

OSVALDO VARGAS JAQUES

SÍNTESE DE ESTRUTURAS NEURAIIS USANDO GRAMÁTICAS VETORIAIS ESTOCÁSTICAS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática pelo Curso de Pós-Graduação em Informática, do Setor de Ciências Exatas da Universidade Federal do Paraná, em convênio com o Departamento de Informática da Universidade Estadual de Maringá.

Orientadores: Prof. Dr. Maurício Fernandes
Figueiredo
Prof.^a Dr.^a Regina Célia Coelho

CURITIBA

2003



Ministério da Educação
Universidade Federal do Paraná
Mestrado em Informática



PARECER

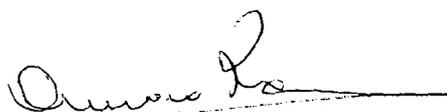
Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, do aluno Osvaldo Vargas Jaques, avaliamos o trabalho intitulado, "Síntese de Estruturas Neurais Usando Gramáticas Vetoriais Estocásticas", cuja defesa foi realizada no dia 22 de agosto de 2003, às quatorze horas, no Anfiteatro A do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação do candidato. (Convênio número 279-00/UFPR de Pós-Graduação entre a UFPR e a UEM - ref. UEM número 1331/2000-UEM).

Curitiba, 22 de agosto de 2003.


Prof.^a Dra. Regina Célia Coelho
FCMNTI/UNIMEP – Orientadora


Prof. Dr. Maurício Fernandes Figueiredo
DIN/UEM - Orientador


Prof.^a Dra. Júlia Sawaki Tanaka
IQ/UNESP – Membro Externo


Prof.^a Dra. Aurora Trinidad Ramirez Pozo
DINF/UFPR – Membro Interno

Este trabalho é dedicado aos meus pais, Floriano e Raquel, que durante estes últimos dois anos souberam tolerar a ausência do filho. E à sabedoria de camponês passada pelo meu pai. "Seu Jacques", tenho muito orgulho de herdar seu nome.

AGRADECIMENTOS

Muitas pessoas passam por nossas vidas neste período de mestrado, passando-nos experiências e vivências, assim como também admiração.

Toda minha consciência como animal racional tentando se tornar humano, deve-se ao Altíssimo, pela sua infinita sabedoria, amor e severidade. Quanto mais espiritualizo a matéria, mais materializo minha convicção em Ti, Grande Arquiteto do Universo.

Meus agradecimentos e admiração à minha orientadora professora Dra. Regina Célia Coelho, pela severidade e conhecimento. Seu método de orientação servirá de guia para meus futuros orientandos (já percebi que funciona). Minhas desculpas pelo cansaço destas viagens e trabalho.

Meus agradecimentos ao professor e orientador Dr. Maurício Fernandes Figueiredo, por suas sugestões, lógica nebulosa é realmente nebuloso.

Agradeço ao professor Dr. Luís Augusto Consularo, por sempre estar disponível para esclarecimento de qualquer dúvida. Seu nível de conhecimento me traz admiração, cujos degraus eu desejo galgar um dia.

Agradeço ao professor Dr. Ronaldo, pelo seu aspecto comunicativo e amigo. Minha admiração pelo conhecimento matemático teórico computacional do professor Dr. Xavier.

E sem deixar de esquecer ao 'bombeiro' Marcos Paulo Moro, coordenador do curso de Computação da UNIGRAN.

Meu afeto aos demais colegas, Gláucia, Radames, Fabrício ('Na boa'), Edmundo ('Animal'), Ligia ('Lili Bolero'), Nicholas ('véio Nico'), ...etc.

Aos amigos da campanha à reitoria da UEMS, grato pela compreensão. Não vencemos, mas mostramos que somos um grupo forte. 'Vai melhorar'.

Por último minha gratidão à UNIGRAN – Centro Universitário da Grande Dourados pela ajuda de custo e confiança.

SUMÁRIO

AGRADECIMENTOS	4
LISTA DE FIGURAS	viii
LISTA DE TABELAS	x
LISTA DE ABREVIATURAS	xi
RESUMO	xii
ABSTRACT	xiii
1 INTRODUÇÃO	1
1.1 OBJETIVO	2
1.2 JUSTIFICATIVA	3
1.3 ORGANIZAÇÃO DO TRABALHO	4
2 SISTEMA VISUAL E NEURÔNIOS	5
2.1 SISTEMA VISUAL	6
2.1.1 <i>Funções receptoras da retina</i>	6
2.1.2 <i>Fluxo da informação visual e processamento</i>	8
2.1.3 <i>Funções das células neurais da retina</i>	9
2.1.3.1 Células ganglionares	9
2.1.3.2 Células amácrimas	9
2.1.3.3 Células bipolares	10
2.1.3.4 Células horizontais	10
2.1.3.5 Células fotorreceptoras	10
2.2 CÓRTEX VISUAL	11
2.2.1 <i>Estrutura Neural</i>	13
2.2.2 <i>Camadas do córtex</i>	14
2.3 NEURÔNIOS	16
2.3.1 <i>Partes da célula neural</i>	17
2.3.2 <i>Botões sinápticos</i>	17
2.3.3 <i>Tipos de células neurais</i>	18
2.3.4 <i>Cones de crescimento e fatores de crescimento</i>	19
2.3.5 <i>Neuroplasticidade</i>	20
2.3.5.1 <i>Habituação</i>	21
2.3.5.2 <i>Aprendizado</i>	21
2.3.5.3 <i>Lesão</i>	21
2.3.6 <i>Forma e função neural</i>	22
2.4 FATORES NEUROTROFICOS	23
3 MÉTODOS E TÉCNICAS RELACIONADOS À SÍNTESE NEURAL	25
3.1 PROBABILIDADE E ESTATÍSTICA	26
3.1.1 <i>Classes e intervalos de classes</i>	26
3.1.2 <i>Eventos e probabilidade</i>	27
3.2 FUNÇÃO DE DISTRIBUIÇÃO	27
3.3 REGRA DE BAYES	28

3.4	ALFABETOS E LINGUAGENS	29
3.5	AUTÔMATO FINITO	31
3.5.1	<i>Analisador Léxico como Autômato Finito</i>	31
3.6	GRAMÁTICAS	34
3.6.1	<i>Hierarquia de Chomsky</i>	36
3.6.2	<i>Gramáticas Regulares e Autômatos Finitos</i>	37
3.6.3	<i>Gramáticas estocásticas</i>	38
3.6.4	<i>Análise sintática utilizando gramáticas</i>	39
3.7	GRAMÁTICAS GRÁFICAS	42
3.7.1	<i>Gramáticas L-System</i>	42
3.7.2	<i>L-System Estocástica</i>	45
3.8	ALGUMAS NOTAÇÕES DE TRAÇADO NEURAL	48
3.8.1	<i>Formato EUTECTIC</i>	48
3.8.2	<i>Notação SWC</i>	49
3.8.3	<i>Formato NEUROLUCIDA</i>	51
3.8.4	<i>Formato DAT</i>	53
3.9	ALGUNS ALGORITMOS PARA GERAÇÃO DE CÉLULAS NEURAIIS	54
4	O SIMULADOR E VISUALIZADOR NEURAL	57
4.1	O INTERPRETADOR GRAMATICAL	57
4.2	PADRONIZAÇÃO DE FORMATO	65
4.3	VISUALIZADOR TRIDIMENSIONAL	65
4.4	NOTAÇÃO VETORIAL	66
4.4.1	<i>Sistema de coordenadas locais</i>	68
4.5	SIMULAÇÃO DE MONTE CARLO USANDO FUNÇÕES DE DISTRIBUIÇÃO	72
4.6	FUNÇÕES DE BASE RADIAL	73
4.6.1	<i>Thin plate splines</i>	75
4.7	SÍNTESE DE CÉLULAS NEURAIIS UTILIZANDO FUNÇÕES ESTOCÁSTICAS	75
5	SÍNTESE NEURAL	78
5.1	FUNÇÃO DE DISTRIBUIÇÃO CONDICIONAL	79
5.2	DEFINIÇÃO DA GRAMÁTICA	80
5.3	GERAÇÃO NEURAL	82
5.4	DESENVOLVIMENTO DO SOFTWARE	85
5.5	VALIDAÇÕES	89
5.5.1	<i>Tamanho de segmento</i>	89
5.5.2	<i>Comprimento de arco</i>	90
5.5.3	<i>Curvatura</i>	91
5.5.4	<i>Torção</i>	92
6	CONSIDERAÇÕES FINAIS	93
6.1	CONTRIBUIÇÕES	93
6.2	TRABALHOS FUTUROS	94
	REFERÊNCIAS BIBLIOGRÁFICAS	96
	APENDICE A	99
	CÉLULAS NATURAIS UTILIZADAS PARA EXTRAÇÃO DAS MEDIDAS	99

APÊNDICE B.....	117
GRÁFICOS DE MEDIDAS DE CÉLULAS PIRAMIDAIAS DO CORTEX VISUAL DE RATO.....	117
APÊNDICE C.....	119
FDS DE DISTRIBUIÇÃO CONDICIONAIS DE CÉLULAS DO CÓRTEX VISUAL DE RATO.....	119
C.1 SEGMENTO DENDRÍTICO APICAL.....	119
C.2 SEGMENTO DENDRÍTICO BASAL.....	121
C.3 ÂNGULO DE TORÇÃO BASAL E APICAL.....	122
APÊNDICE D.....	123
GRÁFICOS DE MEDIDAS DE CÉLULAS PIRAMIDAIAS ARTIFICIAIS.....	123
APÊNDICE E 125	
FDS DE DISTRIBUIÇÃO CONDICIONAIS DE CÉLULAS ARTIFICIAIS.....	125
E.1 SEGMENTO DENDRÍTICO APICAL.....	125
E.2 SEGMENTO DENDRÍTICO BASAL.....	126
E.3 ÂNGULO DE TORÇÃO BASAL E APICAL.....	127

LISTA DE FIGURAS

Figura 2.1 – O olho e suas diferentes camadas neurais.....	7
Figura 2.2 – Mácula e fóvea ao centro.....	8
Figura 2.3 – Campos receptivos <i>on-center</i> e <i>off-center</i>	9
Figura 2.4 – Fotorreceptores.....	11
Figura 2.5 – Áreas do cérebro.....	11
Figura 2.6 – Tipos de neurônios no córtex, marcado pelo método de Golgi.....	12
Figura 2.7 – Caminho Visual.....	13
Figura 2.8 – Neurônios piramidal e estelar da esquerda para a direita.....	14
Figura 2.9 – Neurônios com e sem espículas.....	14
Figura 2.10 – Camadas do córtex visual nos primatas, usando diferentes técnicas de marcação.....	15
Figura 2.11 – Distribuição das células piramidais nas camadas do córtex visual.....	15
Figura 2.12 – Projeção do núcleo geniculado lateral na camada 4C.....	16
Figura 2.13 – Partes de um neurônio e detalhes de uma sinapse.....	18
Figura 2.14 – Alterações neuromórficas após ocorrência de lesão.....	22
Figura 2.15 – Ação de fatores extra celulares influenciando a morfologia cortical.....	24
Figura 3.1 – Redefinição de espaço amostral.....	29
Figura 3.2 – Diagrama de transição para reconhecimento de operadores relacionais.....	32
Figura 3.3 – Uma máquina de estados finitos.....	33
Figura 3.4 – Árvore gramatical.....	34
Figura 3.5 – Diagrama de Transição.....	37
Figura 3.6 – Ramificação gerada após quatro iterações da gramática 3.....	44
Figura 3.7 – Árvores geradas utilizando a gramática 4.....	45
Figura 3.8 – Um ramalhete gerado pela gramática 5 de PRUSINKIEWICZ (1986).....	47
Figura 3.9 – Ramificações no formato EUTECTIC.....	48
Figura 3.10 – Trecho de arquivo em notação EUTECTIC.....	49
Figura 3.11 – Trecho de um arquivo SWC.....	51
Figura 3.12 – Trecho de arquivo com notação NEUROLUCIDA.....	53
Figura 3.13 – Trecho de um arquivo de traçado DAT.....	54
Figura 4.1 – Árvore gerada pelo interpretador utilizando gramática.....	58
Figura 4.2 – Um ponto P sofrendo influência de vários pontos de atração.....	59
Figura 4.3 – Tropismo em locais diferentes para a mesma gramática.....	61
Figura 4.4 – Alguns neurônios artificiais bidimensionais gerados por gramática.....	62
Figura 4.5 – Diagrama de transição de estados para uma L-System estocástica.....	63
Figura 4.6 – Visualizador tridimensional.....	66
Figura 4.7 – Ângulos de torção θ e curvatura ϕ	67
Figura 4.8 – Esquema para início da notação vetorial.....	67
Figura 4.9 – Sistema de coordenadas locais.....	69
Figura 4.10 – Neurônio com notação eut e vet.....	71
Figura 4.11 – Trecho de um arquivo em notação VET.....	72
Figura 4.12 – Funções de distribuição de medidas de neurônios piramidais do córtex de ratos.....	72

Figura 5.1 – Esquema para construção de FD dominantes e dependentes.....	79
Figura 5.2 – Correção da orientação gramatical.....	84
Figura 5.3 – Diagrama de fluxo de dados do software desenvolvido.....	85
Figura 5.4 – Diagrama hierárquico funcional do software desenvolvido.....	86
Figura 5.5 – Visualizador neural com uma célula virtual.....	87
Figura 5.6 – Porcentagem de crescimento.....	88
Figura 5.7 – Algumas células artificiais geradas.....	88
Figura 5.8 – Gráficos para medidas de segmento.....	90
Figura 5.9 – Gráficos para medidas de arco.....	91
Figura 5.10 – Gráficos para curvatura de arco.....	91
Figura 5.11– Gráficos para torção de arco.....	92

LISTA DE TABELAS

Tabela 3.1 – Tipos de pontos no formato SWC.....	50
Tabela 3.2 – Linhas de código NEUROLUCIDA	52

LISTA DE ABREVIATURAS

AF	Autômato Finito
ASC	Extensão de arquivo de traçados de células formato neurolucida
DAT	Extensão de arquivo de traçados de células formato dat
DLL	Biblioteca de Vínculo Dinâmico (<i>Dinamic Link Library</i>)
EUT	Extensão de arquivo de traçados de células formato eutectic
FD	Função de Distribuição
GLC	Gramática Livre de Contexto
GNR	Gramática Não Restritiva
GR	Gramática Regular
GSC	Gramática Sensível ao Contexto
LS	Gramática L-System
SWC	Extensão de arquivo de traçados de células formato SWC
TPS	Thin Plate Spline
VET	Extensão de arquivo de traçados de células formato vetorial

RESUMO

O estudo de modelos representativos dos aspectos morfológicos das células neurais ainda é muito pequeno se tentarmos comparar com modelos neurofisiológicos. Embora o pioneiro da neurociência Ramón y Cajal tenha observado que quanto mais complexa a função da célula neural, mais complexa é sua morfologia, somente agora alguns poucos estão se aventurando nesta odisséia, com pesquisas tanto no campo biológico como no computacional buscando modelos que possam contribuir para uma melhor compreensão da relação forma e função neural.

Iniciamos esta dissertação tratando do sistema visual e seus aspectos morfológicos e funcionais. Fazemos uma introdução ao funcionamento neural e seu desenvolvimento. Buscamos também, na literatura, informações relacionadas às mudanças morfológicas que ocorrem durante o desenvolvimento e maturação (plasticidade) dos neurônios piramidais do córtex visual. Descrevemos os métodos e técnicas existentes relacionados à síntese neural, tais como gramáticas gráficas estocásticas, medidas neuromorfométricas e formatos de traçado neural. Desenvolvemos procedimentos para conversão dos diferentes traçados em um formato padrão, e assim, criamos funções estocásticas extraídas das diferentes medidas de células naturais.

A seguir, definimos uma gramática gráfica para a síntese de células artificiais. À cada regra desta gramática incorporamos funções extraídas de medidas de células naturais dos neurônios piramidais. Deste modo, procuramos criar células artificiais tridimensionais morfologicamente realistas. Para isso, desenvolvemos um compilador e interpretador gramatical (parser) que gera a seqüência de pontos para a geração da célula. Em seguida, criamos o visualizador tridimensional que permite visualizar esta célula, realizando operações tais como rotação, translação, escalonamento, e simulação de crescimento dendrítico em paralelo, além de outras operações utilizadas em ferramentas de visualização 3D.

ABSTRACT

The study of the representative models of morphologic features of neural cells is still very restrict if we attempt to compare it to the neurophysiological models available. Although the neuroscience pioneer Ramón y Cajal observed that as more complex is the function of a neural cell, more complex will be its morphology, only now a few people have engaged this odyssey, with researches both in the biological and computational fields, in the search for models capable of contributing for a better comprehension of the function-form correlation.

We start this work making reference to the visual system and its morphological and functional aspects. Neural function and development were introduced. We also searched in the literature for information related to the morphologic changes that occur during the development and maturation of the pyramidal neurons (plasticity) of the visual cortex. We described the existing methods and techniques related to the neural synthesis, such as stochastic graph grammars, neuromorphometric measurements and neural trace formats. We developed procedures for the conversion of the different trace patterns into a default pattern, and so we created stochastic functions extracted from different measurements of natural cells.

After that, we defined a graph grammar for the synthesis of artificial cells. Into each rule of this grammar we incorporated functions extracted from measurements of natural cells of pyramidal neurons. Therefore, we tried to create realistic morphologically three-dimensional virtual cells. For this purpose we developed a grammatical compiler and interpreter (parser), which generates a sequence of points for cell generation. Afterward, we developed a three-dimensional visualizer, which allows us to see the cell through operations such as rotation, translation, scaling, and simulation of parallel dendritic growth, without to mention some others 3D visualization tools used.

1 INTRODUÇÃO

O cérebro é composto basicamente de neurônios e células gliais. As células gliais são responsáveis pela sustentação e proteção do processo de comunicação neural. Existem praticamente 10^{12} células no cérebro. Embora seja um número assombroso, não é a quantidade de células que garante sua complexidade, mas o modo como estas se distribuem e se conectam entre si. Uma única célula nervosa do cérebro pode receber informações de centenas a milhares de células, sendo que o número total de interconexões pode chegar à ordem de 10^{14} a 10^{15} . É inegável a impossibilidade atual de criar um modelo representativo para todas estas conexões. Contudo, esta quantidade imensa de conexões ocorre devido à variabilidade morfológica da estrutura neural, que possibilita, com isso, tanto a recepção de informações provindas de milhares de células, como o envio de informações para outras centenas. Como outra evidência, temos as pesquisas realizadas em neurofisiologia que possibilitam, através da emissão de eletrodos, o conhecimento da função neural. Estas funções variam de acordo com o formato da célula neural, possibilitando, assim, relacionar forma com função.

A modelagem e simulação neural dos muitos trabalhos desenvolvidos desprezam parâmetros que impedem uma melhor aproximação com a morfologia das células naturais. Os artigos de PRUSINKIEWICZ(1986), HAMILTON(1993) e McCORMICK e MULCHANDANI(1994) apresentam sensíveis contribuições na área, em especial o artigo de McCORMICK e MULCHANDANI(1994). COELHO(1988) utiliza funções de distribuição extraídas de células naturais para gerar neurônios artificiais morfolologicamente realísticos, levando em conta medidas como comprimento e ângulo de bifurcação das ramificações dendríticas e distribuição espacial das células neurais nas estruturas, entre outras. COELHO(1988, 2002) cita ainda a possibilidade de geração de células neurais utilizando gramáticas vetoriais estocásticas, simulando com isso o crescimento em paralelo destas células.

Finalmente, ASCOLI e KRICHMAR(2000) elaboram o software L-Neuron (LN) para a descrição geométrica e topológica das ramificações neurais, utilizando gramáticas L-

System (LS) estocástica. Este software utiliza distribuições estatísticas para gerar células artificiais morfologicamente distintas.

Desenvolvemos neste trabalho um *software* que possibilite a simulação do crescimento em paralelo de células neurais, e as alterações morfológicas sofridas por estas diante de campos de influência. Utilizamos os conceitos apresentados por COELHO(1998, 2002), ASCOLI e KRICHMAR(2000) e as contribuições de McCORMICK e MULCHANDANI(1994), além de outras abordagens. Isso possibilitará uma melhor compreensão da evolução da atividade neural e suas relações com campos de atração, tamanho, forma e função.

1.1 OBJETIVO

A modelagem neural tem um papel fundamental no estudo de características do sistema nervoso tanto quanto experimentos e teorias sobre estruturas neurais. O objetivo deste trabalho é a geração de células neurais tridimensionais que apresentem formas semelhantes às células naturais. Para isso, integramos as abordagens de Mc CORMICK e MULCHANDANI (1994), ASCOLI e KRICHMAR (2000) e COELHO(1988, 2000) e incluímos novas abordagens. Sendo assim, utilizamos:

- LS estocástica para geração das células;
- informações de coordenadas locais para a construção das ramificações tridimensionais;
- funções de distribuição para incorporar nas gramáticas medidas relacionadas à forma das células naturais;
- funções de distribuição condicionais, isto é, o estado atual do crescimento do axônio ou dendrito depende do histórico do estado anterior.

Incluímos as informações sobre o histórico não apenas de uma medida, mas o histórico da interação entre estas medidas. Procuramos transformar algumas destas funções de distribuição (FD) usando funções de interpolação de bases radiais. Desta forma, para cada

- 1) Desenvolvemos um conversor de traçados neurais. Uma vez que existem diferentes notações de traçado neural, tínhamos de desenvolver um conversor destas notações para uma notação própria e adequada aos nossos propósitos;
- 2) Desenvolvemos FD para cada medida natural. Uma vez obtida a notação adequada extraímos as medidas estatísticas destas células neurais e, a seguir, criamos FD para algumas destas medidas;
- 3) Desenvolvemos um compilador e interpretador (parser) de LS estocástica. O programa deve ler a gramática e verificar se não ocorreu nenhum erro sintático. A seguir, o interpretador executa as instruções gráficas relacionadas pela gramática usando as funções de distribuição e as funções polinomiais relacionadas às medidas morfológicas associada a cada passo da instrução. Podemos, assim, adicionar funções de visualização tridimensional apropriada e simular o crescimento em paralelo de vários neurônios.

1.2 JUSTIFICATIVA

Poucos modelos de simulação neural consideram os aspectos morfológicos do comportamento neural. O levantamento dos estudos existentes e a implementação das técnicas propostas possibilitarão a criação de novas propostas e trabalhos. A divulgação destes conhecimentos permitirá a criação e geração de novas aplicações voltadas para o aspecto morfológico.

A ferramenta desenvolvida possibilita :

- o crescimento em paralelo das ramificações das células;
- as alterações morfológicas sofridas pelas células ao se depararem com campos de influências;
- visualização das células tridimensionalmente e o uso das operações típicas do visualizador gráfico para estas formas neurais.

Além da simulação, o projeto desenvolvido possibilita a visualização de células neurais em diferentes traçados. O programa ainda oferece a transformação das notações de um traçado para outro, auxiliando o pesquisador.

1.3 ORGANIZAÇÃO DO TRABALHO

Neste trabalho, fizemos inicialmente uma revisão dos conceitos necessários para a compreensão e desenvolvimento da proposta. No capítulo 2 tratamos do sistema visual, incluindo o córtex visual primário, por ser o local de onde foram extraídas as células utilizadas neste trabalho. Apresentamos as principais células existentes nesta região, seu funcionamento e formato. Neste mesmo capítulo tratamos dos neurônios, sua estrutura e classificação, destacando a relevância de forma e função neural.

No capítulo 3 apresentamos alguns conceitos de estatística, teoria da computação e compiladores necessários para a compreensão e desenvolvimento do projeto. Incluímos também algumas notações de traçado e os algoritmos utilizados no software LN.

No capítulo 4 apresentamos as implementações e notações relacionadas ao traçado neural necessárias para realização do projeto, e os resultados obtidos.

Finalmente, no capítulo 5, apresentamos a conclusão do trabalho e sugestões de trabalhos futuros.

2 SISTEMA VISUAL E NEURÔNIOS

A inteligência humana é consideravelmente influenciada pelo sentido da visão, uma vez que cerca de dois terços das informações que chegam ao cérebro provem do sistema visual. Deste modo, o homem busca desde muito tempo entender a capacidade de ver.

O entendimento do funcionamento da visão tem auxiliado no avanço de algumas tecnologias nas áreas de visão computacional e inteligência artificial. A visão computacional busca compreender o funcionamento visual, criando com isso modelos biológicos que auxiliem neste entendimento. A execução e simulação de tais modelos possibilitam a criação de sistemas artificiais que imitam a visão.

À medida que os organismos evoluem em complexidade, mais preciso e complexo fica o mecanismo de reconhecimento dos sinais visuais, aumentando, por consequência, a variabilidade de células nervosas visuais com diferentes funções e formatos. A luz provoca alterações químicas nestas células e estas enviam sinais ao córtex visual do cérebro, que interpreta estes sinais obtendo então um modelo do estímulo recebido. Portanto, a compreensão do mecanismo da visão envolve a compreensão do processo de transmissão de sinais das células nervosas, apesar de ainda não sabermos direito como estas informações são codificadas, processadas e transmitidas ao sistema nervoso. Atualmente, o desenvolvimento de modelos matemáticos e computacionais auxiliam na validação e complementação de nosso conhecimento a respeito das estruturas neurais (COSTA; OLIVEIRA e MINGHIN, 2000; COELHO, 1998).

A maior parte das pesquisas em simulações relacionadas à neurociência trata dos aspectos fisiológicos dos neurônios, dando pouca importância aos aspectos morfológicos, embora pesquisas anteriores mostram que a função neural está estritamente relacionada à forma. Observando o crescimento neural em espécimes com diferentes níveis de evolução, pode-se perceber que quanto mais complexo o animal, mais complexa é sua estrutura neural. Dentro de um mesmo animal, neurônios motores diferem de neurônios do córtex em complexidade e estrutura.

Considerando estes aspectos, abordaremos neste capítulo o caminho percorrido pelos sinais visuais desde a retina até o córtex visual, passando pelos diferentes tipos de células

neurais existentes neste caminho. Deste modo, apresentamos na seção 1 uma revisão sobre o funcionamento do sistema visual e de seus componentes nervosos. A seguir, na seção 2, veremos o córtex visual e suas células nervosas, tais como os neurônios piramidais e estelares, objeto de interesse desta dissertação quanto ao aspecto morfológico. Na seção 3, conceituamos neurônios, seus aspectos estruturais, funcionais e morfológicos. Nesta mesma seção tratamos dos processos que influenciam a alteração morfológica, como cones de crescimento e plasticidade. Finalmente, concluímos destacando a relevância entre forma e função neural.

2.1 SISTEMA VISUAL

O sistema visual tem a função de reconhecimento, localização de objetos e controle dos movimentos oculares, além de prover informação para o controle da postura e movimento dos membros (LUNDI-EKMAN, 2000).

O olho pode ser comparado à "uma máquina fotográfica comum, sendo composto de lentes, um sistema de diafragma variável e uma retina, que corresponderia ao filme" (GUYTON, 1998).

A retina tem a função de perceber as variações de luz através das células receptoras chamadas cones e bastonetes. Os cones são responsáveis pela diferenciação das cores, e os bastonetes percebem as variações de claro e escuro. Quando estas células são excitadas, transmitem sinais sucessivos que percorrem a retina passando então para o nervo óptico até chegar ao córtex cerebral.

2.1.1 Funções receptoras da retina

A luz, após passar pela retina e pelo humor líquido, segue para as células ganglionares, além de outras células das camadas internas, até chegar aos cones e bastonetes. A retina é composta de uma grande quantidade de substâncias químicas, e estas substâncias provocam junto com a luz uma reação química fazendo com que os cones e bastonetes transformem a luz em sinal elétrico.

A retina está situada na parte interna e posterior do olho e é composta de células nervosas distribuídas em camadas, seguindo a seqüência do caminho da luz citado no parágrafo anterior (Figura 2.1 (a)). Estas camadas estão distribuídas do seguinte modo: (1)

células ganglionares, (2) células amácrimas, (3) células bipolares, (4) células horizontais, (5) e células fotorreceptoras, os cones e bastonetes. A luz, ao chegar na retina, passa pelas células ganglionares, amácrimas, bipolares e horizontais, até finalmente chegar aos cones e bastonetes, passando então ao nervo óptico e seguindo para o encéfalo (Figura 2.1(b)). Cada retina é composta aproximadamente por cerca de 125 milhões de bastonetes, 5,5 milhões de cones e 900 mil fibras ópticas do nervo óptico. Assim, uma média de 125 bastonetes e seis cones convergem sobre cada fibra óptica (GUYTON, 1998). Ocorre então uma convergência 125 para 1, o que resulta em perda de acuidade. No entanto, na região central da retina, chamada fóvea, há uma convergência menor, permitindo uma melhor acuidade visual (HUBEL, 1995).

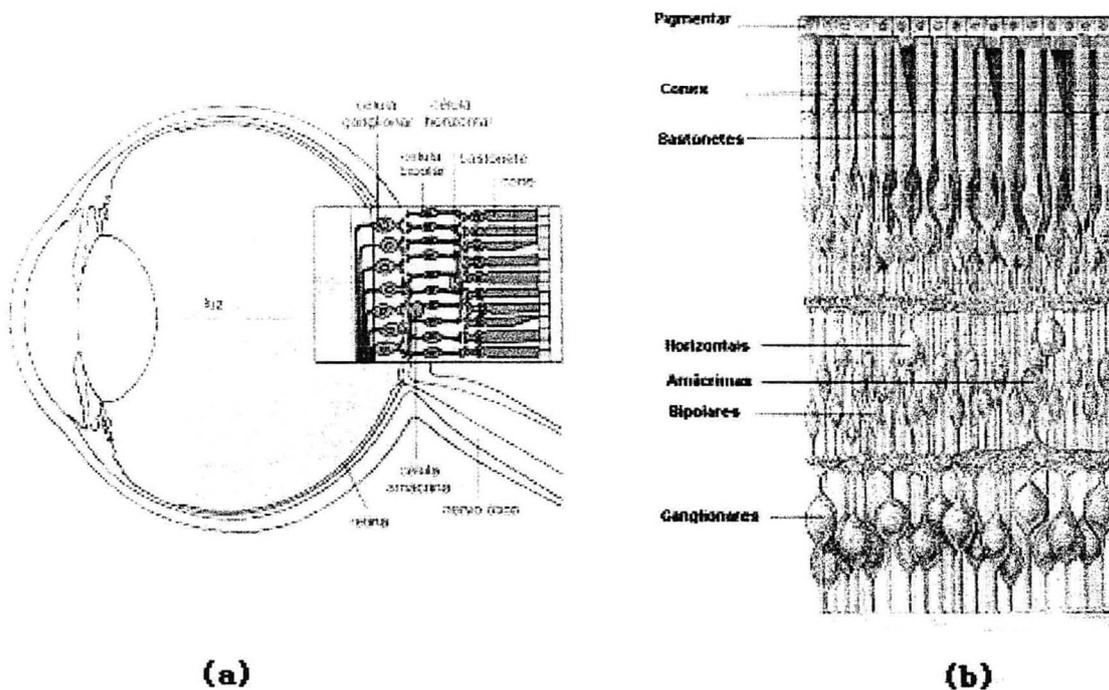


Figura 2.1 – O olho e suas diferentes camadas neurais. (a) surpreendentemente, a luz tem que passar pelas células ganglionares e bipolares antes de atingir os cones e bastonetes, e (b) as camadas neurais ampliadas. Extraída de HUBEL(1995), modificada e traduzida.

A fóvea é parte de uma pequena região da retina chamada de mácula (Figura 2.2), ocupando uma área menor do que 1 mm^2 . Esta área é responsável pela visão nítida e detalhada e é composta inteiramente por cones alongados e com diâmetro de 1,5 micron, enquanto os

cones localizados fora da região central são muito mais espessos. Na fóvea, as células ganglionares, os vasos sanguíneos e as camadas mais internas da retina estão distribuídos mais para o lado ao invés de situarem no topo dos cones, diminuindo, assim, a interferência na passagem da luz. Desse modo, a luz chega aos cones sem atravessar várias camadas, permitindo uma melhor acuidade visual (GUYTON, 1998). O caminho que a luz percorre sem atravessar várias camadas é chamado de caminho direto, enquanto o percurso que atravessa várias camadas é chamado de caminho indireto. O caminho indireto é mais difuso devido as várias conexões laterais. Na fóvea, uma única célula ganglionar está conectada a uma única célula bipolar e esta a um único cone. O processo muda a medida que nos deslocamos do centro para a periferia, onde há uma maior convergência de informações visuais (HUBEL, 1995).

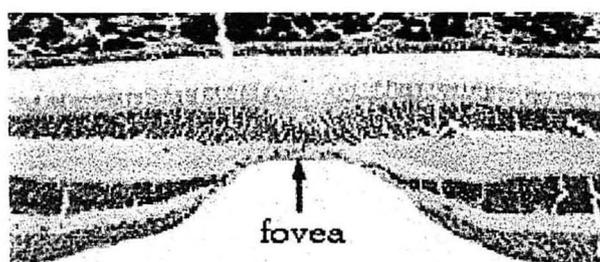


Figura 2.2 – Mácula e fóvea ao centro¹

2.1.2 Fluxo da informação visual e processamento

Como visto anteriormente, as células da retina convertem luz em sinais neurais. Após a saída da retina este sinal é conduzido pelo nervo óptico, pelo trato óptico com sinapses no núcleo geniculado lateral do tálamo. O nervo óptico é um conjunto de fibras que conduz informações desde a retina até o quiasma óptico (quiasma corresponde a letra X em grego, como pode ser visto na Figura 2.7). Aqui todas as fibras das metades nasais das duas retinas cruzam-se, formando o trato óptico, conduzindo informações do quiasma óptico até o núcleo geniculado lateral (GUYTON, 1998; LUNDY-EKMAN, 2000). Assim, as informações do campo visual esquerdo chegam ao córtex visual através do núcleo geniculado lateral do lado oposto. "Nestes núcleos ocorrem as primeiras sinapses no caminho das informações da retina para o córtex" (TANAKA, 2000).

¹ : <http://www.4colorvision.com/neuron/fundneuron.htm>

2.1.3 Funções das células neurais da retina

Conforme a Figura 2.1 (b), estas células distribuem-se de modo a possibilitar o envio e troca de informações entre si. Cada grupo destas células possuem funções específicas para a recepção e percepção de sinais externos.

2.1.3.1 Células ganglionares

As células ganglionares, mesmo quando não estimuladas, transmitem impulsos nervosos contínuos numa frequência de cinco por segundo, podendo tanto ser um sinal inibitório como excitatório (GUYTON, 1998). Com relação à luminosidade podem ser classificadas como *on-center* e *off-center*. Uma célula *on-center* é excitada quando a luz é enviada próxima a área central do campo receptivo, e inibida em caso contrário. O processo inverso ocorre com as células *off-center*, que é excitada quando a luz é disposta em forma de anel em volta do campo receptivo, e inibida de outro modo. Quando uma luz é aplicada na retina inteira, todos os fotorreceptores são estimulados pela luz incidente e as células não são estimuladas nem inibidas, uma vez que são excitadas em uma área e inibidas na outra. A Figura 2.3 mostra um feixe de luz incidindo sobre a retina. Ela excita as células *on-center* e inibe as células *off-center*. Deste modo, a percepção de claro e escuro em um cenário visual é dependente principalmente da iluminação relativa de seus objetos vizinhos.

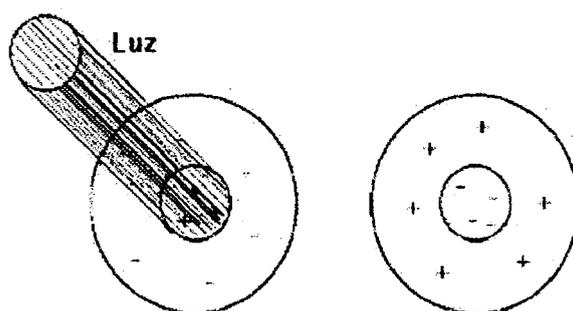


Figura 2.3 – Campos receptivos *on-center* e *off-center*.

2.1.3.2 Células amácrimas

As células amácrimas têm uma ampla variedade de formatos e neurotransmissores, podendo chegar a vinte tipos de formatos diferentes. Muitas de suas funções ainda são

desconhecidas. Verificou-se que algumas delas tem a função de detecção de movimentos em sapos e coelhos (HUBEL, 1995).

Estas células são excitadas pelas células bipolares, mas possivelmente também pelos corpos sinápticos dos fotorreceptores. Por sua vez, transmitem sinal inibitório para as células ganglionares. Mas, diferentemente das células bipolares e horizontais, atuam de modo transitório e não de modo contínuo e fixo. Quando os fotorreceptores são estimulados primeiro, o sinal inibitório transmitido pelas amácrimas é muito intenso, mas desaparece numa fração de segundos (GUYTON, 1998).

2.1.3.3 Células bipolares

As células bipolares permitem a ligação do sinal visual dos fotorreceptores às células ganglionares, sendo que na fóvea existe aproximadamente uma célula bipolar para cada cone e cada célula ganglionar. Contudo, à medida que nos afastamos da região central ocorre uma maior convergência de cones para cada célula bipolar. Estas células possibilitam a excitação das células ganglionares.

2.1.3.4 Células horizontais

As células horizontais, situadas entre as bipolares e fotorreceptores, espalham-se pela retina transmitindo sinais lateralmente. São excitadas pelos corpos sinápticos dos fotorreceptores. Estas, por sua vez, transmitem sinais principalmente às células bipolares, localizadas lateralmente em relação aos fotorreceptores. Tem caráter inibitório, provocando a inibição das células bipolares laterais, permitindo, com isso, que cones e bastonetes transmitam sinais excitatórios em linha direta com as bipolares na área de excitação, enquanto transmitem sinais inibitórios através das células bipolares vizinhas (Figura 2.1 (b)). Deste modo, permitem uma melhor definição de cores e contrastes em cenários visuais (GUYTON, 1998).

