

# Monte Carlo rendering

Jan Havlíček

27. listopadu 2011

## 1 Základní algoritmus path traceru

```
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 = reflectance(hit.pos, -w)
    if 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;
}
```

Algoritmus path traceru počítá radianci v bodě  $x$  a ve směru  $\omega$ . Na počátku je  $x$  pozice oka a  $\omega$  je směr, kterým se kouká do scény pro daný pixel. *thrput* je "váha" sledovaného paprsku. Paprsek zkusí najít průsečík se scénou a pokud nic nenajde, vrátí spočítanou radianci. Pokud paprsek dopadl na světelný zdroj, připočítá se k výsledku radianci světelného zdroje opravenou o váhu sledovaného paprsku. Poté je potřeba se rozhodnout, zda by měl dále pokračovat nebo skončit. Spočte si pravděpodobnost, s jakou by měl pokračovat, vhodná hodnota je odrazivost  $\rho$  (pokud plocha odráží 90%, tak s 90% pravděpodobností bude pokračovat, pokud 20%, tak paprsek bude pokračovat s 20% pravděpodobností). Path tracer tedy spočítá pravděpodobnost, s jakou by měl pokračovat ve výpočtu (nemusí to být nutně odrazivost, ale ta má velmi dobré vlastnosti), vygeneruje náhodné číslo a rozhodne se. Pokud končí, tak vrátí spočítanou hodnotu, pokud pokračuje, tak vygeneruje nový směr  $\omega_i$  podle nějakého rozložení pravděpodobnosti  $p(\omega_i)$ , upraví váhu nového paprsku a nastaví  $x$  na průsečík původního paprsku se scénou a  $\omega$  bude směr nového paprsku.

Sledování jedné cesty je potřeba někdy ukončit, je několik možností:

- podle počtu odrazů cesty - nevhodné, protože občas sleduje nedůležité paprsky, zatímco důležité mohou být ukončeny předčasně
- pokud je *thruput* menší, než konstanta
- ruskou ruletou - jediné nestranné řešení, v každém kroku s nějakou pravděpodobností cestu ukončíme

Při generování náhodných sekvencí je důležité, aby každý pixel měl nezávislé odhady. Pokud jsou paprsky korelované, tak je vizuální dojem podstatně horší. Numerická chyba je stejná, ale subjektivní pocit je horší.

Path tracer je jeden z prvních algoritmů, je velice jednoduchý, ale právě díky své jednoduchosti se v praxi prosazuje (např. filmy 2012, Alice in the Wonderland). Není příliš rychlý, ale je velmi robustní, tzn. funguje pro velkou množinu scén, nemá skoro žádné parametry (=není potřeba zaměstnanec se znalostmi) a dokud běží, tak zlepšuje výsledek.

## 2 Detaily path traceru

Směr sekundárního paprsku generujeme z nějaké hustoty pravděpodobnosti  $p(\omega_i)$  na hemisféře, ideálně bychom chtěli generovat podle  $L_i(\omega_i)f_r(\omega_i, \omega_o) \cos(\theta_i)$ , to ale nejde, protože nevíme, jaká je hodnota příchozího světla  $L_i(\omega_i)$  v bodě  $x$ . Obvykle se tedy snažíme generovat paprsky podle  $f_r(\omega_i, \omega_o) \cos(\theta_i)$  (buď přímo podle této funkce nebo podle nějaké podobné, protože podle  $f_r(\omega_i, \omega_o)$  je často obtížné vzorkovat).

Co by se stalo, kdybychom v path traceru používali přímo  $f_r(\omega_i, \omega_o) \cos(\theta_i)$ ? Neboli co kdyby  $p(\omega_i)$  bylo přímo úměrné  $f_r(\omega_i, \omega_o) \cos(\theta_i)$ ? Nejdříve je potřeba spočítat pdf (*probability distribution function*), pdf má integrál 1, použijí normalizaci a dostanu

$$p(\omega_i) = \frac{f_r(\omega_i, \omega_o) \cos(\theta_i)}{\int_{Hx} f_r(\omega_i, \omega_o) \cos(\theta_i) d\omega_i} = \frac{f_r(\omega_i, \omega_o) \cos(\theta_i)}{\rho} \quad (1)$$

