Metropolis vzorkovanie

Matej Marko

6. mája 2012

Úvod

Našim cieľom, vo všeobecnosti, je určiť hodnotu integrálu $I = \int f(x)\mathrm{d}x$. Odhad pomocou metódy Monte Carlo vyzerá nasledovne:

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\xi_i)}{p(\xi_i)}$$

Vzorky ξ_i generujeme z rozdelenia daného hustotou $p(x)$. Pre lepší odhad je výhodné, aby rozdelenie $p(x)$ zodpovedalo priebehu funkcie $f(x)$ (znížime tak rozptyl odhadu). Túto podmienku je ťažké splniť. Často nepoznáme priebeh funkcie $f(x)$, alebo je zložité efektívne generovať vzorky z daného rozdelenie (integrovanie hustoty + inverzná funkcia).

Tento problém sa prejavuje aj pri odhadovaní illumination integrálu:

$$L_o = \int\limits_{\Omega} L_i(\omega_i)BRDF(\omega_i,\omega_o)\cos\theta\mathrm{d}\omega_i$$

Vzorky generuje iba podľa BRDF nakoľko priebeh $L_i(\omega_i)$ nepoznáme. V skutočnosti by bolo výhodné vzorkovať celé svetelné cesty a to s hustotou, ktorá je úmerná prispévkom týchto ciest. T. j. cheme generovať svetelné cesty s hustotou danou:

$$f_{pdf}(x) = \frac{f(x)}{I(f)}$$

$$I(f) = \int\limits_{\Omega} f(x)\mathrm{d}x$$

Metropolis vzorkovanie

Hlavnou výhodou metropolis vzorkovania je fakt, že dokáže generovať vzorky s prevdepodobnosťou úmernou funkcii $f(x)$. Potrebuje pri tom iba vedieť vyhodnotiť túto funkciu. Môžme tak efektívne vzorkovať svetelné cesty aj v scénach, kde je malá pravdepodobnosť, že (Bidirectional) Path Tracing vygeneruje cestu z nenulovým príspevkom (viď príklady na slidoch).

Máme danú funkciu $f(x) : \Omega \to \mathbb{R}$, kde Ω je stavový priestor. Vzorky sú postupne generované ako Markovský reťazec. Novú vzorku x_i vygeneruje pomocou súčasnej vzorky x_{i-1}. Za predpokladu, že $x_{i-1} \sim f(x)$ (x_{i-1} bolá vygenerovaná s pst. úmernou $f(x)$), bude aj $x_i \sim f(x)$.

Generovanie novej vzorky (prechod do nového stavu) sa uskutočňuje pomocou tzv. mutácií. Sú to náhodné transformácie, ktoré sú popisané prechodovou funkciou:

$$T(x \to x') = \text{ pst. prechodu zo stavu } x \text{ do stavu } x'$$

Mutácia vygenerovaná pomocou $T(x \to x')$ je následne prijatá (alebo zamietnutá) s pravdepodobnosťou $a(x \to x')$. T môžeme voliť ľubovoľne, musíme však dodržať podmienku ergodicity (musíme byť schopní dostať sa do každého stavu $x \in \Omega$). Prechod do nového stavu prebieha nasledovne:

- Vygeneruj x' podľa $T(x \to x')$
- Ak $\xi < a(x \to x')$, potom $x := x'$ (x' prijmeme s pst. $a(x \to x')$)
- Zaznamenaj x

(ξ je náhodná hodnota z intervalu $[0, 1]$)

Metropolis vzorkovanie sa snaží vyhýbať tým častiam Ω, na ktorých je hodnota $f(x)$ malá (napr. tmavé časti obrázka). Aj tieto časti však potrebujú nejaké vzorky, preto v sa modifikovanej variante zaznamenáva hodnota x aj x'. Tieto hodnoty sú však zaznamenané s váhami:

$$(1-a)weight \qquad \text{pre } x$$

$$(a)weight \qquad \text{pre } x'$$

kde $a = a(x \to x')$.

Kľúčom pre dosiahnutie želanej vlastnosti $x_i \sim f(x)$ je dodržanie nasledujúcej rovnosti (tzv. Deatailed Balance):

$$f(x)T(x \to x')a(x \to x') = f(x')T(x' \to x)a(x' \to x)$$

Z detailed balance odvodíme vzťah pre pravdepodobnosť prijatia:

$$a(x \to x') = \min\{1, \frac{f(x')T(x' \to x)}{f(x)T(x \to x')}\}$$

Pokiaľ zvolíme pravdepodobnosť prechodov tak, že platí:

$$\forall a, b : T(a \to b) = T(b \to a)$$

výsledná pravdepodobnosť prijatia závisí iba na príspevkoch pôvodného a nového stavu:

$$a(x \to x') = \min\{1, \frac{f(x')}{f(x)}\}$$

Vidíme teda, že prechod do vzorky, ktorej funkčná hodnota je väčšia, je vždy prijatý.

