

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

**Simulace osvětlení okolí vozidla pomocí světlometů v
reálném čase**

**Simulation of Automotive Headlamps Light Output in
Real Time**

2012/2013	Bc. Jan Křístek
-----------	-----------------

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jan Křístek**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Simulace osvětlení okolí vozidla pomocí světlometů v reálném čase**
Simulation of Automotive Headlamps Light Output in Real Time

Zásady pro vypracování:

Cílem práce je vytvořit simulaci osvětlení okolí vozu pomocí světlometů v reálném čase schopnou spolupracovat s jízdním simulátorem v systému VRUT v prostředí CAVE. Směrová charakteristika světel bude dána souborem ve formátu IES. Příklady světel, model vozu a krajiny jsou také součástí zadání. Pro simulaci nočního osvětlení je nutné upravit vlastnosti materiálů. Pro ověření funkčnosti budou poskytnuty syntetické testy (např. rovina osvětlená jedním světlem s danými vzájemnými pozicemi a rozměry).

1. Seznamte se s architekturou systému VRUT.
2. Navrhněte rozšíření pro umožnění vkládání světelných map (nový typ světla ve scéně).
3. Vyřešte umístění světelných map ve scéně.
4. Umožněte změnu osvětlení okolních objektů (např. změna teploty barev, barevné škálování, definice rozsahu jasu).
5. Na dodaných datech demonstруйте funkčnost řešení.

Seznam doporučené odborné literatury:

- [1] WOLF, David. OpenGL 4.0 Shading Language Cookbook. 2011. ISBN 978-1849514767.
[2] SHREINER, Dave, SELLERS, Graham, KESSENICH, John, LICEA-KANE, Bill. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 4.1. 8th Edition. 2012. ISBN 978-0321773036.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


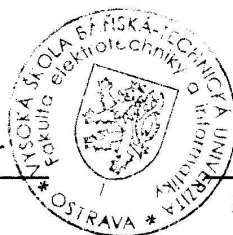
Vedoucí diplomové práce: **Ing. Tomáš Fabián**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 7.5.2013

Jan Křížek
podpis studenta

Poděkování

Rád bych poděkoval Ing. Tomáši Fabiánovi a Mgr. Antonínu Miškovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce.

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“

Dne: 7.5.2013

Jan Křížek
.....
podpis studenta

Abstract

Tato práce popisuje zobrazení osvětlení plochy při použití určitého světelného profilu ve formátu IES umístěného ve scéně programu VRUT, který vyvíjí Škoda Auto a.s.. S touto světelnou mapou lze rotovat ve všech možných osách a měnit polohu za pomoci transformačních matic. Následně bylo zabezpečeno, aby šla světlu upravovat barva pomocí barevné teploty černého tělesa k určitému barevnému profilu. Dále v práci popisují více variant možných implementací zobrazení, jak v metodě raytracing, tak při použití API OpenGL. V implementacích využívajících API OpenGL bylo využito vícero variant vstupních dat (objemová a dvourozměrná textura) pro shader.

Klíčová slova

API OpenGL, GLSL, IES, 2D textura, 3D textura, raytracing, Eulerova transformace, barevná teplota, Planckův vyzařovací zákon, barevný prostor XYZ, barevný prostor RGB, barevný profil, Škoda Auto a.s., VRUT

Abstract

The aim of this work is to describe how to show illuminance of surface by using a light profile in IES format located in the scene of the program VRUT, which develops Škoda Auto a.s.. This light-map can be rotated in all possible axes and vary the position using the transformation matrix. It was ensured, that we can modify color of light using the color temperature of a blackbody to a specific color profile. The work describes the multiple variations of possible implementations, with raytracing method and also by using of API OpenGL. In implementations with OpenGL API was used several variations of the input data (volume and a two-dimensional texture) for the shader.

Key word

API OpenGL, GLSL, IES, 2D texture, 3D texture, raytracing, Euler transform, color temperature, Planck's law, XYZ color system, RGB color system, color profile, Škoda Auto a.s., VRUT

Obsah

1 Úvod.....	1
2 Teoretická část.....	2
2.1 Elektromagnetické záření.....	2
2.1.1 Rozdělení elektromagnetického spektra.....	2
2.2 Fotometrické veličiny.....	3
2.2.1 Světelný tok.....	4
2.2.2 Svítivost.....	5
2.2.3 Osvětlení.....	7
2.3 Radiometrické veličiny.....	10
2.3.1 Zářivý tok.....	10
2.3.2 Zářivost.....	10
2.3.3 Intenzita ozáření.....	11
3 Implementace světelných map.....	12
3.1 Formát světelného profilu IES.....	12
3.1.1 Parametry vybraného světelného profilu IES.....	12
3.1.2 Hledání dat ve vybraném světelném profilu za pomoci bilineární interpolace.....	13
3.2 Raytracing.....	15
3.2.1 Popis získávání konkrétních hodnot svítivosti pro zobrazování osvětlení plochy.....	16
3.3 Objemová textura.....	18
3.3.1 Předgenerování dat části osvětlení v určitých úhlových odchylkách od základního směru světla a vzdálenosti od zdroje světla, před samotným uložením do objemové textury a renderováním.....	19
3.3.2 Hledání koordinátů v objemové textuře pro zjištění předgenerovaných hodnot části osvětlení pro výpočet celkového osvětlení plochy.....	25
3.4 Dvourozměrná textura.....	27
3.4.1 Generování dat svítivosti v určitých úhlových odchylkách od základního směru světla a následné ukládání do dvourozměrné textury.....	28
3.4.2 Výpočet osvětlení z dat svítivosti uložené v dvourozměrné textuře.....	29
3.4.2.1 Varianta inspirovaná raytracingem.....	29
3.4.2.2 Varianta inspirovaná objemovou texturou.....	30
3.5 Transformace světla ve scéně.....	32
3.5.1 Eulerova transformace pro rotaci světlometů.....	33
3.6 Převod barevné teploty na RGB barvu.....	35
3.6.1 Převod barevné teploty do barevného prostoru XYZ.....	35
3.6.2 Převod z barevného prostoru XYZ do barevného prostoru RGB.....	37
4 Vyhodnocení implementace.....	40
4.1 Rychlost vykreslování při určitých variantách.....	40

5 Závěr.....	43
6 Seznam použité literatury	
7 Seznam příloh	

Úvod

1 Úvod

Cílem diplomové práce je zobrazit osvětlení dané plochy při konkrétním světelném profilu, poloze a směru světelného zdroje. Šíření světla prostorem jsme schopni nasimulovat podle fyzikálních zákonů různými metodami. Jako příklad náročnější metody, pro tuto simulaci můžeme uvést raytracer. V této práci je pro rychlejší zobrazení v reálném čase využíváno API OpenGL za pomoci shaderů.

Téma diplomové práce bylo zadáno automobilkou ŠKODA Auto a.s., která potřebovala zobrazovat a manipulovat se světlometry ve scéně v reálném čase ve svém grafickém systému VRUT. Hlavní myšlenka fyzikálního šíření světla je přejata z metody raytracing, kde světelné paprsky vycházející z pozice světla a mohou se šířit jen pod určitým úhlem od směru základního směru světla a se vzdáleností od zdroje dochází k útlumu.

Podstatnou částí výsledné implementace je předgenerování části osvětlení v prostoru, uložení do prostorového úložiště objemové textury. Následně je použito těchto vytvořených hodnot, za běhu programu, pro rychlejší renderování. V dnešní době jsou grafické karty vybavené rychlou a velkou operační pamětí, která nám dovolí předgenerování velkého množství dat. Tímto způsobem můžeme ušetřit mnoho výkonu při samotném renderování výsledné scény. Druhou možnou variantou bylo uložení svítivosti[cd] ze světelného profilu do dvourozměrné textury, která na rozdíl od objemové textury zabírala jen zlomek její velikosti. Do zobrazovací techniky raytracing byl následně přidána možnost použít osvětlovací IES profil v už vytvořeném v grafickém systému VRUT. Následně jsem vytvořil novou renderovací metodu raytraceru za použití API OpenCL pro lepší paralelismus, který nám umožní přiblížení se k renderování v reálném čase. Pro manipulování se světly ve scéně byla použita transformační matice, se kterou šlo snadně manipulovat pomocí Eulerovy transformace. Taktéž byla použita metoda, která dokáže převést barevnou teplotu černého tělesa do RGB barevného prostoru.

Teoretická část

2 Teoretická část

Tato kapitola se bude věnovat teoretické části samotného problému získávání a zobrazování světelného profilu pomocí fyzikálních veličin a vztahů k nim.

2.1 Elektromagnetické záření

O světle můžeme říct, že jde o příčné elektromagnetické vlnění, pro své šíření nepotřebuje žádné látkové prostředí (prostředí, které umožňuje šíření energie) a tudíž se dokáže šířit například vakuem. Velikost rychlosti světla ve vakuu je $c = 299792458 \text{ m s}^{-1} = 3 \cdot 10^8 \text{ m s}^{-1}$. Jedná se tedy o maximální rychlost, kterou se může fyzikální objekt pohybovat.[1]

Podobně jako jiné druhy vlnění, charakterizujeme i světlo vlnovou délkou λ podle vztahu (1), kde f je frekvence tohoto světla.

$$\lambda = \frac{c}{f} \quad (1)$$

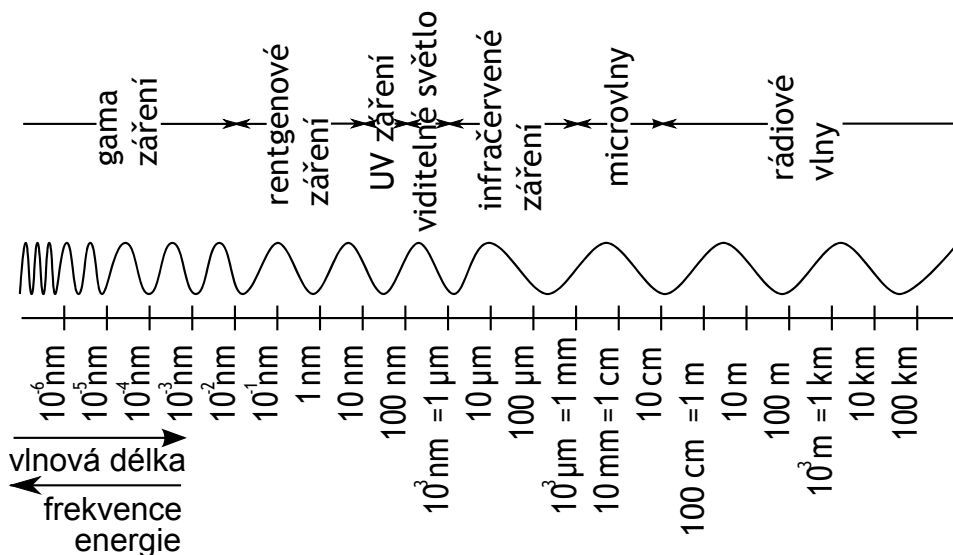
Výpočet energie fotonu E je vidět na vztahu (2), kde h je Planckova konstanta, která udává jak závislost intenzity záření na absolutně černém tělese při určité frekvenci f , tak se využívá i u vztahu mezi energií E (2) a frekvencí f fotonu. Tato konstanta nabývá hodnoty $h = 6,6252 \cdot 10^{-34} \text{ J.s}$.

$$E = hf \quad (2)$$

2.1.1 Rozdělení elektromagnetického spektra

Hlavními kritérii, podle kterých dělíme elektromagnetické záření je jejich vlnovou délka λ popřípadě frekvence f , na ilustraci 1 vidíme jednotlivé druhy a rozsahy známých a používaných druhů elektromagnetického záření. Vlnové délky elektromagnetického spektra nejsou nikterak omezeny a hranice mezi jednotlivými druhy elektromagnetického záření nejsou ostré.

Teoretická část



Ilustrace 1: Zařazení elektromagnetického záření do používaného elektromagnetického spektra

2.2 Fotometrické veličiny

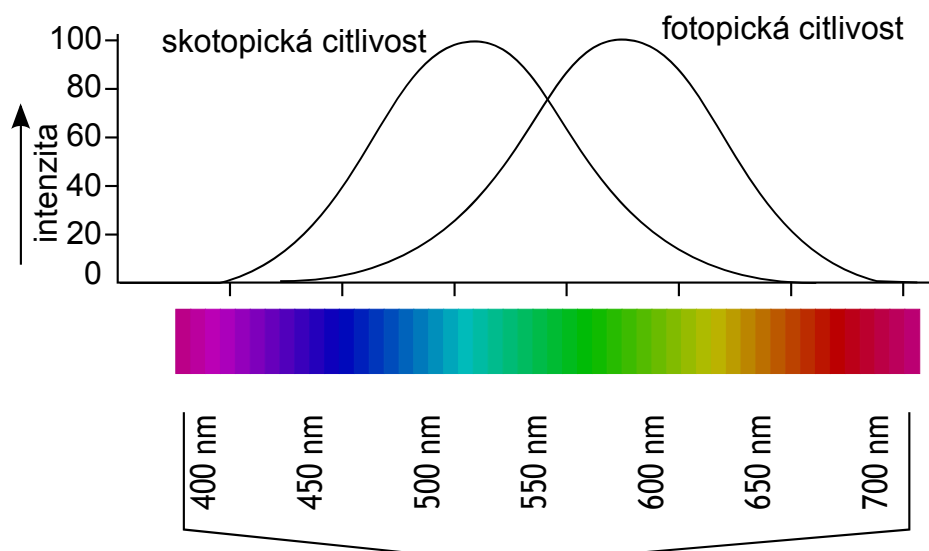
Protože nás zajímá jen ta část elektromagnetického spektra, na kterou je lidské oko citlivé tedy 400–700nm, které nazýváme viditelné světlo, toto elektromagnetické záření jde vidět z ilustrace 1. Z tohoto důvodu se tedy budeme zajímat hlavně o fotometrické veličiny, které jsou na tuto část elektromagnetického spektra zaměřeny. Radiometrickým veličinám, které se věnují celému elektromagnetickému spektru se budeme věnovat jen okrajově. Tyto fotometrické veličiny jsou chronologicky starší než radiometrické.[1]

Pro korektní pochopení daného tématu si musíme definovat dva různé typy vidění z prostých důvodů, citlivost lidského oka je při denním světle rozdílná než při nočním osvětlení a tudíž si musíme definovat tyto dva typy vidění: “fotopické a skotopické“, jejich citlivost na určité vlnové délky jde vidět z ilustrace 2, kde existuje mezi těmito typy vidění určitý fázový posuv, který musíme brát v potaz.

