

3

Técnicas de redução do chamoscamento e de filtragem

Neste capítulo, estudamos técnicas para a obtenção de sombras de boa qualidade sem alterar a amostragem da cena, ou seja, sem tentar melhorar o aproveitamento da resolução do mapa de sombras, como será feito no Capítulo 4. Em particular, analisamos o problema de chamoscamento e a necessidade de filtrar as bordas das sombras, tornando-as suaves e disfarçando o serrilhamento. Discutimos também a adequação de cada técnica à visualização de modelos CAD. Como contribuição adicional, apresentamos na Seção 3.1.2 um método para o cálculo de amostras virtuais [10], em hardware programável, utilizando apenas um programa de fragmento.

Em todo o restante da dissertação, um mapa de sombras será considerado uma textura bidimensional acessada por coordenadas homogêneas (s,t,p,q) , geradas por técnicas de textura projetiva a partir do ponto-de-vista da fonte de luz. As coordenadas (s/q) e (t/q) , no intervalo $[0,1]$, determinam o texel a ser acessado para a obtenção da amostra do mapa (z_{mapa}). A coordenada (p/q) , por sua vez, corresponde à profundidade da amostra da câmera em relação à fonte de luz (z_{pixel}). O teste de sombra resulta num fragmento iluminado quando $z_{pixel} = z_{mapa}$ (ou, usando um teste mais relaxado, quando $z_{pixel} \leq z_{mapa}$).

Ao estudarmos os problemas de chamoscamento e serrilhamento, devemos estar atentos a algumas propriedades de transformações perspectivas. Quando renderizamos uma cena a partir de um certo ponto de vista, a projeção perspectiva faz com que os valores de profundidade (z) sejam amostrados não-uniformemente de maneira hiperbólica ($1/z$). Além disso, quanto mais longe do centro de projeção (a câmera ou uma fonte de luz tipo *spot* para o mapeamento de sombras), a amostragem se torna cada vez menos densa, ou seja, as amostras se tornam mais distantes entre si. No mapeamento de sombras, isso pode dar origem à situação conhecida como “volumes de visão conflitantes” (*dueling frusta*): objetos muito distantes da luz podem ser o foco de atenção na visão da câmera. Nessas condições, a câmera requer uma amostragem densa e com alta precisão dos

objetos, mas o mapa de sombras provê uma amostragem baixa e com pouca precisão. Como veremos, tal situação agrava seriamente tanto o chamoscamento quanto o serrilhamento.

3.1. Tratamento do chamoscamento

Começamos analisando o problema de chamoscamento (também conhecido como auto-sombreamento incorreto, “*incorrect self-shadowing*”) e investigando técnicas para tratá-lo em visualizações de modelos CAD. A Figura 4 ilustra os artefatos em padrões de Moiré que surgem em toda a cena no algoritmo tradicional.

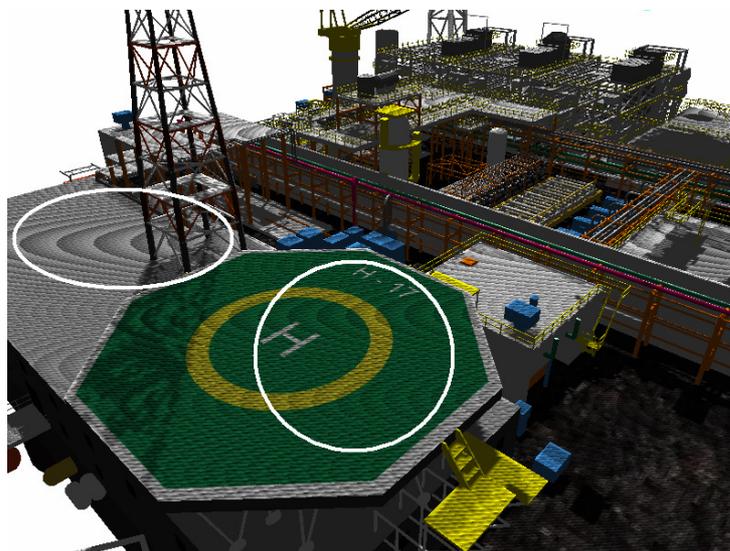


Figura 4 – Exemplo de chamoscamento em padrões de Moiré, com destaque para as regiões onde o efeito é mais perceptível

O chamoscamento pode ser atribuído a questões de precisão numérica e à resolução limitada do mapa de sombras. Primeiramente, lembramos que o teste de sombra deveria produzir um pixel iluminado quando o valor de profundidade armazenado no mapa fosse idêntico ao do ponto sendo testado, indicando que ele pertence a um oclisor visível para a fonte de luz. No entanto, como esses valores são amostrados com precisão diferente e, além disso, a amostra tomada no espaço da câmera passa por uma transformação para o espaço da fonte de luz antes da comparação, a igualdade pode não se verificar. O uso do teste $z_{pixel} \leq z_{mapa}$, mais relaxado, alivia apenas parcialmente o problema. Outro procedimento útil é gerar z_{pixel} e z_{mapa} sempre como distâncias lineares de cada ponto até a fonte.

