

5

Iluminação Baseada em Imagem

Aplicações de computação gráfica geralmente necessitam gerar imagens que representam visualmente uma cena. Para simular corretamente a forma como uma pessoa observaria a cena caso fosse real, é necessário que se utilizem luzes e materiais, de forma a modelar a maneira como estes interagem para gerar a cor final de um objeto. Diferentes modelos de iluminação são utilizados para representar materiais com diferentes propriedades.

Um problema que ocorre especialmente em aplicações em tempo real é a representação fiel das condições de iluminação do ambiente incidindo em um objeto. Como essas condições são geralmente bastante complexas, apresentam dificuldades na modelagem com apenas luzes tradicionais utilizadas em computação gráfica. Mapas de iluminação são um conjunto de técnicas utilizadas para codificar as condições de iluminação do ambiente que cerca um objeto[36]. A representação de mapas de iluminação com texturas LDR apresenta uma falha decorrente da não-linearidade entre os valores dos pixels e as condições reais de exposição da cena e sua radiância medida. Nas próximas seções, os modelos de iluminação padrão e técnicas de Iluminação Baseada em Imagem (Image-Based Lighting) são apresentadas, tanto para o caso não em tempo real como para para o caso em tempo real.

5.1

Modelos de iluminação e sombreamento

5.1.1

Iluminação

Iluminação é termo normalmente designado para a interação entre os objetos, sua geometria e materiais, com as fontes de luz. Sem um algoritmo de iluminação, no pior caso, cenas tridimensionais podem se tornar indistinguíveis de cenas bidimensionais. A função principal de um

modelo de iluminação é representar como os materiais componentes de objetos sintéticos interagem com diferentes fontes luminosas.

Diferentes modelos de iluminação são utilizados para diferentes requisitos de desempenho da aplicação e da qualidade final das renderizações. Modelos mais complexos e computacionalmente pesados podem ser utilizados para algoritmos não interativos onde se espera gerar imagens com maior qualidade em detrimento do tempo necessário para gerá-las. Aplicações interativas ou tempo real possuem requisitos de tempo de renderização mais críticos e modelos de iluminação e sombreado mais simples são utilizados.

A iluminação de objetos sintéticos é geralmente baseada em uma aproximação das funções bidirecionais de distribuição de reflectância (BRDF) associadas a cada material componente dos objetos do mundo real. A derivação dessas funções é muitas vezes complexa e a utilização de aproximações se torna necessária. Tipicamente, ao calcular a solução de iluminação para um objeto sintético, são levadas em consideração três componentes básicos responsáveis por comportamentos particulares para cada superfície. Os componentes são chamados *difuso*, *especular* e *ambiente*.

5.1.1.1

Componente difuso

O componente difuso, é o mais fisicamente correto dos componentes da solução de iluminação de um objeto sintético. Baseado na lei de *Lambert*, que define que em uma superfície perfeitamente difusa, a luz refletida é determinada pelo cosseno na normal da superfície N com o vetor de luz L .

$$I_{diff} = N \cdot L = \cos\phi \quad (5-1)$$

Note que para essa interpretação geométrica, tanto N quanto L devem ser vetores unitários. A Figura 5.1 mostra uma interpretação geométrica da relação entre a intensidade de luminosidade e a distância entre os feixes de luz que atingem a superfície. De forma que, a intensidade da luz diminui quando a distância entre feixes se torna maior. Para o componente difuso do material, são levadas em consideração geralmente a cor da fonte de luz, a cor do material e o coeficiente difuso do material, de forma a simular a interação da cor da luz com a cor do material. Tal fato é importante para definir o quantidade de luz a ser refletida e absorvida. Para simular os efeitos de cor de luz e cor do material, duas constantes são adicionadas ao cálculo

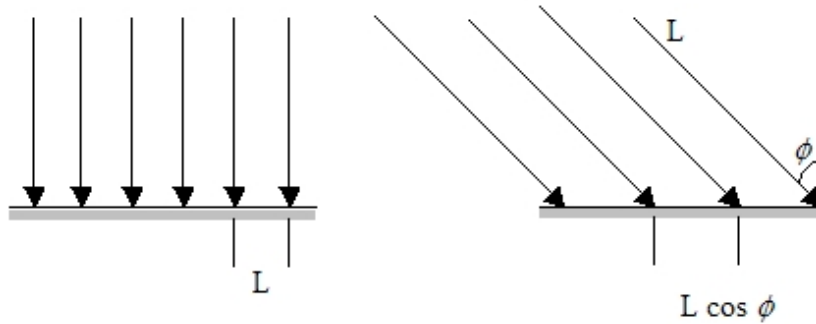


Figura 5.1: Interpretação geométrica do componente difuso

do componente ambiente que se torna a equação 5-2

$$I_{dif} = m_{dif} \otimes l_{dif}(N \cdot L) \quad (5-2)$$

Onde o operador \otimes representa a multiplicação componente à componente de um vetor. Num modelo fisicamente correto, uma luz azul $l_{dif} = (0,0,255)$ iluminando um material vermelho $m_{dif} = (255,0,0)$ seria totalmente absorvida e o resultado visual seria a cor preto $I_{dif} = (0,0,0)$. Ao modular a cor da luz com a cor do material esse efeito é atingido. A contribuição da iluminação difusa é a mais importante para o aspecto visual do objeto.