- Fotopická citlivost – definice pro vidění za denního osvětlení, kde světlo dopadá na čípky v sítnici lidského oka

Teoretická část

- Skotopická citlivost – definice pro noční osvětlení, kde tyčinky na sítnici lidského oka dokáží reagovat na změny světla



Ilustrace 2: Křivky popisující citlivost oka při denním světle fotopická citlivost a při nočním osvětlení skotopická ve spektru viditelného světla

Čípky v lidském oku dokáží lépe reagovat na barevnost světla, a proto se nejlépe uplatní hlavně při denním osvětlení. Tyčinky oproti čípkům jsou více citlivé na změnu intenzity dopadajícího světla, které dokáže lidské oko lépe využít při nočním osvětlení.

- Světelný tok - vyjadřuje šíření světla prostorem
- Svítivost - vyjadřuje vlastnost světla
- Osvětlení - určuje účinky světla dopadajícího na povrch tělesa

2.2.1 Světelný tok

Světelný tok $\Phi [lm]$ (lumeny) udává množství světelné energie světelného záření, které dokáže projít za jednotku času prostorem, kterým se světlo šíří, které vyvolává vjem na detektoru, tedy lidském oku.

Zářivý tok udávaný ve watech je spjat se světelným tokem, který bere do úvahy citlivost lidského oka na jednotlivé barevné spektrum. Pro příklad berme v úvahu žárovku se

Teoretická část

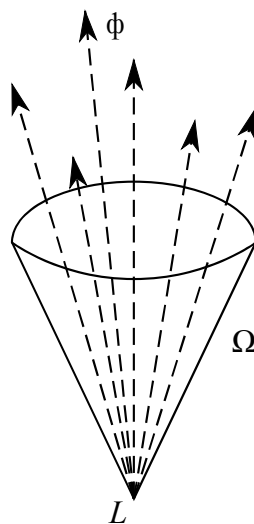
světelným výkonem 100W, která těchto 100W vyzařuje do prostoru ve formě elektromagnetického záření, ale ve formě světelného záření vyzařuje méně. Pro příklad kdyby se jednalo o infračervené záření, které je lidskému zraku neviditelné, tak by světelný tok byl roven 0 lumen.

2.2.2 Svítivost

Svítivost je považována za základní fotometrickou veličinu $I_v[cd]$ (kandela). I když fotometrické veličiny nemají skoro žádný fyzikální význam, z důvodů vyjádření dalších fotometrických veličin musí být jednotka Kandela obsažena v soustavě SI. Jednotka svítivosti 1 kandela odpovídá zhruba svítivosti jedné voskové svíčky. Obyčejná 100W žárovka má svítivost cca $I_v=200[cd]$.

Svítivost světelného zdroje v daném směru můžeme definovat jako podíl světelného toku $\Delta\phi_v$ vyzařovaného zdrojem v tomto směru do malého prostorového úhlu $\Delta\Omega$ dané velikostí popsané vztahem (3) a zobrazena na ilustraci 3.

$$I_v = \frac{\Delta\phi_v}{\Delta\Omega} \quad (3)$$

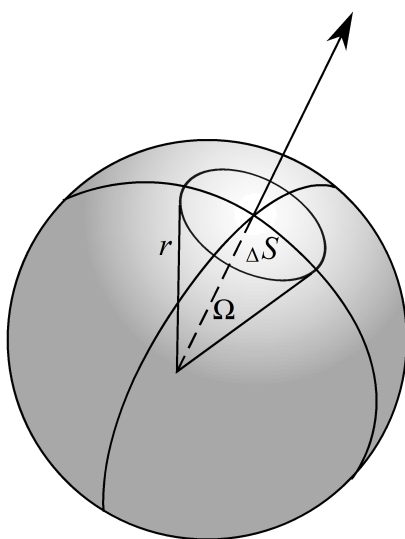


Ilustrace 3: Svítivost závislá na světelném toku ϕ_v v určitém prostorovém úhlu $\Delta\Omega$

Teoretická část

Tato velikost prostorového úhlu Ω je definovaná, jako poměr plochy ΔS , kterou vytíná vrchlík na povrch koule o poloměru r a druhou mocninu poloměru, jak lze vidět na ilustraci 4, a na vztahu(4). Tato hodnota je nezávislá na volbě rádius r .

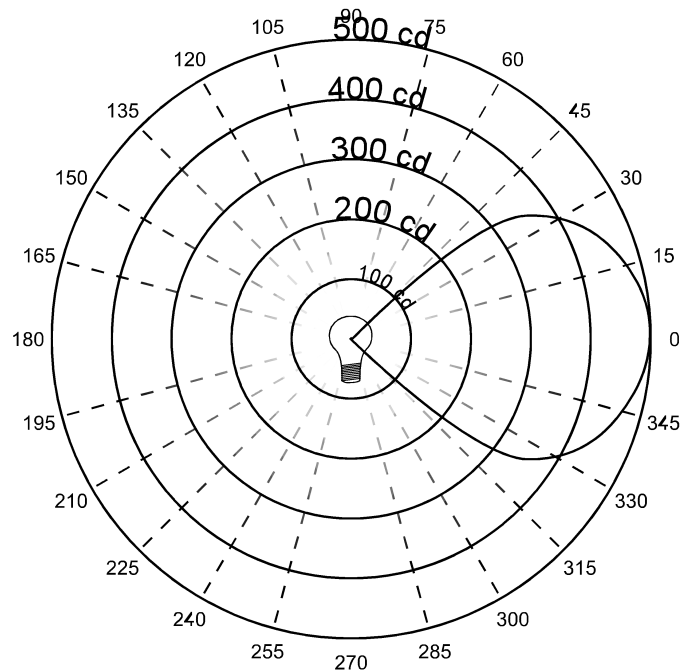
$$\Omega = \frac{\Delta S}{r^2} \quad (4)$$



Ilustrace 4: Znáornění prostorového úhlu

Velikost prostorového úhlu se udává bezrozměrnou jednotkou, která se nazývá steradián [sr]. Prostorový úhel Ω má hodnotu 1 [sr], při výřezu kulové plochy koule o poloměru $r = 1\text{m}$ a obsah této kulové plochy 1m^2 . Svítivost tedy vyjadřuje rozdělení světelných toků do různých směrů, do kterých vyzařuje zdroj světla. Svítivost si taktéž lze představit jako množinu vektorů vyzařované ze zdroje světla, jejichž velikosti odpovídá svítivost v daném směru.

Teoretická část



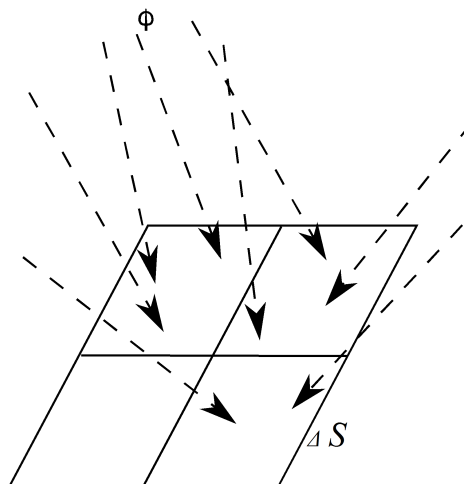
Ilustrace 5: Svítivost v daných úhlech od směru světla, které začíná v nulové úhlové odchylce

2.2.3 Osvětlení

Osvětlení $E_v [lx]$ závisí na části vyzařovaného světelného toku $\Delta\phi_v$, které dopadá pod určitým úhlem na plochu ΔS , které je popsané vztahem(5) a zobrazeno na ilustraci 6.

$$E_v = \frac{\Delta\phi_v}{\Delta S} \quad (5)$$

Teoretická část



Ilustrace 6: Světelný tok Φ dopadající na plochu ΔS , který tuto plochu osvětluje

Z poznatků [1] víme, že s rostoucí vzdáleností od světelného zdroje nám osvětlení plochy prudce klesá. Rovněž závisí i na úhlu pod nímž světelný tok na ozářenou plochu dopadá. Tento světelný útlum se nejvíce projeví, když jsou světelné paprsky rovnoběžné s iluminovanou plochou a naopak dopadají-li kolmo na plochu, pak se tento útlum nikterak neprojeví. Prostorový úhel $\Delta\Omega$, v kterém světlo vyzařuje, je definován vztahem (4).

Tento vztah (4) použijeme pro úpravu definice osvětlení (5) a dostáváme vztah (6).

$$E_v = \frac{\Delta\phi_v}{\Delta\Omega r^2} \quad (6)$$

Aplikujeme-li také definiční vztah svítivosti (3), dostáváme pro kolmě dopadající světlo na plochu v určité vzdálenosti od zdroje světla vztah (7).

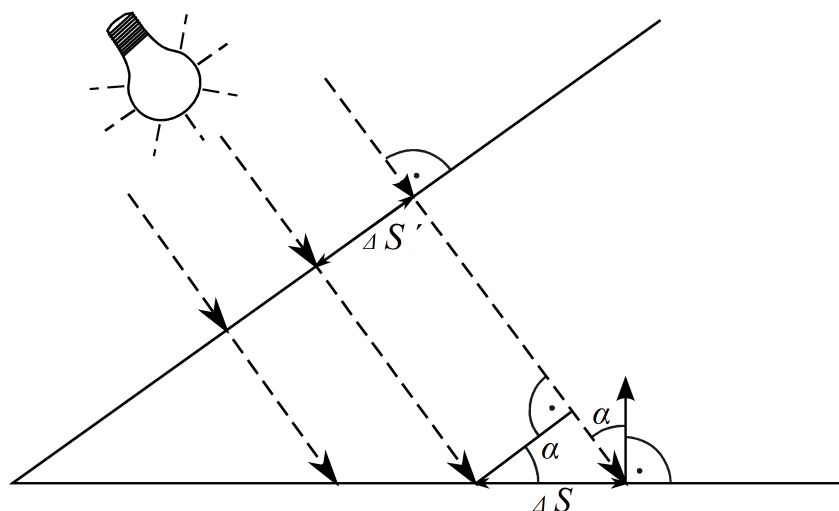
$$E_v = \frac{I_v}{r^2} \quad (7)$$

Chceme-li zahrnout i úhel α , pod kterým dopadají světelné paprsky na osvětlenou plochu ΔS , musíme brát v úvahu situaci na ilustraci 7, Uvažovaná plocha ΔS , na

Teoretická část

kteřou dopadá světlo pod obecným úhlem α , se jeví zdánlivě větší než plocha $\Delta S'$, na kterou světelné paprsky dopadají kolmo. Můžeme tedy napsat vztah(8) pro plochu $\Delta S'$.

$$\Delta S' = \Delta S \cos(\alpha) \quad (8)$$



Ilustrace 7: Obrázek popisující paprsek dopadající kolmo na plochu $\Delta S'$ a na obecnou plochu ΔS pod úhlem α

Uvědomme si, že na plochu $\Delta S'$ dopadají světelné paprsky kolmo a proto můžeme převést obecnou situaci dopadu světla na určitou plochu na případ kolmého dopadu světla na jinou plochu. Zahrnutím vztahu (8) do vztahu(4) a (6) dostáváme pro osvětlení E_v konkrétní plochy ve vzdálenosti r od světelného zdroje, na kterou dopadají světelné paprsky pod určitým úhlem θ , získáme vztah (9).

$$E_v = \frac{I_v \cos(\alpha)}{r^2} \quad (9)$$

I zde, stejně jako v jiných částech optiky, je úhel α úhlem mezi dopadajícím světelným paprskem a normálou, tedy vztyčenou kolmicí v místě dopadu paprsku. Na vztahu (9) je založeno měření osvětlení, které je z pohledu praxe nejdůležitější fotometrickou veličinou. K měření se často používají přístroje založené na přímé transformaci energie

Teoretická část

elektromagnetického záření v elektrickou energii na tak zvaném fotoelektrickém jevu. Přístroj pro měření osvětlení pak nazýváme luxmetr. Ve většině případech je čidlo při měření osvětlení zabudováno do optických přístrojů. Citlivost oka je značná, je schopno rozpoznávat objekty již při osvětlení $3 \cdot 10^{-5} \text{ lx}$.

2.3 Radiometrické veličiny

Protože se tato práce zaměřuje hlavně na fotometrické veličiny, tedy se nám jedná jen o elektromagnetickém záření, které dokáže zachytit lidské oko, proto se radiometrickými veličinami budeme zabývat jen okrajově. Radiometrické veličiny se na rozdíl od fotometrických veličin zabývají přenosem elektromagnetického záření v celém spektru, tedy i v tom, které lidské oko nedokáže vnímat jako je například ultrafialové záření. Proto můžeme o těchto radiometrických veličinách říct, že jsou definované pro všechny druhy elektromagnetického vlnění. Všechny tyto radiometrické veličiny odpovídají veličinám fotometrickým: [1]

- Zářivý tok - odpovídá světelnému toku
- Zářivost – odpovídá svítivosti
- Intenzita ozáření - odpovídá osvětlení

2.3.1 Zářivý tok

Zářivý tok $\Phi_e [W]$ (Watt) reprezentuje množství vyzařené energie elektromagnetického záření zdroje záření, které dokáže projít za jednotku času určitou plochu v prostoru, kterým se elektromagnetické záření šíří.

