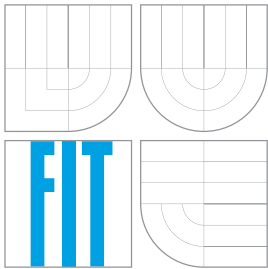


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **SIMULACE VLASTNOSTÍ OBJEKTIVU**

SIMULATION OF LENSES FEATURES

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. MICHAL KUČIŠ**

**VEDOUcí PRÁCE**

SUPERVISOR

**Doc. Dr. Ing. PAVEL ZEMČÍK**

BRNO 2012

## Abstrakt

Algoritmy počítačového vidění typicky zpracovávají snímky získané fotoaparátem nebo kamerou. Tyto snímky obsahují nedokonalosti, které jsou způsobeny konstrukcí objektivu. Táto práce se věnuje popisu simulace, která upravuje počítačem generované snímky 3D generovaného světa na snímky, které simulují výstup konkrétního fotoaparátu. Popisovaný algoritmus umožňuje simulovat geometrické zkreslení, chromatickou aberaci, efekt hloubky ostrosti, pohybové rozostření, vinětaci a odlesky. Simulátor taktéž umožňuje simulovat adaptaci aperturní clony v objektivu.

## Abstract

Computer vision algorithms typically process real world image data acquired by cameras or video cameras. Such image data suffer from imperfections cause by the acquisition process. This paper focuses on simulation of the acquisition process on simulation of the acquisition process in order to enable rendering of images based on a 3D generated model. Imperfections, such as geometry distortions, chromatic aberration, depth of field effect, motion blur, vignetting and lens flare are considered.

## Klíčová slova

simulace nedokonalostí objektivu, efekt hloubky ostrosti, geometrické zkreslení, chromatická aberace, vinětace, odlesk, pohybové rozostření, clona

## Keywords

camera imperfections simulation, depth of field effect, distortions, chromatic aberration, vignetting, lens flare, motion blur, aperture

## Citace

Michal Kučiš: Simulace vlastností objektivu, diplomová práce, Brno, FIT VUT v Brně, 2012

# Simulace vlastností objektivu

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Doc. Dr. Ing. Pavla Zemčíka

.....  
Michal Kučiš  
21. května 2012

## Poděkování

Touto cestou chcem poďakovať Doc. Dr. Ing. Pavlovi Zemčíkovi za jeho cenné rady ohľadom tvorby simulátora a aj za rady ohľadom tvorby odborných textov. Chcem poďakovať aj Oliverovi Zendelovi za vstupné dáta pre simulátor a za rady ohľadom rozhrania simulátora. Ďakujem aj Ing. Michalovi Seemanovi za uvedenie do problematiky. V neposlednej rade chcem poďakovať mojím rodičom za ich podporu pri mojom štúdiu na vysokej škole ako aj za pomoc pri písaní tejto technickej správy.

© Michal Kučiš, 2012.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Zhrnutie súčasného stavu</b>	<b>4</b>
2.1	Rozdelenie kapitoly . . . . .	4
2.2	Optika . . . . .	4
2.3	Objektív . . . . .	9
2.4	Geometrické skreslenie . . . . .	10
2.5	Chromatická aberácia . . . . .	14
2.6	Vinetácia . . . . .	14
2.7	Efekt hĺbky ostrosti . . . . .	16
2.8	Štrukturálne rozostrenie . . . . .	20
2.9	Rozostrenie spôsobené pohybom . . . . .	21
2.10	Odlesky a parazitné svetlo . . . . .	23
2.11	PID regulátor . . . . .	25
<b>3</b>	<b>Zhodnotenie súčasného stavu</b>	<b>28</b>
3.1	Zadanie problému . . . . .	28
3.2	Zhodnotenie stavu vzhľadom na problém . . . . .	29
<b>4</b>	<b>Realizácia</b>	<b>31</b>
4.1	Návrh simulátora . . . . .	31
4.2	Spracovanie vstupných dát . . . . .	31
4.3	Popis simulácie . . . . .	34
4.4	Efekt hĺbky ostrosti a štrukturálne rozostrenie . . . . .	35
4.5	Rozostrenie vplyvom pohybu . . . . .	37
4.6	Odlesky . . . . .	38
4.7	Vinetácia . . . . .	40
4.8	Geometrické skreslenie a chromatická aberácia . . . . .	41
4.9	Mechanizmus clony . . . . .	41
4.10	Výstup . . . . .	43
4.11	Konfiguračný súbor . . . . .	44
<b>5</b>	<b>Záver</b>	<b>46</b>
<b>A</b>	<b>Obsah CD</b>	<b>50</b>

# Kapitola 1

## Úvod

V posledných desaťročiach nastal obrovský technologický vývoj vo výpočtovej technike, ktorá dokáže riešiť mnohé náročné problémy. Jeden z riešených problémov je získavanie informácií z obrazu nasnímaného pomocou fotoaparátu alebo kamery. Odvetvie výpočtovej techniky, ktoré sa zaoberá touto problematikou, sa nazýva počítačové videnie. Využitie algoritmov počítačového videnia je ďalekosiahle a nachádza svoje uplatnenie v odvetviach ako robotika, medicína a v mnohých ďalších.

Algoritmy počítačového videnia využívajú veľké množstvo dát. Tieto dáta sa využívajú v počítačovom videní na učenie rozpoznávania objektov a na testovanie implementácie algoritmov počítačového videnia. Úspešnosť implementácie je výrazne závislá na kvalite snímok, ktoré sú použité na učenie rozpoznávania objektov. Algoritmy počítačového videnia obvykle pracujú so snímkami, ktoré sú nasnímané kamerou alebo fotografickým aparátom. Nasnímaný obraz nie je dokonale presnou kópiou reality, pretože obsahuje optické nedokonalosti, ktoré do snímku vnáša zaznamenávací prístroj. Pôvodcom týchto nedokonalostí môže byť objektív alebo fotocitlivý senzor. Typická optická nedokonalosť zobrazenia, vnášaná do snímku objektívom, je efekt hĺbky ostrosti. Tento efekt sa prejavuje rozostrením objektov v závislosti na ich vzdialenosti od snímacieho zariadenia. Táto nedokonalosť zobrazenia a mnohé ďalšie zhoršujú úspešnosť algoritmov počítačového videnia. Z tohto dôvodu je vhodné o nedokonalostiach objektívu vedieť, a taktiež je vhodné na nedokonalosti obrazu testovať odolnosť aplikácii počítačového videnia.

Vysokú úspešnosť rozpoznávania objektov je možné dosiahnuť učiacimi a testovacími dátami, ktoré sa podobajú dátam, s ktorými bude algoritmus pracovať v reálnom nasadení. Učiace a testovacie dáta pozostávajú zo snímok a z informácií, ktoré popisujú objekty nachádzajúce sa na snímkoch. Vo všeobecnosti platí: čím budú snímky pri vývoji aplikácie obsahovo a kvalitatívne podobnejšie snímkom v reálnom nasadení aplikácie, tým bude dosiahnutá vyššia úspešnosť aplikácie pri rozpoznávaní. Dosiahnuť vysokú podobnosť snímok je v mnohých prípadoch problematické. Taktiež môže byť problematické získavanie snímok určených na učenie algoritmov. Ak by aplikácia mala slúžiť napríklad ako rozpoznávací software na autonómnu navigáciu robota, tak ladenie aplikácie priamo v robotovi by bolo časovo náročné. Druhá možnosť by bola, ladiť aplikáciu v počítači za použitia dát generovaných v počítači. Dáta generované v počítači ale vo všeobecnosti nie sú kvalitatívne podobné dátam, ktoré sa získavajú prostredníctvom kamery v robotovi. Z tohto dôvodu ani generovanie dát v počítači by nemuselo viesť k dobrým výsledkom.

Táto diplomová práca sa zaoberá popisom návrhu aplikácie, ktorej úlohou je vytváranie obrazu s optickými nedokonalosťami tak, ako keby boli vytvorené konkrétnym fotoaparátom alebo video kamerou. K tomuto účelu aplikácia využíva dokonalý 3D obraz scény vygenero-

vaný počítačom. Vyvíjanú aplikáciu je možné využiť všade tam, kde je vyžadovaný fotorealistickej vzhľad počítačom generovaných snímok. Vyvíjaná aplikácia je primárne zameraná na využitie v počítačovom videní, kde umožňuje simulovať optické vlastnosti konkrétneho objektívu. S menšími úpravami uvedeného postupu je možné aplikáciu použiť aj v ďalších oblastiach, napríklad v počítačových hrách, v animovaných filmov a podobne.

V nasledujúcej kapitole tejto práce je uvedený popis dôležitých fyzikálnych javov, ktoré nastávajú v objektíve, a ktoré sú podstatné pre ďalší výklad. Táto kapitola taktiež popisuje niektoré javy a nedokonalosti vo vytváranom obraze. Niektoré z týchto javov je možné simulovať pomocou známych algoritmov. Táto kapitola taktiež uvádza známe algoritmy, ktoré by bolo možné použiť pri vývoji aplikácie, ktorá bude slúžiť ako simulátor vlastností objektívu. Nasledujúca 3 kapitola definuje požiadavky na simulátor a následne analyzuje známe algoritmy s ohľadom na uvedené požiadavky. Posledná kapitola uvádza samotnú realizáciu simulátora. Táto kapitola popisuje návrh implementovaného simulátora, popisuje najdôležitejšie časti simulátora, a taktiež sa venuje popisu použitých algoritmov.

## Kapitola 2

# Zhrnutie súčasného stavu

Táto kapitola má za cieľ uviesť informácie potrebné pre pochopenie návrhu a implementácie simulátora vlastností objektívu. Implementácia sa zaoberá simulovaním mnohých javov, ktoré nastávajú v objektíve. Pre pochopenie jednotlivých javov je nutné pochopiť základné zákonitosti zobrazovania scény pomocou objektívu. Z tohto dôvodu sa úvodná časť tejto kapitoly venuje popisu funkcionality objektívu a venuje sa aj popisu fyzikálnych javov, ktoré nastávajú v objektíve. V ďalších častiach tejto kapitoly je venovaný priestor jednotlivým javom, ako aj známym algoritmom, ktoré umožňujú tieto javy simulovať.

Celá problematika je ďaleko rozsiahlejšia, ako je vyhradený priestor pre túto kapitolu. Z tohto dôvodu sú popisy javov a algoritmov stručné. Rozsiahlejší popis je možné vyhľadať v referenčnej literatúre.

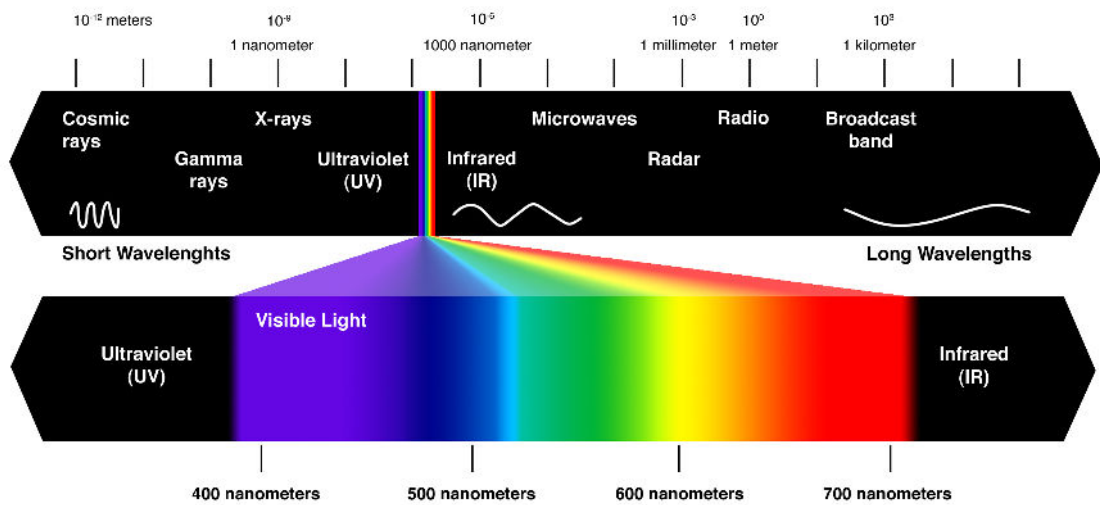
### 2.1 Rozdelenie kapitoly

V úvode tejto kapitoly sú uvedené základné charakteristiky a veličiny, ktoré sa používajú pri popise objektívov a popise javov, ktoré v nich nastávajú. V časti 2.4 je popísané geometrické skreslenie obrazu. Známym prejavom skreslenia je súdkovitost' alebo podúškovitost' obrazu. S touto chybou sa je možné stretnúť aj pri CRT monitoroch. Časť 2.5 popisuje chromatickú aberáciu, ktorú je možné pozorovať v obraze na hranici medzi svetlou a tmavou časťou scény. Na svetlo-tmavej hranici je možné vidieť farebnú kontúru spôsobenú chromatickou aberáciou. V nasledujúcej časti 2.6 je uvedený popis vinetácie, ktorá sa prejavuje stmavovaním obrazu pri jeho okrajoch. Časť 2.7 popisuje efekt hĺbky ostrosti, ktorý sa prejavuje rozostrením objektov v závislosti na vzdialenosti od pozorovateľa. Časť 2.8 popisuje rozostrenie časti obrazu, ktoré je spôsobené nedokonalou konštrukciou objektívu. Časť 2.9 sa venuje popisu rozostrenia obrazu, ktoré spôsobuje pohyb objektov v scéne alebo pohyb samotnej kamery. V časti 2.10 je uvedený popis odleskov, ktoré vznikajú nežiaducim šírením svetla v objektíve. Posledná časť tejto kapitoly sa venuje popisu PID regulátora, ktorý sa je často využívaný v regulačných systémoch. Jeho využitie je možné nájsť pri implementácii mechaniky clony.

### 2.2 Optika

Viditeľná časť elektromagnetického poľa sa nazýva svetlo (obrázok 2.1). Veda zaoberajúca sa vlastnosťami svetla sa nazýva optika. Keďže svetlo tvorí časť z elektromagnetického poľa, zákonitosti platiace v elektromagnetizme platia aj v optike. Z tohto dôvodu sa aj pri svetle

stretávame s javmi ako polarizácia, difrakcia alebo interferencia žiarenia. Ďalšie informácie ohľadom elektromagnetického poľa a jeho zákonitostí je možné vyhľadať v [20].



Obrázok 2.1: Spektrum elektromagnetického poľa; viditeľné svetlo tvorí len malú časť z celkového spektra elektromagnetického žiarenia; prevzaté z [www.vu.union.edu](http://www.vu.union.edu)

Svetelná vlna sa rozptyľuje s narastajúcou vzdialenosťou od zdroja žiarenia. Pri dostatočne vzdialenom zdroji je možné aproximovať dráhu šírenia svetelnej vlny priamkou. V celej práci sa pracuje práve s touto aproximáciou svetelnej vlny. Veda, ktorá sa zaoberá touto problematikou, sa nazýva geometrická optika.

## Svetlo

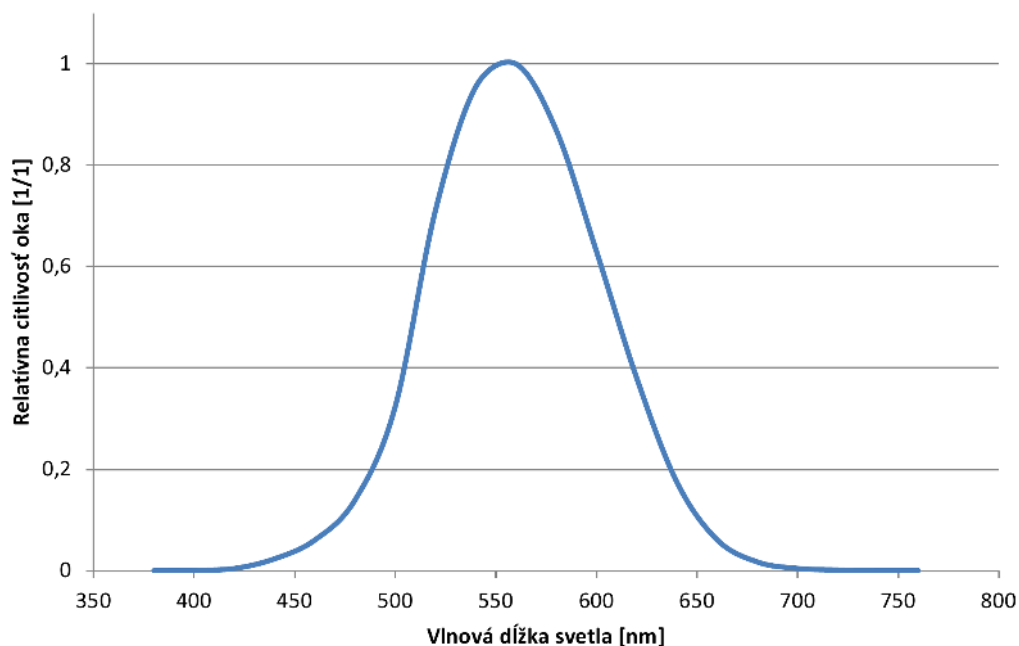
Ľudské oko nedokáže vnímať celý rozsah elektromagnetického žiarenia, dokáže vnímať len jeho malú časť. Obrázok 2.2 demonštruje relatívnu citlivosť oka na elektromagnetické žiarenie v závislosti na vlnovej dĺžke. Oko najcitlivejšie reaguje na žiarenie o vlnovej dĺžke 555 nm. Svetlo o tejto vlnovej dĺžke je vnímané ako zelenožltá farba. Citlivosť oka sa asymptoticky blíži k nule na oboch stranách spektra. Z tohto dôvodu neexistujú presné hranice citlivosti oka. Ak prijmemme za hranice viditeľnosti vlnové dĺžky, pri ktorých klesne relatívna citlivosť oka pod 1%, dostaneme interval 430 nm až 690 nm. [20]

