält die innerste Schale nur zwei Segmente, das entspricht sechs Punkten. Jede weiter außen liegende Schale besteht aus einem Segment mehr. Die fünfte Schale hat schließlich sechs Segmente und 62 Punkte. Die insgesamt 150 Punkte entsprechen den Positionen für die Tiny Cubes.

In der praktischen Ansicht erwies sich diese Darstellung als problematisch, da die Würfel in einem Raster angeordnet waren und aus verschiedenen Blickwinkeln wie auf einer Perlenschnur aufgereiht erschienen. Um das Phänomen zu beseitigen und gleichzeitig die gleichmäßige Verteilung der Würfel innerhalb der Kugel zu erhalten, wurden die Positionen der einzelnen Würfel zufallsgesteuert leicht verändert. Die Abweichung von der ursprünglichen Position beträgt dabei maximal 45% des Abstandes zu den benachbarten Würfelpositionen.

Die Anzeige von Konzentrationen ergibt sich optisch durch die Visualisierung der enthaltenen Stoffmenge einer Zelle und das Volumen, in dem diese Stoffmenge verteilt ist. Die maximal anzeigbare Menge von Tiny Cubes in einer Zelle beträgt 150. Daraus ergibt sich, dass die während der gesamten Simulation maximal in einer Zelle enthaltene Stoffmenge auf diese 150 Einheiten skaliert werden muss. Entsprechend werden die Stoffmengen in allen Zellen mit demselben Faktor skaliert. Schließlich wird die errechnete Anzahl von Würfeln zufällig auf die verfügbaren Positionen in der entsprechenden Zelle abgebildet.

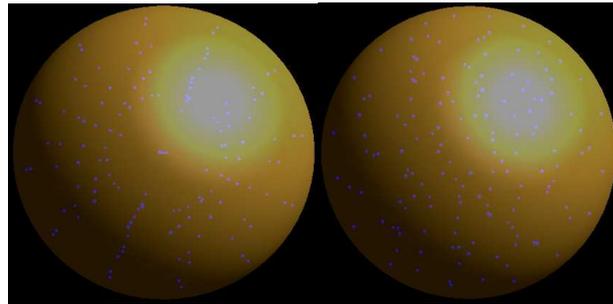


Abb. 5.7: Tiny Cubes Verteilung

Die Verwaltung der Tiny Cubes Positionen wird von der Klasse *Zelle* (bzw. von der von ihr gekapselten Klasse *DeformationsKugel*) übernommen. Sie gibt die Information über die Stoffmenge der anzuzeigenden Substanz an die Klasse *TCController* weiter. Diese nimmt die Skalierung vor und fordert von den Zellen die zugehörigen Tiny Cubes Positionen über die Methode *getNextFreeTinyCubePosition()* an. Anschließend stellt er die Würfel dar. Damit die Zellen keine Positionen doppelt vergeben, werden belegte Positionen als solche markiert. Die Methode *resetTinyCubesPositions()* löscht alle vorhandenen Markierungen für belegte Positionen, um eine Neuverteilung der Tiny Cubes zu ermöglichen.

Wenn mehr als nur eine Substanz in der Zelle angezeigt werden soll, müssen die 150 Positionen auf die geforderten Substanzen aufgeteilt werden. Daher gilt:

beim Anzeigen von einer Substanz: 150 verfügbare Positionen pro Substanz

beim Anzeigen von zwei Substanzen: 75 verfügbare Positionen pro Substanz

beim Anzeigen von drei Substanzen: 50 verfügbare Positionen pro Substanz.

Jede der drei angezeigten Substanzen wird zur Unterscheidung in einer anderen Farbe dargestellt.

Um den Rechenaufwand während der Visualisierung gering zu halten, werden die erforderlichen Tiny Cubes alle beim Start des Programms unsichtbar kreiert⁴¹.

⁴¹ die maximal erforderliche Anzahl beträgt: 150 x Anzahl der maximal gleichzeitig visualisierten Zellen.

Sobald sie angezeigt werden sollen, werden sie nur noch an die ihnen zugeteilte Position verschoben, eingefärbt und sichtbar gemacht.

Die Tiny Cubes bleiben erhalten, solange das Programm aktiv ist. So müssen sie beim Einlesen einer neuen Simulation nicht neu erstellt werden⁴².

5.4.3 2D-Darstellung

Ein wesentlicher Bestandteil der gestellten Aufgabe ist die Erzeugung eines zweidimensionalen Schnittbildes durch den dreidimensionalen Zellhaufen. Es soll die Möglichkeit bieten, die Simulationsergebnisse direkt mit lichtmikroskopischen Aufnahmen zu vergleichen.

Das Schnittbild verläuft dabei senkrecht zur Blickrichtung durch den Zellhaufen und ist gleichzeitig nach vorne und hinten verschiebbar. Um den Zusammenhang innerhalb des Zellhaufens nicht aus den Augen zu verlieren, bleibt eine verkleinerte Darstellung des dreidimensionalen Bildes in einer separaten Anzeige erhalten. Dort wird die Lage der Schnittebene angezeigt. Die Schnittebene wird durch einen Punkt und ihren Normalenvektor, der dem Blickrichtungsvektor entspricht, definiert. Der Punkt bestimmt die Entfernung der Schnittebene vom Betrachter. Er liegt am Anfang im Ursprung und kann in Blickrichtung verschoben werden, um durch alle Schichten des Zellhaufens zu navigieren.

Die Berechnung des Schnittbildes ist die eigentliche Schwierigkeit. Java3D unterstützt zwar eine clipping-Funktion, das erzeugte Bild ist dabei aber immer noch eine 3D-Darstellung. Um ein echtes 2D-Bild zu erhalten, muss der Schnitt durch den Kugelhaufen daher selbst implementiert werden.

Die Grundlage dafür ist die Umrechnung von dreidimensionalen Koordinaten im virtuellen Raum in die zweidimensionalen Koordinaten auf dem Bildschirm. Dies kann in Java3D mit dem Canvas3D-Objekt, das für das Rendern verantwortlich ist, erreicht werden. Dieses Objekt bietet dafür zwei Methoden an. Die Methode *getVworldToImagePlate()* gibt ein Transformationsobjekt zurück, mit dem man die Koordinaten der virtuellen Welt in Koordinaten der ImagePlate umwandeln kann. Die Methode *getPixelLocationFromImagePlate()* wandelt schließlich Koordinaten vom ImagePlate in Bildschirmkoordinaten um. Es ist darauf zu achten, dass die Punkte, die umgewandelt werden sollen, von ihrem lokalen Koordinatensystem in das Wurzelkoordinatensystem der virtuellen Welt umgerechnet werden. Dies kann durch die Methode *getLokalToVworld()* der Node-Objekte erreicht werden.

Für das zweidimensionale Schnittbild müssen also nur die Punkte der Kugeloberflächen in der 3D-Darstellung gefunden werden, die jeweils gleichzeitig

⁴² Es werden lediglich die bei größerer Zellenanzahl zusätzlich erforderlichen Würfel neu erstellt.

auf der Schnittebene liegen. Die Klasse *Schnittbild* lokalisiert alle Zellen, die Kontakt zur Schnittebene haben. Sie ist ein Canvas-Objekt, das die Schnittkreise der Zellen mit der Schnittebene berechnet. Die Kugeloberflächen werden durch Dreiecke repräsentiert. Die Schnittlinie eines Dreiecks mit der Schnittebene enthält die Punkte des Schnittkreises.

Bezüglich der Lage des Dreiecks in Relation zur Ebene müssen sechs verschiedene Fälle unterschieden werden.

1. Fall: Alle drei Punkte des Dreiecks liegen auf derselben Seite der Ebene.

Das Dreieck hat keinen Kontakt zur Ebene und muss deshalb nicht in das Schnittbild übernommen werden.

