

Vyšší odborná škola a Střední průmyslová
škola elektrotechnická v Plzni

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST



Tomáš Šváb

Fotonové mapy v realistickém osvětlení

Únor 2009, Plzeň

Konzultant práce: **RNDr. Josef Pelikán**, MFF UK, Praha

Studijní obor: **Technické lyceum**, 3. ročník

Soutěžní obor: 01

Poděkování

Velmi rád bych poděkoval a vyslovil uznání všem, kteří mi pomáhali při vzniku této práce. Především RNDr. Josefu Pelikánovi, vedoucímu mé práce, za trpělivé vedení a cenné rady.

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

V Plzni dne 2009

Tomáš Šváb
vlastnoruční podpis

Obsah

Úvod	11
1.1 Fotorealistická syntéza obrazu	11
1.1.1 Cíle	12
1.1.2 Omezení	12
1.2 Struktura textu	13
Globální osvětlení	14
2.1 Světlo v počítačové grafice	14
2.2 Radiometrie	14
2.3 BRDF	16
2.4 Zobrazovací rovnice	17
2.5 Potencionální rovnice	19
2.6 Dualita radiance a potenciálu	20
Monte Carlo metody	22
3.1 Monte Carlo integrace	23
3.2 Techniky na snížení rozptylu	25
3.2.1 Vzorkování po částech	26
3.2.2 Vzorkování podle důležitosti	27
3.2.3 Kombinované odhady	29
3.2.4 Řídící funkce	35
3.3 Řešení integrálních rovnic	36
3.3.1 Ruská ruleta	37
3.3.2 Odhad příští události	38
Monte Carlo metody v osvětlení	40
4.1 Přehled algoritmů	41
4.1.1 Rekurzivní sledování paprsku	41
4.1.2 Distribuované sledování paprsku	41
4.1.3 Sledování cesty	42
4.1.4 Radiační metoda	42
4.1.5 Obousměrné sledování cesty	42
4.1.6 Metropolis přenos světla	43

4.1.7	Fotonové mapy	43
4.2	Sledování cesty	44
4.2.1	Základní algoritmus	44
4.2.2	Vzorkování podle důležitosti	46
4.2.3	Odhad příští události	48
4.3	Sledování světla.....	50
4.3.1	Základní algoritmus	50
4.3.2	Vzorkování podle důležitosti	51
4.3.3	Odhad příští události	51
	<i>Fotonové mapy</i>	54
5.1	Nestranné a zkreslené Monte Carlo metody	54
5.2	Fotonová mapa	55
5.2.1	Reprezentace fotonu	55
5.2.2	Reprezentace fotonové mapy.....	55
5.3	Sledování fotonů	56
5.3.1	Vzorkování světél	56
5.3.2	Sledování fotonů.....	58
5.3.3	Ukládání fotonů	60
5.4	Odhad radiance	60
5.4.1	Odhad hustoty.....	60
5.4.2	Algoritmus	62
5.5	Vizualizace	63
5.5.1	Přímé osvětlení	63
5.5.2	Zrcadlové	65
5.5.3	Nepřímé osvětlení.....	65
5.5.4	Kaustiky.....	65
	<i>Rozšíření fotonových map</i>	66
6.1	Filtry	66
6.1.1	Kuželový filtr	66
6.1.2	Gaussův filtr	67
6.2	Adaptivní vzorkování	67
6.3	Irradiance Caching.....	68
6.3.1	Původní algoritmus	68
6.3.2	Gradienty Irradiance	70
6.3.3	Irradiance caching a fotonové mapy	71
6.4	Vzorkování podle fotonové mapy	71
6.5	Předpočítaná irradiance	72
6.6	Stínové fotony	73

6.7	Ostatní.....	74
	<i>Výsledky testů</i>	<i>76</i>
7.1	Odhad radiance	76
7.1.1	Obálky.....	77
7.1.2	Filtry	77
7.2	Přímé osvětlení.....	81
7.2.1	Metody	81
7.2.2	Výsledky	82
7.2.3	Diskuse.....	92
7.2.4	Shrnutí.....	93
7.3	Nepřímé osvětlení	93
7.3.1	Metody	93
7.3.2	Výsledky	94
7.3.3	Diskuse.....	98
7.3.4	Shrnutí.....	99
7.4	Kaustiky	99
7.4.1	Diskuse.....	99
7.5	Kompletní řešení – ukázka	99
7.6	Shrnutí	101
	<i>Závěr a budoucí práce</i>	<i>102</i>
8.1	Budoucí práce	103
8.1.1	Progresivní fotonové mapy	103
8.1.2	Opticky aktivní média.....	103
8.1.3	Poloprůhledné materiály	104
	<i>Modely odrazivosti.....</i>	<i>105</i>
	<i>Implementace</i>	<i>110</i>
B.1	Knihovna JaGrLib	110
B.2	Reprezentace dat	110
B.3	Architektura	111
B.4	Moduly	112
B.5	Uživatelské rozhraní.....	114
	<i>Obsah CD.....</i>	<i>115</i>
	<i>Bibliografie.....</i>	<i>116</i>

Seznam obrázků a grafů

Některé ilustrace byly převzaty z cizích zdrojů s laskavým svolením autorů: J. Pelikán (PE).

2.1.	Dvousměrová odrazová distribuční funkce - BRDF.....	16
2.2.	Zobrazovací rovnice.....	17
2.3.	Dvoubodová transportní geometrie.....	18
2.4.	Potencionální rovnice.....	19
3.1	Primární odhad integrálu.....	23
3.2.	Sekundární odhad integrálu.....	24
3.3.	Vzorkování po částech.....	26
3.4.	Vzorkování podle důležitosti.....	27
3.5.	Kombinované dohady.....	32
3.6.	Aritmetický průměr.....	33
3.7.	Maximální heuristika.....	33
3.8.	Vyvážená heuristika.....	34
3.9.	Mocninná heuristika.....	34
3.10.	Řídící funkce.....	35
4.1	Schéma šíření světla.....	44
4.2.	Vzorkování podle důležitosti.....	47
4.3.	Schéma šíření světla s NEE.....	48
4.4.	Schéma šíření světla (střílení).....	50
4.5.	Schéma šíření světla s NEE (střílení).....	52
5.1	Hledání nejbližších fotonů.....	61
5.2	Rozdíl mezi obálkami.....	61
5.3	Výpočet přímého osvětlení.....	63
5.4	Výpočet spekulárního osvětlení.....	64
5.5	Výpočet nepřímého osvětlení.....	64
5.6	Výpočet kaustik.....	64
6.1	Irradiance caching.....	68
6.2	Stínové fotony.....	73
7.1	Test – obálky.....	78

7.2	Test – filtry	79
7.3	Test – filtry	80
7.4	Test – přímé osvětlení (prázdna CB) – sledování paprsku	83
7.5	Test – přímé osvětlení (prázdna CB) – fotonové mapy	84
7.6	Test – přímé osvětlení (prázdna CB) – fotonové mapy	85
7.7	Test – přímé osvětlení (difuzní CB) – sledování paprsku	87
7.8	Test – přímé osvětlení (difuzní CB) – fotonové mapy	88
7.9	Test – přímé osvětlení (spekulární CB) – sledování paprsku	90
7.10	Test – přímé osvětlení (spekulární CB) – fotonové mapy	91
7.11	Test – nepřímé osvětlení (sledování cesty).....	95
7.12	Test – nepřímé osvětlení (fotonové mapy)	96
7.13	Test – nepřímé osvětlení (finální sbírání).....	97
7.14	Test – kaustiky	100
7.15	Kompletní řešení zobrazovací rovnice	101
A.1	Spekulární odraz paprsku od materiálu	107
A.2	Spekulární lom paprsku	108
A.3	Mikroploškový materiál	108
B.1	Rozhraní Skel	114

Seznam tabulek

1.	Sledování paprsku – prázdná Cornell box.	82
2.	Fotonové mapy – prázdná Cornell box.	82
3.	Sledování paprsku – difuzní Cornell box.	86
4.	Fotonové mapy – difuzní Cornell box.	86
5.	Sledování paprsku – spekulární Cornell box.	89
6.	Fotonové mapy – spekulární Cornell box.	89
7.	Nepřímé osvětlení – sledování cesty	94
8.	Nepřímé osvětlení – fotonové mapy.	94
9.	Nepřímé osvětlení – finální sbírání.	98

Seznam algoritmů

1. Rekurzivní sledování paprsku	40
2. Rekurzivní sledování paprsku	40
3. Sledování cesty.....	41
4. Sledování paprsku (sledování cesty)	42
5. Vzorkování bodového světleného zdroje.....	56
6. Vzorkování směrového světleného zdroje	57
7. Uniformní vzorkování jednotkového disku.....	57
8. Vzorkování reflektoru.....	57
9. Uniformní vzorkování kužele	58
10. Odhad hustoty	62

Název práce: *Fotonové mapy v realistickém osvětlení*

Autor: *Tomáš Šváb*

e-mail autora: *tomas.svab@gmail.com*

Škola: *Vyšší odborná a Střední průmyslová škola elektrotechnická*

Konzultant: *RNDr. Josef Pelikán*

e-mail vedoucího: *Josef.Pelikan@mff.cuni.cz*

Abstrakt:

Práce popisuje nejmodernější zobrazovací techniky pro simulaci globálního osvětlení. Tyto techniky modelují přesné chování světla ve scéně a umožňují simulovat jevy jako nepřímé osvětlení, jemné stíny, lesklé materiály či kaustiky. Cílem práce je podrobný průzkum fotonových map, které je založeny na stochastických Monte Carlo metodách, jež jsou v realistickém osvětlení často používány pro jejich robustnost.

V úvodních partiích jsou vysvětleny základy globálního osvětlení a Monte Carlo metod použitých na aproximaci integrálů, které jsou následně aplikovány na problém globálního osvětlení. Následuje ucelený popis metody fotonových map zaměřený na sledování fotonů a jejich ukládání do fotonových map, odhad radiance a vizualizaci jednotlivých částí osvětlení. Dále jsou prezentována významná rozšíření fotonových map, včetně těch nejnovějších. V dodatku jsou příklady jednotlivých modelů odrazivosti, které byly implementovány, popis architektury knihovny a přehled implementovaných modulů.

Implementace probíhala ve výukové knihovně JaGrLib, která slouží k výuce počítačové grafiky na MFF UK. Celá architektura vykreslování byla výrazně pozměněna tak, aby bylo možné snadno přidávat nové světelné zdroje, materiály i nové vykreslovací algoritmy. Implementovány byly fotonové mapy (včetně několika rozšíření), některé další algoritmy na výpočet globálního osvětlení, několik nových světelných zdrojů a řada materiálů.

Cílem praktická částí je otestovat efektivitu a kvalitu jednotlivých algoritmů založených na nestranných a zkreslených Monte Carlo metodách. Jednotlivé části výpočtu (přímé a nepřímé osvětlení a kaustiky) jsou testovány zvlášť s důrazem na kvalitu výsledného obrázku a rychlost řešení. Bylo zjištěno, že nestranné techniky jsou výborné na výpočet přímého osvětlení, zatímco zkreslené metody jsou nezbytné, pokud chceme použít i nepřímé osvětlení či kaustiky, protože jejich výpočet je daleko efektivnější a za stejný čas jsou schopny zajistit daleko kvalitnější vizuální výsledek.

Klíčová slova: *globální osvětlení, fotonové mapy, Monte Carlo metody, sledování cesty, modely odrazivosti*

Title: *Photon mapping in realistic illumination*
Author: *Tomáš Šváb*
Author's e-mail: *tomas.svab@gmail.com*
School: *The Secondary School of Electrical Engineering*
Supervisor: *RNDr. Josef Pelikán*
Supervisor's e-mail address: *Josef.Pelikan@mff.cuni.cz*
Abstract:

This paper describes one of the most modern rendering methods for global illumination simulation. These techniques perform an accurate simulation of the behavior of light and make possible to simulate such effects as indirect illumination, soft shadow, glossy reflections and caustics. The goal of this paper is to explore photon mapping, which is based on stochastic Monte Carlo methods, whose are often used in realistic illumination because of their robustness.

First is survey of the basics of global illumination and Monte Carlo methods used for approximation of integrals, which are afterwards applied to global illumination problem. After that follows serried description of photon mapping oriented on photon tracing step and saving photons into photon map, radiance estimate as well as visualization of particular components of illumination. Then they are presented significant extensions of photon mapping including newest ones. In appendix there are examples of reflection models, which were implemented, description of architecture of library and survey of implemented modules.

Implementation has been made in educational library JaGrLib, which is used for teaching of computer graphics at MFF UK. The whole architecture has been changed so that it would be able to add easily new light sources, materials and even new rendering algorithms. There was implemented photon mapping, some other algorithms for computing of global illumination, several light sources and plenty of materials.

In practical part we have focused on efficiency and quality of particular algorithms based on unbiased and biased Monte Carlo methods. Particular parts of computing (direct and indirect illumination and caustics) have been tested forcefully on quality of final image and speed of solving. It has been found out that unbiased techniques are great for computing of direct illumination, while biased methods are necessary when we want to use indirect illumination or caustics as well, because they are far more efficient, therefore they compute much more visually realistic image for the same time.

Keywords: *global illumination, photon mapping, Monte Carlo methods, path tracing, reflection models*

Kapitola 1

Úvod

Lidé chtěli odjakživa zobrazovat věci, které viděli v běžném životě a nějakým způsobem je fascinovaly, popřípadě přenést svou fantazii pomocí kresby do skutečnosti. Je dokázáno, že už v pravěku měli naši předci tendenci vytvářet různé nástěnné malby, které vyobrazovaly například lovená zvířata, a už tehdy dokázali „umělci“ využít nerovností stěn k plastickému zobrazení těl zvířat. V průběhu času se díky usilovnému studiu a experimentování začaly objevovat další techniky kresby a umělci začali čím dál více chápat osvětlení, barvu a vjem. Kupříkladu Rembrandt van Rijn, který je obecně považován za jednoho z nejlepších malířů všech dob, proslul způsobem malby, jež velmi důmyslně využívá kontrastu světla a stínu. Další významný mezník byl vynález fotografie v 19. století. Fotoaparát byl neocenitelným pomocníkem k zaznamenání existujících věcí. Až donedávna však nebylo možné fotorealisticky zobrazit věci, které neexistují nebo nejsou vnímatelné.

1.1 Fotorealistická syntéza obrazu

Za posledních pár století vyvinuli fyzikové matematické modely, jež zpracovávají interakci světla s povrchy objektů složených z nejrůznějších materiálů, jeho propagaci skrze prostředí a umožňují simulovat všechny optické jevy. S nástupem počítačů vše dostalo nový rozměr, protože zjednodušený fyzikální model může být použit pro simulaci šíření světla přímo na počítači. Tam lze geometricky popsat strukturu objektů, pomocí materiálů charakterizovat vlastnosti povrchu, vytvářet světelné zdroje s libovolnou emisní charakteristikou a pozicí, umístit kameru do libovolné pozice ve scéně a sledovat tak scénu z libovolného pohledu. Lze tak vytvořit scénu, která bude vypadat po vykreslení k nerozeznání od fotografie nějaké významné památky, místnosti, vás nebo například vašeho auta. „Umělec“ se však nemusí vázat pouze na realitu, ale může vykreslovat i scény, které zatím neexistují nebo jejich existence není reálná.

Využití je jednoduše obrovské. Lze vymodelovat místnosti s nábytkem nebo i celé domy přesně podle skutečnosti, následně je zobrazit a vylepšovat design, popřípadě dělat řízené změny podle požadavků zákazníka. Realizace produktu v reálném světě může nastat například až po finálním dokončení modelu na počítači, a tím lze ušetřit čas a peníze. Dokonce existují i projekty, které se zabývají vizualizací dávno zaniklých měst (Řím) podle dochovaných pozůstatků. Nabízí se tedy možnost, že časem nebude výuka o Římu probíhat jako dnes, ale studenti se budou moci

do Říma přímo podívat, buď za pomoci počítače, nebo dokonce pomoci speciálních přístrojů pro virtuální realitu. Už dnes je velmi rozšířené využití fotorealistickej syntézy obrazu ve filmech, kde se tímto způsobem řeší efekty ve scénách, které by jen těžko šly nasimulovat ve skutečnosti (např. by byly moc drahé). Zatím byla řeč pouze o věcech běžně viditelných. Nicméně fotorealistickej syntézu obrazu lze použít i pro vizualizaci věcí, jež zrakový systém člověka nevidí. Kupříkladu vizualizace lékařských (výsledek elektromagnetické rezonance apod.) nebo vědeckých dat, což bylo donedávna pouze doménou pro umělce a ilustrátory.

1.1.1 Cíle

V reálném světě se světlo šíří v prostoru, odráží se od materiálů a občas se stane, že dopadne z určitého směru s určitou vlnovou délkou do lidského oka, na sítnici. Oko vyšle signál do mozku, kde se výsledek zpracuje, a my si uvědomíme, co vidíme. Tento jev se nazývá vizuální zkušenost.

Nejdůležitější cíl syntézy obrazu je vytvářet výsledný obraz právě pomocí simulace šíření světla ve scéně, a to takovým způsobem, aby co nejvíce připomínal naše vizuální zkušenosti z reality. Přesněji řečeno, syntéza obrazu nevyhodnocuje přesné šíření světla ve scéně, ale vyhodnocuje přenos světelné energie ze světelných zdrojů. Barvy jednotlivých bodů na průmětně zjistíme výpočtem celkové distribuce světelné energie, která na daný bod dopadne.

1.1.2 Omezení

Při simulaci na počítači lze narazit na několik problémů. V přírodě je z pohledu makrosvěta vše spojitě, lze tedy neustále rozlišovat stále jemnější a jemnější detaily, které jsou vidět pouze zblízka nebo za použití určitého nástroje. Kupříkladu, pokud se člověk koukne na libovolný objekt z trochu větší vzdálenosti, tak ho sice uvidí, ale mnoho věcí mu zůstane skryto. Jakmile přistoupí blíže, tak si všimne detailů, které předtím nezaznamenal. Poté by se na objekt mohl podívat pod lupou, pod mikroskopem a stále by registroval nové a nové detaily. V počítačové grafice ale nejsme schopni zajistit spojitost díky omezené datové kapacitě PC. Nelze tedy pojmout všechny detaily ze skutečného světa do virtuálního. Ačkoliv rychlost PC soustavně narůstá, tak má stále velmi omezenou výši. Není tedy možné zpracovávat mnoho dat rozumnou rychlostí ani čekat příliš dlouho. Lze však použít techniky, které dokáží výpočet urychlit.

Další zařízení, které skrývá omezení a díky kterému nelze vytvořit spojitý obraz, je zobrazovací zařízení (např. LCD panel). I kdyby totiž existoval nějaký extrémně výkonný počítač s téměř neomezenou pamětí a obraz by byl spojitou funkcí, tak stejně neexistuje zobrazovací zařízení, které by bylo schopno tuto funkci zobrazit v adekvátním rozlišení a s plným počtem barev. V současnosti dokáže běžný LCD panel zobrazit rozlišení 1900x1200 (pokud se bavíme o širokoúhlých panelech), často je to však ještě méně. Barevné rozlišení se pohybuje na hodnotě 256 pro každý ze tří kanálů, což je přibližně 16-17 milionů barev. Dále je potřeba brát v úvahu pohledové podmínky, jako úroveň okolního osvětlení v místnosti, které budou určovat odezvu oka na přicházející světlo z LCD panelu.

1.2 Struktura textu

V druhé kapitole jsou popsány základní pojmy týkající se realistického osvětlení, jenž jsou nezbytné pro odvození algoritmů v pozdějších kapitolách. Úvodní popis světla následuje podkapitolou o klíčových radiometrických veličinách, které jsou následně použity na definici dvousměrové odrazové funkce a zobrazovací rovnice. V závěru je představena potencionální rovnice a dokázána dualita radiance a potenciálu, která je klíčová pro všechny modernější algoritmy.

Celá třetí kapitola je věnována metodám Monte Carlo. Kapitulu lze rozdělit na tři části. V první části jsou vysvětleny základy Monte Carlo integrace, na které navazuje druhá část, která představuje techniky na snížení rozptylu výsledného řešení. Poslední část aplikuje dosavadní poznatky na řešení integrálních rovnic, včetně ruské rulety a odhadu příští události, které jsou zásadní pro další kapitolu.

Čtvrtá část využívá Monte Carlo metod z předchozí kapitoly na problém globálního osvětlení z druhé kapitoly. V úvodu jsou stručně shrnuty téměř všechny techniky na výpočet globálního osvětlení. Poté jsou zde odvozeny dva základní přístupy podle dvou základních rovnic globálního osvětlení, na které jsou následně aplikovány důležité urychlující techniky.

Pátá kapitola pojednává o hlavním tématu práce – metodě fotonových map. Popsány jsou všechny aspekty fotonových map, od tvorby fotonové mapy, přes sledování fotonů až po vizualizaci jednotlivých druhů osvětlení. K dispozici je i několik algoritmů, které ilustrují základní kroky při vykreslování pomocí fotonových map.

Šestá část pojednává o podstatných rozšířeních fotonových map, které vznikly v průběhu let vývoje. Zabývá se především staršími a dobře prověřenými technikami. Nejnovější techniky jsou popsány pouze stručně.

Sedmá kapitola obsahuje praktické věci. V úvodu se snaží porovnat nestranné a zkreslené metody, respektive poukázat na jednotlivé výhody a nevýhody zkreslených metod. Druhá část se zabývá fotonovými mapami, především výpočty jednotlivých druhů osvětlení, případnými problémy a jejich řešeními.

V osmé kapitole je pak práce shrnuta jako celek. Navíc jsou zde představeny další směrnice pro budoucí práci.

V dodatku jsou popsány jednotlivé BRDF (dvousměrové odrazové distribuční funkce), včetně mikroploškových materiálů. Zbývá část je věnována podstatné části této práce – implementaci. Je popsána nová architektura výpočtu osvětlení v JaGrLib a také rozebrány jednotlivé implementované moduly.

Kapitola 2

Globální osvětlení

2.1 Světlo v počítačové grafice

Pro hlubší pochopení globálního osvětlení a vůbec celé počítačové grafiky je nutné porozumět světlu, které je v počítačové grafice poněkud specifické. Už v předchozí kapitole bylo zmíněno, že světlo se dá popsat více způsoby. V počítačové grafice je však světlo téměř výhradně popsáno geometricky. Ačkoliv je to z hlediska reprezentace světla ten nejjednodušší model, tak umožňuje popsat drtivou většinu jevů, které jsou podstatné pro vizuální vnímání scény, a umožňuje pracovat se světelnými paprsky jako s polopřímkami, které se řídí pomocí relativně jednoduchých geometrických pravidel. V některých případech se používá i popis pomocí částic. Modelování světla pomocí vlnové nebo elektromagnetické optiky není moc vhodné, protože je výpočetně daleko náročnější a neposkytuje pro simulaci světla nic zásadního, co se vizuálního vjemu týče.

Světlo v počítačové grafice je však zjednodušené ještě daleko více. Tady jsou jeho vlastnosti:

- 1) Světlo se šíří přímočaře.
- 2) Rychlost světla je nekonečná. Ihned po aktivaci libovolného světelného zdroje je světlo přítomno ve všech místech scény. Jeho dynamické šíření scénou je tedy zcela vynecháno.
- 3) Světlo není ovlivněno gravitací ani elektromagnetickým polem.
- 4) Ve většině případů se počítačová grafika nezabývá jevy jako difrakce (ohyb vlnění) nebo polarizace. Uvažuje o nich pouze v pokročilých technikách.

2.2 Radiometrie

Radiometrie je věda zabývající se měřením elektromagnetické energie. Protože všechny formy energie se v principu mohou přeměňovat z jednoho druhu na jiný a naopak, tak jsou základní jednotky SI pro energii a výkon jouly, respektive wattly.

Radiometrické veličiny

Radiance L je z hlediska počítačové grafiky nejdůležitější veličinou, protože z ní lze vyjádřit všechny ostatní. Udává množství emitované nebo dopadající energie z jednotkového prostorového úhlu na jednotkovou plochu kolmou na směr záření.

$$L(x, \vec{\omega}_x) = \frac{\partial^2 \Phi}{|\vec{\omega}_x \cdot \vec{N}_x| \partial A_x \partial \vec{\omega}_x} \quad (2.1)$$

Kde:

- $\partial^2 \Phi$ = přicházející, respektive vycházející světelný výkon, který prochází $\vec{\omega}_x$ a ∂A_x ,
- $|\vec{\omega}_x \cdot \vec{N}_x|$ = kosinus úhlu θ , jenž svírá normála povrchu v bodě x a vstupní směr,
- ∂A_x = diferenciální ploška na povrchu okolo bodu x ,
- $\partial \vec{\omega}_x$ = jednotkový prostorový úhel ve směru $\vec{\omega}_x$.

Zářivý výkon (flux) je druhou podstatnou veličinou. Reprezentuje množství energie vyzářené či přijaté za jednotku času.

$$\Phi = \int \int_S L(x, \vec{\omega}_x) |\vec{\omega}_x \cdot \vec{N}_x| d\vec{\omega}_x dA_x \quad (2.2)$$

Kde:

- S = všechny body a směry
- $d\vec{\omega}_x$ = jednotkový prostorový úhel ve směru $\vec{\omega}_x$,
- dA_x = diferenciální ploška na povrchu okolo bodu x .

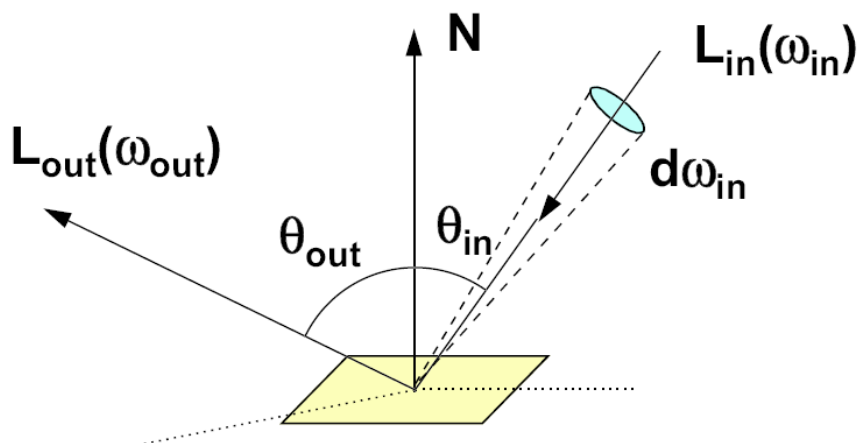
Rovnici lze přepsat jako integrál přes všechny příslušné body A a přes všechny příslušné směry Ω_x :

$$\Phi = \int_A \int_{\Omega_x} L(x, \vec{\omega}_x) W_e(x, \vec{\omega}_x) |\vec{\omega}_x \cdot \vec{N}_x| d\vec{\omega}_x dA_x \quad (2.3)$$

Kde:

- $W_e(x, \vec{\omega}_x) = 1$ pro všechny patřičné body a směry, které patří do sady S , 0 pro ostatní.

Ostatní radiometrické veličiny lze nalézt například v (1), (2), (3), (4), (5).



Obrázek 2.1: Dvousměrová odrazová distribuční funkce - BRDF. (PE)

Na závěr je potřeba zmínit ještě jedno důležité zjednodušení, které se týká světelných paprsků, které nesou energii. V počítačové grafice většinou neuvažujeme o vlivu prostředí, v kterém se světlo šíří, na paprsek, takže je energie paprsku neměnná po celé jeho dráze. Efekty jako kouř, pára, mlha či oheň jsou takzvaná opticky aktivní média (*participating media*), která tuto podmínku nespĺňují a rozptylují světelné paprsky jinde než na povrchu objektů. V této práci o opticky aktivních mediích vůbec neuvažujeme. Více informací o nich lze nalézt v (3),(4),(6).

2.3 BRDF

BRDF (obrázek 2.1) nebo-li dvousměrová odrazová distribuční funkce (*bidirectional reflectance distribution function*), slouží k vyjádření odrazových schopností materiálu v daném bodě. Definuje poměr odražené a vstupní radiance.

$$f_r(x, \vec{\omega}_i, \vec{\omega}_r) = \frac{\partial L_r(x, \vec{\omega}_r)}{L_i(x, \vec{\omega}_i) |\vec{\omega}_i \cdot \vec{N}_x| \partial \vec{\omega}_i} \quad (2.4)$$

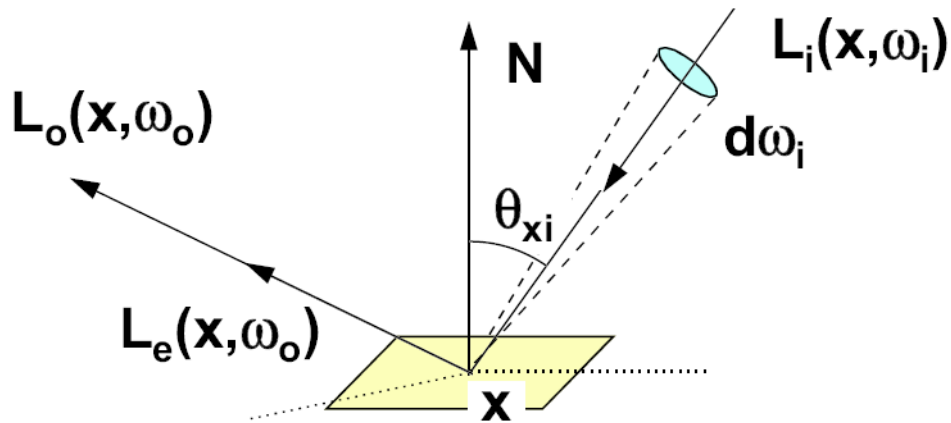
Kde:

- $\partial L_r(x, \vec{\omega}_r)$ = radiance odražená v bodě x ve směru $\vec{\omega}_r$,
- $L_i(x, \vec{\omega}_i)$ = přicházející radiance v bodě x skrze diferenciální prostorový úhel $\partial \vec{\omega}_i$ obklopující směr $\vec{\omega}_i$,
- $|\vec{\omega}_i \cdot \vec{N}_x|$ = kosinus úhlu θ , jenž svírá normála povrchu v bodě x a vstupní směr.

Jak už to v počítačové grafice bývá, existují i zde nějaká zjednodušení a omezení, která umožní vyhnout se spoustě problémům při simulaci. Zde jsou:

1. Světlo se od povrchu odráží okamžitě (vyplývá z nekonečné rychlosti světla). To znemožňuje simulování některých jevů (fosforescence).
2. Světlo nemění po odrazu svou vlnovou délku.

Vlastnosti BRDF:



Obrázek 2.2: Zobrazovací rovnice. (PE)

1. *Helmholtzův zákon* říká, že hodnota BRDF zůstává ve všech bodech stejná i při záměně směru dopadu a odrazu.

$$f_r(x, \vec{\omega}_i, \vec{\omega}_r) = f_r(x, -\vec{\omega}_r, -\vec{\omega}_i)$$

2. *Anizotropie* je obecná vlastnost materiálu. Znamená, že odraz světla závisí mimo jiné i na směru natočení povrchu kolem normálového vektoru.
3. *Z Pozitivity* vyplývá, že hodnota BRDF je vždy nezáporná.
4. *Linearita* vyjadřuje fakt, že hodnota BRDF pro daný vstupní úhel je nezávislá na hodnotách pro jiné vstupní úhly.

Příklady konkrétních BRDF jsou v dodatku A.

2.4 Zobrazovací rovnice

V počítačové grafice se téměř vždy snažíme zjistit barvu pixelů na obrazovce, popřípadě barvu pixelů na jiném zobrazovacím zařízení, což je vlastně zjišťování barvy jednotlivých bodů na průmětně. Než jsme však schopni zjistit barvu jednotlivých bodů a ty potom následně zobrazit, musíme vypočítat právě zobrazovací rovnici (rovnice 2.5), jejíž řešení udává vycházející radiance z určitého bodu pro všechny vstupní směry.

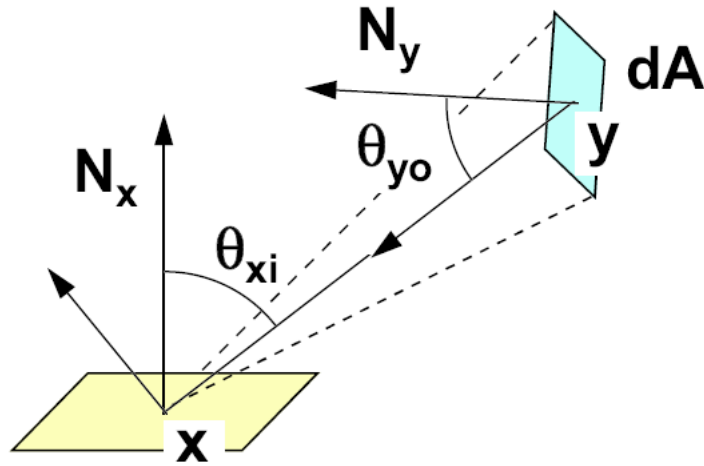
Originální zobrazovací rovnice (*rendering equation*), vyjádřena ještě pomocí intenzity, byla poprvé uvedena Kajiyou (7) v roce 1986. V dnešní době se zobrazovací rovnice vyjadřuje pomocí radiance (obrázek 2.2).

$$L_o(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + L_r(x, \vec{\omega}_r)$$

$$L_o(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \int_{\Omega} L_i(x, \vec{\omega}_i) f_r(x, \vec{\omega}_i, \vec{\omega}_r) |\vec{\omega}_i \cdot \vec{N}_x| d\vec{\omega}_i \quad (2.5)$$

Kde:

- $L_o(x, \vec{\omega}_r)$ = celková odcházející radiance v bodě x a ve směru $\vec{\omega}_r$,
- $L_e(x, \vec{\omega}_r)$ = vlastní emise v bodě x ve směru $\vec{\omega}_r$,



Obrázek 2.3: Dvoubodová transportní geometrie. (PE)

- $L_r(x, \vec{\omega}_r)$ = radiance odražená v bodě x ve směru $\vec{\omega}_r$,
- $L_i(x, \vec{\omega}_i)$ = přicházející radiance v bodě x skrze diferenciální prostorový úhel $d\vec{\omega}_i$ obklopující směr $\vec{\omega}_i$,
- $f_r(x, \vec{\omega}_i, \vec{\omega}_r)$ = BRDF v bodě x pro směry $\vec{\omega}_i$ a $\vec{\omega}_r$.

Jelikož uvažujeme o případě, kdy platí zachování energie paprsku (viz. kapitola 2.2), lze odcházející radianci vyjádřit i jako přicházející radianci pomocí funkce r (*funkce vrhání paprsku – ray casting function*), která mapuje libovolný bod x a směr $\vec{\omega}_x$ na nejbližší bod, který se průsečíkem povrchu a daného paprsku. Tento reciprokový vztah vypadá takto:

$$L_o(x, \vec{\omega}_x) = L_i(r(x, \vec{\omega}_x), \vec{\omega}_x) \quad (2.6)$$

V rovnici 2.5 může být příchozí radiance zapsána jako odchozí.

$$L_o(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \int_{\Omega} L_o(y, \vec{\omega}_i) f_r(x, \vec{\omega}_i, \vec{\omega}_r) |\vec{\omega}_i \cdot \vec{N}_x| d\vec{\omega}_i \quad (2.7)$$

Kde:

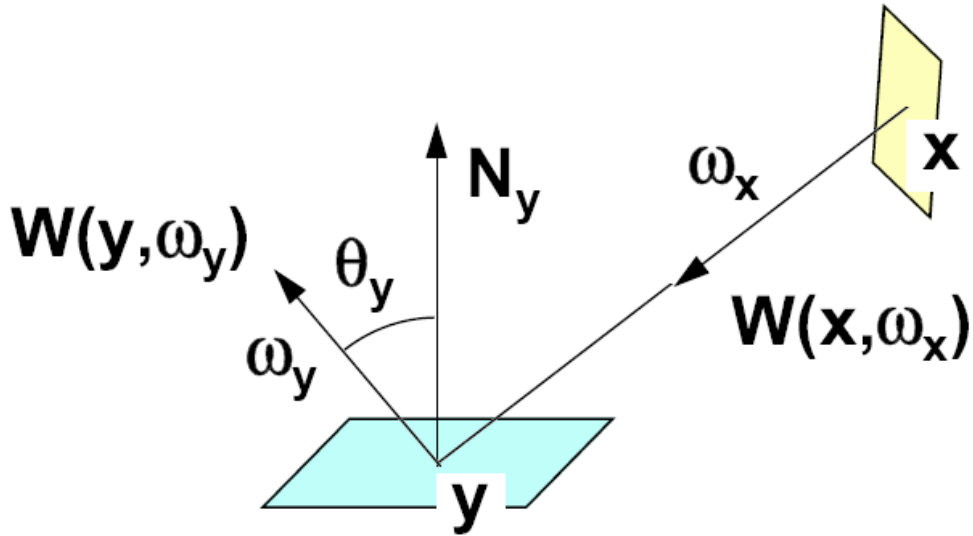
- y = bod mapovaný pomocí funkce $r(x, \vec{\omega}_i^{-1})$

Dále lze vyjádřit radiance pomocí průsečíků x a y (obrázek 2.3):

$$L(x, \vec{\omega}_x) = L_e(x, \vec{\omega}_x) + \int_{\Omega^{-1}} L(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \quad (2.8)$$

Kde:

- Ω^{-1} = integrál přes všechny směry přicházející do bodu x



Obrázek 2.4: Potencionální rovnice. (PE)

Rovnice 2.7 (a 2.8) je označována jako zobrazovací rovnice OVTIGRE (outgoing, vacuum, time/invariant, gray radiance equation) a patří mezi Fredholmovy rovnice druhého řádu a její komplikací je, že se neznámá nachází na obou dvou stranách a na té pravé ještě uvnitř integrálu.

Zobrazovací rovnice může být napsána zkráceně pomocí v operátorové formě (rovnice), kde operátor mapuje jedno funkci do druhé funkce.

$$L = L_e + TL \quad (2.9)$$

Kde integrální operátor T je definován jako:

$$(Tf)(x, \vec{\omega}_x) = \int_{\Omega_x^{-1}} f(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \quad (2.10)$$

2.5 Potencionální rovnice

Termín potenciál poprvé použil Smits (8) v roce 1992 v souvislosti s radiační metodou, kdy ho použil na určení oblastí důležitých pro výsledný obraz. V roce 1993 Pattanaik (9), (10), (11), (12), (13) představil potencionální rovnici a použil ji na řešení problému globálního osvětlení.

Potenciál je definován jako výkon vyzářený skrze svazek S jako důsledek jednotkové radiance emitované z bodu x směrem ω_x :

$$W(x, \vec{\omega}_x) = \frac{\partial^2 \Phi_o}{L_i(x, \vec{\omega}_x) \partial \vec{\omega}_x |\vec{\omega}_x \cdot \vec{N}_x| \partial A_x} \quad (2.11)$$

Kde:

- $\partial^2 \Phi_o$ = diferenciální výkon vyzářený skrze svazek S
- $L_i(x, \vec{\omega}_x)$ = vstupní radiance přicházející ze směru $\vec{\omega}_x$

Potencionální rovnice je v mnohém podobná zobrazovací rovnici 2.8, výsledný potenciál je také součtem dvou členů – vlastních emisí potenciálu a odraženého potenciálu.

$$W(x, \vec{\omega}_x) = W_e(x, \vec{\omega}_x) + W_r(x, \vec{\omega}_x)$$

$$W(x, \vec{\omega}_x) = W_e(x, \vec{\omega}_x) + \int_{\Omega} W(y, \vec{\theta}_y) f_r(y, \vec{\omega}_x, \vec{\omega}_y) |\vec{\omega}_y \cdot \vec{N}_y| d\vec{\omega}_y \quad (2.12)$$

Kde:

- $W(x, \vec{\omega}_x)$ = celková odcházející radiance v bodě x a ve směru $\vec{\omega}_x$,
- $W_e(x, \vec{\omega}_x)$ = vlastní emise v bodě x ve směru $\vec{\omega}_x$,
- $W_r(x, \vec{\omega}_x)$ = radiance odražená v bodě x ve směru $\vec{\omega}_x$,
- $f_r(y, \vec{\omega}_x, \vec{\omega}_y)$ = BRDF v bodě x pro směry $\vec{\omega}_x$ a $\vec{\omega}_y$,
- y = bod mapovaný pomocí funkce $r(x, \vec{\omega}_x)$.

