

# Algoritmos de Iluminação Direta

Prof. Antonio L. Apolinário Jr.

---

*Ufba-Im-Dcc-Bcc- 2014.2*

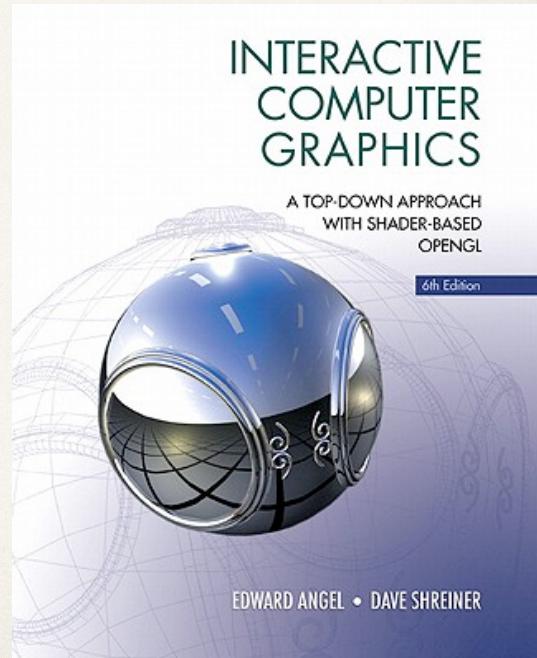
## Roteiro

---

- Algoritmos de renderização básico
- Algoritmo *Flat*
- Algoritmo *Gouraud*
- Algoritmo *Phong*
- Limitações dos modelos locais de reflexão

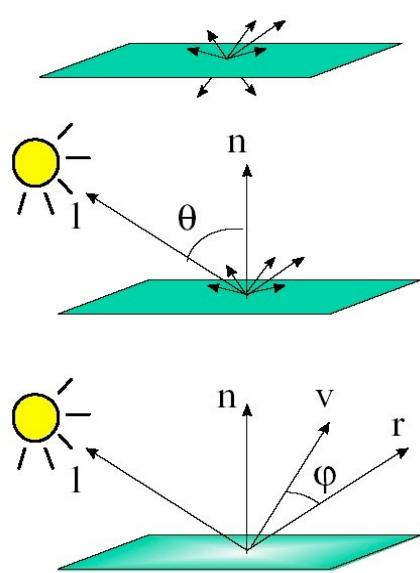
# Leitura de referencia

- \* Capítulo 5  
Interactive Computer Graphics -  
A top-down approach with OpenGL  
6th Edition  
Angel, Edward.  
Addison-Wesley. 2012.



## Modelo de Reflexão de Phong

Ambient + Diffuse + Specular



$$I_a = k_a L_a$$

$$I_d = k_d L_d \cos(\theta) = k_d L_d (\mathbf{n} \cdot \mathbf{l})$$

Lamberth's Law

$$I_d = k_s L_s \cos^\alpha(\phi) = k_s L_s (\mathbf{v} \cdot \mathbf{r})^\alpha$$

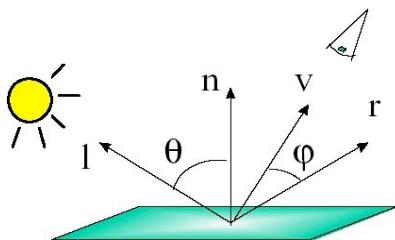
# Modelo de Reflexão de Phong

## Phong Reflection model

For each color (r,g,b) calculate reflected intensity:

$$I = \sum_{\text{All lightsources}} \left( k_a L_a + \frac{1}{a+bd+cd^2} \left( k_d L_d (\mathbf{n} \cdot \mathbf{l}) + k_s L_s (\mathbf{v} \cdot \mathbf{r})^\alpha \right) \right)$$

Ambient  
Distance term  
Diffuse  
Specular  
All lightsources  
Distance to lightsource  
Shininess



## Algoritmo de Iluminação Básico

**Algoritmo ColoreObjeto;**

**Dados : Fonte de Luz, Observador, Objeto.**

**para cada face do objeto faça**

**para cada ponto da face faça**

**calcular a normal no ponto;**

**calcular os ângulos  $\theta$  e  $\Omega$  formado pelos vetores  $L$ ,  $N$  e  $V$ ,  $R$ , respectivamente;**

**aplicar a **equação de Phong** calculando as intensidades refletidas no ponto em cada uma das componentes RGB;**  
**pintar esse ponto com a cor correspondente;**

