2
Trabalhos Relacionados

2.1
Terrenos em Multi-Resolução

Multi-resolução, comumente também referido pela sigla em inglês LOD (*Level of Detail*), consiste na técnica de reduzir a complexidade de um modelo tridimensional com o objetivo de aumentar a eficiência da visualização, mantendo a percepção visual do modelo.

Modelos de terreno possuem algumas particularidades que fazem com que o estudo de LOD nesses modelos evolua de forma paralela aos modelos de propósito geral. O fato de os dados de terrenos serem geralmente massivos em tamanho exige uma atenção na quantidade de geometrias renderizadas e no gerenciamento de memória principal e secundária. Alguns conceitos devem ser levados em consideração para a caracterização de um algoritmo de multi-resolução de terreno, entre eles: forma de representação de terreno, cálculo da medida de erro, algoritmo para a simplificação.

A representação de terrenos pode ser feita a partir de uma malha irregular de triângulos (geralmente referido pela sigla em inglês TIN - *Triangulated Irregular Network*), definida por conjuntos de três vértices devidamente posicionados para formar triângulos não sobrepostos. Porém a representação mais comum é a grade regular, que consiste em uma matriz que representa uma área onde cada posição da matriz possui um valor para a altura naquele ponto. Esse tipo de representação geralmente é armazenada utilizando uma imagem de uma banda de cor, onde a informação armazenada na banda representa a altura de cada pixel. A representação irregular pode prover uma melhor eficiência no armazenamento por evitar redundâncias comuns em malhas regulares, como por exemplo, uma grande área plana na malha irregular pode ser representada por poucos triângulos, enquanto na malha regular serão vários vértices com alturas iguais. De uma forma mais geral, a malha irregular utiliza apenas mais vértices quando a complexidade do terreno demanda. Apesar desse benefício da representação em TIN, a regularidade de terrenos pode ser explorada para facilitar o cálculo da simplificação, assim como a definição de uma hierarquia

PUC-Rio - Certificação Digital Nº 0721348/CB

para os níveis de detalhe.

Como nos algoritmos de multi-resolução o terreno que vai ser visualizado é uma aproximação do terreno original, é necessário que seja definida uma métrica para a distância da malha visualizada para a malha completa, e tal métrica deve ser utilizada no processo de decisão do nível de detalhe a ser carregado para uma dada posição do observador. Um exemplo de métrica de erro é 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.

Para discorrer sobre alguns algoritmos importantes de multi-resolução de terrenos, podemos citar primeiramente o algoritmo de Lindstrom (Lin96), que apresenta uma técnica em tempo real para o LOD contínuo de terrenos regulares. Exemplos e comentários sobre outros algoritmos de multi-resolução são encontrados nos trabalhos de Magalhães (Mag05), Souza (Sou03) e Toledo (Tol00).

O algoritmo em que este trabalho estará embasado é descrito na dissertação de mestrado de Magalhães (Mag05), onde ele utiliza o gerenciamento de ladrilhos através de uma *quadtree* para a multi-resolução. A organização da hierarquia impõe que os nós folhas armazenem a geometria sem nenhuma simplificação, e cada nó pai é formado por uma amostragem de 25% da quantidade de vértices dos seus filhos, após a simplificação. Esse critério causa uma interessante característica para o armazenamento do terreno, pois toda a estrutura simplificada demanda apenas 1/3 a mais de espaço do que o terreno original. O conteúdo dos nós simplificados são calculados em pré-processamento e a decisão de qual ladrilho será renderizado é feita durante a navegação utilizando a métrica do erro do ladrilho projetado na tela. Um cuidado adicional é necessário nas bordas dos ladrilhos carregados em níveis distintos de refinamento, pois a malha se torna inválida e deve ter os triângulos da região remontados. O comportamento do algoritmo provê eficiência na visualização do terreno e permite reduzir drasticamente o número de vértices dos ladrilhos afastados do observador. A Figura 2.1 mostra o exemplo de um terreno visualizado pelo algoritmo. A versão em *wireframe* nos dá a percepção de como o algoritmo funciona, pela densidade de vértices dos ladrilhos escolhidos.

