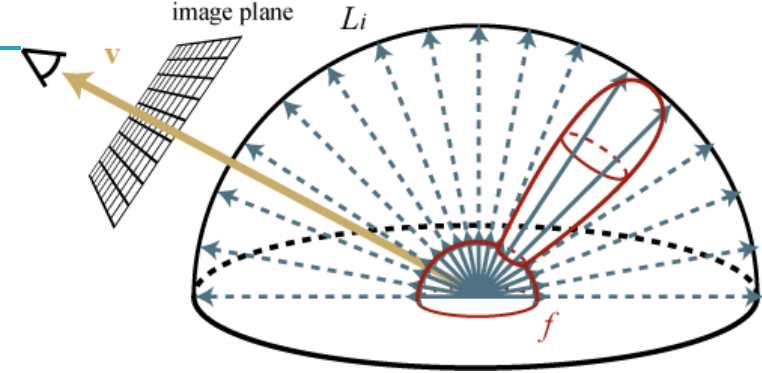

Počítačová grafika III – Zobrazovací rovnice a její řešení: Path tracing

Jaroslav Křivánek, MFF UK

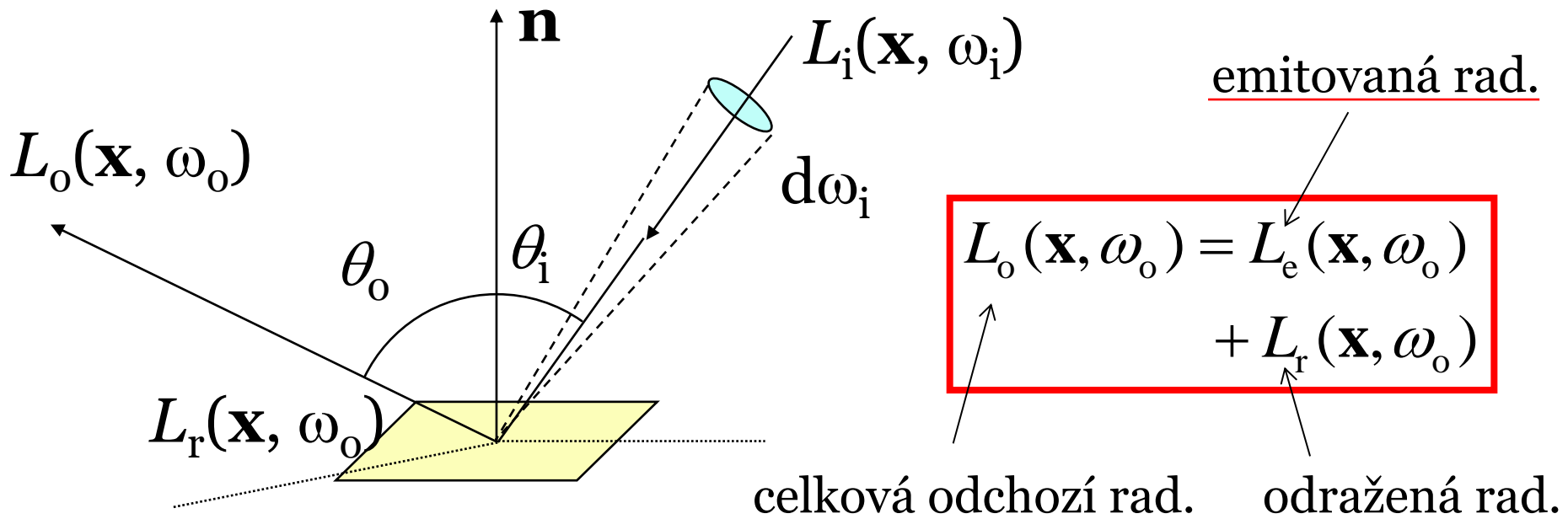
Jaroslav.Krivanek@mff.cuni.cz

Rovnice odrazu



- „Sečtení“ (integrál) příspěvků dL_r přes celou hemisféru:

$$L_r(\mathbf{x}, \omega_o) = \int_{H(\mathbf{x})} L_i(\mathbf{x}, \omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$



$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + L_r(\mathbf{x}, \omega_o)$$

Od lokálního odrazu ke globálnímu šíření světla

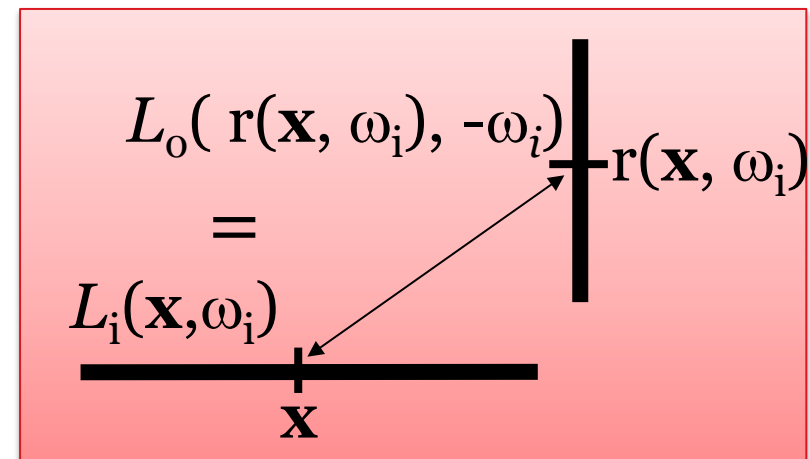
- Rovnice odrazu (lokální odraz)

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{H(\mathbf{x})} L_i(\mathbf{x}, \omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$

- Odkud přichází radiance $L_i(\mathbf{x}, \omega_i)$?
 - Z ostatních míst ve scéně !!!

$$L_i(\mathbf{x}, \omega_i) = L_o(\mathbf{r}(\mathbf{x}, \omega_i), -\omega_i)$$

Funkce vržení paprsku
(ray casting function)



Od lokálního odrazu ke globálnímu šíření světla

- Dosazení za L_i do rovnice odrazu

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{H(\mathbf{x})} L_o(r(\mathbf{x}, \omega_i), -\omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$

- Příchozí radiance L_i vyloučena.
- Odchozí radiance L_o popsána jako funkce L_o jinde ve scéně.

