

UNIVERSITÄT LEIPZIG

Fakultät für Mathematik und Informatik
(Institut für Informatik)

**Ein Wachstumsgrammatikinterpreter für
neuroanatomische Simulationen**

DIPLOMARBEIT

Leipzig, August 2004 vorgelegt von

Rose, Enrico
geb. am: 30.10.1978
Studiengang Informatik

Inhaltsverzeichnis

1	Einleitung	7
2	Aufgabenstellung, Zielsetzung	7
I	Grundlagen	8
3	Repräsentation der Neuronen-Morphologie	8
3.1	Terminologie der Neuronen-Morphologie	8
3.2	Repräsentation von Segmenten und Sektionen	8
3.3	Repräsentation von Verzweigungen	9
4	L-System Modellierung	10
4.1	Lindenmayer Systeme	10
4.1.1	Turtle-Geometrie	10
4.2	Deterministische L-Systeme	12
4.2.1	Formale Definition von L-Systeme	12
4.3	L-Systeme mit Baumstruktur	12
4.4	Stochastische L-Systeme	13
4.5	Kontextabhängige L-Systeme	14
4.6	Parametrisierte L-Systeme	15
4.7	Sensitive Wachstumsgrammatiken	15
5	Mathematik	17
5.1	Definition Wahrscheinlichkeitsverteilung	17
5.2	Gammafunktion	18
5.3	Stetige Gleichverteilung	18
5.4	Normalverteilung	19
5.5	Gammaverteilung	19
II	Eigenschaften des Programms	19
6	Tabellen	20
7	Analyse-Fenster	21
8	Generierung, Generate-Fenster	23
9	L-System-Fenster	23
10	SWC-Fenster	26
11	Die Datei Neuron.cfg und die Skript Befehle	28
11.1	Befehle in Neuron.cfg	28
11.1.1	#, #...# oder Kommentare	29
11.1.2	"..."	29
11.1.3	B, Button, SetButtonText	29
11.1.4	L, Label, SetLabelText	29
11.1.5	M, Menu, SetMenuText	29
11.1.6	ST, SegmentType, SetSegmentTypeName	30
11.1.7	VT, ValueType, SetValueTypeName	30
11.1.8	A, Attribut, SetAttributName	30
11.1.9	V, Value, SetValue	30

11.1.10 Alg, Algorithmus, CreateAlgorithmus	31
11.1.11 AV, AddValue	31
11.1.12 AA, AddAxiom	32
11.1.13 AR, AddRule	32
11.1.14 Fehlerbehandlung	32
12 Beispiele für die Bedienung	33
12.1 Analyse von Neuronen	33
12.2 Erzeugen von L-System	34
12.3 Generieren von L-System aus Neuronen	34
III Analyse von Neuronen	34
13 Algorithmen	35
13.1 Hillman Algorithmus	35
13.1.1 L-System	36
13.1.2 Hillman-Algorithmus Anwendung	38
13.2 Burke Algorithmus	39
13.2.1 L-System	41
13.2.2 Burke-Algorithmus Anwendung	44
13.3 Gleichungen von C.D.Murray	46
13.4 Gleichungen von Tamori	48
14 Analyse von Pyramidenzellen	48
14.1 Pyramidenzellen	48
14.2 Analysen	49
15 Erzeugen der Algorithmen	50
15.1 Gruppe SE von Genmanipulierten Mäusen	58
15.1.1 Erweiterter Hillman-Algorithmus	58
15.1.2 Erweiterter Burke-Algorithmus	59
15.2 Gruppe WT von Wilden Mäusen	61
15.2.1 Erweiterter Hillman-Algorithmus	61
15.2.2 Erweiterter Burke-Algorithmus	64
15.3 Vergleich der Algorithmen	66
16 Ergebnissbeschreibung	71
16.1 Vergleich mit existierender Software	71
16.2 Grenzen und Erweiterungsmöglichkeiten	72
IV Abschließende Bemerkung	73
V Anhang	74
17 Hardware und Software Voraussetzungen	74
18 Implementiere Fähigkeiten	75
19 Implementierung	77
19.1 Beschreibung der Klassen	78
19.1.1 main	78
19.1.2 CMainPanel	78
19.1.3 CGraphics3D	78

