
Počítačová grafika III – Photon mapping

Jaroslav Křivánek, MFF UK

Jaroslav.Krivanek@mff.cuni.cz

Obousměrné sledování cest - opakování

Transport světla jako integrál

- **Cíl:** místo integrální rovnice chceme formulovat transport světla jako integrál přes cesty:

Příspěvek cesty x k hodnotě pixelu
(„contribution function“)

Míra na množině
světelných cest

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

Hodnota (“měření“)
 j -tého pixelu

Prostor všech světelných cest
Spojujících zdroj světla s pixelem j

Aplikace integrálu přes cesty

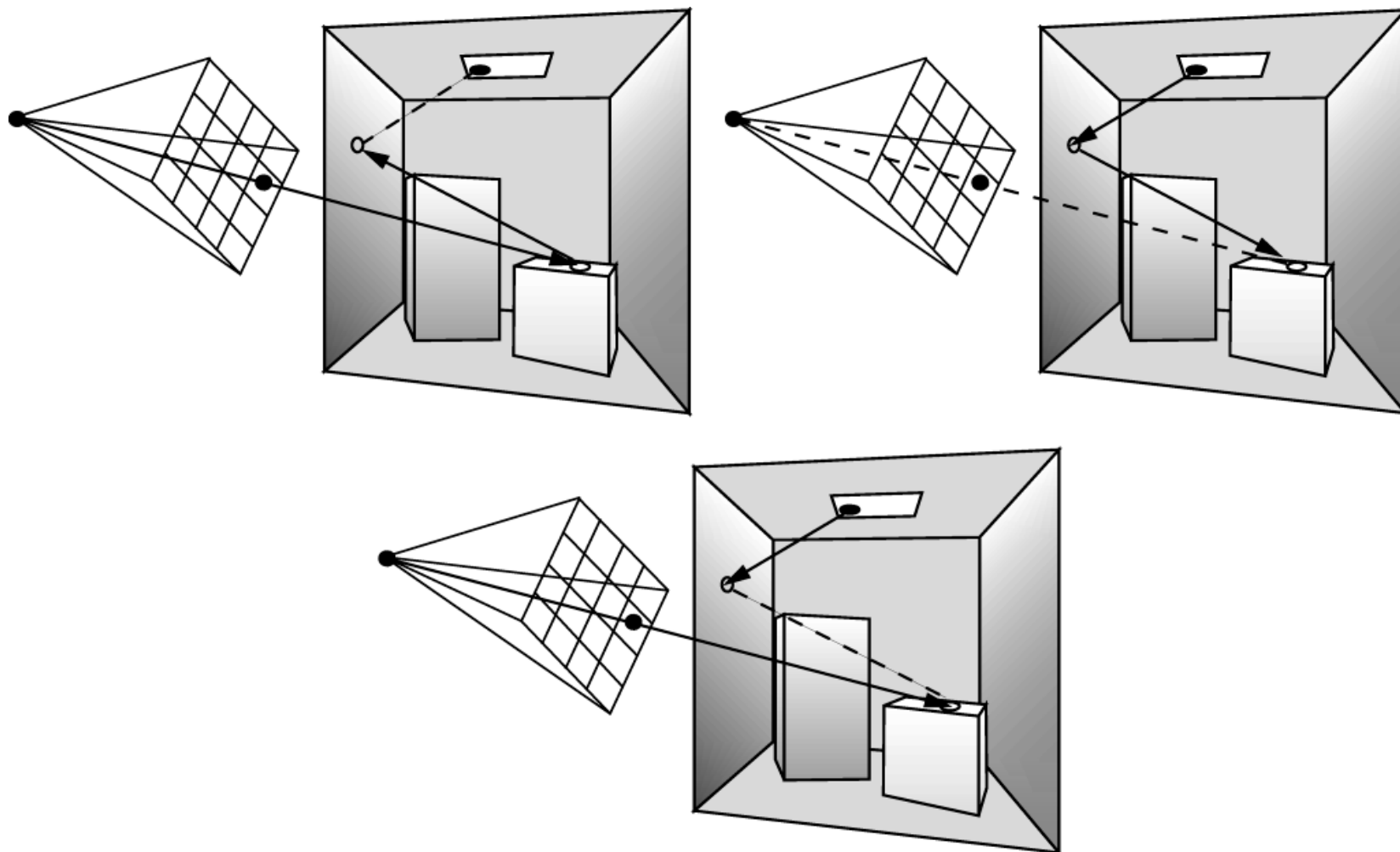
$$I_j = \int_{\Omega} f_j(\bar{x}) d\mu(\bar{x})$$

Odhad integrálu pomocí klasických Monte Carlo metod:

$$I_j \approx \frac{f_j(\bar{X})}{p(\bar{X})}$$

Hustota pravděpodobnosti cesty závisí na způsobu generování (od světla vs. od kamery)

Vzorkovací strategie

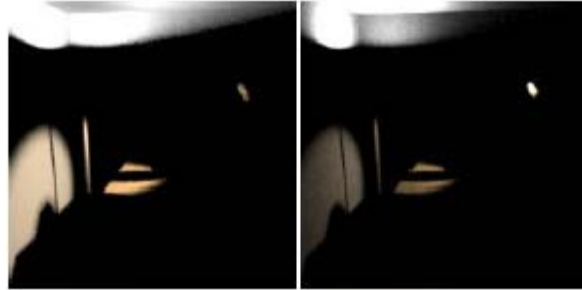


Kombinace vzorkovacích strategií

- Kombinovaný estimátor (MIS)

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

kombinační strategie
(např. vyvážená heuristika)



$k = 2$
(2x)



$k = 3$
(4x)



$k = 4$
(8x)



$k = 5$
(16x)

$s = 1$

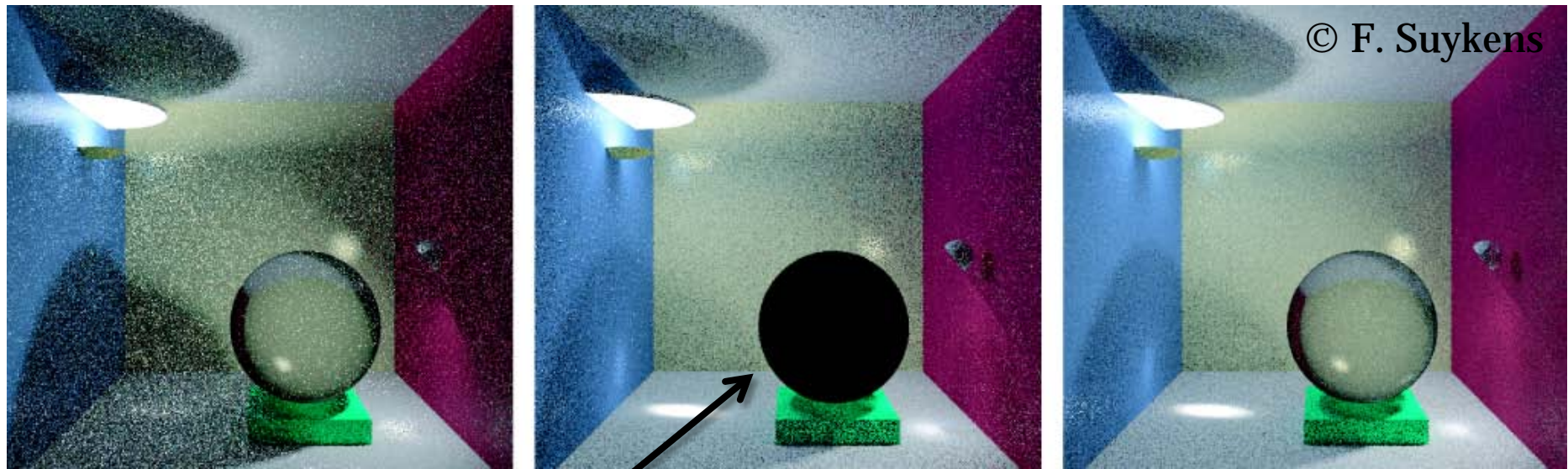
$s = 2 \dots$

$t = 2$

$t = 1$

$s / t =$ počet vrcholů na podcestě od světla / kamery

Porovnání algoritmů



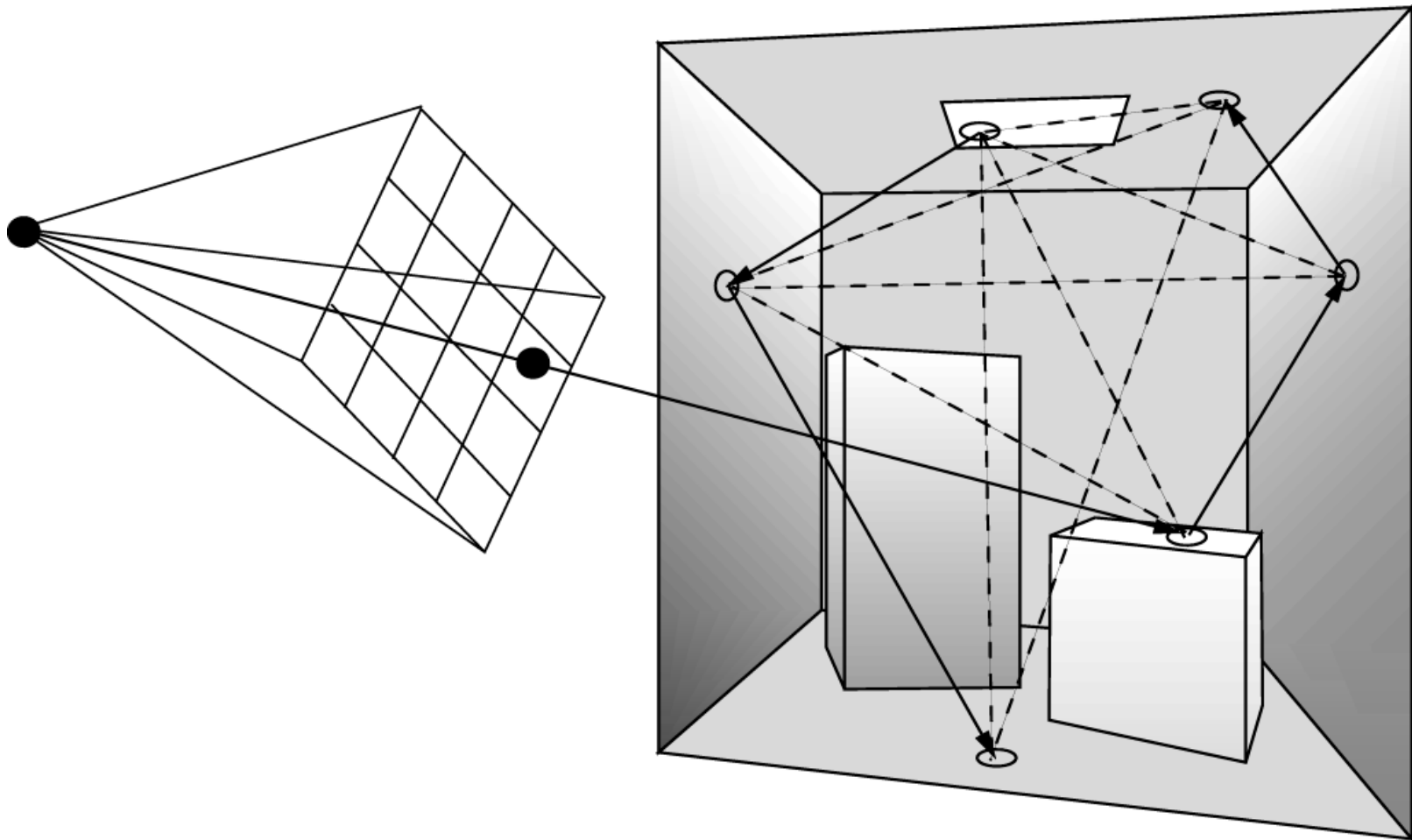
Path tracing

Light tracing

Bidirectional path tracing

Kvíz: Proč je skleněná koule černá?

