

České vysoké učení technické v Praze
Fakulta elektrotechnická

Vizualizace světelného zatížení

Diplomová práce

Vypracoval: Martin Čadík
Vedoucí práce: Doc. Ing. Pavel Slavík CSc.

Praha 23. ledna 2002

Abstrakt

Vizualizace světelného zatížení je problém z oblasti globálních zobrazovacích metod, cílem je zvýraznění oblastí podle účinku světelného záření na objekty v dané vstupní scéně. Tato práce podává přehled známých vyzařovacích metod, popisuje problémy vyskytující se při implementaci a jejich možná řešení. Daný problém řeší pomocí hierarchické Monte Carlo vyzařovací metody. Součástí práce je popis implementace této metody v programovacím jazyce Java.

Abstract

Visualization of the light load is a global illumination problem, the aim is the marking of the areas with regard to illumination of the objects in the given input scene. This work presents a survey of known radiosity methods and describes problems that occur during the implementation phase with some possible answers. It solves the given problem with help of the hierarchical Monte Carlo radiosity method. The part of the work is the description of the implementation in the Java programming language.

Prohlášení

Prohlašuji, že jsem na této práci pracoval samostatně a že jsem uvedl veškerou literaturu, kterou jsem při práci použil. Souhlasím s tím, aby výsledky mé práce byly volně využívány elektrotechnickou fakultou ČVUT.

Martin Čadík

Poděkování

Moje poděkování patří vedoucímu této diplomové práce, panu Doc. Ing. Pavlu Slavíkovi CSc., za cenné rady a připomínky, které mi pomohly vytvořit tuto práci.

Věnování

Tuto diplomovou práci věnuji svým krásným rodičům, kteří mi vždy byli oporou, nejen po dobu vysokoškolského studia.

Následující rčení mi pomohlo překonat mnoho těžkých překážek na cestě životem, díky za něj, Bobši.

Rozsekej hovno na sračku!
Bobeš Valášek

Obsah

1	Úvod	1
1.1	Popis problému a cíl práce	2
2	Metody výpočtu osvětlení	3
2.1	Radiometrie a fotometrie	3
2.1.1	Prostorový úhel	3
2.1.2	Zářivý tok	4
2.1.3	Zářivost	4
2.1.4	Intenzita ozáření	4
2.1.5	Intenzita vyzařování	4
2.1.6	Měrná zářivost	4
2.2	Základní pojmy	5
2.2.1	Osvětlovací model	5
2.2.2	Metody výpočtu osvětlení	6
2.3	Metody založené na sledování paprsku	6
2.4	Vyzařovací metody	7
2.4.1	Podstata metody	7
2.4.2	Vyzařovací metody	9
3	Techniky vyzařovacích metod	12
3.1	Výpočet konfiguračních faktorů	12
3.1.1	Analytické řešení	13
3.1.2	Výpočet pomocí polokoule	13
3.1.3	Výpočet pomocí polokrychle	14
3.1.4	Výpočet pomocí kubického čtyřstěnu	14
3.1.5	Stochastické řešení	15
3.2	Meshing	15
3.2.1	Diskretizační chyba	15
3.2.2	Vlastnosti sítě	17
3.2.3	Automatická konstrukce sítě	18
3.2.4	Radiozitní textury	19
3.3	Zobrazení výsledku	19
3.4	Interpolace	19
3.4.1	Interpolace C^0	20
3.4.2	Interpolace C^1	21
3.5	Modelace venkovního osvětlení	21

3.5.1	Světelné zdroje	22
3.5.2	Sluneční světlo	22
3.5.3	Světlo oblohy	23
4	Hierarchická a Monte Carlo vyzařovací metoda	25
4.1	Hierarchická vyzařovací metoda	25
4.2	Monte Carlo vyzařovací metody	27
4.2.1	Kvazi Monte Carlo	29
4.2.2	Vzorkování po částech	29
4.2.3	Rovnoměrně rozložené paprsky	29
4.2.4	Stochastická Jakobiho iterační metoda	30
4.3	Hierarchická Monte Carlo vyzařovací metoda	31
4.3.1	Algoritmus hierarchické Monte Carlo metody	33
4.3.2	Potřebný počet vzorků	34
5	Implementace	36
5.1	Java 3D	36
5.1.1	Graf scény	37
5.2	Volba metody výpočtu	38
5.2.1	Vyplývající vlastnosti aplikace	38
5.3	Formát vstupních a výstupních dat	39
5.4	Systém MCrad	39
5.4.1	Struktura systému	39
5.4.2	Reprezentace scény	41
5.4.3	Postup výpočtu	41
5.5	Metoda dělení	42
5.5.1	Lokální topologická informace	42
5.5.2	Eliminace T-vrcholů	42
5.6	Rozšíření metody výpočtu	43
5.6.1	Začlenění zdrojů paralelního světla	43
5.6.2	Zrcadlové odrazy	43
5.6.3	Světelná iterace	44
5.6.4	Akumulace dopadlé energie	44
5.7	Zobrazení	44
5.7.1	Zobrazení sítě	45
5.7.2	Zobrazení vypočtených hodnot	45
5.7.3	Crease angle	45
5.7.4	Transformace hodnot energie na barevnou stupnici	46
6	Výsledky	48
6.1	Použité technické vybavení	48
6.2	Nastavení parametrů metody	48
6.2.1	Nevhodné nastavení parametrů metody	49
6.3	Ilustrační scény	50
6.3.1	Transformace hodnot energie na barevnou stupnici	50
6.3.2	Zobrazení energie a světelného zatížení	50
6.3.3	Zobrazení sítě	50

6.3.4	Postup výpočtu	52
6.3.5	Zrcadlové odrazy	52
6.3.6	Scéna „mrakodrapy“	53
6.3.7	Scéna „ulice“	53
6.4	Global Illumination Test Scenes	54
6.4.1	Emission test	55
6.4.2	Shadow test	55
6.4.3	Large spatial extent	55
6.4.4	Complex illumination paths	55
6.4.5	Geometric accuracy	56
7	Závěr	59
A	Slovníček	64
B	Formát souborů	66
C	Uživatelská příručka	68
C.1	Instalace programu	68
C.2	Parametry příkazového řádku	68
C.3	Ovládání programu	68
C.3.1	Výpočet	69
C.3.2	Zobrazení	70
C.3.3	Uložení výsledku	70
D	Obsah přiloženého CD-ROM	73

Seznam obrázků

2.1	Složky odraženého světla	5
2.2	Měrná zářivost $L(x, \vec{\omega})$ vyzářená z bodu x ve směru $\vec{\omega}$	8
2.3	Postupující radiozita – vystřelování energie	10
3.1	Geometrie pro výpočet konfiguračního faktoru	12
3.2	Nusseltova analogie	13
3.3	Výpočet pomocí polokrychle	14
3.4	Wallancova metoda	15
3.5	Aspect ratio	17
3.6	Algoritmy pro automatickou konstrukci sítě	18
3.7	Porovnání interpolace konstantními (a) a lineárními (b) funkcemi	20
3.8	T-vrchol a jedna z jeho možných eliminací	21
3.9	Geometrie pro výpočet záře bodu na obloze	23
4.1	Interakce v hierarchické struktuře	25
4.2	Geometrie pro odhad konfiguračního faktoru	26
4.3	Hierarchická Monte Carlo vyzařovací metoda	33
5.1	Příklad jednoduché stromové struktury grafu scény	37
5.2	Struktura systému MCrad	39
5.3	Hierarchie plošek vytvářená během výpočtu	41
5.4	Dělení plošky	42
5.5	Zákon odrazu	43
5.6	Interpolace hodnot energie na ploškách	45
5.7	Úhel mezi normálami dvou plošek	46
5.8	Barevný model HLS	47
6.1	Vliv nevhodně nastavených parametrů na kvalitu výsledku	49
6.2	Zobrazení (a) monochromatické (b) pomocí „falešných“ barev	50
6.3	Zobrazení (a) vyzařované energie (b) světelného zatížení	51
6.4	Zobrazení sítě	51
6.5	Výsledná scéna po dokončení (a, d) první iterace (b, e) druhé iterace (c, f) třetí iterace	52
6.6	Zrcadlové odrazy v jednoduchém modelu ulice (a) konstantní pozice Slunce (b) krátká trajektorie Slunce	53
6.7	Global Illumination Test Scenes (a) Emission test (b) Shadow test (c) Large spatial extent (d) Complex illumination paths (e) Geometric accuracy	54
6.8	Ilustrační scéna „mrakodrapy“	57

6.9	Ilustrační scéna „ulice“	58
C.1	Uživatelské rozhraní systému MCrad	69
C.2	Dialog pro nastavení parametrů výpočtu	70
C.3	Světelná analýza	71

Seznam algoritmů

4.1	Hierarchická vyzařovací metoda – vytváření vazeb (refine)	27
4.2	Hierarchická vyzařovací metoda	28
4.3	Regulární stochastická Jakobiho iterační metoda	32
4.4	Hierarchická Monte Carlo metoda – vytváření vazeb (refine)	34
4.5	Hierarchická stochastická Jakobiho metoda	35

Kapitola 1

Úvod

Při architektonickém návrhu např. městské zástavby je vhodné zohlednit světelné vlastnosti a zatížení jednotlivých objektů. V současné době není zpravidla velikou překážkou vytvoření prostorového modelu navrhované zástavby. Na základě tohoto modelu a známých světelných podmínek je poté možné vizualizovat světelné zatížení objektů ve scéně přibližně tak, jaké by bylo ve skutečnosti. Vizualizací světelného zatížení přitom miníme zohlednění množství dopadlé světelné energie na jednotlivé části scény. Metody výpočtu takové vizualizace spadají do oblasti počítačové grafiky, která se nazývá *realistická syntéza obrazu* (angl. *realistic image synthesis*).

Výzkumníci z oblasti počítačové grafiky se již více než třicet let snaží uměle vytvářet obrazy, které mají u pozorovatele vzbudit dojem, že sleduje reálnou scénu. V současné době jsou známy postupy, které jsou zpravidla založeny na fyzikálních základech a produkují velmi kvalitní výsledky. Mnoho problémů není však stále dostatečně uspokojivě vyřešeno, a tak výzkum na poli realistické syntézy obrazu nadále pokračuje.

Zřejmě největším problémem známých metod je vysoká *časová složitost* a z ní plynoucí značné výpočetní nároky na příslušnou hardwarovou platformu. Tento problém je zvláště markantní v případě syntézy scén z exteriéru, které se vyznačují zpravidla komplexní geometrií a složitými světelnými podmínkami. S geometrickou náročností úzce souvisí další problém, kterým je vysoká *paměťová složitost*. Paměťové nároky některých metod jsou natolik vysoké, že shora omezují dosažitelné rozlišení scény.

Jedním z přístupů, jak dosáhnout realitě velmi blízkých výsledků, je *vyzařovací* (radiozitní) *metoda* (angl. *radiosity method*). Tato metoda, původně vystavěná na principech termodynamiky, se snaží fyzikálně správně simulovat interakce světla a zobrazované scény. Výstupem vyzařovací metody je globální popis osvětlení všech těles scény.

V současné době se skutečným scénám velmi blíží také výsledky metod založených na *sledování paprsku* (angl. *ray tracing*). Tyto původně spíše empiricky orientované metody byly rozšířeny o správné uvážení fyzikálních principů.

Cílem této diplomové práce je specifikovat vlastnosti a možné přístupy k řešení scén v exteriéru, prostudovat a popsat známé algoritmy připadající v úvahu a navrhnout a implementovat experimentální systém sloužící k vizualizaci světelného zatížení takových scén.

1.1 Popis problému a cíl práce

Cílem práce je navrhnout a implementovat experimentální systém pro vizualizaci světelného zatížení. *Světelným zatížením* se v rámci této práce míní celkové množství dopadlého zářivého toku na jednotku plochy, neboli akumulovaná hodnota intenzity ozáření plošky.

Vstupem systému bude popis třírozměrné (3D) scény a popis uvažovaných světelných zdrojů. Výstup systému se očekává ve formě popisu vypočtené 3D scény, kterým by uživatel mohl procházet za účelem prezentace apod. Ve výsledku by měly být vizuálně odlišeny oblasti podle světelného zatížení. Primárně bude systém orientován na výpočet venkovních scén.

Venkovní scény se od častěji uvažovaných scén z interiéru liší zejména povahou světelných zdrojů. Hlavními světelnými zdroji venkovní scény jsou Slunce a obloha, které mají výrazně odlišné vlastnosti od běžně uvažovaných difúzních zdrojů. Podrobněji se tímto problémem zabývá sekce 3.5.

Realisticky vypadající obraz světelných poměrů ve scéně lze získat výpočtem pomocí dvou různých základních přístupů, pomocí metody sledování paprsku (angl. ray tracing) nebo pomocí vyzařovací metody (angl. radiosity method). Těmito metodami výpočtu se zabývá kapitola 2.

Při implementaci vyzařovací metody je potřeba často implementovat další techniky, jako např. adaptivní dělení sítě. Pomocnými technikami vyzařovacích metod se podrobně zabývá kapitola 3.

Zajímavé vlastnosti z hlediska výpočtu i výsledku má hierarchická vyzařovací metoda. Deterministická varianta spolu se stochastickou variantou této metody je popsána v kapitole 4.

V kapitole 5 je uveden popis vlastní implementace hierarchické Monte Carlo vyzařovací metody v novém experimentálním systému *MCrad* včetně několika rozšíření metody pro výpočet venkovních scén.

Výsledky implementovaného návrhu, systému *MCrad*, ve formě obrázků uvádí a diskutuje kapitola 6.

Kapitola 2

Metody výpočtu osvětlení

Tato kapitola obsahuje definice používaných fyzikálních veličin, jednotek a základních pojmů z oblasti výpočtu osvětlení. Dále jsou v kapitole popsány různé postupy výpočtu, a to jak pohledově závislé, tak zejména pohledově nezávislé. Důraz je kladen na popis vyzařovacích metod.

2.1 Radiometrie a fotometrie

Radiometrie je vědní disciplína, která se zabývá měřeními elektromagnetické energie v rámci optického spektra. Výsledky radiometrických měření vyjadřují radiometrické veličiny v jednotkách SI pro energii (Joule) a výkon (Watt). *Fotometrie* se zabývá měřením viditelného světla v jednotkách, které uvažují citlivost lidského oka.

Radiometrické a fotometrické veličiny jsou obecně závislé na místě, směru, vlnové délce, čase a polarizaci světla. Za účelem zjednodušení výpočtu se provádí zjednodušení těchto závislostí. Typicky lze s malým újmem na přesnosti zanedbat vliv polarizace a konečné rychlosti šíření světla.

Všechny radiometrické veličiny jsou označovány velkými písmeny se spodním indexem „e“. Odpovídající fotometrické veličiny jsou značeny stejnými písmeny bez tohoto indexu. V dalším textu budou veličiny uváděny bez spodního indexu „e“, neboť vždy půjde o veličiny radiometrické [27].

Velmi přehledný a srozumitelný úvod do radiometrie a fotometrie lze nalézt v [10]. České názvosloví je uvedeno např. v [21].

2.1.1 Prostorový úhel

V definici některých veličin se vyskytuje *prostorový úhel*. Prostorový úhel je definován jako velikost plochy, kterou vytíná obecná kuželová plocha na povrchu jednotkové koule, jejíž střed, *vrchol prostorového úhlu*, je totožný s vrcholem uvažované kuželové plochy. Prostorový úhel se značí ω , jeho jednotkou je *steradian* (*sr*), dříve nazývaný stereoradián.

Ve vektorové formě zobecňuje prostorový úhel $\vec{\omega}$ libovolný směr na množinu směrů kolem tohoto vektoru. Norma $\|\vec{\omega}\|$ vyjadřuje velikost prostorového úhlu, v jehož mezích je celá tato množina směrů uzavřena.

2.1.2 Zářivý tok

Energii značíme Q . Jednotkou energie je *Joule* (J). Energie přenesená elektromagnetickým zářením za jednotku času se nazývá *zářivý tok* (angl. *radiant flux*, resp. *radiant power*),

$$P_e = \Phi_e = \frac{dQ}{dt} \quad [W; J, s].$$

Pokud se v dalším textu hovoří o „energii“, je tím myšlena hodnota právě této veličiny.

Fotometrickou veličinou odpovídající zářivému toku je *světelný tok* (angl. *luminous power*). Fyzikální jednotkou světelného toku je *lumen* (značka lm).

2.1.3 Zářivost

Zářivost (angl. *radiant intensity*) vyjadřuje prostorovou hustotu zářivého toku vyzařovaného bodovým zdrojem,

$$I_e = \frac{d\Phi_e}{d\omega} \quad [W \cdot sr^{-1}; W, sr].$$

Obdobou zářivosti je ve fotometrii *svítivost* (angl. *luminous intensity*). Jednotkou svítivosti je *kandela* (značka cd), jejíž rozměr je $lm \cdot sr^{-1}$. Někdy se ještě používají jednotky starší, jako je *Hefnerova svíčka* ($1 HK = 0,92 cd$) nebo *mezinárodní svíčka* ($1 SI = 1,02 cd$).

2.1.4 Intenzita ozáření

Intenzita ozáření (angl. *irradiance*) popisuje plošnou hustotu zářivého toku dopadlého na plochu dA ,

$$E_e = \frac{d\Phi_e}{dA} \quad [W \cdot m^{-2}; W, m^2].$$

Intenzitě ozáření odpovídá fotometrická jednotka *osvětlení* (angl. *illuminance*). Velikost osvětlení udáváme v jednotce *lux*, která má rozměr $lm \cdot m^{-2}$.

2.1.5 Intenzita vyzařování

Intenzita vyzařování (angl. *radiant exitance*) je obdobou intenzity ozáření. Rozdíl je v tom, že měříme plošnou hustotu zářivého toku *emitovaného* z plochy dA ,

$$B_e = \frac{d\Phi_e}{dA} \quad [W \cdot m^{-2}; W, m^2].$$

Standardizovaný anglický název této veličiny je *radiant exitance*, v počítačové grafice a stejně tak v dalším textu se však používá termín *radiozita* (angl. *radiosity*).

Fotometrickou veličinou odpovídající intenzitě vyzařování je *jas* (angl. *luminosity*). Pro jas existuje více jednotek, jako např. *Stilb* (značka Sb) a *Lambert* (značka $lamb$), ale ve většině případů se dnes jas udává v *luxech*, resp. v $lm \cdot m^{-2}$.

2.1.6 Měrná zářivost

Měrná zářivost (angl. *radiance*), nebo též *zář*, je prostorová a plošná hustota zářivého toku,

$$L_e = \frac{d^2\Phi_e}{d\vec{\omega} d\vec{A}} \quad [W \cdot sr^{-1} \cdot m^{-2}; W, sr^{-1}, m^{-2}]. \quad (2.1)$$

Jak prostorový úhel, tak plocha jsou v uvedené rovnici ve vektorové formě. Skalární součin ve jmenovateli znamená

$$d\vec{\omega} \cdot d\vec{A} = d\omega \, dA \cos \theta,$$

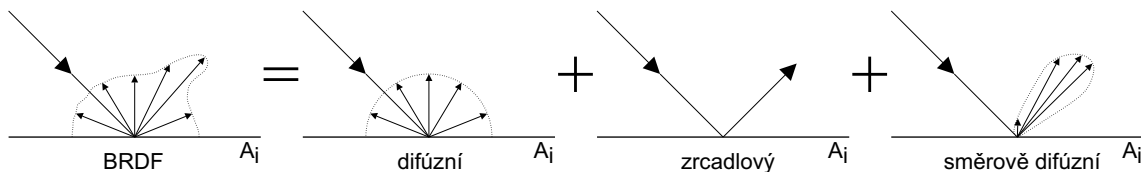
kde θ je úhel mezi osou prostorového úhlu $d\omega$ a normálou plochy dA .

Měrné zářivosti odpovídá ve fotometrii *měrná svítivost* (angl. *luminance*). Jednotkou měrné svítivosti je *nit*, který má rozměr $lm \cdot sr^{-1} \cdot m^{-2}$.

2.2 Základní pojmy

2.2.1 Osvětlovací model

Vlastnosti povrchu tělesa z hlediska vlivu na dopadající světlo umožňuje postihnout *osvětlovací model* [51]. Základem osvětlovacího modelu je *odrazová funkce* (angl. *reflection function*), která vyjadřuje intenzitu paprsku odraženého od povrchu tělesa v závislosti na jeho směru a na směru, intenzitě a vlnové délce paprsku dopadajícího. Nejlépe umožňuje popsat odrazivé vlastnosti plochy dvousměrová distribuční funkce (BRDF, viz níže), často se však používají funkce jednodušší, popisující pouze difúzní, resp. zrcadlový odraz, viz obr. 2.1. Obsáhlý popis odrazových funkcí je uveden v [10].



Obrázek 2.1: Složky odraženého světla

Difúzní odraz

Difúzní odraz (angl. *diffuse reflection*) popisuje odražené světlo, které je nezávislé na směru pohledu. Směr paprsků má při difúzním odrazu náhodný charakter se stejnou pravděpodobností všech směrů. Množství difúzně odraženého světla je závislé pouze na úhlu dopadu.

Zrcadlový odraz

Zrcadlový odraz (angl. *specular reflection*) popisuje světlo směrově odražené od povrchu tělesa. Směrový vektor odraženého paprsku je dán vektorem paprsku dopadlého podle zákona odrazu. Světlo odražené zrcadlově je příčinou vzniku odlesků na zobrazovaném tělese.

Skutečný odraz na povrchu tělesa

Skutečné povrchy odrážejí světlo jak difúzně, tak zrcadlově, přičemž zastoupení jednotlivých složek je odlišné u různých druhů materiálů. Čistě difúzní odraz nastává u ideálně matných materiálů, pouze zrcadlový odraz mají ideální zrcadla. S takovými ideálními povrchy se však ve skutečnosti prakticky nesetkáme.

Nejllepší popis odrazivých vlastností skutečných ploch podává *dvousměrová distribuční funkce* (angl. *BRDF* - *bidirectional reflection distribution function*) [10]. BRDF v bodě $\vec{x}$ je

definována jako poměr odchozí záře L_o ve směru $\vec{\omega}_o$ a intenzity ozáření E ze směru $\vec{\omega}_i$ tohoto bodu:

$$f_r(\vec{x}, \vec{\omega}_o, \vec{\omega}_i) = \frac{L_o(\vec{x}, \vec{\omega}_o)}{E(\vec{x}, \vec{\omega}_i)} = \frac{L_o(\vec{x}, \vec{\omega}_o)}{L_i(\vec{x}, \vec{\omega}_i) \cos \theta_i d\vec{\omega}_i}. \quad (2.2)$$

Obousměrná distribuční funkce popisuje *odraz* světla na ploše. K popisu *lomu* se používá BTDF, *bidirectional transmission distribution function*. K popisu odrazu i lomu zároveň se používá BSDF *bidirectional scattering distribution function*.

2.2.2 Metody výpočtu osvětlení

Metody výpočtu osvětlení lze podle jejich výstupu obecně rozdělit na pohledově závislé a pohledově nezávislé.

Pohledově závislé metody

Pohledově závislé metody jsou takové metody, jejichž výstup je závislý na pozici kamery. Většinou je výstup těchto metod uložen ve formě pouze dvojrozměrného (2D) obrázku.

Pohledově nezávislé metody

Pohledově nezávislé metody jsou takové metody, jejichž výstup je nezávislý na pozici kamery. Výstup těchto metod je typicky uložen jako popis trojrozměrné (3D) scény s definovanými světelnými poměry.

2.3 Metody založené na sledování paprsku

Metody založené na sledování paprsku jsou obecně pohledově závislé metody. Cílem práce je navrhnout systém, jehož výstupem je pohledově nezávislá scéna, a proto budou tyto metody popsány pouze okrajově.

Zpětné sledování paprsku

Metody *sledování paprsku* (angl. *ray tracing*) a *vrhání paprsku* (angl. *ray casting*) jsou ve své původní deterministické formě typické pohledově závislé metody. *Deterministické* sledování paprsku je schopno modelovat pouze přímé osvětlení difúzních povrchů a nepřímé osvětlení povrchů zrcadlových. Podrobný popis metody sledování paprsku včetně metod urychlujících výpočet lze nalézt v [51].

Stochastické zpětné sledování paprsku

Stochastický ray tracing vychází z klasické rekursivní metody sledování paprsku, která je rozšířena o výpočty důležitých světelných příspěvků jako jsou difúzní odrazy a druhotné zdroje. Na rozdíl od původního, deterministického přístupu je při stochastickém řešení totiž zaveden prvek náhodnosti, jehož důsledkem je, že pro stejná vstupní data dostaneme mírně odlišné výsledky. To ovšem správně vystihuje povahu světla, kdy se fotony též šíří náhodně a za pocit stability obrazu vděčíme jejich obrovskému počtu.

Deterministická metoda vysílá tzv. stínový paprsek vždy přímo do středu světelného zdroje, stochastické sledování paprsku naopak vybere náhodný směr, takže paprsek v některých případech dosáhne zdroje, jindy např. opustí scénu. Tento nedeterminismus se ve výsledném obraze projeví jako šum, přičemž další výpočet může skončit mírně odlišně, průměrem z výsledků pak obdržíme výsledek.

Stochastický výpočet založený na vysílání paprsků stejné váhy, jejichž směrnice jsou určeny metodou Monte Carlo, je sice pomalý, ale dle [29] je přesnější než jakákoli jiná zobrazovací metoda včetně metody vyzařovací.

Hybridní *deterministicko - stochastická metoda* je kompromisem mezi rychlostí a přesností výpočtu. Příímá složka (angl. direct component) se počítá pomocí paprsků do náhodných bodů na zdrojích světla. Zrcadlová nepřímá složka (angl. specular indirect component) je vypočtena pomocí paprsků se směrnici určenou metodou Monte Carlo, poslední difúzní složka (angl. diffuse indirect component) je poté získána náhodným vyřešením předchozích dvou složek v dynamicky volených pozicích. Tato metoda je implementována v programu Radiance [29], přičemž je vstupní scéna nejprve předzpracována do struktury oktantového stromu pro zvýšení efektivity výpočtu sledování paprsku.

Path Tracing

Path Tracing popsal Kajiya v [22], podrobný popis metody a návrh implementace lze též nalézt v [46]. Algoritmus metody path tracing je založen na aplikaci Monte Carlo metod. Výpočet je sice z hlediska světelných interakcí správný, nicméně velice časově náročný, neboť spočívá v trasování velmi vysokého počtu paprsků z pozice kamery.

Za účelem zrychlení výpočtu se zavádějí paměťové struktury jako např. fotonová mapa (angl. photon map), do kterých se ukládají vypočtené hodnoty. Hodnoty, které nejsou uloženy v datové struktuře lze ze známých hodnot získat interpolací.

Variantou metody je dvousměrný path tracing (angl. *bi-directional path tracing*), který vysílá paprsky jak ze zdrojů světla, tak z pozice kamery.

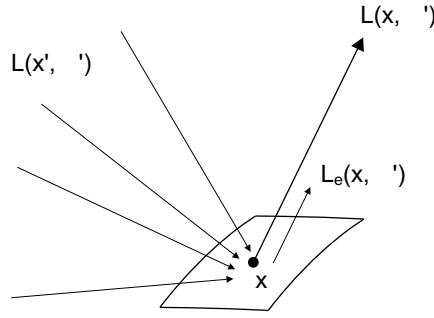
2.4 Vyzařovací metody

Úvod do problematiky vyzařovacích metod lze nalézt v [13, 51], důkladný popis většiny známých vyzařovacích metod a celé oblasti realistické syntézy obrazu v [10].

