

---

B4M39RSO

# **Vychýlené (biased) metody globálního osvětlení**

---

Vlastimil Havran

ČVUT v Praze

Verze 2018

---

# **Vychýlené versus nestranné metody**

---

# Vychýlené vs. nestranné odhady (Biased vs. Unbiased Estimates)

- Trasování cest (path tracing) – nestranné ale se šumem
- Praxe – raději méně šumu za cenu vychýleného odhadu
  - vychýlení v obrázku stejně nikdo nepozná (většinou...)

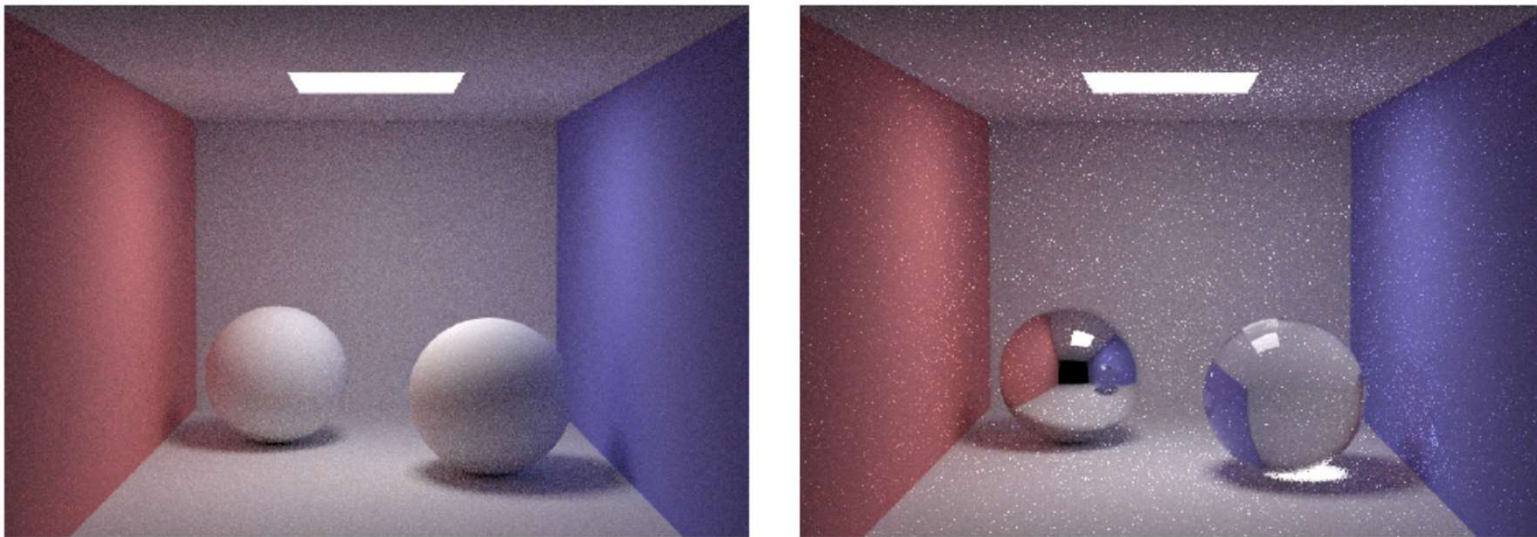


Figure 10.1: A path traced box scene with 10 paths/pixel. On the left the box has two diffuse spheres, and on the right box box has a mirror and a glass sphere. Note how the simple change of the sphere material causes a significant increase in the noise in the right image.

# Nestranný, konzistentní a vychýlený odhad

- Nestranný odhad:  $E[X] = \int \dots$ 
  - variance (t.j. šum) je jediná chyba, žádná systematická chyba
  - jak se zbavit šumu
    - lepší vzorkování (importance sampling, stratified sampling, ...)
  - Příklady: distributed ray tracing, path tracing, bi-directional path tracing
- Konzistentní odhad:
  - systematická chyba, ale v limitě jde ke správnému výsledku
  - efektivnější než nestranné metody  $\lim_{N \rightarrow \infty} E[X] \rightarrow \int \dots$
  - př. fotonové mapy, irradiance caching
- Vychýlený odhad:
  - Velikost chyby není vůbec garantována ani v limitě
  - Pro realtime aplikace se používá velmi často

---

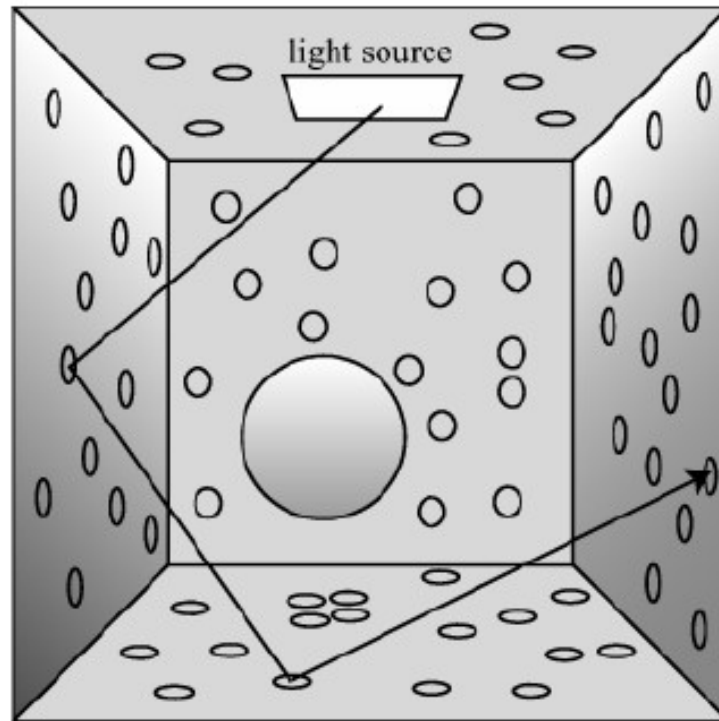
# **Fotonové mapy**

---

# Fotonové mapy

- Podobné obousměrnému trasování cest
- Generování cest ze světla a z kamery jsou dva oddělené procesy, které se potkávají pomocí odhadu hustoty pravděpodobnosti:
  1. Trasování fotonů (photon tracing)
    - od světla
    - tvorba fotonové mapy
  2. Trasování rozprostřených paprsků (distributed ray tracing)
    - od kamery
    - využití fotonové mapy pro tvorbu obrazu s GI

# Průchod 1: Emise a trasování fotonů



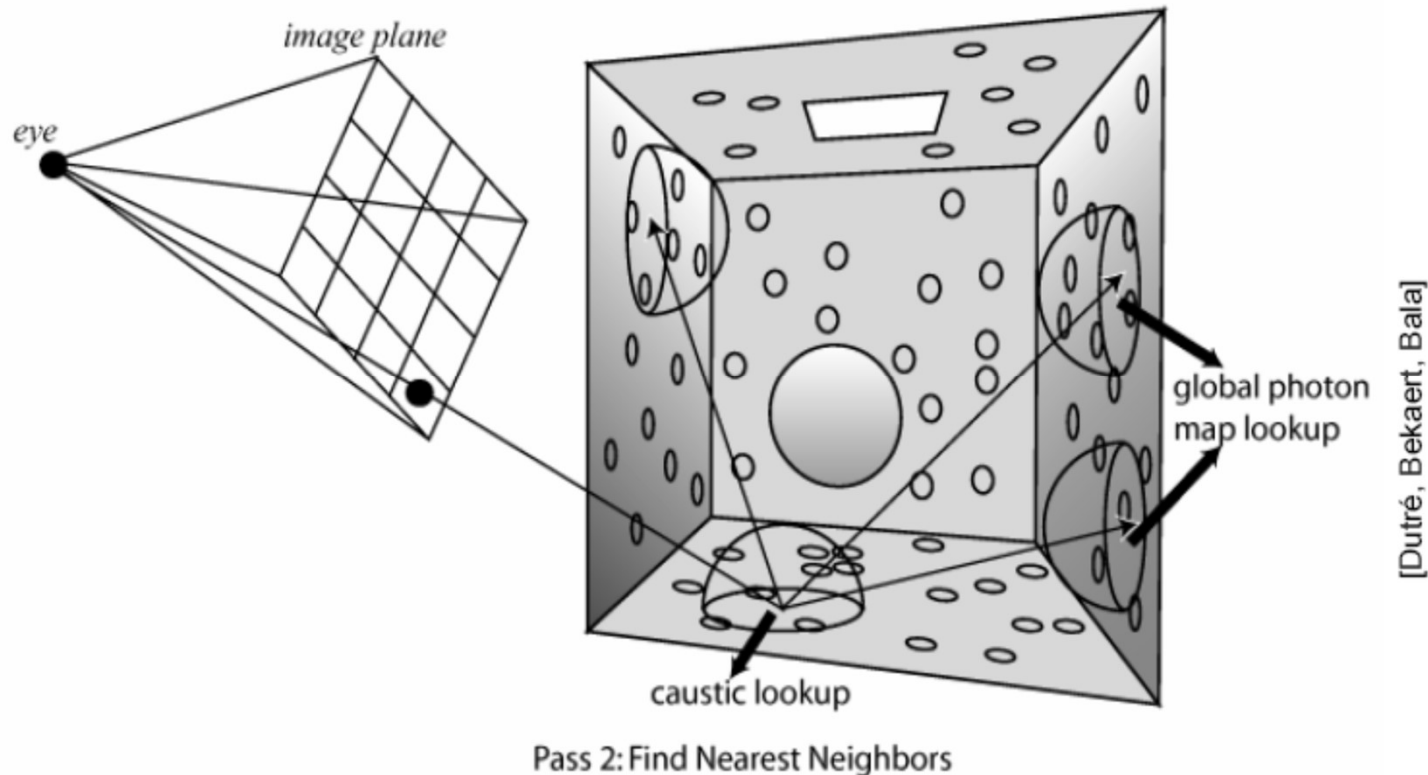
Pass 1: Shoot Photons

[Dutré, Bekaert, Bala]

- Výsledkem je **fotonová mapa**
  - fotony rozmístěné ve scéně, každý foton nese tok [W]

## Průchod 2

# Trasování rozptřených paprsků



- Využití fotonové mapy pro tvorbu obrazu s GI
- Používá se final gathering (cca 50-2000 paprsků na jeden primární paprsek)



# Emise fotonů

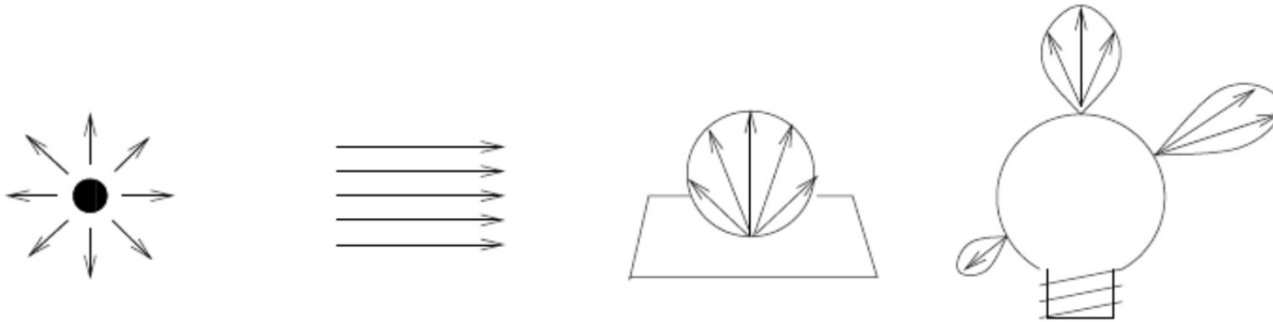


Figure 2.1: Emission from light sources: (a) point light, (b) directional light, (c) square light, (d) general light.

