

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Renan Augusto Dembogurski

**Geração Interativa de Malhas em Multiresolução
através de Marcadores em Realidade Aumentada**

Juiz de Fora

2013

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Renan Augusto Dembogurski

Geração Interativa de Malhas em Multiresolução através de Marcadores em Realidade Aumentada

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Rodrigo Luis de Souza da
Silva

Coorientador: Marcelo Bernardes Vieira

Juiz de Fora

2013

Ficha catalográfica elaborada através do Programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Dembogurski, Renan Augusto.

Geração Interativa de Malhas em Multiresolução através de Marcadores em Realidade Aumentada / Renan Augusto

Dembogurski. -- 2013.

63 f. : il.

Orientador: Rodrigo Luis de Souza da Silva

Coorientador: Marcelo Bernardes Vieira

Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2013.

1. Deformação de malhas. 2. Malhas Semi-Regulares Adaptativas 4-8. 3. Parametrização. 4. Realidade Aumentada. I. de Souza da Silva, Rodrigo Luis, orient. II. Bernardes Vieira, Marcelo, coorient. III. Título.

Renan Augusto Dembogurski

Geração Interativa de Malhas em Multiresolução através de Marcadores em Realidade Aumentada

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 06 de Março de 2013.

BANCA EXAMINADORA

Prof. D.Sc. Rodrigo Luis de Souza da Silva - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Marcelo Bernardes Vieira
Universidade Federal de Juiz de Fora

Prof. D.Sc. Leandro Schaeffer Marturelli
Universidade Federal do Rio de Janeiro

Prof. D.Sc. Eduardo Barrére
Universidade Federal de Juiz de Fora

A minha mãe e meu irmão.

A minha namorada, Lidya.

AGRADECIMENTOS

Agradeço a todas as pessoas que me ajudaram nessa caminhada árdua. De forma mais específica, agradeço a minha mãe por ser uma ouvinte paciente ao longo de incontáveis horas e ao meu irmão por sempre me apoiar e ajudar nos trabalhos desenvolvidos. Agradeço a mulher da minha vida, Lidya, a pessoa mais especial que já conheci e sem a qual não estaria escrevendo esse agradecimento.

Gostaria de agradecer aqui também meus amigos, Carlão, Viviz, Tadeu, Zezão, Dhiego, Barra entre outros. Agradeço porque vocês contribuíram diretamente para o meu sucesso, seja com palavras de amizade ou de trollagem.

Agradeço a CAPES pela bolsa que me permitiu estudar em regime integral no mestrado. Agradeço aos professores que me ajudaram em meu caminho, cada um do seu jeito.

*“Ver o bem e não fazê-lo é sinal
de covardia.”
Confúcio*

RESUMO

Este trabalho apresenta um método que permite a deformação de um terreno pela modificação do seu mapa de alturas em um ambiente de realidade aumentada. A estrutura hierárquica de malhas A4-8 foi utilizada para representar o terreno. Essa estrutura define um espaço paramétrico para calcular as coordenadas de um terreno no espaço Euclidiano \mathbb{R}^3 . Em especial, este trabalho lida com o problema de modelagem de terrenos esféricos. Uma métrica de erro dependente do observador e da geometria do terreno utilizada tanto para sua observação quanto para sua modelagem. Os resultados demonstram que o uso da malha A4-8 em conjunto com o sistema de realidade aumentada tangível é flexível para modelar terrenos esféricos e pode ser facilmente modificada para terrenos com a topologia do cilindro e do toro. O desenvolvimento de um método de geração de malhas eficiente e de uso intuitivo, baseado em marcadores de realidade aumentada, é a principal contribuição deste trabalho.

Palavras-chave: Deformação de malhas. Malhas Semi-Regulares Adaptativas 4-8. Parametrização. Realidade Aumentada.

ABSTRACT

This work presents a method that allows the deformation of a terrain by modifying your heightmap in an augmented reality environment. The hierarchical structure of A4-8 meshes was used to represent terrains. This structure defines a parameter space to calculate the coordinates of a field in the \mathbb{R}^3 Euclidean space. In particular, this paper deals with the problem of modeling spherical terrains. An error metric dependent on the observer and the geometry of the land used for its observation and modeling. The results demonstrate that the use of A4-8 mesh combined with the tangible augmented reality system is flexible to shape spherical terrains and can be easily modified to deal with other topologies, such as the torus and the cylinder. The development of an efficient and intuitive to use method for mesh generation, based on augmented reality markers, is the main contribution of this work.

Keywords: Mesh Deformation. 4-8 Semi-regular Adaptive Meshes. Parametrization. Augmented Reality.

LISTA DE FIGURAS

1.1	Um modelo 3D criado pela interface <i>inTouch</i> (GREGORY et al., 1999).	18
1.2	Exemplo da interface 2D <i>Teddy</i> (IGARASHI et al., 2007).	19
1.3	Seleção de parte de um modelo no sistema <i>3DARModeler</i> (DO; LEE, 2010).	20
1.4	Visualização de parte da superfície, diferentes cores representam as características geológicas de cada região (FAETH; HARDING, 2009).	21
1.5	Deformação interativa de uma submalha (FAETH; HARDING, 2009).	21
2.1	Estrutura de uma malha 4-8, em (a) um bloco básico destacado em roxo e em (b) dois passos de refinamento, o primeiro gera os vértices em azul e o segundo, os vértices em vermelho	25
2.2	(a) Protótipo de subdivisão quaternária e (b) protótipo de subdivisão binária.	25
2.3	Refinamento e simplificação da malha A4-8.	26
2.4	Níveis 0–5 de uma <i>bintree</i> triangular.	28
2.5	Refinamento e simplificação do diamante.	28
2.6	Divisão forçada de um triângulo.	29
2.7	Cilindro sendo conformado em uma esfera. No meridiano da esfera as coordenadas das duas bordas do plano são conectadas (NOEL, 2010).	29
2.8	<i>Wedgies</i> aninhados em 1D, com os dependentes de v (DUCHANEAU et al., 1997).	35
3.1	Visão geral do método.	38
3.2	(a) Vértice de controle \vec{v}_c e (b) a atualização da sua posição devido ao refinamento da malha. A esfera azul representa o vértice de controle \vec{v}_c e a esfera vermelha é a posição do marcador móvel.	39
3.3	Início do processo de expansão. O vértice de controle é \vec{v}_c no domínio discreto e \vec{v}_c no domínio contínuo. O raio que define a área de deformação é r_d e \vec{d}_{ic} é a distância em \mathbb{R}^3 entre o vértice de controle \vec{v}_c e um de seus vizinhos \vec{v}_i	42
3.4	O conjunto \mathbf{V}_{d_c} , em verde, representa os vértices que possuem a propriedade $I(u, v)$. No exemplo, todos os vizinhos estão dentro da área de deformação.	42

3.5	Exemplo de expansão na direção diagonal (a) e na direção vertical/horizontal (b). Um passo do algoritmo de expansão após todos os vizinhos de \vec{v}_c serem computados pode ser visto em (c).	43
3.6	Tratamento de bordas no caso de vértices se expandindo além das dimensões do terreno. Para a latitude, se um vértice \vec{v} extrapola as dimensões do terreno, a expansão é interrompida naquela direção. No caso da longitude, a coordenada u do vértice é modificada para que ele possa continuar expandindo.	44
3.7	Exemplo de propagação de erro correta. No triângulo da esquerda (a) valores de altura para os vértices do triângulo T e no triângulo da direita (b) os respectivos valores de erro.	46
3.8	Exemplo de expansão hierárquica. Em (a) ocorre o primeiro caso, passo na vertical e caminhada diagonal. Em (b) acontece o segundo caso, passo diagonal e caminhada vertical, ambos na iteração de valor zero. Já em (c) repete-se o primeiro caso e em (d) o segundo, agora para iteração de valor um.	47
4.1	Visualização de um terreno esférico. Janela de visão local (a) e global (b).	50
4.2	Mudanças na malha provocadas pela distância entre câmera e terreno. Posição inicial do terreno em (b), câmera aproximada do terreno (c) e afastada (a).	51
4.3	Uma sequência de quadros espaçados no tempo que demonstram a malha sendo deformada utilizando o método proposto.	52
4.4	Resultado da tesselação para valores diferentes de erro máximo. Em (a) o valor de erro utilizado foi $dist_{max} = 1.0$ e em (b) $dist_{max} = 6.0$.	52
4.5	Visualização de diferentes áreas de deformação. O valor do raio de deformação utilizado em (a) foi $r_d = 150\text{km}$, em (b) $r_d = 300\text{km}$ e em (c) $r_d = 600\text{km}$.	53
4.6	Deformação de terrenos utilizando a função de distribuição Gaussiana. A deformação obtida para um valor de $\sigma = 0.17$ e média $\mu = 1$ pode ser visto em (a). Em (b), $\sigma = 0.75$ e $\mu = 1$.	56
4.7	Tratamento de bordas. Em (a) é possível ver o terreno deformado na região da borda sem o tratamento. Em (b) a mesma região do terreno, agora deformada com tratamento de borda.	56

4.8	Visualização de um terreno final gerado através do método proposto. Escala local em (a) e global em (b).	56
-----	--	----

LISTA DE TABELAS

4.1	Tempo médio de renderização do terreno em segundos quando uma quantidade de vértices é modificada em uma determinada latitude. Resolução máxima de 2049×2049 vértices.	54
4.2	Tempo médio de renderização do terreno em segundos quando uma quantidade de vértices é modificada em uma determinada latitude. Resolução máxima de 4097×4097 vértices.	54
4.3	Tempo médio de renderização do terreno em segundos quando uma quantidade de vértices são modificados em uma determinada latitude. Resolução máxima de 8193×8193 vértices.	55

LISTA DE SÍMBOLOS

- \vec{v} Ponto ou coordenada discreta $\vec{v} = (u, v)$, com $u, v \in \mathbb{N}$.
- $\vec{\mathbf{v}}$ Ponto ou coordenada contínua $\vec{\mathbf{v}} = (x, y, z)$, com $x, y, z \in \mathbb{R}$.
- \vec{v}_c Vértice de controle no domínio discreto.
- $\vec{\mathbf{v}}_c$ Vértice de controle no domínio contínuo.
- $\vec{\mathbf{d}}_{ic}$ Distância Euclidiana em \mathbb{R}^3 entre um vetor $\vec{\mathbf{v}}_i$ e o vértice de controle $\vec{\mathbf{v}}_c$.
- $f(u, v)$ Função de mapeamento que realiza a passagem do domínio discreto para o contínuo.
- $I(u, v)$ Função que indica se um vértice $\vec{v} = (u, v)$, ao ser mapeado no domínio contínuo, está a uma determinada distância do vértice de controle $\vec{\mathbf{v}}_c$.
- $dist(\vec{\mathbf{v}})$ Erro dependente do observador. É a distância entre a posição correta no espaço de tela de um vértice $\vec{\mathbf{v}}$ e sua aproximação pela triangulação T.

SUMÁRIO

1	INTRODUÇÃO	15
1.1	DEFINIÇÃO DO PROBLEMA E OBJETIVOS	16
1.2	OBJETIVOS	16
1.3	TRABALHOS RELACIONADOS	16
1.3.1	Deformação de Malhas em Realidade Aumentada	17
1.3.2	Interfaces de Usuário em 3D	18
2	FUNDAMENTOS	23
2.1	MALHAS REGULARES ADAPTATIVAS 4-8	23
2.1.1	Construção da Estrutura Hierárquica	24
2.1.1.1	Abordagens para Construção da Malha A4-8	26
2.1.2	Representação da Estrutura Hierárquica	27
2.1.2.1	<i>Bintree</i> Triangular	27
2.2	PARAMETRIZAÇÃO	30
2.2.1	Partição do Espaço de Parâmetros	32
2.3	MÉTRICA DE ERRO	33
2.3.1	Erro Geométrico	34
2.3.2	Erro Dependente do Observador	35
3	MÉTODO PROPOSTO	37
3.1	VISÃO GERAL DO MÉTODO	37
3.2	DEFINIÇÃO DO PONTO DE CONTROLE	37
3.3	DEFINIÇÃO DA ÁREA DE DEFORMAÇÃO	39
3.3.1	Crescimento de Região	40
3.3.1.1	Visão Geral do Algoritmo	41
3.3.1.2	Função Recursiva de Expansão	42
3.3.2	Tratamento de Borda	44
3.4	PROPAGAÇÃO DE ALTURAS E ERROS	45
3.4.1	Cálculo e Propagação de Alturas	45
3.4.2	Métrica e Propagação de Erro	46

4	RESULTADOS	50
4.1	VISUALIZAÇÃO DO TERRENO ESFÉRICO	50
4.2	EXPERIMENTOS REALIZADOS	51
4.3	PARTICULARIDADES	55
5	CONCLUSÃO	58
	REFERÊNCIAS	60

1 INTRODUÇÃO

A deformação de malhas representa um recurso importante para a área de modelagem de objetos por permitir a modificação de uma superfície para atender a um determinado fim. Devido à sua importância, várias técnicas de deformação foram desenvolvidas ao longo do tempo, como visto em (BARR, 1984; SEDERBERG; PARRY, 1986; COQUILLART, 1990; SINGH; FIUME, 1998; SUMNER et al., 2007). Algumas destas técnicas levam em consideração a decomposição de superfícies muito detalhadas em níveis hierárquicos (KOBELT et al., 2000; SHA et al., 2007). A representação de superfícies em múltiplos níveis de detalhamento permite que alterações sejam feitas em qualquer um desses níveis, resultando em um controle fino da malha.

Uma estrutura hierárquica que possui propriedades favoráveis para a representação e modelagem de terrenos em vários níveis de detalhe é a malha semi-regular adaptativa 4-8 (malha A4-8). Através dessa estrutura, é possível representar terrenos com diferentes resoluções, permitindo controle pontual sobre a malha gerada. A A4-8 também pode definir um espaço paramétrico para calcular as coordenadas de um terreno no espaço Euclidiano \mathbb{R}^3 .

Modificar superfícies em vários níveis de detalhe é de interesse para área de Interfaces de Usuários em 3D (GREGORY et al., 1999; SANTHANAM et al., 2004; IGARASHI et al., 2007), porque muitos aplicativos desenvolvidos nesta área permitem um alto nível de interação entre usuários e modelos virtuais. No caso de aplicativos que envolvam modelagem de objetos, é possível desenvolver interfaces 3D nas quais o usuário consiga interagir com modelos 3D, como também controlar e modificar de maneira pontual a superfície de um modelo através de estruturas hierárquicas.

Um dos possíveis meios para interação com modelos em 3D é a realidade aumentada. Nela, elementos reais e virtuais são combinados para criar a impressão de que coexistem em uma mesma cena. Tal efeito pode ser alcançado através de diversas tecnologias como sistemas de projeção ótica, monitores e dispositivos portáteis. Para interagir com os modelos virtuais foi desenvolvida uma interface chamada *Tangible Augmented Reality (TAR)* (BILLINGHURST HIROKAZU KATO, 2001), onde cada objeto virtual é associado a um objeto físico e o usuário interage com os objetos virtuais manipulando os objetos tangíveis.

Esta dissertação propõe um método para deformar terrenos esféricos através da combinação da estrutura da A4-8 com marcadores fiduciais. A ideia é visualizar uma malha associada a um marcador físico e modificá-la utilizando outro marcador físico móvel. A interface 3D final permite que um usuário rotacione e translate livremente um modelo 3D utilizando suas mãos, permitindo também que o usuário modifique sua superfície em vários níveis de detalhe. O método proposto permite um alto nível de interação com o modelo. O usuário interage com o modelo movendo manualmente o marcador físico móvel ao longo da superfície a ser alterada. O método proposto pode ser utilizado na construção de ambientes de jogos.

1.1 DEFINIÇÃO DO PROBLEMA E OBJETIVOS

O problema tratado nesta dissertação é como deformar terrenos que possuem a topologia da esfera (terrenos esféricos) em múltiplos níveis de detalhamento através de interação por realidade aumentada.

1.2 OBJETIVOS

O principal objetivo desta dissertação é apresentar um método interativo para deformar terrenos em tempo real.

Como objetivos secundários temos:

- Demonstrar, através de uma aplicação do método proposto, que o desempenho de um aplicativo que utiliza a metodologia proposta pode ser mantido em diferentes níveis de resolução do terreno.
- Apresentar uma interface 3D para deformar terrenos em tempo real. No contexto de Realidade Aumentada, tempo real é definido como 25-30 quadros por segundo.

1.3 TRABALHOS RELACIONADOS

Serão apresentados nesta seção diferentes abordagens para deformar malhas através de realidade aumentada. Todas as abordagens relacionadas utilizam interfaces de usuário em 3D.

