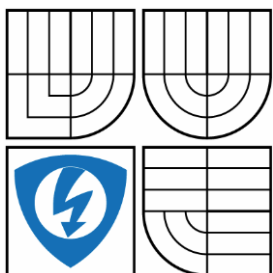


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

KOREKCE SNÍMKŮ

IMAGE CORRECTION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR SEDLO

VEDOUČÍ PRÁCE

SUPERVISOR

Ing. MILOSLAV RICHTER, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Petr Sedlo

ID: 119599

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Korekce snímků

POKYNY PRO VYPRACOVÁNÍ:

Napište původní aplikaci, která umožní stanovit a korigovat vady vzniklé při pořízení digitálního snímku.

1) Stanovte možnosti korekce digitálního snímku pro následující případy:

a) je k dispozici pouze jediný snímek scény

b) je k dispozici snímek a možnost pořídit jiným snímacím zařízením snímky stejné scény

c) je k dispozici zařízení, kterým byl snímek pořízen

d) je k dispozici zařízení, kterým byl snímek pořízen i možnost pořídit snímky stejné scény.

2) Definujte typy zkreslení a možnost jejich identifikace a korekce (uvažujte nejčastější jasová a geometrická zkreslení). Nasnímejte databázi snímků pro testování jednotlivých typů zkreslení.

3) Navrhněte a realizujte algoritmy pro zjištění vad a pro korekci snímku.

4) Zhodnoťte vlastní zvolené řešení a dosažené výsledky pro varianty uvedené v bodu 1.

DOPORUČENÁ LITERATURA:

Hlaváč V., Šonka M.: Počítačové vidění, Grada, Praha 1992, ISBN 80-85424-67-3

Termín zadání: 8.2.2016

Termín odevzdání: 16.5.2016

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

Konzultant diplomové práce:

doc. Ing. Václav Jirsík, 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

Ve své diplomové práci jsem se zabýval různými nežádoucími vlivy, které poškozují digitálně pořízený snímek. Snímek pořízený digitálním fotoaparátem může trpět řadou vad. V první řadě je to šum, který vzniká v důsledku diskretizace, dále jsou to vady vinětace, radiální zkreslení v obraze, chromatická aberace. Také jsem se ve své práci zabýval kompenzací obrazů s podsvícením a tvorbou panoramat ze sekvence snímků. U obrazů s podsvícením je třeba zvýšit jas v tmavých částech obrazu, aby zde nabyly výraznější kresby. Při tvorbě panoramat musíme dbát na vyváženost jasu v dílčích snímcích, která nebývá zachována a následně najít vhodné rozhraní ke spojení snímků a to zhladit. Stěžejní fází bylo napsání algoritmů, které tyto vady dovedou alespoň částečně kompenzovat. K tomuto účelu jsem využíval prostředí Matlab.

Klíčová slova

korekce snímků, obraz s podsvícením, tvorba panoramatu, vinětace, radiální zkreslení, chromatická aberace

Abstract

This work is focussed on different unwanted effects that damage a digitally acquired image. An image acquired using a digital camera can suffer from a series of defects. First, the noise is concerned. It occurs due to discretisation. Next, there are the defects including vignetting, radial image distortion, chromatic aberration. In my thesis I also worked on backlight image compensation and panorama creation from an image sequence. For backlight images we need to increase the brightness in the dark parts of the image, so that the details became more visible. At panorama creation we are careful about brightness evenness in partial images, which is usually not conserved and consequently we try to find a suitable boundary-line to connect images and smooth it. The fundamental part of my work was writing the algorithms that are able to compensate these defects at least partly. For this aim I took use of the Matlab environment.

Keywords

Image correction, vignetting, backlight image compensation, panorama stitching, radial distortion, chromatic aberration

Bibliografická citace:

SEDLO, P. *Korekce snímků*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 87s. Vedoucí diplomové práce byl Ing. Miloslav Richter, Phd.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Korekce snímků jsem vypracoval 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 uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: 27. dubna 2016

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Miloslavovi Richterovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: 27. dubna 2016

.....
podpis autora

Obsah

1	Úvod.....	9
2	Filtrace šumu.....	10
2.1	Filtry pracující v prostorové oblasti	10
2.1.1	Obyčejné průměrování.....	10
2.1.2	Filtrace metodou mediánu	11
2.1.3	Filtrace metodou rotující masky	11
2.2	Filtry pracující ve frekvenční oblasti	13
3	Ostření obrazu.....	14
3.1	Ostření metodou rozmazání původního obrazu	14
3.2	Ostření užitím Laplaceova filtru	16
3.2.1	Laplaceův operátor	17
3.2.2	LoG filtr	18
3.3	Robertsův operátor	21
4	Kompenzace obrazů s podsvícením.....	23
4.1	Jasová analýza obrazu	24
4.2	Automatická neparametrická kompenzace	26
4.3	Implementace řešení.....	29
4.3.1	Chronologický rozbor řešení	30
4.4	HDR snímky.....	37
5	Vinětace	38
5.1	Popis jevu	38
5.2	Odstranění vlivu vinětace.....	41
5.2.1	Měření nesymetrie	42
5.2.2	Aproximace vinětační funkce s využitím modelu	43
6	Tvorba panoramatu ze sekvence snímků	45
6.1	Teoretická příprava	45
6.2	Realizace	48
6.2.1	Požadavky na vstupní sekvence snímků.....	48
6.2.2	Seznam konstant charakteristických pro danou sekvenci snímků	48
6.2.3	Struktura programu	49

7	Radiální zkreslení	57
7.1	Odstranění radiálního zkreslení posunem souřadnic obrazu.....	58
7.1.1	Nastavení koeficientů radiálního zkreslení.....	59
7.1.2	Užívané interpolace při změnách rozlišení	59
7.2	Teoretický rozbor	61
7.3	Realizace	63
7.3.1	Stanovení korekční funkce přístroje ze souboru snímků	67
8	Chromatická aberace.....	74
8.1	Teoretický rozbor	74
8.2	Realizace	77
9	Závěr	80

1 ÚVOD

Obraz je důležitým zdrojem informace. Lidé nejlépe porozumí podávané informaci, je-li sdělována prostřednictvím vizuálního obrazu. Technologie zpracování obrazu jsou proto velmi rozšířenou a komplexní vědní oblastí. Metody zpracování obrazu se stále rozvíjí a prohlubují a tento trend bude pokračovat i v budoucnu. Své využití tyto technologie nalézají téměř ve všech oblastech lidského zájmu, v medicíně, zbrojírenství, umění, dokonce i zemědělském sektoru.

Počítačové vidění je disciplína, která se snaží technickými prostředky alespoň částečně napodobit lidské vidění, tzn. kvalitu lidského senzoru – oka – a také kvalitu analýzy obrazu, protože na kvalitě lidského vjemu se podílí převážně mozek, který zpracovává informace z oka na základě svých zkušeností, očekávání, chodu myšlenek a spousty dalších faktorů. Problematika počítačového vidění zahrnuje více po sobě jdoucích procedur zpracování vizuální informace. Má práce nese název „Korekce snímků“. Bude se zabývat jednou z podkapitol týkajících se počítačového vidění.

Úkolem je upravit snímek z fotoaparátu/kamery tak, aby na něm byly minimalizovány vlivy zkreslení optické trasy, které se uplatňují při jeho pořízení. V této části práce budu navazovat na problematiku zpracování obrazu poškozeného nejběžnějšími nežádoucími jevy a sepiši podrobnou rešerši o mechanismech, které se používají a mají vliv na výslednou kvalitu pořízeného snímku. Popíši nejběžnější vady, kterými může nově pořízený snímek trpět. Zaměřím se na ty, které je potřeba odstranit pomocí patřičného softwaru. Provedu detailnější rozbor způsobů, jak tyto vady odstranit, popíši příčiny jejich vzniku a jakým způsobem znehodnocují výslednou kvalitu snímku. Pokusím se stanovit kritéria, podle kterých lze stanovit míru vlivu jednotlivých mechanismů na zhoršení vzhledu snímku. Popíši vzorce, které charakterizují jednotlivé vady, a na jejich základě se potom pokusím implementovat algoritmy, které budou schopny rozeznat vliv zkreslení uvnitř obrazu a odstranit tato zkreslení dle volby příslušného vzorce, který zkreslení co nejlépe vystihuje. Volba řešení bude záviset i na konkrétních požadavcích konkrétní aplikace, jako je například míra automatizace navrženého algoritmu, univerzálnost, rychlost a výpočetní náročnost nebo kvalita zpracování.

2 FILTRACE ŠUMU

Filtrace šumu je velmi užívanou metodou pro zlepšení kvality obrazu. Obvykle rozlišujeme dva druhy šumu – bílý šum a Gaussův šum. Bílý šum je charakteristický tím, že pravděpodobnost jeho výskytu je pro všechny frekvence stejná. Pravděpodobnost výskytu Gaussova šumu je dána Gaussovým rozložením. Máme-li k dispozici sérii snímků téhož obrazu, jako nejjednodušší možnost potlačení šumu se jeví získání výsledného obrazu průměrováním ze všech vzorků nebo pixely vypočítat prostřednictvím mediánu. Potíž zde může činit jen nalezení příslušných pixelů, které si vzájemně odpovídají. V praxi však většinou řešíme případ, kdy potřebujeme odstranit šum z jediného exempláře předlohy. S výhodou můžeme využít znalosti charakteristiky šumu, v opačném případě šum odstraňujeme pouze na základě informací o rozložení jasu v okolí každého pixelu. [5, 13]

2.1 Filtry pracující v prostorové oblasti

Efektivnost metod pro odstranění šumu závisí na velikostech objektů. Za předpokladu, že oblasti poškozené šumem budou ve srovnání s velikostmi objektů malé, je možné šum v obraze odstranit pomocí technik založených na průměrování hodnot jasu v blízkém okolí bez podstatného rozmazání objektů. [5, 13]

2.1.1 Obyčejné průměrování

Předpokládáme, že v obraze je velmi pravděpodobné, že sousední obrazové elementy mají tutéž nebo podobnou hodnotu jasu. Nejjednodušší metoda, která se nám nabízí, je obyčejné průměrování, při němž je nová hodnota pixelu počítána jako aritmetický průměr z jeho okolí. Pro okolí rozměru 3x3 je konvoluční maska filtrace obyčejným průměrováním

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

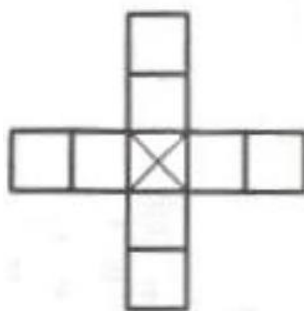
Maximální velikost použitého okna je dána nejmenšími detaily v obraze, které je potřebné zachovat. Metoda umožňuje potlačit šum vysokých frekvencí, nevýhodou je rozmazání hran, proto se většinou obyčejné průměrování používá jako metoda pomocná pro výpočet střední hodnoty jasu nebo jeho rozptylu v daném bodě. Výsledky dále

využívají důmyslnější filtrační metody. Například filtr pro potlačení Gaussova šumu vzniká snížením vah směrem od středového bodu masky. [5, 13]

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

2.1.2 Filtrace metodou mediánu

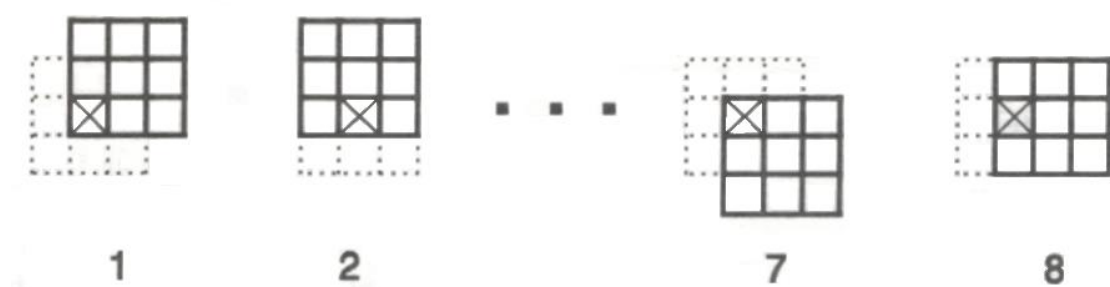
Máme-li množinu prvků o daných hodnotách, představuje medián hodnotu prvku ležícího uprostřed seřazené posloupnosti tvořené těmito prvky. Filtrace metodou mediánu je aplikací lokální statistiky pixelu, kdy určitý pixel nahrazujeme mediánem získaným z hodnot pixelů v jeho okolí. Filtraci mediánem lze použít iterativně, metoda potlačuje impulzní šum a redukuje stupeň rozmazání hran, porušuje ale tenké čáry a ostré rohy, což se eliminuje vhodnou volbou tvaru okolí, jiného než obdélníkového, např. okolí ve tvaru kříže podle Obr. 3.1. [5, 13]



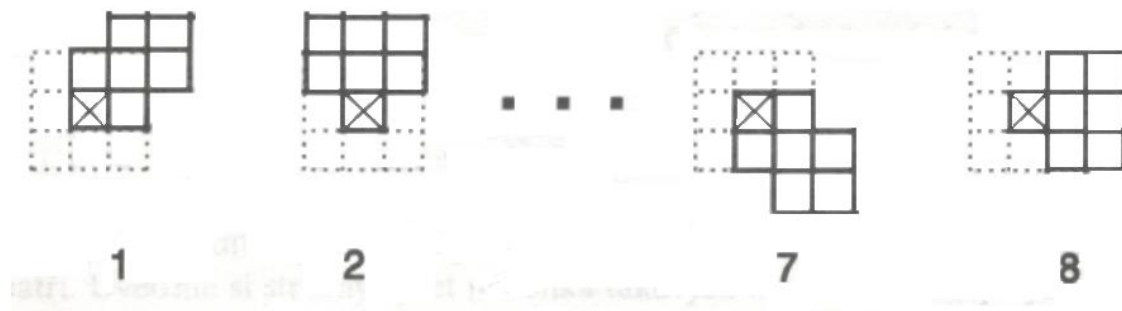
Obr. 3.1: Okolí ve tvaru kříže pro filtraci mediánem[13]

2.1.3 Filtrace metodou rotující masky

Technika se snaží najít podle homogenity jasu k filtrovanému bodu část jeho okolí, ke které pravděpodobně patří. Tato homogenní část se použije pro výpočet výsledné hodnoty jasu průměrováním. Na obrázcích Obr. 3.2 a Obr. 3.3 jsou naznačeny dvě varianty rotujících masek.



Obr. 3.2: Tvary rotujících masek v okolí 5x5. První varianta[13]



Obr. 3.3: Tvary rotujících masek v okolí 5x5. Druhá varianta[13]

Metoda nerozmazává hrany a má mírně ostřící charakter. Ve výsledku šum odstraní velmi dobře, ale obraz se skládá z malých plošek stejného jasu.

Na obrázcích Obr. 3.4 vidíme konkrétní výsledek metody obyčejného průměrování a filtrace mediánem aplikovaných na zašuměný obraz.



Obr. 3.4: Odstranění šumu z obrázku metodami filtrace obyčejného průměrování a mediánem[6]

2.2 Filtry pracující ve frekvenční oblasti

Filtrace ve frekvenční oblasti působí rozmazání vysokých frekvencí, neboť šum je v obraze reprezentován právě vysokými frekvencemi. Ty se nachází nejdále od středu Fourierova obrazu (čím dále od středu, tím vyšší). Rozlišujeme dva typy filtrů. Low-pass filtr vysoké frekvence jednoduše odřeže, zatímco Butterworth filtr je potlačí s určitou váhou. Podrobnější informace o obou typech filtrů jsou dostupné v literatuře [5].

3 OSTŘENÍ OBRAZU

Hlavní úkol při ostření obrazu spočívá ve zvýraznění jemného detailu v obrázku nebo v rekonstrukci ztraceného detailu, u kterého došlo k zastření (obvykle v důsledku šumu nebo jiných jevů, například pohybu). Technika ostření obrazu se snaží zdůraznit vysokofrekvenční část spektra. [6]

3.1 Ostření metodou rozmazání původního obrazu

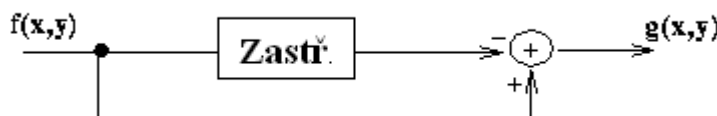
Operátory sloužící k ostření obrazu zvýrazňují hrany (a další vysokofrekvenční komponenty) tak, že se rozmazaná forma obrazu odečte od své původní verze. Toho se běžně využívá v polygrafickém průmyslu.

Proceduru lze popsat rovnicí

$$g(x, y) = f(x, y) - f_{\text{zastř}}(x, y), \quad (4.1)$$

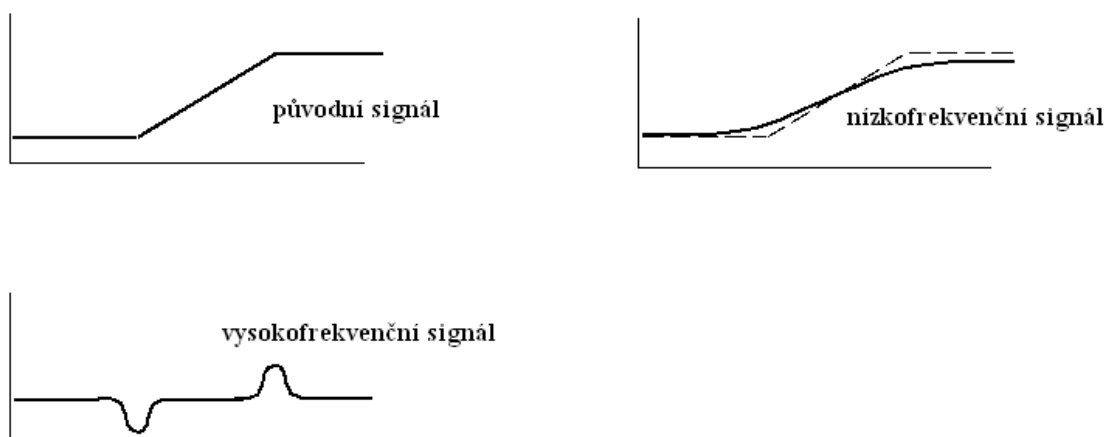
kde $g(x, y)$ v rovnici reprezentuje obraz sloužící k ostření, vstupní obraz popisuje $f(x, y)$ a $f_{\text{zastř}}(x, y)$ je zastřená verze původního obrazu.

Odpovídající blokový popis znázorňuje Obr. 4.1.



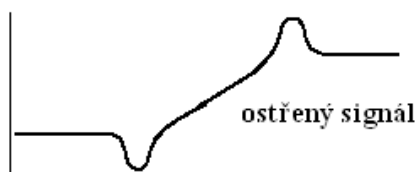
Obr. 4.1: Blokové schéma metody zvýraznění hran pomocí zastřené verze obrazu[6]

Odečteme-li od původního signálu signál z nízkofrekvenčního filtru, získáme tzv. hranovou reprezentaci. Celé funkci lépe porozumíme, prohlédneme-li si dílčí frekvenční charakteristiky. Průběhy všech tří typů signálů ukazuje Obr. 4.2.



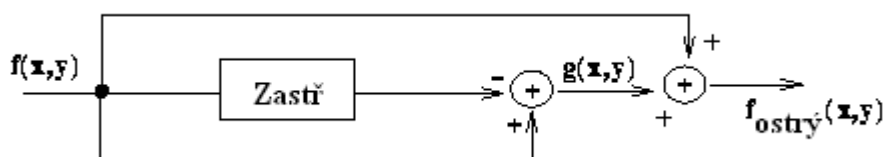
Obr. 4.2: Získání hranové reprezentace obrazu[7]

Hranovou reprezentaci můžeme použít k ostření původního obrazu. Po přičtení ke vstupnímu signálu podle schématu na Obr. 4.1 obdržíme průběh podle Obr. 4.3.



Obr. 4.3: Ostřený signál při aplikaci metody zvýraznění hran[7]

Celou metodu je potom možné graficky znázornit níže uvedeným blokovým schématem.



Obr. 4.4: Kompletní schéma metody rozmazání původního obrazu[7]

Nyní můžeme postup popsat jedinou rovnicí

$$f_{\text{ostřý}}(x, y) = f(x, y) + k \cdot g(x, y), \quad (4.2)$$

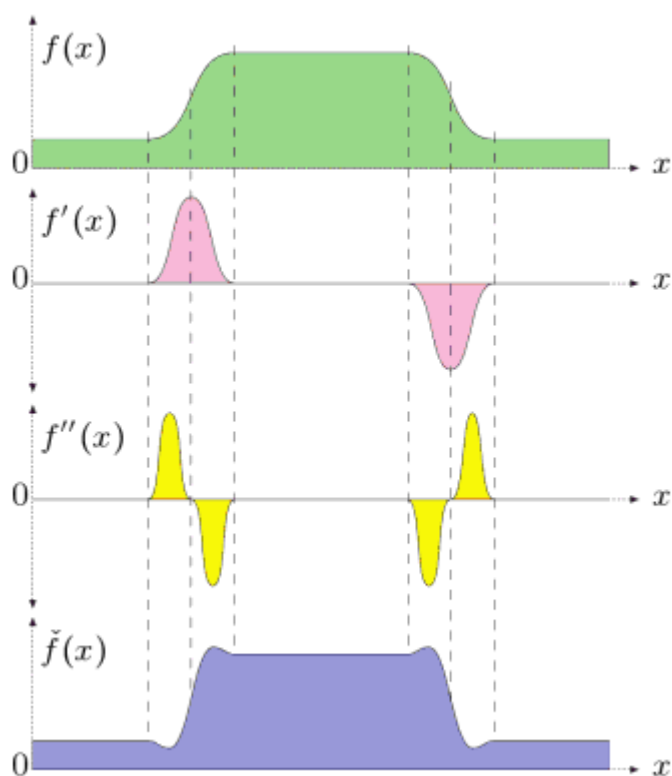
kde koeficientem k určíme váhu ostření. Čím vyšší je jeho hodnota, tím ostřejší bude konečný obraz. Hodnotu nastavujeme v rozmezí od 0,2 do 0,7.

3.2 Ostření užitím Laplaceova filtru

Laplaceův filtr patří mezi filtry pracující s 2. derivací jasové funkce. Obecný princip lze shrnout do následující rovnice

$$\tilde{f}(x) = f(x) - w \cdot f''(x). \quad (4.3)$$

Význam rovnice je podrobně vysvětlen na Obr. 4.5.