2.3.2 Zářivost

Zářivost I_e světelného zdroje v daném směru můžeme definovat jako podíl zářivého toku $\Delta_e \Phi$ vyzařenou zdrojem tohoto elektromagnetického záření v tomto směru do malého prostorového úhlu $\Delta \Omega$. Určení tohoto prostorového úhlu bylo definované ve fotometrických veličinách.

$$I_e = \frac{\Delta_e \Phi}{\Delta \Omega} \quad (10)$$

Zářivost tedy vyjadřuje rozdělení zářivých toků do různých směrů, do kterých vyzařuje

Teoretická část

zdroj elektromagnetického záření. Zářivost si také lze představit jako množinu vektorů vyzařované ze zdroje elektromagnetického záření, jejichž velikosti odpovídá zářivost v daném směru.

2.3.3 Intenzita ozáření

Intenzita ozáření $E_e [lx]$ je závislá na zářivém toku $\Delta_e \Phi$, který dopadá na plochu A o obsahu ΔS .

$$E_e = \frac{\Delta \Phi_e}{\Delta S} \quad (11)$$

Implementace světelných map

3 Implementace světelných map

Při samotném počítání osvětlení E_v dané plochy, byl hlavně kladen důraz na samotnou rychlost vykreslování z důvodů zadání. Vývoj proto musel projít mnoha modifikacemi. Pro lepší pochopení samotné problematiky, byla použita metoda raytracing jako první zobrazovací metoda. Samotný výpočet osvětlení dané plochy v této metodě nebyl zase tak náročný, ale protože se metoda raytracing vyznačuje svou vysokou výpočetní složitostí, tak tato varianta získávání osvětlení E_v nemohla být použita. Z důvodů, že tato renderovací metoda je použita i v systému VRUT, musela být i přesto implementována a k tomu být i dostatečně rychlá pro výpočet osvětlení E_v ze světelného profilu. Kvůli zkušenostem s touto renderovací metodou byl celkový výpočet scény urychlen převedením na grafický akcelerátor s použitím API OpenCL, z důvodů možného paralelismu pro rychlejší vykreslování výsledné scény. I přesto, že tato celková revize umožňovala rychlejší výpočty snímku, i tak tato implementace nebyla dostatečně rychlá pro splnění zadání. Proto muselo být využito API OpenGL s použitím shaderu, kde jedna z variant měla nejrychlejší nalezení konkrétní hodnoty svítivosti E_v v prostoru, kde se určitá část osvětlení $E_{v,l}$ v prostoru uložila do objemové textury, která fungovala jako konstantní zdroj informací pro výpočet celkového osvětlení E_v plochy. Hlavní problém této implementace byl v její velké paměťové náročnosti způsobenou použitím objemové textury při zachování vysokých detailů naměřených hodnot části osvětlení $E_{v,l}$, byla tato textura extrémně datově velká. Proto tato metoda byla nahrazena uchováváním svítivosti I_v v dvourozměrné textuře, která se nechala inspirovat oběma variantami, jak metodou raytracingu, tak použitím objemové textury v API OpenGL.

3.1 Formát světelného profilu IES

Protože se zabýváme pouze elektromagnetickým zářením ve viditelném spektru, tak nám stačí pro vizualizaci zaznamenané fotometrické veličiny, které jsou obsažené ve světelném profilu formátu IES. Parametry tohoto formátu můžeme vidět v následujícím seznamu: [2]

3.1.1 Parametry vybraného světelného profilu IES

- TILT=NONE (tento řádek nám určuje hranici od které lze získávat parametry)
- Počet světél. Za základní hodnotu je považována 1

Implementace světelných map

- Hodnota jednoho světla v lumenech. Za základní hodnotu je považována -1
- Násobitel pro hodnoty v kandelách. Za základní hodnotu je považována 1
- Počet úhlů ve vertikální složce
- Počet úhlů v horizontální složce
- Optometrický typ světla: 1=C, 2=B, 3=A. Za základní hodnotu se považuje 1
- Typ jednotek 1 = stopy (americký), 2 = metry (metrický)
- Šířka, délka a výška širícího se světla. Za základní hodnoty se považuje 0, 0, 0
- Balast faktor, balast světelný faktor a vstupní waty. Za základní hodnoty se považuje 1 1 0
- Hodnoty složek vertikálních úhlů
- Hodnoty složek horizontálních úhlů
- Hodnoty (svítivosti) světla v kandelách k úhlům

Tento formát v sobě uchovává hodnoty svítivosti I_v v určitých zaznamenaných úhlech. Tyto hodnoty svítivosti I_v nám postačí pro simulaci elektromagnetické záření ve spektru, které dokáže zachytit lidské oko. Tento záznam si můžeme představit ve tvaru podobném jako na ilustraci 5. Tyto data jsou vzorkována po určitých stupních v prostoru. Proto se pro výsledné hledání svítivosti I_v musí použít bilineární interpolace pro zjištění mezi mezními hodnotami.

3.1.2 Hledání dat ve vybraném světelném profilu za pomoci bilineární interpolace

Bilineární interpolace se vyznačuje počítáním průměrné hodnoty mezi čtyřmi body Q_{11} , Q_{12} , Q_{21} a Q_{22} . Tyto body představují svítivosti I_v v určitých úhlech. Musíme zjistit vzdálenosti a a b , které představují horizontální a vertikální odchylky od naměřených hodnotách svítivosti v daných úhlech, tady naše body y Q_{11} , Q_{12} , Q_{21} a Q_{22} . Ze vztahu (12), lze vidět, jak se tyto vzdálenosti a a b určují, kde y_1 a

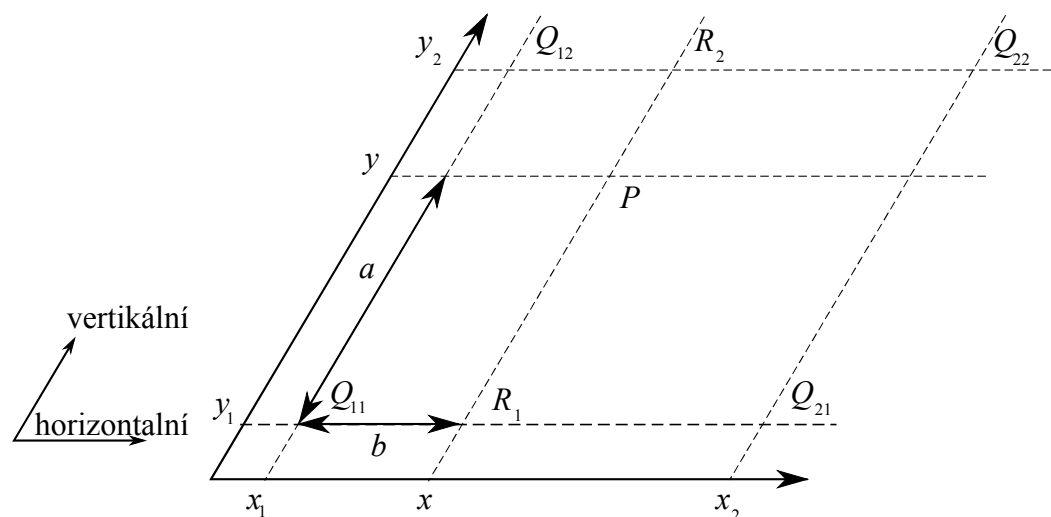
Implementace světelných map

y_2 jsou zaznamenané hodnoty, které jsou mezi hledanou hodnotou vertikálního úhlu y a x_1 a x_2 jsou uložené hodnoty, které jsou mezi hledanou hodnotou horizontálního úhlu x , jak můžeme vidět na ilustraci 8.[3]

$$a = \frac{(y - y_1)}{(y_2 - y_1)} \quad b = \frac{(x - x_1)}{(x_2 - x_1)} \quad (12)$$

Jestliže známe všechny potřebné hodnoty, které byly zmíněny, můžeme podle vztahu (13) vypočítat výslednou svítivost I_v v daném úhlu.

$$I_v = a(bQ_{22} + (1-b)Q_{21}) + (1-a)(bQ_{12} + (1-b)Q_{11}) \quad (13)$$



Ilustrace 8: Hledání hodnot svítivosti I_v z naměřeného světelného profilu IES, pomocí Bilineární interpolace

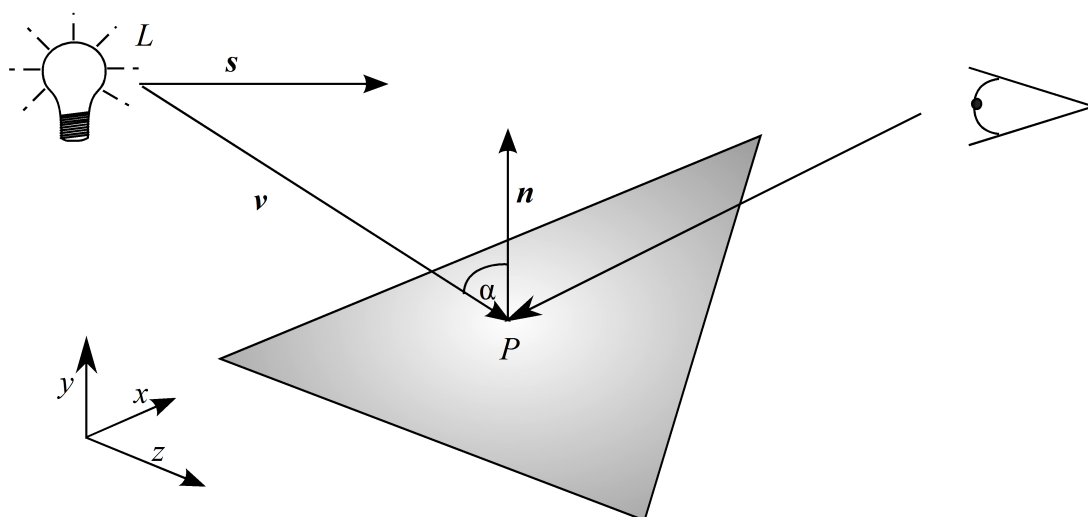
Pro vyhledávání nejbližších 4 bodů Q_{11} , Q_{12} , Q_{21} a Q_{22} hodnot byla využita metoda půlení intervalů. Známe-li svítivost I_v v konkrétním úhlu, použijeme tuto hodnotu při samotném zjišťování osvětlení E_v dané plochy.

Implementace světelných map

Pro převod mezi mírou obloukovou a stupňovou používáme následující konstanty, popsané ve vztazích(14).

$$R2D = \frac{180}{\Pi} \quad DWR = \frac{\Pi}{180} \quad (14)$$

3.2 Raytracing



Ilustrace 9: Průběh metody raytracing, kde dochází k vystřelování polopřímek z oka pozorovatele a hledání průsečíku s tělesem. Po nalezení průsečíku objektu a polopřímky se zjišťuje výsledné osvětlení daného bodu

Tato metoda pro vizualizaci scény je charakteristická svou výpočetní složitostí, protože v systému VRUT byla tato rendrovací metoda použita, muselo dojít k implementaci správného algoritmu pro tuto techniku vykreslování obrazu, jak už bylo zmíněno. Hlavní myšlenka této vykreslovací metody tkví v nalezení průsečíku P objektu, jako je například koule, trojúhelník, toroid s pomyslnou polopřímkou. Tato polopřímka, která má počátek v lidském oku v počítačové grafice se jedná o kameru scény a každý pixel na obrazovce odpovídá jedné polopřímce. Po nalezení tohoto bodu P se zjišťuje z okolních světel jakou polohu L k tomuto bodu P v prostoru má, proto se tato metoda nazývá zpětné sledování paprsku, kvůli opačnému zjišťování osvětlení E_v plochy, kde nevysíláme paprsky ze světla, ale zjišťujeme,

Implementace světelných map

zda nějaký světelný paprsek doputuje do určitého bodu P , průběh metody lze vidět na ilustraci 9. [3]

3.2.1 Popis získávání konkrétních hodnot svítivosti pro zobrazování osvětlení plochy

V této kapitole se budeme zajímat hlavně o světelné paprsky, které se šíří od zdroje světla a dopadají na objekt. Mějme tedy polohu zdroje světla L a nalezený průsečík P v globálním souřadnicovém systému. Máme-li vektor v , který směřuje od světelného zdroje k průsečíku P a směrový vektor světla s , který určuje směr světla, dokážeme zjistit úhly světelných paprsků od směru světla.