A seção 1.3.1 apresenta métodos para modelar objetos 3D em multiresolução através de realidade aumentada. Na seção 1.3.2 são apresentados diferentes tipos de interfaces de usuário 3D, onde há interação através de realidade aumentada.

1.3.1 DEFORMAÇÃO DE MALHAS EM REALIDADE AUMENTADA

Não existem muitos trabalhos que lidam com o problema da deformação de malhas através de realidade aumentada. O foco da maioria dos estudos envolvendo modelagem de objetos e realidade aumentada está nos modelos de deformação em geral, ou no desenvolvimento de aplicativos para áreas como a de simulação de operações cirúrgicas (NEALEN et al., 2006), modelagem de roupas (BARAFF; WITKIN, 1998; NEALEN et al., 2006) e modelagem de acidentes (O'BRIEN; HODGINS, 1999).

Um trabalho que permite a modificação interativa de malhas poligonais em multiresolução através de uma interface 3D pode ser encontrado em (GREGORY et al., 1999). Os autores propõem uma interface chamada *inTouch*, que contém uma cena 3D para visualização do modelo e um menu 2D para realização de operações pelo usuário. O modelo é editado na cena através de um dispositivo háptico.

A interface possui um subsistema de edição de malhas em multiresolução, que utiliza como entrada a posição da sonda do dispositivo háptico projetada na cena e a direção da força aplicada. A medida que o usuário movimentar a sonda, uma modificação na superfície do modelo ocorre de acordo com a operação selecionada. A estrutura hierárquica escolhida para representar o modelo foi a malha poligonal de subdivisão.

O subsistema de edição de malhas recebe um triângulo, um ponto de contato e um vetor de movimento como entrada. Quando a malha é modificada em uma determinada resolução, um conjunto de vértices é movido. A modificação é propagada para os níveis de maior e menor detalhe, através de subdivisões na malha e suavização respectivamente.

Um exemplo de modelo criado pela interface *inTouch* pode ser visto na Figura 1.1. No modelo contido nessa dissertação, o ponto de controle para deformação é um vértice da malha e a deformação é propagada apenas para os níveis de resolução maiores que o desse ponto.

A visualização do modelo através da interface *inTouch* sempre mostra o nível mais refinado da malha. Esse fato pode tornar a interface inviável para deformações que utilizem um número elevado de triângulos em tempo real. Um fator importante a ser mencionado



Figura 1.1: Um modelo 3D criado pela interface *inTouch* (GREGORY et al., 1999).

é que, em regiões distantes do observador, um nível de resolução elevado pode ser desnecessário na visualização do objeto. Nesta dissertação o nível de detalhe depende não só da geometria do objeto, mas também da distância em relação ao observador.

Outro trabalho que segue a mesma ideia da interface *inTouch* pode ser visto em (FOSKEY et al., 2005). Os autores apresentam um sistema chamado *ArtNova* para modelagem de objetos 3D utilizando interface háptica. Diferente da interface citada anteriormente, o sistema *ArtNova* possui diversas técnicas de visualização dinâmica, como navegação baseada na visão e reposicionamento automático.

É importante mencionar que nos trabalhos citados o objeto a ser deformado não consiste de milhares de triângulos como as malhas que adaptam terrenos reais. Não foi encontrado nenhum trabalho na literatura que lidasse diretamente com a interação entre usuário e modelos 3D compostos de milhares de triângulos em multiresolução através de realidade aumentada tangível.

1.3.2 INTERFACES DE USUÁRIO EM 3D

A área que estuda as interfaces de usuário que apresentam interação 3D é chamada de interfaces de usuário em 3D (BOWMAN et al., 2004). Esta área busca estudar as melhores abordagens para criar interfaces que proporcionem uma experiência diferenciada para os usuários. Para alcançar este objetivo, as interfaces desenvolvidas apresentam diversas características distintas.

Dentre as características que tornam as interfaces de usuário em 3D diferenciadas, é possível destacar a promoção de imersão em ambientes virtuais, resolução de problemas

relacionados a tarefas importantes como navegação e manipulação e a capacidade de serem imperceptíveis quando em uso. Um *survey* que relaciona em detalhes as categorias e objetivos desta área pode ser encontrado em (BOWMAN; FROHLICH, 2005; BOWMAN et al., 2008)

As interfaces em 3D que lidam com a manipulação de objetos virtuais através de realidade aumentada são de interesse para esta dissertação. Em específico, é possível citar o trabalho desenvolvido em (IGARASHI et al., 2007), onde o usuário faz esboços de forma livre em uma interface 2D chamada *Teddy* que depois são utilizados para gerar uma malha poligonal 3D plausível (Fig. 1.2).



Figura 1.2: Exemplo da interface 2D *Teddy* (IGARASHI et al., 2007).

Um trabalho que utiliza realidade aumentada tangível como interface 3D para modelar objetos 3D pode ser encontrado em (DO; LEE, 2010), onde os autores apresentam um sistema de modelagem 3D em um ambiente de realidade aumentada chamado *3DARModeler*. Com este sistema é possível criar um modelo 3D através de uma ou várias geometrias primitivas, aplicar texturas, adicionar animações, estimar fontes de luz real e projetar sombras no mesmo.

A diferença entre a interface deste sistema e a que foi desenvolvida nesta dissertação relaciona-se a modelagem de objetos. No sistema *3DARModeler* os objetos são modelados pela combinação de geometrias primitivas, sendo estas estáticas. Em outras palavras, a malha não pode ser deformada de acordo com a vontade do usuário. Já no caso desta dissertação, a proposta é mudar dinamicamente a geometria de uma superfície em múltiplos níveis de detalhe, provendo controle preciso sobre a malha e maior interação entre

usuário e modelo.

Um outro fator que distingue essas interfaces é o método de seleção dos objetos de interesse. No sistema *3DARModeler*, uma parte do objeto é selecionada apontando-se um marcador físico para um ponto de controle representado por um objeto virtual, que indica se o modelo está sendo apontado (Fig. 1.3). Como cada marcador físico possui um sistema próprio de coordenadas, apontar um marcador móvel em direção a outro, para selecioná-lo, pode ser uma tarefa difícil e imprecisa.

A seleção de parte do objeto, utilizando o método proposto nesta dissertação, é mais intuitiva. A posição de um marcador fiducial é projetada na superfície de um objeto e o ponto mais próximo dessa posição, pertencente à malha, é escolhido como ponto de controle. É possível, também, modificar a posição do marcador fiducial movendo-o ao longo da superfície, modificando os pontos de controle resultantes.

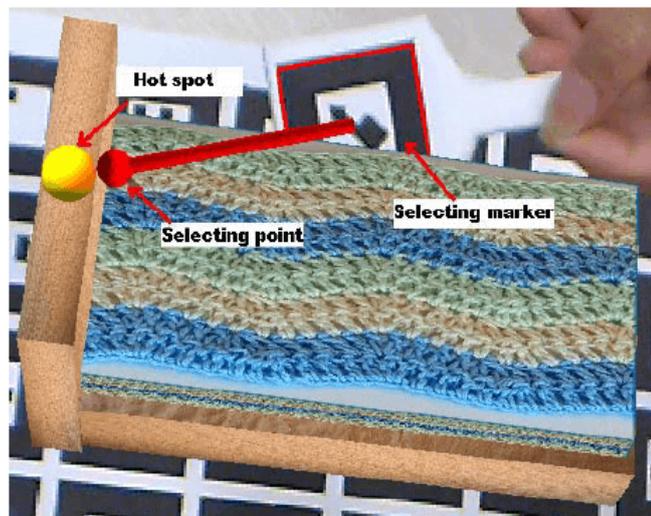


Figura 1.3: Seleção de parte de um modelo no sistema *3DARModeler* (DO; LEE, 2010).

Um trabalho que utiliza dispositivos de resposta de força como interface para modelar objetos 3D pode ser visto em (FAETH; HARDING, 2009), onde os autores apresentam um método que permite a um geocientista interagir com superfícies pintando-as de acordo com sua maleabilidade.

Em um passo inicial, o geocientista classifica as regiões da superfície de acordo com suas características geológicas. Três parâmetros de deformação, alongamento, compressão e rasgo são mapeados nas três cores primárias de uma sistema de cores *RGB*, para que seja possível definir para cada área uma característica. Os autores utilizam um mapa de calor baseado na cor que o ferro adquire ao ser aquecido. Uma cor mais escura repre-

sentam um material mais rígido, enquanto cores próximas ao vermelho, laranja ou branco, representam materiais mais maleáveis (Fig. 1.4).

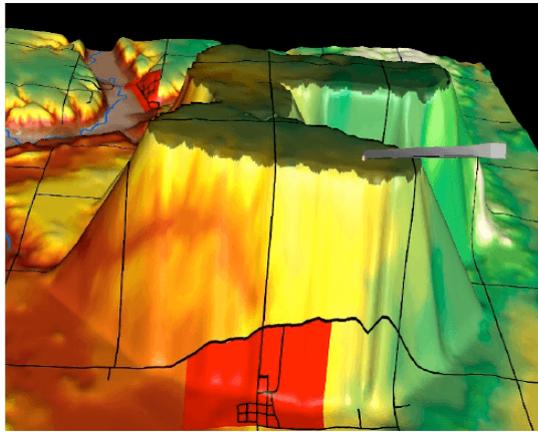
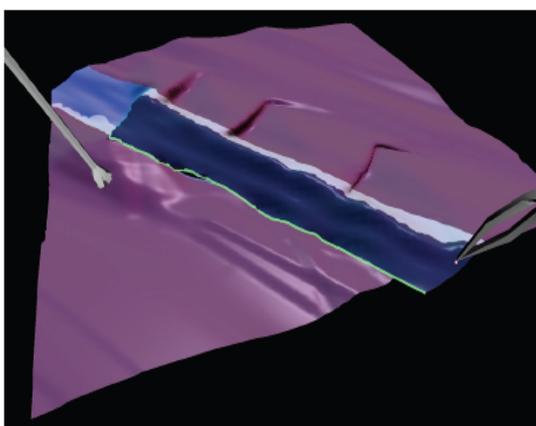


Figura 1.4: Visualização de parte da superfície, diferentes cores representam as características geológicas de cada região (FAETH; HARDING, 2009).

O geocientista utiliza um dispositivo háptico chamado *Phantom* para deformar a superfície do modelo 3D. Utilizando a parte móvel do dispositivo, chamada *stylus*, o usuário determina o local onde será feita a deformação. Uma pinça virtual serve como guia à medida que o usuário movimenta a *stylus*. Determinado este local, o usuário pressiona um botão na parte móvel do dispositivo e continua movimentando-a, agora para determinar o deslocamento que será convertido em deformação através de um algoritmo proposto pelos autores. A Figura 1.5a mostra um momento na seleção de região de deformação e a Figura 1.5b mostra o dispositivo háptico.



(a)



(b)

Figura 1.5: Deformação interativa de uma submalha (FAETH; HARDING, 2009).

Muitos trabalhos na bibliografia utilizam aparelhos específicos para interagir com os modelos, o que às vezes inviabiliza sua reprodução, ou cria a necessidade de treino prévio

do usuário. A interface desenvolvida nesta dissertação, ao contrário, necessita apenas de um marcador móvel e um dispositivo de captura de vídeo para que o usuário consiga interagir com um modelo 3D intuitivamente em tempo real.

2 FUNDAMENTOS

2.1 MALHAS REGULARES ADAPTATIVAS 4-8

Uma malha poligonal é definida pelo par $M := (V, F)$, onde V é o conjunto de vértices $V := \{\vec{v}_i := (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1, \dots, n\}$ e F é um complexo simplicial abstrato que contém informação de conectividade como, por exemplo, a topologia. Em particular, se considerarmos uma *malha triangular* cada elemento no complexo F se enquadra em algum desses tipos: vértice $\{i\}$, aresta $\{i, j\}$ e face $\{i, j, k\}$ ((PATANÈ; SPAGNUOLO, 2003)).

Para criar um modelo virtual de um terreno real, é necessário considerar a informação que descreve os elementos da malha, ou seja, a conectividade e a geometria da mesma. A conectividade da malha, ou topologia, descreve a relação de incidência entre os elementos da malha. A geometria da malha especifica a posição e outras características de cada vértice.

Tipicamente é possível distinguir a conectividade de uma malha entre *irregular* e (*semi*-)*regular*. As malhas irregulares não possuem restrição na valência (número de arestas incidentes) de um vértice, enquanto as malhas regulares são formadas por um domínio base irregular que sofre refinamento regular recursivo.

A conectividade da malha representa, assim, um fator de vital importância no desenvolvimento de aplicativos que realizem a modelagem de objetos e geralmente guia a escolha dos algoritmos a serem utilizados. Neste trabalho foram utilizadas malhas semi-regulares devido à sua estrutura paramétrica previsível e conectividade mais compreensível que no caso das malhas irregulares.

Uma malha pode ser definida como regular, ou de topologia regular, se todas as faces e vértices possuem as mesmas propriedades combinatórias como, por exemplo, mesmo tamanho e mesma valência (AKLEMAN; CHEN, 2005). No caso das malhas semi-regulares, todas as faces ou todos os vértices possuem a mesma estrutura topológica, mas não ambos (AKLEMAN et al., 2005).

Dentre as várias categorias de malhas regulares, é necessário escolher uma que, além de representar um terreno, possa dar suporte a métodos de refinamento e simplificação. As malhas semi-regulares do tipo 4-8 (valência dos vértices igual a 4 ou 8, exceto nas bordas) se encaixam no perfil desejado para resolver o problema tratado nesta dissertação

(Fig. 2.1).

A malha semi-regular 4-8 é uma estrutura hierárquica para subdivisão de superfícies, ou um complexo celular homeomorfo ao ladrilho de Laves [4, 8²] (VELHO; GOMES, 2000).

Várias vantagens desta estrutura podem ser mencionadas:

- são quadrangulações trianguladas refináveis, então combinam as propriedades estruturais de malhas triangulares e quadrangulares;
- suportam refinamento uniforme e não uniforme, tornando possível a geração de tesselações adaptadas em multiresolução;
- o refinamento é composto de dois passos de subdivisão binária, permitindo fatoração dos esquemas de subdivisão;
- são construídas utilizando bisseções, que são fáceis de implementar e produzem transição gradual de resolução.

As malhas semi-regulares 4-8 podem ser classificadas como adaptativas e não-adaptativas. No caso das malhas não-adaptativas, existe apenas uma hierarquia na malha, ou seja, as modificações feitas em uma região da malha afetam a resolução da malha globalmente. Já no caso das malhas adaptativas, existe uma família de hierarquias, de forma que não há interferência entre modificações locais em cada nível.

Esta dissertação utiliza malhas adaptativas para que seja possível deformar um terreno em diferentes níveis de detalhe independentes entre si e visualizá-lo em diferentes resoluções através de simplificações ou refinamentos. Um exemplo desta estrutura pode ser visto na Figura 2.1, onde é mostrado um bloco básico da estrutura e dois passos de refinamento consecutivos realizados sobre ele.

2.1.1 CONSTRUÇÃO DA ESTRUTURA HIERÁRQUICA

As tesselações de uma malha A4-8 são ladrilhamentos refináveis. Esta propriedade significa que é possível subdividir um ladrilho menos detalhado e obter um semelhante mais detalhado, ou seja, é possível construir uma tesselação em multiresolução usando refinamento. Existem dois métodos alternativos de construção: subdivisão quaternária e subdivisão binária intercalada.

