

Relatório de Estudos

MAC 5701 - Tópicos em Ciência da Computação

Luis Carlos Yano Endo

Orientador: Antônio Elias Fabris

Professora: Yoshiko Wakabayashi

26 de junho de 2002

Sumário

1	Introdução	3
1.1	Tema da Dissertação de Mestrado	3
1.2	Passo atual do desenvolvimento	4
2	Estudo de Técnicas Modelagem	5
2.1	Modelagem de Sólidos	5
2.1.1	Sweep Representation	6
2.1.2	Constructive Solid Geometry	6
2.1.3	Boundary Representation	8
2.1.4	Wireframe Systems	8
2.2	Técnicas Avançadas de Modelagem	9
2.2.1	Modelagem Avançada utilizando Splines	10
2.2.2	Modelagem Baseada em Gramática	10
2.2.3	Sistemas de Partículas	11
2.2.4	Modelagem baseada em Física	14
3	Simulação de Fenômenos Naturais	18
3.1	Modelagem de Plantas	18
3.1.1	L-Systems	18
3.2	Modelagem de Vegetação Rasteira	21
3.3	Slices	21
3.4	Modelagem do vento	22
3.5	Simulação de Ecossistemas	23
4	Técnicas para renderização em Tempo Real	24
5	Implementação	25
5.1	Renderização do Terreno	25
5.2	Distribuição de Plantas	25
5.3	Renderização de Grama	26
5.3.1	Adicionando propriedades Físicas para a grama	26
6	Resultados e Próximas Etapas	27
7	Conclusão	28

1 Introdução

Este relatório visa analisar e consolidar o material pesquisado ao longo do semestre a fim de auxiliar a elaboração da dissertação de mestrado. Para isso, serão explicados artigos e documentos estudados até o momento e sua relação com o tema da dissertação de Mestrado.

A docente responsável pela disciplina é a professora Yoshiko Wakabayashi. O trabalho de pesquisa está sendo orientado pelos professores Antonio Elias Fabris e Carlos Hitoshi Morimoto (“co-orientador”). A área de pesquisa é Computação Gráfica.

O relatório foi organizado em seções que abordam temas diferentes da área de Computação Gráfica e que deverão ser utilizados no desenvolvimento da dissertação. Serão explicadas técnicas de modelagem estudadas e quais são mais adequadas para implementação da proposta de dissertação. Na subseção abaixo iremos expor o tema da dissertação de mestrado.

O estudo de simulação de fenômenos naturais no computador é um dos assuntos com maior relacionamento com a dissertação, neste relatório serão descritos artigos que tratam do assunto. A API utilizada para implementação inicial do sistema e técnicas para renderização de cenas em tempo real também são mencionadas.

No final do relatório mostraremos o que foi implementado e os resultados obtidos até o momento. Também serão listados os próximos passos de estudo e implementação para finalização do trabalho.

1.1 Tema da Dissertação de Mestrado

Minha dissertação de mestrado está ligada a área de Computação Gráfica, com ênfase no estudo de Simulação de Fenômenos Naturais e Animação Baseada em Física. Ambos assuntos são bem recentes na área de Computação Gráfica e estão tendo atenção especial da comunidade ligada a pesquisa de Computação Gráfica nos últimos anos.

O objetivo do trabalho será a geração de cenas que possam representar de forma realista e em tempo real paisagens de ecossistemas, gerando exemplos de elementos relativos aos ecossistemas brasileiros especificamente. O trabalho estará focado a geração de imagens de vegetação e sua animação em tempo real.

A modelagem e geração em computador de ecossistemas é uma tarefa muito complexa, pois envolve várias estudos ligados sobre biologia, mais particularmente um estudo ligado a área de ecologia, como cadeias alimentares, ciclos de nutrientes, equilíbrio de populações, entre outros. Por isso, a tese não deverá conter elementos como, por exemplos, animais o que complicaria muito a modelagem e simulação, além de envolver o estudo de outras áreas complexas como inteligência artificial. Apesar disso, o objetivo é estender este trabalho após a conclusão do mestrado para chegar a um resultado que possa representar um mini ecossistema.

Por este motivo estamos focando o trabalho para modelagem, renderização e animação de vegetais em tempo real, o que já é um assunto bastante complexo e recente na área. A movimentação dos vegetais seria resultado do contato de agentes naturais, como vento e chuva ou de agentes artificiais, como por exemplo uma bola de futebol. O contato destes agentes com os vegetais deverão gerar animações realistas, para isto devemos fazer o estudo de animações baseadas em física. Animação e Modelagem baseadas em física são áreas da Computação Gráfica que visam a modelagem e animação de cenas baseadas em leis da física, e são as técnicas atuais que geram os resultados mais próximos ao fenômeno real.

Outro assunto a ser estudado é como gerar cenas com muitos objetos e animá-las em tempo real. A cena pode ter um grande número de vegetais, temos que estudar técnicas para animação de todos vegetais de forma mais realista possível e além disso gerar um frame rate que deixe a animação suave.

1.2 Passo atual do desenvolvimento

Neste semestre foi iniciado o estudo de artigos relacionados ao tema da dissertação. Também teve início o planejamento e o desenvolvimento do simulador de ecossistemas. Neste semestre foi iniciada a implementação da vegetação, particularmente de vegetação rasteira.

Isto envolveu o estudo de técnicas de modelagem e geração de fenômenos naturais, este assunto vem sendo estudado há algum tempo na área, o que vem sendo estudado atualmente é como gerar cenas da natureza em tempo real e com uma animação realista. Para isso, estudamos diversos artigos relacionados a técnicas de modelagem e animação para encontrarmos uma solução adequada para o problema.

Os artigos estudados serão explicados nas próximas seções. O projeto e os resultados gerados até o momento serão mostrados no final do relatório. Iniciaremos a explicação com artigos relacionados as técnicas de modelagem estudadas.

2 Estudo de Técnicas Modelagem

A Modelagem é uma área fundamental em Computação Gráfica, nela são descritos os modelos em que os objetos serão representados para depois serem renderizados e se tornarem imagens. A modelagem trata do problema da criação, manipulação da geometria e topologia dos objetos gráficos no computador [7]. Diversas técnicas foram desenvolvidas nesta área pois é uma das áreas básicas da Computação Gráfica.

O início das pesquisas envolveu o estudo da técnica de modelagem mais apropriada para resolver o problema de modelagem e animação de plantas. Além disso, foi preparado material para apresentação de um seminário sobre técnicas de modelagem.

Para podermos estudar técnicas avançadas de modelagem é importante ter o conhecimento de técnicas básicas de modelagem de sólidos. Estas técnicas já haviam sido estudadas na implementação de meu projeto de Iniciação Científica, Sistemas Gráficos 3D com Antialiasing, realizado no IME-USP sob orientação do professor Antonio Elias Fabris, com duração de dois anos.

A preparação do seminário exigiu a revisão destes conceitos básicos de modelagem. A próxima seção irá explicar de forma resumida as técnicas de Modelagem de Sólidos. Em seguida iremos explicar técnicas mais recentes e avançadas e que serão utilizadas na implementação do simulador de ecossistemas.

2.1 Modelagem de Sólidos

A Modelagem de Sólidos é uma forma de se representar um objeto tridimensional com certas propriedades desejadas. Uma modelagem geométrica de sólidos é uma representação matemática completa e não ambígua de um objeto físico de forma que um computador possa processá-la.

Requicha [21] assume uma lista de propriedades desejáveis em uma representação de um sólido. O *domínio* da representação deve ser amplo o suficiente para permitir uma representação de um conjunto útil de objetos físicos. A representação não deve ser ambígua (representação *completa*), devemos saber exatamente o que está sendo representado, uma dada representação deve corresponder a um e apenas um sólido. Uma representação é *única* se é usada para gerar um dado sólido de apenas uma forma. Uma representação *precisa* permite o objeto ser representado sem aproximações.

A Modelagem de Sólidos atende muito bem ao problema de modelagem de objetos com uma forma bem definida. Em geral objetos manufaturados são bem modelados através destas técnicas. Objetos que não possuem uma geometria bem definida, como os relacionados a fenômenos naturais tais como nuvens, fogo, chuva etc., não podem ser bem modelados usando tais técnicas.

As técnicas de Modelagem de Sólidos são, em sua maior parte, antigas, divulgadas nas décadas de 70 e 80, porém continuam sendo utilizadas em muitos problemas de modelagem atuais. Daremos uma breve descrição de algumas técnicas de modelagem de sólidos e suas principais características. As técnicas descritas serão:

- Sweep Representation
- Constructive Solid Geometry
- Boundary Representation
- Wireframe Systems

Observação: foram estudados os livros de Foley *et al.* [5], Watt [25], Gomes e Velho [7] e Mantyla [11], além dos artigos de Green [8] e Requicha [21] para revisão de Modelagem de Sólidos, estes não serão citados diretamente nas subseções abaixo.

2.1.1 Sweep Representation

Nesta técnica o objeto (curva, superfície, volume) é gerado movendo um objeto geométrico (como um ponto, curva, superfície ou volume) por um caminho no espaço 3D. O movimento da trajetória pode ser uma translação ou uma rotação.

Por exemplo, podemos representar um cilindro movendo um círculo na direção de uma reta. Podemos gerar objetos com curvas, simplesmente fazendo o objeto se mover na trajetória de uma curva, exemplo curva de Bezier, ou um Spline. Geralmente, o termo gerador é usado para denotar o objeto, e o termo diretor para denotar a trajetória. Pode se criar uma grande diversidade de objetos usando essa representação.

Os objetos são particularmente difíceis de se modelar de forma eficiente nesta técnica geralmente, pois dependem muito da equação da trajetória, além do problema de auto-intersecção de trajetória que implica em ambiguidade de representação. Operações booleanas sobre objetos gerados com esta representação não geram um único objeto em sweep representation, isto acontece com o CSG como veremos na próxima subseção.

