Departamento de Engenharia Mecânica

UM AMBIENTE INTEGRADO DE SÍNTESE DE IMAGENS REALISTAS

Aluno: Icaro Bouças
Orientador: Marcelo Dreux

Justificativa

A Síntese de Imagens por Computador visa gerar, a partir da modelagem matemática, objetos no computador. A Síntese de Imagens Realistas tem como objetivo a geração cenas que se aproximem do mundo real. Há diversos métodos para geração de imagens por computador, não existindo um método ótimo, dependendo do tipo de objetos que deseja-se sintetizar.

Objetivo

O objetivo deste projeto é desenvolver um ambiente integrado para síntese de imagens por computador (RISE) a fim de disponibilizar uma ferramenta amigável para o ensino desta área nos cursos de graduação e pós-graduação.

Atualmente, estão disponíveis neste ambiente os algoritmos mais difundidos para Síntese de Imagens, a saber: Scanline, Ray Tracing e o Ray Tracing Distribuído. Os modelos de iluminação Phong-Shading e Gouraud-Shading, assim como o tratamento de texturas e anti-aliasing também fazem parte do sistema.

Metodologia

O ambiente computacional em desenvolvimento (RISE – Realistic Image Synthesis Environment) se propõe a integrar os principais algoritmos de síntese de imagens. Optou-se por iniciar pelos mais difundidos a saber: Ray Tracing [Kuchkuda 87] e [Whitted 70], Scanline [Dreux 88] e [Bouknight 70] e posteriormente o Ray Tracing Distribuído[Cook 01] .

Nessa nova fase de otimização do RISE, foi integrado ao seu ambiente o algoritmo de Ray Tracing Distribuído (figura 1). Atualmente o ambiente do RISE oferece ao usuário grande flexibilidade para gerar cenas com os três diferentes algoritmos citados. Permite ao usuário carregar cenas via arquivo ou então através de entrada de dados pelo teclado.

O projeto continua sendo desenvolvido na linguagem C++, utilizando o compilador Visual C++ versão 6.0. A interface com o usuário foi desenvolvida a partir de bibliotecas do próprio Windows, Glut e Glui.

Conclusão / Resultados

No estágio atual, os algoritmos de Ray Tracing, Scanline e Ray-Tracing Distribuído já estão integrados em um único programa (RISE). Há funções que permitem ao usuário carregar e salvar cenários e mundos, utilizando qualquer dos três algoritmos. As imagens geradas podem ser gravadas no formato bitmap (24 bits) do Windows, formato escolhido por sua grande portabilidade. Há algumas melhorias de qualidade visual (*anti-aliasing*) e peformance (inclusão de *bounding Volumes* (tanto *sphere* quanto *box)* [Ritter 90]), além de um carregador de malhas de triângulos do tipo MD2 [Hawkings 01], possibilitanto, portanto, importar diversos modelos disponíveis na Internet (Figura 1). No algoritmo de Ray Tracing já há mapeamento de textura para as primitivas além da técnica de *Bump Mapping*. No algoritmo de Scanline, ocorrem problemas numéricos onde prejudicam a qualidade visual da imagem. Estudos para corrigir esses problemas já foram iniciados. Como trabalhos futuros, pode-se implementar um algoritmo de Ray Tracing Progressivo [Araujo96] e um algoritmo híbrido (Ray Tracing + Scanline) assim como técnicas de partição espacial (BSP) para uma melhoria significativa na velocidade de processamento do Ray Tracing.

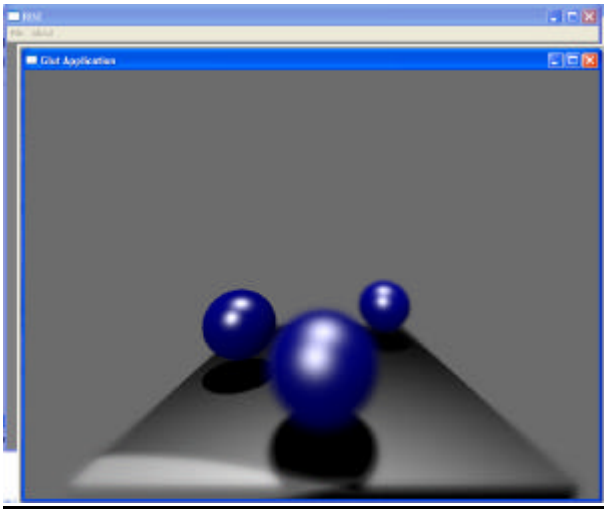

Figura 1 – Imagem gerada com Ray Tracing Distribuído mostrando o efeito de profundidade de campo.

Bibliografia

[Araujo 96] - R. Araujo, M. Gattass, M. Dreux, "Refinamento Progressivo da Cena com Traçado de Raios Distribuído", SIBGRAPI 96, IX Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, Caxambu, MG, outubro de 1996, pp. 1-8.

[Bouknight 70] - W.J. Bouknight, A Procedure for Generation of Three-dimensional Half-toned Computer Graphics Representations, Communications of the ACM, v13, n9, September 1970, pp 527-536.

[Cook 01] – Robert L. Cook, Stochastic Sampling and Distributed Ray tracing, In An Introduction to Ray Tracing, Academic Press, London, U.K, 1989, pp 161-199.

[Dreux 88] - M. Dreux, A Simple Effective Linear-Growth Scanline Algorithm, ICONCG 88, International Conference on Computer Graphics, Singapore, September 1988, 209-218.

[Hawkings 01] - Kevin Hawkings and Dave Astle, OpenGL Game Programming, Ed. Prima Tech's Game Development, 2001.

[Kuchkuda 87] - R. Kuchkuda, An Introductio to Ray Tracing, Theoretical Foundations of Computer Graphics and CAD, Italy, 1987.

[Ritter 90] – Jack Ritter , An efficient Bounding Sphere , in Andrew S. Glassner, ed. , Graphic Gems , pp 301-303, 1990.

[Whitted 80] -T. Whitted, An Improved Illumination Model for Shaded Display, Communications of the ACM, v23, n6, June 1980, pp 343-349.