2.1.3.5 Células fotorreceptoras

A Figura 2.4 apresenta a estrutura básica de um fotorreceptor. Morfologicamente, os cones apresentam uma terminação superior cônica e os bastonetes são em geral mais longos e

finos que os cones. No segmento externo é que se encontram as substâncias fotossensíveis, a rodopsina para os bastonetes e outras substâncias fotoquímicas semelhantes para os cones. Nos cones, estas substâncias se diferenciam de acordo com a sensibilidade a matizes de cada cone. O corpo sináptico é responsável pela conexão entre as células bipolares e horizontais. Já o segmento interno é responsável pela síntese contínua de rodopsina ou outra substância equivalente. Esta substância é então transportada para o segmento externo, viajando através dos discos até serem degeneradas e dissolvidas pelo epitélio pigmentar (GUYTON, 1998).

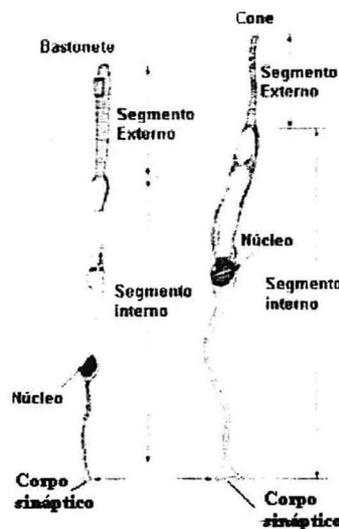


Figura 2.4 – Fotorreceptores. Extraído de PEREZ(1999)² e traduzido.

2.2 CÓRTEX VISUAL

A palavra córtex, significa casca em latim, isso porque o córtex é a camada mais externa do cérebro (Figura 2.5), tendo nos animais mais evoluídos uma espessura de 2 a 6 mm. O córtex cerebral é responsável pelas sensações de consciência, memória e inteligência.

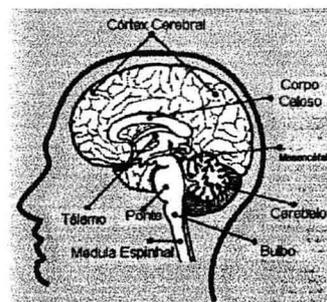


Figura 2.5 – Áreas do cérebro³.

²PEREZ, A. N. Visão Biológica. www.femanet.com.br/~andres/visao/monografia/visao.htm

O córtex é dividido em quatro lóbulos, tendo nos seres humanos as seguintes funções: (1) frontal (movimento e olfato); (2) parietal (sensações somáticas); (3) occipital, (visão); (4) temporal (audição e memória) (DOWLING, 1992). O item (3) é também chamado de córtex visual. Pesquisas mostram que o córtex visual também é responsável pelo processamento de orientação, cores e profundidade (TANAKA, 2000). A Figura 2.6 mostra os principais tipos de neurônios encontrados no córtex humano.

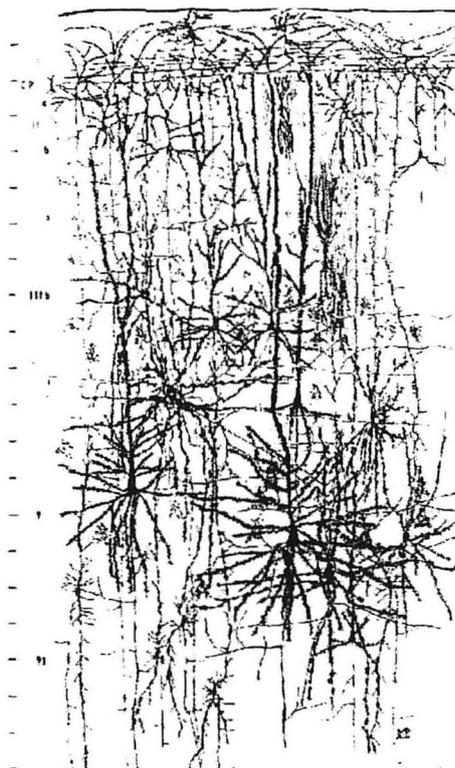


Figura 2.6 – Tipos de neurônios no córtex, marcado pelo método de Golgi. Extraído de SCHMOLESKY(2000).

Na Figura 2.7 temos o caminho visual dos sinais ópticos desde a retina até córtex visual, ou lóbulo occipital. Cada núcleo geniculado dos primatas divide-se em seis camadas empilhadas uma sobre as outras. As camadas recebem informações provenientes da retina de cada um dos olhos de forma alternada, isto é, um dos olhos comunica-se com as camadas 1,3,5 e o outro com as camadas pares. Essa sobreposição de imagens auxilia o mecanismo da percepção visual de profundidade, uma vez que as imagens ficam sobrepostas nas camadas. Semelhantemente às células ganglionares, os campos receptivos do núcleo geniculado lateral apresentam também células do tipo *on-center* e *off-center*. Podemos dividir o córtex visual em

³ http://www.psicobiologia.com.br/sambalele/Saiba/Divisoes_do_SN.htm

duas regiões: córtex estriado ou primário e córtex pré-estriado. O córtex visual primário é a única região do córtex visual que recebe informação visual direta por meio do núcleo geniculado lateral.

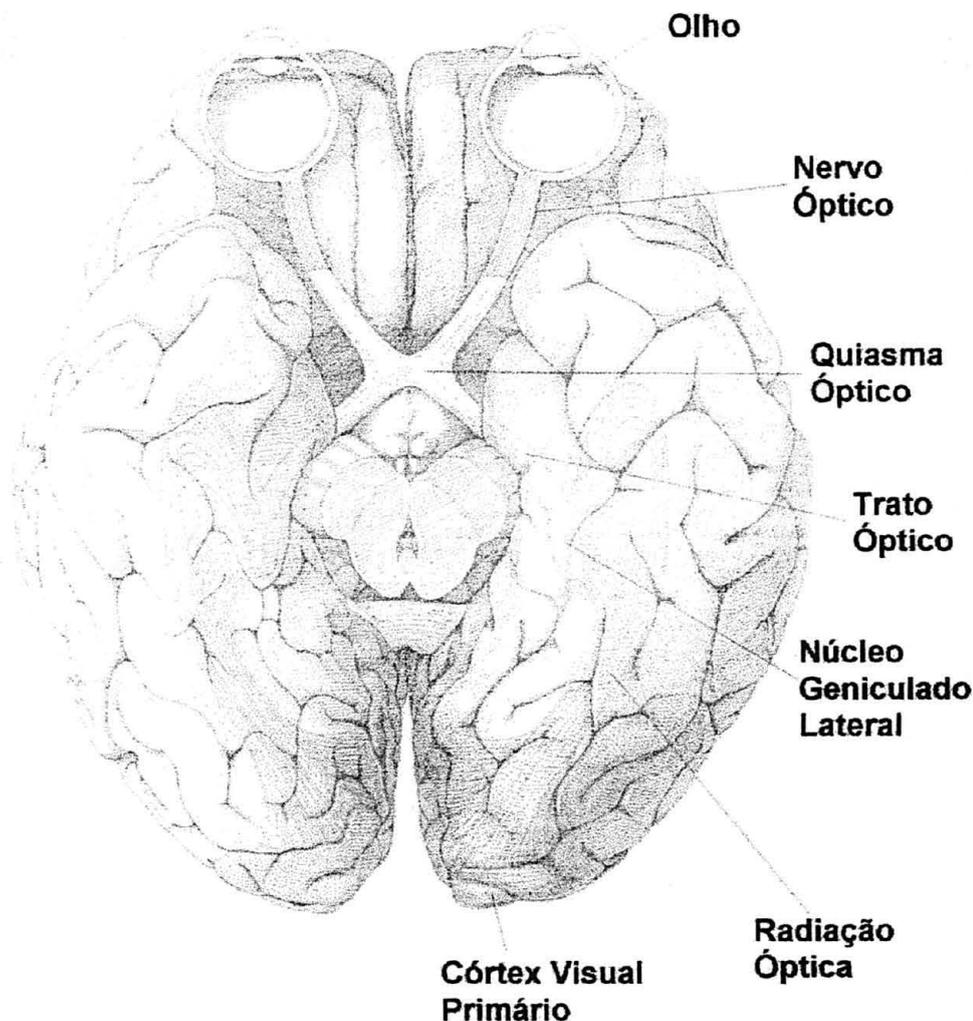


Figura 2.7 – Caminho Visual. Extraído de HUBEL(1995) e traduzido.

2.2.1 Estrutura Neural

No córtex visual, encontramos três tipos básicos de neurônios : piramidais, estelares e os interneurônios planos ou esparsamente estriados. As células piramidais possuem o corpo celular em forma de pirâmide e dendritos apicais e basais sendo que seus dendritos são cobertos por espículas (Figura 2.8 (a)). Seus axônios projetam-se para outras regiões do córtex e até mesmo para outras regiões do encéfalo. Os neurônios estelares (Figura 2.8 (b)), que permanecem sempre na mesma região cortical, são geralmente pequenos e seus corpos celulares lembram o formato de estrela (SCHUMOLESKY,2000). Seus dendritos também

podem ter espículas. Os interneurônios tem o corpo celular arredondado e seus dendritos tem pouca ou nenhuma espícula (Figura 2.9, células (b), (d), (e) e (g))

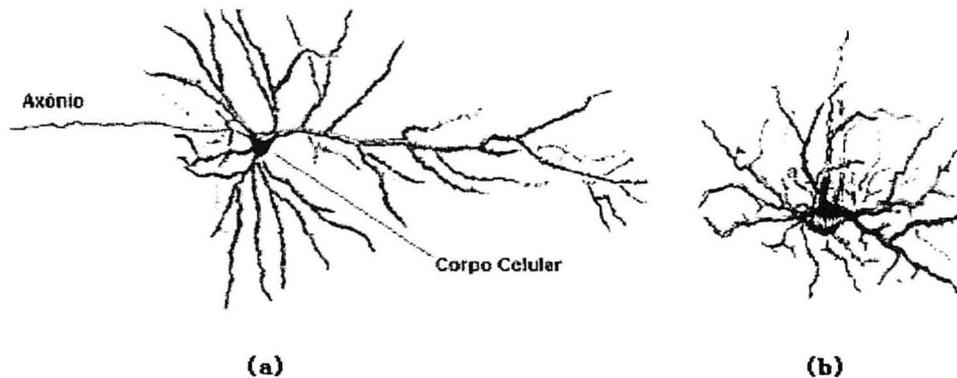


Figura 2.8 – Neurônios piramidal e estelar da esquerda para a direita⁴

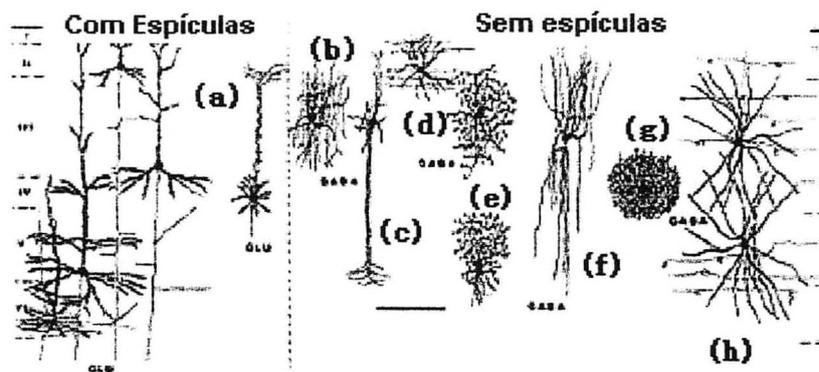


Figura 2.9 – Neurônios com e sem espículas. Extraídos de SCHMOLESKY(2000) e traduzido.

2.2.2 Camadas do córtex

O córtex visual é composto de seis camadas sendo estas 1, 2, 3, 4, 5 e 6. Algumas destas camadas se dividem em subcamadas (Figura 2.10). Dentro destas camadas distribuem-se os principais tipos de células neurais do córtex.

A primeira camada é composta de uma densa malha de sinapses formadas pelos dendritos apicais das células piramidais da camada 2 e os impulsos provenientes do núcleo geniculado lateral. Embora com poucos neurônios, esta rede de conexões afeta o direcionamento dos sinais das células piramidais para as camadas mais abaixo. As camadas 2,

⁴ <http://www.anatomy.dal.ca/html/Human.Neuroanatomy/Labs/Lab8.html>

3A e 3B são compostas de somas e dendritos de células piramidais (Figura 2.11). Muitos dos axônios e dendritos de neurônios de outras camadas do córtex estão nas camadas 2 e 3A.

Nas camadas 4 estão os receptores das inervações do núcleo geniculado lateral. Os neurônios estelares, situados em 4C providenciam uma densa inervação para as camadas 4A e 3B e espalham inervações em 4C, 5 e talvez até a camada 6 (LUND, apud SCHMOLESKY, 2000). Sabe-se que a camada 6 tem axônios na camada 4C e também recebe uma entrada direta do núcleo geniculado lateral (Figura 2.12).

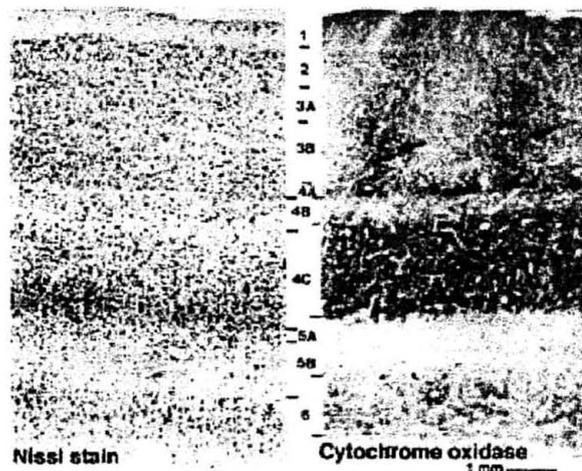


Figura 2.10 – Camadas do córtex visual nos primatas, usando diferentes técnicas de marcação. Extraído de SCHMOLESKY(2000).

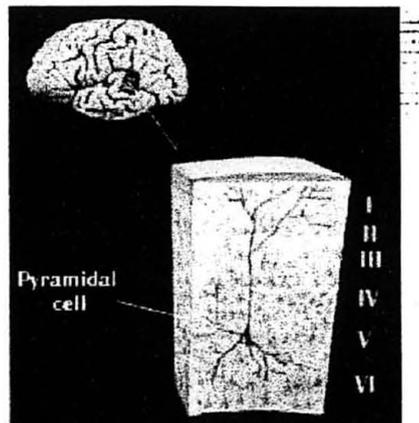


Figura 2.11 – Distribuição das células piramidais nas camadas do córtex visual⁵

⁵ : http://www.epub.org.br/cm/n01/arquitet/camadas_i.htm

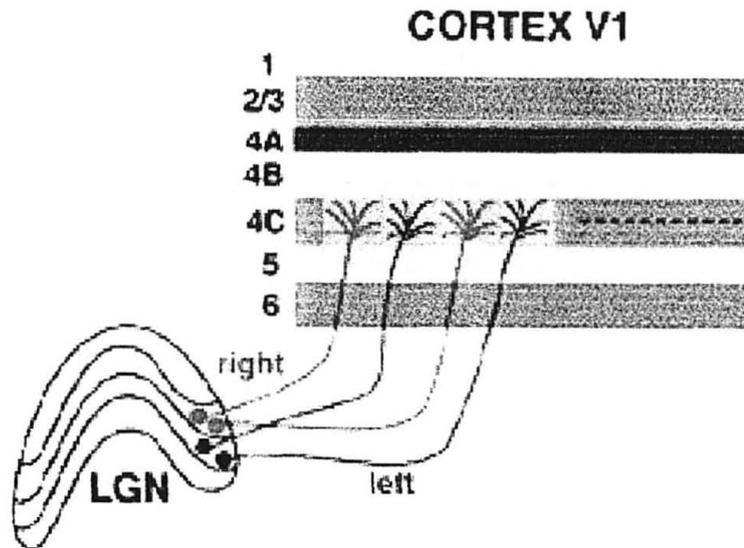


Figura 2.12 – Projeção do núcleo geniculado lateral na camada 4C. Cada olho tem uma projeção do núcleo geniculado lateral (SCHMOLESKY,2000).

2.3 NEURÔNIOS

O sistema nervoso e sistema endócrino são responsáveis pela maior parte das funções do organismo. O sistema nervoso pode ser dividido em duas partes: divisão sensorial e divisão motora.

Na divisão sensorial, as experiências sensoriais são transmitidas através dos receptores sensoriais. Tais experiências podem resultar em uma reação imediata, ou ser armazenada no encéfalo por minutos, semanas, meses ou anos.

A divisão motora é responsável pela coordenação das atividades corporais como contração dos músculos e secreção glandular. Os músculos e glândulas, neste caso, são chamados de efetores.

Deste modo, o sistema nervoso tem por principal função processar informações sensoriais e produzir respostas motoras adequadas. Estas informações são conduzidas e processadas através da conexão entre uma sucessão de células apropriadas para isto, os neurônios. À conexão entre um neurônio e outro, damos o nome de sinapses.

Basicamente todo o sistema nervoso é composto por neurônios e células gliais. As células gliais são células responsáveis pela sustentação e proteção do sistema nervoso. A glia não conduz sinais, mas provem sustentação para a condução dos impulsos elétricos por todo o sistema nervoso.

2.3.1 Partes da célula neural

Pode-se dizer que um neurônio é composto de três partes, corpo celular ou soma, axônio e dendritos. Cada neurônio possui um axônio único que surge de uma região específica do corpo celular, chamada de cone axônico, ou cone de implantação. Os axônios têm comprimento variável podendo chegar a metros, e são revestidos por células da glia, formando a bainha de mielina. Esta bainha é composta por substâncias gordurosas, servindo como isolante elétrico, permitindo, assim, uma melhor rapidez na emissão de sinais elétricos (Figura 2.13).

2.3.2 Botões sinápticos

Na superfície do soma e na superfície dos dendritos encontra-se uma grande quantidade de pequenos botões, denominados botões sinápticos. Aproximadamente 80 a 90 por cento destes botões encontram-se na superfície dos dendritos (GUYTON, 1998).

Microscopicamente, estes botões apresentam uma forma variável sendo a maioria semelhante a pequenos nodos ovalados ou redondos. Portanto, são também chamados de nodos terminais, botões terminais, pés terminais, ou simplesmente terminais sinápticos. Na Figura 2.13 podemos observar que há um espaço entre o terminal emissor da sinapse, ou terminal pré-sináptico, e a membrana receptora da sinapse, ou membrana pós-sináptica. Este espaço é chamado de fenda sináptica e tem uma distância de 200 a 300 angstroms.

Cada terminal sináptico é composto por vesículas sinápticas, que liberam substâncias excitatórias ou inibitórias da sinapse, e por mitocôndrias, responsáveis pelo fornecimento de ATP (Adenosina Trifosfato), que é a energia necessária para a síntese de nova substância transmissora. A quantidade de substâncias armazenadas nas vesículas é suficiente para uma atividade de segundos a poucos minutos de atividade, exigindo então que a substância transmissora seja sintetizada com extrema rapidez (GUYTON, 1998).

Quando um impulso se propaga através de um terminal pré-sináptico, as vesículas liberam as substâncias transmissoras através da fenda sináptica, produzindo uma alteração na permeabilidade da membrana do neurônio receptor, produzindo excitação ou inibição do receptor, dependendo da substância liberada. Este processo ocorre porque o impulso provoca a penetração de íons cálcio no terminal, que supostamente atraem as vesículas para a membrana, provocando a ruptura de uma ou mais delas, derramando as substâncias

transmissoras na fenda sináptica. Acredita-se que a membrana do neurônio pós-sináptico contenha moléculas receptoras específicas que se ligam à substância transmissora (GUYTON, 1998). Como as substâncias podem ser excitatórias ou inibitórias e o neurônio produz somente um tipo destas substâncias, ele pode ser excitatório ou inibitório, dependendo do tipo de substância que ele produz.

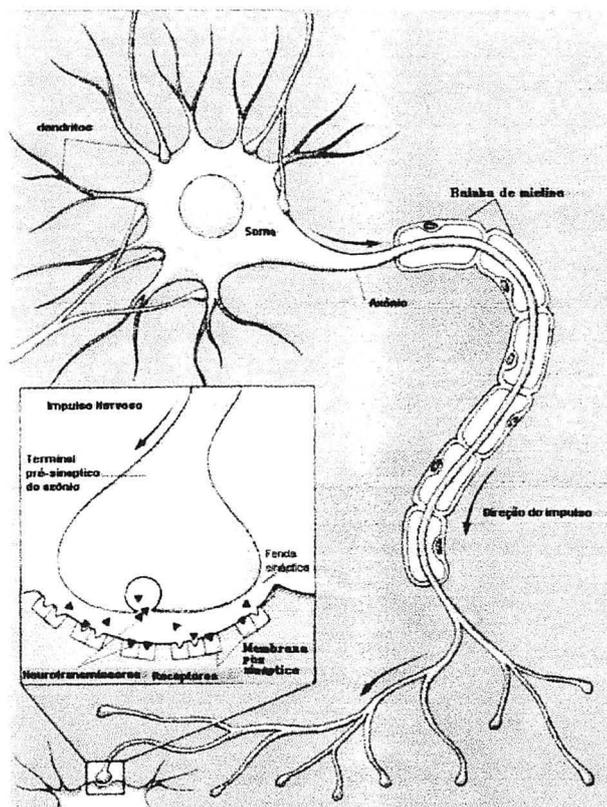


Figura 2.13 – Partes de um neurônio e detalhes de uma sinapse.⁶

2.3.3 Tipos de células neurais

As células neurais podem ser classificadas por diferentes parâmetros, como o número de arborizações que se originam do soma, formato da célula, função, e tamanho ou ausência de axônio.

Quanto ao número de ramificações, temos as células unipolares, bipolares e multipolares. Nas células unipolares uma única arborização dá origem a muitas ramificações, sendo mais predominantes nos invertebrados. Neste caso, a raiz da árvore é um axônio e este se ramifica formando dendritos. As células bipolares são compostas por um dendrito e um

⁶ <http://www.pfizer.com/brain/dlgameprint.html>

axônio. Como exemplo, temos as células da retina (Figura 2.1 (b)). Já nas multipolares, os dendritos partem de diferentes partes do soma, sendo que um axônio pode surgir a partir do soma ou de um dos dendritos. Como exceção, temos as células piramidais e as de Purkinge, que não se ramificam de qualquer parte do soma, mas de locais específicos.

Em relação ao formato do corpo celular, temos no córtex do cerebelo as células de Purkinge, com formato de pêra e as células granulares, com formato esferoidal. Um pouco maiores, mas com formato esferoidal, são as células dos neurônios sensitivos dos gânglios espinhais. No córtex visual temos as células piramidais e estelares, já comentados na seção 2.2.1. As células estelares com espículas são excitatórias, enquanto que as estelares lisas não possuem espículas e têm caráter inibitório.

Quanto à função, temos os neurônios motores, interneurônios e sensitivos. Os neurônios motores transmitem sinais sinápticos. Já os interneurônios têm a função de estabelecer conexões entre os neurônios. Os neurônios sensitivos têm a função de receber os sinais exteriores e retransmitir até o sistema nervoso central.

Finalmente, podemos também classificar as células quanto à presença ou não de axônio. As células com axônio longo são do tipo Golgi I e as com axônio curto ou sem axônio são do tipo Golgi II. As do tipo Golgi I são responsáveis pela conexão entre diferentes regiões do cérebro ou da medula até os órgãos efetores. Como exemplo, temos as células piramidais. No cérebro dos animais mais desenvolvidos, as células Golgi II apresentam maior concentração que as células Golgi I (DOWNLING, 1992). Estas células (Golgi II) não se interconectam com outras regiões, realizando somente interações locais em pequenas regiões.

2.3.4 Cones de crescimento e fatores de crescimento

Algumas substâncias podem inibir ou estimular o crescimento neural. Estas substâncias são chamadas de *fatores de crescimento*.

Durante o desenvolvimento embrionário, o crescimento dos axônios é orientado por estruturas pontudas chamadas de cones de crescimento. Elas localizam-se na borda principal do neurônio e apresentam dois tipos de estruturas móveis: uma longa estrutura em forma de ponta chamada filopódio e uma outra com pés largos e finos na membrana chamados lamellipodia. O filopódio coleta informações do ambiente, permitindo, com isso, que campos de influência alterem a orientação do cone de crescimento.

Nos mamíferos, algumas destas características são mantidas nos axônios na idade

adulta, possibilitando sucesso em algumas cirurgias de reimplante. Contudo, o sistema nervoso central é embebido por proteínas que inibem o crescimento axonal.

Gradientes de concentração local de fatores de crescimento do nervo podem iniciar e dirigir o crescimento e movimento do cone. Neurotransmissores específicos também podem alterar a elongação e o movimento do cone.

Observou-se que o nível intracelular de cálcio é um dos fatores que influenciam o comportamento dos cones de crescimento, havendo um intervalo para crescimento máximo, enquanto que fora deste intervalo o crescimento é moderado. Além disso, taxas muito altas ou muito baixas de cálcio podem ocasionar a morte neural.

Portanto, a presença de fatores que alterem o comportamento natural dos cones de crescimento e taxas inadequadas de fatores de crescimento pode causar profundas alterações morfológicas e funcionais.

2.3.5 Neuroplasticidade

No sistema nervoso, neuroplasticidade é “uma alteração na estrutura ou função causada por aprendizado, habituação ou lesão” (HAMILTON, 1992). Em caso de lesão, essas alterações causam mudanças nos padrões biológicos, no sentido positivo de neutralizar a influência da lesão.

De outro modo, podemos dizer que neuroplasticidade são “alterações estruturais das células em resposta à uma experiência como adaptação às condições mutantes e estímulos repetidos”⁷. Portanto, descrever ramificações neurais não é somente a apresentação aleatória das ramificações ou simplesmente o uso de funções probabilísticas para descrever várias áreas funcionais (HAMILTON, 1992). Podemos ter um entendimento mais profundo de como as redes neurais biológicas se configuram e se interagem, simulando seu crescimento individual e corporativo. Atualmente, estudos realizados na fase embrionária permitem a simulação do crescimento neural utilizando diferentes padrões que influenciam no crescimento, encolhimento, mudança da direção do crescimento, ou alteração da forma neural.

A seguir veremos algumas mudanças que podem ocorrer causadas pelo aprendizado e habituação, e também as alterações causadas por lesão neural.

⁷ <http://www.epub.org.com.br/cm/n05/tecnologia/plasticidade2.html>

2.3.5.1 Habituação

Observou-se que certos reflexos, como afastar um membro de um estímulo moderadamente doloroso, cessam de ser produzidos após uma certa repetição do mesmo estímulo. Parte dessa diminuição ocorre devido à redução das atividades sinápticas entre os neurônios sensoriais e interneurônios, uma vez que o neurônio sensorial diminui a quantidade liberada de substância transmissora. A essa redução de atividade sináptica dá-se o nome de habituação (LUNDY-EKMAN, 2000).

Após um certo intervalo, suficiente para recuperação das substâncias transmissoras dos neurônios sensoriais, os efeitos da habituação cessam. No entanto, com a repetição prolongada do estímulo, ocorrem alterações morfológicas permanentes na estrutura neural, diminuindo o número de conexões sinápticas. O conhecimento deste efeito é utilizado em pessoas com sensibilidade extremada a um estímulo. A pessoa é submetida a estímulos repetitivos e gradativos até que o processo de habituação seja satisfatório.

2.3.5.2 Aprendizado

Estudos revelam que o aprendizado e a memória dependem de alterações persistentes e de longa duração da força das conexões sinápticas (LUNDI-EKMAN, 2000). Durante o início do aprendizado, grandes áreas do cérebro mostram atividades. Posteriormente, após a aprendizagem somente algumas áreas apresentam reações aumentadas às tarefas aprendidas. Deste modo, pode-se dizer que alguns neurônios reforçam e aumentam suas conexões sinápticas, tornando-se mais aptos para execução da tarefa (LUNDI-EKMAN, 2000). Este conceito é muito utilizado em redes neurais computacionais.

2.3.5.3 Lesão

A lesão do corpo celular leva sempre à morte da célula. No entanto, a lesão ou seccionamento axonal levam a alterações degenerativas que nem sempre culminam com a morte celular.

Quando um neurônio de um mamífero adulto morre, ele não é substituído. No entanto, ocorrem alterações morfológicas em seus neurônios vizinhos com a intenção de suprir a atividade funcional do neurônio morto.

No caso de invertebrados simples, a retirada de uma célula particular afeta o destino de células vizinhas. No desenvolvimento do grilo, por exemplo, a arborização axonal dos neurônios sensoriais é mudada de acordo com a ausência ou presença de células vizinhas.

As alterações em um axônio lesado, chamados de brotamento, podem ser colaterais, ou regenerativas. Na Figura 2.14 temos em a) as conexões antes da lesão e em b) a célula pré-sináptica morre e ocorre o brotamento colateral, no sentido de inervar o neurônio pós-sináptico. Em c) ocorre a lesão do axônio da célula pré-sináptica e a morte do neurônio pós-sináptico, ocorrendo brotamento regenerativo e inervação lateral de células não lesadas.

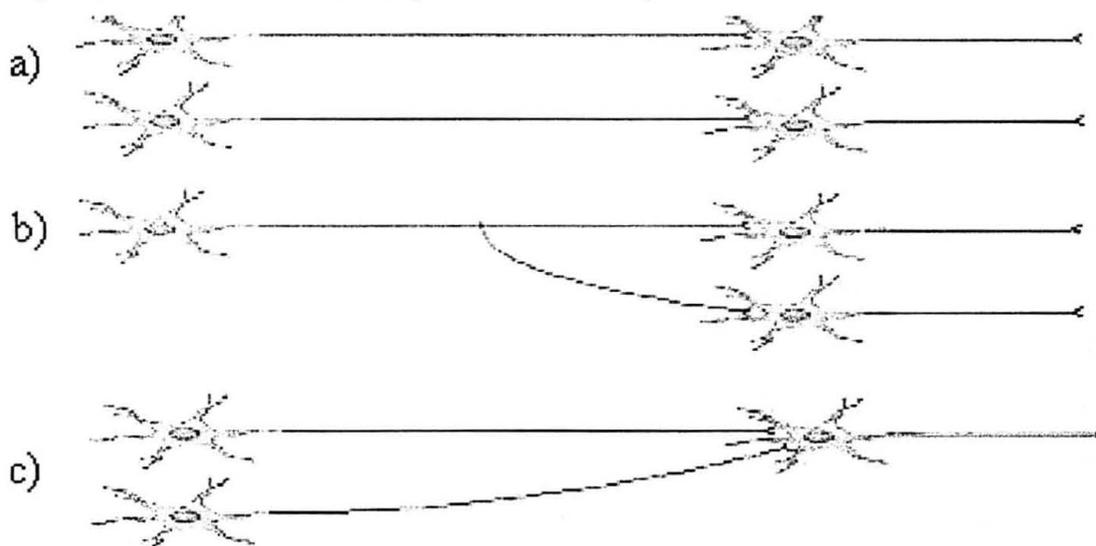


Figura 2.14 – Alterações neuromórficas após ocorrência de lesão. a) conexões antes da lesão, b) brotamento colateral (depois da morte do neurônio pré-sináptico) e c) brotamento regenerativo (depois da morte do neurônio pós-sináptico).

Há uma maior frequência de regeneração neural no sistema nervoso periférico, por causa da produção de fatores de crescimento nesta área, contribuindo para a recuperação axonal (SERMASI, TROPEA e DOMENICI, 1999). Sobre condições normais, não ocorre regeneração no sistema nervoso central.

2.3.6 Forma e função neural

Enquanto alguns neurônios possuem arborescências axonais e dendríticas altamente complexas, outros são relativamente simples comparados com outras células do sistema

nervoso. Estas variações na forma neural têm importantes conseqüências na ontogenia⁸ e filogenia⁹ (PURVES, 1988).

As pesquisas na área da eletrofisiologia, através da inclusão de eletrodos e marcadores em neurônios particulares, permitiram relacionar a forma com a função celular.

Anatomias do sistema nervoso de animais simples apresentam pouca variabilidade de formato para células que servem a uma determinada função, sugerindo que à medida que aumenta a complexidade do animal, surge a necessidade de um maior número de conexões, ocorrendo também uma grande variabilidade dendrítica e axonal (PURVES, 1988).

As diferenças morfológicas em animais de diferentes tamanhos e formas têm uma variedade de implicações funcionais.

“Em resumo, a modulação da forma dendrítica pode mudar o número de entradas que o neurônio recebe, a característica de suas atividades, a maneira na qual eles integram a informação sináptica que os afetam, e o tamanho das unidades neurais na qual eles participam” (COELHO, 1998).

2.4 FATORES NEUOTRÓFICOS

A aquisição das características morfológicas dendríticas depende de diferentes fatores extracelulares. Durante o seu desenvolvimento as células neurais exercem e sofrem influências do ambiente em que estão se desenvolvendo. Alguns desses fatores continuam a atuar mesmo após a célula atingir sua maturidade. Após a migração para o córtex os neurônios piramidais estendem seus dendritos apicais para a superfície da pia-máter¹⁰ (WHITFORD et al., 2002). Esse crescimento é devido à presença de uma substância quimioatratora denominada *Semaphorin 3A (Sema3A)* (POLLEUX, MORROW e GHOSH, 2000; POLLEUX et al., 1998). As pesquisas realizadas por POLLEUX, WHITFORD e colegas, constataram que a presença de *Sema3A* atraem os dendritos apicais tendo efeito oposto com relação aos axônios. Entre outras substâncias que atuam como fatores neurotróficos temos NGF (nerve grown factor), BDNF (brain derived neurotrophic factor), NT-3 (neurotrophin-3) e NT-4 (POLLEUX, MORROW e GHOSH, 2000; POLLEUX et al.,

⁸ Ontogênese : série de transformações que passa o indivíduo

⁹ Filogênese: história genealógica de uma espécie; Filogenia: historia das raças.

¹⁰ Membrana mais interna das três que envolvem o aparelho cerebrospinal

1998). Essas substâncias exercem seus efeitos através da família TRK de receptores tyrosina kinase. Uma grande concentração de TrkB na camada 6 resulta na predominância de pequenos dendritos basais, enquanto que NT-3 parece influenciar fortemente na complexidade dendrítica da camada 4, e nada apresentando na camada 5 (WHITFORD et.al.,2002). As pesquisas utilizando *Sema3A* “revelam que um único sinal quimiotrópico podem padronizar o desenvolvimento dos axônios e dendritos” (POLLEUX, MORROW e GHOSH, 2000). A Figura 2.15 mostra o processo de desenvolvimento de um neurônio piramidal no córtex. Observe que os dendritos apicais são direcionados para cima, em direção a pia-máter, enquanto que os axônios se dirigem para a matéria branca.

Além dos fatores neurotróficos, os dendritos e axônios de um neurônio podem sofrer influências de outras células neurais ao redor, no sentido de realizar novas conexões, seja no período de crescimento da célula ou adaptações de plasticidade. À estas influências de atração ou repulsão que ocorrem no ambiente celular damos o nome de tropismo.

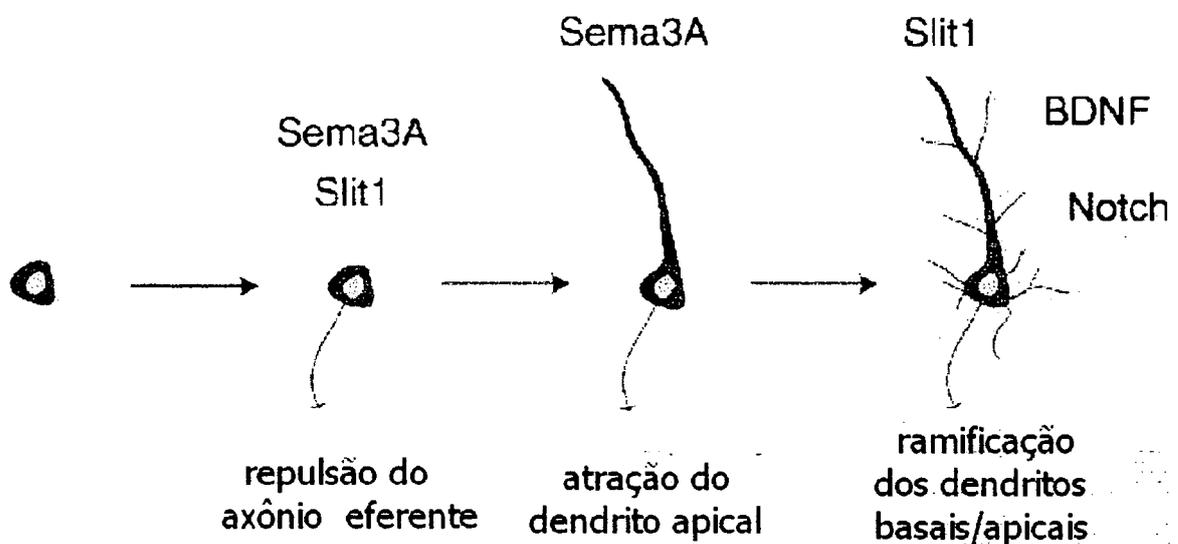


Figura 2.15– Ação de fatores extra celulares influenciando a morfologia cortical. Nesta figura, um novo neurônio pós-mitótico chega à área cortical, onde encontra um gradiente de Sema3A , que direciona o crescimento axonal através da matéria branca. O mesmo gradiente de Sema3A atrai os dendritos apicais do neurônio através da superfície pial. Outros fatores, tais como BDNF e Notch, controlam o crescimento e ramificação subseqüentes. Extraído de WHITFORD,et al. (2002), modificado e traduzido.

3 MÉTODOS E TÉCNICAS RELACIONADOS À SÍNTESE NEURAL

A tentativa de definir precisamente quais eram as sentenças válidas e quais as formas estruturais de uma linguagem, deu origem ao conceito de gramática. Noam Chomsky formalizou o conceito de gramática em 1956, classificando e definindo-as na 'hierarquia de Chomsky'.