Sua formulação é independente da posição do observador, sendo definida apenas pela posição da fonte de luz e da geometria do objeto, em particular, de suas normais. Tal comportamento simula o fato que fótons incidentes em um ponto p numa superfície difusa tendem a refletir em todas as direções do hemisfério superior da esfera com centro em p . Dessa forma, não são notadas diferenças quando o observador se movimenta, levando ao aspecto visual constante em qualquer direção.

5.1.1.2

Componente especular

O componente especular é o responsável pela aparência brilhante de algumas superfícies, através da criação de *highlights*. O componente especular ajuda o observador a entender a curvatura da superfície, bem como definir a direção e as localizações das fontes luminosas [25]. O modelo mais utilizado para representar esse componente em hardware é mostrado na equação 5-4.

$$I_{spec} = (R \cdot V)^{shi} = \cos(\rho)^{shi} \quad (5-3)$$

Onde R e V são a reflexão do vetor de direção da luz L em relação à normal N , e o vetor de um ponto p na superfície em direção ao observador, respectivamente. Pela equação 5-4, é possível perceber que o valor do componente especular terá seu máximo quando R e V são iguais. O que significa que ao observar uma superfície, se o vetor de reflexão da fonte de luz aponta diretamente para o observador, o componente especular está maximizado e produz um *highligh* nessa posição. À medida que o ângulo entre o observador e o vetor de reflexão aumenta, o brilho será menor. O componente especular tem como função simular o fato que fótons incidentes em superfícies especulares, apresentam a tendência de refletir na direção do vetor de reflexão R .

É fácil perceber, pela própria construção do componente especular, que este é um termo dependente da posição do observador. Esse efeito é mais perceptível, uma vez que a localização do brilho muda conforme a posição do observador varia. Uma variação muito popular da equação 5-4 criada por Blinn[48] é:

$$I_{spec} = (N \cdot H)^{shi} \quad (5-4)$$

Onde H é o (*half vector*) normalizado entre L e V .

$$h = \frac{L + V}{\|L + V\|} \quad (5-5)$$

Essa aproximação é interessante pois o cálculo do *half vector* é menos computacionalmente pesada que o cálculo do vetor de reflexão. Para adicionar a influência da cor especular do material e da fonte de luz, esses componentes são adicionados à equação 5-4, gerando uma nova equação:

$$I_{spec} = m_{spec} \otimes l_{spec} (R \cdot V)^{m_{shi}} \quad (5-6)$$

O parâmetro m_{shi} descreve o fator de brilho da superfície. Quanto maior o valor de m_{shi} , mais brilhante a superfície será. Assim como o componente difuso, caso $(N \cdot H)$ seja menor que zero (a luz está atrás da superfície) o termo deverá ser saturado em zero.

5.1.1.3

Componente ambiente

No mundo real, e cenas renderizadas com algoritmos de iluminação global, os feixes de luz provenientes das fontes de iluminação podem refletir em alguma geometria da cena e incidir em outro objeto da cena,

contribuindo assim para a solução de iluminação do objeto. Porém, nos modelos simples utilizados em aplicações tempo real, essa contribuição, não é calculada diretamente. Essa contribuição não é adicionada aos componentes difuso ou especular, mas sim aproximada em um componente particular. A contribuição ambiente é uma aproximação da luz indireta de toda a cena. Esse componente é responsável pelo aspecto das condições gerais de iluminação da cena, e impede que as algumas cenas renderizadas se tornem escuras demais. Geralmente, a aproximação dessa iluminação indireta é feita associando uma constante global ambiente com o componente ambiente para cada material da cena, como mostrado na equação 5-7.

$$I_{amb} = m_{amb} \otimes c_{amb} \quad (5-7)$$

Outro fator que se destaca na utilização do componente ambiente, é o aumento do realismo de cenas com sombras. Geralmente são utilizados algoritmos de múltiplas passadas, onde as sombras são calculadas para o cada fonte de luz na cena e o componente ambiente é adicionado mesmo em áreas em sombra. Esse efeito simula a iluminação do mundo real uma vez que a iluminação indireta provém de diferentes regiões e dessa maneira contribui para a iluminação de áreas em sombra.

5.1.1.4

Modelos de iluminação

Para representar um modelo de iluminação completo, é necessário integrar as contribuições dos diferentes componentes individuais do modelo. O modelo de iluminação mais utilizado atualmente é o modelo Phong [20] que combina os três componentes apresentados anteriormente. A equação geral do modelo Phong é descrito pela Equação 5-8.

$$I = I_a k_a + \sum_{l=0}^n (I d_l k_d (N \cdot L) + I s_l k_s (R \cdot V)^{shi}) \quad (5-8)$$

É fácil notar pela equação 5-8, que tanto as contribuições difusa e especular são independentes para cada luz e são somadas para a criação do resultado final. A contribuição ambiente por representar a aproximação da luz refletida por toda cena, por definição, já leva em consideração todas as fontes de luz na cena e não se encontra no somatório. Outros modelos de iluminação são utilizados para representar de forma mais fiel diferentes tipos de materiais, como superfícies metálicas, superfícies anisotrópicas, etc.

Entre esses se destacam, o modelo de Oren-Nayar[55], o modelo Schlick, e o modelo anisotrópico de Ward [27]. Para maior detalhamento desses modelos, [44] é uma boa referência.

