

Laboratório VISGRAF

Instituto de Matemática Pura e Aplicada

**Visualizacão de Terrenos em Tempo Real com Malhas
Semi-Regulares**

Luiz Velho, Lourena Rocha, Sergio Pinheiro, Marcelo Vieira

Technical Report TR-2004-05 Relatório Técnico

August - 2004 - Agosto

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

Visualização de Terrenos em Tempo Real com Malhas Semi-Regulares

LOURENA ROCHA, SERGIO PINHEIRO, MARCELO VIEIRA, LUIZ VELHO

IMPA–Instituto Nacional de Matemática Pura e Aplicada

Estrada Dona Castorina, 110, 22460

Rio de Janeiro, RJ, Brasil

{lourena, sergio, mbvieira, lvelho}@visgrafimpa.br

Abstract. Este trabalho mostra que os dados de terreno podem ser representados a partir de malhas semi-regulares. A vantagem de utilizar malhas semi-regulares é que permite representar a maioria dos objetos do mundo real de forma única e transparente. A definição de operações estelares proporciona o desenvolvimento de uma interface simples para as operações de adaptação da malha (simplificação e refinamento). Essa interface permite que as operações de culling, geomorphing e cálculos da função de erro sejam facilmente incorporados no algoritmo de adaptação da malha.

1 Introdução

Visualização interativa de terrenos é uma tarefa importante e útil com aplicações em GIS, jogos, realidade virtual, visualização científica, simuladores de voo, além de ter uso militar.

Para aplicações como essas que requerem imagens precisas de grande massa de dados com alta taxa de interatividade, este é um problema complexo e desafiador, pois o tamanho do conjunto de dados torna difícil a renderização em tempo real. Isso se deve ao fato de que os dados do terreno não cabem na memória, precisando serem lidos do disco, e também ao fato de que o renderizador deve processar um grande número de triângulos.

1.1 Motivação

Quando tentamos visualizar terrenos estamos interessados, na verdade, em modelar e visualizar o mundo real. Desse modo, é natural querermos que os sistemas de visualização de terrenos sejam capazes de lidar com todo o globo terrestre.

Com os recentes avanços tecnológicos, tais como satélites, sistemas de armazenamento de alta capacidade e internet, já existem alguns sistemas que visualizam toda a terra ou algumas regiões específicas, com resoluções variadas. Como exemplo podemos citar o *Keyhole* [2], *TerraServer* [5] e *TerraFly* [3]. Ambos se baseiam, principalmente, em fotos aéreas e de satélite, fornecendo apenas uma visualização 2D. A exceção é o *keyhole*, que possui um layer de visualização 3D. No entanto, os dados de elevação são limitados e além disso, ele possui a desvantagem de ser comercial, em contraste com os outros dois que estão disponíveis na internet.

Um sistema completo de visualização de terreno deve dar suporte ao processo de modelagem bem como o

gerenciamento em run-time. Como principais habilidades ele deveria:

- Representar todo o globo terrestre;
- Lidar de maneira eficiente com grande massa de dados;
- Fornecer visualização em tempo real.
- Ter uma estrutura de modelagem flexível para as diversas aplicações;

Para obter taxas interativas de visualização, uma solução amplamente utilizada é a implementação de um esquema de multi-resolução adaptativo. Nesse caso, uma representação em multi-resolução do terreno é precomputada e usada at run-time para construir adaptativamente uma triangulação adequada do terreno a cada frame.

1.2 Contribuição

Apesar do amadurecimento na área de visualização interativa de terrenos, a maioria dos trabalhos se divide entre os que representam o terreno como valores de alturas amostrados num grid retangular – *Triangulação regular* – e os que fazem a amostragem sobre numa network de triângulos – *Triangulação irregular*. No entanto, um outro grupo tem tido grande destaque ao se tratar da representação do terreno. São os trabalhos que usam métodos híbridos, explorando as vantagens das representações regulares e irregulares.

