

# Um Sistema de Visualização de Crescimento Neural Simulado

## Autores

---

Regina Celia Coelho  
Nivaldi Calonego Junior  
Luis Augusto Consularo

## 1. Introdução

---

Muitos desafios têm sido impostos ao entendimento do processamento neural, que tem na análise funcional e comportamental elementos indispensáveis ao entendimento do processo de desenvolvimento humano. No que tange aos fatores biológicos, especificamente na Neurociência, há a necessidade de estudos do crescimento dos neurônios, dado que pesquisas mostram que a geometria tem forte correlação com a funcionalidade [Purves, 1988] [Kossel, 1995]. A geometria dos neurônios pode ser modelada por estruturas ramificadas [Ascoli, 2000], havendo a possibilidade da simulação do crescimento das células, a análise da morte e modificações devido a campos tróficos (campos de atração ou repulsão das células) [Coelho, 2002].

A quantidade de dados gerada nos processos de simulação inviabiliza a observação direta dos valores numéricos, havendo a necessidade de algum processamento que os ilustre graficamente. A visualização de quantidades massivas de dados ainda não é uma tarefa bem resolvida em grande parte das aplicações, pois requer investimentos em dispositivos dedicados a alguma aplicação específica.

Esta necessidade de entendimento dos dados tem exigido grande poder das ferramentas disponíveis de análise de dados, que tem, cada vez mais, lançado mão de abordagens inovadoras, oferecendo aos usuários a capacidade de visualizá-los ou manipulá-los. O processo mais simples de visualização 3D pode se resumir a: (1) construção do modelo 3D e (2) a renderização deste modelo 3D. Aplicações mais efetivas incluem interação, imersão e navegação de um ou mais usuários.

Essas aplicações impõem restrições computacionais relacionadas a cada uma destas etapas ou partes. A primeira etapa, isto é, a construção do modelo 3D deve satisfazer as restrições de tempo impostas pela aplicação. Se a aplicação requerer interação e navegação, por exemplo, as restrições de tempo-real exigirão que a entrada e a saída de dados do modelo construído obedeçam um fluxo que considere até mesmo uma degradação elegante de seus resultados.

Por sua vez, o renderizador também exige um fluxo contínuo de dados, mas suas restrições são diretamente relacionadas com a qualidade visual e tempo de montagem da cena. A interação requer o tratamento de eventos em tempo-real, ou seja, para que o usuário se perceba interagindo com o ambiente de visualização 3D exige-se um reduzido tempo de resposta. A imersão impõe restrições quanto à participação de outros usuários, quanto ao posicionamento do usuário no ambiente visualizado, entre outras. A navegação exige que os limites das incertezas do ambiente visualizado sejam computacionalmente conhecidos, ou que pelo menos sejam estimados.

## 2. Objetivos

---

O objetivo é apresentar os resultados obtidos com a implementação de um ambiente de visualização com

realidade virtual em que os usuários podem visualizar e interagir com células neurais morfológica e estatisticamente realistas. Isto mostra as possibilidades de estudo e análise do comportamento das células neurais em diferentes circunstâncias, por exemplo, na presença de campos tróficos, durante a neurogênese e após a apoptose.

### 3. Desenvolvimento

---

O simulador que gera células tridimensionais baseadas em medidas morfológicas extraídas de células naturais foi implementado [Coelho, 2002][Coelho, 2003]. O crescimento de células é realizado utilizando variações de gramáticas *L-Systems* estocásticas [Lindenmayer, 1975]. Uma das variações considera que cada regra de produção ocorre com uma probabilidade tomada de uma função densidade de probabilidade (função de distribuição) e não em uma única taxa de probabilidade. Cada neurônio é desenhado de forma análoga ao movimento da tartaruga (da linguagem LOGO [Papert, 1988]), porém a decisão do momento e da direção que a tartaruga deve tomar deve ser controlada por um conjunto de funções de distribuição, que descrevem as medidas dos neurônios naturais. As *L-Systems* controlam as ramificações, os tipos de dendritos dos neurônios (por exemplo, neurônios piramidais possuem dois grupos de dendritos bem distintos, sendo um grupo apical e outro basal) e as funções que devem ser utilizadas.

O simulador gera, portanto, os dados que devem ser transformados em cenas 3D com as quais os usuários podem interagir. Esses dados servem como entrada para o módulo lógico, responsável pela geração dinâmica e automática de um código VRML. A partir desse código, é possível que os dados sejam transformados em elementos gráficos e apresentados no terminal de vídeo, utilizando-se diferentes programas de realidade virtual.

A figura 1 ilustra um diagrama de classes, apresentando uma visão geral do protótipo implementado para a visualização de dados. A classe *Network* implementa um protocolo de comunicação em rede, permitindo que a simulação, a geração da cena VRML e a renderização sejam executadas em máquina distintas. A classe *PNDataSet* implementa o subsistema de geração de cenas, automatizando também a distribuição de mensagem entre os possíveis visualizadores instalados. Como os programas são relativamente independentes, há a possibilidade de diversos usuários estarem geograficamente distribuídos e analisando o mesmo conjunto de dados.

