Mestrado Integrado
Engenharia Informática

Path Tracing

Iluminação e Visualização II

Luís Paulo Peixoto dos Santos

Monte Carlo *ray tracing*

$$\left\langle L(p \to \omega_r) \right\rangle = L_e(p \to \omega_r) + L_d(p \to \omega_r) + \frac{\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i)$$

$\left\langle L(p \to \omega_r) \right\rangle$ Radiância reflectida por *p* na direcção ω_r

$L_e(p \to \omega_r)$ Radiância auto-emitida por *p* na direcção ω_r

$L_d(p \to \omega_r)$ Radiância reflectida por *p* na direcção ω_r devida a iluminação directa

$$L_{ind}(p \to \omega_r) = \frac{\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i)$$

Radiância reflectida por *p* na direcção ω_r devida a iluminação indirecta;
Selecção dos ω_i pesada pelo co-seno.

Monte Carlo *ray tracing*

- A amostragem segue uma pdf $p(\omega_i) = {}^{\cos(\theta_i)}/_{\pi}$ obtendo-se

$$L_{ind}(p \to \omega_r) = \frac{\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i)$$

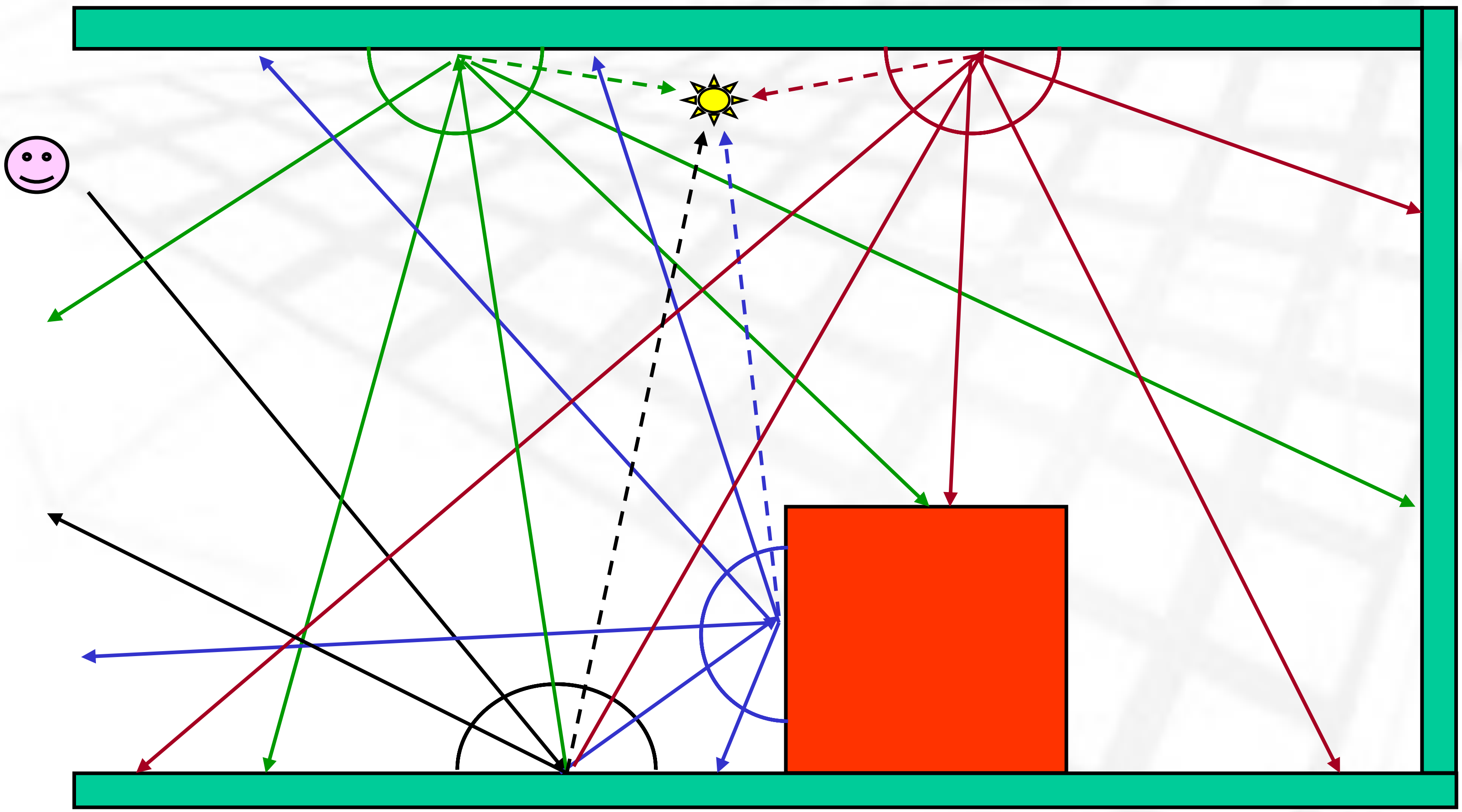
- Note que $$L_{ind}(p \to \omega_r) = \frac{2\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(\theta_i)$$

  corresponde a amostragem uniforme da semiesfera com $p(\omega_i) = {}^{1}/_{2\pi}$

