

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce
Simulace zatmění měsíce

Adam Pospíšil

Vedoucí práce: Ing. Jaroslav Sloup

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

27. května 2011

Poděkování

Zde bych rád poděkoval svému vedoucímu, Ing. Jaroslavu Sloupovi, za to, že měl se mnou trpělivost a vždy si našel čas, když jsem potřeboval poradit.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 26.5.2011

.....

Abstract

The goal of this work is to create realistic picture of the Moon during a lunar eclipse. The method described here calculates both direct and indirect illumination resulting from the refraction in the Earth's atmosphere. Other simulated physical processes are the scattering of light in the air and the absorption by stratospheric aerosols. Direct and indirect light will be simulated by ray tracing and photon mapping respectively. We will use tone mapping for rendering an image with high dynamic range. Moon's surface will be simulated by mipmaps constructed from satellite images.

The resulting image is then compared with digital photograph of lunar eclipse.

Abstrakt

Cílem této práce je vytvoření věrohodného obrazu Měsíce během měsíčního zatmění. Metoda zde popsána počítá jak přímé osvětlení, tak osvětlení nepřímé, vzniklé lomem světla v zemské atmosféře. Další simulované optické jevy jsou rozptyl světla v atmosféře a absorpce světla vzdušnými aerosoly. Nepřímé osvětlení bude simulováno pomocí techniky fotonových map, přímé osvětlení potom technikou sledování paprsků. Pro vykreslení obrazu s vysokým dynamickým rozsahem použijeme mapování tónů. K simulaci povrchu Měsíce budou sloužit mipmapy vytvořené ze satelitních snímků.

Výsledek je pak porovnán s fotografií skutečného zatmění.

Obsah

1	Úvod	1
2	Fyzikální aspekt úlohy	3
2.1	Zatmění	3
2.2	Poloha vesmírných těles	4
2.2.1	Čas	4
2.2.2	Slunce	4
2.2.3	Měsíc	5
2.3	Zemská atmosféra	5
2.3.1	Lom světla v nehomogenním médiu	5
2.3.2	Rozptyl světla	6
2.4	Měsíc	6
2.4.1	BRDF Měsíce	6
2.4.2	Měsíční albedo	7
3	Analýza a návrh řešení	11
3.1	Dosavadní práce	11
3.2	Model sluneční soustavy	11
3.3	Slunce	12
3.4	Atmosféra	12
3.4.1	Lom světla	13
3.4.2	Rozptyl světla	13
3.4.3	Pohlcení světla	14
3.5	Osvětlení Měsíce	15
3.5.1	Nepřímé osvětlení	15
3.5.2	Přímé osvětlení	16
3.5.3	Převod barev	16
3.5.4	Tone mapping	18
3.6	Návrh aplikace	18
4	Realizace	21
4.1	Nástroje	21
4.2	Aplikační logika	21
4.2.1	Nastavení pozic	21
4.2.2	Vystřelování fotonů	22

4.2.3	Vykreslení výsledku	22
4.3	Textura Měsíce	22
4.4	Popis jednotlivých objektů	23
4.4.1	Třída Tracer	23
4.4.1.1	Vytváření fotonů	24
4.4.1.2	Sledování fotonů	25
4.4.2	Třídy PhotonMap, Photon a Node	26
4.4.2.1	Photon	27
4.4.2.2	PhotonMap	27
4.4.2.3	Node	27
4.4.3	Třída Renderer	29
5	Výsledky	31
5.1	Měření	32
6	Závěr	39
6.1	Budoucí rozšíření	39
A	Instalační a uživatelská příručka	43
B	Obsah přiloženého CD	45

Seznam obrázků

2.1	Vzájemná poloha vesmírných těles při měsíčním zatmění.	8
2.2	Lom světla v zemské atmosféře	8
2.3	Danjonova stupnice	9
2.4	Ekliptika	9
3.1	Sluneční spektrum	12
3.2	Rozdělení úlohy na výpočet přímého a nepřímého osvětlení.	15
3.3	Fotony na povrchu	16
3.4	Vliv Gaussovského filtru na osvětlení jednotlivých fotonů.	17
3.5	Vliv poloměru prohledávané oblasti, každá scéna obsahuje tisíc fotonů.	17
3.6	Stejná scéna pro různé hodnoty koeficientu použitého v tone mappingu.	18
4.1	Textura Měsíce	23
4.2	Mipmapy	23
5.1	Scény, na kterých bylo provedeno měření rychlosti.	33
5.2	Částečné zatmění, ke kterému došlo 7. září 2006 18:51 UTC.	34
5.3	Částečné zatmění, ke kterému došlo 26. června 2010 11:31 UTC.	34
5.4	Téměř úplné zatmění, ke kterému došlo 16. srpna 2008 21:10 UTC.	35
5.5	Téměř úplné zatmění, ke kterému došlo 24. března 1997 4:45 UTC.	35
5.6	Průběh úplného zatmění 21. února 2008.	36
5.7	Průběh částečného zatmění 26. června 2010.	37

Seznam tabulek

5.1	Doba sledování fotonů a vykreslení pro první scénu (s).	33
5.2	Doba sledování fotonů a vykreslení pro druhou scénu (s).	33

Kapitola 1

Úvod

Lidé pozorovali měsíční zatmění od pradávna. Nejstarší historicky doložené pozorování měsíčního zatmění se odehrálo ve dvanáctém století př.n.l. v Číně. Zatmělý Měsíc a jeho temně rudá barva měla být zlým znamením. Jeden příklad za všechny – krvavý měsíc, který zasvítl 22. května 1453 během obléhání Konstantinopole[8], měl věstít brzký pád města do rukou Osmanských Turků, ke kterému došlo o sedm dní později.

Přestože podstata samotného jevu není nikterak složitá a k jeho pochopení stačí základní povědomí o astronomii, na výsledný vzhled má vliv mnoho dalších faktorů. Dá se tak říci, že žádná dvě zatmění nejsou stejná. Jelikož k zatměním dochází nepravidelně a mezi dvěma úkazy může být i roční interval, má smysl tento přírodní jev simulovat pomocí počítačové grafiky.

Počítačová grafika je moderní a neustále se rozvíjející obor, jehož část se zaměřuje na stále realističtější vyobrazení skutečného světa. Od třpytu světla v kapce vody po obrazy galaxií v dalekém vesmíru – to všechno lze s použitím dnešní techniky věrohodně simulovat. A právě do této oblasti patří tato úloha.

Cílem této práce je navržení a implementace vhodného modelu pro vytváření obrazů měsíčních zatmění. Tohoto chceme dosáhnout pomocí simulace jednotlivých fyzikálních jevů - lomu, rozptylu a absorpce světla. Pro výpočet nepřímého osvětlení, vzniklého vlivem těchto jevů při průchodu světla zemskou atmosférou, použijeme fotonových map. Pro přímé osvětlení, figurující v částečných zatměních, je použito metody sledování paprsku (ray tracing). Tato osvětlení budou počítána po jednotlivých vlnových délkách slunečního spektra a posléze převedena do hodnot zobrazitelných na obrazovce pomocí mapování tónů. Krátery a další nerovnosti na povrchu měsíce budeme simulovat použitím satelitního snímku Měsíce jako textury.

V druhé kapitole bude čtenář seznámen s fyzikálním aspektem úlohy včetně popisu jednotlivých vesmírných těles, stejně jako optických jevů, ke kterým při zatmění dochází. Ve třetí kapitole je popsáno vlastní řešení úlohy včetně již dříve vytvořených metod. Jsou zde prezentovány matematické modely pro simulaci jednotlivých částí úlohy. Taktéž jsou v této kapitole popsány metody použité pro převedení matematické reprezentace do výsledného obrazu. Cílem čtvrté kapitoly je přiblížit čtenáři implementaci úlohy pomocí popisu vlastní aplikační logiky a jednotlivých objektů včetně ukázek zdrojového kódu. Obsahem páté kapitoly jsou výsledky práce pro různé vstupní parametry.

Kapitola 2

Fyzikální aspekt úlohy

Tato kapitola obsahuje obecný popis jevů a objektů fyzického světa a jejich vzájemných souvislostí. Popisujeme zde podstatu samotného astronomického jevu zatmění. Dále předkládáme způsob, jakým je v astronomii měřen čas a poloha vesmírných těles, stejně jako vzorce používané pro jejich výpočet. Taktéž jsou v této kapitole popsány fyzikální jevy ovlivňující dráhu a intenzitu světla procházejícího zemskou atmosférou. Na závěr je zde pak popsán způsob, jakým se světlo odráží od Měsíce, včetně modelu pro výpočet odrazivosti měsíčního prachu.

2.1 Zatmění

Zatmění je astronomický jev, při kterém je pozorovaný objekt částečně nebo úplně zakryt jiným vesmírným tělesem. Zatmění Měsíce nastává tehdy, když se Slunce, Země a Měsíc nachází v zákrytu a stín vržený Zemí dopadá na povrch Měsíce. Jelikož k měsíčnímu zatmění dochází právě když jsou Měsíc a Slunce v opozici, Měsíc je při něm vždy v úplňku. Vzhledem ke sklonu oběžných drah Země a Měsíce k němu dochází nepravidelně, ne však nijak vzácně. Průměrně k němu dojde dvakrát až třikrát do roka.

Podle dráhy, kterou Měsíc prochází ve stínu Země, rozlišujeme několik typů měsíčního zatmění (obr. 2.1). K polostínovému (penumbrálnímu) zatmění dochází, když Měsíc, jak název napovídá, vstoupí pouze do zemského polostínu. To se projeví mírným ztmavením měsíčního disku. Částečné zatmění nastává, pokud část Měsíce vstoupí do plného stínu (umbry) Země. Pokud se v něm nachází celý Měsíc, nazýváme takové zatmění zatměním úplným (totálním).

Může být poněkud zarážející, že na povrch zatmělého Měsíce stále dopadá světlo. Tato červená barva (obr. 2.3), typická pro každé částečné nebo úplné zatmění, je způsobena lomem a rozptylem světla v zemské atmosféře. Důsledkem lomu je totiž ohyb světla na místa úplně zakrytá Zemí (obr. 2.2). Rozptyl světla pak fakticky tlumí světlo krátkých vlnových délek (ty se rozptylují v atmosféře a způsobují například modrou barvu oblohy), výsledkem je pak pozorovaná rezavá barva měsíčního povrchu.

Na konečný vzhled zatmění má vliv ale i například koncentrace prachu ve vyšších vrstvách zemské atmosféry.

Všechny tyto podmínky způsobují vysokou proměnlivost v konečném vzhledu zatmění. Tato proměnlivost je hodnocena pomocí tzv. Danjonovy stupnice [5] (obr. 2.3).

Danjonova stupnice je pětibodová:

- L = 0 Velmi tmavé zatmění, Měsíc je stěží viditelný, obzvlášť ve středu totality.
- L = 1 Tmavé zatmění šedé až hnědé barvy, detaily na disku jsou stěží viditelné.
- L = 2 Tmavě rudé nebo rezavé zatmění, s tmavou oblastí v centru stínu a světlejšími okraji.
- L = 3 Cihlově červené zatmění s jasně žlutými okraji.
- L = 4 Oranžové nebo bronzové, velmi jasné zatmění s namodralými okraji.

2.2 Poloha vesmírných těles

Odvětví astronomie zabývající se vzájemnou polohou vesmírných těles se nazývá sférická astronomie. Ta je založena na promítnutí vesmírných těles na nekonečně vzdálenou nebeskou sféru, s pozorovatelem (nejčastěji Zemí) v jejím středu. Ve sférické astronomii se nabízí mnoho možných souřadných systémů, ale pro popis těles ve sluneční soustavě je nejvhodnější ekliptický souřadný systém (obr. 2.4) - sférický systém, kde je ekliptika (oběžná dráha Země okolo Slunce) základní rovinou. Pozice každého tělesa je tak popsána jednak dvojicí úhlových souřadnic - ekliptickou šířkou (značenou β) a délkou (λ), a vzdáleností od pozorovatele.

2.2.1 Čas

Astronomie používá pro označení času tzv. juliánský den – počet dní, které uplynuly od 1. ledna 4713 př.n.l. 12:00 UTC. Pro data (D.M.Y) a čas (h:m:s) po zavedení gregoriánského kalendáře (po roce 1582 n.l.) se juliánský den spočítá takto:

$$JD = 1720996.5 - \lfloor Y'/100 \rfloor + \lfloor Y'/400 \rfloor + \lfloor 365, 25Y' \rfloor + \lfloor 30, 6001(M' + 1) \rfloor + D + \\ + (h + (m + s/60)/60)/24$$

Pro Y' a M' platí, že když M je 1 nebo 2, tak $Y' = Y - 1$ a $M' = M + 12$, jinak $Y' = Y$ a $M' = M$.

Časové pásmo výsledného času je GMT a proměnná T používaná v následujících vzorcích v této kapitole je následující: $T = (JD - 2451545, 0)/36525$.