Nesse trabalho, propomos a implementação de um sistema de visualização de terreno em tempo real. Nosso objetivo é, a longo prazo, montar uma plataforma completa que, fazendo uso de uma estrutura híbrida, flexível e

poderosa de representacao, seja capaz de lidar com grande massa de dados, fornecer uma visualizacao em tempo-real com imagens de qualidade e, no caso especifico de terrenos, possa modelar todo o globo terrestre.

Essa plataforma tera' uma aplicacao real em parceria com o INPE - Instituto Nacional de Pesquisas Espaciais - no projeto *TerraLib* [4] inicialmente desenvolvido por eles. A *TerraLib* [9, 10] 'e uma biblioteca que tem como objetivo permitir o desenvolvimento de ambientes de GIS com enfase no uso de sistemas gerenciadores de bancos de dados para armazenar todos os tipos de dados geograficos.

Este paper esta' organizado em seis secoes, incluindo esta introducao. A Secao 2 apresenta uma rapida descricao dos trabalhos relacionados e na Secao 3 mostramos os conceitos e a arquitetura geral de um sistema de visualizacao de terrenos. Secao 4 descreve a estrutura hibrida usada em nossa plataforma e como 'e implementada a primeira versao do sistema, que e' view dependent, e possui tarefas basicas como frustum culling e geomorphing. A Secao 5 resume os principais pontos do trabalho e da' as direcoes de trabalhos futuros.

2 Trabalhos Relacionados

Nos ultimos anos, varios algoritmos e estrategias para visualizar terreno em tempo-real tem sido propostos. Nesta secao faremos um breve overview sobre os principais trabalhos, classificando-os com relacao as estruturas de triangulacao que lidam com a multi-resolucao dinamica.

2.1 Triangulação Regular

Os metodos baseados em triangulacao regular operam sobre um grid retangular de dimensao $2^n + 1$. A posicao dos vertices da triangulacao e' restrita 'a posicoes no grid. Como exemplos de trabalhos nessa linha podemos citar: Lindstrom/Koller [16], ROAM [8] , Lindstrom/Pascucci[17, 18].

Em geral, esses algoritmos sao de facil implementacao e a estrutura regular permite a ausencia de ponteiros, o que implica em menos consumo de memoria.

Em contrapartida, apenas informacoes geometricas sao armazenadas. Assim, sem conhecimento de informacoes topologicas eles ficam incapacitados de representar de maneira simples dados com topologia mais complexa. A estrutura regular forca o uso de metricas saturadas, que nao e' otima* e para algumas aplicacoes e' uma restricao. Alem disso, a regularidade da amostragem impede que a malha se ajuste de uma maneira otima ao modelo, proporcionando assim malhas subotimas com respeito ao numero de triangulos.

A estrutura de triangulacao proposta neste trabalho, em contraste com as usadas nos algoritmos regulares, armazena a estrutura topologica da malha corrente, sem

grande custo de memoria. Isto possibilita o uso de diferentes metricas, tornando o sistema flexivel em termos de aplicacoes.

2.2 Triangulação Irregular

Os esquemas irregulares representam o terreno como um mapa de valores de altura amostrados numa network irregular de triangulos (TIN). Trabalhos que usam essa estruturas sao: Hoppe [11, 12], Puppo[14, 13, 15].

A principal vantagem desses esquemas com relacao aos regulares e' que, como os vertices da malha nao estao restritos a posicoes regulares, eles geralmente requerem menos poligonos para aproximar a superficie para um dado erro. Assim eles conseguem representar bem a geometria do terreno. Alem disso, esses metodos sao flexiveis quanto ao uso de metricas de erro.

No entanto, eles requerem estruturas de dados complexas para representar uma triangulacao da multi-resolucao e as relacoes de dependencias, implicando em custos mais altos associados com as operacoes de simplificacao e refinamento comparados a metodos hierarquicos regulares. Essas estruturas tem menos expressividade[21] para criar malhas adaptadas do que as malhas regulares adaptativas e assim nao conseguem explorar bem a caracteristica de niveis de resolucao.

Nosso esquema de triangulacao, em contraste com TINs, armazena a estrutura da malha corrente sem alto custo, pois nao guarda ponteiros da hierarquia, e satisfaz os criterios (Secao 4.1), segundo Puppo[21], de uma malha ideal para representar superficies gerais multitrianguladas.

