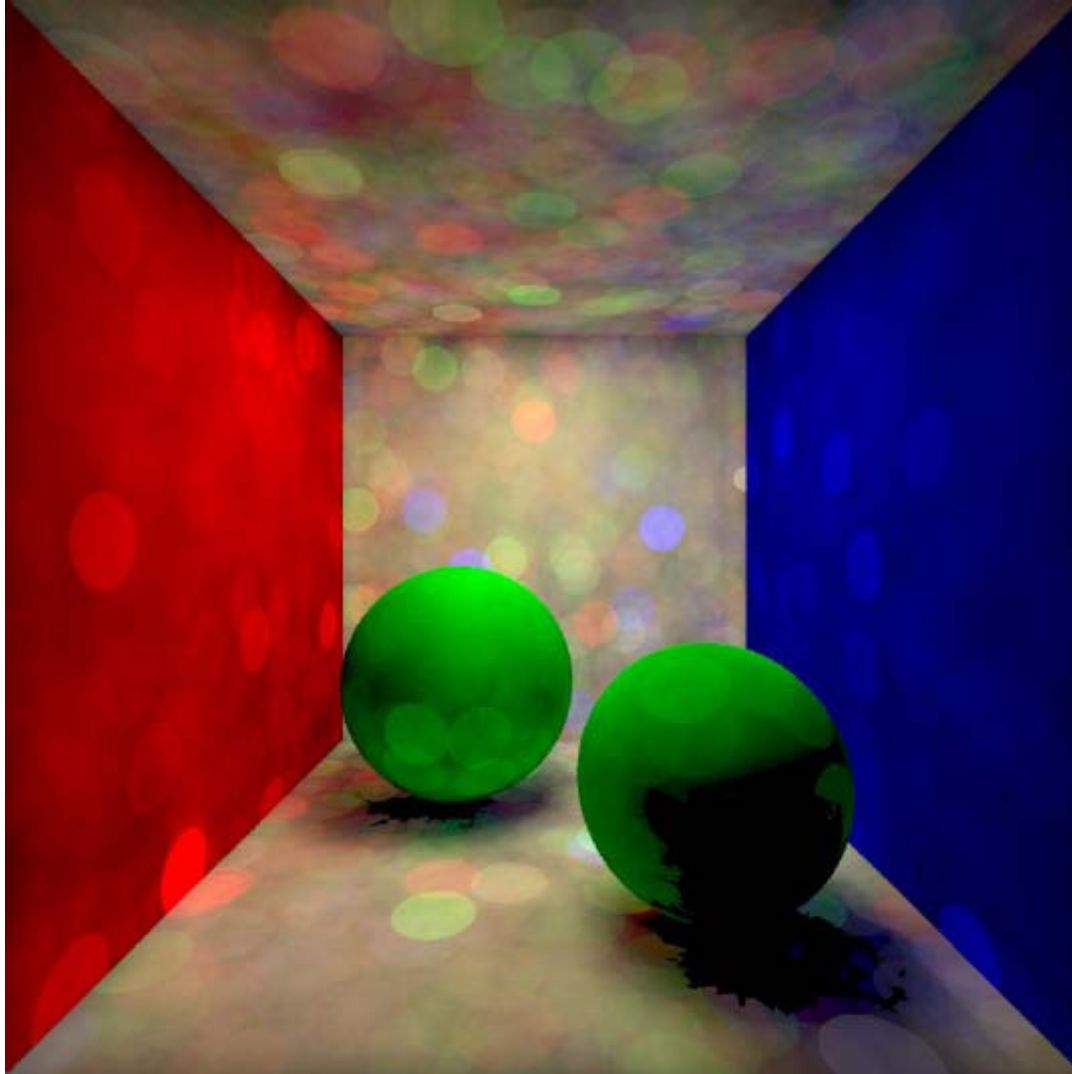

Progressive photon mapping

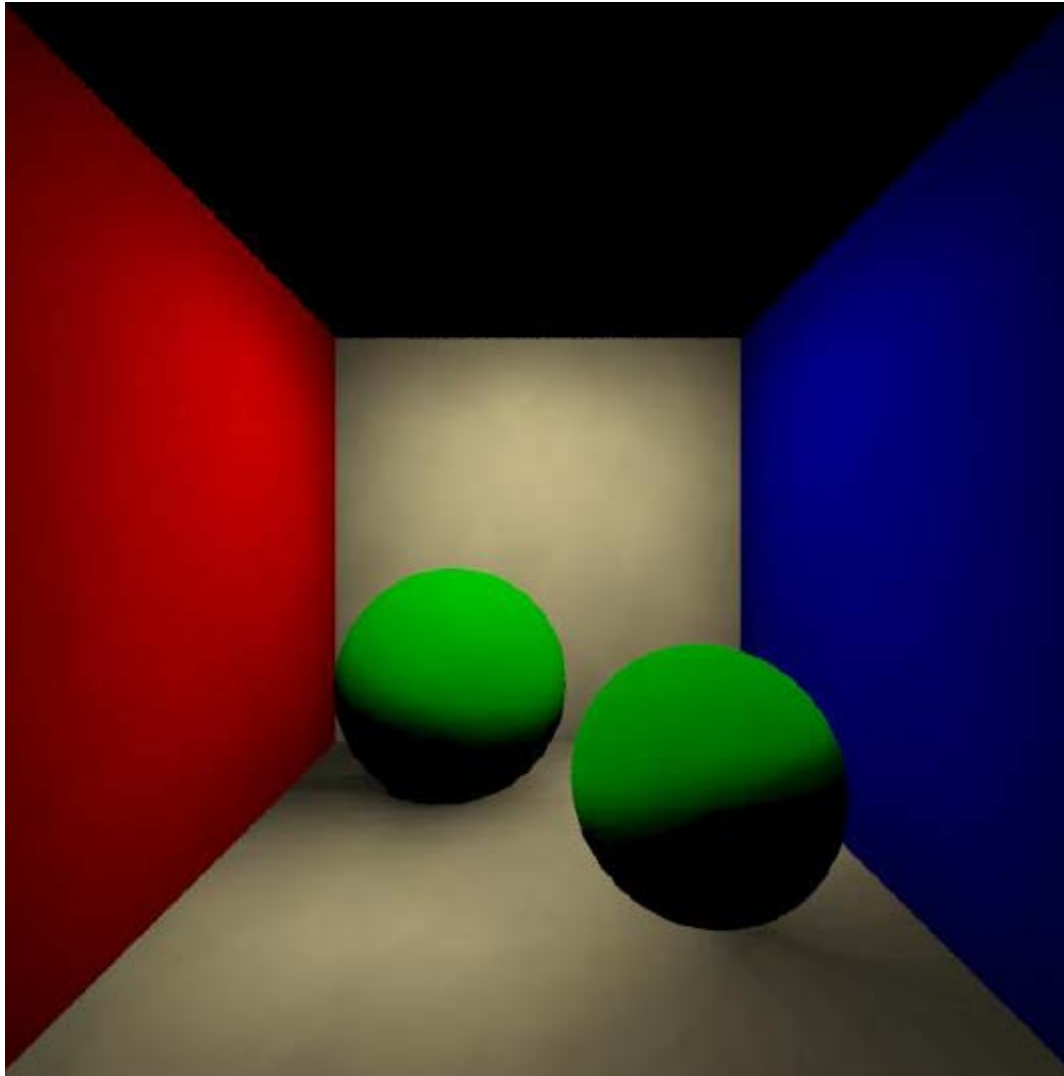
Tomáš Šváb

tomas.svab@gmail.com

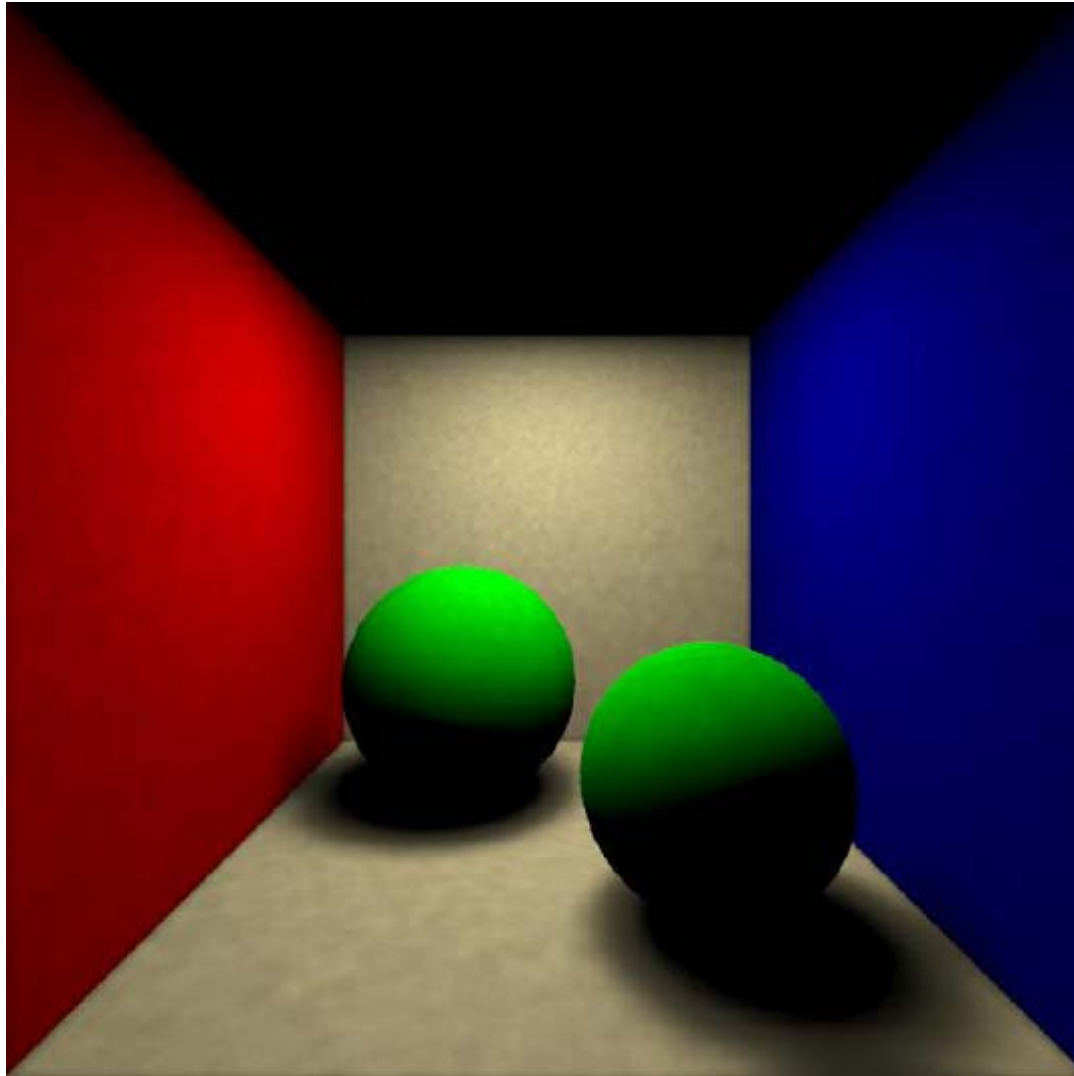
Nevyvážená PM



Přímá vizualizace (PM)



