UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Desenvolvimento de um Sistema de Visualização para Neurônios Artificiais

Danielle Bentivoglio Colturato
Maria Cristina Ferreira de Oliveira
Antonio Valério Netto

Nº 167

RELATÓRIOS TÉCNICOS

São Carlos – SP
Mai./2002

SYSNO 1253593
DATA / /
ICMC - SBAB

DESENVOLVIMENTO DE UM SISTEMA DE VISUALIZAÇÃO PARA NEURÔNIOS ARTIFICIAIS

Danielle Bentivoglio Colturato[1]
Maria Cristina Ferreira de Oliveira[2]
Antonio Valerio Netto[3]

Departamento de Ciências da Computação e Estatística
Instituto de Ciências Matemáticas e de Computação - ICMC
Universidade de São Paulo - USP - Campus São Carlos
Caixa Postal 668 - CEP 13560-970 - São Carlos, SP

Abstract: *This article describes an application developed in collaboration with the researchers of the Cybernetic Vision Group of IFSC–USP (Institute of Physics of São Carlos – University of São Paulo) that presents an interactive environment for data visualization of neurons generated by those users. Aiding in that way, the analysis process of the same ones. This system allows manipulation of the neurons by isosurface cut, parameter choice (color, type of visualization technique, isosurface value, and so on.). To aid the system development, it has used the VTK (The Visualization Toolkit) classes library.*

Keywords: *Computer Graphics, Scientific Visualization, VTK, Visualization System, Neuron.*

Resumo: Este artigo descreve uma aplicação desenvolvida em colaboração com os pesquisadores do Grupo de Visão Cibernética do IFSC-USP (Instituto de Física de São Carlos – Universidade de São Paulo) que visa apresentar um ambiente interativo para visualização de dados de neurônios gerados por esses usuários, auxiliando dessa forma, o processo de análise dos mesmos. Este sistema permite manipulação dos neurônios por meio de cortes na isosuperfície, escolha de parâmetros (cor, tipo de técnica de visualização, valor da isosuperfície, etc.). Como ferramenta para auxiliar o desenvolvimento do sistema foi utilizada a biblioteca de classes VTK (*The Visualization Toolkit*).

Palavras-Chave: Computação Gráfica, Visualização Científica, VTK, Sistema de Visualização, Neurônio.

[1, 2, 3] Os autores agradecem o apoio do CNPq (Processo 521931/97-5) e da FAPESP (Processo 00/04779-2) e 98/161173-0)

1. INTRODUÇÃO

A crescente disponibilidade de recursos computacionais para o cálculo, simulação e aquisição de dados permite que cientistas e engenheiros produzam enormes conjuntos de dados, bi ou tridimensionais, em geral multivariados. A aplicação de técnicas de Computação Gráfica a dados científicos com o objetivo de ganhar compreensão desses dados, testar hipóteses e esclarecer fenômenos e efeitos em geral compreende o objeto de estudo da área conhecida atualmente por Visualização Científica (ViSC - *Visualization in Scientific Computing*) [8].

O Grupo de Computação Gráfica e Processamento de Imagens (CG & PI) do ICMC-USP (Instituto de Ciências Matemática e de Computação) vem trabalhando no estudo, implementação e melhoria de técnicas de Visualização no contexto do projeto *PowerVis* [9] [11] [12]. Esse projeto visa a criação e implementação de técnicas multimodais de visualização para facilitar o uso das técnicas de mapeamento gráfico existentes diretamente por usuários em potencial. Nesse contexto, colaborações estão em andamento com diferentes grupos de pesquisa que são usuários de visualização e que fornecem os dados utilizados para as pesquisas em desenvolvimento.

Uma das colaborações em andamento na área de Visualização é com os pesquisadores do Grupo de Visão Cibernética do IFSC. Como decorrência dessa colaboração, foram identificadas várias técnicas que podem ser aplicadas à visualização dos dados gerados por esses usuários, apoiando o processo de análise dos mesmos. O grupo de Visão Cibernética tem trabalhado intensamente na análise da forma de neurônios e estruturas neurais [1] [2] [3]. Como parte desse trabalho, foram gerados modelos artificiais realísticos de estruturas neurais com os objetivos tanto de simular sistemas naturais como de investigar novas arquiteturas e paradigmas em redes neurais artificiais.