- No caso geral

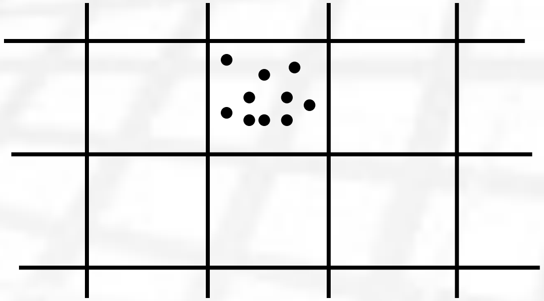
$$L_{ind}(p \to \omega_r) = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(\theta_i)}{p(\omega_i)}$$

Monte Carlo *ray tracing*



Monte Carlo *path tracing*

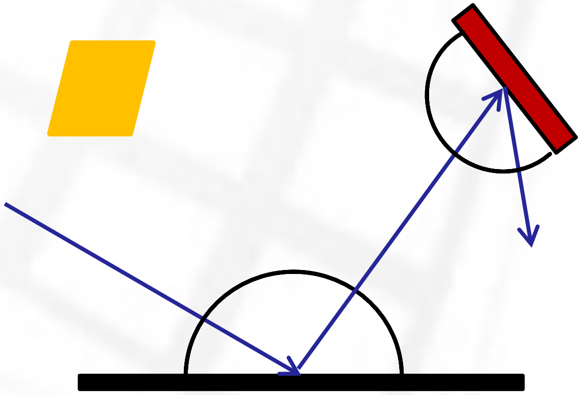
- Para cada pixel gerar N raios primários cujas direcções são seleccionadas estocasticamente sobre a área do pixel



- O valor do pixel resulta da integração de Monte Carlo dos vários *paths*

$$L_{pixel} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{L_{point\ in\ pixel_i}}{p(point\ in\ pixel_i)}$$

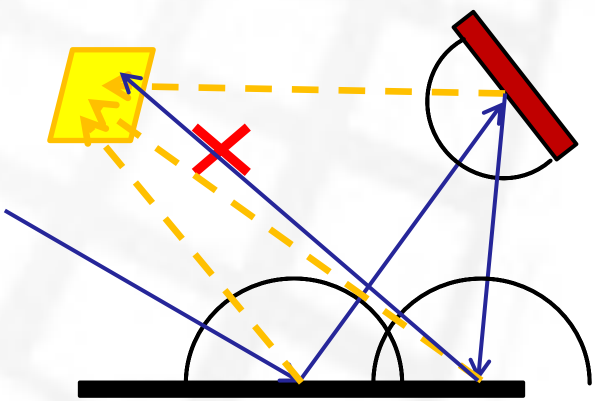
- Quando um raio intersecta um objecto:
  - Seleccionar estocasticamente uma única direcção de amostragem na semi-esfera
  - Continuar este procedimento gerando um *path*



Monte Carlo *path tracing*

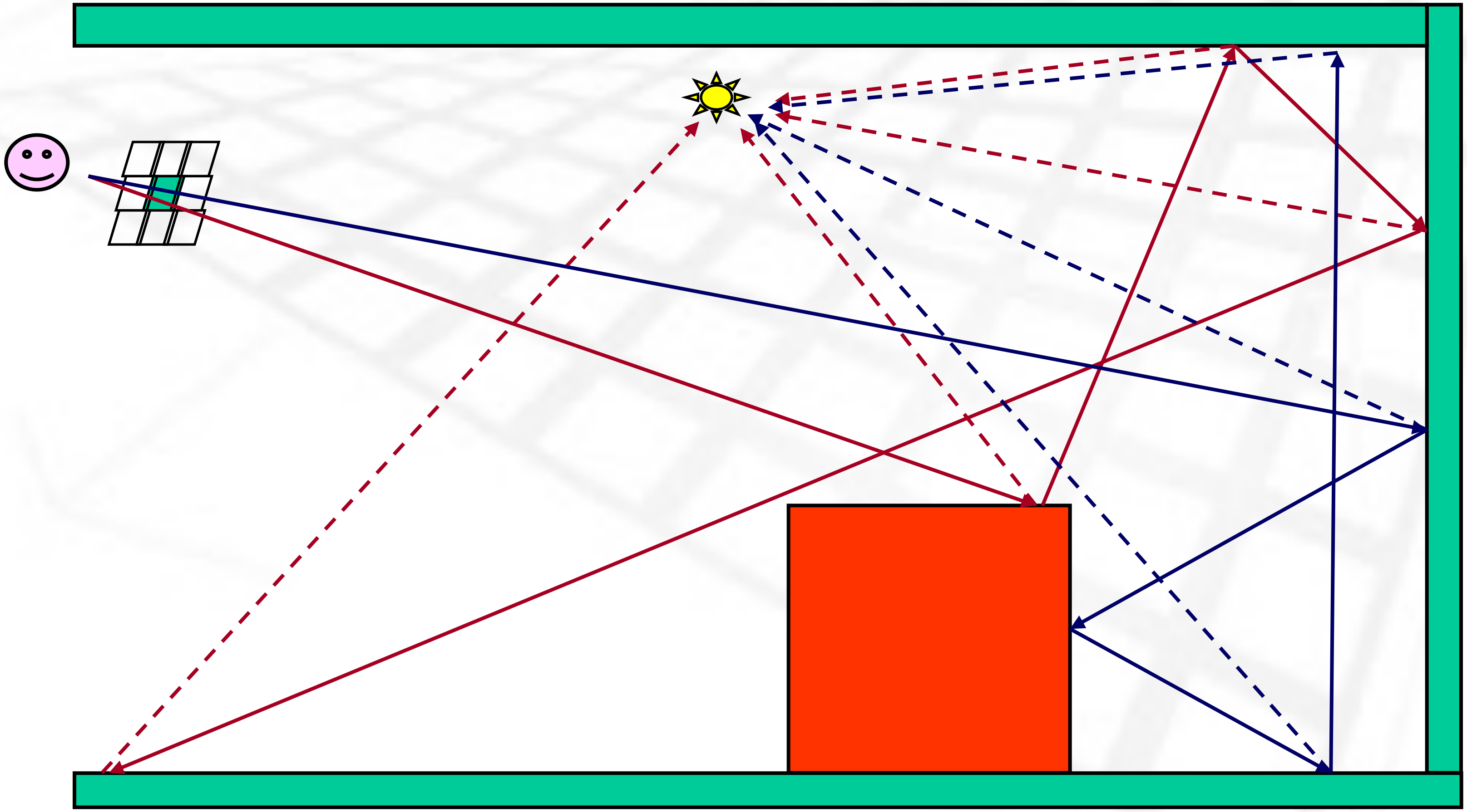
- Na descrição anterior um *path* só tem contribuição diferente de zero se intersectar uma fonte de luz.
- Para **aumentar a convergência** o cálculo da iluminação **directa** pode ser **separado** do cálculo da iluminação **indirecta**, disparando *shadow rays* explicitamente na direcção das fontes de luz