PUC-Rio - Certificação Digital Nº 0721348/CB

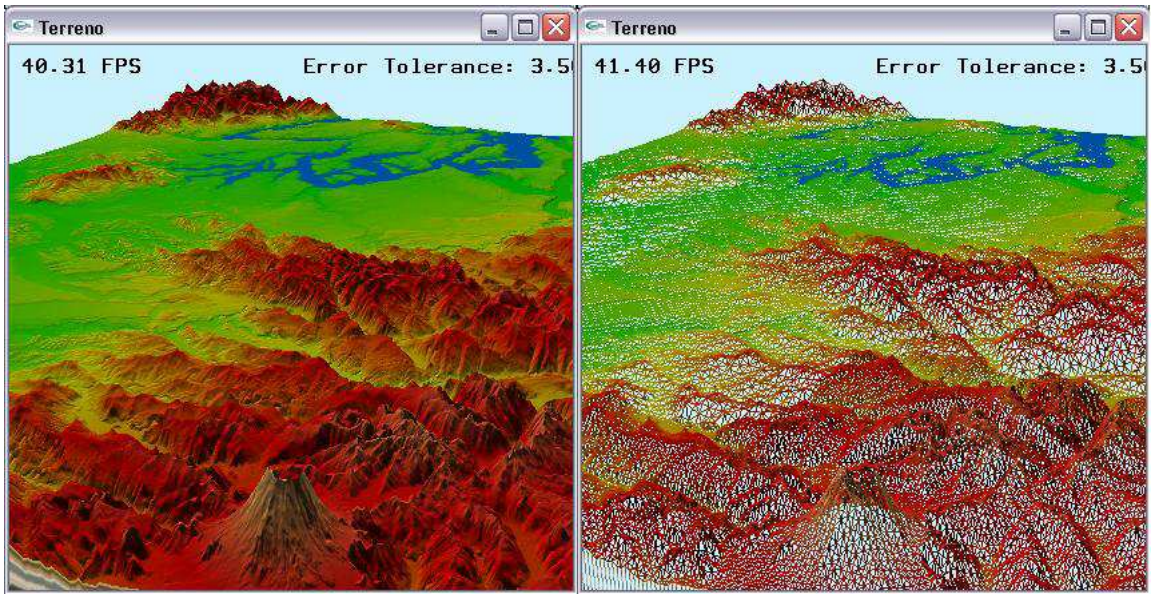

Figura 2.1: Visualização do LOD da dissertação de Magalhães (Mag05)

2.2 Correção Perspectiva em Mapas de Sombra

O algoritmo de mapas de sombra foi apresentado por Williams (Wil78) para a geração de sombras em cenas tridimensionais. Resumidamente, a proposta original consiste em desenhar a cena vista da posição da luz, e gerar uma textura de profundidade guardando o z mais próximo e aplicando na cena. A técnica, eficiente e relativamente simples, tem o revés de ter a qualidade depedente da posição relativa da luz e do observador. Na proposta original, a amostragem do mapa de sombras (textura) é igual para toda a cena e, à medida que a camera vai se movimentando pela cena, é perceptível a presença de *aliasing* por falta de amostragem.

O trabalho de Stamminger e Drettakis (Sta02) oferece uma técnica simples para a melhora visual do algoritmo de mapas de sombra original, chamado Perspective Shadow Maps (PSM). No PSM, os autores conseguiram uma grande melhora na qualidade da sombra gerando a textura de profundidade no espaço pós-perspectiva, transformando a cena e a fonte de luz pelas matrizes correspondentes, assim como na geração de coordenadas de textura. Isso garante maior quantidade de pixels no mapa de sombras para objetos próximos ao observador, dando maior detalhe a sombras próximas à camera.