Vektor v mezi těmito body umožňuje určit úhel α , pod kterým tento paprsek dopadá na osvětlenou plochu. Z normály n dané plochy, tento úhel α lze vypočítat vztahem (15).

$$\cos(\alpha) = \frac{n \cdot v}{|n| \cdot |v|} \quad (15)$$

Vzdálenost r od zdroje světla k průsečíku P je dána velikostí nenormalizovaného vektoru v (16), kterou potřebujeme znát pro výsledný výpočet osvětlení E_v (9).

$$r = |v| \quad (16)$$

Samotnou svítivost I_v , poslední složku, pro výpočet osvětlení E_v (9) plochy se musí dohledat ve světelném profilu IES. Naše světlo se nachází a tedy vektor v v globálním souřadném systému, pro zjištění hodnot se musí světelný paprsek přetransformovat do lokálního souřadného systému pomocí inverzní matice konkrétního světla $M_{L_x}^{-1}$ podle vztahu (17).

$$v' = M_{L_x}^{-1} v \quad (17)$$

Získaný normalizovaný vektor v' , který se už vyskytuje v lokálním souřadném systému nám poslouží pro zjištění samotné svítivosti I_v , rozdělením tohoto vektoru na

Implementace světelných map

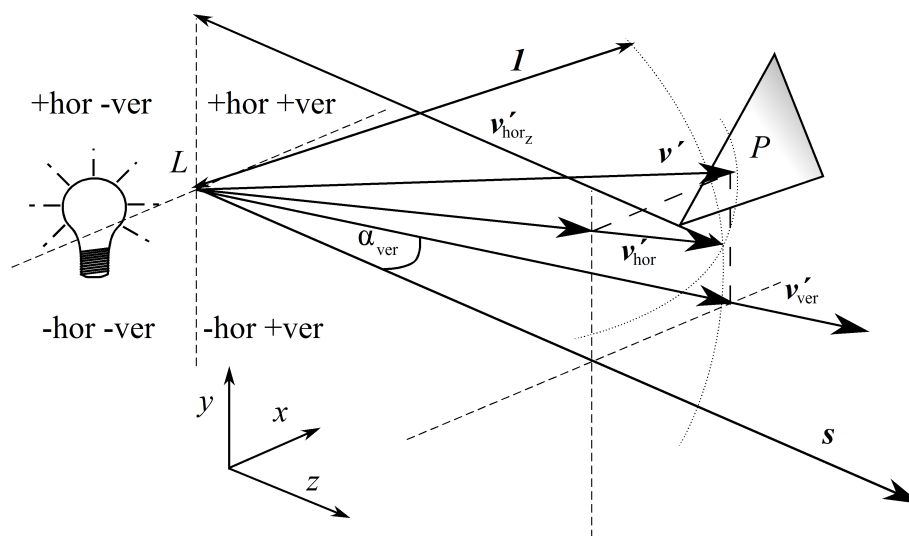
v'_{ver} a v'_{hor} , kde každý z těchto vektorů je rovnoběžný s určitou osou v'_{ver} $[0, v'_y, v'_z]$ a v'_{hor} $[v'_x, 0, v'_z]$.

Po normalizaci těchto vektorů v'_{ver} a v'_{hor} můžeme získat úhlové odchylky od základního směru světla s , tak že spočítáme mezi těmito třemi normalizovanými vektory skalární součin podle vztahu (18).

$$ver = \cos^{-1}\left(\frac{v'_{ver} \cdot s}{|v'_{ver}| |s|}\right) R2D \quad hor = \cos^{-1}\left(\frac{v'_{hor} \cdot s}{|v'_{hor}| |s|}\right) R2D \quad (18)$$

Skalární součin můžeme obejít tak, že získáme ze zetové složky pomocných vektorů v'_{ver_z} a v'_{hor_z} úhlové odchylky od základního směru světla s podle vztah(19).

$$ver = \cos^{-1}(v'_{ver_z}) R2D \quad hor = \cos^{-1}(v'_{hor_z}) R2D \quad (19)$$



Ilustrace 10: Hledání horizontální složky, pro zjištění svítivosti v určité úhlové odchylce od základního směru světla za pomoci z-tové složky pomocného vektoru bez použití skalárního součinu a zjištění vertikální složky za pomoci skalárního součinu mezi pomocným vektorem a vektorem udávajícího základní směr světla.

Implementace světelných map

Tyto hodnoty úhlových odchylek mohou být jak v kladném tak záporném prostoru, tato informace je zjistitelná z nenulových hodnot normalizovaných vektoru, tedy u vertikální složky se jedná o osu y $v'_{ver,y}$ a u horizontální složky se jedná o osu x $v'_{ver,x}$. Je-li jedno z těchto čísel záporné, jedná se o záporný prostor a obráceně, zda-li je kladné, mluvíme o kladném prostoru. Na ilustraci 10, vidíme ukázkou samotného určování těchto složek.

Z vypočítaných hodnot *ver* a *hor* složky je nalezena hodnota svítivosti I_v podle vztahu (13). Z těchto nalezených hodnot svítivosti I_v v určitém úhlu, vzdálenosti r plochy od zdroje světla a úhlu α , pod kterým paprsek dopadá na objekt, dokážeme podle vztahu (9) spočítat osvětlení E_v dané plochy.

3.3 Objemová textura

Tato varianta výsledné vizualizace se vyznačuje snížením samotného výpočtu výsledného obrazu, ale za cenu velké paměťové náročnosti. Z důvodu úspory času při renderování v reálném čase, muselo dojít ke snížení kalkulací v samotném výpočtu celkového osvětlení E_v při konkrétní scéně. Bylo vyzorováno, že některé hodnoty mohou být vypočítány před samotným generováním výsledného obrazu. Zaměřme se tedy na vztah (9) samotného osvětlení E_v . Svítivosti I_v v konkrétních úhlech od směru světla jsou známy z naměřených hodnot, uložené ve světelném profilu IES, které získáme pomocí vztahu (13). Toto elektromagnetické vlnění se šíří určitým nekonečným prostorem. V tomto pomyslném prostoru dokážeme zjistit konkrétní osvětlení E_v , v určité vzdálenosti r od světelného zdroje v určitém úhlu od směru světla. Tyto světelné paprsky, ale mohou zasáhnout jakékoliv objekty ve scéně při samotném vykreslování a tedy úhel α , pod kterým dopadají na osvětlenou plochu objektu, není konstantní. Touto úvahou si můžeme vztah osvětlení (9) rozdělit na dvě části E_{v1} a E_{v2} .

$$E_{v1} = \frac{I_v}{r^2} \quad (20)$$

Vztah (20) vyjadřující část osvětlení E_{v1} , které je nezávislé na poloze a případném dopadu světla na objekty ve scéně, lze tedy o něm tvrdit, že je konstantní (neměnný) a šíří se prázdným nekonečným prostorem.

Implementace světelných map

$$E_{v2} = \cos(\alpha) \quad (21)$$

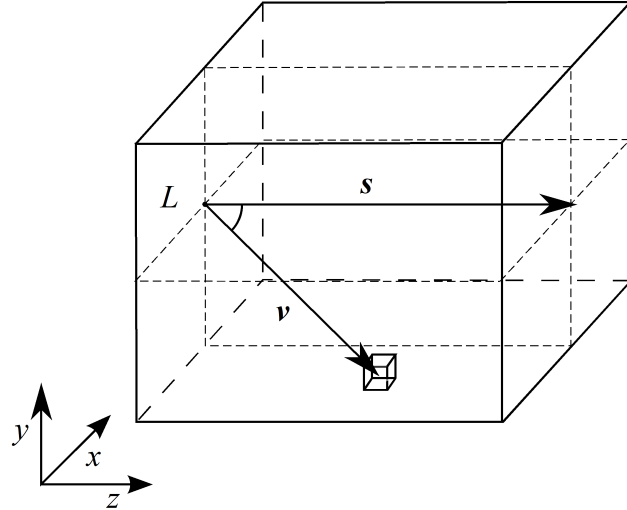
Vztah (21) určuje část osvětlení E_{v2} , které je závislé na úhlu α dopadajícího světelného paprsku na osvětlenou plochu. Tento úhel α dokážeme určit až při konkrétním rozvržení scény renderovací scény.

3.3.1 Předgenerování dat části osvětlení v určitých úhlových odchylkách od základního směru světla a vzdálenosti od zdroje světla, před samotným uložením do objemové textury a renderováním

Tato myšlenka rozdělení vztahu na dva, nám poskytne ušetření drahocenného výpočetního času, při samotném výpočtu výsledného osvětlení E_v plochy, kde hodnoty části osvětlení E_{v1} jsou předvypočítány před samotným vykreslováním scény v reálném čase. Světelné paprsky, jak už bylo zmíněno, podléhají útlumu se zvětšující se vzdálenosti a tudíž prostor, ve kterém se šíří světlo lze reprezentovat v malém prostoru a ne v nekonečnu. Protože tyto vzdálené paprsky jsou už natolik utlumené, že zanikají a výslednou scénu nemohou jakkoliv ovlivnit, můžeme je tedy ignorovat. Proto byla vybrána objemová textura, jako médium pro uložení konstantních dat části osvětlení E_{v1} , která nám tento konečný prostor může dostatečně reprezentovat.

Jedná se tedy o práci s uspořádanými skalárními daty v E^3 . Mějme souřadnice i_x , i_y , i_z tedy indexy jednotlivých buněk v objemové textuře a veličinu e tedy hodnotu voxelu nesoucí konkrétní část osvětlení E_{v1} v daném bodě ve vytyčeném prostoru $\{ \langle i_x, i_y, i_z \rangle, e \}$ v určité vzdálenosti a úhlu od zdroje světla. Mluvíme-li o uspořádaných skalárních datech, pak vzdálenosti mezi jednotlivými body v prostoru Δx , Δy a Δz jsou vždy konstantní.

Implementace světelných map



Ilustrace 11: Šíření paprsku světla ve vymezeném prostoru, kde tento prostor je reprezentován objemovou texturou

Na ilustraci 11 lze vidět pomyslný prostor, vyjádřený objemovou texturou, která nám umožňuje zjistit, jak vzdálenost r od zdroje světla, tak úhel ve kterém se světelný paprsek šíří vůči směru světla v určitém bodě za pomoci indexů $\langle i_x, i_y, i_z \rangle$ v naší 3D textuře. Vzdálenost mezi sousedními voxely Δx , Δy a Δz se volí podle počtu buněk N_x , N_y a N_z v objemové textuře v každém možném směru a správně zvoleného měřítka m pro zachování relativních vzdáleností ve scéně pro reálné zobrazení, které nám popisuje vztah (22).

$$\Delta x = \frac{m}{N_x} \quad \Delta y = \frac{m}{N_y} \quad \Delta z = \frac{m}{N_z} \quad (22)$$

Pro výpočet osvětlení E_{vl} v určitém subprostoru, tedy ve voxelu v naší objemové textuře, potřebujeme znát polohu zdroje světla v prostoru textury v lokálním souřadném systému. Umístíme-li zdroj světla do středu roviny objemové textury tak, že z souřadnice nabývá hodnoty $0 \leq z \leq 1$ a směr světla s je rovnoběžný s osou z v lokálním souřadném systému, poté dostáváme nejlepší možnou polohu a směr světla v objemové textuře, kde se toto světlo může šířit od -90 do $+90$ horizontálně i vertikálně, jak lze vidět na ilustraci 11. Pro výpočet dané části osvětlení E_{vl} potřebujeme zjistit přesnou polohu počátku světelného zdroje L (23) v pomyslném prostoru a bod Q (24). Mezi těmito dvěma body je výsledný vektor určující základní směr světla s , který je rovnoběžný s osou z , jak jde

Implementace světelných map

vidět z ilustrace 11.

$$L = \left[\frac{N_x}{2} \Delta x, \frac{N_y}{2} \Delta y, 0 \right] \quad (23)$$

$$Q = \left[\frac{N_x}{2} \Delta x, \frac{N_y}{2} \Delta y, N_z \Delta z \right] \quad (24)$$

Pro výpočet daného bodu P v pomyslném prostoru korespondující s indexy v objemové textury použijeme vztah (25).

$$P = [i_x \Delta x, i_y \Delta y, i_z \Delta z] \quad (25)$$

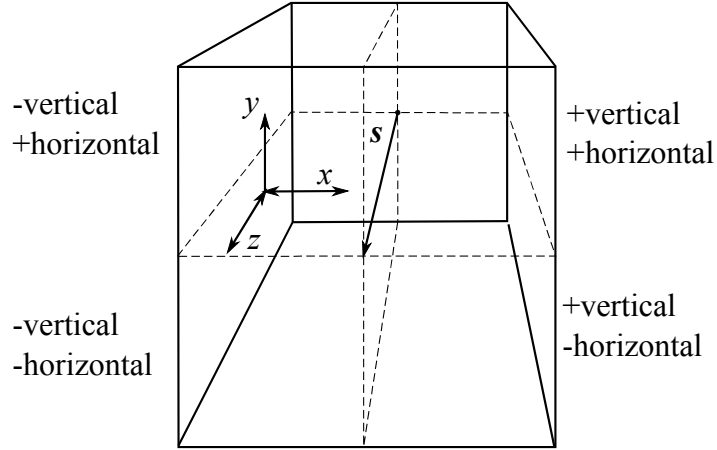
Známe-li tento bod P (25) a počátek zdroje světla L (23) v prostoru, vytvoříme mezi těmito body vektor v , který odpovídá danému světelnému paprsku. Z vytvořeného normalizovaného vektoru v mezi určitým voxelem a zdrojem světla, lze určit jednoduše úhel, který tento vektor v svírá se směrovým vektorem světla s , jak lze vidět na vztahu (26).

$$\cos(\alpha) = \frac{v \cdot s}{|v| \cdot |s|} \quad (26)$$

Vzdálenost r od zdroje světla k voxelu je dána velikostí nenormalizovaného vektoru v (16), kterou potřebujeme znát pro výsledný výpočet části osvětlení $E_{v,l}$ (20). V obou dvou složkách pro výpočet části osvětlení $E_{v,l}$ lze vypočítat, že na jejich hodnoty budou mít vliv vzdálenosti Δx , Δy a Δz mezi jednotlivými buňkami objemové textury, popsané ve vztahu (22).

Svítivost I_v v určitém úhlu od směru světla je zjistitelná z naměřených hodnot uložených v IES profilu. Všechny tyto hodnoty nám umožňují vygenerovat námi definovanou objemovou texturu pro záznam konstantních dat ve tvaru $\{ \langle i_x, i_y, i_z \rangle, e \}$ v E^3 , jak už bylo řečeno. Z ilustrace 12 jde jasně vidět, jak je tento prostor rozdělen do 4 základních prostorů kvůli hodnotám směru v IES profilu, tedy zaznamenané hodnoty v objemové textuře od směru světla od -90 do +90 horizontálně a vertikálně.