## Odraz a lom svetla

Ak svetlo dopadne na rozhranie dvoch opticky rôznych prostredí, nastáva jeho odraz a lom. Odraz nastáva, keď dopadajúce svetlo sa po dopade bude šíriť v pôvodnom prostredí. V tomto prípade platí, že uhly dopadajúceho a odrazeného svetla sú rovnaké. Dopadajúce a odrazené svetlo sa šíri v rovine kolmej na rozhranie. Táto situácia je znázornená na obrázku 2.3. Dopadajúci a odrazený lúč svetla reprezentujú úsečky pri uhloch  $\alpha$  a  $\alpha'$ . [8, 20]

Lom svetla je ďalší jav, ktorý nastáva pri dopade svetla na rozhranie dvoch prostredí. Pri tomto jave časť svetlo prenikne do druhého prostredia a bude sa v ňom šíriť. Tento jav je znázornený na obrázku 2.3. Svetlo pred lomom a po lome reprezentujú úsečky pri uhloch





Obrázok 2.2: Relatívna citlivosť oka v závislosti na vlnovej dĺžke svetla

$\alpha$  a  $\beta$ . Pre tieto uhly sa uplatňuje Snellov zákon lomu, ktorý je v tvare:

$$n_1 \cdot \sin(\alpha) = n_2 \cdot \sin(\beta), \quad (2.1)$$

kde  $n_1$  a  $n_2$  sú indexy lomu svetla. Symboly  $n_1$  a  $n_2$  sú bezrozmerné veličiny, ktoré charakterizujú prostredie, v ktorom sa svetlo šíri. Pre index lomu platí:

$$n = \frac{c}{v}, \quad (2.2)$$

kde  $n$  predstavuje index lomu,  $c$  rýchlosť šírenia svetla vo vákuu ( $2.998 \cdot 10^8 m/s$ ) a  $v$  rýchlosť šírenia svetla v danom prostredí. Nasledujúca tabuľka obsahuje hodnoty indexov lomu pre svetlo o vlnovej dĺžke 589 nm:

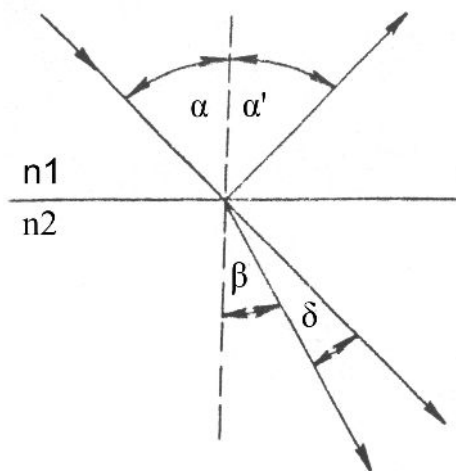
Prostredie	Index lomu
vákuum	1,00
vzduch <sup>1</sup>	1,00029
voda (20°)	1,33
typické sklo	1,52
diamant	2,42

<sup>1</sup> - za štandardných podmienok

Informácie v tejto časti boli čerpané z [20].

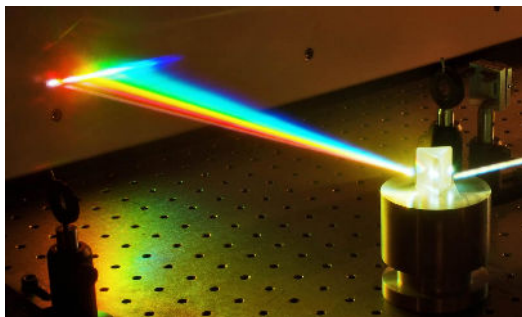
### Chromatická disperzia

Index lomu prostredia závisí na vlnovej dĺžke svetla (platí pre všetky prostredia okrem vákuua). Táto závislosť spôsobí, že zväzok svetla skladajúci sa zo žiarenia o rôznej vlnovej

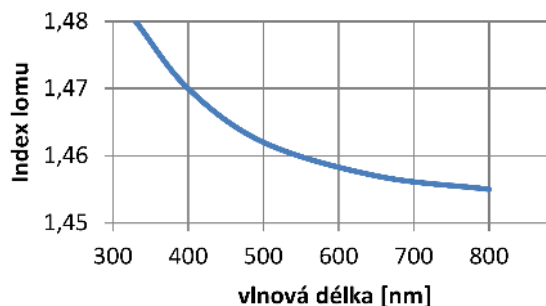


Obrázok 2.3: Odraz a lom svetla; prevzaté z wikipedia.org

dĺžke, sa pri lome rozptýli (obrázok 2.4). Tento jav sa nazýva chromatická disperzia. Graf na obrázku 2.5 demonštruje závislosť indexu lomu na vlnovej dĺžke. Chromatická disperzia je príčinou chromatickej aberácie (jav prejavujúci sa na fotografiách, ďalšie informácie o tomto jave sa nachádzajú v 2.5). [20]



Obrázok 2.4: Rozklad svetla na hranole; prevzaté z thorium.at

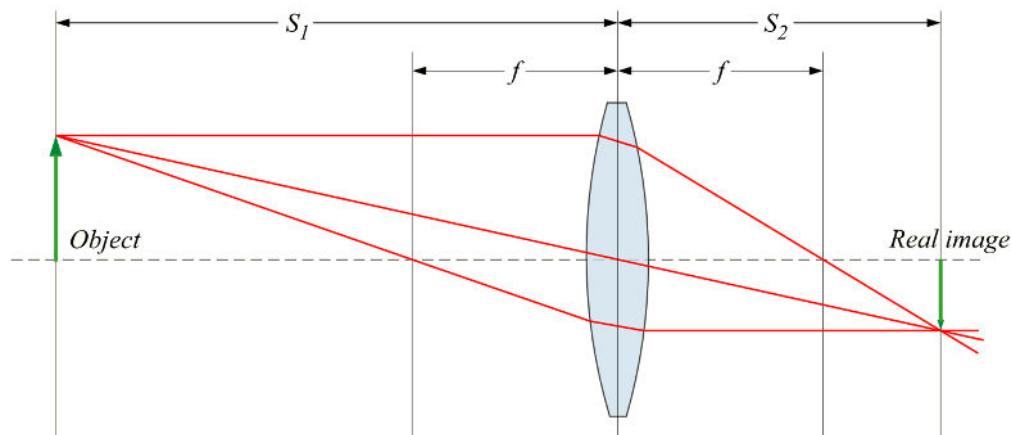


Obrázok 2.5: Indexu lomu taveného kremeňa v závislosti na vlnovej dĺžke

### Tenká šošovka

Šošovka je priehľadné homogénne prostredie s indexom lomu vždy väčším ako jedna. Pri dopade svetla na povrch šošovky sa časť svetla láme podľa Snellovho zákonu (rovnica 2.1). Následne sa toto svetlo šíri v šošovke a opätovne sa šíri pri prieniku zo šošovky do vonkajšieho prostredia. Schéma šírenia svetla šošovkou sa nachádza na obrázku 2.6. Na obrázku je názorne demonštrované vytváranie obrazu predmetu za šošovkou. Podľa spôsobu vytvárania obrazu je možné rozdeliť šošovky na spojné a rozptylné. Spojné šošovky vytvárajú obraz predmetu umiestneného v nekonečne veľkej vzdialenosti za šošovkou, rozptylné pred šošovkou. Na obrázku 2.6 sa nachádza spojná šošovka, ktorá vytvára obraz za šošovkou.

Vzdialenosť obrazu od šošovky je  $S_2$ .



Obrázok 2.6: Ohnisková vzdialenosť; prevzaté z wikipedia.org

Ak sa obraz predmetu vytvorí za šošovkou, šošovka vytvorí reálny obraz. Ak pred šošovkou, vzniknutý obraz bude neskutočný. V tejto súvislosti sa ustanovila znamienková konvencia, ktorá pre oblasť pred šošovkou používa kladnú hodnotu predmetovej vzdialenosti a pre oblasť za šošovkou, zápornú predmetovú vzdialenosť. Obrazová vzdialenosť pri tejto konvencii je záporná pre oblasť pred šošovkou a kladná pre oblasť za šošovkou. [20]

### Ohnisková vzdialenosť

Ohnisková vzdialenosť je veličina popisujúca správanie šošovky. Pre ohniskovú vzdialenosť platí vzťah:

$$\frac{1}{f} = \frac{1}{S_1} + \frac{1}{S_2}, \quad (2.3)$$

kde  $S_1$  predstavuje obrazovú vzdialenosť,  $S_2$  je predmetová vzdialenosť a  $f$  predstavuje ohniskovú vzdialenosť. Jednotlivé veličiny sú znázornené na obrázku 2.6.

Ohnisková vzdialenosť je veličina, ktorá našla svoje uplatnenie aj pri popise objektívov. Čím je ohnisková vzdialenosť objektívu menšia, tým objektív zachycuje väčšie zorné pole. Ohnisková vzdialenosť blízka vnímaniu ľudského oka je 48mm. Hoci je objektív zložený z väčšieho množstva šošoviek (prípadne iných optických prvkov), správa sa ako spojná šošovka. Z tohto dôvodu zákonitosti platiace pre jednu spojnú šošovku platia s určitými obmedzeniami aj pre objektív. [20]

### Priečne zväčšenie

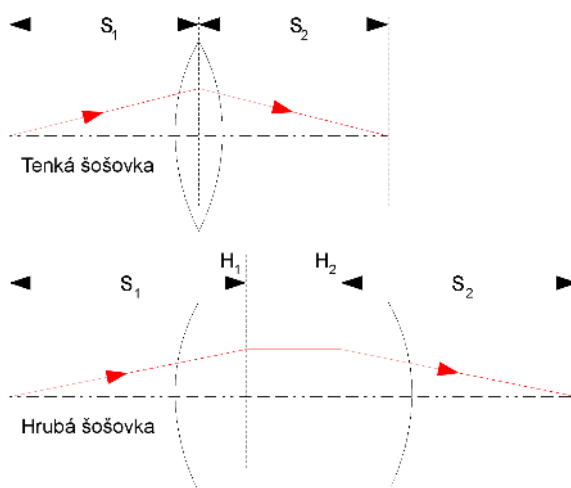
Priečne zväčšenie šošovky je veličina, ktorá vyjadruje koľkokrát sa obraz zväčší oproti reálnemu objektu. Pre priečne zväčšenie platí vzťah:

$$Z = \frac{\text{obraz}}{\text{objekt}} = -\frac{S_2}{S_1} = -\frac{S_2 - f}{f} = -\frac{f}{S_1 - f}, \quad (2.4)$$

kde  $Z$  je priečne zväčšenie,  $S_1$  predstavuje objektovú vzdialenosť,  $S_2$  obrazovú vzdialenosť a  $f$  je ohnisková vzdialenosť. Tieto veličiny sú znázornené na obrázku 2.6. [20]

## 2.3 Objektív

Objektív je sústava niekoľkých šošoviek so spoločnou optickou osou. Objektív ako celok je možné považovať za optickú sústavu alebo optický systém. Pre celý systém platia podobné zákonitosti a podobné vzťahy ako pre hrubú spojnú šošovku. Pri hrubej šošovke sa predmetová a obrazová vzdialenosť meria od hlavných bodov. Roviny, v ktorých ležia hlavné body a ktoré sú kolmé k optickej ose, sa nazývajú hlavné roviny. Na obrázku 2.7 sú znázornené hlavné roviny hrubej šošovky. [7]



Obrázok 2.7: Schéma premietania obrazu pri tenkej a hrubej šošovke; pri tenkej šošovke sú hlavné roviny identické; pri hrubej šošovke sú hlavné roviny premietania rôzne; v schéme sú obe hlavné roviny označené symbolmi  $H_1$  a  $H_2$

### Svetelnosť objektívu

Množstvo svetla vchádzajúceho do objektívu je obmedzené priemerom vstupnej pupily objektívu. Čím je priemer pupily menší a čím je pupila vzdialenejšia od roviny premietania, tým sa menšie množstvo svetla bude podieľať na tvorbe obrazu. Toto správanie definuje parameter svetelnosť objektívu, pre ktorý platí vzťah:

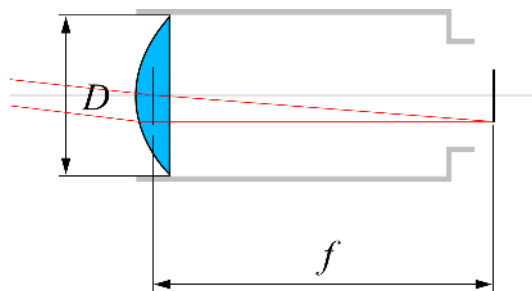
$$b = \frac{f}{D}, \quad (2.5)$$

kde  $b$  je svetelnosť objektívu,  $f$  je ohnisková vzdialenosť a  $D$  je priemer vstupnej pupily (obrázok 2.8). Čím je svetelnosť objektívu väčšia, tým väčšie množstvo svetla sa môže podieľať na tvorbe obrazu. [7]

### Clonové číslo

V objektíve sa nachádza irisová clona, ktorá umožňuje regulovať množstvo svetla, ktoré prechádza cez objektív. Zmena veľkosti clony má za následok správanie podobné zmenšovaniu vstupnej pupily objektívu. Z tohto dôvodu veľkosť clonového čísla vyjadruje podobný vzťah výpočtu svetelnosti objektívu:

$$b = \frac{f}{D}, \quad (2.6)$$



Obrázok 2.8: Prierez zjednodušeným objektívom, prevzaté z wikipedia.org

kde  $b$  je clonové číslo,  $f$  je ohnisková vzdialenosť a  $D$  je priemer clony (ak by bola umiestnená v ohniskovej vzdialenosti a umožnila by prepustiť rovnaké množstvo svetla ku senzoru). Táto situácia je znázornená na obrázku 2.8. Svetelnosť objektívu vyjadruje najväčšie možné clonové číslo, ktoré je možné nastaviť v danom objektíve. [7]

Intenzita výstupného svetla klesá s druhou mocninou hodnoty clonového čísla. Z tejto zákonitosti je možné odvodiť radu clonových čísel:

1/1	1/1,41	1/2	1/2,83	1/4	1/5,66	1/8	...
-----	--------	-----	--------	-----	--------	-----	-----

v ktorej platí, že konkrétny člen (konkrétne clonové číslo) z postupnosti prepustí dvojnásobné množstvo svetla, ako nasledujúci člen postupnosti. [7]

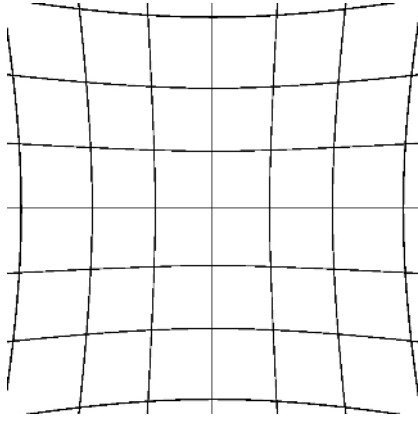
## 2.4 Geometrické skreslenie

Fotoaparát alebo kamera vytvára 2D-obraz reálneho sveta. Ideálny obraz, vytvorený ideálnou optickou sústavou, bude mať v celom svojom obraze rovnaké priečne zväčšenie. V praxi sa stretávame s objektívmi, pri ktorých toto pravidlo neplatí. Rôzne priečne zväčšenie obrazu spôsobí, že rovné čiary v scéne sa do obrazu môžu premietnuť ako krivky - vytváraný obraz bude zdeformovaný. Toto zdeformovanie sa nazýva geometrické skreslenie. Tento jav je možné dobre pozorovať na snímkoch, ktoré zachycujú pravidelnú mriežku v scéne. Na snímku sa môže pravidelná mriežka zobraziť zdeformovane. Existujú dva základné druhy zdeformovania - podúškovitosť (obrázok 2.9) alebo súdkovitosť (obrázok 2.10). V praxi sa je možné stretnúť aj s kombináciou oboch skreslení. Presná podoba skreslenia je daná konštrukciou objektívu.

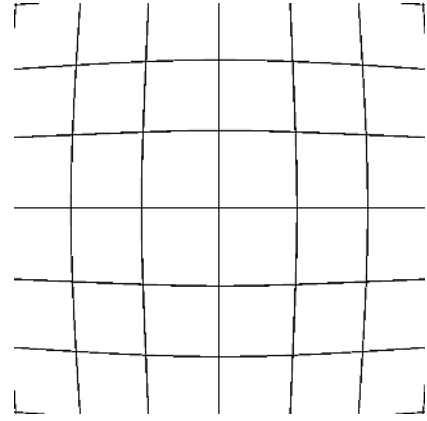
Tento jav je dobre známy a spôsobuje problémy algoritmom počítačového videnia. Z tohto dôvodu existuje veľa informácií o tomto jave, o jeho detekcii a aj o jeho odstránení. Popis tohto javu je možné nájsť napríklad v [22]. Popis detekcie skreslenia a jej parametrizáciu je možné nájsť v článkoch [21, 16]. Odstránenie geometrického skreslenia je popísané v článku [33]. V tomto článku sa na odstránenie skreslenia využíva koncept sieťového warpingu [10] a na premapovanie pixelov sa používa Lanczosov filter [17].

### Parametrizácia

Existuje viacero prístupov popisu geometrického skreslenia. Jeden z prístupov popisuje skreslenie pomocou deformačnej mriežky, kde vstup tvoria vrcholy mriežky. Korekcia tohto skreslenia prebieha aplikáciou sieťového warpingu na vstupný obraz. Druhý prístup predstavuje model radiálne symetrického skreslenia, kde celé skreslenie je popísané pomocou 6-7



Obrázok 2.9: Podúškovitost', prevzaté z wikipedia.org



Obrázok 2.10: Súdkovitost', prevzaté z wikipedia.org

parametrov. Tento model v prvom kroku rekonfiguruje pixely vstupného snímku pomocou lineárnej transformačnej matice a v druhom kroku sa následne odstráni radiálne symetrické skreslenie.

### Model radiálne-symetrického skreslenia

Radiálne symetrické skreslenie je možné popísať:

$$r_d = r + \delta_r, \quad (2.7)$$

kde  $r_d$  je vzdialenosť bodu od stredu optickej sústavy v obraze so skreslením,  $r$  predstavuje vzdialenosť bodu od stredu sústavy po odstránení skreslenia a  $\delta_r$  je koeficient radiálneho skreslenia závislého na  $r$ .

