



UFABC

Computação Gráfica

André Brandão

Aula 07

Ray Tracing (Traçados de Raios)

Autoria

- Os slides da aula de hoje foram inspirados nos materiais disponibilizados pelos professores:
 - Anselmo Montenegro (UFF) -
<http://www2.ic.uff.br/~anselmo/cursos/CGI/CGI20132/CGI20132.html>
 - Wolfgang Hürst (Utrecht Universiteit) -
http://www.cs.uu.nl/docs/vakken/gr/2012-13/Slides/INFOGR_2012-2013_lecture-11_ray-tracing_annotated.pdf

Relismo x Eficiência

- Realismo: filme
 - <https://www.youtube.com/watch?v=sED6FRXIHJc>
- Eficiência: jogo
 - https://www.youtube.com/watch?v=hSLR_lwrOxU

Realismo



<https://www.macxdvd.com/mac-dvd-video-converter-how-to/best-disney-pixar-movies.htm>

Eficiência



Introdução

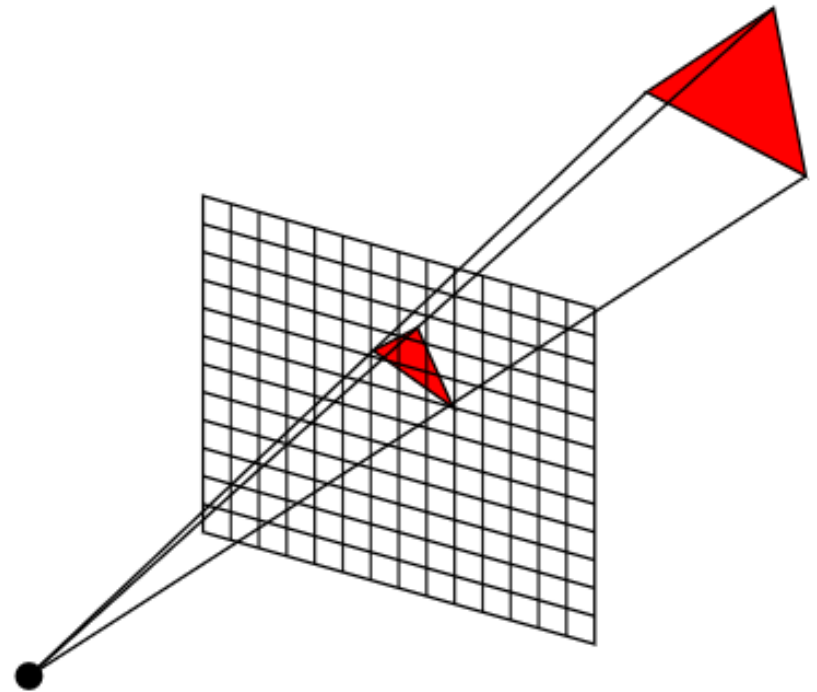
- Renderização (*rendering*) é uma das operações básicas da computação gráfica.
- Renderização é, em termos gerais, tomar uma cena, modelo, objetos geométricos organizados em 3D e produzir uma imagem 2D que apresenta os objetos vistos de um ponto de vista particular.

Introdução

- A entrada da renderização é um conjunto de objetos e a saída é uma lista de pixels.
- Há duas maneiras básicas de se renderizar objetos.
 - Renderização baseada em objeto (*object-order rendering*): para cada objeto, são verificados os pixels que irão compor a imagem do mesmo.
 - Renderização baseada em imagem (*image-order rendering*): para cada pixel, são verificadas quais partes do objeto serão mostradas no plano de apresentação.

Object-order rendering

- Um método popular para geração de imagens de um modelo 3D é por projeção, por exemplo:
 - Triângulos em 3D projetados para triângulos em 2D
 - Projeção de vértices
 - Preenchimento e sombreamento de um triângulo 2D
- Note:
 - Ray tracing = baseado em pixel,
 - Métodos de projeção = baseado em objeto