- Quando um raio intersecta um objecto:
  - Disparar *shadow rays* para calcular a iluminação directa
  - Seleccionar estocasticamente uma única direcção de amostragem na semi-esfera



- Se o *path* intersectar uma fonte de luz terminar, pois a iluminação directa já foi calculada

Monte Carlo *path tracing*



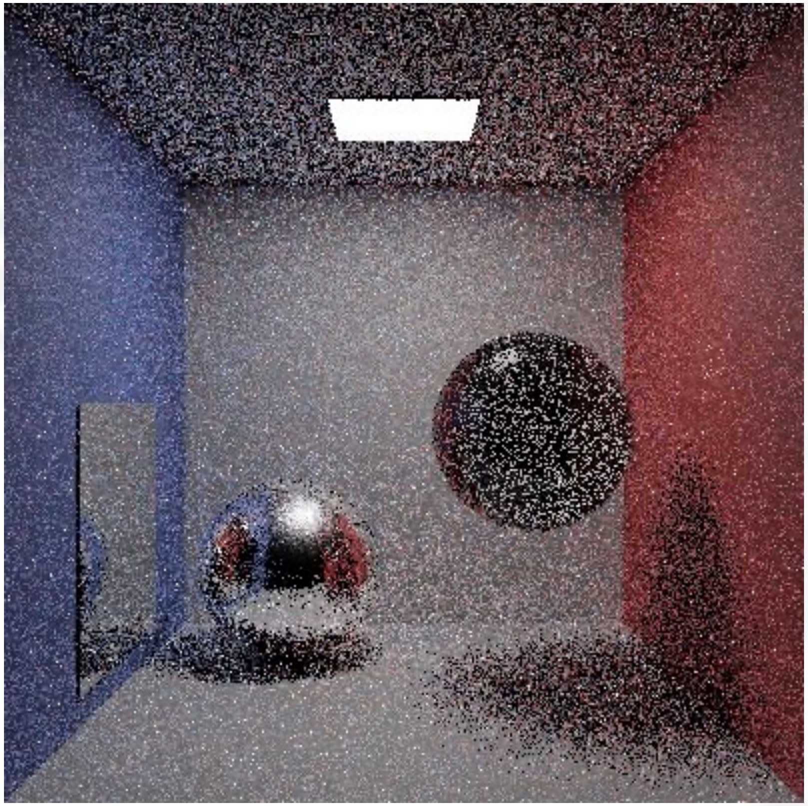
Monte Carlo *path tracing*

```
for each pixel p on the image plane  {
  rad[p]=0
  for (i=0 ; i<N ; i++)
    ray = select_dir_stochastically (p, pdf, &prob_this_dir)
    rad[p] += path_trace (ray, 0) / prob_this_dir
  rad[p] /= N;
}

path_trace (ray, depth) {
  point = intersect (ray, scene)
  rad = direct_lighting (point)   // trace shadow rays
  if (depth < MAX_DEPTH)
    sec_ray = select_dir_stochastically (p, pdf, &prob)
    rad += BRDF * path_trace (sec_ray, depth++) * cos(θ) / prob
  return (rad)
}
```