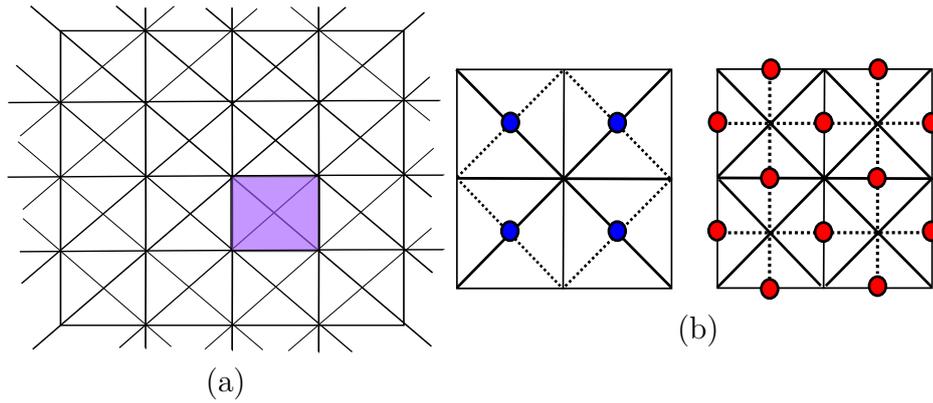


Figura 2.1: Estrutura de uma malha 4-8, em (a) um bloco básico destacado em roxo e em (b) dois passos de refinamento, o primeiro gera os vértices em azul e o segundo, os vértices em vermelho

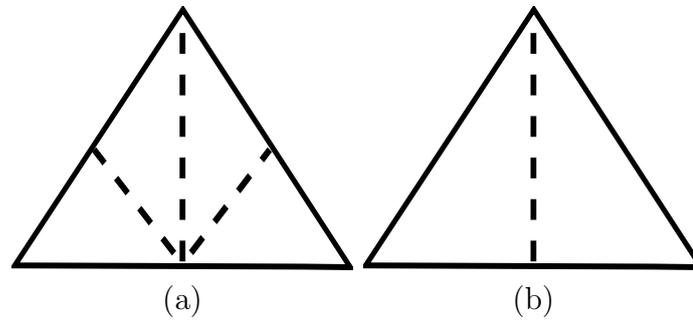


Figura 2.2: (a) Protótipo de subdivisão quaternária e (b) protótipo de subdivisão binária.

Considere uma malha $M = (\mathbf{V}, \mathbf{A}, \mathbf{P})$, onde \mathbf{V} , \mathbf{A} e \mathbf{P} é o conjunto de vértices, arestas e faces (polígonos) pertencentes à esta malha, respectivamente. Considere também, que o conjunto de vértices de valência igual a quatro é \mathbf{V}_4 e o de valência igual a oito é \mathbf{V}_8 . O procedimento de refinamento *subdivisão quaternária* de uma malha M pode ser visto na Figura 2.2a e é descrito através dos seguinte passos:

1. Divida todas as arestas $a \in \mathbf{A}$ em seus pontos médios m ;
2. Subdivida todas as faces $f \in \mathbf{P}$ em quatro novas faces, ligando o vértice de valência 4, $v \in \mathbf{V}_4$, ao ponto médio m da aresta oposta e ligando m aos pontos médios das duas outras arestas.

O procedimento de *subdivisão binária intercalada* é ilustrado pela Figura 2.2b e é realizado através dos seguintes passos:

1. Divida as arestas $a = (v_i, v_j) \in \mathbf{A}$ que são formadas por dois vértices de valência 8, $v_i, v_j \in \mathbf{V}_8$;

2. Subdivida todas as faces $f \in \mathbf{P}$ em duas sub-faces, ligando o vértice de valência 4, $v \in \mathbf{V}_4$, ao ponto médio m da aresta oposta.

2.1.1.1 Abordagens para Construção da Malha A4-8

Uma malha A4-8 pode ser construída através de duas abordagens: *top-down* e *bottom-up*. A abordagem *top-down*, também chamada método de refinamento, se caracteriza por criar uma malha na menor resolução possível e, progressivamente, aumentar a resolução através da criação de novos triângulos e adição dos mesmos à malha. Já a abordagem *bottom-up*, ou método de simplificação, cria uma malha na maior resolução possível e, de maneira análoga, reduz a resolução pela união de triângulos presentes na malha (Fig. 2.3).

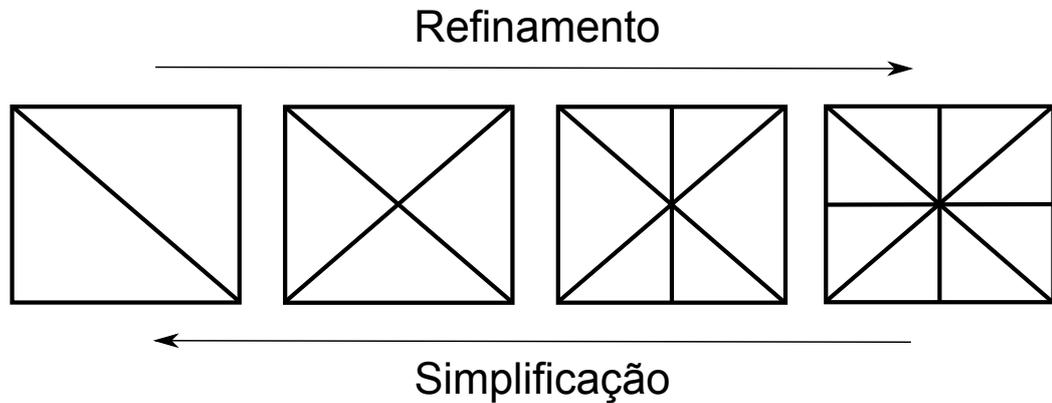


Figura 2.3: Refinamento e simplificação da malha A4-8.

A abordagem *bottom-up* é mais custosa em termos de processamento, então geralmente é utilizada para criar malhas estáticas, pré-processadas. A abordagem *top-down* cria uma malha mais simplificada e consome menos recursos, então é preferida quando necessita-se modificar a malha dinamicamente. Existem também sistemas que utilizam uma abordagem híbrida, simplificando ou refinando a malha a cada quadro, permitindo que uma técnica de coerência quadro-a-quadro seja implementada. Estes sistemas não são classificados como *top-down* ou *bottom-up*.

Nesta dissertação, a malha é construída utilizando-se uma abordagem estritamente *top-down* no processo de tesselação. Em outras palavras, a malha inicial a cada quadro sempre possui a menor resolução possível e a tesselação é responsável por refinar a malha de acordo com a necessidade. Não existe assim, na verdade, simplificação na malha quando ela passa de uma resolução maior para uma menor. Todas as resoluções são alcançadas através do processo de refinamento, que é aplicado aos triângulos da malha (visto na Sec.

2.1.1), partindo-se da malha inicial de menor resolução. Desta maneira, é possível que o aplicativo final mantenha sua eficiência mesmo com modificações sendo feitas na malha em tempo real.

2.1.2 REPRESENTAÇÃO DA ESTRUTURA HIERÁRQUICA

Uma malha M A4-8 pode ser representada através de uma estrutura em árvore de triângulos. Dependendo do tipo de refinamento utilizado, a árvore é quaternária ou binária. Esta dissertação utilizou uma estrutura de árvore binária triangular, ou *bintree* triangular (DUCHAINÉAU et al., 1997), para representar um terreno.

2.1.2.1 *Bintree* Triangular

Considere uma malha A4-8 M que possui um conjunto de vértices $\mathbf{V} \subset \mathbb{R}^2$ dado por $\mathbf{V} := \{\vec{v}_i := (u_i, v_i) \in \mathbb{R}^2, i = 1, \dots, n\}$, onde n é o número de vértices. A *bintree* triangular é uma árvore binária onde, geometricamente, cada nó da árvore representa um triângulo. O triângulo raiz, $T = (\vec{v}_a, \vec{v}_0, \vec{v}_1)$ onde $\vec{v}_a, \vec{v}_0, \vec{v}_1 \in \mathbf{V}$, é definido como um triângulo reto isósceles no nível menos refinado, ($l = 0$). No próximo nível, $l = 1$, os filhos da raiz são definidos através da inserção da aresta definida entre o vértice ápice \vec{v}_a e o ponto médio $\vec{v}_c \in \mathbf{V}$ da aresta base (\vec{v}_0, \vec{v}_1) . O filho à esquerda é $T_0 = (\vec{v}_c, \vec{v}_a, \vec{v}_0)$ e o filho à direita é $T_1 = (\vec{v}_c, \vec{v}_1, \vec{v}_a)$. Repetindo recursivamente este processo, obtém-se o restante da árvore. A Figura 2.4 ilustra alguns níveis de uma *bintree* triangular.

Um triângulo T_b é considerado *triângulo base* de T , caso divida a aresta base (\vec{v}_0, \vec{v}_1) de T . O vizinho à esquerda T_0 é um triângulo que divide a aresta esquerda (\vec{v}_a, \vec{v}_0) e o vizinho à direita T_1 , o que divide a aresta direita (\vec{v}_1, \vec{v}_a) .

Um passo do processo de refinamento substitui T pelos triângulos T_0 e T_1 . Se o triângulo T não possui vizinho na base, apenas T é refinado. Este processo de refinamento pode ser repetido enquanto for possível obter um vértice \vec{v}_c no conjunto de dados. O processo de simplificação funciona de maneira análoga.

Quando dois triângulos T e T_b estão no mesmo nível l (ou seja, no mesmo nível de detalhe), o par (T, T_b) é referido como um diamante. A operação de refinamento sobre o diamante é realizada dividindo os dois triângulos, gerando assim quatro novos filhos: (T_0, T_1) , filhos de T e (T_{b_0}, T_{b_1}) , filhos de T_b . Se o triângulo T_b não existir, somente T é refinado. A operação de simplificação une os filhos $(T_0, T_1, T_{b_0}, T_{b_1})$, os substituindo na malha

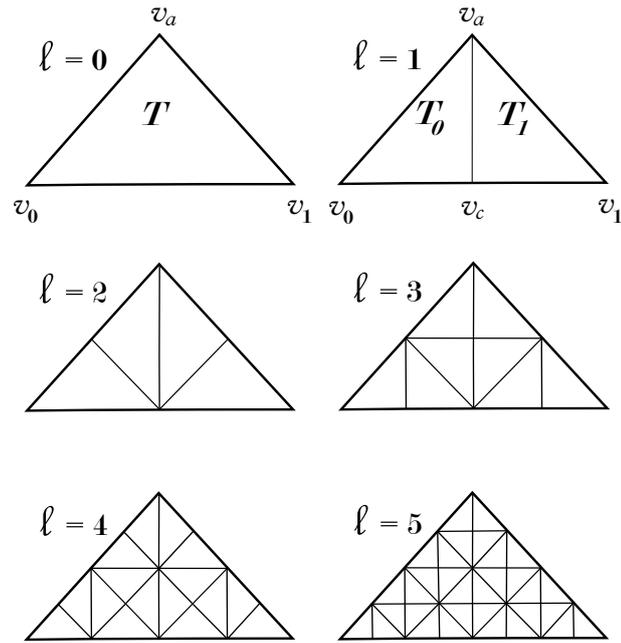


Figura 2.4: Níveis 0–5 de uma *bintree* triangular.

pelos seus pais (T, T_b) . Neste caso, $(T_0, T_1, T_{b_0}, T_{b_1})$ formam um diamante simplificável (Fig. 2.5).

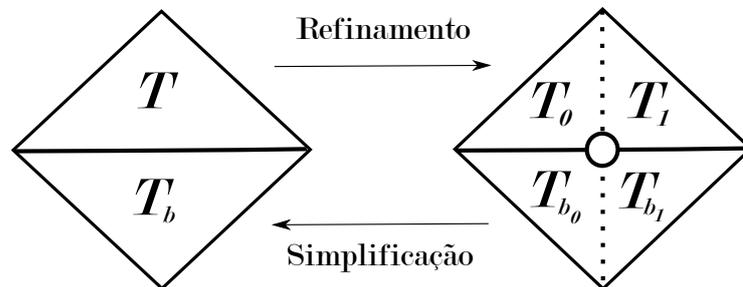


Figura 2.5: Refinamento e simplificação do diamante.

Um triângulo T em um nível l não pode ser dividido imediatamente caso seu triângulo base T_b esteja em um nível l menor (menos refinado). Para dividir T , é preciso antes forçar a divisão de T_b , o que pode exigir outras divisões forçadas. O procedimento de divisão forçada evita que a *bintree* triangular apresente descontinuidades e pode ser visto na Figura 2.6.

É importante ressaltar que a estrutura descrita nesta seção precisa ser adaptada de forma que uma malha A4-8 possa representar um terreno esférico. À medida que a *bintree* triangular é construída, “conecta-se” os vértices dos triângulos de borda da malha para que o plano seja conformado de acordo com a topologia do cilindro. No decorrer do refinamento da malha, os vértices são mapeados de acordo com alguma função f e o

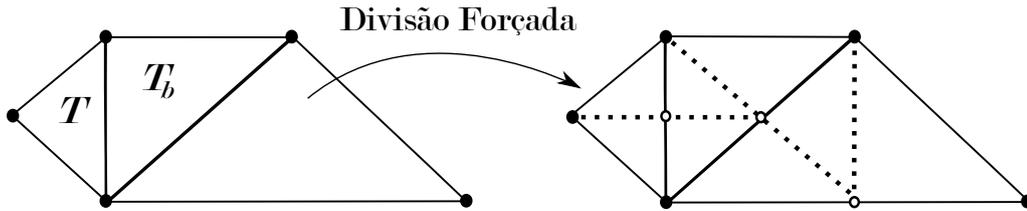


Figura 2.6: Divisão forçada de um triângulo.

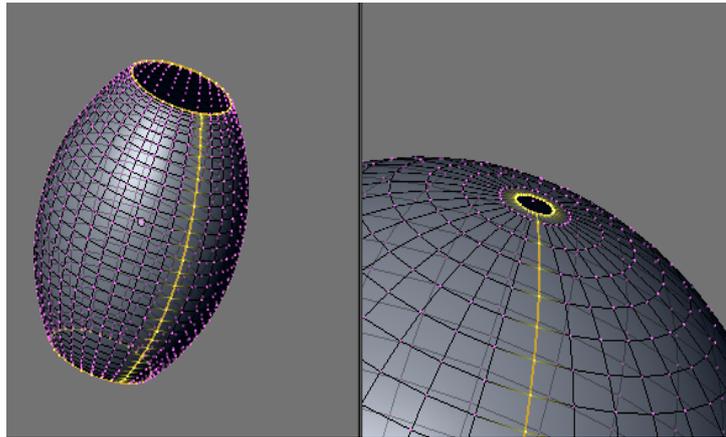


Figura 2.7: Cilindro sendo conformado em uma esfera. No meridiano da esfera as coordenadas das duas bordas do plano são conectadas (NOEL, 2010).

cilindro conforma-se em uma esfera. Os vértices do plano na latitude que passa pelos polos são mapeados em uma mesma coordenada na esfera (Fig. 2.7).

Outro fator importante a ser lembrado é que, para cada vértice $\vec{v} \in \mathbf{V}$, devem ser armazenados dois valores, w e ε . O primeiro valor, w , representa a altura do vértice definida como $w = g(f(u, v)) = g(x, y, z)$, onde f é uma função de mapeamento $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ e g é uma função que utiliza as coordenadas do vértice mapeado, x , y e z , para calcular um valor de altura. O escalar obtido é armazenado em uma matriz de escalares nas coordenadas u e v do vértice \vec{v} . A matriz de escalares que contém todas as alturas dos vértices $v \in \mathbf{V}$ forma o chamado mapa de alturas M_a . Através de modificações em M_a utilizando g , é possível deformar a superfície de um objeto contínuo (terreno esférico) como será mostrado nesta dissertação.

O segundo valor, ε , representa o valor erro de cada vértice $\vec{v} \in \mathbf{V}$. Este valor será detalhado na Seção 2.3.1. De maneira resumida, é possível dizer que o erro de um vértice é definido como $\varepsilon = \varepsilon_T$, onde ε_T é o erro do triângulo que tem \vec{v} como vértice ápice. Na prática, o valor de erro de um triângulo ε_T , para $T = (\vec{v}_a, \vec{v}_0, \vec{v}_1)$, é armazenada em uma matriz de escalares no índice u e v do vértice ápice \vec{v}_a de T . A matriz de escalares com todos os erros dos vértices $v \in \mathbf{V}$ é chamada de mapa de erros M_e .

Os valores de erro e altura de cada vértice da malha são utilizados pelo processo de tesselação para decidir se um triângulo T , do qual o vértice pertence, deve ou não ser refinado ou simplificado.

A resolução da malha também é um fator importante a ser considerado. Esta dissertação utiliza na malha valores de resolução que sejam uma potência do número dois acrescido do valor um, como por exemplo, 1025, 2049 e 4097. Esta escolha garante não só a agilidade nos cálculos realizados, como também permite que a passagem de uma determinada resolução para outra seja feita através de cálculos simples sem a necessidade de tratamentos.

Considere como exemplo uma resolução de valor cinco e uma de valor quatro, ou seja uma malha 5×5 e outra 4×4 . Realizando o refinamento, ou seja, dividindo a soma entre o menor e o maior valor de coordenada (u ou v) na malha por dois, percebe-se que no primeiro caso obtem-se a coordenada $\frac{0+4}{2} = 2$ e no segundo caso $\frac{0+3}{2} = 1.5$. Isso significa que, um valor que é uma potência do número dois acrescido do valor um, sempre permite a passagem de uma resolução para outra com um valor inteiro de coordenada resultante. Se fosse escolhido um valor qualquer para resolução, seria necessário em vários casos realizar o tratamento da coordenada resultante do refinamento, como no segundo caso mostrado acima.