Obr. 4.5.: Zvýraznění hran pomocí 2. derivace jasové funkce[8]

Druhá derivace odezvy na jednotkový skok vykazuje kladný pulz na počátku náběžné hrany a záporný pulz na jejím konci. Hrana je zaostřena odečtením násobku w druhé derivace funkce od funkce primitivní. [8]

Podle volby váhového koeficientu $w \geq 0$ může výraz v rovnici (4.3) přesáhnout úroveň maxima jasové funkce, a tak hrany zvýraznit a zlepšit ostrost obrazu vnímanou člověkem.

3.2.1 Laplaceův operátor

Laplaceův operátor zdůrazní oblasti s rychlou změnou intenzity jasu, proto jej lze použít k detekci hran a ostření obrazu. Před aplikací operátoru bývá obraz často filtrován filtrem napodobujícím Gaussův filtr, čímž se sníží citlivost na šum. Vstupním obrazem je obvykle jeden obraz ve stupních šedé a výstupem tedy také černobílý obraz. Laplaceův operátor pracující s dvourozměrnou funkcí definuje způsob kombinace druhých derivací ve vodorovném a svislém směru. Značení ∇^2 Laplaceova operátoru vyjadřuje součet druhých derivací $f(x, y)$ podle vztahu

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial^2 x}(x, y) + \frac{\partial^2 f}{\partial^2 y}(x, y) \quad (4.4)$$

Jelikož diskretizovaný vstupní obraz je určen hodnotami jednotlivých pixelů, hledáme masku, kterou lze aproximovat druhou derivaci. Podobně jako první derivaci lze i druhou přibližně vypočítat několika jednoduchými lineárními filtry. Opět se nabízí více podob filtrů. Např. součtem dvou jednorozměrných

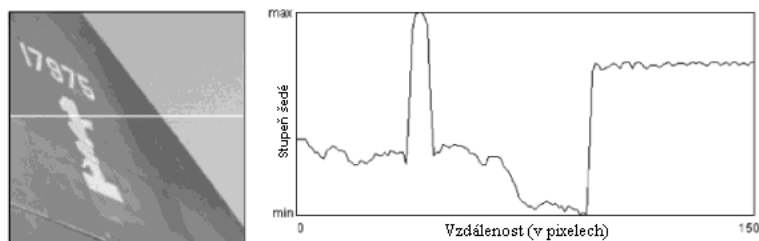
$$\frac{\delta^2 f}{\delta^2 x} = [1 \quad -2 \quad 1] \quad \text{a} \quad \frac{\delta^2 f}{\delta^2 y} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

získáme dvourozměrný Laplaceův filtr

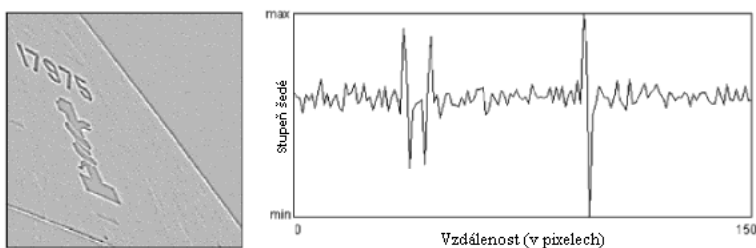
$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Všimněme si, že součet koeficientů uvnitř Laplaceova filtru je vždy nula. Proto také nemění intenzitu v místech s rovnoměrným rozprostřením jasu. [8, 9]

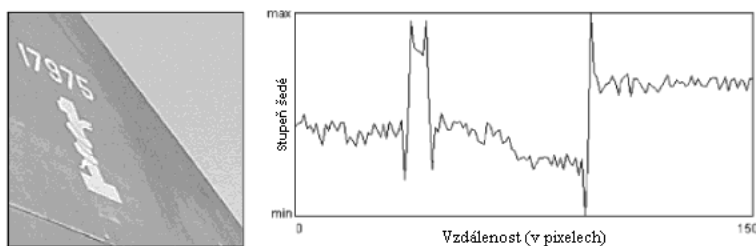
Proceduru ostření užitím Laplaceova filtru, popsanou rovnicí (4.3), na skutečném obraze vidíme na Obr. 4.6. Křivka v grafu odpovídá jasové funkci na řadě pixelů označené bílou linií v obrázku (a).



(a) Původní obraz a průběh jasu na vyznačené lince



(b) Výstup Laplaceova filtru



(c) Ostřený obraz

Obr. 4.6: Aplikace Laplaceova filtru na skutečný obraz[8]

3.2.2 LoG filtr

Jádro Laplaceova filtru aproximuje druhou derivaci, a proto je velmi citlivé na šum. Z toho důvodu se na obraz použije Gaussův filtr ještě před aplikací Laplaceova. Předzpracování obrazu odstraní vysokofrekvenční šum.

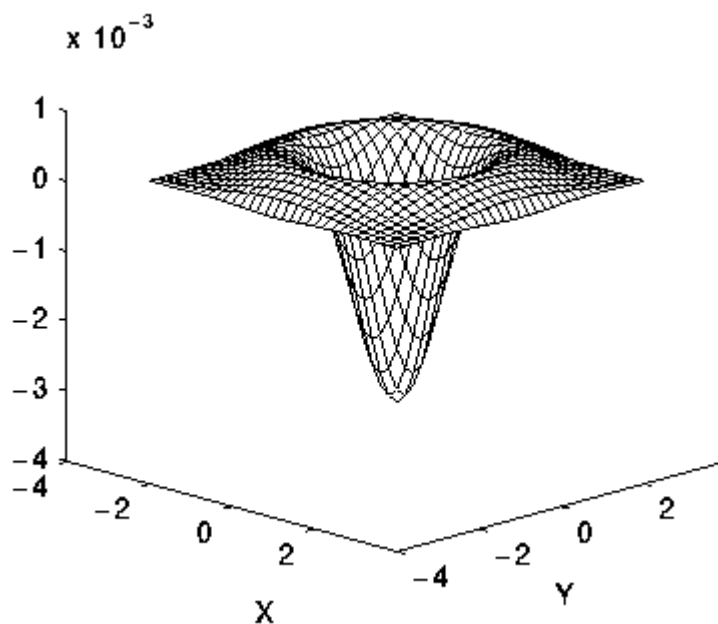
Jedná se o asociativní operace, proto je možné oba filtry spojit v jeden hybridní a aplikovat ho na obraz. Dosáhneme rovněž požadovaného výsledku. Tento postup s sebou přináší dvě výhody:

- 1) jelikož jádra Gaussova i Laplaceova filtru jsou obvykle mnohem menší než zpracováváný obraz, jsme tímto způsobem schopni výrazně snížit počet prováděných aritmetických operací.
- 2) jádro LoG filtru může být spočítáno již před zahájením práce s obrazem a není nutné jej počítat vícekrát při chodu. [9]

Dvourozměrná funkce LoG filtru vycházející z počátku souřadnicového systému má tvar

$$LoG(x, y) = -\frac{1}{\pi \cdot \sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (4.5)$$

kde σ označuje parametr rozostření.



Obr. 4.7: Dvourozměrná funkce LoG filtru[9]

Druhá derivace v rovnici (4.5) způsobí, že v oblastech o konstantní intenzitě jasu je výstupem LoG filtru nula, zatímco v těsném okolí změny intenzity bude výstupem kladná hodnota při tmavší straně hrany a záporná při světlejší. Někde uprostřed se bude

nacházet bod s nulovou výstupní hodnotou, který bude ležet na samé hraně. Výstup LoG filtru v porovnání s původním obrazem můžeme vidět na níže uvedeném obrázku.



Obr. 4.8: Zvýraznění hran užitím LoG filtru[9]

Poznamenejme, že v Obr. 4.8 byl volen LoG filtr s parametrem rozostření $\sigma = 1,0$ a jádrem o rozměru 7×7 . Uvědomme si, že výstup obsahuje záporné a desetinné hodnoty, pro zobrazení byl proto obrázek normalizován pro rozsah 0 – 255. Vhodným sloučením (čímž máme na mysli odečtení) obou obrazů získáme výsledný ostřený obraz.

LoG je filtr, který udává jen velikost hrany ale ne její směr. Směr hrany je možné určit z výstupu jiných operátorů, například Robertsova nebo Sobelova operátoru. [9, 13]

3.3 Robertsův operátor

Robertsův operátor je nejstarším operátorem. Byl proto navržen, aby byl jednoduchý a rychlý pro výpočet. Protože se operátor skládá z dvojice jader 2x2 pixely, je jeho velkou nevýhodou značná citlivost na šum. Výhoda výpočetní rychlosti operátoru je pro rychlost moderních počítačů zanedbatelná, tento operátor byl navržen jako první, nejstarší typ, v době, kdy se programátoři potýkali s malou výkonností procesorů.[10, 11]

Na Obr. 4.9 jsou uvedeny masky Robertsova operátoru.

+1	0
0	-1

G_x

0	+1
-1	0

G_y

Obr. 4.9: Dílčí masky Robertsova operátoru[12]

Masky pro oba směry jsou vůči sobě pootočený o 90° (podobnost se Sobelovým operátorem). Na obraz mohou být aplikovány postupně, změříme tak jednotlivé složky gradientu G_x , G_y . Ty lze sloučit dohromady a určit absolutní hodnotu gradientu ve všech bodech $I(x,y)$ původního obrazu podle vztahu

$$|\nabla I(x, y)| = |G(x, y)| = \sqrt{G_x^2 + G_y^2} \quad (4.6)$$

Obvykle však bývá upřednostněna nižší výpočetní náročnost, pro kterou byl tento operátor navržen, a přibližná velikost gradientu se počítá ze vztahu

$$|G(x, y)| = |G_x| + |G_y|. \quad (4.7)$$

Výpočet se tím dosti urychlí.

Směr gradientu je pak definován následovně

$$\Theta(x, y) = \arctg\left(\frac{Gy(x, y)}{Gx(x, y)}\right) \quad (4.2)$$

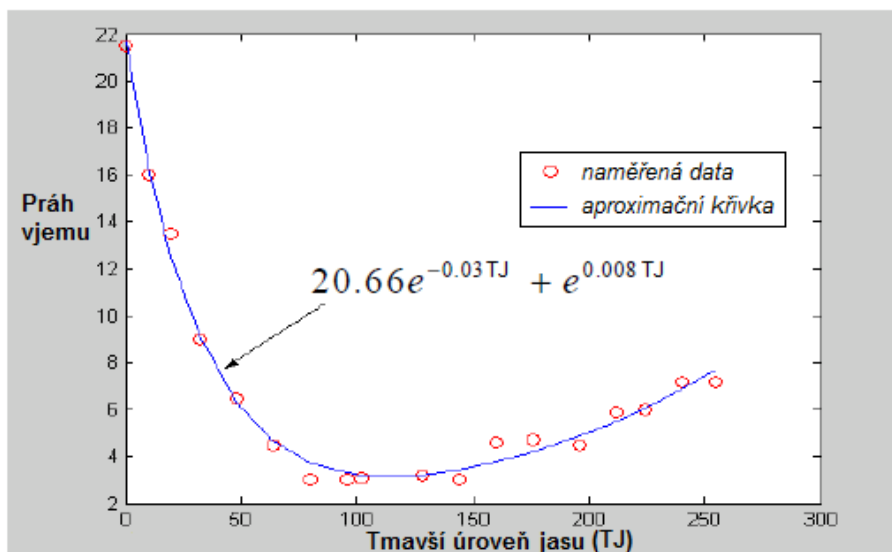
a souvisí s úhlem natočení hrany vztahem

$$\theta(x, y) = \arctg\left(\frac{Gy(x, y)}{Gx(x, y)}\right) - \frac{3}{4}\pi. \quad (4.3)$$

V tomto případě výsledná hodnota úhlu 0° značí největší kontrast z černé do bílé ve směru v obrázku zleva doprava a ostatní úhly se dále měří proti směru hodinových ručiček.[10, 11]

4 KOMPENZACE OBRAZŮ S PODSVÍCENÍM

Při pořizování snímků je třeba dbát na to, aby se zdroj vnějšího světla, například slunce, nacházel za snímacím zařízením. Nachází-li se za objektem našeho zájmu zdroj světla, způsobí zdroj světla zatmavení objektu ve snímku. Ne vždy je ale možné tomuto jevu předejít. Přestože spousta kamer a fotoaparátů disponuje řadou chytrých funkcí, obsahují vyfocené obrázky stále objekty s podsvícením. K tomu dochází v případě, že je velký rozdíl v osvětlení foceného objektu a jeho pozadí. Snímky s podsvícením mohou být potom kompenzovány jedině s využitím výpočetní techniky. Existuje řada algoritmů, které se snaží tento jev kompenzovat. Metody zvýraznění kontrastu v obraze byly vyvinuty pro zlepšení kvality obrazu vnímané člověkem s ohledem na vlastnosti zrakového systému člověka. Bylo prokázáno, že lidský zrakový systém je citlivější na prudké změny jasu (hrany) a míru jasu monotónních oblastí tolik nevnímá. Současně je známo, že odchylky jasu v obrazech jsou člověku mnohem zřetelnější, spadá-li tento jas přibližně do poloviny celkového jasového rozsahu. Na obrázku 5.1 vidíme graf vyjadřující závislost prahu odchylky dvou jasových úrovní na jejich umístění v intervalu celkového rozsahu jasových úrovní [0, 255]. Zvýraznění kontrastu obrazu bývá velmi důležitou fází předzpracování obrazu pro úlohy zpracování obrazu, zpracování videa, zpracování obrazu v medicíně a počítačovém vidění. [16,18]



Obr. 5.1: Závislost prahu vjemu rozdílu dvou úrovní jasu na jejich pozici v intervalu jasového rozsahu [16]

Klasické metody pro zvýraznění kontrastu mají dvě úskalí. Jednak většina z nich nenabízí uspokojivé výsledky při korekci lidské tváře, která je zatmavena v obraze s podsvícením, jednak řada z nich vyžaduje přímo transformační funkci a zadání parametrů manuální cestou. Pro budoucí implementaci algoritmu si zvolíme raději univerzální metodu, která oba tyto nedostatky odstraňuje. Metoda tedy funguje zcela automaticky a nevyžaduje zadávání jakýchkoli parametrů. V první fázi metoda převede obraz z barevného prostoru RGB do subtraktivního barevného prostoru YCbCr. V další fázi je obraz podroben jasové analýze pro vyšetření rozložení jasových úrovní v obraze. Vstupní obraz je dále kompenzován pixel po pixelu s využitím lineárních funkcí. Nakonec je obraz převeden zpět do barevného prostoru RGB. Celý proces proběhne automaticky a bez parametrů. Nevyžaduje zásah do procesu zvenčí. [17]

4.1 Jasová analýza obrazu

Každý barevný obrázek lze popsat kombinací Gaussovských funkcí. Získáme tak praktičtější model pro popis rozložení úrovní složek v obraze. Ke snížení doby potřebné na výpočet analyzujeme rozložení úrovní hodnot pixelů pouze jasové složky obrazu. Vycházíme z toho, že každá z Gaussovských funkcí je reprezentována jedním vrcholem a dvěma úpatími v jasové složce Y. Nejdřív spočítáme jasovou složku barevného obrazu. Poté se spočítá histogram jasové komponenty $H_Y(x_k)$. Budeme-li uvažovat histogram obrazu, jehož úrovně jasu pixelů náleží v intervalu $[0, 255]$, bude funkce $H_Y(x_k) = n_k$ diskrétní, kde x_k značí k -tou úroveň jasu a n_k je počet pixelů v obraze, které mají právě úroveň jasu x_k . Nakonec původní histogram $H_Y(x_k)$ vyhladíme Gaussovým filtrem, abychom potlačili všechny bezvýznamné vrcholy a údolí a ponechali pouze ty dominantní z nich. [17]

Vyhlazení průběhu Gaussovým filtrem funguje na následujícím principu. Konvoluce jasového histogramu $H_Y(x)$ s Gaussovým filtrem je závislá na x i σ_g , značící Gaussovu směrodatnou odchylku. Konvoluční funkce $S_{HY}(x, \sigma_g)$ je určena vztahem (5.1).

$$\begin{aligned}
S_{HY}(x, \sigma_g) &= H_Y(x) * g(x, \sigma_g) = \int_{-\infty}^{\infty} H_Y(u) g(x-u, \sigma_g) du \\
&= \int_{-\infty}^{\infty} H_Y(u) \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{(x-u)^2}{2\sigma_g^2}} du,
\end{aligned} \tag{5.1}$$

kde

* ... značí operátor konvoluce

$g(x-u, \sigma_g)$... značí Gaussovu funkci.

Míru vyhlazení průběhu určuje směrodatná odchylka Gaussovy funkce. Čím větší je směrodatná odchylka σ_g , tím je funkce $S_{HY}(x, \sigma_g)$ hladší. Rovnice (5.1) konvoluje histogram $H_Y(x)$ za účelem získání vyhlazeného histogramu. Základním hlediskem při volbě směrodatné odchylky je vyhladit nejfrekventovanější výčnělky v histogramu a zachovat jen významné módy. To znamená, že budeme schopni v původním histogramu rozlišit důležité vrcholy. Za počet vrcholů v původním histogramu budeme považovat počet vrcholů vyhlazeného histogramu.

Podmínkou je předpoklad, že rozložení drobných vrcholů a údolí stanovíme první derivací jako průměr rozdílů a pomocí té zdetekujeme významné vrcholy a údolí. Derivace v bodě x je definována jako

$$S'_{HY}(x) = \frac{1}{\sigma_g - 1} \sum_{i=1}^{\sigma_g-1} \frac{S_{HY}(x+i) - S_{HY}(x-i)}{2i} \tag{5.2}$$

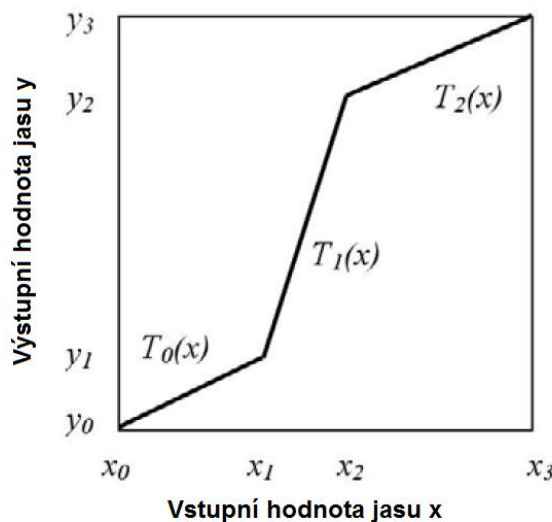
Vrchol se nachází v místě přechodu z kladných do záporných hodnot ve funkci první derivace vyhlazeného histogramu $S'_{HY}(x)$. Údolí určuje přechod ze záporných hodnot do kladných. Tak nalezneme všechny vrcholy a údolí z první derivace vyhlazeného histogramu. V případech, kdy jsou vrcholy a údolí v těsné blízkosti, je odstraníme. Vzdálenost mezi nimi musí být menší než standardní směrodatná odchylka σ_g . Ostatní vrcholy mohou posloužit k novému rozložení jasových úrovní v opraveném obraze.

4.2 Automatická neparametrická kompenzace

Navrhujeme algoritmus kompenzace, který bude založený na po částech lineární transformaci. V tomto odstavci popíšeme princip transformační funkce po částech lineární a způsob, jak tuto funkci sestavit a stanovit automaticky její parametry. Po částech lineární transformaci s počtem lineárních částí $k-1$ určuje $2k$ parametrů. To znamená, že bude-li dána počáteční hodnota vstupních jasových úrovní $x_k | k = 0, 1, \dots, k$ a počáteční hodnota výstupních jasových úrovní $y_k | k = 0, 1, \dots, k$, potom $k-1$ transformační funkce $T_{k-1}(x)$ bude:

$$T_{k-1}(x) = \frac{(y_k - y_{k-1})}{(x_k - x_{k-1})} \cdot (x - x_{k-1}) + y_{k-1}. \quad (5.3)$$

Například, pokud $k=3$, transformační funkce $T_0(x)$, $T_1(x)$ a $T_2(x)$ budou vypadat podle obrázku 5.2.



Obr. 5.2: Po částech lineární transformační funkce [17]

V tomto případě je třeba určit čtyři vstupní a čtyři výstupní parametry. Určení těchto parametrů a dílčích lineárních funkcí je nezbytné k dosažení uspokojivých

výsledků kompenzace kontrastu jasu. V klasických aplikacích jsou parametry a lineární části voleny manuálně vždy podle aktuálních požadavků. Tento problém vyřeší navržený algoritmus neparametrické kompenzace jasu. Rozložení jasu v obraze reprezentuje množství lineárních částí. Bude-li jasový histogram obrazu rozčleněn na k dílů, počet lineárních částí bude $k+1$. V automatickém provedení automatika vstupní parametr jasu stanoví na základě pozice údolí v histogramu. Různé obrazy mají různá rozmístění těchto údolí. Dále je třeba automaticky určit i výstupní parametry jasu. Jejich hodnoty rovněž závisí na konkrétním obraze, který korigujeme. [17,18]

Metoda k určení vstupních a výstupních parametrů funguje následovně. Po tom, co jsme provedli analýzu rozložení jasu, jsme obdrželi několik vrcholů. Nachází-li se v obrázku k intervalů, do kterých je soustředěno rozložení jasu, bude tento počet roven počtu nalezených vrcholů $\{p_1, p_2, \dots, p_k\}$. Vrchol značíme písmenem p (z angl. “peak”). Počet lineárních částí převodní charakteristiky je potom také k . Každý interval je v histogramu ohraničen dvěma údolími značenými v (z angl. “valley”). To znamená, že rozdělení jasových úrovní do k intervalů připadají hodnoty údolí $\{v_0, v_1, \dots, v_k\}$. Tato údolí jsou následně brána jako vstupní parametry $\{x_0, x_1, \dots, x_k\}$ po částech lineární transformační funkce. Výstupní parametry jsou dány

$$y_k = \sum_{x=s}^{v_k} \text{Pr}(x) \cdot 255, \quad (5.4)$$

kde

$\text{Pr}(x)$... je pravděpodobnost x -té úrovně jasu (stupně šedi)

s ... je počáteční hodnota jasu vstupního obrazu.

Pravděpodobnost x -té hodnoty jasu $\text{Pr}(x)$ se spočítá

$$\text{Pr}(x) = \frac{n_x}{n}, \quad (5.5)$$

kde

n_x ... je četnost výskytu této úrovně x v obraze

n ... je celkový počet pixelů obrazu.

