

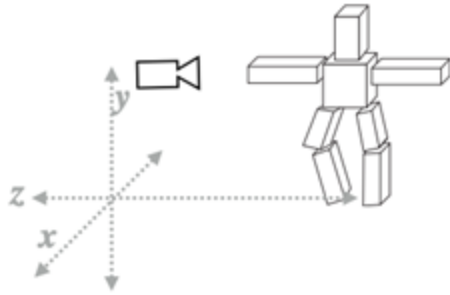
Computação Gráfica

Aula 14: Materiais e Iluminação

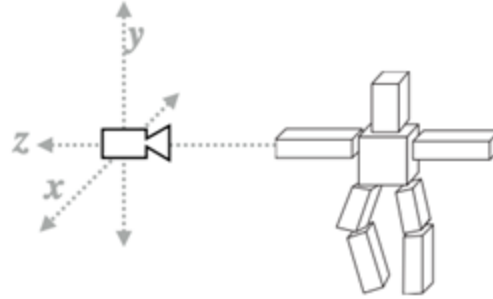
Computação Gráfica

Cálculo de Iluminação

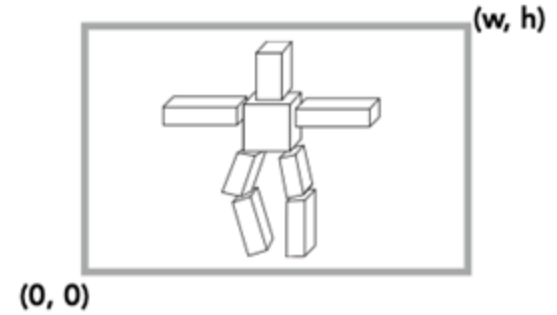
O que vimos até agora



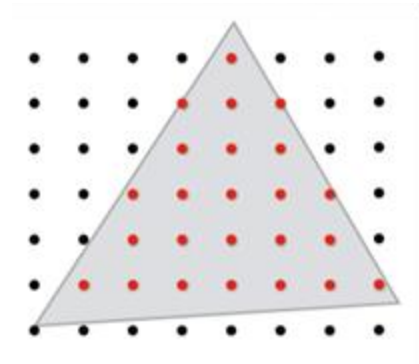
Posicionando câmeras e objetos no mundo



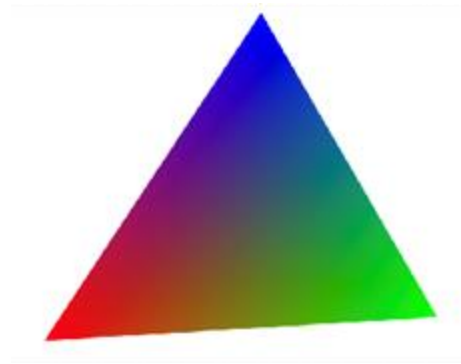
Calculando a posição dos objetos em relação a câmera



Projetando os objetos na tela



Amostrando a cobertura dos triângulos



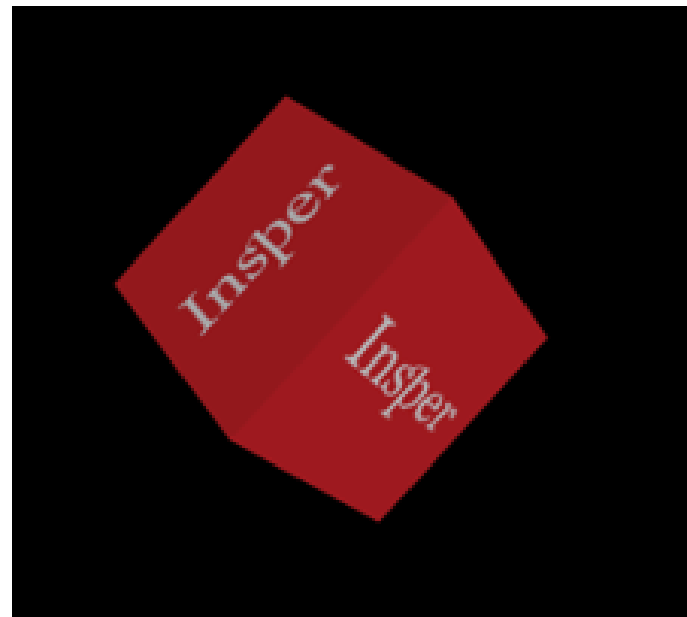
Interpolando os atributos do triângulo



Amostrando texturas mapeada

Iluminação

Iluminação permite perceber a característica de 3D de um objeto



Quais outros efeitos existem?



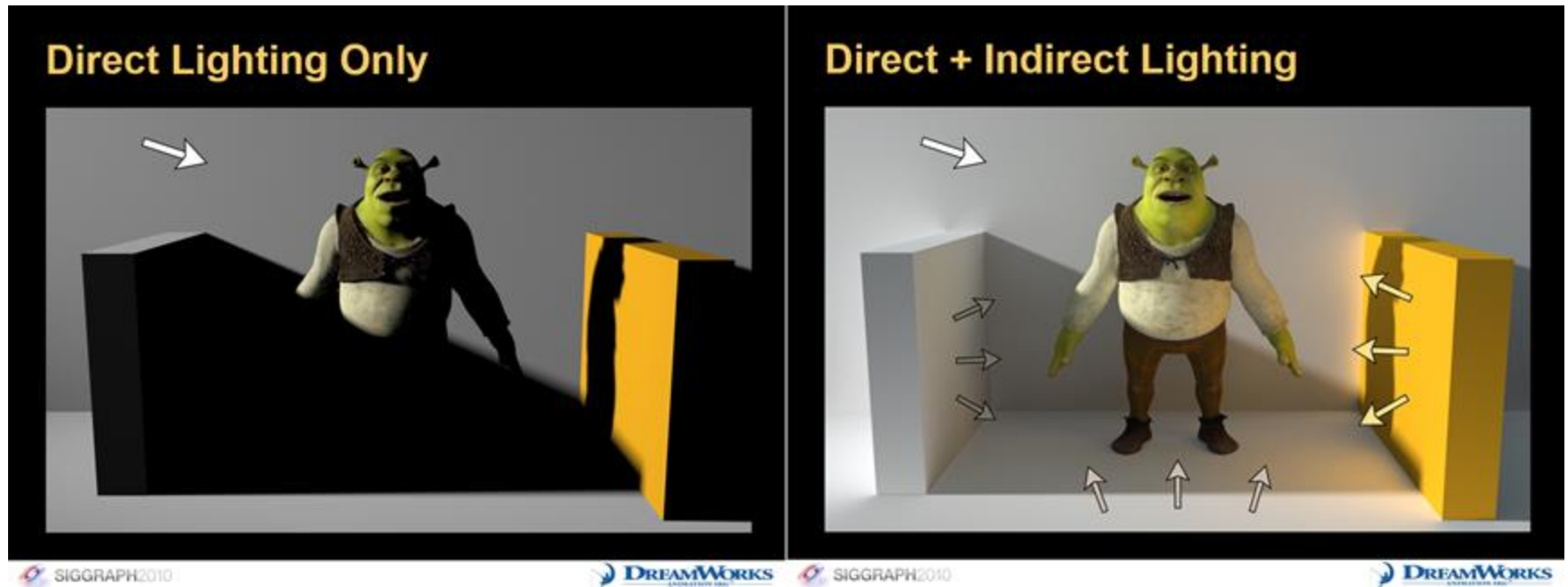
Créditos: Bertrand Benoit. "Sweet Feast," 2009. [Blender /VRay]

Quais outros efeitos existem?