Mutácie

Pri voľbe mutácii musíme zohladniť dva protichodné požiadavky:

- Chceme preskúmať celý stavový priestor Ω
- Chceme využiť stavy (vzorky) s veľkým príspevkom. T.j konať malé zmeny a preskúmavať integrand lokálne.

Riešením je kombinovať viaceré mutačné stratégie.

Pri náhodnej prechádzke závisí pravdepodobnosť prechodu iba na "vzdialenosti" vzoriek:

$$T(x \to x') = T(|x - x'|)$$

T môžme zvoliť tak, že jeho hodnota klesá s rastúcou vzdialenosťou. Takáto mutácia by preferovala malé, lokálne zmeny.

Nezávislé vzorkovanie generuje vzorky z hustoty $p(x)$ a úplne ignoruje súčasný stav:

$$T(x \to x') = p(x)$$

Ak by sme mali $p(x) = f_{pdf}$, tak nepotrebujeme metropolis vzorkovanie, ale priamo generujeme vzorky podľa normalizovanej f. Nezávislé vzorkovanie sa zbavuje informácie o súčasnom stave, ale zaručuje preskúmanie celého stavového priestoru Ω.

Problémom mutácií s malými zmenami je fakt, že keď narazia na stav s veľkým príspevkom, nemusia sa z neho "posunúť preč".Tento jav je vidieť na príklade vzorkovania funkcie $f(x) = (x-0.5)^2$, keď sa použije iba druhá mutačná stratégia. Ak začneme "vľavo" je málo pravdepodobné, že sa podarí vygenerovať postupnosť krokov "doprava" a potom sa "prehupnúť" cez minimum funkcie v bode $x = 0.5$. Tomuto prípadu je možné sa vyhnúť občasným vygenerovaním uplne náhodnej vzorky. Ide v podstate o kombinovanie viacerých mutačných stratégií.

Aplikácia pre odhad integrálu

Majme daný integrál v tvare:

$$I = \int_{\Omega} f(x)g(x)\mathrm{d}\Omega$$

Hodnotu tohto integrálu možeme pomocou monte carlo odhadnúť nasledovne (vzorky x_i sú generované z hustoty p):

$$I \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)g(x_i)}{p(x_i)}$$

Metropolis vzorkovanie nám vracia vzorky$x_1, ..., x_N \sim f(x)$. Pravdepodobnost každej vzorky je teda:

$$\frac{f(x_i)}{\int\limits_\Omega f(x) \mathrm{d}\Omega}$$

Po dosadení môžme teda odhad pomocou metropolis napísať ako:

$$[\frac{1}{N} \sum_{i=1}^{N} g(x_i)] \cdot \int\limits_\Omega f(x) \mathrm{d}\Omega$$

Keď považujeme $f(x)$ za príspevok svetelnej cesty a $g(x)$ za pixel filter, tak vidíme, že použitie metropolis vzoriek určuje relatívny výsledok. Pre získanie absolútneho výsledku potrebujeme relatívny obrázok preškálovať priemerným jasom - hodnotou $\int\limits_\Omega f(x) \mathrm{d}\Omega$.

Start-up bias

Metropolis vzorkovanie predpokladá, že súčasný stav bol vygenerovaný s pst. úmernou $f(x)$. V skutočnosti postupnosť x_i konverguje k tomu, že zodpovedá hustote f_{pdf}, ale v praxi to nedosiahne. Problematické sú hlavne prvé členy postupnost. Zjednošene povedané, potrebujem zvoliť počiatočný stav, aby $x_0 \sim f(x)$.

Jednou z možností je zvoliť x_0 náhodne a následne **nepoužiť** niekoľko prvých iterácii algoritmu. Stále však nevieme koľko vzoriek je potrebné vynechať, preto to nie je dobré riešenie.

Druhou možnosťou je vygenerovať x_0 podľa nejakej hustoty $p(x)$. Príspevok každej vzorky bude potom prenásobený váhou:

$$w = \frac{f(x_0)}{p(x_0)}$$

Tento prístup je korektný, ale váha w vystupuje ako absolútny škálovací faktor, čo môže sposobiť, že výsledný obrázok bude príliš tmavý, alebo naopak svetlý. Výsledok je naviac zavíslý iba na jednej vzorke. Ľahko sa môže stať, že príspevok $f(x_0) = 0$ a teda celý obrázok bude čierny.

