

ANSELMO ANTUNES MONTENEGRO

**INVESTIGAÇÃO DE NOVOS CRITÉRIOS PARA INSERÇÃO DE
PONTOS EM MÉTODOS PARA SIMPLIFICAÇÃO DE
MODELOS DE TERRENO ATRAVÉS DE REFINAMENTO**

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA

Rio de Janeiro, 27 de agosto de 1997

ANSELMO ANTUNES MONTENEGRO

**INVESTIGAÇÃO DE NOVOS CRITÉRIOS PARA INSERÇÃO
DE PONTOS EM MÉTODOS PARA SIMPLIFICAÇÃO DE
MODELOS DE TERRENO ATRAVÉS DE REFINAMENTO**

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA

PUC-Rio

Rio de Janeiro, 27 de agosto de 1997

ANSELMO ANTUNES MONTENEGRO

**INVESTIGAÇÃO DE NOVOS CRITÉRIOS PARA INSERÇÃO
DE PONTOS EM MÉTODOS PARA SIMPLIFICAÇÃO DE
MODELOS DE TERRENO ATRAVÉS DE REFINAMENTO**

Dissertação apresentada ao Departamento
de Informática da PUC-Rio como parte
dos requisitos para a obtenção do título de
Mestre em Ciências em Informática.
Orientador: Paulo Cezar Pinto Carvalho
Co-orientador: Marcelo Gattass

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 27 de agosto de 1997

AGRADECIMENTOS

- A todos os colegas do TeCGraf e do departamento de informática da PUC-Rio, por todo tipo de ajuda que me foi prestado.
- Aos professores que tive ao longo de meus anos de estudo, pelos valiosos ensinamentos que me foram fornecidos.
- Aos meus orientadores Paulo César Pinto Carvalho e Marcelo Gattass por tudo que fizeram para que este trabalho pudesse ser concluído e pelo apoio nos momentos difíceis.
- À CAPES pelo apoio financeiro, sem o qual este trabalho não se realizaria.
- À Geiza Maria Hamazaki da Silva pelo apoio, compreensão, carinho e ajuda.
- Aos meus tios Nelson de Oliveira Carneiro e Sueli Antunes Carneiro pelo incentivo durante toda minha vida.
- A minha avó Honorina Pereira da Costa por sua eterna atenção e preocupação.
- A minha irmã Henriette Antunes Montenegro e meus primos Cíntia Antunes Carneiro e Felipe Antunes Carneiro, por todos os momentos de descontração e alegria.
- Aos meus pais, os quais são responsáveis por tudo que tenho e que sou nesta vida.
- A DEUS.

Resumo

Este trabalho tem por objetivo estudar algoritmos para a obtenção de representações simplificadas para modelos de terreno. Dados de terreno são considerados de difícil tratamento devido à sua complexidade e dimensão. Uma das técnicas mais utilizadas para amenizar a complexidade dos dados de terreno são os métodos de simplificação de superfícies. Dentre os vários métodos de simplificação de superfícies, os que têm fornecido melhores resultados são os métodos de simplificação por inserção gulosa baseada no erro vertical local. Apesar de suas qualidades, percebemos algumas falhas na qualidade perceptual das soluções fornecidas por este método quando aplicado a dados de terreno com regiões nas quais as altitudes apresentam diferentes graus de variabilidade. Neste trabalho investigamos a existência de métodos melhores para abordagem deste problema. Propomos métodos de inserção gulosa que utilizem critérios de seleção em que o erro vertical local seja modificado de acordo com a variabilidade da superfície.

Abstract

In this work we study algorithms for obtaining simplified representations for terrain models. Terrain data is difficult to deal with, due to its complexity and dimension. Simplification methods are one of the techniques used to reduce the natural complexity of real terrain data. Among several simplification methods, greedy insertion with vertical local error measure is the one which has provided best results. In spite of its qualities, the method presents some shortcomings, especially when applied to terrain with different height regions. We investigate the existence of better measures to deal with such problem and propose the adoption of greedy insertion methods in which the local vertical error is modified according to the variability of the surface in a neighborhood of the point under consideration.

Sumário

1. Introdução	1
1.1 Organização da tese	2
2. Modelos de Terreno	3
2.1 Introdução	3
2.2 Definição	3
2.3 Aplicações	6
2.4 Visualização de Modelos de Terreno	8
2.4.1 Estratégias para terrenos representados por mapas de altura.	8
2.4.2 Estratégias baseadas em Triangular Irregular Networks(TINs)	9
3. Representação de terrenos por triangulações.	11
3.1 Introdução	11
3.2 Triangulações	11
3.3 Representações de triangulações	13
4. Triangulações de Delaunay	16
4.1 Introdução	16
4.2 Definição	16
4.3 Descrição do algoritmo	18
5. Métodos de Simplificação de Superfícies.	24
5.1 Introdução	24
5.2 Origem dos métodos de simplificação de superfícies	24
5.3 Simplificação de Curvas	24
5.4 Simplificação de superfícies.	25
6. O Método de Refinamento por Inserção Gulosa	30
6.1 Introdução	30
6.2 Descrição do método	30
6.3 Critério de Seleção de Pontos.	31
6.4 Critérios de parada	34
6.5 Versão seqüencial versus versão paralela.	35
7. Comparação de critérios de seleção no método de inserção gulosa	37
7.1 Introdução	37
7.2 Método de inserção gulosa com critério de seleção baseado no erro vertical local.	37
7.3 Algoritmo	39
7.4 Exemplos de aplicação do algoritmo a dados de terreno	40

7.5	Análise do comportamento do critério de seleção baseado no erro vertical local	42
7.6	Critério de seleção baseado no erro vertical local modificado	44
7.6.1	Determinação do parâmetro de ponderação	45
7.7	Parâmetro de ponderação proporcional ao desvio padrão	47
7.7.1	Análise dos resultados	56
7.8	Parâmetro de ponderação proporcional à norma da segunda derivada.	57
7.8.1	Análise dos resultados	60
8.	Conclusões	62
8.1	Propostas para trabalhos futuros:	63
9.	Apêndice: Quad-edge data structure	65
9.1	Definição da quad-edge	65
9.2	Operadores Topológicos Básicos	67
9.2.1	Operador MakeEdge()	67
9.2.2	Operador Splice	68
9.3	Algoritmo de inserção gulosa utilizando a estrutura quad-edge	70
9.3.1	Operadores básicos	70
9.3.2	Operadores geométricos	70
9.3.3	Algoritmo de inserção de ponto em um triangulação de delaunay.	72
9.3.4	Algoritmo de localização de um ponto na triangulação	72

Lista de figuras

Figura 2.1- Campo de alturas _____	3
Figura 2.2 - Dados de terreno simplificados _____	5
Figura 2.3 - Terreno visualizado pelo Algoritmo de Horizonte Flutuante _____	9
Figura 2.4 - Terreno visualizado através de uma TIN texturizada _____	10
Figura 3.1-Interpolação em R _____	11
Figura 3.2 - Interpolação em R^2 _____	12
Figura 3.3 - Interpolação em R^2 _____	12
Figura 4.1 - Propriedade geométrica da triangulação de Delaunay _____	16
Figura 4.2 - Triangulação inicial em amostragem regular _____	18
Figura 4.3 - Triangulação inicial em amostragem irregular _____	19
Figura 4.4- Localização do ponto _____	20
Figura 4.5 - Ponto sobre uma aresta _____	20
Figura 4.6 - Criação de novas arestas nos casos 1 e 2 _____	21
Figura 4.7 - As novas arestas satisfazem o critério de Delaunay _____	21
Figura 4.8 - Aplicação do teste de Delaunay _____	22
Figura 4.9 - Exemplo de aplicação do teste de Delaunay _____	23
Figura 5.1 - Douglas Peucker (nível 1) _____	25
Figura 5.2 - Douglas Peucker (nível 2) _____	25
Figura 5.3 - Critério de rugosidade _____	28
Figura 6.1 - Medida de erro vertical local _____	32
Figura 6.2 - Convergência não monotônica do erro vertical local _____	32
Figura 7.1 - Visualização de "Cvzbuffalo" simplificado _____	40
Figura 7.2 - Gráfico erro x # pontos da simplificação de "Cvzbuffalo" _____	41
Figura 7.3 - Visualização de "Ilha" simplificado _____	41
Figura 7.4 - Gráfico erro x # pontos inseridos da simplificação de "Ilha" _____	42
Figura 7.5 - Terreno com regiões com características distintas _____	42
Figura 7.6 - Resultado de simplificação por critério de seleção global _____	43
Figura 7.7- Definição da zona de influência _____	46
Figura 7.8 - medida de importância baseada puramente no erro vertical local - Terreno: "Cvzbuffalo" _____	48
Figura 7.9 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=1$ - Terreno: "Cvzbuffalo" _____	49

Figura 7.10 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=5$ - Terreno: "Cvzbuffalo" _____	50
Figura 7.11 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=10$ - Terreno: "Cvzbuffalo" _____	51
Figura 7.12 - medida de importância baseada puramente no erro vertical local - Terreno: "Ilha" _____	52
Figura 7.13 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=1$ - Terreno: "Ilha" _____	53
Figura 7.14 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=5$ - Terreno: "Ilha" _____	54
Figura 7.15 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=10$ - Terreno: "Ilha" _____	55
Figura 7.16 - medida de importância baseada no erro vertical local ponderado pela norma da derivada segunda - Terreno : "Cvzbuffalo" _____	58
Figura 7.17 - medida de importância baseada no erro vertical local ponderado pela norma da derivada segunda - Terreno : "Ilha" _____	59
Figura 7.18 - Terreno simplificado com critério baseado no erro vertical local ____	61
Figura 7.19 - Terreno simplificado com critério de seleção baseado no erro vertical local ponderado pela segunda derivada _____	61
Figura 9.1 - Algebra de arestas (edge algebra) _____	66
Figura 9.2 - Resultado do operador MakeEdge() _____	68
Figura 9.3 - Resultado do operador Splice(a,b) _____	69

Lista de tabelas

Tabela 7.1 - Características dos dados de terrenos testes _____	40
Tabela 7.2 - Resultados da aplicação do método aos dados de "Cvzbuffalo" _____	40
Tabela 7.3 - Resultado dos testes aplicados aos dados de "Ilha" _____	41
Tabela 7.4 - Resultado do método com critério baseado no erro vertical local máximo sobre "Cvzbuffalo" _____	43
Tabela 7.5 - Resultados da aplicação da medida de variabilidade baseada no desvio padrão sobre "Cvzbuffalo" _____	47

1. Introdução

Este trabalho consiste em um estudo sobre métodos de representação e algoritmos para a modelagem de dados de terreno.

A importância dos dados de terreno pode ser notada em diferentes áreas nas mais diversas aplicações como jogos, aplicações militares e de engenharia, Sistemas de Informação Geográfica, etc.

Dados de terreno são considerados de difícil tratamento devido a grande quantidade de informação necessária para a geração de modelos digitais. A gigantesca massa de dados necessária a representação digital de dados de terreno é resultante da complexidade existente na superfície de um terreno natural. Até o momento não existem modelos compactos capazes de capturar precisamente as nuances de um terreno existente na natureza.

A forma mais difundida de representação de um terreno em meio digital são os mapas de elevação. Os mapas de elevação associados a dados de terreno reais podem assumir valores gigantescos, tornando muitas vezes inviável a sua utilização. Uma das técnicas que procura amenizar as dificuldades que surgem devido as dimensões dos dados de terreno são os chamados **métodos de simplificação de superfícies**.

Estes métodos procuram simplificar os dados através da eliminação das redundâncias existentes no conjunto de amostras que compõem um terreno. Em geral, os métodos de simplificação de superfícies utilizam malhas triangulares irregulares (TINs) para representar os dados de terreno, pois estas são capazes de se adaptar ao terreno utilizando o mínimo de redundância possível. Alguns métodos trazem embutidos mecanismos de geração automática de TINs, chegando a tal ponto que os dois processos acabam se fundindo em um só. Já outros requerem que uma malha tenha sido construída previamente antes de sofrer a simplificação. O estudo da geração de TINs e simplificação de superfícies aplicados a modelagem de terrenos são os principais temas deste trabalho.

Neste trabalho também apresentamos uma abordagem alternativa para métodos de simplificação de superfícies, em que propomos a utilização de novas medidas de importância das amostras nos métodos de simplificação por inserção gulosa. A utilização de tais critérios visa cobrir a deficiência dos métodos guiados unicamente pelo erro máximo local, que muitas vezes deixam de incluir pontos importantes nos estágios menos refinados simplesmente por não serem necessários para satisfazer a tolerância estabelecida.

1.1 Organização da tese

Este trabalho está organizado em capítulos que estão descritos resumidamente nos parágrafos abaixo.

No capítulo 1 conceituamos Modelo Digital de Terreno e descrevemos suas aplicações em diversas áreas e as estratégias de visualização e representação.

No capítulo 2 apresentamos os conceitos básicos sobre triangulações e discutimos estruturas para representações de TINs.

No capítulo 3 descrevemos a triangulação de Delaunay e o algoritmo incremental para obter a triangulação de Delaunay.

No capítulo 4 abordamos em detalhes os métodos de simplificação de superfícies, definindo o problema de simplificação. Descrevemos também as técnicas mais utilizadas para o desenvolvimento de métodos de simplificação.

No capítulo 5 estudamos os métodos de inserção gulosa, abordando em detalhes os mecanismos que regem seu funcionamento: o critério de seleção de pontos e o critério de parada. Descrevemos também versões paralelas do algoritmo e os resultados provenientes da literatura afim.

No capítulo 6 fazemos uma comparação entre diversos critérios de seleção de pontos. Nas primeiras seções abordamos o método de inserção gulosa com critério de seleção baseado no erro vertical local, discutindo detalhes de implementação e apresentando alguns resultados. Em seguida analisamos tais resultados e propomos novos critérios de seleção em que o erro vertical local é modificado de acordo com a variabilidade da superfície. Avaliamos dois critérios de seleção baseados nessas idéias, comparando seus resultados e analisando seu desempenho.

Finalmente, no capítulo 7 apresentamos as principais conclusões obtidas através do trabalho e propomos temas para trabalhos futuros.

2. Modelos de Terreno

2.1 Introdução

Neste capítulo abordaremos os principais conceitos sobre modelos de terreno que são a base do estudo desenvolvido neste trabalho. Nos parágrafos seguintes definiremos modelos de terreno intuitivamente e matematicamente. Citaremos diversas aplicações de modelos de terreno classificadas segundo às principais áreas em que são utilizadas. Em seguida, descreveremos sucintamente os principais métodos utilizados para a representação e visualização de dados de terreno.

2.2 Definição

Modelos de terreno podem ser compreendidos como uma representação digital de uma parte da superfície terrestre. Um terreno pode ser visto com uma superfície se assumirmos que cada ponto nele existente possui um único valor de altura. Através desta abstração, descartamos a presença de cavernas, grutas, e qualquer outra estrutura de maior complexidade que não satisfaça a restrição descrita acima. Os modelos de terreno constituem um caso específico de campos de alturas (*height fields*) (Figura 2.1). Um campo de alturas é definido por um conjunto bidimensional de amostras de altura de uma superfície. Matematicamente, podemos representar um campo de alturas por uma função $e: D \subset R^2 \rightarrow R$, chamada função de elevação.

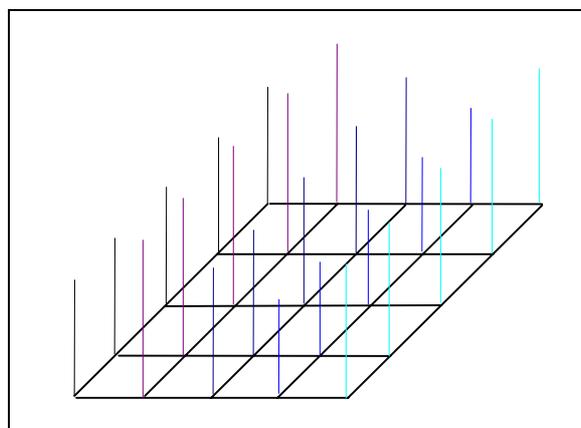


Figura 2.1- Campo de alturas

O tratamento de dados de terreno que não podem ser modelados como campos de altura é bastante complexo, e a especificação de estruturas de representação e formas de processamento destes dados consistem em um problema em aberto na literatura afim. Neste trabalho serão abordados somente dados de terreno modelados como campos de altura.

A fonte mais comum de dados de terreno é o DEM (*Digital Elevation Model*), fornecido pelo U.S Geological Survey, que dispõe de um mapeamento quase completo de toda superfície dos E.U.A. Um DEM consiste basicamente em um vetor em ponto flutuante de valores de altura. Um outro formato similar é o *DTED (Digital Terrain Elevation Data)* que é adotado por estabelecimentos militares dos Estados Unidos. Os DTEDs contém mapeamentos de outras regiões além do próprio território americano.

Estas estruturas introduzem uma quantidade enorme de redundâncias, não sendo adequadas para armazenamento de dados de terreno muito grandes. Em geral estes dados passam por um processo de **simplificação** antes de serem armazenados (Figura 2.2).

O processo de simplificação procura representar a superfície determinada pelos dados de elevação de um terreno através de uma outra superfície aproximada definida por um subconjunto das amostras pertencentes aos dados originais. Em geral a superfície aproximada é determinada por uma interpolação linear por partes do subconjunto dos dados originais. O subconjunto é escolhido de forma que contenha somente as amostras necessárias para que a superfície aproximada represente os dados originais com um certo grau de precisão, eliminado assim uma grande quantidade de redundâncias.

Quando for necessária a utilização dos dados simplificados em alguma aplicação que requeira uma amostragem regular, deve ser realizada uma reamostragem da superfície aproximada.

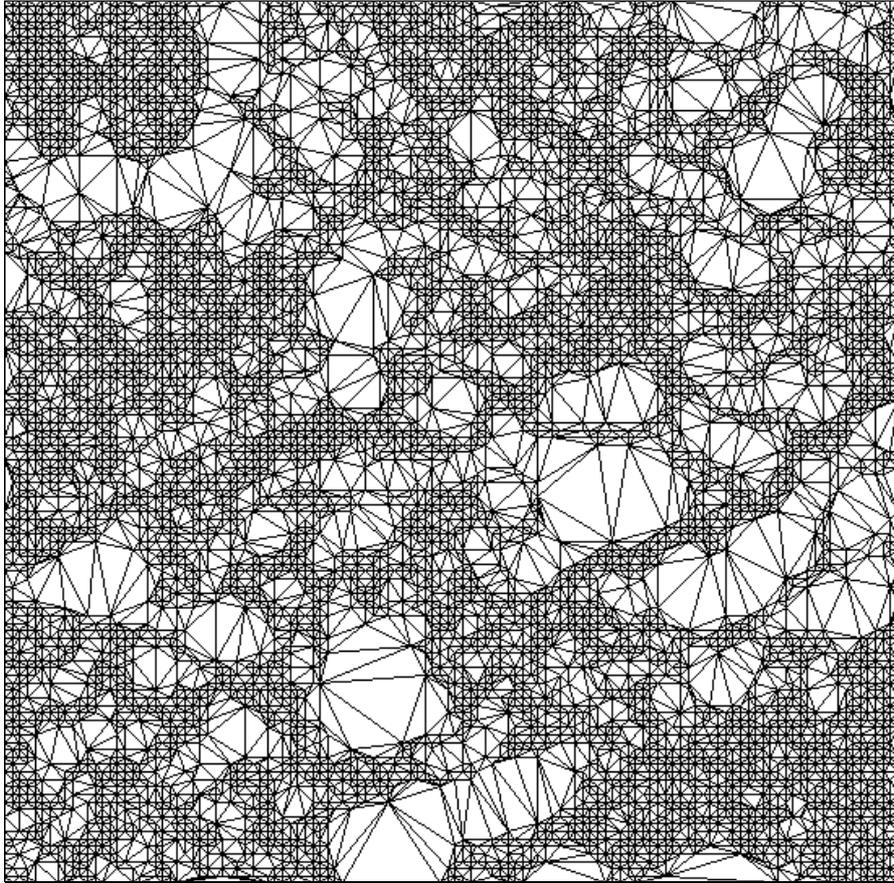


Figura 2.2 - Dados de terreno simplificados

2.3 Aplicações

Devido ao crescente avanço nas técnicas de sensoriamento remoto e construção de sistemas de modelagem, diversas aplicações têm utilizado dados de terreno. Estas aplicações englobam desde jogos para entretenimento até aplicações mais sérias, como Sistemas de Informação Geográfica (SIG), engenharia civil, planejamento e gerenciamento de recursos, Ciências da Terra, aplicações militares, simuladores de voo, robótica, computação gráfica, etc.

Sistemas de Informação Geográfica

Os modelos de terreno fazem parte do conjunto de dados básicos utilizados pelos Sistemas de Informação Geográfica. Uma série de aplicações em SIG utilizam dados de terreno, como análises de declividade, análise de visibilidade e análise multi-critério. Em geral, os SIGs fornecem diversas ferramentas para tratamento de dados de terreno, como métodos de triangulação, métodos para geração de grades regulares, ferramentas para obtenção de DEMs a partir de curvas de nível, ferramentas para georeferenciamento, rendering, visualização iterativa e análises.

Engenharia civil

Os modelos de terreno são utilizados em engenharia civil em aplicações como projeto de estradas, aeroportos, construções etc. A característica principal dos dados de terreno utilizados em engenharia civil é que o volume de dados é bem menor do que o das demais aplicações. A ênfase funcional é dada a cálculos volumétricos e de projeto. Uma metodologia para posicionamento visual de prédios em terrenos acidentados pode ser encontrado em [JG95]. Este mesmo trabalho contém um material introdutório sobre modelagem, aquisição de dados de terreno e obtenção de modelos poliedrais a partir de amostras. O autor do mesmo trabalho desenvolveu um sistema baseado na metodologia proposta denominado **SYSTER**.

Planejamento e gerenciamento de recursos.

Este é um dos principais domínios de aplicação de dados de terreno, envolvendo diversas sub-áreas como planejamento urbano e ambiental, sensoriamento remoto, agronomia, planejamento florestal, meteorologia e climatologia. Todas estas disciplinas estão centradas no gerenciamento de recursos naturais. Aplicações típicas incluem estudos de impacto ambiental; localização de polos industriais; correção radiométrica e geométrica de imagens obtidas via sensoriamento remoto; suporte à

classificação de imagens, modelagem de processos de erosão do solo; desenvolvimento de estratégias de colheita e modelos de dispersão populacional. O aspecto funcional destas aplicações inclui ferramentas para interpretação, visualização e análise. Outra característica importante é a ligação entre modelagem de terreno e funções bidimensionais de SIGs, como processamento de polígonos, malhas e dados *raster*.