2.3 Metodos Híbridos

Esses trabalhos propoem explorar as vantagens dos esquemas regulares e irregulares. Os principais trabalhos a serem citados sao: Pajarola [19] e Cignoni [6].

Em Pajarola [19] uma estrutura hierarquica regular e' usada para aproximar dados irregulares de terreno. Cignoni [6] faz a fast VLOD extraction, considerando um patch como o menor primitivo para simplificacao e refinamento. Embora tenham vantagens dos esquemas anteriores, esse trabalhos sofrem de problemas que vao desde aproximacoes sub-otimas da malha a pobre frustum culling.

A estrutura de triangulacao proposta neste trabalho tambem 'e hibrida e baseada numa quadrangulacao triangulada. Suas principais caracteristicas sao:

- E' semi-regular;
- Armazena somente a topologia corrente do terreno;
- Nao guarda ponteiros da hierarquia, acarretando menos custo associado;

- É flexível quanto ao uso de métricas de erro;
- Atende aos critérios ótimos de expressividade, profundidade logarítmica e crescimento linear definidos por Puppo et al em [21].

3 Visualização em Tempo Real de Terreno

A visualização em tempo-real de terreno ainda é um grande desafio devido a sua complexidade geométrica e topológica. Essa complexidade é uma consequência direta do grande volume de dados que deve ser armazenado e processado para obter uma representação precisa do mundo real.

Para realizar o processo de visualização no menor tempo possível é essencial evitar cálculos complexos e minimizar a quantidade de dados que deve ser processada. Desta forma, o processo de visualização de dados de terreno deve ser dividido em duas etapas. A primeira etapa consiste de técnicas de formatação e extração de informações geométricas e topológicas dos dados. Essa etapa é chamada de *fase de pré-processamento*. A segunda etapa envolve técnicas de rendering que exploram as estruturas de dados e outras informações, geradas na fase de pré-processamento, para acelerar o processo de visualização. Essa segunda etapa é chamada de *fase de execução*.

3.1 Pré-Processamento

O modelo matemático mais utilizado para descrever terreno o define como uma área retangular do plano e associa a cada ponto dentro dessa área uma informação de altura e cor. Assim, um terreno é definido por uma função $h : U \rightarrow \mathbb{R}$ e por uma função $c : U \rightarrow \mathbb{R}^n$. A função h é chamada de *função de altimetria* e a função c é um mapeamento de textura.

A representação discreta da função h é chamada de *campo de altura*. A representação matricial (amostragem uniforme) do campo de altura é conhecida como *modelo digital de elevação* (DEM). A operação de reconstrução da função h é obtida a partir da interpolação linear do DEM. A operação de interpolação gera uma superfície linear por partes. Geralmente, essa superfície é representada por uma malha triangulada, que pode ser regular ou não.

O problema de visualização de malhas em tempo-real pode ser colocado como um problema de minimização de uma função de erro obedecendo um conjunto de restrições. As restrições são determinadas a partir de critérios como taxa de quadros por segundos, número de triângulos, etc. O erro pode ser definido segundo vários critérios, como por exemplo qualidade visual dos quadros, precisão geométricas, etc.

Os critérios baseados em qualidade visual são chamados de *view-dependentes* porque a medida de erro leva em consideração os parâmetros da câmera. Os

critérios baseados em erro geométricos são chamados de *view-independentes*, pois só levam em consideração as propriedades geométricas e topológicas da malha. Normalmente, a função de erro é definida a partir da composição desses dois critérios. Essa composição deve buscar uma boa relação entre qualidade de visual e precisão geométrica.

Os algoritmos gulosos são os mais utilizados para encontrar uma malha que minimize a função de erro. Esses algoritmos partem do princípio de que os elementos topológicos (vértices, arestas e faces) podem ser ordenados a partir dessa função de erro ou função de adaptação.

A estrutura de dados mais adequada para manter os elementos sempre ordenados é a fila de prioridades. A principal característica da fila de prioridade é que permite manter ordenados elementos cuja prioridade varia dinamicamente. Essa característica é essencial para que critérios view-dependentes sejam incorporados aos algoritmos gulosos de forma otimizada.

