



**UNIVERSIDADE METODISTA DE PIRACICABA**  
**FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA**  
**MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

**AMBIENTE DE VISUALIZAÇÃO  
E INTERAÇÃO PARA SIMULAÇÕES NEURAIS**

GALESANDRO HENRIQUE CAPOVILLA

**ORIENTADORA: PROFA. DRA. REGINA CÉLIA COELHO**

PIRACICABA, SP  
2006





**UNIVERSIDADE METODISTA DE PIRACICABA**  
**FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA**  
**MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

**AMBIENTE DE VISUALIZAÇÃO  
E INTERAÇÃO PARA SIMULAÇÕES NEURAIS**

GALESANDRO HENRIQUE CAPOVILLA

**ORIENTADORA: PROFA. DRA. REGINA CÉLIA COELHO**

Dissertação apresentada ao Mestrado em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Ciência da Computação.

**PIRACICABA, SP**  
**2006**

*Ao*

*Meu primeiro filho Felipe*

*Que estará vindo ao mundo nos*

*Próximos meses.*

## **AGRADECIMENTOS**

Agradeço a Deus por ter me dado a oportunidade e a força necessária para que eu desenvolvesse este trabalho.

Agradeço a minha orientadora Prof. Dra. Regina Célia Coelho, pelo apoio, dedicação e confiança que depositou em minha pessoa.

Agradeço ao professor Nivaldi Calonego Junior, pelo apoio e dicas para o desenvolvimento do projeto.

Agradeço ao professor Luís Augusto Consularo, pelas dicas de leitura e pesquisa para elaboração do projeto.

Agradeço a todos os demais professores da UNIMEP, pelo empenho no reconhecimento do Mestrado em Ciência da Computação.

Agradeço ao Centro UNISAL - Campinas, pelo incentivo financeiro dado a este curso.

Agradeço ao Centro BORLAND, mais especificamente ao meu amigo Diego, que muito me auxiliou na conclusão deste trabalho.

Agradeço à minha esposa, pelos conceitos de Biologia que me ajudou a esclarecer.

Agradeço à minha querida Strega, pela sua paciência, dedicação e incentivo na conclusão deste curso.

Agradeço aos meus amigos do mestrado, que juntos trocamos muitas informações durante o curso.

“O que você faz é uma gota no oceano.  
Porém, é isso que dá sentido à sua vida.”

A. Schuweitzer

---

## RESUMO

O uso de artefatos computacionais como apoio a construção do conhecimento e à aprendizagem humana têm recebido, nos últimos tempos, um impulso extraordinariamente grande. Entre estes artefatos está a Realidade Virtual. A incorporação da Realidade Virtual na educação e na pesquisa é muito promissora. Este trabalho é parte de um projeto de pesquisa em desenvolvimento voltado à modelagem e simulação de estruturas neurais morfológicamente realistas. Mais especificamente, esta dissertação visa a construção de um ambiente de Realidade Virtual que permita interações do usuário com a simulação. Tal ambiente permite, inclusive, interações e navegação durante o crescimento, o que poderá resultar na modificação da estrutura obtida.

**PALAVRAS-CHAVE:** Realidade Virtual, Interações em Ambientes Virtuais, Interações com Estruturas Neurais.

---

---

## VISUALIZATION AND INTERACTION ENVIRONMENT OF NEURAL SIMULATION

### *ABSTRACT*

The use of computational artifacts as support to the building of knowledge and human learning has received, in recent times, an extraordinarily huge impulse. Amongst these artifacts is Virtual Reality. The incorporation of Virtual Reality in education and in research is very promising. This task is part of a research project being developed directed to the modeling and simulation of morphologically realistic neural structures. More specifically, the present dissertation aims at the construction of a Virtual Reality environment which allows for user interaction with the simulation. Such environment allows, including, interactions and navigation during growth, which may result in the modification of the obtained structure.

**KEYWORDS:** Virtual Reality, Interaction in Virtual Environment, Interaction with Neural Structures

---

## SUMÁRIO

Lista de Figuras .....	IX
RESUMO.....	IV
ABSTRACT.....	V
FIGURA 2.1 - ESTRUTURA DO NEURÔNIO (NEURÔNIO, 2006). 3.....	IX
FIGURA 2.2 - FORMAÇÃO DO AXÔNIO (NEURÔNIO, 2006). 4.....	IX
FIGURA 3.3 - LAYOUT VIBE – MICROSCÓPIO VIRTUAL (SUBRAMANIAN, 2001) 14.....	IX
FIGURA 3.4 - VOLEDIT – CRIAÇÃO DO ESQUELETO HUMANO (SINGH, 2003) 14.....	IX
FIGURA 3.5 - VENLAB - VIRTUAL ENVIRONMENT NAVIGATION LABORATORY (TARR E WARREN, 2002) 15.....	IX
FIGURA 3.6 - CONSTRUÇÃO DE UMA CENA UTILIZANDO VIRTUS WALKTHROUGH PRO (MENOUSEK E WOLFE, 1997) 16.....	IX
FIGURA 3.7 - EXEMPLO VIRTUS WALKTHROUGH (MENOUSEK E WOLFE, 1997) 17....	IX
FIGURA 4.8 - SUBDIVISÕES DA COMPUTAÇÃO GRÁFICA 20.....	IX
FIGURA 4.9 - MODELO DE ILUMINAÇÃO LOCAL. 23.....	IX
FIGURA 4.10 - INTERAÇÕES DE LUZ MODELADAS PELA RADIOSIDADE E POR MAPA DE FÓTONS 23... IX	
FIGURA 4.11 - PROPRIEDADES DE REFRAÇÃO DA LUZ EM BUMP MAP 25.....	IX
FIGURA 4.12 - EXEMPLO DE BUMP MAP 26.....	IX
FIGURA 4.13 - DEFINIÇÃO SPLINE 27.....	IX
FIGURA 4.14 - EXEMPLOS DE CURVAS DE BÉZIER 28.....	IX
FIGURA 4.15 - MÚLTIPLOS PONTOS DE CONTROLE 28.....	IX
FIGURA 5.16 - EDITOR GLSCENE 31.....	IX
FIGURA 5.17 - ESQUEMA FUNCIONAL 34.....	IX
FIGURA 5.18 - FUNÇÕES DO SIMULADOR 34.....	IX
FIGURA 5.19 - FUNÇÕES DO AMBIENTE DE REALIDADE VIRTUAL 35.....	IX
FIGURA 5.20 - FUNÇÕES DO USUÁRIO 35.....	IX
FIGURA 5.22 - ESTRUTURA DE RECEPÇÃO DOS DADOS 37.....	IX
FIGURA 5.23 - (A) - LEITURA DOS DADOS (B) - CENA GERADA COM OS DADOS LIDOS 38.....	IX
FIGURA 5.24 - LEITURA DOS DADOS (3 NEURÔNIOS) 39.....	IX
FIGURA 5.25 - CRIAÇÃO DINÂMICA DO OBJETO PIPE 40.....	IX
FIGURA 5.26 - OPÇÃO REDE 41.....	IX



<b>FIGURA 5.27 - CONFIGURAÇÃO SERVIDOR 42</b> .....	<b>IX</b>
<b>FIGURA 5.28 - MÓDULO ROBÔ 43</b> .....	<b>IX</b>
<b>FIGURA 5.29 - OPÇÃO INTERAÇÕES 43</b> .....	<b>IX</b>
<b>FIGURA 5.30 - CONFIGURAÇÃO DO CAMPO TRÓFICO 45</b> .....	<b>IX</b>
<b>FIGURA 5.31 - INSERÇÃO E CÁLCULO DO CAMPO TRÓFICO 46</b> .....	<b>IX</b>
<b>FIGURA 5.33 - CAMPO DE ATRAÇÃO COM VALORES DE K DISTINTOS 48</b> .....	<b>IX</b>
<b>48 X</b>	
<b>FIGURA 5.34 - CAMPO DE REPULSÃO COM VALORES DE K DISTINTOS 49</b> .....	<b>X</b>
<b>FIGURA 5.35 - AÇÃO DE MAIS DE UM CAMPO DE ATRAÇÃO 50</b> .....	<b>X</b>
<b>FIGURA 5.37 - OPÇÃO SALVA ESTADO – (A) A CENA INICIAL É GERADA E ARMAZENADA COMO MEMÓRIA ATUAL; ( B ) A MESMA CENA É ALTERADA E ARMAZENADA EM OUTRA MEMÓRIA E EM OUTRA DATA/HORA; (C) A ÚLTIMA CENA É MODIFICADA E SALVA NA MEMÓRIA 1 EM OUTRA DATA/HORA. 51</b> .....	<b>X</b>
<b>FIGURA 5.38 - OPÇÃO FERRAMENTAS 51</b> .....	<b>X</b>
<b>FIGURA 5.39 - OPÇÃO SPLINE; (A) CENA MONTADA SEM A UTILIZAÇÃO DE SPLINE; (B) CENA MONTADA COM A UTILIZAÇÃO DO SPLINE 52</b> .....	<b>X</b>
<b>FIGURA 5.40 - CAMINHO PERCORRIDO POR UM IMPULSO 53</b> .....	<b>X</b>
<b>FIGURA 5.41 - PONTOS DE LUZ 53</b> .....	<b>X</b>
<b>FIGURA 5.44 - CONFIGURAÇÕES 55</b> .....	<b>X</b>
<b>FIGURA 5.45 - SPLASH SCREEN 56</b> .....	<b>X</b>
<b>FIGURA 5.46 - MORTE CELULAR 56</b> .....	<b>X</b>
<b>FIGURA 5.50 - OPÇÃO SOBRE 59</b> .....	<b>X</b>
<b>FIGURA 5.51 - LISTA DE SELEÇÃO 59</b> .....	<b>X</b>
<b>1.INTRODUÇÃO</b> .....	<b>1</b>
<b>2.CONCEITOS SOBRE CÉLULAS E ESTRUTURAS NEURAIS</b> .....	<b>2</b>
3.FORMAÇÃO DOS NEURÔNIOS.....	3
4.MORTE NEURAL.....	4
5.Apoptose.....	4
6.Necrose.....	6
7.NEUROGÊNESE.....	6
8.SINAPSES .....	7
9.NEUROPLASTICIDADE.....	7
10.FATORES NEUROTRÓFICOS.....	8
<b>11.MODELOS E FERRAMENTAS RELACIONADAS À AMBIENTES DE RV EM NEUROCIÊNCIA</b> .....	<b>10</b>
12.NEUROCOMPUTAÇÃO.....	10
13.SOCKETS.....	11
14.TRABALHOS ANTERIORES ENVOLVENDO REALIDADE VIRTUAL E NEUROCIÊNCIA.....	13
<b>15. CONCEITOS DE COMPUTAÇÃO GRÁFICA UTILIZADOS NO PROJETO</b> .....	<b>19</b>
16.ILUMINAÇÃO.....	20
17.TRANSPARÊNCIA.....	23

18. TEXTURA.....	24
19. TRANSFORMAÇÕES GEOMÉTRICAS.....	26
20. SPLINES.....	27
21. Curvas Bézier.....	27
22. Curvas B-Splines.....	28
<b>23. ESPECIFICAÇÃO DO PROJETO E RESULTADOS OBTIDOS.....</b>	<b>30</b>
24. DESCRIÇÃO DAS FERRAMENTAS.....	30
25. VISÃO GERAL DO AMBIENTE.....	32
26. ARQUITETURA DO PROJETO.....	33
27. INICIALIZAÇÃO DO AMBIENTE.....	35
28. RECEPÇÃO DOS DADOS.....	37
29. VISUALIZAÇÃO .....	39
30. OPÇÕES DO USUÁRIO .....	40
31. Arquivo.....	40
32. Rede.....	41
33. Interações.....	43
34. Ferramentas.....	51
35. Configurações.....	55
36. Morte Celular.....	56
37. Ajuda.....	59
<b>38. CONCLUSÃO E TRABALHOS FUTUROS.....</b>	<b>61</b>
39. PESQUISAS FUTURAS.....	62
<b>40. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>64</b>

## LISTA DE FIGURAS

FIGURA 2.1 - ESTRUTURA DO NEURÔNIO (NEURÔNIO, 2006).....	3
FIGURA 2.2 - FORMAÇÃO DO AXÔNIO (NEURÔNIO, 2006).....	4
FIGURA 3.3 - LAYOUT VIBE – MICROSCÓPIO VIRTUAL (SUBRAMANIAN, 2001).....	14
FIGURA 3.4 - VOLEDIT – CRIAÇÃO DO ESQUELETO HUMANO (SINGH, 2003).....	14
FIGURA 3.5 - VENLAB - VIRTUAL ENVIRONMENT NAVIGATION LABORATORY (TARR E WARREN, 2002).....	15
FIGURA 3.6 - CONSTRUÇÃO DE UMA CENA UTILIZANDO VIRTUS WALKTHROUGH PRO (MENOUSEK E WOLFE, 1997).....	16
FIGURA 3.7 - EXEMPLO VIRTUS WALKTHROUGH (MENOUSEK E WOLFE, 1997).....	17
FIGURA 4.8 - SUBDIVISÕES DA COMPUTAÇÃO GRÁFICA.....	20
FIGURA 4.9 - MODELO DE ILUMINAÇÃO LOCAL.....	23
FIGURA 4.10 - INTERAÇÕES DE LUZ MODELADAS PELA RADIOSIDADE E POR MAPA DE FÓTONS....	23
FIGURA 4.11 - PROPRIEDADES DE REFRAÇÃO DA LUZ EM BUMP MAP.....	25
FIGURA 4.12 - EXEMPLO DE BUMP MAP.....	26
FIGURA 4.13 - DEFINIÇÃO SPLINE.....	27
FIGURA 4.14 - EXEMPLOS DE CURVAS DE BÉZIER.....	28
FIGURA 4.15 - MÚLTIPLOS PONTOS DE CONTROLE.....	28
FIGURA 5.16 - EDITOR GLSCENE.....	31
FIGURA 5.17 - ESQUEMA FUNCIONAL.....	34
FIGURA 5.18 - FUNÇÕES DO SIMULADOR.....	34
FIGURA 5.19 - FUNÇÕES DO AMBIENTE DE REALIDADE VIRTUAL.....	35
FIGURA 5.20 - FUNÇÕES DO USUÁRIO.....	35
FIGURA 5.22 - ESTRUTURA DE RECEPÇÃO DOS DADOS.....	37
FIGURA 5.23 - (A) - LEITURA DOS DADOS (B) - CENA GERADA COM OS DADOS LIDOS.....	38
FIGURA 5.24 - LEITURA DOS DADOS (3 NEURÔNIOS).....	39
FIGURA 5.25 - CRIAÇÃO DINÂMICA DO OBJETO PIPE.....	40
FIGURA 5.26 - OPÇÃO REDE.....	41
FIGURA 5.27 - CONFIGURAÇÃO SERVIDOR.....	42
FIGURA 5.28 - MÓDULO ROBÔ.....	43
FIGURA 5.29 - OPÇÃO INTERAÇÕES.....	43
FIGURA 5.30 - CONFIGURAÇÃO DO CAMPO TRÓFICO.....	45
FIGURA 5.31 - INSERÇÃO E CÁLCULO DO CAMPO TRÓFICO.....	46
FIGURA 5.33 - CAMPO DE ATRAÇÃO COM VALORES DE K DISTINTOS.....	48

<b>FIGURA 5.34 - CAMPO DE REPULSÃO COM VALORES DE <math>K</math> DISTINTOS.....</b>	<b>49</b>
<b>FIGURA 5.35 - AÇÃO DE MAIS DE UM CAMPO DE ATRAÇÃO.....</b>	<b>50</b>
<b>FIGURA 5.37 - OPÇÃO SALVA ESTADO – (A) A CENA INICIAL É GERADA E ARMAZENADA COMO MEMÓRIA ATUAL; ( B ) A MESMA CENA É ALTERADA E ARMAZENADA EM OUTRA MEMÓRIA E EM OUTRA DATA/HORA; (C) A ÚLTIMA CENA É MODIFICADA E SALVA NA MEMÓRIA 1 EM OUTRA DATA/HORA.....</b>	<b>51</b>
<b>FIGURA 5.38 - OPÇÃO FERRAMENTAS.....</b>	<b>51</b>
<b>FIGURA 5.39 - OPÇÃO SPLINE; (A) CENA MONTADA SEM A UTILIZAÇÃO DE SPLINE; (B) CENA MONTADA COM A UTILIZAÇÃO DO SPLINE.....</b>	<b>52</b>
<b>FIGURA 5.40 - CAMINHO PERCORRIDO POR UM IMPULSO.....</b>	<b>53</b>
<b>FIGURA 5.41 - PONTOS DE LUZ.....</b>	<b>53</b>
<b>FIGURA 5.44 - CONFIGURAÇÕES.....</b>	<b>55</b>
<b>FIGURA 5.45 - SPLASH SCREEN.....</b>	<b>56</b>
<b>FIGURA 5.46 - MORTE CELULAR.....</b>	<b>56</b>
<b>FIGURA 5.50 - OPÇÃO SOBRE.....</b>	<b>59</b>
<b>FIGURA 5.51 - LISTA DE SELEÇÃO.....</b>	<b>59</b>

## 1. INTRODUÇÃO

A realidade virtual vem crescendo a cada dia e atuando cada vez mais em diversas áreas relacionadas a ciência; como é o caso das simulações na área de neurociência. Hoje, as simulações representam um avanço significativo em pesquisas relacionadas a esta área, possibilitando estudos mais avançados e eficazes em relação ao cérebro. Uma das formas mais interativas e interessantes de visualizar estas simulações encontra-se na Realidade Virtual, que permite, além da visualização tridimensional, a interação, imersão e navegação do usuário no sistema. O advento das tecnologias nesta área tem colaborado para a abertura de uma nova dimensão para diversos pesquisadores, que podem “entrar” em simulações eletrônicas, o que permitem a eles ocupar um espaço imaginário. Pelo fato da Realidade Virtual (RV) estar além da visualização computacional, ela nos permite participar de fato em um ambiente visual simulado (“realidade simulada”) e analisar o funcionamento do “mundo” sendo percorrido. Recentemente, este fato tem se tornado um grande aliado para os neurocientistas, uma vez que a RV pode auxiliá-los na interpretação da grande quantidade de dados geradas pelas simulações. O surgimento de novas tecnologias e a melhoria na qualidade de tecnologias já existentes (como potência computacional e qualidade dos monitores de vídeo) tem tornado esta área muito atrativa tanto para pesquisas quanto para aplicações em neurociência. Várias aplicações da RV e da computação gráfica em neurociência têm surgido em pesquisa, treinamento e tele-imersão (ASCOLI, 1998; BÜLTHOFF, 1999; LEIGH, 1994; TARR, 2002). Em termos de impacto realista nas simulações, estas aplicações podem ser tão boas ou até melhor que a realidade física, pelo fato de poder se executar um determinado experimento diversas vezes, analisando, por exemplo, diferentes ângulos de visualização. Esta tecnologia permite criar universos reais e, ainda mais, criar novas possibilidades ao mundo criado. Além disso, o desenvolvimento de técnicas de realidade virtual e o progresso da computação gráfica permitem a implementação de programas extremamente complexos e rápidos para criar, mostrar e manipular modelos de estruturas neurais.

O trabalho enfoca o desenvolvimento de um ambiente virtual de ensino e pesquisa, relacionado à simulação do crescimento de neurônios e estruturas neurais, podendo, o usuário do sistema, interagir neste crescimento, ou seja, participar do mundo virtual. Uma grande vantagem deste projeto é a

possibilidade de se observar diversas vezes o crescimento dos neurônios de maneira interativa.

No decorrer desta dissertação encontram-se algumas aplicações e simulações já realizadas de realidade virtual relacionadas às áreas biológicas e, com base nisso, o desenvolvimento do projeto utilizando a realidade virtual, enfatizando a neurociência, um dos ramos que está sendo muito estudado em relação a simulações utilizando realidade virtual.

O segundo capítulo trata de alguns conceitos fundamentais para o melhor entendimento do projeto, como algumas definições sobre neurônios, a definição sobre a formação das estruturas neurais, apoptose, neurogênese, as sinapses, plasticidade dos neurônios e por fim o conceito fundamental de campos tróficos.

O terceiro capítulo trata do que vem a ser a neurocomputação, bem como a linguagem de programação adotada Delphi, juntamente com a biblioteca gráfica GLScene, adotada por ser uma biblioteca com ferramentas 3D já pré-definidas. Ainda no terceiro capítulo encontram-se alguns projetos de simuladores desenvolvidos, relacionando a realidade virtual com as diversas outras áreas biológicas sempre enfatizando a pesquisa e o aprendizado.

O quarto capítulo trata de algumas definições relacionadas a Computação Gráfica utilizadas no projeto, tais como, as transformações geométricas, o *spline* e a iluminação.

O quinto capítulo trata da especificação do projeto, bem como a inicialização do ambiente criado, a maneira com que os dados são recebidos, a visualização destes dados na cena e todas as possíveis opções do usuário.

Por fim, o trabalho coloca em destaque um ambiente de Realidade Virtual, interativo, onde o usuário, além de visualizar o ambiente todo, possa modificar o mundo apresentado de acordo com as suas necessidades ou mesmo para sua aprendizagem.

## **2. CONCEITOS SOBRE CÉLULAS E ESTRUTURAS NEURAIS**

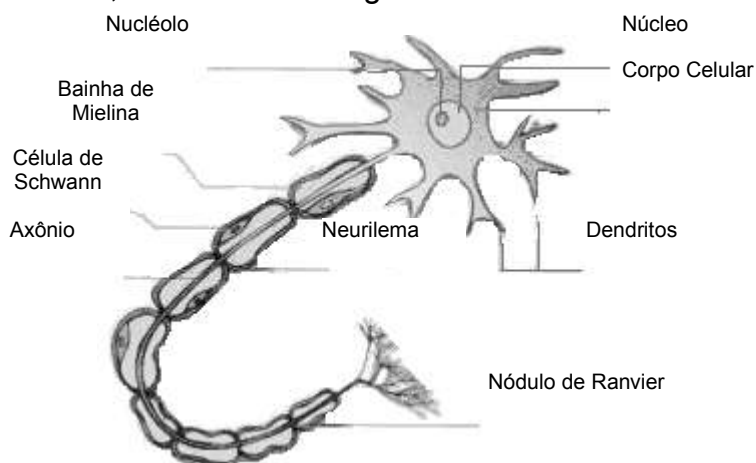
No estudo do cérebro humano, o pouco conhecimento sobre o cérebro é o maior problema a ser enfrentado. Temos muitas vezes indícios ou pesquisas que demonstram como o cérebro funciona, porém são conceitos ainda muito complicados de origem desconhecida. Sabe-se que o cérebro humano é

composto de aproximadamente dez bilhões de neurônios e cada neurônio é conectado a dez mil outros neurônios. Para um entendimento mais claro do desenvolvimento deste trabalho, é relativamente importante o detalhamento de alguns itens relacionados aos neurônios, tais como sua formação, sua morte e sua regeneração, itens estes que são tratados no decorrer deste capítulo.

### 3. FORMAÇÃO DOS NEURÔNIOS

A unidade estrutural e funcional do sistema nervoso é o neurônio. É uma célula muito especializada que possui duas propriedades fundamentais para as funções que exercem: a excitabilidade (capacidade de reagir aos estímulos) e a condutibilidade (uma vez alterados pelos estímulos, os neurônios transmitem essa alteração por toda sua extensão) (DOWNLING, 1992).

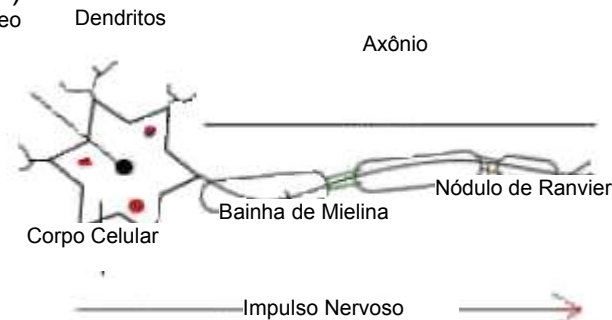
O neurônio é composto de um corpo celular ou soma, onde está o núcleo, e de finos prolongamentos celulares, chamados neuritos, que podem ser dendritos ou podem ser axônios, como ilustra a Figura 2.1.