Várias visualizações foram geradas a partir desses dados com o objetivo de exibir as formas dos neurônios artificiais depois da aplicação da função Gaussiana. Foram aplicadas técnicas de extração de superfícies e de *rendering* volumétrico, utilizando procedimentos do VTK que implementam as técnicas conhecidas como *marching cubes* [7] e *ray casting* [6], respectivamente. O ambiente desenvolvido é dedicado a atender as necessidades encontradas pelos pesquisadores de visualizarem/interagirem os modelos tridimensionais dos neurônios.

O sistema oferece uma interface adequada para o usuário executar as visualizações desejadas e alterar os parâmetros das mesmas, por exemplo: o tipo de técnica a ser usada, o número de isosuperfícies a serem geradas e o valor das mesmas; a tabela de cores a ser adotada; a taxa de amostragem dos dados vetoriais; o tipo de *glyph* a ser exibido, etc. Adicionalmente, foram

incorporadas algumas estratégias de interação, além dos recursos básicos oferecidos pelo VTK (rotação e escala via mouse).

A seção 2 deste artigo apresenta algumas técnicas convencionais de visualização. Na seção 3 são apresentadas noções básicas sobre o *software Visualization Toolkit*. Na seção 4 é descrito o ambiente de visualização que foi desenvolvido e na seção 5 são apresentados os pareceres conclusivos do trabalho.

2. TÉCNICAS DE VISUALIZAÇÃO DE DADOS

Existem inúmeras técnicas para visualizar dados, mas para isto eles devem estar organizados de alguma forma. Esta organização pode variar bastante, os dados podem surgir na forma de um conjunto de pontos esparsos, distribuídos sem qualquer organização pré-definida, ou na forma de uma malha que exibe alguma regularidade.

De acordo com a organização definida pela estrutura, pode-se ter vários tipos de malhas:

- As malhas cartesianas, onde todos os elementos são quadrados (no caso 2D) ou cubos idênticos (no caso 3D) alinhados aos eixos principais.
- As malhas regulares, que possuem todos os seus elementos idênticos e alinhados aos eixos, mas estes são retângulos regulares ao invés de quadrados ou cubos;
- As malhas retilíneas apresentam-se na forma de hexaedros alinhados aos eixos, mas não necessariamente idênticos;
- As malhas estruturadas, onde os elementos são quadriláteros ou hexaedros não alinhados aos eixos principais e;
- As malhas desestruturadas que são formadas por polígonos ou poliedros sem qualquer padrão explícito de conectividade.

Os dados em si podem ser de natureza escalar (números que indicam temperatura, pressão, umidade, etc.), vetorial (vetores que indicam velocidade, campo elétrico, etc.) ou tensoriais (tensores que indicam fadiga de um material, etc.) No caso de dados escalares, existem dois tipos básicos de visualização: a visualização por extração de superfícies e a técnica de *rendering* volumétrico direto (DVR – *Direct Volume Rendering*).

A extração de superfícies inclui dois métodos: de conexão de contornos, muito usados em aplicações médicos e o de intersecção de *voxels*. No primeiro, há a identificação dos contornos

para um valor de interesse em cada fatia do volume de dados, e a seguir, estas curvas fatiadas de forma adjacentes são conectadas, em uma abordagem denominada tecelagem (*patching*), ou triangulação. No segundo método, há a geração de isolinhas (caso 2D) ou isosuperfícies (caso 3D) selecionando valores escalares de interesse. A isosuperfície definida pelo valor escalar só poderá atravessar uma célula segundo um número finito de maneiras distintas, as quais dependem de quais vértices da célula estão dentro da superfície, e quais estão fora. Exemplos de algoritmos pertencentes a este método, são o *marching cubes* e o *dividing cubes* [9].

O volumétrico direto (DVR) não cria estruturas geométricas intermediárias, como superfícies. Ela permite ao usuário enxergar "dentro" (ou "através") do volume de dados, possibilitando a visualização de mais informações que as técnicas de *rendering* de superfícies, mas para que isto ocorra é necessário trabalhar com transparência. O DVR é tradicionalmente realizado pela técnica de lançamento de raios (*ray casting*).

