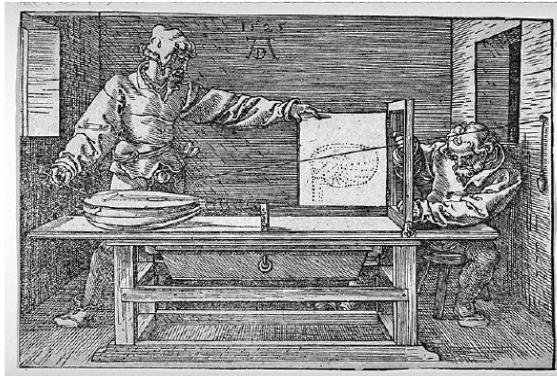
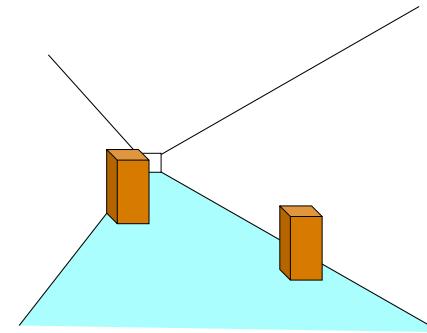


## Algoritmo de Rastreamento de Raios

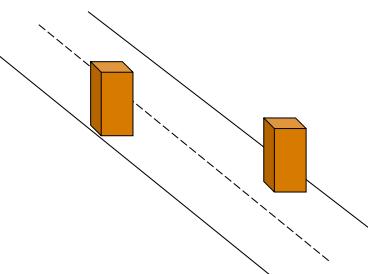


*Man Drawing a Lute (The Draughtsman of the Lute), woodcut 1525, Albrecht Dürer.*

## Perspectiva e tamanhos relativos

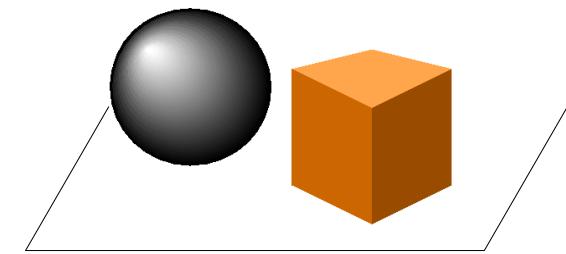


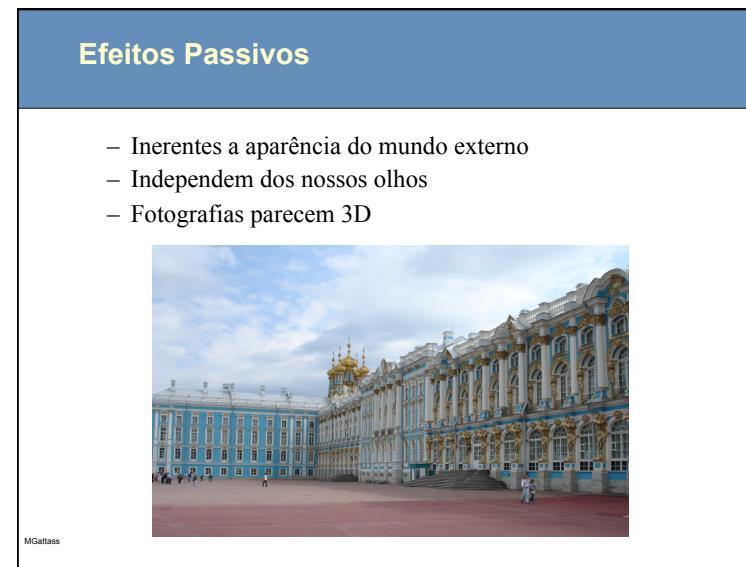
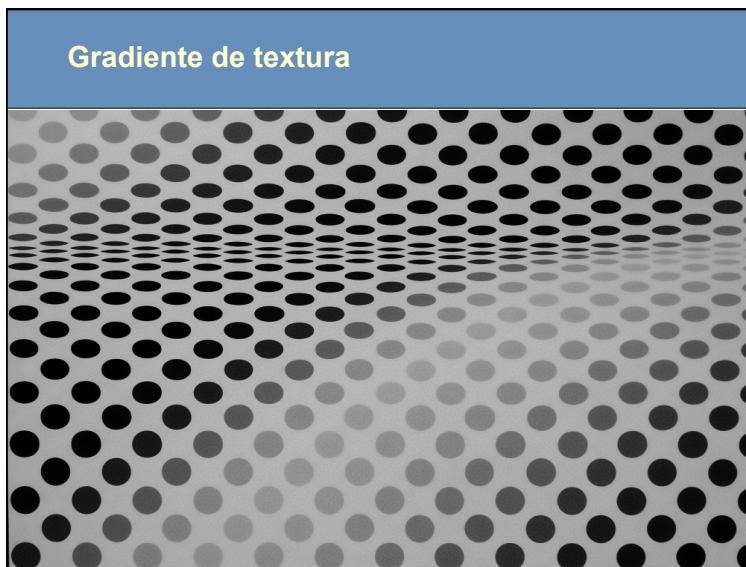
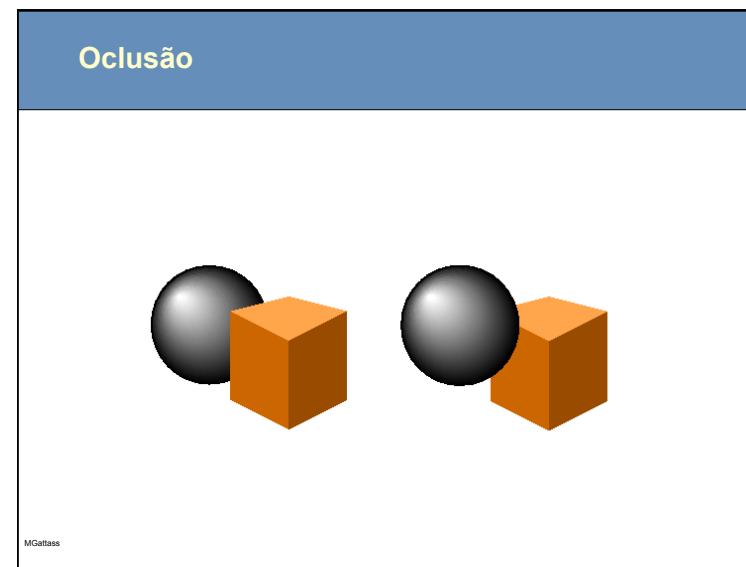
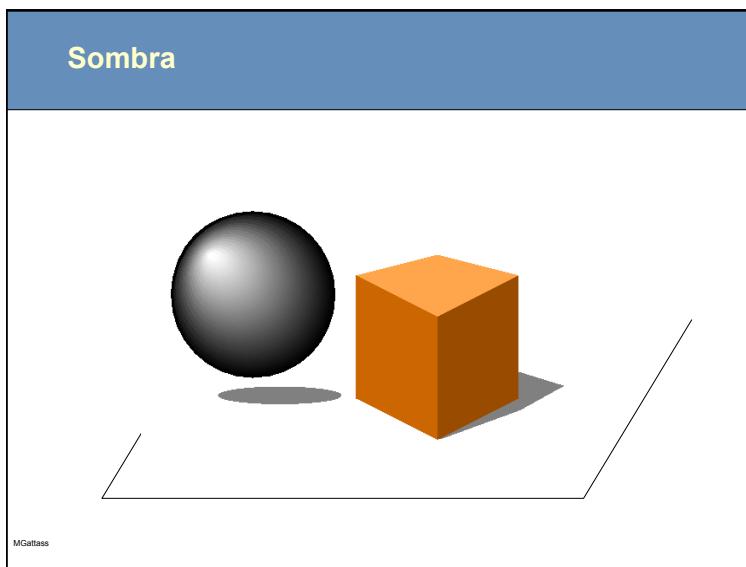
## Perspectiva e tamanhos relativos



## Iluminação e posição

O que está na frente?  
A que distância do apoio?





**Camera obscura**

Plymouth, UK

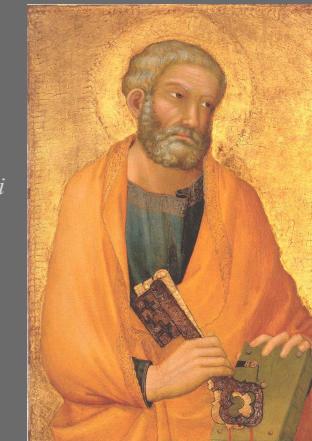
**Camera Obscura - San Francisco**

The Camera Obscura at the Cliff House is one of several remaining camera obscuras in the world. The device is an ancient precursor to modern photography, and well worth a visit, especially if you haven't previously visited a camera obscura.