19.1.4	C3DData	78
19.1.5	C3DObject	78
19.1.6	C3DPoint	79
19.1.7	CTablePanel	79
19.1.8	CColumnData	79
19.1.9	CData	79
19.1.10	CValueAnalyseData	79
19.1.11	CValueData	80
19.1.12	CAlgorithmusData	80
19.1.13	CSkript	80
19.1.14	ITableUser	80
19.1.15	COutputPanel	80
19.1.16	CAxiomTable	80
19.1.17	CRuleTable	80
19.1.18	CInputPanel	81
19.1.19	CGraphicPanel	81
19.1.20	CSWCListTable	81
19.1.21	CDiagrammOptionTable	81
19.1.22	CDataListTable	81
19.1.23	CDiagrammPanel	82
19.1.24	CSkale	82
19.1.25	CListTable	82
19.1.26	CTreePanel	82
19.1.27	CGeneratePanel	82
19.1.28	CVariableData	83
19.1.29	CLSystem	83
19.1.30	CTGStack	83
19.1.31	CDistPos	83
19.1.32	CLinePos	83
19.1.33	CAxiom	83
19.1.34	CRule	84
19.1.35	CTo	84
19.1.36	CCommandList	84
19.1.37	CCommand	84
19.1.38	CValue	84
19.1.39	CVariable	84
19.1.40	CSWC	85
19.1.41	CSWCSegment	85
19.1.42	CSWCSektion	85
19.1.43	CSWCVerzweigung	85
19.1.44	CDataElement	85
19.1.45	CRandom	85
19.1.46	CInputDialog	85
19.2	Dateiformate	86
19.2.1	LSystem Datei	86
19.2.2	Turner Southampton Archive File oder SWC Dateien	86
19.2.3	Liste von SWC-Dateien	87
19.2.4	Neuron.dat	87
19.2.5	Neuron.cfg	87
19.2.6	Select Skript	87
19.3	Algorithmenbeschreibung	88
19.3.1	Mathematische Definitionen	88
19.3.2	Funktionen für die sensitive Wachstumsgrammatik	88
19.3.3	Berechnung der minimalen Entfernung Strecke-Punkt	91

19.3.4	Berechnung minimaler Entfernung zweier Geraden	92
19.3.5	Länge eines Strahles beim Treffen auf eine Strecke	92
19.3.6	Zufallsgeneratoren	92
19.3.7	Gammafunktion	93
19.3.8	Linearverteilung	93
19.3.9	Normalverteilung	93
19.3.10	Gammaverteilung	93
19.3.11	Winkel	94
19.3.12	Laden von Neuronen	94
19.3.13	Konversion von Neuron zum LSystem	94
19.3.14	Konvertieren vom L-System zum Neuron	95
20	Tabellen für die Bedienung des Programms	95
20.1	Tabellen	95
20.2	Analyse-Fenster	96
20.2.1	Menupunkte	96
20.2.2	Attribute	97
20.2.3	Wertetype	99
20.2.4	Segmenttypen	100
20.2.5	Attribute und Werttypen	100
20.3	Generate-Fenster	101
20.3.1	Buttons	101
20.4	L-System-Fenster	102
20.4.1	Menupunkte	102
20.4.2	Buttons	103
20.4.3	Turtle-Befehle	103
20.4.4	Arithmetische Befehle	104
20.4.5	Sensitive Befehle	105
20.4.6	Syntax von Befehle zur Bearbeitung von Axiomen und Regeln	105
20.5	SWC-Fenster	106
20.5.1	Menupunkte	106
20.5.2	Neurondarstellung	108
20.5.3	Markierungsmodi	109
20.6	Die Datei Neuron.cfg und die Skript Befehle	110
20.6.1	B, Button, SetButtonText	110
20.6.2	L, Label, SetLabelText	110
20.6.3	M, Menu, SetMenuText	111
20.6.4	ST, SegmentType, SetSegmentTypeName	113
20.6.5	VT, ValueType, SetValueTypeName	113
20.6.6	A, Attribut, SetAttributName	113
21	Tabellen für die Analyse von Neuronen	115
21.1	Analysewerte	115
21.2	Eigenschaften der originalen Neuronen	118
21.3	Werte für die Verteilungen $P_{br}(d)$ und $P_{trm}(d)$	121
21.4	Gruppe SE von Genmanipulierten Mäusen	124
21.4.1	Ergebnisse des erweiterten Hillman-Algorithmus bei TG Mäusen	124
21.4.2	Ergebnisse des erweiterten Burke-Algorithmus bei TG Mäusen	127
21.5	Gruppe WT von Wilden Mäusen	130
21.5.1	Ergebnisse des erweiterten Hillman-Algorithmus bei WT Mäusen	130
21.5.2	Ergebnisse des erweiterten Burke-Algorithmus bei WT Mäusen	133
22	UML-Diagramm	135
23	Daten und Inhalt der CD-ROM	137

