

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Zdeněk Mrázek



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**SNÍMÁNÍ A ZPRACOVÁNÍ DIGITÁLNÍCH OBRAZŮ S
VYSOKÝM DYNAMICKÝM ROZSAHEM**

ACQUISITION AND PROCESSING OF DIGITAL IMAGES WITH HIGH DYNAMIC RANGE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Zdeněk Mrázek

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Kamil Říha, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Zdeněk Mrázek

ID: 146910

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Snímání a zpracování digitálních obrazů s vysokým dynamickým rozsahem

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte a popište matematický aparát spojený se skládáním několika snímků s různou expozicí do jednoho snímku s vysokým dynamickým rozsahem. Naprogramujte aplikaci, která bude realizovat funkci vybraných algoritmů na skutečných testovacích snímcích včetně zarovnání obrazového obsahu z vícenásobných expozic. Doporučený nástroj pro implementaci: knihovny OpenCV a MS Visual C++.

DOPORUČENÁ LITERATURA:

[1] GONZALEZ, R. C.; WOODS R. E.: Digital Image Processing, Prentice Hall, New Jersey, 2002.

[2] BRADSKI, G.; KAEHLER A.: Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, Inc. USA 2008, ISBN: 978-0-596-51613-0.

[3] LAGANIÈRE, R.: OpenCV 2 Computer Vision Application Programming Cookbook. Packt Publishing, 2011, ISBN: 1849513244.

[4] PRATA, S.: Mistrovství v C++, Computer Press, Brno 2004, ISBN 80-251-0098-7.

Termín zadání: 1.2.2016

Termín odevzdání: 25.5.2016

Vedoucí práce: doc. Ing. Kamil Říha, Ph.D.

Konzultant diplomové práce:

doc. Ing. Jiří Mišurec, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem této práce je popis matematického aparátu metod skládání několika snímků s různou hodnotou expozice do jediného výsledného obrazu s vysokým dynamickým rozsahem. Vybrané metody a algoritmy jsou detailně popsány v teoretické části a následně implementovány v jazyce C++ s použitím knihovny OpenCV.

KLÍČOVÁ SLOVA

HDR, vysoký dynamický rozsah, LDR, skládání snímků, tónové mapování, expozice, C++, OpenCV, Debevec, Exposure Fusion, Median treshlod bitmap, Drago, Durand, Reinhard, Gredientní doménová komprese.

ABSTRACT

The target of this thesis is summarize the theory of methods used for composition image sequence into single high dynamic range image. Selected methods and algorithms are described in theoretical part and implemented in practical part of this thesis. It was used C++ language with using OpenCV library.

KEYWORDS

HDR, high dynamic range, LDR, image composition, tone mapping, exposure, C++, OpenCV, Debevec, Exposure Fusion, Median treshlod bitmap, Drago, Durand, Reinhard, Gradient domain compression.

MRÁZEK, Zdeněk *Snímání a zpracování digitálních obrazů s vysokým dynamickým rozsahem*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 85 s. Vedoucí práce byl doc. Ing. Kamil Říha, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Snímání a zpracování digitálních obrazů s vysokým dynamickým rozsahem“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Kamilu Říhovi, Ph.D. za jeho vstřícnost, konzultace, trpělivost a cenné rady k práci.

Brno

.....

podpis autora(-ky)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

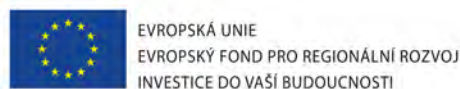
PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

podpis autora(-ky)



OBSAH

Úvod	12
1 Teoretický úvod	13
1.1 Světlo	13
1.2 Kontrast	14
1.3 Fotoaparát	15
1.3.1 Objektiv	15
1.3.2 Clona	15
1.3.3 Závěrka	15
1.3.4 Senzor	16
1.4 Obrazové formáty	17
1.4.1 JPEG	17
1.4.2 RAW	17
1.5 Dynamický rozsah	18
1.5.1 ISO	18
1.5.2 Expozice	18
1.5.3 Dynamický rozsah	19
1.6 Skládání HDR obrazu	21
1.6.1 Vstupní data	21
1.6.2 JPEG snímky	22
1.6.3 Snímky RAW	23
1.7 Tónové mapování	23
1.7.1 Problémy spojené s HDR	24
2 Použité metody a algoritmy	26
2.1 Zarovnání obrazového obsahu	26
2.1.1 Metoda median treshold bitmap	26
2.2 Skládání HDR obrazu	27
2.2.1 Metoda průměru	27
2.2.2 Metoda Debevec, obnova odezvy filmu, response curve	28
2.3 Tónové mapování	32
2.3.1 Lineární tónování	32
2.3.2 Adaptivní logaritmické mapování	32
2.3.3 Reinhardův operátor	34
2.3.4 Metoda bilaterálního filtrování	35
2.3.5 Gradientní doménová komprese	37
2.4 Metoda Exposure Fusion	40

3	Návrh a implementace	42
3.1	Použitý software	42
3.2	Struktura programu	42
3.3	Vstupní sekvence	43
3.4	Zarovnání obrazu	44
3.4.1	Median threshold bitmap	44
3.5	Skládání obrazu	45
3.5.1	Metoda Debevec	45
3.6	Mapování tonality	46
3.6.1	Lineární tónování	46
3.6.2	Metoda Reinhard	47
3.6.3	Metoda Drago	47
3.6.4	Metoda Durand	48
3.7	Metoda Exposure Fusion	49
3.8	Implementace metody gradientní doménové komprese	50
3.9	Postprocessing	59
4	Aplikace a výsledky	62
4.1	Popis aplikace	62
4.2	Výsledky metody gradientní doménové komprese	63
4.3	Srovnání jednotlivých metod	65
4.4	Vstupní sekvence	66
5	Závěr	76
	Literatura	77
	Seznam symbolů, veličin a zkratk	81
	Seznam příloh	82
A	Přílohy	83
A.1	Použité vstupní sekvence	83
B	Obsah přiloženého CD	85

SEZNAM OBRÁZKŮ

1.1	Elektromagnetické spektrum	13
1.2	Ukázka optického klamu	14
1.3	Expozice pro pohled z okna (a) expozice pro pohled do místnosti (b)	20
1.4	Teoretický vs užitný dynamický rozsah. Inspirováno v [10]	21
1.5	Snímky s jinou hodnotou expozice	22
2.1	Ukázka reálné odezvy snímače. Inspirováno v [17].	29
2.2	Bias funkce s hodnotou $b=0,73$	33
3.1	Vývojový diagram procesu vytváření HDR	43
3.2	Vstupní sekvence snímků, s odkrokovanou hodnotu expozice o 1 EV .	43
3.3	Snímky převedené do mediánové bitmapy	44
3.4	Snímek bez filtrace hodnot blízkých mediánu (a), snímek s filtrací (b)	45
3.5	Lineární tónování: sekvence okno	46
3.6	Metoda Reinhard: sekvence okno.	47
3.7	Metoda Drago (adaptivní logaritmické mapování): sekvence okno. . .	48
3.8	Metoda bilaterálního filtrování (Durand): sekvence okno.	49
3.9	Metoda Exposure fusion: sekvence okno.	50
3.10	Vývojový diagram metody gradientní doménové komprese, inspiro- váno v [27].	51
3.11	Vypočítané hodnoty gradientní mapy v ose y (a) a x (b).	52
3.12	Zjednodušený diagram výpočtu tlumící funkce	53
3.13	Gaussova pyramida, kde H_0 je obraz s nejvyšším rozlišením a H_d s nejnižším	54
3.14	Tlumící funkce gradientu $\Phi(x, y)$	55
3.15	Laplaceovo konvoluční jádro.	58
3.16	Výsledná jasová mapa s přesností $\delta = 0.001$	58
3.17	Výsledný zpětně obarvený snímek s přesností $\delta = 0.001$	59
3.18	Histogram snímku 3.16	59
3.19	Ekvalizovaný histogram snímku 3.16	60
3.20	Výsledný jasový snímek s ekvalizací histogramu.	61
3.21	Výsledná jasová mapa s přesností $\delta = 0.001$ a s provedeným zaostře- ním.	61
4.1	Náhled aplikace okna aplikace.	62
4.2	První řádek snímků je s přesností výpočtu iterace $\delta < 0.1$, druhý $\delta < 0.01$, třetí $\delta < 0.001$ a čtvrtý $\delta < 0.0001$. Levé snímky mají parametr $\beta = 0,85$, pravé $\beta = 0,90$	67
4.3	Výsledek s parametrem $\beta = 0.85$ (a), výsledek s parametrem $\beta = 0.90$ (b). Oba snímky jsou s přesností $\delta < 0.001$ a hodnotou $\alpha = 0,1$. . .	68

4.4	Výsledek s parametrem $\alpha = 0,01$ (a) a $\alpha = 1$ (b). Oba snímky jsou s přesností $\delta < 0,001$ a hodnotou $\beta = 0,85$	69
4.5	Srovnání jednotlivých metod s počátečním nastavením vstupních parametrů: Drago (a), Reinhard (b), Durand (c), Mertens (d), Gradientní doménová komprese (e) (s přesností $\delta < 0,001$). (sekvence traktor)	70
4.6	Srovnání jednotlivých metod s modifikovanými vstupními parametry: Drago (a), Reinhard (b), Durand (c), Mertens (d), Gradientní doménová komprese (e). (sekvence traktor)	71
4.7	Srovnání jednotlivých metod s počátečním nastavením vstupních parametrů: Drago (a), Reinhard (b), Durand (c), Mertens (d), Gradientní doménová komprese (e). (sekvence dřevník)	72
4.8	Srovnání jednotlivých metod s modifikovanými vstupními parametry vůči počátečním: Drago - snížená hodnota gamma a bias (a), Reinhard - zvýšená hodnota intenzity a světelné adaptace (b), Durand - zvýšená hodnota kontrastu a snížená hodnota sigma color (c), Mertens - zvýšená hodnota expozice (d), Gradientní doménová komprese - zvýšená hodnota α a β (e) . (sekvence dřevník)	73
4.9	Výsledky gradientní doménové komprese, sekvence okno s počátečními parametry (a), sekvence traktor se změněnou hodnotu $\beta = 0,92$ (b) .	74
4.10	Výsledky gradientní doménové komprese, sekvence dřevník se změněnou hodnotu $\beta = 0,90$ (a), sekvence motor se změněnou hodnotu $\beta = 0,95$ (b)	75
A.1	Vstupní sekvence snímků - okno	83
A.2	Vstupní sekvence snímků - traktor	83
A.3	Vstupní sekvence snímků - dřevník	84
A.4	Vstupní sekvence snímků - motor	84

SEZNAM TABULEK

1.1	Příklady expozičních rozsahů	19
4.1	Výpočetní čas metody GDK pro různé obrazové rozlišení a počty iterací	65
4.2	Výpočetní čas metody GDK pro různé nastavení α a β	65
4.3	Přehled vstupních sekvencí.	66

ÚVOD

V dnešní moderní době má schopnost pořizovat fotografie naprostá většina mobilních zařízení. S nadsázkou tedy můžeme říci, že fotografování už není záležitostí pouze fotografů s drahými přístroji, ale stalo se součástí každodenního života. Snímání a zpracování obrazu je dnes velmi rozšířeným oborem a postupuje stále kupředu, ale i přesto se ani nejsofistikovanější metody a přístroje zdaleka neblíží nejlepšímu snímacímu a zobrazovacímu systému - lidskému vidění.

Při použití jak levnějších, tak i dražších přístrojů pro pořizování fotografií s vysokými světelnými rozdíly narážíme na problém, který je spojen s nedokonalostí technologie, kdy fotoaparát z fyzikální podstaty nemůže zachytit celou scénu se správnou světelností. Pro vyřešení vznikla metoda pro seskládání výsledného obrazu z více vstupních snímků, které jsou nafoceny s různými parametry a každý snímek obsahuje jinou, správně vyfocenou oblast snímané scény. Tato metoda, kterou se v této práci budeme zabývat, je známá pod zkratkou HDR - z anglického High Dynamic Range (vysoký dynamický rozsah) - a skládá se z několika částí, od nasnímání vstupní sekvence, po výsledný obraz s vysokým dynamickým rozsahem.

V první části práce budou popsány základy teorie, které jsou nutné k pochopení algoritmů a metod použitých při procesu HDR. Matematický aparát vybraných metod je vysvětlen a popsán v následující kapitole - od zarovnání obrazového obsahu vstupní sekvence, přes výpočet vysokobitového obrazu HDR, až po jeho zpětné zobrazení pomocí určité metody tónového mapování. Třetí kapitola pojednává o softwarové implementaci metod a matematických algoritmů. Poslední kapitola je věnována výsledné aplikaci a dosaženým výsledkům s různým nastavením vstupních parametrů jednotlivých metod.

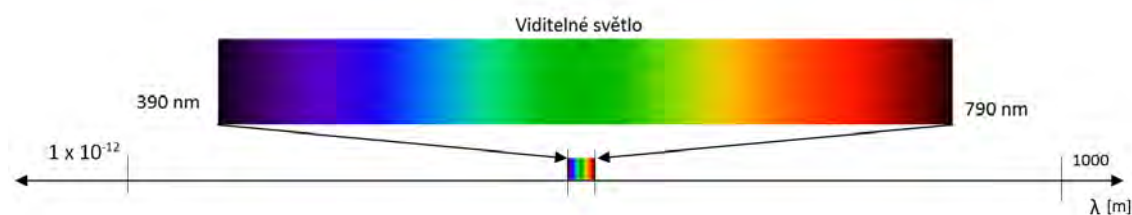
Pro práci bylo zvoleno vývojové prostředí MS Visual studio pomocí jazyku C++ a s použitím knihovny OpenCV a dalších.

1 TEORETICKÝ ÚVOD

Technika snímání a zpracování digitálních obrazů s vysokým dynamickým rozsahem, také známá pod pojmem HDR¹, je velmi zajímavou oblastí v oboru fotografování, respektive v oboru zpracování obrazu. Než však přejdeme k samotnému popisu této techniky, představíme si a popíšeme pojmy a techniky, které je nutné znát k pochopení problematiky tohoto tématu.

1.1 Světlo

V roce 1666 Sir Isaac Newton objevil, že sluneční paprsek po průchodu skleněným hranolem není bílý, ale obsahuje spojitě spektrum barev od fialové po červenou. Tím stanovil teorii, že světlo se skládá z proudu částic. Druhá teorie popisuje světlo jako vlnící se prostředí. V dnešní době užíváme tzv. kvantovou teorii světla, která čerpá z obou teorií. Z fyzikálního hlediska, je elektromagnetické vlnění složeno ze dvou navzájem neoddelitelných složek. Z elektrické a magnetické. Tyto složky jsou na sebe kolmé. Elektromagnetické vlnění o vlnové délce 390 – 730nm se nazývá viditelné světlo, protože tyto vlnové délky dokáže vnímat lidské oko. Na obr. 1.1 můžeme



Obr. 1.1: Elektromagnetické spektrum

vidět, že rozsah světla, které dokážeme vnímat, je pouze velmi malá část elektromagnetického spektra. Na jedné straně spektra jsou rádiové vlny s vlnovou délkou až jeden kilometr, na straně druhé gamma záření s vlnovou délkou v řádech pikometrů. Pro popis elektromagnetického záření jako vlny, používáme veličiny: vlnová délka - λ a frekvence - f , které jsou ve vztahu

$$\lambda = \frac{c}{\nu}, \quad (1.1)$$

kde c je rychlost světla ve vakuu, která činí $2,998 \times 10^8 m/s$. Energie fotonu je dána rovnicí

$$E = h \times \nu, \quad (1.2)$$

¹High dynamic range - vysoký dynamický rozsah

kde h je Planckova konstanta. Jednotka vlnové délky je metr a frekvence Hertz [Hz].

Barvy, které lidské oko vnímá, jsou vlastně světlo odražené od objektů, na které se díváme. Objekt, který odráží světlo všech vlnových délek přibližně ve stejném poměru, se bude jevit lidskému oku jako bílé. Naopak když bude objekt odrážet pouze omezenou část spektra, například v rozmezí $625 - 740nm$, bude se objekt jevit jako červený.

Světlo můžeme rozdělit na monochromatické, achromatické a polychromatické. Monochromatické obsahuje jedinou vlnovou délku (barvu). Typickým příkladem monochromatickosti je laser. Logicky do druhé třídy budeme řadit světlo, obsahující více než jednu vlnovou délku. Příkladem může být denní, nebo žárovkové světlo. Achromatické světlo nemá žádnou barvu, pouze jediný parametr a tím je intenzita světla. Protože intenzita tohoto světla je vnímána od černé, přes šedou, po bílou, užívá se zde pojem stupně šedi.

Analýzou energie, ať světelné (jasu) nebo elektromagnetické, tedy vzniká obraz. Energie potřebná pro vznik obrazu se u lidského oka analyzuje v sítnici, u fotoaparátu na snímači. [1] [2] [3]

1.2 Kontrast

Dalším pojmem, který nepochybně souvisí s tématem, je kontrast. Přesně definovat kontrast je složité. Není důležité, zda hovoříme o jasovém či barevném kontrastu. Jasový kontrast si můžeme matematicky vyjádřit rovnicí:

$$k = \frac{Y_{\max} - Y_{\min}}{Y_{\max} + Y_{\min}}, \quad (1.3)$$

kde Y_{\max} jsou maximální a Y_{\min} minimální hodnoty jasu. Díky kontrastu, je lidské



Obr. 1.2: Ukázka optického klamu

oko schopno rozeznat od sebe objekty a vnímat prostor kolem sebe. Lidský zrakový systém je velmi citlivý na kontrast. Ukázka toho, jak moc je lidské oko na kontrast citlivé, je patrná na obr. 1.2. Na obrázku vidíme přechod z bílé do černé barvy a šest šedých čtverců. Optický klam způsobuje, že se nám čtverce vlevo (na světlém pozadí)

zdají tmavější než vpravo (na tmavém pozadí). Pravdou je, že všechny čtverce mají stejnou barvu. [4][5]

1.3 Fotoaparát

Protože nedílnou součástí této práce je digitální fotoaparát, bude následovat stručný popis nejdůležitějších částí fotoaparátu. Pro práci byl používán fotoaparát Nikon D5100 spolu s objektivem Nikon 18-105mm f/3.5-5.6G ED VR AF-S DX.

1.3.1 Objektiv

Objektiv je pevná, nebo vyměnitelná část fotoaparátu. Zjednodušeně řečeno se jedná o čočku, u lepších objektivů o soustavu čoček. Hlavním úkolem objektivu je soustředění energie - světlo na snímač fotoaparátu. Prvním, ze dvou nejdůležitějších parametrů objektivu je ohnisková vzdálenost, značená f , což je vzdálenost mezi čočkou a ohniskem (obvykle značená v mm). Druhým je clonové číslo. U soustavy čoček se vzájemným posouváním skel může měnit ohnisková vzdálenost a tím se obraz oddaluje nebo přibližuje (optický zoom). [6]

1.3.2 Clona

Clona je zařízení uvnitř objektivu, které reguluje množství propouštěného světla na senzor. Zjednodušeně řečeno, za slunečného dne by měla být clona přivřená, aby propouštěla méně světla, podobně jako v lidském oku, kde funkci regulace množství světla, které dopadá na sítnici, mají pupily. Naopak při focení v místnosti by měla být zcela otevřená. Clona je poskládána z lamel, které vytváří kruh a podle nastavení se stahují, nebo rozevírají a tím mění velikost průchodu pro světlo. Velikost clony značíme F a definujeme následující rovnicí

$$F = f/d, \quad (1.4)$$

kde f je ohnisková vzdálenost a d je průměr otvoru clony. U objektivů se clona udává ve formátu f /clonové číslo např. $f/5.6$, $f/8$ atd. Dalším pojmem úzce související s clonou je závěrka. [7]

1.3.3 Závěrka

Závěrka neboli expoziční čas, je doba, po kterou se propouští světlo na snímač. Jedná se tedy o bránu, která se po uplynutí expozičního času uzavře a tím přestane světlo dopadat na snímací čip. Vyšší expoziční čas tedy propustí více světla a tím

získáme světlejší obraz. U dnešních přístrojů lze tento čas nastavit řádově od sekund po tisíce sekundy - udává se ve zlomkovém tvaru např. 1/1000. [8]

1.3.4 Senzor

Senzor je zařízení, na kterém dochází k převodu obrazové informace na elektrický signál. V dnešních přístrojích se používají dva typy senzorů: CCD a CMOS. Snímače CCD (Charged Coupled Device) jsou starším typem a z důvodu vzniku artefaktů při vysokém jasu scény se nedoporučovaly používat na přeexponované objekty. Lepší architektura tento problém posléze eliminovala. Snímače CMOS (Complementary Metal Oxide Semiconductor) mají menší pořizovací cenu. Jejich citlivost ale nedosahuje takových hodnot jako u CCD a proto jsou vhodné pro scény s dobrými světelnými podmínkami.

