

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007



UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2007.1

SIMULAÇÃO DE ILUMINAÇÃO VOLUMÉTRICA

TRABALHO DE GRADUAÇÃO EM COMPUTAÇÃO GRÁFICA

Aluno - Saulo Andrade Pessoa, sap@cin.ufpe.br.
Orientador - Sílvio de Barros Melo, sbm@cin.ufpe.br.

22 de agosto de 2007

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

“A arrogância científica é a negação da própria ciência...”

Frase dada em uma entrevista pelo pesquisador Gonçalo Guimarães.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

RESUMO

Em computação gráfica, inúmeros métodos vêm sendo criados para aumentar o realismo das imagens sintetizadas. Tais métodos são inspirados principalmente em fenômenos físicos. Um fenômeno bastante estudado é a interação ocorrida entre a luz e os objetos, porém, tal interação é geralmente simplificada devido às limitações computacionais. Uma simplificação comumente realizada é a de desconsiderar a interação ocorrida entre a luz e a atmosfera, tanto durante o deslocamento da luz do ponto luminoso ao objeto quanto do objeto ao observador. Tal simplificação pode ser entendida como se a luz estivesse viajando no vácuo, onde o meio não é participativo, fazendo com que as imagens sintetizadas careçam dos efeitos atmosféricos comumente observados no mundo real.

Inúmeros efeitos atmosféricos podem ser exemplificados, como a variação da cor do céu de azul para laranja durante o crepúsculo, ou quando os faróis de um automóvel são ligados em um dia com neblina. Além destes, pode-se citar os efeitos geralmente empregados em espetáculos, onde gelo seco é espalhado sobre o palco e holofotes coloridos são direcionados sobre as partículas suspensas no ar. O foco do estudo deste trabalho será neste último efeito – comumente chamado de iluminação volumétrica – e contará com a implementação de um simulador interativo.

Palavras-chave: Computação Gráfica, Simulação, Meio Participativo, Iluminação Volumétrica, GPU.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

DEDICATÓRIA

Dedico este trabalho a todos os pesquisadores e profissionais das mais diversas áreas, pois através do desenvolvimento e aplicação do conhecimento científico o mundo pode se tornar um lugar melhor para viver.

AGRADECIMENTOS

Agradeço primeiramente a minha família por terem me apoiado mesmo nas decisões que pareciam mais insanas, e por terem me dado oportunidade de ter acesso a educação de boa qualidade.

Agradeço a meu orientador, Sílvio Melo, pelo tempo que dedicou me auxiliando neste trabalho.

Agradeço a meu irmão, Dinaldo Pessoa, pela orientação dada durante a escrita da monografia.

Agradeço a todos os meus amigos e colegas de trabalho que além de me alertarem para começar o quanto antes possível a escrita da monografia, sempre me incentivaram.

Agradeço novamente a Sílvio Melo e Dinaldo Pessoa pelas contribuições dadas revisando a monografia.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

SUMÁRIO

1. INTRODUÇÃO	7
1.1 MOTIVAÇÃO.....	7
1.2 OBJETIVOS	8
1.3 RELEVÂNCIA DO TRABALHO	9
1.4 METODOLOGIA APLICADA	10
1.5 ORGANIZAÇÃO DA MONOGRAFIA	10
2. CONTEXTO.....	12
2.1 MÉTODOS DE INTEGRAÇÃO NÃO APLICÁVEIS A GPUS	13
2.2 MÉTODOS DE INTEGRAÇÃO APLICÁVEIS A GPUS	14
3. SOLUÇÃO PROPOSTA.....	15
3.1 ESPALHAMENTO DA LUZ	15
3.1.1 <i>Espalhamento de Rayleigh</i>	15
3.1.2 <i>Espalhamento de Mie</i>	15
3.2 MODELO DE SHADING.....	18
3.3 TÉCNICA DE RENDERIZAÇÃO	21
3.3.1 <i>Discretização</i>	21
3.3.2 <i>Função de Iluminação</i>	23
3.3.3 <i>Função de Visibilidade</i>	24
4. RESULTADOS E PROBLEMAS ENCONTRADOS	27
5. TRABALHOS FUTUROS	29
REFERÊNCIAS	31
ANEXO A - IMAGENS DA APLICAÇÃO	33

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

LISTA DE FIGURAS

Figura 1-1. O curta-metragem Lifted da Pixar à esquerda e o jogo Project Offset da Offset Software à direita.	8
Figura 1-2. Atmosfera sombria, misteriosa e aterrorizante do jogo Alan Wake.....	9
Figura 2-1. Uma das primeiras aplicações de efeitos de iluminação volumétrica em jogos eletrônicos.	12
Figura 2-2. Iluminação volumétrica na Torque Game Engine.....	13
Figura 3-1. Raios crepusculares.....	16
Figura 3-2. Os tipos de interação que a luz pode sofrer.	16
Figura 3-3. Modelo de <i>shading</i> proposto.	19
Figura 3-4. Volume de sombra gerado.....	20
Figura 3-5. Como os planos de amostragem devem ser dispostos e subdivididos.....	21
Figura 3-6. Exemplos de mapas de iluminação.	24
Figura 3-7. Como gerar e aplicar os mapas de visibilidade.	26
Figura 4-1. Problemas encontrados na simulação. Em (a), o aparecimento de faixas; em (b), <i>aliasing</i> ; em (c), quantização; e em (d), riscos devido à perpendicularidade entre plano de amostragem e a direção da fonte da luz. Observe, em (d), no detalhe superior, o mapa de visibilidade gerado.....	28
Figura A-1. Iluminação volumétrica com $n = 40$, QUADs = 20x20, $\sigma = 0.1$, $K = 20$, $\Delta_t = 0.1$ e executando a 41 <i>fps</i> . Mapa de iluminação utilizado à direita.	33
Figura A-2. Iluminação volumétrica com $n = 50$, QUADs = 30x30, $\sigma = 0.9$, $K = 20$, $\Delta_t = 0.04$ e executando a 28 <i>fps</i> . Mapa de iluminação utilizado à direita.	33
Figura A-3. Iluminação volumétrica com $n = 40$, QUADs = 30x30, $\sigma = 0.5$, $K = 30$, $\Delta_t = 0.07$ e executando a 37 <i>fps</i> . Mapa de iluminação utilizado à direita.	34
Figura A-4. Iluminação volumétrica com $n = 45$, QUADs = 20x20, $\sigma = 0.3$, $K = 25$, $\Delta_t = 0.06$ e executando a 43 <i>fps</i> . Mapa de iluminação utilizado à direita.	34
Figura A-5. Iluminação volumétrica com $n = 55$, QUADs = 10x10, $\sigma = 0.2$, $K = 25$, $\Delta_t = 0.05$ e executando a 28 <i>fps</i> . Mapa de iluminação utilizado à direita.	35
Figura A-6. Iluminação volumétrica com $n = 50$, QUADs = 30x30, $\sigma = 0.2$, $K = 20$, $\Delta_t = 0.07$ e executando a 28 <i>fps</i> . Mapa de iluminação utilizado à direita.	35

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

1. Introdução

Este primeiro capítulo visa dar uma breve introdução do tema abordado e mostrar as motivações que levaram a sua escolha. Os objetivos e relevância do trabalho também serão explicitados, juntamente com a metodologia adotada para sua realização. Por último, a organização proposta para o este documento será introduzida.

1.1 Motivação

A busca por realismo em cenas sintetizadas por computador é uma corrida constante dos pesquisadores em computação gráfica. Com a crescente capacidade de processamento dos microprocessadores e com o advento das placas gráficas, não é de se admirar que cada vez mais as imagens sintetizadas pareçam reais. Hoje, as placas gráficas possuem poderosas GPUs (*Graphics Processing Unit*), que são microprocessadores especializados em gráficos e que alcançam a ordem de centenas de GFLOPS de performance. Outro fator que tem contribuído para o aumento de realismo é a criação de modelos que tratam de maneira mais fiel os fenômenos físicos ocorridos na natureza (principalmente a interação entre luz e os diferentes tipos de materiais dos objetos). Neste contexto, um assunto que vêm ganhando destaque é a simulação de meios participativos.

Determinados fenômenos naturais só podem ser reproduzidos em computação gráfica caso o meio seja considerado como participativo. Com um meio participativo, a luz interage com as partículas suspensas no ar à medida que se propaga. Como o planeta Terra é naturalmente envolvido por uma camada gasosa (atmosfera), a interação entre luz e meio se torna inevitável. Além da atmosfera, a luz também pode interagir com qualquer outro tipo de partícula que esteja suspensa no ar, como, por exemplo, fumaça, gelo seco, vapor d'água, poeira, etc. Em geral, os efeitos visuais gerados devido à interação entre a luz e as partículas suspensas são chamados de efeitos atmosféricos, porém, quando o efeito é gerado por partículas relativamente grandes e quando há a formação de um volume de luz, ele é chamado de iluminação volumétrica. Este último efeito será o foco deste trabalho.

Inúmeras indústrias demandam a criação de imagens foto-realistas. Entre elas, as que mais se destacam atualmente são a indústria cinematográfica e a indústria de jogos eletrônicos. A indústria cinematográfica prioriza a qualidade visual das imagens geradas, ou seja, é válido se gastar alguns segundos ou até mesmo minutos a mais para se obter resultados melhores. Já a indústria de jogos eletrônicos prioriza o tempo gasto na geração das imagens, ou seja, é melhor que um jogo execute a pelo menos 24 fps (*frames per second*) com um visual razoável ao invés dele ser visualmente perfeito e executar a uma taxa muito baixa. Visto que cada indústria possui requisitos diferentes, métodos, técnicas e modelos que se adequem a cada realidade devem ser criados. A Figura 1-1 mostra dois exemplos

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

recentes da aplicação do efeito de iluminação volumétrica, à esquerda um curta da Pixar [Pixar] e à direita um jogo da Offset Software [OffsetSoftware].



Figura 1-1. O curta-metragem Lifted da Pixar à esquerda e o jogo Project Offset da Offset Software à direita.

Vale salientar que este trabalho visa focar em simulação de iluminação volumétrica em tempo real, logo, a simplificação de modelos mais complexos será realizada de modo que o desempenho da aplicação seja privilegiado.