1 Einleitung

Der Zusammenschluss von Biologie, Medizin und Informatik schreitet immer weiter voran. Computer messen Werte, analysieren und stellen dar damit leichter Diagnosen erstellt werden können. Schon seit einigen Jahren ist das Gehirn ins Zentrum der Aufmerksamkeit gerückt. Die Funktionen des Gehirns wollen verstanden werden, um etwa Patienten durch Mikroelektronik helfen zu können. So gibt es seit einigen Jahren Anstrengungen blinden Patienten durch künstliche Augen das Sehen wieder zu ermöglichen. Damit Projekte dieser Art erfolgreich sein können, muss das Verhalten von Neuronen klassifiziert und analysiert werden, damit es möglich ist Schnittstellen zwischen Mensch und Maschine zu etablieren. Bei den Analysen der Neuronen ist man bemüht, diese durch möglichst wenige Parameter zu beschreiben. Um etwa Neuronen und Neuronale Netze künstlich am Computer wachsen zu lassen, damit das Verhalten der Neuronen auf Reize untersucht werden kann.

Eine andere Einsatzmöglichkeit der Analyse von Neuronen ist die Klassifikation von Unterscheidungsmerkmalen von Neurongruppen. So werden etwa in dieser Diplomarbeit zwei Gruppen von Neuronen untersucht und ihre charakteristischen Unterschiede herausgestellt. Bei diesen Gruppen handelt es sich um jeweils 30 Pyramidenzellen, einmal von Mäusen des Wildtypes und einmal von genmanipulierten Mäusen. Diese beiden Gruppen wurden bei Experimenten im Paul-Flechsig-Institut für Hirnforschung der Universität Leipzig untersucht. Da bei dieser Diplomarbeit nicht nur die Analyse, sondern auch die Generierung von Neuronen im Vordergrund steht, wird ein Programm entwickelt. Zwar gibt es schon eine Reihe von Programmen für die Analyse und Generierung von Neuronen, doch zeigt sich, dass diese meist Algorithmen vorgeben, oder aus mehreren Programmen bestehen. Ziel der Eigenentwicklung soll es sein, dem Nutzer sowohl eine Plattform für die Algorithmenentwicklung zu bieten, ebenso, wie ein Werkzeug zur Analyse und Erzeugung von Neuronen. Dabei handelt es sich bei den Algorithmen um Vorschriften, welche die Erzeugung von Neuronen aus Analysewerten beschreiben.

Da die Beschreibung von Neuronen schon eine mehr als 30 jährige Tradition hat, gibt es mehrere Ansätze. Da gäbe es einmal die Reaktionsdiffusionssysteme, welche von einem Raum ausgehen, der verschiedene Stoffe enthält. Dabei können die chemischen Verbindungen, das Wachstum der Neuronen sowie das Verzweigen und Beenden von Fortsätzen hemmen oder fördern. Für die Dynamik des Systems sorgen die Diffusion der Stoffe und die Reaktion der chemischen Verbindungen miteinander. Die Berechnung des Modells erfolgt meist über Differenzialgleichungen oder über zeitdiskrete Approximationen.

Ein weitere Möglichkeit der Generierung von Neuronen ist das Benutzen von L-Systemen. Dabei sind L-Systeme ein, vom Biologen Aristid Lindenmayer, 1968 entwickeltes System von Grammatiken. Diese Grammatiken bestehen aus Axiomen und Regeln und lassen sich leicht beschreiben und simulieren, weshalb das entwickelte Programm L-System für die Beschreibung der Algorithmen zur Erzeugung von Neuronen benutzt. Da Algorithmen leicht in L-System transformiert werden können, lassen sich klassische Verfahren, wie etwa die Modelle von Hillman oder Burke, in dem entwickelten Programm umsetzen. Das diese Umsetzung gelingt, wird in dieser Diplomarbeit an den oben genannten Pyramidenzellen demonstriert.