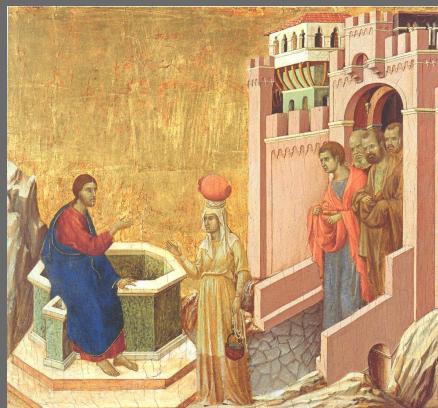
MGattass

**Pintura em 1300's**

1326  
Simone Martini  
São Pedro



MGattass

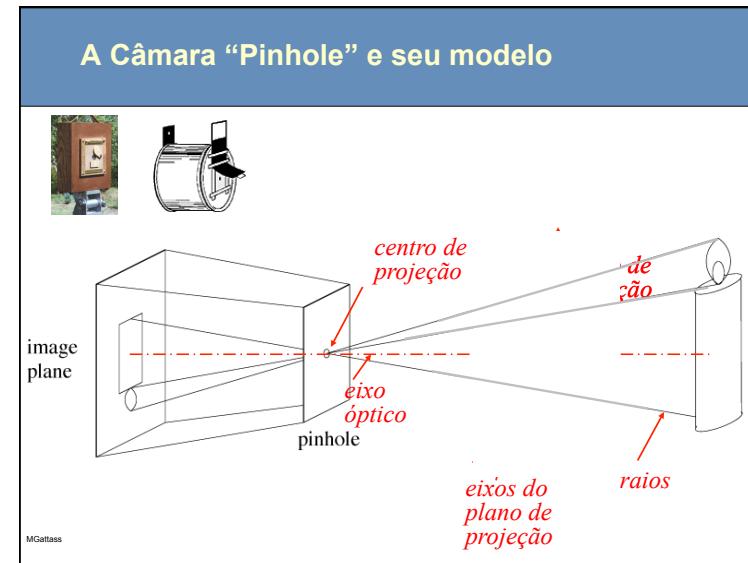
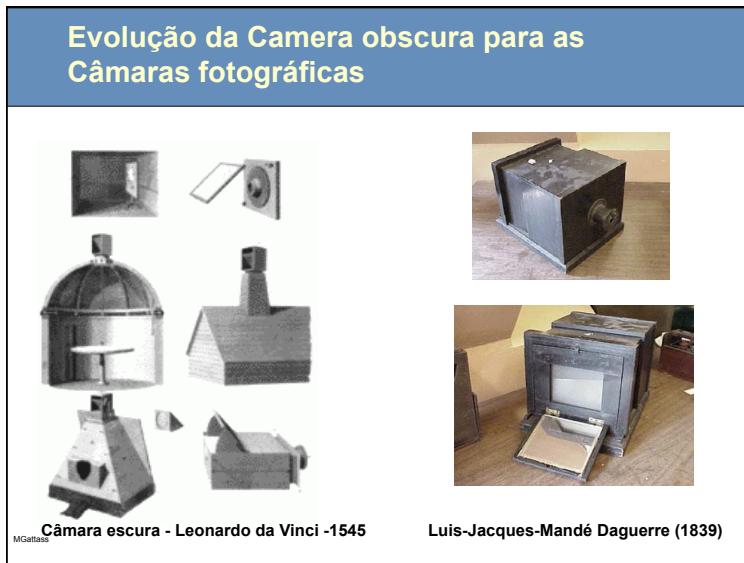
**Pintura em 1300's: busca pelo realismo**

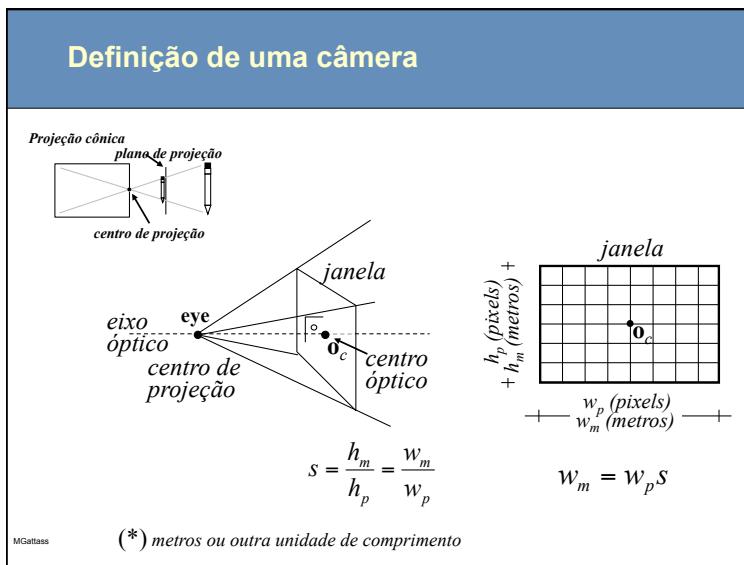
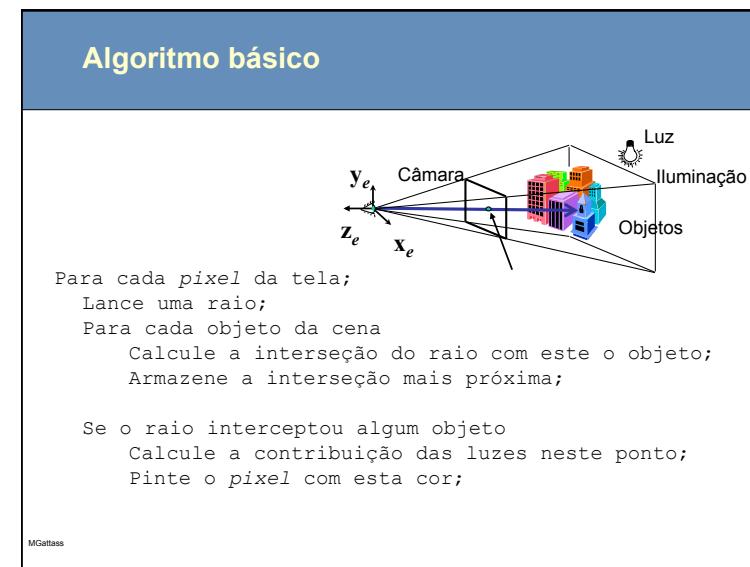
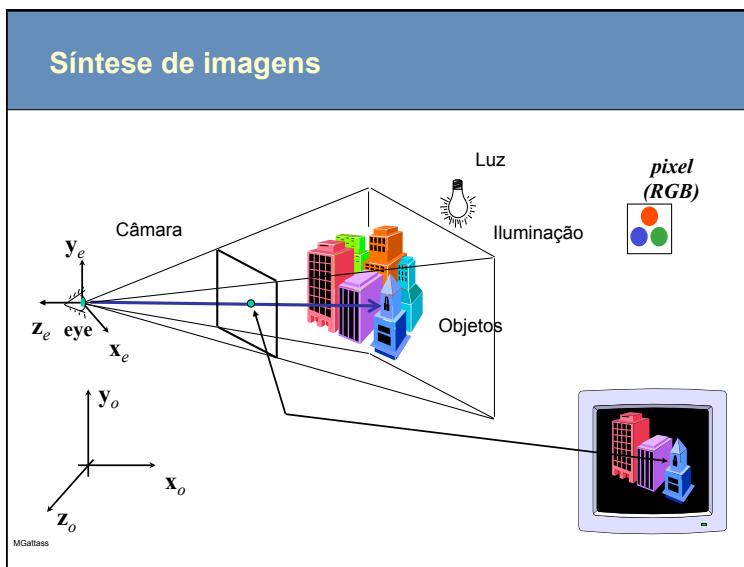
1310-11  
Duccio di Buoninsegna  
Cristo e a Samaritana

MGattass

**Pintores**

Canaletto (Giovanni Antonio Canal) (1697-1768).





### Abertura de uma câmera

$$\tan\left(\frac{\text{fov}_y}{2}\right) = \frac{(h_m/2)}{d_f}$$

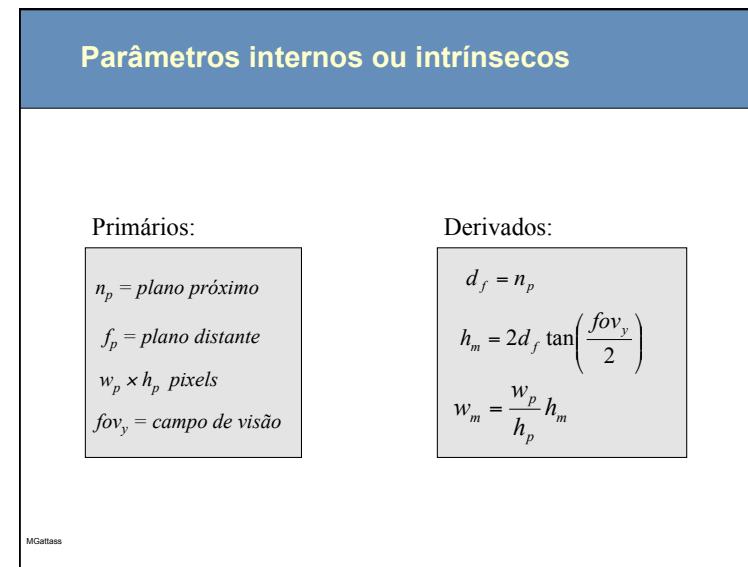
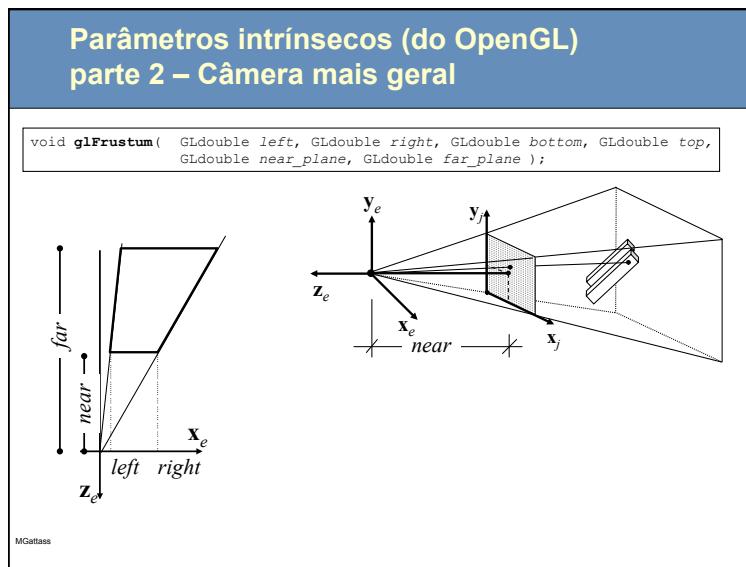
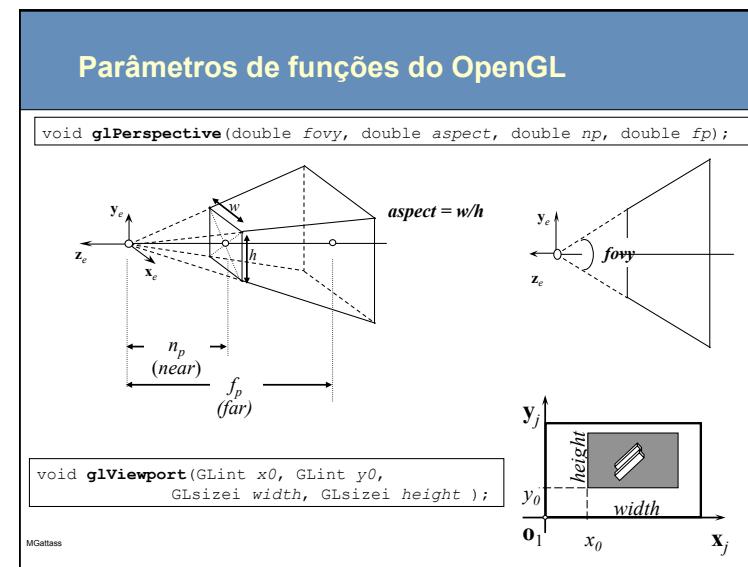
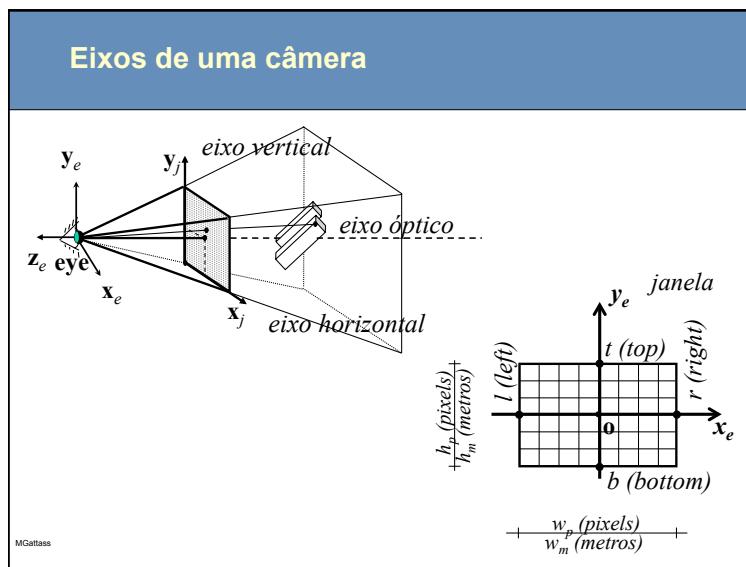
$$h_m = 2d_f \tan\left(\frac{\text{fov}_y}{2}\right)$$