Integração de Monte Carlo : Variância

- O processo estocástico introduz variância, percepcionada como ruído



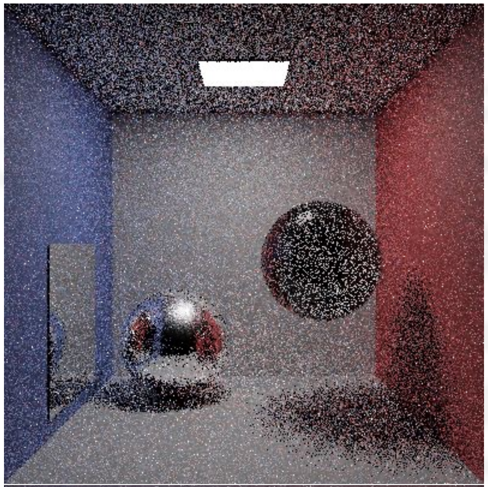
Cornell box
Path Tracing (48 cores)
1spp; 0.1 seg

- A variância varia com 1/N, o desvio padrão com $1/N^2$
- Para reduzir o ruído a metade, precisamos de 4 vezes mais amostras (N*4)

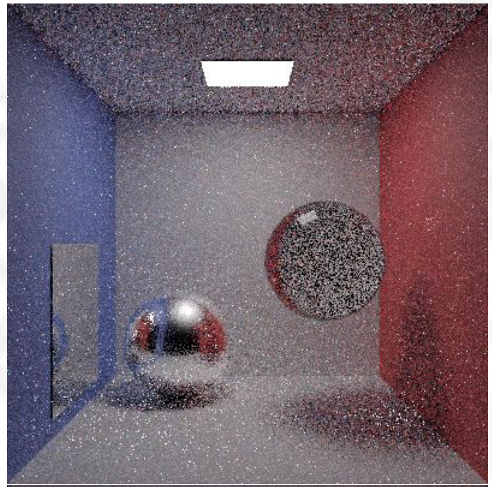
Cornell Box PT: Variância e Número de Amostras

(48 cores)