Ciências da Terra

Ciências da Terra incluem as áreas de geologia, geomorfologia, hidrologia e glaciologia. Estas áreas requerem funções especiais para modelagem e interpretação de descontinuidades em terrenos. Podemos citar como exemplos as seguintes aplicações: monitoramento de drenagens; simulação e classificação geomorfológica; interpretação e mapeamentos geológicos.

Aplicações Militares

As aplicações militares combinam aspectos de diversas áreas. O terreno é considerado um dos principais componentes de um ambiente militar. As agências militares são fontes produtoras de dados de terreno por excelência, dando ênfase considerável em metodologias para captura de dados (correlação estereográfica de modelos digitais de terreno e digitalização automática de curvas de nível). O uso militar de modelos de terreno envolve operações de planejamento bastante similares às de engenharia civil. O gerenciamento de campos de batalha envolve tarefas como análise de intervisibilidade, controle de tráfico de veículos e controle automático de mísseis. As aplicações militares também requerem técnicas avançadas de visualização que possibilitem *display* e animação foto-realísticas para simulações de voo.

Simulação de voo, robótica e computação gráfica.

Estas áreas estudam técnicas de modelagem, manipulação e visualização de dados de terreno para fornecer suporte às aplicações das demais áreas descritas acima.

2.4 Visualização de Modelos de Terreno

A visualização é um dos aspectos mais importantes nos sistemas que utilizam dados de terreno. Os resultados das operações de modelagem de dados de terreno são comunicados ao usuário através de forma gráfica. Existem diversas estratégias de visualização de dados de terreno, cuja aplicação depende da forma adotada para a representação do terreno. Assim, métodos diferentes são usados para terrenos representados por mapas de alturas e terrenos representados por TINs.

2.4.1 Estratégias para terrenos representados por mapas de altura.

Uma forma de modelar um terreno é imaginá-lo como um sólido composto por elementos volumétricos elementares (*voxels*), os quais se erguem a partir de uma base horizontal. Nesta abordagem, um terreno pode ser visto como um conjunto de colunas formadas por uma pilha de *voxels*. Os métodos mais utilizados para a visualização de sólidos representados por *voxels* são o *Ray Tracing* e a composição de planos (*plane composition*) [FO90].

Existe uma razão forte para a não utilização explícita de *voxels* em modelos de terreno: o espaço para armazenamento. Um terreno armazenado através de *voxels* pode ter tamanho excessivamente grande, mesmo em baixa resolução espacial, além de que os *voxels* fornecem um grau de liberdade acima do necessário para a modelagem de um terreno.

Uma forma mais inteligente de armazenamento pode ser utilizada se atentarmos para o fato de que terrenos são estruturas muito mais bidimensionais do que tridimensionais (o que na literatura sobre modelagem geométrica se chama de dimensão $2\frac{1}{2}$). Como consideramos que cada ponto em um terreno assume um único valor de altura, podemos representá-lo através de uma matriz que armazena valores de altura obtidas de uma amostragem regular.

A visualização de terrenos armazenados como mapas de altura pode ser realizada através de utilização de técnicas como *Ray Casting* e variantes como o Algoritmo de Horizonte Flutuante (FS97) (Figura 2.3). Estes métodos são bastante eficientes, pois conseguem explorar a regularidade e a correlação existentes nos mapas de altura.

Os mapas de altura são estruturalmente semelhantes a representações discretas de imagens, tornando válida a utilização de várias técnicas já desenvolvidas e aplicadas em processamentos de imagens, como filtros, transformadas e modelos multiresolução.

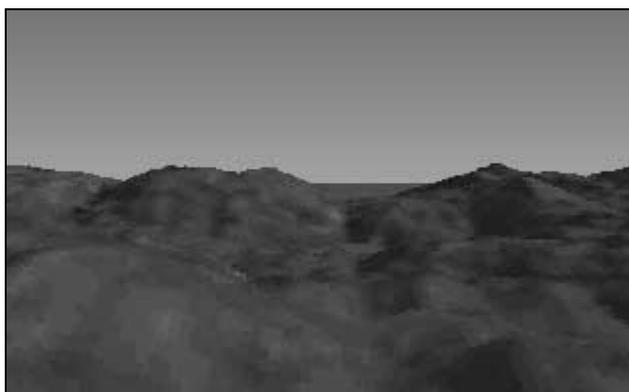


Figura 2.3 - Terreno visualizado pelo Algoritmo de Horizonte Flutuante

2.4.2 Estratégias baseadas em Triangular Irregular Networks(TINs)

As TINs são um conjunto contínuo de triângulos disjuntos cujos vértices são alocados adaptativamente sobre o domínio do DEM[CS95]. Ao contrário dos mapas de alturas, as TINs não só servem como uma forma de armazenamento para visualização, como também para toda a gama de operações já descritas na seção de aplicações.

A grande vantagem das TINs é sua adaptabilidade ao terreno, isto é, a capacidade de conter triângulos de tamanhos e formas diferentes que se “encaixam” na superfície, reduzindo desta forma o grau de redundância dos dados.

A reconstrução de uma superfície contínua através de dados discretos é obtida naturalmente através das malhas triangulares. A obtenção dos valores de altura em pontos não pertencentes às amostras originais pode ser feita através de uma interpolação realizada através dos triângulos da malha.

Os dados de terreno representados através de TINs podem ser visualizados através das técnicas convencionais de visualização de superfícies definidas por malhas poligonais(Figura 2.4). As TINs podem, desta forma, usufruir de todo avanço tecnológico obtido através de *rendering* por *hardware*, como *zbuffering* e mapeamento de texturas. Além da utilização dos métodos convencionais, é possível explorar a geometria e a topologia das malhas triangulares de forma a otimizar ainda mais o processo de *rendering*, tornando viável a visualização de áreas extremamente grandes.

Em relação a dados de terreno extensos, pesquisas têm sido feitas em busca de estruturas hierárquicas e de banco de dados que possam melhorar o grau de realismo nos processos de visualização e manipulação[GU97].

Um outro tópico bastante estudado relacionado a TINs são os métodos de simplificação de superfícies (*surface simplification methods*). O processo de simplificação consiste em reduzir a complexidade de uma malha (em geral, o número de triângulos que a compõe) mantendo dentro de uma certa tolerância, o grau de realismo e precisão da aproximação da mesma em relação aos dados de elevação originais. Este é o principal tema deste trabalho e será abordado em detalhes nos capítulos seguintes.



Figura 2.4 - Terreno visualizado através de uma TIN texturizada

3. Representação de terrenos por triangulações.

3.1 Introdução

Neste capítulo estudaremos a representação de dados de terreno através de triangulações. As triangulações utilizadas na modelagem de superfícies são comumente chamadas de TINs (*Triangular Irregular Networks*). Inicialmente faremos uma revisão dos principais conceitos e resultados sobre triangulações.

3.2 Triangulações

A estratégia das TINs é representar um terreno através de uma função por partes construída a partir de um conjunto de observações esparsas no plano.

Iniciamos com o problema definido em \mathbb{R} . Dado um conjunto $P = \{x_1, x_2, \dots, x_n\}$ de pontos sobre a reta e uma função $f(x_i)$, $i = 1..n$, desejamos estender esta função para uma função F , sobre um domínio D , tal que $F(x_i) = f(x_i)$, para $i = 1..n$. O domínio D consiste no intervalo da reta $I = [m, M]$, tal que $m = \min\{x_i | x_i \in P\}$ e $M = \max\{x_i | x_i \in P\}$. Para determinarmos F devemos estruturar o conjunto P através de uma ordenação de forma a delimitar os intervalos nos quais o valor de F é dado por uma função linear. Seja x um ponto tal que $x \in [m, M]$. O valor de $F(x)$ é obtido identificando-se a que subintervalo $[x_i, x_{i+1}]$ pertence o ponto x , exprimindo-se x em função de $\lambda(x_i) + (1-\lambda)(x_{i+1})$ e posteriormente avaliando-se a função $F(x) = \lambda F(x_i) + (1-\lambda)F(x_{i+1})$ (Figura 3.1).

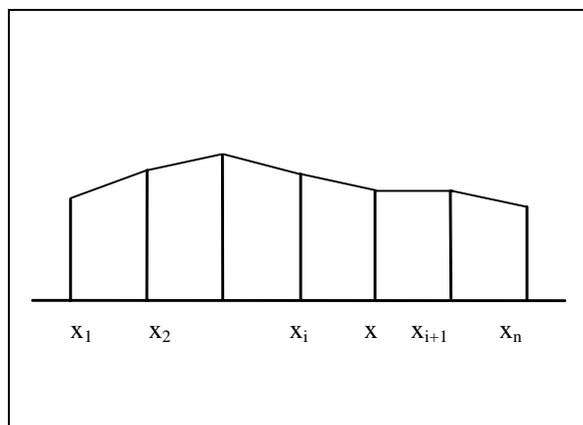


Figura 3.1-Interpolação em \mathbb{R}

O problema no R^2 é análogo ao problema em R . Dado um conjunto de pontos $P = \{p_1, p_2, \dots, p_n\}$ no plano, e uma função $f(p_i), i=1..n$, desejamos estender esta função para uma função F , definida sobre um domínio D , tal que $F(p_i) = f(p_i)$, para $i=1..n$. Neste caso o domínio D deve ser o conjunto união de todos os triângulos que formam o complexo simplicial de dimensão 2; isto é deve-se ter $D = \text{Conv}(P)$. Os intervalos que definem a função por partes são justamente triângulos disjuntos nos quais F é dada por uma função linear. Seja p_i um ponto pertencente a $\text{Conv}(P)$. O valor da função $F(p_i)$ pode ser obtido da seguinte forma: exprime-se p_i em função de λ_i onde $p_i = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3$, tal que $\lambda_1, \lambda_2, \lambda_3 > 0$ e avalia-se a função $F(x) = \lambda_1 F(p_1) + \lambda_2 F(p_2) + \lambda_3 F(p_3)$ (Figura 3.2 e Figura 3.3). As coordenadas $\lambda_1, \lambda_2, \lambda_3$ são as coordenadas baricêntricas do ponto p_i em relação ao triângulo $p_1 p_2 p_3$ que o contém.

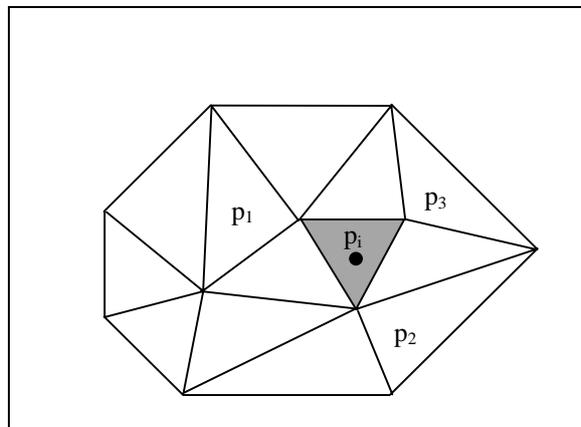


Figura 3.2 - Interpolação em R^2

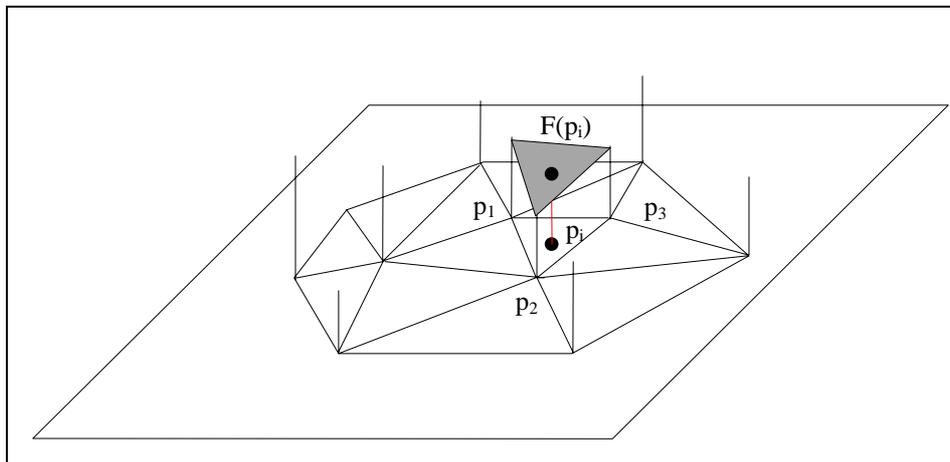


Figura 3.3 - Interpolação em R^2

Formalmente, o problema de triangulação consiste em: dado um conjunto de pontos P , encontrar um conjunto de triângulos $\{T_1, T_2, T_3, \dots, T_k\}$ satisfazendo a:

- i) a união dos triângulos é igual a $\text{conv}(P)$
- ii) o conjunto dos vértices dos triângulos é P .
- iii) a interseção de dois triângulos é \emptyset , um vértice ou um lado comum.

Não existe uma solução única para a triangulação de um conjunto de pontos $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^2$. Contudo, podemos determinar um conjunto de propriedades as quais todas as triangulações devem satisfazer. Estas propriedades são baseadas no fato de que triangulações definem uma subdivisão planar no plano em que estão inseridas. Em particular, dado o número de pontos n e o número de elementos v sobre o fecho convexo de P , o número de triângulos é $T=2n+v-2$ e o número de arestas é $a=3n-v-3$.

Este resultado é bastante importante pelo fato de que nos indica que o tamanho da solução do problema de triangulação é linear no número de elementos de P .

Embora o número de triângulos de uma triangulação seja sempre o mesmo, é desejável em muitas aplicações (em especial na modelagem de terrenos), que os triângulos tenham uma forma conveniente. Em dados de terreno é desejável que os triângulos não possuam ângulos muito pequenos, de forma que eles sejam bem parecidos com um triângulo equilátero. Uma das triangulações mais utilizadas para a modelagem de dados de terreno é a chamada triangulação de Delaunay[PS85,] que, como veremos no próximo capítulo, produz triângulos o mais próximo possível de triângulos equiláteros.

3.3 Representações de triangulações

A escolha da estrutura de dados utilizada para representar uma TIN deve levar em consideração os seguintes aspectos:

- armazenamento - quanto espaço em memória a estrutura de dados ocupa.
- custo de construção - tempo necessário para construir a estrutura.
- custo de manipulação - custo das operações de localização, inserção e remoção de um ponto na estrutura.

Em geral, é muito difícil encontrar uma estrutura que atenda a todos os critérios de forma satisfatória. Estruturas que necessitam de pouco espaço em memória não são poderosas o suficiente para a utilização de operadores mais

inteligentes; por outro lado, estruturas que permitem a utilização de operações mais inteligentes ocupam bastante espaço em memória.

A escolha da estrutura mais adequada deve procurar um balanceamento entre os três critérios descritos, procurando privilegiar aquele que é o mais importante. A seguir faremos uma análise baseada nos processos que envolvem TINs.

Armazenamento

As TINs utilizadas para representar dados de terreno são naturalmente extensas. A utilização de estruturas complexas requer uma grande quantidade de memória, que nem sempre se encontra disponível. Podemos resolver esta questão da seguinte forma: ou utilizamos uma estrutura mais simples e abrimos mão da qualidade dos operadores, ou utilizamos uma estrutura mais poderosa e aplicamos algum mecanismo de simplificação.

Custo de Construção

O custo de construção é um dos fatores menos relevantes, pois esta geralmente é feita como um pré-processamento. A utilização de métodos de simplificação também tem grande utilização na diminuição do custo de construção de uma TIN, principalmente se forem adotados métodos por refinamento já que eles não requerem uma triangulação inicial total.

Custo de manipulação

A manipulação de uma TIN talvez seja o aspecto mais importante a ser considerado para a escolha da estrutura de dados. As operações de manipulação mais utilizadas em TINs e em outros processos envolvendo malhas irregulares, são a localização, inserção e remoção de pontos. O grau de eficiência desejado para estas operações depende essencialmente da utilização final de TINs. Por exemplo: se a TIN for utilizada com o intuito de se obter uma visualização estática, não existe necessidade de operadores extremamente eficientes (na verdade, eles jamais serão utilizados). Porém se utilizarmos a TIN em processos de análise ou visualização interativa, certamente será desejável que estas operações sejam executadas o mais rápido possível.

Como um dos objetivos deste trabalho é propor métodos que facilitem operações mais robustas sobre TINs, indicaremos estruturas que possibilitem que estas sejam realizadas eficientemente, mesmo que a um custo maior de espaço de armazenamento.

Uma estrutura que representa uma TIN deve ser capaz de capturar as diversas propriedades que estas possuem. Mais do que um simples conjunto de triângulos definidos a partir de um conjunto de pontos 3D, as TINs representam uma região do espaço definida por partes. As partes encerram entre si um conjunto de relações de adjacências que podem ser exploradas de forma a permitir operações eficientes. Devemos atentar para o fato de que a projeção de uma TIN em um plano define uma subdivisão planar[PS85], a qual por sua vez está diretamente associada a um grafo planar. A estrutura de um grafo planar associado a uma subdivisão do plano define um conjunto de propriedades topológicas. As estruturas de dados capazes de descrever estes diagramas são as chamadas **estruturas de dados topológicas**.

Uma estrutura de dados topológica eficiente deve ser capaz de produzir todas as relações de adjacência entre vértices, arestas e faces em tempo ótimo.

Diversas estruturas de dados topológicas foram propostas na literatura afim. As mais conhecidas são a clássica Winged-Edge[BA75] e a Half-Edge[MA88]. Neste trabalho utilizamos uma estrutura desenvolvida por Guibas e Stolfi[GS85]: a *quad-edge data structure*. A descrição da quad-edge data structure pode ser encontrada no apêndice deste trabalho.

4. Triangulações de Delaunay

4.1 Introdução

Introduziremos neste capítulo os principais conceitos sobre Triangulações de Delaunay. Descreveremos em detalhes o algoritmo de triangulação incremental de Delaunay que será de grande importância para o desenvolvimento dos capítulos seguintes.

4.2 Definição

O mais usado método de triangulação de um conjunto de pontos do plano é a que produz a chamada **Triangulação de Delaunay**[GS85, FC91 e PS85].

A triangulação de Delaunay é caracterizada pela seguinte propriedade: uma aresta **a** pertence a uma triangulação de Delaunay se e somente se existe um círculo que passa pelos vértices de **a** e deixa os demais pontos de fora (Figura 4.1).

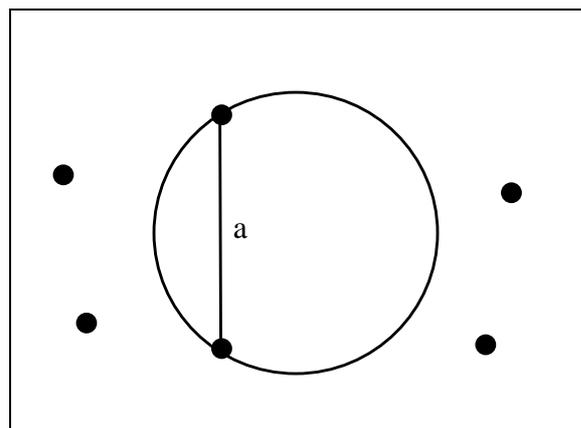


Figura 4.1 - Propriedade geométrica da triangulação de Delaunay

Uma outra propriedade fundamental é que a triangulação de Delaunay maximiza o ângulo mínimo de todos os triângulos dentre todas as possíveis triangulações de um conjunto de pontos. Isto faz com que a triangulação não apresente triângulos muito finos.

A triangulação de Delaunay pode ser computada usando algoritmos de varredura, divisão e conquista ou incrementalmente através da inserção de um ponto por vez[FC91] e [PS85].

O algoritmo de Triangulação de Delaunay Incremental é uma técnica muito interessante utilizada para construir uma TIN a partir de um conjunto de amostras. Apesar de não possuir complexidade ótima (no pior caso sua complexidade é $O(n^2)$), este algoritmo apresenta uma propriedade que torna justificável sua utilização. Ele é capaz de construir triangulações *on-line*, isto é, permite a construção de uma triangulação mesmo que não se conheça todos os pontos antes do início do processo. Existem outros métodos de triangulação que possuem complexidade ótima($O(n \log n)$), porém estes só são aplicáveis quando conhece-se previamente todo o conjunto de pontos que se deseja triangular. É interessante também ressaltar que o algoritmo incremental pode ser implementado de modo a ter complexidade média $O(n \log n)$ [GB97].

A capacidade de construir triangulações incrementais permite que este método seja utilizado para obter triangulações com diversos níveis de refinamento, o que será de extrema importância para os métodos que serão descritas posteriormente neste trabalho.

4.3 Descrição do algoritmo

O primeiro passo do algoritmo de triangulação incremental de Delaunay requer que seja criada uma triangulação inicial mínima. Se os dados que definem a superfície foram obtidos através de uma amostragem sobre uma grade regular, então a triangulação inicial pode ser obtida de forma trivial. Inicialmente são obtidas as coordenadas do retângulo envolvente. A partir dessas coordenadas, cria-se a triangulação mínima conectando-se o canto superior esquerdo ao canto inferior direito do retângulo (Figura 4.2).

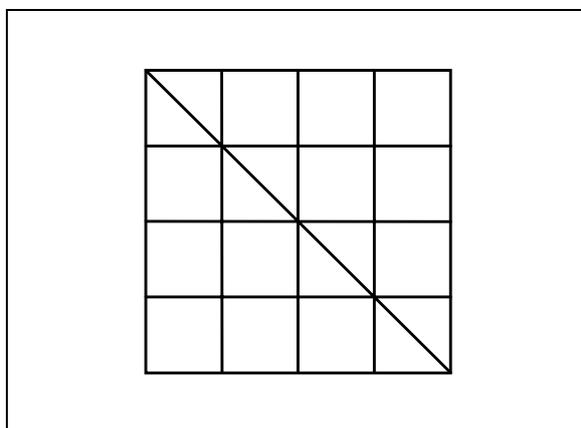


Figura 4.2 - Triangulação inicial em amostragem regular

Se os dados foram amostrados irregularmente, então uma outra estratégia deve ser adotada. Deve-se criar uma triangulação inicial que contenha todos os pontos do modelo. A menor triangulação capaz de satisfazer esta propriedade, é a que tem como entrada o conjunto de pontos que compõem a envoltória convexa dos pontos do modelo. A envoltória dos dados pode ser obtida facilmente em tempo ótimo através de algoritmos como o Mergehull e Quickhull[FC91] e [PS85]. Tendo sido obtida a envoltória convexa do modelo, aplica-se um algoritmo de triangulação de Delaunay [GS85],[FC91] e [PS85] (Figura 4.3) .