2 Aufgabenstellung, Zielsetzung

Im Rahmen dieser Diplomarbeit ist ein Programm zu entwickeln, das die Erzeugung und Analyse anatomisch realistischer Neuron-Analoga ermöglicht. Das Programm soll den Aufbau von Dendriten, mittels modifizierter L-Systeme realisieren, so dass ein breites Spektrum von Verzweigungsstrukturen modelliert werden kann.

Als vereinheitlichte, formale Grundlage für die Beschreibung von Modellen neuronaler Dendriten können erweiterte parametrische Lindenmayer-Systeme (sog. Wachstumsgrammatiken) verwendet werden.

Das soll an zwei bekannten Dendritenmodellen demonstriert werden, die ursprünglich, ohne formale Grammatiken entwickelt wurden (Hillman- und Burke-Modelle).

Es müssen die speziellen Erweiterungen der L-Systeme bestimmt werden, die notwendig sind, um die charakteristischen Eigenschaften jedes dieser Modelle zu reproduzieren. Der erreichte Grad von Universalität und die Beschränkungen des Wachstums-Grammatik-Ansatzes, in Bezug auf Dendritenmodelle, sollen ein-

geschätzt werden.

Es ist ein Wachstumsgrammatikinterpreter zu entwickeln und für neuroanatomische Simulationen von Pyramidenzellen anzuwenden. Die Wachstumsparameter für diese Zellen müssen durch die Analyse von Zell-Rekonstruktionen von zwei Klassen von Pyramidenzellen, mittels eines speziell zu entwerfenden Programms, gewonnen werden.

Die Performanz der Programme ist mit der anderer, vorhandener Wachstumsgrammatikinterpreter (GROGRA, L-Neuron) zu vergleichen.

Die Konzeption und Implementierung ist in einer projektspezifischen Dokumentation festzuhalten. Die Programmdokumentation und der Programmcode sind der Diplomarbeit als Anlage beizufügen.

Teil I

Grundlagen

3 Repräsentation der Neuronen-Morphologie

3.1 Terminologie der Neuronen-Morphologie

Neuronen werden, angelehnt an das Turner Southampton Archive File, in dieser Diplomarbeit durch eine Menge von Stützstellen repräsentiert. Dabei sind Stützstellen Punkte im Raum, an dem das Neuron sich befindet. Da Neuronen in der Natur aus dem Soma, sowie einen oder mehreren Basalen-Dendriten, dem Axon und einem Apicalen-Dendriten bestehen, werden die Stützstellen nach Soma, Axon, basal und apical unterschieden. Eine andere Möglichkeit der Unterscheidung der Stützstellen ergibt sich aus der Töchterzahl, welche aus der Baumstruktur der Neuronen entsteht. Danach werden Wurzelknoten das Soma, Verzweigungsknoten, also Stützstellen mit mehr als einer Tochter, und Endpunkt unterschieden. Endpunkte sind dabei Stützstellen ohne Tochter. Um die Repräsentation einfach zu halten wird bei Verzweigungen immer von Binärverzweigungen ausgegangen. Für die in der Natur auftretenden Verzweigungen mit mehr als zwei Töchtern werden deshalb mehre Verzweigungen durch sehr kurze Zwischenstück verbunden.

Die im Neuronen-Baum vorhandenen Abschnitte zwischen zwei benachbarten Verzweigungsknoten werden Sektion genannt. Dabei wird zwischen terminalen und nicht terminalen Sektionen unterschieden. Bei terminalen Sektionen ist mindestens einer der begrenzenden Verzweigungsknoten ein Endpunkt, bei nicht terminalen Sektionen ist keiner ein Endpunkt. Sektionen können aus mehreren Segmenten bestehen. Diese ergeben sich aus den Stützpunkten zwischen den Verzweigungsknoten. Dabei erstreckt sich ein Segment von einem Stützpunkt zu einem benachbarten. In der Graphik 1 auf Seite 9 werden alle Begriffe noch einmal an einem Neuron dargestellt.