Note que os objetos gerados não são necessariamente tridimensionais, um objeto 2D movendo-se sobre o plano que o contém gera uma forma 2D. Isto é muito utilizado em técnicas de stroking (aumentar o tamanho do pincel utilizado para se desenhar uma curva).

Duas operações são bastante utilizadas para geração de imagens mais realistas na Sweep Representation. A operação **Warp** é usada para provocar rugosidades ou deformações nas formas. A operação **Blend** é usada para provocar transições suaves ao se juntar dois objetos criados utilizando-se esta representação.

2.1.2 Constructive Solid Geometry

Este método de modelagem define sólidos complexos a partir da composição de sólidos simples (Primitivas) e operações booleanas definidas. Nesta representação os objetos estão ordenados em uma árvore binária, onde as folhas são as primitivas, e os nós o resultado de uma operação booleana dos filhos. A raiz representa o objeto final obtido.

É uma técnica bastante simples de ser utilizada e implementada, porém o número de objetos que podem ser representados por esta técnica é bastante reduzido. As primitivas mais comuns neste sistema são: **Cilindro**, **Esfera**, **Cone**, **Toro** e **Poliedros**. As Operações Booleanas geralmente são: **União**, **Diferença** e **Intersecção**.

Este método é a generalização da técnica de *Cell Decomposition*, onde um sólido pode ser representado como soma ou união de um conjunto de células, nas quais é dividido. Um caso mais específico desta técnica é a *Spatial-Occupancy Enumeration* onde o sólido é decomposto em células idênticas e arranjadas em um grid regular fixado. Esta técnica é bastante usada em aplicações biomédicas onde os dados são obtidos de tomografia computadorizada. A célula neste caso é denominada de voxel, em analogia ao pixel. O tipo de célula mais comum neste caso é o cubo.

A estrutura de dados utilizada nesta técnica é a de octrees, derivada das quadrees, que são bastante conhecidas e utilizadas. A idéia por trás destas representações é a de divisão e conquista. Um cubo inicial é subdividido cubos de oito partes iguais, se o conteúdo do cubo não é homogêneo, isto é, não está totalmente dentro nem totalmente fora do objeto, é feita uma nova subdivisão neste cubo. A idéia é repetida até que se tenha todos cubos homogêneos.

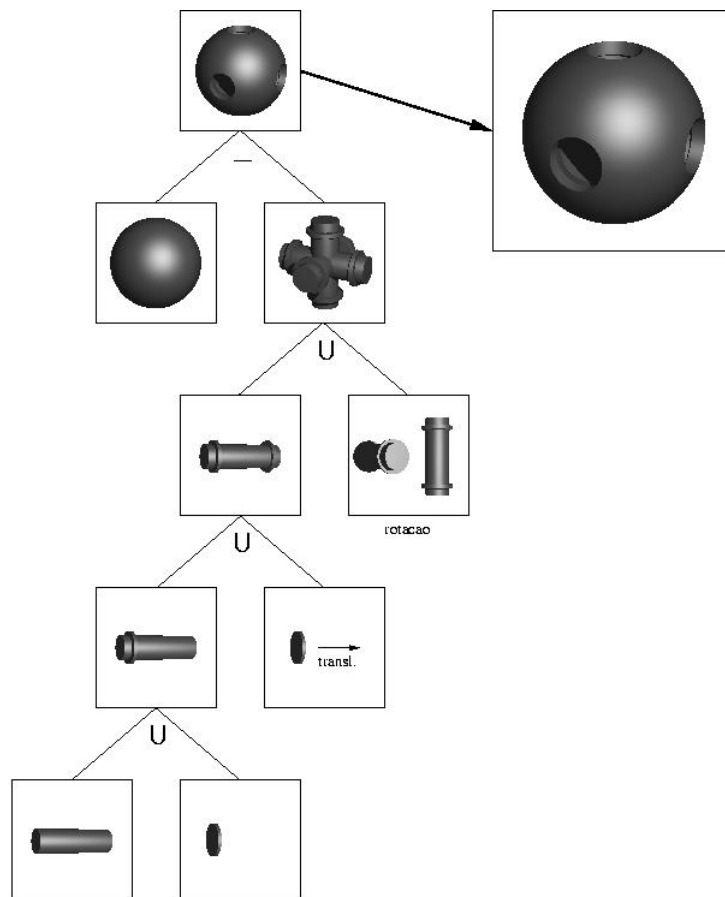


Figura 1: Árvore gerada para geração de um objeto CSG, modelado pelo projeto de Iniciação Científica “Sistema Gráfico 3D com Antialiasing”

2.1.3 Boundary Representation

Na Boundary Representation os objetos são descritos através da superfície de suas bordas. A borda de um sólido separa pontos de dentro do sólido dos pontos fora do sólido. Nela ocorrem os contatos com outros sólidos, e é ela que determina a aparência do objeto, com propriedades como forma, cor e textura.

Para se determinar a borda são necessárias informações sobre os vértices, arestas e faces da superfície de fronteira. Requicha [21] sugere que as faces que definem a borda satisfaçam as condições mínimas:

- Número de faces finito
- A união de todas as faces de um objeto definem sua borda
- Uma face deve ter área finita e ser homogênea

A borda de um objeto pode ser segmentado em faces, arestas e vértices de formas ilimitadas, não há uma representação única da borda. Por exemplo um cubo pode ser decomposto em 6 quadrados, 24 quadrados, ou até em faces triangularizadas. Isto se torna ainda mais difícil quando a superfície que se quer representar é curva, determinar o que constitui uma face nesse caso é bem complicado. É necessária uma estrutura com informações suficientes para que não haja confusão na representação do objeto.

Se representarmos cada face como uma lista de vértices, estaremos duplicando informações em locais onde duas faces compartilham uma aresta. Utilizando uma representação chamada *Winged-Edge* evitamos algumas duplicações de coordenadas compartilhadas por duas faces. Nesta estrutura de dados cada aresta $A1$ é representada com ponteiros para seus dois vértices ($V1$ e $V2$), para suas duas faces ($F1$ e $F2$) e para as outras quatro arestas que saem dos vértices ($A2$, $A3$, $A4$ e $A5$, essas quatro arestas são as asas que dão o nome para a estrutura). As duas arestas que estão em sentido horário com relação a $A1$ (considerando a orientação $V1 \rightarrow V2$) pertencem a face $F1$ e as duas faces em sentido anti-horário pertencem a $F2$.

Podemos usar operações booleanas para combinar novos b-reps, por exemplo para união, as faces que ficam internas no objeto final podem ser descartadas, novas arestas são definidas nos locais onde há intersecção dos objetos. O mesmo vale para intersecção e diferença.

Podemos representar praticamente qualquer objeto utilizando esta técnica. Porém, não se trata de uma representação exata, mas sim uma aproximação do objeto, assim como a técnica de Spatial-partitioning. Quanto maior o número de polígonos utilizados nesta aproximação mais próxima à aparência real o objeto é representado, porém quanto maior o número de polígonos mais memória e tempo se gastará na geração da imagem.

Utilizando-se algoritmos de shading (distribuição de cores sobre um polígono) obtemos imagens bastante realistas com poucos polígonos por objeto, aliando-se isso à grande capacidade de geração de polígonos atual das placas aceleradoras de vídeo, faz com que a B-Rep seja uma técnica bastante utilizada nos sistemas atuais.

2.1.4 Wireframe Systems

O Modelo de Wireframe representa apenas as arestas de um objeto, e consiste apenas de pontos, linhas e curvas. Assim é um modelo de geração rápida de imagens.

Na tentativa de usar um sistema Wireframe puro para representação de sólidos tridimensionais há quatro problemas:

- Possibilidade de criação de modelos ambíguos

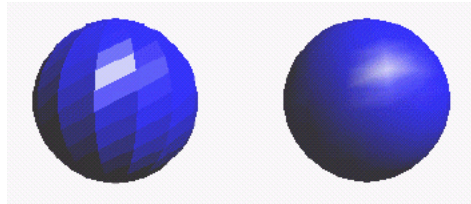


Figura 2: Esferas modeladas utilizando B-rep, a segunda modelada com o mesmo número de polígonos mas com a técnica de shading.

- Possibilidade de criação de objetos sem sentido
- Falta de coerência gráfica ou visual

Para resolver esses problemas podemos tentar eliminar as linhas escondidas, porém para formas complexas o nível de detalhes aumenta rapidamente, o que complica a representação utilizando essa técnica.



Figura 3: Bule de chá representado utilizando WireFrame

2.2 Técnicas Avançadas de Modelagem

As técnicas apresentadas anteriormente funcionam perfeitamente em um mundo de objetos geométricos e de formas bem definidas. Considerando-se a modelagem de fenômenos naturais, tema abordado neste mestrado, não podemos representar a maioria destes fenômenos eficientemente e satisfatoriamente utilizando esses modelos geométricos.

Vamos descrever agora algumas técnicas avançadas de modelagem que podem ser utilizadas para modelagem de fenômenos naturais. Nestas técnicas, temos uma representação de uma classe de objetos mais ampla, por um modelo com parâmetros facilmente ajustáveis e intuitivos.

Outra vantagem desses métodos é que a maioria dos métodos que serão apresentados foram desenvolvidos para modelos dinâmicos ao invés de modelos estáticos, possibilitando modelar crescimento e mudança da forma do objetos. A geração de imagens em sequência combinadas produzem uma animação automática do objeto.

As técnicas avançadas de modelagem tentam ir além do que pode ser representado pela modelagem geométrica, permitindo a modelagem simples de fenômenos complexos. Desta forma podemos representar uma grande quantidade de objetos por um simples modelo que podem ser ajustados facilmente e intuitivamente alterando-se alguns parâmetros. [5]

Nas próximas subseções iremos descrever alguns modelos avançados de modelagem. Os modelos escolhidos para o estudo foram os mais relacionados e utilizados para modelagem de plantas. Em geral, a principal característica destes modelos, como já foi mencionado, é que mudanças simples na especificação e definição dos parâmetros podem resultar em mudanças drásticas na forma. Além disso, os modelos geralmente são **procedurais**, que são modelos que descrevem objetos que podem interagir com eventos externos ou com o ambiente para se auto modificarem.