Člen ve jmenovateli je odrazivost  $\rho$ . Pokud tuto pdf použijí v path traceru místo obecné  $p(\omega_i)$ , tak se členy ve zlomku upravující *thruput* vykrátí a *thruput* se v jednotlivých krocích nebude měnit.

$$thruput* = \frac{f_r(\omega_i, \omega_o) \text{dot}(x.\text{normal}, -\omega)}{\rho * p(\omega_i)}$$

$$thruput* = 1$$

Důsledkem neměnicího se *thruput* je, že dokud cesta neskončí, tak sekundární paprsky cesty budou vždy přispívat stejnou vahou. Při použití jiného rozdělení by se *thruput* neustále zmenšoval a příspěvky sekundárních paprsků by k finálnímu výsledku přispívaly stále menším dílem. To je důvod, proč chceme, aby pdf byla podobná  $f_r(\omega_i, \omega_o) \cos(\theta_i)$ . Pokud není pdf není zcela stejná (ale je velmi podobná), *thruput* bude zůstávat přibližně na hodnotě 1.

Jak je možné, že jeden paprsek stále přispívá zhruba stejně ke konečnému výsledku, třeba i po 20 odrazech? Lze si to představit jako situaci, kdy místo deseti 1 wattových paprsků vyšlu jeden 10 wattový. Pokud se použije správné rozdělení pro vyslání jednoho paprsku, tak to statisticky vyjde stejně.

### 3 Pravděpodobnost přežití cesty

Jak je uvedeno výše, je dobré použít jako pravděpodobnost přežití cesty odrazivost plochy v bodě  $x$ . Pokud by byla použita jiná metoda (např. pro plochu s  $\rho$  30% by cesta pokračovala s pravděpodobností 100%), tak by sekundární paprsky musely být počítány častěji, než je nutné (což stojí výpočetní výkon), a musely by být vynásobeny odrazivostí.

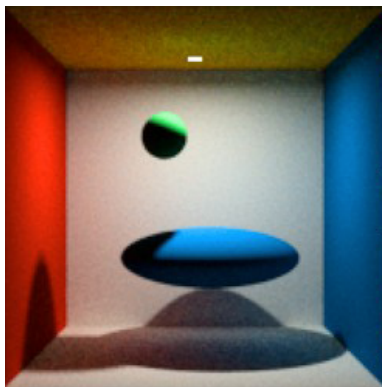
Ne vždy umíme spočítat odrazivost  $\rho$ , v takovém případě lze použít jiný způsob rozhodnutí o "přežití" cesty, který dává podobné výsledky, ale nevyžaduje spočítat odrazivost  $\rho$ : vygeneruji směr sekundárního paprsku z  $p(\omega_i)$  (dříve, než se rozhodnu, zda ho vůbec budu potřebovat) a ze směru spočítám pravděpodobnost, zda by měl paprsek pokračovat:

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

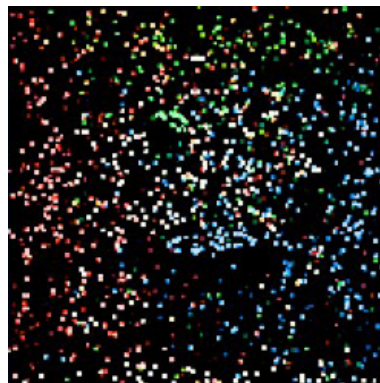
Pokud je  $p(\omega_i)$  splňuje rovnici 1, tak je výsledek  $q_{survival}$  vždy  $\rho$ , což je ekvivalentní s původní metodou. Pro obecné  $p(\omega_i)$  může být  $q_{survival}$  větší než 1 a proto je shora omezené hodnotou 1. Výhodou této metody je, že nevyžaduje znalost odrazivosti  $\rho$ , ale funguje zhruba stejně efektivně, a pokud je  $p(\omega_i)$  "ideální," (tj. pokud splňuje rovnici 1) tak funguje zcela stejně. Pokud je  $p(\omega_i)$  daleko od ideálu (tj. pokud její průběh není podobný  $f_r(\omega_i, \omega_o) \cos(\theta_i)$ ), tak funguje lépe, než metoda s odrazivostí.

