

## 4 Técnicas para melhor aproveitamento de resolução

Neste capítulo, estudamos dois tipos de técnicas que tentam melhorar o aproveitamento da resolução do mapa de sombras: técnicas de reparametrização (ou de transformação), que aplicam uma transformação para deformar a cena antes da amostragem; e técnicas de particionamento, que subdividem a cena em diferentes regiões, cada uma amostrada em sua própria área do mapa de sombras. Inicialmente, analisamos as causas do serrilhamento e definimos uma métrica de erro para quantificá-lo, revisitando [20,27,35]. Em seguida, estudamos as principais técnicas de reparametrização e particionamento em tempo real e discutimos sua adequação à visualização de modelos CAD.

Ao longo do capítulo, apresentamos ainda duas propostas para melhorar a qualidade obtida com os algoritmos estudados: na Seção 4.2.2, derivamos um parâmetro generalizado para adequar a técnica de LiSPSM [20] a diferentes ângulos entre as direções da luz e de visualização. Na Seção 4.3.2, propomos um esquema adaptativo de particionamento em profundidade [33-35].

### 4.1. Métrica de erro de serrilhamento

Para podermos analisar o efeito de serrilhamento, precisamos primeiramente quantificá-lo com algum tipo de métrica de erro. Para isso, consideremos uma região de uma superfície plana que pode ser vista simultaneamente pela câmera e pela fonte de luz, como mostra a Figura 16 para o caso bidimensional. Chamemos de “feixe da fonte de luz” ou simplesmente “feixe da luz” o conjunto de raios que se originam na fonte, atravessam um único texel de um mapa de sombras, e atingem a superfície. De forma semelhante, nos referimos a “feixe da câmera” como o conjunto dos raios que convergem para a câmera, partindo da superfície e atravessando um único pixel da imagem final. As projeções desses feixes sobre a superfície considerada tem área  $w_l'$  e  $w_c'$ , respectivamente.

Podemos quantificar o serrilhamento como a razão entre as áreas das regiões, na superfície considerada, vistas por um texel do mapa de sombras ( $w_l'$ ) e por um pixel da visão da câmera ( $w_c'$ ) [35]. Note que, quando a primeira é maior, cada texel do mapa determina o sombreamento de vários pixels na imagem, deixando a fronteira da sombra com aparência serrilhada. Definimos então o *erro de serrilhamento*:

$$m = \frac{w_l'}{w_c'} \quad (11)$$

O valor do erro  $m$  corresponde ao número de pixels, na imagem final, cujo sombreamento é determinado por um único texel do mapa de sombras. Assim, idealmente, o mapeamento de sombras deveria garantir  $m = 1$  em toda a cena. Quando  $m > 1$ , aparece o efeito de serrilhamento e, quando  $m < 1$ , ocorre superamostragem e parte da resolução do mapa é desperdiçada. Ressalta-se aqui que estamos desconsiderando o desalinhamento entre amostras da câmera e do mapa (Seção 3.1), ou seja, assumimos que os centros dos feixes da fonte e da câmera atingem a superfície no mesmo ponto.

Na Figura 16, os feixes da fonte e da câmera atingem a superfície com espessuras  $w_l$  e  $w_c$  no ponto central de contato e com ângulos de incidência  $\theta_l$  e  $\theta_c$ , respectivamente. Nessa situação, o erro de serrilhamento pode ser aproximado pela seguinte equação, derivada no Apêndice A:

$$m \approx \frac{w_l \cos(\theta_c)}{w_c \cos(\theta_l)} \quad (12)$$

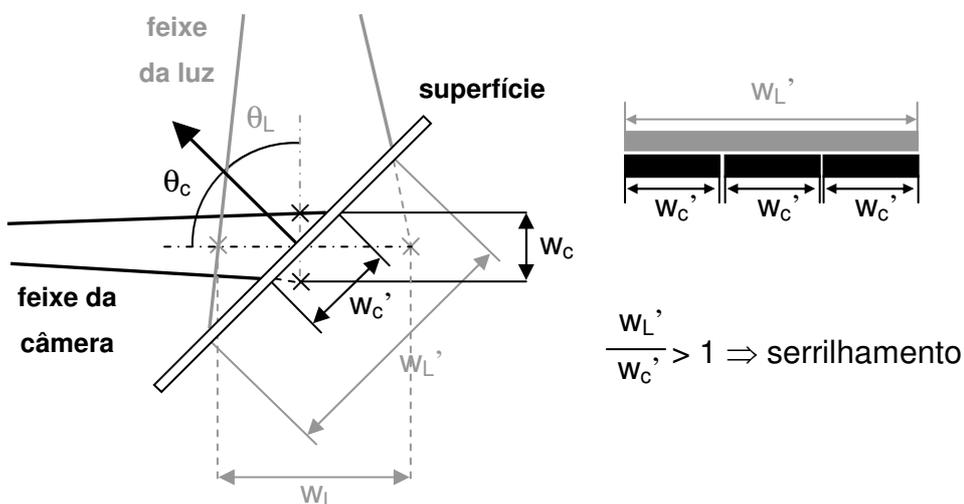


Figura 16 – Efeito de serrilhamento em mapeamento de sombras (adaptada de [35]).

Tomando a Equação (12) como base, podemos identificar dois tipos de serrilhamento [19,35]:

- serrilhamento de perspectiva (*perspective aliasing*): refere-se ao termo ( $w_l / w_c$ ). Esse tipo de serrilhamento surge por falta de resolução do mapa de sombras ou pelo fato de a câmera estar muito mais próxima da superfície do que a fonte de luz, visualizando uma área muito grande que foi amostrada numa área pequena do mapa. Esse problema, assim, é máximo na situação de *dueling frusta*.

- serrilhamento de projeção (*projection aliasing*): refere-se ao termo ( $\cos \theta_c / \cos \theta_l$ ). Ocorre quando os raios da luz estão quase paralelos à superfície receptora, o que faz que a sombra se “estique” em uma única direção de forma potencialmente ilimitada.

Note que, no cálculo de  $m$ , o serrilhamento de projeção depende da orientação de cada superfície da cena em relação a cada fonte de luz, podendo crescer de forma ilimitada em algumas situações. Assim, computá-lo requer uma análise complexa caso-a-caso, que pode comprometer a eficiência de um algoritmo (algumas técnicas, como ASMs [30], mapas de sombra adaptativos, aproximam o erro total com *mipmapping* ou alguma heurística). Por outro lado, o efeito não é tão notável: além de ocorrer apenas em uma direção, as superfícies quase paralelas a raios de luz são pouco iluminadas, tornando as sombras menos distintas [35]. Por esses motivos, o restante da análise levará em conta apenas o serrilhamento de perspectiva, o único tratado pelas técnicas investigadas nas próximas seções.

Seguindo a tendência de artigos recentes [20,25,27,34,35], a análise aqui apresentada tem o objetivo de minimizar a norma  $L_\infty$  definida por Chong [28]. No contexto desta dissertação, a métrica assume a seguinte forma:

$$L_\infty = \max_{p \in \Omega} |m| = \max_{p \in \Omega} \left| \frac{w_l}{w_c} \right| \quad (13)$$

Na Equação (13),  $\Omega$  é o conjunto de todos os pixels da imagem final em que alguma geometria é desenhada [28]. Assim, segundo a métrica, estamos interessados em minimizar o erro de serrilhamento de perspectiva máximo, de pior caso, em todo o campo de visão da câmera. Outras métricas tentam garantir a minimização de algum erro médio.

## 4.2. Técnicas de reparametrização perspectiva

Como foi mencionado anteriormente, as técnicas de reparametrização (ou de transformação) do mapa de sombras aplicam uma transformação a toda a cena antes da geração do mapa. A idéia é deformar os objetos de forma que aqueles que necessitem de maior resolução apareçam com um tamanho adequadamente aumentado no mapa de sombras. É importante notar que, nesse processo, a cena deformada ainda é amostrada uniformemente pelo mapa. No entanto, para a cena original, tudo se passa como se a transformação fosse aplicada ao próprio mapa de sombras, tornando os texels menores e mais próximos uns dos outros nas regiões que requerem amostragem mais densa. Por esse motivo, dizemos que o mapa de sombras é *reparametrizado* devido à transformação.