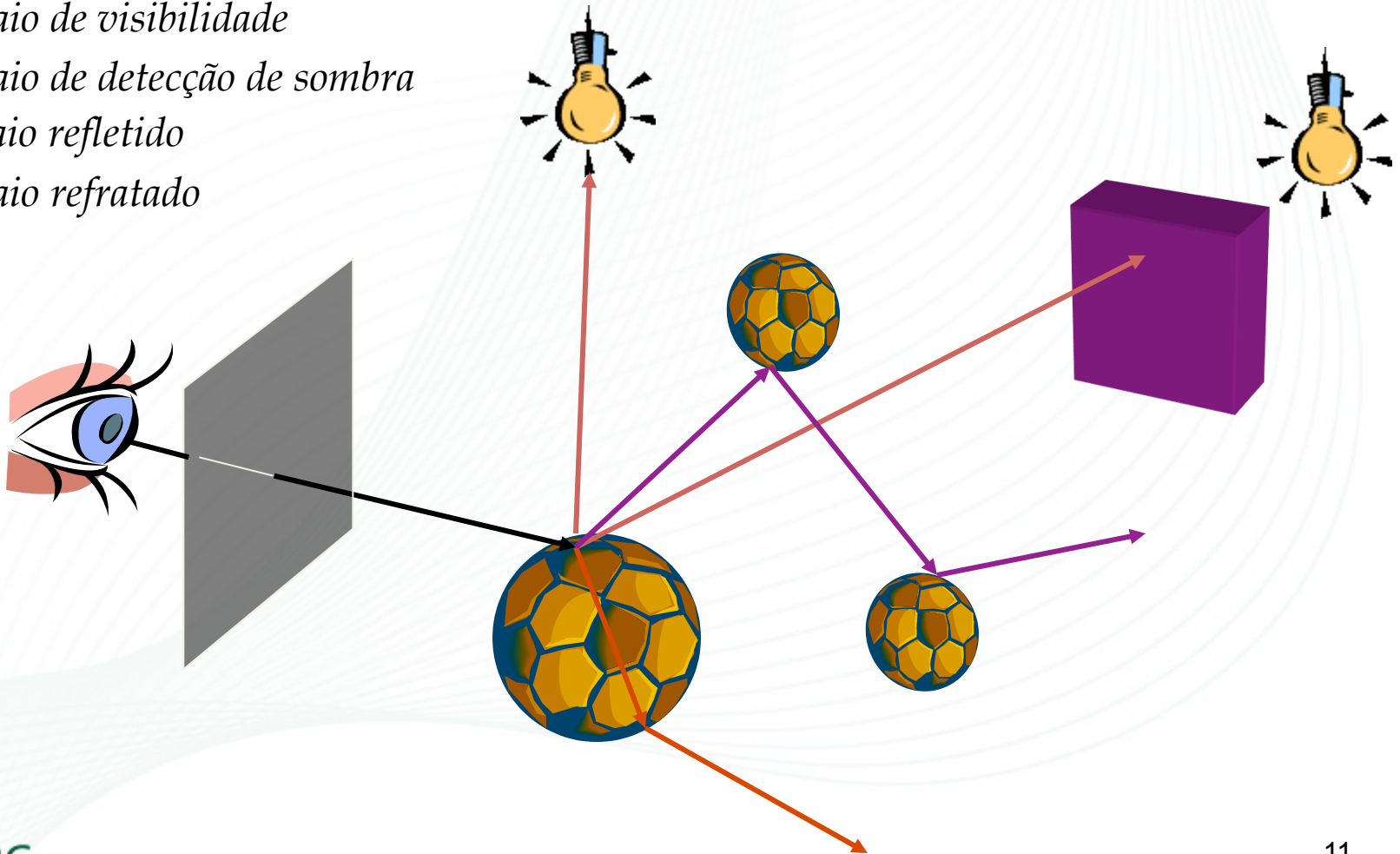


Ray tracing

- Tanto a abordagem baseada em objetos como a de imagens podem gerar imagens 2D.
- Porém, na abordagem baseada em imagens, é mais simples de se gerar imagens com alguns efeitos, como sombras e reflexões.
- Para gerar diferentes efeitos, o custo computacional da abordagem baseada em imagens, que é o caso do Ray Tracing, é mais caro.

Ray tracing

- *Raio de visibilidade*
- *Raio de detecção de sombra*
- *Raio refletido*
- *Raio refratado*



Sumário

- Algoritmo básico de Ray Tracing
- Perspectiva
- Computação dos raios de visualização
- Interseção raio-objeto
- Sombreamento (*Shading*)
- Um programa de Ray Tracing
- Sombras
- Reflexão especular ideal

Sumário

- **Algoritmo básico de Ray Tracing**
- Perspectiva
- Computação dos raios de visualização
- Interseção raio-objeto
- Sombreamento (*Shading*)
- Um programa de Ray Tracing
- Sombras
- Reflexão especular ideal

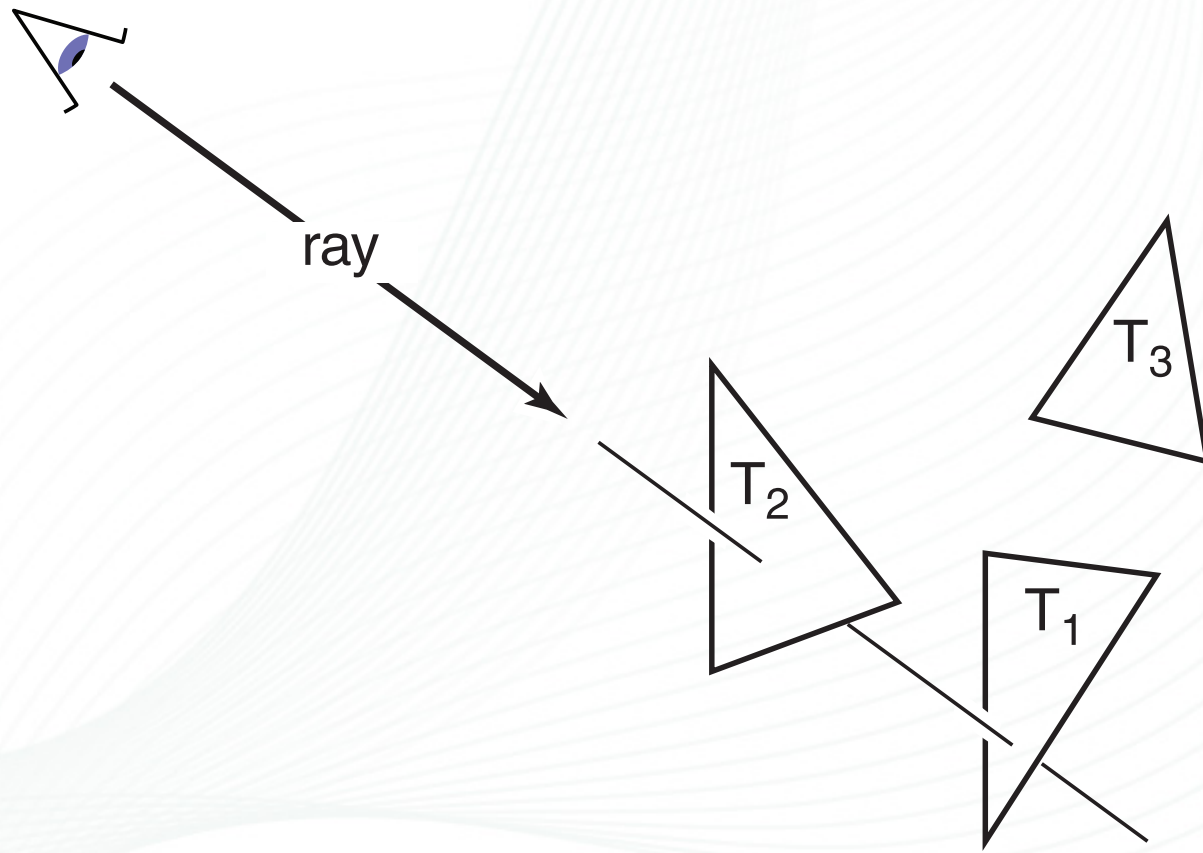
Algoritmo básico de ray tracing

- O traçado de raios funciona pela computação de um pixel por vez e, para cada pixel, a tarefa básica é de encontrar a parte do objeto que é vista na posição do pixel corrente.
- Para tanto, cada pixel traça um raio em direção ao fundo do objeto. Se o raio tem interseção com o objeto, então, este será mapeado no pixel corrente. Então, uma série de computações são geradas para executar os efeitos necessários para gerar os argumentos desse pixel.

