



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

FOTOREALISTICKÉ ZOBRAZOVÁNÍ 3D SCÉN

PHOTOREALISTIC RENDERING OF 3D SCENES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL VLNAS

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2020

Zadání diplomové práce



Student: **Vlnas Michal, Bc.**
Program: Informační technologie Obor: Počítačová grafika a multimedia
Název: **Fotorealistické zobrazování 3D scén**
Photorealistic Rendering of 3D Scenes
Kategorie: Počítačová grafika

Zadání:

1. Seznamte se s algoritmy realistického zobrazování 3D scén se zaměřením na metodu Path Tracing a/nebo obdobné metody využívající "Monte Carlo" postupy.
2. Vyberte vhodný algoritmus zobrazování využívající "Monte Carlo" postupy pro modelování a simulaci šíření světla ve scéně a diskutujte možnosti jeho implementace.
3. Navrhněte vhodný postup implementace a diskutujte dosažitelné vlastnosti, zejména rychlost výpočtu.
4. Implementujte vybraný algoritmus a demonstруйте jeho funkčnost na vhodném příkladu.
5. Vyhodnořte dosažené výsledky a možnosti dalšího vývoje.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 3. června 2020

Datum schválení: 6. listopadu 2019

Abstrakt

Tato práce navrhuje nový koncept vzorkování, zejména do metody sledování cest, za účelem rychlejší konvergence scény, pomocí lokální aproximace rozložení světla ve scéně, s využitím hemisférických harmonických funkcí, jež umožňuje efektivnější směřování paprsků na daném povrchu. V první části jsou popsány základní principy fotorealistického zobrazování spolu s často používanými algoritmy pro syntézu obrazu. Druhá část popisuje matematický aparát, který je posléze využit v návrhu metody vzorkování. Následně jsou představeny již existující řešení v této oblasti. Další část shrnuje současný stav tomto odvětví. Poslední dvě části se již plně věnují návrhu a implementaci ověření již zmíněného rozšíření s využitím CPU.

Abstract

This thesis proposes a concept of sampling, especially for path tracing like algorithms, for faster convergence of the scene, using a local radiance approximation in the scene with hemispherical harmonics, which allows more effective way of ray casting on the given surface. In the first part, the basics of photorealistic rendering are introduced together with commonly used algorithms for image synthesis. The mathematical apparatus used in this thesis is defined in the second part of the thesis. Subsequently, existing solutions in this area are presented. The following chapter summarizes state-of-the-art methods in this branch. The rest of this thesis is focused on proposal and implementation of already mentioned extension.

Klíčová slova

monte carlo, realistické zobrazování, syntéza obrazu, podstatný výběr, hemisférické harmonické funkce, path tracing

Keywords

monte carlo, realistic rendering, image synthesis, importance sampling, hemispherical harmonics, path tracing

Citace

VLNAS, Michal. *Fotorealistické zobrazování 3D scén*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík

Fotorealistické zobrazování 3D scén

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Michal Vlnas
2. června 2020

Poděkování

Děkuji vedoucímu práce prof. Dr. Ing. Pavlu Zemčíkovi za odborné vedení a užitečné rady, které vedly k úspěšnému dokončení práce.

Obsah

1	Úvod	2
2	Fotorealistické zobrazování	3
2.1	Fenomény realistického zobrazování	3
2.2	Obousměrná distribuční funkce odrazu světla – BRDF	4
2.3	Zobrazovací rovnice	7
2.4	Jednoduché metody pro realistické zobrazování	9
2.5	Pokročilé techniky řešení globálního osvětlení	13
2.6	Struktury pro dělení prostoru	17
3	Matematický aparát v realistickém zobrazování	19
3.1	Generátory pseudo-náhodných čísel	19
3.2	Monte Carlo integrace	21
3.3	Sférické harmonické funkce	21
4	Existující řešení fotorealistického zobrazování	25
4.1	Komerční software	25
4.2	Vědecky zaměřené open-source projekty	27
4.3	Systém pro fotorealistické zobrazování PBRT	27
5	Analýza a zhodnocení současného stavu fotorealistického zobrazování	29
5.1	Hodnocení metod pro realistické zobrazování	29
5.2	Cíl realizační části práce	30
5.3	Technické parametry práce	30
6	Aproximace rozložení světla ve scéně	31
6.1	Projekce do radiálních básových funkcí	31
6.2	Úprava podstatného výběru pro hemisférické harmonické	32
6.3	Interpolace vyzařovací funkce	33
7	Implementace rozšíření do metody sledování cest	34
7.1	Akcelerační datové struktury	34
7.2	Sestavení fotonové a integrální mapy	35
7.3	Aproximace vyzařovací charakteristiky a její interpolace	37
7.4	Vzorkovač vyzařovací funkce	38
7.5	Dosažené výsledky	42
8	Závěr	47
	Literatura	48
A	Obsah přiloženého média	52

Kapitola 1

Úvod

Tato práce popisuje část speciálního odvětví počítačové grafiky, které se zaměřuje na zobrazování scén, obsahujících 3D objekty uložené v počítači, tak, aby byl výsledný obraz co nejvíce podobný fotografii. Současně tato práce ukazuje možnosti rozšíření tohoto odvětví a matematický popis základních metod řešící tuto problematiku. V neposlední řadě popisuje vývoj počítačového programu, který se snaží přinést zrychlení do jedné kategorie metod v tomto odvětví.

Již mnoho let se po celém světě provádí výzkum realistického zobrazování počítačových scén. Za posledních několik desítek let vzniklo mnoho způsobů řešení této problematiky, ačkoliv se stále nedosahuje dostatečných výsledků. Většina technik je stále limitována výkonem procesoru, jelikož jejich komplexita neumožňuje dostatečnou akceleraci na grafické kartě. Ačkoliv zejména v posledních letech se technologie grafických karet rozvinula natolik, že se již objevují způsoby kompletního řešení realistického zobrazování na grafických kartách, ovšem zatím zejména v komerční sféře.

Fotorealistická grafika se v dnešní době používá hlavně ve filmovém průmyslu, kde ji lze využít pro řešení speciálních efektů nebo dnes již dokonce pro vytváření celých filmů. Z toho důvodu je velmi kladen důraz na rychlost koncového programu, který daná filmová studia používají. Proto je výzkum zaměřen především na zlepšení výpočetní náročnosti fotorealistických technik. Kromě filmového průmyslu lze toto odvětví počítačové grafiky nalézt taktéž v architektuře nebo ve strojírenství.

Cílem této práce bylo porozumění aspektům fotorealistického zobrazování a zejména jejich vlivu na rychlost různých metod. Druhým cílem byl návrh a implementace konkrétního řešení, které se snaží tyto metody urychlit.

První část práce, respektive druhá kapitola, popisuje základy fotorealistického zobrazování. Jsou zde popsány fenomény, které lze pozorovat při realistické syntéze obrazu, reprezentace povrchového popisu v počítačové scéně od obecné definice, až ke konkrétním modelům, stěžejní rovnice pro simulaci celého problému a zejména konkrétní metody, které se syntézou realistického obrazu zabývají. Třetí kapitola definuje potřebný matematický aparát, jenž se v této oblasti vyskytuje. Taktéž jsou popsány principy aproximace různých funkcí, vyskytujících se v syntéze obrazu, které jsou posléze využity v návrhu nové techniky. Ve čtvrté kapitole je ukázán již existující software z komerční i nekomerční sféry. Pátá kapitola shrnuje současný stav ve fotorealistickém zobrazování a taktéž detailněji definuje zadání práce. V šesté kapitole je představen návrh aproximace rozložení světla ve scéně, jakožto návrh techniky s cílem zrychlit celkový výpočet. Tento návrh je posléze implementován a popsán v sedmé kapitole. Současně jsou zde ukázány výsledky a vyhodnocení představené techniky.

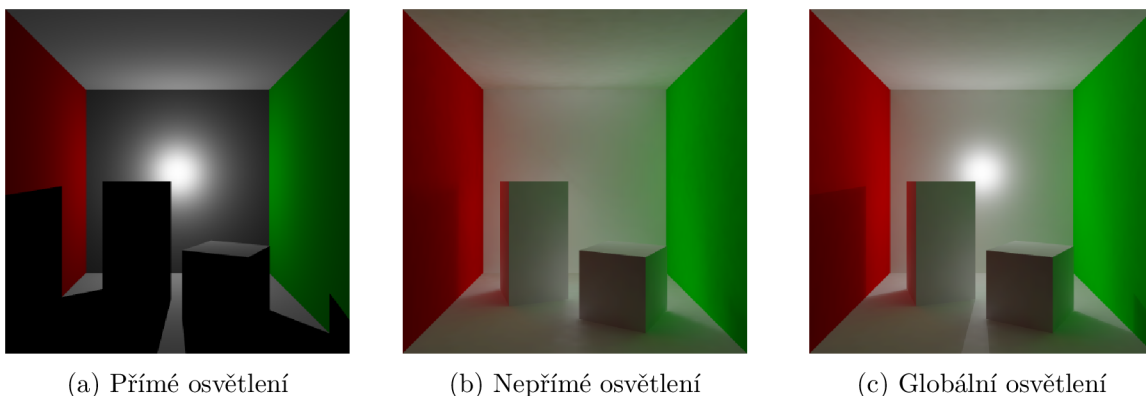
Kapitola 2

Fotorealistické zobrazování

Tato kapitola popisuje principy a metody fotorealistického zobrazování. V první části jsou představeny základní pojmy a fenomény vyskytující se v tomto druhu zobrazování. Posléze jsou definovány vztahy pro řešení této problematiky, včetně možností jejich numerického řešení. Následuje popis konkrétních metod řešících globální osvětlení scény, od základních, až po pokročilé. Tato kapitola si neklade za cíl být encyklopedickým přehledem dané problematiky, ale pouze ukazuje poznatky potřebné k pochopení práce.

2.1 Fenomény realistického zobrazování

Fotorealistické zobrazování, též známé jako realistické zobrazování, je odvětví počítačové grafiky, které se zaměřuje na syntézu obrazu takové kvality, že jej nelze rozpoznat od skutečné fotografie. Toho lze dosáhnout simulací tzv. globálního osvětlení, viz Obrázek 2.1c. To rozděluje osvětlení na dvě části, přímou a nepřímou. Většina dnešních zobrazovacích technik, např. rasterizace trojúhelníků, vykresluje pouze složku přímou, viz Obrázek 2.1a, což má za následek značné omezení vlastností, kterých lze dosáhnout. Simulace nepřímého osvětlení (viz Obrázek 2.1b) je tedy hlavním zdrojem fotorealistických efektů. Stručný výčet různých fenoménů se kterými se lze v této problematice setkat je popsán níže. V praxi se samozřejmě používá mnoho dalších technik, které se snaží nepřímou složku světla aproximovat nebo napodobit.



Obrázek 2.1: Osvětlení scény

Měkké stíny Ačkoliv konvenční zobrazovací techniky, které se hojně využívají např. v počítačových hrách, s přehledem zvládají vykreslovat kvalitní stíny, tak většinou se jedná o tzv. tvrdé (angl. hard) stíny, které se vyznačují ostrými hranami. Opačným případem jsou tzv. měkké (angl. soft) stíny. Ty lze identifikovat tak, že oproti tvrdým stínům mají hrany viditelně neostré. Tento jev je způsobem zejména nepřímou složkou světla.

Color bleeding V počítačové grafice je tímto názvem popsán fenomén, ve kterém jsou objekty nebo povrchy obarveny barvou odraženého světla z okolí. Příkladem může být např. levé, resp. pravé, těleso na Obrázku 2.1c. Světlá stěna kvádrů je zde obarvena na základě barevného odrazu od sousední stěny.

Kaustiky Kaustika je běžný optický jev. Jedná se o obálku světelných paprsků odražených nebo zalomených nějakou zakřivenou plochou. Běžným případem je sklenice vody, na kterou svítí světelný zdroj. Sklenice vrhá kromě stínu rovněž zakřivenou oblast velice jasného světla, tedy kaustiku.

SDS cesty *Specular-Diffuse-Specular path*, je situace, kdy cesta paprsku prochází skrze spekulární prostředí, následně difúzní a posléze opět spekulární. Taková situace může nastat např. když prochází paprsek skleněným prostředím, které vyzařuje kaustiku na difúzní materiál, který je posléze vidět v nějakém zrcadle. Další typický příklad je např. dno bazénu, kde rozhraní voda-dno-voda je SDS cesta. Tento fenomén je nejsložitější na simulaci, protože běžné algoritmy v této situaci selhávají.

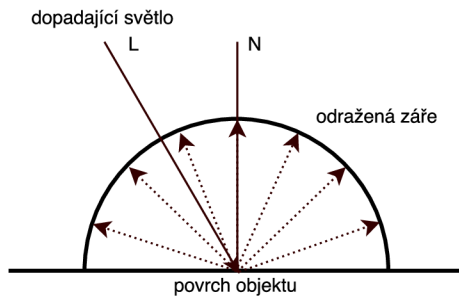
2.2 Obousměrná distribuční funkce odrazu světla – BRDF

Z anglického spojení *bidirectional reflectance distribution function*, dále jen BRDF, je funkce, která v počítačové grafice vyjadřuje vlastnosti daného povrchu ve scéně. Udává hustotu pravděpodobnosti, že světlo dopadající ze směru ω_i se odrazí ve směru ω_o (oba prostorové úhly jsou definované vůči normále). Původní definice byla zavedena F. Nicodemusem [19] roku 1965. Je vyjádřena vztahem:

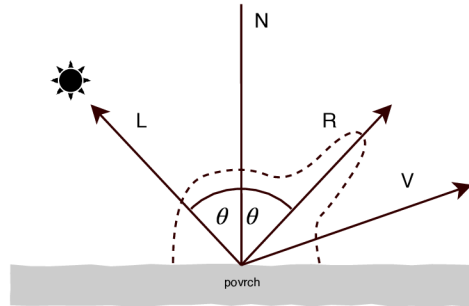
$$f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) = \frac{dL_r(\omega_o)}{dE(\omega_i)} = \frac{dL_r(\omega_o)}{L_i(\omega_i) \cdot \cos(\theta_i) d\omega_i} \quad [\text{sr}^{-1}] \quad (2.1)$$

kde \mathbf{x} označuje bod na povrchu, ω_i, ω_o příchozí, resp. odchozí, směr paprsku, L_r je diferenciální odražená záře ve směru ω_o , E je diferenciální ozáření povrchu ze směru ω_i a θ_i je úhel mezi směrem dopadající záře L_i a normálou. Současně BRDF splňuje následující vlastnosti:

- **Pozitivita** – funkce je vždy nezáporná, $f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \geq 0$
- **Linearita** – BRDF je lineární vzhledem k záři, prakticky to znamená, že příspěvky jednotlivých světelných zdrojů lze sčítat, což přímo vyplývá ze vzorce (2.1)
- **Zákon zachování energie** – poměr odraženého zářivého toku k příchozímu musí být menší nebo rovno 1, tedy $\frac{d\Phi_r}{d\Phi_i} \leq 1$, tzn. nelze odrazit více světla než přichází
- **Helmholzova reciprocita** – toto je vlastnost každé fyzikálně korektní BRDF a vyplývá přímo ze zákona odrazu, tedy platí: $f_r(\mathbf{x}, \omega_o \rightarrow \omega_i) = f_r(\mathbf{x}, \omega_i \rightarrow \omega_o)$



Obrázek 2.2: Lambertův osvětlovací model



Obrázek 2.3: Phongův osvětlovací model

Izotropie vs anizotropie

Mnoho jednoduchých BRDF je v praxi invariantní vůči otočení k normále, neboli jsou tzv. **izotropní**. Jedná se zejména o hladké materiály, které jsou ve všech směrech modelovány stejně. Opačným případem mohou být **anizotropní** materiály, které obsahují mikrostruktury, kde v každém směru od normály je struktura rozdílná. U takovýchto materiálů už nelze vystačit pouze s úhly definovanými výše, ale je nutno zavést i orientaci vzhledem k normále.

Modely

BRDF lze modelovat dvěma způsoby. Buď lze přímo měřit vlastnosti povrchu speciálními přístroji, nebo lze odvodit analytický model. Tyto modely lze posléze rozdělit na empirické a fyzikálně konzistentní. Následuje stručný popis nejzajímavějších vybraných modelů BRDF.

Lambertův model

Nezákladnější BRDF je Lambertův [25] model. Tento model je ideální reprezentací dokonale matného povrchu, tzn. řeší pouze difúzně odrazivý povrch. Z důvodu termodynamických vlastností se ovšem tento model v přírodě nevyskytuje. Jeho specifičnost je v tom, že se jedná o izotropní model, neboli jeho jasové vyzařování je stejné ve všech směrech, nezávisle na pozorovateli (viz Obrázek 2.2). Intenzita odraženého světla odpovídá Lambertovu zákonu [25].

Phongův model

Druhým empirickým modelem je Phongův [24]. Jeho hlavní předností je, že k difúzní složce světla přidává složku spekulární a ambientní (okolní). Ukázka odrazivosti je ilustrována na Obrázku 2.3. Difúzní část světla odpovídá Lambertovskému modelu. Lesklá, neboli spekulární, část je závislá na koeficientu lesklosti a na intenzitě výskytu odlesků. Odražený paprsek je posléze vypočítán podle zákona odrazu. Okolní, resp. ambientní (z anglického ambient), složka simuluje nepřímé osvětlení, ačkoliv se jedná pouze o konstantu, která je úměrná koeficientu odrazu a intenzitě okolního světla.

V praxi (nikoliv ovšem v realistickém zobrazování) se spíše využívá optimalizovaná verze Phongova modelu publikovaná J. Blinnem [2] (Blinn-Phong model).



Obrázek 2.4: Oren-Nayar BRDF



Obrázek 2.5: Torrance sparrow BRDF

Torrance-Sparrow

Tento model, publikován K. Torrance a E. Sparrowem [29], je jeden z nejvíce fyzikálně korektních BRDF pro izotropní materiály, viz Obrázek 2.5. Modeluje povrch pomocí distribuce perfektně spekulárních mikroskopických kavit ve V-formě (každá kavita je tedy tvořena dvěma ploškami) stejné velikosti, nazvaných mikroplošky (angl. microfacets). Orientace plošek a jejich distribuce je parametrizovaná, což umožňuje modelovat různé úrovně hrubosti. Nejčastěji se pro distribuci normál využívá Gaussovo rozložení, ačkoliv velmi užitečná je rovněž Beckmannova [28] distribuce.

Rozšířením tohoto modelu vznikl ještě tzv. **Cook-Torrance** model [3], který funguje principiálně stejně, ačkoliv uvažuje ještě vlnovou délku, jakožto vlastnost pro určení odrazu světla, a tím modeluje fyzikálně přesnější světelné odrazy.

Wardův model

Tento model, který publikoval v roce 1992 G. Ward [36], slouží k reprezentaci anizotropní BRDF. Ačkoliv se jedná o empirický model, tak bývá velmi často využíván pro své specifické možnosti. Difúzní složka je řešena typicky pomocí Lambertova modelu. Naopak druhá komponenta slouží k modelování anizotropie pomocí parametrického „laloku“, kde distribuce vzorků je řízena Gaussovým rozložením. V kombinaci s mikroploškami, každá mikroploška řeší daný „lalok“ dle orientace tangenty povrchu, ve výsledku vzniká anizotropní lesklý efekt.

Oren-Nayar

Oren-Nayarův model [22] je BRDF pro difúzní odraz z hrubých povrchů. Použitý model hrubosti povrchu je odvozenina z mikro-ploškového modelu od Torrance-Sparrow [29]. Hrubost povrchu je specifikována pomocí pravděpodobnostní funkce pro distribuci sklonů plošek. Zejména bývá využito Gaussovo rozložení, kde tedy rozptyl σ^2 je míra hrubosti povrchu.

Posléze je každá ploška uvažována jako Lambertovská, což ve výsledku umožňuje difúzní odraz, viz Obrázek 2.4.

Ashikhmin-Shirley

Tento model publikoval Ashikhmin a Shirley [1] v roce 2000. Řeší difúzní podkladový povrch s lesklou vrstvou nad ní. Efekt difúzního odrazu je modulovaný množstvím energie, která zůstane po započítání Fresnelových efektů na lesklé vrstvě. Pokud je dopadající paprsek blízko normály, tak většina energie prorazí na difúzní vrstvu a tento efekt tedy dominuje, v opačném případě převládá lesklá složka. Hlavní ideou modelu je odvození takového difúzního modelu, který stále zachovává reciprocitu a zákon zachování energie, jelikož celkově je model uvažován jako vážená suma lesklé a difúzní složky, tedy Lambertův model by zde nebyl fyzikálně korektní. Samozřejmě i tato BRDF uvažuje a využívá teorii mikroplošek.