V praxi sa používa model, ktorý dokáže spracovať aj ďalšie optické vady. Odvodený model umožňuje korekciu nesprávneho pomeru strán, korekciu skosenia a upravuje pozíciu obrazu. Toto je možné dosiahnuť pomocou transformačnej matice. Transformačná matica sa používa v nasledujúcej podobe:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X^c \\ Y^c \\ 1 \end{bmatrix} \quad (2.8)$$

kde  $(u, v)$  predstavujú body po transformácii,  $(\alpha, \beta)$  sú koeficienty zväčšenia,  $\gamma$  definuje skosenie,  $(u_0, v_0)$  sú súradnice stredu radiálneho skreslenia a  $(x, y)$  predstavuje súradnice pixelu vo vstupnom snímku.

Následne takto modifikovaný obraz sa transformuje pomocou funkcie odstraňujúcej radiálne symetrické skreslenie. Táto funkcia má obvykle tvar:

$$r_d = rf(r) = r(1 + k_1r^2 + k_2r^4 + k_3r^6 + \dots), \quad (2.9)$$

kde  $k_1, k_2, k_3, \dots$  sú koeficienty skreslenia,  $r_d$  je polomer výstupnej pozície pixelu a  $r$  je polomer vstupnej pozície pixelu. V reálnych aplikáciach sa bežne používa len jeden alebo dva členy, ostatné členy sú nulové. V článku [16] sú uvedené dve varianty vzorca 2.9, ktoré sa používajú v praxi. Jedna varianta má nenulový koeficient  $k_1$  (ostatné sú nulové) a druhá

varianta používa koeficienty  $k_1$  a  $k_2$ . V uvedenom článku sa zavádza nová podoba funkcie  $f(r)$ , ktorá má viaceré pozitívne vlastnosti v porovnaní s funkciami v tvare 2.9. Táto novo zavedená funkcia má tvar:

$$r_d = rf(r) = r(1 + l_1r + l_2r^2). \quad (2.10)$$

Článok [16] sa taktiež zaoberá odvodením vzťahu, ktorý umožňuje priamu transformáciu pozície pixelov v súradnej sieti. Tento vzťah vychádza zo vzťahov:

$$x_d = xf(r), \quad y_d = yf(r), \quad (2.11)$$

kde  $(x, y)$  a  $(x_d, y_d)$  sú súradnice pozície pred a po transformácii a  $r$  predstavuje hodnotu:

$$r = \sqrt{x^2 + y^2} \quad (2.12)$$

Použitím vzťahov 2.7 a 2.11 je možné získať vzťah:

$$u = \alpha xf(r) + \gamma yf(r) + u_0, \quad v = \beta yf(r) + v_0, \quad (2.13)$$

kde funkcia  $f(r)$  predstavuje funkciu radiálneho skreslenia a  $\alpha, \beta, \gamma, u_0, v_0$  sú koeficienty lineárneho skreslenia.

## Sieťový warping

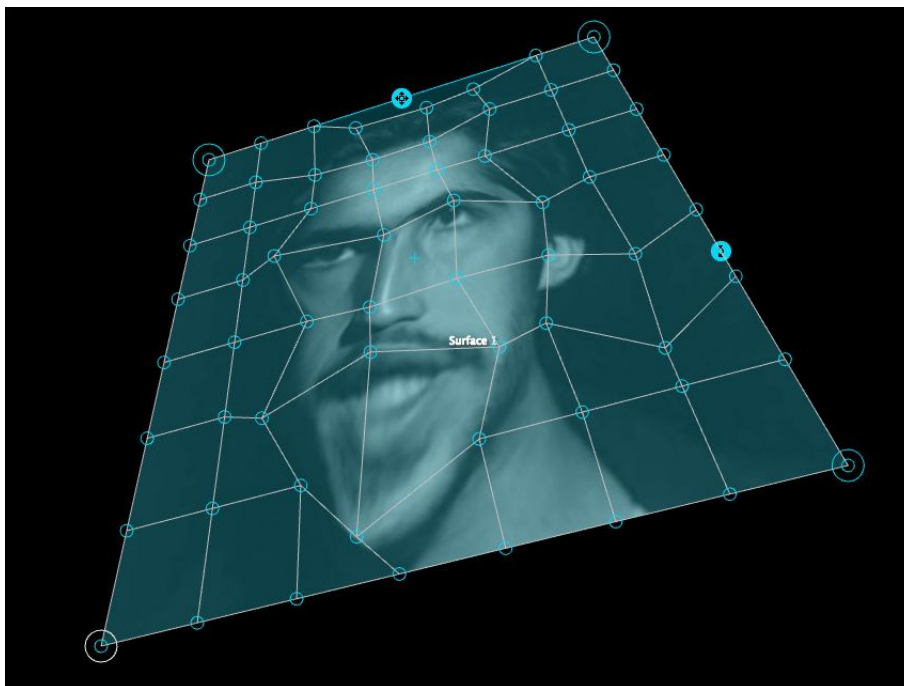
Warping je proces, ktorý umožňuje kontrolovane manipulovať s proporciami obrázku. Sieťový warping sa definuje pomocou mriežky, ktorá sa umiestní na obrázok. Mriežku môžu tvoriť lomené čiary, spline atď. Mriežka rozdelí obrázok na jednotlivé oblasti. Následne užívateľ zmení tvar mriežky buď pomocou modifikácie priesečníkov mriežky alebo zmenou parametrov kriviek. Po zmene tvaru mriežky sa opätovne zistia oblasti ohraničené mriežkou. Algoritmus warpingu premapuje pixely tak, aby obsah oblastí boli pred úpravou a po úprave rovnaké. Zmena mriežky nezmení obsah oblastí, zmení len tvar oblastí. Ukážka warpingu v interaktívnej aplikácii je znázornená na obrázku 2.11.

Algoritmus sieťového warpingu je separabilný pre riadky a stĺpce. Z tohto dôvodu algoritmus prebieha v dvoch fázach. V prvej fáze je prevedená deformácia jednotlivých riadkov. Týmto procesom sa získa medziobraz. V druhej fáze sa v medziobraze zdeformujú jednotlivé stĺpce. Týmto postupom sa získa výstupný obrázok.

Deformácia prebieha nezávisle na jednotlivých riadkoch. Algoritmus načíta všetky pixely v danom riadku. Následne zistí priesečníky riadku s vertikálnymi krivkami (vstupnej) mriežky. Týmto procesom sa riadok rozdelí na jednotlivé oblasti. V ďalšom kroku algoritmus zistí rozdelenie riadku na oblasti v pozmenenej mriežke. Pixely sa premapujú tak, aby obsah jednotlivých oblastí, pred úpravou a po úprave mriežky ostal rovnaký. Pri takomto premapovaní obvykle platí, že oblasti pred a po modifikácii mriežky majú rôzne rozmery. Z tohto dôvodu je nutné použiť niektorú z interpolačných metód na prepočítanie hodnôt pixelov. Bežne sa používa interpolácia najbližším susedom alebo lineárna interpolácia. V článku [33] je uvedená interpolačná metóda, ktorá používa Lanczosov filter. Tento filter pri malých deformáciach poskytuje veľmi kvalitný výstup.

Vyššie uvedeným postupom sa získa medziobraz s deformovanými riadkami. Následne sa obdobným postupom sa zdeformujú v medziobraze stĺpce, čím sa získa výsledný zdeformovaný (skreslený) obraz.

Informácie v tomto odseku boli čerpané z knihy Moderní počítačová grafika [10].



Obrázok 2.11: Sieťový warping v programe Madmapper, prevzaté z [www.madmapper.com](http://www.madmapper.com)

### Lanczosov filter

Lanczosove filtre sú FIR filtre, ktoré je možné zapísať pomocou vzťahu:

$$\text{LanFilter}_N(\tau) = \text{Lanczos}_N(\tau) \text{sinc}(\tau), \quad (2.14)$$

kde funkcia *sinc* definuje okno a  $\text{Lanczos}_N$  predstavuje konvolučné jadro, ktoré je v tvare:

$$\text{Lanczos}_N(\tau) = \begin{cases} \text{sinc}(\tau) & -N \leq \tau \leq N \\ 0 & \text{inak} \end{cases} \quad (2.15)$$

V praktických aplikáciach sa používa  $N = 2$  a  $N = 3$ . Funkcia *sinc* má tvar:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}, \quad (2.16)$$

Filtrácia signálu pomocou Lanczosovho filtru prebieha pomocou nasledujúceho vzťahu:

$$\hat{I}(x) = \sum_{i=-\infty}^{\infty} \text{LanFilter}_N(x - n) I(n), \quad (2.17)$$

kde  $\hat{I}$  predstavuje vstupný signál a  $I$  výstupný signál. Lanczosov filter je možné rozšíriť do podoby filtru  $L(x, y)$ , ktorý umožňuje filtrovať dvojrozmerný signál. Dvojrozmerný filter je separabilný, z tohto dôvodu platí:

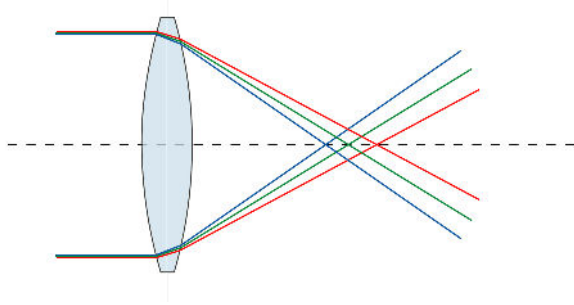
$$L(x, y) = L(x) \cdot L(y), \quad (2.18)$$

kde  $L(x)$  a  $L(y)$  sú jednorozmerné filtre filtrujúce signál v smere osi  $x$ , resp.  $y$ . Všetky uvedené informácie boli čerpané z [17].

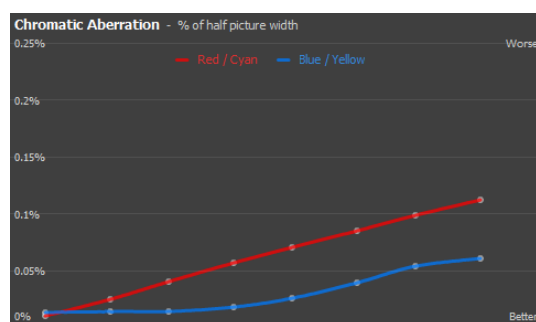


## 2.5 Chromatická aberácia

Chromatická aberácia je farebná chyba šošovky alebo objektívu, ktorá je spôsobená chromatickou disperziou. Objektívy svojou konštrukciou redukujú prejav chromatickej disperzie, napriek tomu sa stále prejavuje. Jej prejav je možné pozorovať na hrane medzi jasnou a tmavou časťou obrazu na perifériách snímku. Na hranách medzi jasnou a tmavou časťou obrazu je možné pozorovať farebnú kontúru (obrázok 2.14). Tvorba farebných kontúr je spôsobená tým, že objektív nie je schopný zaostriť svetlo o rôznej vlnovej dĺžky do jednej zaostrenej roviny (tento typ aberácie sa nazýva axiálna chromatická aberácia, schématicky je znázornená na obrázku 2.12) alebo vytváraný obraz nemá rovnaké priečne zväčšenie pre rôzne vlnové dĺžky svetla (laterálna chromatická aberácia). V objektívoch vo všeobecnosti platí, že chromatická aberácia sa výraznejšie prejavuje na perifériách snímku (obrázok 2.13). [26]



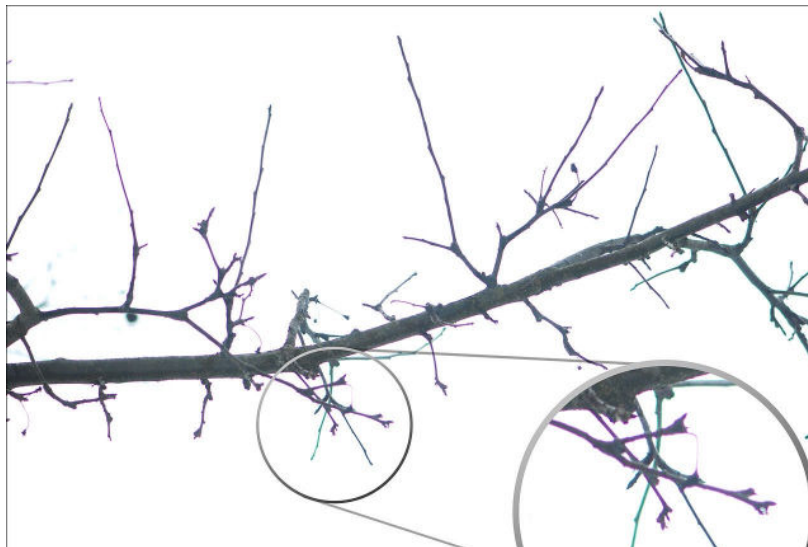
Obrázok 2.12: Axiálna chromatická aberácia na spojnej šošovke, prevzaté z wikipedia.org



Obrázok 2.13: Charakteristika chromatickej aberácie Olympusu M. ZD 14-150mm; grafy vyjadrujú posun farebných zložiek v závislosti na polohe v obraze; prevzaté z www.dpreview.com

## 2.6 Vinetácia

Vinetácia je redukcia jasnosti obrazu na perifériách obrazu. Tento jav spôsobujú mechanické prekážky ako napríklad obruba objektívu, nevhodná slnečná clona atď. Ďalším pôvodcom tohto javu je prirodzená vlastnosť šošovky, pri ktorej platí, že jas obrazu klesá so štvrtou mocninou kosinu uhla dodajúceho svetla. Ak by širokouhlý objektív (jeho zorné pole je väčšie ako  $30^\circ$ ) bol tvorený len jednou šošovkou, útlm intenzity svetla by bol minimálne 44%. Vinetácia môže byť spôsobená aj použitým fotocitlivým senzorom v digitálnych fotoaparátach alebo kamerách. Svetlo dopadajúce kolmo na povrch senzoru produkuje vyšší výstupný signál ako rovnaké množstvo svetla dopadajúce pod nejakým uhlom. Vinetácia sa občas využíva v umeleckej fotografii ako prostriedok na nasmerovanie pohľadu diváka. Hlavný motív v tomto prípade býva výrazne jasnejší ako 'nezaujímavé' časti obrazu, a preto upúta pozornosť diváka (obrázok 2.15). Z tohto dôvodu existujú optické filtre, ktoré umelo vytvárajú vinetáciu. Taktiež je možné nájsť aplikácie, ktoré umožňujú pridávať vinetáciu už do existujúcich snímok pomocou post-processingových filtrov. [25]



Obrázok 2.14: Snímok s prejavom chromatickej aberácie, prevzaté z [www.wikipedia.org](http://www.wikipedia.org)



Obrázok 2.15: Vinetácia v obraze; prevzaté z [www.buzzillions.com](http://www.buzzillions.com)



Obrázok 2.16: Efekt hĺbky ostrosti; objekt v blízkosti zaostrenej roviny je ostrý; ostatné objekty sú tým viac rozmazané, čím sú ďalej od zaostrenej roviny; prevzaté z [www.webdesignbooth.com](http://www.webdesignbooth.com)

## 2.7 Efekt hĺbky ostrosti

Efekt hĺbky ostrosti je možné popísať ako rozostrovanie obrazu v závislosti na vzdialenosti objektov v scéne od pozorovateľa (od kamery). Kamera je teoreticky schopná dokonale zaostriť na objekty, ktoré sa nachádzajú sa v zaostrenej rovine. Body nachádzajúce sa v tejto rovine sa premietnu na fotocitlivý materiál ako jeden bod. Objekty nachádzajúce sa mimo tejto roviny sa premietnu na senzor ako plocha. Pre vytváranú plochu bude platiť, že čím je objekt vzdialenejší od zaostrenej roviny, tým bude táto plocha väčšia. Tento jav je demonštrovaný na obrázku 2.16.

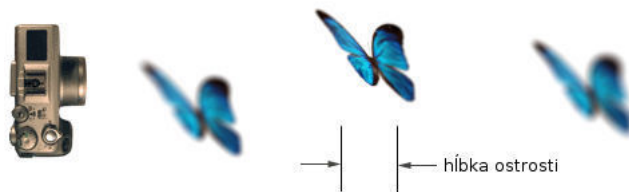
### Hĺbka ostrosti

Hĺbka ostrosti je vzdialenosť medzi najbližším a najvzdialenejším objektom v scéne, ktoré sa javia pri pozorovaní výsledného snímku ako dostatočne ostré (obrázok 2.17). Pojem ostrosti vychádza z rozpoznávacej schopnosti zdravého ľudského oka. Zdravé ľudské oko je schopné rozpoznať 2 body, ktoré sa nachádzajú v uhlovej vzdialenosti minimálne  $1'$ . Ak je uhlová vzdialenosť menšia, tieto body bude oko vnímať ako jeden bod. Oblasť vymedzená  $1'$  sa nazýva rozptylový krúžok. Bežná vzdialenosť pozorovania snímok je  $25\text{ cm}$ . V tejto vzdialenosti bude polomer rozptylový krúžok približne  $1/6\text{mm}$  (na fotografii). Predpokladajme, že sú tieto snímky získané fotoaparátom, ktorý sníma obraz o rozmeroch kinofilmového políčka. V tomto prípade je polomer rozptylový krúžok na sensore fotoaparátu  $1/30\text{mm}$ . V tomto prípade hĺbka ostrosti definuje vzdialenosť medzi najvzdialenejším a najbližším bodom v priestore, ktoré sa premietnú na senzor ako krúžok s polomerom  $1/30\text{mm}$ .