### 4 Výpočet přímého osvětlení v path traceru

V základním algoritmu path traceru postupně střílíme paprsky a pokud náhodou trefíme zdroj světla, tak má paprsek nenulový příspěvek a všechny ostatní cesty, které skončily dříve, než se stačily trefit do světla mají nulový příspěvek. Tento způsob není příliš rychlý a trvá dlouho, než zkonverguje, což je vidět dobře na obrázcích 1a a 1b. Čím je zdroj světla menší, tím je menší pravděpodobnost, že paprsek se do zdroje trefí, pokud je zdroj bodový, tak je pravděpodobnost limitně nula.



(a) Referenční obrázek



(b) Základní path tracer, 150 cest na pixel

Obrázek 1: Cornell box (Autor: Alexander Wilkie)

Pokud se omezíme na problém přímého osvětlení, existují dva přístupy, jak zjistit, kolik světla dopadá na místo  $x$ :

- vzorkování BRDF - posíláme paprsek po hemisféře a doufáme, že se trefíme do zdroje světla.
- vzorkování plochy světla - vzorkujeme plochu zdroje světla (vzorkované místo označuji  $y$ ) při zohlednění viditelnosti mezi body  $x$  a  $y$ .

## 5 Vzorkování BRDF

Pro výpočet odražené radiance  $L_r$  vzorkováním BRDF využijeme formulaci rovnice odrazu jako integrálu přes směry na polokouli:

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

Pro výpočet estimátoru rovnice odrazu generujeme náhodné směry  $\omega_{i,k}$  podle hustoty  $p$ , která je co nejpodobnější členu  $f_r(x, \omega_i, \omega_o) \cos \theta_i$ , vrhneme paprsek z  $x$  ve směru  $\omega_{i,k}$  a pokud paprsek protne nějaký zdroj světla, přičteme  $L_e(r(x, \omega_{i,k}), -\omega_{i,k}) \cdot f_r(x, \omega_{i,k}, \omega_o) \cdot \cos \theta_{i,k} / pdf$ . Výsledný estimátor má tedy tvar:

$$\hat{L}_r(x, \omega_o) = \frac{1}{N} \sum_{k=1}^N \frac{L_e(r(x, \omega_{i,k}), -\omega_{i,k}) \cdot f_r(x, \omega_{i,k}, \omega_o) \cdot \cos \theta_{i,k}}{p(\omega_{i,k})}$$

## 6 Vzorkování světla

Vzorec pro vzorkování světla získám z rovnice odrazu, kterou upravíme tak, aby místo integrování přes hemisféru integrovala přes plochy světla:

$$L_r(x, \omega_o) = \int_A L_e(y \rightarrow x) \cdot f_r(y \rightarrow x \rightarrow \omega_o) \cdot V(y \leftrightarrow x) \cdot G(y \leftrightarrow x) dA_y$$

$V$  je funkce viditelnosti, pokud je mezi  $x$  a  $y$  překážka, je  $V(y \leftrightarrow x) = 0$ , jinak  $V(y \leftrightarrow x) = 1$ . Pro výpočet estimátoru generujeme náhodnou pozici  $y_k$  na zdroji, otestujeme viditelnost mezi  $x$  a  $y$ , a pokud je  $V(y \leftrightarrow x) = 1$ , přičteme  $|A| L_e(y) f_r(\cdot) \cos / pdf$ .

$$\hat{L}_r(x, \omega_o) = \frac{|A|}{N} \sum_{k=1}^N L_e(y_k \rightarrow x) \cdot f_r(y_k \rightarrow x \rightarrow \omega_o) \cdot V(y_k \leftrightarrow x) \cdot G(y_k \leftrightarrow x)$$

