

5 Conclusão

O objetivo deste trabalho foi compreender o problema da visualização interativa de terrenos de modo geral e, simultaneamente, investigar o algoritmo proposto por Lindstrom & Pascucci para este problema. Sendo assim, a tarefa de implementação deste algoritmo, seguida dos testes realizados em diferentes situações e incluindo a comparação prática com outro algoritmo da área, nos forneceu uma experiência importante.

Muitas de nossas constatações, de certo modo, corroboram a maioria das afirmações antecipadas nos trabalhos de Lindstrom & Pascucci. Assim, primeiramente, é preciso destacar os bons resultados fornecidos pelo algoritmo. Em nossos testes em condições normais, o algoritmo sempre conseguiu fornecer taxas interativas ou acima de 20 FPS (*frames per second*) em média. Além disso, foi possível verificar que a tarefa de implementação do algoritmo, de fato, é relativamente simples, principalmente quando comparada com alguns outros da área, bem mais complexos.

Apesar de o algoritmo poder ser implementado e executado sem incorporar as operações de *culling* da malha fora da visão e *geomorphing*, nossos testes revelaram que elas se tornam imprescindíveis à medida que as dimensões do terreno aumentam. Contudo, a escolha de exatamente o que utilizar ou dispensar depende das necessidades específicas de cada aplicação. Por exemplo, a partir de nossos testes, obtivemos bons resultados para o terreno menor, o Itaoca1, em praticamente todas as condições. Em outros termos, se a aplicação irá trabalhar somente com terrenos bem pequenos, como o Itaoca1, então não parece tão necessário implementar *culling* da malha fora da visão e *geomorphing*. Na verdade, neste caso, o algoritmo de De Boer, provavelmente, será suficiente.

Por outro lado, devemos ressaltar que em nossos testes foram analisadas somente questões relativas à malha que aproxima o terreno. Contudo, aplicações reais podem e devem utilizar recursos como texturas de melhor qualidade,

iluminação e *fog* (neblina) adequados com o intuito de melhorar a qualidade final das apresentações.

Além disso, devemos dizer que, neste trabalho, não foram feitos testes para comparar o número de triângulos gerados pelo algoritmo de refinamento estudado aqui e uma malha ótima (possivelmente, gerada por um algoritmo de simplificação). Porém, segundo Lindstrom & Pascucci[3], este algoritmo de refinamento produz uma malha bem próxima a uma malha ótima.

Outra questão não implementada neste trabalho foi o esquema de *multi-threads*, também sugerido por Lindstrom & Pascucci, possivelmente uma *thread* sendo responsável pelo refinamento e outra pela renderização da malha produzida. Espera-se que isto melhore ainda mais o desempenho da aplicação, o que é confirmado por Lindstrom & Pascucci através de uma análise neste sentido mostrada em [3]. Em princípio, e conforme já dito e mostrado por nossos testes, mesmo sem usar *multi-threads* o algoritmo tem fornecido bons resultados. Provavelmente o uso de *multi-threads* se tornará importante quando executando terrenos de dimensões bem maiores que os terrenos usados em nossos testes.

Também não implementamos o esquema de organização dos dados proposto por Lindstrom & Pascucci em [2] e [3] para trabalhar de modo eficiente com terrenos que não cabem completamente na memória. Por conta disso, não conseguimos trabalhar, em nossas plataformas de teste, com um terreno de dimensões 16.385 x 16.385 também disponibilizado em [14] e representando uma região de Washington, nos Estados Unidos. Segundo os próprios Lindstrom & Pascucci, este terreno ocuparia algo em torno de 5 GB em memória e nossa plataforma de teste mais potente está equipada com 3 GB de memória RAM.

Contudo, devemos destacar que nossa implementação, ou seja, nosso módulo Terreno, está sendo utilizado no desenvolvimento de aplicações no laboratório Tecgraf. Destacamos entre elas, um sistema para análise e visualização de dados sísmicos tridimensionais selecionados em camadas e depois agrupados por um algoritmo baseado em rede neurais híbridas que está sendo desenvolvido em parceria com a Petrobras. Outro destaque é o sistema de jogos didáticos para treinamento e ensino que inclui um simulador de vôos sobre terrenos reais e está sendo desenvolvido em parceria com a Marinha do Brasil.

Resumidamente, comprovamos que os trabalhos de Lindstrom & Pascucci e, especificamente, o algoritmo estudado neste trabalho, sem dúvida trouxeram

contribuições significativas para o problema da visualização interativa de terrenos. Inclusive, imagina-se que não será fácil superá-lo tão cedo. Contudo, ainda não se pode considerar que está tudo resolvido e o principal ponto fraco deste algoritmo é o alto consumo de memória.

Sendo assim, apesar dos bons resultados apresentados pelos recentes trabalhos de Lindstrom & Pascucci, as pesquisas nesta área ainda não devem ser consideradas concluídas. Aplicações exigindo terrenos cada vez maiores e com superfícies cada vez mais complexas sinalizam no sentido inverso. No entanto, a tendência parece ser a utilização de estratégias de software inteligentes e que, ao mesmo tempo, procurem tirar o máximo proveito das facilidades oferecidas pela indústria de hardware, que também encontra-se em constante evolução.

5.1. Trabalhos Futuros

Nossa implementação pode ser evoluída, ainda seguindo as idéias propostas por Lindstrom & Pascucci, para trabalhar com terrenos que não podem ser totalmente carregados para a memória e usando o esquema de *multi-threads*. Na verdade, não basta que o mapa de alturas e a textura caibam na memória, pois na fase de pré-processamento algumas informações são incorporadas aos dados de entrada e, se o terreno não se encaixar no formato $(2^{n/2} + 1) \times (2^{n/2} + 1)$, onde n é um inteiro maior ou igual 2, ele precisa ser estendido. Em outras palavras, considerando que, em geral, este algoritmo consome uma quantidade significativa de memória, então para processar terrenos realmente grandes é importante poder manter parte dos dados em disco.

Além disso, conforme já dito, o uso de *multi-threads* pode se tornar um componente importante para acelerar a renderização quando desejarmos trabalhar com terrenos suficientemente grandes.

Podemos também tentar substituir as esferas envolventes associadas com cada vértice por outro volume menos folgado. Se não for um volume cujo teste de interseção com os planos do *frustum* de visão seja complexo, isto poderia acelerar o *culling* da malha fora da visão, permitindo identificar mais cedo se uma região está localizada dentro ou fora do *frustum* de visão e, assim, aumentar a velocidade do algoritmo.

Outra possibilidade de trabalho futuro trata-se da utilização das idéias que incluem estereoscopia publicado em [9] em conjunto com o algoritmo de Lindstrom & Pascucci estudado neste trabalho.

Finalmente, é importante prosseguir investindo na busca por alternativas que não exijam um terreno de dimensões $(2^{n/2} + 1) \times (2^{n/2} + 1)$, simplifiquem ainda mais a complexidade e o tempo consumido no pré-processamento dos dados, não consumam tanta memória e que, ao mesmo tempo, mantenham ou preferencialmente melhorem o desempenho do algoritmo atual.