Em 1968, o biólogo Aristid Lindemayer observou que plantas seguem um padrão de crescimento que podem ser formalizadas em um modelo matemático que descreve as regras a serem utilizadas em suas ramificações. Este modelo, conhecido como gramática L-System (LS), permite a definição de regras de crescimento não somente de plantas, mas outros organismos biológicos, além de outras estruturas ramificadas e segmentadas. Associando ações de recursos gráficos às cadeias geradas, podemos sintetizar a evolução de organismos biológicos de um modo mais realístico.

PRUSINKIEWICZ(1986) propõe em seu artigo o uso de LS estocásticas, para geração de imagens computacionais e HAMILTON(1993) propõe o uso de gramáticas LS para a simulação do crescimento de ramificações neurais. McCORMICK e MULCHANDANI (1994) ampliam a proposta com uma L-System que utiliza informações de torção e curvatura.

Posteriormente, ASCOLI e KRICHMAR (2000), criam um programa que utiliza LS estocásticas para descrever estruturas neuromórficas. Utilizando um conjunto de medidas extraídas de dados experimentais, geram neurônios artificiais diferenciados e de várias classes morfológicas. Estas medidas são retiradas de vários formatos de traçado, tais como a notação EUTECTIC, SWC e NEUROLUCIDA, entre outros.

É impossível tratar de síntese e simulação sem falarmos de alguns conceitos de estatística. Nas seções 3.1 a 3.3 tratamos de conceitos de probabilidade, funções de densidade e regra ou teorema de Bayes. Estes conceitos serão usados nas seções seguintes.

As seções 3.4 a 3.6, tratam inicialmente de algumas definições existentes no campo de linguagens formais, autômatos e compiladores, conhecimento essencial para o desenvolvimento deste projeto e compreensão das seções posteriores. A seguir, na seção 3.7, definiremos gramáticas gráficas e apresentamos as gramáticas LS de Lyndemayer e a LS

estocástica descrita no artigo de PRUSINKIEWICZ(1986). Na seção 3.8, descrevemos alguns formatos utilizados para notação de traçado neural. Finalmente, na seção 3.9, discutimos os algoritmos utilizados em ASCOLI e KRICHMAR(2000).

3.1 PROBABILIDADE E ESTATÍSTICA

Usando coleta de dados e sua organização, é possível caracterizar a variabilidade de um fenômeno observado e elaborar conjecturas a seu respeito. Digamos que queiramos prever ou simular a quantidade de meninos que nascerão em 2003. Digamos, por exemplo, que a quantidade de meninos em relação ao número total de crianças nos anos anteriores foi de 48%. Se o percentual nos anos anteriores foi de 48%, é possível que em 2003 o percentual seja um valor próximo de 48%, com uma certa margem de erro. Em suma, não tem como saber exatamente qual a porcentagem visto que o fenômeno ainda não ocorreu e que sua natureza é aleatória. Ou seja, visto que há uma certa regularidade nos nascimentos, podemos fazer uma previsão do fenômeno com um certo grau de confiabilidade. A esta regularidade damos o nome de probabilidade.

Partindo então de observações de eventos já ocorridos podemos extrair regras que nos permitem dizer quão provável será um determinado resultado. O uso de probabilidade só é válido quando não é possível prever com exatidão um fato que irá acontecer, isto é, não dispomos de nenhuma fórmula para prever o resultado.

3.1.1 Classes e intervalos de classes

Denota-se por *dados brutos* a massa de dados resultante após uma fase de levantamento de dados. Uma vez ordenados estes dados, podemos visualizá-los com maior clareza em relação ao seu comportamento e distribuição.

Para que possamos ter uma melhor visualização desta distribuição, estabelecemos faixas ou intervalos de variação com a finalidade de saber qual a quantidade de dados que se situa em cada faixa. A quantidade de dados na faixa é denotada por *freqüência* e as faixas denotamos por *classes* ou *caixas*.

Intervalo de classe é a diferença entre valores extremos de uma classe, que recebem o nome de *limite inferior e superior* de classe. Normalmente as classes possuem a mesma

amplitude, isto é, o mesmo intervalo. Tais classes são ditas *homogêneas* ou de *grades regulares e irregulares*, em caso contrário.

O intervalo de cada classe, o número de classes, e a regularidade ou não das classes dependem do tipo de dados que se está tratando e do que se irá fazer com estes dados. Em alguns momentos o uso de classes irregulares pode ser interessante. Enfim, não existe uma regra 100% confiável.

3.1.2 Eventos e probabilidade

Sejam E_1, E_2, \dots, E_n , todos os eventos que possam ocorrer em um experimento. Definimos como espaço amostral o conjunto $S = E_1 \cup E_2 \cup \dots \cup E_n$. Por exemplo no lançamento de um dado, os possíveis valores obtidos são 1, 2, 3, 4, 5, 6. O espaço amostral seria então $S = \{1, 2, 3, 4, 5, 6\}$ e um evento E_i é qualquer subconjunto de S .

Desde que esses eventos possam ser discretizados, a probabilidade de um evento E_i é definido como $P(E_i) = \frac{\text{Num}(E_i)}{\text{Num}(S)}$, sendo $\text{Num}(E_i)$ e $\text{Num}(S)$ o número de elementos de E_i e

S respectivamente. Se temos um evento $A = \{2,3,5\}$ então $P(A) = \frac{\text{Num}(A)}{\text{Num}(S)} = \frac{3}{6} = 0.5 = 50\%$.

Usamos os seguintes axiomas para probabilidade:

- 1) $P(S) = 1$
- 2) $0 \leq P(E_i) \leq 1$
- 3) $P(A \cup B) = P(A) + P(B) - P(A \cap B)$, e caso os eventos sejam mutuamente exclusivos temos $P(A \cap B) = 0$, então $P(A \cup B) = P(A) + P(B)$.

3.2 FUNÇÃO DE DISTRIBUIÇÃO

Uma *variável aleatória* X serve para representar um conjunto de valores que podem ocorrer num experimento específico. Por exemplo, suponha que estejamos cadastrando o peso, a altura e a idade de diferentes pessoas. Se estivéssemos interessados em extrair alguns dados estatísticos com relação a peso, teríamos então peso como sendo a nossa variável aleatória, pois ele pode ter diferentes valores para cada tipo de pessoa. Conseqüentemente, se estivéssemos interessados em altura, esta seria nossa variável aleatória, e assim sucessivamente.

Uma *função de densidade* é uma função $p(x)$ correspondente a probabilidade de ocorrência de um valor x para uma variável aleatória X . Deste modo, $p(x)$ deve satisfazer o seguinte critério:

$$p(x) \geq 0 \text{ para qualquer valor } x$$

e

$$\int_{-\infty}^{\infty} p(x) dx = 1$$

Em resumo, a soma das probabilidades calculadas para todos os valores de X deve ser igual a 1.

Uma *função de distribuição* (FD) consiste em acumular para cada frequência de um evento observado a soma das frequências dos eventos anteriores, de modo que o último evento terá valor 1. De outro modo, se tenho n valores representados em uma função de densidade, $p(x_i)$, $1 \leq i \leq n$, é a probabilidade de x_i . Então $pa(x_i)$, é a probabilidade acumulada de x_i , denotada por $pa(x_i) = \sum_{k=1}^i p(x_k)$. Note que, quando k for igual a n , $pa(x_n)=1$. Podemos também definir uma função de densidade $P(x)$ como:

$$P(x) = P(X \leq x) = \int_{-\infty}^x p(s) ds$$

3.3 REGRA DE BAYES

Em muitas situações precisamos especificar a probabilidade de ocorrência de um evento específico B , que é afetada pelo fato de um outro evento A ter ocorrido ou não. Queremos então calcular a ocorrência de B condicionada à ocorrência prévia de A , denotada por $P(B|A)$ (probabilidade de B dado A), isto é, as chances de B ocorrer dependem de A (ARANGO, 2001). E sua expressão é dada por

$$P(B | A) = \frac{P(B \cap A)}{P(A)} \quad (4.4)$$

Do mesmo modo temos

$$P(A | B) = \frac{P(A \cap B)}{P(B)} \quad (4.5)$$

Uma vez $P(A \cap B) = P(B \cap A)$, juntando as duas equações (4.4) e (4.5) temos a equação (4.6) conhecida como *regra de Bayes*, ou *teorema de Bayes* sendo esta equação uma das bases para inferência probabilística (RUSSEL e NORVIG, 1995, p. 426). Em especial, usando esta equação podemos criar classificadores bayesianos, bem como redes bayesianas.

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad (4.6)$$

A Figura 3.1 mostra um importante conceito da probabilidade condicional. Inicialmente, o conjunto universo é o espaço amostral S conforme mostrado em na figura 3.1 a). A seguir, dado que A tenha acontecido, o espaço amostral se reduz de S para A , e as conseqüências favoráveis para a ocorrência de B para a intersecção de A e B .

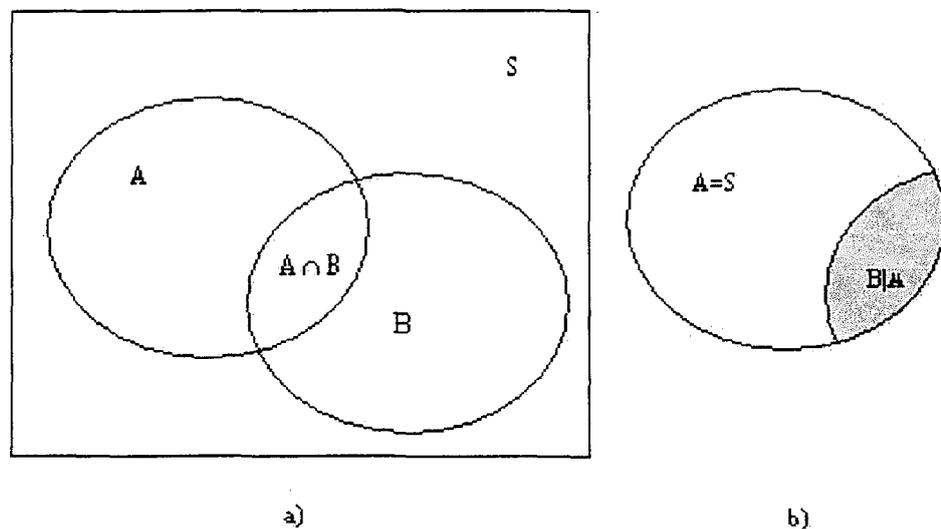


Figura 3.1 – Redefinição de espaço amostral. a)O diagrama de Venn mostra os eventos A , B , e sua intersecção. b)O fato de A ter ocorrido redefine o espaço amostral. Extraído de COSTA e CÉSAR (2001), e traduzido.

3.4 ALFABETOS E LINGUAGENS

As seqüências de código do DNA utilizam símbolos bem definidos que chamamos de A , G , C , T . Podemos ter várias seqüências ou cadeias de DNA utilizando os símbolos A , G , C , T . A um conjunto de símbolos, denotamos por alfabeto. Assim, as seqüências $AGCCTTAC$ e $AGCCTTCTA$ utilizam os símbolos do alfabeto $\{A, G, C, T\}$.

Usaremos a letra grega Σ para representar um alfabeto. Logo, se $\Sigma = \{A, C, G, T\}$ é um alfabeto, então G é um símbolo ou letra de Σ .

Uma cadeia ou palavra sobre um alfabeto é uma seqüência finita de símbolos concatenados desse alfabeto. Se $\Sigma_1 = \{0,1\}$, então 011011 é uma cadeia sobre o alfabeto Σ_1 . Se $\Sigma_2 = \{a, b, c, \dots, z\}$ então $abacaxi$ é uma cadeia sobre Σ_2 . O tamanho de uma cadeia é dado pelo número de símbolos que esta contém. Assim, $w = w_1w_2\dots w_k$ terá tamanho k se $w_i \in \Sigma$, e $i = 1, \dots, k$. Denotamos o tamanho de w por $|w|$ e, portanto $|w| = k$. Se $|w| = 0$ então w é chamado de cadeia vazia e denotamos por ε , isto é $|\varepsilon| = 0$.

Se $x = cao$ e $y = gato$ então $xy = caogato$ é a concatenação de x com y . Se $\Sigma_1 = \{a,b\}$ e $\Sigma_2 = \{0,1\}$ então a concatenação de Σ_1 com Σ_2 é $\Sigma_1\Sigma_2 = \{a0,a1,b0,b1\}$. O símbolo ε é o elemento neutro da concatenação, isto é, se w é uma palavra então $\varepsilon w = w\varepsilon = w$.

Definimos $\Sigma^0 = \varepsilon$, e $\Sigma^1 = \Sigma$ e $\Sigma^n = \Sigma^{n-1}\Sigma$, para $n > 1$. Esta definição não deve ser confundida com expoente de potência em expressões algébricas, já que aqui o que ocorre é a repetição de símbolos justapostos. Deste modo, para Σ_1 do parágrafo anterior, $\Sigma_1^2 = \{aa,ab,ba,bb\}$ e $\Sigma_1^3 = \Sigma_1\Sigma_1^2 = \{aaa,aab,aba,abb,baa,bab,bba,bbb\}$.

Utilizamos o símbolo $*$ acima e à direita de um alfabeto para denotar o conjunto de todas as cadeias que podem ser formadas por este alfabeto, incluindo ε . Se $\Sigma = \{a,b, 0, 1\}$ então $\Sigma^* = \{\varepsilon, a, b, aa,ab, a0, a1, ba, bb, b0, b1, 1a, 1b, 10, 11, aaa, aab, \dots\}$. Em adição, denotamos $\Sigma^+ = \Sigma^* - \varepsilon$, isto é, o conjunto de uma ou mais cadeias sobre Σ , enquanto que Σ^* é o conjunto de 0 ou mais cadeias sobre Σ .

Denotamos por linguagem formal, ou simplesmente linguagem, um conjunto de cadeias ou palavras sobre um alfabeto. Assim, dado o alfabeto $\Sigma = \{0,1\}$:

- a) O conjunto vazio e o conjunto formado pela palavra vazia são linguagens sobre Σ .
- b) O conjunto de todos os números pares sobre Σ é uma linguagem sobre Σ . Isto é, $110,101010, 11100$, são cadeias ou sentenças desta linguagem.

Note que Σ^* é uma linguagem sobre Σ .

Podemos definir uma linguagem utilizando expressões que utilizem os símbolos de um ou mais alfabetos, os operadores de união e concatenação, além dos parênteses balanceados e do símbolo ϵ . As expressões que utilizam estes símbolos são chamadas de expressões regulares (ER), e as linguagens definidas por uma ER são denominadas linguagens regulares (LR).

Exemplo:

$$\text{i) } (a \cup b)^* = \{a,b\}^* = \{\epsilon, a, b, aa, ab, aaa, aab, \dots\}$$

$$\text{ii) } a(a \cup b)^* = \{a, aa, ab, aaa, aab, aaaa, aaab, \dots\}$$

No exemplo i) temos todas as palavras formadas pelas letras do alfabeto $\{a,b\}$ e no exemplo ii) temos todas as palavras sobre $\{a,b\}$ que comecem com a .

3.5 AUTÔMATO FINITO

Um forte apelo ao estudo de uma máquina de estados finito ou simplesmente autômato finito é a facilidade com que podemos utilizá-los para representar diversos tipos de sistemas naturais ou artificiais, além de podermos representá-lo graficamente por um diagrama de transição de estados. Entre as aplicações computacionais temos como exemplo os Analisadores Léxicos e os Processadores de Texto, em que dada uma palavra e um estado inicial, a cada caractere lido a máquina muda para um novo estado informando basicamente o que já se tem de concreto sobre uma determinada palavra (se é um nome, um verbo, ou um tipo de operador) e quais os caracteres esperados na seqüência. Caso ocorra um erro a máquina irá para estado de erro.

Nem tudo pode ser representado através de estados. Imagine o cérebro humano que é composto de bilhões de células. Podemos, a princípio, representar cada neurônio por um número finito de bits, tendo as diversas combinações entre os neurônios como estados. No entanto, tal complexidade de solução não seria considerada eficiente.

3.5.1 Analisador Léxico como Autômato Finito

Podemos considerar que a função básica de um analisador léxico é “fazer a leitura do programa fonte, caractere a caractere, e traduzi-lo para uma seqüência de símbolos léxicos,

também chamados de tokens” (PRICE e TOSCANI, 2001). Vamos considerar o processo de reconhecimento de operadores relacionais na linguagem Pascal, baseando-se na Figura 3.2.

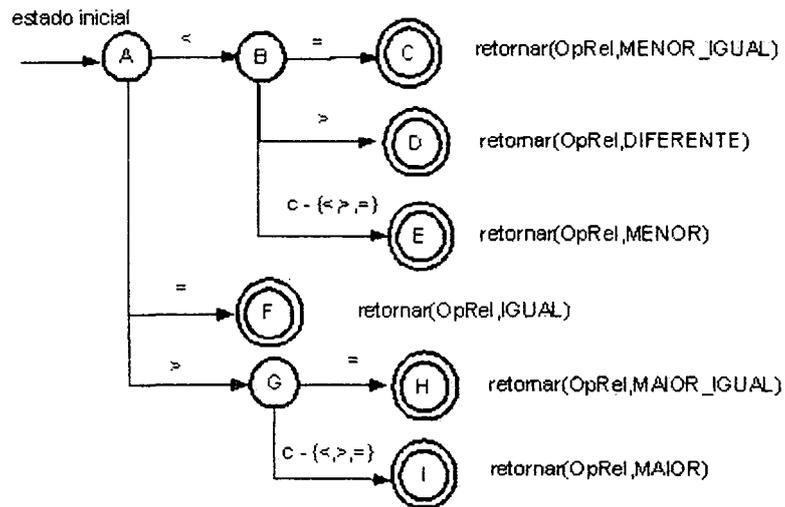


Figura 3.2 – Diagrama de transição para reconhecimento de operadores relacionais.

Nesta figura temos um diagrama de transição de estados, sendo que cada círculo representa um estado e a direção da seta indica a mudança para o próximo estado. Esta mudança é realizada pela leitura do rótulo da seta, denominado símbolo de entrada. Os estados duplamente circulados são considerados estados finais, isto é, estados que servem para identificar uma determinada palavra. No caso, o diagrama consiste de um reconhecedor de operadores relacionais. A análise começa com a leitura de um dos símbolos menor (<), igual (=) ou maior (>). Caso o símbolo lido inicialmente seja menor (<), ocorre uma transição para o estado B que representa então uma nova configuração, indicando que já leu o símbolo menor (<) e aguarda um próximo símbolo para identificar o tipo de operador. Caso seja igual (=), então ocorre a transição para o estado C, indicando que reconheceu um operador do tipo MENOR_IGUAL, sendo então realizado a ação de retorno equivalente. O processo segue de modo análogo caso seja lido o símbolo maior (>) ou o símbolo igual (=). Deste modo, podemos usar um diagrama de transição de estados como um modelo gráfico de um reconhecedor de palavras de uma linguagem.

Este reconhecedor é chamado de Autômato Finito (AF) ou máquina de estados finitos. Portanto, um AF é um reconhecedor de palavras de uma linguagem, respondendo ‘SIM’ se a palavra pertence à linguagem e ‘NÃO’ caso contrário.

É muito comum olharmos para um diagrama de transição de estados e o chamarmos grosseiramente de autômato. Um diagrama de transição de estados é apenas um auxílio visual para o AF.

A Figura 3.3 representa um AF como uma máquina composta por:

- uma fita finita de entrada, contendo em suas células símbolos de uma cadeia a ser reconhecida;
 - uma função de transição de estados, que determina para cada estado e entrada do alfabeto a transição para um novo estado;
 - um controle finito de estados composto pelo estado atual e um ponteiro (cabeça de leitura) para a fita de entrada, lendo os símbolos da fita da esquerda para a direita.
- O objetivo deste controle é classificar as sentenças lidas na fita em duas classes : as que são reconhecidas pelo AF e as rejeitadas.

Inicialmente, o controle está no estado inicial e o ponteiro aponta para o símbolo inicial da fita. O controle chama a função de transição para determinar o novo estado do controle. Caso não exista transição a máquina pára e a cadeia da fita é rejeitada. De outro modo, o controle passa para o novo estado e o ponteiro aponta para o próximo símbolo da fita. O processo se repete até que a cadeia seja rejeitada ou que o ponteiro aponte para o fim da fita. Caso a máquina consiga ler toda a cadeia da fita e o estado do controle seja um estado final, a cadeia é aceita.

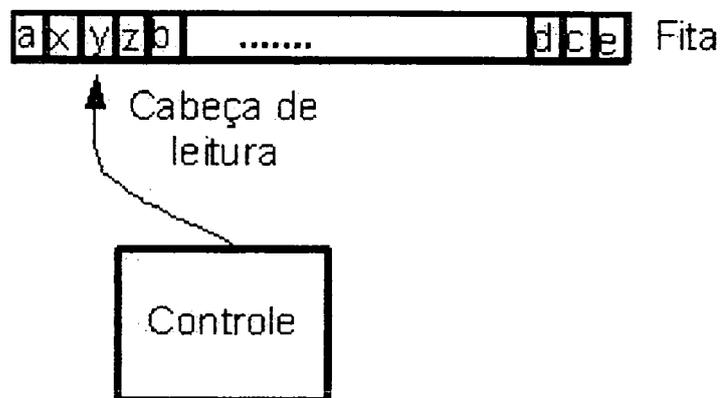


Figura 3.3 – Uma máquina de estados finitos

Formalmente, um AF¹¹ é uma tupla $M = (Q, \Sigma, \delta, q_0, F)$ no qual:

Q : Conjunto de estados

Σ : Alfabeto dos símbolos de entrada da fita

δ : Programa ou função de transição que é uma relação $\delta : Q \times \Sigma \rightarrow Q$, sendo δ uma função parcial¹².

q_0 : Estado inicial tal que $q_0 \in Q$.

F : Conjunto de estados finais tal que $F \subseteq Q$.

3.6 GRAMÁTICAS

No dicionário Michaels da língua portuguesa uma das definições de gramática é “Estudo sistemático dos elementos constitutivos de uma língua”. A noção de gramática teve origem no estudo da linguagem natural, buscando definir o conjunto de sentenças válidas de uma linguagem e estruturas para representá-la. Observe a frase: ‘O tigre feroz matou o búfalo velho’. A Figura 3.4 mostra a análise da frase.

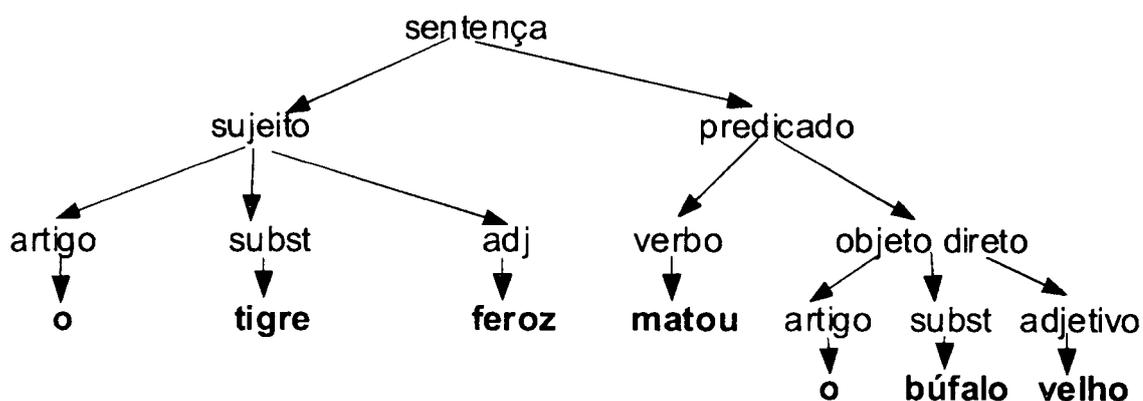


Figura 3.4– Árvore gramatical

¹¹ Chamamos de AF o Autômato Finito Determinístico definido em MENESES (2000), definição suficiente para o propósito desta dissertação.

¹² Uma função parcial é uma relação $f: A \rightarrow B$ tal que pode existir um elemento a de A que não tenha nenhum correspondente em B . Portanto, uma função parcial não necessita que todos os elementos de A tenha algum correspondente em B .

Embora esta análise esteja sintaticamente correta, nem sempre é fácil trabalhar com análise sintática de linguagens naturais, impossibilitando, com isso, a formalização de linguagens naturais. No caso das linguagens artificiais, estas são mais simples podendo ser então mais fáceis de representar, seguindo a mesma idéia da análise sintática das linguagens naturais. Conquanto não exista ainda uma gramática formal para linguagens naturais, o objetivo foi alcançado para as linguagens artificiais. Isso é devido ao fato de as linguagens artificiais terem uma sintaxe rígida e uma semântica bem definida.

Formalmente, uma gramática é uma quádrupla $G = (N, T, P, S)$, no qual:

T : Conjunto finito de símbolos terminais. Corresponde ao alfabeto de uma linguagem, e $T \cap N = \emptyset$.

N : Conjunto finito de símbolos não terminais ou variáveis.

P : Conjunto finito de pares (α, β) , denominadas regras de produção ou regra de reescrita, tal que:

$\alpha \in (T \cup N)^+$ e $\beta \in (T \cup N)^*$. Esta regra de produção (α, β) é representada por $\alpha \rightarrow \beta$, (lê-se: α deriva em β), o que significa que α pode ser substituído por β . O símbolo \Rightarrow , chamado de derivação imediata, é a aplicação da regra de produção para geração de uma sentença. Ou seja, $\alpha \Rightarrow \beta$, se e somente se $\alpha \rightarrow \beta$. Uma seqüência de regras da forma $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \dots, \alpha \rightarrow \beta_n$, pode ser simplificada como $\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$. Deste modo, diz-se que α pode ser substituído por β_1 , ou β_2 , ..., ou β_n .

S : Símbolo de N denominada variável inicial ou axioma.

Exemplo: Seja a gramática $G = (\{S, X, Y\}, \{a, b\}, P, S)$ no qual

$P = \{$ i) $S \rightarrow XY,$
 ii) $X \rightarrow \emptyset,$
 iii) $Y \rightarrow I \}$

Aplicando as regras de derivação temos:

i) ii) iii)
 $S \Rightarrow XY \Rightarrow \emptyset Y \Rightarrow \emptyset I$ que é a única cadeia gerada por esta gramática. No primeiro passo, S é substituído por XY , aplicando a regra i); a seguir X é substituído por \emptyset , aplicando a regra ii);

e finalmente Y é substituído por I , pela aplicação da regra iii). Como a cadeia gerada só tem símbolos terminais o processo pára e $0I$ é a cadeia gerada.

3.6.1 Hierarquia de Chomsky

Noam Chomsky classificou as gramáticas em quatro tipos, chamadas de ‘hierarquia de Chomsky’:

- a) Gramáticas não Restritivas (GNR) ou do tipo 0: são aquelas cujas produções P são do tipo
- $$\alpha \rightarrow \beta, \text{ no qual } \beta \in (N \cup T)^*$$
- $$\alpha \in (N \cup T)^+$$

Note que α é sempre diferente de ϵ .

- b) Gramáticas Sensíveis ao Contexto (GSC) ou do tipo 1: são aquelas cujas produções P são do tipo:

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2, \text{ no qual } \alpha_1, \alpha_2, \beta \in (N \cup T)^* \text{ e } \beta \neq \epsilon$$

$$A \in N$$

Aqui a denominação ‘sensível ao contexto’ pode ser vista, pois A é substituído por β se antes de A vier α_1 , e depois de A vier α_2 . Isto é, A é sensível, depende dos símbolos que vêm antes e depois.

- c) Gramáticas Livres de Contexto (GLC) ou do tipo 2: são aquelas cujas produções P são do tipo

$$A \rightarrow \beta, \text{ no qual } \beta \in (N \cup T)^*$$

$A \in N$, ou seja, do lado esquerdo da produção teremos somente um não terminal ou variável.

Neste caso, A não depende dos símbolos que vem antes ou depois.

- d) Gramáticas Regulares (GR) ou do tipo 3: são aquelas cujas produções P são do tipo

$$A \rightarrow \alpha B \qquad A \in N, B \in (N \cup \epsilon), \alpha \in T^*$$

O lado direito da produção contém no máximo um não terminal e, se existir, será o último símbolo.

3.6.2 Gramáticas Regulares e Autômatos Finitos

Podemos facilmente transformar um AF em uma GR. Considere o AF representado pelo diagrama de transição de estados da Figura 3.5.

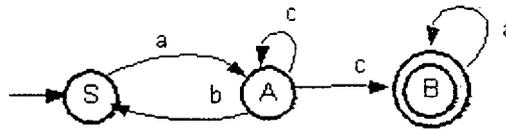


Figura 3.5 – Diagrama de Transição.

Neste exemplo, a GR equivalente seria:

$S \rightarrow aA$

$A \rightarrow cA$

$A \rightarrow bS$

$A \rightarrow cB$

$B \rightarrow aB$

$B \rightarrow \varepsilon$

Cada estado é considerado um símbolo não terminal da GR, sendo que o lado direito é escrito colocando a entrada seguida do novo estado determinado na função de transição. O estado final terá do lado direito uma das produções igual ε . A recíproca também é verdadeira, embora algumas vezes tal GR tenha que ser melhorada, como no caso em que temos mais de um símbolo terminal do lado direito.

Exemplo: No caso da seguinte regra:

$X \rightarrow 01Y,$

temos que transformar a produção em

$X \rightarrow 0X'$

$X' \rightarrow 1Y.$

Deste modo, temos uma relação entre GR e AF já que ambos podem então ser usados como de LR. A vantagem de um AF é que podemos facilmente criar um diagrama para um melhor entendimento.

3.6.3 Gramáticas estocásticas

A evolução de sistemas naturais nem sempre estão definidas através de regras rígidas, mas de algumas variantes probabilísticas. Criar então uma gramática para representação destes padrões necessita da definição em que a aplicação de uma produção depende de valores probabilísticos. Em nosso modelo, uma gramática estocástica $G = (N, T, P, S)$ será então definida como:

N : O conjunto de símbolos não terminais ou variáveis

T : O conjunto de símbolos terminais

S : É a variável inicial, no qual $S \in N$

P : É o conjunto de regras de produção tal que:

$(A, \alpha, p) \in P, A \in N, \alpha \in (N \cup T)^*$ e $0 \leq p \leq 1$, sendo p a probabilidade de α ocorrer.

Usaremos então a notação $A \rightarrow \alpha_i : p_i$, para denotar que A tem a probabilidade p_i de ser substituído por α_i , sendo que $\sum p_i = 1$, para toda derivação de A . Ainda nesta notação, caso a probabilidade não apareça admite-se por *default* que ela é 1.

Exemplo:

1)

1.a) $S \rightarrow A : 0.4$

1.b) $S \rightarrow B : 0.6$

2)

2.a) $A \rightarrow a : 0.3$

2.b) $A \rightarrow aB : 0.7$

3) $B \rightarrow b$

No exemplo acima, a regra 1 teria 40% de probabilidade de substituir S por A (1.a) e 60 % de substituir S por B (1.b). A regra 2 tem 30% de probabilidade de que A seja substituída por a e 70% de A ser substituída por aB . A regra 3 indica que B tem 100% de probabilidade de ser substituído por b .

3.6.4 Análise sintática utilizando gramáticas

Verificar sintaticamente uma sentença significa construir sua árvore de derivação, associando cada símbolo terminal da sentença ao seu não terminal correspondente na produção. Entre os diferentes métodos temos a análise descendente recursiva sem retrocesso (AHO, SETHI e ULLMAN, 1999). Este método evita tentativas repetitivas de reconhecer uma entrada. Para que isso seja possível a gramática não pode ser recursiva à esquerda. Felizmente podemos transformar uma gramática recursiva à esquerda em não recursiva à esquerda.

Consideremos a gramática abaixo:

Gramática 1.

$$(1) S \rightarrow [L] | a$$

$$(2) L \rightarrow L ; S / S$$

A gramática 1 é recursiva à esquerda. Considere a cadeia ' $[a;a;a]$ '. Para que possamos gerar a sentença inicial temos que decidir sobre qual das produções de (1) podemos associar à sentença. Se substituirmos S por ' a ', termina a geração já que a é uma sentença composta de símbolos terminais e não poderemos gerar a cadeia. Retornamos então e substituímos S por ' $[L]$ '. Deste modo, associamos o primeiro e último símbolo da cadeia com os símbolos terminais '[' e ']'. Logo, temos que mostrar que podemos gerar ' $a;a;a$ ' a partir de L . Se substituirmos L por S em (2), temos que substituir S por ' a ' ou ' $[L]$ ' e nada conseguimos. Retornamos então e substituímos L por ' $L;S$ ' em (2). O processo continua até chegarmos a uma árvore cujas folhas são símbolos terminais e correspondem seqüencialmente aos símbolos da cadeia analisada.

No entanto podemos evitar o retrocesso passando toda a recursividade da gramática à direita. A gramática 2 corresponde à gramática 1 sem recursividade à esquerda.

Gramática 2.

$$(1) S \rightarrow [L] | a$$

$$(2) L \rightarrow SL'$$

$$(3) L' \rightarrow ;SL' / \epsilon$$

Na gramática 2, as sentenças substitutivas de (1) e (3) começam sempre com símbolos terminais, evitando assim a dúvida na substituição. Repare que em (2) não há escolha. Deste modo, $[a;a;a]$ segue a seguinte seqüência de derivação em 8 iterações.

$$\begin{array}{cccccccc} (1) & (2) & (1) & (3) & (1) & (3) & (1) & (3) \\ S \rightarrow [L] \rightarrow [SL'] \rightarrow [aL'] \rightarrow [a;SL'] \rightarrow [a;aL'] \rightarrow [a;a;SL'] \rightarrow [a;a;aL'] \rightarrow [a;a;a] \end{array}$$

Abaixo, o algoritmo 1 realiza uma análise descendente recursiva sem retrocesso para a gramática 2. Neste algoritmo, *lookahead* é um ponteiro para o símbolo da cadeia e as funções $S()$ e $L_()$ ‘olham’ sempre para *lookahead* para decidir qual função chamar. A função *reconhece* verifica se o símbolo lido em *Cadeia* é o símbolo esperado, retornando em erro caso contrário. Perceba que $S()$ corresponde a S , $L_()$ corresponde a L , e $L'_()$ corresponde a L' .

Algoritmo 1. Analisador Sintático

Entrada : Cadeia;

Início

lookahead := PrimeiroCaracter(Cadeia);

$S()$;

Se (lookahead = FIMDECADEIA) retorne "SUCESSO...";

Senão Escreva("Era esperado fim de cadeia...");

Fim;

reconhece(c)

Início

```
Se (lookahead = c) lookahead := proximocaracter(Cadeia);
```

```
Senão ERRO(c);
```

```
Fim;
```

```
Erro(str)
```

```
Inicio
```

```
Escreva("ERRO...Era esperado", str);
```

```
PararExecucao;
```

```
Fim
```

```
S()
```

```
Inicio
```

```
Se(lookahead = '[')
```

```
reconhece('[');
```

```
L();
```

```
reconhece(']');
```

```
FimSe;
```

```
Senão Se(lookahead = 'a') reconhece('a');
```

```
Senão ERRO("[ ou a");
```

```
Fim;
```

```
L()
```

```
Inicio
```

```
S();
```

```
L_();
```

```
Fim
```

L_()

Inicio

```
Se(lookahead = ':')
    reconhece(':');
    S( );
    L_( );
FimSe;
```

Fim;

3.7 GRAMÁTICAS GRÁFICAS

Diferente das gramáticas anteriores, em uma gramática gráfica cada caractere de uma cadeia final gerada corresponde à uma instrução gráfica. Isto possibilita o desenho, movimentação, rotação e translação de objetos, entre outros. Tais gramáticas são muito utilizadas na construção de cenas de paisagismo, geração de fractais e jogos. Apresentaremos a seguir o modelo de Lindemayer e uma L-System estocástica proposta por PRUSINKIEWICZ (1986).

3.7.1 Gramáticas L-System

Em 1968, o biólogo Aristid Lindemayer propôs um formalismo matemático para o entendimento e representação da evolução ou crescimento de sistemas biológicos. A essa formalização deu-se o nome de L-System. Um L-System utiliza símbolos e regras para dar uma interpretação visual e geométrica da evolução dos sistemas.

Diferentemente das gramáticas de Chomsky, no qual as produções são aplicadas sequencialmente, em um L-System todas as variáveis de uma sentença são substituídas em paralelo, de modo simultâneo. É similar aos sistemas biológicos no qual várias células sofrem divisões ao mesmo tempo.

Entre as aplicações temos a geração de fractais e o modelamento realístico de plantas e células neurais.

Uma gramática L-System (LS) é uma tupla $L = (N, T, P, S)$, no qual:

N : São os símbolos não terminais ou variáveis, do mesmo modo que as gramáticas de Chomsky;

T : São os símbolos terminais ou constantes, sendo que T pode ser vazio;

P : É o conjunto de regras de produção;

S : São as sentenças que definem a geração inicial de uma sentença, ou axioma, sendo que $S \in (N \cup T)^+$.

Note que, diferentemente das gramáticas de Chomsky, S pode ser uma sentença contendo tanto símbolos terminais como não terminais. A sentença final gerada pode conter ainda símbolos não terminais dependendo da quantidade de iterações a serem aplicadas. Quando uma variável não está definida toma-se por *default* que ela deriva nela mesma.

Exemplo:

Seja o seguinte LS $L = (\{A,B\}, \emptyset, \{A \rightarrow B, B \rightarrow AB\}, AB)$

Partindo de $S = AB$, seguiremos substituindo cada variável por suas respectivas produções. Portanto, com três iterações teríamos:

AB

1ª Iteração) **BAB**

2ª Iteração) **ABBAB**

3ª Iteração) **BABABBAB**

Lindemayer utilizou inicialmente os símbolos $\{F,f, +, -, [,]\}$ sendo que cada símbolo corresponderia à uma ação a ser realizada. Como exemplo, vamos definir a seguinte L-System:

Gramática 3.