Poprvé byla vyzařovací metoda v počítačové grafice použita v roce 1984, ale za počátek jejího vývoje lze považovat již rok 1950, kdy se zkoumaly metody přenosu tepla zářením mezi plochami. Postupně byly implementace vyzařovací metody obohaceny o (v původním návrhu zanedbané) jevy jako např. zobrazení odlesků, zrcadlových odrazů, začlenění kouře, mlhy atd. Společným jmenovatelem a výhodou radiozitivních metod je výstup ve formě pohledově nezávislé třírozměrné scény.

2.4.1 Podstata metody

Vyzařovací metoda se snaží řešit tzv. *zobrazovací rovnici* (angl. *rendering equation*), kterou formuloval Kajiya v [22]. Zobrazovací rovnice říká, že zář (měrná zářivost) $L(x, \vec{\omega})$ vyzářená ve směru $\vec{\omega}$ z bodu x na povrchu tělesa je složena z vlastní záře plochy $L_e(x, \vec{\omega})$ v tomto bodě a ze záření dopadajícího ze všech ostatních těles do tohoto bodu a odrážejícího se ve směru $\vec{\omega}$, viz obr. 2.2.



Obrázek 2.2: Měrná zářivost $L(x, \vec{\omega})$ vyzářená z bodu x ve směru $\vec{\omega}$

Pro neprůhledné plochy lze zobrazovací rovnici formulovat jako:

$$L(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_A f_r(x, \vec{\omega}, \vec{\omega}') L(x, \vec{\omega}') G_A(x, x') dA', \quad (2.3)$$

kde A jsou všechny plochy scény, $f_r(x, \vec{\omega}, \vec{\omega}')$ vyhodnocuje osvětlovací model v bodě x , $L(x, \vec{\omega}')$ je zář dopadající do bodu x z bodu x' . Funkce $G_A(x, x')$ zahrnuje viditelnost, odchylky od normál a vzdálenost povrchů obsahujících body x a x' .

Pro povrchy, od nichž se světlo odráží pouze difúzně, lze výpočetně velmi náročnou zobrazovací rovnici zjednodušit na tzv. *radiozitivní rovnici* (angl. *radiosity equation*):

$$B(x) = E(x) + \rho(x) \int_A B(x') G(x, x') dA', \quad (2.4)$$

kde $\rho(x)$ je difúzní odrazivost povrchu, $B(x)$ je radiozita v bodě x a $E(x)$ je vyzářená radiozita v bodě x . Někdy se závislosti $B(x)$ podle vztahu (2.4) říká také *radiozitivní funkce*.

Pokud radiozitivní funkci aproximujeme ploškami, v rámci kterých má radiozita konstantní hodnotu (Lambertovské plochy), lze vztah (2.4) dále zjednodušit [14, 9] na následující systém rovnic:

$$B_i = E_i + \sum_j \rho_i F_{ij} B_j, \quad (2.5)$$

kde F_{ij} je konfigurační faktor mezi ploškami i, j (výpočet konfiguračních faktorů viz kapitola 3.1), ρ_i je difúzní odrazivost plošky i , B_i, B_j jsou radiozity plošek i, j a E_i je vlastní vyzářená radiozita plošky i (nenulová v případě světelných zdrojů).

Tento systém rovnic lze řešit iteračními numerickými metodami jako je Gauss-Seidelova nebo Jakobiho iterační metoda [38]. V každém iteračním kroku se zpřesní radiozita plošky i na základě hodnot radiozit ostatních plošek. Radiozitivní metody, které jsou založeny na řešení systému rovnic (2.5) se proto nazývají *shromažďovací metody* (angl. *gathering methods*).

Pokud systém rovnic (2.5) vynásobíme faktorem A_i – velikostí plochy záplaty i a využijeme recipročního vztahu $A_i F_{ij} = A_j F_{ji}$ získáme následující systém rovnic:

$$P_i = \Phi_i + \sum_j \rho_i F_{ji} P_j, \quad (2.6)$$

kde $P_i = A_i B_i$ je celková energie plošky i a $\Phi_i = A_i E_i$ je vlastní energie emitovaná ploškou i (nenulová v případě světelných zdrojů).

Radiozitivní metody řešící systém rovnic (2.6) jsou v literatuře označovány jako *vystřelovací* metody (angl. *shooting methods*). Příkladem takové metody je Southwellova relaxační metoda.

Systémy rovnic jako jsou (2.5) nebo (2.6) lze zapsat v maticové formě:

$$\mathbf{C}\mathbf{x} = \mathbf{e}, \quad (2.7)$$

kde $\mathbf{C}$ je matice o $n \times n$ prvcích c_{ij} (n je celkový počet plošek ve scéně), $\mathbf{e}$ je vektor n reálných čísel e_i , hledaným řešením je vektor $\mathbf{x}$, který tvoří n reálných čísel x_i .

Hlavními problémy při výpočtu osvětlení vyzařovací metodou jsou meshing a výpočet konfiguračních faktorů. Problematikou výpočtu konfiguračních faktorů se zabývá kapitola 3.1, meshingem pak kapitola 3.2.

Na rozdíl od mnoha různých nástrojů implementujících metodu sledování paprsku jsou radiozitivní výpočty díky své složitosti a náročnosti na uživatelské zkušenosti a znalosti problematiky využívány znatelně méně. Více či méně komerční nasazení radiozity lze nalézt např. v produktech Lightscape [2], Lightworks [30], Helios [7], Microstation Masterpiece [5], HOOPS [47], Strata 3Dpro [44], Blue Moon Rendering tools [12], Pixar Renderman [36], trueSpace [33], Softimage [43], Raysmith [39], Specter [19], Mental Ray [32], RenderPark [4].

2.4.2 Vyzařovací metody

Klasická vyzařovací metoda

Klasická nebo též *maticová radiozita* (ang. *matrix radiosity*) je původní, jednoduchý a čistě matematický postup, který je založen na výpočtu rozsáhlé *matice konfiguračních faktorů* (angl. *form factor matrix*) a řešení systému rovnic (2.5). Nejprve je scéna rozdělena na *plošky* (angl. *patches*) podle zvolené míry (např. dle maximální délky hrany plošky). Mezi ploškami jsou vypočteny vzájemné konfigurační faktory (viz kapitola 3.1), jenž tvoří matici konfiguračních faktorů $n \times n$, kde n je celkový počet plošek. Na základě této matice je sestaven soubor lineárních rovnic (2.5), které se obvykle řeší nějakou *iterační metodou*, jako např. Gauss-Seidelovou nebo Jacobiho iterační metodou.

Algoritmus tedy sestává z následujících kroků:

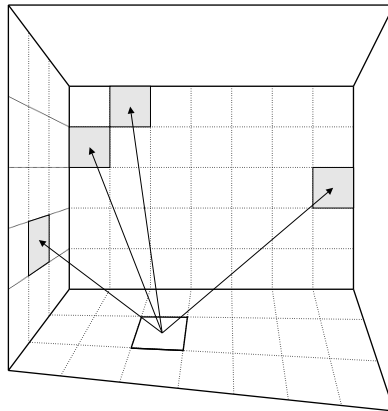
- Výpočet konfiguračních faktorů F_{ij} .
- Řešení soustavy rovnic pro uvažované vlnové délky (např. R, G, B) na základě známých radiozit jednotlivých plošek.
- Zobrazení scény dle parametrů kamery.

Maticové metody mají vysokou časovou ($O(n^2 v)$, kde n je počet plošek a v je složitost výpočtu vzájemné viditelnosti dvou plošek) i paměťovou ($O(n^2)$) náročnost. V praxi téměř nejsou implementovány, neboť nasazení maticových metod výpočtu je z uvedených důvodů možné pouze pro jednoduché scény.

Postupující radiozita

Postupující radiozita (ang. *progressive refinement radiosity*) [8, 10] postupně (přírůstkově) vylepšuje výsledek pomocí sledování zářivého toku (energie) od zdrojů (zářičů) k ostatním plochám scény. V každé iteraci se „vystřeluje“ energie z plošky s největší nevystřelenou energií

(součin vlastní radiozity plošky a velikosti plošky) do ostatních plošek scény, viz obr. 2.3. Výpočet je zpravidla ukončen (konverguje) v okamžiku, kdy jsou veškeré nevystřelené energie menší než explicitně zadaná mez. Často se používá *přídavné ambientní světlo* (angl. *ambient factor (term)*), které nahrazuje doposud neznámé osvětlení, aby bylo možné sledovat průběh výpočtu ještě před jeho ukončením.



Obrázek 2.3: Postupující radiozita – vystřelování energie

V daném okamžiku se počítá pouze jeden sloupec z matice konfiguračních faktorů, takže paměťová náročnost metody je pouze $O(n)$. Pokud je výpočet ukončen po s krocích, je časová složitost metody $O(n \cdot s \cdot v)$, přičemž v praktických případech je $s \ll n$. Problémy způsobují světelné zdroje, neboť musí být rozděleny na mnoho plošek, aby produkovaly „měkké“ stíny. Pro značné urychlení výpočtu lze použít *adaptivní dělení ploch* (angl. *adaptive refinement*), při němž jsou původní plošky s velkým spádem radiozity (např. hranice stínu atd.) dále děleny na plošky menší. Výpočet postupující metodou je možné dále urychlit přepsáním algoritmu pro více procesorů.

Metoda postupující radiozity je podle [54] vhodná i pro velmi komplexní scény, pro praktickou použitelnost je však nutné implementovat algoritmus adaptivního dělení ploch. Metoda postupující radiozity byla v minulých letech zřejmě nejčastěji implementovanou radiozitní metodou a je nasazena např. v komerčním systému Lightscape [2].

Hierarchická radiozita

Poprvé byla *hierarchická vyzařovací metoda* (angl. *hierarchical refinement*) popsána v [15]. V současné době je jednou z nejpopulárnějších a nejčastěji implementovaných vyzařovacích metod.

Algoritmus hierarchické radiozity se skládá obecně ze tří částí: řídicího algoritmu, funkce *orákulum* (angl. *oracle*) a radiozitních rutin. Jednotlivé plošky se uspořádají do několikaúrovňových stromů (např. kvadrantových). Jádrem řešení je pak funkce orákulum, která určuje, jestli přenos energie mezi dvěma ploškami vyhovuje zadaným kritériím, nebo je třeba je jemněji rozdělit. Podrobně se hierarchickou metodou zabývá kapitola 4.1.

Do třídy hierarchických radiozitních metod patří také tzv. *waveletové metody*, které jsou v současnosti objektem zájmu spíše na poli výzkumném. To je dáno vysokou implementační náročností těchto metod a nedostatkem odpovídající literatury. Rodina waveletových metod má společný algoritmus i teorii, jednotlivé metody se navzájem liší pouze volbou bázevých

funkcí, které reprezentují radiozitu na plošce. Podle [54] se waveletová radiozita hodí na výpočet středně komplexních scén, protože v případě velmi složitých scén jsou již vysoké paměťové nároky metody neúnosné.

Monte Carlo radiozita

Obsáhlý úvod do problematiky použití *Monte Carlo metod* při výpočtu globálního osvětlení nalezne čtenář v [46], nasazením stochastických metod ve vyzařovacích metodách se důkladně zabývá [3, v kapitolách 4, 5, 6, 7, 8].

Stochastické metody výpočtu jsou v současnosti velmi populární. Metodou Monte Carlo je možné nejen spolehlivě vypočítat *konfigurační faktory*, ale lze ji též aplikovat v algoritmech, které daný problém řeší *bez* výpočtu a explicitního ukládání konfiguračních faktorů. Výhody takových algoritmů spočívají v malé paměťové náročnosti a odstranění obtížného přesného vyčíslování konfiguračních faktorů, neboť toto vyčíslování metoda převádí na vzorkování náhodné veličiny s odpovídající střední hodnotou pravděpodobnosti. Stochastické metody lze aplikovat také v kombinaci s hierarchickým výpočtem radiozity. Podrobněji se problematikou Monte Carlo metod v radiozitě zabývá kapitola 4.2.

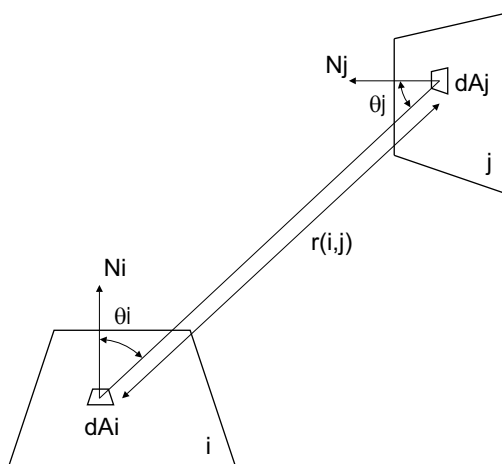
Kapitola 3

Techniky vyzařovacích metod

V této kapitole jsou popsány některé základní techniky a postupy používané při výpočtu pomocí vyzařovací metody. Nejprve jsou uvedeny způsoby výpočtu konfiguračních faktorů, dále následuje sekce o meshingu, zobrazení výsledků a jejich interpolaci. Důležitou částí kapitoly je popis modelace venkovního osvětlení.

3.1 Výpočet konfiguračních faktorů

Většina vyzařovacích metod využívá při výpočtu *matici konfiguračních faktorů* $\mathbf{F}$ (angl. *form factor matrix*). Konfigurační faktor F_{ij} udává množství energie, která dopadne na plochu i od plochy j . Ve většině aplikací lze předpokládat, že vlastní ozáření plošky je nulové, neboli $F_{ii} = 0$.



Obrázek 3.1: Geometrie pro výpočet konfiguračního faktoru

Výhodou je, že konfigurační faktory závisí pouze na geometrii scény, tedy nezávisí na pozici kamery, takže lze výpočet matice $\mathbf{F}$ pro statickou scénu provést pouze jednou. Navíc lze vypočítané konfigurační faktory použít i v případě, že se mění intenzita světelných zdrojů v čase, za předpokladu, že geometrie a pozice zdrojů jsou konstantní.

Výpočet konfiguračních faktorů bývá kritickým místem celého radiozitivního výpočtu, neboť na jeho přesnosti zásadně závisí dosažitelná přesnost konečného řešení. Navíc je výpočet

konfiguračních faktorů časově velmi náročná část výpočtu. V současné době existuje několik přístupů řešení, jejichž cílem je zvýšení rychlosti výpočtu při zachování vysoké přesnosti.

3.1.1 Analytické řešení

Analytická metoda řešení spočívá ve vyčíslení integrálu, kterým je obecně definován konfigurační faktor. Pro případ konstantní aproximace radiozity na plošce má integrální rovnice následující tvar:

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} G(x, x') dx dx', \quad (3.1)$$

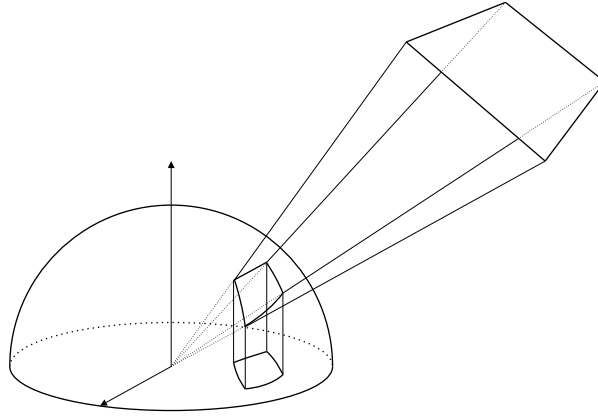
kde A_i, A_j jsou plochy elementů i, j a $G(x, x')$ je geometrický člen, který zahrnuje údaje o vzdálenosti bodů x, x' , jejich vzájemné viditelnosti a odchylky spojnice od normál povrchů, viz obr. 3.1.

Integrál se řeší jako plošný integrál ve vektorovém prostoru, který se transformuje pomocí Stokesova teorému na integrál křivkový. Hottel a Sarofin [17] vyřešili tento problém pro případ jednotkové (úplné) viditelnosti mezi všemi plochami, jejich odvození je založeno na teorii vektorových polí. Předpoklad, že veškeré plochy jsou navzájem zcela viditelné je pro praktické scény nesplnitelný, analytické řešení proto nadále zůstává na poli teoretických možností.

3.1.2 Výpočet pomocí polokoule

Základním předpokladem využití *metody polokoule* (angl. *hemisphere algorithm*) je dostatečně velká vzdálenost plošek, kdy je možné uvažovat, že veškerá energie vyzářená zdrojovou plochou je vyzářena ze středu této plochy. Pak lze výpočet konfiguračního faktoru zdrojové a cílové plochy nahradit výpočtem faktoru mezi diferenciální ploškou ve středu zdrojové plochy a cílovou plochou.

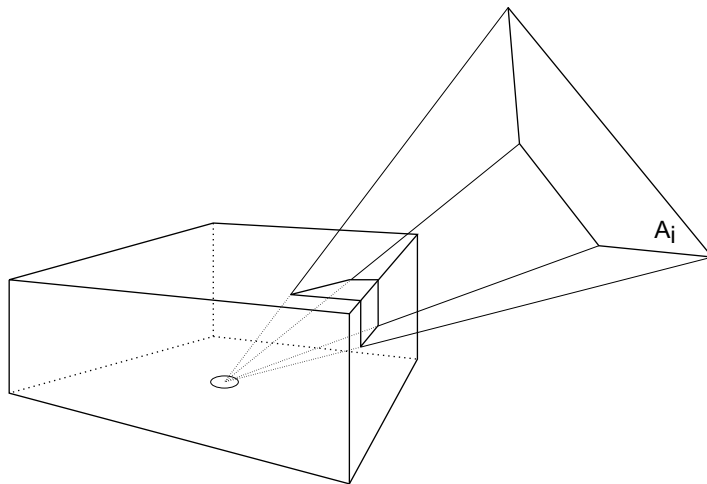
Přístup je založen na využití *Nusseltovy analogie*: konfigurační faktor plošky A_j a diferenciální plošky dA_i odpovídá ploše útvaru, který vznikne projekcí plošky A_j přes jednotkovou polokouli, která je sestrojena okolo plošky dA_i , do roviny této diferenciální plošky, viz obr. 3.2. V praxi je k této analogii nutné ještě přidat řešení viditelnosti daných plošek.



Obrázek 3.2: Nusseltova analogie

3.1.3 Výpočet pomocí polokrychle

Dvojitá projekce plošky v případě metody polokoule je výpočetně poměrně náročná, a proto vzniklo další zjednodušení, algoritmus založený na *jednotkové polokrychli* (angl. *hemicube algorithm*) [9]. Plošky se nyní promítají na stěny polokrychle, na kterých je umístěna síť bodů (pixelů) ohodnocených tzv. *delta konfiguračními faktory*, viz obr. 3.3. Tyto faktory slouží k zachování správných hodnot celkových konfiguračních faktorů.



Obrázek 3.3: Výpočet pomocí polokrychle

Pro zjištění viditelnosti plošek lze použít jednoduchý ray tracing, ale daleko vhodnějším, rychlejším řešením je využití paměti hloubky (angl. z-bufferu): jednotlivé plošky se odliší barvami, střed kamery se umístí do středu vyšetřované plošky a zobrazovaný výřez bude shodný s příslušnou stěnou polokrychle. Poté se zobrazí scéna a dle barev jednotlivých pixelů se získají konfigurační faktory $F_{i(1-n)}$ jako váhový součet (dle delta faktorů) horní a bočních stěn polokrychle. Za předpokladu, že složitost algoritmu z-buffer je $O(NP^2)$, kde P^2 je počet pixelů v paměti hloubky a N je počet ploch, pak výpočet všech konfiguračních faktorů bude mít složitost $O(N^2P^2)$. Hodnota P se typicky pohybuje v intervalu $\langle 50, 100 \rangle$. Tato metoda je obzvláště efektivní při nasazení s grafickým akcelerátorem, který hardwarově podporuje algoritmus z-buffer.

Pokud jsou dvě plošky příliš blízko u sebe, chyby ve výpočtu metodou polokrychle mohou být značné. Klesne-li tedy vzdálenost pod určitou mez, je vhodné použít metody analytického výpočtu konfiguračních faktorů.

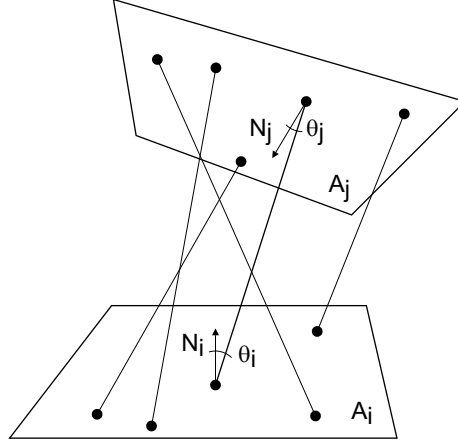
3.1.4 Výpočet pomocí kubického čtyřstěnu

Metoda polokrychle využívá k výpočtu konfiguračních faktorů pěti stěn polokrychle, tomu odpovídá i počet projekcí. Tento počet je možné snížit, použijeme-li na místo polokrychle *kubický čtyřstěn*, jenž umístíme na vyšetřovanou plošku. Kubický čtyřstěn vznikne seříznutím krychle pomocí plochy, která prochází jejími třemi vrcholy. Počet projekcí se tak sníží na tři. Další výhodou, která plyne ze symetrie čtyřstěnu, je zjednodušení výpočtu delta konfiguračních faktorů – stačí vypočítat pouze polovinu faktorů pro každou stěnu. Nevýhodou jsou naopak buňky podél základny čtyřstěnu, které mají trojúhelníkový tvar – mohou být ignorovány (jejich delta faktory jsou obvykle velmi malé), nebo mohou být vyšetřeny pro střed

trojúhelníku namísto středu čtvercového pixelu.

3.1.5 Stochastické řešení

Výpočet konfiguračního faktoru F_{ij} metodou vrhání paprsků (*Wallancova metoda*) provedeme tak, že na ploškách i, j vybereme náhodně určitý počet bodů. Každá úsečka daná body na plochách i a j určuje paprsek, který vyhodnotíme, viz obr. 3.4. Pokud paprsek nenarazí na překážku, je jemu odpovídající tzv. *delta faktor* δF_{ij} přičten k celkovému konfiguračnímu faktoru F_{ij} . Wallancova metoda je pomalejší než např. výpočet pomocí polokrychle. Více o výpočtu konfiguračních faktorů pomocí vrhání paprsků viz [52].



Obrázek 3.4: Wallancova metoda

Zjednodušenou variantou Wallancovy metody je metoda založená na vrhání pouze jednoho paprsku ze středu zdrojové do středu cílové plošky, výsledkem je pak hodnota 0 nebo 1. Test je samozřejmě vhodně provádět pouze v případě, že alespoň jeden bod cílové plošky leží před ploškou zdrojovou.

Výhodou nasazení stochastického řešení je nicméně zejména fakt, že lze explicitní výpočet hodnot konfiguračních faktorů úplně eliminovat. Základním obratem takového přístupu je využití rovnoměrně rozložených paprsků, jak popisuje kapitola 4.2.

3.2 Meshing

Většina radiozitních metod využívá ke své činnosti trojúhelníkové, resp. obdélníkové, sítě (angl. *mesh*). Kvalita sítě velkou měrou ovlivňuje přesnost řešení, které lze výpočtem získat. Síť by měla být dostatečně hustá, aby bylo možno podchytit veškeré aspekty přirozeného osvětlení. Příliš hustá síť může však vést k mnoha zbytečným výpočtům a výraznému zpomalení celého výpočtu, neboť počet konfiguračních faktorů je kvadrátem počtu elementů sítě.

3.2.1 Diskretizační chyba

Aby bylo možné hodnotit kvalitu sítě, je nejprve nutné nějakým způsobem měřit přesnost aproximace radiozitní funkce. Takovou míru nám dají dále popsání výpočty, resp. odhady, diskretizační chyby [3]. *Diskretizační chyba* je rozdíl mezi přesnou hodnotou radiozitní funkce

a její aproximací. Tato chyba vzniká jako důsledek rozdělení scény na konečně velké elementy. Přesnost aproximace radiozitní funkce $\hat{B}(x)$ je tedy přímo závislá na velikosti a struktuře sítě a dále též na volbě báзовých funkcí N_i :

$$B(x) \approx \hat{B}(x) = \sum_{i=1}^n B_i N_i(x). \quad (3.2)$$

Bázové funkce definují konečný podprostor funkcí, ze kterého je realizována aproximace radiozitní funkce pomocí koeficientů B_i .

Skutečná chyba

Skutečná chyba, nebo též Galerkinova diskretizační chyba, je přesná hodnota diskretizační chyby. Velikost skutečné chyby je dána rozdílem radiozitní funkce a její aproximace:

$$\varepsilon(x) = B(x) - \hat{B}(x). \quad (3.3)$$

Rovnici (3.3) v praxi samozřejmě není možné vyčíslit, neboť neznáme skutečnou radiozitní funkci $B(x)$. Hodnota diskretizační chyby $\varepsilon(x)$ proto může být pouze odhadnuta.

Lokální odhad

Aproximace radiozitní funkce pomocí báзовých funkcí vyššího řádu umožňují dosáhnout větší přesnosti než aproximace řádů nižších. Jedním z možných přístupů jak odhadnout diskretizační chybu je proto porovnání původní aproximace

$$\hat{B}(x) = \sum_{i=1}^n B_i N_i(x), \quad (3.4)$$

kde $N_i(x)$ jsou báзовé funkce řádu k , s aproximací vyššího řádu

$$\tilde{B}(x) = \sum_{i=1}^n B_i \tilde{N}_i(x), \quad (3.5)$$

kde nové báзовé funkce $\tilde{N}_i(x)$ jsou řádu $k + 1$ nebo vyššího. Odhad diskretizační chyby $\hat{\varepsilon}(x)$ je tedy dán vztahem:

$$\hat{\varepsilon}(x) = \hat{B}(x) - \tilde{B}(x). \quad (3.6)$$

Reziduum

Reziduum je další možností, jak charakterizovat diskretizační chybu $\varepsilon(x)$. Reziduum $r(x)$ získáme po dosazení aproximace $\hat{B}(x)$ za radiozitní funkci $B(x)$ do původní integrální rovnice (2.4):

$$r(x) = \hat{B}(x) - E(x) - \int_A \rho(x) G(x, x') \hat{B}(x) dA'. \quad (3.7)$$

Pokud je hodnota rezidua nulová na celém definičním oboru, pak je aproximace $\hat{B}(x)$ shodná s přesným řešením a diskretizační chyba je nulová. Jinak ovšem není hodnota rezidua identická s hodnotou diskretizační chyby. Vyčíslení hodnoty rezidua je vlivem integrálu ve vztahu (3.7) poměrně náročné a běžně se proto provádí pouze ve vybraných bodech. Další informace včetně návrhu algoritmů pro výpočet rezidua lze nalézt v [3].

3.2.2 Vlastnosti sítě

Základními vlastnostmi, které zásadně ovlivňují kvalitu sítě, jsou hustota sítě, geometrie a řád elementů sítě a způsob reprezentace diskontinuit.

Hustota sítě

Hustota sítě je dána velikostí jednotlivých elementů, do kterých byla původní scéna rozdělena. Čím je hustota sítě větší, tím je při aproximaci radiozitní funkce k dispozici více stupňů volnosti. Hustší síť tak umožňuje lépe zachytit jemné variace osvětlení, jako jsou např. hranice stínu, ve vypočteném řešení.