Rovnice (5.4) slouží ke zvýraznění obrazu s běžným osvětlením. Abychom byli schopni zvýraznit zatmavenou část v obraze, musíme rovnici (5.4) ještě upravit. U obrazů s podsvícením je popředí obrazu umístěno v tmavých úrovních jasového histogramu. Jasová hodnota černé je menší než 60. Tato hodnota 60 byla stanovena experimentálně s ohledem na vlastnosti lidského zrakového vnímání.[16] Na základě studií lidského zrakového systému se uvádí, že úrovně šedé nižší než 60, jsou člověkem vnímány jako temnota. Proto rovnici (5.4) modifikujeme tak, aby se nejtmavší hodnoty jasu posunuly do světlých úrovní. Díky tomu budou tmavě osvětlené části obrazu vidět zřetelněji. Z tohoto důvodu zavádíme parametr *PosunCerne*, který řeší popsanou problematiku. Parametr *PosunCerne* nastavíme na hodnotu $PosunCerne = 60 - MaxVrcholCerne$. *MaxVrcholCerne* zde reprezentuje největší vrchol v tmavé části jasového histogramu.[17] Rovnice (5.4) poté dostane tvar (5.6) a výstupní parametry obrazů s podsvícením jsou dány

$$y_k = PosunCerne + \sum_{x=s}^{v_k} Pr(x) \cdot 255. \quad (5.6)$$

Při korekci pracujeme vždy pouze s konkrétním snímkem, výsledek korekce závisí na velikosti zatmavené oblasti. Není tedy možné provést kalibraci přístroje, velikost postižené oblasti závisí na velikosti foceného objektu a míra kompenzace je pro různé objekty různá. Vícenásobné focení téhož zatmaveného objektu také nevede ke zdárnější korekci.

4.3 Implementace řešení

Na základě popsané metody pro kompenzaci obrazu s podsvícením nyní navrhujeme program a demonstrujeme jeho funkčnost znázorněním dosažených výsledků. Program je zpracován tak, aby celý proběhl automaticky spuštěním jediného souboru. Jediným vstupním parametrem je obraz s podsvícením, který má být kompenzován.



Obr. 5.3: Vstupní obraz s podsvícením [21]

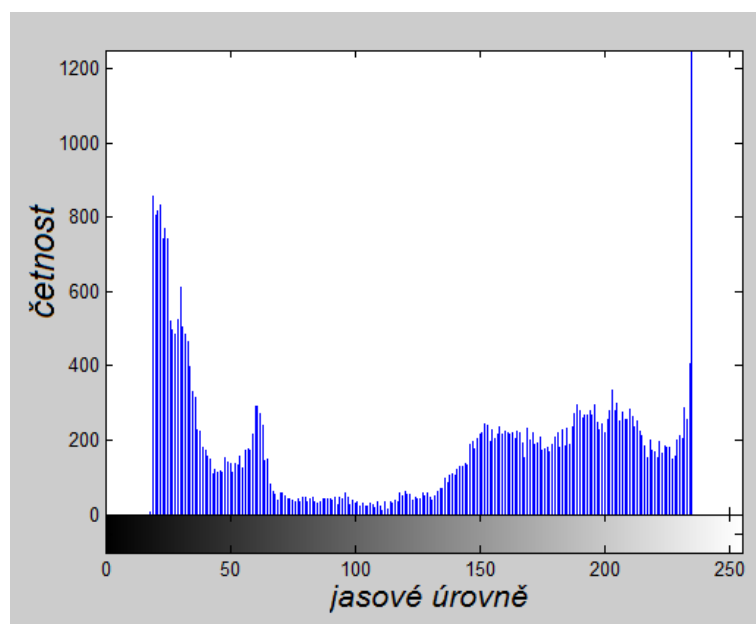
Hierarchie vzájemného volání funkcí je následující.

1. BLCompensation
 - 1.1. NajdiPaVvSHY
 - 1.1.1. OdhodPrvek
 - 1.2. VykresliHistogramsPaV
 - 1.3. Prevod
 - 1.3.1. BLUprava

4.3.1 Chronologický rozbor řešení

1. BLCompensation

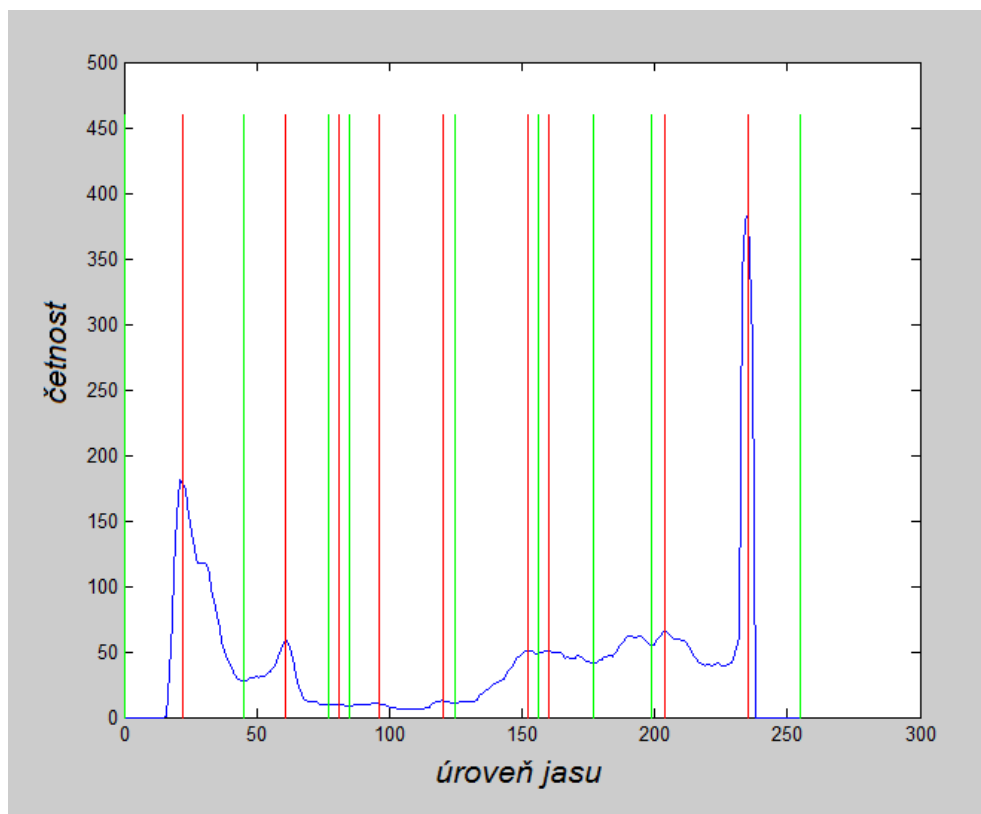
- Parametry: V této chvíli je soubor v podobě skriptu. Jediným vstupním parametrem je obraz s podsvícením, který má být kompenzován. Výstupem je zobrazení výsledků, zejména tedy srovnání původního obrazu a obrazu po kompenzaci s příslušnými histogramy.
- Popis: Nejdříve se načte vstupní obraz pro kompenzaci. Oblasti v obraze s posvícením nejsou příliš postiženy po stránce barevnosti ale spíše jasu, proto pro další analýzu vyjdeme z jasové složky. Obraz nejprve převedeme z barevného prostoru RGB do prostoru YCbCr. Oddělíme jasovou složku a spočítáme její histogram H_Y . Histogram vidíme na následujícím obrázku 5.4.



Obr. 5.4: Histogram jasové složky původního obrazu

Na původní histogram z obrázku 5.4 aplikujeme Gaussův filtr. Filtr histogram vyhladí, tzn. že v něm odstraní většinu bezvýznamných drobných výčnělků. Získáme tak hladší průběh, který budeme schopni dále použít pro řešení problému. Následuje volání funkce *NajdiPaVvSHY()* pro zjištění pozice vrcholů a údolí ve vyhlazeném histogramu – a tedy i v tom původním, nevyhlazeném, histogramu originálního obrazu. Filtr totiž zachová dominantní členitosti histogramu. Rozložení lokálních maxim a minim zachycuje filtrovaný histogram

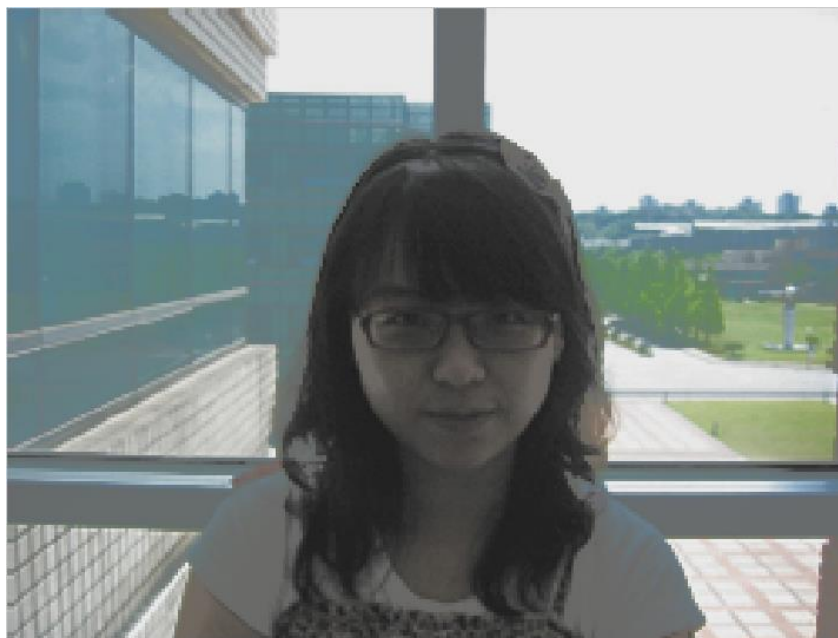
na obrázku 5.5. Červeně jsou zde zobrazeny vrcholy jako lokální maxima a zeleně údolí, lokální minima. Největší i nejmenší jasová úroveň je vždy považována za lokální minimum.



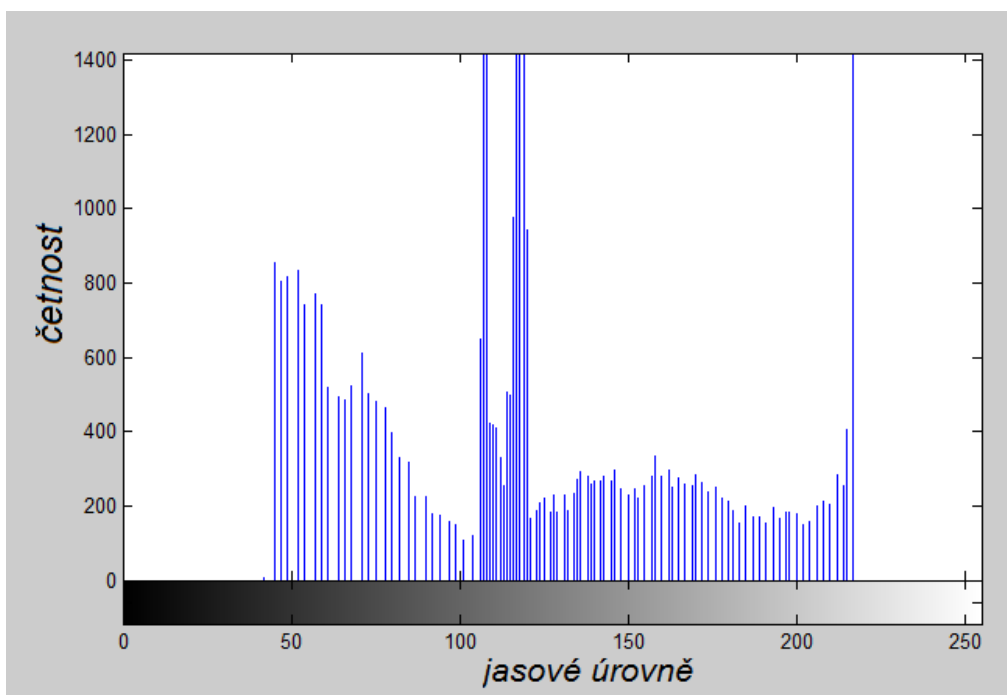
Obr. 5.5: Rozložení lokálních maxim a minim ve vyhlazeném histogramu původního obrazu

V dalším kroku určíme převodní charakteristiku pomocí funkce *Prevod()*. Na základě této charakteristiky jsme schopni transformovat celý obraz pixel po pixelu na výstupní korigovaný obraz. Míra korekce je dána velikostí odchylek převodní charakteristiky od její lineární aproximace. Přesněji řečeno, konkrétně převodní charakteristika transformuje jasovou složku originálního obrazu a touto novou jasovou složkou je nahrazena ta původní v originálním obraze. Takto opravený obraz převedeme z barevného prostoru YCbCr zpět do prostoru RGB. Výstupní obraz po kompenzaci a jeho histogram znázorňují obrázky 5.6 a 5.7. Při porovnání histogramu nové jasové složky výstupního obrazu na obrázku 5.7 s původním histogramem na obrázku 5.4 je patrný posun všech úrovní jasu

směrem ke světlejší části a také jistá změna v rozložení četností v dílčích částech histogramu.



Obr. 5.6: Výsledný obraz po kompenzaci



Obr. 5.7: Kompenzovaná složka jasu histogramu výsledného obrazu

V závěru je volána funkce *VykresliHistogramsPaV()*, která do vyfiltrovaného histogramu původního obrazu vsadí přímký pro znázornění vrcholů a údolí s jeho vykreslením.

1.1. [p, v, der] = NajdiPaVvSHY (SHY, S)

- Parametry: vstupní – vyhlazený histogram jasové složky obrazu *SHY*, sigma parametr Gaussova filtru *S* pro vyhlazení průběhu

výstupní – hodnoty lokálních maxim (vrcholů) *p* a lokálních minim (údolí) *v* ve vyhlazeném histogramu, funkce derivace vyhlazeného histogramu *der*
- Popis: Funkce vyhledá v již předpřipraveném vyhlazeném histogramu hodnoty vrcholů a údolí potřebné pro zhotovení převodní charakteristiky. Funkce nejdříve průběh vstupního histogramu zderivuje podle rovnice 5.2. Derivace podle této rovnice se snaží kompenzovat případné drobné nerovnosti ve vyhlazeném histogramu, které zde i po filtraci mohly zůstat a porušit tak monotónnost průběhu, a také lépe odhadnout (aproximací z okolních hodnot) směrnici v daném bodě histogramu. Činí tak vždy s ohledem na velikost parametru směrodatné odchylky *S*. Všechna lokální maxima a minima v derivované funkci odpovídají jejím nulovým hodnotám. Závislost histogramu v okolí lokálního maxima má vždy konvexní průběh, vrchol proto detekujeme v bodě kladné (případně nulové) funkční hodnoty derivované funkce, po němž bezprostředně následuje záporná funkční hodnota. Naopak za údolí považujeme bod derivované funkce, ve kterém je funkční hodnota záporná (popř. nulová) a hned za ní mění funkční hodnota své znaménko. Vyhledávací automatika čerpá z axiomu, že mezi dvěma sousedními vrcholy musí pokaždé ležet údolí a každé údolí je ohraničeno právě dvěma lokálními maximy. Hledáme proto vždy nejprve lokální maximum a poté lokální minimum. První a poslední hodnotu jasového rozsahu označíme jako lokální minimum (zde údolí). V závěru se funkcí ještě pokoušíme snížit počet nadbytečných lokálních maxim a minim,

které se nachází příliš blízko u sebe. Za nedostatečnou vzdálenost považujeme vzdálenost menší než hodnota směrodatné odchylky použitého Gaussova filtru S . Tyto lokální extrémů označíme tak, že do nich uložíme neplatnou hodnotu. Dále funkce garantuje, že v žádném případě neodstraníme lokální minima ohraničující interval jasového rozsahu. Nakonec jsou nežádoucí vrcholy a údolí vypuštěny pomocí funkce *OdhodPrvek()*.

1.1.1. [P] = OdhodPrvek (p, x)

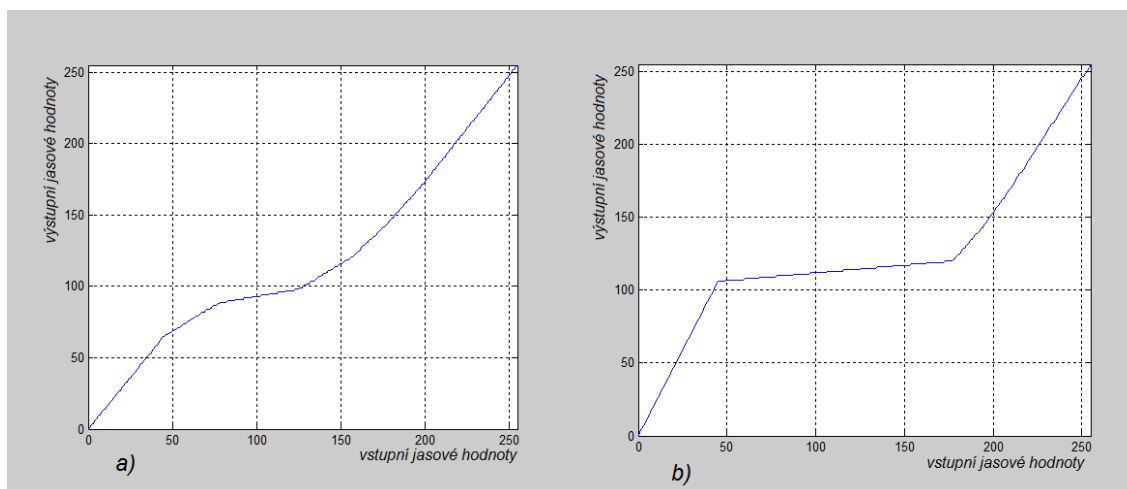
- Parametry: vstupní – vstupní fronta p , proměnná x reprezentující označení neplatných prvků fronty p
výstupní – fronta P obsahující pouze platné prvky
- Popis: Funkce vypustí nechtěné prvky z fronty p označené symbolem x (tzn. uloženou neplatnou hodnotou) a zbylé prvky vrátí ve frontě P se zachovalým pořadím. Funkce si nejprve vytvoří pomocnou proměnnou v podobě vektoru indexů o stejné délce jako vstupní fronta a neplatným prvkům přiřadí na pozici indexu podle umístění prvku ve frontě hodnotu *false*. Ostatní jsou označeny indexem *true*. Součet platných indexů v pomocném vektoru určuje délku výstupního vektoru P . Do výstupní fronty P funkce zkopíruje jen oindexované platné prvky.

1.2. [] = VykresliHistogramsPaV (H, peaks, valleys)

- Parametry: vstupní – vykreslovaný histogram H a jeho vrcholy *peaks* a údolí *valleys*
výstupní – žádné, funkce pouze vykresluje
- Popis: Funkce v histogramu znázorní vrcholy a údolí a ten poté vykreslí. Vrcholy jsou v histogramu znázorněny červenými přímkami, údolí zelenými.

1.3. [y] = Prevod (HY, valleys)

- Parametry: vstupní – histogram jasové složky původního obrazu *HY*, pozice údolí (lokálních minim) tohoto histogramu
výstupní – převodní charakteristika *y* pro transformaci na výstupní obraz
- Popis: Na základě informace o rozložení histogramu ze vstupních parametrů spočítá funkce převodní charakteristiku pro zvýšení kontrastu v postižených částech obrazu. Princip funkce *Prevod()* spočívá ve změně kontrastu pixelů příslušejících vždy dané Gaussově funkci, ze kterých je složen histogram obrazu. Každou Gaussovu funkci reprezentuje jeden vrchol v jejím středu. Funkce *Prevod()* tedy přidělí každé Gaussově funkci, tzn. pixelům v intervalu ohraničeném dvěma údolími a reprezentovanými jedním vrcholem, míru kontrastu neboli jasový rozsah. Míra kontrastu (která vychází z přiděleného jasového rozsahu) je distribuována mezi jednotlivé Gaussovy funkce (dílní intervaly jasu s jedním vrcholem) přímo úměrně počtu pixelů spadajících do daného intervalu, tzn. přímo úměrně ploše, kterou pixely z daného rozmezí jasu zaujímají v původním obraze. Tento princip popisuje rovnice (5.4). Dojde tak ke zvýraznění větších celků s homogenním jasnem a k potlačení menších objektů s velkým kontrastem. Právě takovou větší oblast tvoří popředí v obraze s podsvícením, může to být například zatmavená postava člověka stojícího před oknem. Aby zvýšení kontrastu v takovém obraze mělo význam a přispělo ke zlepšení kvality obrazu, je třeba v postižené oblasti zatmavením zesvětlit jas, protože, jak již bylo popsáno, pouze za úrovní hodnoty jasu 60 bude kontrast v postižené části dostatečně zřetelný. Právě proto definuje funkce nový parametr *BlackShift* pro zesvětlení jasu v postižené části. Parametr *Blackshift* se stanoví ze znalosti histogramu originálního obrazu a poslouží jako vstupní parametr vnořené funkce *BLUprava()*, která převodní charakteristiku přizpůsobí posunu do světlejších hodnot a vrátí i upravené hodnoty lokálních minim (údolí). Příklad převodní charakteristiky před posunem jasu a s posunem do světlejších úrovní pro obraz s podsvícením je znázorněn na obrázku 5.8.



Obr. 5.8: Převodní charakteristika kompenzace jasu u obrazů s rovnoměrným osvětlením a) se zesvětlením nejtmaších objektů b) u obrazů s podsvícením

Doposud byla převodní charakteristika ve funkci *Prevod()* reprezentována v kódu pouze významnými body, kde dochází ke zlomům dílčích lineárních funkcí. V poslední části kódu funkce *Prevod()* jsou proto dopočítány všechny zbývající funkční hodnoty převodní charakteristiky pro celý jasový rozsah.

1.3.1. [X Y] = BLUprava (x, y, BlackShift)

- Parametry: vstupní – pozice uzlů x původní převodní charakteristiky odpovídající pozicím údolí histogramu, funkční hodnoty y původní převodní charakteristiky v uzlech x , vzdálenost posunutí ke světlejší části jasového rozsahu

výstupní – pozice uzlů X upravené převodní charakteristiky a jejich funkční hodnoty Y

- Popis: Funkce přizpůsobí průběh původní převodní charakteristiky určené body x , y posunutím nejtmaších pixelů ke světlejším úrovním jasu. Vzdálenost posunutí určuje parametr *BlackShift* a výsledný průběh upravené převodní charakteristiky je vrácen prostřednictvím významných bodů X , Y . Místa napojení dílčích funkcí vstupní převodní charakteristiky odpovídají údolím histogramu

původního obrazu. Tato místa představují uzly x předané jako vstupní parametr společně s funkčními hodnotami v uzlech y a reprezentují vstupní převodní charakteristiku. Funkce *BLUprava()* posune uzel, který ohraničuje rozložení pixelů zatmaveného objektu, o vzdálenost *BlackShift* do světlejších jasových úrovní, vypustí všechny uzly, které se nachází před tímto posunutým uzlem (tzn. uzly tmavších jasových úrovní) a pozice ponechaných uzlů, které se nachází za ním, přepočítá úměrně změně polohy posunutého uzlu. Tyto nové souřadnice uzlů výstupní převodní charakteristiky jsou předány v návratové hodnotě.