Potencionální rovnice může být také zapsána v operátorovém tvaru:

$$W = W_e + T'W \quad (2.13)$$

Kde integrální operátor T' je definován jako:

$$(T'f)(x, \vec{\omega}_x) = \int_{\Omega_y} f(y, \vec{\omega}_y) f_r(y, \vec{\omega}_x, \vec{\omega}_y) |\vec{\omega}_y \cdot \vec{N}_y| d\vec{\omega}_y \quad (2.14)$$

2.6 Dualita radiance a potenciálu

Nejdříve si připomeňme rovnici na výpočet zářivého výkonu 2.3:

$$\Phi = \int_A \int_{\Omega_x} L(x, \vec{\omega}_x) W_e(x, \vec{\omega}_x) |\vec{\omega}_x \cdot \vec{N}_x| d\vec{\omega}_x dA_x$$

Vzhledem k definici potenciálu (rovnice 2.12), můžeme zářivý výkon vyjádřena i pomocí potenciálu a vlastní radiance na povrchu ploch:

$$\Phi = \int_A \int_{\Omega_x} L_e(x, \vec{\omega}_x) W(x, \vec{\omega}_x) |\vec{\omega}_x \cdot \vec{N}_x| d\vec{\omega}_x dA_x \quad (2.15)$$

Rovnici lze vyjádřit i ve zkráceném tvaru:

$$\Phi = \langle L, W_e \rangle$$

$$\Phi = \langle L_e, W \rangle$$

Za použití notace skalárního součinu:

$$\langle f, g \rangle = \int_A \int_{\Omega_x} f(x, \vec{\omega}_x) \cdot g(x, \vec{\omega}_x) |\vec{\omega}_x \cdot \vec{N}_x| d\vec{\omega}_x dA_x$$

Operátory T (rovnice 2.11) a T' (rovnice 2.15)

$$f = f_e + Tf$$

$$g = g_e + T'g$$

jsou adjungované pokud pro každé f_e a g_e platí:

$$\langle f, g_e \rangle = \langle f_e, g \rangle$$

Jelikož je tato podmínka splněna, rovnice 2.3 a 2.16 dávají stejné výsledky, proto:

$$\langle L, W_e \rangle = \langle L_e, W \rangle$$

Kapitola 3

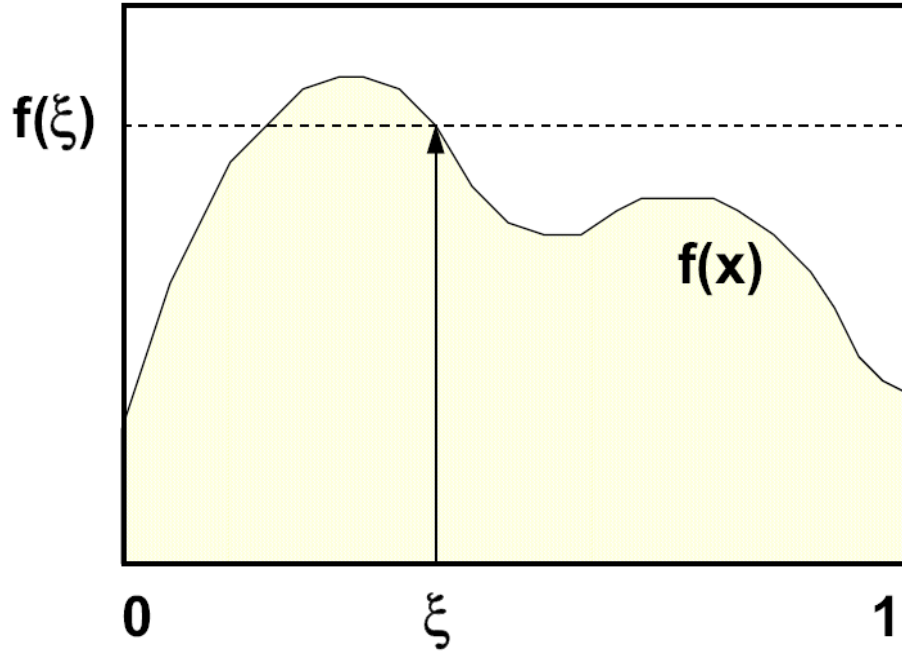
Monte Carlo metody

Stochastické algoritmy založené na generování náhodných či pseudonáhodných čísel většinou spadají do jedné ze dvou kategorií: Monte Carlo nebo Las Vegas. Zatímco Las Vegas algoritmy využívají náhodnost, ale vždy vrací správný výsledek, Monte Carlo metody vrací často špatný výsledek, ale průměrováním několika výsledků jsou schopny vrátit výsledek, který se pravděpodobně blíží přesnému řešení.

Monte Carlo integrace používá náhodné vzorkování k odhadnutí hodnot integrálu. Klíčovou výhodou Monte Carlo metod je, že jim stačí vyhodnotit integrant pouze v několika libovolných bodech, aby byly schopny odhadnout integrál dané funkce. To umožňuje Monte Carlo metodám vyhodnocovat integrály, které je obtížné či nemožné integrovat analyticky. Navíc oproti běžným numerickým metodám založeným na kvadratuře je rychlost konvergence nezávislá na dimenzi integrantu, což činí Monte Carlo metody jediné, které jsou schopny zvládat komplexnost globálního osvětlení ve své základní podobě.

Hlavní nevýhodou Monte Carlo metod je konvergence algoritmu ke správnému řešení, která je oproti ostatním numerickým metodám výrazně pomalejší. Abychom zdvojnásobili přesnost, musíme použít čtyřnásobek vzorků.

Více informací lze nalézt například v knize (14), která je excelentním úvodem do pravděpodobnostních algoritmů. Mnoho informací lze nalézt i v literatuře o klasických Monte Carlo metodách (15), (16), (17), (18), (19), (20).



Obrázek 3.1: Primární odhad integrálu funkce $f(x)$ je roven ploše obdélníka definovaného pomocí $f(\xi)$. (PE)

3.1 Monte Carlo integrace

Předpokládejme, že máme funkci $f(x)$ v intervalu $[0,1]$, kterou budeme integrovat. Odhadovaný integrál je tedy:

$$I = \int_0^1 f(x) dx \quad (3.1)$$

Primární odhad můžeme určit vygenerováním náhodného čísla ξ v intervalu $[0,1]$ (obrázek 3.1):

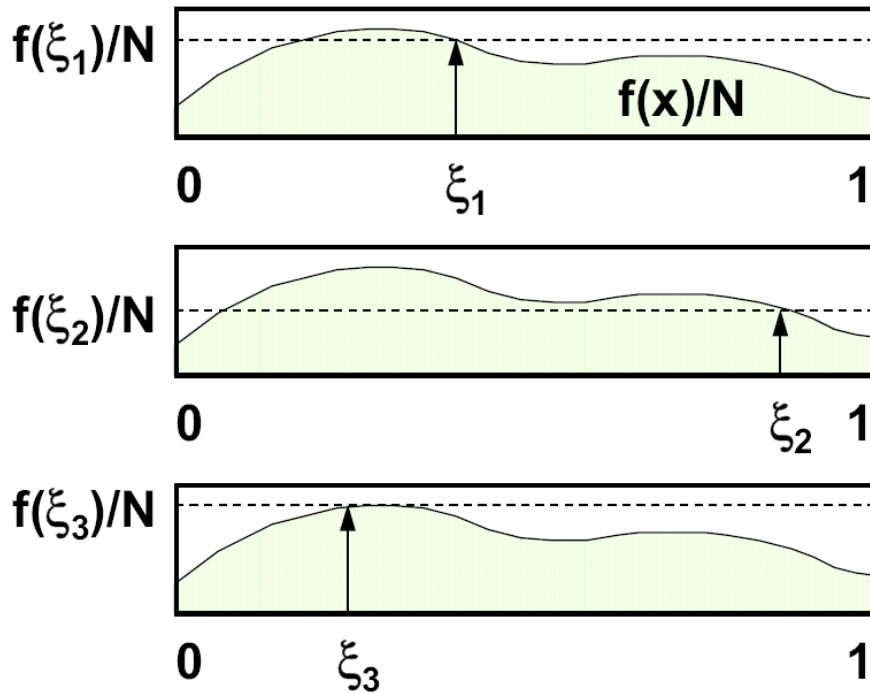
$$\langle I \rangle_{prim} = f(\xi) \quad (3.2)$$

Tento odhad je takzvaně nestranný (*unbiased*), protože jeho střední hodnota je rovna integrálu.

$$E(\langle I \rangle_{prim}) = \int_0^1 f(x) dx = I$$

Pro nestranné odhady můžeme vypočítat *rozptyl* (*variance*) nebo *standardní odchylku* (*standard error*), kterou můžeme změřit kvalitu odhadu.

$$V(\langle I \rangle_{prim}) = \sigma^2_{prim} = \int_0^1 |f(x) - I|^2 dx = \int_0^1 f(x)^2 dx - I^2 \quad (3.3)$$



Obrázek 3.2: Sekundární odhad integrálu funkce $f(x)$ je průměrem primárních odhadů. (PE)

Význam rozptylu či standardní odchylky je pouze teoretický, protože v praxi žádný z potřebných integrálů neznáme, nelze ji tedy vypočítat. Využití nacházíme především při vzájemném porovnávání různých druhů optimalizací, které zajišťují lepší výsledky.

Primární odhad je velmi nepřesný (viz. obrázek 3.1), a proto se v praxi používá *sekundární odhad* (*second estimator*), v kterém se namísto jediného nezávislého vzorku, používá N nezávislých vzorků (obrázek 3.2). Originální integrál můžeme přepsat jako součet N integrálů.

$$\begin{aligned}
 I &= \int_0^1 f(x) dx \\
 I &= \sum_{i=1}^N \int_0^1 \frac{f(x)}{N} dx \\
 I &= \sum_{i=1}^N I_i
 \end{aligned}$$

Sekundární odhad pak vypadá takto:

$$\begin{aligned}
 \langle I \rangle_{sec} &= \sum_{i=1}^N \langle I_i \rangle_{prim} \\
 \langle I \rangle_{sec} &= \frac{1}{N} \sum_{i=1}^N f(\xi_i)
 \end{aligned} \tag{3.4}$$

Pokud je splněna podmínka, že jsou vzorky navzájem nezávislé, odhad je nestranný. Standardní odchylka takového odhadu je pak:

$$\sigma^2_{sec} = \int_0^1 \dots \int_0^1 \left[\frac{1}{N} \sum_{i=1}^N f(x_i) \right]^2 dx_1 \dots dx_N - I^2$$

$$\sigma^2_{sec} = \frac{1}{N} \int_0^1 f^2(x) dx - \frac{1}{N} I^2 \quad (3.5)$$

$$\sigma^2_{sec} = \frac{\sigma^2_{prim}}{N}$$

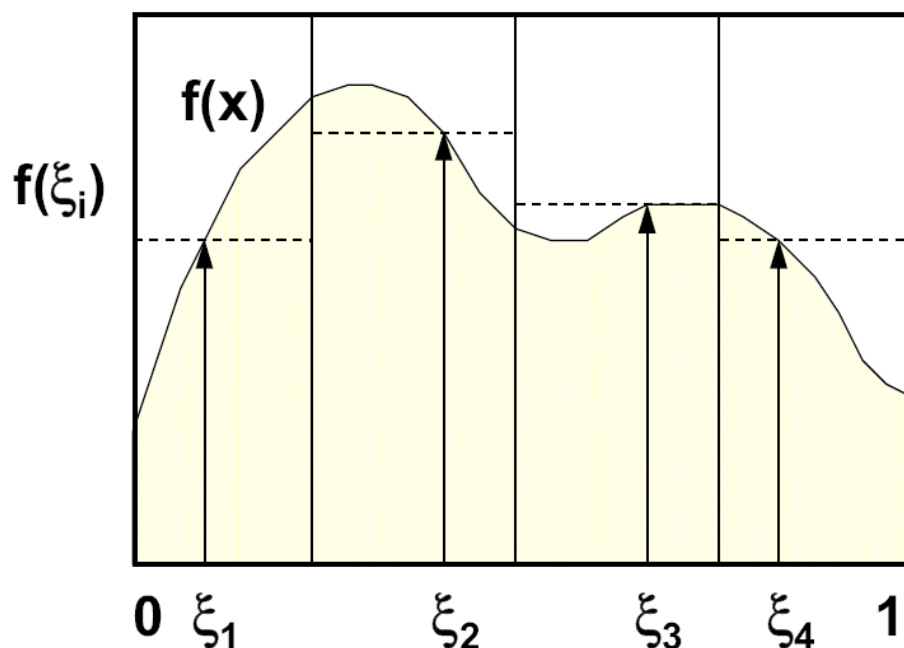
V porovnání s primárním odhadem (rovnice 3.2) je chyba \sqrt{N} menší, takže výsledek konverguje $1/\sqrt{N}$. Jak už bylo poznamenáno v úvodu této kapitoly, v porovnání s ostatními numerickými metodami je konvergence pomalá, proto bylo vyvinuto velké množství optimalizací, které se snaží urychlit konvergenci ke správnému řešení, tedy snížit počet vzorků nutný na dosažení určité přesnosti. Obecně tyto postupy nazýváme techniky na snížení rozptylu.

3.2 Techniky na snížení rozptylu

Existuje mnoho různých technik na snížení rozptylu výsledného řešení. Zde budou popsány základy těch, které se osvědčily jako nejvíce užitečné v počítačové grafice. Podrobný popis technik napsal například Veach (21), který se také zabývá především technikami využitelnými v počítačové grafice.

Všechny techniky lze v zásadě rozdělit do několika skupin:

- Do první skupiny lze zařadit techniky, které se snaží nějakým způsobem zajistit rovnoměrnější rozložení vzorků, ale zároveň takovým způsobem, aby bylo řešení stále nestranné. Vzorkování po částech je dobrým příkladem metody, která patří do této skupiny.
- Druhá skupina obsahuje techniky, které přepokládají, že některé části integrantu jsou důležitější než jiné, tudíž je potřeba tam umístit více vzorků.
- Metody v třetí skupině využívají analytické integrování funkce, která má podobný tvar jako integrant. Zástupcem technika založená na řídicí funkci.
- Další skupinou jsou techniky kombinující několik odhadů.
- Poslední skupinou jsou techniky, které během vzorkování sbírají informace a podle toho adaptivně mění hustotu vzorkování. Tyto techniky budou rozebrány v kapitole 6.2 jako významné rozšíření metody fotonových map. Více informací v (21).



Obrázek 3.3: Vzorkování po částech. (PE)

3.2.1 Vzorkování po částech

Při generování množiny nezávislých vzorků se stejnou hustotou pravděpodobnosti dochází často ke shlukování vzorků. Pokud nevíme nic o integrantu, pak je ideální distribuovat vzorky co nejuniformněji, což zajistí maximální efektivitu vzorků. Vzorkování po částech (*stratified sampling*) rozděluje interval na části, které se odhadují samostatně (obrázek 3.3), tím zajišťuje, že v každém sub-intervalu bude alespoň jeden vzorek.

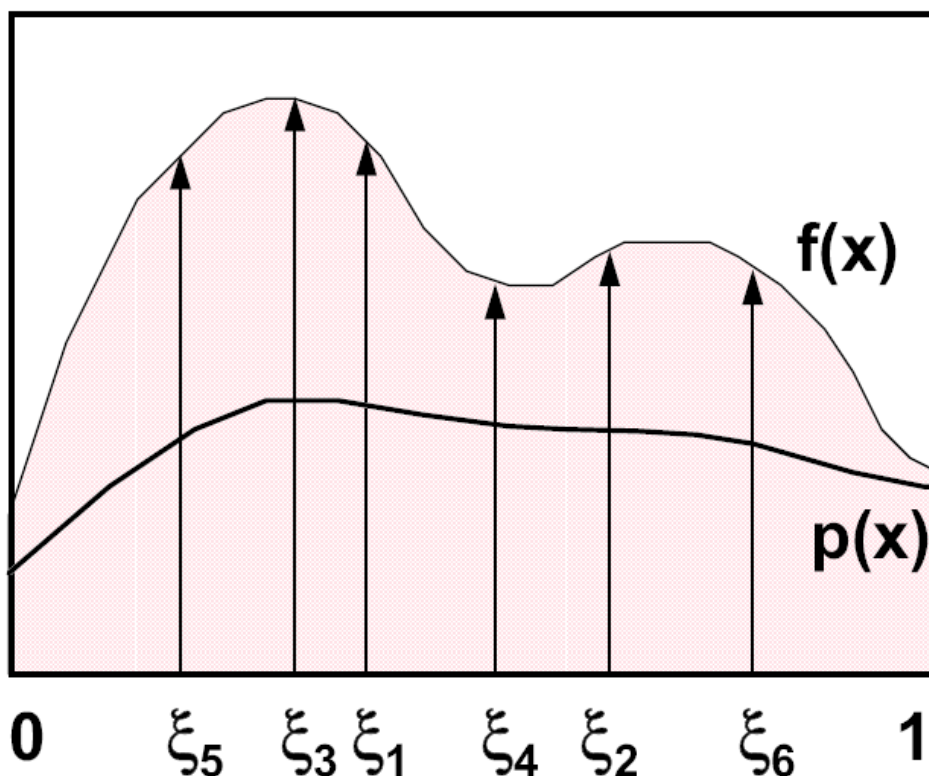
$$I = \int_0^1 f(x) dx$$

$$I = \sum_{i=1}^N \int_{A_i} f(x) dx$$

$$I = \sum_{i=1}^N I_i$$

Celkový odhad je tedy součtem jednotlivých dílčích primárních odhadů:

$$\langle I \rangle_{\text{část}} = \sum_{i=1}^N \langle I_i \rangle_{\text{prim}} \quad (3.6)$$



Obrázek 3.4: Vzorkování podle důležitosti vzorkuje v oblastech, které mají větší vliv na výsledek. (PE)

$$\langle I \rangle_{\text{část}} = \frac{1}{N} \sum_{i=1}^N f(\xi_i)$$

Celkový rozptyl je součet jednotlivých dílčích rozptylů.

$$\sigma^2_{\text{část}} = \sum_{i=1}^N \left[\int_{A_i} \left[\frac{f(x_i)}{N} \right]^2 N dx_i - I_i^2 \right] \quad (3.7)$$

$$\sigma^2_{\text{část}} = \frac{1}{N} \int_0^1 f^2(x) dx - \sum_{i=1}^N I_i^2$$

Pokud porovnáme rovnici 3.7 s rovnicí 3.5, vidíme, že součet jednotlivých členů I_i^2 nemůže být menší než člen I^2 , proto rozptyl vzorkování po částech nemůže být větší než rozptyl sekundárního odhadu, pokud je použit stejný počet vzorků.

$$\sigma^2_{\text{část}} \leq \sigma^2_{\text{sec}}$$

3.2.2 Vzorkování podle důležitosti

Vzorkování podle důležitosti, jak už ostatně název napovídá, předpokládá, že některé oblasti integrálu jsou důležitější než jiné, a tudíž je v nich potřeba mít více vzorků, zatímco v těch méně

důležitých je jich potřeba minimum. Vzorky v důležitých oblastech mají typicky větší vliv na výsledek.

Namísto uniformního vzorkování jako v předchozích případech, zde vzorkujeme podle funkce hustoty pravděpodobnosti (probability density function – PDF) $p(x)$, která má stejný tvar jako integrant.

$$I = \int_0^1 f(x) dx$$

$$I = \int_0^1 \frac{f(x)}{p(x)} p(x) dx$$

Primární odhad musí být správně upraven, jinak by už nebyl dále nestranný.

$$\langle I \rangle_{imp} = \frac{f(\xi)}{p(\xi)} \quad (3.8)$$

Sekundární odhad tedy vypadá následovně.

$$\langle I \rangle_{imp} = \frac{1}{N} \sum_{i=1}^N \frac{f(\xi_i)}{p(\xi_i)} \quad (3.9)$$

Rozptyl je teda následovný:

$$\sigma^2_{imp} = \int_0^1 \left[\frac{f(x)}{p(x)} \right]^2 p(x) dx - I^2 \quad (3.10)$$

$$\sigma^2_{imp} = \int_0^1 \frac{f^2(x)}{p(x)} dx - I^2$$

Pokud se průběh funkce $p(x)$ podobá integrované funkci $f(x)$, pak je rozptyl menší než v případě sekundárního odhadu.

Inverzní metoda

V praxi se namísto přímého výběru náhodné proměnné s hustotou pravděpodobnosti $p(x)$ bere náhodná proměnná τ s rovnoměrným rozdělením pravděpodobnosti a ta je dále transformována:

$$\xi = P^{-1}(\tau)$$

Kde $P(x)$ je kumulativní funkce hustoty pravděpodobnosti $p(x)$ a platí pro ni:

$$P(x) = \int_0^x p(t) dt \quad (3.11)$$

Primární odhad tedy v praxi vypadá takto:

$$\langle I \rangle_{imp} = \frac{f(P^{-1}(\tau))}{p(P^{-1}(\tau))} \quad (3.12)$$

Vzorkování podle důležitosti může být interpretována také jako transformace původního integrálu.

$$I = \int_0^1 f(x) dx$$

$$I = \int_0^1 f(P^{-1}(t)) \frac{dx}{dt} dt$$

$$I = \int_0^1 f(P^{-1}(t)) \frac{dP^{-1}(t)}{dt} dt$$

$$I = \int_0^1 \frac{f(P^{-1}(t))}{p(P^{-1}(t))} dt$$

3.2.3 Kombinované odhady

Někdy je integrant tak složitý, že je obtížné či nemožné použít pouze jedno PDF, které by spolehlivě aproximovalo celý integrant. V tomto případě je lepší použít několik PDF, z nichž každé aproximuje specifickou část integrantu.

Mějme N PDF $p_1(x) \dots p_N(x)$ a N náhodných proměnných $\xi_1 \dots \xi_N$. Pak můžeme součet primárních odhadů napsat jako:

$$\langle I \rangle_{combine} = \sum_{i=1}^N w_i(\xi_i) \frac{f(\xi_i)}{p(\xi_i)} \quad (3.13)$$

Kde $w_i(\xi_i)$ je váhová funkce pro i -té PDF. V obecném případě uvažujeme, že jednotlivé odhady nejsou stejně důležité, proto jsou zavedeny právě váhové funkce, které musí splňovat tuto podmínku:

$$\forall x: \sum_{i=1}^N w_i(x) = 1 \quad (3.14)$$

Jinak by nebyl odhad dále nestranný.

$$E(\langle I \rangle_{combine}) = \sum_{i=1}^N \int_0^1 \left[w_i(x_i) \frac{f(x_i)}{p_i(x_i)} \right] p_i(x_i) dx_i$$

$$E(\langle I \rangle_{combine}) = \int_0^1 \left[\sum_{i=1}^N w_i(x) \right] f(x) dx$$

$$E(\langle I \rangle_{combine}) = \int_0^1 f(x) dx$$

Rozptyl kombinovaného odhadu je roven součtu rozptylů jednotlivých odhadů:

$$\sigma^2_{combine} = \sum_{i=1}^N \left[\int_0^1 \left[w_i(x_i) \frac{f(x_i)}{p_i(x_i)} \right]^2 p_i(x_i) dx_i - \left[\int_0^1 w_i(x_i) \frac{f(x_i)}{p_i(x_i)} p_i(x_i) dx_i \right]^2 \right] \quad (3.15)$$

$$\sigma^2_{combine} = \int_0^1 \left[\sum_{i=1}^N \frac{w_i^2}{p_i(x)} \right] f(x) dx - \sum_{i=1}^N \left[\int_0^1 w_i(x) f(x) dx \right]^2$$

Existuje několik druhů vážících funkcí, zde budou uvedeny pouze některé. Je nutné mít na paměti, že všechny musí dodržovat podmínku 3.14.

- *Aritmetický průměr*

V tomto případě má každá vážící funkce stejnou hodnotu, takže i každý odhad je stejně důležitý.

$$w_i(x) = \frac{1}{N}$$

Celkový odhad je velice podobný jako sekundární odhad vzorkování podle důležitosti 3.9, až na fakt, že každý dílčí odhad má jinou PDF.

$$\langle I \rangle_{průměr} = \frac{1}{N} \sum_{i=1}^N \frac{f(\xi_i)}{p_i(\xi_i)} \quad (3.16)$$

- *Maximum*

Pro jednu vážící funkci platí:

$$w_i(x) = 1$$

Z toho vyplývá, že pro všechny ostatní:

$$w_i(x) = 0$$

Odhad vypadá následovně:

$$\langle I \rangle_{maximum} = \sum_{i=1}^N (p_i(\xi_i) > p_j(\xi_i), \forall j \neq i) ? \frac{f(\xi_i)}{p(\xi_i)} : 0 \quad (3.17)$$

Je třeba poznamenat, že v praxi se tato heuristika nepoužívá, s výjimkou případu, kdy je jeden odhad dobrý, zatímco ostatní nejsou. Tehdy by použití špatných odhadů znamenalo pouze zvýšení rozptylu výsledného odhadu, proto by bylo výhodnější ostatní odhady nepoužít.

- *Vyrovnaná heuristika*

Váhové funkce vyrovnané heuristiky:

$$w_i(x) = \frac{p_i(x)}{\sum_{j=1}^n p_j(x)}$$

Vedou k následujícímu odhadu:

$$\langle I \rangle_{balance} = \sum_{i=1}^N \frac{f(\xi_i)}{\sum_{j=1}^N p_j(\xi_i)} \quad (3.18)$$

Rozptyl vyrovnané heuristiky je následující:

$$\sigma^2_{balance} = \int_0^1 \frac{1}{\sum_{i=1}^N p_i(x)} f^2(x) dx - \sum_{i=1}^N \left[\int_0^1 \frac{p_i(x)}{\sum_{j=1}^N p_j(x)} f(x) dx \right]^2 \quad (3.19)$$

Oproti prostému kombinovanému odhadu 3.13 nabízí vyrovnaná heuristika nižší rozptyl:

$$\sigma^2_{combine} \geq \sigma^2_{balance} - \left(1 - \frac{1}{N}\right) - I^2 \quad (3.20)$$

- *Mocninná heuristika*

Mocninná heuristika je vlastně zobecněná vyrovnaná heuristika:

$$w_i(x) = \frac{p_i^\beta(x)}{\sum_{j=1}^n p_j^\beta(x)}$$

Odhad mocninné heuristiky:

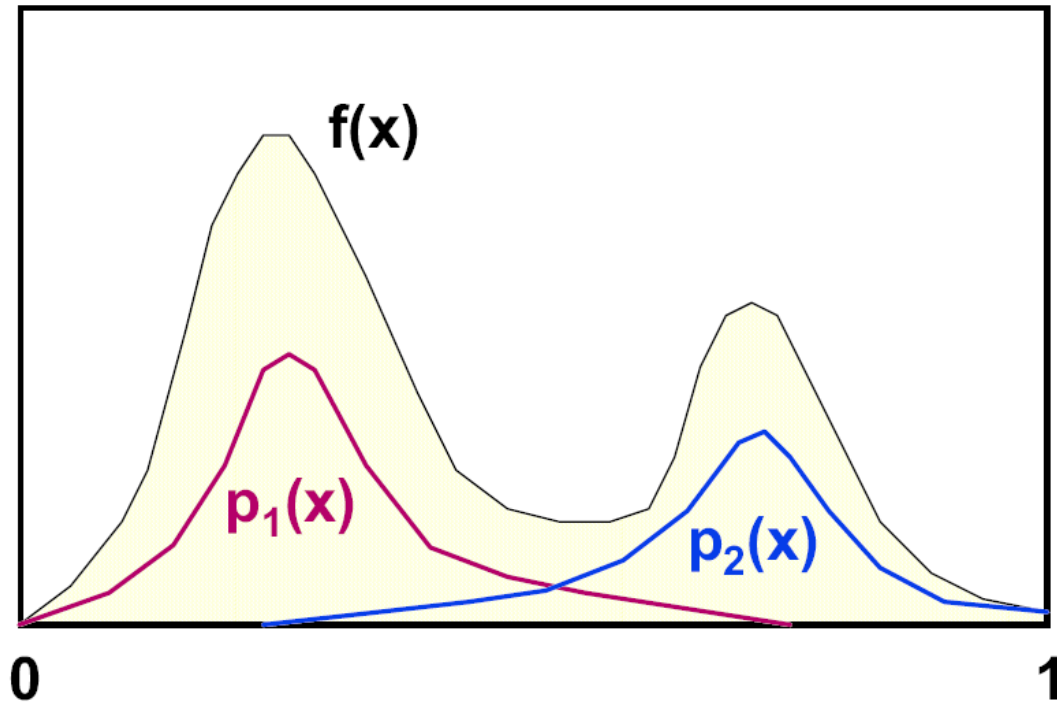
$$\langle I \rangle_{mocninná} = \sum_{i=1}^N \frac{p_i^{\beta-1}(\xi_i)}{\sum_{j=1}^N p_j^\beta(\xi_i)} f(\xi_i) \quad (3.21)$$

Za exponent β můžeme dosadit libovolné přirozené číslo. Pokud dosadíme číslo jedno, dostaneme vyrovnanou heuristiku a pokud ∞ , vznikne maximální heuristika.

Stejně jako v případě vzorkování podle důležitosti můžeme kombinované odhady interpretovat jako transformaci původního integrálu:

$$I = \int_0^1 f(x) dx$$

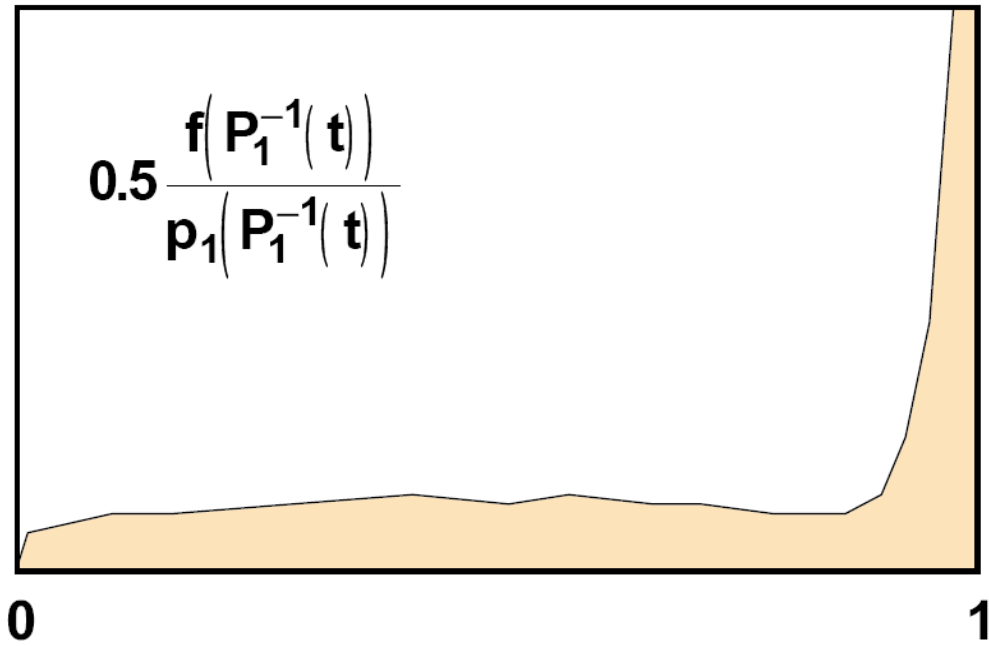
$$I = \sum_{i=1}^N \int_0^1 w_i(x) f(x) dx$$



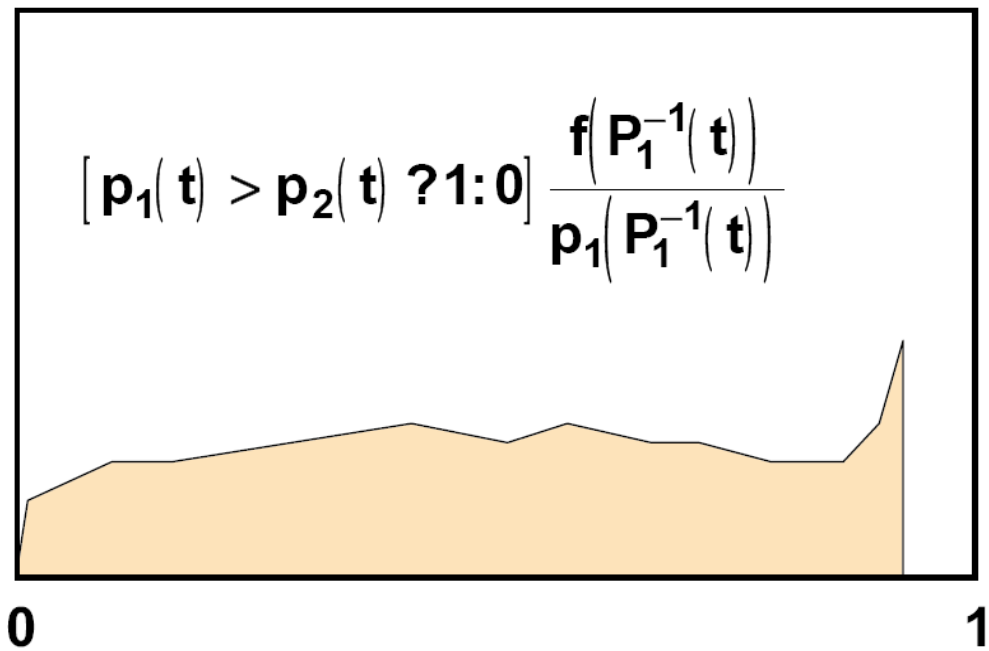
Obrázek 3.5: Funkce $f(x)$ je příliš složitá, aby šla efektivně aproximovat pomocí jedné PDF, proto jsou k její aproximaci použity dvě. Každý PDF bude použita pro patřičný odhad. Všechny odhady budou následně zkombinovány pomocí jedné z heuristik. (PE)

Po přidání vzorkování podle důležitosti další transformací se rovnice změní:

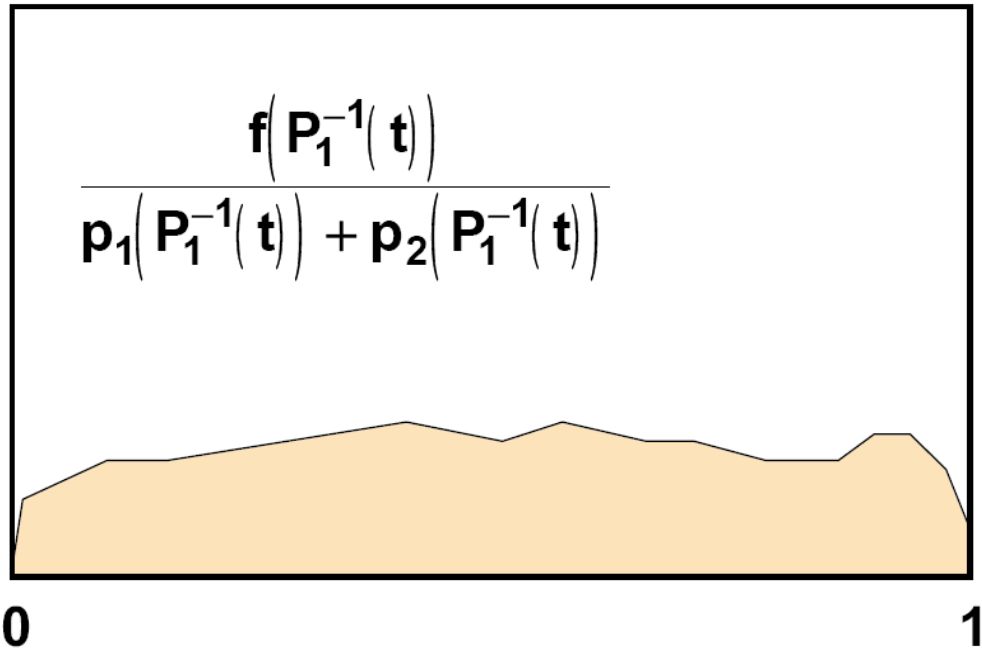
$$I = \sum_{i=1}^N \frac{w_i(P_i^{-1}(t))}{p_i(P_i^{-1}(t))} f(P_i^{-1}(t)) dt$$



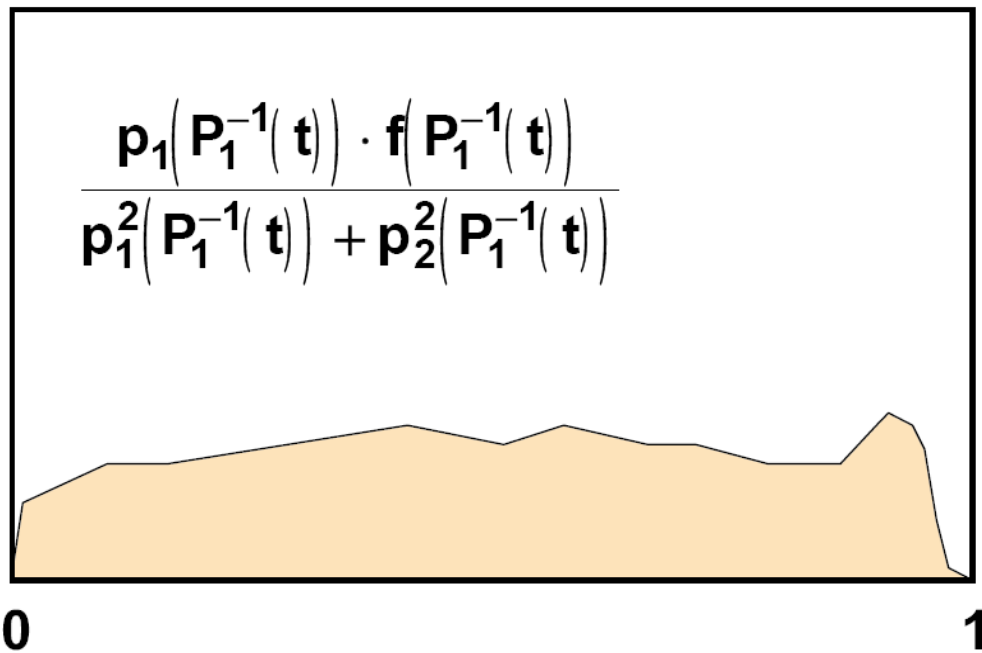
Obrázek 3.6: Kombinování odhadů s použitím aritmetického průměru. (PE)



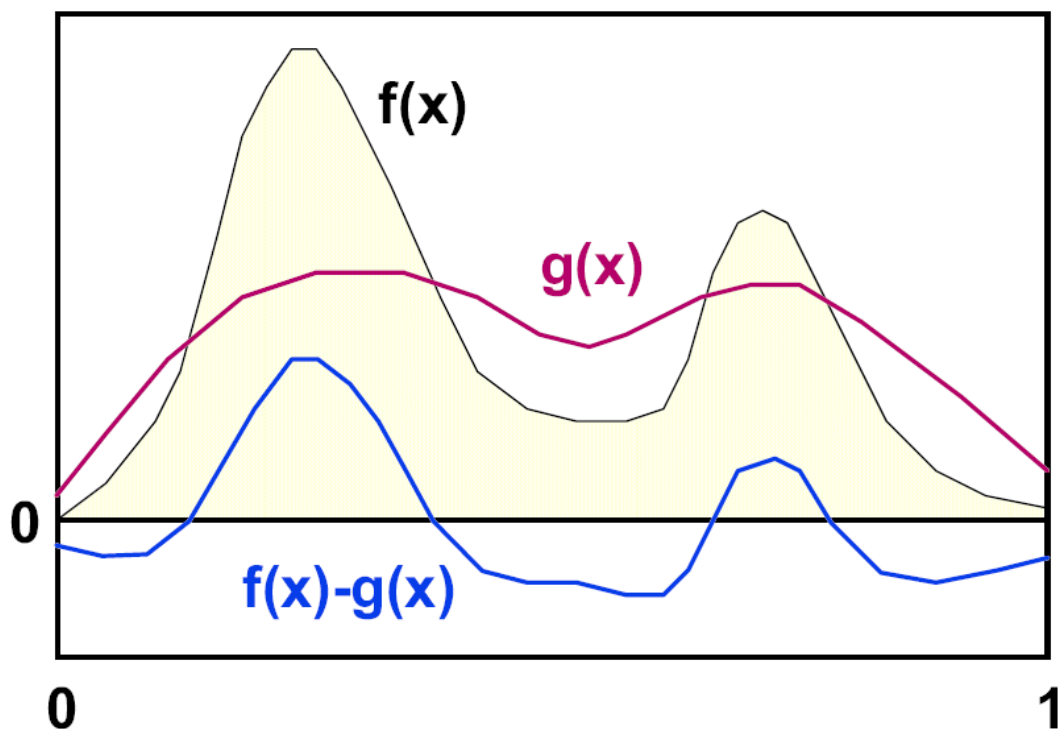
Obrázek 3.7: Kombinování odhadů s použitím maximální heuristiky. (PE)



Obrázek 3.8: Vyrovnaná heuristika. (PE)



Obrázek 3.9: Mocninná heuristika. (PE)



Obrázek 3.10: Funkce $f(x)$ je funkce, kterou integrujeme, zatímco funkce $g(x)$ je řídicí funkce, kterou lze analyticky integrovat. (PE)

3.2.4 Řídicí funkce

Řídicí funkce $g(x)$ je taková funkce, která aproximuje integrant a zároveň ji dokážeme analyticky integrovat. Původní integrál transformujeme:

$$\begin{aligned}
 I &= \int_0^1 f(x) dx \\
 I &= \int_0^1 [f(x) - g(x)] dx + \int_0^1 g(x) dx \\
 I &= \int_0^1 [f(x) - g(x)] dx + J \\
 I &= \int_0^1 [f(x) - g(x) + J] dx
 \end{aligned}$$

Pokud je proměnná ξ vzorkována uniformě, primární odhad je:

$$\langle I \rangle = f(\xi) - g(\xi) + J \quad (3.22)$$

Rozptyl je:

$$\sigma^2_{\text{řídící}} = \int_0^1 [f(x) - g(x) + J]^2 dx - I^2 \quad (3.23)$$

3.3 Řešení integrálních rovnic

Vezměme integrální rovnici, která je stejně jako zobrazovací rovnice Fredholmovou rovnicí druhého řádu.

$$f(x) = g(x) + \int_0^1 K(x, y) f(y) dy \quad (3.24)$$

- $f(x)$ = neznámá funkce,
- $g(x)$ = známá funkce,
- $K(x, y)$ = jádro integrálního operátoru.