Zobrazovací rovnice – Rendering equation

- Odstranění indexu „o“ u odchozí radiance:

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{H(\mathbf{x})} L(\mathbf{r}(\mathbf{x}, \omega_i), -\omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$

- Popis ustáleného stavu = **energetické rovnováhy** ve scéně.
- **Rendering** = výpočet $L(\mathbf{x}, \omega_o)$ pro místa viditelná přes pixely.

Rovnice odrazu vs. zobrazovací rovnice

Podobný tvar – jiný význam

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{H(\mathbf{x})} L_i(\mathbf{x}, \omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$

■ Rovnice odrazu (reflection equation)

- popisuje **lokální odraz světla** v jednom místě
- Integrál, pomocí něhož lze spočítat odchozí radianci z příchozí radiance v daném bodě

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{H(\mathbf{x})} L(r(\mathbf{x}, \omega_i), -\omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$

■ Zobrazovací rovnice (rendering equation)

- Podmínka na **globální rozložení světla ve scéně**
- Integrovaná rovnice – neznámá L vlevo i vpravo

Rendering Equation – Kajiya 1986

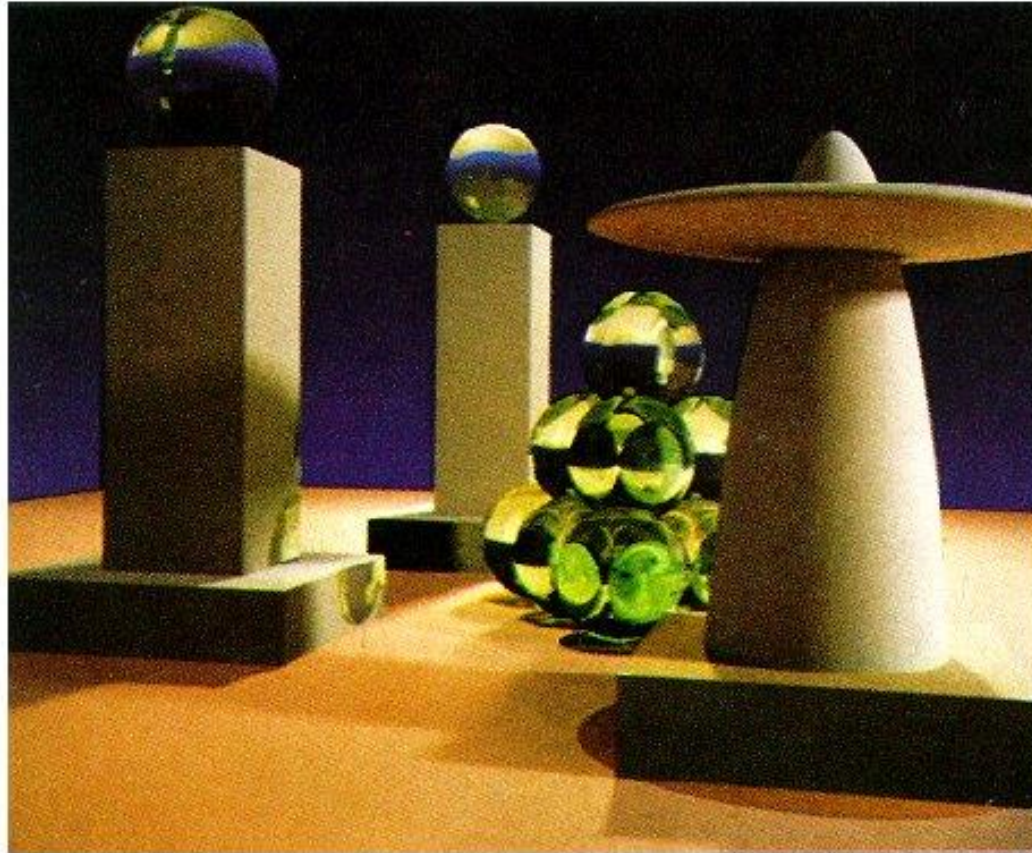


Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.