Nesta dissertação, investigamos apenas as técnicas de reparametrização perspectiva, recomendadas para aplicações de tempo real por poderem ser implementadas com muito baixo custo. Esses algoritmos primeiramente determinam um volume piramidal  $\mathbf{P}$  (ou um trapezóide, no caso bidimensional) que envolve todo o campo de visão  $\mathbf{V}$  da câmera e todos os objetos que possam projetar sombras em seu interior. Então, antes de gerar o mapa de sombras, aplicam à cena a transformação perspectiva associada a  $\mathbf{P}$ . Assim, os objetos são deformados de modo que aqueles mais próximos da câmera passem a ocupar a maior área do mapa.

Para entender o processo, considere o caso ilustrado na Figura 17, com uma fonte de luz direcional incidindo perpendicularmente à direção de visualização da câmera. O volume de visão  $\mathbf{V}$  da câmera tem planos *near* e *far* a distâncias  $n$  e  $f$ , respectivamente, e o ângulo  $\phi$  corresponde à metade de seu ângulo de abertura. Como simplificação, consideramos que todos os possíveis oclusores estão no interior de  $\mathbf{V}$ .

Para definirmos o volume  $\mathbf{P}$  da Figura 17, utilizamos o chamado “espaço da luz” (*light space*) definido por Wimmer et al. [18]. O sistema de coordenadas desse espaço é construído da seguinte forma: o eixo  $y$  alinhado com a direção da luz, mas contrário a ela; o eixo  $z$  no plano de  $y$  e da direção de visualização, apontando no mesmo sentido que essa direção; e o eixo  $x$  completando um



O parâmetro  $n'$  é extremamente importante na reparametrização perspectiva, pois controla a força da influência da transformação de  $\mathbf{P}$ . Quando  $n' \rightarrow \infty$ , os lados de  $\mathbf{P}$  tornam-se paralelos e a transformação deixa de ter efeito de perspectiva, de modo que voltamos ao mapeamento de sombras tradicional. Conforme  $n'$  se aproxima de zero, o efeito de perspectiva fica mais forte. A Figura 18 ilustra o efeito, tomando como exemplos as técnicas de PSM (*Perspective Shadow Map*) [19], com  $n' = n$ , e LiSPSM (*Light-Space PSM*) [20], que faz  $n' = n + \sqrt{f n}$ . Em todo o restante desta seção, buscaremos o valor de  $n'$  que minimiza o erro de serrilhamento máximo.

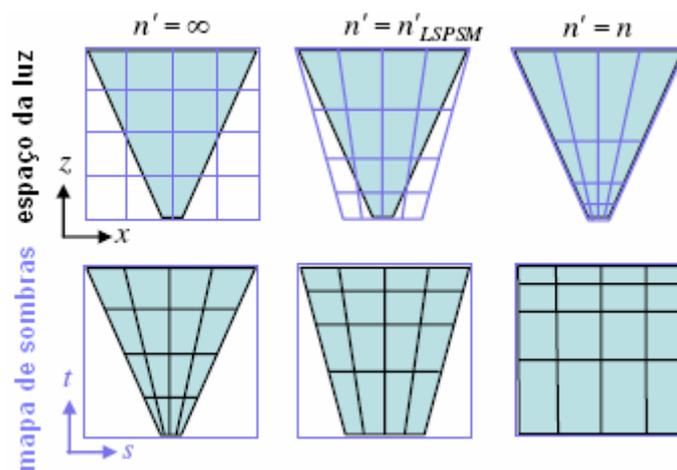


Figura 18 – Efeito da reparametrização perspectiva. Da esquerda para a direita, o parâmetro  $n'$  decresce de  $\infty$  até  $n$ . Acima: na visão da fonte de luz, o mapa de sombras é comprimido e transformado para se ajustar ao volume de visão da câmera. Abaixo: a cena amostrada pelo mapa de sombras é deformada de modo que os objetos mais próximos da câmera ocupem uma maior área do mapa (Figura extraída de [35]).

É interessante notar que a análise a seguir pode ser estendida naturalmente a fontes de luz tipo *spot*, principalmente quando distantes da cena, como poderiam ser utilizadas num modelo CAD. Como esse tipo de fonte se torna direcional após a sua própria transformação perspectiva, basta aplicar  $\mathbf{P}$  nesse espaço transformado [20]. A análise também é válida, com alguma aproximação, para TSMs (*Trapezoidal Shadow Maps*) [21]. Sua transformação, apesar de ser calculada a partir de um trapezóide em 2D, tem o mesmo efeito de perspectiva, igualmente controlável pelo parâmetro  $n'$ .

### 4.2.1. Reparametrização no caso ótimo

Nesta seção, analisamos o serrilhamento e determinamos um parâmetro  $n'$  ótimo para a situação da Figura 17, com a luz incidindo perpendicularmente à direção de visualização. Na Seção 4.2.2, generalizaremos a discussão para outras configurações de luz e câmera.

Inicialmente, computamos o erro de serrilhamento separadamente nas direções  $x$  e  $z$  do espaço da luz. Conforme a Equação (12) e desprezando o serrilhamento de projeção, o erro é dado pela razão entre as espessuras dos feixes da luz e da câmera:

$$m_x = \frac{w_{lx}}{w_{cx}} \quad (14)$$

$$m_z = \frac{w_{lz}}{w_{cy}} \quad (15)$$

A largura do feixe por um único pixel de tamanho  $(d_x \times d_y)$  da imagem pode ser aproximada como a seguir, onde  $n$  é a distância do plano *near*,  $\phi$  a metade do ângulo de visão da câmera e  $(res_{cx} \times res_{cy})$  a resolução da imagem [35].

$$w_{cx} = \frac{d_x z}{n} = \frac{2 z \operatorname{tg}(\phi_x)}{res_{cx}} \quad (16)$$

$$w_{cy} = \frac{d_y z}{n} = \frac{2 z \operatorname{tg}(\phi_y)}{res_{cy}} \quad (17)$$

Num mapa de sombras de resolução  $(res_{ls} \times res_{lt})$  para uma fonte de luz direcional, o tamanho de cada texel nas coordenadas  $(s,t)$  está relacionado a distâncias no espaço do mundo pelas derivadas  $dx/ds$  e  $dz/dt$ , respectivamente, nas direções  $x$  e  $z$ . A largura do feixe por um texel é então dada por [35]:

$$w_{lx} = \frac{1}{res_{ls}} \frac{dx}{ds} \quad (18)$$

$$w_{lz} = \frac{1}{res_{lt}} \frac{dz}{dt} \quad (19)$$

Para calcularmos  $dx/ds$  e  $dz/dt$ , começamos determinando as funções  $s(x',z')$  e  $t(z')$ , que associam coordenadas  $(s,t)$  do mapa de sombras a cada ponto com coordenadas  $x'$  e  $z'$  no espaço da luz. Essas funções podem ser encontradas tomando-se a matriz de projeção perspectiva do OpenGL [48] para  $\mathbf{P}$  (lembrando-se de inverter o sinal de  $z'$ ) e ajustando-se o intervalo de  $[-1,1]$  para  $[0,1]$ .

$$s(x',z') = \frac{1}{2} - \frac{x'}{2(-z')\text{tg}(\phi'_x)} = \frac{1}{2} + \frac{x'}{2z'\text{tg}(\phi_x)} \frac{f'}{f} \quad (20)$$

$$t(z') = \frac{1}{2} + \frac{1}{2} \frac{(f'+n')}{(f'-n')} - \frac{f'n'}{z'(f'-n')} \quad (21)$$