CCD

Dopadem fotonů na polovodičovou součástku dochází k uvolnění elektronů z jeho atomů a tím k fotoelektrickému jevu a v jeho důsledku k vytvoření elektrického náboje. Snímač je rozdělen na řádky a sloupce, kde každá pozice řádku a sloupce představuje jeden pixel - výsledný obrazový bod. Po uzavření závěrky, se náboje jednotlivých pixelů posunují přes celý snímač pomocí transportních registrů do pravého horního rohu, kde se tyto hodnoty zaznamenávají. Protože je struktura snímače černobílá a citlivá ve velkém rozsahu vlnových délek, používá se u snímače barevný filtr pro každý pixel. Filtry jsou zelené, modré a červené, stejně jako v lidském oku, které obsahuje čípky citlivé na tyto barvy (barevný model RGB¹). Protože lidské oko je nejvíce citlivé na zelenou barvu (těchto čípků je v lidském oku přibližně 70%), je počet zelených filtrů dvojnásobný. Jednotlivé barevné složky pixelu jsou získány interpolací okolních pixelů. Výsledné obrazové informace jsou kvalitní, obsahují malé množství šumu. Nevýhodou u této technologie je větší spotřeba energie. [13]

CMOS

Snímače CMOS jsou mladší a modernější než CCD, technologie jejich výroby je skoro totožná s výrobní technologií dnešních integrovaných obvodů a proto je i jejich výroba levnější. Oproti CCD má každý pixel snímače 3 tranzistory a fotodiodu, kde se signál zesílí a následně přesune na okraj, kde dojde k zaznamenání jeho hodnoty. Hodnoty jednotlivých pixelů tedy lze zaznamenávat nezávisle na okolních pixelech. Senzory CMOS jsou díky tomuto vyčítání úspornější, protože energie se spotřebovává pouze při těchto stavových změnách. Nevýhodou je výše zmíněná

¹RGB - Red, Green, Blue

menší světelná citlivost, která je přítomna v důsledku špatně dopadajícího světla na tranzistory. [13]

1.4 Obrazové formáty

1.4.1 JPEG

Dnes nepoužívanější obrazový formát JPEG, byl vytvořen skupinou Joint Photographic Experts Group, ze které pochází jeho název. Data tohoto formátu jsou oproti vstupním datům vždy komprimovaná, a proto dochází k částečné ztrátě informací. Lidské oko je více citlivé na změny nízkých frekvencí barev a jasu než na vyšší frekvence. Proto se na začátku procesu formátu JPEG obraz rozloží na bloky jasové a barevné složky a na nich se provede diskretní kosinova transformace. Výsledné bloky se vynásobí kvantizační maticí, která určuje míru komprese. Dalším krokem je kvantifikace výsledných dat, jejich vyčítání a komprimování bezztrátovým kódováním. Největší výhodou formátu JPEG je tato velice efektivní metoda komprese, která dokáže zachovat většinu důležitých obrazových informací jen s malými nepřesnostmi. Jeho použití je především v běžných fotografiích. Pro profesionální focení má svá omezení, právě z důvodu ztrátové komprese. [14] [15]

1.4.2 RAW

Obrazový formát RAW, z anglického překladu „surový“, není klasicky zobrazitelný obsah jaký známe u formátu JPEG a pro jeho zobrazení je nutné použít software, typicky dodávaný výrobcem fotografického zařízení. Jedná se o soubor informací, které byly získány z obrazového snímače daného zařízení a následně nebyly žádným způsobem upravovány. Takto zaznamenaná data jsou uložena ve formátu, který se liší pro různé snímací čipy a jejich výrobce a je jich velké množství. U použitého fotoaparátu Nikon se obrazová informace ukládá do informačního souboru .NEF. Oproti JPEG formátu při zachycení obrazové informace do RAW, nedochází ke ztrátě informace, což je jeho hlavní výhoda. Díky tomu lze vstupní data zachytit ve větší bitové hloubce, dle možností snímače, např. 14 bitů, oproti maximální bitové hloubce 8 bitů u JPEG. Získáme tím větší dynamický rozsah obrazu. Formát RAW je z tohoto důvodu používán v profesionální fotografii, právě díky většímu množství vstupních dat, které lze následně modifikovat a upravovat, a tím dosáhnout kvalitnějších fotografií. [14]

1.5 Dynamický rozsah

Než přejdeme k samotnému vysvětlení dynamického rozsahu, musíme si objasnit pojmy, které jsou s ním přímo spojené.

1.5.1 ISO

Kromě hodnoty clony a expozičního času lze expozici (o které se budeme bavit následně) upravit změnou ISO hodnoty na fotoaparátu. ISO je zkratka pro Mezinárodní organizace pro normalizaci (International Standards Organisation) a zde popisuje ISO jako citlivost filmu (snímacího senzoru), kde ISO hodnota 100 je definována jako nepříliš citlivá, vhodná pro běžné focení s výsledným minimálním šumem a hodnota 1600 jako velmi citlivá. Vyšší hodnota ISO nám umožní získat stejnou expozici z méně světla. Nevýhodou vyšších hodnot ISO je šum. Ve skutečnosti je snímač citlivý pořád stejně, ale vyšší hodnota ISO způsobí, že se dopadající světlo na snímači vynásobí. Logicky tedy vyplývá, že se bude násobit i šum, který se při vysokých hodnotách ISO na fotkách projevuje.

Šum

Šum u digitální fotografie je nežádoucí signál, který vzniká náhodně, nepředvídatelně, a z různých důvodů. V jeho důsledku vznikají obrazové artefakty na výsledných snímcích, připomínající zrnění, viz obr.1.3. Příčinou jsou např. tepelné změny na snímacím senzoru, nebo nízká hodnota osvětlení. Jeho procento výskytu také vzrůstá při vyšších hodnotách ISO, kde se zvyšující se citlivostí násobí signál a tedy i šum.

1.5.2 Expozice

Expoziční hodnota neboli EV (z anglického Exposition value), je množství dopadajícího světla na sensor fotoaparátu. Přesněji jde o kombinaci clonového čísla, expozičního času a ISO citlivosti. Hodnotu EV jednoduše vypočítáme dle vzorce:

$$EV = \log_2 \frac{N^2}{t}, \quad (1.5)$$

kde N představuje clonové číslo a t expoziční čas. Změna hodnoty EV o 1 mění hodnotu dopadajícího světla dvojnásobně, tzn. snížením EV o 1, dostaneme polovinu světla dopadajícího na snímač, zvýšením o 1, dvakrát tolik světla.

Můžeme říci, že ideální expozice nastane, když hodnota EV je nulová při nastavení clony $f/1$, času 1s a ISO 100. Pokud tedy budeme předpokládat, že ideální snímek bude mít hodnotu $EV = 0$, expozici můžeme upravit, abychom dostali tuto

hodnotu. Jedná se o tzv. korekci expozice a té můžeme dosáhnout třemi různými způsoby. Změnou hodnoty ISO, času závěrky nebo hodnotou clony. Přesto pokud budeme fotografovat se stejným přístrojem tutéž scénu s hodnotou $EV = 0$, ale s různým nastavením těchto tří parametrů, výsledky se budou lišit, např. v hloubce ostroty. [10]

1.5.3 Dynamický rozsah

Dynamický rozsah je rozdíl mezi maximální a minimální intenzitou světla v obraze. Jak můžeme vidět v tabulce 1.1, kde je srovnání expozičních rozsahů u různých scén a médií, nejlepším „snímačem“ je lidské oko, s rozsahem až 30EV. Dnešní přístroje se s lidským okem nemohou měřit, jejich rozsah je 6 – 8EV. Tento rozdíl je možné

Tab. 1.1: Příklady expozičních rozsahů

Scéna / médium	Dynamický rozsah [EV]
Lidské oko - jedna scéna	11 – 15
Lidské oko - s dynamickým přizpůsobováním	až 30
Slunečný den	12 – 15
Zatažená obloha	3
Fotografie	6 – 7
Negativ	7 – 8
Standardní digitální fotoaparáty	6 – 7
Profesionální digitální fotoaparáty	7 – 8

demonstrovat na následujícím příkladu. Lidské oko při pohledu z okna budovy ven nemá problém pozorovat zároveň scénu za oknem i uvnitř. Protože snímač na fotoaparátu se dynamickým rozsahem zdaleka neblíží lidskému oku, nastává problém, který můžeme vidět na obr. 1.3. Na obrázku (a) vidíme správně exponovaný obraz pro vnitřní část, část za oknem je bílá, takzvaně „přepálená“, kde hodnoty jasu dosahují maximální hodnoty bílé a splývají. Naopak na obrázku (b) vidíme správně exponovanou venkovní část a vnitřek vidíme tmavý (podexponovaný). Toto je problém, který způsobuje příliš vysoký dynamický rozsah scény. Lze jej vyřešit seskládáním výsledného obrazu z více obrazů, kde z každého vybereme tu část, která je pro danou část správně exponovaná. [9] [10]



a)



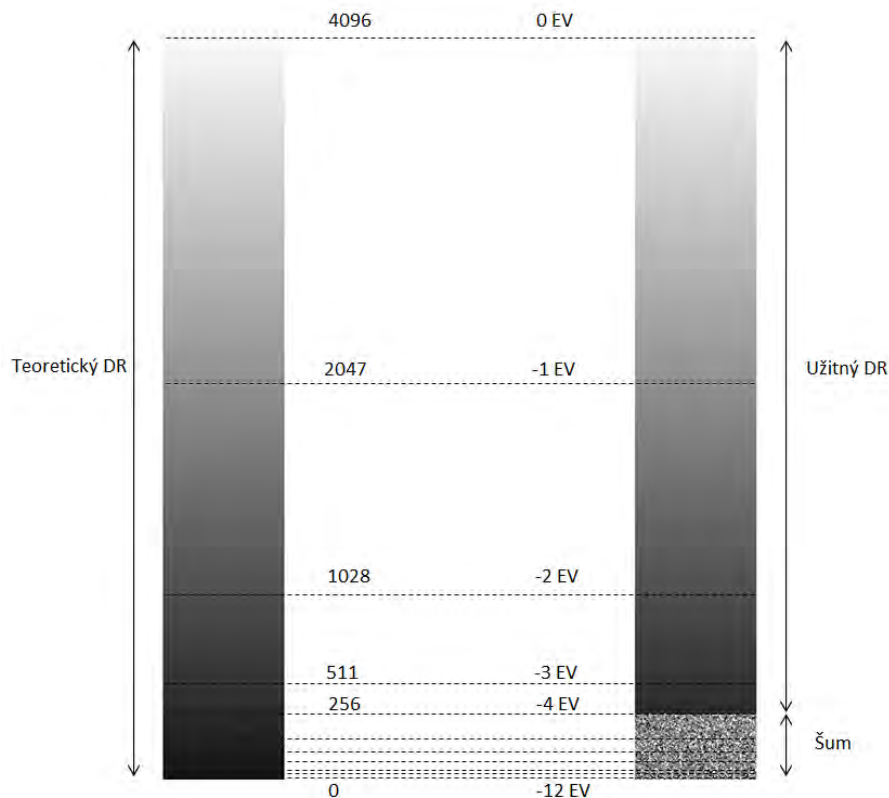
b)

Obr. 1.3: Expozice pro pohled z okna (a) expoziční hodnota pro pohled do místnosti (b)

Teoretický vs užitný dynamický rozsah

Jak bylo řečeno, dynamický rozsah je rozdíl mezi maximální a minimální intenzitou světla v obraze. Maximální hodnota je nejvyšší počet elektronů, které lze zpracovat na jednom pixelu snímače, kde bude nejvyšší hodnota signálu. U minimální hodnoty narážíme na problém spojený se šumem. Dvojnásobek dopadajícího světla (+1 EV) na snímač znamená dvojnásobně větší hodnotu informace. Stejně tak polovina znamená poloviční hodnotu (-1 EV). Při vhodných podmínkách, kdy nastavuje expoziční hodnotu 0 EV, dostaneme na výstupu snímače maximální hodnotu v závislosti na typu snímače. Pro snímač s hloubkou 12 bitů dostaneme hodnotu $2^{12} = 4096$. Pro polovina světla, tj. EV -1 bude hodnota poloviční 2048, atd. viz obr.1.4, až hodnota EV -12 odpovídá informaci 0.

Teoreticky by tedy 12 bitový snímač měl zaznamenat dynamický rozsah 12 EV, což ale nekoresponduje s informacemi uvedenými v tabulce 1.1. V důsledku vzniku šumu na snímači (viz kapitola 1.5.1), např. při málo osvětlené scéně, je část obrazové informace zkrácena šumem. Odstup signál/šum tedy zmenšuje teoretický dynamický rozsah o nízké expoziční hodnoty viz obr.1.4. Užitný dynamický rozsah tedy závisí na odstupu signál/šum u daného obrazového snímače. Nejvhodnější nastavení expoziční



Obr. 1.4: Teoretický vs užitný dynamický rozsah. Inspirováno v [10]

tak, abychom využili co největší dynamický rozsah, je tedy na hodnotě 0 EV. [10]

1.6 Skládání HDR obrazu

Nyní se dostáváme k jádru práce, tj. zvyšování dynamického rozsahu. Protože dnešní přístroje nejsou schopny zachytit scény s vysokým dynamickým rozsahem přímo, pro výsledný obraz HDR potřebujeme více vstupních dat, které musíme nejprve získat.

Standardní výstupní formát dnešních fotoaparátů je JPEG s 8 bity pro každý barevný kanál RGB, který ovšem pro foto typu HDR nestačí, tzn. obsahuje příliš málo informací. Vhodnější volbou jsou RAW snímky, které mají samotné větší bitovou hloubku a proto s nimi lze dosáhnout lepších výsledků.

1.6.1 Vstupní data

Jak již bylo řečeno, abychom dostali HDR obraz s hloubkou informace více než 8 bitů, musíme obraz seskládat z několika vstupních obrazů s různou hodnotou expozice, abychom pokryli celkový dynamický rozsah fotografované scény. Tyto snímky jsou označovány jako LDR (z anglického Low Dynamic Range, přeloženo jako nízký

dynamický rozsah). Seskládáním LDR snímků, které mohou být dva, ale i více (vždy záleží na dané scéně), následně získáme HDR obraz. Problémem ovšem je, že tyto snímky musí být stejné a lišit se pouze v hodnotě EV, tj. scéna musí být neměnná. Nyní si popíšeme způsoby, jak tyto snímky získat.

1.6.2 JPEG snímky

U této metody narážíme na problém spojený s rozdílem scény u snímků. Focení více snímků JPEG znamená, že po vyfocení jednoho musíme upravit expozici. Než dojde k této změně, uběhne jistý časový interval a scéna již nemusí být totožná, jako kdybychom vyfotili všechny obrazy najednou. Problémem není pouze stabilita fotoaparátu, kterou řeší stativ, ale primárně změna samotné scény, která se mění od nepatrných změn, např. pohyb mraků, listů ve větru, až po velké, pohyb lidí, aut, atd. Proto se HDR používá hlavně pro statické scény.

Přesto i při použití stativu může dojít k posunu scény, protože zmáčknutím spouště dojde k otřesu fotoaparátu, způsobeném pohybem mechanických částí uvnitř fotoaparátu (závěrky). Vhodné je používat funkci *bracketing* (pokud fotoaparát tuto funkci má). Jedná se o funkci, která umožňuje vyfocení několika snímků v rychlém sledu po sobě (typicky 3), se změněnou hodnotou EV. Rozdíl kroku EV jednotlivých fotek lze nastavit v závislosti na používaném zařízení. U daného typu fotoaparátu, který byl použit, lze nastavit rozdíl expozice ± 2 EV.



Obr. 1.5: Snímky s jinou hodnotou expozice

Jak jsme si uvedli v kapitole 1.5.2, hodnota EV závisí na třech parametrech. ISO hodnota není vhodná pro změnu expozice z důvodu zvýšení šumu, viz kapitola 1.5.3 Měnit hodnotu clony také není vhodné, protože změna jejích parametrů ovlivňuje více parametrů obrazu než jen jas, především hloubku ostrosti. Nejlepším parametrem pro změnu hodnoty EV je expoziční čas, který nemá vliv na žádné další parametry obrazu. Na obrázku 1.5 jsou tři snímky pořízené funkcí bracketing, s rozdílnou hodnotou expozice 2 EV. Protože se jedná o statickou scénu, pohyb fotoaparátu ani změna scény není patrná. Přesto je použití stativu velmi důležité,

protože i jemně nezarovnané snímky mohou způsobit ve výsledném obrazu viditelné artefakty. [10]

1.6.3 Snímky RAW

Fotit do formátu RAW dnes podporují digitální zrcadlovky a některé kompaktní fotoaparáty. Tento soubor jsou surová data (z anglického „raw“), která byla zachycena snímačem a následně již nijak neupravována, na rozdíl od JPEG, u kterého dochází např. k převodu do 8 bitového formátu pro každý barevný kanál. Zrcadlovka, kterou budu používat pro tuto práci, fotí do formátu RAW s bitovou hloubkou 14 bitů. Z tohoto důvodu je možné z jedné fotky ve formátu RAW získat mnohem více obrazových informací než u JPEG, tudíž zobrazit větší dynamický rozsah. Pomocí softwaru si u souboru RAW můžeme navolit, jaké obrazové parametry má výsledná, již zobrazitelná fotografie, mít. Pro metodu HDR je nejdůležitější a žádaná volba parametru rozdílné expozice. Jako základní fotku budeme považovat tu, s hodnotou expozice 0 EV a další fotky navolíme s rozdílem např. $\pm 3\text{EV}$. Takto dostaneme tři stejné snímky s odlišnou hodnotou expozice. Protože jsou snímky vyfoceny „všechny najednou“, tento způsob eliminuje problém se změnou (pohybem) scény u postupného focení více fotek s různou hodnotou expozice. Má však i svoji nevýhodu, jsme omezeni vlastností snímače, a i přesto, že 14 bitový RAW obsahuje více informací než 8 bitový JPEG, nezískáme stejně velký dynamický rozsah jako u postupného focení a korekce expozice ručně, protože hodnoty rozdílu EV jsou omezené.

Nejlepším řešením je kombinace focení více snímků s jinou expoziční hodnotou do formátu RAW. Takto seskládaný výsledný HDR obraz bude mít největší dynamický rozsah. V potaz musíme samozřejmě brát, že nastanou stejné problémy, které byly uvedeny u skládání více snímků JPEG. Proto se nějakým způsobem musí provést zarovnání obrazové scény, aby se tyto problémy eliminovaly.

U HDR obrazu z jednoho RAW snímku nedosáhneme tak dramatického efektu, ale zcela vymizí problém se změnou scény. Proto je vhodné tento postup aplikovat např. na pohyblivou scénu, nebo videosekvenci. [10]

1.7 Tónové mapování

Protože dnešní zobrazovací zařízení používají 8 bitový barevný model RGB nebo CMYK¹ (resp. 8 bitů na jeden barevný kanál) a nedokáží obraz HDR s vysokým bitovým rozlišením zobrazit přímo, je nutné HDR obraz převést z vysokého bitového rozsahu zpět na 8 bitový LDR obraz, který již zobrazit lze. Umožňuje to takzvaný

¹CMYK - subtraktivní barevný model obsahující barvy Cyan-azurová, Magneta-purpurová, Yellow-žlutá a Key-černá

tone-mapping, přeloženo jako mapování tonality, který je důležitou součástí práce se snímky HDR. Vysoký dynamický rozsah se snižuje s cílem zobrazit obraz co nejlépe v jediném obrazu LDR. Jedná se vlastně o konverzi, kde dochází ke komprimaci a tedy i ztrátě informace, s cílem zachovat co největší přirozenost, barevnost a hodnoty kontrastu a jasu. Protože výsledný obraz LDR se nemůže rovnat dynamickým rozsahem scény realitě vnímanou člověkem, je nutné při tomto procesu použít metody založené na poznatcích o lidském zrakovém systému. Používají se optické klamy a triky, aby výsledný obraz působil co nejrealističtěji.