2. Fall: Zwei Punkte des Dreiecks liegen auf derselben Seite, der dritte Punkt auf der anderen

Die beiden Dreieckskanten, die an dem Punkt, der alleine auf einer Seite der Ebene liegt, entspringen, schneiden die Ebene. Mit ihnen können die Schnittpunkte mit der Ebene berechnet werden. Diese Punkte definieren die Linie, die im Schnittbild angezeigt werden muss.

3. Fall: Einer der Punkte liegt auf der Ebene und die beiden anderen Punkte liegen auf derselben Seite.

In diesem Fall ist der Punkt, der direkt auf der Ebene liegt der einzige Schnittpunkt für das Schnittbild.

4. Fall: Ein Punkt liegt auf der Schnittebene und die beiden anderen liegen auf gegenüberliegenden Seiten der Ebene.

Die gesuchte Schnittlinie ergibt sich durch den Dreieckspunkt auf der Ebene und den Schnittpunkt der Ebene mit der Kante zwischen den beiden anderen Punkten.

5. Fall: Wenn zwei der Dreieckspunkte auf der Ebene liegen, sind diese beiden Punkte die Endpunkte der Schnittlinie.

6. Fall: Wenn alle drei Punkte des Dreiecks auf der Ebene liegen, bedeutet das, dass die zugehörige Kugel die Ebene an dieser Stelle tangiert. Theoretisch müsste das gesamte Dreieck im Schnittbild angezeigt werden, allerdings wäre der Informationsgehalt dieser Anzeige zu gering, als dass sie sinnvoll wäre. Daher wird für diesen Fall keiner der Punkte in das Schnittbild übernommen.

Durch die Unterscheidung der sechs Fälle und durch Vektorrechnung lassen sich zu allen die Ebene schneidenden Dreiecke zugehörige Linien finden. Durch die oben beschriebene Transformation werden sie in Koordinaten für das Schnittbild umgerechnet.

Zwar lässt sich mit diesen zusammenhangslosen Linien ein zweidimensionaler Schnittkreis zeichnen, er kann aber nicht mit Farbe gefüllt werden. Dafür muss ein zusammenhängendes Polygon verwendet werden. Um dieses Problem zu lösen, muss aus den einzelnen Punkten eines deformierten Kreises ein geordnetes Polygon erzeugt werden. Diese Aufgabe wird von der Klasse *Schnittkreis* übernommen. Sie ist ein Attribut der Klasse *Zelle*.

Die Methode *calculate()* berechnet aus den einzelnen Linien ein zusammenhängendes Polygon. Dafür werden nur die Endpunkte der Schnittlinien verwendet, und es wird darauf geachtet, dass alle Punkte nur einmal im Polygon enthalten sind.⁴³ Um das Polygon in geordneter Folge zu erstellen, wird mit einem der Punkte begonnen und der jeweils nächstgelegene als Folgepunkt in das Polygon übertragen. Diese Vorgehensweise ist nur deshalb möglich, weil die Polygone konvex sind.

Die Polygone können über die Methode *getPolygon()* abgerufen werden. Sie werden nun durch das *Schnittbild*-Objekt gezeichnet.

Die Hüllen der 2D-Zellen werden durch die Verwendung eines breiten Strichs deutlicher sichtbar und können wie die Zellinnenfläche eingefärbt werden.

5.5 Beschreibung der GUI

Das Anwendungsfenster ist in mehrere Bereiche unterteilt. Den Hauptteil des Fensters nimmt die Anzeige für die Zellen in Anspruch. Im unteren Teil ist ein Detailbereich für die Darstellung von Informationen und für die Interaktionsmöglichkeiten des Benutzers untergebracht. Auf der linken Seite ist eine Buttonleiste, mit der die verschiedenen Navigations- und Ansichtsmodi eingestellt werden können. Im Menü können weitere Funktionen der Anwendung, wie z.B. das Laden der Daten einer Simulation oder das Beenden der Anwendung, aufgerufen werden.

Über die Buttonleiste können vier Modi eingestellt werden. Die ersten beiden sind für die Navigation im 3D-Bild verantwortlich. Im Rotationsmodus kann das virtuelle, dreidimensionale Objekt um seinen Mittelpunkt gedreht werden. Im Translationsmodus kann das Objekt senkrecht zur Blickrichtung verschoben werden. In beiden Modi wird dies jeweils über die linke Maustaste erreicht. Mit der rechten Maustaste oder dem Mousrad kann das Objekt gezoomt werden. Der dritte Modus aktiviert die 2D-Ansicht des Zellhaufens. Hier kann über die Maus

⁴³ Theoretisch tauchen alle Schnittpunkte einer Kugeloberfläche mit der Ebene zweimal in der Berechnung auf, da sie jeweils in zwei Schnittlinien enthalten sind. Ausnahmen sind Punkte, die direkt auf der Ebene liegen, sie können mehr als zweimal in der Berechnung erscheinen. (Fall 4)

(linke Taste und Mausrad) die Lage der Schnittebene verändert werden. Der vierte Modus ist der Selektionsmodus. Im Selektionsmodus ändert sich das Verhalten der Maus. Statt zu drehen oder zu verschieben können in der zwei- und in der dreidimensionalen Anzeige Zellen direkt durch Anklicken ausgewählt werden.

Der Detailbereich ist in unterschiedliche Panels unterteilt, die über Reiter erreichbar sind.

Im *Zellen* Panel befindet sich eine Liste, in der alle Zellen des aktuell angezeigten Zeitpunktes der Simulation aufgelistet sind. Über diese Liste können gezielt Zellen selektiert werden. Mit Buttons können alle Zellen gleichzeitig selektiert oder deselektiert werden. Die aktuell ausgewählten Zellen können über entsprechende Buttons transparent oder unsichtbar gemacht werden.

Das *Zellinhalt* Panel besteht aus drei Listen. Wenn eine Zelle selektiert ist, werden in den Listen die Substanzen, die sich in der Zelle befinden, die Substanzen an der Oberfläche der Zelle, sowie die Gene der Zelle aufgelistet. Mit den Substanzen wird auch deren Konzentration und mit den Genen deren Aktivität in den Listen angezeigt. Weiterhin werden allgemeine Informationen über die Zelle wie die Größe im Panel angegeben.

Die Anzeige der Konzentrationsverteilungen wird mit dem Panel *Sub. & Gene* gesteuert. Es wird eine Liste mit allen Substanzen, die in der Simulation vorkommen, angezeigt. Aus ihr können die Substanzen ausgewählt werden, deren Konzentrationsgefälle im 2D- oder 3D-Bild erscheinen sollen. Die Anzeige der Konzentrationsverläufe wird mithilfe eines Steuerelementes geregelt. In der dreidimensionalen Ansicht können bis zu drei Substanzen im Inneren der Zellen und eine Substanz außen an der Zelle dargestellt werden. Im zweidimensionalen Bild ist es jeweils nur eine Substanz. Weiterhin können mit dem Panel *Sub. & Gene* Zellen markiert werden, die bezüglich ihres Zellzustandes bestimmte Kriterien erfüllen. Dazu ist ein zweites Steuerelement vorhanden, mit dem die gewünschten Kriterien eingestellt werden können. Um auch Gene auswählen zu können, ist eine Liste mit allen Genen vorhanden.

Mit dem Panel *Zustand* können Markierungen von Zellen bezüglich ihres Zustandes vorgenommen werden. Dazu ist eine Liste mit den vom Simulator vergebenen Zelltypen vorhanden. Durch die Auswahl eines Zelltyps können die entsprechenden Zellen markiert werden. Zusätzlich ist eine Liste der Zellklassen vorhanden, in der eine Klasse gewählt werden kann. Entsprechend werden die Zellen, die die Regel dieser Klasse erfüllen, markiert.