Na verdade, queremos  $s(x,z)$  e  $t(z)$ , ou seja, relações entre as coordenadas do mapa de sombras e as coordenadas  $x$  e  $z$  no espaço da câmera (“espaço do olho” em OpenGL). Com base na Figura 17, podemos fazer as seguintes substituições:

$$\begin{aligned} x' &= x \\ z' &= n' + z - n \\ f' &= n' + f - n \end{aligned} \quad (22)$$

Substituindo a Equação (22) em (20) e (21) e derivando, temos:

$$\frac{ds}{dx} = \frac{1}{2\text{tg}(\phi_x)} \frac{(n'+f-n)}{f(n'+z-n)} \quad (23)$$

$$\frac{dt}{dz} = \frac{(n'+f-n)n'}{(n'+z-n)^2(f-n)} \quad (24)$$

As derivadas  $ds/dx$  e  $dt/dz$  são funções monótonas, de modo que podemos tomar suas recíprocas como  $dx/ds$  e  $dz/dt$  em (18) e (19). Temos então todos os resultados necessários ao cálculo dos erros  $m_x$  e  $m_z$  de serrilhamento em ambas direções, segundo as Equações (14) e (15):

$$m_x(z, n') = \frac{w_{lx}}{w_{cx}} = \frac{res_{cx}}{res_{ls}} f \left( \frac{(n'+z-n)}{z(n'+f-n)} \right) \quad (25)$$

$$m_z(z, n') = \frac{w_{lz}}{w_{cy}} = \frac{res_{cy}}{res_{lt}} \frac{f-n}{2\text{tg}(\phi_y)} \left( \frac{(n'+z-n)^2}{zn'(n'+f-n)} \right) \quad (26)$$

Lloyd et al. [34] identificam três partes distintas nas Equações (25) e (26): a razão entre as resoluções da imagem e do mapa de sombras; um termo que depende apenas da forma do volume de visão da câmera; e um termo que depende da parametrização escolhida, destacado com parênteses nas equações. Segundo nossa métrica de erro, queremos minimizar o valor máximo desse último termo.

Wimmer et al. [20] verificam que, para o erro em  $x$ , a parametrização dos PSMs é ótima, pois a menos dos termos de resolução,  $m_x = 1$  quando  $n' = n$ . Em seu artigo, também mostram que o erro máximo em  $z$  ocorre sempre em um dos planos *near* ou *far* da câmera. Fazendo  $m_z(n, n') = m_z(f, n')$  e resolvendo para  $n'$ , encontram então o parâmetro capaz de minimizar o erro máximo distribuindo-o o mais uniformemente possível dentro do volume de visão:

$$n'_{LiSPSM} = n + \sqrt{fn} \quad (27)$$

A Figura 19 mostra gráficos típicos de  $m_x(z)$  e  $m_z(z)$  para as parametrizações perspectivas com diferentes valores de  $n'$ . No caso dos PSMs, o erro cresce linearmente em  $z$ , atingindo seu maior valor no plano *far*. Por outro lado, no mapeamento tradicional com  $n' \rightarrow \infty$ , o erro decresce hiperbolicamente (em  $1/z$ ) a partir do plano *near*. Com  $n'_{LiSPSM}$ , o maior erro é mínimo e ocorre em ambos planos.

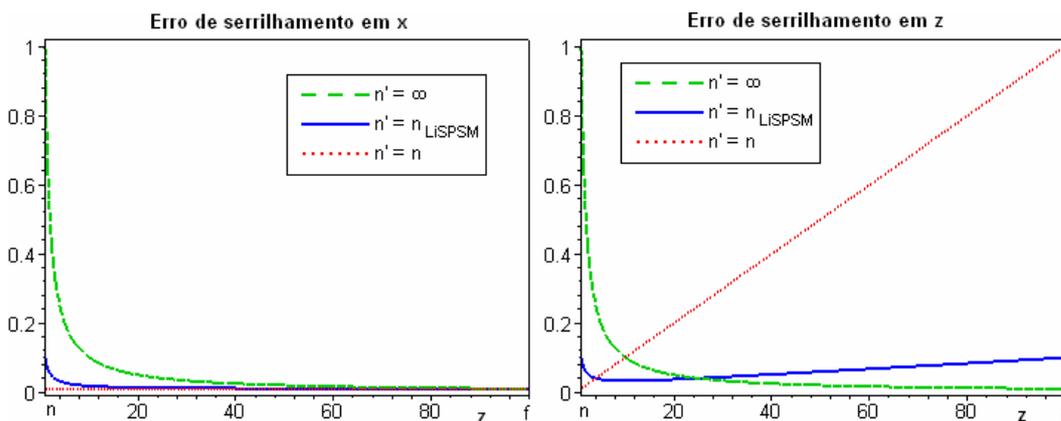


Figura 19 – Erros de serrilhamento de perspectiva nas direções  $x$  e  $z$  ao longo do eixo  $z$  do volume de visão da câmera ( $n = 1$ ,  $f = 100$ ,  $\phi_y = 45^\circ$ ). Os gráficos foram normalizados.

Chamemos de  $M_x$  e  $M_z$  os valores máximos dos erros  $m_x$  e  $m_z$ , respectivamente. Lloyd et al. [34,35] mostram que  $M_x$  ocorre sempre no plano

*near*, enquanto  $M_z$  ocorre no plano *near* ( $z = n$ ) para  $n' > n'_{LiSPSM}$  e no plano *far* ( $z = f$ ) para  $n' < n'_{LiSPSM}$ . Usando esses valores, temos equações para os erros máximos em x e z:

$$M_x(n') = \frac{res_{cx}}{res_{ls}} \frac{f}{n} \frac{n'}{(f - n + n')} \quad (28)$$

$$M_z(n') = \frac{res_{cy}}{res_{lt}} \frac{f - n}{2tg(\phi_y)} \begin{cases} \frac{(n' + f - n)}{f n'}, & n' \leq n'_{LiSPSM} \\ \frac{n'}{n(n' + f - n)}, & n' > n'_{LiSPSM} \end{cases} \quad (29)$$

Lloyd et al. [35] utilizam essas equações para definir uma medida agregada de serrilhamento em x e z, independente das resoluções do mapa de sombras e da imagem. Primeiro, deve-se perceber que o erro desaparece quando  $M_x = M_z = 1$ . Assim, podemos eliminar o serrilhamento utilizando um mapa de sombras de resolução suficientemente alta, chamada *resolução crítica* ( $res_{ls}^* \times res_{lt}^*$ ). Define-se então o *fator de armazenamento* S como a razão entre o número de texels necessários para atingir a resolução crítica e o número de pixels na imagem:

$$S = \frac{res_{ls}^* res_{lt}^*}{res_{cx} res_{cy}} \quad (30)$$

O fator de armazenamento indica quantas vezes maior que a imagem um mapa de sombras deve ser para eliminar o serrilhamento de perspectiva. Ele pode ser calculado multiplicando-se diretamente  $M_x$  e  $M_z$  com os termos de resolução iguais a 1. Temos assim:

$$S = \frac{(f/n - 1)}{2tg(\phi_y)} \begin{cases} 1, & n' \leq n'_{LiSPSM} \\ \frac{f n'^2}{n(n' + f - n)^2}, & n' > n'_{LiSPSM} \end{cases} \quad (31)$$