Ve většině praktických případů nelze řešit rovnici 3.24 analyticky, proto přicházejí na řadu numerické metody, v našem případě Monte Carlo metody, které jsou schopny vypočítat neznámou funkci $f(x)$ pro dané x . Avšak pro výpočet $f(x)$ je nutné napřed vypočítat $f(y)$, pro $f(y)$ zase $f(z)$ a tak dále. Výpočetní proces je tedy rekurzivní. Primární odhad vypadá takto:

$$\langle f(x) \rangle_{rekurzivní} = g(x) + \frac{K(x, \xi_1)}{p_1(\xi_1)} \langle f(\xi_1) \rangle_{rekurzivní} \quad (3.25)$$

$$\langle f(x) \rangle_{rekurzivní} = g(x) + \frac{K(x, \xi_1)}{p_1(\xi_1)} g(\xi_1) + \frac{K(x, \xi_1) K(\xi_1, \xi_2)}{p_1(\xi_1) p_2(\xi_2)} g(\xi_2) + \dots$$

$$\langle f(x) \rangle_{rekurzivní} = g(x) + \frac{K(x, \xi_1)}{p_1(\xi_1)} \left[g(\xi_1) + \frac{K(\xi_1, \xi_2)}{p_2(\xi_2)} \langle f(\xi_2) \rangle_{rekurzivní} \right]$$

$$\langle f(x) \rangle_{rekurzivní} = \sum_{i=0}^{\infty} \left[\prod_{j=1}^i \frac{K(\xi_{j-1}, \xi_j)}{p_j(\xi_j)} \right] g(\xi_i) \quad (3.26)$$

Kde:

- $\xi_1 = x$,
- $\xi_1 \dots \xi_N$ = Markovův řetězec nebo-li náhodná procházka (více v (22))

Stejně jako zobrazovací rovnice (kapitola 2.4) a potencionální rovnice (kapitola 2.5), i rovnice 3.25 může být vyjádřena v operátorovém tvaru:

$$f = g + Tf$$

Nekonečná suma v rovnici 3.26 se nazývá Neumannova řada, kterou lze přepsat do operátorového tvaru:

$$f = \sum_{i=0}^{\infty} T^i g$$

3.3.1 Ruská ruleta

Rovnice 3.26 je v praxi stále nepoužitelná, protože iteruje do nekonečna. Je tedy potřeba nějakým způsobem tuto iteraci ukončit. Nejjednodušším řešením by bylo jednoduše namísto nekonečna dosadit libovolné přirozené číslo, podle toho, jak přesné řešení potřebujeme. Avšak odhad nekonečné řady částečným součtem s pevnou délkou zavádí do řešení systematickou chybu. Výsledek už by nebyl nestranný. Právě proto se velice často používá ruská ruleta, která zajišťuje náhodné ukončení, a zároveň zaručuje nestrannost odhadu. Další nespornou výhodou ruské rulety je fakt, že v případě pevné délky řady, budou jednotlivé cesty vždy stejné dlouho, ačkoliv některé z nich budou důležitější než jiné. V ruské ruletě mohou mít důležitější cesty větší pravděpodobnost, že nebudou ukončeny.

Pomocí jednoduché transformace rovnice 3.1 dostaneme:

$$\begin{aligned} I &= \int_0^1 f(x) dx \\ I &= \int_0^P \frac{1}{P} f\left(\frac{x'}{P}\right) dx' \\ I &= \int_0^1 \frac{1}{P} f\left(\frac{x'}{P}\right) u\left(\frac{x'}{P}\right) dx' \end{aligned}$$

Kde:

- $P =$ libovolné číslo v intervalu $(0, 1)$,
- $u(x) =$ funkce, která je rovna 1, pokud $x \leq 1$ a rovna 0, pokud $x > 1$.

Nestranný primární odhad vypadá tedy takto:

$$\langle I \rangle_{ruleta} = \begin{cases} \frac{1}{P} f\left(\frac{\xi}{P}\right) & \text{pokud } \xi \leq P \\ 0 & \text{jinak} \end{cases} \quad (3.27)$$

Kde

- $P =$ pravděpodobnost, že bude odhad nenulový.

S použitím ruské rulety se také nevyhnutelně zvyšuje rozptyl výsledného odhadu:

$$\sigma^2_{ruleta} = \int_0^1 \left[\frac{1}{P} f\left(\frac{\xi}{P}\right) u\left(\frac{\xi}{P}\right) \right]^2 d\xi - I^2 \quad (3.28)$$

$$\sigma^2_{ruleta} = \int_0^P \frac{1}{P^2} f^2\left(\frac{\xi}{P}\right) d\xi - I^2$$

$$\sigma^2_{ruleta} = \frac{1}{P} \int_0^1 f^2(\xi) d\xi - I^2$$

$$\sigma^2_{ruleta} > \sigma^2_{prim}$$

Rovnice 3.26 s ruskou ruletou vypadá následovně:

$$\langle f(x) \rangle_{rekurzivní} = \sum_{i=0}^{\infty} \left[\prod_{j=1}^i \frac{K(\xi_{j-1}, \xi_j)}{p_j(\xi_j)} \right] g(\xi_i) \quad (3.26)$$

$$\langle f(x) \rangle_{ruleta, rekurzivní} = \sum_{i=0}^k \left[\prod_{j=1}^i \frac{K(\xi_{j-1}, \xi_j)}{P_j p_j(\xi_j)} \right] g(\xi_i) \quad (3.29)$$

Ve fyzikálních aplikacích jádro $K(x, y)$ často splňuje podmínku:

$$\int_0^1 K(x, y) dy < 1$$

Tehdy můžeme jádro použít ke konstrukci takzvaného sub-kritického rozdělení pravděpodobnosti. Kumulativní funkce hustoty pravděpodobnosti je konstruována za pomoci jádra K :

$$P_i = \int_0^1 K(\xi_{i-1}, y) dy$$

To pro PDF znamená:

$$p_1(x) = \frac{K(\xi_{i-1}, y)}{P_i}$$

Pokud použijeme dané PDF v rovnici 3.29, členy K a P se vykrátí a zbude pouze:

$$\langle f(x) \rangle = \sum_{i=1}^k g(\xi_i) \quad (3.30)$$

3.3.2 Odhad příští události

Celkový odhad (rovnice 3.29 a 3.30) často mívá velký rozptyl, protože většina sčítanců je rovna nule. Většinou tedy dostane odhad jednou za čas opravdu vysoký příspěvek. V těchto situacích může být odhad příští události velmi užitečný. V principu rozdělíme integrální rovnici do dvou integrálů, z nich každý je řešen jinak:

$$f(x) = g(x) + \int_0^1 K(x, y) f(y) dy$$

$$f(x) = g(x) + h(x)$$

Ve funkci $h(x)$ můžeme neznámou funkci $f(x)$ vyjádřit jako součet známé funkce $g(x)$ a $h(x)$, kterou můžeme dále iteračně řešit:

$$h(x) = \int_0^1 K(x, y) f(y) dy$$

$$h(x) = \int_0^1 K(x, y) [g(y) + h(y)] dy$$

$$h(x) = \int_0^1 K(x, y) g(y) dy + \int_0^1 K(x, y) h(y) dy$$

Primární odhad tedy můžeme napsat jako:

$$\langle f(x) \rangle_{\text{přístí}} = g(x) + \langle h(x) \rangle_{\text{přístí}}$$

Kde:

$$\langle h(x) \rangle_{\text{přístí}} = \frac{K(x, \xi_1) g(\xi_1)}{p_1(\xi_1)} + \langle h(x) \rangle_{\text{přístí}}$$

$$\langle h(x) \rangle_{\text{přístí}} = \frac{K(x, \xi_1) g(\xi_1)}{p_1(\xi_1)} + \frac{K(\xi_1, \xi_2) g(\xi_2)}{p_2(\xi_2)} + \langle h(x) \rangle_{\text{přístí}}$$

$$\langle h(x) \rangle_{\text{přístí}} = \sum_{i=1}^k \frac{K(\xi_{i-1}, \xi_i) g(\xi_i)}{p_i(\xi_i)}$$

Finální odhad integrální rovnice je:

$$\langle f(x) \rangle_{\text{přístí}} = g(x) + \sum_{i=1}^k \frac{K(\xi_{i-1}, \xi_i) g(\xi_i)}{p_i(\xi_i)} \quad (3.31)$$

Kapitola 4

Monte Carlo metody v osvětlení

Tato kapitola má za úkol ukázat jak aplikovat Monte Carlo metody z předchozí kapitoly na řešení problémů globálního osvětlení. V úvodu jsou ve stručnosti popsány všechny současné algoritmy používající nestranné Monte Carlo metody. Následně jsou rozebrány zejména dva algoritmy – sledování cesty a sledování světla.

Algoritmus 1 Rekurzivní sledování paprsku

- 1) Pro každý pixel
 - (a) Vyberte paprsek procházející pixelem
 - (b) Zjistěte barvu pixelu sledováním paprsku (algoritmus 2)
-

Algoritmus 2 Rekurzivní sledování paprsku

- 1) Najdi nejbližší průsečík paprsku se scénou
 - 2) Pro každý světelný zdroj
 - (a) Vrhni stínový paprsek
 - (b) Pokud protnul paprsek světelný zdroj
 - (i) Vypočti přímé osvětlení daného bodu
 - (ii) Přičti toto osvětlení k barvě pixelu
 - 3) Pokud je materiál spekulární
 - (a) Vypočti odražený (lomený) paprsek
 - (b) Sleduj odražený paprsek
 - (c) Přičti jeho barvu k barvě pixelu
-

4.1 Přehled algoritmů

4.1.1 Rekurzivní sledování paprsku

Rekurzivní sledování paprsku (*recursive ray tracing - RRT*) navrhl na konci 80. let T. Whitted (23). Jedná se asi o nejznámější metodu globálního osvětlení, ačkoliv metoda není schopna vypočítat kompletní řešení zobrazovací rovnice, především vůbec neuvažuje o nepřímém osvětlení či kaustikách. Navíc pracuje pouze s bodovými světelnými zdroji, proto jsou výsledné stíny bez polostínu (tzv. tvrdé – hard), a transport světla omezuje na zrcadlové odrazy.

Principem je rekurzivní sledování dráhy určitého počtu paprsků každým pixelem. Nejdříve se vypočítá průsečík paprsku se scénou. Jestliže žádný není, vrátí se barva pozadí. V případě protnutí objektu se vypočítá barva (vlastní emise + osvětlení). U průsvitných materiálů se dále sleduje lomený paprsek. Pokud je nenulová spekulární složka, pak se paprsek sleduje rekurzivně dále (algoritmus 1). Více o metodě sledování paprsku lze nalézt v (24), (3),(25), (4), (1).

4.1.2 Distribuované sledování paprsku

Distribuované sledování paprsku (*distributed ray tracing - DRT*), někdy také zvané stochastické sledování paprsku, je algoritmus rozšiřující rekurzivní sledování paprsku. Byl představen R. Cookem (26), (27) v roce 1984.

DRT nahrazuje při vzorkování původně jediný vzorek, který používá RRT, integrálem. Při vzorkování lesklého odrazu (nebo lomu) nahrazuje původně jediný odražený paprsek (ve směru ideálního zrcadlového odrazu) několika paprsky. Plošné světelné zdroje jsou vzorkovány několika stínovými paprsky, což zajišťuje vznik stínů a polostínu (tzv. měkkých stínů). Podobně lze vzorkovat čas a zajistit rozmazání pohybem, objektiv za vzniku hloubky ostrosti, či vzorkovat světelné spektrum a na povrchu materiálu umožnit rozklad světla do několika paprsků, z nichž každý obsahuje světlo o jiné vlnové délce. Podrobnější informace v (3), (2), (4), (24).

Algoritmus 3 Sledování cesty

- 1) Pro každý pixel
 - (a) Pro každý vzorek
 - (i) Vygeneruj paprsek vzorkování pixelu
 - (ii) Vzorkuj ostatní věci (objektiv, čas)
 - (iii) Sleduj paprsek (algoritmus 4)
 - (b) Zprůměruj vzorky aritmetickým či váženým průměrem
-

Algoritmus 4 Sledování paprsku (sledování cesty)

- 1) Najdi nejbližší průsečík paprsku se scénou
 - 4) Pro každý světelný zdroj
 - (a) Pokud je světelný zdroj viditelný
 - (i) Vypočti přímé osvětlení daného bodu
 - (ii) Přičti toto osvětlení k barvě pixelu
 - 5) Vygeneruj paprsek podle materiálu
 - 6) Sleduj paprsek
 - 7) Přičti přicházející radianci vrženého paprsku k barvě vzorku
-

4.1.3 Sledování cesty

Sledování cesty (*path tracing* - *PT*) byla představenou J. Kajiyou (7) jako metoda na řešení zobrazovací rovnice. Na rozdíl od rekurzivního sledování paprsku poskytuje kompletní řešení zobrazovací rovnice, ačkoliv za cenu velkého výpočetního výkonu. Především je schopna zvládnout problémy osvětlení, které vyžadují řešení integrálu, jako například nepřímé osvětlení, vzorkování plošných světelných zdrojů a další. Podrobné informace jsou uvedeny v kapitole 4.2 a v algoritmu 3.

4.1.4 Radiační metoda

Radiační metoda byla v roce 1986 uvedena Goralem (28). Na rozdíl od všech ostatních metod v této kapitole, není založena na Monte Carlo metodách, ale na metodách konečných prvků, ačkoliv Monte Carlo metody mohou být použity na řešení dílčích problémů.

Ještě před samotným výpočtem je nutné zajistit, aby byla geometrie složena z elementů (geometrických primitiv, většinou trojúhelníků), není tedy možné použít při výpočtu nějakou matematicky přesnou reprezentaci geometrie. Následuje výpočet konfiguračních faktorů, které reprezentují geometrické vztahy mezi jednotlivými elementy. Posledním krokem je řešení soustavy lineárních rovnic pomocí nějaké z iteračních metod. Podrobnější informace zde (29), (30), (5).

4.1.5 Obousměrné sledování cesty

V roce 1993 představil E. Lafortune (31) obousměrné sledování cesty (*bidirectional path tracing* - *BPT*), o rok později byla tato metoda představena Veachem (32). Tento přístup vychází z duality radiance a potenciálu (kapitola 2.6) - spojuje algoritmy sledování cesty a sledování světla, vystřeluje paprsky, jak ze světelného zdroje, tak od pozorovatele, čímž redukuje nevýhody jednotlivých přístupů. Například pro výpočet kaustik je sledování cesty velice neúčinné, protože málokdy se odrazí paprsek od lesklého materiálu a protne světelný zdroj. Sledování světla naopak dokáže počítat kaustiky celkem efektivně, ale obtížně počítá nepřímé osvětlení, takže na odhad s nízkým rozptylem potřebuje obrovské množství paprsků.

4.1.6 Metropolis přenos světla

Metropolis sledování světla (*Metropolis light transport – MLT*) je metoda založená na myšlenkách N. Metropolis. Samotný algoritmus MLT byl představen Veachem a Guibasem (21), (33) v roce 1997. Celý koncept byl zcela odlišný od předchozích (PT, BPT), na rozdíl od náhodné distribuce vzorků, MLT distribuovalo vzorky podle neznámé funkce. V našem případě to znamená, že je obraz vzorkován podle přicházející radiance, což je velice užitečné, jelikož je vzorkování koncentrováno do více osvětlených oblastí scény. Ačkoliv jsou jednotlivé vzorky souvztažné (*correlated*), algoritmus je stále nestranný.

Metoda dává výborné výsledky především v situacích, kde běžné algoritmy selhávají – například scény, kde přichází veškeré světlo skrze nějaký velmi malý otvor. Jakmile algoritmus najde cestu skrz, pomocí mutací prozkoumá tuto cestu, aby vypočítal osvětlení. Účinnost tohoto řešení je daleko vyšší než náhodné vzorkování, jako v případě ostatních algoritmů.

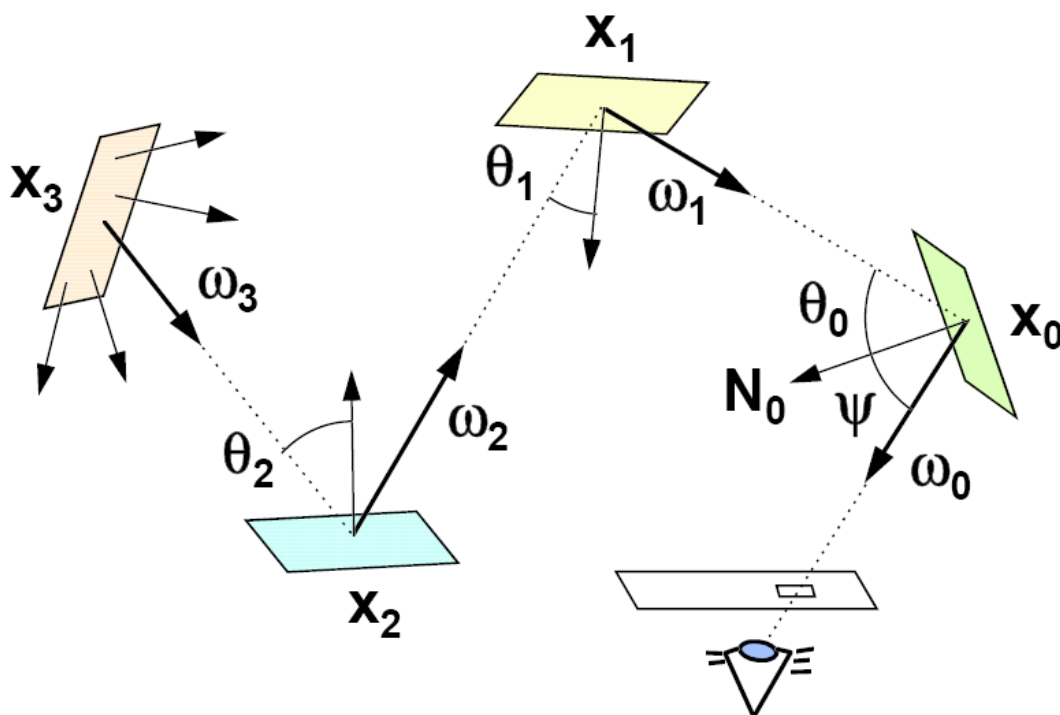
I MLT má problémy s výpočty kaustik, ačkoliv byly vytvořeny speciální mutace, které zlepšují výpočet kaustik. Účinnost těchto technik záleží na velikosti světelného zdroje. V případě malých zdrojů nejsou příliš efektivní a pro bodové světelné zdroje nefungují vůbec.

4.1.7 Fotonové mapy

Fotonové mapy (*photon mapping - PM*) jsou dvou-krokovou hybridní metodou vytvořenou H. Jensenem (34). Základní myšlenkou je reprezentovat osvětlení nezávisle na geometrii. V prvním kroku probíhá takzvané sledování fotonů (*photon tracing*), při kterém jsou sledovány fotony opouštějící světelné zdroje, a zároveň jsou ukládány do takzvané fotonové mapy, která je nezávislá na geometrii. V druhém kroku je fotonová mapa vizualizovaná pomocí sledování paprsku. PM nepatří mezi nestranné Monte Carlo metody (porovnání nestranných a zkruslených Monte Carlo metod v kapitole 5.1).

PM je velice robustní a disponuje stejnými výhodami jako všechny klasické algoritmy založené na Monte Carlo metodách, zároveň je ale daleko efektivnější, protože netrpí šumem jako nestranné MC metody. Algoritmus věnuje zvláštní péči kaustikám, proto, jako jediný zde zmíněný algoritmus, dokáže poměrně lehce vypočítat kaustiky v případě bodových zdrojů.

Metoda je podrobně rozebrána v páté kapitole, zatímco v šesté kapitole jsou uvedeny základní rozšíření. Další informace lze najít i v další literatuře (6), (35), (2), (4), (1).



Obrázek 4.1: Schéma šíření světla – střílení potenciálu, sbírání energie. (PE)

4.2 Sledování cesty

4.2.1 Základní algoritmus

Při odvozování vykreslovacího algoritmu založeném na Monte Carlo metodách, v našem případě sledování cesty, je potřeba si nejdříve uvědomit, že zjišťujeme barvu pixelu, což je vlastně zářivý tok procházející pixelem. Základní rovnicí je tedy rovnice 2.3, která vyjadřuje zářivý tok v jednotkách radiance.

$$\Phi = \int_A \int_{\Omega_x} L(x, \vec{\omega}_x) W_e(x, \vec{\omega}_x) |\vec{\omega}_x \cdot \vec{N}_x| d\vec{\omega}_x dA_x \quad (2.3)$$

Druhou rovnicí, kterou využijeme je zobrazovací rovnice 2.8.

$$L(x, \vec{\omega}_x) = L_e(x, \vec{\omega}_x) + \int_{\Omega^{-1}} L(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \quad (2.8)$$

Primární odhad je kombinací těchto dvou rovnic.

$$\langle \Phi \rangle_{cest} = \frac{W_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{p_0(x_0, \vec{\omega}_0)} \langle L(x_0, \vec{\omega}_0) \rangle_{cest} \quad (4.1)$$

Kde:

- $p_0(x_0, \vec{\omega}_{x_0})$ = funkce hustoty pravděpodobnosti (PDF)

Nultou PDF p_0 lze vypočítat:

$$p_0(x_0, \vec{\omega}_0) = \frac{p_{-1}(x_{-1}, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{\|x_{-1} - x_0\|^2}$$

Kde:

- p_{-1} = PDF na čočce objektivu

$\langle L(x_0, \vec{\omega}_{x_0}) \rangle_{cest}$ jako Fredholmova rovnice druhého řádu vede k Neumannově řadě (viz. kapitola 3.3), která se řeší iterací (rekurzí):

$$\begin{aligned} L^0 &= L_e \\ L^1 &= L_e + TL^0 \\ &\dots \\ L^n &= L_e + TL^{n-1} \end{aligned}$$

$$L^n = \sum_{i=0}^{\infty} T^i L_e \quad (4.2)$$

Rovnici můžeme rozepsat:

$$L^n(\vec{x}, \vec{\omega}) = \int_{\Omega} \dots \int_{\Omega} L_e(\vec{x}_0, \vec{x}_1) f_r(\vec{x}_0, \vec{\omega}_0 \rightarrow \vec{\omega}_1) \dots f_r(\dots) d\vec{\omega}_0 d\vec{\omega}_1 \dots d\vec{\omega}_n$$

Tento integrál si lze snadno intuitivně představit jako cesty světelných paprsků. Každá cesta má určitou délku n (počet odrazů) a sadu bodů (vertexů), která ji charakterizuje, kde bod s nejnižším indexem se nachází na světelném zdroji a další body na ostatních objektech ve scéně. Celkový příspěvek všech cest dané délky lze získat integrováním přes všechny existující body ve scéně. To zahrnuje n integrálů, kde jsou jednotlivé cesty váženy. Úplné řešení je součet všech existujících cest světelných paprsků (cest všech délek).

Rovnice lze převést na takzvanou *Measurement rovnici*, která je vázána k řešení zobrazovací rovnice v procesu formování obrazu v kameře.

$$M = \int_A \int_{\Omega} \int_T R(\vec{x}, \vec{\omega}, t) L(\vec{x}, \vec{\omega}, t) dt d\vec{\omega} dA \quad (4.3)$$

Funkce R závisí na několika věcech:

- Filtru pixelu (\vec{x})
- Cloně ($\vec{\omega}$)
- Závěrci (t)

S použitím ruské rulety můžeme v rovnici 4.2 ukončit iteraci (rekurzí) aniž bychom uvedli do řešení nestrannost:

$$\langle L(x_0, \vec{\omega}_0) \rangle_{cest} = \sum_{i=0}^l \left[\prod_{j=1}^i \frac{f_r(x_{j-1}, \vec{\omega}_j, \vec{\omega}_{j-1}) |\vec{\omega}_j \cdot \vec{N}_{j-1}|}{P_j p_j(\vec{\omega}_j)} \right] L_e(x_i, \vec{\omega}_i) \quad (4.4)$$

Kde:

- l = délka náhodné procházky
- $x_0 \dots x_l$ = body náhodné procházky
- P_j = pravděpodobnost pokračování krokem j
- $p_j(\vec{\omega}_j)$ = hustota pravděpodobnosti pro směr $\vec{\omega}_j$

Celkový odhad sledování cesty je tedy:

$$\langle \Phi \rangle_{cest} = \frac{W_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{p_0(x_0, \vec{\omega}_0)} \times \sum_{i=0}^l \left[\prod_{j=1}^i \frac{f_r(x_{j-1}, \vec{\omega}_j, \vec{\omega}_{j-1}) |\vec{\omega}_j \cdot \vec{N}_{j-1}|}{P_j p_j(\vec{\omega}_j)} \right] L_e(x_i, \vec{\omega}_i) \quad (4.5)$$

Je dobré podotknout, že člen $W_e(x_0, \vec{\omega}_0)$ závisí na modelu kamery, které jsou mimo rámec této práce. Základní informace lze zjistit v disertační práci Erika Lafortuna (36), který se v dodatku A zabývá výpočtem W_e . Podrobněji se kamerami zabývají M. Pharr a G. Humphreys v (4), kde se celou šestou kapitolu věnují různým modelům kamer.

4.2.2 Vzorkování podle důležitosti

Hned prvním místem, kde najde vzorkování podle důležitosti využití, je vzorkování objektivu. Konkrétně nás zajímá PDF $p_0(x_0, \vec{\omega}_0)$, které je použito v odhadu integrálu pomocí sledování cesty:

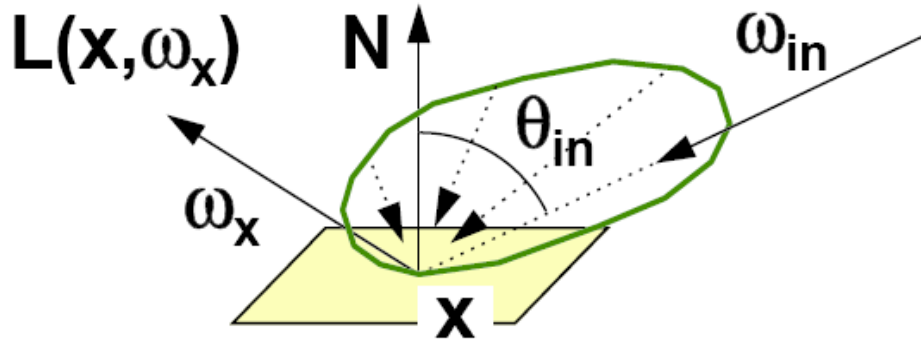
$$\langle \Phi \rangle_{cest} = \frac{W_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{p_0(x_0, \vec{\omega}_0)} \langle L(x_0, \vec{\omega}_0) \rangle_{cest} \quad (4.16)$$

Jelikož člen $W_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|$ známe, můžeme ho použít pro výpočet $p_0(x_0, \vec{\omega}_0)$.

$$p_0(x_0, \vec{\omega}_0) = \frac{W_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{W} \quad (4.6)$$

Kde W je normalizační faktor:

$$W = \int_A \int_{\Omega} W_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0| d\vec{\omega}_0 d\vec{\mu}_0$$



Obrázek 4.2: Vzorkování podle důležitosti podle BRDF. (PE)

PDF 4.21 zajistí, že jsou body a směry vzorkovány podle jejich důležitosti, tedy jejich vlivu na výsledné řešení.

V kontextu metod založených na metodě sledování paprsku se vzorkování podle důležitosti většinou používá při vzorkování hemisféry při výpočtu odražených paprsků. Například při vzorkování lesklého materiálu je optimální vybrat více paprsků v regionech, kam by se odrazil ideálně odražený paprsek (obrázek 4.2). Při vzorkování vycházíme ze zobrazovací rovnice:

$$L(x, \vec{\omega}_x) = L_e(x, \vec{\omega}_x) + \int_{\Omega^{-1}} L(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \quad (2.8)$$

V rovnici v každém případě známe člen $f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x|$ v každém bodu rekurze, který splňuje podmínku, že je menší než 1, což nám umožňuje ho použít ke konstrukci sub-kritické hustoty pravděpodobnosti (obrázek 4.3):

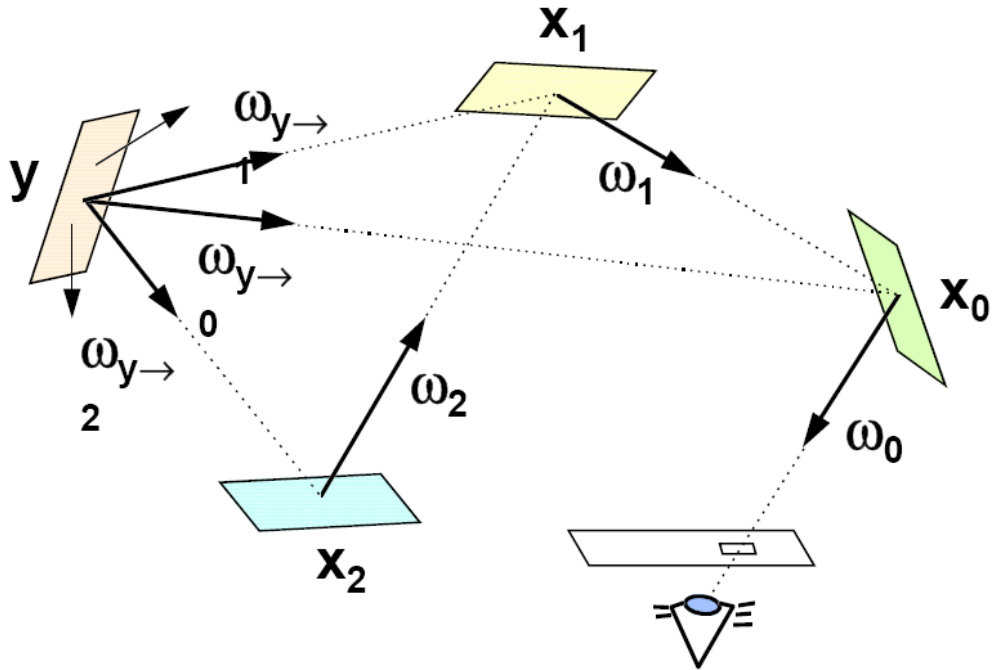
$$p_x(x_x, \vec{\omega}_x) = \frac{f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x|}{P_x} \quad (4.7)$$

Kde P_x je kumulativní funkce hustoty pravděpodobnosti p_x (definice v kapitole 3.2.2). Díky použití sub-kritické hustoty pravděpodobnosti se primární odhad zjednoduší na:

$$\langle \Phi \rangle_{cest, d\ddot{u}l} = W \sum_{i=0}^l L_e(x_i, \vec{\omega}_{x_i}) \quad (4.8)$$

Vzorkování také v mnohém záleží na použitém BRDF (informace o některých BRDF lze nalézt v dodatku A) (4).

Je dobré upozornit na fakt, že PDF není ideální, protože nebere v úvahu přicházející radiance, která má velký vliv na tvar integrantu. Problém je, že radiance je neznámá funkce, nelze s ní tedy počítat v PDF, pokud není provedeno sledování světla a následná radiance či irradiance není uložena v nějaké světelné mapě (*light map*), viz. (37).



Obrázek 4.3: Schéma šíření světla s odhadem příští události – střílení potenciálu, sbírání energie. (PE)

4.2.3 Odhad příští události

Ve většině scén, které se běžně vykreslují, jsou světelné zdroje poměrně malé, takže je poměrně mizivá pravděpodobnost, že daná náhodná procházka přispěje k výslednému odhadu světelného výkonu. Výrazné zlepšení může být dosaženo rozdělením nepřímého osvětlení na dvě složky:

$$L_r(x, \vec{\omega}_x) = \int_{\Omega^{-1}} L(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y$$

$$L_r(x, \vec{\omega}_x) = \int_{\Omega^{-1}} [L_e(y, \vec{\omega}_y) + L_r(y, \vec{\omega}_y)] f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y$$

$$L_r(x, \vec{\omega}_x) = \int_{\Omega^{-1}} L_e(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y + \int_{\Omega^{-1}} L_r(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y$$

$$L_r(x, \vec{\omega}_x) = L_d(x, \vec{\omega}_x) + L_i(x, \vec{\omega}_x)$$

Kde

- $L_d(x, \vec{\omega}_x)$ = příspěvek přímého osvětlení
- $L_i(x, \vec{\omega}_x)$ = příspěvek nepřímého osvětlení

Při výpočtu přímého příspěvku může být vzorkování přes celou hemisféru nepraktické. Daleko lepší výsledky poskytuje vzorkování přes plochu světelného zdroje A (obrázek 4.3):

$$L_d(x, \vec{\omega}_x) = \int_{\Omega^{-1}} L_e(x, -\vec{\omega}_i) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_i$$

$$L_d(x, \vec{\omega}_x) = \int_A L_e(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) G(x, y) d\mu_y$$

$$L_d(x, \vec{\omega}_x) = \int_A L_e(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) v(x, y) \frac{|\vec{\omega}_{y \rightarrow x} \cdot \vec{N}_x| |\vec{\omega}_{y \rightarrow x} \cdot \vec{N}_y|}{\|x - y\|^2} d\mu_y \quad (4.9)$$

Kde:

- $G(x, y)$ = geometrický člen
- $v(x, y)$ = člen viditelnosti

Pokud jsou integrály vzorkovány podle důležitosti (podle PDF 4.21 a 4.22), odhad se změní na:

$$\langle \Phi \rangle_{cest, d\hat{u}l, ud\hat{a}l} = W \left[L_e(x_0, \vec{\omega}_{x_0}) + \frac{L}{L(y)} \sum_{i=0}^t L_e(y, \vec{\omega}_{y \rightarrow x_i}) \times f_r(x, \vec{\omega}_{y \rightarrow x_i}, \vec{\omega}_{x_i}) v(x, y) \frac{|\vec{\omega}_{y \rightarrow x_i} \cdot \vec{N}_x| |\vec{\omega}_{y \rightarrow x_i} \cdot \vec{N}_y|}{\|x_i - y\|^2} \right]$$

Kde $p(y)$ je PDF pro přímé osvětlení:

$$p(y) = \frac{L}{L(y)}$$

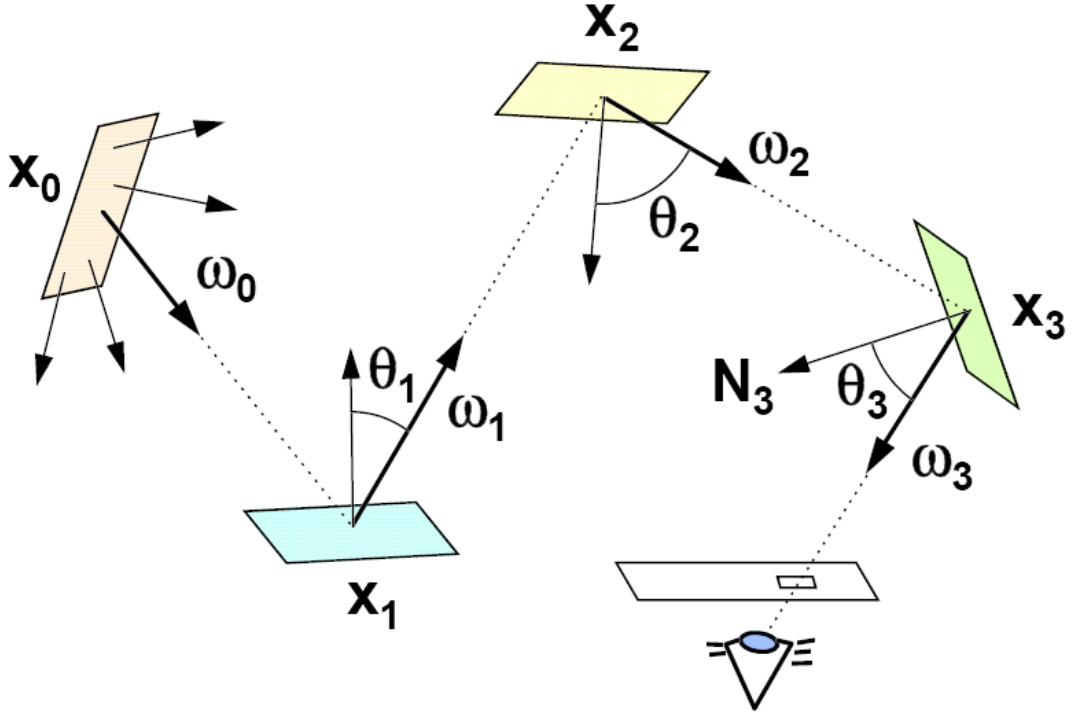
- L = celkový vlastní emitovaný zářivý výkon ve scéně

$$L = \int_A \int_{\Omega_y} L_e(y, \vec{\omega}_y) |\vec{\omega}_y \cdot \vec{N}_y| d\vec{\omega}_y d\vec{\mu}_y$$

- $L(y)$ = vlastní emitovaná radiosita v bodě y

$$L(y) = \int_{\Omega_y} L_e(y, \vec{\omega}_y) |\vec{\omega}_y \cdot \vec{N}_y| d\vec{\omega}_y$$

Odhad příští události výrazně zvyšuje efektivitu ve scénách, kde jsou světelné zdroje poměrně malé, viditelné a přímé osvětlení je dominantní. Významnou roli také hraje, jak jsou vzorkována světla, proto bylo o tomto tématu vydáno několik publikací (38), (39), (40), (22).