A classe *NGrowthCmd* implementa métodos específicos de interação com as estruturas neurais, permitindo, por exemplo, o tingimento das células, a alteração da intensidade de cor, os níveis de transparência ou a morte (eliminação) de partes das cenas.

A classe *Viewer* simboliza possíveis renderizadores para código VRML. Foram testados os renderizadores: Avalon, Shout3D e uma aplicação com a biblioteca OpenVRML. Os resultados apresentados a seguir foram obtidos a partir da implementação da aplicação específica para essa finalidade, que explora técnicas de realidade aumentada para ilustrar o crescimento neural e algumas das funcionalidade de interação implementadas.

### 4. Resultados

---

Os resultados ilustrados neste artigo foram obtidos com o programa desenvolvido especificamente para criar os meios de interação com células neurais geradas com estruturas ramificadas. A interface utiliza o rastreamento óptico de marcadores e foi implementado com a biblioteca ARToolkit, cujos dados são utilizados para atuar no grafo de cena renderizado com a biblioteca OpenVRML. As simulações foram

executadas em máquina Compaq© (Pentium IV, 1.8 GHz), e as interfaces de interação e navegação foram executadas em máquina Toshiba© Satellite A45-S1202 (Celeron, 2.8 GHz); ambos conectados via rede.

O ambiente permite alterar a forma das células. Isso significa que podem ser utilizadas diferentes geometrias, tais como: linha, esferas, ponto ou extrusão, conforme ilustra a figura 2. Também é possível alterar cores, níveis de transparência e inserir campos tróficos, conforme ilustra a Figura 3. Na figura 3, ilustra-se a resposta a um comando “set\_transparency s4 0.9”, que modifica os atributos do campo *transparency* do material cuja elemento é identificado por “material\_s4”, fazendo com que a célula toda se torne transparente. Caso se deseje alterar a cor de uma parte específica da célula, pode-se utilizar o comando “set\_diffuseColor s4\_1\_1 0 0 1”, que modifica apenas o node denominado “s4\_1\_1”. Analogamente, pode-se utilizar “emissiveColor”, entre outros comandos, prefixados por “set\_”, e concatenados ou não com “ALL”.

## 5. Considerações Finais

---

Os dados mostram que a geometria e a aparência utilizadas para a visualização influenciam significativamente o tempo gasto na produção da cena, sugerindo que a visão macroscópica da rede seja produzida por conjunto de pontos, minimizando recursos que definem a aparência.

A arquitetura do software desenvolvido permite a navegação na estrutura neural, a visualização estereoscópica e o uso de módulos simuladores distribuídos em rede, enviando dados ao módulo Controlador. O módulo Controlador é um gargalo do sistema. Porém, sua operação como cliente do Renderizador dá opções ao usuário entre visualizar durante a execução da simulação ou a reprodução do crescimento neural a partir de dados armazenados em arquivos, configurando taxas de transmissão que permitem navegação.

Dada a complexidade do problema, aceita-se que o usuário defina suas condições de tempo-crítico. Por exemplo, há a situação em que o usuário não interage e nem visualiza enquanto a simulação executa e deseja ver apenas o resultado final. Mas, noutros casos, a importância reside no processo de crescimento.

## Referências Bibliográficas

---

Ascoli, G. A.; Krichmar, J. L. (2000) “L-neuron: A modeling tool for the efficient generation and parsimonious description of dendritic morphology”, *Neurocomputing*, v. 32-33, 2000, p1003-1011.

Coelho, R. C.; Costa, L. F., (2002) “Realistic neuromorphic models and their application to neural reorganization simulations”, *Neurocomputing*, n. 48, 2002, p555-571.

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, p675-p682, 2003.

Kossel A, Löwel S, Bolz J. (1995) “Relationships between dendritic fields and functional architecture in striate cortex of normal and visually deprived cats”, *The Journal of Neuroscience*, v15, p3913-3926.

Lindenmayer, A. Developmental algorithms for multicellular organisms: a survey of L-systems, *Journal of Theoretical Biology*, **54**, 3-22, 1975.

