

ANÁLISE DE DESEMPENHO DE UMA METODOLOGIA DE TRATAMENTO DE DADOS ALTIMÉTRICOS APLICADA À VISUALIZAÇÃO TRIDIMENSIONAL DE TERRENOS

Performance Analysis of an Altimetric Data Processing Methodology Applied to Three-dimensional Terrain Visualization

Flávio Luis de Mello¹
Luiz Felipe Coutinho Ferreira da Silva²

¹**Centro Tecnológico do Exército**
Divisão de Tecnologia da Informação
Av. das Américas 28.705, D13, Guaratiba, Rio de Janeiro, RJ, CEP 23020-470
fmello@ctex.eb.br

²**Instituto Militar de Engenharia – IME**
Seção de Engenharia Cartográfica
Praça Gen. Tibúrcio 80, Praia Vermelha, Rio de Janeiro, RJ, CEP 22290-270
felipe@ime.eb.br

RESUMO

A visualização tridimensional de terrenos em ambientes computacionais está geralmente relacionada com a capacidade de resposta em tempo real. O fator predominante para o desempenho adequado deste tipo de aplicação, quando executada em computadores portáteis e de escritório, está associada à eficiência na execução das atividades de desenho das placas de vídeo. Nestas aplicações, o modo através do qual os dados cartográficos são preparados torna-se preponderante. Este artigo apresenta uma análise de resultados obtidos a partir de uma metodologia de pré-processamento de dados altimétricos com vista ao uso em aplicações de visualização de terrenos.

Palavras chaves: Visualização Tridimensional, Modelagem de Terreno, Altimetria, Triangulação, Análise de Desempenho.

ABSTRACT

Three-dimensional terrain visualization on computerized environment is usually associated to real time response capabilities. The predominant factor for an adequate performance for this kind of application, when considering desktop computers or portables, is the video board efficiency for drawing activities. In these applications, the process through which the cartographic data are prepared becomes a preponderant feature. This article presents a performance analysis based on the results obtained by an altimetric data pre-processing methodology in order to make possible its use on terrain visualization applications.

Keywords: Three-dimensional Visualization, Terrain Modeling, Altimetry, Triangulation, Performance Analysis.

1. INTRODUÇÃO

A visualização tridimensional de terrenos a partir de um conteúdo cartográfico real envolve a manipulação de uma quantidade significativa de dados, sendo a camada de altimetria responsável por grande parte destes últimos. Tipicamente, para o caso brasileiro, pode-se assumir que, em média, uma camada

de altimetria proveniente do mapeamento sistemático nacional abrange 76% da quantidade total de dados a serem manipulados por um sistema computacional (ANCIÃES, 2003). Ilustrativamente, considerando cartas na escala de 1:50.000, o valor em questão corresponde a cerca de 400.000 pontos. Esse volume de dados, por sua vez, é um fator impeditivo para a construção de sistemas de visualização tridimensional

de terrenos. Sob esta ótica, entende-se como desejável manipular esta massa de dados a fim de viabilizar seu emprego em aplicações desta natureza.

Este artigo se propõe a apresentar os resultados obtidos através da aplicação de uma metodologia de tratamento de dados altimétricos. O objetivo desta metodologia é permitir o manejo adequado dos dados a fim de viabilizar aplicações relacionadas com a visualização tridimensional (3D) de terrenos, em tempo real. Mais especificamente, os estudos estarão centrados na categoria cartográfica de altimetria, pois esta influencia significativamente o desempenho de sistemas deste tipo.

O procedimento usual de representação 3D/volumétrica consiste em efetuar uma operação conhecida como triangulação, e que resulta em uma malha triangular. Esta malha é formada por um conjunto de triângulos geometricamente organizados, que juntos conferem um aspecto tridimensional à superfície. Paralelamente a esta representação conceitual, existe uma limitação tecnológica relacionada com a quantidade de triângulos a serem utilizados na representação. A relação entre o nível de detalhes de um terreno, e seu dado associado, pode facilmente exceder a capacidade de processamento de uma placa de vídeo o que, em muitos casos, torna impossível uma aplicação em tempo real. Desta forma, os dados do modelo de terreno devem ser trabalhados de tal modo que seja possível viabilizar sua utilização.