Implementace světelných map



Ilustrace 12: Rozdělení objemové textury podle základního směru světla s do čtyř základních prostorů

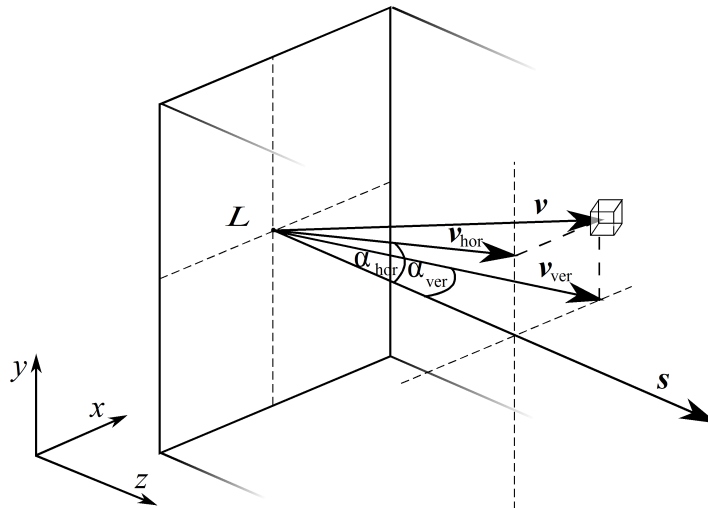
Z důvodu , že jsou data o svítivosti v IES. formátu zapsaná podle úhlové velikosti vertikálních a horizontálních složek, musíme tento vektor v reprezentující paprsek světla rozdělit na dva vektory v_{hor} a v_{ver} , pro výpočet těchto úhlových složek. Tyto dva vektory v_{hor} a v_{ver} se vyznačují tím, že jsou vždy ve stejné rovině se směrovým vektorem světla s , kde jsou tyto roviny rovnoběžné buďto s osou x anebo y . Tedy vektor v_{hor} leží v rovině horizontální a v_{ver} v rovině vertikální. Tyto vektory v_{hor} a v_{ver} jsou popsány vztahem (27).

$$v_{ver} = [L_x, i_y \Delta y, i_z \Delta z] \quad v_{hor} = [i_x \Delta x, L_y, i_z \Delta z] \quad (27)$$

Máme-li tyto dva vektory, můžeme zjistit svítivost v určitém prostorovém úhlu v obou dvou složkách za použití vztahů (28), jak lze vidět na ilustraci 13.

$$\alpha_{hor} = \frac{v_{hor} \cdot s}{|v_{hor}| \cdot |s|} \quad \alpha_{ver} = \frac{v_{ver} \cdot s}{|v_{ver}| \cdot |s|} \quad (28)$$

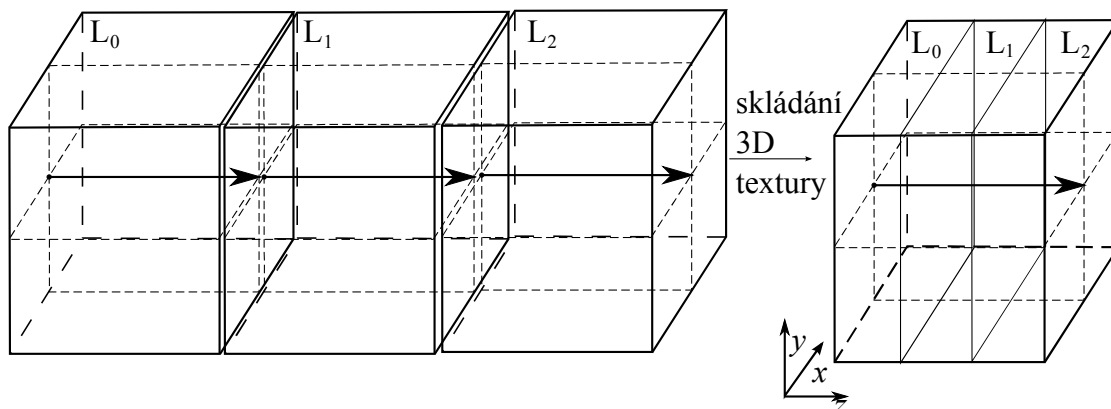
Implementace světelných map



Ilustrace 13: Výpočet horizontální a vertikální složky úhlové odchytky od základního směru světla s , pro zjištění svítivosti I_v ze světelného profilu IES

Chceme-li rozšířit scénu o více světel, dochází k problému. Nelze totiž použít více objemových textur tzv. pole těchto 3D textur. Pro použití více světel ve scéně bylo využito skládání jednotlivých objemových textur za sebe, čímž nám roste pouze hloubka hledaných dat námi generované textury. Pro nesnižující se kvalitu, dochází k navýšení počtu voxelů N_z výsledné objemové textury v tomto směru a tedy vzájemná vzdálenost buněk Δz (22) se nemění. Pro oddělení dat takto seřazených textur, se mezi každé dvě vkládá prázdný prostor, tedy rovina prázdných buněk na konec záznamu každého konkrétního světla, které nám tyto světelné profily navzájem dokáže oddělit, jak lze vidět na ilustraci 14.

Implementace světelných map



Ilustrace 14: Skládání objemových textur za sebe v z-tové ose, pro její sjednocení do jedné, pro možnost záznamu více světelných profilů v jedné scéně

Po zkonstruování těchto konstantních dat v našem případě vytvoření 3D textury, můžeme přejít k samotnému zobrazení za pomoci API OpenGL s využitím shaderů a dopočítání celkového osvětlení E_v scény, kde tyto předvypočítaná data využijeme. Při samotném generování obrazu, jak už bylo řečeno, se bude dopočítávat část osvětlení E_{v2} , které se poté vynásobí s nalezenou hodnotou v objemové textuře E_{v1} , která je konstantní k libovolné scéně a dostaneme námi hledané výsledné osvětlení E_v , jak lze vidět ve vztahu (29).

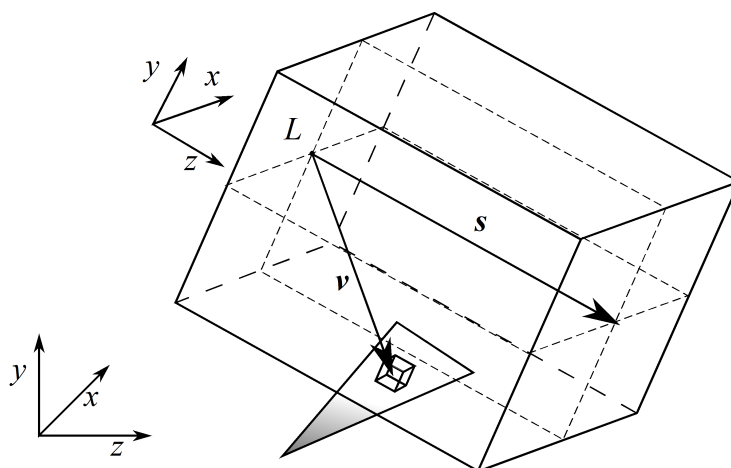
$$E_v = E_{v1} E_{v2} \quad (29)$$

Jak už bylo zmíněno část osvětlení E_{v2} musí být dopočítána až při vykreslování samotné scény, kde vektor v je ve směru od počátku zdroje světla k povrchu objektu představující světelný paprsek a vektor s vyjadřuje směr světlometu, který můžeme libovolně měnit, jak lze vidět na ilustraci 15. K aktivní manipulaci do všech tří směrů se světlem ve scéně bylo využito transformačních matic viz. Kapitola „transformace“. Známe-li počátek zdroje světla a normálu plochy n , na kterou tento světelný paprsek v dopadá v globálním souřadném systému, pak tento vektor v nám umožňuje určit úhel α , pod kterým tento paprsek dopadá na tuto osvětlenou plochu, jak jde vidět ze vztahu (15). Tato hodnota nám umožní určit světelný útlum pro výpočet části osvětlení E_{v2} , pro výslednou kalkulaci celkového osvětlení E_v .

Implementace světelných map

3.3.2 Hledání koordinátů v objemové textuře pro zjištění předgenerovaných hodnot části osvětlení pro výpočet celkového osvětlení plochy

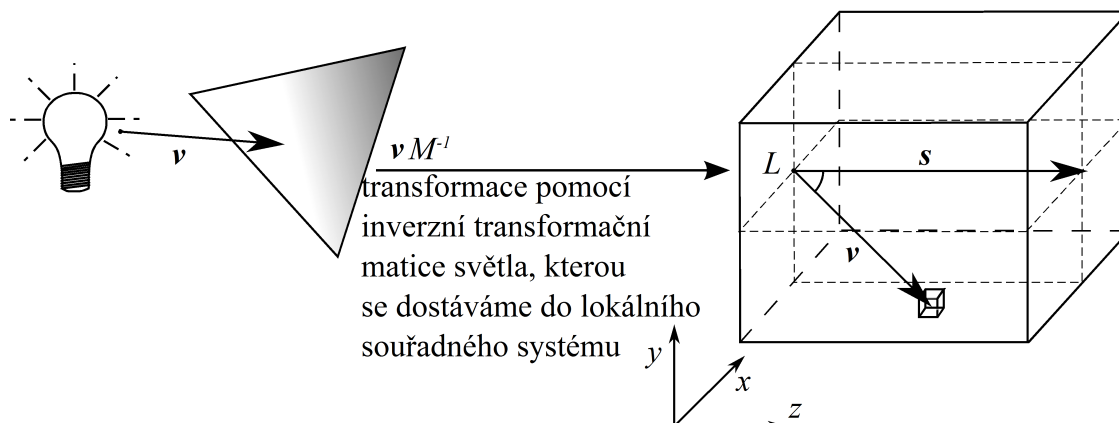
Na ilustraci 15 lze vidět funkci samotného předprocesu generování objemové textury, která koresponduje se samotným hledáním indexů v této textuře. Tato část je zaměřená na samotné hledání indexů v 3D textuře.



Ilustrace 15: Hledání indexů v objemové textuře v odpovídající vzdálenosti r a v určitém úhlu od základního směru světla s

Jak už bylo zmíněno, pro manipulaci světla potřebujeme pracovat s transformační maticí. Pro transformaci z globálního souřadnicového systému do lokálního souřadného systému budeme nuceni použít inverzní maticí konkrétního světla $M_{L_x}^{-1}$. Známe-li směr a velikost světelného paprsku v ve scéně vycházejícího od zdroje světla a končícího na osvětleném objektu, který se nalézá v globálním souřadném systému, dokážeme za pomoci inverzní matice daného světla $M_{L_x}^{-1}$, transformovat tento vektor v na vektor v' , který se vyskytuje v počátku lokálním souřadném systému daného světla, jak vidíme ze vztahu (17) zobrazeného na ilustraci 16.

Implementace světelných map



Ilustrace 16: Transformace světelného paprsku popsaného vektorem v z globálního souřadného systému do lokálního souřadného systému pomocí inverzní transformační matice

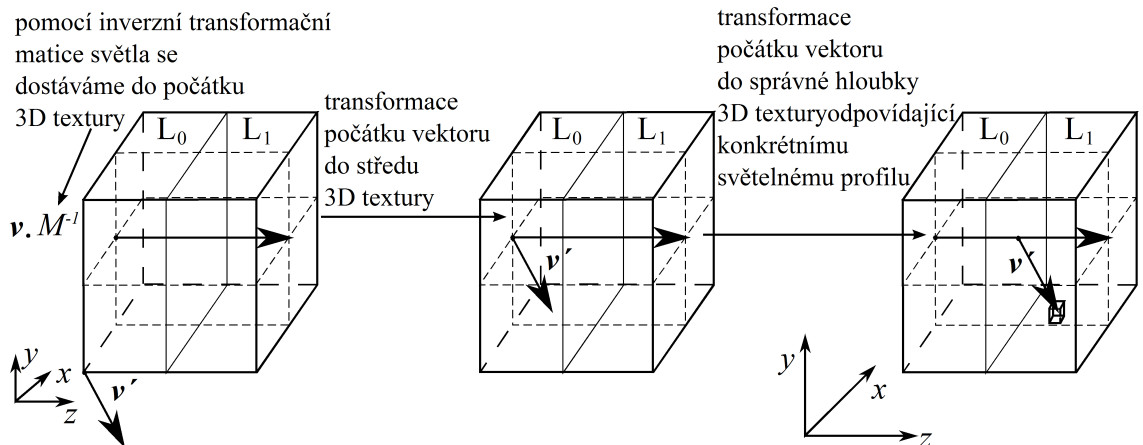
$$[M^{-1}]_{L_x}$$

V předchozí kapitole, kde jsme se zabývali generováním samotné objemové textury, bylo řečeno, že nejvhodnější umístění zdroje světla je do středu roviny, kde z souřadnice nabývá hodnoty $0 \leq z \leq 1$ a směr světla s je tedy rovnoběžný se souřadnou osou z . Chceme-li začínat na této pozici, při hledání koordinátů v objemové textuře, které nabývají hodnot $\langle 0,1 \rangle$ ve všech třech možných osách a víme, že transformovaný světelný paprsek v' začíná v počátku lokálního souřadnicového prostoru světla tedy $[0,0,0]$. Pak tedy musíme transformovaný paprsek v' posunout ve dvou osách x a y o polovinu velikosti objemové textury. Proto, že se v textuře pohybujeme od 0 do 1 , tak musíme vektor v' posunout v těchto dvou osách o $[0.5, 0.5, 0]$. Z důvodu, že záznam více světél je řešen ukládáním do jedné objemové textury, musíme tedy světlu vymezit pomyslný počátek a konec v hloubce objemové textury, kde se bude vyskytovat konkrétní světelný profil světla. Mějme konečný počet profilů světél n a index i , který nám určuje o, které světlo se jedná v množině profilů světél. Víme, že textura má délku 1 , tedy náš hloubkový rozsah v z

ose v textuře se tedy vymezí na $\langle \frac{1}{n}(i-1), \frac{1}{n}(i) \rangle$, který musíme zahrnout do výsledného hledání koordinátů v objemové textuře. Z těchto důvodů musíme vektor v' zmenšit pouze v z souřadnici o počet profilů světél n a posunout počátek světelného paprsku o $[0.0, 0.0, \frac{1}{n}(i)]$. Na ilustraci 17 lze vidět všechny tyto operace seřazené ve správném pořadí, pro zjištění koordinátů v objemové textuře. Hodnoty těchto koordinátů se nastaví podle vztahu (30). Na těchto koordinátech se vyskytuje část osvětlení $E_{v,i}$, pro celkový výpočet hledaného osvětlení E osvětlené plochy.