1. Vyber světlo s pravděpodobností úměrnou jeho toku
  2. Vyber počátek a směr emitovaného fotonu podle emisivní distribuční funkce světla
  3. Trasuj foton (viz dále)
  4. Opakuj dokud není ve scéně dostatek fotonů (zadáno)
- Důsledek bodu 1 a 2:
    - Všechny emitované fotony nesou cca stejný tok:  
$$\Phi_p = \Phi_{\text{total}} / N$$

# Trasování fotonů (Photon Tracing)

- Stejně jako trasování cest – protože BRDF je reciproční !
  - Výběr interakce: absorpce / difúzní odraz / lom ...
    - podle odrazivosti jednotlivých komponent BRDF
  - Generování nových směrů
    - Vzorkování podle důležitosti (importance sampling) funkce  $BRDF \cdot \cos(\theta)$
- Důsledek:
  - Tok fotonů se při trasování scénou téměř nemění  
  
(Neboť normalizace při použití „importance sampling“ vykrátí násobení energie fotonu BRDF.)

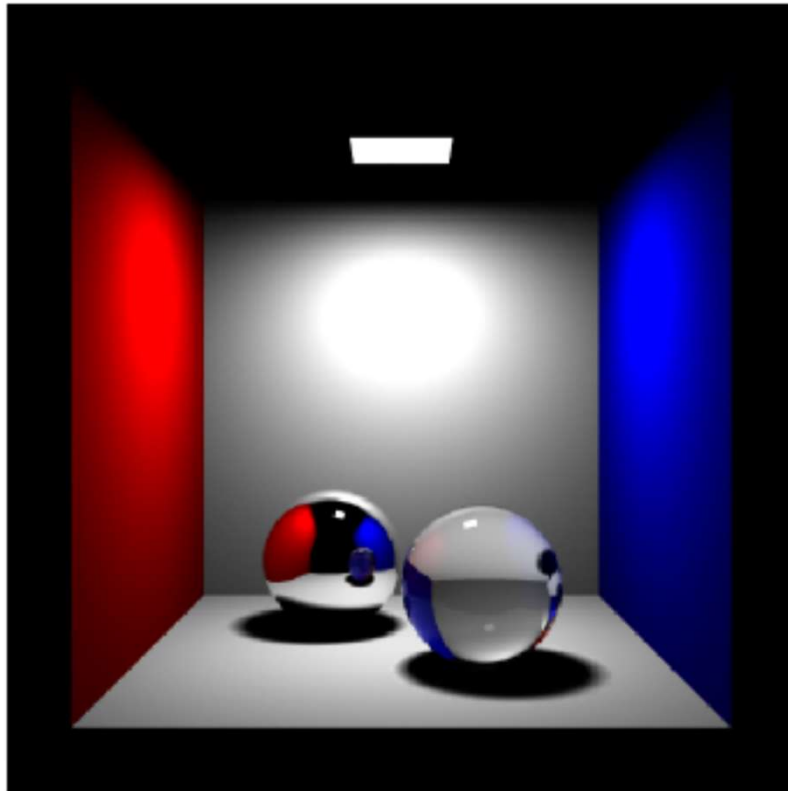
# Trasování fotonů (Photon Tracing)

- Pozor na lom světla
  - při trasování paprsků je třeba při lomu změnit zář podle druhé mocniny indexu lomu
  - avšak fotony nenesou radianci, nýbrž **tok** – žádná změna toku při lomu nenastává
- Ukládání fotonů do fotonové mapy
  - při každé interakci fotonu s difúzní (nebo mírně lesklou, ale ne zrcadlovou) plochou (i při absorpci!!!)

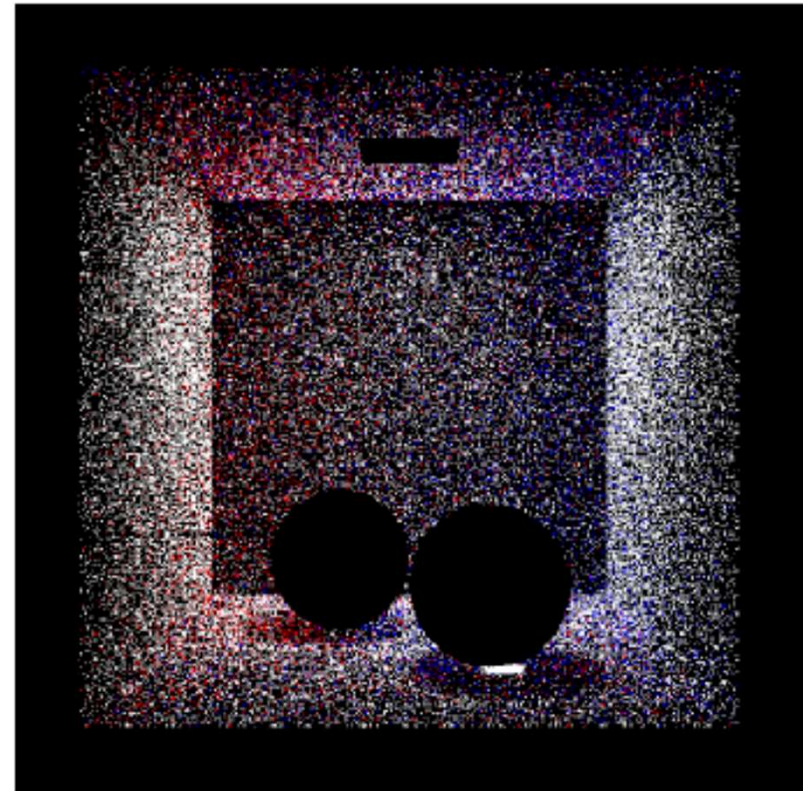
# Fotonová mapa

- Fotonová mapa
  - Během trasování pouze lineární seznam fotonů
  - Po ukončení trasování se postaví **kd-strom** pro rychlejší vyhledávání, viz přednášky předmětu DPG
- Foton
  - pozice:  $\mathbf{x}_p = (x, y, z)$
  - příchozí směr:  $\omega_p = (\theta, \phi)$
  - energie (tok):  $\Phi_p = (r, g, b)$
- Počet fotonů – cca 1 milión stačí pro většinu scén, pro velké scény 10 miliónů fotonů i více.

# Fotonová mapa



(a)



(b)

Henrik Wann Jensen

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.

# Dvě fotonové mapy

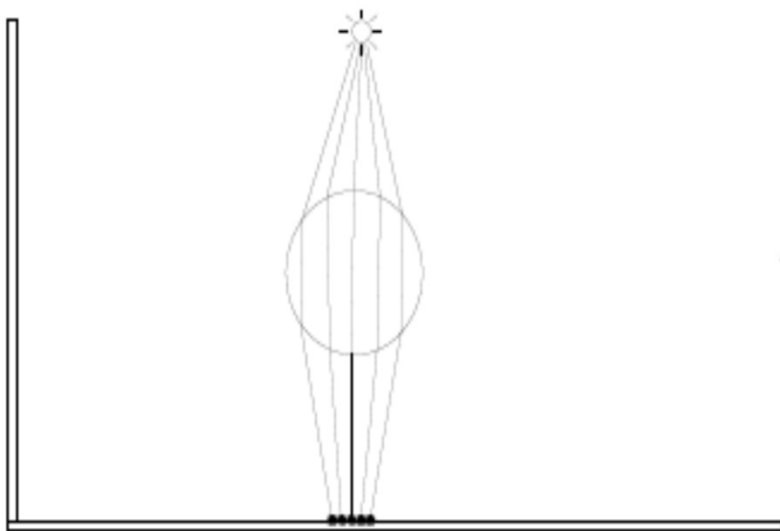
1. Globální mapa:  $L[S|D]^*D$

- Obsahuje i přímé osvětlení

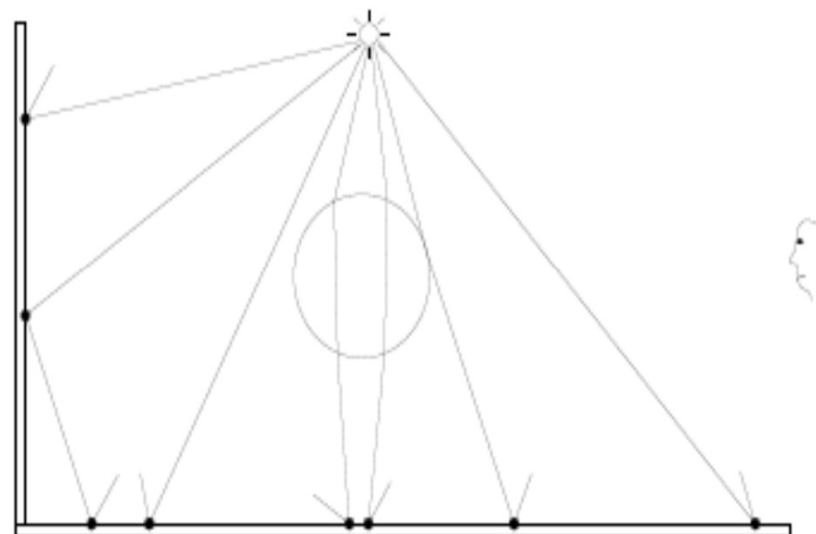
2. Mapa kaustik:  $LS^+D$

- Obsahuje pouze nepřímé osvětlení
- 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 S)

# Dvě fotonové mapy



Mapa kaustik



Globální mapa

---

# Příprava fotonových map pro rendering

- Při trasování – přidávání fotonů do lineárního seznamu
- Poté postavení *kD-stromu*
  - Pro rendering potřebuji rychlé hledání k nejbližším fotonům (kNN search) či vyhledávání v rozsahu



# Odhad radiance z fotonové mapy

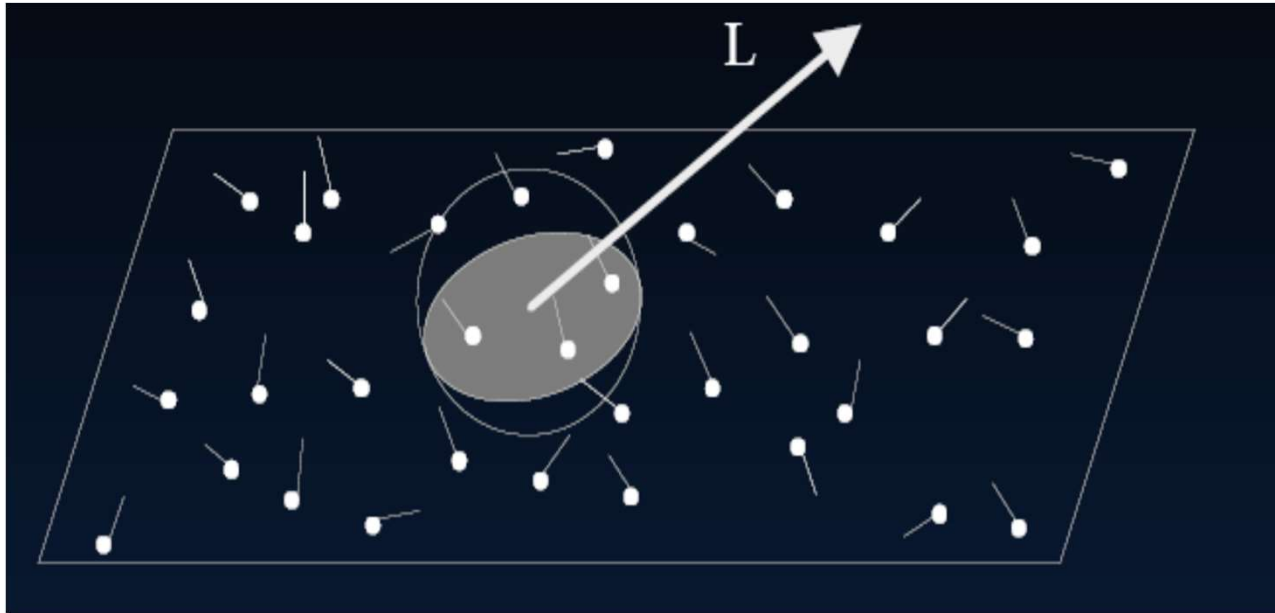
- Bude zapotřebí pro generování obrazu s GI