5.1.2 Sombreamento

Sombreamento é o processo de realizar cálculos de iluminação e desses derivar as cores de pixel resultantes. Os modelos de iluminação definem como a solução de iluminação deve ser calculada para um objeto, porém não define quais são os componentes geométricos utilizados para os cálculos. Quanto maior a frequência desses objetos, melhor o resultado final do objeto. Os modelos de sombreamento definem a unidade na qual os cálculos de iluminação serão realizados. Tipicamente as unidades utilizadas são as faces, os vértices e os pixels em cada superfície do objeto. Existem três modelos principais de sombreamento (*shading*): *Flat*, *Gouraud* e *Phong*. Esses correspondem a calcular a iluminação por face, por vértice e por pixel da superfície de um objeto, respectivamente.

O primeiro modelo de sombreamento é o modelo Flat. Para cada face do objeto, sua iluminação é calculada e a cor resultante é utilizada para preencher todo o polígono. Dessa forma, a iluminação de cada face dos modelos da cena é constante e baseada na normal da face. Esse modelo apresenta variações de iluminação bastante perceptíveis para cada face. O modelo flat apresenta bom desempenho e possui implementação simples. Apesar de não conseguir representar de forma suave superfícies curvas, esse modelo é interessante quando o usuário deseja visualizar as faces do modelo.

No modelo Gouraud, os cálculos de iluminação são feitos para cada vértice individualmente, e as cores resultantes da iluminação dos vértices são interpoladas linearmente para a os pixels da superfície. O valor de iluminação para cada vértice é calculado utilizando-se uma equação de iluminação (importantes bibliotecas gráficas, como OpenGL [28], se utilizam do modelo Phong para os cálculos de iluminação). A maioria das placas gráficas implementa o modelo Gouraud, pois é relativo rápido, praticamente tão rápido quanto o modelo flat, uma vez que a iluminação é calculada apenas nos vértices da face e apresenta qualidade visual muito superior ao modelo flat. O problema mais perceptível desse modelo é a alta dependência do nível de detalhes dos objetos a serem renderizados. Esses problemas incluem a perda de *highlights* e efeitos de iluminação *spot*.

O modelo Phong[20] se utiliza da interpolação linear das normais de cada vértice, para calcular a normal de sombreamento para cada pixel da face. Essa normal é então utilizada para calcular a iluminação em cada pixel individualmente. O hardware atual implementa o modelo Phong através de *register combiners* [14] e da utilização de pixel shaders. O sombreamento



Figura 5.2: Modelos de sombreamento.

Gouraud pode ser utilizado para atingir essencialmente o mesmo resultado do sombreamento Phong através da subdivisão de superfícies em superfícies menores, aumentando a tesselação da superfície, e dessa forma, diminuindo e até mesmo eliminando as imprecisões causadas pela interpolação linear de valores de cor entre vértices distantes. Porém esse método é muito mais caro que a utilização do modelo Phong em hardware que o suporta, e a renderização é muito lenta. Em aplicações não interativas essa técnica gera resultados bastante realistas. Essa técnica é, por exemplo, a base do sistema RenderMan da Pixar [26]

Os modelos de iluminação e sombreamento resumidamente podem ser definidos como a aplicação de diferentes equações de iluminação. Porém os modelos de iluminação e os tipos de luzes normalmente utilizados falham ao tentar simular o comportamento de ambientes com muitas fontes de luzes provenientes de diferentes direções e diferentes dos modelos de luzes tradicionais, como luzes com área. O tempo para calcular a solução de iluminação para uma cena com muitas luzes aumenta consideravelmente, degradando o desempenho de aplicações. Outro fator importante é a dificuldade de derivar posições de luz e características de iluminação para posterior modelagem dessas fontes e utilização no algoritmo de renderização.

5.2

Mapas de ambiente como iluminação

Essa seção descreve como imagens HDR podem ser utilizadas como fontes de iluminação para objetos gerados por computador e cenas. Imagens HDR registram todo o espectro de luz incidente em um ponto no espaço,

contém informação sobre forma, cor e intensidade das fontes de luz diretas, como a informação de cor e de distribuição da iluminação indireta proveniente das superfícies de uma cena. Com a utilização de imagens HDR, algoritmos de renderização podem simular de forma acurada, como objetos seriam iluminados caso estivessem posicionados na cena real. O processo de utilizar imagens como fontes de iluminação se chama *Image based lighting* (IBL). A Figura 5.3 mostra uma cena renderizada utilizando fontes de luz pontuais tradicionais, enquanto a Figura 5.4 mostra a mesma cena renderizada utilizando um mapa de iluminação.

