

# **Urychlování výpočtu osvětlení pomocí metody ReSTIR**

Radek Melčák

Semestrální práce

Vedoucí práce: Ing. Tomáš Fabián, Ph.D.

Ostrava, 2025

# Obsah

<b>Seznam použitých symbolů a zkratek</b>	<b>3</b>
<b>Seznam obrázků</b>	<b>4</b>
<b>Seznam tabulek</b>	<b>5</b>
<b>1 Úvod</b>	<b>6</b>
<b>2 Možnosti výpočtu osvětlení scény</b>	<b>7</b>
2.1 Zobrazovací rovnice . . . . .	7
2.2 Integrace metodou Monte Carlo — strategie vzorkování . . . . .	8
<b>3 Algoritmus ReSTIR</b>	<b>13</b>
3.1 Reservoir-based spatiotemporal importance resampling . . . . .	13
3.2 Průchody algoritmu . . . . .	15
3.3 MIS váhové funkce . . . . .	18
3.4 ReSTIR pro přímé osvětlení . . . . .	22
3.5 ReSTIR pro nepřímé osvětlení . . . . .	22
<b>4 Implementace</b>	<b>23</b>
4.1 Použité technologie . . . . .	23
4.2 Klíčové body implementace . . . . .	24
4.3 Zhodnocení výstupů . . . . .	25
<b>5 Závěr</b>	<b>31</b>
<b>Literatura</b>	<b>32</b>

# **Seznam použitých zkrátek a symbolů**

ReSTIR	– Reservoir-based spatiotemporal importance resampling
PDF	– Funkce hustoty pravděpodobnosti
CDF	– Kumulativní distribuční funkce
MC	– Monte Carlo
IS	– Importance sampling
MIS	– Multiple importance sampling
RIS	– Resampled importance sampling
WRS	– Weighted reservoir sampling
UCW	– Unbiased contribution weight
NEE	– Next event estimation

# **Seznam obrázků**

3.1	Ukázka propagace výstupních vzorků mezi nádržemi, pokud je časové znovupoužití zařazeno před prostorové. . . . .	15
3.2	Ukázka vychýlení vlivem konstantní MIS váhové funkce . . . . .	19
4.1	Ukázka posloupnosti kroků algoritmu ReSTIR v implementaci . . . . .	24
4.2	Ukázka vlivu počtu kandidátů na šum v obrazu (obývací pokoj) . . . . .	26
4.3	Ukázka vlivu počtu sousedů na šum v obrazu (obývací pokoj) . . . . .	27
4.4	Ukázka vlivu jednotlivých průchodů znovupoužití (obývací pokoj) . . . . .	28
4.5	Ukázka rozdílu mezi vykreslovacími metodami (kuchyně) . . . . .	29
4.6	Ukázka rozdílu mezi vykreslovacími metodami (obývací pokoj) . . . . .	30

# **Seznam tabulek**

4.1	Specifikace testovacího systému . . . . .	25
4.2	Parametry testovacích scén . . . . .	25
4.3	Výsledky testu generace prvních kandidátů . . . . .	26
4.4	Výsledky testu prostorového znovupoužití . . . . .	27
4.5	Výsledky testu časového znovupoužití . . . . .	28
4.6	Výsledky testu srovnání s MIS . . . . .	29

# Kapitola 1

## Úvod

V kontextu vizualizace 3D scén tvořených trojúhelníkovými sítěmi se v počítačové grafice standardně využívají dva přístupy – rasterizace a metody založené na rekurzivním sledování paprsků (ray tracing [1]). První z nich je tradičně spojována s případy použití, kdy je preferována interaktivita aplikace, tedy schopnost vykreslení co největšího počtu snímků za sekundu (FPS). Na druhou stranu path tracing [2] umožňuje lépe uchopit úlohy spojené se simulací realistického transportu světla, čímž lze dosáhnout až fotorealistických snímků, často ale na úkor rychlosti vykreslení, a proto se tato metoda doposud spíše uplatňovala při vykreslování statických snímků, například u architektonických vizualizací či animovaných filmů. V posledních letech se však ray tracing začíná objevovat i v interaktivních aplikacích, a to hlavně díky podpoře hardwarové akcelerace na grafických kartách.

Tato práce se zabývá zejména algoritmem ReSTIR (Reservoir-based Spatiotemporal Importance Resampling) [3], jehož hlavní výhodou je možnost nasazení Monte Carlo vykreslovacích technik právě v situacích, kdy je potřeba dosáhnout interaktivní snímkové frekvence (60 FPS a více), jak je tomu například u videoher.<sup>1</sup>

V první části práce je stručně zavedena a popsána samotná zobrazovací rovnice a techniky jejího řešení integrací Monte Carlo. Následně je popsán samotný algoritmus ReSTIR: jeho kroky, hlavní myšlenky a způsoby použití pro řešení přímého i nepřímého osvětlení. V druhé půli práce je představena implementace ukázkové aplikace podporující ReSTIR pro přímé osvětlení, která je následně analyzována a srovnána s referenční standardní implementací využívající MIS.

---

<sup>1</sup>Mezi videohry využívající ReSTIR pro výpočet osvětlení patří například Cyberpunk 2077 [4], Portal RTX [5] nebo Half Life 2 RTX [6].

## Kapitola 2

# Možnosti výpočtu osvětlení scény

Tato kapitola slouží k uvedení výpočtu osvětlení scény. Nejprve je popsána zobrazovací rovnice (nazývána také rovnice transportu světla) a její možné řešení metodou Monte Carlo. Následně jsou vysvětleny Monte Carlo estimátory — od naivního, přes estimátor s vzorkováním podle důležitosti, estimátor s vícečetným vzorkováním podle důležitosti, až po resampled importance sampling estimátor tvůrící jádro samotného algoritmu ReSTIR.

### 2.1 Zobrazovací rovnice

Distribuce radiance v rámci scény lze modelovat pomocí zobrazovací rovnice

$$L(p \rightarrow \vec{\omega}_o) = L_e(p \rightarrow \vec{\omega}_o) + \int_{\Omega} f(p, \vec{\omega}_o \leftrightarrow \vec{\omega}_i) L(p \leftarrow \vec{\omega}_i)(\vec{n}_p \cdot \vec{\omega}_i) d\vec{\omega}_i, \quad [2] \quad (2.1)$$

kde:

- $L(p \rightarrow \vec{\omega}_o)$  je radiance vyzářená odchozím směrem  $\vec{\omega}_o$  z bodu  $p$ ,
- $L_e(p \rightarrow \vec{\omega}_o)$  je emisivita povrchu směrem  $\vec{\omega}_o$  z bodu  $p$ ,
- doména integrantu  $\Omega$  může být buď jednotkovou hemisférou  $H(\vec{n}_p)$  okolo normály, pokud scéna neobsahuje průhledné objekty, nebo jednotkovou sférou  $H^2(p)$ ,
- $f(p, \vec{\omega}_o \leftrightarrow \vec{\omega}_i)$  je BRDF (bidirectional reflectance distribution function), jejíž vyhodnocení pro příchozí  $\vec{\omega}_i$  a odchozí  $\vec{\omega}_o$  směr udává útlum přenesené radiance při odrazu,
- $L(p \leftarrow \vec{\omega}_i)$  je radiance dopadající na bod  $p$  z příchozího směru  $\vec{\omega}_i$ ,
- $\vec{n}_p$  je normála k povrchu v bodě  $p$ .

Jelikož se pracuje s diferenciály prostorových úhlů, je vhodné  $\vec{\omega}_o, \vec{\omega}_i$  modelovat paprsky. Rovnice má rekurzivní charakter a pro počet zanoření výpočtu  $L(p \leftarrow \vec{\omega}_i)$  větší než 1 popisuje kromě přímého také nepřímé osvětlení.

Aplikací vztahu pro převedení diferenciálu prostorového úhlu na diferenciál plochy

$$d\vec{\omega} = \frac{\cos \phi \, dA}{r^2} \quad (2.2)$$

lze přejít do plošného tvaru zobrazovací rovnice, čímž je integrační doména převedena na povrch scény  $A$  [2]:

$$L(p \rightarrow \vec{\omega}_o) = L_e(p \rightarrow \vec{\omega}_o) + \int_A f(p' \rightarrow p \rightarrow \vec{\omega}_o) L(p' \rightarrow p) G(p \leftrightarrow p') V(p \leftrightarrow p') (\vec{n}_p \cdot \vec{\omega}_i) \, dA_{p'}. \quad (2.3)$$

Funkce  $V(p \leftrightarrow p')$  testuje viditelnost mezi body  $p$  a  $p'$  — pokud se mezi nimi nachází jiný objekt, vrací 0, jinak 1. U angulární formy zobrazovací rovnice (2.1) tato funkce není potřeba, jelikož paprsky vyslané pro vyhodnocení  $L(p \leftarrow \vec{\omega}_i)$  mohou trefit pouze takové body  $p'$ , které jsou přímo viditelné z bodu  $p$ . Funkce  $G(p \leftrightarrow p')$  realizuje výpočet geometrického faktoru vztahem 2.2 jako:

$$d\vec{\omega}_i(p) = \frac{\vec{n}_{p'} \cdot (-\vec{\omega}_i)}{\|p' - p\|^2} \, dA_{p'}. \quad (2.4)$$

## 2.2 Integrace metodou Monte Carlo — strategie vzorkování

Jednou z hlavních překážek při výpočtu zobrazovací rovnice je fakt, že integrál obsažený uvnitř nelze, kromě velmi specifických scénářů, vypočít analyticky v uzavřené formě. Všechny body scény se obecně nachází v jiných podmínkách osvětlení, mohou být popsány různými modely materiálů s odlišnými parametry odrazivosti, lesklosti nebo průhlednosti.