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

- Odvození:

$$\begin{aligned} L_o(\mathbf{x}, \omega_o) &= \int_{\Omega} f_r(\omega_i, \mathbf{x}, \omega_o) L_i(\mathbf{x}, \omega_i) \cos \theta_i d\omega_i \\ &= \int_{\Omega} f_r(\omega_i, \mathbf{x}, \omega_o) \frac{d^2\Phi_i(\mathbf{x}, \omega_i)}{\cos \theta_i d\omega_i dA_i} \cos \theta_i d\omega_i \\ &= \int_{\Omega} f_r(\omega_i, \mathbf{x}, \omega_o) \frac{d^2\Phi_i(\mathbf{x}, \omega_i)}{dA_i} \end{aligned}$$

# Odhad radiance z fotonové mapy



$$\begin{aligned} L_o(\mathbf{x}, \omega_o) &= \sum_{p=1}^n f_r(\omega_p, \mathbf{x}, \omega_o) \frac{\Delta\Phi_p(\mathbf{x}, \omega_p)}{\Delta A} \\ &= \frac{1}{\pi r^2} \sum_{p=1}^n f_r(\omega_p, \mathbf{x}, \omega_o) \Delta\Phi_p(\mathbf{x}, \omega_p) \end{aligned}$$

# Odhad radiance z fotonové mapy

```
RadianceEstimate(x, wo):  
    Color L = (0,0,0);  
    int n = locate_nearest_photons(x,wo,n_max, nearest, r);  
    // 'nearest' is an array of n nearest photons to x  
    // r is the distance from x to the farthest of them  
    if ( n < 8 ) return L;  
    for p = 1 to n 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);
```

# Rychlé hledání nejbližších fotonů

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

„*k-nearest neighbor search*“ (kNN)

- viz. Datové struktury v počítačové grafice

## Průchod 2: Rendering

- Trasování rozprostřených paprsků (distributed ray tracing)
  - omezení rekurze pro nepřímé osvětlení na difúzních a mírně lesklých plochách
  - rekurze nahrazena odhadem radiance z fotonové mapy
  - pro ideální zrcadlové plochy se stále používá rekurze jako v klasickém trasování paprsků

# Průchod 2: Rendering

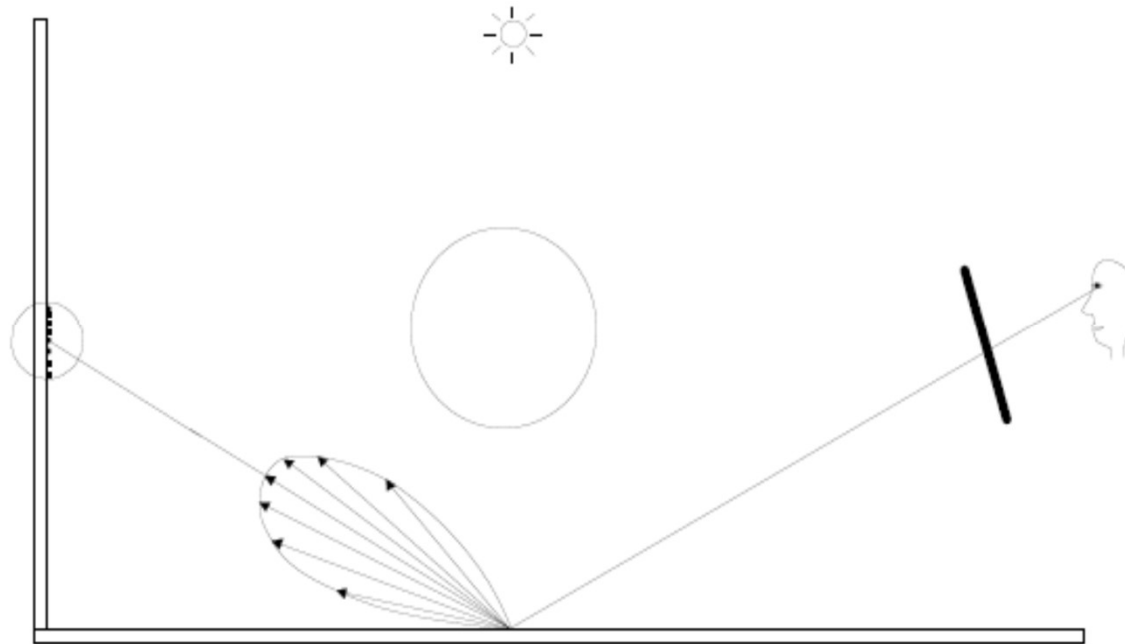
- Přímé osvětlení
  - Jako obvykle: vzorkování světla + stínové paprsky
- Ideální zrcadlové odrazy / lomy
  - Jako obvykle: deterministické sekundární paprsky
- Kaustiky
  - Odhad z fotonové mapy kaustik
    - Pro primární paprsky
    - nebo pro sekundární paprsky po ideálním zrcadlovém lomu/odrazu

## Průchod 2: Rendering

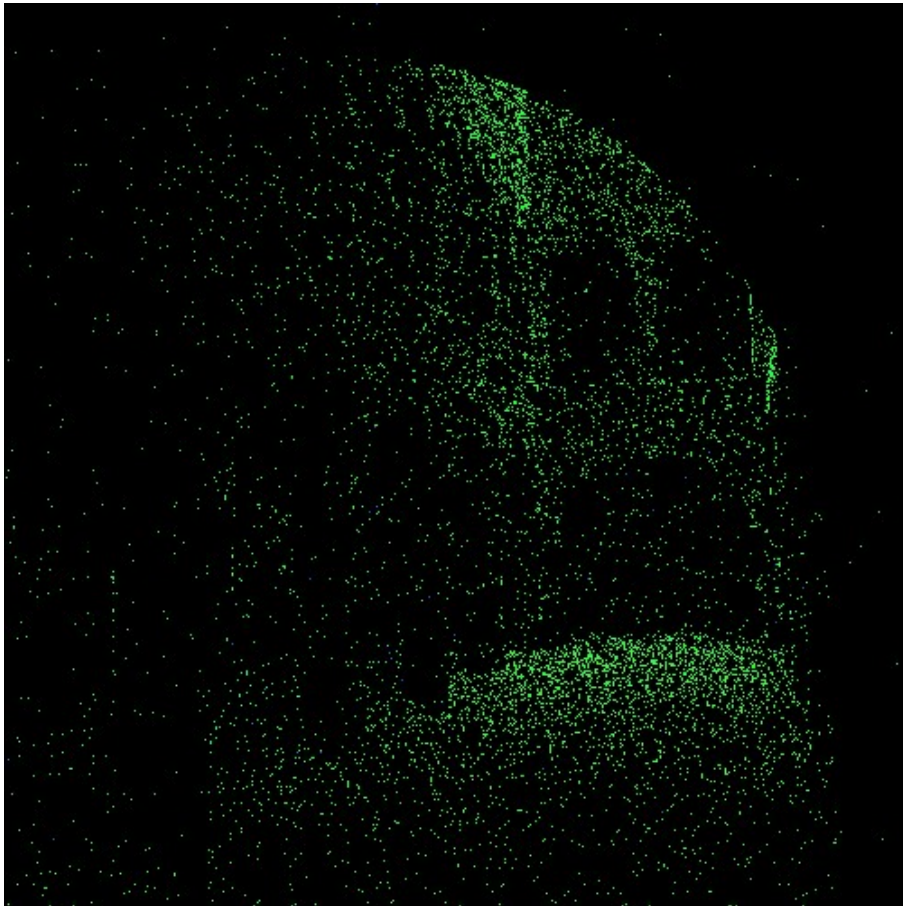
- Nepřímé osvětlení na difúzních a mírně lesklých plochách

### Posbírání světla – Final gathering

- Jedna úroveň rekurze pomocí trasování rozprostřených paprsků
- Pak použij odhad z globální fotonové mapy



# Example of Direct Visualization of Photon Map: why we need final gathering



Photon Hits



Direct Visualization

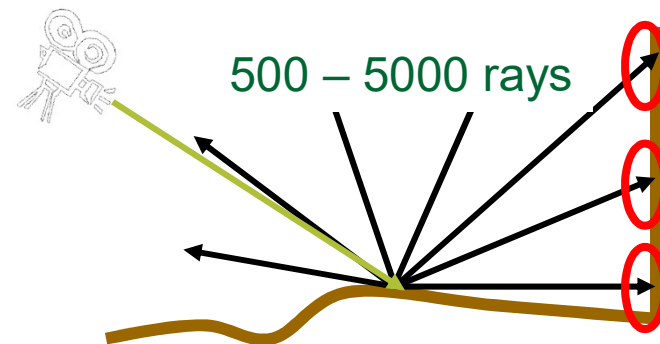
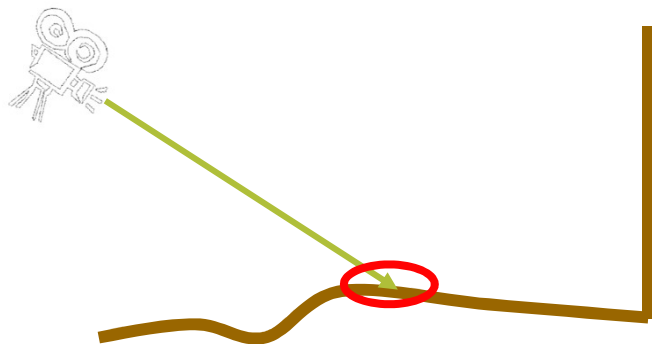


# Důvod pro Final Gathering

přímé použití  
fotonových map



final gathering



# Důvod pro Final Gathering

## Přímé použití

Odhad z globální mapy pro primární paprsky

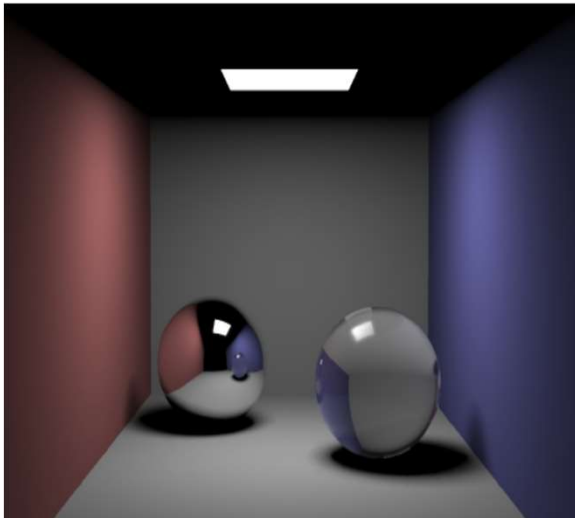
- ❑ špatná kvalita obrazu
- ❑ informace v globální mapě příliš nepřesná

## Final gathering

Odhad z globální mapy až po jedné úrovni trasování rozprostřených paprsků

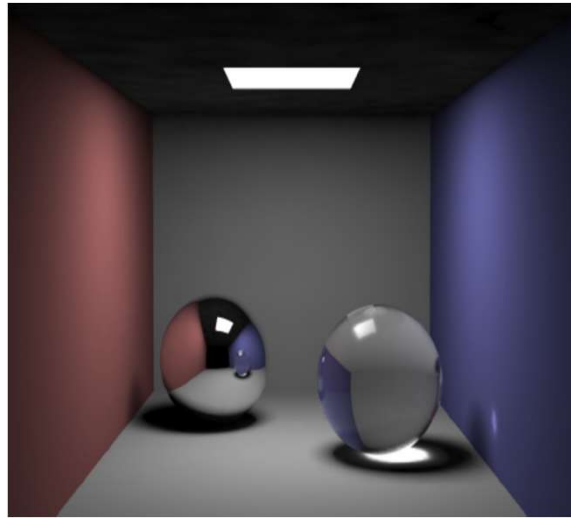
- ❑ výborná kvalita obrazu
- ❑ nepřesnost v globální mapě se „zprůměruje“
- ❑ Časově náročnější výpočet, viz. předmět DPG