2.3 Zobrazovací rovnice

Zobrazovací rovnice, angl. rendering equation, je integrální rovnice popisující přenos světla ve scéně, jejímž řešením je rovnovážný stav rozložení světla. Poprvé byla publikována Jamesem T. Kajiyou [12] v roce 1986. Úhlová forma rovnice je dle původního znění:

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} L(\mathbf{x}', -\omega_i) f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cdot \cos(\theta_i) d\omega_i \quad (2.2)$$

kde $L(\mathbf{x}, \omega_o)$ je celková záře dopadající na bod \mathbf{x} ve směru ω_o , $L_e(\mathbf{x}, \omega_o)$ je emitovaná zář bodem \mathbf{x} v tentýž směr (nastane např. pokud je daný povrch světelný zdroj či jiný emisní povrch), $f_r(\mathbf{x}, \omega_i \rightarrow \omega_o)$ značí koeficient BRDF a θ_i je úhel mezi normálou na povrchu v bodě \mathbf{x} a směrem ω_i . Jak již z názvu formy napovídá, integrál je veden přes všechny směry Ω na hemisféře nad bodem \mathbf{x} .

Z rovnice (2.2) je patrné, že je obecně těžce řešitelná. K jejímu řešení se nejčastěji využívají metody založené na numerickém integrování pomocí např. Monte Carlo nebo Metropolis algoritmu. Taktéž úhlová forma rovnice není vhodná pro složité algoritmy, proto se v praxi zavádí různé modifikace a jiné formy, které lze využít pro komplexnější metody.

Nejčastěji využívanou modifikací je **tří bodová plošná forma** [30] zobrazovací rovnice. Pro odvození této rovnice je nutno se zbavit závislosti na směrech ω_o a ω_i . Z toho důvodu lze zavést následující konvenci (2.3):

$$L(\mathbf{x} \rightarrow \mathbf{x}') = L(\mathbf{x}, \omega_o), \quad (2.3)$$

kde $\omega_o = \mathbf{x}' - \mathbf{x}$, jenž označuje směr z bodu \mathbf{x} do \mathbf{x}' . Analogicky lze převést i BRDF koeficient, viz vztah níže (2.4):

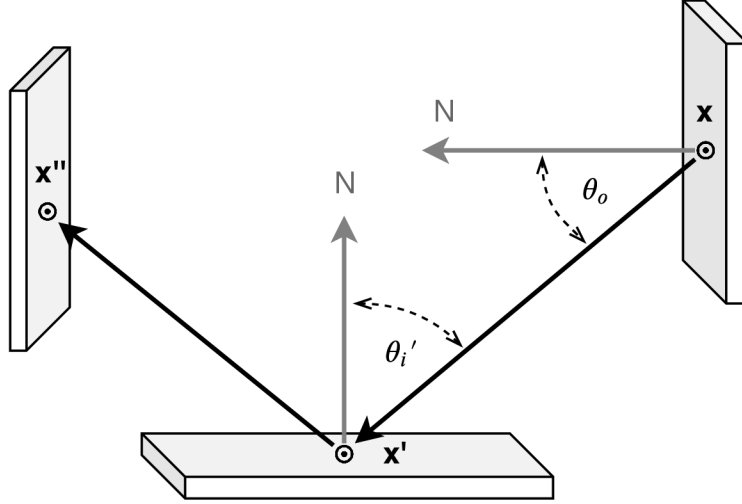
$$f_r(\mathbf{x} \rightarrow \mathbf{x}' \rightarrow \mathbf{x}'') = f_r(\mathbf{x}', \omega_i \rightarrow \omega_o), \quad (2.4)$$

kde $\omega_o = \mathbf{x} - \mathbf{x}'$ a $\omega_i = \mathbf{x}'' - \mathbf{x}'$. Notace $\mathbf{x} \rightarrow \mathbf{x}'$ symbolizuje směr proudění světla. Vztah (2.5) mezi prostorovým úhlem a jednotkovou ploškou je definován jako:

$$d\omega = G(\mathbf{x} \leftrightarrow \mathbf{x}') dA(\mathbf{x}) \quad (2.5)$$

Po zavedení předchozích konvencí může být třibodová plošná forma zobrazovací rovnice (integrována tedy již přes plošky) [30] zapsána následovně:

$$L(\mathbf{x}' \rightarrow \mathbf{x}'') = L_e(\mathbf{x}' \rightarrow \mathbf{x}'') + \int_M L(\mathbf{x} \rightarrow \mathbf{x}') \cdot f_r(\mathbf{x} \rightarrow \mathbf{x}' \rightarrow \mathbf{x}'') \cdot G(\mathbf{x} \leftrightarrow \mathbf{x}') dA(\mathbf{x}), \quad (2.6)$$



Obrázek 2.6: Geometrie zobrazovací rovnice v tří bodové formě

kde M je sjednocení všech povrchů ve scéně, A je tedy jednotková oblast v M , G je tzv. geometrický člen (viz vztah 2.7), který zohledňuje orientaci plošek v prostoru. Definován je následovně:

$$G(\mathbf{x} \leftrightarrow \mathbf{x}') = V(\mathbf{x} \leftrightarrow \mathbf{x}') \cdot \frac{|\cos(\theta_o) \cdot \cos(\theta'_i)|}{\|\mathbf{x} - \mathbf{x}'\|^2}, \quad (2.7)$$

kde θ_o a θ'_i jsou úhly mezi segmentem $\mathbf{x} \leftrightarrow \mathbf{x}'$ a normálou v bodě \mathbf{x} , respektive \mathbf{x}' , viz Obrázek 2.6. Funkce V určuje viditelnost mezi danými body, nabývá hodnoty $V(\mathbf{x} \leftrightarrow \mathbf{x}') = 1$ pokud jsou body navzájem viditelné, jinak nula.

Rovnice měření intenzity a přenosová rovnice

Problém výše definované rovnice (2.6) je ale stále její nekonečně rekurzivní integrál, jenž nelze řešit jinak, než nahrazováním proměnné na pravé straně přímo pravou stranou, které ale vyústí právě v již zmíněné nekonečně složité řešení, což není výpočetně zrovna ideální, viz následující zjednodušený rozvoj (2.8):

$$\begin{aligned} L &= L_e(x) + \int_M L \, dA = \\ &= L_e(x) + \int_M L_e(x) \, dA + \int_M \int_M L \, dA \, dA = \\ &= L_e(x) + \int_M L_e(x) \, dA + \int_M \int_M L_e(x) \, dA \, dA + \int_M \int_M \int_M L \, dA \, dA \, dA = \dots \end{aligned} \quad (2.8)$$

Pro řešení této situace je nutno nejdříve definovat zbývající závislosti, předně rovnici pro měření intenzity [30] pixelu v obraze, viz rovnici (2.9) níže.

$$I_j = \int_{M \times M} W_e^j(\mathbf{x} \rightarrow \mathbf{x}') L(\mathbf{x} \rightarrow \mathbf{x}') G(\mathbf{x} \rightarrow \mathbf{x}') \, dA(\mathbf{x}) \, dA(\mathbf{x}') \quad (2.9)$$

kde W_e^j je potenciál emitovaný z bodu \mathbf{x}' k \mathbf{x} (opačně než u šipkové notace).

Výše definovaná rovnice je již přímo řešením konkrétního pixelu v obrazové rovině. Dle E. Veache [30] lze rovnici pro měření intenzity formulovat jako sumu přes sadu cest, kde

cesta je definována jako $\bar{x} = \mathbf{x}_0 \mathbf{x}_1 \dots \mathbf{x}_k$ pro $1 \leq k < \infty \wedge \forall x_i \in M$, rekurzivní expanzí rovnice (2.6) do formy:

$$I_j = \sum_{k=0}^{\infty} \int_{M^{k+1}} L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1) \cdot G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1) \cdot W_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k) \quad (2.10)$$

$$\left(\prod_{i=1}^{k-1} f_r(\mathbf{x}_{i-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{x}_{i+1}) \cdot G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) \right) dA(\mathbf{x}_0) \dots dA(\mathbf{x}_k)$$

Integrand je definovaný pro každou cestu délky k zvlášť, lze ho získat extrakcí z expanze v rovnici (2.10), např. pro cestu $\bar{x} = \mathbf{x}_0 \mathbf{x}_1 \mathbf{x}_2 \mathbf{x}_3$:

$$f_j(\bar{x}) = L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1) \cdot G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1) \cdot f_r(\mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \mathbf{x}_2) \quad (2.11)$$

$$\cdot G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2) \cdot f_r(\mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \mathbf{x}_3) \cdot G(\mathbf{x}_2 \leftrightarrow \mathbf{x}_3) \cdot W_e^j(\mathbf{x}_2 \rightarrow \mathbf{x}_3)$$

Takto definovaný integrand (2.11) je v literatuře [30] nazván jako funkce přínosu měření intenzity pixelu. Hlavní výhodou výše ukázané rovnice pro měření intenzity a zejména plošné formy zobrazovací rovnice, je její nezávislost na měřené veličině. Ačkoliv je využita pro přenos záře ve scéně, lze ji využít i pro jiné přenosy, zejména pro přenos potenciálu a řešení tzv. potenciálové rovnice. Toto lze posléze využít v určitých algoritmech pro řešení globálního osvětlení, viz následující sekce.

2.4 Jednoduché metody pro realistické zobrazování

V dnešní době existuje řada metod a algoritmů pro zobrazování, některé lze nazvat přímo jako metody řešící globální osvětlení, jiné se pouze snaží dosáhnout co nejlepšího výsledku v rozumném čase. Většina užívaných metod využívá principu *vrhání paprsku* do scény, z toho důvodu je nutno definovat základní pojmy v této problematice.

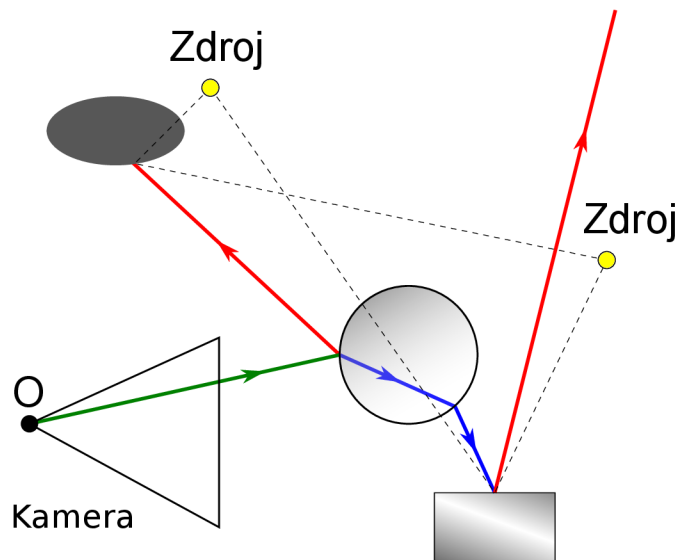
Paprsek V následujících kapitolách je pojmem paprsek myšlena dvojice $r = (\mathbf{x}, \vec{\omega})$, kde \mathbf{x} je počáteční bod a $\vec{\omega}$ směr paprsku.

Ray casting (vrhání paprsku) Tento termín označuje vystřelení paprsku do scény buď z kamery nebo ze světelného zdroje. Vržené paprsky lze označit za **primární**. Jejich následovníky, tedy odražené, rozptýlené či lomené lze nazvat jako **sekundární**.

Biased vs. unbiased V počítačové grafice unbiased vykreslování znamená techniku, která do aproximace radiance nezavádí systematickou chybu, resp. statistický bias. Z toho důvodu unbiased algoritmy mohou konvergovat až k referenčnímu výsledku, kdežto biased techniky nikdy nekonvergují.

Ray tracing

Nejznámější metodou, která se v této oblasti velmi prosazuje, je sledování paprsku, neboli ray tracing. Tento algoritmus publikoval T. Whitted [37] roku 1979. Nutno zmínit, že se nejedná o metodu řešení globálního osvětlení, jelikož neřeší zobrazovací rovnici 2.2, ale jedná se o deterministický algoritmus, který počítá výsledný obraz pouze na základě ohodnocení cest. Důležité je, že využitím základního principu této metody lze vytvořit algoritmy, které již matematicky splňují zobrazovací rovnici.



Obrázek 2.7: Ray tracing; primární paprsky – zelené, odražené – červené, lomené – modré, stínové – čárkované

Algoritmus jako takový je v podstatě deterministické rozšíření principu vrhání paprsku, kde jsou již přítomny i sekundární paprsky. Principiálně tedy jsou vrhány z kamery primární paprsky do scény, u kterých se řeší průsečík s objekty ve scéně. Následně jsou definovány tři druhy paprsků, kterými metoda může pokračovat, viz Obrázek 2.7.

- **Stínové** – neboli shadow rays, jsou vyslány od průsečíku ke všem světelným zdrojům ve scéně. Na základě toho, zda jsou viditelné, či zakryté, se určí možný přínos světla z vyzařovací charakteristiky zdroje na daný průsečík ve scéně.
- **Odražené** – pokud se jedná o odrazivý povrch, tak je vygenerován nový paprsek dle zákona odrazu světla.
- **Lomené** – je-li povrch v bodě průsečíku paprsku s objektem rozhraní umožňující světelný lom, tak lze vygenerovat lomený paprsek dle Snellova zákona. Typickým příkladem je např. skleněný povrch či hladina vody a jiné.

Jelikož v základní variantě ray tracing (RT) neřeší typicky nepřímé osvětlení (protože při zásahu objektu s vlastnostmi pouze pro difúzní část světla algoritmus nepokračuje), tak neumí řešit měkké stíny, ani SDS cesty či kaustiky. Ovšem jak již bylo řečeno, lze tuto metodu rozšířit o různá vylepšení, které zlepší celkové chování algoritmu. Namísto pevně definovaného chování na povrchu (které je typicky řešeno pomocí rekurzivního Phongova, resp. Blinn-Phongova modelu), lze zavést simulace BRDF. Prakticky se jedná o mnohonásobné rekurzivní zanoření, kde z každého průsečíku lze vyslat N paprsků na základě vlastností dané BRDF. Takovému principu se posléze říká **distribuovaný ray tracing**. Tímto způsobem lze již simulovat fyzikálně korektní BRDF, nebo chování kamery (např. motion blur). Ovšem nutno podotknout, že se posléze jedná o velmi výpočetně složitý algoritmus, který sice umožňuje již více možností (přidáním stochastičnosti lze již např. simulovat kompletní nepřímé osvětlení), ovšem prakticky je značně nepoužitelný. Proto je nutno představit komplexnější algoritmy, které danou problematiku řeší efektivnější cestou.

Path tracing

Jednou z nejvyužívanějších základních metod pro řešení globálního osvětlení je metoda sledování cesty, angl. path tracing, která byla poprvé publikovaná spolu se zobrazovací rovnicí od J. T. Kajiya [12] roku 1986. Tato metoda již plně řeší globální osvětlení scény pomocí aproximace zobrazovací rovnice metodou Monte Carlo. Hodnoty integrandu jsou voleny formou náhodné procházky scénou. Zejména z toho důvodu je algoritmus označován jako tzv. *unbiased* technika. Základní princip je popsán v Algoritmu 1 a rovněž odpovídá rovnici:

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \frac{L(\mathbf{x}', -\omega_i) f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) |N_{\mathbf{x}} \cdot \omega_i|}{p(\omega_i)} \quad (2.12)$$

kde ω_i je vygenerovaný paprsek podle hustoty pravděpodobnosti p a $|N_{\mathbf{x}} \cdot \omega_i|$ je pouze jiné vyjádření pro $\cos(\omega_i)$.

Celý algoritmus posléze funguje tak, že jsou vysílány paprsky do obrazové roviny (pro každý pixel), pro něž je následně určena hodnota přenesené záře pomocí rovnice (2.12), resp. Algoritmu 1. Prakticky je jasné, že jeden vzorek na pixel nebude dostačující, vzhledem k povaze integrování pomocí MC. Proto je nutné celý proces opakovat dokud nebude chyba v obraze dostatečně malá (řádově se jedná typicky o tisíce vzorků na pixel). Dalším problémem je, že algoritmus selže, budou-li ve scéně pouze bodová světla jakožto jediné emitory záře. Nejen kvůli již zmíněným problémům, ale taktéž kvůli implementační neefektivitě, byla zavedena různá rozšíření:

- **Vzorkování podle BRDF** – jak je již v Algoritmu 1 naznačeno, je vhodnější nové směry generovat spíše s hustotou pravděpodobnosti, která odpovídá dané BRDF, než uniformně vzorkovat polokouli, ačkoliv ne všechny BRDF mají non-uniformní rozložení.
- **Přímé osvětlení** – významného urychlení konvergence scény a rovněž redukce paprsků s nulovým přínosem, lze dosáhnout pomocí přímého osvětlení [26] povrchu pomocí světelných zdrojů. Prakticky se jedná o vyslání stínového paprsku od náhodného světelného zdroje k danému bodu povrchu. Přínos tohoto paprsku s hlavními nosnými paprsky se posléze sečte. Tuto techniku lze uplatnit pouze pro nelesklé povrchy, v opačném případě by došlo ke zvýšení šumu v obraze.
- **Mapování prostředí** – neboli tzv. environment mapy lze využít k velkému zvýšení realističnosti scény, protože je lze uvažovat rovněž jako světelný zdroj. Prakticky se využívají zejména sférické mapy aplikované na kouli kolem scény, jelikož je zde snazší směrové vzorkování, než u textur namapovaných na krychli. V kombinaci s technikou podstatného výběru lze dosáhnout velmi přesvědčivých výsledků.
- **Obousměrné sledování cest** – jelikož PT jako takový má značné problémy s vykreslením kaustik, jelikož paprsky jsou sledovány od pozorovatele a nikoliv od světelného zdroje, lze zavést techniku obousměrného sledování. V praxi to znamená, že paralelně sledujeme jak kamerovou cestu, tak i světelnou z náhodného zdroje ve scéně. Tyto cesty jsou navzájem posléze propojeny a dle různých heuristik [30, 31] ohodnoceny. Tato technika je podrobněji popsána níže.

Ačkoliv výše zmíněné techniky značně zvyšují výpočetní náročnost algoritmu, tak současně významně redukují náhodný šum v obraze.

Algoritmus 1: PATHTRACING

Data: r je paprsek

Výstup: zář v daném směru r

1. $H \leftarrow$ zjisti průsečík paprsku se scénou
 2. **if** H neobsahuje průsečík nebo dosažena max. hloubka rekurze **then**
 3. | **return** 0 nebo barva pozadí
 4. **else**
 5. | // rekurzivní výpočet
 6. | $N \leftarrow$ urči normálu na povrchu
 7. | $E \leftarrow$ zjisti množství emitované záře daným povrchem
 8. | $r' \leftarrow$ vygeneruj nový paprsek pomocí BRDF s pravděpodobností p
 9. | **return** $E + \text{PATHTRACING}(r') \cdot f_r(H.\text{pos}, r \rightarrow r') \cdot \text{DOT}(r', N) / p$
 10. **end**
-

Photon mapping

Poslední zmíněnou technikou v této kapitole je mapování fotonu. Ačkoliv se jedná již o dvou-průchodovou metodu, tak ji lze řadit stále mezi ty základní, protože již existují mnohem pokročilejší varianty tohoto algoritmu. Původní techniku publikoval H. W. Jensen [11]. Na rozdíl od PT se jedná o tzv. *biased* techniku, ačkoliv se stále jedná o konzistentní techniku, kde kvalita obrazu může být zvýšena užitím většího počtu fotonů.

1) Tvorba fotonové mapy

V prvním průchodu scénou je tvořena tzv. fotonová mapa. Ze všech světelných zdrojů jsou vysílány fotony do scény, když foton zasáhne objekt, tak je určen úbytek zářivého toku Φ_e . Údaje o průsečíku a směru jsou posléze uloženy do mapy. Na základě úbytku Φ_e je paprsek buď pohlcen a tedy nepokračuje, nebo je odražen, resp. zlomen, podle BRDF/BTDF. Jelikož se jedná o prostorová data, ve kterých je nutno mít definovanou operaci vyhledání k-nejbližších sousedů, tak je doporučeno využít např. kD-stromy pro reprezentaci výsledné struktury. Z praktického hlediska se typicky tvoří dvě fotonové mapy, jedna pro nepřímé osvětlení, druhá pro kaustiky. Je to zejména z důvodu, že přesná simulace kaustik vyžaduje mnohonásobně více fotonů, než nepřímé osvětlení.

2) Vykreslování

Druhý průchod funguje podobně jako ray tracing, paprsky jsou vrhány z kamery do scény. Při průsečíku paprsku s povrchem je nutno řešit 4 oddělené situace, které v photon mappingu nastávají – přímé osvětlení, lesklé odrazy, kaustiky a nepřímé osvětlení. Pro výpočet přímého osvětlení může být použit stejný princip, jaký již byl představen u metody path tracing. Podobně lze řešit i lesklé odrazy, jelikož tuto situaci nativně zvládá jak RT, tak i PT.

Naopak pro nepřímé osvětlení je již nutno využít připravenou fotonovou mapu. Nejprve je nutno vyhledat N nejbližších sousedících fotonů v mapě. Uložený zářivý tok fotonů lze převést na zář pomocí koule S , v jejímž poloměru jsou hledány fotony, a směru, ve kterém vyšetřujeme hodnotu záře. Tyto nové hodnoty je nutno ještě vynásobit koeficientem BRDF a posléze sečíst dohromady. Výsledná hodnota je aproximací nepřímého osvětlení.

Pro řešení kaustik je využit identický přístup, ovšem fotony jsou hledány v mapě, která je určena pro simulaci kaustik. Jelikož typicky tato mapa obsahuje řádově více fotonů, tak je tato část výpočtu nejnáročnější.