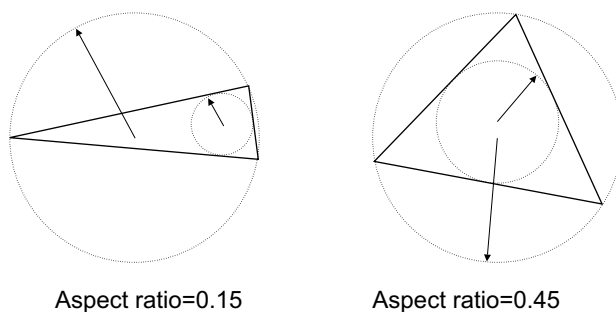
Uniformní zvýšení hustoty sítě má sice pozitivní vliv na kvalitu řešení, zároveň však výrazně zvyšuje výpočetní nároky. Přitom mnohé výpočty jsou v takovém případě prováděny zbytečně (v oblastech, kde je gradient radiozitní funkce nulový, resp. zanedbatelně malý). V praxi se proto často využívá sítí, které nejsou děleny uniformě. S výhodou se v radiozitních systémech používají algoritmy pro adaptivní dělení ploch.

Řád elementů

Řád elementů sítě, resp. řád bazových funkcí definovaných v rámci elementu, je další veličinou, která ovlivňuje kvalitu řešení. Vyšší řády elementů umožňují lépe aproximovat radiozitní funkci, ale jsou výpočetně náročnější. Hodnota řádu dále ovlivňuje stupeň spojitosti aproximace na hranicích elementů, na který je lidské oko zvláště citlivé, více v kapitole 3.4.

Geometrie elementů

V radiozitních systémech se využívají elementy s jednoduchou *geometrií*, typicky trojúhelníkové nebo čtyřúhelníkové. Přitom se při výpočtu často provádí mapování souřadnic na ještě jednodušší elementy, např. na čtverec nebo obdélník. Výpočet s elementy se složitější geometrií je výrazně náročnější a prakticky se nepoužívá.



Obrázek 3.5: Aspect ratio

Ve většině případů je výhodné, aby jednotlivé elementy měly co největší aspect ratio. *Aspect ratio* je definováno jako poměr poloměru největší kružnice, kterou lze celou umístit uvnitř elementu, a poloměru nejmenší kružnice, která leží celá vně elementu, viz obr. 3.5. V případě, že je znám průběh radiozitní funkce, je možné naopak s výhodou použít anizotropickou síť, kde elementy mají typicky aspect ratio velmi malé.

Diskontinuity

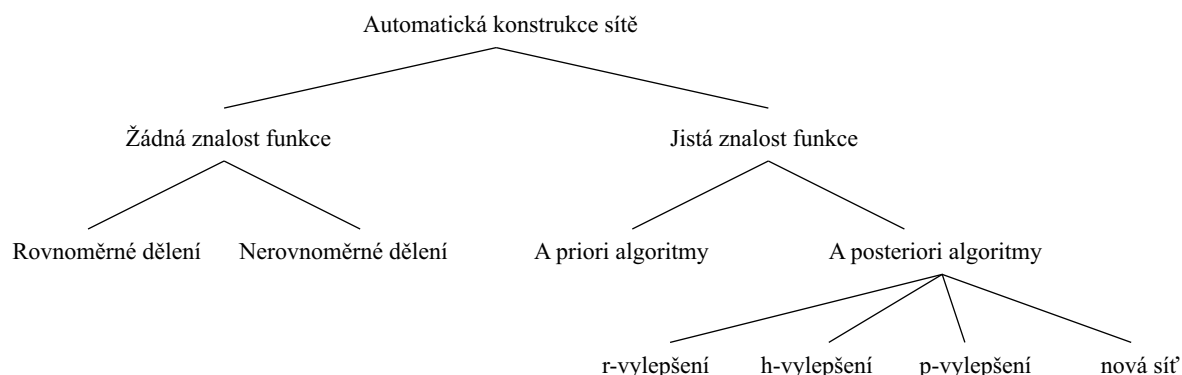
V průběhu radiozitní funkce se obecně mohou vyskytovat diskontinuity kteréhokoli řádu. Z hlediska vizuální kvality jsou nejdůležitější diskontinuity v hodnotě (nulté derivaci) a v první a druhé derivaci.

Diskontinuity v hodnotě vznikají v místech, kde se jedna ploška dotýká druhé. V takových místech se hodnota radiozitní funkce mění skokově, vlivem nevhodné sítě však dochází často k plynulému přechodu, a tedy k chybě. *Diskontinuity v první a druhé derivaci* se objevují na hranici stínu a polostínu. Pokud elementy sítě přesahují tyto hranice, dochází při následné interpolaci vypočtených hodnot k chybě.

Vzhledem k tomu, že interpolace uvnitř elementu je C^∞ spojitá (viz kapitola 3.4), je pro správné ošetření diskontinuit potřebné, aby vždy odpovídaly nějaké hraně elementu. Algoritmus, jehož cílem je z pohledu eliminace diskontinuit správná síť se nazývá *discontinuity meshing*.

3.2.3 Automatická konstrukce sítě

Přestože intervence uživatele při konstrukci sítě může mít velmi pozitivní vliv na kvalitu výsledné sítě, používají se v oblasti vyzařovacích metod téměř výhradně algoritmy automatické, které nevyžadují od daného uživatele žádnou zvláštní zkušenost.



Obrázek 3.6: Algoritmy pro automatickou konstrukci sítě

Algoritmy pro automatickou konstrukci sítě lze dle [40] rozdělit do několika skupin, viz obr. 3.6. První skupinu tvoří algoritmy, které nevyužívají žádné znalosti o průběhu aproximované funkce. Pro dosažení co nejvyšší kvality sítě se většinou nelze obejít bez jakési znalosti daného průběhu, nicméně určitý stupeň konstrukce sítě bez této znalosti je prakticky vždy nutný. Tuto skupinu dělíme na algoritmy, které tvoří rovnoměrnou (angl. uniform) síť a na algoritmy konstruující nerovnoměrnou síť.

Algoritmy, které využívají znalosti aproximované funkce, dělíme z hlediska konstrukce sítě a výpočtu na apriorní a aposteriorní. Apriorní metody konstruují veškeré části sítě před vlastním výpočtem. Typickým představitelem takové metody je *discontinuity meshing*. Aposteriorní metody vytvářejí, resp. vylepšují, síť poté, co je známo alespoň částečné řešení. Primární síť je tedy nutno zkonstruovat metodou z první skupiny, její vylepšení pak již během výpočtu zajistí aposteriorní metoda.

Rozlišujeme několik aposteriorních přístupů. Při implementaci algoritmu *r-vylepšení* (angl. *r-refinement*) se v závislosti na odhadu chyby mění poloha bodů sítě. *H-vylepšení* je používané

velmi často a spočívá v adaptivním rozdělování již existujících elementů sítě. *P-vylepšení* se snaží snížit chybu aproximace za pomoci zvyšování řádu bazových funkcí elementů sítě. Další možností je na základě známých nepřesností dané sítě zkonstruovat úplně novou síť s menšími chybami (angl. *remeshing*).

3.2.4 Radiozitní textury

Dělení scény na plošky s sebou přináší změny v geometrii modelu a jeho implementace je díky tomu poměrně obtížná. Alternativou k dělení scény na plošky je využití *radiozitních textur* – *tex* (angl. *radiosity textures*), které popsal Heckbert v [16]. Radiozitní textura je podobná klasické textuře, avšak její hodnoty reprezentují aproximaci průběhu radiozitní funkce. V Heckbertově přístupu jsou navíc radiozitní textury organizovány do kvadrantového stromu, aby bylo možné aplikovat adaptivní vzorkování. Během zobrazování se průběh radiozitní funkce rekonstruuje ze vzorků v radiozitní textuře stejně jako při klasickém mapování textur. Podobný přístup jako Heckbert aplikoval Vedel v [49], kde použil strukturu podobnou radiozitní textuře.

3.3 Zobrazení výsledku

Lidské oko vnímá v přírodě značný rozsah hodnot osvětlení (cca 15 řádů). Tak velký rozsah nelze samozřejmě získat při zobrazení výsledku na monitoru, a proto je nutné vypočtené hodnoty transformovat (typicky do intervalu $\langle 0, 1 \rangle$, resp. $\langle 0, 255 \rangle$).

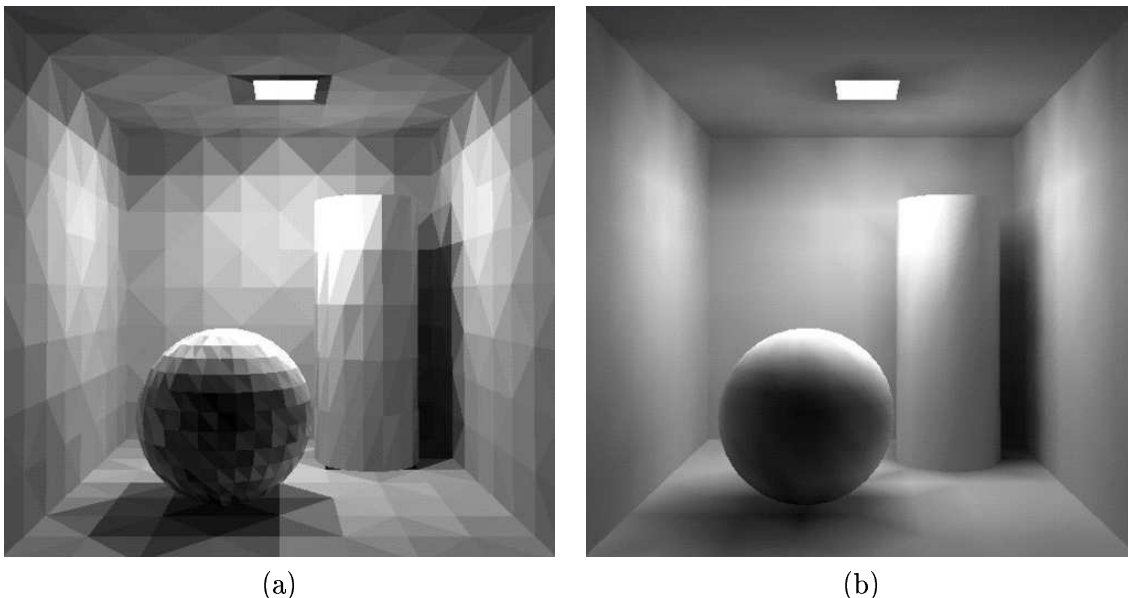
Transformace podělením maximální vypočtenou hodnotou osvětlení není vzhledem k rozdělení hodnot v rámci daného intervalu vhodná. V zobrazené scéně by v takovém případě byly viditelné pouze světelné zdroje (jejichž jas je ve skutečnosti řádově vyšší, než osvětlení ostatních těles), zbytek scény by byl velmi tmavý. Většinou se proto hodnoty transformují do požadovaného intervalu bez uvážení zdrojů světla, získaným maximálním hodnotám se přiřadí hodnota nepatrně menší než horní mez intervalu a světelným zdrojům se přiřadí hodnota rovná této horní mezi. Nevýhodou popsané transformace však je, že scéna osvětlená postupně dvěma zdroji s různými hodnotami jasu je zobrazena v obou případech stejně.

V knize [10] je uveden ilustrační příklad této skutečnosti: nejprve bylo za pomoci vyzařovací metody vypočteno osvětlení místnosti, kde zdroj světla představoval oheň v krbu s fyzikálně odpovídající hodnotou svítivosti. Poté byla svítivost ohně v krbu snížena na jednu kandelu (cca 1 svíčka), přičemž jeho geometrie, stejně jako geometrie místnosti zůstala zachována. Nově vypočtená scéna se od původní scény v ničem nelišila.

3.4 Interpolace

Interpolace hodnot radiozity na ploškách je způsob, jakým lze z *objektivně* přesného řešení získat i *subjektivně* dobrý výsledek, pomocí eliminace některých pro lidské oko rušivých vizuálních vjemů. Zřejmě nejvíce rušivě působí na pozorovatele *nespojitosti hodnot* radiozitní funkce (diskontinuity v nulté derivaci), které jsou způsobeny aproximací konstantními funkcemi v rámci plošky (viz obr. 3.7). Dále je oko poměrně citlivé i na *nespojitosti v první derivaci*. Nespojitosti v hodnotách lze eliminovat interpolací nultého řádu C^0 , nespojitosti v první derivaci interpolací prvního řádu C^1 . Označení C^n se nazývá *parametrická spojitost*, říkáme, že křivka $Q(t)$ je třídy C^n , má-li ve všech bodech spojitě derivace do řádu n . Interpo-

lace řádu C^n pak znamená, že veškeré hodnoty interpoluje tak, aby po jejich spojení vznikaly křivky s parametrickou spojitostí C^n .



Obrázek 3.7: Porovnání interpolace konstantními (a) a lineárními (b) funkcemi

3.4.1 Interpolace C^0

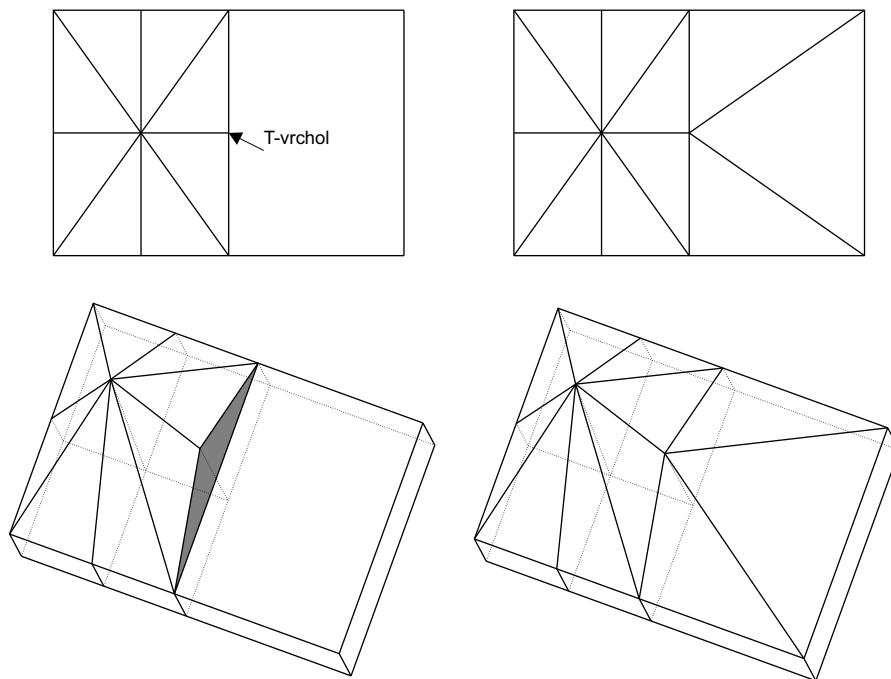
Interpolaci nultého řádu C^0 zajišťují lineární funkce, v praxi máme často k dispozici řešení s pouze konstantními hodnotami radiozity na ploškách. Z těchto hodnot je nutné hledanou interpolaci odvodit. Jednou z možností jak toho dosáhnout je nahradit hodnotu radiozity na plošce hodnotami ve vrcholech plošky. Hodnoty ve vrcholech jsou získány průměrováním hodnot radiozity na přilehlých ploškách [9] a mezi těmito hodnotami je v rámci elementu provedena *lineární* (v případě trojúhelníkových plošek) nebo *bilineární* (v případě obdélníkových plošek) *interpolace*. Průměrování hodnot ve vrcholech „změkčuje“ radiozitní funkci, což může často zamaskovat artefakty dané výpočtem. Na druhou stranu se však může stát, že se průměrováním eliminují malé, ale důležité správně vypočtené změny v osvětlení.

Lineární, resp. bilineární, interpolace se provádí v rámci parametrických souřadnic na plošce, takže je nezávislá na projekci plošky do zobrazovací roviny. Při implementaci se však často z důvodu rychlosti přistupuje k interpolaci až v rovině obrazu. Vlivem toho, že perspektivní transformace z prostoru scény do zobrazovací roviny není afinní, není interpolace v rovině obrazu bohužel ekvivalentní s interpolací v rámci parametrických souřadnic plošky.

Příkladem je *Gouroudovo stínování* (interpolace barvy, viz [51]), při kterém jsou barvy ve vrcholech polygonů lineárně interpolovány podél hran polygonu a mezi těmito hodnotami dále interpolovány algoritmem řádkového rozkladu. Důsledkem způsobu výpočtu však je závislost na orientaci polygonu, interpolace se může viditelně měnit v závislosti na pohledu. Výhodou je vysoká rychlost interpolace, zvláště v případě, že je zajišťována grafickým akcelerátorem.

Problémy při interpolaci způsobují tzv. *T-vrcholy* (viz obr. 3.8), které vznikají např. při rekursivním adaptivním dělení ploch. V okamžiku, kdy rekursivní proces ze dvou sousedních

plošek jednu rozdělí a druhou ne, dojde v jednom případě k rozdělení společné hrany a ve druhém ne, což může způsobit nespojitosti při interpolaci, tzv. praskliny. V případě modelace ploch se tento problém často označuje jako problém rozlomení (angl. cracking problem).



Obrázek 3.8: T-vrchol a jedna z jeho možných eliminací

3.4.2 Interpolace C^1

Lineární interpolace, která má parametrickou spojitost C^0 , je značným vylepšením oproti konstantnímu stínování, nicméně není schopna eliminovat veškeré perceptuální problémy. V některých případech působí na oko velmi rušivě i nespojitosti v první derivaci, a tak má význam zabývat se interpolací C^1 .

Při interpolaci C^1 , resp. vyššího řádu, se vyskytují problémy velmi podobné problémům, které vznikají při konstrukci ploch v třírozměrném prostoru. V případě interpolace radiozitivní funkce máme dvě souřadnice dány parametrickými souřadnicemi bodu na ploše a třetí souřadnicí je hodnota radiozitivní funkce. Plocha, která má v každém bodě parametrickou spojitost C^1 , může být zkonstruována např. pomocí Béziérových ploch druhého nebo vyššího řádu. Interpolaci kvadratickými Béziérovými plochami použil pro případ radiozity Reichert [37], kubické plochy aplikoval v radiozitivním výpočtu Max [31]. Interpolaci s požadovanou spojitostí lze zkonstruovat také z obecnějších B-spline ploch, které jsou popsány např. v [51].

3.5 Modelace venkovního osvětlení

Vizualizace světelného zatížení venkovních scén je obecně náročnější než vizualizace zatížení interiérů, neboť se při výpočtu střetáváme s problémy, které jsou současnými radiozitivními algoritmy poměrně špatně ošetřeny. Exteriéry jsou zpravidla *geometricky* podstatně složitější

než vnitřní scény. Geometrická náročnost dále vzroste při začlenění přírodních objektů, jako jsou stromy, rostliny, reliéf terénu atd. *Osvětlení* venkovních scén je velmi komplexní a má své specifické vlastnosti. Při výpočtu vnitřní scény je např. možné, na rozdíl od exteriéru, urychlit radiozitní výpočet tím, že okna a veškeré další zdroje světla v místnosti nahradíme ploškami s předem vypočtenými hodnotami vlastní zářivosti. Výpočet pak provedeme pro tento velmi omezený soubor světelných zdrojů.

Začlenění *přírodních světelných zdrojů* je klíčovým požadavkem pro vytvoření realistické simulace venkovní scény. V principu se odlišuje charakter osvětlení Slunce a oblohy. Světelný tok *Slunce* je koncentrován na relativně malém prostorovém úhlu, zatímco *obloha*, jakožto druhý světelný zdroj venkovní scény, v podstatě obklopuje všechny objekty. Pro středně rozlehlé scény (např. dům, ulice, nebo část města) lze Slunce modelovat jako zdroj rovnoběžných paprsků. Účinek oblohy se obvykle nahrazuje soustavou plošek uspořádanou do tvaru polokoule.

Problematikou výpočtu osvětlení v exteriéru se zabývá [34], kde je k výpočtu použit radiozitní systém s postupující radiozitou. Před zahájením klasického výpočtu se provede inicializační fáze, při které se do scény „vystřelí“ všechna energie Slunce a oblohy. Přístup v [11] je založen na použití hierarchické radiozity. Výhodou tohoto řešení je možnost reprezentovat hierarchickou strukturu oblohy pouze procedurálně, což zjednodušuje geometrickou náročnost výpočtu.

3.5.1 Světelné zdroje

Klasická vyzařovací metoda uvažuje ve scéně pouze plošné zdroje. Světelné zdroje ve vyzařovací metodě se tedy od klasických světelných zdrojů používaných v ostatních oblastech počítačové grafiky liší tím, že světlo nejen vyzařují, ale mají i geometrii, která může okolní světlo odrážet. Plošným zdrojem je z pohledu vyzařovací metody každá ploška scény, která má nenulovou hodnotu vlastní emise E_i , resp. Φ_i . Předpoklad uvažování pouze plošných zdrojů je pro modelaci venkovního osvětlení příliš omezující, pro reálnější simulaci osvětlení v exteriéru potřebujeme umožnit začlenění zdrojů rovnoběžného světla do scény. Řešení tohoto problému je popsáno v odstavci 5.6.1.

3.5.2 Sluneční světlo

Slunce je nejdůležitější (a často jediný) primární světelný zdroj scény v exteriéru. Pro účely výpočtu scén velikosti např. městské aglomerace můžeme Slunce považovat za *zdroj rovnoběžného světla*. Bodový zdroj v nekonečnu, nebo velmi rozsáhlý plošný zdroj, který vyzařuje světlo pouze v jednom směru $\vec{\omega}$, lze považovat za zdroj rovnoběžného světla.

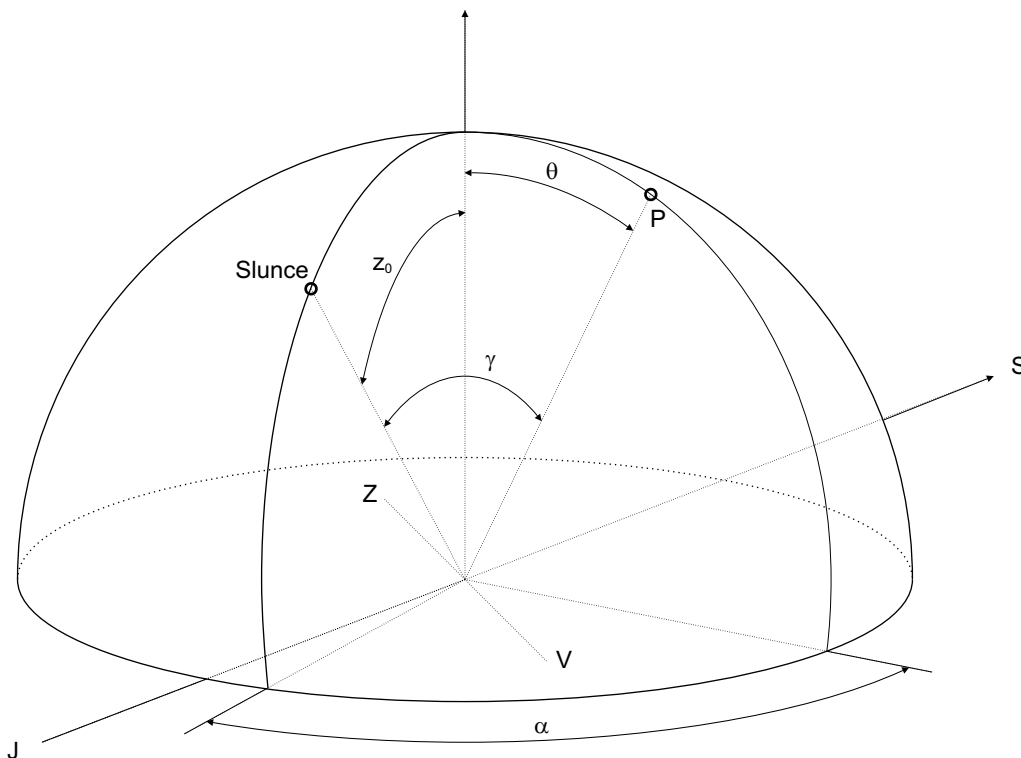
Konfigurační faktor mezi zdrojem rovnoběžného světla a ploškou j je dán projekcí viditelné části plošky do roviny kolmé na směr šíření světla:

$$F_{ij} = \int_{A_j} \cos \theta_j V(-\vec{\omega}, \mathbf{x}_j) dA_j, \quad (3.8)$$

kde $V(-\vec{\omega}, \mathbf{x}_j)$ je funkce *viditelnosti*, která nabývá hodnoty *jedna* právě tehdy, když paprsku ve směru $\vec{\omega}$ z bodu $\mathbf{x}_j$ na plošce j nebrání v cestě ke zdroji paralelního světla žádné jiné těleso, jinak nabývá hodnoty *nula*.

3.5.3 Světlo oblohy

Světlo oblohy (angl. *skylight*) změkčuje ostré stíny, které vytváří přímé sluneční světlo. V počítačové grafice se za účelem zjednodušení a zrychlení výpočtu světelný účinek oblohy často nahrazuje účinkem zdroje rovnoběžného světla ve tvaru polokoule s nekonečným poloměrem.



Obrázek 3.9: Geometrie pro výpočet záře bodu na obloze

Mezinárodní komise CIE (viz příloha A) poskytuje několik formulí pro odhad záře v daném bodě P na hemisféře oblohy v závislosti na poloze Slunce, množství mraků atd.

Pro *kompletně zataženou* (angl. *completely overcast sky*) oblohu je zář dána rovnicí 3.9

$$L(\theta) = L_z \frac{1 + 2 \cos \theta}{3} \quad [W \cdot sr^{-1} \cdot m^{-2}], \quad (3.9)$$

kde L_z je zář v bodě zenitu. L_z je samozřejmě závislá na poloze (výšce) Slunce. Zář oblohy je v tomto jednoduchém empirickém modelu rovnoměrně rozložena v kruzích dle dané výšky, takže je funkcí pouze úhlu θ , který svírají bod P a zenit (viz obr. 3.9). Při výpočtu dle tohoto modelu vychází obloha jasnější v okolí zenitu a tmavší poblíž horizontu.

Pro výpočet *čisté oblohy* (angl. *clear sky*) CIE doporučuje následující vztah:

$$L(\theta, \gamma) = L_z \frac{(0.91 + 10e^{-3\gamma} + 0.45 \cos^2 \gamma)(1 - e^{-0.32 \sec \theta})}{0.274(0.91 + 10e^{-3z_0} + 0.45 \cos^2(z_0))} \quad [W \cdot sr^{-1} \cdot m^{-2}], \quad (3.10)$$

kde L_z a θ mají stejný význam jako výše, z_0 je úhel mezi zenitem a Sluncem, γ je úhel mezi Sluncem a bodem P (viz obr. 3.9). Úhel γ může být též vypočten na základě úhlu α , který mezi sebou svírají projekce Slunce a bodu P do základní roviny (viz obr. 3.9), přičemž $\cos \gamma = \cos z_0 \cos \theta + \sin z_0 \sin \theta \sin \alpha$ [1].