4.4 HDR snímky

Cílem HDR snímků je zvětšit rozsah kresby ve světlech a stínech. Fotoaparát pořídí v rychlém sledu několik snímků a tyto skombinuje do jediného. Čím vyšší je úroveň funkce HDR, tím větší dosáhneme míry detailů ve světlech a stínech.

Postprodukční softwarová funkce HDR

Tvorba snímků HDR pomocí postprodukce využívá 2 procesy. První proces zahrnuje pořízení více snímků objektu (obvykle minimálně 3 snímky) ze stejného místa a jejich kombinaci při postprodukci do jediného snímku. Druhý způsob využívá jediný snímek, který je při postprodukci rozdělen na více snímků, u kterých je poté upravován dynamický rozsah v odlišných částech obrazu. Tyto snímky jsou poté skombinovány do jediného snímku. Tvorba snímků HDR z jediného snímku je více limitována, protože tonální rozsah jednoho snímku není tak všestranný jako výsledky získané s více snímky pořízenými různou expozicí. Výhodou způsobu tvorby snímku HDR z jediného exponovaného snímku je, že lze takto pořizovat snímky HDR pohyblivých objektů.[37]

5 VINĚTACE

5.1 Popis jevu

Vinětace se projevuje poklesem jasu ve směru od středu obrazu. Příčin může být více, zejména se jedná o větší míru pohlcení světelných paprsků na okrajích objektivů, které na objektiv dopadají pod větším úhlem a prochází silnější vrstvou materiálu. Občas se vinětace do snímků přidává i úmyslně jako umělecký prvek. Nicméně obor počítačového vidění se spoléhá na “rozumné“, poměrně pravidelné, rozložení jasu a efekt vinětace je nežádoucí pro další analýzu. Existuje celá řada metod pro odstranění vinětace z obrazu. Některé z nich vyžadují speciálně upravené prostřední pro kalibraci, obvykle rovnoměrně osvětlenou scénu. Jiné metody odstraňují vliv vinětace složením více snímků s překrývajícími okraji nebo analýzou sekvence snímků tytéž scény při různých režimech a nastavení snímacího přístroje.

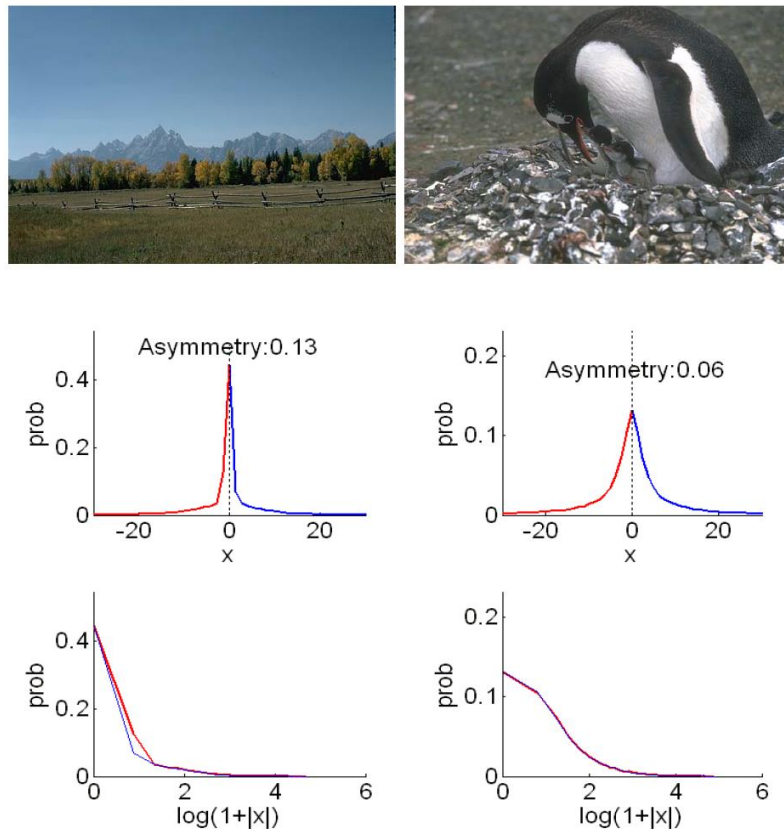
Z důvodu univerzálnosti použití se budeme zabývat způsobem korekce vinětace z jediného obrazu. Taková metoda je v praxi velmi žádoucí, protože umožňuje jednoduchou aplikaci. Často máme k dispozici pouze jeden snímek a nejsou známy parametry snímacího zařízení (typickým příkladem jsou obrázky stažené z webu). Snahou je rozlišit proměnlivost jasu danou následkem vinětačního jevu od změny jasu v přítomných texturách a přirozenou změnou osvětlení. K detekci změn jasu způsobených vinětací využijeme vlastnosti nepoškozeného obrazu (tím rozumíme obraz bez nežádoucích vlivů, jako je vinětace) – přibližně symetrického rozložení gradientu od středu obrazu (tzn. gradientů ve směru od středu obrazu ležících na soustředných kružnicích s tímto středem). Udělíme-li optickému středu (zde středu obrazu) souřadnice (x_0, y_0) , hodnotu gradientu obrazu I každého pixelu o souřadnicích (x, y) spočítáme jako

$$\psi_{(x,y)}^I = \begin{cases} |\nabla I(x, y) \cdot \vec{r}(x, y)| / |\vec{r}(x, y)| & |\vec{r}(x, y)| < 0 \\ 0 & \vec{r}(x, y) = 0 \end{cases}, \quad (6.1)$$

kde je

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix}^T, \quad \vec{r}(x, y) = [x - x_0, y - y_0]^T. \quad (6.2)$$

Podobně jako hodnoty gradientů ve vodorovném nebo svislém směru i hodnoty gradientu od optického středu vykazují v obrazech bez vinětace téměř symetrické rozložení. O této skutečnosti se můžeme přesvědčit na obrázku 6.1. Oproti tomu rozložení gradientu obrazu s vinětací bude nesymetrické, jak je naznačeno v levém dolním histogramu na obrázku 6.1. Vedle obvyklého histogramu je uveden vždy i histogram proměnné $\log(1+|x|)$. Proměnná x vždy značí hodnotu gradientu, zatímco “*prob*” relativní četnost. Histogram proměnné $\log(1+|x|)$ získáme mapováním proměnné x (gradientu) funkcí $\log(1+|x|)$. Zvýrazní se tím nesymetrie přítomná v okolí maxima histogramu. Poznamenejme, že v grafu křivka pro záporná x je přetočena na kladnou poloosu, a proto zde vidíme křivky dvě – červená představuje záporné hodnoty gradientu, modrá kladné). [19]



Obr. 6.1: Histogramy gradientů (od středu) dvou obrazů [19]

Protože průběh vinětační funkce je souměrný podle středu, je vhodné pracovat s polárními souřadnicemi:

$$Z(r, \theta) = I(r, \theta)V(r), \quad (6.3)$$

kde je

Z ... obraz poškozený vlivem vinětace

I ... obraz bez přítomnosti vinětace

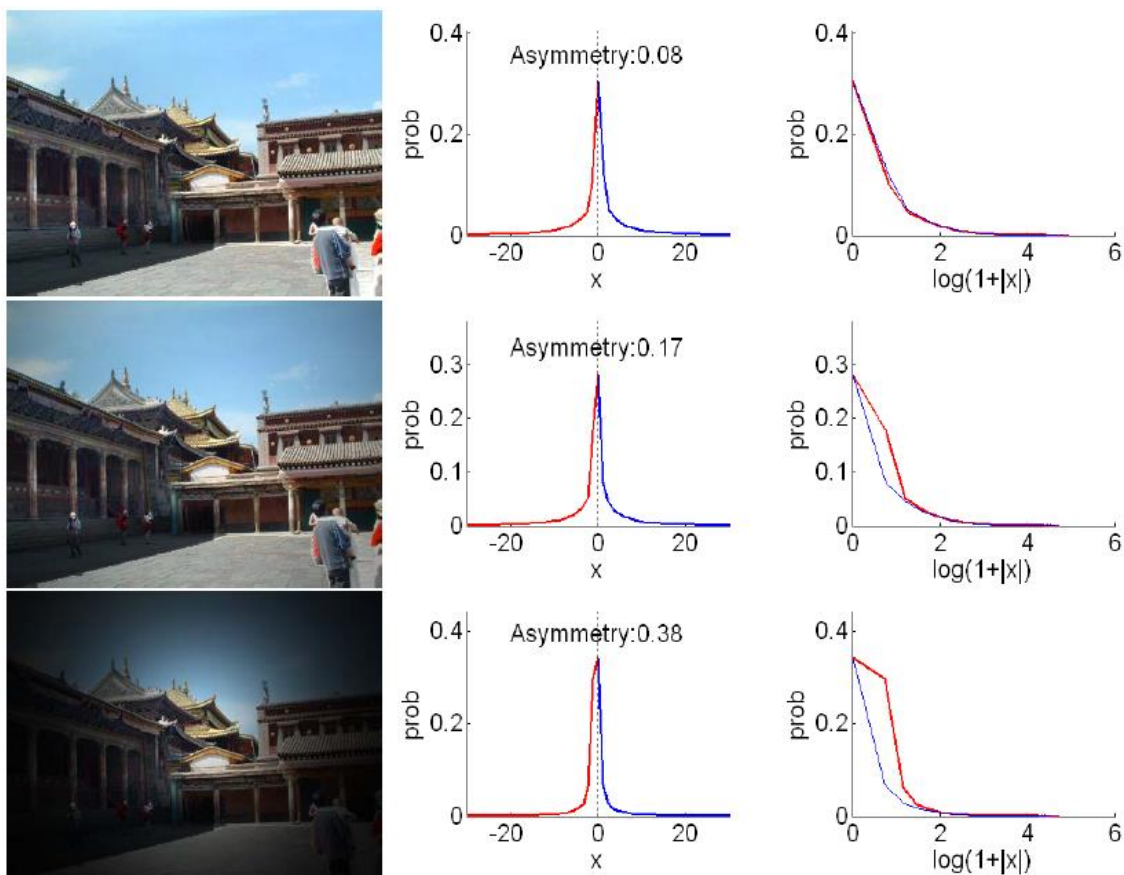
V ... vinětační funkce

Všimněme si, že V je funkcí jediné proměnné r . Je to dáno tím, že ji můžeme uvažovat symetrickou pro různá natočení. Gradient v polárních souřadnicích potom vypočteme z výrazu

$$\frac{dZ(r, \theta)}{dr} = \frac{dI(r, \theta)}{dr}V(r) + I(r, \theta)\frac{dV(r)}{dr}. \quad (6.4)$$

První výraz v rovnici (6.4) pouze váhuje gradient obrazu I vinětační funkcí V . Protože funkce V je středově symetrická, předpokládá se, že váhované rozložení jasu v prvním výrazu je u běžných obrazů přibližně symetrické. Není tomu tak ovšem u druhého výrazu. Příčinou je monotónně klesající průběh vinětační funkce směrem od optického středu (který je v našem případě uvažován jako střed obrazu), tzn. $\frac{dV(r)}{dr} \leq 0$. Protože

hodnoty jasu pixelů požadovaného obrazu I bez efektu vinětace spadají pouze do kladných čísel, má druhý výraz v rovnici vždy záporné znaménko. Rozložení hodnot jasu z druhého výrazu je tedy nesymetrické. Dále platí, že čím výraznější je vinětace, tím nesymetričtější je rozložení gradientu analyzovaného obrazu Z . U světlejších obrazů poškozených vinětací se navíc projevuje nesymetrie v rozložení jasu z druhého výrazu ještě ve větší míře. Vliv vinětace na nesymetrické rozložení gradientu ve směru od středu obrazu vidíme na obrázku (5.2).



Obr. 5.2: Nesymetrie gradientu v obraze při různých mírách vinětace [19]

5.2 Odstranění vlivu vinětace

Pro univerzální použití našeho algoritmu využijeme metodu, která je schopna odstranit vinětaci na základě jediného (námi korigovaného) snímku. Vinětační funkce je nejčastěji symetrická ke středu obrazu. Toto je také předpokladem navrženého algoritmu. Uvažujeme tedy snímek, ve kterém se efekt vinětace projevuje symetricky od středu obrazu.

5.2.1 Měření nesymetrie

Nejdříve popíšeme kvantitativní způsob měření nesymetrie distribuční funkce. Necht' $H(\psi)$ je histogram gradientu ψ ve směru od středu obrazu. Kladnou a zápornou stranu rozložení tohoto gradientu spočítáme ze vztahů

$$H_+(\psi) = \begin{cases} \frac{1}{A_1} H(\psi) & \psi \geq 0 \\ 0 & \psi < 0 \end{cases} \quad (6.5)$$

$$H_-(\psi) = \begin{cases} \frac{1}{A_2} H(-\psi) & \psi \geq 0 \\ 0 & \psi < 0 \end{cases}, \quad (6.6)$$

kde A_1 a A_2 jsou normalizační faktory, které převádí histogramy na distribuční funkce pravděpodobnosti. Jsou definovány vztahy

$$A_1 = \sum_{\psi \geq 0} H(\psi), \quad A_2 = \sum_{\psi \leq 0} H(\psi). \quad (6.7)$$

Míru odlišnosti průběhů funkcí $H_+(\psi)$ a $H_-(\psi)$ stanovíme z výrazu

$$\sum_{\psi} \left(H_+(\psi) \cdot \log \frac{H_+(\psi)}{H_-(\psi)} \right). \quad (6.8)$$

Nezapomeňme, že mohou existovat dva histogramy se dvěma podobnými pravděpodobnostními rozloženími po normalizaci. Toto ošetříme zahrnutím normalizačních faktorů do měření míry nesymetrie Γ :

$$\Gamma(I) = \lambda_h \sum_{\psi} \left(H_+(\psi^I) \cdot \log \frac{H_+(\psi)}{H_-(\psi)} \right) + (1 - \lambda_h) \cdot |A_1 - A_2|^{1/4}. \quad (6.9)$$

5.2.2 Aproximace vinětační funkce s využitím modelu

Pro řešení použijeme Kang-Weissův model, který popisuje hodnoty jasu prostřednictvím iluminačního faktoru A a geometrického faktoru G . Geometrický faktor G reprezentuje polynom.

$$V(r) = A(r)G(r) \quad r \in \Omega \quad (6.10)$$

$$A(r) = \frac{1}{\left(1 + \left(\frac{r}{f}\right)^2\right)^2}$$

$$G(r) = (1 - \alpha_1 r - \dots - \alpha_p r^p),$$

kde je

f ... efektivní ohnisková vzdálenost objektivu

$a_1, \dots, a_p$... koeficienty polynomu p -tého řádu

V ... hledaný model vinětační funkce

Odhad parametrů $f, a_1, \dots, a_p$ vinětačního modelu získáme minimalizací výrazu

$$O = \lambda \Gamma_r \left(\frac{Z}{V} \right) + (1 - \lambda) \left(\frac{N_b}{N_\Omega} \right)^{1/4}, \quad (6.11)$$

kde je

Γ_r ... míra nesymetrie obrazu $\frac{Z}{V}$ podle rovnice (6.9)

N_Ω ... celkový počet pixelů v obraze

N_b ... počet pixelů, jejichž hodnoty příspěvků vinětační funkce neleží uvnitř platného rozsahu $[0, 1]$, nebo jejichž korigované hodnoty jasu neleží uvnitř platného rozsahu $[0, 255]$.

Použití proměnné N_b ve druhém výrazu rovnice znamená, že druhý výraz penalizuje pixely, které přesahují povolený rozsah. Kang-Weissův model je blíže vysvětlen v literatuře [20].

K nalezení optimálního modelu vinětae minimalizujeme energetickou funkci v rovnici (6.11) Levenberg-Marquardtovým algoritmem. Nejdříve hledáme ohniskovou vzdálenost f tak, že geometrický faktor G položíme roven 0. Poté naopak nastavíme nalezenou ohniskovou vzdálenost a vypočítáme optimální hodnoty koeficientů $a_1, \dots, a_p$ geometrického faktoru. Nakonec použijeme odhadnuté hodnoty ohniskové vzdálenosti a geometrické koeficienty v inicializačním kroku a odhad parametrů ještě zpřesníme s využitím Levenberg-Marquardtovy metody. [19]

Použití modelu vinětae ve tvaru z rovnice (6.10) má hned několik výhod. Například dobře modeluje iluminační faktor $A(r)$ pomocí jediného parametru f . Dále závislost energetické funkce (6.11) na parametru ohniskové vzdálenosti dostane průběh, který umožní rychlou konvergenci Levenberg-Marquardtovy optimalizace při hledání přesnějšího odhadu ohniskové vzdálenosti. Díky inicializaci parametrů polynomu na 0 konverguje Levenberg-Marquardtova metoda rychle až k uspokojivému řešení. [19]

6 TVORBA PANORAMATU ZE SEKVENCE SNÍMKŮ

6.1 Teoretická příprava

Při tvorbě panoramatu program nejdříve vyzve uživatele k zadání offsetu pozic dílčích snímků prostřednictvím navzájem si odpovídajících bodů ve snímcích. Program dále spočítá koeficienty barevné korekce pro všechny dvojice sousedních obrazů, a poté vypočte globální parametr g , který redukuje kumulativní korekci barev a riziko saturace barvy. Postupně načítáme jednotlivé snímky a aplikujeme korekci barev s koeficienty barevné korekce a globálním parametrem g . Abychom připojili aktuální snímek k aktuální podobě panoramatu, vyjmeme pouze překrývající se část mezi těmito dvěma obrázky a spočítáme čtverec odchylky obrazů v matici odchylek. V matici nalezneme optimální cestu s využitím dynamického programování. Pomocí ní rozdělíme překrývající se části obrazů a obrazy spojíme dohromady. Dále se pokusíme sladit oba obrázky, tzn. redukujeme barevné rozdíly mezi nimi a zahladíme přechod mezi barevnými odstíny. Díky kombinaci korekce barev a zahlázení přechodu jsme schopni vytvářet vysoce kvalitní panoramata. Algoritmus je implementován pro tvorbu panoramat v horizontálním směru, ale není obtížné jej jednoduše pozměnit pro libovolný směr.

- Kompenzace barev

Z důvodu nerovnoměrného osvětlení scény mívají sousední snímky velké rozdíly v barevných odstínech. Abychom odstranili viditelné artefakty, je potřeba barvy nějakým způsobem přizpůsobit. Spočítáme koeficienty barevné korekce $\alpha_{c,i}$ pro konkrétní barevný kanál c z RGB i -tého snímku v sekvenci:

$$\alpha_{c,i} = \frac{\sum_p (P_{c,i-1}(p))^\gamma}{\sum_p (P_{c,i}(p))^\gamma}, \quad (7.1)$$

kde je

$P_{c,i-1}(p)$... hodnota barevného kanálu c pixelu p

γ ... koeficient gama

Koeficienty barevné korekce se počítají pouze v oblasti překryvu snímků. Koeficient gama je nastaven na hodnotu 2,2. U prvního snímku nastavíme manuálně koeficient $\alpha_{c,0}$ na hodnotu 1. Saturaci barev předejdeme zavedením globálního faktoru g_c u každého barevného kanálu c platného v celé sekvenci snímků. Požadujeme, aby

celkové přizpůsobení $g_c \alpha_{c,i}$ aproximovalo 1 a řešíme rovnici pro výpočet nejmenších čtverců. Minimalizujeme výraz

$$\min_{g_c} \sum_{i=0}^n (g_c \alpha_{c,i} - 1)^2 \quad (7.2)$$

Jedná se o kvadratickou funkci, kterou řešíme tak, že derivaci položíme rovnu 0.

$$g_c = \frac{\sum_{i=0}^n \alpha_{c,i}}{\sum_{i=0}^n \alpha_{c,i}^2} \quad (7.3)$$

Celý obraz transformujeme korekcí barev.

$$(g_c \alpha_{c,i})^{-\gamma} P_{c,i}(p) \rightarrow P_{c,i}(p), \quad (7.4)$$

kde je

$P_{c,i}(p)$... hodnota barvy pixelu p vybraného z celého rozsahu snímku S_i sekvence.

- Nalezení ideálního rozhraní a spojení obrazů

V seskládaném obraze se mohou objevit stíny v důsledku pohybu objektu a rozdílů v prostorových vzdálenostech mezi překrývajícími se částmi snímků. Za účelem odstranění nežádoucích artefaktů hledáme ideální rozhraní sousedících snímků, podél kterého oba snímky vhodně sloučíme. Díky tomu, že každý pixel ve složeném obraze bude pocházet jen z jednoho konkrétního snímku, vyvarujeme se vzniku problematických artefaktů. Pro mobilní zařízení vyžadujeme, aby navržená metoda vyhledala rozhraní mezi obrazy rychle a s malými nároky na paměť tak, abychom ji mohli aplikovat do mobilních telefonů k vytváření panoramat s vysokým rozlišením, ideální rozhraní hledáme pomocí dynamického programování. Snímky chceme sloučit na místech, kde se liší nejméně. Spočítáme matici čtverců odchylek překrývajících se částí snímků

$$e = (I_c^0 - S_c^0)^2, \quad (7.5)$$

kde je

I_c^0 ... překrývající se část aktuálního obrazu panoramatu I_c

S_c^0 ... překrývající se část aktuálního snímku S_c

S využitím dynamického programování nalezneme cestu procházející maticí e s minimálními hodnotami. Postupujeme po řádcích a počítáme kumulativní minimum kvadrátů odchylek E .

$$E(h, w) = e(h, w) + \min(E(h-1, w-1), E(h-1, w), E(h-1, w+1)), \quad (7.6)$$

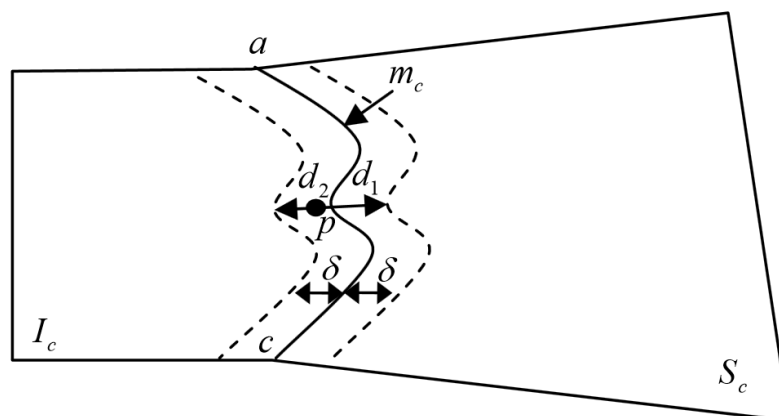
kde

$h = 2, \dots, n_r$ a $w = 2, \dots, n_c$ jsou indexy řádků a sloupců matice.