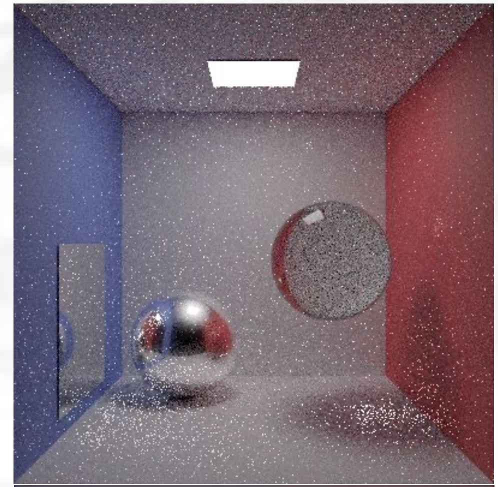
1 spp ; 0.1 seg

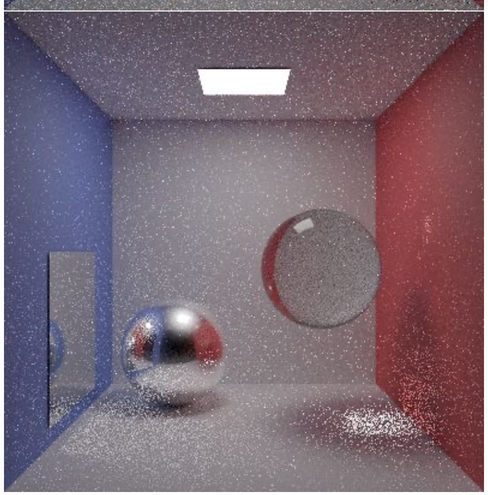


4 spp ; 0.3 seg

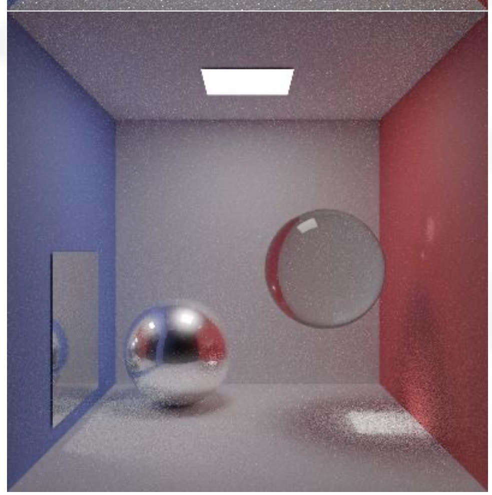


16 spp ; 1.1 seg

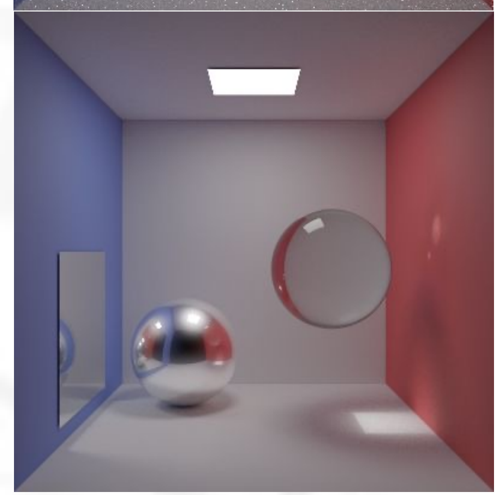




64 spp ; 4.5 seg



1024 spp ; 1m 14 seg



65536 spp ; 1h 17 m

Monte Carlo: bias

- Quando parar a emissão de raios secundários *(path length)?*
  - Usar uma profundidade máxima fixa
  - Quando a contribuição esperada de um raio é inferior a um dado limite
- Estes são métodos determinísticos que afectam o valor do integral (*bias)*!

not biased	*biased*
$\lim_{N \to \infty} \langle I \rangle = I$	$\lim_{N \to \infty} \langle I \rangle = I + \varepsilon$

Monte Carlo: roleta russa

- Definir a probabilidade p_{cont} de continuar a travessia (disparar um raio secundário)
- Antes de disparar um raio gerar um número aleatório, ξ, uniformemente distribuído em [0, 1[
- Se $\xi <= p_{cont}$ então disparar o raio
- Se $\xi > p_{cont}$ então não disparar o raio
- Uma vez que há uma probabilidade de não disparar um raio, a contribuição dos raios disparados deve ser multiplicada por 1/ p_{cont}, para compensar aqueles que não são disparados

$$L(p \leftarrow \Psi) \approx \frac{1}{p_{cont}} (\xi \leq \alpha \ ? \ 0 : L(p \leftarrow \Psi_i))$$

Monte Carlo: roleta russa

```
for each pixel p on the image plane  {
  i=0
  for (i=0 ; i<N ; i++)
    ray = select_dir_stochastically (p, pdf, &prob)
    rad[p] += path_trace (ray) / prob
  rad[p] /= N;
}

path_trace (ray) {
  point = intersect (ray, scene)
  rad = direct_lighting (point)    // trace shadow rays
  if (ξ <= p_cont)
    sec_ray = select_dir_stochastically (p, pdf, &prob)
    rad_incident = path_trace (sec_ray) / ( prob * p_cont)
    rad += BRDF * rad_incident * cos(θ)
  return (rad)
}
```

Integração de Monte Carlo

$$\boxed{\langle L(p \to \omega_r) \rangle = L_e(p \to \omega_r) + L_d(p \to \omega_r) + L_{ind}(p \to \omega_r)}$$

$\langle L(p \to \omega_r) \rangle$ Radiância reflectida por *p* na direcção ω_r

$L_e(p \to \omega_r)$ Radiância auto-emitida por *p* na direcção ω_r

$L_d(p \to \omega_r)$ Radiância reflectida por *p* na direcção ω_r devida a iluminação directa

$L_{ind}(p \to \omega_r)$ Radiância reflectida por *p* na direcção ω_r devida a iluminação indirecta

Monte Carlo: iluminação directa

$$L_d(p \to \omega_r) = \int_{\Omega_d} f_r(p,\omega_r,\omega_i)\, L_d(p \leftarrow \omega_i) \cos\theta_i \, d\omega_i$$

- Nota: apenas são consideradas contribuições que tenham origem directamente nas fontes de luz (NOTAR Ω_d)

Abordagem 1 – Iterar sobre **todas as fontes de luz** e tirar N_{Li} amostras de cada:

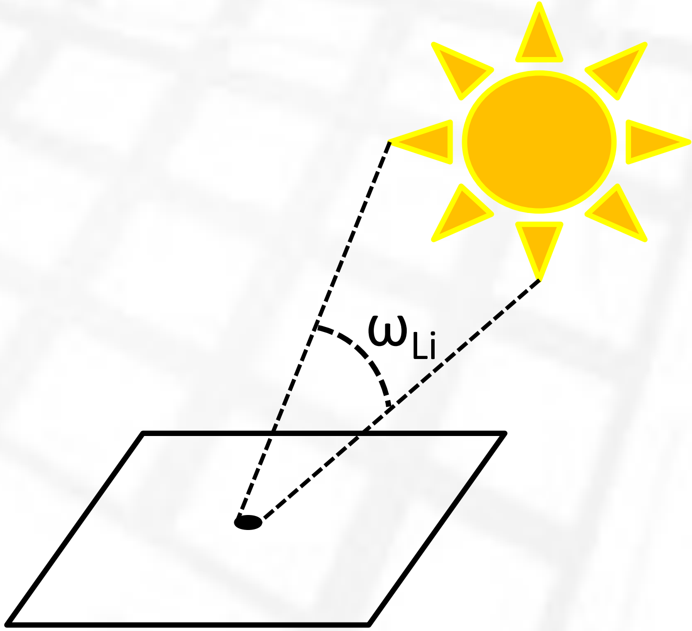
$$L_d(p \to \omega_r) = \sum_{l=1}^{Nlights} \int_{\Omega_{d(l)}} f_r(p,\omega_r,\omega_i) L_{d(l)}(p \leftarrow \omega_i) \cos\theta_i \, d\omega_i, \quad \Omega_d = \bigcup_{l=1}^{Nlights} \Omega_{d(l)}$$

$$L_d(p \to \omega_r) \approx \sum_{l=1}^{Nlights} \frac{1}{N_{Li}} \sum_{i=1}^{N_{Li}} \frac{f_r(p,\omega_r,\omega_i) L_{d(l)}(p \leftarrow \omega_i) \cos\theta_i}{p(\omega_i)}$$