Přímá vizualizace (PM)



Výhody PM

- výborný pro kaustiky
 - oddělená mapa kaustik (vysoká přesnost)
 - lze filtrovat
- s rozšířeními výborný i pro difuzní scény
 - oddělená globální mapa fotonů
 - finální sbírání
 - především irradiance caching + gradienty
 - two-pass rendering
- teoreticky konzistentní
- importance sampling dle PM

Výhody PM

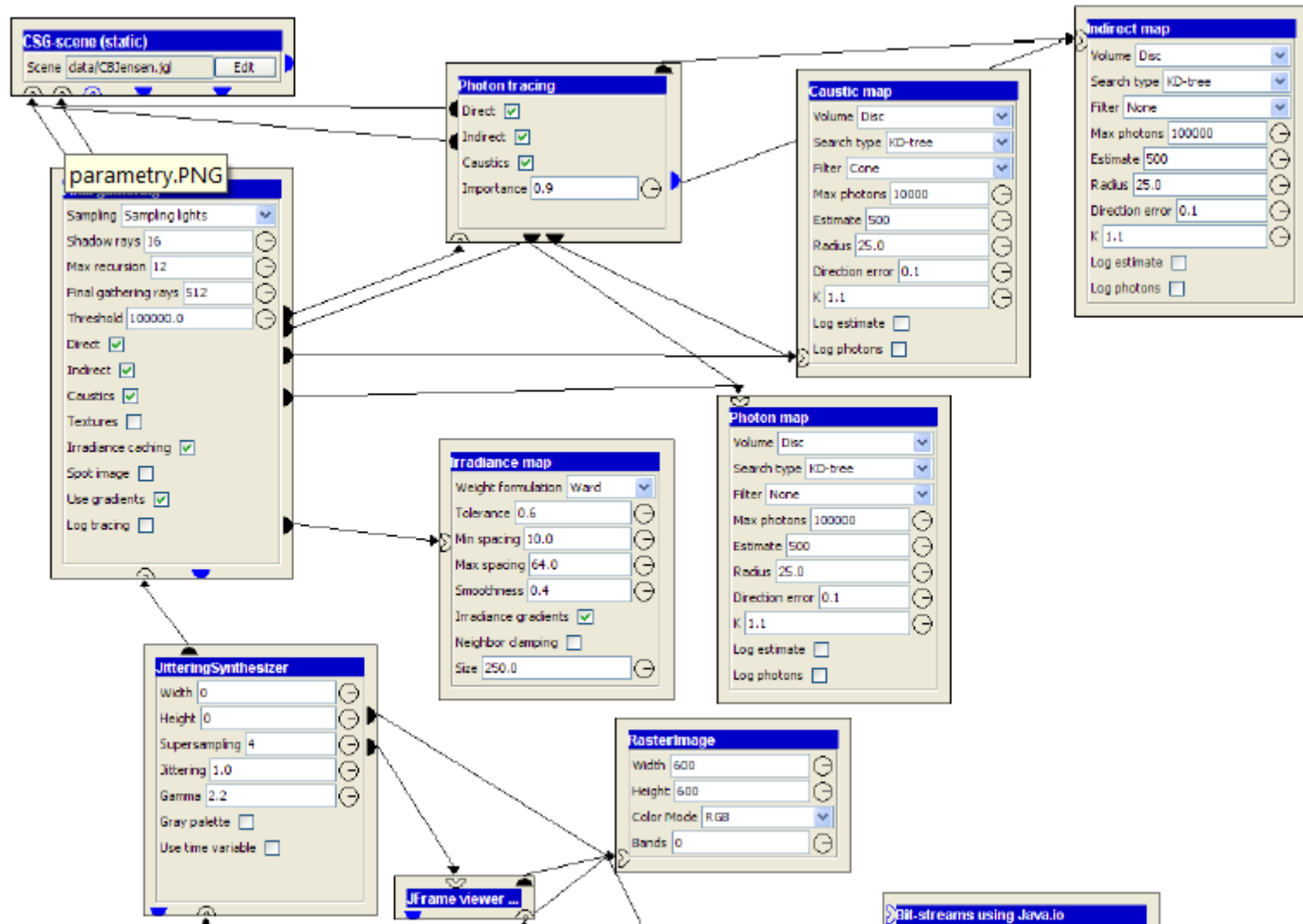
$$L(x, \vec{\omega}) = \lim_{N \rightarrow \infty} \sum_{p=1}^{\lfloor N^\beta \rfloor} \frac{f_r(x, \vec{\omega}, \vec{\omega}_p) \phi_p(x_p, \vec{\omega}_p)}{\pi r^2}$$

- podmínky konzistence
 - nekonečný počet fotonů v PM
 - nekonečný počet fotonů v odhadu
- splnění podmínek
 - N .. počet fotonů ve fotonové mapě
 - $N^\beta, \beta \in]0: 1[$.. počet fotonů v odhadu

Problémy PM – bias

- tolik nevadí (algoritmus je teoreticky konzistentní)
- v praxi ale konzistentní není
 - daný počet fotonů, daný rádius a další parametry
- problémy s nastavováním parametrů
 - těch je mnoho (rádius, počet fotonů ve fotonové mapě, počet fotonů v odhadu, počet paprsků pro FG, ..)
 - potřeba použít relativně „správné“ hodnoty
 - pro nesprávné nedává algoritmus věrohodné výsledky
 - nelze automatizovat nastavování parametrů (tak aby opravdu fungovalo)

Problémy PM – bias



Problémy PM – paměťová náročnost

- všechny fotony vystřelujeme v prvním průchodu
- někam je musíme uložit
- není problém uložit XY fotonů
- pro běžné scény XY fotonů opravdu stačí
- extrémní případ – složité kaustiky
 - ty jsou odhadovány přímo (žádná „berlička“ a la FG)
 - chceme „přesné“ kaustiky => potřebujeme patřičný počet fotonů
 - patřičný počet může být příliš velký

Problémy PM – neprogresivní

- fixní parametry
- fixní průchody
- přesnost výpočtu nastavena na začátku
 - zpřesnění není možné (závislé na rádiusu a počtu fotonů)
 - můžeme střílet další paprsky
 - anti-aliasing, motion blur, hloubka ostrosti (OK)
 - globální osvětlení, kaustiky (kvalitu počet paprsků neovlivní)

Problémy PM – lesklé plochy

- fotony z PM příliš nepřispívají
- vysoká variance při odhadu radiance
- výrazný lalok BRDF
 - sekundární paprsky dopadají na malou oblast
 - korelované výsledky odhadu radiance
 - odhaluje nepřesnosti fotonové mapy
- snaha o řešení
 - další krok finálního sbírání
 - prodloužení času výpočtu
 - zlepšení osvětlení záleží na scéně

Problémy PM – lesklé plochy (1/2)



Problémy PM – lesklé plochy (2/2)



Progressive photon mapping (1/3)

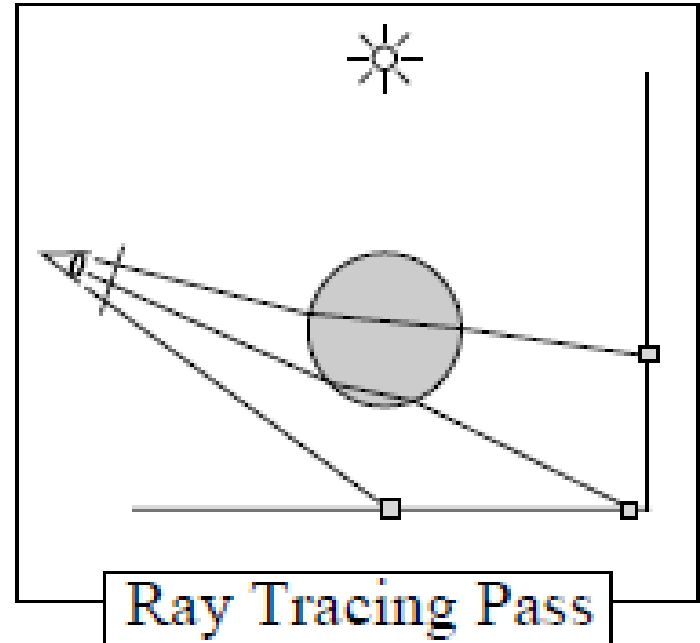
- reverzní fotonové mapy (předchůdce, odlišný cíl)
- vylepšení GI = více fotonů => prohodíme průchody
 - nejdříve ray-tracing (RT)
 - následně photon-tracing (PT)
- PT průchod můžeme opakovat
 - zpřesnění řešení po každém průchodu
- ray tracing
 - slouží k nalezení k viditelných ploch
 - paprsek sledován dokud nenarazí na ne-speculární objekt (omezení délky cesty např. ruskou ruletou)
 - pro každý bod obsahující ne-spekulární složku uložíme data

Progressive photon mapping (2/3)

- data

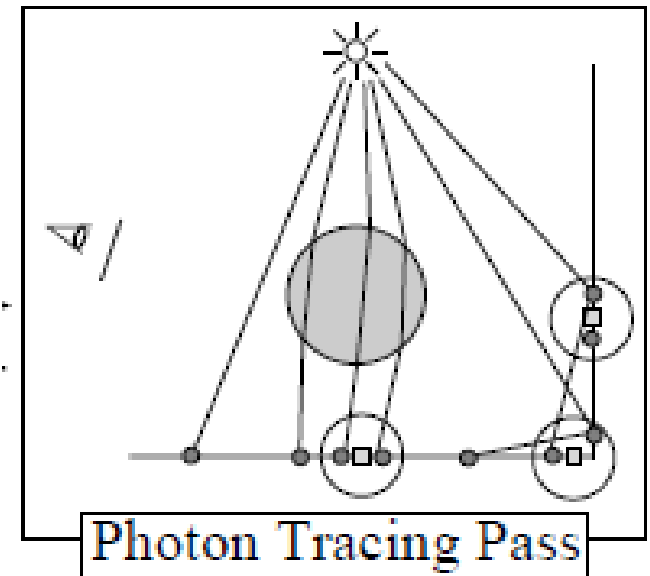
- pozice x
- normála
- směr
- BRDF či index BRDF
- lokace pixelu (x,y)
- váhový koeficient pixelu