**Figura 2.1 - Estrutura do Neurônio (NEURÔNIO, 2006).**

Os dendritos são prolongamentos geralmente muito ramificados e que atuam como receptores de estímulos. Os axônios atuam como condutores dos impulsos nervosos e, em geral, só possuem ramificações na extremidade. Em toda sua extensão, o axônio é envolvido por um tipo celular denominado *célula de Schwann*. Em muitos axônios, as células de *Schwann* determinam a formação da *bainha de mielina* - invólucro lipídico que atua como isolante elétrico e facilita a transmissão do impulso nervoso. Somente os primatas possuem bainha de mielina. Entre uma célula de *Schwann* e outra existe uma região de descontinuidade da bainha de mielina chamada *nódulo de Ranvier*. A parte celular da bainha de mielina, onde estão o citoplasma e o núcleo da

célula de *Schwann*, constitui o neurilema como ilustra a Figura 2.2 (NEURÔNIO, 2006).



**Figura 2.2 - Formação do Axônio (NEURÔNIO, 2006).**

O percurso do impulso nervoso no neurônio é sempre no sentido dendrito → corpo celular → axônio (NEURÔNIO, 2006).

Segundo o número de dendritos, os neurônios se dividem em:

- Neurônio mono ou unipolar: um só axônio, nenhum dendrito. Presente nos órgãos dos sentidos. Os receptores sensoriais ocupam o lugar dos dendritos.
- Neurônio bipolar: um axônio, um dendrito. Presente nos órgãos dos sentidos.
- Neurônio pseudomonopolar: Dendrito e axônio se fundem perto do corpo neuronal. Encontram-se na espinha dorsal.
- Neurônio multipolar: Vários dendritos, um axônio. Predomina no sistema nervoso central.

#### 4. MORTE NEURAL

Quando se pensa em estudar o desenvolvimento dos organismos, acredita-se apenas em fenômenos progressivos: divisão celular, crescimento. Porém, hoje, a morte neural também é um fato normal durante o desenvolvimento. A morte pode ocorrer por um processo mais violento e rápido (necrose), ou envolver o "suicídio" dos neurônios por um processo mais lento (apoptose). Nesses tópicos que procedem são detalhados os dois tipos de mortes neurais que também são abordadas pelo sistema desenvolvido.

#### 5. Apoptose

Por definição, apoptose ou morte celular programada é um tipo de "autodestruição celular" que requer energia e síntese protéica para a sua



execução. Esta morte celular está relacionada com a homeostase na regulação fisiológica do tamanho dos tecidos, exercendo um papel oposto ao da mitose, que seria a divisão e crescimento celular. O termo é derivado do grego "αποπτωσις", que referia-se à queda das folhas das árvores no outono - um exemplo de morte programada fisiológica e apropriada que também implica em renovação (APOPTOSE, 2006; LENT, 2001; MARLANGUE, 1996).

Fisiologicamente, esse suicídio celular ocorre no desenvolvimento embrionário, na renovação de células epiteliais e hematopoiéticas, na involução cíclica dos órgãos reprodutivos da mulher, na atrofia induzida pela remoção de fatores de crescimento ou hormônios, na involução de alguns órgãos e ainda na regressão de tumores. Portanto, consiste em um tipo de morte programada, desejável e necessária que participa na formação dos órgãos e que persiste em alguns sistemas adultos como a pele e o sistema imunológico (APOPTOSE, 2006; LENT, 2001; MARLANGUE, 1996).

A indução de apoptose, seja através de mecanismos imunológicos, seja por outros mecanismos homeostáticos específicos, é extremamente importante no processo de eliminação de células sofrendo transformação maligna. Danos não reparáveis no DNA (por mutações ou infecções virais) aparentemente iniciam o processo de apoptose. É importante salientar que muitos dos genes que condicionam a proliferação celular (chamados oncogenes e genes supressores de tumores) estão também envolvidos na iniciação do processo de apoptose e que a inibição por si só do processo fisiológico da apoptose leva à sobrevivência prolongada das células, favorecendo o acúmulo de mutações e a transformação maligna. Assim, a apoptose representa um mecanismo de eliminação seletiva de células cuja sobrevivência poderia prejudicar o bem estar do organismo (APOPTOSE, 2006; LENT, 2001; MARLANGUE, 1996).

## 6. NECROSE

Deve-se tomar cuidado para não confundir apoptose com necrose; a apoptose é uma morte celular natural ou programada, enquanto que a necrose é a morte celular acidental. Na necrose os fenômenos degenerativos são irreversíveis, causados por uma agressão intensa. É interessante salientar que o mesmo agente pode provocar tanto necrose como apoptose; sendo que a severidade da agressão parece ser o fator determinante do tipo de morte celular. Vários agentes já foram confirmados como indutores de apoptose, entre eles diversas viroses, hipertermia e várias toxinas (LENT, 2001; NECROSE, 2006).

A morte é patológica ou "acidental" quando a célula é impedida de manter seus processos vitais por lesões físicas ou químicas causadas por fatores externos, como temperaturas extremas, radiação, traumas, produtos tóxicos e falta de oxigênio (como no infarto do miocárdio e na gangrena). As lesões podem ter ainda origem biológica, como nas infecções por bactérias ou vírus. Esse tipo de morte celular, o único conhecido pelos cientistas mais antigos, é chamado de necrose.

## 7. NEUROGÊNESE

A formação de novos neurônios no cérebro humano adulto é conhecida como neurogênese, e ocorre principalmente na região do cérebro ligada à memória e a aprendizagem, esta região também é conhecida como hipocampo (GOULD, 1999a; GOULD, 1999b; GOULD, 1998, p.3168-3171).

Até 1998 acreditava-se que seria impossível ao cérebro adulto de um ser humano criar novos neurônios, sendo, portanto, o processo de envelhecimento celular que caracteriza a idade madura sinônimo de perda de inteligência ou, por outras palavras, a diminuição da capacidade de discernir e pensar. A idade adulta traria a incapacidade cerebral de produzir bem, já que não se podia substituir os neurônios que iam morrendo com o avançar da idade. Por volta de 1970, a ciência descobriu que em ratos adultos era possível a criação de novas células cerebrais. Assim, a neurogênese foi descoberta e, a partir de novas pesquisas em 1988, verificou-se o mesmo em macacos adultos. Por fim, em 1998, percebeu-se que também nos seres humanos a neurogênese era possível. No início de 1999, cientistas da Universidade de Princeton, nos Estados Unidos, começaram a fornecer os primeiros indícios que comprovam a capacidade do cérebro humano de um adulto em criar novos neurônios pelo

esforço cerebral e também por meio de exercícios físicos. Assim, o esforço cerebral só se dá quando questionamos, estudamos, pesquisamos, analisamos criteriosamente temas do nosso interesse, porque apenas estes nos permitem a devida postura no campo da concentração. Quando não nos esforçamos com uma reflexão produtiva e persistente, não há atividade cerebral suficiente para que ocorra o processo de neurogênese (GOULD, 1999a; GOULD, 1999b; GOULD, 1998, p.3168-3171).

## **8. SINAPSES**

Sinapse é a região de passagem do impulso nervoso de um neurônio para a célula adjacente. Estes pontos de contato ocorrem onde o ramo terminal de um axônio se põe em contato com os dendritos ou com o corpo do segundo neurônio (SINAPSES, 2006; LENT, 2001).

Numa sinapse distinguem-se partes funcionais morfológicamente distintas, tais como, porção pré-sináptica do neurônio transmissor do impulso e porção pós-sináptica do neurônio receptor (SINAPSES, 2006; LENT, 2001).

Na sinapse, região de contato entre dois neurônios, existe uma pequena distância entre as duas células envolvidas, isto é, não há perfeita continuidade entre a membrana de ambas. A passagem do impulso nervoso nessa região é feita por substâncias químicas: os neuro-hormônios, também chamados mediadores químicos ou neurotransmissores, que são armazenados em vesículas sinápticas dentro do terminal axonal e liberados na fenda sináptica (SINAPSES, 2006; LENT, 2001).

## **9. NEUROPLASTICIDADE**

A plasticidade neural é a capacidade de organização (no sentido de arranjo topológico) ou reorganização, do sistema nervoso frente ao aprendizado e a lesão. Esta organização se relaciona com a modificação de algumas conexões sinápticas (NEURÔNIO, 2006; LENT, 2001).

A plasticidade neural não ocorre apenas em processos patológicos, mas assume funções no desenvolvimento normal do organismo. Podemos citar como exemplo, as organizações das conexões nervosas que ocorrem durante o período embriológico do indivíduo, onde muitas dessas conexões não podem ser determinadas apenas por um programa genético já que mais tarde, no adulto, suas funções neurais de adaptação dependerão de condições do

ambiente. Assim, uma das primeiras formas de plasticidade nervosa é denominada de amadurecimento estímulo-dependente do sistema nervoso central. Uma segunda forma de plasticidade em um organismo normal é o processo de aprendizagem, na qual são considerados: o aprendizado motor inconsciente (automatismo) e o consciente (memória). Os dois processos baseiam-se em mecanismos fisiológicos semelhantes e constituem a base para uma organização do sistema nervoso, e para a reorganização após processos lesionais, que são direcionados por atividades neurais e por conseguinte, são influenciadas por estimulação periférica, uma vez que todas as percepções do nosso corpo e do meio que nos rodeia são captadas e conduzidas ao neuro-eixo pelos sistemas dos sentidos (NEURÔNIO, 2006; LENT, 2001).

#### **10. FATORES NEUROTRÓFICOS**

Fatores Neurotróficos são proteínas que promovem o desenvolvimento, a sobrevivência e a manutenção de populações neurais específicas. Os contatos entre axônios e seus órgãos-alvo são de crucial importância para a proliferação, migração, diferenciação, formação, estabelecimento e manutenção das conexões neurais (ANNUNCIATO, 1994; ANNUNCIATO, 1995).

Ainda que se saiba que os programas genéticos podem direcionar os processos de desenvolvimento do sistema neural, deve ser lembrado que as interações com o meio, mesmo no período embrionário, resultam em mudanças nas estruturas nervosas centrais e que a falta de certas interações podem alterar padrões normais de desenvolvimento. Sabemos que determinadas substâncias químicas produzidas pelos tecidos têm a capacidade de promover a divisão celular durante o desenvolvimento (fatores de crescimento), enquanto outras são essenciais para a sobrevivência de células nervosas (fatores neurotróficos = alimento), participando das interações entre os neurônios e seus alvos (outra célula nervosa, músculo estriado esquelético, cardíaco, músculo liso, tecido glandular e tecidos de revestimento) (ANNUNCIATO, 1994; ANNUNCIATO, 1995).

O fator de crescimento nervoso (NGF – "*Nerve Growth Factor*"), também conhecido como substância química responsável pelo crescimento axonal, mantém 3 papéis fundamentais:

- sobrevivência neural;
- extensão de prolongamentos;
- diferenciação de neurônios.

O mecanismo básico de ação do NGF é ser produzido por células-alvo, ligando-se a receptores dos terminais axonais sendo internalizado e transportado até o corpo celular, onde participa da ativação metabólica neural.

Essa ação é importante tanto na renovação das estruturas celulares como na sobrevivência celular, e em fenômenos plásticos. Evidentemente, a interrupção do transporte retrógrado desse fator bloqueia seu efeito, resultando na morte de neurônios dele dependentes (PURVES, 1988).

## **11. MODELOS E FERRAMENTAS RELACIONADAS À AMBIENTES DE RV EM NEUROCIÊNCIA**

A partir do século passado, a neurociência tenta desvendar os processos que ocorrem no cérebro. Houve muitos progressos ao longo do tempo, mas as dificuldades ainda são imensas, tanto no tratamento de problemas neurológicos quanto no desenvolvimento de sistemas de inteligência artificial mais poderosos. O uso de técnicas computacionais abriu perspectivas promissoras no estudo do sistema nervoso, e expressivos avanços devem decorrer da exploração adequada de seus recursos. A importância da computação aplicada à neurociência é tão grande que surge o termo neurociência computacional ou neurocomputação (BRAGA, 1998).

## **12. NEUROCOMPUTAÇÃO**

As primeiras informações mencionadas sobre a neurocomputação datam de 1943, em artigos de McCulloch e Pitts (1943), em que sugeriam a construção de uma máquina baseada ou inspirada no cérebro humano. Muitos outros artigos e livros surgiram desde então, porém, por um longo período, pouco resultado foi obtido. Em 1949 Donald Hebb escreveu um livro intitulado "*The Organization of Behavior*" (A Organização do Comportamento) que perseguia a idéia de que o condicionamento psicológico clássico está presente em qualquer parte dos animais pelo fato de que esta é uma propriedade de neurônios individuais. Suas idéias não eram completamente novas, mas Hebb foi o primeiro a propor uma lei de aprendizagem específica para as sinapses dos neurônios. Este primeiro e corajoso passo serviu de inspiração para que muitos outros pesquisadores perseguissem a mesma idéia. E, embora muito tenha sido estudado e publicado nos anos que seguiram (1940-1950), somente obtiveram maior importância nos anos posteriores.

Também proveniente deste período de tempo foi a construção do primeiro neurocomputador, denominado Snark, por Mavin Minsky, em 1951. O Snark operava com sucesso a partir de um ponto de partida técnico, ajustando seus pesos automaticamente. Entretanto, ele nunca executou qualquer função de processamento de informação interessante, mas serviu de inspiração para as idéias de estruturas que o sucederam.

O primeiro neurocomputador a obter sucesso (*Mark I Perceptron*) surgiu em 1957 e 1958, criado por Frank Rosenblatt, Charles Wighttman e outros. Devido à profundidade de seus estudos, a suas contribuições técnicas e a sua maneira moderna de pensar, muitos o vêem como o fundador da neurocomputação na forma em que a temos hoje. Seu interesse inicial para a criação do *Perceptron* era o reconhecimento de padrões (BRAGA, 1998).

Os modelos neurais, procuram aproximar o processamento dos computadores ao processamento no cérebro. As redes neurais possuem um grau de interconexão similar à estrutura do cérebro em um computador convencional moderno. A informação é transferida em tempos específicos dentro de um relacionamento com um sinal para sincronização (KOVACS, 1996; TAFNER, 1995).

O motivo pelo qual máquinas inspiradas na biologia são diferentes dos computadores pessoais se encontra no fato de que as máquinas atuais baseiam seu processamento explicitamente em modelos matemáticos. Mecanismos de controle baseado em mecanismos neurais entretanto, não são baseados em modelos determinísticos, utilizam modelos estocásticos, ou seja, cálculos matemáticos para efetuar suas operações, porém podem coordenar diversos graus de liberdade durante a execução de tarefas manipulativas e em ambientes desestruturados. Eles são capazes de lidar com tarefas complexas sem que tenham que desenvolver um modelo matemático determinístico e nem um modelo do ambiente em que operam (KOVACS, 1996; TAFNER, 1995).

### **13. SOCKETS**

Como o projeto inclui a interação utilizando diferentes máquinas e o ambiente de RV está funcionando em uma máquina diferente daquela responsável pela geração das células neurais, é necessário que os dados sejam enviados e capturados da rede, caracterizando, desta forma, um sistema de realidade virtual distribuído. Para permitir que as diferentes partes do sistema possam se comunicar corretamente, utilizam-se *sockets*.

Introduzido em 1981, na Universidade de Berkeley, *sockets* são estruturas que permitem que funções de software se interconectem. O conceito é o mesmo de um soquete (elétrico, telefônico), que serve para interconectar elementos diferentes. O termo é usado para especificar uma estrutura que faz com que

rotinas de software na mesma máquina ou em máquinas diferentes possam se comunicar (OLIVEIRA, 2002; SOCKETS, 2006).

Mais especificamente, o termo é usado para softwares usados na Internet, para especificar a conexão entre o software da camada de aplicação (protocolos HTTP, FTP, TELNET, POP3, etc...) e a camada de transporte (protocolos TCP ou UDP).

No caso de software pela Internet, um *socket* é constituído por um endereço IP (x.y.z.w) e um número de porta. O endereço IP identifica a máquina e a porta o processo rodando nesta máquina. Numa máquina pode haver vários processos em diversas portas, como por exemplo:

- Máquina com endereço IP – 127.115.12.17
- porta 21 – processo FTP
- porta 22 - Telnet
- porta 80 – servidor WEB

Suponha agora que a máquina precisa ser acessada ao mesmo tempo por vários clientes WEB, portanto na porta 80. O processo associado à porta 80 cria uma porta para cada novo cliente, permitindo que todos sejam atendidos ao mesmo tempo, executando as solicitações de cada um.

A seguir serão apresentadas algumas formas de comunicação utilizando *sockets*.

Na comunicação por *sockets* local, são utilizados programas que possuem funções específicas para este tipo de tarefa (trocar mensagens). Este tipo de comunicação é normalmente utilizado por processos comuns de usuário ou sistemas que são executados em um mesmo hardware. Permite ao programador definir como e em que situações será efetuada a comunicação entre os processos (troca de mensagens), entretanto, não possibilita o controle dos processos de "baixo nível" que garantem essa comunicação.

Semelhantemente à comunicação por *sockets* local, são utilizados programas que possuem funções específicas para troca de mensagens entre processos. Comunicação deste tipo é efetuada por processos que são executados em equipamentos distintos, ou em sistemas alocados fisicamente em locais

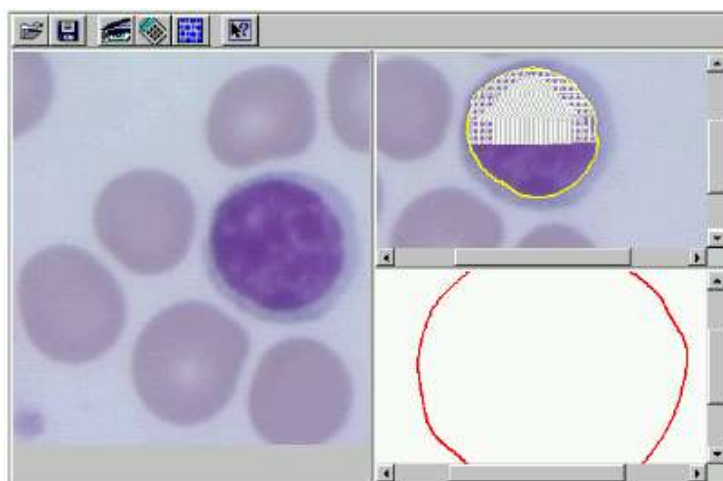


diferentes. Elas permitem ao programador definir as características da troca de mensagens propriamente dita.

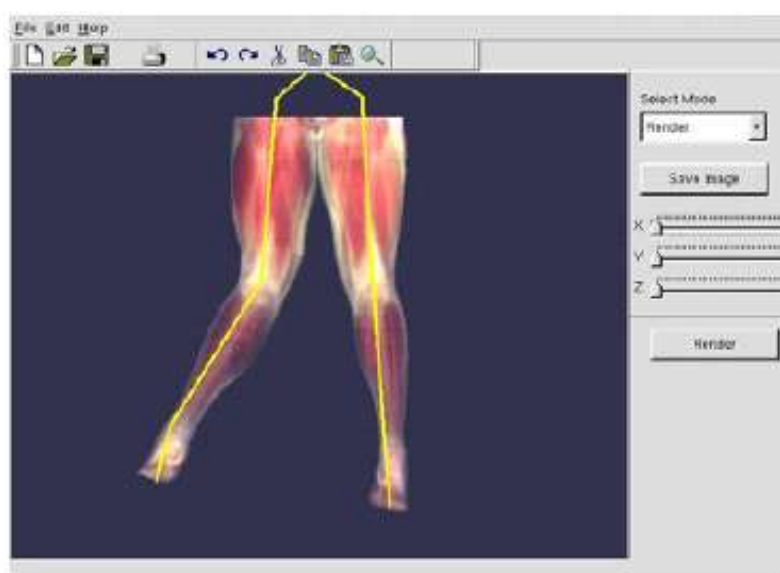
#### **14. TRABALHOS ANTERIORES ENVOLVENDO REALIDADE VIRTUAL E NEUROCIÊNCIA**

Recorrendo a RV, encontramos conceitos que tornam muito mais atrativa uma aplicação em qualquer área de aplicação. Na Neurociência, por exemplo, uma das grandes vantagens da construção de um ambiente RV para as simulações neurais é a possibilidade de executar diversas vezes um experimento sem se preocupar em danificar os tecidos neurais naturais ou com sacrifícios de animais, sem contar que não seria possível repetir o experimento com uma mesma estrutura neural, no caso de tecidos naturais. Este é o principal fator para o crescimento nas pesquisas envolvendo realidade virtual e experimentos biológicos. Um exemplo é o trabalho de SUBRAMANIAN e MARSIC (2001), que descreve um ambiente de RV chamado ViBE (Virtual Biology Experiments) como sendo um laboratório virtual voltado para pesquisas em biologia.

Acessando o ViBE, ilustrado na Figura 3.3, é permitido ao aluno executar diversas vezes um mesmo experimento, sem se preocupar com os gastos de materiais. Utilizando o ViBE, o aluno pode interagir com o software verificando, por exemplo, uma divisão celular, uma mitose, uma meiose. Outro simulador existente é o VOLEDIT, desenvolvido por SINGH (2003), que simula a criação de um esqueleto humano de forma realística, verificando suas deformações, possibilitando alteração de formas, texturas, transformações e possíveis rotações como ilustra a Figura 3.4. Neste software, desenvolvido com o auxílio da biblioteca OpenGL, descreve-se a necessidade da integração da realidade virtual com a computação gráfica relacionadas entre si. É interessante ressaltar que um tipo de software como este pode ser visto de diversas formas, dependendo de quais áreas se deseja atingir. Por exemplo, um radiologista daria um enfoque diferente de um patologista, que também enfocaria outros itens diferentes de um fisioterapeuta.



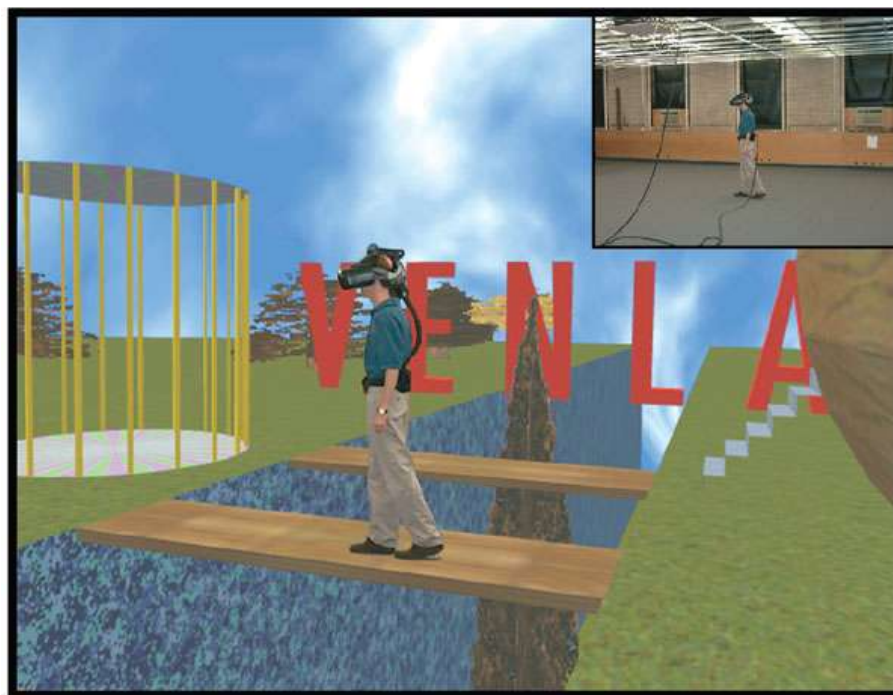
**Figura 3.3 - Layout VIBE – Microscópio Virtual (SUBRAMANIAN, 2001)**



**Figura 3.4 - VOLEDIT – Criação do Esqueleto Humano (SINGH, 2003)**

Outro simulador interessante a ser exemplificado é o SMART, desenvolvido por GRANT(1998), que utiliza luvas 3D e capacete. Surge então uma dificuldade bem definida na RV, a necessidade da calibração de equipamentos, no caso, a luva e o capacete, para que o usuário possa navegar no mundo desenvolvido por ele. A utilização do capacete gera um segundo problema bem discutido também, que é o ângulo de visão do capacete; dificuldade que encontraremos em diversos simuladores já desenvolvidos. Há também o trabalho de TARR e WARREN (2002) que demonstra algumas aplicações da realidade virtual voltada para a neurociência, psicologia e algumas outras áreas afins. Em seu artigo é descrito o laboratório VENlab (Virtual Environment Navigation Laboratory), que realiza as experiências de imersão possibilitando pesquisas

comportamentais. Neste laboratório os usuários exploram a anatomia neural e as conectividades entre elas, e, com isso, podem detectar tumores e como eles se desenvolvem no cérebro humano. A Figura 3.5 ilustra o funcionamento do VENlab.

