



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

**VIZUALIZACE DIGITÁLNÍCH OBRAZŮ S VYSOKÝM
DYNAMICKÝM ROZSAHEM**

HIGH DYNAMIC RANGE IMAGES VISUALIZATION METHODS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ALEKSANDER MARKOVSKY

VEDOUCÍ PRÁCE

SUPERVISOR

prof. RNDr. MILOSLAV DRUCKMÜLLER, CSc.

BRNO 2021

Zadání bakalářské práce

Ústav: Ústav matematiky
Student: **Aleksander Markovsky**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Matematické inženýrství
Vedoucí práce: **prof. RNDr. Miloslav Druckmüller, CSc.**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Vizualizace digitálních obrazů s vysokým dynamickým rozsahem

Stručná charakteristika problematiky úkolu:

Digitální obrazy s dynamickým rozsahem větším než 8 bit/pixel mohou obsahovat detaily neviditelné pro lidský zrak. Proto je nutné použít matematické vizualizační techniky známé pod zkratkou HDR. Matematický popis těchto metod je prvním úkolem práce. Druhým úkolem je implementace HDR metody typu neostrá maska, ve které jsou masky vytvořeny za použití konvoluce s jádrem, které je lineární kombinací Gaussovských jader.

Cíle bakalářské práce:

Popsat matematickou teorii HDR metod pro vizualizaci digitálních obrazů a vytvořit program pro HDR metodu založenou na neostré masce.

Seznam doporučené literatury:

PRATT, W. Digital Image Processing. 4th Edition. A John Wiley & Sons, Inc., Publication, 2007.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zabývá zpracováním obrazu s vysokým dynamickým rozsahem. Popisuje postup kalibrace obrazu s cílem snížit šum, Fourierovu transformaci a konvoluci obrazu, lineární filtry a metodu neostré masky. Následně v praktické části jsou tyto poznatky zúžitkovány v C++ programu, který zpracovává HDR obrazy a vytváří .jpg výstup.

Summary

This thesis describes image processing of pictures with high dynamic range. The first part aims to describe ways to reduce visible noise, Fourier transform and convolution for image processing, linear filtering and unsharp mask method. Theoretical framework is followed by a C++ program, that processes HDR images, enhances details and creates a .jpg output.

Klíčová slova

HDR, vysoký dynamický rozsah, zpracování obrazu, Fourierova transformace, konvoluce, neostrá maska, lineární filtry, Gaussovské jádro, C++, OpenCV, DeepSkyStacker, light frame, dark frame, bias frame, flat field, redukce šumu

Keywords

HDR, high dynamic range, image processing, Fourier transform, convolution, unsharp mask, linear filtering, Gaussian kernel, C++, OpenCV, DeepSkyStacker, light frame, dark frame, bias frame, flat field, noise reduction

MARKOVSKY, A. *Vizualizace digitálních obrazů s vysokým dynamickým rozsahem*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2021. 36 s. Vedoucí prof. RNDr. Miloslav Druckmüller, CSc.

Prohlašuji, že jsem svou bakalářskou práci na téma *Vizualizace digitálních obrazů s vysokým dynamickým rozsahem* zpracoval samostatně pod vedením prof. RNDr. Miloslava Druckmüllera, CSc. Odbornou literaturu, kterou jsem při zpracování využíval, v práci cituji a uvádím v seznamu literatury na konci práce.

Aleksander Markovsky

Chci vyjádřit velký vděk panu profesorovi Miloslavu Druckmüllerovi za veškerý čas věnovaný vedení této práce, know-how a inspiraci. Také bych rád poděkoval příteli Jindřichu Kosíkovi za rady a podporu během studia.

Aleksander Markovsky

Obsah

1	Úvod	2
2	Popis úlohy	3
3	Obraz a jeho kalibrace	5
3.1	Obraz	5
3.2	Kalibrace	6
3.2.1	Light frame	6
3.2.2	Dark frame	7
3.2.3	Bias frame	7
3.2.4	Flat field, dark flat	8
3.2.5	Skládání kalibračních snímků	9
4	Fourierova transformace a konvoluce	11
4.1	Komplexní funkce a její integral	11
4.2	Spojité Fourierova transformace	11
4.2.1	Fourierova transformace v $\mathcal{L}(\mathbb{R})$	11
4.2.2	Fourierova transformace v $\mathcal{L}(\mathbb{R}^2)$	12
4.2.3	Konvoluce	13
4.3	Diskrétní Fourierova transformace	13
4.3.1	Diskrétní konvoluce	15
4.4	Rychlá Fourierova transformace	17
5	Lineární filtry	20
6	Praktická část	24
6.1	Kalibrace pomocí Deep Sky Stacker	25
6.2	Vlastní implementace	26
6.2.1	OpenCV	26
6.2.2	Vlastní funkce	27
6.2.3	Ukázka výsledků	28
7	Závěr	33
	Literatura	35
8	Seznam příloh	36

1. Úvod

Snímací schopnosti moderních fotoaparátů a videokamer převyšují zobrazovací schopnosti současných monitorů. Typický displej je schopný zobrazit 8 bitů na subpixelu, kdežto běžné poloprofesionální fotoaparáty zachytí obraz s 14 bitovou hloubkou, některými postupy je dokonce možné vytvořit zdrojová data s hloubkou 32 nebo i více bitů. Tyto obrazy s vysokým dynamickým rozsahem označujeme jako HDR (High Dynamic Range).

Cílem této práce je prezentovat jeden přístup, jak transformovat HDR obraz do zobrazitelné podoby a zejména jak zobrazit co nejvíce detailů ze zdrojových dat. V teoretické části bude popsáno získávání zdrojových fotografií, jejich kalibrace a následně matematický aparát Fourierovy transformace a konvoluce, který nám umožní provádět úpravy obrazu. V praktické části bude popsáno získávání raw fotografií z běžného fotoaparátu, kalibrace v DeepSkyStackeru a konečně, jak lze využít knihovny OpenCV v C++ pro zpracování 32bitového obrazu.

Definice, věty a důkazy jsou převzaty z [4], pokud nebude uvedeno jinak.

2. Popis úlohy

Kde se setkáme s obrazy s vysokým dynamickým rozsahem a proč bychom se jimi měli zabývat? Vzorovým příkladem je fotografie sluneční koróny. Obraz sluneční koróny má obrovský kontrast, je snadné si to představit, pokud si uvědomíme, že v popředí je zdroj světla pro celou sluneční soustavu a v pozadí je temné prázdno vesmíru. Prvním problémem je takový obraz vůbec zachytit, potom druhým problémem je, jak taková data zobrazit, aby byly struktury koróny pozorovatelné okem. Abychom pochopili přínos HDR, není potřeba chodit až k fotografii koróny, stačí se podívat z okna. Vytáhneme-li telefon a namíříme jeho kameru na okno za jasného dne, zpozorujeme, že pokud chceme mít okno korektně exponované, místnost je zůstane zahalená tmou. Naopak chceme-li zachytit detaily místnosti, světlé okno se stane téměř bílou plochou.



(a) Viditelné detaily v místnosti



(b) Detaily v okně, místnost zahalená tmou



(c) Sjednocení snímků tonemappingem

Obrázek 2.1: Tmavá místnost s výhledem do světlého okna. Foto Axel Jacobs [9].

Jiným příkladem může být fotografie za nočního osvětlení. Naše oko vidí detaily osvětlené budovy i ve stínech pod ní, ovšem na displeji fotoaparátu vidíme, že si musíme vybrat, jaké detaily zůstanou zachovány. Toto dilema řeší techniky HDR. Běžný postup v umělecké fotografii je zachytit několik snímků s různou expozicí, zaměřenou pokaždé na detaily v jiné oblasti a následně tyto snímky spojit pomocí tonemappingu.



(a) Korektně exponovaná světla



(b) Korektně exponované stíny



(c) Sjednacení snímků s lokálním tonemappingem, nepřirozený výsledek

Obrázek 2.2: The Jefferson National Expansion Memorial, including the Gateway Arch and Old Courthouse, in St Louis, MO, USA. Foto Kevin McCoy [11].

Tato metoda může fungovat dobře, ovšem velmi často vypadá výsledek nepřirozeně, přechody mezi světly a stíny mají kolem sebe halo a obraz může působit dojmem, jako kdyby byl namalovaný. Cílem této práce bude demonstrovat jiný přístup k vizualizaci obrazů s vysokým dynamickým rozsahem.

3. Obraz a jeho kalibrace

Před zpracováním obrazu je potřeba získat kvalitní zdrojová data. v této kapitole si zavedeme pojem obrazu a vysvětlíme postup jeho kalibrace.

3.1. Obraz

Definice 3.1 (Digitální obraz v odstínech šedi). Necht' je $R = \{0, 1, \dots, M - 1\} \times \{0, 1, \dots, N - 1\}$ $M, N \in \mathbb{N}$ a necht' $W = \{0, 1, \dots, w - 1\}$, $w \in \mathbb{N}$. Pak

$$f(x, y) : R \rightarrow W$$

se nazývá *digitální obraz v odstínech šedi* nebo pouze *obraz*. Prvky množiny R se nazývají *pixely*, hodnoty funkce $f(x, y)$ se nazývají *hodnoty pixelu*. Různé datové typy zabírají n bitů v paměti, volba datového typu pro reprezentaci pixelu určuje jeho *n bitovou hloubku*. Hodnota w určuje *dynamický rozsah*. Můžeme tedy říct, že obraz má *n bitový dynamický rozsah*, pokud $w = 2^n$.

Definice 3.2 (Barevný digitální obraz). *Barevný digitální obraz* je trojice digitálních obrazů v odstínech šedi (r,g,b), kterým říkáme *červený, zelený a modrý kanál* respektive (z anglického red, green, blue). Tuto specifickou reprezentaci budeme také někdy označovat jako *RGB obraz*. Pixely jednotlivých obrazů (r,g,b) označujeme *subpixely*.

Poznámka 3.1. Reprezentací barevného obrazu existuje několik, například HSL (hue, saturation, luminance), nebo CMYK(cyan, magenta, yellow, key). Pro naše potřeby budeme však využívat RGB reprezentaci, v praktické části se zmíníme o HSL reprezentaci.