$h_m$

$d_{f1} + d_{f2}$

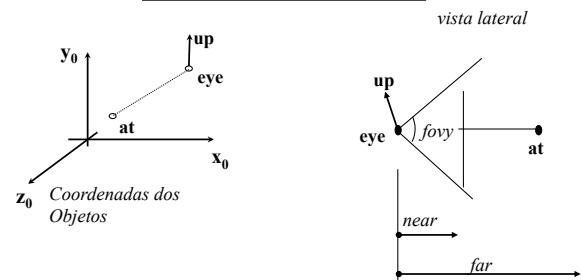
$\text{fov}_{y1}$   $\text{fov}_{y2}$

MGattass



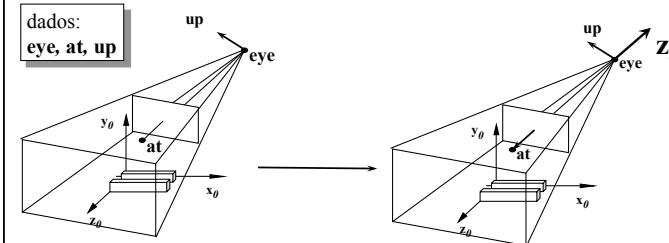
## Posicionamento da câmera (parâmetros externos ou extrínsecos)

**eye** = centro óptico  
**at** = ponto de visada,  
**up** = direção para cima



MGattass

## Calculo do sistema do olho - $x_e y_e z_e$

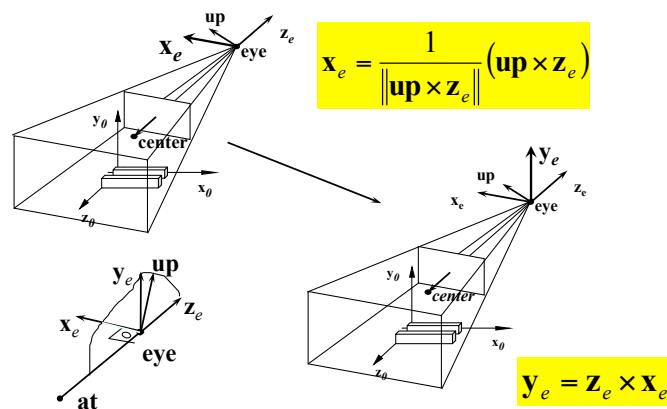


$$z_e = \frac{1}{\|up\|} (up \times (eye - at))$$

MGattass

## Calculo do sistema do olho - $x_e y_e z_e$

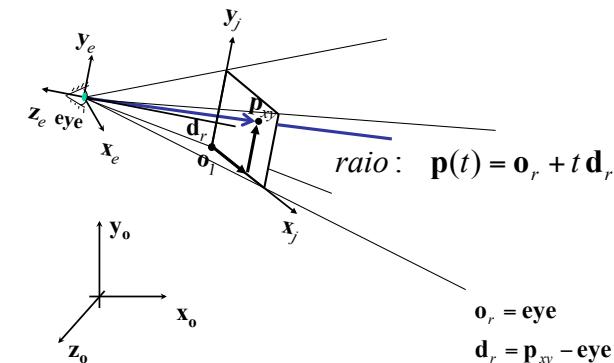
$$x_e = \frac{1}{\|up\|} (up \times z_e)$$



$$y_e = z_e \times x_e$$

MGattass

## Um modelo de câmera

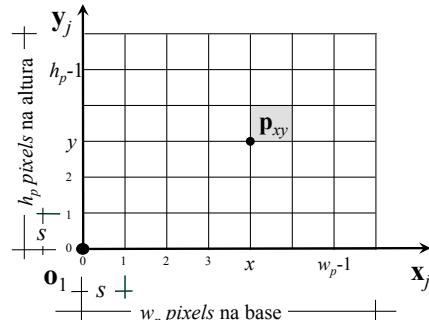


MGattass

## Lançamento de Raios

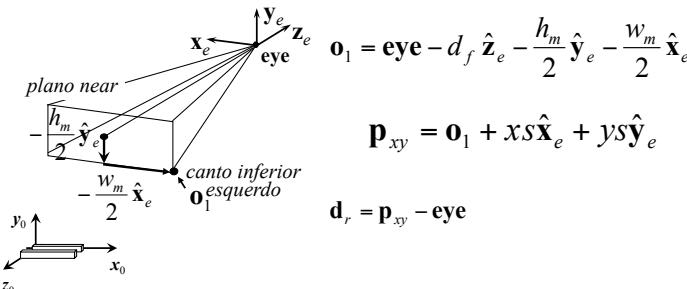
$$\mathbf{p}_{xy} = \mathbf{o}_1 + (xs)\hat{\mathbf{x}}_j + (ys)\hat{\mathbf{y}}_j$$

$$\mathbf{p}_{xy} = \mathbf{o}_1 + xs\hat{\mathbf{x}}_e + ys\hat{\mathbf{y}}_e$$



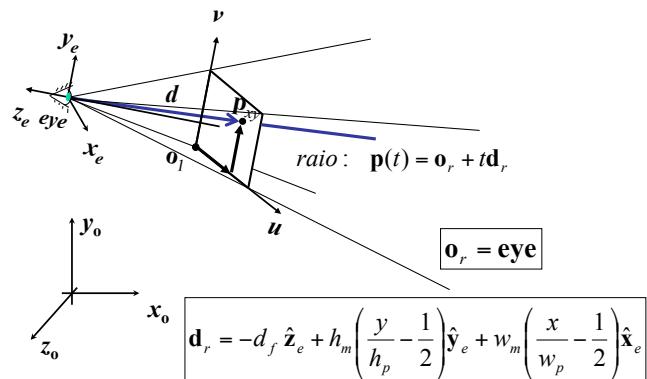
MGattass

## Canto inferior esquerdo da janela no plano near (ou far)



MGattass

## Resultando



MGattass

## Tipo Abstrato de Dados: Camera em C

```
struct _camera {
    /* Definição da câmera */
    Vector Eye, Center, Up;
    float fovy;
    float n,f;
    int wp,hp;

    /* Parâmetros derivados */
    float df;
    float wm,hm;
    Vector Xe,Ye,Ze;
};

typedef struct _camera Camera;
```

```
Camera* camCreate( Vector eye, Vector at, Vector up,
                    double fovy, double _near, double _far, int wp, int hp );
```

```
Ray camGetRay( Camera camera, double x, double y );
```

MGattass

## Objeto câmera

Inicialização (pré-processamento):

Dados:  $fov_y, w_p, h_p, n, f, eye, center, up$

$$\begin{aligned} d_f &= n & h_m &= 2d_f \tan\left(\frac{fov_y}{2}\right) & w_m &= \left(\frac{w_p}{h_p}\right)h_m \\ z_e &= \frac{1}{\|eye - center\|}(eye - center) & x_e &= \frac{1}{\|up \times z_e\|}(up \times z_e) & y_e &= (z_e \times x_e) \end{aligned}$$

Lançamento de raios:  $\mathbf{o}_r + t \mathbf{d}_r$

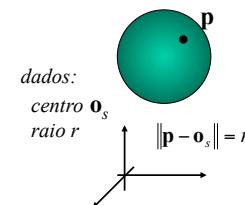
Dados:  $x, y$

$\mathbf{o}_r = eye$

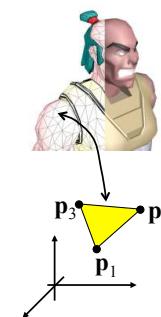
$$\mathbf{d}_r = -d_f \hat{\mathbf{z}}_e + h_m \left( \frac{y}{h_p} - \frac{1}{2} \right) \hat{\mathbf{y}}_e + w_m \left( \frac{x}{w_p} - \frac{1}{2} \right) \hat{\mathbf{x}}_e$$