3.2 Repräsentation von Segmenten und Sektionen

Segmente werden durch gerade Kegelstümpfe dargestellt, welche durch eine Höhe und einen Anfangs- und Endradius definiert sind. Da Segmente von einem Stützpunkt zu einem benachbarten gehen erhält man den Radius am Anfang und Ende durch die Radien an den entsprechenden Stützstellen. Die Höhe des Kegelstumpfes entspricht dem euklidischen Abstand der Stützpunkte. Die Verjüngung nach Hillman entspricht dem Verhältnis der beiden Radien, wobei der dem Soma nähere Radius, den Soma entfernteren teilt $\frac{r_2}{r_1}$. Bei der Verjüngung nach Burke wird der Durchmesserunterschied durch die Länge des Segmentes dividiert $\frac{d_2-d_1}{h}$. Die Verjüngung wird im Programm als taper bezeichnet. Die Darstellung von Sektionen entspricht einer Reihe aufeinander gestapelter, Kegelstümpfe wobei der Endradius des einen dem Anfangsradius des darauf liegenden entspricht (siehe Abbildung 2 Seite 10). Die Radien und deren Reihenfolge ergeben sich

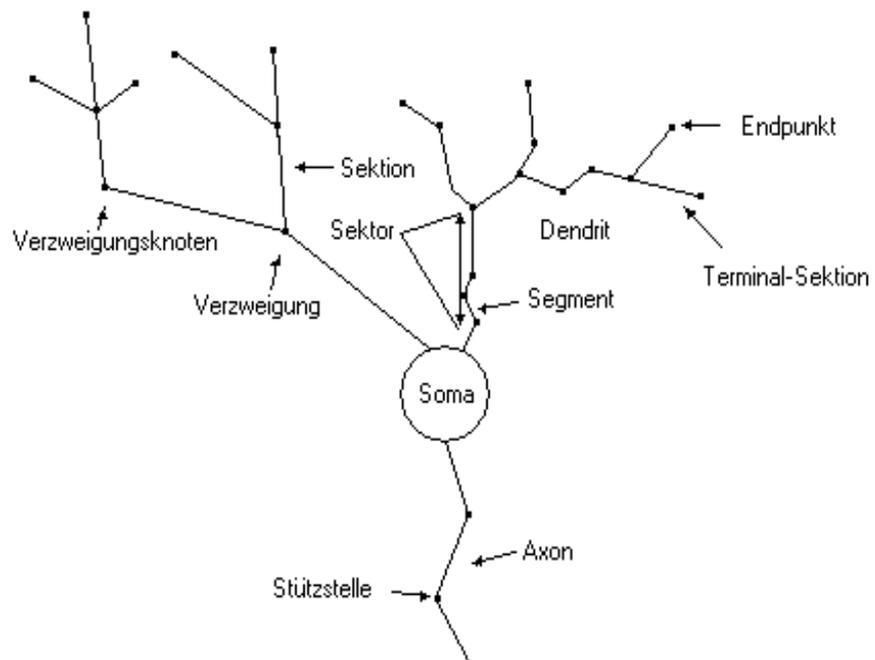


Abbildung 14: Hillman-Algorithmus

die Verzweigung mit dem Winkel von 20 Grad zwischen den Töchtern. "DR(dr) D1(d1) STAM → [+(10) Diameter(d1) START] [-(10) Diameter(mult(dr,d1)) START]". Um die gerade erzeugten Regeln einfach zu halten, wurde auf Eindeutigkeit nicht geachtet, so würde etwa aus "DendritTreeNumber(5)" durch die Regeln auch "Angle(120) DTN2(3) Angle(180) DTN2(2)" entstehen. Aus diesem Grund hier noch einmal die Regel eindeutig.