No caso de dados vetoriais, uma técnica natural é a utilização de setas, ou seja, segmentos de retas orientados para cada vetor (*glyphs*). Outro exemplo de técnica para visualização de dados vetoriais são as *streamlines*, construídas escolhendo-se um ponto inicial de onde partem a construção de segmentos de reta fazendo uma média dos vetores mais próximos.

3. VISUALIZATION TOOLKIT

O VTK (*Visualization Toolkit*) é um *software* multi-plataforma de baixo custo [14] que consiste de uma biblioteca de algoritmos e estruturas de dados para visualização compatível com plataformas UNIX, Windows NT e Windows 9x. O *software* foi desenvolvido em C++ segundo o paradigma de orientação a objetos e é extensível, permitindo o desenvolvimento de novas aplicações com uma interface gráfica própria, a alteração das técnicas já implementadas e a inclusão de novas técnicas de visualização, pois o código fonte está disponível.

A geração de visualizações é feita com a criação de *pipelines* de visualização, que descrevem uma seqüência de transformações sobre os dados de entrada até o *rendering* final na tela. O VTK possui um conjunto de classes com funcionalidades específicas, que podem ser dos tipos Fonte, Filtro, Mapeador e Gráfico. Objetos da classe Fonte iniciam o *pipeline* de visualização, fazendo a leitura de arquivos de dados externos ou criando dados a partir de variáveis de instância, e gerando um ou mais conjuntos de dados de saída. Objetos da classe Filtro recebem pelo menos um conjunto de dados de entrada e geram um ou mais conjuntos (em geral, transformados) de dados de saída. Os Mapeadores finalizam o *pipeline* de visualização, recebendo um ou mais

objetos de entrada, e fazem o mapeamento desses dados para um determinado dispositivo ou para uma biblioteca gráfica. Os Escritores (*writers*) são mapeadores que escrevem dados em um arquivo. Os objetos do tipo Gráficos implementam a funcionalidade do núcleo de *rendering*. Esses objetos permitem a manipulação de fontes de luz, câmeras e atores, das propriedades dos atores e dos atributos de *rendering* (como a cor de fundo).

4. SISTEMA DE VISUALIZAÇÃO DE NEURÔNIOS

Todas as sensações e percepções dos seres humanos dependem da continuada dinâmica de reações eletroquímicas que acontecem nos nossos neurônios. Processos sofisticadíssimos acontecem no nosso cérebro envolvendo comunicações entre bilhões de neurônios e ativações de diversas áreas corticais, um feito que nenhum computador existente consegue reproduzir. Desde sua origem, no século passado, a neurociência moderna tem tentado compreender o funcionamento desses processos. Embora nosso conhecimento sobre o cérebro ainda seja muito limitado, os avanços científicos e tecnológicos alcançados no decorrer deste século representam oportunidade única para grandes descobertas em neurociência. De fato, parcela significativa dos avanços mais recentes na área foi conseqüência direta ou indireta do uso de técnicas computacionais [5].

O grupo de Visão Cibernética do IFSC-USP tem trabalhado intensamente na análise da forma de neurônios e estruturas neurais. Como parte desse trabalho, foram gerados modelos artificiais realísticos de estruturas neurais com os objetivos tanto de simular sistemas naturais como de investigar novas arquiteturas e paradigmas em redes neurais artificiais. Particularmente, Coelho e Costa [3] têm trabalhado na síntese de estruturas neurais por meio de gramáticas estocásticas que operam sobre uma série de regras de produção bem definidas derivadas a partir de caracterizações estatísticas e fractais de cada tipo de neurônio.

Uma das saídas geradas por essas gramáticas, com o objetivo específico de gerar uma representação gráfica para propósitos de visualização, consiste em uma descrição do neurônio na forma de um poligonal bi ou tridimensional definida em um espaço discreto. Nesse espaço discreto, valores 0 ou 1 indicam a presença ou ausência do neurônio, respectivamente. Esses dados foram processados por meio da aplicação de uma Gaussiana, com o objetivo de estudar as interações entre os diferentes componentes dos neurônios. A função Gaussiana 'expande' a fronteira do neurônio num conjunto de valores em uma faixa de variação específica, o resultado é um conjunto de níveis de cinza, ou seja, valores escalares entre 0 e 1 posicionados no espaço

discreto. O campo magnético na região neuronial também pode ser simulado, o que resulta em um conjunto de dados vetoriais, também posicionados em um espaço discreto.