2.5 Pokročilé techniky řešení globálního osvětlení

Tato kapitola se zaměřuje již na složitější techniky pro realistické vykreslování, jelikož výše zmíněné algoritmy nedosahují potřebných kvalit a je tedy nutno zvolit komplexnější přístup.

Stochastický progresivní photon mapping

První z pokročilých technik je vylepšený algoritmus mapování fotonu. Ze základního popisu tohoto algoritmu v předchozí sekci je jasné, že hlavní nevýhodou je jeho obrovská paměťová náročnost. Ačkoliv tento problém lze řešit menším rozlišením fotonové mapy, tak tím ale utrpí výsledný syntetizovaný obraz na kvalitě, tedy není to vhodné řešení. Tento problém se snaží řešit **progresivní mapování fotonu** [9] restrukturalizací celého algoritmu. Opět je výpočet rozdělen na dvě fáze. V prvním průchodu se sleduje cesta z kamery, kde každý pixel uloží geometrickou a BRDF reprezentaci „nelesklých“ bodů, ve kterých paprsek interaguje se scénou (v případě zásahu perfektně spekulárního povrchu a posléze difúzního, se uloží difúzní vlastnosti škálované spekulární BRDF z předchozího zásahu atd.). Tyto uložené informace se nazývají *viditelné body*. Druhý průchod sleduje fotony ze světelných zdrojů. Při každé interakci fotonu se scénou se určuje přínosnost fotonu k nepřímému osvětlení blízkých viditelných bodů.

Ačkoliv se zdá, že toto řešení je již paměťově nenáročné, tak pro situace, kdy je potřeba řešit mnoho vzorků na pixel, může opět dojít k velké zátěži na paměť, jelikož je nutno uložit velké množství viditelných bodů.

Dalším rozšířením je **stochastický progresivní photon mapping** (SPPM) [8], který již netrpí žádným výše zmíněným problémem. Modifikuje progresivní mapování fotonu tím způsobem, že snižuje vzorkovací frekvenci pixelu, typicky na jeden vzorek na pixel. Algoritmus pak vypadá velice podobně. Je vytvořena množina viditelných bodů pro každý pixel, následně jsou vystřeleny fotony ze světelných zdrojů, které akumulují přínosy na blízké viditelné body. Následně se proces opakuje, původní množina viditelných bodů je zahazena a následně vytvořena znovu na jiných pozicích (diferenciál pixelu), a opět se opakuje vystřelování fotonu a tak dále.

Bidirectional path tracing

Velmi zajímavou technikou, která již byla zmíněna u algoritmu PT, je obousměrné sledování cesty, angl. bidirectional path tracing, jenž původně publikoval E. Lafortune [15] paralelně s E. Veach [30]. Hlavním rozdílem oproti klasickému algoritmu sledování cest, je jeho rozšíření o cesty sledované ze světelných zdrojů, jelikož to je důvod proč PT konverguje velmi pomalu a není schopen simulovat všechny efekty, např. kaustiky.

Algoritmus funguje v znatelně komplexnější formě. Současně se tedy sledují dvě nezávislé cesty, kamerová a světelná. Sledování paprsků z kamery funguje stejně jako u klasického PT, při sledování jednotlivých vzorků lze akumulovat přímé osvětlení, případně zavést terminační kritérium typu Ruská ruleta [30], apod. Druhá část je sledování světelné cesty. Principiálně se jedná o podobný proces, ovšem paprsky jsou vystřelovány ze světelných zdrojů a posléze sledovány. Tyto dvě nezávislé cesty jsou následně spojeny, čímž dojde

k velké eliminaci cest s nulovým přínosem, viz PT. Samozřejmě se jedná o ideální případ, protože dost často se může stát, že cesty mezi sebou nejsou viditelné, např. kvůli nějaké překážce ve scéně.

Zavedením sledování světla se definice cesty mění formálně z \bar{x} na $\bar{x}_{s,t}$, kde s je délka světelné cesty a t délka kamerové cesty. Platí tedy, že každá generovaná cesta, dle hustoty pravděpodobnosti $p_{s,t}$, je ve formátu $\bar{x}_{s,t} = \mathbf{y}_0 \dots \mathbf{y}_{s-1} \mathbf{z}_{t-1} \dots \mathbf{z}_0$, kde celková délka je tedy $s + t$ bodů a $k = s + t - 1$ hran. Body s indexy \mathbf{y}_{s-1} a \mathbf{z}_{t-1} jsou nazývány spojovací a hrana mezi nimi jako spojovací hrana. Estimátor vycházející z rovnice (2.10) [30] definující obousměrné sledování cest je definován rovnicí (2.13):

$$F = \sum_{s \geq 0} \sum_{t \geq 0} C_{s,t} = \sum_{s \geq 0} \sum_{t \geq 0} w_{s,t} \cdot \frac{f_j(\bar{x}_{s,t})}{p_{s,t}(\bar{x}_{s,t})}, \quad (2.13)$$

kde $w_{s,t}$ je váhová funkce, viz dále, f_j je přírůstek intenzity pixelu danou cestou a $p_{s,t}$ je hustota pravděpodobnosti vzorkování dané cesty. Pro zjednodušení výpočtu lze rovnici dále rozepsat do tvaru v rovnici (2.14), kde je nezávisle řešena světelná cesta, kamerová cesta i spojovací hrana $c_{s,t}$.

$$C_{s,t} = w_{s,t} \cdot C_{s,t}^* = w_{s,t} \cdot \frac{f_j(\bar{x}_{s,t})}{p_{s,t}(\bar{x}_{s,t})} = w_{s,t} \alpha_s^L c_{s,t} \alpha_t^E \quad (2.14)$$

Pro úplnost je nutno definovat rovnice pro řešení světelné (2.15), resp. kamerové cesty (2.16), a rovněž spojovací hranu. Ovšem tyto definice jsou velmi intuitivní, jelikož přímo vychází z rovnice [30] přenosu světla:

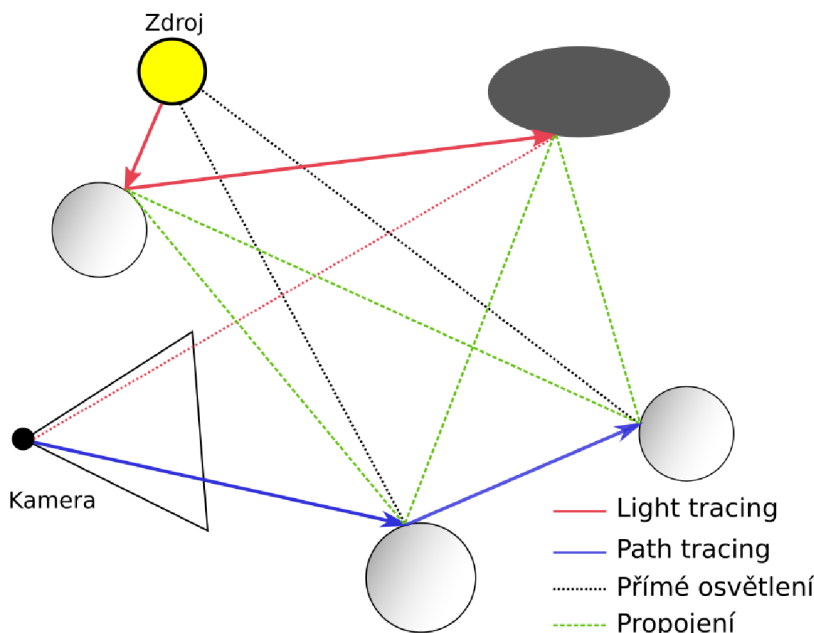
$$\begin{aligned} \alpha_0^L &= 1 \\ \alpha_1^L &= \frac{L_e^0(\mathbf{x}_0)}{P_A(\mathbf{y}_0)} \\ \alpha_i^L &= \frac{f_r(\mathbf{y}_{i-3} \rightarrow \mathbf{y}_{i-2} \rightarrow \mathbf{y}_{i-1})}{P_\sigma(\mathbf{y}_{i-2} \rightarrow \mathbf{y}_{i-1})} \alpha_{i-1}^L \text{ pro } i \geq 2, \end{aligned} \quad (2.15)$$

kde L_e^0 je plošná emitovaná záře ze světelného zdroje, P_A je plošná hustota pravděpodobnosti vzorkování zdroje v bodě \mathbf{x}_0 ve směru \mathbf{y}_0 a P_σ označuje rovněž hustotu pravděpodobnosti, ovšem vzhledem k směru. Podobně lze definovat i kamerovou cestu. [30]

$$\begin{aligned} \alpha_0^E &= 1 \\ \alpha_1^E &= \frac{W_e^0(\mathbf{z}_0)}{P_A(\mathbf{z}_0)} \\ \alpha_i^E &= \frac{f_r(\mathbf{z}_{i-1} \rightarrow \mathbf{z}_{i-2} \rightarrow \mathbf{z}_{i-3})}{P_\sigma(\mathbf{z}_{i-2} \rightarrow \mathbf{z}_{i-1})} \alpha_{i-1}^E \text{ pro } i \geq 2 \end{aligned} \quad (2.16)$$

Nakonec je potřeba ještě doplnit definici spojovací hrany (2.17) mezi světelnou a kamerovou cestou.

$$\begin{aligned} c_{0,t} &= L_e(\mathbf{z}_{t-1} \rightarrow \mathbf{z}_{t-2}) \\ c_{s,0} &= W_e(\mathbf{y}_{s-2} \rightarrow \mathbf{y}_{s-1}) \\ c_{s,t} &= f_r(\mathbf{y}_{s-2} \rightarrow \mathbf{y}_{s-1} \rightarrow \mathbf{z}_{t-1}) G(\mathbf{y}_{s-1} \leftrightarrow \mathbf{z}_{t-1}) \\ &\quad f_r(\mathbf{y}_{s-1} \rightarrow \mathbf{z}_{t-1} \rightarrow \mathbf{z}_{t-2}) \text{ pro } s, t > 0 \end{aligned} \quad (2.17)$$



Obrázek 2.8: Ukázka algoritmu BDPT

Z výše ukázaných rovnic je očividné, jak se spojovací hrana chová. Pro případ $c_{0,t}$ se jedná o přímé osvětlení ze světelného zdroje na bod kamerové cesty. Opačně při situaci $c_{s,0}$ jde o přímý paprsek ze světelné cesty do kamery (může být i přímo zdroj – kamera při $s = t = 0$). Nakonec obecná hrana je při situaci $c_{s,t}$. Jednotlivé hrany jsou graficky znázorněny na Obrázku 2.8.

Váhová funkce

Poslední částí, která nebyla definována v rovnici (2.13), je váhová funkce $w_{s,t}$. Jelikož se v algoritmu BDPT používá více vzorkovacích technik, kde tyto techniky běžně používají princip vzorkování dle podstatného výběru, angl. importance sampling, je nutno tyto techniky nějakým způsobem ohodnotit tak, aby výsledná hustota pravděpodobnosti co nejlépe vzorkovala hodnoty integrandu. K řešení tohoto problému se používá technika tzv. multiple importance sampling od E. Veache [31]. Tato technika formuluje několik heuristik, které různě nastavují váhy jednotlivým bodům cesty. Podle E. Veache je nejvhodnější použít mocninou heuristiku, angl. power heuristic, která je definována následovně:

$$w_{s,t} = \frac{p_s^2}{\sum_i p_i^2} = \frac{1}{\sum_i (p_i/p_s)^2} \quad (2.18)$$

Výsledná váha závisí na hustotách pravděpodobností, se kterými byla cesta $\bar{x}_{s,t}$ vygenerována všemi $s+t+1$ možnostmi. Platí zde konvence (2.19), že p_i označuje pravděpodobnost vygenerování $\bar{x}_{s,t}$ s použitím i světelných vzorků a $s+t-i$ kamerových vzorků:

$$p_i = p_{i,(s+t)-i}(\bar{x}_{s,t}) \text{ pro } i = 0, \dots, s+t \quad (2.19)$$

Pro dokončení definice je nutno ještě zmínit výpočet jednotlivých pravděpodobností. Celkovou hustotu pravděpodobnosti (2.20) lze rozdělit následovně:

$$p_{s,t}(\bar{x}_{s,t}) = p_s^L p_t^E, \quad (2.20)$$

kde jednotlivé hustoty jsou definovány následovně, viz vztah (2.21):

$$\begin{aligned}
p_0^L &= 1 & p_0^E &= 1 \\
p_1^L &= P_A(\mathbf{y}_0) & p_1^E &= P_A(\mathbf{z}_0) \\
p_i^L &= P_\sigma(\mathbf{y}_{i-2} \rightarrow \mathbf{y}_{i-1})G(\mathbf{y}_{i-2} \leftrightarrow \mathbf{y}_{i-1}) & p_i^E &= P_\sigma(\mathbf{z}_{i-2} \rightarrow \mathbf{z}_{i-1})G(\mathbf{z}_{i-2} \leftrightarrow \mathbf{z}_{i-1}) \\
&\text{pro } i \geq 2 & &\text{pro } i \geq 2
\end{aligned} \tag{2.21}$$

Zavedením váhové funkce z rovnice (2.18) lze dosáhnout maximálního možného výsledku v tomto algoritmu. Ačkoliv dnes již existují modifikace této metody (nejen samotného algoritmu, ale i váhových funkcí), tak se stále dá považovat za jednu z nejuniverzálnějších metod. Možná optimalizace může být, dle Georgieva et al. [6], při vyhodnocování vah MIS, kde není nutno počítat váhy až při zjištění kompletní cesty, ovšem progresivně za běhu, což umožňuje lepší paralelizaci. Další optimalizace přichází se strojovým učením [33], kdy není nutno vzorkovat integrand čistě náhodně, ale naopak cíleně tak, aby paprsky směřovali do míst, kde je možnost nalézt světelný zdroj nebo naopak směřovat světelné paprsky do míst, které pomalu konvergují.

Irradiance caching

Další zajímavou technikou je *irradiance caching* (interpolace intenzity záření na základě cache) od Ward et al. [35]. Ačkoliv nelze říci, že se jedná přímo o fyzikálně konzistentní techniku, jelikož se tento algoritmus snaží, pomocí speciálního cache pro okolní irradianci, interpolovat nepřímé osvětlení pro difúzní povrchy. Nutno říci, že ač je tato technika velmi přesvědčivá, tak ji nelze řadit přímo mezi algoritmy řešící globální osvětlení. Z hlediska této práce je ovšem velmi zajímavá, jelikož zavádí několik důležitých konceptů. Tato technika typicky využívá dělení prostoru pomocí *octree* [18], které se využívá jako datová struktura obsahující samotné záznamy cache. Sestavení této struktury může probíhat různými způsoby, např. náhodným procházením scénou a ukládáním hodnot irradiance $E(\mathbf{p})$, kde \mathbf{p} označuje konkrétní bod.

Hlavní částí algoritmu je interpolace nepřímého osvětlení z cache, kde v první fázi je nutno z datové struktury získat dostatečný počet záznamů, aby bylo možné interpolaci provést. Následně lze ze získaných záznamů [35, 14] interpolovat, podle rovnice (2.22):

$$E(\mathbf{p}) = \frac{\sum_{i \in S} E_i(\mathbf{p}) \cdot w_i(\mathbf{p})}{\sum_{i \in S} w_i(\mathbf{p})}, \tag{2.22}$$

kde w_i je váhová funkce, která se může podle potřeb lišit, typicky je ve tvaru dle rovnice (2.23), S je množina přípustných záznamů $S = \{i : w_i(\mathbf{p}) > 1/a\}$, kde a je uživatelem definovaná restrikce pro eliminaci záznamů, jejichž chyba je větší než požadované minimum. Tato vlastnost přímo vyplývá z rovnice (2.23) [35, 14], jelikož platí, že $w_i = 1/\text{chyba}$.

$$w_i(\mathbf{p}) = \frac{1}{\frac{\|\mathbf{p} - \mathbf{p}_i\|}{R_i} + \sqrt{1 - N(\mathbf{p}) \cdot N(\mathbf{p}_i)}} \tag{2.23}$$

Nutno doplnit, že ve výše definované rovnici označuje R_i průměrnou vzdálenost směrem k sousedním povrchům v bodě \mathbf{p}_i , což je poměrně specifická vlastnost pro tuto zvolenou funkci. V praxi lze pro zpřesnění využít i další možné heuristiky, gradienty a jiné prostředky.

Rozšířením této metody vznikl podobný algoritmus, který kromě toho, že řeší difúzní povrchy, přidává možnost řešení nízko frekvenčních lesklých BRDF, což Wardova metoda

neřeší. Název této techniky je *radiance caching* [14]. Směrová radiance je reprezentována pomocí hemisférických harmonických funkcí.

Další zajímavé metody

Kromě výše zmíněných algoritmů, lze nalézt desítky dalších publikací, které definují různé metody či akcelerace v této oblasti. Jedním ze zajímavých metod je technika od Georgiev et al. [6], která kombinuje výhody PM a BDPT algoritmu dohromady. Zjednodušeně se jedná o estimátor, který vhodnými MIS vahami kombinuje tyto techniky, což u BDPT znamená možnost použití světelných cest z okolních pixelů, čímž se výrazně zvýší konvergence u SDS cest.

Další zajímavá metoda je *metropolis light transport* [32, 30], zkráceně MLT, který využívá upravený Metropolis–Hastingsův algoritmus pro řešení zobrazovací rovnice. V podstatě na základě statistiky a již prozkoumaného prostoru modifikuje vytvořené cesty tak, aby byla simulace úspěšná i v místech, kde je konvergence velmi pomalá.

Posledním zmíněným případem není ani tak metoda jako taková, nýbrž akcelerace algoritmů založených na sledování cest. Vorba et al. [33] publikovali techniku, kde pomocí směsice Gaussových funkcí jsou schopni reprezentovat rozložení pravděpodobnosti odrazu paprsku na základě naučených parametrů. V podstatě se tím snaží o to, aby difúzní odrazy byly směrovány do oblastí, kde je reálná šance na nalezení nepřímého osvětlení. Nebo při sledování cesty ze světelného zdroje tam, kde scéna pomalu konverguje. Toto zapojení strojového učení výrazně urychluje konvergenci scén se složitým nepřímým osvětlením.

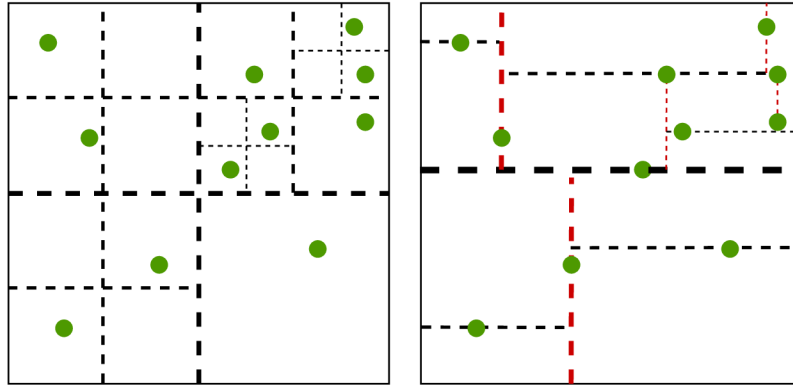
2.6 Struktury pro dělení prostoru

V matematice, ostatně i v počítačové grafice, se dělením prostoru myslí proces, který rozdělí danou oblast na několik disjunktních oblastí, jež se mohou rekurzivně dělit dále. Cílem tohoto procesu je vytvořit takovou strukturu, která značně urychlí vyhledávání v daném prostoru. Klasickým příkladem je *octree*, *kd-tree* nebo *BVH* (Bounding Volume Hierarchy).

Konkrétní použití těchto struktur v počítačové grafice je velmi široké. Typicky, u fotorealistického zobrazování, lze využít např. *kd-tree* pro akceleraci průsečíku paprsku s trojúhelníkem. Konkrétní trojúhelníky lze ohraničit obalovým tělesem (většinou osově zarovnaným) a uložit do listů stromů, následný průchod stromem je z hlediska časové složitosti výhodnější, než lineární prohledávání. Jiný příklad použití je metoda *irradiance caching* [35], kde do listů struktury *octree* jsou ukládány vzorky pro interpolaci.

Octree

Octree [18] je struktura typu strom, ve které každý uzel má přesně 8 potomků, pokud se nejedná o listový uzel. Konstrukce stromu se provádí typicky rekurzivně, kdy je prostor rozdělen ve všech osách uprostřed z daného rozsahu, ačkoliv existují i varianty, kde je vybrán konkrétní bod (např. pomocí mediánu) v němž se rozdělení provede. Tímto řezem se ve 3D daný prostor rozdělí právě na 8 potomků o stejné velikosti. Dělení probíhá až do té doby, dokud v každém uzlu není pouze jeden bod. Případně lze dělení omezit hloubkovým kritériem, čímž se ovšem posléze zpomalí vyhledávání. Ukázka rozdělení mračna bodů je na Obrázku 2.9 vlevo.



Obrázek 2.9: Ukázka rozdělení mračna bodů pomocí octree (pro přesnost, ve 2D se jedná o quadtree, což je analogie octree, pouze s nižší dimenzí) na obrázku vlevo, a současně pomocí kd-tree vpravo

Jelikož se jedná o stromovou strukturu, tak vyhledávání využívá algoritmus DFS (Depth First Search). Z hlediska implementace je nutno o stromu pouze znát ohraničující oblasti, jelikož dělicí místa lze jednoduše dopočítat během vyhledávání.