# Výsledky – přímé + nepřímé osvětlení (kaustiky + difúzní)



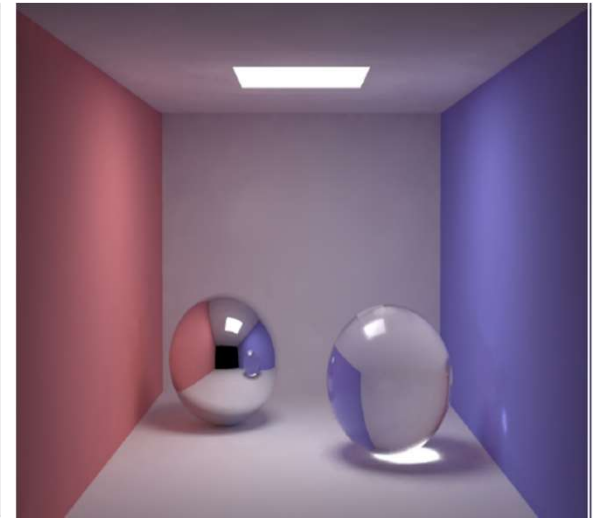
přímé

21 s



kaustiky

50 000 caust. fot.  
45 s



gi

200 000 gl. fotonů  
66 s

# Výsledky



Fotonové mapy – 10 000 fotonů celkem,  
50 fotonů pro odhad radiance



Path tracing – 1000 cest na pixel

# Výsledky



Henrik Wann Jensen

# Funguje i po mlhu...



Henrik Wann Jensen



# Rozptyl světla pod povrchem



---

# **Irradiance Caching**

---



# Irradiance caching

- Irradiance caching:  
Rychlejší výpočet difúzní složky nepřímého osvětlení
- Difúzní odraz - připomenutí

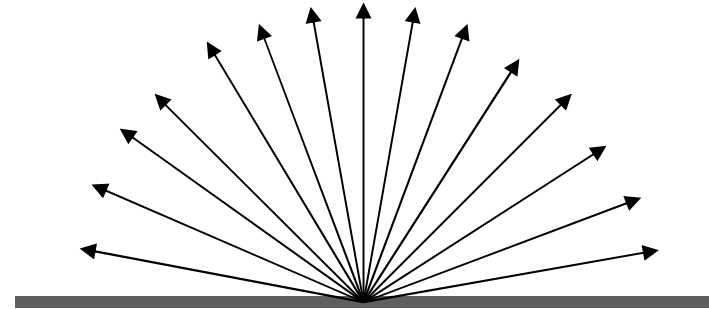
$$L_o(\mathbf{x}) = E(\mathbf{x}) * \rho_d(\mathbf{x}) / \pi$$

- pohledová nezávislost
  - odchozí radiance je nezávislá na směru pohledu
  - pro výpočet odchozí radiance v libovolném směru stačí znát celkovou příchozí irradianci

# Motivace pro Irradiance Caching

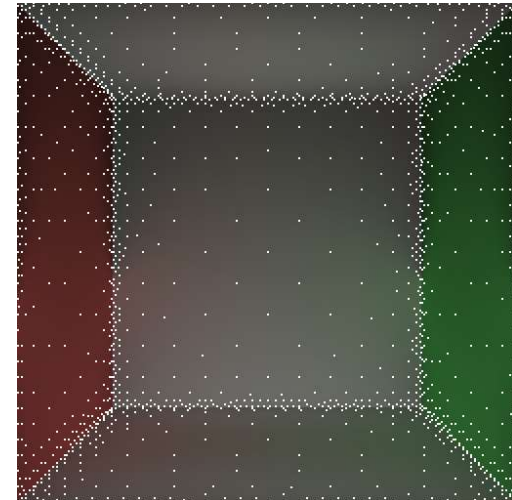
## ■ Final gathering

- ❑ Vzorkování hemisféry
- ❑ 50-2000 paprsků
- ❑ Časově náročný výpočet
- ❑ Prostorová koherence
  - Difúzní nepřímé osvětlení se na plochách mění pomalu



## ■ Irradiance caching

- ❑ Vzorkování hemisféry pouze někde
- ❑ Výsledky uloží do cache jako záznamy
- ❑ Jinde interpoluj z cache



# Difúzní nepřímé osvětlení se mění pomalu

Přímé a nepřímé ozáření v součtu.



# Difúzní nepřímé osvětlení se mění pomalu

Pouze nepřímé ozáření (irradiance) – mění se pomalu.



# Irradiance Caching

- Líné vyhodnocování nových záznamů
- Příklad: Záznamy E1 a E2 již existují
  - V bodě A interpoluj (rychlé)
  - V bodě B extrapoluj (rychlé)
  - V bodě C přidej nový záznam (pomalé)

- **Algoritmus:**

```
GetIrradiance(x) :
```

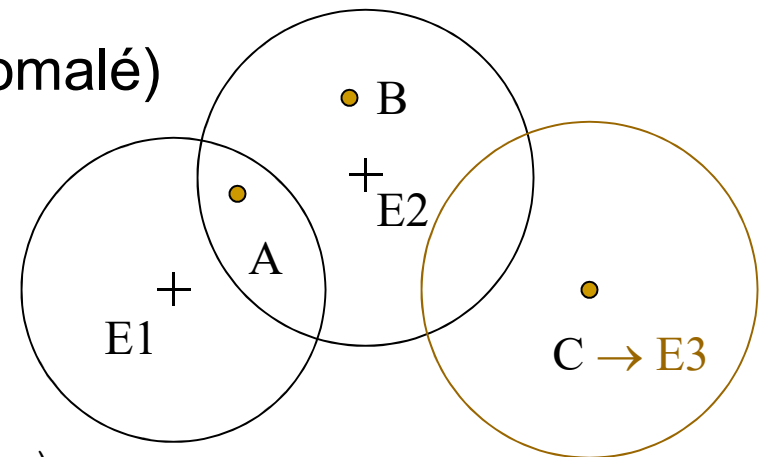
```
    Color E;
```

```
    if( ! InterpolateFromCache(x, E) )
```

```
        E = SampleHemisphere(x);
```

```
        InsertIntoCache(E, x);
```

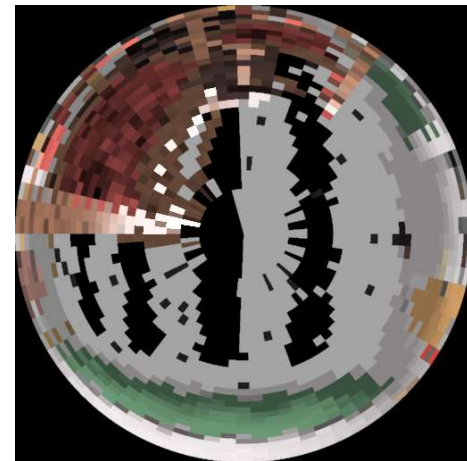
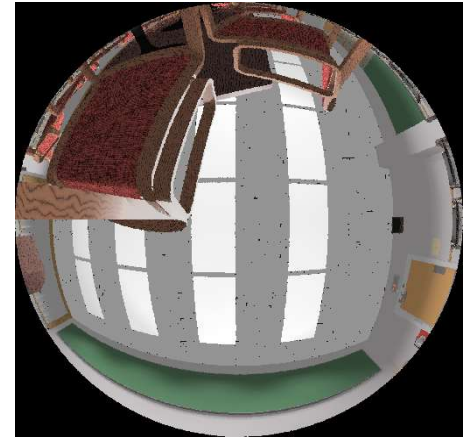
```
    return E;
```



# Výpočet nepřímé irradiance

$E = \text{SampleHemisphere}(\mathbf{x}) ;$

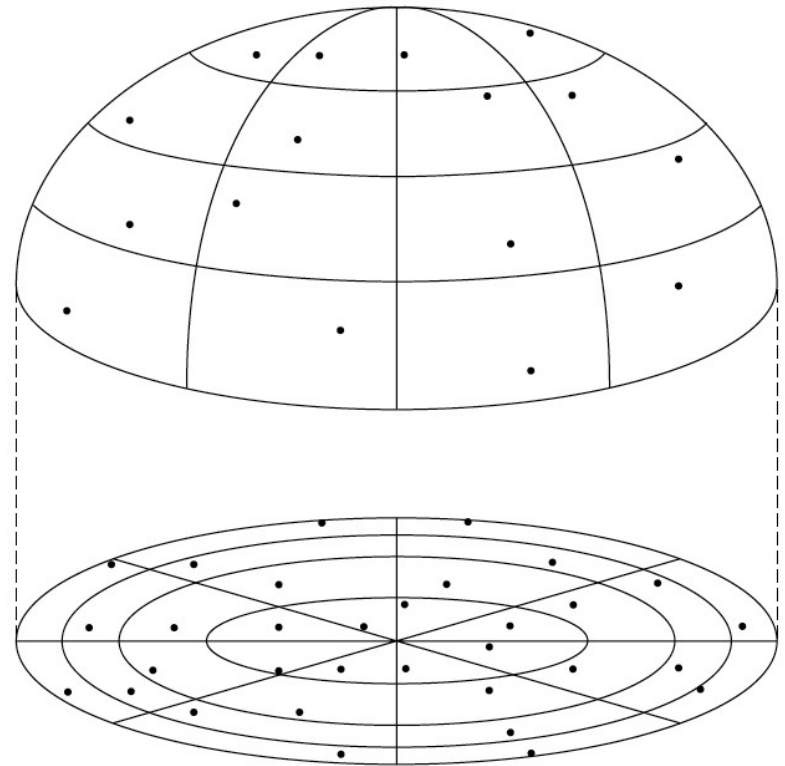
- Vrhni 50-2000 sekundárních paprsků v hemisféře (počet zadán uživatelem)
- Tam, kde protnou scénu, spočítej osvětlení
  - buď dotazem do globální fotonové mapy
  - nebo rekurzivním trasování rozprostřených paprsků
  - Nezapočítávej zdroje přímého osvětlení



# Výpočet nepřímé irradiance

$E = \text{SampleHemisphere}(\mathbf{x}) ;$

- Stratified Monte Carlo  
vzorkování na hemisféře
  - Rozdělení hemisféry na buňky podle elevace a azimutu
  - V každé buňce vyber náhodný směr a vrhni paprsek





# Výpočet nepřímé irradiance

$E = \text{SampleHemisphere}(\mathbf{x}) ;$

$$E_{ind} = \iint L_{ind}(\theta_i, \phi_i) \cos \theta_i \sin \theta_i d\theta_i d\phi_i$$

$$E = \left( \frac{\pi}{M \cdot N} \right) \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} L_{j,k}$$

$L_{j,k}$  is the indirect radiance in the direction  $(\theta_j, \phi_k) = \left( \text{asin} \sqrt{\frac{j+X_j}{M}}, 2\pi \frac{k+Y_k}{N} \right)$

$X_j, Y_k \dots$  rovnoměrně rozdělená náhodná čísla,  $R(0,1)$

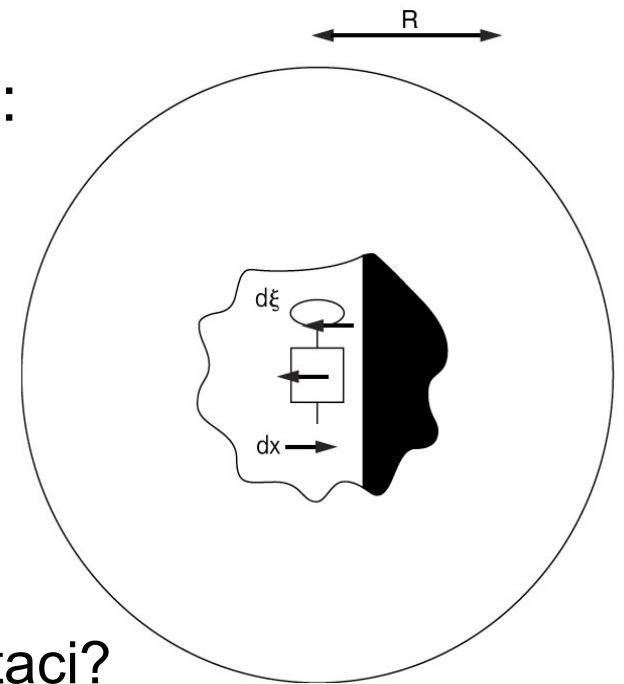
$M \dots$  počet dělení podle elevace ( $\theta$ )

$N \dots$  počet dělení podle azimutu ( $\phi$ )



# Jak daleko lze přepoužít záznam?

- $E(\mathbf{x})$  se mění pomalu  $\Rightarrow$  interpoluj více
- $E(\mathbf{x})$  se mění rychle  $\Rightarrow$  interpoluj méně
- Jak rychle se může irradiance měnit v daném bodě?
- Odpověď: analýza nejhoršího případu: rozdělená koule (split sphere)
  - Polovina koule bez světla, polovina se světlem
  - Bod uprostřed koule
  - Normála směrem k rozdělení světlo/nesvětlo
  - Jaký je gradient  $E(\mathbf{x})$  při translaci a rotaci?