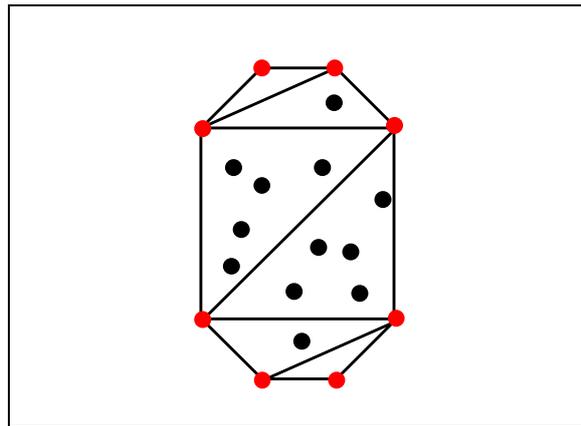


Figura 4.3 - Triangulação inicial em amostragem irregular

Uma vez obtida a triangulação mínima, começamos a fase de inserção dos pontos. O primeiro passo da fase de inserção de pontos consiste na seleção de um ponto a ser inserido. O critério de seleção define a ordem em que os pontos serão inseridos na triangulação. Por enquanto, estamos interessados em inserir todos os pontos, logo, a seleção pode ser feita de forma arbitrária.

Tendo sido escolhido o ponto a ser inserido na triangulação, devemos localizar o triângulo que o contenha(Figura 4.4). O processo de localização depende da estrutura utilizada para armazenar a TIN. Um algoritmo para localização de um ponto em uma triangulação armazenada através de uma estrutura topológica pode ser encontrado em [GS85]. Este algoritmo é descrito através da estrutura *quad-edge* mas pode ser modificado para outras estruturas topológicas.

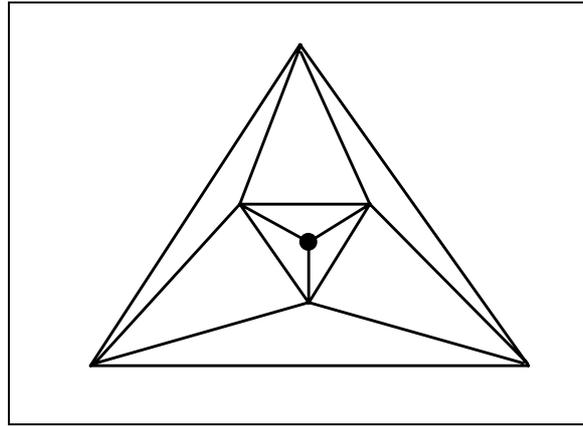


Figura 4.4- Localização do ponto

Uma vez encontrado o triângulo deve-se verificar se o ponto é interno ou se se encontra sobre uma de suas arestas. Se o ponto encontra-se sobre uma aresta devemos removê-la (Figura 4.5).

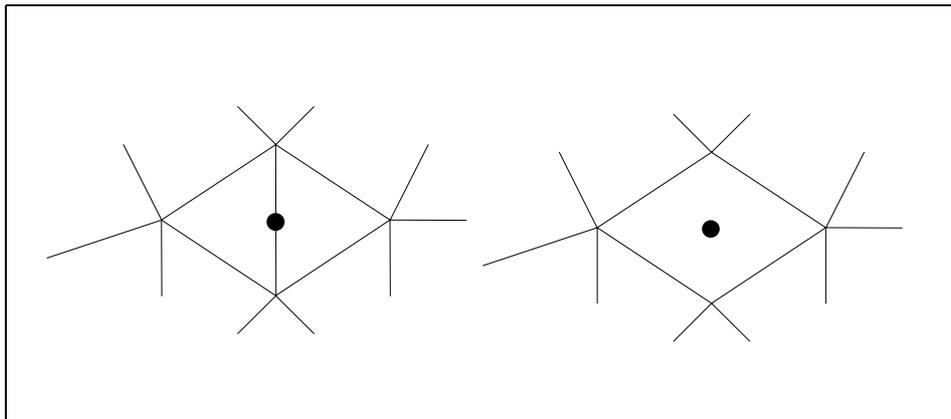


Figura 4.5 - Ponto sobre uma aresta

Em qualquer uma das situações o resultado é um triângulo ou quadrilátero que envolve o ponto inserido. Devemos então criar arestas conectando o ponto inserido aos vértices do polígono(Figura 4.6).

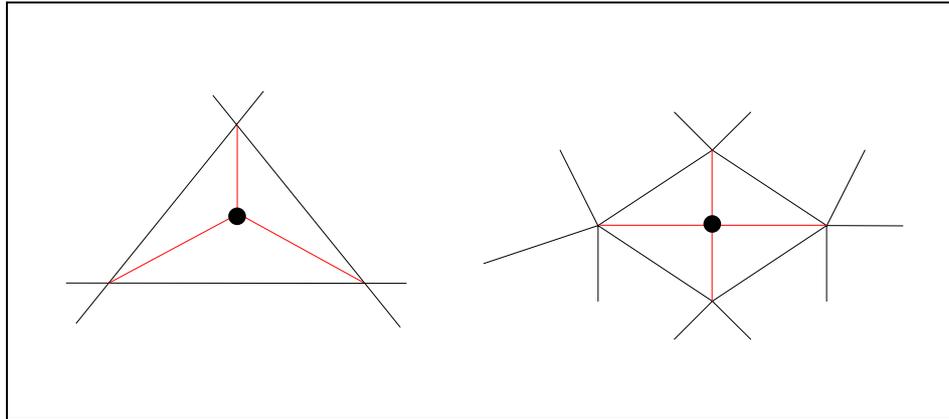


Figura 4.6 - Criação de novas arestas nos casos 1 e 2

As novas arestas criadas satisfazem comprovadamente o critério de Delaunay. Isto pode ser demonstrado da seguinte forma: considere o círculo C que circunscreve o triângulo de Delaunay que contém o novo ponto inserido. Para cada vértice Y do triângulo podemos passar um círculo C' que passa pelo novo ponto X e é tangente a C e Y . Este círculo não contém nenhum ponto, fazendo com que a aresta satisfaça o critério de Delaunay (Figura 4.7).

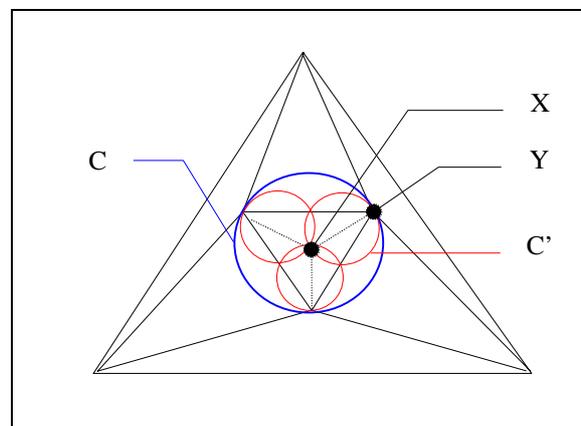


Figura 4.7 - As novas arestas satisfazem o critério de Delaunay

Resta então saber se as arestas do polígono também satisfazem o critério de Delaunay. Para isto é necessário realizar o teste de Delaunay para cada uma de suas arestas.

O teste de Delaunay é aplicado da seguinte forma:

Seja $ABCX$ o quadrilátero formado pela união dos triângulos adjacentes a aresta e a ser testada. Verificamos se o círculo passando pelo triângulo ABC contém X (Figura 4.8).

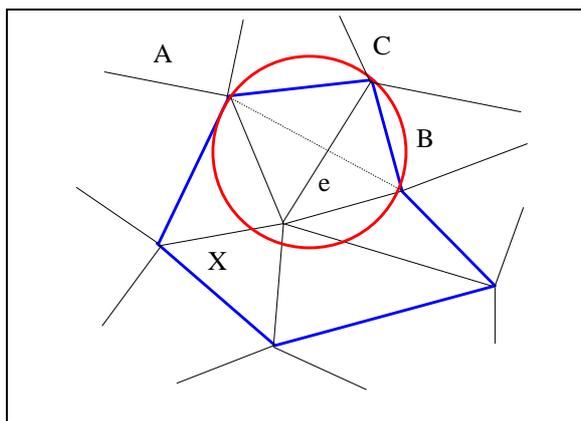


Figura 4.8 - Aplicação do teste de Delaunay

Se obtivermos resposta negativa, então a aresta já é de Delaunay e devemos testar a próxima no sentido anti-horário com mesma origem que e . Se a resposta for positiva devemos trocar a aresta de forma que ela conecte os vértices que determinam a outra diagonal do quadrilátero. Nesta segunda possibilidade o polígono aumenta de tamanho ao se incluir novas arestas que também deverão ser testadas.

Quando todas as arestas da nova triangulação satisfizerem o critério de Delaunay teremos obtido uma nova triangulação na qual foi inserido mais um ponto. O processo de inserção continua até que todos os pontos do conjunto de amostras tenham sido inseridos.

Na Figura 4.9 exemplificamos o processo de inserção de pontos em uma triangulação de Delaunay. As duas últimas figuras representam a inserção de um novo ponto sobre uma aresta da triangulação e seu respectivo efeito (requer a remoção da aresta existente antes da criação das novas arestas conectando o ponto inserido aos vértices do quadrilátero envolvente).

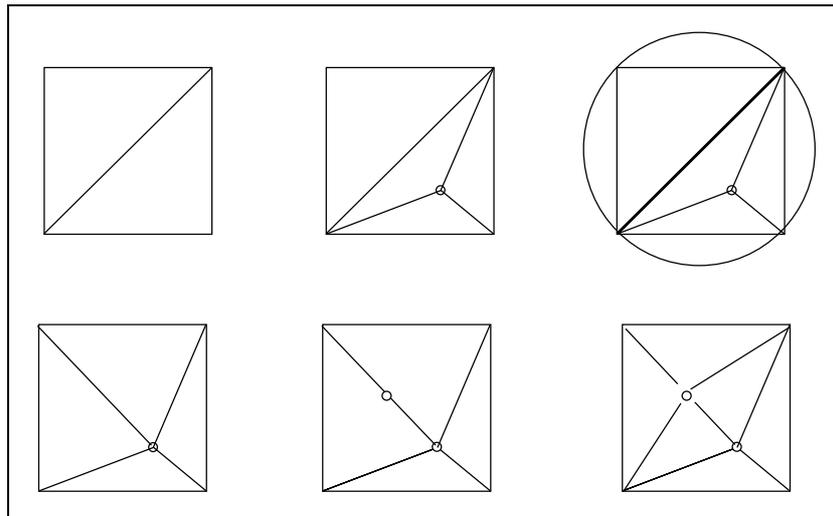


Figura 4.9 - Exemplo de aplicação do teste de Delaunay

Abaixo apresentamos o algoritmo descrito acima em forma de pseudocódigo. O algoritmo retorna em T a triangulação de Delaunay do conjunto P de pontos do plano.

Algoritmo_Incremental_de_Delaunay(P,T)

```

{
   $T = \{\}$ ;

  Construir uma triangulação mínima  $T$  definida pela envoltória
  convexa de  $P$ ;

  Repita
    Selecionar um ponto  $p$  do conjunto de amostras  $P$ ;
     $P = P - \{p\}$ ;
    Localizar o triângulo  $t \in T$  tal que  $t$  contenha  $p$ ;
    Se  $p$  estiver sobre uma aresta  $e$  de  $t$  então remova  $e$  de  $T$ .
    Criar arestas conectando  $p$  aos vértices do polígono que o
    envolve;
    Atualizar a triangulação de forma que satisfaça o critério de
    Delaunay;
  enquanto  $P \neq \{\}$ ;
}

```

5. Métodos de Simplificação de Superfícies.

5.1 Introdução

A grande quantidade de triângulos gerada por triangulações envolvendo todas as amostras de um terreno torna inviável os processos de visualização e manipulação geométrica. Por este motivo, têm sido estudados métodos que procuram reduzir o número de triângulos de uma TIN, mantendo uma aproximação dentro de uma tolerância desejada. Estes métodos são denominados métodos de simplificação de superfícies e serão abordados nos capítulos seguintes deste texto.

5.2 Origem dos métodos de simplificação de superfícies

Os métodos de simplificação de superfícies têm origem nos métodos de simplificação de curvas. Os métodos de simplificação de curvas têm sido extensamente explorados nas literaturas sobre cartografia, sistemas de informação geográfica, geometria computacional, computação gráfica e matemática aplicada. O problema básico de simplificação de curvas consiste em receber como entrada uma curva poligonalizada com n vértices e produzir como saída uma curva poligonalizada com m vértices, sendo $m \leq n$.

5.3 Simplificação de Curvas

O problema de simplificação de curvas é um problema de otimização, pois queremos encontrar um subconjunto mínimo de pontos dentro de um conjunto maior, de forma que o erro de aproximação esteja dentro de um limite pré-estabelecido. Este problema é **NP-Árduo** e é tratado geralmente através de métodos heurísticos que proporcionem uma boa aproximação da solução ótima.

Um algoritmo de alta qualidade bastante usado em simplificação de curvas é o algoritmo chamado Douglas-Peucker[UR72]. A cada passo, o algoritmo tenta aproximar uma sequência de pontos por um segmento de reta ligando o primeiro ao último ponto. Uma vez encontrado o ponto mais distante entre a curva e o segmento aproximador, verifica-se se sua distância está dentro de um limite estabelecido (Figura 5.1). Se a resposta for positiva a aproximação é aceita, caso contrário o algoritmo é aplicado recursivamente às subsequências anterior e posterior ao ponto mais distante (Figura 5.2).

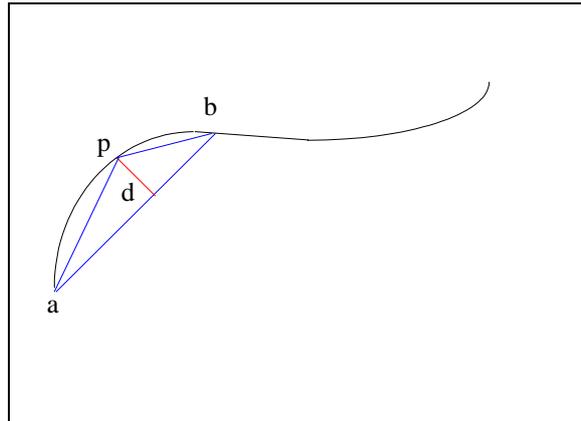


Figura 5.1 - Douglas Peucker (nível 1)

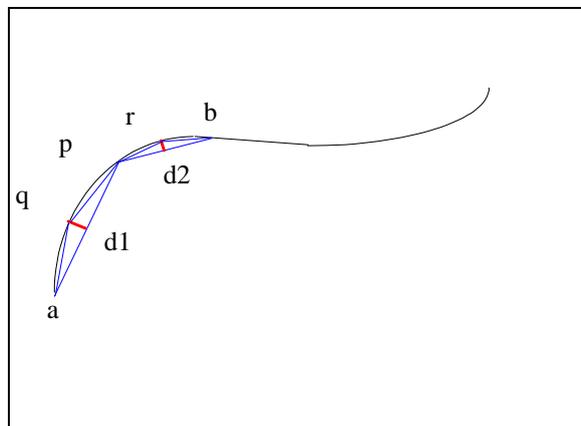


Figura 5.2 - Douglas Peucker (nível 2)

5.4 Simplificação de superfícies.

Os algoritmos de simplificação de superfícies são extensões para três dimensões das idéias básicas dos algoritmos de simplificação de curvas. Dada uma superfície poligonalizada com n vértices, deseja-se obter uma superfície poligonalizada com m vértices tal que $m \leq n$.

Diferentemente dos métodos de simplificação de curvas, os métodos de simplificação de superfícies não foram totalmente explorados. Além disso, não é tão simples utilizar métodos de divisão e conquista como se faz na heurística Douglas-Peucker, visto que a divisão de uma superfície por um ponto não pode ser feita de forma única.

Alguns métodos só funcionam para campos de alturas (*height fields*), enquanto que outros podem ser estendidos a uma superfície qualquer. Em nosso texto, estamos interessados no caso em que uma entrada é um campo de alturas e a saída é uma TIN (*Triangular Irregular Network*).

Os algoritmos de simplificação poligonal de superfícies podem ser divididos em 6 grupos:

- métodos por grade uniforme.
- métodos por subdivisão hierárquica.
- métodos por *features* em único passo.
- métodos por refinamento e híbridos.
- métodos de decimação em múltiplos passos e
- outros.

Métodos em grade uniforme

São métodos que utilizam as técnicas mais simples. Basicamente consiste em fazer uma subamostragem da superfície em uma grade regular, no qual cada k-ésima linha e coluna são guardadas formando uma grade e todos os outros pontos são descartados. A subamostragem uniforme é simples e rápida, contudo é de baixa qualidade, já que pontos importantes podem ser descartados. Os resultados deste método podem ser melhorados se usarmos um filtro passa baixa antes da subamostragem[RH83]. Contudo, isto não resolve seu problema mais crítico que é a falta de adaptabilidade. O processo de subamostragem pode ser útil em modelos multiresolução, o qual é construído hierarquicamente através de uma pirâmide de amostras[PH91],[LW83] e [TP75].

Métodos de subdivisão hierárquica

Os métodos de subdivisão hierárquica constroem a triangulação por subdivisão recursiva da superfície. A subdivisão hierárquica forma uma árvore onde cada nó possui apenas um pai. Estes métodos são simples, rápidos e facilitam a modelagem em multiresolução. Possibilitam também *rendering* em níveis de detalhe adaptativo, isto é, as partes mais próximas de uma cena são visualizadas em nível de detalhe maior que as partes mais distantes. Apesar de todas estas vantagens, em geral os métodos por subdivisão hierárquica fornecem resultados de baixa qualidade, tanto do ponto de vista perceptual como numérico[HG95].

Métodos por *features*

São métodos bastante intuitivos, no qual faz-se uma passagem sobre os pontos de entrada classificando-os segundo uma medida de importância. Os pontos mais importantes são selecionados para fazer parte do conjunto de vértices sobre o qual será feita a triangulação (usa-se tipicamente a triangulação de Delaunay). Os métodos por pontos característicos são muito populares em cartografia. Os pontos críticos geralmente incluem elementos topográficos tais como picos, depressões, cordilheiras e

vales. A filosofia por trás destes métodos é a de que o conhecimento sobre as características do terreno é essencial para uma boa aproximação. Os pontos críticos são incluídos no conjunto de vértices e as arestas que os conectam (linhas de descontinuidade) formam as restrições da triangulação. Em geral usa-se filtros lineares 2×2 e 3×3 para detecção dos pontos característicos seguidos de algum processo para descartar pontos demasiadamente próximos como por exemplo, pontos que se dispõem ao longo de uma cordilheira.

Métodos por refinamento e híbridos

Os métodos por refinamento são algoritmos de passo múltiplo que iniciam com uma aproximação inicial mínima e a cada iteração inserem um ou mais pontos como vértices da triangulação. O algoritmo termina quando um determinado erro é alcançado ou quando foi inserido o número desejado de vértices. Para dados em uma grade retangular pode-se tomar como aproximação inicial a triangulação do retângulo envolvente. Para outras topologias, a aproximação pode ser mais complexa. O refinamento da triangulação pode ser feito através de uma triangulação de Delaunay incremental, onde a escolha dos pontos é feita com base em medidas de importância dos pontos. Enquanto que em métodos por pontos críticos utiliza-se critérios de importância independentes da aproximação, em geral, em métodos por refinamento, a importância de um ponto é dada pela medida do erro entre os dados originais e a aproximação. Os métodos por refinamento possibilitam a combinação com outras técnicas, como por exemplo, a utilização de estruturas hierárquicas, gerando assim métodos híbridos. No capítulo seguinte abordaremos em maiores detalhes um dos métodos de simplificação por refinamento que tem fornecido resultados bastante satisfatórios: **o método de simplificação por inserção gulosa**. Este método será a base de todos os desenvolvimentos efetuados neste trabalho.

Métodos por decréscimo

Nestes métodos parte-se de uma triangulação inicial completa (contendo todos os pontos) que por sua vez é simplificada sucessivamente até que um certo nível de aproximação seja alcançado. Pode-se citar como exemplo deste método a “drop heuristic” de Lee [JL89].

Um método de dizimação bastante elegante foi proposto no trabalho de [SR94] para a simplificação de terrenos e sólidos geométricos em geral. O algoritmo descrito procura remover a cada passo o ponto que menos contribui para os detalhes de aproximação da superfície. O critério de seleção do ponto menos importante é baseado no grau de rugosidade em relação aos seu triângulos vizinhos. Uma superfície tangente ST é colocada através de um vértice **p** que é examinado (Figura 5.3).

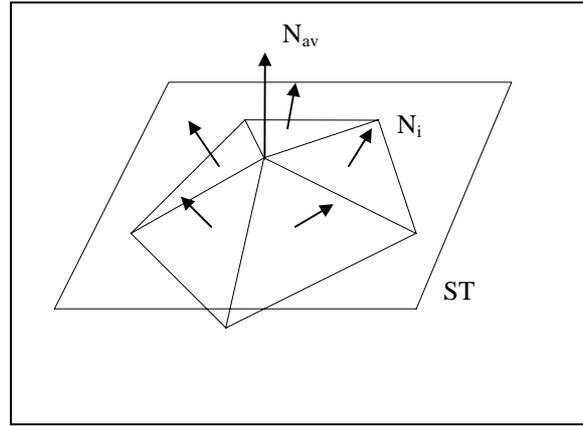


Figura 5.3 - Critério de rugosidade

A orientação n_{av} de ST é dada pela média das normais n_i dos triângulos que envolvem p ponderados pela sua área A_i :

$$n_{av} = \frac{\sum_{i=1}^{tno} \bar{n}_i \cdot A_i}{\sum_{i=1}^{tno} A_i}$$

n_{av} : normal média que determina a orientação da superfície tangente ST em p .

tno : número de triângulos que circundam o vértice examinado p .