Monte Carlo: iluminação directa

$$L_d(p \to \omega_r) \approx \sum_{l=1}^{Nlights} \frac{1}{N_{Li}} \sum_{i=1}^{N_{Li}} \frac{f_r(p,\omega_r,\omega_i) L_{d(l)}(p \leftarrow \omega_i) \cos\theta_i}{p(\omega_i)}$$

- A selecção das amostras ωi pode ser feita usando **distribuição uniforme** ou **estratificada** ou amostragem por **importância** sobre o **ângulo sólido definido pela fonte de luz,** $\boldsymbol{\omega_{Li}}$



Monte Carlo: iluminação directa

$$L_d(p \to \omega_r) = \int_{\Omega_d} f_r(p,\omega_r,\omega_i)\, L_d(p \leftarrow \omega_i) \cos\theta_i \, \mathrm{d}\omega_i$$

Abordagem 2 – **Seleccionar** com probabilidade $p_l = 1/N_{lights}$ **uma fonte de luz** e amostrar apenas essa.

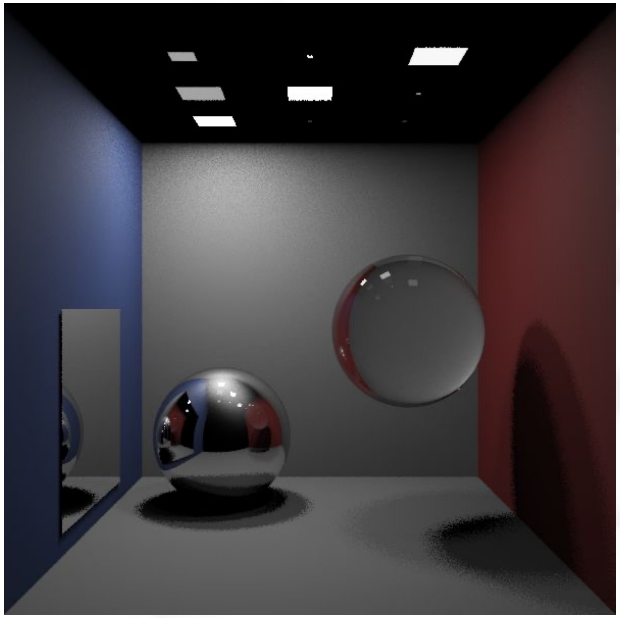
A sua contribuição é, obviamente, dividida pela sua probabilidade:

$$L_d(p \to \omega_r) \approx \frac{f_r(p,\omega_r,\omega_i) L_{d(l)}(p \leftarrow \omega_i) \cos\theta_i}{1/N_{Lights} * p(\omega_i)}$$

$$L_d(p \to \omega_r) \approx \frac{N_{Lights} * f_r(p,\omega_r,\omega_i) L_{d(l)}(p \leftarrow \omega_i) \cos\theta_i}{p(\omega_i)}$$

Em vez de seleccionar a fonte de luz com probabilidade uniforme podem-se usar outras distribuições. Exemplos: potência, área, distância, ...

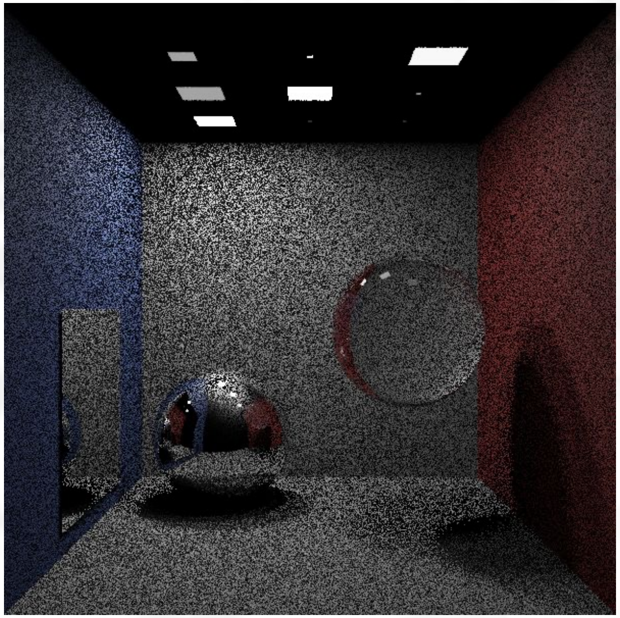
Monte Carlo: iluminação directa



8 spp, 1 spl

Todas as fontes de luz são amostradas (1 raio cada; spl=1)