Tónové mapování lze považovat za systém, kdy vstupem je jeden HDR obraz a výstupem jeden LDR obraz. Jak úspěšný výsledek dostaneme, záleží na algoritmu, který pro tento systém použijeme. Pokud budeme upravovat každý pixel obrazu zvlášť a nebudeme brát v potaz informace, které obsahují okolní pixely, mluvíme o metodě globální. Naopak při analýze okolních pixelů a jejich následném zahrnutí do výpočtů se jedná o metody lokální. Existují i metody hybridní.

Hlavní cíl tohoto procesu je především v myšlence, že globální kontrast HDR obrazu se komprimuje více, a lokální kontrast, který obsahuje důležitější informace, jako jsou detaily, se komprimuje co možná nejméně. [12] [16]

1.7.1 Problémy spojené s HDR

Při procesu skládání snímků a mapování tonality se můžou v důsledku vlastností metod, nebo např. chybně zvolených parametrů se mohou ve výsledném snímku vyskytnout jevy, které působí nepřírozně a celkově snižují jeho kvalitu. Některé nejčastější artefakty jsou krátce popsány dále. [23]

Halo efekt

Kontrast je u lokálních operátorů nastavován v závislosti na okolních pixelech a právě proto zde může nastat problém při ostrých přechodech oblastí s velkým jasnem a oblastí s malým jasnem - hran. Při procesu mapování jsou v blízkosti těchto hran do výpočtu zahrnuty pixely z opačné strany hrany. Efektem je, že u oblasti s velkým jasnem dojde k dalšímu zvýšení jasu a u oblasti s nízkým ke snížení.

Inverze tónů

Lokální metody neupravují jas pouze pomocí tonální křivky, jak je tomu u metod globálních. Hodnoty jasu pixelů na histogramu snímku nejsou pouze upraveny, ale může se změnit jejich pozice. Tímto způsobem se může ve výsledném snímku vyskytnout oblast s menším jasnem než jiná oblast, která může mít po tónovém mapování stejný jas, nebo dokonce vyšší.

Duchové (Ghosts)

Pokud při snímání vstupní sekvence dojde mezi jednotlivými snímky k posunu scény, pohybu ve scéně, nebo pohybu samotného fotoaparátu, ve výsledném obraze vznikne několik siluet, takzvaných duchů. Tento problém lze v některých případech řešit zarovnávacím algoritmem při skládání vstupní sekvence.

Šum

Při tónovém mapování je žádoucí zachovat a zvýraznit co nejvíce detailů. Spolu s detaily se ovšem zvýrazní i šum. Nejvíce náchylné k tomuto jevu jsou velké oblasti bez přechodů, typicky obloha. Řešením je zredukovat parametr, který má zvýrazňovat menší detaily (pokud je dostupný), nebo použití vyhlazovací funkce.

2 POUŽITÉ METODY A ALGORITMY

V následující části si popíšeme vybrané algoritmy a metody, se kterými budeme dále pracovat v praktické části této práce.

2.1 Zarovnání obrazového obsahu

Dříve než přejdeme k metodám skládání vstupní sekvence, je nutné se vrátit k problému, který byl zmíněn dříve, a to problém spojený se snímáním této sekvence. Obrazy s různou hodnotou expozice získáme postupným focením s upravováním času závěrky a mezitím může dojít ke změně obrazové scény, typicky pohyb malých předmětů ve větru (listy), nebo větší oblasti (pohyb mraků po obloze). Protože dnešní fotoaparáty fotí s vysokým rozlišením (použitý fotoaparát Nikon fotí s maximálním rozlišením 4928 na 3264 pixelů), může také dojít k posunu celé scény z důvodu nepatrného pohybu fotoaparátu. Posun celé scény je problematický především pro gradientní oblasti s vysokým rozdílem jasu sousedních pixelů - hrany. Z těchto důvodů je vhodné ještě před procesem skládání fotografií použít metodu zarovnání obrazového obsahu.

2.1.1 Metoda median treshold bitmap

Na začátku metody představené v článku [18] pro zarovnávání snímků je potřeba si převést N obrazů vstupní sekvence do 8 bitových stupňů šedi. Předpokládáme, že na vstupu máme 3 kanálový 24 bitový obraz s 8 bity na jeden barevný kanál. Převod můžeme provést za použití pouze zeleného kanálu, nebo lépe za pomoci všech tří RGB kanálů dle následující rovnice

$$z_i = \frac{54 \times R_i + 183 \times G_i + 19 \times B_i}{256}, \quad (2.1)$$

kde se bere v potaz skutečnost vnímání barev lidského oka a koeficienty se váhují dle citlivosti na jednotlivý barevný kanál.

Ze série vstupní sekvence je jeden obraz zvolen jako referenční. Přestože jedna z výhod tohoto algoritmu je, že nezáleží na hodnotě expozice u snímku, je vhodné zvolit snímek ze středu série, kde nejsou příliš pře/podexponované oblasti. Algoritmus se aplikuje vždy na dva snímky, z nichž jeden je právě onen referenční. Výstupem je série $N-1$ hodnot posunů od referenčního snímku, kde N je počet snímků vstupní sekvence.

První kroky algoritmu median treshold bitmap jsou:

1. Z hodnot převedeného 8 bitového obrazu stupňů šedi zjistíme jeho medián M

2. Vytvoří se bitmapový obraz, kde hodnoty větší než medián budou mít hodnotu 1 (255 pro 8 bitový obraz) a hodnotu menší než medián 0, viz následující rovnice

$$B_i = 1 \text{ pro } z_i > M \quad (2.2)$$

a

$$B_i = 0 \text{ pro } z_i < M, \quad (2.3)$$

kde B_i je hodnota pixelu ve výsledné bitmapě.

Dalším krokem je vytvoření obrazové pyramidy, tedy sekvence dvojic obrazů, kdy první dva jsou aktuální bitmapové obrazy a další jejich vždy o polovinu menší kopie. Počet zmenšení je roven

$$A = \log_2(\max_offset), \quad (2.4)$$

kde *max_offset* je hodnota maximálního posunutí, která se zjistí rekurzivně. Nyní máme vždy dva threshold bitmapové obrazy, které je třeba zarovnat. Snímky se vůči sobě posouvají o ± 1 pixel v osách x a y a přitom se počítá jejich rozdíl - offset. Takto se určí, při jakém posunutí jsou snímky nejvíce korespondující, a tato hodnota se uloží. Protože se postupuje od nejmenší dvojice snímků, hodnota offsetu se při přechodu do vyšší úrovně pyramidy dvojnásobně zvětší, a výpočet se provede znova. Tímto postupem dochází vždy ke zpřesnění hodnoty offsetu. Poslední výpočet se provede nad původní velikostí snímků a tím získáme výslednou hodnotu offsetu.

Při výpočtu může dojít k problému u obrazů, které mají větší počet pixelů blízko hodnotě mediánu. Tyto pixely se projeví v bitmapovém obrazu jako šum a můžou mít negativní vliv na výsledný výpočet. Řešením tohoto problému je vyloučení těchto pixelů z výpočtu pomocí exkluzivní bitmapy. Tato exkluzivní bitmapa vynuluje pixely, které jsou svými hodnotami ± 4 od mediánu. [18]

2.2 Skládání HDR obrazu

2.2.1 Metoda průměru

Nejjednodušší algoritmus pro skládání HDR obrazu je zprůměrování hodnot ze zdrojových snímků. Snímky mají různý expoziční čas, a proto v totožných bodech jednotlivých obrazů mají pixely jiné jasové intenzity. Jasové hodnoty pro každý pixel vypočítáme dle vzorce

$$Z_i = \frac{1}{N} \sum_{i=0}^{N-1} A_i \cdot 2^{H-L}, \quad (2.5)$$

kde H je bitová hloubka výsledného HDR obrazu (např. 32 bitů), L je bitová hloubka vstupních obrazů (typicky 8 u JPEG), N je počet vstupních obrazů a A je vstupní

matice hodnot, ze které se počítá průměr, tzn. hodnoty jasu pro jeden barevný kanál ze všech vstupních obrazů.

Tímto způsobem získáme výsledný HDR obraz. Z vzorce 1.6 vyplývá, že jeho bitová hloubka bude větší než 8 bitů (např. z 5 vstupních obrazů bude bitová hloubka HDR obrazu 32 bitů). Pro zobrazení si ho musíme převést na 8 bitový. Metoda, která nám toto umožní, se nazývá tónové mapování (z anglického tonemapping).

Tyto zprůměrované hodnoty nebudou plně pokrývat celý dynamický rozsah scény, ale pouze určité body, protože hodnoty budou mít mezi sebou rozestupy ne 1, ale 2^{H-L} . Zvětšení tohoto rozsahu a efektivnější mapování tonality dosáhneme přidáním náhodnosti k výsledné hodnotě jasu pro každý pixel. Výsledek bude obsahovat hodnoty celého dynamického rozsahu. Původní vzorec rozšíříme o přičtení náhodného čísla:

$$Z_i = \frac{1}{N} \sum_{i=0}^{N-1} A_i \cdot 2^{H-L} \pm rand(x2^{H-L}, y2^{H-L}), \quad (2.6)$$

kde x je nižší koeficient náhodnosti $x = 0,75$, y vyšší s hodnotou $y = 1,25$ a funkce *rand* generuje náhodné číslo mezi těmito koeficienty.

Tento algoritmus se dá vylepšit přidáním váhy některým obrazům. Je logické, že střední obraz bude obsahovat více relevantních informací než obrázky krajní, jejichž větší plochy mohou být přexponované nebo naopak podexponované. [11] [12]

2.2.2 Metoda Debevec, obnova odezvy filmu, response curve

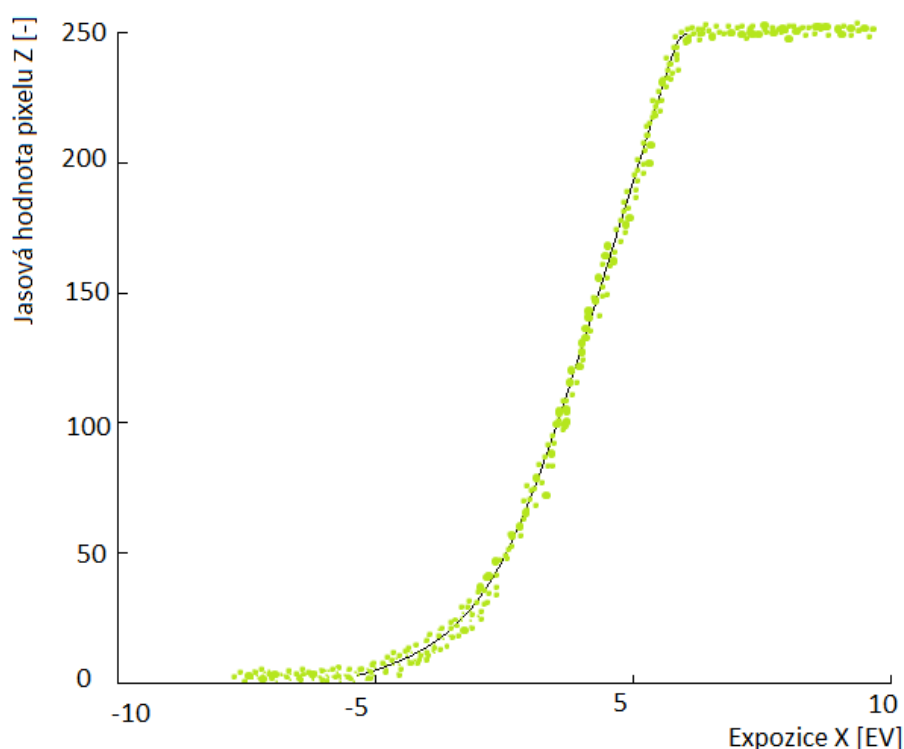
Tento algoritmus je popsán v práci [17] autory P.E. Debevecem a J. Malikem. Jádrem metody spočívá v analýze vstupních obrazů s odkrokovými hodnotami expozice a následném modelování vysokého dynamického kontrastu užitím jasových map. Z jednotlivých bodů si algoritmus rekonstruuje křivku odezvy snímače, která má zobrazovat rozdíl mezi zachycenými a výstupními hodnotami.

Prvním důležitým pojmem, na kterém je tato metoda založena, je fyzikální vlastnost zvaná reciprocita, která vyplývá z rovnice

$$X = E \times \Delta t, \quad (2.7)$$

kde X značí hodnotu expozice, E intenzitu ozáření a Δt expoziční čas. Upravením hodnot, např. zvýšením času na dvojnásobek a snížením intenzity ozáření na polovinu dostaneme tentýž výsledek expozice X . Tento princip reciprocity může selhat při extrémě vysokých, či nízkých hodnotách Δt (křivka bude mít ve krajových expozičních hodnotách velmi malé, nebo nulové hodnoty). Při fotografování LDR snímků, ze kterých se následně bude vytvářet HDR obraz, se postupuje dle tohoto principu. Hodnoty času závěrky se musí volit tak, aby hodnota expozice měla řadovou formu (tj. expoziční kroky musí být mezi jednotlivými snímky totožné). Z času závěrky a hodnoty expozice si dle vzorce 2.7 dopočítáme intenzitu osvětlení E .

Při zpracovávání světelných hodnot na snímači, dochází k nelinearitě, která je způsobena nesprávnou reprezentací obrazu na snímači. Když mají dvě hodnoty v reálné scéně určitý poměr jasových hodnot, snímač je pravděpodobně nedokáže zachytit v naprosto správném poměru. Proto existuje takzvaná křivka odezvy přístroje, která popisuje schopnost fotoaparátu (resp. snímače) správně zachytit dopadající světlo a reprezentovat ho pomocí pixelů. Graf popisující tuto křivku má logaritmickou osu s jasovou hodnotou pixelu a druhou osu stupnicí expozice. V praxi dochází k již zmíněné nelinearitě, a proto dochází ke špatnému mapování hodnot pixelů. U maximálních a minimálních hodnot („přepálené“ a podexponované oblasti) dochází k redukci rozdílů hodnot, a tedy k ztrátě důležité obrazové informace, viz obr. 2.1.



Obr. 2.1: Ukázka reálné odezvy snímače. Inspirováno v [17].

Pro výpočet osvětlení bodů scény E použijeme vzorec

$$E = \frac{X}{\Delta t}. \quad (2.8)$$

Z parametrů dané křivky následně zjistíme vlastnosti snímače např. dynamický rozsah či jeho linearitu. Dále musí být zachycen veškerý dynamický rozsah scény. Protože maximální a minimální hodnoty na snímači zkreslují výslednou informaci, je vhodné pracovat pouze s oblastí, bez těchto okrajových částí snímače, která se nazývá pracovní oblast¹. Abychom pokryli veškerý dynamický rozsah scény, je nutné,

¹z anglického working space

aby hodnota jasu všech bodů byla po přepočítání na expozici alespoň v jednom obraze v pracovní oblasti. Počet fotografií, které jsou pro zachycení dynamického rozsahu scény potřeba, vypočítáme dle rovnice

$$N = \frac{R}{F}, \quad (2.9)$$

kde R značí dynamický rozsah scény a F pracovní oblast snímače.

Výpočet se skládá ze dvou hlavních částí. První je rekonstrukce křivky odezvy senzoru, druhý je výpočet jasové mapy, který bude popsán dále. Předpokládáme, že scéna je statická a proces snímání fotografií je dostatečně rychlý abychom změny světla mezi jednotlivými snímky mohly ignorovat. Nelineární funkce, která popisuje křivku snímače, vypočítáme dle vzorce 2.8. Pomocí rovnice

$$Z_{ij} = f(E_i \Delta t_j), \quad (2.10)$$

kde Z_{ij} jsou hodnoty pixelů s prostorovými indexy i a indexy expozičních časů Δt_j . Protože předpokládáme, že f je monotónní funkce, můžeme rovnici přepsat jako

$$f^{-1}(Z_{ij}) = E_i \Delta t_j. \quad (2.11)$$

Zlogaritmováním obou stran rovnice dostaneme

$$\ln f^{-1} Z_{ij} = \ln E_i \ln \Delta t_j. \quad (2.12)$$

Pro zjednodušení si definujeme funkci $g = \ln f^{-1}$. Dostaneme funkci

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j. \quad (2.13)$$

V této rovnici Z_{ij} a Δt_j známe, neznámými jsou hodnota expozice E_i a funkce g , u které předpokládáme, že je také monotónní. Je potřebné si uvědomit, že obnovením funkce g dostaneme konečnou množinu hodnot $g(z)$, kde Z jsou hodnoty jasu u pixelů. Řekněme, že Z_{min} a Z_{max} jsou nejmenší a největší hodnota Z , N je počet bodů obrazu a P je počet vstupních fotografií. Další postup můžeme popsat jako hledání hodnot $(Z_{max} - Z_{min} + 1)$ z množiny $g(Z)$ a N hodnot $\ln E_i$, které minimalizují tuto kvadratickou funkci

$$\mathcal{O} = \sum_{i=1}^N \sum_{j=1}^N [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2, \quad (2.14)$$

kde první část vychází z předešlých rovnic. Druhá část rovnice slouží pro vyhlazení, kterou pro diskrétní použití vypočítáme dle rovnice

$$g''(z) = g(z-1) - 2g(z) + g(z+1). \quad (2.15)$$

Hodnota λ značí míru vyhlazení a měla by být volena v závislosti na předpokládaném množství šumu v hodnotách Z_{ij} .

Dále může nastat případ, že se body na okraji křivky (tzn. přepočítané hodnoty ozáření) objeví v pracovním rozsahu ve více než jednom snímku. Vyřešit tento problém lze metodou, která dané body třídí. Nejlepším způsobem je vážený průměr jasových hodnot, dle rovnice

$$w(z) = z - Z_{min}, \text{ pro } z < \frac{1}{2} \times (Z_{min} + Z_{max}) \quad (2.16)$$

a

$$w(z) = Z_{max} - z, \text{ pro } z > \frac{1}{2} \times (Z_{min} + Z_{max}). \quad (2.17)$$

Z_{min} a Z_{max} zde udávají maximální a minimální hodnoty pixelů (0 nebo 255). Největší váhu mají pixely, které se svojí hodnotou blíží středu. Dosazením do rovnice 2.14 dostaneme rovnici

$$\mathcal{O} = \sum_{i=1}^N \sum_{j=1}^N \{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2, \quad (2.18)$$

Pro výpočet nemusíme použít všechny pixely ze vstupních obrazů. Zajištění minimálního dostatečného počtu pixelů zjistíme z nerovnice

$$N(P - 1) > (Z_{max} - Z_{min}). \quad (2.19)$$

Všechny výpočty je nutné použít pro každý barevný kanál zvlášť.

Konstrukce jasové mapy

Jakmile je křivka odezvy obnovena, můžeme jednoduše konvertovat hodnoty pixelů na jasové hodnoty. Všimněme si, že křivka může být použita pro určení hodnoty jasu obrazu ze vstupní sekvence i přesto, že nebyl použit pro obnovení křivky odezvy. Protože hodnotu Δt_j známe, pak výsledná rovnice je

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j. \quad (2.20)$$

Výsledná hodnota jasu se vypočítá ze všech totožných pixelů vstupních obrazů. Znovu použijeme váhovací funkci 2.16 a 2.17, abychom dali větší váhu pixelům nacházejícím se nejbližše středu křivky odezvy funkce:

$$\ln E_i = \frac{\sum_{j=1}^P w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(Z_{ij})}. \quad (2.21)$$

Výsledná jasová mapa vypočítána dle tohoto vzorce, se dále musí zpracovat do zobrazitelné podoby, což se provede mapováním tonality.[17]

2.3 Tónové mapování

Globální metody

U těchto metod se výpočty provádí zvlášť pro každý jednotlivý pixel na základě zjištěné jisté globální veličiny, které neberou v potaz informace okolních pixelů. Zakládají se například na hodnotách minimální a maximální světelnosti, nebo průměrné světelnosti. Hodí se pro obrazy, kde jejich dynamický rozsah není příliš velký. Tyto metody jsou jednoduché a tedy i výpočtově méně náročné než metody lokální, avšak mají i své nevýhody, a to především ztráta detailů. [12] [16]

2.3.1 Lineární tónování

Nezákladnější metoda pro převod vysokobitového obrazu HDR zpět do 8 bitového, který lze zobrazit. Jedná se o metodu, kde všechny hodnoty jsou namapovány rovnoměrně, tj. hodnoty se podělí stejným dělitelem a žádné další procesy se neprovádějí. Při této metodě dochází k velké ztrátě informace, především dojde k potlačení lokálních kontrastů a tedy ztrátě detailů obrazu. Lineární namapování lze popsat rovnicí

$$Z_i = \frac{A_i}{2^{H-1}} - 1, \quad (2.22)$$

kde Z_i značí výstupní bod se souřadnicí i , L je bitová hloubka výsledného LDR snímku, H je bitová hloubka vstupního HDR obrazu a A značí vstupní bod se souřadnicí i .