Créditos: Giuseppe Albergo. “Colibri” [Blender]

Iluminação Local vs. Global



Percepções das Observações das Renderizações





Aplicação ao cálculo de iluminação

Fontes de Luz

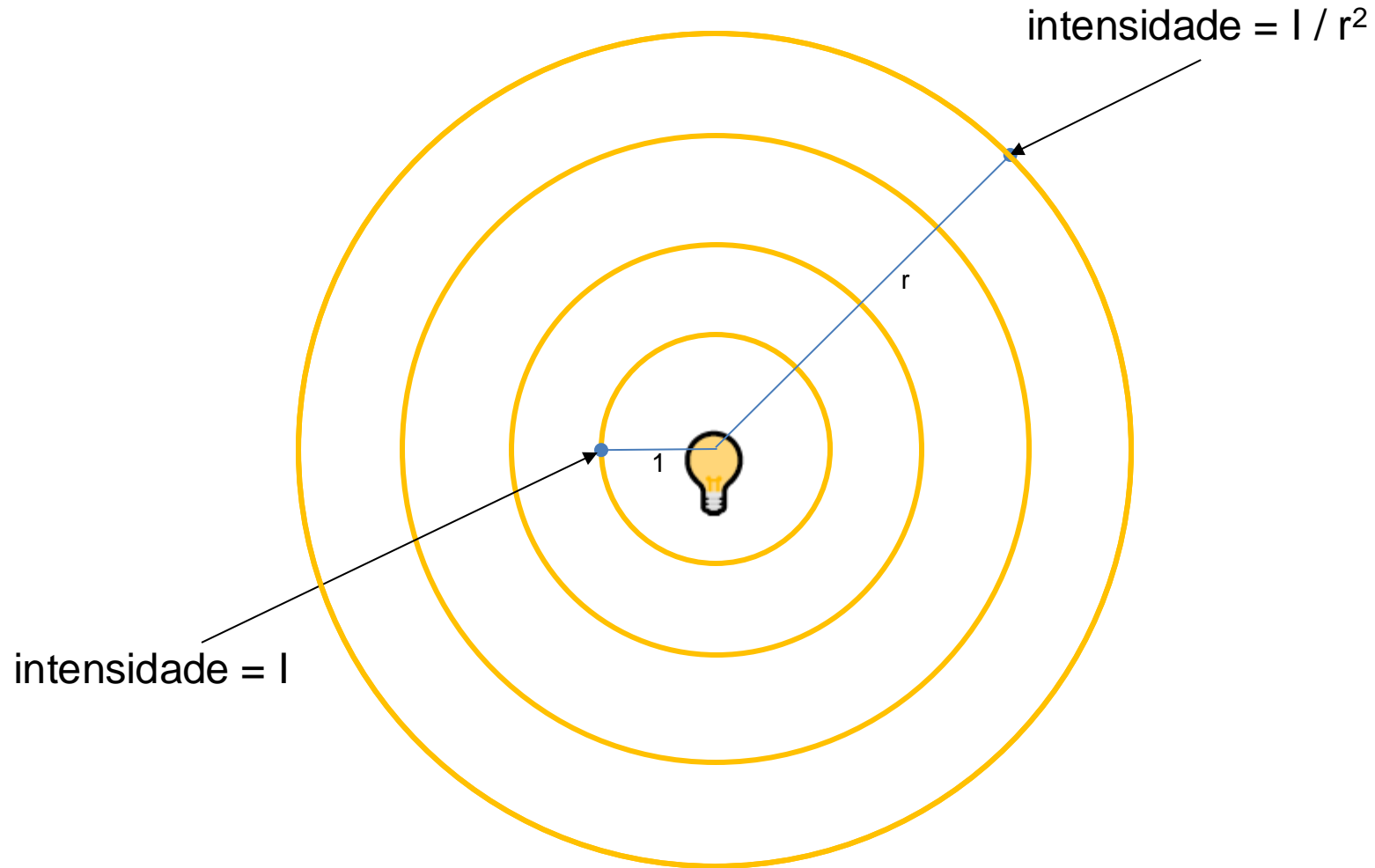
- Pontual: *ponto único que emana em todas as direções*
- Direcional: *usadas para luz do sol ou luar.
(Frequentemente como luz principal)*
- Spot: *foco em um único local*
- Ambiente: *se espalham por toda parte, igualmente*
- Área: *emanam de uma área toda.*

Luz Pontual

- Emite luz para todas as direções
- Definida por uma posição no espaço
- Não possui uma direção que caracterize ela
- Intensidade cai com o quadrado da distância



Decaimento da Luz Pontual



Luz Direcional

- Todos os raios de luz paralelos
- Posição da fonte de luz é irrelevante
- Os raios de luz apontam para uma direção específica
- Intensidade constante, independente da distância



Luz Spot

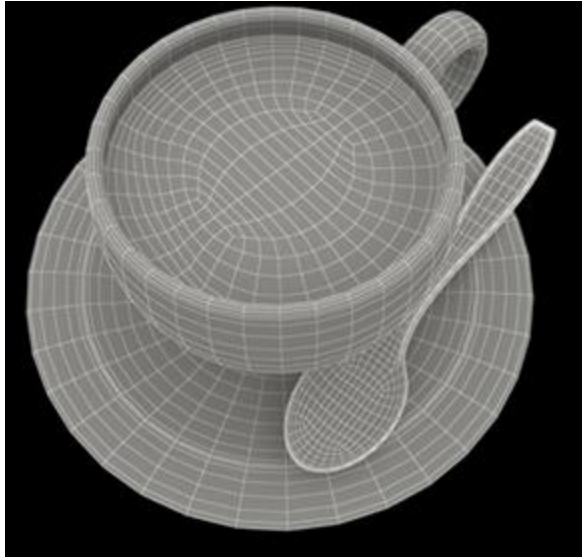
- Todos os raios partem de um ponto no espaço
- Luz dispersa em um formato de cone em uma certa direção
- Pode ter efeitos de penumbra
- Intensidade depende da distância



Luz Ambiente

- Na prática é um truque de iluminação
- As superfícies recebem essa luz de forma uniforme

O que são Materiais em Computação Gráfica



Modelo da xícara



Renderizada



Renderizada

Material: difuso



Material: plástico



Material: pintura semi-gloss vermelha



Material: Tinta laca mística Ford



Material: espelhado



Material: dourado



Propriedade dos Materiais

Principais propriedades:

Cor Difusa

Cor Especular

Brilho

“Cor Ambiente”



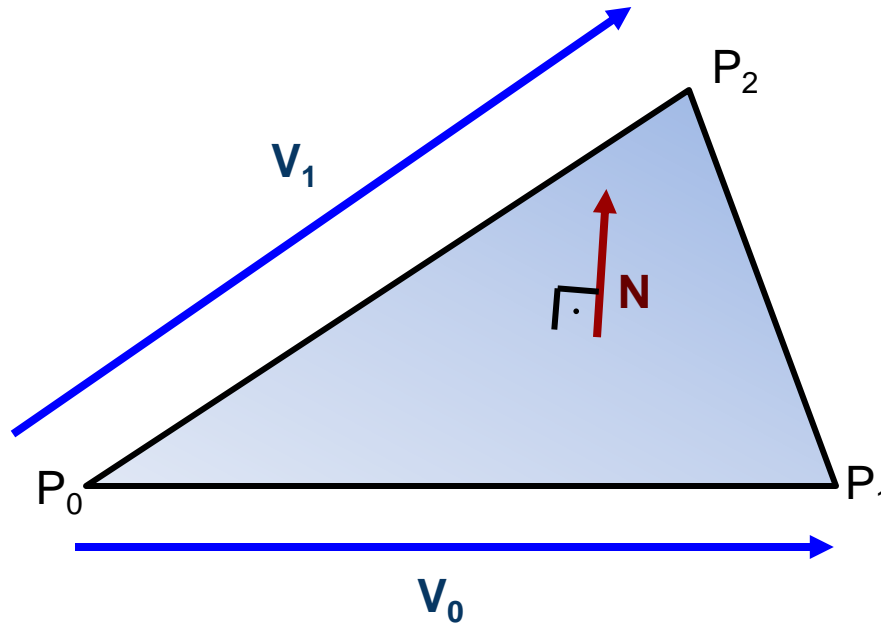
Definindo Vetores Normais por Face

$$\vec{V}_0 = P_1 - P_0$$

$$\vec{V}_1 = P_2 - P_0$$

$$\vec{N} = \vec{V}_0 \times \vec{V}_1$$

$$\hat{N} = \frac{\vec{V}_0 \times \vec{V}_1}{\|\vec{V}_0 \times \vec{V}_1\|}$$



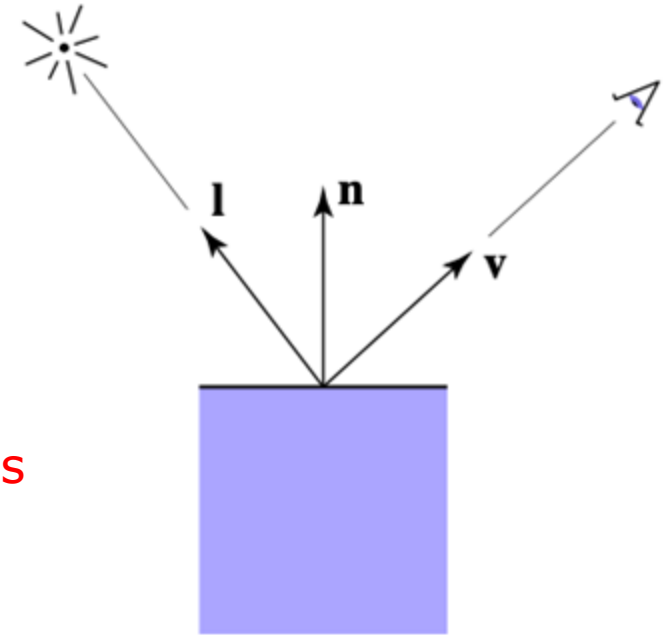
Cálculo Local

Calcular a luz refletida em direção à câmera

Parâmetros:

- Direção do visualizador (v)
- Normal da Superfície (n)
- Direção da luz (l)
para cada uma das luzes existentes
- Parâmetros da superfície (cor, brilho, ...)