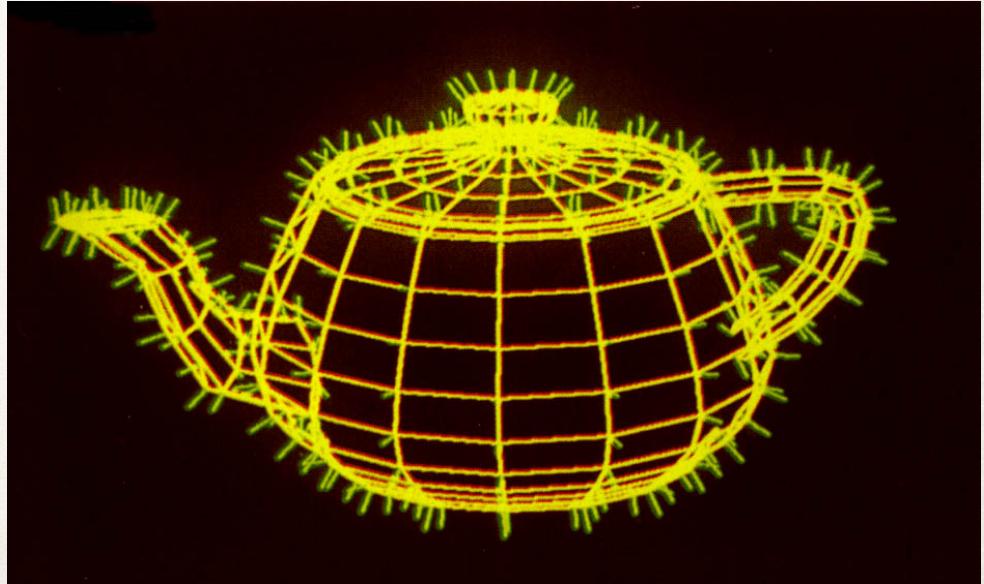
**fim-para;**

**fim-para;**

**fim.**

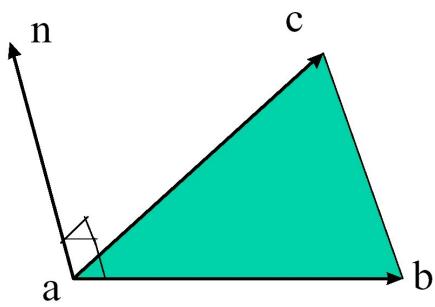
# Renderização de Modelos Poligonais

- Modelos Poligonais
  - Vertices
  - Faces
  - Normais

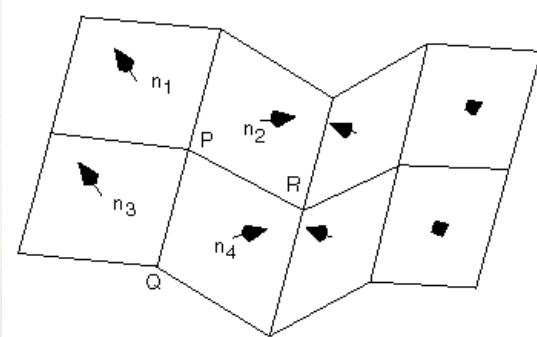
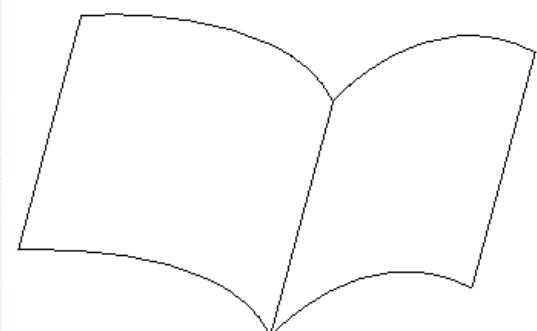


## Cálculo do Vetor Normal em Modelos Poligonais

- Faces são planares
  - Uma normal para cada face do modelo

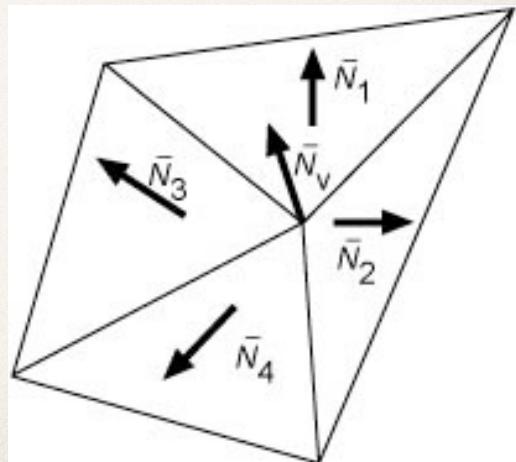
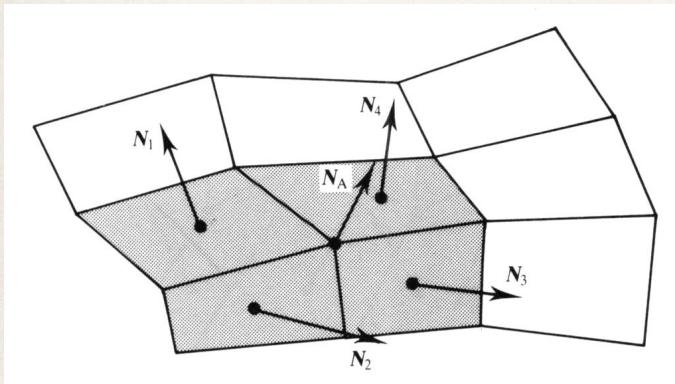


$$n = (b-a) \times (c-a) / |(b-a) \times (c-a)|$$



# Cálculo do Vetor Normal em Modelos Poligonais

- Normal nos vértices
  - Média das normais nas faces
  - Pré- processamento
  - Incluído em alguns modelos



## Algoritmo de Iluminação - Flat

*Algoritmo ColoraçãoFlat;*

*Dados : Fonte de Luz, Observador, Objeto.*

*para cada face do objeto faça*

*calcular a média das normais em cada vértice da face;*

*calcular os ângulos  $\theta$  e  $\Omega$  formado pelos vetores **L**, **N** e **V**, **R**, respectivamente;*

*aplicar a equação de Phong calculando a intensidade refletida em um ponto (em cada uma das componentes RGB);*

*pintar todo a face com a cor correspondente;*

*fim-para;*

*fim.*

# Algoritmo de Iluminação - Flat

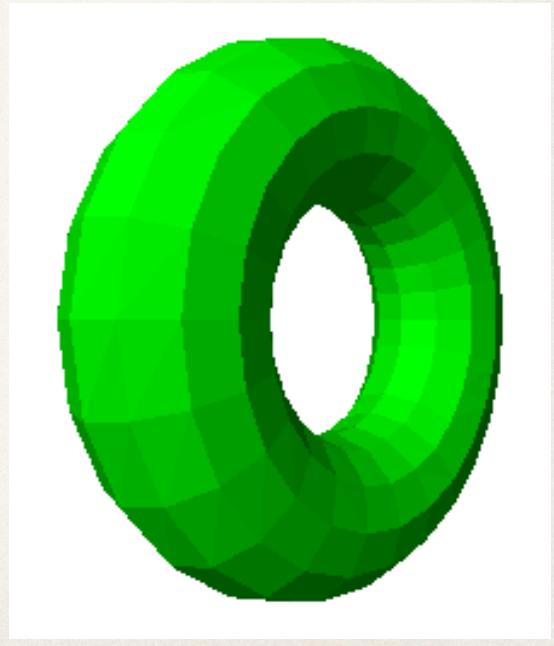
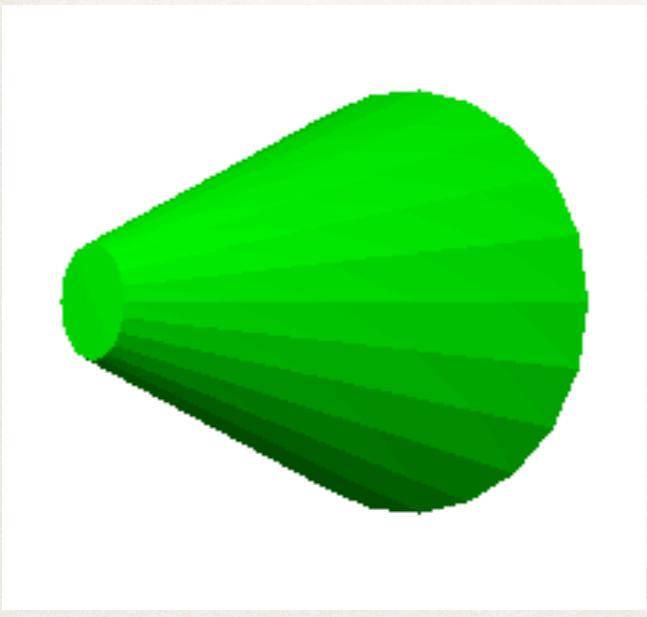
---