Poznámka 3.2. Bavíme-li se o bitové hloubce barevného, mohou nastat nejasnosti v terminologii. V některých případech bitová hloubka obrazu označuje celkový počet bitů dostupný pro reprezentaci barev, například 16bitový obraz alokující 5 bitů pro červený kanál, 6 bitů pro zelený kanál a 5 bitů pro modrý kanál. v této práci bude pojem *bitová hloubka* označovat počet bitů na kanál, tedy *8bitový RGB obraz* znamená obraz, který má 8 bitů pro každý barevný kanál, dohromady tedy 24 bitů na jeden pixel.

Fotografie tedy není nic jiného, než matice o rozměrech $M \times N$, jejíž prvky (pixely) nabývají jedné z w možných hodnot. Běžný monitor je schopný zobrazit 8bitový dynamický rozsah, tedy každý z RGB kanálů nabývá jedné z $2^8 = 256$ hodnot, čímž je schopný zobrazit $256^3 = 16777216$ barev. Lidské oko však je schopné vidět kontrast až 1000:1 odstínů šedi [13]. Proto podstatné zachycovat co největší množství dat. Pokud má obraz bitovou hloubku větší než 8 bitů, tak jej lze nazvat obrazem s vysokým dynamickým rozsahem, neboli HDRI (High Dynamic Range Image), naopak obraz s 8bitovou hloubkou a nižší lze nazvat obrazy s LDR (Low Dynamic Range). Pokud zachycujeme kontrastní scénu s velkým množstvím detailů ve stínech i ve světlech, 8bitová hloubka není dostačující. To je důvod, proč většina moderních fotoaparátů je schopná ukládat data do formátů s 12-14bitovou hloubkou. Pro tyto účely slouží rodina raw formátů, která uchovávají radiometrická data a umožňují větší flexibilitu z hlediska následné manipulace. Konkrétní raw formát se liší v závislosti na výrobci fotoaparátů, příkladem je .arw pro Sony, .nef pro Nikon atp.

3.2. Kalibrace

Před samotným zpracováním obrazu je třeba získat co nejkvalitnější zdrojová data. Podstatným omezením fotočipů je vznikající šum, jehož vliv chceme co nejvíce omezit, cílem je tedy maximalizovat tzv. SNR (Signal to Noise Ratio), poměr signálu a šumu. Dále je potřeba co nejvíce potlačit vliv nedostatků optického aparátu. Mezi relativně snadno odstranitelné vady optického aparátu patří prach, nečistoty, škrábance a vinětae. Dalšími vadami jsou například chromatická aberace, odlesky a různé geometrické vady, těmito se ovšem zabývat nebudeme. Omezení některých optických vad a zvýšení SNR lze zajistit zachycením více zdrojových obrázků a jejich vhodným průměrováním. Tohoto postupu se využívá zejména (ale nejenom) v astrofotografii, neboť se téměř vždy fotí za nízkého světla, s dlouhými časy a vysokou citlivostí ISO, což způsobuje výrazný výskyt šumu. Kalibrační snímky se dělí na tzv. light frame, dark frame, bias frame, flat field a popřípadě dark flat.

3.2.1. Light frame

Light frame je samotný snímek scény při požadované expozici. Pokud se tedy ze stativu vyfotí série expozic se stejným nastavením a vytvoří se aritmetický průměr z nich, podaří se zredukovat náhodný šum, jak je patrné z obrázků.



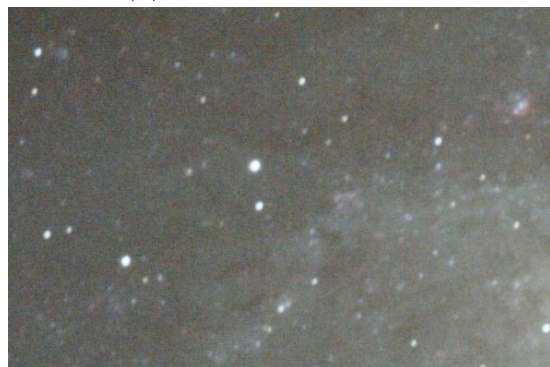
(a) jeden snímek



(b) průměr dvou snímků



(c) průměr čtyř snímků

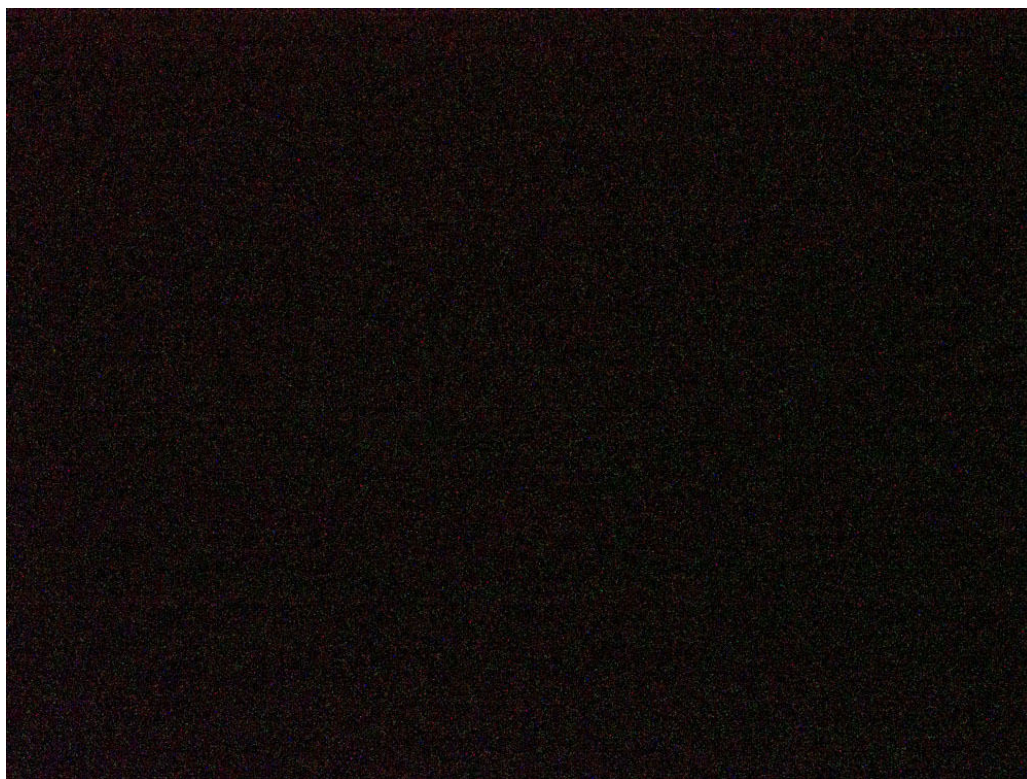


(d) průměr šestnácti snímků

Obrázek 3.1: Redukce šumu průměrováním více expozicí [5].

3.2.2. Dark frame

Když na snímač dopadnou fotony, tak jsou v polovodiči generovány elektrony fotoelektrickým jevem[18] a ty jsou snímačem zaznamenány jako signál. K generaci elektronů může však dojít i náhodně působením tepla[18]. Tak vzniká temný proud (tzv. dark current), který je formou šumu. Vliv tohoto šumu lze omezit pomocí tzv. *dark frame*. Ten se získá vyfocením snímku v úplné tmě s krytkou na objektivu. Ty se následně zprůměrují, čímž vznikne master dark frame. Pro průměrování je vhodné použít medián, neboť zejména při dlouhých expozicích může dojít k saturaci vlivem kosmického záření. Od každého light frame se odečte¹ master dark frame, čímž omezíme pravidelný termální šum. Je důležité, aby měl fotoaparát při získávání dark frameů stejné nastavení, jako u light frameů. Na vznik temného proudu má vliv ISO citlivost snímače, délka expozice a dokonce i okolní teplota. Z tohoto důvodu se při focení dlouhých expozic používají chlazené snímače pro dosažení optimálních výsledků. Pokud není k dispozici fotoaparát s chlazeným snímačem, je vhodné vytvořit dark frame ihned po skočení sběru light frame, aby se příliš nezměnila teplota snímače. Na obrázku 3.2 lze vidět termální šum, ale také vadné pixely.



Obrázek 3.2: Dark frame - zvýšená expozice pro zvýraznění šumu. Foto Andrea Minoia [14].

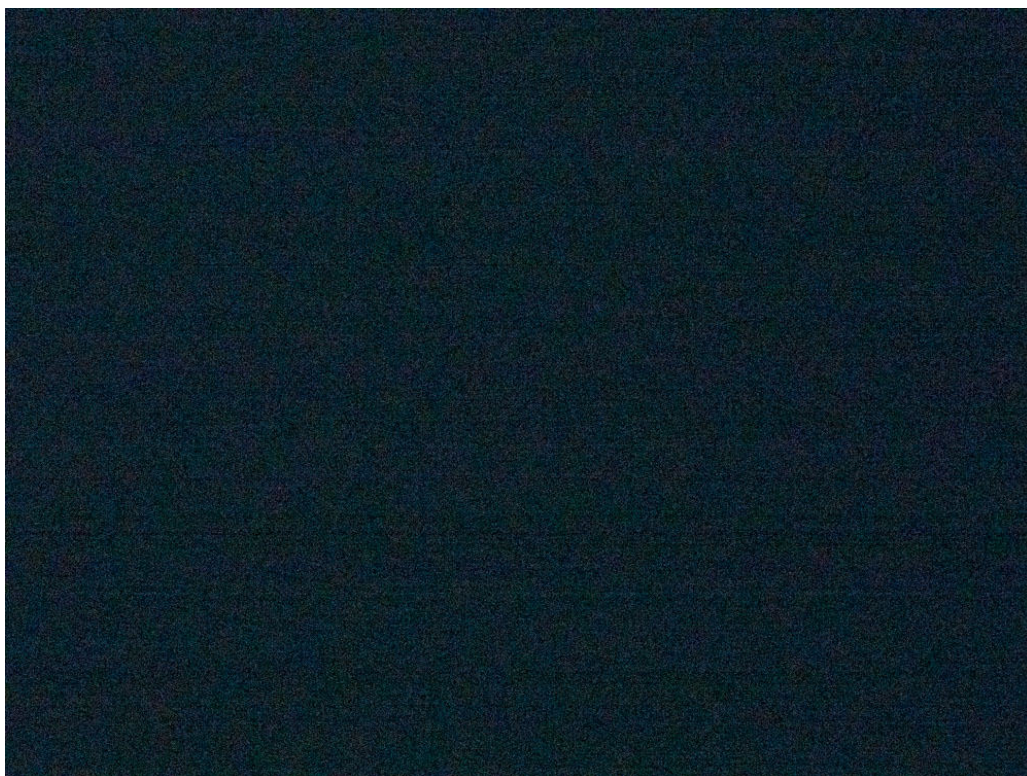
3.2.3. Bias frame

Kvůli tomu, že každý pixel podléhá náhodnému šumu, jsou čipy předpjeté. Tedy kdyby byla zaznamenána expozice bez jediného fotonu a bez temného proudu, čip bude i přesto