Para entender a influência da resolução do mapa de sombras, consideremos a Figura 5a. As amostras de profundidade armazenadas no mapa de sombras e as obtidas pela câmera durante a renderização da cena são tomadas nos centros dos respectivos texels e pixels. Dessa forma, as amostras da câmera, quando transformadas para o espaço da fonte de luz, não se alinham com as amostras do mapa [10]. Assim, mesmo numa superfície plana, as profundidades diferem, ocasionando o chamoscamento. Tal diferença torna-se menor conforme diminui o espaçamento entre as amostras, de modo que o problema não existiria num mapa de resolução infinita. Por outro lado, o pior caso de uma mapa real ocorre na situação de visões de volume conflitantes, onde a distância entre amostras do mapa é muito maior que a verificada entre amostras da câmera.

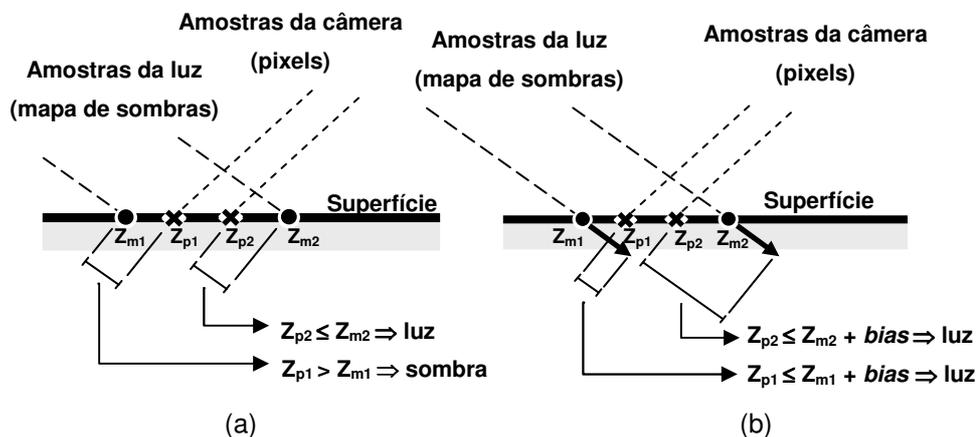


Figura 5 – (a) Chamoscamento por desalinhamento das amostras da fonte de luz e da câmera; (b) Efeito do *bias* para a eliminação do chamoscamento.

3.1.1. Técnicas baseadas em *bias*

Como foi mencionado na Seção 2.1, o problema de chamoscamento é tradicionalmente resolvido com o acréscimo de um pequeno *bias* aos valores do mapa de sombras. Esse valor deve ser suficiente para que as amostras do mapa fiquem atrás de qualquer amostra da câmera na vizinhança (Figura 5b). Esse procedimento, porém, “empurra” as sombras para longe da fonte de luz. Um *bias* muito grande pode, por exemplo, deslocar a sombra de um objeto que está no chão, causando a impressão de que está flutuando. O efeito, conhecido como vazamento de luz (*light bleeding*), encontra-se ilustrado na Figura 6.

A escolha de um bom valor de *bias* depende da cena, das dimensões dos objetos, das posições da câmera e das fontes de luz, e também da resolução do mapa de sombras nas coordenadas (s,t). Assim, estipular um bom *bias* pode ser uma tarefa difícil, exigindo cuidado até mesmo na modelagem da cena. Na verdade, em alguns casos, é impossível eliminar o chamuscamento e evitar vazamentos de luz simultaneamente, tornando-se necessário aumentar a resolução do mapa.

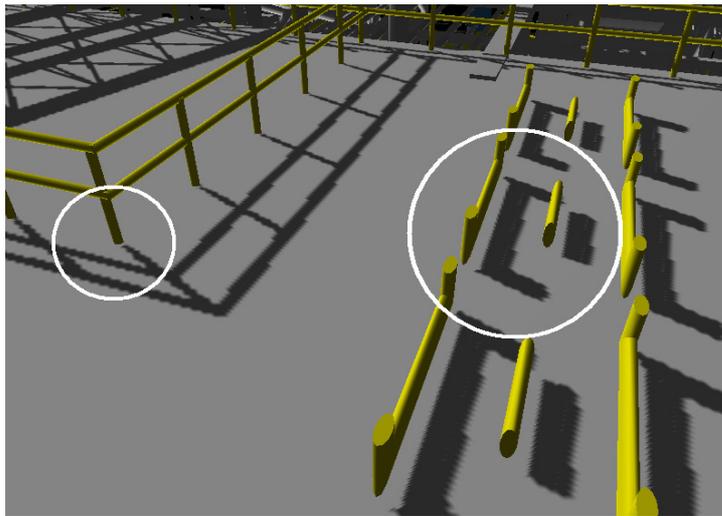


Figura 6 – Efeito de vazamento de luz devido ao uso de um *bias* exagerado

Vimos na Seção 2.1 algumas técnicas que foram propostas visando automatizar, de forma adaptativa, a escolha do *bias*. O valor armazenado no mapa de sombras por essas técnicas pode ser generalizado pela seguinte equação, onde z_1 e z_2 se referem à profundidade da primeira e da segunda superfícies mais próximas da fonte de luz, respectivamente [12]:

$$z_{mapa} = z_1 + z_{bias}(z_1, z_2) \quad (1)$$

Tabela 1 – Funções típicas para z_{bias}

Técnica	Função z_{bias}
(a) Constante	z_{offset}
(b) <i>Second-depth</i> [10]	$z_2 - z_1$
(c) <i>Midpoint</i> [11]	$(z_2 - z_1)/2$
(d) <i>Dual-depth</i> [12]	$\min((z_2 - z_1)/2, z_{max})$

De acordo com a Equação (1), o *bias* passa a ser encarado como uma função, podendo assumir a forma dos diferentes algoritmos, como na Tabela 1.

A função (b) armazena no mapa a segunda superfície mais próxima da fonte de luz ao invés da primeira. Assim, numa cena composta apenas por sólidos fechados, ela transfere a ambigüidade do teste de sombra para uma superfície voltada para trás em relação à fonte, onde os cálculos normais de iluminação garantem o total sombreamento. Com essa restrição, tal função fornece sempre resultados muito bons.

Por outro lado, as funções (c,d) trabalham com o ponto médio entre as duas superfícies (voltadas para a frente) mais próximas da fonte de luz, numa tentativa de eliminar a ambigüidade do teste de sombra. O limite superior de (d) torna-se necessário para evitar artefatos no caso de grandes discontinuidades entre as superfícies ou quando a segunda delas é quase paralela aos raios de luz [12]. Poderíamos pensar ainda em incluir um limite inferior z_{\min} , de modo a garantir a ausência de chamuscamento no caso de superfícies muito próximas umas das outras. No entanto, se pudermos encontrar um valor de z_{\min} pequeno o suficiente para não causar vazamentos de luz, basta utilizar $z_{\text{offset}} = z_{\min}$ constante, recaindo em (a). Apesar de produzirem resultados excelentes, as técnicas de ponto médio requerem duas passadas para a geração do mapa de sombras. Assim, seu custo pode se tornar elevado demais em cenas dinâmicas ou quando forem utilizadas técnicas de reparametrização (Capítulo 4), que precisam reconstruir o mapa de sombras sempre que a câmera se move.

Para escolhermos uma técnica adequada a modelos CAD, devemos considerar que o uso de técnicas de reparametrização é sempre desejável e que os modelos nem sempre são compostos apenas de sólidos fechados. Assim, muitas vezes será necessário fazer o ajuste manual de *bias* constante (a). Se o hardware gráfico tiver grande capacidade de processamento geométrico, podem-se conseguir resultados ainda melhores com a técnica *dual-depth* (d).

3.1.2. Alinhamento de amostras ou reconstrução

Além das técnicas de *bias* mencionadas na seção anterior, outro procedimento muito útil na eliminação de artefatos de chamuscamento é a correção do alinhamento entre as amostras da câmera e do mapa de sombras. Deve-se perceber que, na verdade, temos um problema de reconstrução de uma função bidimensional a partir de uma representação amostrada.

Como uma primeira abordagem, podemos estimar o valor de profundidade que estaria armazenado no mapa de sombras em coordenadas alinhadas com a amostra da câmera. A reconstrução mais simples, nesse caso, seria a interpolação bilinear dos valores do mapa nos texels mais próximos. Esse método apresenta alguns problemas independente da função de reconstrução utilizada: primeiro, torna-se necessário realizar vários acessos ao mapa para a obtenção de um único valor de teste, o que pode reduzir o desempenho do algoritmo. Segundo, a função amostrada no mapa de sombras apresenta muitas discontinuidades entre pontos de oclusores diferentes. Os valores interpolados nessas regiões não têm significado útil e podem dar origem a sombras grossas e vazamentos de luz, como mostra a Figura 7. Finalmente, o método não se adapta bem à utilização do filtro PCF, como veremos na seção 3.2.2.