Kd-tree

Druhým zástupcem struktur dělicích prostor je kd-strom [34] (kd-tree). Význačností této struktury je zejména způsob dělení prostoru, čímž lze tento strom využít v libovolném množství dimenzí. Uvažujeme-li N dimenzionální problém, tak v případě kd-stromu je prostor postupně rozdělován podle osy x_1, x_2, \dots, x_N , tak, že se bodem dělení proloží hyper-rovina rovnoběžná právě s vybranou osou. Bod dělení je typicky určen mediánem souřadnic daných bodů v konkrétní ose. Tímto způsobem lze dosáhnout vyváženého stromu. Podobně jako u octree, dělení probíhá až do doby, kdy v každém vzniklém podprostoru zůstane pouze jediný bod. V některých implementacích může být bod dělení vybírán jiným způsobem, avšak tím pak není zaručena vyváženost stromu. Ukázka kd-stromu nad mračnem bodu je na Obrázku 2.9 vpravo.

Tato struktura lze rovněž upravit a použít pro akceleraci průsečíku paprsku s trojúhelníkem, kdy do každé úrovně stromu je, namísto bodu, ukládáno obalové těleso daného prostoru [34], a samotné trojúhelníky se nacházejí až na úrovni listových uzlů. Stavba takového stromu již není triviální a vyžaduje komplexnější heuristiky, např. *surface area heuristic*. Popis této problematiky je ovšem nad rámec této práce a čtenář je odkázán na odbornou literaturu, viz Havran & Wald [34].

Kapitola 3

Matematický aparát v realistickém zobrazování

Tato část definuje potřebné a užitečné matematické konstrukce, které jsou následně využity v dalších kapitolách. Kapitola se zaměřuje zejména na Monte Carlo techniky a k tomu potřebné generátory náhodných čísel. Dále je ukázán aparát pro modelování sférických, resp. hemisférických, funkcí, pomocí sférických harmonických bázevých funkcí, a příslušná technika podstatného výběru. Cílem této kapitoly není popsat všechny matematické techniky v dané problematice, ale ukázat potřebné metody, které jsou použité v následujících kapitolách.

3.1 Generátory pseudo-náhodných čísel

Pseudo-náhodné generátory čísel (zkráceně PRNG) jsou kompletně deterministické algoritmy pro generování posloupnosti čísel, jejichž distribuce není čistě náhodná, zatímco je často akceptovatelná mnoha aplikacemi a současně může být efektivně vyčíslena z počáteční hodnoty, tzv. seed. Existuje mnoho vlastností, které určují kvalitu generátoru. Klíčová je zejména **perioda**, která říká, po kolika vzorcích se začne sekvence opakovat. Druhou důležitou vlastností je **rovnoměrnost distribuce**, tedy, že rozložení vzorků je rovnoměrné po celé doméně. PRNG se uplatňují nejen v numerických výpočtech založených na metodě Monte Carlo apod., ale i například v kryptografii, kde lze využít kombinaci generátoru s vysokou periodou a pseudo-náhodným volením počáteční hodnoty.

Lineární kongruentní generátor

Nezákladnějším typem PRNG je lineární kongruentní generátor. Jak již název napovídá, výpočet následujícího vzorku je definován pomocí lineárního rekurentního vztahu [4], viz rovnici (3.1).

$$X_{n+1} = (A * X_n + B) \pmod{m} \quad (3.1)$$

Rozsah hodnot daného LCG generátoru je m , ovšem pouze tehdy, je-li perioda generátoru maximální. Toho lze dosáhnout splněním určitých podmínek. Existují různé studie [4, 20] vhodnosti konstant A, B pro různě široké datové typy. Nevýhodou ovšem je, že při špatně zvolených konstantách se projevuje značná závislost vzorků, což lze pozorovat např. promítnutím bodů do roviny či prostoru (pomocí více vzorků), kde při špatně zvolených hodnotách lze pozorovat rovnoběžné přímky, resp. roviny.

Xorshift

Druhým typickým zástupcem PRNG jsou Xorshift [16, 20] generátory. Z názvu je patrné, že jsou založeny na operátoru exkluzivního OR a bitovém posunu. Formálně jsou podmnožinou *linear-feedback shift registers* (LSFR), které právě umožňují efektivní implementaci na dnešních moderních počítačích. Proto jsou tyto generátory označovány jako nejrychlejší. Podobně jako u LCG (rovněž jako u LSFR) je ovšem nutno zvolit správné parametry za účelem získání velké periody. Níže je v ukázková jednoduchá implementace generátoru v jazyku C [16], viz Výpis 3.1.

```
uint32_t xorshift32(struct xorshift32_state* state)
{
    uint32_t x = state->a;
    x ^= x << 13;
    x ^= x >> 17;
    x ^= x << 5;
    return state->a = x;
}
```

Výpis 3.1: Ukázka implementace XOR-Shift generátoru

Z výše uvedeného kódu lze vidět, že pro implementaci rychlého xorshift generátoru stačí několik málo operací XOR a bitshift. Konkrétně pro 32-bitové stavy je nutno použít 3x XOR a 3x bitshift, dle Marsaglia et al. [16].

Mersenne Twister

Nejvíce využívaným generátorem pseudo-náhodných čísel je bez pochyby Mersenne Twister [17], jehož jméno je odvozeno na základě délky periody, jenž je volena jako tzv. Mersennovo prvočíslo (tedy číslo o 1 menší než nejbližší mocnina dvou). Algoritmus generuje čísla v rozsahu $[0, 2^w - 1]$, kde w je bitová velikost daného slova. Nejčastěji se využívá 32-bitový generátor MT19937, jenž má periodu $2^{19937} - 1$. Výhodou tohoto generátoru je již zmíněná velká rozmanitost periody, ale současně i rovnoměrnost distribuce, která splňuje mnoho důležitých statistických testů pro měření náhodnosti. Nevýhodou ovšem je, že např. vůči Xorshift generátorům, je MT pomalejší a složitější na implementaci.

Kvazi-náhodné sekvence

Speciálním případem generování pseudo-náhodných sekvencí jsou posloupnosti s nízkým nesouladem, též známé jako kvazi-náhodné (v angl. low-disperancy sequences). Tato vlastnost souvisí s rovnoměrností distribuce (ekvidistribuce), kdy posloupnost je ekvidistribuovaná právě tehdy, je-li její nesoulad limitně roven nule [20]. Tyto sekvence, a jejich generátory, jsou výhodné v situacích, kdy ekvidistribuce hraje důležitou roli, jelikož PRNG může vzorky shlukovat a tím zvyšovat nesoulad. Z praktického hlediska mohou být použity např. v numerické integraci, pro rovnoměrné pokrytí integrandu. Významné sekvence jsou např. van der Corput, Hammersley, Halton či Sobol. Pro detailnější definice je čtenář odkázán na příslušnou literaturu [20].

3.2 Monte Carlo integrace

Technika řešení integrálu pomocí metody Monte Carlo je běžně užívanou numerickou metodou. Přibližná hodnota integrálu je odhadnuta na základě náhodného vzorkování integrandu. Naproti tomu, většina běžných postupů volí distribuci vzorku integrandu dle určité mřížky nebo vzoru.

Funkci $f : \Omega \rightarrow \mathbb{R}$ lze s využitím metody Monte Carlo [20], za pomoci N náhodných vzorků, aproximovat jako:

$$I = \int_{\Omega} f(x) dx = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}, \quad (3.2)$$

kde x_i je náhodná veličina s hustotou pravděpodobnosti $p(x_i)$.

V této metodě velmi závisí na zvolené distribuci náhodných vzorků. Nejčastějším případem je uniformní rozložení po celé doméně integrandu. To lze efektivně využít v případě, že je funkce dopředu neznámá, ovšem v opačném případě je vhodné zvolit pokročilejší distribuci. Prakticky je užitečné vzorkovat hustěji místa, kde je funkční hodnota vyšší, tedy tam, kde vzorek ovlivní hodnotu integrálu nejvíce. Tomuto se říká vzorkování technikou podstatného výběru, anglicky nazvané *importance sampling*. Tento princip lze ještě rozšířit o tzv. vzorkování po částech (angl. stratified sampling), kdy je daná doména rozdělena na intervaly, ve kterých je integrace prováděna nezávisle. Toto rozdělení se provádí zejména uniformně.

V mnoha aplikacích je ale velice obtížné sestavit hustotu pravděpodobnosti, která by kopírovala hodnoty funkce, protože velmi často není tvar funkce znám a evaluace hodnot v jednotlivých vzorcích není levnou operací.

Metoda Monte Carlo může být rovněž upravena pro práci s kvazi-náhodnými sekvencemi, poté říkáme QMC (Quasi Monte Carlo) [20]. V mnoha situacích bývá vhodnější využít právě QMC, jelikož sekvence s malým nesouladem spolehlivěji pokryjí celou doménu integrandu, aniž by způsobily velké statistické chyby.

3.3 Sférické harmonické funkce

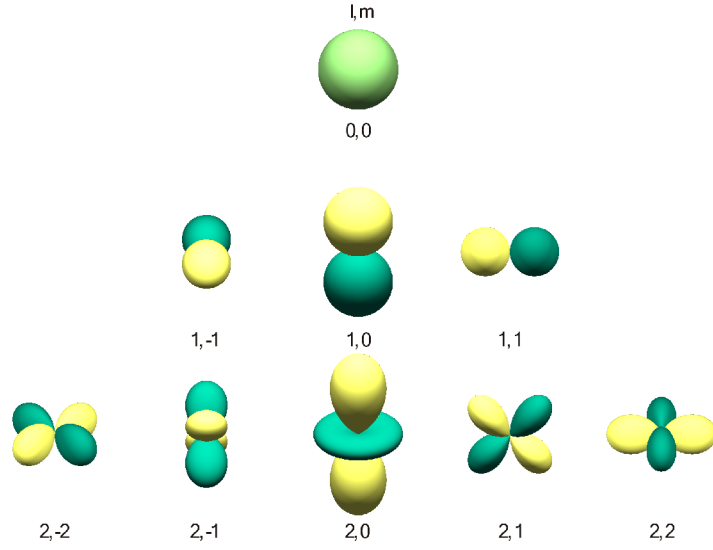
Sférické harmonické funkce (zkráceně jen SH, někdy známé také jako Laplaceovy SH funkce) jsou speciální funkce, které jsou kompletně ortogonální na celé sférické doméně. Z toho důvodu jsou vhodné pro reprezentaci funkcí na povrchu koule. Jsou to báze funkce ze skupiny $SO(3)$. Sférické harmonické jsou důležité v mnoha teoretických i praktických aplikacích, např. reprezentace elektromagnetického či gravitačního pole. V počítačové grafice se využívají zejména v oblasti nepřímého osvětlení – např. předpočítané přenosy světla [13, 27].

SH jsou založené na asociovaných Legendrových polynomech $P_l^m(x)|_{x=\cos \theta}$, které lze definovat pomocí rekurentních vztahů [7]¹, viz rovnici (3.3).

$$\begin{aligned} P_m^m(x) &= (-1)^m (2m-1)!! (1-x^2)^{m/2} \\ P_{m+1}^m(x) &= x(2m+1)P_m^m \\ (l-m)P_l^m(x) &= x(2l-1)P_{l-1}^m - (l+m-1)P_{l-2}^m \end{aligned} \quad (3.3)$$

Pokud je směrový vektor $\vec{\omega}$ reprezentován ve sférickém souřadném systému pomocí standardních parametrů, $\vec{\omega} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$, potom lze reálné sférické harmonické

¹operátor !! znamená dvojitý faktoriál



Obrázek 3.1: Sférické harmonické báze funkce, do řádu $l = 2, m = 2$

báze funkce definovat vztahem v rovnici (3.4):

$$y_l^m(\theta, \phi) = \begin{cases} \sqrt{2}K_l^m P_l^m(\cos \theta) \cos(m\phi) & m > 0 \\ \sqrt{2}K_l^m P_l^{-m}(\cos \theta) \sin(-m\phi) & m < 0 \\ K_l^0 P_l^0(\cos \theta) & m = 0 \end{cases}, \quad (3.4)$$

kde P_l^m jsou již zmíněné asociované Legendrovy polynomy (ALP) a K_l^m jsou normalizační konstanty definované jako:

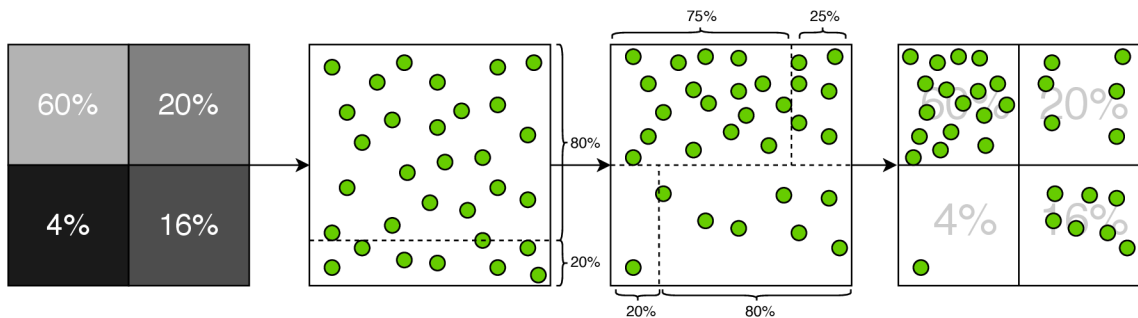
$$K_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}} \quad (3.5)$$

Konstanty l a m jsou celočíselné hodnoty, kde l označuje řád a m stupeň sférických harmonických báze funkcí. Formálně je nutno, aby l bylo nezáporné a m splňovalo rozsahové kritérium $-l \leq m \leq l$. Ukázka báze funkcí je na Obrázku 3.1, kde žlutá barva indikuje kladnou hodnotu a zelená zápornou.

Každá reálná sférická funkce $f(\theta, \phi)$ může být aproximována jako $\tilde{f}(\theta, \phi)$ pomocí lineární kombinace SH báze funkcí (3.6):

$$f(\theta, \phi) \approx \tilde{f}(\theta, \phi) = \sum_{l=0}^{N-1} \sum_{m=-l}^l f_l^m \cdot y_l^m(\theta, \phi) \quad (3.6)$$

Koeficienty f_l^m jsou získány *projekcí* f do každé báze funkce y_l^m . Sférické harmonické funkce jsou rovněž ortonormální a invariantní vůči rotacím [7], což umožňuje efektivní tzv. alias-free projekci a rekonstrukci metodou nejmenších čtverců. Avšak naproti tomu jsou sférické harmonické značně limitované co se týče metody podstatného vzorkování (angl. importance sampling, IS). V současné době existuje pouze jedna technika tohoto typu vzorkování, představená W. Jaroszem [10], založená na hierarchickém deformování a integrování vzorků (podobně jako IS u Haar vlněk).



Obrázek 3.2: Ukázka dvou kroků hierarchického warpování na základě integrálů kvadrantů. V prvním kroku je provedeno warpování podle vertikální osy, na základě pravděpodobností v daném řádku, v druhém kroku je provedeno dělení v horizontální ose, dle pravděpodobností jednotlivých buněk.

Podstatný výběr (importance sampling)

Jak již bylo zmíněno, pro efektivní vzorkování funkcí reprezentovaných pomocí SH je nutno zavést podstatný výběr, aby se předešlo neefektivním způsobům vzorkování, jako např. rejection sampling a jiné. Ačkoliv SH jsou v tomto ohledu značně limitovány, tak současně existuje jediná efektivní metoda pro podstatné vzorkování, viz Jarosz et al [10].

Tato metoda je odvozená od podstatného výběru nad diskretní vlnkovou transformací s použitím Haar vlnek. Vstupem je množina uniformně rozložených bodů na kouli, které pomocí hierarchického warpování dosáhnou dané distribuce. Prostor koule je rozdělen vždy na 4 kvadranty, kde pro výpočet pravděpodobnosti warpování daného kvadrantu se využívá hodnoty integrálu kvadrantu. Tento krok je opakován, dokud se nedosáhne dostatečně přesné distribuce nebo dokud se nenarazí na dostatečnou hloubku, viz Obrázek 3.2.

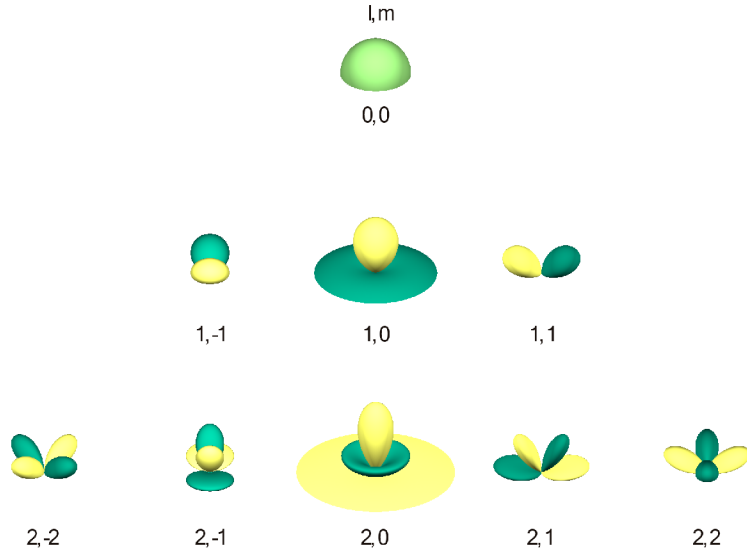
Výpočet integrálu funkce, resp. jednotlivých kvadrantů, se může zdát jako obtížný, avšak díky separabilitě SH báží na θ -závislé a ϕ -závislé prvky, a při zavedení cylindrických souřadnic, je integrál dané báze definován relativně jednoduše: [10]

$$\hat{y}_l^m = \int \int y_l^m(\theta, \phi) \sin \theta d\theta d\phi = K_l^m \left(\int_{z_-}^{z_+} P_l^{|m|}(z) dz \right) \left(\int_{\phi_-}^{\phi_+} \Phi^m(\phi) d\phi \right) \quad (3.7)$$

Rovnice (3.7) je rozdělena na již zmíněné nezávislé komponenty, kde normalizační konstanta zůstává stejná (3.5), pro ALP jsou zavedeny rekurentní vztahy pro výpočet integrálu [10] a Φ komponenta je pouze integrál goniometrické funkce, který může být jednoduše tabelován, taktéž pomocí rekurentních vztahů. Hlavní výhodou tohoto postupu je, že všechny integrály (kromě integrálu celé funkce v závislosti na koeficientech) mohou být vyhodnoceny předem, s použitím 1D mip-map pro danou hloubku. Integrál celé funkce je pak pouze vážená suma integrálu jednotlivých bazových funkcí, dle rovnice (3.7). Hustota pravděpodobnosti vzorkování každého vzorku je dána podílem integrálu obsahujícího kvadrantu a integrálu celé funkce.

Hemisférické harmonické funkce

Ačkoliv sférické harmonické zahrnují celou sférickou doménu, v některých aplikacích je vhodné se omezit pouze na hemisféru. První hemisférické harmonické bazové funkce zavedl Gautron et al. [5] (viz Obrázek 3.3). Tyto bazové funkce jsou stále založené na asociova-



Obrázek 3.3: Hemisférické harmonické, do řádu $l = 2, m = 2$

ných Legendrových polynomech, ovšem s posunutým argumentem (3.8), pomocí lineární transformace.

$$\tilde{P}_l^m(\cos \theta) = P_l^m(2 \cos \theta - 1) \text{ kde } \theta \in [0, \frac{\pi}{2}) \quad (3.8)$$

Hemisférické harmonické funkce (dále jen HSH) jsou ortogonální přes $\langle 0, \frac{\pi}{2} \rangle \times \langle 0, 2\pi \rangle$, tedy je nutno posléze upravit i korespondující normalizační koeficient \tilde{K}_l^m .

$$\tilde{K}_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{2\pi(l+|m|)!}} \quad (3.9)$$

Transformací ALP argumentu a změnou normalizační konstanty lze již zapsat obecnou HSH bázovou funkci [5] H_l^m , pomocí odvození z SH definice, viz následující vztah (3.10).

$$H_l^m(\theta, \phi) = \begin{cases} \sqrt{2}\tilde{K}_l^m \tilde{P}_l^m(\cos \theta) \cos(m\phi) & m > 0 \\ \sqrt{2}\tilde{K}_l^m \tilde{P}_l^{-m}(\cos \theta) \sin(-m\phi) & m < 0 \\ \tilde{K}_l^0 \tilde{P}_l^0(\cos \theta) & m = 0 \end{cases} \quad (3.10)$$

Podobně jako u SH, i tyto bázové funkce jsou invariantní vůči rotacím, ačkoliv vzhledem k povaze těchto bází jsou rotace komplikovanější (zejména proto, že v Y ose se jedná o ztrátovou rotaci). Typicky se rotace provádí rozšířením na sférickou doménu, rotováním a následnou redukcí zpátky na hemisféru.