Kompletní formulace problému

- Dáno
 - M ... plocha, geometrie scény
 - $r(\mathbf{x}, \omega)$... funkce vržení paprsku
 - $V(\mathbf{x}, \mathbf{y})$... funkce viditelnosti
 - $L_e(\mathbf{x}, \omega)$... emitovaná radiance (zdroje světla)
 - $f_r(\mathbf{x}, \omega_i, \omega_o)$... BRDF (materiál povrchů)
- Cíl
 - Vypočítat hodnotu radiance v ustáleném stavu pro množinu bodů $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ na M s příslušnými směry $\omega_{o,1}, \omega_{o,2}, \dots, \omega_{o,n}$
- Abstraktní matematický problém
 - Nyní už žádná fyzika, jen matematika a algoritmy

Směrová a plošná forma ZR

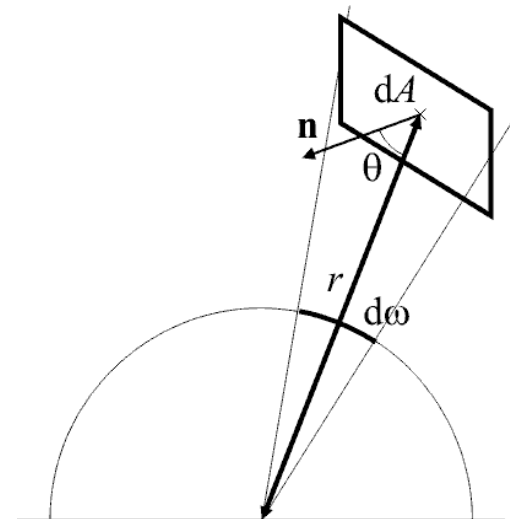
Zobrazovací rovnice – směry vs plochy

- **Úhlová forma:** integrál přes směry

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{H(\mathbf{x})} L(r(\mathbf{x}, \omega_i), -\omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$

- Substituce:

$$d\omega = dA \frac{\cos \theta}{r^2}$$



Zobrazovací rovnice – úhly vs plochy

- **Plošná forma:** integrál přes plochy scény

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o)$$

$$+ \int_M L(\mathbf{y} \rightarrow \mathbf{x}) \cdot f_r(\mathbf{y} \rightarrow \mathbf{x} \rightarrow \omega_o) \cdot G(\mathbf{x} \leftrightarrow \mathbf{y}) \cdot V(\mathbf{x} \leftrightarrow \mathbf{y}) dA_y$$

povrch scény

geometrický člen

$$G(\mathbf{x} \leftrightarrow \mathbf{y}) = \frac{\cos \theta_x \cdot \cos \theta_y}{\|\mathbf{x} - \mathbf{y}\|^2}$$

viditelnost

1 ... \mathbf{y} viditelné z \mathbf{x}
0 ... jinak

Integrál přes směry

- Sčítání příspěvků světla do bodu ze všech směrů
- Pro každý směr najdu nejbližší plochu
- Implementace ve stochastickém sledování paprsku:
 - Pro dané místo \mathbf{x} , generuj náhodné směry, pro každý najdi nejbližší průsečík, v něm spočítej odchozí radianci. To vše sečti přes všechny vygenerované náhodné směry.
- Typické použití: výpočet **nepřímého osvětlení** v bodě

Integrál přes plochy

- Sčítání příspěvků světla do bodu z ploch scény
- Příspěvek započítán pouze pokud je plocha viditelná
- Implementace ve stochastickém sledování paprsku:
 - Generuj náhodně místa \mathbf{y} na geometrii. Pro každé otestuj viditelnost mezi \mathbf{x} a \mathbf{y} . Pokud viditelné, přičti k osvětlení v \mathbf{x} odchozí radianci z \mathbf{y} váženou geometrickým faktorem.
- Typické použití: výpočet **přímého osvětlení** v bodě (plošné zdroje světla)

Způsoby řešení zobrazovací rovnice

- **Lokální osvětlení (OpenGL)**
 - výpočet integrálu odrazu pro bodové zdroje světla
 - bodové zdroje: integrál -> suma
 - Neposkytuje ustálenou radianci, není řešením ZR
- **Metoda konečných prvků (radiační metoda, radiozita), [Goral, '84]**
 - diskretizace plochy scény (konečné prvky)
 - zanedbává směrovost odrazu
 - nezobrazuje lesklé odrazy světla

Způsoby řešení ZR

- **Sledování paprsku (ray tracing)** [Whitted, '80]
 - ❑ pouze přímé osvětlení na lesklých a difúzní plochách a nepřímé osvětlení pouze na ideálně zrcadlových plochách (odraz, lom)
 - ❑ nepostihuje nepřímé osvětlení na difúzních a lesklých plochách, měkké stíny, ...
- **Distribuované sledování paprsku** [Cook, '84]
 - ❑ odhad lokálního integrálu metodou Monte Carlo
 - ❑ počítá měkké odrazy, stíny, hloubku ostrosti, ..

Způsoby řešení ZR

- **Sledování cest (Path tracing)** [Kajiya, '86]
 - řešení zobrazovací rovnice metodou Monte Carlo
 - výpočet náhodné cesty (“náhodné procházky”)
 - postihuje nepřímé osvětlení vyšších řádů

Od zobrazovací rovnice k radiální metodě

Od zobrazovací rovnice k radiozitě

- Formulace ZR pomocí integrálu přes plochy:

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_M L(\mathbf{y} \rightarrow \mathbf{x}) \cdot f_r(\mathbf{y} \rightarrow \mathbf{x} \rightarrow \omega_o) \cdot G(\mathbf{x} \leftrightarrow \mathbf{y}) \cdot V(\mathbf{x} \leftrightarrow \mathbf{y}) dA_y$$

- Radiozita – **předpoklady**
 - Pouze difúzní plochy (BRDF konstantní v ω_i a ω_o)
 - Elementy plochy mají konstantní radiozitu

Od zobrazovací rovnice k radiozitě

- Pouze difúzní plochy
 - BRDF konstantní v ω_i a ω_o

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \frac{\rho(\mathbf{x})}{\pi} \int_M L(\mathbf{y} \rightarrow \mathbf{x}) \cdot G(\mathbf{x} \leftrightarrow \mathbf{y}) \cdot V(\mathbf{x} \leftrightarrow \mathbf{y}) dA_y$$