## Modelagem dos Objetos

Implícita:

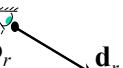


Por fronteira:



## Interseção de um raio com uma esfera

Raio:  $\mathbf{p}(t) = \mathbf{o}_r + t \mathbf{d}_r$



Esfera:  $\|\mathbf{p}(t_i) - \mathbf{o}_s\|^2 = r^2$

$$\|\mathbf{o}_r + t_i \mathbf{d}_r - \mathbf{o}_s\|^2 = r^2$$

$$((\mathbf{o}_r - \mathbf{o}_s) + t_i \mathbf{d}_r) \cdot ((\mathbf{o}_r - \mathbf{o}_s) + t_i \mathbf{d}_r) = r^2$$

$$[\mathbf{d}_r \cdot \mathbf{d}_r]t_i^2 + [2\mathbf{d}_r \cdot (\mathbf{o}_r - \mathbf{o}_s)]t_i + [(\mathbf{o}_r - \mathbf{o}_s) \cdot (\mathbf{o}_r - \mathbf{o}_s) - r^2] = 0$$

$$at_i^2 + bt_i + c = 0$$

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

MGattass

## Objeto esfera: métodos (dados $\mathbf{o}_r, \mathbf{d}_r, \mathbf{o}_s, r$ )

Interseção:

$$a = \mathbf{d}_r \cdot \mathbf{d}_r$$

$$b = 2\mathbf{d}_r \cdot (\mathbf{o}_r - \mathbf{o}_s)$$

$$c = \|\mathbf{o}_r - \mathbf{o}_s\|^2 - r^2$$

$$se \quad \Delta = b^2 - 4ac > 0$$

$$t_1 = \frac{-b - \sqrt{\Delta}}{2a}$$

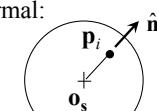
$$t_2 = \frac{-b + \sqrt{\Delta}}{2a}$$

MGattass

se  $t_i > 0$ :

$$\mathbf{p}_i = \mathbf{p}(t_i) = \mathbf{o}_r + t_i \mathbf{d}_r$$

Normal:



$$\hat{\mathbf{n}}_i = \frac{1}{\|\mathbf{p}_i - \mathbf{o}_s\|}(\mathbf{p}_i - \mathbf{o}_s)$$

### Interseção com o plano do triângulo

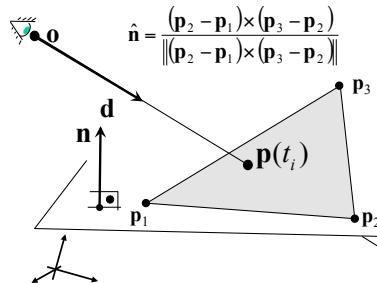
$$Raio: \quad \mathbf{p}(t) = \mathbf{o} + t\mathbf{d}$$

$$Plano: \quad (\mathbf{p}(t_i) - \mathbf{p}_1) \cdot \mathbf{n} = 0$$

$$(\mathbf{o} + t_i \mathbf{d} - \mathbf{p}_1) \cdot \mathbf{n} = 0$$

$$t_i \mathbf{d} \cdot \mathbf{n} + (\mathbf{o} - \mathbf{p}_1) \cdot \mathbf{n} = 0$$

$$t_i = \frac{(\mathbf{p}_1 - \mathbf{o}) \cdot \mathbf{n}}{\mathbf{d} \cdot \mathbf{n}}$$



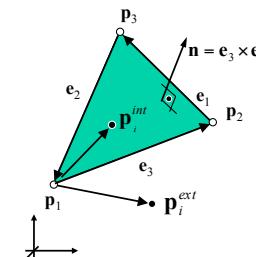
$$\boxed{\mathbf{p}_i = \mathbf{p}(t_i) = \mathbf{o} + t_i \mathbf{d}}$$

MGattass

### Ponto interno a triângulo

$$\mathbf{n} \cdot (\mathbf{e}_3 \times (\mathbf{p}_{i_{int}} - \mathbf{p}_1)) > 0$$

$$\mathbf{n} \cdot (\mathbf{e}_3 \times (\mathbf{p}_i^{ext} - \mathbf{p}_1)) < 0$$



MGattass

### Coordenadas baricêntricas

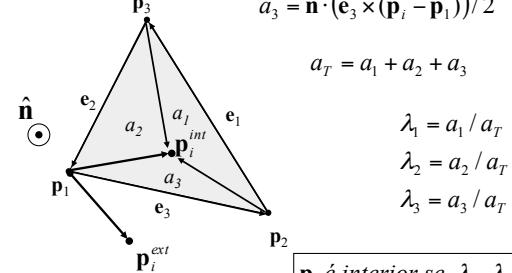
$$\hat{\mathbf{n}} = \text{unit}(\mathbf{e}_3 \times \mathbf{e}_1)$$

$$a_1 = \hat{\mathbf{n}} \cdot (\mathbf{e}_1 \times (\mathbf{p}_i - \mathbf{p}_2)) / 2$$

$$a_2 = \hat{\mathbf{n}} \cdot (\mathbf{e}_2 \times (\mathbf{p}_i - \mathbf{p}_3)) / 2$$

$$a_3 = \hat{\mathbf{n}} \cdot (\mathbf{e}_3 \times (\mathbf{p}_i - \mathbf{p}_1)) / 2$$

$$a_T = a_1 + a_2 + a_3$$

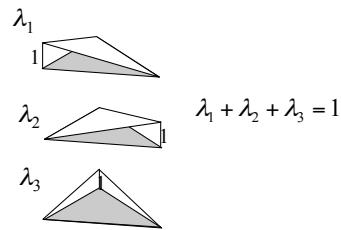
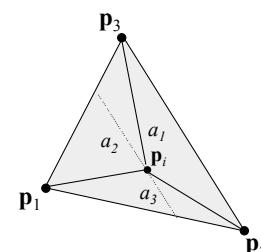


$$\boxed{\mathbf{p}_i \text{ é interior se } \lambda_1, \lambda_2 \text{ e } \lambda_3 \in [0..1]}$$

MGattass

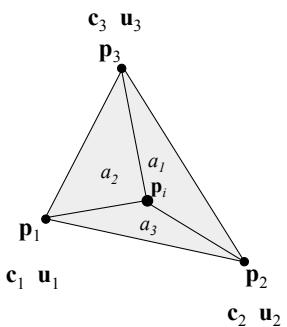
### Coordenadas baricêntricas como funções interpolantes

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \lambda_1 \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + \lambda_2 \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} + \lambda_3 \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}$$



MGattass

### Interpolação de cor e coordenada de textura através das coordenadas baricêtricas



$$\begin{Bmatrix} r_i \\ g_i \\ b_i \end{Bmatrix} = \lambda_1 \begin{Bmatrix} r_1 \\ g_1 \\ b_1 \end{Bmatrix} + \lambda_2 \begin{Bmatrix} r_2 \\ g_2 \\ b_2 \end{Bmatrix} + \lambda_3 \begin{Bmatrix} r_3 \\ g_3 \\ b_3 \end{Bmatrix}$$

ou:

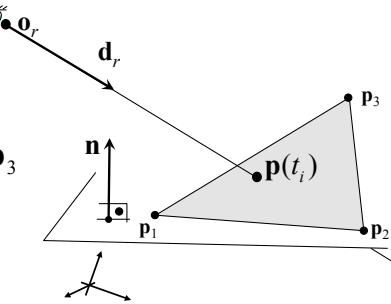
$$\begin{Bmatrix} u_i \\ v_i \end{Bmatrix} = \lambda_1 \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} + \lambda_2 \begin{Bmatrix} u_2 \\ v_2 \end{Bmatrix} + \lambda_3 \begin{Bmatrix} u_3 \\ v_3 \end{Bmatrix}$$

MGattass

### Forma otimizada de calculo

$$\mathbf{p}_i = \mathbf{o}_r + t_i \mathbf{d}_r$$

$$\mathbf{p}_i = \lambda_1 \mathbf{p}_1 + \lambda_2 \mathbf{p}_2 + \lambda_3 \mathbf{p}_3$$



$$\mathbf{p}_i = \mathbf{o}_r + t_i \mathbf{d}_r = (1 - \lambda_2 - \lambda_3) \mathbf{p}_1 + \lambda_2 \mathbf{p}_2 + \lambda_3 \mathbf{p}_3$$

Tomas Möller, Ben Trumbore  
Fast, Minimum Storage Ray/Triangle Intersection

### Forma otimizada de cálculo

$$\mathbf{o}_r + t_i \mathbf{d}_r = (1 - \lambda_2 - \lambda_3) \mathbf{p}_1 + \lambda_2 \mathbf{p}_2 + \lambda_3 \mathbf{p}_3$$

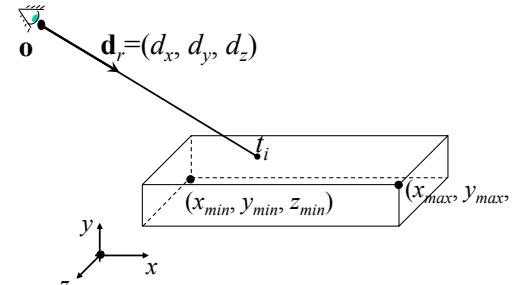
$$-t_i \mathbf{d}_r + \lambda_2 (\mathbf{p}_2 - \mathbf{p}_1) + \lambda_3 (\mathbf{p}_3 - \mathbf{p}_1) = \mathbf{o}_r - \mathbf{p}_1$$

$$[-\mathbf{d}_r \quad (\mathbf{p}_2 - \mathbf{p}_1) \quad (\mathbf{p}_3 - \mathbf{p}_1)] \begin{pmatrix} t_i \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = (\mathbf{o}_r - \mathbf{p}_1)$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

MGattass

### Caixa alinhada com os eixos

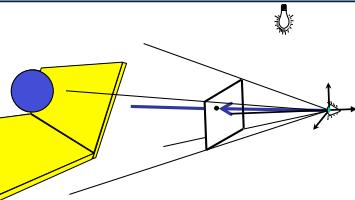


$$d_x > 0 \Rightarrow x = x_{\min}$$

$$d_y < 0 \Rightarrow y = y_{\max}$$

$$d_z < 0 \Rightarrow z = z_{\max}$$

### Motivação: Uma cena simples



Camera:

**eye** = (100,40,40), **center** = (0,0,0), **up**=(0,1,0), **fov**=90°, **near** = 30, **far**=230,  
**w**=230, **h**=230.