Kapitola 4

Existující řešení fotorealistického zobrazování

V dnešní době již existuje spousta řešení realistického zobrazování. Většina již zmíněných je ovšem zatížena komerční licencí a není tedy volně dostupná. Naopak existují i takové, které jsou volně dostupné ovšem za cenu toho, že nejsou tolik propracované a existují spíše jako způsob pro srovnání implementací různých vědeckých článků. Tato kapitola popisuje několik programů právě z komerční, tak i z nekomerční, resp. open-source, sféry, avšak neklade si za cíl být encyklopedickým přehledem softwarových produktů.

4.1 Komerční software

Komerčního řešení fotorealistického zobrazování je opravdu mnoho, v této části je uveden stručný popis těch nejznámějších a nejvíce využívaných.



Obrázek 4.1: Ukázka výstupu z V-Ray



Obrázek 4.2: Obrázek získaný pomocí Octane



Obrázek 4.3: Velmi pokročilá scéna vykreslená pomocí Maxwell

V-Ray

Jedním z nejvíce využívaných, a zejména nejvíce prodávaných, komerčních řešení je V-Ray, viz Obrázek 4.1. První vydání bylo již v roce 1997. Od té doby se tento software vyvíjí a dnes je napsaný formou balíčku (plug-in) pro klasické 3D programy typu Blender, Cinema4D či Autodesk 3Ds Max. Jeho velká rozšířenost se ovšem odrazila v pořizovacích nákladech. V-Ray implementuje zejména algoritmy založené na sledování cest, akcelerované pomocí GPU (konkrétně přímo path tracing a photon mapping), ale i jiné pokročilé metody, jako např. irradiance caching.

Octane Renderer

První verze Octane byla vydána v roce 2012, kde se jednalo o první software, který plně akceleroval celý výpočet pomocí grafické karty, jelikož se jedná o určitou formu algoritmu distribuovaného sledování paprsku. Program je relativně oblíbený mezi architekty a konstruktéry (viz Obrázek 4.2), protože nabízí velké množství předdefinovaných dat. Podobně jako u V-Ray, je jeho pořizovací cena relativně vysoká. Značná nevýhoda ovšem je, že Octane je striktně vyvíjen pouze pro platformu NVIDIA CUDA. Podle vývojářů je ovšem v plánu přeprogramovat program na hardwarovou nezávislost pomocí technologie Vulkan.

Maxwell

Dalším zajímavým komerčním produktem je Maxwell Renderer, který byl původně vydán v roce 2004 jako implementace variace metropolis light transport algoritmu, ovšem za roky vývoje došlo k značnému předělání a současně není známo jaký algoritmus je pro zobrazování využit. Zajímavé na tomto software je, že oproti předchozím zmíněným, Maxwell bývá využíván pro filmové efekty či animace. Naproti tomu rovněž nabízí možnost využití jako

plug-in do 3D programů, což umožňuje využít ho rovněž i v architektuře či designu, ostatně to lze vidět na Obrázku 4.3.

4.2 Vědecky zaměřené open-source projekty

Oproti komerční sféře, open-source projektů není takové množství, ačkoliv hrají důležitou roli ve vědecké sféře, s účelem sloužit jako určitý druh pro porovnávání implementací různých algoritmů v různých odborných publikacích a podobně.

Mitsuba Renderer

Velmi zajímavým projektem je Mitsuba Renderer, který původně založil W. Jakob. Tento software nabízí mnoho možností zejména co se týče reprezentace povrchu, obsahuje téměř všechny existující BRDF/BTDF. Současně implementuje mnoho různých zobrazovacích metod, od velmi jednoduchých, po ty velice složité. Většina metod je implementována na CPU, ovšem některé jsou už i akcelerovány pomocí GPU. Z hlediska návrhu je vytvořený zcela modulárně, což znamená, že je relativně jednoduché integrovat novou komponentu. Nevýhodou je, že neexistuje příliš dobrá implementační dokumentace, ačkoliv naopak obsahuje velmi podrobný popis možného nastavení vstupní scény. Mitsuba nabízí kromě textového prostředí i grafické rozhraní (viz Obrázek 4.4).

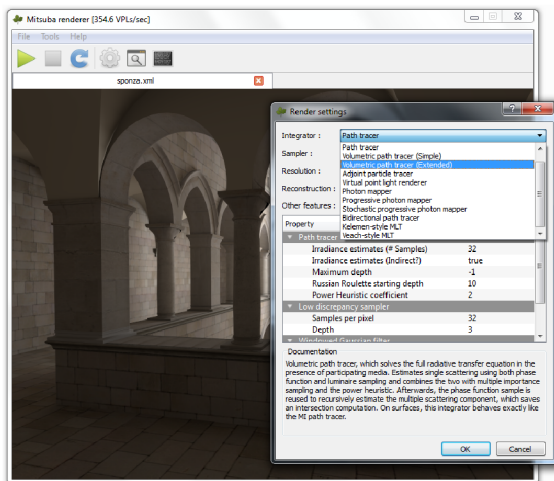
V současné době byla vydána již druhá verze, Mitsuba 2 [21], která je aktuálně napsaná v moderním C++17. Tato verze je stále kompatibilní se starší verzí, tedy všechny dříve definované scény lze stále plně zpracovávat, ovšem marginálně se změnilo vývojové prostředí. Hlavní výhodou nové verze je jeho naprostá univerzálnost, kde lze framework využít od jednoduchého single-thread vykreslování, přes SIMD AVX-512 akcelerace, až k aplikacím s využitím RTX jader u moderních grafických karet. Nová verze již podporuje, kromě základní syntézy v modelu RGB, i vykreslování v celém spektru, či pouze v monochromatické složce.

Druhému zástupci v této kategorii je věnována speciální kapitola, kvůli jeho důležitosti v této práci, viz níže.

4.3 Systém pro fotorealistické zobrazování PBRT

Důležitým zástupcem, zejména v této práci, z kategorie open-source řešení, je PBRT (Physically Based Rendering), které bylo vydáno současně s knihou od M. Pharr et al. [23]. Celkově se jedná již o třetí vydání, kde v každé verzi se zaměřují na nové trendy v této oblasti. Výhodou je, že všechny kód je velmi detailně zdokumentovaný a všechny principy jsou vysvětleny v již zmíněné knize. Současně ke zdrojovým kódům jsou k dispozici i četné scény a naměřené parametry, které lze využít.

Implementuje všechny podstatné fotorealistické metody – **obousměrné sledování cesty**, **stochastické progresivní mapování fotonu** či **metropolis light transport**. PBRT taktéž velmi široce podporuje křivkové struktury pro modelování např. vlasů, trávy apod. Nedílnou součástí je i implementace realistické kamery, která umožňuje simulace spousty optických jevů, kterých lze různými objektivy dosáhnout. Nevýhodou může být, že se jedná čistě o CPU řešení, ovšem vzhledem k účelu využití, které je zejména v akademické sféře, to není tolik zásadní problém.



Obrázek 4.4: GUI v Mitsuba Renderer



Obrázek 4.5: Obrázek vykreslený pomocí PBRT

Dalším důležitým prvkem je možné vykreslování do HDR kontejneru, které umožňuje zachovat celý dynamický rozsah simulovaného světla ve scéně. Podobně jako Mitsuba, taktéž podporuje využití celého spektra nebo čistě RGB model.

Z pohledu vývojáře jsou zde široké možnosti pro jednoduchý vývoj vlastních komponent realistického zobrazování. Celé rozhraní projektu je kompletně abstraktní. Možný vlastní vývoj zahrnuje 10 základních komponent, např. geometrické tvary, integrátory, vzorkovače, kamery, světelné zdroje, filtry, materiály či volumetrická média.

Od roku 2018 je již zmíněná publikace [23] dostupná přímo na stránkách PBRT. Ukázkový výstup lze vidět na Obrázku 4.5. V současné době autoři pracují na 4. vydání jak knihy, tak implementace. Celkově se jedná o velmi zdařilou publikaci, která je ve světě hojně využívána, zejména při výuce na univerzitách, a za níž autoři získali mnoho ocenění.

Kapitola 5

Analýza a zhodnocení současného stavu fotorealistického zobrazování

V dnešní době lze kategorizovat zobrazovací algoritmy do mnoha odvětví dle použití. Fotorealistické zobrazování lze řadit zejména do filmového průmyslu, případně do architektury a strojího průmyslu. Herní průmysl je stále ovládán rasterizací trojúhelníků, díky její rychlosti a flexibilitě. Cílem této kapitoly není předložit čtenáři kompletní hodnocení všech metod, nýbrž sestavit analýzu a hodnocení některých z nich.

5.1 Hodnocení metod pro realistické zobrazování

V oblasti realistického zobrazování není jednoduché vytvořit objektivní hodnocení, jelikož existuje mnoho faktorů, dle kterých lze měřit kvalitu algoritmů. Ať už se jedná o rychlost nebo naopak o fyzikální přesnost, implementační jednoduchost či úroveň paralelizace na grafické kartě. Z hlediska použití lze konstatovat, že mezi nejčastěji využívané algoritmy patří zejména ray tracing, jelikož se v dnešní době objevují grafické karty, které tuto techniku již explicitně řeší nebo zejména kvůli algoritmické jednoduchosti, která stále ještě umožňuje v určité formě zobrazování v reálném čase. Ovšem jak již bylo řečeno, nejedná se o fyzikálně konzistentní techniku, která v základní formě ani neřeší celkové globální osvětlení. Druhým hlavním kandidátem, který se objevuje zejména v komerčních řešeních pro filmový průmysl apod., je algoritmus path tracing a jeho varianty. Méně zřídka lze nalézt použití algoritmu SPPM či MLT, ačkoliv např. výše zmíněný Mitsuba Renderer se zaměřuje zejména na varianty MLT.

	Rychlost	Měkké stíny	Color bleed	Kaustiky	SDS cesty	Impl.
RT	*****	-	-	-	-	*****
PT	****	*****	*****	**	-	***
PM	****	****	***	****	****	***
SPPM	***	****	***	****	*****	**
BDPT	***	*****	*****	****	**	**
MLT	***	****	****	***	***	*

Tabulka 5.1: Srovnání jednotlivých metod, **RT** – ray tracing, **PT** – path tracing, **PM** – photon mapping, **SPPM** – stochastický progresivní PM, **BDPT** – obousměrný PT, **MLT** – metropolis light transport

Výše zobrazená Tabulka 5.1 ukazuje jednoduché srovnání několika vybraných metod dle základních parametrů. Zejména je zajímavé sledovat rozdíly mezi rychlostí a implementací různých algoritmů, posléze i efekty, které zvládají. Lze konstatovat, že různé metody jsou vhodné na různé situace. Konvergence výše zmíněných algoritmů je dost často závislá na charakteru scény, kterou se snaží vykreslit, jelikož zatím neexistuje univerzální algoritmus, který by perfektně zvládal všechny realistické efekty v rozumném čase a s rozumnou mírou implementační či paměťové složitosti.

5.2 Cíl realizační části práce

O téma fotorealisticke zobrazování se zajímám již od své bakalářské práce. Ačkoliv v tomto oboru je sice již mnoho vyřešeno, tak stále je zde prostor pro nové věci, zejména s cílem celkového urychlení. Z toho důvodu se v této práci zaměřuji na návrh a implementaci rozšíření do metody sledování cest s cílem urychlit celkový výpočet.

Cílem této práce bylo nastudovat problematiku fotorealisticke zobrazování, zejména Monte Carlo metody, analyzovat již existující řešení a posléze navrhnout a implementovat experimentální způsob akcelerace redukce šumu pomocí modelování směrových veličin. Současně je zahrnuto vyhodnocení a diskuze potenciálního dalšího rozvoje

5.3 Technické parametry práce

Rozhodl jsem se jako základ implementační části práce využít open-source software PBRT, ve kterém jsou implementovány jednotlivé dílčí komponenty (reprezentace scény, kamery, objektů aj.). Nad tímto systémem byl vystavěn program pro modelování libovolné sférické, resp. hemisférické, funkce pomocí ortogonálních radiálních bázových funkcí, podstatný výběr (importance sampling) nad těmito modely a posléze interpolací nových funkcí, na základě již existujících. Následně byly tyto komponenty zapojeny do Monte Carlo integrátoru typu path tracing, kde promítané funkce jsou reprezentovány pomocí směrové záře (radiance). Je zde snaha o zrychlení konvergence algoritmu cíleným směřováním paprsků do míst, odkud přichází nejvíce světla. Následující výčet shrnuje jednotlivé prvky implementace:

- HSH reprezentace směrové záře
 - Monte Carlo projekce do bázových funkcí
- podstatný výběr nad HSH
 - offline předpočítání jednotlivých separovaných částí
 - hierarchické „warpování“
- interpolace koeficientů HSH pomocí cache ve scéně
 - zahrnuje i stromovou strukturu pomocí které je modelováno cache, v němž je třeba dané funkce ukládat
- zapojení do Monte Carlo integrátoru
 - tvorba fotonové mapy pro HSH projekci
 - online plnění interpolačního cache během sledování cest

Následující část práce pojednává o způsobu realizace aplikace jenž implementuje výše uvedené požadované vlastnosti.

Kapitola 6

Aproximace rozložení světla ve scéně

Tato kapitola se zabývá návrhem řešení aproximace rozložení světla ve scéně za účelem cílenějšího vysílání paprsků tak, aby se snížil celkový počet sledovaných nevýznamných cest. V první části je ukázána projekce libovolné sférické, resp. hemisférické, funkce do sférických harmonických bázevých funkcí. Dále je popsána úprava techniky hierarchického „warování“ pro hemisférické harmonické a v poslední části je ukázána interpolace koeficientů HSH ze sousedních vzorků.

6.1 Projekce do radiálních bázevých funkcí

Jak již bylo zmíněno v kapitole o sférických, resp. hemisférických, harmonických funkcích, projekce do těchto bázev je relativně triviální, jelikož se jedná o ortogonální fáze. Projekce libovolné funkce na dané doméně může být provedena mnoha způsoby. Typicky se snažíme o to, aby chyba dané aproximace byla co nejnižší, čehož lze docílit větším počtem bázevých funkcí a přesnějším odhadem koeficientů f_l^m . Teoreticky je tedy snaha o to aby se čtvercová chyba limitně blížila nule, viz rovnici (6.1).

$$\lim_{N \rightarrow \infty} \int_0^{2\pi} \int_0^\pi \left| f(\theta, \phi) - \sum_{l=0}^N \sum_{m=-l}^l f_l^m Y_l^m(\theta, \phi) \right|^2 \sin \theta \, d\theta d\phi = 0 \quad (6.1)$$

Prakticky toto ovšem není situace výpočetně výhodná. U mnoha funkcí, zejména tedy v počítačové grafice, je dostačující pouze aproximace s určitou chybou. V některých případech ani relativně velká chyba nemusí být znát.

Jeden ze způsobů získání koeficientů SH, resp. HSH, je řešení integrálu funkce vůči bázevým funkcím ve všech směrech, viz rovnici (6.2). Pro jednodušší orientaci je dvojný integrál přes úhel θ a ϕ sloučen do jednoho přes všechny úhly, jelikož existuje snadný převod mezi kartézskou a sférickou formou.

$$f_l^m = \int_{\Omega} f(\omega) Y_l^m(\omega) \, d\omega \quad (6.2)$$

Tento způsob je v podstatě dostačující, ovšem výpočetně opět velmi náročný. Nejjednodušší způsob řešení je pomocí numerických metod. Zde je v podstatě opět více možností. Uvažujeme-li N vzorků pro numerickou aproximaci, tak lze sestavit matici hodnot

dané funkce v různých směrech a pomocí metody SVD (singulární rozklad) vyřešit metodu nejmenších čtverců a získat přímo koeficienty. Ačkoliv tento způsob je relativně přesný, jeho výpočetní náročnost nemusí být vždy vhodná.

Druhá možnost je využití metody Monte Carlo a náhodného vzorkování [14]. V podstatě se jedná o sumu součinů mezi funkční hodnotou a bázovou funkcí v dané konfiguraci při využití N náhodných směrů na kouli, viz rovnici (6.3).

$$f_l^m = \frac{4\pi}{N} \sum_{i=0}^N f(\theta, \phi) \cdot Y_l^m(\theta, \phi) \quad (6.3)$$

Přesnost výše uvedeného způsobu přímo závisí na počtu Monte Carlo vzorků. Jeliž ovšem ale některé funkce jsou užity pouze v rámci hemisférické domény, např. funkce nepřímého osvětlení na povrchu, je nutno pro úplnost definovat i odhad koeficientů pro hemisférické harmonické funkce [14], viz rovnici (6.4) (nutno podotknout, že i hemisférické funkce lze modelovat pomocí SH, ačkoliv to je nevhodné).

$$f_l^m = \frac{2\pi}{N} \sum_{i=0}^N f(\theta, \phi) \cdot H_l^m(\theta, \phi) \quad (6.4)$$

kde $2\pi = \frac{1}{p(\theta, \phi)}$ je převrácená pravděpodobnost vzorkování daného směru vztahovaná k polokouli, sloužící jako váhová funkce, viz vztah (3.2).

Dosadíme-li za funkci $f(\theta, \phi)$ konkrétní veličinu $L(\mathbf{x}, \omega)$, tedy zář v daném bodě a z konkrétního směru, lze takto promítnout do hemisférických harmonik přímo lokální rozložení světla v daném bodě ve scéně.

6.2 Úprava podstatného výběru pro hemisférické harmonické

Výše zmíněno, pro reprezentaci směrové záře (radiance) ve scéně je vhodné využít hemisférické harmonické funkce. Avšak metoda podstatného výběru je zatím definována pouze pro celou sférickou doménou a k ní přidružené bázové funkce.

Vzhledem k povaze této metody lze vyjít z rovnice (3.7), která reprezentuje integrál pomocí produktu integrálu ϕ , resp. z (v sférických souřadnicích θ) složky. Pro sférické harmonické je rozsah integrálu $z \in [-1, 1]$. Pro HSH lze tento interval upravit na $z \in [0, 1]$. Druhá nutná úprava spočívá v argumentu integrálu asociovaného Legendrova polynomu, v již zmíněné rovnici, $P_l^{|m|}(z)$, kde na základě definice lineární transformace argumentu u hemisférických harmonických bázi lze tento výraz přepsat na $\tilde{P}_l^{|m|}(z) = P_l^{|m|}(2z - 1)$. Upravený vztah pro výpočet integrálu HSH báze je nyní definován v rovnici (6.5).

$$\hat{h}_l^m = K_l^m \left(\int_{z_-}^{z_+} P_l^{|m|}(2z - 1) dz \right) \left(\int_{\phi_-}^{\phi_+} \Phi^m(\phi) d\phi \right) \quad (6.5)$$

Integrál celé funkce je následně definován jako suma součinu koeficientu HSH a integrálu příslušné bázové funkce, viz rovnici (6.6):

$$\int_{\Omega} \tilde{f}(\vec{\omega}) d\vec{\omega} = \sum_{l=0}^{n-1} \sum_{m=-l}^l f_l^m \underbrace{\int_{\Omega} h_l^m(\vec{\omega}) d\vec{\omega}}_{\hat{h}_l^m} \quad (6.6)$$

Analogicky jako u SH, lze i zde všechny komponenty vypočítat dopředu a vytvořit 1D mip-map hierarchii pro obě nezávislé komponenty a normalizační konstanty.

6.3 Interpolace vyzařovací funkce

Popis interpolace byl ukázán v sekci o irradiance caching [35], kde pomocí vážené sumy byla interpolována hodnota ozáření (irradiance). Tento princip (2.22) lze využít i pro interpolaci koeficientů radiálních bázových funkcí. Veličinu ozáření lze nahradit vektorem koeficientů dané funkce. Vztah pro interpolaci koeficientu je definován v následující rovnici (6.7):

$$\tilde{f}_l^m = \frac{\sum_{i \in S} f_{il}^m \cdot w_i(\mathbf{p})}{\sum_{i \in S} w_i(\mathbf{p})}, \quad (6.7)$$

kde množina přípustných záznamů bude nyní definována jako $S = \{i : w_i(\mathbf{p}) > 0\}$, což přímo vyplývá z rovnice určující váhu $w_i(\mathbf{p})$ [35, 14], viz rovnici (6.8).

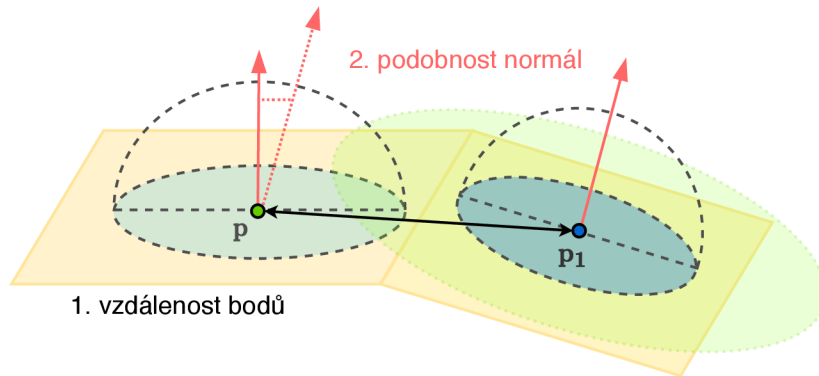
$$w_i(\mathbf{p}) = 1 - \max \left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{\text{clamp}(t, t_{min}, t_{max})}, \sqrt{\frac{1 - N(\mathbf{p}) \cdot N(\mathbf{p}_i)}{1 - \cos a_{max}}} \right) \quad (6.8)$$

Oproti rovnici (2.23) obsahuje výše zmíněná rovnice dvě heuristiky, jež značně limitují přípustnou množinu záznamů. Váhová funkce v podstatě funguje jako odečet největší chyby, tedy pokud je chyba vysoká, váha vzorku je velmi nízká. Funkce max vrací větší ze dvou hodnot.