2.2 PARAMETRIZAÇÃO

Uma superfície paramétrica no espaço Euclidiano \mathbb{R}^3 é definida por uma equação paramétrica de dois parâmetros, chamados neste trabalho de u e v . Em especial, deseja-se um mapeamento $f : \Omega \rightarrow \mathbf{S}$, onde f é uma função de mapeamento, Ω é um subconjunto aberto $\Omega \subset \mathbb{R}^2$ e \mathbf{S} é uma superfície $\mathbf{S} \in \mathbb{R}^3$.

Esta dissertação utilizou a projeção ortográfica como função f para mapear Ω em \mathbf{S} , mais especificamente, um plano, representado pela malha A4-8, em uma esfera. Considere, assim, que o conjunto de vértices \mathbf{V} de uma malha A4-8 representa Ω e que todos vértices $\vec{v} \in \mathbf{V}$ estão associados à coordenadas u e v no plano. Considere também, que $0 \leq u \leq m_u$ e $0 \leq v \leq m_v$.

Para definir uma função f em termos de latitude e longitude de um vértice $\vec{v} \in \mathbf{V}$,

calcula-se a latitude de \vec{v} através da Equação 2.1 e sua longitude pela Equação 2.2.

$$\text{latitude} = \frac{(v * \pi)}{m_v} - \frac{\pi}{2}. \quad (2.1)$$

$$\text{longitude} = \frac{(2u * \pi)}{m_u} - \pi. \quad (2.2)$$

Considere agora que o raio da esfera definida por \mathbf{S} é r . O domínio V e a superfície S podem ser escritos através das Equações 2.3 e 2.4, respectivamente.

$$\mathbf{V} = \{(u, v) \in \mathbb{R}^2 : 0 \leq u \leq m_u, 0 \leq v \leq m_v\}. \quad (2.3)$$

$$\mathbf{S} = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = r^2\}. \quad (2.4)$$

A função de mapeamento do plano na esfera é dada por:

$$f(u, v) = \begin{pmatrix} r * \text{cosseno}(\text{longitude}) * \text{cosseno}(\text{latitude}), \\ r * \text{seno}(\text{longitude}) * \text{cosseno}(\text{latitude}), \\ r * \text{seno}(\text{latitude}) \end{pmatrix}. \quad (2.5)$$

Deste ponto em diante, \vec{v} será utilizado como notação para descrever um vértice que pertence a V , ou seja, um vértice da malha no domínio discreto. Já \vec{v} , será utilizado quando o vértice estiver no domínio contínuo, ou seja, é o vértice de coordenadas x , y e z mapeado por $f(u, v)$.

Vale ressaltar que a função f não é bijetora, ou seja, a função f mapeia um vértice no espaço Euclidiano \mathbb{R}^3 , mas a sua função inversa f^{-1} , pode mapear diversos vértices em uma mesma coordenada em \mathbf{V} dependendo da latitude e longitude. Outro problema a ser mencionado é que a projeção não preserva a área, ou seja, esta é reduzida de acordo com a proximidade dos polos.

Estes problemas influenciam diretamente a deformação de terrenos. O vértice $\vec{v}_c \in \mathbb{R}^3$ utilizado para iniciar o processo de deformação (vértice de controle), não pode ser definido em \mathbf{S} e depois mapeado para o plano. De maneira análoga, um conjunto de vértices $\vec{v} \in \mathbf{V}$ que represente todos os vértices dentro de alguma área de deformação, não pode ser definido diretamente em \mathbf{V} , já que a projeção utilizada não preserva área. Desta forma, o vértice de controle $\vec{v}_c \in S$ é definido partindo-se do domínio discreto para o contínuo.

Já o conjunto de vértices dentro da área de deformação é definido partindo-se do domínio contínuo para o discreto.

Todo o processo de mudança de domínios, do discreto para o contínuo e vice-versa, é de vital importância para deformação de terrenos proposta neste trabalho. Além de ser necessário passar de um domínio para o outro na definição do vértice de controle e na determinação do conjunto de vértices na área de deformação, é preciso realizar essa passagem para observar a deformação no terreno.

Quando o conjunto de vértices a serem deformados é definido, os valores de altura são calculados no domínio contínuo para cada um desses vértices. O cálculo é realizado através de uma função g e os valores obtidos são posteriormente armazenados no mapa de alturas. Na próxima vez que o terreno for visualizado, a função de mapeamento f transmitirá o valor calculado por g para a superfície S no domínio contínuo.

A deformação do terreno é, então, a modificação feita nos valores de altura w dos vértices dentro da área de deformação no domínio discreto. Esta modificação é transmitida para o domínio contínuo através da modificação do raio r na função de mapeamento. Substitui-se o raio r da esfera por $r' = r + w$ na Equação de mapeamento 2.5, transmitindo-se para cada coordenada obtida pela função f a modificação de altura calculada.

2.2.1 PARTIÇÃO DO ESPAÇO DE PARÂMETROS

Para obter uma partição $\mathbf{N} \subset \mathbf{V}$ que represente todos os vértices \vec{v} com uma determinada característica, é preciso discretizar o objeto contínuo representado por \mathbf{S} . De maneira mais específica, deseja-se obter uma partição de \mathbf{V} contendo todos os vértices \vec{v} que, mapeados por $f(u, v)$, se encontrem a uma determinada distância de um vértice de controle $\vec{v}_c \in \mathbb{R}^3$. Em outras palavras, deseja-se todos os vértices $\vec{v} \in \mathbf{V}$ que estejam dentro de uma área de deformação definida no espaço Euclidiano \mathbb{R}^3 .

Para obter \mathbf{N} , calcula-se inicialmente a distância Euclidiana em \mathbb{R}^3 entre os vértices $\vec{v}^T = [x, y, z]^T$ e o vértice de controle $\vec{v}_c^T = [x_c, y_c, z_c]^T$ para todos os vértices $\vec{v} \in \mathbf{V}$. Essa distância é definida como \vec{d}_{ic} e obtida através da seguinte equação:

$$\vec{d}_{ic} = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}. \quad (2.6)$$

Depois, define-se uma função que classifica os vértices \vec{v}_i através de

$$I(u, v) = \begin{cases} 1, & \text{se } \vec{d}_{ic} \leq d \\ 0, & \text{caso contrário} \end{cases}, \quad (2.7)$$

onde $I(u, v)$ é uma função que utiliza as coordenadas u e v de um vértice \vec{v} para indicar se ele está ou não dentro de uma área de deformação e $d \in \mathbb{R}^+$.

Esta classificação do vértice de acordo com a função I será chamada de propriedade $I(u, v)$. Se a função resultar no valor um, o vértice possui a propriedade, caso contrário, não a possui. Se o vértice possui a propriedade $I(u, v)$, ele é colocado no conjunto de vértices que está dentro da área de deformação. Este conjunto de vértices será chamado de \mathbf{V}_d .

Importante ressaltar que o conjunto \mathbf{V}_d é definido no domínio discreto, mas a distância é calculada no domínio contínuo. Isso ocorre porque, se a distância fosse calculada no domínio discreto, a área de deformação resultante seria distorcida pelo mapeamento de acordo com sua localização na esfera.

2.3 MÉTRICA DE ERRO

Por motivos de simplificação, esta dissertação utilizou a métrica de erro proposta no algoritmo Malhas Otimamente Adaptáveis em Tempo Real (*ROAM*). O algoritmo usa a malha A4-8 para adaptar um terreno e a estrutura hierárquica da *bintree* triangular, já detalhada na Seção 2.1.2.1, para representá-lo.

De forma a decidir se um triângulo deve ser refinado ou simplificado, o algoritmo *ROAM* utiliza o conceito de espaços aninhados. Sendo \mathbf{C} o conjunto de descendentes de um vértice $\vec{v}_i \in \mathbf{V}$, o erro $\varepsilon_{\vec{v}_i} \geq \varepsilon_{\vec{v}_j}, \forall j \in \mathbf{C}$. Isso significa que a métrica de erro garante a monotonicidade dos valores de erro, ou seja, os “pais” em nível menos refinado sempre possuem um valor de erro maior do que seus “filhos”, que estão em níveis mais refinados do terreno.

A ideia da métrica é garantir que as áreas mais relevantes do terreno estejam mais refinadas do que as menos relevantes. No primeiro caso, as regiões montanhosas ou mais próximas do observador e no segundo, as regiões planas ou distantes do observador. Para utilizar a métrica de erro do algoritmo *ROAM*, calcula-se o erro geométrico e o erro dependente do observador para cada triângulo.

Outro algoritmo que possui uma métrica de erro mais refinada, considerado uma evolução do *ROAM*, é chamado Refinamento sem Estados de Uma Passagem Adaptativo (*SOAR*). A métrica deste algoritmo é baseada no conceito de esferas aninhadas e pode ser vista em detalhes no trabalho (LINDSTROM; PASCUCCI, 2001). Os resultados visuais obtidos através dessa métrica são mais refinados, porém o algoritmo possui um custo de processamento elevado para um aplicativo em tempo real. Este fato inviabilizou sua utilização nesta dissertação.

2.3.1 ERRO GEOMÉTRICO

O algoritmo *ROAM* usa o conceito de *wedgie* para definir o erro geométrico. Considere que T é um triângulo da *bintree* triangular e a mesma notação de vizinhança da Seção 2.1.2.1 será utilizada. Um *wedgie* é definido como o volume do mundo que contém os vértices (u, v, w) , de forma que $(u, v) \in T$ e $|w - w_T(\vec{v})| \leq e_T$, para uma determinada espessura de *wedgie* $\varepsilon_T \geq 0$. O segmento de linha de $(u, v, w - e_T)$ até $(u, v, w + e_T)$ é chamado espessura de segmento para um vértice \vec{v} .

Os erros aninhados dos *wedgies* são calculados de maneira *bottom-up*. Assume-se $\varepsilon_T = 0$ para todos os triângulos no nível mais refinado possível. A espessura do *wedgie* de um triângulo T é calculada em função das espessuras dos *wedgies* de seus filhos, ε_{T_0} e ε_{T_1} . Considerando um triângulo T de vizinhos T_0 à esquerda e T_1 à direita, calcula-se ε_T como

$$\varepsilon_T = \max(\varepsilon_{T_0}, \varepsilon_{T_1}) + |w(\vec{v}_c) - w_T(\vec{v}_c)|, \quad (2.8)$$

onde \vec{v}_c é o vértice obtido pela bissetão da aresta oposta ao vértice ápice do triângulo T e $w_T(\vec{v}_c)$ é dado por

$$w_T(\vec{v}_c) = \frac{(w(\vec{v}_0) + w(\vec{v}_1))}{2}. \quad (2.9)$$

A ideia do erro geométrico é levar em consideração o nível na estrutura hierárquica e a altura w dos vértices para definir o critério de refinamento ou simplificação da malha. Isso quer dizer que, quanto menos refinado um triângulo, ou quanto menor a sua altura, menor o seu erro geométrico.

A Figura 2.8 mostra um exemplo univariado de *wedgies* aninhados, juntamente com a

cadeia de *wedgies* que dependem de um determinado vértice v .

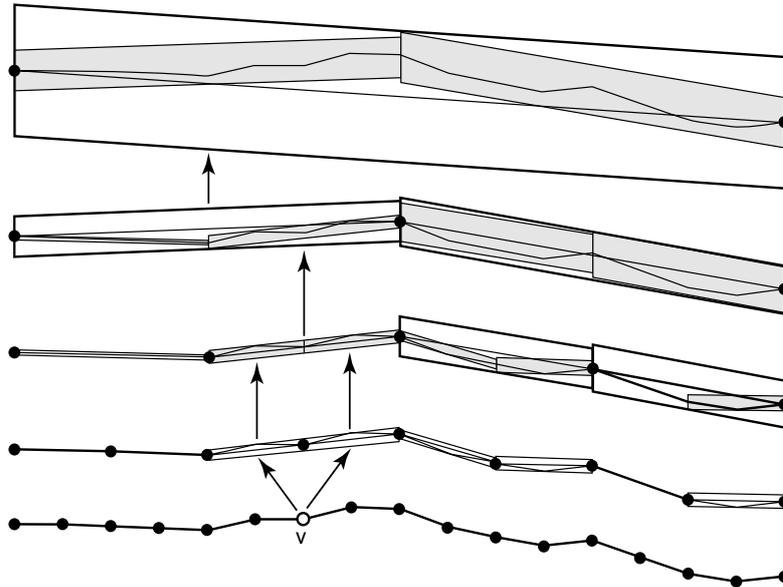


Figura 2.8: *Wedgies* aninhados em 1D, com os dependentes de v (DUCHAINEAU et al., 1997).

2.3.2 ERRO DEPENDENTE DO OBSERVADOR

O erro dependente do observador, chamada de distorção geométrica da tela, é simplesmente um cálculo de distorção geométrica. Este cálculo representa a distância entre a posição onde cada vértice da superfície \mathbf{S} deveria estar no espaço de tela e onde a triangulação toca a tela.

Seja $s(\vec{v})$ a posição correta no espaço de tela de um vértice \vec{v} dado por $f(u_{\vec{v}}, v_{\vec{v}})$ e $s_T(\vec{v})$ sua aproximação pela triangulação T . O erro neste ponto é definido por:

$$dist(\vec{v}) = \|s(\vec{v}) - s_T(\vec{v})\|$$

Na imagem inteira, o erro máximo é dado por $dist_{max} = \max_{\vec{v} \in \mathbf{V}_f} (dist(\vec{v}))$, onde \mathbf{V}_f é o conjunto de vértices \vec{v} da malha que, ao serem mapeados na esfera, estão dentro do *frustum* de visão (região do espaço que pode aparecer na tela).

Na prática, é calculado um limite superior para a distorção máxima. Para cada triângulo T na triangulação, um limite local é calculado, projetando o *wedgie* no espaço de tela. O limite é definido como o maior tamanho das espessuras ε_T de todos os vértices $\vec{v} \in T$ projetadas na tela. Este limite é definido dinamicamente pelo usuário e permite

que a malha seja forçada a se modificar para representar um determinada resolução do terreno.

3 MÉTODO PROPOSTO

3.1 VISÃO GERAL DO MÉTODO

Para deformar um terreno através da modificação do seu mapa de alturas, é proposto um método que possui as seguintes etapas:

- Definição dos pontos de controle;
- Definição da área de deformação;
- Propagação de alturas e erros.

A primeira etapa define onde será feita a deformação no terreno através de pontos de controle. A segunda etapa utiliza os pontos de controle para definir uma área de deformação, ou seja, o conjunto de vértices $\vec{v} \in \mathbf{V}$ que serão modificados. A última etapa descreve a propagação de valores de altura e erro para os vértices do terreno contidos na área de deformação.

Vale ressaltar que o método é uma parte de um sistema que realiza outras funções como: detecção de marcadores fiduciais e cálculo da distância relativa entre eles; renderização de acordo com uma métrica de erro e translação; escala e rotação do terreno esférico. A Figura 3.1 mostra as etapas do método proposto e sua relação com o sistema. Em azul as partes do sistema independentes do método, em vermelho as etapas do método e em verde as entradas controladas dinamicamente pelo usuário.

3.2 DEFINIÇÃO DO PONTO DE CONTROLE

Uma prática comum que precede a aplicação de técnicas para deformação de malhas é a definição de pontos de controle. Como o próprio nome já diz, estes pontos servem para controlar algum aspecto do processo de deformação. Eles podem ser usados, por exemplo, para controlar onde começa o processo de deformação (GREGORY et al., 1999; FOSKEY et al., 2005), ou construir uma grade que defina uma área ou volume da superfície a ser deformado (SEDERBERG; PARRY, 1986; COQUILLART, 1990).