Serão descritos os seguintes modelos avançados de modelagem:

- Modelagem utilizando Splines
- Modelagem Baseada em Gramática
- Sistemas de Partículas
- Modelagem Baseada em Física

Descrições sobre as técnicas avançadas de modelagem foram obtidas estudando o artigo de Ebert [4] e o livro de Foley *et al.* [5]. Artigos específicos sobre as técnicas de modelagem também foram estudados, e serão citados nas respectivas seções.

2.2.1 Modelagem Avançada utilizando Splines

Splines são curvas paramétricas que são definidas através de pontos de controle, muito estudadas em cursos básicos de Computação Gráfica e Análise Numérica. Uma rede (mesh) de curvas pode gerar uma superfície tridimensional, logo um mesh de splines pode ser usado para modelagem.

Esta técnica é capaz de gerar resultados impressionantes, o objeto final é uma união de vários patches de splines, podendo ser renderizados utilizando métodos convencionais de renderização (ex: ray-tracing).

A modelagem do objeto deve ser feita alterando-se os pontos de controle que definem as curvas splines. Pode-se criar mais pontos de controle a partir dos já existentes para melhor modelar o objeto.

Pode-se modelar praticamente qualquer forma definida com esta técnica, porém geralmente os pontos de controle devem ser definidos manualmente para geração da forma desejada, o tempo de criação da imagem também é um problema pois as curvas podem ser tão complexas quanto quisermos.

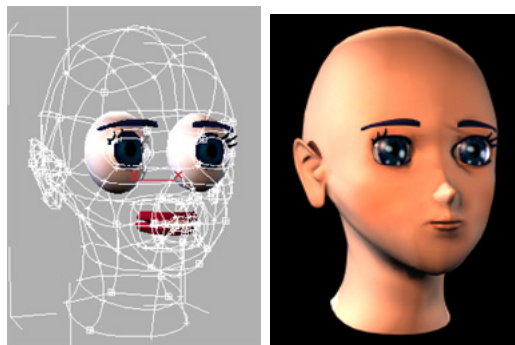


Figura 4:

2.2.2 Modelagem Baseada em Gramática

Os modelos baseados em gramática foram inicialmente criados para descrever a estrutura de certas plantas na década de 80, baseados em uma técnica descrita por Lindenmayer em 1968. Tais modelos passaram a ser conhecidos como L-Systems, técnica muito utilizada para algoritmos de branching como veremos mais a frente.

A idéia básica dos modelos baseados em gramática é desenvolver linguagens descritas por uma gramática que consistem de uma coleção de regras de produções, que serão aplicadas na construção de objeto.

Para melhor ilustrar a técnica vamos criar um exemplo. Seja uma gramática com o alfabeto A, B, [,] e duas regras de produção:

regra 1. $A \rightarrow AA$

regra 2. $B \rightarrow A[B]AA[B]$

A cada passo do algoritmo estas regras são aplicadas para todos elementos da cadeia inicial. O exemplo abaixo ilustra a aplicação das regras dado um passo inicial somente com o elemento B.

iteração 0: B

iteração 1: A[B]AA[B]

iteração 2: AA[A[B]AA[B]]AAAA[A[B]AA[B]]

Se considerarmos A como sendo um galho com direção vertical e B um galho de direção horizontal, verificamos que a partir do passo inicial B, obtemos um crescimento de uma árvore ou arbusto, se a cada iteração aplicamos as regras 1 e 2, na árvore anterior.

Embora o melhor exemplo de aplicação desta técnica ser o de modelos de plantas, qualquer domínio onde os objetos sejam modelados exibindo regularidade, pode ser adaptado ao modelo baseado em gramática. Prusinkiewicz [18] utiliza este modelo para modelar formas de vegetais complexas, e conchas de ostras. Vamos explorar melhor o tema L-Systems na seção sobre Simulação de Fenômenos Naturais, discutiremos o trabalho de alguns artigos lidos e a importância e resultados obtidos por esta técnica.

2.2.3 Sistemas de Partículas

O Sistema de Partículas é uma aproximação para se modelar objetos cujo comportamento não podem ser descritos em termos de sua superfície. A modelagem dos objetos é feita utilizando-se uma nuvem de partículas primitivas, que juntas definem o volume e aspecto de um objeto nebuloso. Essas partículas seguem uma certa regra: são geradas, mudam de aspecto ao longo do tempo e se extinguem. As partículas também podem se mover de acordo com alguma lei estocástica.

A dinâmica proporcionada pela técnica de Sistemas de Partículas, não era possível com métodos clássicos baseados em representação de superfícies. Este sistema é amplamente utilizado para modelar fogo, neblina, fogos de artifício, árvores, grama, explosões e fluidos em geral.

Modelo Básico de um Sistema de Partículas:

- Geração da Partícula
- Atributos (velocidade, posição, cor, forma...)
- Dinâmica (mudança de cor, direção do movimento, tamanho...)
- Extinção

A cada frame da sequência gerada em uma animação, devem ser seguidos os seguintes passos:

- (1) geração de novas partículas pelo sistema.
- (2) parâmetros individuais são associados a cada nova partícula recém gerada.
- (3) as partículas que completaram seu “tempo de vida” são extintas.
- (4) as partículas restantes são movimentadas e transformadas de acordo com a dinâmica descrita nos seus atributos.
- (5) as partículas são renderizadas em um frame buffer.

Para se renderizar partículas deve se usar técnicas de renderização diferentes, usar a técnica de ray-tracing seria impossível, pois cada raio traçado para renderização de um pixel poderia interceptar milhares de partículas. Para se renderizar uma partícula basta adicionar seu ponto de iluminação, ou segmento de reta caso esteja se movendo, na imagem final. Cada pixel acumula valores de cada partícula logo todas as partículas são responsáveis pela imagem final obtida. Observe que a renderização desta forma é bastante cara computacionalmente. É uma técnica poderosa, mas para cada nova aplicação uma nova técnica de renderização pode ser necessária.

Para a animação das partículas, tem que se considerar variáveis como: tipo, idade, eixo espiral, cor, opacidade e tamanho. Outras informações podem ser: velocidade inicial, idade para morrer, aceleração, entre outros. Para maior realismo as partículas podem ser dotadas de propriedades físicas.

Alguns artigos lidos sobre esta técnica serão brevemente discutidos nos parágrafos abaixo.

Particle Systems – A Technique for Modeling a Class of Fuzzy Objects Este artigo de Reeves [19] introduz o conceito de Sistemas de Partículas como um método para se modelar objetos sem aspecto definido. O modelo é capaz de representar movimento, mudanças de forma e dinâmica o que não era permitido com representações clássicas baseadas em superfícies. O artigo discute a geração de partículas por processos estocásticos, define atributos para cada partícula gerada (posição, velocidade, tamanho, cor e transparência iniciais, forma e tempo de vida), define a dinâmica (mudança dos atributos descritos acima), discute também a extinção e renderização das partículas. O texto sobre Sistemas de Partículas acima foi escrito baseado, praticamente, neste artigos assim como todos trabalhos que utilizam esta técnica de modelagem. No final do artigos são exibidos os resultados com imagens de explosões, fogo, plantas e fogos de artifício.

Particle Animation and Rendering Using Data Parallel Computation Neste artigo de Sims [22] são apresentadas usadas para animação e renderização de sistemas de partículas em computadores paralelos. A animação baseada em física com utilização de forças podem criar animações complexas e realistas, porém torna o sistema ainda mais lento. Para animação são definidas operações de posição, velocidade e aceleração das partículas, o autor deixa claro que utiliza aproximações para se evitar equações complexas e caras da física. Para renderização, Sims define a cabeça e o rabo da partícula, a posição cabeça é animada enquanto a posição rabo segue a caebça com a função de borrar um rastro da partícula na imagem. Os resultados do artigo mostram a animação de neve, vento, água e fogo utilizando os operadores descritos, nestes exemplo pode-se ver a alteração dos atributos das partículas no decorrer do tempo, assim como alguns operadores definidos por Sims como amortecimento (exemplo neve ao tocar no chão) e saltos (água ao atingir um obstáculo). Para modelagem de plantas cada galho pode ser uma partícula colocada aleatoriamente no tronco, e cada galho pode ramificar ou crescer de acordo com uma regra probabilística. Cada modelo tem sua lógica, por exemplo a modelagem de fogo é bem diferente da de plantas, as partícula estão em uma estrutura de árvore, uma partícula tem partículas filhas.

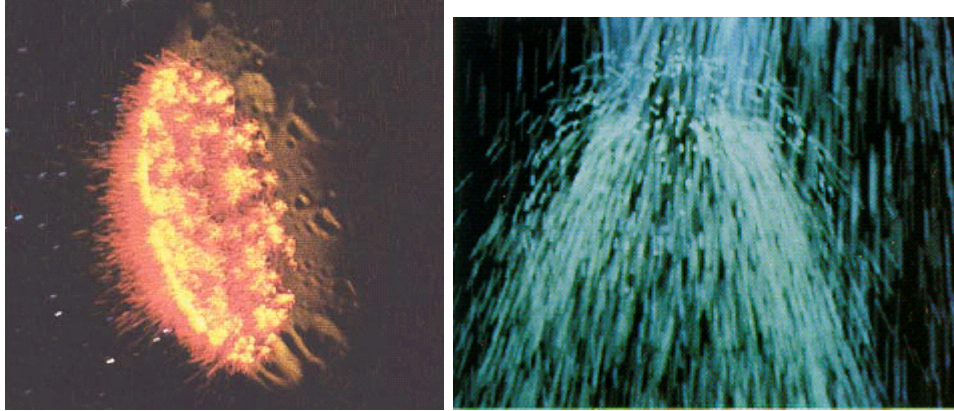


Figura 5: Partículas dos artigos de Reeves(exploração do Star Trek) e Sims (cachoeira)

Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems Outro artigo de Reeves [20] detalha a técnica de modelagem estocástica, shading, sombreamento e texturas utilizada nos sistemas de partículas, mostrando como gerar árvores, grama e outros vegetais utilizando sua técnica de modelagem. Reeves mostra que pode usar sistemas de partículas para modelar objetos complexos e estruturados. As partículas utilizadas para modelar fenômenos naturais como árvores e grama não são independentes como no caso de fogo e água por exemplo. O artigo mostra como modelar e renderizar árvores e grama utilizando sistemas de partículas. As árvores são geradas a partir do tronco com um algoritmo de construção de galhos (branching) recursivo, os galhos gerados a cada passo da recursão herdam os parâmetros de seus pais, mas é feito um controle para que os “sub-galhos” sejam mais finos e próximos ao pai. O algoritmo termina quando os galhos gerados tem uma grossura determinada, folhas são adicionadas a galhos que não tem filhos. Uma árvore gerada tem cerca de 1 milhão de partículas, logo o tempo de construção de uma árvore é alto. As gramas são geradas com parâmetros posição, orientação, área, densidade e tipo, as partículas e suas trajetórias geram cada folha de um tufo de grama. O paper descreve técnicas para shading e determinação de superfícies visíveis para melhorar a performance e aparência das cenas geradas, mas este não é o objetivo deste estudo. A modelagem do vento para a animação também é descrita, porém vamos deixar este assunto para a seção de Simulação de Fenômenos Naturais. A figura 6 mostra os resultados obtidos por Reeves.