Ideální rozhraní m_c je pak cestou s nejmenšími hodnotami kumulovaných odchylek, kterou hledáme opačným směrem.

Kvalitu hledání ideálního rozhraní a spojení obrazů zlepšíme provedením barevné korekce. Zavedením barevné korekce snížíme rozdíly mezi pořízenými snímky, což pomůže zpřesnit vyhledávání vhodného rozhraní mezi snímky a rovněž redukuje možnost vzniku dalších problémů, které by mohly ohrozit správné fungování programu.

Uvažujeme dva přiléhající snímky s podobnými barevnými odstíny a jasem po barevné korekci. Sladění obrazů provedeme v pásmu širokém δ pixelů po obou stranách rozhraní podle Obr. 7.1.



Obr. 7.1: Sladění snímků[25]

Novou hodnotu barevného kanálu pixelu p z překryvné části vypočítáme váhovaním vzdáleností odpovídajících si pixelů podle vzorce

$$P_{I_c, new}(p) = \frac{d_1^n P_{I_c}(p) + d_2^n P_{S_c}(p)}{d_1^n + d_2^n}, \quad (7.7)$$

kde je

$d_1, d_2 \dots$ vzdálenosti pixelu p od okrajů

$P_{I_c, new}(p) \dots$ nová hodnota barevného kanálu pixelu p

$P_{I_c}(p) \dots$ hodnota barevného kanálu pixelu p v obraze I_0

$P_{S_c}(p) \dots$ hodnota barevného kanálu pixelu p v obraze S_c

$n \dots$ volitelný koeficient průběhu sladění obrazů

Metoda je výpočetně jednoduchá a má malé nároky na paměť.

6.2 Realizace

6.2.1 Požadavky na vstupní sekvence snímků

- snímky jsou pojmenovány čísly od 1 podle pořadí v sekvenci. V programu je třeba manuálně nastavit celkový počet snímků v parametru N .
- pořízené snímky mohou být různé velikosti
- snímky nemusí zcela ležet v jedné rovině, ovšem algoritmus je implementován tak, aby snímky skládal v horizontálním směru
- vhodnější jsou snímky s vyšším rozlišením (s ohledem na výpočetní výkon)
- program neprovádí jasovou úpravu, nejsou tedy vhodné snímky s příliš odlišnou jasovou úrovní
- vývojovým prostředím je Matlab
- referenční body je třeba zadávat po celé ploše snímku, především se zaměřit na části s detaily. V případě, že se při skládání panoramatu objeví viditelná nenávaznost mezi původními snímky, lze tuto vadu odstranit zadáním většího počtu referenčních bodů právě v této oblasti.
- v případě, že se přidávaný snímek nějak výrazněji liší ve své geometrii (například trpí radiálním zkreslením), je nezbytné zadat větší počet referenčních bodů

6.2.2 Seznam konstant charakteristických pro danou sekvenci snímků

N ... počet snímků v sekvenci

num ... počet referenčních bodů pro připojení snímku ke skládanému panoramatu, místa s vyšší koncentrací bodů jsou méně postižena chybou spojení

$gama$... konstanta ve vzorci (7.1), defaultně nastavena na hodnotu 2,2, vzorec slouží k výpočtu koeficientu barevné korekce α pro každý snímek a následně globálního parametru g ve funkci *VyrovnejOdstiny()* pro korekci kanálů RGB při různých expozicích snímků

delta ... definuje okolí nalezeného rozhraní snímku a panoramatu, ve kterém je aplikována funkce pro uhlazení rozhraní *LinearBlending()* podle vzorce (7.7)

n ... exponent okolí *delta* ve vzorci (7.7), větší hodnoty mají za následek patrnější rozhraní, výhodou je méně rozmazaný obraz, příliš nízké hodnoty způsobí, že vznikne matný obraz

6.2.3 Struktura programu

Celý program se spustí souborem 'spust.m'. Volání funkcí potom probíhá následovně:

1. Spust
 - 1.1. VyrovnějOdstíny
 - 1.1.1. Alfa
 - 1.1.2. GlobalAdjFac
 - 1.2. ImageStitching
 - 1.2.1. NactiSpojnice
 - 1.2.1.1. Obarvi
 - 1.2.2. FindOffset
 - 1.2.2.1. Odchylka
 - 1.2.3. Pridej
 - 1.2.3.1. LinearBlending

Rozbor dílčích funkcí

1. Spust

- Parametry: Algoritmus je zpracováván v m-filu. Vstupními daty je tedy sekvence obrazů ve formátu 'png' a jsou očíslována podle pořadí v sekvenci. Výstupními daty je výsledný obraz zpracovaného panoramatu.
- Popis: Program načte sekvenci snímků a uloží si výšku a šířku nejmenšího obrazu ze sekvence. Pro sladění především barevných složek v obrazech je volána funkce *VyrovnějOdstíny()*, aby budoucí uživatel nespatrił náznaky

výchozích snímků skládajících panorama. Nakonec je vyhodnocen blok pro spojení sekvencí za volání funkce *ImageStitching()*. Jako vstupní parametr je třeba volit počet referenčních bodů, které bude uživatel manuálně zadávat, definujících vzájemnou návaznost mezi snímky. Výsledkem je vytvořené panorama, které můžeme pozorovat na obrázcích Obr. 7.4.

1.1. [mySS] = VyrovnajOdstíny(SS, gama)

- Parametry: vstupní – sekvence snímků pro korekci SS, konstanta *gama*
výstupní – sekvence snímků s vyladěnými barvami *mySS*
- Popis: Funkce nejdříve určí koeficient barevné korekce α . U prvního obrázku ze sekvence je koeficient volen jako hodnota 1. Koeficient se dále počítá srovnáním s předchozím obrazem ze sekvence a to vždy pouze pro konkrétní barevný kanál. Výpočet hodnoty koeficientu α je realizován funkcí *Alfa()*. Funkce *GlobalAdjFac()* určí globální parametr g a pomocí parametrů α , g , *gama* je provedena barevná transformace snímků sekvence do výstupního parametru *mySS*.

1.1.1. [alfa] = Alfa(Sz, S, gama)

- Parametry: vstupní – aktuální snímek z dílčího barevného kanálu S a snímek téhož barevného kanálu z předchozího kroku S_z , parametr *gama*
výstupní – koeficient barevné korekce *alfa*
- Popis: Koeficient *alfa* se spočte dle vzorce (7.1).

1.1.2. [g] = GlobalAdjFac(ALFA)

- Parametry: vstupní – matice koeficientů barevné korekce *ALFA*
výstupní – globální parametr g
- Popis: Výstupní parametr g se spočte podle vzorce (7.3).

1.2. [JI] = ImageStitching(J, I, num)

- Parametry: vstupní – dosavadní utvořené panorama J , aktuální snímek přidávaný k panoramatu I , počet referenčních bodů num
výstupní – nové panorama JI s přidruženým snímkem
- Popis: Po odstranění barevných odlišností v předchozí funkci *VyrovnejOdstiny()* je hlavní algoritmus tohoto programu implementován právě uvnitř této funkce. Ve funkci jsou tedy volány veškeré funkce výlučně potřebné k vytvoření panoramatu. Nejdříve je uživatel vyzván k zadání referenčních bodů. Na základě vhodně zadaných referenčních bodů funkce *FindOffset()* vyhledá ideální polohu přidávaného snímku, proces přidružení snímku ke vznikajícímu panoramatu zajistí funkce *Pridej()*.

1.2.1. [Xi Yi Xj Yj] = NactiSpojnice (J, I, num)

- Parametry: vstupní – dosavadní panorama J , přidávaný snímek I , počet referenčních bodů num
výstupní – souřadnice X_i, Y_i, X_j, Y_j referenčních bodů na obrazech I, J
- Popis: Panorama J a snímek I jsou odděleny černou mezerou. Pro správnou funkčnost programu je uživatel povinen vždy zvolit referenční bod v panoramatu a jemu odpovídající bod ve snímku, je možné i opačně. Po volbě bodů program vyznačí spojnici mezi oběma body, aby nedošlo k záměně s jinými body a docílilo se rovnoměrnějšího rozložení vybraných bodů.

1.2.1.1. [I] = Obarvi (I, x1, y1, x2, y2)

- Parametry: vstupní – souřadnice referenčního bodu v panoramatu $x1, y1$ a ve snímku $x2, y2$, obraz I pro vykreslení zadaných bodů rozdělený na obraz pro panorama a snímek
výstupní – obraz I s červeně vyznačenými referenčními body a spojnice mezi nimi

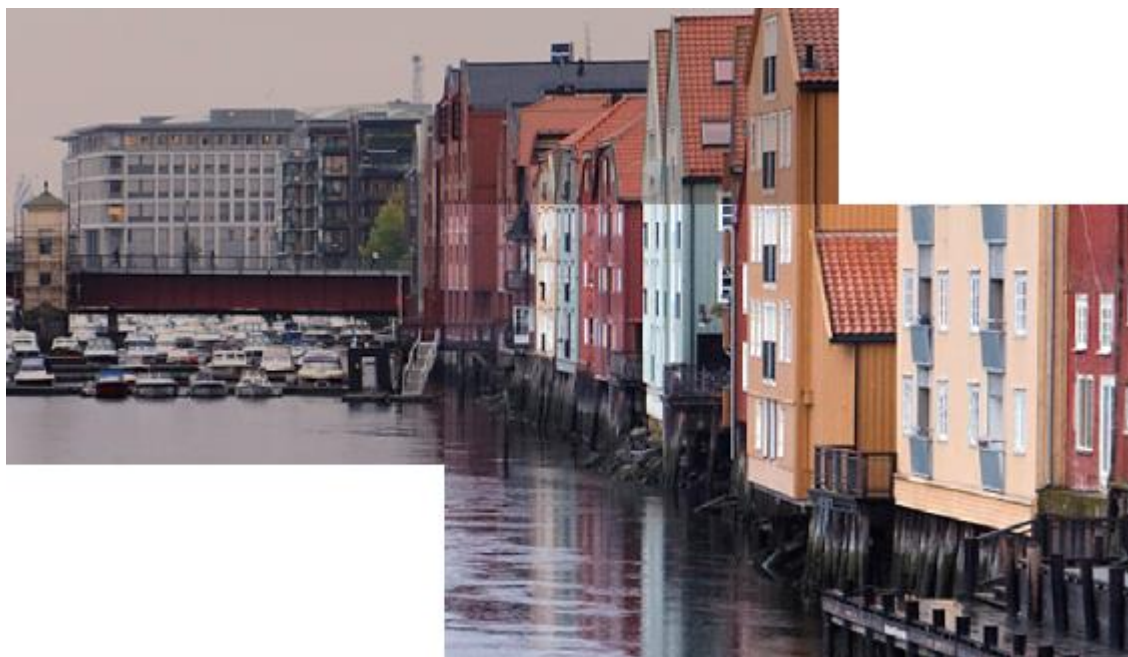
- Popis: Funkce provede pouze vykreslení referenčních bodů se spojnicí do vstupního obrazu. Obraz s vykreslenými body vidíme níže na Obr. 7.2.



Obr. 7.2: Zadávání referenčních bodů[33]

1.2.2. [Offset] = FindOffset(J, I, Xi, Yi, Xj, Yj)

- Parametry: vstupní – obraz utvářeného panoramatu J , aktuální přidávaný snímek I , souřadnice referenčních bodů $[X_i, Y_i]$, $[X_j, Y_j]$
výstupní – poloha snímku v panoramatu $Offset$
- Popis: Parametr $Offset$ odpovídá umístění levého horního rohu snímku I v panoramatu J . Zpočátku je nastaven tak, aby se panorama i snímek nepřekrývaly. Následně algoritmus hledá optimální polohu snímku I tak, aby se vzdálenost mezi referenčními body minimalizovala. Nalezenou ideální polohu pro snímek můžeme pozorovat na Obr. 7.3. V dalších krocích se pokusíme sladit rozhraní mezi oběma obrazy.



Obr. 7.3: Nalezený Offset snímku[34]

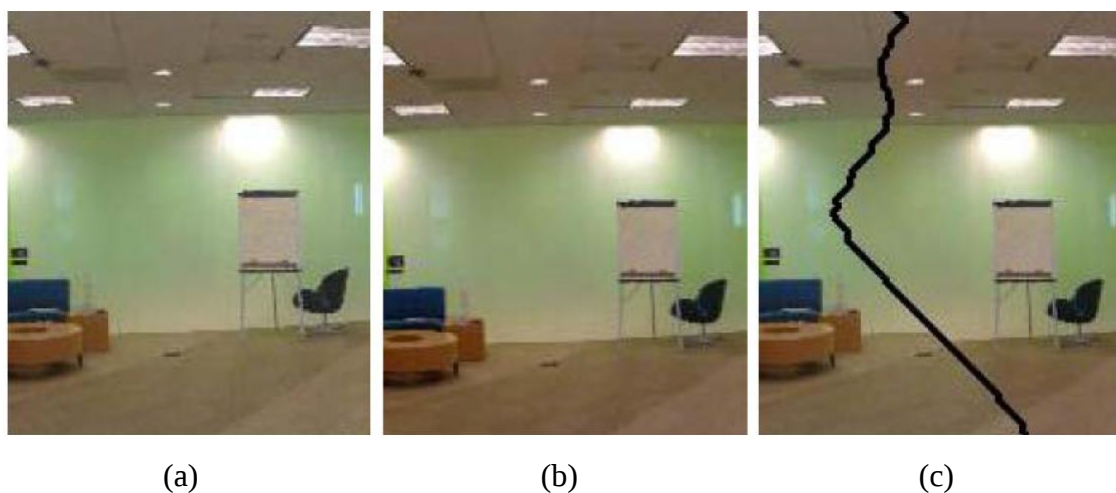
1.2.2.1. [e] = Odchylka (Offset, X_i , Y_i , X_j , Y_j)

- Parametry: vstupní – zvolená poloha snímku *Offset*, souřadnice referenčních bodů X_i , Y_i , X_j , Y_j
výstupní – vypočtená hodnota odchylky e
- Popis: Podle volby polohy snímku (tj. *Offsetu*) dojde k posunutí souřadnic referenčních bodů na snímku a funkce spočítá čtverec vzdálenosti takto utvořených dvojic referenčních bodů e .

1.2.3. [JJ] = Pridej(J , I , Offset)

- Parametry: vstupní – obraz panoramatu J , snímku I , poloha snímku vůči panoramatu *Offset*
výstupní – panorama JJ rozšířené o snímek I
- Popis: Nespadá-li přidávaný snímek zcela do horizontální roviny panoramatu, funkce neprovede ořez, ale naopak zachová původní panorama i nový snímek, tedy prázdné okolí vyplní bílým prostorem. Snímky je vhodné sloučit v místě, kde se navzájem liší nejméně. Hledá se tedy rozhraní *seam* v oblasti, kde se oba

snímky překrývají. Definice ideálního rozhraní (*seam*) byla popsána v teoretické části. Algoritmus toto rozhraní hledá pomocí kumulovaných odchylek v místě, kde se překrývá obraz s panoramatem, které sčítá shora dolů a rozhraní *seam* pak hledá zespoda nahoru. Toto rozhraní nesmí ležet příliš blízko krajů panoramatu J a snímku I s ohledem na vzdálenost *delta*, která představuje okolí rozhraní a je předána jako parametr funkci pro dodatečné uhlazení rozhraní těsně před vrácením návratové hodnoty funkce *Pridej()*. Sloučení obrazu panoramatu a snímku se znázorněným rozhraním ukazuje Obr. 7.3.



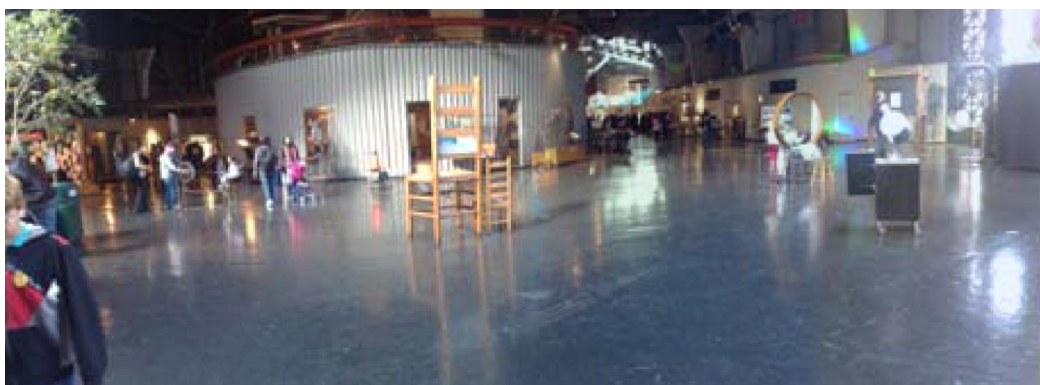
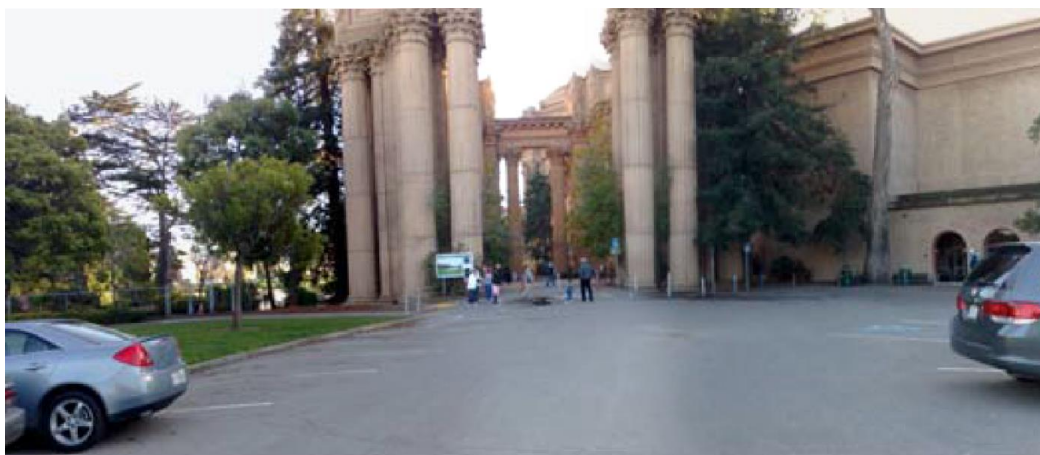
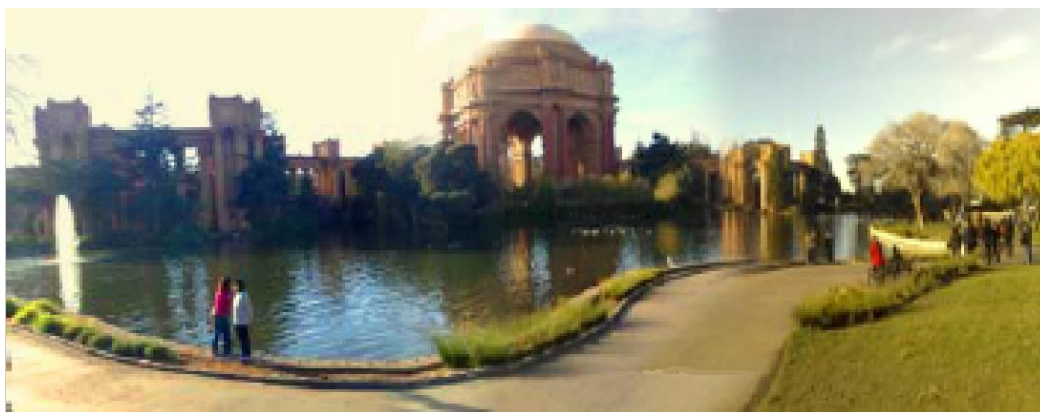
Obr.7.3: Ideální hranice s panoramatem[26]

1.2.3.1. $[J] = \text{LinearBlending} (I1I2, II, \text{seam}, \text{space}, \text{delta})$

- Parametry: vstupní – matice dvou obrazů $I1I2$, obrazu panoramatu $I1$ rozšířeného o prostor pro snímek a obrazu snímku rozšířeného o prostor pro původní panorama, obraz II panoramatu s přidruženým snímkem podle rozhraní *seam*, rozhraní *seam*, velikost okolí pro uhlazení *delta*, prodloužení panoramatu po přidání snímku *space*
výstupní – výsledný obraz panoramatu s lineární úpravou rozhraní J

- Popis: Funkce *LinearBlending()* uhladí rozhraní *seam* mezi panoramatem a snímkem podle vzorce (7.7). Aby mohl být vzorec aplikován, je třeba zajistit, aby se rozhraní nacházelo minimálně ve vzdálenosti *delta* od krajů obrazů. Tato podmínka je ošetřena.

Výsledná panoramata



Obr. 7.4: Některá testovaná panoramata[26,28,32,33]

7 RADIÁLNÍ ZKRESLENÍ

Negativním jevem, souvisejícím s optickými vlastnostmi objektivů, je soudkové a poduškové zkreslení. Častěji se setkáváme se soudkovým zkreslením, ke kterému dochází u snímků pořízených širokoúhlými objektivy. Soudkové zkreslení se projevuje zakřivením přímých linií směrem ven od středu snímku. Snímky pořízené teleobjektivy naopak znehodnocuje poduškové zkreslení, kdy jsou přímé linie obrazu zakřiveny směrem dovnitř ke středu snímku. Tyto typy deformací jsou navzájem inverzní.



Obr. 2.1: Soudkové zkreslení[2]



Obr. 2.2: Poduškové zkreslení[2]

Pokud bychom obraz zkreslený geometrickým zkreslením podrobili například detekci pohybu v obraze, výsledná rychlost pohybujícího se objektu by se lišila od skutečné, a to vždy v závislosti na míře zkreslení.

7.1 Odstranění radiálního zkreslení posunem souřadnic obrazu

Nezkreslené souřadnice označíme $\bar{q} = (x_n, y_n)^T$, souřadnice v deformovaném obraze $q = (x_d, y_d)^T$. Radiální zkreslení lze pomocí Taylorova polynomu aproximovat ve tvaru

$$\Phi(r^2) = 1 + K_1 r^2 + K_2 r^4 + \dots \quad (2.1)$$

kde jsou $K_1, K_2 \dots$ koeficienty radiálního zkreslení

$$r^2 = \bar{x}^2 + \bar{y}^2.$$

Souřadnice $\bar{x}, \bar{y}$ je třeba pro zachování radiální symetrie modifikovat tak, aby byl počátek souřadnicového systému přesunut do středu obrazu a zachovala se radiální symetrie.