Axioma: **X**

X \rightarrow **XFY**

Y \rightarrow **[-F][+F]**

Produções em 4 iterações:

1ª iteração: XFY

2ª iteração: $XFYF[-F][+F]$

3ª iteração: $XFYF[-F][+F]F[-F][+F]$

4ª iteração: $XFYF[-F][+F]F[-F][+F]F[-F][+F]$

Cadeia resultante: $FF[-F][+F]F[-F][+F]F[-F][+F]$

A cadeia resultante surge após a última iteração, no qual todos os símbolos não terminais são retirados. Os símbolos terminais são interpretados de acordo com as seguintes funções relacionadas a cada caracter, para um certo ângulo θ .

F : Desenha uma reta para frente um passo na direção corrente.

f : Move a caneta para frente um passo na direção corrente.

+ : Adiciona θ ao ângulo corrente.

- : Subtrai θ ao ângulo corrente.

[: armazena as informações correntes em uma pilha.

] : restaura as informações armazenadas.

A Figura 3.6 mostra o desenho correspondente a cadeia resultante da gramática 1 após quatro iterações, e um θ de 45° .

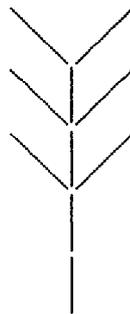


Figura 3.6 – Ramificação gerada após quatro iterações da gramática 3

O poder da L-System está na possibilidade de se adicionar novos símbolos terminais com as ações que desejar. Em alguns casos F pode ser tanto uma variável como um símbolo terminal, de modo que não seja desprezado na cadeia final.

3.7.2 L-System Estocástica

Podemos associar a cada produção uma probabilidade de execução da produção, conforme exemplo da gramática 4. Veja que na produção p_2 Y tem 60% de chance de ser reescrito como F. Note que a soma das probabilidades das regras de produção de um mesmo não terminal deve ser igual a 100%.

Gramática 4:

Axioma : X

$$p_1: X \rightarrow YFX \quad (1.0)$$

$$p_2: Y \rightarrow F \quad (0.6)$$

$$p_3: Y \rightarrow X[-FFX] \quad (0.2)$$

$$p_4: Y \rightarrow X[+FFX] \quad (0.2)$$

Gramáticas sem probabilidade sempre produzem imagem igual para um certo número de iterações. No entanto, organismos biológicos não se comportam de modo determinístico, pois estão sujeitos à influência de inúmeras variáveis. Para gerar plantas artificiais PRUSINKIEWICZ (1986) propõe o uso de L-System cujas regras de substituição contém valores probabilísticos. Deste modo, a gramática pode gerar diferentes ramificações. Basta a cada iteração gerar um número aleatório que corresponda a probabilidade de uma das produções. A Figura 3.7 mostra alguns dos possíveis resultados da execução da gramática 4 após 20 iterações.

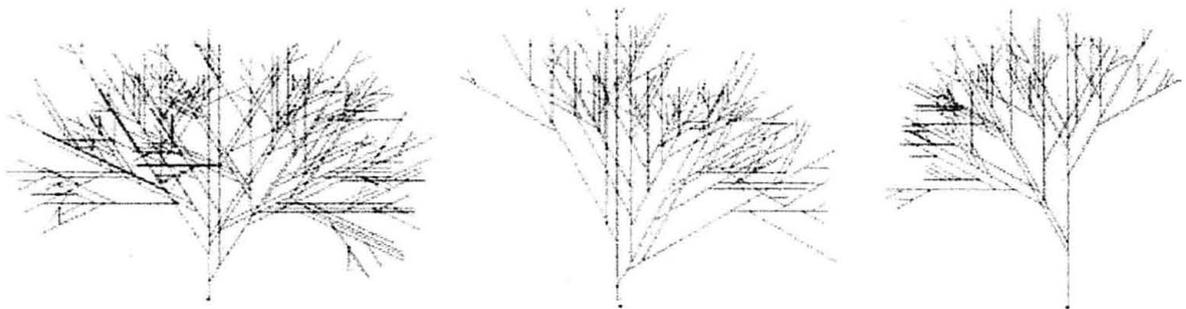


Figura 3.7 – Árvores geradas utilizando a gramática 4.

^: gira a tartaruga em torno do eixo y de um ângulo $-\delta$
/: gira a tartaruga em torno do eixo x de um ângulo δ
|: gira a tartaruga de 180° em torno do eixo z
!: decreta a largura atual da linha de acordo com a fórmula $w' = w - w_inc$
' : incrementa a cor atual de acordo com a fórmula $c' = c + c_inc$
{: armazena as posições subseqüentes da tartaruga como vértices de um polígono a ser preenchido
}: preenche o polígono usando a cor atual c
[: armazena estado atual da tartaruga em um pilha
]: recupera o último estado colocado na pilha e o faz como estado atual da tartaruga, porém sem desenhar nenhuma linha.

Desse modo, após uma série de iterações teríamos o ramalhete da Figura 3.8.

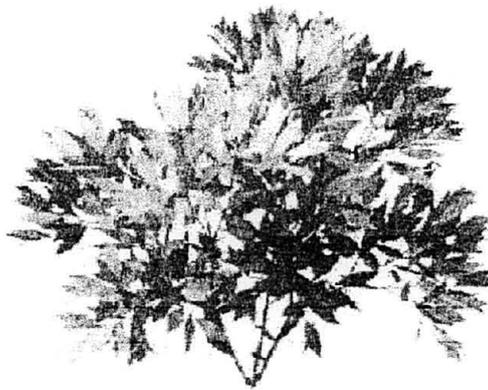


Figura 3.8 – Um ramalhete gerado pela gramática 5 de PRUSINKIEWICZ (1986).

A aplicação das LS estocásticas em áreas como paisagismo é óbvia, pois caso quiséssemos criar uma floresta bastaria aplicar uma gramática geradora de árvores repetidamente até se conseguir o desejado. As árvores desta floresta não seriam iguais, aproximando, então, o desenho da realidade.

HAMILTON (1993) propõe o uso de gramáticas L-System para a simulação do crescimento neural, uma vez que os neurônios apresentam uma estrutura ramificada. Posteriormente, McCORMICK e MULCHANDANI (1994) apresentam uma gramática para geração de células neurais ou qualquer outra estrutura utilizando o conceito de torção e curvatura, para geração de células 3D.

3.8 ALGUMAS NOTAÇÕES DE TRAÇADO NEURAL

Atualmente temos uma grande quantidade de formatos utilizados para representação de traçados neurais. Mostraremos a seguir notações que tratam as ramificações de um neurônio como um grafo. Cada aresta deve ser vista como um cano cujos extremos podem não ter a mesma espessura, e cada vértice define um ponto de início de ramificação ou de continuidade. Desse modo, conhecendo a notação adotada, podemos gerar novamente a disposição espacial da célula neural.

3.8.1 Formato EUTECTIC

A Figura 3.9 apresenta a ramificação de um neurônio no formato EUTECTIC. Nesta notação chamamos de BP (*branch point*) o ponto de ramificação, MTO (*middle tree origin*) o ponto de origem de uma ramificação que parte do soma, NE (*normal end*) o final da ramificação e CP (*continuous point*), os pontos de continuidade.

Um segmento dendrítico é considerado o intervalo entre um MTO e BP, BP e BP, BP e NE, ou ainda entre MTO e NE, no caso de não ocorrer ramificação. Os *arcos* são os intervalos entre cada ponto, seja ele qual for. Cada segmento pode ser composto de um ou mais arcos.

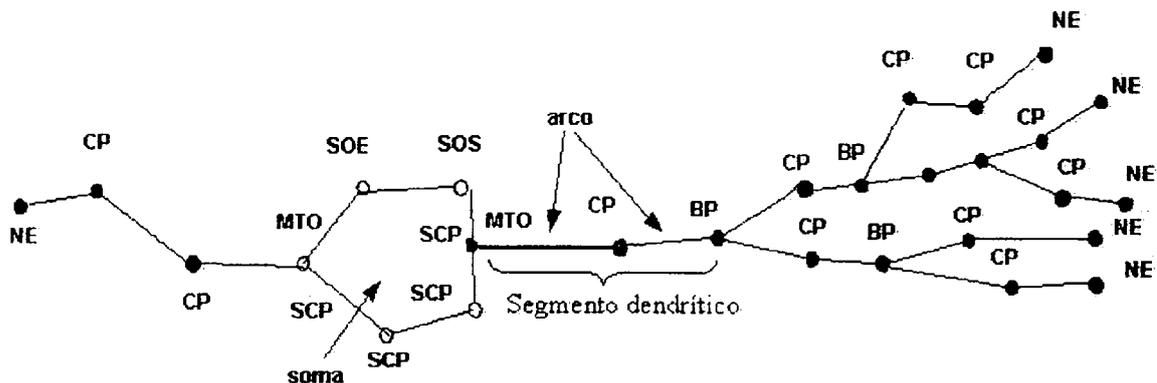


Figura 3.9 – Ramificações no formato EUTECTIC.

O primeiro segmento dendrítico, isto é, aquele que parte do MTO até um BP pertence ao nível 1, incluindo o BP e todos os arcos do segmento. O segundo segmento pertence ao nível 2, e assim sucessivamente.

Além destes, temos os pontos do soma, SOS (*soma outline start*), SCP (*soma continuation point*), e SOE (*soma outline end*).

A Figura 3.10 mostra um trecho de um arquivo em notação EUTECTIC. O campo *Point* serve simplesmente para identificar o ponto, linha a linha, sendo autoincrementado. *Tag* pode ser usado para identificar o tipo da ramificação (se um ponto pertencente ao dendrito, axônio, ou mesmo soma). Os valores x, y, z correspondem às coordenadas do eixo cartesiano. Uma vez que cada segmento pode ser visto como tubos, *thick* é o diâmetro do tubo naquele ponto.

No. points	2692					
No. trees	3					
Comentario						
Point	Type	Tag	X	Y	Z	Thick
1	MTO	1	5.4	-1.5	9.1	0.710
2	CP	3	10.1	-5.2	16.9	0.000
3	BP	3	10.3	-10.0	16.9	0.710
4	CP	3	10.2	-8.6	18.2	1.340
5	CP	3	11.2	-10.5	18.2	1.340
6	BP	3	11.2	-10.4	18.2	1.340
7	CP	3	11.9	-12.7	17.3	0.550
8	CP	3	12.7	-13.0	17.3	0.550
11	CP	3	15.3	-14.8	19.4	0.550
15	CP	3	12.1	-19.7	14.6	0.550
16	CP	3	12.8	-23.0	14.4	0.550
19	CP	3	11.6	-26.2	19.9	0.550
20	CP	3	10.7	-31.5	19.9	0.550
21	NE	3	10.7	-31.6	19.9	0.550
22	CP	3	11.2	-10.4	18.2	0.550
23	CP	3	14.1	-11.4	18.4	0.550

Figura 3.10 – Trecho de arquivo em notação EUTECTIC.

3.8.2 Notação SWC

Muitos arquivos de neurônios piramidais encontram-se na internet disponibilizados no formato SWC. Este formato permite obter o diâmetro inicial e final de um arco possibilitando, assim, uma representação bastante realística do neurônio observado.

Esta notação também trata um neurônio como um grafo e usa o conceito de lista indexada. Os dados de cada ponto utilizam o seguinte formato:

$$n T x y z R P$$

Aqui, n é um inteiro que serve para identificar o ponto referido. Este número é autoincrementável para cada novo nodo anotado. T é um valor que identifica o tipo de ponto de acordo com a Tabela 3.1.

Tabela 3.1 – Tipos de pontos no formato SWC

Valor (T)	Significado
0	Indefinido
1	Soma
2	Axônio
3	Dendrito
4	Dendrito apical
5	Bifurcação
6	Ponto final

O valor 0 significa que a pessoa que estava realizando a notação ficou em dúvida sobre o que significa aquele ponto. Muitos usam também o valor -1 para representar indefinição. O valor 3, no caso dos neurônios piramidais, é considerado dendrito basal e simplesmente dendrito nos demais casos.

Por se tratar de uma notação tridimensional x , y , z são as três coordenadas cartesianas do ponto. R representa o diâmetro do ponto naquela posição. P é um ponteiro cujo índice indica o número do ponto adjacente anterior, ou antecessor. Como estamos utilizando a notação de grafos, cada arborização partindo do soma pode ser considerada uma árvore. Logo, podemos dizer que P é o ponteiro cujo índice aponta para o pai do nodo atual. Quando o ponto não tem antecessor, P tem valor -1 .

A notação SWC contém geralmente um arquivo de cabeçalho que começa com o caractere '#'. Este caractere significa que a linha é uma linha de comentários. A Figura 3.11 mostra parte de um arquivo em formato SWC. Para ajustar erros decorrentes do processo de notação nas diferentes profundidades, a notação SWC utiliza os fatores de correção SHRINKAGE CORRECTION, cujo primeiro valor deve multiplicar o eixo x , o segundo o

eixo y e o terceiro o eixo z, corrigindo os erros de traçado. No exemplo da Figura 3.11, multiplicaríamos x, y e z por 1.13, 1.13 e 4.0, respectivamente.

```
# CREATURE rat F344
# REGION Hippocampus
# FIELD/LAYER CA1
# TYPE CA1 Pyramidal Cell in vitro young
# CONTRIBUTOR Turner_DA
# REFERENCE J. Comp. Neurol. 391: 335-352, 1998
# RAW N120.asc
# EXTRAS Turner_P.CA1
# SOMA_AREA 1.28E3
# SHRINKAGE_CORRECTION 1.13 1.13 4.0
# VERSION_NUMBER 2.0
# VERSION_DATE 1998-03-27
# *****
# SCALE 1.13 1.13 4.0
#
# original source format: Neurolucida
1 1 -3.39 0.181 4.96 8.119 -1
2 1 -1.785 -0.35 4.96 7.96 1
3 1 1.074 -0.701 4.96 7.96 2
4 1 4.814 -1.051 4.96 7.8 3
5 1 5.435 -1.456 32.385 1.146 4
6 3 10.125 -5.198 59.84 0.71 5
7 3 10.328 -10.012 59.84 0.71 6
```

Figura 3.11 – Trecho de um arquivo SWC

3.8.3 Formato NEUROLUCIDA

NEUROLUCIDA¹³ é na verdade um software acoplado a um microscópio que auxilia na leitura de diferentes pontos do tecido neural, guardando esta notação em um arquivo com extensão ASC¹⁴ e com um código bastante distinto das outras notações conhecidas.

Nesta notação, todo o histórico da notação é armazenado como documentação. Os pontos ignorados ou removidos são armazenados, permitindo, com isso, um levantamento histórico da pesquisa. NEUROLUCIDA permite ainda marcação de pontos que o pesquisador considere importante destacar como cones de crescimento e ramificações.

No entanto, há poucas informações sobre esta codificação, mesmo na própria página do software. As poucas informações encontradas até o momento foram obtidas da página

¹³ <http://www.microbrightfield.com>

¹⁴ O nome dos arquivos em notação neurolucida apresentam o formato *nomearquivo.asc*

http://www-ibcm.unil.ch/staff/tettoni/maxsim/file_formats.html que propõe a transformação da notação NEUROLUCIDA para um formato próprio.

Na notação NEUROLUCIDA, cada linha pode ser vista como instruções ou informações para uma *turtle* de modo a orientar o software sobre o que fazer e apresenta o seguinte formato:

[I, N] (x,y,z) r

I é um número que indica se o tecido é de soma, dendrito, ou uma nova ramificação. Além disso, também pode fornecer uma instrução a *turtle*. N auxilia numa melhor definição desta instrução ou informação. Os valores x, y e z são as coordenadas cartesianas, e r o diâmetro do arco.

Na Tabela 3.2 temos alguns exemplos do funcionamento da notação e na Figura 3.12 um trecho de um arquivo nesta notação, com comentários em francês.

Tabela 3.2 – Linhas de código NEUROLUCIDA

Informação ou Ação	Comentário	Código	Exemplo
Mover para o ponto		[2, impar]	[2, 21] (1.11, 7.65, 2.50) 1.49
Traçar segmento	Da posição corrente até o ponto. X pode ser par ou ímpar.	[1, X]	[1, 21] (10.11, 7.00, 2.50) 1.49
Nova junção ou nodo. Indica o tipo do segmento (axônio, dendrito, dendrito apical, etc.)	É sempre composto por um par de linhas. A instrução [10, impar] apenas indica um novo ponto. Em seguida vem a instrução para mover para o ponto. Aqui, impar é diferente de 5.	[10, impar] [2, impar]	[10,21] (10.11, 7.00, 2.50) 1.49 [2,21] (10.11, 7.00, 2.50) 1.49
Ponto de Bifurcação		[10,5]	[10,5] (-1394.91, 429.45, 1554.86) 0.90
Fim de ramificação	X é um dígito, e deve ser seguido da instrução para traçar até o ponto, na próxima linha.	[10, 10X]	[10,104] (-1395.52, 433.35, 1559.27) 0.90 [1,22] (-1395.52, 433.35, 1559.27) 0.90
Remoção de código		[33,X]	[33,104] (1959.45, -4167.16, 34.14) 1.49

```

version 3          -- Neurolucida format numéro trois
file id: ba16c.dat    -- Nom du fichier
MAX:(8183.31, 3109.71, 1585.71) -- Parallélépipède circonscrit
MIN:(-8188.81, -10528.38, -156.81)
filetype: 0          -- Non utilisé
id1: 0
id2: 0
REF:(0.00, 0.00, 0.00)
rotation: -1489.16
gx_alpha: 0
gx_beta: 0
gx_gamma: 0

[2,21] (1969.57, -4252.87, 1.18) 1.49 -- Déplacement vers le début d'un axone
[1,22] (1969.19, -4245.24, 1.18) 1.49 -- Traçage axone rayon 1.49µ
[1,22] (1959.67, -4187.12, 27.42) 1.49 -- suite...
[1,22] (1960.65, -4184.78, 29.86) 1.49 -- suite...
[33,104] (1959.45, -4167.16, 34.14) 1.49 -- Point suivant traverse sommet de coupe
[1,22] (1959.45, -4167.16, 34.14) 1.49 -- Tracage vers ce point
[32,12] (0.00, 50.00, 64.05) 0.29 -- ? Non documenté
[1,22] (1956.48, -4164.82, 34.14) 1.49 -- suite...
[1,22] (1946.36, -4096.72, 64.15) 1.49 -- suite...
[10,5] (-1394.91, 429.45, 1554.86) 0.90 -- Le point suivant est une bifurcation
[1,22] (-1394.91, 429.45, 1554.86) 0.90 -- Traçage vers ce point, rayon 0.9µ
[10,104] (-1395.52, 433.35, 1559.27) 0.90 -- Point suivant termine le segment

```

Figura 3.12 – Trecho de arquivo com notação NEUROLUCIDA. Após o hífen (-), vemos comentários em francês.¹⁵

3.8.4 Formato DAT

Esta notação é o formato de arquivo utilizada pela DLL do visualizador tridimensional. Na Figura 3.13 temos parte de um arquivo em notação DAT. Na primeira linha aparece as coordenadas cartesianas da posição do soma no espaço e o raio do soma (nesta DLL o soma é considerado uma esfera) e um número correspondente à textura do soma. Nas próximas linhas, cada arco dendrítico ou axonal é considerado um cilindro, contendo as coordenadas iniciais, finais, o raio do cilindro e o número da textura que cobrirá este cilindro. A grande vantagem é que podemos gerar arquivos .dat provindos de outros arquivos manipulando a cor da textura para diferentes tipos de segmentos dendríticos. Neste

¹⁵ http://www-ibcm.unil.ch/staff/tettoni/maxsim/file_formats.html

trabalho estamos utilizando texturas de cores vermelha para dendritos basais e azul para apicais.

0.00,	0.00,	0.00,	1.00,	4,			
0.80,	-3.95,	-7.64,	0.24,	-4.01,	-7.06,	5.19,	3,
0.24,	-4.01,	-7.06,	-1.16,	-4.95,	-7.12,	1.80,	3,
-1.16,	-4.95,	-7.12,	-2.95,	-6.20,	-7.18,	1.80,	3,
-2.95,	-6.20,	-7.18,	-3.71,	-7.63,	-7.24,	1.80,	1,
-3.71,	-7.63,	-7.24,	-3.18,	-20.16,	-5.96,	1.80,	1,
-3.18,	-20.16,	-5.96,	-0.00,	-40.34,	-4.63,	1.80,	1,
-6.90,	-40.34,	-4.63,	-6.37,	-40.34,	-4.63,	1.80,	1,
-6.37,	-40.34,	-4.63,	-9.55,	-56.14,	-2.82,	1.80,	1,

Figura 3.13 – Trecho de um arquivo de traçado DAT.

3.9 ALGUNS ALGORITMOS PARA GERAÇÃO DE CÉLULAS NEURAIS

A representação de células neurais através de diferentes modelos tem contribuído grandemente para a evolução de pesquisas relacionadas ao sistema nervoso. Diferentes modelos e notações de traçado são utilizados para representar células neurais. No entanto, a maioria destes modelos não se preocupa com a forma e nem com o padrão de crescimento destes neurônios.

ASCOLI e KRICHMAR (2000) propõem o uso de L-System estocásticas para representar a morfologia neural e desenvolveram um programa, o L-Neuron (LN), que utiliza os algoritmos de Hillman, Tamori e Burke para geração de neurônios morfologicamente distintos de várias classes neuronais.

O algoritmo de Hillman define alguns parâmetros, chamados de ‘fundamentais’, como medidas suficientes para descrever completamente o crescimento dendrítico. Esta regra aproveita-se de uma série de correlações locais, tais como as ‘regras de potência’ de Rall ou *Rall Power*. Esta regra baseia-se na fórmula $d_p^v = df_1^v + df_2^v$, sendo d_p o diâmetro do tronco que sofre bifurcação e df_1 e df_2 o diâmetro dos galhos filhos que saem da bifurcação, e v é o expoente calculado ou *Rall Power*, devendo ser maior do que 0 e menor do que 3. Para que possamos obter v utiliza-se o método da força bruta.

Na descrição de Hillman um dendrito começa com um “diâmetro inicial” (um parâmetro fundamental), crescendo até um certo tamanho e reduzindo seu diâmetro de acordo com uma taxa de redução. Se o diâmetro desse segmento for maior que um certo limiar o

ramo irá então se bifurcar produzindo dois novos galhos cujos diâmetros serão completamente determinados pela *Rall Power*. Estes dois novos galhos continuarão o processo, crescendo e se afinando até que seu diâmetro seja inferior ao valor limiar, tornando-se então um galho terminal. Na proposta de Hillman, alguns poucos parâmetros são suficientes para descrever morfológicamente o crescimento dendrítico. Hillman afirma ainda que os ângulos de bifurcação podem ser adicionados como parâmetros fundamentais medidos de dados experimentais.

Tamori introduz o conceito de volume efetivo como parâmetro fundamental para calcular os ângulos de bifurcação e os diâmetros.

O algoritmo 2 mostra a implementação de Hillman/Tamori.

Algoritmo 2

Início;

Sorteie N o número de árvores partindo do soma;

Seja D_1, D_2, \dots, D_n os diâmetros dos galhos iniciais de cada árvore

Seja $L = [L_{\min}, L_{\max}]$ o intervalo de comprimento do arco

Seja $B = [B_{\min}, B_{\max}]$ o intervalo entre os ângulos de bifurcação

Seja T o diâmetro mínimo aceitável;

Seja $V = [V_{\min}, V_{\max}]$ o intervalo da Rall Power;

HILLMAN(L, B, T, V, D_1, \dots, D_n);

FimAlgoritmo;

Proc HILLMAN((L, B, T, V, D_1, \dots, D_n);

Para cada D_i faça

Repita

Sorteie um tamanho de arco $I \in L$;

$D = I * TaxaDeAfinamento$; //ou taxa de encolhimento

Se $D \geq T$

dp := D;

Sorteie $d_1 \in (0, dp)$; //o diâmetro do 1o galho filho

```
Sorteie  $\alpha \in B$ ;  
Sorteie  $v \in V$ ;  
 $d2 := (dp^v - d1^v)^{1/v}$ ; //diâmetro do 2o galho filho  
HILLMAN(L, B,T,V,d1,d2); //Chamada recursiva  
FimSe;  
Até ( $D < T$ );  
 $D := I * TaxaDeAfunilamento$ ;  
I é galho terminal;  
FimPara;  
FimProc;
```

Por último, o algoritmo de Burke trabalha com medidas estatísticas de ângulos e diâmetros. LN utiliza os três algoritmos e adiciona alguns outros parâmetros de correção destes algoritmos além de adicionar a possibilidade de simulação de tropismo, que imita o efeito de fatores neurotróficos e cones de crescimento.

Em outras palavras, Hillman mede (obtem de dados experimentais) os ângulos e calcula os diâmetros através da *Rall Power*. Tamori calcula diâmetros e ângulos e Burke mede tanto ângulos como diâmetros.

Resumindo, L-Neuron utiliza-se de gramáticas L-Sytem, das funções estatísticas e das regras dos três algoritmos para a geração de neurônios morfologicamente distintos.

4 O SIMULADOR E VISUALIZADOR NEURAL

Com base nos conceitos já definidos anteriormente, propomos a síntese e modelagem de células neurais artificiais utilizando gramáticas gráficas estocásticas. Inicialmente definimos como seria a sintaxe de uma gramática gráfica capaz de gerar estruturas fragmentadas como plantas, fractais e neurônios. Uma vez definida a sintaxe, criamos um interpretador para estas gramáticas. Este interpretador faz a análise sintática e léxica da gramática e a seguir gera uma cadeia de caracteres e as imagens correspondentes. Na segunda fase, inicialmente criamos um programa para converter os arquivos de traçados de neurônios do formato NEUROLUCIDA e SWC para o formato EUTECTIC. A seguir, extraímos as descrições estatísticas destes neurônios e criamos funções de distribuições relativas a diferentes medidas morfológicas, tais como: tamanho de um segmento dendrítico, ângulo de inclinação e de rotação em relação ao segmento anterior, entre outras. COELHO(1998) e TANAKA(2000) chamam tais ângulos de ângulos de ‘curvatura’ e ‘torção’, uma analogia ao conceito de torção e curvatura da geometria diferencial, sendo também citado para o uso em gramáticas gráficas por McCORMICK e MULCHANDANI (1994). O cálculo destes ângulos será detalhado na seção 4.4.

Em adição à proposta de ASCOLI e KRICHMAR (2000), substituímos algumas funções de distribuição (FD), por funções polinomiais de base radial, as *Thin Plate Splines* (TPS), mais apropriadas para simulações, ao invés de FD, e adicionamos o conceito de torção e curvatura. Isto é, ao invés de somente usarmos FD relacionadas às diferentes características da morfologia neural, substituímos algumas destas FD por TPS equivalentes. Usamos ainda um conjunto de funções disponíveis em uma DLL, que é detalhada na seção 4.3. Esta DLL foi desenvolvida com o propósito de visualização tridimensional das células neurais. Com isso, poderemos realizar operações de rotação, translação e escala, controlar operações que utilizem parâmetros como intensidade de luz, transparência e incluir diferentes texturas.

4.1 O INTERPRETADOR GRAMATICAL

A gramática que desenvolvemos utiliza os símbolos terminais F,+,-,[,], já comentados no capítulo 2. É permitido que o usuário altere os dados de tamanho do segmento

inicial, ângulos e axioma. Desse modo, podemos ter mais liberdade na geração destas imagens.

No modelo criado, as produções gramaticais podem ser entradas em tempo de execução ou lidas de um arquivo. Os símbolos não terminais são definidos no momento da declaração das regras de reescrita, ficando estes à esquerda, enquanto as produções ficam à direita separadas por um sinal de igual (=). A ausência de valor probabilístico indica que a probabilidade é 1. Caso exista definição de probabilidade, após a produção ocorre um sinal de dois pontos (:), indicando que serão anexados valores probabilísticos. Os comentários são limitados por chaves ({ e }) em qualquer posição da linha. Depois de declaradas as regras da gramática, o interpretador identifica os símbolos não terminais e verifica se não há nenhum erro sintático. Como erro sintático considera-se colchetes e chaves não balanceados, falta de sinal de igual e a soma das probabilidades das regras de produção de um não terminal diferente de 1, além de símbolos não definidos. Note que os símbolos não terminais estão pré-definidos, ou seja, existem regras associadas a eles. Na Figura 4.1 temos uma árvore gerada em formato de arame utilizando as notações definidas pelo modelo comentado. Caso quiséssemos uma planta mais realista, poderíamos adicionar a cada galho final uma ‘folha’ utilizando curvas de Bézier, como no ramalhete da Figura 3.8 do capítulo 3.

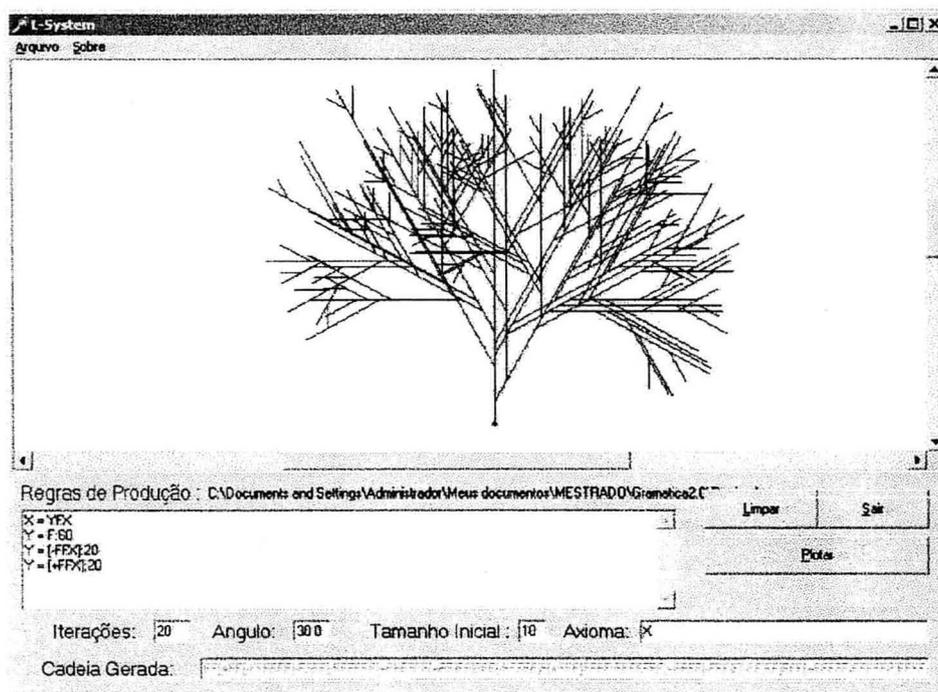


Figura 4.1 – Árvore gerada pelo interpretador utilizando gramática. A figura foi gerada utilizando a gramática 5 do capítulo 3.

Em uma das experiências, adicionamos um ponto de atração cuja influência variava de acordo com uma constante. Usamos uma constante K multiplicada pelo inverso do quadrado da distância euclidiana a este ponto (k/d^2). Com isso, podíamos simular a variação de ângulo e de tamanho que uma determinada ramificação sofria de acordo com a distância e a constante de atração. Tal experimento foi útil para simular o desenvolvimento biológico de um organismo que sofre influências de atração e repulsão do meio. Ou seja, podemos, através de fórmulas matemáticas, representar os fenômenos de tropismo causado nestas células.

Posteriormente, adaptamos o modelo de força de atração descrito em COELHO e COSTA (2002), conforme a Figura 4.2. Temos na figura um ponto P sofrendo atrações de vários pontos. Deste modo, calcula-se um vetor resultante $f = kL^\alpha \text{Versor}(\sum v_i d_i^{-\beta})$, sendo k uma constante correspondente à velocidade com que a arborização dendrítica irá se direcionar para os pontos de atração. Quanto mais negativa for esta constante, mais rápido será o direcionamento e mais próximos dos pontos de atração ficarão os galhos. Se a constante k for positiva, haverá repulsão ao invés de atração. Isto ocorre devido ao modo como calculamos o vetor $v_i = (P-i)$, o vetor do ponto i até o ponto P . Caso calculássemos $v_i = (i-P)$, o processo seria inverso.

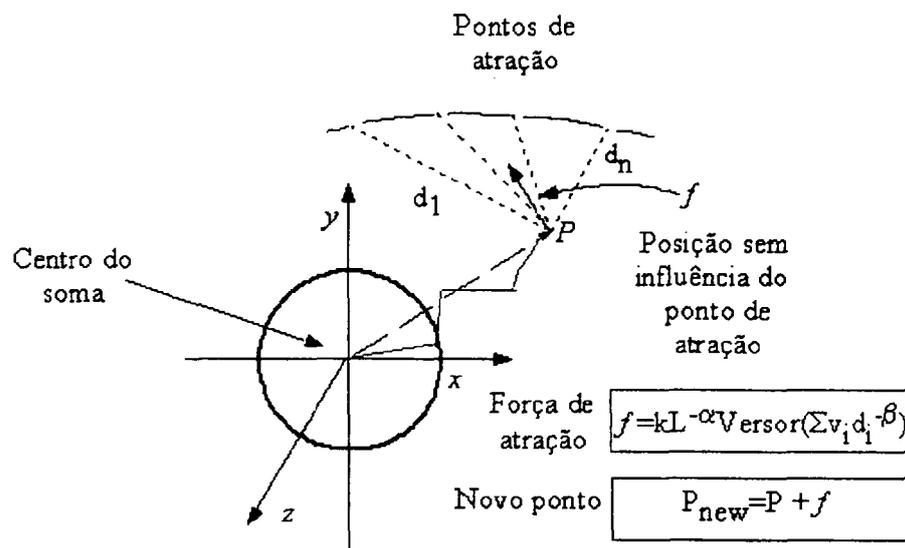


Figura 4.2 – Um ponto P sofrendo influência de vários pontos de atração. d_i é a distância do ponto P até o ponto de atração i , L é a distância percorrida do soma até P , e f é a força resultante.

$\text{Versor}(x)$ é o vetor unitário de x , ou seja, é o vetor x dividido pela sua norma ($|x|$)...

L é a distância percorrida do soma até P , e a constante α serve para direcionar o crescimento dendrítico. Quanto menor este valor, menos direcionado será o crescimento dendrítico.

d_i é a distância de P até o ponto de atração i , e quanto mais próximo i estiver de P , mais forte será a atração, sendo β correspondente à intensidade desta atração. Em nossos cálculos usamos o conceito de atração gravitacional e de campo elétrico, da mecânica clássica, sendo β igual a 2. Ou seja, usamos o conceito de que a força de atração é uma constante multiplicada pelo inverso do quadrado da distância.

Finalmente, P_{new} corresponde à nova posição após o tropismo. Esse tropismo é calculado somando-se o ponto P com o vetor f .

O vetor unitário multiplicado pelas constantes possibilita a reorientação dendrítica e também um leve crescimento dendrítico. Podemos assim simular os fenômenos de tropismo para as ramificações basais e apicais (como visto na seção 2.4 do capítulo 2), na qual substâncias como *Sema3A* e outras influenciam na orientação e também no crescimento dendrítico. Como a norma do vetor resultante f será bem pequena ($0 \leq |f| < 1$), afetará bem pouco o tamanho do arco. Este cálculo também serve para simulação de fenômenos de plasticidade, na qual podemos simular a morte de uma célula neural e os fenômenos de brotamento (seção 2.3 do capítulo 2).

A seguir, mostramos no **algoritmo 3**, a implementação para o cálculo do novo ponto. Neste algoritmo, temos como entrada T , que corresponde aos pontos de influência, composto de n pontos tal que i é um ponto de T , e $1 \leq i \leq n$. P é o ponto que sofrerá a influência, L é a distância percorrida do soma até P , e k , α e β as constantes correspondentes às descrições anteriores. Ao final temos como saída o novo ponto P_{new} .

Algoritmo 3

```

ENTRADA: T,P,L,k, $\alpha$  , $\beta$  ; //Pontos de Tropismo, Ponto que sofre atracao,
//distância percorrida do soma até P, constantes
SAIDA : Pnew; //O ponto apos o tropismo

Inicio
    f:= 0;

```

```

força:= 0;

Para cada ponto i de T faça
    //distância do ponto P ate o ponto da reta
    dist:= Distância(P,i);
    Se dist<0.0000001 força := 1;
    Senão força := 1/distB;
    v := Vetor(i,P); //Vetor de origem no ponto i e final em P
    f := f + v*força;

FimPara;

Delta:= Lα; //direcionamento dendritico
f := Versor(f); //Torna f um vetor unitario
f := k*delta*f; //

Pnew := P+f;

Fim;

```

Na Figura 4.3 temos algumas ramificações sofrendo influências de atração em dois diferentes pontos para uma mesma gramática 5 do capítulo 3. Neste caso, ainda estamos considerando somente a influência de um campo de atração.

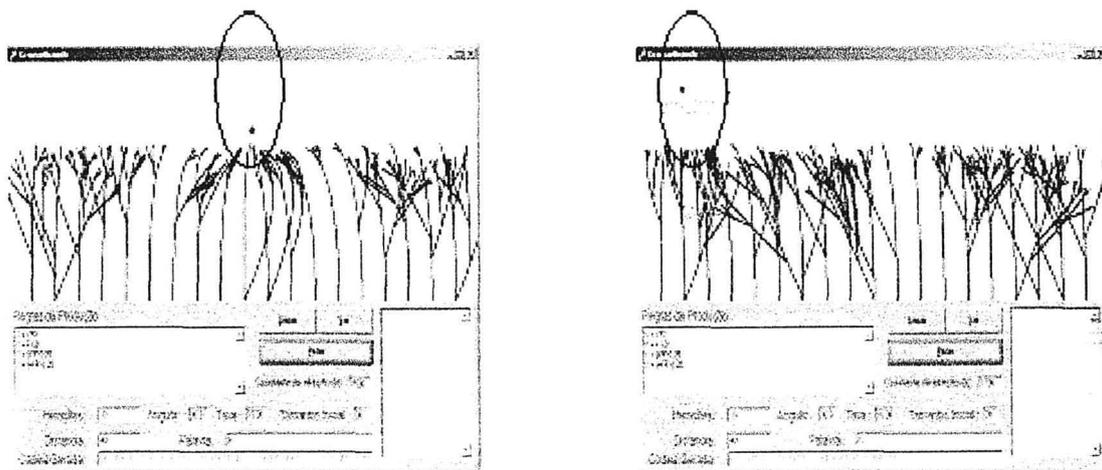


Figura 4.3 – Tropismo em locais diferentes para a mesma gramática.