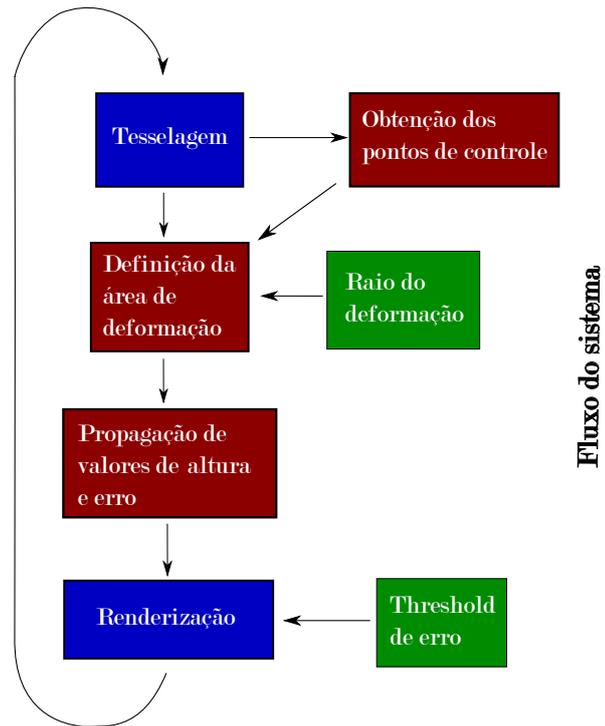


Figura 3.1: Visão geral do método.

Esta dissertação utiliza um vértice \vec{v}_c de controle, que define o centro da área de propagação dos valores de altura e erro, ou seja, onde começa o método aqui proposto. Ele é obtido durante a tesselação da malha, representando o vértice $\vec{v} \in \mathbf{V}$ que, ao ser mapeado na esfera, está o mais próximo possível do objeto deformador (marcador fiducial pré-definido). A medida que a tesselação ocorre, a posição desse vértice é atualizada se algum vértice criado se encontra mais próximo do marcador fiducial. A escolha do ponto de controle só ocorre se o marcador móvel estiver no campo de visão da câmera.

Deste ponto em diante, o ponto de controle será chamado de *vértice de controle* para facilitar o entendimento. Um exemplo de vértice de controle obtido sobre a superfície do terreno esférico pode ser visto na Figura 3.2a, onde a esfera azul representa o vértice de controle \vec{v}_c e a esfera vermelha é a posição do marcador móvel modificada para o sistema de coordenadas do terreno esférico.

Se a posição da câmera se aproxima do terreno ou o usuário aumenta a escala do modelo, a malha passa por um processo de refinamento que pode provocar a mudança do vértice de controle. A Figura 3.2 mostra um exemplo de vértice de controle, definido como \vec{v}_c , antes da atualização devido ao refinamento da malha (Fig. 3.2a) e depois (Fig. 3.2b).

Vale ressaltar que, se o usuário movimentar o marcador móvel, \vec{v}_c também muda de

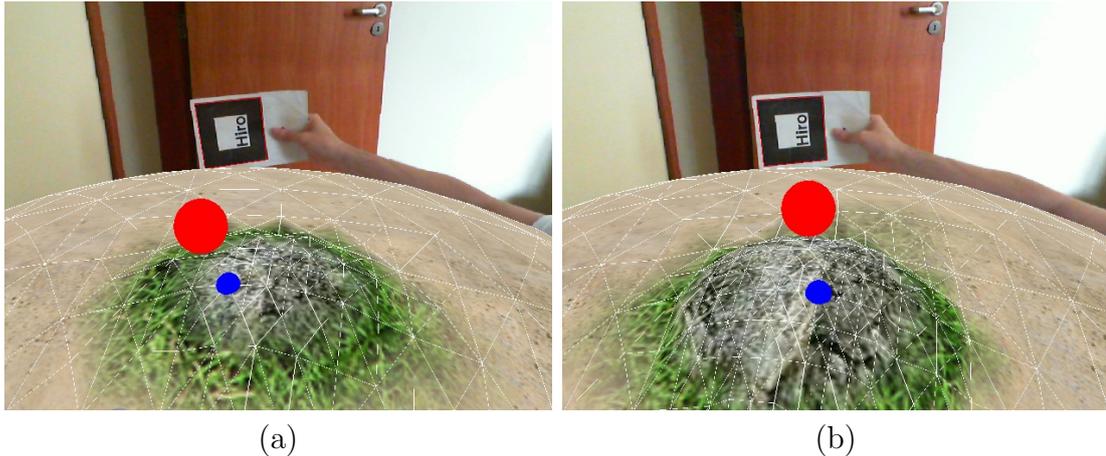


Figura 3.2: (a) Vértice de controle \vec{v}_c e (b) a atualização da sua posição devido ao refinamento da malha. A esfera azul representa o vértice de controle \vec{v}_c e a esfera vermelha é a posição do marcador móvel.

maneira dinâmica. Da mesma forma, se o modelo é rotacionado, transladado ou escalonado, a posição de \vec{v}_c pode ser atualizada. O método aqui proposto utiliza um vértice de controle determinado durante o processo de tesselação, permitindo que ambas as tarefas fossem feitas em paralelo, agilizando o processo.

3.3 DEFINIÇÃO DA ÁREA DE DEFORMAÇÃO

A área de deformação representa o conjunto de vértices $\mathbf{V}_d \subset \mathbf{V}$ que, quando mapeados na esfera através de $f(u, v)$, se encontram a uma distância menor ou igual a um raio em relação a um ponto de controle $\vec{v}_c \in \mathbb{R}^3$ (Seção 2.2.1). Esse raio é definido dinamicamente pelo usuário girando-se o botão do meio do mouse e será chamado daqui em diante de raio de deformação, ou r_d . A área de deformação é construída de acordo com um conjunto de restrições descritas a seguir.

Seja \mathbf{V} o conjunto de vértices da malha A4-8. Sub-regiões $\mathbf{V}_i \in \mathbf{V}$, que representam partições de \mathbf{V} dotadas de alguma propriedade $I(\mathbf{V}_i)$, podem ser construídas através das seguintes restrições:

1. $\bigcup_{i=1}^n \mathbf{V}_i = \mathbf{V}$;
2. \mathbf{V}_i é uma região conectada, $i = 1, 2, \dots, n$;
3. $\mathbf{V}_i \cap \mathbf{V}_j = \emptyset$ para todo i e j , $i \neq j$;
4. $I(\mathbf{V}_i) = 1$ para $i = 1, 2, \dots, n$;

5. $I(\mathbf{V}_i \cup \mathbf{V}_j) = 0$ para $i \neq j$, onde \mathbf{V}_i e \mathbf{V}_j são adjacentes.

A primeira restrição significa que a partição deve ser completa, ou seja, todos os vértices de \mathbf{V}_i devem estar contidos em \mathbf{V} . A segunda, exige que todos os vértices da sub-região estejam “conectados”. A terceira restrição indica que sub-regiões diferentes devem ser desconexas. A quarta exige que todos os vértices $\vec{v}_i \in \mathbf{V}_i$ devem satisfazer as propriedades $I(\mathbf{V}_i)$. A última restrição significa que sub-regiões distintas possuem propriedades diferentes.

As restrições mostradas acima podem ser reinterpretadas de acordo com o problema desta dissertação. Uma propriedade chamada $I(u, v)$ é definida para cada vértice da malha de acordo com a Equação 2.7, onde a distância é calculada pela Equação 2.6. Deseja-se assim, particionar o terreno em dois conjuntos desconexos, um conjunto \mathbf{V}_d que possui os vértices dotados da propriedade $I(u, v)$ e outro complementar a este $\bar{\mathbf{V}}_d$. Daqui em diante nesta dissertação, os vértices de \mathbf{V}_d serão chamados de internos e os vértices de $\bar{\mathbf{V}}_d$ serão chamados de externos.

A formulação proposta acima segue o mesmo princípio dos processos de segmentação de imagens, onde geralmente deseja-se separar o fundo e uma região de interesse. Esta escolha deve-se ao fato de que a estrutura quase totalmente regular das malhas semi-regulares permite o uso de algoritmos tradicionais de processamento de imagens. Como exemplo dessa combinação, é possível citar trabalhos que utilizam algoritmos baseados em subdivisão (ZORIN; SCHRÖDER, 2000), Fourier (PENG et al., 2001) e ondaletas (KHODAKOVSKY et al., 2000).

3.3.1 CRESCIMENTO DE REGIÃO

O algoritmo utilizado nesta dissertação para determinar a área de deformação aplica a técnica de segmentação de imagens chamada crescimento de região para percorrer a malha A4-8 (STOCKMAN; SHAPIRO, 2001). Essa técnica representa a implementação da função recursiva de expansão responsável por percorrer a malha. Ela realiza uma expansão através do domínio espacial de uma imagem, partindo de sementes definidas pelo usuário (início da região de crescimento) e expandindo até pontos adjacentes, de acordo com algum critério pré-definido de filiação a determinada região. Essa técnica possui como vantagem a facilidade de implementação e o caráter de paralelismo em sua formulação, tornando-a uma ferramenta eficiente para percorrer a malha.

É utilizada aqui uma versão da técnica de crescimento de região que usa apenas uma semente como entrada, também chamada de crescimento de região por semente simples. O ponto de partida é o vértice de controle \vec{v}_c e a expansão ocorre enquanto houver vértices que possuem a propriedade $I(u, v)$. Daqui em diante, o conjunto de vizinhos de \vec{v}_c que possuem a propriedade $I(u, v)$ será chamado de \mathbf{V}_{d_c} .

O algoritmo para determinação da área de deformação será apresentado em duas partes. A primeira, contém as linhas gerais do algoritmo, como os dados de entrada e de saída e suas funções. A segunda parte apresenta detalhadamente a parte principal do algoritmo, ou seja, a função que marca todos os vértices que possuem a propriedade $I(u, v)$.

3.3.1.1 Visão Geral do Algoritmo

O algoritmo testa, inicialmente, os vizinhos de \vec{v}_c e passa para a função recursiva de expansão o conjunto \mathbf{V}_{d_c} . Essa função expande \mathbf{V}_{d_c} a partir de \vec{v}_c na direção de cada um dos seus vértices vizinhos $\vec{v}_i \in \mathbf{V}_{d_c}$. A visão geral do processo é apresentada no Algoritmo 1.

Algoritmo 1: Visão geral do algoritmo para determinação de área de deformação.

Entrada: Vértice de controle \vec{v}_c .

Saída: Conjunto de vértices \mathbf{V}_{d_c} .

/ Função que testa e marca os vizinhos de \vec{v}_c . */*

InicializaArea(\vec{v}_c);

para cada *vértice interno* \vec{v}_i *vizinho de* \vec{v}_c **faça**

/ Função recursiva que expande na direção de \vec{v}_i . */*

DeterminaArea(\vec{v}_i);

fim para cada

A vizinhança é percorrida na malha A4-8, porém o teste de distância é feito no espaço Euclidiano \mathbb{R}^3 como mencionado na Seção 2.2. Um exemplo do teste de um vértice \vec{v}_i vizinho de \vec{v}_c , através da distância \vec{d}_{i_c} entre os dois, pode ser visto na Figura 3.3. O conjunto final de vértices \mathbf{V}_{d_c} pode ser visto na Figura 3.4.

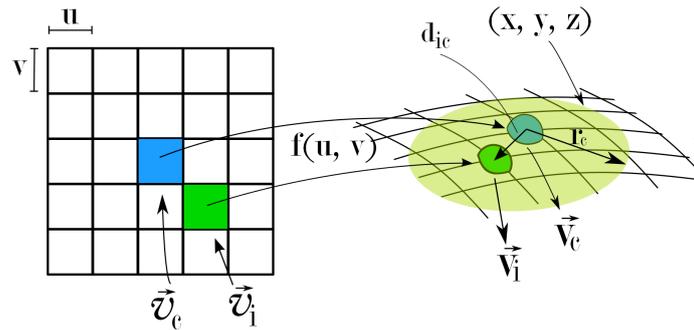


Figura 3.3: Início do processo de expansão. O vértice de controle é \vec{v}_c no domínio discreto e \vec{v}_c no domínio contínuo. O raio que define a área de deformação é r_d e \vec{d}_{ic} é a distância em \mathbb{R}^3 entre o vértice de controle \vec{v}_c e um de seus vizinhos \vec{v}_i .

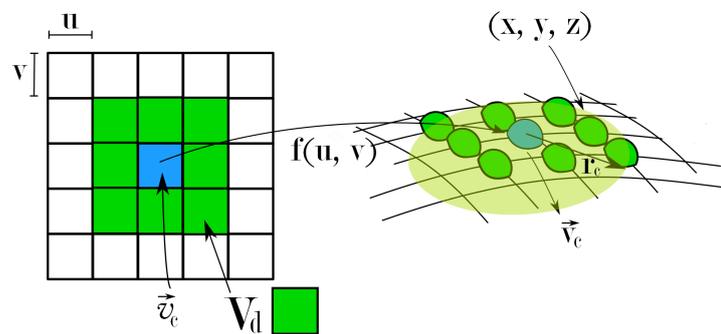


Figura 3.4: O conjunto \mathbf{V}_{d_c} , em verde, representa os vértices que possuem a propriedade $I(u, v)$. No exemplo, todos os vizinhos estão dentro da área de deformação.

3.3.1.2 Função Recursiva de Expansão

A função recursiva é responsável por “caminhar” pela malha na direção de cada um dos vizinhos de \vec{v}_c , aplicando a “propriedade” aos vértices. É possível separar as oito vizinhanças de um vértice em dois casos, direção diagonal e direção vertical/horizontal. Um vizinho está em uma direção diagonal se as duas coordenadas do vértice original no espaço paramétrico, u e v , se modificam em um passo do algoritmo naquela direção. Para a direção vertical/horizontal, um passo do algoritmo modifica apenas uma coordenada, u ou v , do vértice original.

A combinação dos casos mencionados acima permite que o algoritmo percorra toda área de deformação na malha. Se a direção é diagonal, a recursão apenas repete a direção de expansão do passo anterior (Fig. 3.5a). Já no caso da direção vertical/horizontal, a recursão é chamada para direção de expansão do passo anterior e suas duas direções diagonais adjacentes (Fig. 3.5b). A situação após um passo do algoritmo para todos os vértices internos vizinhos de \vec{v}_c é mostrada na Figura 3.5c.

A Figura 3.5b ilustra um passo do algoritmo de expansão para o vértice $\vec{v}_2 \in \mathbf{V}_{d_c}$ de

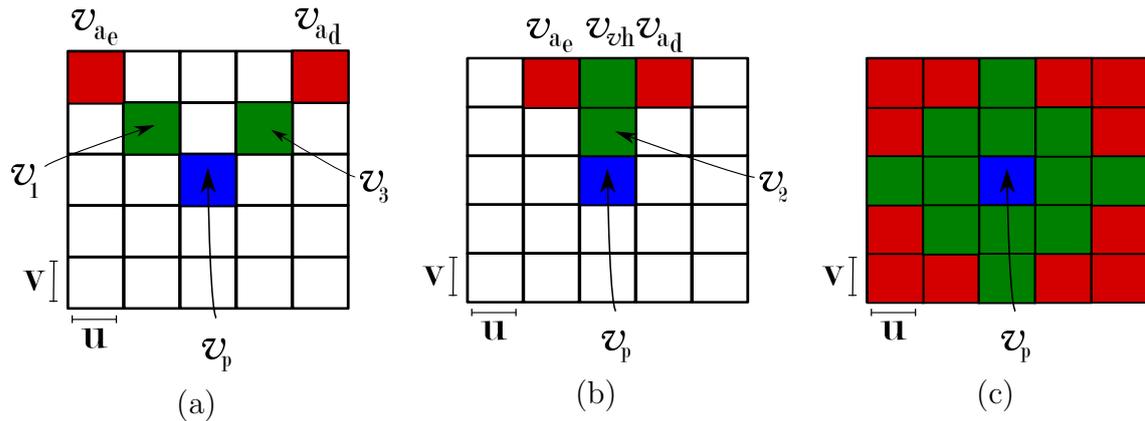


Figura 3.5: Exemplo de expansão na direção diagonal (a) e na direção vertical/horizontal (b). Um passo do algoritmo de expansão após todos os vizinhos de \vec{v}_c serem computados pode ser visto em (c).

coordenadas $(u, v + 1)$ em relação ao vértice principal \vec{v}_p . Nesta figura, \vec{v}_{vh} é o vértice na direção vertical/horizontal em relação ao vértice principal, \vec{v}_{ae} e \vec{v}_{ad} são os vértices nas direções diagonais adjacentes a direção de \vec{v}_{vh} . A função recursiva de expansão está escrita no Algoritmo 2.