### 2.2.1 Monte Carlo estimátor

Tuto obtíž lze obejít pomocí Monte Carlo estimátoru, s nímž je libovolný integrál ve tvaru

$$F = \int_a^b f(x) dx \quad (2.5)$$

možné numericky approximovat jako

$$F \approx F_N = \frac{b-a}{N} \sum_{i=1}^N f(x_i), \quad [7] \quad (2.6)$$

kde  $N$  je počet nezávislých vzorků  $x_i \in (a, b)$  vybíraných s uniformní pravděpodobností. Lze dokázat, že střední hodnota  $E(F_N)$  se rovná hodnotě integrálu. V kontextu vykreslení scény je takový estimátor zaveden každému pixelu obrazu [7]. Pro angulární tvar zobrazovací rovnice (2.1) jsou vzorkovány směry  $\omega_i \in (H(\vec{n}_p) \text{ nebo } H^2(p))$ , zatímco u plošné formy (2.3) jsou vzorkovány body  $p'$  na

povrchu scény s nenulovou emisivitou.<sup>1</sup> Pro dostatečně velký počet vzorků poté snímek konverguje k pravé hodnotě integrálu.

Přístup uniformního vzorkování je však suboptimální v mnoha případech. Například u velmi lesklého povrchu se nezřídka stává, že je vybrán takový směr  $\omega_i$ , který po vstupu do BRDF funkce vyústí ve velmi malý příspěvek radience. Patologický případ nastává v momentě, kdy má povrch materiál perfektního zrcadla — BRDF funkce poté odráží 100 % příchozí radience, pokud se úhel dopadu rovná úhlu odrazu, jinak vrací 0. Obdržet přesně takový vzorek je ale ve spojité doméně nemožné.

## 2.2.2 Vzorkování podle důležitosti (importance sampling)

Monte Carlo estimátor lze zavést tak, aby vzorky byly vybírány z libovolné distribuce, u níž je známa funkce hustoty pravděpodobnosti  $p(x_i)$ :

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}. \quad [7] \quad (2.7)$$

Tento přístup se nazývá vzorkování podle důležitosti (importance sampling). Podmínkou je, aby hodnota  $p(x_i)$  byla nenulová v rámci celého nosiče  $f(x_i)$ , jinak by estimátor nebyl nestranný,<sup>2</sup> protože by distribuce nebyla schopna vygenerovat všechny hodnoty  $f(x_i)$ , jež přispívají k approximovanému integrálu. Čím více pak distribuce vzorků  $x_i$  kopíruje rozdělení hodnot  $f(x_i)$ , tím častěji jsou vybírány takové vzorky, u nichž je pravděpodobnější, že do snímku přinesou více světla, což výrazně zrychluje jeho konvergenci. U objektů s odlišnými typy materiálů a jejich parametry pak lze pracovat s různými distribucemi tak, aby byly prioritizovány takové směry  $\vec{\omega}_i$ , s nimiž výpočet BRDF vrací vysoké hodnoty.<sup>3</sup> Pro plošnou formu zobrazovací rovnice lze vzorky generovat například vzhledem k ploše emisivních těles. V praxi je nutné pro zvolenou distribuci umět nejen vyhodnotit funkci hustoty pravděpodobnosti, ale také vzhledem k ní vzorky generovat, třeba pomocí inverze kumulativní distribuční funkce (CDF).

Problém u estimátoru se vzorkováním podle důležitosti nastává, pokud je distribuce vzorků zvolena špatně, tedy neodpovídá skutečnému rozdělení hodnot  $f(x_i)$ , čímž dochází k častému výskytu „nedůležitých“ (malých) hodnot  $f(x_i)$ . V kontextu vykreslení scény není zaručeno, že zvolená vzorkovací strategie bude vždy optimální: například vzorkování vzhledem k BRDF u méně lesklých povrchů selhává, pokud zdroje světla po promítnutí na hemisféru okolo bodu  $p$  zabírají velmi malou plochu hemisféry — pravděpodobnost vygenerování vzorku směřujícímu ke světlu bude taktéž velmi malá. V tomto případě by bylo vhodnější vzít v potaz plošnou formu zobrazovací rovnice a

---

<sup>1</sup>Teoreticky je korektní integrovat přes povrch celé scény, ale body  $p'$  s nulovou emisivitou budou mít nulové  $L(p \leftarrow p')$ .

<sup>2</sup>Estimátor není nestranný, pokud i s počtem vzorků  $N$  blížícím se limitně k nekonečnu se jeho hodnota nerovná skutečné hodnotě approximovaného integrálu.

<sup>3</sup>U příkladu s materiélem perfektního zrcadla pak distribuce vždy vygeneruje směr  $\vec{\omega}_i$  tak, aby se úhel dopadu rovnal úhlu odrazu a hodnota  $p(x_i)$  bude 1.

vzorky vybírat z emisivních těles. U vysoce lesklých povrchů je zase vhodnější ponechat vzorkování vzhledem k BRDF (viz patologický příklad perfektního zrcadla uvedený výše).

### 2.2.3 Multiple importance sampling

Multiple importance sampling [8] (dále MIS) je metoda dovolující kombinovat více vzorkovacích strategií s odlišnými distribucemi v rámci jednoho estimátoru, který nabývá podoby

$$F = \sum_{i=1}^N \frac{1}{N_i} \sum_{j=1}^{N_i} m_i(x_{i,j}) \frac{f(x_{i,j})}{p_i(x_{i,j})}, \quad [8] \quad (2.8)$$

kde:

- $N$  je počet vzorkovacích strategií,
- $N_i$  značí počet vzorků generovaných strategií  $i$ ,
- $x_{i,j}$  je  $j$ -tý vzorek vygenerovaný strategií  $i$ ,
- $m_i(x_{i,j})$  je váhová funkce pro strategii  $i$ ,
- $p_i(x_{i,j})$  je funkce hustoty pravděpodobnosti pro  $i$ -tou vzorkovací strategii.

Takový estimátor nestranně approximuje integrál za předpokladu, že váhová funkce  $m_i$  splňuje tyto podmínky:

- $\sum_{i=1}^N m_i(x) = 1$  vždy, když  $f(x) \neq 0$ ,
- $m_i(x) = 0$  vždy, když  $p_i(x) = 0$ .

Musí tedy platit, že v každém bodě nosiče funkce  $f(x)$  je alespoň jedna z distribucí vzorkovacích strategií nenulová.

Oproti estimátoru se vzorkováním důležitosti, u něhož je velmi obtížné zajistit jedinou distribuci, která je vhodná ve všech případech, je MIS estimátor robustnější: ve „slabých místech“ jedné distribuce bude funkci  $f(x)$  vzorkovat jiná, jejíž vzorky zde budou mít větší pravděpodobnost výskytu. Tímto způsobem lze vyřešit problém zmíněný na konci předchozí podkapitoly — při výpočtu osvětlení se zároveň vygenerují vzorky vzhledem k BRDF i emisivním plochám, přičemž se naváží funkcemi  $m_i(x)$ .<sup>4</sup>

Pokud je vzorek z  $i$ -té vzorkovací strategie vygenerován s velmi nízkou hodnotou  $p_i(x_{i,j})$ , měla by váhová funkce  $m_i(x_{i,j})$  mít také nízkou hodnotu.<sup>5</sup> Naopak u vyšší hodnoty funkce hustoty pravděpodobnosti by váhová funkce měla mít vyšší hodnoty. Jedna podoba váhové funkce, která je

---

<sup>4</sup>V praxi je také vhodné přidat i vzorkování vzhledem k environmentální mapě, pokud je ve scéně využívána [7].

<sup>5</sup>Je žádoucí, aby podíl v estimátoru „nevystřelil“ v extrémně vysokou hodnotu, což by zvýšilo jeho rozptyl a navýšilo potřebný počet vzorků ke konvergenci.

dokazatelně téměř optimální [8], se nazývá balanční heuristika (balance heuristic):

$$m_i(x) = \frac{p_i(x)}{\sum_j^N p_j(x)}. \quad [8] \quad (2.9)$$

Pro daný vzorek  $x$  z distribuce  $i$  je potřeba umět vyhodnotit pravděpodobnosti, s jakými by byl vygenerován ostatními strategiemi. Tato heuristika může být použita i při výpočtu převzorkovacích vah pro algoritmus ReSTIR [4]. Rychlejší konvergence lze dosáhnout s využitím mocninné heuristiky (power heuristic) [8] nebo i s takovou formulací váhové funkce, jež dovoluje záporné hodnoty [9]. Při implementaci je nutné před vyhodnocením  $m_i(x)$  převést všechny hodnoty PDF do stejné míry.

## 2.2.4 Resampled importance sampling

Ideální distribuce pro generaci vzorků během approximace zobrazovací rovnice je samotné rozdělení příchozí radience nad bodem  $p$ . To však není, jak již bylo zmíněno výše, obecně možné. Za předpokladu využití inverze CDF by bylo nutné znát PDF tohoto rozdělení, tedy znát hodnotu approximovaného integrálu předem.

Metoda resampled importance sampling („převzorkované vzorkování podle důležitosti“, dále RIS) [10] umožňuje uchopit vzorky vygenerované z prvotní, méně kvalitní, ale snadno vzorkovatelné distribuce  $p$  a převzorkovat je tak, aby jejich hustota approximovalo žádanou distribuci  $\hat{p}$ ,<sup>6</sup> která nemusí být normalizovaná ani přímo vzorkovatelná (např. inverzí CDF). Vzorky z distribuce  $\hat{p}$  se poté použijí pro vyhodnocení funkce  $f(x)$  v rámci MC estimátoru. Proces převzorkování probíhá následovně:

1. vygenerování  $M$  vzorků (kandidátů)  $X = \{x_1, x_2, \dots, x_m\}$  distribuce  $p$ ,
2. pro všechny vzorky z  $X$  jsou vypočteny váhy  $W = \{w_1, w_2, \dots, w_m\}$ ,
3. z  $X$  je vybrán výstupní vzorek  $y$  vzhledem k vahám  $W$ .

