

Počítačová grafika III (NPGR010)

Přednáška: Důležitost a dualita v zobrazování, Obousměrné sledování cest

Ivan Kuckir
12. prosince, 2012

1 Důležitost a dualita v zobrazování

1.1 Důležitost

Doted' jsme zkoumali pouze cestu světla ze zdroje ke kameře a s ní související *radianci*. Pro účely následující látky by se nám však hodilo se podívat se na problém z opačné strany, směrem od kamery. Zaved' me si veličinu *důležitost* (importance) se značkou W .

Radiance pochází ze světelného zdroje a značí, jak moc je bod "ovlivněn" zdrojem v daném směru. Důležitost pochází z kamery a značí, jak moc může bod ovlivnit obraz (jak je důležitý pro obraz) na kameře v daném směru. Nulová radiance znamená, že světlo se do bodu z daného směru nikdy nedostane. Obdobně, nulová důležitost znamená, že kamera daný bod z daného směru nikdy "neuvídí".

Pro radianci jsme uvažovali veličinu L_e , která značila emitovanou radianci, tedy že objekt emituje světlo (světelný zdroj). Pro důležitost uvažujeme veličinu W_e , která značí emitovanou důležitost, tedy že bod bude přijímat světlo (kamera).

Emitovanou radianci si volíme sami a určuje vlastnosti osvětlení. Emitovanou důležitost si též volíme sami a určuje vlastnosti snímání.

Formálně můžeme důležitost definovat obdobně jako radianci:

$$W(x, \omega_0) = W_e(x, \omega_0) + \int_{H(x)} W(r(x, \omega_i), -\omega_i) \cdot f_r(x, \omega_o \rightarrow \omega_i) \cdot \cos(\theta_i) d\omega_i$$

Kde

- $W_e(x, \omega_0)$ je emitovaná důležitost
- $W(x, \omega_0)$ je ustálená důležitost
- $f_r(x, \omega_o \rightarrow \omega_i)$ je funkce BRDF s prohozenými argumenty

1.2 Měřicí rovnice

Jako *světelný zdroj* jsme označovali povrchy, které mají nenulovou emitovanou radianci. Obdobně slovem *senzor* označme povrchy, které mají nenulovou emitovanou důležitost.

Dosud jsme řešili pouze výpočet radiance v izolovaných bodech pomocí zobrazovací rovnice (rendering equation). Obdobným způsobem jsme mohli řešit i důležitost v bodech. V praxi nás ale zajímá barva celého pixelu, tedy všech bodů na jeho ploše.

Obecně můžeme mít ve scéně spoustu bodů x majících nenulovou emitovanou důležitost W_e . V praxi tyto body intuitivně rozdělíme na souvislé senzory I_j a jednu funkci emitované důležitosti rozdělíme na několik funkcí $W_e^{(j)}$, které přiřazují nenulovou emitovanou důležitost pouze bodům z daného senzoru. Vyslovme si nyní *Měřicí rovnici*, která by měla říci, co hodláme považovat za barvu senzoru I_j .

$$I_j = \int_M \int_{H(x)} W_e^{(j)}(x, \omega) \cdot L_i(x, \omega) \cdot \cos(\theta_i) d\omega dA$$

Kde

- M jsou všechny povrchy scény, včetně senzorů
- $H(x)$ je hemisféra ve směru normály v bodě x
- $W_e^{(j)}(x, \omega)$ je emitovaná důležitost v bodě x ve směru ω
- $L_i(x, \omega)$ je přichodzí radiance
- I_j je naše "barva", tedy jakási odezva senzoru na radianci v scéně

Ačkoli integrujeme přes všechny povrchy, $W_e^{(j)}(x, \omega)$ zajistí, že nenulové hodnoty budou pouze na daném senzoru. Měřicí funkce je velmi podobná irradianci, avšak navíc zohledňuje jakousi citlivost povrchu na světlo (emitovaná důležitost).