1.2 Objetivos

O objetivo deste trabalho envolve a realização da revisão bibliográfica do estado da arte sobre efeitos atmosféricos, seguida da implementação de um simulador interativo. O foco do estudo será sobre o efeito atmosférico chamado de iluminação volumétrica, onde a luz, ao se propagar por um meio com partículas suspensas, é absorvida e espalhada ao longo de sua trajetória até alcançar o observador, tornando possível, assim, a visualização de um volume de luz. Tal efeito atmosférico é explicado pelo Espalhamento de Mie, entretanto, inúmeras simplificações devem ser realizadas devido às limitações computacionais. Ao final da revisão bibliográfica um método adequado deverá ser escolhido e adaptado para fazer uso das novas tecnologias.

A implementação do simulador deverá ser feita utilizando as novas tecnologias empregadas nas placas gráficas. Uma das tecnologias empregadas será a API gráfica OpenGL [OpenGL], de modo que os recursos da placa gráfica possam ser acessados. O simulador implementado deverá possibilitar que o usuário navegue livremente pela cena e possa alterar alguns parâmetros da simulação, como, por exemplo, a densidade das partículas espalhadas no ar ou o nível de amostragem que será realizado pelo algoritmo. A alteração destes parâmetros durante a simulação possibilita que o usuário possa avaliar o custo/benefício do resultado – beleza visual e taxa de fps alcançada – gerado pelo

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

simulador, e assim poder escolher os parâmetros que melhor satisfaçam suas necessidades.

1.3 Relevância do Trabalho

Não é de hoje que o homem vem estudando a luz na tentativa de entender seu comportamento durante sua propagação e colisão com obstáculos. Hoje, os resultados de tais estudos embasam muitos dos métodos criados no âmbito da computação gráfica. Além de tudo, a crescente demanda da indústria cinematográfica e da indústria de jogos por métodos que reproduzam efeitos naturais de maneira mais realista tem aumentado ainda mais as pesquisas na área. Atualmente, inúmeras empresas de efeitos especiais além de atuarem na produção também possuem uma forte linha de pesquisa, como é o caso da Pixar. No caso dessa, muitas das tecnologias empregadas em seus filmes são desenvolvidas na própria empresa, o que além de evitar o custo com licenciamento de tecnologias de terceiros, garante uma singularidade em suas produções.

Especificamente falando sobre iluminação volumétrica, inúmeros jogos eletrônicos já fazem uso deste efeito. Um título bastante novo - com previsão de lançamento para 2008 - que usa e abusa de efeitos de iluminação volumétrica é o Alan Wake [AlanWake]. Este título, por se enquadrar no gênero "terror de ação psicológica" necessita transmitir para o jogador uma atmosfera sombria, misteriosa e aterrorizante, e é exatamente este tipo de sensação que a iluminação volumétrica transmite. A Figura 1-2 mostra o tipo de ambientação pretendida e os efeitos de iluminação volumétrica do jogo Alan Wake.



Figura 1-2. Atmosfera sombria, misteriosa e aterrorizante do jogo Alan Wake.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

À medida que as GPUs foram ficando mais rápidas, inúmeros efeitos que antes só eram possíveis em imagens pré-renderizadas passaram a ser incorporados em aplicações interativas. Como esta evolução só está no começo, a tendência é que cada vez mais efeitos sofisticados deixem de ser privilégio apenas da indústria cinematográfica e passem a ser incorporados em jogos.

1.4 Metodologia Aplicada

A metodologia adotada basicamente constitui-se das etapas de pesquisa, desenvolvimento e documentação. Cada etapa tinha um propósito específico e elas foram realizadas em seqüência.

As atividades realizadas durante a pesquisa foram basicamente de revisão bibliográfica e de documentação da bibliografia coletada.

Durante o desenvolvimento, inúmeras atividades foram realizadas. A primeira delas foi a definição do modelo de *shading*, que foi seguida da escolha da técnica de renderização e das tecnologias de desenvolvimento. Com as tecnologias escolhidas, foi realizado um estudo superficial delas. Ao fim desta atividade, foram implementadas, fazendo uso do conhecimento obtido anteriormente, as funcionalidades básicas da aplicação, como: o contador de fps (que serviu como medidor de performance); o tratamento de entradas e a navegação da câmera. Em seguida, um estudo mais minucioso foi realizado com intuito de averiguar as funcionalidades das extensões de OpenGL. Esse estudo contou com a implementação de protótipos. Por fim, o modelo de *shading* e a técnica de renderização foram implementados, finalizando-se a implementação do simulador.

A etapa de documentação tratou do que dizia respeito às atividades de redação da monografia. Durante esta etapa também foram realizadas atividades de revisão e correção.

1.5 Organização da Monografia

Esta monografia foi dividida em 5 capítulos de modo que ela pudesse ser melhor organizada. Os capítulos seguintes e seus respectivos propósitos são:

Capítulo 2: Contexto

Explicar detalhadamente em qual contexto e com quais outras áreas da computação o trabalho se relaciona. Este capítulo também mostra a evolução dos estudos sobre simulação de iluminação volumétrica, fazendo uma retrospectiva pelos diversos métodos propostos.

Capítulo 3: Solução Proposta

Apresentar a solução proposta pelo trabalho passando pelo modelo de *shading* e técnica de renderização escolhida. Este capítulo também trata de algumas questões relativas à otimização.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

Capítulo 4: Resultados e Problemas Encontrados

Mostrar os resultados obtidos e relatar os problemas e limitações encontradas na solução proposta.

Capítulo 5: Trabalhos Futuros

Sugerir aprimoramentos que possam ser realizados sobre o trabalho para melhorar seus resultados.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

2. Contexto

Inúmeros efeitos atmosféricos [Cowley03] podem ser exemplificados, como a variação da cor do céu de azul para laranja durante o crepúsculo, ou quando os faróis de um automóvel são ligados em um dia com neblina, ou até mesmo a formação dos arco-íris. Além destes, pode-se citar os efeitos geralmente empregados em espetáculos, onde gelo seco é espalhado sobre o palco e holofotes coloridos são direcionados sobre as partículas suspensas no ar. Este último efeito torna possível a visualização de um volume de luz com a forma de um cone, além de também explicitar os volumes de sombra gerados devido à oclusão causada pelos objetos dentro do volume de luz (caso exista algum). O objetivo deste trabalho é propor uma técnica de renderização que seja capaz de gerar o efeito de iluminação volumétrica.

Desde muito tempo que efeitos de iluminação volumétrica vêm sendo empregados em jogos eletrônicos, entretanto, naturalmente, as primeiras tentativas não obtiveram resultados tão convincentes. Devido às limitações tecnológicas da época, tais efeitos eram obtidos através de formas nada condizentes com o que realmente ocorre na natureza. A Figura 2-1 mostra o jogo Cosmic Ark desenvolvido pela Imagic em 1982 para o Atari 2600. Observe que neste, à direita da nave mãe, existe um disco voador emitindo um feixe de luz na tentativa de abduzir um espécime alienígena.



Figura 2-1. Uma das primeiras aplicações de efeitos de iluminação volumétrica em jogos eletrônicos.

À medida que os jogos eletrônicos foram se popularizando, os processos empregados durante seu desenvolvimento foram evoluindo. Ferramentas específicas foram criadas de modo que o desenvolvimento tornou-se mais rápido. Dentre estas ferramentas, existem os famosos motores de jogo, que já incorporam inúmeras funcionalidades para se desenvolver jogos. Em geral, os motores de jogo

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

já possuem implementadas inúmeras técnicas de iluminação, e como já era de se esperar, alguns possuem recursos para gerar efeitos de iluminação volumétrica. A Figura 2-2 mostra um jogo demonstração desenvolvido na TGE (*Torque Game Engine*) [Torque], no qual um ogro encontra-se embaixo de uma luz do tipo volumétrica. A técnica utilizada pela TGE é similar a utilizada em renderização volumétrica, onde inúmeros planos são dispostos paralelamente e ao final eles são acumulados no *color buffer* habilitando-se o *alpha blending*. Vale ressaltar que na TGE não foi tratada a geração dos volumes de sombra, como pode ser observado embaixo do ogro.



Figura 2-2. Iluminação volumétrica na Torque Game Engine.

Atualmente, uma grande quantidade de pesquisadores já realizaram estudos e desenvolveram métodos para simulação de iluminação volumétrica. Basicamente, tais métodos resolvem o problema integrando-se numericamente a luz que é espalhada pelas partículas suspensas no ar ao longo do raio de visão. Os métodos propostos diferem basicamente em como a integral é calculada, o que leva a classificação dos métodos em dois tipos: aplicáveis a GPUs e os não aplicáveis a GPUs. Os primeiros realizam a integral utilizando recursos específicos – como a renderização de polígonos – do hardware gráfico, enquanto os segundos realizam tal operação na CPU (*Central Processing Unit*). Naturalmente, os métodos aplicáveis a GPUs serão o foco deste trabalho, visto que eles são os únicos passíveis de execução em tempo real.

2.1 Métodos de Integração Não Aplicáveis a GPUs

Dentre estes, Nishita [Nishita87] propôs um método baseado no traçado de raios para encontrar os pontos de interseção entre os volumes de luz e o raio de visão. Tal método obtém bons resultados, entretanto, demanda muito tempo de processamento. Jensen [Jensen98] fez uso de mapas de fótons para obter tal efeito.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

Este último método é tão sofisticado que consegue até gerar efeito de iluminação volumétrica para iluminação cáustica. Por serem computacionalmente custosos, estes dois últimos métodos não são adequados para aplicações em tempo real.

2.2 Métodos de Integração Aplicáveis a GPUs

Estes métodos só começaram a surgir com o início da popularização das placas gráficas no final da década passada. Dobashi [Dobashi00] propôs um método aplicável a GPUs que faz uso de técnicas de renderização volumétrica para integrar a luz espalhada. Por fazer uso de planos de amostragem e utilizar texturas representando a luz, este método facilmente consegue representar uma distribuição de luz não uniforme. Dobashi [Dobashi02] propôs otimizações para o método anterior de modo que a simulação executou mais rápida e obteve melhores resultados. James [James03] e Zhu [Zhu04] propuseram métodos que integram a luz espalhada utilizando *shaders* programáveis e que fazem uso de geometria para representar os volumes de luz. Estes dois últimos métodos necessitam que a geometria que representa o volume de luz e o volume de sombra seja criada manualmente. Mitchell [Mitchell04] propôs uma abordagem para tratar o caso da densidade das partículas não ser uniforme, além de propor novas otimizações sobre o método do Dobashi [Dobashi00].