2.2.2 Slunce

Ekliptická šířka Slunce β je z definice nulová. Ekliptickou délku λ a vzdálenost od středu Země r lze zjistit následovně ($M = 6, 24 + 628, 302T$) [12]:

$$\begin{aligned} \lambda &= 4, 895048 + 628, 331951T + (0, 033417 - 0, 000084T) \sin M + 0, 000351 \sin 2M, \\ r &= 1, 000140 - (0, 016708 - 0, 000042T) \cos M - 0, 000141 \cos 2M, \end{aligned}$$

kde r je udáno v astronomických jednotkách ($1\text{AU} = 1, 496 \times 10^{11}\text{m}$).

2.2.3 Měsíc

Ekliptické souřadnice Měsíce lze spočítat s pomocí těchto vzorců [12]:

$$\begin{aligned} l' &= 3,8104 + 8399,7091T \\ m' &= 2,3554 + 8328,6911T \\ m &= 6,2300 + 628,3019T \\ d &= 5,1985 + 7771,3772T \\ f &= 1,6280 + 8433,4663T \end{aligned}$$

$$\begin{aligned} \lambda = & l' + 0,1098 \sin(m') + 0,0222 \sin(2d - m') + 0,0115 \sin(2d) + 0,0037 \sin(2m') \\ & - 0,0032 \sin(m) - 0,0020 \sin(2f) + 0,0010 \sin(2d - 2m') + 0,0010 \sin(2d - m - m') \\ & + 0,0009 \sin(2d + m') + 0,0008 \sin(2d - m) + 0,0007 \sin(m' - m) - 0,0006 \sin(d) \\ & - 0,0005 \sin(m + m') \end{aligned}$$

$$\begin{aligned} \beta = & 0,0895 \sin(f) + 0,0049 \sin(m' + f) + 0,0048 \sin(m' - f) + 0,0030 \sin(2d - f) \\ & + 0,0010 \sin(2d + f - m') + 0,0008 \sin(2d - f - m') + 0,0006 \sin(2d + f) \end{aligned}$$

$$\begin{aligned} \pi' = & 0,016593 + 0,000904 \cos(m') + 0,000166 \cos(2d - m') + 0,000137 \cos(2d) \\ & + 0,000049 \cos(2m') + 0,000015 \cos(2d + m') + 0,000009 \cos(2d - m) \end{aligned}$$

vzdálenost Měsíce od Země je $d = 1/\pi'$ v jednotkách zemských poloměrů.

2.3 Zemská atmosféra

Plynný obal okolo Země má na konečný vzhled měsíčního zatmění důležitý vliv. Bez něj by totiž na měsíční povrch ve stínu Země nedopadalo žádné sluneční světlo a Měsíc by nebyl pozorovatelný. Ovšem při cestě slunečního paprsku zemskou atmosférou se mění index lomu, světlo se láme, a výsledkem je dopad světla i na místa, kde je Slunce úplně zakryto Zemí (obr. 2.2).

2.3.1 Lom světla v nehomogenním médiu

Cesta světelného paprsku procházejícího médiem, jehož index lomu se spojitě mění, se zakřivuje. Dráha světla je pak popsána touto diferenciální rovnicí [10]:

$$\frac{d}{dl} \left(n \frac{dx_j}{dl} \right) - \frac{\delta n}{\delta x_j} = 0, j \in \{0, 1, 2\} \quad (2.1)$$

kde l je délka oblouku, n je index lomu a x_j jednotlivé souřadnice bodů na dráze.

2.3.2 Rozptyl světla

Kromě lomu dochází při průchodu světla atmosférou také k rozptylu. Tedy k vychylování fotonů z jejich původních drah srážkami s částicemi v nosném médiu. Pro částice s velikostí výrazně nižší než je vlnová délka světla (molekuly plynů) se používá Rayleighův rozptyl. Pro ostatní částice (vodní pára a prach) platí Mieova teorie [14].

Rozptyl světla není jediným jevem, který snižuje intenzitu paprsků cestou skrz atmosféru. Dochází zde totiž navíc k pohlcování světla částicemi prachu v atmosféře. Například v období po zvýšené sopečné aktivitě, ve kterém je koncentrace prachu a popela v atmosféře velmi vysoká, můžeme očekávat celkově nižší úroveň jasu zatmělého Měsíce, způsobenou útlumem světla při průchodu skrz tento aerosol [14].

2.4 Měsíc

Měsíc je jedinou přirozenou družicí Země, jeho rovníkový poloměr je 1738,14 km (cca $\frac{1}{4}$ zemského). Měsíc obíhá Zemi v průměrné vzdálenosti $3,84 \times 10^5$ km s dobou oběhu 27,32 dnů. Rotace měsíce je vůči Zemi synchronní - tzn. že jedna strana Měsíce je stále přivrácena k Zemi.

2.4.1 BRDF Měsíce

Povrch Měsíce se v některých ohledech chová velmi neobvykle, Měsíc totiž dosahuje stejné úrovně jasu na všech pozorovatelných místech - měsíční disk nevykazuje žádné ztmavení směrem ke kraji. Tento jev je způsoben silnou vrstvou prachu na povrchu Měsíce. Pro správné popsání vztahu mezi dopadajícím a odraženým světlem je toto nutné zohlednit v BRDF¹ Měsíce.

BRDF [2] je čtyřrozměrnou funkcí, vyjadřující kolik světla se odrazí od daného materiálu do oka pozorovatele v závislosti na úhlu dopadajícího světla a úhlu pozorování. Jensen et al. potom nabízejí následující postup pro její výpočet [12]:

$$f(\theta_i, \theta_r, \varphi) = \frac{2}{3\pi} B(\varphi, g) S(\varphi) \frac{1}{1 + \cos \theta_r / \cos \theta_i} \quad (2.2)$$

θ_i je úhel svíraný normálou povrchu s dopadajícím světlem, θ_r je úhel mezi normálou a směrem pozorování a φ mezi směrem pozorování a dopadajícím světlem. B je zpětnou a S rozptylovou funkcí.

$$B(\varphi, g) = \begin{cases} 2 - \frac{\tan \varphi}{2g} \left(1 - e^{-\frac{g}{\tan \varphi}}\right) \left(3 - e^{-\frac{g}{\tan \varphi}}\right), & \varphi < \frac{\pi}{2} \\ B(\varphi, g) = 1, & \varphi \geq \frac{\pi}{2} \end{cases}$$

$$S(\varphi) = \frac{\sin |\varphi| + (\pi - |\varphi|) \cos |\varphi|}{\pi} + t \left(1 - \frac{1}{2} \cos |\varphi|\right)^2$$

kde $g = 0,6$ a $t = 0,1$.

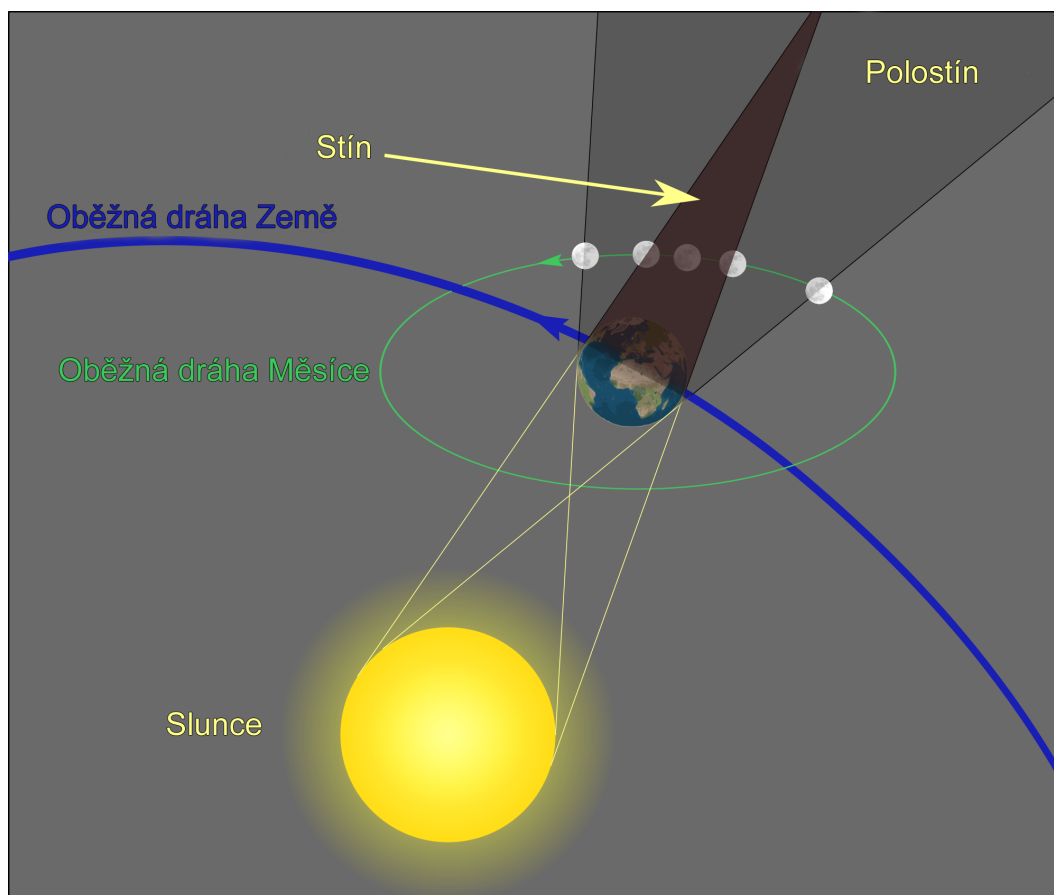
¹bidirectional reflectance distribution function

2.4.2 Měsíční albedo

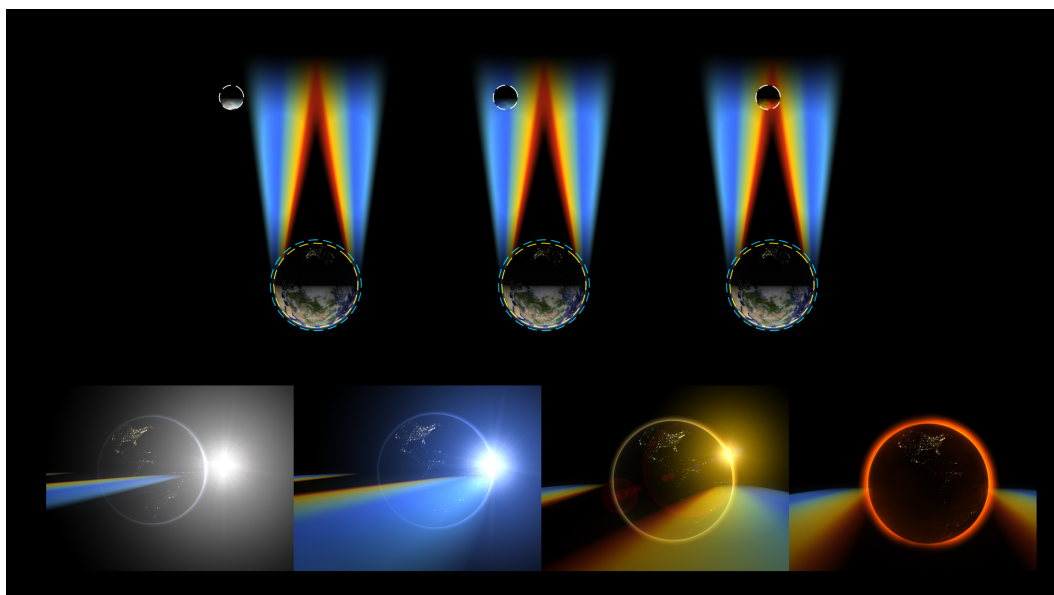
Albedo, neboli koeficient odrazivosti, udává poměr mezi množstvím odraženého světla ku dopadajícímu. Tento koeficient závisí na vlnové délce záření. Albedo Měsíce je téměř dvojnásobné pro červené světlo ($\lambda = 740$ nm) než pro světlo fialové ($\lambda = 340$ nm). To je způsobeno velikostí prášných částic na povrchu Měsíce, které jsou větší než vlnové délky viditelného světla (Mieova teorie rozptylu). Funkce aproximující odrazivost Měsíce k_r , odvozená z hodnot naměřených ze vzorků měsíčního prachu, vypadá následovně [14]:

$$k_r(\lambda) = -0,27063 + 2,6175 \times 10^{-3}\lambda - 1,2673 \times 10^{-6}\lambda^2 \quad (2.3)$$

Polynom je normalizovaný tak, aby byl pro vlnovou délku 780 nm roven 1,0.