- Odchozí radiance je nezávislá na ω a je rovna radiozitě B děleno π

$$B(\mathbf{x}) = B_e(\mathbf{x}) + \rho(\mathbf{x}) \cdot \int_M B(\mathbf{y}) \cdot \underbrace{\frac{G(\mathbf{x} \leftrightarrow \mathbf{y}) \cdot V(\mathbf{x} \leftrightarrow \mathbf{y})}{\pi}}_{G'(\mathbf{x} \leftrightarrow \mathbf{y})} dA_y$$

Od zobrazovací rovnice k radiozitě

- Konstantní radiozita B **přispívajících** plošných elementů

$$B(\mathbf{x}) = B_e(\mathbf{x}) + \rho(\mathbf{x}) \cdot \sum_{j=1}^N B_j \cdot \left(\int_{A_j} G'(\mathbf{x} \leftrightarrow \mathbf{y}) dA_{y,j} \right)$$

radiozita j -tého elementu

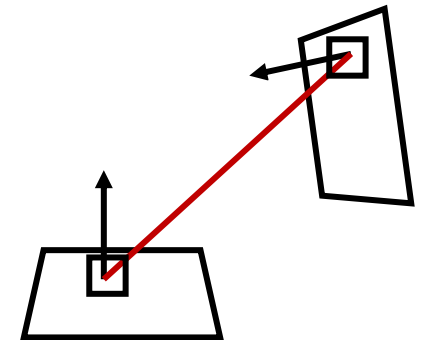
geometrický faktor mezi
ploškou j a bodem \mathbf{x}

Od zobrazovací rovnice k radiozitě

- Konstantní radiozita elementu i **přijímajícího** světlo:
 - Střední hodnota („průměr“) radiozity přes plochu elementu

$$B_i = \frac{1}{A_i} \int_{A_i} B(\mathbf{x}) dA_i =$$

$$= B_{e,i} + \rho_i \cdot \sum_{j=1}^N B_j \cdot \underbrace{\left(\frac{1}{A_i} \int_{A_i} \int_{A_j} G'(\mathbf{x} \leftrightarrow \mathbf{y}) dA_{y,j} dA_{x,i} \right)}_{F_{ij}}$$



F_{ij} ... konfigurační faktor
(„form factor“)

Klasická radiozitivní rovnice

- Soustava lineárních rovnic

$$B_i = B_{e,i} + \rho_i \cdot \sum_{j=1}^N B_j \cdot F_{ij}$$

- Konfigurační faktory

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} G'(\mathbf{x} \leftrightarrow \mathbf{y}) dA_{\mathbf{y},j} dA_{\mathbf{x},i}$$

Radiační metoda

■ Klasická radiozita

- Výpočet konfiguračních faktorů (Monte Carlo, hemicube, ...)
- Řešení radiozitivní rovnice (Gathering, Shooting, ...)

■ Stochastická radiozita

- Obchází explicitní výpočet konfiguračních faktorů
- Metoda Monte Carlo

■ Nepraktická, nepoužívá se v praxi

- Rozdělení na plošky -> citlivost na kvalitu modelu
- Vysoké paměťové nároky, Náročná implementace

Vyjádření ZR pomocí integrálního operátoru

ZR je integrální rovnice

Obecný tvar Fredholmovy integrální rovnice druhého druhu

$$f(x) = g(x) + \int k(x, x') f(x') dx'$$

neznámá fce

známé fce

„jádro“ rce

Zobrazovací rovnice:

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{H(\mathbf{x})} L(r(\mathbf{x}, \omega_i), -\omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta d\omega_i$$

Lineární operátory

- Lineární operátory **působí** na funkce
 - jako matice na vektory

$$h(x) = (L \circ f)(x)$$

- Působení je lineární

$$L \circ (af + bg) = a(L \circ f) + b(L \circ g)$$

- Příklady lineárních operátorů

$$(K \circ f)(x) \equiv \int k(x, x') f(x') dx'$$

$$(D \circ f)(x) \equiv \frac{\partial f}{\partial x}(x)$$

Transportní operátor

$$(T \circ L)(\mathbf{x}, \omega_o) \equiv \int_{H(\mathbf{x})} L(\mathbf{x}, \omega_i) \cdot f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i$$

- Zobrazovací rovnice

$$L = L_e + T \circ L$$

Řešení ZR v operátorovém tvaru

- Zobrazovací rovnice

$$L = L_e + T \circ L$$

- Formální řešení

$$(I - T) \circ L = L_e$$

$$L = (I - T)^{-1} \circ L_e$$

- v praxi nepoužitelné – inverzi nelze explicitně vyjádřit

Expanze zobrazovací rovnice

- Rekurzivní substituce L

$$\begin{aligned}L &= L_e + TL \\ &= L_e + T(L_e + TL) \\ &= L_e + TL_e + T^2L\end{aligned}$$

