

Visualização Interativa Tridimensional de Modelos de Terreno com Textura

PAULA FREDERICK¹, LUIZ CARLOS GUEDES², MARCELO GATTASS¹, PAULO CEZAR P. CARVALHO^{1,3}

¹ TeCGraf-Grupo de Tecnologia em Computação Gráfica, Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente 255, 22453-900 Rio de Janeiro, RJ, Brasil
gattass@inf.puc-rio.br

² Departamento de Ciência da Computação, Universidade Federal Fluminense
Praça Valonguinho s/n, Ed. Instituto de Matemática 4o andar, Centro, 24210-130 Niteroi, RJ, Brasil
guedes@dcc.uff.br

³ IMPA- Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110, 22460-320 Rio de Janeiro, RJ, Brasil
pcezar@impa.br

Abstract. Interactive tridimensional visualization of terrain models can be found in Geographical Information System and computer games. Both applications share the need for high performance algorithms that are tailored to produce textured images in interactive time, say at least 5 frames per second. Computer games are now developed to a point where this performance can be achieved in low-end home computers. The literature of computer games, however, is filled with “tricks” and pays very little attention to the underlying geometric model and image synthesis algorithm. This paper describes an on going research on interactive algorithms for landscape viewing. A preliminary evaluation of one popular algorithm is presented here.

Keywords: Terrain Models, Game Visualization, GIS.

Introdução

O estado da arte em programação de jogos de computador está bastante avançada em termos da eficiência de seus algoritmos.

Tipicamente empregada em jogos de jatos de caça voando sobre desertos apocalípticos, como Comanche, Fury3, Terminal Velocity e Magic Carpet 2, tal tecnologia começa a se difundir para outras áreas onde este nível de realismo é satisfatório. Aplicações pessoais como atlas geográficos [ATR] podem oferecer ao usuário recursos de visualização tridimensional apenas com o intuito de aumentar sua familiaridade com os principais aspectos do relevo pesquisado e não necessariamente apresentar uma visualização cem por cento real do mesmo. Até mesmo certas aplicações de sistemas geográficos de informação podem se utilizar destas técnicas de visualização interativa.

A literatura técnica sobre programação de jogos [LaMothe] [Freese] já reserva um espaço para a visualização interativa tridimensional de terrenos. Entretanto, a descrição das técnicas empregadas é cheia de termos como “truques” e é focada no paradigma “é bom por que funciona”.

Este trabalho apresenta resultados preliminares obtidos em um projeto cujo propósito é estudar em detalhes tais algoritmos, explicitando seus modelos de projeção, analisando as distorções causadas por tais modelos, comparando seu desempenho computacional e

investigando algoritmos alternativos mais precisos ou mais gerais.

Descrição do Problema

O modelo de terreno normalmente utilizado para jogos utiliza uma grade regular. É comum tal grade ser representada através de dois mapas: um de alturas e outro de textura (cor), como ilustra a Fig. 1.

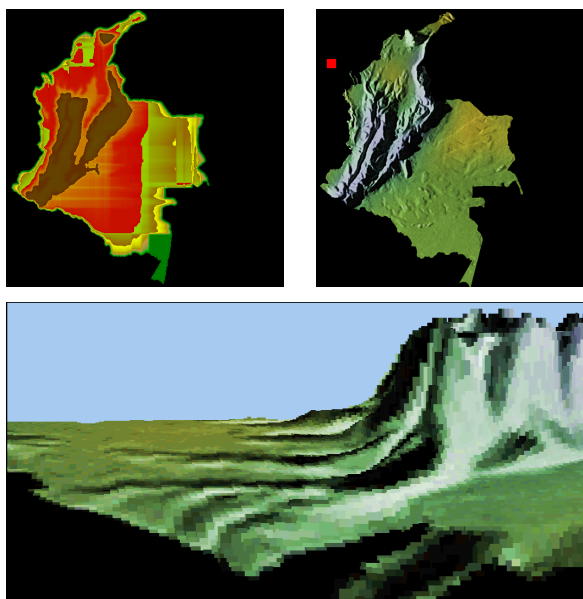


Fig. 1. Mapas de altura e textura da Colômbia, com visualização.

Os mapas da Fig. 1 são utilizados em [ATR] para visualizar o território da Colômbia. A Fig. 1 também mostra o resultado da visualização com o observador situado no ponto assinalado no mapa de texturas (superior direito). Os mapas de altura e textura são, em geral, armazenados em arquivos de imagens onde o valor de cada pixel representa, respectivamente, a altura do ponto e um indicador da cor. Normalmente esta cor já apresenta uma iluminação pré-calculada para aumentar o realismo da visualização.