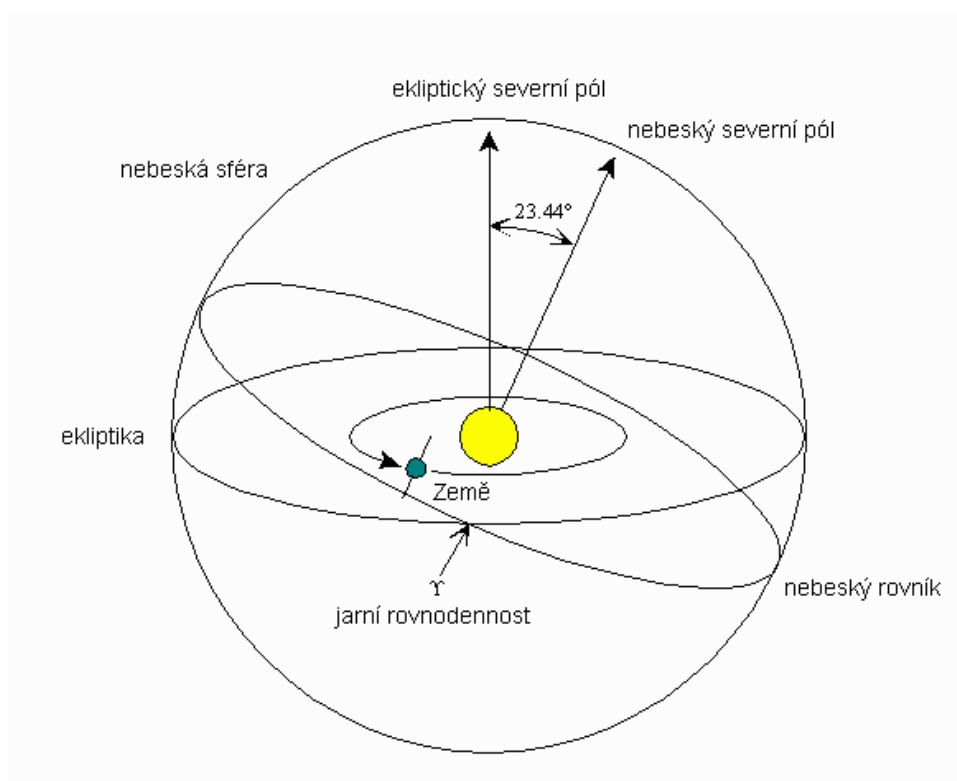
Obrázek 2.1: Vzájemná poloha vesmírných těles při měsíčním zatmění [3].



Obrázek 2.2: Lom světla v zemské atmosféře [3].



Obrázek 2.3: Různě tmavá zatmění, obrázek vlevo má hodnotu na Danjonově stupnici $L=2$, obrázek vpravo $L=4$, časy značí délku expozice snímku [3].



Obrázek 2.4: Vztah mezi ekliptickými a rovníkovými souřadnicemi.

Kapitola 3

Analýza a návrh řešení

Tato kapitola se zabývá popisem jednotlivých postupů simulujících jevy a objekty popsané v předchozí kapitole. Ukážeme zde jak simulovat světlo po jednotlivých spektrálních složkách a jak aproximovat průchod světelných paprsků zemskou atmosférou. Je zde popsán model simulující jednotlivé fyzikální jevy a jejich vliv na světlo opouštějící atmosféru. Kapitola také obsahuje popis jednotlivých technik použitých pro výpočet konečného osvětlení. Jmenovitě tedy fotonové mapy pro nepřímé osvětlení a sledování paprsků pro přímé osvětlení. Závěrem pak nastíníme, jak se všechny doposud popsané části úlohy spojí do výsledného řešení.

3.1 Dosavadní práce

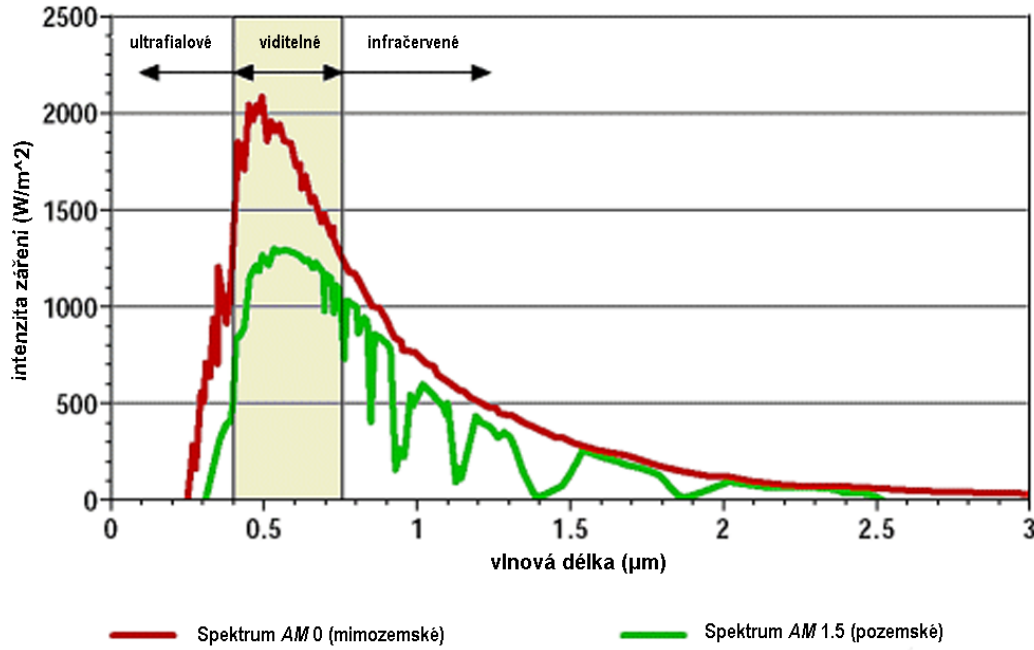
Problémem zobrazování měsíčních zatmění se zatím detailně zabývá pouze [14]. Další prací, která s problematikou souvisí volněji, je [12]. Tato práce se zabývá simulováním noční oblohy a některé postupy v této práci lze využít i pro problém měsíčního zatmění. V prvním článku autoři předkládají metody pro simulaci optických jevů v atmosféře. Ve druhé práci je popsán postup pro výpočet vzájemných poloh vesmírných těles, stejně jako popis chování odrazu světla od Měsíce.

3.2 Model sluneční soustavy

Na to, aby nastalo měsíční zatmění, má vliv pouze vzájemná poloha Slunce, Země a Měsíce. Pro naše potřeby můžeme proto zjednodušit sluneční soustavu pouze na soustavu těchto tří těles. Slunce, Země i Měsíc jsou modelovány jako koule o jejich tabulkových průměrných poloměrech (ve skutečnosti nejsou vesmírná tělesa dokonalé koule, vlivem otáčení okolo vlastní osy je jejich rovníkový poloměr nepatrně větší než na pólech). Konkrétní hodnoty jsou pak $6,955 \times 10^8$ m pro Slunce, $6,378 \times 10^6$ m pro Zemi a $1,737 \times 10^6$ m pro Měsíc. Dodávám, že nerovnosti na povrchu Země nemají na výsledné osvětlení Měsíce takřka žádný vliv a můžeme je proto zanedbat. Postup pro výpočet jejich vzájemných pozic je uveden v druhé kapitole.

3.3 Slunce

Pro tuto úlohu stačí simulovat sluneční záření pouze v rozmezí viditelného světla, tedy vlnové délky mezi 380 a 780 nanometry. Toto rozmezí je pak rovnoměrně navzorkováno, Yapo et al. [14] používají 200 vzorků (krok vzorku jsou tedy 2 nm), v této práci jsme zvolili jako krok 5 nm. Intenzita světla pro každou vlnovou délku (obr. 3.1) je odvozena z *ASTM E490 Solar Spectral Irradiance Tables* [4]. Tyto hodnoty modelují vesmírné sluneční spektrum (odlišné od slunečního spektra v zemské atmosféře).



Obrázek 3.1: Pozemské a mimozemské sluneční spektrum [6].

3.4 Atmosféra

Pro hodnoty fyzikálních veličin v zemské atmosféře používáme stejných zdrojů jako Yapo et al. [14] Jmenovitě tedy *1976 US Standard Atmosphere* [1]. Na úrovni mořské hladiny je pak teplota 15 stupňů Celsia a tlak 101,3 kPa. Teplota v libovolné výšce se spočítá lineární interpolací tabulkových hodnot, atmosférický tlak pak vzorcem:

$$p(h) = p_0 \left(1 - \frac{L \times h}{T_0} \right)^{\frac{g \times M}{R \times L}} \quad (3.1)$$

kde h je výška nad povrchem Země, p_0 a T_0 jsou tlak a teplota vzduchu na mořské hladině, g je gravitační zrychlení $g = 9,80665 \text{ ms}^{-2}$, M je molární hmotnost suchého vzduchu $M = 0,0289644 \text{ kg/mol}$, R je univerzální plynová konstanta $R = 8,31447 \frac{\text{J}}{\text{molK}}$ a L je rychlost poklesu teploty $L = 0,0065 \text{ K/m}$. Teplota a tlak v dané výšce jsou důležité pro výpočet indexu lomu světla a objemové hustoty částic - veličin klíčových pro lom a rozptyl světla.

Zobrazení některých atmosférických jevů vyžaduje zavést do některých míst atmosféry změny, to ale pro měsíční zatmění není potřeba a zemskou atmosféru stačí modelovat rovnoměrně, bez lokálních extrémů.

Atmosféra je modelována jako 42 km silná kulová slupka okolo Země [14]. Pro zamezení nespojitosti v atmosférické hustotě jsou hodnoty ve výškách mezi 39 a 42 kilometry lineárně interpolovány, aby hustota nad 42 km byla nulová.

3.4.1 Lom světla

Jelikož atmosférický tlak ani teplota nejsou konstantní, není konstantní ani index lomu. Zemskou atmosféru musíme proto brát jako nehomogenní médium. Nehomogenní médium lze simulovat pomocí série soustředných homogenních vrstev, avšak tento postup není dostatečně přesný. Daleko lepší strategií je aplikace diferenciální rovnice uvedené v předešlé kapitole (2.1). Pro umožnění výpočtu na počítači je ovšem nutno rovnici nejprve numericky aproximovat. Seron et al. [13] nabízí po diskretizaci rovnice takovýto vztah:

$$\frac{dx_{j,i+1}}{dl} = \frac{n \frac{dx_{j,i}}{dl} + \frac{\delta n}{\delta x_j} \Delta l_i}{n_{i+1}} \quad (3.2)$$

Δl je potom délka kroku v jedné iteraci. Gradient lze pak aproximovat takto:

$$\frac{\delta n}{\delta x_j} \approx \frac{n(x_j + \Delta x_j) - n(x_j)}{\Delta x_j} \quad (3.3)$$

Za délku kroku je nejvhodnější zvolit 1 km, pro vyšší hodnoty dochází ke zkreslení dráhy paprsku, pro nižší pak pouze k malému zlepšení, za cenu vysokého počtu iterací [14]. Pro stanovení indexu lomu světla je třeba nejprve znát jeho hodnotu v úrovni moře. Ta je výsledkem funkce vlnové délky, která je modelována následujícím polynomem (Cauchyho rovnice):

$$n_0(\lambda) \approx 1,00027643 + \frac{1,2288 \times 10^{-1}}{\lambda^2} + \frac{3,555 \times 10^4}{\lambda^4} \quad (3.4)$$

Pro výpočet indexu lomu světla v libovolné výšce je však kromě indexu na hladině moře ještě třeba znát hustotu vzduchu pro danou výšku h . Vzorec pro její vypočtení vypadá takto:

$$\rho(h) = \frac{p(h)}{RT} \quad (3.5)$$

R je specifická plynová konstanta. Atmosférickou hustotu pak dosadíme do rovnice pro výpočet indexu lomu:

$$n(h, \lambda) = 1 + (n_0(\lambda) - 1) \frac{\rho(h)}{\rho_0} \quad (3.6)$$

3.4.2 Rozptyl světla

Yapo et al. [14] odkazují na fakt, že vliv Mieova rozptylu na výsledné osvětlení je patrný pouze u velmi tmavých zatmění (méně než 1 na Danjonově stupnici). Proto modelují pouze Rayleighův rozptyl a ostatní jevy jako Mieův a vícenásobný rozptyl neuvažují.

V naší práci nás z hlediska rozptylu světla zajímá pouze jeho vliv na původní paprsek. Proto je rozptyl simulován jako funkce vlnové délky tlumící intenzitu paprsku:

$$I = I_0 e^{-\tau}, \quad (3.7)$$

τ je potom optickou hloubkou dráhy l . Optická hloubka se vypočítá zintegrováním Rayleighova koeficientu rozptylu $\beta(\lambda, h)$ podle dráhy:

$$\tau_l(\lambda) = \int_l \beta(\lambda, h) dl. \quad (3.8)$$

Samotný koeficient $\beta(\lambda, h)$ lze vyjádřit následovně:

$$\beta(\lambda, h) = N(h)\sigma(\lambda), \quad (3.9)$$

kde $\sigma(\lambda)$ je účinný průřez Rayleighova rozptylu a N je objemová hustota částic pro vzduch ve výšce h . Její hodnota $N(h)$ se odvozuje takto:

$$N(h) = N_0 \frac{\rho(h)}{\rho_0} \quad (3.10)$$

a účinný průřez $\sigma(\lambda)$ pak takto:

$$\sigma(\lambda) = \frac{24\pi^3(n_s^2 - 1)^2}{\lambda^4 N_s^2(n_s^2 + 2)^2} F_k \quad (3.11)$$

kde F_k je Kingův opravný faktor, [14] uvádějí, že se v rozmezí viditelného světla téměř nemění (0,3%) a dosazují za něj konstantu $F_k = 1,050$.