Pokud jsou váhy stanoveny jako  $w_i = \frac{\hat{p}(x_i)}{p(x_i)}$ , tak budou výstupní vzorky distribuovány vzhledem k  $\hat{p}$ .  $M$  vystupuje jako interpolační parametr — pro hodnoty blížící se k nekonečnu jsou výstupní vzorky rozděleny podle  $\hat{p}$ , zatímco  $M = 1$  je ponechá rozdělené podle  $p$ . Nicméně případ s nekonečnem je nemožné numericky realizovat, a proto nebude distribuce výstupních vzorku přesně odpovídat  $\hat{p}$ , vždy bude závislá na konkrétním výběru  $X$ . Estimátor

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(y_i)}{\hat{p}(y_i)} \quad (2.10)$$

je pro splnění nestrannosti nutné korigovat převzorkovací váhovou funkcí, jež zároveň hustotu  $\hat{p}$  normalizuje:

---

<sup>6</sup>Funkce  $\hat{p}(x)$  je taktéž nazývána cílovou funkcí [4].

$$w(X_i, y_i) = \frac{1}{M} \sum_{j=1}^M w_{i,j} = \frac{1}{M} \sum_{j=1}^M \frac{\hat{p}(x_{i,j})}{p(x_{i,j})}. \quad (2.11)$$

Výsledný estimátor má následující podobu:

$$F_N = \frac{1}{N} \sum_{i=1}^N \left( \frac{f(y_i)}{\hat{p}(y_i)} \cdot \frac{1}{M} \sum_{j=1}^M \frac{\hat{p}(x_{i,j})}{p(x_{i,j})} \right). \quad [10] \quad (2.12)$$

Pokud se při výpočtu osvětlení jako  $\hat{p}(x)$  zvolí samotná zobr. rovnice (2.1, 2.3), všech  $M$  vzorků bude převzorkováno tak, jako by pocházely ze samotného rozdělení radiance okolo bodu  $p$ . V praxi je vhodnější  $\hat{p}(x)$  zjednodušit (např. nevyhodnocovat funkci viditelnosti), protože jinak by cena integrace jednoho z  $N$  požadovaných výsledných vzorků mohla být příliš vysoká — viz nutnost  $M$ -krát vyhodnotit zobr. rovnici [10]. Ostatně tento přístup volí i algoritmus ReSTIR [4].

# Kapitola 3

## Algoritmus ReSTIR

V této kapitole bude popsán algoritmus ReSTIR — jeho hlavní myšlenky, zasazení do kontextu vzhledem k předchozí kapitole a možnosti použití k výpočtu jak přímého, tak i nepřímého osvětlení.

### 3.1 Reservoir-based spatiotemporal importance resampling

U běžného RIS přístupu je hlavním limitujícím faktorem volba  $M$  (počet vzorků pro převzorkování). Algoritmus ReSTIR [11, 4] se snaží docílit efektu jeho navýšení bez nutnosti generování více vzorků, a to s pomocí více kol převzorkování: podobné výstupní vzorky z předešlého kola se využijí jako vstup do dalšího. Řetězením převzorkování tak lze kontinuálně vylepšovat rozdělení integrovaných vzorků a přiblížit se velmi blízko skutečné distribuci integrandu. Teoreticky lze algoritmus ReSTIR aplikovat u libovolných problémů, jejichž řešení jsou založena na MC vzorkování, nicméně následující části textu jsou omezeny na kontext počítacové grafiky. Navíc tento algoritmus dokáže udržet svou kvalitu i během pohybu kamery či scény [4].

ReSTIR se snaží z již vygenerovaných vzorků „vytěžit“ co nejvíce — při převzorkování se pro daný pixel pracuje i se vzorky okolních pixelů (prostorové znovupoužití), a také vzorky z předešlých snímků (časové znovupoužití). Vhodné seřazení těchto kroků má za následek, že estimátor při převzorkování pracuje s exponenciálně se zvyšujícím počtem kvalitních vzorků.

U algoritmu ReSTIR jsou dále kritická dvě místa: podoba převzorkovací váhové funkce (rovnice 2.11) a výpočet výrazu zastupujícího  $\frac{f(x)}{p(x)}$  v MC estimátoru. Výraz  $\frac{f(x)}{p(x)}$  je rozložen na  $f(x) \cdot W_y$ , kde hodnota  $W_y = \frac{1}{p(x)}$  je nazývána unbiased contribution weight („nestranná váha příspěvku“, dále UCW). U RIS estimátoru je potom UCW vypočtena jako

$$W_y = \frac{1}{\hat{p}(y)} \cdot \sum_{i=1}^M w(x_i). \quad [4] \quad (3.1)$$

Chybějící část  $\frac{1}{M}$  je přesunuta do výpočtu převzorkovací váhové funkce

$$w(x_i) = \frac{1}{M} \cdot \frac{\hat{p}(x_i)}{p(x_i)}, \quad [4] \quad (3.2)$$

kde zastupuje roli MIS váhové funkce (viz kapitola 2.2.3). V případě, že vzorky pocházejí z odlišných vzorkovacích strategií, což je i případ prostorového a časového převzorkování, je výraz  $\frac{1}{M}$  nahrazen skutečnou váhovou funkcí  $m_i(x)$  [4].

Důležitá je také volba funkce  $\hat{p}(x)$ . Použití plné podoby zobrazovací rovnice zajistí, že výsledné vzorky budou distribuovány vzhledem k ní. To ale může být vzhledem ke zvýšené výpočetní náročnosti nepraktické. V praxi se často vynechává funkce viditelnosti [4, 11]. Nicméně při vyhodnocování MIS váhové funkce je pak nutné zajistit, aby váha byla nulová vždy, když je nulová i hodna funkce  $f(x)$  (viz kapitola 3.3).

### 3.1.1 Weighted reservoir sampling

Weighted reservoir sampling (dále WRS) [4] je algoritmus pro realizaci náhodného výběru  $k$  prvků z datového proudu neznámé délky. Zatímco u běžného RIS je potřeba pro  $M$  původních vzorků vyhradit paměť o velikosti  $M$ , WRS dovoluje zpracovat libovolný počet vzorků s konstantní pamětí.

---

**Algoritmus 1:** Definice struktury nádrže (Reservoir)

---

```

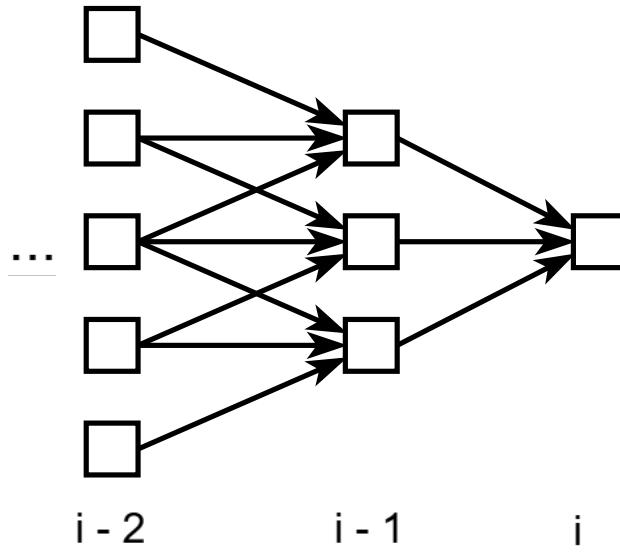
1 Struct Reservoir:
2   Sample Y
3   float wsum
4   float W
5   Function add_sample(Sample x, float w):
6     wsum += w
7     if (random(0, 1) < w/wsum) :
8       Y = x

```

---

Nádrž (reservoir) obsahuje kromě výstupního vzorku navíc sumu výstupů z převzorkovací váhové funkce (rovnice 3.2), již lze dosadit do výpočtu UCW (rovnice 3.1). Během přidání nového vzorku  $x$  je probabilisticky rozhodnuto, zda se stane novým výstupním vzorkem  $Y$ , nebo jestli bude zahozen, a to vzhledem k podílu jeho váhy a sumy vah dosavadně prozkoumaných vzorků. Lze dokázat, že toto chování je ekvivalentní s postupem převzorkování popsaném v kapitole 2.2.4 [4].

U algoritmu ReSTIR je zpravidla vyčleněna nádrž pro každý pixel obrazu, jež po převzorkování prvních kandidátů uchovává výstupní vzorek. Později, během průchodu znovupoužití, probíhá převzorkování nad výstupními vzorky nádrží: vytvoří se nová nádrž, do níž se pomocí funkce v algoritmu 1 přidají vzorky ostatních nádrží.



Obrázek 3.1: Ukázka propagace výstupních vzorků mezi nádržemi, pokud je časové znovupoužití zařazeno před prostorové. Při převzorkování u časového znovupoužití mezi nádržemi  $i$  a  $i-1$  obsahuje starší nádrž informace z prostorového znovupoužití. hodnota  $r.W$  nádrže  $i$  poté s narůstající délkou historie zohledňuje více a více vzorků.

## 3.2 Průchody algoritmu

ReSTIR pracuje ve více průchodech. Každý průchod má vlastní nastavitelné parametry, s jejichž manipulací lze upravovat výkonnostní profil algoritmu, kvalitu jednotlivých snímků nebo také rychlosť reakce na měnící se polohu kamery či objektů ve scéně. Při vykreslení jednoho snímku lze průchody prostorového a časového znovupoužití aplikovat vícekrát, a to dokonce v libovolném pořadí, čímž je navýšen počet kroků převzorkování, avšak za cenu zvýšené korelace mezi vstupními vzorky [4]. Je doporučováno zařadit časové znovupoužití před prostorovým (viz obrázek 3.1). Následující podkapitoly stručně popisují jednotlivé průchody.