¹Odčítáním obrazů se myslí odečtení hodnot pixelů se stejnou pozicí, tedy po prvcích matice. Obdobně pro sčítání a dělení obrazů

3.2. KALIBRACE

generovat proud, tzv. bias current. Aby bylo možné aplikovat lineární transformaci na obraz, je potřeba odečíst toto předpětí. K tomu slouží *bias frame*. Ideální bias frame má nulový expoziční čas, prakticky se vyfotí na nejkratší možný čas, s krytkou na objektivu a ve tmě, aby se zabránilo dopadu fotonů na senzor a vlivu temného proudu. I přes tyto podmínky je bias frame stále ovlivněn náhodným šumem, proto je potřeba jich zaznamenat několik a vzít opět jejich medián, aby se bylo možné se co nejvíce přiblížit skutečné hodnotě předpětí. Tato hodnota závisí na citlivosti ISO a do jisté míry i na okolní teplotě, je proto nutné zachytit bias frame se stejným ISO a nejlépe i za stejných okolních podmínek, jako light frame. Průměrováním tedy vznikne master bias frame, který se odečítá od všech ostatních snímků. Odečtení tohoto předpětí od master dark frame umožňuje škálovatelnost dark frame. Výhoda je, že bias snímky není potřeba zachycovat při každém focení, je možné mít jejich knihovnu pro různé citlivosti ISO a aktualizovat tuto knihovnu průběžně, aby se kompenzovalo opotřebení čipu.



Obrázek 3.3: Bias frame - zvýšená expozice pro zvýraznění šumu. Foto Andrea Minoia [14].

3.2.4. Flat field, dark flat

Optický aparát podléhá vinětaci a při sebevětší opatrnosti není možné odstranit všechny nečistoty a prach z optiky a čipu fotoaparátů. Reálný fotočip je produktem výroby a tedy podléhá odchylkám, velikost a citlivost jednotlivých pixelů není stejná. Všechny tyto nedostatky pomáhá minimalizovat flat field korekce. *Flat field* se pořídí vyfocením rovnoměrně osvětlené uniformní plochy, například plátna, oblohy za soumraku, či svítání, nebo jej lze dokonce i pořídít zakrytím objektivu bílým tričkem a namířením na zdroj světla. Je důležité tento kalibrační snímek pořídít se stejným optickým nastavením, jako light frame, tedy se stejnou clonou a ostřicí vzdáleností, neboť tyto proměnné mají vliv na vinětaci.

Také je vhodné zanechat stejné nastavení ISO citlivosti, aby bylo možné správně korigovat rozdíly citlivosti jednotlivých pixelů (ISO má vliv na zisk čipu). Opět je nutné pořídit několik takových snímků a uvažovat jejich medián, pro každý jednotlivý snímek udělat korekci o bias a také pro flat field vytvořit dark frame, nazývaný zkráceně *dark flat*. Po těchto korekcích vznikne master flat field. Korekce se provede vydělením light frame flat fieldem. Na obrázku 3.4 je viditelná nežádoucí vinětace [14].



Obrázek 3.4: Flat frame. Foto Andrea Minoia [14].

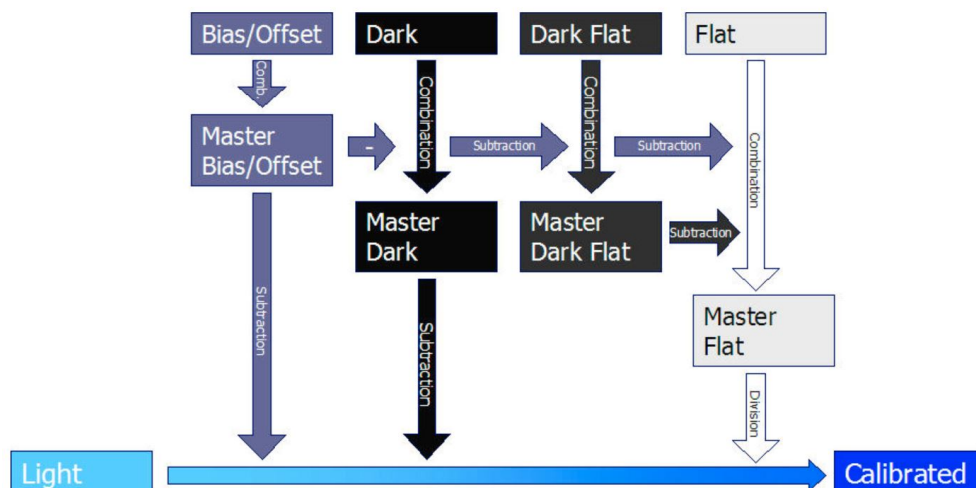
3.2.5. Skládání kalibračních snímků

Postup kalibrace je tedy následující. Ze všech bias framů se mediánem vytvoří master bias frame. Od každého light frame, dark frame, flat field i dark flat je potřeba tento bias odečíst. Poté se vezme medián z dark frame očištěných o bias, vznikne tak master dark frame. Obdobně se vytvoří medián z flat fieldů a dark flatů, následně se takto vzniklý průměr dark flatů odečte od průměru flat fieldů, získá se tím master flat. Od každého light frame se odečte master dark a potom se vydělí master flatem. Nakonec aritmetickým průměrem takto očištěných light frame se získá výsledný zkalibrovaný snímek.

Není vždy nutné mít všechny kalibrační snímky. Například pokud není k dispozici master bias, kalibraci lze stále provést neboť se předpětí v master dark frame odečte od předpětí v light frame, obdobně při vytváření master flat fieldu. Zvýšení SNR je možné pouze pokud jsou kalibrační snímky kvalitní. Je tedy potřeba vytvořit dostatečné množství každého druhu snímků. Pokud bychom uvažovali pouze jeden light frame a odečetli od něj jeden dark frame, zdrojový snímek bychom spíše poškodili a šum zdvojnásobili. SNR snímku stoupá s druhou odmocninou počtu snímků, platí tedy pravidlo „čím více, tím lépe“. Není nutné, aby počet light frame odpovídal počtu dark frame a podobně. Konkrétní počty snímků určuje praxe a náročnost scény.

3.2. KALIBRACE

Full Calibration Process (all files available)



Obrázek 3.5: Schéma kalibračního procesu se všemi dostupnými kalibračními snímky dle [5].

Nakonec je nutné dodat, že tento postup kalibrace není vhodný pro všechny kamery. Některé fotoaparáty mají již zabudované odečítání předpětí a kalibraci vůči temnému proudu, v těchto případech by odečítání bias frame a dark frame pouze přidalo šum. Je proto nutné upravit kalibrační proces pro konkrétní techniku.

4. Fourierova transformace a konvoluce

V minulé kapitole jsme definovali obraz jako nějakou funkci. Z teorie Fourierových řad víme, že některé funkce dokážeme aproximovat řadou vln. Navíc je skutečný obraz složený pouze z diskretních záznamů, jsme proto schopni každý obraz rozdělit na vlny. V této kapitole si popíšeme základní aparát, který nám následně umožní obraz filtrovat.

4.1. Komplexní funkce a její integrál

Definice 4.1. Necht $f(x)$ je *komplexní funkce reálné proměnné* $f : M \rightarrow \mathbb{C}$, $M \subseteq \mathbb{R}$, pak

$$f(x) = u(x) + iv(x),$$

kde $u, v : M \rightarrow \mathbb{R}$ jsou reálné funkce reálné proměnné.

Definice 4.2. Necht $a, b \in \mathbb{R}$, $a < b$ a necht u, v jsou integrovatelné v Riemannově smyslu na intervalu na $\langle a, b \rangle$. Potom řekneme, že je f *integrovatelná* na $\langle a, b \rangle$ a platí

$$\int_a^b f(x) dx = \int_a^b u(x) dx + i \int_a^b v(x) dx.$$

Obdobně definujeme primitivní funkci k f jako

$$\int f(x) dx = \int u(x) dx + i \int v(x) dx.$$

Definice 4.3 ($\mathcal{L}(\mathbb{R})$). Nazveme $\mathcal{L}(\mathbb{R})$ prostorem všech funkcí $f : \mathbb{R} \rightarrow \mathbb{C}$ takových, že pro všechny f integrál

$$\int_{-\infty}^{\infty} |f(x)| dx$$

existuje a je konečný.

Definice 4.4 ($\mathcal{L}(\mathbb{R}^2)$). Nazveme $\mathcal{L}(\mathbb{R}^2)$ prostorem všech funkcí $f : \mathbb{R}^2 \rightarrow \mathbb{C}$ takových, že pro všechny f integrál

$$\iint_{\mathbb{R}^2} |f(x, y)| dx dy$$

existuje a je konečný.

4.2. Spojitá Fourierova transformace

4.2.1. Fourierova transformace v $\mathcal{L}(\mathbb{R})$

Definice 4.5 (Fourierova transformace funkcí v $\mathcal{L}(\mathbb{R})$). Necht je $f(x) \in \mathcal{L}(\mathbb{R})$. *Fourierova transformace* funkce f je funkce $\mathcal{F}\{f\}(\xi) = F(\xi) : \mathbb{R} \rightarrow \mathbb{C}$ definovaná jako

$$F(\xi) = \int_{-\infty}^{\infty} f(x) e^{-ix\xi} dx.$$

4.2. SPOJITÁ FOURIEROVA TRANSFORMACE

Funkce F se také nazývá *Fourierovo spektrum* funkce f .

Poznámka 4.1. Fourierova transformace funkcí z $\mathcal{L}(\mathbb{R}^2)$ existuje a je ohraničená. Důkaz k nalezení v [1].