Výpočet *konfiguračního faktoru* mezi hemisférou a ploškou j vyžaduje integraci přes plochu celé polokoule Ω a přes plochu daného elementu j . $S(\vec{\omega})$ je definováno jako poměr mezi září ve směru $\vec{\omega}$ ku R_z , září v zenitu. Konfigurační faktor je tedy dán rovnicí:

$$F_{ij} = \int_{\Omega} \int_{A_j} \frac{S(\vec{\omega}) \cos \theta_j}{2\pi} V(\vec{\omega}, dA_j) dA_j d\omega. \quad (3.11)$$

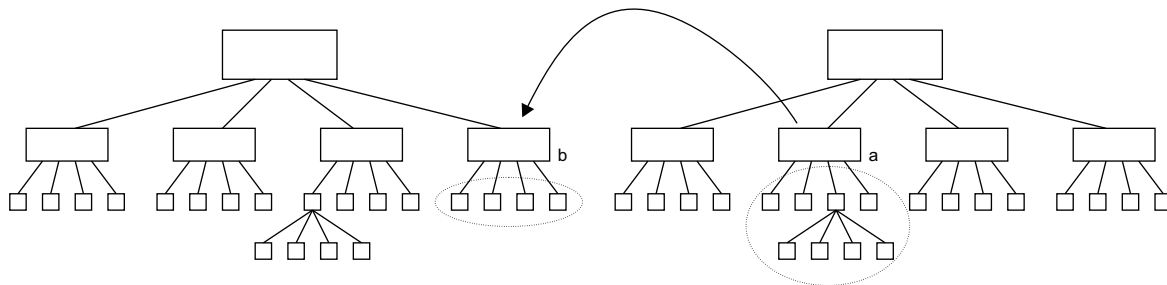
Kapitola 4

Hierarchická a Monte Carlo vyzařovací metoda

V této kapitole budou podrobně popsány dvě populární metody radiozitivního výpočtu globálního osvětlení, hierarchická metoda a stochastická metoda Monte Carlo. Dále bude popsána hierarchická Monte Carlo metoda, která vzniká jako kombinace těchto dvou metod.

4.1 Hierarchická vyzařovací metoda

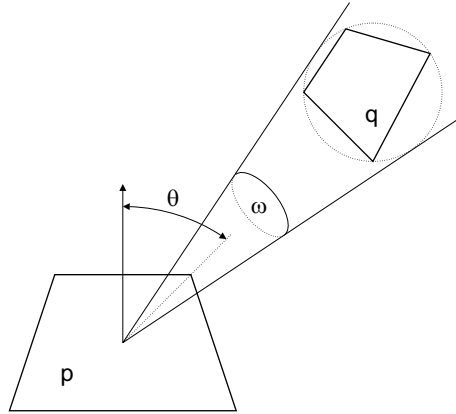
Hierarchická vyzařovací metoda (angl. *hierarchical refinement*) [15] řeší radiozitivní problém s ohledem na dvě nejnáročnější části výpočtu – výpočet konfiguračních faktorů a meshing. V klasickém (např. maticovém nebo progresivním) radiozitivním systému je nutné vypočítat konfigurační faktory mezi každými dvěma ploškami scény, což je časově nejnáročnější část výpočtu ($O(n^2)$). Autoři hierarchické metody ukázali, že tento problém nemá inherentní složitost $O(n^2)$, v reálných scénách vystačíme s mnohem méně interakcemi mezi ploškami, v některých případech pouze s $O(n)$. Ke snížení počtu vyšetřovaných konfiguračních faktorů vede následující úvaha: zjemňování dělení plošek scény má kladný vliv na kvalitu výsledného obrazu, ale zároveň výrazně zvyšuje počet konfiguračních faktorů, které je nutno vyčíslit. Přitom je vliv nově vzniklých malých plošek na vzdálené plochy scény v některých případech zanedbatelný. Vliv mnoha takovýchto plošek, tzn. mnoho konfiguračních faktorů s číselně malou hodnotou, je možné nahradit jedním „velkým“ konfiguračním faktorem.



Obrázek 4.1: Interakce v hierarchické struktuře

Algoritmus hierarchické radiozity vytváří v paměti během výpočtu víceúrovňovou hierarchickou strukturu plošek, jež může být realizována např. kvadrantovým stromem a která se pro účely dalšího výkladu skládá z elementů. Hierarchická struktura je klíčovým bodem algoritmu, neboť jednak zajišťuje rozdělení vstupních ploch scény na jemnější plošky (meshing) a také omezuje počet skutečně vyšetřených konfiguračních faktorů. Přenos energie mezi elementy, v terminologii hierarchické radiozity mluvíme o jednosměrné vazbě (angl. *link*), se totiž může realizovat na kterékoli úrovni hierarchické struktury. Příklad takové interakce je uveden na obrázku 4.1, kde je energie ze skupiny sedmi elementů (a) přenesena do skupiny čtyř elementů (b). Každá vazba tedy reprezentuje určitou podmnožinu z množiny konfiguračních faktorů, které by bylo nutno v případě standardního postupu výpočtu vyčíslit. Tato podmnožina odpovídá konfiguračním faktorům mezi všemi páry elementů v podstromech dvou elementů, mezi kterými se interakce uskutečňuje.

Rozhodnutí, zda vazba mezi dvěma elementy v hierarchické struktuře vyhovuje požadovaným kritériím, má na starost funkce *orákulum* (angl. *link oracle*). Pokud tato funkce určí, že přesnost přenosu energie mezi dvěma elementy je dostatečná, vytvoří se jednosměrná vazba mezi těmito elementy. Pokud naopak chyba překročí danou mez, orákulum určí, zda mají být rozděleny oba elementy, nebo pouze jeden z nich. Podle toho se provede rozdělení elementů a na jejich následníky se rekurzivně znovu zavolá funkce orákulum. Postup je zapsán v dílčím algoritmu 4.1, kde funkce **refine** zkoumá vztah dvou uzlů **p**, **q**.



Obrázek 4.2: Geometrie pro odhad konfiguračního faktoru

Funkce orákulum má zásadní vliv na efektivitu celého výpočtu. Musí být založena na vyčíslení nějaké heuristiky, neboť přesný výpočet hodnoty konfiguračního faktoru by zhoršil časovou složitost na $O(n^2)$. Hanrahan ve své práci [15] používá ve funkci orákulum jednoduchý *odhad konfiguračního faktoru* dvou vzájemně nezastíněných elementů v hierarchii:

$$F_{pq} = \frac{\cos \theta}{\pi} \omega_q, \quad (4.1)$$

kde ω_q je prostorový úhel, který vymezuje koule (nebo disk), která obklopuje celou plochu elementu q (viz obr. 4.2). Tento odhad reprezentuje horní mez hodnoty konfiguračního faktoru, tedy i chyby, reálná hodnota konfiguračního faktoru musí být vždy menší vlivem uvažování přesné geometrie a zastínění elementů.

Množina vazeb, kterou vytvořila funkce **refine** definuje lineární systém, který je třeba vyřešit, abychom získali hodnoty jednotlivých radiozit elementů. Na každé úrovni hierarchie

musí být o daném elementu známa jeho difúzní odrazivost ρ , plocha A_i , vlastní emise E_i a zachycená radiozita B_g . K řešení systému dospějeme po několika iteracích, při kterých vždy provedeme distribuci radiozity mezi elementy podél dříve vytvořených vazeb a zajistíme konzistenci těchto hodnot v rámci celé hierarchie. Algoritmus výpočtu hierarchické radiozity je popsán jako algoritmus 4.2.

Algoritmus 4.1 Hierarchická vyzařovací metoda – vytváření vazeb (refine)

```

refine (Patch p, Patch q, float  $F_\epsilon$ ) {
  if oracle(p, q,  $F_\epsilon$ ) then
    vytvoř vazbu mezi elementy p, q;
  else
    podle potřeby rozděl element p nebo q;
    if rozdělen p then
      for každý potomek r elementu q do
        refine (p, r,  $F_\epsilon$ );
      end for
    else if rozdělen q then
      for každý potomek r elementu p do
        refine (r, q,  $F_\epsilon$ );
      end for
    else
      vytvoř vazbu mezi elementy p, q;
    end if
  end if
}
```

4.2 Monte Carlo vyzařovací metody

Při řešení pomocí *Monte Carlo metody* se výpočet složitého problému (např. výpočet integrálu) převede na stochastický problém, jehož střední hodnota odpovídá hledanému řešení. Vlastní stochastický výpočet poté sestává pouze z generování dostatečného počtu vzorků s daným rozložením, přičemž hledané řešení je dáno střední hodnotou uvažované náhodné veličiny.

Velkou výhodou Monte Carlo metod v případě numerického výpočtu integrálů je, že zabráňují tzv. *dimenziální explozi* (angl. *dimensional explosion*). Dimenziální exploze (nebo též dimenziální jádro) je vlastnost běžných metod pro přibližný výpočet integrálů (jako je např. obdélníková metoda, viz [38, strana 492]), jejichž chyba roste s dimenzí prostoru s podle vztahu $\Delta f / N^{1/s}$, kde Δf je celková změna integrované funkce a N je počet vzorků. Z hlediska počtu vzorků to znamená, že pro danou maximální chybu ϵ musíme s rostoucí dimenzí s počet vzorků zvyšovat dle vztahu $(\Delta f / \epsilon)^s$, takže výpočetní náročnost roste exponenciálně s dimenzí, hovoříme o dimenziální explozi.

Řešení problému radiozity pomocí Monte Carlo metody je plnohodnotnou alternativou k deterministickému výpočtu (vyčíslení konfiguračních faktorů a následnému řešení systému radiozitivních rovnic). Přitom nám stochastický přístup nabízí následující výhody:

- nízké paměťové nároky: metody Monte Carlo umožňují řešit rozsáhlé scény bez nutnosti

Algoritmus 4.2 Hierarchická vyzařovací metoda

```
for každá položka  $i$  do
     $B_i = E_i$ 
end for
for každá dvojice plošek  $i, j$  do
    refine( $i, j, F_\epsilon$ );
end for
repeat
    for každý element v hierarchii  $i$  do
        urči radiozitu přicházející po všech vazbách na element  $i$ 
    end for
    for každý element v hierarchii  $i$  do
        rekurzivně rozděl přijatou radiozitu potomkům a naopak přičti k vyzářené radiozitě
        elementu  $i$  hodnoty vyzářené jeho potomky (pushPull)
    end for
until metoda zkonvergovala
```

uschovávat konfigurační faktory

- efektivita: časová složitost metod je velmi málo závislá na počtu plošek scény
- spolehlivost: metody založené na vrhání paprsků do scény jsou odolnější vůči chybám během výpočtu než deterministické metody
- snadná kontrola výpočetní chyby: chybu výpočtu v metodách Monte Carlo ovlivňuje zejména počet paprsků vyslaných do scény, jenž je možné snadno kontrolovat
- kompatibilita: stochastické metody je možné kombinovat se známými postupy používanými v deterministických metodách, jako jsou např. aproximace bázovými funkcemi vyšších řádů, hierarchická struktura, vzorkování podle důležitosti (angl. importance sampling) atd.

Metody Monte Carlo můžeme podle způsobu jejich aplikace v radiozitivním systému rozdělit do následujících kategorií:

- Stochastický výpočet pouze konfiguračních faktorů scény, k dalšímu výpočtu lze využít běžných vyzařovacích metod
- Stochastické relaxační metody (angl. *relaxation methods*)
- Stochastické průchody scénou (angl. *random walk methods*)
- Různé další přístupy, které jsou založeny na řešení ekvivalentního problému pomocí metody Monte Carlo

Ke zrychlení výpočtu lze v případě Monte Carlo metod využít různých technik, které snižují rozptyl, případně aplikovat vhodnější vzorkování náhodné veličiny (kapitoly 4.2.1 a 4.2.2). V dalším textu (kapitola 4.2.4) bude popsána Jakobiho iterační metoda, zájemce o popis dalších Monte Carlo přístupů nalezne dostatek informací v [3].

4.2.1 Kvazi Monte Carlo

Metody *kvazi Monte Carlo* se od klasických Monte Carlo metod liší tím, že při vzorkování náhodných veličin využívají vzorky, které mají tzv. kvazi-náhodnou distribuci. Kvazináhodné vzorky nejsou sice vzájemně nezávislé, ale mají statistické vlastnosti, které umožňují jejich použití namísto čistě náhodných vzorků. Výhodou kvazináhodných vzorků (resp. sekvencí kvazináhodných čísel) je, že jsou na daném intervalu rozděleny rovnoměrněji, než klasické náhodné a pseudonáhodné sekvence.

V literatuře [23, 53] již bylo publikováno mnoho různých kvazináhodných sekvencí, *Haltonova sekvence* je jednou z nejznámějších. Haltonova sekvence s dimenzí jedna se nazývá též Van der Corputova sekvence.

Prvek i Haltonovy sekvence dimenze jedna se základem b získáme jako hodnotu inverze podle řádové čárky z čísla i zapsaného v bázi b . Pomocí Haltonovy sekvence je možné generovat kvazináhodné vzorky v rámci jedné dimenze z intervalu $\langle 0, 1 \rangle$. Pokud potřebujeme generovat rovnoměrně vzorky z dimenze vyššího řádu (např. na ploše obdélníka), pak by vzorky jednotlivých souřadnic měly být získány z Haltonových sekvencí s různými bázemi. Vhodné dále je, aby tyto báze byly navzájem prvočíslí.

4.2.2 Vzorkování po částech

Cílem *vzorkování po částech* (angl. *stratified sampling*) je rovnoměrnější rozložení vzorků na intervalu $\langle 0, 1 \rangle$. Rovnoměrného rozložení se dosáhne pomocí rozdělení daného intervalu na N částí (angl. *strata*), přičemž v každé z těchto částí se nezávisle na ostatních částech vybere odpovídající počet vzorků.

Vzorkování po částech není vhodné pro použití ve vyšších dimenzích, neboť počet částí roste s dimenzí exponenciálně.

4.2.3 Rovnoměrně rozložené paprsky

Rovnoměrně rozložené paprsky (angl. *uniformly distributed lines*) hrají klíčovou roli nejen při výpočtu konfiguračních faktorů pomocí metody Monte Carlo, ale i při postupech, kdy se tyto konfigurační vůbec nepočítají.

Rovnoměrně rozložený paprsek s počátkem na plošce i je takový paprsek, jehož počáteční bod je vybrán náhodně s rovnoměrným rozložením na plošce i , a jehož směr je dán kosinovým rozložením vzhledem k normále plošky. Hustota pravděpodobnosti rovnoměrně rozloženého paprsku je tedy dána vztahem:

$$p(x, \Theta_x) = \frac{1}{A_i} \frac{\cos \theta_x}{\int_{\Omega(x)} \cos \theta_x d\omega_{\Theta_x}} = \frac{1}{A_i} \frac{\cos \theta_x}{\pi}. \quad (4.2)$$

Význam rovnoměrně rozložených paprsků pro výpočet radiozity pomocí Monte Carlo metod uvádí následující věta.

Věta 3.1. Konfigurační faktor F_{ij} mezi dvěma ploškami i a j ve scéně odpovídá hodnotě pravděpodobnosti p_{ij} , že rovnoměrně rozložený paprsek s počátkem na plošce i má nejbližší průsečík na plošce j .

Odvození této věty včetně důkazu je uvedeno v [3].

4.2.4 Stochastická Jakobiho iterační metoda

Stochastická Jakobiho iterační metoda náleží do třídy stochastických relaxačních metod [41, 35]. Mezi výhody této metody patří poměrně přehledný způsob výpočtu, dobré výsledky z hlediska rychlosti, relativně snadná implementace a možnost integrace s hierarchickou metodou.

V případě výpočtu lineárního systému rovnic $\mathbf{C}\mathbf{x} = \mathbf{e}$ (2.7) relaxační metodou se koeficienty c_{ij} vyskytují pouze ve formě součtů $\sum_j c_{ij}x_j$. Základním obratem stochastických relaxačních metod je odhadnutí těchto součtů jako celku bez nutnosti explicitního vyčíslení každého koeficientu c_{ij} . Při řešení radiozitivního problému v sobě koeficienty c_{ij} zahrnují konfigurační faktory F_{ij} . Velkou výhodou stochastických relaxačních metod tedy je, že není nutné vyčíslovat ani uschovávat v paměti hodnotu konfiguračních faktorů mezi ploškami scény.

Konfigurační faktory F_{ij} definují rozdělení pravděpodobnosti, které je možné s výhodou ovzorkovat aplikací rovnoměrně rozložených paprsků. Konkrétní hodnoty výše uvedených součtů $\sum_j c_{ij}x_j$ lze odhadnout na základě pouze části jejich členů, účinek ostatních členů v součtu je brán v úvahu díky faktu, že tyto členy *mohly* být při vzorkování vybrány. Díky tomu se časová náročnost výpočtu zredukuje z $O(n^2)$ na $O(n \log n)$, kde n je počet plošek ve scéně.

Vlastní výpočet stochastické Jakobiho metody je odvozen z klasické deterministické Jakobiho iterační metody. Jakobiho iterační metoda je velmi podobná Gauss-Seidelově relaxační metodě [38], s tím rozdílem, že relaxace všech hledaných členů probíhá najednou. V kroku k se pro všechny i provede:

$$x_i^{(k+1)} \leftarrow \left(e_i - \sum_{j \neq i} c_{ij} x_j^{(k)} \right) / c_{ii}. \quad (4.3)$$

Korekční vektor je pro iterační krok k definován vztahem:

$$\Delta x_i^{(k)} = \frac{r_i^{(k)}}{c_{ii}}.$$

Hodnotu rezidua poté lze vypočítat z následující rovnice:

$$r_i^{(k+1)} = r_i^{(k)} - \sum_j c_{ij} r_j^{(k)} / c_{jj} = - \sum_{j \neq i} c_{ij} \Delta x_j^{(k)}. \quad (4.4)$$

Regulární stochastická Jakobiho metoda je založena na řešení rovnic (4.3), *inkrementální stochastická Jakobiho metoda* řeší rovnice (4.4). Název inkrementální metody vystihuje postup výpočtu, kdy výsledek je získán jako součet přírůstků vypočtených v jednotlivých krocích. V následujícím textu bude popsána regulární stochastická Jakobiho metoda.

Regulární stochastická Jakobiho iterační metoda

Regulární stochastickou Jakobiho iterační metodu lze v případě vyzařovací metody použít k řešení jak shromažďovacího (angl. gathering), tak vystřelovacího (angl. shooting) systému rovnic. Pro první případ (2.5) dostaneme:

$$B_i^{(k+1)} \leftarrow E_i + \rho_i \sum_j F_{ij} B_j^{(k)}.$$

V případě vystřelovacího systému rovnic (2.6) dostaneme:

$$P_i^{(k+1)} \leftarrow \Phi_i + \rho_i \sum_j F_{ji} P_j^{(k)}.$$

Postup výpočtu vystřelovacího systému rovnic pomocí regulární stochastické Jakobiho iterační metody popisuje algoritmus 4.3. V algoritmu je použito *vzorkování po částech* (angl. *stratified sampling*).

Potřebný počet vzorků

Správná volba potřebného počtu vzorků v případě stochastické relaxační metody je velmi důležitá, neboť na počtu vzorků je závislá nejen kvalita řešení, ale i výpočetní náročnost metody. Podle [3] je třeba v jedné Jakobiho iteraci zvolit N vzorků dle vztahu:

$$N \approx 9 \frac{P_T}{\varepsilon^2} \cdot \max_i \frac{\rho_i(B_i - E_i)}{A_i},$$

kde ε je požadovaná přesnost výsledku a P_T je celková energie ve scéně. V případě n plošek ve scéně je tedy počet vzorků $O(n)$. Generování každého vzorku je složitosti $O(\log n)$, takže výsledná časová složitost algoritmu je $O(n \log n)$.

Místo předchozího vztahu je vhodnější použít heuristiku pro odhad počtu vzorků, kterou z předchozího vztahu odvodíme po dosazení $B_i - E_i = B_{av}$, $\varepsilon = B_{av}$, kde $B_{av} = \sum_i A_i B_i / A_T = P_T / A_T$ je vážený průměr z radiozit. Počet vzorků je poté dán nerovností:

$$N \approx 9 A_T \max_i \frac{\rho_i}{A_i} < 9 \max_i \frac{A_T}{A_i}, \quad (4.5)$$

kde A_T je celková plocha a A_i je plocha trojúhelníku i .

4.3 Hierarchická Monte Carlo vyzařovací metoda

Hierarchická Monte Carlo vyzařovací metoda vzniká jako spojení hierarchické a Monte Carlo metody výpočtu radiozity. Motivací pro tuto syntézu je spojení výhod obou metod. V minulosti se zabývalo několik autorů [28, 26, 25] spojením hierarchické metody s Monte Carlo výpočtem konfiguračních faktorů, v [48] lze nalézt spojení adaptivního dělení s metodou náhodného průchodu (angl. random walk method), kombinaci stochastické relaxační metody a hierarchické radiozity se věnuje [3].

Mezi výhodné vlastnosti *hierarchické* vyzařovací metody patří automatické adaptivní dělení scény a výrazná redukce počtu konfiguračních faktorů, které je třeba vyčíslit a uložit v paměti. K nepříjemným charakteristikám hierarchické metody naopak patří citlivost na chybu ve výpočtu konfiguračních faktorů, která vede v praxi k chybným, nebo dokonce zapomenutým energetickým přenosům. Paměťové nároky metody jsou sice $O(n)$, což je oproti klasické maticové metodě výrazné zlepšení, nicméně jsou stále poměrně vysoké.

Silnou stránkou *Monte Carlo* metody jsou nízké paměťové nároky, neboť konfigurační faktory není nutné explicitně vyčíslovat ani ukládat do paměti. Výhodou stochastických metod je též vyšší spolehlivost řešení. Spolehlivost řešení souvisí s paprsky vysílanými do scény, které jsou využívány přímo k přenosu energie, na rozdíl od metody hierarchické, kde se používají k odhadu zastínění páru plošek, resp. k výpočtu konfiguračních faktorů. Díky tomu jsou Monte Carlo metody méně citlivé na chybu v důsledku opomenutého nebo nepřesně vypočteného zastínění.

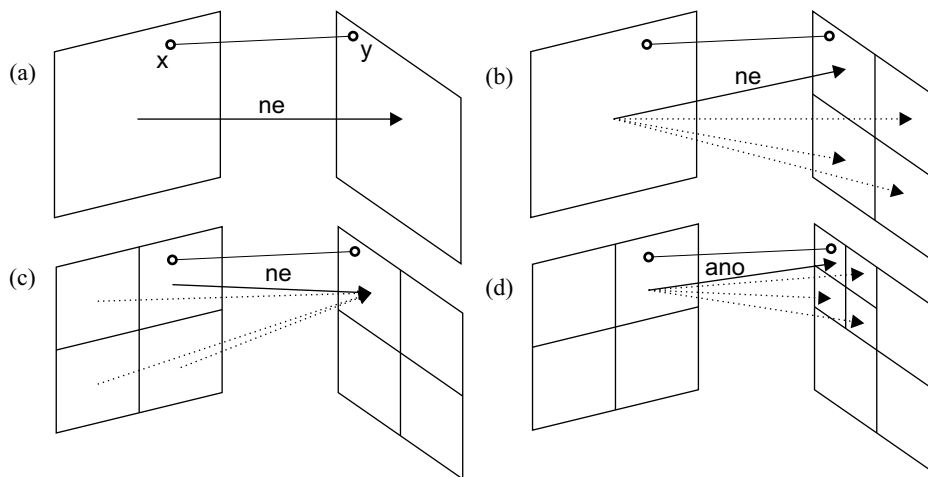
Algoritmus 4.3 Regulární stochastická Jakobiho iterační metoda

```
for každá ploška  $i$  do
     $P_i \leftarrow \Phi_i$ ,  $\delta P_i \leftarrow 0$ 
end for
 $P_T = \sum_i P_i$ ;  $P_T^{(old)} \leftarrow 0$ ;
while  $P_T - P_T^{(old)} \geq \varepsilon$  a počet kroků nepřekročil maximum do
    urči počet vzorků  $N$  dle vztahu (4.5)
    generuj náhodné číslo  $\xi \in (0, 1)$ ;
     $N_{prev} \leftarrow 0$ ;  $q \leftarrow 0$ ;
    for každá ploška  $i$  do
         $q_i \leftarrow \frac{P_i}{P_T}$ ;
         $q \leftarrow q + q_i$ ;
         $N_i \leftarrow \lfloor Nq + \xi \rfloor - N_{prev}$ ;
        for  $k = 0$  to  $N_i$  do
            vyber bod  $x$  na ploše  $A_i$  s rovnoměrným rozložením;
            vyber směr  $\Theta$  v bodě  $x$  s kosinovým rozložením;
            vrhni paprsek ve směru  $\Theta$  z bodu  $x$  a urči první plošku  $j$ , kterou paprsek protne
             $\delta P_j \leftarrow \delta P_j + \frac{1}{N} \varrho_j \frac{P_i}{q_i}$ 
        end for
         $N_{prev} \leftarrow N_{prev} + N_i$ 
    end for
     $P_T^{(old)} \leftarrow P_T$ ;
    for každá ploška  $i$  do
         $P_i \leftarrow \Phi_i + \delta P_i$ ;  $\delta P_i \leftarrow 0$ 
    end for
     $P_T = \sum_i P_i$ ;
    zobraz mezivýsledek pomocí hodnot  $P_i$ 
end while
```

4.3.1 Algoritmus hierarchické Monte Carlo metody

Ve stochastických metodách bez hierarchické struktury slouží každý paprsek vyslaný do scény k přenosu energie mezi ploškami, které spojuje. V případě hierarchické metody je během výpočtu vytvořena na plošce hierarchie elementů. Hierarchická metoda Monte Carlo kombinuje tyto dva přístupy.

Výpočet spočívá ve vrhání paprsků (přesněji rovnoměrně rozložených paprsků) do scény. Pro každý paprsek je zjištěna nejbližší ploška scény, kterou tento paprsek protíná, a souřadnice průsečíku s touto ploškou. Na základě těchto parametrů metoda rekurzivně vytvoří na ploškách, které obsahují koncové body paprsku, hierarchickou strukturu. Hierarchická struktura se tedy vytváří průběžně při výpočtu, na rozdíl od klasické hierarchické metody, kde je zkonstruována v jednom kroku. Postup rekurzivního vytváření této struktury je zapsán v dílčím algoritmu 4.4, jako funkce **refine**. Přenos energie se poté neuskuteční mezi původními ploškami, ale mezi dvěma elementy, které jsou listy v hierarchické struktuře a obsahují koncové body paprsku. Rozhodnutí, zda již je možné přenos uskutečnit (vazba je akceptovatelná), nebo je třeba provést další dělení ploch, zajišťuje podobně jako v případě hierarchické metody funkce *orákulum*.



Obrázek 4.3: Hierarchická Monte Carlo vyzařovací metoda

Po ukončení každé iterace výpočtu je nutné zajistit konzistenci všech vypočtených parametrů v rámci celé hierarchie. Toho je stejně jako v případě deterministické hierarchické metody dosaženo tzv. *pushPull* procesem. Během tohoto procesu jsou nově přijaté energie rekurzivně rozděleny „shora dolů“ (push) od rodičů k potomkům. Po dosažení listu se k vyzářené energii elementu přičte hodnota vlastní emise plošky a následuje postup „zdola nahoru“ (pull), kdy si každý rodičovský element upraví hodnotu vyzářené energie podle údajů od svých potomků. Spojení dříve popsané stochastické Jakobiho iterační metody s hierarchickou metodou je zapsáno jako algoritmus 4.5.

Na obrázku 4.3 je zobrazen příklad činnosti hierarchického Monte Carlo algoritmu. Algoritmus pomocí funkce *orákulum* zkoumá, jestli je možné akceptovat přenos energie mezi elementy, které obsahují koncové body x, y paprsku, jenž byl vyslán do scény (obr. 4.3a). Pokud není vazba přijata (přenos nebyl akceptován), metoda přistoupí k rozdělení plošky (obr. 4.3b). Tím vznikne sada nových vazeb, výpočet však pokračuje pouze pro vazbu mezi elementy, které obsahují koncové body x, y paprsku. Tento postup se rekurzivně opakuje, přitom

může být rozdělen kterýkoli z incidujících elementů (obr. 4.3c). Výpočet je pro daný paprsek ukončen v okamžiku, kdy je akceptována vazba mezi elementy (obr. 4.3d), dochází k přenosu energie.