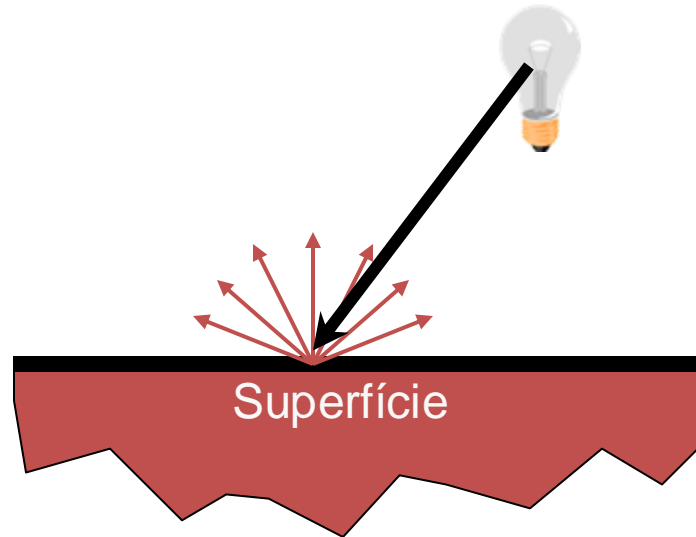
Cuidado para sempre trabalhar com todos os vetores normalizados.



Reflexão Difusa

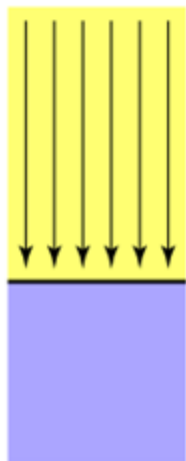
A superfície reflete igualmente em todas as direções

- Exemplos: carvão, argila

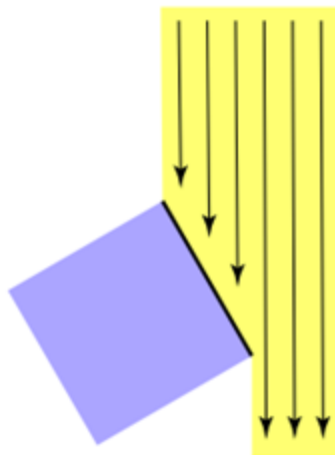


Reflexão Difusa

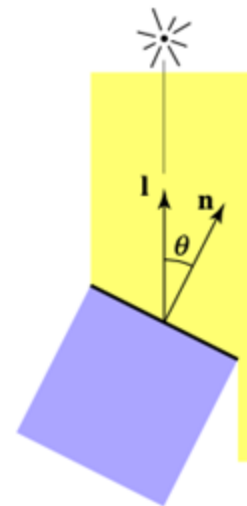
Na reflexão difusa a luz é espalhada uniformemente em todas as direções
A cor da superfície é a mesma olhando de qualquer direção



O topo do cubo recebe uma certa quantidade de luz



O topo do cubo a 60° recebe metade da quantidade de luz



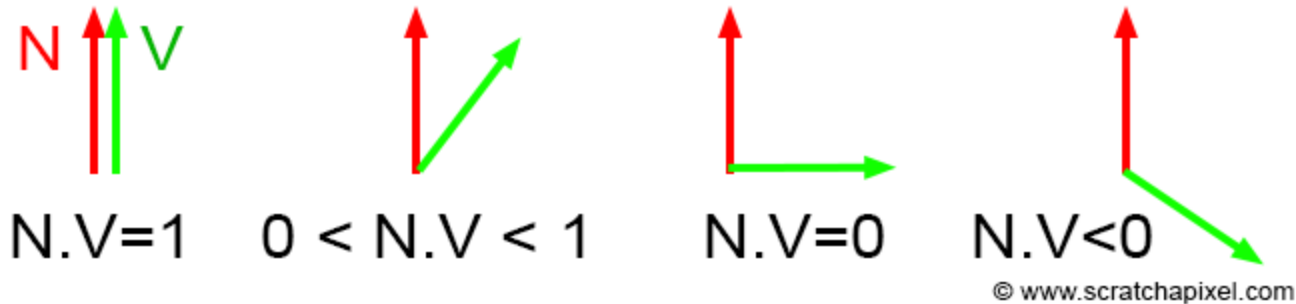
Em geral, a luz por quantidade de área é proporcional a:

$$\cos \theta = \mathbf{l} \cdot \mathbf{n}$$

"Em óptica, a lei do cosseno de Lambert afirma que a intensidade luminosa observada em uma superfície com reflexão difusa ideal é diretamente proporcional ao cosseno do ângulo θ entre a direção de incidência da luz e a normal da superfície, reta perpendicular a esta."

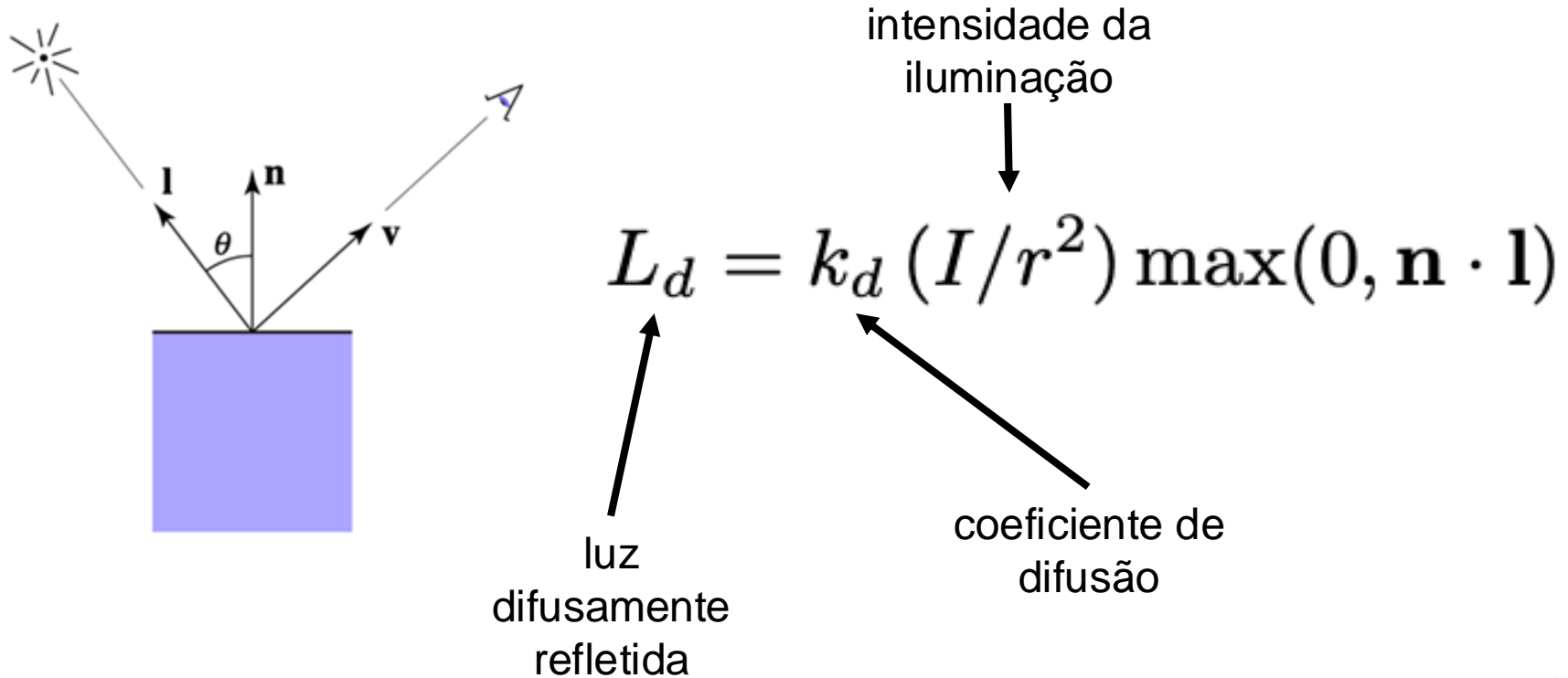
Superfície Faceando

Podemos usar o produto escalar de dois vetores para descobrir o quanto uma superfície está "faceando" algo (luz, ponto de vista)



