

Utilização de L-Systems para Construção de Cenas Virtuais e Análise de Desempenho em Sistema de Realidade Virtual Distribuída

Autores

Luciano Augusto Rabello Vilela

Orientador

Regina Celia Coelho

Apoio Financeiro

Pibic

1. Introdução

Lindenmayer System (L-System) é um tipo particular de sistema simbólico dinâmico que dá uma interpretação gráfica a uma gramática. Possui esse nome como homenagem a seu criador, o biólogo A. Lindenmayer, que os criou em 1968 para aplicações no campo da botânica (LINDENMAYER, 1968).

Os L-Systems utilizam como técnica básica a reescrita de símbolos, criando, assim, objetos pela interação de um conjunto de regras ou produções para substituir partes de uma geometria inicial simples. Diferença desta gramática em relação às gramáticas de Chomsky é que nas L-Systems as produções são aplicadas em paralelo, ou seja, substituindo simultaneamente todas as letras de uma dada palavra, enquanto que nas gramáticas de Chomsky, as produções são aplicadas seqüencialmente. Esse sistema de reescrita permite definir objetos complexos por trocas sucessivas em partes de objetos mais simples.

A geração de imagens tridimensionais com o uso de Gramáticas L-Systems proporcionou uma grande variedade de aplicações, não somente em botânica, mas também em diversas outras aplicações, como para o reconhecimento de padrões, na geração e representação de neurônios, no desenvolvimento de células, na construção de cidades, e em muitos outros tipos de aplicações.

2. Objetivos

Os objetivos iniciais do trabalho foram: a construção de um *parser* para gramáticas L-systems estocásticas e não estocásticas, a interpretação visual dessas gramáticas, a distribuição do sistema e a realização de testes de desempenho.

Dos itens acima, a distribuição do sistema e teste de desempenho não foram desenvolvidas. No entanto, foram incluídas análises das otimizações do grafo de cena para mostrar as imagens geradas pelas L-systems. Estas etapas não desenvolvidas não prejudicou a conclusão do projeto, uma vez que o objetivo principal era conseguir otimizar a quantidade de objetos na cena. Estes resultados são de grande relevância porque o projeto mãe ao qual este projeto de iniciação científica está vinculado gera uma estrutura neural morfológicamente realista utilizando L-Systems para incluir cada célula na estrutura. Isto resulta na utilização de várias L-Systems (uma para cada célula) e numa quantidade de objetos muito grande, o que prejudica a

interação do usuário com a cena, uma vez que a renderização fica muito lenta. O projeto mãe já está sendo executado em um ambiente de realidade virtual distribuída e com os resultados alcançados deste projeto de iniciação científica, será possível melhorar a interação do usuário com a cena devido à otimização resultante deste projeto que será acrescentada ao projeto mãe.

3. Desenvolvimento

Para que a geração de cenas em um ambiente bi ou tridimensional e a construção do parser fossem possíveis, foi necessário, em primeiro lugar, o entendimento do funcionamento das Gramáticas L-Systems, cuja principal referência utilizada para este fim foi o livro *The Algorithmic Beauty of Plants* dos autores Aristid Lindenmayer e Przemyslaw Prusinkiewicz (Lindenmayer, Prusinkiewicz, 1989).

Para a construção de uma estrutura de dados (no caso, do tipo pilha) que fosse capaz de realizar todas as substituições e iterações que as Gramáticas exigem, foi utilizado como ferramenta de programação o Visual C++ 2005 da Microsoft.

A notação EBNF (*Extended-Backus-Naur-Form*) mostra de um modo geral o que o *parser* é capaz de reconhecer. Uma Gramática livre de contexto nesta notação é representada pela quádrupla $\{N, T, S, P\}$, sendo N um conjunto de não terminais, T um conjunto de terminais, S o axioma (tem que ser um membro de N) e P é um conjunto de regras produções que mapeia os elementos de N para T. Isto resulta em:

N = Regra, Predecessor, Sucessor, Símbolo, Cadeia}.