4.3.2 Potřebný počet vzorků

V případě hierarchické Monte Carlo radiozity není možné pro odhad potřebného počtu vzorků odvodit heuristiku podobnou vztahu (4.5), neboť neznáme počet elementů, na které byla rozdělena původní scéna. Tento počet je jistě závislý na geometrii scény a na jejích optických vlastnostech, ale největší vliv na něj má kritérium dělení, které je realizováno ve funkci orákulum. Kritérium dělení bývá často implementováno tak, aby bylo po vazbách, které jsou akceptovány, přenášeno vždy přibližně stejné množství energie P_ϵ . Vzhledem k této hodnotě je proto vhodné zvolit i celkový počet vzorků:

$$N \approx \frac{P_T}{P_\epsilon}. \quad (4.6)$$

Algoritmus 4.4 Hierarchická Monte Carlo metoda – vytváření vazeb (refine)

```

refine (Patch p, Patch q, Point x, Point y, float  $F_\epsilon$ ) {
  if oracle(p, q,  $F_\epsilon$ ) then
    přenes energii mezi elementy p, q;
  else
    podle potřeby rozděl element  $p$  nebo  $q$ ;
     $s, t$  nechť jsou elementy obsahující koncové body  $x, y$  paprsku
    refine ( $s, t, x, y, F_\epsilon$ );
  end if
}
```

Algoritmus 4.5 Hierarchická stochastická Jakobiho metoda

```
for každá ploška  $i$  do
     $P_i \leftarrow \Phi_i, \delta P_i \leftarrow 0$ 
end for
 $P_T = \sum_i P_i; P_T^{(old)} \leftarrow 0;$ 
while  $P_T - P_T^{(old)} \geq \varepsilon$  a počet kroků nepřekročil maximum do
    urči počet vzorků  $N$  dle vztahu (4.6)
    generuj náhodné číslo  $\xi \in (0, 1);$ 
     $N_{prev} \leftarrow 0; q \leftarrow 0;$ 
    for každý element v hierarchii, který je listem  $i$  do
         $q_i \leftarrow \frac{P_i}{P_T};$ 
         $q \leftarrow q + q_i;$ 
         $N_i \leftarrow \lfloor Nq + \xi \rfloor - N_{prev};$ 
        for  $k = 0$  to  $N_i$  do
            vyber bod  $x$  na ploše  $A_i$  s rovnoměrným rozložením;
            vyber směr  $\Theta$  v bodě  $x$  s kosinovým rozložením;
            vrhni paprsek ve směru  $\Theta$  z bodu  $x$  a urči první plošku  $j$ , kterou paprsek protne
            zjistí souřadnice průsečíku  $y$  paprsku s ploškou  $j$ 
            refine ( $i, j, x, y, F_\epsilon$ )
        end for
         $N_{prev} \leftarrow N_{prev} + N_i$ 
    end for
     $P_T^{(old)} \leftarrow P_T;$ 
    for každý element v hierarchii, který je kořenem  $i$  do
        rekurzivně rozděl přijatou energii  $\delta P_i$  potomkům a naopak přičti k vyzářené energii  $P_i$ 
        elementu  $i$  hodnoty vyzářené jeho potomky (pushPull)
    end for
     $P_T = \sum_i P_i;$ 
    zobraz mezivýsledek pomocí hodnot  $P_i$ 
end while
```

Kapitola 5

Implementace

V předchozích kapitolách byla popsána vyzařovací metoda včetně postupů potřebných pro její implementaci, byly popsány způsoby modelace venkovního osvětlení. V této kapitole je uveden popis vlastní implementace hierarchické Monte Carlo vyzařovací metody v novém experimentálním systému *MCrad*. Zdrojový text výsledného programu je k dispozici na přiloženém CD.

Pro ověření teoretických poznatků popsaných v předchozích kapitolách byl navržen a implementován systém *MCrad*. Systém je založen na výpočtu globálního osvětlení pomocí hierarchické Monte Carlo vyzařovací metody a byl implementován v jazyce Java.

Nevýhoda implementace vyzařovací metody v prostředí Java je zřejmá: vlivem toho, že program zapsaný v jazyce Java je interpretován, dochází ke ztrátě výkonu oproti např. kompilovanému kódu jazyka C++. Tuto nevýhodu se snaží zmírnit technologie JIC (angl. Just In time Compiler – HotSpot) – předkompilace za běhu programu, možnost volání nativních (angl. native) metod a v případě Javy 3D i přímá vazba na OpenGL.

Mezi výhody prostředí Java patří standardizované API Java 3D, které umožňuje programátorovi zapsat poměrně přehledný kód, jenž je čitelný pro další potenciální zlepšovatele a pokračovatele projektu. Java 3D nabízí mnoho užitečných tříd a metod, které urychlí vývoj systému, viz odstavec 5.1. Další výhody plynou z vlastností samotného prostředí Java, mezi které patří snadná přenositelnost programů mezi platformami pomocí tzv. bytekódu, možnost modifikace projektu na úrovni tříd, přehlednost kódu atd.

Vzhledem k uvedeným vlastnostem implementace není ctížádostí systému *MCrad* dosažení minimálních výpočetních časů ve srovnání s podobnými systémy, ale spíše vytvoření přehledné testovací platformy pro zkoušení a experimentování s různými postupy globálního osvětlování.

5.1 Java 3D

Java 3D je API (viz příloha A) sloužící k vývoji programů v jazyce Java, které zobrazují 3D grafiku, zajišťují interakci s uživatelem ve virtuálním prostoru, přehrávají 3D zvuk apod. Toto aplikační rozhraní představuje ucelenou kolekci konstruktů (hierarchii tříd) sloužících k vytváření a manipulaci s 3D grafikou a zvukem. Třídy Java 3D jsou uloženy v balíčcích (angl. packages) `javax.media.j3d` (tzv. Java 3D core classes), `com.sun.j3d.utils` (tzv. Java 3D utility classes) a `javax.vecmath` (třídy pro vektorovou algebru a další matematické třídy). Kompletní instalaci poslední verze jakož i specifikaci rozhraní Java 3D lze získat na [45], úvodem do programování v prostředí Java3D se zabývá [6].

Java 3D bývá označována jako API čtvrté generace, přičemž za tři předchozí generace se považují:

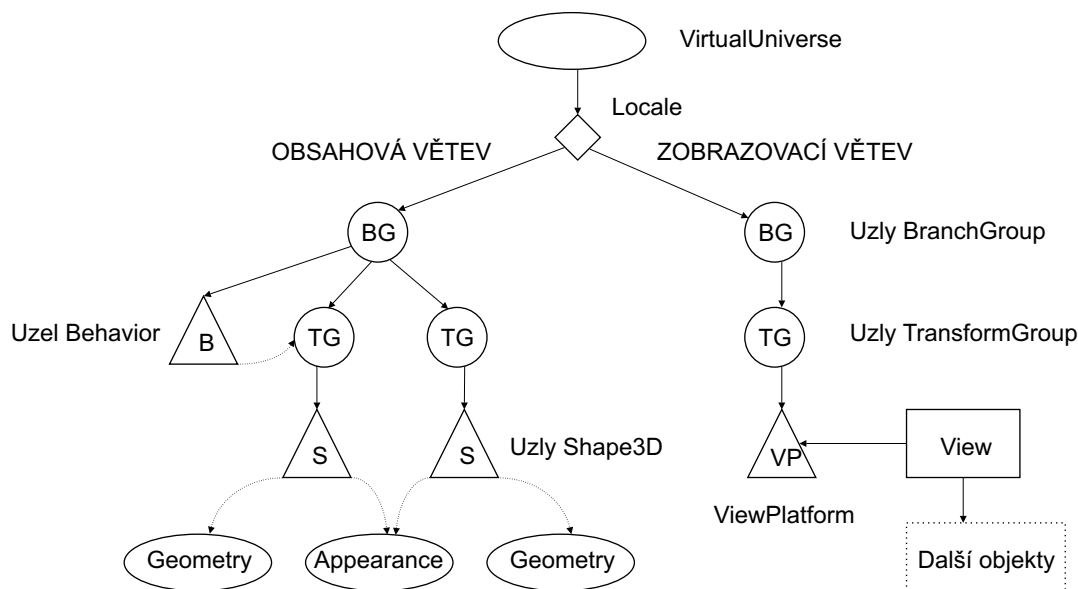
- Siggraph CORE, Fortran
- PHIGS/PEX, C
- Open GL, C++

Mezi společné vlastnosti grafických aplikačních rozhraní čtvrté generace patří to, že jsou založeny na technologii zvané graf scény.

5.1.1 Graf scény

Graf scény (angl. *scene graph*) je stromová datová struktura postavená na objektové filozofii, která reprezentuje scénu. Tuto stromovou strukturu je možné rychle zobrazovat i modifikovat. Změnu grafu scény mohou automaticky zajišťovat tzv. behavior rutiny, které byly poprvé použity v jazyce VRML [50]. Vyjmutí objektu, resp. mnoha objektů, ze zobrazovacího procesu je možné provést odebráním příslušné větve ze stromové struktury.

Graf scény obecně popisuje, *co* má být zobrazeno, nikoli však již, *jak* to má být zobrazeno. Architektura grafu scény totiž umožňuje např. zapojení mnoha procesů (resp. vláken – angl. threads), které mohou současně traverzovat stromovou strukturu za rozličnými cíli. Další vlastností grafu scény je možnost vícenásobného využití jednoho objektu (např. pro modelaci pneumatik automobilu nadefinujeme jeden objekt pneumatika, na který budou existovat čtyři reference) a obráceně též více definic jednoho objektu (vhodné např. pro implementaci techniky LOD, viz příloha A).



Obrázek 5.1: Příklad jednoduché stromové struktury grafu scény

V případě rozhraní Java 3D může současně existovat několik grafů scény, které se vkládají pomocí prvku *Locale* do tzv. virtuálního světa, objektu *VirtualUniverse*. Graf scény je tvořen instancemi Java 3D tříd a má stromovou strukturu, viz obr. 5.1. Tento graf se tedy skládá

ze *zobrazovací větve* (angl. *viewing branch*) a z *obsahové větve* (angl. *content branch*). V rámci zobrazovací větve je možné definovat několik výstupních zobrazovacích zařízení a specifikovat parametry zobrazení jako např. pozici a vlastnosti kamery. Obsahová větev definuje obsah celé scény, tzn. geometrii a umístění objektů, vlastnosti světelných zdrojů, intenzitu zvuků atd.

Konstrukty sloužící k vytvoření grafu scény v Javě 3D jsou následující:

- Třída **Node** je abstraktním předkem každého uzlu, který tvoří graf scény. Tato třída má za potomky třídy **Group** a **Leaf**.
- Třída **Group** je předkem tříd, které umožňují větvení stromové struktury grafu scény (**BranchGroup**, **OrderedGroup**, **SharedGroup**, **Switch**, **TransformGroup**).
- Třída **Leaf** je předkem tříd, které tvoří listy ve stromové struktuře grafu scény. Takové uzly umožňují specifikovat tvar objektu (**Shape3D**), zvuk (**Sound**), světelný zdroj (**Light**), behavior uzlu (**Behavior**), mlhu (**Fog**) apod.
- Třída **NodeComponent** není v pravém slova smyslu součástí grafu scény, neboť by znamenala porušení stromové struktury. Účelem této třídy je umožnit vícenásobné využití objektů. Na instance potomků této abstraktní třídy se proto mohou *odkazovat* objekty **Shape3D**, kterým tyto určují geometrii (**Geometry**), vzhled (**Appearance**), texturu (**Texture**), vlastnosti materiálu (**Material**) atd.

Terminologie Javy 3D využívá následujících výrazů: uzel je *živý* (angl. *live*), pokud je připojen (je součástí) grafu scény. Uzel je *zkompilovaný* (angl. *compiled*) do optimalizovaného formátu, pokud byla na příslušnou větev stromové struktury úspěšně volána metoda **compile**. Kompilace je nevratný proces sloužící k urychlení zobrazování složitých grafů scény.

K dalšímu urychlení zobrazování se využívá tzv. *capability* příznaků (angl. *capability bits*), které určují, jaké operace je možné s daným uzlem provádět poté, co se stane živým. K manipulaci s těmito příznaky slouží metody (**getCapability** a **setCapability**).

5.2 Volba metody výpočtu

Pro výpočet osvětlení vstupní scény byla zvolena *hierarchická Monte Carlo vyzařovací metoda*, popsaná v části 4.3. Tato metoda nabízí mimo již dříve popsaných výhod další výhodné vlastnosti pro výpočet venkovních scén. Monte Carlo hierarchickou metodu je možné rozšířit o výpočet zrcadlových odrazů a o paralelní světelné zdroje. Kvalitu výsledného řešení lze snadno škálovat pomocí funkce orákulum a daných parametrů metody. Rozšíření hierarchické Monte Carlo vyzařovací metody implementovaná v systému MCrad jsou popsána v sekci 5.6.

5.2.1 Vyplývající vlastnosti aplikace

Zvolená vyzařovací metoda pracuje s konstantní aproximací radiozitivní funkce na ploškách. Z této volby plynou následující předpoklady, a tedy i vlastnosti aplikace:

- vstupní scéna je rozdělena na plošky konečné velikosti
- všechny plošky jsou Lambertovské, tzn. každá ploška má na svém povrchu konstantní hodnotu radiozitivity

- okolní prostředí se neúčastní – neabsorbuje, neodráží ani nevysílá světlo (jedná se o vakuum)

5.3 Formát vstupních a výstupních dat

Program MCrad načítá vstupní scénu a ukládá výstupní scénu ve formátu *VRML* [20]. V současnosti je tento formát velmi rozšířen nejen v oblasti internetu a existují vhodné modelační programy, které umožňují výstup do tohoto formátu. Výhodou formátu VRML je, že danou scénu je možné prohlížet a hlavně procházet pomocí snadno dostupných prohlížečů. Mezi další výhody formátu VRML patří snadná čitelnost jeho textové formy a podpora ze strany API Java 3D.

Geometrie plošek, na které je rozdělena vstupní scéna, byla zvolena trojúhelníková. Trojúhelníkové plošky se v počítačové grafice používají často, neboť u trojúhelníků odpadají problémy s určováním konvexnosti, dochází k zjednodušení algoritmů a dalším výhodám. Jednou z takových výhod je to, že Gouraudova interpolace je v případě trojúhelníkového tvaru plošky nezávislá na její orientaci, na rozdíl od plošky čtyřúhelníkové. Další výhodou je poměrně jednoduchý způsob rozdělení plošky při implementaci adaptivního dělení ploch.

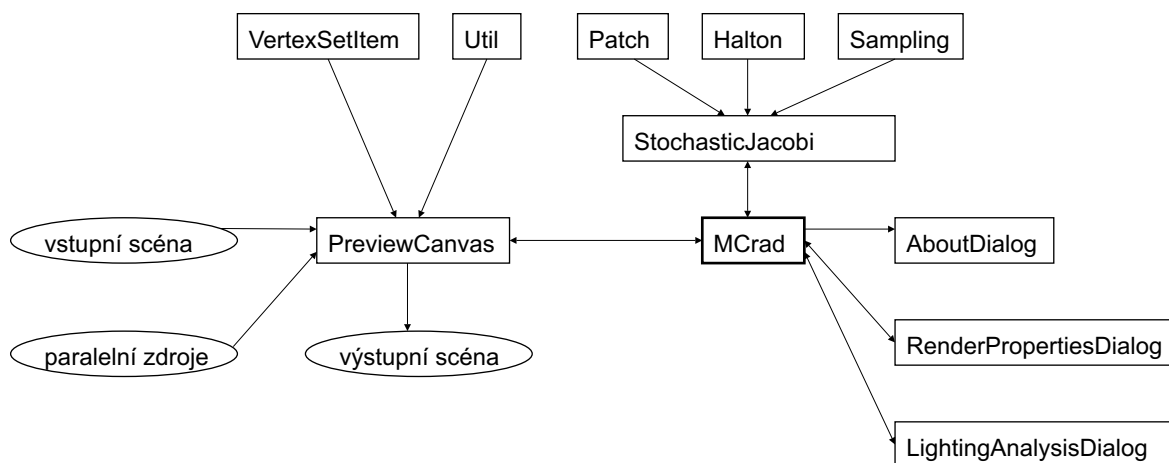
Podrobný popis formátu vstupních dat je uveden v příloze B, popis jazyka VRML je možné nalézt v [50].

5.4 Systém MCrad

Vše, co je popsáno v této sekci, bylo naprogramováno a je k dispozici na přiloženém CD, viz příloha D.

5.4.1 Struktura systému

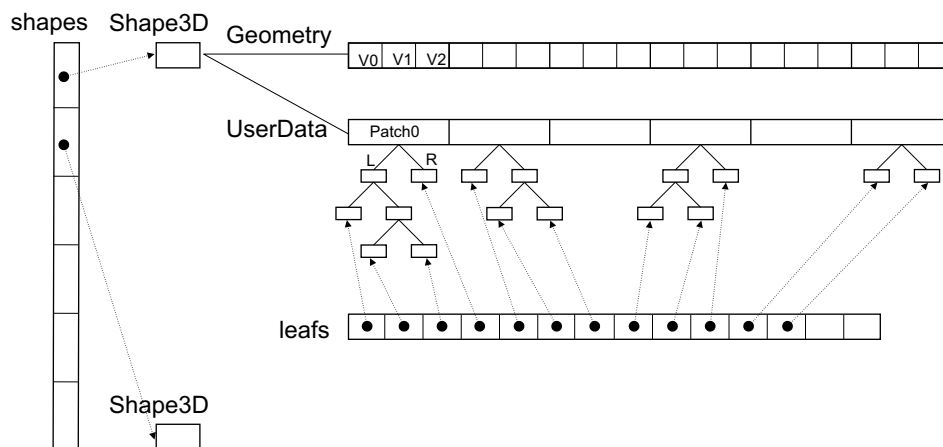
Na obr. 5.2 je zobrazena struktura systému MCrad. Při návrhu byla oddělena zobrazovací část (třída *PreviewCanvas*) a výpočetní část (třída *StochasticJacobi*) od vlastní aplikace (třída *MCrad*) s roletovým menu, dialogy atd.



Obrázek 5.2: Struktura systému MCrad

Funkčnost systému zajišťují následujícími třídy:

- Třída **AboutDialog** slouží k definici dialogu, který zobrazuje informace o programu.
- Třída **Halton** slouží ke generování kvazináhodných čísel z Haltonovy sekvence. K výpočtu prvků sekvence se používá metoda **next**, která realizuje velmi rychlý algoritmus popsáný v [24].
- Třída **MCrad** je hlavní třídou celého systému. Jako jediná obsahuje metodu **main**, která inicializuje a spouští systém MCrad. Třída dále obsahuje metodu **numberOfSamples**, která vrací odhad počtu vzorků.
- Třída **Patch** reprezentuje trojúhelníkovou plošku, která je dána svými třemi vrcholy. Na základě těchto vrcholů jsou vypočteny následující údaje: normála plošky, těžiště, velikost povrchu a souřadný systém na plošce. O každé plošce jsou dále udržovány následující parametry: difúzní odrazivost ϱ , zrcadlová odrazivost ϱ_s , celková energie plošky P , celková energie dopadlá na plošku P_g , pravý a levý potomek při adaptivním dělení. V třídě **Patch** jsou navíc implementovány metody pro barycentrickou parametrizaci (**barycentric**) a porovnávání plošek (**comparePatches**).
- Třída **PreviewCanvas** představuje „třírozměrné kreslicí plátno“, do kterého je zobrazována vstupní a po dokončení výpočtu též vypočtená scéna. Třída zajišťuje inicializaci grafu scény pro použití při výpočtu, nastavení potřebných capability příznaků atd. Dále umožňuje různé režimy zobrazení výsledku – interpolace, zobrazení v barvách dle zatížení, zobrazení sítě.
- Třída **RenderPropertiesDialog** slouží k zobrazení dialogu, kde je možné změnit nastavené parametry výpočtu a zobrazení, viz obr. C.2.
- Třída **LightingAnalysisDialog** slouží k zobrazení dialogu s fyzikálními informacemi o proběhlém výpočtu. Dialog dále umožňuje zadat hodnotu horní meze intervalu zobrazovaných hodnot vypočtených radiozit, viz obr. C.3.
- Třída **Sampling** poskytuje metody pro rovnoměrné a kosinové vzorkování plošky. Rovnoměrné vzorkování plošky realizuje funkce **uniform** pomocí barycentrické parametrizace. Generování vzorků s kosinovým rozložením vzhledem k normále plošky realizuje metoda **cosTheta**. Vstupními parametry metody jsou osy lokálního souřadného systému na plošce (osa z odpovídá normále plošky) a dvojice vzorkovacích parametrů. Výstupem metody je vzorek – vektor udávající směr paprsku.
- Třída **StochasticJacobi** zajišťuje vlastní výpočet osvětlení vstupní scény. Na začátku výpočtu je vytvořeno nové vlákno (angl. **thread**), které poté provádí výpočet.
- Třída **Util** poskytuje metody pro převody mezi barevnými modely.
- Třída **VertexSetItem** reprezentuje prvek v setříděné množině vrcholů, která se využívá při výpočtu průměrných hodnot energie ve společných vrcholech plošek. Pro daný vrchol se uschovávají jeho souřadnice, velikosti povrchů všech incidujících plošek (kolekce **powAreas**) a průměrná hodnota jeho energie.



Obrázek 5.3: Hierarchie plošek vytvářená během výpočtu

5.4.2 Reprezentace scény

Vstupní scéna je načtena z VRML souboru, jehož struktura je popsána v příloze B. Veškerá načtená data jsou poté uložena do datové struktury grafu scény. Během výpočtu hierarchickou vyzařovací metodou se díky implementovanému adaptivnímu dělení (viz níže) v paměti vytváří další datová struktura, stromová hierarchie elementů, viz obr. 5.3. Zobrazitelné objekty jsou tvořeny instancemi třídy **Shape3D**. Tyto instance jsou součástí aktivního grafu scény a je možné je tedy zobrazovat. Objekty **Shape3D** uschovávají informaci pouze o své geometrii pomocí pole se souřadnicemi vrcholů, což není dostačující pro výpočet vyzařovací metodou. Jako uživatelská data (**UserData**) jsou proto k objektům **Shape3D** připojeny objekty **Patch**, které dále tvoří stromovou strukturu. Každá instance třídy **Patch**, neboli každá ploška, obsahuje ukazatele na svého pravého a levého potomka (byla-li dotyčná ploška rozdělena).

Z důvodu zvýšení rychlosti výpočtu se dále udržuje kolekce listů (**leafs**) ve stromové struktuře. V kolekci **shapes** jsou udržovány odkazy na všechny aktivní objekty **Shape3D**, aby k nim byl přístup rychlejší (jinak by bylo nutno vždy při požadavku na dotyčný objekt traverzovat celý graf scény).

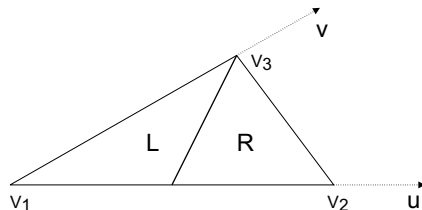
5.4.3 Postup výpočtu

Výpočet probíhá podle dříve popsaného algoritmu 4.5 s nově zavedenými rozšířeními, které jsou popsány v odstavci 5.6. Po příchodu požadavku na spuštění výpočtu (pomocí položky v roletovém menu aplikace) vytvoří objekt třídy **MCrad** nové vlákno (angl. thread), které realizuje třída **StochasticJacobi** a začne výpočet pomocí vrhání rovnoměrně rozložených paprsků. Funkcionalita rovnoměrně rozložených paprsků je zajištěna pomocí metod Javy3D. Pro urychlení vyšetřování průsečíku se scénou je využita technika ohraničujících těles (angl. bounding boxes). Pro daný paprsek se tedy nejprve vyšetřuje, jestli protíná hraniční těleso nějakého objektu scény. Pokud paprsek hraniční těleso neprotíná, neprovádí se již další, výpočetně mnohem náročnější, test na průsečík se skutečnou geometrií tělesa. Protne-li paprsek hraniční těleso, tento test je samozřejmě nutné provést.

Po dokončení výpočtu je výsledná scéna zobrazena pomocí konstantní aproximace hodnot energie na ploškách. Uživatel systému má možnost zvolit jiný druh zobrazení, viz kapitolu 5.7.

5.5 Metoda dělení

Implementovaná metoda adaptivního dělení vychází z předpokladu, že veškeré plošky scény jsou trojúhelníkového tvaru. Za účelem dosažení co možná nejvyšší hodnoty aspect ratio (viz odstavec 3.2.2) se provádí dělení trojúhelníkové plošky rozpůlením její nejdelší hrany, viz obr. 5.4. Takto vzniknou dvě menší trojúhelníkové plošky, které jsou v hierarchii umístěny o úroveň níže než ploška původní. Dělení plošek zajišťuje v programu metoda `subdiv`.



Obrázek 5.4: Dělení plošky

Zásadní vliv na počet a charakter dělení má funkce *orákulum* (v programu metoda `oracle`) a parametry maximální přípustné hodnoty konfiguračního faktoru mezi ploškami F_ε a hranice minimální velikosti plošky A_ε , v programu realizované proměnnými `F_EPS` a `A_EPS`. Funkce *orákulum* je založena na výpočtu odhadu konfiguračního faktoru dvou plošek, viz odstavec 4.1. Pokud je daný odhad větší než zadaná hodnota F_ε , přistupuje se k dělení plošek. K dělení plošky skutečně dochází pouze v případě, že velikost plošky je větší, než hodnota A_ε . V případě eliminace T-vrcholů se pak dělení provádí dále rekurzivně pro sousední plošku, přičemž k zjištění této sousední je využita lokální topologická informace o sousednosti plošek.

5.5.1 Lokální topologická informace

Poté, co jsou geometrická data načtena ze souboru a vytvoří se graf scény, je volána metoda `initNeighbours`, která inicializuje u načtených plošek scény, objektů `Patch`, lokální topologickou informaci. Tato informace je pro každou plošku tvořena třemi ukazateli na „sousední“ plošky. Mají-li tedy dvě plošky A_i a A_j společnou hranu, znamená to, že jsou sousední – ploška A_i obsahuje odkaz na plošku A_j a obráceně.

Popsaná topologická informace se využívá při rekurzivní metodě dělení plošek volané za účelem eliminace T-vrcholů, viz níže. Během rekurzivního procesu se údaje o sousedních ploškách stávají dočasně nekonzistentními, což není na závadu, neboť adaptivní dělení nemůže být v průběhu výpočtu prováděno více procesy najednou.

5.5.2 Eliminace T-vrcholů

Po každém rozdělení plošky vzniká nový vrchol, který se může stát tzv. T-vrcholem a způsobovat problémy při interpolaci (viz odstavec 3.4.1). Za účelem eliminace takových vrcholů je v programu volána funkce `subdivT`, která rekurzivně provádí dělení plošek tak, aby T-vrcholy nevznikaly. Pokud adaptivně rozdělená ploška A_i sdílí společnou hranu s jinou ploškou A_j a právě tato hrana byla rozdělena, tj. vznikl na ní T-vrchol, provádí se rekurzivní dělení dále pro plošku A_j . Proces končí v případě, že daná ploška nemá sousední plošku, nebo sousední ploška je již rozdělena, takže na společné hranici již existuje T-vrchol.