Com o parâmetro  $n'$  variando no intervalo  $[n, \infty)$ , a análise de  $M_x$ ,  $M_z$  e S torna-se bastante inconveniente. Por esse motivo, Lloyd et al. introduzem ainda um novo parâmetro  $\eta \in (-1, 1]$  no lugar de  $n'$ . O mapeamento faz a correspondência de  $\eta \in \{-1, 0, 1\}$  com  $n' \in \{\infty, n'_{LiSPSM}, n\}$ , respectivamente. O máximo  $M_z$  encontra-se no plano *near* com  $\eta < 0$  (e  $n' > n'_{LiSPSM}$ ) e no plano *far* com  $\eta \geq 0$  ( $n \leq n' \leq n'_{LiSPSM}$ ):

$$\eta = \begin{cases} \frac{M_z(n') - M_z(n'_{LiSPSM})}{M_z(n) - M_z(n'_{LiSPSM})}, & n \leq n' \leq n'_{LiSPSM} \\ \frac{M_z(n'_{LiSPSM}) - M_z(n')}{M_z(\infty) - M_z(n'_{LiSPSM})}, & n' > n'_{LiSPSM} \end{cases} \quad (32)$$

Resolvendo a Equação (32) para  $n'$ , temos:

$$n' = \begin{cases} \frac{n + \sqrt{fn} - \eta(f - n)}{\eta + 1}, & -1 < \eta < 0 \\ \frac{n + \sqrt{fn}}{n + \eta\sqrt{fn}}, & 0 \leq \eta \leq 1 \end{cases} \quad (33)$$

A Figura 20 mostra os gráficos de  $M_x$ ,  $M_z$  e  $S$  em função do parâmetro  $\eta$ . As curvas corroboram o fato de que os menores erros máximos em  $x$  e  $z$  ocorrem, respectivamente, em  $n' = n$  e em  $n' = n'_{LiSPSM}$ . No entanto, como a própria Equação (31) indica, podemos ver que o Fator de Armazenamento é mínimo e constante para qualquer  $n' \leq n'_{LiSPSM}$ . Assim, considerando-se o erro máximo de serrilhamento de perspectiva, a escolha de  $n'$  nesse intervalo influencia apenas o local onde esse máximo ocorre e as dimensões relativas de um mapa com resolução crítica [35]. Neste trabalho, assim como em [35], recomendamos a utilização o parâmetro  $n'_{LiSPSM}$  por sua garantia de  $M_z$  mínimo, melhor distribuição do erro em  $x$  e  $z$ , e igualdade de  $M_z$  nos planos *near* e *far*.

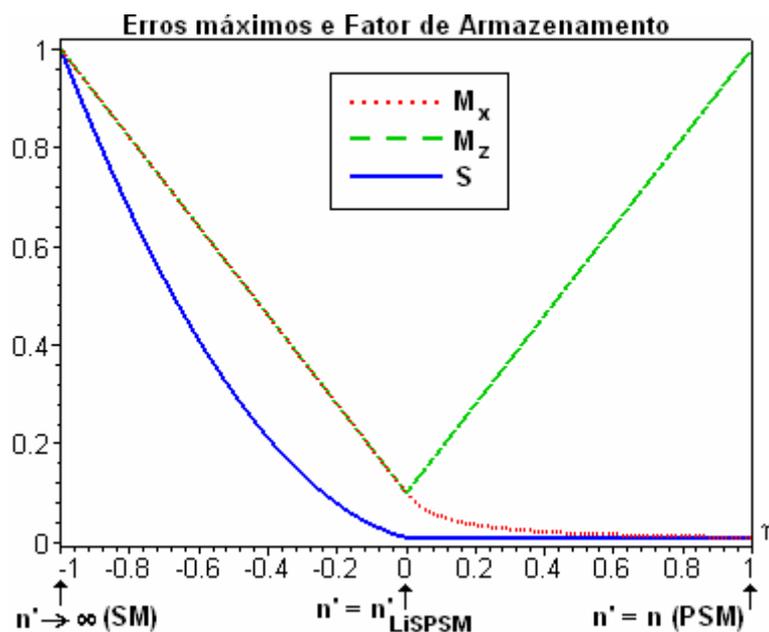


Figura 20 – Erros máximos nas direções  $x$  e  $z$  e Fator de Armazenamento. Os gráficos foram normalizados para escalas compatíveis na ordenada ( $n = 1$ ,  $f = 100$ ,  $\phi_y = 45^\circ$ ).

#### 4.2.2. Reparametrização generalizada

Até agora, foi discutido apenas o caso de uma fonte de luz direcional incidindo perpendicularmente à direção de visualização da câmera. Conforme as direções da luz e de visualização se aproximam da situação paralela, ocorre uma redução do intervalo de influência (*depth range*) afetado pela transformação perspectiva [20]. Isso quer dizer que, para uma mesma variação na coordenada  $z$  de pontos no espaço da câmera, a variação correspondente em  $z'$  no espaço da luz e em  $t$  no mapa de sombras se torna cada vez menor. Na situação limite de luz e câmera paralelas, todos os pontos ao longo do eixo  $z$  no centro do volume de visão da câmera são mapeados para o mesmo texel do mapa de sombras, independente da parametrização utilizada.

Na verdade, quando a luz incide paralelamente à direção de visualização, o serrilhamento de perspectiva se torna dependente apenas da forma do volume de visão da câmera e da diferença de resolução entre a imagem e o mapa de sombras [20,35] (veja na Seção 4.3.2 o cálculo do Fator de Armazenamento para esta situação). Assim, nenhuma reparametrização perspectiva é capaz de tratar o problema. Por esse motivo, qualquer algoritmo desse tipo precisa convergir para o mapeamento tradicional nesse caso, fazendo  $n' \rightarrow \infty$  de forma controlada.

No algoritmo de LiSPSM proposto por Wimmer et al. [20], o parâmetro  $n'_{LiSPSM}$  é dividido pelo seno do ângulo entre as direções da luz e de visualização. Isso faz o mapeamento começar a convergir para o caso não-reparametrizado assim que a luz e a câmera começam a se afastar da configuração ótima. Nesta seção, por outro lado, baseamo-nos no raciocínio de Zhang et al. [27] para derivarmos um parâmetro  $n'_{LiSPSM}$  generalizado, cujo objetivo é manter válida a importante propriedade de erro de serrilhamento mínimo do LiSPSM, mesmo para casos em que a luz não incide perpendicularmente à direção de visualização. Veremos, porém, que a partir de certo ponto ainda será necessário convergir para o mapeamento tradicional.

No desenvolvimento que segue, consideraremos apenas o erro de serrilhamento em  $z$ , utilizado na determinação do parâmetro  $n'_{LiSPSM}$ . A análise também é restrita apenas a pontos sobre o eixo  $z$  no centro do volume de visão da câmera.

Observando cuidadosamente a Figura 21, podemos obter as formas generalizadas para a Equação (22) como a seguir.

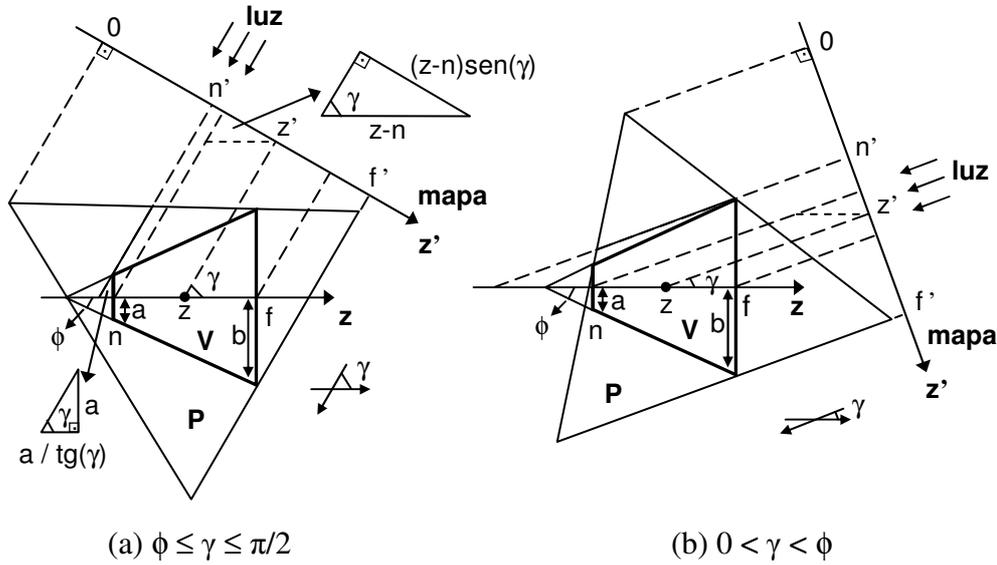


Figura 21 – Reparametrização perspectiva generalizada do mapa de sombras.