Und schließlich ist das Panel *Zeit* für die Regelung des Zeitparameters vorhanden. Hier kann mithilfe eines Schiebereglers durch den zeitlichen Ablauf der Simulation navigiert werden.

Visualisierung von Zelldifferenzierungssimulationen

Wird die 2D-Ansicht aktiviert, so wird in der oberen rechten Ecke ein kleines Übersichtsfenster mit der dreidimensionalen Darstellung des Zellhaufens platziert, damit die Lage der Schnittebene im Zusammenhang mit dem dreidimensionalen Zellhaufen gesehen werden kann.

6 Beurteilung und Ausblick

Zum Anschluss soll noch ein Blick darauf geworfen werden, ob die ursprünglichen Ziele dieser Arbeit erreicht wurden. Ein Ziel der Arbeit war es, eine Analyse des Standes der Forschung im Bereich der Simulation und Visualisierung von Zelldifferenzierung zu machen.

Neben dem Stand der Forschung wurden zuerst in Kapitel 2 die biologischen Grundlagen vorgestellt, um einen Einblick zu gewinnen, wie Zelldifferenzierung funktioniert, welche Prozesse im Inneren der Zelle ablaufen und was den Zustand von Zellen ausmacht. Dies war zunächst notwendig, um beurteilen zu können, inwieweit die bisher existierenden Zell-Simulatoren auch den Anforderungen der Biologie genügen. Im weiteren Verlauf erwiesen sich die Grundlagen aus dem zweiten Kapitel als unabdingbares Hintergrundwissen, um die Anforderungen an *D-VISION* zu spezifizieren. Nur wenn man sich nach den Erfordernissen der aktuellen Zelldifferenzierungsforschung richtet, kann man auch eine sinnvolle Implementierung entwerfen.

Die Recherche zu Kapitel 3 ergab, dass in der Forschung hauptsächlich biochemische Reaktions-Netzwerke, die mithilfe von Differentialgleichungen gelöst werden, für Zell-Simulationen benutzt werden. Sie repräsentieren die komplexen Vorgänge, die im Inneren einer Zelle ablaufen. Auch äußere Vorgänge wie Zellteilung oder Zellbewegung werden durch Reaktionen ausgelöst und sind deshalb ebenfalls durch diese Netzwerke berechenbar.

Neuere Forschungsarbeiten versuchen inzwischen, die biochemischen Modelle feiner zu gestalten und bilden dazu die konkreten Reaktionsabläufe in Zellen nach.

Als sich herausgestellt hatte, dass sich bisherige Simulatoren und Visualisierungssysteme zur Zelldifferenzierung nicht für die gestellten Anforderungen eignen, musste *D-VISION* ohne die Anbindung an einen bestimmten Simulator entwickelt werden. Da keine konkreten Daten oder Datenstrukturen vorlagen, musste ein neues Datenmodell für *D-VISION* entworfen werden.

Von den bekannten Simulatoren wurde die grundlegende Idee, Zellen durch enthaltene Substanzen und aktive Gene zu unterscheiden, berücksichtigt. Prinzipiell wurde bei der Entwicklung davon ausgegangen, dass ein zukünftiger Simulator, der mit *D-VISION* zusammenarbeiten könnte, diese Vorgehensweise als Grundlage nutzt. Das Visualisierungssystem wurde daher darauf ausgelegt, auf die Zustände von Zellen besonderen Wert zu legen.

Ein wichtiger Aspekt der Aufgabenstellung betraf die Vergleichbarkeit mit realen Mikroskopieaufnahmen.

Aus den bisher genannten Aspekten folgten die grundlegenden Informationen, die visualisiert werden mussten.

Die Zellen werden als 3D-Kugel dargestellt. Um eine realistische Darstellung der Zellen zu erreichen, wurden die Kugeln deformiert. Neben der Deformation bietet vor allem die eingesetzte Schnittebenen-Technik eine hohe Vergleichbarkeit mit zweidimensionalen Lichtmikroskopiebildern. Dagegen entspricht die dreidimensionale Ansicht eher solchen Bildern, wie sie Elektronenmikroskope liefern.

Neben den Zellen selbst, die in ihrer Form so realistisch wie möglich aussehen sollten, liegt ein besonderes Augenmerk auf der Darstellung von Substanzen und Konzentrationen. Die einzelnen in der Zelle ablaufenden Prozesse werden nur durch die zeitliche Änderung des Zellzustandes sichtbar. Der Zustand kann über verschiedene Interaktionsmöglichkeiten exploriert werden.

Um Zusammenhänge mit der Zelldifferenzierung erkennbar zu machen, wurde es ermöglicht, die Konzentrationsverläufe von bis zu drei verschiedene Stoffe gleichzeitig anzuzeigen. Insbesondere die Möglichkeit, Klassen basierend auf den Zellzuständen selbst zu definieren und anzeigen zu lassen, kommt dem Konzept vieler, bereits existierender Simulationsansätze entgegen. Die Konzeption des Datenmodells begründet sich also auf den biologischen Grundlagen, bleibt jedoch nahe an den von existierenden Simulatoren gelieferten Daten.

Die Anforderung, das entwickelte System solle eine Schnittstelle mit existierenden Simulatoren enthalten, konnte nur zum Teil erfüllt werden. Da kein Simulator direkt benutzt wird, ergibt sich die Schnittstelle durch das Konzept, welche Daten angezeigt werden sollen. So ist es zwar möglich und vor allem sinnvoll, die von *D-VISION* angebotenen Funktionen für Zellzustände und Zellklassen zu nutzen, es ist aber nicht dringend erforderlich, Substanzen und Gene mit anzugeben. Die wichtigste Voraussetzung für einen Simulator ist, dass er die Positionen und Größen der simulierten Zellen dreidimensional angeben muss. Mit dieser Minimalinformation kann bereits ein Bild generiert werden.

Eine grundlegende Schnittstelle in Form einer XML Dokument Definition (*dvision.dtd*) wurde erstellt. Daten können als XML-Dateien, die der Spezifikation genügen, eingelesen werden.

Die verwendete Datenstruktur dieser Schnittstelle bietet den besonderen Vorteil an, dass Simulationen nicht alle Daten gebündelt in einem großen Datenpaket übergeben müssen. Die Schnittstelle ermöglicht es, die Daten auch von mehreren Rechnern zu erhalten. Somit erfüllt *D-VISION* die Voraussetzungen, die der Einsatz von Grid Computing bei der Simulation stellt. Der Einsatz von Grid

Computing macht für Simulatoren durchaus Sinn. Zellsimulationen sind je nach Komplexität der Modelle mit einem hohen Rechenaufwand verbunden. Die Visualisierung selbst kann nicht durch den Einsatz mehrerer Rechner profitieren.

In Kapitel 4 wurde darauf eingegangen, dass die Möglichkeit besteht, den Aufbereiter zentral im Internet anzulegen. So kann eine verteilte Simulation die Daten an den Aufbereiter liefern. Außerdem können verschiedene Instanzen der Visualisierung ihre Daten von demselben Aufbereiter anfordern. Die Daten sind somit zentral verfügbar.

Das neue Visualisierungssystem *D-VISION* unterscheidet sich besonders durch die gewählten Darstellungsformen von den anderen existierenden Visualisierungen von Zellen und Zellhaufen. Der Abstraktionsgrad der Simulation wurde bisher in den meisten anderen Fällen mit in die Visualisierung übernommen. Die Repräsentation der Zellen nicht durch Kugeln oder Punkte sondern durch deformierte Kugeln wurde in keinem anderen Visualisierungssystem gefunden. Die Deformation der Zellen stellt jedoch die Grundvoraussetzung für die Herstellung eines sinnvollen, zweidimensionalen Schnittbildes dar. Andere Visualisierungen, die Zellen in Form von sich überschneidenden Kugeln darstellen, bieten zwar von außen einen ähnlichen Anblick, würde man hier jedoch ebenfalls ein 2D-Schnittbild erzeugen, so wären darauf viele, sich überschneidende Kreise zu sehen. Das Bild hätte insofern keine Ähnlichkeit mit realen Lichtmikroskopieaufnahmen.