2.3.2 Adaptivní logaritmické mapování

Autoři této metody představili v článku [20] rychlou a kvalitní techniku tónového mapování pro zobrazení výsledného LDR snímku. Metoda je založena na logaritmické kompresi jasových hodnot, imitující reakci lidského oka na světlo. Díky tomuto předpokladu, že lidské oko vnímá jas logaritmicky, je u výsledných obrazů dosaženo kvalitních výsledků s dobře vykreslenými detaily a kontrastem scény.

Obecně používaný pojem jas B je analogií k odezvě lidského zrakového systému na podněty světla L . Tato odezva je však nelineární a lze ji aproximovat dle

$$B = k_1 \ln \left(\frac{L}{L_0} \right), \quad (2.23)$$

kde L_0 je jas scény a k_1 je konstanta. Jas L_d pro každý pixel je vypočítána dle

$$L_d = \frac{\log(L_w + 1)}{\log(L_{max} + 1)}, \quad (2.24)$$

kde L_w je jasová hodnota vstupního pixelu a L_{max} je maximální hodnota jasu v dané scéně.

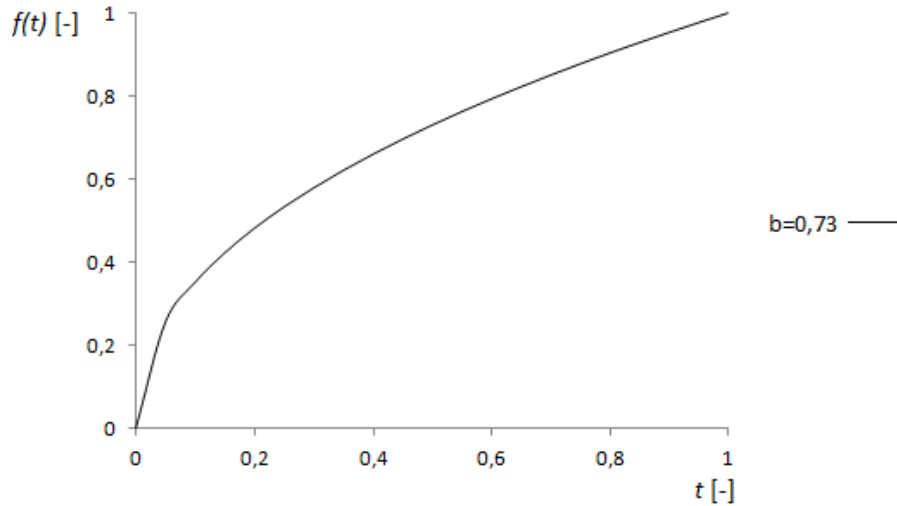
Základ adaptivního logaritmického mapování vychází z toho poznatku o lidském zrakovém systému. Jasové hodnoty jsou interpolovány v rozsahu od $\log_2(L_w)$ do $\log_{10}(L_w)$. Tento proces poskytne zachování kontrastu v tmavých a středně tmavých oblastech (báze logaritmu rovna 2) a zároveň maximální kompresi oblastí s vysokým jasnem (báze logaritmu rovna 10). Následující vzorec založený na vlastnostech logaritmu umožňuje libovolný výběr logaritmické báze

$$\log_{báze}(x) = \frac{\log(x)}{\log(báze)}. \quad (2.25)$$

Pro hladkou interpolaci mezi logaritmickými bázemi použijeme Perlin a Hoffertovu funkci, takzvanou bias funkci. Parametry b mapuje vstupní hodnotu na nižší, nebo vyšší.

$$bias_b(t) = t^{\frac{\log(b)}{\log(0,5)}}. \quad (2.26)$$

Na obr. 2.2 můžeme vidět, že hodnota $b = 0,73$ poskytuje přibližně stejné namapování jako gamma korekce s hodnotou $\gamma = 2,2$.



Obr. 2.2: Bias funkce s hodnotou $b=0,73$

Rovnice 2.27 je použita pro výpočet výsledné hodnoty L_d pro každý pixel a je odvozena dosazením rovnice 2.26 do jmenovatele rovnice 2.25.

$$L_d = \frac{L_{dmax} \times 0,01}{\log_{10}(L_{wmax} + 1)} \times \frac{\log(L_w + 1)}{\log(2 + ((\frac{L_w}{L_{wmax}})^{\frac{\log(b)}{\log(0,5)}}) \times 8)} \quad (2.27)$$

Hodnota L_{wmax} je rovna maximálnímu jas, který je možné zobrazit na výstupním zařízení a L_{dmax} je faktor pro přispůsobení se tomuto zařízení. Nejvhodnější je použít hodnotu $L_{dmax} = 100cd/m^2$. Hodnotu b je vhodné zvolit mezi 0,7 a 0,9, autoři použili hodnotu $b = 0,85$.

Dalším krokem postupu je aplikování gamma korekce na namapovaná data pro kompenzování nelinearity zobrazovacích zařízení, standardně hodnota $\gamma = 2,2$. [20]

2.3.3 Reinhardův operátor

Tato metoda je založena na redukci dynamického rozsahu inspirovanou fotoreceptory nacházejícími se v lidském zrakovém systému. Model pracuje odděleně s každým kanálem zvlášť a obsahuje chromatické transformace.

Algoritmus je založen na automatické adaptaci fotoreceptorů na osvětlení. Výslednou jasovou hodnotu jednoho barevného elementu V lze modelovat jako

$$V = \frac{I}{I + \sigma(I_a)}, \quad (2.28)$$

kde I_a je úroveň adaptace, I hodnota jasu bodu a funkce $\sigma(I_a)$ je popsána rovnicí

$$\sigma(I_a) = (fI_a)^m, \quad (2.29)$$

kde m je vypočteno jako:

$$m = 0,3 + 0,7k^{1,4}. \quad (2.30)$$

Hodnotu klíče k zjistíme dle:

$$k = \frac{\log(L_{max}) - \log(L_{av})}{\log(L_{max}) - \log(L_{min})}, \quad (2.31)$$

kde L_{av} je průměrná hodnota jasu a hodnoty L_{max} a L_{min} jsou maximální resp. minimální hodnoty jasu. Jas je určen ze všech barevných kanálů dle následujícího poměru

$$L = 0,2125 \times I_r + 0,7154 \times I_g + 0,0721 \times I_b. \quad (2.32)$$

Hodnoty I_r , I_g a I_b značí hodnotu jasu pro každý barevný kanál.

Rovnice 2.30 byla autory zjištěna jako nejvýhodnější a nejpraktičtější pro velkou škálu vstupních dat.

Parametr f je vypočten jako

$$f = \exp(-f'). \quad (2.33)$$

Hodnota f' se volí z rozmezí 8 a -8 a ovlivňuje výslednou intenzitu.

Protože algoritmus pracuje s každým barevným kanálem zvlášť, ve výsledném snímku mohou vznikat změny v barevném ladění. Pro odstranění těchto barevných artefaktů slouží výpočet adaptace I_a . Vypočítá se jako váhový součet jasu L a intenzity I pro každý kanál:

$$I_a = cI_{r|g|b} + (1 - c)L. \quad (2.34)$$

Barevná korekce je kontrolována váhovým faktorem c jehož hodnoty jsou z rozsahu $[0, 1]$. Při $c = 1$ se pracuje s každým kanálem odděleně a je dosaženo největší barevné korekce, při $c = 0$ se barevná korekce provádí globálně. V některých případech je vhodné při barevné korekci brát v potaz nejen intenzitu pixelu samotného $I_{r|g|b}$, ale i průměrnou intenzitu kanálu $I_{r|g|b}^{\text{av}}$. Poměr těchto dvou hodnot lze měnit pomocí parametru a opět z rozsahu $[0, 1]$:

$$I_a = aI_{r|g|b} + (1 - a)I_{r|g|b}^{\text{av}}. \quad (2.35)$$

Při $a = 1$ závisí adaptace pouze na intenzitě pixelu, při $a = 0$ je adaptace globální. Kombinací rovnic 2.34 a 2.35 vypočítáme výslednou adaptaci I_a dle:

$$I_a^{\text{local}} = cI_{r|g|b} + (1 - c)L, \quad (2.36)$$

$$I_a^{\text{global}} = cI_{r|g|b}^{\text{av}} + (1 - c)L^{\text{av}}$$

a

$$I_a = aI_a^{\text{local}} + (1 - a)I_a^{\text{global}}.$$

Posledním krokem metody je normalizovat výsledné hodnoty V na požadovaný rozsah a aplikovat gamma korekci.[22]

Lokální metody

U lokálních operátorů výpočet pracuje s hodnotami okolních pixelů. Jsou založeny na modelu lidského vnímání, nebo na matematických principech. Typickým problémem je halo efekt, který vzniká chybným zvolením parametrů algoritmu.

2.3.4 Metoda bilaterálního filtrování

Základem této metody je rozložení vstupních snímků na základní vrstvu a vrstvu s detaily. Základní vrstvu získáme pomocí bilaterálního filtru, který, zjednodušeně řečeno, vyhladí obraz při zachování velkých kontrastních rozdílů - hran. Vhodné je počítat s každou barevnou složkou odděleně. Ještě lepším řešením je oddělit jasovou složku od barevné a na tu užít bilaterální filtr.

Metoda popsána autory Tomasi a Manduchi [24] počítá hodnoty pixelů v závislosti na jejich blízkosti a podobnosti hodnoty. Pro zjištění prostorové blízkosti pixelů v obraze je použita dolní propust. Váha vzdálenosti není určena lineárně, ale klesá s Gaussovou funkcí. Myšlenkou je, že blízké pixely nebudou mít velké rozdíly hodnot intenzity a proto je možné je zprůměrovat. Při výpočtu je také odstraněn šum.

Problém nastane v případě velkých kontrastních změn v obraze - hran, protože by došlo k rozmazání a tím ke ztrátě detailů. Jedním z řešení je použití anizotropní difúze, což je filtr, který pro okolí počítaného bodu zjišťuje lokální prudké změny

- gradienty. Po tomto zjištění se teprve provede výpočet okolí, na které se použije bilaterální filtr.

Bilaterální filtr se zachováním hran je další variantou a funguje za pomoci druhé váhovací funkce. Nyní si vysvětlíme jeho princip. Přestože autoři v [25] uvažují obraz jako spojitou funkci, zde budeme pracovat se snímkem jak je tomu v realitě - s dvourozměrnou diskretní maticí bodů. Váha jednotlivých bodů se u obou funkcí vypočítává na základě Gaussovy funkce. Výsledná intenzita J pixelu s je vypočtena jako normalizovaná suma pixelů z okolí váhovaných dle Gaussovy funkce, kde váha je dána hodnotou vzdálenostní a podobnostní funkce:

$$J_s = \frac{1}{k(s)} \times \sum_{p \in \omega} f(s, p)g(I_s, I_p)I_p, \quad (2.37)$$

kde

$$k(s) = \sum_{p \in \omega} f(s, p)g(I_s, I_p), \quad (2.38)$$

kde prostorová váhovací funkce je označena f , podobnostní g , ω značí veškeré body z okolí pixelu zahrnuté do výpočtu, I_s značí vstupní hodnotu jasu bodu s a I_p hodnotu jasu bodu p . Parametry funkce g jsou prostorové informace a hodnoty jasu jsou parametry funkce f . Jak již bylo zmíněno, bilaterální filtr používá pro váhování Gaussovu funkci:

$$f(s, p) = e^{-\frac{1}{2} \left(\frac{r(s, p)}{\sigma_d} \right)^2}, \quad (2.39)$$

kde $d(s, p) = |s - p|$ je absolutní vzdálenost mezi pixely s a p a σ_d je hodnota rozptylu. Obdobně je vypočtena podobnostní funkce g :

$$g(I_s, I_p) = e^{-\frac{1}{2} \left(\frac{r(I_s, I_p)}{\sigma_r} \right)^2}, \quad (2.40)$$

kde $r(I_s, I_p) = |I_s - I_p|$ je absolutní hodnota jasu mezi pixely s a p a σ_r je hodnota rozptylu.

Tento bilaterální filtr se může aplikovat na každou barevnou složku zvlášť, nebo pro urychlení výpočtu pouze na jasovou složku. Parametry filtru můžeme nastavovat pomocí hodnot σ_d a σ_r . Při nastavení σ_d určujeme, jak velké okolí bude zahrnuto do váženého průměru. Větší hodnota σ_d bude mít za následek rozmazanější obraz. Hodnota σ_r určuje velikost škály intenzity pixelů, které se na výpočtu podílí. Hodnota ovlivňuje lokální kontrast ve výsledném snímku. V implementační části bude demonstrováno různé nastavení těchto parametrů.

Bilineární filtr je jádrem této metody, jejichž největším přínosem je skoro úplné odstranění halo efektu z výsledných snímků, tak typického pro lokální tónové mapování. [25][24]

2.3.5 Gradientní doménová komprese

Tato metoda tónového mapování představená v článku [26] je dle autorů efektivní, robustní a snadno použitelná. Mapování jasových hodnot je provedeno pomocí gradientního pole, na jehož základě probíhá snižování velkých hodnot gradientů. Zakládá se na předpokladu, že velké změny jasu v HDR obraze mají za výsledek velké hodnoty gradientů. Výsledný obraz je získán pomocí Poissonovy rovnice aplikované na modifikované pole gradientů. I přesto, že tato metoda přímo neuvažuje a nepočítá s odezvou lidského zrakového systému, měla by poskytnout velkou dynamickou kompresi při zachování žádoucích detailů a vyhnoutí se typickým problémům při tónovém mapování jako halo artefakty, ztráta kontrastu, inverze tónů atd.

Jinak řečeno, metoda spoléhá na předpoklad, že sítnice lidského vizuálního systému není až tak citlivá na maximální hodnotu jasu (spíše jej potlačuje), ale především na lokální změny intenzit jasu. Algoritmus je založen na jednoduchém principu, a to že jakákoliv velká změna jasu v HDR obraze musí vést k velkým hodnotám gradientu. Drobné detaily, jako jsou struktura a plochy budou mít hodnoty gradientů mnohem menší. Myšlenka je najít tyto velké hodnoty gradientů a určitým způsobem je ztlumit při současném zachování jejich směru. Ztlumení gradientů by mělo být progresivní, větší hodnoty gradientů by měly být zmenšeny více než menší hodnoty. Výsledný LDR obraz se poté rekonstruuje ze sníženého pole gradientů.

Protože lidské oko vnímá jas logaritmicky, je také vhodné všechny výpočty provádět na logaritmech jasů, než na hodnotách jasů samotných.

Na začátku je vhodné si vysvětlit algoritmus v 1D - jednorozměrném prostoru. Budeme uvažovat HDR obraz jako jednorozměrnou funkci. Logaritmus jasu HDR obrazu si označíme jako funkci $H(x)$. Jak bylo popsáno dříve, cílem je zmenšit vysoké hodnoty změn jasu v H při zachování lokálních malých změn jasu. Toho dosáhneme pomocí vhodné prostorově variantní tlumicí funkce $\Phi(x)$ a velikosti derivace $H'(x)$ dle rovnice:

$$G(x) = H'(x) \times \Phi(x), \quad (2.41)$$

kde $G(x)$ je výsledná tlumicí funkce jasových hodnot. Funkce $G(x)$ je stejná jako funkce $H'(x)$, pouze má pozmeněné hodnoty v závislosti na funkci $\Phi(x)$, která je navržena aby zmenšovala větší hodnoty funkce $H'(x)$ více než menší. Výpočet funkce $\Phi(x)$ bude popsán později. Nyní můžeme rekonstruovat výslednou funkci jasových hodnot $I(x)$ se sníženým dynamickým rozsahem zintegrováním funkce dle rovnice:

$$I(x) = C + \int_0^x G(t)dt, \quad (2.42)$$

kde C je konstanta. Výsledek je potřeba umocnit z důvodu navrácení jasu.

Protože obraz je dvourozměrnou funkcí, je potřeba i k výpočtu takto přistupovat. Místo derivace funkce $H(x, y)$ je nyní už nutné počítat s hodnotami gradientů $\nabla H(x, y)$.

Aby se zamezilo prostorové deformaci obrazu, při změnách hodnot gradientů musíme zachovat jejich směr. Výslednou 2D tlumící funkci získáme dle:

$$G(x, y) = \nabla H(x, y) \times \Phi(x, y). \quad (2.43)$$

U dvojrozměrné funkce nelze získat obraz I se sníženým dynamickým rozsahem pouhým zintegrováním funkce $G(x, y)$, protože gradient $\nabla I = \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}$ musí splňovat podmínku

$$\frac{\partial^2 I}{\partial x, \partial y} = \frac{\partial^2 I}{\partial y, \partial x}, \quad (2.44)$$

která je u $G(x, y)$ zřídka splněna. Řešením je vyhledání nejbližší podobné funkce metodou nejmenších čtverců. Jinak řečeno, funkce by měla minimalizovat integrál

$$\int \int F(\nabla I, G) dx dy, \quad (2.45)$$

kde

$$F(\nabla I, G) = \|\nabla I - G\|^2 = \left(\frac{\partial I}{\partial x} - G_x \right)^2 + \left(\frac{\partial I}{\partial y} - G_y \right)^2. \quad (2.46)$$

V souladu s variačním principem musí funkce $I(x, y)$ která minimalizuje integrál splňovat Eulerovu-Lagrangeovu rovnici

$$\frac{\partial F}{\partial I} - \frac{d}{dx} \frac{\partial F}{\partial I_x} - \frac{d}{dy} \frac{\partial F}{\partial I_y} = 0. \quad (2.47)$$

Substitucí za F získáme rovnici

$$2 \times \left(\frac{\partial^2 I}{\partial x^2} - \frac{\partial G_x}{\partial x} \right) - 2 \times \left(\frac{\partial^2 I}{\partial y^2} - \frac{\partial G_y}{\partial y} \right) = 0 \quad (2.48)$$

Jednoduchou úpravou rovnice (podělení 2 a přeskupením proměnných) dostaneme Poissonovu rovnici

$$\nabla^2 I = \text{div} G, \quad (2.49)$$

kde ∇^2 je Laplaceův operátor definovaný jako:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2.50)$$

a $\text{div} G$ je divergence vektorového pole definovaná jako:

$$\text{div} G = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}. \quad (2.51)$$

Numerické řešení této lineární parciální diferenciální rovnice bude popsáno v implementační části.

Tlumící funkce gradientů $\Phi(x, y)$

V této části si uvedeme postup výpočtu tlumící funkce gradientů, která snižuje jejich hodnotu v HDR obrazu o faktor $\Phi(x, y)$. Jak bylo zmíněno, požadujeme aby útlum byl progresivní. Reálné obrazy obsahují hrany různých měřítek. Abychom spolehlivě zjistili všechny významné přechody jasu (hrany) musíme použít víceúrovňové hledání hran na různých rozlišeních. Nelze ovšem zmenšit každý gradient u každého rozlišení který je nalezen, výsledkem by bylo nežádoucí tvoření halo artefaktů. Proto je nalezný útlum pomocí hodnot gradientů z každé úrovně aplikován na snímek s plným rozlišením, čímž se vyhneme halo efektu.

