

6 Conclusão

Este trabalho teve como objetivo o estudo da visualização da geometria de terrenos grandes, levando em consideração a necessidade de acessar os dados do terreno em memória secundária.

Primeiro foi apresentada uma estrutura de dados de fácil compreensão e simples implementação. A estrutura mostrou-se bastante eficiente para a finalidade a que se destina. O pré-processamento dos dados para a construção da estrutura de multi-resolução é simples e escalável.

Foi também proposto um algoritmo de geração de malhas sem falhas ou vértices T levando em consideração a presença ou não de ladrilhos em memória.

As maiores contribuições deste trabalho são as seguintes:

- Foi proposta uma organização de tarefas em 2 linhas de execução que são executadas de forma independente.
- Pode-se ressaltar a solução para o desenho de ladrilhos apresentada. Há uma grande economia de memória e de transferência de dados para a *GPU* com o uso de indexações e malha 2D únicas para todos os ladrilhos, levando-se em consideração que o armazenamento é feito em memória de vídeo.
- Apresentou-se também uma implementação da visualização da malha utilizando uma quantidade limitada de polígonos, introduzindo o menor erro possível. A implementação mostrou-se correta, conseguindo sempre manter uma taxa de quadros por segundo mínima.
- Foi proposta uma forma simples de gerenciar a carga/descarga de ladrilhos para/da memória principal, com o uso de técnicas de predição. Este gerenciamento mostrou-se bastante eficaz, pois, como foi apresentado nos testes, foi possível navegar por um terreno grande carregando em memória apenas 3% dele.

- Muitos testes foram realizados para comprovar a real eficácia do uso da predição de câmera para minimizar o erro introduzido pela não presença em memória dos ladrilhos necessários.

6.1 Trabalhos Futuros

Este trabalho pode evoluir em diversos aspectos. Ele foi desenvolvido com o objetivo de ser facilmente extensível tanto com relação ao seu componente de visualização quanto ao de gerenciamento de ladrilhos.

A visualização é uma das partes deste trabalho que pode ser mais estendida. A seguir, são apresentadas algumas possíveis evoluções deste componente:

- A adição de transições suaves na geometria (*geomorphing*) acrescentaria muito ao componente de visualização. Isto poderia ser feito utilizando a *GPU*, o que causaria um impacto pequeno no desempenho. Esse impacto não deve ser significativo, pois com as transições pode-se aumentar a tolerância de erro projetado sem que uma perda de qualidade seja perceptível.
- Mais recentemente, as *GPUs* passaram a suportar o acesso a texturas em um *vertex program*. Já existem alguns autores propondo soluções de visualização de terrenos utilizando tal característica. Na realidade, a idéia principal não difere muito da apresentada neste trabalho, sendo que as alturas dos vértices estariam em textura. Com isso, um *geomorphing* já seria acrescentado automaticamente pela interpolação das texturas, mas cuidados adicionais para resolver as falhas entre ladrilhos ainda têm que ser investigadas.
- Como o sistema suporta terrenos muito grandes, seria interessante também implementar uma forma de visualização da malha do terreno de forma esférica, podendo-se assim representar planetas.
- Vale a pena investigar ainda uma forma de desenhar o ladrilho que não seja através da força bruta. Uma idéia seria usar malhas irregulares que melhor se adaptem ao terreno para representar um ladrilho. Com isso, seria possível obter uma malha que representa o ladrilho com um erro menor e utilizando uma menor quantidade de triângulos. Porém, seria necessário uma forma mais complexa para garantir a ausência de falhas na malha.

- Foi brevemente testado nesta pesquisa o uso de uma técnica de oclusão para a eliminação de ladrilhos baseada em Bittner *et al.* [7]. A solução testada faz uso de aceleração de hardware para testar a oclusão de ladrilhos verificando, dada a caixa envolvente do ladrilho, o número de pixels que seria desenhado com o *z-buffer* atual. Com isso, para que a oclusão ocorra de forma efetiva, é necessário desenhar os ladrilhos na ordem de frente para trás. A solução procura ainda minimizar o número de testes de oclusão e reduzir os atrasos provocados pelo tempo de espera dos resultados. Porém, a implementação testada provocou uma perda de desempenho grande em situações em que poucos ladrilhos estão oclusos. Fica como trabalho futuro investigar detalhadamente a técnica implementada, além de estudar outras técnicas de oclusão que possam também ser utilizadas em conjunto com este trabalho.

Existem também algumas possibilidades de evolução para o componente de gerenciamento de ladrilhos, conforme apresentado a seguir:

- Foi implementada neste trabalho uma forma de carregamento de ladrilhos baseada no erro projetado do ladrilho. Com isso, os ladrilhos mais importantes, ou seja, com maior erro projetado, têm prioridade de carregamento. Além disso, foi possível limitar o número de ladrilhos carregados em um percurso de carregamento, evitando uma defasagem muito grande da linha de execução de gerenciamento. No entanto, uma investigação detalhada deve ainda ser feita para comprovar a real eficácia dessas modificações.
- Com um terreno muito grande, é possível que a quantidade de metadados cresça de tal forma que passe a ser um problema tão importante quanto o da geometria. Portanto, seria interessante pesquisar uma forma de fazer também o gerenciamento dos metadados em memória secundária.

Além disso, outros aspectos também podem ser explorados como, por exemplo, eliminar as restrições referentes à completude da árvore e às resoluções dos ladrilhos.

É possível também minimizar a quantidade de dados armazenada, assim como a quantidade de dados necessária para carregar cada ladrilho. Uma idéia seria armazenar e comprimir somente as diferenças em alturas dos filhos para os seus pais.

Está prevista para um futuro próximo a integração da implementação deste trabalho com o Terralib [25], que é uma biblioteca de sistemas de informações geográficas (*GIS*).

Apesar do foco deste trabalho ter sido a resolução dos problemas de geometria, o uso de texturas de alta qualidade é tão ou mais importante do que a geometria. Geralmente, as texturas utilizadas para uma região de um terreno apresentam resolução superior à da malha da geometria. Portanto, pode-se perceber que o problema da quantidade de dados neste caso pode ser até mais problemático. Acredita-se que, utilizando um esquema de gerenciamento semelhante ao apresentado neste trabalho, seja possível também resolver o problema do armazenamento em memória secundária de texturas. Deve-se observar que neste caso terá que ser usada uma métrica de erro diferente, além de um gerenciamento adicional da memória de vídeo da *GPU*.