

2

Conceitos Básicos e Trabalhos Relacionados

Neste capítulo apresentamos, resumidamente, alguns conceitos básicos necessários para o entendimento do trabalho. Uma descrição mais detalhada destes conceitos pode ser encontrada em [4]. Depois, segue-se uma seção resumindo alguns outros trabalhos importantes encontrados na literatura e relacionados ao problema principal tratado neste trabalho que é a visualização interativa de terrenos.

2.1.

Conceitos Básicos

Os conceitos básicos considerados importantes para o entendimento deste trabalho tratam da representação e reconstrução de terrenos, métricas de erro, requisitos de malhas, construção da hierarquia de malhas, representação por níveis de detalhes, descarte de regiões invisíveis e transições suaves na geometria. A seguir, apresentamos, brevemente, cada um destes conceitos.

2.1.1.

Representação e Reconstrução de Terrenos

A geometria de um terreno é normalmente representada de forma discreta, ou seja, guardando algumas amostras que representem o terreno. Neste caso, cada amostra contém suas coordenadas x , y e z , onde z representa a altura do terreno naquele ponto (daí a nomenclatura “campos de altura”, comumente utilizada para se referir a este tipo de dado). Estas amostras podem ser regularmente espaçadas, de modo que todas juntas formem uma grade, ou podem ser irregularmente espaçadas.

A reconstrução do terreno a partir da representação discreta mencionada acima e tratada neste trabalho consiste em gerar uma aproximação através de uma malha de polígonos (geralmente triângulos, onde cada vértice corresponde a uma

amostra). Finalmente, esta malha é coberta com alguma textura que represente a radiação luminosa dos materiais do terreno para uma dada iluminação.

Com amostras irregularmente espaçadas é possível obter maior eficiência em termos de armazenamento. Neste caso, é possível adequar o número de amostras à geometria do terreno, ou seja, guardar um número maior de amostras para representar as regiões mais acidentadas e um número menor de amostras para representar as regiões menos acidentadas do terreno. No caso das amostras regularmente espaçadas isto não é possível e, normalmente, acontece uma redundância nas regiões mais planas do terreno. Entretanto, não ter os dados dispostos em uma grade implica não poder tirar vantagem de estruturas de dados simples e de rápido acesso como árvores quaternárias e matrizes. Na prática, isto implica ter algoritmos bem mais complexos para a geração da malha do terreno.

Por questão de simplicidade, atualmente, a maioria das representações de terrenos utiliza grades regulares. Assim, o algoritmo estudado neste trabalho baseia-se nesta representação. Na Figura 1 apresentamos uma malha construída pelo algoritmo descrito neste trabalho. Repare que ela, de fato, é construída sobre uma grade regular e pode ser enxergada através de quatro árvores binárias de triângulos. Conforme será visto mais adiante, na verdade esta malha aproximada é construída executando-se um procedimento recursivo em cada uma destas quatro árvores binárias de triângulos.

2.1.2. Métricas de Erro

Gerar a malha completa, contendo todas as amostras do terreno, se torna impraticável para terrenos de grande ou mesmo médio porte. Por isso, os algoritmos dependentes da visão procuram gerar uma aproximação que contenha o menor número de triângulos e que, ao mesmo tempo, mantenha a melhor qualidade da imagem.

Enfim, como estamos falando em malhas aproximadas, então é necessário ter meios para medir o quanto esta aproximação se distancia da malha original. Na literatura, existem diferentes propostas de métricas de erro, porém a métrica mais comumente usada consiste em medir a diferença entre a altura de uma amostra do terreno e a altura do ponto correspondente na malha aproximada e, depois,

projetar esta medida no espaço da tela. Geralmente, isto é computado apenas nos vértices, mas poderia ser computado sobre triângulos ou mesmo sobre a região de influência de um vértice.