# Jak daleko lze přepoužít záznam?

- Výsledek analýzy na rozdělené kouli:
  - (není důležité si pamatovat)

$$\varepsilon \leq \left| \frac{\partial E}{\partial x}(x - x_0) + \frac{\partial E}{\partial \xi}(\xi - \xi_0) \right|$$

$$\varepsilon \leq \frac{4E}{\pi R} |x - x_0| + E |\xi - \xi_0|$$

$$\varepsilon(\vec{P}) \leq \frac{4}{\pi} E \frac{\|\vec{P} - \vec{P}_0\|}{R_0} + E \sqrt{2 - 2\vec{N}(\vec{P}) \cdot \vec{N}(\vec{P}_0)}$$

$R_0$  = average distance to surfaces at  $\vec{P}_0$

# Interpolace

InterpolateFromCache ( $\mathbf{x}$ ,  $\mathbf{E}$ )

Vážený průměr hodnot z množiny  $S$ .

$$E(\vec{P}) = \frac{\sum_{i \in S} w_i(\vec{P}) E_i(\vec{P})}{\sum_{i \in S} w_i(\vec{P})}$$

Vahou je převrácená hodnota max. možného gradientu (odvozeno z rozdělené koule).

$$w_i(\vec{P}) = \frac{1}{\frac{\|\vec{P} - \vec{P}_i\|}{R_i} + \sqrt{1 - \vec{N}(\vec{P}) \cdot \vec{N}(\vec{P}_i)}}$$

Pro interpolaci se použijí pouze záznamy s vahou přesahující zadaný práh  $a$ .

$$S = \{i : w_i(\vec{P}) > 1/a\}$$

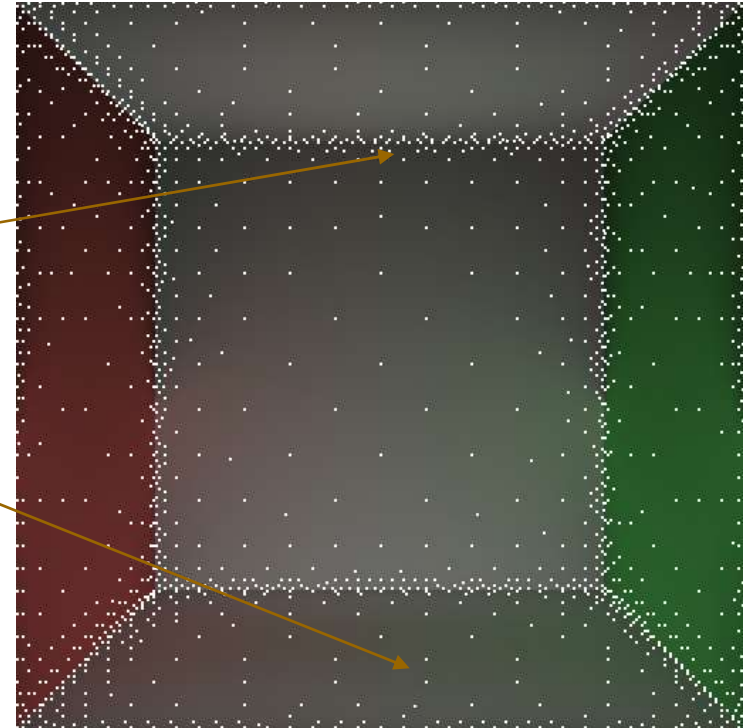
$a$  je parametr algoritmu (0,1 – 0,35)

$R_i$  ... spočítá se jako harmonický průměr délek paprsků při vzorkování hemisféry a uloží se záznamem do cache.

# Rozmístění záznamů v prostoru

- **Blízko geometrie**
  - hustší vzorkování
    - geometrie = zdroj nepřímého osvětlení

- **Otevřené prostory**
  - řidší vzorkování



- Vyplývá z interpolačního schématu
  - záznamy daleko od geometrie mají velké  $R_i$  a lze je tedy přepoužít daleko





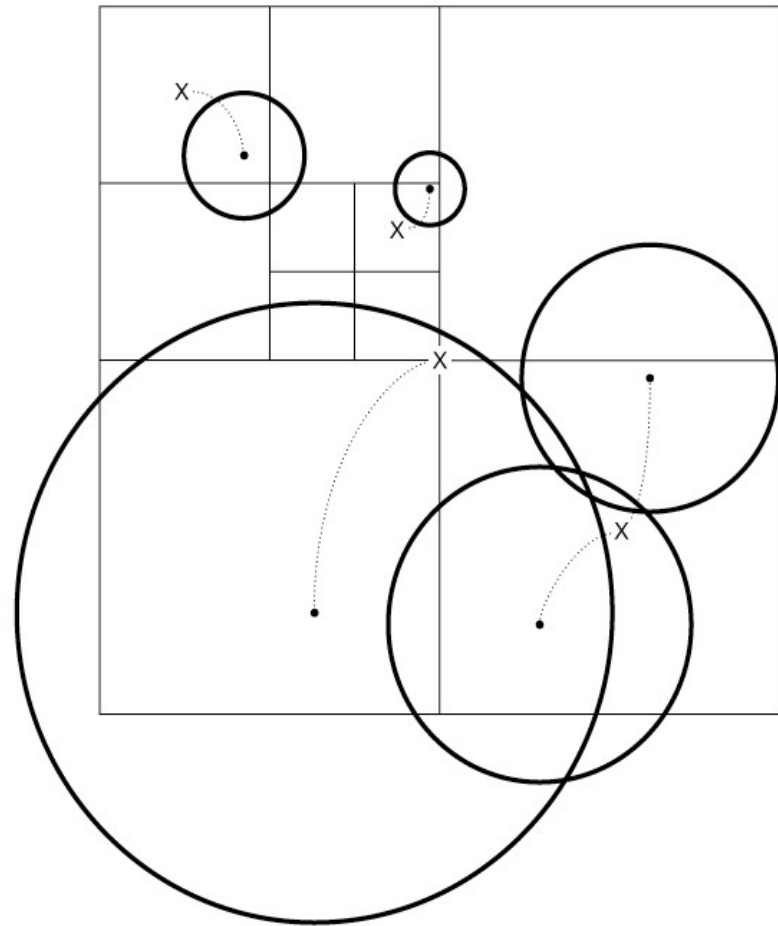
Červené koule = záznamy v cache, jinde se interpoluje.



# Uložení záznamů v cache - Octree

`InsertIntoCache (E, x) ;`

- Uzel obsahuje ty záznamy, pro něž je platný rádius  $a \cdot R_i$  menší než šířka uzlu
- Hledání platných záznamů pro interpolaci – procházení do hloubky



# Záznam v irradiance cache

```
struct indirect_irradiance_value {  
    float P[3];           /* position in space */  
    float N[3];           /* normal direction */  
    float R;               /* validity radius */  
    COLOR E;              /* computed irradiance value */  
    float dP[3];          /* gradient wrt. position */  
    float dN[3];          /* gradient wrt. direction */  
};
```

Gradients na dalších slajdech...

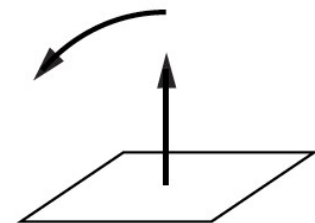
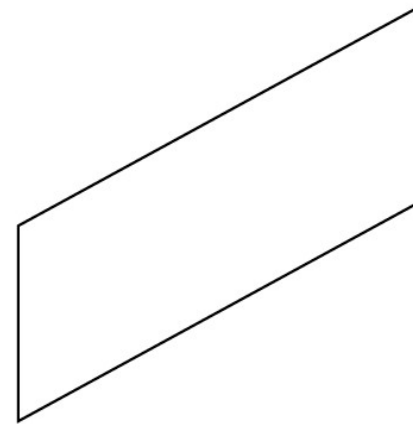


# Gradienty irradiance

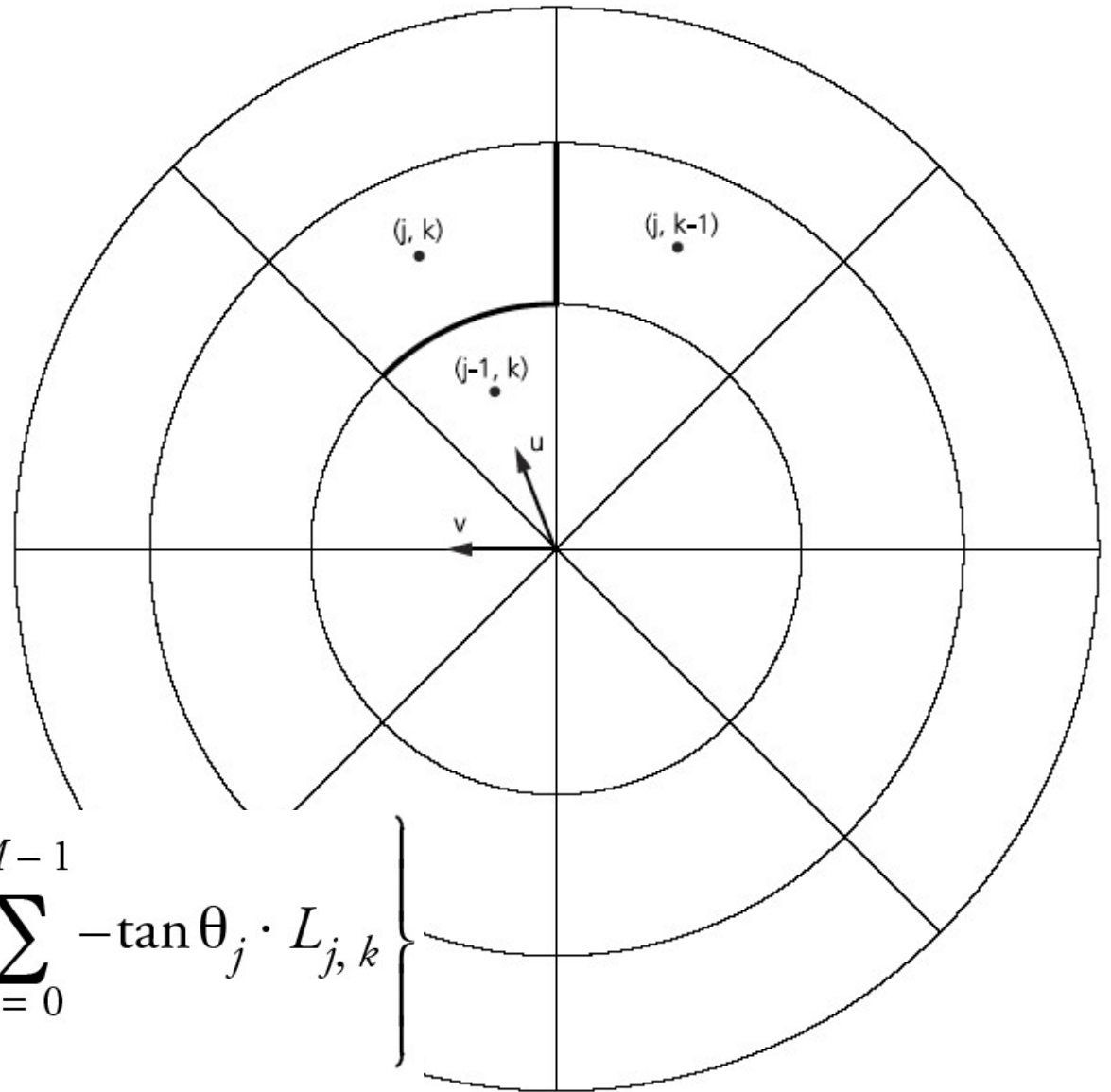
- Zásadní pro kvalitu obrázku
  - bez nich viditelné diskontinuity způsobené interpolací
- Vypočteny z radiancí při vzorkování hemisféry
  - Tj. pro jejich výpočet není potřeba žádné vrhaní paprsků navíc – jsou téměř zadarmo
- Uloženy se záznamem v cache
- Použity při interpolaci