O cálculo do erro geométrico deve permitir estimar o erro máximo de uma determinada região da malha. Esse erro é estático e só depende das características topológicas e geométricas da malha. Assim, é possível calcular e associar um erro geométrico para cada elemento topológico na fase de pré-processamento.

O processo de estimação do erro de cada elemento é realizado a partir da malha gerada com todos os vértices. Em seguida, a cada passo um vértice é retirado da malha. O erro é calculado a partir de uma métrica que estime a diferença entre a malha original e a nova malha obtida com a retirada do elemento.

A métrica mais utilizada é baseada na estrela do vértice. A estrela do vértice é a região da malha que é influenciada quando o vértice é retirado (simplificação). O erro de um vértice é medido estimando o grau de modificação sofrido na sua estrela na ocasião de sua retirada. Existem várias formas de estimar o erro de um vértice como variação de altura, de área ou de volume sofrida na região. Essa variação é normalmente medida utilizando o erro máximo baseado na métrica L^∞ .

Note que no caso de multi-resolução a estrela de um vértice vai crescendo e englobando as estrelas de vértices outros vértices que foram retirados antes deste último. Intuitivamente, isto quer dizer que as estrelas dos vértices formam regiões auto encaixantes. O erro atribuído para esse último vértice é o máximo entre o erro calculado em sua estrela e o calculado nas estrelas que foram englobadas. O resultado final é a criação de uma matriz de erros com as mesmas dimensões do DEM.

De forma geral os cálculos que apenas dependem de informações geométricas e topológicas devem ser realizados na fase de pré-processamento. Os resultados desses cálculos são utilizados para acelerar processos,

como culling e adaptação, que são essenciais para visualização em tempo-real da malha.

Os cálculos de volumes envolventes, realizados na fase de pré-processamento, são utilizados para acererar o processo de culling.

A representação em multi-resolução da textura deve ser criada na fase de pré-processamento. É inviável criar diferentes níveis de resolução em tempo-real. A representação em multi-resolução de uma textura é obtida a partir de operações de filtragem e reamostragem que não podem ser executadas em tempo-real. Após o processamento é criada uma pirâmide de imagens, cada nível de pirâmide representa um nível de resolução da textura.

3.2 RunTime

Os cálculos que estimam os erros baseados em critérios view-dependentes são realizados na fase de visualização. Assim, cada elemento topológico é associado a erro formado pela composição do erro view-independente, calculado na fase pré-processamento, e do erro view-dependente que é calculado a cada quadro.

O processo de adaptação é iniciado a partir da malha base. Nesse estágio são apenas permitidas as operações de refinamento. Uma operação de refinamento consiste em adicionar um vértice a malha subdividindo uma aresta e criando quatro faces quando o vértice pertence ao interior da malha e duas faces quando pertence a borda. Quando todos os vértices são adicionados, a malha atinge o nível máximo de resolução. Neste caso são permitidas apenas as operações de simplificação. A operação de simplificação é a operação inversa da operação de refinamento, remove um vértices e elimina as arestas e faces que foram criadas durante a sua inserção. Essas operações são realizadas obedecendo restrições topológicas para garantir que a nova malha sempre seja válida.

A fila de prioridade tem o objetivo de determinar a ordem as operações de simplificação e refinamento. A implementação mais comum utiliza duas filas de prioridade uma para cada operação. A fila de simplificação contém os vértices que devem ser retirados da malha. A fila de refinamento contém as arestas que devem ser subdivididas.

A ordem dos elementos de cada fila é determinada pela função adaptação. Os elementos são ordenados em ordem decrescente, ou seja, o primeiro elemento tem maior prioridade que os demais. A ordem a qual as filas são acessadas (ordem das operações de refinamento e simplificação) é baseada no algoritmo guloso. O algoritmo guloso sempre acessa uma fila enquanto a prioridade de seus elementos for maior a prioridade dos elementos da outra fila. Os acessos as filas de simplificação e refinamento são realizados até que a malha final represente o mundo real

dentro de um erro máximo tolerado. O erro máximo pode ser definido como estático ou dinâmico.