Die Schnittbilder, die von *D-VISION* erzeugt werden können, sehen zwar auf den ersten Blick aus, wie Bilder, die von zweidimensionalen Visualisierungssystemen geliefert werden könnten, sie unterscheiden sich aber in einem wichtigen Punkt. Zweidimensional simulierte Zellen können nur aus einer einzigen Richtung beobachtet werden. Bei *D-VISION* ist es nicht nur möglich, den dreidimensionalen Zellhaufen zu rotieren, es können auch wie bei einem realen Mikroskop durch Veränderung des Fokus die einzelnen Zellschichten durch Verschieben der Schnittebene stufenlos durchlaufen werden.

Die dritte Besonderheit im Vergleich zu anderen Visualisierungen ist die räumliche Darstellung von Konzentrationen. Dazu werden Tiny Cubes benutzt. Sie wurden bisher für diesen Zweck nur unter Verwendung von Farbskalen und in einer regelmäßigen Anordnung benutzt. Bei *D-VISION* wird die Darstellung volumetrischer Daten durch die Anzahl der verwendeten Objekte repräsentiert. Außerdem wurde auf eine zufällige Verteilung der Würfel im Zellinnenraum geachtet.

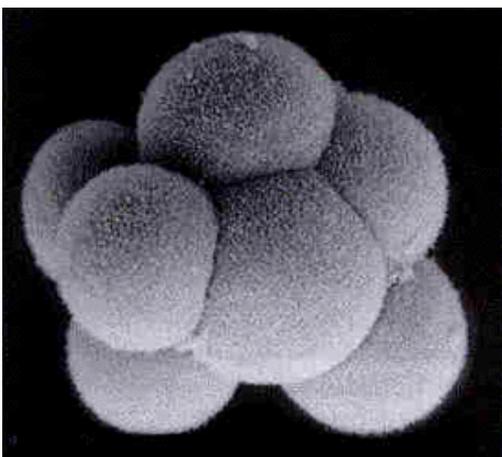
Zum Schluss bleibt noch zu sagen, dass *D-VISION* nur dann wirklich gut funktionieren kann, wenn ein Simulator das angebotene Datenmodell auch voll ausschöpft. Der Prototyp ist flexibel genug, um auch mit einfacheren Simulatoren zusammenzuarbeiten. Auf welchem Weg die visualisierten Daten gewonnen

werden, spielt keine Rolle. Auch reine Messwerte, wie es das „The Perfect C. elegans Project“ [KIT98] benutzt, können zu guten Bildern führen. Allerdings würde das den Chancen, die *D-VISION* bietet, nicht gerecht werden.

Die Möglichkeit, Zellklassen durch Regeln zu definieren, kann nur dann zur Visualisierung von Zelldifferenzierungen dienen, wenn die passenden Regeln aufgestellt werden.

Im Übrigen kann die prinzipielle Struktur, die der Interpretation von Substanzen und Genen zugrunde liegt, auch auf andere Zelleigenschaften angewendet werden. Gene werden als boolesche Werte behandelt. Jede weitere Zelleigenschaft, die ebenfalls durch einen booleschen Wert ausgedrückt werden kann, kann als „Gen“ visualisiert werden. Das Gleiche gilt für die Interpretation von Substanzen. Jede Zelleigenschaft, die sich in Form einer Menge oder Konzentration angeben lässt, kann bei der Visualisierung den Platz einer Substanz einnehmen. Das erweitert das Spektrum der Darstellungsmöglichkeiten um ein Vielfaches und lässt Spielraum auch für bisher unentdeckte Vorgänge bei der Zelldifferenzierung.

Abschließend bleibt zu hoffen, dass diese Arbeit von Herstellern neuer Zelldifferenzierungssimulatoren als Vorschlag angenommen wird, die von den Simulatoren errechneten Daten visuell zu repräsentieren. Dies würde der Arbeit neben den theoretischen Erkenntnissen auch einen praktischen Nutzen geben.



Anhang A: XML Dokument Definition

```
<!-- ===== -->
<!-- D-VISION Simualtion Data DTD
XML - Dokumenttypdefinition
fuer die Schnittstelle zum Aufbereiter von D-VISION

Hiemit wird festgelegt, wie die XML-Dateien, die die
Daten einer
Simulation enthalten, aufgebaut sein muessen, um in
D-VISION importiert
werden zu koennen.

Autor: Paul Izquierdo Rojas
-->

<!ELEMENT dvision ( simulation? , simulationdata* ) >

<!-- ===== -->
<!-- SIMULATION -->
<!-- Definition des Simulations-Tag -->
<!-- ===== -->

<!-- "simulation" enthaelt allgemeine Informationen ueber
eine Simulation. Mit dem Attribut "by" des Unterelementes
"cellsize" wird angegeben, ob die Zellgroesse in dieser
Simulaion durch einen Radius oder durch das Volumen
angegeben wird.
Das optionale Unterelement "units" gibt Masseinheiten
fuer die Anzeige der Daten durch D-VISION an.
-->
<!ELEMENT simulation ( name,
                        identifier,
                        date,
                        description,
                        cellsize,
                        units?
                      )
>
  <!ELEMENT name          (#PCDATA) >
  <!ELEMENT identifier    (#PCDATA) >
  <!ELEMENT date          (#PCDATA) >
  <!ELEMENT description   (#PCDATA) >
  <!ELEMENT cellsize     EMPTY >
  <!ELEMENT units        EMPTY >

  <!ATTLIST cellsize by      (radius|volume) #REQUIRED >
  <!ATTLIST units length    CDATA #IMPLIED
              volume      CDATA #IMPLIED
              concentration CDATA #IMPLIED
  >

<!-- ===== -->
<!-- SIMULATIONS DATEN -->
<!-- Definition der Simulations-Daten -->
<!-- ===== -->

<!-- "simulationsdata" enthaelt Informationen ueber die
Zellhaufen zu bestimmten Zeitpunkten oder ueber die
Zelllinien. Ueber das Attribut "simulation" wird die
Simulation, zur der die Daten gehoeren, spezifiziert.
```