# Rotační gradient

- Při rotaci plošky přichází více či méně světla z druhého objektu
- Lze odhadnout ze vzorků při vzorkování hemisféry



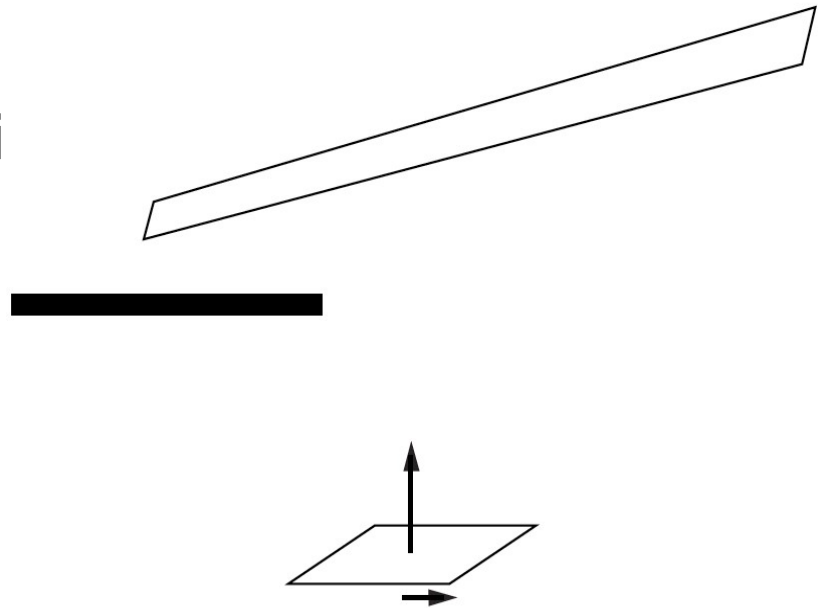
# Vzorec pro rotační gradient



$$\vec{\nabla}_r E = \frac{\pi}{M \cdot N} \sum_{k=0}^{N-1} \left\{ \hat{v}_k \sum_{j=0}^{M-1} -\tan \theta_j \cdot L_{j, k} \right\}$$

# Translační gradient

- Translace plochy odkryje (nebo skryje) část světlého objektu
- Lze odhadnout ze vzorků při vzorkování hemisféry

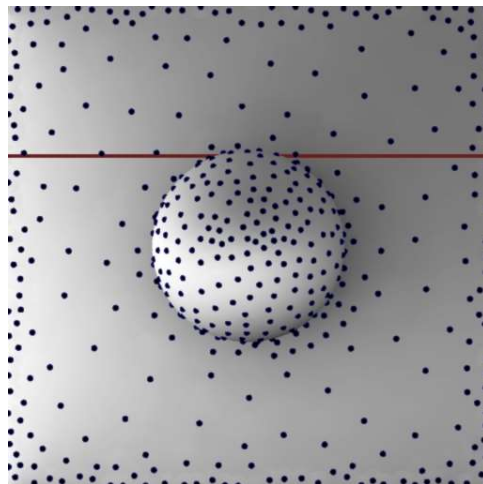
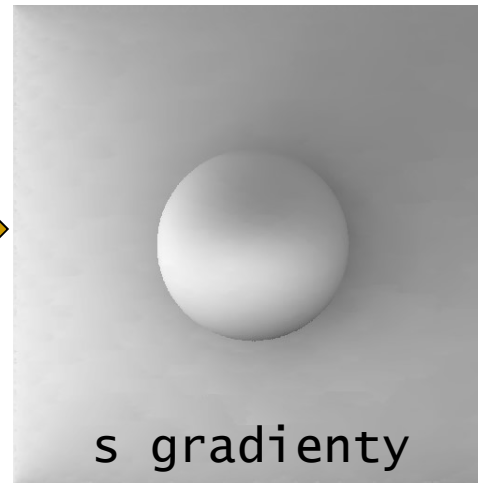


# Interpolace s gradienty

$$E(\vec{P}) = \frac{\sum_{\mathbf{S}} w_i(\vec{P}) \left[ E_i + (\hat{n}_i \times \hat{n}) \cdot \vec{\nabla}_r E_i + (\vec{P} - \vec{P}_i) \cdot \vec{\nabla}_t E_i \right]}{\sum_{\mathbf{S}} w_i(\vec{P})}$$

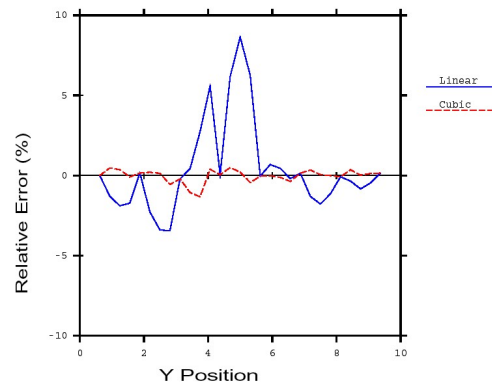
- Váhy  $w_i(\mathbf{P})$  jako bez gradientu
- Gradient modifikuje hodnotu  $E_i$  použitou pro interpolaci

# Výsledky s gradienty a bez

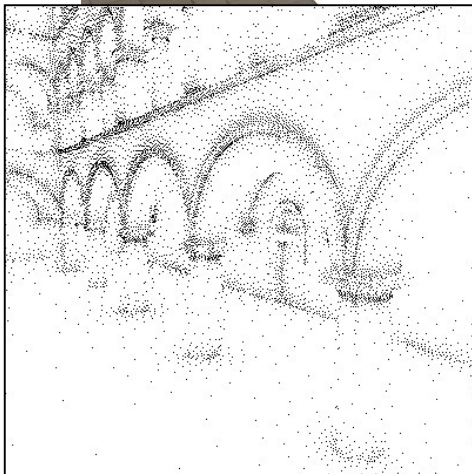
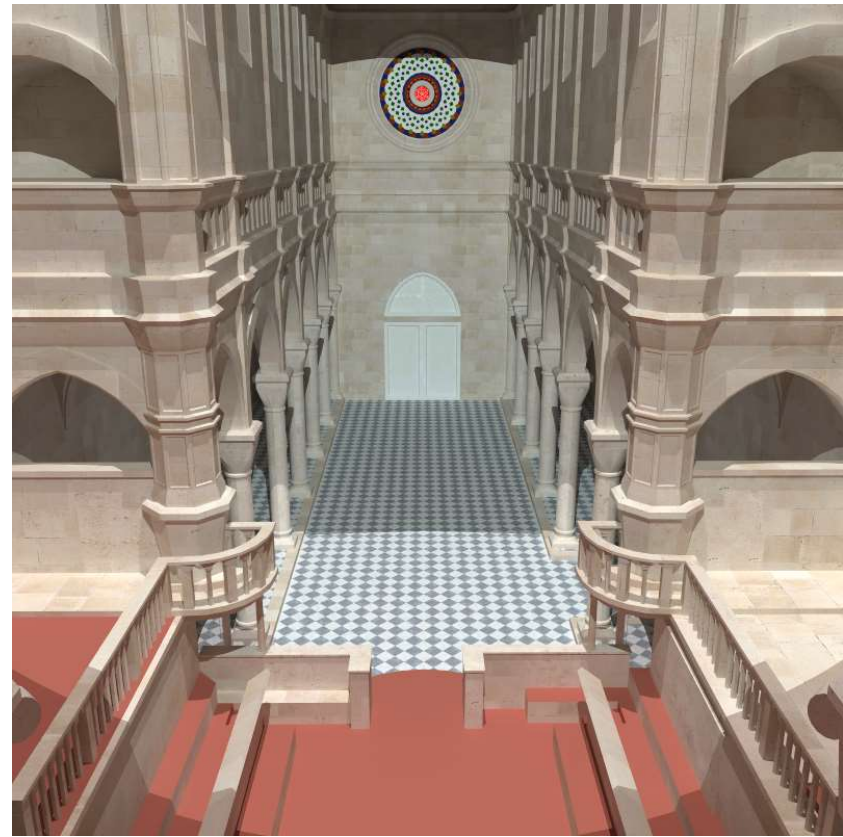
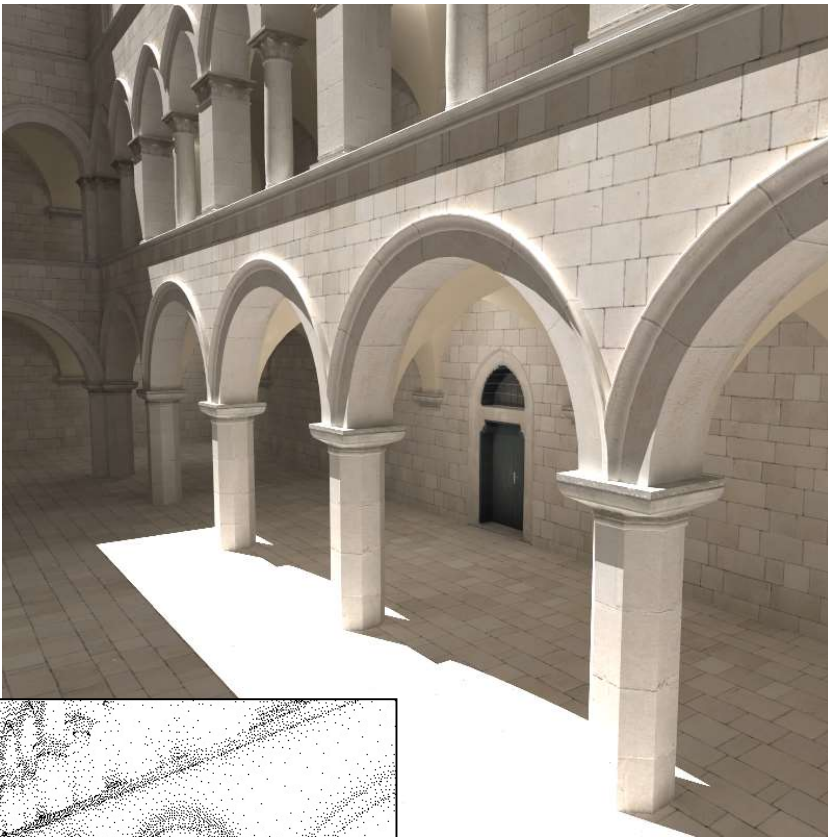