- Simplificação do algoritmo básico
  - Uma única normal por face
- Problema :
  - A resolução da malha poligonal fica aparente
  - Aumentar a resolução da malha
    - Aumento no tempo de renderização



# Algoritmo de Iluminação - Flat

---



# Algoritmo de Iluminação - Gouraud

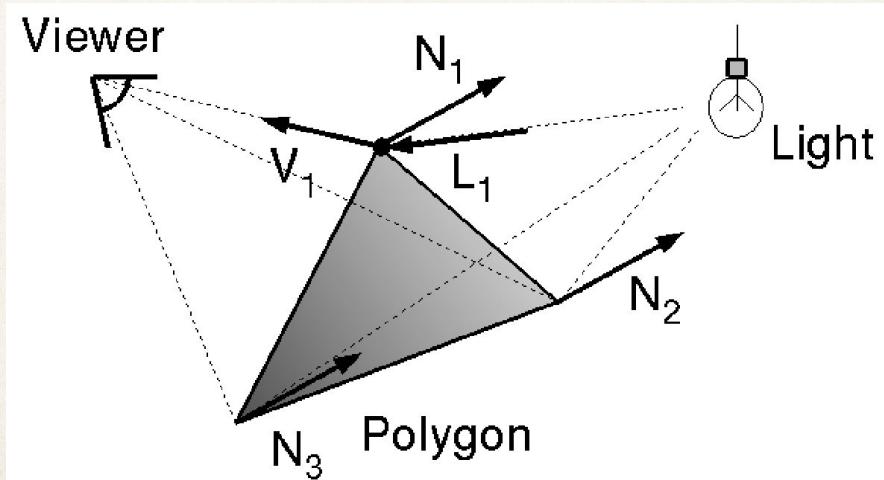
**Algoritmo ColoreGouraud;**

**Dados : Fonte de Luz, Observador, Objeto.**

```
para cada face do objeto faça
    calcular a normal em cada vértice da face;
    calcular os ângulos  $\theta$  e  $\Omega$  formado pelos vetores  $L$ ,  $N$  e  $V$ ,  $R$ ,
    respectivamente;
    aplicar a equação de Phong calculando as intensidades
        refletidas em cada vértice para cada uma das componentes
        RGB;
    para cada ponto da face faça
        calcular a intensidade no ponto com base na interpolação linear
            das intensidades nos vértices;
        pintar esse ponto com a cor correspondente;
Fim-para;
fim-para;
fim.
```

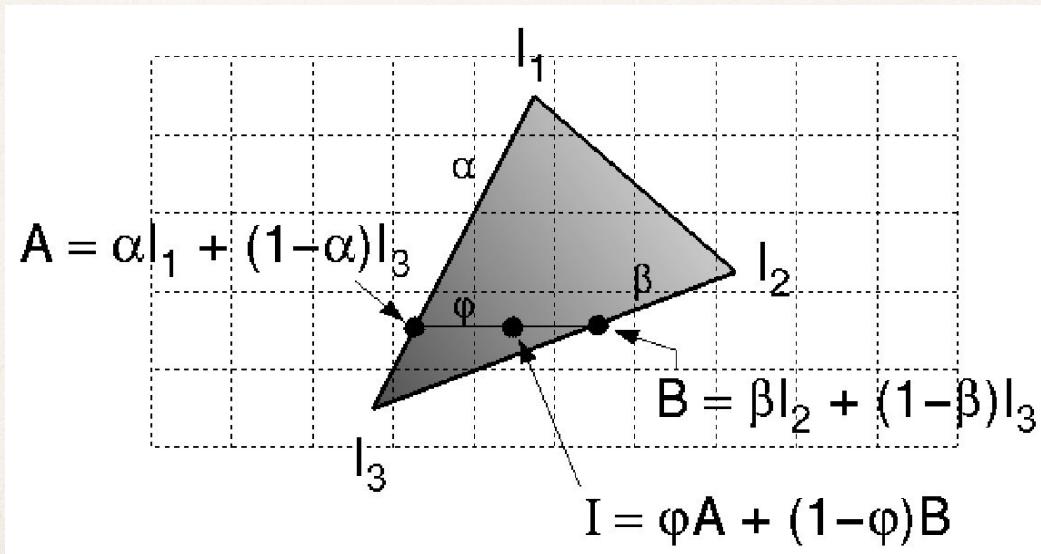
# Algoritmo de Iluminação - Gouraud

- Suaviza as transições entre as faces
  - Normais nos vértices são medias das normais nas faces incidentes ao vértice