A idéia conceitualmente simples do PSM gera, porém, certos casos extremos que dificultam sua implementação e causam situações onde a solução não é satisfatória. A transformação para o espaço pós-perspectiva pode mudar o tipo da fonte de luz (direcional para pontual, ou vice-versa) ou até forçar a

PUC-Rio - Certificação Digital Nº 0721348/CB

luz a ser tratada como uma "fonte de sombra", ou uma fonte de luz "invertida".

O trabalho de Wimmer et al. (Wim04) apresenta uma maneira de evitar os casos especiais da técnica original do PSM, gerando uma transformação conveniente que melhora a qualidade da visualização sem alterar as características originais da fonte de luz. O artigo salienta que, apesar de transformações perspectivas serem válidas para melhorar o aproveitamento do mapa de sombras, não é mandatório que essa transformação esteja vinculado à camera como no PSM. O trabalho define o que é chamado de "espaço da luz" que é um conceito didático para definir o espaço onde o mapa de sombras é gerado. Tratando todas as fontes de luz como direcionais, os eixos do espaço da luz são definidos levando em consideração a direção da luz e informações do volume de visão.

O espaço da luz é definido da seguinte forma: O eixo y é o vetor direção da luz $\vec{L}$ com sentido apontando para a fonte. O eixo z é perpendicular ao eixo y e pertence ao plano que contém os vetores direção da câmera $\vec{v}$ e direção da luz. Finalmente, o eixo x é o produto vetorial entre os vetores definidos anteriormente. A Figura 2.2 explica como os eixos são definidos.

PUC-Rio - Certificação Digital Nº 0721348/CB

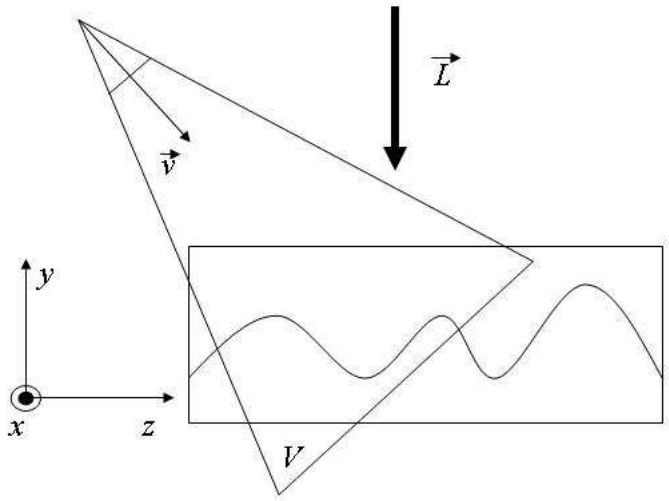

Figura 2.2: Definição dos eixos do espaço da luz

A sequência de passos do algoritmo de correção, incluindo o cálculo da matriz de transformação para a correção perspectiva, é descrita a seguir.

1. Define-se o poliedro B como sendo a interseção da caixa envolvente da cena com o fecho convexo dos pontos que definem o volume de visão mais o ponto onde está a fonte de luz. Esse poliedro representa a região atingida por raios de luz relevantes para o volume de visão atual, ou seja, raios de luz que podem gerar sombras na parte visível da cena nesse instante.

2. Define-se P como um volume de visão que englobe todo o poliedro B, de forma que o vetor direção da câmera desse volume seja perpendicular à direção da luz.

3. Define-se n como a distância entre o plano *near* e o ponto de referência do volume P. Este é um parâmetro livre do algoritmo. O controle do chamado "*warping effect*" é feito por esse parâmetro.

4. Aplica-se as matrizes definidas pelo volume P durante a renderização do mapa de sombras, de forma similar à transformação utilizada no PSM.