Souřadnice pak udává předpis

$$\bar{x} = \frac{x_n - c_u}{R}$$

$$\bar{y} = \frac{y_n - c_v}{R},$$

kde jsou $c_u, c_v \dots$ souřadnice středu obrazu.

Přechod ze souřadnic rekonstruovaného obrazu (x_n, y_n) do souřadnic původního deformovaného obrazu (x_d, y_d) provede následující transformační rovnice

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} x_n - c_u \\ y_n - c_v \end{pmatrix} \Phi^{-1}(r^2) + \begin{pmatrix} c_u \\ c_v \end{pmatrix} \quad (2.2)$$

Rovnice umožní přepočít souřadnic jednotlivých pixelů rekonstruovaného obrazu na souřadnice deformovaného obrazu. Následně se provede diskretizace hodnot získaných souřadnic pro jednotlivé pixely v deformovaném obraze pomocí interpolace mezi sousedními pixely. [3]

Obdobně lze postupovat při transformaci ze souřadnic deformovaného obrazu do souřadnic obrazu nezkresleného. Taková transformace se řídí podle vztahu

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} c_u \\ c_v \end{pmatrix} + \begin{pmatrix} x_d - c_u \\ y_d - c_v \end{pmatrix} \Phi(r^2) \quad (2.3)$$

Problém pak spočívá v tom, že část obrazu leží mimo a kvůli nutné aproximaci nejsou transformace invertovatelné. [4]

7.1.1 Nastavení koeficientů radiálního zkreslení

Manuální nastavování koeficientů je pracné, proto se parametry určují automatickou kalibrací. Jednou z nejjednodušších metod je metoda nazývaná *plumb line*.

Plumb line algoritmus

V nezkráceném obraze vyhledáme přímkou a vyznačíme několik bodů na čáře ve zkresleném obraze, představované touto přímkou.

Probíhá vlastní algoritmus hledání koeficientů radiálního zkreslení, při kterém se body na čáře nejvíce blíží přímce.

Pomocí nalezené transformační funkce je zdeformovaný snímek promítnut do snímku bez zkreslení.

7.1.2 Užívané interpolace při změnách rozlišení

7.1.2.1 Interpolace nejbližším sousedem (*nearest neighbour interpolation, point shift, sample and hold*)

Interpolace nejbližším sousedem se řídí podle vztahu

$$g(x) = f(x_k); \frac{x_{k-1} + x_k}{2} < x \leq \frac{x_k + x_{k+1}}{2}, \quad (4.4)$$

kde funkce $g(x)$ je spojitá a funkce $f(x_k)$ diskrétní definovaná v diskrétních bodech x_k , $k = 1, \dots, n$.

Metoda na požadovanou pozici tedy okopíruje nejbližšího souseda. Jedná se o separabilní operace, tzn. kopíruje se nejbližší řádek, resp. sloupec. Jednoduchost metody předurčuje její rychlost, proto se používá zejména k rychlému náhledu (*preview*). Nižší kvalita se potom projevuje při zvětšení na hranách s malým sklonem, kdy jsou zvýrazněny skoky, a při zmenšení bývají poškozeny tenké čáry.

7.1.2.2 Bilineární interpolace

Bilineární interpolaci získáme potupným prováděním lineární interpolace. Lze ji proto zapsat vztahem

$$f(x) = f_0 + \left[\frac{x - x_0}{x_1 - x_0} \cdot (f_1 - f_0) \right] \quad (4.5)$$

kde jsou

$x_0, x_1 \dots$ souřadnice umístění dvou sousedních vzorků,

$f_0, f_1 \dots$ jejich funkční hodnoty,

$f(x)$... hodnota v našem bodě x výstupního obrazu.

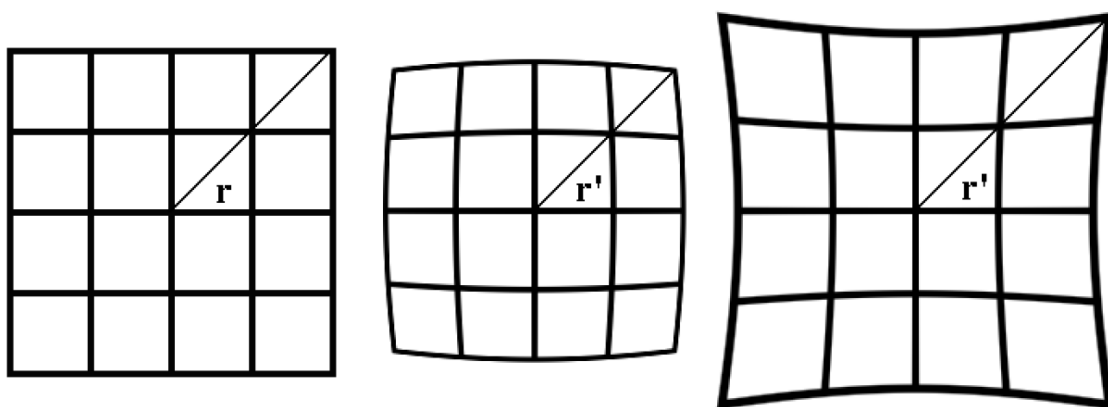
Metoda využívá pouze čtyři body z těsného okolí, a proto je rovněž rychlá. Negativním důsledkem tohoto přepočtu je rozmazání původních ostrých přechodů.

Přesnějších výsledků dosahují interpolace, které pro výpočet hodnoty v novém bodě využívají více informací z původní předlohy. Kladou samozřejmě mnohem vyšší nároky na výpočetní výkon. Některé aproximace zachovávají ostré hrany, čehož se využívá např. u obrazů obsahujících písmo, jindy jsou naopak žádoucí postupy, při kterých rozmazání hran nastává. [5]

Složitější geometrické transformace (zkreslení) se aproximují tak, že se obraz rozdělí na menší obdélníkové podobrazy. Pro každý z podobrazů se použije jednodušší geometrické transformace odhadnuté z vlíčovacích bodů. [4]

7.2 Teoretický rozbor

Radiální zkreslení je nejzávažnější ze všech optických vad. Projevuje se především u levnějších širokoúhlých fotoaparátů. Způsobuje zakřivení přímých linií. Chyba vzniká na čočce kamery nebo fotoaparátu. Míra zkreslení závisí na vzdálenosti od středu zkreslení. Směrem ke krajům obrazu se uplatňuje více. Existují dva typy této vady. Zkreslení může mít poduškovitý nebo soudkovitý tvar. U poduškovitého radiálního zkreslení dochází ke zvětšení okrajových částí obrazu, naopak u soudkovitého ke zmenšení. Průběhy obou typů vidíme na Obr. 8.1.



Obr. 8.1: Radiální zkreslení pravoúhlé mřížky. Vlevo: Neporušená mřížka. Uprostřed: Mřížka deformovaná soudkovitým zkreslením obrazu. Vpravo: Mřížka deformovaná poduškovitým zkreslením.[27]

Kvůli snížení výrobních nákladů čočky objektivů trpí vadou radiálního zkreslení, kterou je třeba následně korigovat. Bohužel se míra potřebné kompenzace vady liší u každého výrobce i u různých typů fotoaparátů stejného výrobce. Míra zkreslení závisí na ohniskové vzdálenosti čoček. Obvykle se u čoček s krátkou ohniskovou vzdáleností projevuje soudkovité zkreslení, zatímco čočky s dlouhou ohniskovou vzdáleností trpí nejčastěji poduškovitým zkreslením. Snímky pořízené různými objektivy mívají proto do jisté míry odlišný tvar.

Zkreslení se aproximuje polynomem sudého stupně. Často se použijí pouze stupně dva.

$$\begin{aligned}\Delta x &= (x' - x_0)(K_1 r^2 + K_2 r^4 + K_3 r^6) \\ \Delta y &= (y' - y_0)(K_1 r^2 + K_2 r^4 + K_3 r^6)\end{aligned}\tag{8.1}$$

Rádus r v nekorigovaném obraze se určí ze vzorce

$$r = \sqrt{(x' - x_0)^2 + (y' - y_0)^2},\tag{8.2}$$

kde jsou

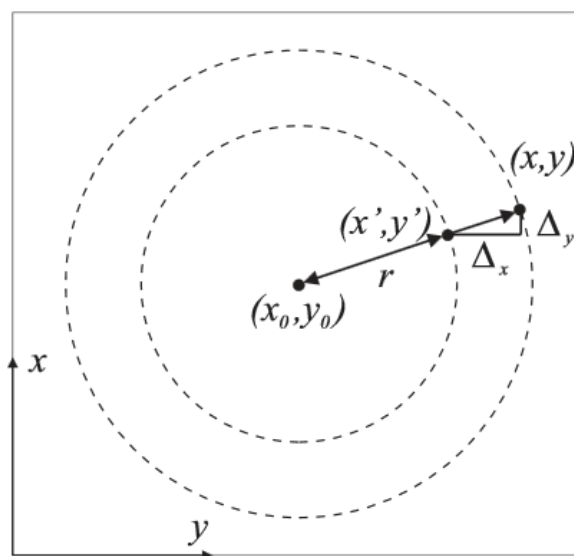
x', y' ... nekorigované souřadnice v obraze

x, y ... korigované souřadnice

x_0, y_0 ... souřadnice hlavního bodu

$\Delta x, \Delta y$... složky opravy

K_1, K_2, K_3 ... koeficienty aproximačního polynomu



Obr. 8.2: Posun souřadnic v důsledku radiálního zkreslení[29]

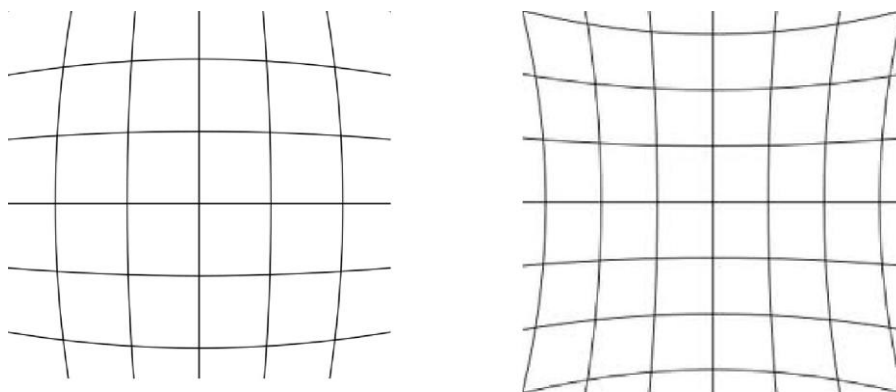
Znaménko hledaných koeficientů se liší podle korigovaného typu zkreslení – polynom s kladnými koeficienty volíme v přítomnosti soudkovitého zkreslení, záporné koeficienty, jedná-li se o poduškovité zkreslení.

Devernay použil k výpočtu radiálního zkreslení metodu přímých linií. Vycházel z principu, že netrpí-li čočka vadou radiálního zkreslení, každá přímka (úsečka) v původním snímáném obraze by měla být ve výsledném snímku zobrazena opět jako rovná, bez zakřivení.[27,29,30]

Při kalibraci zařízení je možné odvodit model chyby pro naše zařízení. Pro takovou kalibraci je výhodné použít šachovnici. Parametry modelu chyby je rovněž možné získat z různých testovacích snímků, máme-li možnost získat snímky scény tímž snímacím zařízením. Korekční koeficienty nám program vždy vypíše.

7.3 Realizace

Program jsem testoval nejdříve na dvou obrázcích zobrazujících deformovanou pravoúhlou mřížku. Testované vzorky vidíme na Obr. 8.3.



Obr. 8.3: Soudkovité (vlevo) a poduškovité (vpravo) zkreslení mřížky[22,23]

Po spuštění programu je uživatel vyzván k označení úseček, které by měly být přímé, ale v originálním obraze jsou vlivem zkreslení deformovány. Každou úsečku uživatel označí předepsaným počtem bodů, vhodně rozmístěných po celé délce úsečky. Takto označených úseček by mělo být dostatečné množství a měly by být pro správné nalezení korekční funkce rozprostřeny po celém snímku. Každá přímka úměrně snižuje rychlost výpočtu. Dále je pro správné stanovení aproximačního modelu důležité určit střed radiálního zkreslení – ten uvažuji pouze ve středu korigovaného snímku. Program postupně odhaduje míru radiálního zkreslení a pro každý z těchto odhadů počítá aproximační odchylku, kterou posléze porovnává s předchozími hodnotami. Vzorec (8.1) je implementován v programu k transformaci souřadnic námi zadaných bodů. Nové souřadnice bodů pak reprezentují úsečky v korigovaném snímku, tedy napříměné úsečky. Správnost aproximace je ohodnocena hodnotou odchylky, která odpovídá míře, s jakou nové souřadnice bodů leží v jedné přímce. Vychýlení bodů od aproximační přímky je počítáno pomocí metody nejmenších čtverců. Metoda je podrobněji popsána v literatuře [31]. Přesnost matematických výpočtů je nízká u svislých přímek, proto pro výpočet směrnice a posunutí aproximační přímky přímku otočím o 90° do horizontální polohy a poté nazpět. Směrnici přímky k a posunutí q lze z přímky otočené o 90° dopočítat

$$k = \frac{1}{k'} \quad (8.3)$$

$$q = -\frac{q'}{k'},$$

kde je

k' ... směrnice přímky otočené o 90° do horizontální polohy

q' ... posunutí přímky otočené o 90° do horizontální polohy.

Model aproximační funkce je zadán trojicí korekčních koeficientů. Pro každý z koeficientů je stanoven rozsah jeho možných hodnot a koeficienty jsou následně kombinovány každý s každým. Koeficient K_3 u šesté mocniny rádiusu ve vzorci (8.1) se při hledání modelu příliš neuplatňoval, pro urychlení výpočtu jsem ho proto často volil $K_3 = 0$. Interval koeficientů počítám ve funkci *KkrokKmax()*, kde je interval koeficientu volen od 0 po maximální hodnotu $K_{i\max}$ a délka intervalu je určena rozdílem $K_{i\text{krok}}$ dvou po sobě jdoucích koeficientů. Malá hodnota $K_{i\text{krok}}$ je důležitá pro přesnost výsledné korekce a vysoká hodnota $K_{i\max}$ pro aproximaci vysoké míry radiálního zkreslení. Oba koeficienty jsou spočteny z velikosti snímku a konstant $d_{\min}$ a $d_{\max}$ zadaných při inicializaci programu. Nalezený model je určen kombinací koeficientů, které vykazovaly nejmenší odchylku. U soudkovitého zkreslení jsou koeficienty modelu kladné, u poduškovitého záporné, znaménko koeficientů je proto nutné zadat při inicializaci programu. Ke zjištění koeficientů aproximačního modelu je použita pouze transformace souřadnic zadaných bodů úseček. Po nalezení nejvhodnějších koeficientů poté transformujeme stejným způsobem celý obraz. Nové souřadnice transformovaných pixelů jsou reálná čísla, pro vytvoření finálního obrazu je tedy nutná interpolace. Zpětnou transformaci neprovádíme. Vzniklá prázdná místa v obraze program dopočítává jako průměrnou hodnotu ze všech jeho sousedních pixelů. V případě, že počet neprázdných sousedních pixelů je menší než stanovené minimum zadané volitelným parametrem, zvětší se automaticky okolí sousedních pixelů. Program vykreslí korigovaný obraz a vypíše výslednou minimální odchylku E , koeficienty nalezeného modelu K_1 , K_2 , K_3 a uvažovaný střed zkreslení o souřadnicích x_0 , y_0 (osy x a y mají počátek v levém spodním rohu obrazu a měřítko odpovídá vzdálenosti mezi

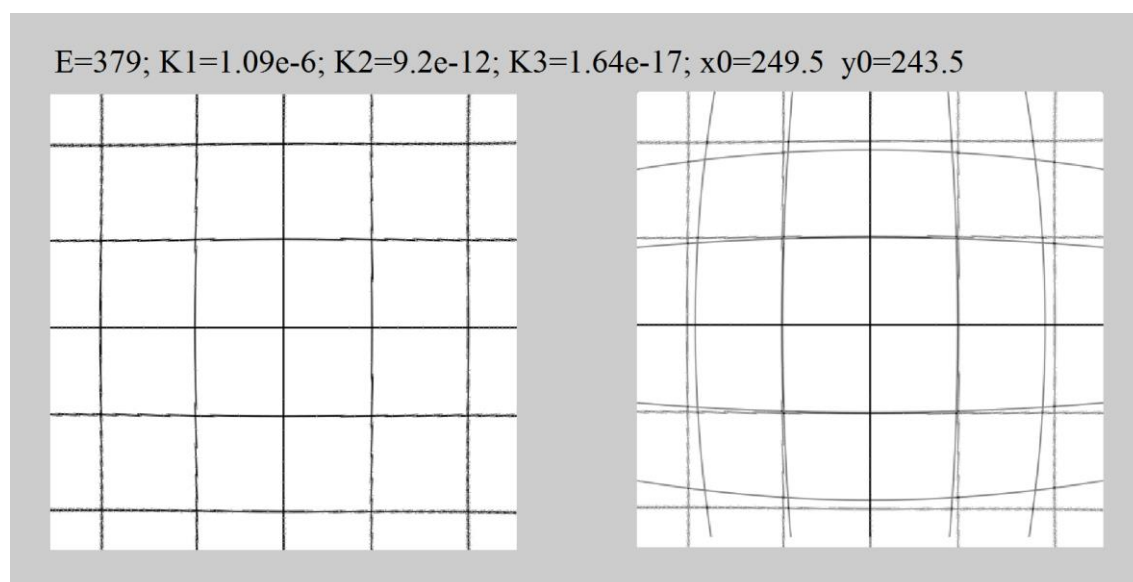
sloupce a řádky v matici obrazu). Dále vykreslí tento výstupní snímek společně s původním zkresleným v jednom obraze pro porovnání.

Korekce radiálních zkreslení v testovaném vzorku

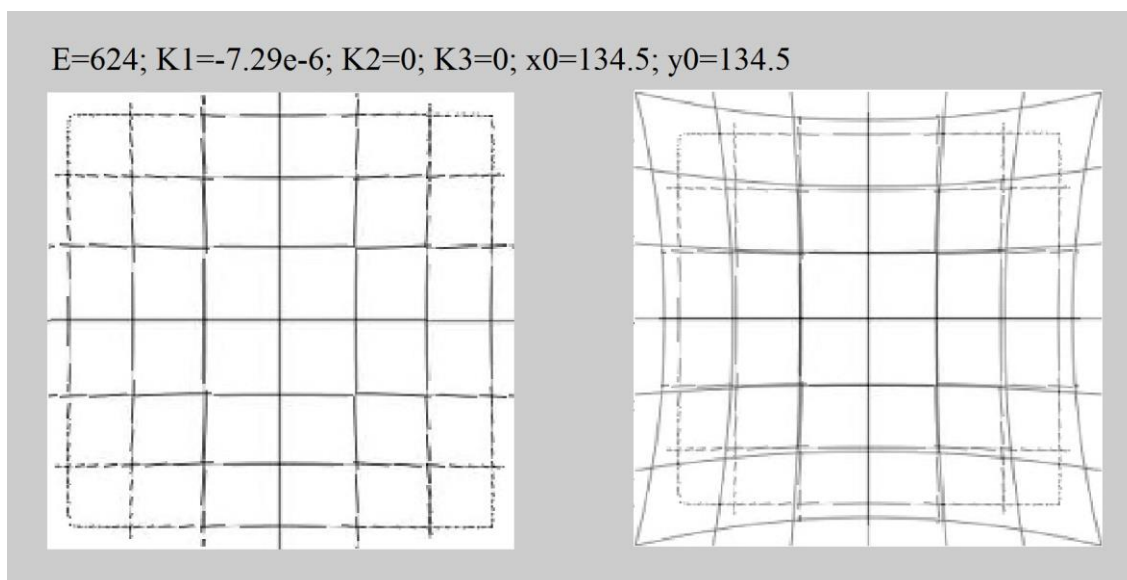
Funkčnost programu jsem ověřoval na pravoúhlé černé mřížce s bílým pozadím z Obr. 8.3. Konstanty programu byly nastaveny následovně:

```
typZkr = 1 (-1);      % typ radiálního zkreslení: +1 pro soudkovité a -1 pro  
                      % poduškovité  
N = 4;               % počet úseček hledaných v obraze  
Nb = 5;              % počet zadaných bodů, které leží na jedné úsečce  
d_min = 1;           % minimální posun při korekci  
d_max = 20;          % maximální posun při korekci  
Npr = 5;             % minimální počet sousedních pixelů pro výpočet hodnoty  
                      % prázdného pixelu
```

Výsledek korekce s naměřenou odchylkou i výsledným modelem můžeme pozorovat na obrázcích Obr. 8.4 a Obr. 8.5. Na Obr. 8.4 je zobrazena korekce mřížky deformované soudkovitým zkreslením. V levé části obrázku vidíme narovnanou mřížku, napravo je potom tato mřížka zobrazena společně s původním deformovaným snímkem v jednom obraze pro srovnání. Analogicky je výstup programu zobrazen i na Obr. 8.5, kde rozdíl spočívá pouze v tom, že se jednalo o korekci poduškovitého zkreslení.



Obr. 8.4: Výsledek po korekci soudkovitého zkreslení. Vlevo: Narovnaná mřížka po korekci. Vpravo: Porovnání výsledku s originálem.



Obr. 8.5: Výsledek po korekci poduškovitého zkreslení. Vlevo: Narovnaná mřížka po korekci. Vpravo: Porovnání výsledku s originálem.

Seznam konstant pro inicializaci programu

typZkr ... nastavíme typ radiálního zkreslení, který bude program korigovat: hodnotu 1 pro soudkovité zkreslení, hodnotu -1 pro poduškovité zkreslení

N ... počet úseček, které by měly být rovné, ale v originálním obraze jsou v důsledku zkreslení zobrazeny jako zakřivené, a které uživatel zadá pro zpracování do programu

Nb ... počet bodů, které uživatel přiřadí každé úsečce

d_min ... konstanta určuje jemnost výsledné korekce, je to rozdíl vzdáleností, s jakou dva po sobě následující odhadované modely posunou bod, který se nachází na hraně originálního obrazu nejvzdálenější od uvažovaného středu radiálního zkreslení

d_max ... konstanta určuje vzdálenost, s jakou ještě může výsledný model přesunout bod, který se nachází na hraně originálního obrazu nejvzdálenější od uvažovaného středu radiálního zkreslení

Npr ... minimální počet sousedních pixelů pro výpočet hodnoty prázdného pixelu po transformaci obrazu

7.3.1 Stanovení korekční funkce přístroje ze souboru snímků

Ve snímkách pořízených digitálním fotoaparátem se pokusíme odstranit radiální zkreslení. Využijeme k tomu výše zmíněný algoritmus. Snímky byly pořízeny mobilním telefonem MOBI Q890 s fotoaparátem o rozlišení 3,2Mpix. Jsou postiženy zkreslením typu soudek. Budeme tedy hledat koeficienty do korekční funkce (8.1). Všechny koeficienty budou kladné. Soubor snímků i vyznačené hrany, které byly zpracovány programem, můžeme vidět na Obr. 8.6-8.10.