# Algoritmo de Iluminação - Gouraud

- Interpolação das intensidades nos vértice

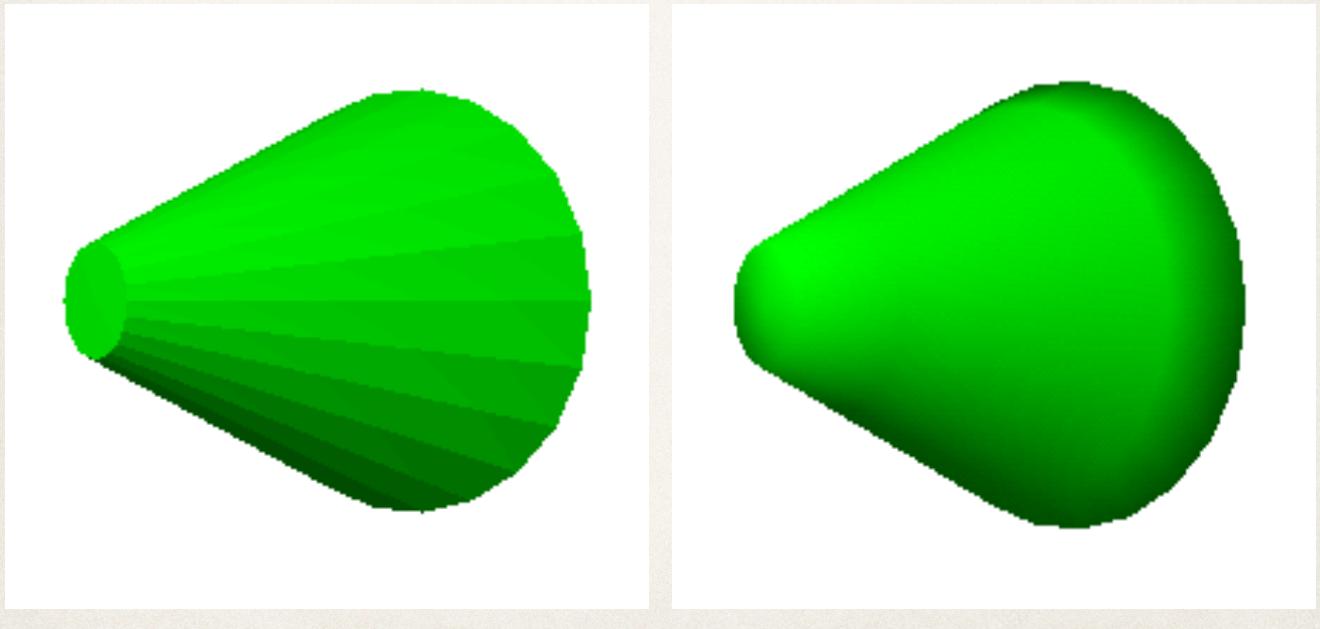


# Algoritmo de Iluminação - Gouraud



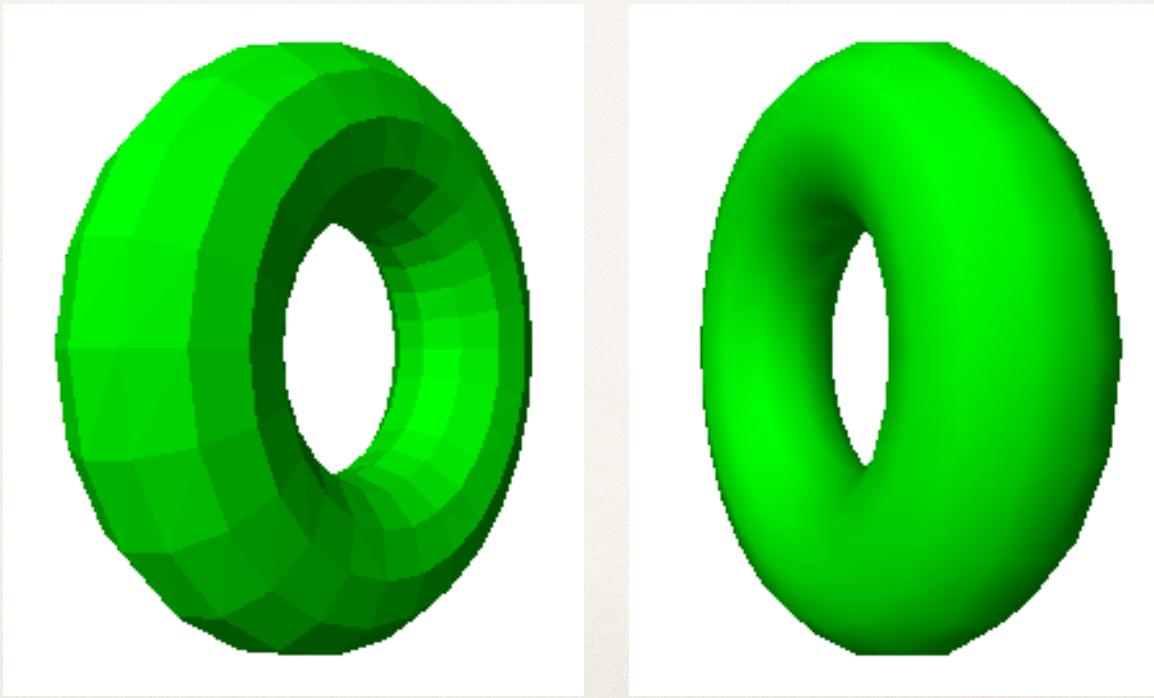
# Algoritmo de Iluminação - Gouraud

---



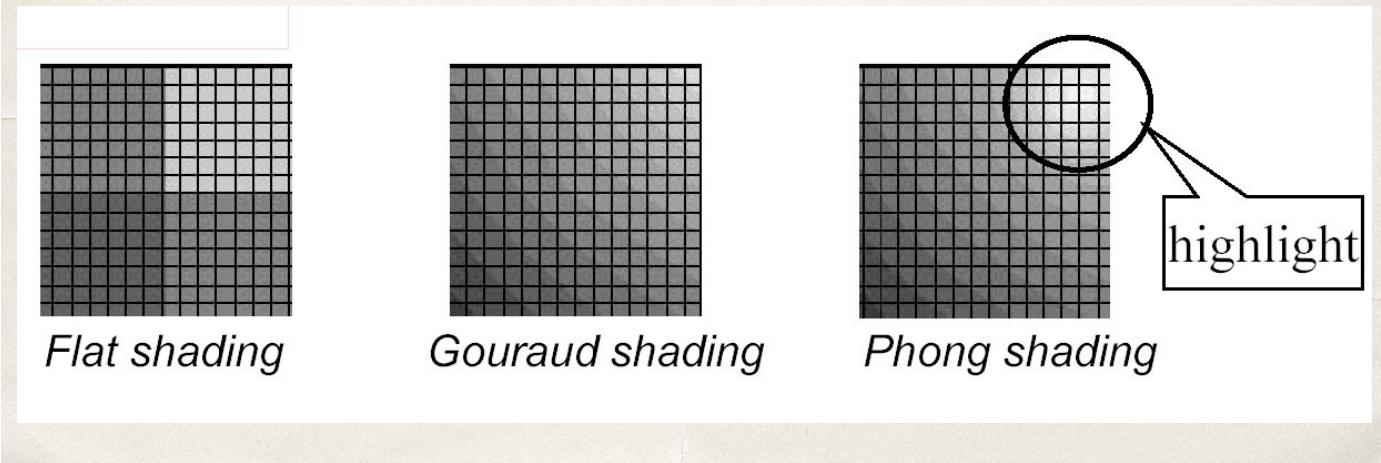
# Algoritmo de Iluminação - Gouraud

---



# Algoritmo de Iluminação - Gouraud

- Problema
  - Não representa *highlights* típicos de materiais especulares
    - Média das intensidades nos vértices