V první heuristice se řeší chyba vzhledem k vzdálenosti. Jsou zde definovány tři parametry, t označuje maximální odhadnutou vzdálenost působnosti. Tato hodnota je následně oříznuta pomocí t_{min} a t_{max} z toho důvodu, aby daný vzorek nemohl např. kvůli numerické chybě ovlivňovat všechny vyzařovací funkce. Díky podílu absolutní vzdálenosti bodů (bod v kterém interpolujeme a konkrétní vzorek z S) s takto omezeným parametrem je jasné, že vzdálenější záznam vytvoří větší chybu.

Druhá heuristika řeší úhlovou podobnost normály v bodě interpolace a záznamu z množiny S . Parametr a_{max} je definované kritérium maximální úhlové odchylky (v radiánech). Analogicky, jako u první heuristiky, vyšší úhlové odchylky vygenerují větší chybu.

Praktická implementace interpolace pomocí vyhledávání nejbližších sousedů ve stromu octree je popsána v následující kapitole. Názorná ukázka heuristických kritérií, mezi dvěma polokoulemi na povrchu, je na Obrázku 6.1.



Obrázek 6.1: Ukázka interpolačních heuristik, kde \mathbf{p} je bod v kterém se interpoluje a \mathbf{p}_1 je jeden ze sousedů, z kterých se interpoluje. Zeleně je vyznačen radius působnosti bodu \mathbf{p}_1

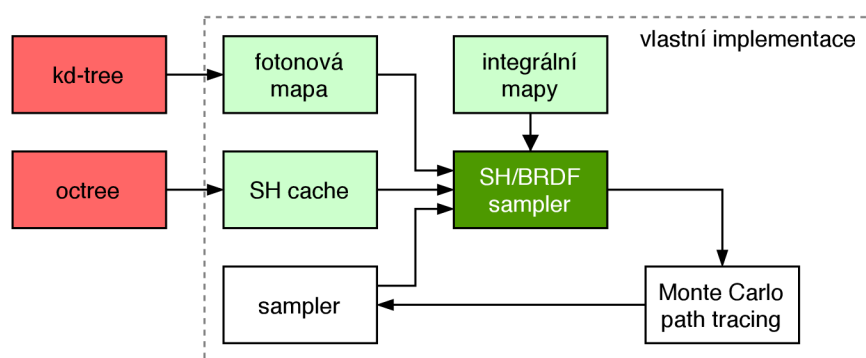
Kapitola 7

Implementace rozšíření do metody sledování cest

Tato kapitola popisuje implementační detaily vzorkovače využívající lokální aproximace rozložení světla ve scéně pomocí hemisférických harmonických funkcí. Tento vzorkovač je použit při Monte Carlo integraci zobrazovací rovnice s využitím metody sledování cest. V první části jsou popsány použité datové struktury, následně je ukázán postup sestavení fotonové mapy a způsob vytváření separabilních 1D mip-map pro výpočet integrálu HSH. Ve třetí části je popsán způsob interpolace koeficientů hemisférických harmonických bází pomocí prostorového cache. Posléze je ukázán hlavní princip vzorkovače, způsoby použití, vlastnosti a omezení. Na závěr jsou zobrazeny dosažené výsledky a možnosti dalšího vývoje.

7.1 Akcelerační datové struktury

Tato část se zabývá popisem použitých akceleračních datových struktur v této práci. Zejména jsou ukázány detaily implementace dvou struktur, jenž jsou použity pro cachování koeficientů radiálních bázových funkcí a ukládání fotonů do mapy. Shrnutí využitých datových struktur je ukázáno na Obrázku 7.1.



Obrázek 7.1: Blokové schéma nejvyšší úrovně algoritmu a datových struktur

Octree cache koeficientů (H)SH

Z praktického hlediska nelze jednoduše nalézt takovou strukturu, jenž by pokryla celý spojitý prostor (scéna) tak, aby bylo možné v každém bodě nalézt odpovídající vyzařovací charakteristiku. Proto v této práci je použit princip interpolace ze speciálního cache tvořeného pomocí octree. Tato struktura je zapouzdřena ve třídě `SHCache`, jenž se skládá z již zmíněného octree a současně z pole koeficientů, jelikož z praktického a rovněž z výkonnostního hlediska, je vhodnější do listových uzlů octree ukládat pouze parametr pro indexování, než-li pole hodnot. Tento index slouží pro přístup do klasického pole hodnot, kde každý záznam cache má konstantní velikost, dle rozměru vektoru bázových funkcí. Spolu s indexem je do listových uzlů octree ukládána rovněž 3D pozice, normála a validační vzdálenost. Použití těchto atributů je dále popsáno v sekci o implementaci interpolace. Samotná implementace struktury octree je převzata ze zdrojových kódů PBRT-v2¹. Přístup do cache je vzhledem k paralelnímu přístupu celého algoritmu synchronizován.

Fotonová mapa pomocí kd-tree

Druhá hlavní datová struktura je fotonová mapa jenž posléze slouží k aproximaci vyzařovacích charakteristik, viz kapitola 7.3. Jelikož se opět jedná o strukturu, která musí být schopna pracovat ve 3D prostoru, tak nelze využít konvenční přístupy, ale je nutno přistoupit k dělení prostoru. V tomto případě je využita struktura *kd-tree*, zejména z důvodu vyhledávání nejbližších sousedů, které je v případě kd-stromu prováděno na základě vzdálenosti, nikoliv podle obalujícího tělesa, jako tomu je u octree. Jelikož foton přenáší směrovou informaci, je nutno kromě spektra světelného toku ukládat i příchozí směr a rovněž bod, ke kterému je daný směr vztažen. Stejně jako u výše zmíněné struktury, i zde je využito již implementovaného kd-stromu v PBRT.

7.2 Sestavení fotonové a integrální mapy

Prvním krokem před samotnou syntézou obrazu pomocí zvolené Monte Carlo metody je nutné vypočítat potřebné závislosti, které jsou využity ve vyzařovacích charakteristikách.

Fotonová mapa

Pro aproximaci lokálního rozložení světla je nutno nejdříve sestavit fotonovou mapu, která bude využita jako zdroj směrové radiance v daných bodech. Tento postup je výpočetně mnohem výhodnější, než ad-hoc vyhodnocování záře viděné v daném bodě ze všech směrů.

Metoda sběru fotonů je ilustrována na Algoritmu 2, kde celý algoritmus je prováděn paralelně na zvoleném počtu vláken. V první části se vybere světelný zdroj, který se bude vzorkovat. Tento výběr je určen zvolenou distribucí vzorkování světelných zdrojů. V PBRT lze frekvenci vzorkování zdroje určit na základě energie, plochy či náhodně. Ovšem vzhledem k tomu, že cílem této fáze je přenést co nejvíce světelného toku, tak je zvolena distribuce dle energie zdroje. Posléze je vytvořen vzorek a vystřelen foton, který je dále sledován skrz scénu. V každém bodě, kde foton narazí na povrch ve scéně, je vytvořen záznam ve fotonové mapě. Jak již bylo řečeno, tento záznam obsahuje kromě energie i směr fotonu a bod průsečíku s tělesem. Pro zefektivnění náhodné distribuce fotonů je zaveden princip

¹ zdroj: <https://github.com/mmp/pbrt-v2>

Algoritmus 2: VYTVOŘENÍ FOTONOVÉ MAPY

Data: N je celkový počet fotonů
Data: $scene$ je scéna
Data: S je strategie vzorkování svět. zdrojů
Data: RNG je generátor náhodných čísel
Výstup: F je fotonová mapa

1. `ComputeStrategy(S)`
2. **for** $n \in N$ **do in parallel**
3. `// vzorkování zdroje dle strategie distribuce`
4. $s \leftarrow \text{SampleLightSource}(scene.L, RNG)$
5. **for** $i \in (0, maxDepth)$ **do**
6. $t \leftarrow \text{Intersect}(scene, s)$
7. `// uložení fotonu do mapy`
8. `Store($F, s.power, s.dir, t.p$)`
9. $s \leftarrow \text{Sample}(t, RNG)$
10. **end**
11. **end**
12. `// vytvoření KD-tree z mračna bodů`
13. `BuildKDTree(F)`
14. **return** F

Ruské rulety, který ukončuje cesty tak, aby zabránil zbytečnému zanořování do míst, kde to již není třeba.

Současně z důvodu, že zde není potřeba zajištění unbiased přístupu, je zde využita metoda QMC (Quasi Monte Carlo) se Sobolovým generátorem kvazi-náhodných čísel. Použití QMC zde zajistí rovnoměrné rozprostření vzorků po světelném zdroji a zamezí se tím potenciálnímu směřování fotonů do shluků.

Integrální mapy

Pro efektivní výpočet rovnice (6.5) je nutné nejdříve vypočítat jednotlivé komponenty a utvořit 1D mip-mapy, ze kterých lze tuto rovnici posléze rychle řešit. Komponenta představující normalizační konstantu K_l^m je určena dle základního vztahu (3.9) a vypočítána sekvenčně do maximálního řádu bázových funkcí.

Výpočet integrálu Φ^m složky je integrál goniometrické funkce $\sin(m\phi)$, resp. $\cos(m\phi)$, na základě hodnoty m , lomený hodnotou m . Podobně jako Jarosz et al. [10] je zde tento integrál řešený pomocí rekurentních vztahů:

$$\sin(m\phi) = 2 \sin((m-1)\phi) \cos(\phi) - \sin((m-2)\phi) \quad (7.1)$$

$$\cos(m\phi) = 2 \cos((m-1)\phi) \cos(\phi) - \cos((m-2)\phi), \quad (7.2)$$

kde počáteční hodnoty rekurence rovnic (7.1), (7.2) jsou nastaveny na (7.3):

$$\begin{aligned} m = 0 : \cos(0) &= 1 & \sin(0) &= 0 \\ m = 1 : \cos(\phi) &= \cos(\phi) & \sin(\phi) &= \sin(\phi) \end{aligned} \quad (7.3)$$

Výchozí meze integrálu jsou nastaveny na $[0, 2\pi]$. Integrace probíhá rekurzivně až do takové hloubky, jak hluboko probíhá hierarchické warpování. V každé úrovni se integrační meze dělí na půl, jelikož každá úroveň již představuje 1D mip-map polární složky.

Poslední integrační složkou je $\tilde{P}_l^{|m|}$ komponenta, která opět požaduje zavedení rekurence, dle Jarosz et al. [10], kde integrál lze vypočítat na základě integrálu nižšího řádu. Nejprve jsou ovšem vyhodnoceny všechny ALP do zadaného řádu pro obě meze integrace, jelikož je nevhodné je počítat za běhu integrace, zejména kvůli znovupoužití. Následně lze pomocí rekurence určit $\hat{P}_l^{|m|} = \int_{z_-}^{z_+} P_l^m(z) dz$ jako:

$$\hat{P}_l^m = \frac{(l-2)(l-1+m)}{(l+1)(l-m)} \hat{P}_{l-2}^m - \frac{2l-1}{(l+1)(l-m)} (1-z^2) \tilde{P}_{l-1}^m(z) \Big|_{z_-}^{z_+} \quad (7.4)$$

$$\hat{P}_l^l = \frac{1}{l+1} \left(l(2l-3)(2l-1) \hat{P}_{l-2}^{l-1} + z \tilde{P}_l^l(z) \Big|_{z_-}^{z_+} \right), \quad (7.5)$$

kde opět lze definovat iniciální hodnoty rovnic (7.4), (7.5) poměrně jednoduše [10], na základě integračních mezí (7.6):

$$\hat{P}_0^0 = z_+ - z_-, \quad \hat{P}_1^0 = \frac{z_+^2 - z_-^2}{2}, \quad \hat{P}_1^1 = -\frac{1}{2} \left[z \sqrt{1-z^2} + \arcsin(z) \right] \Big|_{z_-}^{z_+} \quad (7.6)$$

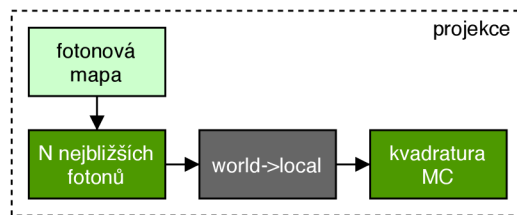
Integrace posléze probíhá stejně jako u polárních integrálů, tedy do maximální hloubky warpování. V každé úrovni mip-mapy je posléze spočítán integrál ALP pro všechny bazové funkce.

Tím je dopočítána poslední složka nutná k určení integrálu celé funkce, aproximované pomocí HSH. Tento funkční integrál je pak jednoduše určen na základě bazového vektoru s nahlížením na předpočítané mapy jako vážená suma, viz rovnice (6.6), což umožňuje velmi rychlé vzorkování. Implementace výpočtu integrálu, tak i offline zpracování jednotlivých komponent, je uložena ve třídě `SHSampler` (zdrojové kódy jsou součástí příloženého média).

7.3 Aproximace vyzařovací charakteristiky a její interpolace

K aproximaci lokálního rozložení světla ve scéně, tedy k sestavení vyzařovací charakteristiky, se využívá již dříve naplněná struktura fotonové mapy. Nad touto strukturou je proveden dotaz vyhledání N nejbližších fotonů do nastavené maximální vzdálenosti. Tato vzdálenost je určena před zahájením výpočtu, na základě velikosti scény (dle diagonály ohraničujícího boxu). Nalezené fotony jsou následně převedeny do lokálních souřadnic daného povrchu, tzn. aby bylo možné provést lokální projekci na hemisféře. Posléze přichází na řadu Monte Carlo kvadratura, viz rovnici (6.4). V implementaci je nutno zavést, oproti již zmíněné rovnici, škálování na základě maximální vzdálenosti a korigovat reálný počet nalezených fotonů, vůči požadovanému počtu pomocí $1/(\pi \cdot N_{max} \cdot t_{max})$. Projekce je provedena pouze tehdy, jsou-li nalezeny nejméně 3 fotony, maximální počet fotonů je naopak řízen parametrem (typicky 64), aby se předešlo výpočetně náročné projekci. Funkční hodnota projekce je zde určena dle síly fotonu. Zjednodušený náhled postupu lze vidět na Obrázku 7.2.

Vzhledem k povaze a směrovosti fotonů se jedná o řídkou (sparse) projekci. Z numerického a praktického hlediska je nutno tuto situaci ošetřit přidáním stejnosměrné složky. Je to z důvodu, že promítaná data jsou v podstatě δ -funkce, jenž se promítnou, jak do kladných, tak i záporných laloků bazových funkcí (podobnost lze nalézt např. u Fourierovi transformace), což má v případě řídké projekce za následek záporné integrály na intervalech. Toto



Obrázek 7.2: Blokové schéma projekce fotonů do HSH

omezení posléze způsobuje problémy během vzorkování, viz následující kapitolu. Implementačně je stejnosměrná složka přidána pouze před vzorkováním, protože ukládání takto modifikovaných vektorů do cache by mělo zkreslující vliv během interpolace ze sousedních vzorků.

Interpolace z cache

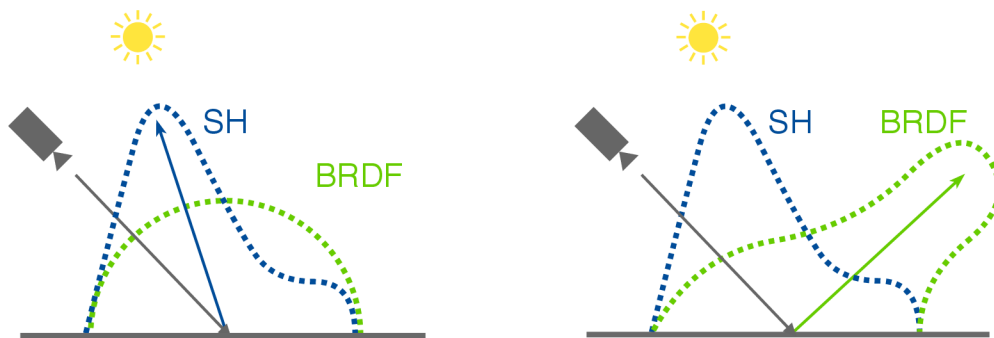
Implementace interpolace, stejně jako cache, je v již zmíněné třídě `SHCache`. Samotný proces interpolace začíná požadavkem (query) na získání všech okolních záznamů, v jejichž působnosti je interpolovaný bod, ze struktury octree. Při vyhledávání se již vypočítává heuristika z rovnice (6.8). V případě, že heuristická chyba není větší než 1 (v opačném případě by váha byla záporná), tak se rovněž progresivně určuje celkový výsledek pomocí rovnice (6.7). Interpolace je úspěšná pouze tehdy, je-li celková váha větší nebo rovna minimální požadované váze `minWeight`, kterou lze definovat v konfiguračním souboru. Parametr úhlové podobnosti, a_{max} definovaný v užití heuristice, je reprezentován proměnnou `angleDifference`, která je taktéž součástí konfigurace.

7.4 Vzorkovač vyzařovací funkce

Vzorkování z aproximovaných/interpolovaných funkcí rozložení světla je esenciálním prvkem této práce. V první řadě je nutno definovat, za jakých okolností je možné implementovaný vzorkovač použít. Na začátku této práce byly představeny zástupci jednotlivých BRDF, které lze v tomto odvětví použít. Na Obrázku 7.3 jsou ukázány dvě situace, které mohou nastat v typické fotorealistické scéně. V případě, že paprsek narazí do objektu, jehož BRDF se blíží uniformnímu rozložení (viz Obrázek 7.3a), tak lze v plném rozsahu využít představený algoritmus. Naopak, pokud objekt bude obsahovat pouze vysoce směrovou složku BRDF, např. Obrázek 7.3b, a současně světelný zdroj bude přicházet ze značně odlišeného směru, tak je důležité zajistit použití BRDF vzorkování, jelikož se typicky jedná o lesklou složku povrchu, která musí splňovat fyzikální zákony. Poslední situace, která může nastat, je, že povrch objektu obsahuje vícesložkovou BRDF. V tomto případě je na základě pravděpodobnosti vybrána jedna složka, čímž se problém redukuje zpátky na již zmíněné dvě situace. Z hlediska implementace v PBRT lze pravděpodobnost výběru definovat vztahem:

$$p_{SH} = \frac{BSDF_{diffuse+reflection}}{BSDF_{all}}, \quad (7.7)$$

kde `BSDF` označuje počet složek povrchového popisu s danou vlastností. Z definice v PBRT musí každá `BSDF` obsahovat složku `reflection` nebo `transmission`, k tomu posléze lze přidružit složky typu `diffuse`, `specular` a `glossy`. Do kategorií `diffuse` a `reflection`



(a) Je-li BRDF nezávislá na směru, tedy uniformní ve všech směrech, je vhodné vzorkovat lokální aproximovanou vyzařovací charakteristiku

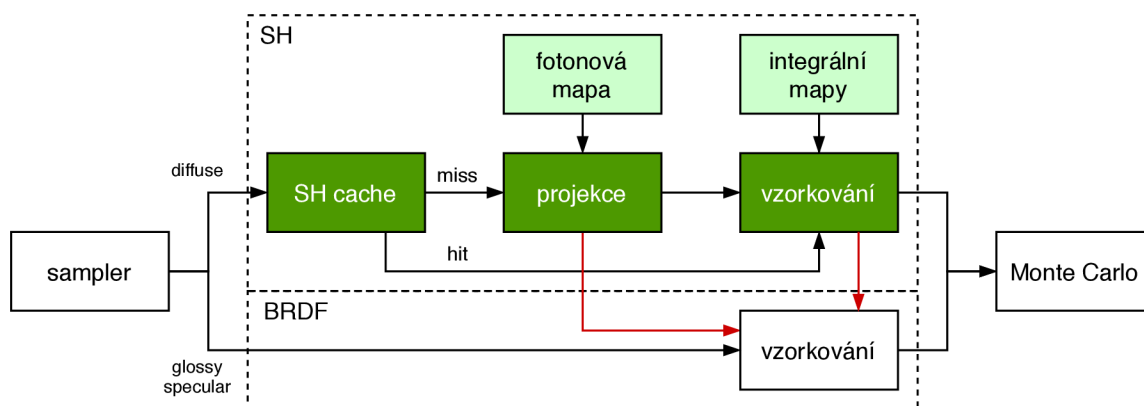
(b) V případě povrchu typu glossy/specular, či jiných podobných BRDF s vysoce směrovou charakteristikou, je nutno vzorkovat právě BRDF

Obrázek 7.3: Příklad aproximované vyzařovací charakteristiky vůči BRDF

spadají funkce typu Lambert, Oren Nayar, Fourier a různé další mikroploškové modely a jejich kombinace.

