



# Compilador de Funções de Distribuição de Refletância Bidirecional descritas em $\text{\LaTeX}$ para Linguagem de Shading

Everton Santos de Andrade Júnior  
Orientador(a): Dra. Beatriz Trinchão Andrade

Universidade Federal de Sergipe

Dez / 2024

# Sumário

① Introdução

② Conceitos

# Primeiramente

Na computação gráfica, a representação realista de cenas tridimensionais depende fortemente da modelagem da luz e dos materiais que compõem os objetos na cena. A interação da luminosidade incidente com esses materiais é crucial para a geração de imagens fiéis à realidade. Uma forma fundamental para modelar essa interação é por meio das funções de distribuição de refletância bidirecional, conhecidas como BRDFs (do inglês, *Bidirectional Reflectance Distribution Functions*).

# BRDFs e sua importância

- As BRDFs calculam a proporção entre a energia luminosa que atinge um ponto na superfície e como essa energia é:
  - Refletida,
  - Transmitida,
  - Ou absorvida.
- Referência: **Pharr, Jakob e Humphreys (2016) [pbr]**.

# Shaders e implementação

- Na renderização, BRDFs são implementadas por *shaders*, programas especializados executados na GPU.
- APIs gráficas permitem programar esses shaders em diferentes etapas do processo de renderização ( OpenGL).
- Com os shaders, cada objeto renderizado pode ter sua aparência configurada por meio de códigos que implementam BRDFs específicas.

# Motivação

- Programar *shaders* requer conhecimento técnico especializado.
- Essa barreira dificulta a criação e exploração de BRDFs por profissionais de áreas não técnicas em programação.
- Surge a necessidade de ferramentas mais acessíveis para facilitar essa tarefa.

# Motivação

- No meio acadêmico, BRDFs são descritas usando fórmulas escritas em  $\text{\LaTeX}$ .
- Proposta: desenvolver um compilador que traduza BRDFs escritas em  $\text{\LaTeX}$  para *shaders*.
- Essa solução torna a criação de efeitos visuais mais acessível e democrática.

# Texto em tópicos numerados

Lorem ipsum dolor sit amet, consectetur adipiscing elit:

- ① Lorem ipsum dolor sit amet.
- ② Lorem ipsum dolor sit amet.



# Texto em tópicos

Lorem ipsum dolor sit amet, consectetur adipiscing elit:

- Lorem ipsum dolor sit amet.
- Lorem ipsum dolor sit amet.

# Objetivo

- Desenvolver um compilador que traduz BRDFs escritas em  $\text{\LaTeX}$  para código *shader* em OpenGL.
- O *shader* gerado deve reproduzir ou aproximar as características de reflexão da BRDF original.
- Considerar as limitações da linguagem de shading, como a representação discreta de dados.

# Radiometria: Introdução

- Estuda a interação da luz com superfícies na computação gráfica.
- Quantifica energia luminosa:
  - Brilho da fonte de luz.
  - Iluminação e refletância da superfície.
- Fundamenta a renderização realista de cenas tridimensionais.

# Energia Radiante

- Quantifica a energia total dos fótons atingindo uma superfície.
- Fórmula:

$$A_r = \frac{hc}{\lambda}$$

onde:

- $h$ : Constante de Planck.
- $c$ : Velocidade da luz.
- $\lambda$ : Comprimento de onda.

# Fluxo Radiante e Irradiância

- Fluxo Radiante: Energia por unidade de tempo ( $J/s$ ):

$$\Phi = \frac{A_r}{t}$$

- Irradiância: Fluxo radiante por unidade de área:

$$E = \frac{d\Phi}{dA}$$

- Quantifica os impactos de fótons em uma superfície.

# Radiância

- Radiância ( $L$ ): Fluxo radiante por unidade de área e ângulo sólido:

$$L = \frac{d^2\Phi}{dA d\omega \cos(\theta)}$$

- Onde:
  - $d\omega$ : Ângulo sólido (em sr).
  - $\theta$ : Ângulo entre direção de incidência e normal da superfície.

# Visualização da Radiância

- Radiância considera direção específica no hemisfério sobre a superfície.

Figure: Visualização da radiância em uma direção específica do hemisfério.

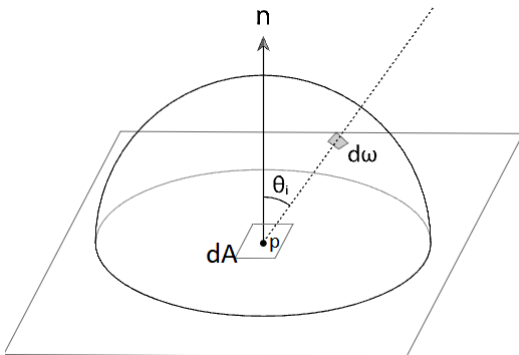


Figure: Ângulo sólido  $s$  do objeto B visto pelo ponto  $p$ .

# Equação de Renderização

- A radiância de saída ( $L_o$ ) em um ponto é calculada por:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{\Omega} f_r(\omega_i, \omega_o) L_i(p, \omega_i) \cos(\theta_i) d\omega_i$$

- Termos:
  - $L_e$ : Radiância emitida.
  - $f_r$ : BRDF (refletância).
  - $L_i$ : Radiância incidente.