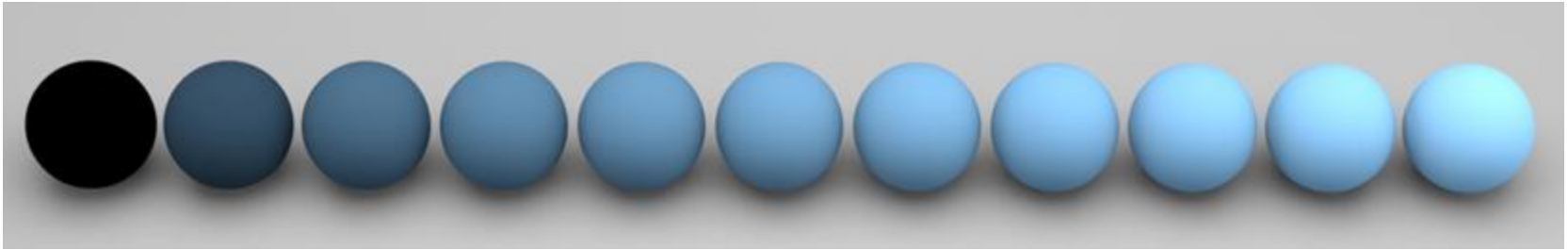
Propriedades Lambertianas (Difusas)

aparência da cor e brilho independente da direção de visualização



Propriedades Lambertianas (Difusas)

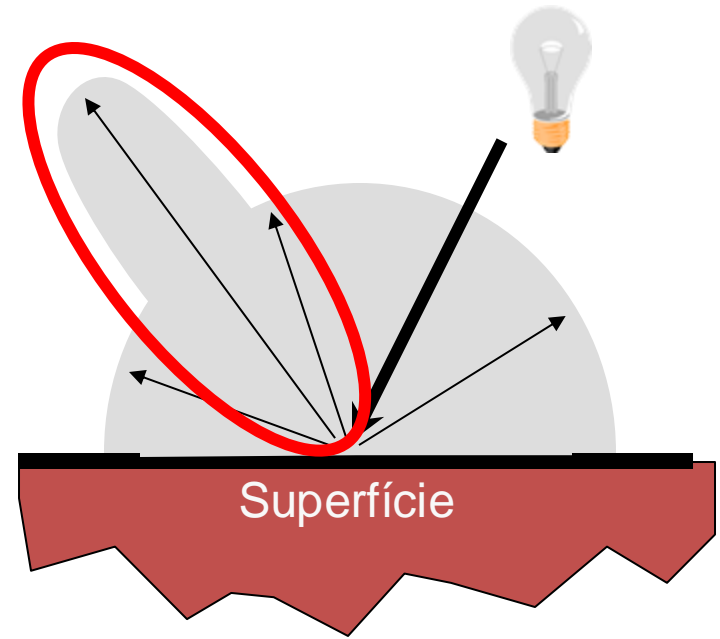
Leva a uma aparência fosca



K_d variando de 0 até 1

Reflexão Especular

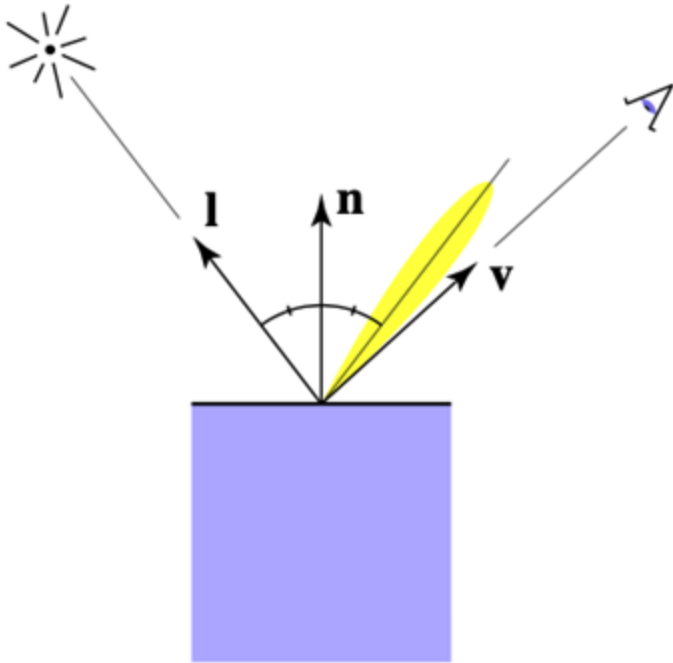
- Reflexão é mais intensa no ângulo que espelha o raio
 - Exemplos: espelhos, metais



Reflexão Especular (Blinn / Phong)

Intensidade depende da direção de visualização

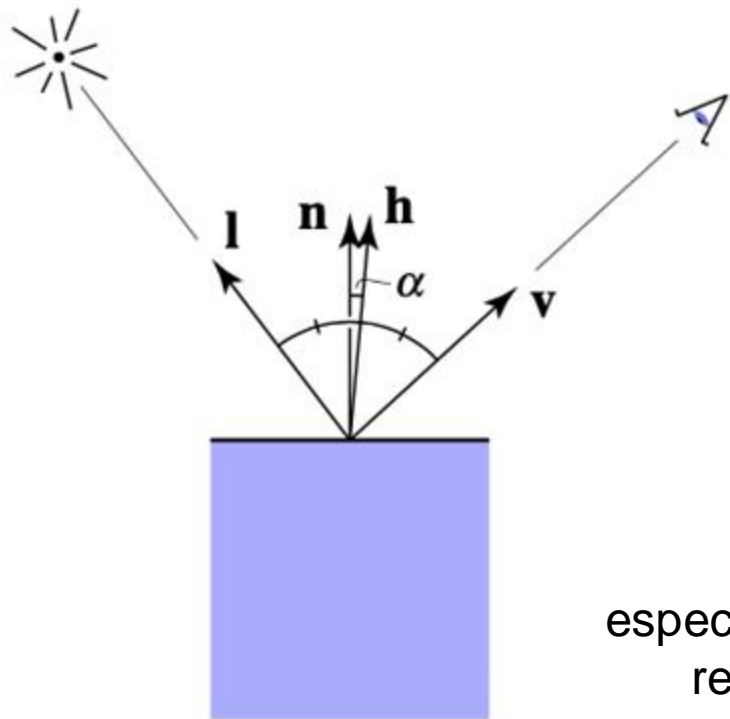
Maior intensidade de brilho quando o ângulo de reflexão é o mesmo do ângulo de incidência em relação a normal da superfície.



Reflexão Especular (Blinn / Phong)

Quando próximo ao ângulo de incidência, maior brilho.

O quão "próximo" é medido pelo produto escalar dos vetores unitários.



$$h = \text{bissetriz}(v, l)$$

$$= \frac{v + l}{\|v + l\|}$$

$$L_s = K_s \left(\frac{I}{r^2} \right) \max(0, \cos \alpha)^p$$

luz
especularmente
refletida

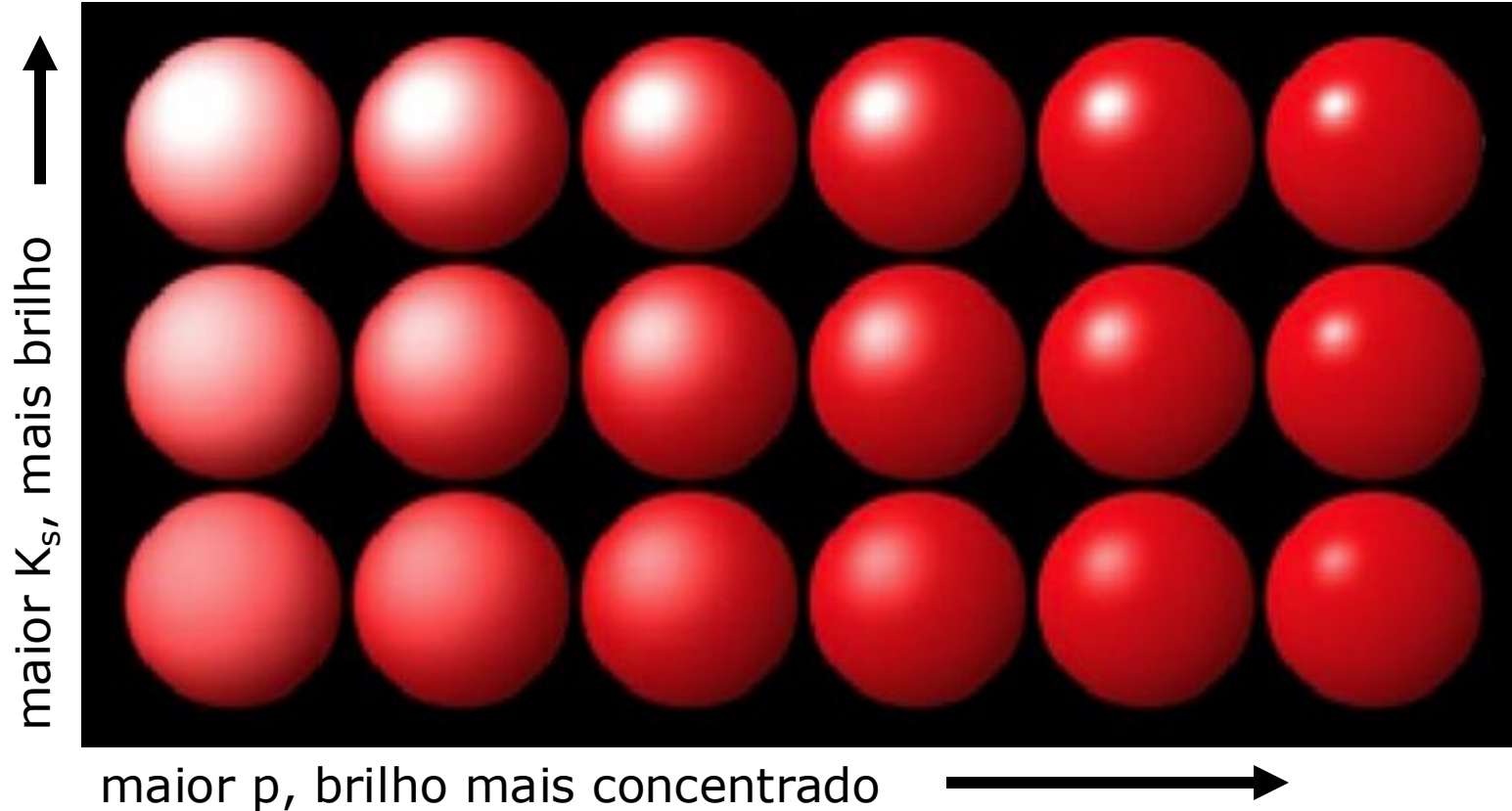
$$= K_s \left(\frac{I}{r^2} \right) \max(0, n \cdot h)^p$$