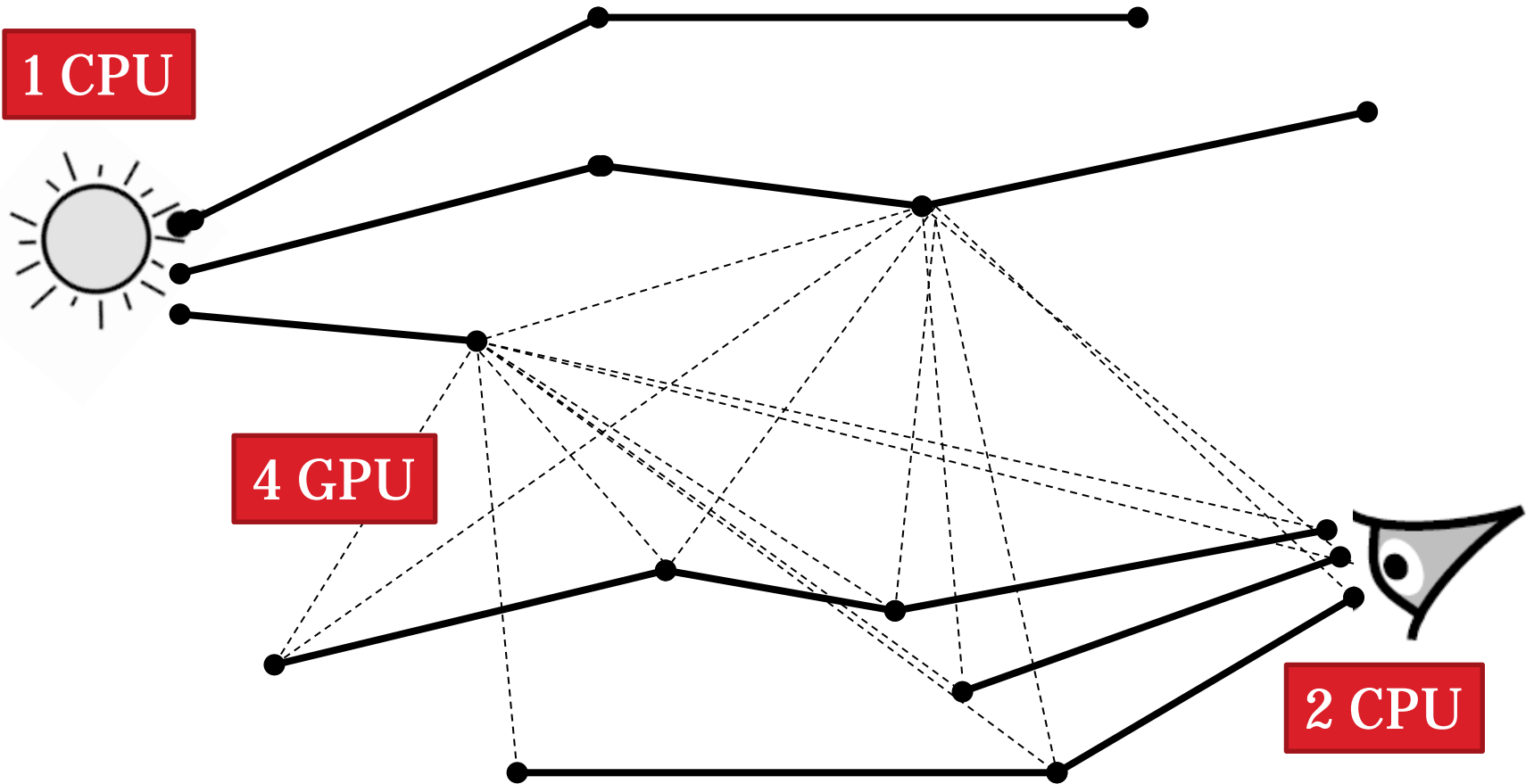
Implementace BPT: Generování cest po skupinách



Jiná implementace BPT

- Využití výpočetní síly GPU pro spojování (kompletování) cest [*Pajot et al. Eurographics 2011*]
- 1. Vygeneruj N prefixů cest ze světla (CPU)
- 2. Vygeneruj N sufixů cest z kamery (CPU)
- 3. Nahraj oboje na GPU
- 4. Spoj každý vrchol prefixů s každým vrcholem sufixů cest (GPU)

Jiná implementace BPT



Jiná implementace BPT



50 sec. (nová implementace, CPU+GPU)
11 min. (původní implementace, CPU)



1 hod.
13 hod.

Fotonové mapy

Fotonové mapy



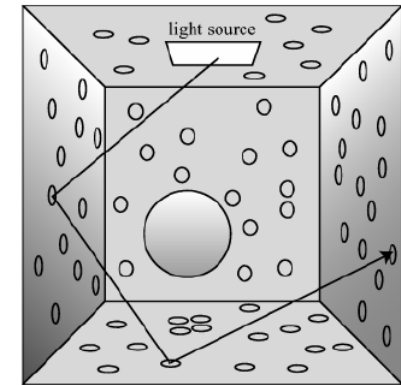
Fotonové mapy

- **Cesty se sledují od zdrojů světla i od kamery**
 - Světla jsou zdroji radiance
 - Kamera reprezentuje důležitost (potenciál)
- Podobné obousměrnému sledování cest
 - Generování cest ze světla a z kamery jsou však dva oddělené procesy
- **Přepoužití** stejných světelných cest pro všechny pixely
 - Fotonová mapa = „**cache cest od světla**“
- Při stejné kvalitě je obvykle mnohem rychlejší než M-C techniky
- **Ztráta nestrannosti !**
 - ale konzistentní (konverguje při zvětšování počtu fotonů)

Fáze výpočtu

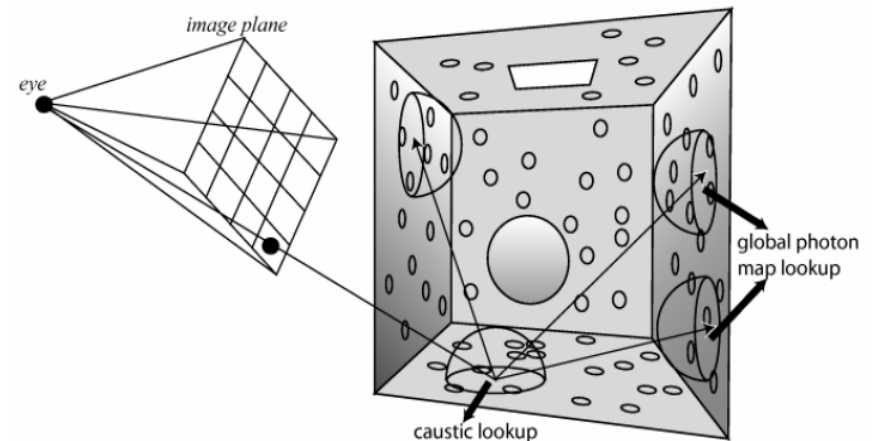
1. Rozmístění fotonů

- ❑ „Fotony“ jsou emitovány světelnými zdroji
- ❑ Propagují se do scény (a la light tracing)
- ❑ Ukládají se do „fotonových map“

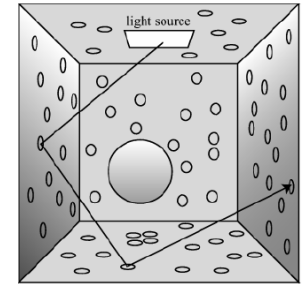


2. Rendering s využitím fotonových map

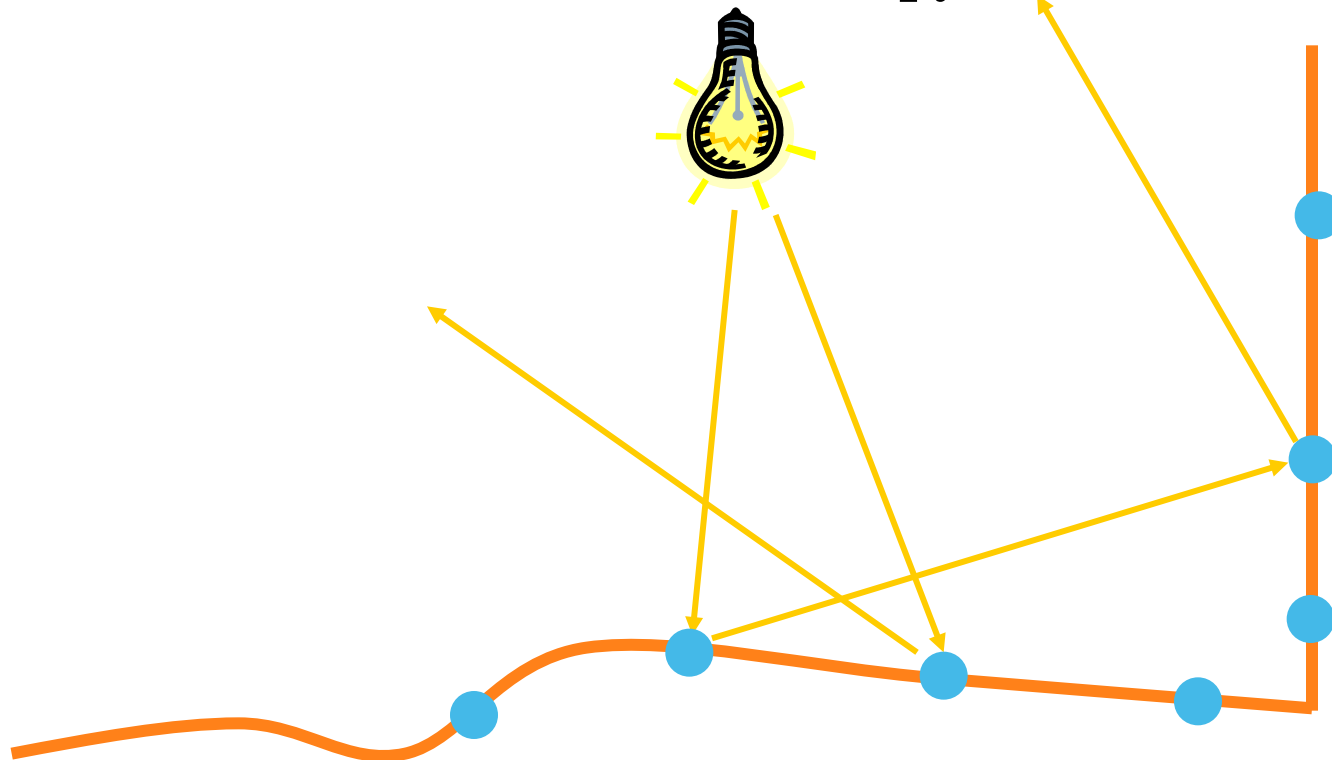
- ❑ Jako path tracing
- ❑ Místo rekurze dotaz do fotonové mapy



