LEI / LCC
Departamento de Informática
Universidade do Minho

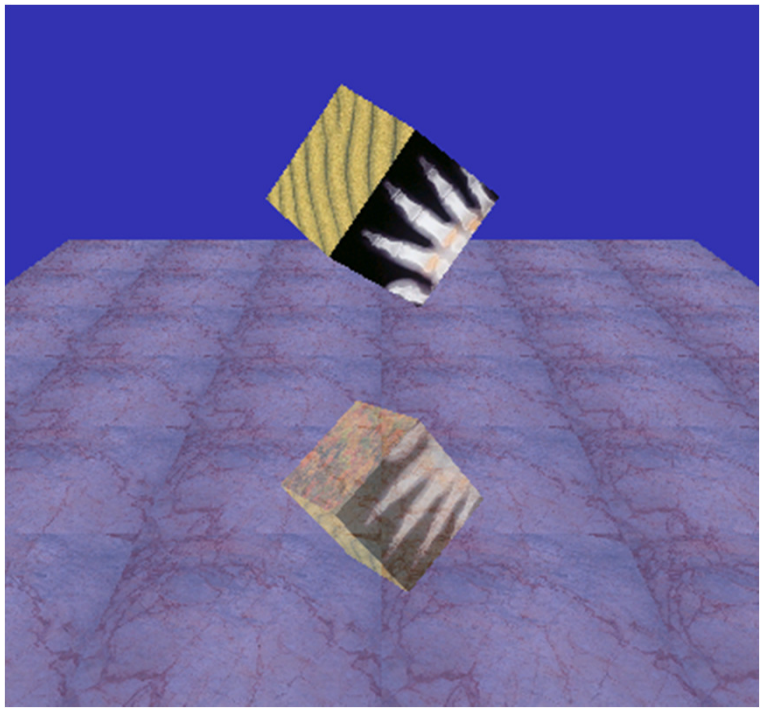
Reflexões, Sombras e Projective Texturing

Shadow Volumes e Shadow Maps

Sumário

- **Reflexões**
- Projective Texturing
- Sombras - Introdução
- Sombras no Plano
- Shadow Volumes
- Shadow Maps

Reflexões



Reflexões

- Estratégia:

```
DrawCube();

glPushMatrix();

glScaled(1,-1,1);

glFrontFace(GL_CW);

DrawCube();

glFrontFace(GL_CCW);

glPopMatrix();
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glColor4f(1.0f, 1.0f, 1.0f, 0.6f);

drawPlane();
glDisable(GL_BLEND);
```

Desenhar cubo original com transformação a partir do plano

Inverter a escala relativamente ao plano

Inverter ordem das faces

Desenhar novamente o cubo

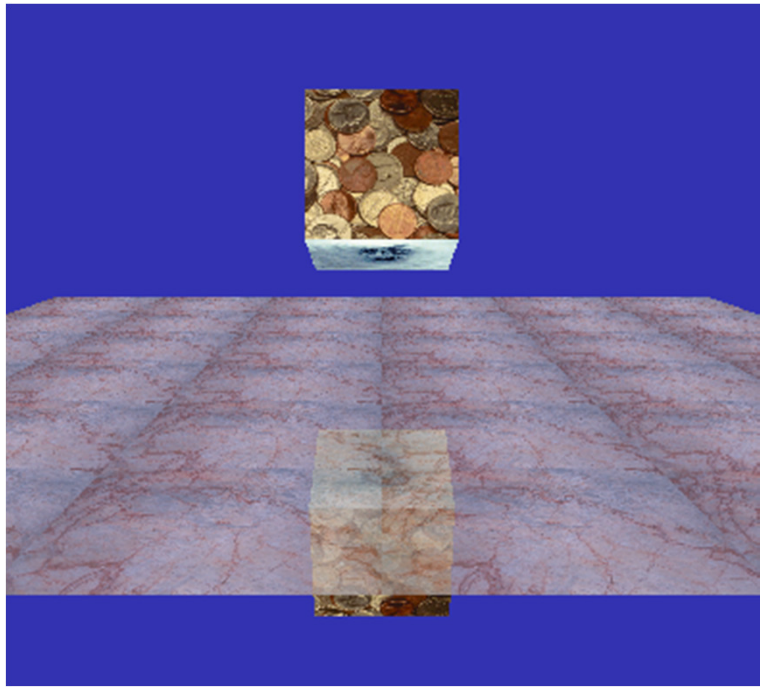
Definição de transparência

Desenhar plano por cima do cubo reflectido

Reflexões

- Ooops!

- Como limitar a área de desenho do cubo reflectido à área ocupada pelo plano?



Reflexões

- Stencil Buffer
  - Buffer de pixels que permite limitar a área de desenho, funcionando como uma máscara.
  - Tal como o Z-Buffer, os pixels são testados individualmente.
  - Este teste ocorre antes do teste do Z-Buffer

Reflexões

- Stencil Buffer – Operações
  - Criar um app com Stencil Buffer em GLUT
    - `glutInitDisplayMode(GLUT_STENCIL ...);`

Reflexões

- Stencil Buffer – Operações
  - Activar/Desactivar
    - `glEnable( GL_STENCIL_TEST);`
    - `glDisable(GL_STENCIL_TEST);`
  - Limpar
    - `glClear(GL_STENCIL_BUFFER_BIT);`

Reflexões

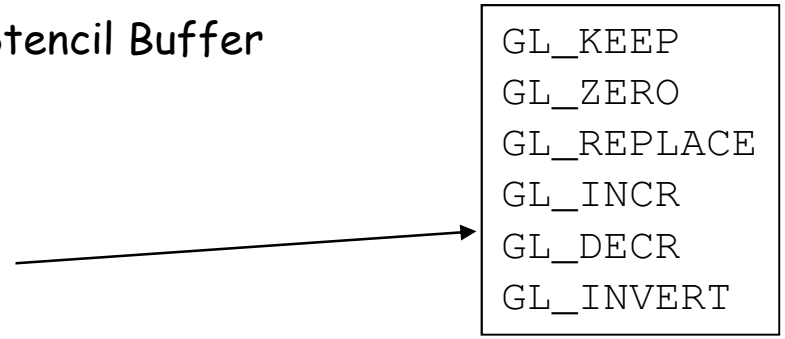
- Stencil Buffer – Operações
  - Definir o teste
    - `glStencilFunc(`
      `GLenum func,`
      `GLint ref,`
      `GLuint mask);`

```
GL_NEVER
GL_ALWAYS
GL_LESS
GL_EQUAL
...
```

  - Compara o valor de referência com o valor presente no stencil

Reflexões

- Stencil Buffer – Operações
  - Definir actualizações do Stencil Buffer
    - `glStencilOp(`
      `GLenum fail,`
      `GLenum zfail,`
      `GLenum zpass);`



  - Define como o valor do stencil é alterado

Reflexões

- Nova Estratégia c/ Stencil Buffer
  - Desenhar cubo original
  - Desenhar plano só no Stencil Buffer
  - Desenhar cubo reflectido, com teste de stencil
  - Desenhar plano normalmente

Reflexões

- Nova Estratégia c/ Stencil Buffer
  - Desenhar plano só no Stencil Buffer

```
glEnable(GL_STENCIL_TEST);
glClear(GL_STENCIL_BUFFER_BIT);
glStencilFunc(GL_NEVER, 1, 0xffffffff);
glStencilOp(GL_REPLACE, GL_REPLACE, GL_REPLACE);

/* Os pixels do plano só afectam o stencil. */
DrawPlane();
```

Nunca passa o teste

set stencil=1

Reflexões

- Nova Estratégia c/ Stencil Buffer

  - Desenhar cubo reflectido, com stencil test

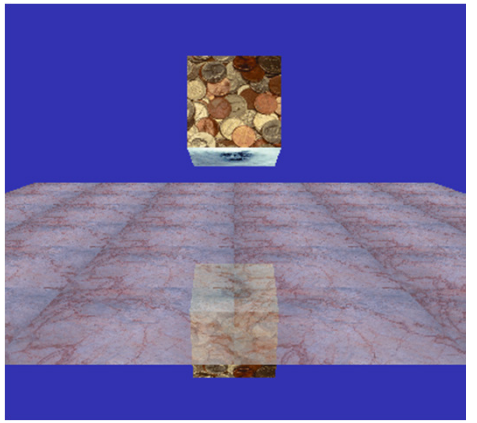
```
/* draw if stencil == 1 */
glStencilFunc(GL_EQUAL, 1, 0xffffffff);
glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);

//Draw reflected cube
...
glDisable(GL_STENCIL_TEST);
```

Passa o teste se stencil == 1

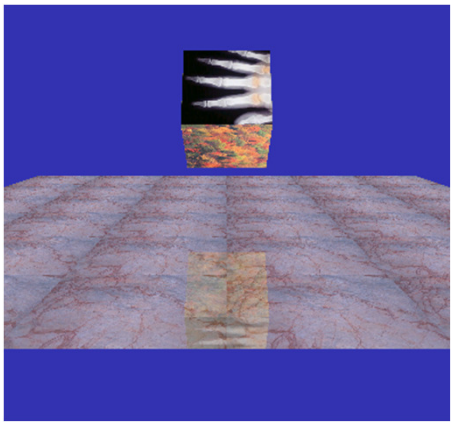
não altera o stencil

Reflexões

- Sem Stencil



- Com Stencil



Sumário

- Reflexões
- **Projective Texturing**
- Sombras - Introdução
- Sombras no Plano
- Shadow Volumes
- Shadow Maps