5.6 Rozšíření metody výpočtu

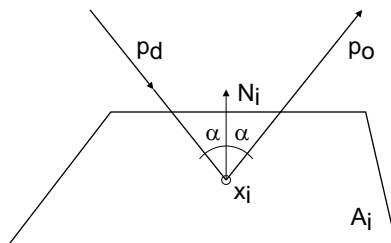
Specifika světelných poměrů ve venkovních scénách si vyžádala několik rozšíření dříve popisované hierarchické Monte Carlo vyzařovací metody. Implementovaná vyzařovací metoda uvažuje pouze difúzní odrazy ve scéně a jen jeden druh světelných zdrojů – difúzní světelné zdroje. S takto omezenými možnostmi není vizualizace světelného zatížení v exteriéru prakticky možná, je nutné implementovat alespoň zdroje rovnoběžného světla a uvažovat též zrcadlové odrazy. Dalším problémem venkovních scén je jejich značná geometrická složitost a velká vzdálenost objektů od světelných zdrojů. Tato sekce popisuje řešení uvedených aspektů pomocí rozšíření hierarchické Monte Carlo metody tak, jak byla implementována v systému MCrad.

5.6.1 Začlenění zdrojů paralelního světla

Rozšíření stochastické metody, která je založena na vrhání rovnoměrně rozložených paprsků (viz odstavec 4.2.3), o paralelní světelné zdroje konečné velikosti je implementačně poměrně nenáročné. Výpočet podle algoritmu 4.5 proběhne pro plošku paralelního světelného zdroje obdobně jako pro plošku zdroje difúzního, s následující změnou: v okamžiku vzorkování paprsku s počátkem na dané plošce A_i difúzního zdroje vybereme počáteční bod paprsku s rovnoměrným rozdělením na celé ploše A_i stejně jako v případě zdroje difúzního, ale směrnici paprsku vůbec nevzorkujeme, nýbrž ji přímo ztotožníme s normálou plošky A_i . Další výpočet probíhá již v souladu s algoritmem 4.5.

5.6.2 Zrcadlové odrazy

Správné uvážení zrcadlových odrazů je velmi potřebné pro zachování co možná nejvyšší korepondence výsledku s realitou. V případě venkovních scén, ve kterých se často vyskytují např. okna, výkladní skříně, skleněné úpravy povrchů a další vysoce směrově odrazivé objekty, to platí dvojnásob.



Obrázek 5.5: Zákon odrazu

Rozšíření metody výpočtu (viz algoritmus 4.5) o zrcadlové odrazy spočívá v implementaci rekurzivní metody sledování paprsku. Tato metoda je volána po nalezení průsečíku x_i dopadlého rovnoměrně rozloženého paprsku p_d s ploškami scény v případě, že ploška A_i , na které leží bod x_i má nenulovou hodnotu zrcadlové odrazivosti ϱ_S . Metoda na základě známé směrnice dopadlého paprsku a normály plošky $\vec{N}_i$ rekurzivně vysílá další paprsek p_o podle zákona odrazu, viz obr. 5.5. Množství energie P_o , které přenáší nově vyslaný paprsek je dáno množstvím energie dopadlého paprsku P_d a zrcadlovou odrazivostí plošky ϱ_S : $P_o = \varrho_S P_d$. Rekurze končí v okamžiku, kdy se vyslaný paprsek protne s ploškou, jejíž zrcadlová odrazivost je nulová ($\varrho_S = 0$), nebo v případě, že hloubka rekurze již přesahuje explicitně danou hodnotu.

Sledování paprsku v programu MCrad zajišťuje rekurzivní funkce `rayTrace`, která se nachází ve třídě `StochasticJacobi`. K nastavení hloubky rekurze slouží celočíselná konstanta `RAYTRACE_DEPTH`.

5.6.3 Světelná iterace

Rozdíl mezi osvětlením scény globálními zdroji světla (Slunce, obloha – převážně paralelní zdroje světla) a zdroji lokálními (žárovky, zářivky – převážně difúzní zdroje světla) je natolik veliký, že se z hlediska výpočtu jeví jako vhodné je oddělit. Z toho důvodu byla výpočtu podle algoritmu 4.5 předřazena ještě jedna, nultá, *světelná iterace*.

V průběhu světelné iterace jsou uváženy příspěvky všech paralelních zdrojů světla načtených ze vstupního souboru (viz příloha B). Při výpočtu nulté iterace se využívá funkce `oracleLighting`, která se liší od funkce `oracleLighting` používané v dalších, běžných iteracích. Rozdíl spočívá v tom, že se při odhadu konfiguračního faktoru neuvažuje vzdálenost plošek, která je v případě globálních zdrojů teoreticky nekonečná. Navíc tato funkce využívá parametr `FL_EPS`, který má obecně jinou číselnou hodnotu než `F_EPS`.

Pozměněna je i funkce pro dělení plošek (`refineLighting`), která provádí dělení pouze pro plošky scény. Plošky globálních světelných zdrojů dělit není nutné. Po „vystřílení“ veškeré energie globálních světelných zdrojů se provádí pozměněný rekurzivní `pushPull` proces (viz odstavec 4.3.1), který zajistí konzistenci hodnot energií na všech úrovních hierarchické struktury a zjištěné výsledné hodnoty uloží jako hodnoty vlastní emise Φ příslušným ploškám scény. Při dalších iteracích výpočtu již globální světelné zdroje nejsou uvažovány, jejich účinek je zajištěn vlastními emisemi plošek a difúzní odrazy od Slunce, resp. oblohy, lze zanedbat.

Celou světelnou iteraci v programu MCrad zajišťuje metoda `lightingIteration` ve třídě `StochasticJacobi`. Přístup podobný právě popsanému, použitý v kombinaci s metodou postupující radiozity je popsán v [34].

5.6.4 Akumulace dopadlé energie

Množství skutečně dopadlé energie P_g na plošku A_i není obecně totožné s množstvím energie P_i , které ploška vyzařuje, resp. odráží. Rozdíl spočívá v tom, že energii P_i vyzařovanou ploškou A_i ovlivňují odrazivosti plošky ϱ a ϱ_s . Tato skutečnost koresponduje se zkušeností z reálného světa, kde se pozorovateli také jeví více odrazivé těleso jasnější, než těleso málo odrazivé. Při vizualizaci světelného zatížení však potřebujeme znát hodnotu dopadlé energie P_g .

Během výpočtu jsou proto pro každou plošku scény udržovány obě hodnoty (P_i i P_g). Hodnota P_g je akumulována beze změny aktuálně dopadlé energie, zatímco hodnota P_i je multiplikována faktorem ϱ . Po dokončení výpočtu je tedy možné zvolit způsob zobrazení energie na ploškách – rozlišujeme zobrazení energie P (angl. power), tzn. zobrazení scény přibližně tak, jak by vypadala ve skutečnosti a zobrazení dopadlé energie P_g , tzn. zobrazení světelného zatížení (angl. light load).

5.7 Zobrazení

V této sekci budou popsány některé postupy z oblasti zobrazování implementované do systému MCrad. Tyto postupy jsou v programu realizovány pomocí metod třídy `PreviewCanvas`. Dané metody předpokládají, že v kolekci `leafs` jsou uloženy plošky s již vypočtenými hodnotami energií.

5.7.1 Zobrazení sítě

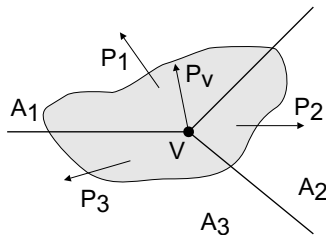
Zobrazení sítě umožňuje uživateli získat představu o dosažitelné přesnosti výpočtu při zobrazení původní sítě před výpočtem, resp. o konečné síti vzniklé adaptivním dělením sítě původní, viz obr. 6.4. V rámci programu zajišťuje zobrazení sítě metoda **drawMesh**, která za tímto účelem vytvoří novou větev grafu scény. Tato nová větev reprezentuje hrany sítě, jež jsou zobrazovány jako červené úsečky.

5.7.2 Zobrazení vypočtených hodnot

Systém MCrad umožňuje zobrazit vypočtené hodnoty pomocí konstantní aproximace, kdy je zachována konstantní hodnota energie na celé plošce, a pomocí lineární interpolace, která zaručuje parametrickou spojitost C^0 (viz odstavec 3.4).

Zobrazení hodnot energií pomocí *konstantní aproximace* v rámci plošky zajišťuje metoda **renderPatchesHierarchical**, která zkonstruuje nový graf scény z právě vypočtených hodnot. Těmito hodnotami jsou energie plošek, které tvoří listy v hierarchické struktuře plošek jež vzniká v důsledku adaptivního dělení během výpočtu, v kolekci **leafs**.

Zobrazení pomocí *lineární interpolace* hodnot energií na ploškách je postup obdobný tzv. interpolaci barvy [51]. Hodnota energie v daném vrcholu se získá jako váhový průměr hodnot energií na ploškách incidujících s tímto vrcholem, viz obr. 5.6 (v obrázku jsou velikosti energií reprezentovány délkou vektoru). Při výpočtu se bere v úvahu hodnota zvolené odchylky – crease angle (viz odstavec 5.7.3). Energetické hodnoty ve vrcholech jsou dále převedeny na stupnici šedi, resp. barev (viz 5.7.4), a zobrazeny pomocí Gouraudova stínování [51].

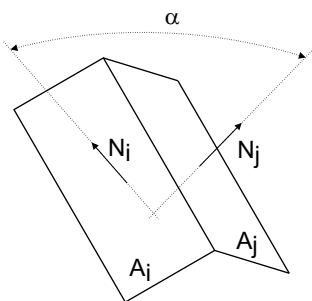


Obrázek 5.6: Interpolace hodnot energie na ploškách

Interpolaci hodnot energií zajišťuje metoda **drawInterpolation** ve třídě **PreviewCanvas**. Vzhledem k tomu, že z vypočtených dat není možné získat přímou informaci o ploškách incidujících s daným vrcholem, musí tato metoda potřebné údaje dopočítat. Během tohoto výpočtu se vytváří pomocná stromová struktura (na základě standardní Java třídy pro uspořádanou množinu hodnot – **TreeSet**) vrcholů, která je lexikograficky uspořádána [18]. Pro každý vrchol je udržována kolekce incidujících plošek, na základě které je po vytvoření celé struktury možné vypočítat průměrnou hodnotu energie daného vrcholu.

5.7.3 Crease angle

Vstupní scéna je načtena ze souboru ve formě objektů složených z trojúhelníkových sítí. U takových geometrických dat chybí informace o topologii, zejména není známo, které hrany těles jsou skutečné a které jsou pouze pomocné. *Skutečná* (ostrá) hrana je opravdová hrana tělesa, která je zobrazována; *pomocná* hrana je spojnice mezi rovinnými ploškami, které aproximují



Obrázek 5.7: Úhel mezi normálami dvou plošek

nějaký zakřivený povrch, pomocná hrana se nezobrazuje. Rozhodnutí o typu hrany leží tedy až na zobrazovací fázi.

V případě konstantní aproximace hodnot energií na ploškách je situace jednoduchá, neboť stačí vykreslit všechny vypočtené trojúhelníkové plošky konstantními barvami. Složitější je zobrazení při interpolaci hodnot energií, neboť je třeba klasifikovat všechny plošky, které incidují s daným bodem (vrcholem) a ovlivňují tak jeho barvu. Jinými slovy je třeba rozhodnout, jestli dané dvě plošky náleží do jedné aproximované plochy a jejich společná hrana je pouze pomocná, nebo jsou součástí dvou ploch a jejich společná hrana je tedy skutečná. Takovéto rozhodnutí získáme na základě porovnání úhlu α , který mezi sebou svírají normály dvou plošek A_i a A_j , se zvolenou odchylkou, viz obr. 5.7. Pokud je úhel menší než daná odchylka, považujeme plošky za součást jedné plochy a hranu mezi nimi za pomocnou. Tato předem zvolená odchylka, která prakticky určuje mez, kdy bude hrana považována za pomocnou či skutečnou, se anglicky nazývá *crease angle*.

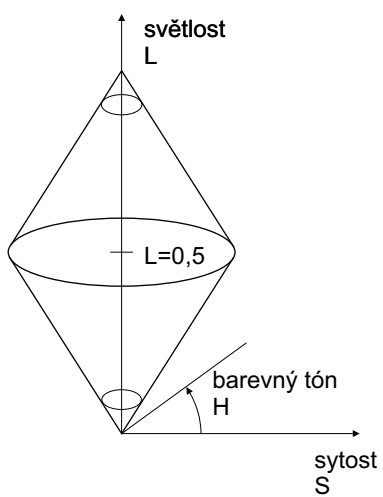
V programu je uschovávána hodnota kosinu této odchylky v proměnné `COS_CREASE_ANGLE` ve třídě `PreviewCanvas`.

5.7.4 Transformace hodnot energie na barevnou stupnici

Při vizualizaci světelného zatížení je často výhodné zobrazit původně monochromatické hodnoty energie barevně. Na výsledném obrázku, viz obr. 6.2, jsou v takovém případě výrazněji patrné oblasti s vysokým světelným zatížením, oblasti s podobným množstvím dopadlé energie atd. V anglické literatuře se zobrazení v takovýchto „falešných“ barvách říká *false color*.

V programu je převod hodnot energie na barevnou škálu realizován s pomocí barevného modelu HLS, viz obr. 5.8. V modelu HLS je barva definována trojicí složek: *H* (angl. hue) – barevný tón, *L* (angl. lightness) – světlost, *S* (angl. saturation) – sytost. Vlastní převod se uskutečňuje pro hodnoty $L = 0,5$, $S = 0,5$, *H* odpovídá velikosti dané energie. Pro nízké hodnoty energie takto získáme „studené“ barevné tóny, pro vyšší hodnoty energie pak barvy „teplejší“.

Popsaný převod na barevnou stupnici realizuje v programu metoda `HSV2RGB` třídy `Util`.



Obrázek 5.8: Barevný model HLS

Kapitola 6

Výsledky

V této kapitole jsou na ilustračních obrázcích diskutovány výstupy a za pomoci sady testovacích scén jsou otestovány vlastnosti implementovaného návrhu, experimentálního systému MCrad.

6.1 Použité technické vybavení

Výpočty všech scén uvedených v této diplomové práci se uskutečnily na počítači PC s procesorem AMD Duron 650 MHz a 384 MB paměti RAM. Výstup na monitor zajišťovala grafická karta S3 ViRGE DX/GX se 4 MB paměti bez 3D akcelerace. Použitý operační systém byl Microsoft Windows 2000.

6.2 Nastavení parametrů metody

Vysoké nároky kladené na uživatele systémů, které realizují výpočet pomocí vyzařovací metody, jsou jedním z důvodů, proč se tyto systémy poměrně obtížně aplikují v praxi. V případě systému MCrad je pro dosažení dobrých výsledků vhodná znalost následujících parametrů výpočtu:

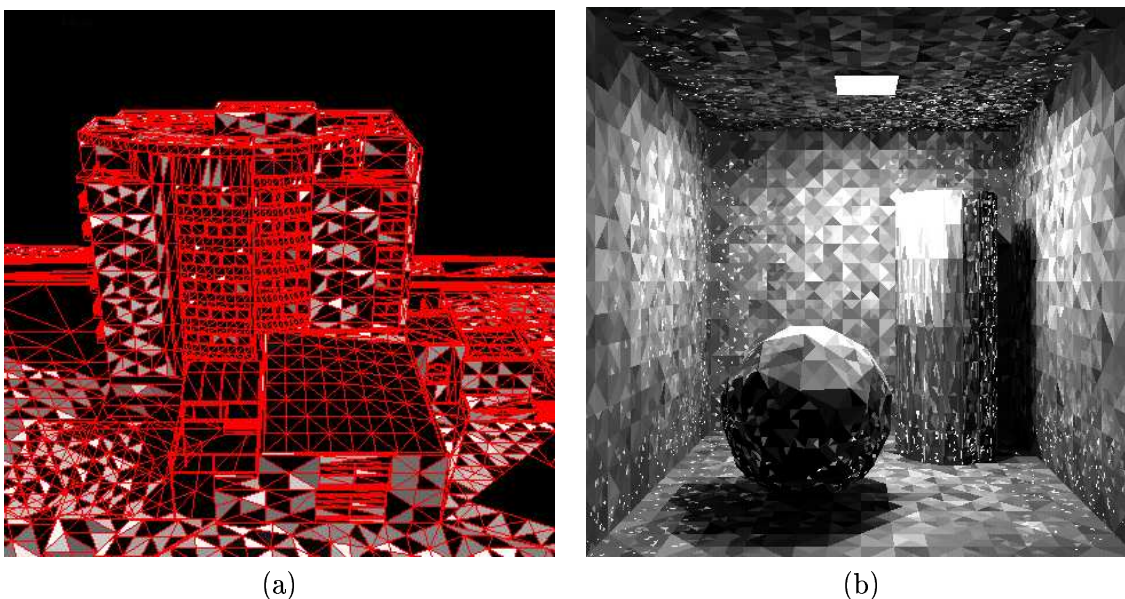
Počet iterací	má vliv na vyšetření světelných odrazů ve scéně; pokud je převážná část scény přímo osvětlena světelnými zdroji, postačí často dvě až tři iterace; pokud je však podstatná část scény osvětlena světlem nepřímým, je třeba hodnotu tohoto parametru zvýšit
F_ϵ (F_EPS)	nejvyšší akceptovatelný odhad konfiguračního faktoru; tento parametr má zásadní vliv na počet vzniklých plošek při adaptivním dělení; zjednodušeně lze říci, že vyšší hodnota tohoto parametru znamená menší počet vzniklých plošek; na správné nastavení tohoto parametru je implementovaná metoda obzvláště citlivá
F_{l_ϵ} (F1_EPS)	tento parametr je obdobou parametru F_ϵ pro světelnou iteraci

A_ε (A_EPS)	určuje minimální plochu plošky, kdy se již neprovádí její dělení; pokud je hodnota parametru F_ε velmi nízká, dochází k zastavení dělení až na základě hodnoty A_ε , takže vzniklé plošky scény jsou plošně téměř stejně velké, vzniká tedy téměř uniformní síť
P_ε (P_EPS)	určuje maximální hodnotu energie, jejíž přenos již lze akceptovat; vzhledem k tomu, že hodnota tohoto parametru figuruje ve vztahu pro počet vzorků N (4.6), určuje v podstatě množství paprsků vyslaných do scény; čím je hodnota P_ε nižší, tím více bude vysláno paprsků

Dále následuje několik příkladů nevhodně nastavených parametrů metody.

6.2.1 Nevhodné nastavení parametrů metody

Vlivem nevhodně nastavených parametrů může výpočet trvat zbytečně dlouhou dobu nebo výsledná síť může obsahovat přílišné množství plošek. Nevhodně nastavené parametry mohou však zapříčinit dokonce vznik šumu a artefaktů ve výsledné scéně.



Obrázek 6.1: Vliv nevhodně nastavených parametrů na kvalitu výsledku

Příliš nízká hodnota parametru P_ε v kombinaci s nízkou hodnotou parametru F_ε vede k vyslání nedostatečného počtu paprsků do scény, která je však rozdělena na vysoký počet plošek. Ve výsledku se proto vyskytuje množina poměrně malých plošek, které byly zasaženy paprskem a mají relativně vysokou hodnotou energie, a množina ostatních plošek, které vlivem malého počtu paprsků zasaženy nebyly (resp. byly zasaženy malým počtem), takže hodnota jejich energie je nízká. Příklad takové scény je uveden na obr. 6.1a, pro větší názornost je na obrázku zobrazena také výsledná síť.

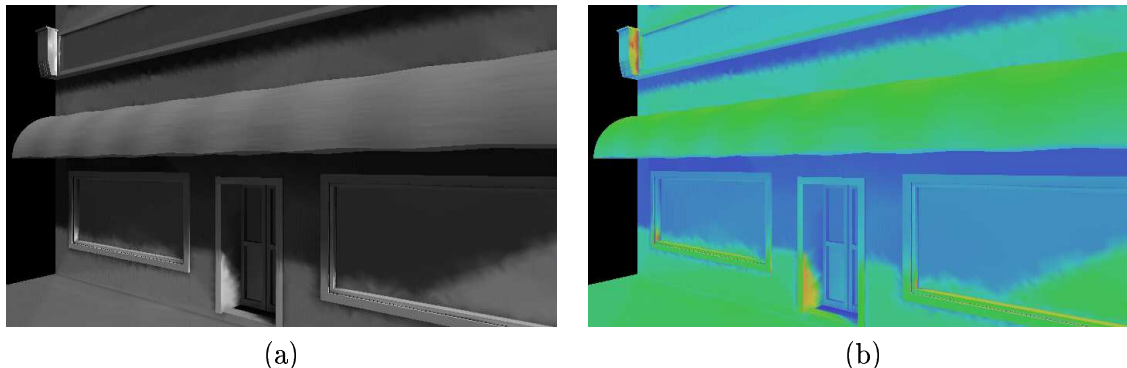
Podobná situace nastane i v případě, že zvolíme příliš nízkou hodnotu F_ε současně s velmi malou hodnotou minimálního povrchu A_ε . Vlivem implementované funkce orákulum pak zejména u blízkých objektů (např. v rozích místnosti) dochází k velmi jemnému dělení sítě,

které opět vede ke vzniku šumu, viz obr. 6.1b. Aby byl šum v těchto obrázcích dostatečně patrný, bylo při zobrazení scén použito pouze konstantní aproximace energie na ploškách.

6.3 Ilustrační scény

6.3.1 Transformace hodnot energie na barevnou stupnici

Na obrázku 6.2a je zachycena scéna, ve které jsou vypočtené hodnoty energie zobrazeny jako stupně šedi. Tuto scénu je možné zobrazit také ve formě tzv. obrazu ve falešných barvách (angl. false color image), viz obr. 6.2b.



Obrázek 6.2: Zobrazení (a) monochromatické (b) pomocí „falešných“ barev

6.3.2 Zobrazení energie a světelného zatížení

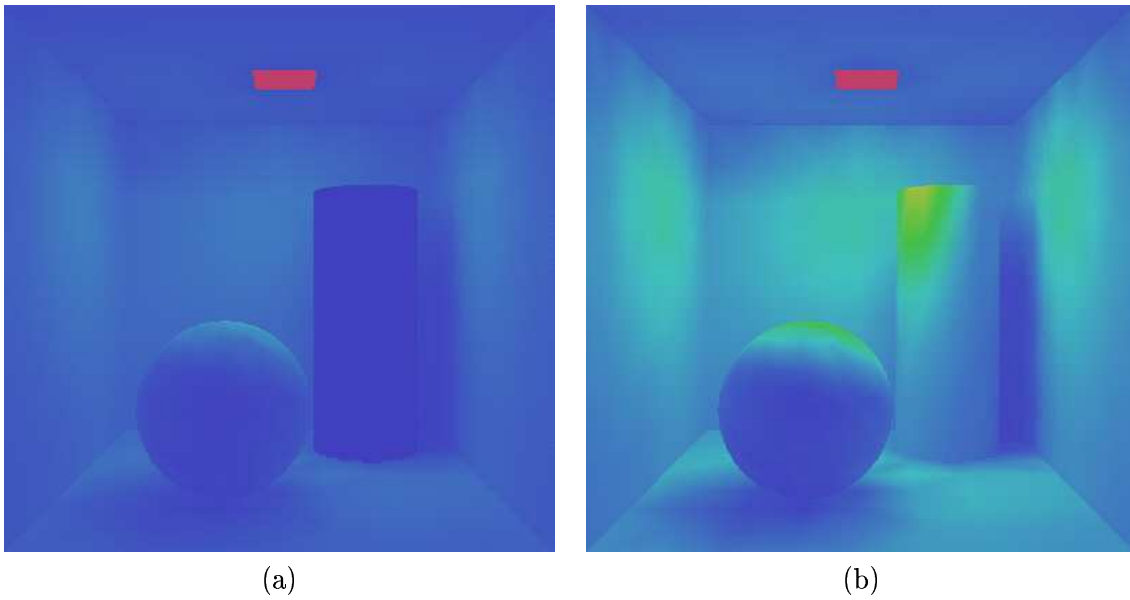
Na obrázku 6.3 jsou zachyceny dva možné způsoby reprezentace vypočtených hodnot – zobrazení vyzařované energie a zobrazení světelného zatížení (dopadlé energie). Zobrazení vyzařované energie dokumentuje obr. 6.3a – jednoduchá vstupní scéna s nastavenými záměrně nízkými hodnotami odrazivosti všech ploch se pozorovateli jeví jako tmavá, neboť tělesa převážnou část světelné energie absorbují. Energie *vyzařovaná* jednotlivými tělesy je tedy malá.

Množství energie skutečně *dopadlé* na dané objekty scény je však výrazně vyšší, takovou informaci získáme pomocí zobrazení v režimu světelného zatížení. U reálného pozorovatele toto zobrazení nemá ekvivalent, ale je vhodné např. pro zjištění účinků světla na fasády domů apod.

Příkladem výše popsaného je válec ve vstupní scéně, který má nulovou difúzní odrazivost, $\rho = 0$. Při zobrazení vyzařované energie i ve skutečnosti se válec jeví jako černé těleso, které nevyzařuje žádnou energii, viz obr. 6.3a. Množství dopadlé energie na válec však nulové není, viz obr. 6.3b.

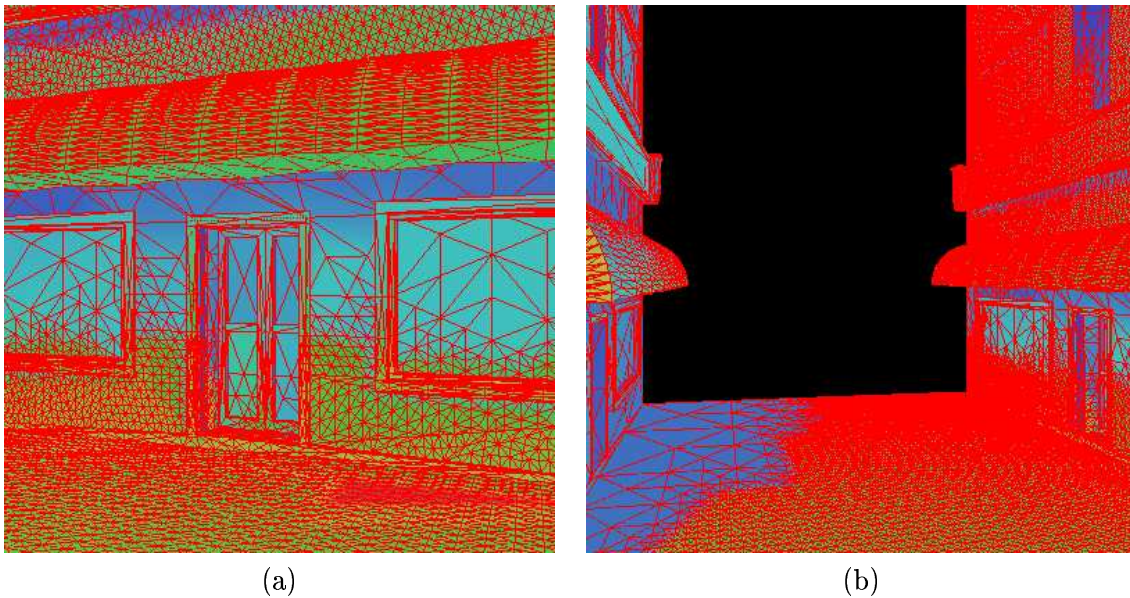
6.3.3 Zobrazení sítě

Obrázek 6.4 ilustruje možnost systému MCrad zobrazit síť plošek, ze kterých je daná scéna složena. Na obrázku je zobrazená scéna pro jednu pozici Slunce pomocí falešných barev současně s výslednou sítí trojúhelníků. Při bližším pohledu na scénu, na obr. 6.4a, je



Obrázek 6.3: Zobrazení (a) vyzařované energie (b) světelného zatížení

patrné, že síť vzniklá adaptivním dělením s eliminací T-vrcholů skutečně T-vrcholy neobsahuje.