T = {+, -, <, >, &, ^, [,], (,), |, \$}

S = {<Regra>}

As regras de produção são definidas como:

<Axioma> ::= [{<Operador>}] <Símbolo>

<Regra> ::= <Predecessor> " = " <Sucessor>

<Predecessor> ::= <Símbolo> [(probabilidade)]

<Sucessor> ::= {<Cadeia>}

<Cadeia> ::= <Símbolo> | "[< Cadeia > "]" | <Operador> <Cadeia>

<Operador> ::= + | - | & | ^ | < | > | "|

<Símbolo> ::= A-Z | \$ | "|

Para exemplificar o funcionamento *parser*, foi gerado um diagrama de transição de estados a partir da implementação. No diagrama de transição de estados da Figura 1 é representado o processo de análise sintática de uma Gramática L-System livre de contexto que pode ser determinística ou estocástica. Cada nova linha presente no arquivo de entrada é considerada uma nova entrada para o diagrama.

A legenda do diagrama de transição de estados da Figura 1 é mostrada a seguir.

S	Estado inicial
NT	Não terminal
WPRD	Análise de início das regras produção para o não terminal (<i>wait production</i>)

PRD	Análise de cadeia de produção
WPRB	Análise de início de probabilidade (<i>wait probability</i>)
INIREG	Análise de regra
INT	Análise de inteiros
WFLT	Análise de início de ponto flutuante
FLT	Análise de ponto flutuante
ERRCMT	Erro: chaves de comentário não balanceadas
ERREQ	Erro: sinal de igual ('=') esperado não ocorreu
ERRDIG	Erro: dígito esperado não ocorreu
ERRPRD	Erro: era esperado caracteres após o sinal de igual ('=')
#	Marcador para comentário. Após encontrar esse marcador, o restante da linha será ignorado.

As entradas são:

C	Caractere qualquer
PT	Ponto
AP	– Abre parênteses – “(”
FP	Fecha parênteses
EOL	– Fim de linha
BL	– Tabulação, espaço
EQ	– Sinal de igual
D	Dígito
OP	–

Para realizar a otimização da cadeia de caracteres e, conseqüentemente, da figura gerada, uma nova estrutura de dados foi adicionada ao projeto. Essa estrutura é formada por uma lista de níveis em que cada nó dessa lista aponta para uma lista de células e cada nó célula aponta para uma lista de rotações.

Na lista de níveis os nós são utilizados para armazenar o número do nível atual e a lista de células geradas em cada nível. Na lista de células os nós guardam a quantidade de caracteres ‘F’ a ser considerada para a criação de cada objeto (essa quantidade indicará a altura de cada cilindro) e a lista de rotações. A lista de rotações, por sua vez, é responsável por colocar na ordem correta as rotações que devem ser realizadas num determinado objeto, sendo que para isso, cada nó armazena o ângulo e o eixo em que a rotação deve ser realizada.

Operadores

A alimentação dessa nova estrutura se dá pela manipulação da cadeia de caracteres original gerada pelo *parser*, que é feita da seguinte forma: para cada símbolo lido na cadeia original, um conjunto de instruções é executado e os dados produzidos são armazenados na estrutura. Por exemplo, se um símbolo '[' é encontrado, o nível atual é incrementado e caso este nível ainda não exista na lista, ele é criado e este se torna o nível atual. Se o próximo símbolo for '+', por exemplo, e o número de células desse nível (nível atual) é igual à zero, então uma nova célula com uma rotação no eixo x é criada.

A otimização da cadeia é feita primeiramente, somando todos os ‘F’ seguidos, isto é, que não tenham quaisquer outros símbolos entre eles. Uma otimização semelhante é feita também quando existe caracteres ‘F’ separados por ramificações (‘[’ e ‘]’), desde que antes de cada ‘F’ não exista caractere relacionado a ângulo.

Outra otimização da cadeia também pode ocorrer em alguns casos em que ela tenha seus caracteres ‘F’ separados por rotações. Se dois caracteres ‘F’ estiverem separados por rotações que se anulam, como por exemplo, “F+-F”, as rotações são eliminadas e a seqüência “FF” será representada como um único objeto. Rotações que não se anulam também são eliminadas da cadeia caso a soma delas for 360° ou múltiplo de 360° sobre o mesmo eixo de rotação.

4. Resultados

O primeiro resultado do projeto foi a construção de uma interface gráfica para a visualização dos resultados das gramáticas e interação do usuário com a cena. Nesta interface, é permitido ao usuário interagir com cena gerada realizando operações geométricas (rotação e translação), operações de visualização (rotação da fonte de luz, visualização do objeto na forma sólida ou aramada), manipulação de arquivos (arquivos relacionados à definição das gramáticas) e definição de parâmetros, como número de interações e alguns parâmetros do cilindro. A interface apresenta 6 botões relacionados com operações geométricas, sendo eles: um para rotacionar o objeto, um para rotacionar a fonte a luz, um para transladar nos eixos X e Y, e 3 para transladar em somente um dos eixos (X, Y ou Z). A rotação da imagem também pode ser feita clicando e arrastando o *mouse* diretamente sobre ela.