Em um ambiente tridimensional interativo, uma questão importante é a animação. Ela precisa ser suave e contínua, isto é, o usuário não deve perceber a transição entre imagens consecutivas, ainda que estas imagens sejam muito semelhantes entre si. Este efeito de transição de quadros é bastante nítido quando se compara a imagem de televisão atual com o, já ultrapassado, projetor de Super 8. No caso da televisão, a animação é feita utilizando mais imagens por segundo do que no projetor, isto é, a animação de televisão é mais rápida que este último. Tipicamente, o sistema Super 8 utiliza 18 fps (quadros por segundo – *frames per second*); a codificação NTSC utiliza 30 fps; e o PAL-M, 25 fps. Em um sistema computacional, por sua vez, o fator responsável pelo atraso na transição de uma cena para outra é a quantidade de objetos presente na mesma (LASSETER, 1987). Se uma cena possui muitos objetos, a placa de vídeo demora muito tempo para reproduzi-los na tela do computador, retardando o início da visualização da próxima cena.

Assim, considerando uma mesma capacidade de reprodução de dados, a fluidez da animação no sobrevôo do terreno, ou qualquer outro movimento virtual do usuário sobre o mesmo, depende inversamente da quantidade de triângulos utilizados para a representação de um cenário. Com o objetivo de assegurar uma taxa de atualização aceitável da imagem na tela do computador, tem-se que uma visão estratégica do terreno (com grande área de abrangência) deve ser representada com menor nível de detalhamento, enquanto que uma visão tática do mesmo (com pequena

região de abrangência) pode ser representada com maior quantidade de detalhes. A taxa de atualização da tela do computador é mantida em níveis compatíveis com os requisitos do sistema, equilibrando-se a abrangência da região de interesse com o grau de detalhamento da mesma. Desta forma, existe uma relação de compromisso entre qualidade visual e tamanho da região apresentada ao usuário.

2. SISTEMAS DE TEMPO REAL

O pré-processamento de dados altimétricos tem por objetivo estruturá-los de modo que possam atender as necessidades de aplicações de visualização 3D de terrenos. No contexto deste artigo, a taxa de atualização dos desenhos no cenário é a métrica utilizada para a análise de desempenho do sistema para a visualização tridimensional. Estes valores de velocidade estão relacionados com a capacidade de efetuar operações em tempo real, o que sugere a necessidade de compreender o que são sistemas dedicados a aplicações com esta característica.

Um sistema é um conjunto de elementos conectados, ou relacionados, de tal modo que o conjunto possa ser considerado como uma unidade (FREGOSI, 1980). Já um processo pode ser entendido como uma seqüência de estados que um sistema assume no tempo, à medida que as transformações ocorrem no seu interior (COSTA, 1992). Assim, um sistema que administra um determinado processo é considerado um sistema trabalhando em ambiente de tempo real se seus tempos de processamento, e conseqüentes respostas, forem compatíveis com a dinâmica do processo (ALLWORTH, 1981).

Deste modo, a caracterização de um sistema em ambiente de tempo real está relacionada com os tempos de atuação, onde cada aplicação possui suas próprias limitações. A denominação de sistemas que trabalham em ambiente de tempo real tende a ser atribuída àqueles responsáveis por supervisionar ou controlar processos que exijam tempos exíguos de resposta. Entre estes sistemas pode-se citar plantas industriais, usinas elétricas ou atômicas, centros médicos de tratamento intensivo, entre outros.

Por outro lado, considere sistemas tais como o controle de tráfego de rotas marítimas, ou a navegação de sondas cósmicas. No primeiro caso, uma possível unidade de medida para o processo pode ser representada por milhas náuticas/hora, enquanto que no segundo, pode-se utilizar anos-luz/ano. Ao realizar uma análise superficial destes casos, existe uma propensão em se questionar sua classificação como sistemas que trabalham em tempo real pois eles possuem tempos mais elásticos para respostas. Entretanto, este tipo de questionamento não é sustentável pois a definição de um ambiente em tempo real está relacionada não só com o tempo de resposta, mas também com a capacidade de tratar o problema oportunamente. Segundo Martin (1987), sistemas de tempo real podem possuir tempos de resposta mais alongados, contanto que o retardo na

resposta não prejudique o correto funcionamento da aplicação.

Em um ambiente computacional interativo, um sistema de visualização tridimensional de terrenos também deve permitir respostas em tempo real. Na atualidade, o projeto de equipamentos para sistemas computacionais móveis e embarcados é fortemente influenciado pelos requisitos operacionais que estabelecem as características a serem atingidas por um sistema. Com base nestas características, são elaborados os requisitos técnicos que interpretam através de termos claros e precisos as funcionalidades descritas no nível operacional. Em geral existe uma demanda que se contrapõe a restrições tecnológicas relacionadas com viabilidade de acondicionamento, limitação de peso, autonomia de operação e capacidade de processamento.