Obrázek 4.4: Schéma šíření světla – střílení energie, sbírání potenciálu. (PE)

4.3 Sledování světla

4.3.1 Základní algoritmus

Sledování světla, stejně jako sledování cesty, využívá při odvozování dvou rovnic. Opět se snažíme zjistit zářivý výkon procházející pixelem, který zde vyjádříme pomocí potenciálu (obrázek 4.4):

$$\Phi = \int_A \int_{\Omega_x} L_e(x, \vec{\omega}_x) W(x, \vec{\omega}_x) |\vec{\omega}_x \cdot \vec{N}_x| d\vec{\omega}_x d\vec{\mu}_x \quad (2.17)$$

Druhou rovnicí je potenciaální rovnice, kterou použijeme pro výpočet odraženého potenciálu.

$$W(x, \vec{\omega}_x) = W_e(x, \vec{\omega}_x) + \int_{\Omega} W(y, \vec{\omega}_y) f_r(y, \vec{\omega}_x, \vec{\omega}_y) |\vec{\omega}_y \cdot \vec{N}_y| d\omega_y \quad (2.12)$$

Primární odhad je podobný, jako u sledování cesty, jen je použit potenciál:

$$\langle \Phi \rangle_{\text{svět}} = \frac{L_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{p_0(x_0, \vec{\omega}_0)} \langle W(x_0, \vec{\omega}_0) \rangle$$

$\langle W(x_0, \vec{\omega}_0) \rangle$ je Neumannova řada, kterou můžeme řešit iterací. S použitím ruské rulety na omezení délky náhodné procházky můžeme odhad rozepsat jako:

$$\langle \Phi \rangle_{svět} = \frac{L_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{p_0(x_0, \vec{\omega}_0)} \sum_{i=0}^l \left[\prod_{j=1}^i \frac{f_r(x_j, \vec{\omega}_{j-1}, \vec{\omega}_j) |\vec{\omega}_j \cdot \vec{N}_j|}{P_j p_j(\vec{\omega}_j)} \right] W_e(x_i, \vec{\omega}_i)$$

Kde:

- l = délka náhodné procházky
- $x_0 \dots x_l$ = body náhodné procházky
- P_j = pravděpodobnost pokračování krokem j
- $p_j(\vec{\omega}_j)$ = hustota pravděpodobnosti pro směr $\vec{\omega}_j$

Algoritmus sledování světla je v literatuře také někdy nazýván zpětné sledování cesty, protože na rozdíl od běžného sledování cesty neposíláme paprsky od pozorovatele, ale raději od světelného zdroje.

4.3.2 Vzorkování podle důležitosti

Vzorkování nultého PDF je podobné jako v metodě sledování cesty (rovnice 4.21), jediným rozdílem je, že zde, namísto s potenciály, pracujeme s radiancí.

$$p_0(x_0, \vec{\omega}_0) = \frac{L_e(x_0, \vec{\omega}_0) |\vec{\omega}_0 \cdot \vec{N}_0|}{L} \quad (4.10)$$

Při sledování světla vzorkujeme světelné zdroje, kterých může být více, na rozdíl od pozorovatelů. PDF 4.25 zajišťuje, že světelné zdroje s největší energií jsou vzorkovány nejčastěji a stejně tak, aby bylo nejvíce paprsků ve směru, kde přispějí nejvíce k výslednému osvětlení.

Vzorkování podle BRDF (4.22) je úplně stejné jako v případě sledování cesty:

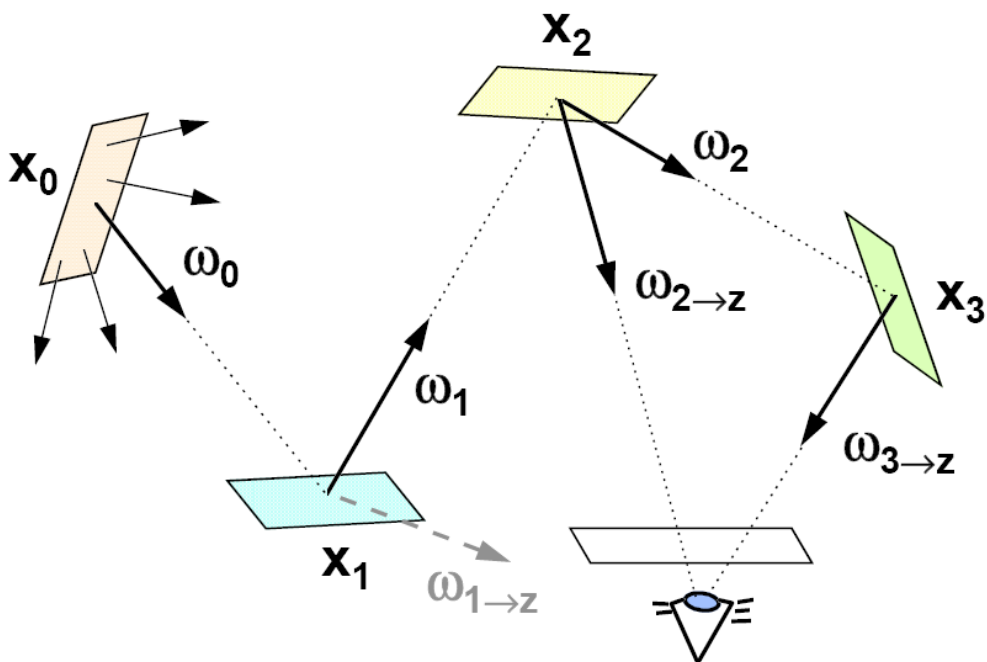
$$p_x(x_x, \vec{\omega}_x) = \frac{f_r(x_i, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x|}{P_x}$$

Pokud použijeme toto sub-kritické PDF, primární odhad se zjednoduší na:

$$\langle \Phi \rangle_{svět, důl} = L \sum_{i=0}^l W_e(x_i, \vec{\omega}_{x_i})$$

4.3.3 Odhad příští události

Stejně jako pro metodu sledování cesty, i zde je odhad příští události klíčovým rozšířením, možná ještě důležitějším než v prvním algoritmu. Opět rozdělíme integrovanou funkci na dvě části:



Obrázek 4.5: Schéma šíření světla s odhadem příští události – střílení energie, sbírání potenciálu. (PE)

$$\begin{aligned}
 W_r(x, \vec{\omega}_x) &= \int_{\Omega^{-1}} W(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \\
 W_r(x, \vec{\omega}_x) &= \int_{\Omega^{-1}} [W_e(y, \vec{\omega}_y) + W_r(y, \vec{\omega}_y)] f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \\
 W_r(x, \vec{\omega}_x) &= \int_{\Omega^{-1}} W_e(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \\
 &\quad + \int_{\Omega^{-1}} W_r(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_y \\
 W_r(x, \vec{\omega}_x) &= W_d(x, \vec{\omega}_x) + W_i(x, \vec{\omega}_x)
 \end{aligned}$$

Kde:

- $W_d(x, \vec{\omega}_x)$ = přímý potenciál
- $W_i(x, \vec{\omega}_x)$ = nepřímý potenciál

Přímý potenciál může být účinněji vzorkován přes objektiv kamery A:

$$\begin{aligned}
 W_d(x, \vec{\omega}_x) &= \int_{\Omega^{-1}} W_e(x, -\vec{\omega}_i) f_r(x, \vec{\omega}_y, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_i \\
 W_d(x, \vec{\omega}_x) &= \int_A W_e(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) G(x, y) d\mu_y \\
 W_d(x, \vec{\omega}_x) &= \int_A L_e(y, \vec{\omega}_y) f_r(x, \vec{\omega}_y, \vec{\omega}_x) v(x, y) \frac{|\vec{\omega}_{y \rightarrow x} \cdot \vec{N}_x| |\vec{\omega}_{y \rightarrow x} \cdot \vec{N}_y|}{\|x - y\|^2} d\mu_y
 \end{aligned} \tag{4.11}$$

Kde:

- $G(x, y)$ = geometrický člen
- $v(x, y)$ = člen viditelnosti

Výsledný primární odhad metody sledování světla se vzorkováním podle důležitosti a odhadem další události lze napsat jako:

$$\langle \Phi \rangle_{svět, dût, udál} = L \left[W_e(x_0, \vec{\omega}_0) + A \sum_{i=0}^t W_e(x_{i+1}, \vec{\omega}_{x_{i+1} \rightarrow z}) \times f_r(x_{i+1}, \vec{\omega}_{y \rightarrow x_i}, \vec{\omega}_{x_i}) v(x, y) \frac{|\vec{\omega}_{y \rightarrow x_i} \cdot \vec{N}_x| |\vec{\omega}_{y \rightarrow x_i} \cdot \vec{N}_y|}{\|x_i - y\|^2} \right]$$

Kde $p(z)$ je PDF pro vzorkování přímého potenciálu:

$$p(z) = \frac{1}{A}$$

Kapitola 5

Fotonové mapy

V této kapitole bude vysvětlen základní algoritmus dvou-krokových fotonových map. Ještě před tím bude nastíněna problematika zkreslených Monte Carlo metod, mezi které metoda fotonových map patří. Následně budou probrány jednotlivé kroky při výpočtu osvětlení, od tvorby fotonové mapy, přes sledování fotonů až po vizualizaci fotonových map.

5.1 Nestranné a zkreslené Monte Carlo metody

V kapitole 3.1 byl definován odhad jako nestranný, pokud splňoval následující podmínku:

$$E(\langle I \rangle_{prim}) = \int_0^1 f(x) dx = I \quad (3.3)$$

Ačkoliv zkreslené Monte Carlo metody tuto podmínku nesplňují, neznamená to, že všechny zkreslené metody dávají špatné výsledky. Některé zkreslené metody lze označit jako konzistentní (včetně fotonových map), což znamená, že čím více je vzorků, tím přesnější je řešení. Respektive, pokud zvýšíme počet vzorků, řešení se zpřesní.

Zkreslené metody mají oproti nestranným ještě další zdroj nepřesnosti ϵ :

$$E(\langle I \rangle_{prim}) = I + \epsilon \quad (5.1)$$

Proč tedy používat zkreslené MC metody? Hlavním důvodem je pomalá konvergence nestranných MC metod. Pokud není použito opravdu velké množství paprsků, ve výsledném řešení je obsažen šum, jako vysokofrekvenční rozptyl. S použitím zkreslených MC metod lze poměrně snadno redukovat šum, řešení tedy vypadá přirozeněji. Vše se bohužel neobejde bez uvedení další chyby (viz. rovnice 5.1), která nemůže být kontrolována. V tomto tkví hlavní nebezpečí zkreslených metod. Každá technika musí být řádně otestována a musí být i zjištěn její vliv na výsledné řešení. Přirozeně tedy chceme, aby všechny zkreslené techniky, které použijeme, byly konzistentní a při zvýšení počtu vzorků se zvýšila i přesnost výsledku.

Dalším problémem nestranných metod je fakt, že nemůžeme ukončit výpočet, když je řešení dostatečně dobré, protože už tímto bychom uvedli do řešení zkreslení.

5.2 Fotonová mapa

Fotonová mapa (*photon map* – *PM*) je datová struktura, do které jsou ukládány jednotlivé fotony „cestující“ scénou při sledování fotonů (kapitola 5.4). PM zajišťuje, že uložené osvětlení (respektive zářivý výkon přicházející k danému povrchu) je nezávislé na geometrii scény, což umožňuje použít přesnou matematickou reprezentaci geometrie jako u klasických algoritmů založených na MC metodě.

PM není odpovědná pouze za ukládání fotonů, ale zároveň za vyhledávání nejbližších fotonů vzhledem k nějakému bodu. To nachází využití především v druhé fázi algoritmu, kdy probíhá vizualizace fotonové mapy. Pro komplexní scény je často potřeba uložit i několik milionů fotonů, z kterých je potřeba co nejrychleji vyhledat jen několik stovek či tisíc nejbližších. Algoritmus musí být tedy velmi rychlý. O tom, jaké datové struktury se hodí na reprezentaci fotonové mapy, je diskutováno v kapitole 5.2.2. Podstatná je i reprezentace jednotlivých fotonů.

5.2.1 Reprezentace fotonu

Pro každý foton musíme uložit následující údaje:

- Pozice v 3D prostoru
- Nesený světelný výkon
- Směr, ze kterého přišel
- (Dělicí rovina)

Za předpokladu, že bude výsledný program používán k vykreslování komplexních scén, je poměrně výhodné použít datové typy, které nezabírají moc místa v paměti, jelikož v případě sledování několika desítek milionů fotonů jsou paměťové nároky značné. Nesený výkon může být buď pomocí tří složek (RGB) nebo přesněji reprezentován pomocí několika vzorků v případě, že je použito spektrální vykreslování. Je silně doporučeno předpočítat všechny možné směry ještě před ukládáním fotonů do fotonové mapy. Dělicí rovina je uvedena v závorce, protože je použita, pouze pokud je použit KD-strom k reprezentaci fotonové mapy.

5.2.2 Reprezentace fotonové mapy

Jak bylo naznačeno v předchozí kapitole, nejčastěji používanou datovou strukturou na ukládání fotonů je KD-strom (41),(42). KD-strom je v podstatě K -dimenzionální binární strom určený především pro rychlé vyhledávání. Každý uzel si lze představit jako K -dimenzionální bod. Pokud má uzel dva potomky, pak v podstatě vytváří rovinu, jenž je kolmá na jednu z hlavních os, a která dělí prostor na dva pod-prostory. Všechny body vlevo od roviny jsou v levém podstromu a všechny body napravo od roviny jsou v pravém podstromu. KD-stromy může být takzvaně vybalancován, což znamená, že se snažíme co nejvíc snížit výšku stromu a tím i potencionální nejdelší cestu od kořene stromu do listu, který je v nejnižší vrstvě. Při hledání fotonu ve vybalancovaném KD-stromu může v nejhorším případě nastat $O(\log n)$. Pokud vezmeme tento fakt a zároveň uvážíme, že KD-strom může být reprezentován velice efektivně, je jasné, proč je pro reprezentaci fotonové mapy přirozenou volbou.

5.3 Sledování fotonů

5.3.1 Vzorkování světla

Při sledování fotonů je nejdříve nutné, aby byl samotný foton „vygenerován“, což v praxi znamená, že mu musí být vypočítána jeho počáteční pozice, jeho směr, kterým se bude pohybovat a také výkon, která ponese. Především jeho pozice a směr jsou velmi závislé na typu světelného zdroje, který foton opouští.

Bodový světelný zdroj

Úplně nejjednodušším světelným zdrojem je bodový světelný zdroj (*point light*), který si lze představit jako nekonečně malý bod ve scéně, který svítí do všech stran stejnou intenzitou.

Vzorkování bodového světelného zdroje je velice jednoduché, protože jeho počáteční poloha je vždy stejná, není tedy nutné vzorkovat samotnou plochu světelného zdroje. Při vzorkování směru lze použít dvě metody. První, jednodušší, je pomocí takzvaného zamítacího vzorkování (*rejection sampling*), které pracuje na jednoduchém principu (algoritmus 5). Druhou možností je vzorkovat jednotkovou sféru, která obaluje daný bod, pomocí vzorkování jednotkového čtverce. Následně můžeme čtverec jednoduše namapovat na sféru. V tomto případě je samozřejmě ideální zajistit, aby byly paprsky rovnoměrně rozmístěny v celé doméně vzorkování, což lze například vzorkováním po částech (kapitola 3.2.1).

Algoritmus 5 Vzorkování bodového světelného zdroje

- 1) Nastav počet fotonů na nulu
 - 2) Opakuj, dokud není vystřeleno požadované množství fotonů:
 - (a) Opakuj
 - (i) Vygeneruj pro každou souřadnici x, y, z náhodné číslo v intervalu $[0,1]$
 - (b) dokud není čtvereční délka vektoru $\vec{v}(x, y, z)$ větší než jedna
$$x^2 + y^2 + z^2 > 1$$
 - (c) Nastav pozici fotonu na pozici světla
 - (d) Nastav směr fotonu na vygenerovaný směr
 - (e) Zvyš počet fotonů o jedna
 - (f) Sleduj foton
-

Směrový světelný zdroj

Směrový světelný zdroj je vlastně bodový zdroj v nekonečné vzdálenosti. Všechny paprsky směřují stejným směrem, který je pro daný zdroj předem daný. Zbývá tedy vzorkovat „povrch“ světla (algoritmus 6). Většinou je vzorkován disk (vzorkování jednotkového disku - algoritmus 7) o poloměru, který závisí na velikosti scéně. Normála disku je shodná se směrem paprsků.

Algoritmus 6 Vzorkování směrového světleného zdroje

- 1) Nastav počet fotonů na nulu
 - 2) Opakuj, dokud není vystřeleno požadované množství fotonů:
 - (a) Nastav směr fotonu na směr světla
 - (b) Vzorkuj disk (algoritmus 3)
 - (c) Transformuj počátek paprsku (vzorkovaný bod) do světových souřadnic
 - (d) Zvyš počet fotonů o jedna
 - (e) Sleduj foton
-

Algoritmus 7 Uniformní vzorkování jednotkového disku

- 1) Vygeneruj dvě náhodná čísla ξ_1 a ξ_2 (vzorkování jednotkového čtverce)
- 2) Vzdálenost od středu může být vypočtena jako:

$$r = \sqrt{\xi_1}$$

- 3) Úhel θ jako:

$$\theta = 2\pi\xi_2$$

- 4) Převed' souřadnice na kartézské:

$$x = r \cos \theta$$

$$y = r \sin \theta$$

Reflektor

Reflektor je také bodový zdroj, který ovšem nesvítí do všech stran stejně. Směr svícení je definován dvěma souosými kužely. Vnitřní kužel reprezentuje jasné světlo, jehož intenzita klesá pouze se vzdáleností od vrcholu. Vnější kužel představuje hranici světla, za kterou už nedopadá. Navíc je světlo tlumeno se vzrůstající vzdáleností od osy kuželu.

Vzorkování reflektoru je poměrně jednoduché (algoritmus 8). Počáteční bod paprsku je roven pozici světla a směry získáme pomocí vzorkování kužele (algoritmus 9).

Algoritmus 8 Vzorkování reflektoru

- 2) Nastav počet fotonů na nulu
 - 3) Opakuj, dokud není vystřeleno požadované množství fotonů:
 - (a) Nastav pozici fotonu na pozici světelného zdroje
 - (b) Vzorkuj kužel (algoritmus 9)
 - (c) Transformuj počátek paprsku (vzorkovaný bod) do světových souřadnic
 - (d) Zvyš počet fotonů o jedna
 - (e) Sleduj foton
-

Algoritmus 9 Uniformní vzorkování jednotkového kužele

1) Vypočti $\cos \theta$ lineární interpolací:

$$\cos \theta = (1 - \xi_1) * \cos \theta_{max} + \xi_1$$

2) Vypočti $\sin \theta$:

$$\sin \theta = \sqrt{1 - \cos^2 \theta}$$

3) Vypočti ϕ (azimut):

$$\phi = 2\pi \xi_2$$

4) Převed' souřadnice na kartézské:

$$x = \cos \phi \sin \theta$$

$$y = \sin \phi \sin \theta$$

$$z = \cos \theta$$

Obdélníkový světelný zdroj

Obdélníkový světelný zdroj je vlastně speciálním případem plošného světleného zdroje. V moderní počítačové grafice je asi nejpoužívanějším zdrojem při testování algoritmů, které je většinou prováděno ve scéně zvané Cornell box.

Při vzorkování můžeme vzorkovat přímo obdélník nebo ho rozdělit na dva trojúhelníky a ty vzorkovat zvlášť. V druhém případě je však horší rovnoměrnost vzorkování než v prvním případě, pokud jsou použity techniky vzorkování po částech, protože je nutné náhodně vzorkovat jeden či druhý trojúhelník.

Vzorkování směru je vlastně neuniformní vzorkování hemisféry podle kosinu úhlu mezi normálou a daným paprskem Podrobnější informace v (39),(38).

5.3.2 Sledování fotonů

Jakmile máme vygenerovaný foton, vrhneme ho do scény (podobně jako paprsek při sledování paprsků). Mohou nastat dvě situace. První, poměrně neobvyklá, je, že foton neprotne žádný povrch ve scéně. V tomto případě okamžitě ukončíme další sledování a vygenerujeme nový foton. V případě scény, která by byla „otevřená“ natolik, že by byla při generování fotonů velká šance, že půjde směrem, kde nic není, je podstatné omezit generování fotonů na určitý směr. To lze pomocí takzvaných projekčních map (*projection maps*) (6).

Ve většině případů foton narazí na nějaký povrch. Tehdy by měl být uložen do fotonové mapy, pokud je povrch difúzní. Při samotné interakci s povrchem materiálu se mohou stát tři věci. Foton může být odražen (reflected), absorbován (absorbed) nebo může být odražen dovnitř materiálu (transmitted). Jaká z těchto událostí proběhne, záleží na daném materiálu.

Ve zjednodušeném případě můžeme uvažovat o třech událostech:

- Foton byl absorbován
- Foton byl odražen difúzně

- Foton byl odražen spekulárně

Reprezentovat odrazivost můžeme například hodnotou v intervalu $\langle 0,1 \rangle$, kde 1 znamená, že se foton odrazí na 100%, zatímco 0, že bude zcela pohlcen. V praxi tedy můžeme mít reprezentovanou odrazivost pro každou složku zvlášť a jednoduše vynásobit energii fotonu touto odrazivostí. Předtím budeme počítat směr, ve kterém se foton odrazil, je dobré zkontrolovat, zda foton nese nějakou energii, protože pokud je například povrch naprosto černý, veškerá energie bude pohlcena a nemá smysl sledovat foton, který nenese žádnou energii, tedy nepřispěje k výslednému řešení. Důležité je si také uvědomit, že nejen po odrazu od černého povrchu může foton „přijít“ o energii. Foton může při protnutí povrchu nést už pouze „červenou barvu“ a pokud narazí na modrý povrch, červená složka bude pohlcena. Je dobré si uvědomit, že takto jednoduchá pravidla platí, pouze pokud zanedbáváme fakt, že světlo má určitou vlnovou délku a redukuje ji na 3 složky (RGB).

Stále zbývá otázka, jak rozhodnout, zda se má foton odrazit difúzně nebo spekulárně, pokud je materiál kombinací obou. K tomuto účelu využijeme opět pravděpodobnosti, přesněji, ruské rulety. Řekněme, že máme indexy ρ_d a ρ_s , jejichž součet je roven jedné. Pak můžeme vygenerovat náhodné číslo ξ v intervalu $\langle 0,1 \rangle$ a platí:

$$\begin{aligned}\xi \in \langle 0, \rho_d \rangle &\rightarrow \text{difuzní odraz} \\ \xi \in \langle \rho_d, 1 \rangle &\rightarrow \text{spekulární odraz}\end{aligned}$$

Ruská ruleta může být rozšířena ještě o jednu možnost – absorpci, protože ne všechny materiály mají $\rho_d + \rho_s$ rovny jedné.

$$\begin{aligned}\xi \in \langle 0, \rho_d \rangle &\rightarrow \text{difuzní odraz} \\ \xi \in \langle \rho_d, \rho_d + \rho_s \rangle &\rightarrow \text{spekulární odraz} \\ \xi \in \langle \rho_d + \rho_s, 1 \rangle &\rightarrow \text{absorbce}\end{aligned}$$

To teda znamená, že pokud proběhne absorpce, foton není dále sledován. Pokud celkové ρ rovno 0,5, pak bude polovina fotonů absorbována, takže tím bude významně redukován počet fotonů, které budou muset být dále sledovány, což je významné urychlení. Nicméně je nutné udělat patřičné úpravy při výpočtu odraženého výkonu fotonů, které nebyly absorbovány, aby byla udržována rovnováha odražené energie.

$$\Phi_o = \frac{\Phi_i}{\rho}$$

Jelikož je dané ρ menší než 1 (většinou), odcházející výkon bude větší, protože musíme připočítat i výkon, který by byl odražen paprsky, které by byly vygenerovány a dále sledovány v případě, kdybychom nepoužili ruskou ruletu na jejich ukončení.

Zatím jsme počítali pouze s jednou složkou, v praxi se používají složky 3 (RGB) nebo více. Jediným rozdílem je, že je potřeba vypočítat průměrnou složku a tu použít při zjišťování, zda bude foton odražen či absorbován. Odcházející výkon bude samozřejmě počítán pro všechny složky.

Použití ruské rulety má i další výhodu – pomáhá odstraňovat fotony s nízkou energií, které nejsou příliš podstatné pro výsledné řešení. Nebyť ruské rulety, ve fotonové mapě by byla většina fotonů, jenž by měly nízký výkon. Pak by se našlo daleko méně fotonů, kteří by měli naopak opravdu velkou energii (např. by byly vystřeleny z nějakého malého, ale výkonného světelného zdroje, a uloženy před prvním odrazem). Při následné vizualizaci mohou takto

rozdílné fotony způsobovat problémy. Daleko lepší je, pokud všechny fotony nesou přibližně stejnou energii.

5.3.3 Ukládání fotonů

Jak již bylo zmíněno, fotony má smysl ukládat pouze na povrchu, který není čistě spekulární. U čistě spekulárních materiálů se fotonová mapa při vykreslování nepoužívá. Dále bylo dříve uvedeno, že metoda fotonových map se speciálně zaměřuje na kaustiky, proto jsou běžně používány dvě fotonové mapy. Do první jsou ukládány fotony, které byly odraženy od spekulárního materiálu a dopadli na difuzní. Do druhé všechny fotony (včetně těch, které jsou uloženy v mapě kaustik). První fotonové mapě se říká mapa kaustik a druhé globální fotonová mapa (v některých případech se dále dělí na přímou a nepřímou).

5.4 Odhad radiance

Fotonová mapa ukládá přicházející zářivý výkon na povrchy jednotlivých objektů. Při obrazování, které je prováděno pomocí sledování paprsku (viz. kapitola 5.5) je nutné odhadnout přicházející radiance, aby mohla být následně vypočítána odcházející:

$$L_r(x, \vec{\omega}_r) = \int_{\Omega} L_i(x, \vec{\omega}_i) f_r(x, \vec{\omega}_i, \vec{\omega}_r) |\vec{\omega}_i \cdot \vec{N}_x| d\vec{\omega}_i$$

Přicházející radianci $L_i(x, \vec{\omega}_i)$ odhadujeme právě z fotonové mapy, kde je uložen přicházející výkon, vyjádříme tedy radianci pomocí výkonu:

$$L_i(x, \vec{\omega}_i) = \frac{d^2\Phi_i(x, \vec{\omega}_i)}{|\vec{\omega}_i \cdot \vec{N}_x| d\vec{\omega}_i d\vec{\mu}_i}$$

Odraženou radianci můžeme následně vyjádřit jako:

$$L_r(x, \vec{\omega}_r) = \int_{\Omega} \frac{d^2\Phi_i(x, \vec{\omega}_i)}{|\vec{\omega}_i \cdot \vec{N}_x| d\vec{\omega}_i d\vec{\mu}_i} f_r(x, \vec{\omega}_i, \vec{\omega}_r) |\vec{\omega}_i \cdot \vec{N}_x| d\vec{\omega}_i$$

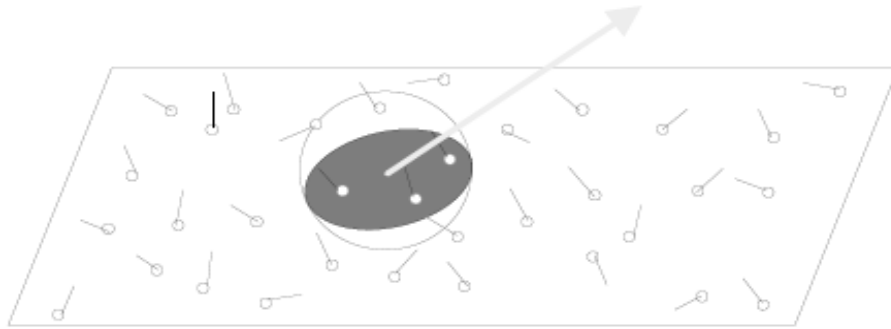
$$L_r(x, \vec{\omega}_r) = \int_{\Omega} \frac{d^2\Phi_i(x, \vec{\omega}_i)}{d\vec{\mu}_i} f_r(x, \vec{\omega}_i, \vec{\omega}_r)$$

Ačkoliv je ve fotonové mapě zářivý výkon uložen, zůstává otázka, jak odhadnout výkon pro konkrétní bod. Odpověď je, pomocí odhadu hustoty.

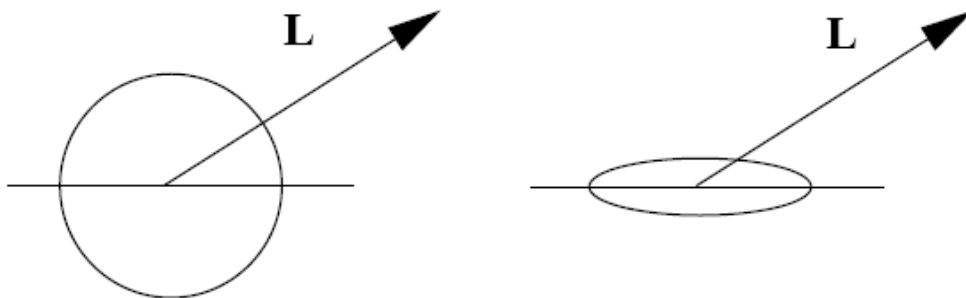
5.4.1 Odhad hustoty

Odhadovat budeme hustotu fotonů na určité ploše, ale zároveň také hustotu světelného toku na určité ploše – irradianci. Okolo daného bodu x nalezneme n fotonů, které od něj mají nejmenší vzdálenost. Každý takový foton nese nějaký výkon Φ_p . Odcházející radianci můžeme aproximovat součtem výkonů všech fotonů, kde každý foton musí brát v úvahu BRDF pro jeho přicházející směr. Celý výsledný výkon je vydělen plochou, na které se odhad hustoty prováděl.

$$L_r(x, \vec{\omega}_r) \approx \frac{1}{\Delta A} \sum_{p=1}^n \Phi_p(x, \vec{\omega}_p) f_r(x, \vec{\omega}_p, \vec{\omega}_r)$$



Obrázek 5.1: Hledání nejbližších fotonů při odhadu hustoty zářivého výkonu. Převzato z (6).



Obrázek 5.2: Rozdíl mezi obálkami – koule, disk (elipsoid). Převzato z (75).

Nejjednodušší obálka, která bude specifikovat oblast, ve které bude odhad hustoty prováděn, je koule (obrázek 5.1). Při odhadu zadáme poloměr koule, což je vlastně maximální vzdálenost fotonu od bodu x , kterou může daný mít, aby byl přijat mezi n blízkých fotonů. I výpočet plochy povrchu je jednoduchý:

$$\Delta A = \pi r^2$$

Samotná přesnost tohoto odhadu velice závisí na množství fotonů, které jsou uloženy ve fotonové mapě. Dále musíme specifikovat maximální počet fotonů, které mohou být použity k odhadu a také poloměr koule, která je použita při vyhledávání nejbližších fotonů. A tady se právě dostáváme k jednomu problému zkreslených metod – většinou mají mnoho parametrů, které musí uživatel správně nastavit, jinak je řešení velice nepřesné (více v sedmé kapitole).

Většinou se v praxi koule vyměňuje za disk (elipsoid), který dává v rozích a na hranách lepší výsledky než koule (obrázek 5.2, více v sedmé kapitole).

Metoda fotonových map je však stále konzistentní, což znamená, že pokud uložíme do fotonové mapy více fotonů, řešení bude přesnější. Důkazem toho je případ, kdy budeme počet fotonů přibližovat k nekonečnu:

$$L_r(x, \vec{\omega}_r) = \lim_{n \rightarrow \infty} \frac{1}{\Delta A} \sum_{p=1}^{|n^\alpha|} \Phi_p(x, \vec{\omega}_p) f_r(x, \vec{\omega}_p, \vec{\omega}_r)$$

Kde:

- $\forall \alpha \in]0,1[$
- α zajišťuje, že počet fotonů v odhadu hustoty bude menší než celkový počet fotonů ve fotonové mapě. To je nutné, protože jak v odhadu, tak ve fotonové mapě je nekonečný počet fotonů a je nutná k ujištění, že tyto nekonečna budou různě velká.

5.4.2 Algoritmus

V této kapitole podrobněji popíšeme jednotlivé body algoritmu 10. Na začátku je potřeba si uvědomit, že odhad hustoty řešení může mít zásadní vliv na kvalitu výsledného řešení, ačkoliv v případě sofistikovanějších vykreslovacích algoritmů není už vliv tak značný.

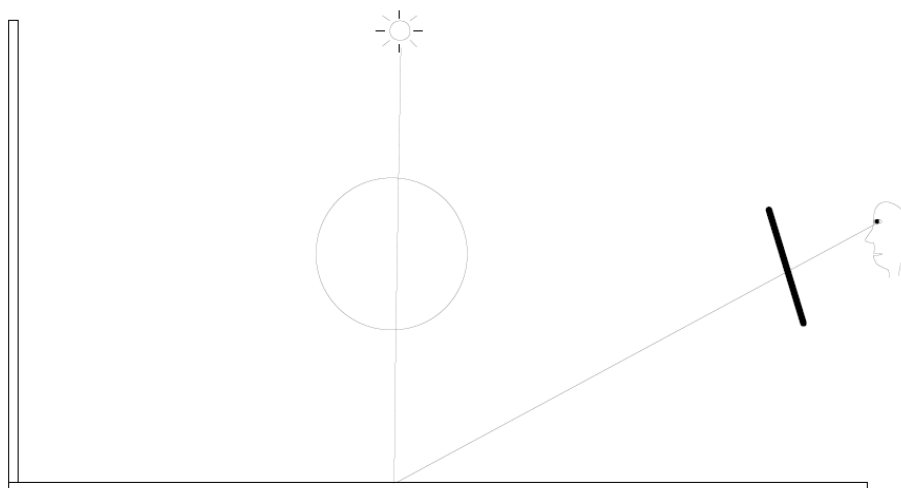
Pro každý foton musíme testovat jeho směr vzhledem k normálovému vektoru, jelikož je pravděpodobné, že v komplexnějších scénách se budou nacházet fotony na obou stranách povrchu. V těchto případech chceme pouze fotony, které přicházejí ve směru normály (tzn. nepřicházejí ze zadní strany povrchu). V algoritmu 10 uvažujeme pouze o nejjednodušším odmítání fotonů. Mezi blízké fotony jsou často započítány i ty, které splňují podmínku týkající se vzdálenosti od bodu x , ale dopadají na jiný povrch. Tento problém nastává především v blízkosti hran a rohů. Pokud namísto koule použijeme disk, zajistíme, že (některé) nesprávné fotony budou vyřazeny a nebudou ovlivňovat řešení. Ke zmenšení vlivu vzdálených fotonů lze použít filtry, o kterých je řeč v kapitole 7.1. Podrobné porovnání jednotlivých obálek, filtrů či řešení problému spojených s vyhledáváním fotonů a odhadem hustoty, lze nalézt v kapitole sedmé.

Jak už bylo odvozeno v předchozí kapitole. Odraženou radianci vypočteme jako celkový přicházející světelný výkon, kde výkon každého fotonu je násoben BRDF pro jeho přicházející směr. Následně je tento výkon vydělen obsahem plochou, na které byl odhadován (v případě koule a elipsoidu je to obsah kruhu).

Algoritmus 10 Odhad hustoty

- 1) Vyhledej n nejbližších fotonů
- 2) Nastav celkový zářivý výkon na nulu
- 3) Pro každý foton:
 - (a) Pokud foton přichází ve směru normály (nepřichází z „druhé strany“)
 - (i) Přičti k celkovému výkonu odražený výkon daného fotonu:
- 4) Vypočti odraženou radianci jako celkový odcházející výkon dělený plochou:

$$L_r = \frac{\Phi}{\pi r^2}$$



Obrázek 5.3: Výpočet přímého osvětlení pomocí sledování paprsku. Převzato z (75).

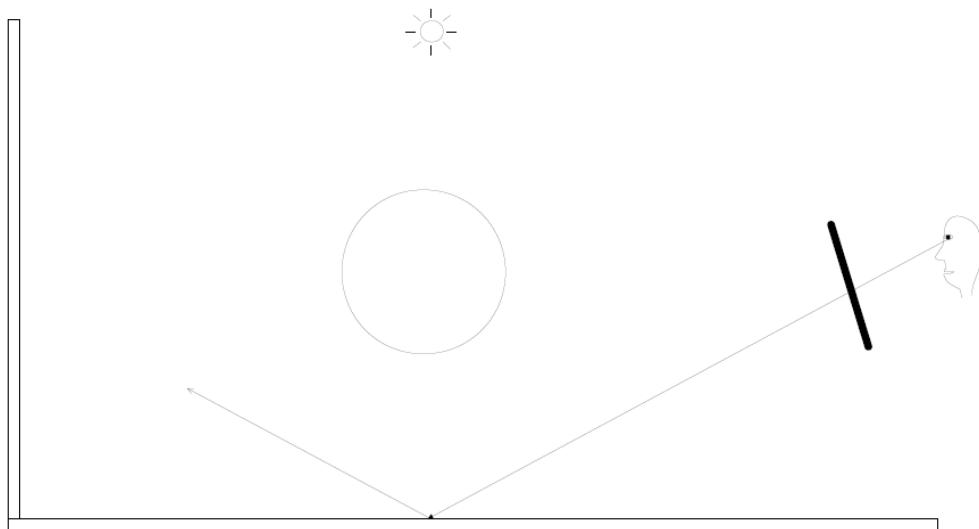
5.5 Vizualizace

Vizualizace, nebo-li také vykreslování je druhou částí algoritmu fotonových map. Všechny algoritmy, které se na vizualizaci podílí, jsou založeny na sledování několika paprsků od pozorovatele. Nejrychlejšího vykreslování lze dosáhnout přímou vizualizací globální fotonové mapy pomocí algoritmus sledování paprsku, kdy přímé osvětlení na difuzních površích nepočítáme pomocí vzorkování světelných zdrojů a stínových paprsků, ale využíváme informace uložené ve fotonové mapě. I tato nejjednodušší vizualizace vykresluje kompletní zobrazovací rovnici. Praktické ukázky lze najít v sedmé kapitole.

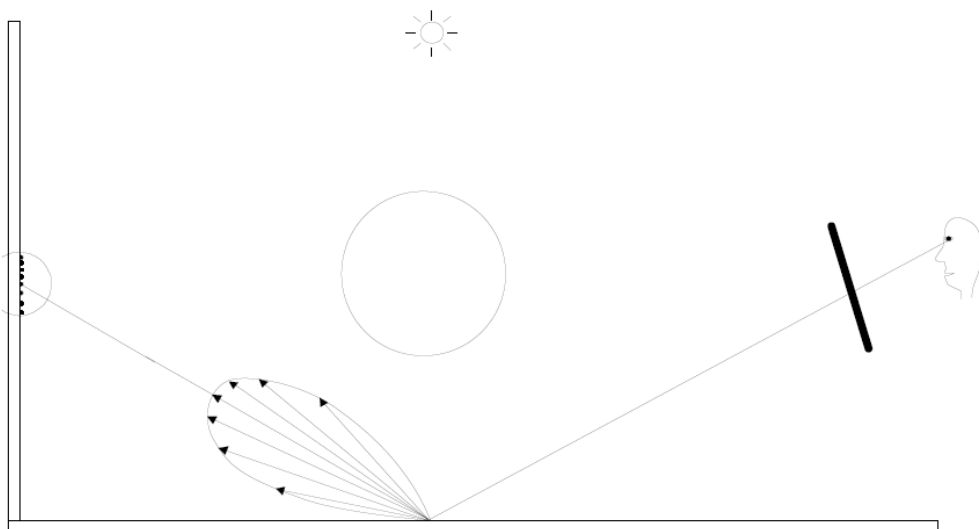
5.5.1 Přímé osvětlení

Přímé osvětlení (*direct illumination – DI*; obrázek 5.3) vzniká přímým dopadem fotonů na povrch nějakého objektu. DI je nejdůležitější složkou osvětlení, která by měla být vypočítána co nejpřesněji, což znamená vzorkováním světelných zdrojů stejně jako u sledování cesty s odhadem příští události (kapitola 4.2.3). V bodě x (průsečík paprsku a tělesa) se vystřelí jeden či více stínových paprsků směrem ke světelným zdrojům, které zjistí, zda je v cestě nějaká překážka a vypočtou osvětlení bodů. V případě bodového světleného zdroje stačí pouze jeden paprsek, pokud ale pracujeme s velkými plošnými zdroji, je většinou nutné použít mnoho paprsků, jinak dochází ke vzniku šumu, který je patrný především v oblasti polostíňů.