Adicionamos ao programa então a possibilidade de gerar neurônios artificiais com somas dispostos aleatoriamente na tela. Para acrescentar um pouco de realismo nestas células, adicionamos uma taxa de redução para tamanho e ângulo de inclinação em cada traçado.

A Figura 4.4 mostra alguns neurônios bidimensionais gerados em posições aleatórias utilizando a gramática 5 do capítulo 3. Neste caso utilizamos uma taxa de redução para os ângulos e crescimento dendrítico, ou seja, à medida que o dendrito cresce o tamanho de seus arcos e variação angular vão diminuindo. As posições iniciais das ramificações ao redor do soma foram aleatórias.

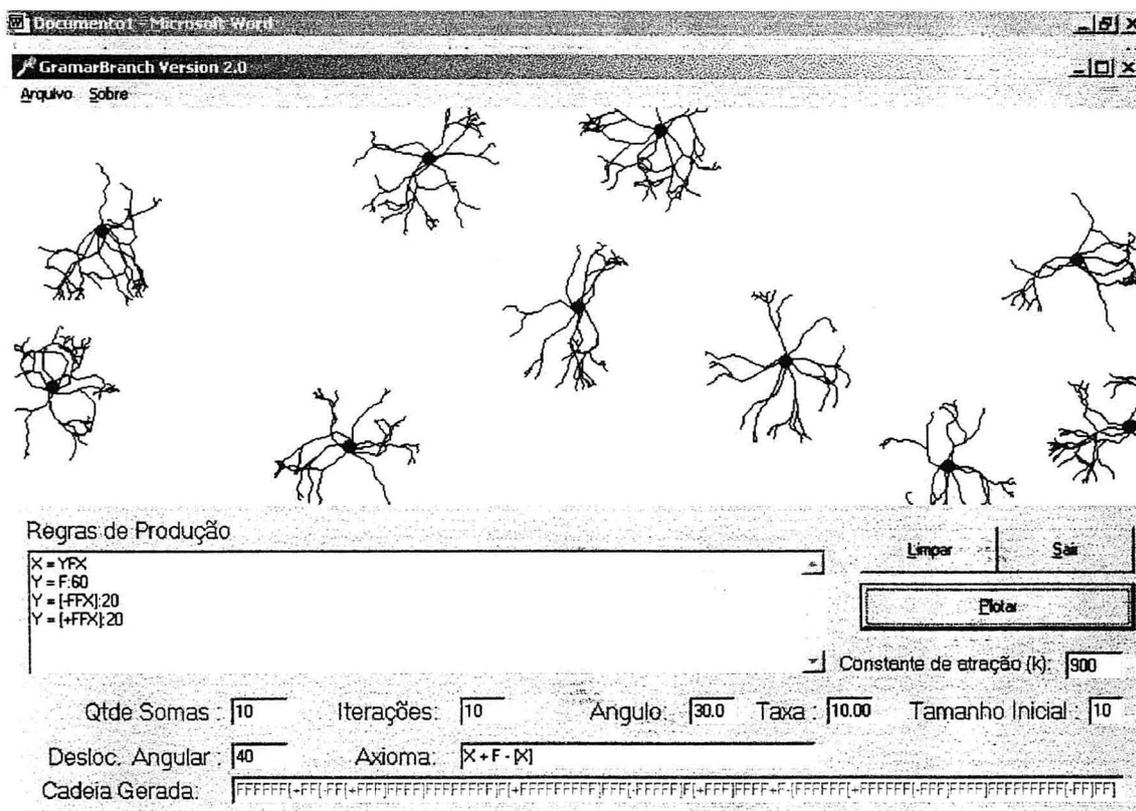


Figura 4.4 – Alguns neurônios artificiais bidimensionais gerados por gramática.

Para uma melhor compreensão da implementação, geramos um diagrama de transição de estados para que pudéssemos construir o interpretador. Para cada estado associamos a execução de uma determinada operação. O conceito aqui utilizado é o mesmo do desenvolvimento de compiladores.

O diagrama de transição de estados da Figura 4.5 representa o processo utilizado na análise sintática da gramática. Cada linha é considerada uma nova entrada para o diagrama.

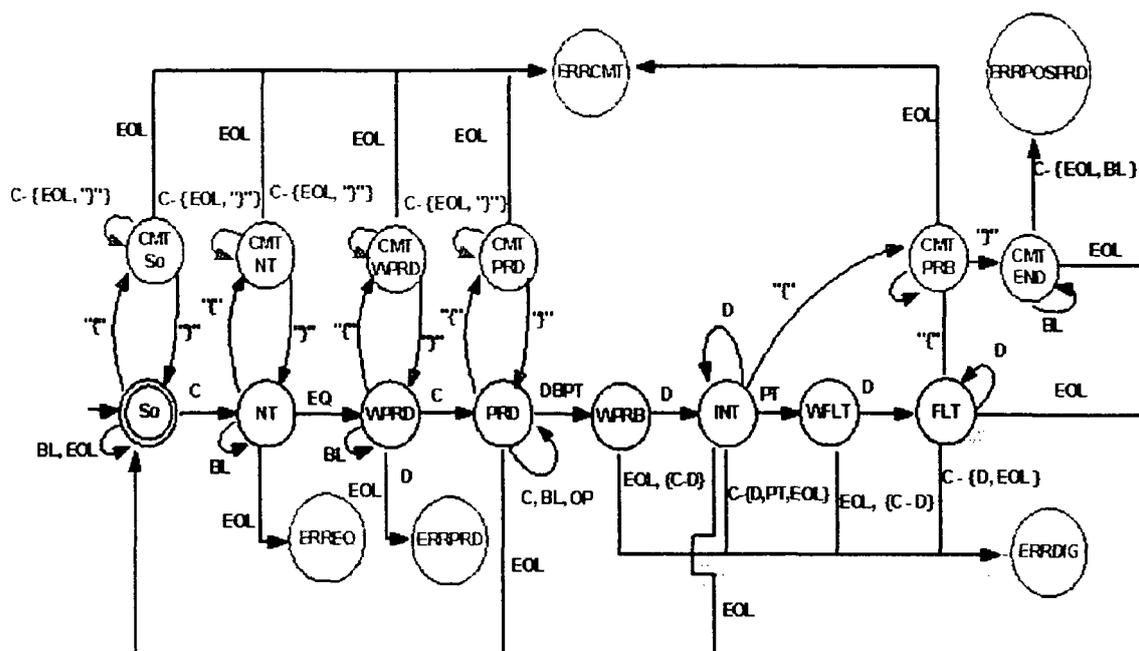


Figura 4.5 – Diagrama de transição de estados para uma L-System estocástica

Na Figura 4.5 temos as seguintes legendas para os estados:

S_0	– estado inicial
NT	– não terminal
WPRD	– análise de início das regras de produção para o não terminal (wait production)
PRD	– análise de cadeia de produção
WPRB	– análise de início de probabilidade (wait probability)
INT	– análise de inteiros
WFLT	– análise de início de ponto flutuante
FLT	– análise de ponto flutuante
ERRRCMT	– erro, chaves de comentário não balanceadas
ERRREQ	– erro, sinal de igual ('=') esperado não ocorreu
ERRDIG	– erro, dígito esperado não ocorreu
ERRPRD	– erro, era esperado caracteres após o sinal de igual ('=')
ERRPOSPRD	– erro, era esperado fim de linha, após comentário final ou fim de produção
CMT S_0	– comentários antes da definição das regras
CMTNT	– comentários após a definição de não terminal ou variável

- CMTWPRD – comentários após o sinal de igual e início das regras da cadeia de substituição do não terminal
- CMTPRD – comentários entre os caracteres da cadeia de substituição do não terminal
- CMTPRB – comentários entre os valores de probabilidade
- CMTEND – comentários após o final, após a definição das regras de produção e probabilidade

E as seguintes entradas:

- C – caractere qualquer
- PT – Ponto
- DBPT – dois pontos (Double Point)
- EOL – fim de linha (End Of Line)
- BL – tabulação, espaço (Blank)
- EQ – sinal de igual (Equal)
- D – Dígito
- OP – operadores LS (+, -, [,], F)

Inicialmente o diagrama encontra-se no estado S_0 e após a leitura de um caractere armazena este como sendo um símbolo não terminal (NT) e aguarda os próximos caracteres. Caso o próximo caractere seja um símbolo de comentário, o estado lerá todos caracteres até que este seja fechado. Caso ocorra um erro de balanceamento de parênteses ou qualquer outra exceção, este erro é notificado ao usuário e a análise termina. Em caso contrário, é lido o sinal de igual (=), deslocando-se para o estado WPRD, que identificará as cadeias de reescrita. O processo é similar ao anterior e caso não ocorra nenhuma exceção e surja o sinal de dois pontos (:), prossegue para o tratamento das probabilidades. É bom lembrar que, se A é um símbolo não terminal que pode ser reescrito por cadeias $\alpha_1, \alpha_2, \dots, \alpha_n$, cada uma delas com probabilidades p_1, p_2, \dots, p_n , então o programa verifica se a soma destes p_i , para este não terminal, é igual a 1, retornando em erro caso contrário. O estado INT armazena números inteiros e WFLT inicia o tratamento para ponto flutuante. Ao final, caso não ocorra erro de somatório das probabilidades ou nenhum outro, o programa partindo do axioma, começa a substituir os símbolos não terminais, de acordo com suas regras de produção e probabilidade. Após um número especificado de iterações, o programa retira da cadeia resultante todos os

símbolos não terminais. Essa cadeia resultante é então lida e a cada caractere é realizada a ação correspondente.

4.2 PADRONIZAÇÃO DE FORMATO

Para que pudéssemos fazer uso das técnicas estocásticas utilizamos os arquivos disponibilizados na internet¹⁶, em notação de traçado NEUROLUCIDA e SWC, criado pelo autor da página. Estes traçados correspondem às células piramidais do córtex do rato. Utilizamos os arquivos no formato SWC por estarem mais completos que os do formato NEUROLUCIDA, mas tivemos que corrigir problemas contidos nestes traçados relacionados ao tipo de ponto armazenado.

4.3 VISUALIZADOR TRIDIMENSIONAL

Para o propósito do projeto precisávamos de funções específicas de computação gráfica que permitissem a visualização tridimensional das células neurais naturais ou artificiais. Estas funções precisavam oferecer recursos de rotação, translação, alteração de textura e formação da imagem passo a passo. Para criar este visualizador, utilizamos as DLLs desenvolvidas em *OpenGL*, em projetos de iniciação científica dos acadêmicos Lúcio Gerônimo Valentim e Marcos Silvano Almeida, do curso de Ciência da Computação da UEM (VALENTIN, 2002; ALMEIDA, 2002). Estas DLLs foram desenvolvidas com o objetivo de visualizar formas neurais em 3D, podendo ser utilizadas em aplicações desenvolvidas em *Borland Delphi* ou *Builder*.

Este visualizador possui todos os recursos citados acima, incluindo a possibilidade de simular o crescimento dendrítico ou axonal, arco por arco. Geralmente, ao anotar o traçado neural, os pesquisadores anotam todos os pontos de um galho para, só então, começar a notação de outro galho. Fazendo uma busca em largura no grafo do neurônio a partir do soma, podemos, em cada passo, anotar os pontos e fornecer ao visualizador na ordem em que foram encontrados. Deste modo, poderemos simular o crescimento em paralelo das ramificações a partir do soma, usando a função de formação do neurônio passo a passo.

¹⁶ <http://www.cns.soton.ac.uk/~jehad/cellArchive/index/topindex.html>

Como uma aplicação inicial temos o caso da visualização de células neurais provindas da notação SWC. Em muitos pontos do arquivo em formato SWC encontramos pontos com o valor -1 ou 0 , indicando que o autor da notação não conseguiu identificar o ponto. Conseguimos resolver algumas destas indefinições aplicando diferentes texturas para cada tipo de ponto. Na Figura 4.6 o autor não conseguiu identificar os pontos em destaque no círculo. Após a alteração da textura e algumas operações de rotação e translação pudemos ver com clareza que os pontos situados no círculo pertencem ao soma. Podemos ainda identificar os dendritos basais na parte superior da imagem, em cores escuras, e os apicais, na parte inferior em cores mais claras.

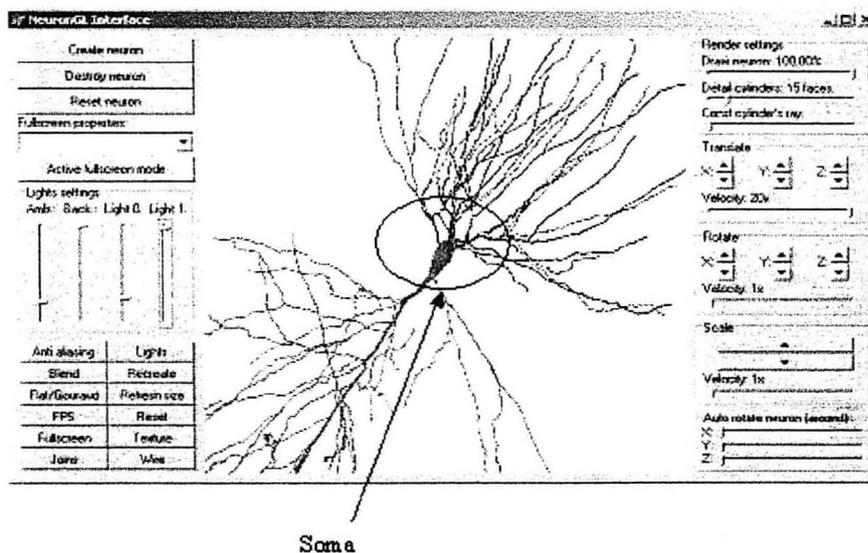


Figura 4.6 – Visualizador tridimensional. Nesta aplicação, pode-se perceber o soma do neurônio, com textura diferente. Acima os dendritos basais e abaixo os apicais.

4.4 NOTAÇÃO VETORIAL

Como mostrado anteriormente, podemos gerar células mais realistas extraindo medidas referentes à forma neural e ao nível de ramificação e montando diversas funções de distribuição referentes à estas medidas. Em cada nível, aplicamos então, diferentes funções de distribuição (FD).

No grafo da Figura 3.9 do capítulo 3, subtraindo os pontos iniciais dos finais em cada arco, temos um vetor correspondente ao arco e o módulo deste vetor corresponde ao tamanho do arco (TamArc). A soma de todos os TamArc em um segmento corresponde ao tamanho do segmento (TamSeg).

COELHO(1998) e TANAKA(2000), propõem que para podermos ter uma representação espacial precisamos também extrair funções de distribuição correspondentes aos ângulos existentes entre um arco e outro. Neste modelo, temos dois ângulos: o ângulo de inclinação em relação ao arco anterior (ϕ) e o ângulo de rotação em relação ao segmento anterior (θ), como mostra a Figura 4.7. Ou seja, “ ϕ é o ângulo de desvio de w em relação a v que é análogo a curvatura e θ o ângulo de desvio do plano osculante (plano uv) em torno do vetor v , análogo a torção” na geometria diferencial (COELHO,1998).

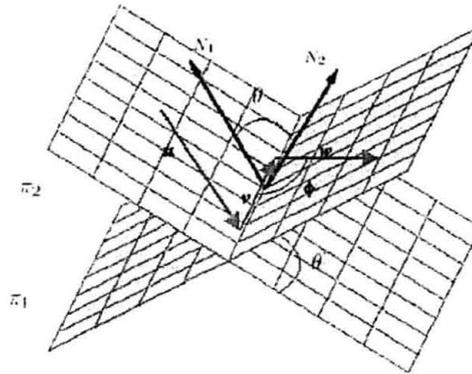


Figura 4.7 – Ângulos de torção θ e curvatura ϕ .

Consideramos v como sendo o vetor correspondente ao arco atual, w o próximo arco e u o vetor anterior a v para um dado ponto P (MTO, CP ou BP). No caso de MTO, considera-se v como arco que sai do centro de massa C do soma até o MTO (Figura 4.8), e u o eixo x, ou seja, $u = (1, 0, 0)$. Algumas vezes, quando a coordenada x de v é 0 muda-se u para $(0,1,0)$

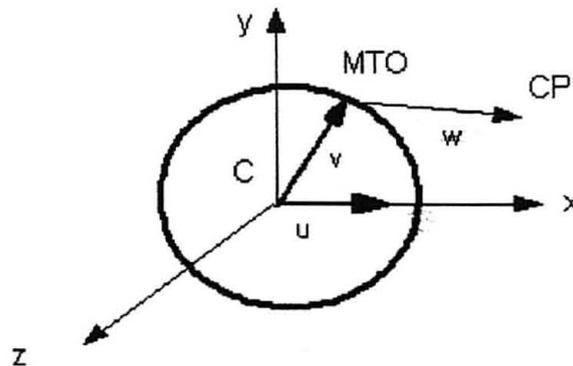


Figura 4.8 – Esquema para início da notação vetorial.

4.4.1 Sistema de coordenadas locais

O modelo de COELHO(1998) e TANAKA(2000) leva a um sistema de equações não linear, conforme descrevemos nas equações a seguir.

Seja $n = (n_1, n_2, n_3)$ o vetor normal do plano formado pelos vetores u e v , e $m = (m_1, m_2, m_3)$, o vetor normal ao plano formado pelos vetores v e w . Denotamos por $u \wedge v$ o produto vetorial entre os vetores u e v , e $\langle u, v \rangle$ o produto interno de u com v (BOLDRINI, et. al., 2000). Então, podemos calcular os vetores unitários n e m através das equações (4.1) e (4.2)

$$n = \frac{u \wedge v}{|u \wedge v|} \quad (4.1)$$

$$m = \frac{v \wedge w}{|v \wedge w|} \quad (4.2)$$

Calculando o produto interno entre n e m , temos o ângulo θ entre os planos e do produto interno entre v e w , temos o ângulo ϕ

$$\langle n, m \rangle = |n| |m| \cos \theta \quad (4.3)$$

$$\langle v, w \rangle = |v| |w| \cos \phi \quad (4.4)$$

Seja

$$v = (v_1, v_2, v_3)$$

$w = (x, y, z)$, o vetor que queremos encontrar e $|w|$ é o tamanho de w , ou seja

$$|w|^2 = x^2 + y^2 + z^2 \quad (4.5)$$

Da equação (4.3), uma vez que $|n| = |m| = 1$, teremos então

$$\langle n, v \wedge w \rangle = |v \wedge w| \cos \theta = |v| |w| \sin \phi \cos \theta \quad (4.6)$$

Na simulação, temos então o tamanho do próximo vetor ($|w|$), os vetores u, v , e os ângulos θ e ϕ . Podemos então construir um sistema de equações composto das equações (4.4), (4.5) e (4.6), o que resulta numa equação não linear. Deste modo, caso usássemos estas equações encontraríamos dois vetores w_1 e w_2 , um acima do plano e outro abaixo do plano formado por u e v . A solução tomada foi então encontrar um sistema de coordenadas local em cada ponto de encontro de v com w , e a partir daí, usando o sistema de coordenadas esféricas

encontrar o novo vetor w . Na Figura 4.9 temos os vetores u , v , e w e os ângulos θ e ϕ existentes.

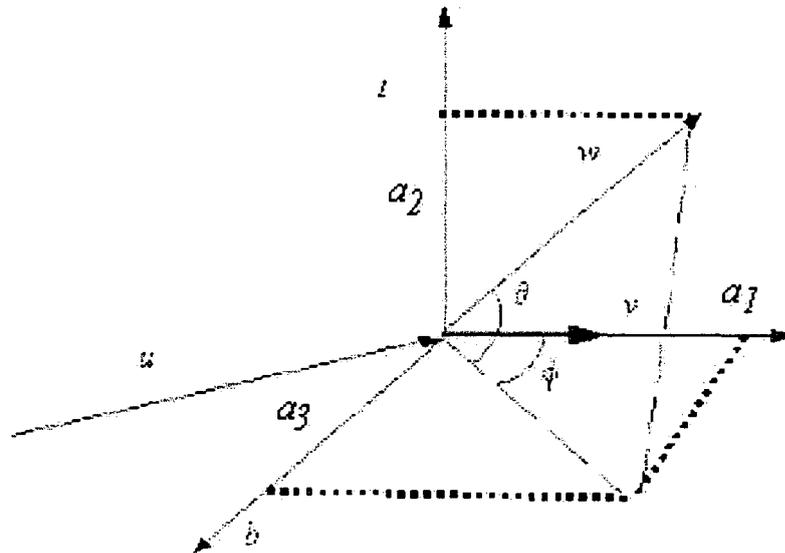


Figura 4.9– Sistema de coordenadas locais

Seja

$$\begin{cases} t = \frac{u \wedge v}{|u \wedge v|} \\ v' = \frac{v}{|v|} \\ b = \frac{v' \wedge t}{|v' \wedge t|} \end{cases} \quad (4.7)$$

Temos então um conjunto de vetores ortogonais e unitários. Queremos então saber quais são as coordenadas de w tendo uma combinação linear com v' , t e b . Podemos escrever w como

$$w = a_1.v' + a_2.t + a_3.b \quad (4.8)$$

Pode-se verificar que

$$\begin{cases} \theta = \arcsen\left(\frac{a_2}{|w|}\right) \\ \phi = \arctg\left(\frac{a_3}{a_1}\right) \end{cases} \quad (4.9)$$

Tendo então ϕ , θ , u , v e $|w|$ encontramos v' , t e b . Encontramos os coeficientes a_1 , a_2 e a_3 por

$$\begin{cases} a_2 = |w| \operatorname{sen} \theta \\ a_1 = |w| \cos \theta \cos \phi \\ a_3 = |w| \cos \theta \operatorname{sen} \phi \end{cases} \quad (4.10)$$

Deste modo, uma vez que temos as medidas de u , v , θ e ϕ e o tamanho de w , temos condições de encontrar w , sem equações não lineares. Para verificar a veracidade da notação desenvolvemos um programa que imprime em uma janela o neurônio na notação original (EUTECTIC) e na outra o neurônio reconstruído. Para verificar a ocorrência de erros foi criado um arquivo de *log* que analisava as pontuações do original e do construído.

Constatamos que quando ocorriam erros eram de casas decimais da ordem de 10^{-10} , sendo considerados desprezíveis para nossos cálculos. Na imagem da Figura 4.10, a caixa de texto circulada em destaque indica que não ocorreram erros para este caso.

Construímos, então, um programa que armazena para cada MTO o nível de ramificação, o vetor correspondente aos pontos do centro do soma ao MTO, os ângulos θ e ϕ e o tamanho do próximo arco. Para cada CP é armazenado o nível de ramificação, os ângulos θ e ϕ e o tamanho do próximo arco. Para BP, o programa armazena o nível de ramificação, o tamanho do segmento anterior, os tamanhos dos dois próximos arcos e os dois ângulos θ e os dois ângulos ϕ , correspondentes a cada próximo arco. Para um NE são armazenados o nível de ramificação e o tamanho do segmento dendrítico anterior. Para cada nível construímos funções de distribuição acumuladas correspondentes às medidas relacionadas acima. Além disso, extraímos medidas correspondentes ao número de arborizações existentes (MTO). As funções de distribuição acumuladas evitam que na simulação seja sorteado um número randômico y cujo correspondente possa ser mais de um elemento x . Ou seja, uma função de distribuição acumulada é uma função injetora para cada intervalo em x . Uma vez que tratamos de neurônios piramidais, dividimos estas funções de distribuição em dois grupos: funções correspondentes aos dendritos apicais e aquelas correspondentes aos basais, além dos axônios. Os arquivos em notação SWC utilizados para nossa pesquisa apresentavam poucas ramificações axonais, ou nenhuma. Deste modo, embora tenhamos criado funções de distribuição correspondentes aos axônios, não estamos utilizando em nossas simulações.

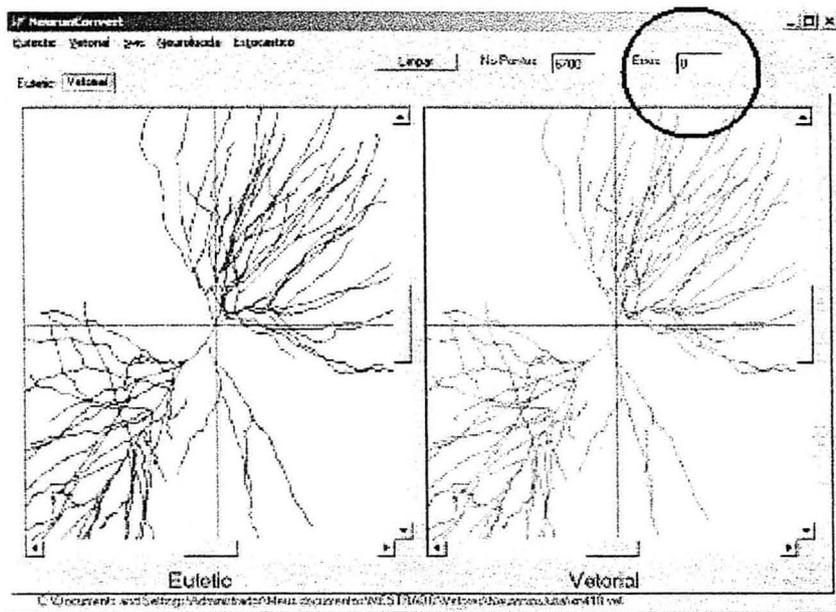


Figura 4.10 – Neurônio com notação eut e vet. A figura mostra um mesmo neurônio gerado utilizando notação EUTECTIC e notação vetorial. Neste caso, não houve erros (em destaque), ou seja, todos os pontos dos dois neurônios foram coincidentes.

A Figura 4.11 apresenta parte de um arquivo em notação vetorial, com os seguintes campos:

- ID – Identificação do número do nodo
- TNodo – Tipo do nodo (MTO, BP, CP ou NE);
- TArc – Tipo de arco (1,2,3,4 ou -1) para soma, axônio, basal, apical, ou indefinido;
- x,y,z – Coordenadas do ponto, quando MTO;
- Curv1,
- Curv2 – Valor da curvatura do primeiro e segundo ramo (caso de ramificação). Se TNodo não for BP, ignora-se Curv2;
- Torc1,
- Torc2 – Valor da torção do primeiro e segundo ramo (caso de ramificação). Se TNodo não for BP, ignora-se Torc2;
- TamArco1,
- TamArco2 – Valor do tamanho dos próximos arcos do primeiro e segundo ramo (caso de ramificação). Se TNodo não for BP, ignora-se TamArco2;
- TamSeg – Tamanho do segmento;
- Raio – Diâmetro do segmento.

No Points: 4638

No MTOS : 5

Data : 12/5/2003

ID	TNode	TArc	Nivel	x	y	z	Curv1	Torc1	Curv2	Torc2	TamArco1	TamArco2	TamSeg	Raio
1	MTO	3	1	-1.20	-2.30	5.80	0.57493	1.95673	0.00000	0.00000	9.41063	0.00000	0.00000	0.532
2	BP	3	1	0.00	0.00	0.00	2.36921	0.77876	0.66898	0.25075	5.80086	2.12132	9.41063	0.260
3	CP	3	2	0.00	0.00	0.00	1.23110	3.11298	0.00000	0.00000	3.08869	0.00000	0.00000	0.260
4	CP	3	2	0.00	0.00	0.00	0.19348	2.37348	0.00000	0.00000	3.88973	0.00000	0.00000	0.260
5	CP	3	2	0.00	0.00	0.00	0.57584	1.76897	0.00000	0.00000	6.93037	0.00000	0.00000	0.260
6	BP	3	2	0.00	0.00	0.00	0.56373	2.95573	0.14351	0.94412	2.73130	1.61245	19.70965	0.260

Figura 4.11 – Trecho de um arquivo em notação VET.

A Figura 4.12 mostra gráficos de funções de distribuição relacionadas ao tamanho do arco e do segmento para 37 níveis, extraídos de medidas de neurônios piramidais encontradas em formato SWC. O programa converteu estas células em notação vetorial e a seguir construiu funções de distribuição acumulada para estas medidas.

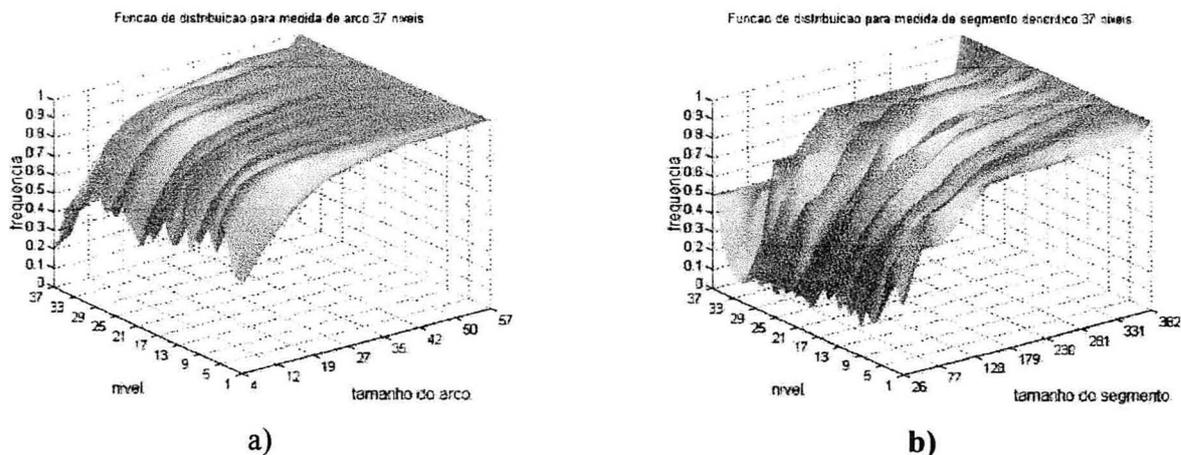


Figura 4.12 – Funções de distribuição de medidas de neurônios piramidais do córtex de ratos. (a) arco e nível e (b) segmento e nível.

4.5 SIMULAÇÃO DE MONTE CARLO USANDO FUNÇÕES DE DISTRIBUIÇÃO

Tendo as funções de distribuição acumuladas para as diferentes medidas, podemos então extrair as medidas a serem geradas sorteando um número aleatório entre 0 e 1 e verificando qual medida no eixo x equivale ao valor sorteado (BANKS, CARSON e NELSON, 1996). Com isso podemos então sortear tamanhos de arco, os ângulos θ e ϕ , o número de ramificações e assim por diante. COELHO(1998) propõe um algoritmo simples para a geração neural utilizando funções de distribuição acumulada e simulação de Monte

Carlo, além do conceito de notação vetorial. O algoritmo proposto por COELHO(1998) era para neurônios bidimensionais. Com algumas modificações, adaptamos o algoritmo para geração de células tridimensionais. O **Algoritmo 4** corresponde ao algoritmo adaptado.

Algoritmo 4

Início

Sortear NumMTO, o número de ramificações do soma;

Para cada ramificação

sortear *TamSeg Diam*, o tamanho do segmento e espesura respectivamente;

Enquanto não atingir o tamanho sorteado

sortear *AngCrv*, *AngTrc* e *TamArc* e traça uma reta com tamanho

TamArc curvatura *AngCrv* e torção *AngTrc*;

FimEnquanto;

sortear a probabilidade de ramificação e verificar se o número sorteado é menor que a probabilidade atribuída àquele nível do galho (quando esta probabilidade sorteada é maior, entendemos que foi encontrado um NE);

Fimpara;

Fim.

Estas distribuições são utilizadas apenas para o crescimento dendrítico. Para o soma consideramos apenas uma esfera.

4.6 FUNÇÕES DE BASE RADIAL.

Para encontrar o tamanho de um arco em uma função de distribuição por nível e tamanho de arco, precisamos de dois valores, um associado à probabilidade e outro ao nível do segmento. Logo, temos uma função que associa cada tamanho a um vetor bidimensional cujas coordenadas são a probabilidade e o nível. Isto é, cada tamanho corresponde a um ponto coletado, e cada ponto é um vetor bidimensional. Podemos estender este conceito para vetores

n-dimensionais. No entanto, como os dados das funções de distribuição são discretos, obtemos na simulação apenas uma aproximação do valor real. A interpolação destes pontos possibilita uma melhor aproximação.

Neste trabalho, utilizamos para interpolação as funções de base radial (FBR), por ser um dos métodos mais adequados para valores de mais de uma dimensão, e que não possuem distribuições regulares. Segundo CONSULARO (2001), “sua popularidade para aplicação de reconhecimentos de padrões e aprendizado de máquina ocorre porque não é necessário um espaçamento regular entre os pontos de origem, viabilizando o uso de casos reais em treinamento”.

Suponha um conjunto de n vetores correspondentes aos pontos conhecidos $V = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n\}$ sendo $\vec{v}_i = (c_1, c_2, \dots, c_d)$, $1 \leq i \leq n$, um vetor de d dimensões. Associado a V , temos um conjunto $F_v = \{f(\vec{v}_1), f(\vec{v}_2), \dots, f(\vec{v}_n)\}$, sendo que cada $\vec{f}(v_i)$ é o resultado da aplicação de uma função sobre \vec{v}_i . No caso descrito anteriormente $\vec{f}(v_i)$ seria o tamanho do arco. Queremos agora determinar quais são os valores correspondentes à um conjunto de m pontos $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m\}$, sendo $\vec{x}_i = (l_1, l_2, \dots, l_d)$. Ou seja, queremos associar a X um $F_x = \{f(\vec{x}_1), f(\vec{x}_2), \dots, f(\vec{x}_m)\}$ (JAQUES, COELHO, 2003c).

Em uma FBR, a função de interpolação é definida como :

$$F(\vec{v}) = \sum_{i=1}^n w_i \phi(\|\vec{v} - \vec{v}_i\|) + \sum_{k=1}^d a_k c_k + b \quad (4.11)$$

sendo $\phi(r)$ uma função sobre a norma de r ; w_i , a_k e b são os coeficientes utilizados para o ajuste da interpolação. Tendo V e F_v , podemos obter os coeficientes w_i , a_k e b . Logo,

$$F(\vec{x}) = \sum_{i=1}^n w_i \phi(\|\vec{x} - \vec{v}_i\|) + \sum_{k=1}^d a_k l_k + b \quad (4.12)$$

Podemos ver a equação (4.11) na forma matricial

$$F_v = \begin{bmatrix} \Phi & V & 1 \\ V^T & 0 & 0 \\ 1^T & 0^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a \\ b \end{bmatrix},$$

e (4.12) na forma

$$F_x = [\Phi_x \quad X \quad 1] \begin{bmatrix} w \\ a \\ b \end{bmatrix}$$

4.6.1 Thin plate splines

Devido as nossas FD apresentarem mais de um parâmetro, podendo ser representadas graficamente como superfícies, usamos um tipo especial de FBR, as *thin plate splines* (TPS), ou *splines de superfícies*. Elas modelam uma superfície no qual pontos de carga provocam deformações. Em uma TPS, assegura-se que a superfície na qual os pontos de carga atuam apresentem energia mínima de deformação, desenhando assim uma superfície suave e contínua (CASTANHO e TOZZI, 1996; COSTA e CÉSAR, 2001). Na TPS que utilizamos, Φ foi definida por:

$$\Phi(x) = \begin{cases} 0 & \text{se } x = 0 \\ x^2 \log(x) & \text{caso contrário} \end{cases} \quad (4.13)$$

Logo, aplicamos (4.13) em (4.11) e obtemos os coeficientes. A seguir aplicamos (4.13) em (4.12) juntamente com os coeficientes, e obtemos os F_x desejados.

Deste modo, uma vez que tenhamos medidas das células de uma mesma classe, podemos ter uma função de distribuição associada à ocorrência de cada característica morfológica da classe, ou uma função interpoladora.

Usando TPS como auxiliar para a geração de células artificiais (JAQUES; COELHO, 2003b) diminuímos a quantidade de dados a consultar e melhoramos a precisão no sorteio das medidas. Para este trabalho, utilizamos TPS para arco e diâmetro. As TPS utilizadas foram desenvolvidas pelo mestrando em Ciência da Computação da UEM, Mitio Yoshida.

4.7 SÍNTESE DE CÉLULAS NEURAIS UTILIZANDO FUNÇÕES ESTOCÁSTICAS

Utilizando diferentes medidas estatísticas podemos então associar uma gramática capaz de produzir ramificações morfológicamente semelhantes às ramificações dendríticas de neurônios naturais. Isto é, podemos associar a alguns símbolos não terminais algumas regras

para a formação de um dendrito basal, um axônio e assim por diante. Pode-se ainda adicionar em cada um desses não terminais as funções de distribuição relacionadas. Na *gramática 6*, temos um exemplo de utilização destas funções.