# BRDF: Função de Refletância Bidirecional

- BRDF ( $f_r$ ) descreve como a luz reflete em diferentes direções:

$$f_r(\omega_i, \omega_o) = \frac{dL_o(\omega_o)}{L_i(\omega_i) \cos(\theta_i) d\omega_i}$$

- Propriedades:
  - Positividade:  $f_r \geq 0$ .
  - Reciprocidade de Helmholtz:  $f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i)$ .
  - Conservação de energia:  $\int_{\Omega} f_r \cos(\theta_i) d\omega_i \leq 1$ .

# Introdução aos Shaders e ao *Pipeline* de GPU

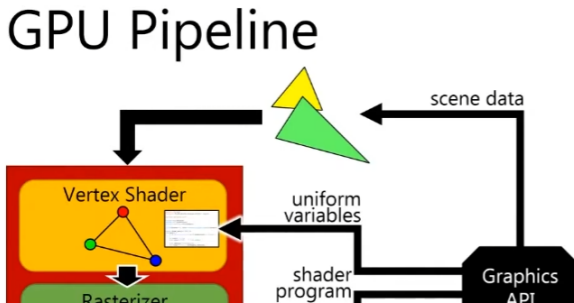
- **Shaders:** Pequenos programas executados na GPU para simular a interação da luz com superfícies.
- ***Pipeline* de GPU:** Fluxo de dados da CPU para GPU que transforma vértices e gera a imagem final.
- Principais componentes:
  - Vértices: Pontos das primitivas geométricas (ex.: triângulos).
  - Fragmentos: Elementos discretos na tela (pixels) processados.
- API OpenGL: Facilita o acesso às funções da GPU.

# Estrutura do *Pipeline* de GPU

- **Etapas principais:**

- 1 **Shader de Vértice:** Processa e transforma vértices.
- 2 **Rasterização:** Gera fragmentos a partir de primitivas.
- 3 **Shader de Fragmento:** Determina a cor final dos fragmentos.

- Fluxo de dados: CPU ⇄ GPU ⇄ Shaders ⇄ Imagem final.
- *Shaders* adicionais:
  - Tesselação: Subdivide superfícies em triângulos menores.
  - Geometria: Manipula primitivas antes da rasterização.



# Shader de Vértice

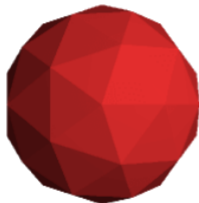
- **Função principal:** Transformar vértices e preparar dados para o próximo estágio.
- Tarefas:
  - Aplicação de transformações (ex.: rotação, projeção).
  - Transmissão de dados (ex.: normais) ao *shader* de fragmento.

## Exemplo GLSL:

## Shader de Fragmento e Níveis de *Shading*

- **Shader de Fragmento:** Processa fragmentos e define cores dos pixels.
- Interpolação: Normais e vértices são interpolados com coordenadas baricêntricas.
- Comparação:
  - **Vertex Shading:** Menor precisão, menos processamento.
  - **Fragment Shading:** Maior precisão, transições mais suaves.

Shading para cada vértice



Shading para cada fragmento



# Introdução

- BRDF (Bidirectional Reflectance Distribution Function): Função que descreve como a luz é refletida por uma superfície.
- Apresentaremos modelos clássicos de BRDFs:
  - BRDF Pura Especular.
  - BRDF Difusa Ideal.
  - BRDF Brilhante.
  - BRDF Retro-Refletora.

## BRDF Pura Especular

- Superfície reflete luz apenas em uma direção.
- Segue a Lei da Reflexão:

$$f(\omega_i, \omega_o) = k_s \cdot \delta(\omega_i - \omega_o)$$

- Materiais típicos: metal polido, vidro.

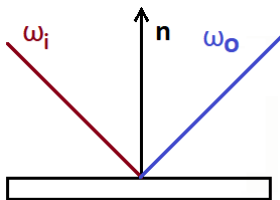


Figure: Reflexão especular. Raio incidente em vermelho e raio refletido em azul.

# BRDF Difusa Ideal

- Reflexão uniforme em todas as direções.
- Função BRDF:

$$f(\omega_i, \omega_o) = \frac{\rho_d}{\pi} \cdot \cos \theta$$

- Exemplos: tinta fosca, papel.

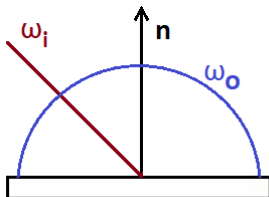


Figure: Reflexão difusa. Raios refletidos independem do ângulo de entrada.



# BRDF Brilhante

- Combina reflexões especulares e difusas.
- Modelo típico: Blinn-Phong.

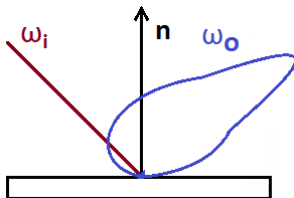


Figure: Reflexão *glossy*.

# BRDF Retro-Refletora

- Redireciona a luz de volta à fonte.
- Utilizado em superfícies como placas de trânsito e sinais de segurança.

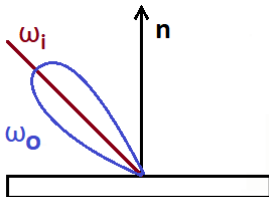


Figure: Reflexão retro-refletora.

Fim!