Projective Texturing

- Efeito do projector de slides
- Geração automática de coordenadas de texturas GL_EYE_LINEAR

Projective Texturing

- Geração de Coordenadas
  - com GL_EYE_LINEAR

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix} = \begin{bmatrix} \textit{Eye} \\ \textit{view} \\ \textit{(look at)} \\ \textit{matrix} \end{bmatrix} \begin{bmatrix} \textit{Modeling} \\ \textit{matrix} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \\ w_o \end{bmatrix}$$

Gera coordenadas de textura como se a câmara fosse o projector de slides

Projective Texturing

- Geração de Coordenadas

O que se pretende é projectar a textura a partir de um ponto arbitrário, e com uma orientação arbitrária

$$\begin{bmatrix} s \\ t \\ r \\ q \end{bmatrix} = \begin{bmatrix} 1/2 & & & 1/2 \\ & 1/2 & & 1/2 \\ & & 1/2 & 1/2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} \text{Light} \\ \text{frustum} \\ \text{(projection)} \\ \text{matrix} \end{bmatrix} \begin{bmatrix} \text{Light} \\ \text{view} \\ \text{(look at)} \\ \text{matrix} \end{bmatrix} \begin{bmatrix} \text{Inverse} \\ \text{eye} \\ \text{view} \\ \text{(look at)} \\ \text{matrix} \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

Sumário

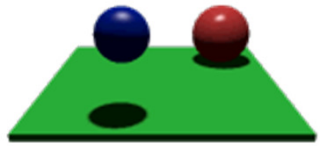
- Reflexões
- Projective Texturing
- **Sombras - Introdução**
- Sombras no Plano
- Shadow Volumes
- Shadow Maps

Sombras - Introdução

- Porquê sombras?
  - Mais realismo
  - Melhor sensação de profundidade
  - Melhor noção do posicionamento relativo dos objectos

Sombras - Introdução

- Onde estão as bolas?



Sombras - Introdução

- Vale tudo?

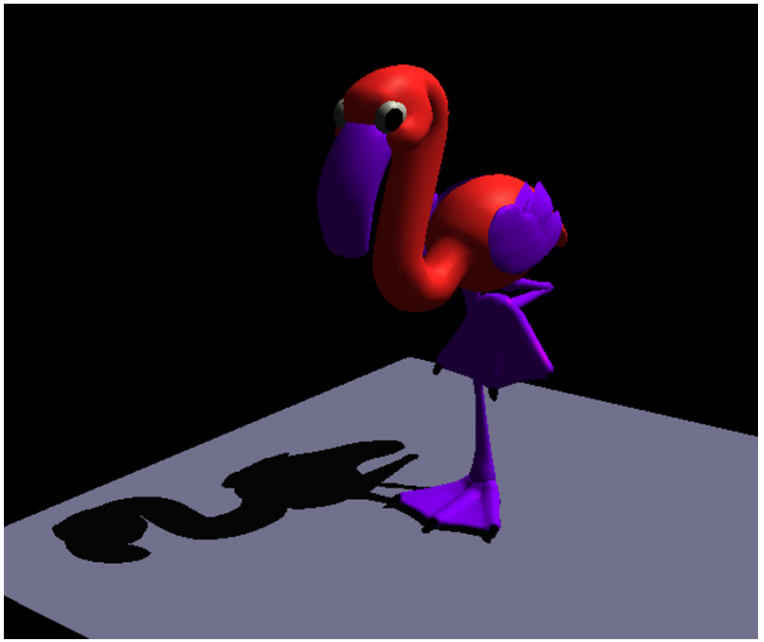
Images from TombRaider. ©Eidos Interactive

Sumário

- Reflexões
- Projective Texturing
- Sombras - Introdução
- **Sombras no Plano**
- Shadow Volumes
- Shadow Maps

Sombras no Plano

- A sombra é projectada no plano.
- Implica redesenhar o objecto escurecido e "achatado" no plano.

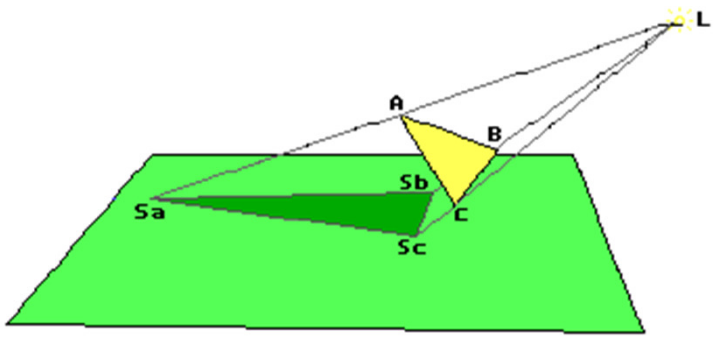


Sombras no Plano

- Estratégia:
  - Desenhar o objecto normalmente
  - Calcular a transformação para projectar o objecto no plano onde se vai criar a sombra
  - Desenhar de novo o objecto, com a transformação aplicada, e com uma cor escura.

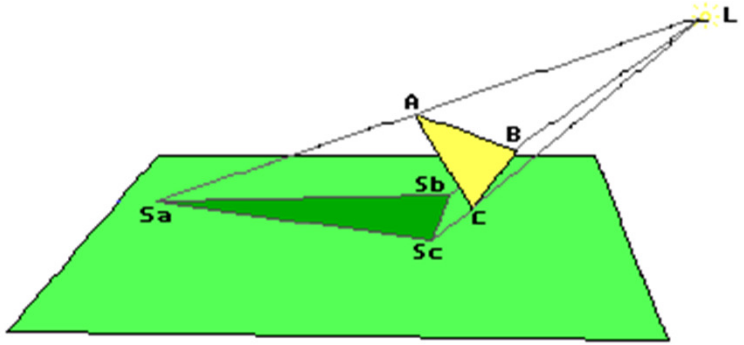
Sombras no Plano

- Transformação para desenhar a sombra
  - É necessário definir uma projecção no plano onde será criada a sombra para aplicar a todos os triângulos do objecto



Sombras no Plano

- Exemplo com plano = xz



$$sa_x = \frac{l_y a_x - l_x a_y}{l_y - a_y}$$

$$sa_z = \frac{l_y a_z - l_z a_y}{l_y - a_y}$$

Forma matricial

$$sa = \begin{pmatrix} l_y & -l_x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -l_z & l_y & 0 \\ 0 & -1 & 0 & l_y \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \\ 1 \end{pmatrix}$$

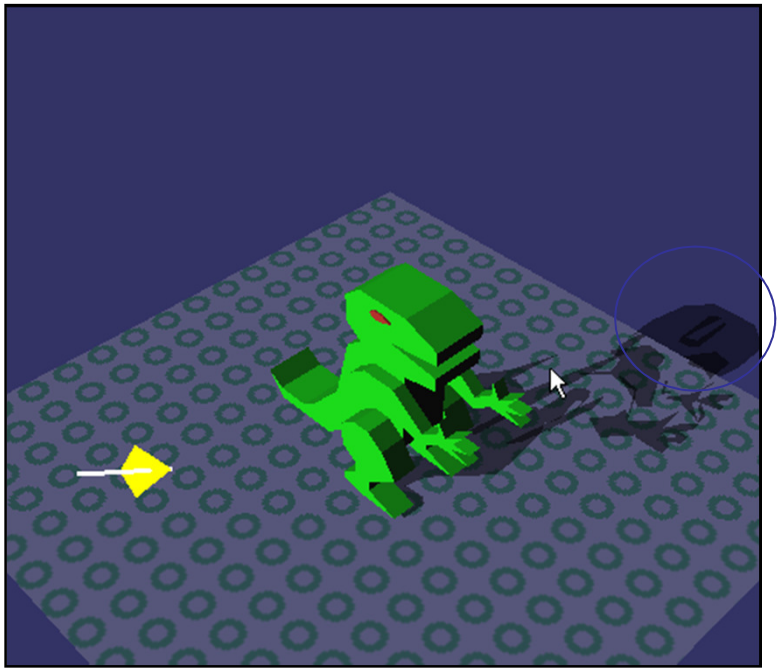
Sombras no Plano

- Estratégia:

1. Desenhar todos os triângulos do objecto
2. Calcular a matriz de projecção da sombra
3. Multiplicar a MODELVIEW pela matriz calculada
4. Cor = cinza escuro, activar blend
5. Desenhar de novo todos os triângulos do objecto

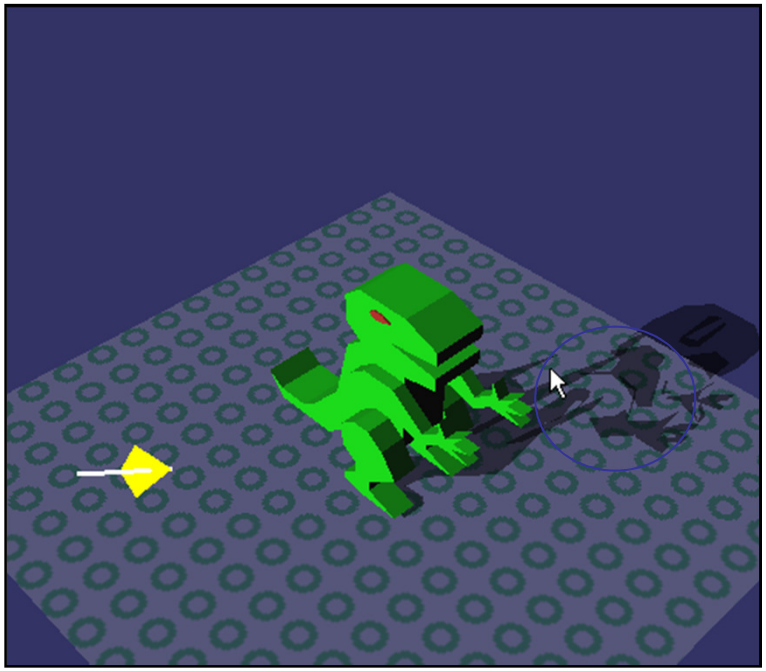
Sombras no Plano

- Questões:
  - A sombra estende-se para além do plano desenhado
    - Solução: Stencil Buffer