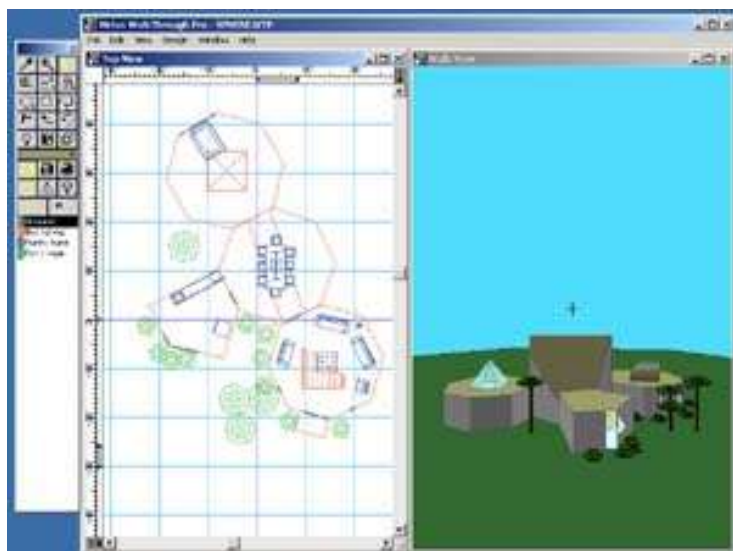


**Figura 3.5 - VENLab - Virtual Environment Navigation Laboratory (TARR e WARREN, 2002)**

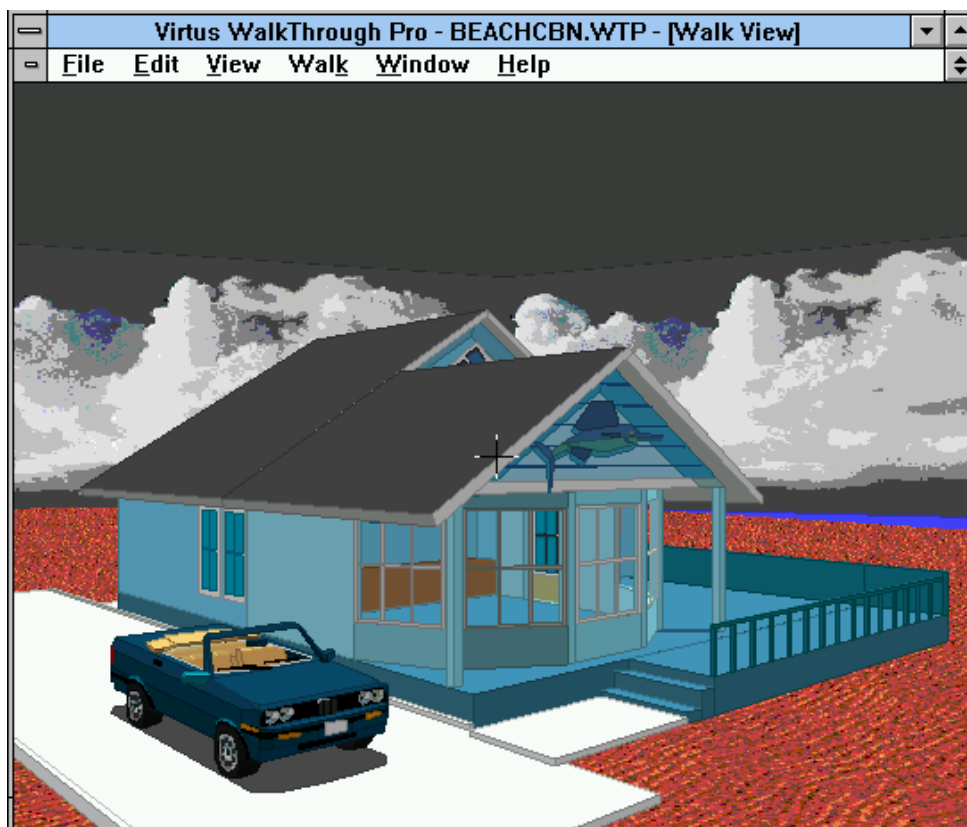
O projeto desenvolvido depende de um simulador para gerar os dados e esses dados são tratados e visualizados na tela; como também encontra-se no trabalho de COSTA, OLIVEIRA e MINGHIM (2000) que trata da visualização tridimensional de neurônios e mostra as diferentes formas dos neurônios dependendo da função que cada um executa. O crescimento neural não é feito de forma linear e sim em paralelo, como exemplifica SULLIVAN e KOCK (1990) e expõe um problema real que é a quantidade de dados sendo visualizados ao mesmo tempo e trata em seu artigo a necessidade de se montar um software que tenha atrativos e principalmente que não seja lento. Para sanar esta dificuldade, SULLIVAN e KOCK adotaram a redução do tamanho dos neurônios para a sua representação, que pode não ser uma solução mais interessante, mas resolveria o problema.

A realidade virtual é muito atrativa na área de educação, e para difundir mais esta área, no SIGGRAPH 97, MENOUSEK e WOLFE (1997) apresentaram um software para um conjunto de alunos, para experimentarem a idéia de

imersão em um ambiente de RV. Para se efetuar esta apresentação foi utilizado um software chamado “Virtus WalkThrough Pro” ilustrado na Figura 3.6 e na Figura 3.7, que possui algumas características bem intuitivas, funciona em qualquer plataforma e principalmente promove a visão estéreo. Para tal evento, dividiram o processo de aprendizagem em etapas bem simples e bem definidas, desde a idéia de navegação, a idéia de três dimensões até, por fim, a própria criação do mundo virtual.



**Figura 3.6 - Construção de uma cena utilizando Virtus WalkThrough Pro (MENOUSEK e WOLFE, 1997)**



**Figura 3.7 - Exemplo Virtus WalkThrough (MENOUSEK e WOLFE, 1997)**

Também voltado à educação ZAJTCHUK e SATAVA (1997) tratam em seu artigo um aspecto interessante e até então ainda não mencionado da realidade virtual. Além de frisar a idéia de que a realidade virtual é uma ferramenta muito poderosa para simulações e aprendizagem em salas de aula e principalmente na psiquiatria, com deficientes físicos, possibilitando interação real com o mundo, enfoca o trabalho com pessoas introvertidas, que no meio virtual tendem a se expor e crescer.

Em conjunto com especialistas em robótica, alguns neurocientistas desenvolveram um avatar aprendiz que simula os passos de um avatar mestre, como descrito no trabalho de BILLARD (2000). No software, à medida que o avatar mestre guiado por um usuário realiza uma determinada ação, do tipo, pular, levantar, andar, o aprendiz segue os passos fielmente.

Trabalhar com respostas em tempo real, calibração de câmeras, quantidade de informações trafegadas na rede são tópicos ainda bem discutidos quando se trabalha com softwares que utilizaram para a comunicação as redes, ou dispositivos de realidade virtual. BRYSON (1996) se baseou muito estes parâmetros quando montou, na Universidade da Carolina do Norte, um Túnel

de Vento Virtual, que para ser implantado levou principalmente em conta o desempenho do sistema, devido à quantidade excessiva de dados processados ao mesmo tempo.

## **1.1.**

## 15. CONCEITOS DE COMPUTAÇÃO GRÁFICA UTILIZADOS NO PROJETO

A Realidade Virtual utiliza diversos conceitos relacionados a Computação Gráfica, sendo de suma importância, destacar alguns desses conceitos utilizados no projeto desenvolvido. O conceito de rotação, importante para podermos rotacionar a cena gerada nos diversos eixos, o conceito de escala para efetuarmos um *zoom* na cena gerada, o conceito de iluminação, que torna a cena mais atrativa dependendo do ponto de luz acionado pelo usuário e o tipo de iluminação selecionada, entre outros conceitos relacionados a Computação Gráfica.

Computação gráfica é a área da Ciência da Computação que estuda a geração, manipulação e interpretação de imagens por meio de computadores (FOLEY, 1996; HEARN, 1997).

A área pode ser dividida em três grandes subáreas: síntese de imagens, processamento de imagens e análise de imagens.

- Síntese de imagens

Esta subárea tem como objetivo a produção de representações visuais a partir de especificações geométricas de seus componentes, sendo uma das áreas mais difundidas. É comumente confundida com a própria computação gráfica. Parte-se da descrição dos componentes tais como segmentos de reta, polígonos, esferas, poliedros, onde uma imagem é produzida em algum meio que possa ser visualizado. As imagens assim obtidas são exibidas em terminais, desenhadas em papel, impressas em filmes fotográficos ou impressoras matriciais, a jato de tinta ou a laser. Essas imagens são uma representação visual de objetos bi ou tridimensionais.

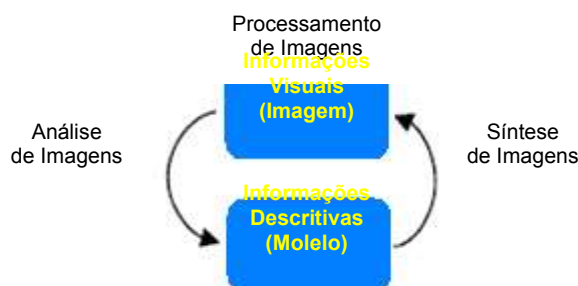
- Processamento de imagens

Esta subárea envolve técnicas de transformação de imagens em que tanto a imagem inicial quanto a imagem resultante são representadas visualmente. As técnicas empregadas tem como objetivo, geralmente, melhorar as características visuais da imagem. Como exemplo cita-se manipulação de contraste, correções geométricas, entre outras. As imagens iniciais são então transformadas em outras que apresentam as características visuais alteradas.

- Análise de Imagens

Visa obter a especificação dos componentes de uma determinada cena a partir da sua representação visual. Envolve, portanto, técnicas que efetuam a operação inversa à síntese de imagem. Esta área é principalmente relacionada à visão de robô.

A Figura 4.8 ilustra de uma maneira bem intuitiva estas três subdivisões:



**Figura 4.8 - Subdivisões da Computação Gráfica**

A área de Processamento de Imagens abrange operações que são realizadas sobre imagens e que resultam em imagens. A área de Síntese de Imagens abrange as operações de síntese de imagem, ou seja, a geração de uma visualização do modelo. A área de Análise de Imagens abrange as operações de análise dos objetos contidos na imagem e a geração de modelos matemáticos desses objetos.

## **16. ILUMINAÇÃO**

O alto grau de realismo alcançado hoje na Computação Gráfica deve-se, em grande parte, às diferentes formas de representar as superfícies dos objetos, e aos diversos modelos matemáticos que traduzem os efeitos físicos da interação da luz com as superfícies dos objetos. Pode-se separar o problema da aparência que uma superfície apresenta no computador em dois grupos: aqueles relacionados com a iluminação e aqueles relacionados com a textura da superfície.

Num ambiente físico qualquer, os objetos não são iluminados apenas pela lâmpada que acende no teto, nem tampouco apenas pela iluminação que entra pela janela durante o dia. Tanto a iluminação da lâmpada quanto a iluminação da janela iluminam os objetos, mas uma vez iluminados, estes objetos passam a refletir a luz em outros objetos, e estes, por sua vez iluminam outros e assim por diante. O fenômeno da iluminação real é, portanto, bastante complexo. Ademais, as superfícies parecem refletir mais ou menos a luz em função do



seu grau de polimento. Após estudos feitos em laboratório indicam que a luz refletida por uma superfície depende:

- do ângulo de incidência da luz, tanto na direção normal quanto tangencial.
- da composição espectral da luz incidente.
- da direção na qual a luz refletida está sendo medida.

A iluminação é uma técnica importante em objetos gráficos criados através do computador. Sem iluminação os objetos tendem a não apresentarem características realistas. O princípio da iluminação consiste em simular como os objetos refletem as luzes. A iluminação é dividida em três partes: propriedades dos materiais, propriedades de luzes e parâmetros globais de iluminação (WOO, 1999).

Segue a definição das quatro componentes que contribuem para a cor do *pixel*:

- Iluminação Ambiente : iluminação ambiente é uma luz que se difundiu tanto pelo ambiente que sua direção é impossível determinar, parece vir de todas as direções.
- Componente Difusa: é uma propriedade dos materiais de possuírem esta componente; uma vez que esta luz que vem em uma direção encontre uma superfície, ela é difundida igualmente em todas as direções, o que faz com que pareça igualmente luminosa, não importando de onde é visualizada. Qualquer luz vinda de uma posição particular ou direção tem provavelmente uma componente difusa;
- Especular: também é uma luz que vem de uma direção, porém tende a ser refletida numa única direção;
- Emissiva: simula a luz que se origina de um objeto; a cor emissiva de uma superfície adiciona intensidade ao objeto, mas não é afetada por qualquer fonte de luz; ela também não introduz luz adicional da cena.

Para adicionar luz a uma cena são necessários os seguintes passos:

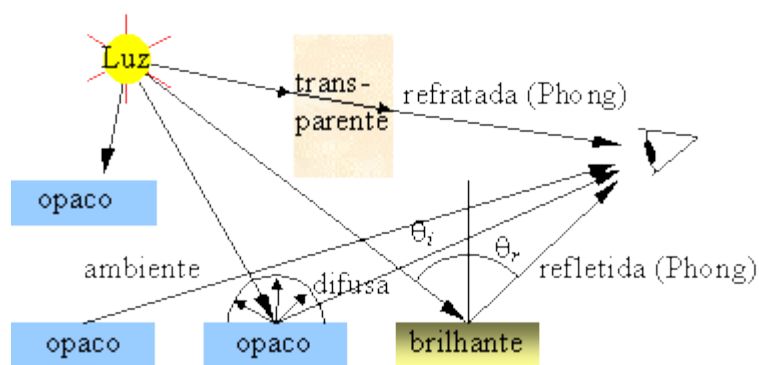
- Definir os vetores normais em relação às fontes de luz para cada face dos objetos. Estes vetores normais determinam a orientação de cada face em relação ao objeto relativo às fontes de luz;

- Criar, selecionar, e posicionar uma ou mais fontes de luzes;
- Criar e selecionar um modelo de iluminação que define o nível de luz ambiente global e a localização efetiva do ponto de visão (para os cálculos de iluminação);
- Definir as propriedades de materiais para os objetos na cena.

Mesmo utilizando modelos matemáticos para iluminação já existentes, estes modelos não são totalmente realísticos. Os seguintes problemas ocorrem:

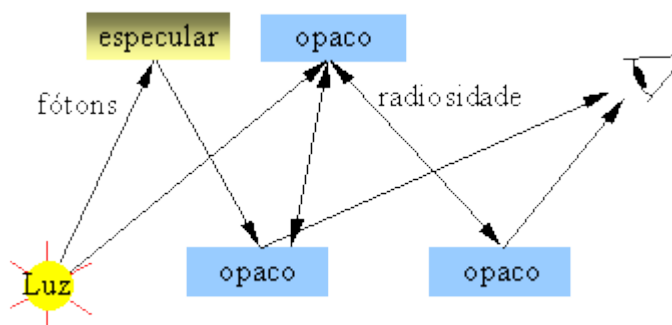
- No modelo da reflexão difusa, as partes não iluminadas ficam totalmente negras. Para compensar este efeito introduz-se um efeito de reflexão ou iluminação "ambiente" nos objetos, que simula uma iluminação uniforme em toda a superfície do objeto. Além de ser uma aproximação, o fenômeno físico real (onde um objeto ilumina o outro) não fica corretamente representado.
- Na reflexão especular, a fonte de luz não aparece refletida na superfície. Para simular o efeito de "brilho" no objeto, o modelo da reflexão especular é dividido em duas partes: a primeira reflete os objetos da cena, enquanto a segunda (denominada de modelo Phong) reflete apenas a fonte de luz.
- Se a superfície refletora for ligeiramente rugosa, então o reflexo será também ligeiramente fosco. Este efeito é difícil de modelar, porém alguns programas gráficos, como o MegaPov, possuem esse recurso.
- A refração não produz o efeito de cáustica. Este efeito é obtido quando a luz, ao atravessar uma lente convergente, por exemplo, cria um ponto altamente luminoso num anteparo colocado logo após a lente. Há, também neste caso, alguns programas gráficos que conseguem representar razoavelmente bem este efeito (MegaPov, por exemplo).

Justamente por estas limitações, este modelo de iluminação é conhecido como iluminação local, pois a característica de cada objeto depende apenas da fonte de luz, e não dos outros objetos ao seu redor, conforme pode ser visto no esquema mostrado na Figura 4.9. Infelizmente, não existem ainda modelos "globais" que sejam completos. Existem, na verdade, modelos que se aplicam a um ou outro fenômeno isoladamente.



**Figura 4.9 - Modelo de iluminação local.**

Para considerar os efeitos de iluminação de um objeto no outro, deve-se considerar que as superfícies podem ser difusas ou especulares. O efeito da reflexão entre objetos opacos (ou pela parcela difusa da reflexão), pode ser modelado pela técnica da radiosidade. Por sua vez, a iluminação provocada pelo reflexo da luz em objetos especulares é modelada pelo mapeamento de fótons, conforme ilustra a Figura 4.10.



**Figura 4.10 - Interações de luz modeladas pela radiosidade e por mapa de fótons**

## 17. TRANSPARÊNCIA

As funções de mistura de cores (*blending*) suportam efeitos como transparência, que podem ser usadas nas simulações de janelas, copos, e outros objetos transparentes (FOLEY, 1996; HEARN, 1997).

Quando “*blending*” está habilitado, o valor de alfa é usado freqüentemente para combinar o valor de cor do fragmento que está sendo processado com o do *pixel* armazenado no *framebuffer*. O efeito de “*blending*” acontece após a cena ter sido processada e convertida em fragmentos, mas antes os pixels finais são retirados do *framebuffer*.

Durante o “*blending*”, valores de cores do fragmento entrante (a fonte) são combinados com os valores de cor do *pixel* atualmente armazenado (o destino)

em um processo de duas fases. Primeiro deve ser especificado como calcular os fatores da fonte e do destino. Estes fatores são quádruplas de RGBA que são multiplicados por cada valor de componente R, G, B, e A na fonte e no destino, respectivamente. Então as componentes correspondentes nos dois conjuntos de quádruplas RGBA são somadas, como ilustra a Eq 4.1:

$$(RsSr+RdDr, GsSg+GdDg, BsSb+BdDb, AsSa+AdDa) \quad (Eq\ 4.1)$$

sendo que (Sr, Sg, Sb, Sa) são as componentes da fonte e (Dr, Dg, Db, Da) são as componentes do destino.

## 18. TEXTURA

Uma forma de incluir as texturas é a aplicação de imagens em uma superfície, conforme se procede na aplicação de decalques em um brinquedo. Quando uma imagem é mapeada sobre um objeto, a cor de cada *pixel* do objeto é modificada por uma cor correspondente da imagem. A imagem é armazenada em uma matriz como uma imagem contínua e deve ser reconstruída a partir desta matriz. Logo, a imagem deve ser deformada para adequar-se a qualquer distorção (causada, talvez, pela perspectiva) no objeto a ter a imagem projetada. Então, esta imagem deformada é filtrada para remover componentes de alta frequência, que geram o efeito de “*aliasing*”<sup>1</sup> no passo final. Este mapeamento aumenta as cores especificadas para uma primitiva geométrica com as cores armazenadas na imagem. Uma imagem pode conter conjuntos de cores 2D ou 3D que são chamadas “*texels*” (FOLEY, 1996; HEARN, 1997).

Alguns dos muitos usos de texturas incluem :

- Simulação de materiais como madeira, tijolos ou granito;
- Redução da complexidade (número de polígonos) de um objeto geométrico;
- Simulação de superfícies refletivas como espelhos ou chão polido.

Um objeto de textura armazena dados de textura. Podem ser controladas muitas texturas simultaneamente, podendo ser utilizadas a qualquer momento desde que previamente carregadas. A utilização de objetos de texturas, em geral, é o modo mais rápido para aplicação de texturas e resulta em ganhos de

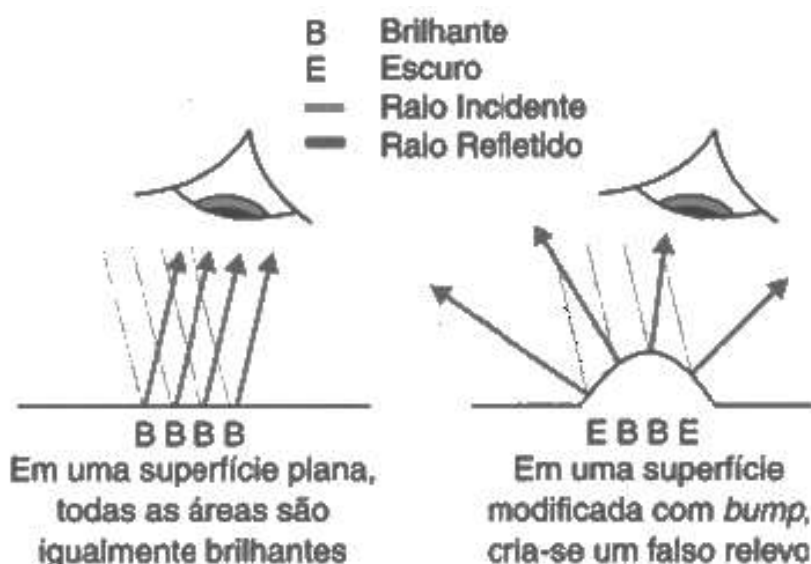
<sup>1</sup> Distorção visual na imagem movimentada na tela. Geralmente perceptível nas bordas da imagem distorcida no espaço 2D ou 3D.

desempenho porque quase sempre é mais eficiente ligar (usar novamente) um objeto de textura existente do que recarregar uma imagem de textura. Existe também outra forma de se trabalhar com texturas, denominada *Bump Map*.