Definice 4.6 (Inverzní Fourierova transformace funkcí v $\mathcal{L}(\mathbb{R})$). Necht je $G(\xi) \in \mathcal{L}(\mathbb{R})$. *Inverzní Fourierova transformace* funkce G je funkce $\mathcal{F}^{-1}\{G\}(x) = g(x) : \mathbb{R} \rightarrow \mathbb{C}$ definovaná jako

$$g(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\xi) e^{ix\xi} d\xi.$$

Věta 4.1 (O Fourierově transformaci funkcí z $\mathcal{L}(\mathbb{R})$). Necht je funkce $f(x)$ z $\mathcal{L}(\mathbb{R})$ a je všude spojitá, pak platí

$$\mathcal{F}^{-1}\{\mathcal{F}\{f(x)\}\} = f(x).$$

Poznámka 4.2. Vynecháme-li požadavek spojitosti, věta 4.1 obecně neplatí. Důvodem je, že u nespojité funkce integrál nenávratně vyhladí ony nespojitosti. Důkaz této věty a ilustraci nutnosti spojitosti na příkladu lze nalézt v [4].

4.2.2. Fourierova transformace v $\mathcal{L}(\mathbb{R}^2)$

Definice 4.7 (Fourierova transformace funkcí v $\mathcal{L}(\mathbb{R}^2)$). Necht je $f(x, y) \in \mathcal{L}(\mathbb{R}^2)$. *Fourierova transformace* funkce f je funkce $\mathcal{F}\{f\}(\xi, \eta) = F(\xi, \eta) : \mathbb{R}^2 \rightarrow \mathbb{C}$ definovaná jako

$$F(\xi, \eta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i(x\xi + y\eta)} dx dy.$$

Funkce F se také nazývá *Fourierovo spektrum* funkce f .

Definice 4.8 (Inverzní Fourierova transformace funkcí v $\mathcal{L}(\mathbb{R}^2)$). Necht je $G(\xi, \eta) \in \mathcal{L}(\mathbb{R}^2)$. *Inverzní Fourierova transformace* funkce G je funkce $\mathcal{F}^{-1}\{G\}(x, y) = g(x, y) : \mathbb{R}^2 \rightarrow \mathbb{C}$ definovaná jako

$$g(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(\xi, \eta) e^{i(x\xi + y\eta)} d\xi d\eta.$$

Věta 4.2 (O Fourierově transformaci funkcí z $\mathcal{L}(\mathbb{R}^2)$). Necht je funkce $f(x, y)$ z $\mathcal{L}(\mathbb{R}^2)$ a je všude spojitá, pak platí

$$\mathcal{F}^{-1}\{\mathcal{F}\{f(x, y)\}\} = f(x, y).$$

Poznámka 4.3. Analogicky s 4.2 je klíčová podmínka spojitosti. Důkaz lze nalézt v [16].

4.2.3. Konvoluce

Definice 4.9 (Konvoluce). Necht $f_1(x, y), f_2(x, y) \in \mathcal{L}(\mathbb{R}^2)$. Konvoluce $f_1 * f_2$ funkcí $f_1(x, y), f_2(x, y)$ je funkce

$$f_1 * f_2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(s, t) f_2(x - s, y - t) ds dt.$$

Věta 4.3. Necht $f_1(x, y), f_2(x, y) \in \mathcal{L}(\mathbb{R}^2)$, pak $f_1 * f_2 \in \mathcal{L}(\mathbb{R}^2)$.

Věta 4.4 (Konvoluční teorém). Necht $f_1(x, y), f_2(x, y) \in \mathcal{L}(\mathbb{R}^2)$ mají Fourierovy spektra $F_1(\xi, \eta), F_2(\xi, \eta)$. Potom platí

$$\mathcal{F}\{f_1 * f_2\} = F_1 \cdot F_2.$$

Věta 4.5. Necht $f_1(x, y), f_2(x, y) \in \mathcal{L}(\mathbb{R}^2)$ mají Fourierovy spektra $F_1(\xi, \eta), F_2(\xi, \eta)$ a f_1, f_2 jsou spojité. Potom platí

$$\mathcal{F}\{f_1 \cdot f_2\} = \frac{1}{4\pi^2} F_1 * F_2.$$

Poznámka 4.4. Důkazy k výše uvedeným větám k nalezení v [4].

4.3. Diskrétní Fourierova transformace

Definice 4.10 (Diskrétní Fourierova transformace). Necht $f(x, y)$ je funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$. Diskrétní Fourierova transformace funkce $f(x, y)$ je funkce $\mathcal{D}\{f\}(\xi, \eta) = F(\xi, \eta) : \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$ definovaná jako

$$F(\xi, \eta) = \mathcal{D}\{f\}(\xi, \eta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})}$$

. Funkce F se také nazývá *Fourierovo spektrum* funkce f .

Definice 4.11 (Inverzní diskretní Fourierova transformace). Necht $f(x, y)$ je funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$ a necht $F(\xi, \eta)$ je její diskretní Fourierova transformace. Inverzní diskretní Fourierova transformace funkce $F(\xi, \eta)$ je funkce $\mathcal{D}^{-1}\{F\}(x, y) : \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$ definovaná jako

$$\mathcal{D}^{-1}\{F\}(x, y) = \frac{1}{MN} \sum_{\xi=0}^{M-1} \sum_{\eta=0}^{N-1} F(\xi, \eta) e^{i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})}.$$

Věta 4.6 (o inverzní Fourierově transformaci). Necht $f(x, y)$ je funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$ a necht je $F(\xi, \eta)$ její diskretní Fourierova transformace. Potom inverzní diskretní Fourierova transformace funkce $F(\xi, \eta)$ je funkce $f(x, y)$, platí tedy

$$\mathcal{D}^{-1}\{\mathcal{D}\{f(x, y)\}\} = f(x, y).$$

4.3. DISKRÉTNÍ FOURIEROVA TRANSFORMACE

Poznámka 4.5. Důkaz této věty lze nalézt v [4]. Věta 4.6 zaručuje, že v diskrétním případě je Fourierova transformace vratný proces. Díky faktu, že diskrétní Fourierova transformace je suma konečného počtu prvků, je její existence vždy zaručena, narozdíl od spojitého případu. Tyto dvě vlastnosti jsou fundamentální pro zpracování obrazu.

Definice 4.12 (Amplitudové spektrum, fázové spektrum). Nechť $f(x, y)$ je funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$ a nechť je $F(\xi, \eta)$ její diskrétní Fourierova transformace. *Amplitudové spektrum* funkce f je funkce $A(\xi, \eta) : \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{R}$ definovaná jako

$$A(\xi, \eta) = |\mathcal{D}\{f(x, y)\}| = |F(\xi, \eta)|.$$

Fázové spektrum funkce f je funkce $\Phi(\xi, \eta) : \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow]0, 2\pi)$ definovaná jako

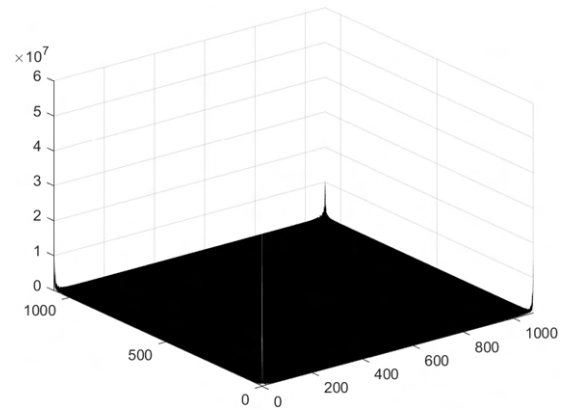
$$\operatorname{Re}(F(\xi, \eta)) = A(\xi, \eta) \cos \Phi(\xi, \eta),$$

$$\operatorname{Im}(F(\xi, \eta)) = A(\xi, \eta) \sin \Phi(\xi, \eta).$$

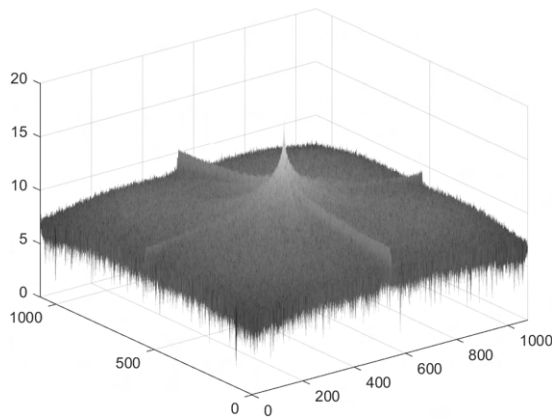
Pokud je $A(\xi, \eta) = 0$ pro daná (ξ, η) , definujeme $\Phi(\xi, \eta) = 0$.



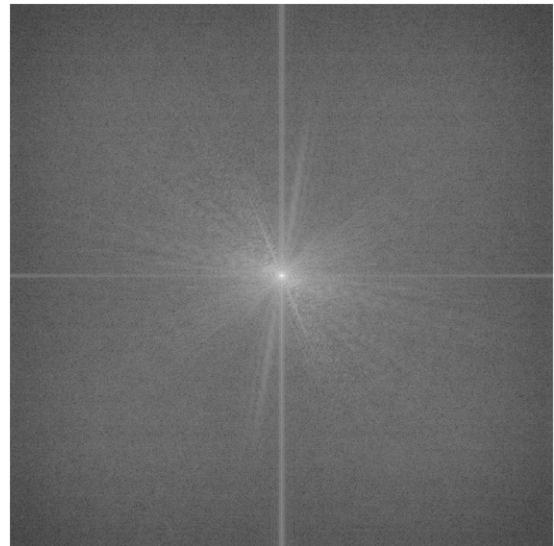
(a) Zdrojový obraz



(b) jeho fázové spektrum



(c) Vystředěné fázové spektrum v logaritmických souřadnicích



(d) Tradiční způsob zobrazení fázového spektra obrazu

Obrázek 4.1: Demonstrace fázového spektra obrazu. Z obrázku 4.1b je patrný kontrast maximálních a minimálních hodnot, je proto zvykem uvádět fázové spektrum v logaritmických souřadnicích. Počátek souřadnicového systému je v bodě $[0, 0]$, ovšem pro nás je přirozenější počátek souřadného systému ve středu obrazu, z tohoto důvodu jej vystředíme. Foto autor.