Integrál rozptylového koeficientu se aproximuje krokováním po dráze paprsku:

$$I_{j+1} = I_j e^{-\beta_j \Delta l} \quad (3.12)$$

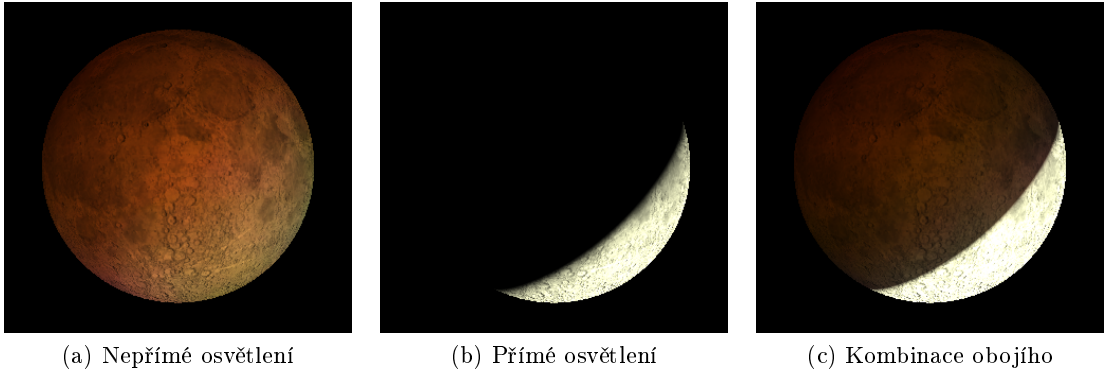
vzhledem k tomu, že krokování po dráze je využito už při simulaci lomu světla, nabízí se zvolit si stejnou délku kroku a oba jevy simulovat v jedné iterační smyčce.

3.4.3 Pohlcení světla

Pro simulaci absorpce světla aerosoly v zemské atmosféře používám jednoduchý model navržený v [14] - tedy rozšířit vztah používaný pro simulaci útlumu světla vlivem rozptylu popsaného v předchozí podkapitole o drobnou konstantu. Konstanta se započítá pouze ve výškách mezi 15 a 20 kilometry, mimo tento interval zůstává vztah nezměněn. Upravený vzorec vypadá následovně:

$$I_{j+1} = I_j e^{-(\alpha + \beta_j) \Delta l} \quad (3.13)$$

pro $\alpha = 10^{-4}$ je téměř všechno světlo pohlceno, pro $\alpha = 10^{-6}$ a menší je výsledné zatmění velmi jasné.



Obrázek 3.2: Rozdělení úlohy na výpočet přímého a nepřímého osvětlení.

3.5 Osvětlení Měsíce

Yapo et al. [14] rozdělují simulaci osvětlení Měsíce na dvě části. První je nepřímé osvětlení (obr. 3.2a) - tedy osvětlení pocházející z paprsků zakřivených v zemské atmosféře, které je modelováno pomocí sledování fotonů. Fotony jsou vystřelovány ze Slunce směrem do atmosféry a jejich dráha je zakřívována podle výše popsaného vztahu (3.2). Pokud nejsou fotony pohlceny Zemí a opustí atmosféru, pokračují po přímce dále. Zde je nutné zmínit osovou souměrnost úlohy - jelikož je výpočet zakřivení poměrně náročný, je vhodné šetřit čas rotací fotonů okolo osy Země-Slunce. Není ale dobré replikovat foton více než $100\times$, protože pak dochází ke vzniku viditelných artefaktů na povrchu Měsíce. Pokud pak foton zasáhne povrch Měsíce, je jeho přínos k výsledné intenzitě záření uložen do dvojrozměrné mapy osvětlení, která je poté nanesena na povrch Měsíce.

Druhou částí je pak přímé osvětlení (obr. 3.2b) - osvětlení, které během přímé cesty nenarazí na žádnou překážku, ani nevstoupí do atmosféry. V [14] jsou tyto paprsky modelovány pomocí Monte Carlo ray traceru. Z každého bodu na povrchu Měsíce jsou paprsky vystřeleny směrem ke Slunci, přičítají se pak ty, které nejsou zakryty Zemí včetně vrstvy atmosféry.

V této práci jsme také rozdělili část zabývající se osvětlením na dvě podúlohy o přímém a nepřímém osvětlení, použili jsme ale jiných metod.

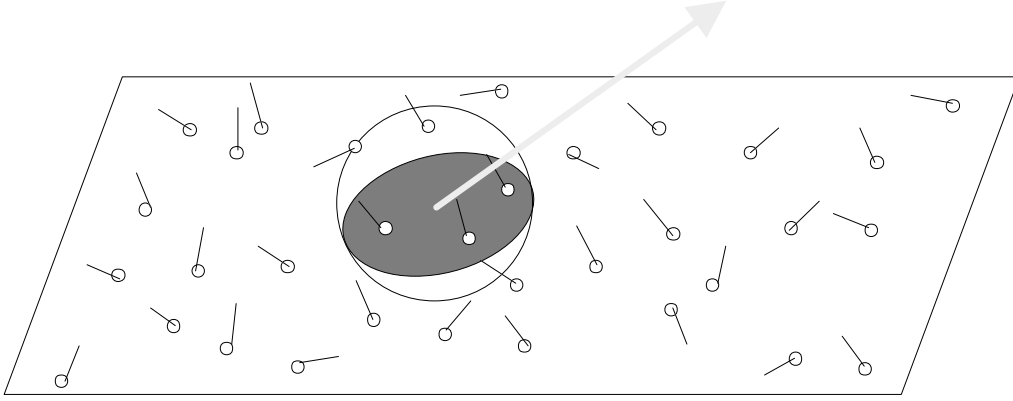
3.5.1 Nepřímé osvětlení

Pro reprezentaci nepřímého osvětlení Měsíce jsme použili metody zvané photon mapping [11]. Jedná se o dvoupřůchodový algoritmus, při jehož prvním průchodu se fotony, podobně jako u photon tracingu vystřelují do scény, jejich konečná pozice se pak ukládá do fotonové mapy.

Při druhém průchodu se pak pro každý pixel scény prohledává fotonová mapa a započítávají se fotony pouze v dosahu prohledávaného bodu (obr. 3.3). Výsledná intenzita L_r vzniklá důsledkem nepřímého osvětlení se tedy dá zapsat následujícím vzorcem [11]:

$$L_r = \frac{1}{\pi r^2} \sum_{i=1}^N f(\theta_i, \theta_r, \varphi) L_p w(d) \alpha_m(\lambda_p) \quad (3.14)$$

kde r je poloměr prohledávané koule (obr. 3.5), N je počet fotonů v této kouli, f je BRDF Měsíce, L_p je intenzita nesená fotonem, α_m je albedo povrchu Měsíce a w je váha fotonu v závislosti na d (vzdálenosti pozice fotonu od prohledávaného bodu).



Obrázek 3.3: Prohledávaná oblast na povrchu [11].

Váha fotonu udává jak velkou měrou se foton podílí na výsledném osvětlení. Jeho intenzita závisí na poloze vůči prohledávanému bodu a s rostoucí vzdáleností slábne (viz obr. 3.4) a je výsledkem Gaussovského filtru reprezentovaného tímto vzorcem [11]:

$$w(d) = 0,918 \left(1 - \frac{1 - e^{-1,953 \frac{d^2}{2r^2}}}{1 - e^{-1,953}} \right) \quad (3.15)$$

3.5.2 Přímé osvětlení

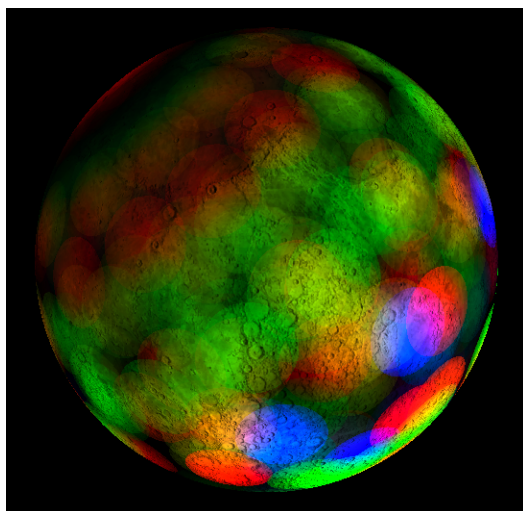
Pro přímé osvětlení jsme použili obyčejný ray tracer, kdy je z oka pozorovatele do každého pixelu stínítka vystřelen paprsek a pokud zasáhne Měsíc je výsledná intenzita přímého osvětlení spočítána takto:

$$L_r = L_i f(\theta_i, \theta_r, \varphi) \omega_S \cos \theta_i \alpha_m(\lambda) \quad (3.16)$$

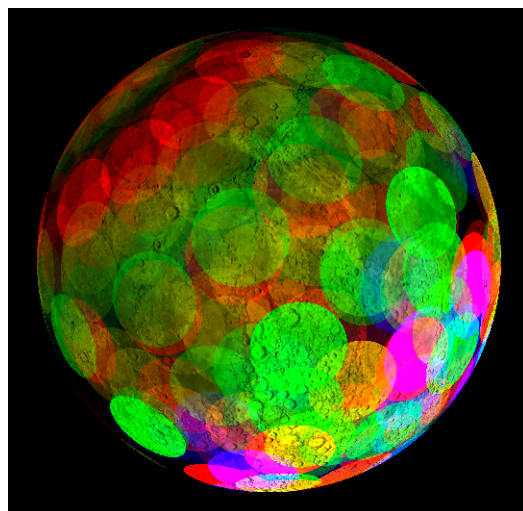
kde L_i je dopadající světlo, ω_S je prostorový úhel části slunečního disku viditelného z daného bodu Měsíce a θ_i je úhel svíraný normálou a dopadajícím světlem.

3.5.3 Převod barev

Doposud je světelná intenzita stále reprezentována jako série intenzit jednotlivých spektrálních složek (spektrum). Pro vykreslení je třeba toto spektrum nejprve převést do barevného prostoru CIE XYZ (integrováním součinů spektrálních složek a funkcí reprezentujících citlivost buněk lidského oka na jednotlivé vlnové délky). Tuto trojici hodnot lze potom vynásobením transformační maticí převést do finálního barevného modelu RGB, který je použit pro uložení výsledného obrázku.

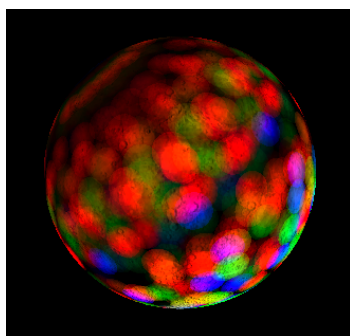


(a) S filtrem

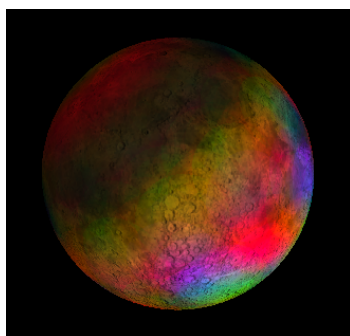


(b) Bez filtru

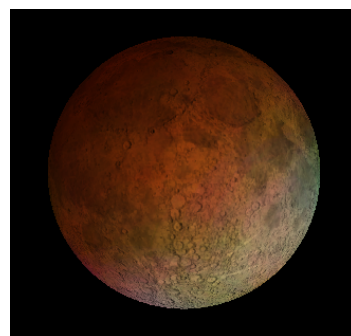
Obrázek 3.4: Vliv Gaussovského filtru na osvětlení jednotlivých fotonů.