- rádius
- počet akumulovaných fotonů
- akumulovaný flux



Progressive photon mapping (3/3)

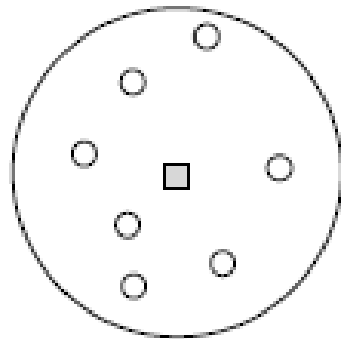
- photon tracing
 - libovolný počet průchodů (jsou na sobě nezávislé)
 - každý průchod zpřesňuje globální osvětlení (lze vykreslovat progresivně)
 - pracujeme s omezeným počtem fotonů
 - v každém průchodu
 - vystřelíme n fotonů
 - pro každý bod zjistíme jaké fotony k němu přispívají
 - ty použijeme pro upřesnění globálního osvětlení
 - poté můžeme fotony zahodit
 - opakujeme proces



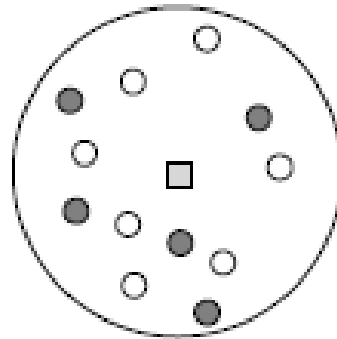
Odhad hustoty

- původní PM
 - odhad lokální hustoty fotonů $d(x) = \frac{n}{\pi r^2}$
 - vylepšení – kombinace několika PM
 - několik odhadů se stejným rádiusem a následné zprůměrování
 - přesnější odhad hustoty („jemnější“ výsledek)
 - nevylepšuje malé detaily!
 - není konzistentní (nekonverguje se zvyšujícím se počtem PM)
- progresivní PM
 - cíl: odhad hustoty fotonů, který konverguje
 - redukce radiusu
 - zvyšování hustoty akumulovaných fotonů

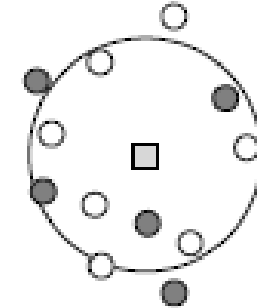
Redukce rádiusu (1/4)



Radius: $R(\mathbf{x})$
Photons: $N(\mathbf{x})$



Radius: $R(\mathbf{x})$
Photons: $N(\mathbf{x}) + M(\mathbf{x})$



Radius: $R(\mathbf{x}) - dR(\mathbf{x})$
Photons: $N(\mathbf{x}) + \alpha M(\mathbf{x})$

- máme hit-point
- při sledování fotonů akumulujeme flux
- po každém průchodu snížíme rádius dle příspěvku fotonů

Redukce rádiusu (2/4)

- nová hustota fotonů

$$\hat{d}(x) = \frac{N(x) + M(x)}{\pi R(x)^2}$$

- $N(x)$.. počet dosud akumulovaných fotonů
- $M(x)$.. počet nově akumulovaných fotonů
- $R(x)$.. současný rádius
- $\hat{d}(x)$.. nová hustota fotonů

Redukce rádiusu (3/4)

- redukovaný rádius

$$\hat{R}(x) = R(x) - dR(x)$$

- počet fotonů v redukovaném rádiusu

$$\hat{N}(x) = \pi \hat{R}(x)^2 \hat{d}(x)$$

$$\hat{N}(x) = \pi (R(x) - dR(x))^2 \hat{d}(x)$$

$$\underline{\hat{N}(x) = N(x) + \alpha M(x)}$$

po každé iteraci se musí zvýšit počet fotonů v odhadu

$\alpha \in (0,1)$.. zlomek fotonů, které ponecháme

Redukce rádiusu (4/4)

- redukovaný rádius

$$\hat{N}(x) = \pi(R(x) - dR(x))^2 \hat{d}(x)$$

$$\underline{N(x) + \alpha M(x)} = \pi(R(x) - dR(x))^2 \frac{N(x) + M(x)}{\pi R(x)^2}$$

$$R(x) - dR(x) = R(x) \sqrt{\frac{N(x) + \alpha M(x)}{N(x) + M(x)}}$$

$$\hat{R}(x) = R(x) \sqrt{\frac{N(x) + \alpha M(x)}{N(x) + M(x)}}$$

Korekce fluxu (1/2)

- akumulovaný flux od $N(x)$ fotonů, které přispěli do daného hit-pointu x

$$\tau_N(x, \vec{\omega}) = \sum_{p=1}^{N(x)} f_r(x, \vec{\omega}, \vec{\omega}_p) \phi_p(x_p, \vec{\omega}_p)$$

- x .. pozice hit-pointu
- $\vec{\omega}$.. směr paprsku
- $\vec{\omega}_p$.. směr fotonu
- f_r .. BRDF
- ϕ_p .. nenormalizovaný flux fotonu p

Korekce fluxu (2/2)

- $M(x)$ nových fotonů, které přispívají

$$\tau_M(x, \vec{\omega}) = ..$$

- musíme udělat update v hit-pointu x
- pokud by se rádius neměnil

$$\tau_{\hat{N}}(x, \vec{\omega}) = \tau_N(x, \vec{\omega}) + \tau_M(x, \vec{\omega})$$

- rádius se mění
 - předpoklad: konstantní hustota fotonů

$$\tau_{\hat{N}}(x, \vec{\omega}) = (\tau_N(x, \vec{\omega}) + \tau_M(x, \vec{\omega})) \frac{\pi \hat{R}(x)^2}{\pi R(x)^2}$$

$$\tau_{\hat{N}}(x, \vec{\omega}) = \tau_N(x, \vec{\omega}) + \tau_M(x, \vec{\omega}) \frac{N(x) + \alpha M(x)}{N(x) + M(x)}$$

Odhad radiance

- po libovolném photon tracingu můžeme renderovat
 - stačí odhadnout radianci v každém hit-pointu

$$L(x, \vec{\omega}) = \int_{\Omega} f_r(x, \vec{\omega}, \vec{\omega}') L(x, \vec{\omega}') (\vec{n} \cdot \vec{\omega}') d\omega'$$

$$L(x, \vec{\omega}) \approx \frac{1}{\Delta A} \sum_{p=1}^n f_r(x, \vec{\omega}, \vec{\omega}_p) \Delta\phi_p(x_p, \vec{\omega}_p)$$

$$L(x, \vec{\omega}) \approx \frac{\tau_i(x, \vec{\omega})}{N_e(i) \pi R_i(x)^2}$$

pro libovolné BRDF

Výsledky (1/5)

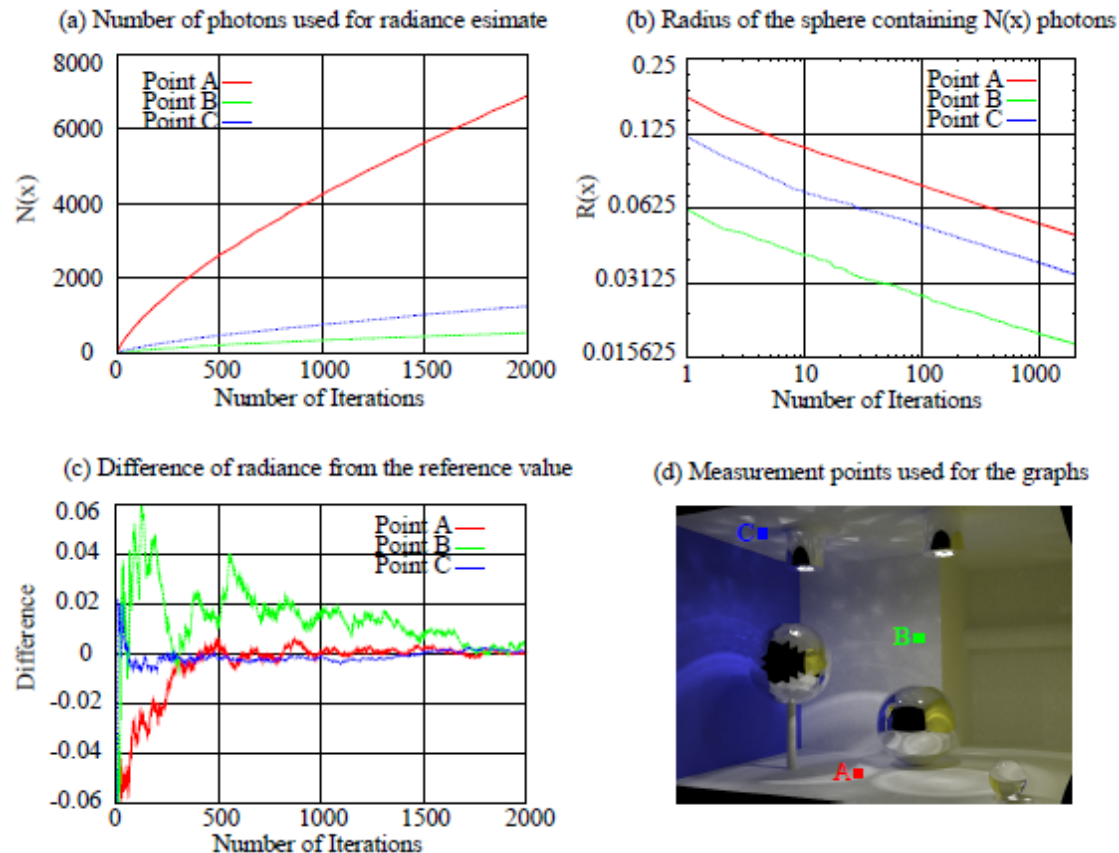


Figure 4: Statistics at the hit points as a function of the number of iterations. The three hit points A, B, and C are indicated in (d). Each iteration is using 100000 photons. Note that the graph of $R(x)$ uses log scale for both axes.

