

Zápisky z přednášky  
Global illumination with many-light methods

Tomáš Zámečník (2012)  
Martin Kahoun (2011)

# 1 Výpočet globálního osvětlení

## 1.1 Zobrazovací rovnice v 3b formulaci

V této úvodní části se budeme zabývat výpočtem intenzity světelného paprsku. Paprsek bude vždy zadán dvěma body (místem vzniku a místem dopadu).

Následující rovnice popisuje intenzitu světelného paprsku, směřujícího z bodu  $x'$  do bodu  $x''$ .

$$L(x' \rightarrow x'') = L_e(x' \rightarrow x'') + L_r(x' \rightarrow x'') \quad (1)$$

$L$  se skládá ze dvou příspěvků:

- $L_e$  - intenzita světelného paprsku emitovaného z  $x'$  do  $x''$
- $L_r$  - intenzita světelných paprsků odražených v  $x'$

Hodnota  $L_r$  se vypočte integrací z intenzity paprsků dopadajících do  $x'$  ze všech bodů  $x$  všech povrchů scény  $M$ . Po dosazení za  $L_r$  je funkce  $L$  zadána následujícím rekurentním vztahem, známým jako zobrazovací rovnice:

$$L(x' \rightarrow x'') = L_e(x' \rightarrow x'') + \int_M L(x \rightarrow x') \cdot f_r(x \rightarrow x' \rightarrow x'') \cdot G(x \leftrightarrow x') dA_x \quad (2)$$

$f_r$  je odrazová funkce (BRDF).  $G$  je geometrický člen, závisející na vzájemné poloze bodů  $x$  a  $x'$ .

$$G(x \leftrightarrow x') = V(x \leftrightarrow x') \frac{|\cos\theta_o| |\cos\theta'_i|}{\|x - x'\|^2} \quad (3)$$

$V$  je funkce vzájemné viditelnosti bodů  $x$  a  $x'$ .

$$V(x \leftrightarrow x') = \begin{cases} 1 & \text{pokud jsou body } x, x' \text{ vzájemně viditelné} \\ 0 & \text{jinak} \end{cases} \quad (4)$$

Existuje více ekvivalentních formulací zobrazovací rovnice. Varianta zadaná vztahem (2) je význačná tím, že nepoužívá integraci přes hemisféru, ale přes všechny body scény. Neintegrujeme po úhlech, ale po bodech.

## 1.2 Měřicí rovnice v 3b formulaci

Nyní navážeme na předchozí kapitolu a uvedeme způsob výpočtu intenzity světla dopadajícího na plochu jednoho pixelu. Intenzita  $j$ -tého pixelu se vypočte podle následujícího předpisu (měřicí rovnice).

$$I_j = \int_{M \times M} W_e^{(j)}(x \rightarrow x') \cdot L(x \rightarrow x') \cdot G(x \leftrightarrow x') dA_x dA_{x'} \quad (5)$$

Zde  $x$  jsou body scény a  $x'$  body senzoru (pixelu).  $W_e^{(j)}(x \rightarrow x')$  je hodnota váhové funkce senzoru (odezva) pro paprsek z  $x$  do  $x'$ .

$L(x \rightarrow x')$  je hodnota radiance bodu  $x'$ , vypočtená podle (2).

## 1.3 Integrovaní přes světelné cesty

Nyní přeformulujeme rovnici (5) tak, že budeme integrovat místo přes body scény přes celé světelné cesty (posloupnosti bodů, v nichž se paprsky odrážejí při cestě ze zdroje světla do senzoru - pixelu)

$$I_j = \int_{\Omega} f_j(\bar{x}) d\mu(\bar{x}) \quad (6)$$

$\Omega$  je zde prostor (množina) světelných cest mezi zdroji světla a  $j$ -tým pixelem.  $\mu(\bar{x})$  je míra na tomto prostoru.

$$d\mu(\bar{x}) = dA_{x_0}dA_{x_1}dA_{x_2}\dots \quad (7)$$

$x_0, x_1, x_2, \dots$  odpovídají 1., 2., 3., ... bodům cest  $\bar{x}$ .