Spôsob ako odhadnúť správnu váhu w je nasledovný:

- Vygenerujeme N vzoriek $x_1, ..., x_N$, podľa hustoty $p(x)$
- Spočítame váhy $w_i = \frac{f(x_i)}{p(x_i)}$
- Iniciálnu vzorku x_0 vyberieme z x_i podľa diskrétneho rozdelenia daného váhami w_i
- Za váhu w zvolíme priemer $w = \frac{1}{N} \sum_{i=1}^{N} w_i$

Pozrime sa akú hodnotu má vlastne použitá váha w:

$$w = \frac{1}{N}\sum_{i=1}^{N} w_i = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \approx \int_{\Omega} f(x)\mathrm{d}\Omega$$

Vidíme, že takto vypočítaná váha zodpovedá odhadu priemernému jasu obrázku. Je to teda správna váha, ktorou sa majú škálovať relatívne výsledky získane metropolis vzorkovaním.

Motion blur

Predstavme si zjednodušenú simuláciu kamery s konečne dlhou uzávierkou. Majme danú funkciu $L(u, v, t)$, ktorá predstavuje hodnotu radiancie na súradniciach u, v v čase t. Stavový priestor je potom: $\Omega = [0, u_{max}] \times [0, v_{max}] \times [0, 1]$. Hodnotu j-teho pixelu (s rekonštrukčným filtrom h_j) môžeme vyjadriť:

$$I_j = \int_{\Omega} h_j(u, v) L(u, v, t)\mathrm{d}u\mathrm{d}v\mathrm{d}t$$

Ak generujeme vzorky $x_i \sim L_{pdf}$ (normalizovanej funkcie L), dostávame:

$$I_j \approx \frac{1}{N}\sum_{i=1}^{N} h_j(x_i) \cdot \int_{\Omega} L(x)\mathrm{d}\Omega$$

Mutácie použité pri generovaní vzoriek:

- Úplne nová hodnota (u, v, t) (ergodicita)
- Zmena $u, v \pm 8$pixelov, $t \pm 0.01$ (využitie miest s veľkým príspevkom)

Ďalšiu použiteľnú mutáciu predstavuje exponenciálne rozdelenie, ktoré prirodzene preferuje malé zmeny, ale zároveň je nenulové na celom definičnom obore a zaručuje tak ergodicitu.

Light Transport

Aplikácia metropolis vzorkovania v realistickej syntéze obrazu.

Stavový priestor Ω tvoria svetelné cesty od svetla do kamery. Jedna svetelná cesta je $\overline{x} \in \Omega$. Máme definovanú funciu príspevku cesty $f(\overline{x})$:

$$\begin{aligned} f(\overline{x}) = {} & L_e(x_0 \to x_1) G(x_0 \leftrightarrow x_1) W_e(x_{k-1} \to x_k) \\ & \cdot \prod_{i=1}^{k-1} BRDF(x_{i-1} \to x_i \to x_{i+1}) G(x_i \leftrightarrow x_{i+1}) \end{aligned}$$

Iniciálnu cestu je možné generovať pomocou path tracingu (prípadne použiť metódu na odstránenie start-up biasu).

Použité mútácie sú troch druhov:

- Caustic pertrubation - cesta vychádza zo svetla a po niekoľkých (resp. jednom) lesklom odraze/lome dopadne na difúzny povrch. Mierne sa zmení smer lúča zo zdroja svetla, po difúznej ploche sa cesta napojí naspäť na pôvodnú.
- Lens pertrubation - podobne ako pri kaustikách, ale vychýlime lúč idúci z kamery.
- Bidiractional mutation - úsek cesty sa nahradí novým.

Prvé dve mutácie sú lokálne, zatiaľ čo tretia zabezpečuje ergodicitu.

MLT je algoritmus úspešný predovšetkým na scénach, ktoré su komplikované pre iné, jednoduchšie algoritmy (napr. scéna obsahuje úzku štrbinu, ktorá je jediným zdrojom svetla). Na druhej strane však zanecháva typické artefakty, ktoré sú spôsobené sekvenciami lokálnych mutácií.

Najvačším problémom všetkých mutácii je nutnosť vypočítať pravdepodobnosť prechodu zo súčasného do nového (mutovaného) stavu. Napr. pri obojsmernej mutácii, musíme uvážiť všetky možnosti, ako mohol byť nový segment vygenerovaný. Výpočet týchto pravdepodobností je hlavný problém pri implementácii algoritmu MLT.

MLT in primary space

"MLT" in primary space predstavuje jednoduchšiu variantu MLT, keď využíva fakt, že vygenerovaná cesta je jednoznačne daná hodnotami nahodných čísel, ktoré sa pri generovaní použijú. Tento vektor náhodných čísel z uniformného rozdelenia na intervale $[0,1]$ sa použije pre generovanie náhodných hodnôt z iných rozdelení (napr. pre vzorkovanie BRDF).

Formálne teda môžme náš integračný problém previesť na integrovanie v mnohorozmernej jednotkovej kocke. V tomto priestore potom používame iba 2 mutácie:

- Malé kroky - pomocou exponenciálneho rozdelenia
- Veľké kroky - náhodný výber bez ohľadu na súčasný stav