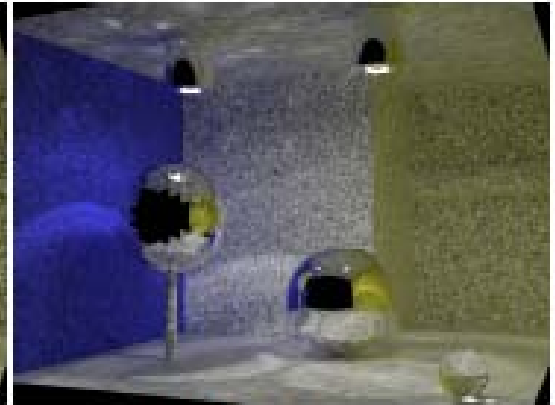
Výsledky (2/5)



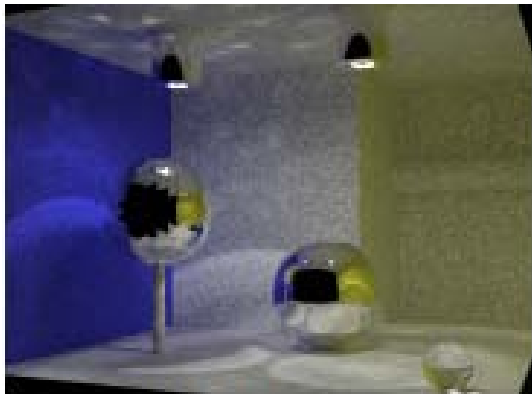
0.1M photons



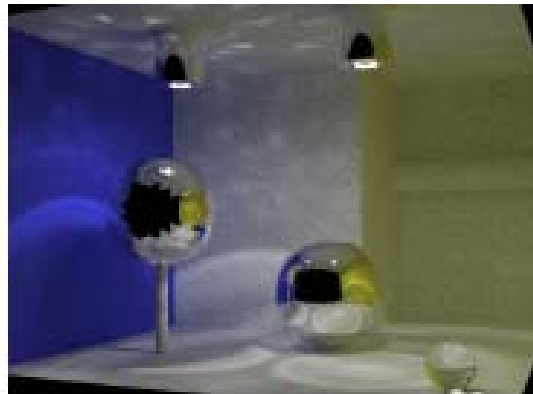
0.4M photons



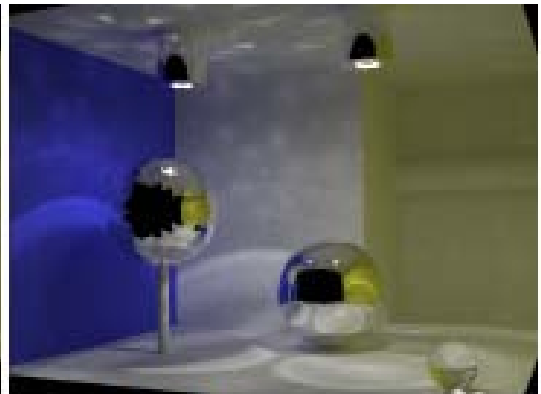
1.6M photons



6.4M photons



25.6M photons



102.4M photons

Výsledky (3/5)

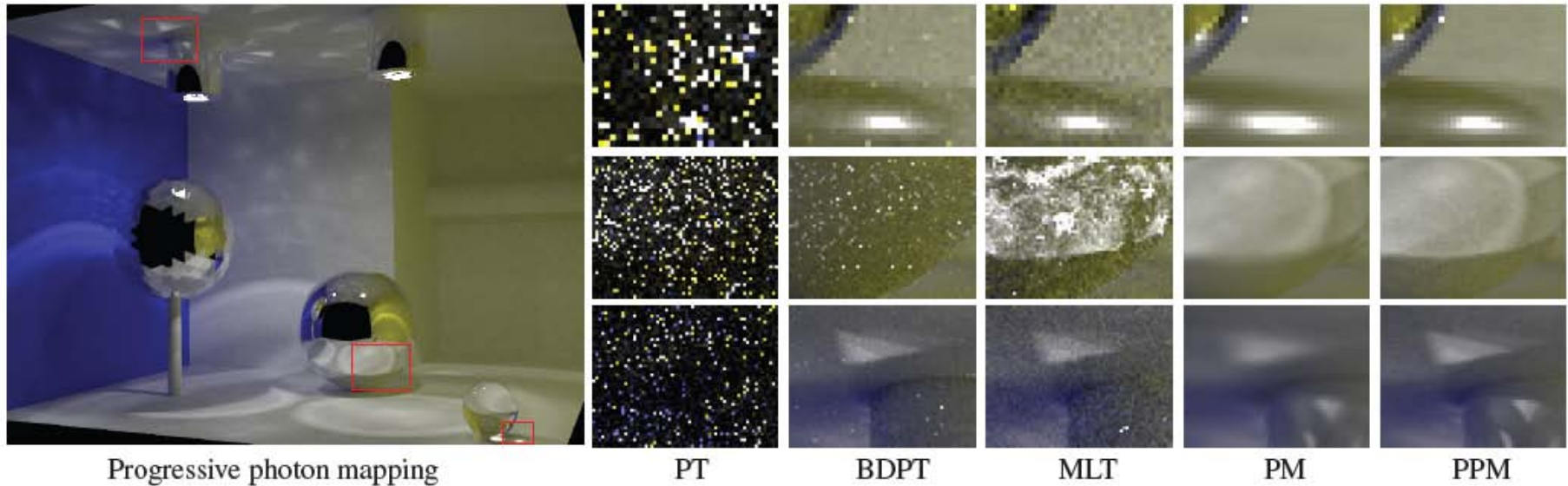


Figure 6: A box scene illuminated by a lighting fixture. The lighting fixture is behind glass and the illumination in the scene is dominated by caustics. The specular reflections and refractions have significant noise even with Metropolis light transport. Standard photon mapping cannot resolve the sharp illumination details in the scene with the maximum 20 million photons in the photon map. With progressive photon mapping we could use 213 million photons, which resolves all the details in the scene and provides a noise free image in the same rendering time as the Monte Carlo ray tracing methods.

Výsledky (4/5)

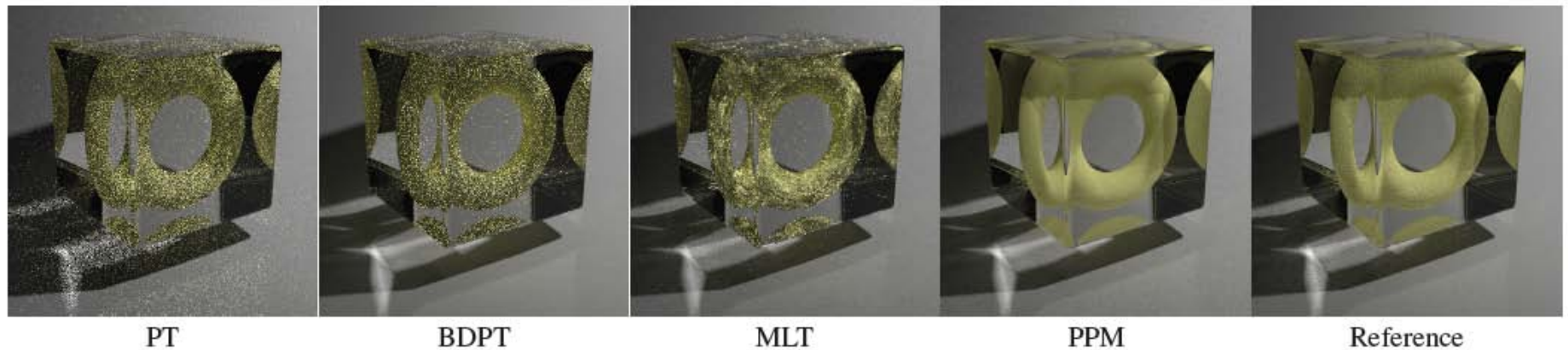


Figure 7: *Torus embedded in a glass cube. The reference image on the far right have been rendered using path tracing with 51500 samples per pixel. The Monte Carlo ray tracing methods fail to capture the lighting within the glass cube, while progressive photon mapping provides a smooth result using the same rendering time.*

Výsledky (5/5)

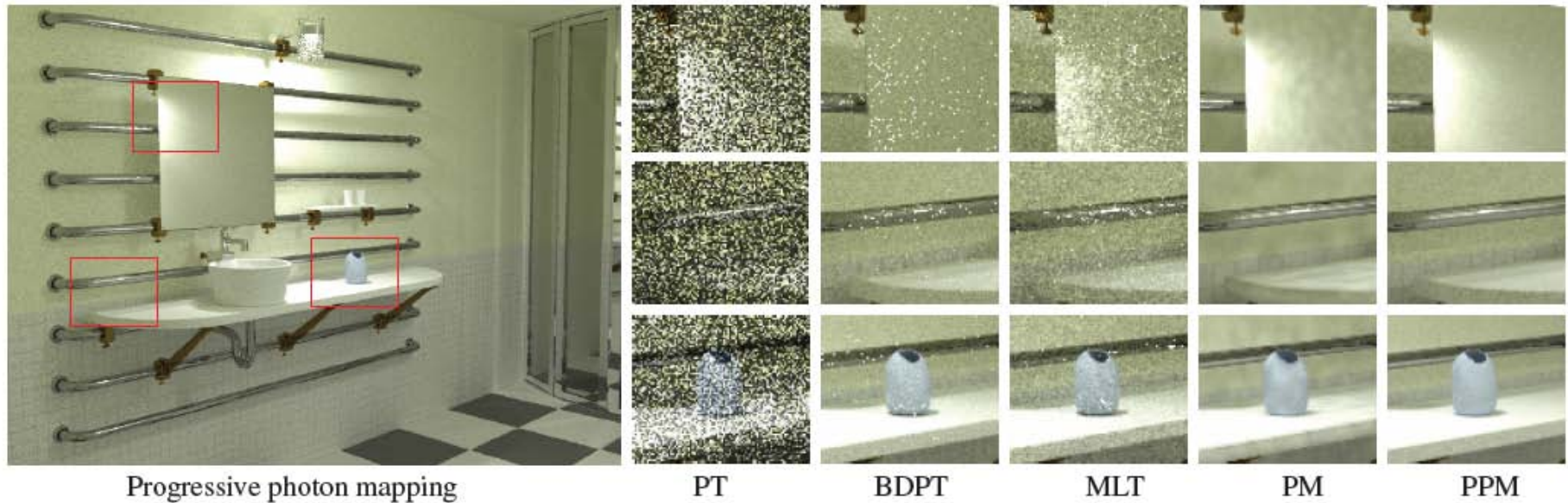


Figure 8: *Lighting simulation in a bathroom. The scene is illuminated by a small lighting fixture consisting of a light source embedded in glass. The illumination in the mirror cannot be resolved using Monte Carlo ray tracing. Photon mapping with 20 million photons results in a noisy and blurry image, while progressive photon mapping is able to resolve the details in the mirror and in the illumination without noise.*