Para este trabalho, o método do Dobashi [Dobashi00] foi tomado como referência principal devido à sua capacidade de representar uma distribuição de luz não uniforme através da utilização de mapas de iluminação.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

3. Solução Proposta

Este capítulo visa apresentar a solução proposta por este trabalho. Para tal, uma explicação de como o espalhamento da luz ocorre será dada, mostrando quais teorias são aplicáveis a cada tipo de espalhamento. Também será apresentado o modelo de *shading* utilizado e, ao final, será mostrada a técnica de renderização escolhida.

A solução proposta por este trabalho baseia-se principalmente no método criado por Dobashi [Dobashi00]. Por fazer uso de recursos específicos da GPU, este método mostrou-se bastante adequado para ser utilizado em aplicações interativas. Outro fator que levou à sua escolha – e este de fato foi o que mais influenciou a sua escolha dentre os métodos aplicáveis a GPUs – foi a sua capacidade de representar uma distribuição de luz não uniforme, que é conseguida através da utilização de mapas de iluminação.

3.1 Espalhamento da luz

Efeitos atmosféricos ocorrem devido ao espalhamento e absorção da luz ao se chocar com partículas suspensas no ar. O espalhamento pode ocorrer diferentemente para cada comprimento de onda da luz, isto é, comprimentos de onda diferentes podem ser espalhados em direções diferentes dependendo do tipo de partículas suspensas. Existem basicamente dois tipos de espalhamento, o Espalhamento de Rayleigh e o Espalhamento de Mie.

3.1.1 *Espalhamento de Rayleigh*

Em atmosferas secas, a forma primária de espalhamento é o Espalhamento de Rayleigh. É este tipo de espalhamento que dá o tom azul ou laranja do céu em diferentes horários do dia. Este tipo de espalhamento ocorre diferentemente para cada comprimento de onda dependendo do tipo de partícula suspensa e do ângulo de incidência da luz.

O Espalhamento de Rayleigh ocorre quando o tamanho das partículas é equiparável ao comprimento de onda da luz, ou seja, as partículas necessitam ser muito pequenas para que possa haver o Espalhamento de Rayleigh. Outro fator que influencia a observação do Espalhamento de Rayleigh é a distância percorrida pela luz. Devido à influência das partículas sobre a luz ser muito pequena, este tipo de espalhamento só pode ser observado quando a luz viaja grandes distâncias.

Este não é o tipo de espalhamento que será focado durante este trabalho, pois não é ele quem explica a formação dos volumes de luz.

3.1.2 *Espalhamento de Mie*

Em atmosferas pesadas, com partículas “grandes”, como água ou fumaça, o tipo de espalhamento predominante é o Espalhamento de Mie. É este tipo de

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

espalhamento que gera os efeitos de iluminação volumétrica. Este tipo de espalhamento também explica o aparecimento de feixes de luz oriundos das nuvens em dias nublados e úmidos (estes raios são chamados de raios crepusculares) [James03]. A Figura 3-1 mostra os raios crepusculares observados em um dia úmido.



Figura 3-1. Raios crepusculares.

O Espalhamento de Mie ocorre quando o tamanho das partículas é grande em relação ao tamanho do comprimento de onda da luz. O efeito deste espalhamento pode ser facilmente observado mesmo em pequenos intervalos de espaço, pois as partículas influenciam consideravelmente a trajetória da luz.

Existem quatro tipos de interação que a luz pode sofrer: absorção; emissão; espalhamento e ajuntamento [Biri02]. A Figura 3-2 mostra os tipos de interações que podem ocorrer durante a propagação da luz.

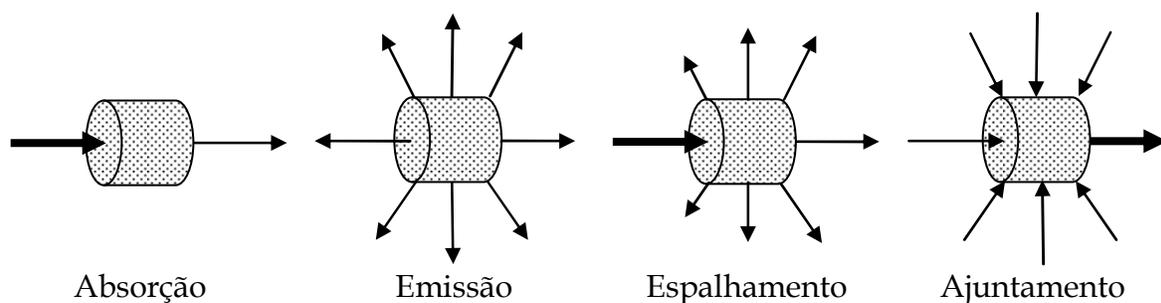


Figura 3-2. Os tipos de interação que a luz pode sofrer.

Os tipos de interação e a relevância de cada um para a simulação de iluminação volumétrica são explicados em detalhes a seguir.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

Absorção

Este tipo de interação é o maior responsável pela perda de energia luminosa. Esta perda é devido às colisões que ocorrem entre a luz e as partículas suspensas no ar. Como na natureza nada se perde, a energia luminosa absorvida na verdade é transformada em energia térmica. A densidade das partículas tem papel fundamental na absorção, isto é, quanto maior a densidade das partículas maior será a absorção do meio.

A absorção influencia consideravelmente o resultado final da simulação. Na prática, quando a densidade das partículas é baixa, o resultado obtido mostra que a luz consegue percorrer um caminho mais longo do que no caso em que a densidade das partículas é alta, o que acarreta na exibição de um feixe de luz mais longo.

Emissão

A emissão é o processo inverso ao da absorção, ou seja, ao invés da energia luminosa ser transformada em energia térmica, o fato ocorrido é que outro tipo de energia - como a energia elétrica - é transformada em energia luminosa. Este tipo de interação é muito rara e só ocorre com determinados tipos de partículas. Além do tipo de partícula, o meio também deve ser favorável para que ela possa ocorrer. Um bom exemplo deste tipo de interação é o empregado nas lâmpadas de néon, na qual o gás néon quando está submetido a uma baixa pressão e uma corrente elétrica é capaz de produzir luz.

Este tipo de interação não é considerado para a simulação de iluminação volumétrica, pois se trata de um caso muito particular e que pouco se relaciona com a geração dos volumes de luz.

Espalhamento

Este tipo de interação é o mais importante para a simulação de iluminação volumétrica, pois é devido a ele que é possível a visualização dos volumes de luz. Quando a luz se propaga em um ambiente com partículas suspensas, ela é espalhada em todas as direções fazendo com que o observador consiga enxergá-la mesmo quando posicionado em um local que não é diretamente atingido. As partículas em suspensão funcionam similarmente a um anteparo alterando a trajetória de deslocamento da luz.

A quantidade de luz espalhada em cada direção depende principalmente da direção em que a luz incidiu sobre as partículas. A relação entre quantidade de luz e direção do espalhamento é modelada através de uma função chamada função de fase. Funções de fase diferentes servem para modelar diferentes condições atmosféricas.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

Como dito anteriormente, este tipo de interação é o mais importante para simulação de iluminação volumétrica, e conseqüentemente será levado em consideração.

Ajuntamento

O ajuntamento pode ser entendido como sendo o processo inverso ao espalhamento. Neste tipo de interação, os raios de luz que foram anteriormente espalhados podem ocasionalmente ser direcionados para uma mesma trajetória. Quando isto ocorre, a intensidade luminosa resultante aumenta devido à contribuição de energia de cada direção. Este tipo de interação só ocorre devido aos múltiplos espalhamentos sofrido pela luz.

O ajuntamento certamente influencia o resultado final, entretanto, ele não será utilizado devido ao alto custo computacional que isso implicaria. Para que este tipo de interação pudesse ser tratado, os múltiplos espalhamentos citados anteriormente deveriam ser de alguma forma registrados, e é aí onde está todo o problema.

3.2 Modelo de *Shading*

Inúmeros modelos de *shading* para simulação de iluminação volumétrica já foram propostos. O trabalho realizado por Dobashi [Dobashi00] foi tomado como principal fonte de referência devido à sua adequação a GPUs. A adequação a GPUs vem do fato do modelo de *shading* ser simples e da utilização de recursos do *hardware* gráfico para realizar a computação.

Por uma questão de simplicidade, será assumido o caso em que existe apenas uma fonte de luz na cena. O método pode ser facilmente estendido para suportar múltiplas fontes de luz apenas repetindo-se o mesmo passo para tantas quantas forem as fontes de luz na cena, ou seja, calculando a contribuição de cada uma.

Basicamente, o objetivo final é obter a intensidade de luz que chega ao observador – daqui em diante câmera – que é expressa como

$$I_c = I_o\beta(T) + \int_0^T F(\alpha)H(r)I_p(r)\beta(t)\alpha dt, \quad (1)$$

onde I_c é a intensidade de luz que chega à câmera, I_o é a intensidade radiada pelo objeto, T é a distância entre o objeto e a câmera, $\beta(T)$ é a taxa de absorção devido as partículas suspensas entre o objeto e a câmera, α é o ângulo de fase, $F(\alpha)$ é a função de fase, r é a distância entre o ponto P e a fonte de luz, $H(r)$ é a função de visibilidade que retorna 1 caso o ponto P seja visível da fonte de luz ou 0 caso não seja visível, $I_p(r)$ é a intensidade de luz que chega ao ponto P , t é a

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

distância entre o ponto P e a câmera, $\beta(t)$ é a taxa de absorção devido às partículas suspensas entre o ponto P e a câmera, σ é o coeficiente de extinção que é diretamente proporcional à densidade das partículas suspensas. A Figura 3-3 ilustra bem estes termos.