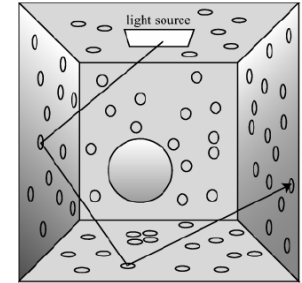
Fáze 1: Rozmístění fotonů



1. **Emise** fotonů ze světelných zdrojů
2. **Sledování** fotonů
3. Uložení fotonů do „fotonové mapy“ (=seznam fotonů)



Emise fotonů



■ Cíl

- Všechny emitované fotony nesou stejný (podobný) tok (variance odhadů osvětlení z fotonové mapy je pak nižší)

1. **Emise** jednoho fotonu (tj. světelné cesty)

1. **Vyber světelný zdroj**

- Náhodně s pravděpodobností úměrnou jeho celkovému toku

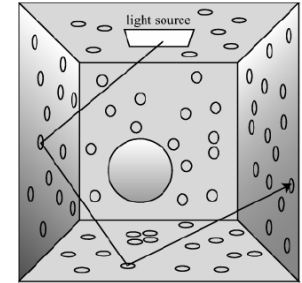
2. **Vyber počátek emitovaného fotonu**

- Pozice světla pro bodové zdroje
- Náhodně (nejčastěji uniformě) pro plošné zdroje světla

3. **Vyber směr emitovaného fotonu**

- Náhodně podle emisní distribuční funkce světla

Emise fotonů



- Tok emitovaného fotonu:

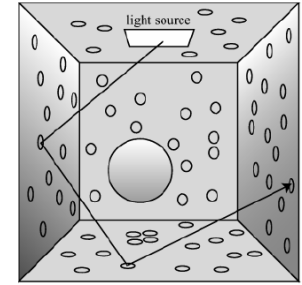
$$\Phi_{p,0} = \frac{1}{N} \frac{L_e(\mathbf{x}_0, \omega_0) |\cos \theta_0|}{P_l p(\mathbf{x}_0, \omega_0)}$$

celkový počet
emitovaných fotonů

(diskrétní) p-nost výběru
světelného zdroje l

hustota p-nosti vzorkování
pozice \mathbf{x}_0 a směru ω_0

Emise fotonů



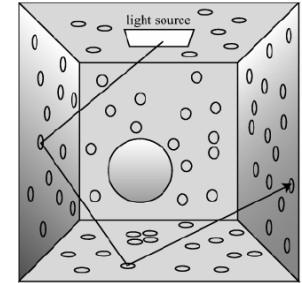
- „Ideální“ vzorkování

$$p(\mathbf{x}, \omega) = \frac{L_e(\mathbf{x}, \omega) |\cos \theta|}{\int_{A_{light}} \int_{H(\mathbf{x})} L_e(\mathbf{x}, \omega) |\cos \theta| d\omega dA} = \frac{L_e(\mathbf{x}, \omega) |\cos \theta|}{\Phi_l}$$

$$P_l = \frac{\Phi_l}{\sum_{i \in lights} \Phi_i} = \frac{\Phi_l}{\Phi_{total}}$$

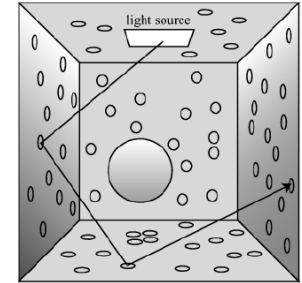
- Všechny emitované fotony pak nesou stejný tok: $\Phi_{p,0} = \frac{\Phi_{total}}{N}$

Sledování fotonů (Photon tracing)



- Podobné jako light tracing
- Interakce foton - plocha
 1. **Ulož „foton“** do fotonové mapy
 - foton = (pozice, příchozí směr, tok)
 2. **Generuj směr odraženého paprsku**
 - BRDF importance sampling
 3. **Aktualizuj tok fotonu**
 - ...
 4. **Ruská ruleta** – náhodná absorpce fotonu
 - ...
- **Požadavek**
 - Zachovat tok fotonu pokud možno konstantní

Sledování fotonů (Photon tracing)



3. Aktualizuj tok fotonu

$$\Phi_{p,j+1}^{tentative} = \Phi_{p,j} \frac{f_r(\mathbf{x}, \omega_o \rightarrow \omega_i) |\cos \theta_o|}{p(\omega_o)}$$

4. Ruská ruleta – náhodná absorpce fotonu

$$q_{p,j+1} = \min \left\{ 1, \frac{\text{Lum}[\Phi_{p,j+1}^{tentative}]}{\text{Lum}[\Phi_{p,j}]} \right\}$$

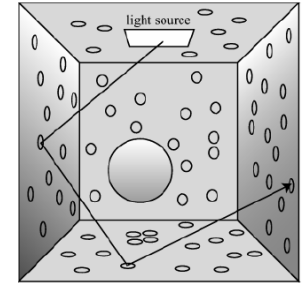
Pravděpodobnost
přežití

$$\Phi_{p,j+1} = \frac{\Phi_{p,j+1}^{tentative}}{q_{p,j+1}}$$

Výsledný tok fotonu
při přežití

- Tato strategie zachovává luminanci fotonů

Sledování fotonů (Photon tracing)



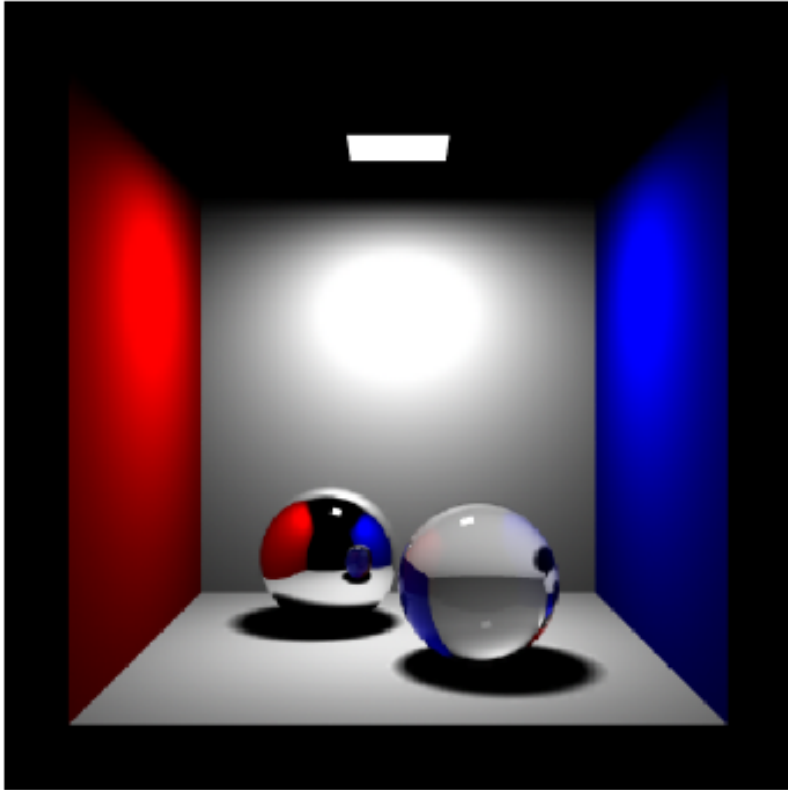
■ Pozor na **lom světla**

- při sledování cest **od kamery** je třeba při lomu **změnit radianci** podle druhé mocniny změny indexu lomu
- avšak fotony nenesou radianci, nýbrž **tok** – **žádná změna toku při lomu nenastává**

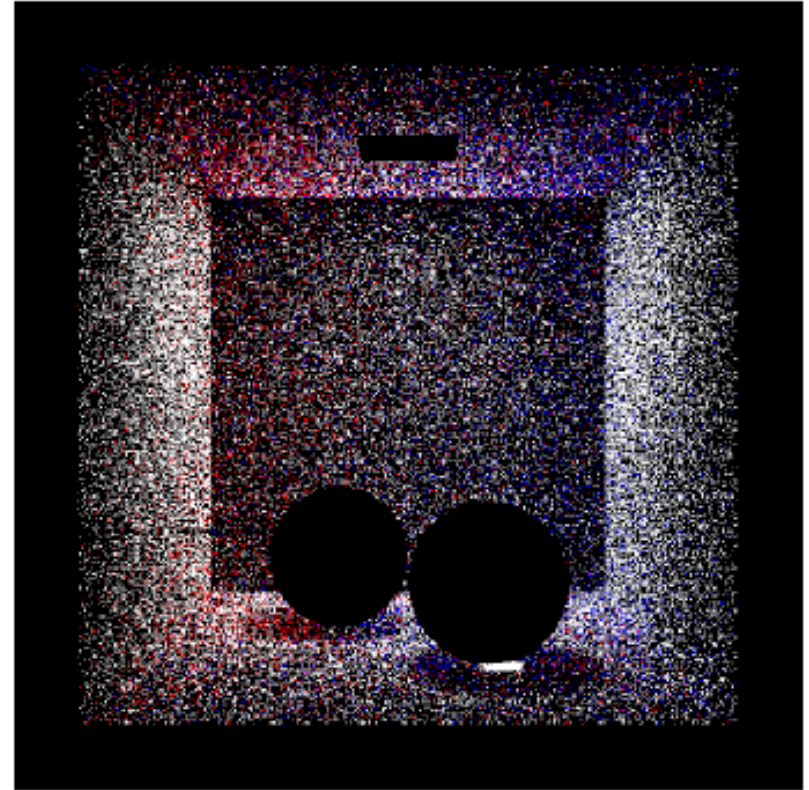
Fotonová mapa

- **Ukládání fotonů do fotonové mapy**
 - Při každé interakci fotonu s difúzní (nebo mírně lesklou, ale ne zrcadlovou) plochou (i při absorpci!!!)
- **Fotonová mapa**
 - Během rozmísťování fotonů pouze lineární seznam fotonů
 - Po rozmístění všech fotonů se postaví *kD*-strom pro rychlejší vyhledávání
- **Foton**
 - pozice: $\mathbf{x}_p = (x, y, z)$
 - příchozí směr: $\omega_p = (\theta, \phi)$ (stačí char[2])
 - energie (tok): $\Phi_p = (r, g, b)$ (stačí RGBE: char[4])
- Počet fotonů – cca $10^5 - 10^7$ stačí pro většinu scén