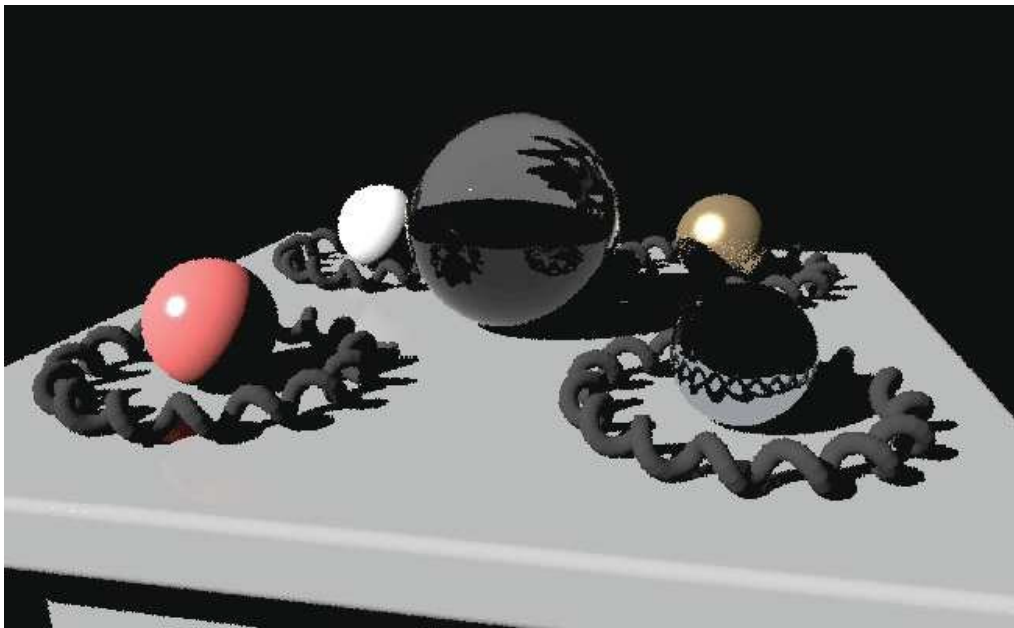


Figura 5.3: Cena iluminada por luzes sintéticas

O utilização de IBL se baseia em três processos básicos[47]. O primeiro, claramente é a utilização de fotografia HDR. O segundo é fotografia omnidirecional, que consiste em capturar imagens que representam todas as direções partindo de um ponto específico no espaço. Como nos ambientes reais, a luz incidente nos objetos provém de todas as direções, a capacidade de captura dessas direções é fundamental. O capítulo 2 descreve em detalhes técnicas para a captura de imagens HDR omnidirecionais, conhecidas também como *light probes* ou mapas de ambiente HDR.

O terceiro processo tipicamente associado a IBL são as técnicas de iluminação global, ou seja, algoritmos que simulam a forma como a luz é transportada das fontes de luz, gerando a aparência final dos objetos sintéticos. A diferença básica da iluminação global para iluminação local está na capacidade de simular o efeito de interreflexão, ou seja, a



Figura 5.4: Cena iluminada por mapa de iluminação

iluminação indireta incidente em um objeto, proveniente de outros objetos na cena, especialmente os mais próximos. Os algoritmos de iluminação global capazes de representar a interreflexão da luz em superfícies difusas são conhecidos como algoritmos de radiosidade [54]. Geralmente, os algoritmos de iluminação global se baseiam em algoritmos de *ray tracing*, que calculam a trajetória dos raios de luz, derivando a cor final de cada pixel através de um modelos de iluminação.

5.2.1

Conceitos básicos de iluminação baseada em imagem

As técnicas básicas associadas a utilização de *image based lighting* são derivadas da técnica básica apresentada na animação *Rendering with Natural Light* (RNL). Essa animação consiste de um conjunto de esferas iluminadas com um iluminação capturada em um imagem HDR. Os processos básicos dessa animação e das técnicas de IBL são:

1. Aquisição do mapa de iluminação (*light probes*)
2. Modelar a geometria da cena (geometria local)
3. Mapear o mapa de iluminação em uma superfície que cerca a cena.
4. Renderizar a cena iluminada pelo mapa de iluminação.

5. Pós processar e aplicar *tone mapping* na renderização.

Esses passos são os mesmos apresentados no Capítulo 3. Nessa seção esses processos serão explicados mais detalhadamente, fornecendo os passos padrões para renderização em tempo real de cenas utilizando mapas de iluminação em alta precisão.

5.2.1.1

Aquisição do mapa de iluminação

A aquisição de mapas de iluminação é tradicionalmente realizada através de técnicas de fotografia omnidirecional. Tipicamente, posiciona-se um esfera espelhada em um tripé a certa distância do chão e ajusta-se o zoom da câmera para que a esfera ocupe todo o quadro. Um conjunto de fotografias com diferentes exposições é então tirado. Cada fotografia do conjunto geralmente possui tempo de exposição variando de um *stop* em relação a imagem anterior. Esferas espelhadas são capazes de refletir praticamente todo o ambiente que a circunda, não apenas o hemisfério entre a esfera e a câmera. Tal fato ocorre devido a fórmula básica de reflexão, o ângulo de incidência é igual ao ângulo de reflexão. Dessa forma, raios partindo da borda da esfera em direção a câmera, possuem praticamente 90 graus. Com isso, o ângulo de incidência original relativo à câmera é de quase 180 graus. Conseqüentemente, o raio é original de um ponto localizado aproximadamente do lado oposto da esfera em relação à câmera. O conjunto de imagens é então processado para gerar uma imagem HDR do ambiente. O algoritmo para geração de uma imagem HDR a partir de um conjunto LDR é descrito no Capítulo 3. Outro algoritmo bastante robusto para recuperação de curvas de resposta podem ser encontrado em [15].

A partir da imagem gerada, este deve ser salva utilizando algum dos formatos de arquivo próprio para imagens HDR, descritos no Capítulo 3. Uma técnica adicional para remover artefatos causados pela reflexão do tripé e do fotógrafo consiste em utilizar duas esferas posicionadas a cerca de 90 graus uma da outra e depois processadas para composição das partes corretas. Outras formas para a aquisição de imagens omnidirecionais podem ser encontradas em [50, 56, 25].