Algoritmo 2: Função principal do algoritmo de determinação de área.

```

/* Este teste é feito para todos os vizinhos de  $\vec{v}_c$ .                                     */
se  $\vec{v}_i$  possui a propriedade  $I(u, v)$  então
    seleccione direção de propagação faça
        caso Diagonal
            DeterminaArea( $(\vec{v}_i)$ );
        caso Vertical/Horizontal
            DeterminaArea( $(\vec{v}_{vh})$ );
            DeterminaArea( $(\vec{v}_{ae})$ );
            DeterminaArea( $(\vec{v}_{ad})$ );
    fim selec
fim se
retorne;

```

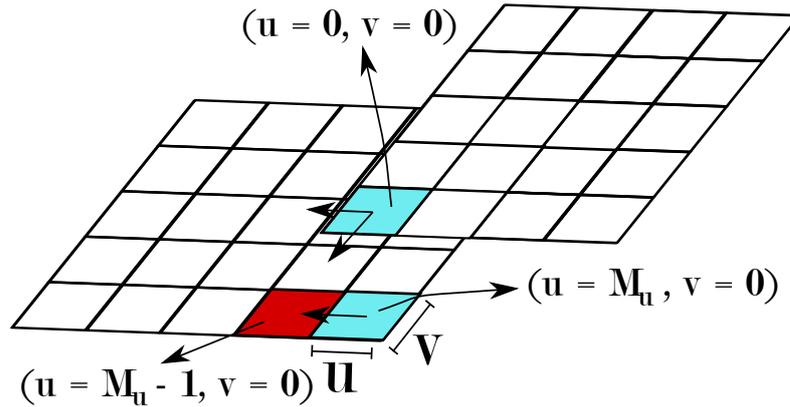


Figura 3.6: Tratamento de bordas no caso de vértices se expandindo além das dimensões do terreno. Para a latitude, se um vértice \vec{v} extrapola as dimensões do terreno, a expansão é interrompida naquela direção. No caso da longitude, a coordenada u do vértice é modificada para que ele possa continuar expandindo.

3.3.2 TRATAMENTO DE BORDA

Quando um vértice $\vec{v}_i \in \mathbf{V}_{d_c}$ se encontra na borda da malha, ou seja, possui coordenada $u = \{0 \text{ ou } m_u\}$ e/ou $v = \{0 \text{ ou } m_v\}$, o método precisa modificar sua rotina para promover a expansão. Se o vértice está na borda em sua coordenada u (longitude igual a 180°), sua coordenada deve ser modificada de forma a refletir que o plano foi inicialmente conformado em um cilindro (Seção 2.2). Em resumo, se o vértice atingir a coordenada $u = 0$, o próximo passo do algoritmo irá modificá-la para $u = m_u - 1$. De forma análoga, se a coordenada for $u = m_u$, o próximo passo do algoritmo modificará esta coordenada para $u = 1$.

O pulo de uma coordenada para a longitude é proposital, refletindo que para as coordenadas $u = 0$ e $u = m_u$ existe uma sobreposição na malha. É importante ressaltar que se um vértice possui coordenada u de borda no terreno, qualquer modificação feita no mapa de alturas e erros para este vértice deve ser repetida na outra extremidade do mapa para evitar rachaduras no terreno.

No caso da coordenada v (latitude), se um vértice atingir qualquer uma das coordenadas da borda, o algoritmo simplesmente para de propagar. Isso se deve ao fato de que, para a topologia da esfera, é necessário apenas conectar as bordas da malha no sentido leste e oeste. Para a topologia do toro, o caso seria o inverso, tornando o método facilmente modificável. Um exemplo do tratamento de bordas pode ser visto na Figura 3.6.

3.4 PROPAGAÇÃO DE ALTURAS E ERROS

Determinada a área de deformação, ou conjunto $\mathbf{V}_d \in \mathbf{V}$, é preciso propagar valores de alturas e erro para todos os vértices do conjunto \mathbf{V}_d . O algoritmo de propagação pode ser definido como uma extensão do algoritmo de determinação de área, já que ambos percorrem todos os vértices que possuem a propriedade $I(u, v)$. O algoritmo final para modificação de alturas e erros utiliza a função recursiva de expansão para percorrer os vértices e novas funções para calcular altura e os erros dos vértices no conjunto \mathbf{V}_d . No caso da propagação de erro, a rotina de expansão deve ser modificada para refletir a métrica proposta pelo algoritmo *ROAM*.

3.4.1 CÁLCULO E PROPAGAÇÃO DE ALTURAS

A função utilizada nesta dissertação para calcular os valores de alturas é a função de distribuição Gaussiana $g(x)$ descrita pela Equação 3.1. A função tem valor máximo no vértice de controle e valor mínimo nos limiares da área de deformação. Nesta função, se o desvio padrão σ for pequeno em relação a média μ , a parte visível da deformação se concentra perto de $\vec{\mathbf{v}}_c$, enquanto no caso inverso, a parte visível da deformação apresenta mudança abrupta de alturas nas regiões limites da área de deformação.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (3.1)$$

O processo envolve a passagem do domínio discreto para o contínuo e vice-versa. Um vértice $\vec{v} \in \mathbf{V}$ é mapeado na esfera e testado, para saber se está ou não dentro da área de deformação. Se ele for um vértice interno, a distância Euclidiana em \mathbb{R}^3 do vértice \vec{v} , mapeado por f , para o vértice de controle $\vec{\mathbf{v}}_c$, é passada para função Gaussiana definida pela Equação 3.1 e um valor de altura é obtido. Este valor é, então, associado a um escalar w e armazenado no mapa de alturas representado por uma matriz, através de suas coordenadas (u, v) . O novo valor de w é acessado na próxima tesselação, onde a função de mapeamento transmitirá a modificação feita no mapa de alturas para a superfície do terreno.

3.4.2 MÉTRICA E PROPAGAÇÃO DE ERRO

A métrica de erro proposta nesta dissertação é a mesma utilizada no algoritmo *ROAM*, porém com uma implementação diferente. Da mesma forma que para a propagação dos valores de altura, deseja-se percorrer todos os vértices $\vec{v} \in \mathbf{V}_d$ para propagação de erro. Porém uma restrição fundamental da métrica de erro é que a monotonicidade seja garantida, ou seja, os pais possuem erro maior que o erro dos filhos (Seção 2.3).

Essa condição seria facilmente violada utilizando apenas a função de expansão descrita pelo Algoritmo 2. Considere, por exemplo, um determinado vértice \vec{v} que ao ser mapeado na esfera localiza-se próximo ao limiar da área de propagação. Este vértice tem um valor de altura baixo devido a distância que se encontra da posição de \vec{v}_c . Se o erro fosse propagado de maneira análoga à propagação de alturas, o vértice \vec{v} teria também um valor de erro baixo, o que não é verdade para maioria dos casos.

Para exemplificar o cálculo correto de erros em um triângulo T , considere a Figura 3.7. Os valores no triângulo da esquerda (Fig. 3.7a) representam as alturas w dos vértices $\vec{v} \in T$. No triângulo da direita (Fig. 3.7b) os respectivos valores de erro calculados pela Equação 2.8. Os valores de erro $\varepsilon = 0$ de um vértice \vec{v} significam que ele faz parte de um triângulo que está no último nível de resolução.

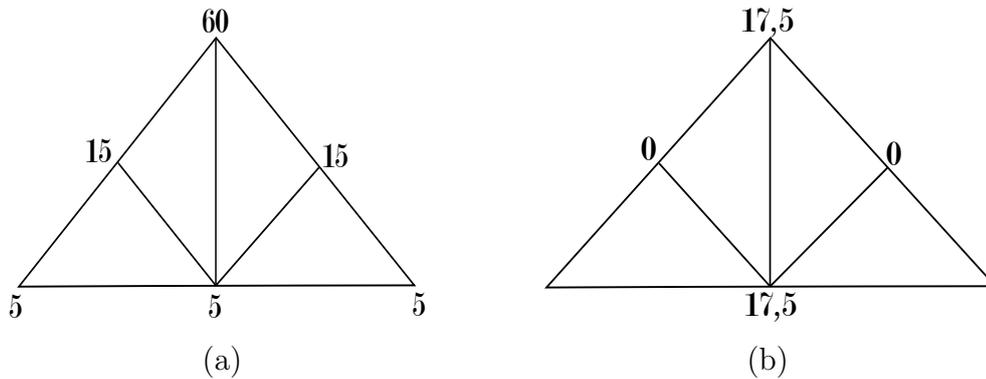


Figura 3.7: Exemplo de propagação de erro correta. No triângulo da esquerda (a) valores de altura para os vértices do triângulo T e no triângulo da direita (b) os respectivos valores de erro.

Para propagar erros de forma correta, é necessário utilizar uma função de expansão que percorra a malha de maneira hierárquica. Os mesmos casos relativos a direções de vizinhos de um vértice \vec{v} , visto na Seção 3.3.1.2, podem ser utilizados. Apenas a lógica precisa ser modificada.

A malha é percorrida de maneira hierárquica utilizando o conceito de passo alternado.

Considere um número de iterações $i = 0, 1, 2, \dots, n$ e um tamanho de passo p_a . Sabendo que todos os nós folha da árvore binária triangular possuem erro $\varepsilon_T = 0$ e que o erro cresce a medida que reduzimos o nível de resolução até atingir-se o nó raiz, é possível averiguar na árvore que a posição dos nós é espaçada de maneira homogênea para valores similares de erro.

Considere, por exemplo, a Figura 3.8a que mostra a disposição dos vértices que possuem erro $\varepsilon_T = 0$ em um triângulo T no maior nível de resolução. Percebe-se que é possível caminhar na estrutura percorrendo todos os vértices nesta resolução, utilizando-se passos p_l de tamanho ímpar na direção vertical, seguido de “caminhadas” nas duas direções diagonais ao vértice alcançado pelo passo inicial.

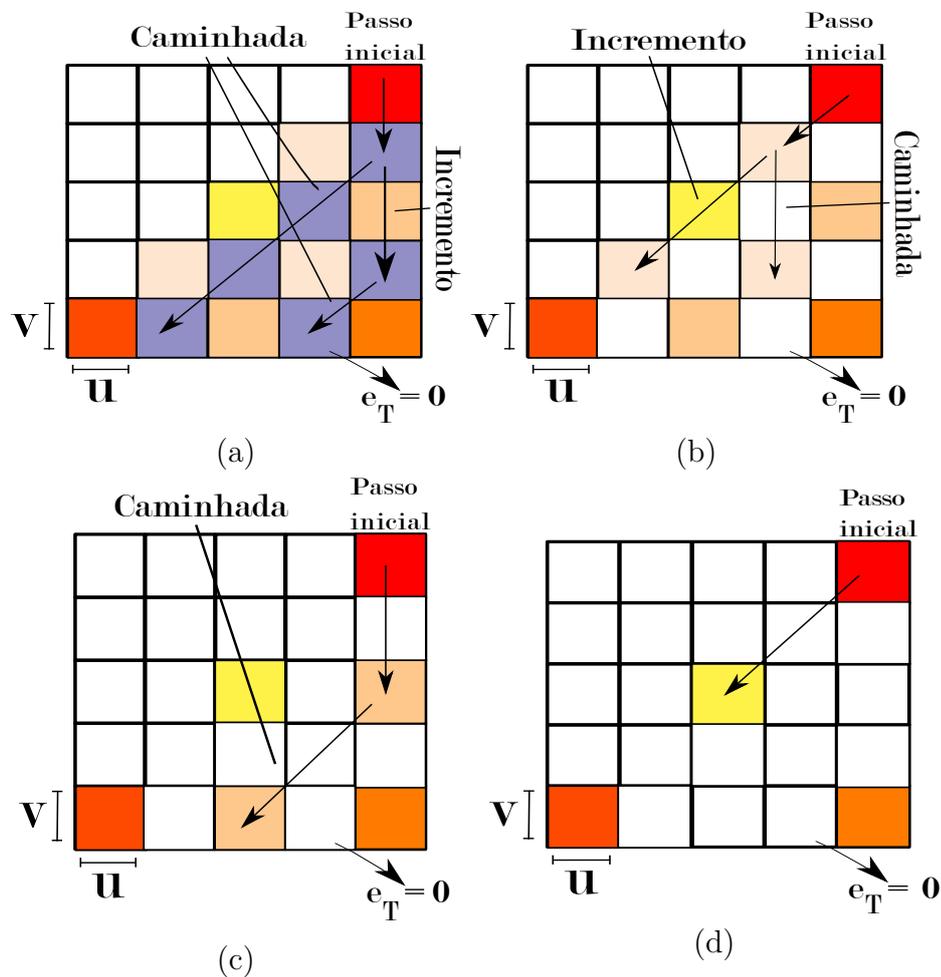


Figura 3.8: Exemplo de expansão hierárquica. Em (a) ocorre o primeiro caso, passo na vertical e caminhada diagonal. Em (b) acontece o segundo caso, passo diagonal e caminhada vertical, ambos na iteração de valor zero. Já em (c) repete-se o primeiro caso e em (d) o segundo, agora para iteração de valor um.

De maneira mais específica é possível afirmar neste caso, que o passo inicial a cada iteração i é dado pela Equação 3.2 e o incremento do passo i_c é dado pela Equação 3.3.

Nesse caso onde a resolução é máxima, a iteração $i = 0$, o passo inicial $p_l = 1$, o incremento do passo $i_c = 2$ na direção vertical e o passo para caminhada na diagonal $p_d = p_l$.

$$p_l = 2^i. \quad (3.2)$$

$$i_c = 2 * p_l. \quad (3.3)$$

Considere agora o exemplo da Figura 3.8c, dois níveis de resolução abaixo do máximo. A iteração agora é $i = 1$ e o passo inicial $p_l = 2$, validando por indução as Equações 3.2 e 3.3. O motivo de i ser igual a um é que para cada iteração é possível percorrer dois níveis de resolução, um para cada caso sendo apresentado aqui.

Para o nível de resolução que está imediatamente abaixo do máximo, o modelo se modifica (Fig. 3.8b). O passo inicial e o incremento são obtidos da mesma maneira que no caso anterior, seguindo as Equações 3.2 e 3.3. A iteração é novamente $i = 0$, $p_l = 1$ e $i_c = 2$. Neste caso, porém, o passo inicial é feito na direção diagonal e a “caminhada” pela estrutura é feita na direção vertical (o inverso do caso anterior). A recursão caminha na vertical com um passo $p_a = i_c$, sendo que para esta resolução $p_a = 2$.

A combinação destes casos permite que todos os vértices sejam percorridos, como pode ser visto na Figura 3.8d. Um algoritmo pode ser facilmente extraído destes dois casos, representando uma maneira hierárquica de percorrer a malha A4-8. A monotonicidade é garantida desta forma, pois o erro pode continuar sendo calculado por uma abordagem *bottom-up*. A propagação de erro está descrita no Algoritmo 3, onde $\vec{d}_{\mathbf{v}_{vh}}$, $\vec{d}_{\mathbf{v}_{ae}}$ e $\vec{d}_{\mathbf{v}_{ad}}$ são as distâncias entre o vértice de controle $\vec{\mathbf{v}}_c$ e os vértices nas direções vertical/horizontal,

diagonal esquerda e diagonal direita, respectivamente.

Algoritmo 3: Algoritmo de propagação de erros.

```

iteracao ← 0;
propaga ← true;
enquanto propaga = true faça
    passo = 2iteracao;
    incremento = 2 * passo;
    se  $\vec{d}_{v_{vh}} \leq r_d$  então
        | PropagaErro( $v_{vh}$ passo, passo);
    fim se
    se  $\vec{d}_{v_{ae}} \leq r_d$  então
        | PropagaErro( $v_{ae}$ passo, incremento);
    fim se
    se  $\vec{d}_{v_{ad}} \leq r_d$  então
        | PropagaErro( $v_{ad}$ passo, incremento);
    fim se
    se  $\vec{d}_{v_{ad}} \geq r_d$  e  $\vec{d}_{v_{ae}} \geq r_d$  e  $\vec{d}_{v_{vh}} \geq r_d$  então
        | propaga ← false;
    fim se
    iteracao ← iteracao + 1;
fim enqto

```

A função *PropagaErro()* é responsável por caminhar pela estrutura como mencionado nos dois casos acima. Ela tem como argumentos o vértice alcançado pelo passo inicial e o tamanho do passo. Essa função é recursiva e caminha na direção do vértice passado como argumento. Ela pára quando um vértice \vec{v} alcançado for externo. A propagação de erro utiliza a Equação 2.8 para calcular o erro ε_T de cada vértice. Os valores de erro ε_T de cada vértice \vec{v} são associados a um escalar ε e armazenados em uma matriz de acordo com as coordenadas do vértice u e v .

4 RESULTADOS

Este capítulo apresenta os resultados obtidos com o método proposto, detalhes da visualização, tabelas de desempenho do método e particularidades do problema. Para validar o método foram utilizados malhas com resoluções máximas de 2049×2049 , 4097×4097 e 8193×8193 vértices. O número máximo de triângulos gerados pelo processo de tesselação nos testes realizados foi definido como 70000.