Visualisierung von Zelldifferenzierungssimulationen

```
-->
<!ELEMENT simulationdata ( (cellheapsnapshot | celllines)* ) >
  <!ATTLIST simulationdata      simulation ID #REQUIRED >

<!-- "cellheapsnapshot" ist ein Zellhaufen zu einem bestimmten
Zeitpunkt. Es enthaelt die entsprechenden Zellen.
ueber die Attribute "day" und "time" wird der Zeitpunkt
bestimmt.
-->
<!ELEMENT cellheapsnapshot (cell*) >
  <!ATTLIST cellheapsnapshot  day   CDATA #REQUIRED
                                time  CDATA #REQUIRED
  >

<!-- "cell" entspricht einer Zelle zu einem bestimmten
Zeitpunkt. Sie enthaelt neben der Information ueber die
Lage und die Groesse auch moegliche Zelltypen der Zelle,
sowie die Daten zu den Substanzen im Inneren und ausserhalb
der Zelle und die Daten der Gene. Die Groesse wird entweder
durch einen Radius oder durch das Volumen angegeben.
-->
<!ELEMENT cell ( position,
                (radius | volume),
                type*,
                innersubstances?,
                outersubstances?,
                genes?
                )
>
  <!ATTLIST cell      identifier ID #REQUIRED >

  <!ELEMENT position      EMPTY >
  <!ELEMENT radius       EMPTY >
  <!ELEMENT volume       EMPTY >
  <!ELEMENT type         (#PCDATA) >
  <!ELEMENT innersubstances (substance*) >
  <!ELEMENT outersubstances (substance*) >
  <!ELEMENT genes       (gen*) >

  <!ATTLIST position  x      CDATA #REQUIRED
                    y      CDATA #REQUIRED
                    z      CDATA #REQUIRED
  >
  <!ATTLIST radius    value  CDATA #REQUIRED >
  <!ATTLIST volume    value  CDATA #REQUIRED >

  <!ELEMENT substance      EMPTY >
  <!ELEMENT gen            EMPTY >

  <!ATTLIST substance  name      CDATA #REQUIRED
                    concentration CDATA #REQUIRED
  >
  <!ATTLIST gen        name      CDATA #REQUIRED
                    active     (true|false) #REQUIRED
  >

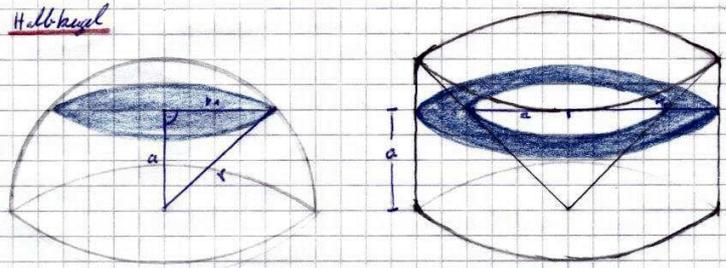
<!-- "celllines" wird benutzt um die Zelllinien anzugeben.
Mit "cellrelation" werden die Tochterzellen einer Zelle
angegeben.
-->
<!ELEMENT celllines (cellrelation*) >

<!ELEMENT cellrelation EMPTY >
  <!ATTLIST cellrelation identifier ID #REQUIRED
                    child1 IDREF #IMPLIED
                    child2 IDREF #IMPLIED
  >
```

Anhang B: Volumenberechnung deformierter Kugeln

Der Satz von Cavalieri besagt, dass zwei Körper das gleiche Volumen haben wenn sie gleiche Grundflächen und in gleicher Höhe gleiche Querschnitte haben. Damit lässt sich die Formel für das Volumen eines deformierten Kugelsegmentes herleiten:

Halbkuugel



$$A = \pi \cdot r_1^2 = \pi (r^2 - a^2)$$

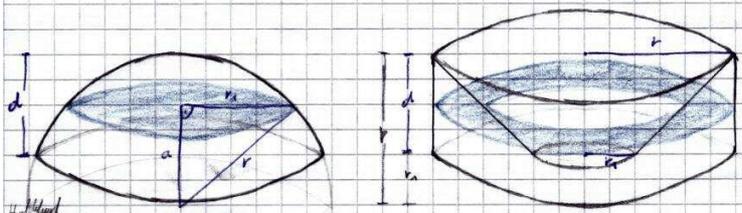
$$A = \pi r^2 - \pi a^2$$

⇒ $V_{\text{Halbkugel}} \stackrel{!}{=} V_{\text{Zylinder}} - V_{\text{Kegel}}$

$$V_{HK} = \pi r^2 \cdot r - \frac{1}{3} \pi r^2 \cdot r = \frac{2}{3} \pi r^3$$

⇒ $V_{\text{Kugel}} = \frac{4}{3} \pi r^3$

Kugelsegment



$r_1 = r - d$

$$V_{\text{Kugelsegment}} \stackrel{!}{=} V_{\text{Zylinder}} - V_{\text{Kegelstumpf}}$$

$$V_{KS} = \pi r^2 d - (V_{\text{Kegel Radius } r} - V_{\text{Kegel Radius } r_1})$$

$$= \pi r^2 d - \left(\frac{1}{3} \pi r^2 \cdot r - \frac{1}{3} \pi r_1^2 \cdot r_1 \right)$$

$$= \pi r^2 d - \left(\frac{1}{3} \pi r^3 - \frac{1}{3} \pi (r-d)^3 \right)$$

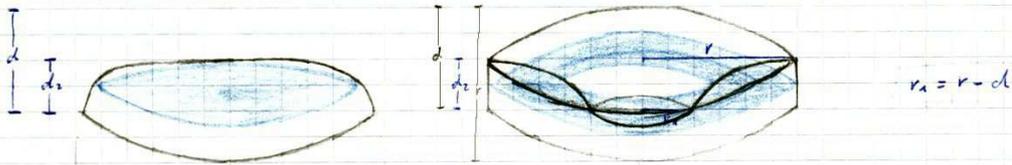
$$= \pi r^2 d - \left(\frac{1}{3} \pi r^3 - \frac{1}{3} \pi (r^3 - 3r^2 d + 3r d^2 - d^3) \right)$$

$$= \pi r^2 d - \left(\frac{1}{3} \pi r^3 - \frac{1}{3} \pi r^3 + \pi r^2 d - \pi r d^2 + \frac{1}{3} \pi d^3 \right)$$

$$= \pi r^2 d - \pi r^2 d + \pi r d^2 - \frac{1}{3} \pi d^3$$

⇒ $V_{KS} = \frac{1}{3} \pi d^2 (3r - d)$

Deformiertes Kugelsegment



Volumen des Kugelsegment \cong $V_{\text{Zylinder}} - V_{\text{K}}$

Bestimmung des Volumens V_{K} \leftarrow über rotationsvolumen des Integralrechnung

● linearer Dipol $n=1$
● quadratischer Dipol $n=2$
● kubischer Dipol $n=3$

$f(x) = r - d \cdot \left(\frac{x}{d_2}\right)^{1/n}$

$$V_{\text{K}} = \pi \int_0^{d_2} f(x)^2 dx$$

$$= \pi \int_0^{d_2} \left(r - d \cdot \frac{x^{1/n}}{d_2^{1/n}} \right)^2 dx$$

$$= \pi \int_0^{d_2} \left(r^2 - 2r d \frac{x^{1/n}}{d_2^{1/n}} + d^2 \frac{x^{2/n}}{d_2^{2/n}} \right) dx$$

$$= \pi \left[r^2 x - \frac{n}{n+1} 2r d \frac{x^{(n+1)/n}}{d_2^{1/n}} + \frac{n}{n+2} d^2 \frac{x^{(n+2)/n}}{d_2^{2/n}} \right]_0^{d_2}$$

$$= \pi \left(r^2 d_2 - \frac{n}{n+1} 2r d \frac{d_2^{(n+1)/n}}{d_2^{1/n}} + \frac{n}{n+2} d^2 \frac{d_2^{(n+2)/n}}{d_2^{2/n}} \right) - \pi(0)$$

$$= \pi \left(r^2 d_2 - \frac{n}{n+1} 2r d + \frac{n}{n+2} d^2 \right)$$

$$= \pi r^2 d_2 + \pi d d_2 \left(-\frac{n}{n+1} 2r + \frac{n}{n+2} d \right)$$

$V_{\text{def KS}} = \pi r^2 d_2 - \left(\pi r^2 d_2 + \pi d d_2 \left(-\frac{n}{n+1} 2r + \frac{n}{n+2} d \right) \right)$

$V_{\text{def KS}} = \pi d d_2 \left(\frac{n}{n+1} 2r - \frac{n}{n+2} d \right)$

für $n=3$ $\pi d d_2 \left(\frac{3}{2} r - \frac{3}{5} d \right) = \frac{3}{10} \pi d d_2 (5r - 3d)$

für $n=2$ $\pi d d_2 \left(\frac{4}{3} r - \frac{1}{2} d \right) = \frac{1}{6} \pi d d_2 (8r - 3d)$

Anhang C: Glossar

Adhäsion

Die Adhäsion (auch Anhangskraft) bezeichnet die Zusammenhangkräfte zwischen den Molekülen zweier verschiedener Stoffe.