Problémy PPM

- PPM je schopen počítat korektní radianci v bodě x
- některé efekty vyžadují korektní radianci v nějaké oblasti
 - motion blur
 - průměrná radiance pro daný časové úsek
 - hloubka ostrosti
 - průměrná radiance viditelná skrze čočku
- to ale PPM neumí => vznik SPPM
- dokonce i s glossy jsou problémy
 - příspěvek pouze od fotonů přímo ze zdroje

Stochastic Progressive PM (1/2)

- přidání nového průchodu
 - distribuovaný ray-tracing (DRT)
 - po každém sledování fotonů
- důsledky nového průchodu
 - hit-pointy již nejsou fixní
- základní myšlenka – sdílený statistiky v rámci oblasti
 - stochastický odhad radiance
 - sdílený rádius
 - sdílený akumulovaný flux

Stochastic Progressive PM (2/2)

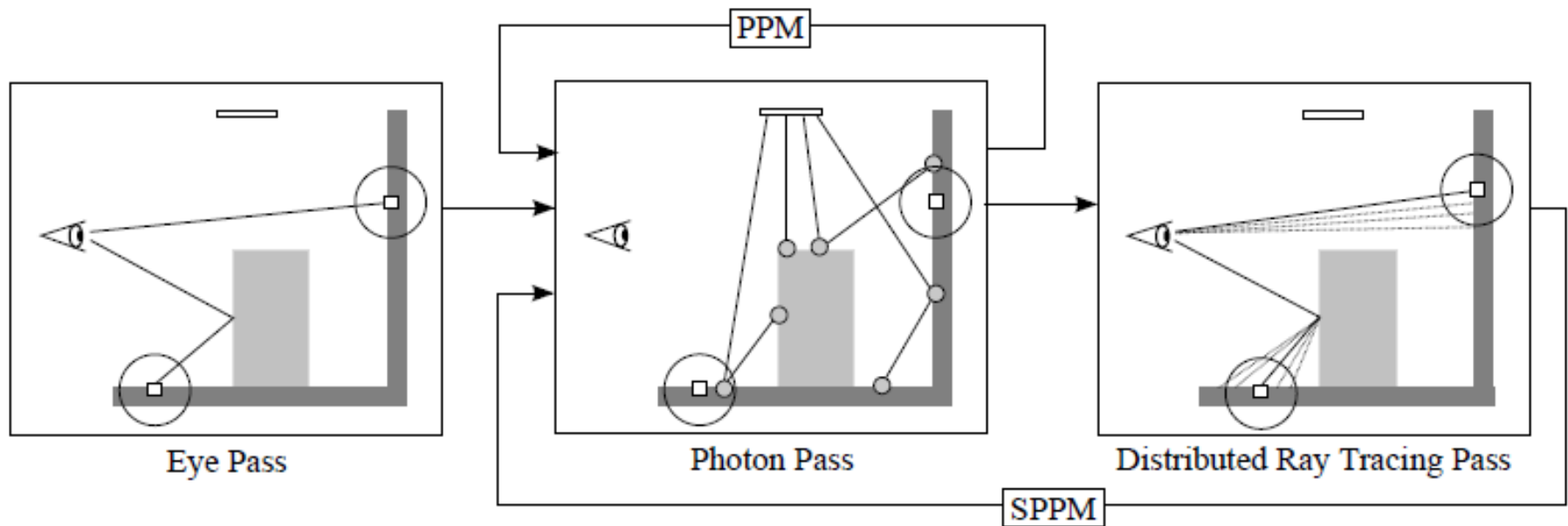


Figure 2: Difference between the algorithms of progressive photon mapping (PPM) and stochastic progressive photon mapping (SPPM). In order to compute the average radiance values, SPPM adds a new distributed ray tracing pass after each photon tracing pass. The photon tracing algorithm itself stays the same, but PPM uses a fixed set of hit points (shown as squares), whereas SPPM uses randomly generated hit points by the distributed ray tracing pass.

Od PPM k SPPM (1/2)

- odhad radiance v daném bodě

$$L(x, \vec{\omega}) \approx \frac{\tau_i(x, \vec{\omega})}{N_e(i)\pi R_i(x)^2}$$

- odhad radiance v dané oblasti (pixelu)

$$L(S, \vec{\omega}) \approx \frac{\tau_i(S, \vec{\omega})}{N_e(i)\pi R_i(S)^2}$$

- změna počtu fotonů

$$N_{i+1}(\vec{x}) = N_i(\vec{x}) + \alpha M_i(\vec{x})$$

- rozdíl: zobecněné veličiny

Od PPM k SPPM (2/2)

- update sdílených statistik (stejný jako u PPM)

$$N_{i+1}(S) = N_i(S) + \alpha M_i(x_i)$$

$$\underline{R_{i+1}(S)} = R_i(S) \sqrt{\frac{N_i(S) + \alpha M_i(x_i)}{N_i(S) + M_i(x_i)}}$$

$$\underline{\phi_i(x_i, \vec{\omega})} = \sum_{p=1}^{M_i(\vec{x}_i)} f_r(\vec{x}_i, \vec{\omega}, \vec{\omega}_p) \phi_p(x_p, \vec{\omega}_p)$$

$$\tau_{\hat{N}}(S, \vec{\omega}) = (\tau_N(S, \vec{\omega}) + \phi_i(x_i, \vec{\omega})) \frac{\pi \hat{R}(S)}{\pi R(S)^2}$$

- x_i .. náhodně generovaný bod z S

SRE – oblast S (1/8)

- stochastic radiance estimate
- rozšíření původního estimátoru na oblast S

$$L(S, \vec{\omega}) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n L(\vec{x}_k, \vec{\omega})$$

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim_{i \rightarrow \infty} \frac{\tau_i(\vec{x}_k, \vec{\omega})}{N_e(i) \pi R_i(\vec{x}_k)^2}$$

- je nutné udržovat statistiky pro všechny vzorky radiance!
- n by mělo být nekonečné => potřeba nekonečno paměti
- cíl: korektní průměrná radiance bez nutnosti ukládat nekonečno vzorků radiance

SRE – nezávislost rádiusu (2/8)

- cíl: odstranění závislosti rádiusu na hitpointu x
- ověření nezávislosti
 - předpoklady – rádius R_0 a α jsou konstantní pro dané S

$$\hat{R}(\vec{x}) = R(\vec{x}) \sqrt{\frac{N_i(\vec{x}) + \alpha M_i(\vec{x})}{N_i(\vec{x}) + M_i(\vec{x})}}$$

$$N_i(\vec{x}) = C_N L(\vec{x}) R(\vec{x})^2$$

$$M_i(\vec{x}) = C_M L(\vec{x}) R(\vec{x})^2$$

$$\hat{R}(\vec{x}) = R(\vec{x}) C_p$$

- C_N, C_M, C_p .. konstanty nezávislé na x

SRE – nezávislost rádiusu (3/8)

- redukce rádiusu je nezávislá na $x \Rightarrow$ rádius je také nezávislý
 - pokud R_0 je konstantní
- proč zrovna $N_i(\vec{x}) = C_N L(\vec{x}) R(\vec{x})^2$
 - počet fotonů závisí na
 - rádiusu v kterém hledáme
 - radianci v daném bodě

SRE – nezávislost rádiusu (4/8)

- důsledek
 - částečné odstranění závislosti rádiusu na x
 - použití jediné hodnoty rádiusu

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim_{i \rightarrow \infty} \frac{\tau_i(\vec{x}_k, \vec{\omega})}{N_e(i) \pi R_i(\vec{x}_k)^2}$$

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim_{i \rightarrow \infty} \frac{\tau_i(\vec{x}_k, \vec{\omega})}{N_e(i) \pi R_i(\vec{x}_0)^2}$$

SRE – sdílený rádius (5/8)

- další odstranění závislosti rádiusu na x
- použití sdíleného rádiusu

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n L(x_k, \vec{\omega})$$

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim_{i \rightarrow \infty} \frac{\tau_{R(S),i}(\vec{x}_k, \vec{\omega})}{N_e(i) \pi R_i(S)^2}$$

– $\tau_{R(S),i}$.. akumulovaný flux

$$\tau_{R(S),i}(\vec{x}_k, \vec{\omega}) = \frac{R_i(S)^2}{R_i(x_0)^2} \tau_i(\vec{x}_k, \vec{\omega})$$

SRE – sdílený rádius (6/8)

- odvození

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim_{i \rightarrow \infty} \frac{\tau_{R(S),i}(\vec{x}_k, \vec{\omega})}{N_e(i) \pi R_i(S)^2}$$

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim_{i \rightarrow \infty} \frac{R_i(S)^2}{R_i(x_0)^2} \frac{\tau_i(\vec{x}_k, \vec{\omega})}{N_e(i) \pi R_i(S)^2}$$

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim_{i \rightarrow \infty} \frac{R_i(S)^2}{R_i(x_0)^2} \frac{R_i(x_0)^2}{R_i(S)^2} \frac{\tau_i(\vec{x}_k, \vec{\omega})}{N_e(i) \pi R_i(S)^2}$$

- podmínka: podíl rádiusů se limitně blíží k nenulové konstantě („stejně rychlá“ redukce)

$$L(S, \vec{\omega}) = \frac{1}{n} \sum_{k=1}^n \lim L(x_k, \vec{\omega})$$

SRE – sdílený flux (7/8)

- problém
 - momentálně nutné ukládat nekonečné množství příspěvků
 - aproximace pomocí konečného množství příspěvků (to není konzistentní!)
- řešení
 - sdílený akumulovaný flux
 - jediná proměnná obsahující akumulovaný flux

SRE – sdílený flux (8/8)

- flux -> sdílený flux

$$\tau_i(\vec{x}_k, \vec{\omega}) \rightarrow \tau_i(S, \vec{\omega})$$

$$L(S, \vec{\omega})' = \lim_{i \rightarrow \infty} \frac{\tau_i(S, \vec{\omega})}{N_e(i) \pi R_i(S)^2}$$