Figura 6: Paisagens Naturais Modeladas utilizando Partículas

2.2.4 Modelagem baseada em Física

A Modelagem baseada em Física usa propriedades da Física para determinar o aspecto, forma e movimento dos objetos. Os objetos gerados apresentam comportamento descritos por leis da física, isto proporciona maior realismo ao movimento do corpo. A matemática utilizada neste tipo de modelagem é bastante pesada, o que aumenta o custo computacional do sistema.

Para ilustrarmos o poder desta técnica vamos criar um exemplo simples utilizando uma bola de plástico modelada por esta técnica. Suponha que bola seja lançada com uma certa velocidade inicial e por ação da força da gravidade vai caindo até tocar no chão. Acontece uma detecção da colisão que causa a deformação do formato da bola. Após o contato, a bola deve ter a capacidade de retomar seu formato original e deve ser calculada a direção e velocidade que a bola vai seguir. A animação segue até que a velocidade da bola seja zero e seu movimento termine.

A cena descrita parece simples mas sua implementação requer estudos de teorias complexas, que serão descritas nesta subseção. Neste exemplo da bola não temos nenhuma utilidade prática (a não ser o estudo de movimento em física), porém podemos usar esta técnica para fazer simulações realistas por utilizarmos conceitos da física. Exemplos de aplicações interessantes na prática seriam a colisão de um carro em uma parede, ou outro em carro e o que acontece com os objetos dentro do carro, lançamentos de objetos, teste de resistência de materiais, animação realista de objetos flexíveis e elásticos, entreternimento, entre outros.

Podemos gerar animações muito realistas, porém o custo computacional é alto pois para gerarmos um movimento fisicamente correto é necessário aplicação de fórmulas da Física. Isto requer um dispositivo para resolução de equações diferenciais, para resolver as equações descritas na Física, estes programas são geralmente de difícil implementação e caros computacionalmente.

Abaixo serão descritos alguns conceitos que mostram o poder desta técnica. Os conceitos estão descritos no curso *Physically Based Modeling: Principles and Practice* de Witkin e Baraff [27].

Conceitos de Dinâmica de Corpo Rígido

A Dinâmica do Corpo Rígido são conceitos da dinâmica de Newton aplicadas na modelagem de objetos. Os conceitos abaixo devem ser implementados em um sistema com movimentação de objetos baseados em física.

- *Posição e Orientação* - Cada objetos tem uma posição em relação a cena (vetor $x(t)$), que indica o local em que se encontra e uma orientação (matriz 3×3 $R(t)$ de rotação) que indica para onde tende a se mover.
- *Velocidade Linear e Angular* - Os objetos podem ter uma velocidade linear ($v(t) = \dot{x}(t)$) e angular ($w(t)$), descritas pelas leis da física.
- *Massa do corpo e Centro de Massa* - Se um corpo rígido é feito de partículas, a massa total do corpo é a somatória das massas das partículas. O centro de massa é definido por ($x(t) = \Sigma m_i r_i(t) \div M$)
- *Força e Torque* - Seja F_i a força total agindo na i -ésima partícula no tempo t , o Torque, intuitivamente é o eixo em que o corpo gira devido à força $F_i(t)$, novamente são definidas usando equações da Física.

Controles (Constraints)

Sistemas baseados em Controle permitem o usuário especificar uma coleção de controles ou restrições que as partes do modelos devem satisfazer. Nestes tipos de sistema a prioridade da restrição deve ser especificada, assim restrições mais importantes podem ser satisfeitas antes.

A especificação de controles é complexa, pode ser dada por um conjunto de igualdades ou desigualdades matemáticas, podem ser dadas por leis da Física. A especificação do controle pode ser do tipo energy-constraint, este é um sistema onde as restrições são representadas por funções que são não negativas em todos lugares exceto onde a restrição é satisfeita.

Abaixo são descritos alguns exemplos do artigo de Barzel e Barr [2] de possíveis controle em um sistema:

- *Ponto de controle* - um ponto de controle é fixado em um local do objeto fazendo com que este fique preso neste ponto. (ex: Pêndulo)
- *Ponto a Ponto* - Dois corpos têm movimentos livres, mas estão sempre juntos, presos por um ponto.
- *Ponto a Caminho* - Um ponto do objeto deve seguir um caminho especificado.
- *Dinâmica Inversa* - Dados um corpo e um controle, a dinâmica inversa consiste em:
 - 1) achar as forças que levam o objeto ao controle.
 - 2) achar as forças que mantêm o controle

Controle para não Penetração e Detecção de Colisão

Em corpos rígidos, que são completamente não flexíveis, não deve haver inter-penetração de objetos com o contato. Para problemas que requerem nonpenetration constraints devem ser considerados dois casos:

- *Colisão com mudança de velocidade abrupta* - Como não há penetração em corpos rígidos, após a colisão deve-se calcular a nova velocidade e as forças que vão descrever o movimento após o contato, isto vai depender diretamente das forças que evitam a interpenetração. A detecção de colisão pode ser calculada utilizando-se Bounding Boxes, estas são “caixas” que envolvem o corpo, quando duas bounding boxes se sobrepõem temos a indicação que ocorreu um contato, algoritmos estudados em Geometria Computacional resolvem o problema de sobreposição de bounding boxes eficientemente.
- *Corpos em repouso* - Para o contato em repouso entre dois corpos rígidos, ou seja o caso em que um corpo está sobre o outro e ambos estão sem movimento, devemos considerar três condições:
 - (1) As forças normais de contato devem evitar a inter-penetração.
 - (2) As forças de contato devem ser repulsivas entre si.
 - (3) A força no ponto de contato deve ser zero caso os corpos comecem a se separar. Além disso, as forças de contato devem evitar que o objeto entre em movimento sem motivo, isto é sem aplicação de nenhuma força durante o repouso do objeto.

Corpos não rígidos (deformáveis)

A teoria de elasticidade descreve materiais deformáveis como borracha, tecidos, papel e metais flexíveis. Ao aplicarmos uma força em um objeto deformável, deve-se calcular se a força foi necessária para provocar uma deformação na forma, caso isto aconteça deve-se calcular sua nova forma. Outras características podem ser consideradas, por exemplo se o objeto tiver a característica de restituição da forma antiga, ele deve voltar a forma anterior quando a força que o levou a se deformar para agir. Os objetos devem responder de forma natural às forças aplicadas.

Abaixo seguem algumas discussões de alguns textos relacionados a Modelagem baseada em Física, descrevendo as contribuições de cada um.

Physically Based Modeling: Principles and Practice. SIGGRAPH'97 Course Notes Neste curso apresentado na Siggraph em 1997 por Witkin e Baraff [27] são apresentados diversos conceitos de Physically Based Modeling. Além de descrever os conceitos, o curso mostra como deve ser feita sua implementação. O curso é bastante completo, iniciando com implementação de equações diferenciais e sistemas de partículas, seguindo para explicação da Modelagem baseada em Física. A maioria das idéias apresentadas no texto acima foram retiradas deste documento. A parte de Modelagem baseada em Física inicia-se com a descrição de corpo rígido e a dinâmica do corpo rígido, nos capítulos de Rigid Body Simulation, Unconstrained Rigid Body Dynamics, e segue descrevendo os problemas de penetração e detecção de colisão nos capítulo Nonpenetration Constraints.

A Modeling System Based On Dynamic Constraints Neste artigo de Barzel e Barr [2] é apresentado o conceito de *Dynamics Constraints*, uma técnica capaz de alterar a posição dos objetos para que estes satisfaçam as restrições ou controles (constraints) através de forças (constraints forces). No artigo é comentado o fato de Modelagem baseada em Física ser uma técnica difícil de implementar, controlar e lenta, daí a necessidade das restrições para melhor controlar os objetos gerados nesta modelagem. É feita uma descrição da mecânica Newtoniana utilizada e deifinição de corpos primitivos com parâmetros próprios, como raio, comprimento... São citados alguns exemplos de controles como: Ponto de Controle, Ponto a Ponto, Ponto a Caminho, já mencionados no texto acima. Constraint forces são definidas como forças internas que fixam pontos do objeto. Também é definido o conceito de Inverse Dynamics, que são forças que devem ser definidas para podermos descrever um determinado comportamento para o objeto. Este problema consiste em achar as forças que levam o objeto satisfazer a restrição e após isso manter a restrição para o objeto.