# Algoritmo de Iluminação - Phong

**Algoritmo ColorePhong;**

**Dados : Fonte de Luz, Observador, Objeto.**

para cada face do objeto faça

    calcular a normal em cada vértice da face;

    para cada ponto da face faça

        calcular a normal no ponto com base na interpolação linear  
        das normais nos vértices;

        calcular os ângulos  $\theta$  e  $\Omega$  formado pelos vetores  $L$ ,  $N$  e  $V$ ,  $R$ ,  
        respectivamente;

        aplicar a equação de Phong calculando a intensidade refletida  
        no ponto para cada uma das componentes RGB;

        pintar esse ponto com a cor correspondente;

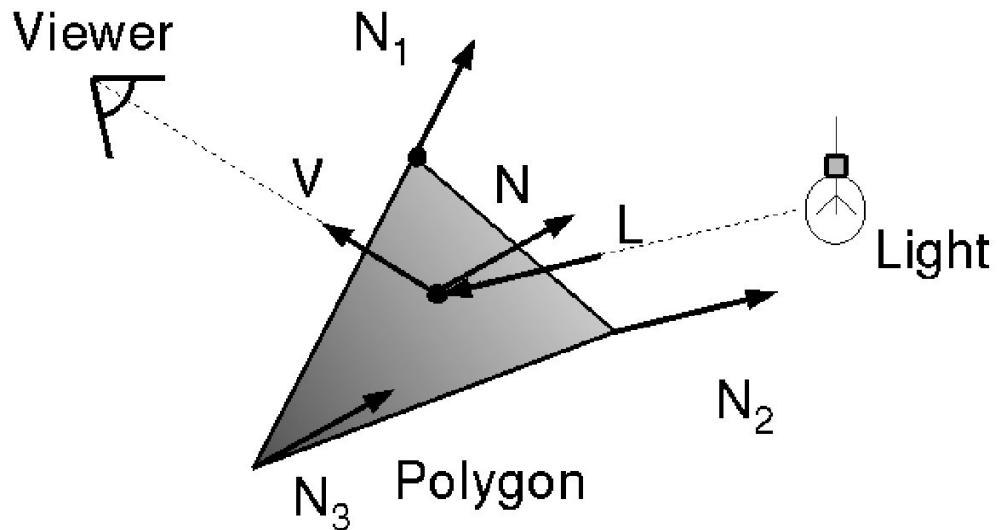
    fim-para;

fim-para;

fim.

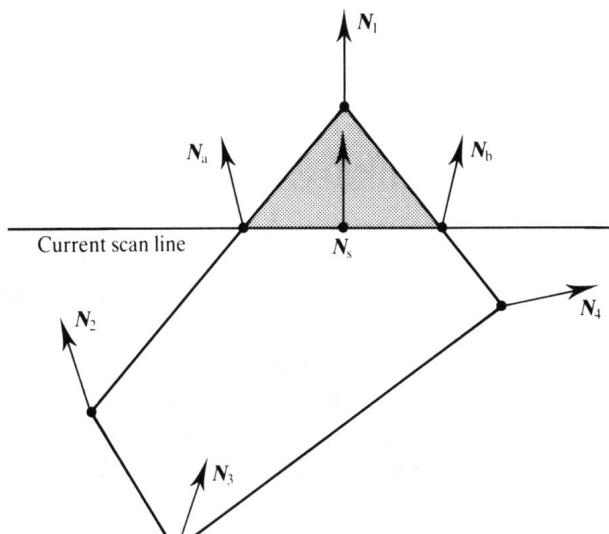
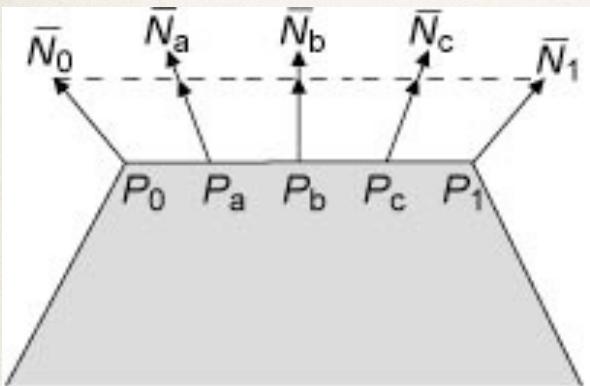
# Algoritmo de Iluminação - Phong