\bar{n}_i : superfície normal do triângulo i .

A_i : área do triângulo i .

Calcula-se então o ângulo máximo a_{max} entre a normal média n_{av} e as normais das superfícies definidas pelos triângulos que circundam o ponto p .

$$a_{max} = \max_{i=1}^{tno} \left(\arccos \frac{\bar{n}_{av} \cdot \bar{n}_i}{|\bar{n}_{av}| \cdot |\bar{n}_i|} \right)$$

O resultado em a_{max} representa o maior ângulo entre n_{av} e n_i e assume valores entre 0 e π . Se n_{av} e todas as outras normais forem normalizadas, podemos então usar a seguinte expressão:

$$a_{max} = \max_{i=1}^{tno} \left(\arccos(\bar{n}_{av} \cdot \bar{n}_i) \right)$$

Pelo critério de qualidade, o algoritmo procurará remover todos os pontos cujo valor a_{\max} é menor que o valor limite estabelecido **AngMax**. Se o objetivo é remover uma certa quantidade de pontos, então deve-se ordenar previamente todos os pontos por a_{\max} em ordem crescente de forma a manter os mais significativos.

O autor deste trabalho propõe três técnicas para retriangular o conjunto de pontos resultante do processo de decimação:

Triangulação de Delaunay: retriangular-se segundo o critério de Delaunay todos os pontos que sobrevivem ao processo.

Small Hole approach: a cada ponto removido, utiliza-se um algoritmo de triangulação simples para retriangular os “buracos” gerados.

Big Hole approach: permite-se que os pontos sejam removidos e ao final do processo, retriangular-se os buracos restantes.

Um estudo comparativo das três estratégias pode ser encontrado neste mesmo trabalho.

Por considerar apenas a TIN inicial, este algoritmo pode ter um comportamento de pior caso quando a superfície aproximada é bastante suave, como por exemplo, uma gaussiana. Neste caso todo o modelo é removido, já que a inclinação dos triângulos muda muito pouco em relação ao seus vizinhos (não há rugosidade). O autor soluciona este problema propondo que o valor a_{\max} seja computado tanto para a TIN original como para o estágio corrente da TIN decimada. Somente quando os dois valores a_{\max} são menores que o valor limite estabelecido, o ponto é removido. Desta forma, pontos que sejam importantes nos estágios iniciais de decimação não são removidos. Caso deseja-se adotar um critério de quantidade, torna-se necessário a atualização da lista dinâmica ordenada dos a_{\max} (ver[SR94]).

Este método fornece bons resultados, mas não permite uma fácil avaliação da precisão da aproximação. Uma outra desvantagem consiste no fato de que a estratégia utilizada só pode ser aplicada a métodos de decimação e não aos métodos por refinamento.

6. O Método de Refinamento por Inserção Gulosa

6.1 Introdução

O método de simplificação por inserção gulosa é um dos métodos mais difundidos na literatura afim. Este método foi desenvolvido por diversos autores simultaneamente, havendo versões ligeiramente diferentes. Dentre estas várias abordagens, podemos destacar os trabalhos de W. Franklin e Paul Heckbert[HG95]. Neste capítulo descreveremos o método de simplificação gulosa, discutindo em detalhes seus dois principais componentes: o critério de seleção e o critério de parada. Descreveremos também as versões paralelas dos algoritmos de inserção gulosa.

6.2 Descrição do método

Apesar de ser baseado em uma idéia bastante simples, o método de simplificação por inserção gulosa fornece resultados de boa qualidade. A grande virtude deste método consiste em sua simplicidade e flexibilidade, visto que é muito simples acrescentar novas heurísticas e conjuntos de regras ao algoritmo básico.

O método de simplificação por inserção gulosa é um método de simplificação construído com base no algoritmo de triangulação incremental de Delaunay, que constrói uma triangulação inserindo um ponto por vez. Se quisermos obter uma triangulação completa, a ordem na qual os pontos são inseridos é completamente irrelevante, podendo ser escolhida arbitrariamente. Porém, podemos não querer inserir todos os pontos, e sim obter uma versão simplificada do modelo(exatamente o que os métodos de simplificação propõem). Para isto é necessário escolher um mecanismo que guie o método na escolha dos pontos mais relevantes para a modelagem da superfície. Este mecanismo é denominado **critério de seleção de pontos**. Também é necessária a escolha de um critério capaz de indicar o momento em que o processo de inserção deve ser interrompido. O critério de seleção de pontos e o critério de parada serão vistos em detalhes nas duas seções seguintes.

6.3 Critério de Seleção de Pontos.

O critério de seleção de pontos é o mecanismo que fornece a inteligência necessária para que o algoritmo possa realizar a seleção dos pontos mais adequados. Este critério é que determina quais e em que ordem devem ser inseridos os pontos do conjunto de amostras na triangulação. Em geral, o critério de seleção de pontos utiliza uma **medida de importância** de um ponto. A medida de importância $M(\mathbf{p})$ de um ponto \mathbf{p} pode ser expressa matematicamente da seguinte forma:

$$M(\mathbf{p})=f_{T,P}(\mathbf{p}),$$

onde T é a triangulação que fornece a topologia da superfície e P é o conjunto de pontos associado às amostras do campo de alturas.

Logo, o critério de seleção dos métodos de inserção gulosa fica definido por

selecionar p_i tal que $M(p_i)$ é máximo

A medida de importância de um ponto pode ser feita de diversas formas. Descrevemos abaixo algumas medidas citadas por Heckbert[HG95].

Medida de Curvatura

Esta medida associa um valor de importância a cada ponto baseado no grau de curvatura da região de sua vizinhança. Pontos visualmente significativos como vales, picos, depressões, etc. apresentam graus de curvatura elevados; logo, tendem a ser escolhidos para inserção se esse tipo de medida é usado.

A curvatura de uma superfície pode ser avaliada através da derivada segunda da função H que define uma superfície $z=H(x,y)$ (por exemplo através da norma de Frobenius da matriz Hessiana)[SP79]. Estas medidas são, no entanto, excessivamente sensíveis a regiões de alta frequência. Por exemplo, regiões em forma de dente de serra muito pequenas em escala apareceriam como áreas de altíssima importância, fazendo com que o algoritmo selecionasse muitos de seus pontos, ao invés de capturar pontos de regiões mais significativas.

Este problema pode ser solucionado através da aplicação de filtros de suavização sobre os dados originais. Ao avaliar esta medida na simplificação de curvas, Heckbert não alcançou resultados tão bons quanto os da medida de erro vertical local (abordado a seguir). Tudo indica que esta medida também não seja eficaz quando aplicada isoladamente à simplificação de superfícies, que por sua vez constitui um problema bem mais complexo. No capítulo seguinte proporemos uma

estratégia que utiliza uma combinação de uma medida baseada na curvatura e de erro vertical local.

Medida de erro vertical local

A medida de erro vertical local consiste simplesmente em medir o erro dado por $M(p)=|H(p)-TS(p)|$, onde $H(p)$ é o valor da altura do ponto p com coordenadas (x,y) e $TS(p)$ é o valor obtido através da interpolação, baseada na projeção de p no plano horizontal, dos valores que a função assume nos vértices do triângulo da aproximação corrente que contém p (Figura 6.1).

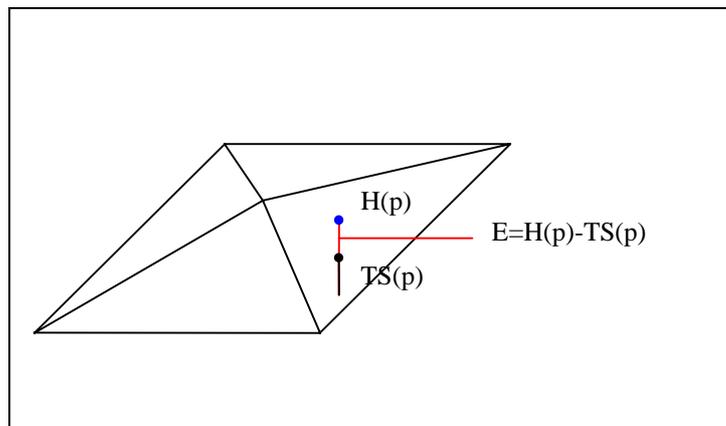


Figura 6.1 - Medida de erro vertical local

Em relação ao método por inserção gulosa, esta medida é dinâmica no sentido de que os erros dos pontos se modificam à medida que novos pontos são inseridos. Não há nenhuma garantia de que o erro vertical máximo diminua a cada nova inserção (Figura 6.2). Contudo, este converge a zero à medida que o processo progride.

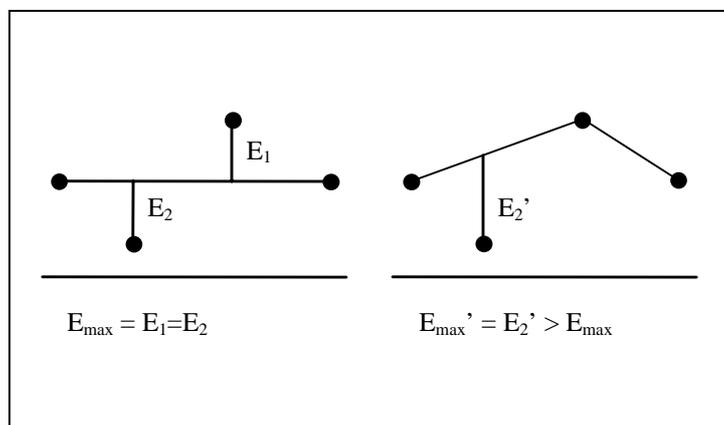


Figura 6.2 - Convergência não monotônica do erro vertical local

A convergência não monotônica não deve ser vista como uma desvantagem. Na verdade, é ela que permite que movimentos ruins a curto prazo possam ser feitos para que a longo prazo o algoritmo possa chegar a um resultado melhor. Em outras palavras, o fato de convergir não monotonicamente permite que o algoritmo consiga fugir de alguns ótimos locais, podendo chegar a soluções melhores.

Medida de erro vertical global

A medida de erro vertical global avalia para cada ponto p , o erro global da nova aproximação obtida através da possível inserção de p na aproximação corrente. A medida de erro vertical global é expressa da seguinte forma:

$$MG(p_i) = -(E(T \cup p_i) - E(T)), \text{ onde } E(T) = \sum_{p_i \in T} |H(p_i) - TS(p_i)|$$

Para cada ponto p_i do conjunto de amostras não inserido na aproximação corrente T , avaliamos a medida de importância $MG(p_i)$. Esta medida indica qual foi o decréscimo obtido pela inserção de p_i em T . Quanto maior o decréscimo, maior erro devido a p_i e, por conseguinte, maior sua importância.

Apesar de aparentemente fornecer uma medida de importância mais inteligente, a medida por erro global não fornece resultados melhores que a medida de importância baseada em erro local [HG95]. Este problema surge devido à natureza excessivamente conservadora da medida, isto é, não são permitidos movimentos que a curto prazo ocasionam um aumento no erro global para a longo prazo proporcionar uma redução maior. A convergência do algoritmo de simplificação por inserção gulosa que utiliza esta medida de importância é monotônica decrescente.

6.4 Critérios de parada

O estabelecimento de um critério de parada adequado é outro fator bastante importante nos métodos de simplificação em geral. São os critérios de parada que especificam o grau de refinamento de uma TIN gerada através de um método de simplificação de superfícies.

O critério de parada de um algoritmo de simplificação de superfícies pode ser baseado em duas diferentes abordagens básicas :

- número máximo de triângulos gerados
- erro máximo tolerado

A primeira abordagem especifica que o algoritmo deve inserir pontos enquanto o número de triângulos criados for menor que o valor limite. Este critério é útil quando precisamos limitar o tamanho da estrutura: por exemplo, quando temos uma quantidade restrita de memória ou *hardware* de pouca eficiência. O problema com este critério é a dificuldade do usuário de especificar o grau de precisão desejado da aproximação.

O segundo critério de parada é baseado no erro entre a TIN obtida através de um processo de simplificação e a superfície original descrita por uma função discreta. Este critério determina que devemos inserir pontos até que o erro entre as duas superfícies seja menor que a tolerância pré-estabelecida.

Podemos utilizar diversas métricas para medir o erro da aproximação. Uma possibilidade consiste em medir o erro global entre as duas superfícies. O problema em se utilizar o erro global advém do fato de que ela permite a existência de imprecisões locais indesejáveis. Isto é, mesmo que o erro global seja atendido considerando-se uma certa tolerância, podem existir localidades em que o erro seja maior que a média dos erros.

Uma alternativa melhor é a utilização do **erro máximo local** como medida de erro. No pior dos casos, existe a garantia de que nenhum dos pontos possuirá um erro maior do que o limite estipulado.

A maior dificuldade na utilização de critérios de parada por erro situa-se na escolha da **tolerância**(isto é, no valor máximo de erro permitido). A tolerância é um valor que deve ser adequado às características da superfície.

A utilização de medidas de distância arbitrárias é inadequada. Para uma determinada superfície, um valor fornece bons resultados, enquanto, para outra, os resultados podem ser totalmente inaceitáveis.

Um forma mais adequada consiste em utilizar como tolerância um valor percentual de alguma medida característica da superfície. Esta medida característica pode ser a variação máxima dos valores, o valor médio, o desvio padrão, ou outra medida que represente a distribuição dos valores assumidos pela função. O objetivo a ser alcançado é o de especificar parâmetros de tolerância que sejam independentes da superfície.

6.5 Versão seqüencial versus versão paralela.

Na seção anterior descrevemos versões **sequenciais** do algoritmo de inserção gulosa, isto é, apenas um ponto é inserido a cada iteração. Existem versões deste algoritmo nas quais mais de um ponto é inserido por vez. São as chamadas versões **paralelas**. Alguns trabalhos foram feitos com o intuito de comparar as versões paralela e sequencial. Puppo et al.[PU65] defenderam a hipótese, através de dados experimentais, de que as versões paralelas fornecem resultados melhores que as versões sequenciais. Sua hipótese é baseada no fato de que, no momento em que o teste de Delaunay é aplicado, a triangulação corrente encontra-se mais refinada que no método sequencial, já que mais pontos foram inseridos.

Existem duas versões básicas de algoritmos de inserção gulosa paralela: algoritmo com limite constante(*constant threshold*) e algoritmo com limite fracionário(*fractional threshold*).

No algoritmo com limite constante são selecionados todos os pontos candidatos p_i tal que $\text{Erro}(p_i) > e$, sendo e o erro máximo especificado pelo usuário.

No algoritmo de limite fracionário são selecionados todos os pontos p_i tais que $\text{Erro}(p_i) > \alpha e_{\max}$, onde $0 \leq \alpha \leq 1$ e e_{\max} o erro máximo de todos os candidatos no estágio corrente da triangulação. Isto é, são selecionados todos os pontos que tem erro maior que uma fração do erro do pior candidato. Quando $\alpha=0$, todos os candidatos de cada triângulo são selecionados, tornando o algoritmo excessivamente “guloso”. Quando $\alpha=1$ o algoritmo tende para o algoritmo sequencial, diferenciando-se apenas pelo fato de que pode inserir dois ou mais pontos, desde que tenham erro iguais a e_{\max} .

Resultados comparativos entre o método sequencial e as versões paralelas podem ser vistos em [HG95]. Neste trabalho, Heckbert demonstra que as versões paralelas tendem a causar explosões de densidade em regiões onde o erro é menor que o erro máximo especificado pelo usuário, criando um efeito de bola de neve. Candidatos de triângulos pequenos são selecionados, levando a subdivisões que os tornam menores ainda. Isto ocorre porque os algoritmos paralelos, por inserirem mais de um ponto por vez, não são capazes de se adaptar às mudanças geométricas ocorridas na superfície cada vez que um ponto é inserido. No algoritmo sequencial, a inserção do ponto associado ao pior candidato pode fazer com que o segundo pior ponto tenha seu erro minimizado de tal forma que ele não seja mais necessário. O algoritmo paralelo, por sua vez, funciona a pacotes, isto é, seleciona para inserção

um grupo de pontos que satisfaz um determinado critério de erro. Não importa se a inserção do primeiro ponto já tenha sido capaz de ajustar a superfície aos demais do grupo: eles serão inseridos assim mesmo. Esta característica faz com que o algoritmo paralelo insira uma grande quantidade de pontos que não são significativos para a melhora da precisão da aproximação.

Os testes efetuados por Heckbert confirmaram esta suspeita. Gráficos comparativos dos métodos mostram que a função associada ao erro converge para zero muito mais rápido nos algoritmos sequenciais que nos algoritmos paralelos.

7. Comparação de critérios de seleção no método de inserção gulosa

7.1 Introdução

Segundo Heckbert em [HG95], a medida de importância baseada em erro vertical local é a que fornece melhores resultados quando utilizada no método de inserção gulosa. Neste capítulo estudaremos o método de inserção gulosa com critério de seleção baseado no erro vertical local, abordando seus detalhes de implementação e apresentaremos os resultados de sua aplicação a modelos de terreno.

Através destes mesmos resultados observamos o comportamento desta medida de importância perante terrenos com regiões em diferentes escalas. Com base nestes testes, levantamos a possibilidade de que existam medidas mais adequadas a estes tipos de terreno, em que o erro vertical local é modificado de modo a refletir sua importância perante a variabilidade dos dados de altura na vizinhança do ponto considerado. Propomos, então, duas medidas de importância construídas de acordo com essas idéias, avaliamos seus comportamentos e comparamos seus resultados com a medida baseada puramente no erro vertical local.

7.2 Método de inserção gulosa com critério de seleção baseado no erro vertical local.

Este método desfruta de grande prestígio na literatura por ser um dos métodos de simplificação que fornece os melhores resultados.

O método de inserção gulosa com critério de seleção baseado no erro vertical local, procura a cada iteração selecionar o ponto que possui o maior erro vertical local. Isto pode ser expresso como:

selecionar p tal que $M(p)=|H(p)-TS(p)|$ é máximo

A idéia por trás deste critério de seleção é a de que se quisermos melhorar a precisão de nossa aproximação devemos procurar reduzir os erros locais máximos. Um ou outro movimento de uma determinada iteração pode aumentar o erro máximo local. Contudo, se prosseguirmos inserindo os pontos segundo este critério, certamente o erro máximo local convergirá a zero.

O algoritmo do método de inserção gulosa com critério de seleção baseado no erro local máximo utiliza como base o algoritmo de triangulação de Delaunay incremental. Apenas os critérios de parada e de seleção são modificados de forma a representarem nossa estratégia.

Apesar de haver poucas modificações em relação ao algoritmo de triangulação de Delaunay incremental, devemos ficar atentos no sentido de que a complexidade do método pode aumentar se não utilizarmos o critério de seleção de forma cuidadosa.

Se a cada iteração precisarmos calcular o erro local de cada ponto não inserido, para posteriormente selecionar o que tem o maior valor, iremos aumentar significativamente a complexidade do algoritmo.

O cálculo do critério de seleção pode ser efetuado de forma mais eficiente se observarmos que as modificações causadas pela inserção de um novo ponto na triangulação de Delaunay são locais, sendo que grande parte dos pontos continua com seus erros locais inalterados.

Isto indica que devemos utilizar uma estrutura que guarde o ponto de maior erro de cada triângulo existente na triangulação corrente. Assim quando formos selecionar um ponto, devemos selecionar o melhor candidato apenas dentre os pontos de maior erro de cada triângulo.

Na verdade podemos realizar este processo de forma ainda melhor. Se a estrutura utilizada for capaz de manter os pontos de maior erro de cada triângulo ordenados de forma dinâmica, podemos realizar a seleção em tempo $O(1)$. Exemplos de estruturas capazes de manter ordenações dinâmicas são os *heaps* e as árvores balanceadas.

Quando um novo ponto for inserido na triangulação, devemos reavaliar apenas os pontos dos triângulos modificados de forma a determinar o ponto de maior erro dos novos triângulos criados. Em seguida, estes pontos devem ser inseridos em nossa estrutura balanceada de forma a satisfazerem a configuração atual da triangulação. A inserção de um novo elemento numa estrutura de ordenação dinâmica pode ser feita em tempo $O(\log n)$.

À seguir encontra-se o algoritmo de inserção gulosa com as otimizações descritas acima:

7.3 Algoritmo

Selecionar_Ponto(Triangulação T , Heap h , Ponto p , Erro e)

```
{
  elemento=Remove_Heap (h);

  p= elemento.p;

  e =elemento.erro;
}
```

Algoritmo_de_Inserção_Gulosa(Conjunto_de_Pontos P , Triangulação T)

```
{
   $T = \{\}$ ;

  Construir uma triangulação mínima  $T$  definida pela envoltória convexa
  de  $P$ ;

  Para cada triângulo  $t \in T$ 
    Encontrar o ponto  $p_i \in t$  tal que  $e_i = |H(p_i)-T(p_i)|$  é máximo;
    Inserir_Heap( $h, p, e$ );

  Repita

    Selecionar_Ponto( $T, h, p, e$ );

     $P = P - \{p\}$ ;

    Localizar um triângulo  $t \in T$  tal que  $t$  contenha  $p$ ;

    Se  $p$  estiver sobre uma aresta  $e$  de  $t$  então remova  $e$ .

    Criar arestas conectando  $p$  aos vértices do polígono  $pl$  que o
    envolve;

    Atualizar a triangulação de forma que satisfaça o critério de
    Delaunay;

    Para cada novo triângulo criado  $nt_i$ 
      Encontrar o ponto  $p_i \in nt_i$  tal que  $e_i = |H(p_i)-T(p_i)|$  é
      máximo;
      Inserir_Heap( $h, p_i, e$ );

  Enquanto  $e < \text{tolerância}$ ;
}
```

7.4 Exemplos de aplicação do algoritmo a dados de terreno

Aplicamos o algoritmo acima sobre dois modelos de terreno. O erro final descrito nas tabelas é definido em termos de percentagem da variação máxima de altura do terreno(Figura 7.1).