Energy Constraints on Parameterized Models Neste artigo de Witkin, Fleischer e Barr [28] é dada uma definição para energy constraints, que se comportam como forças que puxam e deformam partes do modelo para uma configuração desejada e após isto acontecer manter a configuração da mesma forma. As constraints são utilizadas como funções de energia, que se anulam quando a restrição é satisfeita. Algumas Constraints úteis são catalogadas no artigo, Anexação de um objeto a um ponto fixo, Atração entre superfícies, Cálculo de colisão sem calcular intersecções entre surperfícies (a idéia é utilizar forças em um campo de repulsão como constraints), fixação de restrição em uma linha, entre outros.

Dynamic Deformation of Solid Primitives with Constraints No paper de Metaxas e Terzopoulos [14] primitivas sólidas comuns como esfera, cilindro, cones, quadricas, são deformadas globalmente ou localmente para gerar novas formas. Conceitos como elasticidade e flexibilidade são discutidos e algumas fórmulas Físicas definidas para deformação dos objetos. Após a deformação a distribuição de massa no objeto deformado deve estar coerente. O artigo também discute um método baseado em Constraints para conectar as novas primitivas geradas para criação de modelos articulados, ao final do artigo alguns exemplos são mostrados como resultados.

Elastically Deformable Models Neste paper Terzopoulos, Platt, Barr e Fleischer [23] aplicam a teoria da elasticidade para contruir equações diferenciais que modelam o comportamento de superfícies e sólidos não rígidos. Modelos elasticamente deformáveis são ativos, respondem de forma natural a forças aplicadas e controles. Simulando propriedades físicas como tensão e rigidez, podemos modelar formas estáticas como objetos deformáveis. Incluindo massa e amortecimento simulamos dinâmica nestes objetos. A modelagem é feita utilizando-se splines e dinâmica. São também descritos algumas forças aplicadas para deformação dos modelos elásticos, como por exemplo força gravitacional e colisão

de modelos elásticos e objetos fixos e impenetráveis. A implementação da deformação é baseada em equações diferenciais descritas no artigo. Os resultados são exibidos no final do artigo, como podemos ver na figura 7.

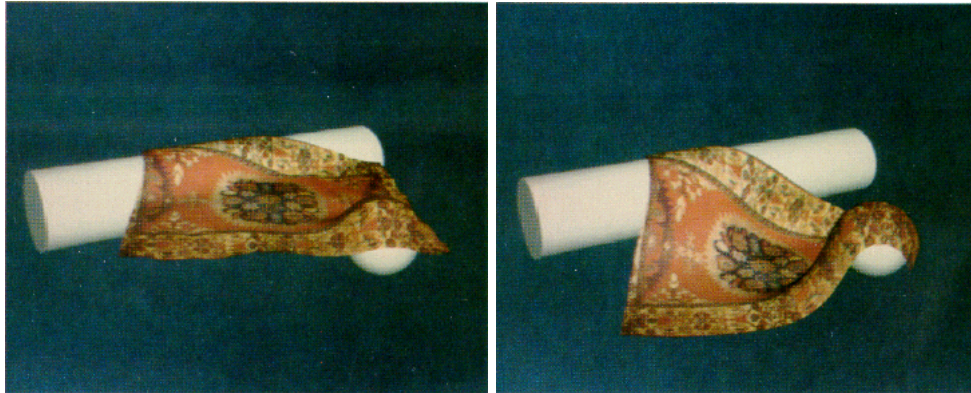


Figura 7: O tapete cai devido a ação do peso, sobre obstáculos fixos e sofre deformação

3 Simulação de Fenômenos Naturais

A modelagem e simulação de fenômenos naturais são geralmente tarefas complexas, já que estes fenômenos, na maioria dos casos, não tem forma bem definida ou regular. Já vimos neste texto a modelagem de fenômenos naturais utilizando modelagem com Sistema de Partículas, com este tipo de modelagem, podemos modelar bem fenômenos naturais tais como água, fogo, neblina e até mesmo plantas como foi descrito.

Nesta seção vamos focar o estudo a artigos relacionados a simulação de fenômenos naturais, estudando outras técnicas que viabilizam este tipo de simulação. O estudo está focado a modelagem de plantas para início da implementação de um simulador de ecossistemas. Veremos técnicas diferentes utilizadas para modelagem de vegetação, e as vantagens e desvantagens de cada técnica apresentada.

Nesta seção iremos descrever técnicas para modelagem de plantas, modelagem do vento e simulação de ecossistemas. Para modelagem de plantas vamos estudar técnicas para modelagem de vegetação rasteira e árvores, estudando técnicas bastante utilizadas como L-System e Slicing. Para simulação de ecossistemas iremos estudar artigos relacionados a simulação de ecossistemas diversos para conhecimento do assunto bastante relacionado ao tema da tese.

3.1 Modelagem de Plantas

Nesta subseção iremos descrever técnicas utilizadas para modelagem de plantas. Várias técnicas foram criadas e cada uma atende melhor um requisito. Podemos ter técnicas que produzem imagens bastante realistas como no artigo de Deussen *et al.* [3], porém são muito caras computacionalmente em termos de memória utilizada de tempo de geração de imagem. Também temos técnicas que produzem imagens de aproximações dos modelos rapidamente, mas menor qualidade e realismo, como na técnica de Slicing de Jakulin [9].

Alguns artigos utilizam várias técnicas para criação de uma cena com muitos vegetais, utilizando técnica com resultados mais realistas para vegetais próximos ao observador, e técnicas baseadas em aproximações para vegetais distantes do observador. Esta técnica é conhecida como Níveis de Detalhes, e será discutida nas próximas seções.

Uma forma de geração de plantas é utilizando Sistemas de Partículas como foi descrito no artigo de Reeves [20], porém esta técnica é muito cara computacionalmente, como já foi discutido. Vamos descrever algumas outras técnicas desenvolvidas para modelagem de vegetais e alguns artigos relacionados.

3.1.1 L-Systems

Lindenmayer Systems (L-Systems) são mecanismos para escrita de regras com palavras (string rewriting mechanisms), originalmente desenvolvidas por Lindenmayer como uma teoria matemática para o desenvolvimento de plantas. Originalmente, os L-Systems modelavam o desenvolvimento de organismos multicelulares simples (como por exemplo algas), em termos de divisão, crescimento e morte de células individuais. Porém, sua utilização logo foi estendida a plantas maiores e com estruturas de galhos complexas.

Os L-Systems demonstraram habilidade para modelar objetos naturais, particularmente modelos botânicos e celulares. Esta técnica usa o princípio de “database amplification”, que indica a possibilidade da geração de estruturas complexas a partir de conjunto de dados compactos.

A idéia do desenvolvimento é reescrever o sistema que substitui paralelamente módulos pais individualmente por configurações dos filhos, como nas técnicas de modelagem baseada em gramática já citadas. Todos os módulos devem pertencer a um alfabeto finito de tipos, e o comportamento dos

módulos podem ser especificados utilizando-se um conjunto finito de regras ou produções. A figura 8 indica três exemplos de aplicações destas produções.

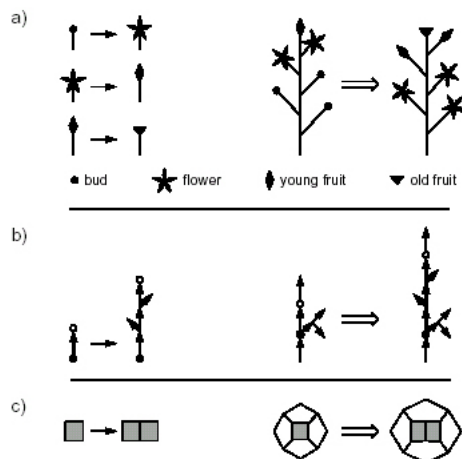


Figure 2: Examples of production specification and application: (a) development of a flower, (b) development of a branch, and (c) cell division.

Figura 8:

As produções podem ser aplicadas sequencialmente, um módulo por vez, ou em paralelo, com todos módulos sendo reescritos simultaneamente a cada passo da recursão. A escrita paralela é mais apropriada para modelagem de desenvolvimento de vegetais, pois o desenvolvimento ocorre simultaneamente em todas partes do organismo.

L-Systems paramétricos tem módulos indicados por letras associados a parâmetros. Vamos citar algumas notações utilizadas em L-Systems paramétricos.

Seja o alfabeto V , um módulo $A \in \mathfrak{R}$ com parâmetros a_1, a_2, \dots, a_n é denotado por $A(a_1, a_2, \dots, a_n)$. Expressões podem ser criadas utilizando-se os parâmetros e números constantes combinados com os operadores aritméticos $+$, $-$, $*$, $/$; exponencial \wedge ; relacionais $<$, \leq , $>$, \geq , $==$; lógicos $!$, $\&\&$, $||$, e parenteses $()$, além de outras funções matemáticas, como por exemplo logaritmo, seno, entre outras.

Os símbolos $:$ e \rightarrow são utilizados para separar os três componentes da produção: a *predecessor*, *condição* e *sucessor*. Uma produção segue o seguinte formato:

predecessor:condição \rightarrow sucessor.

Um exemplo de um L-System paramétrico é mostrado abaixo.

$\omega : B(2)A(4; 4)$

p1 : $A(x; y) : y \leq 3 \rightarrow A(x * 2; x + y)$

p2 : $A(x; y) : y > 3 \rightarrow B(x)A(x/y; 0)$

p3 : $B(x) : x < 1 \rightarrow C$

p4 : $B(x) : x \geq 1 \rightarrow B(x - 1)$

A figura 9 mostra a aplicação das regras de produção definidas acima. Assume-se que o módulo se substitui por si mesmo quando nenhuma produção definida satisfaz suas propriedades.

Podemos definir uma regra de produção com contextos, isto é só é aplicada quando aparece em um determinado contexto. A produção passa a ter o seguinte formato:

left-context<predecessor>right-context:condição \rightarrow sucessor.

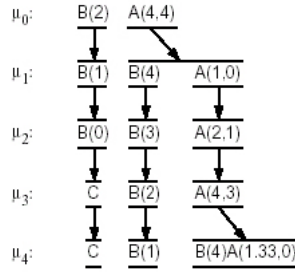


Figura 9: Aplicação das regras de produção.