O erro estático é especificado pelo usuário antes de iniciar a visualização e permanece inalterado até que o usuário determine um outro valor. A vantagem dessa abordagem é a facilidade de implementação. As desvantagens são a dificuldade de especificar um valor de erro que dê uma boa relação entre qualidade visual e desempenho e a grande dependência entre a especificação do erro a o tipo de hardware onde o sistema está rodando.

O erro dinâmico é determinado automaticamente pelo sistema para satisfazer restrições como quatros por segundos, número de triângulos por quadro ou foco do usuário. Neste caso o sistema tenta estimar o erro máximo para que a restrições determinado pelo usuário sempre seja obedecida. Assim, o valor do erro varia dinamicamente como uma consequência do sistema tentar se adequar ao modo de visualização e as diferenças de complexidades entre as regiões da malha. As vantagens de utilizar o erro dinâmico é que é mais intuitivo para o usuário especificar valores para as restrições e que permite ao sistema se adequar mais facilmente a diferentes configurações de hardware sem a intervenção do usuário.

As técnicas de erro estático e dinâmico podem ser combinadas durante o processo de visualização. O modo de visualização é chamado de *iterativo* quando os parâmetros da câmera estão variando no tempo, caso contrário o modo de visualização é chamado de *progressivo* (câmera em repouso). No modo iterativo é mais adequado a utilização do erro dinâmico, enquanto que no modo processivo o erro estático é o adequado.

O problema de visualizar grandes volumes de dados utilizando uma quantidade limitada de recursos de armazenamento (memórias de textura, RAM) é resolvido a partir do desenvolvimento de sistemas gerenciamento de memória.

O modelo de gerenciamento de memória mais utilizado é o sistema paginado. No modelo paginado os recursos de armazenamento são particionado em blocos de mesmo tamanho. Esse modelo é ideal para gerenciar dados que podem ser particionados uniformemente. Outra característica muito importante é que possui baixa complexidade de implementação.

A qualidade visual do processo de visualização está intimamente relacionado com o desempenho do sistema de gerenciamento de memória. A eficiência de um sistema de memória está relacionada à capacidade de tornar disponível a maior quantidade de dados possível necessária para gerar um quadro. Esse requisito é atingindo a partir do carregamento antecipado dos dados. Os sistemas que realizam carregamento antecipados são chamados de sistema preditivos. A operação de predição explora

coerência espacial e temporal para determinar os dados que deverão ser carregados num futuro próximo [20].

4 Implementação

Nesta seção descrevemos os aspectos relevantes da implementação da primeira versão do nosso sistema. O método é view-dependent e possui view frustum culling e geomorphing. At this point, estamos interessados apenas em ressaltar a flexibilidade e poder da estrutura híbrida usada em nossa plataforma: a malha 4-8 semi-regular (subseção 4.1). Assim, frustum culling, geomorphing, bem como os cálculos de erros geométricos e dos raios de esferas envolventes feitos na etapa de pré-processamento, são semelhantes aos descritos em Lindstrom [18]. A subseção 4.2 resume o algoritmo top-down implementado e mostra um esboço do que seria o algoritmo incremental, ainda em desenvolvimento.

4.1 Malhas Semi-Regulares 4-8

A malha 4-8 é um caso particular das malhas $4-k$ [25] que são uma especialização das estruturas de resolução variável introduzidas independentemente por De Berg et al [7] e Puppo et al [21]. Classificando a malha 4-8 com relação ao tipo de subdivisão 4-8, obtemos as malhas 4-8 semi-regulares, que é uma tesselação que contém vértices extraordinários isolados com valência diferente de 4 ou 8 [25].

Essa malha é criada a partir de uma malha irregular inicial, com a estrutura de quadrangulação triangulada [22], aplicando um método de refinamento 4-8 semi-regular que introduz somente vértices regulares [24]. Desse modo, como a malha é refinada, vértices extraordinários da malha inicial são cercados por vértices regulares com valência 4 ou 8.

A estrutura de resolução variável [25] de uma malha 4-8 semi-regular tem propriedades ótimas, de acordo com os critérios discutidos em Puppo et al [21]: *alto poder expressivo, profundidade logarítmica e crescimento linear*.