Dados de terreno							
Nome	Dim X	Dim Y	Total de triângulos	Alt. Mínima	Alt. Máxima	Variação Máxima	Desvio Padrão
cvzbuffalo	120	120	28832	15,80	64,20	48,40	13,22
ilha	256	256	130050	0,070	17,08	17,01	3,39

Tabela 7.1 - Características dos dados de terrenos testes

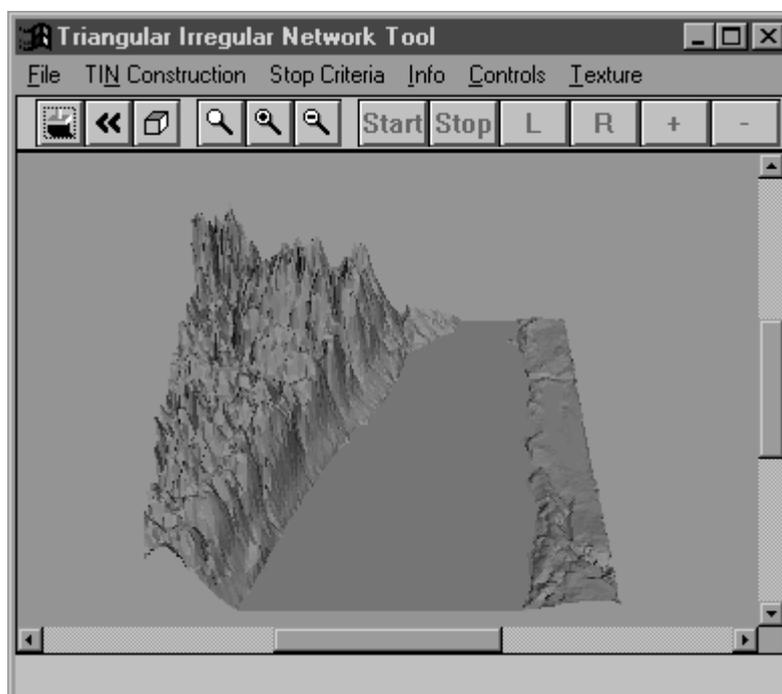


Figura 7.1 - Visualização de "Cvzbuffalo" simplificado

cvzbuffalo				
Nº de Triângulos	redução %	Nº de Arestas	Erro	Erro % (*)
2497	91,18	3788	2,10	4,34
4998	82,35	7572	1,00	2,07
7498	73,53	11351	0,50	1,03
9947	64,70	15136	0,33	0,69
12497	55,88	18908	0,08	0,16
14998	47,05	22682	0,00	0,00

Tabela 7.2 - Resultados da aplicação do método aos dados de "Cvzbuffalo"

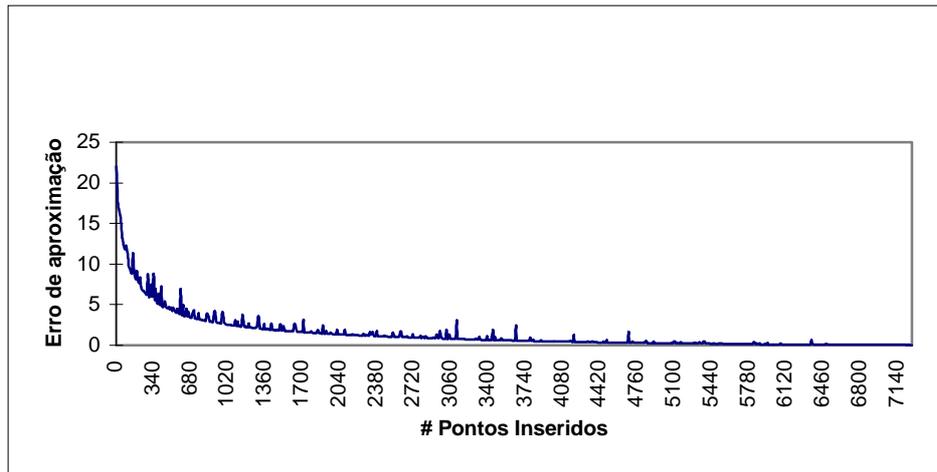


Figura 7.2 - Gráfico erro x # pontos da simplificação de “Cvzbuffalo”

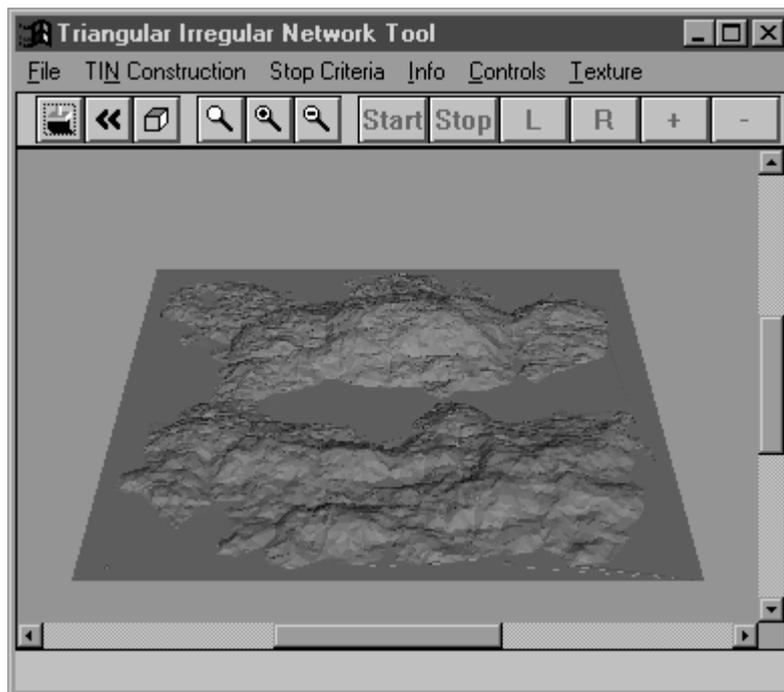


Figura 7.3 - Visualização de "Ilha" simplificado

ilha				
Nº de Triângulos	Redução %	Nº de Arestas	Erro	Erro% (*)
2498	98,08	3749	0,97	5,71
4998	96,16	7499	0,56	3,29
7498	94,24	11249	0,39	2,26
9998	92,31	14999	0,32	1,85
12498	90,39	18749	0,28	1,65
14998	88,47	22499	0,25	1,44

Tabela 7.3 - Resultado dos testes aplicados aos dados de "Ilha"

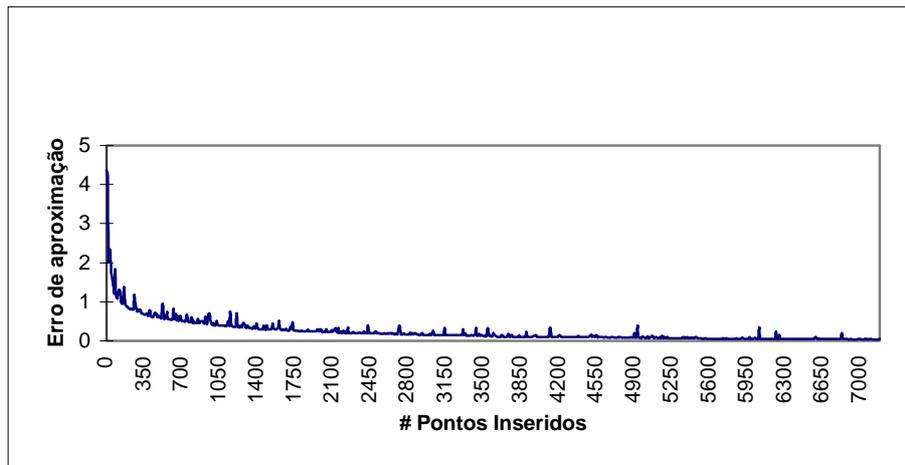


Figura 7.4 - Gráfico erro x # pontos inseridos da simplificação de "Ilha"

7.5 Análise do comportamento do critério de seleção baseado no erro vertical local

Apesar de fornecer resultados satisfatórios tendo-se em vista a precisão numérica, o método de inserção gulosa com critério de seleção baseado no erro vertical máximo pode apresentar alguns comportamentos indesejáveis em relação à precisão visual e à densidade dos triângulos nas regiões do modelo gerado.

O problema surge quando a superfície a ser simplificada tem como característica a presença de regiões de escalas bastante diferentes como na Figura 7.5.

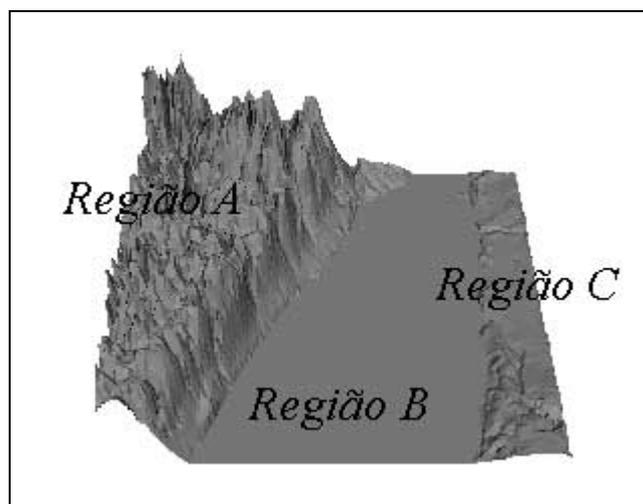


Figura 7.5 - Terreno com regiões com características distintas

Se aplicarmos o algoritmo de inserção gulosa com critério de seleção baseado no erro vertical local máximo sobre os dados de terreno com estas características, verificaremos que, na solução obtida, a região A ficará extremamente refinada, enquanto que a região C terá um refinamento pobre. Isto torna-se mais evidente quando observamos os resultados fornecidos com graus de refinamento mais baixos. Nestes a região C aparece como inexistente.

Isto se dá porque a região A possui variações de amplitude muita altas se comparadas às da região C. Isto faz com que os erros locais na região A tenham valor absoluto quase sempre maiores do que a região C, fazendo com que o algoritmo chegue à conclusão de que todos os pontos em A sejam muito mais representativos para a modelagem do terreno que os pontos de C(Figura 7.6).

Se fossemos construir uma pirâmide de aproximações com diferentes resoluções dos dados do terreno da figura 6.5, obteríamos resultados estranhos, pois a região C só apareceria nos estágios mais refinados, dando a falsa impressão que surgiu do nada .

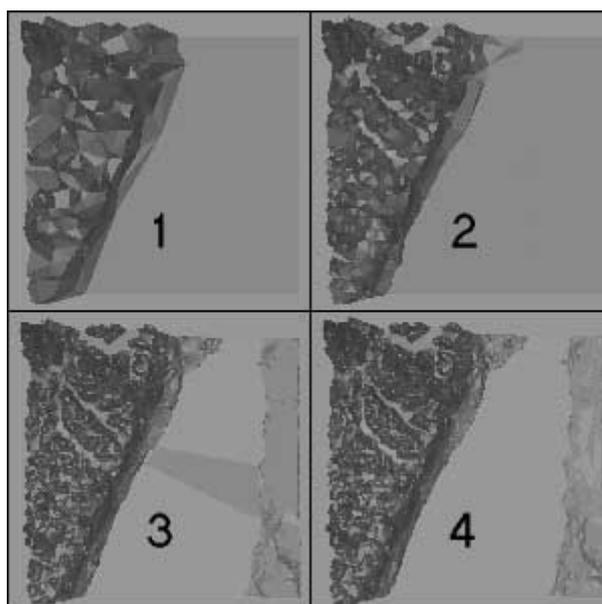


Figura 7.6 - Resultado de simplificação por critério de seleção global

Nível	Número de Triângulos	% triângulos utilizados	Erro	Erro% (*)
1	806	2,80	4,84	0,1
2	3151	10,93	2,42	0,05
3	7652	26,54	0,48	0,01
4	14192	49,22	0,24	0,005

Tabela 7.4 - Resultado do método com critério baseado no erro vertical local máximo sobre "Cvzbuffalo"

Observamos que este comportamento é resultado do fato de que o critério de seleção baseado unicamente no erro vertical local, apesar de utilizar uma medida local, faz comparações de forma global. Isto é, a comparação entre os erros verticais locais dos pontos é feita sem levar em consideração a importância destes erros nas regiões onde estão localizados. Em outras palavras o critério de seleção faz uma comparação que não leva em conta as particularidades de cada região, fazendo com que algumas regiões sejam prejudicadas quando comparadas com outras.

Agora que descrevemos a natureza do problema, iremos propor uma estratégia que trate destes casos de forma adequada.

7.6 Critério de seleção baseado no erro vertical local modificado

Para solucionar o problema de falta de adaptabilidade do algoritmo a dados de terreno nos quais coexistem regiões com diferentes escalas, propomos a adoção de um critério de seleção que leve em conta a importância local do erro vertical.

O objetivo desta estratégia é de que o algoritmo possa distribuir os pontos de forma mais inteligente, sem privilegiar de forma errônea certas regiões e preterir totalmente outras que seriam importantes para a modelagem de terreno.

A medição da importância local do erro vertical é feita através da medida de erro vertical local ponderada por algum parâmetro que seja sensível às características locais da região na qual o ponto se encontra inserido. Desta forma, quando o algoritmo seleciona o ponto de maior erro, já está automaticamente considerando a importância local do erro associado a este ponto. Estudaremos, portanto, medidas de importância da forma:

$$ML(p_i) = \frac{|H(p_i) - TS(p_i)|}{v(p_i)},$$

obtidas dividindo o erro vertical local por um parâmetro de ponderação.

Chamaremos tais medidas de *medidas de importância local do erro vertical*.

A seguir, discutiremos critérios a serem usados na escolha do parâmetro de ponderação.

7.6.1 Determinação do parâmetro de ponderação

O parâmetro utilizado para ponderar o erro vertical local deve ser capaz de medir a importância de um ponto dentro da sua região de vizinhança. A medida de importância de um ponto está relacionada não somente com seu erro vertical, mas também com a escala e a variabilidade da região a que pertence.

Propomos, então, que este parâmetro de ponderação seja representado por uma medida de variabilidade da região na qual o ponto se encontra inserido. Desta forma estamos adotando a seguinte estratégia: nas regiões em que existe uma desorganização natural (alta oscilação), a inclusão de um ponto apresentando determinado erro vertical faz menos efeito para o aumento da precisão perceptual do que em uma região que tenha naturalmente um bom comportamento (variação suave). Assim o efeito da escala de uma região é amenizado, impedindo que o algoritmo refine excessivamente regiões desnecessárias.

A determinação das regiões que contém amostras com características semelhantes é uma tarefa bastante importante na utilização do critério de seleção baseado em medida de importância local do erro vertical. É desejável que as regiões delimitem áreas da superfície com características semelhantes, de forma que a cada ponto seja associada uma medida de importância adequada.

Inicialmente tentamos adotar uma divisão dos dados em regiões uniformes, associando a medida de variabilidade (parâmetro de ponderação) de cada região a todos os pontos nela contidos. Esta metodologia não forneceu bons resultados, pois na maioria das vezes a subdivisão ideal dos dados em regiões não pode ser feita de forma regular. Um outro fator também a ser observado é o de que estas regiões interpenetram-se uma nas outras, devendo haver um tratamento especial no caso das fronteiras de forma que não haja perdas na continuidade da superfície. Podemos ilustrar este fato através de uma analogia com as paisagens geográficas. Por exemplo, a vegetação de uma região não muda de forma descontínua quando mudamos de uma região para a outra.

Ao invés de agruparmos conjuntos de pontos em regiões da superfície com características comuns, utilizaremos o conceito de **região de influência local** de um ponto. A região de influência local é determinada por um círculo de raio r centrado em um ponto p_i (Figura 7.7). Nesta estratégia, não associamos aos pontos de uma região um mesmo parâmetro de ponderação, e sim, cada ponto passa a ter seu próprio parâmetro em separado. Apesar de cada ponto possuir um valor próprio estes tendem a ser próximos em regiões vizinhas, fazendo com não haja perdas de continuidade como na primeira estratégia proposta.

Este processo torna o cálculo da medida de variabilidade bastante similar ao conceito de filtragem em processamento de sinais[MV95].

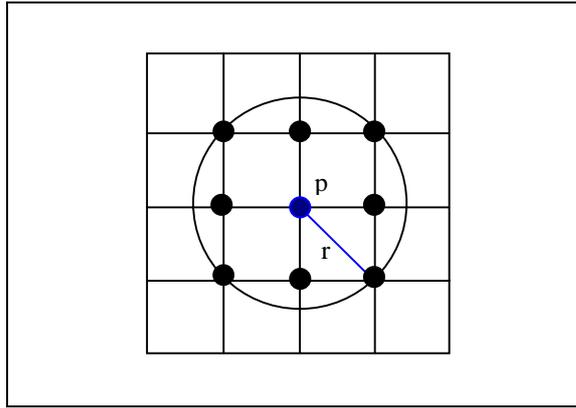


Figura 7.7- Definição da zona de influência

A escolha do tamanho da área, isto é do raio r , é importante no sucesso do algoritmo. Se tomarmos uma área muito pequena, a medida de variabilidade pode se tornar extremamente sensível a ruídos. Por outro lado, uma região muito grande pode associar medidas inadequadas de variabilidade, além de aumentar o custo do pré-processamento de forma desnecessária.

Abaixo descrevemos o algoritmo genérico representando o método de inserção gulosa com critério de seleção baseado na medida de importância local do erro vertical.

```

Para cada  $p_i \in DEM$  faça
{
  Calcule uma medida de variabilidade  $v(p_i)$  em alguma vizinhança  $R(p_i)$  ;

   $T = \text{Delaunay}(\text{FechoConvexo}(DEM))$ ;

  Repita
  {
    Selecione um novo  $p$ , tal que  $|p.z - TS(p)|/v(p_i)$  é máximo;
    Insira  $p$  em  $T$ ;

  }até (  $|p.z - TS(p)|/v(p_i) \leq \text{tolerância}/v(p_i)$ ;
}

```

Nas seções seguintes proporemos alguns parâmetros de ponderação e avaliaremos suas implicações no método de inserção gulosa com critério de seleção baseado no erro vertical local modificado.

7.7 Parâmetro de ponderação proporcional ao desvio padrão

Nesta proposta, tomamos a medida de variabilidade como sendo o desvio padrão em R_i :

$$v(p_i) = \sigma(R(p_i)),$$

A escolha desta medida de variabilidade faz com que a medida de importância local do erro vertical cresça nos pontos que pertencem a regiões em que o desvio padrão local é baixo e diminua onde o desvio padrão local é alto.

A razão entre o erro vertical local e o desvio padrão indica uma medida de quanto o erro se distancia do desvio natural da região. Se o erro de aproximação em um ponto for muito maior que o desvio padrão, significa que a aproximação da superfície em torno deste ponto precisa ser refinada através de sua inclusão. Se o erro estiver próximo ao desvio padrão, então o erro pode ser tolerado até que passe a ser importante em um estágio de refinamento maior.

Podemos observar que o erro vertical local medido sobre aproximações geradas pelo método que utiliza um critério baseado no erro vertical local modificado é maior do que o erro vertical local medido sobre uma aproximação com os mesmos dados e parâmetros gerada pelo método que utiliza o critério baseado puramente no erro vertical local. A razão deste comportamento é justificada pelo fato de que o critério de seleção baseado no erro vertical local utiliza uma métrica que não leva em consideração apenas o erro dado pela distância euclidiana vertical, mas também o grau de variabilidade em torno da região sendo considerada. Abaixo apresentamos uma tabela que demonstra o comportamento observado.