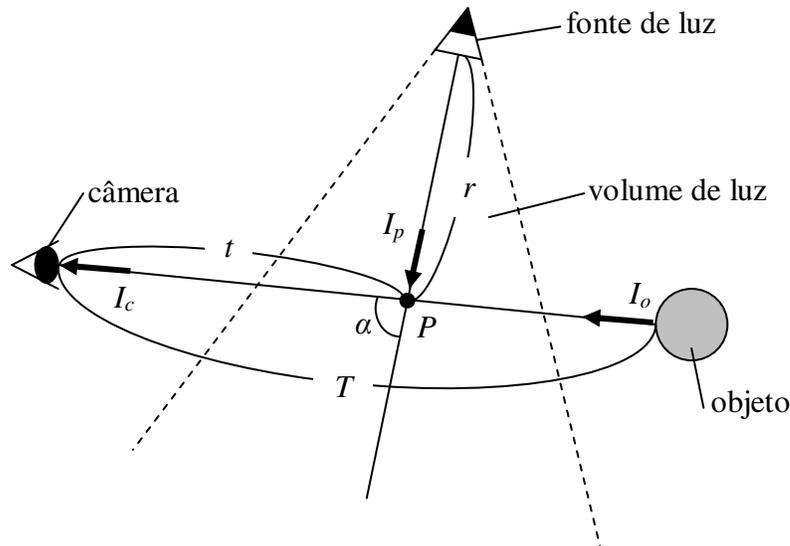


Figura 3-3. Modelo de shading proposto.

Considerando-se a densidade das partículas como sendo uniforme, a taxa de absorção $\beta(t)$ pode ser expressa pela equação de Beer-Lambert

$$\beta(t) = e^{-\sigma t}. \quad (2)$$

A função de fase $F(\alpha)$ diz respeito à intensidade de luz que é espalhada na direção da câmera quando esta faz um ângulo α com a direção do raio de luz. Esta função é influenciada não só pelo ângulo α , mas também pelo tamanho das partículas e pelo comprimento de onda da luz. Devido à complexidade da teoria de Mie, a função de fase utilizada neste trabalho será fruto de uma aproximação experimental proposta por Gibbons [Gibbons58]. A seguinte função de fase será utilizada para modelar o espalhamento da luz,

$$F(\alpha) = K(1 + 9 \cos^{16}(\frac{\alpha}{2})), \quad (3)$$

onde K é uma constante.

A exibição dos volumes de sombra é imprescindível para se obter um bom resultado. Estes volumes são gerados devido à oclusão causada pelos objetos dentro do volume de luz. Assim como as sombras que são projetadas, os volumes

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

de sombra também dão “dicas” das posições dos objetos uns em relação aos outros, aumentando a percepção de distância entre estes. Os volumes de sombra diferenciam-se das sombras projetadas por poderem ser observados no ar, enquanto as sombras projetadas só podem ser vistas sobre a superfície dos objetos. A função de visibilidade $H(r)$ é a responsável pela geração dos volumes de sombra, e ela será implementada fazendo uso de texturas projetivas. A Figura 3-4 mostra o volume de sombra gerado por uma esfera.

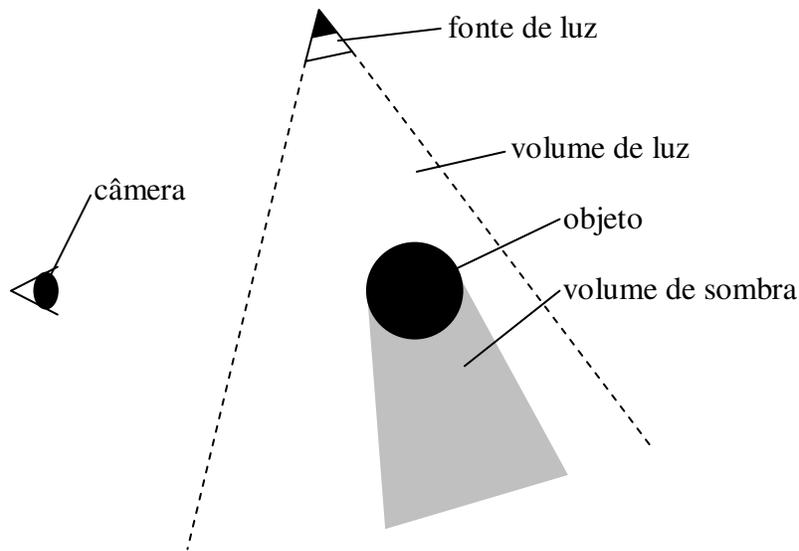


Figura 3-4. Volume de sombra gerado.

A intensidade de luz que alcança o ponto P é dada por $I_p(r)$. Esta intensidade final é influenciada pela intensidade emitida da fonte de luz, pela distância r e pela taxa de absorção $\beta(r)$. Assim, é obtido que

$$I_p(r) = \frac{I(\theta, \phi)}{r^2} \beta(r), \quad (4)$$

onde $I(\theta, \phi)$ é a intensidade emitida pela fonte de luz na direção de P e (θ, ϕ) indica esta direção. A função de iluminação $I(\theta, \phi)$ será implementada similarmente à função de visibilidade $H(r)$, ou seja, fazendo uso de uma textura que será projetada. A utilização de texturas é bastante conveniente para representar esta função, pois além de poderem ser facilmente criadas e alteradas por artistas, as GPUs possuem recursos para armazená-las e acessá-las rapidamente.

3.3 Técnica de Renderização

Dado que o entendimento de como e por que o efeito de iluminação volumétrica ocorre na natureza foi adquirido, agora resta apenas tornar este conhecimento passível de implementação computacional. Em (1), o primeiro termo representa a luz que está sendo irradiada pelo objeto atenuada pela absorção das partículas suspensas, enquanto o segundo é quem propriamente diz respeito ao espalhamento atmosférico, ou seja, apenas o segundo termo é responsável pelo efeito de iluminação volumétrica. É fácil perceber que a equação (1) sem o segundo termo nada mais é que o modelo de *fog* implementado nas placas gráficas [Woo97]. Como o primeiro termo pode ser facilmente computado pelas placas gráficas, nada seria mais conveniente do que utilizá-las, restando apenas o cálculo do segundo termo. Por não existir uma solução analítica para a integral presente no segundo termo de (1), esta terá que ser estimada numericamente, e preferencialmente utilizando-se recursos da placa gráfica. A próxima subseção mostra como a integral será estimada.

3.3.1 Discretização

O segundo termo da equação (1) será computado fazendo uso de planos de amostragem que serão posicionados na frente da câmera. Cada plano sempre estará localizado em uma posição fixa relativa à câmera, isto é, eles sempre acompanharão a câmera para onde quer que ela vá. Os planos são dispostos de forma que fiquem paralelos ao plano de projeção da câmera, e a distância entre eles é uniforme. A Figura 3-5 mostra como os planos de amostragem devem ser dispostos em relação à câmera e como eles são subdivididos. A idéia é utilizar cada plano para representar uma amostragem da luz que é espalhada na direção da câmera e, ao final, acumular a contribuição de cada um no *color buffer* utilizando um *blending* aditivo.

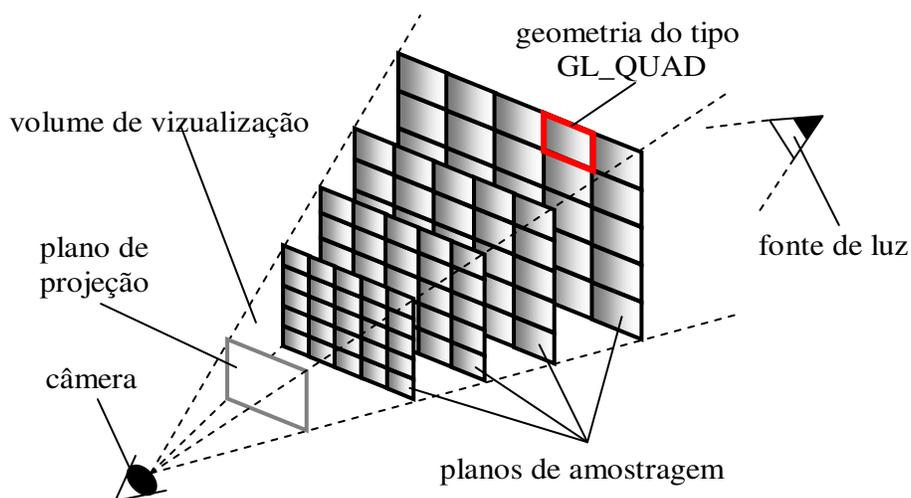


Figura 3-5. Como os planos de amostragem devem ser dispostos e subdivididos.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

Os planos de amostragem são particionados em células retangulares (ver Figura 3-5). É muito importante a divisão dos planos desta forma, pois adiante cada vértice de cada célula deverá ser tratado independentemente. Uma boa maneira de se criar os planos de amostragem é utilizando primitivas do tipo GL_QUAD_STRIP, desta forma, filas de QUADS (horizontalmente ou verticalmente) podem ser criadas sem que haja a necessidade de especificar vértices repetidamente.

Considerando-se I_s o termo relativo ao espalhamento atmosférico na equação (1), obtém-se a seguinte equação discretizando a integral:

$$I_s = \sum_{k=1}^n F(\alpha_k) H(r_k) I_p(r_k) \beta(t_k) \Delta t, \quad (5)$$

onde n é a quantidade de planos de amostragem e k é o k -ésimo plano de amostragem sendo avaliado. A equação (5) após algumas manipulações pode ser reescrita como

$$I_s = \sum_{k=1}^n I(\theta_k, \phi_k) H(r_k) \xi(t_k), \quad (6)$$

onde

$$\xi(t_k) = \frac{F(\alpha_k) \beta(r_k) \beta(t_k)}{r_k^2} \Delta t. \quad (7)$$

Simplificando-se (7) obtém-se

$$\xi(t_k) = \frac{F(\alpha_k) e^{-\rho(r_k+t_k)}}{r_k^2} \Delta t. \quad (8)$$

Como a distância entre planos sucessivos é uniforme, Δt é constante. Vale ressaltar que devido à pequena precisão do *color buffer* (geralmente 8 bits por componente), podem ocorrer problemas numéricos quando Δt for muito pequeno. Portanto, é bom evitar o posicionamento dos planos muito próximos uns dos outros.