De modo a reduzir o peso do equipamento, pode-se utilizar materiais mais leves, empregar novas tecnologias, e eliminar supérfluos. Entretanto, as baterias são elementos necessários, pesados, e que não podem ser simplesmente descartados. A solução adotada pelos pólos de pesquisa e desenvolvimento em todo mundo para contornar este problema tem sido a redução da quantidade de baterias. De modo a preservar o tempo de autonomia do sistema procura-se reduzir o consumo de energia dos componentes eletrônicos para se adequarem à disponibilidade das fontes de energia. Esta redução de consumo é viabilizada através da diminuição da capacidade de processamento do sistema.

Neste contexto, o desenvolvimento de aplicações para visualização tridimensional de terrenos torna-se um desafio. Primeiro porque a natureza deste tipo de aplicação envolve uma grande quantidade de dados a serem processados. Segundo, porque os recursos computacionais disponíveis tendem a ser exíguos. A implicação direta destes dois fatores é que a aplicação torna-se incapaz de disponibilizar os dados em tempo real. Neste sentido, é recomendável uma solução de pré-processamento de dados que permita desonerar a aplicação de parte do processamento, conforme descrito sucintamente na seção a seguir.

3. RESUMO DA METODOLOGIA

Um mapa é composto por uma série de camadas de informação tais como: altimetria, sistema de transporte, hidrografia, etc. Estas camadas de informação, no âmbito do mapeamento topográfico sistemático (DSG/IBGE), são denominadas categorias cartográficas, e representam a informação gráfica do mapa através de geometrias dos tipos ponto, linha e área. A carga de todas as geometrias contidas em cada categoria cartográfica pode se tornar bastante custosa, seja sob o ponto de vista do processamento gráfico, seja do consumo de memória. Em ambos os casos a aplicação final pode se tornar lenta.

Conforme descrito por Mello e Ferreira (2005), em geral, apenas uma fração das geometrias de um ambiente virtual estão efetivamente visíveis em um determinado momento, tal como ilustrado na Figura 1.

Observe que no instante descrito na figura, o soldado visualiza apenas o blindado e parte da via de acesso. O helicóptero não está visível na situação em questão, e neste caso, não seria preciso exibir todas as feições componentes do cenário, isto é, o conjunto de todos os objetos que serve para capturar a realidade virtual ou a atmosfera do ambiente o qual deseja-se representar. Desta forma, se fosse possível determinar quando uma destas geometrias está invisível antes dela ser desenhada, seria possível simplesmente ignorá-la, evitando assim o processamento de dados desnecessários para a aplicação. Este procedimento de recorte é conhecido como *frustum culling* (ASTLE et alii 2002).

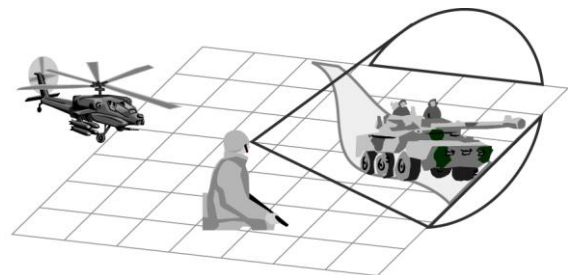


Fig. 1 - Visibilidade de objetos em um determinado instante: a via de acesso e o blindado estão visíveis para o soldado, mas o helicóptero não.

A fração de geometrias visíveis de um ambiente virtual é definida como sendo a interseção entre o cone de visão do observador e as categorias cartográficas. Por simplicidade, considere apenas a camada de altimetria de uma região qualquer de interesse, tal como ilustrado na Figura 2a. Neste exemplo, observa-se uma área quadrangular em comum entre o cone de visão e os dados de altimetria. Em sistemas computadorizados é comum representar o cone de visão do observador por uma pirâmide de visão (LAMOTHE, 1995). O volume de cálculos relacionados com pirâmides de visão tende a ser significativamente menor que o volume de cálculos associados a cones de visão, e por este motivo, as pirâmides de visão são mais utilizadas.

A região de interesse, por sua vez, é dividida em células espacialmente indexadas (MELLO & FERREIRA, 2004) de forma que seja possível identificar quais áreas são interceptadas pela pirâmide de visão. Um exemplo de indexação é representado na Figura 2b. Por fim, é realizado o teste de interseção para determinar quais células encontram-se totalmente, ou parcialmente contidas na pirâmide de visão. No caso da Figura 2b observa-se que as células 0-4-5-10-15 foram descartadas durante este procedimento, significando que nenhuma das geometrias contidas no interior destas células seria remetida para a placa de vídeo, desonerando o processamento da mesma.