Fotonová mapa



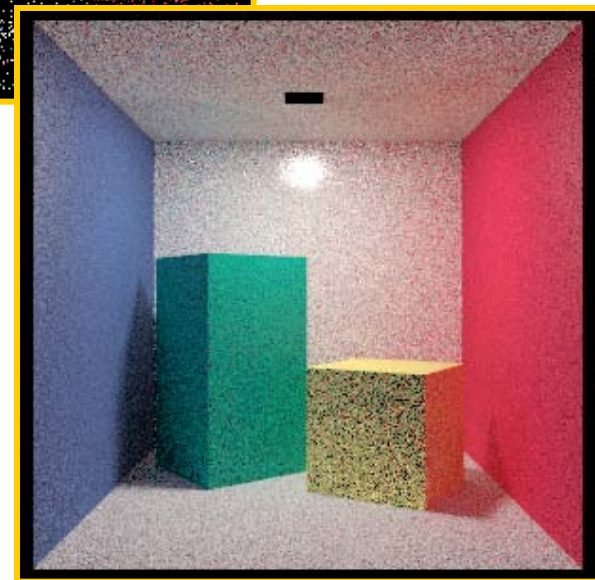
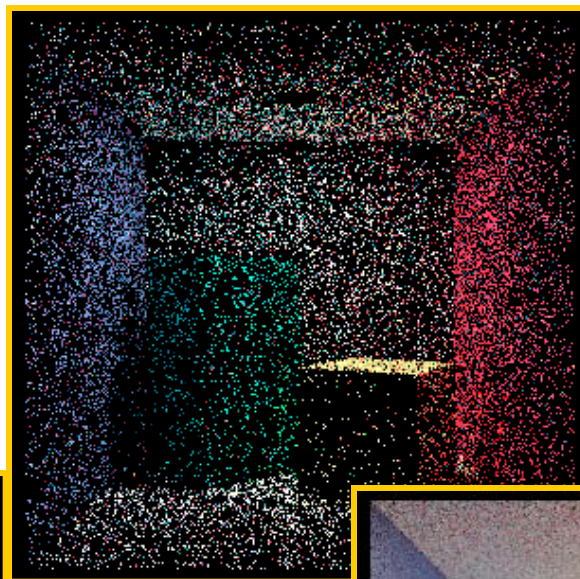
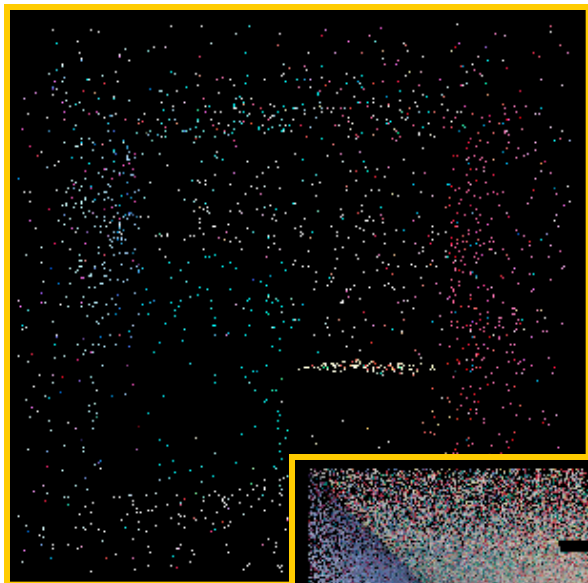
(a)



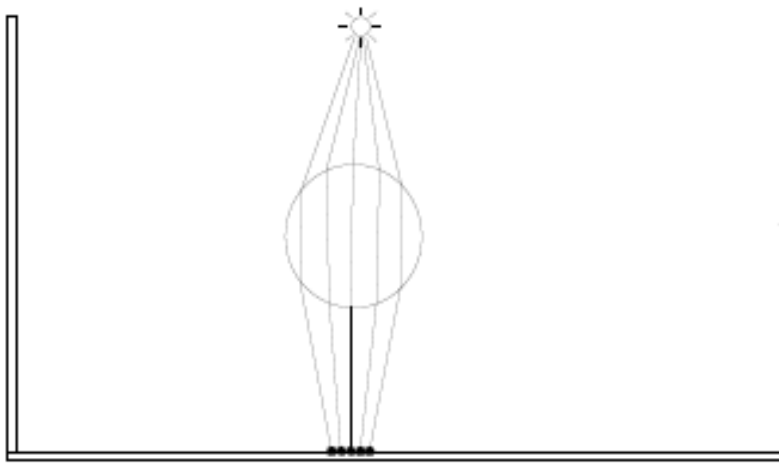
(b)

Figure 2.4: “Cornell box” with glass and chrome spheres: (a) ray traced image (direct illumination and specular reflection and transmission), (b) the photons in the corresponding photon map.

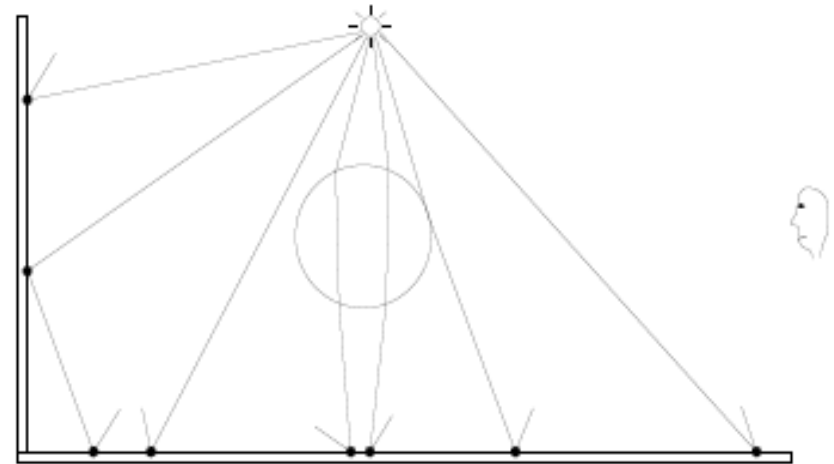
Fotony reprezentují rovnovážnou radianci ve scéně



Dvě fotonové mapy



Mapa kaustik



Globální mapa

Dvě fotonové mapy

1. Globální mapa: **L[S|D]*D**

- Obsahuje i přímé osvětlení

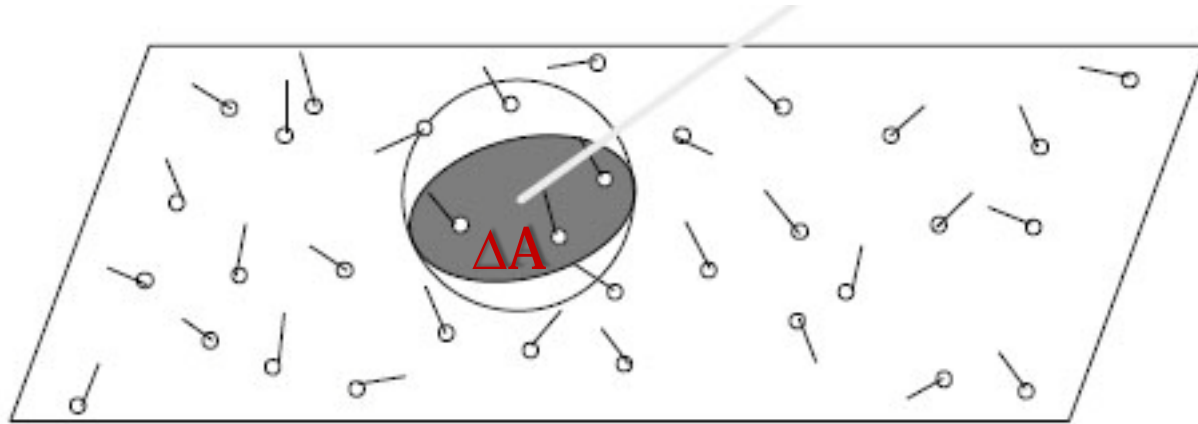
2. Mapa kaustik: **LS+D**

- Obsahuje pouze nepřímé osvětlení
- Je podmnožinou globální mapy
- Různé použití obou map při generování obrázku
 - Je lepší udržovat je zvlášť
- Regulární **gramatika světelných cest**
E ... eye, L ... light, D ... diffuse, S ... specular
G ... glossy (často se zahrnuje do D)

Příprava fotonových map pro rendering

- Při trasování – přidávání fotonů do lineárního seznamu
- Poté postavení *kD-stromu*
 - Pro rendering potřebuji rychlé hledání k nejbližších fotonů

Odhad radiance z fotonové mapy



k .. počet fotonů
v okolí bodu \mathbf{x}

směr incidence
fotonu p

$$L_r(\mathbf{x}, \omega_o) \approx \sum_{p=1}^k f_r(\mathbf{x}, \omega_p, \omega_o) \frac{\Phi_p(\mathbf{x}, \omega_p)}{\Delta A}$$

$$\approx \frac{1}{\pi r^2} \sum_{p=1}^k f_r(\mathbf{x}, \omega_p, \omega_o) \Phi_p(\mathbf{x}, \omega_p)$$

obsah kruhové oblasti ΔA

Odhad radiance z fotonové mapy

RadianceEstimate(x, wo):

```
Color L = (0,0,0);
```

```
int k = locateNearestPhotons(x, wo, n_max, nearest, r);
```

```
// 'nearest' is an array of k nearest photons to x
```

```
// r is the distance from x to the farthest of them
```

```
if ( k < 5 ) return L;
```

```
for p = 1 to k do
```

```
{
```

```
    if( dot ( nearest[p].wi, N) <= 0 ) continue;
```