A equação (6) possui três termos independentes que serão avaliados separadamente. Os dois primeiros, que representam respectivamente a intensidade emitida pela fonte de luz e a função de visibilidade, são avaliados aplicando-se texturas, uma para cada termo, nos planos de amostragem. Já o último termo, representado por (8), será avaliado em cada vértice dos planos de amostragem. Primeiramente será explicado como este último termo é avaliado, e nas subseções seguintes, os dois primeiros.

A equação (8) é avaliada na CPU para cada vértice de cada plano de amostragem e, ao final, o resultado é passado para GPU na forma de uma cor

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

através da função `glColor`. A vantagem de se utilizar este meio é que os *pixels* entre os vértices serão avaliados automaticamente pela GPU através uma interpolação, isto é possível habilitando-se o modelo de *shading* de Gouraud invocando `glShadeModel(GL_SMOOTH)`. Desta forma, a quantidade de avaliações de $\xi(t_k)$ é reduzida, aumentando-se o desempenho da simulação. O uso deste método também é justificado por $\xi(t_k)$ ser suficientemente suave. Assim, ela pode ser amostrada mais espaçadamente e os valores intermediários interpolados sem que haja o aparecimento de *aliasing*.

3.3.2 Função de Iluminação

Como dito anteriormente, a função de iluminação, $I(\theta_k, \phi_k)$, será representada por uma textura, isto é, um mapa de iluminação. Esta característica foi a principal razão para a escolha deste método. Ao utilizar um mapa de iluminação, a especificação da função de iluminação torna-se bastante flexível, pois apenas alterando-se uma imagem pode-se obter inúmeras e diferentes funções. Pensando mais do ponto de vista de produção, a criação da função de iluminação deixa de ser um processo de escolha de parâmetros adequados, passando a ser apenas um processo de pintura em ferramentas gráficas, que é uma tarefa rotineira dos artistas.

As possibilidades de tipos de mapas de iluminação são infinitas, podendo ser imagens coloridas ou em tons de cinza e com formas variadas. A Figura 3-6 mostra alguns exemplos de mapas de iluminação. Mesmo com tanta flexibilidade, é bom tomar alguns cuidados no momento da escolha do mapa de iluminação, pois podem surgir problemas de *aliasing* em certas circunstâncias. Os mapas que possuem altas frequências são os candidatos mais fortes a gerarem *aliasing*, pois eles necessitam que seja realizada uma amostragem mais minuciosa a fim de se minimizar este problema. A taxa de amostragem pode ser facilmente elevada aumentando-se a quantidade de planos de amostragem, entretanto, essa solução acarretaria em um custo computacional mais elevado, além de gerar os problemas de quantização citados anteriormente. Devido a esses problemas, o ideal é não fazer uso de mapas de iluminação que possuam altas frequências. Por exemplo, na Figura 3-6 os mapas (a), (b) e (d) são muito bons; os (c) e (e) são razoáveis; e o mapa (f) é ruim.

Com um mapa de iluminação apropriado escolhido, agora ele deve ser aplicado a cada um dos planos de amostragem. Este mapa é aplicado utilizando-se técnicas de mapeamento de texturas projetivas [Segal92]. Para implementar tais técnicas em OpenGL, funcionalidades de geração de coordenadas de texturas devem ser utilizadas. Basicamente, o que deve ser feito é habilitar a geração de coordenadas de texturas chamando `glEnable(GL_TEXTURE_GEN_x)` e inicializar os parâmetros desejados invocando `glTexGen(GL_x,param)`, onde x pode ser S, T, R ou Q. Vale lembrar também que devido ao fato da função de iluminação estar sendo multiplicada em (6), deve-se utilizar a função de textura do tipo modulativa,

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

que pode ser escolhida invocando-se `glTexEnv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE)`. Everitt [Everitt01] explica em detalhes como se deve proceder para criar texturas projetivas em OpenGL.

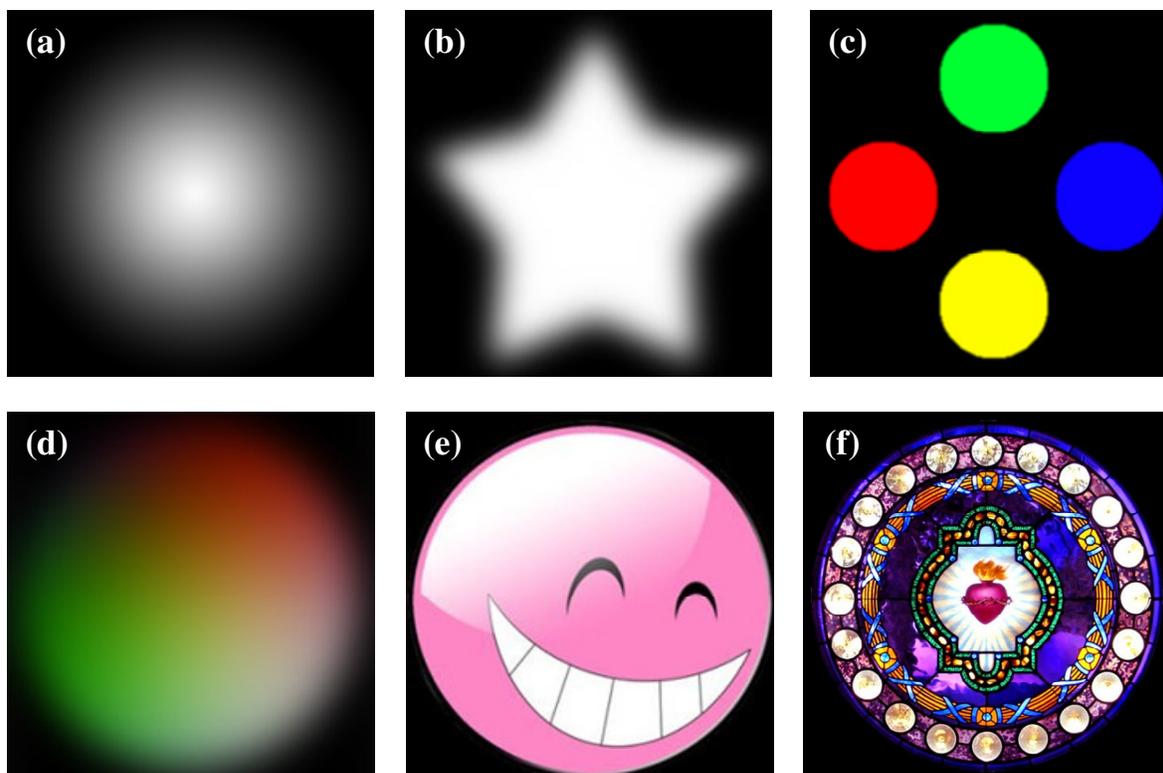


Figura 3-6. Exemplos de mapas de iluminação.

3.3.3 Função de Visibilidade

Os volumes de sombra que surgem quando existe algum objeto dentro do volume de luz são imprescindíveis para o realismo da simulação. Assim como as sombras que são projetadas sobre os objetos, o volume de sombra dá “dicas” importantes para o observador. Estes volumes servem para indicar mais claramente onde os objetos estão posicionados dentro do volume de luz, sem eles fica difícil identificar se os objetos estão, por exemplo, próximos ou distantes uns dos outros.

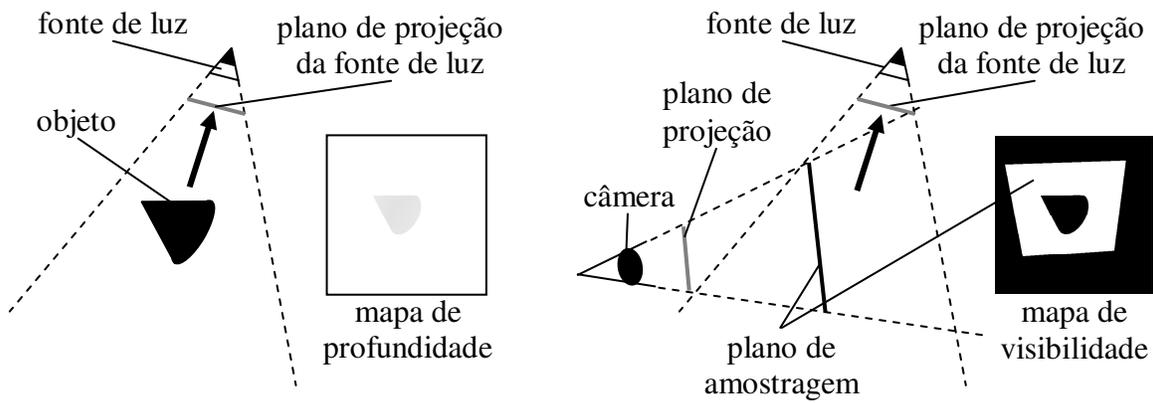
A função de visibilidade é responsável pelo surgimento dos volumes de sombra. Esta função basicamente retorna zero caso um ponto não seja visível do ponto de luz, ou um caso seja visível. Os pontos que não são visíveis da fonte de luz também podem ser entendidos como os pontos que não são iluminados e os pontos visíveis como os pontos que são iluminados. Da mesma forma que a função de iluminação, esta função será implementada utilizando-se texturas, sendo que para a função de visibilidade a textura será uma imagem em preto e branco. As partes em preto representam os pontos que não são visíveis, enquanto as em branco representam os pontos visíveis. Diferentemente da função de iluminação, a

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

função da visibilidade necessita que seja criado um mapa de visibilidade (textura) diferente para cada plano de amostragem, pois os pontos visíveis da fonte de luz não são os mesmos de um plano para outro. Ao final, a idéia é similar a empregada nos mapas de iluminação, em cada plano de amostragem será aplicado projetivamente um mapa de visibilidade fazendo uso de uma função de textura do tipo modulativa.