V prním kroku se sestaví Gaussova pyramida H_k , kde k je počet úrovní v rozsahu H_0, H_1, \dots, H_d , kde H_0 je snímek s nejvyšším rozlišením a H_d je snímek s nejmenším rozlišením, jehož výška, nebo šířka, je nejméně 32 pixelů. Na každé úrovni pyramidy se gradienty vypočítají pomocí centrálních rozdílů dle

$$\nabla H_k(x, y) = \left(\frac{H_k(x+1, y) - H_k(x-1, y)}{2^{k+1}}, \frac{H_k(x, y+1) - H_k(x, y-1)}{2^{k+1}} \right). \quad (2.52)$$

Pro každou úroveň k se vypočítá tlumící faktor $\varphi(x, y)$ pro každý pixel na základě velikosti gradientu dle

$$\varphi_k(x, y) = \frac{\alpha}{\|\nabla H_k(x, y)\|} \left(\frac{\|\nabla H_k(x, y)\|}{\alpha} \right)^\beta, \quad (2.53)$$

kde parametr α určuje hranici, které gradientní změny zůstanou beze změny a které ne. Parametr β určuje míru ztlumení. Gradienty větších změn jsou ztlumeny (za předpokladu $\beta < 1$, zatímco gradienty hodnot menších než α jsou mírně zvětšeny. Výsledky metody prokázaly, že nejlepší nastavení hodnoty α je na 0,1 násobek průměrné velikosti gradientu a β mezi 0,8 a 0,9. Velikost výsledné tlumící funkce $\Phi(x, y)$, která dle velikosti snižuje hodnotu těchto gradientů ve výsledném obraze, je pro snímek s plným rozlišením vypočtena od nejmenší úrovně po nejvyšší pomocí $\varphi(x, y)$ pro každou úroveň, kdy se při přechodu do vyšší úrovně použije lineární interpolace. Pomocí rovnic lze tento proces popsat jako:

$$\Phi_d(x, y) = \varphi_d(x, y), \quad (2.54)$$

$$\Phi_k(x, y) = L(\Phi_{k+1})(x, y)\varphi_k(x, y) \quad (2.55)$$

a

$$\Phi(x, y) = \Phi_0(x, y), \quad (2.56)$$

kde d je úroveň nejnižšího rozlišení, $\Phi_k(x, y)$ označuje velikost útlumu k -té úrovně a L je lineární interpolace zvyšující velikost na dvojnásobek. Výsledkem je tlumící hodnota gradientu $\Phi_k(x, y)$ pro každý pixel vypočtena ze všech úrovní snímků Gaussovy pyramidy. Tento pyramidový výpočet zaručuje nalezení všech gradientních změn nezávisle na jejich velikosti ve scéně snímku. [26]

2.4 Metoda Exposure Fusion

Pro tuto metodu je vyhrazena samostatná podkapitola, protože její princip se liší od klasického HDR postupu. Proces vytvoření výsledného snímku vynechává konvertování vstupní sekvence do vysokobitového obrazu HDR, což velice zjednodušuje celý proces. Tímto se vyhýbáme výpočtu odezvy snímače a proto ani nepotřebujeme znát hodnoty expozice. Je dokonce možné použít fotografie s bleskem. Technika využívá jednoduché výpočty za pomoci hodnot saturace a kontrastu u vstupních snímků. Výsledná kvalita je srovnatelná s operátory tónového mapování. Na začátku je potřeba si oblasti scény u všech snímků rozdělit na důležitější a méně důležité, dle hodnoty správné expozice. Těmto správným oblastem se bude přiřadit větší váha než ostatním. Tyto oblasti najdeme z následujících hodnot.

- Kontrast: použijeme Laplaceův filtr na šedotónové kopie snímků, z nichž zjistíme absolutní hodnotu odezvy tohoto filtru. Tato hodnota nám poskytne jednoduchý indikátor C - kontrast.
- Saturace: když je fotografie pořízena s delším časem expozice, výsledné barvy jsou nejprve desaturovány a následně oříznuty. Saturace je žádoucí, aby obraz vypadal živě. Hodnotu saturace C vypočítáme jako směrodatnou odchylku pro každý pixel v kanálech RGB.
- Správně exponované oblasti: největší váhu chceme logicky dát pixelům v oblasti, kde jsou správně exponovány. Oblastem, kde se hodnoty jasu pixelů blíží minimální nebo maximální hodnotě, chceme dát naopak nejmenší váhu. Využita je zde Gausova funkce. Čím blíže jsou pixely i (respektive jejich jasová hodnota) hodnotě 0,5, tím větší váhu mají:

$$f(i) = \exp\left(-\frac{(i - 0,5)^2}{2\sigma^2}\right), \quad (2.57)$$

kde hodnota $\sigma = 0,2$. Pro každý barevný kanál je nutné Gausovu funkci použít zvlášť.

Pro každý pixel zkombinujeme tyto tři hodnoty pomocí skalární váhové mapy. Každou z hodnot můžeme ovlivnit pomocí váhové funkce

$$W_{ij,k} = (C_{ij,k})^{\omega_C} \times (S_{ij,k})^{\omega_S} \times (E_{ij,k})^{\omega_E}, \quad (2.58)$$

kde ω_C, ω_S a ω_E jsou váhovací exponenty hodnot kontrastu, saturace a expozice. Hodnoty (ij, k) reprezentují polohu pixelu (i, j) v k -tém obraze. Pokud se hodnota ω v některém exponentu rovná nule, není brána v potaz u výpočtu.

První krok výpočtu je normalizování N váhových map vstupních snímků jednotlivě pro každý pixel:

$$\hat{W}_{ij,k} = \left[\sum_{k'=1}^N W_{ij,k'} \right]^{-1} W_{ij,k}. \quad (2.59)$$

Výstupní obraz R je získán sloučením normalizovaných váhových map všech vstupních snímků:

$$R_{ij} = \sum_{k=1}^N \hat{W}_{ij,k} I_{ij,k}, \quad (2.60)$$

kde I_k je k -tý vstupní snímek. Tento jednoduchý výpočet není dostačující z důvodu rozdílných absolutních světelných intenzit ve vstupních snímcích.

Proto je použita technika propojení dvou snímků pomocí alfa masky a to na několika rozlišeních, kdy se snímky postupně pyramidově zmenšují. Prvním krokem je rozložení na tzv. Laplaceovu pyramidu, která obsahuje pásmově filtrované verze snímků v různých měřítkách a Gaussovou pyramidu. Spojení dvou snímků se vždy provádí na stejné úrovni rozlišení. N normalizovaných váhových map získaných z N snímků použijeme jako alfa masky. l -tou úroveň snímku A v Laplaceově pyramidě definujeme jako $\mathbf{L}\{A\}^l$ a l -tou úroveň snímku B v Gaussově pyramidě jako $\mathbf{G}\{B\}^l$. Každá úroveň výsledné Laplaceovy pyramidy l je tedy vypočítána jako vážený průměr původních Laplaceových rozkladů s Gaussovou pyramidou:

$$\mathbf{L}\{R\}_{ij}^l = \sum_{k=1}^N \mathbf{G}\{\hat{W}\}_{ij,k}^l \mathbf{L}\{I\}_{ij,k}^l. \quad (2.61)$$

Finálním krokem je zpětné složení pyramidy $\mathbf{L}\{R\}^l$ do výsledného obrazu R . Protože jsou jednotlivé úrovně pyramidy počítány odděleně, u výsledného obrazu nedochází k poklesu ostrosti hran. Dále je pro dobré výsledky vhodné počítat odděleně každý barevný kanál.

Výsledky metody jsou srovnatelné s ostatními metodami pro HDR obrazy. Velkým pozitivem je její rychlost, z důvodu relativně jednoduchých výpočtů oproti např. gradientním metodám, které pro svůj výpočet vyžaduje řešení parciálních derivací. Dále pro výpočet není potřebná odezva snímače, jak je tomu u metody Debevec. Metoda je robustní proti změně vzhledu, avšak může dojít k barevným posunům, nebo mírnému snížení kontrastu. Nejlepších výsledků (až na výjimky) bylo dosaženo s hodnotami $\omega_C = \omega_S = \omega_E = 1$. [21]

3 NÁVRH A IMPLEMENTACE

3.1 Použitý software

- MS Visual Studio 2012 - Pro návrh a implementaci metod a algoritmů bylo použito vývojové prostředí MS Visual Studio 2012 a jazyk C++.
- OpenCV 3.0.0 - Knihovna OpenCV¹ je volně šiřitelná, multiplatformní zaměřena na zpracování obrazu a videa. Je určena zároveň pro akademické i komerční využití. Základy této knihovny položila společnost Intel. Má C++, C, Python a Java rozhraní. Knihovna je psaná v jazyku C++ a je schopná pracovat pod operačními systémy Windows, Mac OS i Linux. Je distribuována pod BSD licencí. Byla použita verze 3.0.0.
- LibRaw - Knihovna LibRaw² je volně šiřitelná knihovna pro čtení a zpracování RAW snímků z digitálních fotoaparátů. Podporuje velkou škálu formátů od různých výrobců (CRW/CR2, NEF, RAF, DNG a další). V programu byla použita verze LibRaw 0.18-201604.
- Qt 5.2.1 - Knihovna Qt³ je multiplatformní knihovna pro tvoření aplikací s grafickým rozhraním. V aplikaci byla použita pro intuitivnější ovládání a lepší zobrazení výsledných snímků.
- CMake 3.5.0 - CMake³ je volně šiřitelný software používaný pro překlad a kompilaci pro vytvoření adresářové struktury a zdrojových souborů s použitím konkrétních překladačů. Pro překlad knihovny Qt byla použita verze CMake 3.5.0.

3.2 Struktura programu

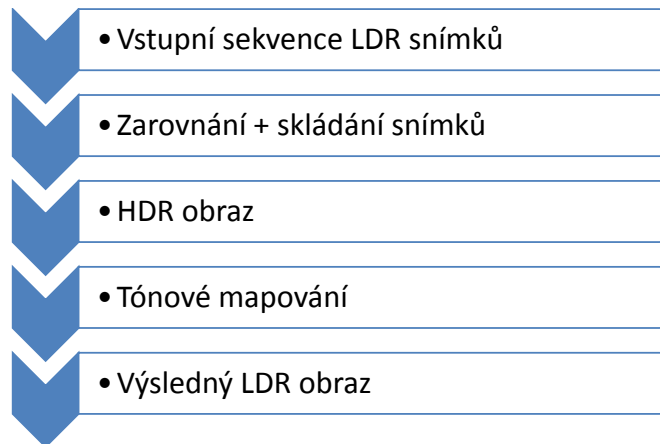
V této části si popíšeme základní strukturu algoritmu programu. Proces vytvoření obrazu s vysokým dynamickým rozsahem je vidět na vývojovém diagramu 3.1. Jako první je načtena vstupní sekvence LDR obrazů se stejnou scénou a různou hodnotou expozice. Dalším krokem je jejich zarovnání a následné složení zvolenou metodou do formátu HDR. Abychom si mohli tento formát zobrazit na běžném zobrazovacím zařízení s 8 bitovou hloubkou, je nutné použít určitou metodu mapování tonality pro získání výsledného samostatného LDR obrazu. Jednotlivé kroky algoritmu si nyní blíže popíšeme. Základ programu je inspirován z článku [19], kde jsou vysvětleny základy používání funkcí pro práci s HDR obrazy z knihovny OpenCV.

¹Dostupná z <http://opencv.org/>

²Dostupná z <http://www.libraw.org/>

³Dostupná z <https://www.qt.io/>

³Dostupný z <https://cmake.org/>



Obr. 3.1: Vývojový diagram procesu vytváření HDR

3.3 Vstupní sekvence

Pro testování vybraných algoritmů je použito několik sekvencí s různým počtem snímků, např. sada 15 fotek - okno - s hodnotou expozičních časů od 4 po $\frac{1}{4000}$ sekundy, kterou vidíme na obr. 3.2. Všechny sekvence lze najít v příloze A A.1. Vstupní sekvence byly voleny tak, aby jejich dynamický rozsah byl co největší a zároveň aby scéna obsahovala co nejvíce detailů a míst s rozdílnou hodnotou intenzity světla. Dalším faktorem pro výběr scény byla její neměnnost, z důvodu vyhnutí se nežádoucímu pohybu ve scéně. Nejmenší počet snímků, které je nutné použít pro



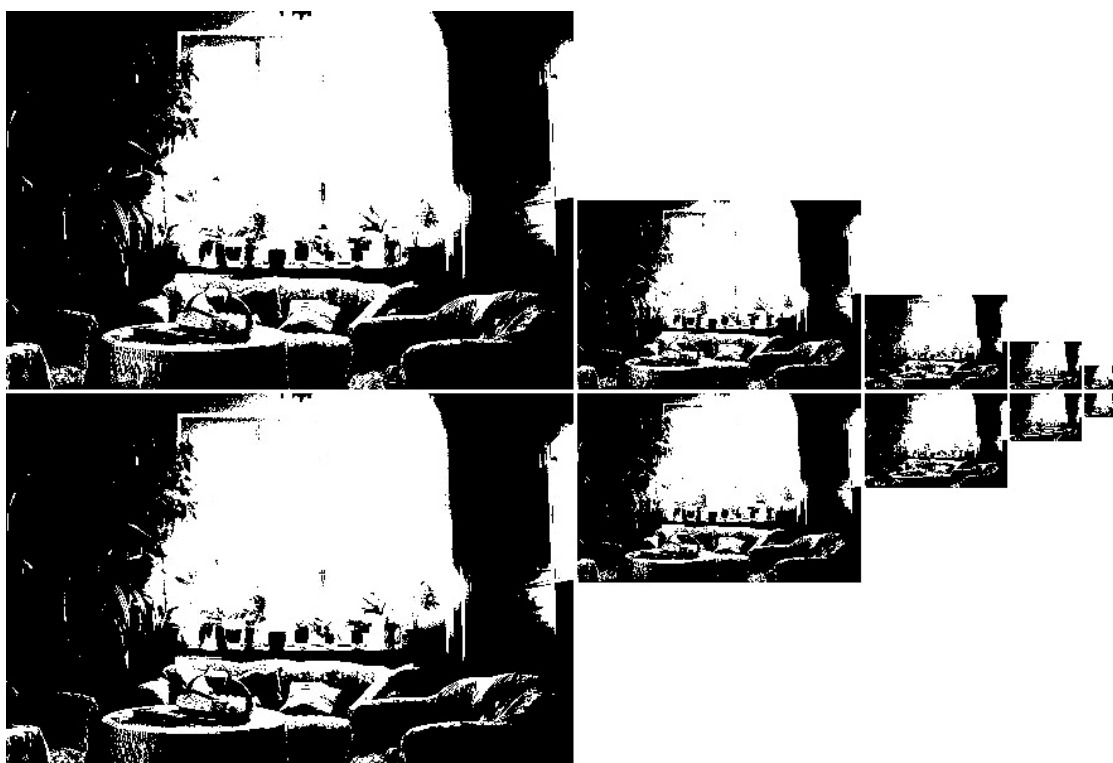
Obr. 3.2: Vstupní sekvence snímků, s odkrokovanou hodnotou expozice o 1 EV

vytvoření obrazu HDR je logicky 2, avšak s více snímky dosáhneme daleko lepšího zachycení veškerých detailů scény.

3.4 Zarovnání obrazu

3.4.1 Median threshold bitmap

Jako zarovnávací algoritmus byla zvolena metoda MTB - median threshold bitmap. Do algoritmu vstupují vždy dva snímky, jeden referenční a jeden zarovnávaný (zjišťuje se hodnota posunu druhého snímku vůči referenčnímu) Jako referenční je určen prostřední snímek ze vstupní sekvence. V případě sudého počtu je ze dvou středových snímků zvolen snímek s vyšší hodnotou expozice. Algoritmus se tedy vždy provede $N-1$ krát. Nejprve jsou obrazy převedeny do stupňů šedi a dále nalezena hodnota mediánu. Medián slouží jako hodnota threshold, při převodu do bitmapy. Na obrázku 3.3 vidíme dva snímky vybrané pro zarovnání, již převedené, nakopírované a zmenšené pro pyramidovou metodu. Pro zlepšení výsledků algoritmu je



Obr. 3.3: Snímky převedené do mediánové bitmapy

použito odstranění pixelů blízkých hodnotě mediánu o ± 4 , které se projevují jako šum. Obraz bez a s požitím této filtrovací funkce můžeme vidět na obrázku 3.4.

Samotné zarovnání výstupního obrazu je provedeno v dalším kroku, kterým je skládání vstupní sekvence do jediného snímku. Metoda MTB¹ je určena především pro zarovnávaní snímků, které jsou vůči sobě posunuty ve směrech osy x a y . Metoda

¹MTB - median treshold bitmap

však neřeší problém u snímků, kde jsou scény vůči sobě posunuty o rotaci. Hlavním předpokladem je, že posunutí snímků vůči sobě bude velmi malé, řádově jednotky pixelů a to právě pouze v horizontálním/vertikálním směru. Použitím detekčního



Obr. 3.4: Snímek bez filtrace hodnot blízkých mediánu (a), snímek s filtrací (b)

okna malé velikosti a následným zvětšováním by bylo možné aplikovat toto zarovnání jen na lokální oblasti, kde je posun způsoben pohybem ve snímané scéně (např. pohyb listů). Přestože je algoritmus nezávislý na délce expozice a tedy nemusí znát jejich expoziční hodnotu, v případech velkých a malých hodnot expozic, kde je většina plochy přepálená, nebo podexponovaná, algoritmus selhává. Je to způsobeno podstatou mediánu, kdy se jeho hodnota nachází právě v některé z těchto absolutních oblastí. Výpočet jednotlivých posunů mezi referenčním snímkem a ostatními je proveden pomocí metody `calculateShift`, které jsou následně použity v metodě `warpAffine`. Tato metoda provádí různé geometrické transformace, nemění obrazový obsah, ale pouze přetransformuje mříž obrazových bodů na jinou, v závislosti na hodnotách transformační matice, která je vstupním parametrem. Výsledné posuny jednotlivých snímků dosadíme do této transformační matice a následně metoda `warpAffine` provede posunutí, resp. zarovnání snímku.

3.5 Skládání obrazu

3.5.1 Metoda Debevec

Metoda skládání obrazů založena na výpočtu odezvy snímače s použitím informací o hodnotách expozice snímků a následném výpočtu jasové mapy. Tato metoda je tedy použitelná za předpokladu, že máme dostupná metadata² fotografií. Postup výpočtu odezvy snímače je popsán v kapitole 2.2.2. Metoda pro výpočet jasové mapy

²metadata - data o datech, u fotografií informace o hodnotách času závěrky, clony, ISO atd.

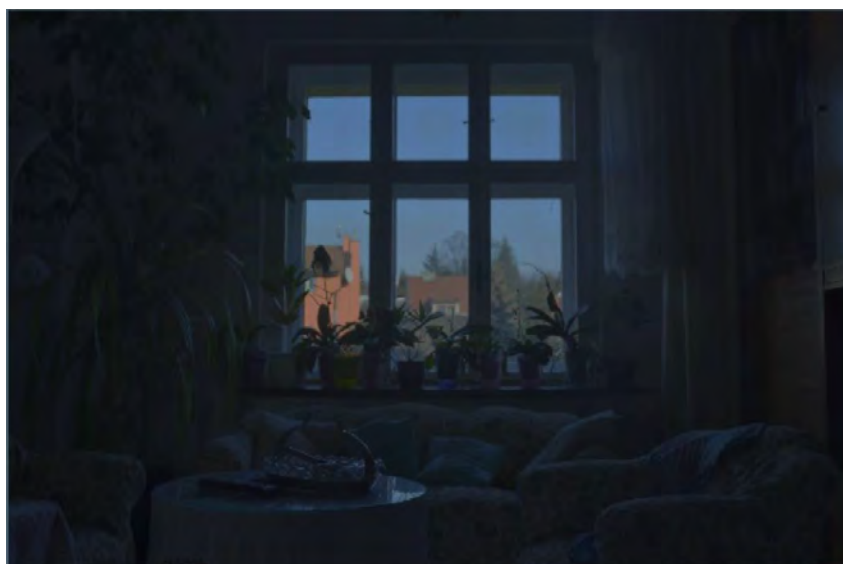
`createCalibrateDebevec()` má 2 vstupní parametry, kterými můžeme ovlivnit výpočet. Jako první je parametr λ značící míru vyhlazení, viz rovnice 2.18, jehož hodnota je ponechána na doporučené hodnotě 10. Dalším parametrem je počet pixelů, ze kterých se vypočítává odezva senzoru, jehož hodnota je nastavena na 70, což je dostatečně vysoká hodnota, aby splňovala dodržení podmínky pro minimální počet pixelů, viz rovnice 2.19.

Výsledný HDR obraz (jasová mapa) je tedy vytvořen pomocí metody `createMergeDebevec()` jako vážený průměr expozičních časů s ohledem na odezvu snímače.