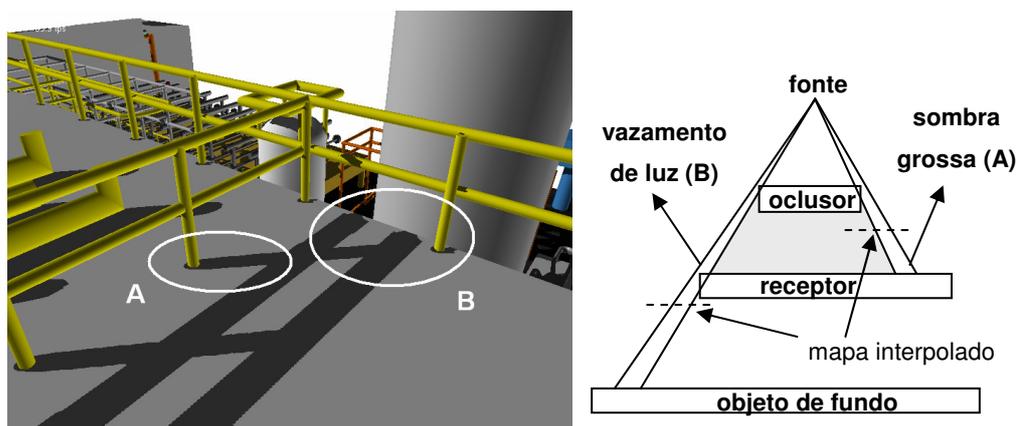


Figura 7 – Sombras grossas (A) e vazamento de luz (B) com interpolação do mapa

Uma abordagem alternativa, que corresponde à reconstrução das amostras da câmera, foi proposta por Wang & Molnar [10]. Quando analisamos um ponto, em vez de compararmos com o mapa de sombras a profundidade do próprio pixel, tomamos a profundidade de uma amostra virtual, tomada sobre o plano tangente à superfície naquele ponto, completamente alinhada com a amostra do mapa. Esse método não requer acessos adicionais ao mapa de sombras e se adequa bem ao filtro PCF, sendo então mais recomendado. Artefatos de chamuscamento podem surgir em regiões onde o plano tangente é quase paralelo aos raios de luz incidentes, como mostra a Figura 8. No entanto, essas regiões têm naturalmente iluminação reduzida, de modo que o efeito é pouco perceptível.

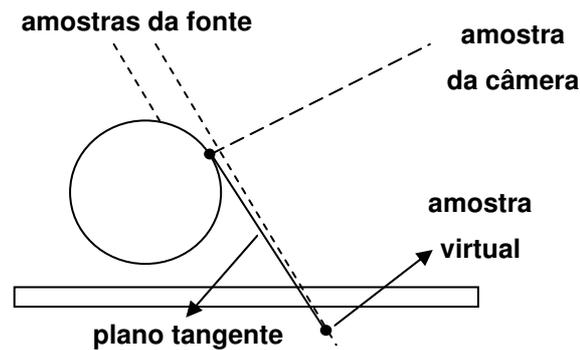


Figura 8 – Sombreamento incorreto onde o plano tangente é quase paralelo à luz

Em seu trabalho, Wang & Molnar sugerem a obtenção do plano tangente a partir da posição do ponto considerado e da normal à superfície, informações geralmente disponíveis durante o processo de renderização. No entanto, em hardware gráfico programável atual, isso requer a utilização de um programa de vértice (*vertex shader*) apenas para repassar esses dois valores ao programa de fragmento (*fragment shader*). Isso pode ser inconveniente, pois torna necessário implementar manualmente toda a funcionalidade desejada de processamento geométrico e iluminação. Alternativamente, propomos a seguir um método para a obtenção de amostras virtuais utilizando exclusivamente um programa de fragmento. A técnica pode ser útil para aplicações que desejem manter o mapeamento de sombras independente de programas de vértice, e seu custo adicional limita-se a algumas operações de soma e multiplicação por fragmento.

Chamemos ddx e ddy as variações nas coordenadas (s, t, p, q) do mapa de sombras entre o fragmento atual e seu vizinho em x e y da tela, respectivamente. Como notação, usamos $ddx(s)$, por exemplo, para nos referirmos à variação apenas na coordenada s . Os valores $ddx(s, t, p, q)$ e $ddy(s, t, p, q)$ podem ser obtidos diretamente num programa de fragmento. A amostra da câmera correspondente ao fragmento atual tem coordenadas $f_0 = (s_0, t_0, p_0, q_0)$ no espaço do mapa. Nosso objetivo é encontrar as coordenadas $f_v = (s_v, t_v, p_v, q_v)$ da amostra virtual.

Considere uma amostra virtual f_v já determinada. Se a transformarmos de volta ao espaço da imagem, teremos um ponto distante n_x pixels do fragmento atual na direção x da tela e n_y pixels na direção y . Além disso, como a amostra virtual está localizada sobre o plano tangente, podemos expressá-la usando uma combinação linear de ddx e ddy , como a seguir:

$$f_v = f_0 + n_x ddx(s, t, p, q) + n_y ddy(s, t, p, q) \quad (2)$$

Na verdade, como nosso objetivo é alinhar a amostra virtual com algum ponto do mapa de sombras, as coordenadas (s_v/q_v) e (t_v/q_v) já são conhecidas, e precisamos determinar apenas (p_v/q_v) . Chamemos de $(\Delta s, \Delta t)$ as variações, após a divisão de perspectiva, necessárias para a obtenção da amostra. As distâncias n_x e n_y que resultariam em $(\Delta s, \Delta t)$ podem ser encontradas pelo sistema abaixo:

$$\begin{cases} n_x \cdot (ds/dx) + n_y \cdot (ds/dy) = \Delta s \\ n_x \cdot (dt/dx) + n_y \cdot (dt/dy) = \Delta t \end{cases} \quad (3)$$