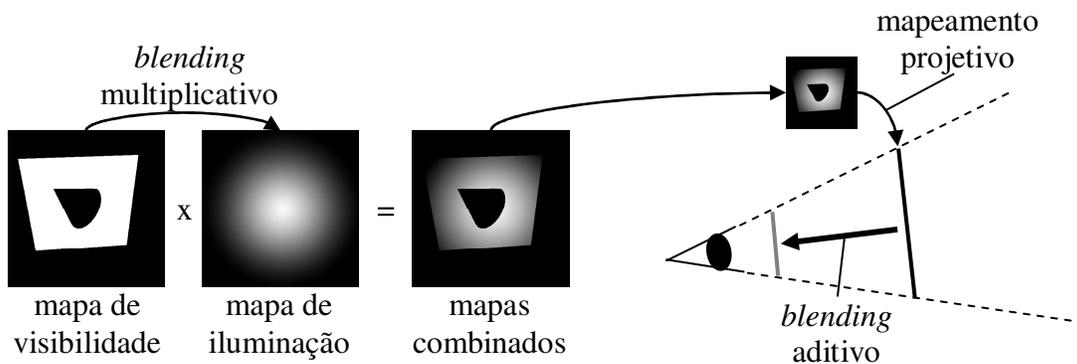
O processo de geração dos mapas de visibilidade é a parte mais complicada e custosa do método. A Figura 3-7 ilustra os passos necessários para a geração e aplicação destes mapas. Inicialmente em (a), um mapa de profundidade é gerado para os objetos da cena considerando-se a fonte de luz como ponto de vista. Este mapa é gerado renderizando-se apenas os objetos que devem gerar sombra e em seguida copiando-se os valores do *buffer* de profundidade para uma textura. Este mapa é único para todos os planos de amostragem e só necessita ser gerado uma vez no início de cada quadro. Em (b), o mapa de profundidade é aplicado sobre o plano de amostragem e a renderização desta vez é feita da câmera. A função de textura habilitada para este mapa é similar à utilizada na técnica de geração de sombras *shadow mapping*, onde os valores armazenados no mapa são testados com as profundidades de cada fragmento gerado da geometria (plano de amostragem), caso o teste seja satisfeito a cor branca é escrita, caso contrário a cor preta é escrita. O resultado dessa renderização já é propriamente o mapa de visibilidade que deve ser copiado para uma textura para ser utilizado posteriormente. Como essa renderização tem que ser feita para tantos quantos forem os planos de amostragem, a extensão `GL_EXT_framebuffer_object` pode ser utilizada para agilizar o processo de escrita nas texturas. Em (c), é ilustrado como o mapa de visibilidade é combinado com o mapa de iluminação, entretanto, o que de fato é feito é ligar cada textura a uma unidade de textura de OpenGL e escolher a função de textura `GL_MODULATE` para ambas. Por fim, em (d), o resultado da combinação dos mapas é aplicado sobre cada um dos planos de amostragem fazendo uso de técnicas de texturas projetivas. Vale lembrar que os planos de amostragem devem ser renderizados habilitando-se *blending* aditivo de modo que o resultado de cada plano vá sendo acumulado no *color bufer*. O *blending* aditivo pode ser facilmente habilitado chamando-se a função `glEnable(GL_BLEND)` seguida da `glBlendFunc(GL_ONE, GL_ONE)`.

Até então o efeito de iluminação volumétrica já foi reproduzido, entretanto, ainda falta um pequeno detalhe que não está relacionado com efeitos atmosféricos. Mesmo não se relacionando com efeitos atmosféricos, a geração das sombras que os objetos projetam uns sobre os outros melhora consideravelmente os resultados. Como o método proposto por este trabalho já faz uso de um mapa de profundidade para geração dos mapas de visibilidade, a melhor forma de se gerar as sombras é utilizando a técnica *shadow mapping*.



(a) Criando mapa de profundidade

(b) Criando mapa de visibilidade



(c) Combinando mapas

(d) Mapeando mapa combinado

Figura 3-7. Como gerar e aplicar os mapas de visibilidade.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

4. Resultados e Problemas Encontrados

Até pouco tempo atrás, principalmente no âmbito de jogos eletrônicos, quase todo tipo de simulação que era implementada tinha que ser simplificada ao ponto em que os resultados claramente transpareciam falsidade. Com o advento das placas gráficas, as simulações até então falsas passaram a exibir um realismo nunca antes alcançado. O método proposto por este trabalho notavelmente conseguiu reproduzir (simular) o efeito de iluminação volumétrica, e é de se admirar o ganho visual obtido ao se utilizar um modelo fisicamente mais correto.

Analisando visualmente os resultados da aplicação, pode-se notar que eles são satisfatórios principalmente para utilização em jogos eletrônicos. É clara a influência da função de fase no resultado final. Em pontos que a câmera está olhando contra a fonte de luz a intensidade luminosa tende a ser muito grande, enquanto quando câmera está olhando de lado a intensidade é pequena. Outro ponto a se destacar é a performance alcançada. Como durante todo o desenvolvimento procurou-se utilizar o máximo de recursos disponíveis na placa gráfica, conseguiu-se atingir uma velocidade suficiente para aplicação em tempo real. No ANEXO A - são exibidos os resultados obtidos.

Mesmo com a adoção de um modelo fisicamente mais correto, ainda há o aparecimento de artefatos indesejados devido a algumas limitações do *hardware*. Os problemas de visualização observados são, na sua maioria, oriundos da utilização da técnica de renderização volumétrica. Um destes problemas é o aparecimento de faixas onde os planos de amostragem intersectam a geometria da cena, como pode ser visto na Figura 4-1 (a). Estas faixas surgem pelo fato dos planos de amostragem, durante a fase de operações por fragmento, serem testados contra o *buffer* de profundidade. Outro problema encontrado ocorre ao se utilizar mapas de iluminação que possuem altas frequências, o que acarreta o aparecimento de *aliasing*. O problema de *aliasing* é mostrado na Figura 4-1 (b). Teoricamente, estes dois últimos problemas poderiam ser sanados apenas aumentando-se a quantidade de planos de amostragem, isto é, aumentando-se a taxa de amostragem, entretanto, outro tipo de artefato indesejado aparecerá caso isto seja feito. Com o número muito elevado de planos de amostragem, cada valor de cor atribuído a cada vértice destes planos é oriundo de uma multiplicação por um Δt muito pequeno, levando a problemas de quantização, como mostrado na Figura 4-1 (c). Por fim, outro problema observado ocorre quando a direção da luz está próxima de atingir ou atinge a perpendicularidade em relação aos planos de amostragem. Quando esta perpendicularidade ocorre, a geração dos mapas de visibilidade fica extremamente prejudicada, levando ao aparecimento de artefatos pretos que se parecem com riscos. Na Figura 4-1 (d), esses riscos são claramente percebidos. O que de fato ocorre é que os planos de amostragem tendem a não serem vistos da fonte de luz por serem finos, gerando um mapa de visibilidade quase que totalmente preto, isto é, com nenhum ponto visível.

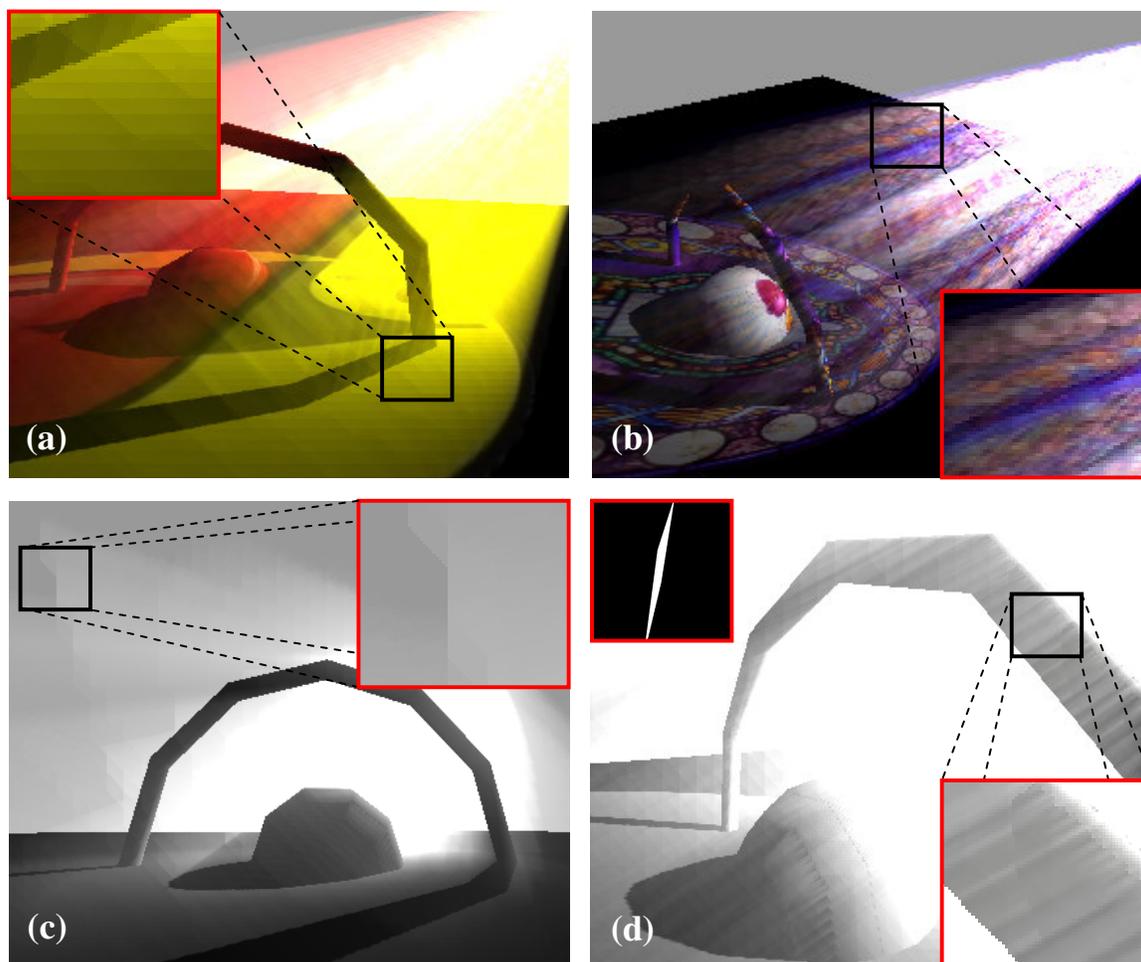


Figura 4-1. Problemas encontrados na simulação. Em (a), o aparecimento de faixas; em (b), *aliasing*; em (c), quantização; e em (d), riscos devido à perpendicularidade entre plano de amostragem e a direção da fonte da luz. Observe, em (d), no detalhe superior, o mapa de visibilidade gerado.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

5. Trabalhos Futuros

Este capítulo visa propor possíveis melhorias que possam ser realizadas futuramente para que a solução proposta seja aperfeiçoada. Estas melhorias dizem respeito principalmente a questões relacionadas com performance e qualidade visual final.