1.3 Sledování světla

V předchozím textu jsme do scény zavedli kromě světél s emitovanou radiancí také senzory s emitovanou důležitostí.

V algoritmu Sledování cest jsme vysílali paprsky z kamery (senzoru), které cestovaly scénou a snažily se trefit na světelný zdroj. Obdobně můžeme zavést algoritmus Sledování světla, který bude fungovat opačným směrem. Bude vysílat paprsky ze světelných zdrojů, které budou cestovat scénou a snažit se trefit kameru (senzor). Při realizaci sledování světla můžeme použít obdobné postupy, jako u sledování cest.

V praxi se Sledování světla příliš nepoužívá. Obvyklé scény totiž většinou mají velké světelné zdroje a malé senzory, navíc světla bývají často difúzní (emitovaná radiance obdobná ve všech směrech), zatímco senzory snímají hlavně "dopředu" (emitovaná důležitost je nejvyšší ve směru normály). Tím pádem při cestě od světla je mnohem těžší trefit kameru (Light tracing), než při cestě opačným směrem z kamery ke světlu (Path tracing), a výsledné obrázky konvergují příliš pomalu.

Jsou však případy, kdy Sledování světla má lepší výsledky než Sledování cest. Představme si skleněnou kouli na rovině a u ní světelný zdroj. Koule by měla vrhat kaustiku. Pokud vyšleme paprsky z kamery, jen málo z nich trefí oblast kaustiky. Po následném odrazu a průchodu koulí se rozptýlí ještě víc téměř žádný nezasáhne světelný zdroj. Pokud vyšleme paprsky ze zdroje, téměř polovina z nich zasáhne kouli a po lomu se soustředí v kaustice. Při následném odrazu je velká šance, že některé z nich zasáhnou kameru.



Obrázek 1: První obrázek je vygenerován PT, druhý LT a třetí BPT.

Na obrázku výše byl použitý bodový světelný zdroj a bodová kamera. V prostředním obrázku se body povrchů vykreslují tak, že se

při každém dopadu paprsku zkusí spojit přímo s kamerou a podle aktuálních hodnot a úhlů se spočte příspěvek k obrázku. Avšak všechny paprsky, které projdou skleněnou koulí, se musí zalomit podle jednoznačného pravidla. Proto tímto postupem nelze vykreslit kouli (tj. paprsky, které se lámou skrze kouli a trefují kameru). Stejný problém by byl i u zrcadlového povrchu.

Pokud mezi bodem povrchu a kamerou leží koule, šlo by analyticky spočítat odchozí úhel takový, aby po zalomení trefil kameru. Avšak tento postup zde nebyl použit.

2 Přenos světla jako integrál přes cesty

2.1 Motivace

V předchozím textu jsme definovali naše funkce (zobrazovací rovnice, rovnice odrazu) pro jednotlivé body a jejich hodnota byla integrálem přes všechny úhly nad daným bodem. Levá strana rovnice se pak objevovala na pravé straně, což automaticky vedlo k algoritmům, které rekurzivně "rozbalují" pravou stranu, dokud hodnota není dost přesná.

V praxi používáme postupy, které hledají cesty (posloupnosti bodů) světla ve scéně a bylo by dobré naše rovnice těmto postupům přiblížit. Proto zkusíme naše rovnice přepsat na *integrál přes prostor cest*.