- Suaviza as transições entre as normais nos vértices



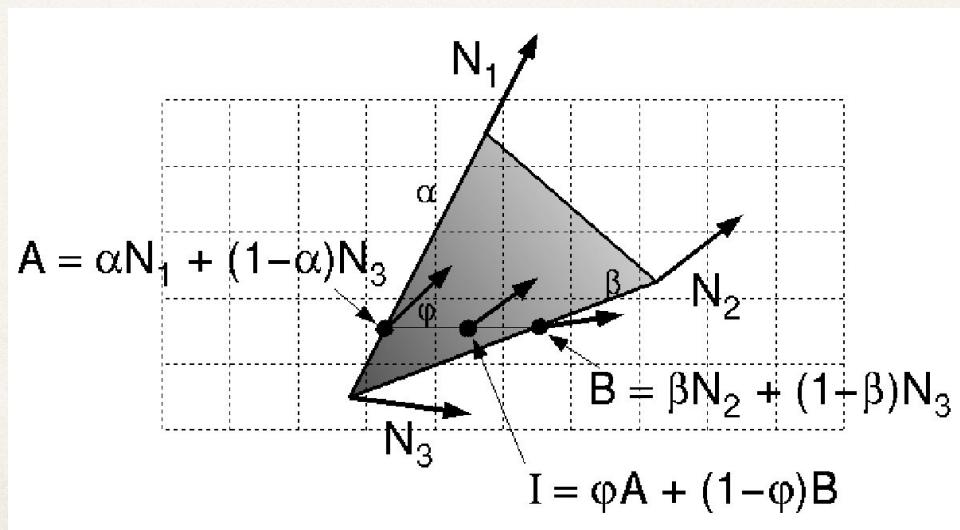
# Algoritmo de Iluminação - Phong

- Interpolação das normais nos vértices
  - Intensidade é calculada a cada ponto

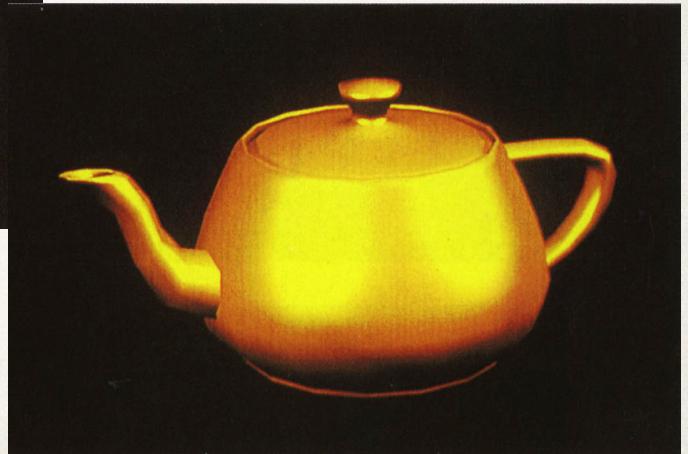
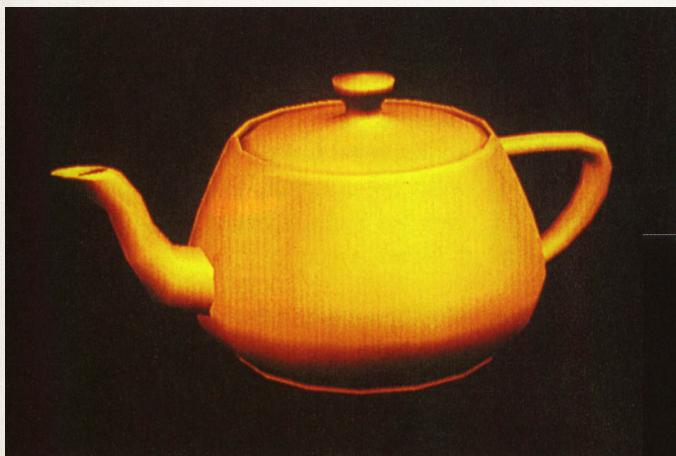


# Algoritmo de Iluminação - Phong

- Interpolação das normais nos vértices
  - Intensidade é calculada a cada ponto

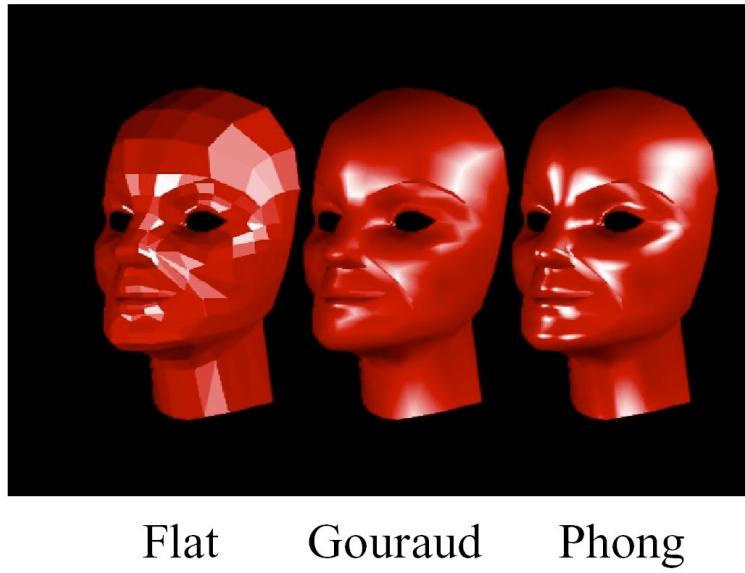


# Algoritmo de Iluminação - Phong



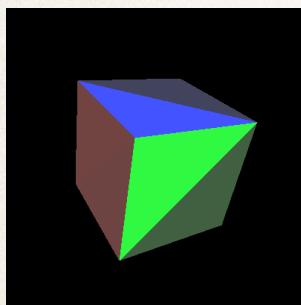
# Algoritmos de Iluminação

- Comparando os resultados

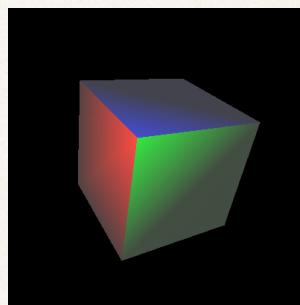


# Algoritmos de Iluminação

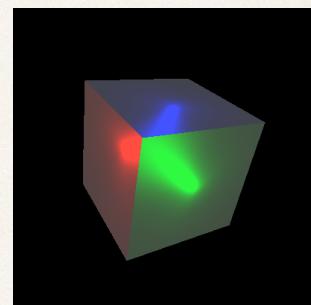
- Vários algoritmos para o cálculo da iluminação baseado no modelo de *Phong*:
  - Como a iluminação dos vértices é interpolada no interior de uma face



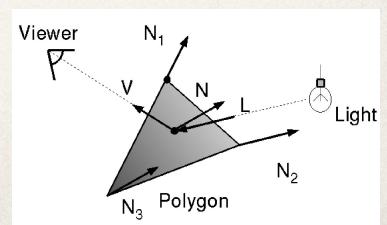
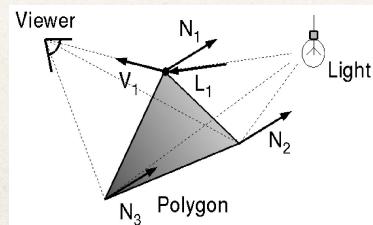
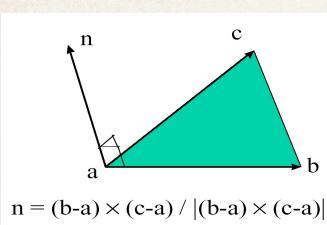
Flat