Sombras no Plano

- Questões:
  - Problemas devidos à precisão do Z-buffer
    - Solução:
      - offset dos polígonos



Sombras no Plano

- Questões:
  - Sombras de múltiplos triângulos sobrepostos causam sombras de intensidade diferente
    - Solução:
      - blend com equação GL_MIN

```
glBlendEquation(GL_MIN);
```

      - Stencil

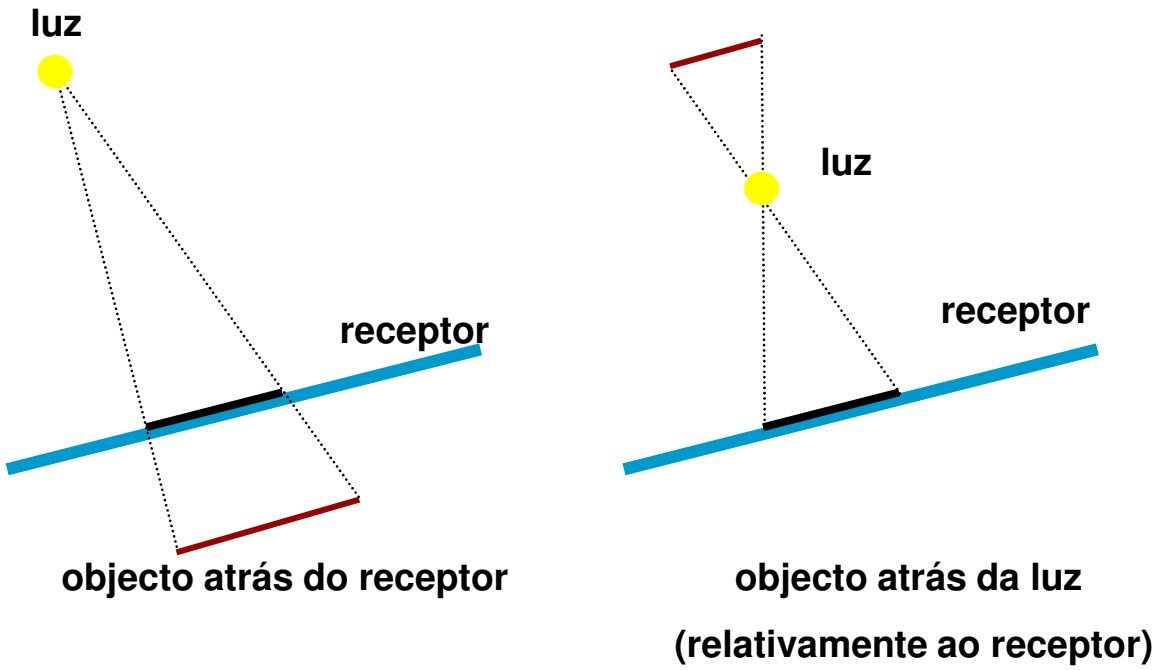
Sombras no Plano

- Estratégia com stencil:

1. Desenhar objecto
2. Activar o stencil (set = 1)
3. Desenhar o plano receptor
4. Activar o teste (stencil == 1)
5. Desactivar o Z-Buffer
6. Definir actualização do stencil
   (se passar teste do stencil então GL_ZERO)
7. Blending: final = dest * 0.2
8. Definir matriz de projecção e multiplicar pela MODELVIEW
9. Desenhar o objecto novamente

Sombras no Plano

- Problemas causados pela projecção:



Sombras no Plano

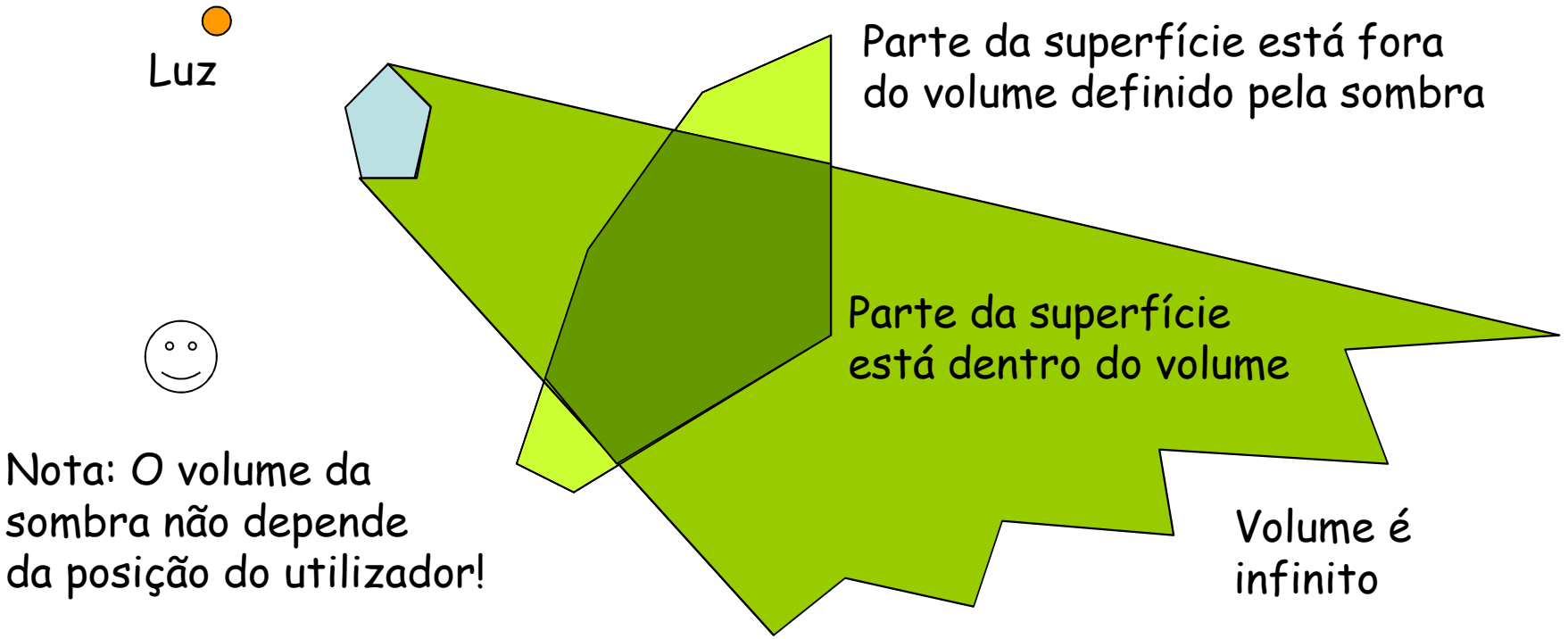
- Problemas:
  - Não são práticas quando a sombra é projectada num objecto complexo.
    - Cada triângulo do objecto receptor de sombra define um plano!!!

Sumário

- Reflexões
- Projective Texturing
- Sombras - Introdução
- Sombras no Plano
- **Shadow Volumes**
- Shadow Maps

Sombras – Shadow Volumes

- Volume definido pela sombra



Nota: O volume da sombra não depende da posição do utilizador!

Sombras – Shadow Volumes

- Algoritmo:
  - Determinar o *shadow volume*.
  - Render da cena com luz ambiente (com Z-Buffer)
  - Desenhar as *front faces* do volume (sem escrita no color ou Z Buffers)
    - incrementar stencil quando z-test OK
  - Desenhar as *back faces* do volume (sem escrita no color ou Z Buffers)
    - decrementar stencil quando z-test OK
  - Limpar o Z-Buffer
  - Render da cena com luzes quando stencil=0

Sombras – Shadow Volumes

- Problema:
  - Objectos em sombra ficam com um aspecto "flat".
  - Objectos reais recebem iluminação indirecta.
- Solução:
  - A cena deve ser "apropriadamente" iluminada para os objectos em sombra.