O segundo resultado está relacionado às otimizações realizadas. Para avaliar os resultados obtidos na otimização das gramáticas não estocásticas, foram selecionadas 27 gramáticas desse tipo. Esta avaliação foi feita calculando-se o percentual de otimização e o percentual de ganho obtido para cada iteração realizada de cada gramática.

O percentual de otimização foi calculado dividindo o número de objetos (cilindros) após a otimização pelo número total de objetos caso a cena fosse construída sem a otimização. O percentual de ganho é calculado subtraindo o valor do percentual de otimização de um.

Feito isso, foi calculado a média e o desvio padrão dos percentuais de ganho obtido nas iterações de cada gramática.

Foi observado que depois da 4^a. ou 5^a. interação, o valor do percentual de ganho praticamente se estabiliza.

Com os cálculos dos valores das médias e desvios padrão de todas às vinte e sete gramáticas, obteve-se o gráfico da figura 2 que representa a média do percentual de ganho e desvio padrão para cada gramática. Pelo gráfico, pode-se notar que, na maioria das gramáticas, o desvio padrão dos percentuais de ganho de cada gramática foi menor que 2%, mesmo considerando que nas interações iniciais o ganho é muito menor que após 4 ou 5 interações. Pelo gráfico é possível verificar que 13 gramáticas apresentaram ganhos médios de quase 50% ou mais, sendo que 3 dela tiveram ganhos maiores que 70%.

Na avaliação das gramáticas estocásticas, foram escolhidas 10 gramáticas diferentes. Cada uma das dez gramáticas foi executada cem vezes para cada iteração, começando com uma única iteração até no máximo dez. Isso foi necessário já que com uma mesma gramática estocástica e um mesmo número de interações podem-se obter infinitos resultados diferentes. Cada vez que a gramática é utilizada, devem ser feitas escolhas de quais regras serão utilizadas a cada substituição dependendo da probabilidade de ocorrência de cada regra. Para cada interação foram obtidos o número de objetos a serem gerados sem otimização e o número de objetos com a otimização; com esses dados, foram calculados o percentual de otimização e o percentual de ganho para cada execução da gramática. Tendo estes dados, foi calculada a média e desvio padrão para cada interação e gerado um gráfico para cada gramática.

Foi observado que em todas as gramáticas, o desvio padrão decresce de acordo com o aumento do número de iterações, o que já era previsível. Normalmente o desvio padrão nas primeiras interações é alto devido ao limitado número de escolhas randômicas pelas regras de substituição da gramática a cada execução. À medida que o número de interações aumenta, o número de escolhas também aumenta, fazendo com que as regras sejam escolhidas com uma frequência semelhante, apesar da escolha ser randômica.

Foi calculado também a média das médias do percentual de ganho, e o desvio padrão dessas médias para cada gramática. Com estes valores foi possível avaliar, de uma forma geral, como foi o ganho para este tipo

de gramática. O gráfico resultante deste cálculo pode ser visto na figura 3. Observando este gráfico, pode-se dizer que o desvio padrão dos percentuais de ganho da maioria das gramáticas foi menor que 2%. Para estas médias foram considerados apenas os resultados obtidos a partir da quarta interação, uma vez que a variação dos resultados é muito grande com poucas interações. Apenas duas tiveram desvio padrão maior que 2%, chegando no máximo a 4,86%.

Estes resultados mostram que por menor que seja o ganho, este tipo de otimização diminui consideravelmente a quantidade de objetos na cena, considerando que cada gramática pode ser utilizada várias vezes para gerar toda a cena.

5. Considerações Finais

O desenvolvimento de um programa capaz de gerar cenas bi ou tridimensionais por meio da interpretação de Gramáticas L-Systems foi essencial para o estudo de otimização da cena, uma vez que estas gramáticas permitem a geração de cenas com diferentes níveis de complexidade.

Os ganhos apresentados nos resultados demonstram a eficácia do método. É muito raro encontrar gramáticas em que o ganho não seja acima de pelo menos 30%. Considerando que numa cena completa, como por exemplo, uma paisagem, várias gramáticas são utilizadas, o ganho na construção de toda a cena é realmente muito representativo.