### 3.2.1 Generace prvotních vzorků

Před převzorkováním mezi nádržemi je potřeba je nejprve naplnit: vygenerovat prvotní vzorky a převzorkovat je. Vzorky je vhodné generovat z více vzorkovacích strategií metodou MIS, v nádrži pak s vysokou pravděpodobností zůstane vzorek z takové strategie, která přináší do estimátoru největší příspěvek. Čím více úvodních vzorků bude prozkoumáno, tím kvalitnější distribuci bude mít výstupní vzorek. Jelikož není obecně dopředu známo, jak bude vypadat projekce povrchu scény na pixely obrazu, je vhodné zařadit více vzorkovacích strategií. Každá nádrž pak bude pravděpodobně obsahovat vzorek z takové strategie, která nejlépe vyhovuje danému materiálu. Pokud má každá

strategie  $M$  vlastních vzorků, MIS váhová funkce má tvar:

$$m_i(x) = \frac{p_i(x)}{\sum_{j=0}^M M_j p_j(x)}. \quad [4, 8] \quad (3.3)$$

---

**Algoritmus 2:** Proces generace prvotních vzorků

---

```

1 Function generate_sample():
2   Reservoir r
3   foreach str  $\in$  strategies do
4     for j = 1 to  $M_{str}$  do
5       Sample s = take_sample_from_str()
6       w =  $m_{str}(s) \cdot \hat{p}(s) \cdot s.W$ 
7       r.addSample(s, w)
8   r.W =  $1/\hat{p}(r.Y) \cdot r.w_{sum}$ 
```

---

Člen  $s.W$  v algoritmu 2 zastupuje  $\frac{1}{p_{str}(x)}$ , kde funkce hustoty pravděpodobnosti  $p_{str}(x)$  je v tomto případě známá a vyhodnotitelná. Hodnota  $r.W$ , jež je použita při dalších průchodech, reprezentuje „vylepšenou“ distribuci výstupního vzorku. Její konkrétní realizace vždy závisí na podobě vstupních vzorků, a proto neodpovídají distribuci  $\hat{p}$ . To však není nutné — stačí pouze, aby hodnota  $r.W$  distribuci  $\hat{p}$  kopírovala v průměru [4, 10].

### 3.2.2 Prostorové znovupoužití

U prostorového znovupoužití pro daný pixel  $p$  je nejprve inicializována prázdná nádrž. Následně je přijato  $M$  nádrží vybraných sousedních pixelů (včetně pixelu  $p$ ), jimiž je tato nová nádrž naplněna. Důvod převzorkování z blízkých pixelů je založen na presumpci podobnosti distribuce radience nad blízkými body scény. Pokud nádrž pixelu  $p$  původně neměla validní výstupní vzorek, je velmi pravděpodobné, že adoptuje některý ze sousedních. I pokud zůstane vzorek nenahrazen, výsledná hodnota  $r.W$  bude více odpovídat rozdělení hodnot  $\hat{p}$ . Se zvyšujícím se  $M$  poroste i kvalita převzorkování. Výběr sousedů nesmí záviset na obsahu sousedních nádrží, jinak výsledný estimátor bude vychýlený. Technika výběru sousedních nádrží ovlivňuje charakteristiku šumu v obrazu: vhodné je například uniformní vzorkování disku o zvoleném poloměru nebo využití sekvence s nízkou diskrepancí [4]. Možné MIS váhové funkce  $m_{space}(x)$  jsou blíže popsány v kapitole 3.3.

---

**Algoritmus 3:** Prostorové znovupoužití

---

```
1 Function spatial_reuse(Pixel p):
2     Reservoir r
3     for i = 1 to M do
4         p_n = get_neighbor_index()
5         Sample s = sample_at_pixel(p_n)
6         w = m_space(s) ·  $\hat{p}_p(s)$  · s.W
7         r.addSample(s, w)
8     r.W = 1/ $\hat{p}(r.Y)$  · r.wsum
```

---

### 3.2.3 Časové znovupoužití

Časové znovupoužití je postaveno na stejném principu jako prostorové znovupoužití, pouze se pro pixel  $p$  liší zdroj přidaných vzorků pro převzorkování: zde pochází z nádrže pixelu  $p'$  určující pozici v minulém snímku, kde  $p$  a  $p'$  překrývají totožný, nebo velmi blízký bod povrchu scény. Opět, obdobně jako u prostorového znovupoužití, se předpokládá, že distribuce radiance nad tímto bodem bude v obou časech velmi podobná, ne-li totožná,<sup>1</sup> což z korespondujících nádrží činí ideální kandidáty pro převzorkování [4, 11].

---

**Algoritmus 4:** Časové znovupoužití

---

```
1 Function temporal_reuse(Pixel p):
2     Reservoir r
3     Sample s = sample_at_pixel(p)
4     w = m_time(s) ·  $\hat{p}_p(s)$  · s.W
5     r.addSample(s, w)
6     p' = get_pixel_index_last_frame(p)
7     Sample s' = sample_at_pixel_last_frame(p')
8     w' = m_time(s') ·  $\hat{p}_p(s')$  · s'.W
9     r.addSample(s', w')
10    r.W = 1/ $\hat{p}(r.Y)$  · r.wsum
```

---

Časové znovupoužití vyžaduje uchování nádrží z minulého snímku v paměti. Během vyhodnocení nestranné varianty MIS váhové funkce  $m_{time}(x)$  (popsáno v kapitole 3.3) je také nutné vyhodnotit  $\hat{p}(x)$  pro vzorek z aktuálního snímku tak, jako by pocházel ze snímku předešlého a naopak, tj. je potřeba pro minulý snímek uchovávat informace o parametrech viditelného povrchu scény, např.

---

<sup>1</sup>V případě, kdy scéna není animovaná.

v G-bufferu. Pokud je vyžadována nestrannost u animovaných scén, je potřeba zajistit přístup k akcelerační struktuře minulého snímku, což zvyšuje nároky na paměť [4].

Pixel, potažmo nádrž  $p'$  lze identifikovat reprojekcí z pixelu  $p$  pomocí inverze kamerové matice minulého snímku. Pokud vlivem pohybu kamery či objektů scény dojde mezi snímky k okluzi nebo disokluzi, je potřeba zvážit, zda vzorky z minulého snímku nezahodit, jelikož  $p$  a  $p'$  již nebudou sdílet totožný bod ve scéně — distribuce popsané nádržemi se budou lišit, což může vést k suboptimálnímu určení výstupního vzorku.

### 3.3 MIS váhové funkce

MIS váhové funkce  $m_{space}(x)$  a  $m_{time}(x)$  mohou nabývat několika různých podob. Nejjednodušší varianta sice neučiní výsledný estimátor nestranným, nicméně její výpočetní nenáročnost může být k užitku, pokud je preferována vysoká snímková frekvence. Nestrannost lze pak zaručit korekcí vychýlení, nebo pokročilejšími formulacemi těchto funkcí. V této kapitole jsou popsány možné realizace MIS váhových funkcí, jejich vlastnosti, úskalí a možné postupy korekce.

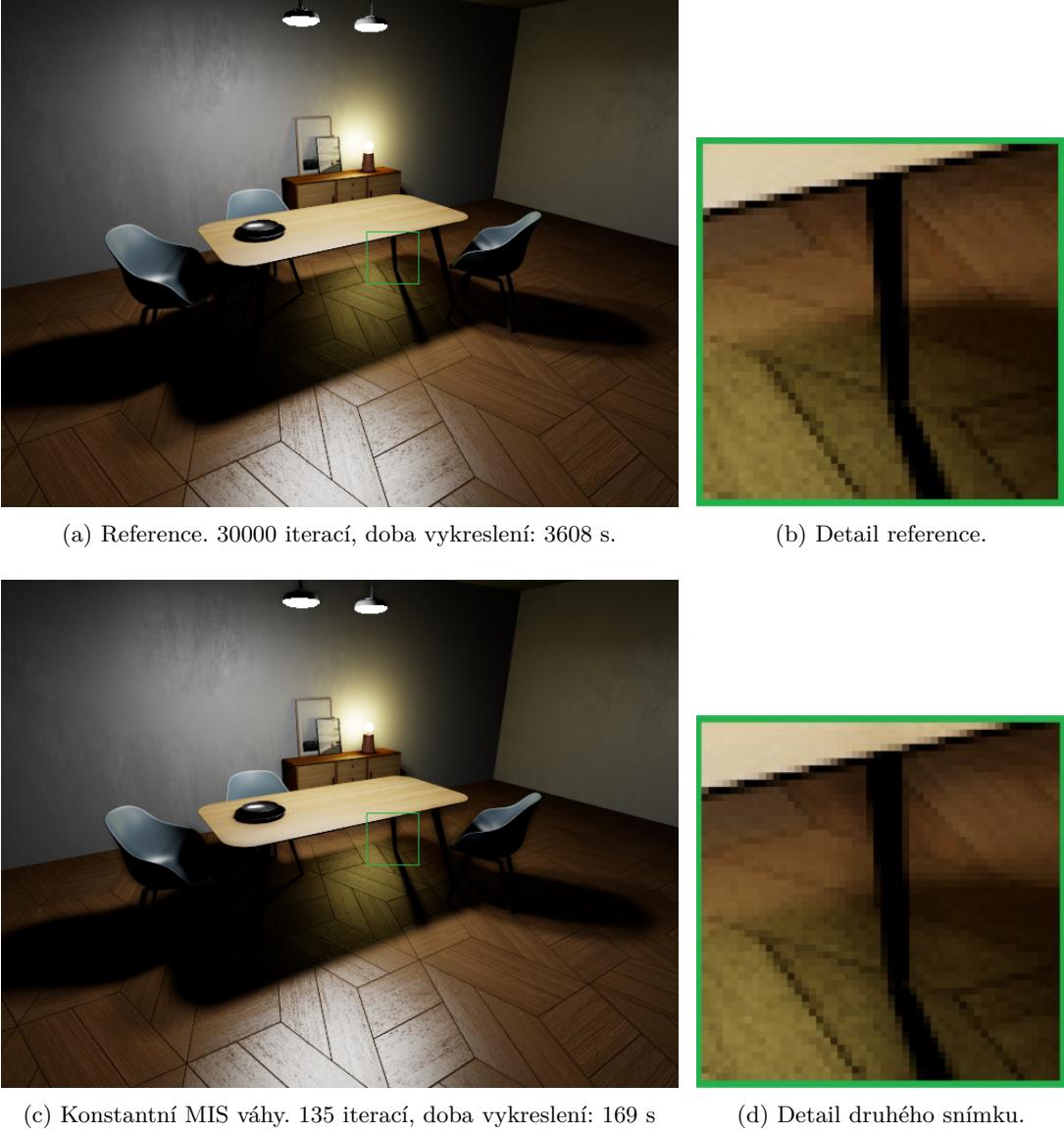
#### 3.3.1 Konstantní váhová funkce a korekce vychýlení

Jako triviální MIS váhovou funkci lze považovat člen  $\frac{1}{M}$  z převzorkovací váhové funkce (vztah 3.2):

$$m(x) = \frac{1}{M}, \quad [4, 11] \quad (3.4)$$

kde  $M$  je počet nádrží pro převzorkování. Hlavní výhodou je již zmíněná zanedbatelná výpočetní náročnost, navíc lze stejnou hodnotu použít pro převzorkovací váhovou funkci všech  $M$  dotčených nádrží.