Figura 5.5: *light probes* com diferentes exposições

5.2.1.2

Modelagem da geometria da cena

Nessa fase, a geometria dos modelos componentes da cena deve ser modelada, bem como a descrição dos materiais componentes de cada objeto da cena. A definição dos materiais é comumente feita através de constantes em um arquivo, que definem as propriedades difusa e especular desse material, definindo assim se o objeto terá aparência metálica, brilhante, *matte*, entre outras.

5.2.1.3

Mapeamento do mapa de iluminação na superfície do ambiente

Uma das suposições para IBL é que a cena a ser renderizada está cercada por um ambiente, representado por uma superfície, que emite luz em direção à cena. Essa superfície pode ser representada de diferentes formas, geralmente com geometria simples, por exemplo uma esfera ou um cubo capazes de englobar toda a geometria da cena. Nessa geometria é mapeada o mapa de iluminação omnidirecional. Essa geometria faz parte da cena

distante definida no Capítulo 3 e é definida como uma geometria emissiva, ou seja, emissora de luz.

O mapeamento do mapa de iluminação na superfície emissiva representante do ambiente pode ser feito de diferentes maneiras, de acordo com o formato utilizado para representar o mapa de iluminação, de forma esférica ou através de *cube maps*. Essas duas formas de codificar direções do mundo em coordenadas de textura no mapa de iluminação serão discutidas na próxima seção. Alguns renderizadores como o Radiance necessitam de um arquivo contendo o algoritmo de mapeamento, que é descrito em [7]. É importante lembrar que esse estágio é utilizado para aumentar o realismo da cena, dando a ilusão que os objetos estão realmente circundados pelo ambiente. A não geração dessa superfície, no entanto, não impede a correta renderização dos objetos na cena.

5.2.1.4

Renderização da cena e pós processamento

Com a cena já modelada e o mapa de ambiente mapeado na superfície emissiva, basta submeter esses dados em um formato aceitável pelo renderizador de iluminação global. Na animação RNL[?], o sistema de renderização utilizado foi o Radiance. Para melhores resultados, alguns parâmetros de renderização devem ser definidos, como a quantidade de raios a ser utilizado. Diferentes pacotes de renderização, integrados à softwares de modelagem como 3D Studio Max [16], já apresentam suporte à IBL, como MentalRay [8].

O passo final para a criação da renderização baseada em mapas de ambiente é o pós processamento da renderização. Como pacotes de renderização geram imagens HDR, estas podem ser salvas em algum formato próprio para imagens HDR, ou ser visualizadas diretamente. Para a visualização é necessário recorrer aos algoritmos de *tonemapping* apresentados no Capítulo 4. Outros efeitos podem ser adicionados antes do processo de tonemapping, visando aumentar o realismo da cena, ou simular algum efeito especial em particular. Na animação RNL, o resultado da renderização foi pós-processado para simular o efeito de *Bloom*. As imagens foram processadas com filtros gaussianos de diferentes raios para gerar o resultado final. O capítulo 4 apresenta em detalhes os conceitos de pós-processamento no *pipeline* HDR. Outro efeito sutil aplicado é o efeito de *vignetting*. *Vignetting* é o processo de escurecer de forma delicada os cantos da imagem, que ocorre naturalmente em câmeras fotográficas.

Esse efeito ajuda na sensação de aumento de *dynamic range* especialmente em animações. Isso ocorre pois pixels numa região muito iluminada ao se distanciarem do centro da imagem tem seu valor diminuído gradualmente, enquanto outros pixels da região, permanecem no centro e tem seu valor mantido, ocorrendo dessa forma uma transição suave em animações. O efeito de *vignetting* pode ser facilmente atingido ao se utilizar uma imagem que codifica uma função de *falloff* em relação ao centro da imagem.

5.3

Formatos de mapas de iluminação

Os Mapas de iluminação utilizados em IBL são omnidirecionais, logo, permitem o mapeamento das condições de iluminação de todas as direções do mundo. Para acessar os valores de iluminação referentes a uma direção $D = (D_x, D_y, D_z)$ no mundo, é necessário transformar essa direção nas coordenadas de textura que correspondem a esta direção no ambiente real codificado. Tal transformação depende do formato utilizado para armazenar os mapas de iluminação. Os principais formatos para essa tarefa são apresentados a seguir.

5.3.1

Mapas esféricos

O mapeamento esférico se baseia na transformação das coordenadas de um vetor direção em coordenadas esféricas com domínio $u \in [0, 1], v \in [0, 1]$. A equação de mapeamento é a seguinte:

$$r = \frac{\sin(\frac{1}{2} \arccos(-D_z))}{2\sqrt{D_x^2 + D_y^2}}$$

$$(u, v) = \left(\frac{1}{2} + rD_x, \frac{1}{2} - rD_y \right)$$

O mapeamento esférico representa o mundo a partir de sua reflexão em uma esfera espelhada, assumindo uma projeção ortográfica. Como outros formatos para armazenagem de fotografia omnidirecional, a esfera codifica todas as direções do ambiente. É fácil perceber que a direção $D = (0, 0, -1)$ é mapeada para o centro da esfera. Uma importante propriedade do mapeamento esférico é que quaisquer regiões de mesmo ângulo sólido na cena, irão mapear a mesma quantidade de área na imagem. O formato esférico apresenta alguns problemas como distorções no mapeamento da

parte de trás do ambiente, bem como problemas de mapeamento perto dos extremos da esfera. Outros formatos como *angular maps* podem ser utilizados, mas também apresentam problemas de amostragem perto das extremidades. O formato mais comum para mapeamento de ambiente é o de mapas cúbicos.