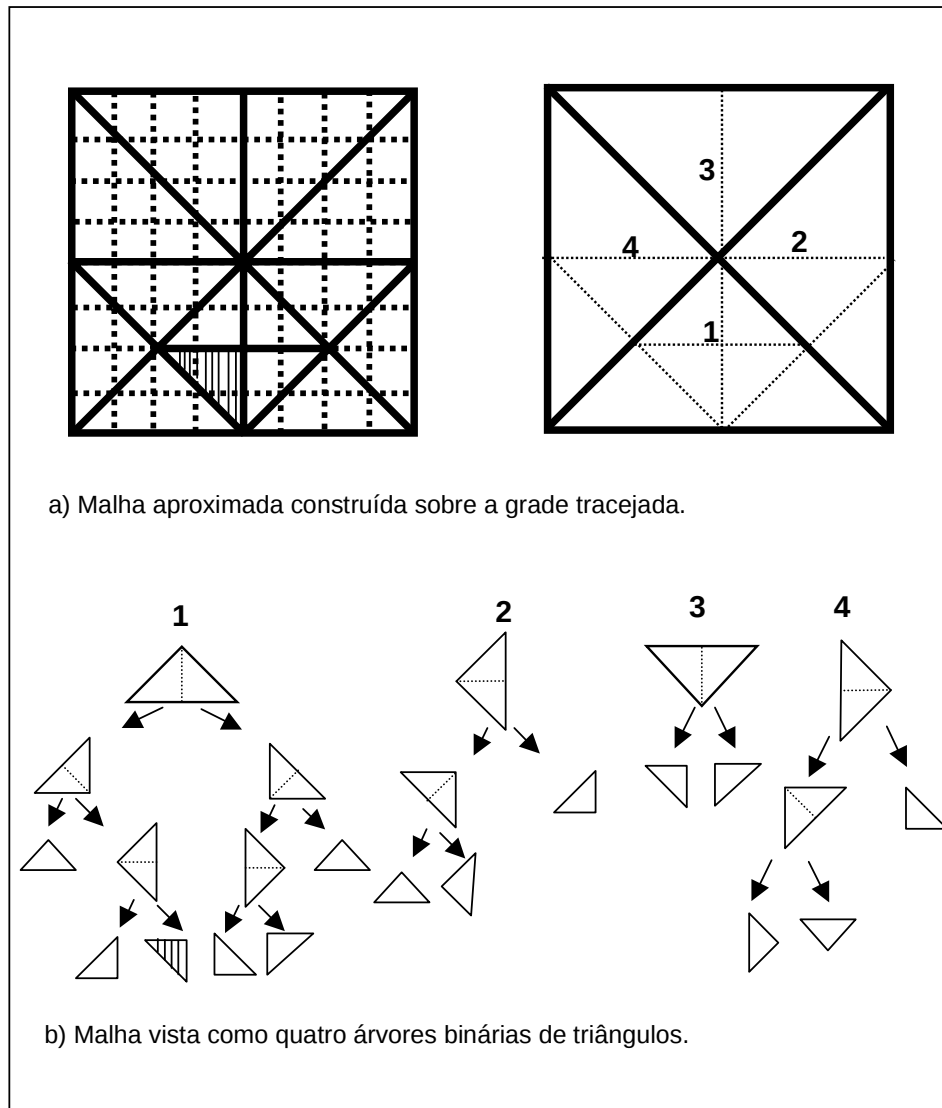


Figura 1 - Exemplo de Malha produzida pelo algoritmo em estudo.

Além disso, pode-se medir o erro em termos absolutos, isto é, a distância entre a malha aproximada atual e a malha de resolução máxima, ou pode-se escolher medir um erro relativo, ou seja, entre níveis consecutivos de resolução da malha.

De antemão, podemos dizer que, no algoritmo discutido neste trabalho, a escolha de qual métrica de erro utilizar é ortogonal ao algoritmo de construção da

malha a cada quadro. Além disso, Lindstrom & Pascucci analisam estes dois tipos de erro. Segundo eles, de fato, usando o erro absoluto ou erro máximo, conforme nomeado por eles, produz-se uma malha com um número um pouco maior de triângulos e que portanto representa mais fielmente o terreno. Contudo, computacionalmente é bem mais caro trabalhar com erro absoluto e a diferença entre o número de triângulos gerados não é grande. Assim, considerando a relação custo-benefício, eles sugerem trabalhar com erro incremental. Por isso, em nossa implementação e nossos resultados apresentados neste trabalho, estamos utilizando sempre o erro relativo ou incremental.

Na Figura 2, tentamos ilustrar com um exemplo simples e unidimensional a diferença entre erro absoluto ou máximo e erro relativo ou incremental. Note que a figura procura destacar estes dois tipos de erros no ponto 1 e para o nível de resolução 0.