- důkaz platnosti (viz. paper)

$$L(S, \vec{\omega}) = L(S, \vec{\omega})'$$

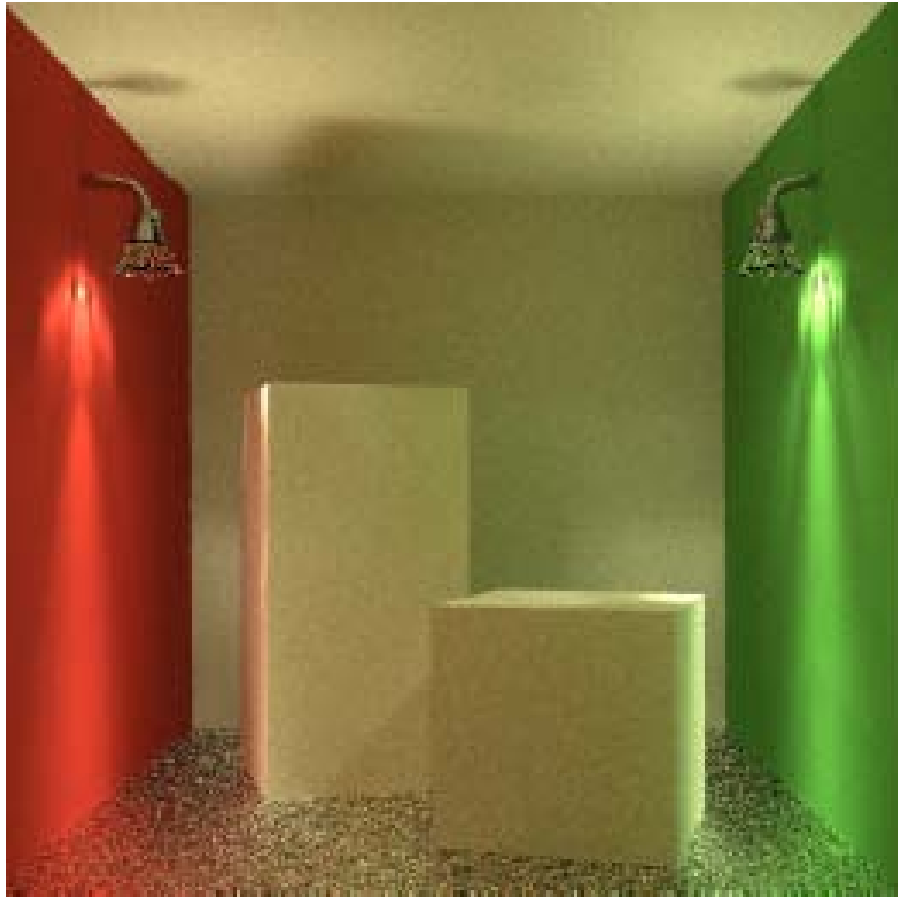
- platí

$$L(S, \vec{\omega}) = L(S, \vec{\omega})' = \lim_{i \rightarrow \infty} \frac{\tau_i(S, \vec{\omega})}{N_e(i) \pi R_i(S)^2}$$

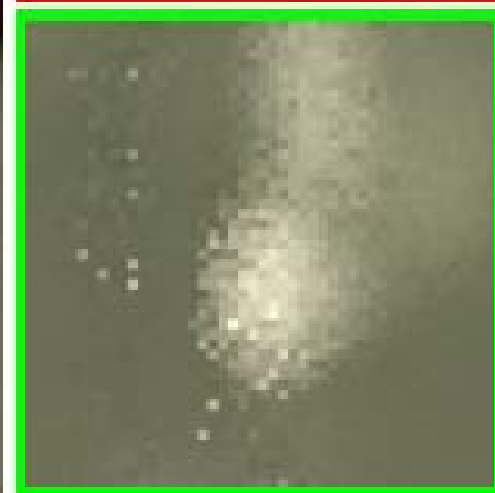
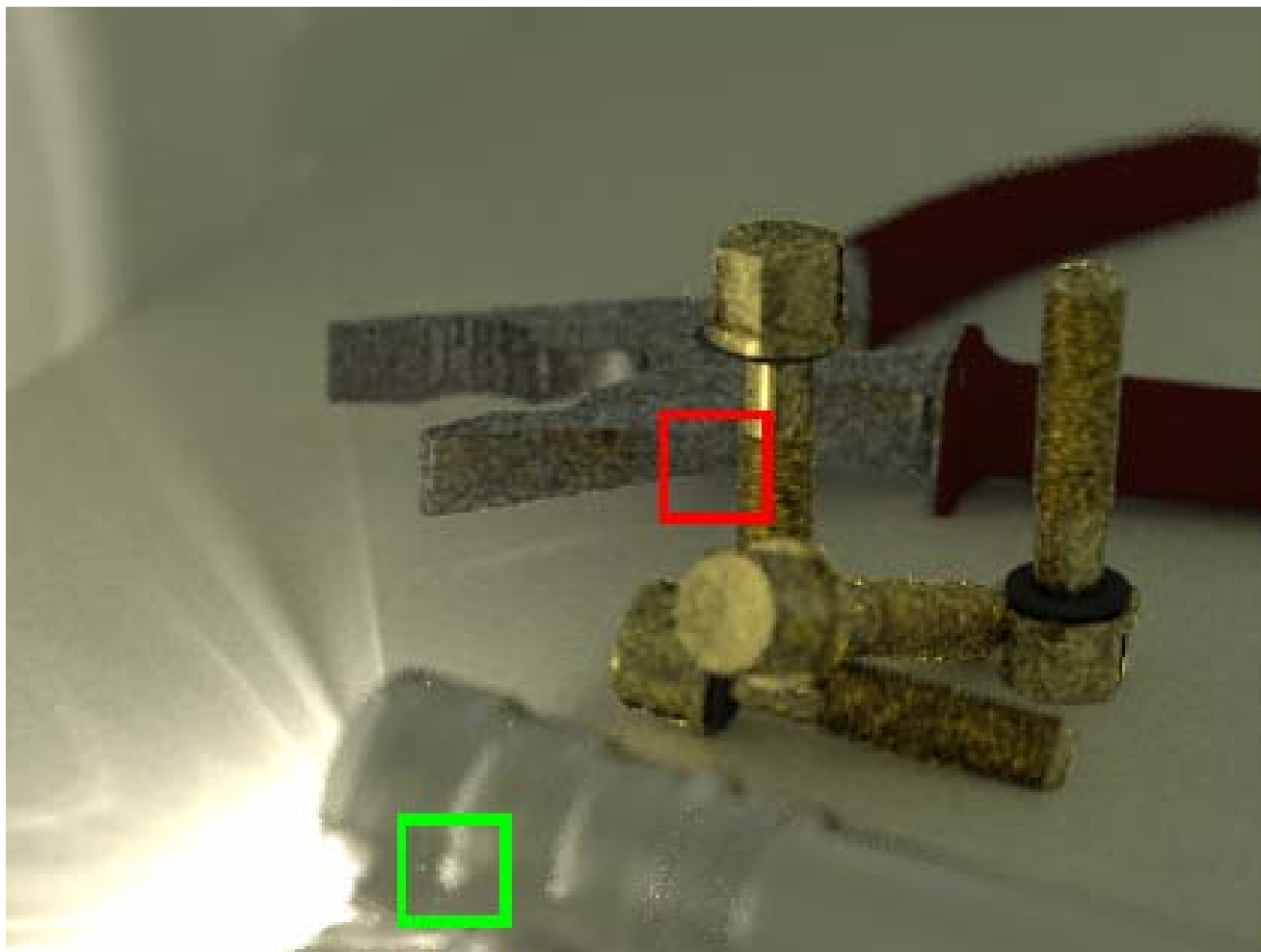
Výsledky (1/9)



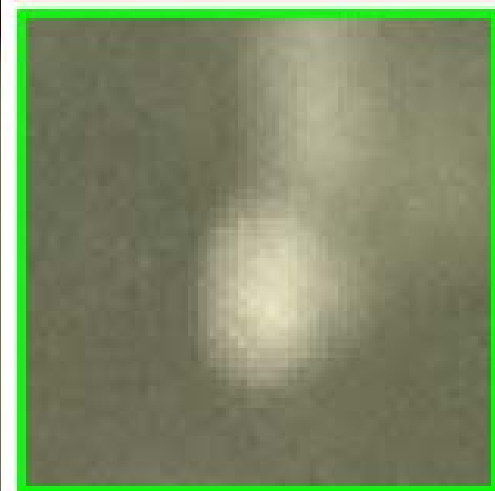
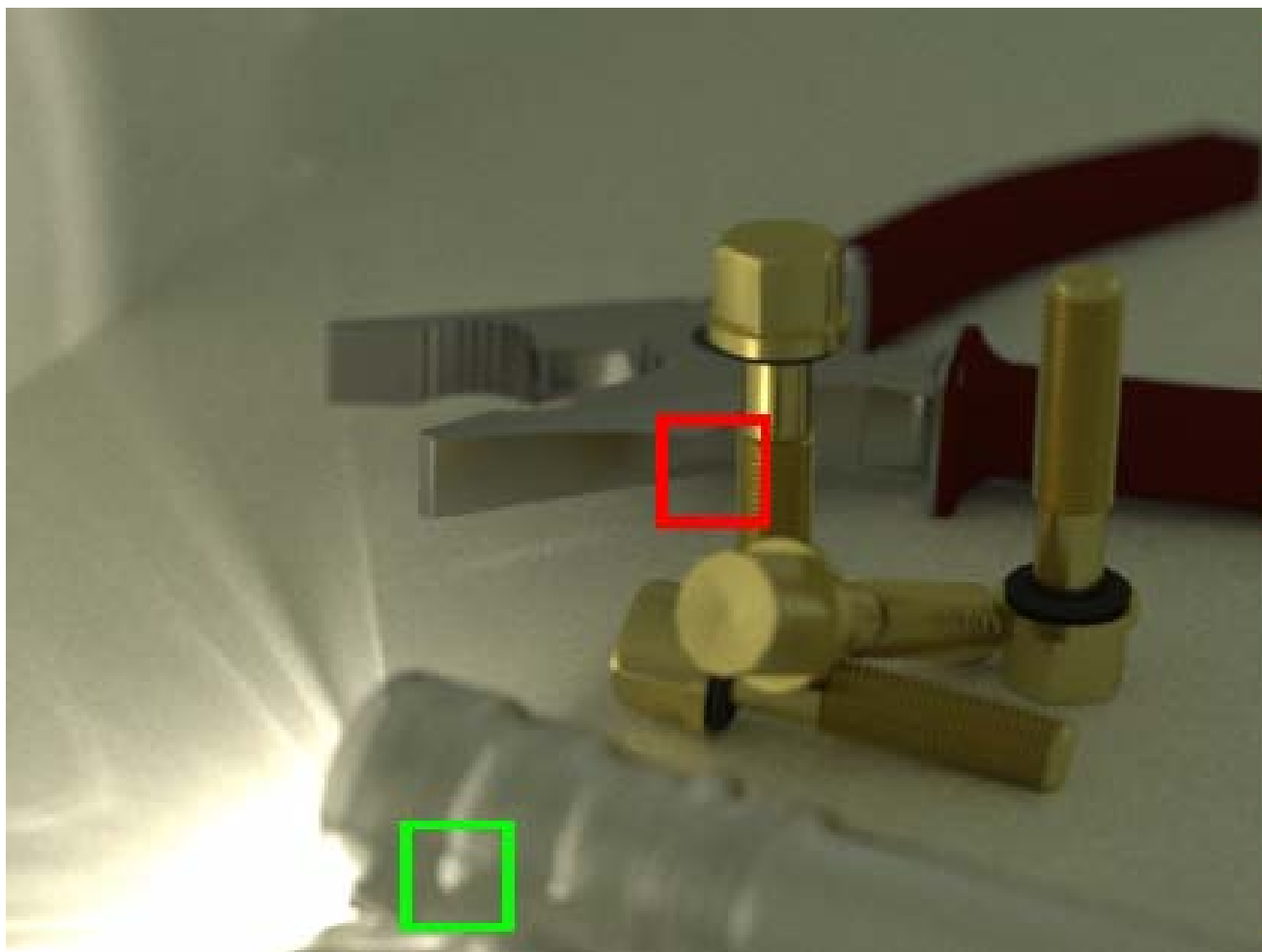
Výsledky (2/9)