Gramática 6

Axioma : X

nível := 1;
 tamanho := 0;
 tamanhoSD := FTamSD(nível);
 D := FDiametro(nível);

Regras

1. para (tamanho < tamanhoSD) : X → FX
 - 1.1 tamanhoCP := FTamCP(nível);
 - 1.2 angcrv := FAngCrv(nível);
 - 1.3 angtrc := FangTrc(nível);
 - 1.4 tamanho := tamanho + tamanhoCP
2. para (tamanho ≥ tamanhoSD) : X → FY
 - 2.1 prob := FProbRamificar(nível);
3. prob: Y → ZZ
 - 3.1 nível := nível + 1;
4. 100% - prob: Y → NULL
 - 4.1 nível := nível -1;
 - 4.2 tamanho := 0;
5. 100% Z → [FX]
 - 5.1 tamanhoSD := TamSD(nível);
 - 5.2 D= Fdiametro(nível);
 - 5.3 tamanho := 0;

Aqui, $FDProbRamificar(nível)$ controla a probabilidade de ramificação em cada nível do galho. $tamanhoSD$ e $tamanhoCP$ são os tamanhos do ponto atual até um final de segmento (BP ou NE), e de arco (CP), respectivamente. $FangCrv(nível)$, $FangTrc(nível)$, $FtamCp(nível)$ e $FtamSD(nível)$ são as FDs correspondentes aos ângulos de torção e curvatura e aos tamanhos de arco e segmento para um determinado nível.

Na regra 2, X é substituído por FY e a seguir é atualizada a probabilidade de ramificação para aquele nível. Caso a probabilidade sorteada na simulação seja menor ou igual a probabilidade naquele nível, ocorrerá uma bifurcação, sendo aplicada a regra 3, senão a ramificação será terminada (regra 4). Quando a ramificação é terminada o último nó do galho é rotulado como NE.

Uma das vantagens da utilização de gramáticas vetoriais estocásticas é o fato de alguns galhos apresentarem regras de ramificação diferentes. Deste modo, podemos, por exemplo, associar uma gramática para o crescimento dos axônios, outra para os dendritos basais e outra para os dendritos apicais das células piramidais.

5 SÍNTESE NEURAL

Baseando-se no modelo proposto, buscamos utilizar as ferramentas disponíveis e os conceitos até agora pesquisados e experimentados para elaboração de nosso simulador neural. No entanto, experimentamos alguns percalços. A DLL desenvolvida pelos alunos da graduação não pôde ter continuidade, uma vez que a bolsa de iniciação científica terminou, estes se graduaram e o aluno que daria continuidade ao projeto acabou desistindo. O desejo final era que a DLL pudesse ler informações de traçado de mais de um neurônio. Embora no manual do visualizador neural esteja descrita esta possibilidade, constatamos em nossos experimentos que não era possível.

Após alguns experimentos, escolhemos a gramática e, utilizando o conceito de busca em largura, adaptamos a simulação de crescimento dendrítico em paralelo, para o formato de leitura da referida DLL. Uma vez que a gramática gerava seus traçados em paralelo, usamos o conceito de busca em profundidade com retrocesso para armazenar as notações de traçado em notação vetorial. Assim, pudemos fazer comparações dos neurônios naturais com os artificiais.

Como nosso objetivo principal é a geração de neurônios morfologicamente parecidos com os neurônios piramidais, tínhamos que direcionar o crescimento dendrítico corrigindo a gramática de modo a satisfazer esta morfologia. Esta correção foi feita utilizando campos de atração que simulavam a influência de fatores neurotróficos nos dendritos basais e apicais (POLLEUX, MORROW e GHOSH, 2000).

Na seção 5.1 descrevemos como criamos FD condicionais e na seção 5.2 apresentamos a definição do modelo de gramática utilizada, a *gramática estocástica condicional*.

A seguir, na seção 5.3, apresentamos o algoritmo utilizado para geração do crescimento neural em paralelo e finalmente, na seção 5.4 mostramos os processos e funções disponíveis ao software mostrando uma breve descrição de suas funções e à seguir, mostramos um diagrama com as opções do software desenvolvido, e apresentamos algumas células geradas e gráficos referentes.

Finalmente, na seção 5.5 analisamos graficamente os resultados obtidos comparando gráficos de células naturais e células artificiais.

5.1 FUNÇÃO DE DISTRIBUIÇÃO CONDICIONAL

A fim de melhorar nossa simulação, resolvemos, além das TPS, utilizar FD condicionais para algumas medidas (JAQUES; COELHO, 2003b, 2003c). Usando o conceito da *regra de Bayes* (seção 4.3, do capítulo 4), desejamos saber qual a probabilidade de ocorrer um determinado valor, dado que um valor anterior tenha ocorrido. Estávamos interessados, por exemplo, em saber qual a probabilidade de ocorrer um tamanho B no segmento de nível n , dado que tenha ocorrido um tamanho A no nível $n-1$. No entanto, uma vez que dispúnhamos de mais de 30 níveis para alguns tipos de arcos, tal arquivamento de dados se tornaria desastroso, visto que o tamanho no nível 30 era dependente do nível 29, que era dependente do nível 28, e assim sucessivamente. Como temos nossas medidas distribuídas em 15 classes, para o nível 1 e 2 teríamos $225 = 15 \times 15$ combinações, e para o nível 3, teríamos 3375 combinações.

Na seção 4.3 do capítulo 4, usamos o conceito de redução de espaço amostral. Uma vez que tenha ocorrido um evento A , o nosso espaço amostral é reduzido para A , e a chance de ocorrer B depende então de A e B . Deste modo, criamos uma FD de $B|A$ e nada mais. Encontrando B , localizamos na FD a classe correspondente a este valor e então podemos encontrar C , pois temos uma FD para $C|B$. Na Figura 5.1 temos do lado esquerdo as FD para as classes dominantes ou que devem ocorrer previamente, e do lado direito as FD para as classes dependentes.

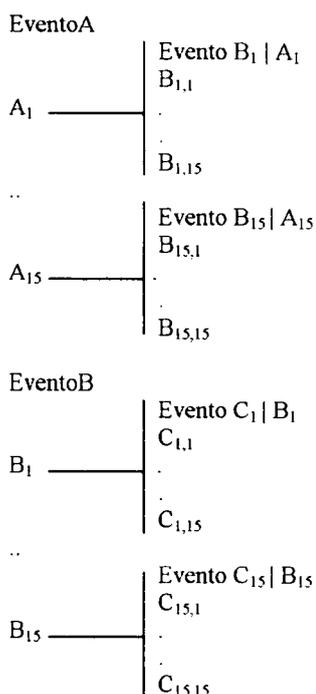


Figura 5.1 – Esquema para construção de FD dominantes e dependentes.

Utilizamos FD condicionais para tamanho e diâmetro dos segmentos, curvatura e torção dos arcos. O tamanho do segmento n depende de qual tamanho tenha ocorrido no segmento $n-1$. O processo é análogo para a espessura do diâmetro. No caso de curvatura e torção, construímos uma FD cujo valor de torção depende da ocorrência da curvatura.

O algoritmo 5 mostra como é feito para simular a ocorrência de uma medida, dado que tenha ocorrido uma medida anterior. Para fins genéricos, supomos que já tenha ocorrido pelo menos um sorteio. Temos então a classe do evento anterior e buscamos sortear qual seria o próximo valor, dado que uma classe de um evento anterior tenha ocorrido. Uma vez localizado este valor na FD dependente, localizamos a que classe pertence na classe dominante. Esta classe será a classe do evento anterior para o próximo evento, e assim sucessivamente.

Algoritmo 5

Entrada : $ClAnt$; //Classe do evento anterior

SAÍDA : $ClAt$, $ValAt$ //Classe Atual, Valor Atual

Início

$Prob := random$;

$ValAt := Valor(Prob, Dep[ClAnt]);$ //procura na classe dependente

$ClAt := Classe(ValAt);$ //Localiza valor na classe dominante

Fim;

5.2 DEFINIÇÃO DA GRAMÁTICA

Para que pudéssemos gerar células neurais adaptamos a proposta de MC CORMICK e MULCHANDANI(1994) (JAQUES; COELHO, 2002, 2003a). A gramática 7 é o modelo que utilizamos em nosso projeto. Na gramática proposta, inserimos algumas ações de controle (linhas 2, 3, 4, 5, 7, 8, 8.1, 8.1.2, 8.2.1, 9 e 10), sendo que a execução de operações gráficas é condicionada a valores lógicos de expressões (linhas 7.1, 8.1, e 8.2). Na linha 4, a função $SortSD(Nivel, Tipo)$ sorteia o tamanho do segmento dendrítico tendo como parâmetros o nível e o tipo de segmento dendrítico, basal ou apical, enquanto que na linha 5 a função $SortDiametro(Nivel, Tipo)$ sorteia o diâmetro do segmento. Estas funções são

condicionadas ao sorteio das medidas do nível anterior, exceto para o nível 1. Ou seja, o sorteio no nível n depende do tamanho sorteado no nível anterior. O uso de uma função condicionada ao tamanho anterior é mais realística do que simplesmente o sorteio não condicional.

Uma vez que a gramática proposta utiliza-se não somente de valores probabilísticos mas de valores lógicos, denotaremos esta gramática de gramática estocástica condicional.

Na linha 6, X corresponde ao desenho do segmento dendrítico. Temos então associado ao arco as funções $TamCP(Nível, Tipo)$, $CurvTorc(Nível, Tipo)$. Sorteia-se então o tamanho do arco, a curvatura e a torção, respectivamente. A função $CurvTorc(Nível, Tipo)$ sorteia inicialmente a curvatura e a seguir sorteia a torção para a ocorrência daquela curvatura.

O processo de ramificação começa no nível 1 com o axioma X e a escolha do tamanho do segmento dendrítico (linha 4). Enquanto a soma dos arcos não atingirem o tamanho do segmento (linha 7.1), sorteia-se novo tamanho de arco, curvatura e torção. Os ângulos de curvatura e torção servem para orientar o crescimento do galho.

Quando a soma dos arcos atinge o tamanho do segmento sorteamos um número aleatório entre 0 e 1. Usamos este número para simular a possibilidade de ocorrer uma ramificação. Caso este número sorteado seja menor ou igual à probabilidade de ramificação (linha 8.1), dois novos galhos começarão a crescer, começando então dois novos segmentos dendríticos. Em caso contrário, não ocorrerá nenhuma ramificação significando que o galho parará de crescer. O processo segue recursivamente para os demais galhos até que termine o crescimento de todos os galhos.

Devido ao uso do método de Monte Carlo, cada célula gerada será morfologicamente diferente das demais.

As correções de orientação do crescimento é feita usando um campo de atração representado pela função $FieldTrop(Tipo)$, aplicada aos galhos basais (linha 9) e apicais (linha 10), simulando assim a orientação por fatores neurotróficos

Cada dendrito é representado como uma série de pequenos cilindros, e cada cilindro é correspondente aos arcos do dendrito. Deste modo, estes pequenos arcos representam as curvaturas e torções sofridas pelos galhos.

Gramática 7

1. Axioma : X
2. Nivel := 1;
- 3 SomaCp := 0;
4. TamSD := SortSD(Nivel,Tipo);
5. Diam := SortDiametro(Nivel,Tipo);
6. $X \rightarrow F_{FTamCP(Nivel,Tipo) CurvTorc(Nivel,Tipo)} Y$
7. SomaCP := SomaCP+TamCP(Nivel,Tipo);
 - 7.1 [SomaCP < TamSD] : $Y \rightarrow X$
8. Prob:=SortProb();
 - 8.1 [Prob <= ProbRam(Nivel,Tipo)] : $Y \rightarrow [X][X]$:
 - 8.1.1. Nivel := Nivel + 1;
 - 8.1.2. TamSD := SortTamSD(Nivel,Tipo);
 - 8.2 [Prob > ProbRam (Nivel,Tipo)] : $Y \rightarrow NULL$
 - 8.2.1 Nivel := Nivel -1;
9. FieldTrop(Basal)
10. FieldTrop(Apical)

5.3 GERAÇÃO NEURAL

Após as transformações dos traçados neurais em notação vetorial, geramos as FD e para algumas destas FD criamos TPS. Bastava então utilizar a gramática para gerar as células. Embora uma gramática L-System possa ser vista pelo ser humano como uma geração em paralelo, o desenho das ramificações em paralelo depende do algoritmo desenvolvido. Como a estrutura gerada pode ser vista como um grafo, utilizamos então o conceito de expansão ou busca em largura (BONDY e MURTY, 1979; BALAKRISHNAN, 2000). Inicialmente marcamos cada nó do grafo com o rótulo *NÃOVISITADO*. Partindo de um nó inicial em uma árvore, marcamos este com o rótulo *ABERTO*. A seguir, marcamos este nó como o rótulo *VISITADO* e rotulamos todos os nós adjacentes ainda não visitados com o rótulo *ABERTO*. O

processo segue análogo agora para cada nó com rótulo *ABERTO*, marcando-os como *VISITADO*. Quando todos os nós estiverem rotulados como *VISITADOS* o processo termina. Deste modo, são visitados todos os nós de um nível da árvore, antes de prosseguir para outro nível, o que permite a simulação de paralelismo. Como as funções da DLL desenvolvida desenham a célula lendo os pontos de um arquivo, gravamos então as informações destes pontos, sendo cada um destes pontos os nós visitados da árvore segundo a busca em largura. O algoritmo 6 apresenta a idéia geral. Este algoritmo foi adaptado às sutilezas da linguagem Delphi.

Algoritmo 6

```
CarregaGramatica();
SorteiaPosicaoSoma();
SorteiaPosicaoMTOs();
Para cada PontoMTO
    StringSegmento := Axioma;
    PoeNaFila(StringSegmento);
FimPara;
Enquanto FILA NÃO VAZIA faça
    Para cada Segmento da FILA faça
        RetireDaFila(StringSegmento);
        GraveAcao(StringSegmento,PosSegmento);
        Se Gramática do Segmento Gerar ramificação
            PoeNaFilaAux(StringNovoSegmento1);
            PoeNaFilaAux(StringNovoSegmento2);
        FimSe;
        Se string do segmento ainda tem caracteres a ler
            PoeNaFilaAux(StringSegmento);
    FimPara
FILA := FILAAUX;
```

FimEnquanto;

Fim;

Consideramos cada segmento contendo uma string correspondente ao conjunto de ações de desenho a serem executadas, e um ponteiro *PosSegmento* que aponta para a próxima ação a ser executada. Inicialmente temos um segmento partindo de cada MTO e nada mais. Cada um destes segmentos tem inicialmente como string o axioma e o ponteiro está na posição 1. Coloca-se estas strings em uma fila e, posteriormente, vamos retirando da fila e realizando as ações gramaticais para caractere em que *PosSegmento* aponta. Deste modo, quando a ação é de desenho, gravamos os pontos na notação dat para posterior simulação do crescimento em paralelo. Se a ação exigir o surgimento de uma nova ramificação, inserimos as strings destes novos segmentos na fila, com o ponteiro na posição 1. Caso as strings anteriores não tenham sido lidas completamente, são novamente inseridas na fila e o processo se repete. Em caso contrário, significa que aquele segmento já atingiu o seu crescimento máximo.

Após estas operações, corrigimos a orientação da gramática, aplicando campos de atração. Na Figura 5.2, à direita simulamos a célula com atrações de duas linhas de força. A linha A representa a atração dos dendritos apicais em direção a pia mater, enquanto a linha B representa o crescimento basal em direção à matéria branca, hipocampo e outras áreas. A força de atração dos dendritos basais é bem menor que a força de atração dos apicais. Utilizamos este critério tendo por base as observações da morfologia neural. O formulário no canto superior esquerdo mostra as constantes utilizadas.

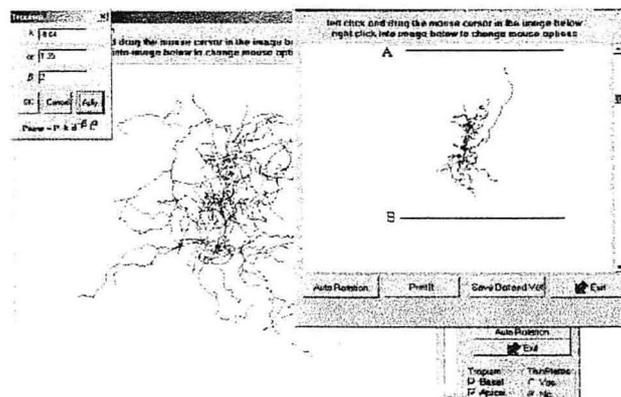


Figura 5.2 – Correção da orientação gramatical. Ao fundo a célula gerada sem as correções. À esquerda as constantes utilizadas, e à direita a célula com simulação de tropismo.

5.4 DESENVOLVIMENTO DO SOFTWARE

O diagrama mostra uma breve descrição das funções e processos desenvolvidos para o software que denotamos por ViewNeuron.

O processo *CONVERT* pode tomar qualquer um dos formatos de arquivo delimitados pela linha pontilhada (dat para o visualizador, swc, asc ou neurolucida ,eut ou eutectic) e converter para os outros restantes. Por exemplo, se tivéssemos um arquivo em formato swc, este pode ser convertido para dat, asc e eut. *CONVERT* utiliza ainda arquivos em formato eut para gerar arquivos formato vet ou vetorial. Após a geração dos arquivos em formato vet o processo pode gerar os arquivos frq ou de frequência.

O processo *VIEW* toma os arquivos em formatos eutectic ou vetorial e visualiza na tela. Os demais formatos podem ser convertidos para eutectic.

O processo *PARSER*, utilizando a gramática, as funções de distribuição e as informações de tropismo fornecidas pelo usuário, chama o processo *VIEW* que permite a visualização das células artificiais geradas.

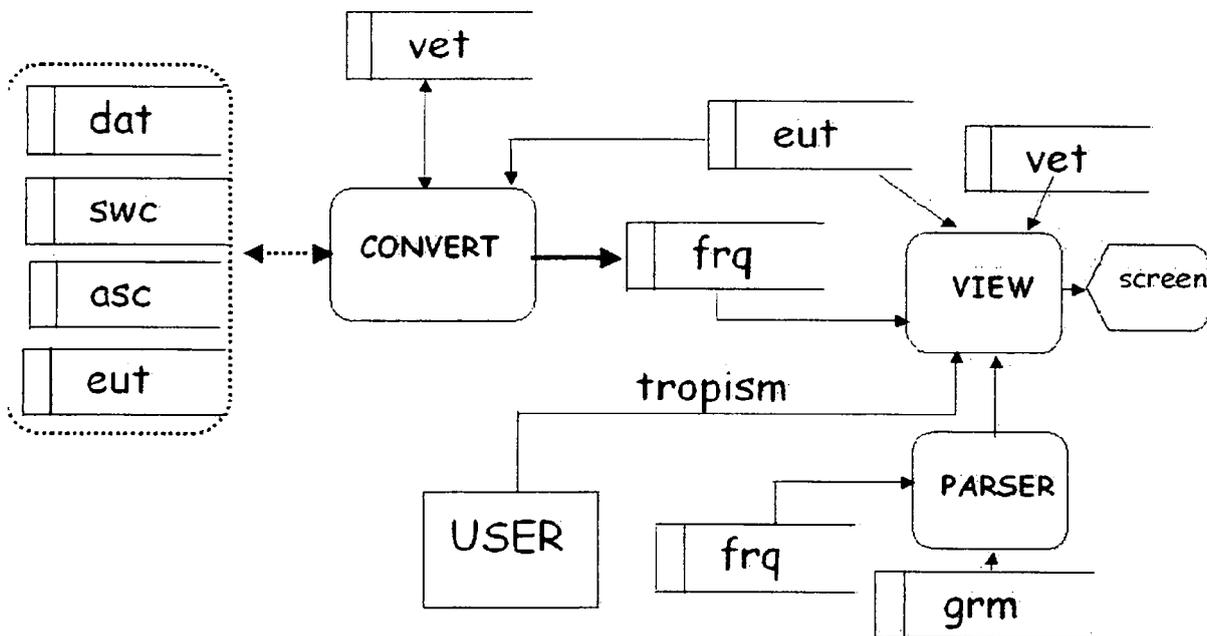


Figura 5.3 – Diagrama de fluxo de dados do software desenvolvido.

A Figura 5.4 apresenta o diagrama hierárquico funcional (DHF) do software desenvolvido. Aqui foram colocadas todas as opções necessárias para visualização de células naturais nos traçados de formato eutectic, dat e vetorial. Embora a visualização de traçado

neural tais como *neuro* e *swc* não seja possível diretamente, pode-se transformar estes formatos nas notações que possibilitam a visualização.

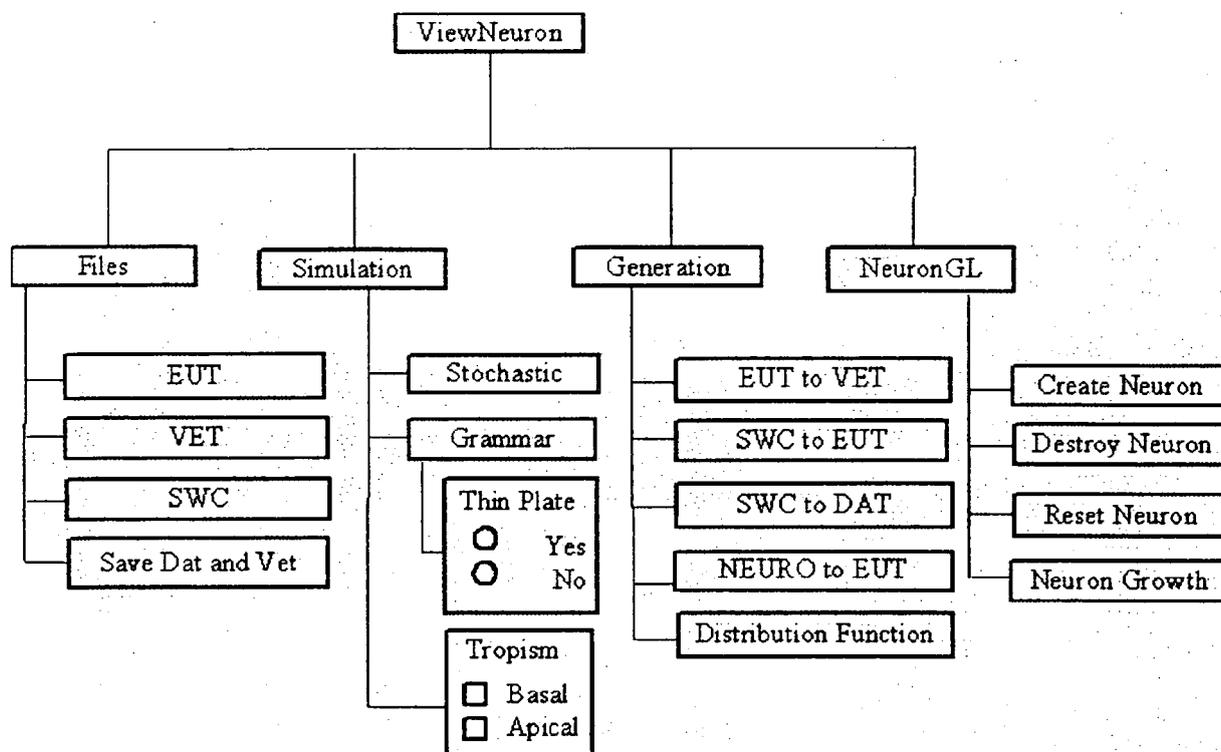


Figura 5.4 – Diagrama hierárquico funcional do software desenvolvido

O menu Files permite a visualização em notação eutectic, vetorial e SWC. (opções EUT, VET e SWC, respectivamente). A opção Save Dat and Vet possibilita ao usuário salvar a célula virtual gerada pela gramática nas notações dat e vetorial. A notação dat servirá para visualização da célula pelo visualizador neural NeuronGL, que utiliza a DLL Neuron3d, e a notação vetorial permitirá a realização de comparações das medidas das células artificiais com as medidas das células naturais.

O menu Simulation permite a geração das células. A opção Stochastic permite a geração de uma célula neural usando simplesmente as funções de distribuição. A opção Grammar permite a geração usando a gramática, na qual pode-se escolher o uso ou não de Thin Plate Splines incorporada na gramática. A opção Tropism permite a reorientação dos dendritos apicais e basais de acordo com as características das células piramidais (aqui utilizamos os procedimentos do algoritmo 3, da seção 4.1). Podemos ainda aqui escolher se queremos tropismo somente nos dendritos apicais ou somente nos basais.

O menu Generation permite a transformação dos traçados e a geração das funções de distribuição. A opção EUT to VET transforma os arquivos em notação eutectic para a notação

vetorial, necessária para levantamento estatístico. As opções SWC to EUT e SWC to DAT transformam os arquivos de notação SWC em eutectic e dat, respectivamente. Já a opção NEURO to EUT, converte para o formato eutectic. Esta opção utiliza os códigos neuroLucida conhecidos e comentados nesta dissertação, ignorando os códigos não conhecidos. Cumpre lembrar que o formato neuroLucida é de um software comercial e portanto, muitos de seus códigos não são divulgados.

Finalmente, o menu NeuronGL chama o visualizador neural, que abre um formulário para visualização de células em notação dat. A opção Create Neuron abre um arquivo em notação dat, Destroy Neuron limpa as informações da memória e Reset Neuron reescreve novamente a célula com as informações padrão. A opção Neuron Growth possibilita a visualização do crescimento neural em paralelo.

A Figura 5.5, mostra o visualizador 3D, com a geração de uma célula virtual.

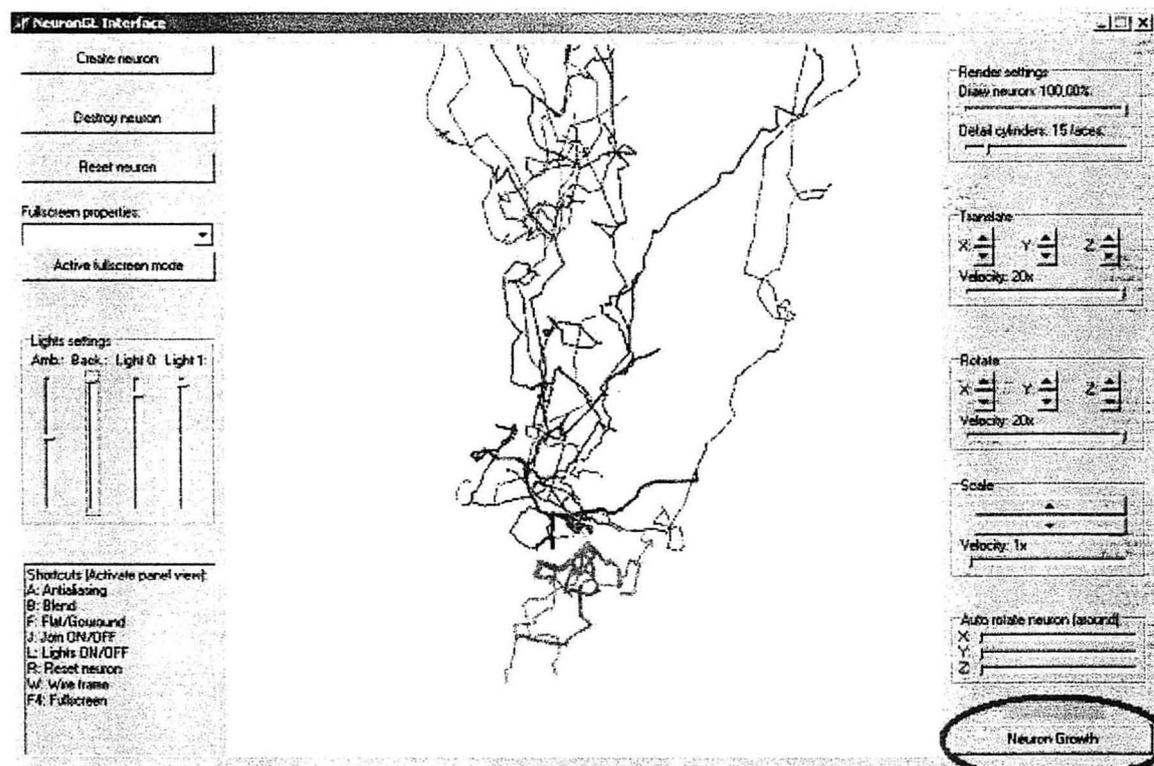


Figura 5.5 – Visualizador neural com uma célula virtual. Em destaque o botão correspondente à simulação do crescimento em paralelo.

Deste modo, podemos visualizar seu crescimento vendo todos os dendritos crescendo. Na Figura 5.6 podemos ver a mesma célula da Figura 5.5 com 15%, 30% e 60% do

seu crescimento. Usando a opção Neuron Growth podemos ver as diversas fases do crescimento neural.

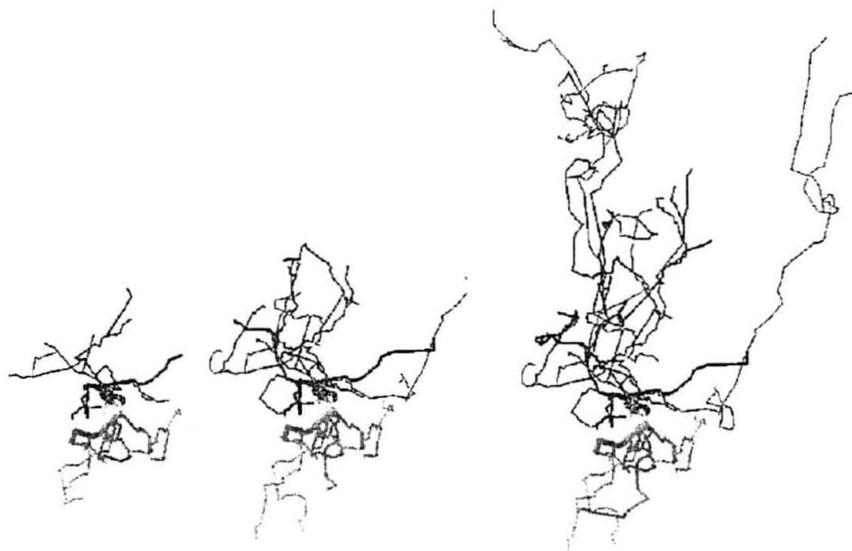


Figura 5.6 – Porcentagem de crescimento. Da esquerda para a direita, 15%, 30% e 60% do seu crescimento dendrítico final.

A seguir temos na Figura 5.7, algumas células geradas pelo programa, sendo a primeira célula a mesma da Figura 5.6. As células da parte de cima utilizam funções de distribuição, e as de baixo utilizam *Thin Plate Splines*.

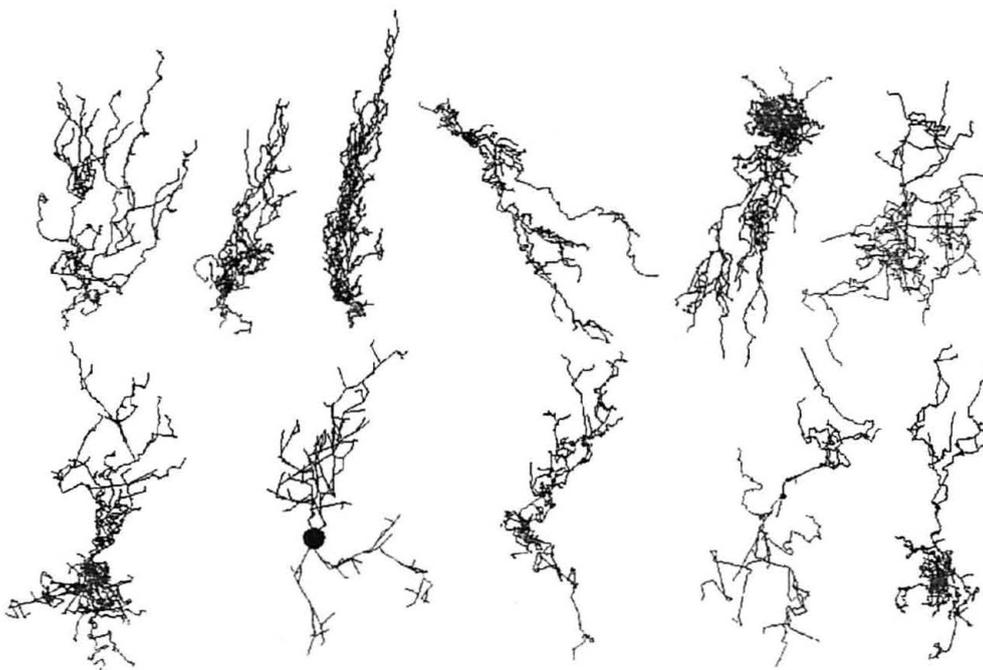


Figura 5.7 – Algumas células artificiais geradas. A primeira célula é a mesma da Figura 5.6, mostrada sem o uso do visualizador 3D.

As Thin Plate Splines desenvolvidas pelo aluno Mitio Yoshida necessitavam de algumas correções. Em alguns momentos a curva da *spline* apresentava mínimos ou máximos locais que não condiziam com a idéia de suavidade da superfície. Ou seja, uma vez que temos dois pontos conhecidos x e z , temos os valores para $F(x)$ e $F(z)$. Desejamos encontrar então um valor para $F(y)$, sendo $x \leq y \leq z$, e esperamos que a curva que passe por $F(x)$, $F(y)$ e $F(z)$ apresente-se de modo suave. Quando esse erro ocorria, a solução encontrada foi aproximar o valor de $F(y)$ de modo que a curva entre $F(x)$, $F(y)$ e $F(z)$ fosse uma linha reta. Deste modo, temos uma reta $y = ax + b$ que passa entre $F(x)$ e $F(z)$, e fica fácil encontrar um melhor valor para $F(y)$.

5.5 VALIDAÇÕES

A seguir, comentaremos alguns gráficos obtidos, analisando suas características. Os gráficos restantes encontram-se nos apêndices de B, C, D e E. Criamos funções extraídas com base em medidas de 68 células naturais, e geramos para nossos testes 50 células artificiais. Comentaremos nesta seção gráficos para tamanho de segmento, comprimento de arco, torção e curvatura.

5.5.1 Tamanho de segmento

A Figura 5.8 mostra do lado esquerdo os gráficos para segmentos de células naturais e do lado direito células artificiais.

Nas células naturais, o nível máximo atingido foi de 24 para dendritos basais e 36 para dendritos apicais, com tamanhos máximos de 790 e 585, respectivamente. Nas células artificiais, atingimos níveis máximos de 18 para basais e 30 para apicais, com tamanhos máximos de 488 e 496 respectivamente.

A razão de não termos atingidos os últimos níveis e tamanhos é que dificilmente é sorteado um número aleatório tão grande, já que para estes valores a probabilidade acumulada se aproxima de 1.

Percebe-se que os gráficos das células artificiais têm uma certa semelhança com as naturais, mostrando assim, que o sorteio do tamanho de segmento se aproxima do tamanho de células naturais.

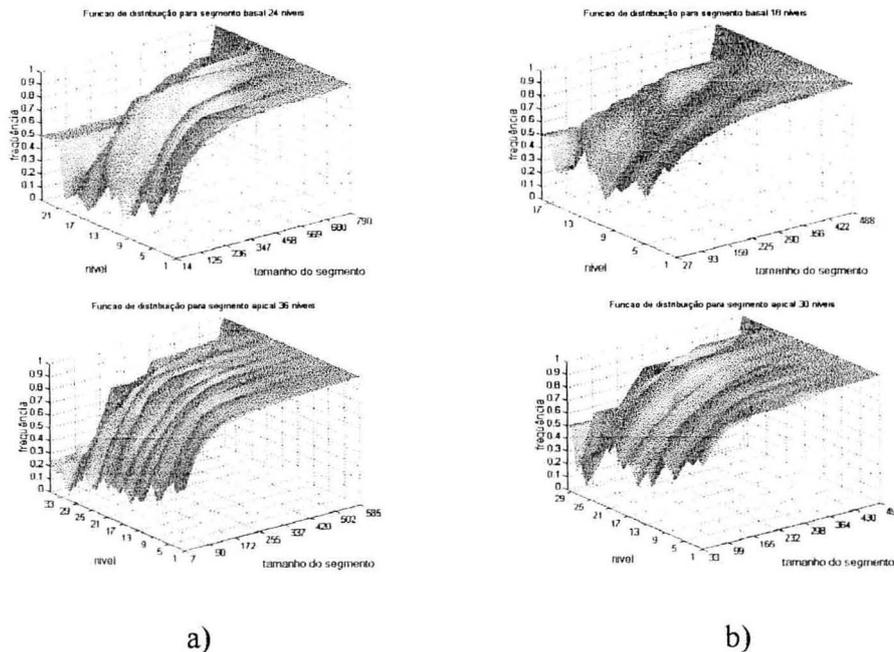


Figura 5.8 – Gráficos para medidas de segmento. a) células naturais e b) células artificiais.

5.5.2 Comprimento de arco

A Figura 5.9 mostra os gráficos para comprimento de arco. Nas células naturais, o tamanho máximo de arco é de 80 para basais e 92 para os apicais. Nas células artificiais, vemos 49 para basais e 56 para apicais. Do mesmo modo que para os tamanhos de segmentos alguns tamanhos dificilmente são atingidos na simulação, devido ao valor próximo de 1 da probabilidade acumulada. Em cada nível, cada tamanho tem uma certa probabilidade. Quando a probabilidade sorteada é bem menor do que a probabilidade da FD, temos que calcular este tamanho. Para calcular este tamanho, determinamos o coeficiente angular e linear da reta formada entre os dois limites da caixa de nível n . Quando o nível é 1, consideramos o valor de comprimento mínimo como 0 e com probabilidade 0. Deste modo, quando a frequência de valores altos é muito grande, pode acontecer de sortearmos valores bem menores do que esta frequência. Quando o valor sorteado é muito pequeno, pode acontecer de ser calculado um valor muito pequeno. Mas isso só acontece no nível 1 conforme se pode ver nos gráficos do lado direito da figura. Mesmo assim vemos que os gráficos dos dois lados apresentam uma certa semelhança.