Irradiance Interpolation Error

x=6.875

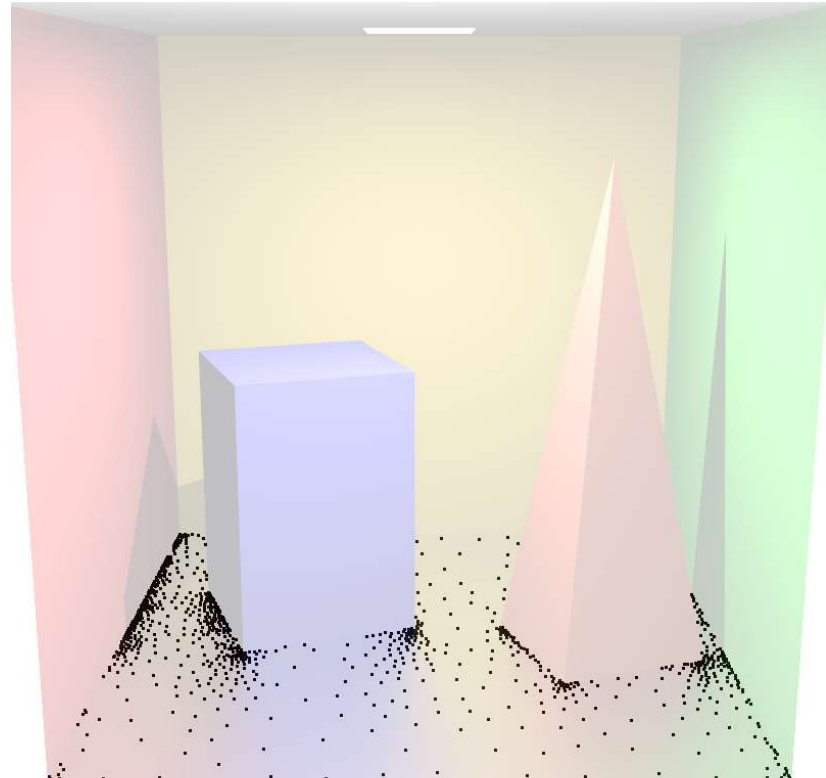
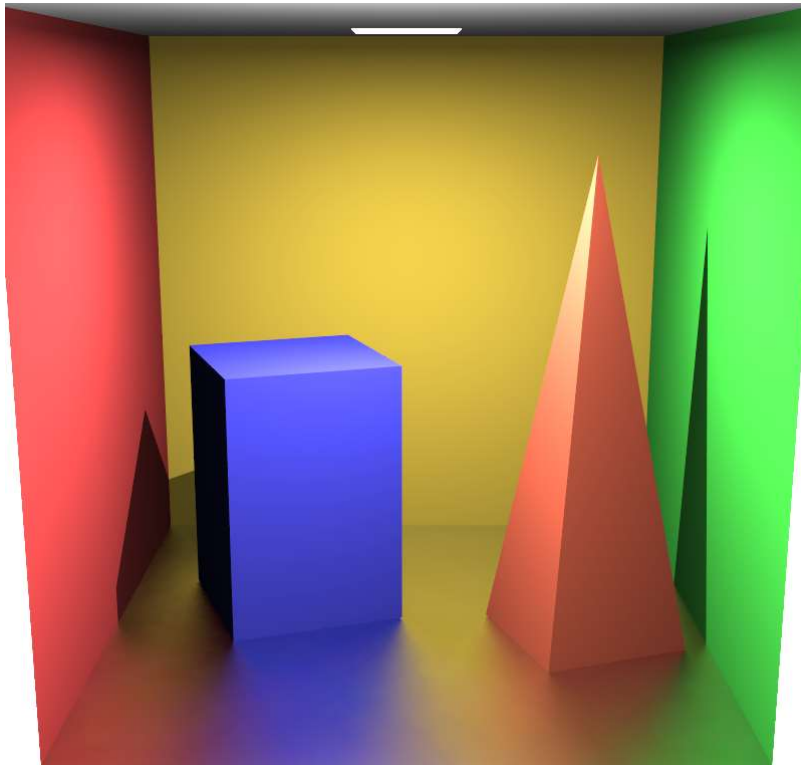