Algoritmo básico de ray tracing

- Se o raio gerado a partir do ponto de vista do pixel corrente não tiver interseção com qualquer objeto, então o pixel recebe a cor de fundo dos objetos considerados na cena.

Algoritmo básico de ray tracing



Algoritmo básico de ray tracing

- O traçado de raios contém três etapas:
 1. Geração de raio: computa a origem e direção de cada ponto de vista de cada pixel, baseado na geometria de câmera;
 2. Interseção de raio: encontra a interseção mais próxima do raio com um objeto na cena; e
 3. Shading: computa a cor do pixel, baseado nos resultados da interseção de raio.

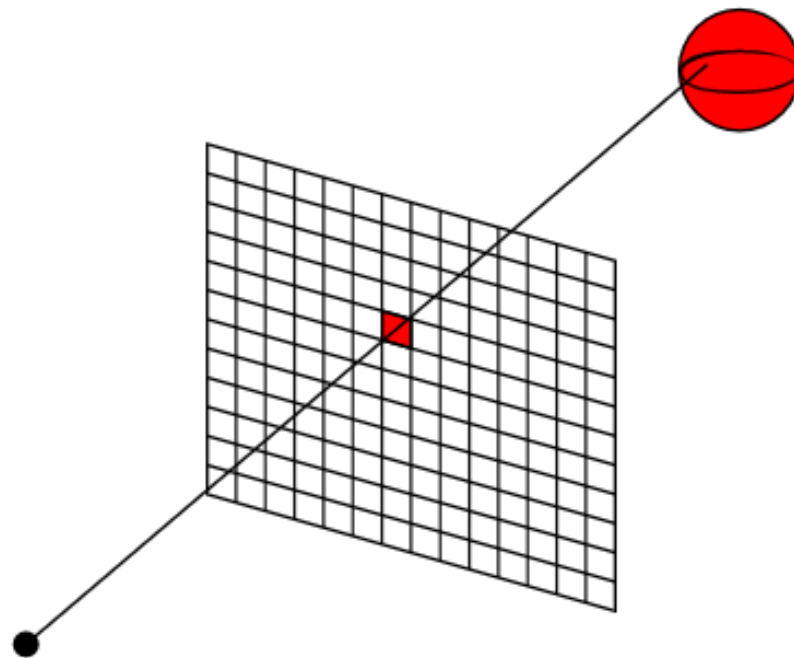
Algoritmo básico de ray tracing

- O traçado de raios contém três etapas:
 1. **Geração de raio: computa a origem e direção de cada ponto de vista de cada pixel, baseado na geometria de câmera;**
 2. **Interseção de raio: encontra a interseção mais próxima do raio com um objeto na cena; e**
 3. Shading: computa a cor do pixel, baseado nos resultados da interseção de raio **(próxima aula)**.

Algoritmo básico de ray tracing

Para cada **pixel** faça

- compute o raio de visualização
- encontre o 1º objeto interceptado pelo raio e a sua normal \vec{n}
- atribua a cor do pixel o mesmo valor do ponto interceptado pelo raio, considerando a luz e, \vec{n}



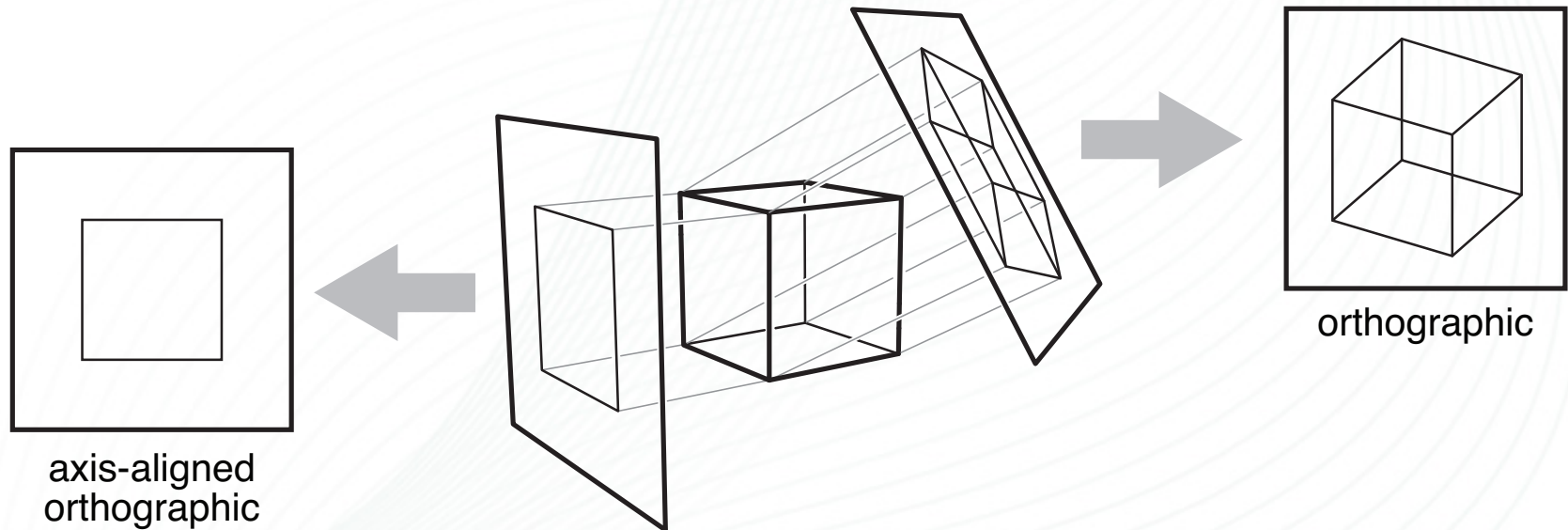
Sumário

- ~~Algoritmo básico de Ray Tracing~~
- **Perspectiva**
- Computação dos raios de visualização
- Interseção raio-objeto
- Sombreamento (*Shading*)
- Um programa de Ray Tracing
- Sombras
- Reflexão especular ideal

Perspectiva

- A maneira mais simples de se ter uma projeção de uma cena 3D em uma representação 2D é por meio da perspectiva linear.
- Na perspectiva linear, um plano é colocado na cena e os objetos são mapeados para este plano.
- Se o plano estiver perpendicular, então a projeção é chamada de ortográfica.