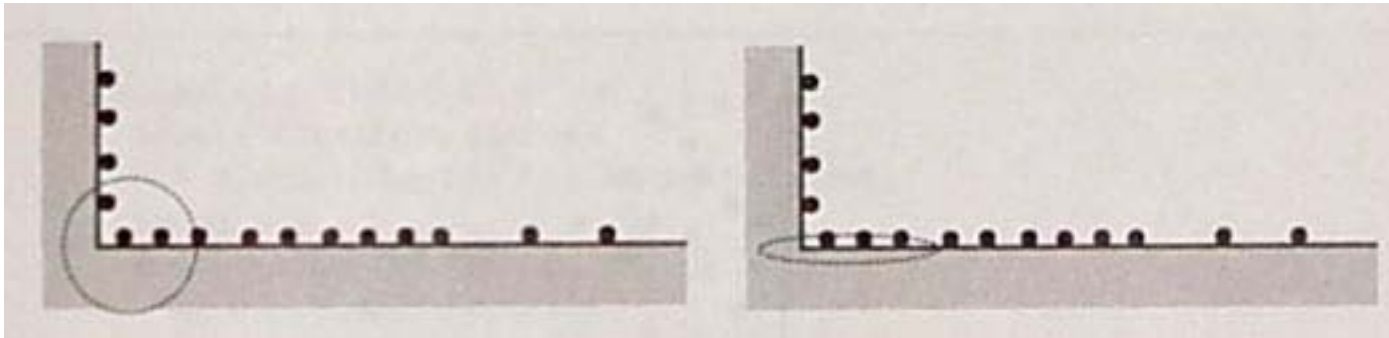
```
    L += fr(x, wo, nearest[p].wi) * nearest[p].flux;
```

```
}
```

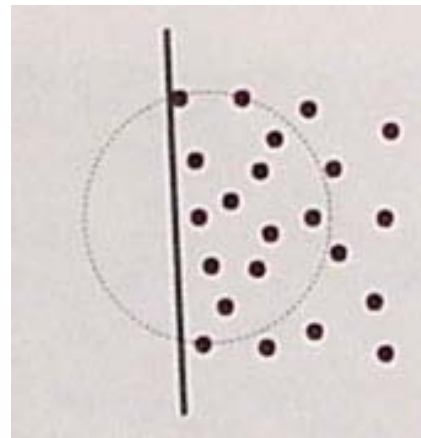
```
return L / (M_PI * r*r);
```

Odhad radiance – problémy

- Zahrnutí nesprávných fotonů do odhadu



- Nesprávný odhad velikost ΔA
 - stěna, hrana kaustiky



Rychlé hledání nejbližších fotonů

- Potřebuji pro odhad radiance z fotonové mapy
- Hledání k nejbližších fotonů je

k-nearest neighbor search (k-NN)

***k*-D strom – Konstrukce**

- Rekurzivní dělení podél osy s maximálním rozsahem
- Dělení
 - Dělicí rovina prochází přímo mediánem
 - Korespondující uzel stromu obsahuje medián
- Reprezentace stromu v lineárním poli, potomky fotonu na indexu i jsou na indexech $2i$ a $2i+1$

***k*-D strom – Hledání nejbližších sousedů**

- Ořezávání průchodu
 - Podle vzdálenosti již nalezeného k -tého nejbližšího fotonu (hledám-li k nejbližších)
 - Dosud nalezené fotony se udržují v max-haldě
 - Podle daného poloměru vyhledávání r (při hledání uvnitř fixního poloměru – „range query“)

Fáze 2: Rendering

- Sledování paprsku z kamery (distributed ray tracing)
 - Rekurze nahrazena odhadem radiance z fotonové mapy
 - Pro ideální zrcadlové plochy se stále používá rekurze jako v klasickém trasování paprsků

Výpočet osvětlení pro primární paprsek (nebo po zrcadlovém odrazu)

- Odražená radiance

$$L_r(\mathbf{x}, \omega_o) = \int_{\Omega} \underbrace{L_i(\mathbf{x}, \omega_i)}_{\text{blue}} \underbrace{f_r(\mathbf{x}, \omega_i \rightarrow \omega_o)}_{\text{red}} \cos \theta_i d\omega_i$$

- Rozdělení příchozí radiance

$$L_i = L_{i,d} + L_{i,c} + L_{i,l}$$

- Rozdělení BRDF

$$f_r = f_{r,D} + f_{r,S}$$

PM ... photon map
FG ... final gathering

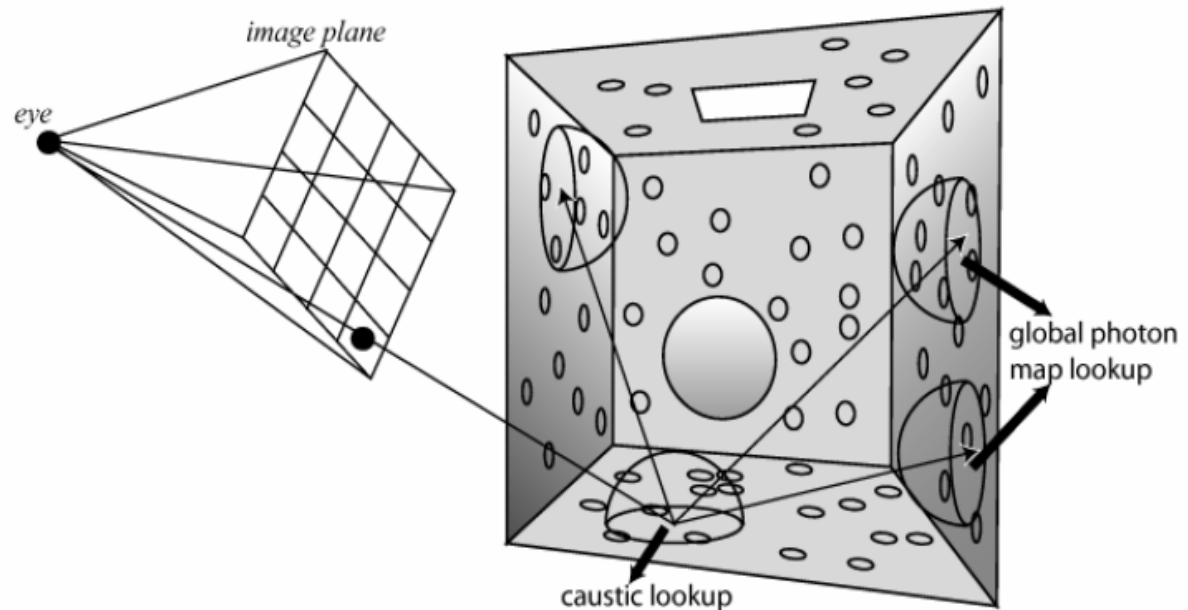
	$f_{r,D}$	$f_{r,S}$
$L_{i,d}$	přímé osvětlení	ideální zrcadlové odrazy/lomy
$L_{i,c}$	kaustiky (PM)	
$L_{i,l}$	difúzní nepřímé (FG + PM)	

Výpočet osvětlení pro primární paprsek (nebo po zrcadlovém odrazu)

- Bez použití fotonových map
 - Přímé osvětlení
 - Jako obvykle: vzorkování světla + stínové paprsky
 - Ideální zrcadlové odrazy / lomy
 - Jako obvykle: deterministické sekundární paprsky
- S použitím fotonových map
 - ...

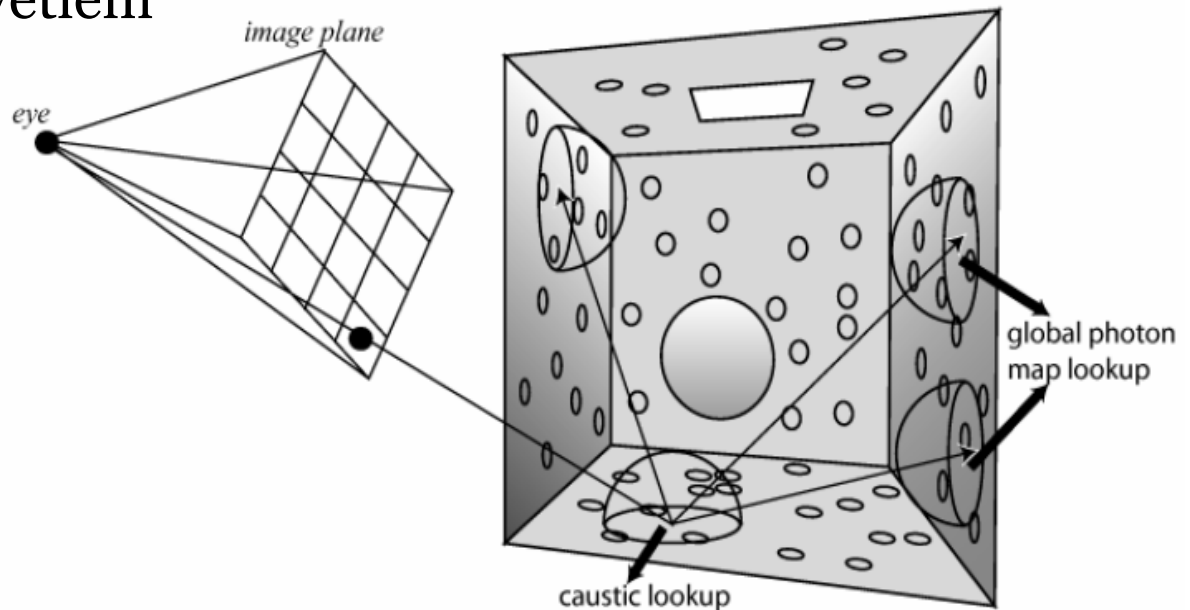
Výpočet osvětlení pro primární paprsek (nebo po zrcadlovém odrazu)

- S použitím fotonových map
 - Kaustiky
 - Odhad radiance z *fotonové mapy kaustik*
 - Nepřímé osvětlení na difúzních a mírně lesklých plochách
 - Final gathering ...



Final gathering (FG)

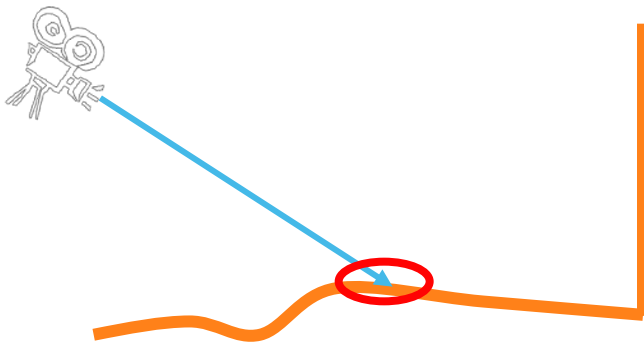
- Nepřímé osvětlení na difúzních a mírně lesklých plochách
- Jedna úroveň rekurze pomocí trasování rozprostřených paprsků (distribution ray tracing)
- Pro průsečíky sekundárních paprsků použij odhad radiance z globální fotonové mapy
 - Není potřeba explicitně počítat přímé osvětlení (je obsaženo v globální mapě)



Proč potřebujeme final gathering?