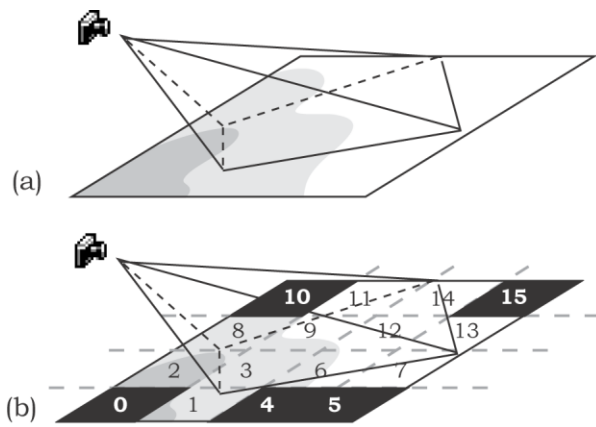


Fig. 2 - (a) Uma pirâmide de visão delimitando a área visível de uma camada de altimetria; (b) A subdivisão da região de interesse em células espacialmente indexadas, com a respectiva identificação das células visíveis na pirâmide de visão.

O descarte de células permite a eliminação, através de um único teste, de vários objetos simultaneamente. Este artifício é utilizado para potencializar a capacidade de um sistema representar um dado cenário em tempo real, principalmente quando a aplicação precisa manipular milhares de primitivas geométricas. Segundo Anciães (2003), uma carta topográfica na escala de 1:50.000 possui no pior caso cerca de 400.000 pontos. Em aplicações com características de tempo real, assume-se que a taxa de atualização da imagem no monitor do computador deve ser superior a 12 fps (*frames per second* - fps) (LASSETER, 1987). Lasseter também sugere que existe uma faixa transitória, definida pelos valores entre 9 e 12 fps, na qual animações computacionais podem ser classificadas como tempo real com perda de continuidade.

A indexação das células é realizada segundo o que se chama de curva de preenchimento espacial. Uma curva de preenchimento de espaço N -dimensional é contínua e bijetiva sobre o intervalo $[0,1]^N$. Em particular, uma curva de preenchimento 2-dimensional, é uma curva contínua que passa por todos os pontos de um quadrado unitário $[0,1]^2$. Como exemplo de curvas pode-se citar as de Hilbert (COLE, 1987 e BULTZ, 1971), de Moore (VELHO, 1992), de Sierpinski (BULTZ, 1971) e de Morton (PATRICK & ANDERSON, 1968 e LAURINI, 1985).

Para aglomerar os dados de uma forma ótima, a literatura (KAMEL & FALOUTSOS, 1993) propõe o uso de curvas de preenchimento do espaço (ou fractais), em especial a curva de Hilbert. Entretanto, observa-se que as curvas de Hilbert, Moore e Sierpinski, possuem implementações notadamente recursivas, e ainda, apresentam o computo lento do mapeamento célula \rightarrow espaço N -dimensional.

Entretanto, ao contrário das curvas já descritas, a curva de Morton apresenta uma implementação seqüencial. Além disto, o mapeamento célula \rightarrow espaço N -dimensional poder ser realizado através de uma

simples manipulação de bits, o que torna seu computo extremamente rápido. Assim, na metodologia implementada foi dado preferência ao método de classificação de células utilizando curvas de Morton.

4. DESCRIÇÃO DOS TESTES

Os testes têm por objetivo analisar como o tamanho das células que dividem a região de interesse pode influenciar no desempenho de uma aplicação dedicada a visualização tridimensional de terrenos. Optou-se por utilizar a camada de dados altimétricos nos testes, pois ela contribui com grande parte das primitivas geométricas de um cenário, conforme descrito na seção anterior.

Durante os testes realizados, foi utilizada a carta MI-2842-4-SE (Piraquara). A massa de dados foi obtida a partir de curvas de nível com equidistância de vinte metros, e generalizada através do algoritmo de Douglas-Peucker com tolerância de três metros, resultando em 137.841 triângulos a serem desenhados. Os equipamentos utilizados foram dois computadores pessoais com 256MB de memória, sendo um deles equipado com um processador Pentium III de 1GHz, e outro com processador Pentium I de 233MHz. As placas de vídeo utilizadas foram: uma S3 Virge de 1MB com barramento PCI, uma NVidia GeForce 2 de 64MB com barramento PCI e uma NVidia GeForce 2 de 64MB com barramento AGP.