<http://www.net-lexikon.de/Adhaesion.html>

Algorithmus

[griechisch] der, Folge von exakten Arbeitsanweisungen zum Lösen einer Rechenaufgabe in endlich vielen, eindeutig festgelegten, auch wiederholbaren Schritten.

API

[Abk. für Application Programming Interface; dt. »Schnittstelle für die Programmierung von Anwendungsprogrammen«] das, eine genormte Schnittstelle, die es sowohl ermöglicht, von einem Anwendungsprogramm aus auf bestimmte Funktionen eines Betriebssystems (u.a. bei Windows, Windows NT/2000, OS/2, OS/390 und Linux) oder einer Benutzeroberfläche zuzugreifen, als auch auf Softwareprogramme insgesamt.

Man unterscheidet dabei verschiedene API-Arten, z.B. MAPI für Mailboxen, ODAPI (Abk. für Open Database API, dt. »API für offene Datenbanken) für Datenbanksysteme sowie TAPI für Verbindungen über Modems.

Blastozyste

(Blasenkeim, Blastula), Bezeichnung für die Keimblase, die sich etwa am vierten Tag nach der Befruchtung aus der befruchteten Eizelle entwickelt hat; dabei bildet sich eine Höhle (das Keimbläschen oder Blastozöl), und es lässt sich erstmals eine äußere Zellschicht (Trophoblast) und eine innere Zellmasse (Embryoblast) unterscheiden. Der Trophoblast dient zur Ernährung und als Hülle, aus dem Embryoblasten bildet sich der eigentliche Embryo.

Clipping

[dt. Abschneiden], eine Methode bei der Erstellung von 3D-Grafiken, die es ermöglicht, verdeckte, also nicht sichtbare Bereiche auszuschneiden, beispielsweise beim Betrachten einer Landschaft durch ein Fenster alle Bereiche, die nicht innerhalb des Fensterrahmens liegen.

Datenbank

[engl. database] (Datenbankprogramm, Datenbanksystem, Datenbank-Managementsystem, Abk. DBMS), System zur Speicherung und Verwaltung von großen Datenmengen. Es besteht aus einer Datenbasis, in der die Daten nach einer bestimmten Struktur abgelegt werden, und Verwaltungsprogrammen, die Daten abspeichern, suchen oder andere Operationen mit den Daten durchführen.

Deformation

[lateinisch] die, allgemein: Formveränderung, Abweichen von der normalen Gestalt, Missbildung.

Desoxyribonucleinsäure

Abkürzung DNS, englische Abkürzung DNA (von desoxyribonucleic acid), polymere Verbindung, die aus Desoxyribonucleotiden aufgebaut ist. Die Nucleotide bestehen jeweils aus einer organischen Base (Adenin, Thymin, Guanin oder Cytosin), einem Zuckermolekül (Desoxyribose) und einem Phosphorsäurerest. Ein DNA-Molekül besteht in der Regel aus zwei Polynucleotidsträngen, die schraubig umeinander gewunden (Doppelhelix) und durch Wasserstoffbrücken zwischen den Basen miteinander verbunden sind. Die genetische Information ist in der DNA als Sequenz der vier verschiedenen Basen verschlüsselt. (DNA-Replikation, Nucleinsäuren)

Visualisierung von Zelldifferenzierungssimulationen

Enzyme

[griechisch] (früher Fermente), in der Zelle gebildete hochmolekulare Proteine, die als Biokatalysatoren der Organismen deren Stoffwechsel katalytisch steuern.

Eukaryonten

[griechisch] (Eukaryoten, Eukaryota), zusammenfassende Bezeichnung für alle Organismen, deren Zellen einen echten Zellkern besitzen.

extrazellulär

[lateinisch], außerhalb der Zelle gelegen; Gegensatz: intrazellulär.

Furchung

(Blastogenese), erste Phase der Embryonalentwicklung, in der sich die befruchtete Eizelle ohne Volumenzunahme in Furchungszellen (Blastomeren) aufteilt. Der Furchungsverlauf ist von der Dottermenge des Eies abhängig und äußerlich durch das Auftreten von Furchen gekennzeichnet. (Entwicklung)

Gastrulation

[griechisch-neulateinisch] die, frühembryonaler Entwicklungsvorgang bei Tieren, der von der einschichtigen Blastula durch Einstülpung, Umwachsung, Zelleinwanderung oder Abspaltung zur zweischichtigen, aus Ekto- und Entoderm bestehenden Gastrula (Becherkeim) führt.

Grid Computing

[dt. »(auf einem) Gitter rechnen«, verteiltes Rechnen] das, das Benutzen von Ressourcen entfernter Computer für die elektronische Datenverarbeitung. Dieses Konzept kann als Erweiterung des Parallelrechners angesehen werden, bei dem mehrere Prozessoren in einem Computer mithilfe von Bussystemen zusammenarbeiten. Beim Grid Computing findet der Datenaustausch zwischen den kommunizierenden Prozessoren dagegen über Internetverbindungen statt. Obwohl diese erheblich langsamer als lokale Datenleitungen sind, wird dieser Nachteil durch die schiere Anzahl der über das Internet zugänglichen Rechenkapazitäten mehr als wettgemacht.

GUI

[Abk. für Graphical User Interface, dt. »grafische Benutzeroberfläche«, Benutzeroberfläche. der Teil des Betriebssystems oder eines Anwendungsprogramms, der dem Nutzer die Kommunikation mit dem Computer ermöglicht, also die Schnittstelle zwischen Mensch und Maschine.

hydrophil

[griechisch], Chemie, Physik: Wasser anziehend, in Wasser löslich.

hydrophob

[griechisch], Chemie, Physik: Wasser abstoßend, nicht in Wasser löslich.

Java

[, englisch; umgangssprachliche Bezeichnung für »Kaffee«, von der amerikanischen Firma Sun Microsystems entwickelte objektorientierte Programmiersprache, die vom verwendeten Betriebssystem unabhängig ist. Java wird v.a. zum Programmieren von Anwendungen für das WWW eingesetzt.

Katalysator

der, Chemie: Stoff, der durch Bildung aktiver Zwischenprodukte eine chemische Reaktion ermöglicht, beschleunigt oder in eine bestimmte Richtung lenkt. Katalysatoren setzen die Aktivierungsenergie der Reaktion herab und erhöhen dadurch ihre Reaktionsgeschwindigkeit, ohne selbst verbraucht zu werden.

Kompartiment

[lateinisch] das, Zellbiologie: durch eine einfache Elementarmembran oder durch Doppelmembran abgegrenzter Bezirk oder Reaktionsraum in einer Zelle. Die Membranen können Barrieren für bestimmte Substanzen sein. Größere Kompartimente mit besonderen Funktionen heißen Zellorganellen.

Lipide

[griechisch], Sammelbezeichnung für Fette und fettähnliche Stoffe (Lipoide), die von pflanzlichen und tierischen Organismen gebildet werden.

Membran

[lateinisch »Haut«, »Häutchen«] die, Biologie: dünnes, feines Häutchen, das trennende oder abgrenzende Funktion hat (z.B. Trommelfell).

Mitose

[griechisch] die (indirekte Kernteilung), Kernteilungsvorgang, bei dem aus einem Zellkern zwei Tochterkerne gebildet werden, die gleiches (mit dem Ausgangsmaterial identisches) Genmaterial und die gleiche Chromosomenzahl haben.

Morphogenese

[griechisch] die (Morphogenesis, Morphogenie), Ausformung eines Lebewesens während seiner Entwicklung; ist von den Erbanlagen sowie von Reizen aus der Umwelt abhängig.

multivariat

[aus gleichbedeutend englisch multivariate], mehrere Variablen betreffend, mehrdimensional; insbesondere in der Statistik verwendet (z.B. multivariate Statistik, multivariate Analyse).