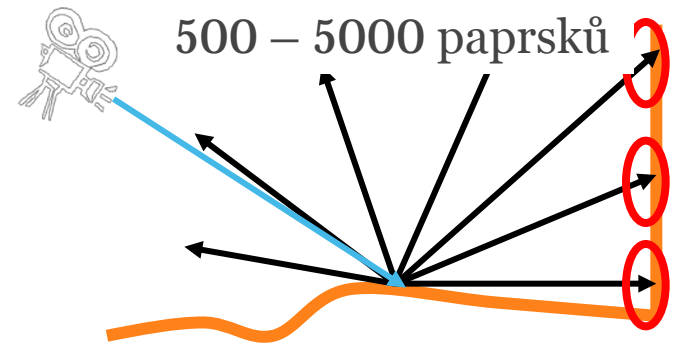
přímé použití



informace v globální mapě
příliš nepřesná



final gathering



nepřesnost v globální
mapě se „zprůměruje“

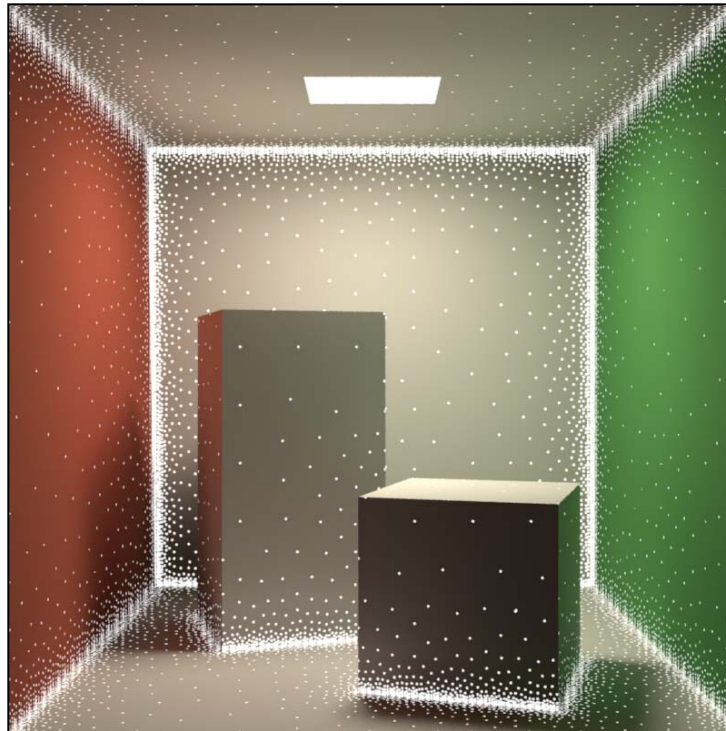
Proč nepotřebujeme final gathering pro kaustiky?

- Kaustiky = zaostření světla => dostatečná hustota fotonů (pozor, jde pouze o heuristiku – nemusí vždy platit)



Možnosti urychlení final gatheringu

1. Irradiance caching (příště)

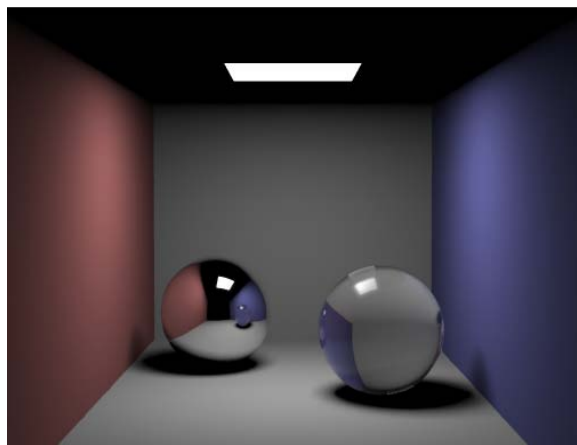


Možnosti urychlení final gatheringu

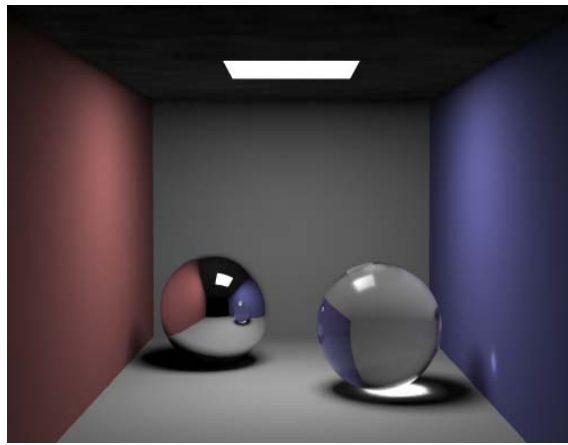
2. Předvýpočet radiance na pozici fotonů

- ❑ Po rozmístění všech fotonů (fáze 1) se vybere podmnožina fotonů, na jejich pozicích se provede odhad radiance z fotonové mapy a výsledek se uloží do separátního kd-stromu.
- ❑ Při renderingu se pro sekundární paprsky z final gatheringu **nemusí provádět k-NN** dotaz do fotonové mapy => stačí pouze najít jeden nejbližší záznam s předpočítanou radiancí
- ❑ Funguje pouze pro dotaz do fotonové mapy na difúzní ploše

Výsledky

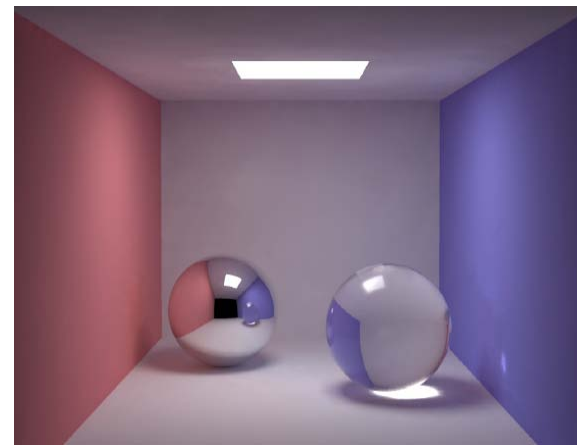


přímé osvětlení (21 s)



kaustiky (45 s)

50 000 fotonů
v mapě kaustik

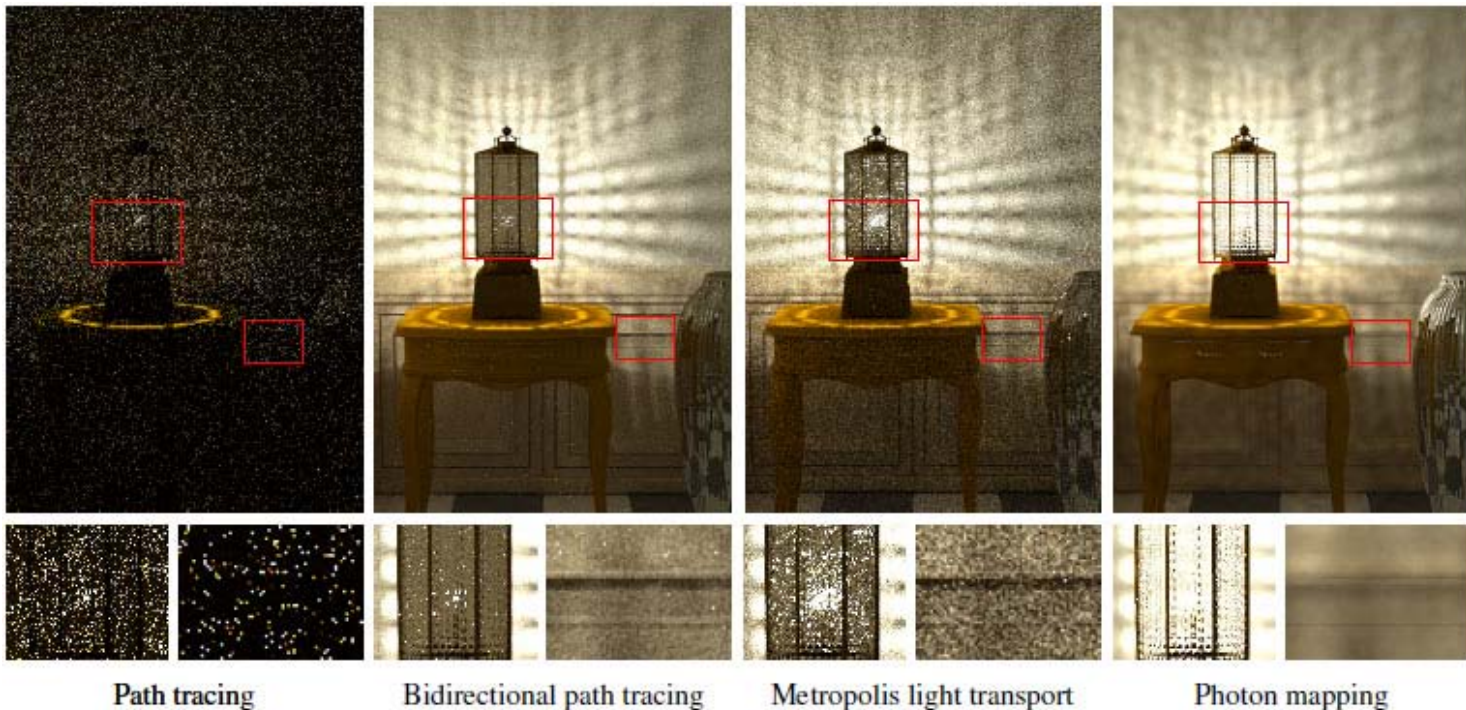


GI (66 s)

200 000 fotonů
v globální mapě

V čem vynikají fotonové mapy?