(a) 200 km

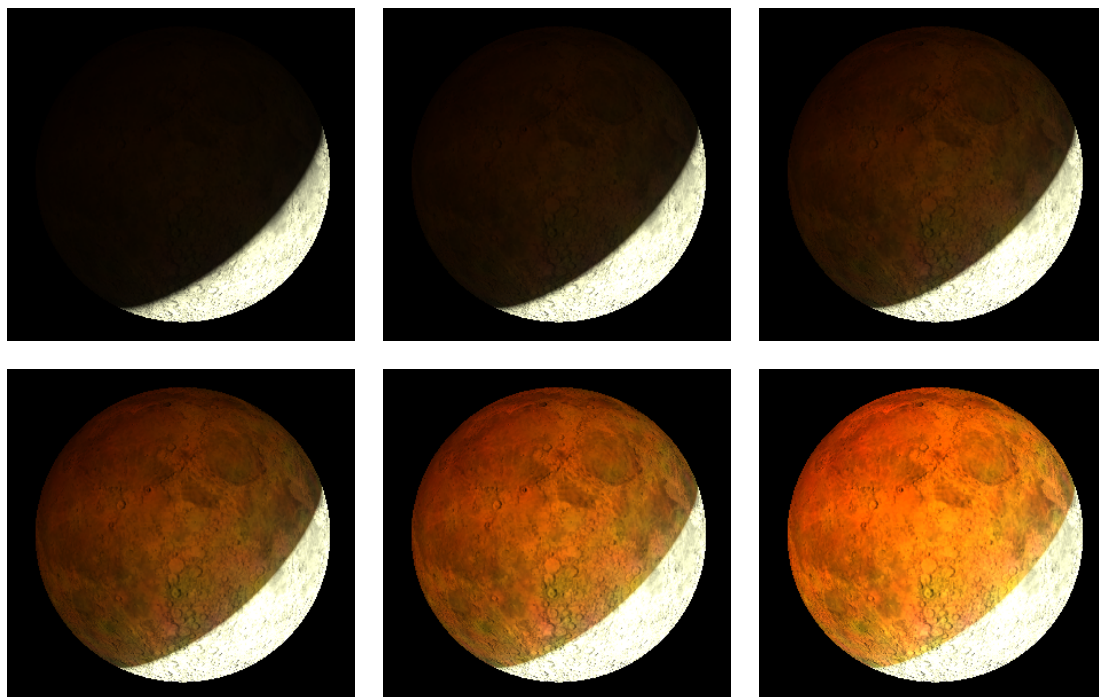


(b) 500 km



(c) 1000 km

Obrázek 3.5: Vliv poloměru prohledávané oblasti, každá scéna obsahuje tisíc fotonů.



Obrázek 3.6: Stejná scéna pro různé hodnoty koeficientu použitého v tone mappingu.

3.5.4 Tone mapping

Tone mapping je postup pro převod HDR¹ obrazu na výstup s omezeným dynamickým rozsahem. Jelikož jednotlivé zobrazované osvětlení mohou nabývat různých velikostí - velmi malá pro úplná zatmění až po velmi vysoká pro přímé sluneční světlo, musí se tento rozsah vhodně převést na rozsah zobrazitelný na daném zařízení. Změnou tónovacího koeficientu lze simulovat délku expozice snímku a s tím spojený jas výsledného obrazu (obr. 3.6). Tone mapping v barevném modelu CIE XYZ vypadá následovně [9]:

$$\begin{aligned} X' &= \frac{kX}{1 + kY} \\ Y' &= \frac{kY}{1 + kY} \\ Z' &= \frac{kZ}{1 + kY} \end{aligned}$$

kde k je hodnota tónovacího koeficientu.

3.6 Návrh aplikace

Nejprve musíme inicializovat výchozí stav aplikace, zde jde především o datum, pro které se má obrázek vykreslit a také pro parametry zobrazovacích algoritmů (rozměry obrázku,

¹high dynamic range

velikost fotonové mapy atd.). Použitím vzorců pro výpočet pozic vesmírných těles (kap. 2.2) zjistíme, v jaké konfiguraci se soustava právě nachází. Pak přejdeme k vystřelování fotonů do scény. Pro simulaci cesty fotonu skrz atmosféru používáme modely fyzikálních jevů (3.2 a 3.13). Po naplnění její kapacity provedeme výpočet osvětlení Měsíce. Výsledné spektrum převedeme do barevného modelu RGB a aplikujeme tone mapping. Konečnou barvu získáme kombinací osvětlení s texturou Měsíce. Celý běh aplikace zjednodušeně popisuje následující pseudokód:

```

načti vstupní parametry
nastav pozice vesmírných těles
while(fotonová mapa není plná) do{
    vytvoř foton ve slunci
    vystřel ho do zemské atmosféry
    while(foton je v atmosféře) do{
        ztlum intenzitu fotonu
        spočítej nový směr fotonu
        posuň foton o 1km v novém směru
        if(foton narazil do země){
            přeruš cyklus a začni pro nový foton
        }
    }
    for(i=0, i < počet replikací fotonu, i=i+1){
        if(foton zasáhne měsíc){
            ulož foton do fotonové mapy
        }
        otoč foton okolo osy země-slunce
    }
}
sestav viewport
foreach(pixel ve viewportu){
    vystřel paprsek
    if(paprsek zasáhl měsíc){
        spočítej nepřímé osvětlení
        spočítej přímé osvětlení
        sečti osvětlení a převed' spektrum do CIE XYZ
        aplikuj tone mapping
        převed' CIE do RGB
        spočítej hodnotu textury v daném místě
        vynásob hodnoty osvětlení s texturou
        nastav pixel na výslednou barvu
    }else{
        nastav pixel na černou barvu
    }
}

```


Kapitola 4

Realizace

V této kapitole je ukázáno, jak jsou jednotlivé entity, vztahy a postupy popsané v předchozích kapitolách, převedeny do prvků programovacího jazyka – objektů, metod, proměnných apod. Ukážeme si jak reprezentovat letící a uložený foton. Také předvedeme jak efektivně prohlédávat fotonovou mapu. Též je v této kapitole detailně popsána aplikační logika, ze které by, spolu s návrhem řešení ve třetí kapitole, mělo být jasné, jak vlastně výsledný program funguje.

4.1 Nástroje

Program je napsán v jazyce C++ ve vývojovém prostředí NetBeans verze 6.8. Pro zobrazení a komunikaci s grafickou kartou bylo použito API OpenGL¹, pro správu oken, obslužných funkcí a renderovacích smyček pak knihovna GLUT².

4.2 Aplikační logika

I když je teoretický aspekt úlohy poměrně komplikovaný, vlastní aplikační logika výsledného programu není nikterak složitá.

4.2.1 Nastavení pozic

Aplikace si ze vstupu (zde konfiguračního souboru) načte vstupní parametry, mezi nimi i data reprezentující čas, ve kterém dochází k měsíčnímu zatmění. Aplikace nejprve převede zadaný čas do juliánského kalendáře a poté sadou funkcí implementující astronomické vzorce (`getSunCoordinates`, `getMoonCoordinates`) vypočte aktuální pozice vesmírných těles pro zadaný čas. Výsledkem je instance třídy `SolarSystem` obsahující pozice a rozměry jednotlivých vesmírných těles figurujících v úloze.

¹Open Graphics Library

²OpenGL Utility Toolkit

4.2.2 Vystřelování fotonů

Po nastavení modelu sluneční soustavy do požadovaných pozic, přikročí aplikace k další fázi – tvorbě a vysílání fotonů do scény. Toto má na starosti třída **Tracer**, které je předána instance třídy **SolarSystem** vytvořená v předchozím kroku a nově vytvořená instance třídy **PhotonMap**. **PhotonMap** je implementací datové struktury, do které se budou ukládat fotony, které úspěšně dorazily k Měsíci.

Samotné trasování fotonů je pak implementováno následovně: třída **Tracer** nejprve vytvoří novou instanci třídy **FlyingPhoton**, potom ve smyčce iteruje její aktuální pozici a směr aplikací metod, které reprezentují lom světla. Zde také dochází k postupnému snižování intenzity fotonu vlivem dalších optických jevů. Pokud je **FlyingPhoton** úspěšně „proveden“ atmosférou, dochází k postupné rotaci jeho pozice a směru (osová souměrnost soustavy), pokud potom tento foton protne kouli Měsíce, jsou údaje v něm obsažené použity pro vytvoření třídy **Photon** (reprezentující „dopadlý“ foton), ta je následně uložena do fotonové mapy.

Tento postup se opakuje, dokud se fotonová mapa nenaplní (počet uložených fotonů nedosáhne požadovaného počtu, specifikovaném v konfiguračním souboru).

4.2.3 Vykreslení výsledku

Po naplnění scény fotony se přejde k jejímu vykreslení. To je zodpovědností třídy **Renderer**. Té je předána, nyní již naplněná, třída **PhotonMap** a třída **SolarSystem**, stejně jako informace o rozměrech výsledného obrázku a pozice pozorovatele.

Třída **Renderer** nejprve sestrojí stínítko (viewport) a poté pro každý jeho pixel vystřeluje do scény paprsky, které jsou, stejně jako všechny ostatní vektory v této úloze, implementovány třídou **Vector3**. Jestliže paprsek mine Měsíc, metoda vrací černou barvu. Pokud dojde k zásahu, třída odděleně spočítá přímé a nepřímé osvětlení (traverzováním nad fotonovou mapou), tyto dvě sady spektrálních intenzit sečte, převede do RGB a výsledek uloží do dvojrozměrného pole reprezentujícího osvětlení výsledného obrazu.

Do jiného pole ukládá **Renderer** informace o povrchu Měsíce. Ty zjišťuje z dodané textury. Globální souřadnice bodu na povrch Měsíce nejprve převede na lokální (s počátkem ve středu Měsíce). Trojici lokálních souřadnic převede do sférického souřadného systému a z něj do *uv* souřadnic textury. Nalezenou hodnotu si uloží do výše zmiňovaného pole.

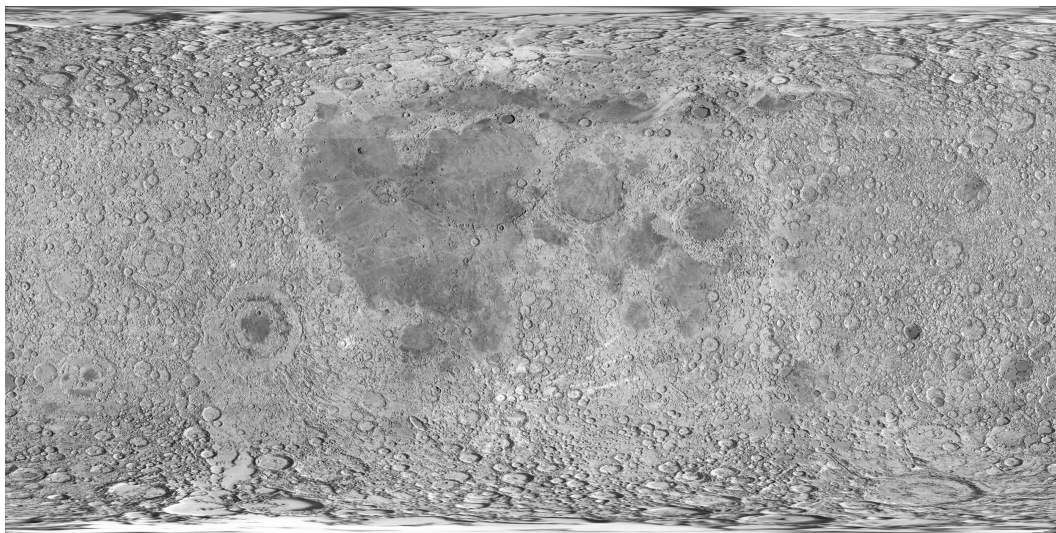
Pro samotné vykreslení si kreslicí smyčka GLUTu vyžádá tato dvě pole a na součiny jejich hodnot aplikuje tone mapping tak, aby byly výsledné hodnoty zobrazitelné.

4.3 Textura Měsíce

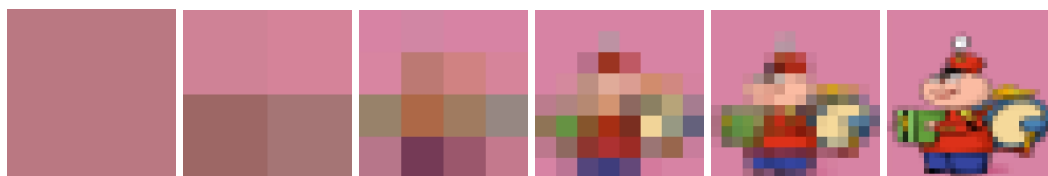
Způsob, jakým program zjišťuje údaje o povrchu Měsíce z textury je zjednodušeně popsán v předchozí kapitole. Celý proces je však o něco komplikovanější. Dvojrozměrný obrázek, ve kterém je textura uložena, je poměrně detailní (obr. 4.1). Pokud pak požadovaný render Měsíce má výrazně nižší rozměry než textura, není mapování jednoho texelu na jeden pixel vhodné. Výsledek totiž vypadá zrnitě, což je způsobeno přeskokováním texelů mezi jednotlivými paprsky. Pro odstranění tohoto nežádoucího efektu se používá mipmap [11], tedy řady textur, kde každá následující textura má poloviční rozměry textury předcházející - pixel nové textury je průměrem čtyř pixelů předchozí textury (obr. 4.2).

Volba vhodné textury pak vypadá následovně: jeden pixel scény se promítne na otexturovaný objekt a zjistí se uv souřadnice texelů nejdetailnější textury na jeho krajích. Řadové číslo nejvhodnější textury je pak dvojkový logaritmus počtu texelů, přes které je pixel scény promítnut.

Pro „hladký“ konečný vzhled textury je ještě vrácen vážený průměr čtyř nejbližších texelů k dotazovanému bodu namísto jednoho.



Obrázek 4.1: Textura Měsíce [7].



Obrázek 4.2: Řada mipmap od nejhrubší po nejdetailnější [11].

4.4 Popis jednotlivých objektů

V této kapitole si podrobněji popíšeme některé entity, které figuruji v aplikaci a jejich zodpovědnosti. Stejně tak jako ukázky některých částí vlastního zdrojového kódu programu.