Na Equação (3), as derivadas se referem às variações de (s, t) , após a aplicação da divisão perspectiva, com as coordenadas (x, y) da tela em número de pixels:

$$\frac{ds}{dx} = \frac{s_0 + ddx(s)}{q_0 + ddx(q)} - \frac{s_0}{q_0} = \frac{q_0 ddx(s) - s_0 ddx(q)}{q_0 (q_0 + ddx(q))} \quad (4)$$

As equações para os demais termos que aparecem na Equação (3) são análogas. Resolvendo agora o sistema, temos:

$$\begin{aligned} n_x &= [\Delta s (dt/dy) - \Delta t (ds/dy)] / \det \\ n_y &= [\Delta t (ds/dx) - \Delta s (dt/dx)] / \det \\ \det &= (ds/dx)(dt/dy) - (dt/dx)(ds/dy) \end{aligned} \quad (5)$$

Para eliminarmos o cálculo intermediário de n_x e n_y , substituímos (5) na Equação (2). Obtemos assim a expressão final para a amostra virtual:

$$f_v = f_0 + (\Delta s \Delta F_s + \Delta t \Delta F_t) / \det \quad (6)$$

onde :

$$\begin{aligned} \Delta F_s &= (dt/dy) ddx(s, t, p, q) - (dt/dx) ddy(s, t, p, q) \\ \Delta F_t &= (ds/dx) ddy(s, t, p, q) - (ds/dy) ddx(s, t, p, q) \end{aligned}$$

Infelizmente, a determinação do ponto central de um texel para calcular $(\Delta s, \Delta t)$ e realizar o alinhamento é um processo caro, que requer a utilização da função pouco eficiente “*floor()*” no programa de fragmento. Na Seção 3.2.2, ao apresentarmos o filtro PCF, vamos propor um método de amostras virtuais alinhadas às amostras do filtro, mas não ao centro dos texels do mapa de sombras. Assim, a Equação (6) ainda será utilizada, mas $(\Delta s, \Delta t)$ poderão ser obtidos de forma mais simples.

3.2. Tratamento do serrilhamento com filtragem

Muitas vezes, a amostragem da cena realizada pelo mapa de sombras é insuficiente para capturar a forma correta da silhueta dos objetos. Nesse caso, as bordas das sombras aparecem com aspecto serrilhado (*aliasing*), como mostra a Figura 9.

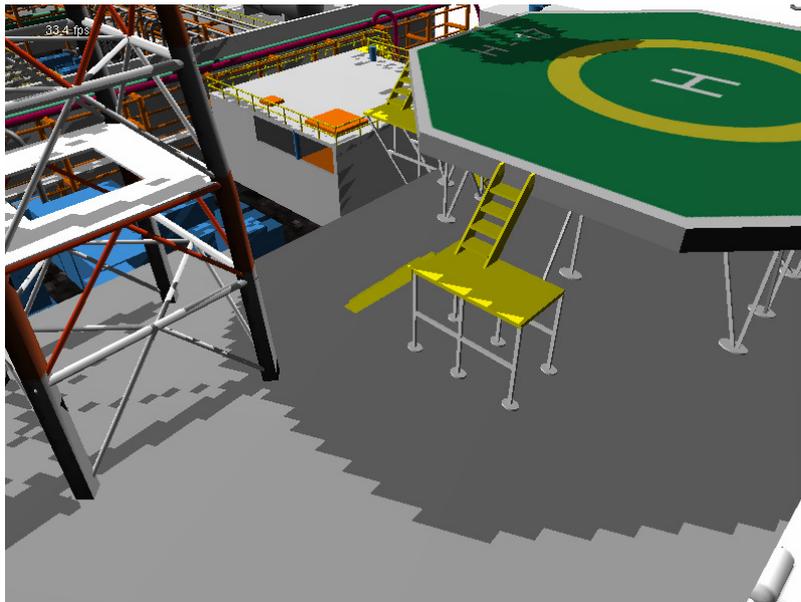


Figura 9 – Efeito de serrilhamento

Nesta seção, investigamos técnicas para tratar o serrilhamento por meio de algum tipo de filtro aplicado às bordas das sombras. O objetivo é “borrar” as sombras, dissimulando o efeito de serrilhamento e tornando suave a transição entre regiões iluminadas e sombreadas.

Antes de prosseguirmos, é importante ressaltar que o tratamento desejado não pode ser conseguido simplesmente filtrando-se o mapa de sombras. Como vimos na Seção 3.1.2, esse procedimento resultaria em valores interpolados de profundidades sem significado real, o que causaria o aparecimento de artefatos na imagem. Além disso, o teste de sombra continuaria sendo binário, produzindo sempre pixels totalmente iluminados ou sombreados, e não a região de transição suave que buscamos. Na verdade, queremos que o valor de iluminação gerado pelo filtro seja proporcional ao número de testes de sombra, no interior de seu núcleo (*kernel*), que indicariam que o pixel está iluminado.