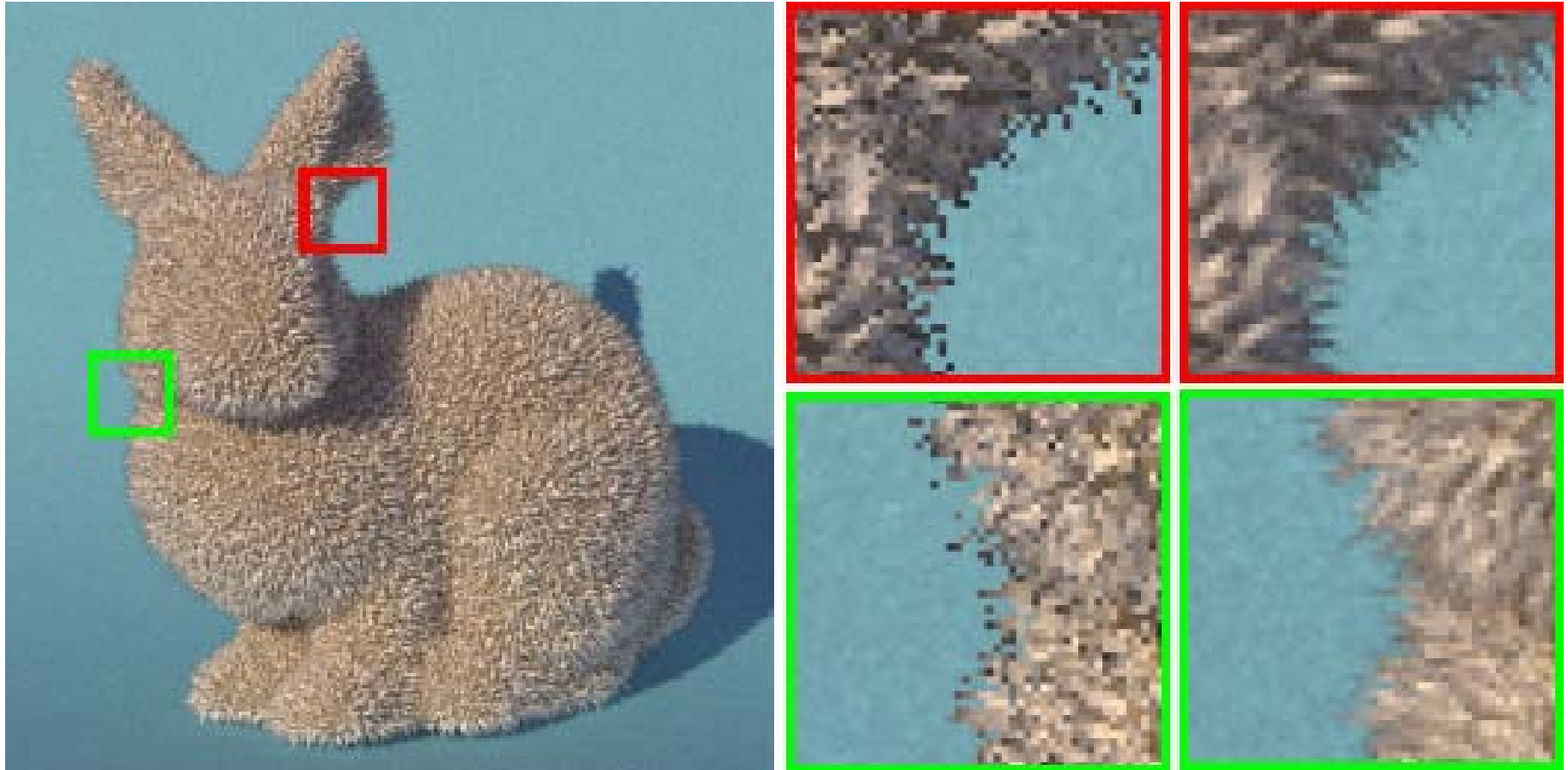
Výsledky (3/9)



Výsledky (4/9)

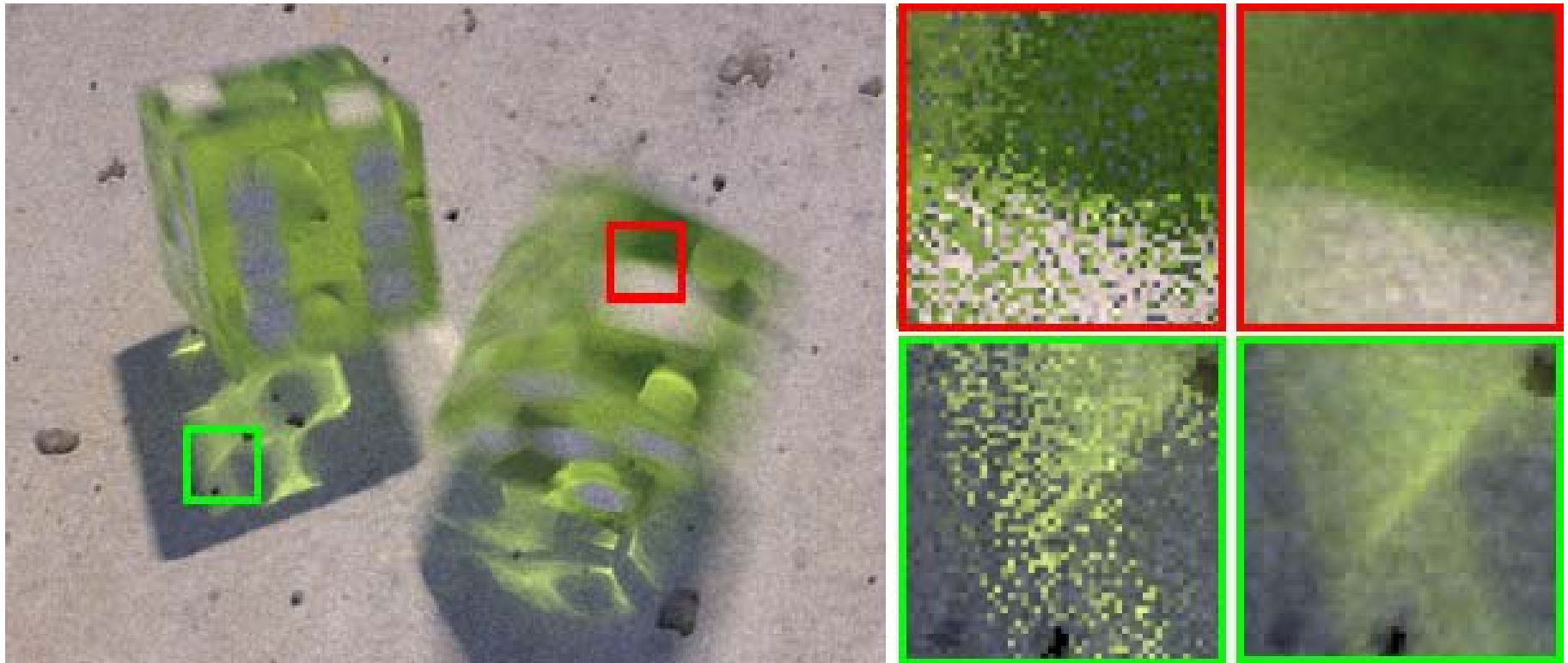


Výsledky (5/9)



Furry bunny illuminated by the skylight

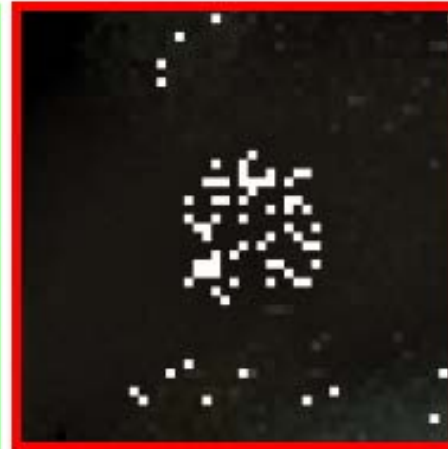
Výsledky (6/9)



The dices are illuminated by the sunlight,
and blur of caustics and shadows is due to motion blur of the dices.