#Triângulos	erro vertical local		erro vertical local / desvio padrão (r=1)		erro vertical local / desvio padrão (r=5)		erro vertical local / desvio padrão (r=10)	
	Erro	Erro%	Erro	Erro%	Erro	Erro%	Erro	Erro%
1000	4,21	8,70	13,86	28,63	11,59	23,95	13,13	27,12
2000	2,50	5,16	14,50	29,94	8,66	17,89	7,40	15,28
3000	1,76	3,63	9,23	19,06	6,45	13,32	5,55	11,47
4000	1,30	2,69	7,15	14,77	5,40	11,15	4,50	9,29
5000	1,00	2,07	5,00	10,32	3,48	7,18	3,76	7,77

Tabela 7.5 - Resultados da aplicação da medida de variabilidade baseada no desvio padrão sobre "Cvzbuffalo"

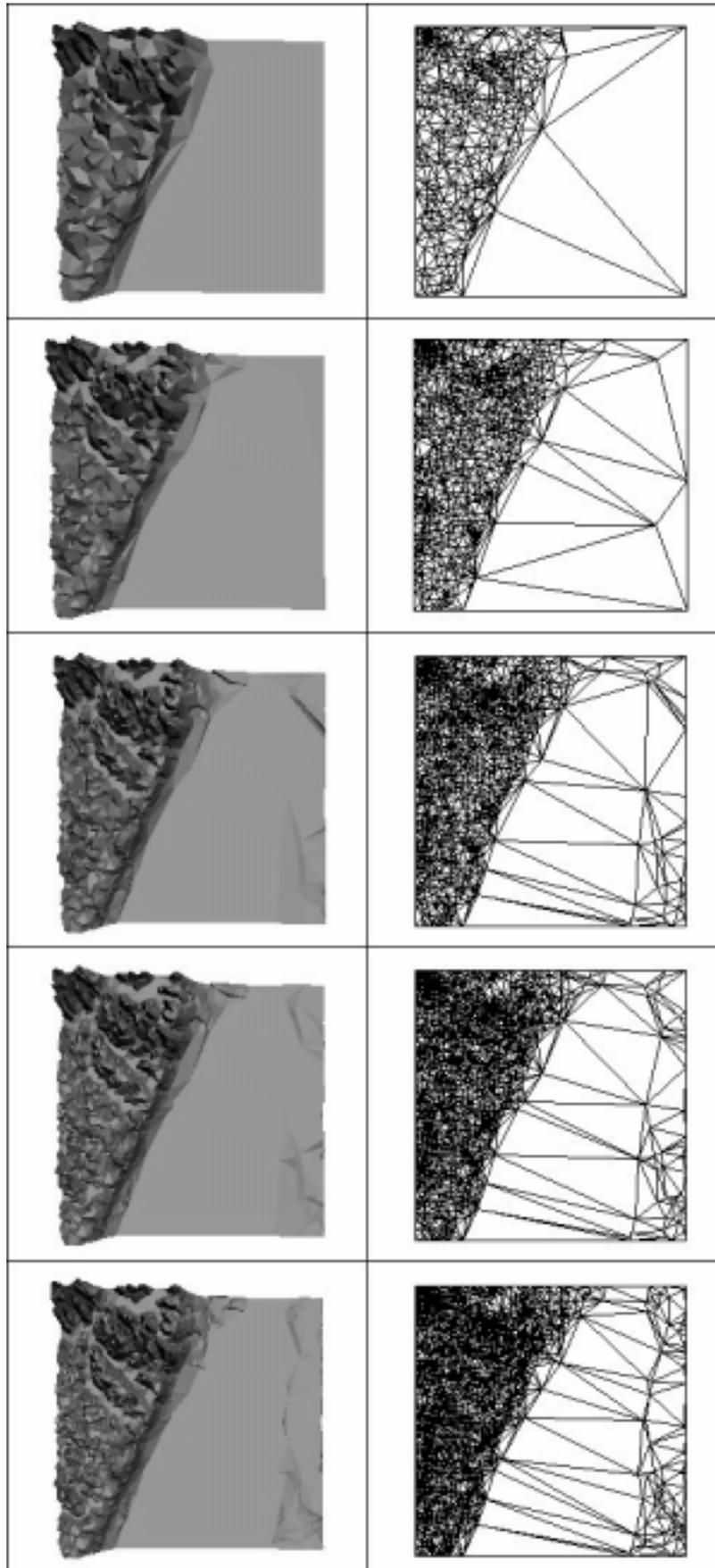


Figura 7.8 - medida de importância baseada puramente no erro vertical local -
Terreno: “Cvzbuffalo”

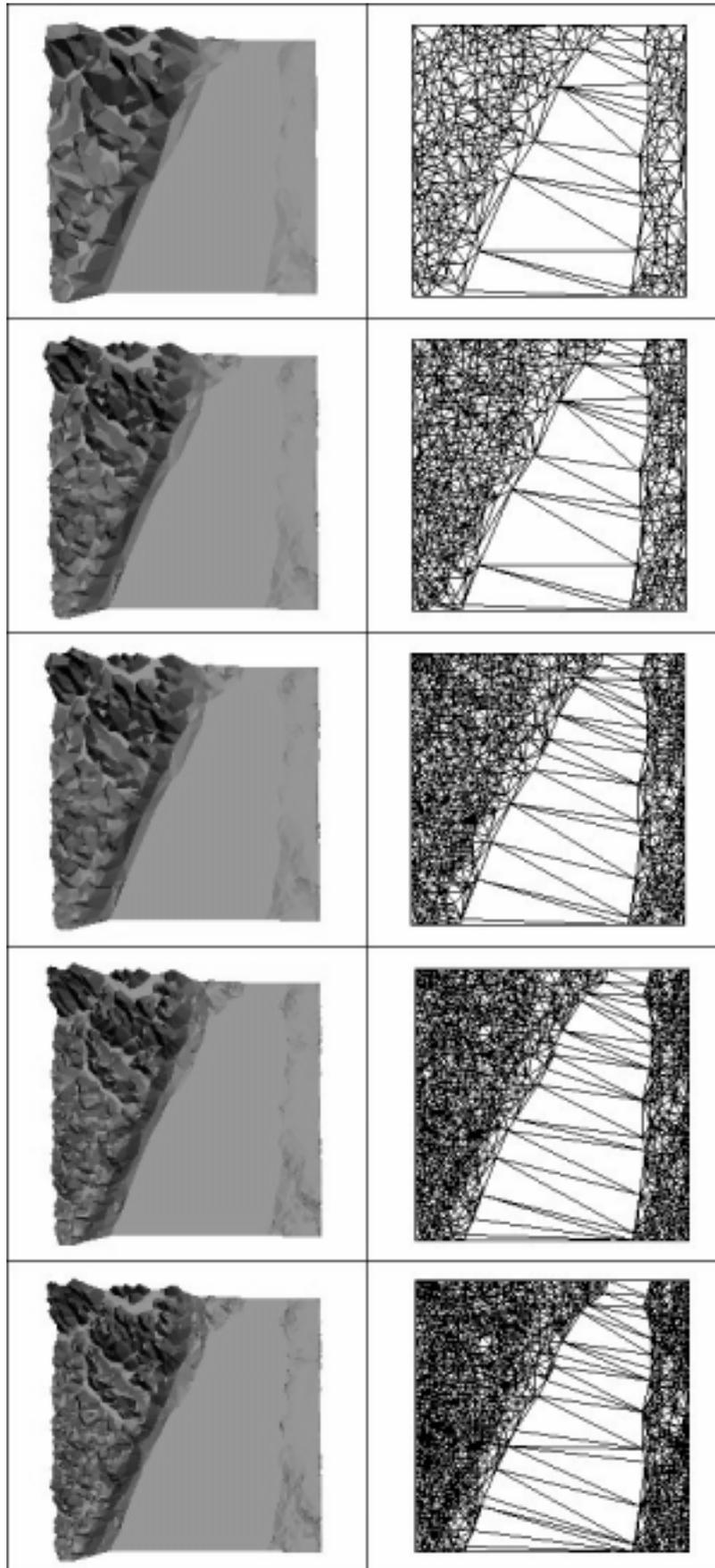


Figura 7.9 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=1$ - Terreno: “Cvzbuffalo”

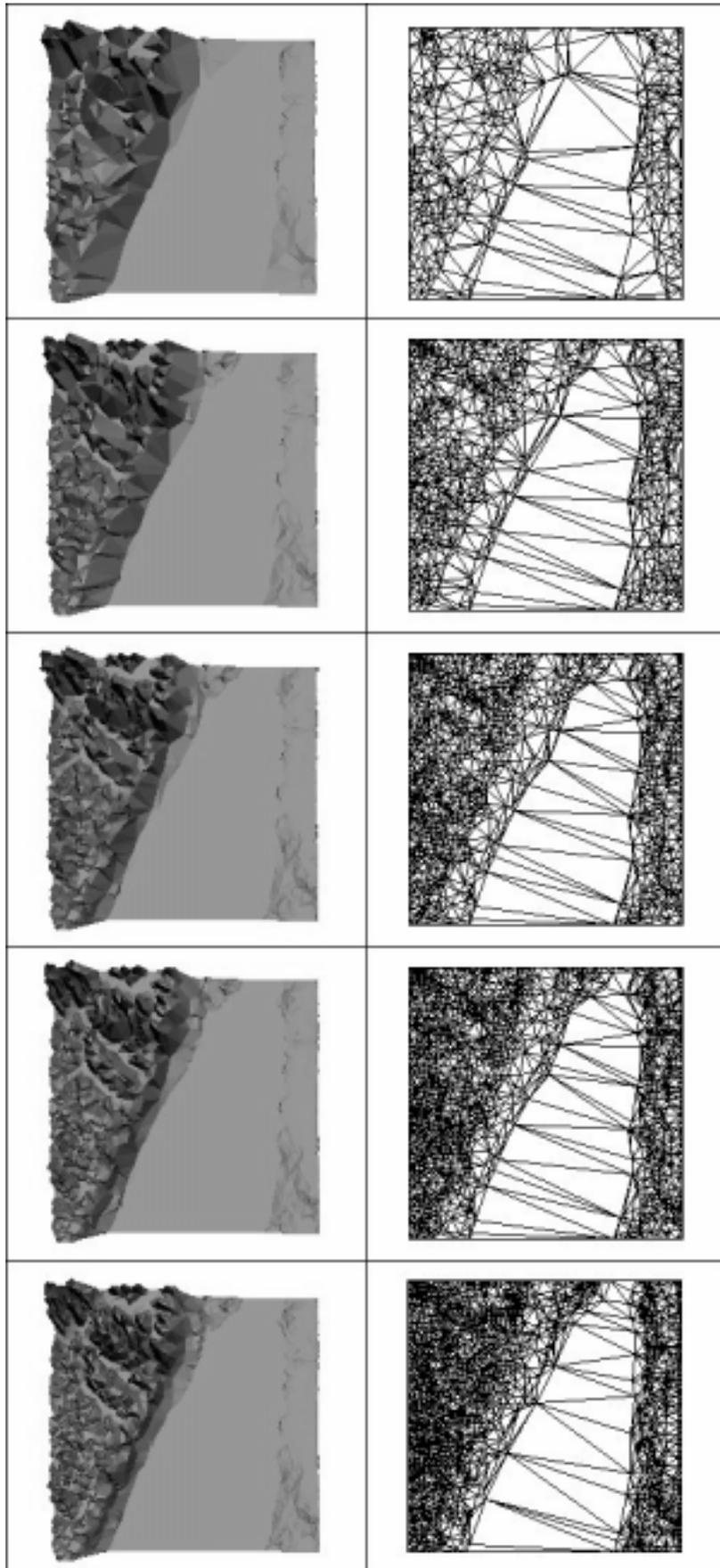


Figura 7.10 -medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=5$ - Terreno: “Cvzbuffalo”

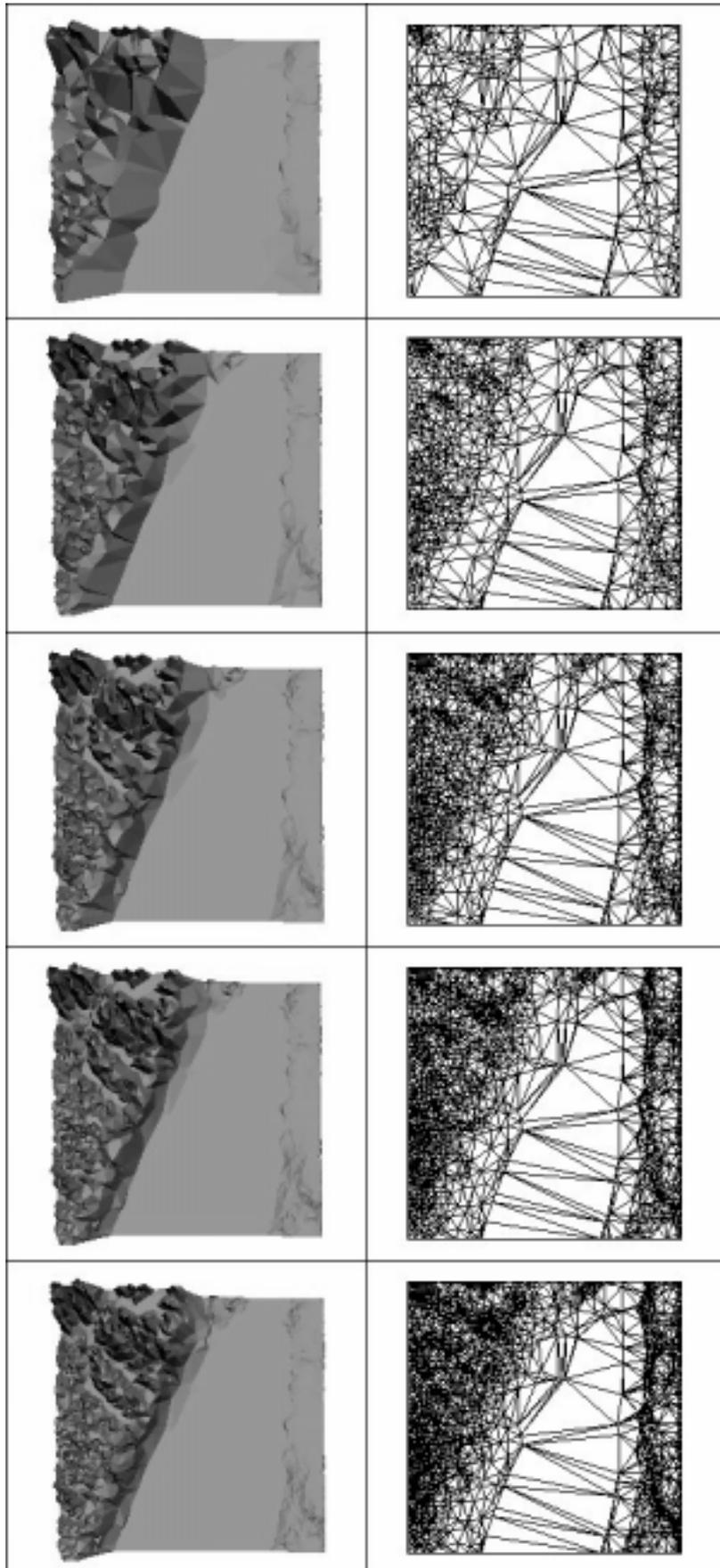


Figura 7.11- medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=10$ - Terreno: “Cvzbuffalo”

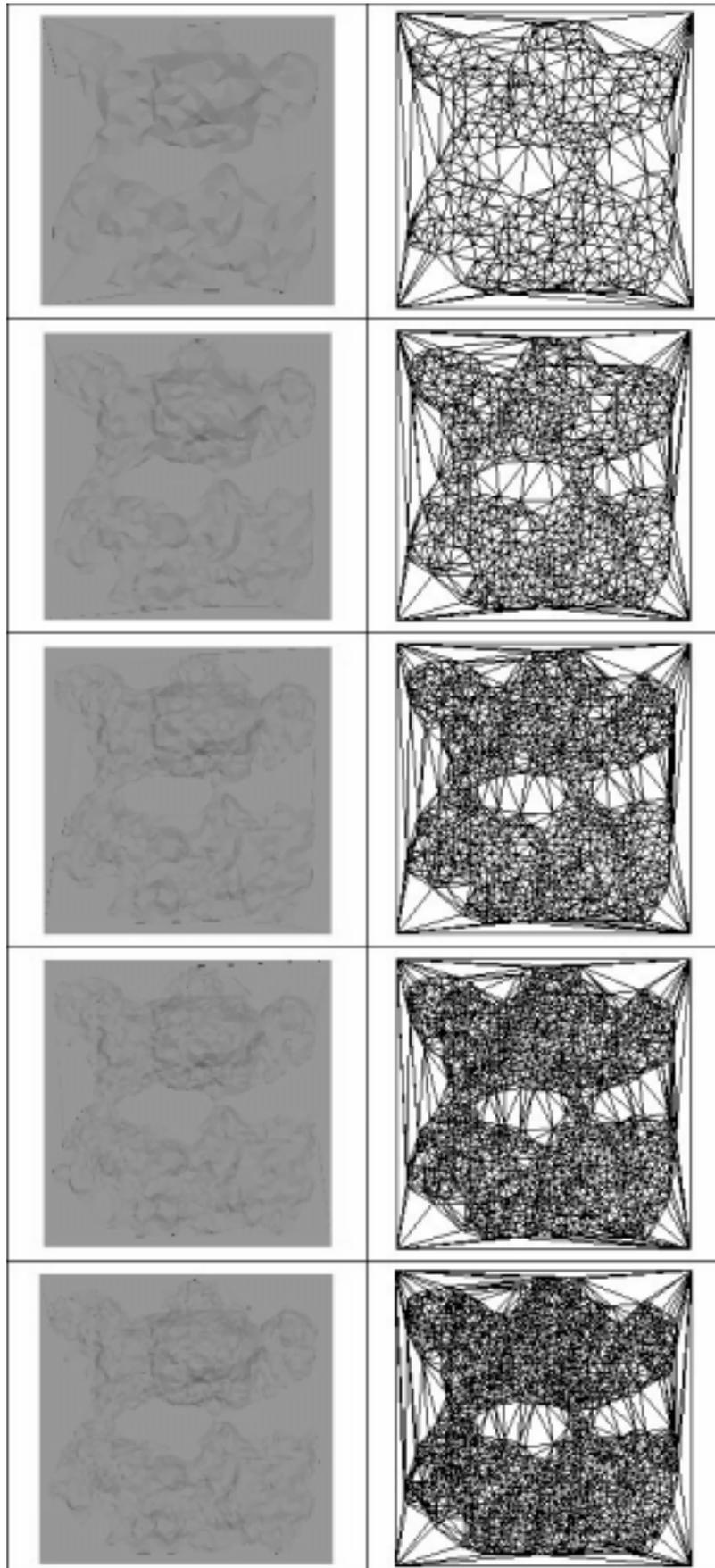


Figura 7.12 - medida de importância baseada puramente no erro vertical local -
Terreno: "Ilha"

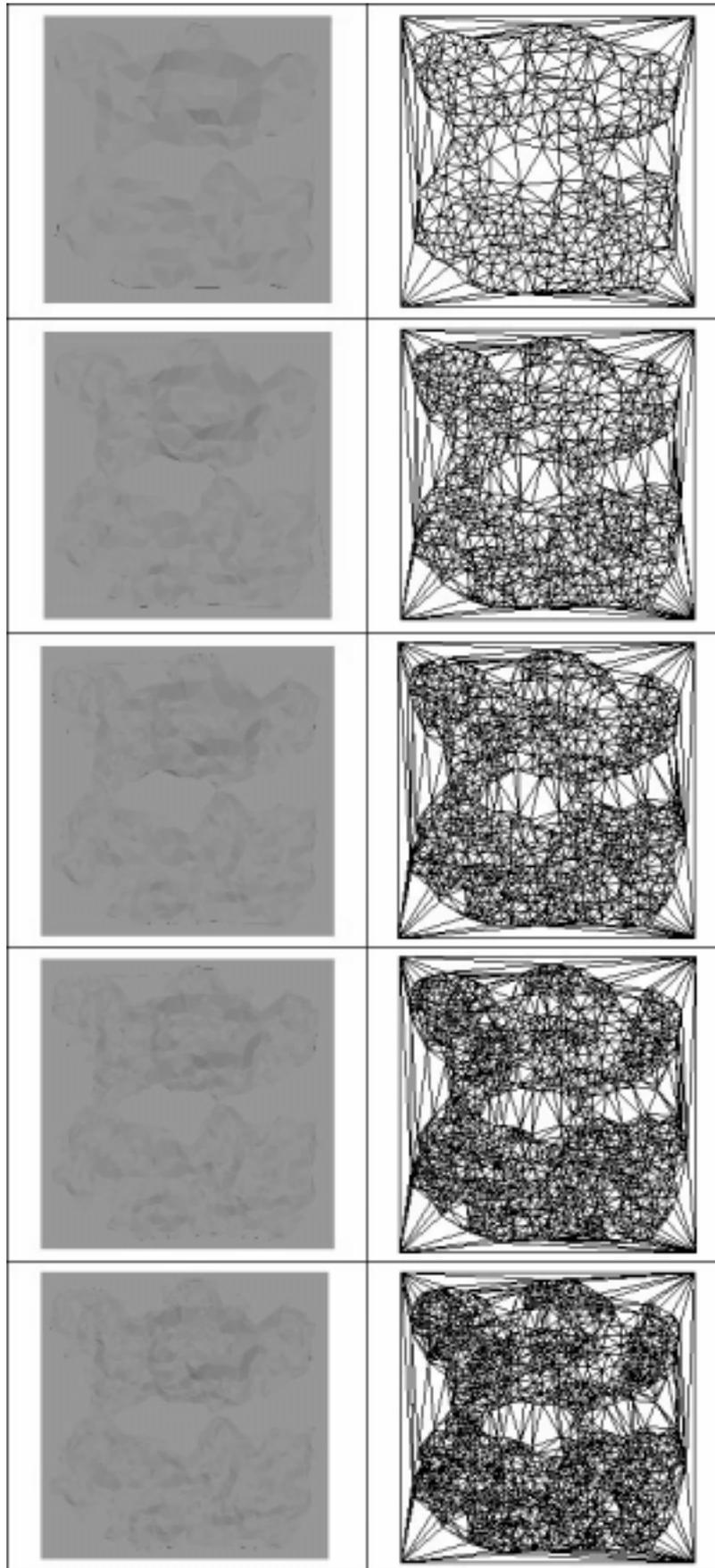


Figura 7.13 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=1$ - Terreno: “Ilha”

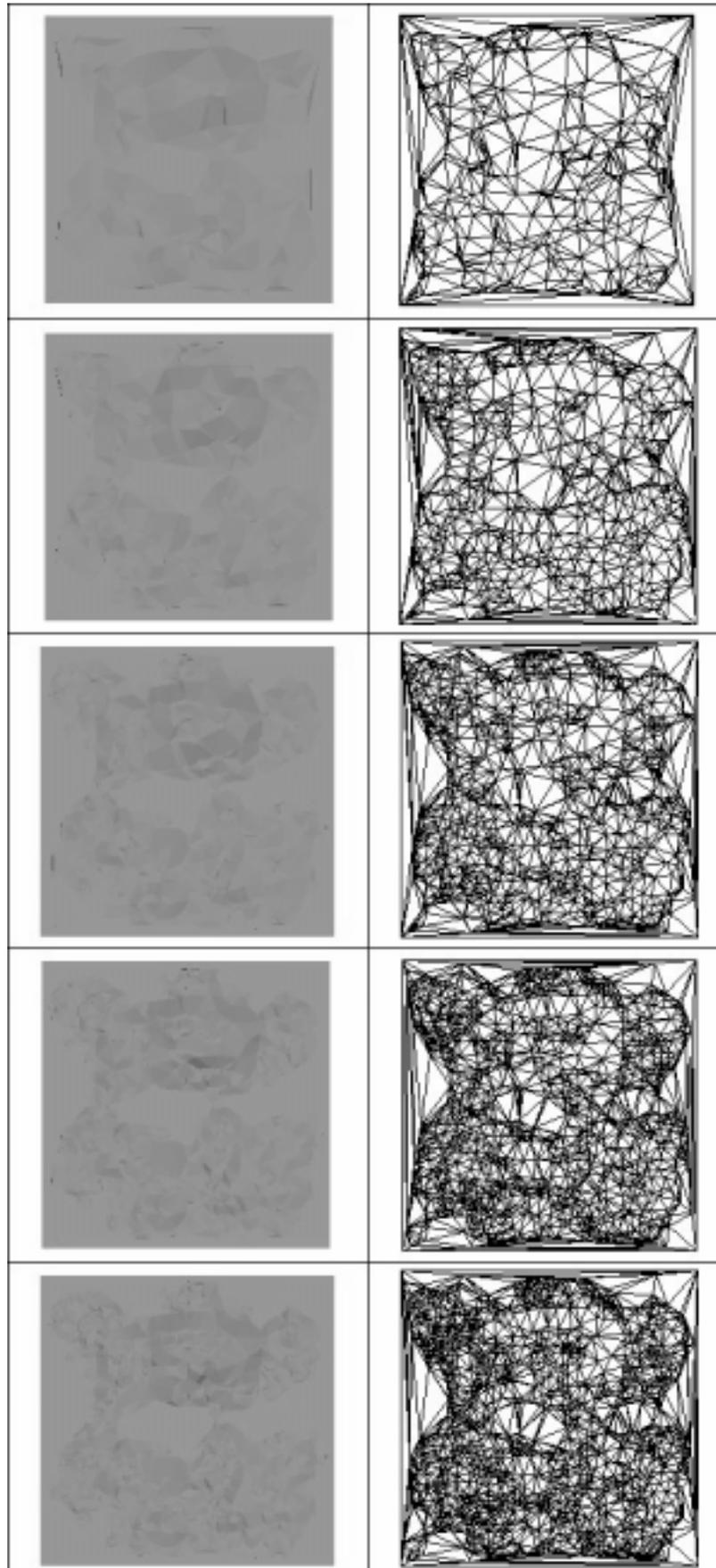


Figura 7.14 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=5$ - Terreno: “Ilha”