Sombras – Shadow Volumes

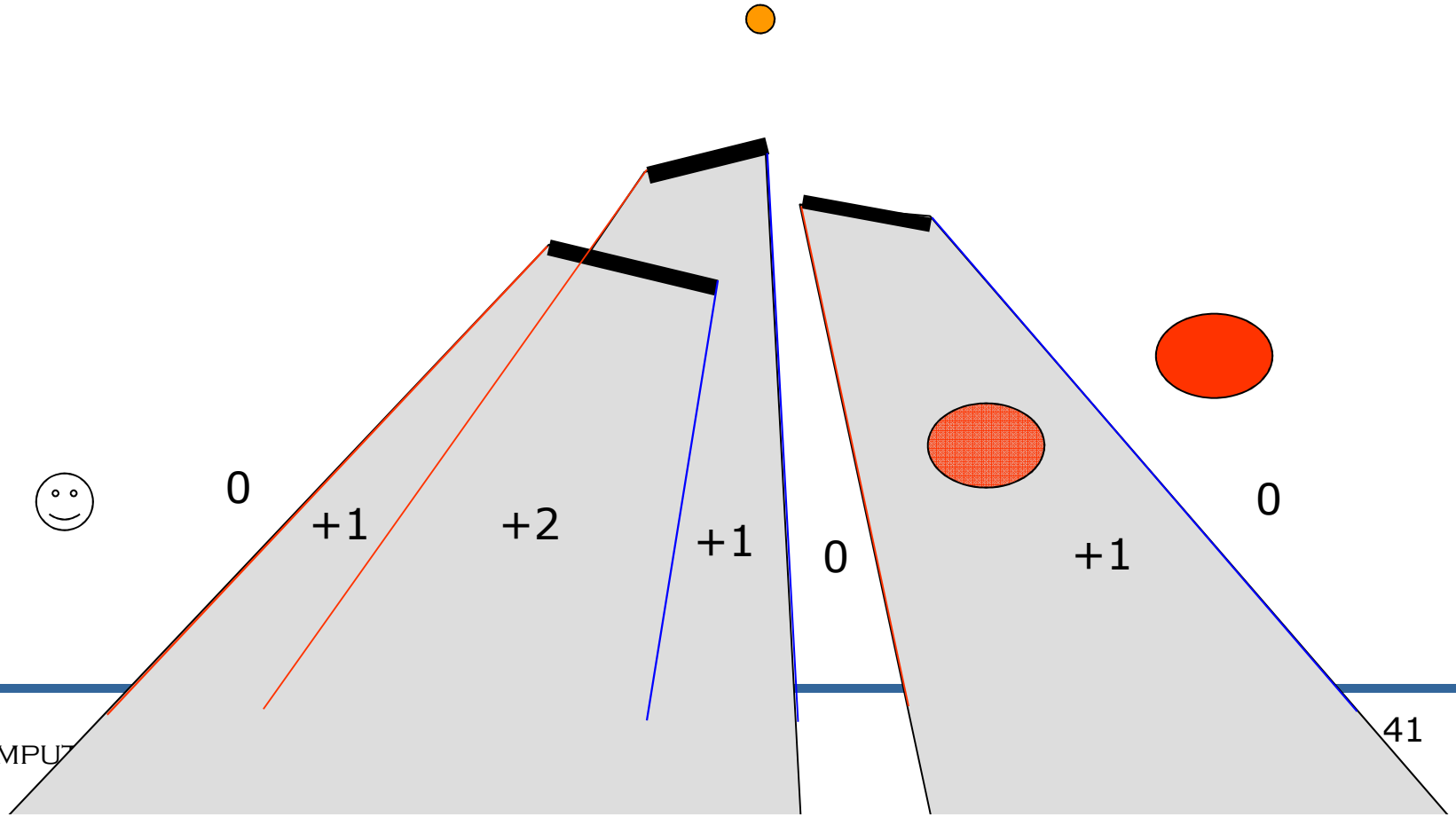
- Versão simplificada
  - Determinar o *shadow volume*.
  - Render da cena com luz normal (com Z-Buffer)
  - Desenhar as *front faces* do volume (sem escrita no color ou Z Buffer)
    - incrementar stencil quando z-test OK
  - Desenhar as *back faces* do volume (sem escrita no color ou Z Buffer)
    - decrementar stencil quando z-test OK
  - Render de um quad escuro em 2D que ocupe o ecrã, limitado pelo stencil (!= 0), e com blend activado.

Sombras – Shadow Volumes

- Versão Simplificada
  - também não é correcta porque:
    - os objectos em sombra foram também iluminados pela luz
    - O sombrear é uniforme, mas a iluminação não pode ser efectuada pela mesma luz!

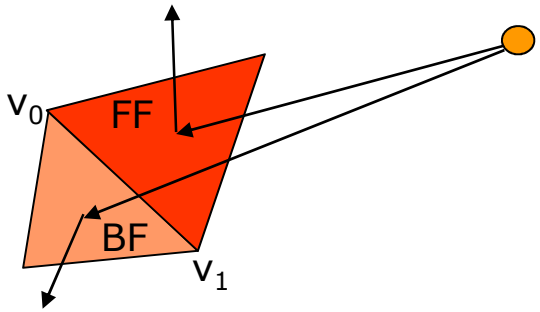
Sombras - Shadow Volumes

- Algoritmo: Exemplo 2D



Sombras – Shadow Volumes

- Determinar *Shadow Volume*
  - Uma aresta do modelo é uma aresta do SV se um dos polígonos for *front facing* e o outro for *back facing*



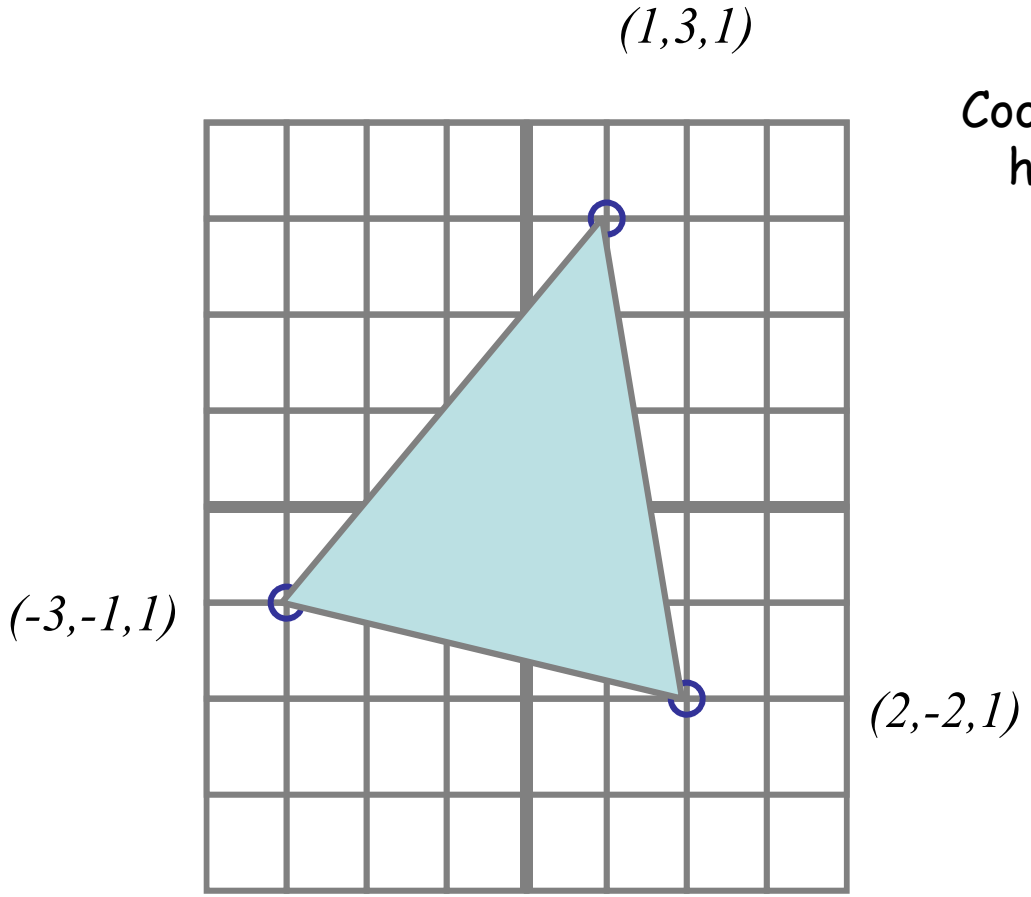
(v_0,v_1) é uma aresta do SV

Sombras – Shadow Volumes

- Determinar *Shadow Volume*
  - Para cada aresta estender um polígono da aresta até ao infinito!
  - Na prática bastaria estender até sair do view frustum...
  - ... mas em OpenGL é mais fácil até ao infinito!!!

Sombras – Shadow Volumes

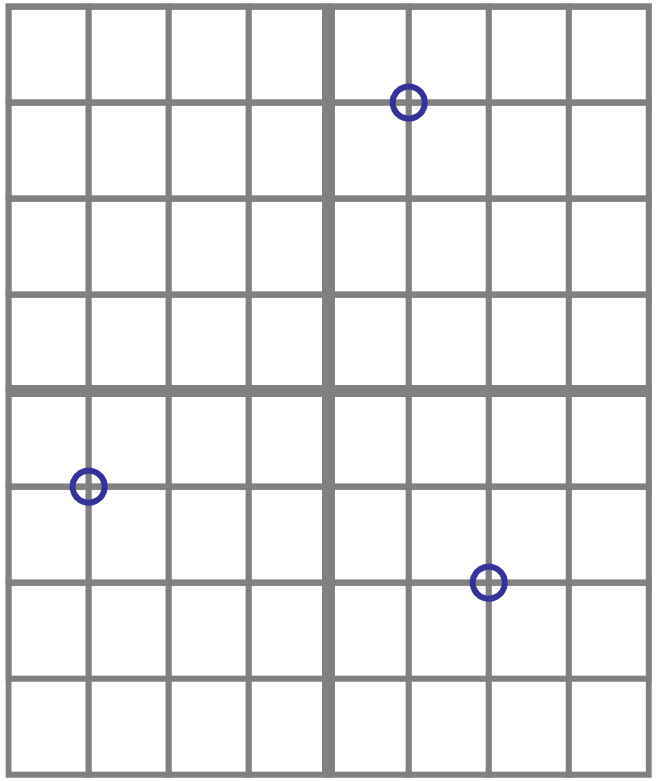
- Até ao infinito e mais além!