Přítomnost geometrického faktoru

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

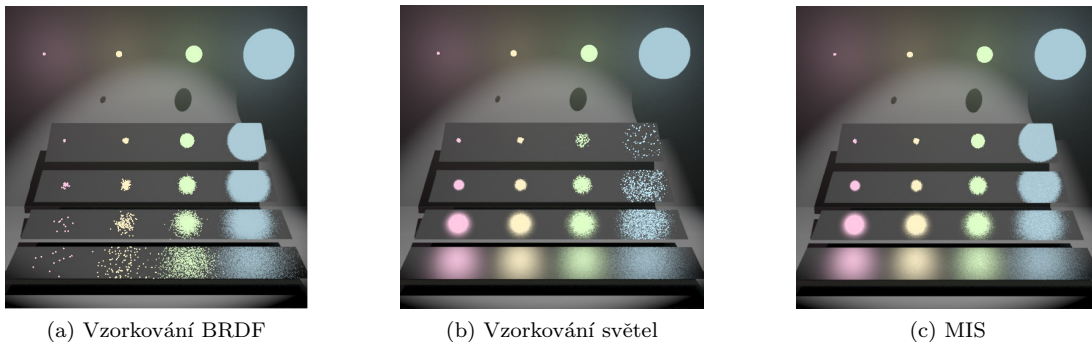
v této formulaci způsobuje úbytek světla se čtvercem vzdálenosti. Povšimněte si, že tento úbytek je dán 3D geometrií prostoru (substituce  $d\omega = \cos \theta / r^2 * dA$ ), nikoli vlastnostmi světla.

## 7 Vlastnosti vzorkování BRDF a vzorkování světla

Máme dva různé a korektní způsoby, jak vyřešit (spočítat) jeden problém. Kdy je vhodné použít kterou metodu?

- vzorkování BRDF - výhodnější pro velká světla, pro malé zdroje je malá pravděpodobnost zásahu zdroje → vysoký rozptyl, šum
- vzorkování světla - výhodnější pro malé zdroje, jediná možnost pro bodové zdroje. Pro velké zdroje je mnoho vzorků mimo lalok BRDF → vysoký rozptyl, šum

Na obrázcích 2a a 2b je identická scéna zobrazená oběma metodami. Ve scéně jsou 4 světla s různými velikostmi a 4 plochy s různou lesklostí (nejvzdálenější jsou nejvíce odrazivé, blízké nejvíce difúzní). Na obrázku 2a vlevo dole je malý zdroj a difúzní plocha. V takové situaci metoda se vzorkování BRDF skoro nikdy netrefí (metoda náhodně volí směry na difúzní ploše a proto zhruba rovnoměrně po celé hemisféře) a výsledkem je náhodný shluk pixelů. Metoda vzorkování světla naopak selhává na obrázku 2b vpravo nahoře (velký zdroj světla s lesklým povrchem). Příčinou tohoto chování je, že když náhodně vygeneruji bod na světle, tak se světelný příspěvek toho bodu musí vynásobit BRDF a na lesklém povrchu se paprsek málokdy treť do laloku BRDF.



Obrázek 2: Ukázka vzorkovacích technik (Autor: Eric Veach)

## 8 Multiple importance sampling

Vzhledem k tomu, že obě techniky selhávají na vzájemně komplementárních případech, by bylo dobré, kdyby bylo možné je zkombinovat do jedné, kde by slabost jedné byla vyvážena silou druhé. Výsledky obou metod nemůžeme sčítat (výsledek by byl  $2L_r(x, \omega_o)$ ), potřebujeme tedy nějaký vážený průměr obou metod. Otázka zní, jak korektně zvolit váhy. Metoda MIS používá váhy závislé na pdf vzorků a dodává velmi dobré výsledky (viz. obrázek 2c).

Metoda *Multiple Importance Sampling* má váhy závislé na pdf vzorků, minimalizuje rozptyl kombinovaného estimátoru a má téměř optimální řešení. Mějme  $n$  vzorkovacích technik (=hus-tot pravděpodobností, protože každé vzorkování má jednoznačně určenou pdf, s jakou vzorkuje),  $p_1(x), p_2(x) \dots p_n(x)$ , z každé pdf vybereme  $n_i$  vzorků  $X_{i,1}, X_{i,1} \dots X_{i,n_i}$  a spočítáme kombinovaný estimátor MIS podle následujícího vzorce:

$$F = \sum_{i=1}^n \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})} \quad (2)$$

První suma iteruje přes  $n$  vzorkovacích technik, druhá přes všechny vzorky z jednotlivých technik,  $w_i$  jsou kombinační váhy jednotlivých vzorků a  $f/p$  je totéž, jako v klasickém Monte Carlo

estimátoru. Kombinovaný estimátor  $F$  je nestranný, pokud  $\forall x : \sum_{i=1}^n w_i(x) = 1$  (tj. pokud pro všechna  $x$  je součet jejich vah roven jedné):

$$E[F] = \int \left[ \sum_{i=1}^n w_i(x) \right] f(x) dx \equiv \int f(x)$$

## 8.1 Vyrovnaná heuristika

Libovolně definovaná sada vah, která se sečte do 1 nám tedy dává nestranný estimátor MIS. To ale nestačí, chceme také minimalizovat rozptyl. Jako první řešení se nabízí aritmetický průměr:  $w_i = 1/n$ , která v sobě spojuje to nejhorší z obou technik a není vůbec vhodný. Mnohem lepší volba je *vyrovnaná heuristika*, kde jsou váhy jednotlivých technik dány následujícím vzorcem:

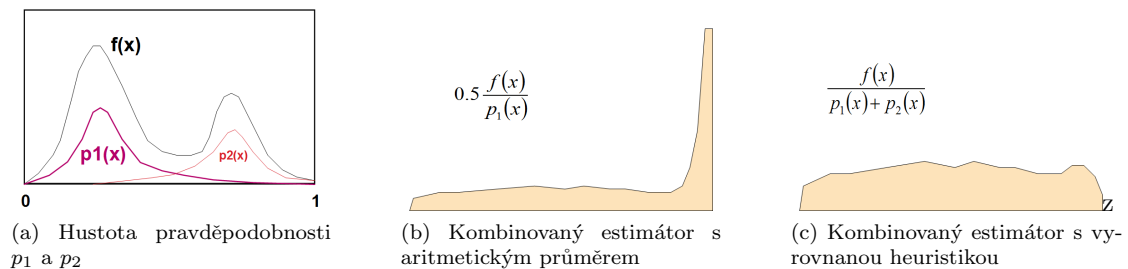
$$\hat{w}_i(x) = \frac{n_i p_i(x)}{\sum_k n_k p_k(x)}$$

Vyrovnaná heuristika nám tedy dodá váhy, které můžeme dosadit do kombinovaného estimátoru:

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

kde je vidět, že při vyrovnané heuristice hodnota příspěvku libovolného vzorku  $X_{i,j}$  k estimátoru nezávisí na tom, ze které vzorkovací techniky byl vzorek pořízen (tato vlastnost je vyjádřena v názvu heuristiky).

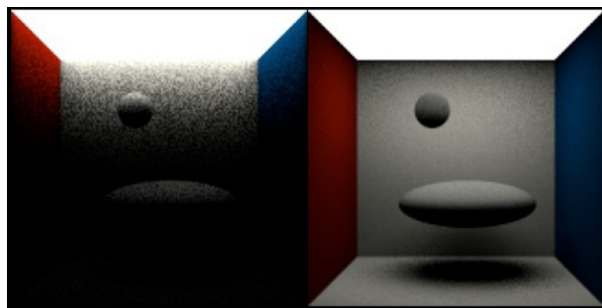
Vyrovnaná heuristika je téměř optimální, žádný jiný kombinovaný estimátor ve tvaru 2 nemůže mít rozptyl o mnoho menší. Proč funguje tak dobře? V estimátoru je  $f/p$  a pokud je někde  $p$  malé, tak jsou příspěvky estimátoru na takových místech velké, ačkoliv je to nevhodné (to přesně dělá aritmetický průměr [obrázek 3b], protože sčítá obě hustoty, které mají váhu 0.5 i v místech, kde jedna z nich má velmi malé hodnoty). V kombinovaném estimátoru s vyrovnanou heuristikou jsou ve jmenovateli všechny hustoty a stačí, aby jejich součet nešel k nule a příspěvky jsou v takových místech rozumnější (obrázek 3c).



Obrázek 3: Vztah vah a vzorkovacích technik vzhledem k příspěvku kombinovaného estimátoru

## 8.2 Aplikace MIS na výpočet přímého osvětlení

Vrátíme se k vzorkování světla a BRDF, kde je při použití vyrovnané heuristiky velmi důležité si uvědomit, že obě pravděpodobnosti musí mít stejnou doménu. Uniformní vzorkování plochy světla nemá z pohledu bodu  $x$  uniformní pdf. Pokud použijeme vzorkování BRDF a světla, tak