No “*bump map*” as propriedades de refração da luz são usadas para indicar quais partes são mais escuras ou mais claras em um objeto. A idéia básica do “*bump map*” é perturbar o vetor normal em vários pontos da superfície, criando assim uma ilusão de que algumas partes da superfície estariam elevadas ou rebaixadas, sem que para isso seja alterada a geometria. Com a finalidade de uma perturbação adequada da normal para produzir determinado efeito é usado um mapa de altura, fazendo com que zonas de maior intensidade de cor (branco) apareçam em alto relevo e as de menor intensidade (preto) apareçam em baixo relevo, ou vice-versa. Matematicamente:

$$n' = n + \frac{F_u(n * P_v) - F_v(n * P_u)}{\|n\|}$$

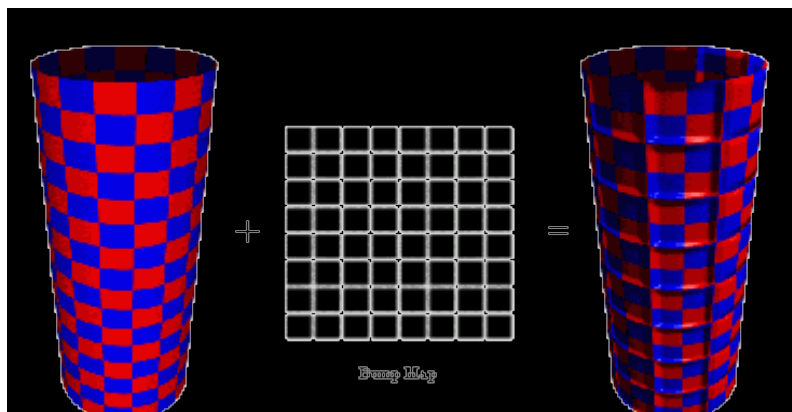
sendo  $n$  a normal da superfície,  $P_u$  e  $P_v$  as derivadas parciais da superfície nas direções  $u$  e  $v$  respectivamente.  $F_u$  e  $F_v$  são os gradientes do *bump map*, conforme ilustra a Figura 4.11.



**Figura 4.11 - Propriedades de refração da luz em *Bump Map***

Segue na Figura 4.12 um exemplo de aplicação de textura utilizando *bump map*.

**Figura 4.12 - Exemplo de *Bump Map***



Cilindro c/ mapa de  
texturas difuso

Cilindro c/ mapa de texturas  
difuso + *bump map*

No projeto desenvolvido utilizamos a ferramenta GLTexture para definir a textura utilizada nos galhos dos neurônios.

### 19. TRANSFORMAÇÕES GEOMÉTRICAS

As transformações geométricas são usadas para manipular um modelo, ou seja, mover, rotacionar ou alterar a escala de um objeto. A aparência final da cena ou do objeto depende muito da ordem na qual estas transformações são aplicadas. As operações de translação, escala e rotação são executadas utilizando uma multiplicação de matrizes. A idéia central destas transformações é que elas podem ser combinadas em uma única matriz, de tal maneira que várias transformações geométricas possam ser aplicadas por uma única operação. Toda vez que se realiza uma transformação geométrica é feita uma nova composição de matrizes, e esta nova matriz é utilizada para desenhar os objetos a partir daquelas novas coordenadas (WOO, 1999; OPENGL, 2004).

A translação é feita pela função  $glTranslate\{f,d\}(Tx, Ty, Tz)$ , que pode receber três números *float* ou *double* como parâmetro. Neste caso, a matriz atual é multiplicada por uma matriz de translação baseada nos valores dados.

A rotação é feita pela função  $glRotate\{f,d\}(\text{Ângulo}, x, y, z)$ , que pode receber quatro números *float* ou *double* como parâmetro. Neste caso, a matriz atual é multiplicada por uma matriz de rotação de "Ângulo" graus ao redor do eixo definido pelo vetor "x,y,z" no sentido anti-horário.

A escala é feita pela função  $glScale\{f,d\}$  ( $E_x$ ,  $E_y$ ,  $E_z$ ), que pode receber três números *float* ou *double* como parâmetro. Neste caso, a matriz atual é multiplicada por uma matriz de escala baseada nos valores dados.

## 20. SPLINES

O conceito de *spline*, vem trazendo grande contribuição à área da computação gráfica desde o aparecimento dos primeiros softwares. Para tal, *splines* vêm sendo indiretamente utilizados para elaboração de trabalhos criados em um computador, pois possibilitam a criação de qualquer tipo de curva, a critério do “artista”, utilizando os pontos de controle e pesos relacionados aos pontos. Uma *spline* é uma curva definida matematicamente por dois ou mais pontos de controle. Os pontos de controle que ficam na curva são chamados de nós. As *splines* podem ser divididos em duas categorias:

- *Splines* de interpolação que passam por todos os pontos de controle;
- *Splines* de aproximação que passam perto de todos os pontos de controle.

Usualmente, os *splines* de aproximação são curvas suaves, dado que as *splines* de interpolação podem ter curvas mais acentuadas perto dos nós. Na Figura 4.13, a curva que passa pelos pontos  $P_0$ ,  $P_1$ ,  $P_2$  e  $P_3$  é um *spline* interpolador (especificamente, um *spline linear*) e a curva que passa através de  $P_0$  e  $P_3$ , mas não por  $P_1$  e  $P_2$ , é um *spline* de aproximação (especificamente, um *spline Bézier*).

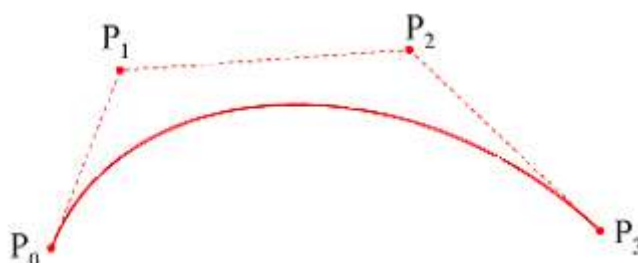


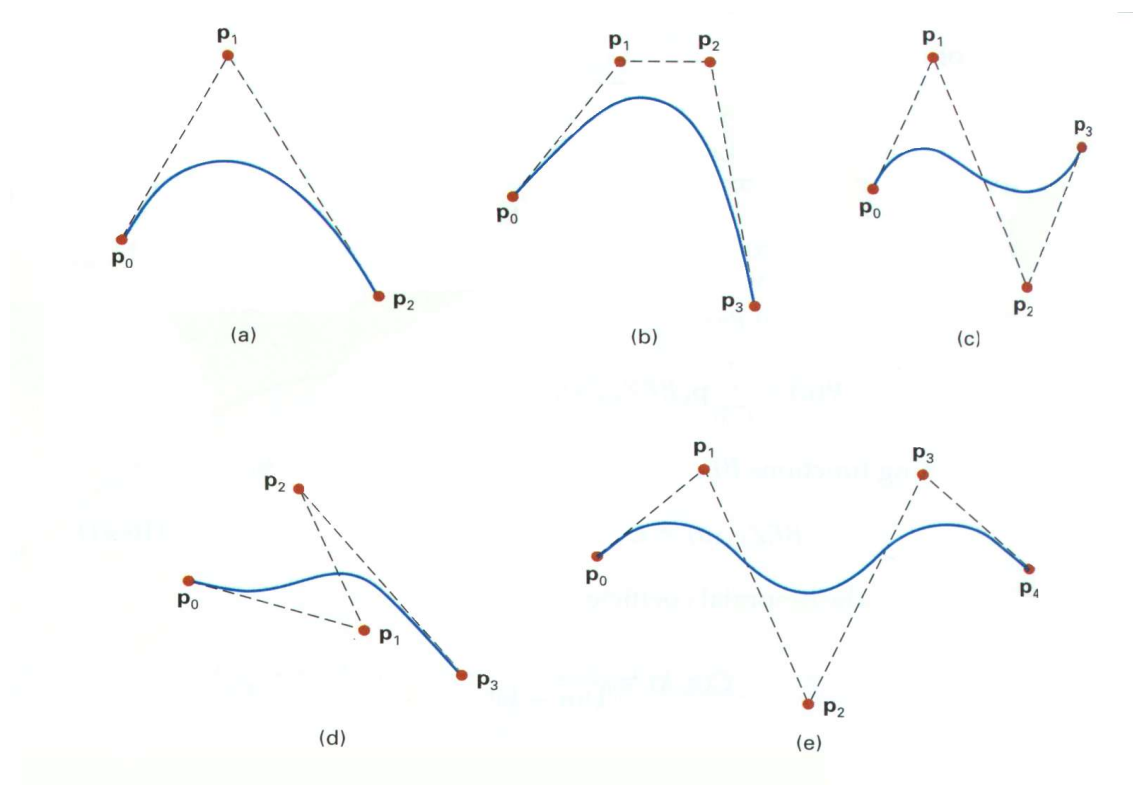
Figura 4.13 - Definição *Spline*

## 21. Curvas Bézier

As curvas Bézier foram desenvolvidas pelo engenheiro francês Pierre Bézier para usar no projeto de corpos de automóveis da Renault. Elas possuem

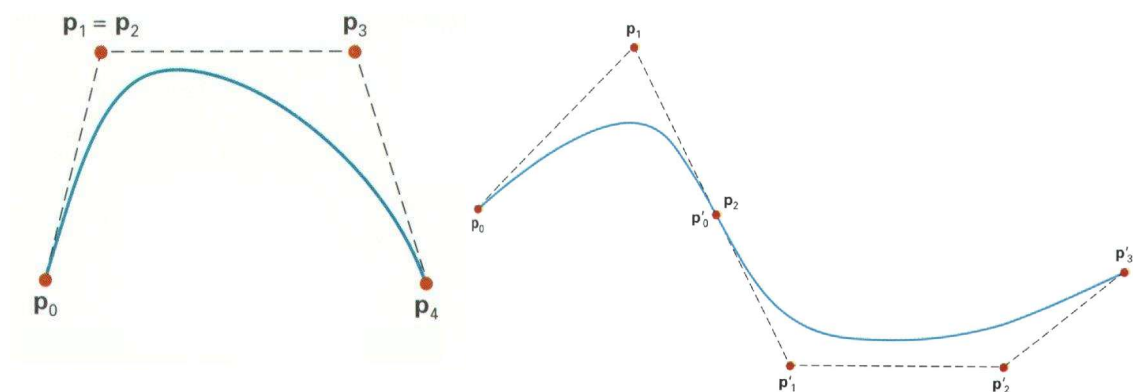
propriedades que as tornam altamente usuais e convenientes para projetos de curvas e superfícies, além de serem de fácil implementação.

Para calcular a curva ilustrada na Figura 4.14, pode ser usado qualquer quantidade de pontos de controle, mas o número de pontos determina o grau do polinômio de Bézier.



**Figura 4.14 - Exemplos de curvas de Bézier**

Uma curva Bézier sempre passa pelo primeiro e último ponto de controle e está sempre situada dentro do limite do polígono convexo dos pontos de controle. Podem-se especificar múltiplos pontos de controle em uma única posição de coordenada para dar um peso maior àquela posição, como no exemplo da Figura 4.15:





- o grau do polinômio pode ser estabelecido independentemente do número de pontos de controle;
- B-Splines permitem controle local sobre a forma de uma curva ou superfície *splines*.

Além de serem mais complexas que as Bézier, possuem as seguintes propriedades:

- Natureza local da definição, tal que se movermos um ponto  $p_i$ , somente os segmentos da curva associados com este ponto serão afetados;
- Uma curva B-Spline não necessariamente passa pelos pontos de controle (mesmo os iniciais);
- Se o ponto de controle inicial é repetido 3 vezes então a curva passará sobre ele. Isto é semelhante para o último ponto de controle também. Assim, uma seqüência de pontos de controle  $P_0 P_0 P_0 P_1 P_2 P_3 P_4 P_4 P_4$  gerará uma curva que passa pelos pontos  $P_0$  e  $P_4$ .

## **23.ESPECIFICAÇÃO DO PROJETO E RESULTADOS OBTIDOS**

Neste capítulo será apresentado o desenvolvimento do projeto em si, indicando as ferramentas utilizadas para a elaboração do mesmo e todas as funcionalidades do projeto, incluindo todas as ações que o usuário pode realizar para interagir com o projeto.

## **24. DESCRIÇÃO DAS FERRAMENTAS**

No projeto foi utilizado o ambiente Delphi com o auxílio de um conjunto de componentes gráficos denominado GLScene, que é uma biblioteca que contém objetos já desenvolvidos para elaboração de cenas tridimensionais. e permite que se utilize como base alguns conceitos do próprio OpenGL.

OpenGL é uma biblioteca de rotinas gráficas de modelagem, manipulação de objetos e exibição tridimensional que permite a criação de aplicações que usam Computação Gráfica. Seus recursos permitem ao usuário criar objetos gráficos com qualidade, de modo rápido, além de incluir recursos avançados de animação, tratamento de imagens e texturas. É possível também obter visualização em vários ângulos (OPENGL, 2006; WOO, 1999).

Diante das funcionalidades providas pelo OpenGL, tal biblioteca tem se tornado um padrão amplamente utilizado na indústria de desenvolvimento de aplicações. O fato de se tornar um padrão mundial promove a facilidade na aprendizagem, a estabilidade nas rotinas e os resultados visuais consistentes para qualquer sistema de exibição concordante com este padrão. Diversos jogos, aplicações científicas e comerciais têm utilizado OpenGL como ferramenta de apresentação de recursos visuais, principalmente com a adoção deste padrão por parte dos fabricantes de placas de vídeo destinadas a consumidores domésticos. Todas as rotinas do OpenGL são implementadas na linguagem C, tornando fácil sua utilização em qualquer programa escrito em C ou C++ (OPENGL, 2006; WOO, 1999).

Entre os recursos gráficos disponíveis pelo OpenGL, pode-se destacar os seguintes: modos de desenho de pontos; ajuste de largura de linhas; aplicação de transparência; ativação/desativação de serrilhamento; mapeamento de superfícies com textura; seleção de janela de desenho; manipulação de fontes/tipos de iluminação e sombreado; transformação de sistemas de

coordenadas; transformações em perspectiva e combinação de imagens (OPENGL, 2006; WOO, 1999).

Criou-se então uma biblioteca mais completa para Delphi, denominada GLScene, contendo componentes que possibilitam a criação de um ambiente 3D, utilizando câmeras, objetos tridimensionais e rotinas do OpenGL (GLSCENE, 2006).

A seguir serão descritos alguns componentes da biblioteca GLScene utilizados no projeto. O GLScene é o principal componente principal da biblioteca mencionada. A partir dele colocam-se todos os outros objetos 3D. Outro componente que utilizado é o GLCamera, que, como o próprio nome diz, é responsável pela colocação da câmera em posições definidas pelo usuário. Juntamente com o GLCamera, pode-se citar o GLSceneViewer, responsável pela visualização da cena 3D. Pode-se definir texturas e cores utilizando GLMaterialLibrary, que contém esses dados armazenados. Dentro ainda da biblioteca GLScene encontra-se o GLCadencer, objeto que define o tempo de atualização de uma determinada cena gerada e é utilizado em animações.

Esta biblioteca possui alguns componentes já desenvolvidos que facilitam a elaboração de um ambiente gráfico 3D e é *freeware*. A Figura 5.1 ilustra o editor GLScene com alguns dos objetos que estão sendo utilizados no ambiente. Primeiramente, define-se um objeto denominado GLSceneViewer, que determina onde a cena será gerada; todos os demais objetos, para serem vistos devem ser colocados e relacionados ao GLSceneViewer, ou seja, onde a cena toda é gerada.



**Figura 5.16 - Editor GLScene**

## 25. VISÃO GERAL DO AMBIENTE

O objetivo principal deste projeto é a visualização com dados recebidos de um simulador neural, com uma estrutura de dados pré-definida, promovendo interatividade e imersão do usuário, ou seja, permitir que o usuário possa analisar o crescimento e formação de uma estrutura neural e alguns aspectos relacionados à neurogênese e plasticidade destas estruturas, como por exemplo, a reorganização neural na presença de campos tróficos (campos de atração ou repulsão) e na apoptose (morte neural natural).

Atualmente, já existe um simulador neural desenvolvido por COELHO (2002) em que uma das funções é gerar células com características morfológicas estatisticamente idênticas às células naturais e, para isso, utiliza *L-Systems* estocásticas e modificadas para considerarem medidas morfológicas extraídas de células naturais. O ambiente de RV fornece o suporte necessário ao simulador para que o usuário possa analisar o que ocorre durante o crescimento destas células.

As estruturas neurais atualmente utilizadas no ambiente em desenvolvimento são geradas utilizando-se características biológicas das células, conceitos matemáticos e de computação, tais como, tipo de célula, conexão, propriedades morfológicas de cada neurônio, funções de distribuição (funções de densidade e probabilidade acumuladas) e *L-Systems* (COELHO, 2003). O programa de simulação gera células neurais artificiais, que possuem características morfológicas reais (COELHO, 2003). Para o ambiente de RV, consideramos uma estrutura neural como sendo dados armazenados na memória do computador.

Os segmentos que formam os neurônios são representados no projeto utilizando-se cilindros, o qual fora escolhido com base no modelo do cabo. Os cilindros, por sua vez, são mais simples de se renderizar devido à otimização dos pontos gerados, ou seja, com informações de dois pontos (x,y,z) e um raio é possível descrever um cilindro, resultando, assim, em um ganho de velocidade de processamento e um menor tráfego na rede, já que uma menor quantidade de dados será enviado para o ambiente quando comparado ao envio de todos os pontos que formam as células neurais.

O ambiente desenvolvido possibilita ao usuário executar interações do tipo alterar o ponto de vista, visualizar o crescimento neural em diferentes ângulos

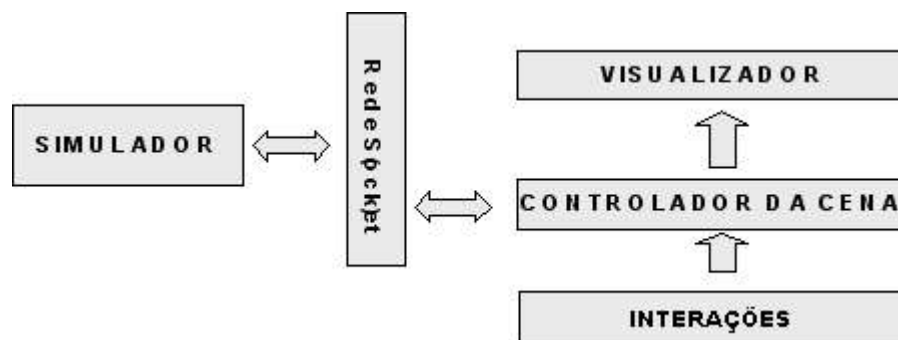
e ampliar a cena utilizando a opção *zoom*. O simulador já implementa a apoptose considerando que as células que conseguirem menos conexões não sobrevivem, uma vez que a troca de informação entre estas células e as outras existentes na estrutura será muito pequena, o que resultará em pouca atividade delas na estrutura. Após a morte, naturalmente ocorre uma reorganização neural das células vizinhas para que novas conexões sejam formadas, suprimindo, desta forma, a falta das células mortas. Esta morte e reorganização neural são acompanhadas pelo ambiente de realidade virtual, o que possibilita estudos mais detalhados sobre os fatores que levam algumas células a efetuarem mais conexões que outras. Outro tipo de reorganização que o simulador considera é o surgimento de um campo trófico (de atração ou repulsão) na estrutura neural. As arborizações dendríticas se reorganizam em direção ao campo (no caso de campo de atração), o que poderá resultar em novas conexões. Este tipo de reorganização também pode ser analisada no ambiente proposto, o que possibilitará um estudo mais detalhado sobre o impacto dos diferentes tipos de campos tróficos na estrutura neural.

Primeiramente, da biblioteca GLScene, colocamos o GLSceneViewer, que é o “visualizador da cena”. A partir daí os demais objetos podem ser adicionados ao sistema, no caso os diversos *pipes* e câmeras em posições alternadas podem gerar visualizações diferentes. Além das câmeras, o usuário pode interagir com o ambiente utilizando simplesmente o mouse convencional, ou equipamentos de realidade virtual, tais como óculos, que possibilitem a visão estéreo, luvas, capacetes ou outros dispositivos de visualização e interação.

Para um projeto futuro, a visualização dos neurônios se dará pela visão estéreo, que é um ponto fundamental em um sistema de realidade virtual. Porém, por enquanto ainda não está implementada.

## **26. ARQUITETURA DO PROJETO**

De acordo com o esquema ilustrado na Figura 5.17, pode-se definir o funcionamento do ambiente como um todo, possibilitando a visualização global e definindo as possibilidades de interação e navegação existentes no projeto.

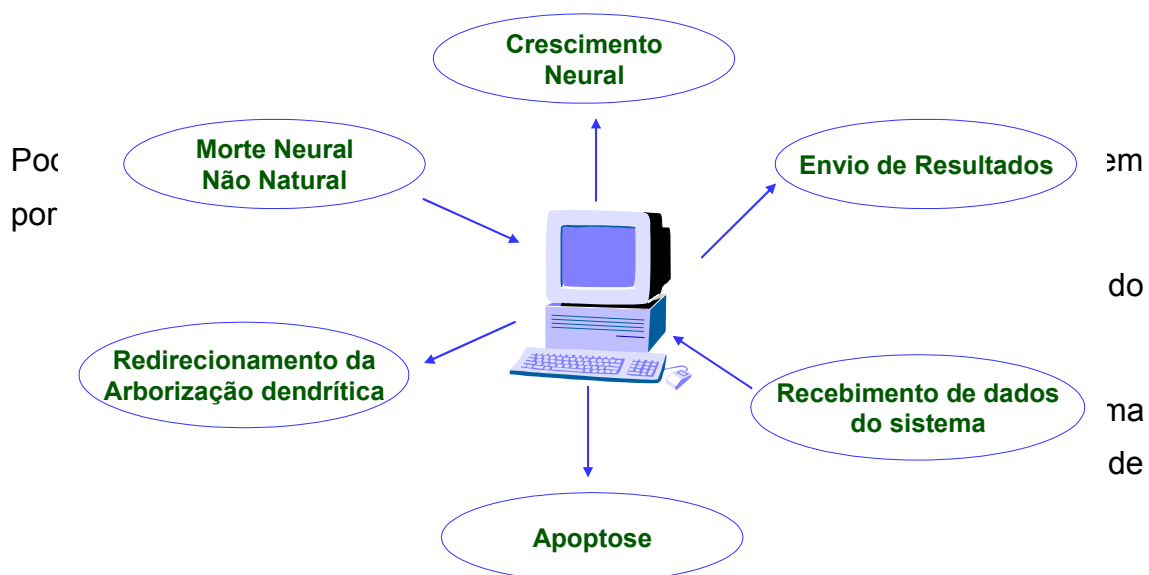


**Figura 5.17 - Esquema Funcional**

Pode-se perceber na Figura 5.17, que o simulador envia e recebe dados da rede, através de *sockets*, sendo que ele sempre estará enviando e recebendo informações do controlador da cena. A saída do simulador representa informações sobre o crescimento, formação e modificações no crescimento das estruturas neurais, como pode ocorrer quando acontece uma apoptose. O controlador da cena captura estes dados e os envia para o visualizador. O usuário participa do ambiente realizando interações durante e após o crescimento da estrutura neural, tais como, zoom para aproximação ou afastamento, rotação da cena, *spline* (ver Ferramentas – *Spline*).