- n -násobným opakováním vznikne Neumannova řada

$$L = \sum_{i=0}^n T^i L_e + T^{n+1}L$$

Expanze zobrazovací rovnice

- Pokud je T kontrakce (tj. $\|T\| < 1$, v ZR platí), pak

$$\lim_{n \rightarrow \infty} T^{n+1} L = 0$$

- **Řešení zobrazovací rovnice** je pak dáno

$$L = \sum_{i=0}^{\infty} T^i L_e$$

Jiné odvození Neumannovy řady

- Formální řešení zobrazovací rovnice

$$L = (I - T)^{-1} \circ L_e$$

- Platí

$$(I - T)^{-1} = I + T + T^2 + \dots$$

- Důkaz

$$\begin{aligned}(I - T) \circ (I - T)^{-1} &= (I - T) \circ (I + T + T^2 + \dots) \\ &= (I + T + T^2 + \dots) - (T + T^2 + T^3 + \dots) \\ &= I\end{aligned}$$

Postupné aproximace

- Každá aplikace T odpovídá jednomu odrazu & přenosu světla

$$L = L_e + TL_e + T^2 L_e + T^3 L_e + \dots$$

emise z povrchu zdrojů

přímé osvětlení

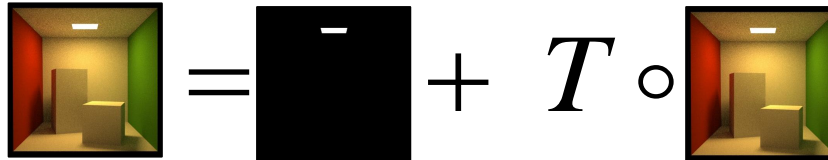
nepřímé osvětlení prvního řádu
(one-bounce indirect)

nepřímé osvětlení druhého řádu
(two-bounce indirect)

OpenGL stínování

Zobrazovací rovnice – Kajiya 1986

$$L = L_e + T \circ L$$

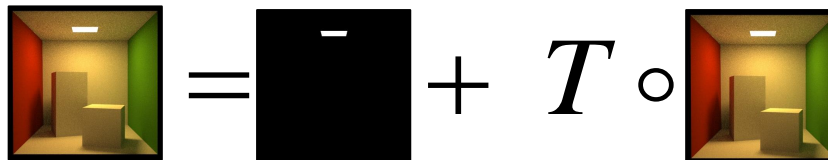


- L : Ustálená zář
- L_e : Emitovaná zář
- T : Operátor přenosu & odrazu světla

$$(T \circ L)(\mathbf{x}, \omega_o) \equiv \int_M L(\mathbf{y} \rightarrow \mathbf{x}) \cdot f_r(\mathbf{y} \rightarrow \mathbf{x} \rightarrow \omega_o) \cdot G(\mathbf{y} \leftrightarrow \mathbf{x}) dA_y$$

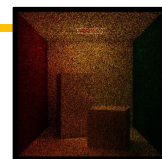
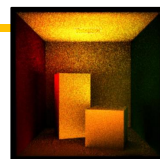
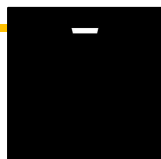
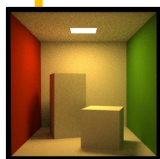
Zobrazovací rovnice – Kajiya 1986

$$L = L_e + T \circ L$$

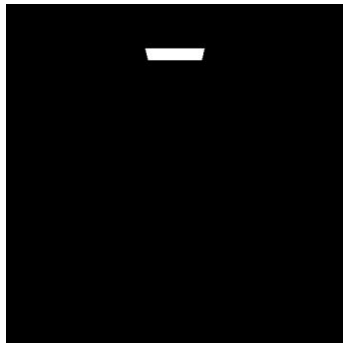


- **Řešení:** Neumannova řada

$$L = L_e + TL_e + T^2 L_e + T^3 L_e + \dots$$



Postupné aproximace



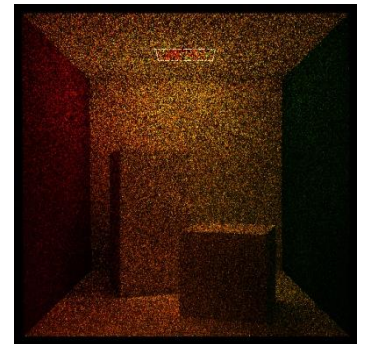
L_e



$T \circ L_e$



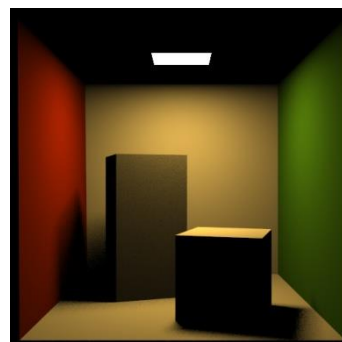
$T \circ T \circ L_e$



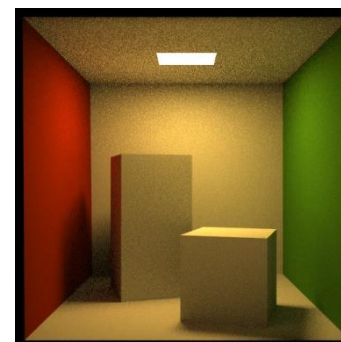
$T \circ T \circ T \circ L_e$



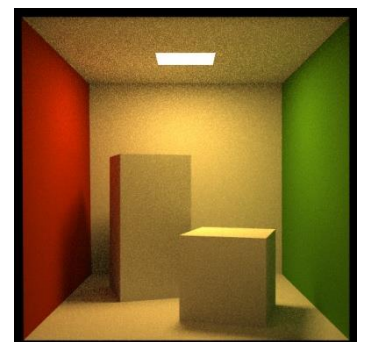
L_e



$L_e + T \circ L_e$



$L_e + T L_e + T^2 L_e$



$L_e + \dots + T^3 L_e$

Kontraktivita T

- Platí pro fyzikálně korektní modely
 - Vyplývá ze zachování energie
- Znamená, že opakované aplikace operátoru snižují energii (odrazivosti všech ploch jsou < 1)
- Scény s velmi lesklými povrchy
 - odrazivost blízká 1
 - konvergence vyžaduje simulovat větší množství odrazů světla než v difúzních scénách

Čeho jsme tím dosáhli?