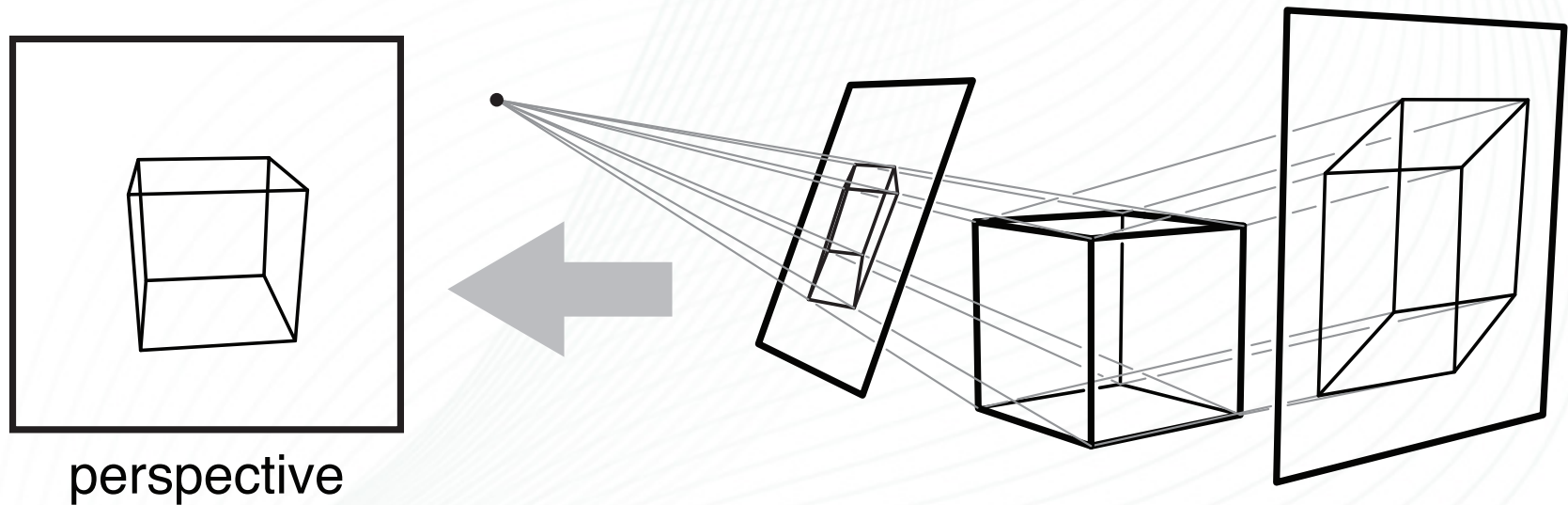
Projeção ortográfica



Projeção perspectiva

- Outra maneira de se projetar 3D em 2D é por meio de perspectiva.
- Neste caso, os raios são disparados por um mesmo ponto de vista, ao invés de diversos raios serem disparados, paralelamente, em diferentes pontos de vista (ortográfica).

Projeção perspectiva



Sumário

- ~~Algoritmo básico de Ray Tracing~~
- ~~Perspectiva~~
- **Computação dos raios de visualização**
- Interseção raio-objeto
- Sombreamento (*Shading*)
- Um programa de Ray Tracing
- Sombras
- Reflexão especular ideal

Computação dos raios de visualização

Representação matemática de um raio:

- Exemplo: vamos supor que uma reta é representada, no espaço 3D, pela seguinte equação paramétrica:

$$x = 2 + 7t;$$

$$y = 1 + 2t;$$

$$z = 3 - 5t.$$

Computação dos raios de visualização

- Porém, pode-se representar a reta pela seguinte forma de vetor:

$$p = o + td,$$

onde,

$$o = (2, 1, 3)$$

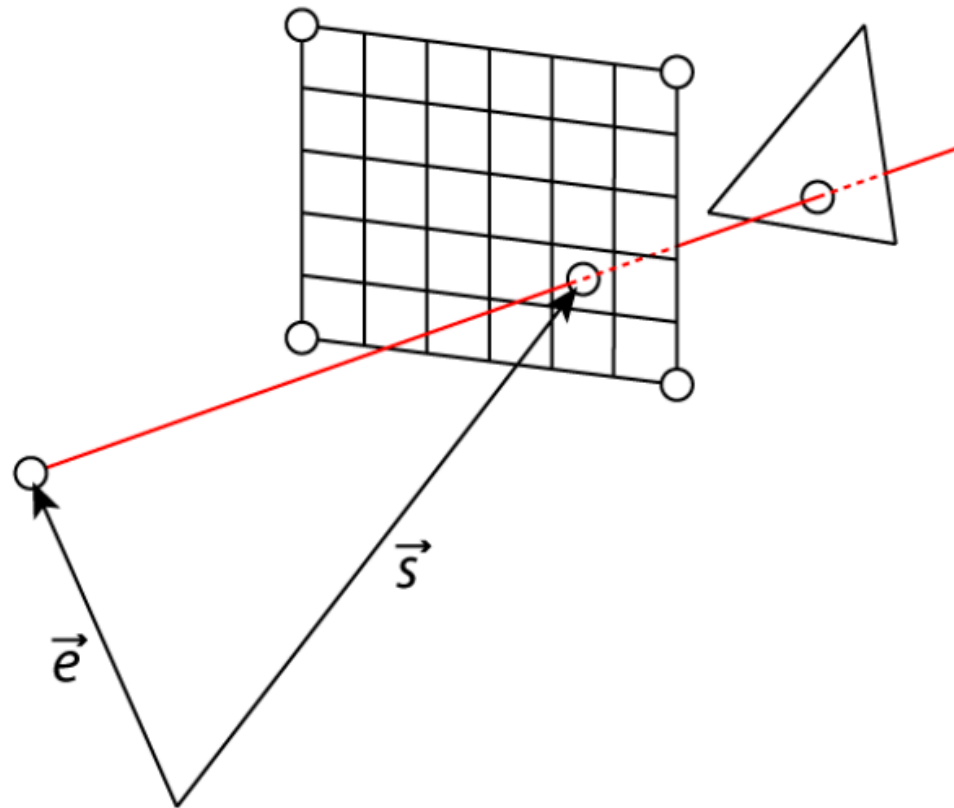
$$d = (7, 2, -5)$$

Computação dos raios de visualização

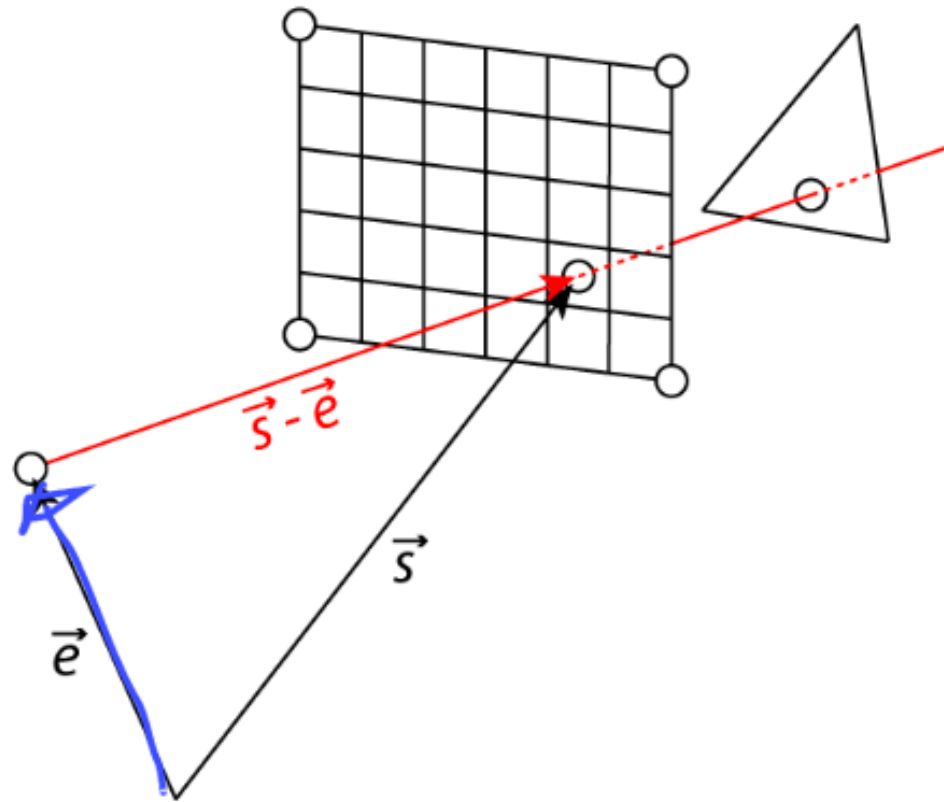
- Para representar um raio, pode-se usar a representação de uma reta.
- Um raio é um ponto de origem e uma direção de propagação. No nosso caso, a origem será o olho do observador. O vetor “e” é a distância do observador até a altura do pixel corrente. “s” é a distância da origem até o pixel corrente. (s-e) é um vetor que define a direção de um raio.