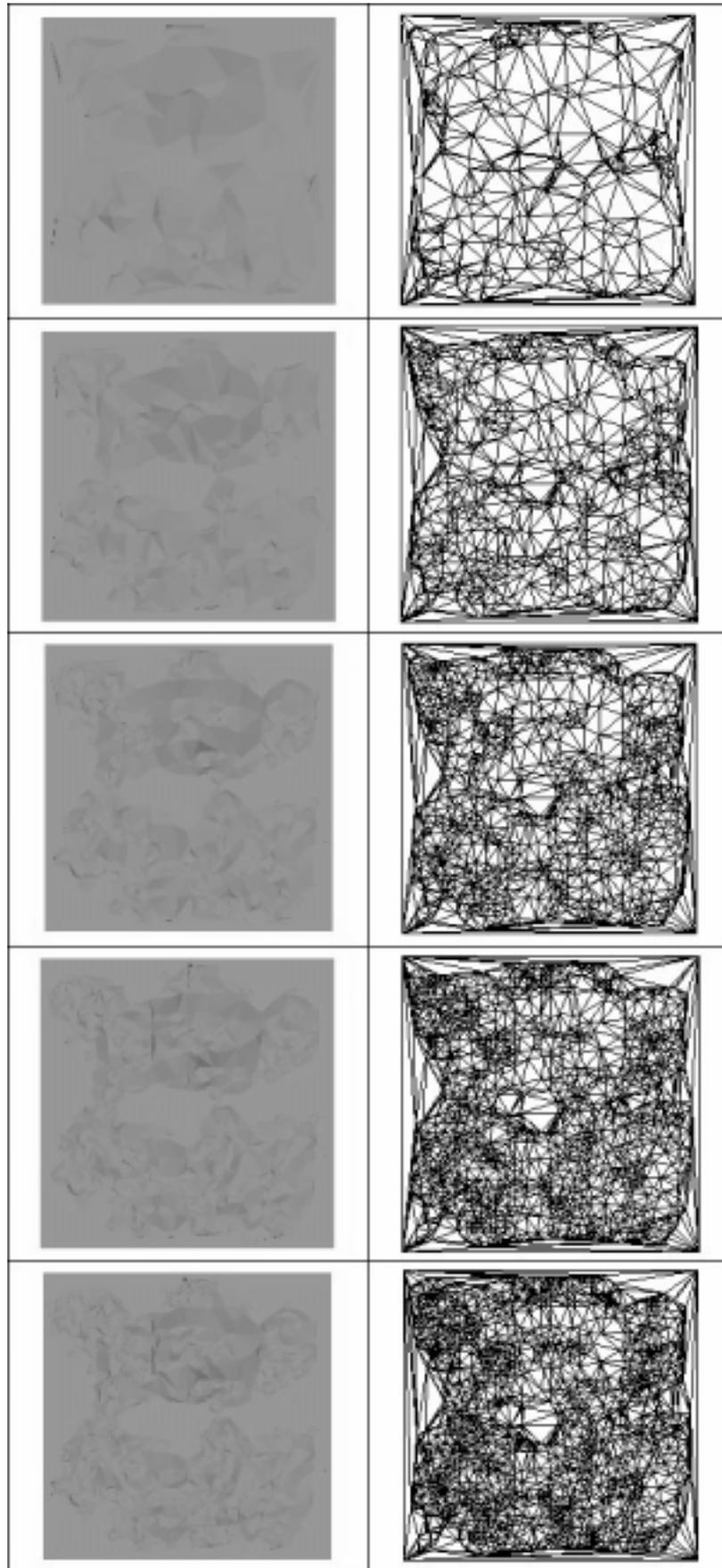


Figura 7.15 - medida de importância baseada no erro vertical local ponderado pelo desvio padrão com $r=10$ - Terreno: "Ilha"

7.7.1 Análise dos resultados

Observando os testes efetuados sobre o primeiro terreno (“Cvzbuffalo”) verificamos que o método de inserção gulosa com critério de seleção baseado unicamente no erro vertical local apresenta exatamente o comportamento descrito no início da seção anterior. Seu ímpeto de reduzir o erro local máximo é tão grande que o faz incapaz de capturar as características do relevo à direita do rio. O algoritmo é excessivamente influenciado pelos erros existentes na região rugosa com variação de alta amplitude, o que o faz desprezar os elementos de outras regiões do terreno.

Já o método com o critério de seleção baseado no erro vertical local ponderado pelo desvio padrão não fica retido nas regiões com alta variação, conseguindo assim capturar elementos do relevo à direita do rio desde os níveis mais baixos de refinamento (inserção de 1000 a 2000 triângulos).

Em relação ao segundo terreno (“Ilha”), observamos que a estratégia baseada no erro vertical local ponderado pelo desvio padrão não forneceu resultados tão bons quanto aos do algoritmo que utiliza critério de seleção baseado puramente no erro vertical local. Foi possível notar uma certa tendência à inserção de pontos em regiões de área bastante restrita, formando conglomerados. Este fenômeno ocorre devido ao fato de que determinadas regiões acabam sendo classificadas como de alta importância, levando a escolha de diversos pontos próximos de sua área de influência. Especificamente ao terreno “Ilha”, a concentração de pontos ocorreu nas regiões que determinam o seu contorno. Estas regiões apresentam baixo desvio padrão, já que são regiões de baixa elevação em relação ao resto do terreno, tornando a razão **erro local/desvio padrão**, em geral com valor absoluto maior que as demais áreas do terreno.

Em relação ao tamanho da área de influência, chegamos a conclusão, de certa forma surpreendente, de que os melhores resultados obtidos, em geral ocorreram quando o raio de influência é mínimo, isto é, $r=1$.

Todas estas observações podem ser explicadas pelo fato de que o parâmetro determinado pelo desvio padrão não fornece uma medida de variabilidade adequada aos nossos propósitos. O desvio padrão é muito sensível às variações de altura e não às “oscilações”, que é o que desejamos. No caso do terreno “Ilha”, o desvio padrão assume valores altos em pontos pertencentes às regiões de alta declividade fazendo com que elas fossem consideradas como sendo bastante desorganizadas, apesar de não serem. Este comportamento faz com que o algoritmo crie áreas com baixa densidade de pontos nas montanhas maiores e áreas de alta concentração de pontos nos relevos mais baixos.

Por este motivo, a estratégia forneceu resultados melhores quando escolhemos uma região de influência mínima ($r=1$). Como a área é pequena, o desvio padrão não sofreu tanta influência dos valores extremos, conseguindo assim avaliar a importância local do erro vertical de forma razoável.

O uso do desvio padrão como parâmetro de ponderação, apesar de não ser exatamente o parâmetro que procuramos, foi capaz de fornecer pistas de que uma medida de importância baseada no erro vertical local ponderado por uma medida de variabilidade das regiões da superfície pode ser uma boa estratégia para resolver o problema de modelagem de terrenos compostos por regiões com altitudes diferentes.

7.8 Parâmetro de ponderação proporcional à norma da segunda derivada.

Com base nos resultados obtidos pela aplicação do critério de seleção baseado no erro vertical local ponderado pelo desvio padrão, propomos um parâmetro de ponderação proporcional à norma da segunda derivada.

A segunda derivada tem a vantagem de ser sensível à taxa de variação da declividade, sendo bem mais próxima da medida de variabilidade que desejamos.

Uma aproximação discreta da segunda derivada de um campo de alturas discretizado pode ser obtida da seguinte forma:

$$f_{11}(x,y) \approx f(x+1) - 2f(x) + f(x-1),$$

$$f_{22}(x,y) \approx f(y+1) - 2f(y) + f(y-1)$$

O valor do parâmetro de ponderação em um ponto $p_i(v(p_i))$ é determinado pela expressão:

$$v(p_i) = |f(x+1) - 2f(x) + f(x-1)| + |f(y+1) - 2f(y) + f(y-1)|$$

Tomamos o módulo das derivadas parciais de segunda ordem de forma que pontos distribuídos em torno de um ponto de sela não levem a uma anulação do valor de $v(p_i)$. Esta distribuição de pontos deve ser encarada com uma região de grande variação do gradiente, e não de variação nula.

De forma análoga ao parâmetro determinado pelo desvio padrão, a idéia proposta é de diminuir a importância de pontos com alta variação e aumentar a importância dos pontos onde a variação é mais suave.

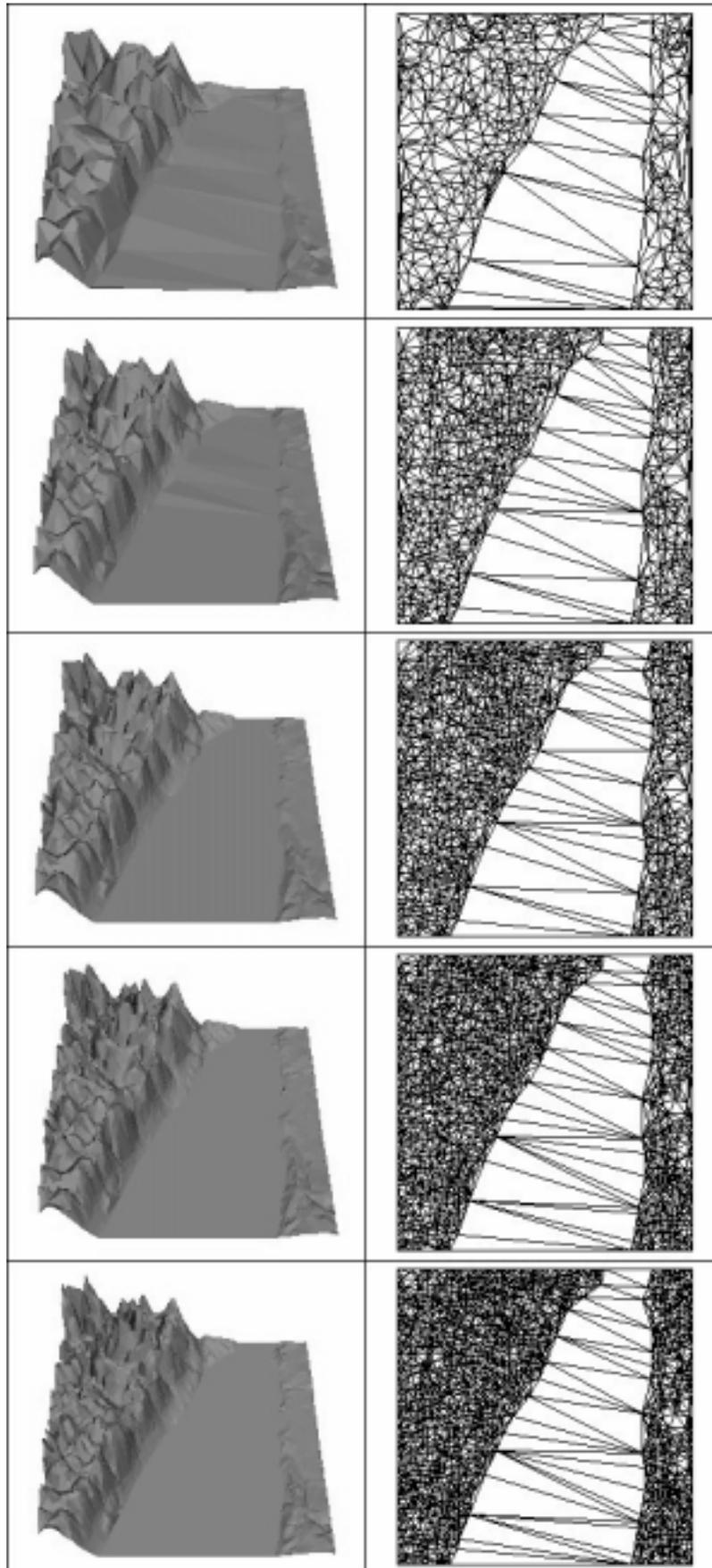


Figura 7.16 - medida de importância baseada no erro vertical local ponderado pela norma da derivada segunda - Terreno : “Cvzbuffalo”

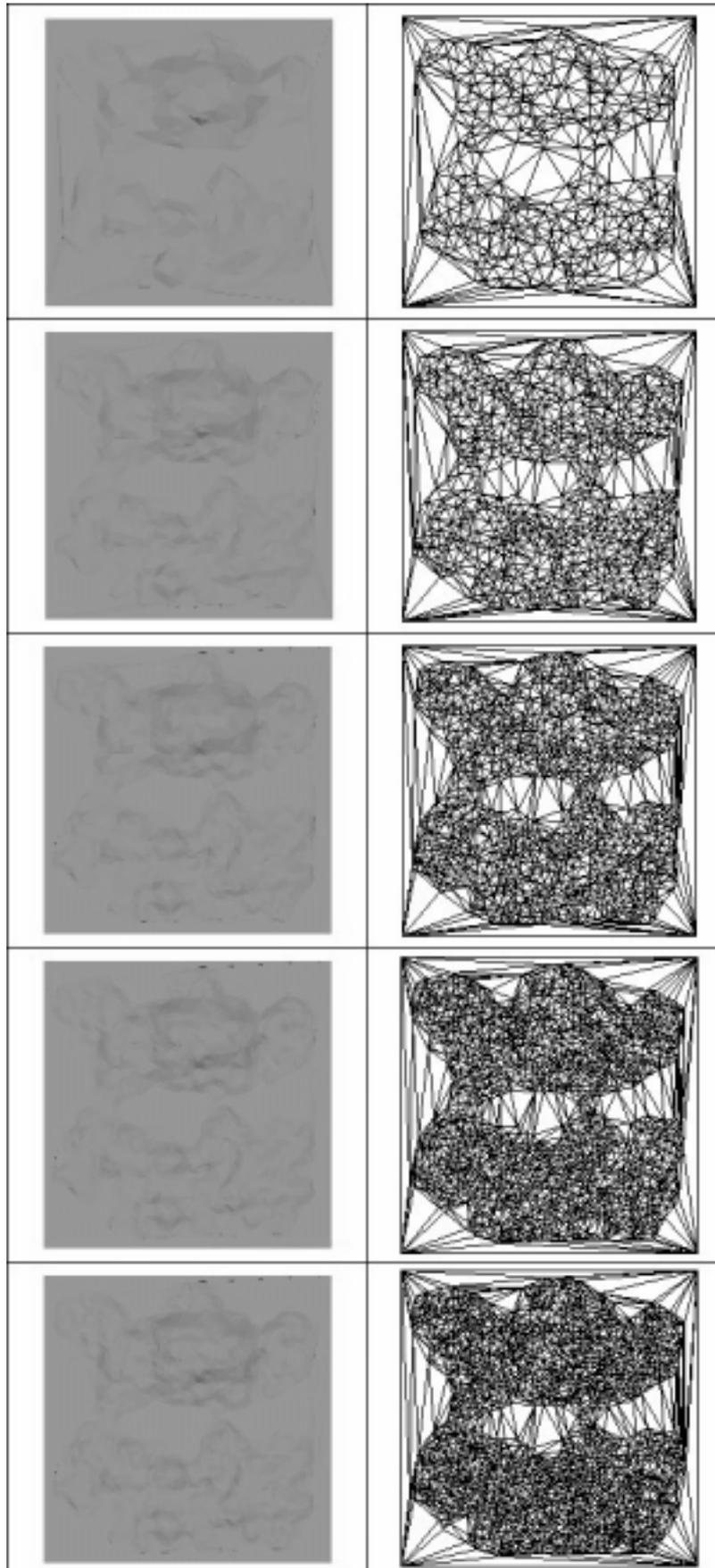


Figura 7.17 - medida de importância baseada no erro vertical local ponderado pela norma da derivada segunda - Terreno : "Ilha"

7.8.1 Análise dos resultados

Observando os testes realizados com o parâmetro de ponderação determinado pela norma da segunda derivada, podemos notar que os modelos gerados não mais apresentaram concentrações indesejáveis de pontos.

Em ambos os dados de terreno utilizados nos testes, o algoritmo que utiliza este parâmetro de ponderação conseguiu capturar modelos bastante representativos das características dos dados.

Por outro lado, este parâmetro não consegue capturar de forma ideal os contornos das regiões (por exemplo, os contornos da margem do rio de “Cvzbuffalo”). Observamos que os contornos são regiões de altíssima importância para a representação do terreno, levando à conclusão de que este parâmetro de variabilidade também apresenta suas falhas.

Apesar disso, o parâmetro de variabilidade baseado na derivada segunda dos pontos da superfície fornece um efeito muito interessante. Por estar relacionado a um filtro de passa baixa, este parâmetro permite que os processos de simplificação gerem suavizações em modelos tridimensionais. Podemos verificar através das ilustrações que a iluminação das faces das regiões vizinhas apresenta um grau de continuidade muito maior que a iluminação de modelos simplificados através de outras medidas de importância (Quanto menos refinado for o modelo, mais a superfície apresenta um aspecto desfocado (*blur*), e quanto mais refinado o modelo mais claro se tornam as características do mesmo).

As seguintes figuras ilustram o comportamento descrito acima. Na Figura 7.18 temos um terreno simplificado pelo critério de seleção baseado no erro vertical local puro contendo 10000 triângulos e na Figura 7.19 o mesmo terreno simplificado pelo critério de seleção baseado no erro vertical local ponderado por uma norma da segunda derivada. Podemos observar que apesar do mesmo número e densidade de triângulos que a simplificação obtida na segunda figura apresenta um aspecto bem mais suave que o da primeira.

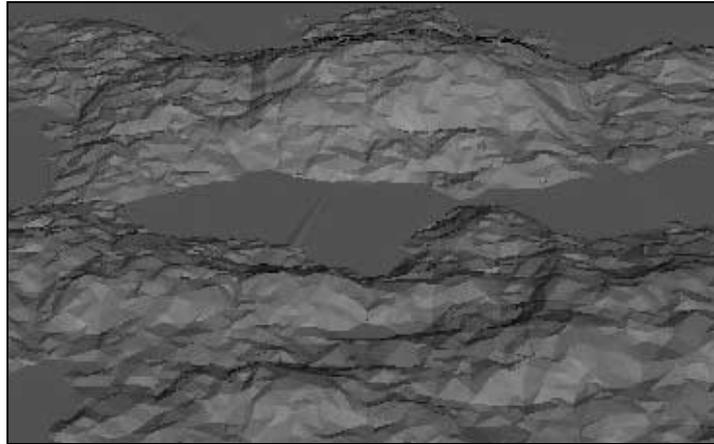


Figura 7.18 - Terreno simplificado com critério baseado no erro vertical local

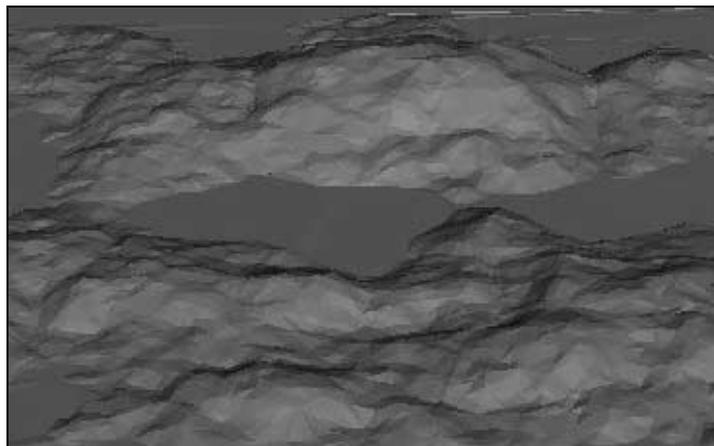


Figura 7.19 - Terreno simplificado com critério de seleção baseado no erro vertical local ponderado pela segunda derivada

Este comportamento pode ser bastante útil se desejarmos construir estruturas multiresolução.

8. Conclusões

Neste trabalho procuramos desenvolver um estudo sobre métodos de representação e manipulação de dados de terreno.

Verificamos a necessidade de utilização de estruturas e métodos capazes de lidar com toda a complexidade inerente aos dados de terreno de forma a possibilitar a construção de métodos e formas de representações eficientes, em especial os métodos de simplificação de superfícies.

Dentre todos os métodos de simplificação de superfícies, os métodos de inserção gulosa demonstraram ser bastante promissores no auxílio à obtenção de melhores representações de dados de terreno através de TINs. A grande vantagem dos métodos de inserção gulosa são a simplicidade, flexibilidade e facilidade de controle.

Através de testes empíricos e análises minuciosas do processo, podemos constatar que apesar de fornecerem bons resultados na maioria dos casos, os métodos por inserção gulosa baseados unicamente no erro vertical local apresentam algumas falhas no que diz respeito à qualidade perceptual dos modelos gerados quando a superfície apresenta regiões com altitudes diferentes. Os métodos de simplificação fornecem excelentes resultados nos níveis de refinamento mais altos, mas deixam a desejar nos níveis de refinamento mais baixos.

Com o intuito de solucionar este problema propomos critérios de seleção que sejam sensíveis à importância local do erro vertical dos pontos. Este critério de seleção se baseia em uma medida de importância obtida dividindo-se o erro vertical local por uma medida de variabilidade da superfície. Avaliamos dois deles: o critério de seleção com medida de variabilidade baseada no desvio padrão e o critério de seleção com medida de variabilidade baseado na derivada segunda.

Uma outra observação que deve ser feita é de que os critérios de seleção que utilizam medidas sensíveis à importância local do erro vertical fornecem resultados com precisão numérica bem inferior aos apresentados pelo critério de seleção baseado unicamente no erro vertical local. Isto é um comportamento natural, visto que estamos sacrificando um pouco da precisão numérica numa tentativa de melhorar a precisão perceptual. Esperamos encontrar, através de novas pesquisas, critérios que não se distanciem excessivamente da precisão numérica do critério baseado no erro vertical local.