$$x' = x \quad (34)$$

$$z' = \begin{cases} n' + a \cos(\gamma) + (z - n) \sin(\gamma), & \phi_y \leq \gamma \leq \pi/2 \\ n' + b \cos(\gamma) + (z - f) \sin(\gamma), & 0 < \gamma < \phi_y \end{cases} \quad (35)$$

$$f' = \begin{cases} n' + (a + b) \cos(\gamma) + (f - n) \sin(\gamma), & \phi_y \leq \gamma \leq \pi/2 \\ n' + 2b \cos(\gamma) & 0 < \gamma < \phi_y \end{cases} \quad (36)$$

Nas Equações (35) e (36),  $a$  e  $b$  correspondem à metade das medidas dos planos *near* e *far* da câmera ( $a = n \operatorname{tg}(\phi_y)$ ,  $b = f \operatorname{tg}(\phi_y)$ ) e  $\gamma$  é o menor ângulo entre a direção da fonte de luz e a direção de visualização, considerado sempre na posição padrão  $\gamma \in [0, \pi/2]$ . Note que essas expressões convergem para (22) quando  $\gamma \rightarrow \pi/2$ . Substituindo (35) em (21) e seguindo o mesmo procedimento descrito na seção anterior, chegamos à forma generalizada da Equação (26), o erro de serrilhamento em  $z$ :

$$m_z(z, n', \gamma) = \frac{w_{lz}}{w_{cy}} = \frac{\operatorname{res}_{cy}}{\operatorname{res}_{lt}} \frac{1}{2 \operatorname{tg}(\phi_y)} \frac{(f' - n')}{n' f'} \frac{(\psi(z, n', \gamma))^2}{z \sin(\gamma)} \quad (37)$$

$$\psi(z, n', \gamma) = \begin{cases} n' + a \cos(\gamma) + (z - n) \sin(\gamma), & \phi_y \leq \gamma \leq \pi/2 \\ n' + b \cos(\gamma) + (z - f) \sin(\gamma), & 0 < \gamma < \phi_y \end{cases}$$

Queremos saber agora se  $m_z$  se comporta da mesma maneira que antes para qualquer valor de  $\gamma$ , ou seja, se o erro é máximo sempre nos extremos  $z = n$  ou  $z = f$ . Diferenciando em relação a  $z$ , encontramos um único extremo local nesse intervalo:

$$\frac{\partial m_z}{\partial z} = 0 \Rightarrow z = \frac{1}{\sin(\gamma)} \begin{cases} n' + a \cos(\gamma) - n \sin(\gamma), & \phi_y \leq \gamma \leq \pi/2 \\ n' + b \cos(\gamma) - f \sin(\gamma), & 0 < \gamma < \phi_y \end{cases} \quad (38)$$

A segunda derivada é positiva em todo o intervalo  $z \in [n, f]$ , de modo que a Equação (38) corresponde a um ponto de mínimo. Assim, o erro máximo encontra-se sempre num dos planos *near* ou *far* da câmera. Para minimizá-lo, basta igualar seu valor nesses dois extremos, analogamente ao trabalho de Wimmer et al. [20]:

$$\begin{aligned} m_x(n, n', \gamma) &= m_x(f, n', \gamma) \\ \Rightarrow n'_{LISPSM} &= \begin{cases} n'_1(\gamma) = (n + \sqrt{fn}) \sin(\gamma) - a \cos(\gamma), & \phi_y \leq \gamma \leq \pi/2 \\ n'_2(\gamma) = (f + \sqrt{fn}) \sin(\gamma) - b \cos(\gamma), & 0 < \gamma < \phi_y \end{cases} \end{aligned} \quad (39)$$

A Figura 22a mostra um gráfico típico de  $n'_{LISPSM}$  generalizado. Como se pode ver, à medida que as direções da luz e da câmera se alinham, é preciso aumentar a força da transformação perspectiva, numa tentativa de compensar a redução do intervalo de influência. Quando  $\gamma \leq \phi$ , porém, verifica-se a necessidade de tornar  $n'_2$  negativo, o que é impossível pela modelagem do problema. Isso está de acordo a afirmação anterior de que técnicas de reparametrização perspectiva são incapazes de lidar com essa situação.

Na verdade, os valores negativos de  $n'_2$  não são os únicos que não devem ser utilizados da Equação (41). Quando  $\gamma \leq \phi$ , por exemplo, o plano *near* da câmera aparece inteiramente dentro do plano *far* na visão da fonte de luz. Nessa situação, como veremos na Seção 4.3.1, a qualidade das sombras pode piorar com a utilização de uma reparametrização perspectiva muito forte. É necessário então escolher um valor de  $\gamma$  limite, que chamaremos de  $\gamma_{lim}$ . Quando o ângulo entre a luz e a direção de visualização for menor que esse limite, deve ser utilizado um parâmetro  $n'_2$  modificado, de modo que a parametrização seja forçada a convergir para o mapeamento tradicional.

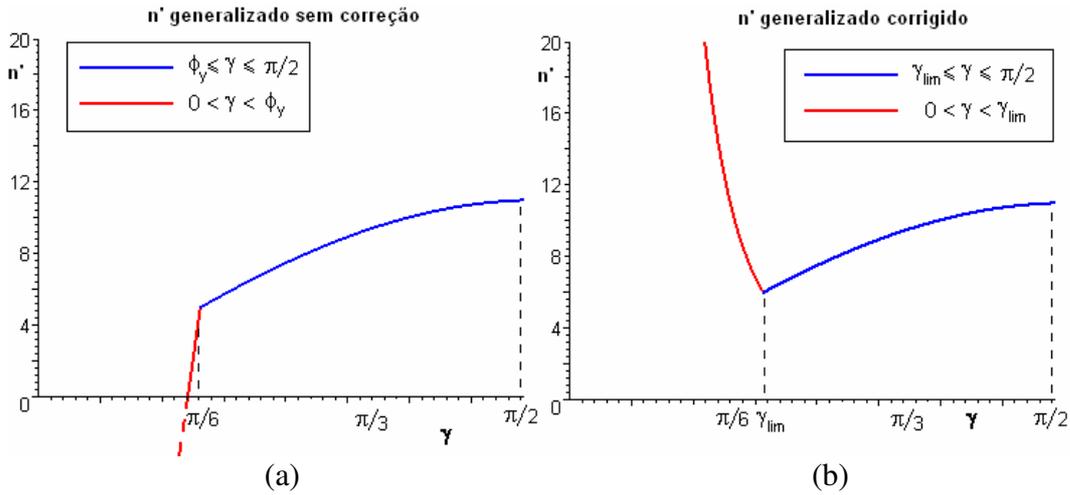


Figura 22 – Parâmetro  $n'_{\text{LISPSM}}$  generalizado (a) antes e (b) após a correção do caso limite ( $n = 1$ ,  $f = 100$ ,  $\phi_y = 30^\circ$ ).

Ao estabelecermos  $\gamma_{\text{lim}}$ , nossa intenção é evitar que o volume  $\mathbf{P}$  tenha um ângulo de abertura muito elevado, o que deixaria a transformação perspectiva exageradamente forte. Recomendamos aqui a utilização do seguinte limite:

$$\gamma_{\text{lim}} = \text{arctg} \left( \frac{a+b}{f-n} \right) \quad (40)$$

Pode-se mostrar que a Equação (40) corresponde ao maior valor de  $\gamma$  para o qual, na visão da fonte de luz, os planos *near* e *far* da câmera começam a se sobrepor. Verificamos em nosso estudo que, a partir desse ponto, a abertura dos volumes  $\mathbf{P}$  gerados tenderiam a aumentar rapidamente com a utilização da Equação (39).