Uma regra de produção como por exemplo: $A(x) < B(y) > C(z) : x+y+z > 10 \rightarrow E(x+y)F(y+z)$ só seria aplicada caso a string da regra em um certo passo de iteração é, por exemplo, da forma $\dots A(5)B(4)C(3)\dots$

Diversas formas de interpretação das strings geradas pelo L-System foram criadas, a interpretação chamada “turtle interpretation” descreve uma série de símbolos para utilização e alteração da geometria do objeto criado. Alguns símbolos definidos na “turtle interpretation” são descritos abaixo.

- @ $O(r)$ Desenha uma esfera de raio r na posição atual.
- + (θ) Vira para esquerda por um ângulo θ ao redor do eixo vertical do objeto.
- $\wedge(\theta)$ Aumenta o ângulo com o eixo horizontal de θ .
- ⋮

L-Systems estocásticos definem uma probabilidade para uma string ser substituída por uma regra. A notação $A(x) \rightarrow^{0.5} A(x)B(x)$ indica que a produção tem 50% de chances de ser aplicada.

Timed L-Systems atribuem tempo de vida a cada módulo em uma regra de produção. A produção $(A(x), 5) \rightarrow (A(x), 0.5)(B(x), 0)$ indica que o módulo A tem vida de 5 segundos. Quando A atinge a idade de 5 segundos a regra de produção é aplicada. Um novo módulo A é gerado com idade 0.5 segundos e um novo módulo B é gerado com idade 0 segundos.

L-SYSTEMS: FROM THE THEORY TO VISUAL MODELS OF PLANTS Este texto de Prusinkiewicz *et al.* [17] descreve detalhadamente a técnica, mostrando como se originou, como foi desenvolvida e como utilizá-la, servindo como texto base para o estudo desta técnica. O texto escrito acima foi baseado neste trabalho bastante completo. Descreve formas e padrão para geração e desenvolvimento de galhos e raízes de plantas de acordo com a concentração de nutrientes no vegetal. Simula o ataque da planta por insetos que impede a transmissão dos nutrientes para galhos na parte superior a parte atacada o que faz com os galhos murchem. Também descreve o comportamento das plantas ao se efetuar podas em alguns galhos gerados, verificando como isto influi no crescimento e desenvolvimento do vegetal.

Interactive Evolution of L-System Grammars for Computer Graphics Modeling Neste artigo McCormack [13] descreve os L-Systems e a evolução da técnica com contribuições dadas por diversos pesquisadores, explicando brevemente L-Systems paramétricos, estocásticos e por tempo. O artigo aplica o princípio da evolução, proposta por Charles Darwin, nas gramáticas criadas utilizando-se L-Systems, as regras poderiam sofrer mutação e passar pelo processo de seleção. As regras e parâmetros fariam o papel de genótipo, que contém as informações genéticas de cada indivíduo. As mutações podem alterar regras, módulos, parâmetros e funções utilizadas nas regras. Uma probabilidade para mutação

deve ser especificada. Podemos ter, por exemplo, mutações em que um módulo é removido, adicionado ou alterado por outro módulo em uma regra. Para mutações sobre parâmetros basta aplicar operadores aritméticos e funções matemáticas.

3.2 Modelagem de Vegetação Rasteira

Não são encontrados muitos trabalhos sobre modelagem de vegetação rasteira em específico até o momento, o único artigo que trata somente deste assunto é o artigo de Perbet [16], que descreveremos melhor no resumo abaixo.

Encontramos muitos artigos que descrevem como modelar vegetação rasteira utilizando a técnica que estão desenvolvendo, este é o caso dos artigos de Reeves [20] e Deussen *et al.* [3]. Ambas as técnicas apresentam excelentes resultados, porém o tempo de geração da cena é alto, não poderíamos gerar animações em tempo real com tais técnicas.

Um dos principais problemas de geração de imagens de vegetação rasteira é o número de instâncias que uma cena pode conter. Apesar de serem objetos simples de se modelar e animar, por exemplo uma folha de grama, muitas instâncias se tornam bastante caras computacionalmente caso cuidados não venham a ser tomados.

Animating Prairies in Real-Time Neste artigo Perbet e Cani [16] descrevem uma forma de se modelar e animar campos de grama em tempo real utilizando a técnica de níveis de detalhes. No artigos três níveis de detalhes são utilizados para modelagem da grama, modelos 3D, textura mapeada em um plano 3D e textura 2D mapeada no terreno, a técnica de níveis de detalhes vai ser melhor detalhada na seção *Técnicas para Renderização em Tempo Real*. Os três níveis de detalhes são utilizados para proporcionar a renderização rápida de um campo de gramas. Objetos mais próximos ao observador são modelados utilizando-se polígonos, objetos a média distância são modelados sobre um retângulo que contem a imagem de várias folhas de grama e podendo assim ser animado, e por último a textura da grama mapeada sobre o terreno, para objetos muito distantes em relação ao observador. Um algoritmo para controle de transição de níveis de detalhes, caso o observador se movimente, foi implementado. A animação da grama foi implementada utilizando cálculos, de posição da grama com modelagem baseada em física, realizados antes do início de execução do programa para geração de animação em tempo real. Dados sobre as posições, que as folhas deveriam ter quando dobradas por ação de alguma força, são armazenados. A animação é produzida utilizando-se rajadas de ventos com ações ou formatos pré-estabelecidos, as primitivas implementadas são descritas no artigo de Wejchert e Haumann [26], que será detalhado na seção sobre modelagem de vento.

3.3 Slices

A técnica de Slices foi proposta por Jakulin [9] em 2000 para uma aproximação de imagens de plantas utilizando image-based rendering. A simplificação deste modelo, se baseou, como a maioria dos artigos que propõem aproximações para modelagem de objetos, nas limitações do olho humano para percepção de geometria complexa. O resumo do artigo abaixo explicará a idéia por trás da técnica de Slices. É uma outra técnica que surgiu para criação de cenas complexas em tempo real.

Interactive Vegetation Rendering with Slicing and Blending A técnica proposta por Jakulin representa uma árvore modelando seu tronco com uma técnica simples baseada em mesh e as partes “esparças” da árvore como um conjunto de “cortes” (slices). Um slice é uma camada plana onde imagem das folhas e galhos são mapeadas, semelhante ao segundo nível de detalhe do artigo de Perbet e Cani [16]. Um conjunto de Slices paralelos é chamado de Slicing. Um conjunto de slices é colocado

paralelamente uns aos outros e outro conjunto perpendicularmente em relação aos primeiros. Cada primitiva, é mapeada então, no slice mais próximo. Quanto maior o número de slices maior a qualidade da imagem gerada. Para pontos de vista variando em 360° podemos ter slices em várias direções, inclusive com planos colocados de forma que circulem completamente a árvore, porém quanto mais slices mais memória é necessária e maior o tempo necessário para renderização. A técnica de blending faz com que primitivas que se sobrepõem na imagem sejam unidas em uma imagem com coeficiente de cor maior e de transparência menor para o slice que está na frente, é uma função suportada pela maioria das placas aceleradoras gráficas.



Figura 10: Campos de grama em níveis de detalhes de Perbet. Árvore modelada com slices.

3.4 Modelagem do vento

A modelagem do vento pode ser utilizada para animação de vários outros fenômenos naturais, como geração de ondas em um lago, movimentação do fogo, e de vegetais.

O vento pode ser modelado definindo-se áreas de influência do vento como no artigo de Giacomo *et al.* [6]. Nesta aproximação, todos os objetos dentro desta área de influência definida são afetados por uma força definida nesta área. A área pode ter o formato desejado pelo usuário.

Wejchert e Haumann [26] criaram um método baseado em aerodinâmica para animação de objetos sob ação do vento. Este artigo será descrito no parágrafo abaixo.

Animation Aerodynamics Métodos baseado em aerodinâmica são desenvolvidos para simular o controle de movimento de objetos em fluidos. Ao invés de criar um fluxo numericamente Wejchert e Haumann definem um conjunto de primitivas de fluxo, estas primitivas descrevem campos de velocidade do fluido. Ao interceptar objetos o fluxo aplica nas superfícies deste uma força para deslocamento do mesmo. As primitivas são do tipo uniforme unidirecional, campos de vento que se dispersam ou se atraem, a até mesmo um em formato de redemoinho. A união de primitivas pode gerar fluxos complexos. Obstáculos podem ser feitos com primitivas fortes o suficiente para alterar a direção de um fluxo, isto pode ser mais eficiente que detecção de colisão, além de permitir movimento natural e suave do fluxo. A força aplicada no objeto dentro do fluxo é proporcional à área e posição deste em relação ao fluxo. Ao se calcular o efeito do vetor de velocidade na superfície do objeto, o vetor de velocidade é separado em componente normal e tangencial, que representam a pressão e viscosidade do fluido. Como exemplo foi mostrada a animação de folhas de árvores soltas sendo carregadas pelo vento e batendo em obstáculos.

3.5 Simulação de Ecossistemas

Artigos relacionados a simulação de ecossistemas descrevem o comportamento de espécies em determinados ecossistemas. São trabalhos diretamente relacionados ao tema de minha dissertação de mestrado.

Alguns artigos foram lidos apresentando problemas de ecossistemas específicos tais como: Insect Population Simulation [1], The Digital Simulation of River Plankton Population Dynamics [15] e SIMPLE: A Simplified Ecosystem Model for Lake George, New York [10] que foram importantes para entendimento de conceitos relacionados a ecossistemas e entender o foco de estudo de tais artigos.

Os artigos falam a respeito do controle de população de insetos ou plancton em um certo lago, estudos que ajudam a fazer projeções da população destes seres no futuro, em função dos nutrientes, poluição, inseticida que são encontrados nos ambientes que estes vivem. Alguns conceitos como ciclo de população e competição por nutrientes e espaço são discutidos.