Nevýhodou je nesplnění podmínek MIS váhové funkce (viz kapitola 2.2.3), což vede k porušení nestrannosti. Příkladem může být prostorové znovupoužití v následujícím scénáři: pokud výstupní vzorek vybraného sousedního pixelu  $p'$  nemůže být vygenerován samotným pixelem  $p$  (třeba v důsledku odlišností geometrie scény vedoucí k zastínění), pak by MIS váhová funkce měla být nulová pro tento sousední pixel, pro ostatní by však součet měl být stále 1. Nicméně jako důsledek konstantní váhy  $\frac{1}{M}$  nebude příspěvek ostatních, validních sousedů vracet žádaný součet, což způsobí nekorektní ztmavnutí obrazu (viz obrázek 3.2).



Obrázek 3.2: Ukázka vychýlení vlivem konstantní MIS váhové funkce.

Vychýlení estimátoru lze zmírnit odmítnutím nepodobných sousedů. Například pokud se normály k povrchům mezi pixely  $p$  a  $p'$  liší více, než je stanovená hranice, soused bude zamítnut. Obdobně je možné se rozhodnout na základě hloubek obou pixelů. Nutno podotknout, že takové pixely  $p$ , pro něž bude odmítnuto mnoho sousedů, budou trpět na delší dobu konvergence.

Během výpočtu UCW (rovnice 3.1) po převzorkování pomocí konstantní MIS váhové funkce lze korekcí vychýlení (debiasing) estimátor učinit nestranným. Nejjednodušší takovou korekcí je úprava výpočtu UCW do tvaru

$$W_y = \frac{\frac{1}{Z}}{\frac{1}{M}} \cdot \frac{1}{\hat{p}(y)} \cdot \sum_{i=1}^M w(x_i), \quad [11] \quad (3.5)$$

kde  $\frac{1}{M}$  je doposud popisovaná MIS váhová funkce a  $Z$  je počet sousedů, jejichž vzorky jsou pro pixel  $p$  validní. Pokud zvolená funkce  $\hat{p}(x)$  neobsahuje test viditelnosti, výpočet  $Z$  vyžaduje trasování paprsků pro její zjištění [4]. Časová složitost této metody je  $O(M)$ .

Korekce lze docílit také výpočtem korektní MIS váhy pro výstupní vzorek z převzorkování (v literatuře nazýváno contribution MIS weight [11, 4]). UCW potom nabývá tvaru

$$W_y = \frac{c_s(y)}{\frac{1}{M}} \cdot \frac{1}{\hat{p}(y)} \cdot \sum_{i=1}^M w(x_i), \quad [11] \quad (3.6)$$

kde  $c_s(y)$  je korektní MIS váha výstupního vzorku vypočtena generalizovanou vyrovnanou heuristikou (blíže popsána v následující kapitole). Ačkoli je časová složitost opět  $O(M)$ , výpočetní náročnost této korekční metody je o něco vyšší, což je však vykoupeno spolehlivější redukcí šumu v obraze. Závisí tedy na požadovaném výkonnostním profilu aplikace.

### 3.3.2 Nestranné váhové funkce

Nestrannost estimátoru lze také zajistit přímo, pokud jsou funkce  $m_{space}(x)$  a  $m_{time}(x)$  vhodně formulovány. Standardní vyrovnaná heuristika, jak je použita při generaci prvních kandidátů, však nelze během prostorového a časového znovupoužití využít, jelikož opravdová hustota výstupních vzorků v nádržích je neznámá. Tuto nepříjemnost je možné obejít s využitím generalizované vyrovnané heuristiky [12, 4]

$$m_i(x) = \frac{\tilde{p}_i(x)}{\sum_j^N \tilde{p}_j(x)}, \quad [12, 4] \quad (3.7)$$

kde funkce  $\tilde{p}(x)$  je libovolná funkce splňující podmínky MIS váhové funkce. V praxi lze uvažovat  $\tilde{p}(x) = \hat{p}(x)$  — vzhledem k tomu, že vzorky vstupující do průchodu znovupoužití již musely „přežít“ proces převzorkování, lze očekávat, že budou kvalitní, tj. jejich  $\hat{p}(x)$  bude vracet vysoké hodnoty v místech, kde by byla vysoká i opravdová hustota radience. Tato technika má však zásadní nevýhodu ve své časové složitosti, která je  $O(M^2)$ .<sup>2</sup>

Kvadratickou časovou složitost lze redukovat na lineární využitím párových MIS vah (pairwise MIS) [13, 4]. V této metodě jsou vzorkovací strategie při výpočtu MIS váhy rozděleny na jednu kanonickou a zbylé nekanonické. Kanonická strategie musí pokrývat celou doménu integrantu, zatímco ostatní strategie nemusí. Například v kontextu prostorového znovupoužití jako kanonická strategie slouží nádrž pixelu  $p$ , zatímco nádrže vybraných sousedních pixelů  $p'$  jsou brány jako nekanonické. Při výpočtu MIS vah je potom spočtena vyrovnaná heuristika párově vždy mezi dvojicemi sousední pixel – kanonický pixel. Tvar funkce v defenzivní formě<sup>3</sup> pro kanonický vzorek s indexem  $c$  je následující:

---

<sup>2</sup>Pro každou vybranou sousední nádrž je nutné vyhodnotit  $\hat{p}(x)$  všech ostatních nádrží užastnících se kola převzorkování.

<sup>3</sup>Funkce párových MIS vah existují v několika variantách. Defenzivní forma brání případům, kdy je kanonickému vzorku přiřazena příliš nízká váha [4, 13].

$$m_c(x) = \frac{1}{M} \left( 1 + \sum_{i \neq c}^M \frac{\frac{\hat{p}_c(x)}{M-1}}{\hat{p}_i(x) + \frac{\hat{p}_c(x)}{M-1}} \right), \quad [13, 4] \quad (3.8)$$

zatímco pro nekanonické vzorky má funkce tvar

$$m_{nc}(x) = \frac{1}{M} \cdot \frac{\hat{p}_i(x)}{\hat{p}_i(x) + \frac{\hat{p}_c(x)}{M-1}}. \quad [13, 4] \quad (3.9)$$

I přestože je časová složitost výpočtu párových MIS vah v  $O(M)$ , stále je výpočetně o něco náročnější, než konstantní váhová funkce s korekcí vychýlení. Párové MIS váhy jsou však vhodnější při použití algoritmu ReSTIR pro výpočet nepřímého osvětlení, kde výrazněji snižují šum, a to na úroveň srovnatelnou s variantou, kdy je použita generalizovaná vyrovnaná heuristika s kvadratickou složitostí [4].

### 3.3.3 Váhy důvěryhodnosti

MIS váhové funkce, jak jsou uvedeny v kapitolách výše, opomíjejí zohlednit historii — pro nádrž, jež prošla kombinovaným kolem časoprostorového znovupoužití, bude v dalším snímku během časového znovupoužití v MIS váze zohledněna pouze hodnota  $\hat{p}(x)$ . Taková nádrž přitom pravděpodobně lépe popisuje distribuci radiance, a proto je žádoucí jí přiřadit vyšší váhu.

Přidáním parametru důvěryhodnosti [11, 4] (confidence)  $c$  do definice nádrže lze sledovat, kolik vzorků bylo nádrží prozkoumáno, na základě čehož lze potom při výpočtu MIS váhových funkcí prioritizovat nádrže s větším počtem vzorků. Generalizovaná vyrovnaná heuristika má potom tvar

$$m_i(x) = \frac{c_i \tilde{p}_i(x)}{\sum_j^N c_j \tilde{p}_j(x)}. \quad [4] \quad (3.10)$$

Obdobně existuje je i varianta pro párové MIS váhy. Podoba výpočtu pro kanonický vzorek s indexem  $c$  je následující:

$$m_c(x) = \frac{c_c}{\sum_{j=1}^M c_j} + \sum_{i \neq c}^M \left( \frac{c_i}{\sum_{j=1}^M c_j} \cdot \frac{c_c \hat{p}_c(x)}{\left( \sum_{j \neq c}^M c_j \right) \hat{p}_i(x) + c_c \hat{p}_c(x)} \right), \quad [4] \quad (3.11)$$

zatímco pro nekanonické vzorky má funkce tvar

$$m_{nc}(x) = \frac{c_i}{\sum_{j=1}^M c_j} \cdot \frac{\left( \sum_{j \neq c}^M c_j \right) \hat{p}_i(x)}{\left( \sum_{j \neq c}^M c_j \right) \hat{p}_i(x) + c_c \hat{p}_c(x)}. \quad [4] \quad (3.12)$$

Každý vzorek obsažený v průchodu generování prvních kandidátů inkrementuje důvěryhodnost nádrže o 1. Při převzorkování mezi nádržemi se důvěryhodnost výsledné nádrži rovná součtu důvěryhodností všech účastnících se nádrží. Po několika průchodech časoprostorového znovupoužití

však minulé snímky svou důvěryhodností velmi převáží nové snímky až do bodu, kdy se pravděpodobnost ponechání nových vzorků bude blížit nule. Proto je vhodné po převzorkování důvěryhodnost shora omezit hodnotou  $c_{cap}$  (doporučené rozmezí hodnot je 5-30) [4].