4.1 VISUALIZAÇÃO DO TERRENO ESFÉRICO

Pensando na interatividade do usuário com o terreno esférico, duas janelas de visualização foram definidas. Desta forma, o usuário consegue ver os resultados da tesselação em uma escala global e local. Na janela de visão local, a malha está imersa em um ambiente de realidade aumentada. Na janela de visão global, a mesma malha é visualizada em um ambiente puramente virtual, para que o usuário consiga visualizar o *frustum* local e a tesselação do planeta, possibilitando um maior controle sobre a modelagem como um todo.

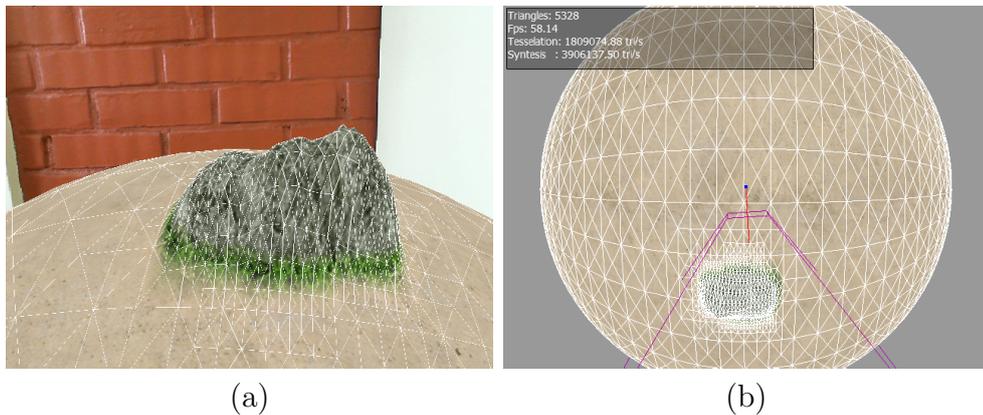


Figura 4.1: Visualização de um terreno esférico. Janela de visão local (a) e global (b).

O usuário pode realizar as operações básicas de translação, rotação e escala no terreno para melhor visualizá-lo. Para transladar o terreno, o usuário movimenta o marcador fiducial utilizando suas mãos. A rotação do terreno pode ser feita através de atalhos no teclado ou de movimentos de giro segurando o marcador. Para modificar a escala do terreno, o usuário pode utilizar o mouse ou afastar/aproximar o marcador da câmera.

Como visto anteriormente, a medida que o usuário modifica a distância do terreno para câmera, o erro dependente do observador muda e a malha se adapta através de refinamento ou simplificação. É possível modificar essa distância mudando a escala do modelo, movendo o marcador associado ao terreno ou movendo a câmera. A Figura 4.2 mostra um exemplo de mudança na malha devido a escala do terreno. A posição inicial do terreno pode ser vista na Figura 4.2b, a modificação na malha quando a câmera se aproxima do terreno é vista em 4.2c e quando se afasta na Figura 4.2a.

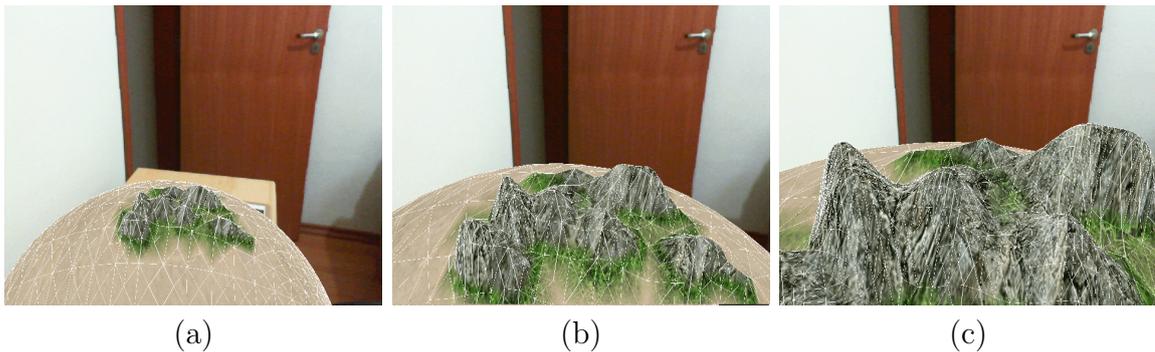


Figura 4.2: Mudanças na malha provocadas pela distância entre câmera e terreno. Posição inicial do terreno em (b), câmera aproximada do terreno (c) e afastada (a).

4.2 EXPERIMENTOS REALIZADOS

Todos os experimentos foram feitos em um computador com processador AMD FX(tm)-6100 de seis núcleos, memória RAM de 8GB e placa de vídeo GeForce GTX 550 Ti com 2GB de memória RAM no Sistema Operacional Windows 7. A câmera Microsoft LifeCam foi utilizada como dispositivo de captura de vídeo. A resolução de 640×480 *pixels* foi escolhida para as janelas de visualização.

O resultado da deformação de um terreno esférico com o método proposto pode ser observado na Figura 4.3. A imagem mostra uma sequência de quadros, da esquerda para direita, onde o usuário realiza um movimento com o marcador fiducial através da superfície do terreno provocando a deformação do mesmo.

As matrizes contendo os valores de alturas e erros de todos os vértices $\vec{v} \in \mathbf{V}$ foram inicializadas com o valor zero. Isso significa que, inicialmente, o terreno é renderizado na menor resolução possível com todos os vértices possuindo altura $w = 0$. A medida que o usuário deforma o terreno, a resolução e os valores de altura dos vértices mudam dinamicamente devido a vários fatores já mencionados. Vale ressaltar que é possível

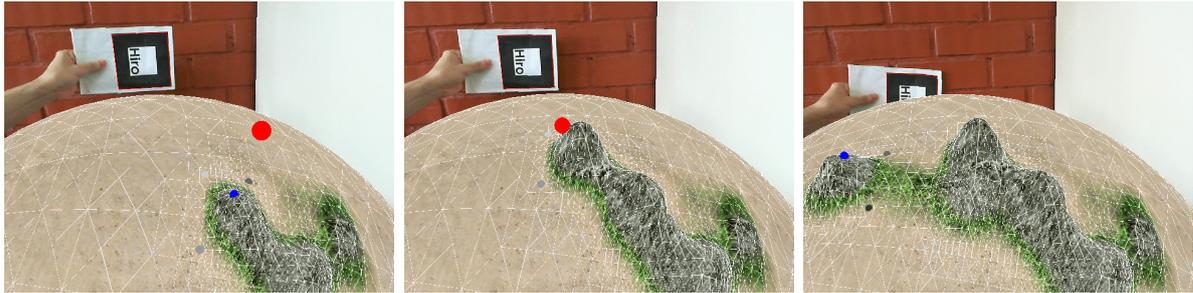


Figura 4.3: Uma sequência de quadros espaçados no tempo que demonstram a malha sendo deformada utilizando o método proposto.

também carregar um mapa de alturas e erros pré-computado.

Nesta dissertação, o valor de erro máximo da imagem $dist_{max}$ foi testado para valores variando entre 0.1 e 6.0. Os experimentos realizados mostraram que o valor ideal que promove um equilíbrio entre quantidade de triângulos e qualidade de visualização é $dist_{max} = 4.0$. Um exemplo da diferença na visualização devido a mudança do valor de erro máximo pode ser visto na Figura 4.4, onde temos em (a) um valor de erro máximo $dist_{max} = 1.0$ e $dist_{max} = 6.0$ em (b).

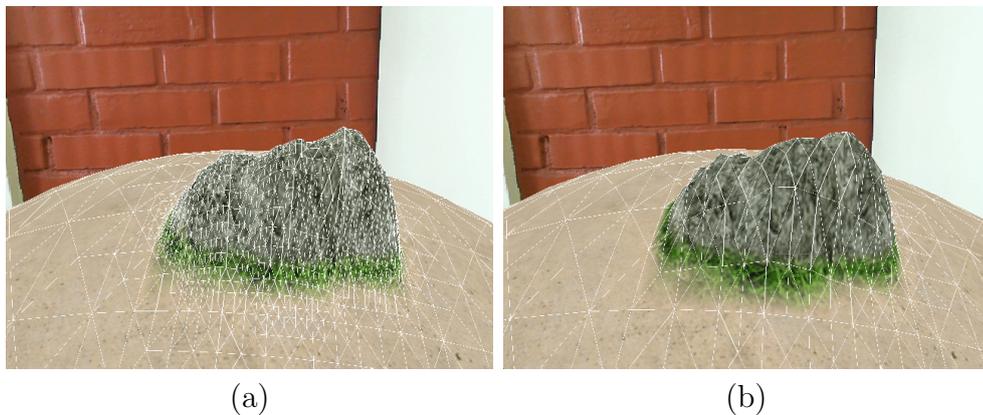


Figura 4.4: Resultado da tesselação para valores diferentes de erro máximo. Em (a) o valor de erro utilizado foi $dist_{max} = 1.0$ e em (b) $dist_{max} = 6.0$.

O raio de deformação r_d utilizado durante os testes variou entre 150km e 600km, sendo o raio do terreno esférico igual a 6372.79km. A diferença na área de deformação no espaço \mathbb{R}^3 para três raios de deformação diferentes pode ser vista na Figura 4.5, onde a Figura 4.5a mostra a área para um raio de 150km, a Figura 4.5b mostra a área para o raio de 300km e na Figura 4.5c é possível visualizar a área para um raio de 600km.

O tempo médio que o aplicativo demora para renderizar um quadro com o terreno sendo deformado foi calculado para diferentes resoluções. Os dados de entrada do método aumentam diretamente com a resolução do terreno, para um mesmo raio de deformação

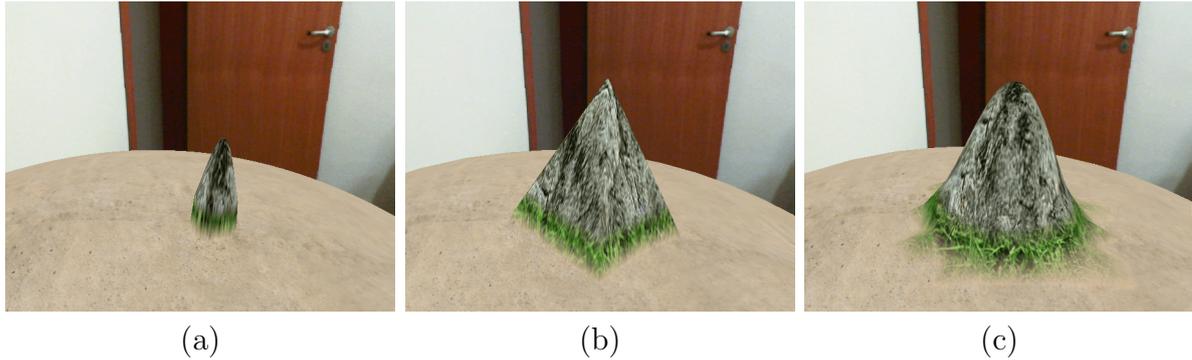


Figura 4.5: Visualização de diferentes áreas de deformação. O valor do raio de deformação utilizado em (a) foi $r_d = 150\text{km}$, em (b) $r_d = 300\text{km}$ e em (c) $r_d = 600\text{km}$.

r_d . Se a resolução for muito alta, pode ser inviável deformar regiões que possuem alta concentração de vértices (polos) em tempo real. Isso ocorre porque o método proposto sempre realiza a propagação de valores de altura e erro até o maior nível de resolução possível. Seria possível, no entanto, adaptar o método para propagar valores de altura e erro apenas em determinados níveis de resolução.

A câmera foi fixada em uma posição única para todos os valores contidos nas Tabelas 4.1, 4.2 e 4.3. A posição escolhida esteve a uma distância de três vezes o raio da esfera para o centro da mesma, ou seja, aproximadamente 19000km . No caso da deformação, foi escolhido o pior caso para avaliar o método. Em outras palavras, a longitude foi fixada em 180° , onde os vértices apresentam coordenada $u = 0$ e $u = m_u$ para valores variáveis de latitude. Os valores de latitude começam em 0° e aumentam 30° até atingirem 180° , ou seja, de um polo ao outro. O tempo médio para renderizar um quadro sem deformações sendo aplicadas ao terreno foi de 0.033 segundos.

Os valores na Tabela 4.1 representam os tempos médios de renderização de um quadro com o terreno sendo deformado para uma resolução máxima de 2049×2049 vértices. É possível afirmar que para um raio de deformação $r_d = 150\text{km}$, qualquer região do planeta pode ser deformada sem que o usuário perceba queda no número de quadros por segundo. Para um raio de $r_d = 600\text{km}$, o aplicativo demora três vezes mais tempo para renderizar um quadro com deformação ocorrendo no polo do que no equador, porém o número de vértices deformados no polo é aproximadamente 90 vezes maior do que no equador.

A Tabela 4.2 mostra os mesmos valores agora para uma resolução de 4097×4097 vértices. É possível perceber que a mesma situação da resolução anterior se repete, o aumento no número de vértices deformados é muito maior que o aumento no tempo

Latitude	Tempo médio(s)			Latitude	Vértices modificados		
0	0.064	0.096	0.117	0	32764	63471	126700
30	0.041	0.047	0.051	30	698	2789	11467
60	0.041	0.047	0.044	60	397	1593	6650
90	0.041	0.046	0.043	90	360	1431	5737
120	0.038	0.044	0.050	120	405	1629	6560
150	0.045	0.047	0.050	150	704	2916	11543
180	0.063	0.086	0.116	180	32759	63463	126888
	$r_d = 150$	$r_d = 300$	$r_d = 600$				
	Raio de Deformação(km)						

Tabela 4.1: Tempo médio de renderização do terreno em segundos quando uma quantidade de vértices é modificada em uma determinada latitude. Resolução máxima de 2049×2049 vértices.

médio de renderização. Isso significa que, o método é eficiente para lidar com o aumento no número de vértices a medida que a resolução aumenta. No caso dos polos, porém, a quantidade de vértices é suficiente para fazer o tempo médio de renderização aumentar de maneira significativa.

Latitude	Tempo médio(s)			Latitude	Vértices modificados		
0	0.117	0.211	0.368	0	126955	253855	503383
30	0.043	0.050	0.067	30	2808	11356	45087
60	0.043	0.046	0.048	60	1672	6540	25920
90	0.041	0.043	0.048	90	1403	5660	22484
120	0.041	0.043	0.049	120	1627	6518	26389
150	0.044	0.047	0.065	150	2756	11267	45102
180	0.116	0.203	0.369	180	126959	253755	503400
	$r_d = 150$	$r_d = 300$	$r_d = 600$				
	Raio de Deformação(km)						

Tabela 4.2: Tempo médio de renderização do terreno em segundos quando uma quantidade de vértices é modificada em uma determinada latitude. Resolução máxima de 4097×4097 vértices.

Na última tabela 4.3, são listados os valores para maior resolução de terreno testada, 8193×8193 vértices. Para um raio de deformação $r_d = 600$ percebe-se que a tempo médio começa a ficar elevado para todas as latitudes testadas. Isso significa que, a medida que a resolução aumenta é necessário reduzir o raio de forma a trocar desempenho do aplicativo e qualidade de visualização. Para um raio de deformação $r_d = 150$, os valores de tempo médio se mantém baixos, exceto para os polos que apresentam um número elevado de vértices mesmo nesse raio.

Para lidar com o problema da concentração de vértices nos polos, a medida que a

resolução cresce, duas soluções podem ser apresentadas. Uma primeira solução seria utilizar uma projeção que possui uma compensação diferente entre forma e área do que a projeção ortográfica utilizada nesta dissertação. É possível citar como exemplo a projeção pseudo-cilíndrica de Mollweide, que reduziria a concentração de vértices próximos aos polos (SNYDER, 1987).

Outra solução seria diminuir a raio de deformação, ou limitar o número máximo de vértices modificados na deformação, de acordo com a resolução. No caso da diminuição do raio, o usuário pode modificá-lo para um valor que faça o tempo médio de renderização ficar baixo. Já no caso de limitar o número de vértices, o raio de deformação teria um valor máximo de acordo com esse número.

Latitude	Tempo médio(s)			Latitude	Vértices modificados		
0	0.356	0.711	1.328	0	507744	1006861	2013967
30	0.052	0.053	0.114	30	11302	45437	180019
60	0.045	0.052	0.098	60	6532	25752	106635
90	0.035	0.047	0.082	90	5765	23056	89330
120	0.046	0.051	0.083	120	6511	26024	101521
150	0.045	0.059	0.107	150	11329	45625	179929
180	0.350	0.720	1.333	180	507805	1007216	2013721
	$r_d = 150$	$r_d = 300$	$r_d = 600$				
	Raio de Deformação(km)						

Tabela 4.3: Tempo médio de renderização do terreno em segundos quando uma quantidade de vértices são modificados em uma determinada latitude. Resolução máxima de 8193×8193 vértices.