Outro fato importante a se notar é que, apesar de obtermos sombras suaves com as técnicas apresentadas a seguir, as regiões de transição entre luz e sombra não correspondem a penumbras realistas. Como vimos na Seção 2.5, existem algoritmos específicos para o cálculo de regiões de penumbra. Esse tópico, no entanto, está fora do escopo desta dissertação.

3.2.1. Mapas de variância (VSMs)

Recentemente, Donnelly & Lauritzen [18] propuseram os chamados VSMs (*Variance Shadow Maps*), um método eficiente para a obtenção de sombras com bordas suavizadas e serrilhamento dissimulado. Nesta seção, revisitamos esses “mapas de sombra de variância” e mostramos por que eles são inadequados para a visualização de modelos CAD.

Na técnica proposta por Donnelly & Lauritzen, armazenam-se no mapa de sombras não só a distância z de cada oclusor até a fonte de luz, mas também z^2 . Esses dois valores são então submetidos a um filtro gaussiano bidimensional de tamanho arbitrário. Posteriormente, o acesso ao mapa durante a renderização da cena é feito com todos os filtros de *hardware* ativados como em uma textura normal. O resultado desse processo pode ser encarado como os momentos M_1 (média) e M_2 (média dos quadrados) de uma distribuição de profundidades, ou seja, das amostras do mapa interiores ao núcleo dos filtros utilizados. A definição estatística de M_1 e M_2 é:

$$\begin{aligned} M_1 &= E(z) = \int_{-\infty}^{+\infty} z p(z) dz \\ M_2 &= E(z^2) = \int_{-\infty}^{+\infty} z^2 p(z) dz \end{aligned} \quad (7)$$

Podemos calcular então a média μ e a variância σ^2 da distribuição:

$$\begin{aligned} \mu &= E(z) = M_1 \\ \sigma^2 &= E(z^2) - E^2(z) = M_2 - M_1^2 \end{aligned} \quad (8)$$

Finalmente, usamos a Desigualdade de Chebychev, que informa um limite máximo para a probabilidade de uma variável aleatória (no caso, z) ser maior que determinado valor (no caso, z_{pixel} , a profundidade da amostra da câmera):

$$P(z \geq z_{pixel}) \leq \frac{\sigma^2}{\sigma^2 + (z_{pixel} - \mu)^2} \equiv p_{max} \quad (9)$$

A probabilidade $P(z \geq z_{pixel})$ calculada em (9) representa a fração dos texels, na região de filtragem considerada, cuja profundidade é maior que a da amostra da câmera, ou seja, a fração dos testes de sombra que indicariam que o pixel está iluminado. Esse é exatamente o resultado que desejamos.

A proposta dos VSMs é utilizar o limite superior p_{max} como uma aproximação para a probabilidade P desejada. Como mostrado em [18], a aproximação é boa quando temos apenas um oclisor e um receptor, tornando-se exata no caso de ambos serem planos.

Infelizmente, a abordagem dos VSM apresenta um sério problema de vazamento de luz quando a cena possui elevada complexidade em profundidade, como é comum em modelos CAD. Quando temos dois ou mais oclusores distantes um do outro, o numerador de (9) tem valor considerável na silhueta de cada um deles. Assim, as regiões de uma superfície que receberiam sombras de mais de um oclisor simultaneamente aparecem envoltas por uma fina camada erroneamente iluminada. Esse efeito está destacado na parte A da Figura 10. O vazamento de luz também ocorre quando a silhueta do oclisor atinge uma fronteira da superfície receptora, como destacado em B na figura. Nesse caso, o valor M_1 armazenado no mapa é filtrado entre o oclisor e um objeto possivelmente bem mais distante que o receptor, alterando o resultado do teste de sombra. Uma terceira situação de vazamento de luz também acontece quando o oclisor é muito estreito ou quando utilizamos a técnica de *second-depth* na geração do mapa. Em locais onde o oclisor e o receptor estão muito próximos, o valor $(z_{pixel} - \mu)$ em (9) torna-se muito pequeno, resultando em iluminação quase total (parte C da figura).

3.2.2. Filtro de percentual mais próximo (PCF)

Com a inviabilidade de utilizar VSMs na visualização de modelos CAD, nos voltamos para a solução mais comumente adotada no tratamento do serrilhamento: um filtro especial chamado PCF (*Percentage-Closer Filter*) [14].