A placa S3 Virge não implementa qualquer instrução de aceleração de vídeo por *hardware*, enquanto que as placas GeForce 2 possuem esta implementação. Por este motivo, é esperado um melhor desempenho destas últimas. As placas de vídeo foram permutadas entre os dois computadores, exceto a placa GeForce 2 AGP que não foi possível instalar no computador Pentium 233MHz pois este não dispunha de barramento AGP.

A região de interesse foi subdividida em grades 1×1 , 2×2 , 4×4 , 8×8 , 16×16 , 32×32 , 64×64 , 128×128 , isto é, grades $2^x \times 2^x$, onde $x = 0, \dots, 7$. Cada uma destas grades divide a região respectivamente em 1, 8, 16, 64, 256, 1.024, 4.096, 16.384 células, indexadas pelo código de Morton.

Para cada 3-tupla de configuração $\langle \text{computador}, \text{placa de vídeo}, \text{grade} \rangle$ foi executado um mesmo sobrevôo virtual sobre a região de interesse. A construção da rota de sobrevôo foi realizada através de um simples exercício com seis indivíduos. Estes indivíduos foram igualmente distribuídos entre homens e mulheres, engenheiros e não engenheiros. Para cada indivíduo, foi apresentado um mesmo conjunto de 39 pontos de controle distribuídos aleatoriamente sobre uma superfície retangular, tal como ilustrado na Figura 3.

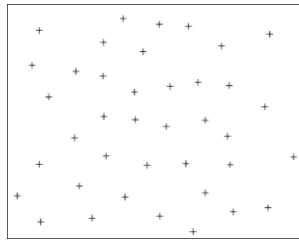


Fig. 3 - Distribuição aleatória dos pontos de controle para definição de trajetórias.

Os indivíduos puderam escolher livremente o ponto de partida de suas trajetórias, bem como a quantidade de pontos de controle utilizados ao longo do percurso. De modo a produzir uma trajetória suave e contínua, foi criada uma curva *spline* passando pelos pontos de controle definidos pelo indivíduo. A quantidade de pontos empregada em cada *spline* foi determinada a partir dos segmentos entre dois pontos de controle. Entre cada segmento foram inseridos dez pontos descritores da curva. A Figura 4 ilustra o grupo 1 de trajetórias contendo os percursos em questão numerados de 1 a 6.

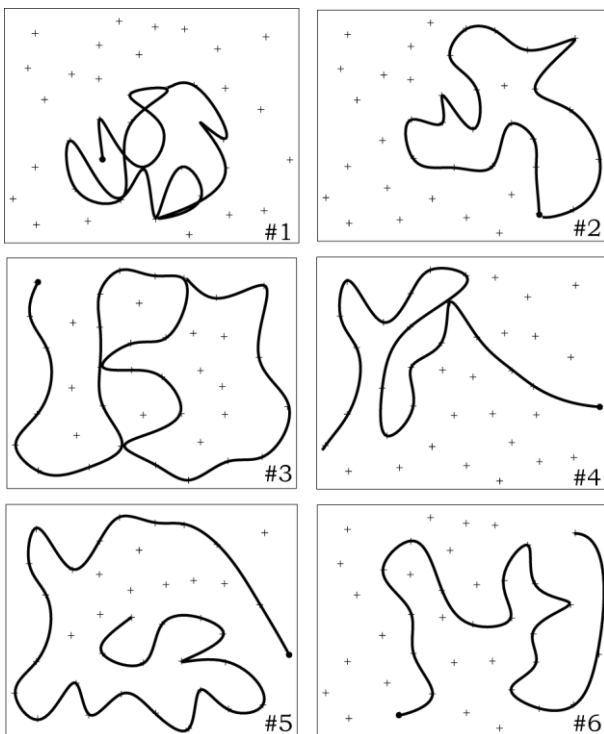


Fig. 4 - Grupo 1: Trajetórias aleatórias empregadas nos testes de desempenho.