### 3.4 ReSTIR pro přímé osvětlení

Během výpočtu přímého osvětlení jsou cesty paprsků světla omezeny na tři body  $x_0, x_1, x_2$ . Body  $x_0, x_1$  tvoří spojnice mezi bodem na senzoru kamery a bodem ve scéně, jenž byl trefen vystřelením paprsku skrze bod senzoru kamery. Bod  $x_2$  je pak vybrán vzorkováním, z čehož je určena spojnica  $x_1, x_2$ . Zkonstruované cesty  $x_0, x_1, x_2$  jsou pak vystaveny Monte Carlo integraci [4].

První část cesty  $x_0, x_1$  pro každý pixel definuje kontext funkce  $\hat{p}(x)$  — v závislosti na  $x_1$  se odvíjí distribuce radiance nad tímto bodem, která se pro různé  $x_1$  obecně liší. Lze tedy uvažovat, že každý pixel pracuje s vlastní verzí funkce  $\hat{p}_i(x)$ , kde úsečka  $x_{i,0}, x_{i,1}$  je implicitně definována  $i$ -tým pixelem v obraze. Při převzorkování mezi nádržemi odlišných pixelů  $p, p'$  pak záleží, z jakých distribucí byly vygenerovány vzorky nacházející se v jednotlivých nádržích, respektive v jaké míře jsou uchovány hodnoty UCW. Pokud jsou uvedeny v míře prostorových úhlů, pak je nutné při výpočtu převzorkovací váhové funkce (rovnice 3.2) zahrnout jakobián determinantu pro korekci změny hodnoty prostorového úhlu vzorku: když pixel  $p'$  z bodu  $x_{p',1}$  vygeneruje vzorek  $x_{p',2}$ , tak pravděpodobnost vygenerování stejného vzorku z bodu  $x_{p,1}$  bude jiná. V případě uchování UCW v plošné míře tato korekce není nutná, avšak při zahrnutí vzorkování environmentální mapy pro tyto vzorky není možné vyhodnotit geometrický faktor [4, 12].

### 3.5 ReSTIR pro nepřímé osvětlení

K algoritmu ReSTIR existují také rozšíření, jež poskytují možnost kromě přímého osvětlení řetězeně převzorkovat i delší cesty paprsků, a to v podobě ReSTIR GI [13] a ReSTIR PT [12]. První z nich při převzorkování mezi cestami  $x_0, x_1, x_2, \dots, x_n$  a  $y_0, y_1, y_2, \dots, y_n$  využije místo dopadu prvního nekamerového paprsku jako bod pro přepojení cesty na posloupnost  $x_0, x_1, y_2, \dots, y_n$ , což umožní suboptimálním nádržím zmocnit se lepších cest sousedů. V případech, kdy je materiál prvního trefeného povrchu velmi spekulární, může tento způsob převzorkování selhat z důvodu nedodržení podmínky, kdy se úhel dopadu musí rovnat úhlu odrazu. U průhledných materiálu se taktéž postupuje obtížně.

ReSTIR PT používá sofistikovanější metody výběru sousedů, jež si automaticky zvládnou vybrat cesty tak, aby byly uspokojeny restrikce kládené na znovupoužití dané materiálovými modely. Více druhů výběru vzorků ke znovupoužití je probabilisticky kombinováno tak, aby při zaručené nestrannosti výsledného obrazu byly vybírány vzorky maximalizující příspěvek radiance [12].

# Kapitola 4

## Implementace

V této kapitole je popsána ukázková aplikace implementující algoritmus ReSTIR pro přímé osvětlení. Nejprve jsou uvedeny použité technologie, následně jsou přiblíženy klíčové body implementace a nakonec jsou předvedeny výstupy algoritmu včetně porovnání se standardním MIS estimátorem.

### 4.1 Použité technologie

Aplikace je napsána v jazyce C++ a využívá funkcionality několika knihoven:

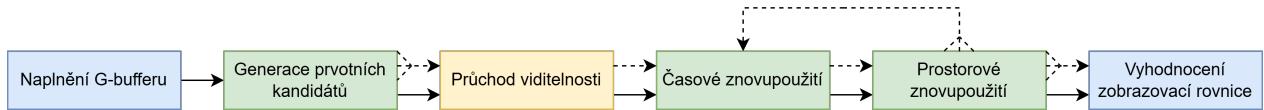
- DirectX 11 [14] jako vizualizační backend,
- Intel Embree [15] pro trasování paprsků na CPU,
- Intel Open Image Denoise [16] pro odšumění obrazů,
- Open Asset Import Library [17] pro načítání 3D modelů,
- FreeImage [18] pro načítání a ukládání obrázků,
- Dear ImGui [19] pro tvorbu základního uživatelského rozhraní,
- GLM [20] pro matematické operace.

Algoritmus ReSTIR byl přidán jako nástavba k projektu do předmětu Počítačová grafika I, v jehož dokončené verzi se již nacházela ray tracer a path tracer s NEE, oba s podporou několika materiálových modelů. Část některých funkcí byla pro realizaci algoritmu ReSTIR zrecyklována, jiné byly pro přehlednost odstraněny. Některé byly ponechány pro účely následného porovnání s algoritmem ReSTIR.

## 4.2 Klíčové body implementace

Aplikace, jak již bylo zmíněno, implementuje algoritmus ReSTIR pro výpočet přímého osvětlení scény. Veškeré vykreslování probíhá na CPU, a to zejména pro jednoduchost a možnosti využití debuggeru při hledání chyb. Pro urychlení výpočtu je využito vícevláknového zpracování, nicméně mezi jednotlivými průchody jsou vlákna vždy synchronizována, aby byla zaručena konzistence dat.

Před samotnými průchody algoritmu ReSTIR jsou do scény nejprve vrženy paprsky pro naplnění G-bufferu, jež pro každý pixel obsahuje informace o povrchu scény nutné k vyhodnocení zobrazovací rovnice. Tyto G-buffery jsou dva: jeden pro aktuální snímek, druhý ze snímku minulého. Stejně tak jsou vyhrazena pole pro nádrže aktuálního a minulého snímku. Při běhu aplikace se pouze mění indexy čtecích a zapisujících bufferů.



Obrázek 4.1: Ukázka posloupnosti kroků algoritmu ReSTIR v implementaci. Přerušované čáry znázorňují tok vzorků mezi průchody.

Generace prvních kandidátů obsahuje volitelné počty vzorků pro vzorkování BRDF a světel. V budoucnu by bylo vhodné aplikaci rozšířit o důležitostní vzorkování environmentální mapy. Dále je možné použít tvar funkce  $\hat{p}(x)$  pro první kandidáty bez testu viditelnosti, k jejíž vyhodnocení dojde až v samostatném průchodu viditelnosti. V ostatních průchodech funkce  $\hat{p}(x)$  vždy viditelnost obsahuje, zejména z důvodu jednoduššího zachování nestrannosti a korektnosti výsledků. V tomto ohledu by aplikace šla dále optimalizovat, jelikož ve speciálních případech lze znovupoužít i viditelnost.

U prostorového znovupoužití jsou nastavitelné následující parametry:

- počet opakování průchodu,
- počet sousedů,
- poloměr disku pro výběr sousedů,
- způsob výpočtu MIS vah,
- zda přeskočit příliš rozdílné sousedy.

Sousedé jsou vždy vybíráni uniformním vzorkováním disku vycentrovaného kolem daného pixelu. Mezi implementované MIS váhové funkce patří:

- konstantní váhová funkce,
- korekce vychýlení faktorem  $\frac{1}{Z}$ ,

- korekce vychýlení s využitím contribution MIS weight,
- generalizovaná vyrovnaná heuristika,
- párové váhové funkce.

U časového znovupoužití je implementována pouze generalizovaná vyrovnaná heuristika, se kterou v případě převzorkování ze dvou nádrží není kvadratická složitost znatelná. Všechny MIS váhové funkce využívají váhy důvěryhodnosti, hodnota  $c_{cap}$  je taktéž nastavitelným parametrem. Je také možné zapnout vizualizaci detekce okluzí a disokluzí během pohybu kamery.

## 4.3 Zhodnocení výstupů

V této kapitole je nejprve uveden kontext k testování výsledků. Následně jsou výstupy algoritmu ReSTIR srovnány s referenční implementací využívající pouze MIS. Sledováno je zejména urychlení konvergence oproti referenci.

### 4.3.1 Testovací kontext

Parametry testovacího systému jsou uvedeny v tabulce 4.1. Implementovaná aplikace pro veškeré výpočty využívá pouze procesor, a proto je rychlosť vykreslování omezena primárně počtem jader a frekvencí CPU, na ostatních komponentách tolik nezáleží.

Tabulka 4.1: Specifikace testovacího systému

CPU	AMD Ryzen™ 5 4600H
GPU	NVIDIA® GeForce RTX™ 2060M 6GB
RAM	16 GB DDR4 3200 MHz
OS	Windows 10

Testovací scény byly zvoleny dvě: kuchyně [21] a obývací pokoj [22]. Jelikož aplikace nepodporuje důležitostní vzorkování environmentální mapy, obě scény jsou v uzavřených místnostech bez vlivu okolního světla. Pro otestování škálovatelnosti aplikace jsou scény vykresleny s odlišným rozlišením. Bližší informace ke scénám jsou uvedeny v tabulce 4.2.