- Kaustiky
- **SDS cesty** (světlo na dně bazénu)
 - Klasické MC algoritmy nefungují (path tracing, bidirectional path tracing, metropolis light transport)



SDS cesty – dno bazénu



© H.W.Jensen



© Wojciech Jarosz

Lze rozšířit na přenos světla v médiu



Henrik Wann Jensen

... a na rozptyl světla pod povrchem



Praktické nedostatky fotonových map

- Algoritmus nefunguje dobře na lesklých plochách



fotonové mapy



reference

Praktické nedostatky fotonových map

- Algoritmus nefunguje dobře na lesklých plochách
- Co je špatně?
 - **Odhad radiance** z fotonové mapy na lesklé ploše trpí **vysokým rozptylem**
 - Sekundární paprsky z final gatheringu na lesklé ploše často dopadnou do podobného místa ve scéně => korelované výsledky odhadu radiance => vidíme “odraz” fotonové mapy (včetně všech nepřesností, které obsahuje)

Teoretické nedostatky fotonových map

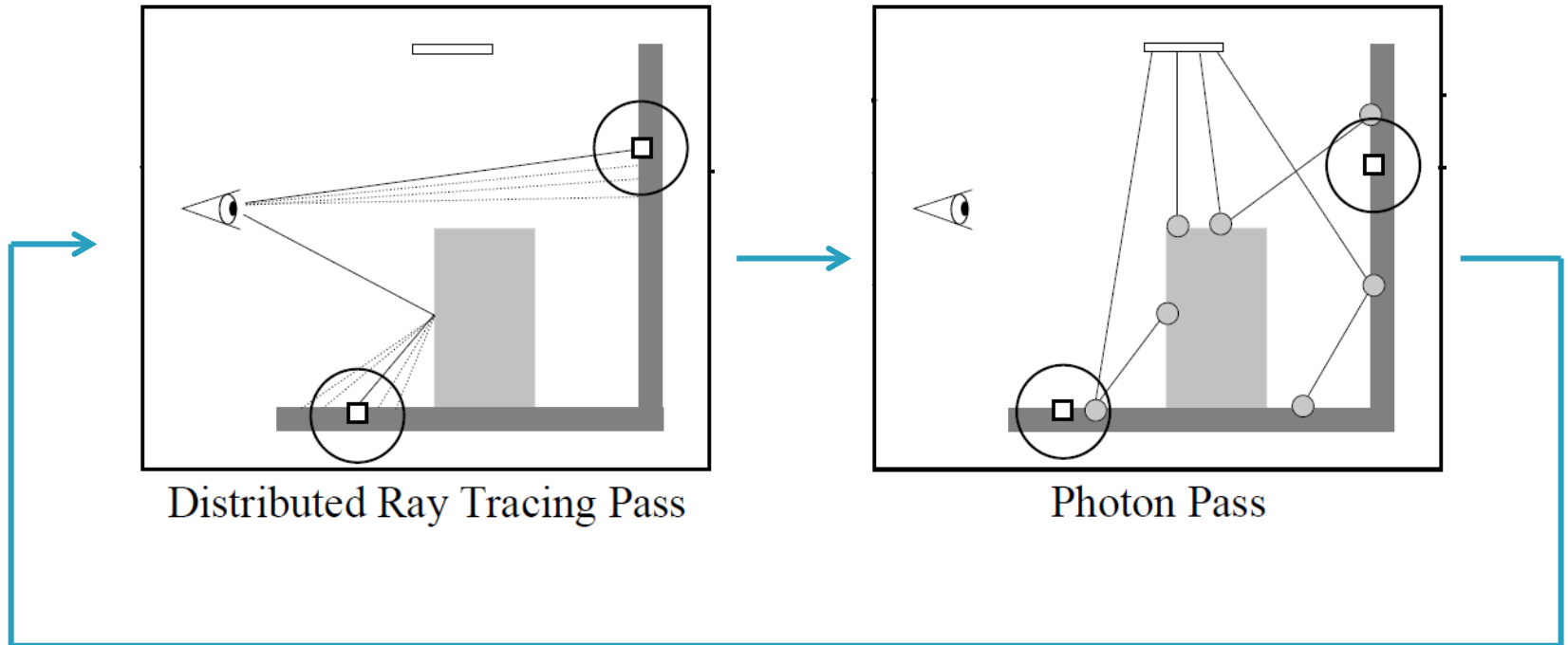
- Výsledek **není nestranný**
 - obsahuje systematickou chybu
- Výsledek je **konzistentní**
 - Konverguje pro nekonečný počet fotonů
 - Toho ale prakticky není možné dosáhnout z důvodu omezené velikosti paměti
 - Řešení: **progressive photon mapping**

Progresivní fotonové mapy

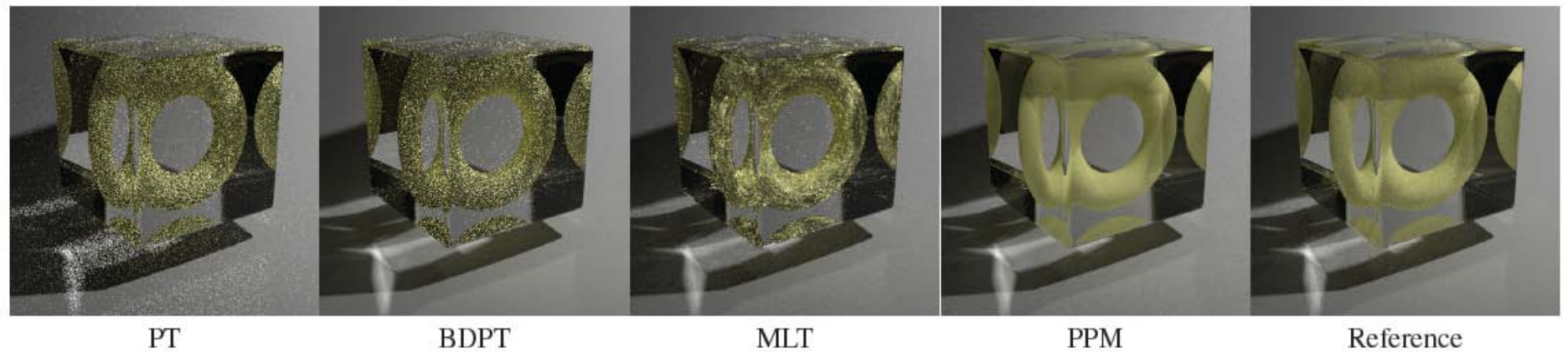
- Rendering v iteracích
- V každé iteraci **zmenšení poloměru hledání nejbližších** fotonů tak, aby:
 - Celkový **bias šel k nule**
 - Celkový **rozptyl šel k nule**
 - (tj. postačující podmínka pro konzistentnost estimátoru)

Progresivní fotonové mapy

- Iterativní procedura



Progresivní fotonové mapy



PT

BDPT

MLT

PPM

Reference

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.*

Progresivní fotonové mapy

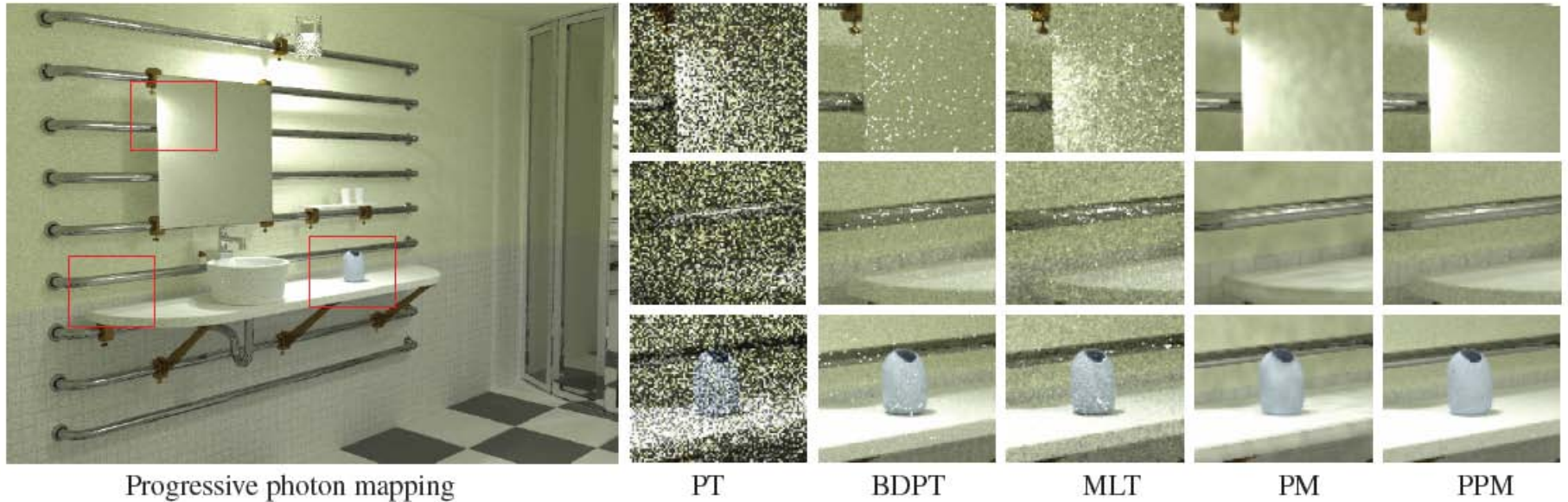
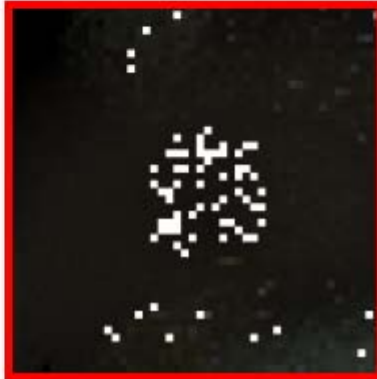
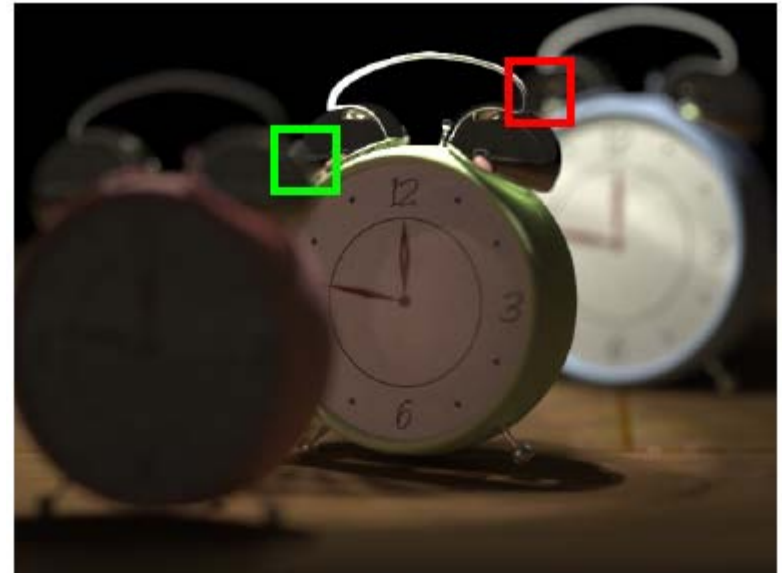
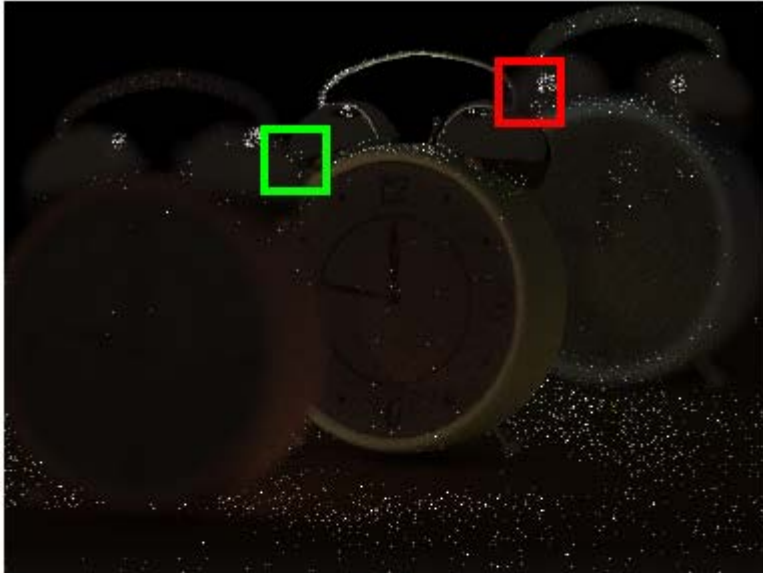
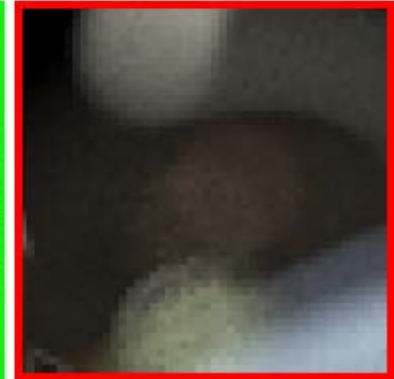