coeficiente de
especularidade

expoente de
reflexão especular
Insper

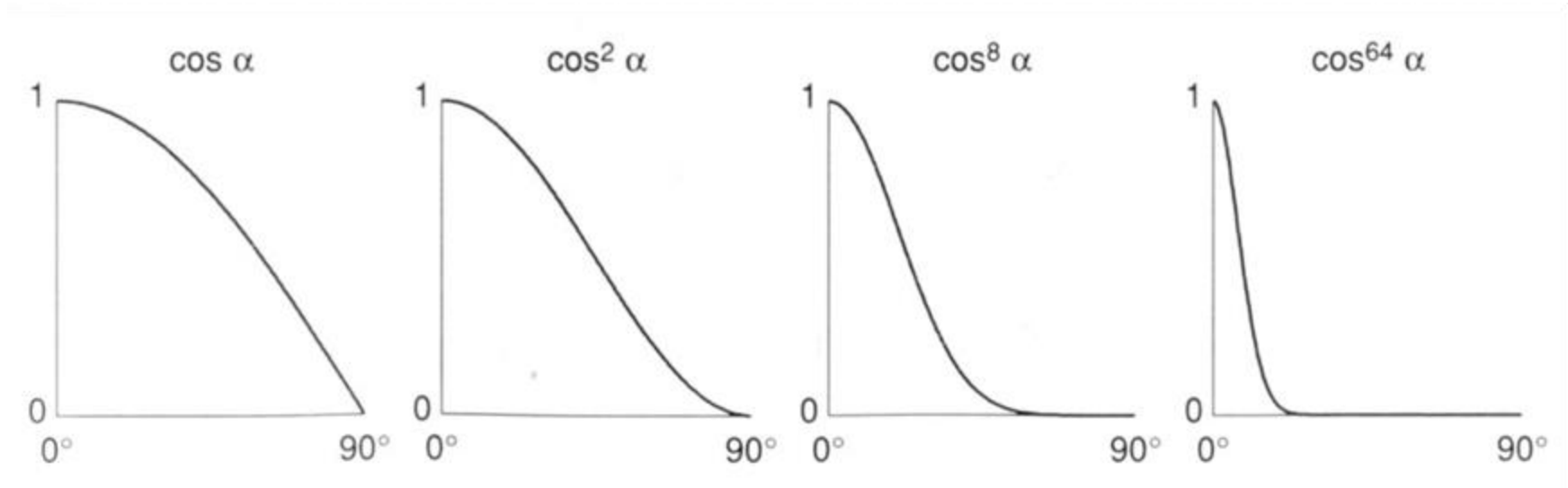
Reflexão Especular (Blinn / Phong)

$$L_s = K_s \left(\frac{I}{r^2} \right) \max(0, n \cdot h)^p$$



Região Saturada de Brilho

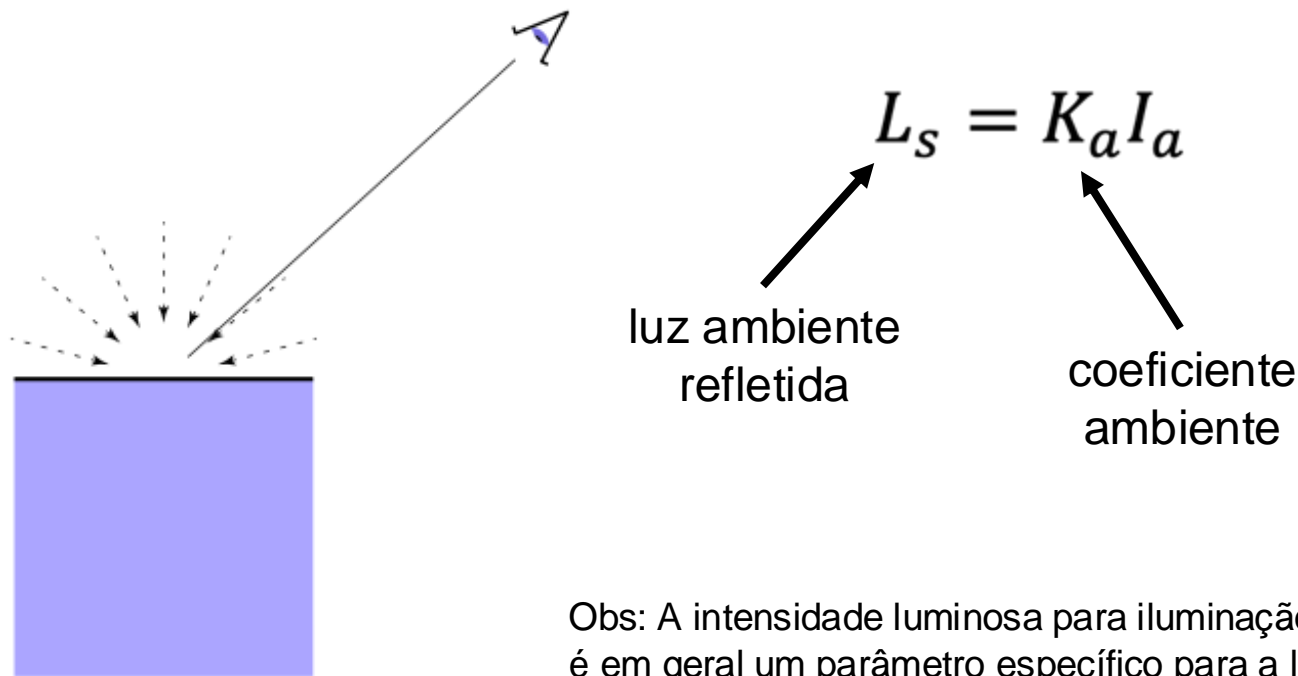
Aumentar o expoente p irá estreitar a região saturada de brilho



Reflexão/Iluminação Ambiente

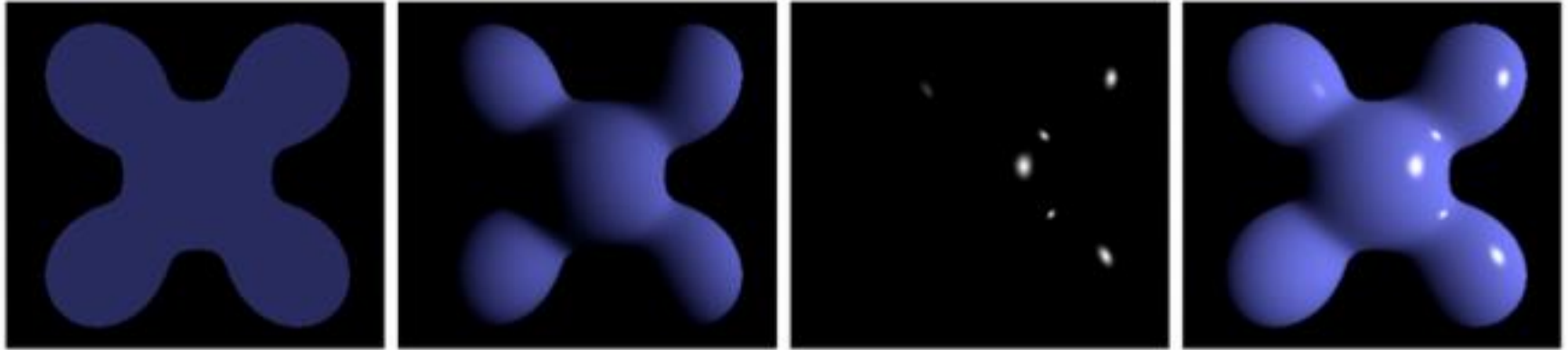
Reflexão/Iluminação que não depende de nada

Adiciona uma cor constante a superfície para compensar qualquer falta de iluminação, preenchendo regiões escuras



Obs: A intensidade luminosa para iluminação ambiente é em geral um parâmetro específico para a luz.