Esfera:

**c** = (0,20,0), **r** = 25, cor azul = (0,0,1)

Caixas alinhadas com os eixos:

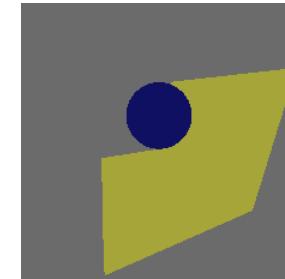
**p**<sub>0</sub> = (-80,-50,-50), **p**<sub>1</sub> = (50,-45,50) e cor amarela = (0.7,0.7,0)  
**p**<sub>0</sub> = (-80,-50,-60), **p**<sub>1</sub> = (50,50,-50) e cor amarela = (0.7,0.7,0)

Luz Pontual:

Posição=(60,120,40) e intensidade RGB **l**=(0.8,0.8,0.8)

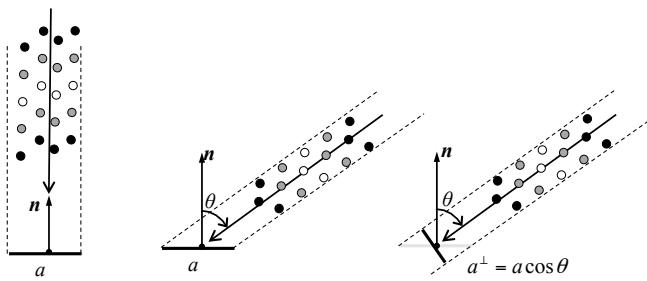
MGattass

O que conseguiríamos se simplesmente atribuissemos aos *pixels* a cor dos objetos?



MGattass

### Área aparente



MGattass

### Área aparente

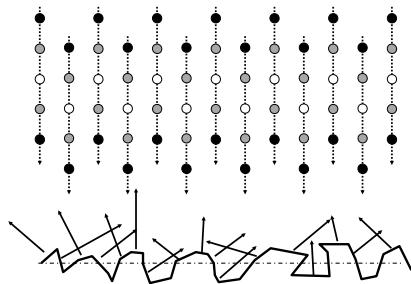
$$\frac{\text{energia luminosa}}{\text{luminosa}} \rightarrow l \text{ (lumens)} \quad |a| \quad i = \frac{l}{a} \text{ (lumens / m}^2\text{)}$$

$$\frac{\text{energia luminosa}}{\text{luminosa}} \rightarrow l \text{ (lumens)} \quad \sqrt{a'} \quad i' = \frac{l}{a'} = \frac{l}{a} \cos \theta \text{ (lumens / m}^2\text{)}$$

$$a' = \frac{a}{\cos \theta}$$

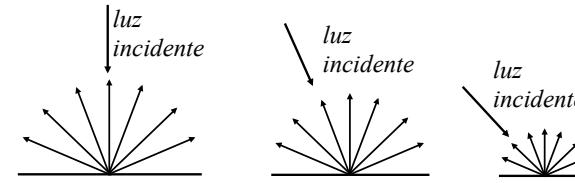
MGattass

### Espalhamento dos fótons



MGattass

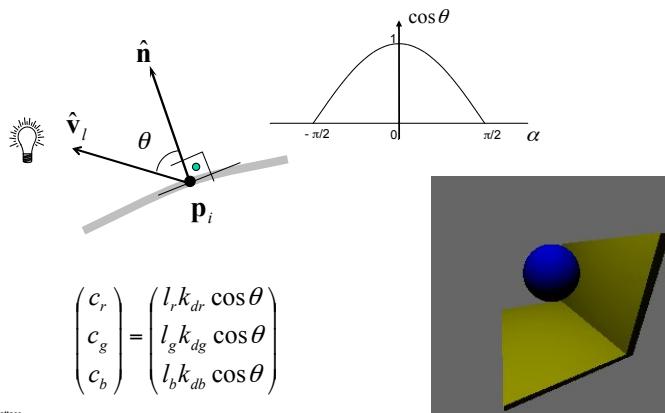
### Modelo de reflexão de superfícies Lambertianas



1. Reflete igualmente em todas as direções
2. A intensidade é proporcional ao co-seno

MGattass

### Componente de reflexão difusa



MGattass

### Outras maneiras de se escrever:

$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_r k_{dr} \cos \theta \\ l_g k_{dg} \cos \theta \\ l_b k_{db} \cos \theta \end{pmatrix}$$

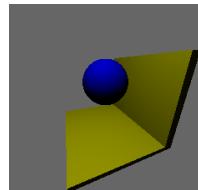
$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_r k_{dr} (\hat{n} \cdot \hat{v}_l) \\ l_g k_{dg} (\hat{n} \cdot \hat{v}_l) \\ l_b k_{db} (\hat{n} \cdot \hat{v}_l) \end{pmatrix} = \begin{pmatrix} l_r k_{dr} \\ l_g k_{dg} \\ l_b k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l) = \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l) \quad \text{Eq. 1}$$

$$c, l, k \in [0, 1]$$

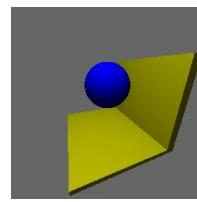
MGattass

### Luz difusa mais ambiente:

$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l)$$

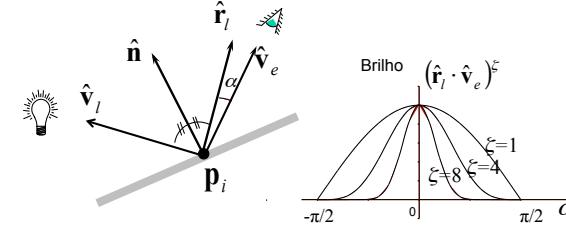


$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_{ar} \\ l_{ag} \\ l_{ab} \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} + \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l)$$



MGattass

### Componente de reflexão especular

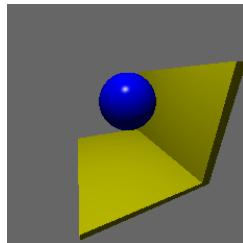


$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix}_{especular} = \begin{pmatrix} l_r k_{sr} (\cos \alpha)^z \\ l_g k_{sg} (\cos \alpha)^z \\ l_b k_{sb} (\cos \alpha)^z \end{pmatrix}$$

Eq.2

 $c, l, k \in [0, 1]$ 

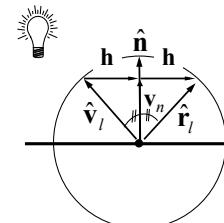
### Reflexão especular



$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_{ar} \\ l_{ag} \\ l_{ab} \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} + \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l) + \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{sr} \\ k_{sg} \\ k_{sb} \end{pmatrix} (\hat{r}_l \cdot \hat{v}_e)^z$$

MGattass

### Cálculo da reflexão de um vetor sobre outros



$$\mathbf{v}_n = (\hat{v}_l \cdot \hat{n})\hat{n}$$

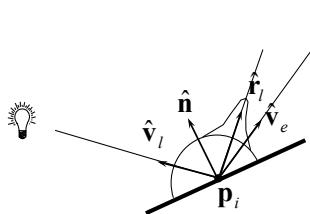
$$\mathbf{h} = \mathbf{v}_n - \hat{v}_l$$

$$\hat{r}_l = \mathbf{v}_n + \mathbf{h}$$

$$\hat{r}_l = 2(\hat{v}_l \cdot \hat{n})\hat{n} - \hat{v}_l$$

MGattass

### Distribuição da luz direta sobre um ponto



$$c_\lambda = c_{amb_\lambda} + c_{luz_\lambda} k_{dif_\lambda} (\hat{n} \cdot \hat{v}_l) + c_{luz_\lambda} k_{s_\lambda} (\hat{r}_l \cdot \hat{v}_e)^\zeta$$

MGattass

### Modelo de várias luzes

$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_{ar} \\ l_{ag} \\ l_{ab} \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} + \sum_{\text{luzes}} \left( \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l) + \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{sr} \\ k_{sg} \\ k_{sb} \end{pmatrix} (\hat{r}_e \cdot \hat{v}_l)^\zeta \right)$$

$$\hat{r}_l = 2(\hat{v}_l \cdot \hat{n})\hat{n} - \hat{v}_l \quad \text{para cada fonte de luz}$$

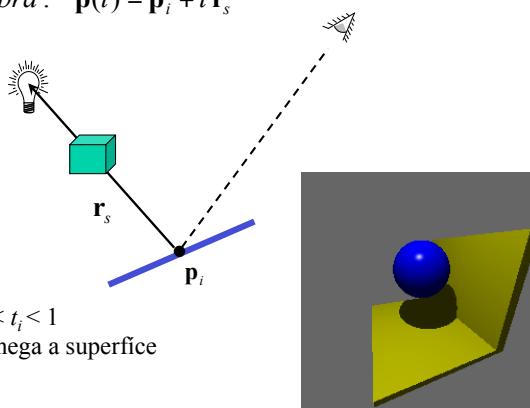
$$\hat{r}_e = 2(\hat{v}_e \cdot \hat{n})\hat{n} - \hat{v}_e \quad \text{uma reflexão apenas}$$

$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_{ar} \\ l_{ag} \\ l_{ab} \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} + \sum_{\text{luzes}} \left( \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l) + \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{sr} \\ k_{sg} \\ k_{sb} \end{pmatrix} (\hat{r}_e \cdot \hat{v}_l)^\zeta \right)$$