Pode-se exemplificar que o ambiente desenvolvido é composto de três entidades distintas:

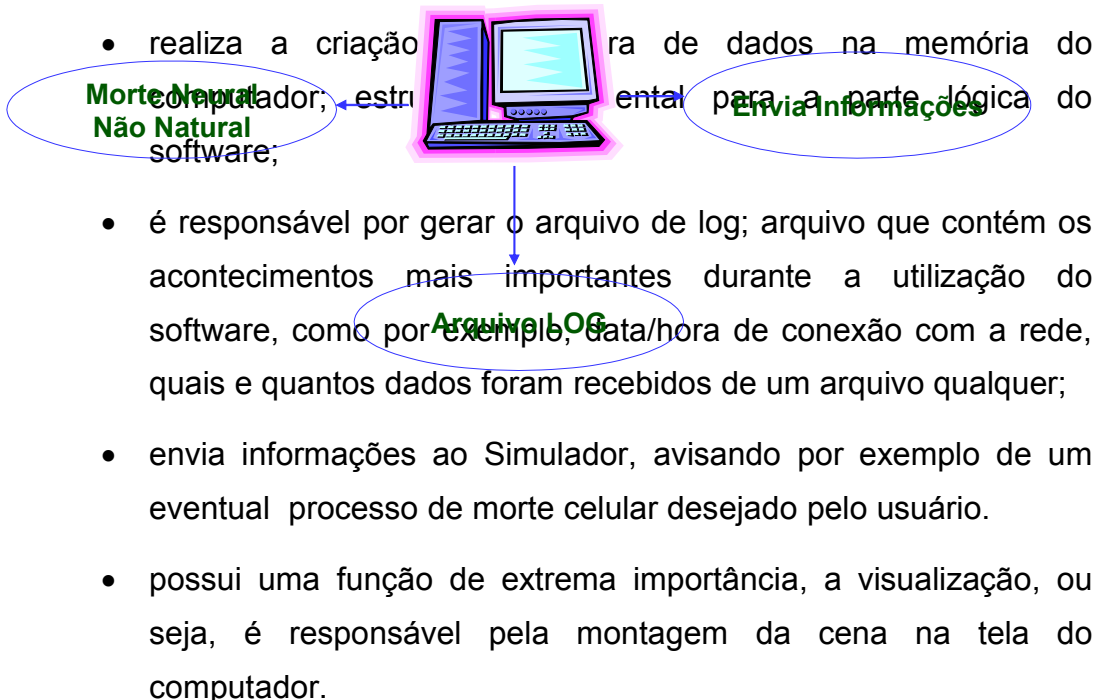
- O simulador, ilustrado na Figura 5.18;
- O ambiente de realidade virtual, ilustrado na Figura 5.4; Figura 5.20
- O usuário, ilustrado na Figura 5.20.



- realizar morte de células que o usuário determinar que deverão morrer (neste caso, poderia estar simulando uma necrose). Todas as informações relacionadas ao crescimento ou modificações nas estruturas neurais são enviadas ao visualizador.

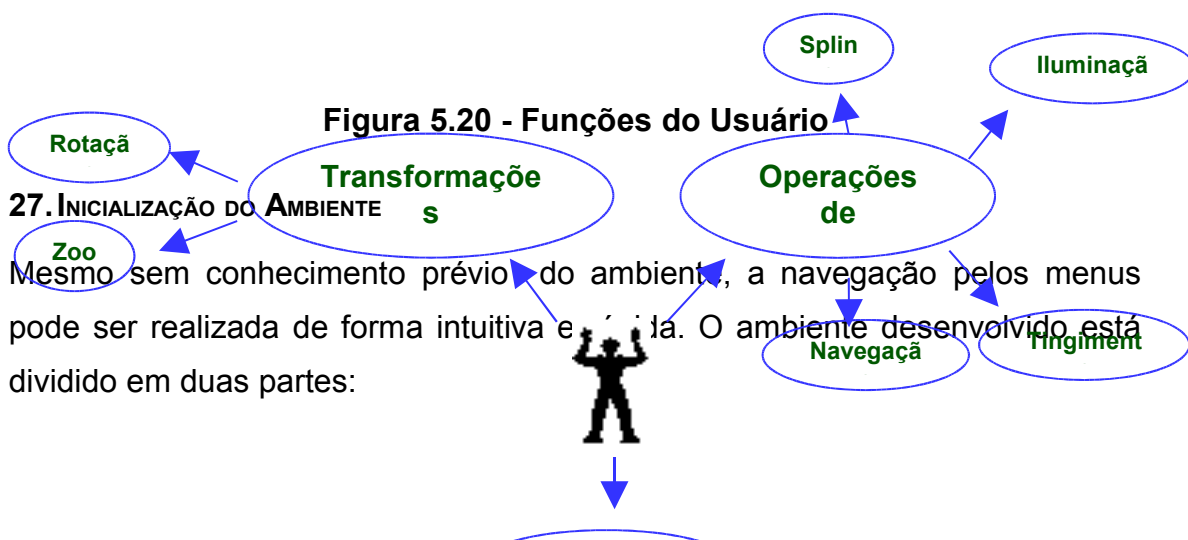
**Figura 5.19 - Funções do Ambiente de Realidade Virtual**

O Ambiente de Realidade Virtual, tem como principal função o recebimento de dados enviados pelo simulador, como ilustra a figura 5.4. É ele propriamente dito que:



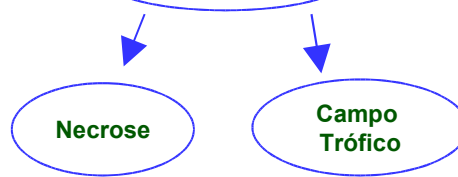
- realiza a criação de dados na memória do computador; estrutura lógica do software;
- é responsável por gerar o arquivo de log; arquivo que contém os acontecimentos mais importantes durante a utilização do software, como por exemplo, data/hora de conexão com a rede, quais e quantos dados foram recebidos de um arquivo qualquer;
- envia informações ao Simulador, avisando por exemplo de um eventual processo de morte celular desejado pelo usuário.
- possui uma função de extrema importância, a visualização, ou seja, é responsável pela montagem da cena na tela do computador.

O usuário por sua vez, como ilustra a Figura 5.20, tem uma gama de opções que podem ser acionadas, desde manipulação de arquivos locais, como configuração da rede, como as principais interações ou ferramentas disponíveis no software que serão propriamente descritas na seção 5.5.

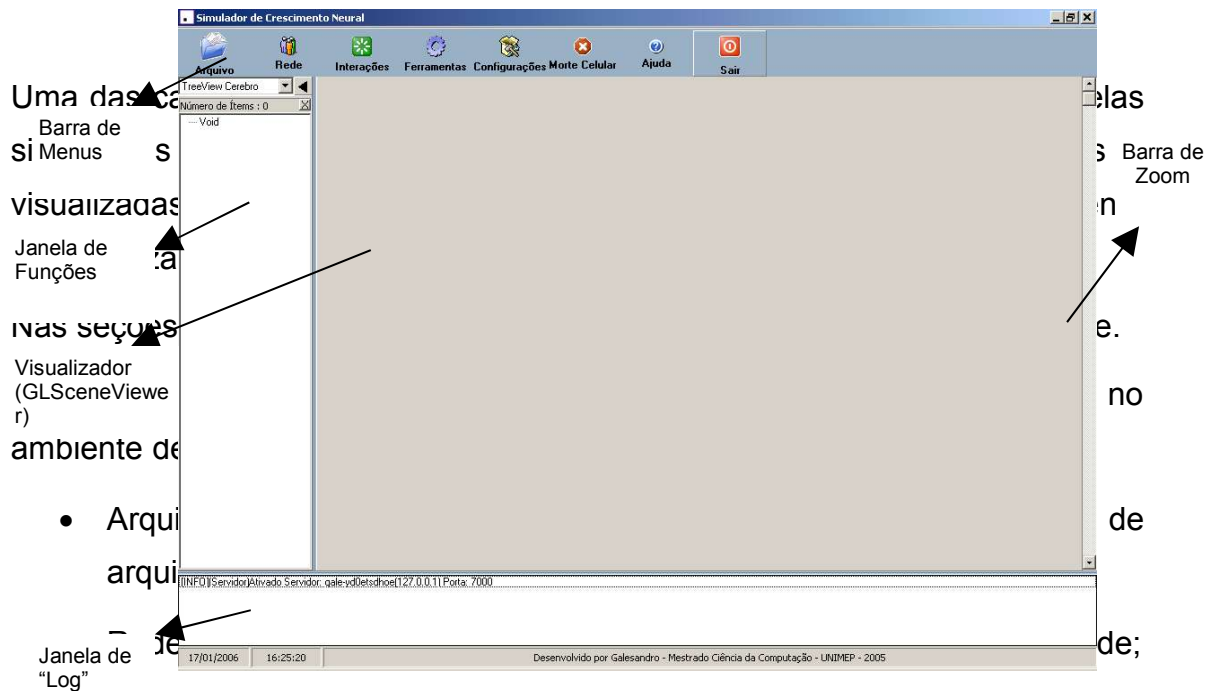


**27. INICIALIZAÇÃO DO AMBIENTE**

Mesmo sem conhecimento prévio do ambiente, a navegação pelos menus pode ser realizada de forma intuitiva e rápida. O ambiente desenvolvido está dividido em duas partes:



- a parte lógica, que engloba a montagem da estrutura dos dados, ou seja, a criação na memória do computador de uma árvore propriamente dita;
- a parte gráfica do ambiente. Os parâmetros para a primeira parte são transparentes para o usuário (o usuário não tem acesso), porém a parte gráfica do ambiente pode ser ilustrada na Figura 5.6.



- Arquivo
- Interações: contém alguns itens de interatividade entre o usuário e o ambiente desenvolvido, como as transformações geométricas e inserção de campos tróficos;
- Ferramentas: contém itens que podem modificar a visualização, tais como iluminação, *spline*;
- Configurações: opção que permite o usuário definir alguns parâmetros do projeto, tais como habilitar ou não geração do arquivo de log, inicializar a conexão com a rede no momento em que o software é inicializado, entre outras funções descritas posteriormente;
- Morte Celular: ilustra a morte de algumas células na estrutura neural, seja por Necrose ou por Apoptose;
- Ajuda: contém a informações básicas do projeto e o tópico de Ajuda (*help*);
- Sair: finaliza o projeto.



Todos os itens mencionados serão descritos detalhadamente no decorrer desta dissertação, com as funcionalidades e características de cada opção.

Ao inicializar o software, é verificado em um arquivo .ini, as configurações iniciais do projeto, tais como a apresentação da tela de *splash screen*<sup>2</sup> e a criação de um arquivo de log,

## 28. RECEPÇÃO DOS DADOS

Os dados podem ser recebidos através de uma conexão *socket* (TCP/IP) ou lidos de um arquivo local como mostrado na Figura 5.22. Os dados são processados linha a linha obedecendo a seguinte estrutura:

Número Neurônio	Número Galho Pai	Número Galho Filho	Coord X	Coord Y	Coord Z	Raio
-----------------	------------------	--------------------	---------	---------	---------	------

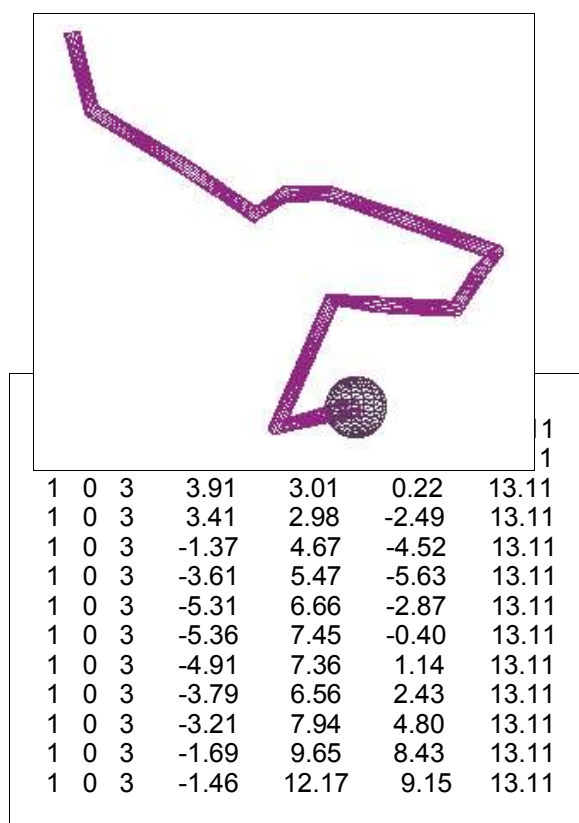
**Figura 5.22 - Estrutura de recepção dos dados**

Neste caso, o número do neurônio especifica qual o neurônio que está sendo gerado ou incrementado, cientes de que uma estrutura neural é formada por um conjunto de neurônios. O número do galho pai identifica a origem do galho e o número do galho filho especifica a ramificação; lembrando que um neurônio pode ou não ser formado por um conjunto de ramificações. As três coordenadas especificam o ponto 3D (Coord X, Coord Y, Coord Z) onde será fixado um determinado ponto para que seja crescido uma das ramificações do neurônio. A estrutura lida envia ao simulador apenas pontos, ou seja, cada linha corresponde a um ponto, e a medida que são lidos dois pontos um determinado cilindro é gerado. Pode-se haver continuidade no cilindro já montado ou não. O ponto final deste cilindro pode ser o inicial de outro, formando na verdade um segmento de um mesmo cilindro, ou as informações lidas podem formar um novo cilindro em qualquer outro lugar da cena, sendo assim criada uma ramificação.

Considere que cada galho é formado por um conjunto de segmentos. Para finalizar a entrada dos dados, especificamos o raio do segmento, que representará a espessura do galho, e este raio poderá ser diferente de galho para galho. Segue na Figura 5.23, um exemplo de como os dados serão lidos.

<sup>2</sup> Formulário inicial do sistema que apresenta a imagem de um neurônio juntamente com o nome do sistema.

Note que os dados da leitura são numéricos, sendo os três primeiros inteiros; e o restante, reais. Antes de serem enviados ao visualizador, eles são normalizados (dividimos os valores lidos por 10 (padrão) ou qualquer outro valor especificado pelo usuário) com o objetivo de diminuir o tamanho da cena gerada e, utilizando um objeto do GLScene denominado *Pipe*<sup>3</sup>, a visualização é gerada. É importante ressaltar que todo neurônio é composto de neuritos (dendritos e axônio) e de um único soma, que será representado em nossa base de dados como sendo o número do galho pai com o valor 0 (zero) e também o número do galho filho com o valor 0 (zero). Esta informação foi colocada para diferenciar os dados de crescimento de um neurito, do soma do mesmo neurônio. De acordo com os dados da Figura 5.23 pode-se facilmente perceber tal informação; a primeira linha representa o soma do neurônio de número 1, os dados das linhas seguintes são usados para promover o crescimento do galho neste neurônio gerado. Não há necessidade que todos os somas sejam definidos no início do arquivo lido, porém definiu-se que antes de indicar qualquer crescimento neural de um determinado neurônio fosse definido primeiramente a posição do soma correspondente a este neurônio.



(b)

**Figura 5.23 - (a) - Leitura dos dados (b) - Cena gerada com os dados lidos**

Os dados lidos (Figura 5.23 (a)) são enviados ao simulador, que gera a estrutura dos dados em árvore e plota os dados lidos no visualizador como ilustra a Figura 5.23(b).

<sup>3</sup> Pipe: objeto que possibilita, com auxílio de algumas propriedades, desenhar um cilindro, tendo ainda a possibilidade de definir um raio diferente para cada extremo do cilindro.

Pode-se observar na Figura 5.24 a existência de três neurônios, localizados nas três primeiras linhas do arquivo lido, indicando as coordenadas x,y,z e raio respectivos. Percebe-se, no entanto, a existência desses 3 neurônios quando se observa o terceiro dado de cada linha; lendo-se 0 (zero) entende-se que é o soma de um neurônio.

<b>Figura</b>	1	0	0	0.00	0.00	0.00	3	<b>5.24 -</b>
	2	0	0	248.00	214.00	59.00	3	
	3	0	0	165.00	331.00	4.00	3	
	2	0	5	-1.25	-1.33	2.38	1.94	
	2	0	5	0.24	-1.56	3.63	1.94	
	2	0	5	-0.21	-4.23	6.72	1.94	

**Leitura dos dados (3 neurônios)**

Para representar o corpo celular (ou soma) no visualizador utiliza-se um objeto do GLScene denominado *sphere*<sup>4</sup>.

## 29. VISUALIZAÇÃO

A visualização da estrutura é feita pelo processamento dos dados recebidos e formatados em uma estrutura de dados, que representa a camada lógica da rede neural. As entidades que a compõem (neurônios, galhos e ramos) ganharam uma representação visual por meio dos elementos da API do OpenGL.

Cada mudança na estrutura lógica implica na atualização de sua apresentação, a fim de que seja garantido sincronismo entre o que é mostrado e os dados recebidos.

A cena é criada de acordo com o recebimento dos dados enviados pela simulação. Quando há ramificação para um novo galho (bifurcação), um novo *Pipe* é gerado dinamicamente (Figura 5.25), que possuirá novas coordenadas, gerando, assim, a visualização do crescimento neural. Note que em ramificações, apenas um *pipe* novo é gerado e não dois *pipes*, uma vez que o segundo *pipe* (o outro lado da bifurcação) poderá ser a continuação do *pipe* anterior. Isto é feito para minimizar a criação de novos objetos na visualização, o que resulta na otimização da renderização.

<sup>4</sup> Sphere: objeto que gera uma esfera em um lugar determinado da cena.

Neste  
dispon

**30. Opç**

```
ind := 'galho' + inttostr(grupo) + inttostr(numero); // nome do novo galho
temp := TPipe.Create(GLSceneViewer1); // gera um novo objeto na cena
temp.Parent := DummyCube1; // indica a quem pertence o objeto
temp.Name := ind; // define o nome do novo objeto
```

usuário,

Seguem descritas todas as opções disponíveis para o usuário.

### **31. Arquivo**

Neste item do menu o usuário encontrará as seguintes opções: Abrir, Anexar, Gravar e Sair

#### ***Abrir***

Nesta opção, o usuário pode escolher um determinado arquivo-texto definido pela extensão .txt ou .dat, que esteja gravado em sua máquina local.

Após a seleção do arquivo, é montada a estrutura em árvore com base nos dados contidos no arquivo e a cena no ambiente de realidade virtual é montada.

#### ***Anexar***

Nesta opção pode-se dar continuidade a uma estrutura já existente no visualizador, ou seja, um arquivo inicialmente é lido e posteriormente são acrescentadas novas ramificações ou mesmo novos neurônios a esta estrutura.

É importante salientar que no arquivo anexado não poderão conter dados ambíguos, ou seja, dados que já foram plotados na tela não poderão ser lidos novamente, o que causaria um erro e por consequência um travamento no sistema, pois a estrutura em árvore existente permite crescimento e não alteração parcial de alguns dados já existentes.

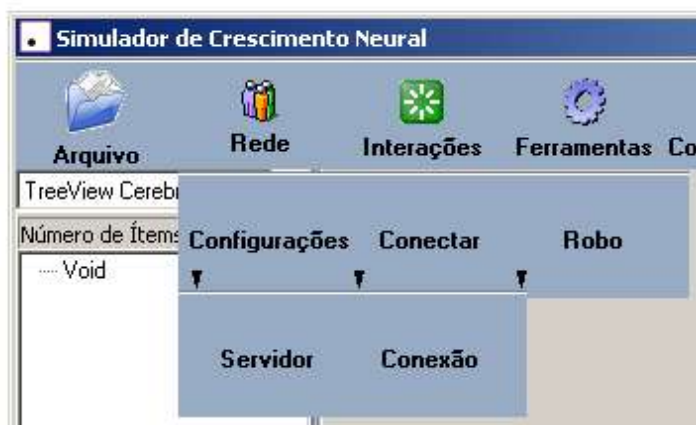
#### ***Gravar***

Nesta opção o usuário tem a oportunidade de gravar a cena de três formas distintas: como pontos, como imagem ou como cena. Gravando os pontos ele terá em um arquivo texto com toda a estrutura mostrada na tela, como foi formada, como se fosse um arquivo a ser lido pela opção Abrir que será armazenado na sua máquina. Assim seria possível recarregar a mesma

estrutura novamente. Ao escolher a opção Gravar Imagem, o usuário pode, com o auxílio do *mouse*, mover a cena, utilizar um *zoom* ou algumas ferramentas disponíveis no software e obter uma imagem (*bitmap*) da cena gerada em 2D. Pode-se aplicar algumas interações e gravar uma nova imagem, podendo assim realizar, por exemplo, uma comparação entre as imagens gravadas. Esta comparação não é possível ser visualizada no software desenvolvido. Gravando a cena, o usuário terá acesso a todos os componentes utilizados e suas principais propriedades contidas na cena gerada mesmo não visuais, como por exemplo, a câmera e suas principais propriedades, tais como campo de visão (*DepthOfView*), o foco (*FocalLength*), direção (*Direction.Coordinates*), estilo (*CameraStyle*), posicionamento (*Left/Top*), entre outras.

### 32. Rede

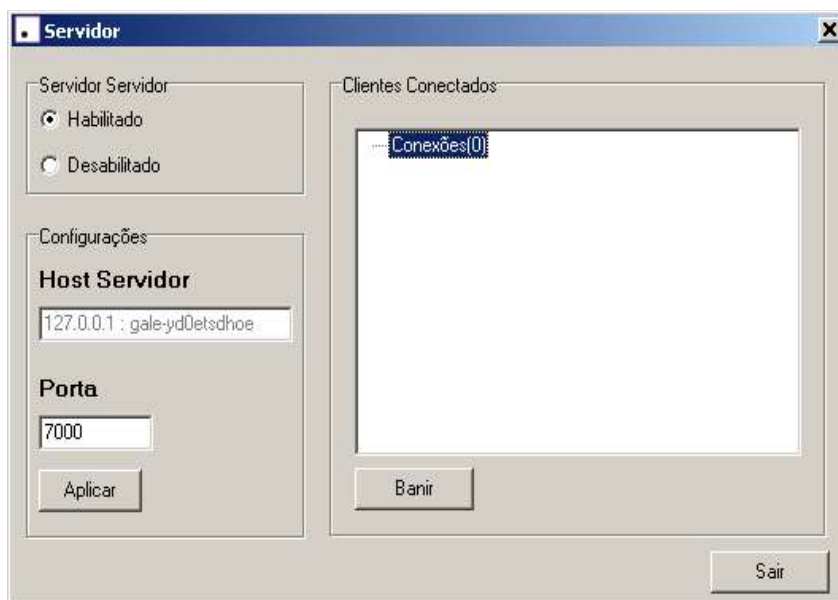
Uma das opções deste projeto é o usuário trabalhar em rede, possibilitando ao simulador enviar e receber dados da rede, utilizando *sockets*, sendo que ele sempre estará enviando e recebendo informações do controlador da cena. Uma proposta futura seria a implementação de diversos usuários interagindo com a mesma cena, sendo neste caso, a comunicação em rede de extrema importância. Neste item o usuário poderá contar com três novos subitens: configurações, conectar e robô, como ilustra a Figura 5.26.



**Figura 5.26 -  
Opção rede**

Na opção Configurações, o usuário poderá configurar tanto o servidor como a conexão. Conforme ilustra a Figura 5.27, o usuário pode Habilitar ou Desabilitar o servidor, e principalmente identificar o endereço para comunicação com o cliente juntamente com a porta de comunicação que deve ser habilitada. Como podemos perceber na figura, o servidor está ativo, com o

endereço 127.0.0.1 e utilizando a porta 7000 para comunicação. Ao lado, há possibilidade de mostrar os endereços dos clientes que estão ativos no momento, item provavelmente que fará parte de um projeto futuro.



**Figura 5.27 -**

### **Configuração Servidor**

Ainda no item Configurações tem-se a oportunidade de configurar a conexão, fazendo o papel do cliente na rede. Mesmo que haja um único computador conectado a rede, o software permite que ele seja tanto servidor como cliente, possibilitando enviar dados pela rede dele para ele mesmo. Desenvolveu-se esta possibilidade para implementar o módulo Robô descrito logo abaixo.

Outro item interessante ainda nesta opção de rede é o módulo denominado Robô, que envia e recebe dados dele mesmo, utilizando para isso a mesma máquina e conseqüentemente o mesmo endereço. Outra particularidade é a possibilidade de abrir um arquivo texto e montar a cena com todos os dados desse arquivo ou apenas com uma seqüência de dados selecionada pelo próprio usuário, como ilustra a Figura 5.28.