Korekční koeficienty jsem počítal ve dvou fázích. V první fázi se na těchto patnácti snímkách uplatnil koeficient $K1$ třikrát, koeficient $K2$ čtyřikrát, zatímco koeficient $K3$ až třináctkrát. Ve druhé fázi jsem proto vyloučil koeficienty $K1$ a $K2$ ($K1=K2=0$) a korekci všech snímků provedl pouze s parametrem $K3$. Výpočet korekčních koeficientů u jednotlivých snímků pro obě fáze výpočtu je zobrazen v tabulce 8.1. Zprůměrováním hodnoty koeficientu $K3$ získáme podle vzorce (8.1) korekční funkci

$$\Delta x = 2,314 \cdot (x' - x_0)r^6$$
$$\Delta y = 2,314 \cdot (y' - y_0)r^6, \text{ kde}$$

$x_0, y_0 \dots$ uvažujeme střed snímku.

Stejnou korekční funkci jsme obdrželi při kalibraci přístroje pomocí šachovnice.

Konstanty programu byly nastaveny následovně:

typZkr = 1;	% radiální zkreslení soudkovité
N = 2 až 4;	% počet úseček hledaných v obraze
Nb = 5;	% počet zadaných bodů, které leží na jedné úsečce
d_min = 1;	% minimální posun při korekci
d_max = 20;	% maximální posun při korekci
Npr = 5;	% minimální počet sousedních pixelů pro výpočet hodnoty
	% prázdného pixelu

1. FÁZE					
Snímek	hrany	E	K1	K2	K3
1. list papíru A4 na černém pozadí	4	411,1	0	0	2,54*e-21
2. zákryt postele	3	39,21	0	0	1,69*e-21
3. zábradlí balkonu a chodník před domem	3	1,223	0	0	1,69*e-21
4. sloup a chodník před domem	2	14,87	0	0	1,69*e-21
5. hasičská věž zprava	2	37,93	0	0	8,47*e-22
6. hasičská věž zleva	2	1,306	0	8,877*e-16	2,54*e-21
7. hasičská věž - pohled zespoda	3	16,81	0	0	5,08*e-21
8. klavír - čelní pohled	4	662,8	9,309*e-10	0	1,69*e-21
9. klavír - horní levý pohled	3	8,679	9,309*e-10	2,663*e-15	0
10. klavír - spodní pohled zprava	2	1,063	0	0	0
11. skříň - čelní pohled	2	3,75	0	8,877*e-16	8,466*e-22
12. prostřední díl skříň - čelní pohled	2	9,039	9,309*e-10	0	8,466*e-22
13. prostřední díl skříň - pohled zespoda	3	90,25	0	3,551*e-15	8,466*e-22
14. spodní díl skříň - čelní pohled	3	527,5	0	0	2,54*e-21
15. spodní díl skříň - pohled shora	2	10631	0	0	2,54*e-21
2. FÁZE - jen pro vybrané snímky					
Snímek	hrany	E	K1	K2	K3
6. hasičská věž zleva	2	1,881	0	0	3,387*e-21
8. klavír - čelní pohled	4	290,3	0	0	2,54*e-21
9. klavír - horní levý pohled	3	8,49	0	0	4,233*e-21
11. skříň - čelní pohled	2	2,925	0	0	8,47*e-22
12. prostřední díl skříň - čelní pohled	2	47,92	0	0	2,54*e-21
13. prostřední díl skříň - pohled zespoda	3	15,85	0	0	2,54*e-21

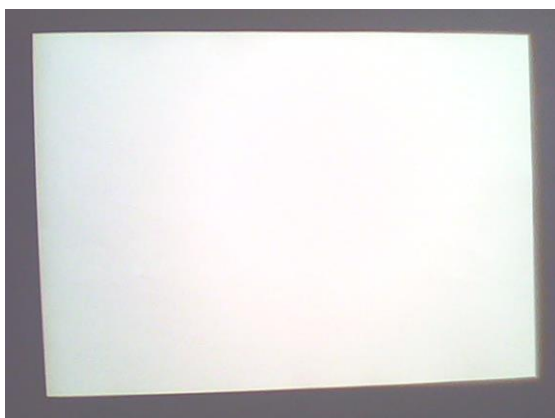
Pozn.: hrany ... počet zadaných přímých linií ve snímku

E ... vypočtená odchylka od přímek

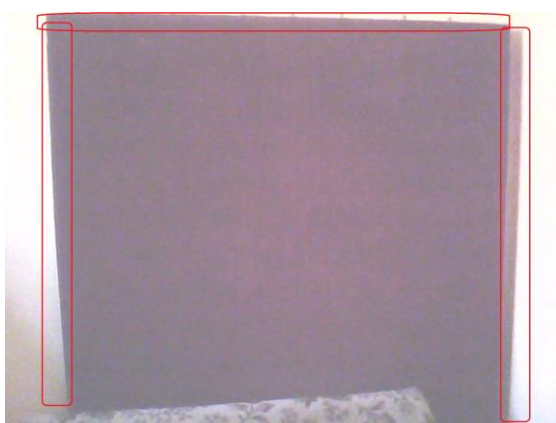
K1,K2,K3 ... nalezené koeficienty korekční funkce

Tab. 8.1: Hledání korekčních koeficientů fotoaparátu mobilního telefonu MOBI Q890

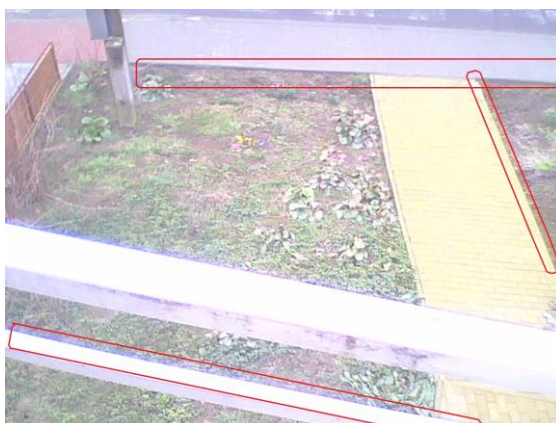
1. list papíru A4 na černém pozadí



2. zákryt postele



3. zábradlí balkonu a chodník před domem



Obr. 8.6: Soubor snímků z mobilu MOBI Q890 (1. část)

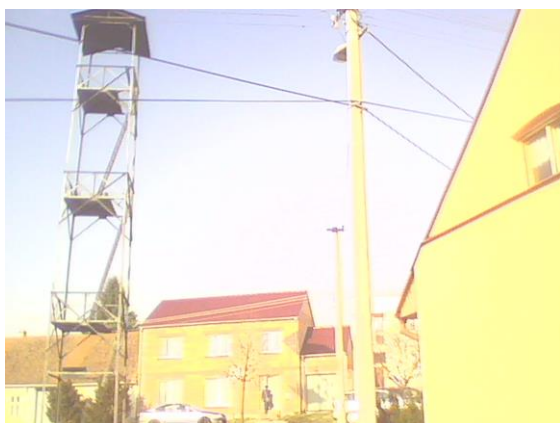
4. sloup a chodník před domem



5. hasičská věž zprava



6. hasičská věž zleva



Obr. 8.7: Soubor snímků z mobilu MOBI Q890 (2. část)

7. hasičská věž – pohled zespoda



8. klavír – čelní pohled



9. klavír – horní levý pohled



Obr. 8.8: Soubor snímků z mobilu MOBI Q890 (3. část)

10. klavír – spodní pohled zprava



11. skříně – čelní pohled



12. prostřední díl skříně – čelní pohled

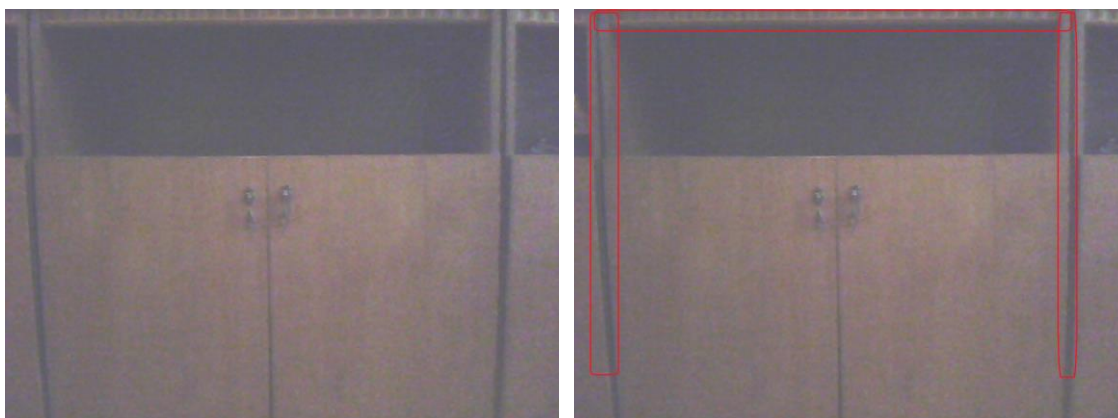


Obr. 8.9: Soubor snímků z mobilu MOBI Q890 (4. část)

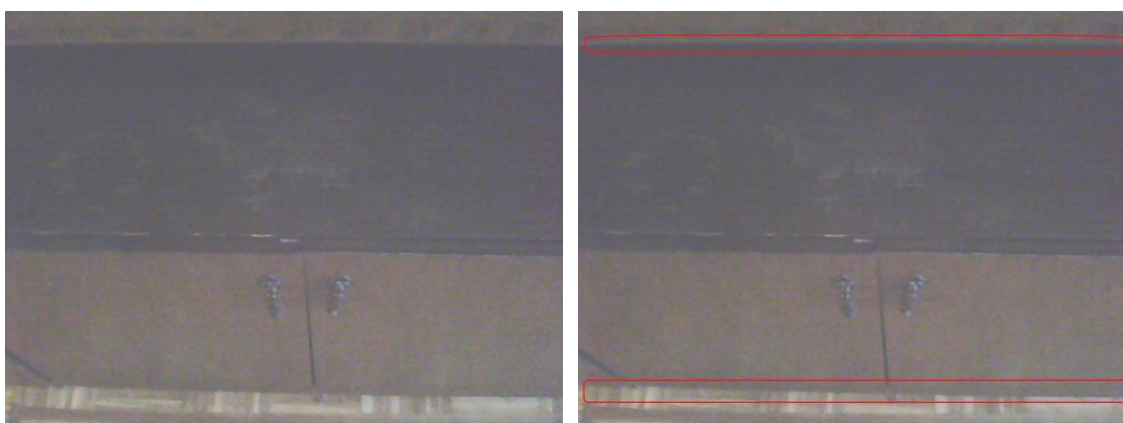
13. prostřední díl skříně – pohled zespoda



14. spodní díl skříně – čelní pohled



15. spodní díl skříně – pohled shora



Obr. 8.10: Soubor snímků z mobilu MOBI Q890 (5. část)

8 CHROMATICKÁ ABERACE

8.1 Teoretický rozbor

Chromatická aberace se projevuje nežádoucími barevnými lemy podél okrajů objektů ve snímku. Pokusíme se detekovat a eliminovat tuto vadu. Využijeme znalosti, že k vadě dochází v červeném a modrém kanálu, protože objektivy jsou obvykle kalibrovány pro zelený kanál. Algoritmus nevyžaduje žádnou apriorní znalost a lze ho proto aplikovat na kterýkoli snímek.

Na každé hraně najdeme oblast, kde se mění intenzita barev. Tuto oblast označíme Z . Lze ji definovat jako skupinu pixelů nacházejících se mezi l a r . l a r představuje levý a pravý hraniční pixel, který vymezuje hranu, tzn. přestává se zde měnit intenzita barev. Na neporušených hranách bez chromatické aberace existuje chromatická korelace, tzn. že změna v každém barevném kanálu probíhá na stejném místě. Proto spočítáme rozdíl mezi červenými a zelenými signály a rozdíl mezi modrými a zelenými signály v RGB barevném prostoru. Rozdílové signály vykazují na neporušených hranách určitou vlastnost. Každý rozdílový signál v oblasti Z má hodnotu spadající do intervalu ohraničeného hodnotami pixelů r a l . Tuto vlastnost vyjádříme

$$\begin{aligned} \min\{D_R(l), D_R(r)\} \leq D_R(j) \leq \max\{D_R(l), D_R(r)\} \\ \min\{D_B(l), D_B(r)\} \leq D_B(j) \leq \max\{D_B(l), D_B(r)\}, \end{aligned} \quad (9.1)$$

kde je

$$D_R(\cdot) = R(\cdot) - G(\cdot)$$

$$D_B(\cdot) = B(\cdot) - G(\cdot)$$

j ... umístění pixelu v oblasti Z

Tento vztah neplatí pro hrany s chromatickou aberací. Chromatická korelace je porušena, z čehož vyplývá, že ke změně intenzity barvy dochází v každém barevném kanálu na jiném místě. Takovou oblast hrany lze definovat jako skupinu pixelů, kde se mění alespoň jeden z barevných kanálů. Právě tyto pixely vytváří nežádoucí barevné lemy kolem hran. Než budeme detekovat oblasti, ve kterých je přítomna aberace, musíme u každé hrany vyhledat oblast Z . K nalezení této oblasti zderivujeme obraz pro každý z barevných kanálů. Pro derivaci obrazu použijeme Sobelův operátor. Algoritmus zpracovává data zvlášť v horizontálním a zvlášť ve vertikálním směru. Nyní budeme uvažovat pouze horizontální směr. Při zpracování v horizontálním směru si můžeme všechny 2D signály představit jako jednorozměrný případ o konstantním indexu i . V prvním kroku nalezneme hranu. Vyjdeme z počáteční hodnoty $j=1$ a inkrementací umístění pixelu j přejdeme až k umístění prvního pixelu hrany k .

$$k=j, \text{ kde } j \text{ je inkrementováno až } E_G(j) \geq T_1 \text{ a } E_{Gx}(j) \geq E_{Gy}(j) \quad (9.2)$$

T_1 je práh hrany. Opět nás zajímá pouze zelený kanál, protože objektiv většiny fotoaparátů je kalibrován právě pro zelený kanál. Jako důsledek obsahují hodnoty derivace červeného a modrého kanálu rozmazanou a geometricky posunutou informaci, což způsobuje chromatická aberace. Při zpracování obrazu ve vertikálním směru platí obrácená nerovnost derivací v zeleném kanálu $E_{Gx}(j)$, $E_{Gy}(j)$. Umístění hrany k lze považovat za jeden z bodů v oblasti Z . Hledáme celou oblast Z_k hrany k . Algoritmus vyhledává podobné nejbližší pixely z levé strany až po l_k a z pravé strany po r_k od bodu k , dokud nenarazí na homogenní oblast nebo další hranu.

$$Hx(j) = \max\{E_{Rx}(j), E_{Gx}(j), E_{Bx}(j)\} \quad (9.3)$$

H značí největší hodnotu derivace z dílčích barevných kanálů.

$$l_k = j, \text{ kde } j \text{ je dekrementací } k \text{ po } Hx(j) < T_2 \text{ nebo } E_G(j) < T_1 \leq E_G(j-1) \quad (9.4)$$

$$r_k = j, \text{ kde } j \text{ je inkrementací } k \text{ po } Hx(j) < T_2 \text{ nebo } E_G(j) < T_1 \leq E_G(j+1), \quad (9.5)$$

kde T_2 je práh pro homogenní oblast. Homogenní oblasti jsou oblasti s nízkými hodnotami pixelů derivovaných obrazů z každého barevného kanálu. Hranice homogenní oblasti se nachází v místě, kde maximum H klesne pod daný práh T_2 . Oblast Z hrany k definujeme jako oblast mezi l_k a r_k . Místa, ve kterých je přítomna aberace, určíme vyhledáním pixelů, které nesplňují podmínku pro neporušené hrany.

$$\rho_s(j|k) = \begin{cases} 1, & \text{kde } D_s(j) < \min\{D_s(l_k), D_s(r_k)\} \\ & \text{nebo } D_s(j) > \max\{D_s(l_k), D_s(r_k)\} \\ 0, & \text{jinak} \end{cases} \quad (9.6)$$

kde je

$$S \in \{R, B\}$$

$j|k$... umístění pixelu j příslušné hrany k , $l_k < j < r_k$

ρ_s ... binární signál vymezující místa s aberací

Po skončení detekce bude do j dosazena hodnota r_k+1 a algoritmus vyhledá pro analýzu další hranu.

K odstranění chromatické aberace použijeme eliminační metodu. Metoda je použita zvlášť v červeném a zvlášť v modrém kanálu. Upravuje hodnoty rozdílových signálů v oblastech, ve kterých je detekována aberace, tak, aby byly podobné minimu rozdílových

signálů přilehlých neporušených pixelů. K nalezení vhodné hodnoty pro úpravu rozdílových signálů, spočítáme nejbližší nepoškozené pixely, tj. $\rho_s = 0$, v levém i pravém směru $j_{\alpha s}$ a $j_{\beta s}$. Všechny hodnoty rozdílového signálu z $j_{\alpha s} < j < j_{\beta s}$ jsou nastaveny na hodnotu rozdílového signálu na pozici $j_{\alpha s}$ nebo $j_{\beta s}$, která vykazuje nejmenší absolutní hodnotu rozdílového signálu. Pixel s nejmenší absolutní hodnotou rozdílového signálu vybíráme proto, že tak zmaximalizujeme redukci saturačního signálu. Po úpravě hodnot rozdílových signálů, spočítáme výstupní červený a modrý signál přičtením zeleného vstupního signálu.[24]

$$S_{out}(j) = \begin{cases} D_s(j_{\alpha s}) + G_{in}(j), & D_s(j_{\alpha s}) \leq D_s(j_{\beta s}) \\ D_s(j_{\beta s}) + G_{in}(j), & D_s(j_{\beta s}) < D_s(j_{\alpha s}), \end{cases} \quad (9.7)$$

kde je $S \in \{R, B\}$ a $j_{\alpha s} < j < j_{\beta s}$

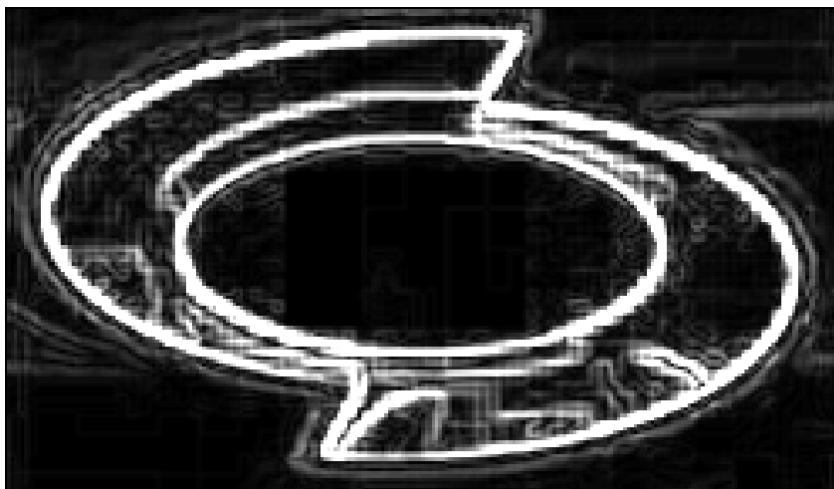
8.2 Realizace

Na základě výše popsaných úvah jsem vytvořil program pro redukci chromatické vady přítomné ve snímku. Program funguje automaticky a nevyžaduje součinnost uživatele. Pro ilustraci rozebereme, jak program postupně zpracovává snímek, aby v něm potlačil vliv chromatické aberace. Vstupním obrázkem je Obr. 9.1. Na hranách předmětu v obrázku je chromatická aberace dosti zřetelná.



Obr. 9.1: Originální obraz s chromatickou aberací[24]

Uložíme si jednotlivé barevné kanály obrazu. U každého z těchto barevných kanálů provedeme filtraci Sobelovým operátorem k nalezení hran. Jako referenční považujeme zelenou složku obrazu. Hrany budeme vyhledávat v této barevné složce. Nalezení hrany můžeme pozorovat na zderivovaném Obr. 9.2 po filtraci.



Obr. 9.2: Zderivovaný obraz zeleného kanálu po aplikaci Sobelova operátoru

Pro hrany je stanoven práh T_1 . Je nastaven na hodnotu $T_1=100$. Na následujícím obrázku jsou vidět pouze zřetelné hrany, které přesahují práh $T_1=100$.



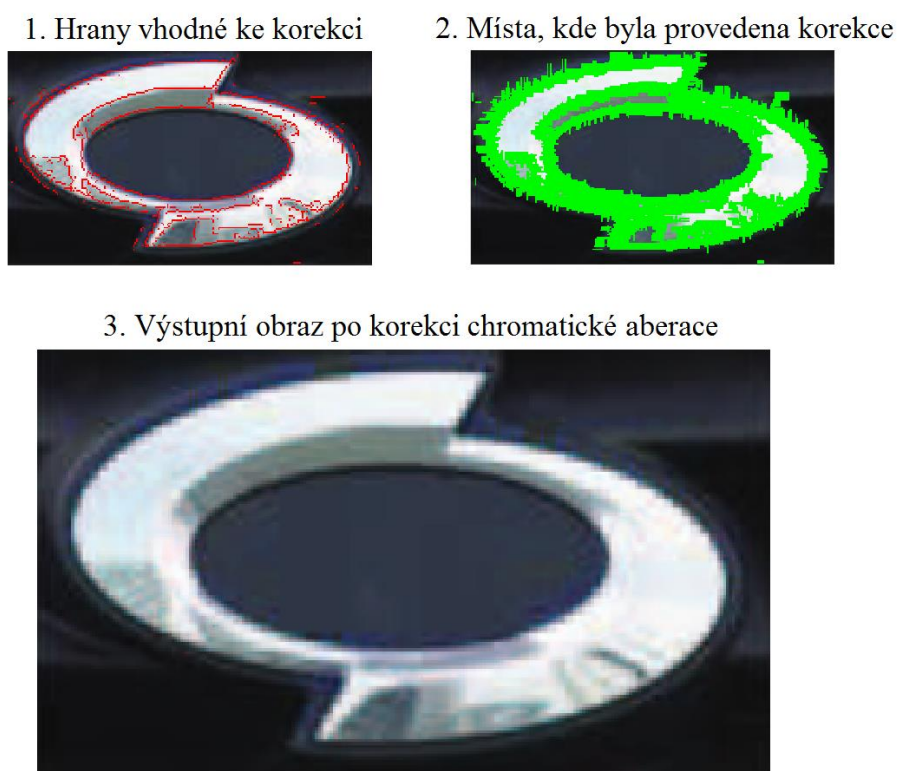
Obr. 9.3: Zderivovaný obraz zelené složky po naprahování

Program odstraňuje chromatickou aberaci v horizontálním a ve vertikálním směru. V horizontálním směru proto detekujeme svislé hrany, tedy hrany otočené o 45° až 135° . Detekce takových hran spočívá ve splnění podmínky, že pixel hrany v obraze filtrovaným Sobelovým operátorem v horizontálním směru má větší hodnotu, než pixel hrany z filtrace vertikálním směrem. Hrana je tedy výraznější v obraze filtrovaném horizontálně než v obraze filtrovaném vertikálně. Obdobně ve směru vertikálním detekujeme hrany otočené o úhel -45° až 45° . Korekce obrazu potom probíhá ve dvou fázích. První fází je korekce v horizontálním směru, druhou fází korekce ve vertikálním směru. Druhá fáze probíhá naprosto analogicky jako první fáze, proto podrobněji rozepíšeme pouze první fázi.