MGattass

### Sombra

$$\text{Raio de sombra: } \mathbf{p}(t) = \mathbf{p}_i + t \mathbf{r}_s$$



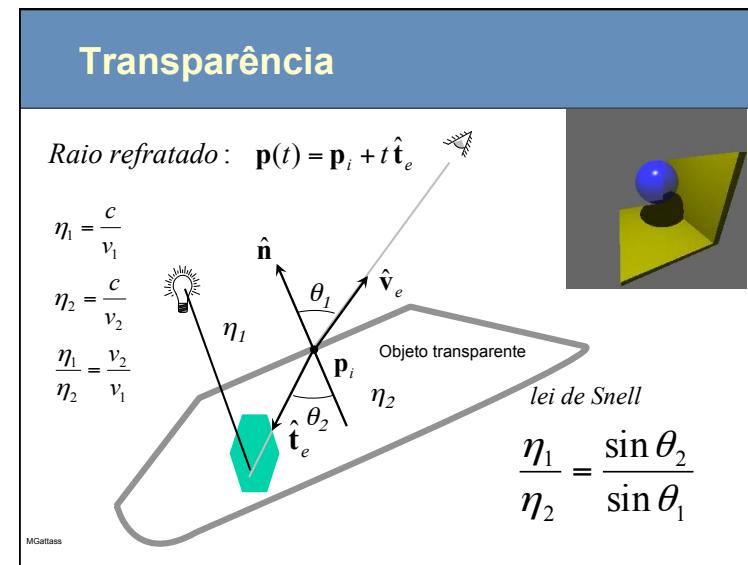
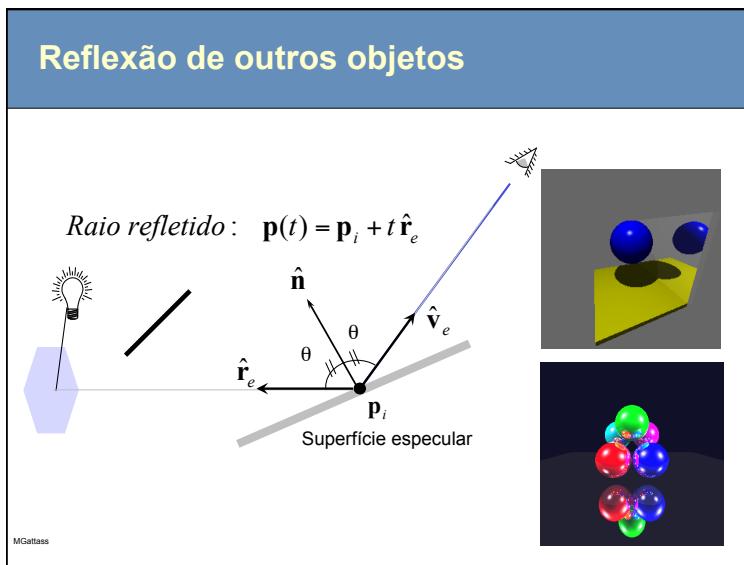
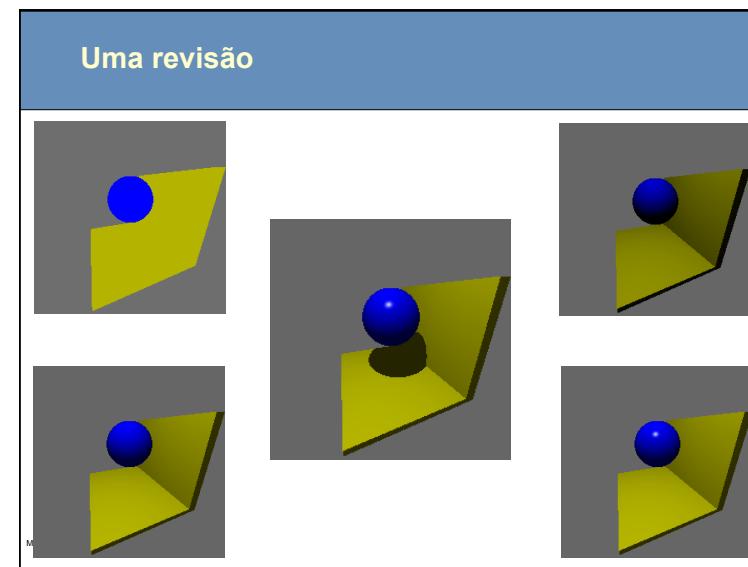
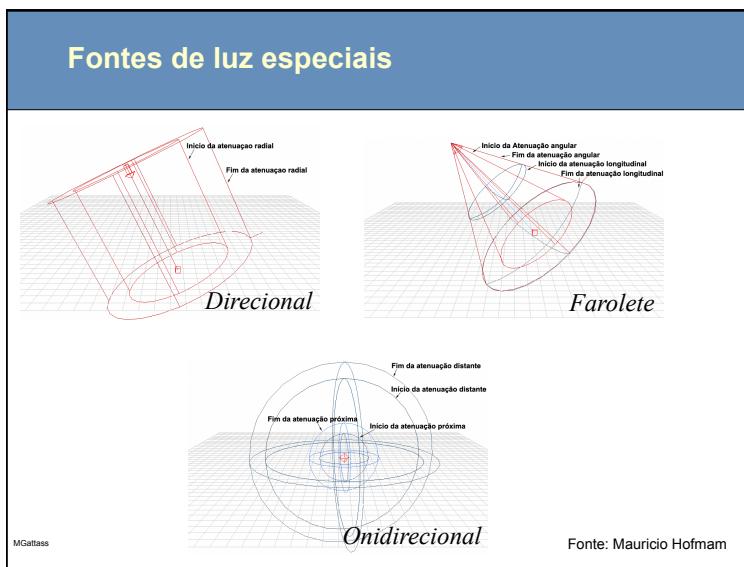
MGattass

### Modelo de várias luzes e sombra

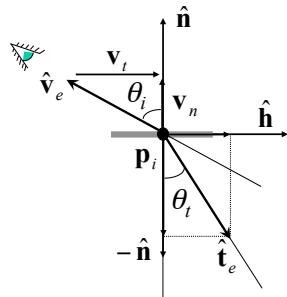
$$\begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} = \begin{pmatrix} l_{ar} \\ l_{ag} \\ l_{ab} \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} + \sum_{\text{luzes}} f_s \left( \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{n} \cdot \hat{v}_l) + \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{sr} \\ k_{sg} \\ k_{sb} \end{pmatrix} (\hat{r}_e \cdot \hat{v}_l)^\zeta \right)$$

$$f_s = \begin{cases} 0 & \text{se sombra} \\ 1 & \text{caso contrário} \end{cases}$$

MGattass



### Cálculo do Raio Refratado



$$\begin{aligned} \mathbf{v}_t &= (\hat{\mathbf{v}}_e \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} - \hat{\mathbf{v}}_e \\ \sin \theta_i &= \|\mathbf{v}_t\| \\ \sin \theta_t &= \frac{n_l}{n_t} \sin \theta_i \\ \cos \theta_t &= \sqrt{1 - \sin^2 \theta_t} \\ \hat{\mathbf{h}} &= \frac{1}{\|\mathbf{v}_t\|} \mathbf{v}_t \\ \hat{\mathbf{t}}_e &= \sin \theta_t \hat{\mathbf{h}} - \cos \theta_t \hat{\mathbf{n}} \end{aligned}$$

Raio refratado :  $\mathbf{p}(t) = \mathbf{p}_i + t \hat{\mathbf{t}}_e$

MGattass

### Advertência: Refração não é simples!



MGattass

### Illuminação considerando superfícies refletoras e objetos transparentes

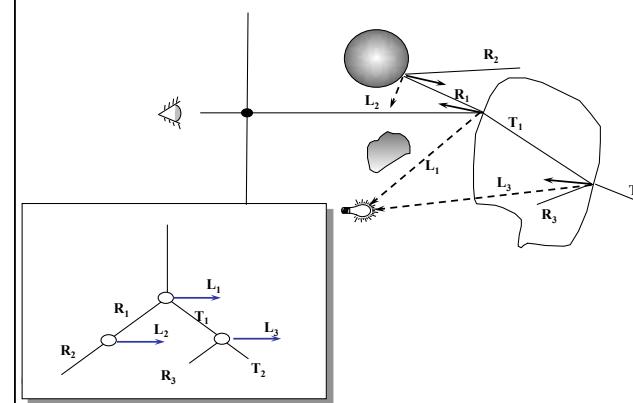
$$\begin{aligned} \begin{pmatrix} c_r \\ c_g \\ c_b \end{pmatrix} &= \begin{pmatrix} l_{ar} \\ l_{ag} \\ l_{ab} \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} + \\ &\sum_{\text{luzes}} f_s \left( \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{dr} \\ k_{dg} \\ k_{db} \end{pmatrix} (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}_l) + \begin{pmatrix} l_r \\ l_g \\ l_b \end{pmatrix} \otimes \begin{pmatrix} k_{sr} \\ k_{sg} \\ k_{sb} \end{pmatrix} (\hat{\mathbf{r}}_e \cdot \hat{\mathbf{v}}_l)^c \right) + \\ &s \begin{pmatrix} c_r(\hat{\mathbf{r}}_e) \\ c_g(\hat{\mathbf{r}}_e) \\ c_b(\hat{\mathbf{r}}_e) \end{pmatrix} + (1-o) \begin{pmatrix} c_r(\hat{\mathbf{r}}_t) \\ c_g(\hat{\mathbf{r}}_t) \\ c_b(\hat{\mathbf{r}}_t) \end{pmatrix} \end{aligned}$$

redução da reflexão

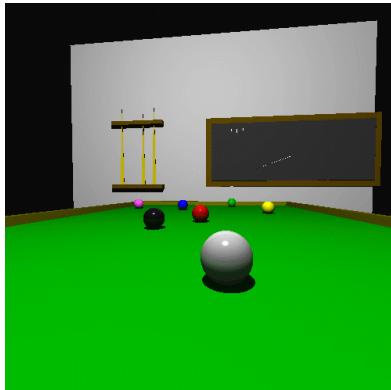
redução da transparência

MGattass

### Natureza recursiva do algoritmo de Rastreamento de Raios



## Resultado de curso



Alunos de CGI98

MGattass

## Algoritmo de traçado de raios

```
selecione o centro de projeção(eye) e uma janela no plano de projeção
for (cada pixel da tela)
{
    determine o raio ray que vai do centro de projeção ao pixel;
    pixel = trace ( ray, 1);
}
```

```
Color trace (Scene scene, Vector3d eye, Vector3d ray, int depth)
{
    determine a intersecção mais próxima com um objeto
    if (intercepta objeto)
    {
        calcule a normal no ponto de intersecção
        return ( shade ( scene, object, ray, point, normal, depth));
    }
    return BACKGROUND;
}
```