Polygon

[griech. polygonos »vieleckig«, »vielwinklig«] das, (Vieleck), eine geschlossene ebene Figur, die von mehreren Strecken begrenzt wird.

Prokaryonten

[griechisch] (Prokaryoten, Akaryobionten), Organismen ohne echten Zellkern: Archaeobakterien, Bakterien, Cyanobakterien.

Proteine

[griechisch] (Eiweiße), aus Aminosäuren zusammengesetzte Makromoleküle mit vielfältigen Funktionen in allen Organismen. Da Proteine zu einem beträchtlichen Teil aus essenziellen Aminosäuren bestehen, die nur von grünen Pflanzen aufgebaut werden können, müssen Mensch und Tier Proteine über die Nahrung aufnehmen, um körpereigene Proteine aufbauen zu können.

Rendering

[dt. »Wiedergabe«, »Gestaltung«], die realitätsnahe Gestaltung dreidimensionaler Objekte durch Farb- und Lichteffekte.

Ribose

die, zu den Pentosen gehörendes Monosaccharid, das als Baustein u.a. in Coenzymen und in Nucleinsäuren vorkommt. Ribose wird im Körper aus Glucose gebildet.

Ribosomen

Pl., aus zwei Untereinheiten bestehende, elektronenmikroskopisch sichtbare Körnchen mit einem Durchmesser von 1520 nm, die aus Ribonucleinsäure und Eiweißen aufgebaut sind. Polyribosomen sind die Orte der Proteinsynthese.

Sequenz

[lateinisch sequentia »Folge«] die, Biochemie: (Primärstruktur), Abfolge der Bausteine in biologischen Makromolekülen, z.B. die Aminosäuresequenz in Proteinen und Peptiden und die Nucleotidsequenz in den Nucleinsäuren.

Simulation

[lat. »Vorspiegelung«], in der Informatik die Nachbildung von Vorgängen mithilfe eines Computers. Sie wird zur Untersuchung von Abläufen eingesetzt, die man in der Wirklichkeit aus Zeit-, Kosten-, Gefahren- oder anderen Gründen nicht durchführen kann oder will. Typische Beispiele sind das Studium von möglichen Umweltkatastrophen, Pilotentraining, Entwicklung neuer Auto- oder Flugzeugtypen usw. Geräte und Programme, die eine Simulation durchführen, bezeichnet man als Simulatoren.

SQL

[Abk. für Structured Query Language, dt. »strukturierte Abfragesprache«, Aussprache wie »siekwel« oder »es-ku-el«], eine seit den frühen 1970er-Jahren von IBM unter dem Namen Sequel entwickelte Abfragesprache zur Bearbeitung der Inhalte relationaler Datenbanken. 1981 wurde Sequel in SQL umbenannt.

Stammzellen

undifferenzierte, das heißt keinem endgültigen Zelltyp angehörende Zellen eines Organismus, die definitorisch zwei Funktionen erfüllen müssen. Eine Funktion besteht in der Selbsterneuerung der Stammzellen durch Zellteilung, bei der identische Tochterzellen entstehen. Die zweite Funktion umfasst die Möglichkeit, Tochterzellen zu generieren (hervorzubringen), die zu bestimmten Zellarten oder Zelltypen differenzieren (ausreifen) können.

XML

[Abk. für Extensible Markup Language, dt. »erweiterbare Beschreibungssprache«], eine Sprache, mit der die Struktur von Dokumenten beschrieben wird (eine sog. Metasprache). XML ist eine eingeschränkte Form (man sagt auch: eine Untermenge) von SGML. Ein XML-Dokument ist medien- und plattformneutral, es kann unter allen Betriebssystemen bearbeitet und sowohl auf einem Drucker als auch in elektronischer Form, auf einer CD-ROM oder im Internet, ausgegeben/publiziert werden.

Zelle

Zellen sind die kleinsten Bauelemente lebender Organismen - es gibt Lebewesen, die nur aus einer Zelle (Einzeller) bestehen, und andere, wie der Mensch, die sich aus einer Vielzahl einzelner Zellen (beim Menschen ca. 10000 Milliarden) zusammensetzen (Vielzeller).

Zellorganellen

Biologie: in Zellen vorkommende organähnliche Gebilde mit spezifischer Struktur und Funktion, z.B. Mitochondrien und Plastiden.

Zytoplasma

außerhalb des Zellkerns gelegenes Zellplasma der eukaryotischen Zellen, von der Zellmembran nach außen begrenzt. Grundkomponente des Zytoplasmas ist das Grundplasma (Hyaloplasma), das alle wesentlichen chemischen Zellbausteine (Wasser, Salze, Proteine, Lipide, Nucleinsäuren) enthält. In das Zytoplasma sind die Zellorganellen sowie für spezielle Zelleistungen erforderliche Strukturen (z.B. Mikrotubuli) oder auch tote Einschlüsse in flüssiger oder fester Form (z.B. Sekrete, Fette, Proteinkristalle, Pigmente) eingebettet.

Für alle Definitionen gilt, wenn nicht anders angegeben:

(c) Bibliographisches Institut & F. A. Brockhaus AG, 2003.

Anhang D: Literaturverzeichnis

- [AGA93] P. Agarwal: *Cell-based Computer Models in Developmental Biology*, PhD thesis, Courant Institute of Mathematical Sciences, New York University, 1993
<http://citeseer.ist.psu.edu/133889.html>
- [AGA94] P. AGARWAL: *The Cell Programming Language*, *Artificial Life II* (1), S. 37-77, 1994
<http://citeseer.ist.psu.edu/14750.html>
- [ALB95] B. ALBERTS, D. BRAY, J. LEWIS, M. Raff, K. Roberts, J. D. Watson: *Molekularbiologie der Zelle, Dritte Auflage*, VCH, Weinheim, 1995
- [BAK02] C. A. H. BAKER, M. S. T. CARPENDALE, P. PRUSINKIEWICZ, M. G. SURETTE: *GeneVis: Visualization Tools for Genetic Regulatory Network Dynamics*, Proceedings of the IEEE Conference on Visualization, 2002.
- [BAR01] J. BARRILLEAUX: *3D User Interfaces with Java3D*, Manning, 2001
- [BIL03] T. BILL: *Einfach ausprobieren: Präzise Vorhersagen per Computersimulation*. In: *c't* 16/2003, S. 170-175, Heise, 2003
- [BOU88] P. BOURKE: *private Homepage*, The Swinburne Center for Astrophysics and Supercomputing, (online), seit 1988
<http://astronomy.swin.edu.au/~pbourke/>
- [BRO04] F. A. BROCKHAUS AG: *Brockhaus*, (online), Stand 2004
<http://www.brockhaus.de/>
- [CHA04] R. CHATURVEDI, C. HUANG, B. KAZMIERCZAK, T. SCHNEIDER, J. A. IZAGUIRRE, S. A. NEWMAN, J. A. GLAZIER, M. ALBER: *On Multiscale Approaches to 3-Dimensional Modeling of Morphogenesis*, *an Journal of Computational Physics* gesendet, 2004
<http://www.nd.edu/~izaguirr/papers/CHAT0x.pdf>