$f_j(\bar{x})$  je příspěvek cesty  $\bar{x}$  k intenzitě  $j$ -tého pixelu. Tato funkce říká, kolik procent světla dojde ze zdroje do senzoru.

Integrováním přes cesty jsme se zbavili rekurentního zápisu. Jinými slovy  $I_j$  v (6) je hodnotou integrálu, nikoliv řešením integrální rovnice.

Nestranný odhad hodnoty  $I_j$  vypočteme Monte-Carlo integrací:

$$I_j \approx \frac{f_j(\bar{x})}{p(\bar{x})} \quad (8)$$

$p(\bar{x})$  je hodnota hustoty pravděpodobnosti cesty  $\bar{x}$  v prostoru cest  $\Omega$ . Nebo-li jedná se o pravděpodobnost, že cesta  $\bar{x}$  půjde z vrcholu  $x_0$  do  $x_1, \dots$ , až nakonec skončí v  $x_k$ :

$$p(\bar{x}) = p(x_0, \dots, x_k) = p(x_0)p(x_1|x_0)p(x_2|x_0, x_1)\dots p(x_k|x_0, \dots, x_{k-1}) \quad (9)$$

Poznámka: Užitím Bayesova vzorce odvodíme, že  $p(x_0, \dots, x_k) = p(x_k, \dots, x_0)$ . Tzn. nezáleží, zda uvažujeme cesty směřující ze zdroje světla do senzoru nebo obráceně.

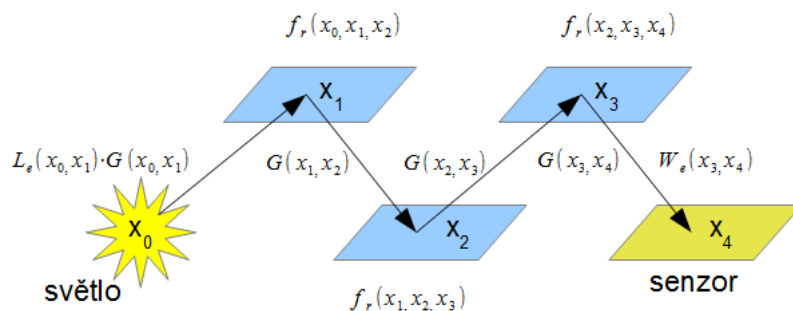
### 1.3.1 Výpočet $f_j(\bar{x})$

Nechť  $\bar{x} = (x_0, \dots, x_k)$ , potom platí:

$$f_j(\bar{x}) = L_e(x_0 \rightarrow x_1) \cdot f_r(x_0 \rightarrow x_1 \rightarrow x_2) \cdot G(x_0 \leftrightarrow x_1) \cdot \dots \cdot f_r(x_{k-2} \rightarrow x_{k-1} \rightarrow x_k) \cdot G(x_{k-1} \leftrightarrow x_k) \cdot W_e^{(j)}(x_{k-1} \rightarrow x_k) \quad (10)$$

Poznámka: Emisní složka  $L_e$  se započítává pouze v bodě  $x_0$ .

Situace pro cestu čítající pět bodů je ilustrována následujícím obrázkem:



### 1.3.2 Renderování

Konkrétní odhad intenzity pixelu  $I_j$  se získá Monte Carlo integrací následovně:

for  $i = 1$  to  $\text{pocet\_cest}$  do:

- Vygeneruj cestu  $x_i$ , pro níž  $W_e$  je nenulové (tj. "skrz" pixel).
- Aktualizuj odhad  $I_j$ :

$$I_j += \frac{1}{\text{pocet\_cest}} \cdot \frac{f_j(x_i)}{p(x_i)}$$

## 2 Instant radiosity

Ukážeme si nyní metodu publikovanou Kellerem v roce 1997, kterou nazval instatní (okamžitá) radiozita [2]. Celá třída metod odvozených z instantní radiozity je pak často nazývána *many-lights methods* nebo *virtual point light methods*.