### 33. Interação

Este item é considerado como uma c

1	0	0	0.00	0.00	0.00	3
2	0	0	68.00	165.00	143.00	3
3	0	0	241.00	322.00	64.00	3
4	0	0	346.00	126.00	49.00	3
5	0	0	311.00	241.00	98.00	3
1	0	3	0.66	-0.78	2.82	13.11
1	0	3	4.71	3.07	1.97	13.11
1	0	3	3.91	3.01	0.22	13.11
1	0	3	3.41	2.98	-2.49	13.11
1	0	3	-1.37	4.67	-4.52	13.11
1	0	3	-3.61	5.47	-5.63	13.11
1	0	3	-5.31	6.66	-2.87	13.11
1	0	3	-5.36	7.45	-0.40	13.11
1	0	3	-4.91	7.36	1.14	13.11
1	0	3	-3.79	6.56	2.43	13.11
1	0	3	-3.21	7.94	4.80	13.11
1	0	3	-1.69	9.65	8.43	13.11
1	0	3	-1.46	12.17	9.15	13.11
5	0	19	2.51	1.58	0.43	2.54
5	0	19	5.25	4.09	1.83	2.54
5	0	19	7.99	5.67	4.12	2.54

nsiderado onde são

encontradas as transformações geométricas, e a inserção de campos tróficos e impulsos, como ilustra a Figura 5.14.



Encerrar a cena. Para controlar o zoom, existem três botões para controlar o zoom, sendo três delas exemplificadas abaixo, e a última é a barra de rolagem vertical no canto direito do visualizador (vide Figura 5.6) que também tem a função de aproximação ou distanciamento da cena.

#### **Zoom In**

Nesta opção o usuário pode aproximar a cena toda visualizada com uma escala pré-definida de 0.1 cada vez que pressiona o botão. Não há um limite pré-determinado para esta opção. A medida que a cena é aproximada, automaticamente a barra de rolagem vertical também é atualizada.

#### **Zoom Out**

Nesta opção o usuário pode promover um distanciamento da cena toda visualizada com uma escala pré-definida de 0.1 cada vez que pressiona o

botão. Não há um limite pré-determinado para esta opção. A medida que a cena é distanciada, automaticamente a barra de rolagem vertical também é atualizada.

### **Zoom Auto**

Definiu-se uma determinada alteração de escala para esta opção: inicializa-se a escala da cena com o valor 0.5, variando até o valor 2 com um incremento de 0.1. Quando esta opção é selecionada a cena é aproximada e quando ela é selecionada novamente a cena volta ao estado original.

### **Inserir**

Nesta opção, com a utilização do *mouse*, o usuário pode ir até a janela correspondente à árvore no canto esquerdo do projeto (*treeview*), selecionar um galho ou ramo qualquer. Após esta seleção, é identificado o soma (corpo celular) correspondente a este neurônio e são simulados o envio de impulsos eletrônicos representados por esferas percorrendo todo o neurônio, passando por todas as ramificações. À medida em que o impulso vai se distanciando do soma, ele vai enfraquecendo, sendo simbolizado por um grau maior de transparência quando se aproxima do seu final do galho. Quando finalizar o percurso, os impulsos eletrônicos são eliminados. O impulso representa uma transmissão de informação de um neurônio para outro neurônio, que posteriormente poderia ser implementado pelo simulador. Escolhendo o outro item, campo trófico, que pode ser definido como um conjunto de um ou mais campos que provocam uma reorganização na estrutura neural de acordo com a sua intensidade e tipo (repulsão / atração).

A fórmula utilizada para o cálculo do(s) campo(s) trófico(s) foi:

$$F = k * L * v / (d^2) \quad (\text{Eq. 5.1})$$

sendo:

- $k \rightarrow$  velocidade de crescimento (quanto mais negativo, mais rápido será o crescimento e se for positivo haverá repulsão).
- $L \rightarrow$  distância do ponto P ao soma, sendo P um ponto do dendrito
- $v \rightarrow$  cada vetor formado entre P e cada ponto de atração



- $d \rightarrow$  distância do ponto P a cada ponto de atração

### Configuração do Campo Trófico

, o usuário pode aplicar um ou mais campo(s) necessário definir: a posição do campo (ponto campo trófico é um campo de repulsão ou de atral da constante “k” citada acima; o valor do k tra a Figura 5.31.

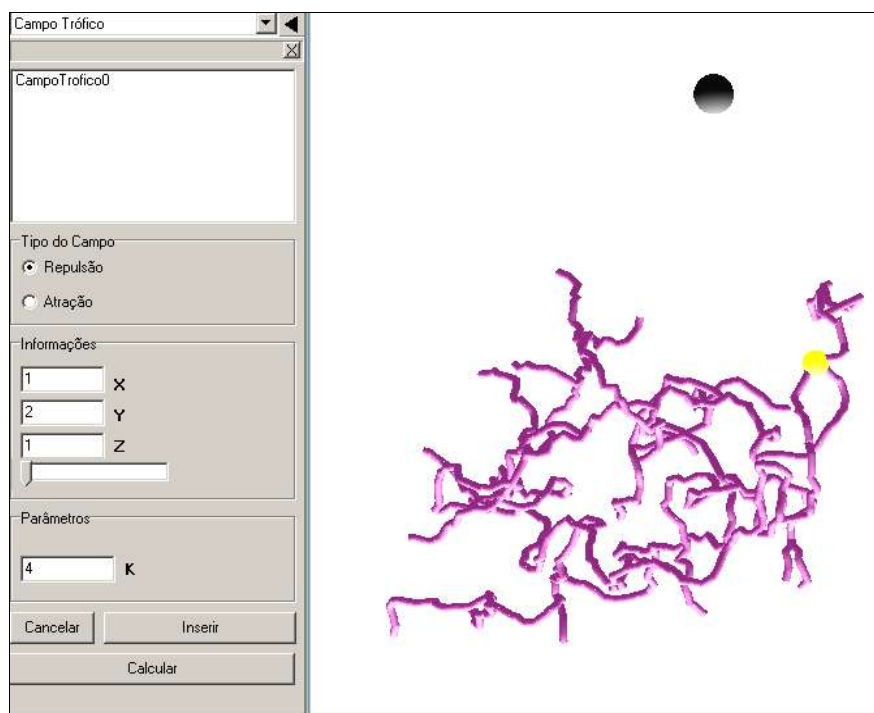
cancelar o campo. Em seguida, pode-se inserir ento efetuando o mesmo processo.

pressionar o botão “calcular” que possibilitará a reorganização da estrutura neural.

O ambiente propicia também ao usuário escolher se deseja visualizar a estrutura atual sem os cálculos dos campos tróficos (utilizando para isso uma transparência) junto com a estrutura redirecionada; ou apenas visualizar a nova estrutura neural reorganizada. É importante salientar que os campos tróficos, após realizados os cálculos, são deixados transparentes na cena, para que se possa observar a alteração.

É interessante salientar também que o “L” utilizado no cálculo é referente ao comprimento do galho a partir do soma até o ponto que está sendo redirecionado e não a distância. Após calculada a força, é calculado o vetor que será utilizado para redirecionar o galho.

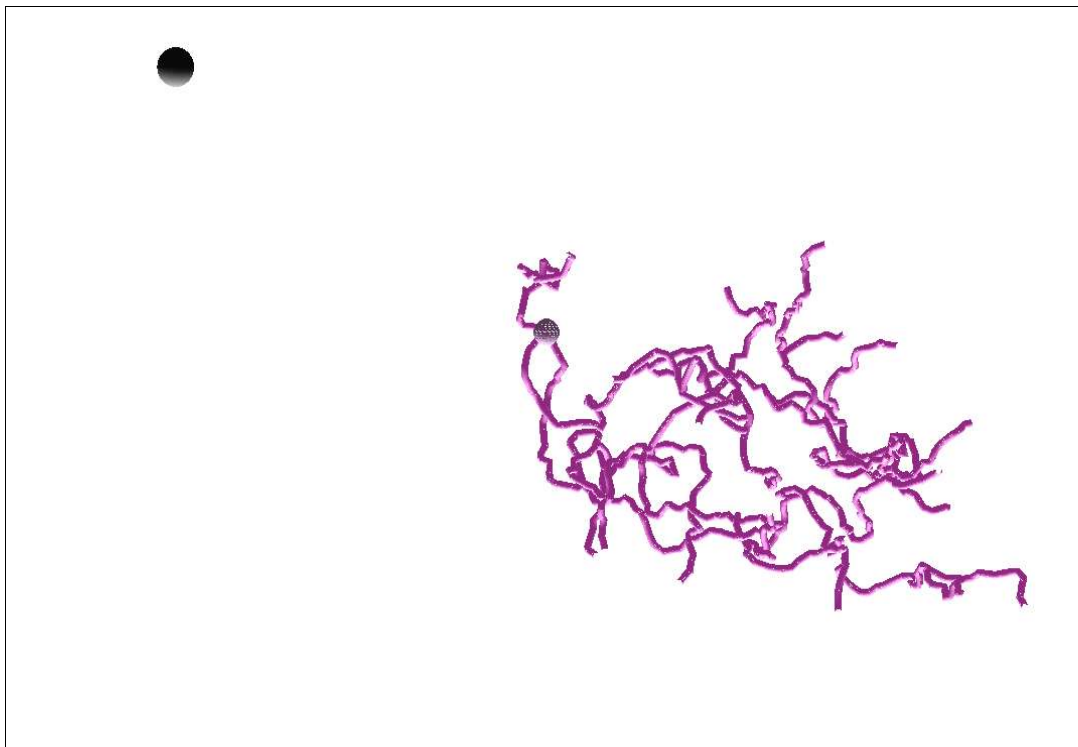
Segue na Figura 5.31 um exemplo de execução de um campo trófico de atração e o seu resultado. Para efetuar os cálculos foram utilizados os valores de posição do campo ilustrados na Figura 5.30.



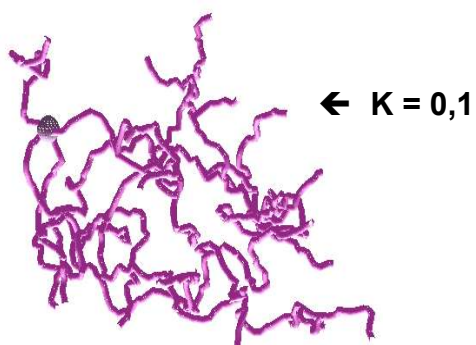
**Figura 5.31 - Inserção e Cálculo do Campo Trófico**

Pode-se observar na Figura 5.31, a presença de um campo trófico inserido na cena e na Figura 5.32, uma seqüência dos efeitos deste campo trófico na mesma cena, atualizada. Neste caso observam-se as duas cenas isoladas, mas podem-se observar as duas cenas juntas. Para realizar tal evento utiliza-se a memória desenvolvida no software e colocam-se cores distintas em cada imagem gerada.

**Figura 5.32 - Cena original com a presença do Campo Trófico**



Nas figuras que seguem foi utilizada sempre a mesma figura original 5.17, com a mesma posição do campo trófico, apenas alterando o valor da constante  $K$ , e determinando se este campo é de atração ou repulsão. Os resultados são mostrados nas Figuras 5.18 e 5.19.



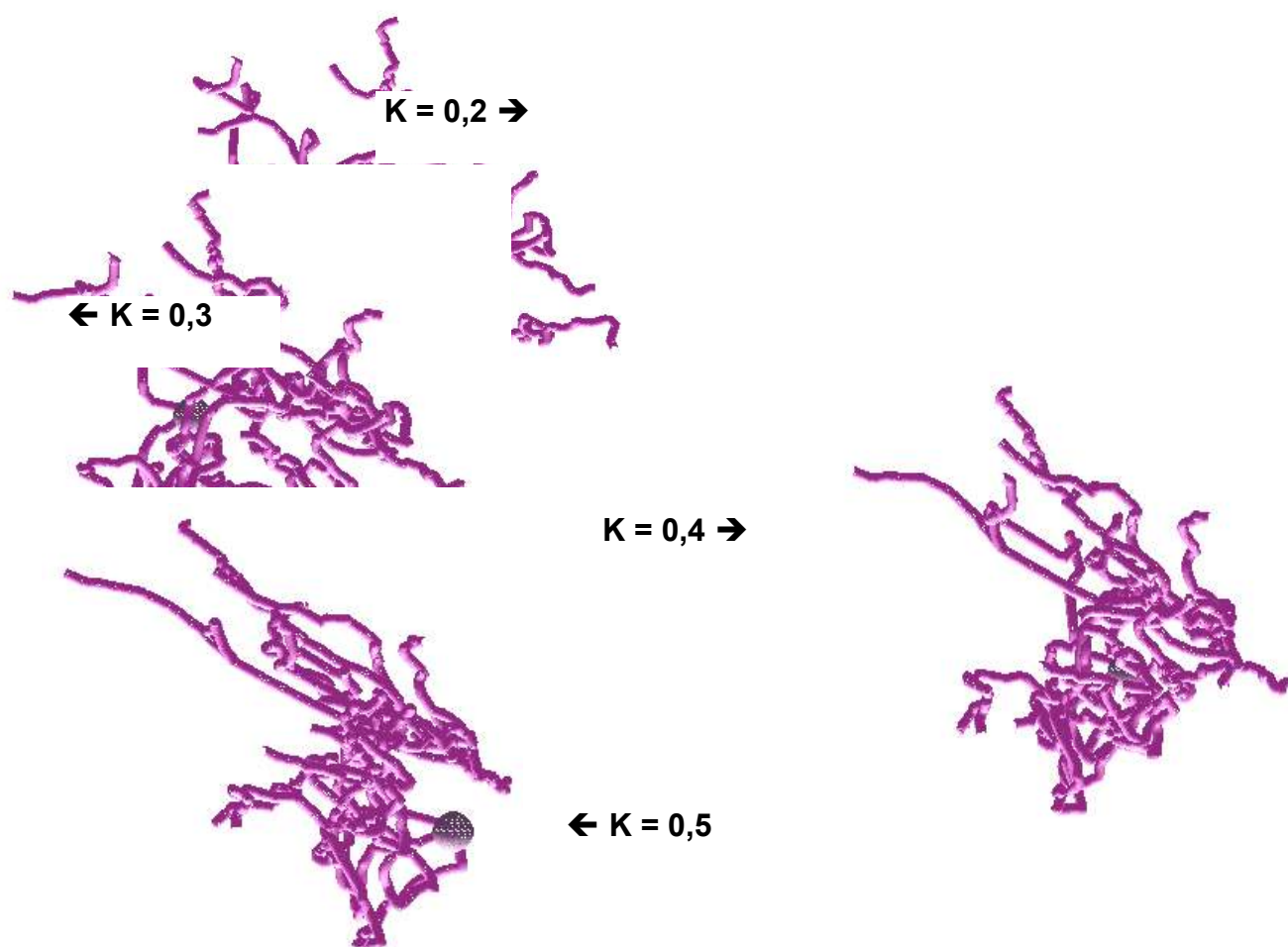
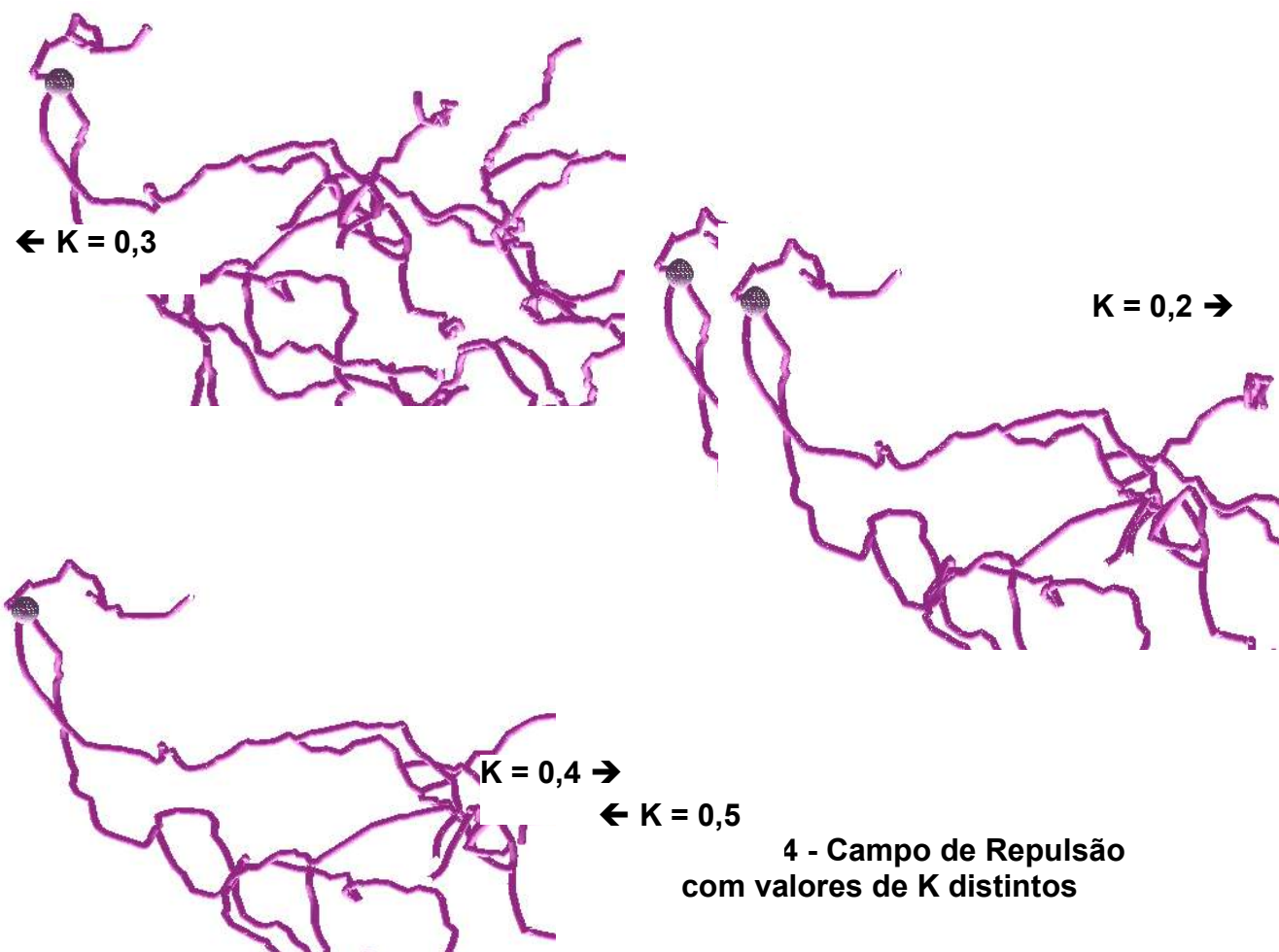


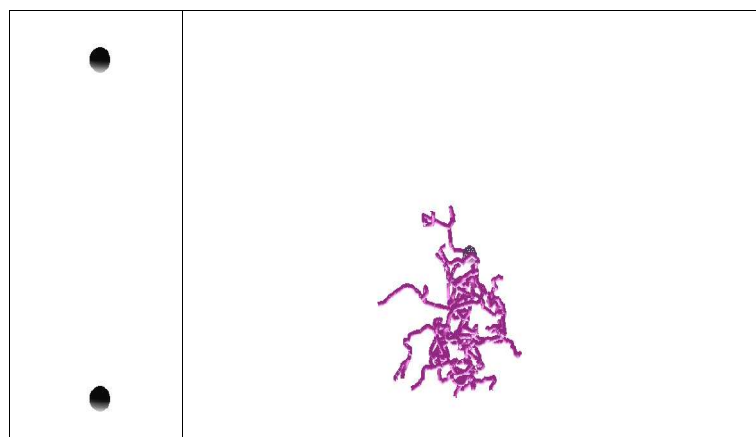
Figura 5.33 - Campo de Atração com valores de K distintos



Podemos ter mais de um campo trófico na mesma cena agindo sobre a mesma célula, o que é ilustrado na figura 5.20 (a), utilizando dois campos tróficos de atração. Na Figura 5.35 (b) pode-se observar a ação desses dois campos definidos como campos de atração utilizando  $k$  com valor de 0,2. A Figura 5.20 (c) ilustra as duas cenas (a) e (b) juntas, mostrando a célula original e a nova célula modificada.

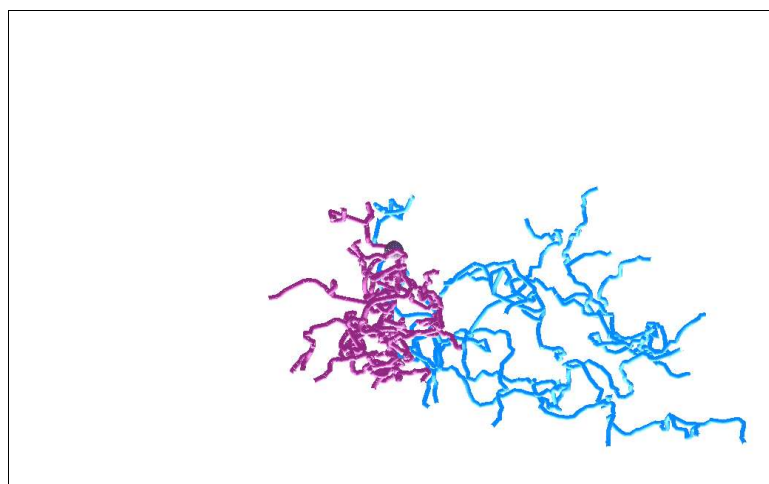
(a)

(b)



(c)

Figura

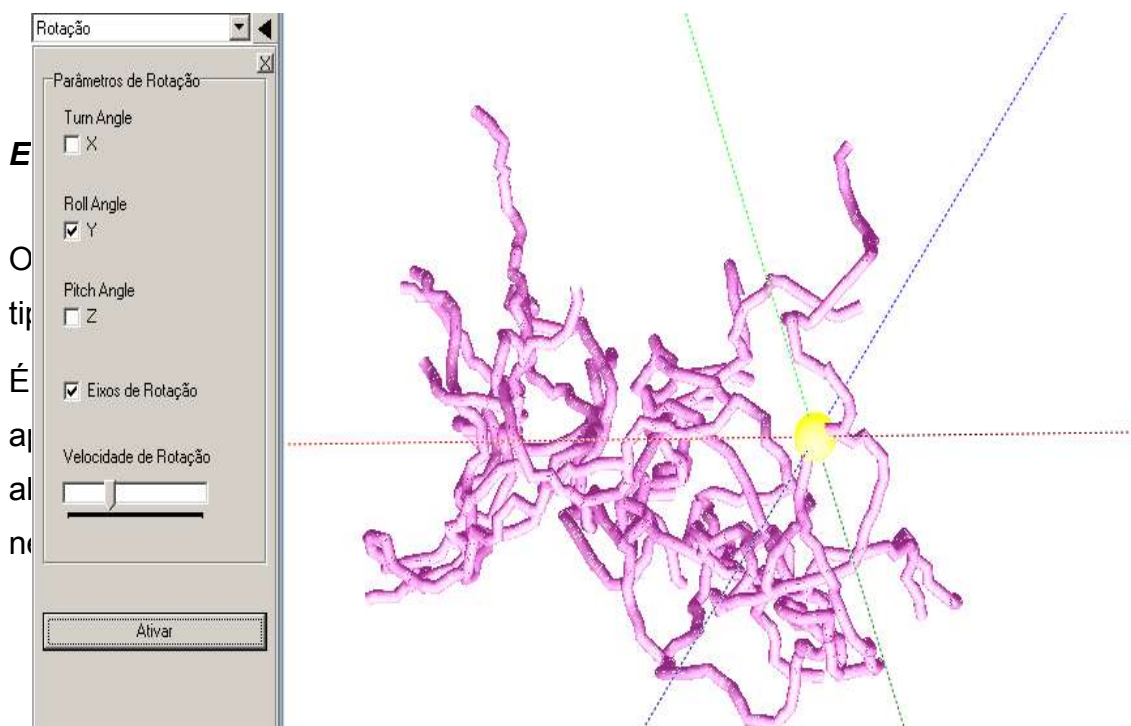


Ação de mais de um Campo de Atração

5.35 -

### Rotação

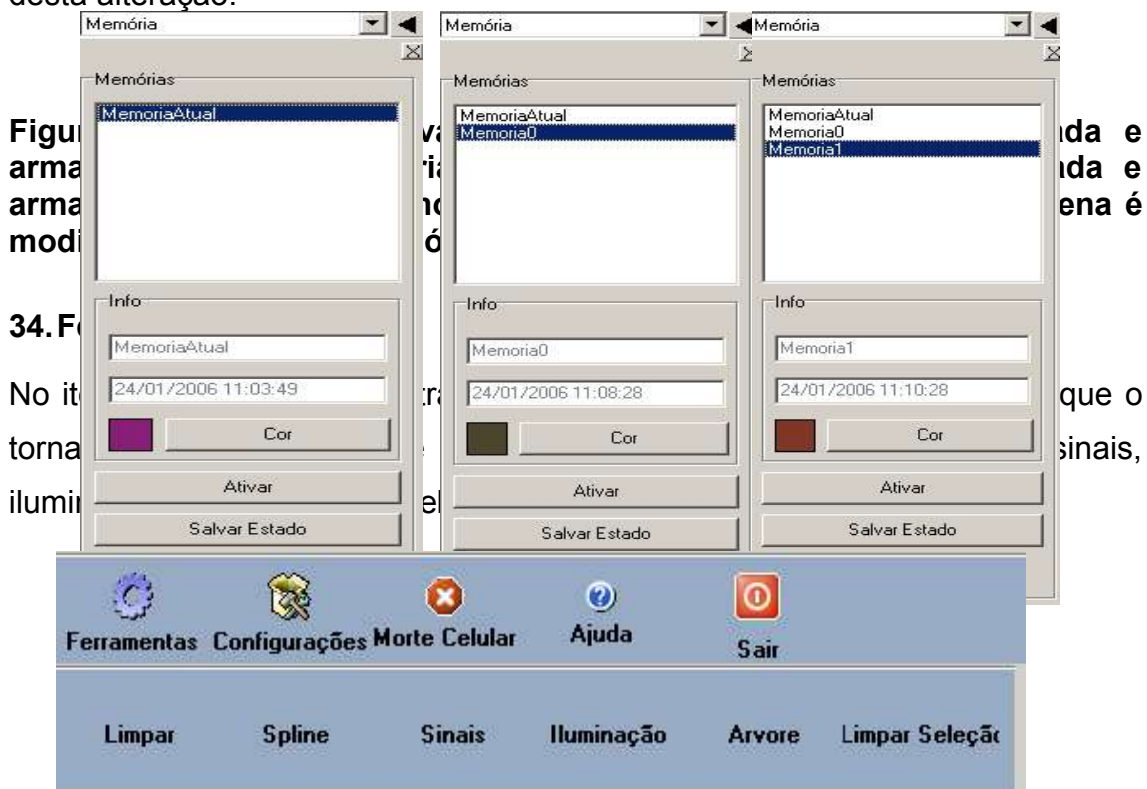
Conforme ilustra a Figura 5.21, nesta opção, tem-se a oportunidade de escolher um ou mais eixos nos quais a cena será rotacionada. Além de escolher os eixos, pode-se também determinar a velocidade com que esta rotação é realizada, desde um valor mínimo definido pelo software até um valor máximo estipulado pelo mesmo. Os ângulos variam entre zero e 360 graus. Tem-se a oportunidade de escolher a visualização dos eixos na cena ou não, e por fim, existe ainda um botão “Ativar” que a qualquer momento pode ser utilizado para iniciar ou parar a rotação automática (contínua) utilizando o ângulo definido na velocidade a cada rotação.



realizada uma normalização dos dados obtidos que tornam a cena mais compacta e posicionada a câmera com base nesses valores encontrados em uma determinada direção, para que a cena possa ser visualizada de maneira clara e centrada no primeiro neurônio.