4.3.1. Diskrétní konvoluce

Diskrétní konvoluce je analogická klasické konvoluci. V následující kapitole diskrétní konvoluci využijeme pro zavedení lineárních filtrů obrazu. Nejprve je však potřeba zavést pojmy periodizace funkce a jejich diskrétní Fourierovy transformace.

4.3. DISKRÉTNÍ FOURIEROVA TRANSFORMACE

Definice 4.13 (Periodizace funkcí a jejich Fourierových spekter). Necht $f(x, y)$ je funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$ a $F(\xi, \eta)$ je její Fourierovo spektrum. Periodizace Fourierova spektra F je funkce $\tilde{F}(\xi, \eta) : \mathbb{Z}^2 \rightarrow \mathbb{C}$ definovaná jako

$$\tilde{F}(\xi, \eta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})}.$$

Periodizace funkce f je funkce $\tilde{f}(x, y) : \mathbb{Z}^2 \rightarrow \mathbb{C}$ definovaná jako

$$\tilde{f}(x, y) = \frac{1}{MN} \sum_{\xi=0}^{M-1} \sum_{\eta=0}^{N-1} F(\xi, \eta) e^{i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})}.$$

Definice 4.14 (Diskrétní Fourierova transformace periodizované funkce). Necht $f(x, y)$ je funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$. Diskrétní Fourierova transformace periodizované funkce $f(x, y)$, $\tilde{f}(x, y) : \mathbb{Z}^2 \rightarrow \mathbb{C}$ je funkce $\mathcal{D}\{\tilde{f}\}(\xi, \eta) = F(\xi, \eta) : \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$ definovaná jako

$$F(\xi, \eta) = \mathcal{D}\{\tilde{f}\}(\xi, \eta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \tilde{f}(x, y) e^{-i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})}.$$

Definice 4.15 (Inverzní diskretní Fourierova transformace periodizované funkce). Necht $f(x, y)$ je funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$ a necht $F(\xi, \eta)$ je její diskretní Fourierova transformace s periodizací $\tilde{F}(\xi, \eta) : \mathbb{Z}^2 \rightarrow \mathbb{C}$. Inverzní diskretní Fourierova transformace funkce $\tilde{F}(\xi, \eta)$ je funkce $\mathcal{D}^{-1}\{\tilde{F}\}(x, y) : \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$ definovaná jako

$$\mathcal{D}^{-1}\{\tilde{F}\}(x, y) = \frac{1}{MN} \sum_{\xi=0}^{M-1} \sum_{\eta=0}^{N-1} \tilde{F}(\xi, \eta) e^{i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})}.$$

Poznámka 4.6. Uvedme si vybrané vlastnosti diskretní Fourierovy transformace.

1.	$f(x, y)$	$F(\xi, \eta)$
2.	$\alpha f(x, y) + \beta g(x, y)$	$\alpha F(\xi, \eta) + \beta G(\xi, \eta)$
3.	$\tilde{f}(x - x_0, y - y_0)$	$F(\xi, \eta) \cdot e^{-i((\xi x_0 + \eta y_0))}$
4.	$e^{i(\xi_0 x + \eta_0 y)} f(x, y)$	$\tilde{F}(\xi - \xi_0, \eta - \eta_0)$
5.	$\tilde{f}(-x, -y)$	$\tilde{F}(-\xi, -\eta)$
6.	$(f * g)(x, y)$	$F(\xi, \eta) \cdot G(\xi, \eta)$
7.	$f(x, y) \cdot g(x, y)$	$1/MN \cdot (F * G)(\xi, \eta)$

Další vlastnosti společně s jejich důkazy lze nalézt v [4][2].

Definice 4.16 (Diskrétní periodická konvoluce). Necht $f_1(x, y), f_2(x, y)$ jsou funkce $\{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$. Funkce $f_1 * f_2$ se nazývá *diskrétní periodická konvoluce* funkcí $f_1(x, y), f_2(x, y)$, definovaná jako

$$f_1 * f_2 = \sum_{s=0}^{M-1} \sum_{t=0}^{N-1} f_1(s, t) \tilde{f}_2(x - s, y - t).$$

4. FOURIEROVA TRANSFORMACE A KONVOLUCE

Věta 4.7 (Konvoluční teorém). Necht $f_1(x, y), f_2(x, y) \in \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $M, N \in \mathbb{N}$ mají Fourierovy spektra $F_1(\xi, \eta), F_2(\xi, \eta)$. Potom platí

$$\mathcal{D}\{f_1 * f_2\} = F_1 \cdot F_2.$$

Důkaz. Důkaz je převzatý z [4]. Necht $f(x, y) = f_1(x, y) * f_2(x, y)$. Pak platí

$$\begin{aligned} \mathcal{D}\{f(x, y)\} &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(\sum_{s=0}^{M-1} \sum_{t=0}^{N-1} f_1(s, t) \tilde{f}_2(x-s, y-t) \right) e^{-i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})} = \\ &= \sum_{s=0}^{M-1} \sum_{t=0}^{N-1} f_1(s, t) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \tilde{f}_2(x-s, y-t) e^{-i2\pi(\frac{x\xi}{M} + \frac{y\eta}{N})} = \begin{vmatrix} p = x-s \\ q = y-t \end{vmatrix} = \\ &= \sum_{s=0}^{M-1} \sum_{t=0}^{N-1} f_1(s, t) \sum_{p=-s}^{M-1-s} \sum_{q=-t}^{N-1-t} \tilde{f}_2(p, q) e^{-i2\pi(\frac{\xi(p+s)}{M} + \frac{\eta(q+t)}{N})} = \\ &= \sum_{s=0}^{M-1} \sum_{t=0}^{N-1} f_1(s, t) e^{-i2\pi(\frac{s\xi}{M} + \frac{t\eta}{N})} \sum_{p=-s}^{M-1-s} \sum_{q=-t}^{N-1-t} \tilde{f}_2(p, q) e^{-i2\pi(\frac{p\xi}{M} + \frac{q\eta}{N})} = \\ &= F_1(\xi, \eta) * F_2(\xi, \eta). \end{aligned}$$

□

Poznámka 4.7. Analogicky pro inverzní případ platí

$$\mathcal{D}\{f_1 \cdot f_2\} = \frac{1}{MN} F_1 * F_2.$$

Tuto větu oceníme zejména při použití filtrů. Umožňuje nám si vybrat, zda použijeme konvoluci nebo násobení Fourierových spekter pro aplikaci filtrů, neboť dle této věty jsou výsledky ekvivalentní.

Poznámka 4.8. Necht f, g, h jsou reálné, nebo diskrétní funkce jedné, nebo dvou proměnných. Konvoluce má následující vlastnosti:

1. Komutativita

$$f * g = g * f$$

2. Asociativita

$$f * (g * h) = (f * g) * h$$

3. Distributivita

$$f * (g + h) = (f * g) + (f * h)$$

Vlastnost 1. plyne z definice z multiplikativní komutativnosti. Vlastnost 2. vyplývá z Fubiniho věty. Vlastnost 3. plyne z linearitě integrálu/sumy. Důkazy lze najít v [10].

4.4. Rychlá Fourierova transformace

Poznámka 4.9. Algoritmům, které cílí na zrychlení výpočtu diskrétní Fourierovy transformace se říká *rychlá Fourierova transformace*, anglicky *fast Fourier transform*, z čehož plyne často užívaná zkratka *FFT*.

4.4. RYCHLÁ FOURIEROVA TRANSFORMACE

Pro ilustraci, uveďme si definici jednorozměrné diskrétní Fourierovy transformace.

Definice 4.17 (Diskrétní Fourierova transformace). Necht $f(x)$ je funkce $\{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$, $N \in \mathbb{N}$. *Diskrétní Fourierova transformace* funkce $f(x)$ je funkce $\mathcal{D}\{f\}(\xi) = F(\xi) : \{0, 1, \dots, N-1\} \rightarrow \mathbb{C}$ definovaná jako

$$F(\xi) = \mathcal{D}\{f\}(\xi) = \sum_{x=0}^{N-1} f(x) e^{-\frac{2\pi i}{N} x \xi}.$$

Připomeňme si užitečnou identitu.

Lemma 4.8 (Eulerův vzorec).

$$e^{-\frac{2\pi i}{N}} = \cos\left(\frac{2\pi}{N}\right) - i \sin\left(\frac{2\pi}{N}\right).$$

Evaluační náročnost diskrétní Fourierovy transformace je řádu $O(N^2)$. Existuje však rychlejší způsob.

Lemma 4.9 (Danielsonovo-Lanczosovo lemma[19]). Necht je $f(x)$ diskrétní funkce, $F(\xi)$ je její diskrétní Fourierova transformace ve smyslu definice 4.17 a necht N je sudé. Pak platí

$$F(\xi) = \sum_{x=0}^{N/2-1} f(2x) e^{-\frac{2\pi i}{N/2} x \xi} + e^{-\frac{2\pi i}{N/2} \xi} \sum_{x=0}^{N/2-1} f(2x+1) e^{-\frac{2\pi i}{N/2} x \xi}.$$

Obraz sudých členů označme

$$F_{\xi}^0 = \sum_{x=0}^{N/2-1} f(2x) e^{-\frac{2\pi i}{N/2} x \xi}$$

a obraz lichých členů označme

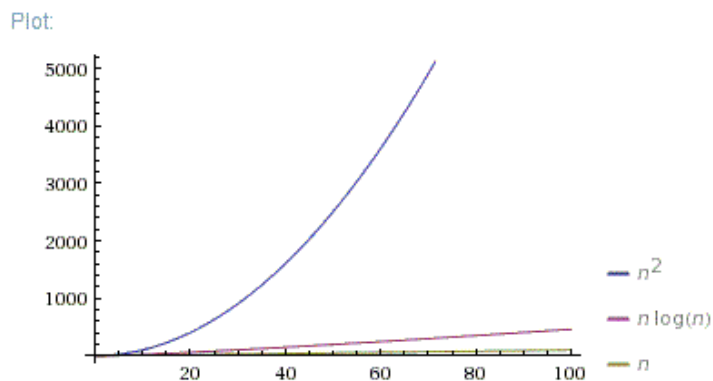
$$F_{\xi}^1 = \sum_{x=0}^{N/2-1} f(2x+1) e^{-\frac{2\pi i}{N/2} x \xi}.$$

Platí pak

$$F(\xi) = \mathcal{D}\{f\}(\xi) = F_{\xi}^0 + e^{-\frac{2\pi i}{N/2} \xi} F_{\xi}^1.$$

F_{ξ}^0 a F_{ξ}^1 jsou opět diskrétní Fourierovy transformace. Danielsonovo-Lanczosovo lemma lze tedy aplikovat opakovaně, každá aplikace snižuje náročnost výpočtu. Na tomto principu je založen Cooleyho-Tukeyův algoritmus[3]. Analogicky funguje inverzní rychlá Fourierova transformace. K maximálnímu využití potenciálu tohoto algoritmu dochází, pokud mají data rozměr mocniny dvojky. Tímto způsobem lze snížit výpočetní náročnost na řád $O(N \log_2 N)$.