Muito foi feito para que se pudesse ter um ambiente interativo e de fácil uso pelo usuário e gerar imagens otimizadas. O maior diferencial desse projeto é sem dúvida a otimização alcançada da interpretação visual das L-systems com a proposta implementada.

Embora a análise de desempenho em um sistema de realidade virtual distribuída não tenha sido realizada, foi incluída uma análise da otimização da cena, que compensa a análise sugerida originalmente. Consideramos que a análise da otimização da cena estaria mais de acordo com o projeto, uma vez que diminuindo a quantidade de objetos que devem ser gerados na cena, também diminuimos a quantidade de dados que devem ser enviados para a rede para que esta possa ser visualizada em outras máquinas. Sendo assim, obtendo bons resultados com a otimização do grafo de cena, e como consequência uma boa otimização do tráfego da rede.

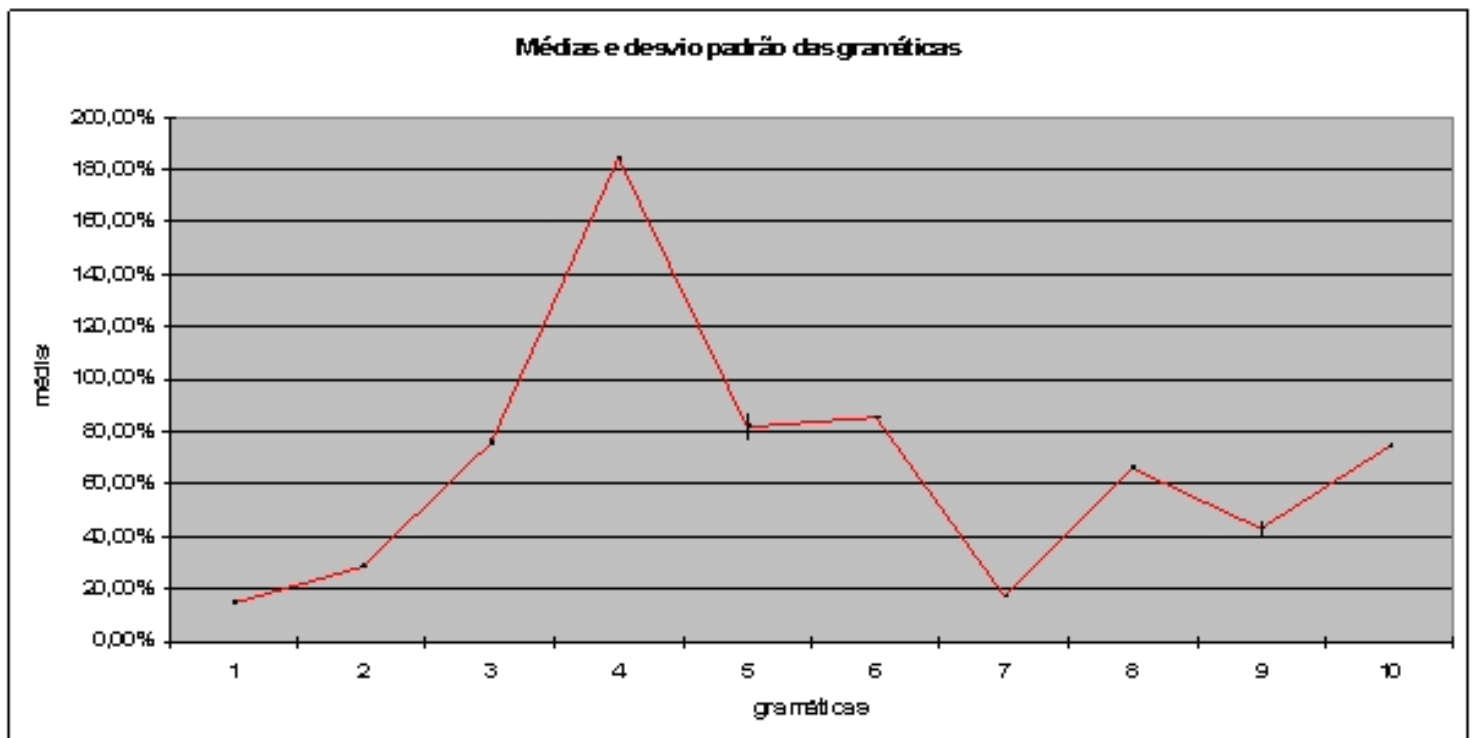
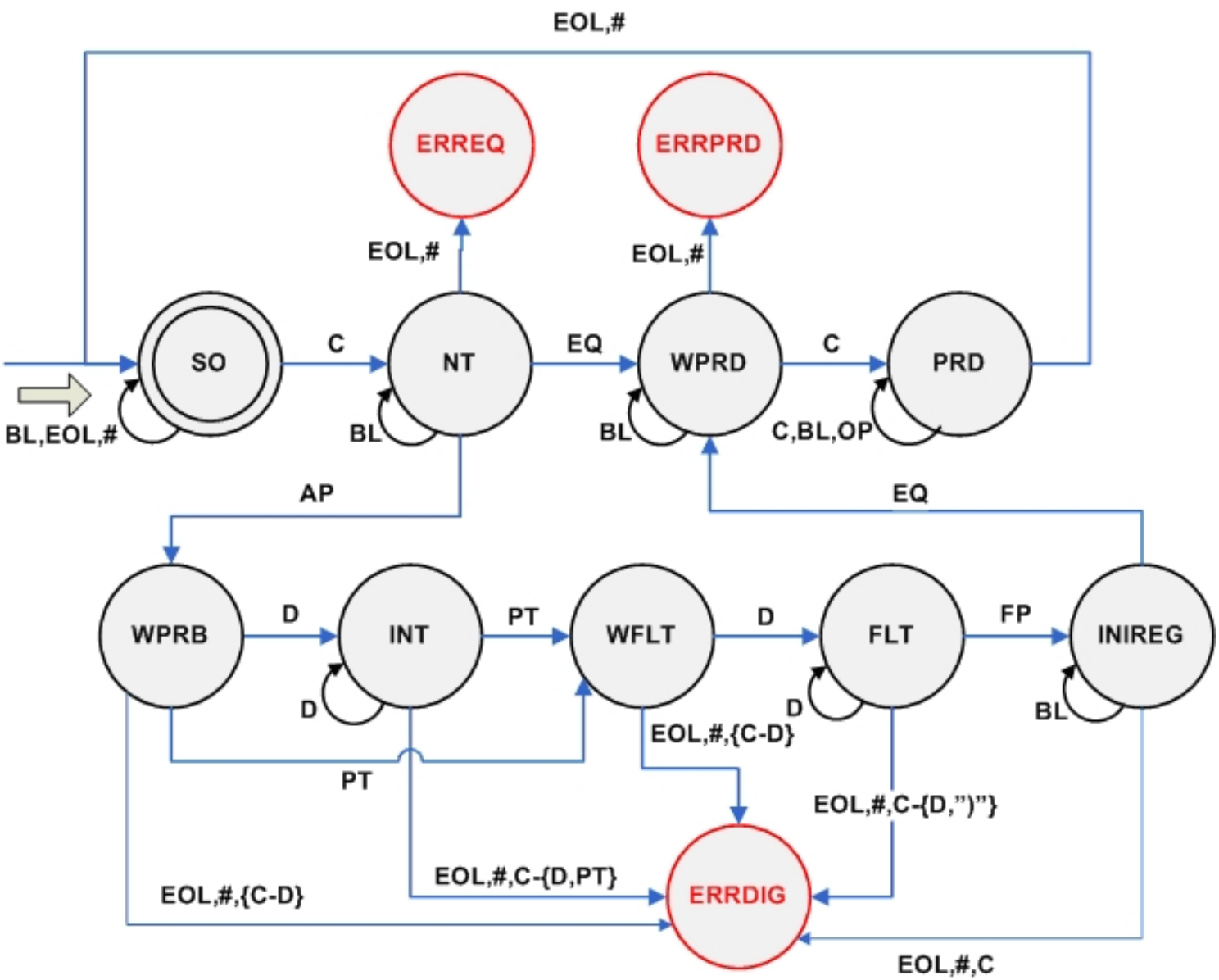
Referências Bibliográficas

PRUSINKIEWICZ, P. Animation of Plant Development, Computer Graphics Proceedings. In SIGGRAPH 93, Annual Conference Series, ACM, 1993.

LINDENMAYER, P. The Algorithmic Beauty of Plants. In Springer, 1990.

HART, C. The Object Instancing Paradigm for Linear Fractal Modeling. In Proceedings of Graphics Interface, 1992, Vancouver, BC.

Anexos



Média e Desvio Padrão das gramáticas não estocásticas