Para obtermos uma forma aplicável de  $n'_2$  quando  $\gamma < \gamma_{\text{lim}}$ , procedemos como Zhang et al. [27]. Inicialmente, lembramos que, à medida que  $\gamma \rightarrow 0$ , a parametrização deve convergir suavemente para o mapeamento de sombras tradicional. Além disso, queremos que  $n'$  seja contínuo em  $\gamma = \gamma_{\text{lim}}$  para evitar mudanças muito bruscas na qualidade das sombras. Assim:

$$\lim_{\gamma \rightarrow 0} n'_2 = \infty \quad (41)$$

$$n'_2(\gamma_{\text{lim}}) = n'_1(\gamma_{\text{lim}})$$

As restrições impostas pela Equação (41) podem ser satisfeitas como a seguir, onde  $r(\gamma)$  é uma função suave que decresce de  $r(\gamma_{\text{lim}}) = 1$  até  $r(0) = 0$ .

$$n'_{LiSPSM} = \begin{cases} n'_1(\gamma) = (n + \sqrt{fn}) \sin(\gamma) - a \cos(\gamma), & \gamma_{lim} \leq \gamma \leq \pi/2 \\ n'_2(\gamma) = \frac{n'_1(\gamma_{lim})}{r(\gamma)}, & 0 < \gamma < \gamma_{lim} \end{cases} \quad (42)$$

A expressão de  $n'_2$  na Equação (42) foi decidida arbitrariamente, simplesmente de forma a garantir as restrições da Equação (41). A função  $r(\gamma)$  é escolhida de modo igualmente arbitrário. Em seu trabalho, Zhang et al. [27] utilizam a seguinte função:

$$r(\gamma) = \sin\left(\frac{\gamma}{\gamma_{lim}} \frac{\pi}{2}\right) \quad (43)$$

Em nosso estudo, verificamos a necessidade de se fazer o algoritmo convergir para o mapeamento tradicional de forma mais rápida do que impõe a Equação (43). Utilizamos então a seguinte função, cuja variação é mais rápida para ângulos próximos a  $\gamma_{lim}$ :

$$r(\gamma) = \left[ 1 + \sin\left(\left(\frac{\gamma}{\gamma_{lim}} - 1\right) \frac{\pi}{2}\right) \right]^2 \quad (44)$$

A Figura 22b mostra um gráfico do parâmetro  $n'_{LiSPSM}$  generalizado proposto, que utiliza as Equações (42) e (44). O erro de serrilhamento resultante na direção  $z$  é ilustrado na Figura 23 para  $\gamma_{lim} \leq \gamma \leq \pi/2$ . Note que o erro é máximo e idêntico para  $z = n$  e  $z = f$ .

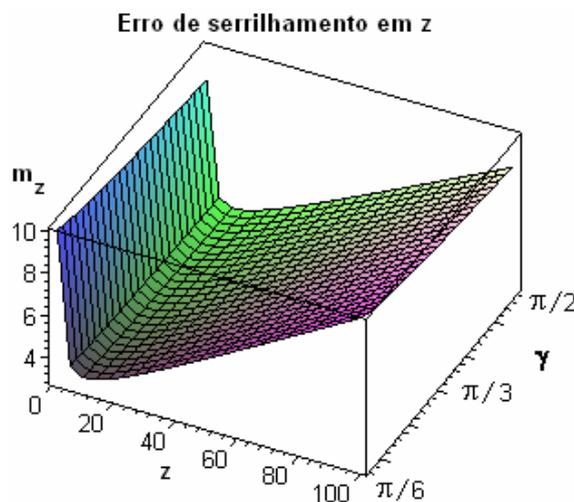


Figura 23 – Erro de serrilhamento para diferentes valores de  $\gamma$  ( $n = 1$ ,  $f = 100$ ,  $\phi_y = 30^\circ$ ).

### 4.3. Técnicas de particionamento

Vimos, no Capítulo 2, que existem diversas maneiras de se particionar um mapa de sombras visando o melhor aproveitamento de sua resolução. Nesta seção, analisamos as técnicas mais diretamente relacionadas com as reparametrizações perspectivas: o particionamento segundo as faces [34,35] e segundo a profundidade (coordenada  $z$ ) [33-35] do volume de visão da câmera.

O desenvolvimento a seguir acompanha o trabalho de Lloyd et al. [34,35]. Adicionalmente, na Seção 4.3.2, propomos um esquema adaptativo de particionamento em profundidade para minimizar o efeito de “costuras” (*seams*) associado à técnica.

#### 4.3.1. Particionamento por faces

Vimos, na Seção 4.2.2, que qualquer parametrização perspectiva do mapa de sombras precisa convergir para o mapeamento tradicional quando a fonte de luz incide quase paralelamente à direção de visualização da câmera. Para entendermos melhor esse caso, consideremos a Figura 24a. Idealmente, queremos que a largura do feixe de cada texel do mapa de sombras seja igual à do feixe de pixel mais estreito que ele intercepta ao atravessar o volume de visão da câmera. Assim, na região A, os texels devem diminuir da coordenada 0 até  $t_1$ . Na região B, os feixes mais estreitos têm aproximadamente o mesmo tamanho e estão junto ao plano *near*, de modo que os texels devem se manter também com espessura constante. Finalmente, na região C, devemos ter novamente texels cada vez maiores da coordenada  $t_2$  até 1.

Quando aplicamos uma transformação perspectiva à cena que será amostrada pelo mapa de sombras, os texels adquirem tamanho sempre crescente ao longo do eixo do volume  $\mathbf{P}$  escolhido. Isso quer dizer que, independente da transformação, sempre haverá uma região (A ou C) em que o tamanho dos texels varia no sentido inverso ao que deveria, causando uma piora na qualidade das sombras. Daí surge a necessidade de convergir qualquer parametrização de volta para o mapeamento tradicional quando a fonte de luz está logo à frente ou atrás da câmera. A solução proposta por alguns autores e formalizada por Lloyd et al. [35]

consiste em gerar um mapa diferente, com uma transformação independente, para cada face do volume de visão da câmera conforme visto pelo ponto-de-vista da fonte (Figura 24b).

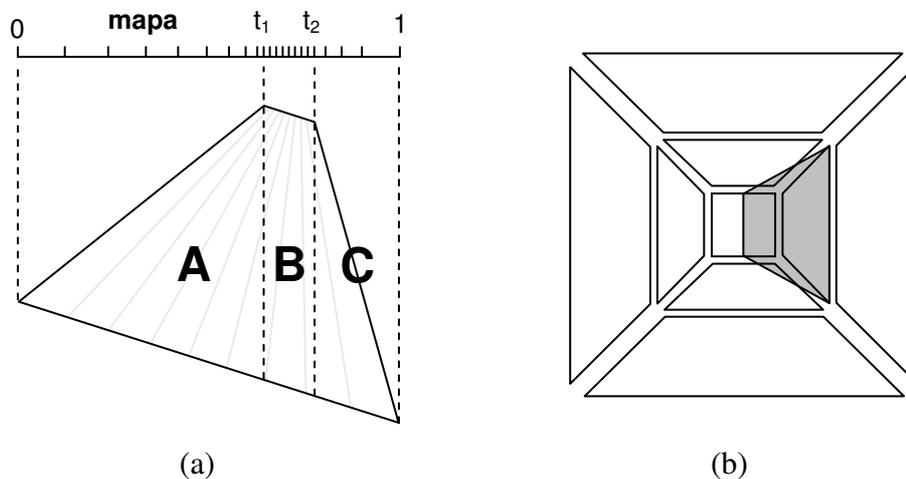


Figura 24 – Particionamento segundo as faces do volume de visão da câmera. (a) Espaçamento não monótono dos texels do mapa de sombras; (b) volume de visão da câmera explodido de acordo com o particionamento em faces e em  $z$ , com destaque para uma partição alongada de forma a incluir o plano *near*.