- 1 "S → A DendritTreeNumber(N_stems) A"
- 2 "A DendritTreeNumber(dtn) A → A Angle(div(360,dtn)) DTN2(dtn) A"
- 3 "A Angle (angle) DTN2 → +(angle) [DendritTree] A Angle(angle)"
- 4 "]" A Angle A →]"
- 5 "DendritTree → A Diameter(Sterm_Diamter) START"
- 6 "A Diameter(d) START → A Diameter(d) Test(<(d,Diam_threshold)) A"
- 7 "A Diameter(d) Test(0) A → A Diameter(d) Length(lbf_Branch_pathlength) STAM"
- 8 "A Diameter(d) Test(1) A → A Diameter(d) Length(Term_Branch_pathlength) TERM"
- 9 "A Diameter(d) Length(l) STAM → D(d) F(div(l,2.0)) A D(mult(d,Taper_2)) F(div(l,2.0)) STAM"
- 10 "A Diameter(d) Length(l) TERM → D(d) F(div(l,2.0)) D(mult(d,Taper_2)) F(div(l,2.0))"
- 11 "A D(d) F(f) STAM → D(d) F(f) A PK(PK) DR(Daughter_ratio) D(d) STAM"
- 12 "A PK(pk) DR(dr) D(d) STAM → A DR(dr) D1(mult(d,pow(div(pk,add(1,pow(dr,1.5))),div(2,3)))) STAM"
- 13 "A DR(dr) D1(d1) STAM → [+(10) A Diameter(d1) START] [-(10) A Diameter(mult(dr,d1)) START]"

Eine Definition der Regeln in der Skriptsprache des Programms findet sich in der Datei "Neuron.cfg" unter dem Hillman-Algorithmus.

13.1.2 Hillman-Algorithmus Anwendung

In diesem Abschnitt soll die Anwendung des Hillman-Algorithmus demonstriert werden. Dabei wird das Neuron "SE10.swc" analysiert und aus den ermittelten Werten, werden dann 10 Neuronen durch den Hillman-Algorithmus erzeugt, welche mit dem Original verglichen werden. Dabei werden sowohl die fundamentalen Parameter, wie auch emergente Parameter verglichen. In der nachfolgenden Tabelle sind einige Parameter aufgelistet. Bei der Erzeugung der 10 Neuronen werden alle Parameterverteilungen durch Gammaverteilungen mit den entsprechenden Erwartungswerten und Varianzen erzeugt. Wie gut der Hillman-Algorithmus bei der Rekonstruktion von Neuronen ist, zeigen die Tabellen 5 und 6 an fundamentalen und emergenten Parametern. Dabei wurden in der Tabelle 5 für jeden Fundamentalenwert zwei Zeilen benutzt die erste Zeile enthält die Werte des Original- Neurons die zweite die Werte der 10 erzeugten Neuronen

Wert	Wertanzahl	Erwartungswert	Varianz	Minimum	Maximum
lbf_Branch_pathlength	15	27.54497	359.2562	4.1950073	73.30497
	127	24.590805	328.36334	2.0322266	103.99902
Term_Branch_pathlength	24	49.037807	825.80646	16.05338	135.42491
	207	50.308895	780.59937	3.7578125	187.125
Taper_2	39	0.5497045	0.1419498	0.0162680	1.4878049
	334	0.62785757	0.114697754	0.0	1.7750549
Sterm_Diamter	8	1.2249999	0.2496	0.61	2.04
	80	1.2131538	0.29865736	0.22680664	3.0617676
N_stems	1	8.0	0.0	8.0	8.0
	10	8.0	0.0	8.0	8.0

Tabelle 5: fundamentale Parameter erste Zeile original Neuron
zweite Zeile erzeugten Neuronen

Wert	Wertanzahl	Erwartungswert	Varianz	Minimum	Maximum
Diam_threshold	24	0.4229166	0.0630623	0.2	1.23
	207	0.19894116	0.04943519	0.0	1.4818062
PK	16	1.4087691	0.3074904	0.81356543	2.924755
	137	1.4132118	1.6682566	0.0	8.69874
Daughter_ratio	16	0.6734682	0.0532178	0.2990200	1.0
	137	0.6141365	0.047587436	0.0	0.99246216

Tabelle 5: fundamentale Parameter erste Zeile original Neuron
zweite Zeile erzeugten Neuronen