1. Tratar o caso em que densidade das partículas é variável

O método proposto por este trabalho não trata o caso em que a densidade das partículas é variável. Casos em que a densidade não é constante são comumente observados no mundo real. Um bom exemplo deste caso é quando gelo seco é liberado na atmosfera, inicialmente as partículas encontram-se concentradas em certas regiões, e só depois de alguns minutos elas são dissipadas. Como densidade variável ocorre comumente no mundo real, seria bastante interessante considerá-la na simulação.

2. Utilizar *shaders* programáveis para aumentar a performance

O advento dos *shaders* programáveis é um dos mais importantes acontecimentos em computação gráfica nas últimas décadas. O método proposto por este trabalho não faz uso de *shaders*, entretanto, possui um enorme potencial para sua aplicação. A utilização de *shaders* seria bastante conveniente no momento em que (8) é avaliada, pois como a avaliação é feita por vértice ela poderia ser facilmente movida para um *vertex shader*. Como as atuais GPUs possuem até 128 processadores de vértice em paralelo, a utilização de um *vertex shader* traria um grande aumento no desempenho.

3. Resolver os problemas visuais encontrados

Como mostrado no capítulo 4, alguns artefatos indesejados são gerados durante a simulação. Estes artefatos são, na sua maioria, herdados das técnicas de renderização volumétrica. Enquanto alguns artefatos podem ser evitados escolhendo-se os parâmetros apropriadamente, outros necessitariam de mais recursos de *hardware* para serem sanados. Um exemplo deste último tipo de artefato são os erros de quantização gerados devido à baixa precisão do *color buffer*. Este tipo de artefato poderia ser facilmente resolvido aumentando-se precisão do *color buffer*, porém, enquanto os fabricantes de placas gráficas não fazem isto, uma boa idéia seria utilizar o *buffer* de acumulação. O *buffer* de acumulação usualmente possui 8 *bits* a mais por componente que o *color buffer*, dando um total de 16 *bits* por componente, isto é, o dobro de precisão. Outro tipo de artefato gerado é o *aliasing*, que é devido ao fato da amostragem não estar sendo realizada adequadamente. Keller [Keller01] propôs um método de amostragem mais rápido e que reduz o aparecimento de *aliasing*.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

4. Tratar o caso em que a fonte de luz é onidirecional

Outro problema não abordado neste trabalho é o caso em que a fonte de luz é onidirecional, isto é, ela emite luz em todas as direções. A origem deste problema está no fato de não se poder definir o campo de visão superior a 180 graus para uma câmera em OpenGL. Para que as luzes onidirecionais pudessem ser naturalmente tratadas pelo método proposto por este trabalho, seria necessário se criar uma câmera com 360 graus de campo de visão. Esta limitação é similar a que surge quando se deseja utilizar a técnica *shadow mapping* para gerar sombras de luzes onidirecionais. Atualmente, algumas abordagens que tratam *shadow mapping* com luzes onidirecionais já foram propostas, como a de [Gerasimov04]. É possível que aplicação destas abordagens trate do caso das luzes onidirecionais para a simulação de iluminação volumétrica.

5. Combinar o efeito de iluminação volumétrica com outros efeitos.

Como este trabalho foca em um efeito de iluminação, um idéia que pode render bons frutos é a combinação deste efeito com tantos outros que já existem na literatura. Um efeito bastante interessante e que poderia melhorar consideravelmente os resultados é o efeito de *bloom*. Este efeito ocorre em cenas onde existe uma intensidade elevada de energia luminosa, levando o observador a ficar encandeado. Como este efeito tende a borrar os pontos onde existe muita energia luminosa, ele poderia ajudar a mascarar alguns dos problemas citados anteriormente. Outra idéia bastante interessante seria fazer uso de HDRR (*High Dynamic Range Rendering*) para realizar a renderização em uma escala dinâmica maior.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

Referências

[**Pixar**]. Disponível em: <http://www.pixar.com/index.html>. Último acesso: 26/07/2007.

[**OffsetSoftware**]. Disponível em: <http://www.projectoffset.com/>. Último acesso: 27/07/2007.

[**OpenGL**]. Disponível em: <http://www.opengl.org/>. Último acesso: 24/07/2007.

[**AlanWake**]. Disponível em: <http://www.alanwake.com/>. Último acesso: 26/07/2007.

[**Cowley**]. L. Cowley. "Atmospheric Optics". Disponível em: <http://www.atoptics.co.uk/>. Último acesso: 31/07/2007.

[**Torque**]. Disponível em: <http://www.garagegames.com/>. Último acesso: 16/08/2007.

[**Nishita87**]. T. Nishita, Y. Miyawaki e E. Nakamae. "A shading model for atmospheric scattering considering luminous intensity distribution of light sources". In SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 303-310, New York, NY, USA, 1987. ACM Press.

[**Jensen98**]. H. Jensen e P. Christensen. "Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps". In Proceedings of SIGGRAPH'98, pages 311-320, Orlando, July 1998.

[**Dobashi00**]. Y. Dobashi, T. Yamamoto e T. Nishita. "Interactive rendering method for displaying shafts of light". In PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications, page 31, Washington, DC, USA, 2000. IEEE Computer Society.

[**Dobashi02**]. Y. Dobashi, T. Yamamoto e T. Nishita, "Interactive Rendering of Atmospheric Scattering Effects Using Graphics Hardware," Graphics Hardware, 2002.

[**James03**]. R. James. "Graphics programming methods", cap. 3.12 "True volumetric shadows". Charles River Media, Rockland, MA, USA. 2003.

Trabalho de Graduação	Versão: 1.1
Simulação de Iluminação Volumétrica	Data da versão: 22/08/2007

[Zhu04]. Y. Zhu, G. Owen, F. Liu e A. Aquilio. "Gpu-based volumetric lighting simulation". In CGIM '04: Proceedings of The 7th IASTED International Conference on COMPUTER GRAPHICS AND IMAGING, page 479, Kauai, Hawaii, USA, 2004. IASTED/ACTA Press.

[Mitchell04]. J. Mitchell. "ShaderX3: Advanced Rendering with DirectX and OpenGL", cap. 8.1 "Light Shaft Rendering". Charles River Media. 2004.

[Biri02]. V. Biri, S. Michelin e D. Arquès. "Real-Time Animation of Realistic Fog". Thirteenth Eurographics Workshop on Rendering, 2002.

[Gibbons58]. M. Gibbons. "Radiation Received by Uncollimated Receiver from a 4-PI Source". Journal of the Optical Society of America, vol. 48, issue 8, pages 550-555, 1958.

[Woo97]. M. Woo, J. Neider, T. Davis e OpenGL ARB. "Opengl Programming Guide: The Official Guide to Learning Opengl, Version 1.1", cap. 6 "Blending, Antialiasing, Fog, and Polygon Offset". Addison-Wesley Pub. 1997. Disponível em: <http://www.glprogramming.com/red/chapter06.html#name3>. Último acesso: 16/08/2007.

[Segal92] M. Segal, C. Korobkin, R. V. Widenfelt, J. Foran e P. E. Haeberli. "Fast Shadows and Lighting Effects Using Texture Mapping". Computer Graphics, Vol. 26, No. 2, pages 249-252, 1992.

[Everitt01]. C. Everitt. "Projective Texture Mapping". 2001. Disponível em: http://developer.nvidia.com/object/Projective_Texture_Mapping.html. Último acesso: 16/08/2007.

[Keller01]. A. Keller e W. Heidrich, "Interleaved Sampling". Eurographics Workshop on Rendering Techniques, 2001.

[Gerasimov04]. P. Gerasimov. "GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics", cap. 12 "Omnidirectional Shadow Mapping". Addison-Wesley Professional. 2004.

ANEXO A - Imagens da Aplicação

Os Resultados mostrados neste capítulo foram obtidos em um AMD Athlon 64 3200+ com 1024MB RAM e uma placa gráfica NVidia GeForce 6600 GT de 128MB.

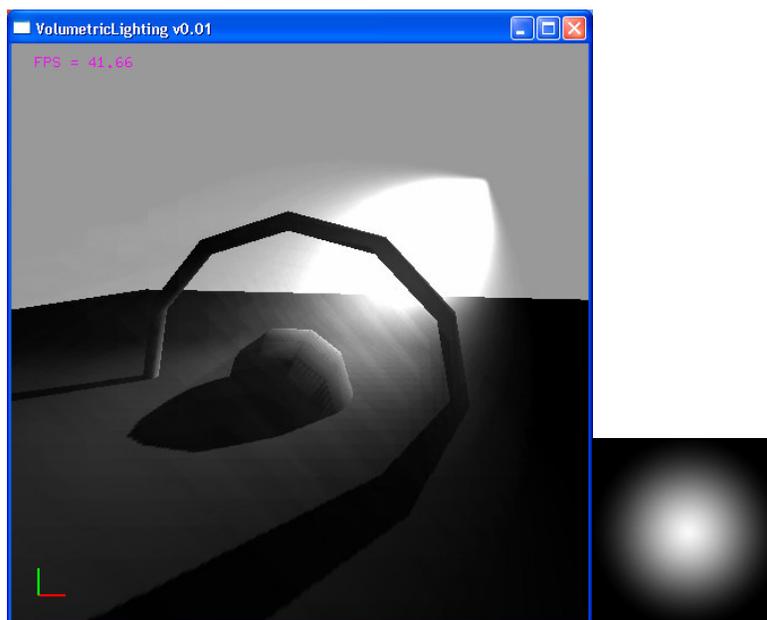


Figura A-1. Iluminação volumétrica com $n = 40$, QUADs = 20x20, $\sigma = 0.1$, $K = 20$, $\Delta_i = 0.1$ e executando a 41 *fps*. Mapa de iluminação utilizado à direita.