4.3 PARTICULARIDADES

Um dos fatores relevantes para deformar o terreno é a calibragem do desvio padrão na função de deformação $g(x)$ calculada pela Equação 3.1. Como mencionado na Seção 3.4.1, valores muito pequenos ou grandes do desvio padrão podem provocar um aspecto indesejável na deformação do terreno, como pode-se observar na Figura 4.6. O resultado da deformação para $g(x)$ com um valor de desvio padrão $\sigma = 0.17$ (Fig. 4.6a) e o resultado para $\sigma = 0.75$ (Fig. 4.6b). A média utilizada em ambos os caso foi $\mu = 1$.

Outro fator importante mencionado na Seção 3.3.2 é o tratamento de bordas. Para demonstrar como o método lida com os vértices que se encontram no meridiano considere a Figura 4.7. Na Figura 4.7a a deformação sem o tratamento de bordas, na Figura 4.7b após o tratamento.

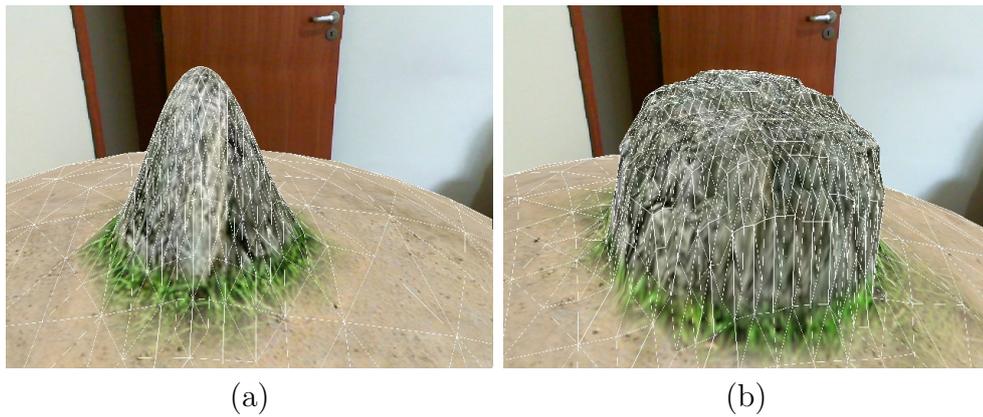


Figura 4.6: Deformação de terrenos utilizando a função de distribuição Gaussiana. A deformação obtida para um valor de $\sigma = 0.17$ e média $\mu = 1$ pode ser visto em (a). Em (b), $\sigma = 0.75$ e $\mu = 1$.



Figura 4.7: Tratamento de bordas. Em (a) é possível ver o terreno deformado na região da borda sem o tratamento. Em (b) a mesma região do terreno, agora deformada com tratamento de borda.

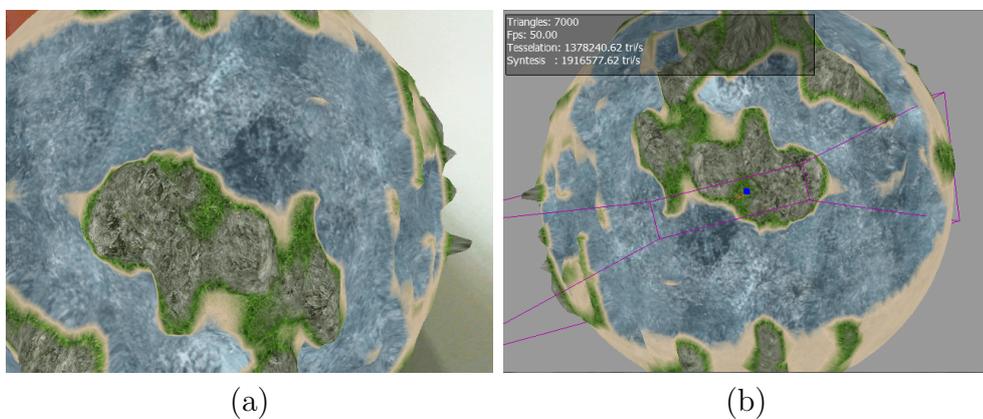


Figura 4.8: Visualização de um terreno final gerado através do método proposto. Escala local em (a) e global em (b).

Um planeta gerado com o método proposto é apresentado na Figura 4.8. Este terreno foi modelado em menos de três minutos. A implementação do método proposto consegue lidar com deformações em tempo real mesmo se a velocidade de rotação do planeta for definida com um valor elevado. É possível afirmar que a interface final é fácil de ser manipulada e o processo de deformação é intuitivo para o usuário que esculpe o terreno em tempo real.

5 CONCLUSÃO

Sistemas proporcionando liberdade real em três dimensões para modelagem são raros. Nesta dissertação, foi proposto um método para modelagem 3D de terrenos esféricos em tempo real através de realidade aumentada. Todo o processo é fácil de ser controlado pelo usuário. Apesar disso, os terrenos gerados podem ser bastante complexos, com aparência natural, mesmo que estes sejam definidos através de um pequeno número de ações do usuário.

O método alcançou os objetivos propostos e se mostrou eficiente para lidar com diversas resoluções. O aumento no tempo médio de renderização é menor do que o aumento no número de vértices deformados quando a resolução aumenta para todos os testes realizados. A propagação hierárquica permite que todos os vértices em uma área de deformação sejam visitados cumprindo com a restrição de monotonicidade da métrica de erro utilizada.

A utilização do marcador físico, para deformar a malha, fornece uma interface intuitiva para criar formas e padrões através do terreno. Essa interface permite uma modelagem fácil de superfícies sem que o usuário necessite de conhecimentos prévios em *softwares* de modelagem 3D como *Maya* e *3D Studio*. A visualização e a modelagem são feitas de maneira sequencial, no entanto o aplicativo funciona em tempo real para diversas resoluções de malha. Isso significa que terrenos grandes, porém detalhados, podem ser gerados com o método proposto.

Como mencionado anteriormente, não são muito comuns os aplicativos que realizam a deformação de um terreno formado por milhares de triângulos em tempo real através de realidade aumentada. A combinação de técnicas conhecidas com a estrutura da malha A4-8 produziu os resultados desejados, sendo possível, não só gerar um terreno com poucos movimentos intuitivos, mas também adicionar detalhes em qualquer resolução.

Uma limitação deste trabalho é a resolução do equipamento utilizado. Dependendo da câmara, o volume de visualização pode ser pequeno para criar um terreno e manipulá-lo. Mesmo que o método de deformação dependa do tamanho do conjunto de dados de entrada, é possível modificar o limiar de erro máximo para que a quantidade de quadros por segundo na visualização do terreno seja aumentada.

Como trabalhos futuros é possível mencionar as melhorias do método mencionadas na

seção de resultados e a aplicação do método em diferentes topologias, como a do toro. A rotina de expansão e todos os cálculos são realizados atualmente pelo processador. Estes cálculos poderiam ser feitos de forma mais eficiente se processados em GPU.

O trabalho apresentado nesta dissertação começou como um aplicativo para deformação de terrenos de topologia plana através de realidade aumentada tangível. Ao longo do tempo, o trabalho evoluiu para um aplicativo que permitia a construção de ambientes virtuais através dos terrenos gerados. Esta evolução proporcionou uma publicação no *Journal on 3D Interactive Systems* (JIS) 2012 intitulada *Interactive Virtual Terrain Generation using Augmented Reality Markers*. A dissertação aqui apresentada expandiu essa publicação em vários aspectos, permitindo a geração e modelagem de terrenos de topologia esférica, melhorando a estrutura de dados utilizada e aplicando diversas técnicas de visualização de terrenos.

REFERÊNCIAS

- AKLEMAN, E.; CHEN, J. Regular meshes. In: **Proceedings of the 2005 ACM symposium on Solid and physical modeling**, 2005. (SPM '05), p. 213–219. ISBN 1-59593-015-9. Disponível em: <<http://doi.acm.org/10.1145/1060244.1060268>>.
- AKLEMAN, E.; SRINIVASAN, V.; MANDAL, E. Remeshing schemes for semi-regular tilings. In: **Proceedings of the International Conference on Shape Modeling and Applications 2005**, 2005. (SMI '05), p. 44–50. ISBN 0-7695-2379-X. Disponível em: <<http://dx.doi.org/10.1109/SMI.2005.40>>.
- BARAFF, D.; WITKIN, A. Large steps in cloth simulation. In: **Proceedings of the 25th annual conference on Computer graphics and interactive techniques**, 1998. (SIGGRAPH '98), p. 43–54. ISBN 0-89791-999-8. Disponível em: <<http://doi.acm.org/10.1145/280814.280821>>.
- BARR, A. H. Global and local deformations of solid primitives. In: **Proceedings of the 11th annual conference on Computer graphics and interactive techniques**, 1984. (SIGGRAPH '84), p. 21–30. ISBN 0-89791-138-5. Disponível em: <<http://doi.acm.org/10.1145/800031.808573>>.
- BILLINGHURST HIROKAZU KATO, I. P. M. **Collaboration with Tangible Augmented Reality Interfaces**. 2001. 234-241 p.
- BOWMAN, D.; COQUILLART, S.; FROEHLICH, B.; HIROSE, M.; KITAMURA, Y.; KIYOKAWA, K.; STUERZLINGER, W. 3d user interfaces: New directions and perspectives. **Computer Graphics and Applications, IEEE**, v. 28, n. 6, p. 20–36, nov.-dec. 2008. ISSN 0272-1716.
- BOWMAN, D. A.; FROHLICH, B. New directions in 3d user interfaces. In: **Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality**, 2005. (VR '05), p. 312–. ISBN 0-7803-8929-8. Disponível em: <<http://dx.doi.org/10.1109/VR.2005.58>>.
- BOWMAN, D. A.; KRUIJFF, E.; LAVIOLA, J. J.; POUPYREV, I. **3D User Interfaces: Theory and Practice**, 2004. ISBN 0201758679.

- COQUILLART, S. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In: **Proceedings of the 17th annual conference on Computer graphics and interactive techniques**, 1990. (SIGGRAPH '90), p. 187–196. ISBN 0-89791-344-2. Disponível em: <<http://doi.acm.org/10.1145/97879.97900>>.
- DO, T. V.; LEE, J.-w. 3darmodeler : a 3d modeling system in augmented reality environment. **Systems Engineering**, v. 4, p. 2, 2010.
- DUCHAINEAU, M. A.; WOLINSKY, M.; SIGETI, D. E.; MILLER, M. C.; ALDRICH, C.; MINEEV-WEINSTEIN, M. B. Roaming terrain: real-time optimally adapting meshes. In: **IEEE Visualization**, 1997. p. 81–88.
- FAETH, A.; HARDING, C. Supporting Interactive Haptic Shaping of 3D Geologic Surfaces with Deformation Property Painting. In: **Eurographics 2009**, 2009. p. 11–17. Disponível em: <<http://diglib.eg.org/EG/DL/conf/EG2009/areas/011-017.pdf>>.
- FOSKEY, M.; OTADUY, M. A.; LIN, M. C. Artnova: touch-enabled 3d model design. In: **ACM SIGGRAPH 2005 Courses**, 2005. (SIGGRAPH '05). Disponível em: <<http://doi.acm.org/10.1145/1198555.1198619>>.
- GREGORY, A. D.; EHMANN, S. A.; LIN, M. C. intouch: Interactive multiresolution modeling and 3d painting with a haptic interface. In: **Proc. of IEEE VR Conference**, 1999. p. 45–52.
- IGARASHI, T.; MATSUOKA, S.; TANAKA, H. Teddy: a sketching interface for 3d freeform design. In: **ACM SIGGRAPH 2007 courses**, 2007. (SIGGRAPH '07). Disponível em: <<http://doi.acm.org/10.1145/1281500.1281532>>.
- KHODAKOVSKY, A.; SCHRÖDER, P.; SWELDENS, W. Progressive geometry compression. In: **Proceedings of the 27th annual conference on Computer graphics and interactive techniques**, 2000. (SIGGRAPH '00), p. 271–278. ISBN 1-58113-208-5. Disponível em: <<http://dx.doi.org/10.1145/344779.344922>>.
- KOBBELT, L. P.; BAREUTHER, T.; SEIDEL, H. P. Multiresolution shape deformations for meshes with dynamic vertex connectivity. In: **Proceedings of Eurographics 2000**, 2000. v. 19.

- LINDSTROM, P.; PASCUCCI, V. Visualization of large terrains made easy. In: **Proceedings of the conference on Visualization '01**, 2001. (VIS '01), p. 363–371. ISBN 0-7803-7200-X. Disponível em: <<http://dl.acm.org/citation.cfm?id=601671.601729>>.
- NEALEN, A.; MUELLER, M.; KEISER, R.; BOXERMAN, E.; CARLSON, M. Physically Based Deformable Models in Computer Graphics. **Computer Graphics Forum**, Blackwell Publishing, v. 25, n. 4, p. 809–836, dez. 2006. ISSN 0167-7055. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659.2006.01000.x>>.
- NOEL. **3D Modeller @ONLINE**. 2010. Disponível em: <<http://community.secondlife.com/t5/Building/Art-of-Illusion-and-Sculpted-Prims/td-p/430040>>.
- O'BRIEN, J. F.; HODGINS, J. K. Graphical modeling and animation of brittle fracture. In: **Proceedings of the 26th annual conference on Computer graphics and interactive techniques**, 1999. (SIGGRAPH '99), p. 137–146. ISBN 0-201-48560-5. Disponível em: <<http://dx.doi.org/10.1145/311535.311550>>.
- PATANÈ, G.; SPAGNUOLO, M. Triangle mesh duality: Reconstruction and smoothing. In: WILSON, M. J.; MARTIN, R. R. (Ed.). **Mathematics of Surfaces, 10th IMA International Conference, Leeds, UK, September 15-17, 2003, Proceedings**, 2003. (Lecture Notes in Computer Science, v. 2768), p. 111–128. ISBN 3-540-20053-3.
- PENG, J.; STRELA, V.; ZORIN, D. A simple algorithm for surface denoising. In: **Proceedings of the conference on Visualization '01**, 2001. (VIS '01), p. 107–112. ISBN 0-7803-7200-X. Disponível em: <<http://dl.acm.org/citation.cfm?id=601671.601686>>.
- SANTHANAM, A. P.; FIDOPIASTIS, C. M.; HAMZA-LUP, F. G.; ROLLAND, J. P.; IMIELINSKA, C. Physically-based deformation of high-resolution 3d models for augmented reality based medical visualization. In: **MICCAI -AMI-ARCS International Workshop On Augmented Environments For Medical Imaging and Computer-aided Surgery 2004 , Rennes/St.Malo France**, 2004.
- SEDERBERG, T. W.; PARRY, S. R. Free-form deformation of solid geometric models. In: **Proceedings of the 13th annual conference on Computer graphics and interactive techniques**, 1986. (SIGGRAPH '86), p. 151–160. ISBN 0-89791-196-2. Disponível em: <<http://doi.acm.org/10.1145/15922.15903>>.

- SHA, C.; LIU, B.; MA, Z.; ZHANG, H. Multi-resolution meshes deformation based on pyramid coordinates. In: **CGIV**, 2007. p. 200–204.
- SINGH, K.; FIUME, E. Wires: a geometric deformation technique. In: **Proceedings of the 25th annual conference on Computer graphics and interactive techniques**, 1998. (SIGGRAPH '98), p. 405–414. ISBN 0-89791-999-8.
- SNYDER, J. P. **Map projections – A working manual**, 1987. (U. S. G. S. Professional Paper, 1395). Disponível em: <<http://pubs.er.usgs.gov/usgspubs/pp/pp1395>>.
- STOCKMAN, G.; SHAPIRO, L. G. **Computer Vision**. 1st. ed., 2001. ISBN 0130307963.
- SUMNER, R. W.; SCHMID, J.; PAULY, M. Embedded deformation for shape manipulation. **ACM Trans. Graph**, v. 26, p. 80, 2007.
- VELHO, L.; GOMES, J. Variable resolution 4-k meshes: Concepts and applications. **Comput. Graph. Forum**, v. 19, n. 4, p. 195–212, 2000.
- ZORIN, D.; SCHRÖDER, P. **Subdivision for Modeling and Animation**, 2000. (Course Notes for SIGGRAPH 2000, 23).