$$p(t) = e + t(s - e)$$

Computação dos raios de visualização

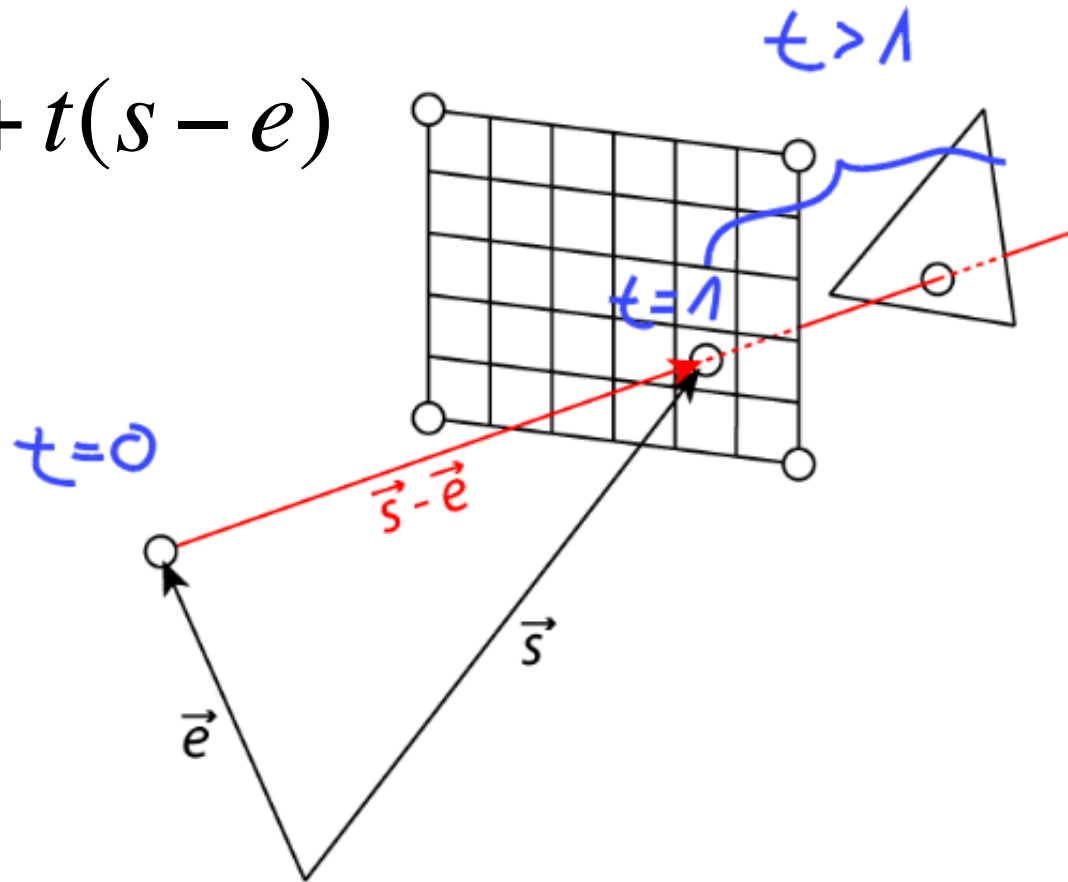


Computação dos raios de visualização



Computação dos raios de visualização

$$p(t) = e + t(s - e)$$



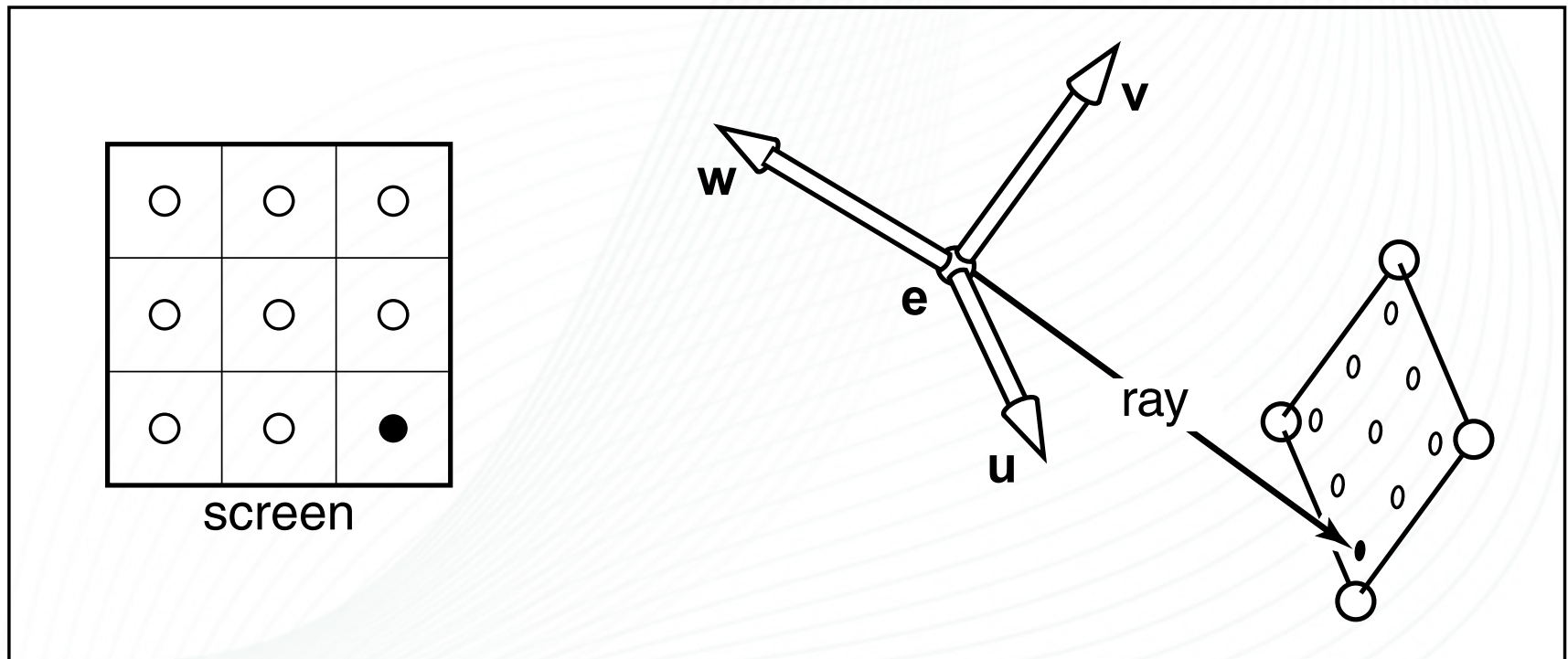
Computação dos raios de visualização

- Pode-se interpretar o raio como:
“Nós avançamos de ‘e’ pelo vetor (s-e) em uma fração de distância ‘t’ para encontrarmos o ponto ‘p’ que está no raio”.
- Dado o valor t, nós podemos determinar um ponto p.
- O ponto “e” é a origem do raio.
- A direção do raio é dada por (s-e).

Computação dos raios de visualização

- Para valores $t < 0$, o ponto p estará atrás do olho do observador.
- A geração de raios parte do olho do observador (“ e ” é a origem) e, a partir de “ e ”, forma-se uma base ortonormal.
- Para a formação da base ortonormal, são necessários os vetores $\{u, v, w\}$, onde u é o vetor que aponta à direita de e , v aponta para cima e w aponta para trás.