Modelo de Reflexão Blinn-Phong



Ambiente + Difusa + Especular = Reflexão Phong

$$\begin{aligned} L &= L_a + L_d + L_s \\ &= k_a I_a + k_d (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{l}) + k_s (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{h})^p \end{aligned}$$

Shading em Malhas de Triângulos

Infelizmente não conheço nenhuma tradução adequada. A tradução usada é "sombreamento", mas fica estranha. Prefiro "tonalização" mas não é muito usado

shading *noun*

 /'ʃeɪdɪŋ/

 /'ʃeɪdɪŋ/

- 1 ★ [uncountable] the use of colour, pencil lines, etc. to give an impression of light and shade in a picture or to emphasize areas of a map, diagram, etc.

Shading em Triângulos, Vértices e Pixels

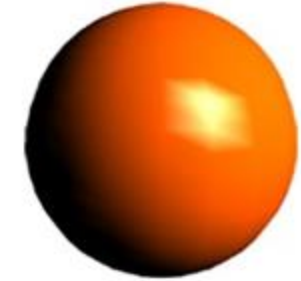
Shading por triângulo (flat shading)

- A face do triângulo é plana (um vetor normal)
- Nada bom para superfícies suaves



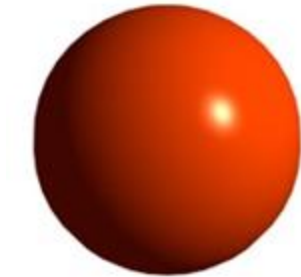
Shading por vértice ("Gouraud" shading)

- Interpolando as cores através do triângulo
- Cada vértice possui um vetor normal



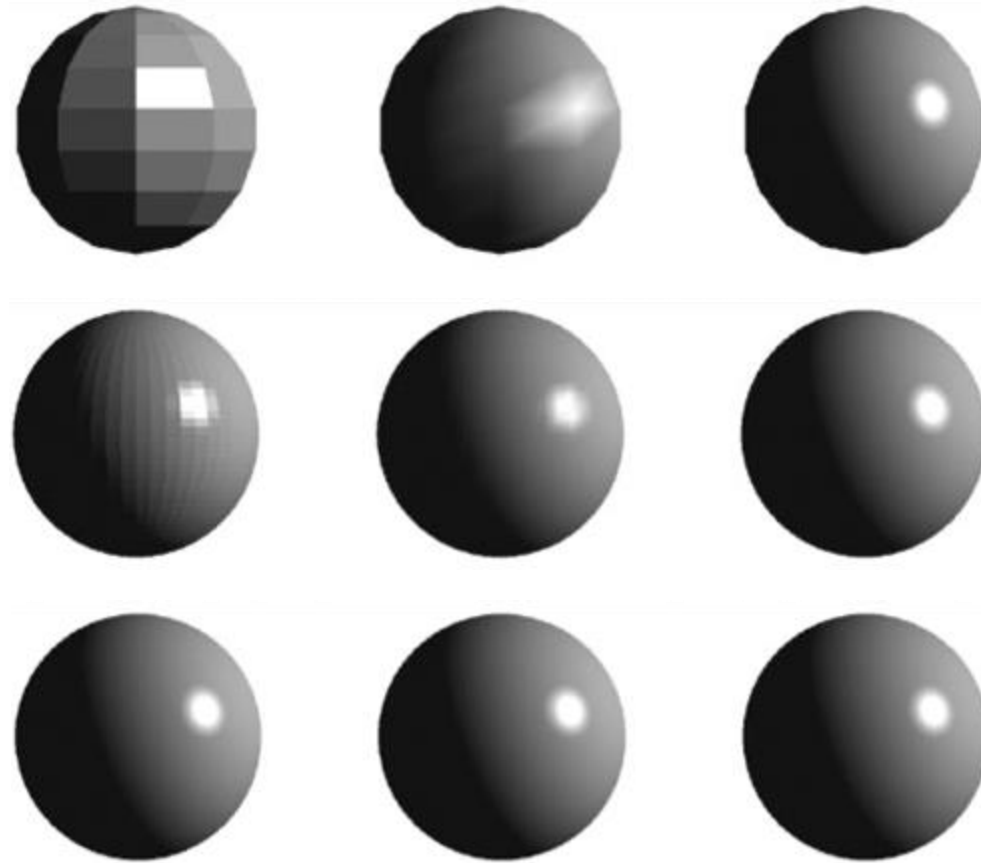
Shading por pixel ("Phong" shading)

- Interpolando a normal através do triângulo
- Calcula em cada pixel qual seria a cor



Shading em Triângulos, Vértices e Pixels

Quantidade de vértices



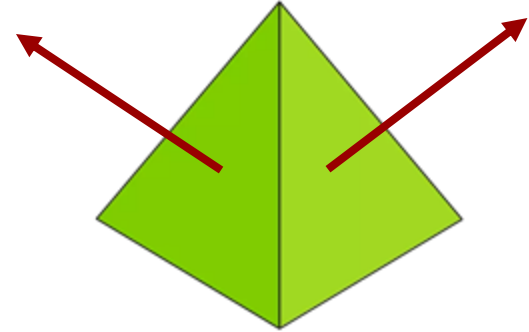
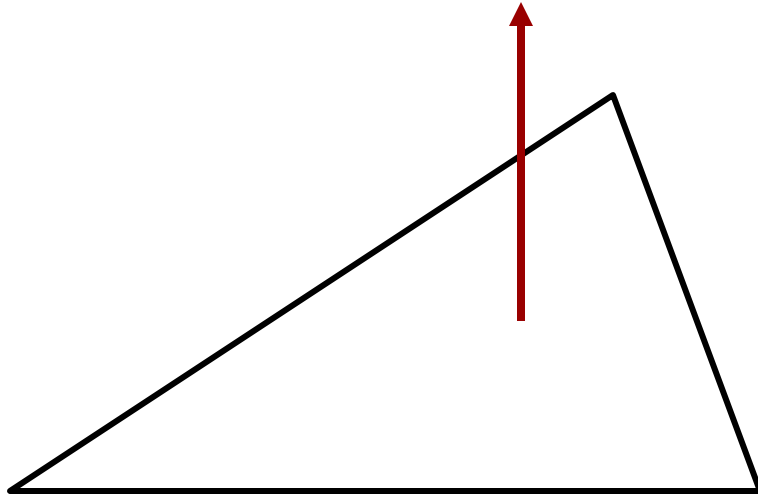
Tipo de Shading:

Flat

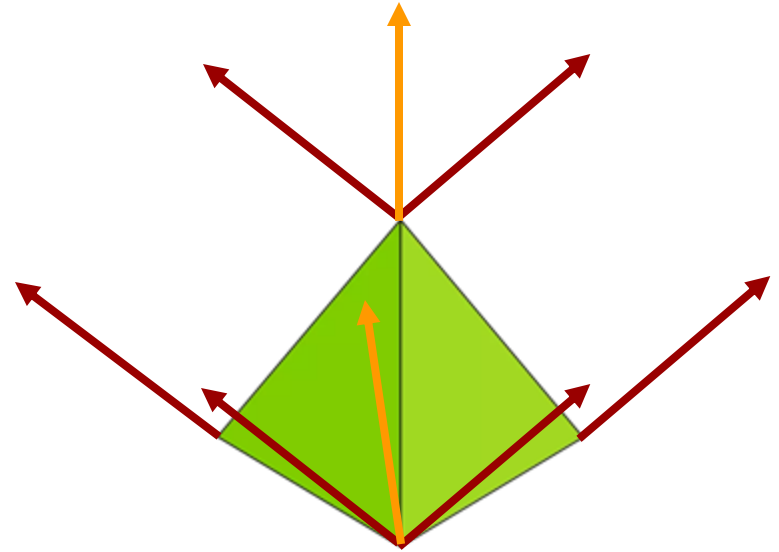
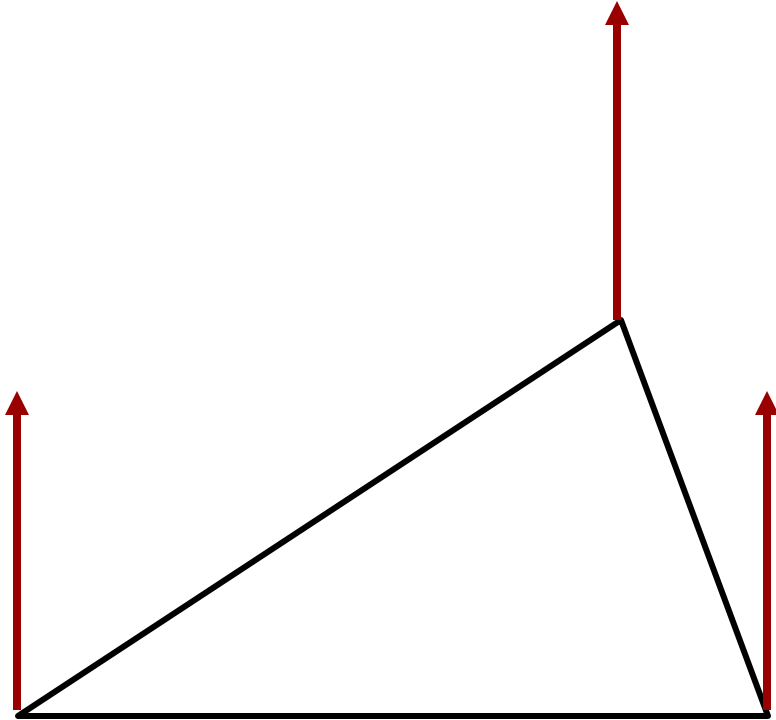
Gouraud