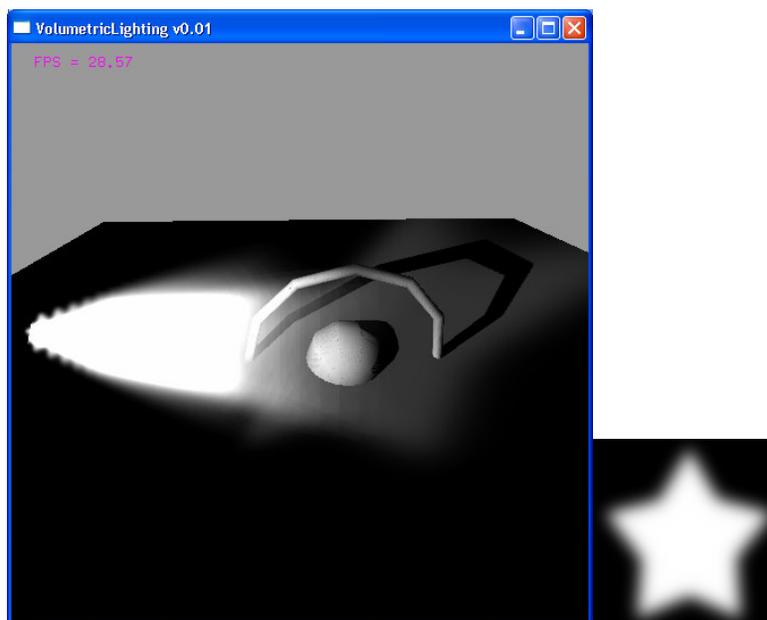


Figura A-2. Iluminação volumétrica com $n = 50$, QUADs = 30x30, $\sigma = 0.9$, $K = 20$, $\Delta_i = 0.04$ e executando a 28 *fps*. Mapa de iluminação utilizado à direita.

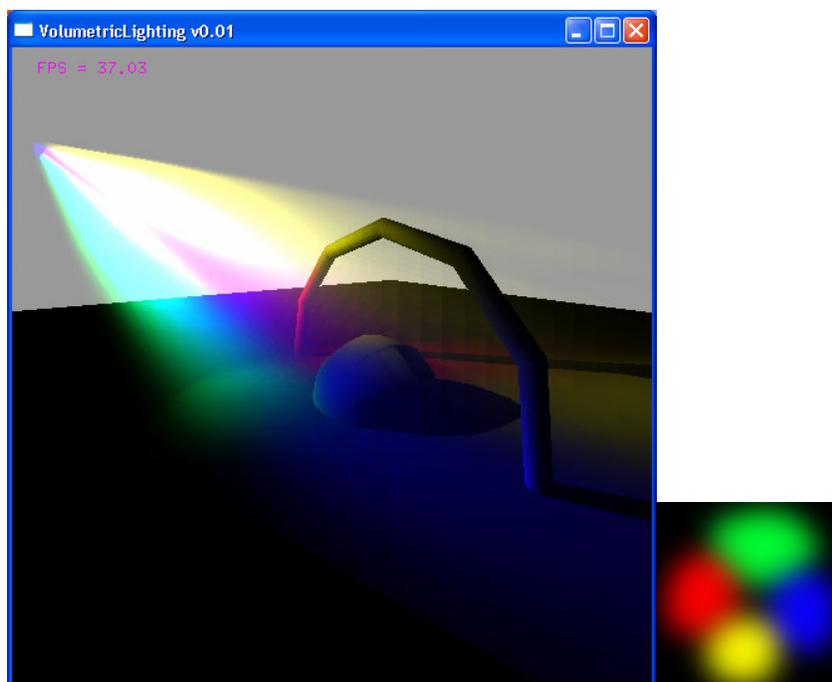


Figura A-3. Iluminação volumétrica com $n = 40$, QUADs = 30x30, $\sigma = 0.5$, $K = 30$, $\Delta_t = 0.07$ e executando a 37 *fps*. Mapa de iluminação utilizado à direita.

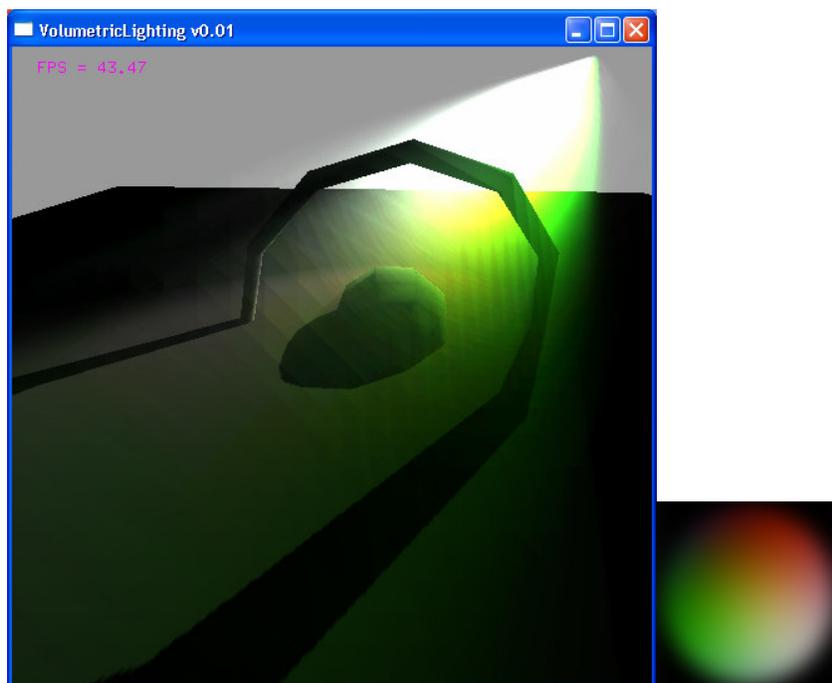


Figura A-4. Iluminação volumétrica com $n = 45$, QUADs = 20x20, $\sigma = 0.3$, $K = 25$, $\Delta_t = 0.06$ e executando a 43 *fps*. Mapa de iluminação utilizado à direita.

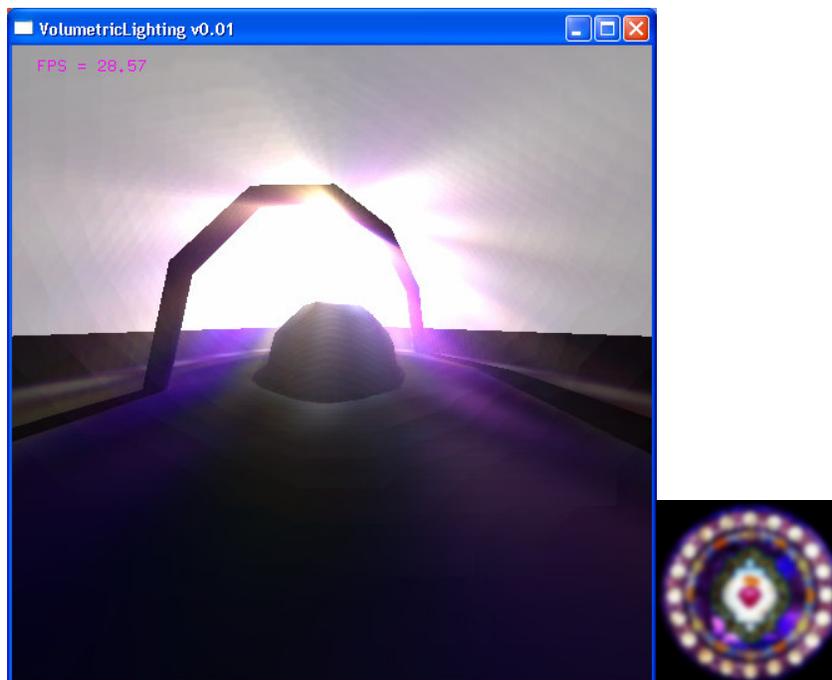


Figura A-5. Iluminação volumétrica com $n = 55$, QUADs = 10x10, $\sigma = 0.2$, $K = 25$, $\Delta_t = 0.05$ e executando a 28 *fps*. Mapa de iluminação utilizado à direita.

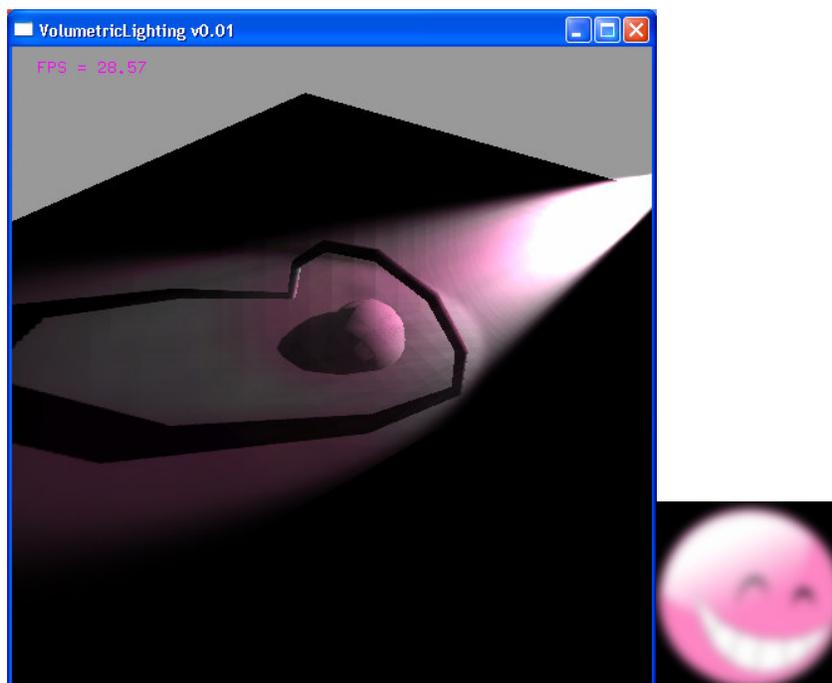


Figura A-6. Iluminação volumétrica com $n = 50$, QUADs = 30x30, $\sigma = 0.2$, $K = 20$, $\Delta_t = 0.07$ e executando a 28 *fps*. Mapa de iluminação utilizado à direita.