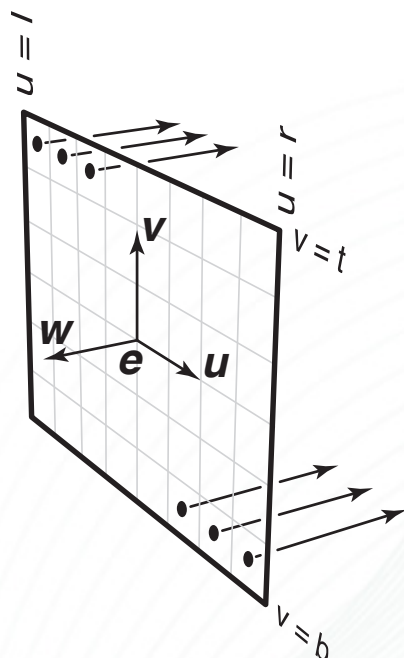
Computação dos raios de visualização



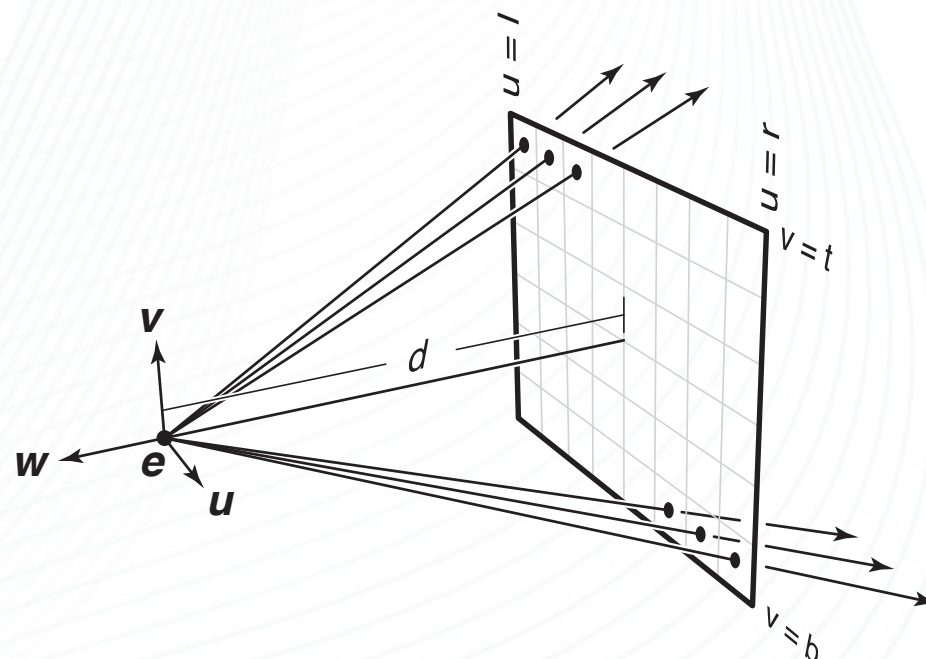
Diferentes visões

- Visão ortográfica
- Visão em perspectiva

Ortográfica X Perspectiva



Parallel projection
same direction, different origins



Perspective projection
same origin, different directions

Visão ortográfica

- Todos os raios têm a mesma direção $-w$. Porém, diferentes vetores paralelos são usados com origens nos pixels do plano de projeção.
- Portanto, é necessário definir a base $\{u, v, -w\}$, para cada pixel do plano de projeção.
- A dimensão da imagem é definida pelos números: l e r (*left, right*), e t e b (*top, bottom*).

Visão ortográfica

- l e r referem-se às posições direita e à esquerda de “e”, por meio do vetor u .
- b e t referem-se às posições acima e abaixo de “e”, por meio do vetor v .
- Usualmente, os valores dos quatro números são:

$$l < 0 < r,$$

$$b < 0 < t$$

Visão ortográfica

- Para arranjar uma imagem com $n_x \times n_y$ pixels em um plano de projeção, com $(r-l) \times (t-b)$, os pixels devem ser dimensionados por meio de $(r-l)/n_x$, para a parte horizontal e $(t-b)/n_y$, para a parte vertical.
- Isso, com um espaço de meio pixel em torno da própria borda, para centralizar a grade de pixels no retângulo de projeção.

Visão ortográfica

- Assim, um pixel na posição (i,j) terá a posição, na imagem renderizada:

$$u = l + (r - l)(i + 0.5) / n_x,$$

$$v = b + (t - b)(j + 0.5) / n_y$$

- Onde (u,v) são as coordenadas da posição do pixel no plano de projeção.

Algoritmo genérico para projeção ortográfica

- Calcule u e v

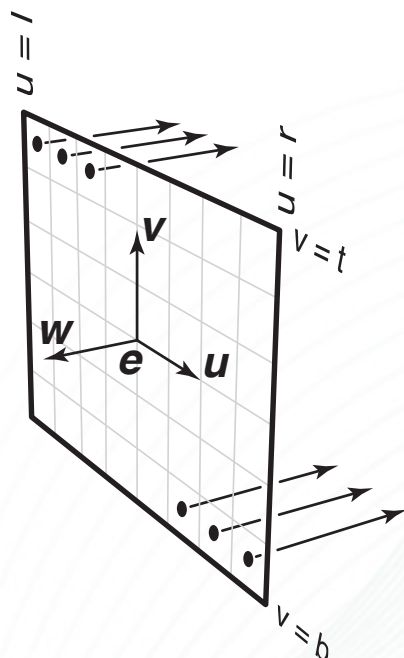
$$u = l + (r - l)(i + 0.5) / n_x$$

$$v = b + (t - b)(j + 0.5) / n_y$$

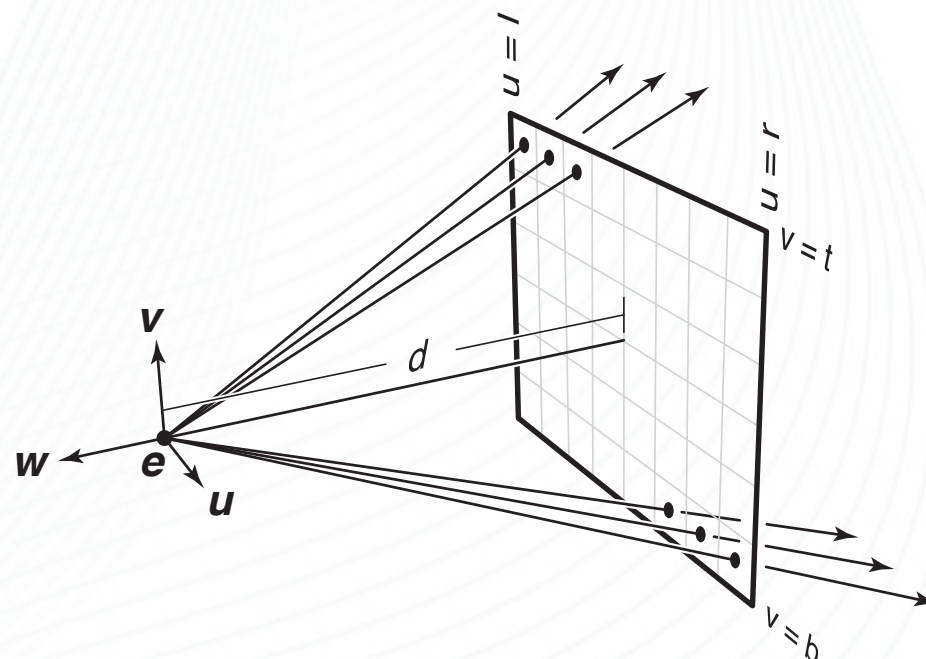
$$\text{raio.direção} = -w$$

$$\text{raio.origem} = e + u * \text{vetor_}u + v * \text{vetor_}v$$

Ortográfica X Perspectiva



Parallel projection
same direction, different origins



Perspective projection
same origin, different directions

Visão em perspectiva

- Na Visão em Perspectiva, todos os raios têm a mesma origem, que é o olho do observador. Assim, as direções, de cada pixel, são diferentes.
- O plano de projeção não está mais posicionado em e . Portanto, existe uma distância d entre o observador e o plano de projeção.
- Essa distância d também pode ser chamada de distância focal de uma câmera real.

Visão em Perspectiva

- A direção de cada raio é definida pela posição do observador em relação a cada pixel da imagem de projeção.
- O algoritmo genérico para a Visão em Perspectiva é similar ao algoritmo da Projeção Ortográfica.