Adicionalmente, foram criados outros quatro percursos, associados ao grupo 2 de trajetórias, numerados de 7 a 10. Estas trajetórias têm como característica principal percorrerem regiões com particularidades geométricas, e são apresentadas na Figura 5. Na trajetória #7, o percurso é realizado sobre a borda da região triangulada, situação na qual o cone de visão apresenta uma de suas metades isenta de quaisquer feições geométricas a serem desenhadas. As trajetórias #8 e #9 também representam deslocamentos

em linha reta, e buscam densificar o cone de visão com as células espacialmente indexadas. Por fim, a trajetória #10 representa uma composição de percursos retilíneos e arcos de circunferência.

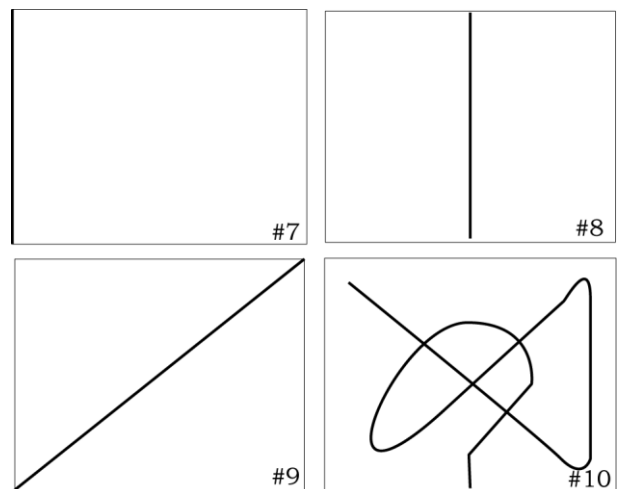


Fig. 5 - Grupo 2: Trajetórias geométricas empregadas nos testes de desempenho.

Nos testes realizados, a métrica de avaliação de desempenho do sistema foi a taxa de atualização dos desenhos na cena, medida em quadros por segundo (*frames per second - fps*). Para cada trajetória foram executadas medições do valor médio de quadros por segundo de acordo com dois critérios: (1) sem o emprego do algoritmo de recorte, (2) com o emprego do algoritmo de recorte (*frustum culling*).

5. RESULTADOS

A Figura 5a ilustra os resultados obtidos a partir do sobrevôo #5 no qual não foi realizado qualquer tipo de descarte de células, isto é, todas as geometrias (triângulos) foram desenhadas a cada quadro da animação, obtendo-se em média 3,47 fps. A Figura 5b, por sua vez, representa os resultados obtidos quando utilizado o artifício de descarte de células, isto é, quando somente as geometrias visíveis para o observador são desenhadas.

Observa-se que o uso do algoritmo de recorte de células permite que as configurações migrem da zona de não-tempo real para a zona de tempo real (exceto a configuração Pentium 233MHz com placa de vídeo S3 Virge), de acordo com os valores preconizados por Lasseter (1987). A concavidade negativa das curvas da Figura 5b também ilustra a relação de custo/benefício entre o aumento da quantidade de células e a taxa de atualização de vídeo. Inicialmente, o aumento de divisões na grade incrementa a capacidade de desenho da placa de vídeo. Isto ocorre porque o custo dos testes de visibilidade das células é compensado pelo ganho obtido com o descarte de primitivas geométricas invisíveis na cena. Esta situação se mantém até que se atinge um ponto de inflexão o qual caracteriza um estado onde a relação custo/benefício torna-se ótima. Esta situação é caracterizada por uma taxa de

atualização de tela variando entre 22 e 40 fps, associada à 2⁵ divisões, isto é, um cenário dividido em 32x32 células. Em seguida, é iniciado um novo estado no qual constata-se uma depreciação da taxa de atualização de vídeo conforme o aumento do número de células de descarte. Neste caso, o custo computacional de testar todas as células de descarte aumenta mais rapidamente que o benefício computacional do próprio descarte. Gráficos similares foram produzidos para todas as trajetórias, e apresentaram resultados semelhantes.