2.2 Tříbodová formulace přenosu světla

Místo bodu a směru zkusme uvažovat bod a k němu další bod scény, který je v daném směru nejbližší. Při této úvaze můžeme přeformulovat značení radiance a BRDF:

$$L(x \rightarrow x') \equiv L(x, \omega)$$

$$f_r(x \rightarrow x' \rightarrow x'') \equiv f_r(x', \omega_i \rightarrow \omega_o)$$

Stejným způsobem můžeme přeznačit i zobrazovací rovnici:

$$L(x', x'') = L_e(x', x'') + \int_M L(x, x') \cdot f_r(x \rightarrow x' \rightarrow x'') \cdot G(x \leftrightarrow x') dM_x$$

Kde M jsou všechny povrchy scény a $G(x \leftrightarrow x')$ zastupuje kosínové faktory z původní rovnice a zároveň viditelnost bodů:

$$G(x \leftrightarrow x') = V(x \leftrightarrow x') \cdot \frac{|\cos \theta_0 \cos \theta'_i|}{\|x - x'\|^2}$$

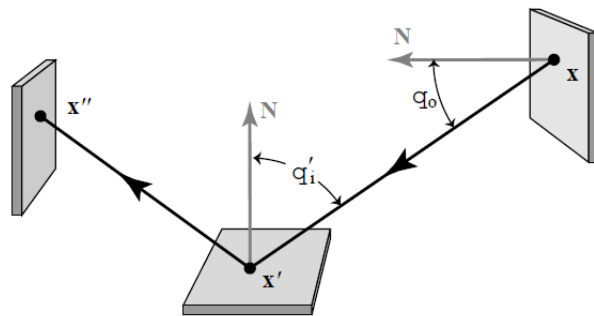
Stejným způsobem můžeme přepsat i měřicí rovnici. Jen místo všech dvojic (bod, směr) uvažujeme všechny dvojice (bod, bod).

$$I_j = \int_M \int_M W_e^{(j)}(x \rightarrow x') \cdot L(x \rightarrow x') \cdot G(x \leftrightarrow x') dM_x dM_{x'}$$

2.3 Světlo jako integrál přes cesty

Pod pojmem *cesta* délky k uvažujme nějakou k -tici bodů scény.

$$\bar{x} = x_0 \dots x_k$$



Obrázek 2: Tříbodová formulace přenosu světla.

Prostor všech cest (všech délek) označme Ω . Naše nová měřicí rovnice by měla být integrálem přes všechny cesty. Měla by vypadat nějak takto:

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

Kde f_j má být funkce, která stanoví příspěvek dané cesty k celkové hodnotě na senzoru. Tato funkce by měla brát v potaz viditelnost na jednotlivých úsecích cesty, BRDF na jednotlivých bodech, radianci na začátku cesty a důležitost na její konci. Můžeme to zapsat takto:

$$f_j(\bar{x}) = L_e(x_0 \rightarrow x_1) \cdot G(x_0 \leftrightarrow x_1) \cdot$$

$$\left(\prod_{i=1}^{k-1} f_r(x_{i-1} \rightarrow x_i \rightarrow x_{i+1}) \cdot G(x_i \leftrightarrow x_{i+1}) \right) \cdot W_e^{(j)}(x_{k-1} \rightarrow x_k)$$

2.4 Aplikace Monte Carlo metod

Připomeňme si, jak vypadá obecně zapsaná měřicí rovnice, kterou řešíme:

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

Tu lze řešit klasickými metodami Monte Carlo s estimátorem:

$$I_j \approx \frac{f(\bar{x})}{p(\bar{x})}$$

Jak vypočítat $f(\bar{x})$ již víme z předchozí kapitoly, takže zbývá určit hustotu pravděpodobnosti pro danou světelnou cestu.

Všimněme si, že pokud vybíráme k -tý bod cesty x_k , potom

$$p(x_k) = p(x_0 \dots x_k | x_0 \dots x_{k-1}) = \frac{p((x_0 \dots x_{k-1}) \wedge x_k)}{p(x_0 \dots x_{k-1})}$$

To můžeme upravit na

$$p(x_0 \dots x_k) = p(x_k) * p(x_0 \dots x_{k-1})$$

Tedy hustota pravděpodobnosti výběru celé cesty odpovídá součinu hustot pravděpodobností výběrů všech její úseků.