Esses dados foram processados para gerar uma malha cartesiana estruturada (com geometria e topologia regular) de forma a permitir a aplicação de técnicas convencionais de extração de superfícies e de *rendering* volumétrico. Para isso, foram inseridos vértices no volume de dados aos quais foi associado o atributo escalar 2.0. Dessa forma, na malha de dados, os valores entre 0 e 1 representam informação de interesse, e o valor 2.0 corresponde ao 'fundo'. Várias visualizações foram geradas a partir desses dados, utilizando a biblioteca VTK. O objetivo das visualizações iniciais era exibir as formas dos neurônios depois da aplicação da Gaussiana, o que poderia fornecer subsídios para uma análise quantitativa dos resultados produzidos pela gramática de simulação.

Uma das dificuldades encontradas é que os usuários, em geral, não dominam o processo de geração das visualizações, que é feito através de programação em C++ utilizando as classes da biblioteca VTK. Isso implica na necessidade de apoio constante de alguém que domine o processo sempre que precisam gerar visualizações de novos conjuntos de dados, mesmo quando isso não envolve a identificação de técnicas e estratégias de visualização alternativas.

Sendo assim, foi proposta a implementação de um sistema que integrasse as técnicas básicas de visualização de superfícies, volumétricas e vetoriais e que adicionalmente implementasse algumas técnicas de interação que auxiliasse a exploração e manipulação dos modelos gerados.

O ambiente desenvolvido permite ao usuário a definição de parâmetros como, no caso de extração de superfícies, definir o valor do contorno a ser visualizado, a cor relativa ao mesmo e o coeficiente de transparência. Também permite ao usuário alguns tipos de interações com o neurônio visualizado, como a definição do ângulo de rotação e zoom. Também é possível realizar 'cortes' no volume de dados segundo regiões previamente definidas; passar uma sonda pelo volume colhendo informações sobre os dados, gerar um arquivo VRML do modelo resultante e atualizar animações. No caso do *rendering* volumétrico direto, é possível definir os intervalos a serem visualizados, definir a tabela de cores utilizada para cada intervalo e o grau de transparência para os mesmos. Também é possível utilizar alguns recursos de interação como por exemplo, definir o ângulo de rotação e *zoom*, passar uma sonda pelo volume de dados colhendo informações sobre os dados e apresentá-los na forma de som [13].

Algumas visualizações obtidas para dois conjuntos de dados escalares distintos são mostradas nas Figuras 1 e 2. A Figura 1 ilustra uma visualização gerada por extração de superfícies, contendo três isosuperfícies com valores 0.77, 0.92 e 0.93. As duas isosuperfícies

mais externas possuem um certo grau de transparência para que as mais internas possam ser visualizadas. A Figura 2 ilustra uma visualização obtida por técnica de *rendering* volumétrico direto. O modelo possui três faixas de interesse, sendo que a mais interna é totalmente opaca e nas outras duas também são aplicadas transparências.

As Figuras 3 e 4 apresentam visualizações obtidas a partir dos conjuntos de dados vetoriais que descrevem o campo magnético ao redor dos neurônios. A Figura 3 apresenta os dados por meio de *glyphs* e a Figura 4 por *Streamlines*.

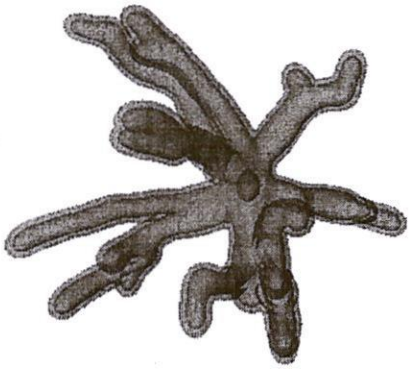

Figura 1: Neurônio visualizado por extração de superfícies, mostrando diferentes superfícies, com transparência.