Tabulka 4.2: Parametry testovacích scén

	Počet troj.	Počet emis. troj.	Počet povrchů	rozlišení snímku
Kuchyně	81456	1904	18	$640 \times 480$
Obývací pokoj	580638	6	42	$1280 \times 720$

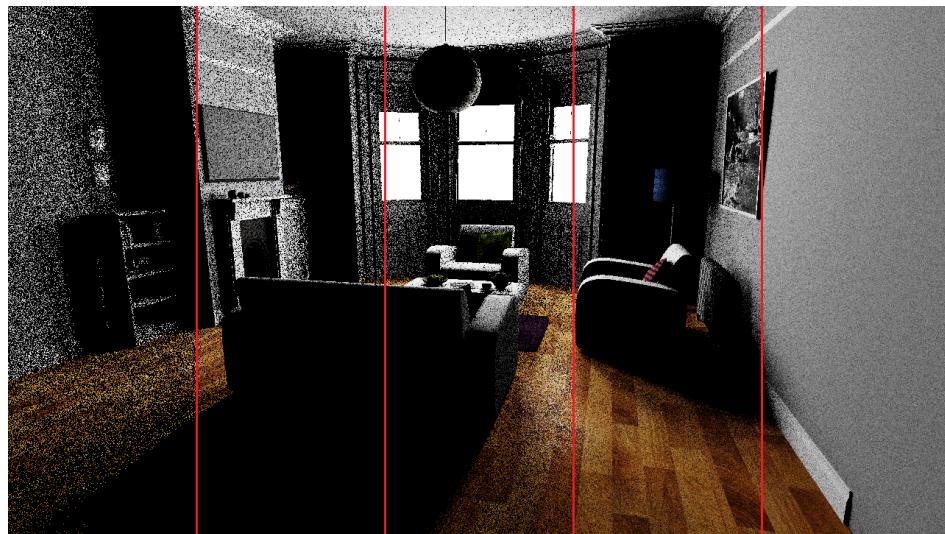
V následujících podkapitolách jsou popsány testy jednotlivých průchodů algoritmu ReSTIR, jejichž cílem je přiblížit, jak změna parametrů průchodů ovlivňuje nejen kvalitu výsledných snímků, ale i výpočetní náročnost. V každém testu jsou pro určité nastavení porovnány časy vykreslení, odchylky od referenčního snímku a vizuální podoba.

#### 4.3.2 Generace prvotních kandidátů

Vzhledem k tomu, že se tento průchod odehrává u algoritmu ReSTIR jako první, má speciální postavení: počet vzorků pro první kolo převzorkování přímo určuje, jak dobře budou rozprostřeny vzorky vstupující do dalších kol. Tento test si klade za cíl změřit, jak se počet prvotních kandidátů promítne do kvality (obrázek 4.2) a výkonu (tabulka 4.3) aplikace. Testováno bylo 5 různých počtů prvotních kandidátů s vypnutými průchody znovupoužití.

Tabulka 4.3: Výsledky testu generace prvotních kandidátů. Hodnoty uvedené v závorkách platí pro scénu obývacího pokoje, mimo ně platí pro kuchyni.

Počet vzorků (plošné, BRDF)	Doba vykreslení (s)	odchylka (RMSE)
1, 1	0.914 (0.379)	80.64 (52.14)
4, 1	1.321 (0.52)	50.13 (28.82)
8, 1	1.806 (0.71)	35.49 (21.24)
16, 1	2.7340 (1.07)	24.77 (15.44)
32, 1	4.622 (1.753)	17.29 (11.35)



Obrazek 4.2: Ukázka vlivu počtu kandidátů na šum v obraze (obývací pokoj). Počty vzorků (plošné, BRDF) zleva doprava: 1,1; 4,1; 8,1; 16,1; 32,1.

Z výsledků je jasné, že zvyšující se počet prvních kandidátů má pozitivní dopad na kvalitu zachyceného obrazu: pokud pixel vygeneruje neplatný vzorek, v nádrži bude pravděpodobně nahrazen jiným, validním vzorkem. Distribuce výstupních vzorků taktéž lépe approximuje distribuci radiance nad každým pixelem. Prozkoumání více vzorků však znamená delší čas vykreslení.

### 4.3.3 Prostorové znovupoužití

Jedním z hlavních parametrů, jež ovlivňuje kvalitu snímku během prostorového znovupoužití, je počet sousedních nádrží pro převzorkování. U vyššího počtu se předpokládá menší šum v obraze a to zejména ze dvou důvodů: vyšší počet vstupních vzorků umožní lépe approximovat distribuci radiance; pokud pixel  $p$  nebude mít na začátku validní vzorek, je pravděpodobné, že si „vypůjčí“ vzorek od souseda. V tomto testu jsou změřeny doby vykreslení a odchylky snímků pro různé počty sousedů. Při generaci prvních kandidátů je vyhodnocen 1 plošný a 1 BRDF vzorek. Mezi nejmenším a největším počtem kandidátů je nárůst času zhruba pětinásobný, pokles odchylky taktéž.

Tabulka 4.4: Výsledky testu prostorového znovupoužití. Hodnoty uvedené v závorkách platí pro scénu obývacího pokoje, mimo ně platí pro kuchyni.

Počet sousedů	Čas (s, generalizované MIS)	Čas (s, párové MIS)	odchylka (RMSE)
1	1.17 (0.534)	1.136 (0.559)	68.98 (42.59)
5	1.979 (1.156)	1.608 (0.851)	45.25 (26.78)
10	3.73 (2.571)	2.107 (1.192)	35.04 (20.9)
15	6.441 (5.3)	2.659 (1.575)	30.14 (18.18)
20	9.943 (7.816)	3.07 (1.946)	27.14 (16.43)



Obrázek 4.3: Ukázka vlivu počtu sousedů na šum v obraze (obývací pokoj). Počty sousedů zleva doprava: 1, 5, 10, 15, 20.

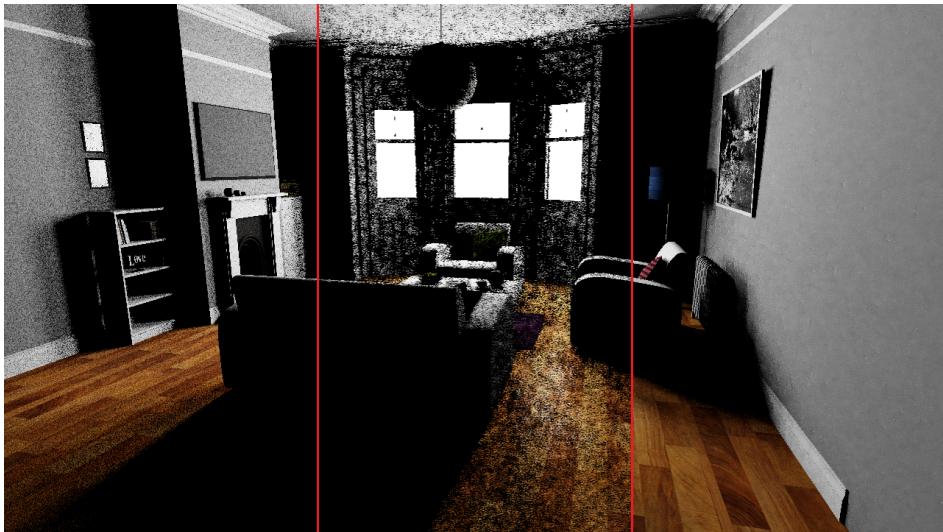
Z výsledků testů (tabulka 4.4 s obrázkem 4.3) lze pozorovat, že zvýšení počtu sousedů vede k redukci šumu, nicméně zlepšení není takové, jako při zvýšení počtu prvních kandidátů. Párové MIS váhy mají menší nárůst času vykreslení oproti generalizované vyrovnané heuristice.

#### 4.3.4 Časové znovupoužití

Tato podkapitola se snaží ukázat, jak se algoritmus ReSTIR chová s různými kombinacemi zapojení průchodů. Nejprve jsou testovány průchody znovupoužití samostatně, poté při vzájemné aktivaci. Výsledky jsou k dispozici v tabulce 4.5 a na obrázku 4.4.

Tabulka 4.5: Výsledky testu časového znovupoužití. Hodnoty uvedené v závorkách platí pro scénu obývacího pokoje, mimo ně platí pro kuchyni.

Typ znovupoužití	Čas (s)	odchylka (RMSE)
Žádné (pouze MIS)	0.924 (0.427)	80.636 (51.92)
Pouze časové	1.471 (0.637)	20.313 (13.23)
Pouze prostorové	1.685 (0.954)	45.259 (26.78)
Prostorové i časové	3.222 (1.415)	7.589 (6.4)



Obrázek 4.4: Ukázka vlivu jednotlivých průchodů znovupoužití (obývací pokoj). Každý obrázek je tvořen jednou iterací MC integrace. Zleva doprava: časové znovupoužití, prostorové znovupoužití, časové + prostorové znovupoužití.

Z testu je ihned patrné, že průchody prostorového a časového znovupoužití samostatně skoro dvakrát snižují odchylku oproti variantě bez jakéhokoli znovupoužití. Časové použití by v případě zachycení animované scény mohlo mít větší odchylku. Pro nejpřesvědčivější výsledky je vhodné za-

pnot oba průchody znovupoužití, kde je odchylka redukována více než  $10 \times$  ve srovnání s variantou bez znovupoužití.