- [DEL94] F. DELLAERT, R. D. BEER: *Toward an Evolvable Model of Development for Autonomous Agent Synthesis*. In: R. A. BROOKS, P. MAES (Hrsg.): *Artificial Life IV*, S. 246-257, MIT, Cambridge, 1994
<http://citeseer.ist.psu.edu/dellaert94toward.html>
- [DOW03] T. DOWNER: *McKoi SQL Database Manual*, (online), 2003
<http://mckoi.com/database/docindex.html>
- [ECK00] R. ECKSTEIN: *XML kurz & gut*, O'Reilly, 2000
- [FLE93] K. FLEISCHER, A. H. BARR: *A Simulation Testbed for the Study of Multicellular Development: The Multiple Mechanisms of Morphogenesis*. In: C. G. LANGTON (Hrsg.): *Artificial Life III*, S. 389-416, Addison-Wesley, 1993
<http://citeseer.ist.psu.edu/fleischer93simulation.html>
- [FLE96] K. FLEISCHER: *Investigations with a Multicellular Developmental Model*. In: C. G. LANGTON, K. SHIMOHARA (Hrsg.): *Artificial Life V*, S. 229-236, MIT, 1996
<http://citeseer.ist.psu.edu/fleischer96investigation.html>
- [GRA01] G. GRAMELSBERGER: *Computersimulationen - Neue Instrumente der Wissensproduktion*, 2001
<http://www.sciencepolicystudies.de/Gramelsberger-Neu-Expertise.pdf>
- [GRO98] A. GROTE: *Versuchstier Computer: Computersimulationen ersetzen Tierversuche*. In: c't Januar 98, S. 60-63, Heise, 1998
- [HEA99] C. HEALEY: *Preattentive Processing*, (online), 1999
<http://www.csc.ncsu.edu/faculty/healey/PP/>
- [IAB95] INSTITUTE FOR ADVANCED BIOSCIENCES: *E-Cell Project*, (online), Keio University, seit 1995
<http://www.e-cell.org/>
- [IZA04] J. A. IZAGUIRRE, R. CHATURVEDI, C. HUANG, T. CICKOVSKI, J. COFFLAND, G. THOMAS, G. FORGACS, M. ALBER, S. A. NEWMAN, J. A. GLAZIER: *CompuCell, a multi-model framework for simulation of morphogenesis*, im Druck, Bioinformatics, 2004.
<http://www.nd.edu/~izaguirr/papers/bioinformatics.pdf>

- [JAC94] C. JACOB: *Genetic L-System Programming*. In: Y. DAVIDOR, H. P. SCHWEFEL, R. MÄNNER (Hrsg.): *Parallel Problem Solving from Nature III*, S. 334-343, Jerusalem, 1994
<http://citeseer.ist.psu.edu/15611.html>
- [KIT98] H. KITANO, S. HAMahashi, S. Luke: *The Perfect C. elegans Project: An Initial Report*. In *Artificial Life IV (2)*, S. 141-156, MIT, 1998
- [KLE99] H. KLEINIG, P. SITTE: *Zellbiologie, 4. Auflage, neu bearbeitet von: H. KLEINIG, U. MAIER*, Gustav Fischer Verlag, Stuttgart, Jena, Lübeck, Ulm, 1999
- [KUN03] S. KUMAR, P. J. BENTLEY: *Mechanisms of Oriented Cell Division in Computational Development*. In proceedings of the first Australian Conference on Artificial Life (ACAL 2003), Canberra, Australia, 2003
<http://www.cs.ucl.ac.uk/staff/Sanjeev.Kumar/publications.htm>
- [LAN97] M. LANTIN: *Computer Simulations of Developmental Processes*, 1997
- [LIC67] J. C. R. LICKLIDER: *Interactive Dynamic Modeling*. In: G. SHAPIRO, M. ROGERS (Hrsg.): *Prospects for Simulation and Simulators of Dynamic Systems*, New York/London, S. 281-289, 1967
- [MAX02] MAX-PLANCK-GESELLSCHAFT: *vCell. die virtuelle Zelle*, (online), seit 2002
<http://www.vcell.de/>
- [MEN96] P. MENDES: *Gepasi: Biochemical Simulation*, (online), seit 1996
<http://www.gepasi.org>
- [MOH98] S. MOHAN: *The Forth Generation of 3D Graphics APIs has arrived!*, 1998
http://java.sun.com/products/java-media/3D/collateral/wp_mktg/j3d_wp.pdf
- [NET98] AKADEMIE.DE: *Net-Lexikon*, (online), seit 1998
<http://www.net-lexikon.de/>

- [NIE90] G. M. NIELSON, B. HAMANN: *Techniques for the interactive visualization of volumetric data*. IEEE: Proceedings of the 1st conference on Visualization '90, Session: volume visualization techniques, S. 45-50, San Francisco, 1990
- [PAL01] I. PALMER: *Essential Java 3D fast: Developing 3D Graphics Applications in Java*, Springer, 2001
- [PRU93] P. PRUSINKIEWICZ: *Modeling and Visualization of Biological Structures*. In: Proceeding of Graphics Interface '93, S. 128-137, Toronto, 1993.
<http://www.cpsc.ucalgary.ca/Research/bmv/papers/vismodels.gi93.html>
- [PRU96] P. PRUSINKIEWICZ, M. HAMMEL, J. HANAN, R. MECH: *Visual models of plant development*. In: G. ROSENBERG, A. SALOMAA (Hrsg.): *Handbook of Formal Languages*, S. 535-597, Springer, Berlin, 1996
<http://www.cpsc.ucalgary.ca/Research/bmv/papers/handbook.95.html>
- [PRU97] P. PRUSINKIEWICZ, M. HAMMEL, AND R. MECH: *Visual Models of Morphogenesis: A Guided Tour*, (online), 1997
<http://www.cpsc.ucalgary.ca/Research/bmv/vmm-deluxe/TitlePage.html>
- [RAS97] J. RASHBASS, A. VAWER: *The biological toolbox: a computer program for simulating basic biological and pathological processes*. In: T. GROTH (Hrsg.): *Computer Methods and Programs in Biomedicine*, B. 52 (3), S. 203-211, 1997
- [ROD02] L. RODRÍGUEZ HERNÁNDEZ: *Stochastische Modellierung der Nanopartikelbildung in Mikroemulsionen*, Fakultät II – Fachgruppe Chemie – der Technischen Universität Berlin, Berlin, 2002
- [RYD99] E. F. RYDER, J. H. HONE, J. OLMSTEAD, M. O. WARD: *Graphical simulation of early development of the cerebral cortex*. In: *Computer Methods and Programs in Biomedicine*, Band 59, S. 107-114, 1999
- [SCH00] H. SCHUMANN: *Visualisierung: Grundlagen und allgemeine Methoden*, Springer, Berlin Heidelberg, 2000

- [SHA01] B. E. SHAPIRO, E. D. MJOLSNESS: *Developmental Simulations with Cellerator*, Proceedings of the Second International Conference on Systems Biology, S.342-351, 2001
<http://www-aig.jpl.nasa.gov/public/mls/cellerator/>
- [SPR95] AXEL SPRINGER AG: *Die Welt – Archiv*, (online), seit 1995
<http://www.welt.de/finden/index.htx>
- [STI04] W. STIELER: *Der große summende Gott*. In: c't 7/2004, S. 84-86, Heise, 2004
- [STO00] A. STORK: *Effiziente 3D-Interaktions- und Visualisierungstechniken für benutzer-zentrierte Modellierungssysteme*, Fachbereich Informatik der Technischen Universität Darmstadt, Darmstadt, 2000
- [SUN02] SUN: *The Java3D Tutorial*, (online), 2002
<http://java.sun.com/products/java-media/3D/collateral/>
- [SUN03] SUN: *Java3D API Documentation*, (online), 2003
http://java.sun.com/products/java-media/3D/forDevelopers/J3D_1_3_API/j3dapi/
- [TUR52] A. TURING: *The chemical basis for morphogenesis*, Philosophical Transactions of the Royal Society of London, B 237, S. 37-72, 1952
<http://www.turingarchive.org/browse.php/B/22>
- [ULL02] C. ULLENBOOM: *Java ist auch eine Insel*, 2. Aufl., Galileo Computing, 2002