### Hyperfokálna vzdialenosť

Hyperfokálna vzdialenosť je najmenšia zaostrená vzdialenosť, v rámci ktorej bude hĺbka ostrosti rovná nekonečnu. Tento prípad nastane, ak budú ostré všetky objekty za zaostrenou vzdialenosťou. Pre hyperfokálnu vzdialenosť platí:

$$H = \frac{f^2}{Nc} + f, \quad (2.19)$$



Obrázok 2.17: Hĺbka ostrosti; obrázok demonštruje rozsah hĺbky ostrosti v scéne; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)

kde  $H$  je hyperfokálna vzdialenosť,  $f$  je ohnisková vzdialenosť,  $N$  predstavuje f-číslo (clonové číslo) a  $c$  je rozptylový krúžok. V praxi sa používa aproximácia hyperfokálnej vzdialenosti:

$$H \approx \frac{f^2}{Nc}. \quad (2.20)$$

### Výpočet hĺbky ostrosti

Najmenšia vzdialenosť dostatočne ostrých objektov sa vypočíta pomocou vzťahu:

$$D_N \approx \frac{Hs}{H+s}, \quad (2.21)$$

kde  $D_N$  je počítaná vzdialenosť,  $H$  je hyperfokálna vzdialenosť a  $s$  predstavuje zaostrenú vzdialenosť. Pre najväčšiu vzdialenosť ostrých objektov platí:

$$D_F \approx \frac{Hs}{H-s}. \quad (2.22)$$

Hĺbka ostrosti sa vypočíta ako rozdiel najväčšej a najmenšej vzdialenosti:

$$DOF \approx \frac{2Hs^2}{H^2 - s^2} \quad \text{pre } s < H. \quad (2.23)$$

Dosadením výpočtu fokálnej vzdialenosti do predchádzajúcej rovnice dostaneme:

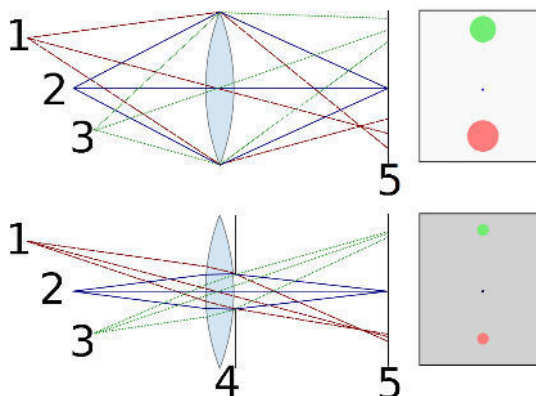
$$DOF \approx \frac{2Ncf^2s^2}{f^4 - N^2c^2s^2}. \quad (2.24)$$

Z rovnice 2.24 vyplýva, že hĺbka ostrosti závisí od clonového čísla, rozptylový krúžok, ohniskovej vzdialenosti a zaostrenej vzdialenosti. Vyššie uvedené vzťahy platia v prípade, že platí  $f \ll s$ .

### Výpočet rozostrenia

Predpokladajme, že zaostrená rovina je vo vzdialenosti  $s$  od pozorovateľa a objekt sa nachádza vo vzdialenosti  $D$  od pozorovateľa. V tomto prípade bude platiť, že objekt bude od zaostrenej roviny vzdialený:

$$x_d = |D - s|. \quad (2.25)$$



Obrázok 2.18: Rozostrenie bodov v závislosti na ich vzdialenosti a veľkosti clony; 1 - bod za zaostrenou rovinou; 2 - bod ležiaci v zaostrenej rovine ; 3 - bod pred zaostrenou rovinou; 4 - irisová clona; 5 - fotocitlivá vrstva; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)

Bod v predmetovom priestore sa po prechode optickou sústavou premietne na fotocitlivú vrstvu na senzore ako krúžok (obrázok 4.5). Priemer tohto krúžku je možné vypočítať pomocou vzťahu:

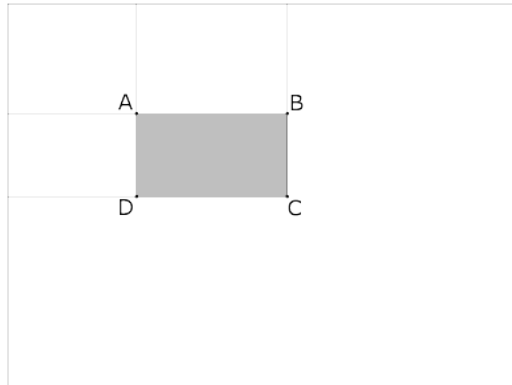
$$b = \frac{f m_s}{N} \frac{x_d}{s \pm x_d}, \quad (2.26)$$

kde  $f$  predstavuje ohniskovú vzdialenosť,  $m_s$  priečne zväčšenie pre zaostrenie na vzdialenosť  $s$ ,  $N$  clonové číslo a  $s$  je zaostrená vzdialenosť. Ak je počítané rozostrenie pre predmety pred zaostrenou rovinou (majú menšiu vzdialenosť od pozorovateľa ako zaostrená vzdialenosť), tak je použité záporné znamienko v menovateli. Kladné znamienko sa použije, ak sa počíta rozptylný krúžok pre predmety za rovinou. Vzťah 2.4 definuje výpočet priečného zväčšenia pre známu predmetovú vzdialenosť a známu ohniskovú vzdialenosť.

## Úvod k algoritmom

Algoritmy simulujúce efekt hĺbky ostrosti, ďalej len algoritmy DoF, je možné rozdeliť do dvoch skupín. Prvá skupina pracuje v objektovom priestore scény. Typickým predstaviteľom tejto skupiny je algoritmus distribuovaného ray tracingu [3]. Algoritmy tejto skupiny vyžadujú modifikáciu rendera, resp. generátoru snímok. Druhú skupinu tvoria algoritmy DoF pracujúce v obrazovom priestore. Tieto algoritmy dostávajú na vstup farebný snímok a informáciu o vzdialenosti bodov od pozorovateľa (hĺbkovú mapu). Na základe týchto informácií DoF algoritmy simulujú snímok s efektom hĺbky ostrosti. Algoritmy tejto skupiny pracujú s výstupom renderera, a preto je ich možné zaradiť medzi post-processové filtre. Tieto algoritmy nemajú informácie o celkovej geometrii scény, z tohto dôvodu nedokážu zobrazit' DoF presne. Ich výstup je zaťažený artefaktmi. [1]

Problematika simulácie efektu hĺbky ostrosti je dobre preskúmaná a venuje sa jej relatívne mnoho publikácií. Prehľad niektorých existujúcich algoritmov je možné vyhľadať v [5, 14, 1]. Existuje veľké množstvo algoritmov, z tohto dôvodu sa budeme venovať v ďalších sekciách len tým, ktoré je vhodné použiť pri implementácii simulátora.



Obrázok 2.19: Integrálny obraz, prevzaté z www.wikipedia.org

### Efekt hĺbky ostrosti založený na lineárnom FIR filtri s variabilným jadrom

Lineárny FIR filter s variabilným filtrom v dvojrozmernom priestore filtruje vstupný obrázok pomocou nasledujúceho vzťahu:

$$B(x, y) = \sum_i \sum_j k(x, y, i, j) S(i, j), \quad (2.27)$$

kde  $B$  je výstupná signál,  $S$  je vstupný signál a  $k$  predstavuje variabilné jadro filtru jadro filtru.[1]

V publikácii [5] je uvedený algoritmus efektu hĺbky ostrosti, ktorý je založený práve na vyššie uvedenom filtri. Jeho názov je Algoritmus Reverse-Mapped Z-Buffer Depth of Field Effect. Pre každý pixel výstupného snímku sa na základe hĺbky pixelu vypočíta veľkosť rozostrovacieho krúžku. Táto hĺbka definuje oblasť vo vstupnom snímku, ktorej priemerná hodnota bude hodnotou vo výstupnom snímku.

### Integrálny obraz

Integrálny obraz  $I_\Sigma$  je spôsob reprezentácie obrazu, ktorý sa vytvorí zo štandardného snímku  $I$  aplikovaním nasledujúceho vzťahu:

$$I_\Sigma(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j). \quad (2.28)$$

Táto reprezentácia umožňuje vypočítať priemernú hodnotu ľubovoľne veľkej obdĺžnikovej oblasti (obrázok 2.19) v konštantnom čase. Toto je možné dosiahnuť aplikovaním nasledujúceho vzťahu:

$$\sum_{i=a_x+1}^{c_x} \sum_{j=a_y+1}^{c_y} (i, j) = I_\Sigma(a_x, a_y) - I_\Sigma(b_x, b_y) + I_\Sigma(c_x, c_y) - I_\Sigma(d_x, d_y), \quad (2.29)$$

kde  $A = (a_x, a_y)$ ,  $B = (b_x, b_y)$ ,  $C = (c_x, c_y)$  a  $D = (d_x, d_y)$ .  $A, B, C, D$  sú súradnice vrcholov obdĺžnika a ich umiestnenie je znázornené na obrázku 2.19. [4]

## Efekt hĺbky ostrosti založený na difúzii

Tento algoritmus vychádza z princípu difúzie tepla v uzavretom heterogénnom prostredí. Princíp tohto algoritmu je najjednoduchšie vysvetliť v analógii s difúziou tepla. Farebná obrazová časť snímku predstavuje teplo. Hodnota rozostrovacieho krúžku (ktorá je počítaná z hĺbkovej mapy) definuje vodivosť prostredia. Čím je rozptylný krúžok pre daný pixel väčší, tým je vodivosť väčšia. Pre pixely nachádzajúce sa v zaostrenej vzdialenosti je vodivosť nulová (nenastáva difúzia). Rozostrenie prebehne ako difúzia, pri ktorej sa na základe vodivosti distribuujú farba pixelov medzi susednými pixelmi.

Z dôvodu časovej zložitosti je vyššie uvedený postup zjednodušený, počíta sa samostatne pre riadky a samostatne pre stĺpce v obrázku. Pre každý riadok(stĺpec) sa zostaví sústava rovníc. Každá rovnica popisuje difúziu konkrétneho pixelu s jeho dvoma bezprostrednými susedmi (počíta sa každý riadok zvlášť). Pre okrajové pixely nastáva difúzia len s jedným susedom. Tvar rovnice je nasledujúci:

$$y_i - x_i = \beta_{i+1}(y_{i+1} - y_i) + \beta_i(y_{i-1} - y_i), \quad (2.30)$$

kde  $x_i$  je  $i$ -ty pixel pre vstupný snímok,  $y_i$  je  $i$ -ty pixel pre výstupný snímok. Hodnota  $\beta_i$  predstavuje vodivosť medzi  $i$  pixelom a  $i-1$  pixelom, ktorá sa získa pomocou nasledujúceho vzťahu:

$$\beta_i = \min(b_{i-1}^2, b_i^2), \quad (2.31)$$

kde  $b_i$  predstavuje priemer rozostrovacieho krúžku  $i$ -teho pixelu.

Úpravou sústavy rovníc 2.30 je možné získať symetrickú tridiagonálnu lineárnu sústavu rovníc v nasledujúcom tvare:

$$\begin{pmatrix} b_1 & c_1 & & & 0 \\ & a_2 & b_2 & c_2 & \\ & & a_3 & b_3 & c_3 \\ & & & \ddots & \ddots & \ddots \\ 0 & & & & a_n & b_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \quad (2.32)$$

kde  $c_i = a_{i+1}$ . Tridiagonálnu sústavu rovníc je možné efektívne riešiť sekvenčným algoritmom v konštantnom čase na vzorku [23].

Informácie, uvedené v tomto odseku, boli čerpané z článku [15].

## 2.8 Štruktúrne rozostrenie

Objektív pri vytváraní obrazu scény dokáže zaostriť na určitú vzdialenosť. V tejto vzdialenosti sa nachádza zaostrená rovina. Ideálny objektív premietne body v zaostrenej rovine na body na senzore. V praxi toto bežne neplatí, niektoré body v závislosti na polohe v zaostrenej rovine sa premietnú mimo rovinu senzoru, dôsledkom čoho sa na sensor premietnú ako krúžky. Toto správanie je dané nedokonalou konštrukciou objektívu. Na obrázku 2.20 je uvedený graf rozostrenia objektívu Olympusu M. Zuiko Digital 14-150mm F4-5.6 pri ohnisku 150mm a clone F5.6. Tento graf znázorňuje rozostrenie v závislosti na pozícii v obraze. V grafe je na x-ovej osi vynesená relatívna vzdialenosť od stredu optickej osi, kde 0% predstavuje stred snímku. Na y-ovej osi je vynesená hodnota rozostrenia. Táto hodnota definuje hraničnú hodnotu počtu riadkov v predmetovom priestore, ktoré by bolo možné v obrazovom priestore rozoznať. Čím je táto hodnota väčšia, tým je výsledný snímok ostrejší (a kvalitnejší). Zmena ohniskovej vzdialenosti alebo zmena clonového čísla vedie aj k



Obrázok 2.20: Rozostrenie Olympusu M. Zuiko Digital 14-150mm F4-5.6: graf znázorňuje ostrosť obrazu v závislosti na pozícii v obraze; prevzaté z [www.dpreview.com](http://www.dpreview.com)

zmene charakteristiky. Hodnoty rozostrenia sú úzko späté s konštrukciou objektívu, preto sa líšia charakteristiky rozostrenia pre jednotlivé objektívy.

## 2.9 Rozostrenie spôsobené pohybom

Fotoaparát zaznamenáva scénu v priebehu určitého časového úseku. V priebehu zaznamenávania sa môže zmeniť poloha objektov v scéne (obrázok 2.21). Táto zmena spôsobí, že pohybujúce sa objekty v scéne budú vo výslednom snímku rozostrené. Toto rozostrenie sa nazýva pohybové rozostrenie. Pohybové rozostrenie môže spôsobiť aj pohyb kamery (obrázok 2.22).



Obrázok 2.21: Rozmazanie spôsobené pohybom objektu v scéne; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)



Obrázok 2.22: Rozmazanie spôsobené pohybom fotoaparátu; fotograf sledoval pohybujúceho sa cyklistu, čím spôsobil, že cyklista ostal na snímku nerozmazaný; prevzaté z [www.flickr.com](http://www.flickr.com)



## Simulovanie

Algoritmy, ktoré simulujú pohybové rozostrenie, je možné rozdeliť do dvoch veľkých skupín. Prvá skupina simuluje rozostrenie pri vykresľovaní scény. V práci [3] je predstavená simulácia pohybového rozostrenia pri použití distribuovaného ray tracingu. Ďalší prístup je popísaný v práci [11], kde je rozostrenie dosiahnuté modifikovaním geometrie objektov v scéne.

Druhú skupinu tvoria algoritmy, ktoré pracujú s 2D snímkom scény. Z tohto dôvodu je možné algoritmy z tejto skupiny zaradiť medzi postprocesové filtre. Jeden z prvých algoritmov, ktoré simuluje tento jav, je popísaný v [19]. V tejto práci je pohyb simulovaný pomocou konvolúcie s rozostrovacou funkciou. Tento prístup umožňuje simulovať rozostrenie, ktoré je identické pre celý priestor obrazu. Ak sa v scéne nachádzajú rôzne sa pohybujúce objekty, je nutné zdefinovať pre každý pohybujúci sa objekt samostatnú rozostrovaciu funkciu, objekty samostatne rozostriť (samostatne skonvoluovať) a výsledky rozostrovania zlúčiť do jediného snímku.

Ďalší algoritmus je uvedený v knihe [18]. Popis tohto algoritmu je uvedený v nasledujúcej sekcii. Zoznam a popis ďalších algoritmov je možné nájsť v práci [6].

### Pohybové rozostrenie popísané v GPU Gems 3 [18]

Uvedený algoritmus bol navrhovaný na GPU, pričom pri jeho tvorbe bolo vyžadované, aby jeho vykonávanie nevyžadovalo veľký výpočetný výkon. Základná verzia tohto algoritmu umožňuje simulovať pohyb kamery v scéne. Algoritmus pracuje ako postprocesový filter, pričom na vstupe vyžaduje farebný snímok a hĺbkovú mapu. Priebeh algoritmu je možné rozdeliť do dvoch fáz:

1. fáza: Získanie pohybového vektora. Pohybový vektor sa získa nasledujúcim postupom:

$$\mathbf{h} = (2x_c - 1, 2y_c - 1, z, 1), \quad (2.33)$$

kde  $(x_c, y_c)$  je relatívna poloha pixelu v obraze ( $x_c, y_c \in \langle 0; 1 \rangle$ ) a  $z$  je hĺbka získaná z hĺbkovej mapy.

$$\mathbf{d} = \mathbf{h} \mathbf{M}^{-1}, \quad (2.34)$$

kde  $\mathbf{M}^{-1}$  predstavuje inverznú maticu projection-view matice, ktorá bola použitá pri generovaní snímku.

$$\mathbf{w} = \frac{\mathbf{d}^T}{d_w}, \quad (2.35)$$

kde  $\mathbf{d}^T$  je transponovaný vektor  $\mathbf{d}$  a  $d_w$  je 4. zložka vektora  $\mathbf{d}$ .

$$\mathbf{d}' = \mathbf{h} \mathbf{M}', \quad (2.36)$$

kde  $\mathbf{M}'$  predstavuje projection-view maticu "pohnutej" kamery.

$$\mathbf{h}' = \frac{\mathbf{d}'^T}{d'_w}, \quad (2.37)$$

kde  $\mathbf{d}'^T$  je transponovaný vektor  $\mathbf{d}'$  a  $d'_w$  je 4. zložka vektora  $\mathbf{d}'$ .

$$(x_r, y_r, z_r, w_r) = \mathbf{h}' - \mathbf{h}, \quad (2.38)$$

čím dostávame pohybový vektor  $(x_r, y_r)$ , ktorý definuje pohyb v obrazovom priestore.

2. fáza: Vytvorenie obrázka s pohybovým rozostrením.

Priebeh rozostrenia je možné popísať nasledujúcou rovnicou:

$$out_{x_c, y_c} = \frac{1}{n} \sum_{i=0}^{n-1} S \left( x_c + x_r \frac{i}{2(n-1)}, y_c + y_r \frac{i}{2(n-1)} \right), \quad (2.39)$$

kde  $out_{x_c, y_c}$  je výstupná farba pixelu,  $n$  je počet pixelov zo vstupného snímku použitých pri pohybovom rozostrení (definuje ju užívateľ),  $(x_c, y_c)$  je poloha bodu v obraze (bola použitá vo vzorci 2.33),  $(x_r, y_r)$  definuje posun bodu v rámci obrazu (získaný zo vzorca 2.38) a  $S$  predstavuje vstupný obrázok.