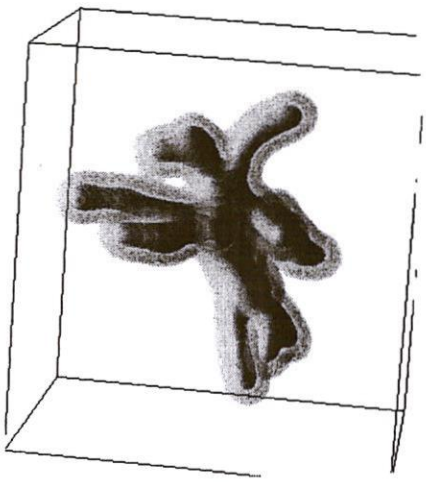

Figura 2: Neurônio visualizado por *Rendering* Volumétrico Direto.

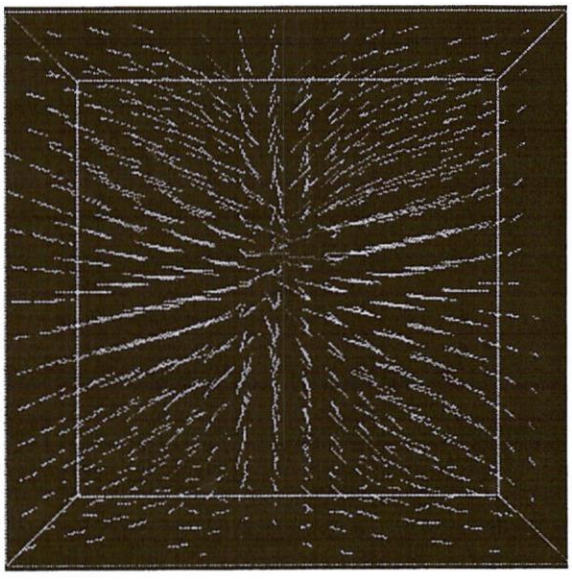

Figura 3: Visualização por *Glyphs* do campo vetorial de um neurônio.

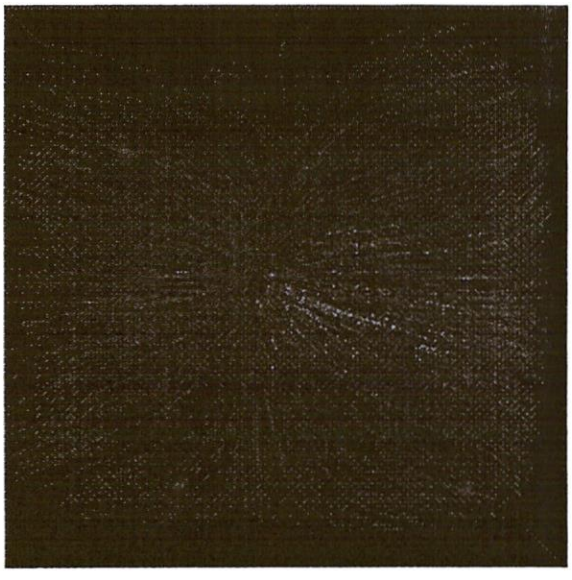

Figura 4: Visualização por *Streamlines* do campo vetorial de um neurônio.

No módulo das isosuperfícies, a visualização do arquivo VRML possibilita ao usuário chamar o *browser default* e abrir o arquivo ".wrl" para exploração ou navegação. Para a realização do processo de extração de regiões deve-se, primeiramente, escolher a isosuperfície de interesse. A escolha da isosuperfície é feita da seguinte maneira: o usuário escolhe a isosuperfície de interesse no *grid* do formulário de entrada, clicando sobre a linha correspondente e a isosuperfície especificada será destacada em relação às demais. O número da isosuperfície, assim como os seus valores, ficam disponíveis no formulário de entrada para que o usuário possa consultá-los quando necessário. No caso apresentado na Figura 5, foi escolhida a isosuperfície número 1, que corresponde ao valor 0,77. As demais isosuperfícies são mostradas como nuvens de pontos, deixando a isosuperfície escolhida em destaque. Após a escolha da isosuperfície o usuário pode eliminar as demais e trabalhar somente com a superfície escolhida.