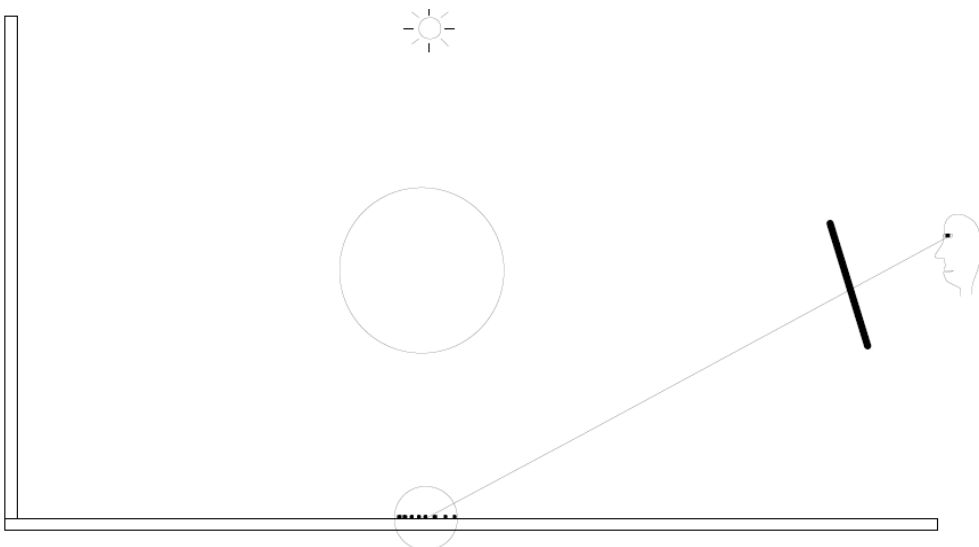
Pokud je ve scéně přítomno několik velkých plošných zdrojů, může být výpočet přímého osvětlení velice náročný, především díky stínům. V této situaci mohou pomoci takzvané stínové foton (shadow photons), které podrobně rozebírá kapitola 6.6. V prvním průchodu, kdy se tvoří fotonové mapy, přibude mapa stínových fotonů. Ze všech světelných zdrojů budou vystřelovány stínové fotony, které se na rozdíl od běžných fotonů nebudou odrážet, pouze protnou několik povrchů. V každém (kromě prvního) průsečíku bude foton uložen do stínové fotonové mapy, která bude udržovat „negativní“ energii scény. Při vykreslování bude v každém bodě vypočítávána „negativní“ energie. Stínové fotony lze využít různými způsoby (více v kapitole 6.6).



Obrázek 5.4: Výpočet zrcadlového (spekulární) osvětlení pomocí sledování paprsku. Převzato z (75).



Obrázek 5.5: Výpočet nepřímého osvětlení pomocí finálního sbírání. Převzato z (75).



Obrázek 5.6: Výpočet kaustik. Převzato z (75).

Přímé osvětlení může být odhadováno také pomocí globální fotonové mapy. V tomto případě je výsledné řešení „hladší“, není žádný šum, nicméně je potřeba vystřelit mnoho fotonů a nastavit správně parametry při odhadu hustoty, aby se . I přesto mohou vznikat problémy, spojené především s „falešnými“ fotony, které nepatří do odhadu (viz. kapitola 5.4.1). Další informace v kapitole sedmé.

5.5.2 Zrcadlové

Na spekulárních (zrcadlových) či lesklých (*glossy*) površích nejsou ukládány fotony do fotonové mapy, protože je přicházející světlo omezeno pouze na jeden (v případě ideálního zrcadlového odrazu) nebo několik málo směrů (v případě lesklého odrazu). Opět je zde použito klasické Monte Carlo sledování paprsku se vzorkováním podle důležitosti s využitím BRDF jako PDF. Ve většině případu stačí pár paprsků na lesklé materiály a jeden na spekulární. Ve speciálních případech je pro lesklé materiály výhodnější použít globální fotonovou mapu.

5.5.3 Nepřímé osvětlení

Nepřímé osvětlení (*indirect illumination*) je výpočetně nejnáročnější část osvětlení (alespoň v naprosto drtivém počtu scén). Běžné starší metody jako sledování cesty samozřejmě nepřímé osvětlení také počítají, ale nejsou příliš efektivní a potřebují mnoho paprsků, aby byl výsledný obraz bez šumu. Především to platí pro scény se spekulárními materiály, které vytváří kaustiky, protože právě kaustiky jsou hlavním zdrojem šumu v nestranných Monte Carlo metodách založených na sledování paprsku.

Fotonové mapy mají dvě datové struktury (globální mapu fotonů a mapu kaustik), takže jsou kaustiky zvlášť a nepřímé osvětlení je tudíž velmi „hladké“. Navíc máme díky globální fotonové mapě informace o přicházejícím zářivém výkonu. Při vzorkování podle BRDF máme tedy dodatečné informace o integrantu (viz. kapitola 4.2.2 o vzorkování podle důležitosti), takže můžeme podle nich vzorkovat (viz. kapitola 6.4).

Samotný výpočet nepřímého osvětlení lze provést několika způsoby. Nejpoužívanější způsob je takzvané finální sbírání (*final gathering*). Tento algoritmus je podobný distribuovanému sledování paprsku, které je omezeno pouze na jeden stupeň rekurze. Na obrázku 5.5 je znázorněn princip algoritmu. Na prvním difuzním průsečíku je vystřeleno několika paprsků (řádově stovky až tisíce), které následně protnou scénu v nějakém bodě, kde vzorkování končí. V tomto konečném bodě je k výpočtu osvětlení použita globální fotonová mapa. Existují velice významná urychlení této části výpočtu, která jsou popsána v následující kapitole.

Nepřímé osvětlení může být také počítáno pomocí sledování cesty či distribuovaného sledování paprsku. S použitím fotonových map lze tyto algoritmy urychlit a zlepšit i kvalitu řešení.

5.5.4 Kaustiky

Vykreslování kaustik je velice jednoduché (obrázek 5.6). Ačkoliv jsou kaustiky „ukryty“ i v globální fotonové mapě, protože ta ukládá všechny fotony, jak již bylo zmíněno, přímo osvětlení většinou není počítáno pomocí globální fotonové mapy. Je tedy nutné kaustiky přidat zvlášť. Při prvním protnutí difuzního materiálu paprsku se vypočte odhad z mapy kaustik, který určí radianci, která byla odražena od lesklých materiálů. Radiance kaustik je přičtena k celkové radianci.

Kapitola 6

Rozšíření fotonových map

V průběhu let bylo vytvořeno mnoho rozšíření základního algoritmu fotonových map, které přispěly ke zvýšení kvality výsledného obrazu či urychlily výpočet. Zde budou uvedeny především ty nejdůležitější lety prověřené algoritmy. Několik nejnovějších přístupů je pak nastíněno v kapitole 6.7. Více o rozšířeních fotonových map lze nalézt v (6), (43).

6.1 Filtry

Při běžném odhadu hustoty je každý foton, který byl vybrán jako dostatečně blízký, stejně důležitý jako jiný, ačkoliv některé fotony jsou téměř u bodu x , zatímco další mohou být vzdáleny poměrně výrazně, protože jsme mohli vystřelit málo fotonů, a tudíž jsme museli hledat ve velké oblasti. Pokud budeme odhadovat hustotu ve dvou bodech, které jsou od sebe pouze nepatrně vzdáleny, výsledky budou téměř totožné, protože většina nalezených fotonů bude společná pro oba dva body. Tento fakt vede k rozmazaným hranám jinak poměrně ostrých přechodů. Při výpočtu nepřímého osvětlení to může být i výhodou, jelikož řešení poté vypadá spojitější, ale například pro kaustiky chceme realistické ostré přechody, protože rozmazané kaustiky ve většině případů nevypadají příliš realisticky.

Realisticky vypadající kaustiky lze zajistit použitím filtrů, ačkoliv samotný filtr dobře vyhlížející kaustiky nezajistí, je potřeba mít i odpovídající počet fotonů v mapě kaustik. Princip filtrování je jednoduchý, vypočítat každému fotonu váhový koeficient podle jeho vzdálenosti od bodu x . Tento váhový koeficient bude mít poté vliv na jeho důležitost pro odhad hustoty irradiance v bodě x .

6.1.1 Kuželový filtr

Kuželový filtr má váhové koeficienty rovny:

$$w_p = 1 - \frac{d_p}{kr}$$

Kde:

- d_p = vzdálenost fotonu od bodu x

- k = konstanta charakterizující filtr (měla by být větší nebo rovna jedné)
- r = maximální možná vzdálenost

Ještě je potřeba udělat malou změnu v odhadu radiance. V případě, že je obalové těleso koule či disk, odhad vypadá následovně:

$$L_r(x, \vec{\omega}_r) \approx \frac{1}{\left(1 - \frac{2}{3}k\right) \pi r^2} \sum_{p=1}^n \Phi_p(x, \vec{\omega}_p) f_r(x, \vec{\omega}_p, \vec{\omega}_r) w_p$$

6.1.2 Gaussův filtr

Váhové koeficienty Gaussova filtru jsou:

$$w_p = \alpha \left[1 - \frac{1 - e^{-\beta \frac{d_p^2}{2r^2}}}{1 - e^{-\beta}} \right]$$

Kde:

- d_p = vzdálenost fotonu od bodu x
- $\alpha = 0,918$
- $\beta = 1,953$

Samotný odhad radiance se nezmění, pouze přibudou váhové koeficienty w_p :

$$L_r(x, \vec{\omega}_r) \approx \frac{1}{\pi r^2} \sum_{p=1}^n \Phi_p(x, \vec{\omega}_p) f_r(x, \vec{\omega}_p, \vec{\omega}_r) w_p$$

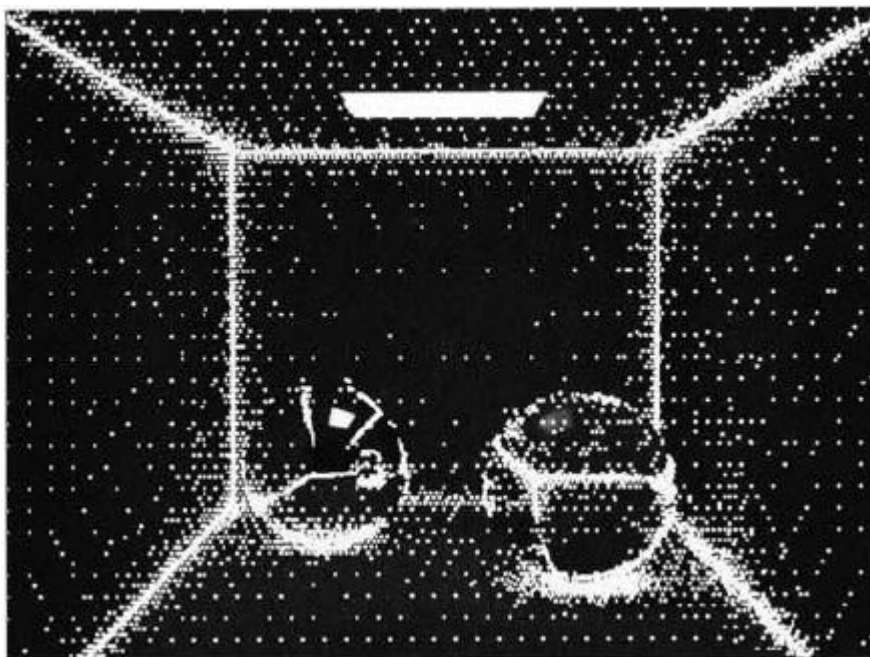
6.2 Adaptivní vzorkování

V kapitole 3.2 byly obecně probrány techniky na snížení rozptylu výsledného řešení. Adaptivní vzorkování byla jednou ze čtyř skupin a jako jediná nebyla podrobněji probrána, protože zavede do řešení určité zkreslení a ve třetí kapitole jsme se soustředili především na nestranné Monte Carlo metody.

Jak již bylo řečeno, adaptivní vzorkování je založeno na adaptivním měnění hustoty vzorkování podle nějakých dat, které algoritmus sbírá během vzorkování.

Při vykreslování využíváme adaptivní vzorkování na úrovni pixelů. Většina technik při vzorkování pixelů posílá skrze několik paprsků (někdy několik stovek až tisíc). Konkrétní počet je většinou nastaven ještě před spuštěním vykreslování, ale s použitím adaptivního vzorkování můžeme zajistit více paprsků pro „problémové“ pixely a redukovat počet paprsků pro běžné pixely.

Základní principem je vzorkovat pixel, dokud není řešení dostatečně přesné. Musíme tedy vypočítat rozptyl řešení pomocí:



Obrázek 6.1: Irradiance caching. Bílé tečky znázorňují body, v kterých bylo nutné počítat nepřímé osvětlení, zatímco černé tečky body, v kterých bylo nepřímé osvětlení dopočítáno interpolací. Převzato z (6).

$$s^2 = \frac{1}{N-1} \left[\frac{1}{N} \sum_{i=1}^N L_i^2 - \left(\frac{1}{N} \sum_{i=1}^N L_i \right)^2 \right]$$

Arvo a Kirk (44) zjistili, že použít rozptyl jako ukončovací kritérium je účinné a dává dobré výsledky, ale uvádí do řešení zkreslení. Proto navrhli způsob, jakým je možné se vyhnout zkreslení. Použili, namísto neustálého počítání rozptylu, takzvaný „úvodní“ vzorek, jenž odhadne množství paprsků, které by měly být použity. Samotný „úvodní“ vzorek by neměl být do řešení započítán. Tento způsob je opravdu nestranný, ale už není tak účinný.

Jensen a Tamstorf (45) následně zjistili, že množství nestrannosti je ve většině případů opravdu malé.

6.3 Irradiance Caching

Irradiance caching byl uveden Wardem (46) v roce 1988. Tato technika využívá faktu, že nepřímé osvětlení je „hladké“ (jeho hodnota se mění velice pomalu), proto je možné hodnoty irradiance ukládat a následně interpolovat nové hodnoty, což přijde ve většině scén velice vhod, protože nepřímé osvětlení je výpočetně nejnáročnější složkou světla. Důležité je, že tato technika je použitelná pouze pro ideální difuzní materiál (Lambertův, kapitola A.1).

6.3.1 Původní algoritmus

Přicházející irradianci můžeme získat pomocí jednoduchého vzorkování hemisféry podle BRDF a kosinu úhlu θ :

$$E(x) = \frac{\pi}{N} \sum_{i=1}^n L_i(\theta_i, \phi_i) \quad (6.1)$$

Kde:

- N = počet vzorků
- $\theta_i = \sin^{-1} \sqrt{\xi_1}$ (úhel mezi normálou a směrem)
- $\phi_i = 2\pi\xi_2$ (azimut)
- ξ_1 a ξ_2 = náhodná čísla v intervalu $[0,1]$

V literatuře se většinou při definici irradiance cachingu nepoužívá jednoduché náhodné vzorkování, ale raději vzorkování po částech:

$$E(x) = \frac{\pi}{MN} \sum_{j=1}^M \sum_{i=1}^N L_{i,j}(\theta_j, \phi_i) \quad (6.2)$$

Kde:

- M, N = části hemisféry
- $M*N$ = počet vzorků
- $\theta_j = \sin^{-1} \sqrt{\frac{j-\xi_1}{M}}$
- $\phi_i = 2\pi \frac{i-\xi_2}{N}$

Řešení rovnice 6.1 (respektive 6.2) vyžaduje většinou 200-5000 paprsků odhadujících odcházející radiance z mnoha směrů. I kdybychom uvažovali pouze o případě 200 paprsků, výpočet bude časově velice náročný. Jelikož je ale nepřímé osvětlení spojité, lze využít právě irradiance caching.

Pokud chceme získat nepřímé osvětlení v bodě x , musíme se rozhodnout, zda ho budeme počítat nebo použijeme již uložené irradiance a interpolujeme z nich irradianci v bodě x . Toto rozhodnutí záleží především na již uložených hodnotách irradiance. Jednoduše řečeno, musíme zjistit, zda máme uloženy takové hodnoty, které se hodí na interpolaci irradiance pro daný bod x . Vypočteme váhu pro daný vzorek irradiance, jenž je uložen v bodě x_i :

$$w_i(x, \vec{N}) = \frac{1}{\epsilon_i(x, \vec{N}_i)}$$

Tato váha symbolizuje, jak vhodný je daný vzorek irradiance pro použití při interpolaci. Člen $\epsilon_i(x, \vec{N}_i)$ reprezentuje množství změny v irradianci mezi body x a x_i . Čím větší nepřesnost, tím menší váha w_i a tedy i šance, že bude daný vzorek použit. $\epsilon_i(x, \vec{N}_i)$ vypočteme jako:

$$\epsilon_i(x, \vec{N}_i) = \frac{\|x_i - x\|}{R_0} + \sqrt{1 - \vec{N} \cdot \vec{N}_i} \quad (6.3)$$

Kde:

- \vec{N} = normála v bodě x
- \vec{N}_i = normála v bodě x_i
- R_0 = harmonická vzdálenost protnutých povrchů

Váhový koeficient můžeme rozepsat jako:

$$w_i(x, \vec{N}) = \frac{1}{\frac{\|x_i - x\|}{R_0} + \sqrt{1 - \vec{N} \cdot \vec{N}_i}} \quad (6.4)$$

Otázkou zůstává, jak poznat, že je váha vzorku dostačující a může být použit k interpolaci. Ward navrhl jednoduché řešení. Uživatel sám zvolí hodnotu parametru a , který reprezentuje maximální tolerovanou chybu:

$$w_i(x, \vec{N}) > \frac{1}{a}$$

Jakmile je tato podmínka splněna, irradiance v bodě x_i může být použita k interpolaci irradiance v bodě x . Po sehnání všech uspokojujících irradiancí můžeme irradianci v bodě x vypočítat jako:

$$E(x, \vec{N}) \approx \frac{\sum_{i, w_i > \frac{1}{a}} w_i(x, \vec{N}) E_i(x_i)}{\sum_{i, w_i > \frac{1}{a}} w_i(x, \vec{N})} \quad (6.5)$$

Pokud jsme nenalezli ani jednu uloženou irradianci, která by mohla být použita k interpolaci, budeme muset irradiance vypočítat běžným způsobem (následně ji samozřejmě uložíme).

V praxi se snažíme o co největší urychlení, takže musíme provést ještě pár úprav, aby byl algoritmus praktický. Počítání váhových koeficientů pořád dokola je poměrně časově náročné, proto je výhodnější vypočítat pro každý vzorek jeho maximální vzdálenost, do které může být použit. Tento výpočet vyžaduje ignorování normály (neznáme dopředu normálu bodu, ve kterém budeme odhadovat irradianci, nehledě na fakt, že budeme odhadovat desítky bodů a předpočítáváme pouze jednu hodnotu) a způsobuje tedy, že řešení už nebude natolik přesné. Předpočítání maximální použitelné vzdálenosti si lze také představit jako obalení bodu koulí o poloměru, který je roven dané maximální vzdálenosti. Ještě větší účinnosti lze dosáhnout kombinací s oktanovým stromem.

Je třeba poznamenat, že irradiance caching uvádí do řešení zkreslení. Pokud ho tedy použijeme například v algoritmu sledování cesty, může výrazně zlepšit kvalitu výsledného řešení (především v případě, kdy jsou materiály ve scéně převážně difuzní) a také zastavit náhodnou procházku, ale interpolací (a „násilné“ přerušování náhodné procházky) způsobí, že algoritmus sledování cesty už nebude dále nestranný.

6.3.2 Gradienty Irradiance

Ward (47) původní algoritmus ještě rozšířil o gradienty irradiance. Ve svém paperu odvodil rovnice gradientů. Gradient pozice je:

$$\nabla_p E = \sum_{i=1}^N \left\{ \vec{T}_i \frac{2\pi}{n} \sum_{j=2}^M \frac{\sin \theta_{j-} \cos^2 \theta_{j-}}{\min(d_{j,i}, d_{j-1,i})} (L_{j,i} - L_{j-1,i}) + \vec{T}_{i-} \sum_{j=1}^M \frac{\sin \theta_{j+} - \cos^2 \theta_{j-}}{\min(d_{j,i}, d_{j-1,i})} (L_{j,i}, L_{j,i-1}) \right\}$$

Kde:

- \vec{T}_i = vektor kolmý k normále ve směru ϕ_i
- \vec{T}_{i-} = vektor kolmý k normále ve směru ϕ_{i-}
- θ_{j-} = je roven $\sin^{-1} \sqrt{j/m}$
- θ_{j+} = je roven $\sin^{-1} \sqrt{(j+1)/m}$
- $d_{j,i}$ = vzdálenost k objektu, který je viděn ve směru j, i
- $L_{j,i}$ = radiance přicházející ze směru j, i

Gradient rotace:

$$\nabla_r E = \frac{\pi}{MN} \sum_{i=1}^N \left\{ \vec{T}_i \sum_{j=1}^M -L_{j,i} \tan \theta_j \right\}$$

Kde:

- \vec{T}_i = vektor kolmý k normále ve směru $\phi_i + \pi/2$

Pokud rozšíříme rovnici 6.4 o tyto gradienty, dostaneme:

$$E(x, \vec{N}) \approx \frac{\sum_{i, w_i > 1/a} w_i(x, \vec{N}) \{E_i(x_i) + (x - x_i) \nabla_p E_i + (\vec{N}_t \times \vec{N}) \nabla_r E_i\}}{\sum_{i, w_i > 1/a} w_i(x, \vec{N})} \quad (6.6)$$

Použití gradientů zajistí daleko realističtější výsledky.

6.3.3 Irradiance caching a fotonové mapy

Fotonové mapy a irradiance caching jsou výbornou kombinací, protože nepřímé osvětlení je při vykreslování fotonových map zdaleka nejpomalejší. Irradiance caching může tento výpočet značně urychlit a také zajistit, aby výsledné řešení vypadalo hladší. Navíc, fotonové mapy oddělují kaustiky od přímého a nepřímého osvětlení, což je pro irradiance caching výhodné, protože právě kaustiky jsou hlavní příčinou vzniku šumu. Dále lze při vzorkování paprsků použít globální fotonovou mapu a tím získat výhodu.

6.4 Vzorkování podle fotonové mapy

V kapitolách 4.2.2 a 4.3.2 bylo vysvětleno základní vzorkování podle důležitosti a byl také nastíněn problém při vzorkování v aplikacích globálního osvětlení – nelze vzorkovat neznámou funkci radiance, která má největší vliv na to, jaký vzorek je důležitý a jaký ne. V případě, že máme fotonovou mapu, která ukládá veškerou irradianci scény, lze ji snadno využít ke konstrukci funkce hustoty pravděpodobnosti (PDF), která je daleko účinnější než PDF 4.7.

$$p(x, \vec{\omega}_x) = f_r(x, \vec{\omega}_y, \vec{\omega}_x) L_i(x, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x|$$

$$p(x, \vec{\omega}_x) = \frac{f_r(x, \vec{\omega}_y, \vec{\omega}_x) d^2 \Phi_i(x, \vec{\omega}_x) |\vec{\omega}_y \cdot \vec{N}_x|}{|\vec{\omega}_y \cdot \vec{N}_x| d\vec{\omega}_x d\vec{\mu}_x}$$

$$p(x, \vec{\omega}_x) = \frac{f_r(x, \vec{\omega}_y, \vec{\omega}_x) d^2 \Phi_i(x, \vec{\omega}_x)}{d\vec{\omega}_x d\vec{\mu}_x}$$

V praxi je postup vzorkování podle přicházejícího výkonu a BRDF následující:

1) Rozdělení domény do buněk

Při běžném vzorkování podle důležitosti, kdy nepoužíváme dodatečnou informaci v podobě výkonu, se snažíme uniformně vzorkovat jednotkový čtverec, která následně namapujeme na hemisféru pomocí mapovací funkce.

Pro přidání informace o výkonu rozdělíme celou doménu (čtverec) do několika buněk, kde jedna buňka po namapování reprezentuje část hemisféry (sadu směrů).

2) Vypočítání výkonu v jednotlivých buňkách

Pro každou buňku vypočítáme přicházející výkon pomocí fotonové mapy. Využijeme funkci, kterou používá odhad hustoty, na vyhledání několika nejbližších fotonů. Každý foton by měl mít uložen směr. Na zjištění buňky, do které daný foton patří, použijeme mapování hemisféry na čtverec (inverzní mapování k mapování, jenž používáme při obyčejném vzorkování podle důležitosti).

3) Vzorkování

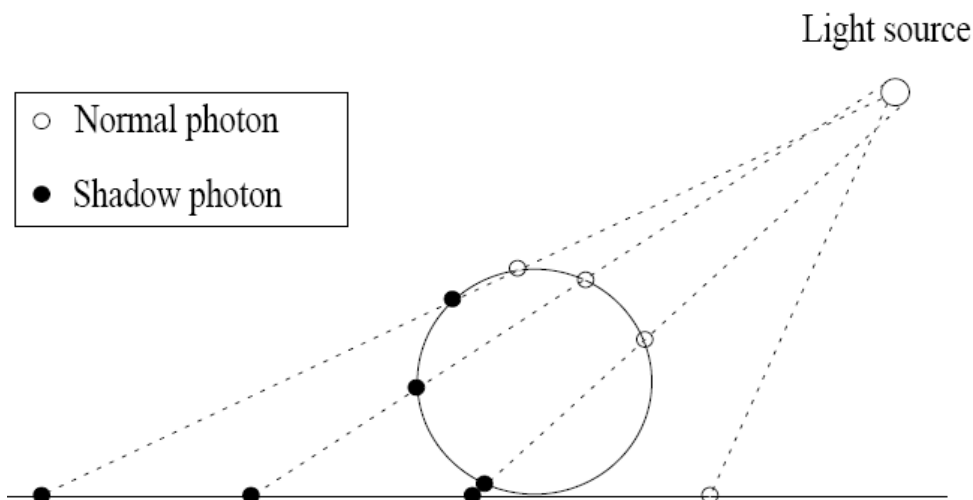
Vzorkování probíhá trochu odlišně než u „klasického“ vzorkování - nejdříve musíme zjistit, v jaké buňce budeme vzorkovat. Normalizovaný výkon v dané buňce slouží jako diskretní PDF. Celkově musí tedy být vygenerována tři náhodná čísla. První rozhodne, v jaké buňce se bude vzorkovat, a další dvě čísla jsou použita k vzorkování dané buňky.

Výrazného urychlení můžeme dosáhnout pokud předpočítáme irradiance.

6.5 Předpočítaná irradiance

O nápadu předpočítat irradianci se poprvé zmínil Christensen ve svém paperu (48). Všiml si, že každý foton je během vykreslování mnohokrát použit a přitom při počítání irradiance počítáme u každého fotonu pořád to samé. Pokud bychom tedy předpočítali irradianci ve scéně ještě předtím než začneme vykreslovat, může to vést k urychlení celého procesu.

Základním problémem je, v jakých místech scény irradianci vypočítat. Nabízí se rovnoměrná distribuce ve scéně, ta ovšem neuvažuje o faktu, že některé části scény jsou důležitější než jiné, protože přijímají více světla (zářivého výkonu). Proto využijeme globální fotonovou mapu nejen na získání přicházejícího výkonu, ale také využijeme pozice fotonů jako místa, kde vypočítáme irradianci. Tu můžeme uložit přímo do fotonové mapy, stejně jako normálový vektor v daném bodě. Velikost fotonové mapy tedy vzroste, ale ne nijak výrazně, pokud použijeme kompresi irradiance navrženou Wardem. Christensem také otestoval, že není nutné počítat irradianci v pozici každého fotonu, stačí pouze u některých fotonů (otestováno bylo použití každého čtvrtého fotonu). Naopak je právě většinou nutné snížit počet míst, kde se bude



Obrázek 6.2: Ukládání normálních (bílé) a stínových fotonů (černé). Převzato z (6).

irradiance počítat, pokud chceme zajistit výraznější urychlení. Pokud není počet míst odhadu výrazně menší, většinou by neměl být znát rozdíl.

Vykreslování probíhá podobně jako v klasické metodě fotonových map, avšak namísto zářivého výkonu používáme irradianci. Normálu používáme na odmítnutí odhadů, které mají příliš velkou odchylku od našeho směru, což se hodí, zejména pokud odhadujeme hustotu irradiance v blízkosti nějaké hrany.

Samotné urychlení závisí především na počtu paprsků, které jsou použity na výpočet nepřímého osvětlení. Jelikož je těchto paprsků většinou celkově daleko více než počet bodů, v kterých je odhadována irradiance, je urychlení několikanásobné (Christensen uvádí, že typické urychlení je pěti až sedminásobné).

6.6 Stínové fotony

V roce 1995 představili Jensen a Christensen (49) rozšíření fotonových map – stínové fotony (*shadow photons*). Ve většině scén je nutné v každém bodě, v kterém počítáme osvětlení, vystřelit několik stínových paprsků směrem ke světelným zdrojům, protože se ve scéně vyskytují vržené stíny. Pro bodový světelný zdroj postačí pouze jediný stínový paprsek, ale pro velké plošné zdroje je nutné vystřelit často až několik stovek stínových paprsků, což je značně výpočetně náročné. Navíc, v komplexních scénách může být několik světelných zdrojů, z nich každý bude vyžadovat mnoho stínových paprsků pro většinu bodů scény. Toto rozšíření se snaží využít faktu, že některé části scény by nemusely vůči některým světelným zdrojům vrhat žádné stínové paprsky, protože vidí celý světelný zdroj. Naopak některé by nemusely vrhat stínové paprsky k určitému světelnému zdroji, protože je světelný zdroj celý v zákrytu.

Při použití stínových fotonů budeme potřebovat ještě jednu mapu fotonů – takzvanou stínovou fotonovou mapu. Do té ukládáme stínové fotony (obrázek 6.2). V první fázi metody fotonových map sledujeme fotony ve scéně. První průsečík fotonu obsahuje kladnou energii a je uložen jako normální foton do globální fotonové mapy, ostatní průsečíky obsahují „zápornou energii“ a jsou

uloženy do stínové fotonové mapy. Do ní ukládáme fotony pouze, pokud normála jejich průsečíku směřuje směrem ke světelnému zdroji (nemusí přímo protínat světelný zdroj, stačí, když bude odchylka mezi normálou a směrovým vektorem přicházejícího fotonu dostatečně velká), z něhož byl foton vystřelen. Na obrázku 6.2 jsou 4 fotony. V průsečíkách reprezentovaných nevyplněnou kružnicí jsou uloženy normální fotony, zatímco v těch, jenž jsou reprezentovány vyplněnou kružnicí, mohou být uloženy stínové fotony. Ovšem pouze stínové fotony na spodní ploše budou uloženy, protože ty nacházející se na kouli mají normálu, jenž nesměruje ke světelnému zdroji (normála má příliš malou odchylku od přicházejícího směru). Každý stínový foton musí mít také uloženo číslo světelného zdroje, z kterého byl vystřelen, protože v případě scény s více světelnými zdroji musíme vědět, který světelný zdroj je zastíněn a který je vidět.

Při vykreslování je možné využít stínové fotony několika způsoby. Pokud odhadneme počet normální fotonů z globální mapy fotonů a počet stínových fotonů ze stínové fotonové mapy, můžeme porovnat jejich počty. Jsou tři možnosti, jak může porovnání dopadnout:

- Poblíž bodu x jsou pouze normální fotony, proto není nutné vrhat stínové paprsky, jelikož jsou všechny světelné zdroje viditelné.
- Poblíž bodu x jsou pouze stínové fotony. Opět není potřeba použít žádné stínové paprsky, protože jsou všechny světelné zdroje v zákrytu a bod x se nachází v úplném stínu.
- Můžeme najít oba druhy fotonů, bod x se tedy nachází v polostínu.

V třetím případě jsme právě našli oblast, kde je výhodné použít stínové paprsky k výpočtu viditelnosti. Další možností je využít k výpočtu právě stínové fotony:

$$viditelnost = \frac{n_s}{n_s + n_n}$$

Kde:

- n_s = počet stínových fotonů
- n_n = počet normálních fotonů

Aby tato aproximace dávala dobré výsledky, je nutné použít více stínových fotonů než v prvním případě.

6.7 Ostatní

V této kapitole jsou zmíněny některé další, neméně podstatné, rozšíření. Počínaje vizuální důležitostí, která je popsána pouze ve stručnosti, protože jsme se v práci nezabývali vykreslováním komplexních scén, jelikož jsme neměli k dispozici žádné takové scény a hlavně nebyl čas vykreslovat natolik komplexní scény. Další zajímavou technikou jsou reverzní fotonové mapy, které nabízí poměrně zajímavý a účinný přístup. Progresivní fotonové mapy vycházející z reverzních fotonových map a přinášejí řadu vylepšení, která jsou užitečná zejména ve scénách, kde jsou dominantní kaustiky, nebo kde je potřeba uložit velké množství fotonů, které by se v běžném algoritmu nevešly do paměti.

Vizuální důležitost

Vizuální důležitost je rozšíření určené pro vykreslování komplexních scén, kde pozorovatel vidí pouze malou část scény. V tomto případě je zbytečné ukládat fotony v celé scéně. Daleko výhodnější by bylo určit „důležité“ oblasti scény (z hlediska pozorovatele) a ukládat fotony především tam.

V praxi znamená použití vizuální důležitosti přidání jednoho kroku.

1. Nejdříve vytvoříme mapu vizuální důležitosti vystřelování importonů od pozorovatele. Importony jsou následně sledovány ve scéně, ale počet odrazů je omezen na jeden (od difuzního materiálu).
2. Poté jsou sledovány fotony stejně jako dříve. Při ukládání daného fotonu se kontroluje, zda se nachází v oblasti, která je dostatečně důležitá na to, aby mělo smysl tam daná foton ukládat. Důležité oblasti se vyznačují dostatečnou hustotou importonů. Pokud není oblast „důležitá“, foton není uložen. Tento přístup uvádí do řešení zkreslení, ale dokáže několikanásobně urychlit výpočet a navíc, zkreslení se bude nacházet především v oblastech, které nejsou pro výsledné řešení podstatné.
3. Vykreslování probíhá stejně jako bez použití vizuální důležitosti.

Reverzní fotonové mapy

Tento přístup představeným Havranem (50) prohazuje pořadí, v jakém vykonáváme kroky výpočtu. Nejdříve tedy provede sledování paprsku (zde takzvaných reverzních fotonů) a uloží průsečíky s paprsky do datové struktury (KD-strom) – reverzní fotonové mapy, která opět slouží k efektivnímu vyhledávání.

V druhé průchodu probíhá sledování fotonů podobné jako v případě normálních fotonových map. Jakmile je fotonová mapa dostatečně zaplněna, jsou vypočteny příspěvky jednotlivých fotonů k reverzním fotonům. Pro získání barvy pixelu použijeme reverzní fotony vystřelené skrze daný pixel.

Tento přístup umožňuje výrazně urychlit fotonové mapy v případě, že je použito mnoho paprsků (reverzních fotonů), protože jsou, na rozdíl od klasického přístupu, také uloženy v datové struktuře, která urychlí jejich vyhledávání (logaritmická složitosti).

Progresivní fotonové mapy

Metoda teprve nedávno uvedená Jensenem (51), která částečně navazuje na reverzní fotonové mapy. Opět používá sledování paprsku jako první krok a sledování fotonů jako druhý. Na rozdíl od reverzních fotonových map, neprobíhá pouze sledování fotonů pouze jednou, ale n-krát. Po každém sledování fotonů jsou upraveny reverzní fotony, mapa je vymazána a následně probíhá další krok sledování fotonů. Tento přístup zajišťuje, že jsou na ukládání dat kladeny poměrně malé nároky.

Další vylepšení kvality zajišťuje progresivní odhad hustoty irradiance, jenž po každém průběhu sledování fotonů, zmenšuje poloměr koule (disku).

Výborné výsledky dával tento přístup především ve scénách, kde bylo nutné vykreslit velice komplexní kaustiky pomocí sledování několika desítek (až stovek) milionů fotonů.

Kapitola 7

Výsledky testů

Celá tato kapitola prakticky dokazuje některé věci zmíněné v předchozích sekcích. V úvodu je řešen odhad radiance a jeho vliv na výsledné řešení, případné problémy při vyhledávání a odmítání fotonů, či použití filtrů pro přímé osvětlení. Druhá část je věnována výpočtům přímého a nepřímého osvětlení a kaustik. Nejdříve jsou vždy popsány metody, kterou budou k řešení daného osvětlení použity. Ty jsou následně porovnány a výsledky jsou rozebrány v následné diskusi.

Generování čísel ξ probíhalo zcela náhodně. Nebyly tedy použity žádné deterministické techniky jako Kvazi-Monte Carlo metody, které by zajistili jejich rovnoměrnější distribuci. Vzorkování po částech rozdělení domény do několika sub-domén nemohlo být použito, protože není dopředu znám počet vystřelených fotonů.

Veškeré výpočty byly prováděny na AMD Athlon 4850e s 3,25GB RAM v OS Windows XP.

Jako testovací scény byly použity scény založené na oblíbené Cornell box. Jednotlivé varianty jsou stručně popsány v kapitole 7.2.2. Některé věci však zůstávají stejné pro všechny varianty. Rozměry byly trochu zmenšeny oproti originálu. Světelný zdroj má rozměry originální – 130×105 . Ve výsledných obrázcích však nejsou světelné zdroje zobrazeny.

Všechny vykreslené obrázky jsou ve vyšším rozlišení uloženy na CD nosiči, který je přibalen k práci. Na CD jsou také k dispozici obrázky, které se do práce nevešly.

7.1 Odhad radiance

Odhad radiance je podstatná část fotonových map (kapitola 5.4), která má velký vliv na kvalitu výsledného řešení, jak bude ostatně dokázáno v dalších kapitolách. Pro odhad radiance jsou klíčové dvě věci. Obálka, která vlastně rozhoduje, jaké fotony budou vyloučeny a jaké přijaty. Dále také použití filtrů (kapitola 6.1), které jsou schopny vyhladit výsledné řešení. V této kapitole budeme filtry používat pouze na přímé osvětlení. Jejich použití na zlepšení vzhledu kaustik je testováno v kapitole 7.4

7.1.1 Obálky

V této kapitole porovnáme dvě nejpoužívanější obálky – kouli a disk. Na obrázku 7.1 vidíme čtyři obrázky. Levé dva používají kouli popsanou dříve. V blízkosti hran můžeme vidět výrazné zesvětlení, které se v případě většího počtu fotonů a zmenšení poloměru koule (levý spodní obrázek) stane ještě patrnější. Uprostřed stěny se následně zobrazí jakýsi čtverec, který je ze stran a zespoda ohraničen světlými pruhy. Tento jev vzniká, protože koule přijímá i fotony, které by neměla – ty, jenž se nacházejí na vedlejších površích. Čím větší poloměr koule, tím větší oblast bude zasažena, ale i v případě, že bychom použili opravdu maličkou kouli (třeba s poloměrem jedna), tento problém nevyřešíme, protože by hrany „svítily“ (stále by byly nalezeny nějaké fotony).