A modelagem de um ecossistema complexo que requer um sistema bastante poderoso para poder simular a interação dos seres, o controle e estabilização das populações de animais e vegetais, dadas competição por espaço, alimento e outros recursos disponíveis no ecossistema. Além de computação gráfica tal dispositivo até mesmo requer conceitos de inteligência artificial como veremos no artigo de

Abaixo faremos discussões sobre textos importantes relacionados ao assunto, um exemplo de ecossistemas vegetais e o outro de ecossistemas animais, observaremos a complexidade de tais sistemas.

Realistic Modeling and Rendering of Plant Ecosystems Neste artigo de Deussen *et al.* [3] são descritos diversos problemas relacionados a renderização de cenas naturais com grande número de plantas. No texto são descritos algoritmos de geração de terrenos e especificação de população das plantas, a definição do terreno e da distribuição podem ser feitos por especificação explícita (o usuário intervém na descrição da cena) ou por geração procedural (a distribuição é realizada simulando interações em modelos ecológicos), ou mesmo por ambas as técnicas. As plantas individuais são modeladas utilizando L-Systems. O caso do algoritmos de geração de distribuição de plantas de diferentes espécies requer parâmetros como: (i) o número de novas plantas adicionadas por passo de simulação, (ii) o tamanho máximo das plantas, (iii) seu crescimento médio, (iv) probabilidade de sobreviver com uma planta com vantagens naturais em um competição e (v) preferência por áreas úmidas ou secas. O artigo inclui exemplos gerados com imagens bastante realistas porém o tempo de geração de cenas com muitos vegetais é bastante alto.

Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World Este artigo de Terzopoulos *et al.* [24] descreve um ecossistema marinhos habitado por algumas espécies de peixes artificiais. É um artigo completo sobre sistemas de simulação de ecossistemas. A modelagem dos peixes é feita utilizando Splines, texturas de fotos de peixes reais são mapeadas sobre o modelo para gerar modelos parecidos com os reais sem o trabalho de se modelar propriedades como cor, escamas dos peixes, entre outros. O artigo mostra o estudo da anatomia e comportamento dos peixes, o modelo e a locomoção dos peixes é baseado em física, modelando músculos utilizados para movimento do peixes e teoria de hidrodinâmica. Algoritmos de aprendizagem são utilizados para que dados os músculos necessários para locomoção do peixe, este aprender a nadar. Até mesmo a visão dos peixes é modelada de forma realista, com bases em estudos biológicos e de visão computacional. Algumas rotinas para sobrevivência dos peixes são implementadas, dentre elas: evitar obstáculos estáticos, evitar outros peixes, comer, cruzar, perambular, fugir e aprender. Tipos de peixes são definidos, os predadores, e presas, as presas aprendem a evitar os predadores. O sistema define até forma que os machos contejam a fêmea para o acasalamento. Os resultados são surpreendentes, vídeos gerados mostram o resultado deste estudo.

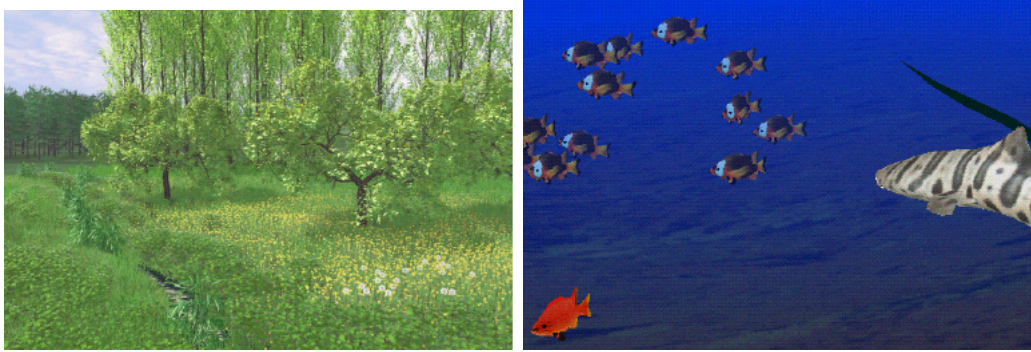


Figura 11: Ecossistemas de plantas e peixes dos artigos de Deussen e Terzopoulos.

4 Técnicas para renderização em Tempo Real

Nesta seção descreveremos a utilização de uma técnica conhecida por **Níveis de Detalhes**. Nesta técnica, os objetos que compõem a cena são exibidos utilizando diferentes técnicas de modelagem. A ideia básica é a seguinte: Para objetos mais próximos ao observador uma técnica mais precisa e realista, porém mais custosa computacionalmente, é utilizada. Para objetos mais distantes aproximações ou técnicas menos realistas, porém com menor tempo de renderização, são utilizadas. Desta forma uma cena pode ser gerada com imagens realistas e com um frame-rate (imagens geradas por segundo em uma animação) suficiente para ser animada em tempo real.

Alguns exemplos de artigos que utilizam a técnica de níveis de detalhes, ou do artifício de se enganar a vista humana para objetos distantes, o artigos de Perbet [16] e de Jakulin [9] são exemplos deles. No artigo de Markosian *et al.* [12] a técnica é utilizada para renderização de desenhos animados, Art-based Rendering, desenhos de árvores, pêlo e grama são feitos utilizando técnicas de stroking para que eles tenham aparência desenhos feitos a mão. Vamos estudar melhor o artigo de Giacomo *et al.* [6] que também utiliza esta técnica para modelar florestas.

An Interactive Forest Na animação das árvores da floresta modelada por Giacomo, Capo e Faure [6] dois métodos de modelagem são combinados. Um método procedural trata a maioria das árvores eficientemente. Para outras árvores um método com Modelagem baseada em Física é utilizado, este é utilizado para árvores próximas ao observador e permite a interação do usuário com as árvores modeladas. A animação das árvores também pode ser gerada pelo vento, a aproximação para modelagem do vento neste artigo é definição de áreas de influência do vento, todas as árvores ou galhos dentro desta área são afetados por uma força definida nesta área. As equações do movimento das árvores próximas são descritas assim como a transição de níveis de detalhes.

5 Implementação

Esta seção mostra o que foi implementado até o momento, após o estudo de artigos relacionados ao tema de pesquisa. Apenas o início do sistema simulador de ecossistemas vegetais foi implementado, até a entrega deste relatório, apenas a geração de campos de grama era possível.

Nas próximas subseções veremos o que foi necessário para a implementação dos campos de grama, assim como os algoritmos criados para isto. A cena gerada é tridimensional e a ferramenta utilizada para auxílio na geração das cenas foi o Java3D.

Serão explicados detalhes sobre a geração dos terrenos onde a vegetação será distribuída, o algoritmo utilizado para a distribuição de grama será detalhado. Serão comentadas melhorias para o sistema que deverão ser feitas quando as árvores e arbustos forem implementados.

A última seção trata da modelagem da grama e como conseguir modelar um campo com muitas folhas de grama em tempo real utilizando modelagem baseada em Física.

5.1 Renderização do Terreno

O terreno foi implementado como uma rede de triângulos dispostos paralelamente. Os pontos da rede (mes) de triângulos distam uns dos outros de um tamanho fixo nos eixos x e z, pertencentes ao plano definido por $y = 0$ ¹, a altura porém pode variar.

O algoritmo utilizado para geração de terrenos irregulares é bastante simples, para cada dupla de x e z distantes de uma distância d definida calculamos a altura do ponto com um algoritmo que gera um valor aleatório, porém próximo a uma distância mínima definida dos seus vizinhos já criados.

O algoritmo de geração de terrenos irregulares depende de muitos parâmetros como número de pontos no mesh, distância entre os pontos do mesh, diferença máxima da altura entre pontos vizinhos, desta forma, podemos criar terrenos diferentes alterando esses parâmetros.

Para criação de colinas implementamos um algoritmo simples onde são passados parâmetros como altura da colina e um parâmetro definido abertura e variação de abertura. A abertura indica o quanto o diâmetro da colina irá aumentar do cume até a base, para cada nível de altura da colina. A variação da abertura indica a proporção que a abertura irá aumentar ou diminuir.

A altura dos níveis é definida com relação ao diâmetro final da base e da altura da colina escolhidas pelo usuário. Pelo algoritmo descrito são calculados os pontos da colina nos planos $x = \text{centroX}$, e $z = \text{centroZ}$, onde centroX é a posição do centro da colina no eixo x e centroZ a posição do centro da colina no eixo z. Os demais pontos da colina são calculados por interpolação.

Algoritmos para geração de formas mais complexas como montanhas e chapadas, deverão ser implementados futuramente.

5.2 Distribuição de Plantas

Um algoritmo simples de distribuição foi implementado, pois como só há um tipo de vegetação ainda não foi necessário a implementação de competição considerando propriedades dos vegetais.

A idéia do algoritmo é verificar se uma dada posição do terreno tem nutrientes suficientes para criação de um tufo de grama. Os nutrientes são propriedades do terreno. Caso haja nutrientes um tufo é criado, caso não haja uma posição de terreno adjacente é verificada. O algoritmo distribui um número de tufos de grama definidos pelo usuário.

Dois tufos não irão ocupar o mesmo espaço pois é guardada uma matriz que indica quais volumes de terrenos já estão ocupados. Os tufos tendem a se espalhar mais em regiões com maior concentração de nutrientes.

¹No Java3D o eixo y é o eixo vertical da cena

Caso outras formas de vegetação venham a ser implementadas deve-se verificar além da competição por espaço e nutrientes se o vegetal deixa luz o suficiente para os vegetais menores. Por exemplo, uma árvore faria sombra abaixo dela, o que inibiria o crescimento de grama ou flores perto dela.

5.3 Renderização de Grama

As gramas são modeladas utilizando conjunto de vetores da geometria, um conjunto de vetores descreve a direção do talo da grama e dois outros conjuntos de vetores indicam a largura da grama para cada ponto definido pelos vetores de direção do talo da grama.