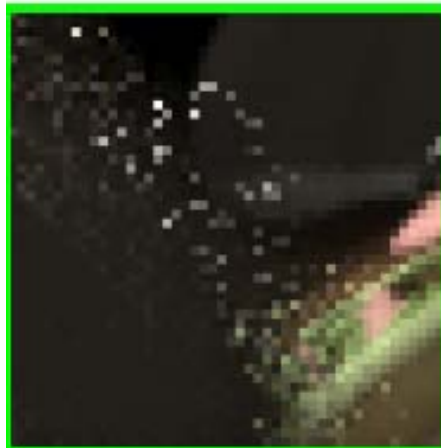
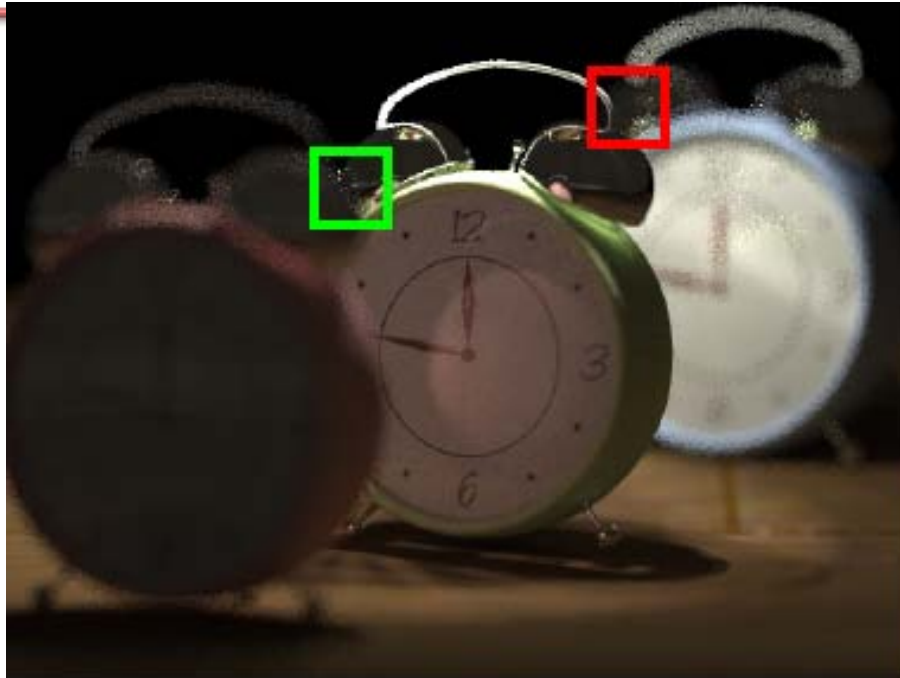
Výsledky (7/9)

- illuminated by a desk lamp with light bulb
- depth-of-field
- same rendering time
- lens for depth-of-field and caustics from the desk light => scene dominated by SDS paths
- BDPT used 11406 paths per pixel
- dark image
- noisy bright pixels.



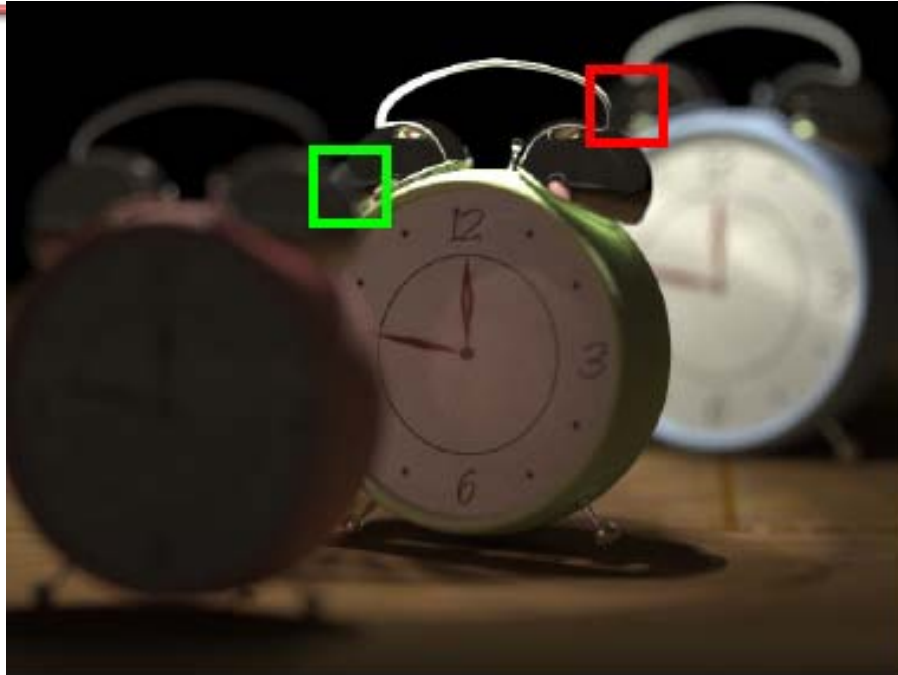
Výsledky (8/9)

PPM can handle such an illumination setting, but the close-ups show rendering of depth-of-field causes visually noticeable noise due to the fixed number of samples per pixels.



Výsledky (9/9)

SPPM robustly handles illumination by the desk lamp as well as depth-of-field in the same rendering time with less memory consumption



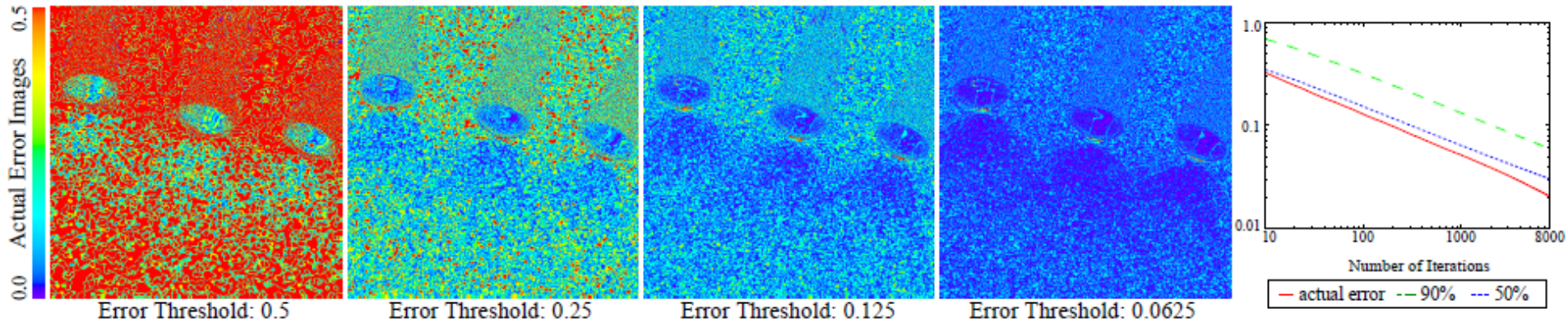
Shrnutí SPPM

- další zobecnění PPM
- nezajímá nás radiance v bodech, ale v oblasti (typicky pixel)
- rychleji konverguje (záleží na scéně)
- zabere méně paměti
- ...
- co zbývá udělat?
 - odhad chyby
 - adaptivní SPPM?
 - případné rozšíření pro volume rendering

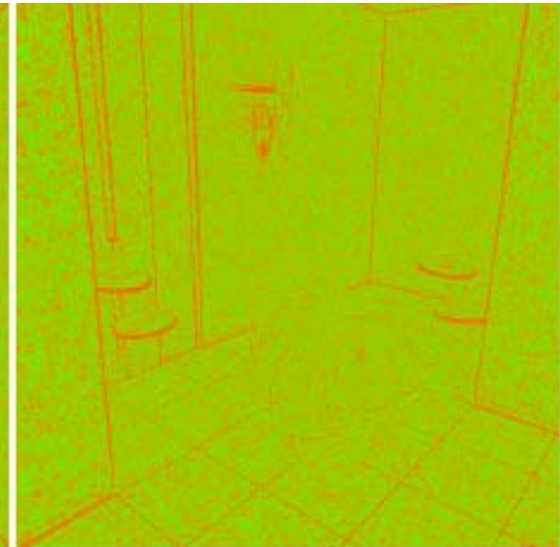
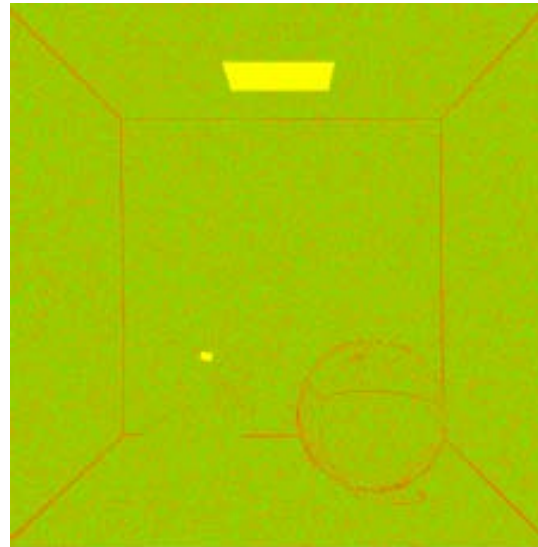
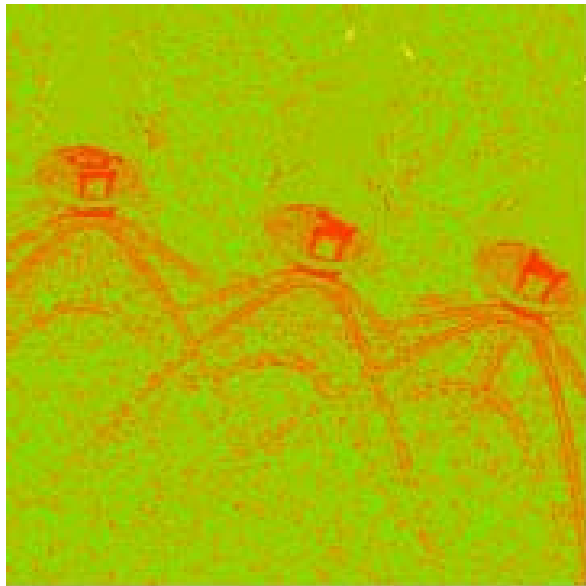
Error estimation framework (1/3)

- odhad chyby výsledného obrázku
 - unbiased metody
 - chyba = variance
 - různé techniky
 - biased metody
 - chyba = variance + bias
 - metody šité na daný algoritmus
- využití
 - analýza chyby
 - automatické zastavení výpočtu

Error estimation framework (2/3)



Error estimation framework (3/3)



Noise/Bias Ratio

The ratio of the estimated bias and the estimated noise bound in error estimation with 15M emitted photons. The red pixel indicates the bias is dominant and the green pixel indicates the noise is dominant in the estimated error. The result for the flashlights scene is in Figure 1.

Literatura

- Toshiya Hachisuka: ["Progressive Photon Mapping,"](#)
- Toshiya Hachisuka: Stochastic Progressive Photon Mapping
- Toshiya Hachisuka: A Progressive Error Estimation Framework for Photon Density Estimation

Děkuji za pozornost!