Tento problém vyřešíme použitím disku namísto koule. V praxi spíše elipsoidu, protože samotný disk by nemusel dávat uspokojivé výsledky na nerovných površích. Obrázky napravo (7.1) používají jako obálku elipsoid, který vyřadí (odmítne) většinu nepatřičných foton. Výsledek je díky poté vizuálně přijatelnější.

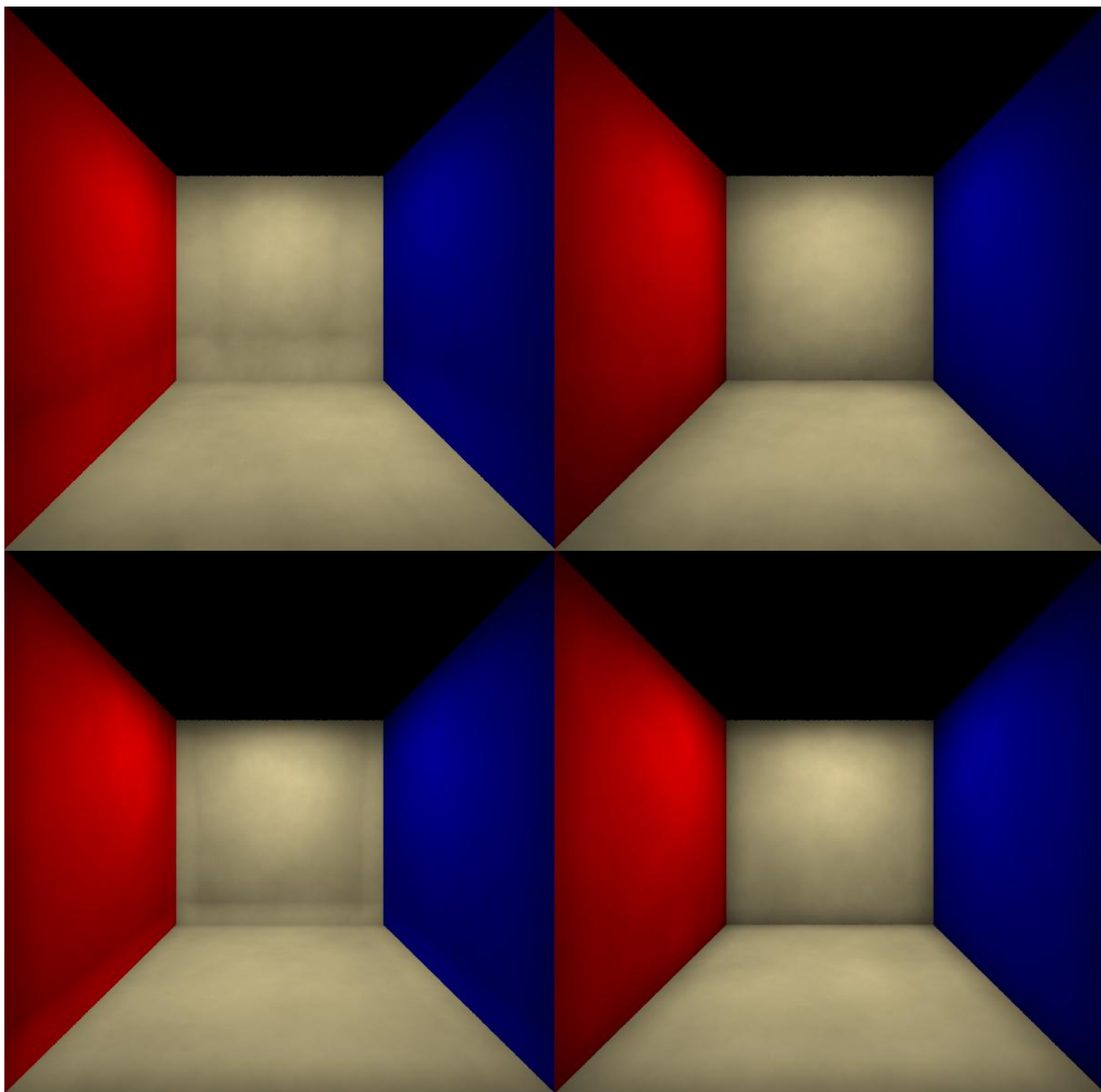
7.1.2 Filtry

Byly otestovány filtry zmíněné v kapitole 6.1. Navíc bylo pro srovnání přidáno Simpsonovo jádro, které, na rozdíl konstantního (případ, kdy nejsou použity žádné filtry) jádra, také vyhlazuje osvětlení, podobně jako filtry.

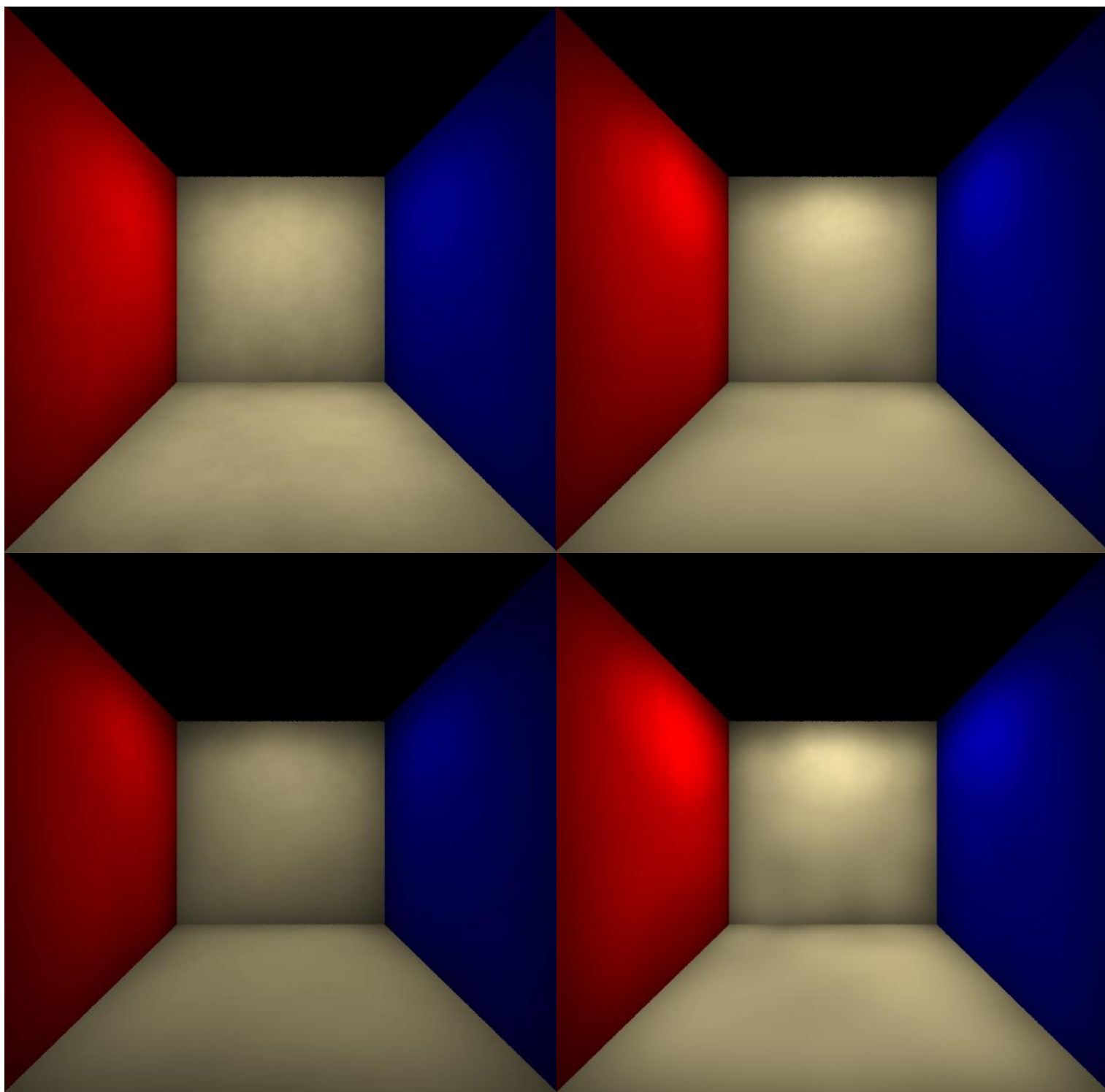
Na obrázku 7.2 je scéně s „vyváženými“ parametry, které zajišťují, že scéna vypadá i bez filtrů poměrně dobře. Nicméně, zadní stěna není úplně hladká. Právě filtry by měli pomoci tento problém řešit. Kuželový filtr řešení skutečně vyhladil, ale zároveň také trochu zesvětlil, zejména v místech kam dopadá nejvíce světelného výkonu. Gaussův filtr dává asi vizuálně nejlepší řešení, které je ale nepatrně tmavší než v případě bez filtru. Simpsonovo jádro dává podobné výsledky jako kuželový filtr, akorát přesvětluje ještě větší oblast.

Na obrázku 7.3 jsou naopak parametry nastaveny tak, aby řešení nebylo příliš spojitě a byl dobře vidět účinek filtrů. Na obrázku bez filtrů jsou vidět „kolečka“ (jak probíhal odhad radiance pomocí disku a bylo nastaveno málo fotonů v odhadu). Filtry tyto artefakty spolehlivě vyhladí, ačkoliv výslednou vizuální kvalitu nezachrání. Samotné výsledky konkrétních filtrů jsou podobné jako v předchozím případě. Kuželový filtr mírně přesvětluje, ale jinak dává poměrně dobré výsledky. Gaussův filtr vyhladí řešení nejméně a zároveň ho o něco ztmaví. Simpsonovo jádro vyhladí řešení perfektně, některé části scény se zesvětlí a některé ztmaví (podle toho jak vzdálené fotony byly použity k odhadu radiance).

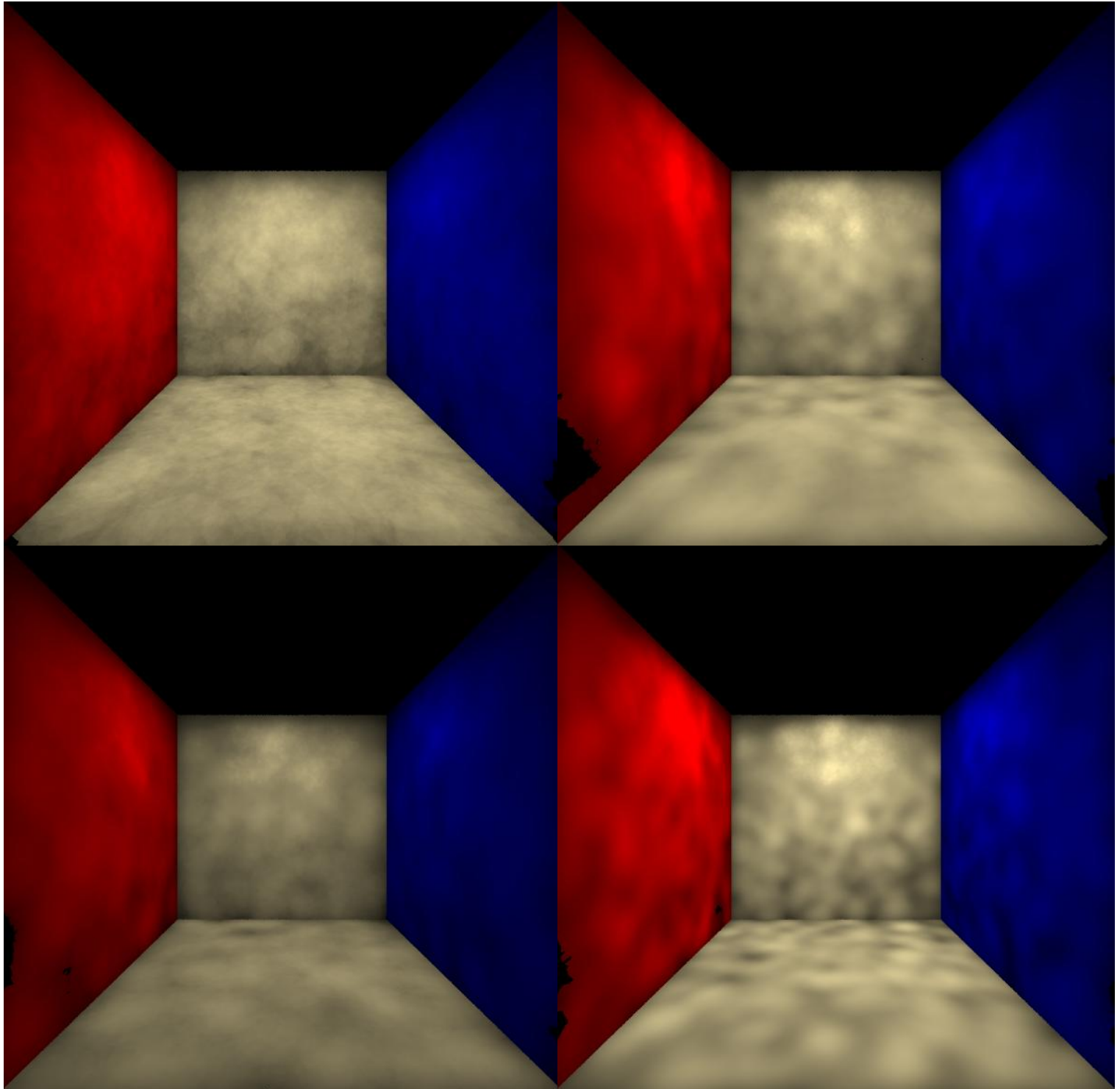
Tento test dokázal, že filtry jsou opravdu určeny především pro kaustiky (více v kapitole 7.4), protože uvádí do řešení zkreslení (zesvětlují či ztmavují), ale ve speciálních případech je lze použít (především pokud používáme finální sbírání, tedy nezobrazujeme globální fotonovou mapu přímo). Jako nejvýhodnější filtr byl shledán Gaussův filtr, jelikož dává nejspokojivější výsledky, ačkoliv ztmavuje scénu. Simpsonova jádro výborně vyhlazuje, proto je vhodné pro nepřímé osvětlení, které je přirozeně hladké.



Obrázek 7.1: Obrázky vlevo používaly jako obálku kouli, zatímco obrázky vpravo disk (elipsoid).



Obrázek 7.2: Obrázek vykreslený bez filtru (vlevo nahoře), s kuželovým filtrem (vpravo nahoře), s Gaussovým filtrem (vlevo dole) a nakonec obrázek používající Simpsonovo jádro (vpravo dole).



Obrázek 7.3: Obrázek vykreslený bez filtru (vlevo nahoře), s kuželovým filtrem (vpravo nahoře), s Gaussovým filtrem (vlevo dole) a nakonec obrázek používající Simpsonovo jádro (vpravo dole).

7.2 Přímé osvětlení

Jako obálka při odhadu radiance byl použit disk. Filtry nebyly použity vůbec, aby nebyla do řešení přidána další nestrannost.

7.2.1 Metody

Monte Carlo sledování paprsku

Algoritmus sledování paprsku (kapitola 4.1.1) byl první algoritmus na výpočet přímého světlení (kromě vrhání paprsku, ale to nebylo rekurzivní a nepodporovalo spekulární materiály). Na jeho základě byly založeny téměř všechny ostatní zobrazovací algoritmy kromě radiosity. V dnešní době se používá takzvané Monte Carlo sledování paprsku, které nahrazuje některé vzorky integrálem. Podporuje také plošné světelné zdroje, měkké stíny, rozmazání, hloubku ostrosti a další.

Přímá vizualizace fotonových map

Jak bylo naznačeno v kapitole 5.5.3, globální fotonová mapa může být vizualizovaná přímo pomocí jednoduchého Monte Carlo sledování paprsku, kdy namísto vzorkování světla použijeme fotonovou mapu k získání přicházející světelného výkonu. Tento případ lze upravit tak, aby byly fotony ukládány při prvním protnutí scény. Ačkoliv tedy normálně umí algoritmus vypočítat kompletní zobrazovací rovnice, zde budeme uvažovat pouze o přímém osvětlení, protože to je zde objekt našeho zájmu.

7.2.2 Výsledky

Cornell box – prázdná

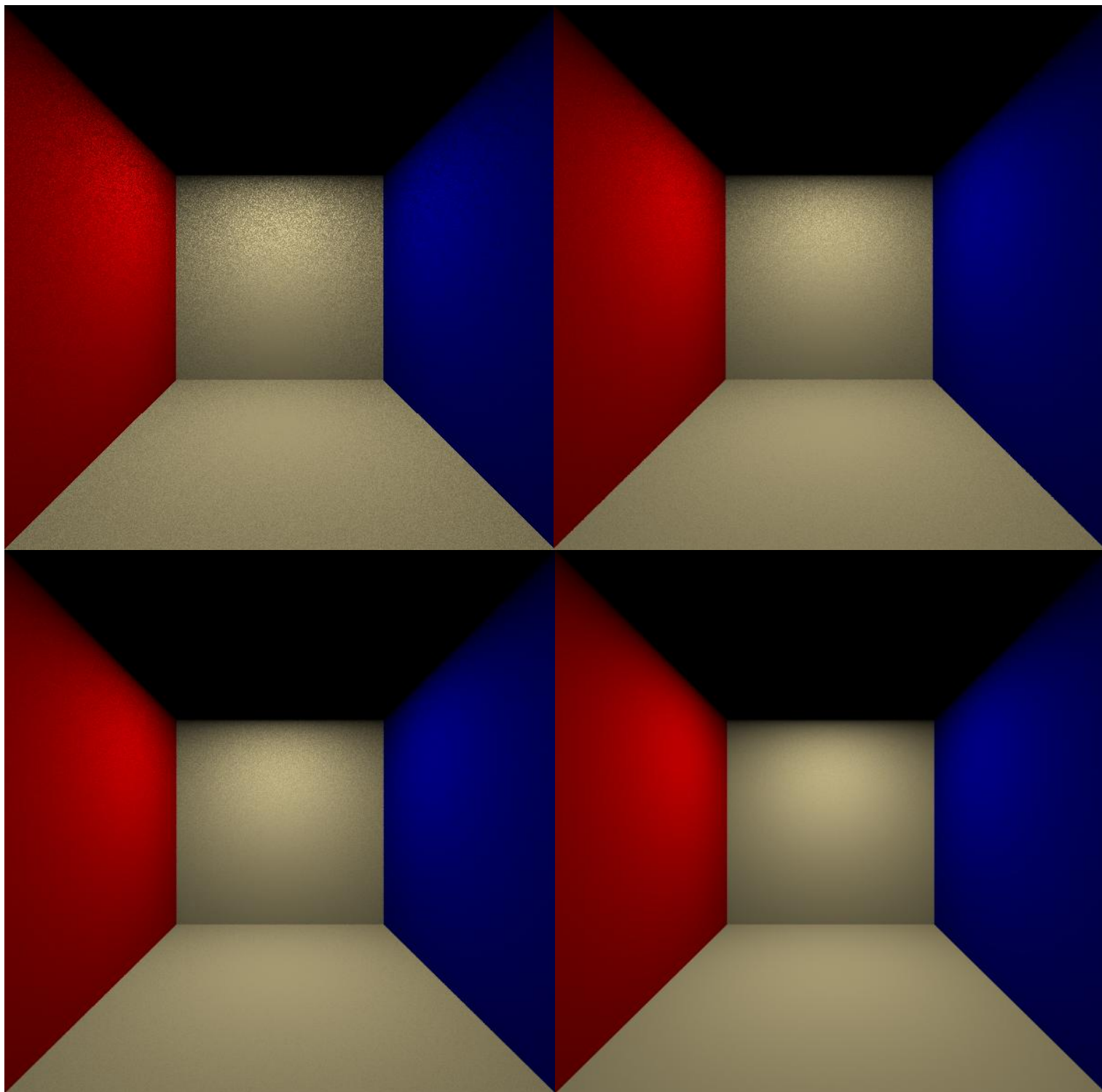
Prázdná Cornell box je vůbec nejjednodušší scénou, která je k dispozici, protože zde nejsou žádné objekty, které by mohly potenciálně způsobit vržené stíny či kaustiky.

Sledování paprsku		
ID	Paprsky srze pixel	Čas [s]
1	1	3,6
2	4	11,9
3	10	26,8
4	50	146,5
5	100	290,9

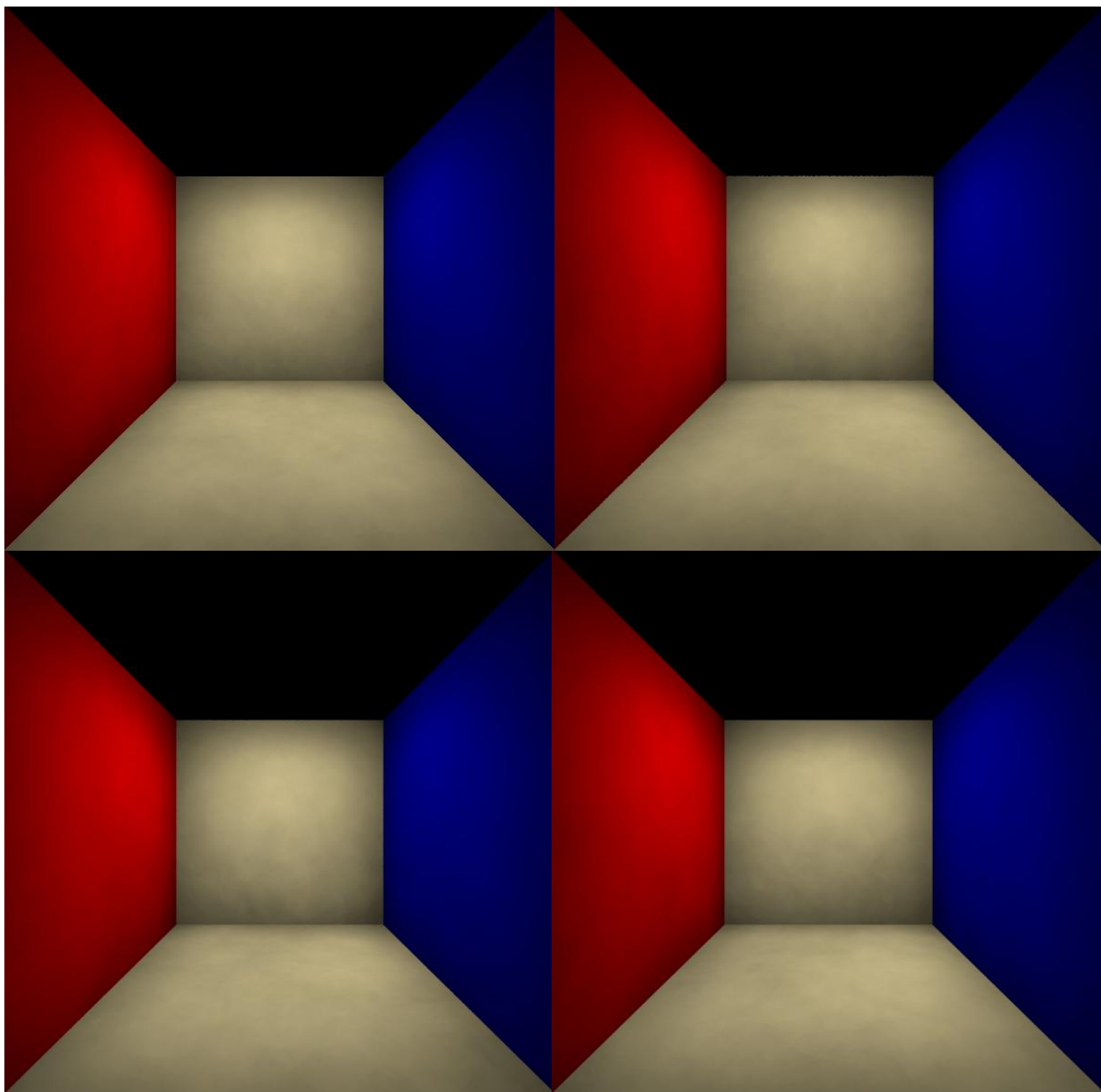
Tabulka 1: Sledování paprsku – prázdná Cornell box.

Fotonové mapy					
ID	Počet fotonů (FM)	Počet fotonů (odhad)	Poloměr obálky	Paprsky srze pixel	Čas [s]
1	10 000	500	100	1	34,1
2	10 000	500	100	4	132,1
3	10 000	500	100	10	294,7
4	10 000	500	100	50	1551,7
5	100 000	500	50	4	316,7
6	1 000 000	1000	200	4	2101,7
7	10 000	500	30	4	75,8
8	10 000	15	50	4	20,4

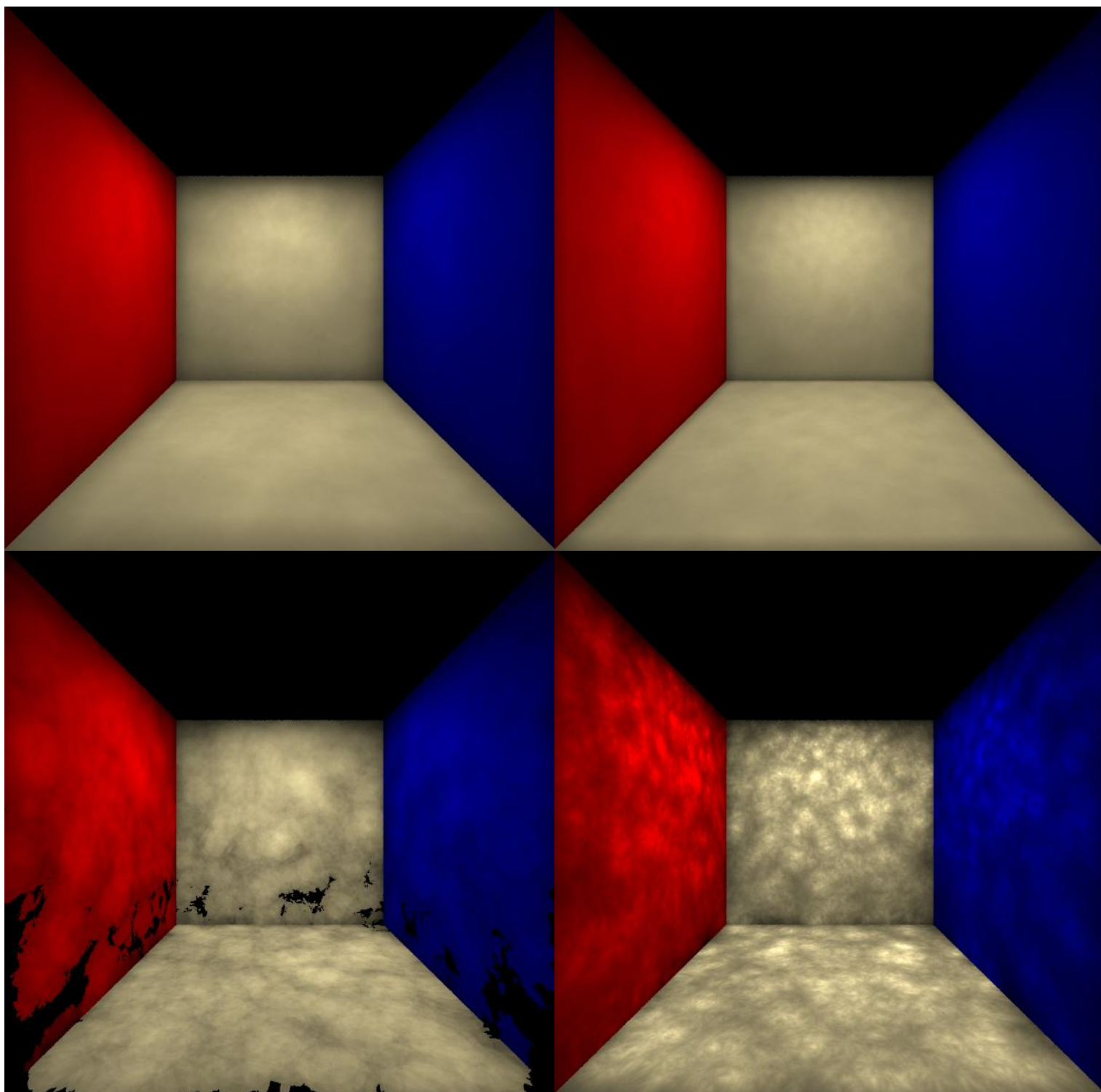
Tabulka 2: Fotonové mapy – prázdná Cornell box.



Obrázek 7.4: Obrázky vykreslené pomocí sledování paprsku za použití 1 paprsku procházejícího skrze pixel (vlevo nahoře), 4 paprsků (vpravo nahoře), 10 paprsků (vlevo dole) a 100 paprsků (vpravo dole). Bližší informace o parametrech v tabulce 1.



Obrázek 7.5: První čtyři obrázky z tabulky 2 vykreslené pomocí přímé vizualizace fotonových map.



Obrázek 7.6: Druhé čtyři obrázky z tabulky 2 vykreslené pomocí přímé vizualizace fotonových map.

Cornell box – difuzní

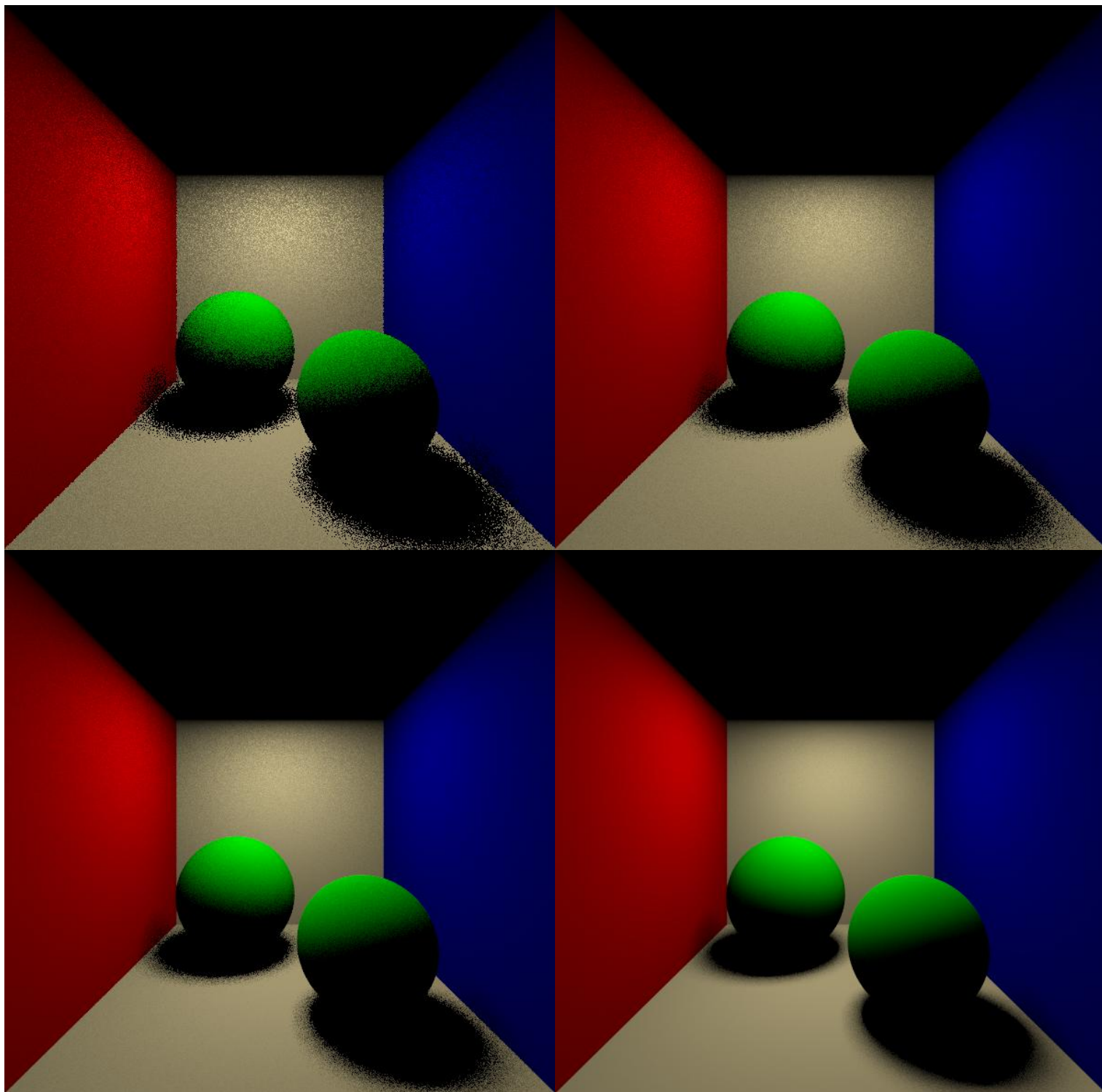
Difuzní Cornell box obsahuje navíc dvě ideálně difuzní (Lambertův model odrazivosti – kapitola A.1) koule, které jsou umístěny tak, aby vrhaly stíny na spodní postavu i na boční stěny.

Sledování paprsku		
ID	Paprsky srze pixel	Čas [s]
1	1	5,2
2	4	19,0
3	10	45,7
4	50	237,8
5	100	470,0

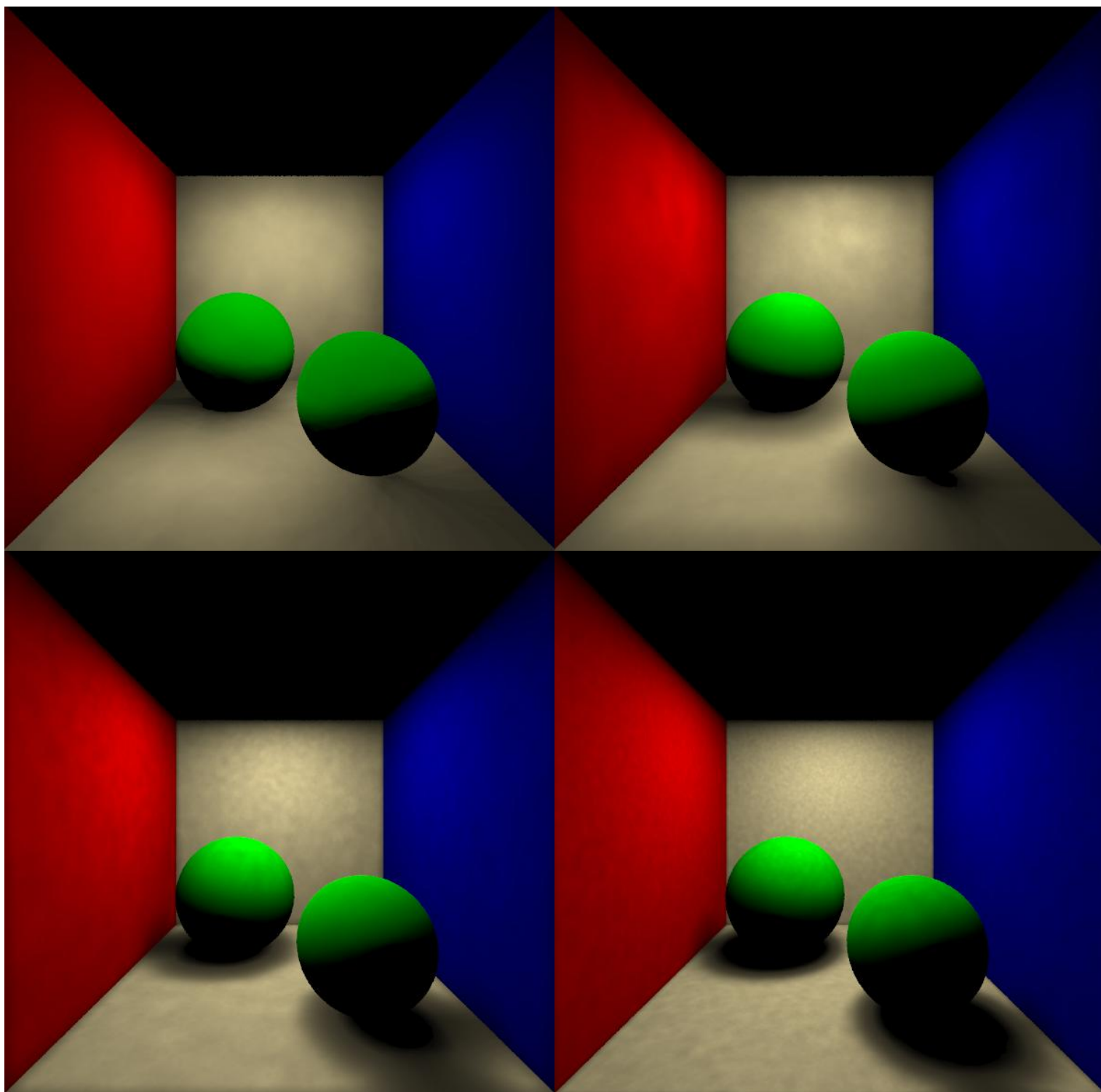
Tabulka 3: Sledování paprsku – difuzní Cornell box.

Fotonové mapy					
ID	Počet fotonů (FM)	Počet fotonů (odhad)	Poloměr obálky	Paprsky srze pixel	Čas [s]
1	10 000	500	100	4	160,3
2	100 000	500	80	4	430,8
3	1 000 000	500	50	4	957,3
4	5 000 000	500	25	4	1327,6

Tabulka 4: Fotonové mapy – difuzní Cornell box.



Obrázek 7.7: Obrázky vykreslené pomocí sledování paprsku za použití 1 paprsku procházejícího skrze pixel (vlevo nahoře), 4 paprsků (vpravo nahoře), 10 paprsků (vlevo dole) a 100 paprsků (vpravo dole). Bližší informace o parametrech v tabulce 3.



Obrázek 7.8: Obrázky z tabulky 4 vykreslené pomocí přímé vizualizace fotonových map.

Cornell box – spekulární

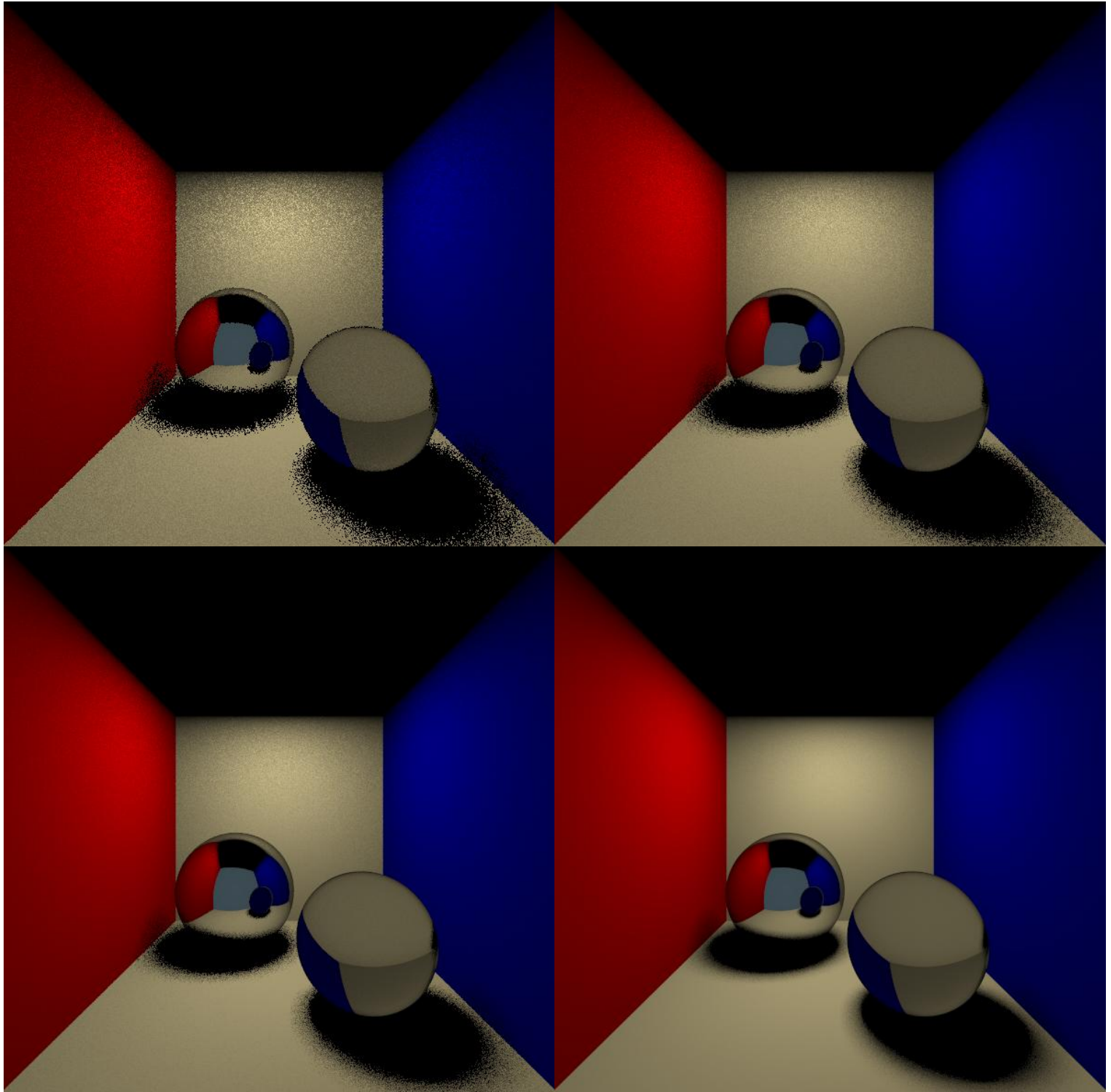
Spekulární Cornell box je vůbec nesložitější scénou, protože obsahuje dvě spekulární koule, z nichž jedna světlo odráží ve směru ideálního odrazu, zatímco druhá ve směru ideálního lomu. Nejenže jsou ve scéně obsaženy vržené stíny, ale také lze najít kaustiky, které nejsou v této kapitole vyšetřovány.