Na obrázku 2.23 je ukážka vstupného snímku a na obrázku 2.24 je uvedený výstup po aplikácii vyššie uvedeného algoritmu. Popisovaný algoritmus umožňuje simulovať pohybové rozostrenie spôsobené kamerou. Z tohto dôvodu je tento algoritmus vhodný na simulovanie statických scén. Ak by sme požadovali simulovanie pohybového rozostrenia pohybujúcich sa objektov v scéne, je nutné pozmeniť popisovaný algoritmus a je nutné dodať simulátoru na vstup informáciu o pohybe objektov v scéne. Túto informáciu je možné dodať v podobe pohybovej mapy, ktorá pre každý pixel v obraze definuje jeho pohyb v scéne.



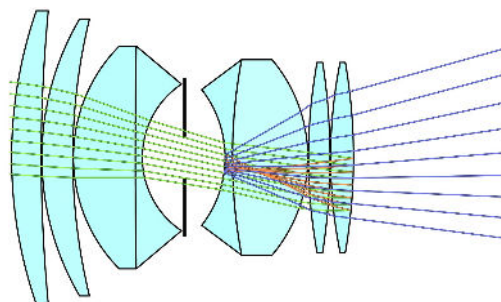
Obrázok 2.23: Obrázok bez pohybového rozostrenia; prevzaté z developer.nvidia.com



Obrázok 2.24: Obrázok so simulovaným pohybovým rozostrením; prevzaté z developer.nvidia.com

## 2.10 Odlesky a parazitné svetlo

Odlesky a parazitné svetlo v obraze je možné definovať ako šírenie svetla v objektíve neželaným spôsobom (šíri sa iným spôsobom ako bolo zamýšľané pri návrhu objektívu). Hlavnou príčinou odleskov sú neželené odrazy medzi optickými prvkami (obrázok 2.25). Odlesky sú dobre viditeľné, ak sa v scéne nachádza malé množstvo veľmi jasných zdrojov svetla. Tieto zdroje sa môžu nachádzať v obraze ale aj mimo obrazu. Odlesky je možné redukovať pridaním optickej clony na objektíve. [9]



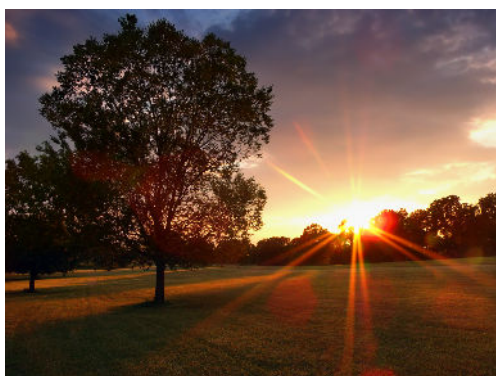
Obrázok 2.25: Odlesky; demonstrácia neželaných odrazov v rámci objektívu; prevzaté z [9]

### Zdroje odleskov

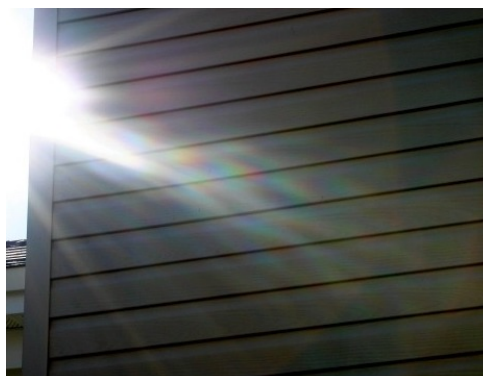
Objektív sa skladá prevažne zo šošoviek, ktorých hlavnou úlohou je usmerňovať šírenie svetla pomocou lomu svetla. Lom svetla nastáva na rozhraní šošovky s okolitým priestorom. Okrem lomu nastáva na šošovke aj odraz svetla. Z tohto dôvodu sa na povrchu šošoviek nachádza anti-reflexná vrstva, ktorá minimalizuje odrazy svetla. Napriek tomuto opatreniu nastávajú odrazy v objektíve a spôsobujú na snímku neželané artefakty (obrázok 2.28).

Druhá príčina vzniku odleskov je difrakcia svetla. Difrakcia môže nastávať na lamelách irisovej clony (obrázok 2.26), alebo môže byť spôsobená nedokonalosťami optických prvkov, znečistením šošoviek, prachom, at. Obrázok 2.27 demonštruje artefakt, ktorý je spôsobený rozptylom svetla v objektíve. Tento artefakt vytvára svetlo vychádzajúce zo slnka, ktoré sa nachádza mimo záberu kamery.

Objektív je konštruovaný tak, aby svetlo vchádzalo do objektívu len cez vstupnú pupilu objektívu. Pri chybe v konštrukcii objektívu môže vniknúť svetlo aj inou cestou, napríklad dierou v tele objektívu. Táto chyba je demonštrovaná na obrázku 2.29. Prejavy odleskov a parazitného svetla sú úzko späté s konštrukciou objektívu. Z tohto dôvodu sa prejavy rôznych objektívov na totožné svetelné podmienky môžu výrazne líšiť.



Obrázok 2.26: Lúče spôsobené difrakciou na lamelách clony, prevzaté z [www.flickr.com](http://www.flickr.com)



Obrázok 2.27: Rozptyl svetla v objektíve, prevzaté z [www.flickr.com](http://www.flickr.com)

### Simulovanie

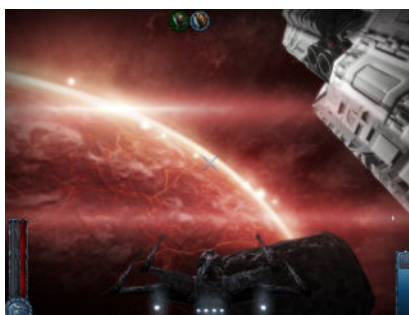
Odlesky vo fotografiách sú prevažne neželaným efektom, ktorému sa fotograf snaží vyhýbať. Napriek tomu sú odlesky neodmysliteľnou súčasťou fotografií a fotorealistického zob-



Obrázok 2.28: Artefakty spôsobené odrazmi v objektíve, prevzaté z [www.flickr.com](http://www.flickr.com)



Obrázok 2.29: Snímok s červeným závojom spôsobeným poškodeným objektívom, prevzaté z [www.wikipedia.org](http://www.wikipedia.org)



Obrázok 2.30: Simulované odlesky v hre; prevzaté z [www.desura.com](http://www.desura.com)

razenia. Z tohto dôvodu sa do počítačom generovaných snímok pridávajú odlesky, aby tieto snímky pôsobili reálnejšie. So simulovanými odleskami je možné sa stretnúť napríklad v počítačových hrách, kde sa využíva výrazne zjednodušený model odleskov. Tento model je navrhovaný tak, aby ho bolo možné generovať v reálnom čase (obrázok 2.30).

Na druhej strane sú sofistikované algoritmy, ktoré umožňujú simulovať odlesky skutočných optických sústav. Tieto algoritmy poskytujú možnosť verne simulovať prejavy odleskov. Aby odrazy boli simulované presne, je nutné vedieť usporiadanie optických prvkov v sústave ako aj ich kvalitu (odrazivosť, pohltivosť...). Výpočet odrazov v sústave s veľkým množstvom prvkov je problematické z dôvodu vysokej výpočtovej náročnosti. Článok [9] popisuje simuláciu refrakcie na lamelách clony a popisuje aj simuláciu odleskov spôsobených neželanými odrazmi v objektíve. Arash Keshmirian vo svojej diplovej práci [13] popisuje simuláciu odleskov na základe fyzikálnych zákonov o šírení svetla.

## 2.11 PID regulátor

PID regulátor je regulačný prvok, ktorý je používaný v regulačných systémoch. Regulačný systém slúži na automatické riadenie regulovaných systémov. Regulovaný systém popisuje funkcia:

$$y(t) = f_t(u(t)), \quad (2.40)$$

kde  $y(t)$  predstavuje výstupný signál regulovaného signálu,  $f_t$  predstavuje regulovaný systém a  $u(t)$  predstavuje riadiaci signál. Funkcia  $f_t$  popisuje systém, ktorého presné správanie nepoznáme. Regulovaný systém môže predstavovať elektromotor. Funkcia  $f_t$  môže popisovať správanie elektromotora,  $u(t)$  upravuje otáčky elektromotora a  $y(t)$  sú otáčky motora. Ak je  $u(t)$  rovné nule, elektromotor nebude upravovať otáčky. Ak  $u(t)$  má kladnú hodnotu, otáčky sa budú zvyšovať a naopak. Motor má určité oneskorenie, preto zmena riadiaceho signálu  $u(t)$  sa prejaví s určitým časovým oneskorením. Regulačný systém bude slúžiť na automatickú úpravu otáčiek za účelom dosiahnutia a udržania otáčiek na hodnote  $y'$ .

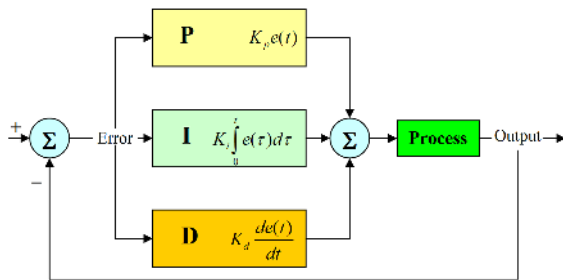
PID regulátor je definovaný následne:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}, \quad (2.41)$$

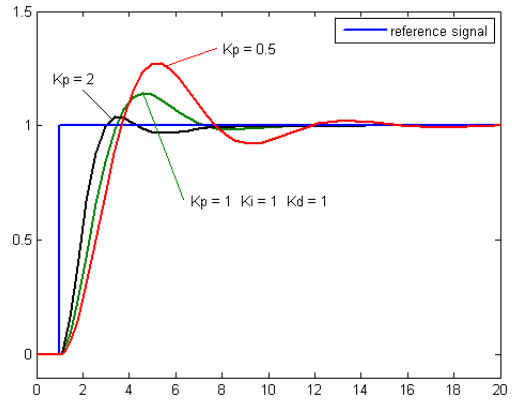
kde  $K_p$  je proporcionálna zložka,  $K_i$  je integračná zložka a  $K_d$  je derivačná zložka. Funkcia  $e(t)$  predstavuje chybu výstupného signálu:

$$e(t) = y' - y(t), \quad (2.42)$$

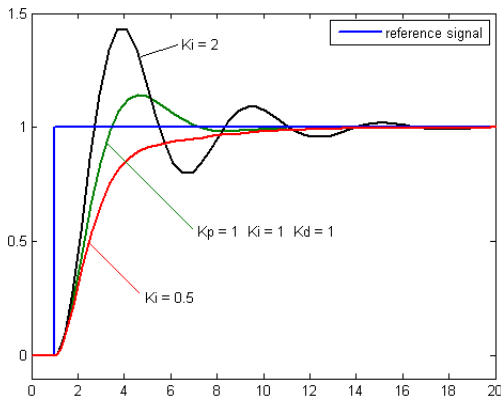
kde  $e$  je chyba,  $y$  požadovaný výstupný signál a  $y(t)$  je výstupný signál regulovaného systému. Schématický popis PID regulátora sa nachádza na obrázku 2.31. Proporcionálna zložka nastavuje kontrolný signál v závislosti na aktuálnom výstupe. Derivačná zložka redukuje záchvevy v kontrolnom signále (obrázok 2.32). Ak by regulovaný systém vykazoval chybu a pri  $y(t) = y'$  a  $u(t) = 0$  by regulovaný systém nebol schopný udržať požadovaný stav, regulovaný systém by nekonvergoval ku želanému stavu. Na konvergenciu systému k požadovanému stavu slúži integrálna zložka PID regulátora. Vplyv zložiek je znázornený na obrázkoch 2.32, 2.33 a 2.34. Z grafov je možné vyčítať rýchlosť konvergence pri rôznych hodnotách jednotlivých zložiek. Ďalšie informácie sú uvedené v [12].



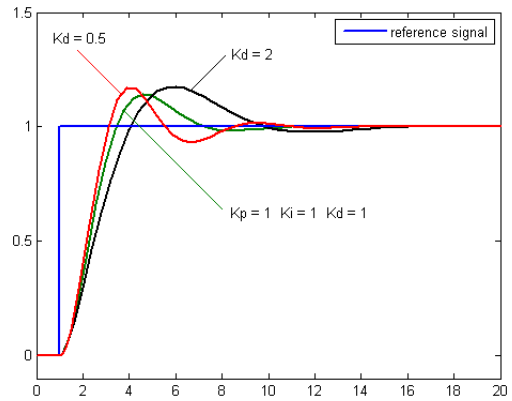
Obrázok 2.31: Schéma PID regulátora; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)



Obrázok 2.32: Graf demonštrujúci konvergenciu výstupného signálu v závislosti na proporcionálnej zložke PID regulátora; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)



Obrázok 2.33: Graf demonštrujúci konvergenciu výstupného signálu v závislosti na integrálnej zložke PID regulátora; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)



Obrázok 2.34: Graf demonštrujúci konvergenciu výstupného signálu v závislosti na derivačnej zložke PID regulátora; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)

## Kapitola 3

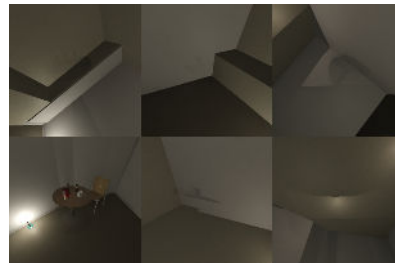
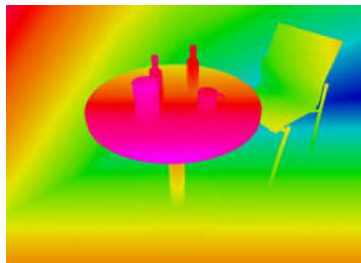
# Zhodnotenie súčasného stavu

V tejto kapitole je uvedené podrobnejšie zadanie projektu a je tu uvedené zhodnotenie existujúcich algoritmov na jeho splnenie.

### 3.1 Zadanie problému

Hlavným cieľom tejto práce bolo navrhnuť a naimplementovať simulátor vlastností kamery, ktorý má umožňovať pridať do počítačom generovaných snímok optické vlastnosti objektívu charakteristické pre konkrétny objektív. Výsledná implementácia má byť použitá v rámci projektu R3-COP a má umožniť generovať vstup pre algoritmy počítačového videnia. Z tohto dôvodu bol projekt konzultovaný s Wolfgangom Herznerom a Oliverom Zendelom z Austrian Institute of Technology. V rámci konzultácii bola upresnená podoba vstupov a výstupov simulátora, ako aj bolo upresnené správanie simulátora.

Vstup pre simuláciu tvorí trojica snímok: farebná mapa (obrázok 3.1), hĺbková mapa (obrázok 3.2) a environment mapa (obrázok 3.3). Farebná mapa predstavuje dokonalú 2d projekciu snímanej scény. Táto mapa je reprezentovaná HDR obrázkom. Druhý typ mapy je hĺbková mapa, ktorá definuje vzdialenosť objektov v scéne od kamery/pozorovateľa scény. Táto mapa sníma ten istý výrez scény, ako výrez ktorý bol použitý vo farebnej mape. Posledný typ mapy je environment mapa, ktorá popisuje okolie kamery. Všetky mapy budú načítavané zo súborov. Súbory budú v štandardnom obrázkovom formáte, pričom testovacie obrázky dodal Oliver Zendel. Niektoré vstupné obrázky dosahovali svojim rozlíšením 2592x1944. Pri implementácii je nutné brať do úvahy, že vstupné obrázky sú vo vysokom rozlíšení a v HDR formáte, čo predstavuje výraznú záťaž pre operačnú pamäť. Demonstračné obrázky 3.1, 3.2 a 3.3 dodal Oliver Zendel.



Obrázok 3.1: farebná mapa    Obrázok 3.2: hĺbková mapa    Obrázok 3.3: environment mapa

Samotný simulátor má za úlohu transformovať vstupnú farebnú mapu s použitím dát

z hĺbkovej mapy a environment mapy na snímok, ktorý bude obsahovať optické vlastnosti objektívu, ktoré sú charakteristické pre konkrétny fotoaparát/kameru. Parametre optických vlastností budú zadané pomocou konfiguračného súboru. Konfiguračný súbor okrem iného modifikuje priebeh simulácie. Simulátora má za úlohu simulovať nasledujúce optické javy:

- efekt hĺbky ostrosti
- pohybové rozostrenie
- odlesky v objektíve, pričom je nutné brať do úvahy aj odlesky spôsobené svetlom mimo uhlu pohľadu
- geometrické skreslenie
- chromatická aberácia
- vinetácia
- zmena jasnosti obrazu na základe zmeny clonového čísla

Výstup zo simulátora má tvoriť obrázok, ktorý bude obsahovať niektoré z vyššie uvedených vlastností. Tento obrázok bude buď zobrazený priamo v okne, alebo bude uložený do súboru. Zmena jasnosti obrazu bude demonštrovaná pomocou sekvencie snímok, ktoré budú uložené v samostatných súboroch.

## 3.2 Zhodnotenie stavu vzhľadom na problém

V predchádzajúcej kapitole je uvedený zoznam algoritmov, ktoré umožňujú simulovať niektoré z požadovaných vlastností objektívu. V tejto časti je uvedená analýza algoritmov pre potreby simulátora.

### Efekt hĺbky ostrosti

Problematika zobrazovania efektu hĺbky ostrosti je dobre preskúmaná. Z tohto dôvodu existuje väčšie množstvo algoritmov, ktoré umožňujú tento jav simulovať. V časti 2.7 sú uvedené referencie na publikácie, ktoré sa zaoberajú simuláciami tohto javu. Algoritmy je možné rozdeliť na algoritmy pracujúce v priestore scény a algoritmy pracujúce v obrazovom priestore. Simulátor pracuje výhradne s obrázkami, informácie o scéne nemá k dispozícii. Z tohto dôvodu pri implementácii simulátora je možné použiť len skupinu algoritmov, ktoré pracujú v obrazovom priestore.

Po preštudovaní dostupných materiálov som sa rozhodol implementovať algoritmus založený na variabilnom FIR filtre [5] a algoritmus založený na difúzii tepla [15]. Tieto algoritmy nie sú časovo náročné a podávajú relatívne dobré výsledky pre scény, v ktorých sa nenachádzajú objekty medzi zaostrenou rovinou a pozorovateľom. Ak sa v tejto oblasti vyskytnú objekty, na výstupných snímkoch sa prejaví nesprávne rozostrenie (obrázok 3.5).