Dado um ponto inicial no espaço 3D $(x_{i,j}, y_{i,j}, z_{i,j})$, para um certo i e j pertencentes as coordenadas dos eixos x e z . Soma-se os vetores de direção (dx_a, dy_a, dz_a) , e de largura (lx_a, ly_a, lz_a) e (rx_a, ry_a, rz_a) , com a menor que o número de vetores no conjunto definido, podemos criar os pontos para geração dos polígonos (triângulos) que irão gerar a folha de grama.

Gramas de diferentes ângulos são pré-computados, utilizando-se os vetores criados, observe que basta apenas calcular a diferença de ângulo nos vetores de direção, já que os de largura não são alterados. A diferença dos ângulos deve ser especificado pelo usuário, quanto mais ângulos mais posições de grama serão criados, mas maior será a quantidade de memória utilizada. As gramas com diferentes ângulos serão úteis na criação de tufos de gramas e para uma futura animação.

5.3.1 Adicionando propriedades Físicas para a grama

Gramas criadas apenas com o algoritmo acima serão pouco realistas, pois todas são retas. Na natureza as folhas são ligeiramente dobradas devido a ação do peso. Seguindo esta idéia criamos um algoritmo que dobra as folhas utilizando Constraints (controles), conceito explicado na seção de Modelagem baseada em Física.

As constraints irão impedir que a força peso dobre completamente a folha. As constraints são definidas levando em conta a largura da folha e uma constante que define a grossura do talo da folha.

Quando a constraint é maior que a força peso a posição não se altera. Quando o peso é maior, a folha se curva até que a componente normal em relação a folha da constraint e da força peso se anulem conforme é mostrado na figura. 12

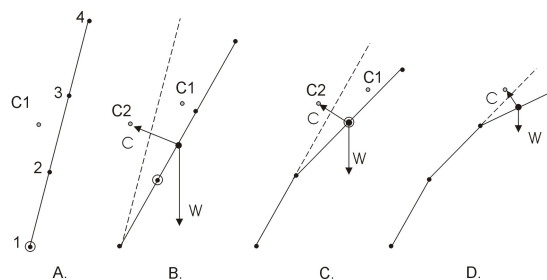


Figura 12: Esquema da folha modelada com Constraints.

A folha vai sendo “dobrada” nas partes onde estão os pontos definidos pelos vetores de direção. Para dobrar a folha basta alterar estes vetores de direção. A folha começa a ser dobrada do vetor de direção do conjunto mais próximo à base da folha e segue até que todos vetores sejam dobrados. Note que na parte mais próximas à base a força de constraint deve ser maior, pois a massa das partes superiores da folha estão influenciando no aumento da força peso, porém isto não é estranho já que geralmente o talo das folhas são mais grossos na região inicial da folha em geral.

6 Resultados e Próximas Etapas

Geramos cenas de tufos de grama sobre faixas de terrenos, o tempo de geração é de poucos segundos devido aos cálculos pré-efetuados. A animação está em fase de implementação, ainda é lenta e não funciona corretamente, mas alguns resultados foram gerados e mostrados nas imagens a seguir.

O algoritmos de nível de detalhes utilizado é o disponibilizado pelo Java3D que utiliza menos polígonos para gramas mais distantes. Algoritmos utilizando slices devem ser implementados para melhorarmos o tempo de animação.

Os próximos passos são terminar a implementação da animação e níveis de detalhes de gramas. Gerar outras formas de vegetação rasteira, como flores e ervas. Mais futuramente deve ser estudar a implementação de arbustos e árvores utilizando L-Systems ou uma aproximação para renderizar árvores em tempo real.

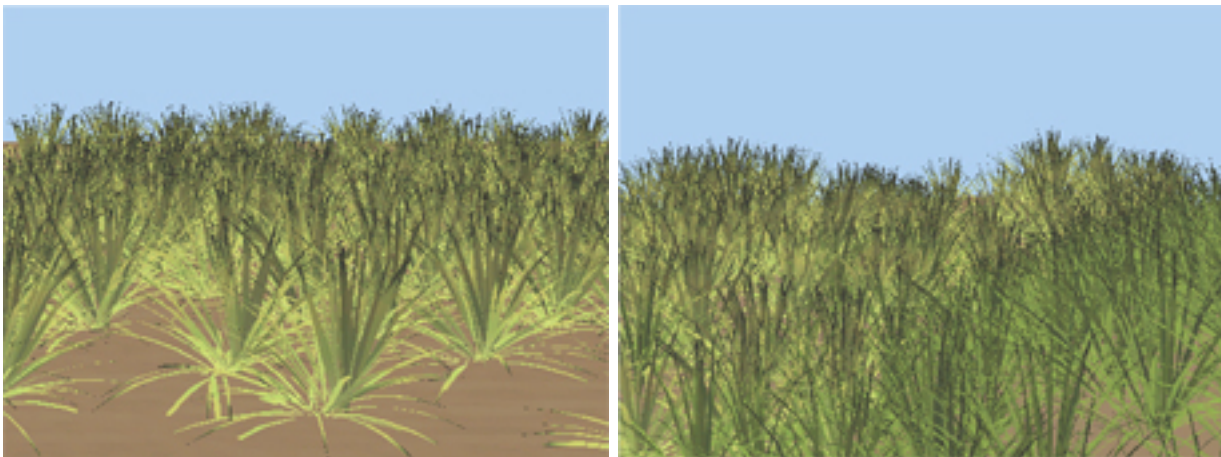


Figura 13: Campos de grama modelados pelo nosso sistema.

7 Conclusão

Os objetivos planejados no início do semestre foram quase todos cumpridos, com exceção do término da animação de gramas. Nos prolongamos mais na parte de estudos de artigos para ter conhecimento do maior número de técnicas já desenvolvidas para termos como pensar na melhor forma de resolver os problemas.

O cronograma também foi atrasado devido a elaboração de um artigo baseado no que foi implementado até o momento. As imagens do sistema apresentadas neste relatório foram utilizadas no artigo também.

O texto gerado neste relatório vai ser de grande importância para escrever a dissertação de mestrado, por conter um grande resumo das técnicas estudadas e idéias utilizadas na implementação.

Uma técnica de geração de folhas baseadas em Constraints foi desenvolvida pela primeira vez, servindo para modelagem de qualquer tipo de folha, não apenas para folhas de grama. Além disso, fomos capazes de gerar imagens com boa qualidade, considerando-se os recursos que tiveram que ser sacrificados para geração rápida das folhas, como repetição de instâncias.

O mais importante foi o estudo de artigos relacionados a modelagem de árvores e simuladores de ecossistemas, estudo que será muito importante para implementação das partes futuras do sistema.

Referências

- [1] John R. Barrett, John E. Foster, Greg A. Wong, and Elizabeth C. Stanley. Insect population simulation.
- [2] Rouen Barzel and Alan H. Barr. A Modeling System Based On Dynamic Constraints. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 22(4):179–188, August 1988.
- [3] Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomír Měch, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic Modeling and Rendering of Plant Ecosystems. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 32(Annual Conference Series):275–286, 1998.
- [4] David S. Ebert.
- [5] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes.
- [6] Thomas Di Giacomo, Stéphane Capo, and François Faure. An Interactive Forest. In *Eurographics Workshop on Computer Animation and Simulation*, pages 65–74, sept. 2001.
- [7] Jonas Gomes and Luiz Velho.
- [8] M. Green. Topics in virtual reality and interactive 3d graphics, part two.
- [9] A. Jakulin. Interactive Vegetation Rendering with Slicing and Blending. In *Proc. Eurographics 2000 (Short Presentations)*. Eurographics, Aug 2000.
- [10] J.P. Killus. Simple: A simplified ecosystem model for lake george, new york.
- [11] Martti Mantyla.
- [12] L. Markosian, B. Meier, M. Kowalski, L. Holden, J. Northrup, and J. Hughes. Art-based Rendering with Continuous Levels of Detail, 2000.
- [13] J. P. McCormack. Interactive evolution of L-system grammars for computer graphics modelling. In D. G. Green and T. Bossomaier, editors, *Complex Systems: from biology to computation*, pages 118–130. ISO Press, Amsterdam, 1993.
- [14] Dimitri Metaxas and Demetri Terzopoulos. Dynamic Deformation of Solid Primitives with Constraints. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 26(2):309–312, 1992.
- [15] The Digital Simulation of River Plankton Population Dynamics. R. mark claudson, 1975.
- [16] Frank Perbet and Marie-Paule Cani. Animating Prairies in Real-Time. In *ACM Interactive 3D Graphics*, USA, Mar 2001.
- [17] Przemyslaw Prusinkiewicz, Mark Hammel, Jim Hanan, and Radomír Měch. L-SYSTEMS: FROM THE THEORY TO VISUAL MODELS OF PLANTS, 1996.
- [18] Przemyslaw Prusinkiewicz, Mark Hammel, and Radomir Mech. Visual models of morphogenesis: A guided tour.
- [19] W. T. Reeves. Particle Systems – A Technique for Modeling a Class of Fuzzy Objects. In *ACM Transactions of Graphics*, pages 91–108, April 1983.

- [20] William T. Reeves and Ricki Blain. Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 19(3):313–322, 1985.
- [21] A.A.G. Requicha. Representations for rigid solids: Theory, methods and systems. *ACM Computing Surveys*, 12(4):437–464, 1980.
- [22] Karl Sims. Particle Animation and Rendering Using Data Parallel Computation. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 24(4):405–413, August 1990.
- [23] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischert. Elastically Deformable Models. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 21(4):205–214, July 1987.
- [24] Demetri Terzopoulos, Xiaoyuan Tu, and Radek Grzeszczuk. Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4), 1994.
- [25] Alan Watt.
- [26] J. Wejchert and D. Haumann. Animation Aerodynamics. *Computer Graphics (ACM SIGGRAPH Proceedings)*, pages 19–22, July 1991.
- [27] Andrew Witkin and David Baraff. Physically Based Modeling: Principles and Practice (SIGGRAPH'97 Course Notes).
- [28] Andrew Witkin, Kurt Fleischer, and Alan Barr. Energy Constraints on Parameterized Models. *Computer Graphics (ACM SIGGRAPH Proceedings)*, 21(4):225–232, 1987.