A Figura 2.3 mostra a sequência de passos para um caso hipotético, de forma a explicar melhor o funcionamento do algoritmo.

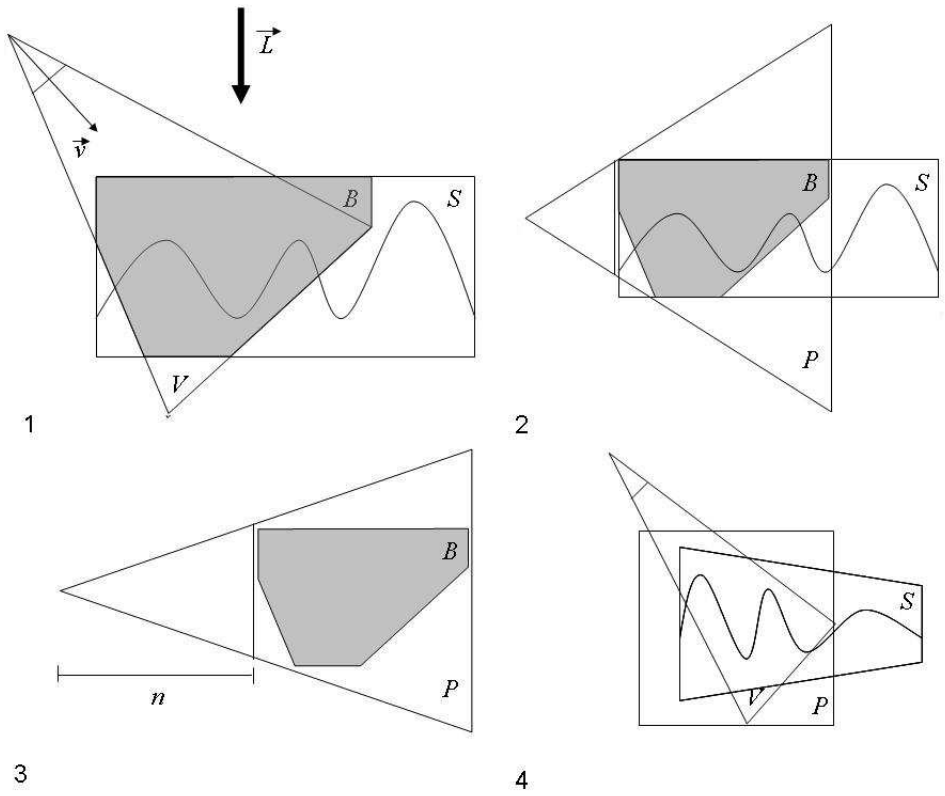

Figura 2.3: Passos do algoritmo do *Light Space*

O algoritmo elimina os tratamentos especiais que ocorrem no PSM, além de gerar uma melhor distribuição da utilização de forma que objetos distantes não sejam tão prejudicados na visualização. É plausível que esse tipo de correção possa ser aplicado em texturas em outros tipos de aplicações, o que explicaremos a seguir.

PUC-Rio - Certificação Digital Nº 0721348/CB

2.3 Desenho de Dados Vetoriais sobre Terrenos

É possível distinguir duas frentes possíveis para o desenho de dados vetoriais sobre malhas de terrenos. São elas a abordagem de geometria e a de textura.

2.3.1 Abordagem de Geometria

A abordagem de geometria consiste em obter a geometria 3D a ser desenhada dados os dados geográficos bidimensionais e a malha de terreno em questão. Para o casos de linhas bidimensionais é intuitivo construir uma linha tridimensional correspondente a um terreno. Cada segmento da linha de entrada é subdivido em vários segmentos menores. É possível calcular no terreno, dado um ponto, qual a sua altura correspondente. Utilizando esse cálculo, podemos gerar o segmento tridimensional que acompanhe o terreno. Observe que o número de vezes que os segmentos de entrada são subdivididos é extremamente relevante para a qualidade e eficiência do desenho da linha final. Caso a linha não seja subdividida o suficiente, a linha resultante pode não acompanhar o terreno de forma adequada. Por outro lado, uma alta quantidade de subdivisões pode gerar um número grande de vértices que não necessariamente contribuem para a qualidade obtida. Para o caso clássico de terrenos regulares, uma medida razoável para o tamanho do segmento a ser subdividido é o espaçamento da grade que representa o terreno. Para terrenos não regulares, tal medida pode não ser trivial, e calcular a linha tridimensional que contenha o número mínimo de vértices é um algoritmo geométrico complexo.