Anexos

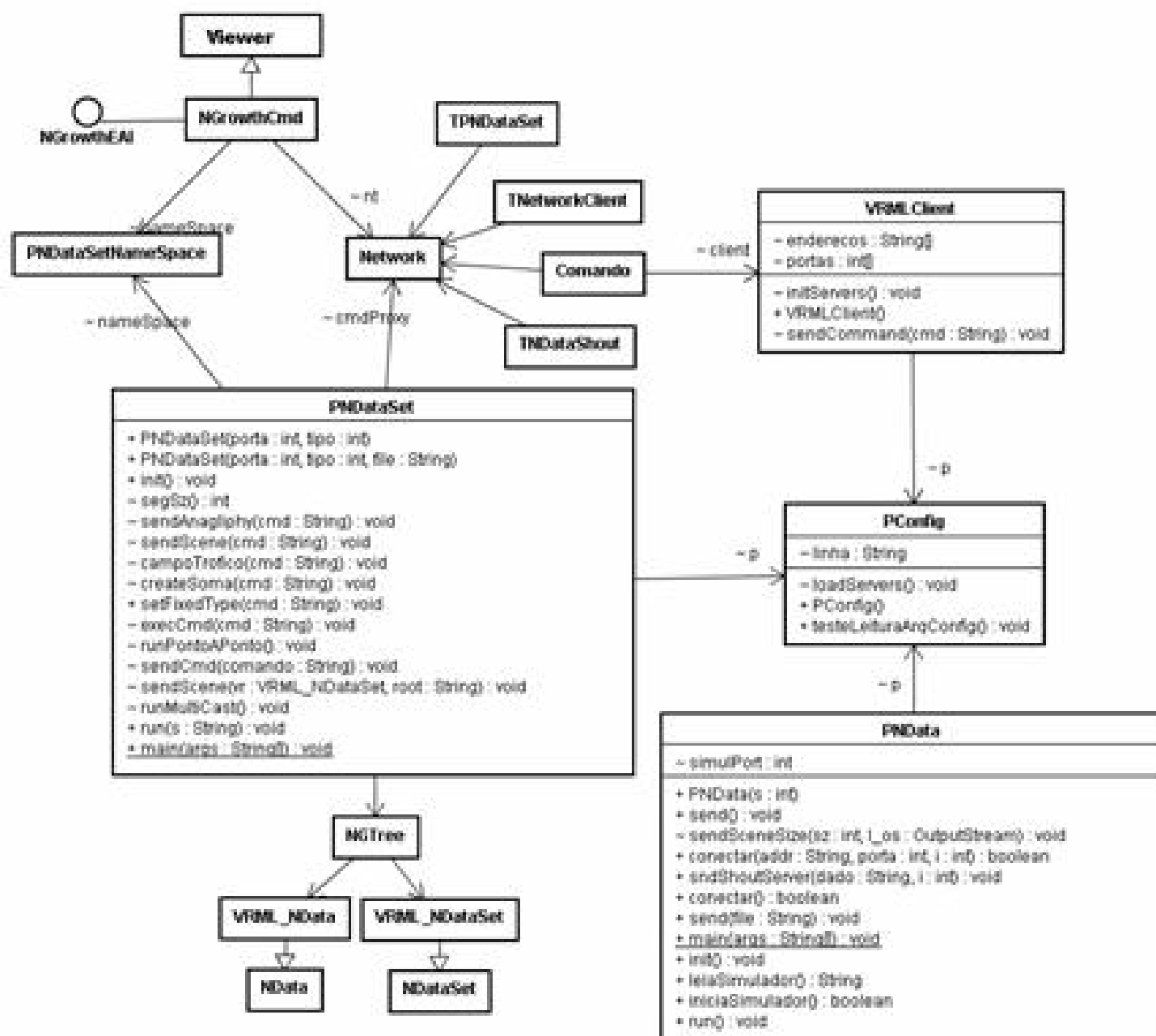
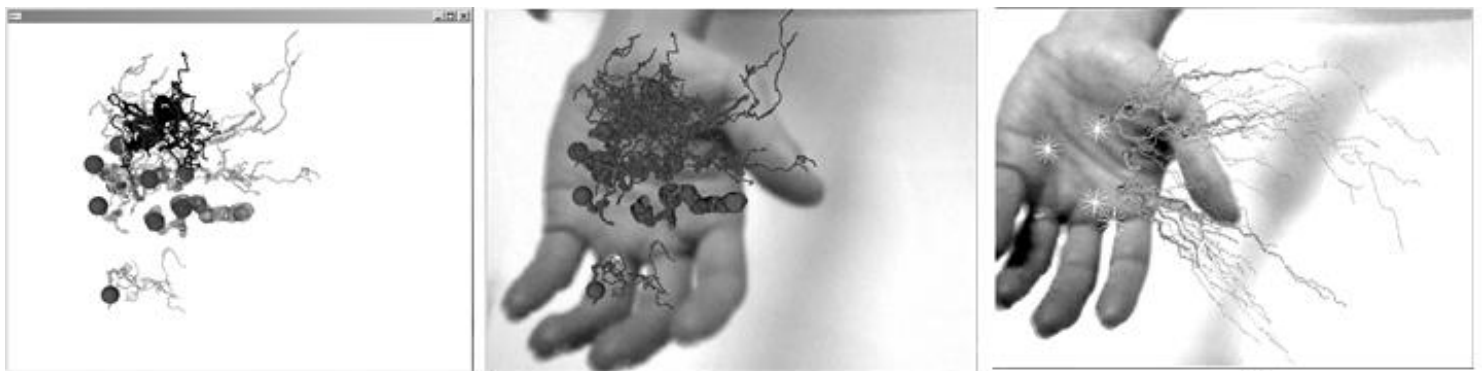


Figura 1 – Diagrama de classes do sistema.



(a) (b) (c)  
**Figura 2 – Formas de visualização. (a) Linhas. (b) Esferas. (c) Extrusão.**



(a) (b) (c)  
**Figura 3 - Visualização dos resultados. (a) Com extrusão e transparência. (b) Visualização de uma estrutura neural com realidade aumentada. (c) A mesma estrutura em b com campos de repulsão.**