- Nahradili jsme integrální rovnici řadou integrálů s postupně rostoucí dimenzionalitou
- Numericky vyhodnocovat integrály umíme (metoda Monte Carlo) -> umíme řešit zobrazovací rovnici -> umíme renderovat obrázky, hurá!
- Rekurzivní aplikace T odpovídá rekurzivnímu sledování paprsku od kamery

Co to tedy vlastně počítáme?

$$L(\mathbf{x}, \omega_0) = L_e(\mathbf{x}, \omega_0) +$$

$$\int_M L_e(\mathbf{y} \rightarrow \mathbf{x}) \cdot f_r(\mathbf{y} \rightarrow \mathbf{x} \rightarrow \omega_0) \cdot G(\mathbf{x} \leftrightarrow \mathbf{y}) \cdot V(\mathbf{x} \leftrightarrow \mathbf{y}) \, dA_y +$$

$$\iint_M L_e(\mathbf{z} \rightarrow \mathbf{y}) \cdot [f_r(\mathbf{z} \rightarrow \mathbf{y} \rightarrow \mathbf{x}) \cdot G(\mathbf{y} \leftrightarrow \mathbf{z}) \cdot V(\mathbf{y} \leftrightarrow \mathbf{z})] \cdot$$
$$\cdot [f_r(\mathbf{y} \rightarrow \mathbf{x} \rightarrow \omega_0) \cdot G(\mathbf{x} \leftrightarrow \mathbf{y}) \cdot V(\mathbf{x} \leftrightarrow \mathbf{y})] \, dA_y \, dA_z +$$

$$\iiint_M L_e(\check{\mathbf{z}} \rightarrow \mathbf{z}) \dots \, dA_y \, dA_z \, dA_{\check{z}}$$

Cesty vs. rekurze: Otázka interpretace

- Nezávislé cesty ve vysokodimenzionálním prostoru

$$L = L_e + TL_e + T^2 L_e + T^3 L_e + \dots$$

- Rekurzivní řešení dvojných integrálů přes (hemi)sféru:

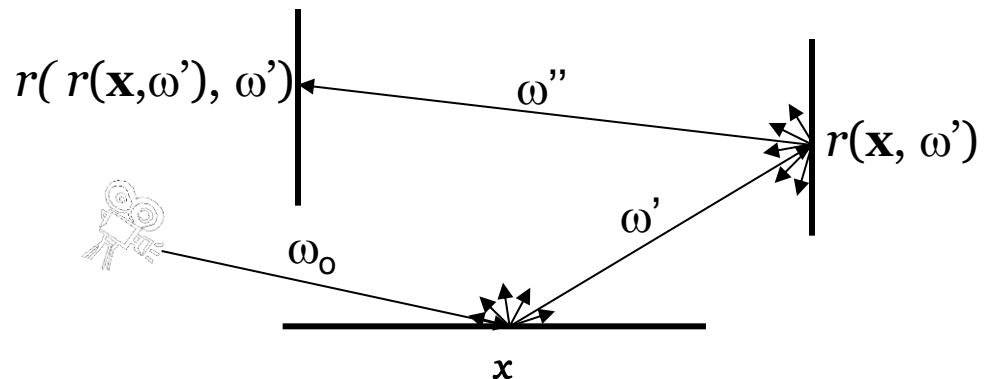
$$L = L_e + T(L_e + T(L_e + T(L_e + \dots$$

Rekurzivní interpretace

- Úhlová formulace ZR

$$L(\mathbf{x}, \omega_0) = L_e(\mathbf{x}, \omega_0) + \int_{H(\mathbf{x})} L(r(\mathbf{x}, \omega'), -\omega') \cdot f_r(\mathbf{x}, \omega' \rightarrow \omega_0) \cdot \cos \theta' d\omega'$$

- Pro výpočet $L(\mathbf{x}, \omega_0)$ potřebuji spočítat $L(r(\mathbf{x}, \omega'), -\omega')$ pro všechny směry ω' okolo bodu \mathbf{x} .
- Pro výpočet každého $L(r(\mathbf{x}, \omega'), -\omega')$ potřebuji spočítat $L(r(r(\mathbf{x}, \omega'), -\omega''), -\omega')$ pro všechny směry ω'' okolo bodu $r(\mathbf{x}, \omega')$
- Atd... => rekurze



Path tracing, v. 0

getLi (x, ω):

$\mathbf{y} = \text{traceRay}(\mathbf{x}, \omega)$

return

$\text{Le}(\mathbf{y}, -\omega) +$ // emitted radiance

$\text{Lr}(\mathbf{y}, -\omega)$ // reflected radiance

Lr(x, ω):

$\omega' = \text{genUniformRandomDir}(\mathbf{n}(\mathbf{x}))$

return $\pi * \text{brdf}(\mathbf{x}, \omega, \omega') * \text{rayRadianceEst}(\mathbf{x}, \omega')$

Path tracing, v. 2012



© 2012 Columbia Pictures Industries, Inc. All Rights Reserved.