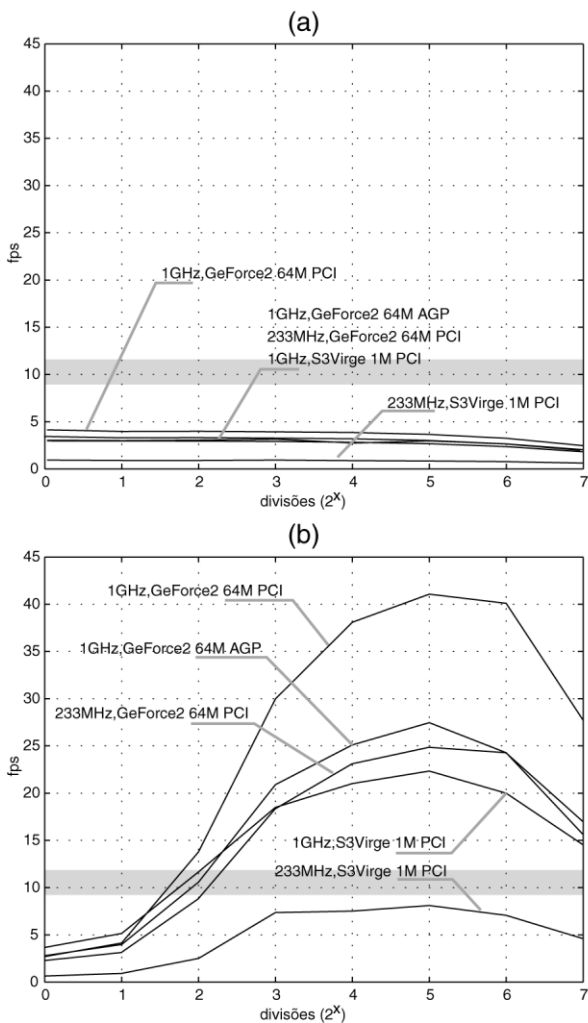


Fig. 5 - Desempenho associado à trajetória aleatórias #5 nos testes de desempenho: (a) sem utilizar algoritmo de descarte; (b) utilizando algoritmo de descarte.

A Figura 6a apresenta uma contagem média de células desenhadas para cada uma das trajetórias testadas quando empregado o algoritmo de recorte. Adicionalmente, também é apresentada a quantidade total destas mesmas células quando não há o uso do algoritmo de recorte, evidenciando um aumento significativo do número de objetos que seriam remetidos desnecessariamente para a placa de vídeo. Além disto, também se observa que as trajetórias do Grupo 2 (#7, #8 e #9) exigiram um maior número de células a serem desenhadas do que aquelas pertencentes ao Grupo 1, exceto o caso da trajetória #10. Entende-se

que a exceção a esta regra deve-se ao fato de que a referida trajetória combina características relevantes dos dois grupos, isto é, realiza um movimento pseudo-aleatório combinado com formas geométricas.

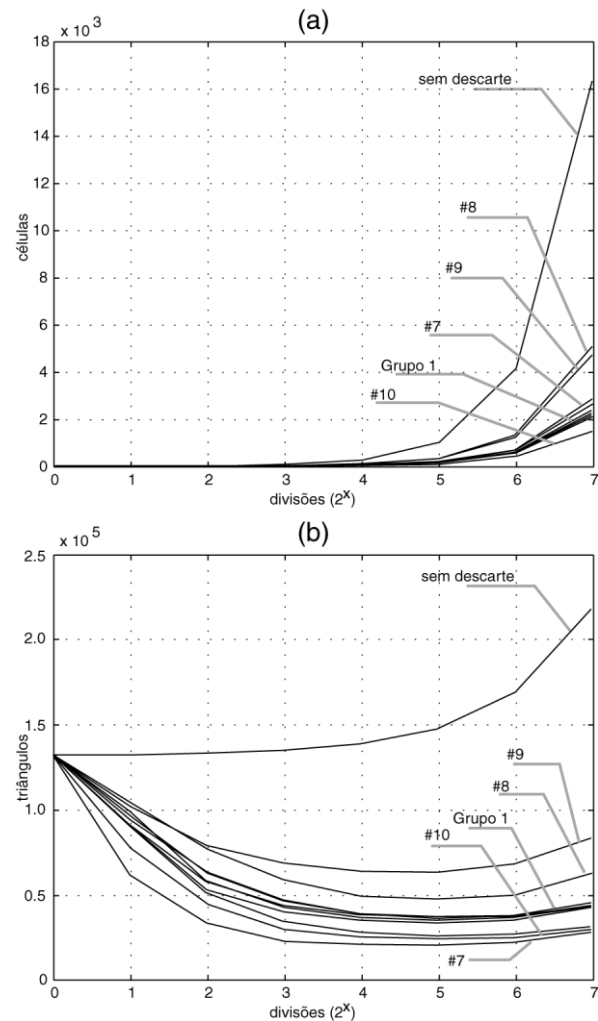


Fig. 6 - Valores médios associados ao algoritmo de descarte de objetos: (a) número de células visíveis; (b) número de triângulos visíveis.