Obrázek 4: Příspěvky jednotlivých technik do výsledného obrázku. Vlevo:  $w_1 * \text{BRDF}$ . Vpravo:  $w_2 * \text{vzorkování zdroje}$

máme dvě vzorkovací strategie  $p_1$  a  $p_2$ , nechť obě používají stejný počet vzorků  $n$ , v takovém případě mají následující váhy:

$$\hat{w}_1(\omega) = \frac{n \cdot p_1(\omega)}{n \cdot p_1(\omega) + n \cdot p_2(\omega)} = \frac{p_1(\omega)}{p_1(\omega) + p_2(\omega)}$$

Jak se spočítají  $p_1$  a  $p_2$ ?  $p_1$  je hustota vzorkování podle BRDF a závisí tedy na BRDF v bodě  $x$ , např. pro Lambertovskou BRDF  $p_1(\omega) = \frac{\cos \theta_x}{\pi}$ . Uniformní vzorkování zdroje vypadá z pohledu bodu  $x$  jako

$$p_2(x) = \frac{1}{|A|} \frac{\|x - y\|^2}{\cos \theta_y}$$

kde první člen  $1/|A|$  je tam kvůli tomu, aby  $\int p_2 = 1$  (pdf je všude mimo světlo 0), vzdálenostní člen  $\|x - y\|^2$  korekci vzdálenosti (pokud se světlo dvakrát vzdálí, tak bude jeho plocha z pohledu bodu 4x menší a protože pdf  $p_2$  není nula jen na světle, tak musí mít větší hodnotu, aby integrál zůstal 1) a  $\cos \theta_y$  kvůli natočení.

## 9 Použití MIS v path traceru

V každém bodě potřebujeme spočítat dva příspěvky: přímé a nepřímé osvětlení. Na přímé osvětlení používáme dvě vzorkovací techniky – vzorkování světla a BRDF – které se spojí v kombinovaném estimátoru. Protože technika vysílání paprsku pro nepřímé osvětlení (vzorkování BRDF) je stejná, jako jedna z technik pro počítání přímé, lze "recyklovat" paprsek vyslaný podle vzorkování BRDF při výpočtu přímého osvětlení.

Pro výpočet osvětlení na bodě  $x$  vždy vygeneruji dva paprsky – jeden vznikne tak, že vygeneruji bod na povrchu světla a spojím ho s bodem  $x$  a druhý vznikne tak, že vyberu směr pomocí BRDF a pošlu paprsek tím směrem. První paprsek bude přispívat pouze na přímé osvětlení, zatímco druhý bude přispívat na přímé osvětlení (pokud se náhodou strefí do světla, což nemusí být moc často, viz obrázek 1b) a na nepřímé osvětlení, kde přijímá světlo od nějakého jiného bodu ve scéně (což může občas být i zdroj světla, protože i na ten dopadá jiné světlo).

Je potřeba si uvědomit, že MIS váhy se použijí pouze u výpočtu přímého osvětlení, protože to je počítané pomocí kombinovaného estimátorů dvou technik. Nepřímé osvětlení se připočte celé. Při výpočtu MIS vah je potřeba vzít v úvahu pravděpodobnost ukončení cesty (ruská ruleta)

Pokud se ve scéně vyskytuje více světla, je několik možností, jak se s tím vypořádat:

- Poslat stínový paprsek ke každému zdroji světla. Tento způsob může být nepraktický při vyšším počtu světla (100+)

- Vybrat náhodně jeden zdroj (ideálně pravděpodobnost podle výkonu) a na tom teprve náhodně vybrat bod  $y$  určující stínový paprsek. Existuje dobrá heuristika, kdy tuto metodu používám až po prvním nelesklém odrazu, zatímco pro všechny ostatní vrcholy cesty počítám přímé osvětlení od všech zdrojů světla. Pravděpodobnost výběru zdroje ovlivňuje hustoty (a tedy i váhy) v MIS

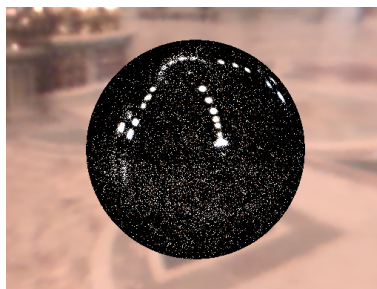
Druhá technika (BRDF) je přirozeně nezměněna tím, kolik je ve scéně světla. Ta vždy vrhne právě jeden paprsek a zjistí, zda jste zasáhli světlo.