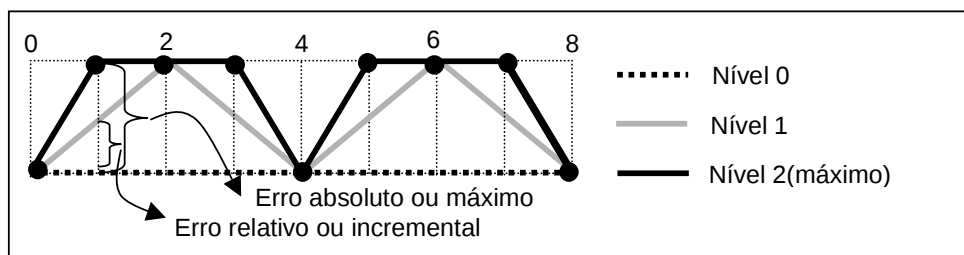


Figura 2 - Erro absoluto e erro relativo.

2.1.3. Malha Válida

Uma malha pode ser definida como um conjunto de faces e, no nosso contexto, ela será considerada válida ou consistente se não contiver vértices T nem arestas sem face. Em vez disso, toda aresta será uma fronteira entre duas e somente duas faces diferentes. Neste caso, consideramos a região exterior à malha como uma face.

A Figura 3a ilustra um vértice T visto de cima. Já a Figura 3b ilustra o vértice T visto em perspectiva, quando a fenda ou descontinuidade provocada por ele se torna visível. Finalmente, a Figura 3c ilustra uma aresta que não é fronteira entre duas faces e que, portanto, é uma aresta inválida. O esquema proposto por Lindstrom & Pascucci sempre gera malhas válidas.

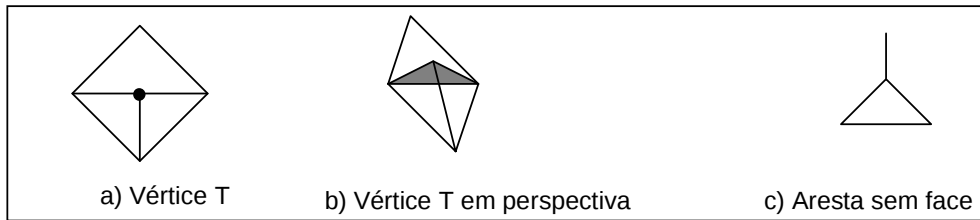


Figura 3 - Exemplo de vértice T e aresta sem face.

2.1.4. Construção da Hierarquia de Malhas

A hierarquia de malhas permite extrair malhas de complexidades diferentes para representar o mesmo terreno. A construção desta hierarquia, porém, pode acontecer através de dois processos distintos e inversos: simplificação e refinamento. Veja a representação deste dois processos na Figura 4. No processo de simplificação o algoritmo trabalha no sentido de baixo para cima. Portanto, ele parte da malha de resolução máxima e vai combinando triângulos enquanto o erro não excede uma tolerância especificada pelo usuário. Já no processo de refinamento, o algoritmo trabalha no sentido de cima para baixo. Neste caso, ele parte de uma malha grosseira e vai dividindo triângulos enquanto o erro for maior que uma tolerância especificada.

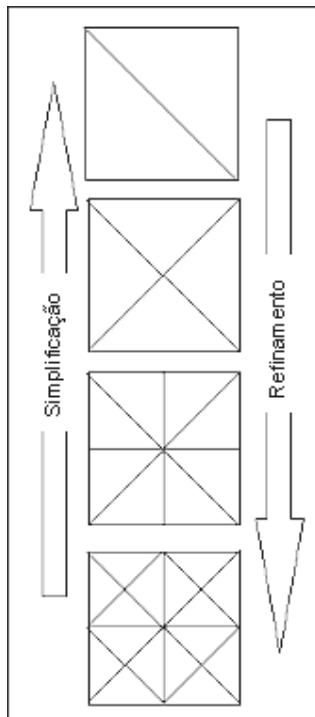


Figura 4 - Seqüência de malhas da mesma superfície.

Como os algoritmos de simplificação sempre partem da malha de resolução máxima em direção à malha de saída e os algoritmos de refinamento da malha de resolução mínima em direção à malha de saída, então a complexidade computacional dos algoritmos de simplificação é sempre diretamente proporcional à malha de resolução máxima, enquanto nos algoritmos de refinamento esta complexidade é proporcional à malha de saída. Portanto, como a malha de saída costuma ser muito menor que a malha de resolução máxima, os algoritmos de refinamento também costumam ser mais rápidos.

Por outro lado, os algoritmos de simplificação costumam gerar malhas com menos triângulos. A razão disso é que eles começam analisando o erro em todos os pontos, enquanto os algoritmos de refinamento precisam usar alguma aproximação ou artifício para iniciar analisando o erro em grandes regiões de uma só vez.