Apesar dos resultados das duas medidas propostas neste trabalho não terem sido capazes de resolver todos os casos possíveis, eles foram importantes no sentido de demonstrar que um critério de seleção que utilize uma medida adequada de importância local do erro vertical pode fornecer boas soluções para o problema de simplificação de superfícies com regiões com altitudes diferentes.

No momento estamos investigando a utilização de uma medida que calcule a variabilidade com base no erro quadrático dos pontos em relação a um plano que aproxime os pontos em uma vizinhança do ponto considerado. Acreditamos que esta medida consiga corrigir as falhas existentes na parâmetro de variabilidade baseado no desvio padrão.

Concluimos, finalmente, que a representação de dados de terreno através de TINs é um campo de pesquisa extremamente em aberto com diversos problemas a serem resolvidos.

8.1 Propostas para trabalhos futuros:

Este trabalho apenas deu início a um conjunto de problemas que deverão ser tratados em trabalhos futuros. Dentre os temas a serem abordados destacamos os seguintes:

- Estudo sobre novos critérios de seleção de pontos. Este é um tema que continua sendo investigado pelo autor deste trabalho. Em especial, está sendo investigado um critério de seleção que utilize uma medida de importância ponderada pelo erro quadrático médio.
- Estudo e desenvolvimento de estruturas para representação em multiresolução. Este tema tem recebido especial atenção em modelagem e visualização de dados de terreno pelos pesquisadores de áreas afim[GU97]. Uma proposta para futuros trabalhos é adaptar a estrutura topológica utilizada neste trabalho de forma que possa suportar multiresolução. Caso não seja possível, o estudo de novas estruturas é um assunto bastante relevante que pode ser abordado.
- Tratamento de dados de terreno extremamente extensos. Este é um tema de extrema importância em modelagem de terreno. Os sistemas comerciais atuais que se propõem a modelagem de terreno, em especial alguns módulos de SIGs, não são capazes de lidar de forma conveniente com dados extremamente grandes. Propomos o desenvolvimento de métodos de simplificação para tratamento dos dados em memória secundária. Em relação ao tema de multiresolução já discutido no item anterior, propomos o estudo de uma estrutura que possa armazenar e tratar multiresolução em memória secundária, visto que as estruturas propostas atualmente não são eficientes no tratamento de dados reais.
- Estudo e aplicação de técnicas para visualização em tempo real. Este é também um outro problema de alta relevância em visualização de dados de terreno. A visualização em tempo real de dados de terreno representadas por TIN é extremamente complicada. A aplicação de texturas é um dos fatores que determina um excessivo *overhead* no processo de *rendering* de dados de terreno. A utilização de estratégias de *clipping* capazes de usufruir das propriedades existentes nas estruturas topológicas pode fornecer bons

resultados no aceleração do processo de visualização. Este também é um assunto que já está sendo estudado pelo autor.

- Extensão do método de simplificação para sólidos. É natural conjecturar a respeito da aplicação dos métodos aqui descritos em modelagem de sólidos. Certamente algumas adaptações deverão ser efetuadas no intuito de aplicar os métodos aqui abordados, uma vez que são destinados a campos de altura (*height fields*). As áreas que poderão usufruir do resultados obtidos através dos estudo destas técnicas são a visualização de dados provenientes da medicina, geologia e outras.

9. Apêndice: *Quad-edge data structure*

A *quad-edge data structure* foi proposta por Guibas e Stolfi como uma estrutura topológica alternativa para a manipulação de subdivisões genéricas. Esta estrutura tem como característica especial a capacidade de representar simultaneamente uma subdivisão planar e o seu diagrama dual. Devido a esta propriedade, ela torna especialmente fácil calcular o Diagrama de Voronoi de um conjunto de pontos a partir de sua Triangulação de Delaunay.

Uma outra característica que diferencia a *quad-edge data structure* das demais estruturas topológicas é o fato de que ela é totalmente baseada em um rigorosa álgebra de topologias denominada pelos autores de *edge algebra*. A *edge algebra* captura precisamente todas as propriedades topológicas de um subdivisão.

Os detalhes da *edge algebra* não estão dentro do escopo deste trabalho, mas podem ser encontrados em [GS85].

9.1 Definição da *quad-edge*

Antes de definirmos detalhadamente a *quad-edge data structure*, especificaremos uma notação capaz de representar as relações existentes entre as arestas de uma subdivisão. Esta notação é baseada na *edge algebra* de Guibas e Stolfi.

Seja e uma aresta orientada pertencente a uma subdivisão S . Temos então a seguinte notação para cada uma das relações existentes entre uma aresta de uma subdivisão planar e as entidades (vértices, arestas e faces) a ela relacionadas (Figura 9.1):

- $e.Org$ - vértice origem de e .
- $e.Dest$ - vértice destino de e .
- $e.Onext$ - próxima aresta no sentido anti-horário com a mesma origem que e .
- $e.Oprev$ - aresta anterior no sentido anti-horário com a mesma origem que e .
- $e.Dnext$ - próxima aresta no sentido anti-horário com o mesmo destino que e .
- $e.Dprev$ - aresta anterior no sentido anti-horário com o mesmo destino que e .
- $e.Left$ - face à esquerda de e .
- $e.Right$ - face à direita de e .
- $e.Lnext$ - próxima aresta no sentido anti-horário na face à esquerda de e .
- $e.Rnext$ - próxima aresta no sentido anti-horário na face à direita de e .
- $e.Sym$ - aresta simétrica a e .
- $e.Flip$ - aresta obtida invertendo-se a orientação de e .

- e.Dual - aresta dual de e (por convenção o dual tem orientação contrária a de e).
- e.Rot - aresta dual de e direcionada da direita para sua esquerda. (*)
- e.InvRot - aresta dual de e direcionada da esquerda para sua direita. (**)
- (*),(**) e.Rot é equivalente a e.Dual.Flip.

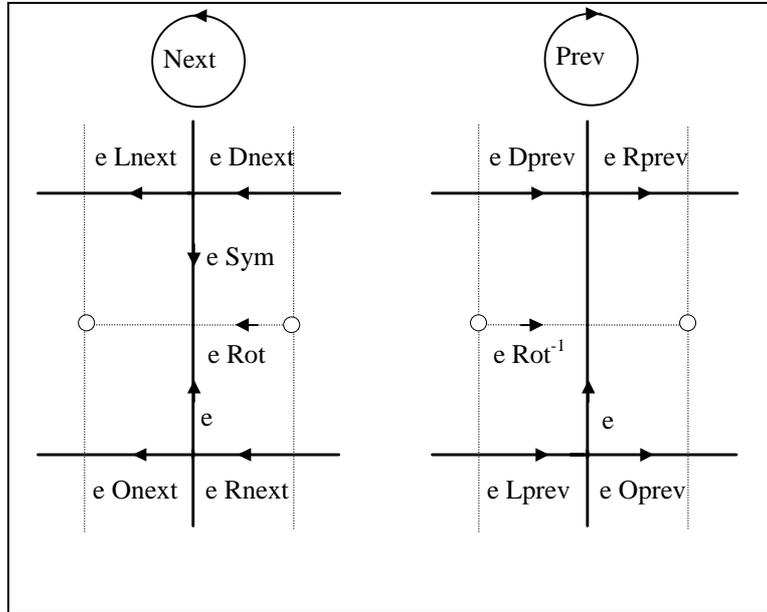


Figura 9.1 - Álgebra de arestas (edge algebra)

A quad-edge é formada pela composição de oito arestas direcionadas e orientadas. Cada grupo pode ser subdividido em 2 subgrupos de quatro, sendo que um deles pertence a subdivisão S e o outro ao dual S^* de S .

O grupo que contém uma aresta particular e é gerado a partir da aplicação da sub-álgebra Rot e Flip sobre e . Para construirmos a estrutura de dados escolhemos uma aresta canônica representativa em cada grupo. Qualquer aresta do grupo é definida por $e_0 \text{Rot}^r \text{Flip}^f$, onde $r \in \{0,1,2,3\}$, $f \in \{0,1\}$ e e_0 é a aresta canônica representativa.

A nível de implementação, o grupo de arestas que contém uma aresta e é representado por um *edge record* dividido em quatro partes: $e[0]$, $e[1]$, $e[2]$ e $e[3]$. Cada parte $e[r]$ corresponde a um subgrupo $e_0 \text{Rot}^r$. Uma aresta genérica $e = e_0 \text{Rot}^r \text{Flip}^f$ é representada pela tupla (e,r,f) denominada referência a uma aresta. Esta tupla é um ponteiro para uma parte $e[r]$, mais um *bit* (f) que informa a orientação (indica se devemos observá-la de cima para baixo ou de baixo para cima).

Cada parte $e[r]$ contém dois campos: *Data* e *Next*. O campo *Data* é usado para guardar as informações geométricas e outras informações não topológicas sobre aresta do grupo Rot^r . Este campo nunca é alterado pelas operações topológicas.

O campo *Next* de $e[r]$ contém uma referência para $e_0 \text{Rot}^r \text{Onext}$, isto é, para a próxima aresta no sentido anti-horário com mesma origem que $e_0 \text{Rot}^r$.

As funções básicas de arestas são definidas da seguinte forma:

$$\begin{aligned}(e,r,f)\text{Rot}&=(e,r+1+2f,f) \\ (e,r,f)\text{Flip}&=(e,r,f+1), \\ (e,r,f)\text{Onext}&=(e[r+f].\text{Next})\text{Rot}^f\text{Flip}^f\end{aligned}$$

Sobre os campos r e f são aplicados respectivamente o módulo por 4 e por 2. Isto implica que as operações executadas no sentido anti-horário em um lado do *manifold* correspondem a operações semelhantes no sentido horário do outro lado.

Seguindo o mesmo raciocínio, podemos definir outras funções de arestas:

$$\begin{aligned}(e,r,f)\text{Sym}&=(e,r+2,f), \\ (e,r,f)\text{Rot}^{-1}&=(e,r+3+2f,f), \\ (e,r,f)\text{Oprev}&=[(e.[r+1-f]).\text{Next})\text{Rot}^{(1-f)}\text{Flip}^f\end{aligned}$$

9.2 Operadores Topológicos Básicos

A *quad-edge* contém apenas dois operadores topológicos básicos: $e \rightarrow \text{MakeEdge}()$ e $\text{Splice}(a,b)$. Qualquer diagrama pode ser construído e/ou modificado através deste dois operadores.

9.2.1 Operador MakeEdge()

O operador $\text{MakeEdge}()$ cria uma estrutura de dados representando uma subdivisão de uma esfera e retorna a sua aresta canônica (Figura 9.2). Se não considerarmos a orientação e a direção, podemos verificar que e é a única aresta existente nesta subdivisão e não forma um *loop*. Nesta configuração temos as seguintes propriedades $e.\text{Org} \neq e.\text{Dest}$, $e.\text{Left} = e.\text{Right}$, $e.\text{Lnext} = e.\text{Rnext} = e.\text{Sym}$ e $e.\text{Onext} = e.\text{Oprev}$. Caso desejarmos obter um *loop*, basta aplicarmos a função Rot sobre o resultado do operador $\text{MakeEdge}()$, isto é, $e \leftarrow \text{MakeEdge}().\text{Rot}$.

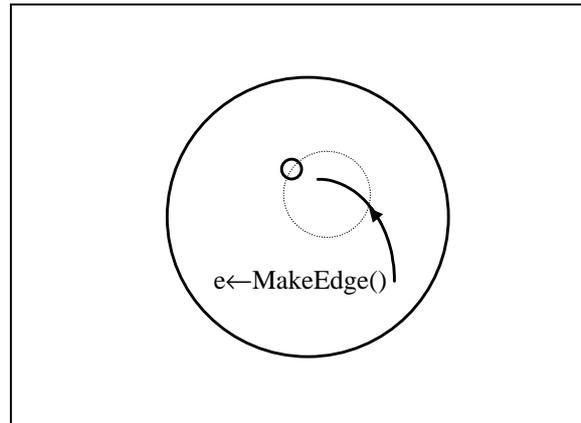


Figura 9.2 - Resultado do operador MakeEdge()

Neste caso teremos a seguinte configuração: $e.Org=e.Dest$, $e.Left \neq e.Right$, $e.Lnext=e.Rnext=e$, e $e.Onext=e.Oprev=e.Sym$.

9.2.2 Operador Splice

O operador $Splice(a,b)$ recebe como parâmetros duas arestas. Ele afeta os anéis $a.Org$ e $b.Org$ e, independentemente, os anéis $a.Left$ e $b.Left$.

Dependendo do caso, este operador tem os seguintes efeitos:

- (a) se os dois anéis são distintos, *splice* os unirá em um único anel.
- (b) se os dois anéis são exatamente os mesmos, então *splice* os separará em duas partes distintas (Figura 9.3).
- (c) se os dois anéis são os mesmos porém possuem orientações opostas, *splice* trocará a orientação (*flip*) e inverterá a ordem de um dos segmentos do anel.

Abaixo temos a descrição do operador *splice* através da álgebra de arestas:

- $a.Onext=b.Onext$
- $b.Onext=a.Onext$
- $a.Onext.Rot.Onext=b.Onext.Rot.Onext$
- $b.Onext.Rot.Onext=a.Onext.Rot.Onext$

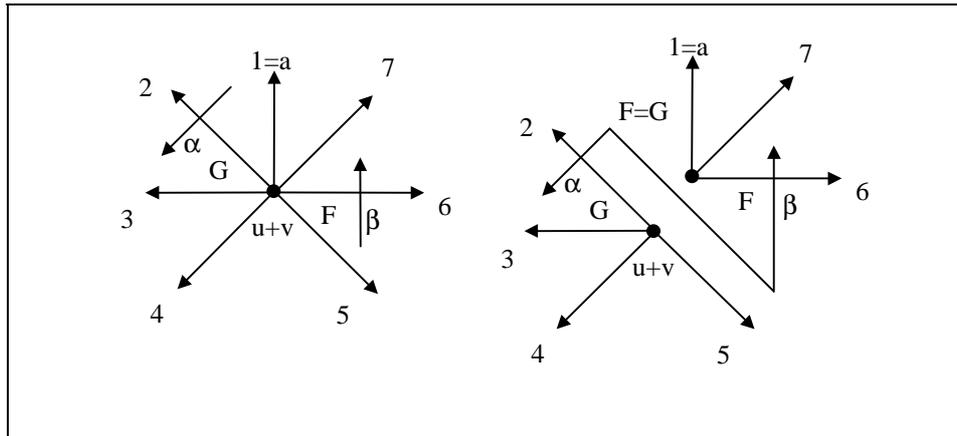


Figura 9.3 - Resultado do operador Splice(a,b)

9.3 Algoritmo de inserção gulosa utilizando a estrutura quad-edge

Abaixo descrevemos o algoritmo de inserção de um ponto em uma triangulação de delaunay utilizando a estrutura topológica quadedge.

9.3.1 Operadores básicos

```
PROCEDURE Connect[a, b, side] RETURNS [e]
    e ← MakeEdge [ ];
    e.Org ← a.Dest;
    e.Dest ← b.Org;
    Splice [e, a.Lnext];
    Splice[e.Sym, b]
END Connect.
```

```
PROCEDURE DeleteEdge[e]:
    Splice[e, e.Oprev];
    Splice[e, e.Sym, e.Sym.Oprev]
END DeleteEdge.
```

```
PROCEDURE Swap[e]
    a ← e.Oprev;
    b ← e.Sym.Oprev;
    Splice[e, a]; Splice [e.Sym, b];
    Splice[e, a.Lnext]; Splice [e.Sym, b.Lnext];
    e.Org ← a.Dest; e.Dest ← b.Dest
END DeleteEdge.
```

9.3.2 Operadores geométricos

```
PROCEDURE RightOf[X,e]:
    RETURN CCW[X, e.Dest, e.Org]
END RightOf.
```

```
PROCEDURE LeftOf[X,e]:
    RETURN CCW[X, e.Org, e.Dest]
END LeftOf.
```

O predicado $CCW(A,B,C)$ (testa se os pontos A,B e C estão orientados em sentido anti-horário) pode ser implementado como o teste:

$$CCW(A, B, C) = \begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix} > 0$$

O predicado $InCircle(A,B,C,D)$ (testa se o ponto D é interior ao círculo circunscrito ao triângulo definido por A,B e C) pode ser implementado pela seguinte expressão [GS85]:

$$InCircle(A, B, C, D) = \begin{vmatrix} X_A & Y_A & X_A^2 + Y_A^2 & 1 \\ X_B & Y_B & X_B^2 + Y_B^2 & 1 \\ X_C & Y_C & X_C^2 + Y_C^2 & 1 \\ X_D & Y_D & X_D^2 + Y_D^2 & 1 \end{vmatrix} > 0$$

9.3.3 Algoritmo de inserção de ponto em um triangulação de delaunay.

```
PROCEDURE InsetSite[X]:
  e ← Locate[X];
  IF X = e.Org OR X=e.Dest THEN {Ignore it;} RETURN
  ELSEIF X is on e THEN t ← e.Oprev; DeleteEdge{e}; e ← t FI;
  {Connect S to vertices around it.}
  base ← MakeEdge[ ];
  first ← e.Org; base.Org ← first; base. Dest ← X;
  Splice[base,e];
  REPEAT
    base ← Connect[ e, base.Sym]; e ← base. Oprev
  UNTIL e. Dest = first;
  e ← base.Oprev;
  {The suspect edges(from top to bottom) are e(.Onext.Lprev)k for k=0,1,...}
  {The bottom edge has .Org = first}
  DO
    t ← e.Oprev;
    IF RightOf[t.Dest,e] AND InCircle[e.Org, t.Dest, e.Dest, X]
    THEN Swap[e]; e ← t
    ELSEIF e.Org = first THEN {No more supect edges.} RETURN
    ELSE {Pop a suspect edge:} e ← e.Onext.Lprev FI
  OD
END InsetSite.
```

9.3.4 Algoritmo de localização de um ponto na triângulação

```
PROCEDURE Locate[X] RETURNS [e]:
  e ← some edge;
  DO
    IF X = e.Org OR X=e.Dest THEN RETURN e
    ELSEIF RightOf [X, e] THEN e ← e.Sym
    ELSEIF NOT RightOf[X, e.Onext] THEN e ← e.Onext
    ELSEIF NOT RightOf[X, e.Dprev] THEN e ← e.Dprev
    ELSE RETURN e FI
  OD
END locate.
```

Referências

- [BA75] B. Baumgart, A polyhedron representation for computer vision, AFIPS Conf. Proc, 589-596(1975).
- [CS95] Cláudio T. Silva, Joseph S. B. Mitchell and Arie E. Kaufman, Automatic Generation of Triangular Irregular Networks using Greedy Cuts(1995).
- [FC91] Luiz Henrique Figueiredo e Paulo Cezar Pinto Carvalho, Introdução a Geometria Computacional. 18º Colóquio Brasileiro de Matemática(1991).
- [FO90] Foley et al., Computer Graphics , Principles and Praticce. Addison-Wesley Publishing Company, 1990.
- [FS97] F. Szenberg, M. Gattass, P. C. P. Carvalho, An Algorithm for the visualization of a Terrain with Objects. Submetido ao SIBGRAPI 97.
- [GB97] L. Guibas, Geometric Algorithms, Stanford University,1997.
- [GS85] L.J. Guibas e J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Trans. Graphics 4, 141-155(1985).
- [GU97] L.C.C. Guedes, Real-time surface extraction at variable resolution. Submetido ao SIBGRAPI 97.
- [HG95] Paul S. Heckbert and Michael Garland, Fast Polygonal Approximation of Terrains and Height Fields(1995).Technical Report.
- [JG95] J. G. O. Militão, Uma metodologia para posicionamento de prédios em terrenos acidentados. Dissertação de Mestrado. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro, 1995.
- [JL89] Jay Lee. A drop heuristic conversion method for extracting irregular network for digital elevation models. In GIS/LIS '89 Proc., volume 1, pages 30-39. American Congress on Surveying and Mapping, Nov. 1989.
- [LW83] Lance Williams. Pyramidal parametrics. Computer Graphics(SIGGRAPH '83 Proc.), 17(3):1-11,July.
- [MA88] M. Mäntylä, An Introduction to Solid Modeling, Comp. Science Press, Rockville(1988).
- [MV95] Jonas Miranda, Luiz Velho, Computação Gráfica - Imagens.

[PH91] Peter Hughes. Building a terrain renderer. Computers in Physics, pages 434-437, July/ August 1991.

[PS85] Franco P. Preparata, Michael Ian Shamos, Computational Geometry - An Introduction, Springer-Verlag, 1985.

[PU91] Enrico Puppo, Larry Davis, Daniel DeMenthon, and Y., Ansel Teng. Parallel terrain triangulation using the Connection Machine. Technical Report CAR-TR-561, CS-TR-2693, Center for Automation Research, University of Maryland , College Park, Maryland, June 1991.

[RH83] Richard W. Hamming. Digital Filters. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[SP79] Michael Spivak, A comprehensive introduction to differential geometry. Publish or Perish, Inc., 1979.

[SR94] Florian Schröder and Patrick Roßbach, Managing the complexity of digital terrain models, Comput. & Graphics, Vol. 18, No.6, pp. 775-783, 1994.

[TP75] Steven L. Tanimoto and Theo Pavlidis. A hierarchical data structure for picture processing. Computer Graphics and Image Processing, 4(2):104-119, June 1975.

[UR72] Urs Ramer. An interactive procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing 1:244-256, 1972.

Investigação de novos critérios para inserção de pontos em métodos de simplificação de modelos de terreno através de refinamento

Dissertação de Mestrado apresentada por ANSELMO ANTUNES MONTENEGRO em 27 de agosto de 1997 ao Departamento de Informática da PUC-Rio e aprovada pela comissão julgadora formada por:

**Prof. Paulo Cezar Pinto Carvalho(IMPA)
Orientador**

**Prof. Marcelo Gattass(DI/PUC-Rio)
Co-orientador**

Prof. Luiz Carlos Castro Guedes(DCC/UFF)

Prof. Marcelo de Andrade Dreux(MEC/PUC-Rio)

**Visto e permitida a impressão.
Rio de Janeiro,**

**Coordenador dos Programas de Pós-Graduação do
Centro Técnico Científico da PUC-Rio**