Závěr

- Potřebujeme nástroj pro numerické integrování
 - Přes hemisféru
 - Přes plochy
 - A přes Kartézské součiny předchozího
 - **Mnohodimenzionální integrály**
- Nástroj
 - **Monte Carlo metody**

Sledování cest (Path tracing, Kajiya86)

- Pouze jeden sekundární paprsek
 1. Náhodný výběr interakce (ideální lom, difúzní odraz, ...)
 2. Importance sampling podle vybrané interakce
- Přímé osvětlení
 - Doufej, že náhodně vygenerovaný paprsek trefí zdroj, anebo
 - Vyber náhodně jeden vzorek na jednom zdroji světla
- Trasuj stovky cest přes každý pixel a zprůměruj výsledek
- Výhoda: žádná exploze počtu paprsků kvůli rekurzi

Sledování cest od kamery

```
renderImage()  
{  
  for all pixels  
  {  
    Color pixelCol = (0,0,0);  
    for k = 1 to N  
    {  
      wk := náhodný směr skrz k-tý pixel  
      pixelCol += getLi(camPos, wk)  
    }  
    return Lo / N  
  }  
}
```

Path Tracing – Implicitní osvětlení

```
getLi(x, w)
{
    Color thrput = (1,1,1)
    Color accum = (0,0,0)
    while(1)
    {
        hit = NearestIntersect(x, w)
        if no intersection
            return accum + thrput * bgRadiance(x, w)
        if isOnLightSource(hit)
            accum += thrput * Le(hit.pos, -w)
        ρ = reflectance(hit.pos, -w)
        if rand() < ρ // russian roulette - survive (reflect)
            wi := SampleDir(hit)
            thrput *= fr(hit.pos, wi, -w) * dot(hit.n, wi) / (ρ*pdf(wi))
            x := hit.pos
            w := wi
        else // absorb
            break;
    }
    return accum;
}
```

Ukončení rekurze – Ruská ruleta

- Pokračuj v rekurzi s pravděpodobností q
- Uprav váhu faktorem $1 / q$

$$Z = \begin{cases} Y / q & \text{pokud } \xi < q \\ 0 & \text{jinak} \end{cases}$$

$$E[Z] = \frac{E[Y]}{q} \cdot q + 0 \cdot \frac{1}{q-1} = E[Y]$$

Výběr náhodného směru – Importance Sampling

```
getLi(x, w)
{
    Color thrput = (1,1,1)
    Color accum = (0,0,0)
    while(1)
    {
        hit = NearestIntersect(x, w)
        if no intersection
            return accum + thrput * bgRadiance(x, w)
        if isOnLightSource(hit)
            accum += thrput * Le(hit.pos, -w)
        ρ = reflectance(hit.pos, -w)
        if rand() < ρ // russian roulette - survive (reflect)
            wi := SampleDir(hit)
            thrput *= fr(hit.pos, wi, -w) * dot(hit.n, wi) / (ρ * pdf(wi))
            x := hit.pos
            w := wi
        else // absorb
            break;
    }
    return accum;
}
```


Výběr náhodného směru – Importance Sampling

- Obyčejně vzorkujeme s hustotou „co nejpodobnější“ součinu

$$f_r(\omega_i, \omega_o) \cos \theta_i$$

- Ideálně bychom chtěli vzorkovat podle

$$L_i(\omega_i) f_r(\omega_i, \omega_o) \cos \theta_i,$$

ale to neumíme, protože neznáme L_i

- Co když bude hustota přesně úměrná $f_r(\omega_i, \omega_o) \cos \theta_i$?

„Ideální“ BRDF Importance Sampling

$$p(\omega_i) \propto f_r(\omega_i \rightarrow \omega_o) \cdot \cos \theta_i$$

- Normalizace (integrál pdf musí být = 1)

$$p(\omega_i) = \frac{f_r(\omega_i \rightarrow \omega_o) \cdot \cos \theta_i}{\int_{H(\mathbf{x})} f_r(\omega_i \rightarrow \omega_o) \cdot \cos \theta_i \, d\omega_i}$$

odrazivost ρ

„Ideální“ BRDF IS v Path Traceru

- Obecná hustota (pdf)

```
...  
thruput *= fr(.) * dot(.) / ( rho * p(wi) )
```

- „Ideální“ BRDF importance sampling

$$p(\omega_i) = f_r(\omega_i \rightarrow \omega_o) \cdot \cos \theta_i / \rho$$

```
...  
thruput *= 1
```

Pravděpodobnost přežití cesty

```
getLi(x, w)
{
    Color thrput = (1,1,1)
    Color accum = (0,0,0)
    while(1)
    {
        hit = NearestIntersect(x, w)
        if no intersection
            return accum + thrput * bgRadiance(x, w)
        if isOnLightSource(hit)
            accum += thrput * Le(hit.pos, -w)
         $\rho = \text{reflectance}(\text{hit.pos}, -w)$ 
        if  $\text{rand}() < \rho$  // russian roulette - survive (reflect)
            wi := SampleDir(hit)
            thrput *= fr(hit.pos, wi, -w) * dot(hit.n, wi) / ( $\rho$  * p(wi))
            x := hit.pos
            w := wi
        else // absorb
            break;
    }
    return accum;
}
```

Pravděpodobnost přežití cesty

- Použití odrazivosti ρ jako p-nosti přežití dává smysl
 - Pokud plocha odráží jen 30% energie, pokračujeme pouze s 30% pravděpodobností.

- Co když neumím spočítat ρ ?