Phong

Normais por Face



Normais por Vértice



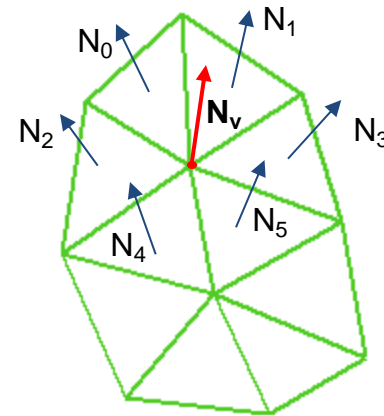
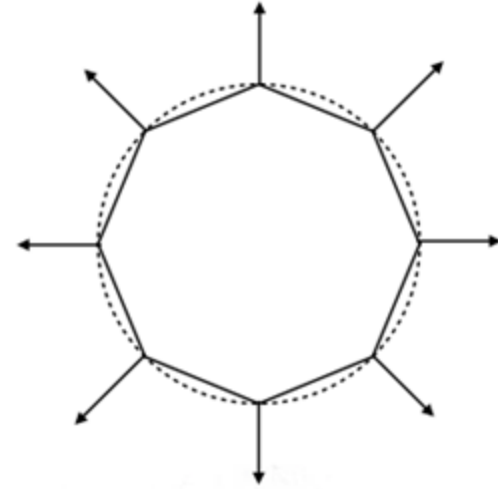
Definindo Vetores Normais por Vértice

Melhor obter normais da geometria desejada, por exemplo: uma esfera

Senão inferir as normais das faces dos triângulos

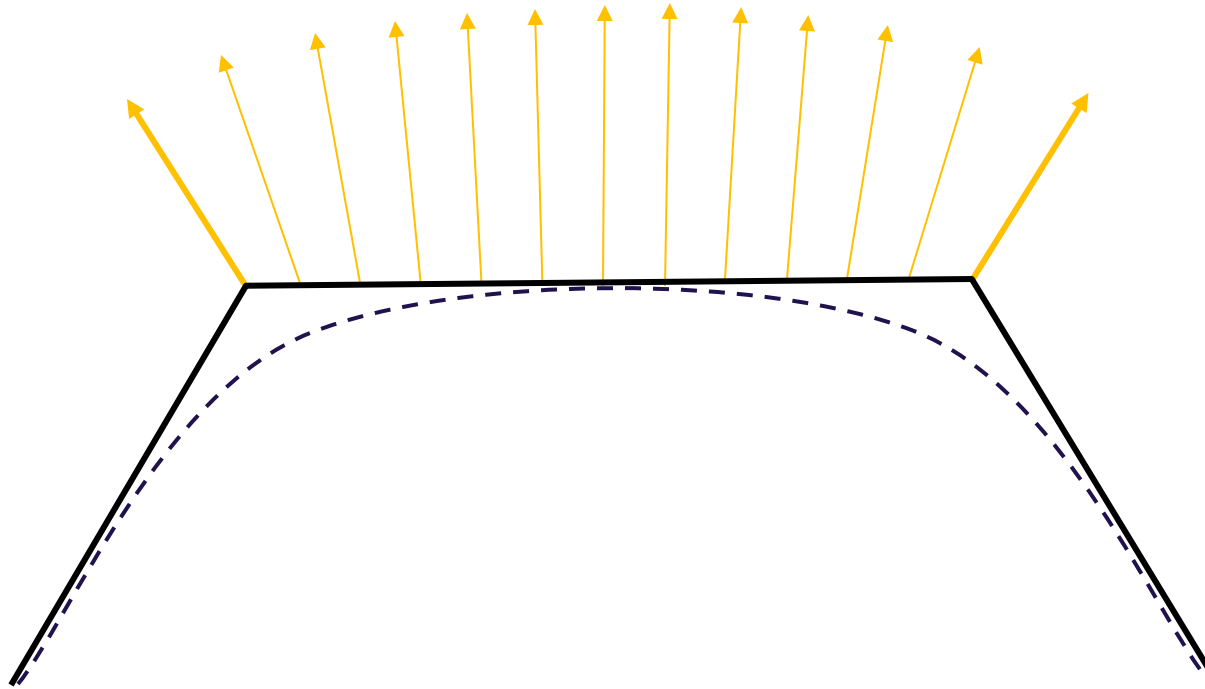
Proposta mais simples: Médias das normais das faces ao redor