Sledování paprsku		
ID	Paprsky srze pixel	Čas [s]
1	1	6,0
2	4	22,9
3	10	52,1
4	50	277,2
5	100	559,5

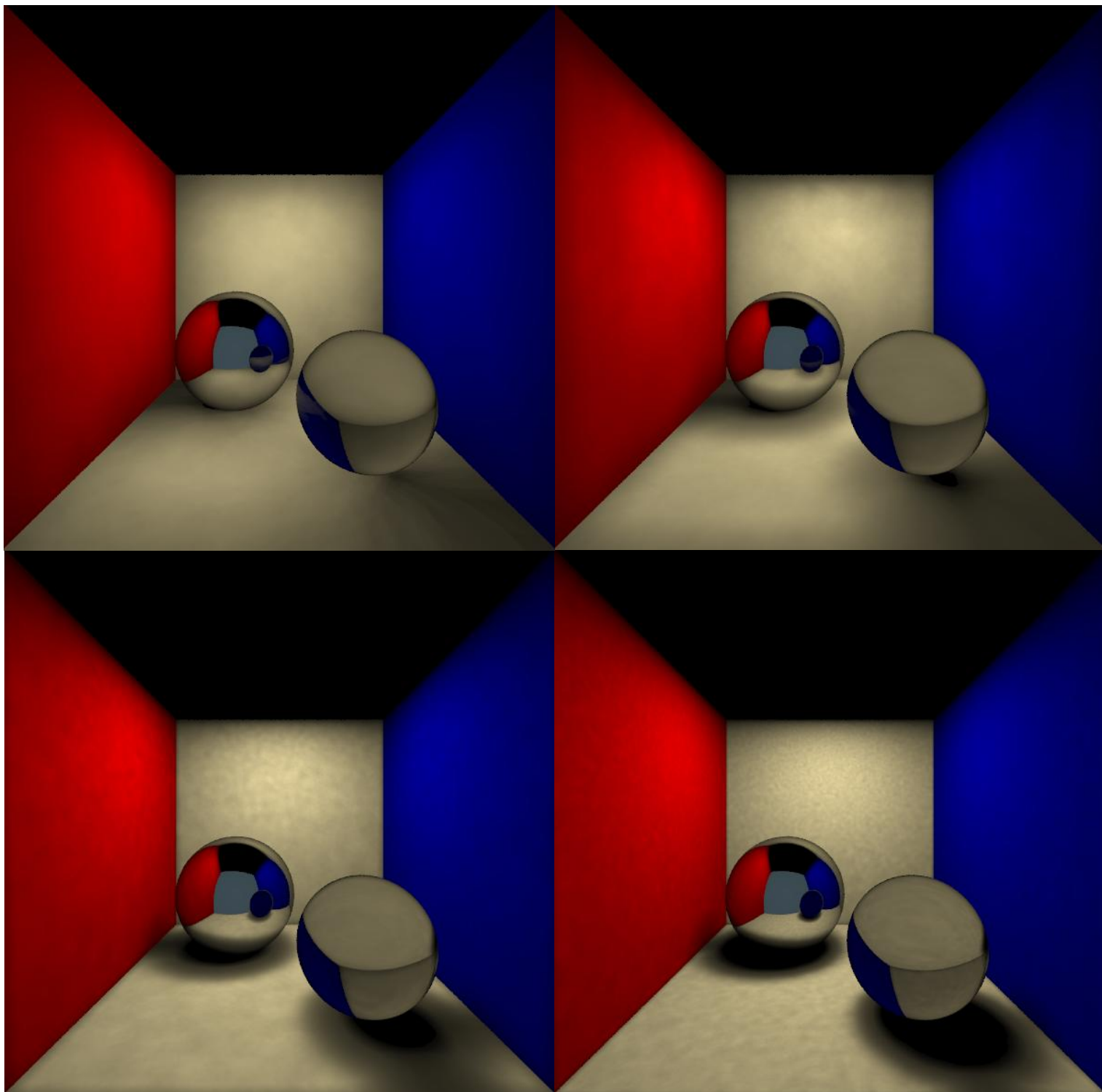
Tabulka 5: Sledování paprsku – spekulární Cornell box.

Fotonové mapy					
ID	Počet fotonů (FM)	Počet fotonů (odhad)	Poloměr obálky	Paprsky srze pixel	Čas [s]
1	10 000	500	100	4	143,9
2	100 000	500	80	4	397,1
3	1 000 000	500	50	4	1110,8
4	5 000 000	500	25	4	1267,5

Tabulka 6: Fotonové mapy – spekulární Cornell box.



Obrázek 7.9: Obrázky vykreslené pomocí sledování paprsku za použití 1 paprsku procházejícího skrze pixel (vlevo nahoře), 4 paprsků (vpravo nahoře), 10 paprsků (vlevo dole) a 100 paprsků (vpravo dole). Bližší informace o parametrech v tabulce 5.



Obrázek 7.10: Obrázky z tabulky 6 vykreslené pomocí přímé vizualizace fotonových map.

7.2.3 Diskuse

Cornell box - prázdná

V první porovnávané scéně nejsou žádné stíny ani kaustiky, proto je naše pozornost soustředěna na samotnou kvalitu přímého osvětlení. Algoritmus sledování paprsku krásně demonstruje nestranný přístup. Čím více paprsků vystřelíme, tím přesnější řešení budeme mít. Algoritmus založený na přímé vizualizaci fotonových map je naopak zkreslený algoritmus, ačkoliv je konzistentní, takže také platí, že čím více fotonů vystřelíme, tím přesnější řešení můžeme dostat. Zkreslení znamená, že je uvedena v algoritmu další chyba (více v kapitole 5.1).

Na obrázku 7.4 jsou scény vykreslené pomocí sledování paprsku. Všechny čtyři obrázky krásně demonstrují to, co bylo právě řečeno. Tabulka 1 obsahuje důležité parametry jednotlivých vykreslování. Při vykreslování stačí měnit pouze jediný parametr a řešení konverguje ke správnému řešení. Je pouze potřeba zjistit, kolik paprsků je opravdu nutných. Obrázek vykreslený pomocí 100 paprsků je bez šumu, nicméně to je téměř i obrázek vykreslený 50 paprsky (uložen na CD). Ideální hodnota by tedy byla něco mezi nimi. Za pomoci adaptivních technik (kapitola 6.2) můžeme redukovat počty paprsků v závislosti na chybě řešení v daném pixelu.

Na obrázku 7.5 a 7.6 jsou zobrazeny výsledky vykreslené pomocí fotonových map. Tabulka 2 obsahuje důležité hodnoty týkající se jednotlivých vykreslování. Tabulka je značně rozsáhlejší, protože je nutné nastavit daleko více parametrů než jen počet paprsků procházejících pixelem. První obrázek (vlevo nahoře) byl vykreslen za poměrně slušný čas ve slušné kvalitě. Stále je však v některých částech nespojitý. Další tři obrázky jsou stejné až na počet paprsků procházejících pixelem. I když zvedneme počet paprsků na 50, stále se nedostavuje cílový efekt – „hladký“ výsledek. Následující 4 obrázky (obrázek 7.6) tedy posílají pixelem pouze 4 paprsky (z důvodu zajištění vyhlazení hran). Pro zlepšení kvality výsledku posíláme (levý horní obrázek) 100 000 fotonů a zmenšíme poloměr disku. V dalších obrázku zvětšíme počet fotonů na 1 000 000 a poloměr opět zmenšíme. Ani v jednom případě se nám nepodařilo dosáhnout požadované kvality. Poslední dva obrázky ukazují, co se stane, pokud zvolíme poloměr nebo počet fotonů v odhadu opravdu špatně. V prvním případě jsme zvolili příliš malý poloměr, takže bylo nalezeno málo fotonů vzhledem k počtu hledaných. Ve většině regionů to způsobilo, že jsou vidět „kruhy“. V některých bylo dokonce nalezeno tak málo fotonů, že byl výsledkem černý pixel. V druhé scéně bylo naopak hledáno příliš málo fotonů, což způsobilo naprostou nespojitost řešení.

Při porovnání algoritmů musíme vzít v potaz několik faktorů:

- Kvalita řešení mluví jednoznačně ve prospěch sledování paprsku, které od deseti paprsků skrz pixel poskytuje vizuálně slušná řešení a nejhůře od sto paprsků skrze pixel vizuálně dokonalá řešení.
- Čas potřebný k řešení má sledování paprsků výrazně nižší, takže i zde je jasným vítězem.

Proč tedy používat přímou vizualizaci fotonových map? Jednoduše, protože umožňuje vypočítat nepřímé osvětlení spolu s přímým osvětlením. Zde jsme nepřímé osvětlení schválně nepoužili, protože jsme chtěli porovnat kvalitu přímého osvětlení vykresleného pomocí fotonových map s metodou sledování paprsku, která se na tento výpočet běžně používá.

Cornell box – difuzní a Cornell box – spekulární

V těchto dvou scénách přibudou vržené stíny. V druhé scéně navíc spekulární materiály a tedy i kaustiky. Obrázky 7.7 a 7.9 byly vykresleny pomocí sledování paprsku a 7.8 a 7.9 pomocí přímé vizualizace fotonových map. V tabulkách 3 a 5 jsou uloženy parametry pro sledování paprsku a v 4 a 6 pro fotonové mapy.

Sledování paprsku musí tentokrát použít i stínové paprsky, které zjišťují, zda je bod pro světelný zdroj viditelný nebo ne. Díky stínovým paprskům se čas nutný k výpočtu prodlouží. Snížit lze například pomocí stínových fotonů (kapitola 6.6). Situace sledování paprsku je podobná jako v případě prázdné scény. Je nutné použít buď, více paprsků procházejících pixelech nebo více stínových paprsků. V našem případě jsme měnili pouze počet paprsků procházejících pixelem. Na pravých spodních obrázcích bylo použito 100 paprsků a stíny vypadají přirozeně, ačkoliv by neškodilo zvýšit počet stínových paprsků.

Fotonové mají s vrženými poměrně značné problémy (levý horní obrázek). Nezbyvá než zvyšovat počet fotonů ve fotonové mapě a snižovat poloměr disku. Na pravém spodním obrázku je vidět, že v případě použití 5 000 000 fotonů už vypadají stíny vizuálně dobře (ačkoliv by měl být plný stín ještě o něco větší a polostín o něco menší). Nicméně doba trvání výpočtu je neúnosná. Navíc přímé osvětlení na jednotlivých stěnách není úplně hladké (bylo by potřeba jinak nastavit parametry, případně vystřelit více fotonů). Řešení tohoto problému mohou přinášet právě filtry, které zajistí jemné přechody, ale uvedou do řešení další zkreslení.

7.2.4 Shrnutí

Přímá vizualizace fotonových je, co se přímého osvětlení týče, ve všem horší než sledování paprsku. Má především problém s vrženými stíny a spojitostí osvětlení v některých částech scény. Zároveň však toto srovnání dobře demonstruje rozdíl mezi nestrannými a zkreslenými metodami. Zatímco zkreslené mají pouze několik málo parametrů na nastavení celého procesu vykreslování, zkreslené mají většinou parametrů více a ty musí být správně nastaveny, aby se při zvětšení počtu vzorků skutečně zpřesnilo řešení. Vzhledem k této „závislosti“ zkreslených metod na nastavení správných hodnot parametrům je přímá vizualizace map opravdu jediný algoritmus, kdy se využívají pro výpočet přímého osvětlení. Jejich hlavní využití je pro výpočet nepřímého osvětlení.

7.3 Nepřímé osvětlení

I v této kapitole je použit výhradně disk (elipsoid) jako obálka a filtry opět nejsou použity.

7.3.1 Metody

Sledování cesty

Algoritmus sledování cesty je zde jediným zástupcem zcela nestranné metody. Není použit originální algoritmus, ale je vylepšen vzorkování podle důležitosti a samozřejmě odhadem příští události.

Fotonové mapy (přímá vizualizace)

Tento přístup je vlastně metoda použitá v minulé kapitole na výpočet přímého osvětlení s jedinou změnou – zde bude použito pouze nepřímé osvětlení. Při sledování fotonů je použita ruská ruleta, která odmítá fotony s nízkou důležitostí. Tím zajišťuje, že bude ve fotonové mapě málo „nedůležitých“ fotonů. Při výpočtu odraženého fotonu by měl mít foton energii i za fotony, které byly zamítnuty jako nedostatečně důležité (energie fotonu by měla být vydělena pravděpodobností pokračování fotonů). Díky tomuto však vznikají občas fotony, které nesou obrovské množství energie (vzhledem k ostatním) a ty poté narušují korektnost odhadu radiance, protože v nějaké oblasti zvýší množství přicházejícího výkonu. Následně v obrázku vznikají světlé kruhy. Proto jsme do řešení raději uvedli zkreslení tím, že jsme energii fotonů nedělili pravděpodobností. Po této úpravě vypadají výsledné obrázky vizuálně korektně.

Fotonové mapy (finální sbírání)

Algoritmus finálního sbírání byl popsán v kapitole 5.5.3. Namísto přímé vizualizace fotonové mapy vystřeluje mnoho paprsků (vzorkováním BRDF)

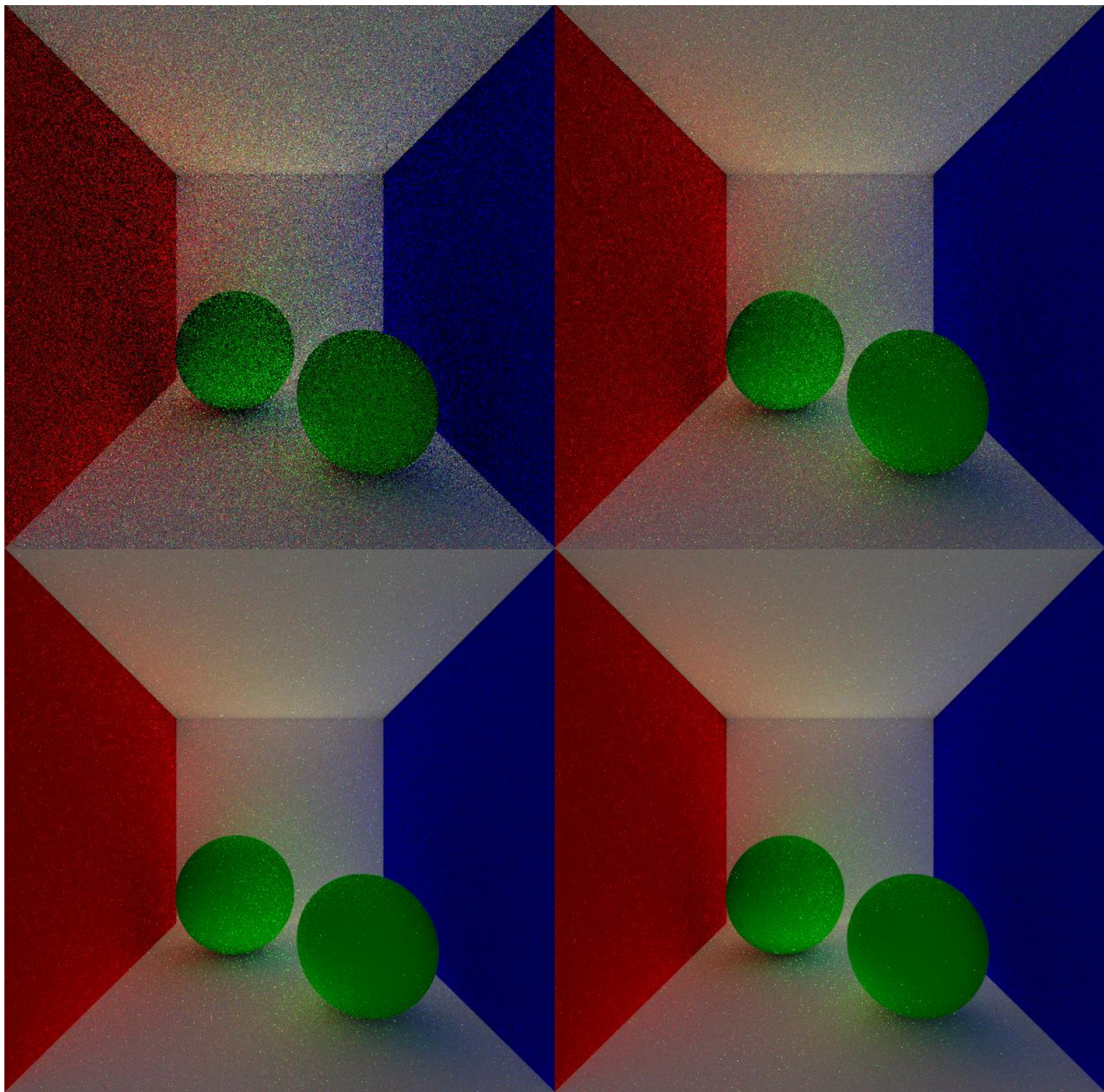
7.3.2 Výsledky

Sledování cesty			
ID	Stínové paprsky	Paprsky srze pixel	Čas [s]
1	10	10	133,1
2	5	50	669,0
3	1	500	3860,4
4	1	1000	6739,3

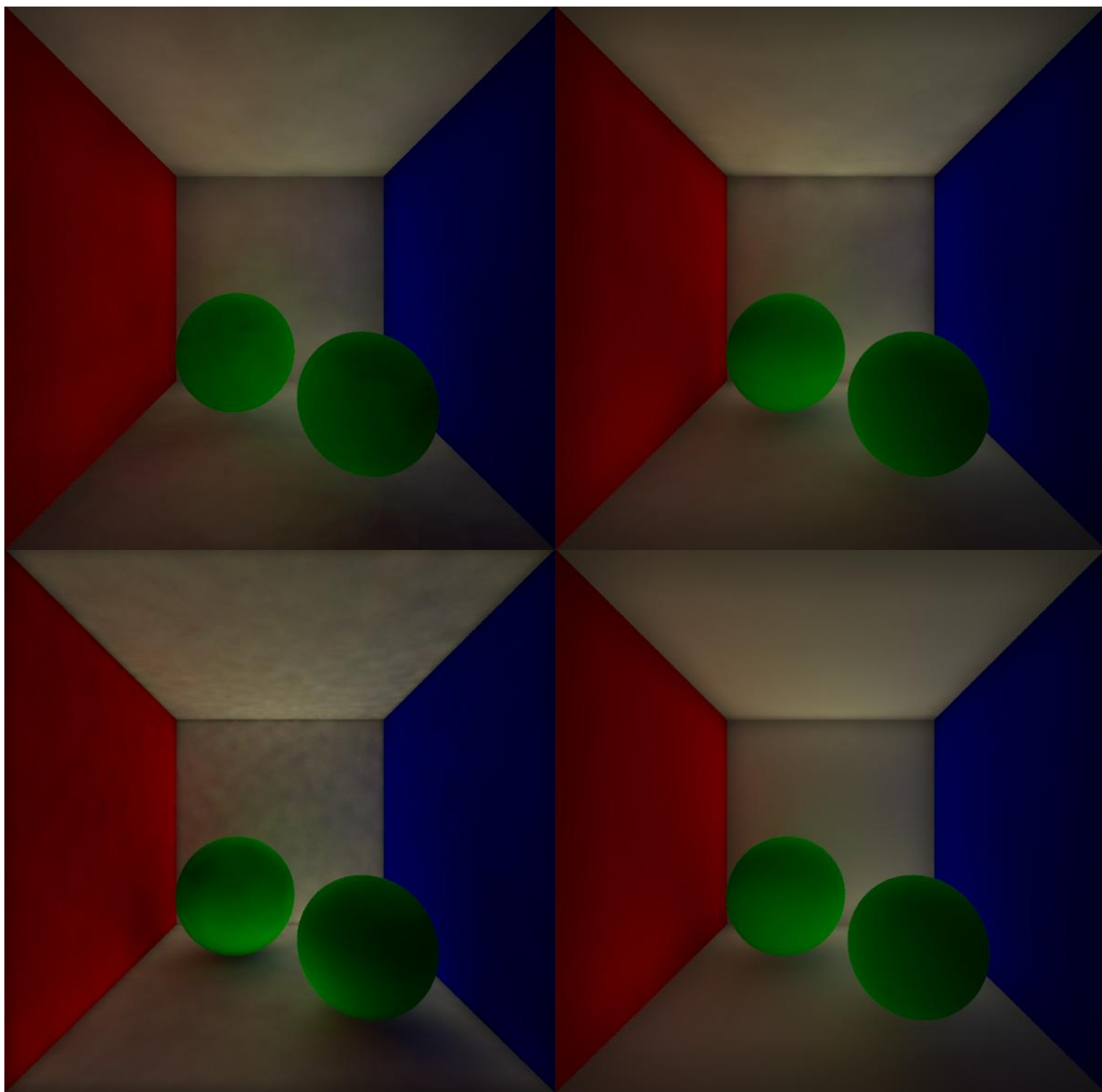
Tabulka 7: Nepřímé osvětlení – sledování cesty.

Fotonové mapy					
ID	Počet fotonů (FM)	Počet fotonů (odhad)	Poloměr obálky	Paprsky srze pixel	Čas [s]
1	10 000	500	100	4	158,4
2	100 000	800	100	4	851,1
3	1 000 000	500	50	4	916,5
4	1 000 000	10 000	100	4	20 083

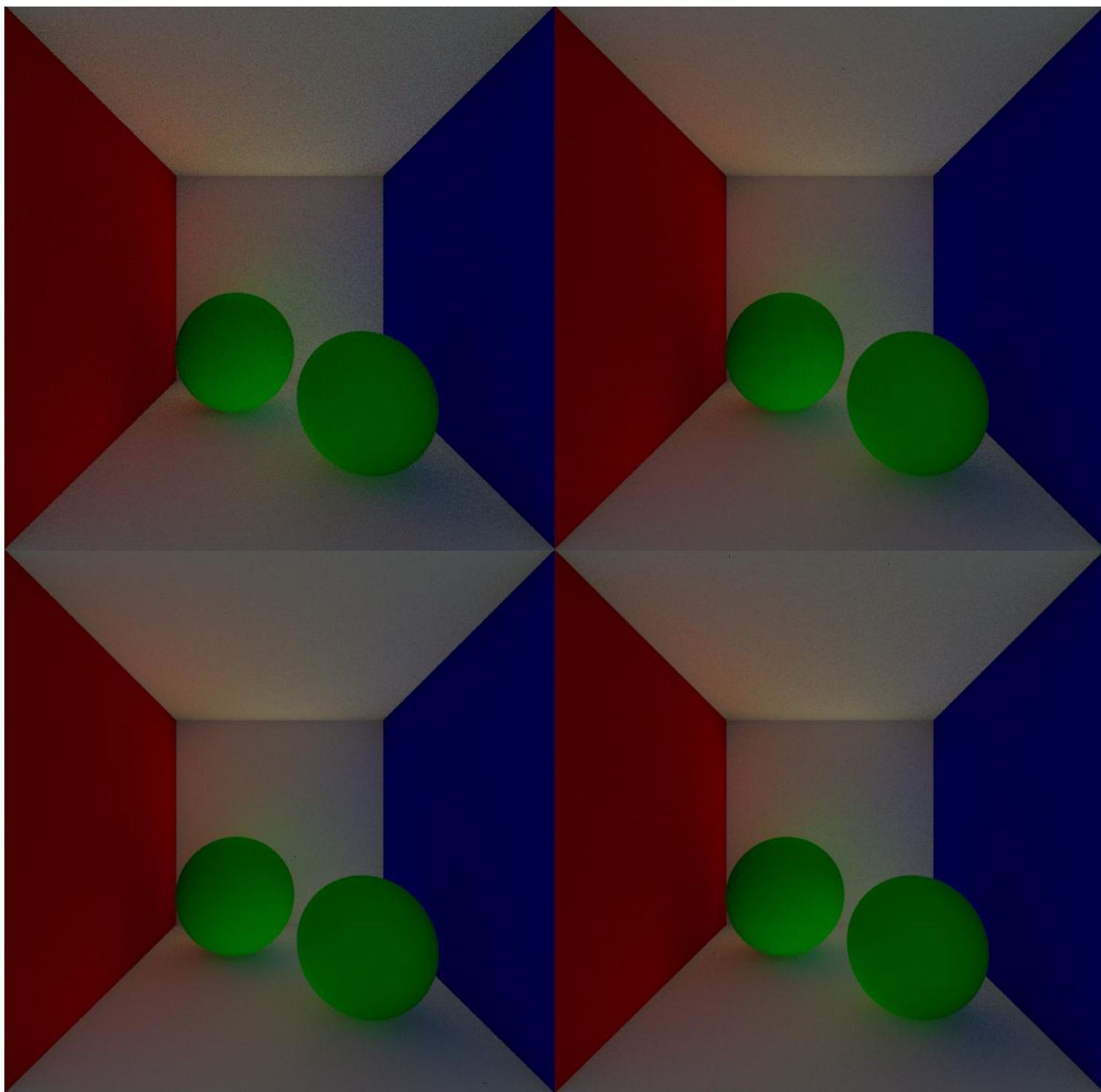
Tabulka 8: Nepřímé osvětlení – fotonové mapy.



Obrázek 7.11: Obrázky z tabulky 7 vykreslené pomocí sledování cesty.



Obrázek 7.12: Obrázky z tabulky 8 vykreslené pomocí přímého zobrazování fotonových map.



Obrázek 7.13: Obrázky z tabulky 9 vykreslené pomocí finálního sbírání fotonů z globální fotonové mapy.

Finální sbírání						
ID	Počet fotonů (FM)	Počet fotonů (odhad)	Poloměr obálky	Finální paprsky	Paprsky srze pixel	Čas [s]
1	10 000	500	100	10	4	1344,0
2	10 000	500	100	20	4	2535,0
3	10 000	500	100	40	4	5026,3
4	10 000	500	100	1	100	3403,9

Tabulka 9: Nepřímé osvětlení – finální sbírání.

7.3.3 Diskuse

Pro porovnávání jsme záměrně zvolili scénu obsahující dvě difuzní koule, které jsou navíc zcela zelené. Metoda sledování cesty zde demonstruje kompletně nestranný algoritmus založený na Monte Carlo metodách. Naopak přímé zobrazování fotonových map reprezentuje zkreslený algoritmus, ve kterém je obtížné kontrolovat dodatečnou chybu. Finální sbírání je kompromisem mezi těmito přístupy. Vzhledem k vlastnostem těchto metod, můžeme jejich porovnání brát také jako porovnání nestranných a zkreslených metod.

Obrázek 7.11 zobrazuje 4 obrázky vykreslené pomocí sledování cesty několika pixelů. Při sledování pouhých 10 či 50 paprsků (ačkoliv bylo vystřeleno navíc několik stínových paprsků) je šum opravdu značný i v nízkém rozlišení. Je třeba podotknout, že pixely byly vzorkovány pomocí techniky roztřesení, která zajišťuje rovnoměrné rozložení vzorků v oblasti pixelu. Ačkoliv bylo také použito vzorkování podle důležitosti (a samozřejmě také odhad příští události) při vzorkování nových paprsků, i při vystřelení celých 1000 paprsků skrze každý pixel byl výsledný obrázek plný šumu. Výrazného zlepšení lze dosáhnout, pokud použijeme vzorkování podle fotonové mapy (kapitola 6.4).

Přímá vizualizace fotonových map dává výsledky bez šumu, ale také nejsou zcela spojitě, protože je poměrně obtížné najít optimální parametry. V tabulce 8 můžeme vidět použité hodnoty. V případě prvních dvou obrázků bylo použito malé množství fotonů, které nemohlo zachytit všechny detaily řešení. Po navýšení počtu fotonů a snížení poloměru se zlepší některé detaily a hlavně stíny. Poloměr je však příliš malý a řešení není dostatečně spojitě. Čtvrtý obrázek demonstruje scénu s velkým počtem fotonů (vzhledem k jednoduchosti scény), větším poloměrem a hlavně velkým množstvím fotonů použitých k odhadu. Co se vizuální kvality týče, můžeme být spokojeni, protože řešení je spojitě a zachycuje i jemné detaily, ačkoliv „krvácení barev“ je dosti nevýrazné. Hlavním problémem je čas výpočtu, který je opravdu vysoký.

U finálního sbírání jsme v každém ze čtyř případů použili pouze 10 000 fotonů (tabulka 9), poloměr obálky 100 a 500 fotonů v odhadu radiance. Měnil se především počet paprsků pro finální sbírání, případně počet paprsků procházejících pixelem. První obrázek (levý horní 7.13) není sice zdaleka bez šumu, ale rozhodně zobrazuje korektně nepřímé osvětlení ve všech místech scény. Na rozdíl od sledování cesty neobsahuje velmi světlé pixely, které doslova bijí do očí. Navíc byl obrázek vykreslen 5x krát rychleji než pomocí sledování paprsku. Pokud

zvýšíme počet paprsků (ať už finálních či hlavních) řešení se stává spojitějším. Ani při použití 40x4 finálních paprsků (třetí obrázek) není řešení zcela spojitě.

7.3.4 Shrnutí

Jednoznačně nejvýhodnější metodou je finální sbírání, které je jako jediná metoda schopná vypočítat přesné nepřímé osvětlení ve slušném čase. Nicméně hlavní výhodou finálního sbírání je dostupnost mnoha urychlovacích technik popsanych v šesté kapitole (např. irradiance caching, předpocítaná irradiance), které výpočet několikanásobně urychlí. Podotýkám, že žádná z těchto technik nebyla při testech použita a i tak byly výsledky poměrně uspokojivé.

7.4 Kaustiky

V této kapitole se zaměříme na vykreslování kaustik pomocí fotonových map a zlepšení jejich kvality pomocí filtrů.

7.4.1 Diskuse

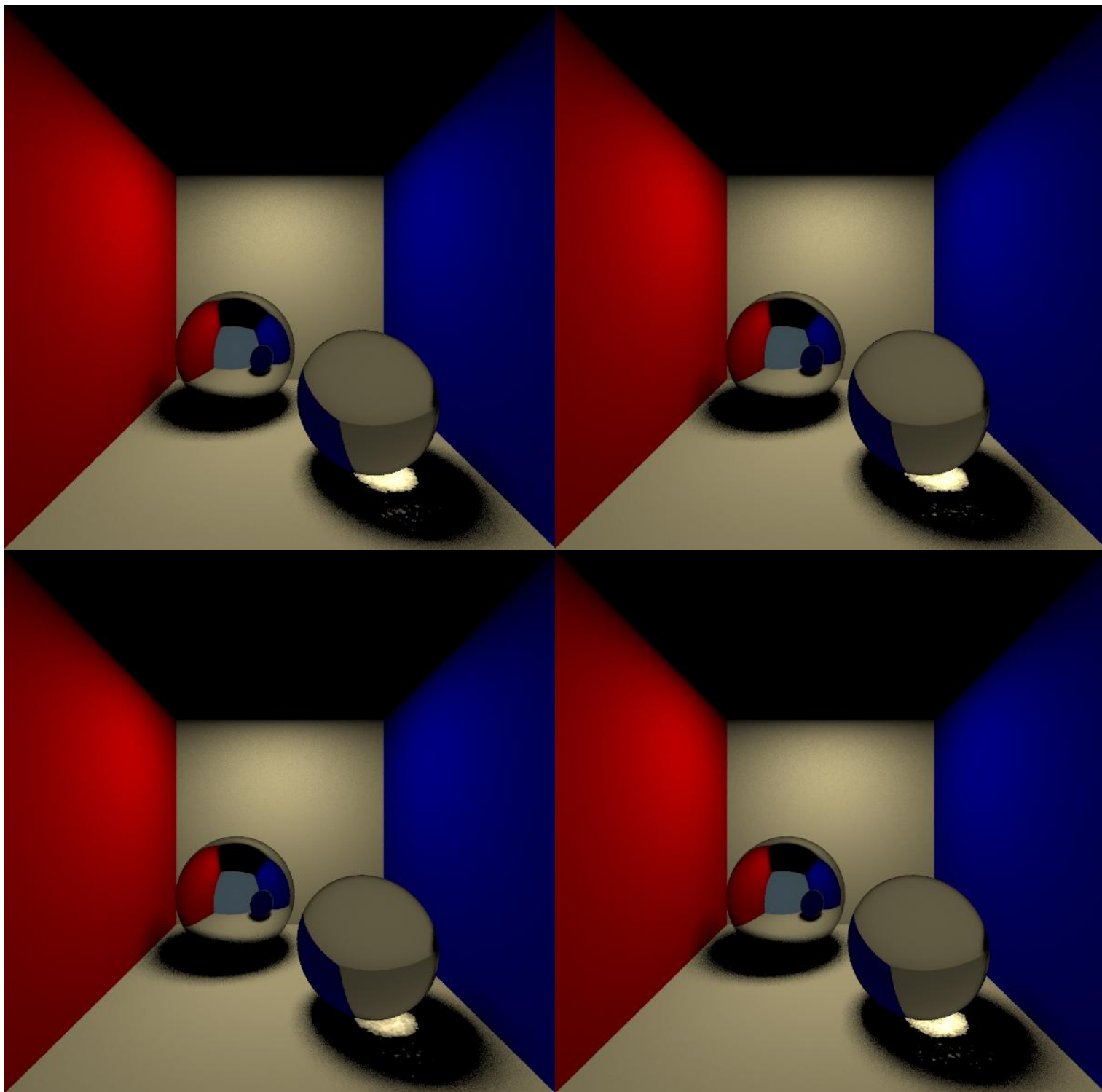
Kaustiky jsou nejčastějším zdrojem šumu ve výsledném obrázku, pokud je zobrazování prováděno pomocí nestranných metod. Metoda fotonových map ukládá „fotony kaustik“, jak do globální fotonové mapy, tak do mapy kaustik. Přímým vykreslením fotonové mapy teda také dostaneme kaustiky, ale pravděpodobně budou velice nepřesné. Ve většině případů tedy zobrazujeme kaustiky zvlášť pomocí fotonové mapy kaustik (viz. kapitola 5.5.4).

Na obrázku 7.14 můžeme vidět obrázky vykreslené pomocí fotonové mapy kaustik. Vlevo nahoře jsou kaustiky vykresleny zcela bez filtrů, zatímco na dalších obrázcích jsou aplikovány filtry: kuželový, Gaussův a Simpsonův (Simpsonovo jádro). V tomto případě nedochází k výraznému zlepšení, protože je poloměr (1,0) a počet fotonů (50 000) ve fotonové mapě relativně malý.

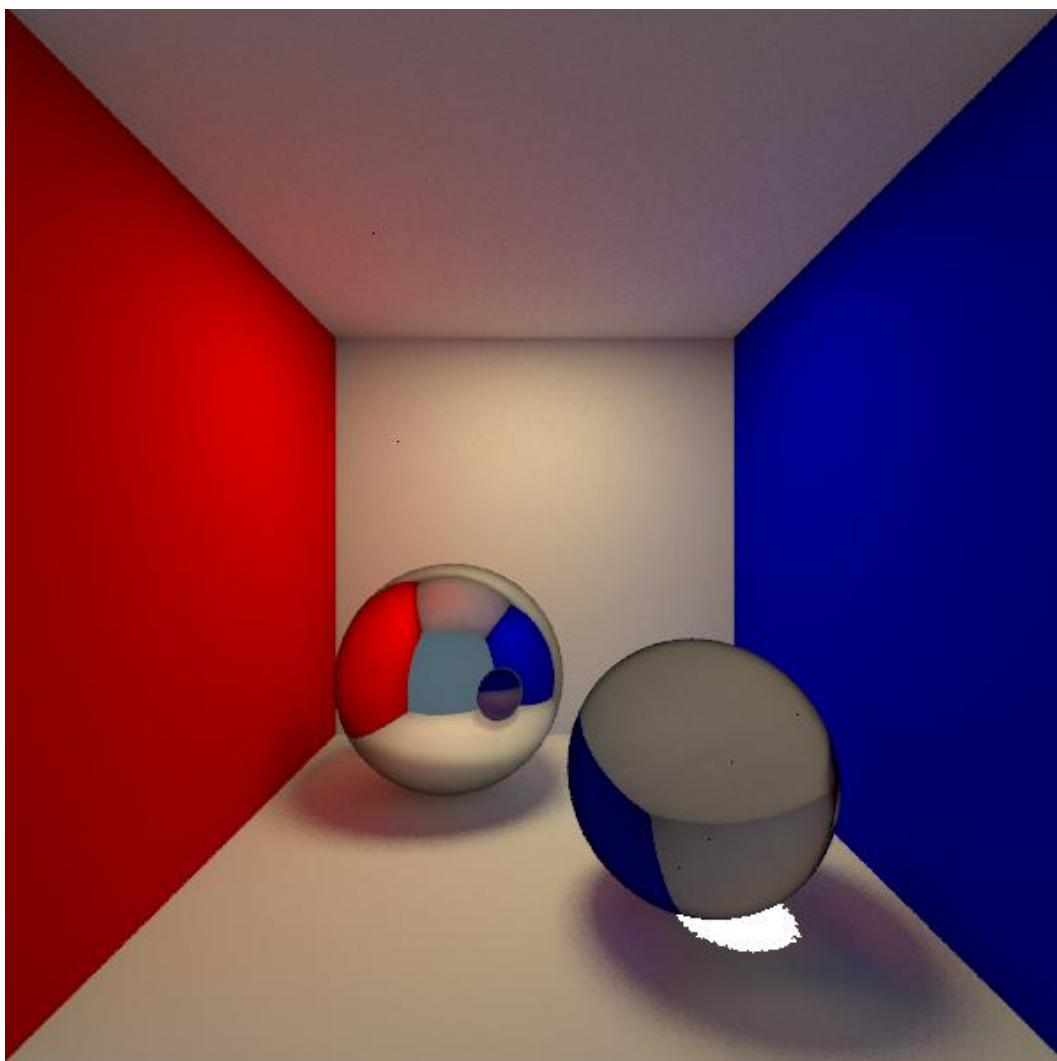
Největší využití filtrů je, pokud se ve scéně nachází komplikované kaustiky a potřebujeme zajistit ostré hrany.