4.4.1 Třída Tracer

Třída **Tracer** v aplikaci zodpovídá za dvě úzce související věci. První je vytváření nových fotonů ve Slunci, druhá je potom jejich trasování vesmírem, skrz zemskou atmosféru a pak směrem k Měsíci.

4.4.1.1 Vytváření fotonů

Tento úkol má na starost metoda `emitPhoton()`. Nejprve se vytvoří vektor uvnitř sluneční koule, který bude sloužit jako výchozí poloha nového fotonu. Potom se spočítá náhodná souřadnice v mezikruží atmosféry okolo Země, tak jak je viděno ze Slunce. Nakonec metoda určí náhodnou vlnovou délku v rozmezí 380 - 780 nm a vrátí novou instanci třídy `FlyingPhoton` vytvořenou pomocí výše popsaných údajů.

```
FlyingPhoton* emitPhoton() {
    double x, y, z;
    // random position in the Sun
    do {
        x = (double) rand() / (RAND_MAX / 2) - 1;
        y = (double) rand() / (RAND_MAX / 2) - 1;
        z = (double) rand() / (RAND_MAX / 2) - 1;
    } while (x * x + y * y + z * z > 1);

    // photon's vector of origin
    Vector3 origin = (Vector3(x, y, z) * syst->sun->getRadius())
        + syst->sun->getCenter();

    //sun-earth axis, up and right vector - used for aiming at earth's atmosphere
    Vector3 axis = (syst->earth->getCenter()
        - syst->sun->getCenter()).normalize();
    Vector3 up = Vector3(0, 1, 0);
    Vector3 right = Vector3::crossProduct(axis, up);

    Vector3 aim;

    do { // get random position in the annulus of atmosphere and earth
        do {
            x = (double) rand() / (RAND_MAX / 2) - 1;
            y = (double) rand() / (RAND_MAX / 2) - 1;
        } while (x * x + y * y > 1);
        x *= (syst->earth->getRadius() + 42);
        y *= (syst->earth->getRadius() + 42);
        aim = (right * x) + (up * y);
        // aims at the annulus of atmosphere and earth
    } while (aim.getSize() < syst->earth->getRadius());

    // random wavelength between 380 and 780
    int wavelength = (rand() % WAVELENGTH_SAMPLES) * WAVELENGTH_STEP_SIZE + 380;

    // new photon ready for tracing
    return new FlyingPhoton(origin, (aim - origin).normalize(), wavelength, photonCnt);
}
```

4.4.1.2 Sledování fotonů

Tato část úlohy je implementována metodou `tracePhoton()`. Metoda přijímá za vstupní parametr instanci třídy `FlyingPhoton` vytvořenou podle výše popsaného postupu. Ze všeho nejdříve se zjistí bod, ve kterém foton vstoupí do atmosféry. Poté se v cyklu iteruje jeho intenzita (vlivem rozptylu a absorpce) a hlavně poloha a směr (lom). Cyklus běží, dokud foton neopustí atmosféru nebo není pohlcen povrchem Země.

```
double prevIndex = 1;
do {
    // light attenuation due to scattering and atmospheric dust
    photon->intensity *= pow(M_E, -opticalDepth(photon->position.getSize()
        - EARTH_RADIUS, photon->wavelength, stepSize));

    // direction iteration logic here
    double currIndex = refractiveIndex(photon->position.getSize() - EARTH_RADIUS,
        photon->wavelength);
    Vector3 newDirection = getNewDirection(photon, prevIndex, currIndex);
    prevIndex = currIndex;

    //make 1km step in new direction
    photon->direction = newDirection.normalize();
    photon->position = photon->position + (photon->direction * stepSize);

    //end tracing if the photon is absorbed by Earth
    if (photon->hasHitEarth(syst->earth)) {
        delete photon;
        return;
    }
} while (photon->isInAtmo());
```

Metoda `getNewDirection()` je přímou implementací diskretizace diferenciální rovnice popsané v předchozích kapitolách (3.2). Vektor g je pak aproximací gradientu popsaného tamtéž (3.3).

```

Vector3 getNewDirection(FlyingPhoton* photon, double prevIndex, double currIndex) {
    Vector3 up = photon->position;
    up.normalize();
    double height = photon->position.getSize() - EARTH_RADIUS;
    double w = photon->wavelength;
    Vector3 g = up *
        ((refractiveIndex(height + 0.01, w) - refractiveIndex(height, w)) / 0.01);
    return (photon->direction * prevIndex + g * stepSize) / currIndex;
}

```

Pokud foton úspěšně opustí atmosféru, dojde zde k využití osové souměrnosti soustavy podle osy Slunce-Země. Jelikož je iterační smyčka zdaleka nejnáročnějším místem v této části aplikace, lze ušetřit výpočetní čas a další fotony simulovat pootočením polohy a směru fotonu podle této osy. Počet takto replikovaných fotonů není konstantní, ale závisí na vzdálenosti od osy souměrnosti. Zabrání se tak umělému nárůstu hustoty fotonů blíže osy. Dále zde program počet iterací náhodně pozmění, aby nedocházelo ke zřetelným symetrickým obrazcům na povrchu Měsíce. Pokud foton nakonec zasáhne Měsíc, uloží se do fotonové mapy.

```

// photon replication
for (int i = 0; i < iterations; i++) {
    Vector3* impact = photon->getSphereIntersection(syst->moon);
    // photon hits moon
    if (impact != NULL) {
        // insert in photon map (unless the map is already full)
        if (map->isFull()) {
            delete impact;
            delete photon;
            return;
        } else {
            map->add(new Photon(*impact, photon->direction,
                               photon->wavelength, photon->intensity));
        }
        delete impact;
    }

    // rotate photon position and direction around sun-earth axis
    photon->position.arbitraryAxisRotation((2 * M_PI) / iterations, axis);
    photon->direction.arbitraryAxisRotation((2 * M_PI) / iterations, axis);
}

```

4.4.2 Třídy PhotonMap, Photon a Node

Tyto tři třídy slouží jak pro ukládání fotonů během jejich vystřelování do scény, tak pro jejich následné vyhledávání během vykreslování.

4.4.2.1 Photon

Třída Photon je přepravkou pro všechny údaje související s uložením fotonu, tedy místem dopadu, směrem ze kterého přiletěl, vlnovou délku a světelnou intenzitu.

```
class Photon {
    Photon(Vector3 &pos, Vector3 &direction, int wavelength, double intensity);
    float* position; // coordinates of moon's surface where the photon fell
    float* direction; // coordinates of direction from which the photon arrived
    int wavelength; // wavelength of the light
    double intensity; // light intensity carried by photon
};
```

4.4.2.2 PhotonMap

PhotonMap obsahuje pole všech fotonů, které zasáhly povrch Měsíce. Samotné pole sice stačí pro ukládání fotonů, ale pro jejich vyhledávání je velmi neefektivní. Fotony totiž dopadají na Měsíc na náhodná místa a při vkládání do pole nejsou nijak řazeny. Při zobrazování by pak algoritmus musel pro každý pixel projít celé pole a porovnat vzdálenost s každým uloženým fotonem. Pro zefektivnění hledání nejbližších fotonů je nad polem vybudován kd-strom, který je popsán v následující podkapitole.

4.4.2.3 Node

Třída Node je implementací uzlu kd-stromu. Pokud je listem, obsahuje v sobě část původního pole fotonů. Pokud listem není, obsahuje odkazy na své dva potomky (kd-stromy jsou binární), identifikátor roviny řezu a souřadnici, kterou je řez veden.

```
typedef struct {
    Node* left; // left subtree
    Node* right; // right subtree
    int cut; // plane of the cut
    double coord; // coordinate of the cut
} CHILDREN; // structure of inner node

typedef struct {
    Photon** array; // photons contained in this leaf
    int length; // number of photons contained in this leaf
} PHOTONS; // structure of leaf node

typedef union {
    CHILDREN children; // inner node
    PHOTONS photons; // leaf node
} DATA;

class Node {
```

```

public:
    Node(Photon** a, int len, int currDepth, int cut, int maxDepth);
    Node(const Node& orig);
    virtual ~Node();
    bool isLeaf; // indicator if node is a leaf
    DATA data; // contents of node
};

```

Vytvoření kd-stromu je potom pouze otázkou zavolání konstruktoru pro kořenový uzel. Celý strom se pak rekurzivně postaví. Pokud není dosaženo maximální povolené délky stromu, je vytvářený uzel uzlem vnitřním. V takovém případě si setřídí jemu předané fotony podle roviny, kterou povede řez. Řez je veden tak, aby dělil toto seřazené pole na dvě stejné poloviny. každou polovinu pak předá konstruktorům svých potomků. Rovina řezu se pravidelně střídá. Pokud je ale dosažena hloubka maximální, je tento uzel listem a pouze si uloží všechny fotony do svého pole.

```

Node::Node(Photon** a, int len, int currDepth, int cut, int maxDepth) {
    if (currDepth < maxDepth) { // inner node
        this->isLeaf = false;
        this->data.children.cut = cut;
        // sort photons by cutting plane
        quicksort(a, 0, len - 1, cut);
        // compute coordinate of the cut
        this->data.children.coord =
            (a[len / 2 - 1]->position[cut] + a[len / 2]->position[cut]) / 2;
        // create left subtree
        this->data.children.left =
            new Node(a, len / 2, currDepth + 1, (cut + 1) % 3, maxDepth);
        // create right subtree
        if (len % 2 == 0) {
            this->data.children.right = new Node(&a[len / 2], len / 2,
                currDepth + 1, (cut + 1) % 3, maxDepth);
        } else {
            this->data.children.right = new Node(&a[len / 2], (len / 2) + 1,
                currDepth + 1, (cut + 1) % 3, maxDepth);
        }
    }

    } else { // leaf
        this->isLeaf = true;
        this->data.photons.array = new Photon*[len];
        this->data.photons.length = len;
        for (int i = 0; i < len; i++) {
            this->data.photons.array[i] = a[i];
        }
    }
}

```

4.4.3 Třída `Renderer`

Třída `Renderer` je co do délky kódu největší částí aplikace. Zodpovídá za sestavení výsledného obrazu ze všech předchozích částí aplikace, tedy ze vzájemných pozic vesmírných těles, fotonové mapy sestavené třídou `Tracer` a textury měsíčního povrchu. Jak již bylo zmíněno v popisu aplikační logiky, přímé a nepřímé osvětlení se počítá odděleně (metody `directIllumination()` a `indirectIllumination()`). A zatímco výpočet přímého osvětlení je poměrně jednoduchý – je to vlastně pouze implementace vzorce uvedeného v návrhu řešení (3.16), výpočet nepřímého osvětlení a s ním spojené prohledávání fotonové mapy je o něco komplikovanější:

```
void indirectIllumination(Node* node, Vector3* moonpoint, double* intensities,
    Vector3 normal, Vector3 reflectedLight) {
    // if leaf, iterate on leaf's photons, if within perimeter add to intensities array
    if (node->isLeaf) {
        const int len = node->data.photons.length;

        for (int i = 0; i < len; i++) {
            Photon* ph = node->data.photons.array[i];
            Vector3 phPos = Vector3(ph->position);
            double dist = (phPos - *moonpoint).getSize();

            // photon intensity should be added to intensities
            if (dist < searchRadius) {
                Vector3 incidentLight = (Vector3() - Vector3(ph->direction));
                double area = (M_PI * searchRadius * searchRadius);
                double brdf = moonBRDF(
                    Vector3::getAngle(reflectedLight, normal),
                    Vector3::getAngle(incidentLight, normal),
                    Vector3::getAngle(reflectedLight, incidentLight));
                double weight = gaussianFilter(dist);

                intensities[(ph->wavelength - 380) / WAVELENGTH_STEP_SIZE] +=
                    ph->intensity * brdf *
                    getLunarReflectanceCoefficient(ph->wavelength) * weight / area;
            }
        }
    }
    else { // recursively search node's children
        int cut = node->data.children.cut;
        double coord = node->data.children.coord;

        if ((*moonpoint)[cut] + searchRadius < coord) {
            // search perimeter lies completely in the left child
            indirectIllumination(node->data.children.left, moonpoint,
                intensities, normal, reflectedLight);
        }
    }
}
```

```
} else if ((*moonpoint)[cut] - searchRadius > coord) {
    // search perimeter lies completely in the right child
    indirectIllumination(node->data.children.right, moonpoint,
        intensities, normal, reflectedLight);

} else {
    // search perimeter is located in both children, has to search both
    indirectIllumination(node->data.children.left, moonpoint,
        intensities, normal, reflectedLight);
    indirectIllumination(node->data.children.right, moonpoint,
        intensities, normal, reflectedLight);
}
}
```


Kapitola 5

Výsledky

Tato kapitola obsahuje výstupy programu. Všechny prezentované obrázky a údaje byly pořízeny na počítači s procesorem Intel Core 2 Duo 2,53 GHz, 4 GB RAM a grafickou kartou NVIDIA GeForce 9400M. Obrázky jednotlivých zatmění jsou porovnány se skutečnými fotografiemi Měsíce. V desetiminutových intervalech je zde také zachycen průběh dvou zatmění - úplného zatmění, ke kterému došlo 21. února 2008 a částečného, které se událo 26. června 2010, jak je simulovala finální aplikace. Všechny fotografie byly převzaty z www.wikipedia.org.