5.3.2 Mapas cúbicos

Mapeamento cúbico (*cube mapping* [25]) é de longe o mais popular de todos os métodos para mapeamento de ambiente. Tal fato se deve a sua velocidade e flexibilidade. Um mapa de ambiente cúbico é obtido ao posicionar uma câmera no centro do ambiente e projetar o ambiente nas faces do cubo com centro na posição da câmera. Cada face representa então o ambiente visto pela câmera em uma particular direção.

A codificação de uma direção do mundo em coordenadas de textura é feita utilizando a coordenada de maior magnitude da direção d para seleção da face do cubo. A partir da seleção, as coordenadas remanescentes do vetor d são divididas pelo valor absoluto da maior coordenada e remapeadas para o domínio $[0,1]$ de coordenadas de textura. A Figura 5.6 mostra um exemplo de mapa de ambiente cúbico.



Figura 5.6: Mapa de iluminação cúbico

Uma das principais fatores pelo sucesso desse método se deve à facilidade da geração de mapas de ambiente em tempo real por um renderizador em comparação com os métodos de codificação esférico e

variantes. Dessa forma, esses mapas podem ser gerados em tempo real. Outro fator se deve às suas características de amostragem mais uniforme em relação ao método de Blinn. Além disso o mapeamento cúbico é independente do ponto de vista. Uma desvantagem desse método é a necessidade de suporte a hardware. Porém como tal suporte já fornecido a algumas gerações de placas gráficas, esse modelo é o preferido para aplicações em tempo real.

5.4

IBL em tempo real

Até agora, as técnicas de IBL explicadas nesse capítulo se baseiam em renderizadores de iluminação global, que apesar de apresentar resultados finais ótimos, não são capazes de apresentar renderizações em taxas interativas, muito menos em tempo real. A próxima seção se dedica a explorar as técnicas de IBL em tempo real. É importante lembrar que as técnicas apresentadas são aproximações das técnicas não interativas, porém em alguns casos conseguem atingir o mesmo resultado final. Os códigos e imagens apresentados nesta seção foram desenvolvidos pelo autor desta dissertação.

5.4.1

Environment mapping

Environment mapping ou mapas de ambiente [21, 1, 25] é uma técnica muito a frente de seu tempo, precursora de *image based lighting*, onde uma imagem omnidirecional de um ambiente é mapeada na superfície de um objeto para produzir a aparência do objeto refletindo o ambiente. O mapa de ambiente é mapeado no objeto de acordo com a normal em cada ponto de sua superfície. Dessa forma o processo de renderização é extremamente rápido, uma vez que depende apenas de técnicas de mapeamento de texturas tradicional. *Environment mapping* é a técnica que apresenta os melhores resultados na simulação do componente especular de objetos, especialmente em objetos reflexivos. A cálculo do vetor de reflexão R é:

$$R = E - 2(E \cdot N)N$$

Onde N é a normal em um ponto p da superfície, e E é o vetor partindo da posição da câmera até o ponto p . A partir desse vetor direcional são

calculadas as coordenadas de textura utilizadas para acessar o mapa de ambiente e definir a cor do pixel correspondente ao ponto p . Tal esquema é ilustrado na Figura 5.7. O mapa de ambiente permite que objetos aparentem refletir o ambiente que o cerca, porém esse método não é capaz de refletir outros objetos sintéticos na vizinha do objeto com o mapa de ambiente aplicado, resultando em uma aparência menos realista. Para renderizações contendo apenas objetos distantes entre si, de forma que a reflexão fosse pouco notada, ou apenas um objeto sintético, os resultados são praticamente indistinguíveis de técnicas baseadas em *ray tracing*. O *vertex shader* a seguir calcula o vetor de um vértice v até a posição da câmera, que será posteriormente utilizado no *pixel shader*.

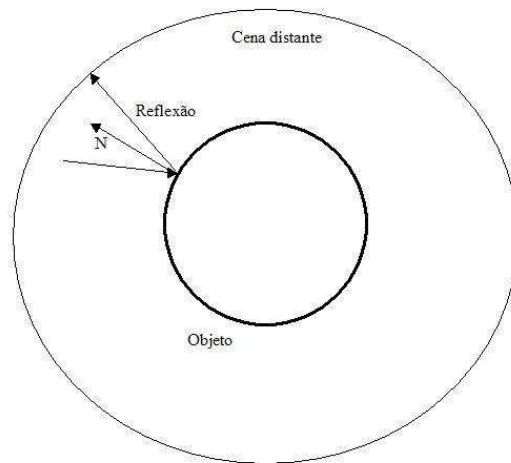


Figura 5.7: Acesso ao valor de radiância da cena distante (*light probe*)

```
float3 LightDir;  
float4x4 matViewProjection;  
float4x4 matView;  
  
struct VS_INPUT  
{  
    float4 Position : POSITION0;  
    float3 Normal   : NORMAL;  
    float2 UV       : TEXCOORD;  
};  
  
struct VS_OUTPUT
```



```

{
    float4 Position : POSITION;
    float3 Eye       : TEXCOORD1;
    float3 Normal    : TEXCOORD2;
};

VS_OUTPUT vs_main( VS_INPUT Input )
{
    VS_OUTPUT Output;

    Output.Position = mul( matViewProjection ,
        Input.Position );

    float3 pos_eye = viewInverse[3].xyz;

    Output.Eye = (normalize(pos_eye - Input.Position));

    Output.Normal = Input.Normal;

    return( Output );
}