3.6 Mapování tonality

3.6.1 Lineární tónování

Pro zobrazení HDR obrazu je třeba použít nějaký algoritmus pro namapování tonalit z vysokobitového obrazu zpět do 8 bitového formátu. Při použití nejjednoduššího mapování a to lineárního, výsledná kvalita obrazu není příliš uspokojivá. Proces můžeme ovlivnit pouze jediným parametrem a to gamma korekcí, která je standardně volena na hodnotu 2.2, což je hodnota vhodná pro naprostou většinu běžných zobrazovacích zařízení. Na obr. 3.5 vidíme výsledek pro sekvenci okno. Hodnota celkového



Obr. 3.5: Lineární tónování: sekvence okno

dynamického rozsahu se zvětšila, ovšem již na první pohled působí obrazy nebarevným a celkově tmavším dojmem. Je to způsobeno právě lineárním tónováním, kdy

nevyužíváme potenciál HDR obrazu, ale pouze rovnoměrně převedeme všechny bity.

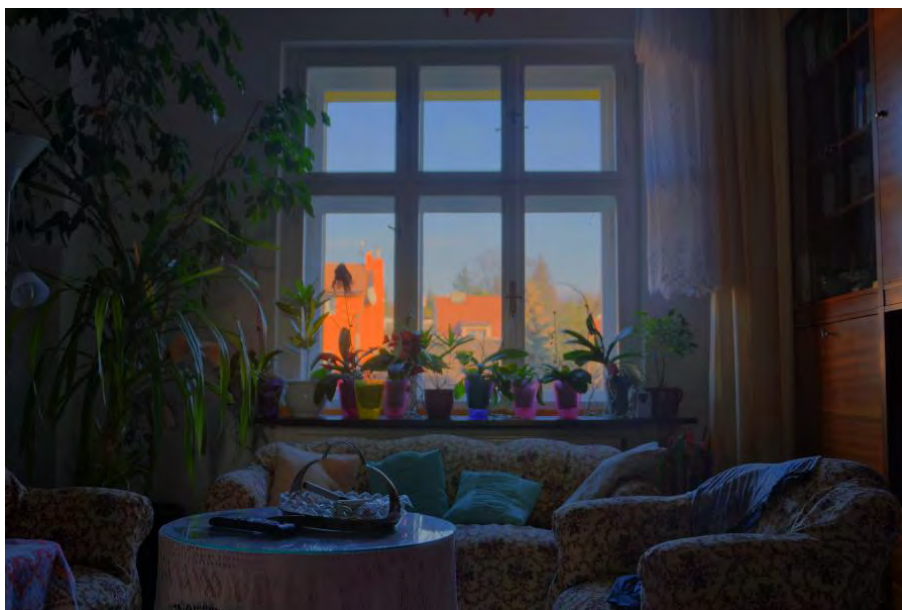
3.6.2 Metoda Reinhard

Principem této globální metody je redukce dynamického rozsahu podobně, jako je tomu u lidského zrakového systému.

Použitá zapouzdřená metoda `openCV TonemapReinhard()` má vstupní parametry γ , a , f' a c . Při prvním načtení jsou nastaveny hodnoty gamma $\gamma = 2,2$, barevná korekce $c = 0$, hodnota poměru pro volbu lokální/globální adaptace $a = 1$ a hodnota pro ovlivnění výsledné intenzity $f' = 0$, které lze následně modifikovat.

Metoda jako první pomocí rovnice 2.32 získá jasovou složku snímku a vypočte hodnoty průmerného, maximálního a minimálního logaritmu jasu. Podle rovnice 2.31 je získán klíč k a následně parametr m dle rovnice 2.30.

Nyní se na každý barevný kanál aplikuje série rovnic 2.36, kde výsledkem je hodnota I_a . Následuje výpočet rovnice 2.28 taktéž pro každý barevný kanál a normalizování hodnot za užití již známého minima a maxima jasu. Na závěr se provede gamma korekce a převod do zobrazitelného rozsahu hodnot. Na obr.3.6 vidíme výsledek metody.



Obr. 3.6: Metoda Reinhard: sekvence okno.

3.6.3 Metoda Drago

Výpočet výsledného jasu pixelu L_d této globální metody adaptivního logaritmického mapování `TonemapDrago()` probíhá na základě rovnice 2.27. Faktor přispůso-

bení L_{dmax} je nastaven na hodnotu $100cd/m^2$, hodnota L_{wmax} je rovna maximální hodnotě jasu v obraze a hodnota L_w je vstupní hodnota jasu pixelu. Hodnota parametru b je volitelná, autoři doporučují jako nejlepší volit hodnotu 0,85. Metoda **TonemapDrago**, jež je pro výpočet použita má 3 vstupní parametry, kromě hodnoty b také gamma korekci (v základu nastavena na hodnotu 2,2) a hodnoty saturace, jež je nastavena na hodnotu 1. Změna na hodnotu větší než 1 saturaci zvětší, menší než 1 naopak sníží. Výsledek metody je vidět na obr. 3.7.



Obr. 3.7: Metoda Drago (adaptivní logaritmické mapování): sekvence okno.

3.6.4 Metoda Durand

Lokální metoda tónového mapování jejímž jádrem je použití bilaterálního filtru. U lokálních metod se váha jednotlivých bodů často počítá na základě Gaussovy funkce, která má ovšem za následek halo efekt. I u této metody je Gaussova funkce použita, avšak ne pro přímý výpočet výsledné hodnoty jasu. Pomocí rovnic 2.39 a 2.40 se vypočítá váhová funkce a podobnostní funkce. Parametr rozptylu těchto funkcí je nastavitelná hodnota metody `TonemapDurand()`. Rozptyl váhové funkce ovlivňuje velikost okolí zahrnuté do výpočtu (větší okolí znamená větší rozmazání) a hodnota rozptylu podobnostní funkce škálu intenzity pixelů, které jsou zahrnuty do výpočtu (větší hodnota znamená větší lokální kontrast). Dalšími vstupními parametry jsou hodnoty gamma korekce, saturace a kontrastu. Největší předností této metody je potlačení halo efektu. Výsledný obraz je na obr. 3.8



Obr. 3.8: Metoda bilaterálního filtrování (Durand): sekvence okno.

3.7 Metoda Exposure Fusion

Naprosto odlišná metoda pro vytvoření snímku s vysokým dynamickým rozsahem. Na rozdíl od ostatních metod zde v práci popsaných, nepracuje s převodem do vysokobitového snímku a následném namapování zpět do zobrazitelného rozsahu, ale používá jednoduché výpočty za pomoci saturace, kontrastu a jasů vstupní sekvence. V prvním kroku metoda MergeMertens pomocí rovnice 2.57 vypočítá váhovací Gaussovu funkci pro každý pixel, pro určení největší váhy správně exponovaných pixelů. Dále se aplikací Laplaceova filtru na šedotónové kopie snímků zjistí jeho absolutní odezva, která poslouží jako hodnota kontrastu pro další výpočet. Hodnota saturace je vypočítána jako směrodatná odchylka každého pixelu v každém barevném kanálu. Tyto tři hodnoty jsou následně použity pro výpočet výsledné váhové mapy pro každý pixel dle rovnice 2.59. Tyto váhové mapy jsou použity jako alfa masky v rovnici 2.61, které spojují snímky Gaussovy a Laplaceovy pyramidy. Zpětné složení této pyramidy do výsledného snímku je posledním krokem metody. Celý výpočet je prováděn zvlášť pro každý barevný kanál. Velkou výhodou této metody je její rychlost a to, že není třeba počítat odezvu snímače a není tedy třeba mít k dispozici metadata vstupních snímků. Vstupními měnitelnými parametry metody jsou váhovací exponenty kontrastu, saturace a expozice. S hodnotou přednastavenou na 1 u všech parametrů dosahovali autoři metody nejlepších výsledků. Výsledek metody je na obr. 3.9



Obr. 3.9: Metoda Exposure fusion: sekvence okno.

3.8 Implementace metody gradientní doménové komprese

Nyní se dostáváme k samotné implementaci metody gradientní doménové komprese. Jako u ostatních metod tónového mapování uvedených v předchozí části, i zde budeme pracovat s obrazem HDR vytvořeným pomocí metody Debevec. Vývojový diagram metody je vidět na obr. 3.10, jejíž jednotlivé kroky budou postupně detailně vysvětleny.

Výpočet logaritmické jasové mapy $H(x, y)$

První operace se vstupním HDR obrazem je převod na logaritmus jasových hodnoty L dle rovnice

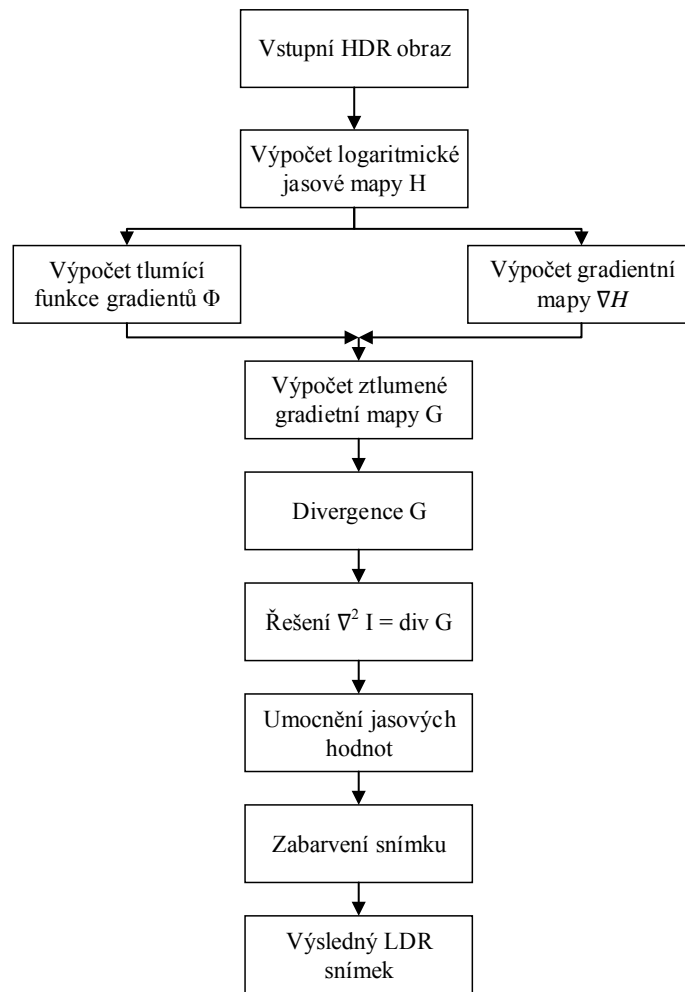
$$L = \log(0,299 \times R + 0,587 \times G + 0,114 \times B), \quad (3.1)$$

kde RGB jsou jednotlivé barevné kanály. Jasová mapa je následně použita v dalších krocích metody pro výpočet tlumící funkce gradientu $\Phi(x, y)$ a pro výpočet gradientní mapy $\nabla H(x, y)$.

Výpočet gradientní mapy $\nabla H(x, y)$

Pro výpočet ztlumené gradientní mapy $G(x, y)$, je nutné nejprve provést výpočet gradientní mapy $\nabla H(x, y)$, jejíž výpočet můžeme aproximovat pomocí rovnice

$$\nabla H(x, y) \approx (H(x + 1, y) - H(x, y), H(x, y + 1) - H(x, y)). \quad (3.2)$$

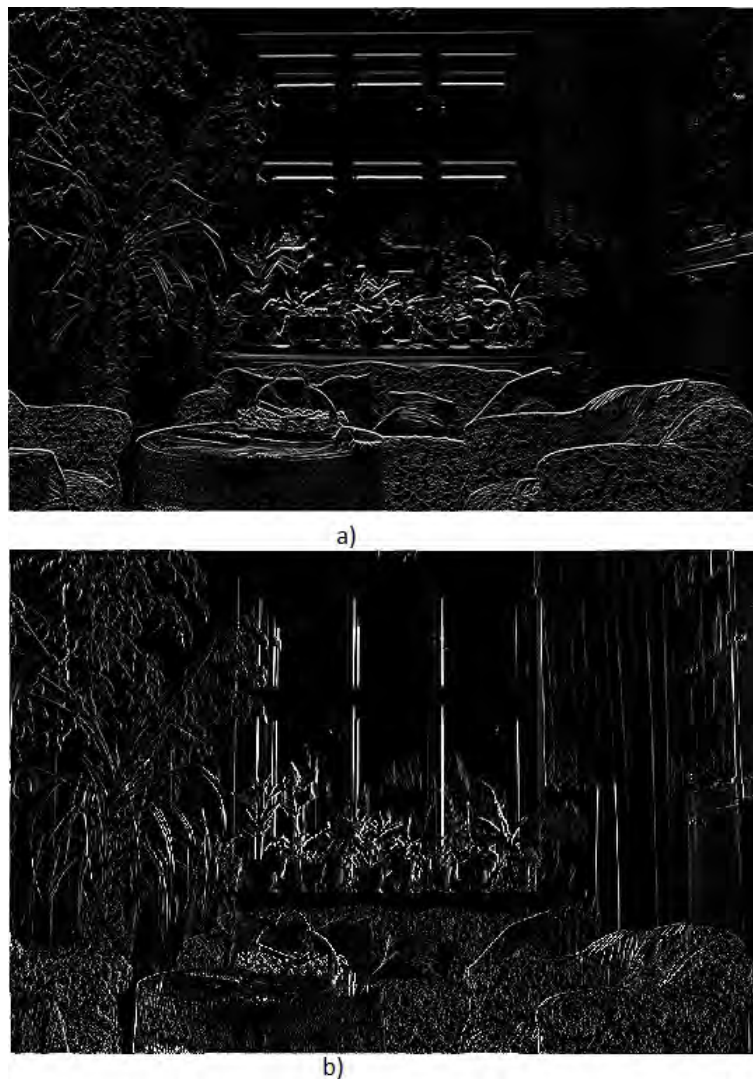


Obr. 3.10: Vývojový diagram metody gradientní doménové komprese, inspirováno v [27].

Na obr.3.11 můžeme vidět vypočítané hodnoty gradientní mapy $\nabla H(x, y)$ v obou osách. Z rovnice 3.2 vyplývá, že je nutné počítat s gradientní mapou pro každou osu odděleně.

Výpočet tlumící funkce gradientu $\Phi(x, y)$

Na zjednodušeném schématu 3.12 vidíme, jak probíhá výpočet tlumící funkce gradientu $\Phi(x, y)$. V první fázi je potřeba zkonstruovat Gaussovu pyramidu. Ze snímku s plným rozlišením postupně vytvoříme pyramidu snímků, kde každá úroveň bude mít o polovinu menší rozlišení, než se dosáhne spodní hranice, kdy nejmenší obraz musí mít alespoň 32 pixelů v obou osách. Gaussova pyramida jejíž úrovně jsou H_0, H_1, \dots, H_d je znázorněna na obr 3.13, kde H_0 je snímek s nejvyšším rozlišením,



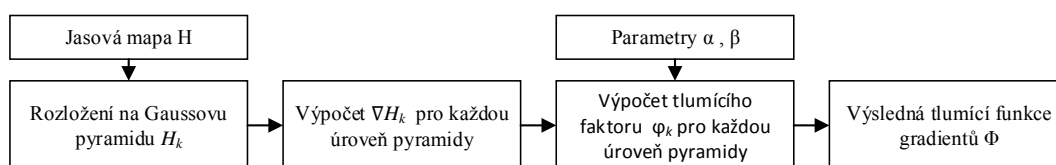
Obr. 3.11: Vypočítané hodnoty gradientní mapy v ose y (a) a x (b).

H_d je snímek s nejmenším rozlišením a H_k snímek k -té úrovně. Pro výpočet obrazu nižší úrovně H_{k+1} je použita zapouzdřená metoda opencv `pyrDown()`. V prvním kroku se nad obrazem provede výpočet následující rovnice s Gaussovým konvolučním jádrem:

$$H_{k+1} = H_k * \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}. \quad (3.3)$$

V druhém kroku se obraz podvzorkuje na nižší úroveň vynecháním sudých řádků a sloupců. [28]

Nyní, když známe počet obrazů d , se začne s výpočtem gradientních map pro



Obr. 3.12: Zjednodušený diagram výpočtu tlumící funkce

každou úroveň pyramidy pomocí centrálních rozdílů dle vzorce 2.52.

Na základě těchto hodnot je třeba vypočítat pro každou úroveň tlumící faktor $\varphi(x, y)$ podle rovnice 2.53. Do tohoto výpočtu vstupují dva parametry α a β , které ovlivňují výslednou hodnotu tlumícího faktoru. Volitelný parametr α bude nastaven na začátku výpočtu na hodnotu desetiny průměrné hodnoty gradientu pro každou úroveň a parametr β , který určuje míru utlumení, bude nastaven na hodnotu 0,85 a je také volitelný.

Výpočet normy obecně pro vektor $\mathbf{x} = (x_1, x_2, \dots, x_n)$ je definován jako

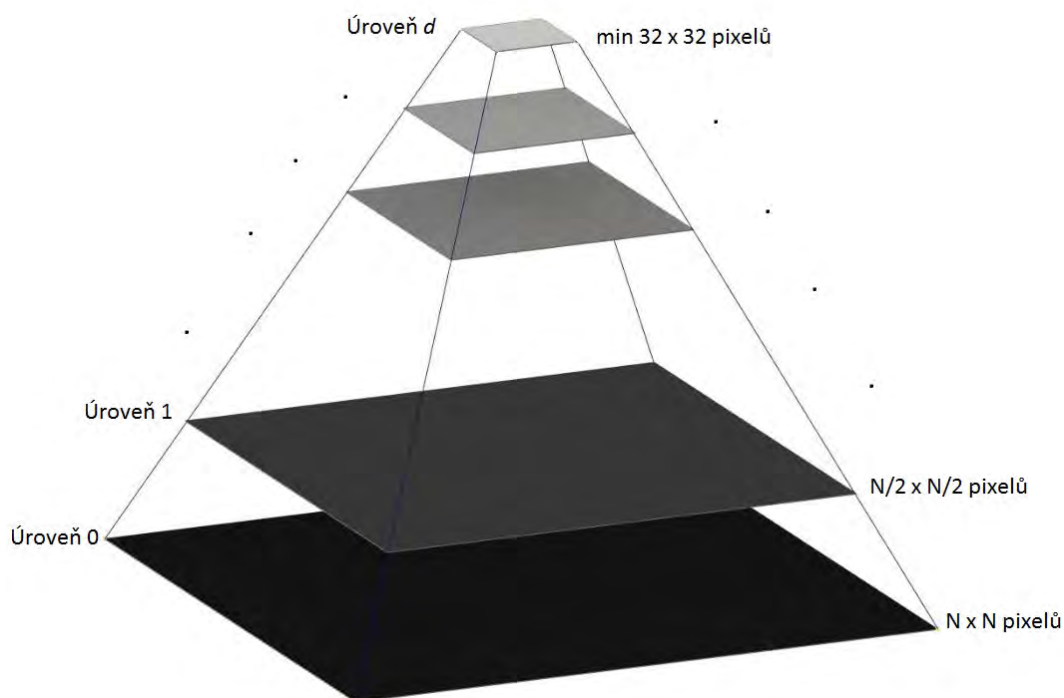
$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}. \quad (3.4)$$

Na základě vzorce 3.4 se vypočítá norma gradientu pro každou úroveň pyramidy. Dalším krokem je finální výpočet výsledné tlumící funkce gradientu $\Phi(x, y)$. Postupuje se dle vzorců 2.54, 2.55 a 2.56. Nejprve se vezme snímek tlumícího faktoru $\varphi(x, y)$ nejnižší úrovně pyramidy, který se pomocí lineární interpolace zvětší na velikost obrazu vyšší úrovně pyramidy. Zde se provede znásobení tlumící funkce gradientu zvětšeného obrazu a aktuálního dané úrovně. Tento proces se opakuje než se dosáhne nulté úrovně s původním plným rozlišením, kdy získáme výslednou hodnotu tlumící funkce gradientu $\Phi(x, y)$. Na obr. 3.14 vidíme příklad vypočítané tlumící funkce. Na tmavších místech dochází k většímu útlumu. Pro výpočet interpolace je použita metoda `resize()`, u níž je pomocí parametrů nastaveno zvětšení na vyšší úroveň pyramidy a jako typ interpolace je zvolena lineární.