Mesmo com a obtenção de uma linha tridimensional satisfatória, a visualização ainda pode apresentar problemas. O mais básico é o conflito de *z-buffer*, pois se a linha está exatamente sobre o terreno, os valores de z entre a malha do terreno e a linha são tão próximos que a linha aparece falhada ou tracejada. Uma solução simples é acrescentar um *offset* para a altura da linha, para que esta fique um pouco acima do terreno, porém este artífício pode ser perceptível quando o observador chega próximo à linha, além de não garantir qualidade para casos distantes. Uma abordagem mais adequada é utilizar *Polygon Offset* do OpenGL (Shr05), onde esse deslocamento é feita no z projetado, funcionando melhor independente da distância do observador à geometria que está sofrendo o conflito do *z-buffer*. No entanto, é difícil escolher um valor de *offset* global que funcione para todos os casos.

Outro problema mais difícil de ser contornado é que as linhas devem se

PUC-Rio - Certificação Digital Nº 0721348/CB

dispor sobre um terreno em multi-resolução, o que implica que a linhas obtidas em um nível se tornam inválidas quando a posição do observador e o nível de detalhe do terreno se alteram. Quando o LOD se baseia em níveis discretos de detalhe, é possível que se obtenha uma linha adequada para cada um desses níveis. Mas para algoritmos de LOD contínuo, é necessário que o cálculo seja feito "*on the fly*", já que este depende da posição do observador. Tal cálculo tornaria essa opção impraticável, devido ao grande custo computacional que demanda. Algo a ser considerado também é o aumento no número de vértices a serem processados. Na Figura 2.4, utilizando um gráfico de perfil de terreno, uma ferramenta comum em software de manipulação de dados de altimetria, podemos perceber o quão complexa pode se tornar uma linha tridimensional feita a partir de um simples segmento de reta. Como dados geográficos reais tendem a ter um tamanho considerável, tal aumento do número de vértices a serem renderizados parece tornar inviável a abordagem para um caso de um mapa de polilinhas real.