Obrázek 6.4: Zobrazení sítě

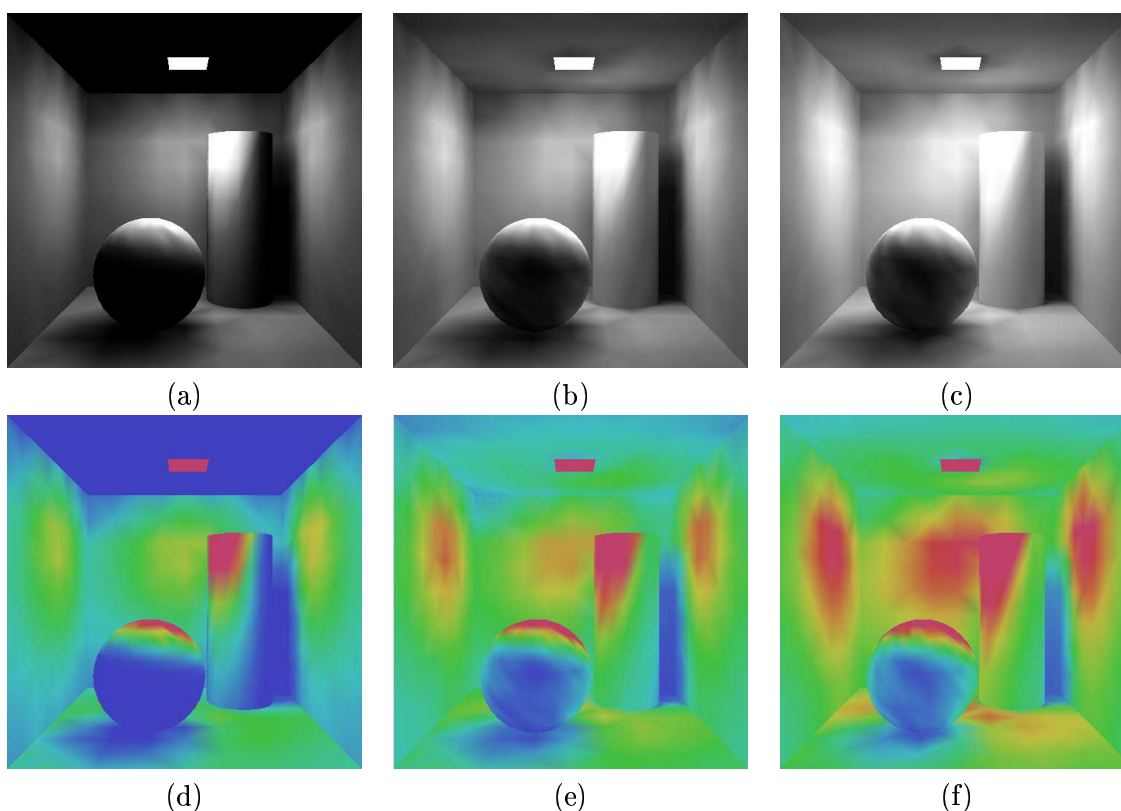
Další vlastnost implementovaného adaptivního dělení ploch dokumentuje obr. 6.4b – v oblastech s vyšší hodnotou dopadlého zářivého toku, neboli ve více osvětlených oblastech, dochází k jemnějšímu dělení než v oblastech s nižšími hodnotami zářivého toku, kterými jsou např. zastíněné části scény. V důsledku toho je výsledná síť v osvětlených oblastech hustší

a obsahuje více detailů, než síť v oblastech stínů.

6.3.4 Postup výpočtu

První tři iterace výpočtu jednoduché scény zachycuje obr. 6.5. Po dokončení první iterace, viz obr. 6.5a, d je scéna vypočtena pouze pro primární osvětlení, v konkrétním případě pro jediný plošný zdroj světla na stropě. V této fázi nemůže žádný paprsek protnout plochu stropu, a proto je tato plocha celá černá.

Během dalších iterací, viz obr. 6.5b, c se scéna obohacuje o stíny a osvětlení vznikající odrazem od ploch, které nejsou primárně světelnými zdroji. Vzhledem k charakteru objektů scény, zejména jejich odrazivosti, je více či méně patrné mírné „rozjasnění“ scény, které je způsobeno právě sekundárními světelnými odrazy. U metod typu klasický ray tracing se pro zjednodušení právě popsaná vlastnost nahrazuje tzv. okolním světlem, resp. světelným šumem (angl. ambient light), který má v rámci scény konstantní hodnotu a je přičten ke každé ploše scény.

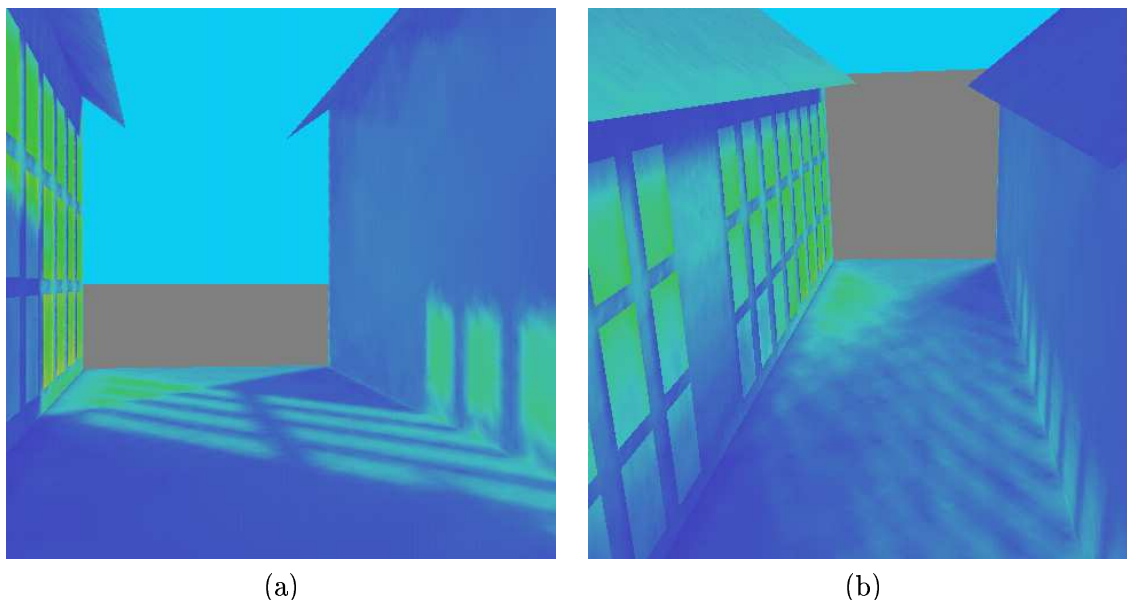


Obrázek 6.5: Výsledná scéna po dokončení (a, d) první iterace (b, e) druhé iterace (c, f) třetí iterace

6.3.5 Zrcadlové odrazy

Na obrázku 6.6 jsou zobrazeny výsledky výpočtu jednoduché scény ulice. Na levé straně ulice se nachází budova s okny, jejichž koeficient odrazu je $\rho = 0,85$, pravou část tvoří stěna druhé

budovy bez oken. Na obr. 6.6a je scéna vypočtena pro jednu pozici Slunce. Na podložce a pravé stěně budovy jsou zřetelné zrcadlové odrazy od nezastíněných oken. Obrázek 6.6b zobrazuje výsledek výpočtu jednoduché scény pro velmi krátkou trajektorii Slunce.



Obrázek 6.6: Zrcadlové odrazy v jednoduchém modelu ulice (a) konstantní pozice Slunce (b) krátká trajektorie Slunce

6.3.6 Scéna „mrakodrapy“

Scéna *mrakodrapy* se skládá z pěti výškových budov. Jedna z budov, levý přední mrakodrap, má koeficient zrcadlového odrazu ρ_s větší než 0, ostatní plochy jsou pouze difúzní. Budovy jsou reprezentovány sítí 4500 plošek, což není mnoho. Scéna je poměrně jednoduchá, nicméně je rozsáhlá, a proto výsledná síť obsahuje cca 70000 plošek.

Výsledek výpočtu scény pro konstantní polohu Slunce uvádí obr. 6.8a, b. Na obr. 6.8a je dobře patrný odlesk od nezastíněné části budovy. Na obrázcích 6.8c, d je scéna vypočtena pro trajektorii Slunce. Změna úhlu dopadu slunečních paprsků je patrná např. na odlescích od skla levého předního mrakodrapu. Představu o tvaru sítě po dokončení třetí iterace výpočtu umožňuje získat 6.8f, původní (vstupní) síť je zachycena na obr.6.8e.

6.3.7 Scéna „ulice“

Ilustrační scéna *ulice* je modelem ulice, která sestává ze čtyř budov uspořádaných do běžné řadové zástavby. Síť scény se skládá z 12424 plošek, část této sítě je zobrazena na obr. 6.9f. Výpočet každé z uvedených variant scény trval v průměru 5h, během kterých byla scéna adaptivně rozdělena na přibližně 60000 plošek.

Scénu vypočtenou pro jednu pozici Slunce s barevným zvýrazněním světelného zatížení zachycuje obr. 6.9a, b. Na obr. 6.9a je zobrazen pohled „z okna“ na protější osvětlenou stranu ulice, obr. 6.9b zachycuje zastíněnou část ulice.

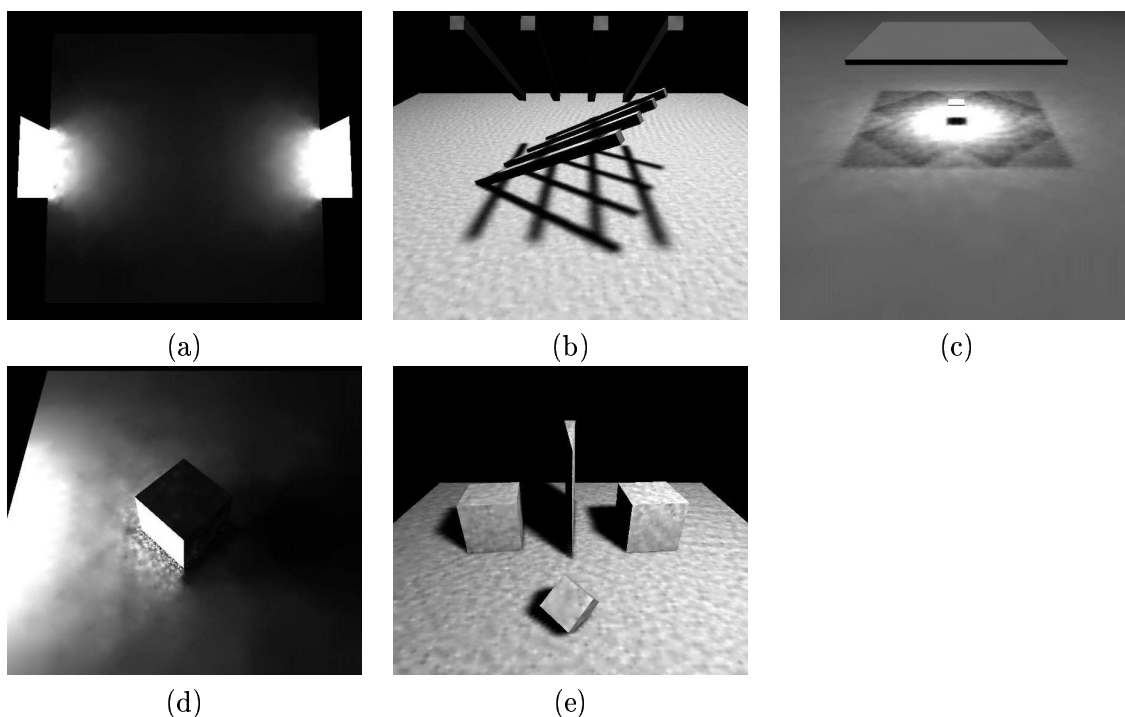
Obrázky 6.9c, d ilustrují změnu světelných poměrů v ulici v důsledku změny fasády jednoho z domů. Původní dům s běžnou fasádou, viz obr. 6.9c, byl nahrazen novým, převážně proskleným objektem, viz obr. 6.9d. Výrazné zvýšení světelného zatížení je patrné zejména na chodníku poblíž nového objektu, ale mírná změna v osvětlení je viditelná i na protější budově.

Výsledek výpočtu pro trajektorii Slunce ilustruje obr. 6.9e. Ulice i směr pohledu jsou orientovány na východ. Trajektorie Slunce byla zadána pomocí devíti poloh Slunce a odpovídala časovému úseku cca 6h od východu Slunce.

6.4 Global Illumination Test Scenes

Tyto scény [42] vznikly za účelem testování programů pro výpočet globálního osvětlení. Každá z nich se zaměřuje na jiné aspekty výpočtu osvětlení a všechny se snaží daný aspekt prověřit opravdu důkladně (přestože se každá z nich při pohledu na výsledný obrázek zdá velmi jednoduchá). Scény jsou monochromatické a jsou v nich použity pouze dva materiály: difúzní reflektor a difúzní emitör. Jména scén odpovídají názvům v anglickém originálu.

Všechny scény byly upraveny pro použití v systému *MCrad*. Úpravy spočívaly v transformaci scén na soustavu trojúhelníků a v některých případech i v úpravě hodnot emisí světelných zdrojů.



Obrázek 6.7: Global Illumination Test Scenes (a) Emission test (b) Shadow test (c) Large spatial extent (d) Complex illumination paths (e) Geometric accuracy

6.4.1 Emission test

Tato scéna je složena ze dvou čtvercových zdrojů světla osvětlujících podložku. Levý zdroj je modelován jedním polygonem, zatímco pravý je složen ze sta polygonů (10×10). Scéna testuje, zda je výsledek osvětlení daného 100 malými zdroji stejný jako osvětlení z jednoho velkého zdroje.

Výsledný obrázek (obr. 6.7a) by měl být symetrický vzhledem k svislé ose.

6.4.2 Shadow test

Tato scéna testuje přesný výpočet stínů. Obsahuje jeden světelný zdroj, osm dlouhých kvádrů zastíňujících světlo a podložku. Kvádry jsou natočeny tak, že jejich vzdálenost k podložce se mění.

Výsledek zachycuje obr. 6.7b. Vzhledem k tomu, že cílem této scény je otestování výpočtu stínů, byla vypočtena pouze pro primární osvětlení.

6.4.3 Large spatial extent

Tato scéna je prostorově velice rozsáhlá, ale kamera je zaměřena pouze na velmi malou část scény. Scéna je tvořena velkou podložkou, nad kterou je umístěn zdroj světla, jenž je stíněn stejně velkou deskou. Pod touto deskou je umístěn desetkrát zmenšený podobný zdroj světla se stínící deskou. Většina světelné energie ve scéně (99%) pochází tedy od velkého zdroje světla, avšak převážná část energie viditelná z pohledu kamery je dána zdrojem malým.

Výsledek výpočtu této scény je zobrazen na obr. 6.7c. Na hranicích stínu od velkého zdroje světla je patrné neuspokojivé rozdělení podložky na velké trojúhelníky. Přenosy energie od velkého zdroje světla zjevně proběhly na příliš vysoké úrovni hierarchie. Tyto přenosy se uskutečnily v důsledku toho, že v implementované funkci *orákulum* je rozhodnutí o dalším dělení dvou plošek založeno na odhadu konfiguračního faktoru, ve kterém figuruje vzdálenost těchto dvou plošek. Vlivem velkého rozdílu vzdálenosti podložky od velkého a malého zdroje světla pak došlo k uvedenému přenosu. Lepšího výsledku by zřejmě bylo možné dosáhnout snížením závislosti výsledku funkce *orákulum* na vzdálenosti plošek.

6.4.4 Complex illumination paths

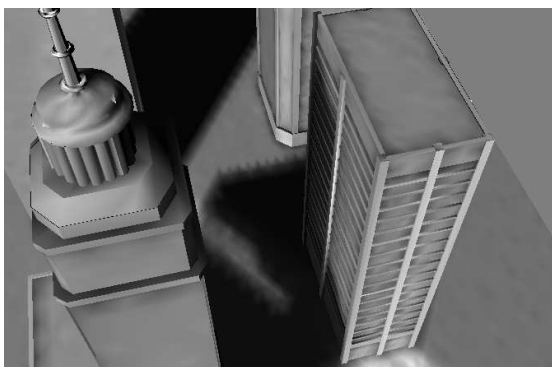
Tato scéna se skládá z podložky, pod kterou je umístěn zdroj světla. Zdroj svítí na difúzní plochu, která je umístěna vedle podložky. Na podložce se nachází krychle. Kamera je orientována tak, že veškeré viditelné osvětlení je vlivem difúzní plochy. Na krychli může světlo dopadnout až po minimálně třech interakcích s difúzním povrchem.

Nedostatkem vypočtené scény na obr. 6.7d je šum na podložce poblíž hran krychle. Tento šum je způsoben tím, že vlivem malé vzdálenosti bodů na podložce a stěně krychle poblíž hrany krychle dochází k jemnému dělení trojúhelníků, ale počet paprsků dopadlých na tyto malé plošky není dostačující. Lepšího výsledku by mohlo být dosaženo zvýšením počtu paprsků vysílaných do scény, za cenu zvýšení časových nároků. Další možností jak zlepšit výsledek je zvýšení meze A_ϵ pro minimální povrch plošky, což vede na hrubší dělení, které do jisté míry zamezí vzniku šumu.

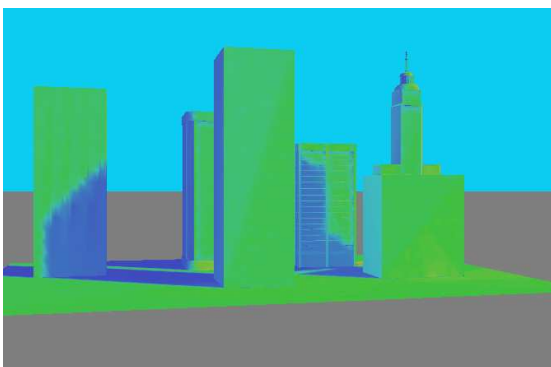
6.4.5 Geometric accuracy

Tato scéna je určena k prověření přesnosti výpočtů průsečíků paprsku a scény. Scéna obsahuje tři krychle, klín, podložku a světelný zdroj. Levá krychle je ponořena do podložky o 1% své výšky. Pravá krychle je o stejnou výšku nad podložku zdvižena. Jeden vrchol prostřední krychle je pod podložkou. Klín je natočen tak, aby byl vidět z úzké strany.

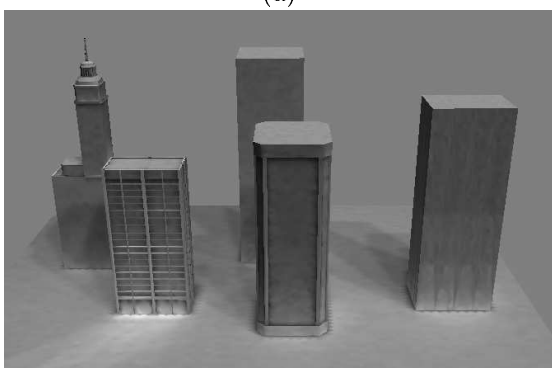
Na obr. 6.7e je zachycena vypočtená scéna. Tato scéna obsahuje šum, který byl zaviněn nedostatečným počtem paprsků vyslaných do scény. Dále jsou na obrázku patrné nepříjemné artefakty, které vznikly jako důsledek rozdělení scény bez znalosti hranic stínů. Stín klínu je např. patrný i na nasvícené straně klínu. Eliminace těchto artefaktů je poměrně obtížná, jednou z možností je implementace discontinuity meshingu.



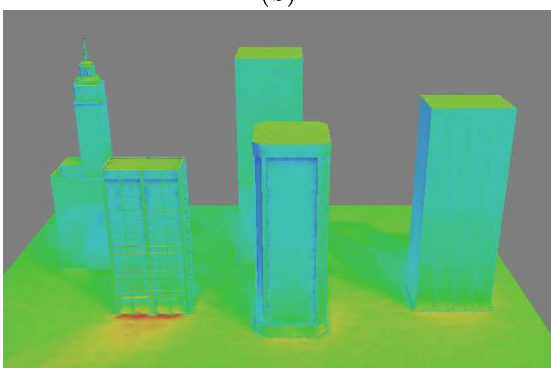
(a)



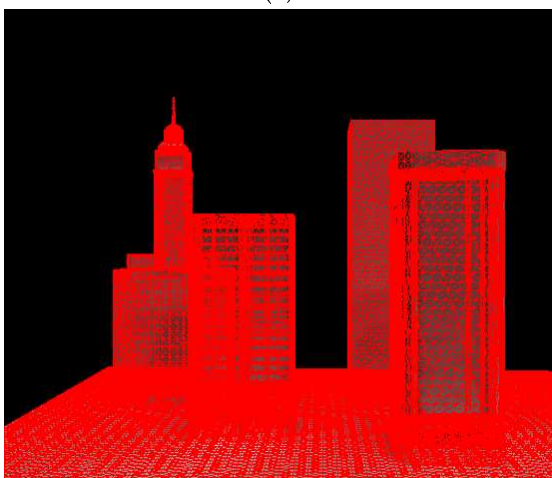
(b)



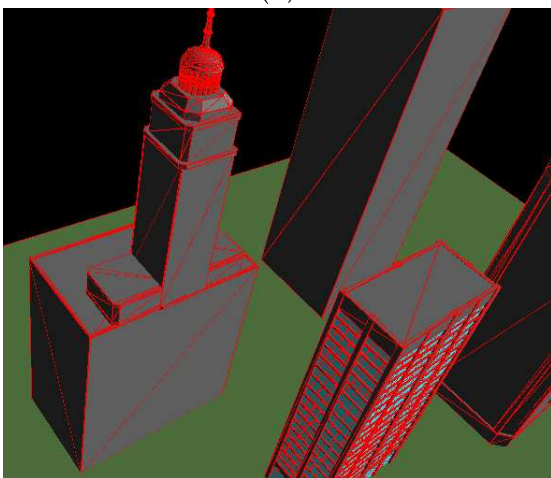
(c)



(d)

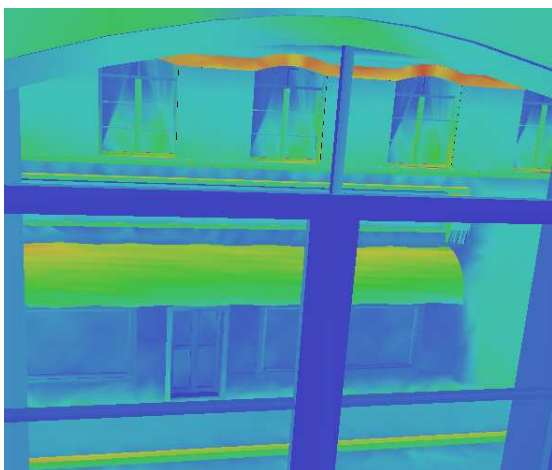


(e)



(f)

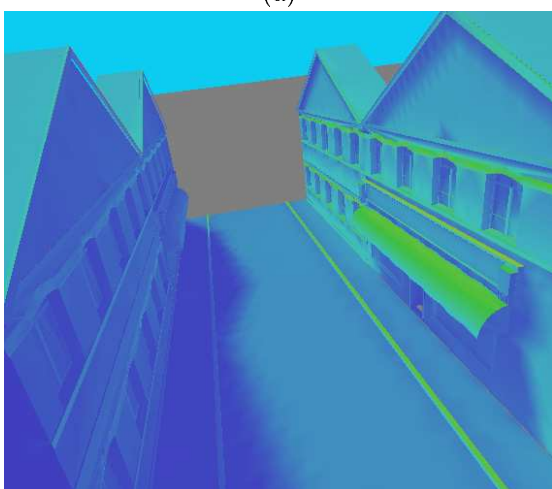
Obrázek 6.8: Ilustrační scéna „mrakodrapy“



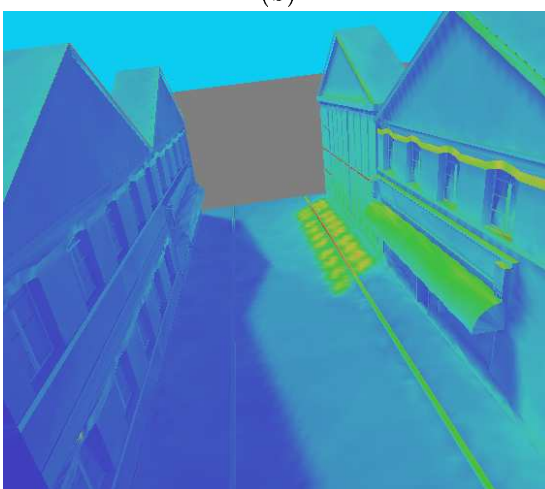
(a)



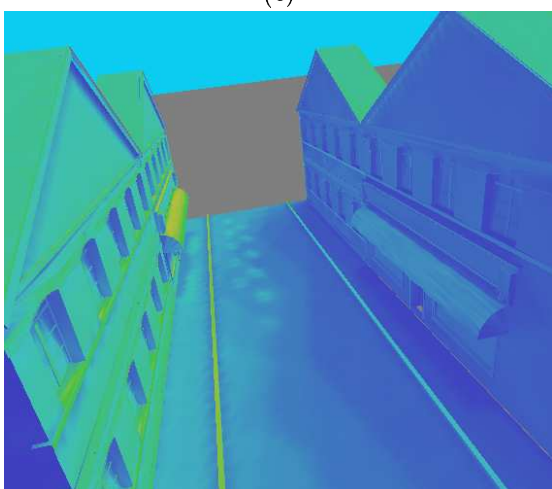
(b)



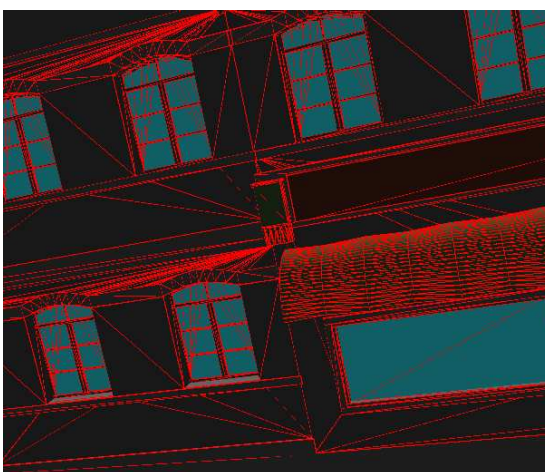
(c)



(d)



(e)



(f)

Obrázek 6.9: Ilustrační scéna „ulice“

Kapitola 7

Závěr

Tato diplomová práce se zabývá problémem vizualizace světelného zatížení venkovních scén. K řešení daného problému se nabízejí metody realistické syntézy obrazu. Nejprve bylo proto nutné nastudovat a klasifikovat postupy známé v této oblasti počítačové grafiky. Obecně lze postupy realistické syntézy rozdělit na pohledově závislé (např. zpětné sledování paprsku) a pohledově nezávislé (např. vyzařovací metoda). Vzhledem k požadavku pohledové nezávislosti výsledné scény byla k řešení použita vyzařovací metoda.