Výpočet divergence $\text{div}G$

Výpočet divergence gradientního pole G , značené $\text{div}G$, je definován rovnicí 2.51. Podobně jako u $H(x, y)$ v rovnici 3.2, kde jsme aproximovali dle dopředné diference, nyní můžeme pomocí zpětné diference aproximovat $\text{div}G$ podle

$$\text{div}G \approx G_x(x, y) - G_x(x - 1, y) + G_y(x, y) - G_y(x, y - 1). \quad (3.5)$$



Obr. 3.13: Gaussova pyramida, kde H_0 je obraz s nejvyšším rozlišením a H_d s nejnižším

Poissonova rovnice

Dalším krokem výpočtu je řešení Poissonovy rovnice $\nabla^2 I = \text{div}G$. Jedná se o výpočetně nejnáročnější část metody. Na levé straně rovnice máme Laplaceův operátor a na straně pravé máme funkci o více proměnných zapsanou jako divergenci gradientu. Laplaceův operátor i $\text{div}G$ jsou lineární operátory a proto pomocí jejich aproximace získáme systém lineární rovnic. Kombinace dopředné a zpětné diference u výpočtů $H(x, y)$ a $\text{div}G$ zajišťuje, že aproximace $\nabla^2 I(x, y)$ je konzistentní pomocí centrálních diferencí dle

$$\nabla^2 I(x, y) = I(x + 1, y) + I(x - 1, y) + I(x, y + 1) + I(x, y - 1) - 4I(x, y). \quad (3.6)$$

Na hranicích obrazu se použije stejná rovnice, ale předpokládá se, že derivace za hranicí obrazu jsou rovny 0. Například pro pixely levého sloupce bude platit $I(-1, y) - I(0, y) = 0$.

Nyní přistoupíme k samotnému řešení Poissonovy rovnice. Výsledkem je obraz, jehož Laplaceův operátor se rovná divergenci gradientu G . Stejně jako autoři metody v [26] i zde byla použita pro řešení Poissonovy rovnice Gauss-Seidlovu iterační metoda.



Obr. 3.14: Tlumicí funkce gradientu $\Phi(x, y)$

Tato iterační metoda tvoří soustavu lineárních rovnic $A\mathbf{x} = \mathbf{b}$:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\
 a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\
 \vdots & \\
 a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n,
 \end{aligned} \tag{3.7}$$

kde

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}, \tag{3.8}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{3.9}$$

a

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \tag{3.10}$$

kde hodnoty matice A jsou koeficienty dané dle Laplaceova operátoru, hodnoty matice \mathbf{x} jsou hledané hodnoty aproximace pro výstupní obraz a hodnoty matice \mathbf{b} jsou body vstupní obrazu $\text{div}G$. [31]

Řešení iterační rovnice nevede k přesnému řešení, ale pouze přibližnému. Jejich hlavní výhodou je mnohem menší nárok na paměť počítače, oproti metodám přímým. Na začátku iterační metody se zvolí počáteční aproximace řešení a postupem dané metody se každým krokem zlepši a přiblíží výsledku. Výpočet se provádí určitý počet kroků (předem zadaný), nebo dokud není splněna podmínka určující přesnost řešení. Poté se výpočet ukončí s výsledným přibližným řešením.

Obecně je postup vytvoření iterační metody takový, že se pro každou rovnici soustavy $A\mathbf{x} = \mathbf{b}$ vyjádří neznámá \mathbf{x} a rovnice pak bude vypadat následovně

$$\mathbf{x} = H\mathbf{x} + \mathbf{g}. \quad (3.11)$$

Iterační zápis pak bude

$$\mathbf{x}^{(k+1)} = H\mathbf{x}^{(k)} + \mathbf{g}. \quad (3.12)$$

Na začátku procesu se zvolí počáteční iterace \mathbf{x}^0 a pomocí rovnice 3.12 zjistíme řešení pro $k = 0, 1, 2, \dots, n$. Proces výpočtu je ukončen v momentě, kdy počet iterací dosáhne předem stanovené hodnoty, nebo pokud bude splněna podmínka

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \delta, \quad (3.13)$$

kde δ určuje přesnost řešení \mathbf{x} . Poslední krokem je odhad chyby ε ($k+1$)-ní iterace, pro aproximaci iterace \mathbf{x}^{k+1} dle

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}\| < \varepsilon. \quad (3.14)$$

Výpočet normy přesnosti řešení δ a chyby iterace ε je proveden opět dle 3.4. Než se dostaneme k výpočtu aproximace Gauss-Seidelovy metody, je důležité zjistit, bude-li metoda konvergovat. To se dá dokázat zjištěním, zda je matice soustavy ostře řádkově diagonálně dominantní. Jinak řečeno, matice A musí být pozitivně definitní a to platí právě tehdy, pokud každý nenulový sloupcový vektor $\mathbf{x} = (x_1, \dots, x_n)^T$ platí:

$$\mathbf{x}^T A \mathbf{x} > 0. \quad (3.15)$$

Splnění této podmínky vyplývá ze sestavení matice A popsané v následující části.

Při výpočtu metoda počítá vždy s nejnovějšími hodnotami pro každou další aproximaci řešení (na rozdíl od velmi podobné Jacobiho metody). Obecně výpočet aproximace lze zapsat jako

$$x_n^{(r+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}x_1^{(r+1)} - a_{n2}x_2^{(r+1)} - \dots - a_{nn-1}x_{n-1}^{(r+1)} \right), \quad (3.16)$$

kde r označuje r -tou iteraci. [29] [30]

Matice A musí mít hodnoty sestavené tak, aby pro všechny výsledné pixely ve všech krocích iterace splňovala funkci Laplaceova operátoru dle rovnice 3.6. Sestavení matice A je tedy klíčové. V rovnici

$$A = \begin{bmatrix} D & -I & 0 & 0 & 0 & \dots & 0 \\ -I & D & -I & 0 & 0 & \dots & 0 \\ 0 & -I & D & -I & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -I & D & -I & 0 \\ 0 & \dots & \dots & 0 & -I & D & -I \\ 0 & \dots & \dots & \dots & 0 & -I & D \end{bmatrix}, \quad (3.17)$$

kde I je jednotková matice

$$I = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (3.18)$$

a matice D je dána jako

$$D = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 4 & -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 4 & -1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -1 & 4 & -1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 4 & -1 \\ 0 & \dots & \dots & \dots & 0 & -1 & 4 \end{bmatrix}. \quad (3.19)$$

Velikost matice A , D , i I je závislá na velikosti vstupního obrazu. Pokud je jeho velikost m sloupců a n řádků vstupního $\text{div}G$ obrazu, tak velikost matice A bude $mn \times mn$, a matice D a I $m \times n$. [32] [33]

Z toho popisu vyplývá, že každý řádek (resp. každá lineární rovnice) má pouze 5 nenulových členů (kromě hraničních bodů, kdy hranové pixely mají 4 nenulové členy a rohové pouhé 3). Proto není nutné pro výpočet matici A sestavovat. Každý výsledný pixel se tedy iteruje pouze pomocí váhy 4 okolních pixelů, dle Laplaceova konvolučního jádra, které můžeme vidět na obr. 3.15.

Zpětné zbarvení obrazu

Posledním krokem metody je zpětné zbarvení výstupního rekonstruovaného obrazu, protože metoda pracuje pouze s jasovou složkou vstupního HDR snímku. Přiřazení

0	-1	0
-1	4	-1
0	-1	0

Obr. 3.15: Laplaceovo konvoluční jádro.



Obr. 3.16: Výsledná jasová mapa s přesností $\delta = 0.001$.

barev pixelům výstupního obrazu se sníženým dynamickým rozsahem je provedeno pomocí rovnice

$$C_{\text{out}} = \left(\frac{C_{\text{in}}}{L_{\text{in}}} \right)^s L_{\text{out}} \quad (3.20)$$

kde C_{out} je výstupní hodnota pro jednotlivé barevné kanály RGB, C_{in} je vstupní hodnota jednotlivého barevného kanálu RGB vstupního HDR obrazu, L_{in} a L_{out} je intenzita jasu vstupního a výstupního jednobarevného obrazu. Exponent s ovlivňuje výslednou saturaci a jeho optimální hodnota leží mezi 0,4 a 0,6. Počátečně je nastavena na hodnotu 0,5. Před tímto výpočtem je ještě nutné převést jas vstupního obrazu zpět z logaritmického měřítka pomocí metody `normalize()`. Výsledný zpětně obarvený obraz je na obr. 3.17.



Obr. 3.17: Výsledný zpětně obarvený snímek s přesností $\delta = 0.001$

3.9 Postprocessing

Ekvalizace histogramu

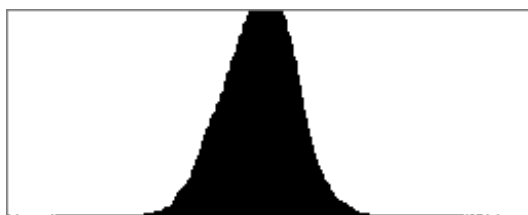
Histogram obrazu znázorňuje četnost jednotlivých úrovní intenzit v rozsahu $[0, L-1]$, kde L je celkový počet úrovní. Histogram je možné zapsat jako diskrétní funkci:

$$h(r_k) = n_k, \quad (3.21)$$

kde r_k je k -tá úroveň intenzity $[0, 1, 2, \dots, L-1]$ a n_k počet pixelů v obraze s intenzitou r_k . Pro lepší informativní charakter se používá pro zobrazení normalizovaný histogram, jenž má jednotlivé složky vydělené celkovým počtem pixelů v obraze n :

$$p(r_k) = \frac{n_k}{n}. \quad (3.22)$$

Při aplikování na obr. 3.16 je z jeho histogramu na obr. 3.18 vidět, že je využita pouze přibližně třetina jasových hodnot a není využita celá škála úrovní (0-255). Proto byla jako další postprocessingová metoda zvolena ekvalizace histogramu.



Obr. 3.18: Histogram snímku 3.16

Transformační (ekvalizační) rovnici lze obecně zapsat jako

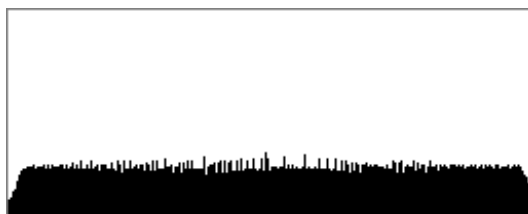
$$s = T(r) \quad (3.23)$$

pro $0 \leq r \leq 255$ pro 8-bitový obraz, kde T značí převodní funkci mezi vstupními úrovněmi jasu r a výslednými úrovněmi s . Výsledná ekvalizační rovnice je definována jako

$$s_k = T(r_k) = \sum_{i=0}^k \frac{n_i}{n}, \quad (3.24)$$

kde r_k je k -tá úroveň jasu, n_k je počet pixelů s danou úrovní jasu r_k a n značí celkový počet pixelů obrazu.

Protože v aplikaci pracujeme s vysokobitovým obrazem HDR ve formátu `float`, při převodu do 8-bitového rozsahu a následné ekvalizaci by došlo ke ztrátě informace. Proto jsou hodnoty nejprve konvertovány pomocí metody `convertTo()` na datový typ `ushort` (16-bitový) s počtem 65536 úrovní a poté ekvalizovány. Posledním krokem je normalizace hodnot do zobrazitelného rozsahu. Na obr. 3.20 je výsledný obraz s provedenou ekvalizací histogramu a na následujícím obr. 3.19 jeho histogram. [3]



Obr. 3.19: Ekvalizovaný histogram snímku 3.16

Zaostření

Výsledná kvalita obrazu namapovaného metodou gradientní doménové komprese, je dle autorů práce v [26] ve většině případů zvýšena provedením zaostřovací operace. Pro zaostření byla zvolena metoda neostrého maskování. [34] První fází procesu je vytvoření kopie výsledného obrazu. Nad touto kopií se provede Gaussovo rozostření pomocí funkce `GaussianBlur()`. Tato kopie se porovná s originálním obrazem pomocí funkce `addWeighted()`, která v případě překročení prahové hodnoty tyto hodnoty od sebe odečte. Vstupními parametry metody jsou:

- prahová hodnota - která určuje minimální změnu jasu, aby byli pixely zaostřeny, neboli jak daleko od sebe musí být jasové hodnoty, aby se zaostření provedlo či nikoliv.
- Oblast - určuje velikost zaostření. Menší hodnota způsobí zaostření spíše menších detailů a naopak.



Obr. 3.20: Výsledný jasový snímek s ekvalizací histogramu.

- Rozsah - procentuální hodnota řídící o kolik tmavší budou tmavé pixely a o kolik světlejší ty světlé.

Tyto parametry jsou ponechány na počátečních hodnotách, které (dle subjektivního posouzení) přináší dobré výsledky. Na obr. 3.21 vidíme zaostřený výsledný obraz. Porovnat můžeme s nezaostřeným obrazem 3.17. [34]



Obr. 3.21: Výsledná jasová mapa s přesností $\delta = 0.001$ a s provedeným zaostřením.

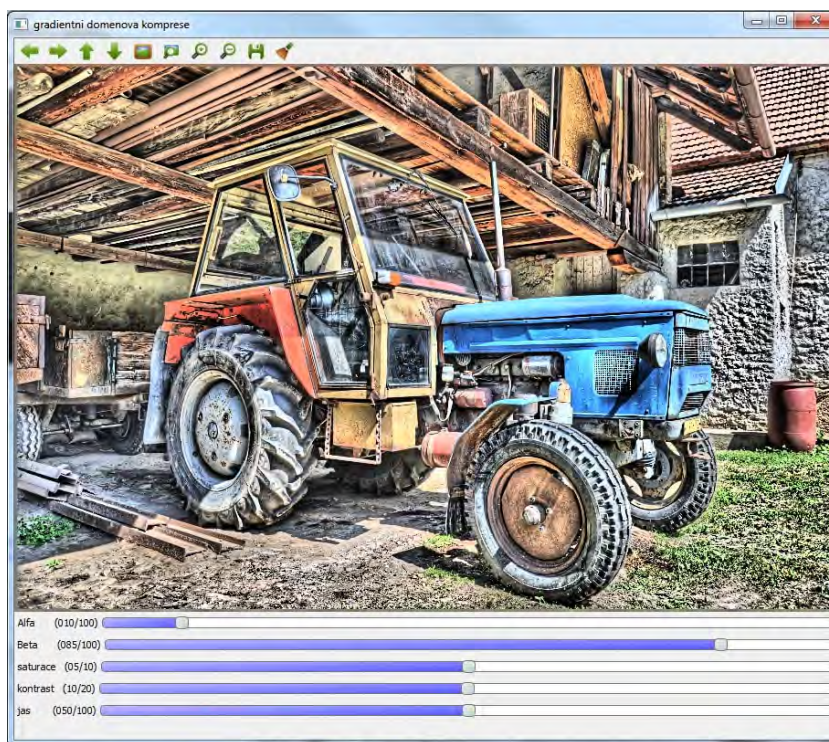
4 APLIKACE A VÝSLEDKY

4.1 Popis aplikace

Výsledná aplikace pro realizaci popsaných algoritmů je konzolová s prvky GUI použité z knihoven funkcí OpenCv a Qt. Po spuštění aplikace dojde k načtení vstupní sekvence, která musí být ve stejné složce jako spouštěcí soubor `HDR_v3.exe` a musí obsahovat soubor `list.txt` se seznamem názvu snímků a jejich inverzními hodnotami času expozice (pro výpočet odezvy snímače metody Debevec). Po automatickém provedení zarovnání vstupní sekvence je na výběr pokračování výpočtem metody Debevec do vysokobitového obrazu, nebo zobrazení metody Mertens (protože jako jediná z metod zde v práci uvedených nepracuje s vysokobitovým HDR obrazem). Po výpočtu odezvy snímače metodou Debevec je na výběr ze 4 metod tónového mapování:

- globální: metoda Drago a Reinhard,
- lokální: metoda Durand,
- gradientní: metoda gradientní doménové komprese - GDR.

Při zvolení kterékoliv metody dojde k spuštění náhledového okna pro danou metodu. Každá metoda má své specifické měnitelné vstupní parametry a ty je možné pomocí trackbarů (táhel) modifikovat, viz obr. 4.1. Vedle ikony pro uložení náhledu



Obr. 4.1: Náhled aplikace okna aplikace.

je ikona pro rozkliknutí okna vlastností, kde si můžeme zavolat jinou metodu tónového mapování pro porovnání a také možnost uložení dané metody v plném rozlišení s aktuálně nastavenými parametry.

4.2 Výsledky metody gradientní doménové komprese

U implementované metody gradientní doménové komprese je nutné před výpočtem zadat vstupní parametry, jak bylo uvedeno v předchozí části. Prvním je hodnota δ , určující přesnost řešení dle rovnice 3.13 a dalším maximální počet iterací, které budou provedeny. Je nutné si uvědomit, že při zadání přesnosti výsledku např. $\delta = 0.001$ a nízké hodnoty maximálního počtu iterací (např. 100), s největší pravděpodobností nebude této přesnosti dosaženo. Proto je vhodné zadat větší počet iterací než očekáváme.

Přesnost řešení δ a parametr β

Výpočetně nejnáročnější část této metody, kterou je výpočet Poissonovy rovnice pomocí iterační Gauss-Seidlovoy metody, také nejvíce ovlivňuje výsledný snímek. Na obr. 4.2 je 8 výsledných snímků. Na prvním řádku jsou snímky s přesností výpočtu iterace $\delta < 0.1$, na druhém řádku $\delta < 0.01$, třetím $\delta < 0.001$ a posledním $\delta < 0.0001$. Druhým parametrem, který nejvíce ovlivňuje výsledek je parametr β , který mají snímky v levé části ponechaný na počáteční hodnotu ($\beta = 0,85$) a pravé snímky na $\beta = 0,90$.

Se stoupající přesností řešení Poissonovy rovnice vzrůstá i kvalita výsledného snímku, avšak z důvodu výpočetní náročnosti dramaticky narůstá čas výpočtu. Stručné porovnání různých iteračních přesností v závislosti na velikosti obrazu se nachází v následující části. Jak bylo zmíněno, parametr β taktéž velmi výrazně ovlivňuje výsledek výpočtu. Z rovnice 2.53 a jejího popisu vyplynulo, že ovlivňuje míru tlumící funkce $\Phi_k(x, y)$ (za předpokladu, že $\beta < 1$). Volba tohoto parametru je dobře viditelná na obrazech traktoru s přesností $\delta < 0.001$ z obr. 4.2 v plném rozlišení na obr. 4.3. Na snímku (a) je parametr β ponechán na základní hodnotu 0,85. Tlumící funkce $\Phi_k(x, y)$ tedy bude mít větší hodnotu a gradientní rozdíly světla budou tlumeny více. Proto jsou zde oproti obr. (b) lépe viditelné místa scény, kde je největší rozdíl světelných intenzit scény (místo pod zadní nápravou traktoru). Naпротив tomu na snímku (b) je hodnota β zvýšena na hodnotu 0,90 a proto i tlumící funkce $\Phi_k(x, y)$ bude menší. Proto místo pod zadní nápravou již není tak dobře viditelné. Obraz je však více vyhlazen a působí přirozeněji, právě z důvodu menšího

tlumení, kdy se netlumí i nejmenší gradientní rozdíly a nedochází tak k (místy až k přílišnému) nárůstu lokálního kontrastu, jak je tomu na snímku (a).

Při nastavení vyššího parametru β se může na větších plochách bez změn (obloha) vyskytnout zvýšený šum a jev zvaný posterizace, kdy i malé gradientní změny jasu jsou ve výsledném obrazu mnohonásobně zvýrazněny. Tento nežádoucí jev lze omezit nastavením parametru β na nižší hodnotu.

Parametr α

Dalším parametrem ovlivňujícím výpočet tlumící funkce je měnitelná hodnota α . Jeho nastavením se určuje hranice, které gradientní změny zůstanou beze změny a které ne. Jeho počáteční hodnota je nastavena na 0,1, která vychází dle dostupných výsledků jako nejvhodnější. Nicméně změna tohoto parametru je možná a její výsledný efekt můžeme vidět na obr. 4.4. Zvýšení ani snížení hodnoty tohoto parametru nezlepšilo výsledek. Efekt změny parametru se zdá být srovnatelný s parametrem β , tzn. jeho zmenšení navýší lokální kontrast a naopak.