Implementace světelných map

$$\langle i_x, i_y, i_z \rangle = [v'_x + 0.5, v'_y + 0.5, v'_z \frac{1}{n} + \frac{1}{n} i] \quad (30)$$



Ilustrace 17: Převod světelného paprsku v' vyskytujícího se v lokálním souřadném systému světla tak, aby jeho koordináty respektovaly základní polohu a směr světla s vymezily se k odpovídajícímu světelnému profilu ze všech možných uložených světelných profilů v jedné objemové textuře

Máme-li všechny části osvětlení E_{v1} a E_{v2} můžeme vypočítat výsledné osvětlení E_v v naší scéně, podle vztahu (15).

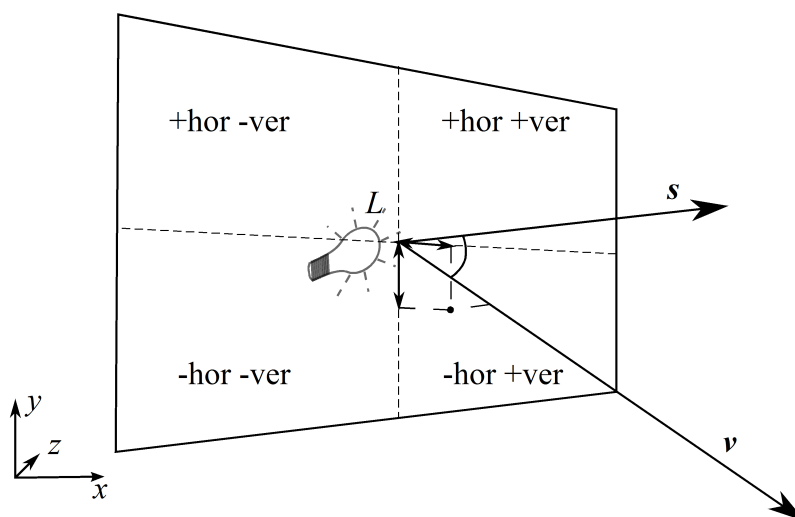
3.4 Dvourozměrná textura

Tato varianta výsledné vizualizace se vyznačuje sice komplikovanějším výpočtem výsledného osvětlení E_v plochy, tedy pomalejším vykreslováním samotné scény, ale za cenu poměrně menší paměťové náročnosti, při zachování detailnějšího osvětlení zobrazované scény. Nejen z důvodu úspory paměťových nároků byla použita tato varianta za použití dvojrozměrné textury pro renderování v reálném čase, jako hlavní metoda pro zjišťování celkového osvětlení E_v při konkrétním rozvržení scény. Přes tuto 2D texturu se předávaly data svítivosti I_v do shaderů v API OpenGL a indexy v této textuře odpovídaly konkrétním úhlům od základního směru světla popsané vektorem s .

Implementace světelných map

3.4.1 Generování dat svítivosti v určitých úhlových odchylkách od základního směru světla a následné ukládání do dvourozměrné textury

Opět se tedy jedná o práci s uspořádanými skalárními daty, ale už ne v E^3 , jak tomu bylo v objemové textuře, ale v E^2 . Mějme indexy i_x a i_y tedy souřadnice v naší textuře a veličinu e , tedy hodnotu texelu, nesoucí v sobě konkrétní svítivost. Mluvíme-li o uspořádaných skalárních datech, pak vzdálenosti mezi jednotlivými body v rovině textury Δx a Δy musí být vždy konstantní. V tomto případě koordináty určují horizontální a vertikální úhel vůči směru světla viz ilustrace 18.



Ilustrace 18: Rozvržení dvourozměrné textury do čtyř základních sektorů, podle základního směru světla s a ukázka ukládání hodnot svítivosti I_v podle vektoru v na konkrétní indexy dvourozměrné textury

Pro univerzálnost na změnu velikosti textury musel být použit jednoduchý vztah, který vypočítá konstantní vzdálenosti mezi texely, podobně jako tomu bylo u vzdálenosti mezi sousedními voxely, při počítání hodnot v objemové textuře.

Ve vztahu (31) jsou tyto vzdálenosti Δx a Δy mezi sousedními texely popsány, kde hor_{max} a ver_{max} jsou maximálně možné prostorové úhly v obou směrech tedy 360° a N_x N_y jsou počty texelů v daných osách.

$$\Delta y = \frac{ver_{max}}{N_y} \quad \Delta x = \frac{hor_{max}}{N_x} \quad (31)$$

Implementace světelných map

Po zjištění konstantních vzdáleností mezi texely Δx a Δy , nám tyto hodnoty umožní pro každý texel, vypočítat úhlovou odchylku v obou dvou možných směrech, jak ve vertikálním tak v horizontálním směru. Směr světla má pro tyto dvě složky nulovou hodnotu, tedy musí být posunutý o půlku maximálního rozsahu $\frac{ver_{max}}{2}$ a $\frac{hor_{max}}{2}$, aby se počátek dostal do středu dvourozměrné textury. Na vztahu (32) lze vidět, jak se pro každý z indexů v textuře $\{<i_x, i_y>, e\}$ spočte jedna z úhlových složek.

$$ver = (i_x \Delta x) - \frac{ver_{max}}{2} \quad hor = (i_y \Delta y) - \frac{hor_{max}}{2} \quad (32)$$

Tyto hodnoty vertikální a horizontální složky nám umožní zjistit svítivost I_v v daném úhlu ze světelného profilu IES za pomoci indexace texelů. Pro uložení více světél bylo využito opět skládání textur, tedy vkládáním za sebe v ose x , čímž roste pouze rozměr textury, ale konstantní vzdálenost Δx mezi texely zůstává a tady vztahy (22) se nemění, až při samotném hledání přijde na řadu vyhledávání profilu podle indexu.

3.4.2 Výpočet osvětlení z dat svítivosti uložené v dvourozměrné textuře

Po vygenerování dvourozměrné textury, která slouží jako úschovna konstantních dat svítivosti I_v a není v ní část osvětlení E_{vl} , jak tomu bylo u objemové textury. Tyto vygenerované data svítivosti I_v nám poslouží pro výpočet samotného osvětlení E_v plochy ve scéně. Známe-li vektor s , tedy směr světla, který lze získat z transformační matice světla M_{L_i} a vektor v , který má počátek ve zdroji světla a dopadá na naši plochu ve scéně. Dokážeme zjistit osvětlení ze vztahu (9) na dotyčné ploše. Vzdálenost r lze získat z velikosti vektoru v , a úhel α světelného útlumu, který je vypočten ze skalárního součinu normalizovaných vektorů světelného paprsku v a normály k dané ploše n , který byl popsán ve vztahu(15).

3.4.2.1 Varianta inspirovaná raytracingem

Poslední hodnotu pro výpočet samotného osvětlení E_v ozářené plochy, která je obsažena v předgenerovaných datech, tedy svítivost I_v v určitém úhlu od základního směru světla s . Tyto data jsou, jak už bylo zmíněno v dvourozměrné textuře pro zpracování na API

Implementace světelných map

OpenGL za použití shaderů. Toto API nám umožňuje hardwarovou interpolaci mezi jednotlivými sousedními texely. Vektor v , tedy směr světelného paprsku, je v globálním souřadnicovém systému, aby se tento vektor dostal do lokálního souřadnicového systému musíme použít vztah (17) a dostáváme vektor v' . Pro získání indexu v naší dvourozměrné textuře můžeme použít stejné vztahy, které byly použity u implementace do raytracingu vztah (18) nebo (19). Tyto vztahy se upraví pro přesné nalezení svítivosti I_v . Vztahy pro toto zjištění koordinátů v textuře vypadá následovně (33) a (34), tím že použijeme skládání textur v ose x za sebe dochází v x koordinátu k vymezení začátku a zmenšení samotné hodnoty v této ose pro přesné nalezení odpovídajících dat, kde index i značí index konkrétního světelného profilu v dvourozměrné textuře.

$$\langle i_x, i_y \rangle = \left[\frac{\cos^{-1}\left(\frac{v'_{hor} \cdot S}{|v'_{hor}| |S|}\right) R2D}{hor_{max}} \frac{1}{n} + \frac{1}{n} i, \frac{\cos^{-1}\left(\frac{v'_{ver} \cdot S}{|v'_{ver}| |S|}\right) R2D}{ver_{max}} \right] \quad (33)$$

$$\langle i_x, i_y \rangle = \frac{\cos^{-1}(v'_{hor_z}) R2D}{hor_{max}} \frac{1}{n} + \frac{1}{n} i, \frac{\cos^{-1}(v'_{ver_z}) R2D}{ver_{max}} \quad (34)$$

3.4.2.2 Varianta inspirovaná objemovou texturou

Tato varianta byla dostatečná, ale inspirováním objemové textury, se dospělo k vhodnější variantě zjišťování koordinátů v samotné dvojrozměrné textuře. Toto řešení potřebuje taktéž transformovat vektor v z globálního souřadnicového systému do lokálního souřadnicového systému na vektor v' , který musí být znormalizován, jak je tomu stejně u všech zmíněných variantách. Hlavní myšlenka je založena na změně velikosti vektoru v' tak, aby z složka vektoru v'_z nabývala hodnoty 1, zjištění parametru d , který tento vektor vektoru v' , který tuto podmínku bude splňovat, lze vidět ve vztahu (35).

$$d = \frac{1}{v'_z} \quad (35)$$

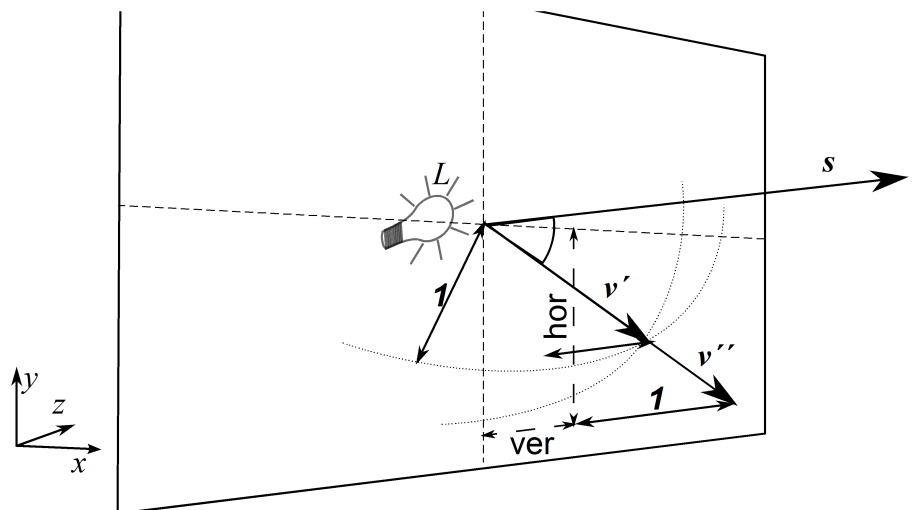
Zjištěný parametr d ze vztahu (35) můžeme tento vektor v' přetransformovat pomocí vztahu (36) na vektor v'' .

Implementace světelných map

$$v'' = dv' \quad (36)$$

Tento zmodifikovaný vektor v'' nám pomůže při hledání koordinátů v dvourozměrné textuře. Protože se tento paprsek může šířit v rozmezí od -180° do 180° od směrového vektoru světla s a každý sektor je rozdělen po 90° , kde znaménka korespondují s konkrétním prostorem, musíme tedy tento vektor vynásobit $\frac{1}{4}$. Jako poslední krok musíme tento vektor v'' posunout do hodnot mezních hodnot koordinátů $\langle 0,1 \rangle$ dvourozměrné textury konkrétního světelného profilu podle vztahu (37). Tento postup je znázorněn na ilustraci 19.

$$\langle i_x, i_y \rangle = \left[\left(\frac{1}{n}n + (v''_x \frac{1}{4} + 1) \frac{1}{2} \right) \frac{1}{n}, (v''_x \frac{1}{4} + 1) \frac{1}{2} \right] \quad (37)$$



Ilustrace 19: Nalezení koordinátů ve dvourozměrné textuře za pomoci světelného paprsku v lokálním souřadnicovém systému, který se transformuje podle vztahů (35), (36) a (37)

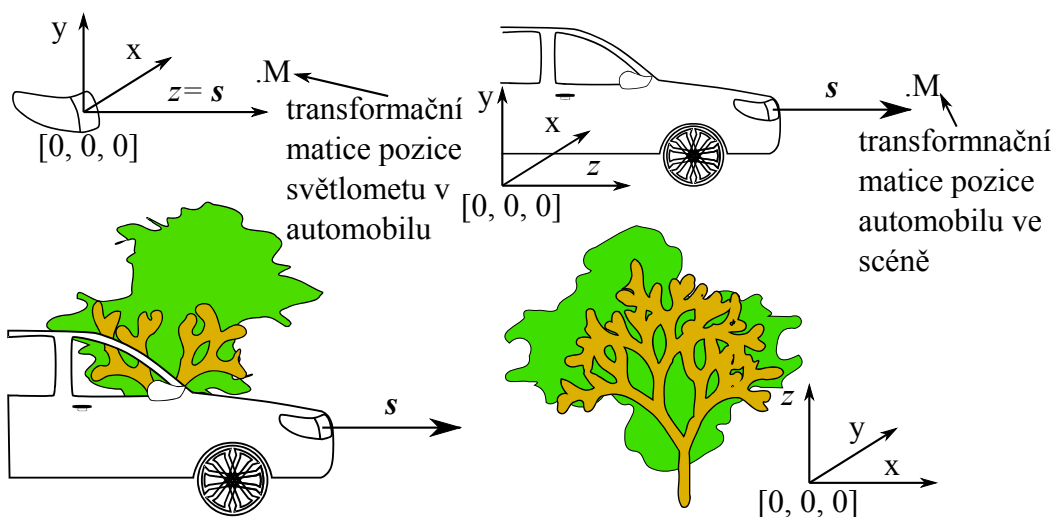
Jak lze vidět z ilustrace 19 tento postup je jednodušší, ale i přesto nedošlo ke zrychlení oproti variantě popsané vztahem (34) při samotném vykreslování scény. Naměřené hodnoty jsou v kapitole výsledky.