Alem disso, a malha 4-8 semi-regular pode ser representada por uma estrutura sem ponteiros, o que implica em uma representação compacta.

Em nossa plataforma usamos a biblioteca A48 [1] descrita por Velho em [23] que tem como base a implementação da estrutura de multi-resolução da malha 4-8 semi-regular. A48 é uma biblioteca de malha adaptativa dinâmica que mantém a conforming triangulation of time-varying surfaces. Suas principais características são:

- A malha é representada pela estrutura de dados topológica half-edge. A implementação da nova funcionalidade de multiresolução adaptativa não requer nenhum custo extra na representação. Alem

disso, como a half-edge é amplamente usada, nosso sistema é fácil de ser usado em muitas aplicações.

- Elementos topológicos são vértice, aresta e face.
- Contém apenas dois operadores topológicos implementados pelas operações de simplificação e refinamento. Simplificação e Refinamento são operações estelares [26], simples de implementar e garantidas de funcionar.
- A estrutura de malha adaptada muda dinamicamente e sua resolução é baseada no critério definido pelo usuário. Isto torna a adaptação muito geral e poderosa. O critério de adaptação é associado com a estrela de vértices e arestas (note que essa região da malha é a mesma para simplificação e refinamento).

A definição das operações estelares proporcionou o desenvolvimento de uma interface simples para as operações de adaptação da malha (simplificação e refinamento). Essa interface permitiu que as operações de culling, de geomorphing e de cálculos da função de erro fossem facilmente incorporados no algoritmo de adaptação da malha.

4.2 Algoritmo

Como explicado na Seção 3, alguns cálculos precisam ser feitos numa fase de pré-processamento no intuito de minimizar a quantidade de dados a ser processada a runtime. Desta fase obtemos como input para a etapa de execução o mapa de erros geométricos e o mapa de raios das esferas envolventes, calculados de modo semelhante a Lindstrom [18].

Nosso algoritmo é guloso e parte do princípio que os elementos topológicos, nesse caso vértices e arestas, podem ser ordenados a partir da função de erro. A estrutura de dados utilizada to perform this task is the priority queue. Vértices são armazenados na fila de simplificação e arestas na fila de refinamento. Então, a partir de uma malha inicial o algoritmo de adaptação pode retirar vértices da malha e colocar arestas na priority queue (Operação de Simplificação), ou retirar arestas da fila e colocar vértices na malha (Operação de Refinamento).

A seguir resumimos o algoritmo top-down que utiliza apenas a priority queue de refinamento. Depois descrevemos uma abordagem ainda em desenvolvimento para um algoritmo incremental que usa ambos heaps de simplificação e refinamento.

Algoritmo Top-Down:

1. Le os arquivos de entrada: mapa de altura, mapa de erros geométricos, mapa de raios das esferas envolventes.

2. Para cada frame:
 - (a) Inicializa a malha base;
 - (b) Le dados da camera que sao os parametros dependentes da visualizacao.
 - (c) Preenche a priority queue de refinamento utilizando o erro geometrico, a distancia do vertice ao observador e o raio da sua esfera envolvente. Assim, as arestas com maior error terao prioridade de refinamento.
 - (d) A priority queue e' lida e para cada aresta, se ela passa no teste de culling e no criterio de adaptacao, entao e' refinada e o novo vertice e' adicionado 'a malha. A coordenada z desse novo vertice no mundo real 'e determinada a partir do criterio de geomorphing.
 - (e) Desenha a malha corrente.

Algoritmo Incremental:

1. Le os arquivos de entrada: mapa de altura, mapa de erros geometricos, mapa de raios das esferas envolventes.
2. Inicialmente preenche a priority queue de refinamento usando os mesmos criterios do algoritmo anterior e faz um primeiro refinamento. Cada aresta refinada produz um novo vertice que e' inserido no heap de simplificacao.
3. Para cada frame:
 - (a) Le dados da camera que sao os parametros dependentes da visualizacao.
 - (b) Le o heap de simplificacao e, de acordo com o criterio de adaptacao, simplifica os vertices que passarem no teste, retirando-os da malha e insere as novas arestas no heap de refinamento.
 - (c) Em seguida le o heap de refinamento, refinando as arestas que passarem no teste.
 - (d) Desenha a malha corrente.