Výpočetní náročnost

Jak bylo již dříve zmíněno, výpočetní náročnost metody GDK výrazně stoupá při zvýšení přesnosti jejího řešení (počtu iterací). Aplikace je u této metody schopna okamžitě reagovat na změny při počtu jednotek, maximálně několik desítek iterací pro obraz. Vyšší hodnoty iterací již mají za následek delší čas výpočtu. V tabulce 4.1 můžeme porovnat jednotlivé výpočetní časy iterace (s defaultně nastavenými vstupními parametry) pro procesor Intel® Core™ i5-480M 2.66 GHz. Pro viditelnou změnu parametrů je vhodné volit nízký počet iterací (nebo přesnosti). Z tabulky je také dále patrné, že při vyšším rozlišení je pro požadovanou přesnost potřeba provedení většího počtu iterací. Na obr. 4.2 můžeme porovnat rozdíl viditelný rozdíl sekvence Traktor - náhled, od nejmenšího počtu iterací po největší (levé snímky s počátečním nastavením a parametrů metody).

Parametry α i β také ovlivňují výsledný počet iterací, viz tabulka 4.2. Zvýšení hodnoty parametru α má za následek zvětšení oblasti gradientních změn, který má za následek zvýšení počtu iterací pro přiblížení se správnosti řešení Poissonovy rovnice. Tím pádem narůstá výpočetní čas. Nižší hodnota naopak zmenší oblast výpočtu a čas se snižuje. Podobně je tomu u parametru β , jehož nižší hodnota způsobí nižší tlumení gradientů a tím rychlejší výpočet a naopak.

Tab. 4.1: Výpočetní čas metody GDK pro různé obrazové rozlišení a počty iterací

Sekvence	Rozlišení	Přesnost řešení δ	Počet iterací	Výpočetní čas [s]
Okno - náhled	900 × 600	<0,1	9	0,094
		<0,01	79	0,749
		<0,001	949	9,428
		<0,001	5000	51,329
Okno	2464 × 1632	<0,1	14	0,872
		<0,01	124	7,489
		<0,001	1405	84,233
Traktor - náhled	900 × 600	<0,1	8	0,093
		<0,01	63	0,667
		<0,001	650	6,855
		<0,0001	4756	48,586
Traktor	2464 × 1632	<0,1	15	0,94
		<0,01	103	6,199
		<0,001	1149	68,347
		<0,0001	10933	676,832

Tab. 4.2: Výpočetní čas metody GDK pro různé nastavení α a β

Sekvence	α [-]	β [-]	Přesnost řešení δ	Počet iterací	Výpočetní čas [s]
Traktor - náhled (900 × 600)	0,1	0,8	<0,001	313	3,379
	0,1	0,95	<0,001	2514	26,937
	0,01	0,85	<0,001	114	1,125
	1	0,85	<0,001	3235	32,828

4.3 Srovnání jednotlivých metod

Výsledek všech popsanych a implementovaných metod tónového mapování sekvence traktor s počátečním nastavením parametrů můžeme vidět na obr. 4.5. Počáteční nastavení vstupních parametrů jednotlivých metod je uveden v následujícím seznamu.

- Drago: gamma korekce: 2,2, saturace: 1,0, bias funkce: 0,85.
- Reinhard: gamma korekce: 1,0, intenzita: 0,0, světelná adaptace (globální/lokální): 1,0, barevná adaptace: 0,0.
- Durand: gamma korekce: 2,2, kontrast: 4,0, saturace: 1,0, sigma space (podob-

nostní funkce): 1,0, sigma color (váhovací funkce): 1,0.

- Exposure fusion: kontrast 1,0, saturace 1,0, expozice 1,0.
- Gradientní doménová komprese: α : 0,1, β : 0,85, saturace: 0,5.

Srovnání poskytne následující obr. 4.6, na kterém jsou u jednotlivých metod pozměněny některé parametry. U metody Drago na snímku (a) je oproti počátečním hodnotám zvednutá hodnota bias, která mírně zlepšila jeho světelné podmínky. U metody Reinhard (b), byla zvýšena hodnota světelné intenzity a snížena hodnota světelné adaptace, došlo tak k lepšímu prokreslení celé scény. U metody Durand (c) byla zvýšena hodnota rozptylu váhovací funkce (sigma space), která mírně navýšila lokální kontrast. U metody Mertens (d) se jakákoliv změna volitelných parametrů odrazí ve snížení kvality výsledku, viz pozměněný obraz. Na snímku (e) je výsledek metody gradientní doménové komprese. Protože pro počáteční hodnotu β má výsledek příliš vysoký lokální kontrast, jeho zvýšením docílíme přirozenějšího vzhledu. Další porovnání se naskýtá na snímcích sekvence dřevník s počátečními parametry na obr. 4.7 a modifikovanými na obr. 4.8. Největší změnu můžeme pozorovat na snímku (e), kdy při počátečních podmínkách je v levé horní části snímku na obrazu viditelná posterizace a šum, z důvodu příliš velkého lokálního kontrastu. Při zvýšení parametru α a β je tento artefakt odstraněn, avšak v důsledku příliš velkého okolí výpočtu gradientů a tlumících funkcí je kolem střechy budovy patrná aura kolem hran - halo efekt.

4.4 Vstupní sekvence

Nasnímané vstupní sekvence byly voleny tak, aby scéna obsahovala co největší dynamický rozsah intenzity světla spolu s co největším počtem detailů. Všechny sekvence jsou nasnímány s hodnotou ISO 100 a clonovým číslem $f/5$. Voleným parametrem pro jinou hodnotu EV scény byl expoziční čas. Základní informace o sekvencích můžeme vidět v tabulce 4.3. Výsledky metody gradientní doménové komprese všech

Tab. 4.3: Přehled vstupních sekvencí.

Sekvence	Rozlišení	Počet snímků	Expoziční čas snímků [s]
Okno	2464×1632	15	$4 - \frac{1}{4000}$
Traktor	2464×1632	11	$2 - \frac{1}{1600}$
Dřevník	2464×1632	8	$\frac{1}{10} - \frac{1}{1250}$
Motor	3696×2448	9	$\frac{1}{10} - \frac{1}{2500}$

sekvencích jsou na obr. 4.9 a 4.10.



Obr. 4.2: První řádek snímků je s přesností výpočtu iterace $\delta < 0.1$, druhý $\delta < 0.01$, třetí $\delta < 0.001$ a čtvrtý $\delta < 0.0001$. Levé snímky mají parametr $\beta = 0,85$, pravé $\beta = 0,90$.



a)



b)

Obr. 4.3: Výsledek s parametrem $\beta = 0.85$ (a), výsledek s parametrem $\beta = 0.90$ (b).
Oba snímky jsou s přesností $\delta < 0.001$ a hodnotou $\alpha = 0,1$



a)



b)

Obr. 4.4: Výsledek s parametrem $\alpha = 0,01$ (a) a $\alpha = 1$ (b). Oba snímky jsou s přesností $\delta < 0.001$ a hodnotou $\beta = 0,85$



a)



b)



c)



d)



e)

Obr. 4.5: Srovnání jednotlivých metod s počátečním nastavením vstupních parametrů: Drago (a), Reinhard (b), Durand (c), Mertens (d), Gradientní doménová komprese (e) (s přesností $\delta < 0,001$). (sekvence traktor)



a)



b)



c)



d)



e)

Obr. 4.6: Srovnání jednotlivých metod s modifikovanými vstupními parametry: Drago (a), Reinhard (b), Durand (c), Mertens (d), Gradientní doménová komprese (e). (sekvence traktor)



a)



b)



c)



d)



e)

Obr. 4.7: Srovnání jednotlivých metod s počátečním nastavením vstupních parametrů: Drago (a), Reinhard (b), Durand (c), Mertens (d), Gradientní doménová komprese (e). (sekvence dřevníků)



a)



b)



c)



d)



e)

Obr. 4.8: Srovnání jednotlivých metod s modifikovanými vstupními parametry vůči počátečnímu: Drago - snížená hodnota gamma a bias (a), Reinhard - zvýšená hodnota intenzity a světelné adaptace (b), Durand - zvýšená hodnota kontrastu a snížená hodnota sigma color (c), Mertens - zvýšená hodnota expozice (d), Gradientní doménová komprese - zvýšená hodnota α a β (e) . (sekvence dřevníků)



a)



b)

Obr. 4.9: Výsedky gradientní doménové komprese, sekvence okno s počátečními parametry (a), sekvence traktor se změnou hodnotu $\beta = 0,92$ (b)



a)



b)

Obr. 4.10: Výsedky gradientní doménové komprese, sekvence dřevník se změnou hodnotu $\beta = 0,90$ (a), sekvence motor se změnou hodnotu $\beta = 0,95$ (b)

5 ZÁVĚR

Cílem této práce bylo nastudování a popsání matematického aparátu pro skládání snímků s různou expoziční hodnotou do jediného snímku s vysokým dynamickým rozsahem a dále tyto algoritmy a metody implementovat.

V první části práce jsme se seznámili s teoretickým základem, nutným pro pochopení použitých metod a algoritmů. V druhé kapitole jsou popsány matematické aparáty metod a algoritmů použitých v realizované aplikaci. Největší a také nejkomplikovanější část práce s obrazem HDR představuje tónové mapování - převod z vysokobitového obrazu zpět na zobrazitelný rozsah. Proto je pro tuto část procesu představeno více metod.

Pro zarovnání obrazů vstupních sekvencí je vysvětlena a demonstrována metoda median threshold bitmap, která byla vyvinuta právě pro proces HDR. Výpočet vysokobitového obrazu je proveden pomocí metody Debevec. Pro proces zpětného převodu do zobrazitelného rozsahu HDR obrazu je použito několik metod tónového mapování implementovaných jako zapouzdřené funkce v knihovně OpenCV: dvě globální (Drago a Reinhard), jedna lokální (Durand). Dále byla v rámci této diplomové práce implementována metoda Gradientní doménové komprese. Pro porovnání byla také představena metoda Exposure fusion, která nepracuje s převodem vstupní sekvence do vysokobitového obrazu, ale převod provádí za pomoci jednoduchých výpočtů na základě hodnot kontrastu, saturace a expozice. V třetí kapitole je popsána návrh a implementace jednotlivých metod, detailně je rozebrána metoda Gradientní doménové komprese. Poslední kapitola popisuje samotnou realizovanou aplikaci a výsledky jednotlivých metod při různém nastavení jejich vstupních parametrů.

Nejlepších výsledků dosahuje metoda gradientní. Oproti metodám globálním u ní dochází k lepšímu prokreslení celé scény, především detailů. Její nevýhodou je však výpočetní náročnost, která je způsobena nutností řešení parciální diferenciální Poissonovy rovnice.

Hlavním přínosem práce je detailní popis složitého procesu vytváření obrazu s vysokým dynamickým rozsahem, především procesu tónového mapování s detailně popsanou a kompletně implementovanou metodou Gradientní doménové komprese. Pokračováním práce by mohlo být vyřešení rychlejší konvergence Poissonovy rovnice u této metody a také vylepšení grafického rozhraní vytvořené aplikace.

LITERATURA

- [1] Jandora, R. *Maturitní otázky do fyziky* [online]. 2000, [cit. 7. 10. 2015]. Dostupné z URL: <<http://radek.jandora.sweb.cz/fyzika.htm>>.
- [2] Gonzales, R. Woods, R., 3. vydání. *Digital Image Procesing*. 3. vyd. New Jersey: Pearson Education, 2008. ISBN 978-0-13-168728-8.
- [3] Říha, K., *Pokročilé techniky zpracování obrazu*. Vysoké učení technické v Brně, 2007
- [4] Peli, E., *Contrast in complex images* [online]. 2000, [cit. 8. 10. 2015]. Dostupné z URL: <<http://serinet.meei.harvard.edu/faculty/peli/papers/ContrastJOSA.pdf>>.
- [5] Rajmic, K., *Základy počítačové sazby a grafiky*. Vysoké učení technické v Brně, 2012
- [6] Burian, P.K, Caputo, R., *National Geographic: Škola fotografování - krajina a příroda*. 2008. ISBN 80-7026-253-2.
- [7] Jonathan *Beginners guide to your camera : The Aperture* [online]. 2009, [cit. 10. 10. 2015]. Dostupné z URL: <<http://mediaformations.com/beginners-guide-to-your-camera-the-aperture/>>.
- [8] Jonathan *Beginners guide to your camera : The Aperture* [online]. 2009, [cit. 10. 10. 2015]. Dostupné z URL: <<http://mediaformations.com/beginners-guide-to-your-camera-the-shutter//>>.
- [9] Pihan, R., *Dynamický rozsah a kontrast scény* [online]. 2012, [cit. 15. 10. 2015]. Dostupné z URL: <http://www.fotoroman.cz/techniques3/expozice4_rozsah.htm>.
- [10] Fučík, K., *Dynamický rozsah digitálních fotoaparátů* [online]. 2008, [cit. 15. 10. 2015]. Dostupné z URL: <<http://www.karf.cz/dynamicky-rozsah-digitalnich-fotoaparatu/>>.
- [11] Johnson, T., McGee, S., Ortman, R., Yang, T., *Exploring High Dynamic Range Imaging* [online]. 2006, [cit. 17. 10. 2015]. Dostupné z URL: <<http://cnx.org/contents/08f77b6f-2a7f-4851-bf18-d886d041a26302/Exploring-High-Dynamic-Range-I>>.
- [12] Meylan, L., *tone mapping for high dynamic range images* [online]. 2006, [cit. 22. 10. 2015]. Dostupné z URL: <http://infoscience.epfl.ch/record/86005/files/EPFL_TH3588.pdf>.

- [13] Burda, K., Stražil, I., *Zabezpečovací systémy*. Vysoké učení technické v Brně, 2011
- [14] Mansurov, N., *RAW vs JPEG* [online]. 2010, January 26, [cit. 5. 11. 2015]. Dostupné z URL: <<https://photographylife.com/raw-vs-jpeg>>.
- [15] Číka, P., *Multimediální služby*. Vysoké učení technické v Brně, 2012
- [16] Čadík, M., Wimmer, M., Neumann, L., Artusi, A., *Evaluation of HDR tone mapping methods using essential perceptual attributes* [online]. 2007, June 18, [cit. 8. 11. 2015]. Dostupné z URL: <<http://www.sciencedirect.com/science/article/pii/S0097849308000460>>.
- [17] Debevec, P.E., Malik, J., *Recovering High Dynamic Range Radiance Maps from Photographs* [online]. 1997, [cit. 16. 11. 2015]. Dostupné z URL: <<http://www.pauldebevec.com/Research/HDR/debevec-siggraph97.pdf>>.
- [18] Ward, G., *Fast, Robust Image Registration for Compositing High Dynamic Range Photographs from Handheld Exposures* [online]. 2003, [cit. 26. 11. 2015]. Dostupné z URL: <<http://pages.cs.wisc.edu/~lizhang/courses/cs766-2008f/projects/hdr/jgtpap2.pdf>>.
- [19] Morozov, F., *High Dynamic Range Imaging* [online]. 2015, [cit. 26. 11. 2015]. Dostupné z URL: <http://docs.opencv.org/master/d3/db7/tutorial_hdr_imaging.html>.
- [20] Drago, F., Myszkowski, K., Annen, T., Chiba, N., *Adaptive Logarithmic Mapping For Displaying High Contrast Scenes* [online]. 2003, [cit. 10. 12. 2015]. Dostupné z URL: <<http://resources.mpi-inf.mpg.de/tmo/logmap/logmap.pdf>>.
- [21] Mertens, T., Kautz, J., Reeth, F., V., *Exposure Fusion* [online]. 2005, [cit. 10. 12. 2015]. Dostupné z URL: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4392748>>.
- [22] Reinhard, E., Devlin, K., *Dynamic Range Reduction Inspired by Photoreceptor Physiology* [online]. 2005, January/February [cit. 3. 3. 2016]. Dostupné z URL: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1359728>>.
- [23] Freeman, M., *Fotografie a vysoký dynamický rozsah* [online]. 2008, [cit. 26. 3. 2016]. Dostupné z URL: <<http://www.beletrie.eu/data/attachments/Fotografie%20a%20vysok%C3%BD%20dynamick%C3%BD%20rozsah.pdf>>. 1.vydání, ISBN 978-80-86815-95-4

- [24] Durand, F., Dorsey, J., *Fast Bilateral Filtering for the Display of High-Dynamic-Range Images* [online]. July 2002 , Massachusetts Institute of Technology, [cit. 3. 3. 2016]. Dostupné z URL: <<https://people.csail.mit.edu/fredo/PUBLI/Siggraph2002/DurandBilateral.pdf>>.
- [25] Tomasi, C., Manduchi, R., *Bilateral Filtering for Gray and Color Images* [online]. 1998 , IEEE International Conference on Computer Vision, Bombay, India, [cit. 3. 3. 2016]. Dostupné z URL: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.4.634&rep=rep1&type=pdf>>.
- [26] Fattal, R., Lischinski, D., Werman, M., *Gradient Domain High Dynamic Range Compression* [online]. 2001 , [cit. 15. 3. 2016]. Dostupné z URL: <<http://www.cs.huji.ac.il/~danix/hdr/hdrc.pdf>>.
- [27] Fattal, R., Lischinski, D., Werman, M., *Gradient Domain High Dynamic Range Compression - Final Report* [online]. 2011.6.12, [cit. 18. 3. 2016]. Dostupné z URL: <http://web.xidian.edu.cn/zhengzk/files/20121122_154734.pdf>.
- [28] Huamán, A., *Image Pyramids* [online]. 2016, [cit. 10. 4. 2016]. Dostupné z URL: <<http://docs.opencv.org/2.4/doc/tutorials/imgproc/pyramids/pyramids.html#pyramids>>.
- [29] Mošová, V., Hlavička, R., *Numerické metody* [online]. Vysoké učení technické v Brně, [cit. 10. 4. 2016]. Dostupné z URL: <<http://physics.ujep.cz/~jskvor/NME/DalsiSkripta/Numerika.pdf>>.
- [30] Mošová, V., *Numerické metody* [online]. Vysoké učení technické v Brně, [cit. 11. 4. 2016]. Dostupné z URL: <<http://physics.ujep.cz/~jskvor/NME/DalsiSkripta/numerickemetody.pdf>>.
- [31] Wikipedia, *Gauss–Seidel method* [online]. Wikipedia, [cit. 11. 4. 2016]. Dostupné z URL: <https://en.wikipedia.org/wiki/Gauss%E2%80%93Seidel_method>.
- [32] Wikipedia, *Discrete Poisson equation* [online]. Wikipedia, [cit. 11. 4. 2016]. Dostupné z URL: <https://en.wikipedia.org/wiki/Discrete_Poisson_equation>.
- [33] Demmel, J., *Solving the Discrete Poisson Equation using Jacobi, SOR, Conjugate Gradients, and the FFT* [online]. [cit. 11. 4. 2016]. Dostupné z URL: <<http://www.cs.berkeley.edu/~demmel/cs267/lecture24/lecture24.html>>.

- [34] Fulton, W., *Sharpening - Unsharp Mask* [online]. Wikipedia, [cit. 11. 5. 2016].
Dostupné z URL: <<http://www.scantips.com/simple6.html>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

HDR	High dynamic range - vysoký dynamický rozsah
LDR	Low dynamic range - nízký dynamický rozsah
EV	Exposition value - expoziční hodnota
MTB	Median Threshold Bitmap
CMOS	Complementary Metal Oxide Semiconductor

SEZNAM PŘÍLOH

A Přílohy	83
A.1 Použité vstupní sekvence	83
B Obsah přiloženého CD	85

A PŘÍLOHY

A.1 Použité vstupní sekvence



Obr. A.1: Vstupní sekvence snímků - okno



Obr. A.2: Vstupní sekvence snímků - traktor



Obr. A.3: Vstupní sekvence snímků - dřevník



Obr. A.4: Vstupní sekvence snímků - motor

B OBSAH PŘILOŽENÉHO CD

- Elektronická verze práce
- Zdrojové kódy k aplikaci, projekt pro MS Visual studio 2012
- Vstupní sekvence snímků
- Dynamické knihovny OpenCV, Libraw a Qt příslušné verze, jež jsou nutné pro správné spuštění programu (ve složce Release).
- Instalační soubor knihovny Qt 5.2.1
- Soubor `readme.txt` s návodem pro spuštění aplikace.