Schéma vzorkovače

Z implementačního hlediska vzorkovač jako takový algoritmicky obsahuje výše popsané prvky (projekce a interpolace), jelikož jsou to kroky, které je nutné provést před samotným podstatným vzorkováním funkce. Ilustrace vzorkovače je na Obrázku 7.4. Na blokovém schématu lze vidět, jak algoritmus pracuje. Nejprve je určeno, zda se zvolí (H)SH nebo BRDF cesta, na základě výše definované pravděpodobnosti (7.7). V případě zvolení horní cesty se nejdříve pomocí interpolace zkusí získat koeficienty HSH pomocí cache. Pokud je tento proces úspěšný, lze přejít ihned k vzorkování. Nelze-li koeficienty interpolovat z cache, např. není-li v okolí dostatečný počet funkcí, které by splňovali heuristické kritéria, je nutno nejdříve provést projekci z fotonové mapy. Červená šipka u projekce, vedoucí k BRDF, znamená neúspěšnou projekci, která je způsobena již řečeným nedostatkem fotonů, nebo příliš uniformním výsledkem, tzn. pokud je výsledek projekce velmi podobný polokouli, není třeba provádět importance sampling. V poslední části je již prováděn hierarchický warping,



Obrázek 7.4: Blokové schéma hlavní části SH vzorkovače

který rovněž obsahuje záložní cestu k BRDF, jenž je využita v případě, že funkční integrál je menší nebo roven nule.

Hierarchické warpování

Princip hierarchického warpingu byl již definován dříve, ovšem obecně tento princip uvažuje kladné integrály na intervalech. V případě, že některé integrály jsou záporné, není zaručeno správné chování. Z toho důvodu byla zavedena stejnosměrná složka, která se těsně před samotným vzorkováním vloží do vyzářovací charakteristiky. Jelikož princip podstatného vzorkování pracuje v cylindrických souřadnicích a využívá pravděpodobností pro určení dělicí přímky, není možné pracovat se zápornými čísly. Je ovšem důležité říci, že zavedením stejnosměrné složky nelze garantovat absolutní odstranění záporných oblastí. Ačkoliv algoritmus záporné oblasti nevzorkuje, principiálně jsou tyto oblasti kompletně odříznuty od vzorků, tak to není správné chování, protože se tím mohou objevit „mrtvá“ místa ve scéně, kde díky vyloučení vzorků bude porušena statistická přesnost, na které si metoda Monte Carlo zakládá, se zvyšujícím se počtem iterací.

Nevhodné je, že odhad stejnosměrné složky je velice komplikovaný a závislý od konkrétní funkce. V implementaci byl proto zaveden prvek, který uměle vzorkuje záporné oblasti tím, že jim zvýší pravděpodobnost a posléze je vzorkuje uniformně. Ačkoliv se tím značně poruší modelovaná pravděpodobnostní funkce, tak se tím zároveň vyřeší problém s bias při špatně odhadnuté stejnosměrné složce.

Z pohledu implementace je vstupem vzorkování integrál funkce, u kterého je již garance kladnosti, HSH koeficienty a dva náhodné vzorky $[u, v]$ z generátoru XOR-Shift. Algoritmus pracuje iteračně až do zvolené maximální hloubky. V každé iteraci jsou určeny hodnoty integrálů jednotlivých kvadrantů a na základě vertikální, resp. horizontální, pravděpodobnosti je určena vertikální, resp. horizontální, osa, ve které se provede warping. Hodnoty dělicích os jsou oříznuty do povolených rozsahů. Zde právě dochází k vyloučení oblastí, kde je parciální integrál záporný. Výpočet probíhá v lokálních souřadnicích a vygenerovaný směr (vzorek) je převeden do world-space až po skončení vzorkování. Implementace vzorkování je, podobně jako již řečený výpočet integrálu a separabilní mip-mapy, ve třídě `SHSampler`. Jelikož, z již popsaných numerických důvodů, je v implementaci zavedena stejnosměrná složka a uniformní vzorkování na negativních integrálech, je nutno upravit vzorkovací hustotu pravděpodobnosti, jelikož zde už neplatí vztah $p = I_{region}/I_{all}$ (jakožto podíl integrálu v němž se vzorek nachází vůči celému integrálu). Upravená pravděpodobnost je definována vztahem v rovnici (7.8):

$$p(\phi, \theta) = \frac{1}{2\pi} \prod_{i=0}^d 4 \cdot p_h \cdot p_v, \quad (7.8)$$

kde p_h, p_v odpovídá horizontální, resp. vertikální, pravděpodobnosti během warpování. Výpočet probíhá progresivně do hloubky d . Škálování pomocí čísla 4 je nutno z toho důvodu, že původní pravděpodobnost výběru kvadrantu právě odpovídá $\frac{1}{4}$ a každý horizontální, resp. vertikální, krok warpingu tuto výchozí pravděpodobnost modifikuje. Lze jednoduše určit, že v případě, kdy nastane situace uniformního vzorkování, kvůli negativnímu integrálu, bude $p_v = p_h = 0.5$, tzn. daný krok upraví pravděpodobnost jako $4 \cdot 0.5 \cdot 0.5 = 1$, což vede na to, že pravděpodobnost zůstane výchozí $\frac{1}{2\pi}$ a neporuší se tím žádné statistické pravidlo.

Po úspěšném vytvoření vzorku z vyzářovací funkce jsou zde provedeny ještě dvě kontroly, které se snaží zabránit numerickým chybám. První z nich kontroluje, zda není úhel mezi paprskem a povrchem příliš malý, jelikož by zde mohl vzniknout, vlivem nepřesnosti,

Algoritmus 3: SH PT – ZJEDNODUŠENÁ METODA RENDER INTEGRÁTORU

Data: N je celkový počet vzorků na pixel

Výstup: F je syntetizovaný obraz

```
1. SetupIntegralMipMaps()
2. SetupPhotonMap()
3. for each  $pixel \in film$  do in parallel
4.     for  $i \in (0, N)$  do
5.         // vytvoření paprsku z kamery
6.          $s \leftarrow \text{SampleCamera}(pixel)$ 
7.         for  $depth \in (0, maxDepth)$  do
8.              $t \leftarrow \text{Intersect}(scene, s)$ 
9.             ComputeBSDF()
10.            if  $bsdf$  not specular then
11.                |  $s \leftarrow \text{DirectLighting}(scene.lights)$ 
12.            // pref označuje preferenci BSDF vzorkovače
13.            if  $RNG + pref > \frac{BSDF_{diffuse+reflection}}{BSDF_{all}}$  then
14.                |  $s \leftarrow \text{SampleBSDF}()$ 
15.            else
16.                | // zahrnuje interpolaci, projekci a samotné vzorkování
17.                |  $s \leftarrow \text{SampleSH}()$ 
18.            end
19.            // akumulace záře dle nového směru a BSDF
20.             $f \leftarrow \text{UpdateRadiance}(s)$ 
21.            if RussianRoulette fail then
22.                | break
23.            end
24.        end
25.         $F \leftarrow \text{AddSample}(s)$ 
26.    end
27. end
28. return  $F$ 
```

průsečík s tím samým povrchem. Druhá kontrola se snaží zabránit možnému bílému šumu, který může vzniknout v případě velmi nízké pravděpodobnosti, která se následně využívá jako dělitel.

Zapojení do sledování cest

Poslední část implementace je samotné zapojení do integrátoru. K tomu je využito rozhraní `Integrator`, třída implementující toto abstraktní rozhraní je nazvána `ShGuidedIntegrator`. V kódu se objevuje zkratka SH, namísto HSH, jelikož sférické harmonické jsou uvažovány jako generalizované řešení a HSH jsou brány pouze jako podmnožina. V hlavní metodě `Render()` je implementován algoritmus *path tracing*, který využívá HSH vzorkovač.

Zjednodušená verze metody je ukázána v Algoritmu 3. Před samotnou procedurou je vytvořeno již popsané cache, integrální mip-mapy a fotonová mapa. Následně je film (projímací rovina) rozdělen na dlaždice velikosti 16x16 (v pseudo-kódu zjednodušeno na pixely),

které jsou paralelně zpracovávány. Každá dlaždice pak zpracovává každý pixel zvlášť. Potom je pro každý pixel vytvořeno až N vzorků a následně probíhá hlavní algoritmus pro každý vzorek. Jelikož se jedná o sledování cesty, tak první směr je generován vzorkováním kamery. Paprsek je pak sledován ve scéně. V případě průsečíku jsou určeny povrchové BSDF. Pokud povrch neobsahuje lesklou složku, lze provést přímé osvětlení ze světelných zdrojů – jsou-li viditelné. V dalším kroku je určena pravděpodobnost výběru vzorkovače, dle výše definovaného vzorce a na základě omezení různých fyzikálně zaměřených BRDF. Tato pravděpodobnost je porovnána s náhodně vygenerovaným vzorkem, který může být preferenčně posunut, čímž se zvýší pravděpodobnost užití BSDF vzorkovače – jelikož v některých situacích můžeme chtít využít i BSDF vzorkování pro uniformní povrchy. Na základě podmínky je posléze vybrán vzorkovač, v případě BSDF se jedná o klasické vzorkování jedné složky BRDF, pokud je zvolen SH vzorkovač, tak algoritmus dále probíhá dle schématu na Obrázku 7.4. Po vytvoření vzorku je určen lokální osvětlovací model, na základě příchozího a odchozího směru, BRDF koeficientu, normály a pravděpodobnosti vzorkování. Princip Ruské rulety následně určí, zda má cesta pokračovat, či skončit. Po dokončení sledování cesty je její celkový přínos uložen na příslušný pixel do filmu.

7.5 Dosažené výsledky

Pro ověření správného chování algoritmu a analýzu výsledků, bylo použito několik testů. Nejprve byly provedeny testy pro ověření konvergence scény současně s ověřením správnosti vzorkování vyzařovacích funkcí. Následně byl algoritmus porovnán s existujícími metodami. Jelikož lidské vnímání obrazu je značně subjektivní jev, je nutno pro potřeby správných měření využít správné metriky.

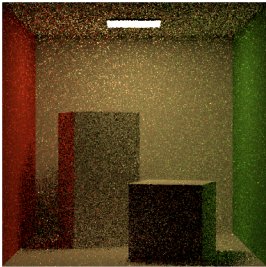
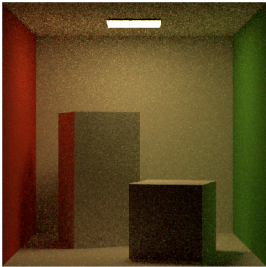
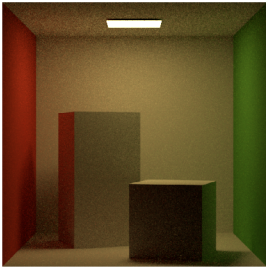
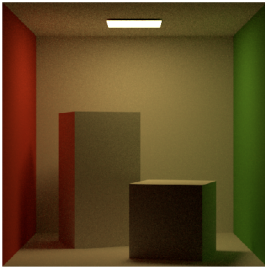
MSE První využitou metrikou je *střední kvadratická chyba*, dále jen MSE (z angl. mean square error). Při měření platí, že čím je nižší hodnota MSE, v porovnání s referenčním řešením, tím je menší chyba v syntetizovaném obraze.

PSNR Při použití MSE lze posléze využít i *špičkový poměr signálu k šumu*, neboli PSNR, z anglického peak signal-to-noise ratio. Oproti předchozí metrice zde naopak platí, že vyšší hodnoty značí kvalitnější obraz (měřeno v decibelech).

SSIM Poslední metrika, která je v této práci použita, je strukturální podobnost SSIM. Tato metoda bere v potaz skutečnost, že lidské vnímání obrazu je přizpůsobeno k extrahování strukturální informace. Rozsah hodnot je zde -1 až 1, kde 1 znamená totožné obrazy.

Ověření konvergence metody

První část měření byla zaměřena na ověření konvergence algoritmu při použití implementovaného vzorkovače. Pro zjištění, zda algoritmus nediverguje, zejména při více vzorcích na pixel, byly využity výše uvedené metriky. Naměřené hodnoty, vůči referenci, na jednoduché *cornell box* scéně a syntetizované výsledky lze vidět v Tabulce 7.1. Referenční obrázek byl vygenerován pomocí PT s 8 tisíci vzorky na pixel, jelikož v této scéně je nejlepší volbou. U takto jednoduché scény lze vidět, že míra konvergence je zde relativně vysoká, jelikož se poměrně rychle snižuje MSE a současně zvyšuje SSIM a PSNR. Tato scéna obsahuje pouze matné povrchy, modelované pomocí Lambertovské BRDF, tzn. všechny vzorky ve

1 vzorek/pixel	8 vzorků/pixel	32 vzorků/pixel	64 vzorků/pixel
			
SSIM: 0.23 MSE: 498.05 PSNR: 21.16 dB	SSIM: 0.55 MSE: 82.47 PSNR: 28.97 dB	SSIM: 0.79 MSE: 22.38 PSNR: 34.63 dB	SSIM: 0.90 MSE: 9.17 PSNR: 38.51 dB

Tabulka 7.1: Měření konvergence s vyhodnocením metrik v jednoduché scéně

1 vzorek/pixel	8 vzorků/pixel	64 vzorků/pixel
		
SSIM: 0.04 MSE: 7526.03 PSNR: 9.37 dB	SSIM: 0.15 MSE: 1091.83 PSNR: 17.75 dB	SSIM: 0.43 MSE: 187.86 PSNR: 25.39 dB

Tabulka 7.2: Ukázka konvergence scény, kde se nachází kombinované BRDF, s lesklými prvky, obsahující mikroplošky


scéně jsou generovány pomocí SH vzorkovače. Samozřejmě, až na extrémní situace, které si nutnost BRDF vzorkování vyžadují (nedostatek fotonů, příliš malý úhel s povrchem aj.), avšak jedná se pouze o méně než 1 % vzorků.

Je důležité vyzkoušet rovněž konvergenci ve scéně, kde se nachází i non-uniformní BRDF, pro ověření správného výběru vzorkovače a správné kombinace vzorkovacích technik.

V Tabulce 7.2 je ukázána jednoduchá scéna, která obsahuje smíšené BRDF, kde objekt draka je tvořen jak difúzní, tak i glossy-specular složkou. Zmíněná BRDF je rovněž tvořena mikroploškami, což ovšem ničemu nevádí, jelikož modelované funkce jsou vztažené k normále povrchu. Z hlediska poměru vzorkování SH/BRDF je zde zhruba 50 % vzorků v obou technikách. Ve výsledku lze říci, že scéna postupně konverguje, což vyplývá z naměřených hodnot. Reference k měření je obraz z datové sady v PBRT².

Poslední měření v této kategorii bylo provedeno na velmi složité scéně, která obsahuje opět smíšené BRDF (glossy a diffuse), ale i skleněné materiály, dřevo a velmi komplikovaný

²Zdroj datové sady: <https://www.pbrt.org/scenes-v3.html>

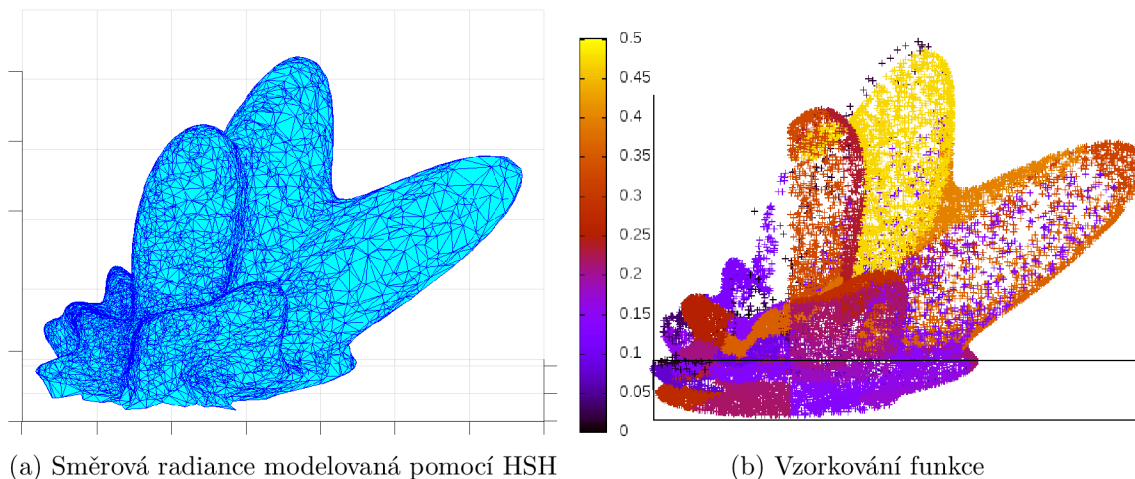
1 vzorek/pixel	8 vzorků/pixel	64 vzorků/pixel
		
SSIM: 0.07 MSE: 4512.08 PSNR: 11.59 dB	SSIM: 0.20 MSE: 881.04 PSNR: 18.68 dB	SSIM: 0.46 MSE: 177.50 PSNR: 25.64 dB

Tabulka 7.3: Měření v relativně složité scéně, obsahující četné lesklé (pohledově závislé) BRDF, s využitím kvazi-náhodných sekvencí

světelný zdroj. Naměřené hodnoty, současně s výslednými obrázky, jsou uloženy v Tabulce 7.3. Oproti předchozím scénám je zde současně, namísto XOR-shift generátoru, využit Sobolův generátor pro sekvence s nízkým nesouladem. Lze pozorovat, že vzorkovací technika dokáže pracovat i s kvazi-náhodnými čísly. Ačkoliv jsou zde patrné strukturální šумы, jsou způsobeny zejména pozicí světelného zdroje (zadní dveře), jelikož se část paprsků odráží od lesklé složky podlahy. Jak bylo již řečeno, path tracing tuto vlastnost nezvládá příliš dobře. Z hlediska konvergence lze ovšem znovu říci, že scéna postupně konverguje k referenčnímu obrazu, jenž byl opět převzat z datové sady scén PBRT. Pomalejší konvergenci lze ovšem pozorovat zejména u metriky SSIM, která je závislá právě na strukturálních změnách v obraze.

Ukázka vzorkování lokálního rozložení světla

Druhá kategorie testů zahrnovala ověření vzorkování vyzářovacích funkcí. Ačkoliv technika podstatného výběru je již ověřena autorem [10], aplikace na hemisférické harmonické funkce zatím ověřena nebyla. Jelikož se ovšem jedná pouze o podmnožinu problému, tak jde spíše o ověření správnosti implementace. Na Obrázku 7.5a je znázorněna funkce směrové radiance, modelována pomocí hemisférických básových funkcí. Tato funkce je získána přímo z aplikace, projekcí fotonů a přidáním stejnosměrné složky. V tomto konkrétním případě se podařilo téměř odstranit všechny záporné artefakty, ačkoliv se zde stále nacházejí, ovšem větší uniformní složka by značně poškodila směrovost modelované funkce. Následně je na Obrázku 7.5b ukázáno vzorkování metodou podstatného výběru. Ze stupnice pravděpodobnosti lze vidět, že nejvyšší pravděpodobnost mají vzorky na vrcholech „laloků“, což lze považovat za správné chování (ačkoliv to není z obrázku příliš patrné, je zde i nejvyšší hustota vzorků). V levé dolní části lze pozorovat, že zde vzorky mají relativně vysokou pravděpodobnost (dosahující až hodnot 0.3), to je způsobeno tím, že v této části je integrál záporný a přechází se k uniformnímu vzorkování, což právě uměle zvyšuje pravděpodobnost. Tím je zajištěno, že nenastane situace nevzorkování celých prostorových intervalů (nenastane statistický bias), ačkoliv zde může být pouze pár vzorků.



Obrázek 7.5: Ukázka modelování funkce pomocí HSH a její následné vzorkování, dle podstatného výběru, kde stupnice označuje vzorkovací pravděpodobnost

64 vzorků/pixel	čas	MSE	SSIM	PSNR
bidir. path tracing	137s	21.62	0.77	34.78 dB
path tracing + HSH	58s	124.91	0.40	27.16 dB
metropolis light transport	21s	161.41	0.36	26.05 dB
path tracing	38s	192.43	0.32	25.29 dB

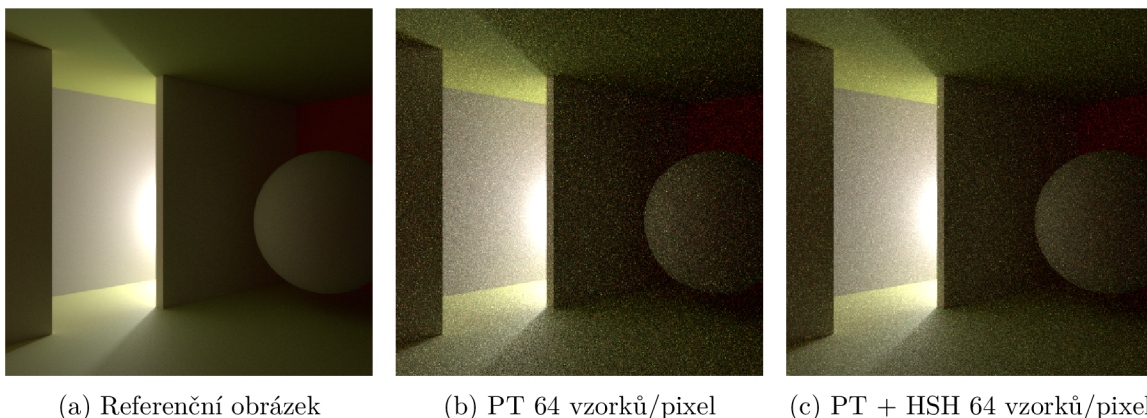
Tabulka 7.4: Porovnání implementovaného algoritmu s různými metodami, při stejném nastavení

Porovnání s existujícími metodami

Poslední testovaná kategorie měla za cíl porovnat implementovanou metodu s již existujícími a používanými metodami. Pro tyto účely byla vytvořena speciální scéna, která je svou povahou složitá na vykreslení. Jako reference byl použit Obrázek 7.6a, který byl vygenerován pomocí algoritmu BDPT s velmi vysokým rozlišením. Lze vidět, že scéna obsahuje zejména matné povrchy, ovšem podlaha místnosti je tvořena BSDF obsahující dvě BRDF – difúzní a glossy. Pozice světelného zdroje znemožňuje většinu přímého osvětlení a tedy marginální část scény je osvětlena pouze přes nepřímou část světla. Obrázky 7.6b a 7.6c posléze ukazují tuto scénu vykreslenou pomocí metody sledování cest, resp. s využitím implementovaného vzorkovače do metody cest.