Na Figura 6a pode ser visto os cortes na isosuperfície que são realizados selecionando pontos da superfície por meio de um cursor. Ao posicionar o cursor sobre um ponto de interesse, este ponto e suas arestas são ressaltados com uma cor previamente escolhida pelo usuário. Ao selecionar este ponto, por meio de um clique sobre o mesmo, somente os pontos vizinhos poderão ser selecionados. O processo de demarcação só é finalizado após a demarcação de toda região (Figura 6b). Após a demarcação é possível realizar o corte propriamente dito, bastando pressionar a tecla "u" e separar a região delimitada da superfície original (Figura 6c).

O processo de separação é realizado da seguinte maneira: o algoritmo escolhe uma célula qualquer pertencente ao modelo e verifica quais são suas células vizinhas. As células vizinhas são encontradas verificando-se quais células compartilham uma mesma aresta, sendo que sempre que for encontrada uma aresta pertencente à região de interesse a nova célula que compartilhar esta aresta não será selecionada. Este processo é recursivo, até que todas as células, com exceção das que estiverem do lado contrário das arestas de interesse, sejam encontradas. Após o processo de separação o usuário pode selecionar uma das regiões de interesse. Por exemplo, na Figura 6c o usuário pode escolher a região em azul ou a região em rosa utilizando as teclas "c" ou "o", que permitem a alternância entre o modo câmera e o modo objeto. No modo câmera os eventos do *mouse* afetam a posição da câmera e do ponto focal, e no modo objeto os eventos do *mouse* afetam o ator que está sob o ponteiro do mesmo. Dessa forma, pode-se realizar ações sobre cada objeto independentemente, como operações de translação, etc.

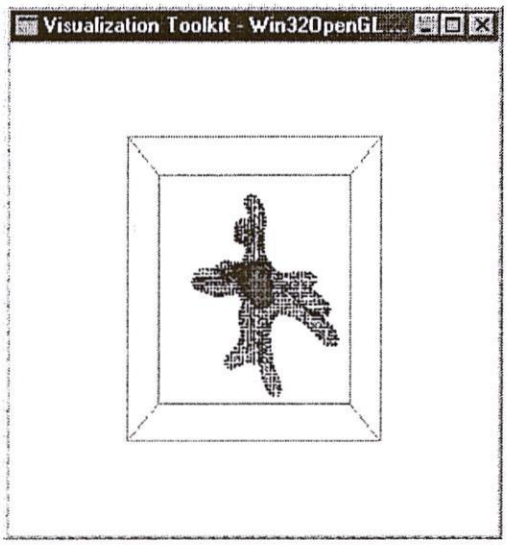

Figura 5: Isosuperfície de interesse em destaque em relação as demais.

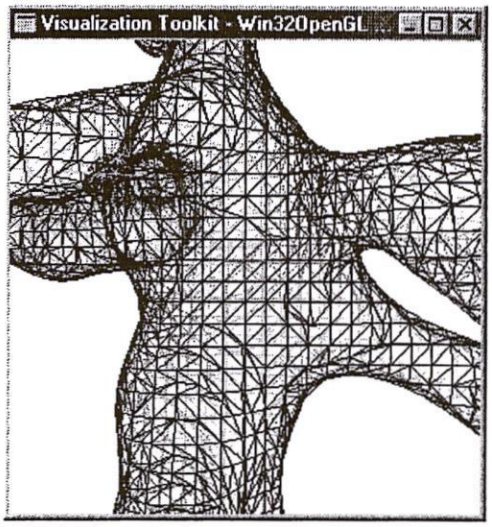

Figura 6a: Cursor utilizado para selecionar os pontos da região a ser cortada da superfície.

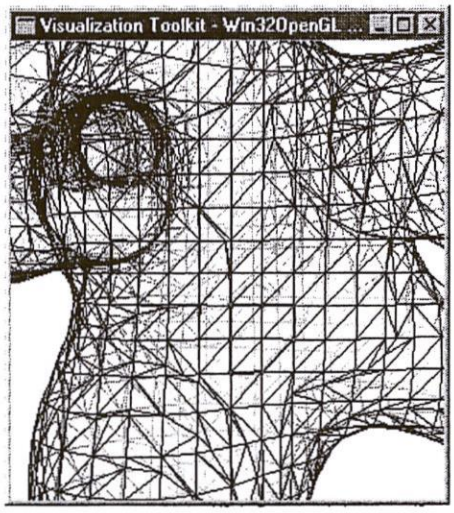

Figura 6b: Demarcação da região de corte.