Variantě FFT, kdy dělíme koeficienty na sudé a liché se říká *radix-2*, není to však jediná implementace, pro různé účely je vhodné zvolit jinou faktorizaci N . Také lze tento algoritmus rozšířit pro vícerozměrnou diskrétní Fourierovu transformaci. Pro některé aplikace lze využít řídké rychlé Fourierovy transformace, čímž je možné dosáhnout sublineární[8] výpočetní náročnosti.



Obrázek 4.2: Úspora operací je znatelná už i pro $n \leq 100$

FFT bývá označován za nejdůležitější numerický algoritmus století. Díky své super-lineární náročnosti nachází uplatnění v širokém spektru inženýrských aplikací, jako je komprese signálu pro Wi-Fi a LTE, JPEG komprese, kódování videoformátů, komprese zvukových signálů. Algoritmus umožňuje velmi efektivní násobení velkých celých čísel a polynomiální multiplikaci, nebo řešení diferenciálních rovnic. Díky konvolučnímu teoremu 4.7 lze využít FFT k rychlému výpočtu konvoluce, algoritmus je tedy klíčový pro zpracování obrazu.

5. Lineární filtry

Lineární filtry jsou operace, které umožňují manipulaci s obrazem, například jeho rozmazání, nebo zaostření. V této kapitole si popíšeme princip fungování filtru typu neostrá maska založené na Gaussovském jádře. Nejprve si definujeme jádro konvoluce.

Definice 5.1. *Jádro konvoluce* neboli *kernel* je zobrazení $k(s, t) : \{-a, -a+1, \dots, 0, \dots, a-1, a\} \times \{-b, -b+1, \dots, 0, \dots, b-1, b\} \rightarrow \mathbb{R}$, $a, b \geq 0$, $a, b \in \mathbb{Z}$. Říkáme, že jádro má *rozměr* $m \times n$, pokud platí $m = 2a + 1$, $n = 2b + 1$.

Poznámka 5.1. Jádro konvoluce lze reprezentovat maticí $m \times n$. Pro názornost budou dále uvedené příklady konvolučních jader zapsané jako matice.

Definice 5.2 (Lineární prostorový filtr). Mějme kernel $k(s, t)$ o rozměru $m \times n$ a obraz ve smyslu definice 3.1 $f(x, y)$. Obraz $g(x, y)$ se nazývá *lineární prostorový filtr* [7] s jádrem $k(s, t)$, pokud platí

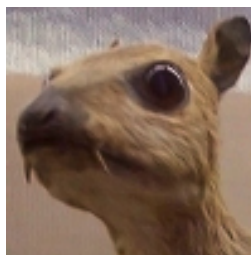
$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b k(s, t) f(x + s, y + t).$$

Poznámka 5.2. Zřejmě lineární prostorový filtr je konvoluce jádra $k(s, t)$ s obrazem $f(x, y)$. Budeme proto používat značení $g(x, y) = k(s, t) * f(x, y)$

Při použití této definice je třeba být opatrný, neboť při výpočtu se můžeme dostat mimo definiční obor obrazu. Abychom vyřešili tento problém, je třeba k obrazu přidat rámeček. Tento rámeček můžeme vyplnit konstantou, nebo obraz můžeme rozšířit periodicky, či zrcadlově. Více o této problematice se dá zjistit v [6].

Myšlenka lineárních filtrů spočívá v tom, že pro transformaci hodnoty pixelu vstupního obrazu se uvažuje jeho okolí. Jakým způsobem lineární filtr uvažuje okolí je charakterizováno konvolučním jádrem. Pokud tedy použijeme jádro, které má všude nuly kromě středu, kde jednička, dostaneme opět vstupní obrázek. Pokud by se v tomto jádře jednička nacházela mimo střed, došlo by k posunu obrázku. Pokud by místo jedničky bylo jiné kladné číslo, došlo by ke zesvětlení, nebo ztmavení obrazu.

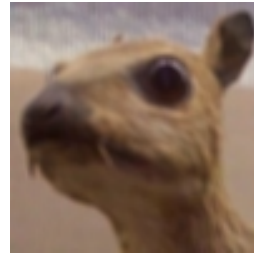
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



Obrázek 5.1: Aplikace jádra identity. Foto Michael Plotke [15].

Můžeme vzít aritmetický průměr okolí, tím dojde k rozmazání obrázku. Jako jádro použijeme matici jedniček. Matici je potřeba normalizovat, tedy vydělit počtem prvků, aby se celková světlost obrazu nezměnila, a aby nedošlo k přetečení datového typu. Takovému jádru se anglicky říká *box blur*.

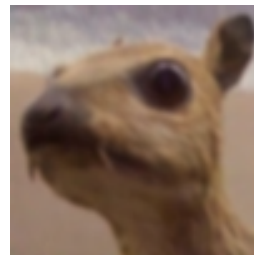
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Obrázek 5.2: Rozostření obrazu pomocí box blur. Foto Michael Plotke [15].

Pokud použijeme větší jádro pro boxblur, bude efekt rozmazání větší. Lze takto docílit i rozmazání v určitém směru, kdy použijeme větší koeficienty v daném směru, nebo použitím jedničkového jádra obdelníkového tvaru. Přírozenějším způsobem, jak docílit rozmazání obrazu, je použít koeficienty dle dvojrozměrného normálního rozdělení $e^{-\frac{x^2+y^2}{2\sigma^2}}$. Volbou σ a velikosti jádra lze ovlivnit míru rozmazání.

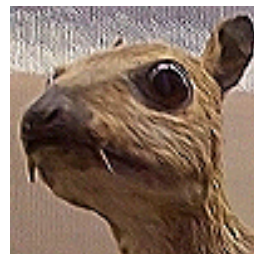
$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$



Obrázek 5.3: Rozostření obrazu Gaussovským jádrem 5×5 . Foto Michael Plotke [15].

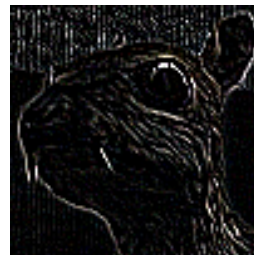
Analogicky, pokud je potřeba obraz zaostřit, lze použít k tomu určené jádro. Také existuje celá řada kernelů určených hledání hran, nebo zvýrazňování struktur v daném směru. Níže jsou uvedeny dva jednoduché příklady.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



Obrázek 5.4: Ukázka ostríčího jádra. Foto Michael Plotke [15].

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Obrázek 5.5: Aplikace jádra pro zvýraznění přechodů v obraze - hledání hran. Foto Michael Plotke [15].

Sofistikovanější způsob, jak docílit zvýraznění požadovaných struktur, se nazývá *metoda neostré masky*. Myšlenka je následující, rozmazeme původní obraz, k tomu využijeme Gaussovského jádra, odečteme tento rozmazaný obraz od původního obrazu, tím získáme

obraz, který obsahuje pouze vysoké frekvence, tedy detaily. Tento obraz nazveme neostrou maskou. Nakonec, vynásobíme původní obraz a neostrou masku vhodnými vahami a sečteme je. V závislosti na míře rozmazání a váhových koeficientech vznikne obraz se zvýrazněnými detaily.

Definice 5.3 (Metoda neostré masky). Mějme obraz $f(x, y)$ a Gaussovské jádro $g(x, y)$. Obraz $u(x, y)$ se nazývá *neostrá maska*, pokud platí

$$u(x, y) = f(x, y) - f(x, y) * g(x, y).$$

Výstupní obraz $f'(x, y)$ získaný vztahem

$$f'(x, y) = a \cdot f(x, y) + b \cdot u(x, y),$$

kde $a, b \in \mathbb{R}$, je získaný aplikací *metody neostré masky*.

Poznámka 5.3. Tato definice vychází z [7][17].

Výše popsaná metoda neostré masky funguje na principu konvoluce obrazu s Gaussovským jádrem. Díky konvolučnímu teorému 4.7 víme, že tuto metodu lze ekvivalentně vyjádřit ve frekvenční doméně.

Věta 5.1 (Metoda neostré masky). Mějme obraz $f(x, y)$ a Gaussovské jádro $g(x, y)$, kde $F(\xi, \eta)$ a $G(\xi, \eta)$ respektive jsou jejich Fourierovy transformace. Obraz $u(x, y)$ je *neostrá maska*, pokud platí

$$u(x, y) = f(x, y) - \mathcal{F}^{-1} \{F(\xi, \eta) \cdot G(\xi, \eta)\}.$$

Výstupní obraz $f'(x, y)$ získaný vztahem

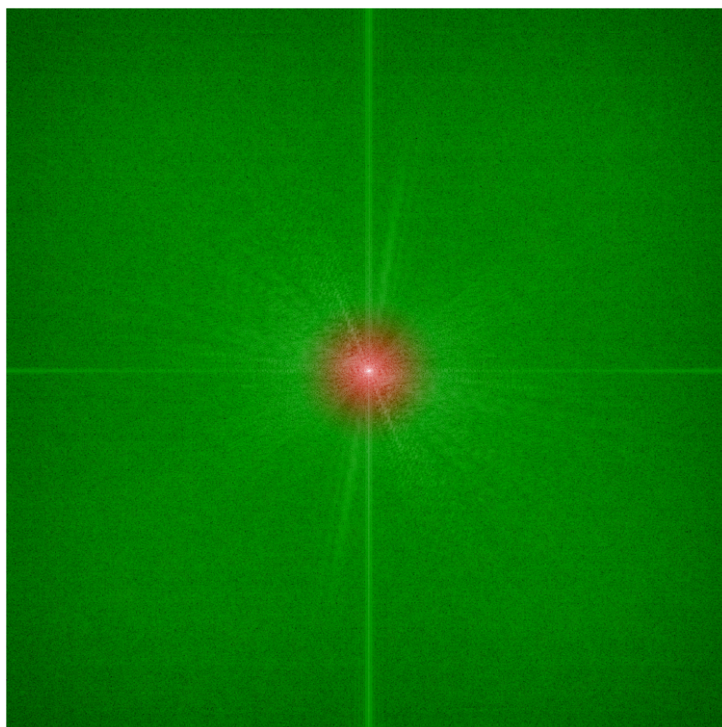
$$f'(x, y) = a \cdot f(x, y) + b \cdot u(x, y),$$

kde $a, b \in \mathbb{R}$, je získaný aplikací *metody neostré masky*.