Nejdříve v horizontálním směru detekujeme hranu a ověříme její směr. Dále této hraně přiřadíme příslušné okolí, ve kterém provedeme korekci. Umístění levého a pravého pixelu tohoto okolí označíme l a r . V těchto místech hrana končí. Znamená to, že zde buď začíná homogenní oblast, nebo je ještě předtím detekována už nová hrana. Homogenní oblast je definována pomocí prahu T_2 . Práh T_2 je nastaven na hodnotu $T_2 = 40$. Oblasti ve zderivovaném obraze, ve kterých se hodnoty pohybují pod prahem T_2 , považujeme za homogenní. Oblasti v celém prostoru RGB ale označíme za homogenní jen tehdy, jsou-li podprahové hodnoty současně ve všech třech barevných kanálech. Nová hrana může být detekována ještě před výskytem oblasti. Potom je splněna druhá nerovnost v podmínce (9.4),(9.5). Nalezený interval označíme oblast Z . Této oblasti přiřadíme řádkový vektor p . Vektor p slouží k označení pixelů, u kterých bude provedena korekce chromatické aberace. Tyto pixely značí číslem 1, ostatní mají hodnotu 0. Korekce bude provedena u všech pixelů, které nevyhovují podmínce (9.1). To jsou všechny pixely v modrém a červeném kanálu, jejichž rozdíl od hodnoty příslušného pixelu v zeleném kanálu neleží uvnitř intervalu ohraničeného hodnotami rozdílových signálů D_S v krajních pozicích oblasti Z . Postupně procházíme oblastí Z a

korigujeme pixely, které jsou označené jako poškozené v proměnné p . Pro každý takový pixel vyhledáme jeho nejbližší nepoškozený pixel zleva a zprava. Na pozicích nepoškozených pixelů vypočítáme hodnoty rozdílových signálů od zelené složky. Vybereme tu menší hodnotu a přičteme k ní hodnotu pixelu ze zeleného kanálu, který se nachází na pozici poškozeného pixelu. Výsledek dosadíme do poškozeného pixelu příslušného barevného kanálu.

Poté pracujeme s obrazem ve vertikálním směru. Vše probíhá naprosto analogicky, jak jsem již popsál. Na Obr. 9.4 vidíme konečný výsledek korekce chromatické aberace.



Obr. 9.4: Detekce aberace a výsledek po korekci

Máme-li k dispozici původní zařízení, kterým můžeme pořídit snímky, nepomůže nám to ke korekci, protože metoda nevyužívá žádného modelu ke stanovení parametrů chyby. Rovněž z nasnímání téže scény nejsme schopni získat žádné užitečné informace.

9 ZÁVĚR

Zpracoval jsem diplomovou práci s názvem “Korekce snímků”. Jak již název napovídá, zabýval jsem se několika jevy, které se často negativně projevují při pořízení snímků. Jedná se o vinětaci, která vzniká především v důsledku hardwarového provedení přístroje, kompenzaci obrazů s podsvícením, které získáváme při focení proti zdroji osvětlení, například slunci, radiálním zkreslením v obraze, které se projevuje buď poduškovitým zkreslením, to je časté u širokoúhlých objektivů s dlouhou ohniskovou vzdáleností, nebo soudkovitým zkreslením, které se projevuje hlavně u širokoúhlých objektivů s krátkou ohniskovou vzdáleností. Dále jsem se zabýval korekcí chromatické aberace, což je vada, která tvoří barevnou linku podél kontrastních ploch, protože je způsobena ohniskovou vzdáleností čoček na vlnové délce světla. Také jsem implementoval program pro tvorbu panoramat ze sekvence snímků.

Při kompenzaci obrazů s podsvícením bývá objekt zájmu zatmaven a poškozenou oblast v obraze je poté třeba opravit s využitím výpočetní techniky. Ta si klade za cíl šetrným způsobem zvýšit v postižené části kontrast a také jas. Objekt zájmu bude potom zřetelnější. Algoritmus je navržen tak, aby fungoval na libovolném obraze a to zcela automaticky. Nevyžaduje zadání žádných parametrů. Vinětace je jev, kdy v důsledku nedokonalosti soustavy přístroje dochází ke snižování jasu od středu obrazu směrem ke krajům. Patrná bývá především v těsné blízkosti v okolí krajů. Někdy bývá do obrazu vinětace přidána i jako umělecký prvek, častěji se ale setkáváme s případy, kdy je tento jev nežádoucí, a proto se jej snažíme odstranit, aby obraz mohl být podroben další analýze při zpracování obrazu.

Radiální zkreslení se kompenzuje pomocí vzorce (8.1). Algoritmus hledá takové koeficienty, pro které bude dané zkreslení minimalizováno. Protože program využívá metodu přímých linií, je nutné ruční zadání linií, které by byly v nepoškozeném obraze přímé. Toto je kritérium pro správnou funkčnost obrazu. V případě, že bychom analyzovali snímek, na kterém nelze žádné přímé linie nalézt, což může být například dlaň ruky, nebo lidský obličej, nelze toto zkreslení odstranit, protože algoritmus nemá prostředky, jak tuto vadu zanalyzovat. Přímé linie by se měly také nalézat na krajích obrazu, protože zde je zkreslení nejzřetelnější. Do programu je nutné ručně zadat počet přímých linií a poté body, kterými tyto linie definujeme a zbytek korekce proběhne automaticky. K rovnání obrázku dojde na základě metody nejmenších čtverců. V případě, že bychom fotili s konstantní ohniskovou vzdáleností, bychom mohli pro daný fotoaparát a vzdálenost vypočítat předem korekční koeficienty a ty použít ke korekci libovolných snímků pořízených v libovolném prostředí, i bez přímých linií. Jako kalibr by posloužila pravoúhlá černobílá mřížka. Její narovnání do pravoúhlého tvaru by představovalo získání hledaných korekčních koeficientů.

Program na odstranění chromatické aberace pracuje s naprosto libovolným snímkem a to zcela automaticky. Jeho jádrem je podmínka chromatické korelace. Na neporušených hranách bez chromatické aberace existuje chromatická korelace, tzn. že změna v každém barevném kanálu probíhá na stejném místě. Všude tam, kde je

podmínka porušena, dojde ke korekci podle nejbližšího nepoškozeného pixelu. Program tedy nemá žádné specifické požadavky pro svoji funkčnost.

Mým dalším úkolem byla implementace programu pro tvorbu panoramat. Nejprve si přitom musíme poradit s nerovnoměrným osvětlením scény, kvůli kterému mívají sousední snímky velké rozdíly v barevných odstínech. Provedeme proto kompenzaci barev snímků s využitím koeficientů barevné korekce a globálního faktoru g pro zabránění saturace barev. Dále je třeba vyhledat v překrývající se části snímku a vznikajícího panoramatu cestu, která umožní nejšetrnější přechod mezi těmito dvěma obrazy. Cestu program hledá pomocí kumulovaného kvadrátu odchylek. Nakonec ještě v okolí této cesty zahladíme rozdíly v barvách překryvu podle vzorce (7.7). Po uživateli je přitom vyžadováno zadání bodů v překrývající části vznikajícího panoramatu a přidávaného snímku, které si navzájem odpovídají. Zbytek zajistí program automaticky. Program je navržen tak, aby fungoval v horizontálním směru.

Všechny algoritmy pro korekci vad fungují automaticky a nevyžadují zadávání parametrů, s výjimkou algoritmu pro korekci radiálního zkreslení, který vyžaduje manuální zadání přímých linií u rovných objektů vyhledaných ve zkresleném snímku. Je tedy třeba, aby daný snímek obsahoval rovné čáry. To se může jevit jako problém, máme-li pouze jediný snímek scény. Pak jsme odkázáni na to, co je ve snímku obsaženo a jak to bude uživatel interpretovat – zda bude považovat zakřivenou čáru za přímou linii u objektu, který nemusí být zrovna rovný. Nenažde-li zde přímé linie, nelze provést korekci radiálního zkreslení. U korekce obrazu s podsvícením je důležité, aby byla zatmavená oblast ve snímku dostatečně velká, potom bude na takovou vadu program vhodně reagovat.

Pokud budeme chtít detekovat barevnou vadu, je vhodné, aby ve snímku bylo rozhraní bílá-černá, a to nejlépe u kraje snímku, protože zde je ta vada největší. Radiální zkreslení je vhodné detekovat po celé ploše snímku. K tomu je vhodné, abychom měli možnost zjistit parametry zařízení, kterým snímky pořizujeme, mít je k dispozici, protože v daném snímku scény většinou nebývají přímé linie pravidelně rozmístěny po celé ploše. Například se ve střední části nacházejí keře a v horní obloha. Korekce je pak spočítána například jen pro spodní oblast snímku. Nemáme-li k dispozici originální snímek, ale můžeme nasnímat totéž, můžeme si ověřit, jak vypadá skutečnost. Původní snímek máme s čím srovnat. Můžeme například zjistit, že stromy mají nějaký charakteristický tvar a získat tak informaci ve střední části snímku a zpřesnit tak korekci. Můžeme-li nasnímat totéž, máme víc informací, což je výhodou. Máme-li k dispozici originální snímek, můžeme provést jeho kalibraci. Pro kalibraci je vhodné využít šachovnici. U šachovnice lze vidět jas na kraji i jas uprostřed – můžeme posoudit vinětači. Dále na ní lze vidět, jestli jsou hrany bílo-černé, nebo jsou tam barevné lemy. Můžeme posoudit chromatickou aberaci. Také je na ní patrné radiální zkreslení. Máme-li k dispozici originální snímek a pomocí něj můžeme pořídit snímky stejné scény, dokážeme získat maximum informací. Provedeme kalibraci snímku, nasnímáme scénu z různých úhlů, důkladně proměříme scénu. Následně můžeme porovnat výsledky z kalibrace snímku se skutečnými daty pořízenými nasnímáním scény.

Literatura

- [1] FISHER, Robert. *CVonline: Vision Geometry and Mathematics* [online]. Date of last change to this page: 07/01/2011[cit. 2012-03-19]. Dostupné z: <http://homepages.inf.ed.ac.uk/rbf/CVonline/geom.htm>
- [2] Co je korekce zkreslení?. [online]. Publikováno 19/03/2009 [cit. 2012-04-22]. Dostupné z: [https://nikoneurope-cz.custhelp.com/app/answers/detail/a_id/27167/~co-je-korekce-zkreslen%E3%AD%3F](https://nikoneurope-cz.custhelp.com/app/answers/detail/a_id/27167/~/co-je-korekce-zkreslen%E3%AD%3F)
- [3] *Odstranění geometrických zkreslení obrazu* [online]. [cit. 2012-04-15]. Dostupné z: http://mrl.cs.vsb.cz/people/gaura/dzo/geom_distortion_cz.pdf
- [4] HLAVÁČ, Václav. *Jasové a geometrické transformace* [online]. [cit. 2012-04-12]. Dostupné z: <http://cw.felk.cvut.cz/lib/exe/fetch.php/courses/a4m33dzo/18brightgeomtxc z.pdf>
- [5] ŽÁRA, Jiří, Bedřich BENEŠ a Petr FELKEL. *Moderní počítačová grafika*. 1. vyd. Brno: Computer Press, 1998, 448 s. ISBN 80-722-6049-9.
- [6] OWENS, Robyn. *Spatial domain methods* [online]. 10/29/1997[cit. 2012-04-22]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node3.html#SECTION00033000000000000000
- [7] FISHER, Robert, Simon PERKINS, Ashley WALKER a Erik WOLFART. Spatial Filters: Unsharp Filter. *HIPR2* [online]. 2003 [cit. 2012-04-12]. Dostupné z: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm#2>
- [8] WILHELM BURGER, Mark James Burge a Mark J. BURGE. *Digital image processing an algorithmic introduction using Java: An Algorithmic Introduction Using Java* [online]. 1st ed. New York: Springer, 2008 [cit. 2012-04-22]. ISBN 978-184-6289-682. Dostupné z: http://books.google.cz/books?id=4gBUz_IkkSsC&pg=PA130&dq=edge+sharpening+with+the+laplace+filter+image+function+can+also+be+estimated&hl=cs&sa=X&ei=OVCRT-SwAcbMsgatk4XEBA&ved=0CC8Q6AEwAA#v=onepage&q=edge%20sharpening%20with%20the%20laplace%20filter%20image%20function%20can%20also%20be%20estimated&f=false
- [9] FISHER, Robert, Simon PERKINS, Ashley WALKER a Erik WOLFART. Spatial Filters: Laplacian/Laplacian of Gaussian. *HIPR2* [online]. 2003 [cit. 2012-04-22]. Dostupné z: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>

- [10] Roberts Cross. *Wikipedia, the free encyclopedia* [online]. This page was last modified on 12 September 2011 [cit. 2012-04-22]. Dostupné z: http://en.wikipedia.org/wiki/Roberts_Cross
- [11] MAINI, Raman a Himanshu AGGARWAL. Study and Comparison of Various Image Edge Detection Techniques. *International Journal of Image Processing* [online]. s. 12, modified: 16.4. 2004 [cit. 2012-05-04]. Dostupné z: <http://www.cscjournals.org/csc/manuscript/Journals/IJIP/volume3/Issue1/IJI-P-15.pdf>
- [12] *Edge Detection Methods* [online]. [cit. 2012-04-22]. Dostupné z: http://www.robotics.technion.ac.il/courses/Advanced_Laboratory/Lab7/ARL_7_read.pdf
- [13] HLAVÁČ, Václav a Milan ŠONKA. *Počítačové vidění*. Praha: Grada, 1992, 272 s. ISBN 80-854-2467-3.
- [14] Rábová, Z., Hanáček, P., Peringer, P., Přikryl, P., Křena, B.: Užitečné rady pro psaní odborného textu [online]. Brno: c1993-2008, aktualizováno 2008-11-01 [cit. 2008-11-28]. Dostupné z: http://www.fit.vutbr.cz/info/statnice/psani_textu.html
- [15] Kolektiv autorů: *Pravidla českého pravopisu*. Academia, Praha, 1993. ISBN 80-200-0475-0
- [16] LU, Shin-Mao, Sheng-Fu LIANG a Chin-Teng LIN. A HVS-Directed Neural-Network-Based Approach for Salt-Pepper Impulse Noise Removal. *Journal of Information Science and Engineering* [online]. 2006, no.22, s. 925-939 [cit. 2012-12-23]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.1177&rep=rep1&type=pdf>
- [17] TSAI, Chun-Ming, Zong-Mu YEH a WANG. Contrast Compensation for Back-lit and Front-lit Color Face Image via Fuzzy Logic Classification and Image Illumination Analysis. *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics* [online]. 2008, č. 8, s. 3563-3568 [cit. 2012-12-26]. Dostupné z: http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDAQFjAA&url=http%3A%2F%2Fexcelsior.cs.ucsb.edu%2Fpapers%2Ficml2008.pdf&ei=3krlUIvmIIX6sgbQ24Fg&usg=AFQjCNF_8b7NZzubUdij7vHF9PDf86ouPw&bvm=bv.1355534169,d.d2k&cad=rja

- [18] SU, Mu-Chun, Jia-Hau GUO, Daw-Tung LIN a Guo Chung WANG. Algorithm for Color Backlight Images. In: [online]. 2002. vyd. [cit. 2012-12-28]. Dostupné z:
http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDYQFjAB&url=http%3A%2F%2Fwww.researchgate.net%2Fpublication%2F3949971_New_compensation_algorithm_for_color_backlight_images%2Ffile%2F9fcfd5085fb3d3c8e6.pdf&ei=gk3lUPS4LInFswbfz4DAAw&usg=AFQjCNGYmnsXkDZ8vu8h1SBpTnjzzhWobg&bvm=bv.1355534169,d.d2k
- [19] ZHENG, Yuanjie, Jingyi YU, Sing Bing KANG, Stephen LIN, Chandra KAMBHAMETTU. Single-Image Vignetting Correction Using Radial Gradient Symmetry. [online]. [cit. 2012-12-25]. Dostupné z:
http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&sqi=2&ved=0CC0QFjAA&url=http%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fum%2Fpeople%2Fstevlin%2Fpapers%2Fcvpr08zheng.pdf&ei=0U_IUP_4LoXjtQbc7YDgBg&usg=AFQjCNHytDIVlIAAU6NXtq45nbfrX08kOg&bvm=bv.1355534169,d.d2k
- [20] ZHENG, Yuanjie, Stephen LIN a Sing Bing KANG. *Single-Image Vignetting Correction* [online]. [cit. 2013-01-03]. Výzkum. Shanghai Jiaotong University, Microsoft Research Asia, Microsoft Research. Vedoucí práce Stephen Lin, Sing Bing Kang.
- [21] KONG, Haejung, Chanhoo JUNG, Wonjun KIM, Jaeho LEE a Changick KIM. Adaptive Image Backlight Compensation. *Applications of Digital Image Processing* [online]. 2010, no. 33, s. 1-7 [cit. 2012-12-29]. Dostupné z:
<http://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CDAQFjAA&url=http%3A%2F%2Fkoasas.kaist.ac.kr%2Fbitstream%2F10203%2F24825%2F1%2FPSI77981P.pdf&ei=Xl3lUKvNKonAswaKqYCYCw&usg=AFQjCNE9mnsMI51OSzsjBV9kN2u4OeALtA&bvm=bv.1355534169,d.Yms>
- [22] *Wide Angle Underwater Photography: Equipment For Wide Angle Underwater Photography*. Dive Photo Guide [online]. [cit. 2014-09-01]. Dostupné z:
<http://www.divephotoguide.com/underwater-photography-techniques/article/equipment-wide-angle-underwater/>
- [23] TARUN, Orly. *Correcting Geometric Distortions*. Pixel of Knowledge [online]. [10 September 2009] [cit. 2014-09-01]. Dostupné z:
<http://otarun.blogspot.cz/2009/09/activity-13-correcting-geometric.html>
- [24] SOON-WOOK CHUNG, BYOUNG-KWANG KIM a WOO-JIN SONG. *Detecting and eliminating chromatic aberration in digital images*. In: 2009 16th IEEE

- International Conference on Image Processing (ICIP) [online]. IEEE, 2009, s. 3905-3908 [cit. 2014-09-01]. ISBN 978-1-4244-5653-6. DOI: 10.1109/ICIP.2009.5413971. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5413971>
- [25] XIONG, Yingen a Kari PULLI. Fast panorama stitching for high-quality panoramic images on mobile phones. In: IEEE Transactions on Consumer Electronics [online]. 2010, s. 298-306 [cit. 2014-09-01]. ISSN 0098-3063. DOI: 10.1109/TCE.2010.5505931. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5505931>
- [26] YINGEN XIONG a K. PULLI. Fast panorama stitching on mobile devices. In: 2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE) [online]. IEEE, 2010, s. 319-320 [cit. 2014-09-01]. ISBN 978-1-4244-4314-7. DOI: 10.1109/ICCE.2010.5419027. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5419027>
- [27] CHOI, Kai San, Edmund Y. LAM, Kenneth K. Y. WONG, Nitin SAMPAT, Jeffrey M. DICARLO a Russel A. MARTIN. . In: [online]. 60690J-60690J-8 [cit. 2014-09-01]. DOI: 10.1117/12.649775. Dostupné z: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=729163>
- [28] XIONG, Yingen a Kari PULLI. Sequential image stitching for mobile panoramas. In: 2009 7th International Conference on Information, Communications and Signal Processing (ICICS) [online]. IEEE, 2009, s. 1-5 [cit. 2014-09-01]. ISBN 978-1-4244-4656-8. DOI: 10.1109/ICICS.2009.5397590. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5397590>
- [29] HLAVÁČ, Václav. CENTRUM STROJOVÉHO VNÍMÁNÍ ČVUT. Praktická geometrická optika. [J]20 August 2013. Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/>
- [30] GAURA, Jan. VSB - TECHNICAL UNIVERSITY OF OSTRAVA. Odstranění geometrických zkreslení obrazu. I. 11. 2011. Dostupné z: <http://mrl.cs.vsb.cz/people/gaura/dzo/>
- [31] MAŘÍK, Robert. MENDELOVA UNIVERZITA V BRNĚ, Lesnická a dřevařská fakulta. Metoda nejmenších čtverců. 23. 4. 2009. Dostupné z: <http://user.mendelu.cz/marik/wiki/aplikace/mnc-cz.pdf>

- [32] XIONG, Yingen a Kari PULLI. *Fast image stitching and editing for panorama painting on mobile phones*. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops [online]. IEEE, 2010, s. 47-52 [cit. 2014-09-01]. ISBN 978-1-4244-7029-7. DOI: 10.1109/CVPRW.2010.5543259. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5543259>

- [33] XIONG, Yingen a Kari PULLI. *Fast panorama stitching for high-quality panoramic images on mobile phones*. In: IEEE Transactions on Consumer Electronics [online]. 2010, s. 298-306 [cit. 2014-09-01]. ISSN 0098-3063. DOI: 10.1109/TCE.2010.5505931. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5505931>

- [34] *Itálie-Malta-Web.cz* [online]. 23. 1. 2014 [cit. 2014-09-01]. Dostupné z: <http://www.italie-malta-web.cz/clanky>

- [35] *Co je fotografie ve formátu HDR?*. [online]. 2009-06-16 [cit. 2015-04-28]. Dostupné z: https://nikoneurope-cz.custhelp.com/app/answers/detail/a_id/27522

Seznam příloh

Příloha 1. CD s kódem programu