Implementace světelných map

3.5 Transformace světla ve scéně

Pro interaktivnost scény nemůže být světlo reprezentováno jen jako statický objekt, tedy neměnné, ale musí umět měnit směr a polohu vůči scéně. Pro příklad této dynamičnosti můžeme uvést světlomet, který je součástí automobilu, kde polohu a směr světla udává poloha a směr samotného automobilu ve scéně. Tento automobil musí být nějak reprezentován v prostoru, proto se v počítačové grafice nejčastěji používá pro objekty ve scéně transformační matice

M , které nám umožní tento objekt přetransformovat z lokálního souřadného systému objektu do globálního souřadného systému scény a naopak pomocí inverzní transformační matice M^{-1} , jak uvádí [3].



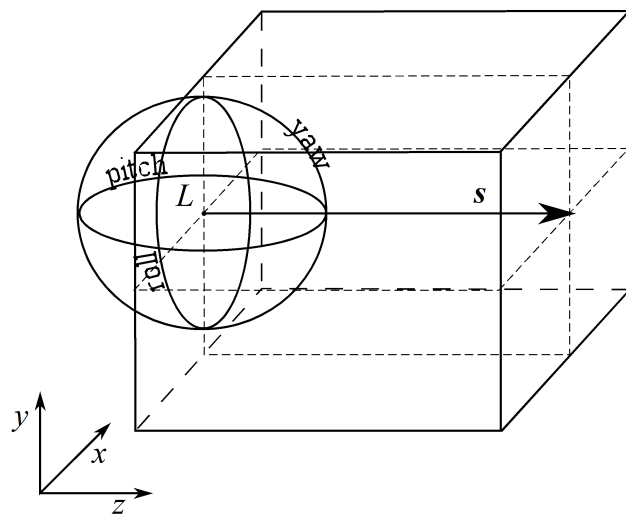
Ilustrace 20: Transformace světlometu do lokálního souřadného systému automobilu pomocí transformační matice světla, poté transformace samotného automobilu do globálního souřadného systému za pomoci transformační matice automobilu

Na ilustraci 20 lze vidět, jak z lokálního souřadného systému světla můžeme pomocí transformační matice světla přesunout světlo do přesné polohy světlometu na automobilu, a pak pomocí transformační matice automobilu můžeme dostat automobil s příslušným světlem do scény, tedy do globálního souřadného systému. Tyto transformační matice se mezi sebou násobí v určitém pořadí, které lze vidět z ilustrace 20.

Implementace světelných map

3.5.1 Eulerova transformace pro rotaci světlometů

Protože se světlometry mohou otáčet i kolem osy směru světla, který tento druh světla potřebuje oproti základním druhům světla, musela být použita transformační matice pro samotný popis světla ve scéně, samotný vektor nám neumožňuje zaznamenat tuto rotaci. Pro manipulaci s transformační maticí byla použita Eulerova transformace, na ilustraci 21 můžeme vidět tuto transformaci po jednotlivých složkách otáčení Eulerovi transformace, kde se pro příklad transformuje objemová textura, vyjadřující část osvětlení scény. [4]



Ilustrace 21: Názorná ukázka Eulerovy transformace na objemové textuře, kde jednotlivé složky rotace yaw, roll a pitch, jsou popsány na obrázku

Základní složky otočení a jejich odvozené názvy pro Eulerovu transformaci vidíme v následujícím seznamu:

- *yaw* - rotace kolem x souřadnice (heading, azimuth)
- *pitch* - rotace kolem y souřadnice (altitude, elevation)
- *roll* - rotace kolem z souřadnice (bank, tilt)

Tato transformace vytváří transformační matici M , která nám umožní transformovat světlo z lokálního souřadného systému do globálního souřadnicového systému. Tato matice je tvořena na základě vztahu (38)

Implementace světelných map

$$[M] = \begin{pmatrix} \cos(\textit{yaw}) * \cos(\textit{pitch}) & & & \\ -\cos(\textit{yaw}) * \sin(\textit{pitch}) * \cos(\textit{roll}) + \sin(\textit{yaw}) * \sin(\textit{roll}) & & & \\ \cos(\textit{yaw}) * \sin(\textit{pitch}) * \sin(\textit{roll}) + \sin(\textit{yaw}) * \cos(\textit{roll}) & & & \\ 0 & & & \end{pmatrix} \quad (38)$$

$$\begin{pmatrix} \sin(\textit{pitch}) & & & \\ \cos(\textit{pitch}) * \cos(\textit{roll}) & & & \\ \cos(\textit{pitch}) * \sin(\textit{roll}) & & & \\ 0 & & & \end{pmatrix}$$

$$\begin{pmatrix} \sin(\textit{yaw}) * \cos(\textit{pitch}) & & & \textit{pos}_x \\ \sin(\textit{yaw}) * \sin(\textit{pitch}) * \cos(\textit{roll}) + \cos(\textit{yaw}) * \sin(\textit{roll}) & & & \textit{pos}_y \\ -\sin(\textit{yaw}) * \sin(\textit{pitch}) * \sin(\textit{roll}) + \cos(\textit{yaw}) * \cos(\textit{roll}) & & & \textit{pos}_z \\ 0 & & & 1 \end{pmatrix}$$

Chceme-li hodnoty z transformační matice dostat do složek Eulerovy transformace použijeme vztahy (39), pro složku zvanou *pitch* .

$$\textit{pitch} = \arcsin(M_{12}) \quad (39)$$

Pro další dvě složky *yaw* a *roll* existuje podmínka, která rozhoduje o budoucím vypočtu. Když hodnota matice na index M_{12} nabývá hodnoty -1 nebo 1 , pak se musí složky vypočíst podle vztahů (40) a (41):

$$\textit{yaw} = \arctan\left(\frac{-M_{31}}{M_{33}}\right) \quad (40)$$

$$\textit{roll} = 0 \quad (41)$$

Když tato hodnota matice na indexu M_{12} nabývá jiných hodnot spočítají se příslušné složky pomocí vztahů (42) a (43).

$$\textit{yaw} = \arctan\left(\frac{-M_{13}}{M_{11}}\right) \quad (42)$$

Implementace světelných map

$$\text{roll} = \arctan\left(\frac{-M_{32}}{M_{22}}\right) \quad (43)$$

3.6 Převod barevné teploty na RGB barvu

Pro převod potřebujeme nejprve dostat intenzitu ozáření E_e ve viditelném světle elektromagnetického záření černého tělesa zahřátého na určitou barevnou teplotu T do barevného prostoru XYZ. Jako poslední krok musíme tyto hodnoty XYZ transformovat pomocí transformační matice získané z barevného profilu.

3.6.1 Převod barevné teploty do barevného prostoru XYZ

Barevná teplota popisuje spektrum bílého světla. Světlo určité barevné teploty odpovídá barvě tepelného záření vydávaného černým tělesem zahřátým na danou teplotu T . Barevná teplota se udává v kelvinech. Pro převod tedy potřebujeme znát intenzitu záření E_e absolutně černého tělesa při konkrétní teplotě T podle Planckova vyzařovacího zákona (44) můžeme zjistit intenzitu záření E_e v dané vlnové délce λ . [5]

$$E_e = \frac{h_r}{2 \Pi^2 c^2} \frac{\omega^3}{e^{\frac{h_r \omega}{kT}} - 1} \omega \quad (44)$$

Pro převod na úhlovou frekvenci, použijeme vztahu (1) pro převod na frekvenci f a pro získání uhlové frekvence použijeme vztahu (45).

$$\omega = 2 \Pi f \quad (45)$$

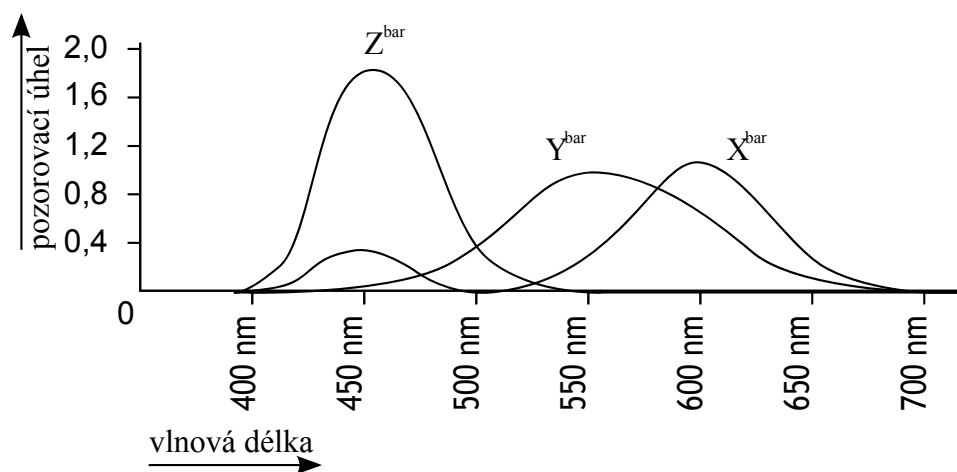
Konstanta h_r je redukována Planckova konstanta, která je vypočtena pomocí vztahu (46) a k Boltzmannova konstanta.

$$h_r = \frac{h}{2 \Pi} \text{ J.s}^{-1} \quad k = 1,380\,6488 \cdot 10^{-23} \text{ J.K}^{-1} \quad (46)$$

Implementace světelných map

Tento vztah (44) nám umožní vypočítat intenzitu ozáření E_e v námi definovaném části spektra. Pro praxi nás zajímá jen ta část spektra, která se nazývá viditelné světlo, jak už bylo zmíněno, tato část elektromagnetického spektra se pohybuje v $\langle 400,700 \rangle nm$.

Známe-li tuto intenzitu ozáření E_e v určité vlnové délce λ při určité teplotě T převedeme tuto hodnotu do barevného prostoru XYZ, tím že tuto hodnotu E_e vynásobíme s hodnotou na křivce X^{bar} , Y^{bar} a Z^{bar} popisující citlivostí lidského oko standardního pozorovatele, při určité vlnové délce λ . Tyto křivky reprezentují každou složku barevného prostoru XYZ, při určité vlnové délce, které jsou zobrazeny na ilustraci 22 pro 2° pozorovací úhel. [6]



Ilustrace 22: Křivky popisující citlivost lidského oko standardního pozorovatele na vlnovou délku elektromagnetického záření, která zveřejnila CIE roku 1931

Pomocí těchto vztahů vypočteme hodnoty v celém viditelném světle v elektromagnetickém spektru tedy $\langle 380,780 \rangle nm$ vždy po určitém kroku $\Delta \lambda$, protože pracujeme v diskrétním prostoru. Tyto hodnoty spočteme pro všechny tři složky XYZ stejně, podle vztahu (47).

$$X = \sum_{i=1}^N E_e(\lambda_{start} + i \Delta \lambda) X^{bar}(\lambda_{start} + i \Delta \lambda) \quad (47)$$

Pro úplnost musíme tyto hodnoty normalizovat pro rozložení v barevném prostoru

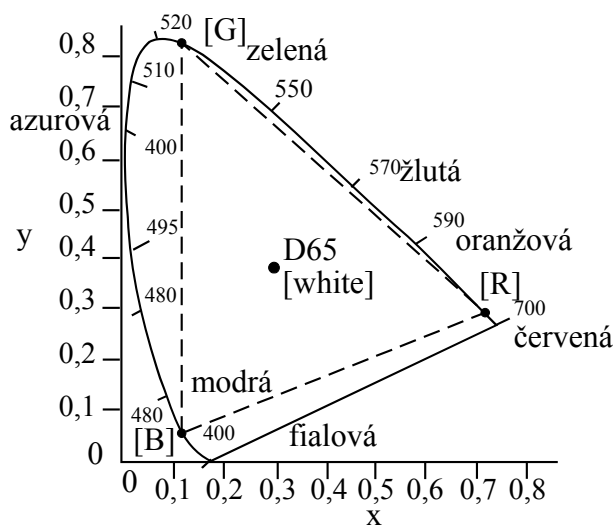
Implementace světelných map

XYZ podle vztahu (48).

$$X = \frac{X}{X+Y+Z} \quad Y = \frac{Y}{X+Y+Z} \quad Z = \frac{Z}{X+Y+Z} \quad (48)$$

3.6.2 Převod z barevného prostoru XYZ do barevného prostoru RGB

Protože jsme tyto složky normalizovali můžeme tedy odstranit jasovou složku Z , která se dá vypočítat zpětně ze zbývajících dvou složek tedy $Z = 1 - X - Y$ a vyjádřit barvu bez jasu pomocí dvou proměnných X a Y , které mohou definovat bod v rovině, jak lze vidět na ilustraci 23. Chceme-li zjištěné hodnoty XYZ transformovat do RGB barevného prostoru budeme potřebovat znát jakýkoliv barevný profil nesoucí v sobě informace o přesných polohách XYZ v chromatickém diagramu pro červenou X_R , Y_R a Z_R , modrou X_B , Y_B a Z_B , zelenou X_G , Y_G a Z_G barvu a polohu bílého bodu X_W , Y_W a Z_W . Tyto hodnoty barevného profilu korespondují s diagramem na ilustraci 23.