Základem metody je přidání nových (virtuálních) světél do scény (VPL - virtual point lights). Při samotném vykreslování počítáme pouze přímé osvětlení a to jak z reálných, tak i z virtuálních světél.

Výpočet osvětlení má dvě fáze:

1. Vygenerování virtuálních světél
2. Vykreslení scény výpočtem přímého osvětlení

Poznámka: Metoda předpokládá, že všechny povrchy jsou čistě difuzní.

### 2.1 Tvorba VPL

Přidáváním nových světél se snažíme dosáhnout vzhledu scény, kde je použito i nepřímého osvětlení. VPL generujeme tak, že vrháme paprsky ze skutečných světelných zdrojů. Tam, kde paprsek dopadne na povrch scény, přidáme nové virtuální světlo. Paprsek se může dále odrážet. S každým odrazem vzniká v daném bodě nové VPL. Cestu paprsku ukončíme principem Russian roulette.

Každý virtuální zdroj považujeme za čistě difuzní emitör a pro radianci přicházející od něj platí:

$$L(y \rightarrow x) = \Phi \frac{\rho_d(y)}{\pi} \cos \theta_y,$$

kde  $\Phi$  je světelný tok nesený částicí, která vygenerovala virtuální světlo (výpočet  $\Phi$  viz. photon tracing ve fotonových mapách). Dále  $\rho_d(y)$  je difúzní odrazivost (albedo) v bodě  $y$ .

### 2.2 Informace uložené ve VPL

Pokud je splněn předpoklad, že všechny povrchy jsou čistě difuzní, uchovááme u každého světla jeho pozici, normálový vektor povrchu a intenzitu nesenou paprskem světla, z něž VPL vzniknul. Lesklé povrchy jsou probrány v další přednášce.

### 2.3 Vykreslení scény s VPL

Ve druhém kroku počítáme pomocí tohoto seznamu virtuálních světél přímé osvětlení scény a to následujícím způsobem:

$$L(x, \omega_o) = L_e(x, \omega_o) + \sum_{k=1}^n L(y_k \rightarrow x) G(y_k \leftrightarrow x) f_r(y_k \rightarrow x \rightarrow \omega_o) \quad (11)$$

Tato rovnice není ničím jiným, než Monte Carlo estimátorem zobrazovací rovnice ve tvaru (2). Virtuální bodové zdroje pak představují vzorky radiance na plochách scény. Instantní radiozita je tedy podmnožinou obousměrného sledování cest (*bidirectional path tracing*). Používáme ovšem jen prvního dopadu paprsku z kamery a navíc vrcholy cest od světelných zdrojů (tj. virtuální bodové zdroje) se přepoužívají pro všechny pixely.

## 2.4 Stínové mapy

Pro výpočet viditelnosti mezi jedním bodovým zdrojem a všemi viditelnými body scény lze s výhodou využít algoritmu *stínových map (shadow maps)*. Stínové mapy fungují ve dvou krocích: Nejprve vyrenderujeme scénu z pohledu světla a do bufferu ukládáme vzdálenost světelného zdroje a objektu scény, čímž vytvoříme stínovou mapu. Ve druhém průchodu renderujeme už normálně z pohledu kamery. V každém bodě scény se pak dotazujeme do stínové mapy a porovnáváme vzdálenost bodu od světla s hodnotou ve stínové mapě: pokud je vzdálenost bodu větší než hodnota ve stínové mapě, pak je bod zastíněn, jinak je osvětlen.

## 2.5 Šum

Přidáním VPL vneseme do scény i chyby (artefakty) v podobě míst s vysokou intenzitou, lokalizovaných typicky v "rozích scény" - pokud je zde umístěno nějaké VPL. Pokud je VPL v rohu, jsou body sousední přivrácené stěny velmi blízko (vzdálenost se blíží nule) a tudíž přesevřtlené (geometrický člen  $G(x \leftrightarrow x')$  zobrazovací rovnice je extrémně vysoký (jde k  $\infty$ )).