Vários algoritmos, por exemplo aqueles citados em [4], [5] e [9], em tempo de pré-processamento, dividem o terreno em blocos e utilizam uma estrutura de árvores quaternárias para organizar estes blocos hierarquicamente. Em tempo real, a construção da malha aproximada será feita percorrendo tal árvore quaternária.

Contudo, conforme será visto mais a frente, o algoritmo deste trabalho dispensa este modelo explícito de árvores quaternárias, embora, de certo modo, seja inspirado nele.

2.1.5.

Representação por Níveis de Detalhes Dependente da Visão

Uma representação por níveis de detalhes, ou seja, usando uma estrutura de multi-resolução, requer o uso simultâneo de diferentes escalas de representação para o mesmo objeto, que no nosso caso trata-se de um terreno. Isto permite dar maior destaque ao que está mais próximo do observador e regiões menos uniformes do objeto, representando-os com mais detalhes (maior resolução ou maior número de triângulos). Por outro lado, o que está mais distante do observador e áreas mais uniformes podem ser representados com menos detalhes (menor resolução ou menor número de triângulos) sem que o usuário perceba degradação na qualidade visual da imagem.

Entretanto, se não houver uma transição suave entre diferentes níveis de detalhes (ou resolução), vértices T e conseqüentes fendas aparecerão nas fronteiras entre os diferentes níveis. Então, para garantir uma malha final sem vértices T e fendas, é preciso um tratamento especial nas fronteiras entre diferentes níveis de resolução. Veja as figuras 5a e 5b.

Nos últimos anos, muito esforço de pesquisa tem sido dedicado à geração eficiente de modelos de multi-resolução, empregados não apenas para a visualização de terrenos mas, em geral, para quaisquer objetos ou cenas tridimensionais complexos. Alguns autores propõem esquemas complexos e que requerem um considerável tempo de pré-processamento.

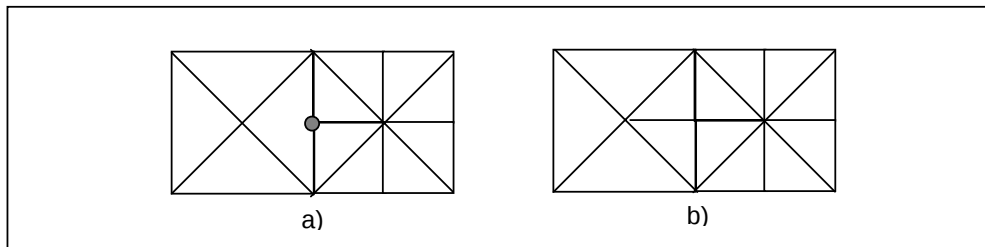


Figura 5 - a) Vértice T formado entre diferentes níveis de detalhes; b) Transição suave entre diferentes níveis de detalhes.

A Figura 6 ilustra um modelo de multi-resolução, iniciado com uma malha-base de dois triângulos e depois construído conforme um esquema de partição da maior aresta.

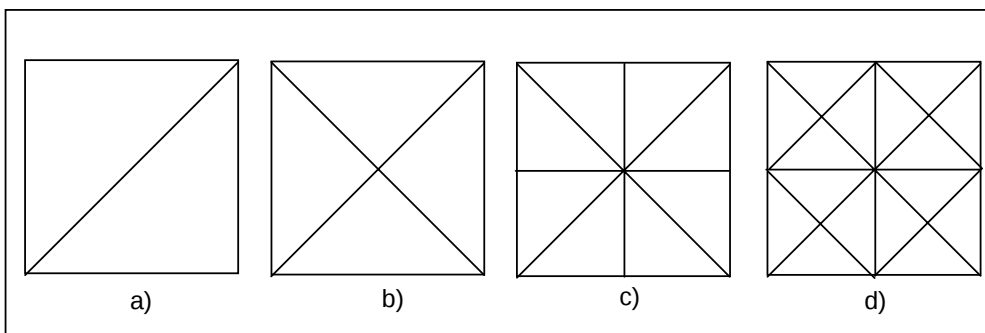


Figura 6 - Modelo de multi-resolução de uma superfície: a) nível 0; b) nível 1; c) nível 2 e d) nível 3.

2.1.6.