### Salva Estado

Quando o cena é gerada, todos os seus dados são gravados em memória, como ilustra a Figura 5.37. Existe a possibilidade de se alterar a cena modificando tamanho, rotacionando, realizando uma eliminação de alguns neurônios e outras interatividades. A seguir é possível salvar a nova cena na memória, de modo que o usuário possa restaurá-la posteriormente, porém antes da finalização do programa, caso queira estudar ou realizar novas interações. Essa opção é fundamental quando realizamos a inserção de um campo trófico, em que os dados modificados são armazenados na memória, de maneira que pode-se visualizar a estrutura anterior e a nova estrutura calculada. Cabe ao usuário, definir quando gravar a nova cena gerada, ficando limitado apenas ao tamanho da memória do seu computador. Na Figura 5.37 pode-se observar o armazenamento das cenas, a data/hora das cenas gravadas e a possibilidade de escolher uma cor para que a cena seja mostrada na tela. Esta última opção é importante quando é necessário, por exemplo, comparar uma estrutura neural redirecionada e esta mesma estrutura antes desta alteração.



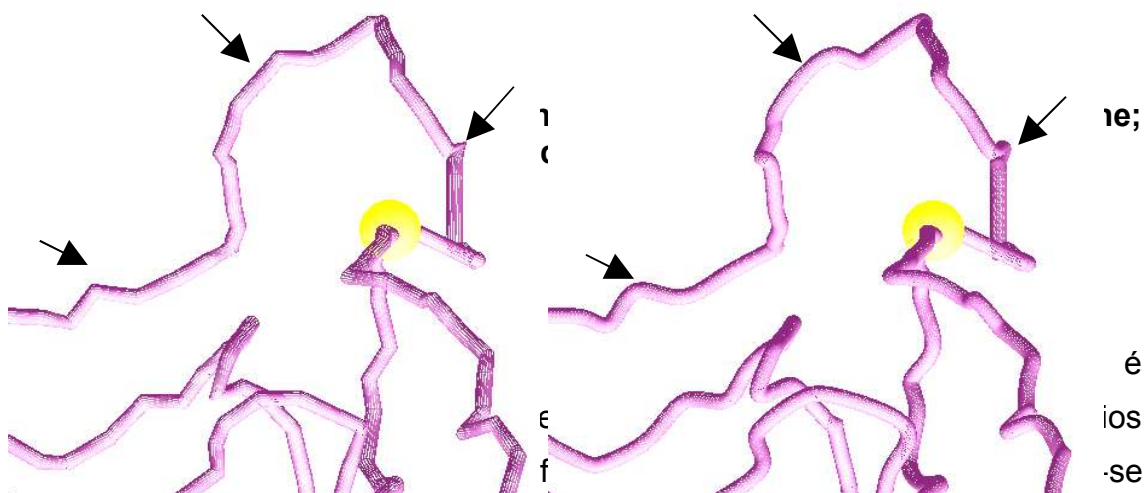


### ***Limpar***

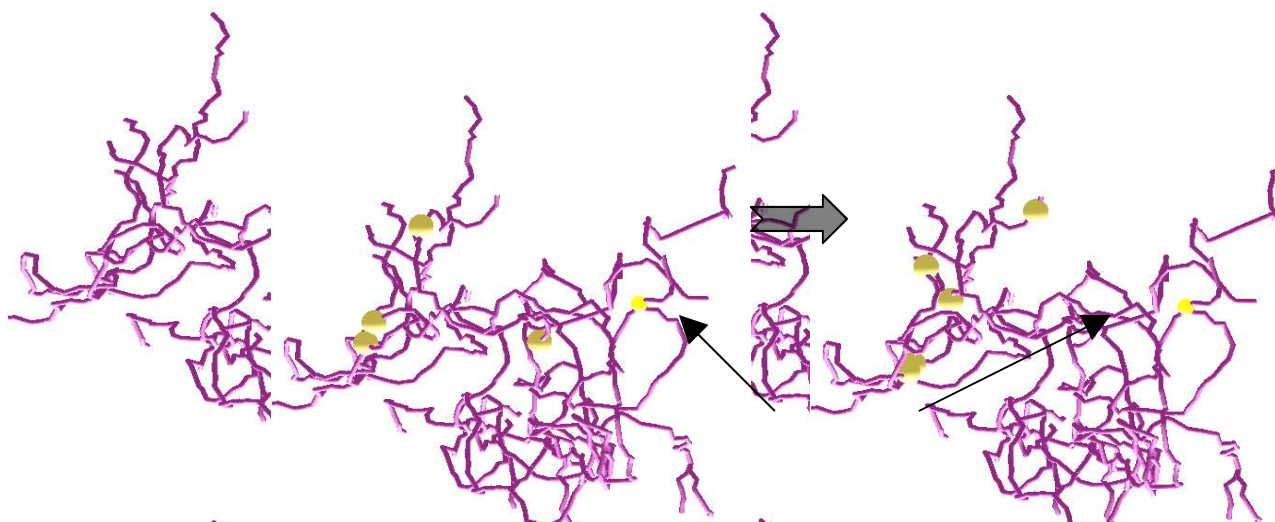
Simple e eficaz, esta opção permite limpar e eliminar a cena gerada juntamente com a árvore gerada, ou seja, toda a estrutura da árvore criada na memória do computador é eliminada juntamente com a cena visualizada.

### ***Spline***

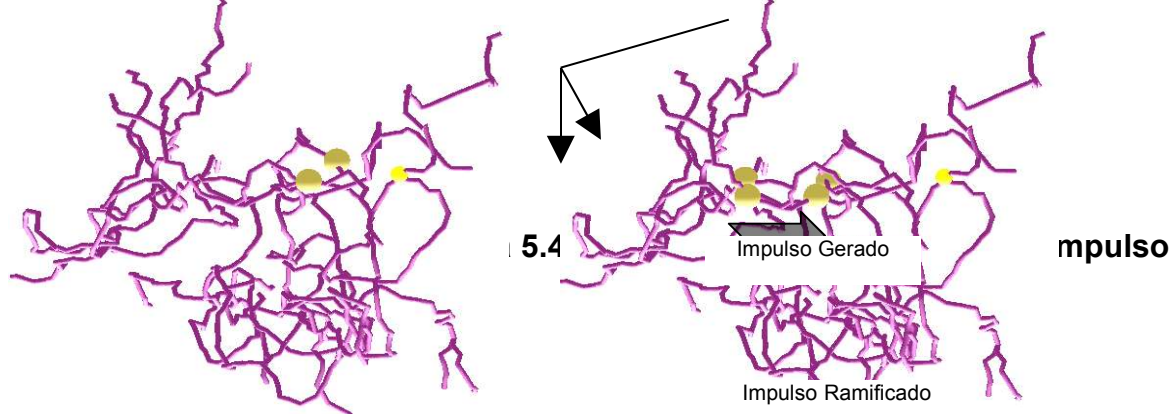
No software a opção *spline* realiza suavização em uma reta, como ilustra a Figura 5.39. Para facilitar ainda mais o entendimento foi destacado alguns pontos visualmente mais interessantes para observar tal ferramenta. Para habilitar/desabilitar esta ferramenta basta clicar sobre a opção do menu



um galho e com o botão direito do *mouse*, insere-se o impulso. Este impulso irá percorrer todo o neurônio desde o soma até o final de todas as ramificações, inclusive gerando outros impulsos a medida que o *pipe* é bifurcado. A Figura 5.40 ilustra este impulso sendo gerado. Nesta opção, pode-se simular o caminho de uma corrente elétrica no neurônio.



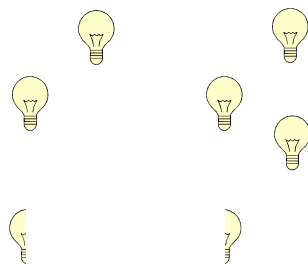




## Iluminação

A iluminação utilizada é baseada no modelo de iluminação de Phong. A cada vértice da primitiva, uma cor é calculada usando as propriedades dos materiais junto com suas propriedades de luzes. A cor para o vértice é computada pela soma das quatro cores calculadas para a cor final do vértice. As quatro componentes que contribuem para a cor dos vértice são: Iluminação Ambiente, componente Difusa, Luz Especular e emissiva.

No software desenvolvido, a iluminação é gerada dentro de um cubo denominado Dummy Cube (GLSCENE, 2004), e em cada aresta deste cubo fora colocado um ponto de luz; conseqüentemente tem-se 8 pontos de luz, como ilustra a Figura 5.41. Para isso, foi utilizado o objeto GLLight do GLScene, que permite a determinação do tipo de iluminação a ser utilizada, seja ela paralela, spot ou omni.



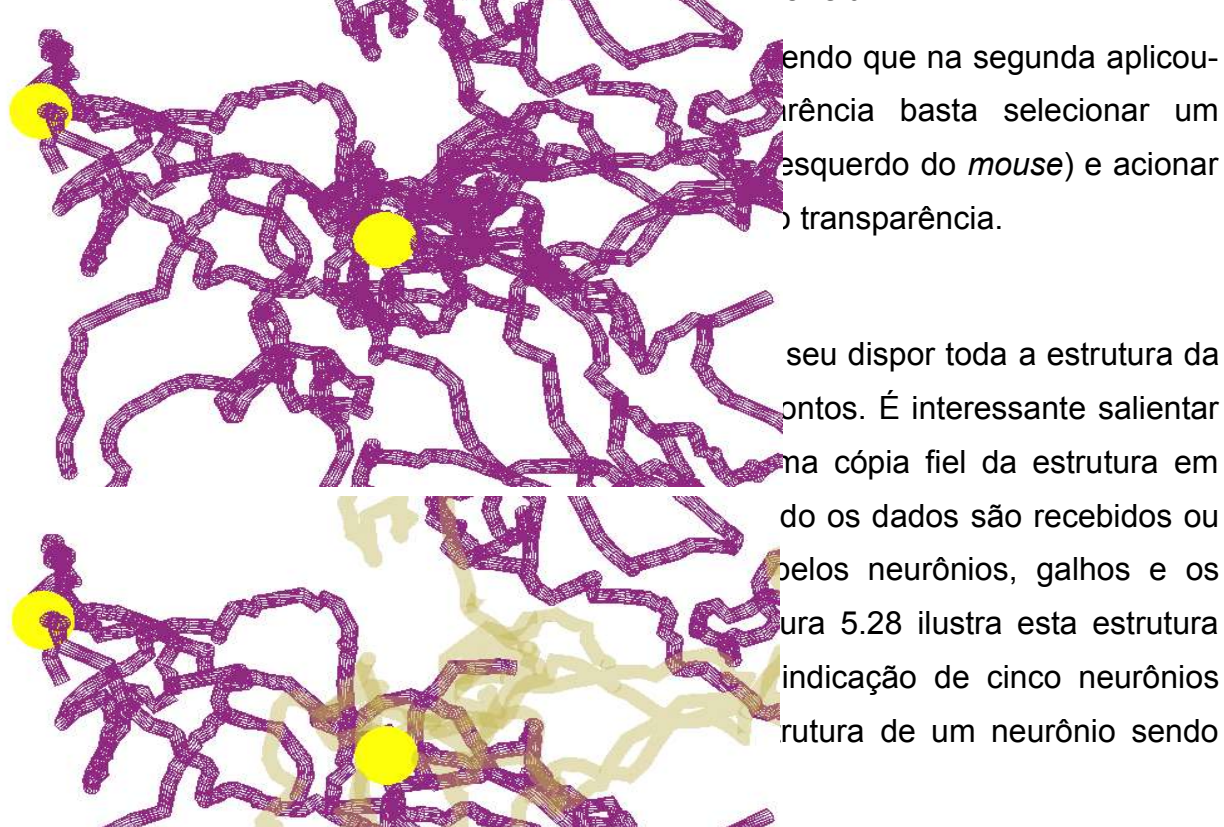
**Figura 5.41 - Pontos de Luz**

A *Omni* é uma luz que irradia seus raios em todas as direções. Já o *Spot* produz um cone de iluminação, a partir de um ponto de origem (semelhante a uma luz de palco) e a paralela, também conhecida como *Target Direct*, é semelhante ao *Spot*, que também produz um cone de iluminação, mas diferentemente do *Spot*, o cone da paralela adquire a forma de um cilindro. Isto porque simula uma fonte de luz no infinito e, conseqüentemente, seus raios de luz são paralelos (WOO, 1999).

Em algumas partes do software utiliza-se a transparência, denominada *Blending* (WOO, 1999). Os efeitos de transparência são acionados utilizando

uma propriedade do componente *GLLight* denominada alfa, variando de 0 a 1, sendo 0 totalmente transparente e 1 significa sem transparência.

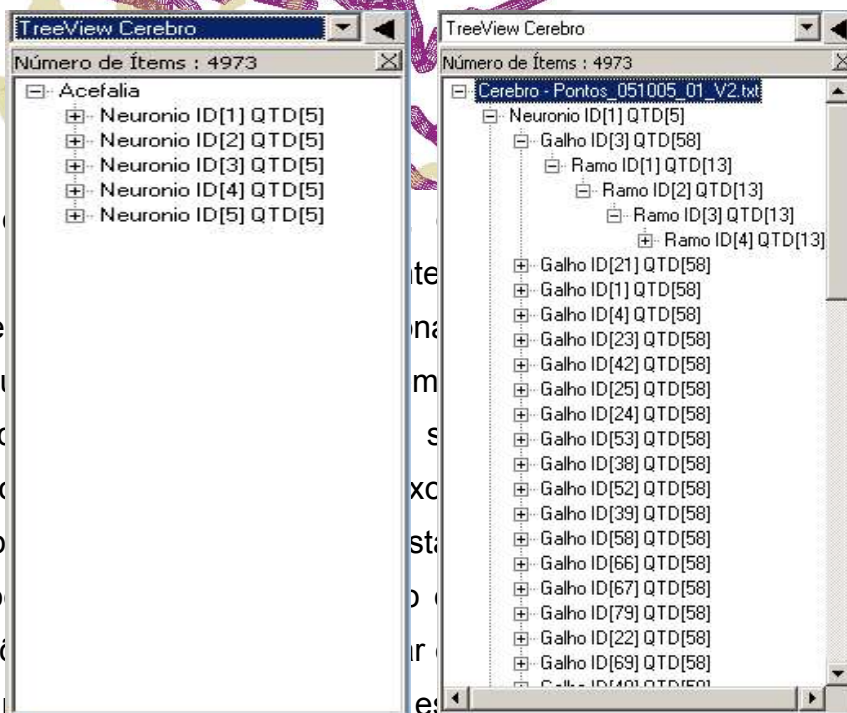
**Figura 5.12 - Transparência**



Quando se aplica transparência basta selecionar um ponto (clicando com o botão esquerdo do mouse) e acionar a propriedade de transparência.

Quando se dispõe toda a estrutura de pontos. É interessante salientar que na cópia fiel da estrutura em memória os dados são recebidos ou armazenados pelos neurônios, galhos e os ramos. A figura 5.28 ilustra esta estrutura hierárquica de um neurônio sendo

clicando no conteúdo do usuário te soma de trabalho bidimensional CTRL + b as seleções das seleções pressionar



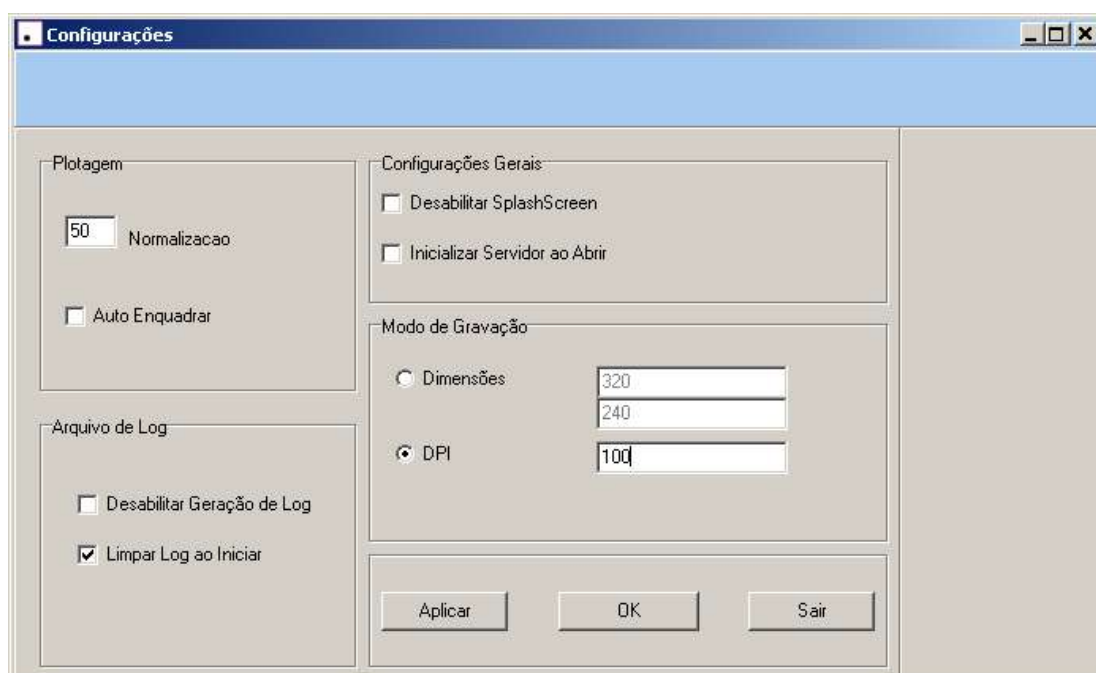
Normalmente flutuante na cena, o mesmo objeto. Mesmo na figura as teclas para manipular esta

seleção foi utilizado uma estrutura computacional denominada de recursividade, ou seja, a medida que um galho é selecionado todos os seus filhos, a partir deste galho são selecionados; conseqüentemente, em alguns

casos os filhos se tornam pais de outros galhos, pertencentes ao mesmo galho selecionado; necessitando assim uma estrutura recursiva.

### 35. Configurações

A opção configurações, como ilustra a Figura 5.29, é importante para selecionar e definir algumas características iniciais e principais do software. Estas configurações, após serem definidas, são armazenadas em um arquivo chamado “NeuroConfig.ini”, transparente ao usuário. Porém toda vez que o software é inicializado, as informações contidas no arquivo “NeuroConfig.ini” são utilizadas para definir alguns parâmetros essenciais ao software. Alguns parâmetros são utilizados a medida em que o software está sendo utilizado, outros, porém, a medida que é inicializado.



momento, seja posicionada de frente com a camera; a camera por sua vez, esta direcionada para o centro do soma e levemente afastada para que possibilite uma melhor visualização.

Na janela “Arquivo de log”, define-se a opção para geração do arquivo de log e ao mesmo tempo, caso este arquivo já exista, ele possa ser zerado. Arquivo de log é um artefato no software que registra todas as ações do software, desde a inicialização até o término. No arquivo de log, por exemplo, registra-se a data/hora que o servidor for acionado, registra-se os dados que são lidos tanto pelo arquivo local como pela rede, indica se houve uma conexão entre o simulador e o ambiente, indica a quantidade de neurônios plotados, quantidade de segmentos entre outras tarefas mais simples do software.

Na janela “Configurações Gerais” tem-se a oportunidade de desabilitar a tela de SplashScreen, que é uma tela de inicialização do software, como ilustra a Figura 5.30; tem-se a oportunidade também de inicializar a SplashScreen a medida em que o software é requisitado. Nesta inicialização, caso positiva, possibilita que uma porta de comunicação seja aberta pelo ambiente de realidade virtual esperando para que haja uma conexão com o simulador. Esta conexão pode ser refeita, ou mesmo a conexão desabilitada durante a execução do software.



**Figura 5.45 - Splash Screen**

No item “Modo de Gravação”, pode-se definir as dimensões da imagem a ser gravada, bem como definir os dpi's (*dot per inch*), pontos por polegadas, que determinará a qualidade da imagem gravada; quanto maior a quantidade de dpi's, melhor a qualidade da imagem, porém maior o tamanho da figura gerada.