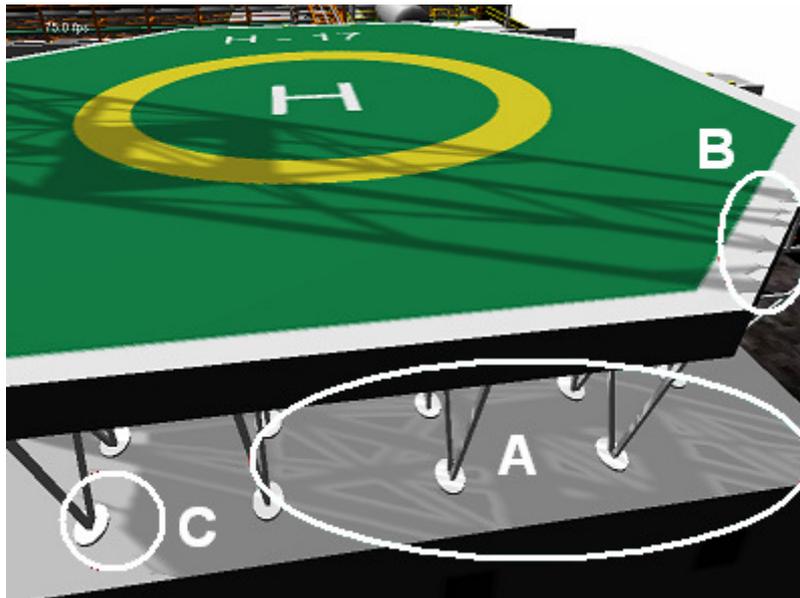


Figura 10 – Efeito de vazamento de luz com VSM

No algoritmo de PCF, realizam-se vários testes de sombra durante o processamento de cada pixel na renderização da cena. A amostra da câmera é comparada com diferentes amostras do mapa de sombras interiores ao núcleo do filtro. Considerando que o resultado de cada teste é um valor binário de iluminação, sendo 0 para sombra e 1 para luz, a filtragem pode ser aplicada diretamente a esses valores, resultando na iluminação parcial desejada.

É interessante notar que o núcleo do filtro PCF pode ser baseado nos pixels da imagem final ou nos texels do mapa de sombras. Na primeira abordagem, acessamos o mapa de sombras nas coordenadas associadas a cada pixel na vizinhança do ponto central considerado. O resultado é uma região de transição com espessura constante na tela, de forma que as sombras se mantêm suaves mesmo quando a câmera está distante. Entretanto, numa cena arbitrária, a intensidade do serrilhamento pode variar de um local para outro, o que cria uma situação inconveniente: regiões com pouco serrilhamento podem ser filtradas em excesso, enquanto regiões de serrilhamento intenso não são filtradas o suficiente (Figura 11).

Tomando como base os texels do mapa de sombras na construção do núcleo do filtro, devemos acessar sempre diferentes texels na vizinhança daquele correspondente ao ponto central. Assim, tomamos amostras com espaçamento sempre proporcional às dimensões que os texels ocupam no espaço do mundo. Esse procedimento apresenta uma grande vantagem: a região de filtragem será

sempre do tamanho necessário para dissimular adequadamente o serrilhamento, independente de sua intensidade local. Por outro lado, as sombras deixam de apresentar transição suave em regiões onde o serrilhamento não é visível na tela. Como nosso objetivo é tratar o serrilhamento e não propriamente obter sombras suaves, preferimos este método.



Figura 11 – Filtragem insuficiente com núcleo baseado nos pixels da imagem final

A aplicação de PCF com um núcleo regular, como na Figura 12a, resulta no aparecimento do efeito de faixas (*banding*) nas bordas das sombras: a transição entre regiões sombreadas e iluminadas não é totalmente suave, mas passa por níveis de tonalidade bem definida (Figura 13). Para eliminarmos esse problema, podemos utilizar pesos gaussianos nas amostras, bem como acrescentar um pequeno deslocamento aleatório ($\Delta x, \Delta y$), chamado *jitter*, como na Figura 12b. Adicionalmente, as quinas das sombras podem ser suavizadas mais corretamente com a utilização de um núcleo circular de raio R . Na Figura 12c, a área de todas as células do núcleo deve ser a mesma, o que pode ser conseguido com o seguinte mapeamento [17]:

$$\begin{aligned} x' &= R \sqrt{(y/y_{\max})} \cos(2\pi(x/x_{\max})) \\ y' &= R \sqrt{(y/y_{\max})} \text{sen}(2\pi(x/x_{\max})) \end{aligned} \quad (10)$$

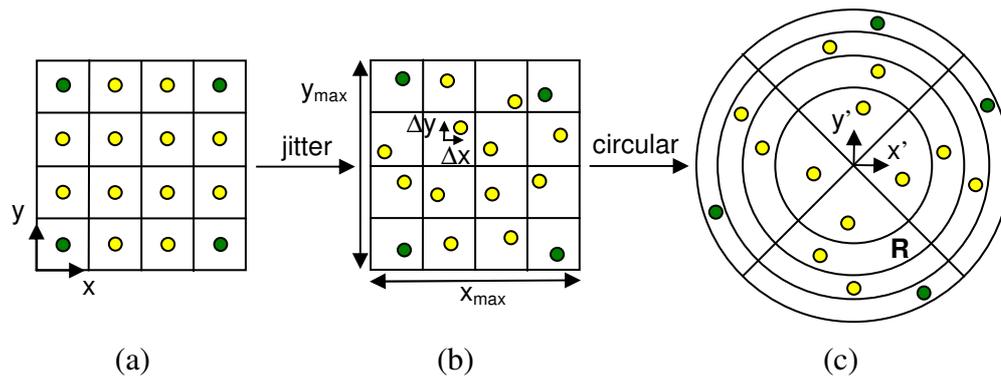


Figura 12 – Aprimoramentos ao kernel do filtro PCF, com amostras de teste em destaque