7.5 Kompletní řešení – ukázka

Na obrázku 7.15 je vidět scéna, která kombinuje přímé i nepřímé osvětlení a kaustiky. Přímé osvětlení je počítáno pomocí sledování paprsku (10 paprsku skrze pixel, 10 stínových paprsků). Nepřímé osvětlení využívá finálního sbírání (celkem asi 200 paprsků na pixel) a kaustiky jsou vypočítány pomocí fotonové mapy kaustik (100 000 fotonů). Celý výpočet trval necelých 7 hodin.



Obrázek 7.14: Kaustiky zobrazené pomocí fotonové mapy kaustik.



Obrázek 7.15: Kompletní řešení zobrazovací rovnice obsahující přímé i nepřímé osvětlení a kaustiky.

7.6 Shrnutí

Tato kapitola potvrdila kvality fotonových map a vůbec zkraslených metod. Jejich využití je především na rychlý a přesný výpočet nepřímého osvětlení, se kterým mohou mít některé nestranné metody problémy. Vizualizace globální fotonové mapy by měla být nepřímá, tedy pomocí finálního sbírání, které zajistí „hladké“ nepřímé osvětlení. Přímé zobrazování může být použito pro rychlý (a spíše orientační) výpočet osvětlení, případně jako odhad nepřímého osvětlení pro sledování paprsku.

Ještě podstatnější roli hrají fotonové mapy při výpočtu kaustik, protože jsou často jedinou metodou, která je schopná zvládnout složité kaustiky. Ostré hrany kaustik mohou být zajištěny použitím filtrů.

Kapitola 8

Závěr a budoucí práce

V této práci byla představena moderní metoda na výpočet globálního osvětlení – fotonové mapy. V úvodních partiích byly nastíněny cíle a omezení fotorealistickej syntézy obrazu, které byly nezbytné pro pochopení dalších kapitol. Následuje úvod do globálního osvětlení. Počínaje krátkou specifikací světla v počítačové grafice a definicí základních radiometrických veličin. Následuje popis klíčových termínů globálního osvětlení. Jako první je popsána dvousměrová distribuční funkce popisující daný materiál. Dualita radiance a potenciálu je klíčovým tématem druhé kapitole, protože na této dualitě jsou založeny všechny moderní zobrazovací algoritmy.

Další kapitola byla věnována Monte Carlo metodám. Na počátku třetí kapitoly byly popsány výhody Monte Carlo metod a důvody proč jsou ve fotorealistickej syntéze obrazu základní principy Monte Carlo integrace. Následně byly aplikovány jednotlivé techniky na výpočet rozptylu, vysvětlen jejich princip a dokázáno snížení rozptylu výsledného řešení. Monte Carlo metody byly použity na řešení Fredholmovy rovnice druhého řádu pomocí náhodné procházky, jenž byla ukončena ruskou ruletou.

Čtvrtá kapitola se zabývala aplikací Monte Carlo metod na problém globálního osvětlení. V úvodu byly představeny téměř všechny metody zabývající se globálním osvětlením. Následně byly odvozeny dva algoritmy využívající na výpočet zářivého výkonu procházejícího pixelem odlišné rovnice – sledování cesty a sledování světla. Na oba dva byly aplikovány techniky na snížení rozptylu z předcházející kapitoly.

Pátá část představila téměř všechny aspekty fotonových map, včetně porovnání nestranných a zkreslených metod. Úvodem se zabývá prvním krokem fotonových map – vytvořením a naplněním fotonové mapy. Následně i druhým krokem fotonových map – vizualizací fotonových map. Zabývá se problémem odhadu hustoty irradiance.

V šesté kapitole byly shrnuty nejdůležitější rozšíření algoritmu fotonových map, včetně několika nejnovějších. Poměrně podrobně rozebírá jejich principy, využití i problémy.

Sedmá část je čistě praktická. Byly představeny různé možnosti při výpočtu rozličných druhů osvětlení – přímého, nepřímého a kaustik. Ukazuje problémy jednotlivých řešení a nastiňuje, jak takové situace řešit. Úzce se také zabývá porovnáním nestranných a zkreslených Monte Carlo metod. Porovnává sledování paprsku a fotonové mapy. Poukazuje na problémy jednotlivých řešení a dokazuje, že zkreslené Monte Carlo metody jsou pro praktické využití daleko lepší, často dokonce nezbytné.

Dalším významným příspěvkem práce je implementace v knihovně JaGrLib. Zaměřili jsme se především na vytvoření maximálně modulární architektury pro světelné zdroje, materiály a jednotlivé algoritmy pro vykreslování. Výsledná architektura splnila naše cíle a díky ní, je teď v JaGrLib přidávání jednotlivých světelných zdrojů, materiálů i dalších algoritmů týkajících se globálního osvětlení a vykreslování, daleko jednodušší. Také bylo implementováno několik světelných zdrojů a řada materiálů, včetně několika mikroploškových. Hlavně byla ale implementována plně funkční metoda fotonových map (včetně několika rozšíření) a metoda sledování paprsků. Také byly refaktorizovány již implementované algoritmy vrhání paprsku a sledování paprsku, aby podporovali nové světelné zdroje a materiály.

Shrnutí

Práce velice podrobně rozebírá fotorealistickou syntézu – od základů globálního osvětlení až po fotonové mapy. Je tedy velice výhodná pro výuku fotorealistické syntézy obrazu, například i pro naprosté nováčky v oboru, kteří potřebují vysvětlit i základní pojmy. Poskytuje také veškeré informace nutné o Monte Carlo metodách pro potřeby počítačové grafiky i odvození základních algoritmů – sledování cesty a sledování světla. Srozumitelně popisuje problematiku fotonových map, která následně ověřuje i praktickými testy.

Samotná implementace je dobrou ukázkou, jak by měla vypadat kvalitní modulární architektura pro algoritmy globálního osvětlení. Vše je snadno rozšiřitelné, takže se může programátor soustředit pouze na jeden problém (třeba nový materiál) a nemusí řešit nic jiného. Po aplikaci dalších urychlení bude velice snadné vykreslovat obrázky pro výuku.

8.1 Budoucí práce

8.1.1 Progresivní fotonové mapy

Fotonové mapy jsou v současné době velice podstatné téma a stále se objevují nová rozšíření, která většinou zlepšují vizuální kvalitu či umožňují urychlení. Progresivní fotonové mapy jsou asi nejnovějším rozšířením, a proto nabízí prostor pro výzkum. Především se jedná o:

- Zjistit optimální hodnoty pro některé parametry.
- Kritérium, které by bylo schopné zastavit vykreslování, jakmile by bylo dosaženo požadované přesnosti. V souvislosti s tímto je nutné nějakým způsobem odhadnout rozptyl řešení pomocí fotonové mapy poblíž nějakého bodu.
- Prozkoumat adaptivní sledování fotonů, které by mohlo být, díky iteračnímu procesu střílení fotonů, velice účinné.

8.1.2 Opticky aktivní média

Opticky aktivní média (*participating media*) jsou částičky, které nějakým způsobem narušují jinak „poklidný“ pohyb fotonů a paprsků scénou. Může se jednat o mlhu, kouř, oheň a mnoho dalších věcí. Tento případ transportu světla způsobuje, že hodně zjednodušení, které jsme v této práci používali, už neplatí. Samotná zobrazovací či potencionální rovnice tento druh transportu nepopisuje, je potřeba ji zobecnit. Je totiž nutné počítat rozptyl světla i při cestě paprsku skrze prostředí, nejen při interakci světla s materiály.

8.1.3 Poloprůhledné materiály

Poloprůhledné materiály (*translucent materials*) může být v češtině poměrně matoucí termín, protože se jedná o materiály s vnitřním rozptylem (*subsurface scattering*). V této práci jsme používali materiály, které dopadající paprsek odrazili ze stejného bodu, což je pro mnohé materiály v pořádku. Nicméně existují i materiály, které jsou složitější, a které se tímto způsobem nedají napodobit, protože neodrazí paprsek ze stejného bodu. Příkladem takových materiálu je například lidská kůže, rostliny nebo i obyčejný mramor.

V praxi se tyto materiály chovají jako opticky aktivní média, proto je potřeba vyřešit stejnou rovnicí jako v případě opticky aktivních materiálu.

Dodatek A

Modely odrazivosti

Kromě metody fotonových map a dalších algoritmů založených na Monte Carlo metodách se tato práce zabývá modely odrazivosti, proto zde v dodatku ve stručnosti popíšeme některé materiály, kterými jsme se při implementaci zabývali.

A.1 Lambertův model

Lambertův model odrazivosti je jeden z nejjednodušších, protože modeluje ideální difuzní materiál, který rozptyluje světlo do všech směrů. Ačkoliv není fyzikálně korektní, může být dobrou aproximací pro mnoho reálných materiálů (především pro matné a hladké materiály).

Lambertův zákon říká, že intenzita odcházejícího světla je úměrná $\cos \alpha$, kde α je úhel mezi normálou a odcházejícím směrem. Hemisféru tedy budeme vzorkovat podle PDF:

$$p(x) = \cos \alpha$$

Odrazivost tedy bude:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = \frac{\rho}{\pi}$$

Kde

- ρ = barva materiálu v intervalu $\langle 0,1 \rangle$

A.2 Fresnelovy rovnice

Fresnelovy rovnice popisují množství světla odraženého od povrchu. Jsou řešením Maxwellových rovnic na hladkých površích. Rozlišujeme dva druhy Fresnelových rovnic, první je určen pro dielektrika (materiály, které nevedou elektrický proud) a druhé pro vodiče. Každá z těchto dvou rovnic má ještě dvě formy, které závisí na polarizaci světla. V počítačové grafice se většinou předpokládá, že světlo je kruhově polarizováno. S tímto zjednodušením je výpočet Fresnelových rovnic jednoduchý.

Odrazivost můžeme pomocí Fresnelových rovnic vypočítat (s použitím předchozích zjednodušení):

$$r = \frac{1}{2}(r_{\parallel}^2 + r_{\perp}^2) \quad (\text{A.1})$$

Kde:

- r_{\parallel}^2 = paralelní polarizační termín
- r_{\perp}^2 = rovnoběžný polarizační termín

Tyto termíny se dají pro dielektrika vypočítat jako:

$$r_{\parallel} = \frac{\eta_t(\vec{N} \cdot \vec{\omega}_o) - \eta_o(\vec{N} \cdot \vec{\omega}_t)}{\eta_t(\vec{N} \cdot \vec{\omega}_o) + \eta_o(\vec{N} \cdot \vec{\omega}_t)} \quad (\text{A.2})$$

$$r_{\perp} = \frac{\eta_o(\vec{N} \cdot \vec{\omega}_o) - \eta_t(\vec{N} \cdot \vec{\omega}_t)}{\eta_o(\vec{N} \cdot \vec{\omega}_o) + \eta_t(\vec{N} \cdot \vec{\omega}_t)} \quad (\text{A.3})$$

Kde:

- η_o = index refrakce pro vstupní směr
- η_t = index refrakce pro lomený směr
- $\vec{\omega}_o$ = vstupní směr
- $\vec{\omega}_t$ = lomený směr

Lomený směr $\vec{\omega}_t$ by měl být vypočten pomocí Snellova zákona lomu:

$$\eta_i \sin \theta_i = \eta_t \sin \theta_t$$

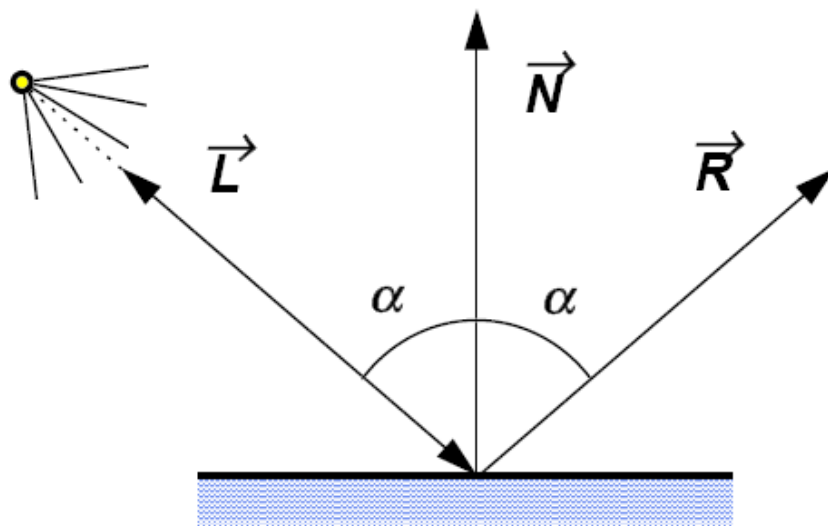
Polarizační termíny Fresnelova rovnice pro vodiče jsou:

$$r_{\parallel}^2 = \frac{(\eta^2 + k^2)(\vec{N} \cdot \vec{\omega}_i)^2 - 2\eta(\vec{N} \cdot \vec{\omega}_i) + 1}{(\eta^2 + k^2)(\vec{N} \cdot \vec{\omega}_i)^2 + 2\eta(\vec{N} \cdot \vec{\omega}_i) + 1} \quad (\text{A.4})$$

$$r_{\perp}^2 = \frac{(\eta^2 + k^2) - 2\eta(\vec{N} \cdot \vec{\omega}_i) + (\vec{N} \cdot \vec{\omega}_i)^2}{(\eta^2 + k^2) + 2\eta(\vec{N} \cdot \vec{\omega}_i) + (\vec{N} \cdot \vec{\omega}_i)^2} \quad (\text{A.5})$$

Kde:

- k = koeficient absorpce
- η = index refrakce vodiče



Obrázek A.1: Spekulární odraz paprsku od materiálu (PE).

A.3 Spekulární odraz

Ideální spekulární odraz má pro daný přicházející směr pouze jeden odcházející směr, kam se světlo může odrazit. Pro všechny ostatní směry je BRDF nulové; hodnota BRDF je tedy pro ideální směr odrazu nekonečná. Popsat takovouto funkci můžeme matematicky pomocí funkcí δ (Diracovo delta distribuce) – viz. (4).

Odražený paprsek lze vypočítat jako:

$$\vec{\omega}_o = 2(\vec{N}, \vec{\omega}_i)\vec{N} - \vec{\omega}_i$$

Jaká část dopadající radiance se odrazí lze popsat jako:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = r * \rho$$

Kde:

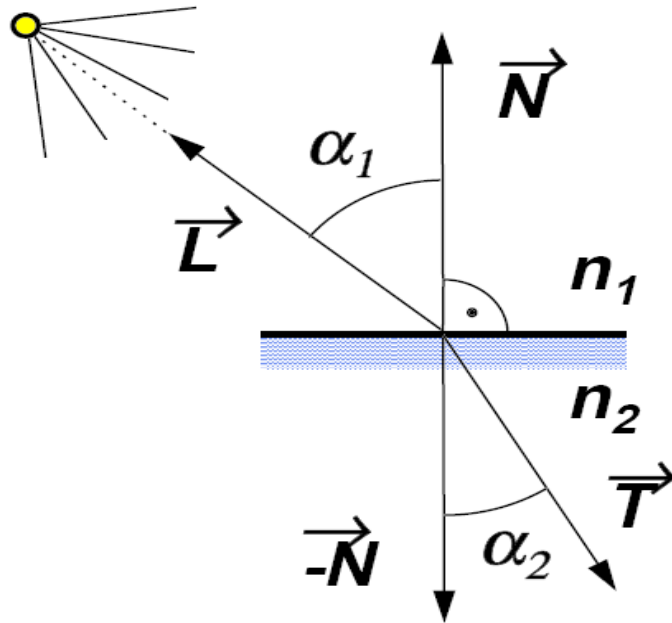
- r = Fresnelův člen (popsán rovnicí A.1, kde jsou za polarizační termíny dosazeny rovnice A.4 a A.5 – termíny pro vodiče)
- ρ = barva materiálu v intervalu $\langle 0,1 \rangle$

A.4 Spekulární lom

Stejně jako ideální odraz i ideální lom se odráží pouze do jediného směru; hodnota BRDF je pro jeden směr nekonečná a pro ostatní nulová.

Odražený paprsek lze vypočítat podle Snellova zákona lomu

$$\eta_i \sin \theta_i = \eta_t \sin \theta_t$$



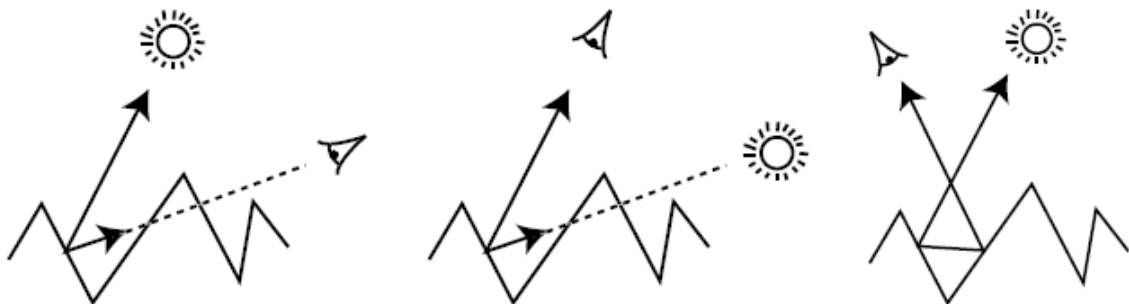
Obrázek A.2: Spekulární lom paprsku (PE).

Množství odraženého světla je popsáno stejnou rovnicí jako v případě ideálního spekulárního odrazu:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = r * \rho$$

Kde:

- r = Fresnelův člen (popsán rovnicí A.1, kde jsou za polarizační termíny dosazeny rovnice A.2 a A.3 – termíny pro dielektrika)
- ρ = barva materiálu v intervalu $\langle 0,1 \rangle$



Obrázek A.3: Mikroploškový materiál. Převzato z (4).

A.5 Oren-Nayar model

Ideální Lambertův difuzní model je nevhodný pro hrubé či zrnité povrchy (např. písek). Čím hrubší je povrch, tím je tmavší oproti hladkému povrchu se stejnými optickými vlastnostmi. Navíc, povrch se jeví světlejší, čím více se směr pohledu přibližuje směru dopadu světla.

Tento model patří mezi takzvané mikroploškové modely, které moduluji libovolný povrch jako sadu malých mikroplošek. Povrch si tedy můžeme představit jako výškové pole (obrázek A.3). Přesný popis tohoto pole by byl poměrně náročný¹, nehledě na výpočetní náročnost, proto používáme v praxi povrch idealizovaný, který je naprosto rovný a samotnou „hrbolatost“ simulujeme statisticky.

Model by měl podporovat stínování, maskování a mnohonásobné odrazy (obrázek A.3) mezi mikroploškami. Jelikož neexistuje uzavřená forma řešení, použijeme aproximaci:

$$f_r(\vec{\omega}_i, \vec{\omega}_o) = \frac{\rho}{\pi} (A + B \max(0, \cos(\phi_i - \phi_o)) \sin \vec{\omega}_+ \tan \vec{\omega}_-)$$

Kde:

- $A = 1 - \frac{\sigma^2}{2(\sigma^2 + 0,33)}$
- $B = \frac{0,45\sigma^2}{\sigma^2 + 0,09}$
- σ = „hrubost“ povrchu – střední hodnota odchylka normály mikroplošky (v radiánech)
- $\omega_+ = \max(\theta_i, \theta_o)$
- $\omega_- = \min(\theta_i, \theta_o)$

¹ Oren-Nayar model je v tom ohledu na popis poměrně jednoduchý, protože popisuje povrch jako několik stejně velkých věček. Obecně si však pod mikroploškovým materiálem představíme povrch, který má každý V-útvary jiný (viz. obrázek A.3).

Dodatek B

Implementace

B.1 Knihovna JaGrLib

JaGrLib (Java Graphics Library) je knihovna (obecný modulární framework) použitelná pro vykreslování. Původně byla navržena především jako výuková knihovna pro počítačovou grafiku a kompresi dat, speciálně na obecnou počítačovou grafiku podle osnov CGG (Computer Graphics Group – skupina počítačové grafiky) na MFF UK (Matematicko-Fyzikální Fakultě Univerzity Karlovy). Supervizor JaGrLib je RNDr. Josef Pelikán.

Knihovna je maximálně modulární. Výsledná aplikace se skládá z několika vzájemně propojených modulů. Samotný modul je potomek třídy Piece. Každý modul má minimálně jednu zásuvku (Plug), do které lze připojit jiný modul. Zásuvek může být i více a lze je rozdělit na vstupní a výstupní. Nezávisle na tom má každá zásuvka nadefinovaný typ připojení (pomocí interface). Vzájemné propojení dvou kompatibilních zásuvek je realizováno kanálem (Channel). Směr šipky udává směr volání metod (v současnosti použito pouze jednosměrné volání). Více informací lze nalézt na domovských stránkách JaGrLib (52).

B.2 Reprezentace dat

Data byla reprezentována jako CSG (*constructive solid geometry*), jenž se skládá z elementárních geometrických těles jako například kvádrů, koule, válce, kužele a dalších. Pokud chceme vytvořit složitější těleso, musíme použít množinové operace (sjednocení, průnik, rozdíl a další) mezi elementárními tělesy. Tělesa mohou být transformována pomocí transformačních matic.

Bylo vytvořeno mnoho scén, z nichž většina byla založena na oblíbené Cornell box, jenž byla v roce 1984 vytvořena na Cornellově univerzitě jako standardní testovací scéna pro globální osvětlení. Na testování byly použity tyto scény:

- Prázdna Cornell box – CBEmpty.jgl
- Cornell box obsahující dvě difuzní koule – CBSphereDiffuse.jgl
- Cornell box obsahující dvě spekulární koule – CBSphereSpecular.jgl

- Cornell box testující dané materiály:
 - CBDCookTorrance.jgl
 - CBDFresnelTest.jgl
 - CBDOrenNayar.jgl
 - CBSphereRefraction.jgl a další
- Cornell box testující dané světelné zdroje:
 - CBDDirectionalLight.jgl
 - CBDPointLight.jgl
 - CBDSpotLight.jgl

B.3 Architektura

Architektura byla během tvoření této práce výrazně předělána. Na prvním místě byla snaha udělat interfací co nejobecnější, aby bylo možné snadno přidávat nové světelné zdroje, materiály a další algoritmy založené na sledování paprsku. Zde je seznam interfací, které byly změněny či přidány:

- *BSSRDF*

V původní architektuře byl rozdíl mezi materiálem a modelem odrazivosti. Model odrazivosti popisoval, jak vypočítávat osvětlení v daném bodě a materiál specifikovat konkrétní hodnoty. V praxi to tedy vypadalo, že se ve scéně nadefinovaly modely odrazivosti a následně několik materiálů, jež byly vlastně instancemi daných modelů odrazivosti.

Nyní jsme tyto dva interfací sloučili do jednoho – BSSRDF. Název BSSRDF (namísto BRDF), protože by interface mělo podporovat i poloprůhledné materiály s rozptylem dovnitř materiálu, které popisuje právě BSSRDF. Teď si tedy pod slovem materiál, můžeme představit model odrazivosti s konkrétními hodnotami. Díky tomuto lze při tvorbě scén materiály snadno vytvořit, protože není nutné předtím definovat model odrazivosti. Implementace je také trochu jednodušší, protože jsou algoritmy a proměnné uloženy v jedné třídě a není nutné je získávat z nějaké jiné.

- *Fresnel*

Pro složitější materiály je potřeba aproximovat Fresnelovi rovnice. Existuje několik druhů aproximací pro různé druhy materiálů (viz. kapitola A.2).

- *LightMap*

LightMap (světelná mapa) je nové interface, které do knihovny vkládá podporu různých světelných map, které jsou dnes použity ve většině novějších algoritmů.

- *LightTracing*

Jednorázové interface, které slouží pro podporu algoritmů založených na sledování světla.

- *LightSource*

Interface reprezentující světelný zdroj bylo zobecněno. Světelný zdroj se teď umí vzorkovat a to jak v případě, kdy je potřeba vzorkovat zdroj vzhledem k výpočtu osvětlení v nějakém bodě, tak při vzorkování světelného zdroje i jeho směrů (tj. například při sledování fotonů).

- *MicrofacetDistribution*

Interface pro mikroploškové materiály (kapitola A.6).

B.4 Moduly

Bylo také implementováno mnoho modulů a mnoho dalších modulů bylo refaktorizováno, aby podporovali nový systém materiálů či světelných zdrojů. Zde je seznam modulů, které byly nějak výrazněji upraveny (menší úpravy byly provedeny i v dalších modulech, které zde nejsou zmíněny). Zcela nové moduly mají na konci svého stručného popisu napsáno v závorce napsáno „nový“, aby bylo možné snadno rozlišit, jaké moduly byly vytvořeny a jaké pouze upraveny.

- *AnisotropicMicrofacet*

Anizotropický mikroploškový model, který popisuje interakci světelných paprsků s mikroploškami (nový).

- *BlinnMicrofacet*

Blinnův mikroploškový model popisující interakci paprsku s mikroploškami. Je využíván některými modely odrazivosti: Torrance-Sparrow, Cook-Torrance (nový).

- *DirectionalLightSource*

Směrový světelný zdroj, který svítí uniformně v jediném směru (nový).

- *DirectLightMapVisualizer*

Modul, který je schopen přímo zobrazit libovolnou světelnou mapu pomocí jednoduchého sledování paprsku (nový).

- *FinalGathering*

Algoritmus speciálně určený pro nepřímé zobrazování světelných map (nový). Navržen především pro spolupráci s fotonovými mapami.

- *FresnelConductor*

Fresnelovy rovnice pro vodiče (nový).

- *FresnelDielectric*

Fresnelovy rovnice pro dielektrika – nevodiče (nový).

- *LambertianModel*

Lambertův model reprezentující ideálně difuzní povrch (nový).

- *MicroFacet*

Důležitá třída reprezentující „mikroplošku“. Nemá nic společného a mikroploškovými modely odrazivosti. Slouží především k ukládání důležitých dat týkajících se průsečíku paprsku a scény.

- *MonteCarlo*

Třída obsahující všechny funkce týkající se Monte Carlo, především vzorkování různých útvarů či rozličné heuristiky (nový).

- *OrenNayarModel*

Reprezentuje mikroploškový model odrazivosti OrenNayar (nový) na modelování hrubých matných materiálů

- *PathTracing*

Sledování cesty pro zobrazování libovolných scén s řadou vylepšení (nový).

- *PhotonMap*

Třída reprezentující fotonovou mapu jako KD-strom. Modul je napsaný maximálně obecně, lze ho tedy použít na reprezentaci téměř libovolné fotonové mapy (nový).

- *PhotonTracing*

Algoritmus sledování fotonů, který slouží jako první část metody fotonových map (nový).

- *SpecularReflection*

Spekulární odraz (nový).

- *SpecularTransmission*

Spekulární lom (nový).

- *StaticAreaLightSource*

Statický plošný světelný zdroj ve tvaru čtyřúhelníka (nový).

- *StaticSphericalLightSource*

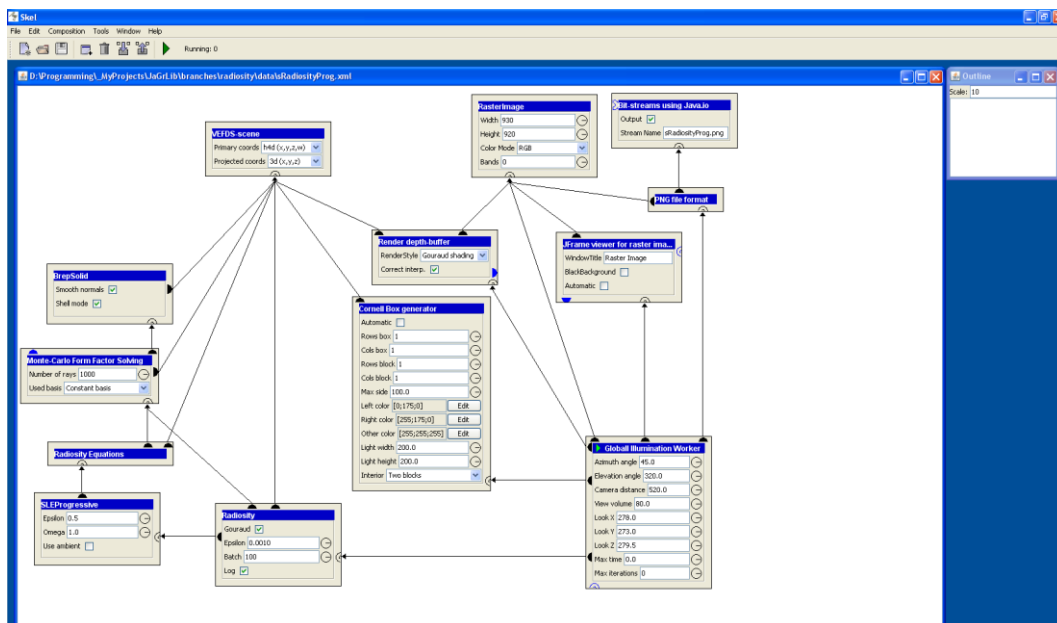
Statický kulový světelný zdroj (nový).

- *StaticSpotLightSource*

Statický bodový světelný zdroj, který byl přizpůsoben nové architektuře.

Díky podstatným změnám v architektuře, které proběhly až v závěru psaní této práce, nebylo možné stihnout do požadovaného termínu implementovat všechny představené metody a rozšíření. Některé starší scény a zapojení nemusí být funkční. Některé moduly nemusí být využity ani v jednom zapojení.

Zdrojové kódy jsou uloženy na přiloženém CD nosiči. Je však doporučeno stáhnout nejnovější zdrojové kódy ze stránky (53) pomocí protokolu SVN (sub-version).



Obrázek B.1: Rozhraní Skel.

B.5 Grafické rozhraní

Grafické rozhraní (GUI – Graphics User Interface) v knihovně JaGrLib je velmi intuitivní, jak je ostatně vidět i na obrázku B.1. V každém zapojení (aplikaci) lze najít několik viditelných modulů, jenž jsou logicky uskupeny², z nichž většina obsahuje nějaké proměnné vlastnosti, kterou jsou různého typu. Ty může uživatel lehce změnit na libovolnou hodnotu, kterou dovolují omezení dané proměnné, a ovlivnit tím následné vykreslování.

² Vpravo dole bývá worker, jenž řídí aplikaci, nad ním moduly zaručující vykreslení a následný výstup, vlevo nahoře data a vlevo dole moduly specifické pro konkrétní aplikaci. Žádný takový standart pro rozložení modulů v JaGrLib není, ale já jsem se touto pomyslnou konvencí řídit a téměř všechna existující zapojení mají víceméně tuto strukturu.

Dodatek C

Obsah CD

JaGrLib

Větev radiosity z grafické knihovny JaGrLib.

data

Existující zapojení modulů, scény a OBJ soubory.

lib

Knihovny potřebné k přeložení a spuštění projektu.

obscure

Moduly patřící do balíčku obscure.

reg

Registrované moduly.

src

Zdrojové kódy.

Doc

Text práce a programátorská dokumentace.

Images

Obrázky vykreslené v jednotlivých testech.

readme.txt

Podrobný popis obsahu.

Bibliografie

1. **Žára, J., Beneš, B., Sochor, J., Felkel, P.** *Moderní počítačová grafika (2. vydání)*. Computer Press, 2005.
2. **Philip Dutre, Kavita Bala, Philippe Bekaert and Peter (FWD) Shirley.** *Advanced Global Illumination*. AK Peters, 2006.
3. **Glassner, Andrew S.** *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995.
4. **Humphreys, G. and Pharr, P.** *Physically Based Rendering : From Theory to Implementation*. Morgan Kaufmann, 2004.
5. **Wallace, M. F. a Cohen J. R.** *Radiosity and Realistic Image Synthesis*. Academic Press, 1993.
6. **Jensen, H. W.** *Realistic Image Synthesis Using Photon Mapping*. A K Peters, 2001.
7. **Kajiya, J. T.** *The rendering equation*. SIGGRAPH '86 Proceedings, 1986.
8. **Smits, B., Arvo, J., and Salesin D.** *An importance-driven radiosity algorithm*. 1992.
9. **Mudur, S. Pattanaik and S.** *Adjoint equations and random walks for illumination computation*. 1995.
10. **S., Mudur and S. Pattanaik.** *Efficient potential equation solutions for global illumination computation*. 1993.
11. **Mudur, S. Pattanaik and S.** *The potential equation and importance in illumination computations*. 1993.
12. **Pattanaik, S.** *Computational Methods for Global Illumination and Visualisation of Complex 3D Environments*. PhD thesis. Pilani, India : 1993.
13. **Pattanaik, S.** *The mathematical framework of adjoint equations for illumination computation*. Bombay, India : 1993.
14. **Rajeev, M., Prabhakar R.** *Randomized Algorithms*. Cambridge : Cambridge University Press, 1995.
15. **Halton, John H.** *A retrospective and prospective of the Monte Carlo method*. SIAM Review, 1970.
16. **Handscomb, J. Hammersly and D.** *Monte Carlo Methods*. London : Chapman and Hall, 1964.
17. **Shreider, Y.** *The Monte Carlo Method*. Oxford : Pergamon Press, 1966.

18. **Whitlock, M. Kalos and P.** *Monte Carlo Methods*. New York : Wiley & Sons, 1986.
19. **Rabinowitz, Ph. Davis and Ph.** *Methods of Numerical Integration*. Orlando : Academic press, 1984.
20. **Yakowitz, Sidney J.** *Computational Probability and Simulation*. New York : Addison-Wesley, 1977.
21. **Veach, E.** *Robust monte carlo methods for light transport simulation. PhD thesis*. CA, USA : 1998.
22. **Jensen, H. W. a spol.** *State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis*. Siggraph 2001 Course 29, 2001.
23. **Whitted, T.** *An improved illumination model for shaded display*. 1980.
24. **Glassner, A. S.** *An Introduction to Ray Tracing*. Morgan Kaufmann, 1989.
25. **Shirley, P., Keith Morley R.** *Realistic Ray Tracing*. A K Peters, Ltd., 2003.
26. **Cook, R., Porter, T. and Carpenter L.** *Distributed ray tracing*. 1984.
27. **Cook, R.** *Stochastic sampling in computer graphics*. 1986.
28. **Greenberg, D. P., Cohen, M., and Torrance, K. E.** *Radiosity: A method for computing global illumination*. 1986.
29. **Šváb, T.** *Radiosita jako metoda výpočtu realistického osvětlení*. Plzeň : 2008.
30. **Gortler, S., Cohen M. F., Slusallek, P.** Radiosity and Relaxation Methods Progressive Refinement is Southwell Relaxation. [Online] 1993. <http://citeseer.ist.psu.edu/645866.html>.
31. **Lafortune, E. and Willems Y.** *Bi-directional path tracing*. Alvor, Portugal : 1993.
32. **Veach, E. and Guibas, L.** *Bidirectional estimators for light transport*. Darmstadt, Germany : 1994.
33. **Veach, E. and Guibas, L. J.** *Metropolis light transport*. 1997.
34. **Jensen, H. W.** *Global illumination using photon maps*. 1996.
35. **John, M.** *Focus on Photon Mapping*. Thomson Course Technology, 2002.
36. **Lafortune, E.** *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. 1996.
37. **Jensen, H. W.** *Importance driven path tracing using the photon map*. 1995.
38. **Shirley, P., Wang, C. and and Zimmerman, K.** *Monte Carlo techniques for direct lighting calculations*. 1996.
39. **Shirley, P., a Wang, C.** *Direct lighting calculation by Monte Carlo integration*. In Second Eurographics Workshop on Rendering, 1991.
40. **Zimmerman, K. and Shirley, P.** *A two-pass solution to the rendering equation with a source visibility preprocess*. 1995.
41. *Data Structures for Range Searching*. **Bentley, J. L, Friedman, J. H.** 1979.

42. *Multidimensional binary search trees used for associative searching.* **Bentley, Jon Louis.** 1975.
43. *Extended photon map implementation.* **Pharr, Matt.** 2005.
44. **Arvo, J. and Kirk, D. B.** *Unbiased sampling techniques for image synthesis.* 1991.
45. **Jensen, H. W. and Tamstorf, R.** *Adaptive sampling and bias estimation in path tracing.* 1997.
46. **Ward, G. J., Rubinstein, F. M., Clear R. D.** *A Ray Tracing Solution for Diffuse Interreflection.* 1988.
47. **Ward, G. J., Heckbert, P. S.** *Irradiance Gradients.*
48. **Christensen, P. H.** *Faster Photon Map Global Illumination.* 1999.
49. *Efficiently rendering shadows using the photon map.* **Jensen, H. W., Christensen, N. J.** 1995.
50. *Fast Final Gathering via Reverse Photon Mapping.* **Havran, V., Herzog, R., Seidel, H. P.** 2005.
51. *Progressive Photon Mapping.* **Jensen, H. W., Hachisuka, T., Ogaki, S.** 2008.
52. **Pelikán, Josef.** JaGrLib. *JaGrLib - library for computer graphics.* [Online] <http://cgg.ms.mff.cuni.cz/JaGrLib/>.
53. **Pelikán, Josef.** [Online] <svn://cgg.mff.cuni.cz/JaGrLib/>.
54. **Puech, François X. Sillion and Claude.** *Radiosity and Global Illumination.* Morgan Kaufmann, 1994.
55. **James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes.** *Computer Graphics: Principles and Practice in C (2nd Edition).* Addison-Wesley Professional, 1995.
56. **Howel, J. R.** *A Catalog of Radiation Configuration Factors.* New York : McGraw, 1982.
57. **Siegel, R., a Howell, J. R.** *Thermal Radiation Heat Transfer, 3rd Edition.* New York : Hemisphere Publishing Corporation, 1992.
58. **Rushmeier, Holly Edith.** ACM - portal - participating media. *Realistic image synthesis for scenes with radiatively participating media.* [Online] <http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=914720>.
59. **Pueyo, X.** *Diffuse interreflection. techniques for form-factor computation:.* 1991.
60. **Warnock, J.** *A hidden-surface algorithm for computer generated halftone.* Technical Report TR 4-15, Dept. of Computer Science, University of Utah, 1969.
61. **Recker, R. J., George, D. W., a Greenberg, D. P.** *Acceleration techniques for progressive refinement radiosity.* Symposium on Interactive 3D Graphics, 1990.
62. **Dippe, M. A. Z., a Wold, E. H.** *Stochastic sampling: Theory and application.* NJ : Ed. Ablex Publishing, 1991.
63. **Lee, M., Redner, R., a Uselton, S.** *Statistically optimized sampling for distributed ray tracing.* SIGGRAPH '85 Proceedings, 1985.

64. **Schumaker, R. B., Brand, M. G., a Sharp, W.** *Study for applying computer-generated images to visual simulation.* U.S. Air Force Human Resources Lab, 1969.
65. **Recker, R. J., George, D. W., a Greenberg, D. P.** *Acceleration techniques for progressive refinement radiosity.* Computer Graphics (Symposium on Interactive 3D Graphics), 1990.
66. **Arvo, J., a Kirk, D.** *Fast ray tracing by ray classification.* Computer Graphics (SIGGRAPH '87 Proceedings), 1987.
67. **Gortler, S., a Cohen, M. F.** *Radiosity and relaxation methods.* Princeton University, 1993.
68. **Shirley, P.** *Fundamentals of Computer Graphics.* A K Peters, Ltd., 2005.
69. **Ashikhmin, M., Shirley P.** *Fundamentals of Computer Graphics.* AK Peters, 2002.
70. **Sillion, F. X., Puech, C.** *Radiosity and Global Illumination.* Morgan Kaufmann, 1994.
71. **University, Cornell.** The Cornell Box. [Online]
<http://www.graphics.cornell.edu/online/box/>.
72. **Cohen, M.** *A radiosity method for the realistic image synthesis of complex diffuse environments.* Program of Computer Graphics, Cornell University, 1985.
73. **Guibas, E. Veach and L.** *Optimally combining sampling techniques for Monte Carlo rendering.* Computer Graphics, vol. 29, pp. 419–428, 1995.
74. **Bouatouch, S. Pattanaik and K.** *Adjoint equations and particle tracing for global illumination.* Rennes Cedex : IRISA, 1995.
75. **Jensen, H. W.** *A Practical Guide to Global Illumination using Photon Mapping.* 2001.