Apesar de garantir a eficácia da reparametrização perspectiva em qualquer caso, reduzindo drasticamente o serrilhamento quando a luz incide quase paralelamente à direção de visualização, o particionamento de faces tem uma séria desvantagem: ele requer até quatro passadas de renderização para se gerar o mapa de sombras. Para evitar uma possível quinta passada, as faces laterais do volume de visão podem ser consideradas alongadas, como mostra a Figura 24b, de modo a cobrirem o plano *near*. Além disso, o custo extra de geometria pode ser bastante reduzido com a utilização de uma técnica de descarte em relação ao volume de visão (*frustum culling*) ajustado a cada partição. Apesar disso, Lloyd et al. [35] mostram que, no caso geral, o particionamento apenas em  $z$  (Seção 4.3.2) tem um melhor custo-benefício com relação ao Fator de Armazenamento.

#### 4.3.2. Particionamento em $z$

Antes de estudarmos o particionamento em  $z$  (ou particionamento em profundidade), é importante entendermos a motivação por trás da técnica. Para

isso, precisamos relembrar a situação ótima para parametrizações perspectivas, ilustrada na Figura 17 da Seção 4.2.1, com uma fonte de luz incidindo perpendicularmente à direção de visualização da câmera. Como vimos, o erro de serrilhamento na direção  $z$  pode ser calculado pelas Equações (15), (17) e (19). Combinando essas equações, temos a expressão geral a seguir:

$$m_z = \frac{w_{lz}}{w_{cy}} = \frac{res_{cy}}{res_{lt}} \frac{1}{2 \operatorname{tg}(\phi_y)} \frac{1}{z} \frac{dz}{dt} \quad (45)$$

Queremos descobrir qual seria a parametrização ideal para um mapa de sombras, ou seja, aquela que garantiria  $m_z = 1$  a menos dos termos de resolução e das constantes que dependem apenas da forma do volume de visão da câmera. Assim:

$$\frac{1}{z} \frac{dz}{dt} = 1 \Rightarrow \frac{dz}{z} = dt \quad (46)$$

A parametrização ideal para a coordenada  $t$  do mapa de sombras (associada ao erro em  $z$ ) pode ser encontrada agora integrando-se  $dt$  na Equação (46):

$$\bar{t} = \int_0^t dt = \int_n^z \frac{dz}{z} = \ln\left(\frac{z}{n}\right) \quad (47)$$

Como as coordenadas do mapa de sombras estão restritas ao intervalo  $[0,1]$ , precisamos ainda normalizar o resultado:

$$t = \frac{\ln(z/n)}{\ln(f/n)} \quad (48)$$

O erro em  $z$ , para essa parametrização normalizada, é dado por:

$$m_z = \frac{res_{cy}}{res_{lt}} \frac{\ln(f/n)}{2 \operatorname{tg}(\phi_y)} \quad (49)$$

Note que, como desejado, o erro  $m_z$  é constante, de modo que a parametrização logarítmica expressa na Equação (48) é ótima.

Infelizmente, nenhuma parametrização perspectiva é capaz de produzir a variação logarítmica ideal na coordenada  $t$  do mapa de sombras. Para perceber isso, note que, pela Equação (46), o espaçamento  $dz/dt$  entre os texels do mapa

deve ser linear, ou seja, a largura dos texels transformados deve ser proporcional a  $z$ . Por outro lado, a Equação (24) da Seção 4.2.1 mostra que, numa parametrização perspectiva, o espaçamento  $dz/dt$  será sempre proporcional a  $(z + c_0)^2$ , onde  $c_0$  é uma constante. Isso quer dizer que a largura dos texels cresce de forma quadrática, e não linear.

Na verdade, a parametrização logarítmica ideal (LogSM) não pode ser implementada em hardware gráfico atual, já que a rasterização de triângulos é feita com interpolação linear em perspectiva. Lloyd et al. [26] estão correntemente investigando as extensões de hardware necessárias.

Apesar da impossibilidade de se conseguir a parametrização logarítmica de forma eficiente, podemos ainda tentar obter uma aproximação por partes. Essa é a idéia da técnica de particionamento em profundidade: primeiro, subdividimos o volume de visão da câmera ao longo do eixo  $z$ , conforme Figura 25a; então, obtemos um mapa de sombras independente para cada parte. Se utilizarmos parametrizações perspectivas, a aproximação da função linear que define o tamanho ideal dos texels do mapa se dá em trechos quadráticos; caso contrário, a aproximação é feita por trechos de texels com tamanho constante.

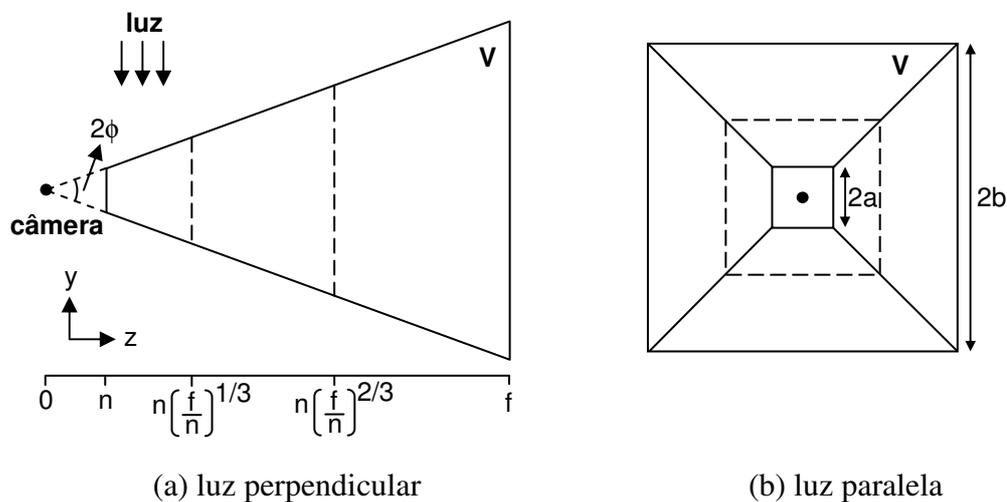


Figura 25 – Particionamento em  $z$  com partições semelhantes entre si.

Para decidirmos os pontos em que o particionamento deve ser efetuado, utilizamos o Fator de Armazenamento definido na Seção 4.2.1. Segundo a Equação (31), o Fator de Armazenamento  $S$  de um mapa de sombras aumenta com a razão  $(f/n)$ . Assim, para minimizar o valor máximo de  $S$  entre todas as

partições, devemos ter o mesmo valor de  $(f/n)$  em cada parte, ou seja, as partições devem ser semelhantes umas às outras. Portanto, para  $k$  partições, temos:

$$n_i = n \left( \frac{f}{n} \right)^{(i-1)/k}, \quad f_i = n_{i+1} = n \left( \frac{f}{n} \right)^{i/k}, \quad i \in \{1, 2, \dots, k\} \quad (50)$$

Chamemos de  $ZP_k$  um esquema de particionamento em profundidade com  $k$  partições ( $ZP_1$  indica a ausência desse tipo de particionamento). O Fator de Armazenamento para um esquema  $ZP_k$  é dado por:

$$S_{ZP_k} = k \frac{(f/n)^{1/k} - 1}{2 \operatorname{tg}(\phi_y)} \begin{cases} 1, & n' \leq n'_{\text{LISPSM}} \\ (f/n)^{1/k}, & n' \rightarrow \infty \end{cases} \quad (51)$$

Como base de comparação, o Fator de Armazenamento da parametrização logarítmica (48) é dado por:

$$S_{\text{LogSM}} = \frac{\ln(f/n)}{2 \operatorname{tg}(\phi_y)} \quad (52)$$

A Figura 26 plota os gráficos de  $S_{\text{LogSM}}$  e  $S_{ZP_k}$ , em função de  $(f/n)$ , com diferentes valores de  $k$ . Como se pode ver, o ganho já é significativo com apenas 2 ou 4 partições e o uso de  $n'_{\text{LISPSM}}$ , o que evita a necessidade de um grande número de mapas de sombra e renderizações da cena.