Poukazujeme na fakt, že co se týče polohy Měsíce, jsou naše výpočty velmi přesné. Pro každé testované datum, ve kterém došlo k měsíčnímu zatmění, k němu došlo i v naší aplikaci. Pozice Měsíce v zemském stínu se velice blížila skutečnosti a dosahovali jsme pouze minimálních odchylek.

Zjistili jsme, že námi použitá textura povrchu Měsíce není pro pozorování zatmění ze Země ideální. Jelikož je vytvořena kombinací satelitních snímků, obsahuje velké množství detailů, které nejsou lidským okem pozorovatelné. Za zmínku stojí především vysoká ostrost kráterů a jejích stínů.

Simulace intenzity přímého osvětlení je velmi přesná. Z obrázků 5.2 a 5.3 je jasné patrné, že pokles intenzity směrem ke středu zemského stínu je blízký skutečnosti.

Barva přímého osvětlení má v naší simulaci oproti realitě lehce zelený odstín. Domníváme se, že jde o efekt vzniklý zanedbáním atmosféry pro světlo odražené od Měsíce k pozorovateli. Yapo et al. [14] například uvádí, že ozónová vrstva pohlcuje některé vlnové délky. Toto uvádíme jako jednu z možností pro budoucí rozšíření.

U fotografií zatmění s delší expozicí, kde přímé osvětlení dopadá pouze na zlomek měsíčního disku, dochází v těchto místech k přepalu (obr. 5.4b). V simulovaných obrázcích k němu ale nedochází, to je přímým důsledkem tone mappingu. Tone mapping totiž snižuje dynamický rozsah obrázku tak, aby byl celý zobrazitelný. Použitím jiného algoritmu pro převod barev by bylo možné dosáhnout vzhledu více podobného fotografiím.

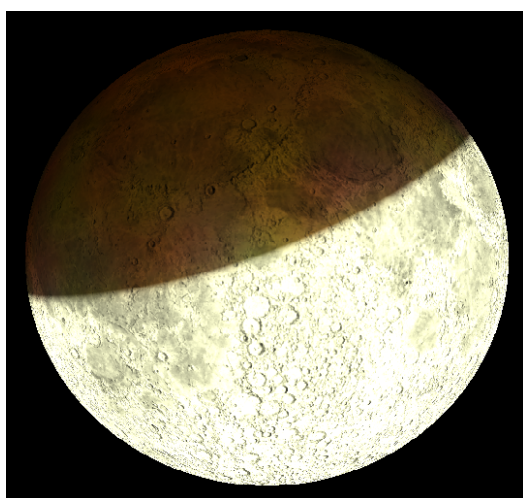
Skvrnitý vzhled nepřímého osvětlení pozorovatelný na některých obrázcích (např. obr. 5.5a nahoře) je způsoben nerovnoměrným rozvrstvením fotonů ve scéně. Lze se ho zbavit dvěma způsoby: zvýšením počtu fotonů na množství bránící vzniku pozorovatelných lokálních extrémů (jde o řádově miliony fotonů), druhá možnost je pak zvýšení poloměru prohledávané oblasti, tuto možnost ale nedoporučujeme, protože u ní dochází k jinému defektu -

nazelenalému pruhu na hranici přímého a nepřímého osvětlení. K tomuto jevu dochází důsledkem započítání fotonů z míst, na která dopadá přímé osvětlení, i do oblasti, kde dopadá pouze osvětlení nepřímé.

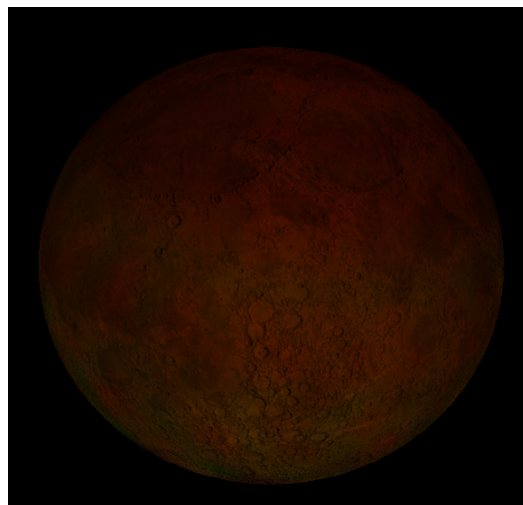
5.1 Měření

Měření rychlosti jsme provedli pro dvě scény, pro částečné zatmění, ke kterému došlo 26. června 2010 ve 12:00 UTC (obr. 5.1a, tab. 5.1) a pro úplné zatmění, které nastalo 21. února 2008 (obr. 5.1b, tab. 5.2). Měřili jsme jednak dobu, kterou program zabere sledování fotonů a také čas, za který bude vykreslen výsledný obrázek. Oboje jsme měřili v závislosti na rozlišení obrázku a na počet fotonů vystřelených do scény. Poloměr prohledávané oblasti byl pro všechna měření 100 km.

Z naměřených údajů můžeme vyčíst následující poznatky: doba, po kterou program sleduje fotony, roste lineárně s jejich počtem, velikost rozlišení na ni nemá vliv. Čas, který aplikace stráví vykreslováním, je lineárně závislý na počtu pixelů výsledného obrazu. Vzhledem k tomu, že při prohledávání fotonové mapy je použito kd-stromu, není vykreslovací čas lineárně závislý na počtu fotonů, při vhodné zvolené hloubce stromu se závislost bude blížit logaritmické závislosti. Za povšimnutí stojí fakt, že sledování fotonů ve druhé scéně, bylo více jak $2\times$ rychlejší než pro první scénu, to je způsobeno větší blízkostí Měsíce ke středu zemského stínu, na Měsíc tak dopadá větší část replikovaných fotonů.



(a) Scéna 1 - 26.6. 2010 12:00 UTC



(b) Scéna 2 - 21.2. 2008 3:20 UTC

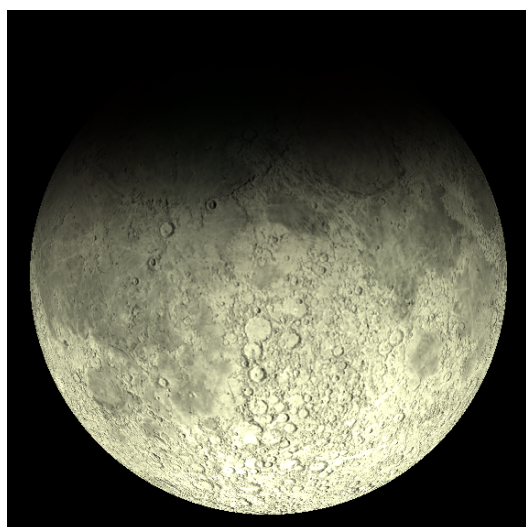
Obrázek 5.1: Scény, na kterých bylo provedeno měření rychlosti.

	sledování při rozlišení (px)			vykreslení při rozlišení (px)		
fotony	640×480	800×600	1280×800	640×480	800×600	1280×800
1000	2	2	2	2	3	6
5000	13	13	13	3	4	8
10000	27	27	27	4	5	11
50000	137	137	135	11	16	39
100000	273	273	276	20	30	74

Tabulka 5.1: Doba sledování fotonů a vykreslení pro první scénu (s).

	sledování při rozlišení (px)			vykreslení při rozlišení (px)		
fotony	640×480	800×600	1280×800	640×480	800×600	1280×800
1000	1	1	1	2	2	5
5000	5	5	5	2	3	7
10000	10	10	10	4	5	11
50000	54	52	53	10	16	38
100000	106	106	108	19	29	72

Tabulka 5.2: Doba sledování fotonů a vykreslení pro druhou scénu (s).

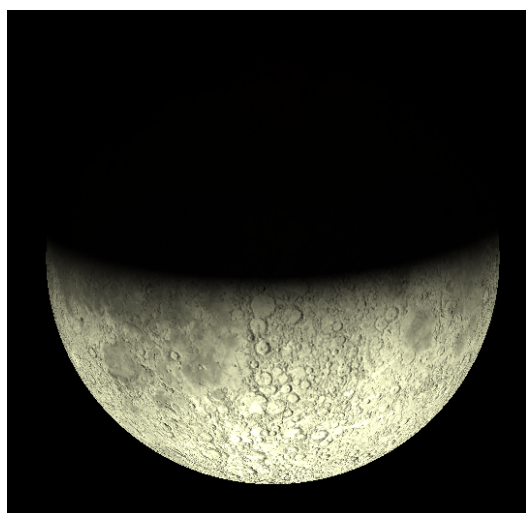


(a) Simulace

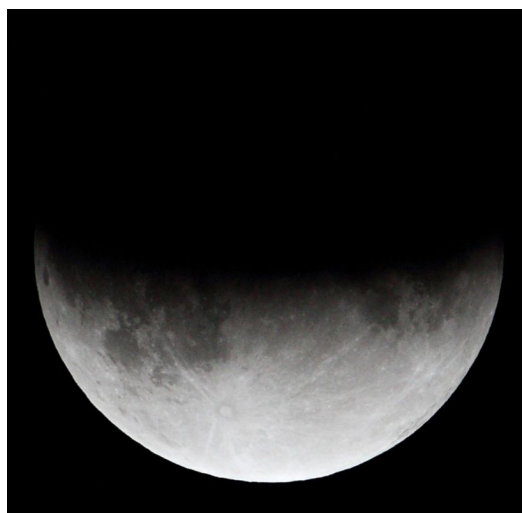


(b) Fotografie

Obrázek 5.2: Částečné zatmění, ke kterému došlo 7. září 2006 18:51 UTC.

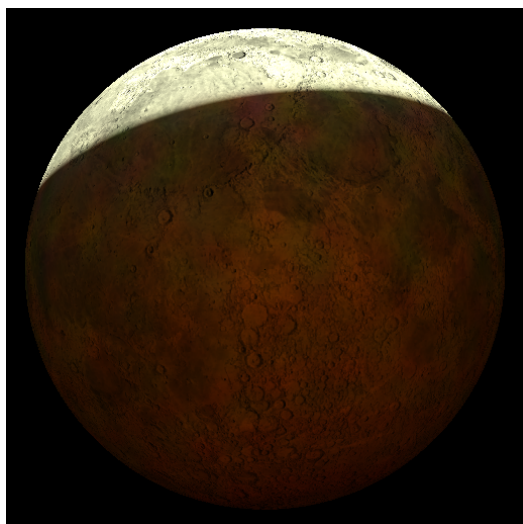


(a) Simulace



(b) Fotografie

Obrázek 5.3: Částečné zatmění, ke kterému došlo 26. června 2010 11:31 UTC.

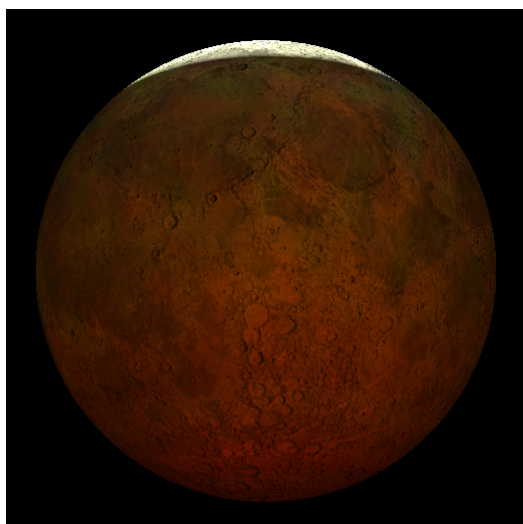


(a) Simulace



(b) Fotografie

Obrázek 5.4: Téměř úplné zatmění, ke kterému došlo 16. srpna 2008 21:10 UTC.