Figura 13 – Efeito de faixas usando PCF com núcleo regular

Outro aprimoramento a se considerar é a realização de uma amostragem adaptativa [17]. Primeiramente, são realizados testes de sombra apenas para as amostras mais externas do núcleo, destacadas na Figura 12. Se o resultado do teste for o mesmo em todas elas, o pixel é considerado sombreado ou iluminado sem necessidade de fazer os demais testes de sombra. Se algum resultado for diferente, é feito o refinamento com todas as amostras. Essa técnica permite diminuir o custo de rasterização do algoritmo, mas é sujeita a falhas. Quando as silhuetas dos objetos são muito complexas, dando origem a regiões de luz e sombra se alternando rapidamente, podem aparecer artefatos na imagem final (Figura 14). Felizmente, os artefatos são pouco perceptíveis em mapas de sombra com bom aproveitamento de resolução, de modo que a amostragem adaptativa poderá ser utilizada após a aplicação das técnicas do Capítulo 4.

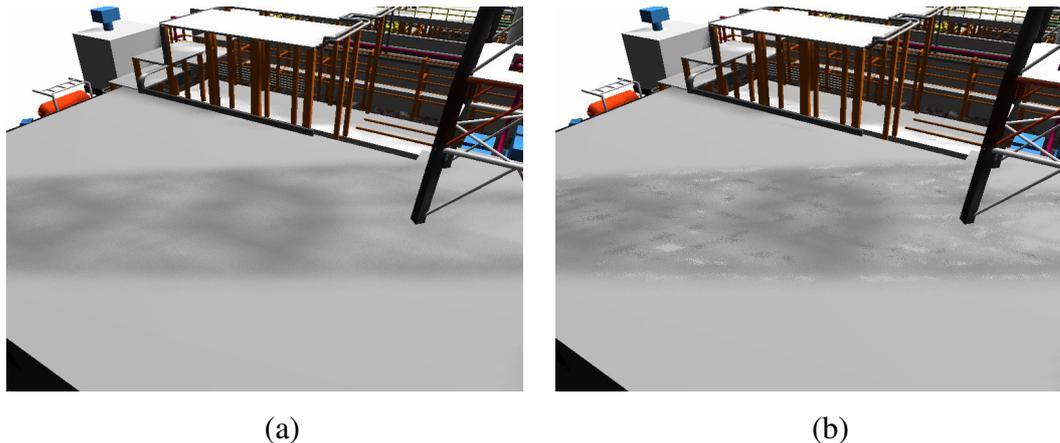


Figura 14 – Artefatos causados pela filtragem adaptativa em baixa resolução
(a) filtro normal (b) filtro com refinamento adaptativo

Como foi mencionado anteriormente, no algoritmo de PCF tradicional, uma única amostra da imagem (pixel) é comparada com diversas amostras do mapa de sombras (texels) ao longo do núcleo do filtro. Esse procedimento, no entanto, agrava o problema de chamoscamento, pois o *bias* escolhido deve ser suficiente para compensar o desalinhamento da amostra da imagem com o texel mais distante no interior do filtro (Seção 3.1). Para contornar essa dificuldade, podemos utilizar não apenas uma amostra da imagem, mas sim várias amostras virtuais, conforme a Seção 3.1.2, cada uma alinhada com a amostra correspondente do mapa de sombras. Perceba que, nesse caso, não existe a possibilidade inversa, de reconstruirmos cada amostra do mapa de modo a se alinhar com a amostra da câmera, pois isso anularia o efeito do PCF.

Em princípio, o alinhamento de cada amostra da imagem deveria ser feito com o centro do texel em que é tomada a amostra correspondente do mapa. No entanto, como mencionamos na Seção 3.1.2, a determinação desse centro requer uma computação pouco eficiente. Para evitarmos esse custo adicional, propomos um alinhamento parcial, suficiente para tornar o chamoscamento tão pequeno quanto na renderização sem PCF (e possivelmente inexistente). Neste método, as amostras da imagem devem ser reconstruídas nas coordenadas exatas dadas pelo núcleo do filtro. No caso do núcleo regular da Figura 12a, por exemplo, a amostra central não seria alterada, enquanto as demais amostras seriam reconstruídas pela Equação (6) utilizando-se, para Δs e Δt , múltiplos inteiros de $(1/res)$, onde “res” corresponde à resolução do mapa de sombras.

A Figura 15 ilustra o resultado obtido com um mapa de sombras simples após o tratamento por PCF 5×5 baseado nos texels do mapa, com *jitter*, núcleo circular e amostras virtuais da câmera alinhadas às amostras do núcleo. Um núcleo de 3×3 a 5×5 deve ser suficiente para qualquer aplicação quando aplicado com base nos texels do mapa.



Figura 15 – Sombras suaves com PCF 5×5 baseado nos texels do mapa, com *jitter*, núcleo circular e amostras da câmera alinhadas às amostras do núcleo