MGattass

```
Color shade (Scene scene, Object object, Vector3D ray,
            Vector3D point, Vector3D normal, int depth)
{
    color = termo ambiente do material do objeto ;

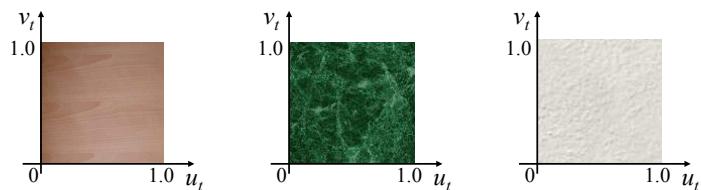
    for (cada luz) {
        L = vetor unitário na direção de point para a posição da luz;
        if (L * normal>0) {
            if (a luz não for bloqueada no ponto) {
                color += componente difusa (Eq.1) + componente especular (Eq.2)
            }
        }
    }

    if (depth >= maxDepth) return color;

    if (objeto é refletor) {
        rRay = raio na direção de reflexão;
        rColor = trace(scene, point, rRay, depth+1);
        reduza rColor pelo coeficiente de reflexão especular e some a color;
    }

    return color;
}
```

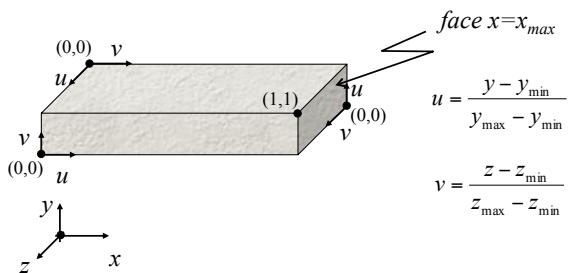
## Texturas



Texturas 2D = Imagens onde o domínio é  $u_p, v_t \in [0,1] \times [0,1] \subset \mathbb{R}^2$

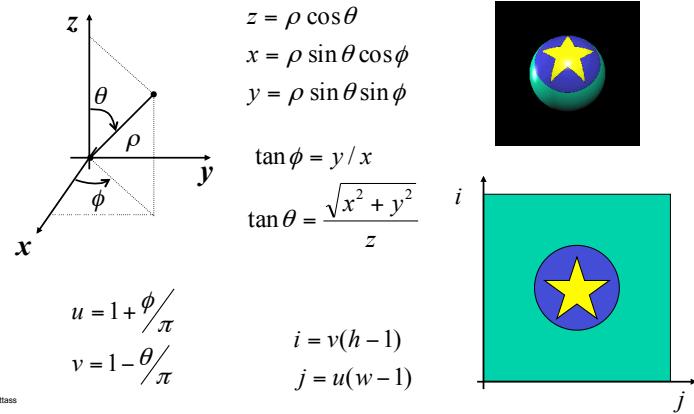
MGattass

### Sistemas de coordenada de textura na caixa



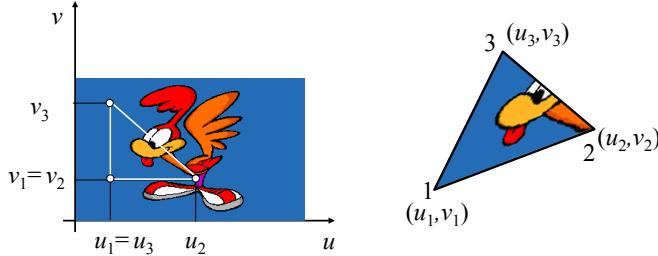
MGattass

### Sistema de coordenada de textura na esfera



MGattass

### Sistema de coordenada de textura no triângulo



MGattass

### Textura no triângulo e coordenadas baricêntricas

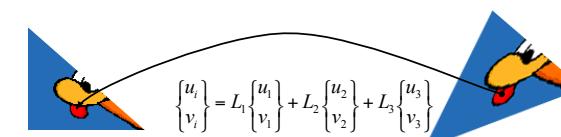
$\hat{\mathbf{n}} = \text{unit}(\mathbf{e}_3 \times \mathbf{e}_1)$

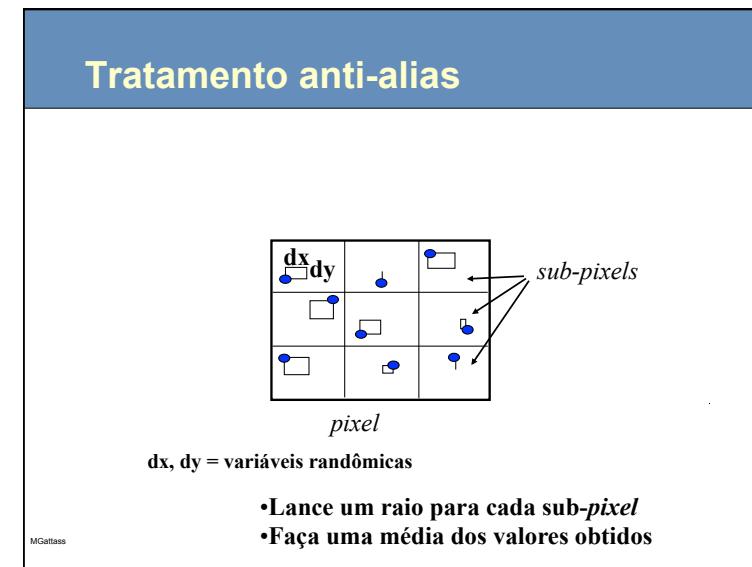
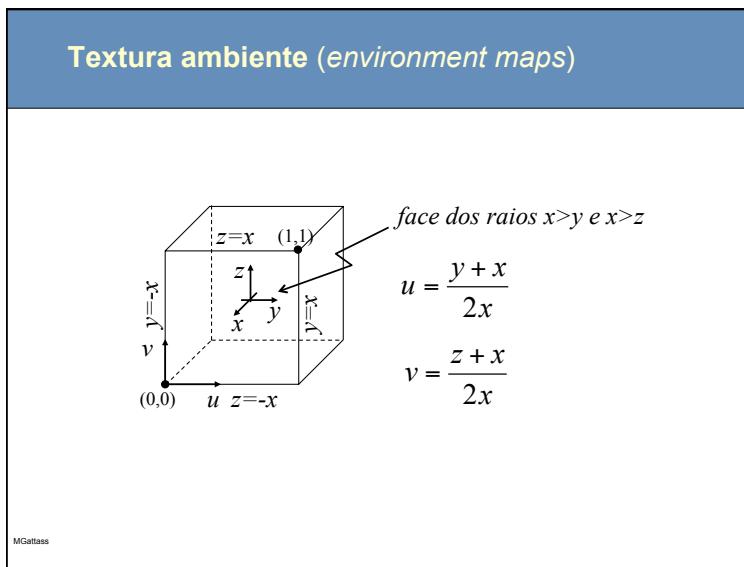
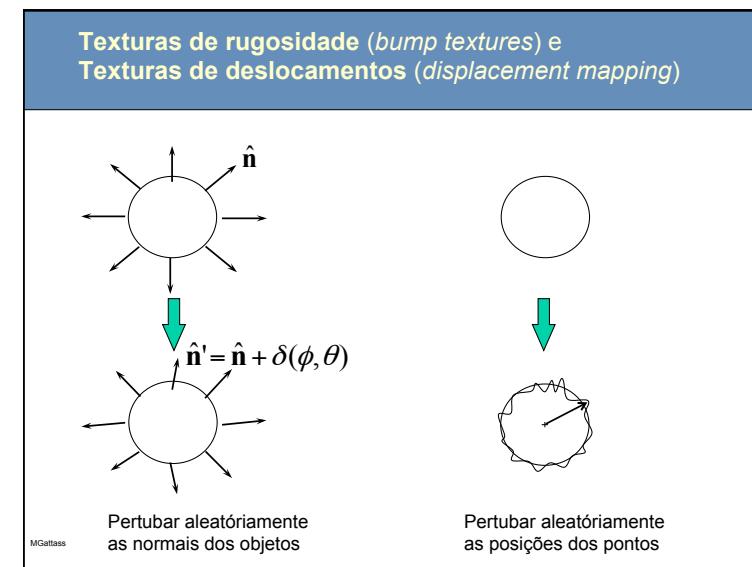
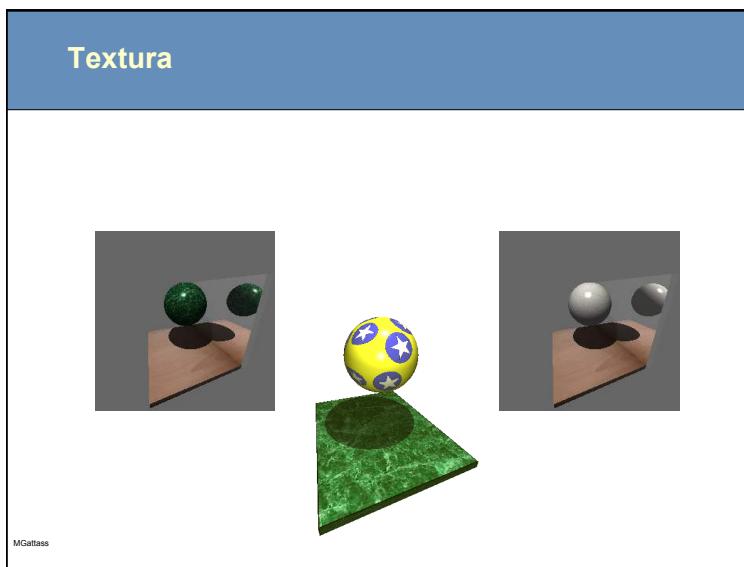
$a_1 = \hat{\mathbf{n}} \cdot (\mathbf{e}_1 \times (\mathbf{p}_i - \mathbf{p}_2)) / 2$   
 $a_2 = \hat{\mathbf{n}} \cdot (\mathbf{e}_2 \times (\mathbf{p}_i - \mathbf{p}_3)) / 2$   
 $a_3 = \hat{\mathbf{n}} \cdot (\mathbf{e}_3 \times (\mathbf{p}_i - \mathbf{p}_1)) / 2$