Algoritmus založený na variabilnom FIR filtri podáva relatívne dobré výsledky pre snímky, ktoré neobsahujú veľké zmeny vo veľkosti rozostrovacieho krúžku. Ak snímok obsahuje veľké zmeny rozostrovacích krúžkov, v obraze sa môžu vyskytovať viditeľný artefakt nazvaný "bleeding" (obrázok 3.4). Tento artefakt sa prejaví na hrane medzi zaostrenými a rozostrenými objektami 'roztečením' hrany ostrého objektu.





Obrázok 3.4: Artefakt "bleeding"; obrysy zaostreného tanku "pretekajú" do pozadia; prevzaté z [5]



Obrázok 3.5: Nesprávne rozostrenie objektov v popredí; prevzaté z [5]

Druhý vhodný algoritmus je založený na difúzii tepla [15]. Výhodou tohto algoritmu je, že algoritmus netrpí artefaktom "bleeding". Nevýhodou použitia tohto algoritmu je, že rozostrenie obrazu nie je presné, nakoľko je založené na inom princípe rozostrovania.

### Rozostrenie spôsobené pohybom

V časti 2.9 sú uvedené odkazy na publikácie zaoberajúce sa problematikou zobrazovania rozostrenia spôsobeného pohybom. Simulátor pracuje len so snímkami, preto je možné použiť len algoritmy, ktoré nevyžadujú informácie ohľadom geometrie scény. Algoritmus uvedený v [18] nevyžaduje veľký výpočtový výkon a jeho výstup dosahuje relatívne dobré výsledky.

### Odlesky v objektíve

V časti 2.10 sú uvedené algoritmy, ktoré umožňujú simulovať odlesky v objektíve. Dostupné publikácie uvádzajú algoritmy, ktoré využívajú pri simulácii informácie o optickej sústave. Simulátor nebude mať k dispozícii informáciu o usporiadaní optickej sústavy, preto tieto algoritmy nie je možné použiť. Ďalšie známe algoritmy pracujú len v priestore snímku (neuvažujú priestor mimo zorného poľa kamery). V požiadavkách na simulátor je uvedené, že je nutné brať do úvahy aj zdroje svetla, ktoré sa nachádzajú mimo zorného poľa objektívu. Z tohto dôvodu nie je známy algoritmus, ktorý by vyhovoval požiadavkám na simulátor, a preto je nutné navrhnúť vlastný algoritmus simulácie odleskov.

### Geometrické skreslenie a chromatická aberácia

Časť 2.4 uvádza algoritmus, ktorý umožňuje eliminovať geometrické skreslenie. Výsledné snímky sú pri uvedenom algoritme veľmi kvalitné a nevyžadujú veľký výpočtový výkon. Na pridávanie geometrického skreslenia je možné použiť rovnaký postup.

Na simuláciu chromatickej aberácie je možné použiť rovnaký algoritmus, pričom posun farebných zložiek v obraze bude dosiahnutý rôznymi skresleniami pre jednotlivé farebné zložky.

## Kapitola 4

# Realizácia

Táto kapitola popisuje návrh a implementáciu všetkých hlavných častí simulátora. Táto kapitola popisuje celkový návrh simulátora, uvádza spôsob načítavania dát, priebeh simulácie a na záver uvádza stručný popis výstupu simulátora. Demonštračné snímky, ktoré sú uvedené v tejto kapitole, boli vygenerované navrhovaným a popisovaným simulátorom.

### 4.1 Návrh simulátora

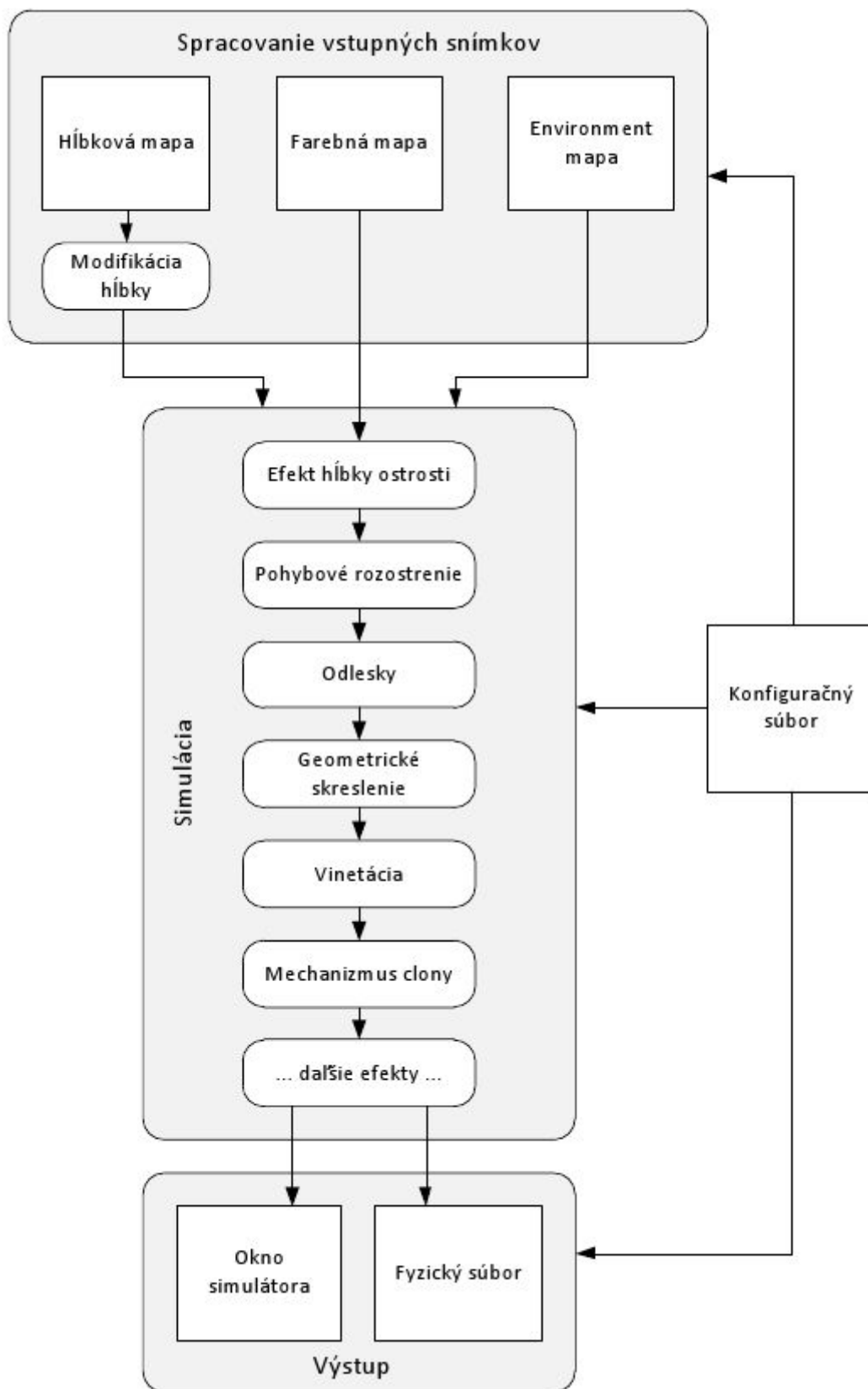
Úlohou navrhovaného simulátora je pridávať do počítačom generovaných snímok optické nedokonalosti objektívu. Na obrázku 4.1 sa nachádza schéma návrhu a rozdelenia simulátora na jednotlivé celky. Simulátor je možné rozdeliť na tri samostatné celky. Tieto celky plnia relatívne samostatné úlohy.

Prvá časť simulátora sa stará o načítavanie a spracovávanie vstupných dát. Simulátor vyžaduje na vstupe farebnú mapu, hĺbkovú mapu a environment mapu. Tieto mapy sú načítavané zo štandardných obrázkových súborov. Druhá časť reprezentuje samotná simulácia objektívu. V tejto časti sa vstupná farebná mapa modifikuje pomocou sady efektov, ktoré simulujú niektoré z vlastností objektívu. Posledná časť obsluhuje spracovanie výsledku simulácie. Výsledok simulácie je v podobe upravenej farebnej mapy. Simulátor umožňuje výslednú farebnú mapu zobrazíť užívateľovi v okne alebo môže výslednú farebnú mapu uložiť do súboru.

Priebeh simulátora je možné ovplyvniť konfiguračným súborom. Konfiguračný súbor ovplyvňuje načítavanie snímok, priebeh simulácie, ako aj upravuje spôsob zobrazenia výslednej farebnej mapy. Pomocou konfiguračného súboru je možné zapínať alebo vypínať simuláciu jednotlivých vlastností objektívu. Taktiež je možné upravovať priebeh simulácie vlastností. Na priloženom DVD sa nachádza konfiguračný súbor, ktorý detailne popisuje jednotlivé parametre.

### 4.2 Spracovanie vstupných dát

Simulácia pre svoj správny priebeh vyžaduje tri rôzne mapy: farebnú mapu, hĺbkovú mapu a environment mapu. Tieto mapy sú uložené v štandardných obrázkových súboroch, ktoré simulátor načítava pomocou externých knižníc. Simulátor používa knižnicu SDL [31] s nadstavbou SDL\_image [27] na načítavanie obrazových formátov JPG, PNG a niektorých ďalších. Kompletný výpis formátov je možné nájsť v dokumentácii ku knižnici [24]. Simulátor



Obrázok 4.1: Schéma rozdelenia simulátora

pracuje s HDR obrazovými dátami (s dátami s vysokým dynamickým rozsahom). Z tohto dôvodu simulátor konvertuje vstupné obrázky do ich HDR reprezentácie.

Simulátor pracuje s HDR obrázkami, preto je vhodné používať obrazové formáty, ktoré umožňujú uchovávať priamo HDR obrázky. Simulátor podporuje formáty OpenEXR a HDR. Formát OpenEXR sa spracováva pomocou knižnice OpenEXR [28] a formát HDR pomocou knižnice HDR Image Reader [30].

Načítané HDR obrázky sa uložia do farebnej, hĺbkovej a environment mapy. Po získaní všetkých potrebných máp simulátor pristúpi k úprave hĺbkovej mapy.

## Úprava hĺbkovej mapy

Simulácia vyžaduje pre svoj správny priebeh hĺbkovú mapu, ktorá definuje vzdialenosť objektov (pixelov) v metroch. Vstupné hĺbkové mapy môžu obsahovať vzdialenosti v rôznych jednotkách. Z tohto dôvodu simulátor implementuje možnosť prevodu vstupnej hĺbkovej mapy na vzdialenosť v metroch. Modifikácia hĺbkovej mapy prebieha v dvoch krokoch. V prvom kroku prebehne linearizácia (ak je nutná), v druhom kroku prebehne prevod na vzdialenosť v metroch.

Knižnice Direct3D a OpenGL umožňujú generovať hĺbkovú mapu. Táto hĺbková mapa je obvykle nelineárna a platí pri nej nasledujúci vzťah:

$$x' = \frac{2x}{f+n-x(f-n)}, \quad (4.1)$$

kde  $x'$  predstavuje lineárnu vzdialenosť,  $x$  predstavuje nelineárnu vzdialenosť,  $n$  vzdialenosť blízkej premietacej roviny,  $f$  vzdialenosť pozorovateľa od vzdialenej premietacej roviny. Linearizácia prebieha pomocou aplikácie vyššie uvedeného vzorca na vzdialenosti uložené vo vstupnej hĺbkovej mape. Ďalšie informácie ohľadom linearizácie je možné vyhľadať v [29]. Ak je vstupná hĺbková mapa lineárna, linearizácia sa nevykonáva. V tomto prípade bude platiť:

$$x' = x. \quad (4.2)$$

V druhom kroku sa prevedie lineárna hĺbka  $x'$  na vzdialenosť v metroch:

$$y = a(b + cx') \frac{1}{\cos(du)}, \quad (4.3)$$

kde  $a$ ,  $b$ ,  $c$  a  $d$  predstavujú konštanty definované užívateľom v konfiguračnom súbore. Premenná  $u$  predstavuje vzdialenosť bodu v obraze od stredu snímky. Jej hodnota sa pohybuje v rozsahu od 0 (pre stred snímky) až po hodnotu  $0.5\pi$  (pre body v nachádzajúce sa v rohoch snímky). Ak nie je nutné lineárnu vzdialenosť modifikovať, je nutné užívateľské konštanty nastaviť na nasledujúce hodnoty:

$$\begin{aligned} a &= 1, \\ b &= 0, \\ c &= 1, \\ d &= 0. \end{aligned}$$

Pri nastavení vyššie uvedených hodnôt bude platiť  $y = x'$ .

### 4.3 Popis simulácie

Navrhovaný algoritmus simulácie pozostáva z niekoľkých efektov, ktoré postupne upravujú vstupnú farebnú mapu. Efekty modifikujú vstupnú farebnú mapu krok za krokom, kde výstup z jedného efektu tvorí vstup pre druhý efekt. Každý efekt simuluje niektorý z javov, ktorý nastáva v objektíve. Poradie efektov je podstatné a výrazne ovplyvňuje výstupný snímok. Pre dosiahnutie želaného výsledku je nutné nastaviť správne poradie efektov. Toto poradie sa nastavuje v konfiguračnom súbore. Pre správne nastavenie poradia je nutné pochopiť vplyv efektov na výsledný snímok. Napríklad, ak bude na vstupnú farebnú mapu aplikovaná vinetácia a následne geometrické skreslenie, simulátor dosiahne iný výstup (obrázok 4.2), ako keď budú efekty aplikované v obrátenom poradí (obrázok 4.3). (Rozdiel v snímkoch je možné pozorovať na svetle nachádzajúcom v ľavom dolnom rohu.)



Obrázok 4.2: Snímok bol najprv ovplyvnený geometrickým skreslením a následne vinetáciou



Obrázok 4.3: Snímok bol najprv ovplyvnený vinetáciou a následne geometrickým skreslením

Simulácia tohto správania je principiálne jednoduchá. Vychádza z návrhového vzoru Stratégia, ktorá oddeľuje rozhranie algoritmu od samotného algoritmu. Návrh sady efektov je možné popísať pomocou nasledujúceho kódu:

```
class IFeature {
    public: virtual void apply (HDRImage* in , HDRImage* &out) = 0;
};

class FeaturetLensFlare: public IFeature;

class Simulation {
    IFeature* m_features;
    int m_countFeatures;
public:
    void simulate (HDRImage* in , HDRImage* &out)
    {
        for (int i = 0; i < m_countFeatures; i++) {
            m_features [ i ]->apply ( in , out );
            in = out;
        }
    }
};
```

Tento prístup umožňuje jednoduchú modifikáciu poradia efektov, ako aj ľubovoľné pridá-

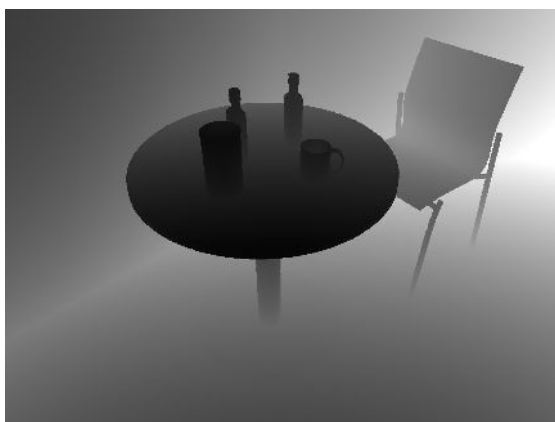
vane a odstraňovanie efektov zo simulácie. Trieda **IFeature** poskytuje rozhranie, ktoré jednotlivé efekty implementujú. Na základe nastavenia v konfiguračnom súbore sa vytvoria inštancie tried efektov, ktoré budú priradené v inštancii triedy **Simulation** do premennej **m\_features**.

V nasledujúcich častiach kapitoly sú uvedené jednotlivé efekty. Mnohé efekty využívajú algoritmy uvedené v kapitole 2. V týchto prípadoch bude uvedený odkaz na príslušnú sekciu.

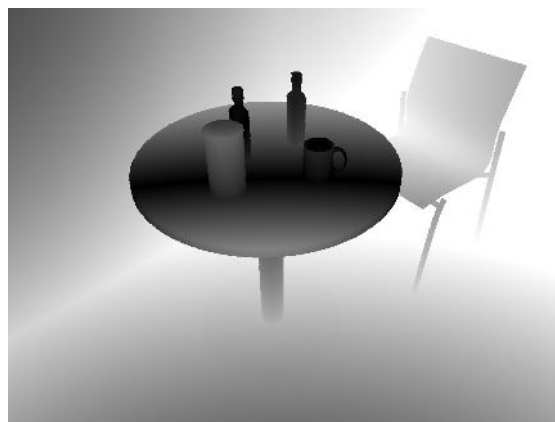
## 4.4 Efekt hĺbky ostrosti a štrukturálne rozostrenie

Efekt hĺbky ostrosti je relatívne dobre preskúmaný jav (viz. sekcia 2.7). Existuje väčšie množstvo algoritmov, ktoré simulujú tento jav. V rámci simulátora sú naimplementované 2 rôzne algoritmy rozostrovania, pričom každý má svoje výhody a nevýhody (viz. sekcia 3.2). Celý priebeh tohto efektu je možné rozdeliť do dvoch fáz:

1. vytvorenie mapy priemerov rozostrovacích krúžkov (obrázok 4.5). Na výpočet rozostrovacích krúžkov sa používa hĺbková mapa. Výpočet rozostrovacích krúžkov prebieha pomocou vzorca 2.24.
2. rozostrenie farebnej mapy na základe priemerov rozostrovacích krúžkov. Rozostrenie v simulátore prebieha pomocou dvoch algoritmov:
  - rozostrenie založené na difúzii (obrázok 4.6)
  - rozostrenie využívajúce variabilný FIR filter (obrázok 4.7)



Obrázok 4.4: Hĺbková mapa; svetlejšie pixely sú bližšie k užívateľovi



Obrázok 4.5: Mapa reprezentujúca priemery rozostrovacích krúžkov; svetlejšie pixely sú definujú väčší rozostrovací krúžok

### Rozostrenie založené na difúzii

Podrobný popis tohto typu rozostrenia je možné vyhľadať v sekcii 2.7 a v článku [15].