### **36. Morte Celular**

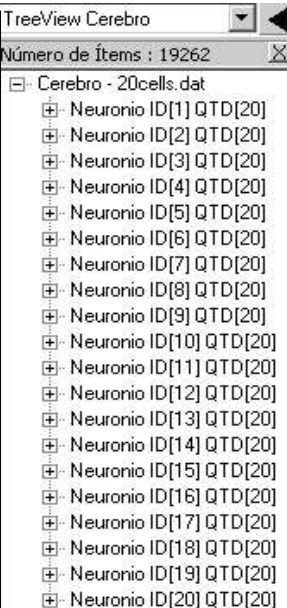
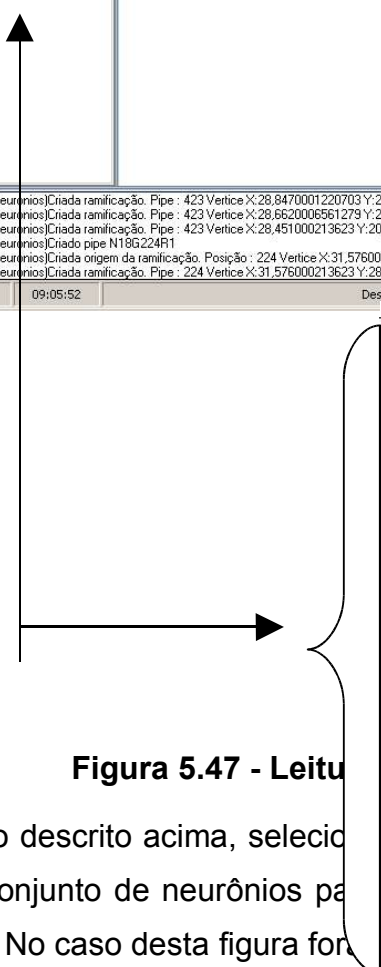
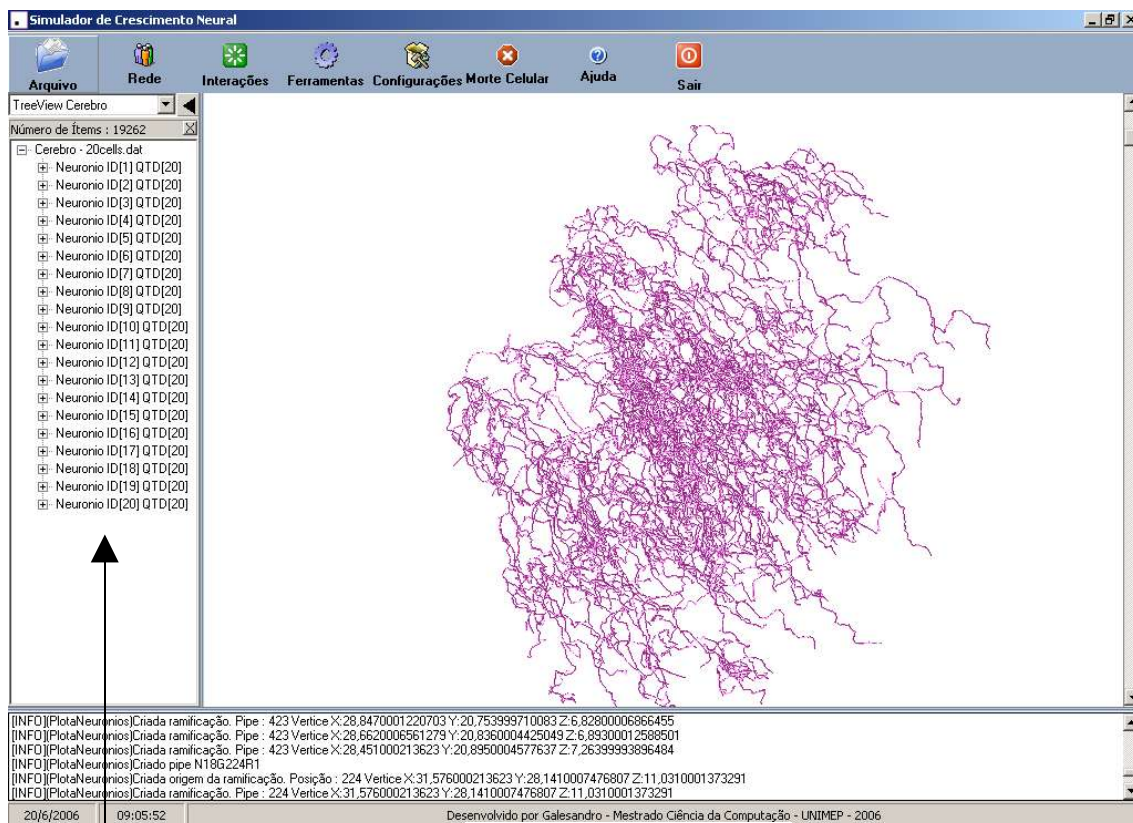
Na opção Morte Celular, permite-se realizar a necrose ou a apoptose, como ilustra a Figura 5.31.



Na necrose, morte celular utilizando a tecla CTRL permite que seja escolhido um determinado neurônio. Feito isso, é possível, com o mouse, um determinado neurônio é eliminado e a estrutura neural é reorganizada e por fim totalmente redeseenhada, excluindo, no caso, o neurônio desejado.



A Figura 5.32 ilustra a imagem gerada após a leitura de um arquivo contendo 20 neurônios. Ao lado da imagem gerada é exibida a *treeview* deste conjunto de dados lidos, bem como a identificação de cada neurônio com as suas respectivas ramificações. Para melhor visualizar, exibimos a *treeview* logo abaixo da figura mostrada com um determinado zoom.

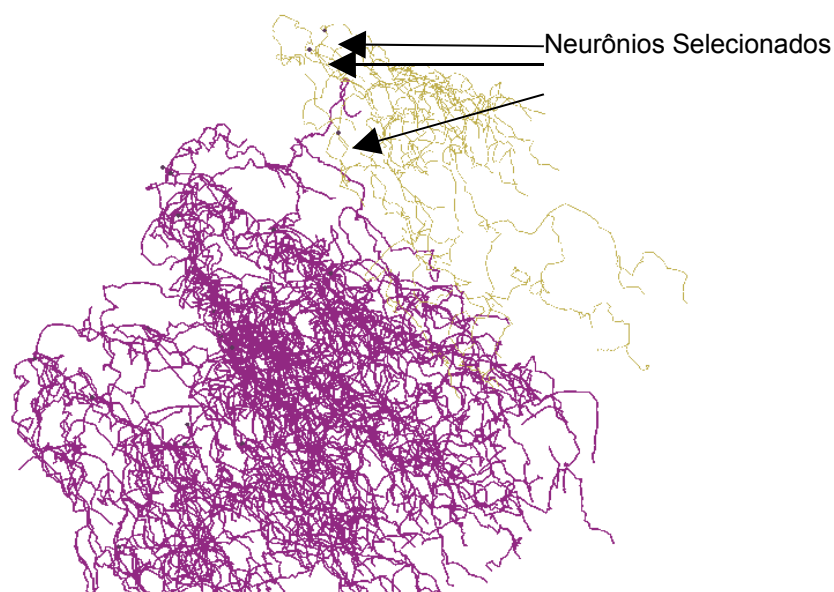


**Figura 5.47 - Leitu**

Como descrito acima, selecio  
um conjunto de neurônios pa  
5.33. No caso desta figura for

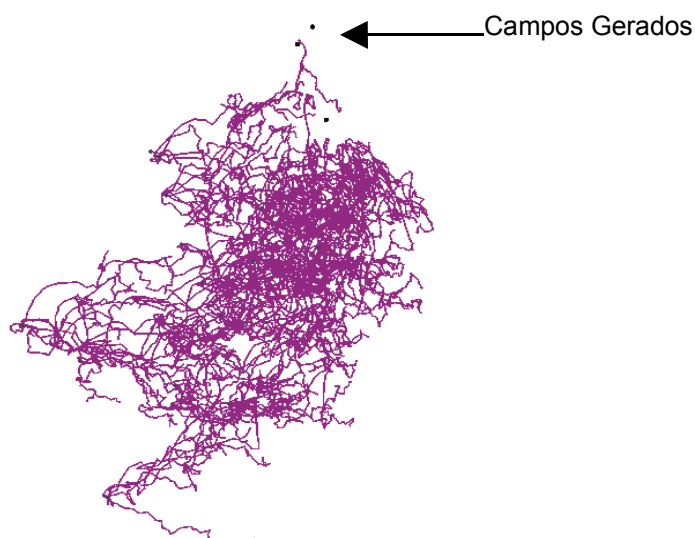
**ução necrose**

gerada um neurônio ou  
se, como ilustra a figura  
nios.



**Figura 5.48 - Neurônios selecionados**

Aplicando-se a Necrose com este conjunto de neurônios selecionados obtém-se a figura 5.34.



**Figura 5.49 - Resultados da Necrose**

Pode-se observar na figura gerada que os neurônios selecionados para Morte Celular somem e geram campos de atração, os quais fazem com que haja um redirecionamento de toda a estrutura neural a fim de suprir a falta daqueles neurônios eliminados.

No caso de apoptose, é transmitido para o simulador a palavra "Apoptose"; nesse caso o simulador também realiza a reorganização neural e transmite os

dados novamente para o ambiente desenvolvido que plota a cena com esta alteração realizada. Na apoptose, as células com menor número de conexões é que são escolhidas para morrer.

### 37. Ajuda

Nesta opção o usuário tem acesso a todos os objetos do sistema, bem como suas classes e propriedades, descrevendo como se navega nos diversos itens do menu do projeto. Dentro ainda desta ajuda encontra-se a definição de todos os itens dos menus, em alguns casos com exemplos, outros com definições. Nesta opção “ajuda”, ainda encontra-se um formulário denominado “sobre” que especifica o nome do trabalho, do desenvolvedor e a versão do software, como ilustra a figura 5.35.

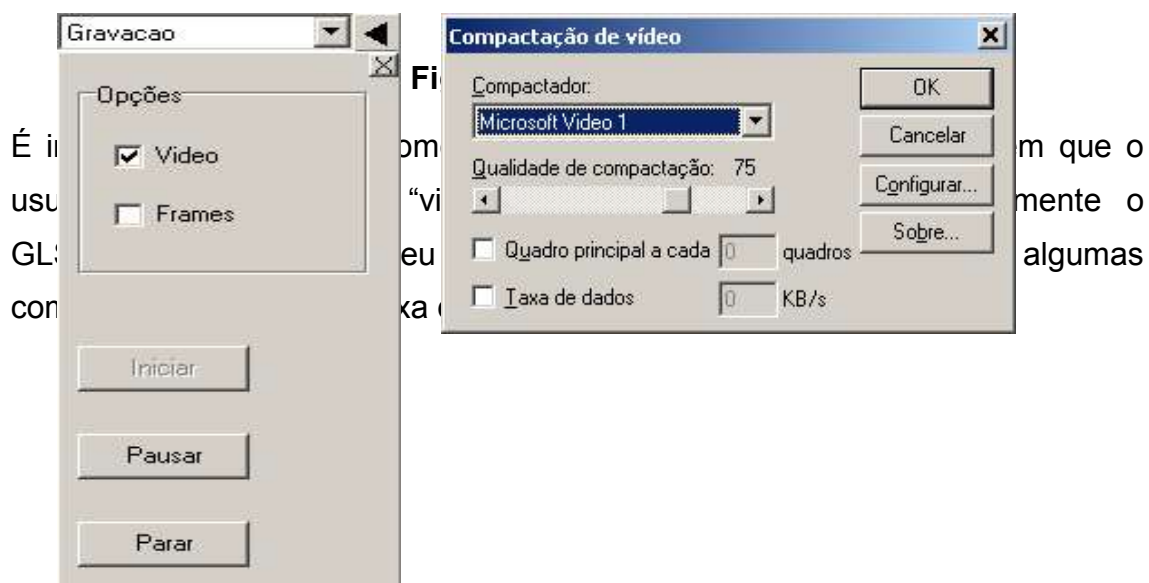


Na janela de ferramentas, como ilustra a figura 5.36, para a medida que seleciona algum objeto, o usuário seleciona o botão CTRL + botão esquerdo do mouse para selecionar os ramos selecionados, logando a quantidade de elementos selecionados. Cada neurônio como sendo: N (neurônio) + G (galho) + R (ramo) + No. Ramo, como ilustra a figura 5.36. Cada neurônio possui vários galhos saindo do soma. Cada galho possui vários ramos por causa das ramificações que ocorrem durante o crescimento.



Encontra-se disponível as ferramentas a opção gravação, podendo o usuário definir se deseja gravar um vídeo com o que está acontecendo na cena, suas alterações de visões, acessos ao menu, *zooms*

entre outras atividades e armazená-lo em um arquivo .avi, ou se deseja salvar em forma de *frames*, ou melhor, quadros com cenas diferentes.





### **38. CONCLUSÃO E TRABALHOS FUTUROS**

Neste trabalho foi apresentado um ambiente gráfico que permite a visualização e interação do usuário com o ambiente em tempo de simulação do crescimento de estruturas neurais. Além de interações relativas à navegação, foram incluídas diversas outras que possibilitam a modificação da estrutura resultante, como por exemplo, morte de algumas células e a inclusão de campos tróficos, sendo que ambos podem provocar uma reorganização das arborizações dendríticas. Algumas outras possibilidades de interações incluídas no ambiente são: tingimento de partes das células ou de toda a célula, transparências de partes ou de toda a célula, inclusão de três tipos diferentes de fontes de luz, etc. Além disso, é permitido também gravar todas as interações e alterações realizadas pelo usuário em forma de vídeo para que possam ser analisadas futuramente.

A possibilidade de alterar vários parâmetros iniciais de configuração do ambiente oferece maior flexibilidade para o uso desta ferramenta da maneira mais conveniente ao usuário.

Com a conclusão deste trabalho, é possível verificar que, se a Realidade Virtual simular situações suficientemente próximas ao real, o seu uso, como um artefato de apoio à construção do conhecimento e da aprendizagem, é possível. Entre as suas inúmeras vantagens, a ilimitada capacidade de exercício proporcionada pela Realidade Virtual nos parece a mais importante. Em algumas situações, quanto mais exercitar, mais habilitado estará o aprendiz. E, quanto mais habilitado estiver este aprendiz, menores serão suas chances de cometer erros durante as situações reais.

O ambiente desenvolvido será útil não apenas para a análise em tempo real das simulações, como também para a observação e análise de cada etapa de todo processo de crescimento e formação da estrutura neural, incluindo fatores que influenciam no crescimento. Assim, o usuário poderá acompanhar e estudar a plasticidade que ocorre com as células na presença de substâncias químicas. Outro grande atrativo do ambiente é que ele permitirá que algumas células sejam mortas, simulando a apoptose que ocorre naturalmente nos seres vivos. Com isso, as células que estiverem em volta das que morreram se reorganizarão para tentarem formar novas conexões, suprimindo, desta forma, a falta das células mortas. Além disso, pode-se dizer que a possibilidade de se

executar diversas vezes o mesmo experimento, sem se preocupar em danificar os tecidos neurais é uma outra característica importante do sistema. Conseqüentemente, o ambiente gerado poderá contribuir consideravelmente para o avanço nos estudos do comportamento de células neurais.

### **39. PESQUISAS FUTURAS**

O ambiente gráfico aqui apresentado, embora inclua diversas opções de interação, ainda necessita de vários ajustes e melhorias. A seguir serão apresentadas algumas sugestões para a continuidade do trabalho:

- O ambiente de realidade virtual trabalhar com mais de uma única conexão, ou seja, existirem mais pessoas conectadas ao mesmo simulador, visualizando e interagindo de maneiras diferentes a mesma cena. Sendo assim, é necessário que o ambiente gerado permita a inclusão de várias conexões ao mesmo tempo;
- Hoje, calcula-se o campo trófico como sendo um ponto no centro da esfera. Seria interessante (e mais realista) considerar o campo como diversos pontos do mesmo campo ou mesmo de diferentes campos;
- Implementação da visão estéreo, que tornaria o ambiente muito mais atrativo e interessante;
- Otimização dos objetos da cena para que a visualização e interação tornem-se mais rápidas. Isto também permitiria a inclusão de uma quantidade maior de células na estrutura;
- Implementação e visualização do potencial de ação. Esta implementação seria tarefa do simulador, porém o ambiente gráfico já deveria estar considerando as conexões das células para permitir que o usuário possa acompanhar as trocas de sinais entre elas.



#### 40.REFERÊNCIAS BIBLIOGRÁFICAS

- ANDERSON, J. A. *An Introduction to Neural Networks*, MIT Press, 1995
- ANNUNCIATO, N.F. O processo plástico do sistema nervoso – publicação *Temas sobre o desenvolvimento*, V. 3, n. 17, p. 4 – 12, 1994.
- ANNUNCIATO, N.F. Participação dos fatores neurotróficos na regeneração do sistema nervoso, V. 4, n. 23, p. 30 – 38, 1995.
- APOPTOSE - <<http://www.icb.ufmg.br/~pat/Apoptose.htm>> Acesso em 18/05/04.
- ASCOLI, G. A. et al. Computational Neuroanatomy of the Hippocampus, Draft of the poster for *SFN98*, Los Angeles, 1998, <http://www.krasnow.gmu.edu/ascoli/sfn98/index.html> (acessado em 07/12/2002).
- BANGAY, Shaun. Experiencies in porting a Virtual Reality system to Java, *ACM*, p. 33-37, november 2001.
- BILLARD, Aude; MATARIC, Maja J. A biologically inspired robotic model for learning by imitation, *Agents 2000*, Barcelona Spain, *ACM 2000*.
- BOWER, J. M.; BEEMAN, D. *The Book of GENESIS*, Springer-Verlag New York , Inc. Published by TELOS, 1995.
- BRAGA, Antônio Pádua; CARVALHO, André Ponce; LUDEMIR, Teresa Bernarda. *Fundamentos de Redes Neurais Artificiais, 11º Escola de Computação*, Rio de Janeiro: UFRJ, 1998.
- BRYSON, Steve. Virtual Reality in Scientific Visualization. *Communications of the ACM*, Vol. 39, No. 5, may 1996.
- BÜLTHOFF, H. H.; VAN VEEN, A. H. C. Vision and Action in Virtual Enviroments: Modern Psychophysics In Spatial Cognition Research, *Technical Report No. 77*, Max-Planck-Institut für Biologische Kybernetik, Germany, 1999.
- CANTU, Marco. *Dominando o Delphi 7: A Bíblia*. Ed: Makron Books, 2003. ISBN: 8534615187.
- COELHO, Regina Célia; COSTA, Luciano Fontora, Realistic neuromorphic models and their application to neural reorganization simulations, *Neurocomputing*, No. 48, p. 555-571,2002.
- COELHO, R. C. ; JAQUES,O. Generating Three-Dimensional Neural Cells Based On Bayes Rules And Interpolation With Thin Plate Splines, 8th Iberoamerican Congress on Pattern Recognition, Havana, Cuba, in: *Progress in*

Pattern Recognition, Speech and Image Analysis, *Lecture Notes in Computer Science*, v. 2905, p. 675-682, 2003.

COSTA, Luciano; OLIVEIRA, M. Cristina; MINGHIM, Rosane. De olho nos neurônios – As formas virtuais do pensamento, *Ciência Hoje*, vol. 28, No. 167, 2000.

DOWNLING, J. E. *Neurons and Networks: an introduction to neuroscience*, Harvard University Press, 1992.

FOLEY, J. D.; DAM, A. V.; FEINER, S. K.; HUGHES, J. F. *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Company, 1996.

GLSCENE - <<http://glscene.sourceforge.net/index.php>> Acesso em 02/04/04.

[GOULD, 1999a] GOULD, E; REEVES, AJ; GRAZIANO, MS; GROSS, CG. Neurogenesis in the neocortex of adult primates, *Science*, Oct 15, 286(5439): 548-52, 1999.

[GOULD, 1999b] GOULD, E; TANAPAT, P. Stress and hippocampal neurogenesis. *Biol Psychiatry*, Dec 1;46(11):1472-9, 1999.

GOULD, E; TANAPAT, P; MCEWEN, BS; FLUGGE, G; FUCHS, E. Proliferation of granule cell precursors in the dentate gyrus of adult monkeys is diminished by stress, *Proc Natl Acad Sci U S A*, Mar 17;95(6):3168-71, 1998.

GRANT, H; LAI, Chuen-Ki. Simulation modeling with artificial reality technology (SMART): an integration of virtual reality and simulation modeling, *Winter Simulation Conference - Proceedings of the 30th conference on Winter simulation*, p. 437 – 442, 1998.

HEARN D.; BAKER M. P. *Computer Graphics, C Version*, Prentice Hall, 1997.

IOVINE, John. Step into Virtual Reality – Ed. WindCrest/McGraw-Hill, 1995 - ISBN:007911906-9.

KOCH, C.; SEGEV, I. *Methods in Neuronal Modeling*, The MIT Press, 1995.

KOVACS, Zolt. *Redes Neurais Artificiais: Fundamentos e Aplicações*, São Paulo: Collegium Cognitio, 1996.

LEIGH, J. et al. Virtual Reality in Computational Neuroscience, in: *Virtual Reality and Its Applications*, eds: R. Earnshaw; H. Jones; J. Vince, Academic Press, London, 1994.

LENT, Roberto. Cem Bilhões de Neurônios: conceitos fundamentais de neurociência. Editora: Atheneu, 2001.

MARLANGUE, C. C.; ZOUAOU, D. A.; REPRESA, A.; ARI, Y. B. Apoptotic features of selective neuronal death in ischemia, epilepsy and gp 120 toxicity, *TINS*, v. 19, n. 3, p. 109-114, 1996.

MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, No. 5, p. 115 - 133, 1943.

MENOUSEK, Bill; WOLFE, Terry. Virtual Reality the Modular Way, p. 66 – 68, August 1997.

NECROSE - <http://www.fo.usp.br/lido/patoartegeral/patoarteneec.htm>> Acesso em 20/05/2006.

NEURÔNIO - <[http://www.corpohumano.hpg.ig.com.br/sist\\_nervoso](http://www.corpohumano.hpg.ig.com.br/sist_nervoso)> - Acesso em 20/05/2006.

OLIVEIRA, Wilson José de. PROGRAMAÇÃO PARA INTERNET COM APLICAÇÕES EM DELPHI. Editora: VISUAL BOOKS, ISBN: 8575020358, 2002.

OPENGL - <<http://www.opengl.org>> - Acesso em 31/01/2006.

PURVES, D. *Body and Brain- A Trophic Theory of Neural Connections*, Harvard University Press, 1988.

PURVES, D. *Neural Activity and Growth of the Brain*, Cambridge University Press, 1994.

SILVA, Arídio. Dominando a Tecnologia de Objetos. Editora: BOOK EXPRESS, 2002. ISBN: 8586846961.

SINAPSES - <[http://www.webciencia.com/11\\_29nervoso.htm](http://www.webciencia.com/11_29nervoso.htm)> - ACESSO EM 31/01/2006.

SINGH, V. Real-time Volume Manipulation, *The Eurographics Association*, 2003

SOCKETS - <<http://www.indyproject.org/sockets/index.en.iwp>> - ACESSO EM 31/01/2006.

SUBRAMANIAN, Rajaram; MARSIC, Ivan. Vibe: Virtual Biology Experiments, Hong Kong, May, 1-5,2001.

SULLIVAN, M.; KAMPF, F.; KOCK, P. Optimization of a digital neuron design, *ACM SIGSIM Simulation Digest*, Proceedings of the 23rd annual symposium on Simulation, v. 20,P. 73 – 80, 1990.

TAFNER, Malcon; FILHO, Marcos de Xerez Rodrigues. Redes Neurais Artificiais: Introdução e Princípio da Neurocomputação, *FURB*, 1995.

TARR, Michael; WARREN, Willian H. Virtual reality in behavioral neuroscience and beyond, *Nature neuroscience supplement*, vol. 5, p. 1089 –1092, november 2002.

ZAJTCHUK, Russ; SATAVA, Richard M. Medical Applications of Virtual Reality, *Communications of the ACM*, vol. 40, No. 9, September 1997.

WOO, Mason; NEIDER, Jackie; DAVIS, Tom; SHREINER, Dave. OpenGL Architecture Review Board: *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL*, 3rd edition, 800 pages, Addison-Wesley Pub Co, ISBN: 0201604582, 1999.