$$N_v = \frac{\sum_i N_i}{\|\sum_i N_i\|}$$



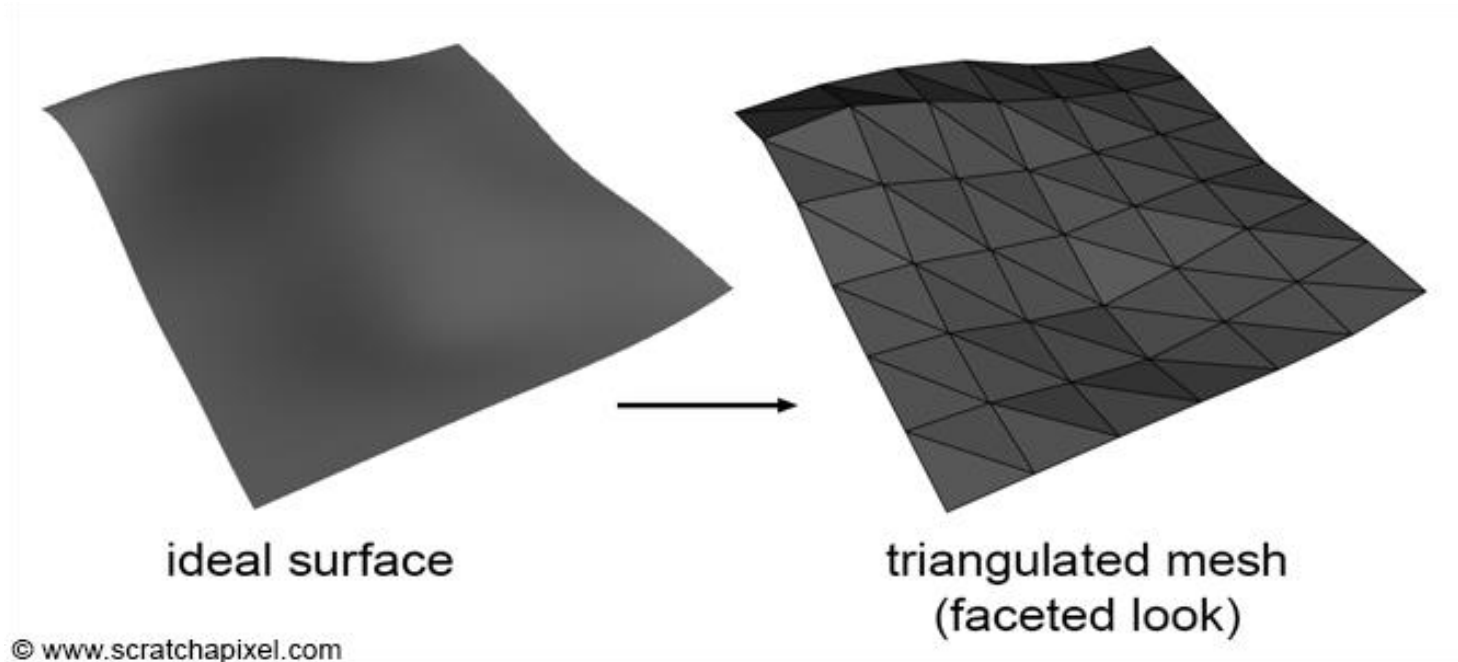
Definindo Vetores Normais por Vértice

Interpolação baricêntrica das normais

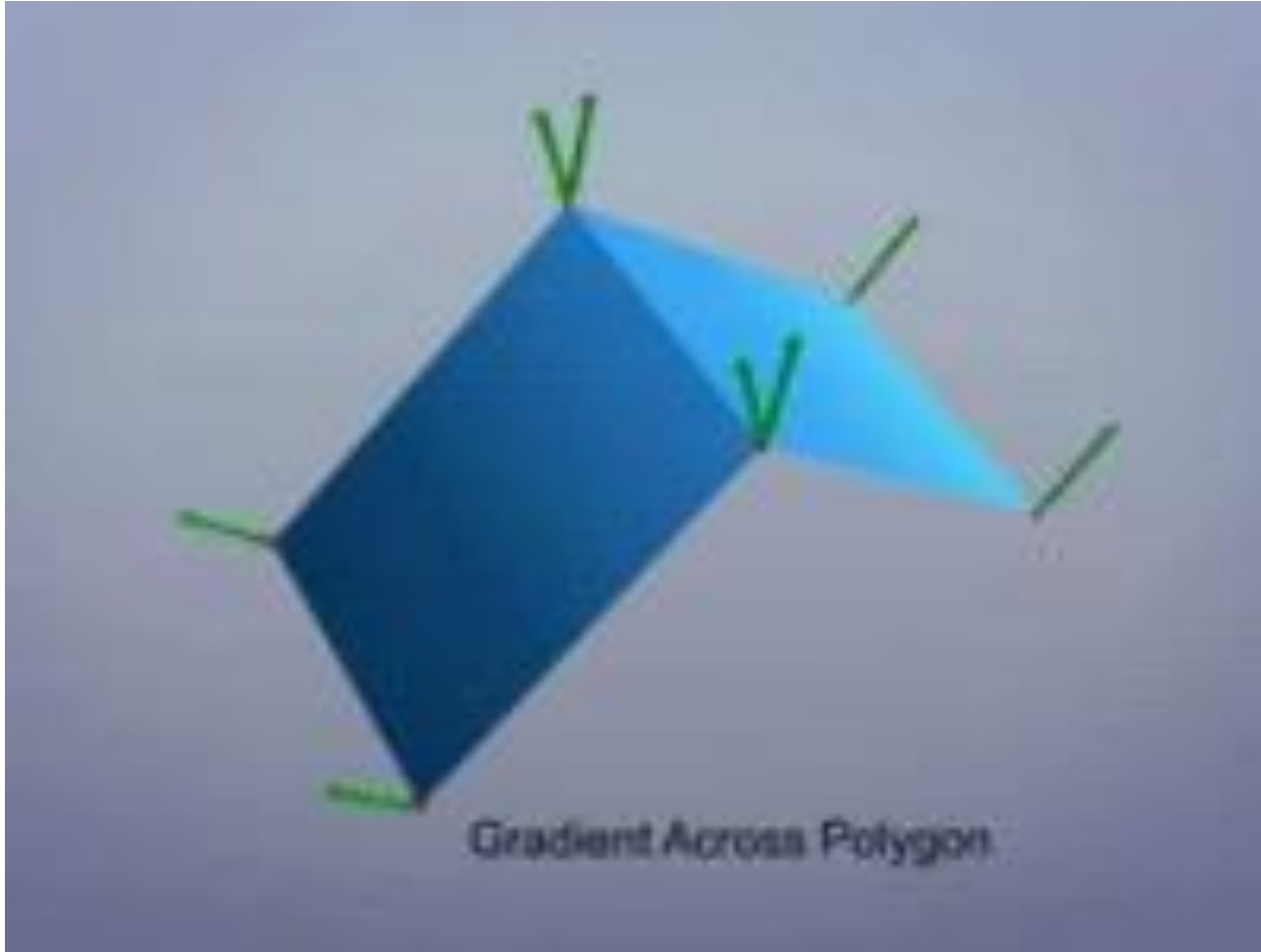


Suavizando

Com essa técnica podemos produzir superfícies que parecem suaves, quando não são.



Smooth Shading



<https://www.youtube.com/watch?v=PMgjVJoglbc>

Material

O nó **Material** especifica propriedades de material de superfície para nós de geometria associados e é usado pelas equações de iluminação X3D durante a renderização.

```
Material : X3DMaterialNode {  
    SFFloat  [in,out]  ambientIntensity  0.2          [0,1]  
    SFColor  [in,out]  diffuseColor      0.8 0.8 0.8    [0,1]  
    SFColor  [in,out]  emissiveColor     0 0 0         [0,1]  
    SFNode   [in,out]  metadata          NULL         [X3DMetadataObject]  
    SFFloat  [in,out]  shininess         0.2          [0,1]  
    SFColor  [in,out]  specularColor     0 0 0         [0,1]  
    SFFloat  [in,out]  transparency      0            [0,1]  
}
```

Novos Nós X3D : NavigationInfo

O campo do **headlight** especifica se um navegador deve acender uma luz direcional que sempre aponta na direção que o usuário está olhando. Definir este campo como TRUE faz com que o visualizador forneça sempre uma luz do ponto de vista do usuário. A luz headlight deve ser direcional, ter intensidade = 1, cor = (1, 1, 1), ambientIntensity = 0.0 e direção = (0, 0, -1).

```
NavigationInfo : X3DBindableNode {
  SFFloat   [in]    set_bind
  MFFloat   [in,out] avatarSize      [0.25 1.6 0.75]   [0,∞)
  SFBool    [in,out] headlight      TRUE
  SFNode    [in,out] metadata        NULL             [X3DMetadataObject]
  SFFloat   [in,out] speed            1.0              [0,∞)
  SFTime    [in,out] transitionTime  1.0              [0, ∞)
  MFString  [in,out] transitionType  ["LINEAR"]        ["TELEPORT","LINEAR", "ANIMATE",...]
  MFString  [in,out] type              ["EXAMINE" "ANY"] ["ANY","WALK","EXAMINE","FLY","LOOKAT","NONE","EXPLORE",...]
  SFFloat   [in,out] visibilityLimit  0.0              [0,∞)
  SFTime    [out]    bindTime
  SFBool    [out]    isBound
  SFBool    [out]    transitionComplete
}
```


Novos Nós X3D : DirectionalLight

Define uma fonte de luz direcional que ilumina ao longo de raios paralelos em um determinado vetor tridimensional. Possui os campos básicos **ambientIntensity**, **color**, **intensity**. O campo de **direction** especifica o vetor de direção da iluminação que emana da fonte de luz no sistema de coordenadas local. A luz é emitida ao longo de raios paralelos de uma distância infinita.

```
DirectionalLight : X3DLightNode {  
  SFFloat [in,out] ambientIntensity 0      [0,1]  
  SFColor  [in,out] color            1 1 1  [0,1]  
  SFVec3f  [in,out] direction        0 0 -1 (-∞,∞)  
  SFBool   [in,out] global           FALSE  
  SFFloat  [in,out] intensity        1      [0,1]  
  SFNode   [in,out] metadata         NULL   [X3DMetadataObject]  
  SFBool   [in,out] on               TRUE  
}
```

Equação de Cores (padrão X3D simplificado)

$$\mathbf{I}_{\text{rgb}} = O_{\text{Ergb}} + \text{SUM}(\mathbf{I}_{\text{Lrgb}} \times (\text{ambient}_i + \text{diffuse}_i + \text{specular}_i))$$

$$\text{ambient}_i = I_{\text{ia}} \times O_{\text{Drgb}} \times O_{\text{a}}$$

$$\text{diffuse}_i = I_i \times O_{\text{Drgb}} \times (\mathbf{N} \cdot \mathbf{L})$$

$$\text{specular}_i = I_i \times O_{\text{Srgb}} \times (\mathbf{N} \cdot ((\mathbf{L} + \mathbf{v}) / |\mathbf{L} + \mathbf{v}|))^{\text{shininess} \times 128}$$

$$\mathbf{I}_{\text{Lrgb}} = \text{light color} \quad \mathbf{I}_i = \text{light intensity} \quad \mathbf{I}_{\text{ia}} = \text{light ambientIntensity}$$

$$O_{\text{Ergb}} = \text{material emissiveColor} \quad O_{\text{Drgb}} = \text{material diffuse colour} \quad O_{\text{Srgb}} = \text{material specularColor}$$

$$O_{\text{a}} = \text{material ambientIntensity}$$

$$\mathbf{L} = \text{direction of light source}$$

$$\mathbf{N} = \text{normalized normal vector at this point on geometry}$$

$$\mathbf{v} = \text{normalized vector from point on geometry to viewer's position}$$

Computação Gráfica

Luciano Soares

<lpsoares@insper.edu.br>

Fabio Orfali

<fabioO1@insper.edu.br>