Gouraud



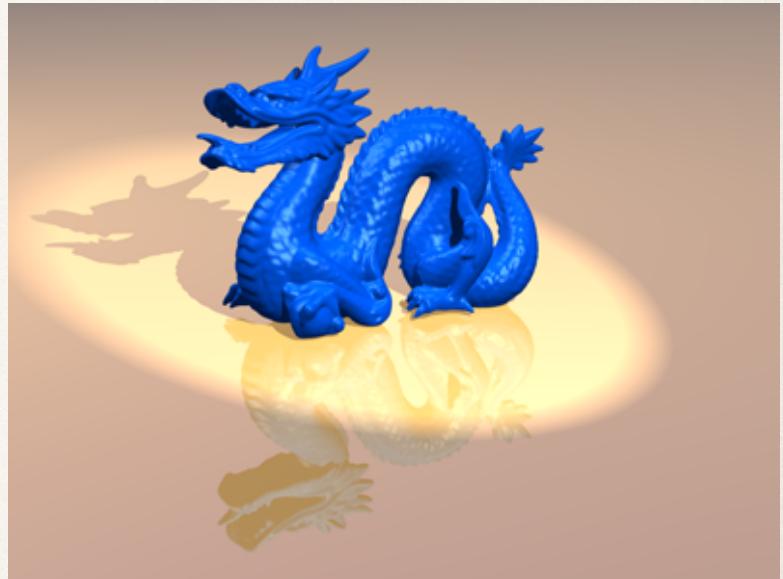
Phong



# Restrições dos Algoritmos de Iluminação Locais

---

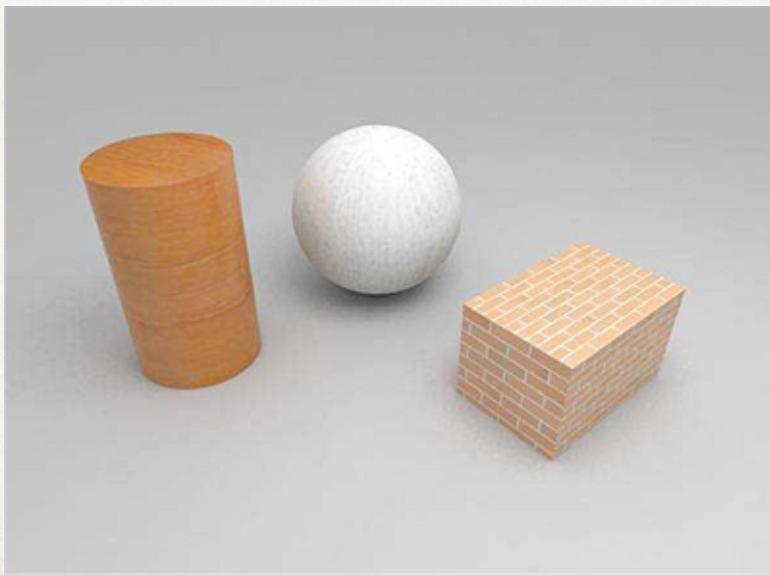
- Como representar?
  - Sombras
  - Penumbra
  - Reflexões indiretas



# Restrições dos Algoritmos de Iluminação Locais

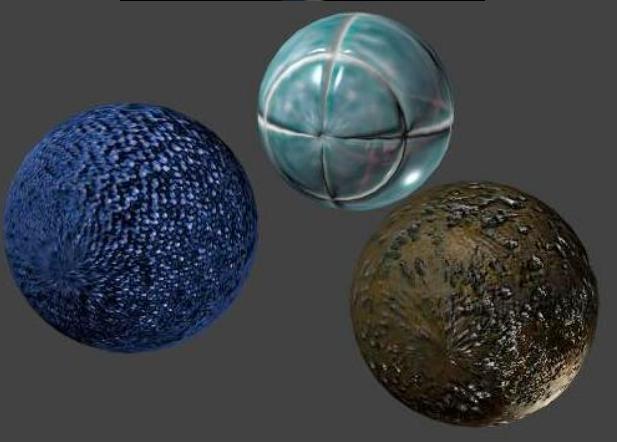
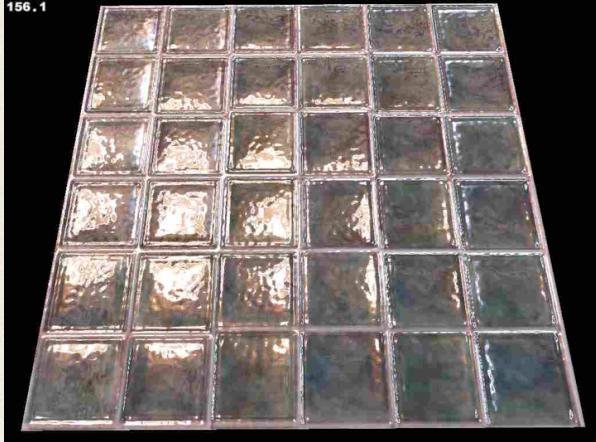
---

- Como representar?
  - Materiais heterogêneos



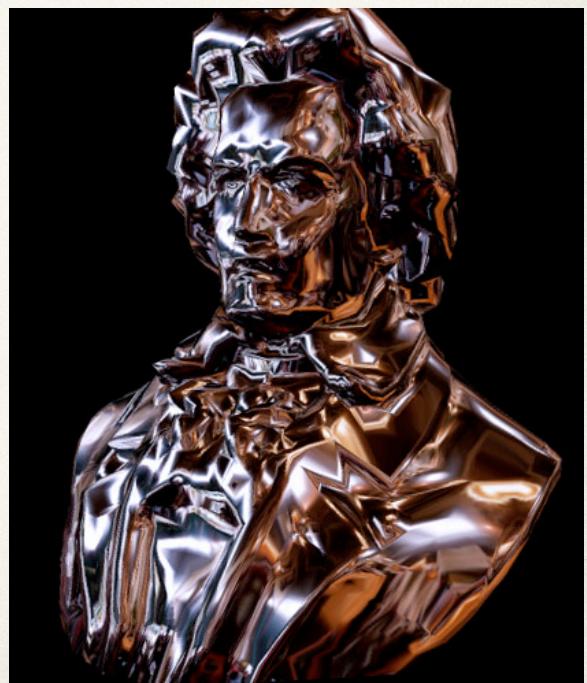
# Restrições dos Algoritmos de Iluminação Locais