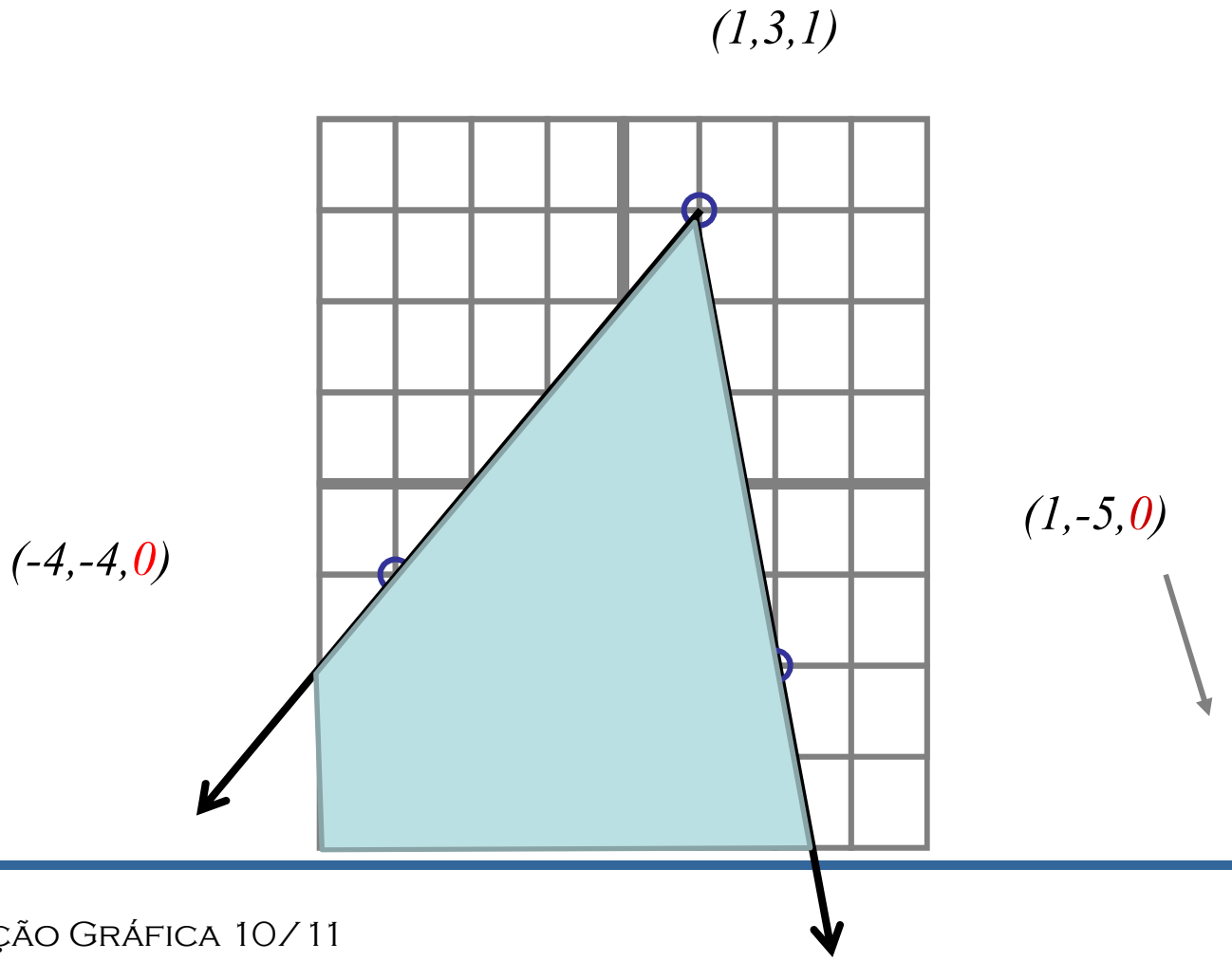
Coordenadas 2D homogéneas

Sombras – Shadow Volumes

- Até ao infinito e mais além!



Sombras - Shadow Volumes

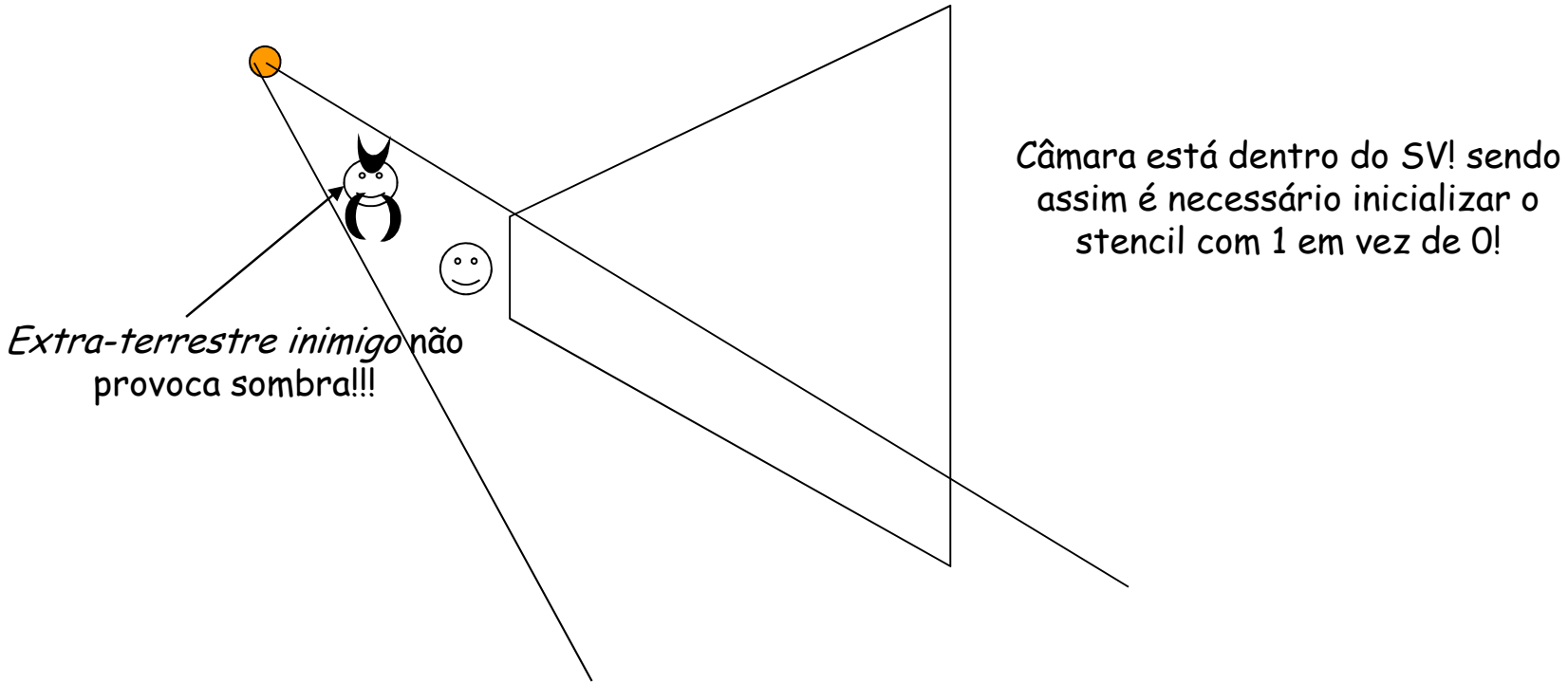


Sombras – Shadow Volumes

- Determinar *Shadow Volume*
  - Processo potencialmente demorado se for realizado online (luzes dinâmicas)
  - Para luzes estáticas o processo pode ser efectuado off-line.

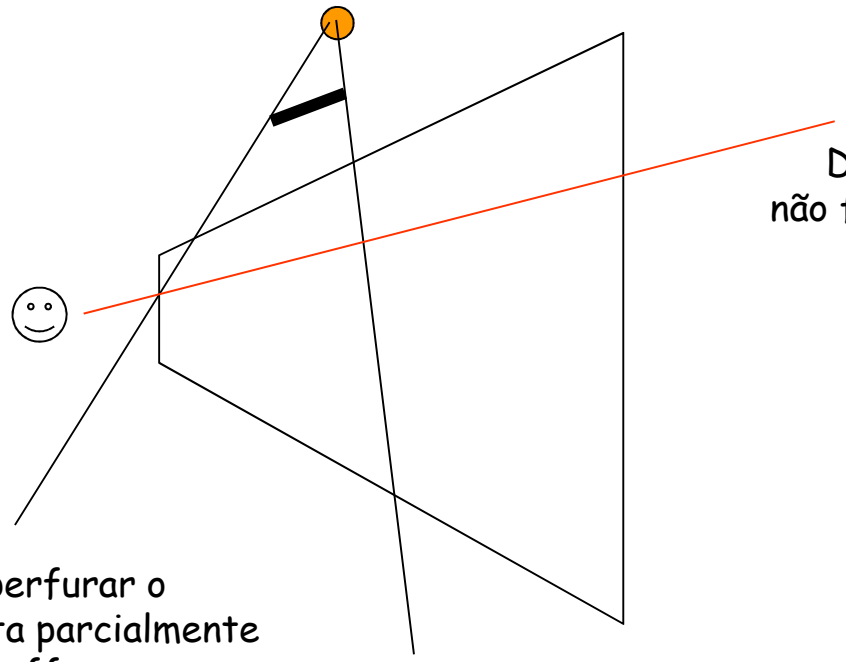
Sombras – Shadow Volumes

- Problemas:



Sombras – Shadow Volumes

- Problemas:

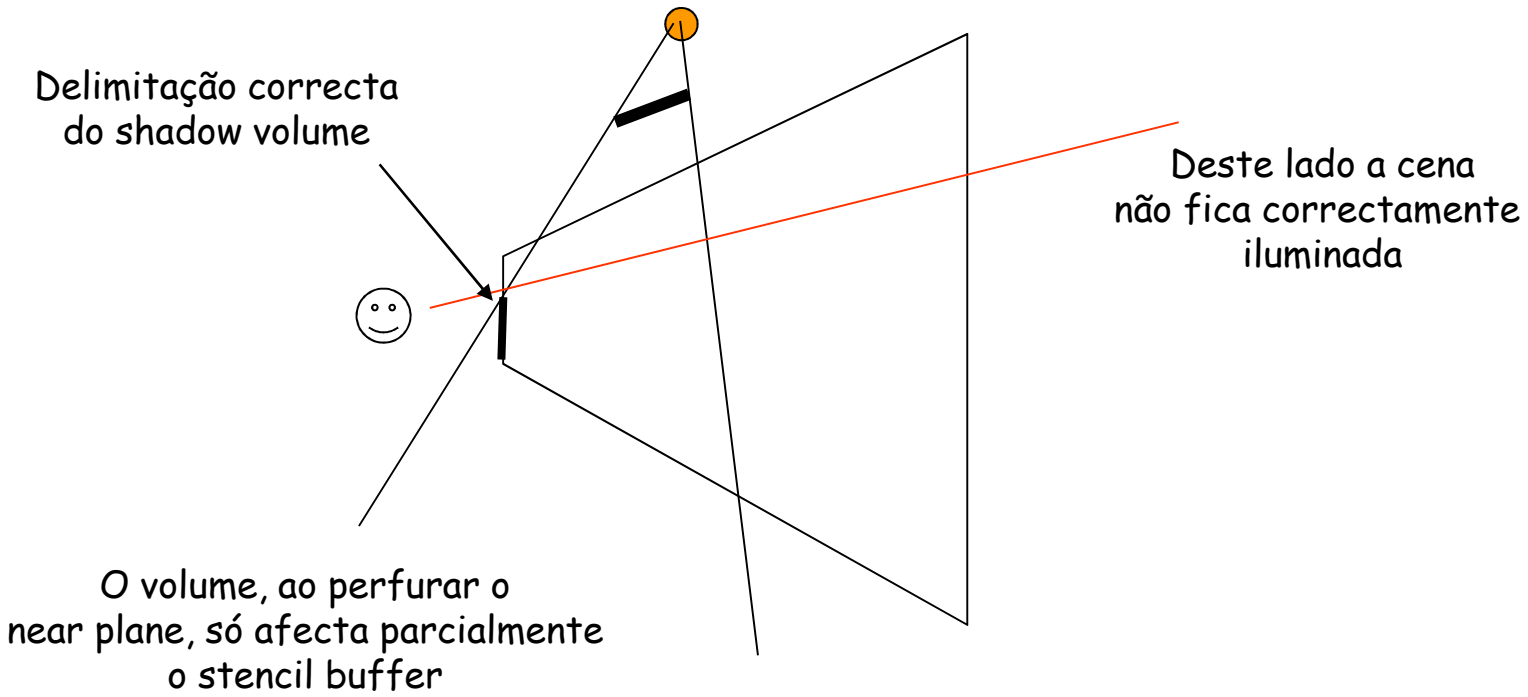


Deste lado a cena
não fica correctamente
iluminada

O volume, ao perfurar o
near plane, só afecta parcialmente
o stencil buffer

Sombras – Shadow Volumes

- Solução 1:



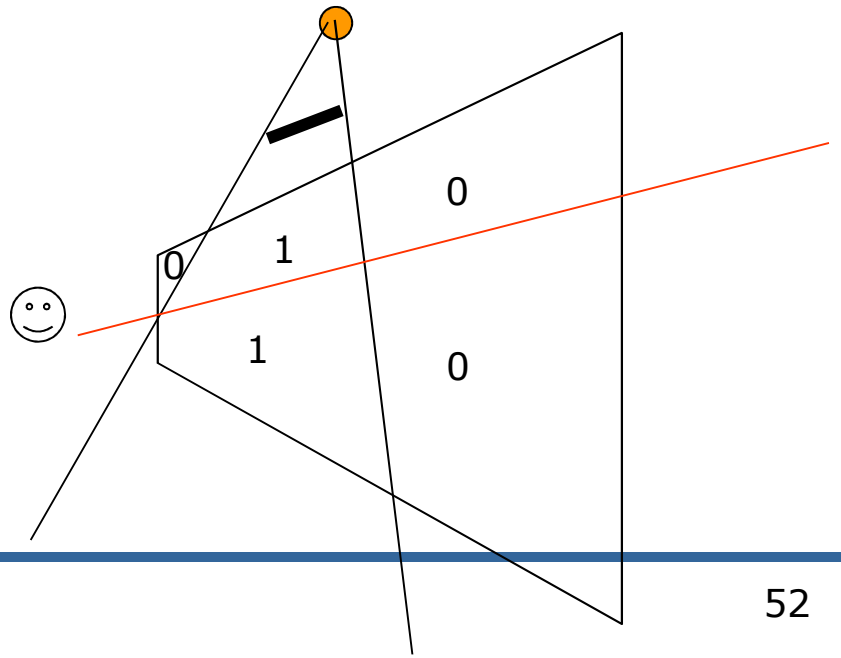
Sombras – Shadow Volumes

- Solução 2: Carmack's Reverse
  - solução prévia:
    - incrementar o stencil para as front faces quando z-buffer passa
    - decrementar o stencil para as back faces quando z-buffer passa
  - Carmack
    - incrementar o stencil quando o teste z-buffer falha para as back faces
    - decrementar o stencil quando o teste z-buffer falha para as front faces

Sombras – Shadow Volumes

- Carmack's Reverse
  - incrementar o stencil quando z-buffer teste falha para as back faces
  - decrementar o stencil quando z-buffer teste falha para as front faces

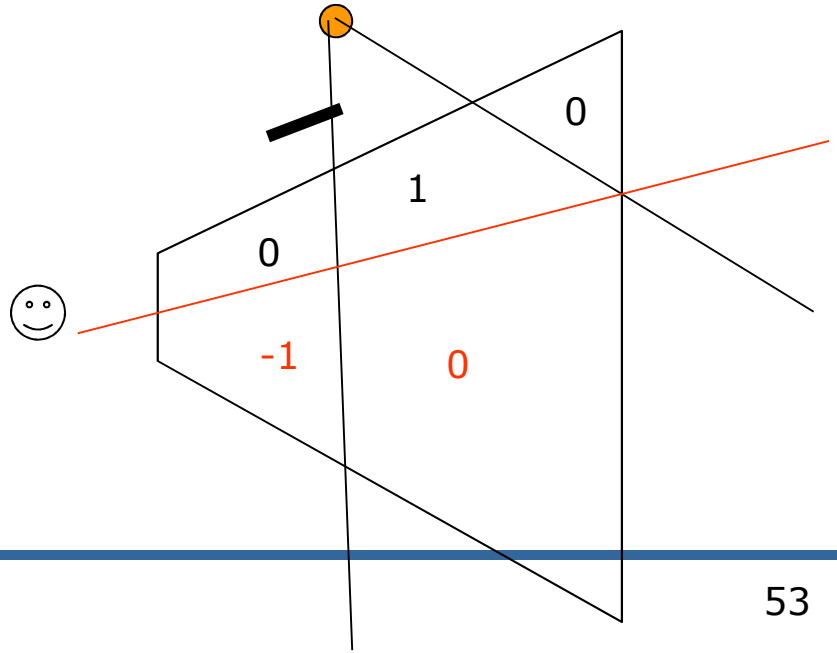
Resolve o problema do near plane



Sombras – Shadow Volumes

- Carmack's reverse
  - incrementar o stencil quando o teste z-buffer falha para as back faces
  - decrementar o stencil quando o teste z-buffer falha para as front faces

Novo problema com o far plane!



Sombras – Shadow Volumes

- Carmac's Reverse: Far Plane
  - Problema: Far Plane
  - Soluções?
    - Semelhante à apresentada para o near plane, ou,
    - Considerar luzes com atenuação e afastar o far plane de forma a evitar a situação!

Sombras – Shadow Volumes

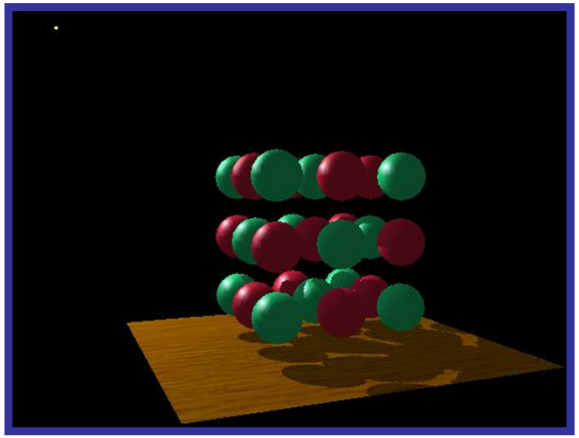
- Algoritmo que funciona com qualquer cena
- Problema de desempenho:
- ...implica o rendering de potencialmente grandes superfícies invisíveis: as paredes do shadow volume.