A Figura 6b ilustra a quantidade média de triângulos remetidos para a placa de vídeo, de acordo com a quantidade de divisões do cenário. A opção de não empregar um algoritmo de descarte também aumenta significativamente o número de primitivas geométricas a serem desenhadas, quando comparada com os outros casos. Neste gráfico, as trajetórias do Grupo 1 também apresentaram desempenho similar, ficando agrupadas em torno dos mesmos valores.

6. CONCLUSÃO

A relação entre a resolução de um terreno, e seu dado associado, pode facilmente exceder a capacidade de processamento de uma placa de vídeo, o que em muitos casos, torna impossível uma aplicação em tempo real. Esta problemática torna-se ainda mais evidente quando o objetivo é viabilizar aplicações de

visualização 3D para cartas produzidas pelos órgãos de mapeamento sistemático nacional em ambientes computacionais com baixa disponibilidade de processamento.

Conforme descrito nos resultados da pesquisa, explicitou-se que uma aplicação de visualização 3D de terrenos utilizando dados reais é incapaz de prover uma taxa de atualização de vídeo compatível com um sistema de tempo real, independentemente do equipamento utilizado em sua execução. Através do uso do algoritmo de recorte, foi possível incrementar o desempenho desta aplicação para níveis superiores ao limite mínimo preconizado pela literatura clássica de animação.

A análise dos resultados permite afirmar que é viável a utilização dos dados cartográficos, disponíveis atualmente, em aplicações de visualização 3D de terrenos, em tempo real. A metodologia de pré-processamento de dados cartográficos proposta provou ser um manejo adequado para este propósito.

Durante a execução dos testes, observou-se que as trajetórias pseudo aleatórias do Grupo 1 apresentam padrões que podem ser associados ao sexo e à formação acadêmica dos indivíduos. Entretanto, face a quantidade diminuta de amostras para este tipo de estudo, espera-se abordar esta questão em um momento mais apropriado.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALLWORTH, S. T., **Introduction to Real Time Software Design**. The MacMillan Press, New York, 1981.
- ANCIÃES, C. L. C., **Transformação entre redes Geodésicas: uso de coordenadas 3D, 3D com restrição, e 2D**. *Dissertação de Mestrado*, IME, p.48, 2003.
- ASTLE, D., HAWKINS, K., LAMOTHE, A. **OpenGL Game Programming**. Premier, California, 2002.
- BULTZ, A., **Alternative Algorithm for Hilbert's Space-Filling Curve**. IEEE Transactions on Computer, pp.424-426, April 1971.
- COLE, A. J., **Compaction Techniques for Raster Scan Graphics Using Space-filling Curves**. In: The Computer Journal, pp. 87-92, 1987. v.30, no.1.
- FREGOSI, A. E., et alii, **Enfoque Clássico da Teoria de Controle**. Campus, São Paulo, 1980.
- KAMEL, I., FALOUTSOS, C., **On packing R-Trees**. In: 2nd International Conference on Information Knowledge Management, pp. 490-499, 1993.
- LAMOTHE, A. **Black Art of 3D Game Programming**. California, Waite Group, pp.813-816, 1995.
- LASSETER, John, **Principles of Traditional Animation Applied to 3D Computer Animation**, Computer Graphics, pp. 35-44, 1987. v.21, no.4.
- COSTA, R. S. N., **Sistemas Baseados em Conhecimento em Engenharia de Processos**. Dissertação de Mestrado, Instituto Militar de Engenharia, IME, Rio de Janeiro, 1992.
- LAURINI, R., **Graphical data Bases Built on Peano Space-Filling Curves**. In: Eurographics 85, pp.327-338, 1985.
- MARTIN, J., **Design of Real Time Computer Systems**, Prentice Hall, 1987.
- MELLO, F. L., FERREIRA, L. F. C., **Visualização Tridimensional de Terrenos: Triangulação dos Dados Altimétricos**. Revista Militar de Ciência e Tecnologia, p.51-59, 2005. v.22, no.2.
- MELLO, F. L., FERREIRA, L. F. C., **Visualização Tridimensional de Teatros de Guerra: o Pré-Processamento dos Dados**. Revista Militar de Ciência e Tecnologia, pp. 35-47, 2004. v.21, no.3.
- PATRICK, E., ANDERSON, D., **Mapping Multidimensional Space to One Dimension for Computer Output Display**. IEEE Transactions on Computers, v.C-17, pp.949-953, October 1968.
- VELHO, L., GOMES, J., **Space-Filling Curve Dither with Adaptive Clustering**. In: Proceedings of SIBGRAPI V, pp.1-9, November 1992.