Figura 6c: Corte de parte da superfície do neurônio.

5. CONCLUSÕES

Neste artigo, foi apresentado um ambiente computacional que englobasse as principais técnicas de visualização e que possibilitasse alguns tipos de interações com os modelos para fins de exploração e manipulação. O ambiente vem sendo testado e tem cumprindo os seus propósitos. No futuro várias atualizações ou extensões podem ser realizadas, como, por exemplo, incorporar uma função que possibilite cortes em dados volumétricos.

AGRADECIMENTOS

Os autores agradecem o apoio recebido pelo Professor Luciano da Fontoura Costa, coordenador do Grupo de Visão Cibernética do IFSC-USP, pelos arquivos de dados que foram utilizados pelo sistema. Este projeto tem o apoio financeiro da FAPESP (Fundação de Apoio à Pesquisa do Estado de São Paulo).

REFERÊNCIAS BIBLIOGRÁFICAS

1. Cesar R.M. Jr.; Coelho, R.C.; Costa, L.F. Automatic Classification of Retinal Ganglion Cells. Proceedings II Workshop on Cybernetic Vision, Dec. 1996, São Carlos, SP, Brazil. IEEE CS Press, 1997, pp. 51-56.
2. Cesar, R.M. Jr.; Costa, L. F. Semi-Automated Dendogram Generation for Neural Shape Analysis. Proceedings SIBGRAPI'97, Campos do Jordão, SP, 1997, pp. 147-154.
3. Coelho, R.C.; Costa, L.F. Gramáticas para Sínteses de Estruturas Neurais. Anais do Workshop sobre Visão Cibernética, São Carlos, IFSC-USP, 1994, pp. 74-79.
4. Coelho, R.C.; Costa, L.F. The Box-Counting Fractal Dimension: Does it Provide an Accurate Subsidy for Experimental Shape Characterization? If So, How to Use It? Anais do SIBGRAPI'95, São Carlos, pp. 183-191.
5. Costa, L. F.; De Oliveira, M. C. F.; Minghim, R. Visualizando as Formas do Pensamento. Ciência Hoje, vol. 28, n. 167, dez. 2000, pp. 44-51.
6. Elvins, T. T. A Survey of Algorithms for Volume Visualization. ACM SIGGRAPH Computer Graphics, vol. 26, n. 3, mar. 1992, pp. 34-44.
7. Lorensen, W. E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm.

ACM SIGGRAPH Computer Graphics, vol. 21, 1987, pp. 163-169.

8. McCormick, B.H.; De Fanti, T.A.; Brown, M.D. Visualization in Scientific Computing: A Synopsis. IEEE CG&A, vol. 7, no. 4, pp. 61-70, 1987.
9. Minghim, R.; Oliveira, M.C.F. de. PowerVis: Empowering the User with a Multi-Modal Visualization System. Proceedings II Workshop on Cybernetic Vision, Dezembro de 1996, IEEE CS Press, 1997, pp. 106-111.
10. Minghim, R.; Olveira, M.C.F. de. Uma Introdução à Visualização Computacional. XVI JAI - Jornadas de Atualização em Informática, XVII Congresso da Sociedade Brasileira de Computação, cap. 3, pp. 85-132.
11. Minghim, R.; Oliveira, M.C.F. de. Visualization Experiences in the Light of Visualization Effectiveness. Proceedings Int. Symposium on Advances on Signal, Image Processing, Computer Vision and Graphics of the World Multiconference on Systemics, Cybernetics and Informatics (SCI-98), Orlando, USA, 12 a 16 de Julho de 1998, pp. 529-536.
12. Oliveira, M.C.F. de; Minghim, R.; Tutida, S.M.; Salvador, V.C.; "Towards Perception Driven Visualization", Anais do 1998 International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI'98), Rio de Janeiro, RJ, 20 a 23 de Outubro de 1998, pp. 127-134.
13. Salvador, V. C. L. Sonificação para apoio a tarefas de visualização. São Carlos, 1998. 113p. Dissertação (Mestrado em Ciência da Computação), Universidade de São Paulo.
14. Schroeder, W.; Martin K.; Lorensen B. The Visualization Toolkit – an object-oriented approach to 3D graphics. Prentice Hall, 1998.