Sumário

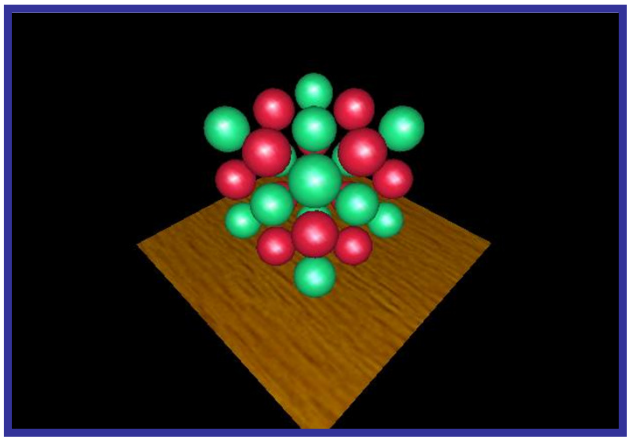
- Reflexões
- Projective Texturing
- Sombras - Introdução
- Sombras no Plano
- Shadow Volumes
- **Shadow Maps**

Sombras - Shadow Maps

Cena vista da câmara

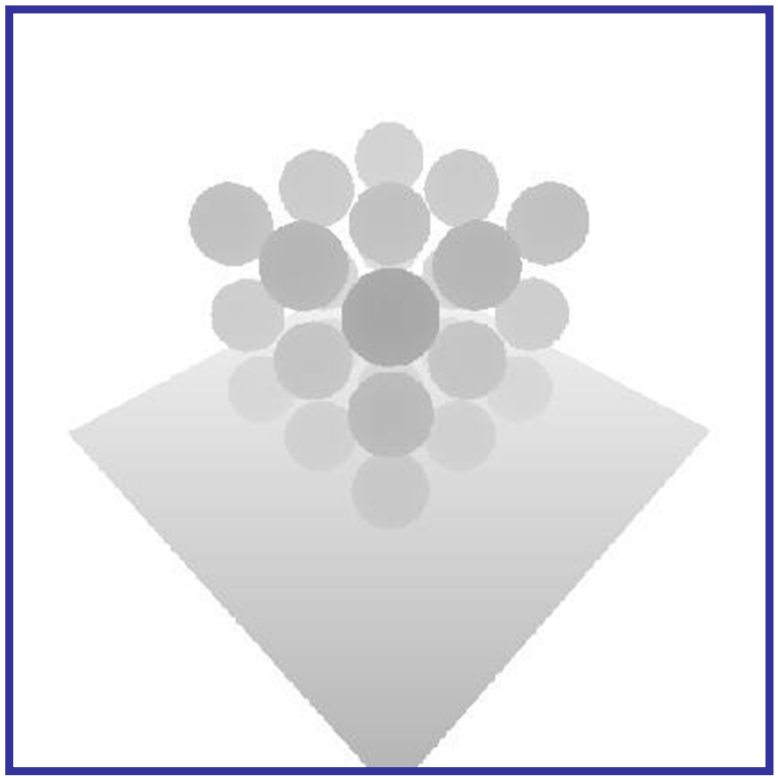


Cena vista da luz



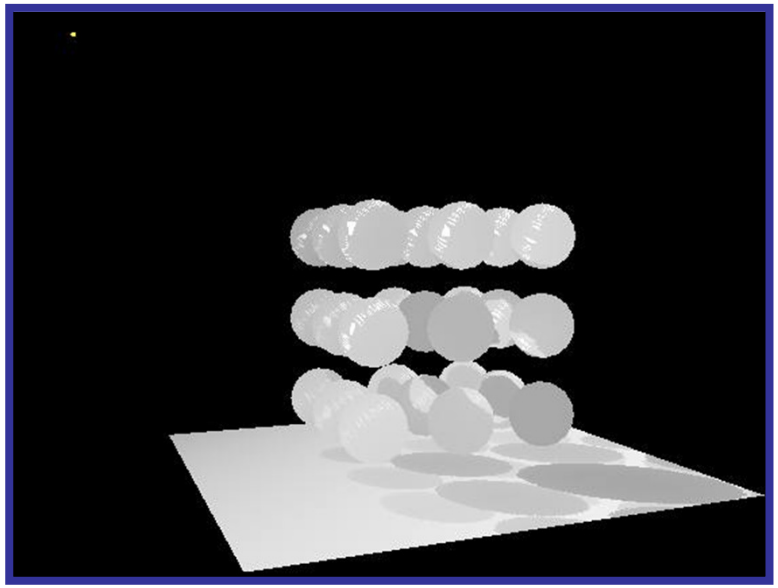
Sombras - Shadow Maps

- Mapa de profundidade gerado a partir do ponto de luz



Sombras - Shadow Maps

- Vista da câmara com o mapa de profundidade projectado (projective texturing)



Sombras – Shadow Maps

- Algoritmo:
  - Para cada pixel:
    - determinar posição ponto 3D
    - calcular d1 = distância à fonte de luz
    - ler distância obtida no mapa de profundidade = d2
    - se (d1 > d2)
      - Ponto está em sombra
    - senão
      - Ponto está iluminado

Sombras – Shadow Maps

- A comparação pode ser realizada em hardware
  - Coordenadas de textura geradas (s,t,r,q)
  - (s/q, r/q) fornecem a localização no mapa de profundidade aplicado ao pixel
  - r/q fornece a distância à fonte de luz (devido ao projective texturing).
  - Se depthmap (s/q,r/q) < r/q então ponto está em sombra, senão está iluminado.

Sombras – Shadow Maps

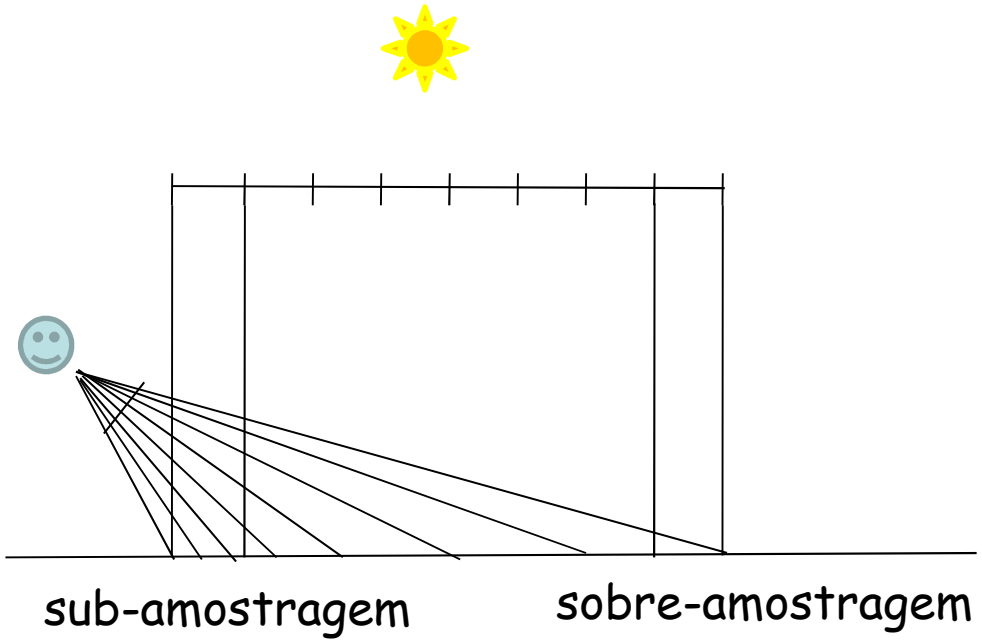
- Algoritmo:
  - Render da cena com "luz ambiente"
  - Render da cena com luz normal
    - Projectar textura do mapa de profundidade
      - resultado da comparação diz-nos, a intensidade da iluminação (0 ou 1).
      - utilizar alpha test para prevenir que se apague a iluminação ambiente.