# Příklady



# Irradiance caching lze zobecnit i pro lesklé plochy

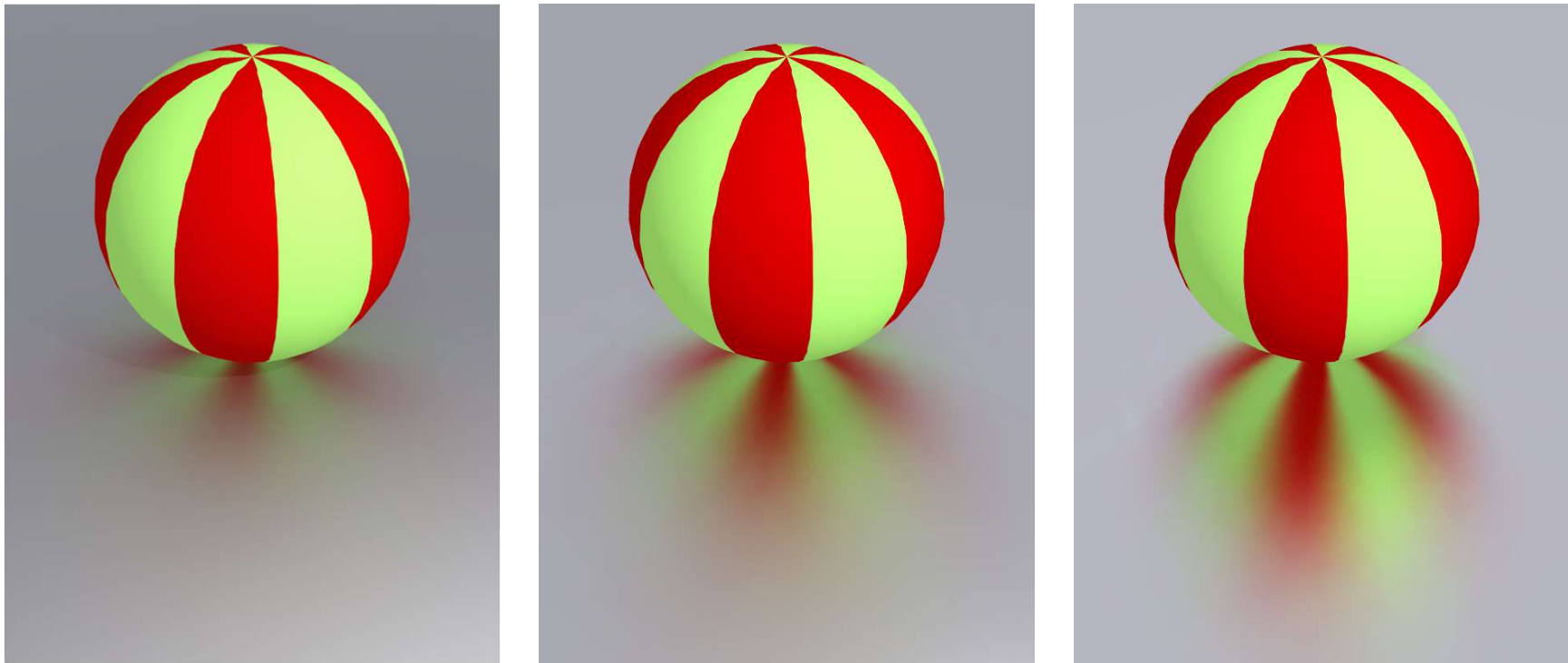
- Radiance caching





# Radiance caching

- Přizpůsobení hustoty záznamů lesklosti povrchu

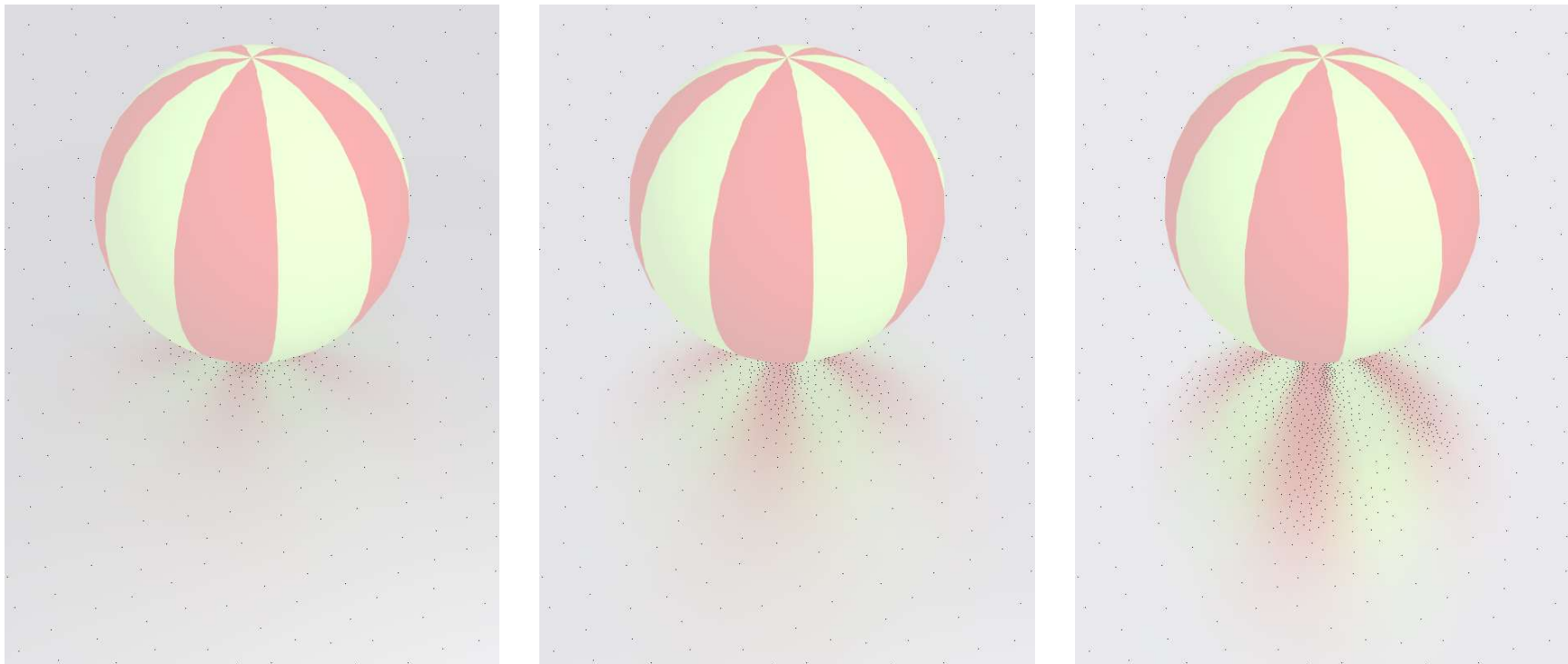


Lesklejší povrch → ostřejší odrazy → větší gradient → více záznamů



# Radiance caching

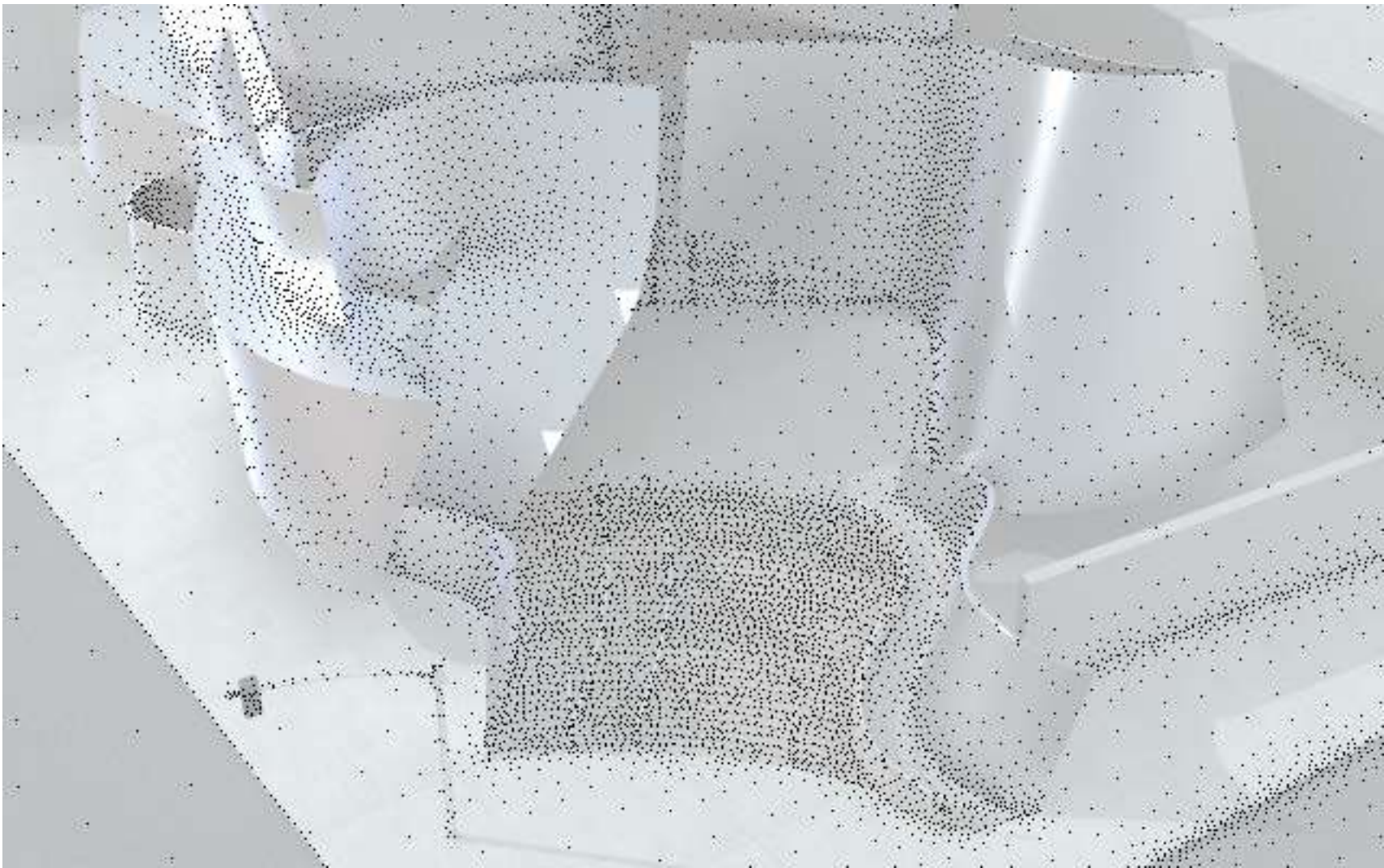
- Adaptation hustoty záznamů na lesklost povrchu



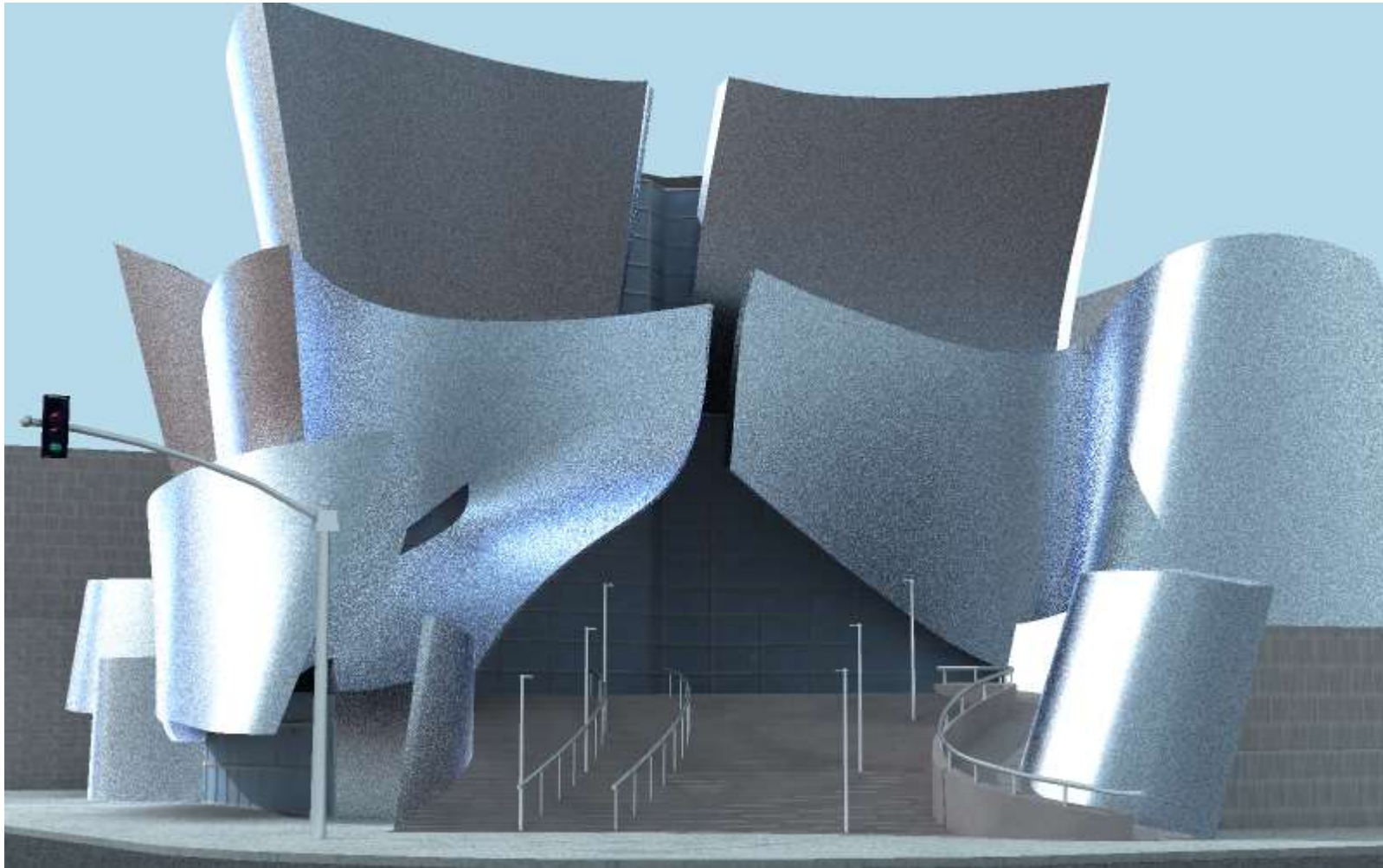
Lesklejší povrch → ostřejší odrazy → větší gradient → více záznamů



# Radiance caching



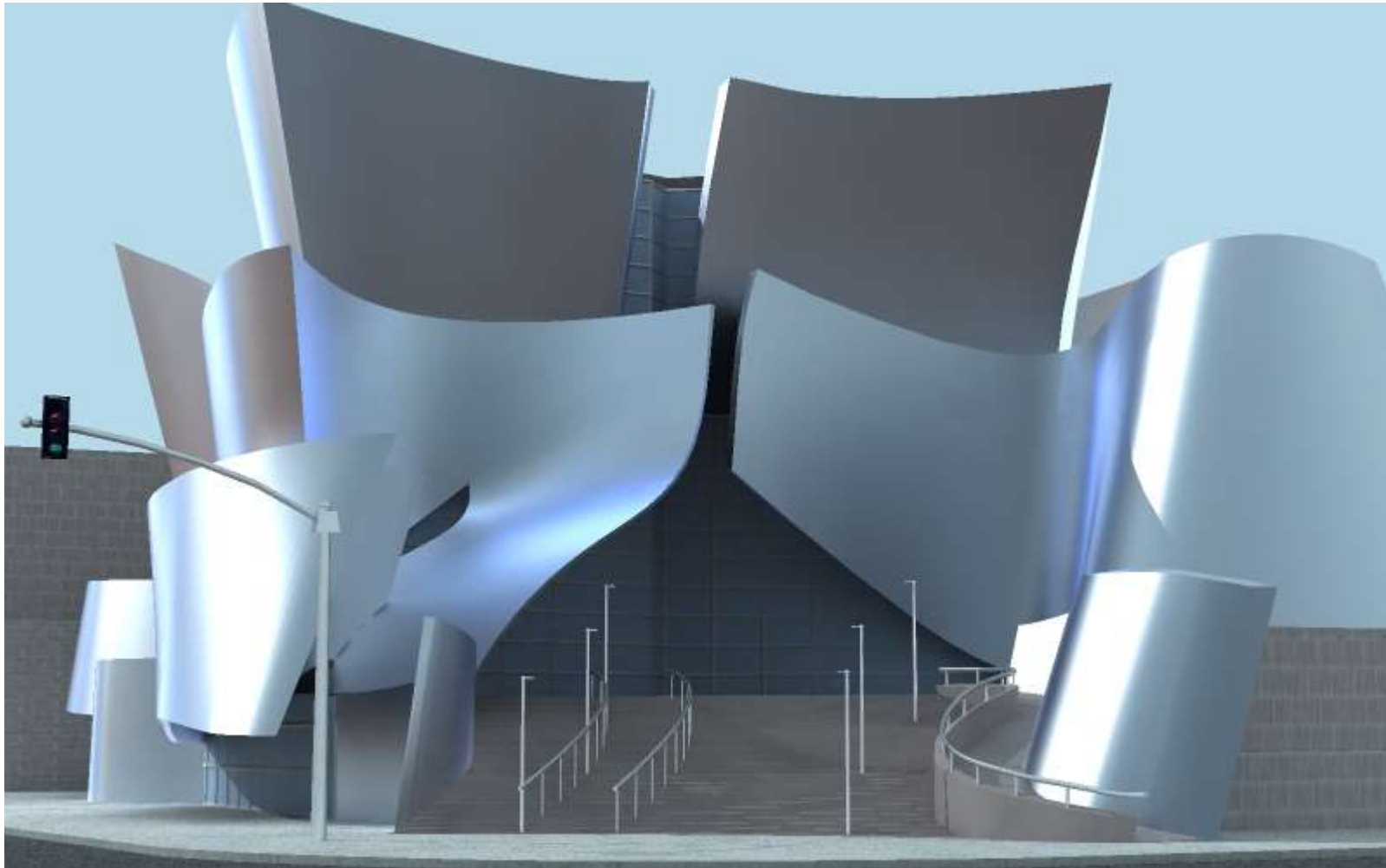
# Radiance Caching vs. Monte Carlo: Stejná doba výpočtu



Monte Carlo Importance Sampling



# Radiance Caching vs. Monte Carlo: Stejná doba výpočtu



Radiance caching

# Fotografie vypadá jinak



Frank Gehry: Walt Disney Concert Hall, Los Angeles, CA

# Závěr

- Fotonové mapy a irradiance caching byly donedávna nejdůležitější a stále jsou velmi používané základní algoritmy pro výpočet globálního osvětlení v současné produkci (filmový průmysl a reklamy).
- Důvod:
  - žádná omezení geometrie scény
    - fotonová mapa i irradiance cache jsou nezávislé na reprezentaci geometrie
    - z toho plyná dobrá škálovatelnost pro obrovské scény
- Fotonové mapy potřebují final gathering pro kvalitní výsledky
- Bez irradiance cachingu by byl final gathering příliš pomalý
- Kombinace fotonové mapy + irradiance caching je vhodné za předpokladu většiny difúzních ploch.
- Některé novější konsistentní algoritmy (mapy paprsků, relaxace fotonů [EG 2009] umožňují ještě rychlejší výpočet), ale zatím se nepoužívají v praxi. Dnes se většinou používá sledování cest.