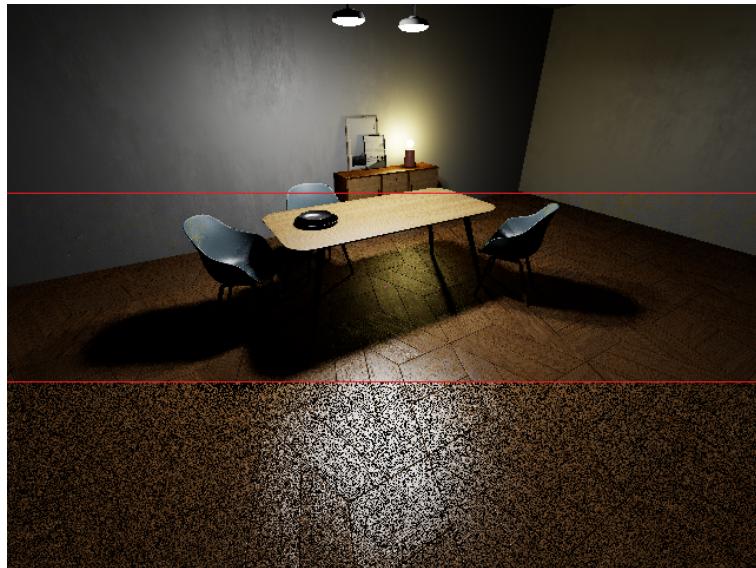
#### 4.3.5 Srovnání s MIS

Nakonec je patřičné algoritmus ReSTIR srovnat se standardním MIS estimátorem. V obou testovacích scénách bylo provedeno srovnání vykreslení jednoho snímku pomocí metod MIS a ReSTIR s referenčním snímkem. Výsledky jsou zachyceny v tabulce 4.6, vizuální podoba testů je na obrázcích 4.5, 4.6.

Tabulka 4.6: Výsledky testu srovnání s MIS. Hodnoty uvedené v závorkách platí pro scénu obývacího pokoje, mimo ně platí pro kuchyni.

Typ snímku	Doba vykreslení (s)	odchylka (RMSE)
Jeden MIS snímek	0.946 (0.395)	80.61 (52.1)
Jeden ReSTIR snímek	2.752 (1.26)	8.1 (6.28)
Zkonvergovaná reference	5027 (3608)	0 (0)

Z výsledků je patrné, že vykreslení jednoho snímku algoritmem ReSTIR sice trvá cca 3-4× déle než vykreslení pomocí MIS, nicméně odchylka oproti referenci výrazně klesá (cca 9-10× nižší než u MIS). Algoritmus ReSTIR tedy jedním snímkem integrace mnohem lépe approximuje integrál v rovnici transportu světla, než samotná technika MIS.



Obrázek 4.5: Ukázka rozdílu mezi vykreslovacími metodami (kuchyně). Zdola nahoru: 1 iterace standardního MIS estimátoru, 1 iterace algoritmu ReSTIR, zkonvergovaná reference.



Obrázek 4.6: Ukázka rozdílu mezi vykreslovacími metodami (obývací pokoj). Zleva doprava: 1 iterace standardního MIS estimátoru, 1 iterace algoritmu ReSTIR, zkonzvergovaná reference.

# Kapitola 5

## Závěr

V této práci byl prozkoumán algoritmus ReSTIR (Reservoir-based spatiotemporal importance resampling) a jeho možnosti pro řešení přímého osvětlení. Nejprve byly stručně uvedeny dvě základní podoby zobrazovací rovnice. Následně byly popsány základní techniky Monte Carlo integrace od naivního estimátoru až po estimátor s převzorkováním, na jehož principech staví algoritmus ReSTIR. Dále byl zaveden samotný algoritmus ReSTIR, byly popsány jeho hlavní myšlenky a průchody. Prozkoumány byly rovněž možnosti zavedení MIS váhových funkcí.

Pro demonstraci schopností algoritmu ReSTIR byla vytvořena ukázková implementace v C++, která obsahuje kromě standardního path traceru také implementaci algoritmu ReSTIR pro přímé osvětlení. V závěru práce je uvedena sekce s porovnáním výkonu algoritmu oproti standardnímu MIS estimátoru.

Při další navazující práci by bylo vhodné přenést aplikaci do kontextu hardwarové akcelerace grafické karty a využít tak masivně paralelní povahy GPU k urychlení kroků algoritmu ReSTIR. Implementaci algoritmu by bylo vhodné rozšířit o výpočet nepřímého osvětlení, čímž by snímky získaly značné vylepšení vizuálního vjemu.

# Literatura

1. WHITTED, Turner. An improved illumination model for shaded display. *Commun. ACM.* 1980-06, roč. 23, č. 6, s. 343–349. ISSN 0001-0782. Dostupné z DOI: 10.1145/358876.358882.
2. KAJIYA, James T. The rendering equation. *SIGGRAPH Comput. Graph.* 1986-08, roč. 20, č. 4, s. 143–150. ISSN 0097-8930. Dostupné z DOI: 10.1145/15886.15902.
3. BITTERLI, Benedikt; WYMAN, Chris; PHARR, Matt; SHIRLEY, Peter; LEFOHN, Aaron; JAROSZ, Wojciech. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*. 2020-07, roč. 39, č. 4. Dostupné z DOI: 10/gg8xc7.
4. WYMAN, Chris; KETTUNEN, Markus; LIN, Daqi; BITTERLI, Benedikt; YUKSEL, Cem; JAROSZ, Wojciech; KOZLOWSKI, Paweł. A Gentle Introduction to ReSTIR Path Reuse in Real-Time. In: *ACM SIGGRAPH 2023 Courses*. Los Angeles, California: Association for Computing Machinery, 2023. SIGGRAPH '23. ISBN 9798400701450. Dostupné z DOI: 10.1145/3587423.3595511.
5. *Portal with RTX Out Now: A Breathtaking Reimagining Of Valve's Classic With Full Ray Tracing & DLSS 3* — nvidia.com [<https://www.nvidia.com/en-us/geforce/news/portal-with-rtx-full-ray-tracing-december-8/>]. [B.r.]. [Accessed 25-05-2025].
6. *Half-Life 2 RTX Playable Demo Out Now, NVIDIA RTX Remix Officially Released With DLSS 4 & RTX Neural Shaders* — nvidia.com [<https://www.nvidia.com/en-us/geforce/news/rtx-remix-half-life-2-rtx-demo-launching-march-18/>]. [B.r.]. [Accessed 25-05-2025].
7. PHARR, M.; JAKOB, W.; HUMPHREYS, G. *Physically Based Rendering, fourth edition: From Theory to Implementation*. MIT Press, 2023. ISBN 9780262048026. Dostupné také z: <https://books.google.cz/books?id=ENSMEAAQBAJ>.
8. VEAH, Eric. *Robust monte carlo methods for light transport simulation*. Stanford, CA, USA: Stanford University, 1998. ISBN 0591907801. Dis. pr. Stanford University. AAI9837162.

9. KONDAPANENI, Ivo; VÉVODA, Petr; GRITTMANN, Pascal; SKŘIVAN, Tomáš; SLUSALLEK, Philipp; KRIVÁNEK, Jaroslav. Optimal Multiple Importance Sampling. *ACM Trans. Graph.* 2019, roč. 38, č. 4. Dostupné z DOI: 10.1145/3306346.3323009.
10. TALBOT, Justin F.; CLINE, David; EGBERT, Parris. Importance resampling for global illumination. In: *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*. Konstanz, Germany: Eurographics Association, 2005, s. 139–146. EGSR '05. ISBN 3905673231.
11. BITTERLI, Benedikt; WYMAN, Chris; PHARR, Matt; SHIRLEY, Peter; LEFOHN, Aaron; JAROSZ, Wojciech. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*. 2020-07, roč. 39, č. 4. Dostupné z DOI: 10/gg8xc7.
12. LIN, Daqi; KETTUNEN, Markus; BITTERLI, Benedikt; PANTALEONI, Jacopo; YUKSEL, Cem; WYMAN, Chris. Generalized resampled importance sampling: foundations of ReSTIR. *ACM Trans. Graph.* 2022-07, roč. 41, č. 4. ISSN 0730-0301. Dostupné z DOI: 10.1145/3528223.3530158.
13. BITTERLI, Benedikt. *Correlations and Reuse for Fast and Accurate Physically Based Light Transport*. Dartmouth College, 2022. Dostupné také z: <https://digitalcommons.dartmouth.edu/dissertations/77>. Dis. pr. Dartmouth College.
14. MICROSOFT CORPORATION. *DirectX 11* [online]. 2024-07-25. [cit. 2025-05-25]. Dostupné z: <https://www.microsoft.com/en-us/download/details.aspx?id=17431>.
15. INTEL CORPORATION. *Intel Embree* [online]. 2025-04-03. [cit. 2025-05-25]. Dostupné z: <https://www.embree.org/>.
16. INTEL CORPORATION. *Intel Open Image Denoise* [online]. 2025-05-01. [cit. 2025-05-25]. Dostupné z: <https://www.openimagedenoise.org/>.
17. KULLING, Kim. *Open Asset Import Library* [online]. 2024-01-01. [cit. 2025-05-25]. Dostupné z: <https://assimp.org/index.html>.
18. DROLON, Hervé. *FreeImage* [online]. 2018-07-31. [cit. 2025-05-25]. Dostupné z: <https://freeimage.sourceforge.io/>.
19. CORNUT, Omar. *Dear ImGui* [online]. 2025-05-20. [cit. 2025-05-25]. Dostupné z: <https://github.com/ocornut/imgui>.
20. RICCIO, Christophe. *GLM* [online]. 2025-02-20. [cit. 2025-05-25]. Dostupné z: <https://github.com/g-truc/glm>.
21. ALLQUAD. *Scandinavian Kitchen scene* [online]. 2024-02-20. [cit. 2025-05-25]. Dostupné z: <https://sketchfab.com/3d-models/free-dining-set-scandinavian-style-61275205fe084208887d748>
22. MCGUIRE, Morgan. *Computer Graphics Archive*. 2017-07. Dostupné také z: <https://casual-effects.com/data>. <https://casual-effects.com/data>.