```

Primeiramente a posição do vértice é transformada para o espaço de recorte, a seguir a posição da câmera no espaço do objeto é calculada (repare que nesse exemplo, o espaço do mundo é o mesmo espaço do objeto). Finalmente o vetor partindo do vértice até a posição da câmera é calculado e o vetor normal do vértice é passado ao *pixel shader*. O código do *pixel shader* é apresentado a seguir.

```

samplerCUBE illumMap;

struct PS_INPUT
{
    float4 Position :POSITION;
    float3 Eye       :TEXCOORD1;
    float3 Normal    :TEXCOORD2;
};

float4 AmbientColor;

```

```
float4 ps_main(PS_INPUT In) : COLOR0
{
    float3 vReflect = normalize(reflect
        (In.Eye, In.Normal));

    return texCUBE(illumMap, -vReflect);
}
```

O pixel shader tem como função acessar o valor de iluminação associado a direção do vetor de reflexão. O formato de codificação utilizado para o mapa de iluminação é o *cube map*. A Figura 5.8 mostra um modelo renderizado



Figura 5.8: Cena renderizada pelo autor utilizando utilizando mapa de iluminação.

utilizando um mapa de iluminação em alta precisão. O material associado ao modelo simula um superfície perfeitamente reflexiva.

5.4.2 Mapas de iluminação difusos

Os mapas de ambiente utilizados anteriormente permitem uma boa aproximação da solução de iluminação global para superfícies reflexivas ou para a componente especular de certos materiais. Esse modelo é o mais difundido e geralmente associado às técnicas de *environment mapping*. Porém é possível simular o comportamento de superfícies predominantemente difusas com uma técnica similar.

Ramamoorti e Hanrahan [34] observaram que a reflexão da iluminação incidente em um objeto difuso é semelhante a uma versão borrada da iluminação em um objeto perfeitamente reflexivo. Dessa maneira, o comportamento de objetos difusos pode ser simulado utilizando tais imagens e soluções de iluminação de objetos reflexivos, mantendo assim, o cerne da técnica de *environment maps*. Para simular a reflexão do ambiente em superfícies difusas a técnica de *environment mapping* deve ser combinada com um estágio de pré-processamento da imagem original. Diferentes filtros podem ser utilizados na realização dessa tarefa [1, 49].

A reflexão lambertiana difusa com *environment mapping* se baseia tipicamente em um lóbulo esférico da função cosseno como filtro de convolução no mapa de iluminação original gerando um mapa de irradiância ambiente. De forma análoga aos mapas de ambiente tradicionais, um vetor direção é utilizado para acessar as coordenadas de textura no mapa. Porém o vector utilizado aqui é a normal N em um ponto p da superfície do objeto difuso. O processo de convolução dessas imagens pode se tornar computacionalmente pesado devido ao tamanho das imagens e da complexidade do algoritmo. Porém devido ao fato dos mapas de irradiância ambiente não apresentarem áreas com muitos detalhes, versões menores das imagens podem ser utilizadas, como 32x32 pixels por exemplo. Em [49] é mostrado que uma decomposição em Harmônicos esféricos pode representar a iluminação incidente de forma precisa. Ramamoorthi e Hanrahan demonstraram posteriormente que utilizando os primeiros nove termos da decomposição aproximam com grande precisão a iluminação incidente. Uma boa referência para iluminação baseada em Harmônicos esféricos se encontra em [37].

A utilização de imagens HDR para a criação desses mapas é fundamental, uma vez que imagens tradicionais saturam os valores de pixels. Dessa forma quando o filtro de convolução é aplicado na imagem, a imagem se torna mais escura e perde seus detalhes. A Figura 5.9 mostra um de uma cena renderizada utilizando de um mapa de iluminação HDR que

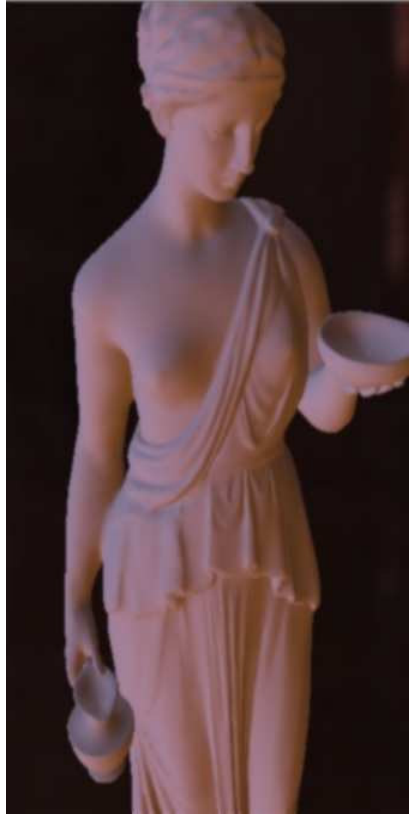


Figura 5.9: Objeto renderizado pelo autor com mapa de iluminação difuso sofreu o processo de convolução e foi utilizado para simular as condições de iluminação do ambiente em um objeto difuso.