Wertname	Originalwert	generierter Wert
Anzahl terminaler Sektionen	24.0	20.7
Anzahl nicht terminaler Sektionen	15.0	12.7
Anzahl von Verzweigungen	16.0	13.7
Anzahl Sektoren auf Baumtiefe 1	8.0	8.0
Anzahl Sektoren auf Baumtiefe 2	10.0	12.2
Anzahl Sektoren auf Baumtiefe 3	15.0	7.4
Anzahl Sektoren auf Baumtiefe 4	4.0	4.4
Anzahl Sektoren auf Baumtiefe 5	2.0	1.2
Radius Sektoren auf Baumtiefe 6	0.0	2.0
Radius Sektoren auf Baumtiefe 1	0.8425	0.7534
Radius Sektoren auf Baumtiefe 2	0.5920	0.3462
Radius Sektoren auf Baumtiefe 3	0.4053	0.2843
Radius Sektoren auf Baumtiefe 4	0.4575	0.2555
Radius Sektoren auf Baumtiefe 5	0.3050	0.1799
Radius Sektoren auf Baumtiefe 6	0	0.0727

Tabelle 6: emergente Parameter

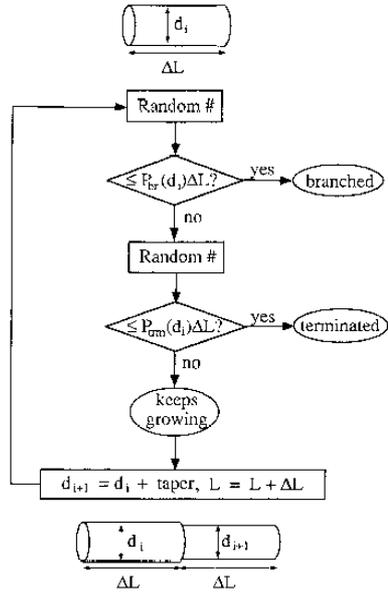
Wenn man die Messergebnisse zwischen dem Original und den erzeugten Neuronen vergleicht, schneidet die Realisierung des Algorithmus in den Fundamentalen Parameter gut ab, mit Fehler kleiner als 5%. Dies ist auch nicht verwunderlich, da die Verteilungen direkt in den Algorithmus eingingen. Betrachtet man die emergenten Analysewerte, so zeigt sich, dass auch diese recht gut reproduziert wurden. So sind zwar die Anzahlen an Sektionen und Verzweigungen bei den generierten Neuronen 15% geringer als beim Original, dies resultiert aber aus dem 10% kleineren Anfangsradien der Dendriten (siehe Radius Sektoren auf Baumtiefe 1).

Zum Schluss des Abschnittes noch ein Bild 15 des Original- Neurons, sowie drei Bilder 16 von erzeugten Neuronen.

13.2 Burke Algorithmus

Dieser von R. E. Burke, in der Arbeit [Burke92] im Jahr 1992, vorgestellte Algorithmus orientiert sich, wie der Hillman Algorithmus, ebenfalls an Fundamentalen Parametern. Wie bei Hillman werden Sektionen erzeugt, aber diesmal nicht über eine Verteilung der Längen von Sektionen, sondern über zwei Zufallszahlen, die das Anfangsstück einer Sektion zufällig wachsen oder verzweigen lassen. Wie auch bei dem Algorithmus von Hillman werden keine Winkel für Verzweigungen erzeugt. Zu den Parametern gehört die Verjüngung, welche aber durch $\frac{\text{Durchmesser}_{\text{Ende}} - \text{Durchmesser}_{\text{Anfang}}}{\text{Segment}_L \text{aenge}}$ bestimmt wird. Außerdem werden benötigt, Verteilungen über die Durchmesser und Anzahlen von Dendriten am Soma. Dazu kommt noch eine frei wählbare Konstante ΔL , die angibt, um wie viel ein Dendrit wächst, wenn weder terminiert noch verzweigt wird.