Pro srovnání implementovaného algoritmu s běžně používanými metodami byly opět využity základní definované metriky. Kromě těchto metrik je ovšem potřeba sledovat i čas výpočtu. V Tabulce 7.4 jsou sepsány naměřené hodnoty pro 4 různé metody, včetně implementované, s použitím 64 vzorků na každý pixel. Implementovaná metoda je nastavena na rovnoměrné využití BRDF/HSH vzorkování, s minimální vahou interpolace 0.5. Fotonová mapa pro projekci je tvořena 2 miliony vystřelených fotonů.

V posledním řádku tabulky je změřena jednoduchá metoda sledování cest (s přímým osvětlením), kde lze vidět relativně rychlý výpočet a vysokou hodnotu PSNR. V porovnání s implementovanou metodou (druhý řádek) je zde ovšem více šumu. Naopak právě v implementovaném algoritmu je šumu méně, tím i větší PSNR, avšak doba výpočtu je



Obrázek 7.6: Ukázka scény pro porovnání algoritmu s existujícími metodami

zde ztelně pomalejší, zejména z důvodu započítání tvorby fotonové mapy. Algoritmus MLT se ukazuje jako nejrychlejší z hlediska doby výpočtu a prokazuje se i ztelně vysokou mírou redukce šumu, zejména proto, že je navrhnut na podobné scény. Z pohledu poměru šum/počet vzorků na pixel dosáhl nejlepších výsledků obousměrný path tracing, ovšem zde je doba výpočtu nejvyšší. Metrika pro strukturální integritu SSIM dosahuje ve většině případů (kromě BDPT) podobných výsledků, ovšem nutno říci, že použitá scéna je hodně tmavá a tedy nelze pozorovat příliš strukturálních detailů.

Ačkoliv v této konkrétní scéně implementovaná metoda konverguje lépe, než konvenční algoritmus sledování cest, není tomu tak vždy. Vzhledem k povaze projekce, a zejména z důvodu limitace podstatného výběru, lze dosáhnout lepších výsledků spíše ve scénách s velmi nepřímým, či velmi slabým nebo špatně dostupným zdrojem světla. Naopak v relativně otevřených scénách (např. scéna na Obrázku 7.3) algoritmus konverguje pomaleji než většina konvenčních metod. Implementace je přizpůsobivá k většině definovatelných materiálů ve scéně, je konzistentní z hlediska fyzikálně závislých povrchů, avšak zvolené radiální báze funkce nejsou robustní k směrové projekci. Současně metoda podstatného výběru není v základní variantě schopna řešit problémy špatně promítnuté funkce, ačkoliv implementace tento problém řeší, tak nalezení vhodné stejnosměrné složky není automatizováno a je nutno zvolit globální nastavení pro celou scénu.

Z pohledu doby výpočtu, je algoritmus schopen velmi rychle generovat vzorky, na základě poskytnuté vyzařovací funkce. Interpolace koeficientů se ukázala být mnohem vhodnější než Monte Carlo projekce fotonů, jelikož operace vyhledání N nejbližších fotonů je časově velmi náročná. Ačkoliv je konstrukce fotonové mapy zahrnuta do celkového času výpočtu, je možné implementaci rozšířit o samostatné generování a ukládání fotonové mapy a později ji pouze nahrát a znovu použít. Tím lze dobu výpočtu ještě snížit, jelikož v testované scéně v Tabulce 7.4 se fotonová mapa generuje zhruba 10 vteřin.

Kapitola 8

Závěr

Cílem této práce bylo nastudovat a implementovat vhodnou Monte Carlo metodu fotorealistickeho zobrazování s důrazem na rychlost a dosažitelné vlastnosti. Tento cíl byl splněn.

Prostudované materiály a náležitosti fotorealistickeho zobrazování, včetně pokročilých technik, jsou popsány v první části, následované matematickým aparátem, jenž se v této problematice využívá. V další kapitole je shrnuto vybrané existující řešení. Následně je provedena analýza a hodnocení fotorealistickeho metod. Ve zbytku práce je proveden matematický návrh a posléze implementace řešení Monte Carlo metody sledování cest, s využitím radiálních bázových funkcí, konkrétně hemisférických harmonických funkcí, pro aproximaci lokálního rozložení světla ve scéně, které je použito pro efektivnější směřování paprsků. Současně jsou ukázány dosažené výsledky a vlastnosti algoritmu.

Výsledná implementace je konzistentní a konverguje ke správnému výsledku na libovolné scéně. V případě složitěho nebo špatně dostupného osvětlení lze dosáhnout lepších výsledků než u některých konvenčních metod. Algoritmus využívá hierarchický warping, upravený pro potřeby hemisférických harmonických bází, pro velmi rychlé generování vzorků, dle podstatného výběru na povrchu scény. Aplikace dosahuje dobrého poměru výpočetního času vůči redukovanému šumu v syntetizovaném obraze.

Práci v tomto odvětví počítačové grafiky jsem prohloubil svoje znalosti a zkušenosti v metodách fotorealistickeho zobrazování, ovšem taktéž v prostředí modelování a vzorkování směrových funkcí a s tím spojenou problematikou.

Z hlediska budoucí práce se nejvíce nabízí lepší řešení projekce směrových veličin, či nalezení vhodnějších bázových funkcí, tak aby se předešlo velkým projekčním chybám při řídkém promítání velkých špiček modelované funkce. Druhá možnost je nalezení lepší metody podstatného výběru, která lépe řeší funkční artefakty způsobené projekcí. Třetím možným směrem zaměření je vymyšlení algoritmu pro progresivní učení odhadů lokálního rozložení světla tak, aby se rychleji zvyšovala konvergence při vzorkování vyzařovacích funkcí. Posledním směrem, který by mohl přispět pro celkové zlepšení chování a konvergence algoritmu, je využití techniky product importance sampling, pro efektivní vzorkování produktů BRDF a lokální estimace světla, což umožňuje kombinování odrazivosti a osvětlení do jedné funkce.

Literatura

- [1] ASHIKMIN, M., PREMOŽE, S. a SHIRLEY, P. A Microfacet-based BRDF Generator. In: SIGGRAPH. *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. 1. vyd. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., červenec 2000, s. 65–74. SIGGRAPH '00. DOI: 10.1145/344779.344814. ISBN 1-58113-208-5. Dostupné z: <http://dx.doi.org/10.1145/344779.344814>.
- [2] BLINN, J. F. Models of Light Reflection for Computer Synthesized Pictures. *SIGGRAPH Comput. Graph.* 1. vyd. New York, NY, USA: ACM. červenec 1977, sv. 11, č. 2, s. 192–198. DOI: 10.1145/965141.563893. ISSN 0097-8930. Dostupné z: <http://doi.acm.org/10.1145/965141.563893>.
- [3] COOK, R. L. a TORRANCE, K. E. A Reflectance Model for Computer Graphics. *SIGGRAPH Comput. Graph.* 1. vyd. New York, NY, USA: ACM. srpen 1981, sv. 15, č. 3, s. 307–316. DOI: 10.1145/965161.806819. ISSN 0097-8930. Dostupné z: <http://doi.acm.org/10.1145/965161.806819>.
- [4] ENTACHER, K. Bad Subsequences of Well-Known Linear Congruential Pseudorandom Number Generators. *ACM Trans. Model. Comput. Simul.* 1. vyd. New York, NY, USA: Association for Computing Machinery. leden 1998, sv. 8, č. 1, s. 61–70. DOI: 10.1145/272991.273009. ISSN 1049-3301. Dostupné z: <https://doi.org/10.1145/272991.273009>.
- [5] GAUTRON, P., KRIVANEK, J., PATTANAIK, S. a BOUATOUCH, K. A Novel Hemispherical Basis for Accurate and Efficient Rendering. In: *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*. 1. vyd. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, červen 2004, s. 321–330. EGSR'04. DOI: 10.2312/EGWR/EGSR04/321-330. ISBN 3-905673-12-6. Dostupné z: <http://dx.doi.org/10.2312/EGWR/EGSR04/321-330>.
- [6] GEORGIEV, I., KŘIVÁNEK, J., DAVIDOVIČ, T. a SLUSALLEK, P. Light Transport Simulation with Vertex Connection and Merging. *ACM Trans. Graph.* 1. vyd. New York, NY, USA: ACM. listopad 2012, sv. 31, č. 6, s. 192:1–192:10. DOI: 10.1145/2366145.2366211. ISSN 0730-0301. Dostupné z: <http://doi.acm.org/10.1145/2366145.2366211>.
- [7] GREEN, R. Spherical harmonic lighting: The gritty details. In: *Archives of the Game Developers Conference* [online]. 2003, sv. 56, s. 4 [cit. 2020-02-20]. Dostupné z: <https://www.semanticscholar.org/paper/Spherical-Harmonic-Lighting%3A-The-Gritty-Details-Green/4664a258afb97b73e84903c57b608ce66b70a2f6>.

- [8] HACHISUKA, T. a JENSEN, H. W. Stochastic Progressive Photon Mapping. *ACM Trans. Graph.* 1. vyd. New York, NY, USA: Association for Computing Machinery. prosinec 2009, sv. 28, č. 5, s. 1–8. DOI: 10.1145/1618452.1618487. ISSN 0730-0301. Dostupné z: <https://doi.org/10.1145/1618452.1618487>.
- [9] HACHISUKA, T., OGAKI, S. a JENSEN, H. W. Progressive Photon Mapping. *ACM Trans. Graph.* 1. vyd. New York, NY, USA: Association for Computing Machinery. prosinec 2008, sv. 27, č. 5. DOI: 10.1145/1409060.1409083. ISSN 0730-0301. Dostupné z: <https://doi.org/10.1145/1409060.1409083>.
- [10] JAROSZ, W., CARR, N. A. a JENSEN, H. W. Importance Sampling Spherical Harmonics. *Computer Graphics Forum (Proceedings of Eurographics)*. 1. vyd. Munich, Germany: [b.n.]. apr 2009, sv. 28, č. 2, s. 577–586. DOI: 10.1111/j.1467-8659.2009.01398.x.
- [11] JENSEN, H. W. Global Illumination Using Photon Maps. In: *Proceedings of the Eurographics Workshop on Rendering Techniques '96*. London, UK, UK: Springer-Verlag, Prosinec 1996, s. 21–30. ISBN 3-211-82883-4. Dostupné z: <http://dl.acm.org/citation.cfm?id=275458.275461>.
- [12] KAJIYA, J. T. The Rendering Equation. In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*. 1. vyd. New York, NY, USA: ACM, Srpen 1986, sv. 20, č. 4, s. 143–150. SIGGRAPH '86. DOI: 10.1145/15922.15902. ISBN 0-89791-196-2. Dostupné z: <http://doi.acm.org/10.1145/15922.15902>.
- [13] KAUTZ, J., SLOAN, P.-P. a SNYDER, J. Fast, Arbitrary BRDF Shading for Low-frequency Lighting Using Spherical Harmonics. In: *Proceedings of the 13th Eurographics Workshop on Rendering*. 1. vyd. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, červenec 2002, s. 291–296. EGRW '02. ISBN 1-58113-534-3. Dostupné z: <http://dl.acm.org/citation.cfm?id=581896.581934>.
- [14] KŘIVÁNEK, J., GAUTRON, P., PATTANAIK, S. a BOUATOUCH, K. Radiance Caching for Efficient Global Illumination Computation. In: *ACM SIGGRAPH 2008 Classes*. 1. vyd. New York, NY, USA: Association for Computing Machinery, Srpen 2008, s. 1–19. SIGGRAPH '08. DOI: 10.1145/1401132.1401228. ISBN 9781450378451. Dostupné z: <https://doi.org/10.1145/1401132.1401228>.
- [15] LAFORTUNE, E. P. a WILLEMS, Y. D. Bi-directional path tracing. In: *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*. Alvor, Portugal: [b.n.], Prosinec 1993, s. 145–153. Dostupné z: <http://graphics.cs.kuleuven.be/publications/BDPT/index.html>.
- [16] MARSAGLIA, G. Xorshift RNGs. *Journal of Statistical Software, Articles*. 1. vyd. The Florida State University. 2003, sv. 8, č. 14, s. 1–6. DOI: 10.18637/jss.v008.i14. ISSN 1548-7660. Dostupné z: <https://www.jstatsoft.org/v008/i14>.
- [17] MATSUMOTO, M. a NISHIMURA, T. Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator. *ACM Trans. Model. Comput. Simul.* 1. vyd. New York, NY, USA: ACM. leden 1998, sv. 8, č. 1, s. 3–30. DOI: 10.1145/272991.272995. ISSN 1049-3301. Dostupné z: <http://doi.acm.org/10.1145/272991.272995>.

- [18] MEAGHER, D. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*. 1. vyd. Červenec 1982, sv. 19, č. 2, s. 129–147. DOI: [https://doi.org/10.1016/0146-664X\(82\)90104-6](https://doi.org/10.1016/0146-664X(82)90104-6). ISSN 0146-664X. Dostupné z: <http://www.sciencedirect.com/science/article/pii/0146664X82901046>.
- [19] NICODEMUS, F. E. Directional Reflectance and Emissivity of an Opaque Surface. *Appl. Opt.* 1. vyd. Washington, DC, USA: OSA. Jul 1965, sv. 4, č. 7, s. 767–775. DOI: 10.1364/AO.4.000767. Dostupné z: <http://ao.osa.org/abstract.cfm?URI=ao-4-7-767>.
- [20] NIEDERREITER, H. *Random Number Generation and Quasi-Monte Carlo Methods*. 1. vyd. University City Science Center Philadelphia, PA United States: Society for Industrial and Applied Mathematics, srpen 1992. ISBN 978-0-89871-295-7. Dostupné z: <https://epubs.siam.org/doi/abs/10.1137/1.9781611970081>.
- [21] NIMIER DAVID, M., VICINI, D., ZELTNER, T. a JAKOB, W. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Trans. Graph.* 1. vyd. New York, NY, USA: Association for Computing Machinery. listopad 2019, sv. 38, č. 6. DOI: 10.1145/3355089.3356498. ISSN 0730-0301. Dostupné z: <https://doi.org/10.1145/3355089.3356498>.
- [22] OREN, M. a NAYAR, S. K. Generalization of Lambert’s Reflectance Model. In: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. 1. vyd. New York, NY, USA: ACM, červenec 1994, s. 239–246. SIGGRAPH ’94. DOI: 10.1145/192161.192213. ISBN 0-89791-667-0. Dostupné z: <http://doi.acm.org/10.1145/192161.192213>.
- [23] PHARR, M., JAKOB, W. a HUMPHREYS, G. *Physically Based Rendering: From Theory to Implementation*. 3. vyd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., srpen 2016. ISBN 0128006455, 9780128006450.
- [24] PHONG, B. T. Illumination for Computer Generated Pictures. *Commun. ACM*. 1. vyd. New York, NY, USA: ACM. červen 1975, sv. 18, č. 6, s. 311–317. DOI: 10.1145/360825.360839. ISSN 0001-0782. Dostupné z: <http://doi.acm.org/10.1145/360825.360839>.
- [25] SHIRLEY, P. a MORLEY, R. K. *Realistic Ray Tracing*. 2. vyd. Natick, MA, USA: A. K. Peters, Ltd., červenec 2003. ISBN 978-1-56881-198-7.
- [26] SHIRLEY, P., WANG, C. a ZIMMERMAN, K. Monte Carlo Techniques for Direct Lighting Calculations. *ACM Trans. Graph.* 1. vyd. New York, NY, USA: ACM. leden 1996, sv. 15, č. 1, s. 1–36. DOI: 10.1145/226150.226151. ISSN 0730-0301. Dostupné z: <http://doi.acm.org/10.1145/226150.226151>.
- [27] SLOAN, P.-P., KAUTZ, J. a SNYDER, J. Precomputed Radiance Transfer for Real-time Rendering in Dynamic, Low-frequency Lighting Environments. *ACM Trans. Graph.* 1. vyd. New York, NY, USA: ACM. červenec 2002, sv. 21, č. 3, s. 527–536. DOI: 10.1145/566654.566612. ISSN 0730-0301. Dostupné z: <http://doi.acm.org/10.1145/566654.566612>.
- [28] SPIGA, P., SORIANO, G. a SAILLARD, M. Scattering of Electromagnetic Waves From Rough Surfaces: A Boundary Integral Method for Low-Grazing Angles. *IEEE*

Transactions on Antennas and Propagation. 1. vyd. IEEE. červenec 2008, sv. 56, č. 7, s. 2043–2050. DOI: 10.1109/TAP.2008.924710. ISSN 1558-2221.

- [29] TORRANCE, K. E. a SPARROW, E. M. Theory for Off-Specular Reflection From Roughened Surfaces. *J. Opt. Soc. Am.* 1. vyd. OSA. Sep 1967, sv. 57, č. 9, s. 1105–1114. DOI: 10.1364/JOSA.57.001105. Dostupné z: <http://www.osapublishing.org/abstract.cfm?URI=josa-57-9-1105>.
- [30] VEACH, E. *Robust Monte Carlo Methods for Light Transport Simulation*. Stanford, CA, USA, 1998. Disertační práce. Stanford University. ISBN 0-591-90780-1. AAI9837162.
- [31] VEACH, E. a GUIBAS, L. J. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. 1. vyd. New York, NY, USA: ACM, Zář 1995, s. 419–428. SIGGRAPH '95. DOI: 10.1145/218380.218498. ISBN 0-89791-701-4. Dostupné z: <http://doi.acm.org/10.1145/218380.218498>.
- [32] VEACH, E. a GUIBAS, L. J. Metropolis Light Transport. In: SIGGRAPH. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. 1. vyd. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, s. 65–76. SIGGRAPH '97. DOI: 10.1145/258734.258775. ISBN 0-89791-896-7. Dostupné z: <https://doi.org/10.1145/258734.258775>.
- [33] VORBA, J., KARLÍK, O., ŠIK, M., RITSCHER, T. a KŘIVÁNEK, J. On-Line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Trans. Graph.* 1. vyd. New York, NY, USA: Association for Computing Machinery. červenec 2014, sv. 33, č. 4. DOI: 10.1145/2601097.2601203. ISSN 0730-0301. Dostupné z: <https://doi.org/10.1145/2601097.2601203>.
- [34] WALD, I. a HAVRAN, V. On Building Fast kd-trees for Ray Tracing, and on Doing that in $O(N \log N)$. *Symposium on Interactive Ray Tracing*. 1. vyd. Zář 2006, sv. 0, č. 1, s. 61–69. DOI: 10.1109/RT.2006.280216.
- [35] WARD, G. Irradiance Caching Algorithm. In: *ACM SIGGRAPH 2008 Classes*. 1. vyd. New York, NY, USA: Association for Computing Machinery, Srpen 2008, s. 1–28. SIGGRAPH '08. DOI: 10.1145/1401132.1401215. ISBN 9781450378451. Dostupné z: <https://doi.org/10.1145/1401132.1401215>.
- [36] WARD, G. J. Measuring and Modeling Anisotropic Reflection. *SIGGRAPH Comput. Graph.* 1. vyd. New York, NY, USA: ACM. červenec 1992, sv. 26, č. 2, s. 265–272. DOI: 10.1145/142920.134078. ISSN 0097-8930. Dostupné z: <http://doi.acm.org/10.1145/142920.134078>.
- [37] WHITTED, T. An Improved Illumination Model for Shaded Display. *Commun. ACM*. 1. vyd. New York, NY, USA: ACM. červen 1980, sv. 23, č. 6, s. 343–349. DOI: 10.1145/358876.358882. ISSN 0001-0782. Dostupné z: <http://doi.acm.org/10.1145/358876.358882>.

Příloha A

Obsah příloženého média

Příložené paměťové médium je členěno do tří složek a několika souborů.

- `binary/` – obsahuje již zkompilevané binární soubory včetně ukázkové scény
- `implementation/` – tento adresář obsahuje všechny zdrojové kódy aplikace, včetně PBRT závislostí
- `thesis-source/` – zdrojové soubory textové části práce

Kromě výše zmíněných složek se zde nachází i soubor `README.html`, kde jsou popsány základní nutné úkony pro samostatné zkompilevání práce a následné spuštění aplikace. Pro přehlednost je zde zahrnut i soubor `.patch`, který obsahuje všechny provedené změny v systému PBRT a umožňuje jednoduché integrování do „čisté“ verze PBRT.