5.4.2.1

Oclusão de Ambiente

A técnica anterior para utilização de mapas de ambiente em superfícies difusas apresenta o problema de não apresentar sombras. Para corrigir tal falha, um algoritmo posterior de sombras como *shadow mapping* ou *shadow volumes* deve ser utilizado. Outra maneira, que geram sombras em objetos difusos, bastante realista é a técnica de oclusão de ambiente *Ambient occlusion mapping* [17].

Oclusão de ambiente pode ser utilizada para aproximar as questões de visibilidade resolvidas pelos algoritmos de ray tracing com iluminação global. A técnica consiste basicamente de três passos:

- 1– Renderizar o ambiente utilizando um algoritmo de iluminação global.
- 2– Guardar o resultado da renderização em um mapa

de oclusão .

3– Renderizar a cena iluminada pelo mapa de iluminação difuso

4– Modular o resultado da iluminação anterior com o mapa de oclusão

Na primeira etapa do processo o ambiente é renderizado utilizando um ambiente de iluminação homogêneo de cor branca e os materiais das superfícies da cena são definidos como difusos. Um algoritmo de iluminação global como *Monte Carlo ray tracing* pode então ser utilizado para calcular a solução de iluminação para a cena. Esse passo é responsável pela determinação dos valores de iluminação considerando a interferência de um objeto em outro no cálculo de iluminação. Além de permitir cálculos precisos para auto-sombreamento *self-shadowing*, ou seja, parte de um objeto projetarem sombras nele mesmo. O próximo passo consiste em utilizar a técnica de mapas de iluminação difusos descrita anteriormente para iluminar (colorir) a cena. Com os dois resultados prontos basta multiplicá-los para obter o resultado final. É fácil perceber que o primeiro passo serve como um determinante do termo de visibilidade encontrado em BRDFs de superfícies.

É importante notar que a primeira etapa do processo é computacionalmente pesada e não pode executada em tempo real, porém como esta depende basicamente da geometria da cena a ser computada, esta parte pode ser pré-computada e armazenada em texturas para serem utilizadas nas partes posteriores do algoritmo, que apresentam performance de tempo real. Devido a esta limitação, essa técnica é aplicada apenas em cenas estáticas.

Apesar de fornecer uma boa aproximação para iluminação difusa com sombras, a técnica de oclusão de ambiente não apresenta exatamente os mesmos resultados caso a cena fosse totalmente iluminada por IBL padrão. Mais precisamente, as sombras apresentadas por esse método tendem a ser mais suaves que as apresentadas por IBL tradicional.

5.4.3

Mapas de ambiente dinâmicos

As técnicas descritas anteriormente permitem a geração de mapas de iluminação *HDR*, capazes de modelar a radiância real do ambiente e

consequentemente gerando imagens com maior fidelidade. Um problema que ocorre com a simples utilização de mapas de iluminação HDR, sem algoritmo de *ray-tracing* é que tais mapas não são capazes de interagir com mudanças na cena.

No capítulo 3 é apresentada uma técnica que permite a integração de objetos sintéticos em ambientes reais. Apesar dessa técnica se basear em algoritmos de iluminação global, sua estrutura de organização da cena é útil para a utilização de mapas de iluminação dinâmicos em tempo real. A técnica separa os objetos sintéticos do ambiente que os ilumina. O ambiente é denominado cena distante, e não sofre interação dos objetos, uma vez que a iluminação proveniente dos objetos pode ser facilmente ignorada, devido a cena distante ser muito mais emissiva que os objetos. A cena local é composta pelos objetos na proximidade do objeto sintético a ser renderizado e suas reflexões devem aparecer nesses. A parte final é renderização dos objetos sintéticos propriamente ditos, que devem apresentar tanto a reflexão dos objetos reais quanto os sintéticos que estejam próximos. Como o propósito é a integração de diferentes objetos sintéticos, permitindo a reflexão do ambiente e de outros objetos na proximidade, um algoritmo de várias passadas é necessário. A técnica base consiste em renderizar diferentes pontos de visada da câmera e criar um mapa de ambiente em formato de *cube map*. Esse formato é escolhido pois além de apresentar a melhor qualidade das renderizações, descrito na seção anterior, as principais APIs gráficas como OpenGL e Direct3D já permitem a utilização das faces do *cube map* como superfície de renderização. Dessa maneira o processo de codificar as diferentes renderizações em um único mapa esférico é suprimido. O algoritmo para renderização utilizando mapas de iluminação em alta precisão criados em tempo real é:

Para cada face **do** mapa de iluminação

Renderizar cena com ponto de visada no centro
da face **do** mapa de iluminação;

Para cada objeto reflexivo na cena

Renderizar objeto com mapa de iluminação criado;

Reduzir o tamanho **do** framebuffer

Aplicar bright pass filter;

Adicionar efeitos de iluminação;

Aplicar tonemapping;

É importante destacar que objetos reflexivos não apresentam reflexão uma vez que são desenhados na mesma passada. Caso a reflexão de um objeto em outro seja interessante, basta inclui-lo na lista de objetos a serem renderizados na primeira passada. Esta técnica é muito útil para representar objetos com altos coeficientes especular e reflexivos. Apesar de ser possível renderizar a cena e aplicar a técnica descrita anteriormente para objetos difusos, o cálculo das convoluções pode acarretar em taxas de atualização não interativas.