Vliv šumu (přesevřtlená místa) je dobře patrný na následujícím obrázku:



## 2.6 Odstranění šumu ořezáváním

Nahradíme-li výpočet (11) příspěvku VPL k osvětlení bodu  $x$  předpisem:

$$L(x, \omega_o) = L_e(x, \omega_o) + \sum_{k=1}^n L(y_k \rightarrow x) \min\{c, G(y_k \leftrightarrow x) f_r(y_k \rightarrow x \rightarrow \omega_o)\}, \quad (12)$$

přesevřtlená místa zmizí. Stane se tak díky, tomu, že hodnotu  $G(y_k \leftrightarrow x) f_r(y_k \rightarrow x \rightarrow \omega_o)$  shora omezili vhodně zvolenou konstantou  $c$ . Toto řešení však není uspokojivé, neboť přestože se vzhled scény zlepšil (zmizela přesevřtlená místa), vnesli jsme do ní ve skutečnosti další chybu, protože ořezáním jsme ztratili i část energie, která do scény patřila (nevznikla šumem).

## 3 Výpočet globálního osvětlení v reálném čase

V této kapitole se budeme zabývat metodami, urychlujícími výpočet globálního osvětlení s VPL.

### 3.1 Reflective Shadow Maps

První metoda se soustředí na urychlení distribuce virtuálních světél. Budeme generovat pouze světla přímo viditelná z reálného světelného zdroje. Využijeme stínové mapy, popsané v kapitole 2.4.

Každý bod uložený ve stínové mapě se stane novým VPL. Generujeme tedy pouze cesty čítající dva body (světelný zdroj a VPL).

### 3.1.1 Příliš mnoho světél

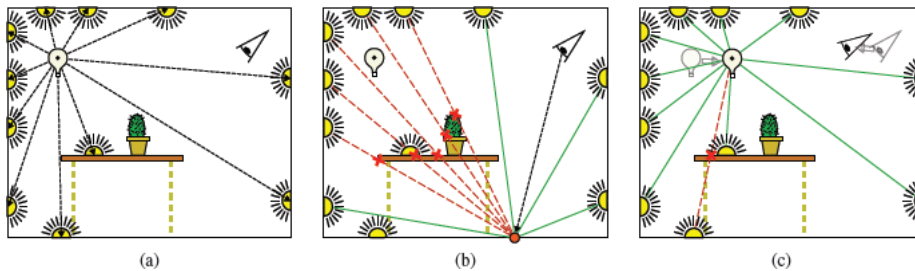
Nevýhodou této metody je, že pokud má stínová mapa vysoké rozlišení, vygeneruje se zbytečně velké množství světél, což má za následek zpomalení výpočtu nepřímého osvětlení při vykreslování scény.

Tento problém lze řešit například podvzorkováním světelné mapy.

Jinou možností je do výpočtu osvětlení bodu  $x$  zahrnout pouze relevantní VPL. Příspěvek virtuálního světla k intenzitě bodu  $x$  se vynásobí vahou v závislosti na vzdálenosti VPL od  $x$  ve stínové mapě (v projekci světla).

## 3.2 Incremental Instant Radiosity

Nyní popíšeme urychlovací metodu, která se zaměřuje na případ, kdy se světla mohou pohybovat, scéna však zůstává statická. Klíčem k urychlení je předpoklad, že se světla mezi dvěma snímky pohnula pouze "málo" a tedy, že stínové mapy v obou snímcích pro dané světlo sdílí většinu bodů. V novém snímku využijeme virtuálních světél ze snímku předchozího. VPL, která jsou zakrytá překážkou nebo se nacházejí mimo "zorné pole" světla odstraníme a přidáme nová. Budeme se snažit přidávat nová virtuální světla tak, abychom zároveň zachovávali rovnoměrnost jejich pokrytí.