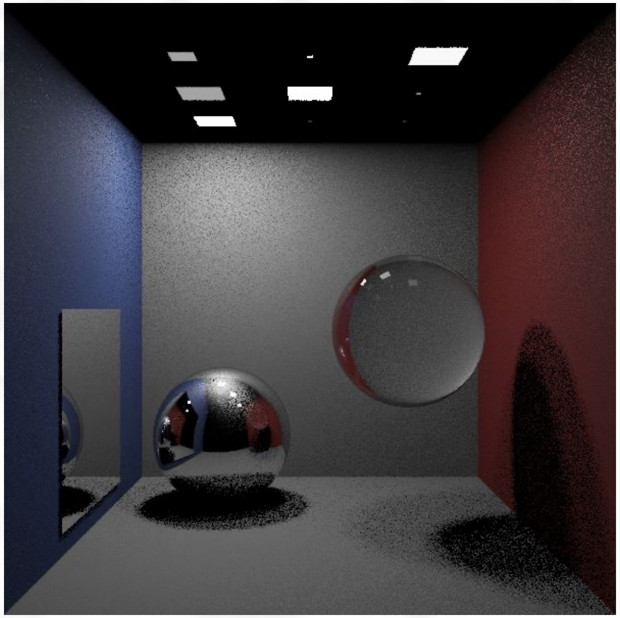
T = 8 sec



8 spp, 1/9 spl

1 fonte de luz seleccionada com distribuição uniforme

T = 4 sec



8 spp, 1/9 spl

1 fonte de luz seleccionada segundo a sua potência relativa

T = 4 sec

Monte Carlo: iluminação indirecta

$$L_{ind}(p \to \omega_r) = \int_{\Omega} f_r(p, \omega_r, \omega_i)\, L_d(p \leftarrow \omega_i) \cos\theta_i \, d\omega_i$$

- Nota: apenas são consideradas contribuições que **não** tenham origem directamente nas fontes de luz.

$$L_{ind}(p \to \omega_r) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(p, \omega_r, \omega_i) L_{ind}(p \leftarrow \omega_i) \cos\theta_i}{p(\omega_i)}$$

- A selecção de quais as direcções da semi-esfera a amostrar é vulgarmente baseada em:
  - Amostragem estratificada
  - Amostragem por importância baseada na distribuição do cosseno
  - Amostragem por importância baseada na forma da BRDF
    - Neste caso é necessário derivar uma distribuição da probabilidade que tenha uma forma semelhante à da BRDF

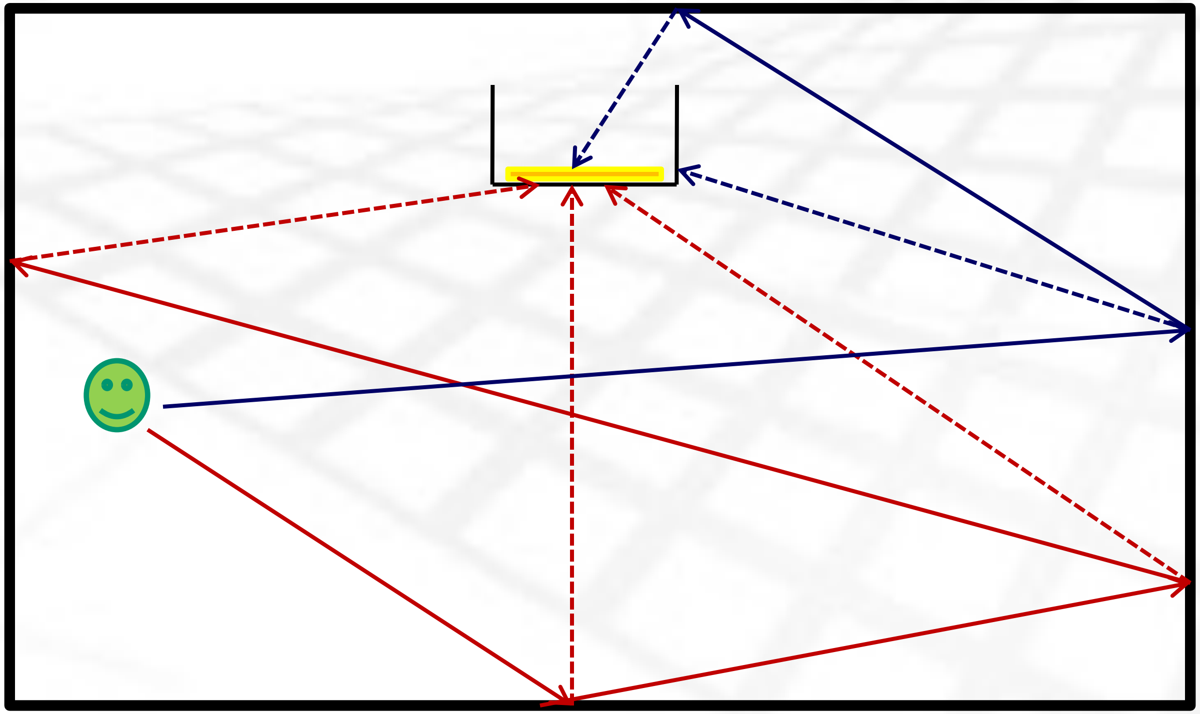
Monte Carlo: iluminação indirecta

$$L_{ind}(p \to \omega_r) = \int_{\Omega} f_r(p, \omega_r, \omega_i)\, L_d(p \leftarrow \omega_i) \cos\theta_i \, d\omega_i$$