Algoritmo genérico para projeção em perspectiva

- Calcule u e v

$$u = l + (r - l)(i + 0.5) / n_x$$

$$v = b + (t - b)(j + 0.5) / n_y$$

$$\text{raio.direção} = -dw + u * \text{vetor_}u + v * \text{vetor_}v$$

$$\text{raio.origem} = e$$

Sumário

- ~~Algoritmo básico de Ray Tracing~~
- ~~Perspectiva~~
- ~~Computação dos raios de visualização~~
- **Interseção raio-objeto**
- Sombreamento (*Shading*)
- Um programa de Ray Tracing
- Sombras
- Reflexão especular ideal

Interseção raio-objeto

- Interseção raio-esfera
- Interseção raio-triângulo
- Interseção raio-polígono

Interseção raio-objeto

- **Interseção raio-esfera**
- Interseção raio-triângulo
- Interseção raio-polígono

Interseção raio-esfera

$$p(t) = e + t(s - e)$$

- Onde, $p(t)$ representa uma equação de raio.
- Se f é uma função da esfera, então $f(p) = 0$ representa a interseção entre a esfera e o raio. Considere c como o centro da esfera.

$$f(p(t)) = 0$$

$$f(e + td) = 0$$

$$c = (x_c, y_c, z_c)$$

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - R^2 = 0$$

$$(p - c) \cdot (p - c) - R^2 = 0$$

Interseção raio-esfera

- Qualquer ponto que satisfaz a equação mencionada está na esfera.
- Podemos substituir p em função de t e fazermos o rearranjo.

$$(e + td - c) \cdot (e + td - c) - R^2 = 0$$

$$(d \cdot d)t^2 + 2d \cdot (e - c)t + (e - c) \cdot (e - c) - R^2 = 0$$

$$A = (d \cdot d)$$

$$B = 2d \cdot (e - c)$$

$$C = (e - c) \cdot (e - c) - R^2$$

$$At^2 + Bt + C = 0$$

Interseção raio-esfera

- Temos uma função quadrática.
- Por meio do valor de delta (Δ), é possível definir:
 - Se o raio intercepta a esfera em dois pontos: o raio atravessa a esfera;
 - Se o raio passa por um ponto só da esfera: o raio tangencia a esfera; ou
 - Se o raio não tem interseção com a esfera.

Interseção raio-esfera

- Se o raio intercepta a esfera em dois pontos: o raio atravessa a esfera:

$$At^2 + Bt + C = 0$$

$$\Delta = B^2 - 4AC$$

$$\Delta > 0$$

- O pixel receberá a cor da esfera.

Interseção raio-esfera

- Se o raio passa por um ponto só da esfera: o raio tangencia a esfera.

$$At^2 + Bt + C = 0$$

$$\Delta = B^2 - 4AC$$

$$\Delta = 0$$

- O pixel receberá a cor da esfera.

Interseção raio-esfera

- Se o raio não tem interseção com a esfera.

$$At^2 + Bt + C = 0$$

$$\Delta = B^2 - 4AC$$

$$\Delta < 0$$

- O pixel receberá a cor do fundo da cena.

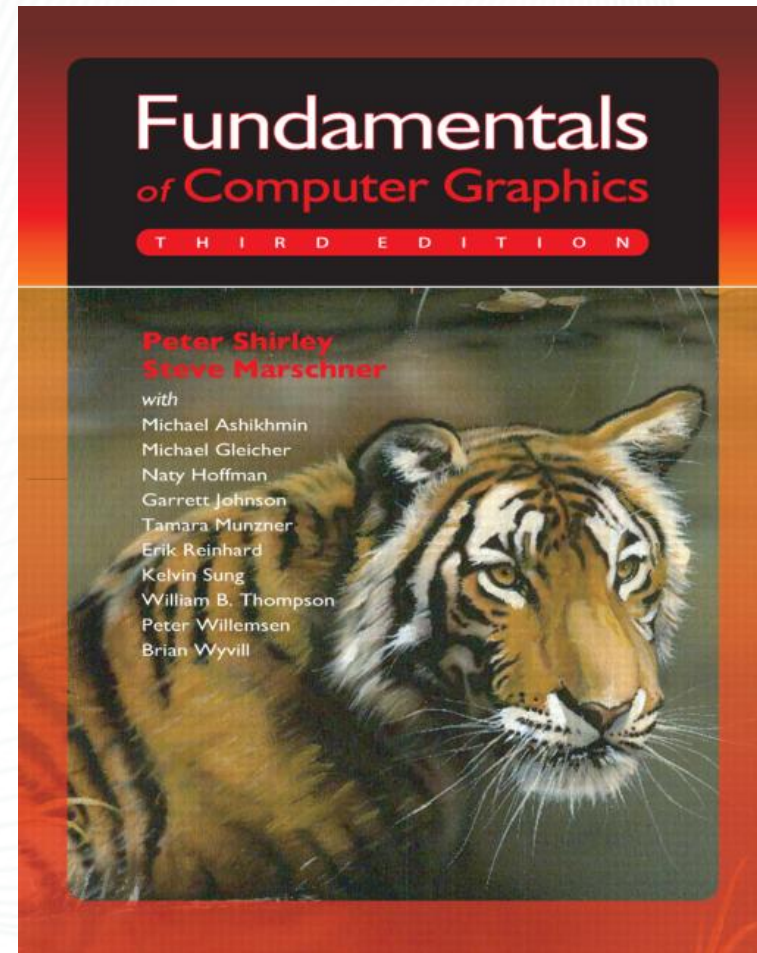
Sumário

- ~~Algoritmo básico de Ray Tracing~~
- ~~Perspectiva~~
- ~~Computação dos raios de visualização~~
- **Interseção raio-objeto (continua)**
- Sombreamento (*Shading*)
- Um programa de Ray Tracing
- Sombras
- Reflexão especular ideal

Aula de hoje

Shirley, Peter, Michael Ashikhmin, and Steve Marschner. Fundamentals of computer graphics. CRC Press, 3rd Edition, 2009.

- **Capítulo 4**



Atividade 01

- Será passada na próxima quarta-feira, dia 12/10.
- As informações da atividade estarão descritas na especificação da mesma.

Fim da Aula 07

André Luiz Brandão