- Como representar?
  - Superfícies rugosas e irregulares



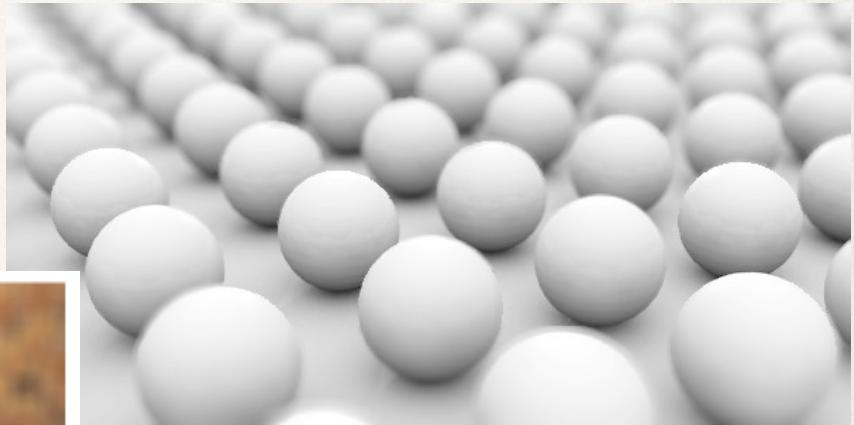
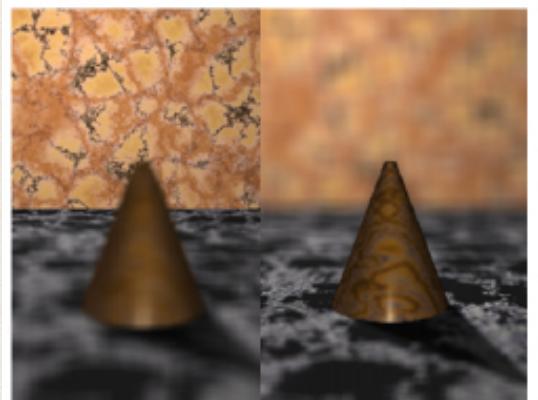
# Restrições dos Algoritmos de Iluminação Locais

- Como representar?
  - Reflexões indiretas



# Restrições dos Algoritmos de Iluminação Locais

- Como representar?
  - Controle de focalização



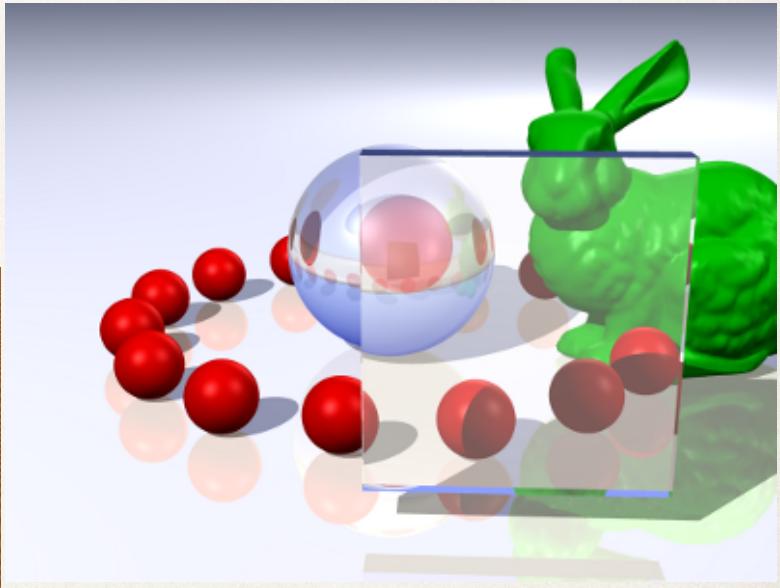
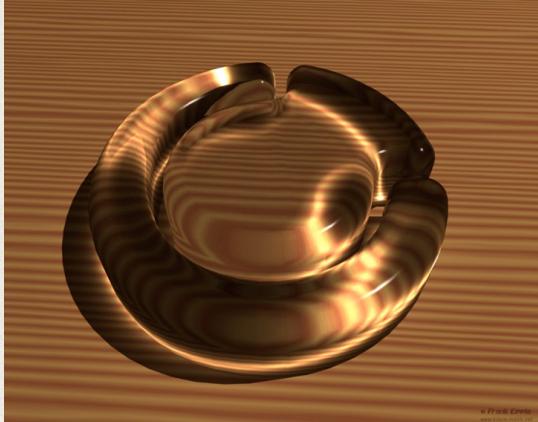
# Restrições dos Algoritmos de Iluminação Locais

- Como representar?
  - Borramento por movimento (*Motion Blur*)



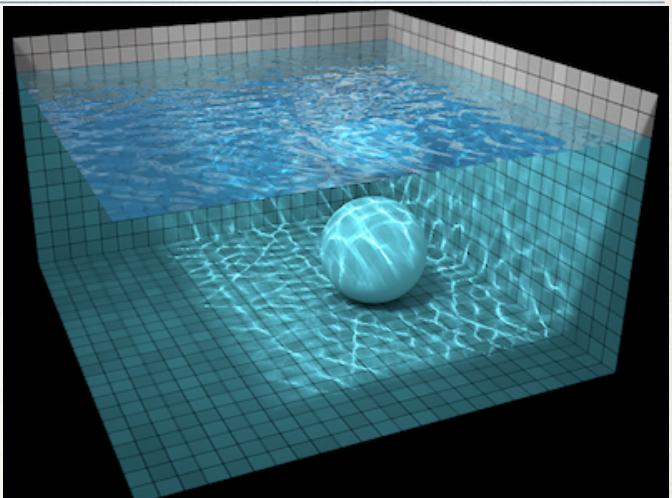
# Restrições dos Algoritmos de Iluminação Locais

- Como representar?
  - Distorções ópticas



# Restrições dos Algoritmos de Iluminação Locais

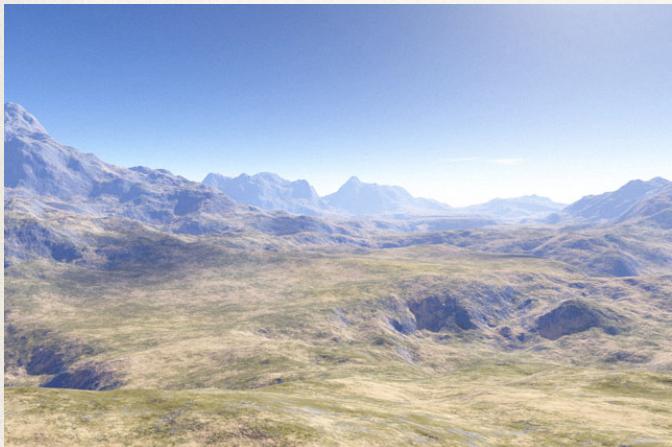
- Como representar?
  - Fenômenos naturais



# Restrições dos Algoritmos de Iluminação Locais

---

- Como representar?
  - Iluminação natural



A Seguir....  
Técnicas de Renderização

---