Sombras – Shadow Maps

- Problemas:
  - Resolução do Z-Buffer
  - Resolução do mapa de profundidade
  - Amostragem

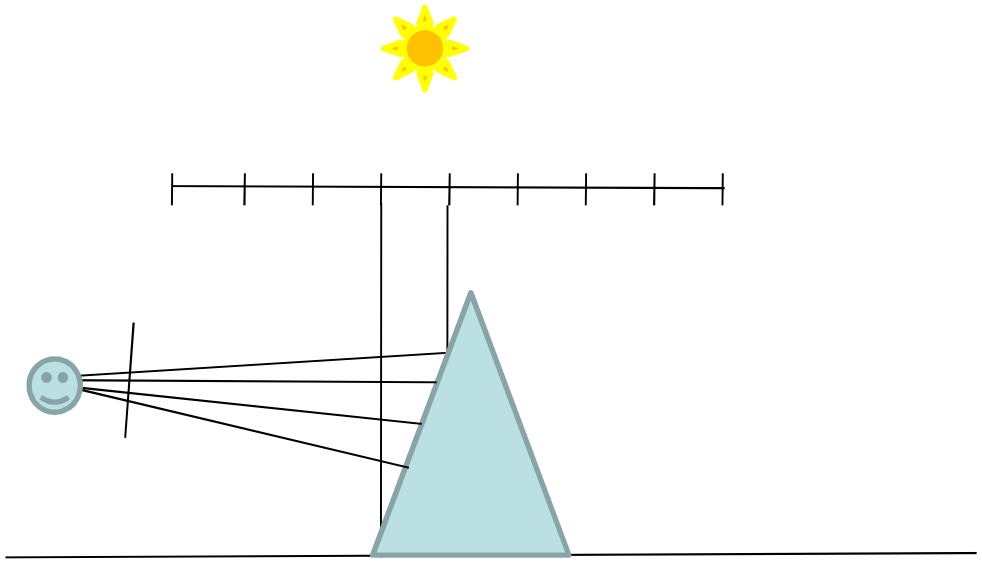
Sombras - Shadow Maps

- Perspectiva



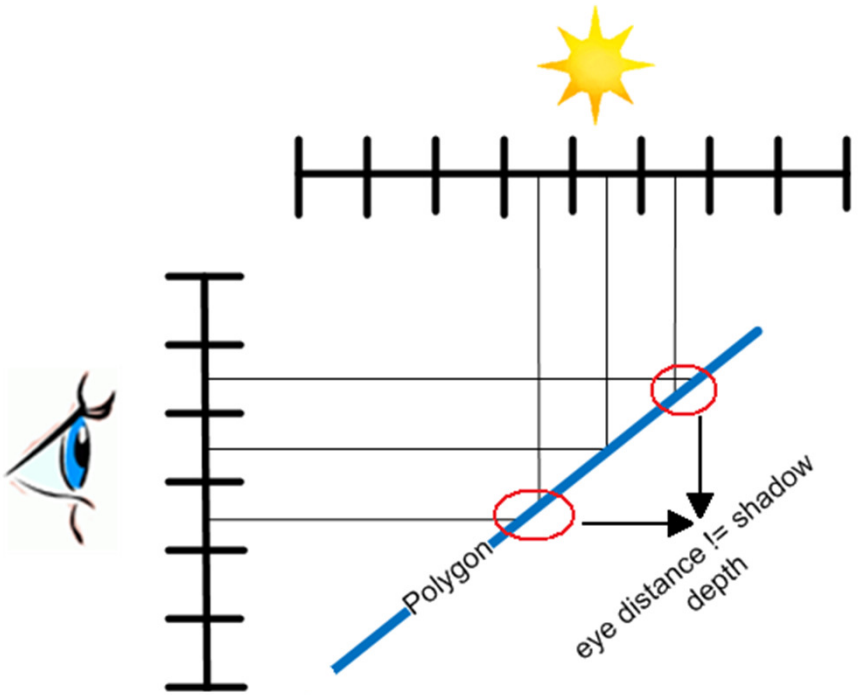
Sombras - Shadow Maps

- Perspectiva



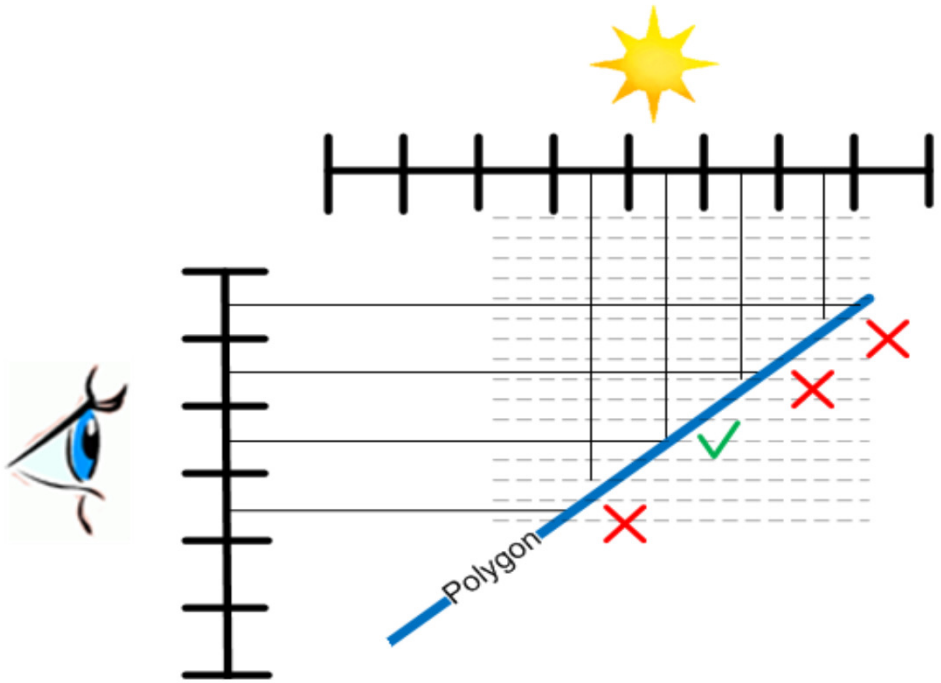
Sombras – Shadow Maps

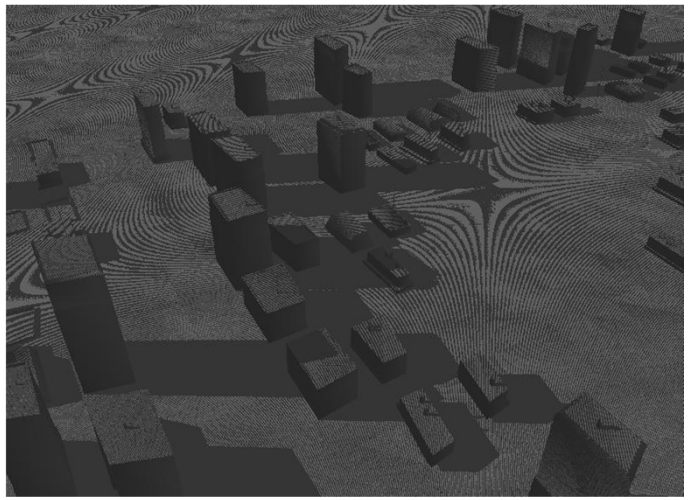
- Amostragem – Desacerto (mismatch)



Sombras – Shadow Maps

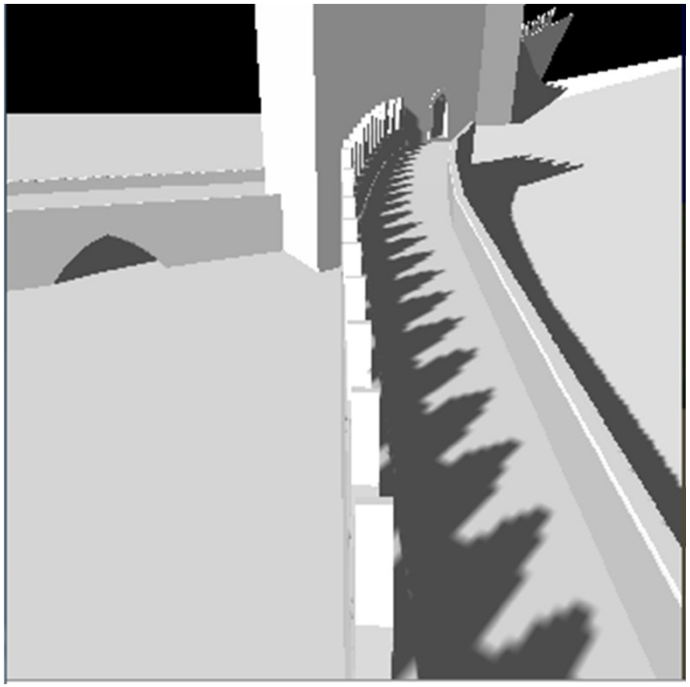
- Amostragem – Resolução das profundidades no Z Buffer



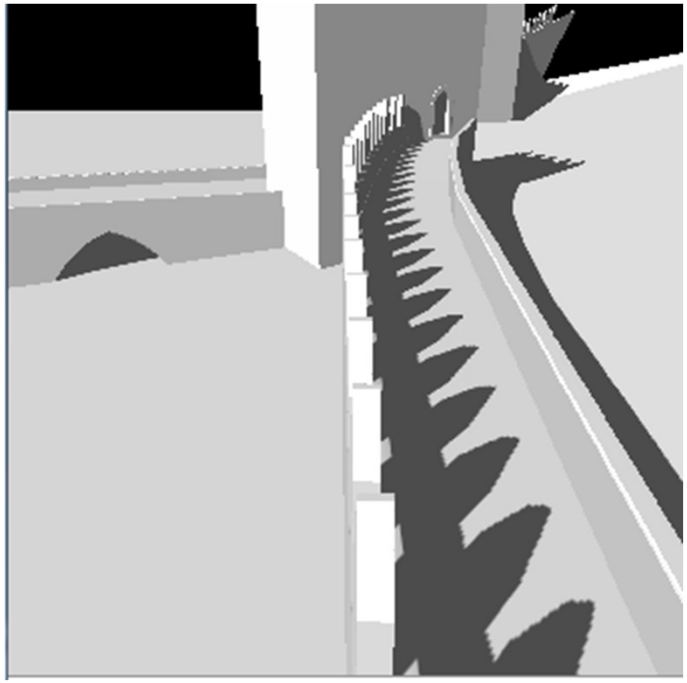


Sombras – Shadow Maps

- Dimensão do Mapa de Profundidades



1024x1024



4096x4096

Referências

- developer.nvidia.com