A técnica mais utilizada para visualização de terrenos é a de lançamento de raios, que consiste em, para cada pixel da tela (considerada perpendicular ao solo), determinar o ponto em que o raio que parte do observador passando pelo pixel intercepta pela primeira vez a superfície do terreno. A cor do pixel é determinada consultando-se o mapa de texturas na posição do ponto de interseção. Em geral, este cálculo de interseção do raio com a superfície é custoso e deve ser evitado.

Uma estratégia mais eficiente [LaMothe] de lançamento de raios consiste em, para cada coluna da tela, lançar raios de cima para baixo, a partir da linha do horizonte. Para cada raio, é calculado o ponto de interseção com o plano horizontal de altura nula e levantada uma coluna de pixels com a cor correspondente no mapa de texturas e com altura obtida projetando a coluna de terreno sobre a tela. Como o algoritmo percorre o terreno de trás para frente, realiza o algoritmo do pintor [Foley], de modo que as colunas da frente escondem as partes invisíveis das colunas de trás. Como este algoritmo se orienta pelos pontos da tela, os pontos do mapa situados entre dois raios consecutivos não contribuem na determinação da coluna correspondente, fenômeno que se agrava nos pontos mais distantes do mapa. Outra desvantagem é que os pontos das colunas posteriores escondidos pelas colunas anteriores são desnecessariamente pintados na tela. Além disso, o algoritmo introduz distorções de visualização, causadas por calcular a interseção do raio com o plano de altura nula e não com a superfície do terreno.

Uma outra estratégia, que resulta em um algoritmo mais eficiente, processa os pixels de cada coluna da tela de baixo para cima. Para o primeiro raio visual, o algoritmo percorre as colunas de terreno por ele atravessadas, a partir do observador, até encontrar uma interseção [Freese]. Quando isso ocorre, um ou mais pixels são pintados com a textura da coluna atravessada, dependendo do número de raios subsequentes que ainda interceptam a mesma coluna. Quando, finalmente, um raio visual não mais intercepta a coluna, o algoritmo percorre este raio a partir da próxima coluna de terreno até encontrar uma nova interseção ou atingir uma profundidade máxima pré-estabelecida. Este processo admite uma implementação

extremamente otimizada, que emprega métodos incrementais.

Para tornar os algoritmos acima mais eficientes, é comum admitir-se que as colunas da tela estão situadas à mesma distância do observador e contidas em planos verticais uniformemente espaçados angularmente. Isto significa considerar que a tela de projeção é cilíndrica. Esta geometria de tela promove distorções na imagem. Em particular, retas horizontais se projetam segundo arcos de elipse como ilustra a Fig. 2 que apresenta uma imagem de um terreno horizontal obtida usando o método acima à esquerda de uma vista em que a tela de projeção é plana.

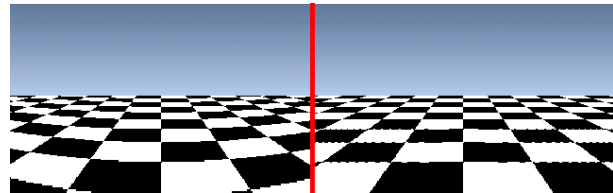


Fig. 2 Visualização de terreno horizontal

Deve-se observar que, no caso de um terreno mais acidentado, como o da Fig. 3, estas distorções são menos perceptíveis.

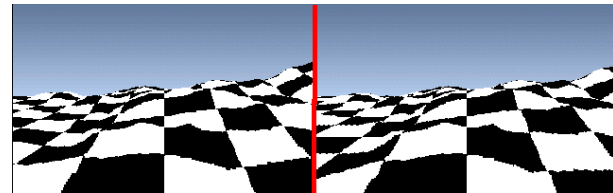


Fig. 3 Visualização de terreno acidentado

Plano de Trabalho

No decorrer deste trabalho, pretende-se investigar alternativas para algoritmos como os descritos acima, procurando eliminar distorções e torná-los mais gerais sem comprometer seu desempenho. Alguns resultados preliminares foram obtidos no desenvolvimento de algoritmos interativos para planos de projeção inclinados, mas as distorções introduzidas podem ser consideradas inaceitáveis dependendo da aplicação.

Agradecimentos

PETROBRAS, ATR Multimedia, MCT e CNPq.

Referências

- A. La Mothe, *Black Art of 3D Game Programming*, Waite Group Press, 1995.
- P. Freese, *More Tricks of the Game Programming Gurus*, Chapter 7, SAMS Publishing, 1995.
- J. Foley et. alli, *Computer Graphics - Principles and Practice*, Addison-Wesley, 1990.
- Enciclopédia Geográfica ATR*, ATR Multimedia, 1996.