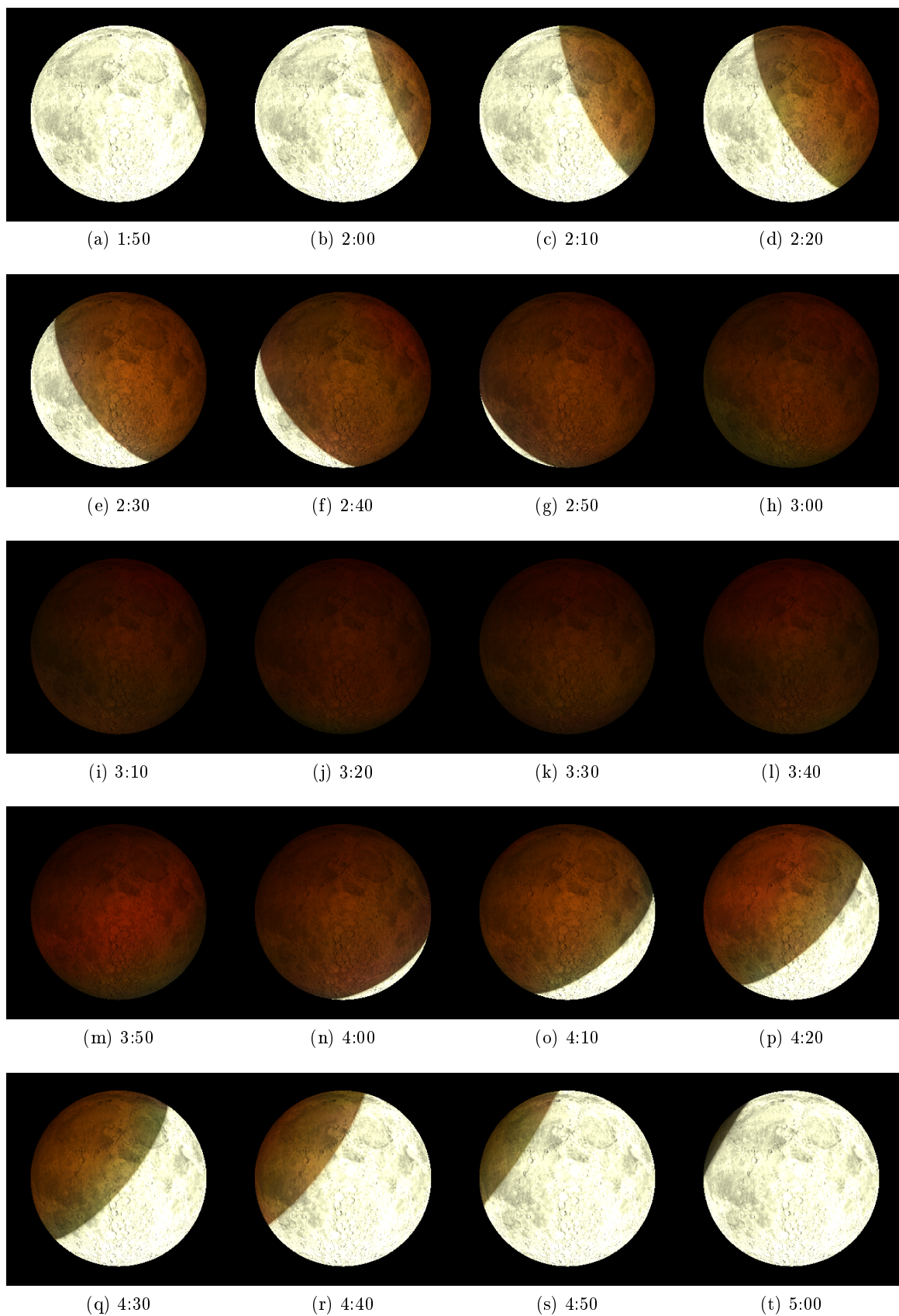


(a) Simulace

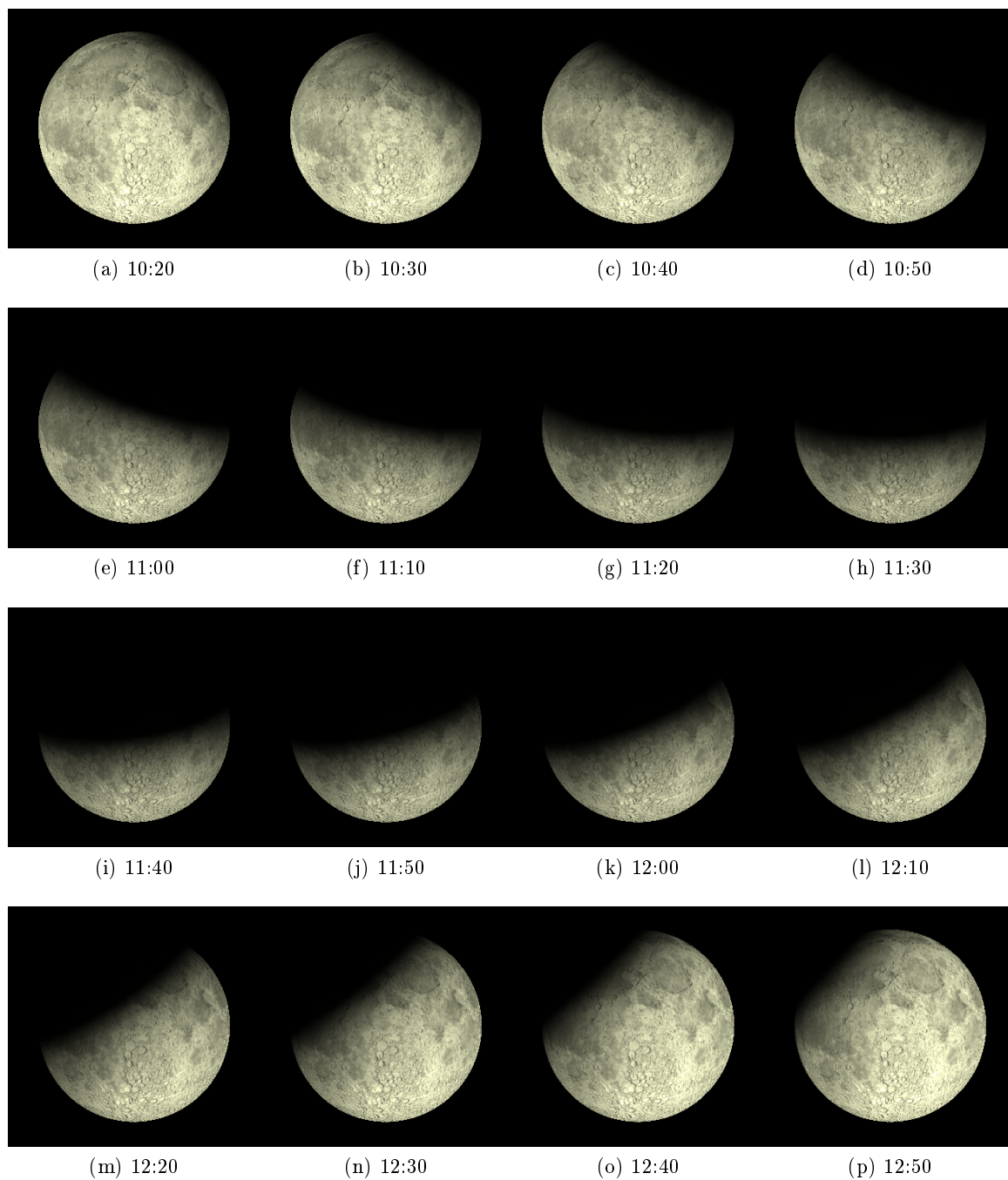


(b) Fotografie

Obrázek 5.5: Téměř úplné zatmění, ke kterému došlo 24. března 1997 4:45 UTC.



Obrázek 5.6: Průběh úplného zatmění 21. února 2008.



Obrázek 5.7: Průběh částečného zatmění 26. června 2010.

Kapitola 6

Závěr

V této práci jsme popsali jednotlivé kroky nezbytné k simulaci zatmění Měsíce. Práce obsahuje popis jak modelovat jednotlivé optické jevy, které se na výsledném vzhledu podílejí, jmenovitě pak lom, rozptyl a absorpci světla. Stejně tak jsou v práci obsaženy i astronomické výpočty sloužící k přesnému stanovení pozic jednotlivých vesmírných těles. Jsou zde popsány i jednotlivé algoritmy počítačové grafiky použité ke konečnému zobrazení. Pro simulaci nepřímého osvětlení jsme použili fotonových map spolu s modifikovaným ray tracerem, který, kvůli cestě skrz médium s nestejným indexem lomu, trasuje světlo po zakřivených drahách. Pro přímé osvětlení jsme potom aplikovali obyčejný ray tracer. Všechny tyto části pak byly implementovány ve výsledné aplikaci.

Naše aplikace věrně simuluje vzájemnou polohu vesmírných těles pro zadané datum. Přesná je i intenzita přímého osvětlení a její pokles směrem ke středu zemského stínu. V barvě přímého osvětlení dochází k malé odchylce od skutečnosti. Transformace monochromatických složek spektra do výsledné barvy nepřímého osvětlení funguje správně a výsledná barva povrchu zatmělého Měsíce je věrohodná. Konstatujeme, že námi použitá textura Měsíce je pro naše potřeby příliš detailní. Jelikož náš program používá tone mapping, nedochází na výsledných obrázcích k přepalům, což je hlavní rozdíl oproti fotografiím skutečných zatmění.

6.1 Budoucí rozšíření

Aplikace splňuje požadovanou funkčnost, nicméně se nabízí další možnosti pro zdokonalení.

Jedním z možných vylepšení by byla implementace vlivu atmosféry na světlo odražené z Měsíce k pozorovateli. Atmosféra je totiž brána v potaz pouze při cestě světla na Měsíc. Dalším potenciálním rozšířením je možnost nastavení polohy kamery a tedy pozorování zatmění i z dalších míst ve vesmíru, popř. přesně zadaných míst na povrchu Země.

Je zde také prostor pro zrychlení vlastního běhu programu. Poukazujeme na fakt, že úloha je velmi dobře paralelizovatelná a vícevláknová implementace, popř. implementace na GPU¹ by vedla ke znatelné časové úspoře.

¹graphics processing unit

Literatura

- [1] *1976 Standard Atmosphere Calculator* [online]. 2008. Dostupné z: <<http://www.digitaldutch.com/atmoscalc/>>.
- [2] *Bidirectional reflectance distribution function* [online]. 2011. [cit. 9.5.2011]. Dostupné z: <<http://en.wikipedia.org/wiki/Brdf>>.
- [3] *Lunar eclipse* [online]. 2011. [cit. 17.5.2011]. Dostupné z: <http://en.wikipedia.org/wiki/Lunar_eclipse>.
- [4] *2000 ASTM Standard Extraterrestrial Spectrum Reference E-490-00* [online]. 2004. [cit. 18.10.2004]. Dostupné z: <<http://www.cs.utah.edu/~bigler/classes/cs6650/tonemapping.html>>.
- [5] *Danjon scale of lunar eclipse brightness* [online]. 2005. [cit. 11.8.2005]. Dostupné z: <<http://eclipse.gsfc.nasa.gov/OH/Danjon.html>>.
- [6] *The Sun as an Energy Resource* [online]. 2003. Dostupné z: <<http://www.volker-quaschnig.de/articles/fundamentals1/index.php>>.
- [7] *Planet Earth Texture Maps* [online]. 2006. Dostupné z: <<http://planetpixelemporium.com/earth.html>>.
- [8] *May 1453 lunar eclipse* [online]. 2011. [cit. 20.1.2011]. Dostupné z: <http://en.wikipedia.org/wiki/May_1453_lunar_eclipse>.
- [9] BIGLER, J. *Spectral Tone Mapping* [online]. Dostupné z: <<http://www.cs.utah.edu/~bigler/classes/cs6650/tonemapping.html>>.
- [10] GUTIERREZ, D. et al. Simulation of atmospheric phenomena. *Computer & Graphics*. 2006, 6, 30, s. 994–1010.
- [11] JAROSZ, W. – JENSEN, H. W. – DONNER, C. Advanced global illumination using photon mapping. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, s. 2:1–2:112, New York, NY, USA, 2008. ACM. doi: <http://doi.acm.org/10.1145/1401132.1401136>. Dostupné z: <<http://doi.acm.org/10.1145/1401132.1401136>>.
- [12] JENSEN, H. W. et al. A physically-based night sky model. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, s. 399–408, New York, NY, USA, 2001. ACM. doi: <http://doi.acm.org/10.1145/383259.383306>. Dostupné z: <<http://doi.acm.org/10.1145/383259.383306>>. ISBN 1-58113-374-X.

- [13] SERON, F. J. et al. Visualizing Sunsets through Inhomogeneous Atmospheres. *Computer Graphics International Conference*. 2004, 0, s. 349–356. ISSN 1530-1052. doi: <http://doi.ieeecomputersociety.org/10.1109/CGI.2004.1309232>.
- [14] YAPO, T. C. – CUTLER, B. Rendering lunar eclipses. In *Proceedings of Graphics Interface 2009*, GI '09, s. 63–69, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society. Dostupné z: <<http://portal.acm.org/citation.cfm?id=1555880.1555901>>. ISBN 978-1-56881-470-4.

Příloha A

Instalační a uživatelská příručka

Aplikace je zkompileována do souboru `lunareclipse.exe`, před spuštěním se ujistěte se, že jsou ve stejném adresáři i následující soubory:

- `glut32.dll` - dynamická knihovna OpenGL
- `config.txt` - konfigurační soubor
- `MoonMap.tga` - textura Měsíce

Struktura konfiguračního souboru je velmi jednoduchá, na jednom řádku je vždy název parametru a na následujícím řádku jeho hodnota. Pro správný běh je nutné dodržet tuto strukturu a pořadí parametrů.

Příloha B

Obsah přiloženého CD

- `bin/` - výsledný program
- `documentation/` - dokumentace zdrojových kódů
- `src/` - zdrojové kódy programu
- `text/` - text této bakalářské práce ve formátu PDF a \LaTeX