- A BRDF é frequentemente constituída por múltiplos componentes. Ex., Phong:
  K_d – indirecta difusa; o seu domínio é a semi-esfera;
  K_s – reflexão especular; domínio é a direcção de reflexão especular;
  K_t – transmissão especular; domínio é a direcção de transmissão especular;

- Em vez de amostrar explicitamente estas 3 componentes (com 3 raios que gerarão novas árvores), um *path tracer*:
  - Selecciona com probabilidade $p_{compBRDF}$ apenas um destes componentes;
  - Avalia apenas este componente, i.e., dispara um raio apenas de acordo com as regras associadas aquele componente da BRDF;
  - Divide a radiância calculada por $p_{compBRDF}$.

Path Tracing: limitações



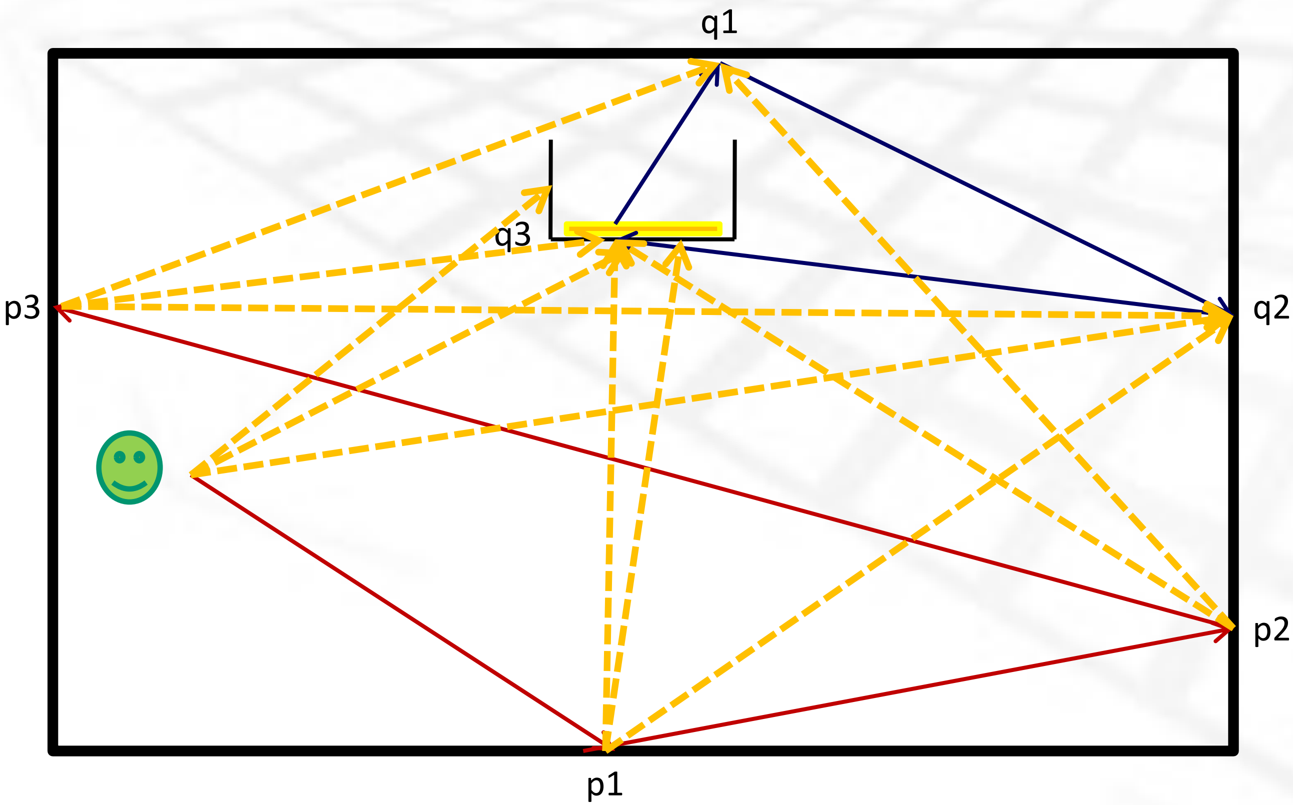
- Para algumas condições de iluminação a maioria dos trajectos gerados **NUNCA** encontram as fontes de luz: a sua contribuição será **ZERO**
- Os poucos trajectos que encontram a fonte de luz são os únicos que contribuem para o valor do pixel, resultando em grande variância (ruído)

Mestrado Integrado
Engª Informática

Path Tracing: limitações

Qual destas cenas exige menos "esforço computacional" ao *path tracer*?

BiDirectional Path Tracing



1. Gerar caminho a partir do observador
2. Gerar caminho a partir da fonte de luz
3. Unir os 2 caminhos
4. e pesar a contribuição de cada q_l para cada p_i