PUC-Rio - Certificação Digital Nº 0721348/CB

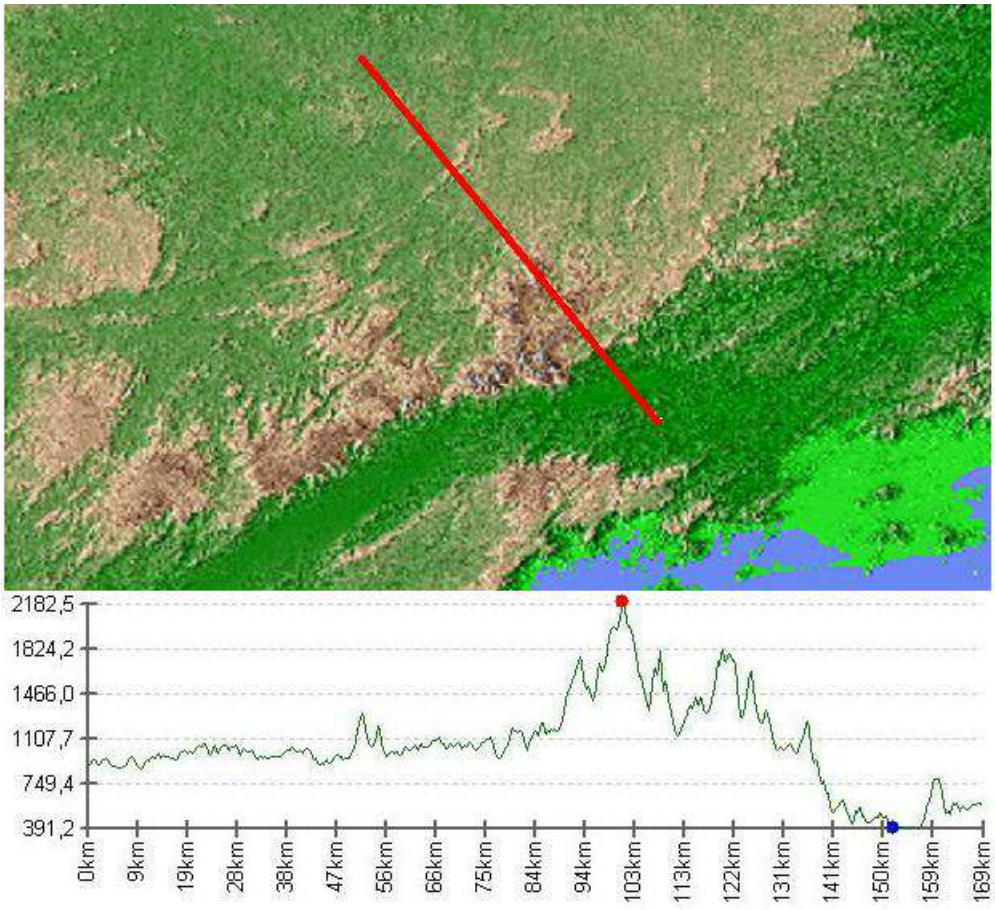

Figura 2.4: Gráfico do perfil do terreno para um segmento

O caso de polígonos é ainda mais complicado. O trabalho de Schneider et al. (Sch05) explica brevemente uma técnica geométrica para polígonos. Basica-

mente, ele projeta a borda do polígono bidimensional na malha do terreno, e, por ora, desconsidera a altitude, tratando a malha como bidimensional. A partir daí, são realizadas testes de interseção segmento-segmento entre a borda do polígono e os segmentos que formam os triângulos da malha do terreno. Tendo tais pontos de interseção, calcula-se a altitude destes no terreno e, para concluir a construção do polígono tridimensional, os triângulos interiores ao polígono bidimensional de entrada são incluídos no resultado final, juntamente com os triângulos formados pelas interseções de borda. Apesar da explicação parecer simples, os passos do algoritmo são baseados em procedimentos geométricos não triviais, como interseções entre conjuntos de segmento, e inúmeros testes de localização de ponto em polígono. Assim como explicitado em (Sch05), fica claro a impossibilidade de executar tal série de operações em tempo real, forçando que os polígonos sejam gerados em pré-processamento.

Além disso, as preocupações com a visualização são similares ao caso das linhas. Para que a visualização dos polígonos seja adequada, além de definir o *offset*, é necessário que o polígono tridimensional se comporte da mesma forma que a multi-resolução do terreno, tornando a renderização dos dados geográficos extremamente dependente do algoritmo utilizado para visualizar o terreno.