Descarte de Regiões Invisíveis

A idéia, neste caso, é que durante um vôle sobre um grande terreno, na maioria das vezes somente uma fração do terreno estará visível. Portanto, se todo o terreno for enviado para *renderização*, muito esforço computacional estará sendo desperdiçado. Por outro lado, se houver uma maneira rápida de identificar e descartar as regiões que caem fora do volume ou *frustum* de visão, o desempenho da aplicação pode ser consideravelmente melhorado.

Assim, este descarte rápido de regiões não visíveis, ou seja, o *culling* da malha fora do campo de visão, acaba se tornando imprescindível para as aplicações que desejam executar a visualização de terrenos relativamente grandes obtendo taxas interativas.

2.1.7.

Transições Suaves na Geometria

Muitas vezes, estabelecer um erro tolerável muito pequeno para a malha aproximada implica gerar um número de triângulos muito grande mesmo para os hardwares gráficos mais avançados. Ao aumentar o erro tolerável este problema é minimizado, mas aí ocorre o aparecimento de alguns artefatos temporais. Isto acontece porque cada inserção de vértice resulta em uma mudança brusca na geometria da ordem da tolerância (em *pixels*). O resultado na navegação sobre terrenos é que montanhas e picos do terreno ficam aparecendo ou desaparecendo bruscamente, produzindo uma sensação bastante desagradável.

Obviamente, na busca por realismo é crucial que estes artefatos visuais sejam eliminados. Afinal de contas, em um sobrevôo por um terreno de verdade, as montanhas e picos são vistos aparecendo ou desaparecendo de forma relativamente suave. Para simular isto, estes picos e montanhas devem ser desenhados com alturas intermediárias, de acordo com a distância do observador, como uma espécie de animação, comumente denominada *geomorphing*. Vários esquemas para fazer estas animações têm sido propostos, como em [4], [6] e [7]. Alguns, porém, baseados em tempo, necessitam guardar informações históricas e são bastante complexos para serem implementados.

Entretanto, o algoritmo discutido neste trabalho propõe um esquema de *geomorphing* bastante simples, baseado apenas na distância de cada ponto até o observador e que produz um resultado bastante satisfatório, conforme será visto mais adiante.

2.2. Trabalhos Relacionados

Nos últimos anos, vários pesquisadores têm se dedicado intensamente à área de visualização de terrenos e à criação e gerenciamento de níveis de detalhes para objetos 3D genéricos. Destacamos aqui apenas alguns trabalhos importantes e recentemente publicados na área de visualização de terrenos dependente da visão, em ordem cronológica.

Em [5] Lindstrom et al. propõem um algoritmo iterativo para *renderização* de terrenos cujos dados estão representados sobre uma grade regular e utilizando um limiar no espaço da tela para limitar o erro da imagem projetada na tela. Este algoritmo trabalha no sentido de baixo para cima, combinando triângulos e executando recursivamente um tipo de partição da maior aresta. Na verdade, uma simplificação num nível grosseiro é realizada para selecionar níveis de detalhes por bloco. Em seguida, vértices individuais, dentro do bloco selecionado, são analisados para remoção.

Em tempo de pré-processamento, os dados do terreno são organizados em blocos formando uma árvore quaternária. Entretanto, devem ser tomados cuidados especiais para assegurar que não haja vértices T e fendas entre blocos. Para isto, é mantido um esquema de dependências entre vértices. O problema é que este esquema é custoso, tanto em termos computacionais quanto de armazenamento. Além disso, a complexidade deste algoritmo independe da malha final gerada, pois, por ser baseado em simplificação, ele sempre começa com a malha completa (de resolução máxima).

Em [6] Hoppe propõe um algoritmo que é uma especialização de seus trabalhos anteriores para malhas de objetos tridimensionais genéricos aplicado ao caso de terrenos. Neste trabalho, Hoppe reconhece a importância da visualização de terrenos. O algoritmo deste trabalho é similar ao desenvolvido em [5], porém estendido para malhas irregulares. Além disso, é proposto um tipo de

geomorphing, ou seja, eliminação de artefatos visuais que explora a coerência temporal, suavemente interpolando a geometria ao longo do tempo. Um novo método de cálculo para a aproximação do erro em tempo de pré-processamento é sugerido. A questão é que, segundo ele, a aproximação do erro somente em pontos da grade é inadequada para níveis de detalhes dependentes da visão executado sobre malhas irregulares. O problema desta abordagem é a sua grande complexidade e dificuldade de implementação.