### Rozostrenie využívajúce variabilný FIR filter

V algoritme sa používa rozostrovací štvorec. Rozostrovací štvorec predstavuje aproximáciu rozostrovacieho krúžku. Dĺžka strany rozostrovacieho štvorca je rovná priemeru rozostro-



Obrázok 4.6: Efekt hĺbky ostrosti na základe difúzie



Obrázok 4.7: Efekt hĺbky ostrosti založený na lineárnom FIR filtre

vacieho krúžku. Priebeh rozostrovacieho algoritmu je možné popísať vzorcom 2.27, ktorý je v podobe:

$$B(x, y) = \sum_i \sum_j k(x, y, i, j) S(i, j), \quad (4.4)$$

kde  $B$  je výstupná mapa,  $S$  je vstupná mapa a  $k$  predstavuje variabilné konvolučné jadro. Konvolučné jadro pri rozostrovaní má tvar:

$$k(x, y, i, j) = \begin{cases} (2r_{xy} - 1)^{-2} & r_{xy} > (x - i) \ \&\& \ r_{xy} > (y - j) \\ 0 & \text{ostatné prípady} \end{cases} \quad (4.5)$$

kde  $r_{xy}$  vyjadruje polomer rozostrovacieho krúžku pre pixel  $(x, y)$ . Polomer krúžku  $r_{xy}$  musí byť vyjadrený pomocou prirodzeného čísla. Konvolučné jadro spriemeruje pixely nachádzajúce sa v rozptylnom štvorci. Ak by sa algoritmus vykonával podľa vyššie uvedeného postupu, doba jeho vykonávania by bola závislá na veľkosti rozptylného štvorca.

Závislosť časovej zložitosti rozostrenia od veľkosti rozostrovacieho krúžku/štvorca je možné eliminovať pomocou integrálneho obrazu (popis integrálneho obrazu sa nachádza v časti 2.7). Pri použití integrálneho obrazu bude časová zložitosť rozostrenia závislá len na rozmeroch mapy. Priebeh rozostrenia pri použití integrálneho obrazu je nasledovný:

1. získa sa integrálny obraz farebnej mapy
2. pre každý pixel vo farebnej mape sa vypočíta rozptylný štvorec
3. pre každý pixel vo farebnej mape sa pomocou vzorca 2.29 vypočíta suma hodnôt pixelov v rozptylnom štvorci. V integrálnom obraze prebehne výpočet sumy pixelov v konštantnom čase.
4. pre každý pixel sa vypočíta priemerná hodnota pixelov v rozptylnom štvorci. Priemerná hodnota pixelov sa rovná podielu sumy pixelov a počtu pixelov nachádzajúcich sa v rozptylnom štvorci
5. priemerná hodnota predstavuje výstupnú hodnotu pre daný pixel

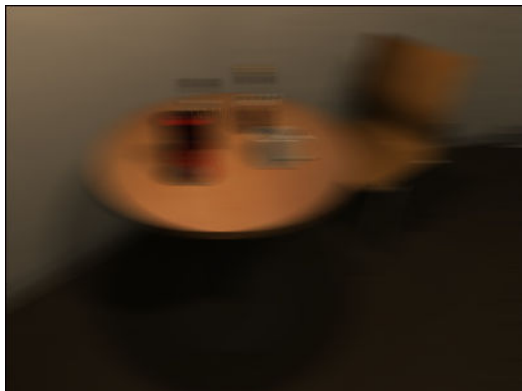
## Štrukturálne rozostrenie

Implementácia štrukturálneho rozostrenia vychádza z implementácie efektu hĺbky ostrosti. Jediný rozdiel je ten, že krúžok rozostrenia definuje užívateľ v konfiguračnom súbore pomocou užívateľskej funkcie (viz. sekcia 4.11).

## 4.5 Rozostrenie vplyvom pohybu

Simulátor implementuje rozostrenie vplyvom kamery (obrázky 4.8 a 4.9). Simulácia prebieha v dvoch fázach:

1. vytvorí sa mapa pohybu, ktorá definuje zmenu polohy pixelov v obraze
2. na základe mapy pohybu prebehne rozostrenie farebnej mapy. Táto fáza využíva rozostrenie popísané v sekcii 2.9.



Obrázok 4.8: Simulované pohybové rozostrenie v horizontálnom smere



Obrázok 4.9: Simulované pohybové rozostrenie vo vertikálnom smere

### Tvorba mapy pohybu

Mapa pohybu definuje zmenu polohy pixelov v rámci obrazu. Simulátor generuje túto mapu na základe informácií o zmene polohy kamery. Zmena polohy kamery sa definuje v konfiguračnom súbore. Z tohto dôvodu simulátor podporuje výhradne simuláciu pohybu kamery, nepodporuje simuláciu pohybu objektov v scéne. Simulovať pohyb objektov v scéne by bolo možné pomocou mapy pohybu, ktorá by bola dodaná ako vstupná mapa simulátora. Vytvorenie pohybovej mapy s pohybom objektov v scéne, je problematické a vyžadovalo by špecializovaný software. Z tohto dôvodu implementovaný simulátor nepodporuje pohybovú mapu na vstupe.

Generovanie mapy pohybu pozostáva z niekoľkých krokov. Pre každý pixel mapy prebieha nasledujúci postup:

1. na základe polohy pixelu v obraze, vzdialenosti v hĺbkovej mape a informácii o zornom poli kamery sa vypočíta poloha bodu v priestore
2. na základe informácie o zmene polohy kamery a zmene optickej osi sa vypočíta nová poloha bodu v obraze



- rozdiel medzi pôvodnou a novou polohou v obraze definuje výstupnú hodnotu pohybu bodu v obraze

## 4.6 Odlesky

Presná simulácia odleskov vyžaduje presné informácie o vnútornom usporiadaní objektívu (viz. sekcia 2.10). Bez tejto znalosti nie je možné dostatočne presne simulovať výstup z objektívu. Z tohto dôvodu simulácia tohto efektu je len aproximáciou odleskov na základe pozorovaného prejavu odleskov na reálnych fotografiách. Simulátor pridáva odlesky prejavujúce sa ako refrakcia na lamelách clony (obrázok 2.26) a centrálny symetrický obraz jasných zdrojov svetla. Obrázok 4.15 demonštruje dosiahnutý výstup.



Obrázok 4.10: Vstupný snímok s jasným zdrojom svetla v obraze



Obrázok 4.11: Do obrázku 4.10 je pridaný efekt odleskov; v červenom krúžku sa nachádza centrálny symetrický odlesk svetla; jas obrázka je výrazne znížený

### Simulácia refrakcia na clone

Simulácia refrakcie vychádza z prejavu odlesku, ktorý sa nachádza na obrázku 2.26. Na obrázku je možné pozorovať odlesk v tvare hviezdy okolo jasného zdroju svetla, slnka. Odlesk môže byť spôsobený aj zdrojom svetla, ktorý sa nachádza mimo záberu kamery (viz. obrázok 2.27). Z tohto dôvodu sa pri simulácii refrakcii vychádza z environment mapy, ktorá obsahuje informácie o priestore mimo záberu farebnej mapy.

Vstupná environment mapa je v podobe kubickej mapy. Táto reprezentácia nie je pre simuláciu odleskov vhodná, preto simulátor rekonfiguruje kubickú mapu do vhodnejšej podoby. Po rekonfigurácii environment mapa reprezentuje súvislý snímok, ktorý zaberá 180 stupňové zorné pole kamery (obrázok 4.14). Následne sa v tejto mape stmavia oblasti nachádzajúce sa na periferiách snímku. Stmavenie periferií sa dosahuje prenasobením mapy vhodnou maskou (príklad masky je na obrázku 4.13).

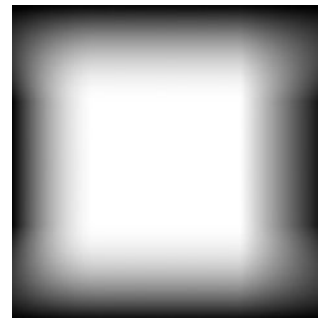
Získaná environment mapa so stmavenými okrajmi je skonvoluovaná s jadrom (obrázok 4.12), ktorý pridáva artefakt refrakciu svetla na clone. Použitá konvolúcia je definovaná:

$$O(x, y) = \sum_{i=0}^{x_{max}} \sum_{j=0}^{j_{max}} I(i, j) \delta(x - i, y - j), \quad (4.6)$$

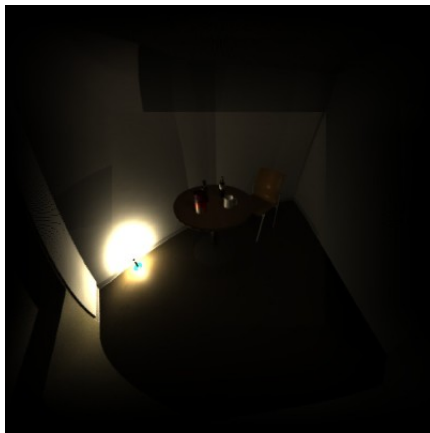
kde  $I$  predstavuje vstupný snímok s rozmermi  $(x_{max}, y_{max})$  a  $\delta$  konvolučné jadro. Pre urýchlenie výpočtu konvolúcie sa prevádza environment mapa a konvolučné jadro do frekvenčnej domény pomocou knižnice fftw [32]. Vo frekvenčnej doméne sa prvky environment mapy a konvolučného jadra sčítajú a výsledok sa prevedie cez inverznú fourierovú transformáciu. Tento postup pri veľkých konvolučných jadrách výrazne urychlí konvolúciu s veľkým konvolučným jadrom. Ďalšie informácie ohľadom konvolúcie sú mimo rozsah tejto práce. Ďalšie informácie je možné vyhľadať v [2, 32]. Výsledok konvolúcie je znázornený na obrázku 4.15. Simulácia efektu pozostáva z vytvorenia výrezu z environment mapy so simulovanými odleskami (obrázok 4.16). Výrez environment mapy zodpovedá zornému poľu farebnej mapy. Výsledný výrez z environment mapy sa pridá do farebného snímku (obrázok 4.17).



Obrázok 4.12: Konvolučné jadro použité na simuláciu refrakcie na clone



Obrázok 4.13: Maska použitá na stmavovanie okrajov



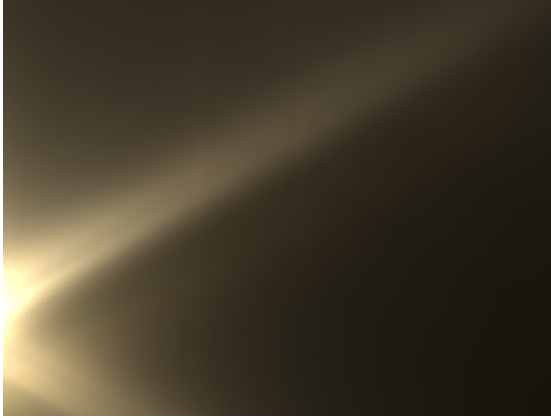
Obrázok 4.14: Environment mapa zaberajúca 180° zorné pole



Obrázok 4.15: Environment mapa, ktorá s pridanými odleskami

### Simulácia centrálne symetrického obrazu

Simulácia tohto javu vychádza z testovacích snímok, ktoré boli vytvorené s fotoaparátom Canonu EOS400D a objektívu EF-S 18-55mm f/3.5-5.6 IS. Snímky boli nasnímané v tmavej miestnosti, v ktorej sa nachádzal len jeden zdroj svetla - LED baterka. Na obrázkoch 4.18 a 4.19 sa nachádza ukážka získaných fotiek. Obe fotky boli získané pri identickom expozičnom čase a pri identickej hodnote clony. Na snímkoch je možné pozorovať, že viditeľnosť centrálne symetrického obrazu je závislá na polohe svetla v obraze.



Obrázok 4.16: Výrez z environment mapy, ktorý obsahuje oblasť zodpovedajúcu výstupnému snímku



Obrázok 4.17: Výsledný snímok s pridaným efektom refrakcie



Obrázok 4.18: V obraze sa nachádza jasný zdroj svetla blízko stredu optickej osi



Obrázok 4.19: V obraze sa nachádza jasný zdroj svetla ďalej od stredu optickej osi

Na základe pozorovaného správania bola naimplementované pridávanie centrálne symetrického obrazu farebnej mapy. Toto sa dosahuje pomocou nasledujúcej funkcie:

$$O(x, y) = I(x, y) + \alpha(x, y) I(x_{max} - x, y_{max} - y), \quad (4.7)$$

kde  $O$  predstavuje výstupnú farebnú mapu,  $I$  vstupnú farebnú mapu,  $(x_{max}, y_{max})$  predstavujú rozmery snímku a  $\alpha$  je funkcia vracajúca viditeľnosť symetrického obrazu. Funkcia  $\alpha$  prevádza súradnice  $(x, y)$  na vzdialenosť od stredu obrázku:

$$\alpha(x, y) = \beta \left( \frac{\sqrt{(x - x_{max}/2)^2 + (y - y_{max}/2)^2}}{0.5\sqrt{x_{max}^2 + y_{max}^2}} \right). \quad (4.8)$$

Získaná vzdialenosť od stredu používa funkcia  $\beta$ , ktorá má nasledujúci tvar:

$$\beta(r) = m_1 e^{r m_2}, \quad (4.9)$$

kde  $m_1$  a  $m_2$  predstavujú koeficienty nastavované v konfiguračnom súbore.

## 4.7 Vinetácia

Simulácia vinetácie pozostáva z prenášobenia vstupnej farebnej mapy (obrázok 4.20) vinetačnou maskou. Vinetačná maska je buď zadaná ako obrázkový súbor, alebo je definovaná

pomocou užívateľskej funkcie. Pri použití užívateľskej funkcie sa vinetačná maska inicializuje pomocou nasledujúceho predpisu:

$$M(x, y) = f \left( \frac{\sqrt{(x - x_{max}/2)^2 + (y - y_{max}/2)^2}}{0.5\sqrt{x_{max}^2 + y_{max}^2}} \right), \quad (4.10)$$

kde  $x_{max}$  a  $y_{max}$  predstavujú rozmery vinetačnej masky a  $f$  je užívateľská funkcia, ktorá je definovaná v konfiguračnom súbore (viz. sekcia 4.11). Ukážka výstupu vinetácie je na obrázku 4.21.



Obrázok 4.20: Vstupná farebná mapa



Obrázok 4.21: Farebná mapa s vinetáciou

## 4.8 Geometrické skreslenie a chromatická aberácia

Simulácia geometrického skreslenia vychádza z algoritmu uvedenom v [33]. Geometrické skreslenie je simulované pomocou sieťového warpingu, pričom na premapovanie polohy pixelov je použitý Lanczosov filter. Priebeh sieťového warpingu ako aj tvar Lanczosovho filtru je uvedený v časti 2.4. Geometrické skreslenie v simulátore je možné definovať buď pomocou mriežky (kde je mriežka definovaná priamo v konfiguračnom súbore) alebo pomocou modelu radiálne symetrického skreslenia. Neexistuje ustálená podoba radiálne symetrického skreslenia (len časť 2.4 uvádza 3 možné podoby). Z tohto dôvodu sa v simulátore nepoužíva konkrétny model skreslenia. Model skreslenia je v konfiguračnom súbore definovaný čiastočne pomocou transformačnej matice (ktorá je v podobe 2.8) a pomocou užívateľskej funkcie (ktorá definuje samotné radiálne symetrické skreslenie a nahrádza funkciu 2.9). Model radiálne symetrického skreslenia definuje skreslenie vstupnej mriežky pri sieťovom warpingu. Ukážka výstupu geometrického skreslenia sa nachádza na obrázku 4.23.

Simulácia chromatickej aberácie vychádza z vyššie uvedeného postupu. Pri chromatickej aberácii prebieha deformácia obrazu samostatne pre červenú a modrú zložku, pričom každá farebná zložka sa upravuje pomocou rozdielneho geometrického skreslenia. Výstup z efektu chromatickej aberácie sa nachádza na obrázku 4.22.

## 4.9 Mechanizmus clony

Clona je zariadenie v objektíve, ktoré ovplyvňuje množstvo svetla, ktoré prejde cez objektiv ak fotocitlivému sensoru. Úlohou clony v simulátore je upravovať jas obrazu tak, aby sa



Obrázok 4.22: Obrázok so simulovaným geometrickým skreslením



Obrázok 4.23: Obrázok so simulovanou chromatickou aberáciou

priemerný jas obrazu priblížil hodnote stredne šedej. Vplyv clony na jas výsledného snímku je možné definovať pomocou vzťahu:

$$I_f = I_{f=1/1} f^2, \quad (4.11)$$

kde  $I_f$  je jas obrazu získaný pri clonovom čísle  $f$  a  $I_{1/1}$  je jas obrazu pri clonovom čísle 1/1. Cieľom adaptácie je dosiahnuť stav:

$$I_f = c, \quad (4.12)$$

kde  $c$  je hodnota strednej šedej a  $I_f$  je priemerný jas obrazu pri clonovom čísle  $f$ . Po dosadení vzťahu 4.11 do vzťahu 4.12 a úprave, dostávame hodnotu optimálneho clonového čísla, pri ktorom výstupný jas obrazu sa bude rovnať stredne šedej:

$$f = \sqrt{\frac{c}{I_{f=1/1}}}. \quad (4.13)$$

Adaptácia clony v simulátore vychádza z konceptu PID regulátora. PID regulátor pracuje so spojitým signálom. Z tohto dôvodu nie je možné použiť PID regulátor v takej podobe, v akej je uvedený vo vzťahu 2.41. Simulátor využíva aproximáciu PID regulátora:

$$u_t = K_p e_t + K_i \sum_{i=0}^t e_i + K_d (e_t - e_{t-1}), \quad (4.14)$$

kde  $u_t$  je zmena kontrolného signálu pre  $t$ -tu iteráciu,  $K_p$  je proporcionálna zložka,  $K_i$  je integračná zložka a  $K_d$  je derivačná zložka a  $e_t$  predstavuje chybu výstupného signálu pre  $t$ -tu iteráciu.