PUC-Rio - Certificação Digital Nº 0721348/CB

2.3.2 Abordagem de Textura

A abordagem de textura desenha as primitivas geográficas em uma imagem, aplicando-a sobre o terreno como textura. Essa é a forma mais eficiente de desenho e elimina dificuldades inerentes à abordagem de geometria, como a precisão do *z buffer*, *offsets* e a preocupação com a segmentação adequada, além das dificuldades de obter os polígonos correspondentes à malha do terreno. Uma nova variável a ser considerada que impacta diretamente na qualidade final da visualização é a resolução utilizada para a textura. É possível que a resolução da textura, que é geralmente limitada por hardware, seja consideravelmente menor que a resolução necessária para cobrir a janela de visualização do terreno onde esta é aplicada, podendo gerar situações onde as linhas fiquem "borradas", efeito conhecido como falta de amostragem (*undersampling*). Por outro lado, uma textura de resolução muito grande não implica em grande qualidade de visualização, pois nesses casos, linhas podem parecer pontilhadas quando na verdade são contínuas, gerando o efeito chamado de excesso de amostragem (*oversampling*). A Figura 2.5 mostra uma textura de uma mapa aplicada a um quadrilátero exemplificando como a diferença da resolução da textura influencia a visualização. Em uma área de

visualização de 800x600 pixels, são aplicadas texturas de 256x256 e 2048x2048 pixels, respectivamente.

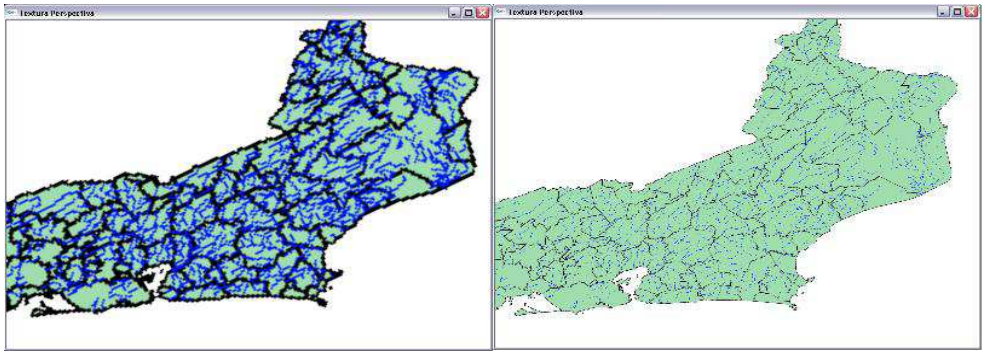

Figura 2.5: Efeitos de *undersampling* e *oversampling*, respectivamente

O trabalho de Kersting e Döllner (Ker02) descreve uma técnica de pirâmides de textura para o desenho vetorial sobre terrenos multi-resolução. É assumido que o terreno utiliza uma quad-tree para a organização da multi-resolução, e uma textura é gerada para cada nó da hierarquia, renderizando os dados vetoriais pertencentes à região representada. Um critério de carga da textura mais adequada é utilizado durante a navegação, como a distância para a câmera, por exemplo. Geralmente, a pirâmide é calculada em pré-processamento, gerando uma textura muito grande para o nível mais alto da árvore (que representa a área completa do terreno) e esta é escalada em texturas menores para os níveis inferiores da estrutura. Esta pirâmide estática é a mais utilizada pelos software do ramo, pois apresenta boa eficiência em troca de um gasto adicional de memória para armazenar as texturas. O texto apresenta uma alternativa à essa técnica, que é a geração "*on the fly*" das texturas relativas à pirâmide, o que permite um uso mais inteligente da memória, assim como uma melhor decisão quanto à resolução da textura utilizada, resultando em melhor qualidade visual. Mas a técnica de pirâmides, como previamente exemplificado na Seção 1.1, causa falhas visuais notáveis, como descontinuidade e *aliasing* de acordo com a perspectiva, independente da geração estática ou dinâmica.

O trabalho de Schneider et al (Sch05) apresenta uma solução de desenho de dados vetoriais sobre o terreno usando a abordagem de textura, com uma melhoria da qualidade usando técnicas de correção perspectiva. Fazendo uma analogia da direção de projeção dos dados geográficos como uma fonte de luz, utiliza-se a transformação do *Light Space Perspective Shadow Maps* (Wim04) durante o desenho da textura do mapa e na geração de coordenadas de textura. Utilizando essa técnica, os autores conseguiram um melhor aproveitamento da resolução da textura e uma redução do *aliasing* perspectivo. A Figura

PUC-Rio - Certificação Digital Nº 0721348/CB

2.6 mostra o resultado conseguido por esse trabalho. Porém, os autores não apresentam soluções para minimizar o tempo da geração da textura dependendo do tamanho dos dados vetoriais que serão visualizados. O tamanho desses dados não é explicitado nos resultados apresentados no artigo.