Em [7] Duchaineau et al. propuseram um algoritmo que sugere várias melhorias ao trabalho de Lindstrom et al.[5]. Assim como no algoritmo descrito em [5], o algoritmo de Duchaineau et al. é executado sobre uma grade uniforme. Porém, como eles organizam os dados em uma árvore binária, nenhum cuidado especial precisa ser tomado para evitar vértices T ou fendas na malha final aproximada. Além disso, o algoritmo explora a coerência entre quadros para melhorar a interatividade. Um esquema de *geomorphing* temporal é proposto e os requisitos de memória são similares aos de [5]. Por fim, o algoritmo também promete flexibilidade na escolha da métrica de erro a ser utilizada. Novamente, o problema desta abordagem é que ela ainda é considerada complexa e implementar todas as melhorias sugeridas neste trabalho não parece ser uma tarefa simples.

Em [4] Toledo desenvolve uma proposta que se caracteriza por ser uma combinação de duas estratégias conhecidas: malhas progressivas e árvores quaternárias. Em tempo de pré-processamento, o terreno é inicialmente dividido em uma grade de blocos, onde cada bloco contém um certo número de triângulos. A malha é simplificada, mantendo-se os vértices das fronteiras para evitar fendas em junções tipo T. A partir de um certo nível de simplificação, quatro blocos adjacentes no sentido da árvore quaternária são unidos em um único bloco, permitindo a eliminação dos vértices entre eles. Neste momento formam-se os primeiros nós internos da árvore quaternária. Estes nós, por sua vez, possuem como filhos as quatro malhas progressivas dos nós que os geraram. Este processo de baixo para cima prossegue até que se obtenha a raiz da árvore quaternária.

Os resultados obtidos foram considerados bons e atendem aos propósitos estabelecidos. Entretanto, a implementação desta proposta é complexa e a fase de pré-processamento é demorada.

Ainda inspirado no trabalho desenvolvido em [5], De Boer[8] propõe um algoritmo para visualização de terrenos dependente da visão que agora procura

tirar proveito das facilidades para a renderização 3D implementadas em hardware. Assim, utiliza-se uma abordagem baseada em blocos organizados em uma árvore quaternária. Deste modo, um tratamento especial deve ser executado para evitar vértices T e fendas entre blocos com diferentes níveis de resolução. O processamento geométrico e a rasterização sempre foram considerados os grandes gargalos nos algoritmos para visualização de terrenos. Desta forma, a idéia central deste trabalho parte do princípio de que agora o estágio de rasterização praticamente deixou de ser um gargalo e, portanto, a ordem é empurrar tantos triângulos para a rasterização quanto o hardware puder manipular. Assim, a idéia é fazer cálculos aproximados, de forma a reduzir a sobrecarga da CPU por conta do processamento geométrico. Esta abordagem é simples, porém não se preocupa, por exemplo, com o formato dos triângulos gerados e, por ser baseada em blocos, ainda exige tratamento especial para conseguir garantir uma malha final sem vértices T e fendas.

Finalmente, destacamos um trabalho bastante recente publicado em [9]. Este trabalho tem como objetivo atender às necessidades da visualização de terrenos em aplicações de realidade virtual, executando uma visualização estereoscópica dependente da visão. O problema da estereoscopia é que ela necessita dobrar a velocidade de geração do correspondente monoscópico para alcançar a mesma interatividade, já que neste caso é preciso produzir duas imagens, uma para o olho direito e outra para o olho esquerdo. O algoritmo, conforme tantos outros trabalhos, representa os dados em um modelo de multi-resolução baseado em árvores quaternárias e utiliza um esquema de partição da maior aresta. Assim, para evitar vértices T e fendas, é mantido um esquema de dependências entre vértices. A novidade é que, para não perder informações de profundidade, essenciais na visualização estereoscópica, adota-se um critério de simplificação diferente, ou seja, um limiar de erro angular baseado na distância. Além disso, as duas imagens, para o olho direito e para o olho esquerdo, são geradas simultaneamente. Tira-se proveito do fato de que estas duas imagens são muito parecidas, então operações como *culling* da malha fora do volume de visão são realizadas uma única vez. Com o intuito de melhorar a qualidade visual, uma operação de *geomorphing* também é incluída.

Vale ressaltar que a grande contribuição desse trabalho está nas idéias que incluem a estereoscopia. Se estas idéias fossem utilizadas mescladas ao algoritmo

principal descrito neste trabalho, provavelmente seriam obtidos resultados ainda melhores.