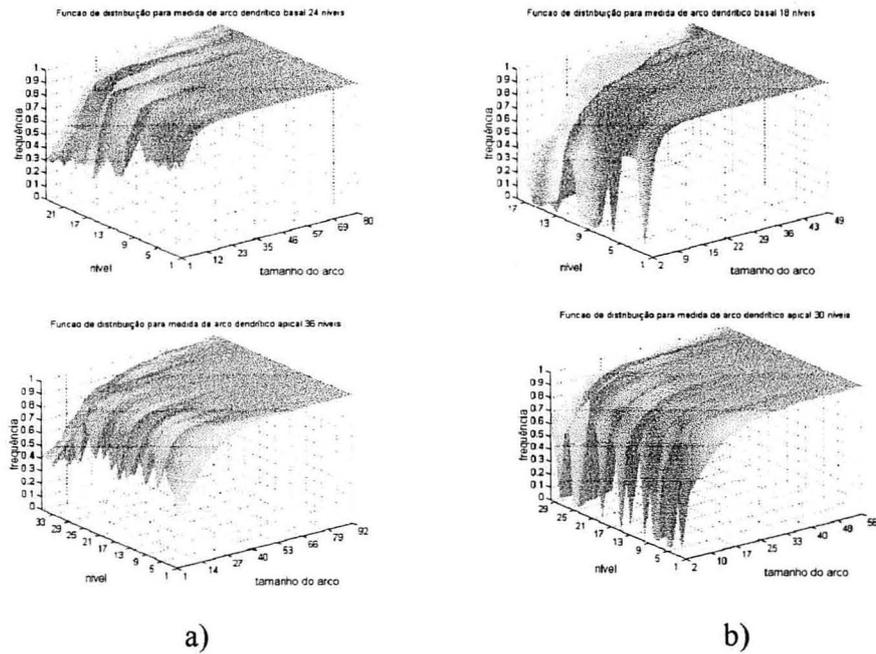


Figura 5.9 – Gráficos para medidas de arco. a) células naturais e b) células artificiais.

5.5.3 Curvatura

As medidas de curvatura são as que mais apresentam semelhanças, já que existem valores com freqüências acumuladas bem próximas de 0 e bem próximas de 1. Na Figura 5.10, as células naturais apresentam valores em graus de -156 a 157 para basais e apicais. Estes valores foram convertidos de radianos para graus, ocasionando alguns arredondamentos, como no caso das artificiais onde encontramos valores entre -156 e 158 .

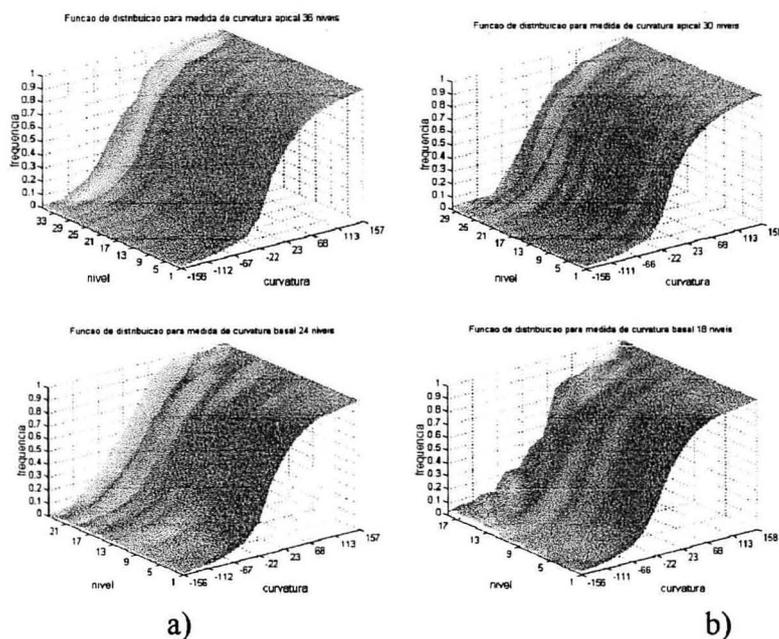


Figura 5.10 – Gráficos para curvatura de arco. a) células naturais e b) células artificiais.

5.5.4 Torção

Os gráficos da Figura 5.11 mostram, para as células naturais, ocorrências de torção entre -68 e 79 graus para basais e -70 a 79 graus para apicais. Para as células artificiais, temos valores entre -78 e 79 para basais e -70 a 79 para apicais. O gráfico de torção basal das células artificiais apresenta algumas rugosidades em sua superfície. Como a torção é dependente do sorteio anterior de curvatura muitas vezes podem ocorrer sorteios de curvaturas nas quais não temos nenhuma torção. Temos então de escolher a torção da classe ou caixa mais próxima. Mesmo assim a figura ainda apresenta uma superfície em forma de 's' semelhante ao gráfico da célula natural.

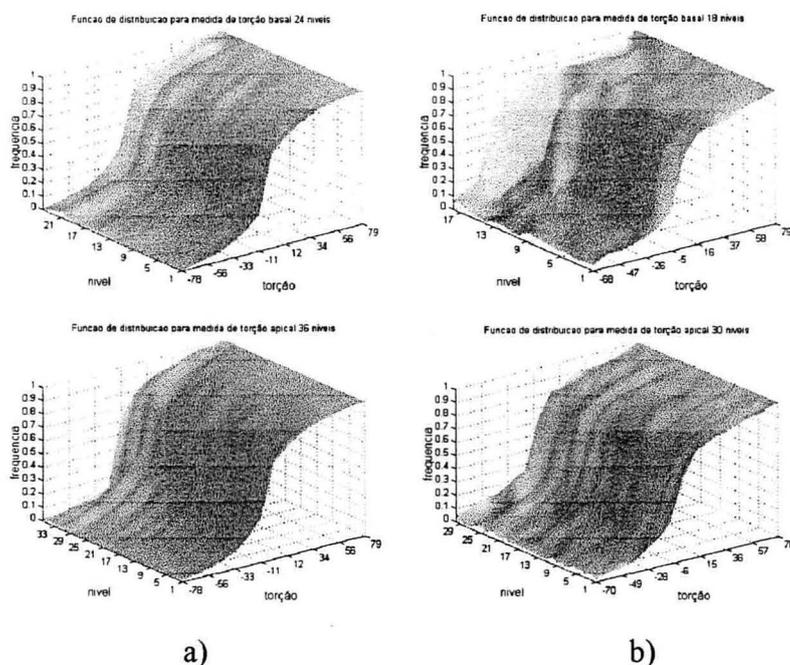


Figura 5.11– Gráficos para torção de arco. a) células naturais e b) células artificiais.

REFERÊNCIAS BIBLIOGRÁFICAS

AHO, A. V.; SETHI, R.; ULLMAN, J. D. **Compiladores: princípios técnicas e ferramentas**. São Paulo: Makron Books, 1999.

ALMEIDA, M. S. Visualização Tridimensional de Estruturas Neurais Artificiais Morfológicamente Realistas, **Relatório Técnico de bolsa PIBIC**, Universidade Estadual de Maringá, 2002.

ARANGO, H. G. **Bioestatística teórica e computacional**. Rio, Guanabara Koogan, 2001.

ASCOLI, G.A; KRICHMAR, J. L. L-neuron: A modeling tool for the efficient generation and parsimonious description of dendritic morphology. **Neurocomputing**, v. 32-33 (2000) 1003-1011.

BANKS, J.; CARSON, J. S. II; NELSON, B. L. **Discrete-event system simulation**. USA, Prentice Hall, 2nd ed, 1996.

BOLDRINI, J.L; COSTA, S. I. R; FIGUEIREDO, V. L. ; WETZLER, H. G. **Álgebra Linear**. São Paulo, HARBRA, 1984.

BONDY, J.A.; MURTY, U. S. R. **Graph theory with applications**. USA, North-Holland, 1979.

BALAKRISHNAN, V. K. **Graph theory**. USA, McGraw Hill, 2000.

CASTANHO, J. C.; TOZZI, C. L. Obtenção da transformação de corpo rígido em registro de imagens com deformação usando thin plate splines. In: ANAIS DO XI SIBGRAPI, 1996, p. 181-187.

COELHO, R. C.; COSTA, L. da F. Realistic neuromorphic models and their application to neural reorganization simulation. **Neurocomputing**, USA, 2002.

COELHO, R. C. **Síntese, modelagem e simulação de estruturas neurais morfológicamente realísticas**. São Carlos, 1998. (Tese de doutorado) – Instituto de Física de São Carlos, Universidade de São Paulo.

CONSULARO, L. A. **Técnicas de Mineração de Dados para Análise de Imagens**. São Carlos, 2000. (Tese de Doutorado) – Instituto de Física de São Carlos, Universidade de São Paulo.

COSTA, L. da F.; CESAR Jr., R. M. **Shape analysis and classification : theory and practice**. USA, CRC Press, 2001.

COSTA, L. da F.; OLIVEIRA, M. C. F. de ; MINGHIM, R. De olho nos neurônios: as 'formas' virtuais do pensamento. **Ciência Hoje**, v. 28, n. 167, p. 44-51, dez. 2000.

DOWLING, J. E. **Neurons and Networks: An Introduction to Neuroscience**. USA: The Belknap Press of Harvard University Press, 1992.

HAMILTON, P. A language to describe the growth of neurites. **Biologic Cybernetic**, USA, v. 68, 1993. p. 559-565.

HARARY, Frank. **Graph theory**. USA, Addison-Wesley, 1998.

HUBEL, D. H. **Eye, brain and vision**. USA: Scientific American Library, 1995.

JAQUES, O. V.; COELHO, R. C. Synthesis of Neural Cells using L-System. In: SIBGRAPI 2002 – XV SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, Fortaleza, 2002. **Proceedings of SIBGRAPI 2002**. Los Alamitos: Califórnia: IEEE Computer Society, 2002, p. 431.

JAQUES, O. V.; COELHO, R. C. Synthesis of 3D structures using graph grammars. In: ICOBICOB-INTERNATIONAL CONFERENCE IN BIOINFORMATIC AND COMPUTATIONAL BIOLOGY. Ribeirão Preto, 2003 (a).

JAQUES, O. V.; COELHO R. C. The use of Thin Plate Splines interpolation to auxiliary in the generation of neural cells. In: SIBGRAPI 2003 - XVI SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO E PROCESSAMENTO DE IMAGENS. São Carlos, SP, 2003(b). *In press*.

JAQUES, O. V.; COELHO, R. C. Generating Three Dimensional Neural Cells Based on Bayes Rules and interpolation with Thin Plate Splines. In: 8TH IBEROAMERICAN CONGRESS ON PATTERN RECOGNITION. Havana, Cuba, 2003 (c). *In press*.

LUND, J. S. Organizations of neurons in the visual cortex of the monkey (Macaca Mulata). **Journal Comp. Neurol**, v. 157, p.455-496, 1973

LUNDY-EKMAN, L. **Neurociência e fundamentos para reabilitação**. Rio: Guanabara Koogan, 2000.

McALLISTER, A. K.; LO, D. C.; KATZ, L.C. Neurotrophins regulate dendritic growth in developing visual cortex. **Neuron**, v. 15, oct. 1995, p.791-803

McCORMICK, B.H.; MULCHANDANI, K. L-system modeling of neurons visualization. In: I BIOMEDICAL COMPUTING CONFERENCE PROCEEDINGS, 1994, p. 693-705.

MENESES, P. B. **Linguagens formais e autômatos**. Porto Alegre: Sagra-Luzato – Instituto de Informática da UFRGS, 2000. (2 Ed. Série Livros Didáticos)

POLLEUX, F; MORROW, T.; GHOSH, A.. Semaphorin 3A is a chemoattractant for cortical apical dendrites. **Nature**, v 404, 2000, p. 567-573.

POLLLEUX, F.; GIGER, R. J.; GINTY, D. D.; KOLODIKIN; GHOSH, A. Patterning of cortical efferent projections by semaphoring-neuropilin interactions. *Science*, v. 282, dec, 1998, p. 1904-1906.

PRICE, A. M.; TOSCANI, S. S. **Implementação de linguagens de programação: Compiladores**. Porto Alegre: Sagra-Luzato – Instituto de Informática da UFRGS, 2001. (2 Ed. Série Livros Didáticos)

PRUSINKIEWICZ, P. Applications of L-systems to computer imagery. In: *COMPUTER SCIENCE – 3rd INTERNATIONAL WORKSHOP ON GRAPH-GRAMMARS AND THEIR APPLICATION TO COMPUTER SCIENCE*, v.291, Dec. 1986. . 534-348

PURVES, D. **Body and brain: a trophic theory of neural connections**. USA: Harvard Universit Press, 1988.

SCHMOLESKY, M. **The primary visual córtex**. Disponível em: <http://webvision.med.utah.edu/VisualCortex.html> Acesso em: 22 jun. 2000.

SERMASI, E.; TROPEA, D.; DOMENICI, L. A new form of synaptic plasticity in transiently expressed in the developing rat visual cortex: a modulatory role for visual experience and brain derived neurotrophic factor. *Science*, v. 91, n. 1, 1999, p. 163-173.

TANAKA, J. S. **Um modelo neuomórfico de estimativa de orientação e curvatura**. São Carlos, 2000. (Tese de doutorado) – Instituto de Física de São Carlos, Universidade de São Paulo.

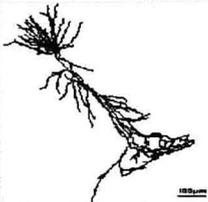
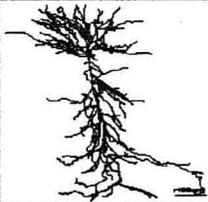
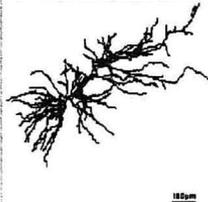
VALENTIN, L. G. **Representação de Células e Estruturas Neurais Bi e Tridimensionais, Relatório Técnico de bolsa PIBIC**, Universidade Estadual de Maringá, 2002.

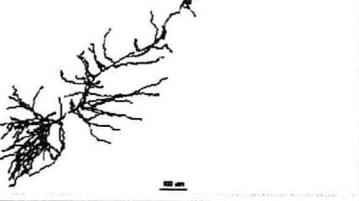
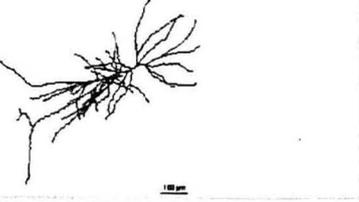
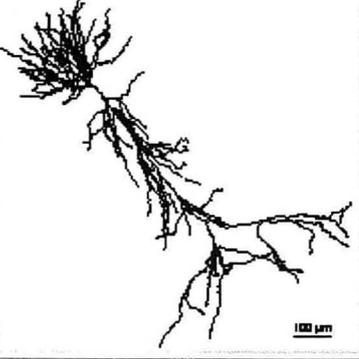
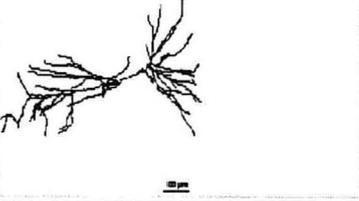
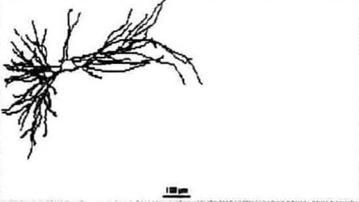
WHITFORD, K. L.; DJKHUIZEN, P.; POLEUX, F.; GHOSH, A. Molecular control of cortical dendrite development. *Annu. Rev. Neurosci.* , 2002, p. 27 –152.

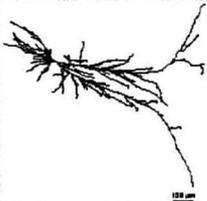
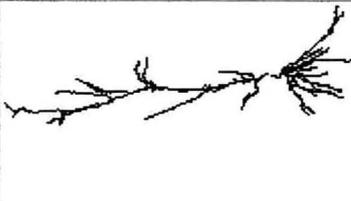
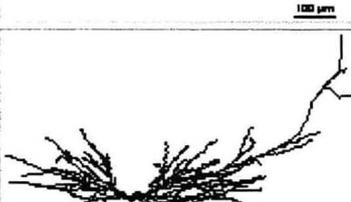
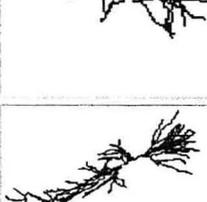
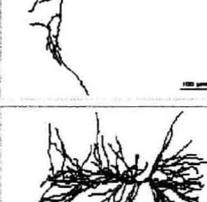
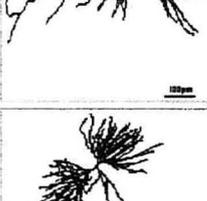
APENDICE A

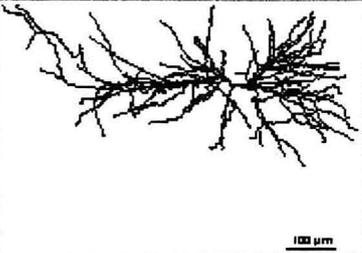
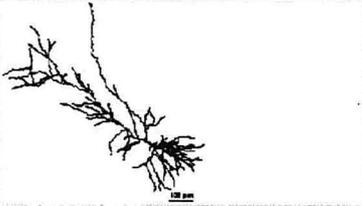
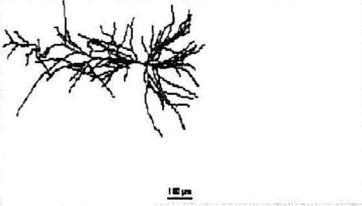
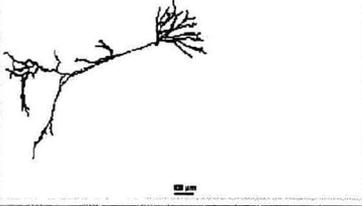
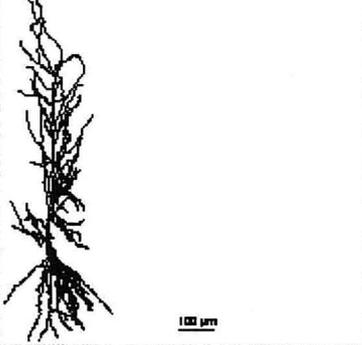
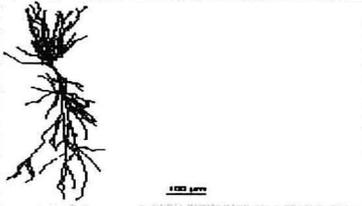
CÉLULAS NATURAIS UTILIZADAS PARA EXTRAÇÃO DAS MEDIDAS

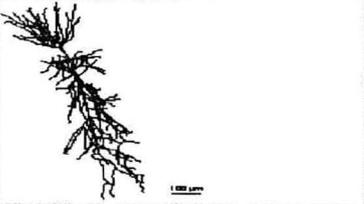
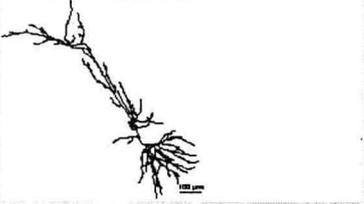
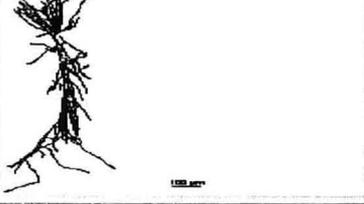
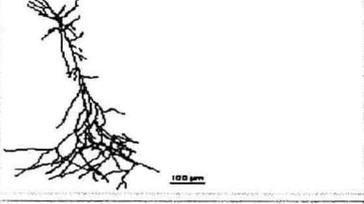
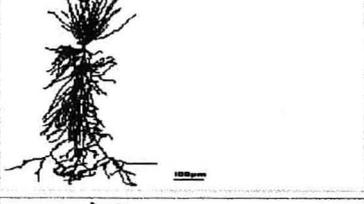
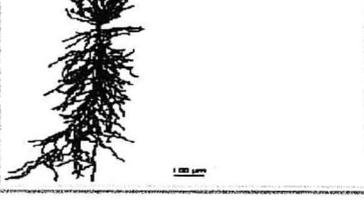
(<http://www.cns.soton.ac.uk/~jchad/cellArchive/index/topindex.html>. Acesso em 23 de junho de 2001).

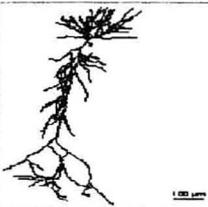
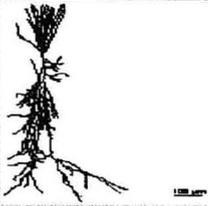
rat	<i>cell details</i>	<i>swc file</i>	<i>notes</i>	
hippocampus				
cal				
cal pyramidal cell in vivo young	n400	n400.swc	turner_p_cal.txt	
	n401	n401.swc	turner_p_cal.txt	
	n402	n402.swc	turner_p_cal.txt	
	n403	n403.swc	turner_p_cal.txt	
	n404	n404.swc	turner_p_cal.txt	

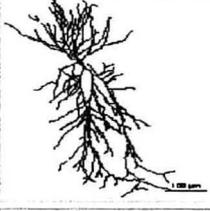
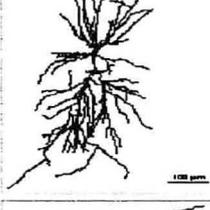
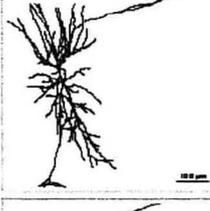
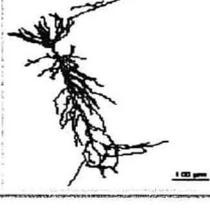
<p><u>n405</u></p>	<p><u>n405.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n406</u></p>	<p><u>n406.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n407</u></p>	<p><u>n407.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n408</u></p>	<p><u>n408.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n409</u></p>	<p><u>n409.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n410</u></p>	<p><u>n410.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n411</u></p>	<p><u>n411.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	

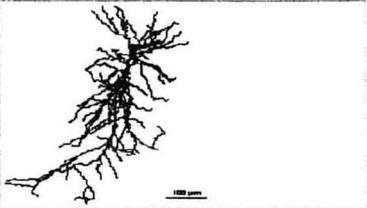
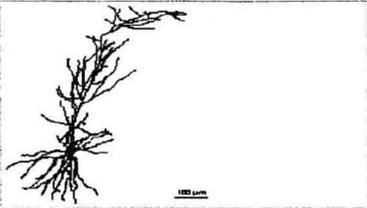
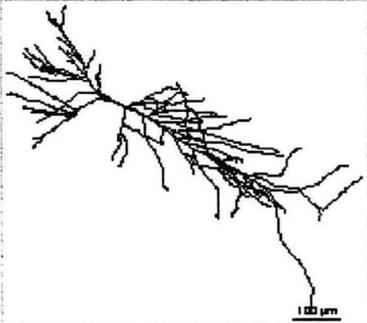
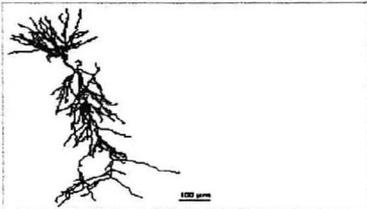
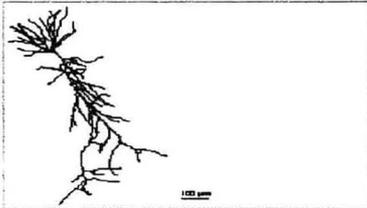
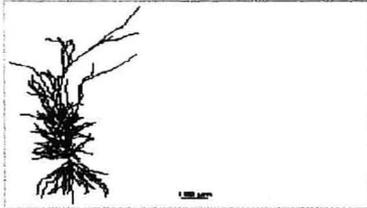
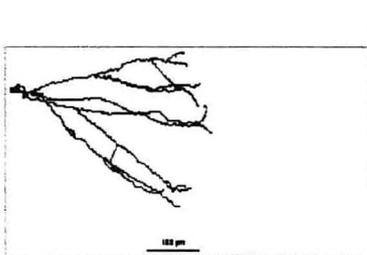
n412	n412.swc	turner_p_cal.txt	
n413	n413.swc	turner_p_cal.txt	
n414	n414.swc	turner_p_cal.txt	
n415	n415.swc	turner_p_cal.txt	
n416	n416.swc	turner_p_cal.txt	
n417	n417.swc	turner_p_cal.txt	
n418	n418.swc	turner_p_cal.txt	

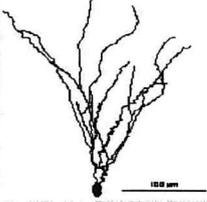
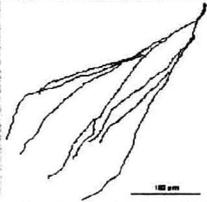
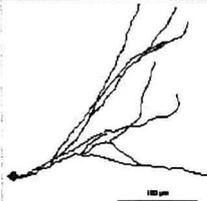
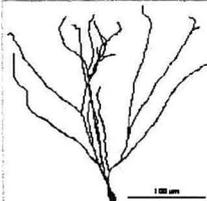
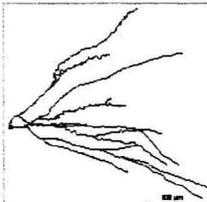
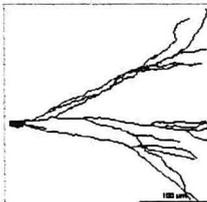
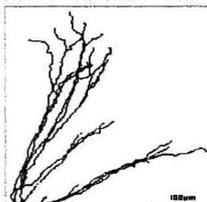
	n419	n419.swc	turner_p_cal.txt	
	n420	n420.swc	turner_p_cal.txt	
	n421	n421.swc	turner_p_cal.txt	
	n422	n422.swc	turner_p_cal.txt	
	n423	n423.swc	turner_p_cal.txt	
ca1 pyramidal cell in vitro aged	n170	n170.swc	turner_p_cal.txt	
	n171	n171.swc	turner_p_cal.txt	

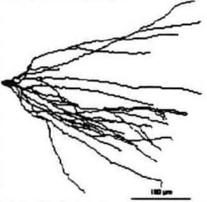
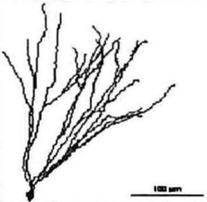
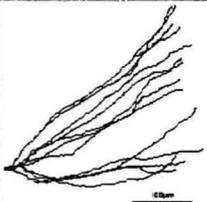
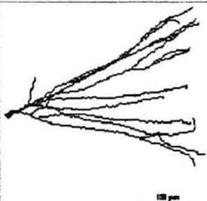
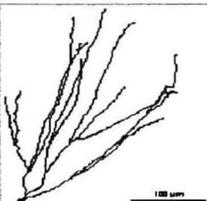
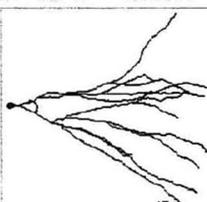
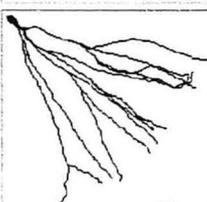
<p><u>n172</u></p>	<p><u>n172.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n173</u></p>	<p><u>n173.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n174</u></p>	<p><u>n174.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n175</u></p>	<p><u>n175.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n176</u></p>	<p><u>n176.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n177</u></p>	<p><u>n177.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n178</u></p>	<p><u>n178.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	
<p><u>n179</u></p>	<p><u>n179.swc</u></p>	<p><u>turner_p_cal.txt</u></p>	

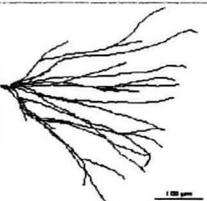
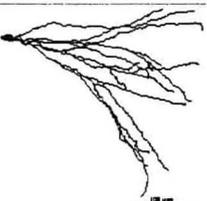
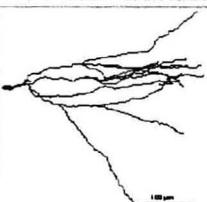
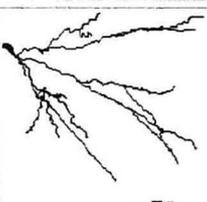
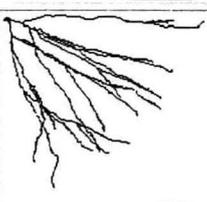
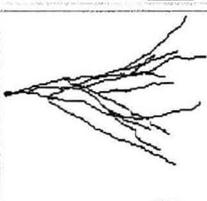
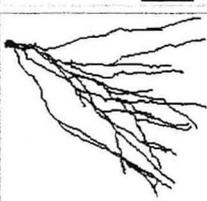
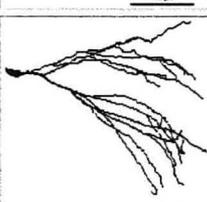
	n180	n180.swc	turner_p_cal.txt	
	n181	n181.swc	turner_p_cal.txt	
	n182	n182.swc	turner_p_cal.txt	
	n183	n183.swc	turner_p_cal.txt	
	n184	n184.swc	turner_p_cal.txt	
ca1 pyramidal cell in vitro ka lesion	n140	n140.swc	turner_p_cal.txt	
	n141	n141.swc	turner_p_cal.txt	

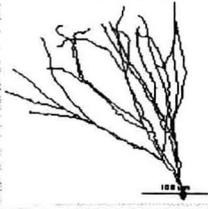
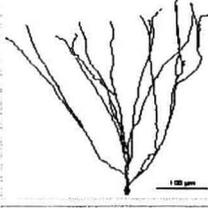
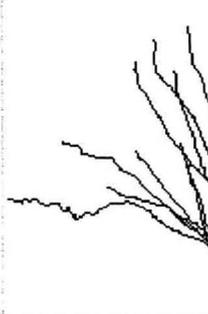
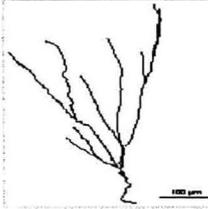
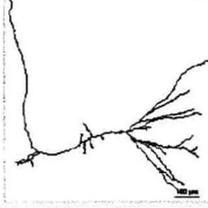
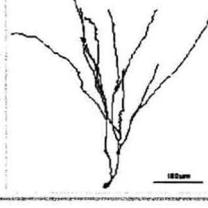
	<u>n142</u>	<u>n142.swc</u>	<u>turner_p_cal.txt</u>	
	<u>n143</u>	<u>n143.swc</u>	<u>turner_p_cal.txt</u>	
	<u>n144</u>	<u>n144.swc</u>	<u>turner_p_cal.txt</u>	
	<u>n145</u>	<u>n145.swc</u>	<u>turner_p_cal.txt</u>	
cal pyramidal cell in vitro young	<u>n120</u>	<u>n120.swc</u>	<u>turner_p_cal.txt</u>	
	<u>n121</u>	<u>n121.swc</u>	<u>turner_p_cal.txt</u>	
	<u>n122</u>	<u>n122.swc</u>	<u>turner_p_cal.txt</u>	
	<u>n123</u>	<u>n123.swc</u>	<u>turner_p_cal.txt</u>	

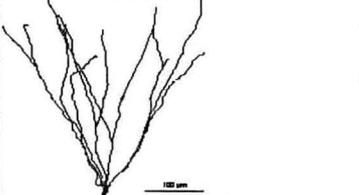
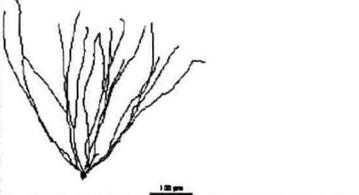
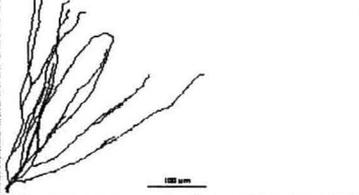
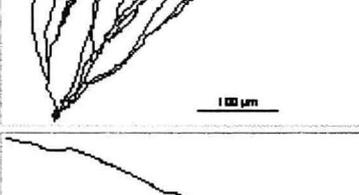
		n125	n125.swc	turner_p_cal.txt	
		n126	n126.swc	turner_p_cal.txt	
		n127	n127.swc	turner_p_cal.txt	
		n128	n128.swc	turner_p_cal.txt	
		n129	n129.swc	turner_p_cal.txt	
		n130	n130.swc	turner_p_cal.txt	
dg	dg granule cell in vitro ec lesion	n240	n240.swc	turner_g_dg.txt	

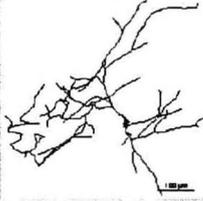
	<u>n241</u>	<u>n241.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n242</u>	<u>n242.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n243</u>	<u>n243.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n244</u>	<u>n244.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n245</u>	<u>n245.swc</u>	<u>turner_g_dg.txt</u>	
dg granule cell in vivo young	<u>n500</u>	<u>n500.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n501</u>	<u>n501.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n502</u>	<u>n502.swc</u>	<u>turner_g_dg.txt</u>	

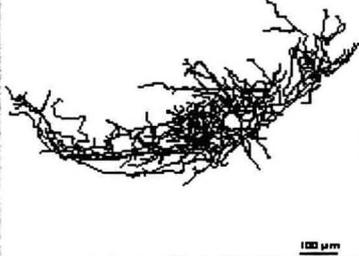
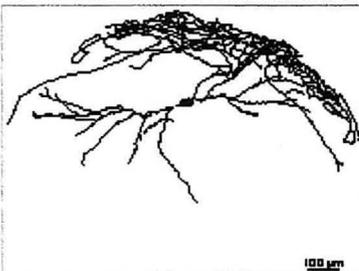
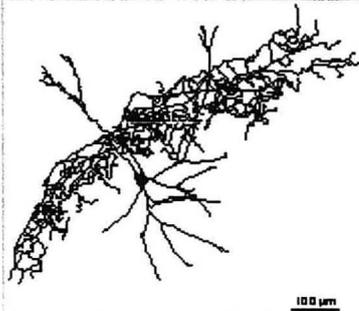
<u>n503</u>	<u>n503.swc</u>	<u>turner_g_dg.txt</u>	
<u>n504</u>	<u>n504.swc</u>	<u>turner_g_dg.txt</u>	
<u>n505</u>	<u>n505.swc</u>	<u>turner_g_dg.txt</u>	
<u>n506</u>	<u>n506.swc</u>	<u>turner_g_dg.txt</u>	
<u>n507</u>	<u>n507.swc</u>	<u>turner_g_dg.txt</u>	
<u>n508</u>	<u>n508.swc</u>	<u>turner_g_dg.txt</u>	
<u>n509</u>	<u>n509.swc</u>	<u>turner_g_dg.txt</u>	
<u>n510</u>	<u>n510.swc</u>	<u>turner_g_dg.txt</u>	

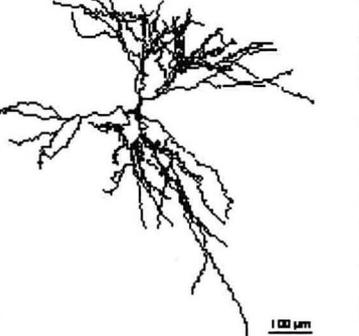
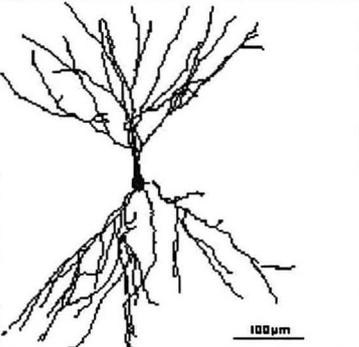
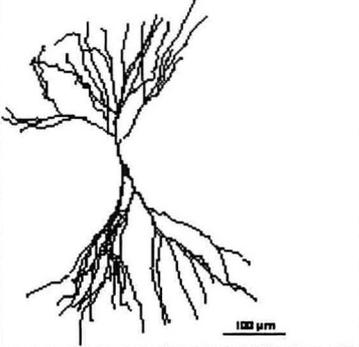
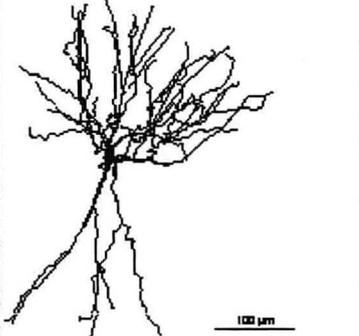
<u>n511</u>	<u>n511.swc</u>	<u>turner_g_dg.txt</u>	
<u>n512</u>	<u>n512.swc</u>	<u>turner_g_dg.txt</u>	
<u>n513</u>	<u>n513.swc</u>	<u>turner_g_dg.txt</u>	
<u>n514</u>	<u>n514.swc</u>	<u>turner_g_dg.txt</u>	
<u>n515</u>	<u>n515.swc</u>	<u>turner_g_dg.txt</u>	
<u>n516</u>	<u>n516.swc</u>	<u>turner_g_dg.txt</u>	
<u>n517</u>	<u>n517.swc</u>	<u>turner_g_dg.txt</u>	
<u>n518</u>	<u>n518.swc</u>	<u>turner_g_dg.txt</u>	

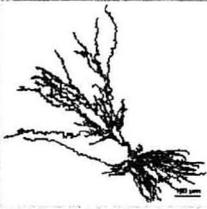
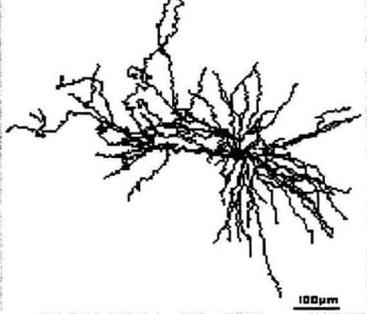
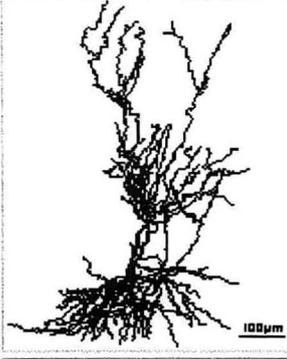
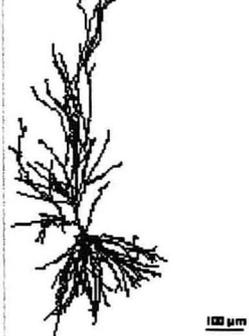
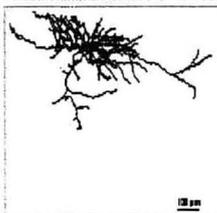
dg granule cell in vitro aged	<u>n270</u>	<u>n270.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n271</u>	<u>n271.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n272</u>	<u>n272.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n273</u>	<u>n273.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n274</u>	<u>n274.swc</u>	<u>turner_g_dg.txt</u>	
	<u>n275</u>	<u>n275.swc</u>	<u>turner_g_dg.txt</u>	
dg granule cell in vitro young	<u>n220</u>	<u>n220.swc</u>	<u>turner_g_dg.txt</u>	