## Simulácia

Simulácia mechanizmu clonového čísla prebieha v rámci niekoľkých simulačných iterácií. V prvom kroku sa vypočíta optimálne clonové číslo:

$$f = \sqrt{\frac{c}{I_{f=1/1}}}, \quad (4.15)$$

kde hodnota stredne šedej  $c$  je zadaná v konfiguračnom súbore a  $I_{f=1/1}$  predstavuje priemerný jas vstupnej farebnej mapy. V nasledujúcom kroku sa vypočíta chyba clonového čísla:

$$e_t = \frac{1}{f} - f_1 t - 1, \quad (4.16)$$

kde  $f_{t-1}$  je clonové číslo získane pri predchádzajúcej iterácii výpočtu. Ak neprebehla ešte žiadna iterácia výpočtu, použije sa ako inicializačná hodnota hodnota clonového čísla, ktorá je nastavená v konfiguračnom súbore. Následne sa vypočíta invertovaný prírastok clonového čísla:

$$u_t = K_p e_t + K_i \sum_{i=0}^t e_i + K_d (e_t - e_{t-1}), \quad (4.17)$$

kde  $K_p$ ,  $K_i$ ,  $K_d$  definuje užívateľ v konfiguračnom súbore. Hodnota  $u_t$  sa upraví tak, aby ležala v rozsahu:

$$u_t \in (-s; s), \quad (4.18)$$

kde  $s$  definuje najväčšiu povolenú zmenu clony v rámci jednej iterácie. Táto hodnota je zadaná v konfiguračnom súbore. V ďalšom kroku sa vypočíta nová hodnota clonového čísla:

$$f_t = f_{t-1} + \frac{1}{u_t}. \quad (4.19)$$

Vypočítané clonové číslo sa upraví tak, aby sa nachádzalo v požadovanom rozsahu:

$$f_t \in (n; x), \quad (4.20)$$

kde  $n$  je najmenšia povolená hodnota clonového čísla,  $x$  je najväčšia povolená hodnota clonového čísla. Hodnoty  $n$  a  $x$  sú definované v konfiguračnom súbore. Vypočítaná hodnota clonového čísla sa použije na zmenu jasu obrazu v nasledujúcej podobe:

$$I_o = I_{f=1/1} f_t^2, \quad (4.21)$$

kde  $I_o$  je výstupný snímok a  $I_{f=1/1}$  je vstupný snímok do efektu.

Tento proces umožňuje simulovať postupnú adaptáciu clony na základe jasu obrazu. Týmto postupom je možné generovať sekvenciu snímok, ktoré budú názorne demonštrovať postupnú zmenu clony pri reálnych fotoaparátach/kamerách. Pri nastavení vhodných hodnôt je možné dosiahnuť skokovú zmenu clony, a tým celý tento proces "zjednodušiť" na vzťah uvedený vo vzorci 4.13.

## 4.10 Výstup

Výstup simulácie tvorí modifikovaná farebná mapa. Simulátor umožňuje zobrazíť farebnú mapu na obrazovku alebo ju uložiť do súboru. Podoba výstupu sa špecifikuje pomocou režimu simulátora v konfiguračnom súbore. Sú podporované dva režimy: režim **demo** a režim **súbor**. Režim **demo** umožňuje špecifikovať efekty, ktoré sa majú aplikovať pri simulácii za behu simulácie. V priebehu simulácie užívateľ môže zapínať/vypínať efekty, ktoré sa budú aplikovať v nasledujúcom cykle simulácie. Výstup zo simulátora je zobrazovaný do okna aplikácie. Po vykonaní jedného cyklu simulácie sa začne vykonávať ďalší cyklus simulácie, pričom nový cyklus simulácie pracuje s rovnakými vstupnými mapami, ako pracoval predchádzajúca simulácia. Ukončenie simulácie nastáva až pri ukončení behu aplikácie.

Druhý podporovaný režim je režim **súbor**. Tento režim je neinteraktívny, neumožňuje modifikovať priebeh simulácie v priebehu spustenia simulácie. Priebeh tohto režimu sa riadi výhradne nastaveniami v konfiguračnom súbore. Konfiguračný súbor špecifikuje ktoré iterácie simulácie sa majú uložiť do obrázkových súborov, a koľko iterácii sa má vykonať. Po vykonaní požadovaného počtu iterácii sa simulátor automaticky ukončí. Tento režim umožňuje generovať samostatné snímky ako aj sekvenciu snímok.

## 4.11 Konfiguračný súbor

Dôležitou súčasťou simulátora je konfiguračný súbor, ktorý ovplyvňuje priebeh simulácie. Prostredníctvom konfiguračného súboru je možné upravovať priebeh jednotlivých efektov, ako aj je možné zapínať/vypínať, ktoré efekty sa majú v priebehu simulácie aplikovať. Význam položiek konfiguračného súboru je uvedený priamo v konfiguračnom súbore. Z tohto dôvodu sa nebudeme venovať v tejto časti jednotlivým položkám konfiguračného súboru. Informácie ohľadom konfiguračného súboru sa nachádzajú priamo v konfiguračnom súbore na priloženom DVD. V tejto časti sa budeme len jedinému prvku - užívateľskej funkcii. Užívateľská funkcia umožňuje adaptovať správanie niektorých efektov.

Užívateľské funkcie sa v konfiguračnom súbore definujú pomocou elementu **func**. Užívateľská funkcia má jeden reálny parameter. Výstup funkcie tvorí jedna reálna hodnota. Ak je užívateľská funkcia definovaná v rámci definície vinetácie, definuje radiálne symetrickú vinetačnú masku, ak je definovaná v rámci geometrického skreslenia, definuje radiálne symetrické skreslenie obrazu atď. V aktuálnej verzii simulátora sú podporované dva typy funkcií - polynomiálne funkcie a funkcie podporujúce lineárnu interpoláciu.

### Polynomiálna funkcia

Polynomiálna funkcia má tvar:

$$f(x) = \sum_{i=0}^m a_i x^i, \quad (4.22)$$

kde  $m$  je stupeň polynomu a  $a_0, a_1, \dots$  sú koeficienty polynomu. Túto funkciu je možné definovať v konfiguračnom súbore pomocou elementu *func* v nasledujúcom tvare:

```
<func type="polynomial" degree= m >
  <value> a_m </value>
  <value> a_{m-1} </value>
  ...
  <value> a_0 </value>
</func>
```

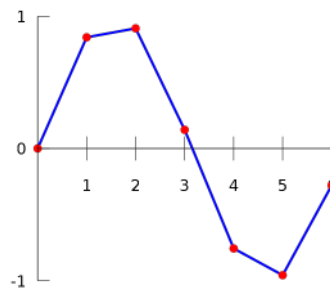
### Funkcia podporujúca lineárnu interpoláciu

Priebeh výpočtu priebehu lineárnej interpolácie je možné zjednodušené popísať, ako preloženie susedných bodov úsečkami (obrázok 4.24). Bližšie informácie ohľadom lineárnej interpolácii je možné vyhľadať v [10].

Priebeh funkcie lineárnej interpolácie je možné jednoznačne definovať pomocou množiny 2d bodov:

$$M = \{(x_0, y_0), (x_1, y_1), \dots, (x_m, y_m)\}. \quad (4.23)$$

V konfiguračnom súbore je možné ekvivalentnú definíciu dosiahnuť pomocou elementu *func* v nasledujúcom tvare:



Obrázok 4.24: Ukážka priebehu lineárnej interpolácie; prevzaté z [www.wikipedia.org](http://www.wikipedia.org)

```
<func type="lininter">  
  <point2 x= $x_0$  y= $y_0$ >  
  <point2 x= $x_1$  y= $y_1$ >  
  ...  
  <point2 x= $x_m$  y= $y_m$ >  
</func>
```



## Kapitola 5

# Záver

V tejto diplomovej práci je uvedený popis simulátora vlastností objektívu, ktorý modifikuje počítačom generované ideálne snímky a upravuje ich do podoby reálnych snímok, ktoré sú ovplyvnené optickými vadami a vlastnosťami objektívu. Plány a požiadavky, ktoré boli kladené na simulátor v rámci diplomovej práce boli splnené. Simulátor umožňuje na základe nastavení v konfiguračnom súbore simulovať vlastnosti konkrétneho fotoaparátu alebo kamery. Podľa požiadaviek užívateľa simulátor generuje jednotlivé snímky alebo sekvencie snímok.

Niektoré javy, ako vinetácia alebo geometrické skreslenie, je možné dostatočne presne simulovať. Na druhú stranu javy ako odlesky a parazitné svetlo nie je možné presne simulovať. Prejav odleskov sa výrazne líši u rôznych objektívov. Z tohto dôvodu simulácia všetkých vlastností objektívu je zložitá a môže byť stále zlepšovaná. Efekty hĺbky ostrosti, pohybového rozostrenia, geometrického skreslenia a vinetácie generujú očakávané výsledky, ktoré sa podobajú prejavom skutočných objektívov. Úspech taktiež predstavuje návrh celého simulátora a parametrizácia simulácie. Návrh simulátora umožňuje relatívne jednoduché pridávanie ďalších efektov alebo modifikáciu stávajúcich efektov. Ďalšie efekty je možné do simulátora pridávať bez výraznejších zásahov do ďalších častí projektu. Na druhú stranu efekt odleskov nie je presný a jeho parametrizácia je v stávajúcej verzii veľmi obmedzená. Simulátor je relatívne rozsiahly, a preto môže obsahovať viaceré chyby.

Snímky, ktoré bolo požadované spracovať, presahovali svojím rozlíšením 1080p (1920x1080 pixelov). Požiadavka na veľkosť vstupného snímku musela byť zohľadnená pri návrhu a implementácii simulátora tak, aby sa zabránilo vyčerpaniu operačnej pamäte, čo malo negatívny dopad na rýchlosti simulácie. Vygenerovanie jedného snímku v rozlíšení 2592x1944 na 2GHz procesore trvalo 88,4 sekundy.

V budúcnosti plánujeme pridať podporu spracovávanía videa na vstupe a výstupe a chceme rozšíriť možnosti parametrizácie efektu odleskov. Javy, ktoré sú simulované, tvoria len časť z javov, ktoré sa prejavujú pri fotoaparátoch. Na výstupný snímok má výrazný vplyv fotocitlivý sensor, ktorý pridáva ďalšie nedokonalosti do snímku, ako je napríklad šum alebo blooming. Z tohto dôvodu by bolo vhodné rozšíriť simulátor aj o simuláciu nedokonalostí sensora. Aktuálne je simulácia pomalá, preto by bolo vhodné zrýchliť priebeh simulácie. Toto bude možné dosiahnuť optimalizáciou simulácie alebo presunom časti časovo náročných výpočtov na GPU.

# Literatura

- [1] Barsky, B. A.; Kosloff, T. J.: Algorithms for rendering depth of field effects in computer graphics. In *Proceedings of the 12th WSEAS international conference on Computers*, Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2008, ISBN 978-960-6766-85-5, s. 999–1010.  
URL <<http://dl.acm.org/citation.cfm?id=1513605.1513775>>
- [2] Bergland, G. D.: A guided tour of the fast Fourier transform. *IEEE Spectrum*, ročník 6, 1969: s. 41–52.
- [3] Cook, R. L.; Porter, T.; Carpenter, L.: Distributed ray tracing. *SIGGRAPH Comput. Graph.*, volume 18, January 1984: s. 137–145, ISSN 0097-8930.  
URL <<http://doi.acm.org/10.1145/964965.808590>>
- [4] Derpanis, K. G.: Integral image-based representations. *Department of Computer Science and Engineering York University Paper*, ročník 1, č. 2, 2007: s. 1–6.  
URL <[http://www.cse.yorku.ca/~kosta/CompVis\\_Notes/integral\\_representations.pdf](http://www.cse.yorku.ca/~kosta/CompVis_Notes/integral_representations.pdf)>
- [5] Fernando, R.: *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education, 2004, ISBN 0321228324.
- [6] Grünscloß, L.: *Motion Blur*. Diplomová práce, Germany, 2008.
- [7] Hruška, E.: *Fotografie na malý formát*. Státní nakladatelství technické literatury, 1959, typové číslo L16-A-4-II/6227.
- [8] Hruška, P.: Geometrická optika. Vysoké Učení Technické v Brně. Studijní materiál k předmětu Fyzikálna optika.
- [9] Hullin, M. B.; Eisemann, E.; Seidel, H.-P.; aj.: Physically-Based Real-Time Lens Flare Rendering. *ACM Trans. Graph. (Proc. SIGGRAPH 2011)*, ročník 30, č. 4, 2011: s. 108:1–108:9.
- [10] Jiří Žára, J. S. P. F., Bedřich Beneš: *Moderní počítačova grafika*. 2004, ISBN 80-251-0454-0.
- [11] Jones, N.; Keyser, J.: Real-time geometric motion blur for a deforming polygonal mesh. In *Proceedings of the Computer Graphics International 2005, CGI '05*, Washington, DC, USA: IEEE Computer Society, 2005, ISBN 0-7803-9330-9, s. 26–31.  
URL <<http://dl.acm.org/citation.cfm?id=1251973.1252477>>
- [12] Katebi, R.: *Robust multivariable tuning methods*. Springer-Verlag, 2012, ISBN 978-1447124245, s. 255–280, book ID: 300199.

- [13] Keshmirian, A.; University of California, S. D. C. S.; Engineering: *A Physically-based Approach for Lens Flare Simulation*. University of California, San Diego, 2008, ISBN 9780549438335.  
URL <<http://books.google.cz/books?id=8aQU0pUNkZ0C>>
- [14] Kosloff, T. J.; Barsky, B. A.: Three Techniques for Rendering Generalized Depth of Field Effects. In *Proceedings of the Fourth SIAM Conference on Mathematics for Industry: Challenges and Frontiers (MI09)*, October 2009, s. 42–48.  
URL <<http://graphics.berkeley.edu/papers/Kosloff-TTR-2009-10/>>
- [15] Lefohn, A.; Owens, J.: Interactive depth of field using simulated diffusion. Technická zpráva, University of California, Davis, 2006.
- [16] Lili Ma, K. L. M., Yangquan Chen: A New Analytical Radial Distortion Model for Camera Calibration. *CoRR*, 2003.
- [17] Nevařil, L.: Pravděpodobnostní převzorkování signálů. Technická zpráva, České vysoké učení technické v Praze, 2007.
- [18] Nguyen, H.: *GPU Gems 3*. Addison-Wesley Professional, 2007, ISBN 9780321515261.
- [19] Potmesil, M.; Chakravarty, I.: Modeling motion blur in computer-generated images. *SIGGRAPH Comput. Graph.*, ročník 17, č. 3, Červenec 1983: s. 389–399, ISSN 0097-8930, doi:10.1145/964967.801169.  
URL <<http://doi.acm.org/10.1145/964967.801169>>
- [20] Resnick, R.; Halliday, D.; Krane, K.: *Physics*. číslo sv. 2 in Physics Series, Wiley, 1992, ISBN 9780471804574.  
URL <<http://books.google.cz/books?id=8MvvAAAAMAAJ>>
- [21] Tsai, R.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, ročník 3, č. 4, 1987: s. 323–344.  
URL  
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1087109>>
- [22] van Walree, P.: Distortion. <http://toothwalker.org/optics/distortion.html>, 2011-08-21 [cit. 2012-04-27].
- [23] WWW stránky: Tridiagonal Matrices.  
<http://faculty.washington.edu/finlayso/ebook/algebraic/advanced/LUtri.htm>, 2006-03-30 [cit. 2012-04-17].
- [24] WWW stránky: SDL\_image 1.2.8.  
[http://www.libsdl.org/projects/SDL\\_image/docs/SDL\\_image.html](http://www.libsdl.org/projects/SDL_image/docs/SDL_image.html), 2009-09-04 [cit. 2012-05-04].
- [25] WWW stránky: Vignetting - Wikipedia.  
<http://en.wikipedia.org/wiki/Vignetting>, 2011-12-24 [cit. 2011-12-30].
- [26] WWW stránky: Ako potlačit chromatickú aberáciu.  
<http://digital-lightroom.net/ako-potlacit-chromaticku-aberaciju>, 2011-12-29 [cit. 2011-12-29].

- [27] WWW stránky: (bez názvu). [http://www.libsdl.org/projects/SDL\\_image/](http://www.libsdl.org/projects/SDL_image/), 2012-01-12 [cit. 2012-05-04].
- [28] WWW stránky: flipcode - HDR Image Reader. [http://www.flipcode.com/archives/HDR\\_Image\\_Reader.shtml](http://www.flipcode.com/archives/HDR_Image_Reader.shtml), 2012-05-04 [cit. 2012-05-04].
- [29] WWW stránky: [GeeXLab] How to Visualize the Depth Buffer in GLSL. <http://www.geeks3d.com/20091216/geexlab-how-to-visualize-the-depth-buffer-in-gls1/>, 2012-05-04 [cit. 2012-05-04].
- [30] WWW stránky: OpenEXR. <http://www.openexr.com>, 2012-05-04 [cit. 2012-05-04].
- [31] WWW stránky: Simple DirectMedia Layer. <http://www.libsdl.org>, 2012-05-04 [cit. 2012-05-04].
- [32] WWW stránky: FFTW Home Page. <http://www.fftw.org/>, 2012-05-06 [cit. 2012-05-06].
- [33] Zemčík, P.; Příbyl, B.; Herout, A.; aj.: Accelerated Image Resampling for Geometry Correction. *Journal of Real-Time Image Processing*, ročník 6, č. 3, 2011: str. 9, ISSN 1861-8200.  
URL <[http://www.fit.vutbr.cz/research/view\\_pub.php?id=9724](http://www.fit.vutbr.cz/research/view_pub.php?id=9724)>

# Příloha A

## Obsah CD

- Elektronická verzia technickej správy
- Spustiteľná verzia implementovaného simulátoru
- Komentovaný konfiguračný súbor
- Zdrojové súbory simulátora
- Zdrojové súbory technickej správy
- Knižnice použité v simulátore
- Vstupné snímky pre simulátor
- Výstupne snímky zo simulátora