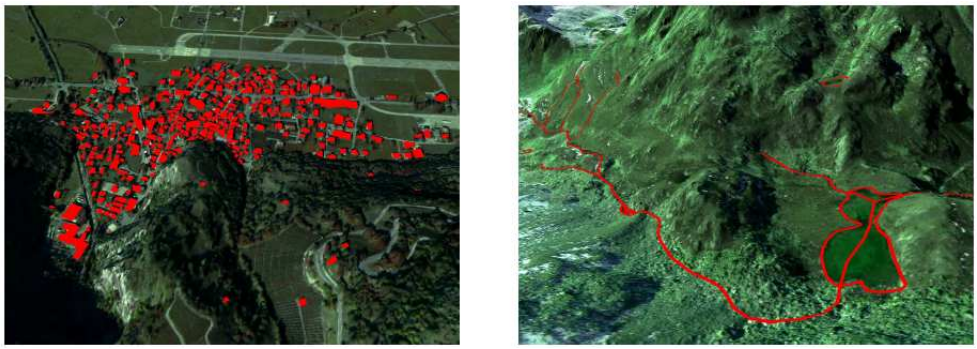

Figura 2.6: Resultado do trabalho de Schneider et al. (Extraída de (Sch05))

2.4 Simplificação de Geometrias

A simplificação de geometrias tem o mesmo objetivo básico das técnicas de LOD em malhas tridimensionais de triângulos, ou seja, permitir uma visualização com pouca ou nenhuma perda de qualidade utilizando uma quantidade consideravelmente menor de vértices. O trabalho mais citado sobre simplificação de linhas 2D é o trabalho de Douglas e Peucker (Dou73), que foi revisitado e aprimorado em vários trabalhos posteriores (Her92, Her94, Ebi02).

A abordagem para a simplificação mostrada no artigo original é feita linha a linha. São dados de entrada uma linha de n vértices p_1, p_2 ... p_n e um valor numérico ϵ para a tolerância. Simplificadamente, parte-se do segmento formado pelo primeiro e o último vértice da linha, encontra-se o vértice mais distante deste segmento entre os demais. Considerando que o vértice escolhido possui índice i, se a distância do segmento $\overline{p_1 p_n}$ a p_i for menor que ϵ, o algoritmo termina. Senão, o algoritmo é chamado recursivamente para as linhas $p_1 \dots p_i$ e $p_{i+1} \dots p_n$.

É comprovada a redução da quantidade de vértices necessários para reproduzir a linha com fidelidade. Durante uma visualização, uma métrica considerada adequada para a tolerância é o valor de um pixel em coordenadas de mundo, pois isso gera um resultado final que difere da linha original em no máximo um pixel, o que torna praticamente imperceptível a simplificação.

Para a visualização de dados geográficos reais, foi observado que a abordagem linha a linha do algoritmo não é satisfatória. Com o avanço do hardware para processamento gráfico, tornou se mais vantajoso enviar mais

PUC-Rio - Certificação Digital Nº 0721348/CB

vértices para o pipeline de visualização do que executar certas decisões para eliminar apenas alguns vértices. Em média, o tempo necessário para simplificar uma linha e renderizá-la supera o tempo para renderizar a linha original. Para que a simplificação gere uma maior eficiência, é necessária uma abordagem para uma quantidade maior de dados, onde apenas um comando elimine uma quantidade de vértices que seja suficiente para interferir na velocidade geral da visualização.

PUC-Rio - Certificação Digital Nº 0721348/CB