Poznámka 5.4. Gaussovské jádro nemusí mít stejnou velikost, jako vstupní obraz. Pokud je menší, tak je potřeba rozšířit definiční obor $GF(\xi, \eta)$ nulami tak, aby původní definiční obor zůstal ve středu rozšířeného definičního oboru.

V praktické části využijeme tuto variantu metody neostré masky, abychom mohli využít FFT. Nicméně většina moderních řešičů využívá FFT pro zrychlení konvoluce, tudíž i konvoluční implementace je stejně dobře použitelná.

Poznámka 5.5 (Princip neostré masky.). Při použití metody neostré masky potlačujeme střed, tedy nízké frekvence obrazu a zvýrazňujeme okraje, tedy vysoké frekvence obrazu. Právě tyto frekvence jsou pro naše oko důležité, neboť obsahují detaily obrazu. Na obrázku níže je červenou označena oblast, kterou potlačujeme a zelenou oblast, kterou si ponecháváme. Pokud takovýmto způsobem upravíme frekvenční spektrum obrazu, získáme právě frekvenční spektrum neostré masky. Tímto způsobem zároveň můžeme snížit bitovou hloubku obrazu. Je nutné si uvědomit, že amplituda je zobrazena v logaritmickém měřítku, pokud by tak nebylo, zdál by se obrázek celý černý s malou bílou tečkou uprostřed.



Obrázek 5.6: Schéma ukazující, jaké frekvence obsahuje neostrá maska.

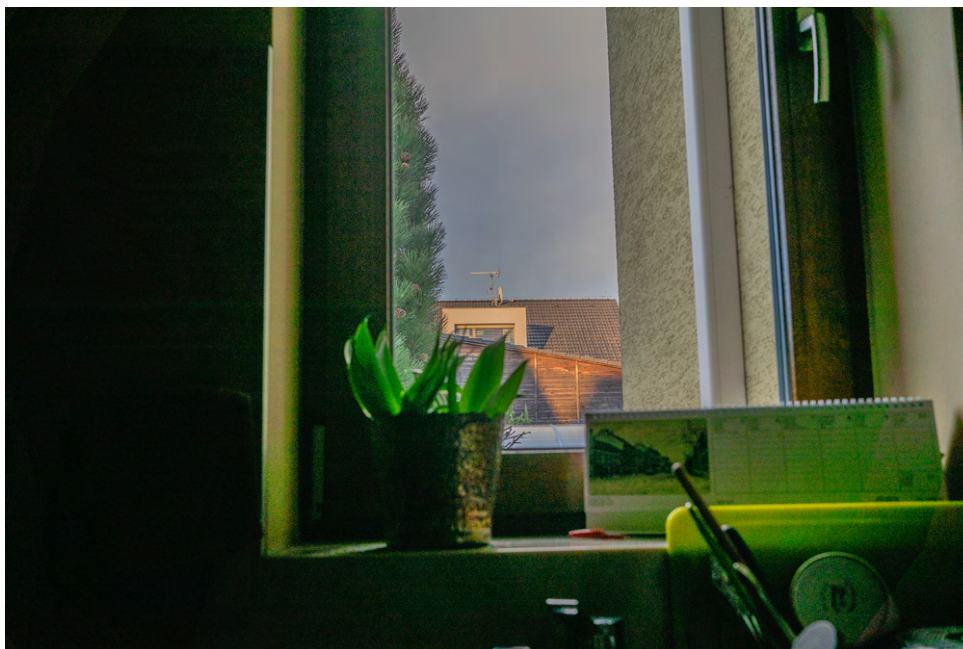
6. Praktická část

V této kapitole se budeme věnovat aplikaci teoretických poznatků popsaných v předešlých kapitolách na konkrétní problém. Máme před sebou následující scénu.



Obrázek 6.1: Výchozí scéna.

Scéna je náročná, z okna vidíme nebe a protější dům. Aby nedošlo k saturaci pixelů v oblasti nebe a protějšího domu, musíme exponovat pro světlou část. To ovšem znamená, že popředí, většina fotografie, zanikne ve tmě. Nicméně to neznamená, že o data v této oblasti přijdeme, byť se zdá být většina obrazu zcela černá, moderní senzor dokáže dobře zachytit detaily i ve stínech, je potřeba je pouze zviditelnit. Abychom v postprocesu mohli zviditelnit co nejvíce detailů, ukládáme snímek do formátu RAW. Takovéto soubory lze zpracovávat v celé škále programů, naší volbou bude Adobe Lightroom. Výsledek snahy vrátit detaily ze stínů běžným postupem vypadá následovně.



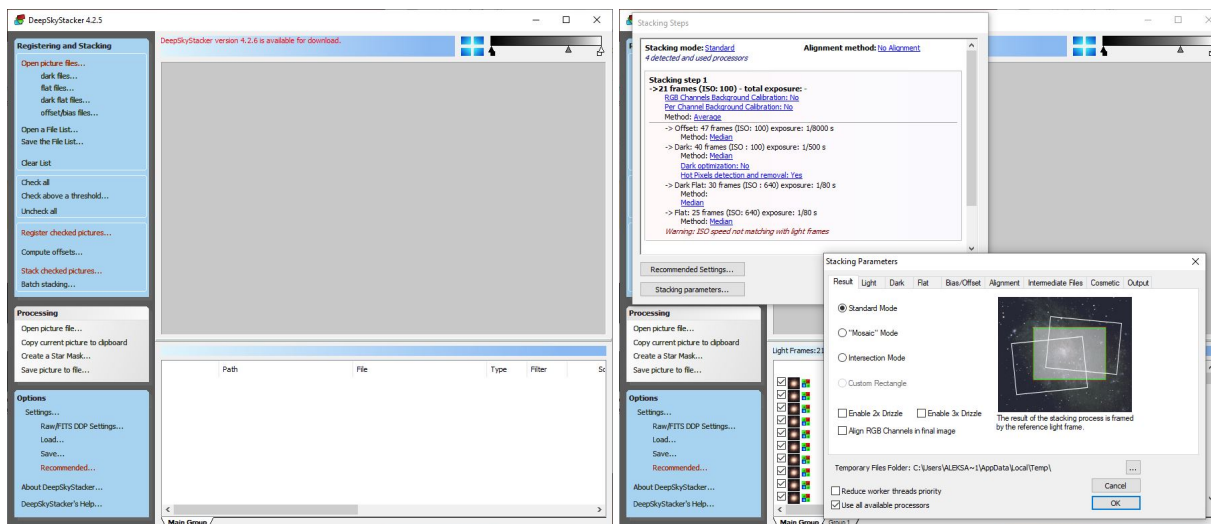
Obrázek 6.2: Úprava jednoho .raw souboru v Adobe Lightroom.

Podařilo se nám obnovit mnoho detailů, vidíme okno, květináč, nějaké nádobí, texturu omítky. Nicméně levá strana je stále úplně zahalená, v obraze je velmi výrazný šum a zesvětlené části mají nepřírozené barvy a zelené fleky. Výsledek je nedostačující, musíme přistoupit k pokročilejší technice.

6.1. Kalibrace pomocí Deep Sky Stacker

Jak bylo popsáno v kapitole 3, potřebujeme zvýšit poměr signálu ku šumu, pořídíme proto sérii kalibračních snímků. K tomu budeme potřebovat stativ a nejlépe dálkovou spoušť. Pokud je to možné, je vhodné zapnout elektronickou závěrku. To vše, abychom minimalizovali otřesy fotoaparátu a vzájemné posuny vůči jednotlivým snímkům. Nastavení pro expozici této scény je čas 1/30 sekundy, f/8 při citlivosti ISO 100. Pro tuto scénu jsou nejpodstatnější light framey, tedy obrazové snímky, neboť expozice je krátká s nízkou citlivostí, termální šum v této scéně nebude tolik výrazný. I přesto si však připravíme kompletní sérii bias a dark frameů. Pro skládání snímků využijeme DeepSkyStacker. DSS je přístupný zdarma, má českou lokalizaci a navíc jde o open-source projekt napsaný v C++.

6.2. VLASTNÍ IMPLEMENTACE



Obrázek 6.3: Prostřední DeepSkyStacker.

DSS je program zaměřený na astrofotografii, poslouží ovšem skvěle i pro obecnou fotografii. V sekci *Registering and Stacking* otevřeme příslušné kalibrační snímky. Zvolíme *Check all* a následně *Stack checked pictures...*, v *Stacking parameters* zvolíme požadované parametry. Pro tuto scénu bude nejvhodnější použít *average* pro light snímky a pro všechny ostatní *median*. Deepskystacker využívá polohu viditelných hvězd ke srovnání snímků, ovšem v naší scéně nejsou žádné hvězdy viditelné, proto volíme *No alignment*. Výsledný 32bitový snímek necháme uložit do formátu TIF. Máme tedy HDR snímek připravený ke zpracování.

6.2. Vlastní implementace

Nyní v HDR obraze zviditelníme detaily zapomocí gamma korekce a metody neostré masky. Implementace je napsaná v jazyce C++ s využitím knihovny OpenCV. OpenCV je otevřená knihovna pro manipulaci s obrazem, zaměřená především na počítačové vidění (Computer Vision) a zpracování obrazu. Je vysoce optimalizovaná, dokáže využít GPU akcelerace, techniky používané zde na statický obraz lze rozšířit na video. Knihovna funguje i pro další programovací jazyky, jako například Python, Java, Octave apod. Zdrojový kód implementace společně s dokumentací lze nalézt v příloze.