Figure 8: *Lighting simulation in a bathroom. The scene is illuminated by a small lighting fixture consisting of a light source embedded in glass. The illumination in the mirror cannot be resolved using Monte Carlo ray tracing. Photon mapping with 20 million photons results in a noisy and blurry image,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

Progresivní fotonové mapy



BDPT



PPM

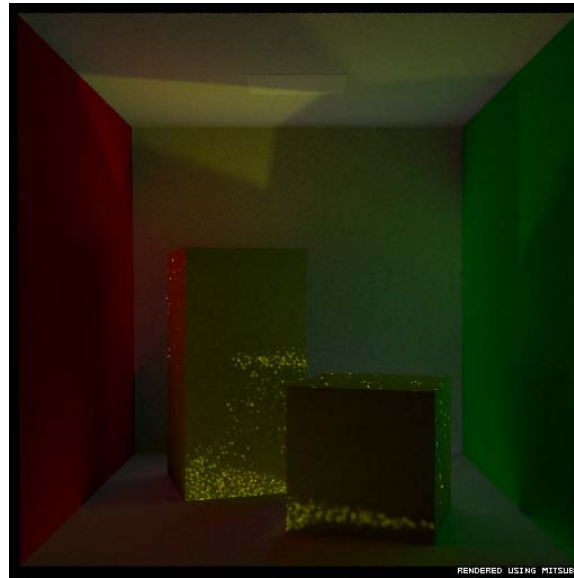
Naše práce: Robustní fotonové mapy

- Kde přesně na cestě od kamery provést odhad radiance (tj. napojit se na cesty od kamery)?
- **Heuristika** ve fotonových mapách:
 - Difúzní povrch ... napoj se co nejdříve
 - Lesklý povrch ... pokračuj cestu od kamery
- Chceme **univerzálnější, robustnější** řešení
 - [Vorba & Křivánek, 2011]

Robustní fotonové mapy



Final gather
(dotaz do PM na druhém
vrcholu cesty od kamery)

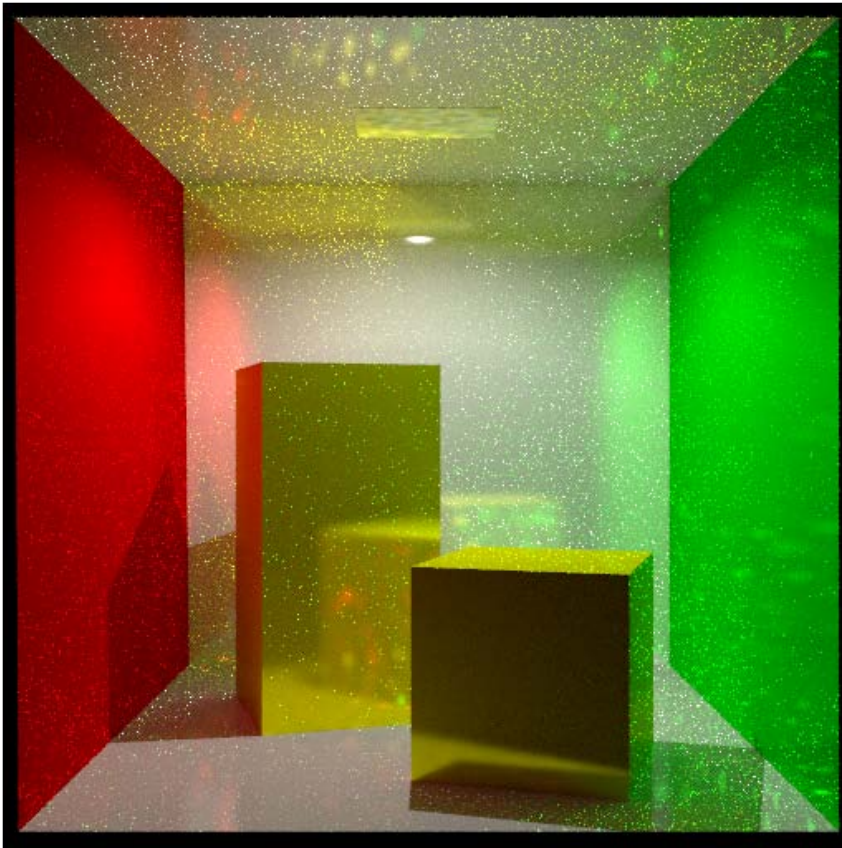


Direct visualization
(dotaz do PM na prvním
vrcholu cesty od kamery)

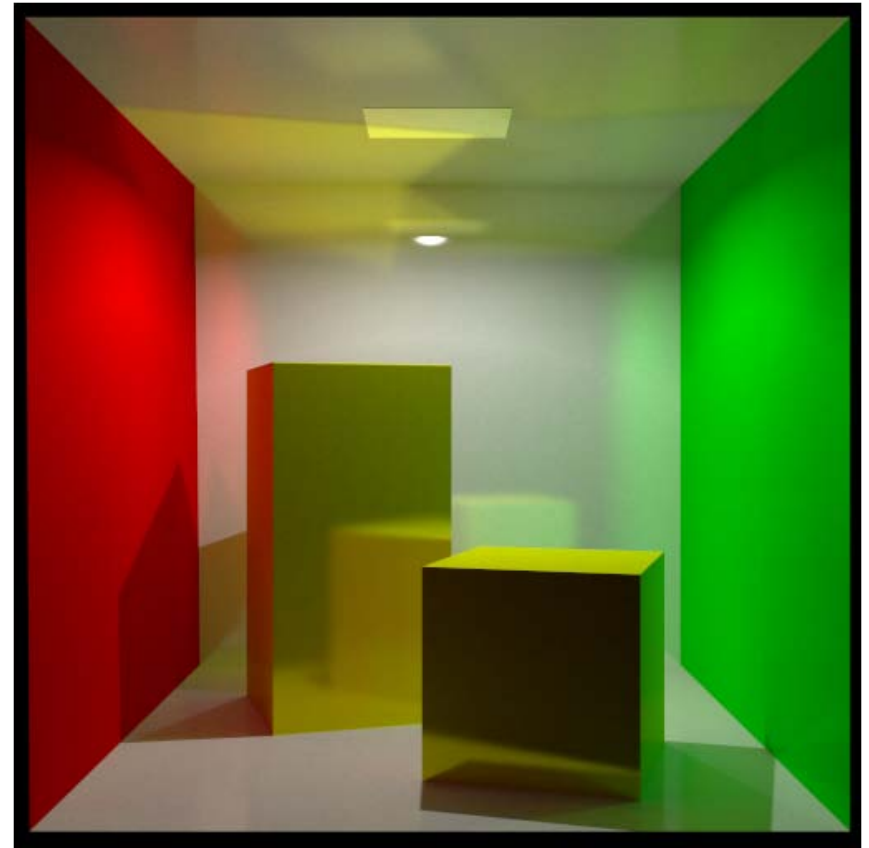


Naše řešení
(dotaz do PM na všech
vrcholech cesty do
kamery + kombinace
pomocí MIS)

Robustní fotonové mapy



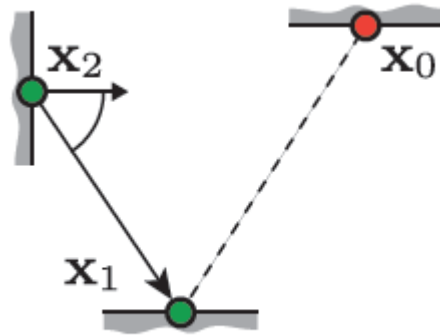
Fotonové mapy
(final gathering)



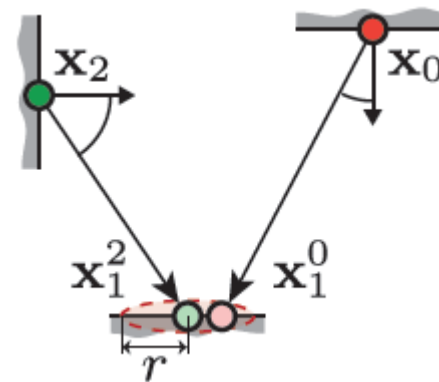
Naše řešení

Robustní kombinace BDPT + PPM

- Interpretace fotonových map jako další možné strategie vzorkování cest v BDPT [Georgiev & Křivánek, 2011]



BDPT
(spojení vrcholů pomocí stínového paprsku)



PM
(spojení vrcholů pomocí dotazu do PM)

Robustní kombinace BDPT + PPM



Path Tracing

Robustní kombinace BDPT + PPM



Robustní kombinace BDPT + PPM



Robustní kombinace BDPT + PPM



Náš výsledek (Vorba): Robustní PPM

Robustní kombinace BDPT + PPM



Literatura

- Jensen H.W.: **Realistic Image Synthesis using Photon Mapping**. A.K. Peters, 2001
- Hachisuka & Jensen. **Stochastic Progressive Photon Mapping**, ACM Trans. Graph. (SIGGRAPH Asia 2009).
- Knaus & Zwicker. **Progressive photon mapping: A probabilistic approach**. ACM Trans. Graph. 30(3): 25 (2011)