5 Conclusão

O objetivo principal deste trabalho foi mostrar a flexibilidade e poder da estrutura hibrida usada em nossa plataforma: a malha 4-8 semi-regular. Assim, frustum culling, geomorphing, bem como os calculos de erros geometricos e dos raios de esferas envolventes feitos na etapa de pre-processamento, são baseados nos descritos em Lindstrom [18].

A representação em multi-resolução do terreno é obtida a partir das operações estelares de simplificações

e refinamento que são aplicadas sobre a malha. Essas operações são realizadas com o objetivo de construir adaptativamente uma malha que represente adequadamente o terreno em cada quadro.

A função de erro descrita no trabalho de Lindstrom é baseada em métrica saturada. A métrica saturada é ideal para construir a malha utilizando a técnica top-down. Assim, para cada quadro é necessário iniciar com a malha base. Em seguida, refinar a malha base (reconstruir toda a estrutura topológica) até obter a malha adequada definida a partir da função de erro.

Apesar de não usar o método incremental que seria mais eficiente, o desempenho do sistema foi bastante satisfatório atingindo no mínimo 30 quadro por segundo em situações onde a região da malha possuía uma geometria complexa.

Os resultados foram obtidos com uma implementação inicial do sistema, ainda existe a possibilidade de melhorar ainda mais o desempenho da visualização, como por exemplo utilizar a técnica de strip triangle. Outra questão que ser mencionada é que em nenhum momento foi explorado o fato de que a topologia da malha é armazenada explicitamente. As informações topológicas podem ser utilizadas para o desenvolvimento de um algoritmo de adaptação incremental, em oposição a adaptação top-down que foi utilizada para gerar os resultados, e também em outros tipos de procedimento, como por exemplo cálculos na malha.

5.1 Sumario

Este trabalho mostrou que os dados de terreno podem ser representados a partir de malhas semi-regulares. A vantagem de utilizar malhas semi-regulares é que permite representar a maioria dos objetos do mundo real de forma única e transparente. A malha semi-regular é baseada na estrutura hierárquica chamada malha 4-8 semi-regular. A malha 4-8 semi-regular possui as características ideais para representação em multi-resolução. O método de construção da malha é baseada nas operações estelares de refinamento e simplificação.

A definição das operações estelares proporcionou o desenvolvimento de uma interface simples para as operações de adaptação da malha (simplificação e refinamento). Essa interface permitiu que as operações de culling, de geomorphing e de cálculos da função de erro fossem facilmente incorporados no algoritmo de adaptação da malha.

5.2 Pesquisa em Andamento

Devido a coerência temporal e espacial existente durante o processo de visualização, a diferença entre as malhas geradas em um conjunto de quadros consecutivos é muito

pequena. Assim, na adaptação incremental apenas as regiões que estiverem fora do critério de erro é que são adaptadas. Assim, o desenvolvimento de um algoritmo incremental é essencial para o aumento do desempenho do sistema de visualização.

Hoje, já é possível obter dados de terreno e imagens de satélite de todo o globo terrestre. A visualização em tempo-real dessa grande massa de dados só é viável a partir do desenvolvimento de um sistema de gerenciamento de memória. O sistema de memória ideal deve suportar multiresolução e carregamento preditivo [20].

Este trabalho ainda encontra-se no início do seu desenvolvimento, assim não foi possível fazer experimentos para comparar a técnica aqui proposta com outras técnicas de representação e visualização de dados de terreno. Com o amadurecimento do sistema de visualização será possível fazer um estudo de caso detalhado com o objetivo de medir o desempenho do nosso sistema em relação aos outros sistemas já propostos nessa área.