- (a) VPL vytvořená metodou Reflective Shadow Maps
  - (b) Výpočet nepřímého osvětlení (pro oranžový bod)
  - (c) Posun světla - jedno z virtuálních světél se stane nevalidním.
- Obrázek převzat z [5].

Algoritmus v bodech:

1. Určení platnosti všech stávajících VPL
2. Odstranění nevalidních VPL (v zájmu zachování rovnoměrného pokrytí i některých validních)
3. Vytvoření nových VPL
4. Výpočet intenzity nových VPL
5. Projekce VPL podle nové pozice světla

### 3.2.1 Odstraňování VPL

Kromě validních světél odstraníme i některá další, abychom udržovali rovnoměrnost pokrytí. Kvůli rychlosti využijeme pro odstraňování jednoduchého hladového algoritmu. Budeme odstraňovat virtuální světla z míst, kde mají největší hustotu (jsou mezi nimi malé vzdálenosti). Nejprve z projekce VPL (z pohledu světla) vytvoříme triangulaci (vrcholy odpovídají virtuálním světélům). Smažeme světlo, které má v triangulaci nejkratší hrany. Počet světél, která se mohou smazat většinou určíme předem (kvůli zachování snímkovací frekvence).

### 3.2.2 Přidávání VPL

Nová světla budeme přidávat do míst s nejmenším pokrytím.

Z triangulace vytvoříme Voronoiův diagram a nalezneme bod, který má největší vzdálenost k nejbližším virtuálním světlům (bude to některý z průsečíků hran diagramu) a do něj přidáme nové VPL. Vzdálenost se nepočítá ve scéně, ale na projekci světelného zdroje. Stejně jako v případě mazání, budeme počet přidávaných světél shora omezovat konstantou, abychom příliš nesnižovali snímkovou frekvenci.

## 3.3 Imperfect Shadow Maps

Technika probraná v této kapitole se zabývá urychlením výpočtu stínových map. Cílem je aproximace globálního osvětlení dynamických scén v reálném čase.

Při generování stínové mapy nejprve náhodně určíme množinu bodů, kde vypočteme hloubku. Hodnoty ve zbývajících bodech budeme interpolovat pomocí Pull-push algoritmu.

### 3.3.1 Pull-push algoritmus

Nechť je stínová mapa uložena ve čtvercové matici se stranou délky  $2^n$ . Vygenerujeme stínové mapy s nižším rozlišením o velikostech  $2^{n-1}, \dots, 1$ . Cílem je interpolovat chybějící hodnoty. Interpolace má dvě fáze:

1. Procházíme od mapy s nejvyšším rozlišením k mapě s nejnižším a počítáme hodnoty pixelů průměrováním (čtyř jmenších pixelů z předchozí mapy). Pokud žádný ze čtyř pixelů není definován, necháme hodnotu nevyplněnou.
2. Procházíme od mapy s nejnižším rozlišením. Tam, kde narazíme na chybějící hodnotu ji vyplníme zkopírováním hodnoty z příslušného místa hrubší mapy.

Poznámka: Pokud při průchodu dolů dospějeme k mapě, jež má všechny hodnoty definované, nemusíme vytvářet další mapy s hrubším rozlišením.

## Reference

- [1] Hašan M., Křivánek J., Walter B., Bala K., *Virtual Spherical Lights for Many-Light Rendering of Glossy Scenes*, Proc SIGGRAPH Asia '09.
- [2] Keller Alexander, *Instant radiosity*. Proc. SIGGRAPH '97.
- [3] Kollig Thomas, Keller Alexander, *Illumination in the Presence of Weak Singularities*, Monte Carlo And Quasi-monte Carlo Methods, 2004.
- [4] S. Laine et al., *Incremental Instant Radiosity for Real-Time Indirect Illumination*, Eurographics Symposium on Rendering (2007).
- [5] T. Ritschel et al., *Imperfect Shadow Maps for Efficient Computation of Indirect Illumination*, ACM Trans. Graph. 27(5) (Proceedings SIGGRAPH Asia 2008).