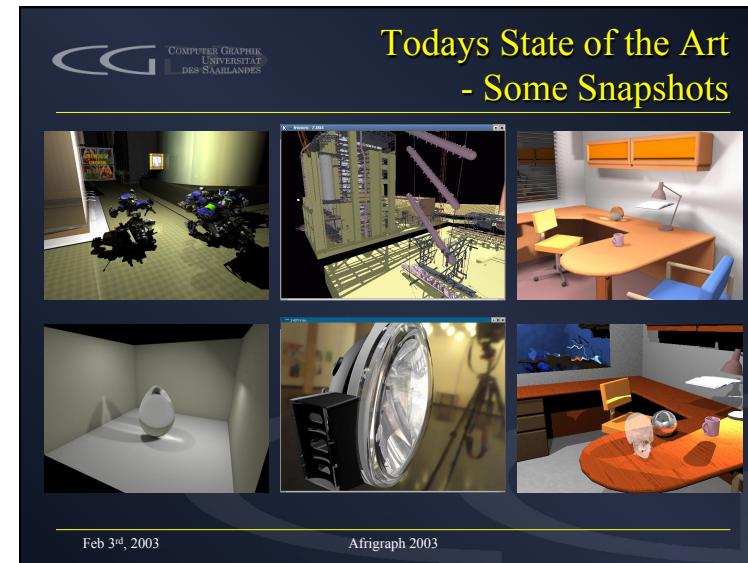
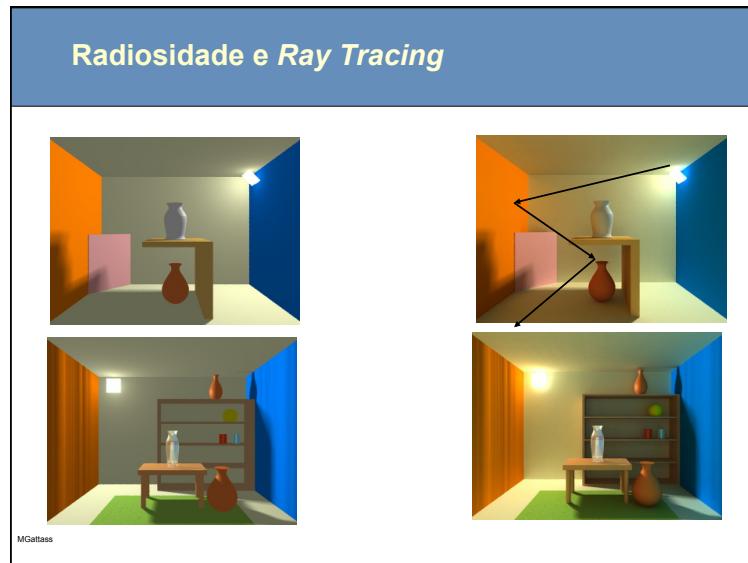
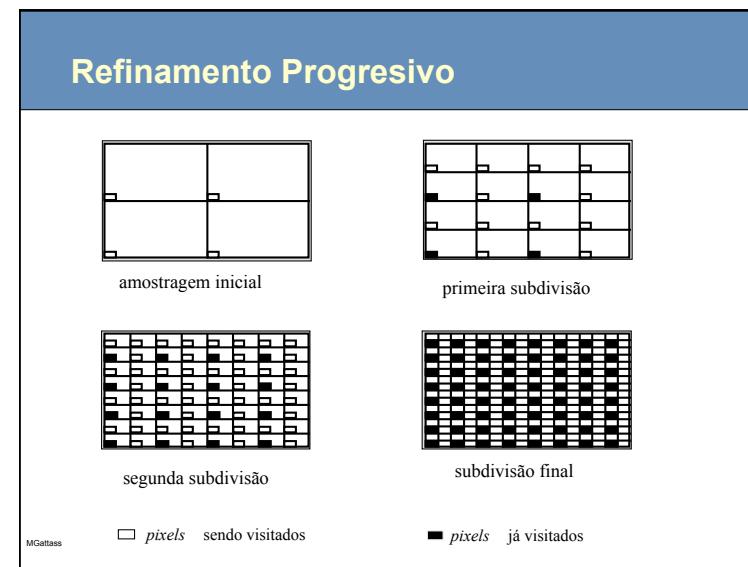
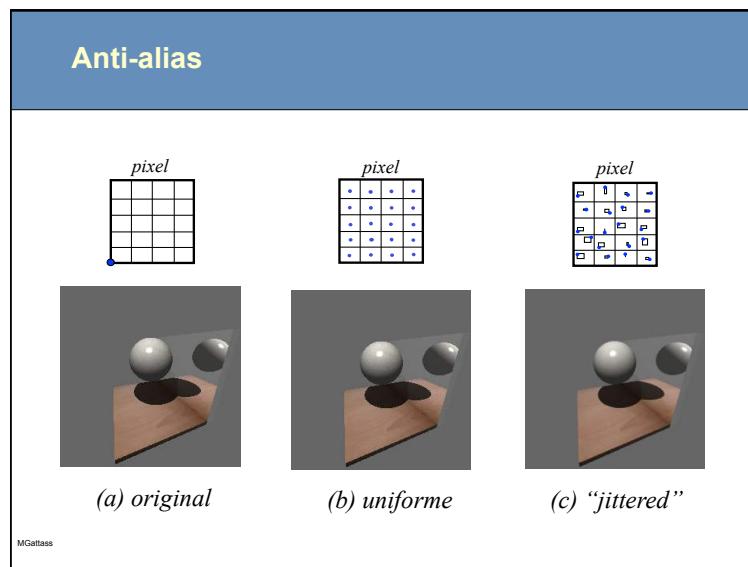
$\lambda_1 = a_1 / a_T$   
 $\lambda_2 = a_2 / a_T$   
 $\lambda_3 = a_3 / a_T$

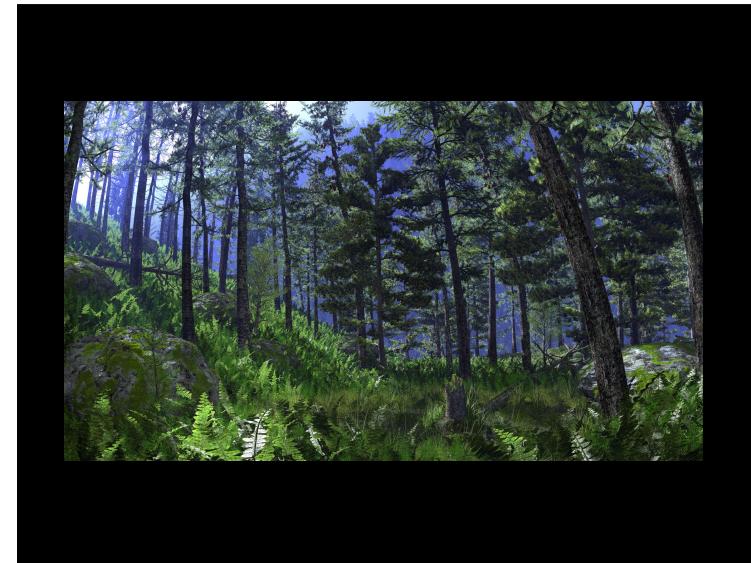
$a_T = a_1 + a_2 + a_3$

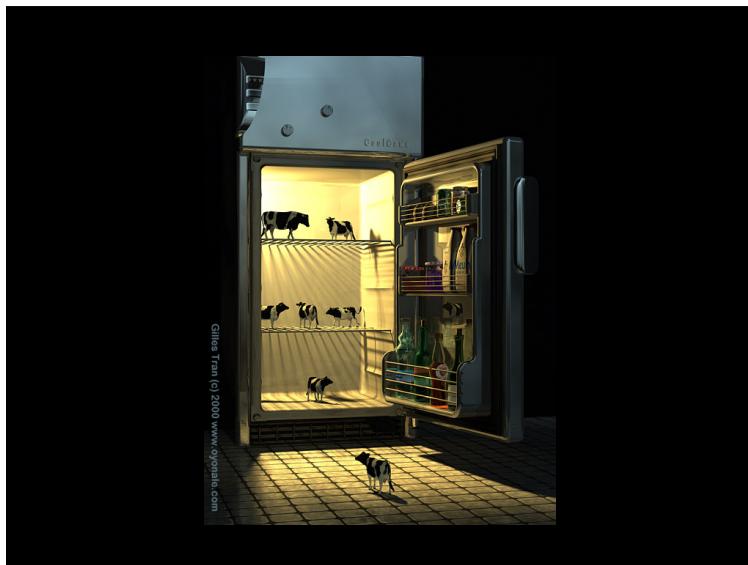
MGattass









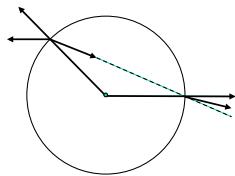


## Aceleracao do RT

- Cálculos mais eficientes
- Uso de uma Kd Tree

MGattass

## Refração na esfera



MGattass

## Ray Tracing