Pokud jsme v bodě x a vybíráme další bod cesty x' , jeho výběr záleží jednak na výběru daného úhlu, jednak na vzdálenosti mezi dvěma body (body blíže mají větší hustotu pravděpodobnosti) a na úhlu, pod jakým bod x' leží (body na kolmých plochách mají větší hustotu pravděpodobnosti). Lze to zapsat takto:

$$p(x') = p(\omega') \cdot \left(\frac{|\cos \theta'_i|}{\|x - x'\|^2} \right)$$

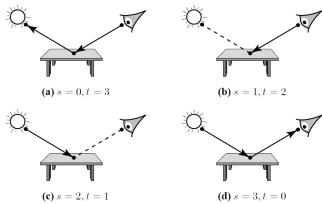
Navíc můžeme použít různé strategie (importance sampling) při výběrech směrů na kamere / zdroji (např. dle emitované důležitosti / radiance), pak tím bude ovlivněna hustota pravděpodobnosti počátečního / koncového úseku cesty.

2.5 Obousměrné sledování cest

Obousměrné sledování cest (bidirectional path tracing) má být jakýmsi zobecněním Sledování cest a Sledování světla. Stejně jako u nich se i zde snažíme najít cesty mezi světlem a kamerou. Algoritmus navíc zavádí zajímavý způsob, jak tyto cesty hledat.

Principem je hledat dvě nezávislé cesty, jednu ze světla a druhou z kamery. Jejich délky jsou kontrolovány ruskou ruletou. Poté se cesty spojí a vytvoří jednu souvislou cestu mezi světlem a kamerou.

Jednu konkrétní cestu je někdy možné získat různými způsoby v závislosti na tom, jak velká část byla vygenerovaná směrem od zdroje a jak velká část směrem od kamery. Obrázek níže ukazuje 4 možnosti vygenerování cesty délky 2. Přerušovaná úsečka značí spojení dvou cest. Obrázek (a) odpovídá algoritmu Sledování cest, zatímco obrázek (d) odpovídá algoritmu Sledování světla.



Obrázek 3: Různé vzorkovací techniky pro získávání světelných cest používané v BPT.

2.6 Implementace

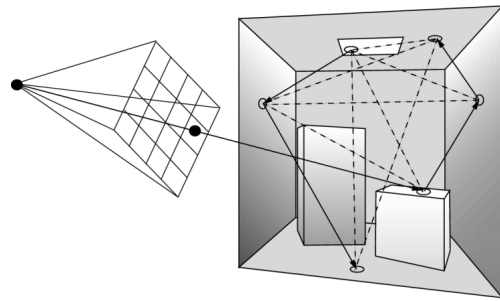
Algoritmus BPT navíc zavádí způsob, jak takové cesty efektivně hledat. Máme-li dvě části cesty (jednu od zdroje a druhou od kamery), spojováním různých bodů z první a z druhé části můžeme vytvořit spoustu kratších cest, které nám pomohou zpřesnit hodnotu. Nemusíme tedy vždy generovat a přepočítávat úplně nové cesty a ušetříme výpočetní čas.

Pro tuto metodu si můžeme sestavit kombinační estimátor. Jeho hodnota bude součet primárních estimátorů přes všechny možné "kombinační" cesty. Výběr konkrétních podcest můžeme navíc ovlivnit nějakou chytrou strategií či heuristikou, kterou reprezentuje funkce ω :

$$\sum_{s \geq 0} \sum_{t \geq 0} \omega_{s,t}(\bar{x}_{s,t}) \frac{f_j(\bar{x}_{s,t})}{p_{s,t}(\bar{x}_{s,t})}$$

Kde

- $\omega_{s,t}(\bar{x}_{s,t})$ je váha, která závisí na kombinační strategii (např. vyvážená heuristika)



Obrázek 4: Generování světelných cest v praxi.

- s a t jsou pořadová čísla vrcholů na cestách od světla a od kamery