Vyzařovací metody se vyznačují vysokou paměťovou a výpočetní složitostí. Přístupů, kterak snížit tyto nároky, zná současná realistická syntéza obrazu několik. Obrat použitý v hierarchické vyzařovací metodě, stejně jako využití stochastických principů při výpočtu, patří k těm nejpovedenějším. Kombinací těchto dvou přístupů vzniká hierarchická vyzařovací metoda Monte Carlo. Tato metoda byla nastudována, podrobně popsána a aplikována při řešení daného problému.

Specifické vlastnosti venkovních scén si vyžádaly rozšíření hierarchické Monte Carlo vyzařovací metody o nové postupy. Metoda byla rozšířena o uvážení zdrojů rovnoběžného světla a zrcadlových odrazů. Dále byla při výpočtu použita nová, tzv. nultá světelná iterace, která umožnila nastavit různé parametry výpočtu pro principiálně rozdílné umělé (např. žárovka) a přirozené (např. Slunce) světelné zdroje.

V rámci práce byl navržen a implementován experimentální radiozitní systém *MCrad*, který realizuje výpočet globálního osvětlení pomocí hierarchické Monte Carlo vyzařovací metody. Systém byl odladěn proti chybám a otestován na sadě testovacích scén, nicméně jeho vývoj může nadále pokračovat. Experimentální povaha systému spolu se zvoleným implementačním jazykem Java tomu rozhodně nebrání.

Zajímavý námět na další práci, resp. výzkum, nabízí datová struktura grafu scény. Na základě této struktury by mohly být odvozeny nové, nekonvenční způsoby urychlení výpočtu. Kvalitu výstupů programu by výrazně zlepšilo začlenění discontinuity meshingu, dostatek prostoru pro experimentování poskytuje funkce orákulum. Zrychlení výpočtu by mohla přinést implementace shlukování (angl. clustering) plošek scény a zejména pak optimalizace vrhání paprsků.

Literatura

- [1] *CIE Recommendations on Uniform Color Spaces, Color-difference Equations, and Psychometric Color Terms*. Bureau de la CIE, Paris, 1978.
- [2] Autodesk. Lightscape.
<http://www.lightscape.com>, 2001.
- [3] P. Bekaert. *Hierarchical and Stochastic Algorithms for Radiosity*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 1999.
- [4] P. Bekaert. Renderpark a Photorealistic Rendering Tool.
<http://www.cs.kuleuven.ac.be/cwis/research/graphics/RENDERPARK>, 2000.
- [5] Bentley Systems, Inc. MicroStation.
<http://www.bentley.com>, 2002.
- [6] D. J. Bouvier. Getting Started with the Java3D API. Technical report, Sun microsystems, 2000. Available from <http://java.sun.com>.
- [7] byHeart Consultants Limited. Helios32.
<http://www.helios32.com>, 2002.
- [8] M. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. A Progressive Refinement Approach to Fast Radiosity Image Generation. In *Computer Graphics (ACM SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, August 1988.
- [9] M. Cohen and D. P. Greenberg. The Hemi-Cube: A Radiosity Solution for Complex Environments. In *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, volume 19, pages 31–40, August 1985.
- [10] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, MA, 1993.
- [11] K. Daubert, H. Schirmacher, F. cois Sillion, and G. Drettakis. Hierarchical lighting simulation for outdoor scenes. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 229–238, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.
- [12] Exluna, Inc. Blue Moon Rendering Tools.
<http://www.bmrt.org>, 2002.
- [13] A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1995.

- [14] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modelling the Interaction of Light Between Diffuse Surfaces. In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 212–222, July 1984.
- [15] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.
- [16] P. Heckbert. Adaptive Radiosity Textures for Bidirectional Ray Tracing. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24, pages 145–154, August 1990.
- [17] H. C. Hottel and A. F. Sarofin. *Radiative Transfer*. McGraw Hill, New York, NY, 1967.
- [18] B. Hudec. *Programovací techniky*. Czech Technical University Publishing House, Prague, 1993. In Czech.
- [19] Integra Inc., Visual Computing. Specter.
<http://www.integra.jp/eng/>, 2002.
- [20] ISO. 14772-1. *Information Technology - Virtual Reality Modelling Language (VRML)*. International standard ISO/IEC, 1997.
- [21] J. Jelen and J. Lego. *Fyzika II*. Czech Technical University Publishing House, Prague, 1997. In Czech.
- [22] J. T. Kajiya. The Rendering Equation. In *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.
- [23] A. Keller. A quasi-monte carlo algorithm for the global illumination problem in the radiosity setting. In H. Niederreiter and P. Shiue, editors, *Lecture Notes in Statistics (Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing)*, volume 106, pages 239–251. Springer-Verlag, New York, NY, 1995.
- [24] A. Keller. The Fast Calculation of Form Factors Using Low Discrepancy Sequences. In *Proceedings of the Spring Conference on Computer Graphics (SCCG '96)*, pages 195–204, Bratislava, Slovakia, June 1996. Comenius University Press. Available from <http://www.dcs.fmph.uniba.sk/sccg/proceedings/1996.index.htm>.
- [25] A. Keller. *Quasi-Monte Carlo Methods for Photorealistic Image Synthesis*. PhD thesis, University of Kaiserslautern, June 1997. ISBN 3-8265-3330-5.
- [26] A. J. F. Kok. *Ray Tracing and Radiosity Algorithms for Photorealistic Image Synthesis*. Ph.D. thesis, Delft, The Netherlands, 1994.
- [27] J. Křivánek. Moderní algoritmy pro syntézu obrazu. Master's thesis, Czech Technical University in Prague, 2001.
- [28] E. Languenou, K. Bouatouch, and P. Tellier. An Adaptive Discretization Method for Radiosity. In *Computer Graphics Forum (Eurographics '92)*, volume 11, pages C205–C216, Cambridge, UK, September 1992.

- [29] G. W. Larson and R. Shakespeare. *Rendering with Radiance: The Art and Science of Lighting Visualization*. Morgan Kaufmann, San Francisco, CA, 1998. ISBN 1-55860-499-5.
- [30] LightWork Design. Lightworks.
<http://www.lightworkdesign.com>, 2002.
- [31] N. Max. Smooth appearance for polygonal surfaces. *The Visual Computer*, 5(3):160–173, 1989.
- [32] Mental Images. Mental Ray.
<http://www.mentalimages.com>, 2002.
- [33] Micro-PACE, Inc. trueSpace.
<http://www.micropace.com>, 2002.
- [34] S. Müller, W. Kresse, N. Gatenby, and F. Schöffel. A radiosity approach for the simulation of daylight. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 137–146, New York, NY, 1995. Springer-Verlag.
- [35] L. Neumann, M. Feda, M. Kopp, and W. Purgathofer. A New Stochastic Radiosity Method for Highly Complex Scenes. In *Fifth Eurographics Workshop on Rendering*, pages 195–206, Darmstadt, Germany, June 1994.
- [36] Pixar Animation Studios. Pixar Renderman.
<http://www.pixar.com>, 2002.
- [37] M. C. Reichert. A Two-Pass Radiosity Method Driven by Lights and Viewer Position. M.Sc. thesis, Ithaca, NY, January 1992.
- [38] K. Rektorys. *Přehled užití matematiky*. SNTL – Nakladatelství technické literatury, 2 edition, 1968. In Czech.
- [39] G. Rougeron. Raysmith.
<http://wuarchive.wustl.edu/graphics/graphics/radiosity/raysmith/>, 2002.
- [40] M. S. Shephard. Approaches to the automatic generation and control of finite element meshes. In *Applied Mechanics Review* 41:4, pages 169–185, 1988.
- [41] P. Shirley. A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes. In *Proceedings of Graphics Interface '90*, pages 205–212, San Francisco, CA, May 1990. Morgan Kaufmann.
- [42] B. Smits and H. W. Jensen. Global illumination test scenes. Technical Report UUCS-00-013, University of Utah, 2000. Available from <http://www.cs.utah.edu/bes/papers/scenes>.
- [43] Softimage, Co. SoftImage.
<http://www.softimage.com>, 2002.
- [44] Strata Software. Strata 3Dpro.
<http://www.strata.com>, 2002.

- [45] Sun microsystems. Java programming language.
<http://java.sun.com>, 2001.
- [46] L. Szirmay-Kalos. Monte-carlo methods in global illumination. Institute of Computer Graphics, Vienna University of Technology, 1999.
- [47] Tech Soft. HOOPS.
<http://www.hoops3d.com>, 2002.
- [48] R. F. Tobler, A. Wilkie, M. Feda, and W. Purgathofer. A hierarchical subdivision algorithm for stochastic radiosity methods. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 193–204, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4.
- [49] C. Vedel. Improved Storage and Reconstruction of Light Intensities on Surfaces. In *Third Eurographics Workshop on Rendering*, pages 113–121, Bristol, UK, May 1992.
- [50] J. Žára. *Jazyky pro popis virtuální reality*. Czech Technical University Publishing House, Prague, 2000. In Czech.
- [51] J. Žára, B. Beneš, and P. Felkel. *Moderní počítačová grafika*. Computer press s.r.o, Brno, 1 edition, 1998. In Czech).
- [52] J. R. Wallace, K. A. Elmquist, and E. A. Haines. A Ray Tracing Algorithm for Progressive Radiosity. In *Computer Graphics (ACM SIGGRAPH '89 Proceedings)*, volume 23, pages 315–324, July 1989.
- [53] T. Warnock. Computational investigations of low-discrepancy point sets. In H. Niederreiter and P. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 354–361. Springer-Verlag, New York, NY, 1995.
- [54] A. Willmott and P. Heckbert. An empirical comparison of radiosity algorithms. Technical Report CMU-CS-97-115, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, April 1997. Available from <http://www.cs.cmu.edu/~radiosity/emprad-tr.html>.

Příloha A

Slovníček

Většina existující literatury v oboru realistické syntézy obrazu je psána anglicky. Následující slovníček pojmů by měl pomoci čtenáři při orientaci v této literatuře i samotné diplomové práci. Pojmy, které ve slovníčku nemají český ekvivalent nejsou překládány a v rámci diplomové práce jsou používány jejich původní znění. V závorkách jsou uvedeny jiné, někdy používané názvy.

albedo	bělost (poměr odraženého a dopadajícího světla)
ambient light	okolní světlo
API	aplikační programové rozhraní
aspect ratio	
behavior routine	
capability	
central limit theorem of probability	centrální limitní věta
	$\lim_{n \rightarrow \infty} P[\frac{X-\mu}{\sigma/\sqrt{n}} \leq t] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\frac{u^2}{2}} du$
Chebyshev's inequality	Čebyševova nerovnost
	$P[X - EX \geq \varepsilon] \leq \frac{DX}{\varepsilon^2}$
CIE	Commission Internationale de l'Éclairage [1]
clustering	shlukování
crease angle	
cumulative probability distribution (cdf)	distribuční funkce
diffuse	difúzní
direct illumination	přímé osvětlení
discontinuity meshing	
estimate, estimator	odhad
expectation	střední hodnota
	diskrétní: $EX = \sum_i x_i p_i$
	spojitá: $EX = \int_{-\infty}^{\infty} x f(x) dx$
false color image	obraz ve falešných (neskutečných) barvách
glossy	lesklý
illuminance	osvětlení

importance sampling	vzorkování podle důležitosti
indirect illumination	nepřímé osvětlení
LOD	level of detail
luminance	měrná svítivost
luminosity	jas
luminous intensity	svítivost
luminous power	světelný tok
mesh	síť
meshing	(konstrukce sítě)
patch	ploška (polygon, záplata)
penumbra	polostín
probability density function (pdf)	hustota pravděpodobnosti
radiance	zář
radiant exitance	intenzita vyzařování
radiant flux, radiant power	zářivý tok
radiant intensity	zářivost
radiosity	radiozita (intenzita vyzařování)
random variable	náhodná veličina
sample	vzorek
sampling	vzorkování
specular	zrcadlový
standard deviation	směrodatná odchylka
stratified sampling	vzorkování po částech
unbiased	nestranný
variance	rozptyl
	$DX = \sigma^2 = E(X - EX)^2 = EX^2 - (EX)^2$
visibility function	funkce viditelnosti

Příloha B

Formát souborů

Systém MCrad načítá dva typy vstupních souborů – soubor se vstupní scénou a soubor s definicí paralelních zdrojů. Vypočtenou scénu umožňuje systém uložit ve formě výstupního souboru. V této příloze jsou popsány uvedené tři typy souborů.

Soubor se vstupní scénou je VRML soubor, který má některé specifické vlastnosti pro systém MCrad. Všechny objekty v souboru musí být rozděleny na trojúhelníkové plošky, parametry uzlu **Material** mají následující interpretaci:

diffuseColor	absolutní hodnota tohoto parametru udává velikost součinitele difúzní odrazivosti plošky ρ , v rozsahu $\langle 0, 1 \rangle$
specularColor	absolutní hodnota tohoto parametru udává velikost součinitele zrcadlové odrazivosti plošky ρ_S , v rozsahu $\langle 0, 1 \rangle$
emissiveColor	absolutní hodnota tohoto parametru udává sumu vlastní emise $\Phi = \sum_i \Phi_i$ všech plošek, které mají přiřazen daný uzel Material ; tato celková hodnota je váhově rozdělena podle velikostí povrchů mezi jednotlivé plošky

Jsou-li v souboru se vstupní scénou obsaženy světelné zdroje (tzn. plošky s nenulovou hodnotou **emissiveColor**), jsou tyto uvažovány jako difúzní zdroje, pro specifikaci paralelních světelných zdrojů je určen druhý vstupní soubor.

Soubor s definicí paralelních zdrojů je druhým (nepovinným) vstupním souborem, který obsahuje údaje o emisi a geometrii zdrojů rovnoběžného světla. Informace načtené z tohoto souboru se využívají pouze při nulté, tzv. světelné iteraci výpočtu a nejsou zahrnuty do vyšetřování průsečíků a do dalších výpočtů. Podobně jako v případě souboru se vstupní scénou musí být objekty rozděleny na trojúhelníkové plošky. Z parametrů uzlu **Material** se uvažuje pouze:

emissiveColor jehož interpretace je stejná jako u souboru se vstupní scénou

V případě zdrojů paralelního světla nejsou podstatné, a proto se neuvažují, hodnoty součinitelů difúzního a zrcadlového odrazu.

Program MCrad načítá vstupní VRML soubory jak v klasickém textovém formátu, tak i ve formátu komprimovaném.

Výstupní soubor je ukládán taktéž ve formátu VRML. Celá vypočtená scéna ve formě trojúhelníkových plošek je reprezentována pomocí jediného uzlu `IndexedFaceSet`. V závislosti na aktuálním režimu zobrazení je do výstupního souboru uložena scéna s konstantní aproximací nebo s lineární interpolací energií plošek, a to buď monochromaticky, nebo s barevným zvýrazněním světelného zatížení (tzv. *false color image*).

Příloha C

Uživatelská příručka

C.1 Instalace programu

Vzhledem k tomu, že systém MCrad je naprogramován v jazyce Java, je k jeho spuštění potřeba tzv. Java Virtual Machine (JVM) verze 2. JVM 2 je součástí balíku Java Development Kit (JDK) [45], resp. Java Runtime Environment (JRE) [45], který je třeba nainstalovat jako první. Dalším krokem je instalace API Java 3D [45].

Pokud byly tyto instalace úspěšné, včetně nastavení správných parametrů a cest pro interpret jazyka Java [45], je možné spustit program MCrad v prostředí MS Windows pomocí dávky `mcrad.bat`, resp. pomocí obdobného skriptu pod jiným operačním systémem.

C.2 Parametry příkazového řádku

Ke spouštění systému MCrad slouží dávka `mcrad.bat`, která má dva nepovinné parametry:

`mcrad [vstupní scéna] [paralelní zdroje]`

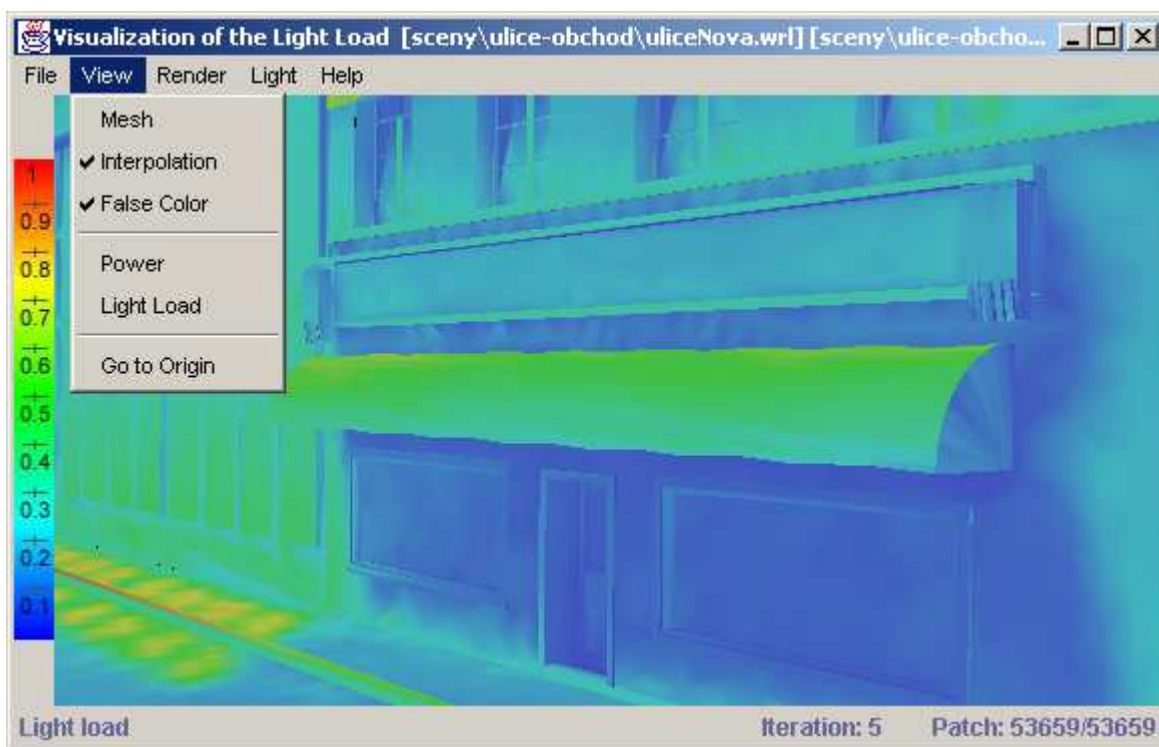
vstupní scéna	tento parametr udává jméno VRML souboru, který obsahuje vstupní scénu, s případnou cestou k němu; formát souboru je popsán v příloze B
paralelní zdroje	tento parametr udává jméno nepovinného VRML souboru, který obsahuje specifikaci paralelních zdrojů světla, s případnou cestou k němu; formát souboru je popsán v příloze B

Přípony souborů `.wrl` jsou nepovinné a v případě, že nejsou uvedeny se automaticky doplní.

C.3 Ovládání programu

Hlavní okno programu je zobrazeno na obr. C.1. V horní části okna je umístěno roletové menu, uprostřed se nachází pohled na scénu, ve spodní části okna je zobrazen stavový řádek a v levé části okna je umístěna barevná stupnice relativních hodnot.

Ovládání programu je možné za pomoci roletového menu, klávesnice a myši. Strukturu a význam jednotlivých prvků roletového menu stručně popisuje tabulka C.3.



Obrázek C.1: Uživatelské rozhraní systému MCrad

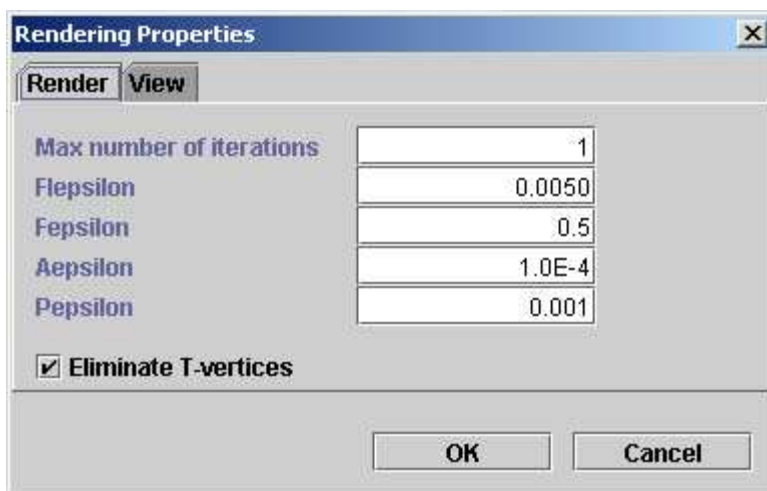
Navigace v 3D scéně, tzn. změna polohy a parametrů kamery, je možná pomocí klávesnice a myši. Navigaci pomocí klávesnice popisuje tabulka C.1, navigaci pomocí myši popisuje tabulka C.2.

C.3.1 Výpočet

Spuštění výpočtu je možné pouze z režimu zobrazení **Preview** (viz níže) pomocí položky menu **Render**. Iterační výpočet probíhá podle algoritmu popsaného v předchozích kapitolách. Během výpočtu není aktualizován pohled na scénu, informace o průběhu výpočtu udává stavový řádek. V pravé části stavového řádku se zobrazuje pořadové číslo aktuálně počítané plošky scény a celkový počet plošek, které tvoří listy hierarchické výpočetní struktury. Ve střední části stavového řádku je zobrazeno pořadové číslo právě probíhající iterace. Po dokončení výpočtu přejde systém do režimu **Power** a zobrazí vypočtenou scénu pomocí konstantní aproximace energie na ploškách.

Výpočet je možné přerušit pomocí položky menu **Stop**. Systém v případě zastavení výpočtu při probíhající iteraci s pořadovým číslem i přejde do režimu **Power**, kde zobrazí výslednou scénu získanou po dokončení iterace $i - 1$. Po zastavení výpočtu není již možné v dalším výpočtu pokračovat.

Parametry výpočtu je možné nastavit v dialogovém okně, které se zobrazí po aktivaci položky **Properties** v menu programu, viz obr. C.2.



Obrázek C.2: Dialog pro nastavení parametrů výpočtu

C.3.2 Zobrazení

Systém MCrad se v daném okamžiku může nacházet v jednom z následujících tří režimů zobrazení:

Preview	zobrazení náhledu na scénu pomocí Gouraudova stínování; v tomto režimu se nachází systém po inicializaci a po každém načtení nové scény; spuštění výpočtu je možné provést pouze z tohoto režimu
Power	zobrazení hodnot energií P_i na ploškách tak, jak by se jevily skutečnému pozorovateli
Light load	zobrazení dopadlé světelné energie P_g na plošky scény

Aktuálně zvolený režim zobrazení je indikován v levé části stavového řádku programu. Ve všech třech popsanych režimech je možné zapnout zobrazení sítě (**Mesh**). Režimy **Power** a **Light load** navíc umožňují zobrazit hodnoty energií na ploškách pomocí interpolace prvního řádu **Interpolation** a pomocí barevné škály (**False color**).

Transformace vypočtených hodnot na interval $\langle 0, 1 \rangle$ se provádí postupem popsáným v odstavci 3.3. Hodnotu intenzity ozáření, která odpovídá horní mezi intervalu, je možné nastavit v dialogu světelné analýzy, viz obr. C.3 (položka menu **Lighting Analysis**). Uvedený dialog zobrazuje současně hodnoty nejnižší, průměrné a nejvyšší intenzity ozáření ve scéně a jejich poměry.

C.3.3 Uložení výsledku

Vypočítanou scénu je možné uložit do výstupního souboru ve formátu VRML (viz příloha B) pomocí položky menu **Save**. Scéna je uložena takovým způsobem, jakým je aktuálně zobrazována. V řadě případů je velikost výstupního souboru poměrně značná (stovky kB, ale i desítky MB). Velikost souboru závisí na režimu zobrazení – nejmenší velikost souboru lze dosáhnout při uložení v režimu zobrazení interpolace, kdy jsou společné vrcholy plošek ve výstupním souboru indexovány a mají stejnou barvu.



Obrázek C.3: Světelná analýza

Klávesa	Význam
Šipka doleva ←	Rotace kamery doleva
Šipka doprava →	Rotace kamery doprava
Page Down	Rotace kamery dolů
Page Up	Rotace kamery nahoru
Šipka dolů ↓	Posun kamery dozadu
Šipka nahoru ↑	Posun kamery dopředu
Alt + Šipka doleva ←	Posun kamery doleva
Alt + Šipka doprava →	Posun kamery doprava
Alt + Page Down	Posun kamery dolů
Alt + Page Up	Posun kamery nahoru
Rovnítko =	Návrat kamery do počátku

Tabulka C.1: Ovládání programu pomocí klávesnice

Akce	Význam
Levé tlačítko + pohyb	Rotace kamery
Prostřední tlačítko + pohyb	Posun kamery dopředu/dozadu
Alt + Levé tlačítko + pohyb	Posun kamery dopředu/dozadu
Pravé tlačítko + pohyb	Posun kamery vlevo/vpravo, nahoru/dolů

Tabulka C.2: Ovládání programu pomocí myši

Položka	Význam
File	toto submenu obsahuje položky pro načítání a ukládání souborů
Open scene Open lighting Save	otevření souboru se vstupní scénou otevření souboru s definicí paralelních zdrojů uložení vypočtené scény do výstupního souboru
View	toto submenu obsahuje přepínače, kterými lze změnit parametry zobrazení
Mesh Interpolation False Color Power Light Load Go to Origin	zapnutí/vypnutí zobrazení sítě, viz obr. 6.4 zobrazení pomocí interpolace hodnot energií, resp. pomocí konstantní aproximace, viz obr. 3.7 zobrazení barevné vizualizace světelného zatížení scény, resp. monochromatické zobrazení, viz obr. 6.2 zobrazení skutečných hodnot energií P_i na ploškách tak, jak by se jevily pozorovateli (zobrazovací režim Power), viz obr. 6.3a zobrazení akumulované dopadlé světelné energie na plošky scény (zobrazovací režim Light Load), viz obr. 6.3b přesun kamery do počátku souřadného systému
Render	toto submenu obsahuje položky pro ovládání radiozitivního výpočtu
Render Stop Properties	spuštění výpočtu právě zobrazené scény (pouze v režimu Preview) zastavení výpočtu, po zastavení není možné dále pokračovat ve výpočtu zobrazí dialog pro nastavení parametrů výpočtu, viz obr. C.2, kde je možné nastavit příslušné parametry výpočtu
Light	toto submenu obsahuje položky pro zobrazení informací o ploškách scény
Lighting Analysis Trace Scene	zobrazí dialog se souhrnnými informacemi o ozáření (angl. irradiance) scény, viz obr. C.3; v tomto dialogu je možné změnit hodnotu horní meze intervalu zobrazovaných hodnot ozáření vypíše v textové formě parametry všech plošek scény; pozn.: u rozsáhlých scén může tento výpis trvat velmi dlouho
Help	toto submenu obsahuje informace o systému
About	zobrazí informační dialog s označením verze, jménem autora a datem vytvoření programu

Tabulka C.3: Položky roletového menu programu

Příloha D

Obsah přiloženého CD-ROM

Tato příloha obsahuje základní popis přiloženého CD-ROM. Podrobnější informace, jako je seznam souborů nebo pokyny pro spuštění programu, jsou uvedeny v souborech `readme.txt` a `index.html` v kořenovém adresáři přiloženého CD-ROM. To obsahuje:

- Abstrakt diplomové práce v češtině a angličtině ve formátu HTML.
- Kompletní text diplomové práce.
- Program *MCrad* ve formě tzv. univerzálního bytekódu.
- Zdrojové kódy všech tříd systému *MCrad*.
- Programátorskou dokumentaci systému.
- Instalační soubory prostředí Java – JRE a Java3D.
- Ukázkové scény.