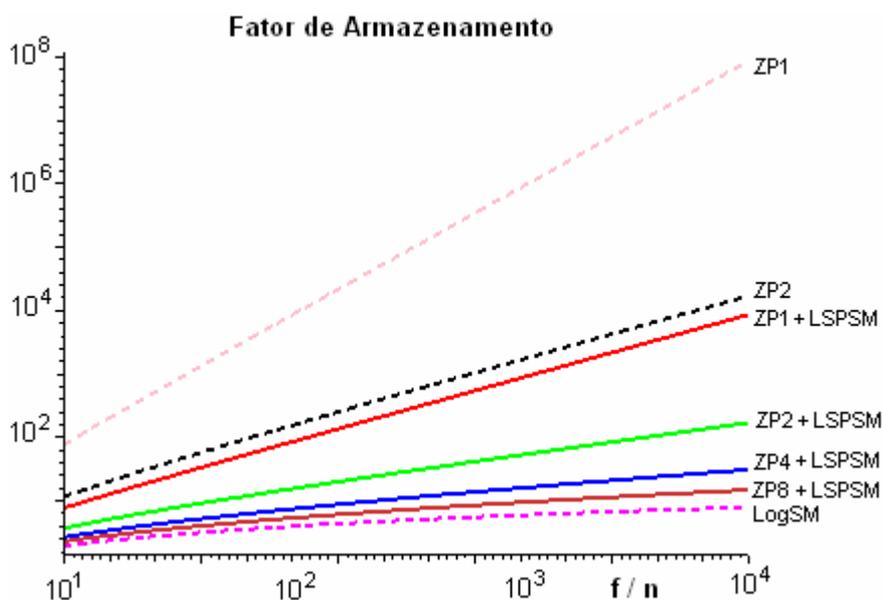


Figura 26 – Fator de Armazenamento para particionamento em  $z$ .

É importante notar que, se os erros  $m_x$  e  $m_z$  sofrerem uma alteração brusca na interface entre duas partições, a variação na qualidade das sombras pode ser facilmente perceptível na visualização, aparecendo como um efeito de “costura” (*seam*) desagradável (Figura 27). Assim, o uso do parâmetro otimizado  $n'_{LiSPSM}$  torna-se especialmente útil: com ele os erros nos planos *near* e *far* na direção  $z$  são iguais entre duas partições, e na direção  $x$  têm diferença pequena.

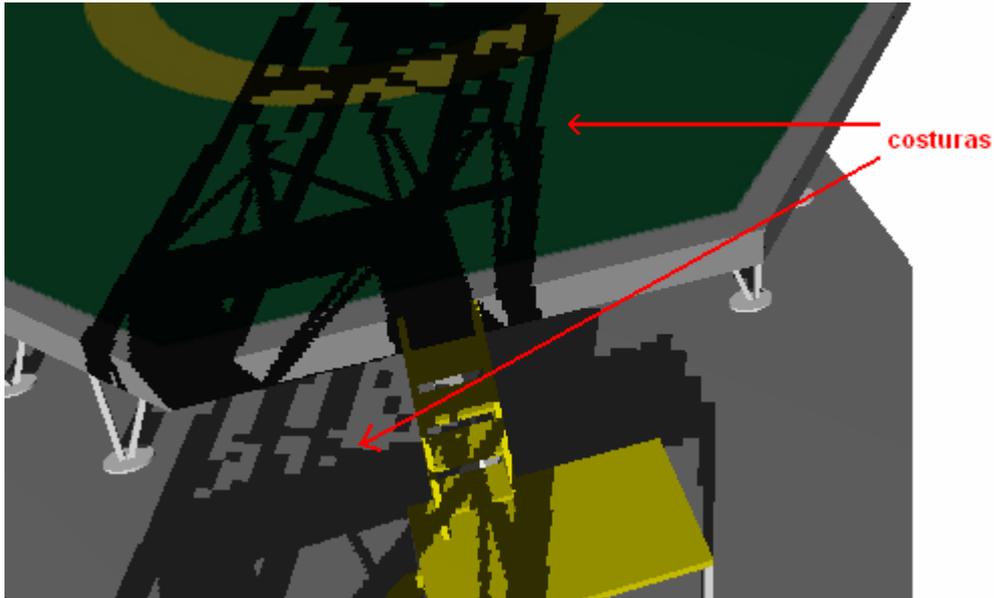


Figura 27 – Efeito de “costura” devido à descontinuidade do erro de serrilhamento.

O efeito de costura torna-se um problema sério quando as direções da luz e da câmera são paralelas, como na Figura 25b. Nesse caso, um mapa de sombras de resolução crítica deve ter texels com o mesmo espaçamento que os pixels da imagem [35]. O Fator de Armazenamento é dado então pela razão entre as áreas cobertas pelo mapa e pelo plano *near*:

$$S_{paralelo} = \frac{(2b)^2}{(2a)^2} = \left(\frac{b}{a}\right)^2 = \left(\frac{f}{n}\right)^2 \quad (53)$$

Se tomarmos  $k$  partições,  $S$  é mínimo também quando são todas semelhantes entre si:

$$S_{ZPk-paralelo} = k \left(\frac{f}{n}\right)^{2/k} \quad (54)$$

Infelizmente, no entanto, o erro máximo ocorre sempre no plano *near* de cada partição em ambas direções. Dessa forma, a interface entre o plano *far* de uma partição e o *near* da seguinte vai sempre apresentar um forte efeito de costura (a Figura 27 foi obtida dessa forma). Para minimizarmos o artefato, somos obrigados a manter a razão ( $f_i / n_i$ ) pequena em todas as partições, exceto a mais próxima da câmera. Recomenda-se, empiricamente, um valor entre 2 e 4.

Com base nas observações precedentes, propomos aqui um esquema dinâmico de particionamento em profundidade. Nosso objetivo é obtermos o particionamento ótimo na situação ideal de luz perpendicular à direção de visualização, ao mesmo tempo em que evitamos ao máximo o efeito de costura na situação limite de luz paralela. Isso pode ser feito com uma interpolação linear entre os dois pontos de particionamento, como segue:

$$\lambda = \frac{\gamma}{\pi/2}, \quad 0 \leq \lambda \leq 1$$

$$\begin{cases} f_k = f \\ n_i = f_{i-1} = \lambda \left[ n \left( \frac{f_i}{n} \right)^{\frac{i-1}{i}} \right] + (1-\lambda) \frac{f_i}{\alpha_i}, \quad i \in \{k, \dots, 3, 2\} \\ n_1 = n \end{cases} \quad (55)$$

Na Equação (55), chamamos de  $\alpha_i$  a razão ( $f_i / n_i$ ) desejada em cada partição para se evitar o efeito de costura no caso limite. O peso da interpolação linear é dado por  $\lambda$ , que indica o quão próximo de zero ou de  $\pi/2$  está o ângulo  $\gamma$  entre a luz e a direção de visualização. O processo deve ser feito de forma iterativa, começando pela última partição ( $i = k$ ) e retrocedendo até a segunda ( $i = 2$ ), sempre levando em conta os valores de  $f_i$  já computados com a interpolação. A primeira partição não precisa ser determinada, pois reside sempre entre  $n_1 = n$  e  $f_1 = n_2$ . No Apêndice B, provamos que a forma iterativa da Equação (55) é equivalente à Equação (50) quando  $\lambda = 1$ .