Branch Model Flow Chart



23 Daten und Inhalt der CD-ROM

Verzeichnis	Bemerkung
arbeit	Kopie der Diplomarbeit
arbeit/pdf	Alles für die Erzeugung der PDF-Datei
grogra/wgrogra	Windows Version der Software Grogra
grogra/xgrogra	Suse Linux Version der Software Grogra
grogra/dokumentation	Dokumentation der Software Grogra
lneuron	Windows Version der Software lneuron
lmeasure	Windows Version der Software lmeasure mit Java Interface
swc/se	Neuronen der genmanipulierten Mäuse
swc/wt	Neuronen der wilden Mäuse
swc/Burke Anwendung	Neuronen die vom erweiterten Burke- Algorithmus erzeugt wurden
swc/Hillman Anwendung	Neuronen die vom erweiterten Hillman- Algorithmus erzeugt wurden
swc/Burke Analyse	Neuronen von der Untersuchung des Burke- Algorithmus
swc/Hillman Analyse	Neuronen von der Untersuchung des Hillman- Algorithmus
software	Quellcode der erstellten Anwendung

24 Quellen

- [*GeDe*] Duncan E. Donohue, Ruggero Scorcioni, Gorgio A. Ascoli. Generation and Description of Neuronal Morphology Using L-Neuron
- [*LSys98*] Dr. Hansrudi Noser Prof. Dr. P. Stucki. Einführung in die Theorie der L-Systeme. Universität Zürich, Institut für Informatik, Juni 1998
- [*YoTa93*] Yoshihide Tamori. Theory of dendritic morphology. The American Physical Society, Oktober 1993
- [*SeWa*] Sandy Lobe. Sensitive Wachstumsgrammatiken
- [*LSys01*] Johannes Waldmann. L-System Vorlesungsskript, Institut für Informatik Universität Leipzig, Sommersemester 2001
- [*Zell95*] Th. Worsch, D. Merkle. Zellularautomaten Vorlesungsskript, Informatik für Ingenieure und Naturwissenschaftler Universität Karlsruhe (TH) Institut für Informatik Universität Leipzig, 1995-2003
- [*Bere04*] Gerhard Brewka. Berechenbarkeit Vorlesungsskript, Institut für Informatik Universität Leipzig, Sommersemester 2004
- [*AutoFS*] S. Gerber. Automaten und Formale Sprachen, Institut für Informatik Universität Leipzig, 2000
- [*InfHand2*] Rechenberg Pomberger. Informatik-Handbuch 2., und erweiterte Auflage, Carl Hanser Verlag München Wien, ISBN 3-446-19601-3, 1999
- [*TasInf*] Uwe Schneider, Dieter Werner. Taschenbuch der Informatik 4. Auflage, Fachbuchverlag Leipzig, ISBN 3446-21753-3, 2001
- [*Wahr80*] Dr. Gert Maibaum. Wahrscheinlichkeitsrechnung, Verlag Harri Deutsch, Thun und Frankfurt Main, ISBN 3-87144-166-3, 1980
- [*Wahr00*] Prof. Dr. Matthias Günther, Wahrscheinlichkeitsrechnung Stochastik Vorlesungsskript, Institut für Mathematik Universität Leipzig, 2000
- [*MatHand*] Dr. Üng. Hans-Jochen Bartsch. Taschenbuch mathematischer Formeln 18. verbesserte Auflage, Fachbuchverlag Leipzig, ISBN 3-446-19396-0, 1998
- [*NeBra04*] Orit Stefi, Amir Harel, Dimitri B, Chklovskii, Eshel Ben-Jacob, Amir-Ayali. Biophysical constraints on neuronal branching. Department of Zoology, Tel-Aviv University, Tel-Aviv 69978, Israel. School of Physics and Astronomy, Tel-Aviv University, Tel-Aviv 69978, Israel. Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, USA, 2004
- [*Murray10*] C.D. Murray. A relationship between circumference and weight in trees and its bearing on branching angles, *J. Gen Physiol* 10 (1926) 725
- [*Burke92*] R. E. Burke, W. B. Marks und B. Ulfhake. A Parsimonious Description of Motoneuron Dendritic Morphology using Computer Simulation, *The Journal of Neuroscience*, June 1992, 12(6): 2403-2416
- [*FrakW*] Peter Jossen, Daniel Eyer. Wachstumssimulation <http://www.fraktalwelt.de/lsys/lsysinhalt.html>. Aktualisierungsdatum: 01.04.2001
- [*SWCDef*] Attila I. Gulyas. Duke-Southampton archive of neuronal morphology format (.swc) description <http://www.koki.hu/~gulyas/ca1cells/dsaformat.htm> Institute of Experimental Medicine Hungarian Academy of Sciences Budapest, Ungarn