References

- [1] A48 mesh library. <http://w3.impa.br/lvelho/a48>.
- [2] Keyhole. <http://www.keyhole.com/>.
- [3] TerraFly. <http://terrafly.com/>.
- [4] Terralib. <http://terralib.dpi.inpe.br>.
- [5] Terraserver. <http://terraserver.microsoft.com/>.
- [6] Paolo Cignoni, Fabio Ganovelli, Enrico Gobbetti, Fabio Marton, Federico Ponchio, and Roberto Scopigno. BDAM – batched dynamic adaptive meshes for high performance terrain visualization. *Computer Graphics Forum*, 22(3):505–514, September 2003.
- [7] Mark de Berg and Katrin Dobrindt. On level of details in terrains. In *Proc. 11th Annual ACM Symp. on Computational Geometry*, Vancouver, B.C., June 1995.
- [8] Mark A. Duchaineau, Murray Wolinsky, David E. Sieti, Mark C. Miller, Charles Aldrich, and Mark B. Mineev-Weinstein. ROAMing terrain: real-time optimally adapting meshes. In *IEEE Visualization*, pages 81–88, 1997.
- [9] G. Camara et al. Terralib: Technology in support of gis innovation. *II Workshop Brasileiro de Geoinformatica, GeoInfo*, 2000.
- [10] G. Camara et al. Design patterns in gis development: The terralib experience. *III Workshop Brasileiro de Geoinformatica, GeoInfo*, 2001.
- [11] Hugues Hoppe. View-dependent refinement of progressive meshes. *Computer Graphics*, 31(Annual Conference Series):189–198, 1997.
- [12] Hugues H. Hoppe. Smooth view-dependent level-of-detail control and its application to terrain rendering. In David Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 35–42, 1998.
- [13] F. Morando L. De Floriani, P. Magillo and E. Puppo. Dynamic view-dependent multiresolution on a client-server architecture. *Computer-Aided Design*, Volume 32(Issue 13):805–823, November 2000.
- [14] P. Magillo L. De Floriani, E. Puppo. A formal approach to multiresolution modeling. In *Geometric Modeling: Theory and Practice*, pages 302–323. Springer-Verlag, 1997.
- [15] P. Magillo L. De Floriani, E. Puppo. VARIANT: A system for terrain modeling at variable resolution. *GeoInformatica*, 4(3):287–315, 2000.
- [16] P. Lindstrom, D. Koller, W. Ribarsky, L. Hodges, N. Faust, and G. Turner. Real-time continuous level of detail rendering of height fields. *Proceedings of SIGGRAPH'96*, pages 109–118, 1996.
- [17] P. Lindstrom and V. Pascucci. Visualization of large terrains made easy. In *Proceedings of the Conference on Visualization '01*, pages 363–371. IEEE Computer Society, 2001.
- [18] Peter Lindstrom and Valerio Pascucci. Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):239–254, 2002.
- [19] Renato Pajarola, Marc Antonijuan, and Roberto Lario. Quadtree based triangulated irregular networks. In *Proceedings of the Conference on Visualization '02*, pages 395–402. IEEE Computer Society, 2002.
- [20] Sergio Pinheiro and Luiz Velho. A virtual memory system for real-time visualization of multi-resolution 2d objects. In *Proceedings of WSCG*, 2002.
- [21] E. Puppo. Variable resolution triangulations. *Computational Geometry Theory and Applications*, 11(3-4):219–238, 1998.
- [22] Luiz Velho. Quadrangulation using 4-8 clustering. In Luiz Eloy Vaz, editor, *21st Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE 2000)*, IMPA, Rio de Janeiro, Dezembro 2000.

- [23] Luiz Velho. A dynamic adaptive mesh library based on stellar operators. *Journal of Graphics Tools*, 9(2):1–29, 2004.
- [24] Luiz Velho and Jonas Gomes. Semi-regular 4-8 refinement and box spline surfaces. Preprint, 1999.
- [25] Luiz Velho and Jonas Gomes. Variable resolution 4-k meshes: Concepts and applications. *Computer Graphics Forum*, 19(4):195–214, 2000.
- [26] Antônio Wilson Vieira, Luiz Velho, Hã©lio Lopes, Geovan Tavares, and Thomas Lewiner. Fast stellar mesh simplification. In *16th Brazilian Symposium on Computer Graphics and Image Processing*, S ao Carlos, SP, October 2003.