- Alternativa

1. Nejdříve vygeneruj náhodný směr podle $p(\omega_i)$

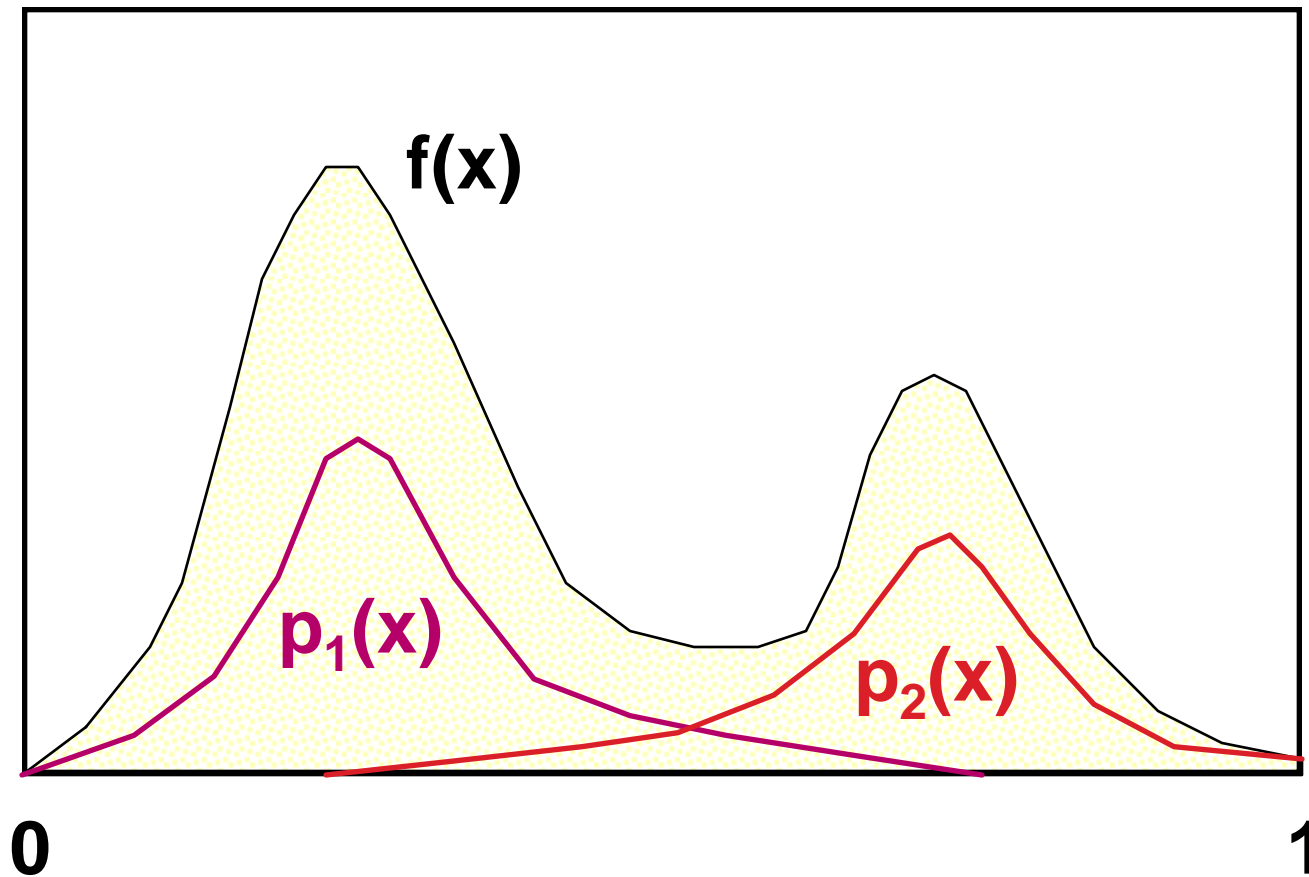
2.
$$q_{\text{survival}} = \min \left\{ 1, \frac{f_r(\omega_i \rightarrow \omega_o) \cos \theta_i}{p(\omega_i)} \right\}$$

- Pro „ideální“ BRDF IS stejné jako původní metoda

Výpočet přímého osvětlení pomocí MIS v path traceru

Multiple Importance Sampling

(Veach & Guibas, 95)



Vyrovnaná heuristika (Balance heurist.)

- Výsledný estimátor (po dosazení vah)

$$F = \sum_{i=1}^n \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{\sum_k n_k p_k(X_{i,j})}$$

- příspěvek vzorku nezávisí na tom, ze které byl pořízen techniky (tj. pdf)

Použití MIS v path traceru

- Pro každý vrchol cesty generované z kamery:
 - Generování explicitního stínového paprsku pro techniku p_2 (vzorkování plochy zdroje)
 - Sekundární paprsek pro techniku p_1 (vzorkování zdroje)
 - Sdílený pro výpočet **přímého** i **nepřímého** osvětlení
 - Pouze na přímé osvětlení se aplikuje MIS váha (nepřímé osvětlení se připočte celé)
 - Při výpočtu MIS vah je potřeba vzít v úvahu pravděpodobnost ukončení cesty (ruská ruleta)

Více zdrojů světla

- Možnost 1:
 - Stínový paprsek pro náhodný bod na každém zdroji světla
- Možnost 2 (často lepší):
 - Náhodný výběr zdroje (s p-ností podle výkonu)
 - Stínový paprsek k náhodně vybranému bodu na vybraném zdroji
- Pozor: Pravděpodobnost výběru zdroje ovlivňuje hustoty (a tedy i váhy) v MIS