Ilustrace 23: Diagram vyjadřující barvu na rovině pomocí složek X a Y v barevném prostoru XYZ zvaný Chromaticity Diagram

Tyto hodnoty nám umožní vytvořit transformační matici M , pro převod mezi prostory XYZ a RGB za použití hodnot vybraného barevného profilu, podle vztahu (49). [6]

Implementace světelných map

$$[M_{\text{bez bílé}}] = \begin{bmatrix} R_X & R_Y & R_Z \\ G_X & G_Y & G_Z \\ B_X & B_Y & B_Z \end{bmatrix} = \begin{bmatrix} Y_G Z_B - Y_B Z_G & X_B Z_G - X_G Z_B & X_G Y_B - X_B Y_G \\ Y_B Z_R - Y_R Z_B & X_R Z_B - X_B Z_R & X_B Y_R - X_R Y_B \\ Y_R Z_G - Y_G Z_R & X_G Z_R - X_R Z_G & X_R Y_G - X_G Y_R \end{bmatrix} \quad (49)$$

Poté se tyto hodnoty matice M musí zkosit podle hodnot bílého bodu. Tato hodnota zkosení R_W , G_W a B_W se pro každou barvu se vypočte podle vztahu (50).

$$R_W = \frac{R_X X_W + R_Y Y_W + R_Z Z_W}{Y_W} \quad (50)$$

Poté se tato hodnota aplikuje na matici M a následně získáváme konečnou matici pro převod mezi těmito barevnými prostory XYZ a RGB.

$$[M] = \begin{bmatrix} \frac{R_X}{R_W} & \frac{R_Y}{R_W} & \frac{R_Z}{R_W} \\ \frac{G_X}{G_W} & \frac{G_Y}{G_W} & \frac{G_Z}{G_W} \\ \frac{B_X}{B_W} & \frac{B_Y}{B_W} & \frac{B_Z}{B_W} \end{bmatrix} \quad (51)$$

Chceme-li získat hodnoty barvy v barevném prostoru RGB, stačí vynásobit tuto matici se složkami XYZ, kterou jsme získali z převodu barevné teploty do XYZ prostoru.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [M] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (52)$$

Pro převod z RGB na XYZ použijeme inverzní matici $[M]^{-1}$

Implementace světelných map

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M]^{-1} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (53)$$

Vyhodnocení implementace

4 Vyhodnocení implementace

Pro zjištění rychlosti renderování určitých variant musela být použita méně výkonná grafická karta Intel HD 3000, protože při použití výkonnější grafické karty Nvidia 540M se renderování snímku nekleslo 60 snímku za sekundu (fps) v testovací scéně. Ve scéně bylo až 8 světel a 4 různé druhy světelných profilů IES, kde jedno světlo rotovalo kolem osy Y v Eulerovi transformaci se jedná a složku pitch.

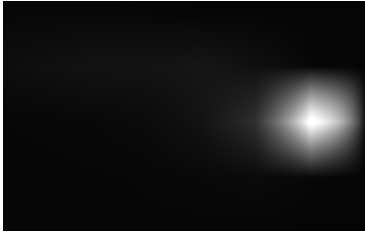

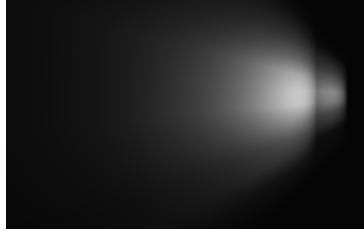
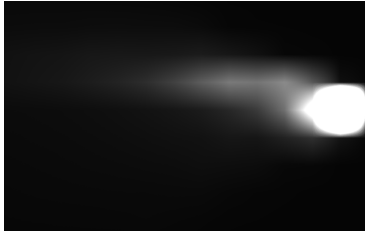
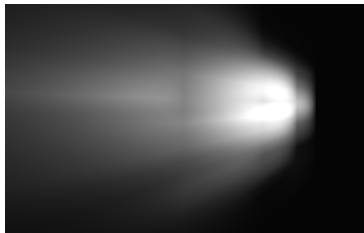
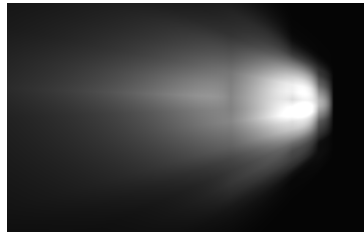






4.1 Rychlost vykreslování při určitých variantách

Metoda hledání koordinátů [počet světel ve scéně, počet používaných světelných profilů]	Hledání koordinátů v 3D textuře popsané vztahem (30)	Hledání koordinátů v 2D textuře popsané vztahy (18) a (33)	Hledání koordinátů v 2D textuře popsané vztahy(19) a (34)	Hledání koordinátů v 2D textuře popsané vztahem (37)
[2,4]	11,6fps	5,3fps	5,7fps	5,4fps
[2,8]	7,5fps	3,7fps	3,3fps	3,3fps
[4,4]	11,3fps	5,2fps	5,6fps	5,3fps
[4,8]	6,5fps	2,9fps	3,3fps	2,9fps







Těmito výsledky bylo zjištěno že, vyhledávání koordinátů v objemové textuře je rychlejší než vyhledávání v dvourozměrné textuře. Při změnách rozlišení textur se počet snímku za sekundu nemění.

Na těchto vizuálních hodnotách v tabulce můžeme vidět kvalitu zobrazování osvětlení vůči rozlišení textury a tedy její velikost vzhledem ke kvalitě zobrazení světelného profilu.

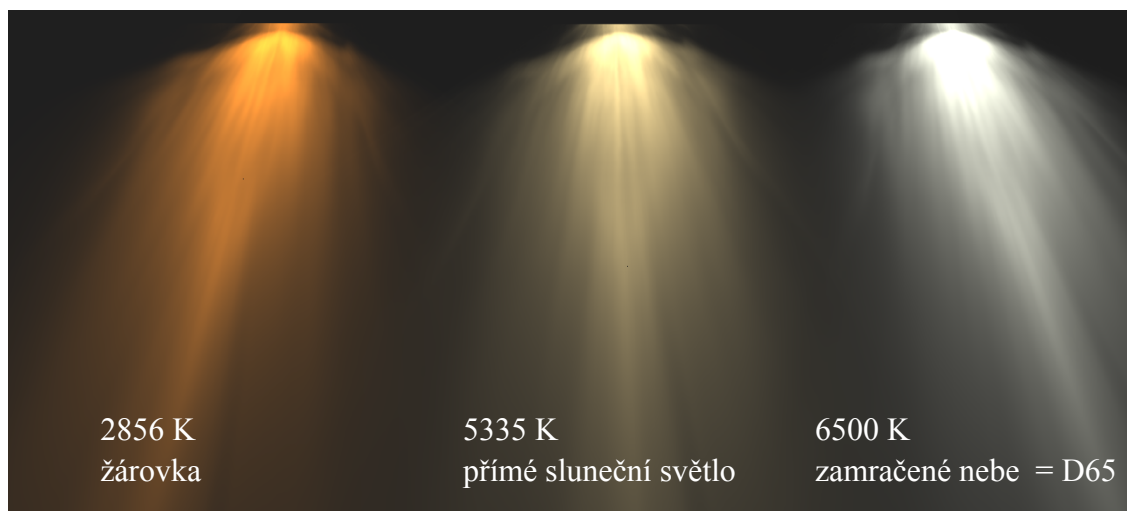
Vyhodnocení implementace

3D (30)	2D (18) a (33) (19) a (34)	2Db (37)
 <p data-bbox="384 703 676 741">< 50x50x50 > 0,0065 s</p>	 <p data-bbox="767 703 1050 741">< 50x50 > 0,005102 s</p>	
 <p data-bbox="368 1014 708 1052">< 100x100x100 > 0,2843 s</p>	 <p data-bbox="756 1014 1064 1052">< 100x100 > 0,005707 s</p>	
 <p data-bbox="368 1326 708 1364">< 200x200x200 > 2,1570 s</p>	 <p data-bbox="756 1326 1064 1364">< 200x200 > 0,007435 s</p>	
 <p data-bbox="368 1637 724 1675">< 400x400x400 > 19,9420 s</p>	 <p data-bbox="756 1637 1064 1675">< 400x400 > 0.027054s</p>	

Vyhodnocení implementace

		
< 400x400x800 > 42,3226 s	< 800x800 > 0,051708 s	
		
< 400,400,1000 > 51,9598 s	< 1600x1600 > 0,173425 s	

Na ilustraci 24 jde vidět osvětlení v určitých barevných teplotách, při použití barevného profilu IES.



Ilustrace 24: Osvětlení plochy při určité hodnotě barevné teploty udávané černým tělesem pro převod mezi XYZ a RGB byl použit barevný profil CIE

Závěr

5 Závěr

Cílem této práce bylo z naměřených hodnot svítivosti I_v uložených v IES profilu zobrazit osvětlení E_v dané plochy, při určitých polohách světelných map ve scéně v reálném čase a umožnit změnu osvětlení podle barevné teploty černého tělesa.

Všechny stanovené cíle byly splněny, jak samotné vykreslování, s použitou zobrazovací metodou raytracing, tak bylo následně použito standartu víceplatformního rozhraní přístupu ke grafickým kartám, využívané při tvorbu grafických aplikací OpenGL s použitím shaderů. Tady došlo k mnoha modifikacím vstupních dat samotného IES profilu a vyhledávání v těchto datech. Pro transformaci světla v prostoru mezi lokálním souřadným systémem a globálním souřadným systémem bylo použito transformačních matic. Pro samotný směr světla a polohu, byla taktéž použita transformační matice. Pro změnu osvětlení podle barevné teploty černého tělesa musela být hodnota prvně převedena do barevného prostoru XYZ a potom transformována do barevného prostoru RGB podle vybraného barevného profilu.

Z výsledků jsem se dozvěděl, že nejrychlejší metoda pro zjišťování osvětlení dané plochy je z objemové textury. Nevýhoda této metody je extrémní paměťová náročnost z důvodů vykreslení detailnějších světelných profilů. Proto se jako vhodnější varianta jeví použití dvourozměrné textury, kde je sice výpočet osvětlení výpočetně složitější, ale při malé paměťové náročnosti dokáže vykreslit osvětlení plochy ve vysokých detailech konkrétního světelného profilu.

Seznam použité literatury

6 Seznam použité literatury

- 1: Lepil, O., *Fyzika pro gymnázia Elektřina a magnetismus*. Prométheus Praha, 1998. 206 s. ISBN 978-80-71963-84-4
- 3: Illuminating Engineering Society, *IES lighting handbook 10th Edition: the standard lighting guide*. New York,; McGraw-Hill, 1947. 1328 s. ISBN 978-08-79952-41-9
- 4: Sojka, E., *Počítačová grafika II: Metody a nástroje zobrazování 3D scén.*. Ostrava: VŠB TUO, 2003. 143 s. ISBN 802-48-02-93-7
- 5: Prof.Ing.Václav Skala, CSc., *Algoritmy počítačové grafiky II*. Vaclav Skala – UNION Agency, 2011. 165 s. ISBN 978-80-86943-20-6
- 6: Z. Horák, F. Krupka, *Fyzika, 3. vydání.*. SNTL / Alfa, Praha, 1981. 132 s. ISBN 978-80-86943-21-3
- 7: Prof.Ing.Václav Skala, CSc., *Algoritmy počítačové grafiky III*. Vaclav Skala – UNION Agency, 2011. 165 s. ISBN 978-80-86943-21-3
- 7: Bumbálek, J., *Základy technické optiky*. ČVUT Praha, 1987. 227 s. ISBN 978-80-01013-52-6
- 8: Prof.Ing.Václav Skala, CSc., *Algoritmy počítačové grafiky I*. Vaclav Skala – UNION Agency, 2011. 165 s. ISBN 978-80-86943-19-0
- 9: Beiser Arthur, *Úvod do moderní fyziky*. Academia, Praha, 1978. 201 s. ISBN 978-80-86943-20-6
- 10: Jan Kopečný, *ZÁKLADY FYZIKY Modul 4 – Optika a atomové jádro*. Ostrava: VŠB TUO, 2008. 97 s. ISBN 978-80-24800-21-9
- 11: WOLF, David, *OpenGL 4.0 Shading Language Cookbook*. Packt Publishing Ltd, 2011, 2011. 323 s. ISBN 978-18-49514-76-7
- 12: Shreiner, Dave, Sellers, Graham, Kessenich, John M., *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3 (8th Edition) 8th (eighth) edition*. Lincea-K published by Addison-Wesley Professional, 2013. 616 s. ISBN 978-0321773036

Seznam příloh

7 Seznam příloh

Příloha A Obrázky generované programem

1

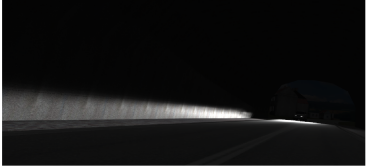











Součástí DP je DVD.

Adresářová struktura přiloženého DVD:

- | | |
|--|-----------------|
| 1. zdrojové kódy programu | ./Zdrojove kody |
| 2. obrázky vygenerované programem | ./Obrázky |
| 3. Vektorové obrázky z diplomové práce | ./Obrázky |
| 4. text diplomové práce | ./Text |

Seznam příloh

Obrázky generované programem

		
		
		
	 <p>2856 K žárovka 5335 K přímé sluneční světlo 6500 K Zamračené nebe = D65</p>	 <p>2856 K žárovka 5335 K přímé sluneční světlo 6500 K Zamračené nebe = D65</p>