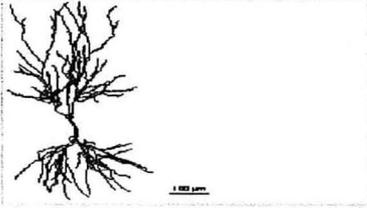
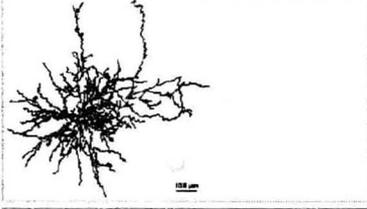
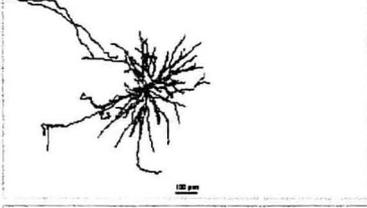
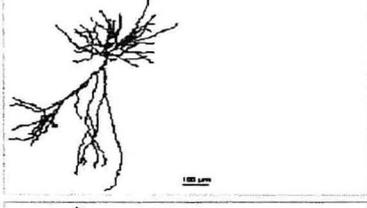
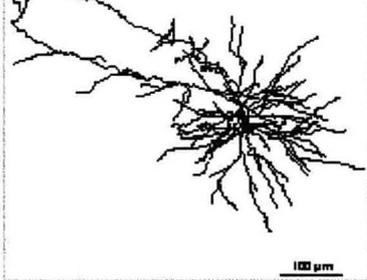
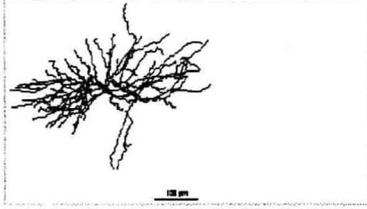
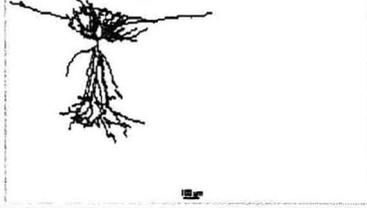
<p><u>n221</u></p>	<p><u>n221.swc</u></p>	<p><u>turner_g_dg.txt</u></p>	
<p><u>n222</u></p>	<p><u>n222.swc</u></p>	<p><u>turner_g_dg.txt</u></p>	
<p><u>n223</u></p>	<p><u>n223.swc</u></p>	<p><u>turner_g_dg.txt</u></p>	
<p><u>n224</u></p>	<p><u>n224.swc</u></p>	<p><u>turner_g_dg.txt</u></p>	
<p><u>n225</u></p>	<p><u>n225.swc</u></p>	<p><u>turner_g_dg.txt</u></p>	
<p><u>n226</u></p>	<p><u>n226.swc</u></p>	<p><u>turner_g_dg.txt</u></p>	

tml dg interneuron in vitro young	n257	n257.swc	turner_i_dg.txt	
	n258	n258.swc	turner_i_dg.txt	
	n259	n259.swc	turner_i_dg.txt	
	n260	n260.swc	turner_i_dg.txt	
	n261	n261.swc	turner_i_dg.txt	
	n262	n262.swc	turner_i_dg.txt	
iml dg interneuron in vitro young	n253	n253.swc	turner_i_dg.txt	

	<u>n254</u>	<u>n254.swc</u>	<u>turner_i_dg.txt</u>	 <p>100 μm</p>
	<u>n255</u>	<u>n255.swc</u>	<u>turner_i_dg.txt</u>	 <p>100 μm</p>
	<u>n256</u>	<u>n256.swc</u>	<u>turner_i_dg.txt</u>	 <p>100 μm</p>
oml dg interneuron in vitro young	<u>n251</u>	<u>n251.swc</u>	<u>turner_i_dg.txt</u>	 <p>100 μm</p>
	<u>n252</u>	<u>n252.swc</u>	<u>turner_i_dg.txt</u>	 <p>100 μm</p>
gcl dg interneuron in vitro young	<u>n250</u>	<u>n250.swc</u>	<u>turner_i_dg.txt</u>	 <p>100 μm</p>

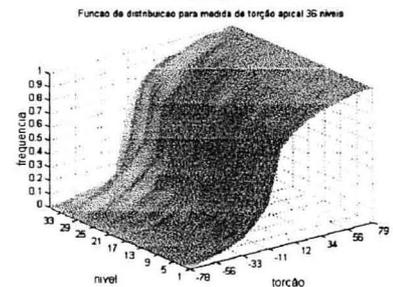
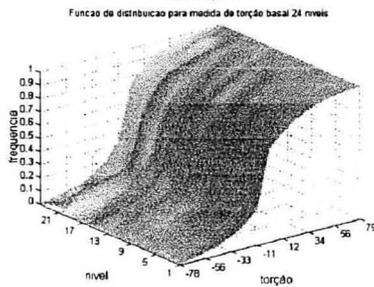
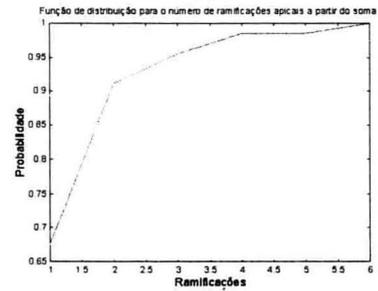
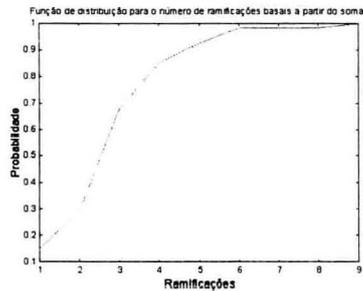
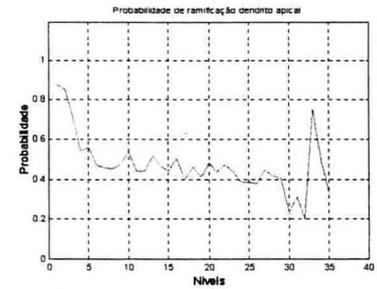
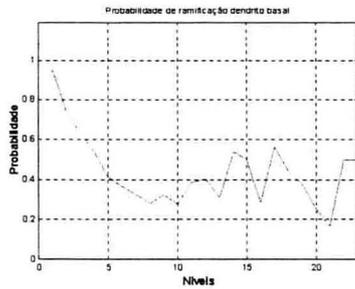
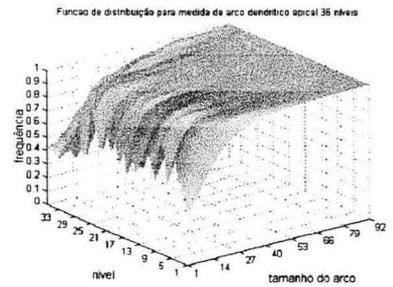
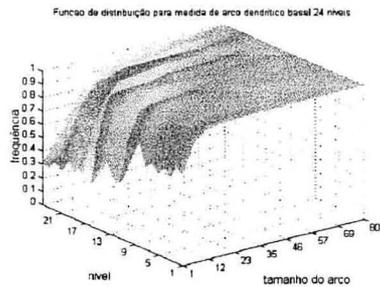
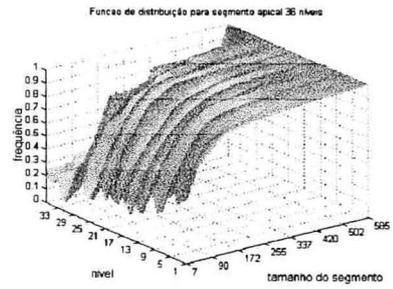
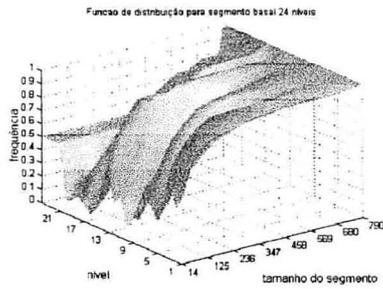
ca3				
ca3c pyramidal cell in vivo young	<u>116</u>	<u>116.swc</u>	<u>turner_p_ca3.txt</u>	
	<u>122</u>	<u>122.swc</u>	<u>turner_p_ca3.txt</u>	
	<u>171</u>	<u>171.swc</u>	<u>turner_p_ca3.txt</u>	
	<u>182</u>	<u>182.swc</u>	<u>turner_p_ca3.txt</u>	
p/v ca3 pyramidal cell in vivo young	<u>148b</u>	<u>148b.swc</u>	<u>turner_p_ca3.txt</u>	

	<u>156b</u>	<u>156b.swc</u>	<u>turner_p_ca3.txt</u>	
	<u>160a</u>	<u>160a.swc</u>	<u>turner_p_ca3.txt</u>	
ca3b pyramidal cell in vivo young	<u>118</u>	<u>118.swc</u>	<u>turner_p_ca3.txt</u>	
	<u>124b</u>	<u>124b.swc</u>	<u>turner_p_ca3.txt</u>	
	<u>151</u>	<u>151.swc</u>	<u>turner_p_ca3.txt</u>	
	<u>155</u>	<u>155.swc</u>	<u>turner_p_ca3.txt</u>	

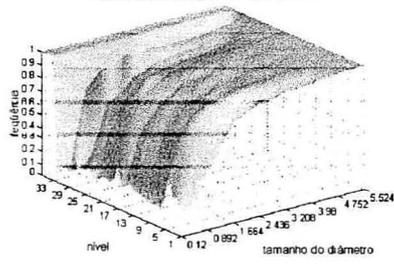
	156a	156a.swc	turner_p_ca3.txt	
ca3a pyramidal cell in vivo young	110	110.swc	turner_p_ca3.txt	
	148a	148a.swc	turner_p_ca3.txt	
	160b	160b.swc	turner_p_ca3.txt	
	164	164.swc	turner_p_ca3.txt	
zone 3 hilar cell in vivo young	124a	124a.swc	turner_p_ca3.txt	
ca2 pyramidal cell in vivo young	169	169.swc	turner_p_ca3.txt	

APÊNDICE B

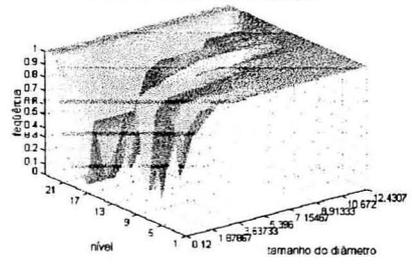
GRÁFICOS DE MEDIDAS DE CÉLULAS PIRAMIDAIS DO CORTEX VISUAL DE RATO



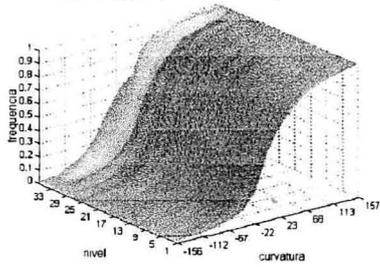
Funcao de distribucao para diametro apical 36 niveis



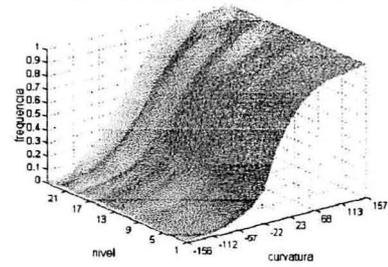
Funcao de distribucao para diametro basal 24 niveis



Funcao de distribucao para medida de curvatura apical 36 niveis



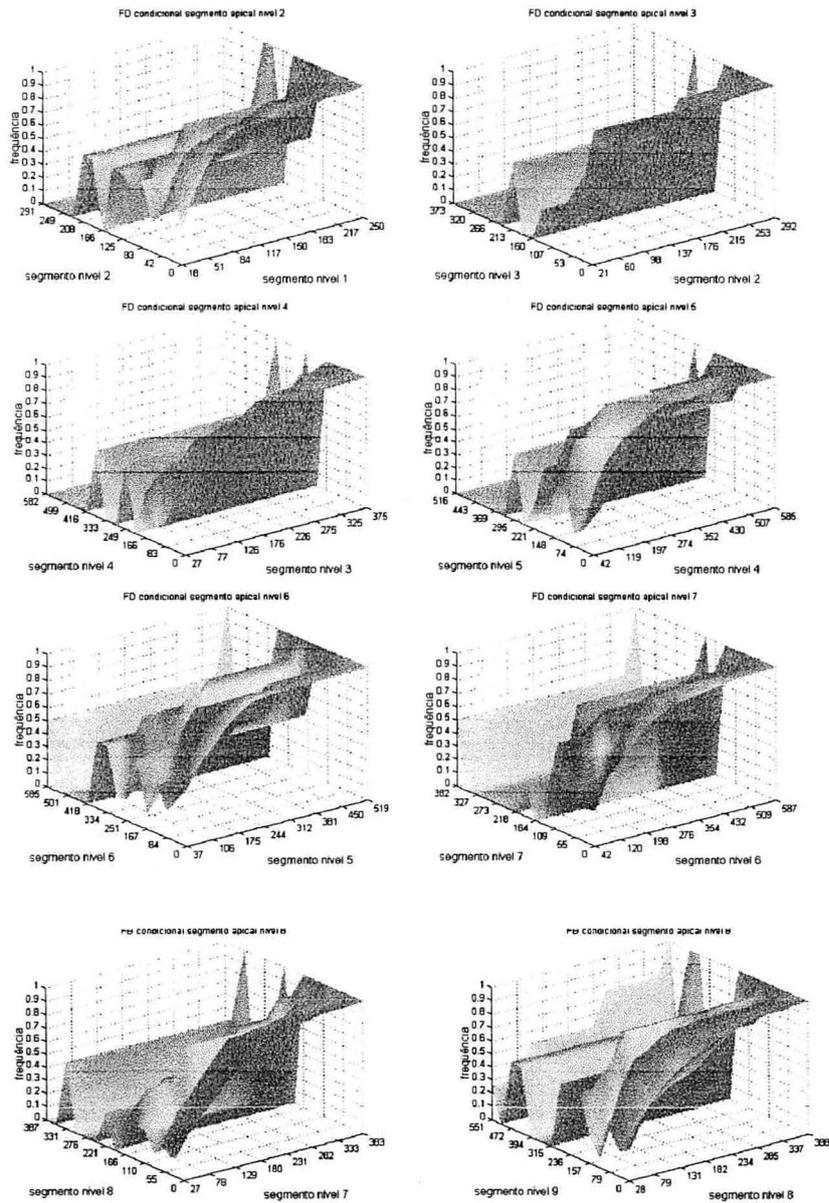
Funcao de distribucao para medida de curvatura basal 24 niveis

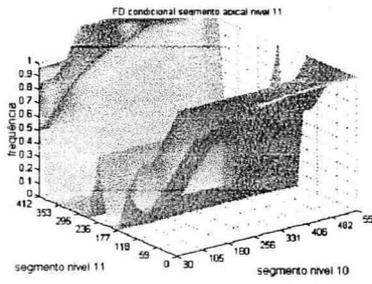
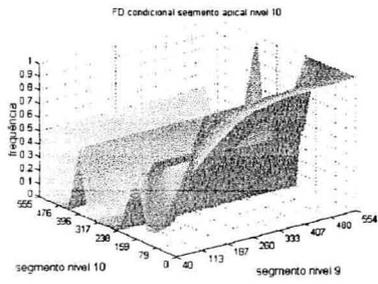


APÊNDICE C

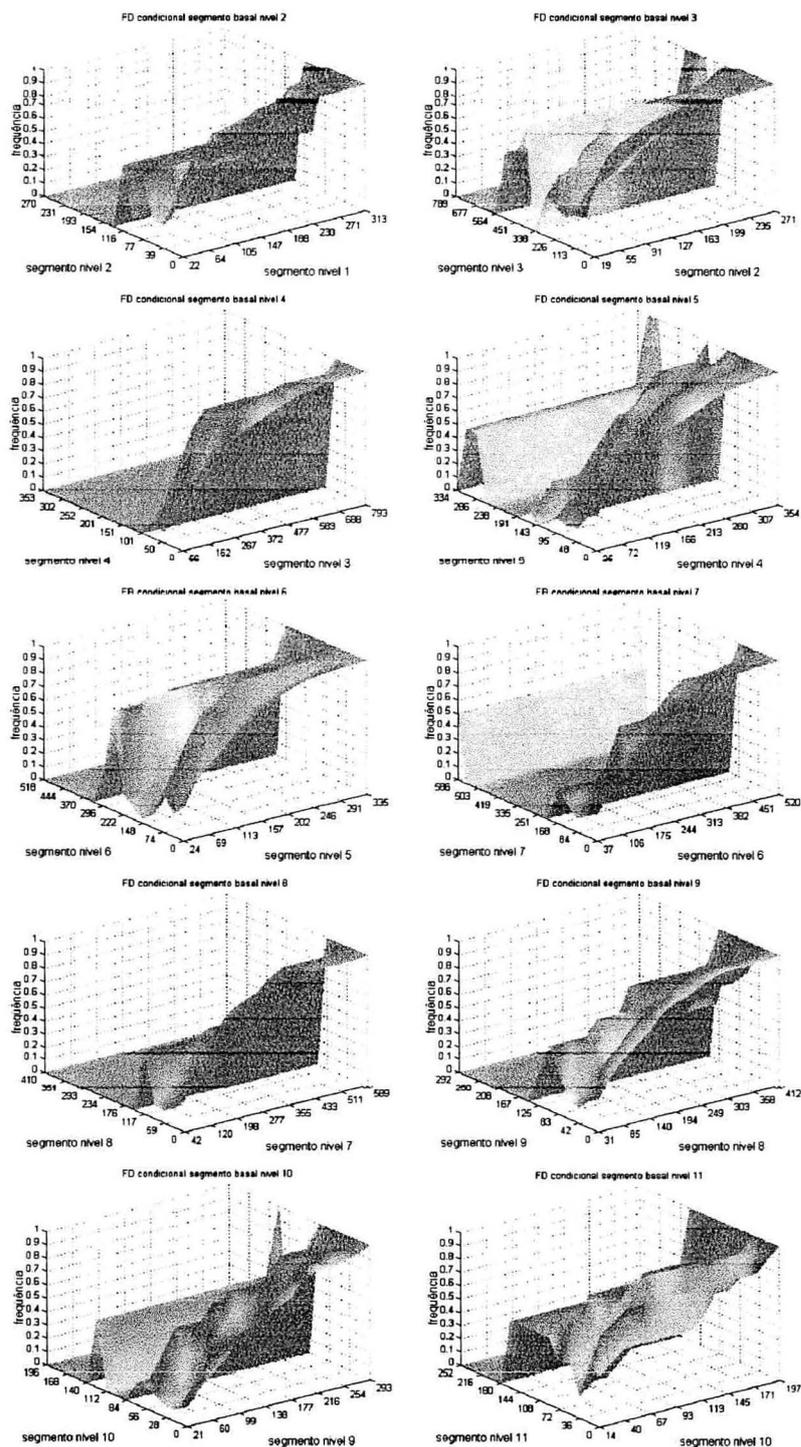
FDS DE DISTRIBUIÇÃO CONDICIONAIS DE CÉLULAS DO CÓRTEX VISUAL DE RATO

C.1 Segmento dendrítico apical

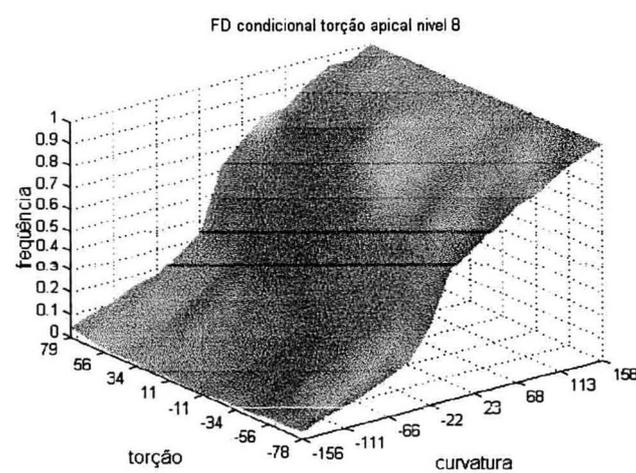
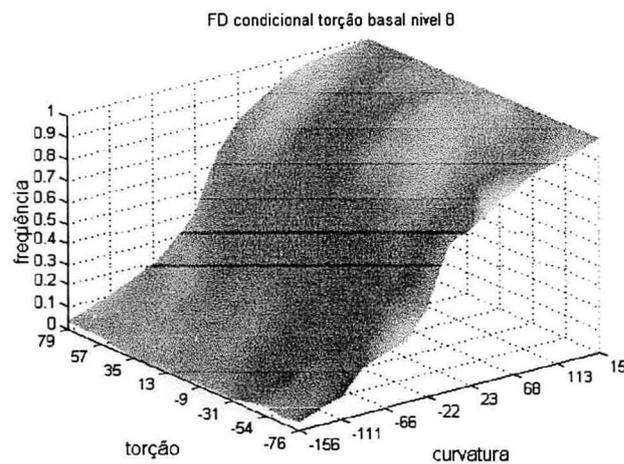
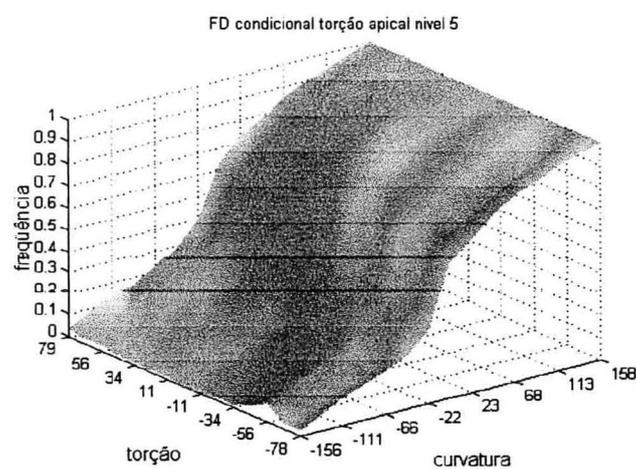
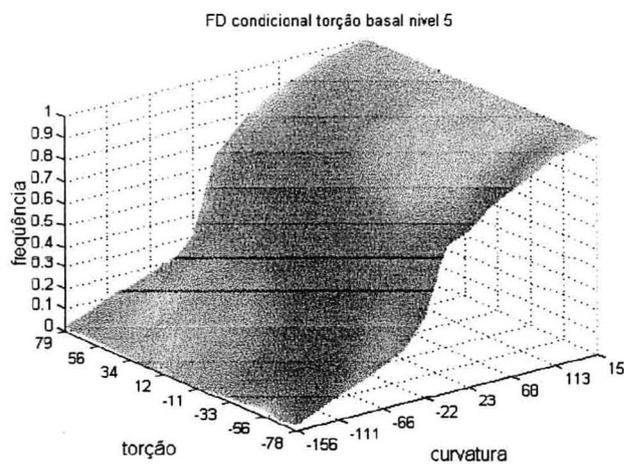
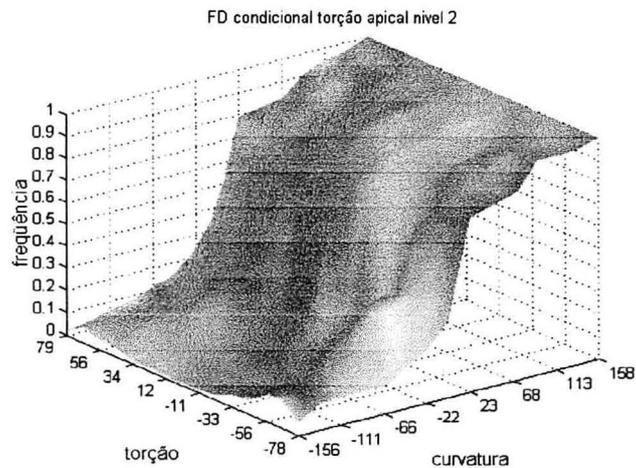
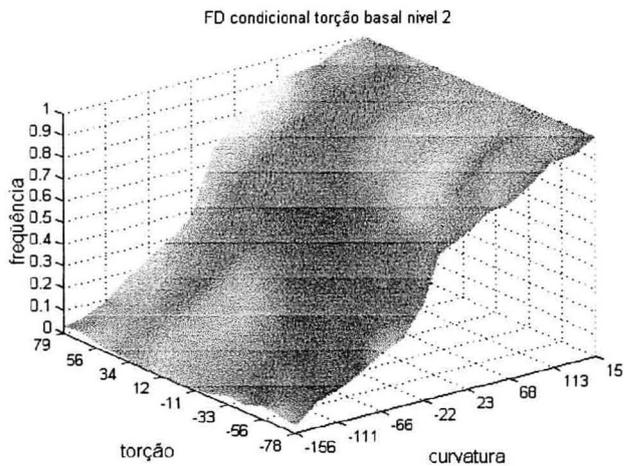




C.2 Segmento dendrítico basal

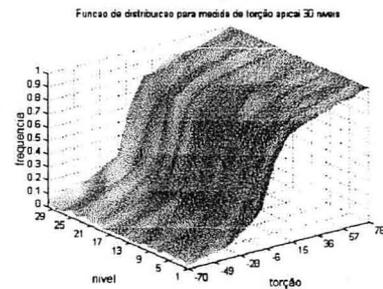
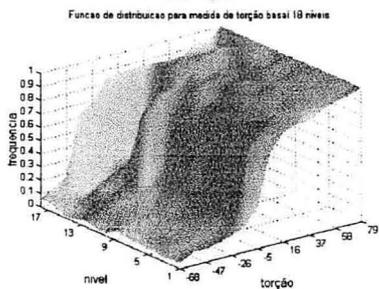
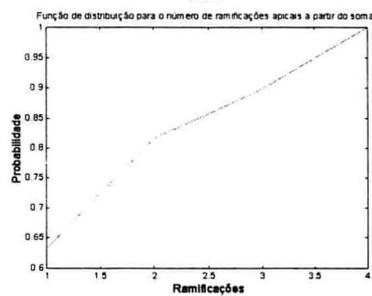
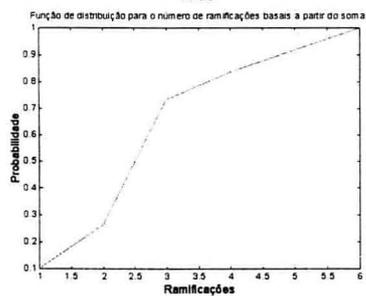
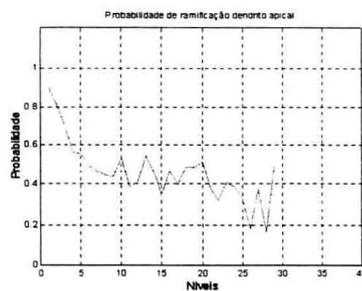
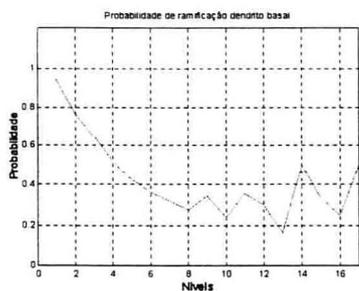
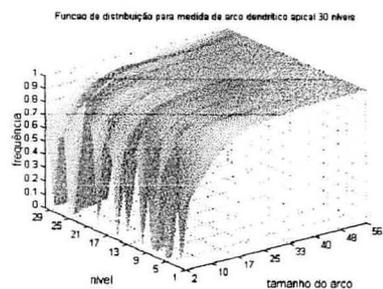
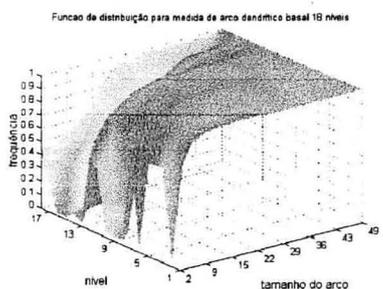
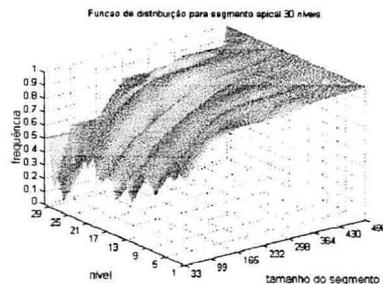
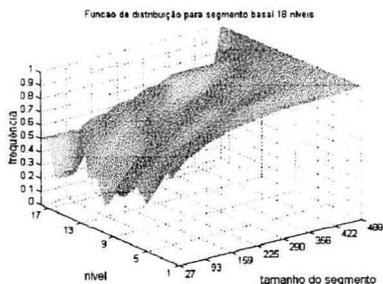


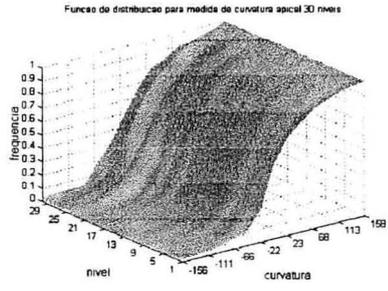
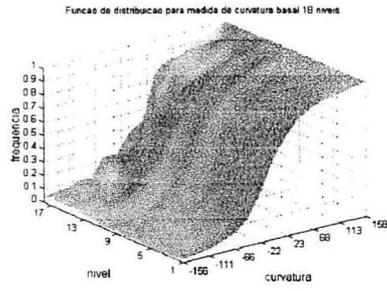
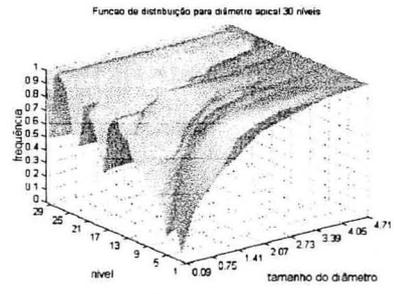
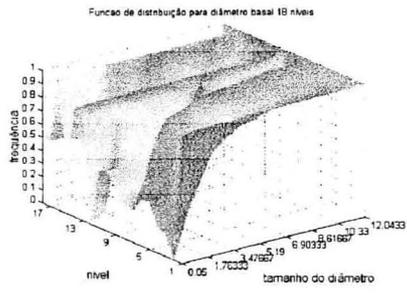
C.3 Ângulo de Torção Basal e Apical



APÊNDICE D

GRÁFICOS DE MEDIDAS DE CÉLULAS PIRAMIDAIS ARTIFICIAIS

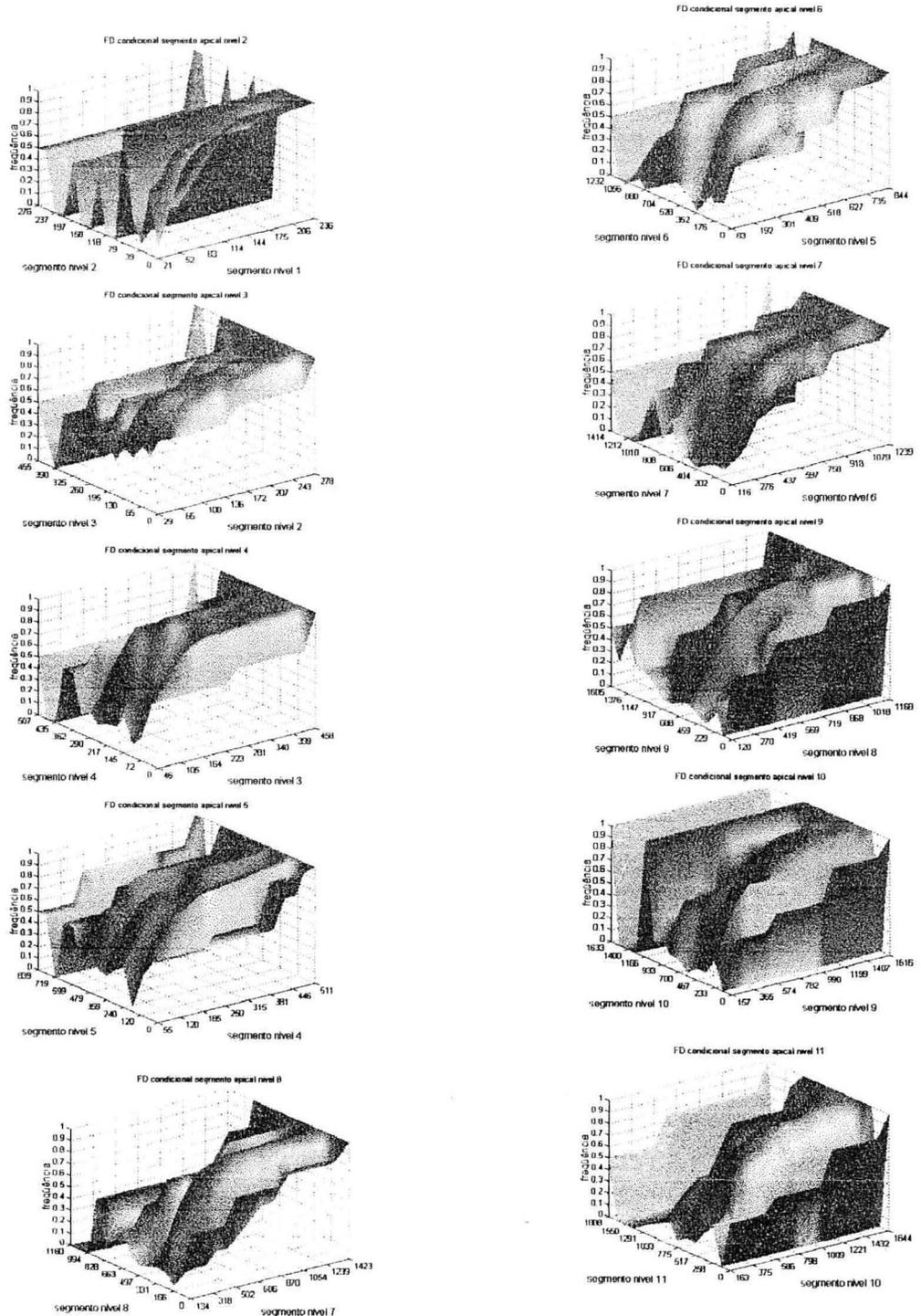




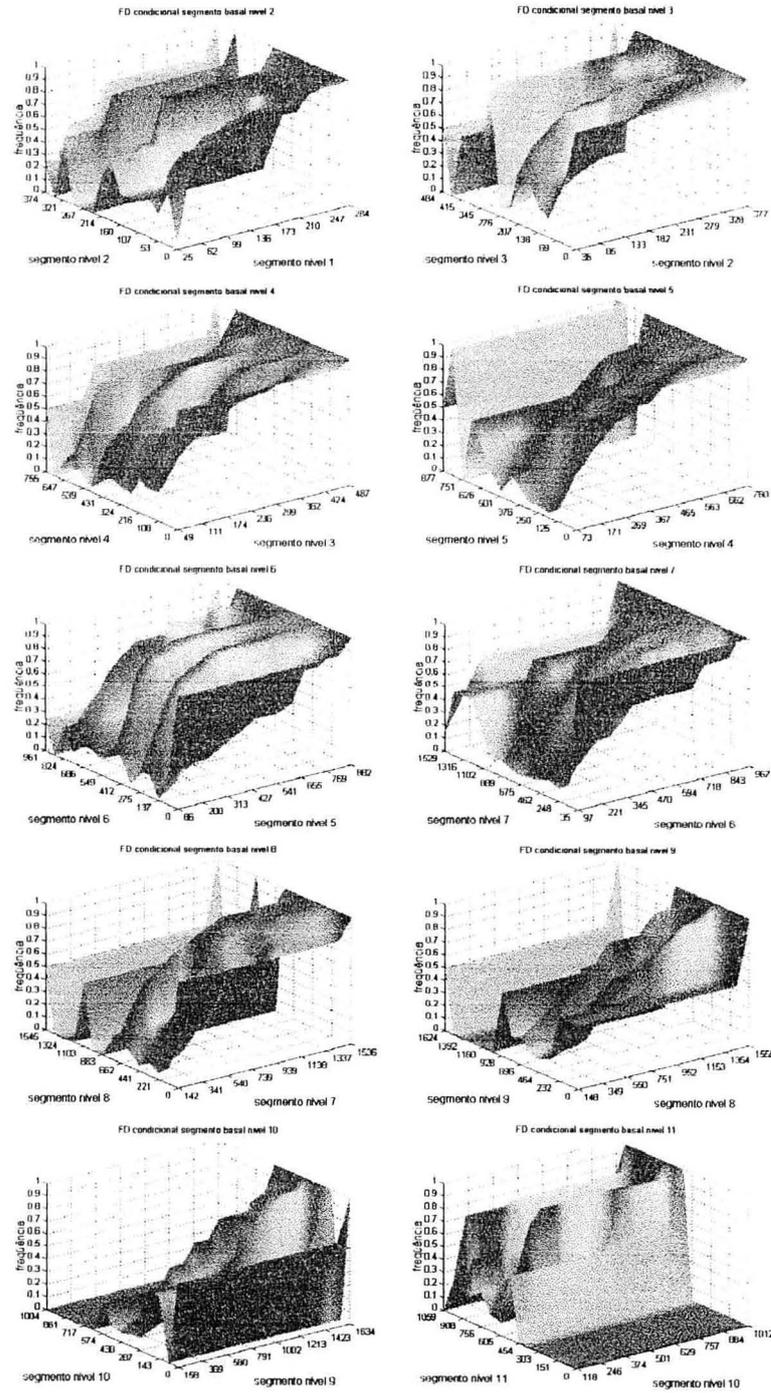
APÊNDICE E

FDS DE DISTRIBUIÇÃO CONDICIONAIS DE CÉLULAS ARTIFICIAIS

E.1 Segmento dendrítico apical



E.2 Segmento dendrítico basal



E.3 Ângulo de Torção Basal e Apical