## 10 Osvětlení z mapy prostředí

Mapa prostředí je jenom jiný typ zdroje světla, místo zdroje světla ve scéně máme nekonečně velkou kouli s obrazem kolem celé scény, která ji osvětluje. Tato technika je často používána ve filmech a hrách. Mapu prostředí lze na kouli různě mapovat.

Pro počítání osvětlení se u mapy prostředí používají dvě techniky, vzorkování BRDF (generuje paprsky podle BRDF bodu a pokud nezasáhnou žádný objekt, použije se intenzita definovaná koupí mapy prostředí) a vzorkování podle důležitosti, *importance sampling* (paprsky se generují podle toho, jakou intenzitu má mapa prostředí v daném směru, např. světlá obloha má vysokou intenzitu a tudíž i vysokou pravděpodobnost).

Máme dvě vzorkovací techniky a proto lze použít MIS, která dává lepší výsledek, než individuální použití technik.



Obrázek 5: EM pomocí BRDF (300 vzorků) na difúzním povrchu. BRDF je vrch, protože BRDF má veliký lalok, tak se konstantní a málokdy se strefí do světla a tím vytváří šum

Obrázek 6: EM IS (300 vzorků) - lesklý povrch většina vygenerovaných vzorků se netrefí do laloku



Obrázek 7: MIS (300 BRDF + 300 IS) na difúzním povrchu

Obrázek 8: MIS (300 BRDF + 300 IS) na lesklém povrchu



## 11 Vlastnosti estimátorů

Řekneme, že estimátor je *nestranný*, pokud v průměru dává správnou veličinu (bez systematické chyby):

$$E[F] = Q$$

kde  $F$  je estimátor veličiny  $Q$  (náhodná veličina) a  $Q$  je odhadovaná veličina (např. integrál). Pokud estimátor není nestranný, tak je *vychýlený* ("biased"), kde systematická chyba (bias) je dána následujícím vztahem:

$$\beta = Q - E[F]$$

Mějme estimátor, který je funkcí mnoha náhodných veličin (např. v path traceru máme příspěvky cest), pak řekneme, že je *konzistentní*, pokud

$$Pr \left\{ \lim_{N \rightarrow \infty} F_N(X_1, X_2 \dots X_n) = Q \right\} = 1$$

tj. pokud chyba  $F_N - Q$  jde k nule s pravděpodobností 1. Konzistentnost a nestrannost se liší, např. estimátor, který osciluje kolem své střední hodnoty je nestranný, ale ne konzistentní.

Postačující podmínka konzistentnosti estimátoru (musí platit obě rovnosti; nestačí, aby bias šel limitně k nule) je

$$\lim_{N \rightarrow \infty} \beta[F_N] = \lim_{N \rightarrow \infty} V[F_N] = 0$$

*Střední kvadratická chyba* (Mean Square Error):

$$MSE[F] = E[(F - Q)^2]$$

Platí, že

$$MSE[F] = V[F] + \beta[F]^2$$

Z toho vyplývá, že pokud je algoritmus nestranný, tak jediným zdrojem chyby je rozptyl ( $MSE[F] = V[F]$ ). Rozptyl lze odhadnout ze vzorků  $Y_i$ , nestranný estimátor rozptylu:

$$\hat{V}[F_N] = \frac{1}{N-1} \left\{ \left( \frac{1}{N} \sum_{i=1}^N Y_i^2 \right) - \left( \frac{1}{N} \sum_{i=1}^N Y_i \right)^2 \right\}$$

*Eficiency* (účinnost) *estimátoru*:

$$\epsilon[F] = \frac{1}{V[F]TF}$$

$V[F]$  je rozptyl a  $T[F]$  je čas výpočtu (např. počet operací, počet vržených paprsků či jiná metrika).

## 12 Zobrazovací algoritmy

Zobrazovací algoritmy jsou počítány pomocí estimátorů. Různé algoritmy můžeme rozdělit podle toho, jaké vlastnosti mají:

- Nestranné: path tracing, bidirectional path tracing, metropolis light transport
- Konzistentní: progresivní fotonové mapy
- Nekonzistentní, vychýlené: nekonvergují ke správnému výsledku, ale k něčemu "hezkému" konvergují rychleji, než jiné algoritmy (=mají z principu chyby, ale uživatel to nevidí). Fotonové mapy, irradiance/radiance caching.