6.2.1. OpenCV

Pro práci s OpenCV je stěžejní třída `Mat`¹. Stručně řečeno, třída `Mat` reprezentuje n-dimenzionální pole, tuto třídu budeme využívat k uchovávání obrazu. Obraz ve stupních šedi reálných hodnot se uloží v jednom kanále. Po Fourierově transformaci dostaneme obraz obecně komplexní, ten je reprezentován dvěma kanály. Pokud máme barevný obraz, může být ve třídě `Mat` reprezentován například třemi kanály dle RGB (nebo i například HLS) schématu, kde každý kanál je dvourozměrné pole. Bitová hloubka záleží na použitém datovém typu, pro nás budou podstatné dva datové typy, 32bitový float, datový typ s pohyblivou desetinnou čárkou, jednotlivé pixely budou nabývat hodnot mezi 0 a 1,

¹Dokumentace https://docs.opencv.org/master/d3/d63/classcv_1_1Mat.html

tomu náleží Mat typu `CV_32F`. V tomto datovém typu budeme provádět veškeré výpočty. Výsledný obraz převedeme do Mat typu `CV_8U`, jde o 8bitový unsigned char, nabývající hodnoty mezi 0 a 255. Pro provedení diskretní Fourierovy transformace je použita funkce `dft()`².

6.2.2. Vlastní funkce

Pro aplikaci dříve popsaných metod bylo potřeba napsat následující funkce:

```
1  #pragma once
2  #include <opencv.hpp>
3  #include <core.hpp>
4  #include <imgcodecs.hpp>
5  #include <highgui.hpp>
6  #include <iostream>
7
8  using namespace cv;
9  using namespace std;
10
11 void gauss(const Size& size, Mat& output, float sigmax, float sigmay, float amplitude = 1.0f);
12 // creates a mask using normal distribution
13
14 void gammaCorrection(Mat& input, Mat& output, float gamma);
15 // adjusts the image brightness
16
17 void rearrange(const Mat& input, Mat& output);
18 // rearrange the quadrants of fourier image so that the origin is at the image center
19
20 void complexMultiplication(const Mat& inputa, const Mat& inputb, Mat& output); //inputa and inputb need to be of same size m x n
21 // splits complex inputa matrix into a real and imaginary parts
22 // both real and imaginary parts are multiplied pixelwise inputb, merged again into output
23
24 void amplitudeSpectrum(const Mat& input, Mat& output);
25 // computes a DFT of an image, centers the spectrum and displays its logarithmic amplitude
26
27 void bwHDR(const Mat& input, Mat& output, float gamma, float sigma, float im_koef, float unsharp_koef);
28 // makes details of an HDR image visible - for black&white result
29
30 void rgbHDR(const Mat& input, Mat& output, float gamma, float sigma, float im_koef, float unsharp_koef);
31 // makes details of an HDR image visible - for color result - edits rgb channels separately
32
33 void hlsHDR(const Mat& input, Mat& output, float gamma, float sigma, float im_koef, float unsharp_koef);
34 // makes details of an HDR image visible - for color result - edits only luminance channel of hls representation
```

Obrázek 6.4: Hlavičkový soubor s vytvořenými funkcemi.

Stručný popis pomocných funkcí:

1. `gauss()`. Vytvoří Mat s hodnotami dle Gaussovy funkce. Lze nastavit střední hodnotu i rozptyl v obou směrech. Tato matice bude následně vynásobena s Fourierovým spektrem obrazu.
2. `gammaCorrection()`. Podle vzorce

$$x_{out} = \left(\frac{x_{in} - b}{w - b} \right)^{\frac{1}{\gamma}}$$

se jednotlivé pixely obrazu transformují. x je hodnota pixelu, w je nejsvětlejší pixel obrazu, tedy whitepoint, b je nejtmavší hodnota obrazu, tedy blackpoint. γ je parametr. Platí, že čím větší γ , tím světlejší obraz. Tato transformace nám umožní zesvětlit obraz, aniž by došlo k saturaci pixelů.

3. `rearrange()`. Tato funkce slouží k vystředění Fourierova spektra.
4. `complexMultiplication()`. Slouží k násobení dvou obrazů o stejných rozměrech po prvcích, tedy po příslušných pixelech. OpenCV nemá zabudovanou funkci pro násobení komplexního obrazu reálným obrazem.

²Dokumentace https://docs.opencv.org/3.4/d8/d01/tutorial_discrete_fourier_transform.html

6.2. VLASTNÍ IMPLEMENTACE

5. `amplitudeSpectrum()`. Vypočte DFT obrazu, vycentruje jeho frekvenční spektrum a zobrazí jej v logaritmických souřadnicích.

Zpracování obrazu probíhá ve funkcích `bwHDR()`, `rgbHDR()`, `hlsHDR()`. Popišme si, co vše dělá funkce `bwHDR()`.

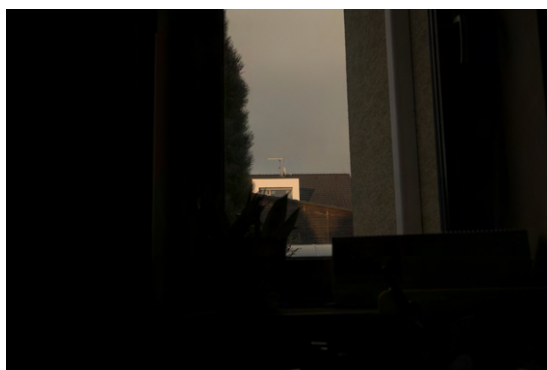
- Konvertuje vstupní obraz do 32bitového obrazu ve stupních šedi.
- Pomocí gamma korekce zesvětlí obraz, aniž by satureoval světlé oblasti. Míru zesvětlení určuje parametr `gamma`.
- Proveďte diskrétní Fourierovu transformaci vstupního obrazu. Vytvoří Gaussovu masku dle parametru `sigma`. Tuto masku vynásobí s Fourierovým obrazem vstupního obrazu a provede zpětnou diskrétní Fourierovu transformaci. Takto vznikne rozmazaný obraz. Míru rozmazání určuje parametr `sigma`, čím menší je, tím rozmazanější je obraz.
- Vytvoří neostrou masku ve smyslu 5.1.
- Výstup funkce je obraz vzniklý metodou neostře masky ve smyslu 5.1, parametry `im_koef` a `unsharp_koef` určují váhy v součtu původního obrazu a neostře masky.

Takto získáme výsledný obraz ve stupních šedi. Pokud chceme získat barevný výstup, je třeba myšlenky aplikované v `bwHDR()` zobecnit. Návrhy možných řešení jsou funkce `rgbHDR()` a `hlsHDR()`. Zjednodušeně, `rgbHDR()` rozloží RGB obraz do tří obrazů ve stupních šedi, aplikuje na ně stejné kroky jako v `bwHDR()` a výsledné obrazy složí opět do jednoho RGB obrazu. `hlsHDR()` pracuje s reprezentací barevného obrazu. HLS reprezentuje barevný obraz třemi složkami: Hue, Saturation, Luminance. Hue a Saturation reprezentují barvy, Luminance reprezentuje světlost. Můžeme tedy chápat Luminance složku jako obraz ve stupních šedi a aplikovat na něj postup z `bwHDR()`, následně obrazy opět složit do jednoho barevného HLS obrazu.

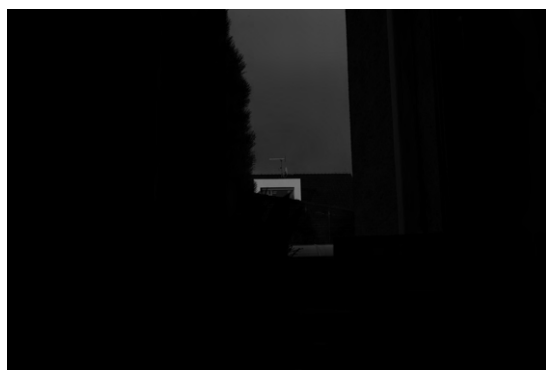
Po aplikaci těchto metod je možné obrazy uložit do téměř jakéhokoliv formátu. Pro naše potřeby je převedeme do `CV_8U` a následně uložíme do 8bitového `.jpg` formátu.

6.2.3. Ukázka výsledků

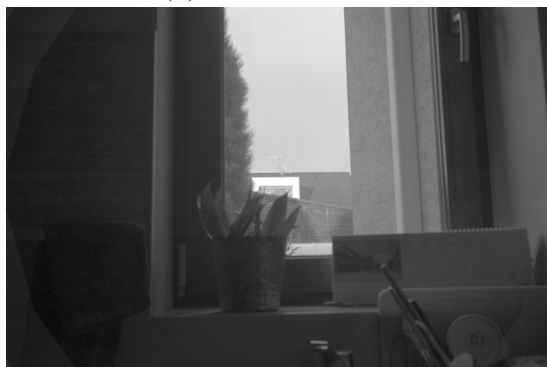
Na následujících obrázcích vidíme výsledky aplikovaných metod. Na obrázcích 6.5 vidíme jednotlivé mezikroky potřebné k vytvoření finálního obrazu. Obraz převedeme do stupní šedi, aplikujeme gamma korekci, obraz rozmazeme, vytvoříme neostrou masku, kterou nakonec přičteme k původnímu obrazu. Výsledek byl získán s nastavením parametrů `gamma= 4.7`, `sigma= 22`, `im_koef= 1` a `unsharp_koef= 1` při rozlišení obrazu 1500×1000 . Metodami `bwHDR()` a `rgbHDR()` se podařilo získat výstup s přirozenými barvami, který odpovídá tomu, jak scénu ve skutečnosti vidíme. Obrazy vytvořené metodou `hlsHDR()` jsou výrazně sytější, než u RGB metody, v 6.8d a 6.6c dokonce příliš syté. Vyvodíme z toho závěr, že pouhá manipulace Luminance kanálu není dostatečná, je k tomu potřeba korigovat i kanály Hue a Saturation.



(a) Vstupní obraz.



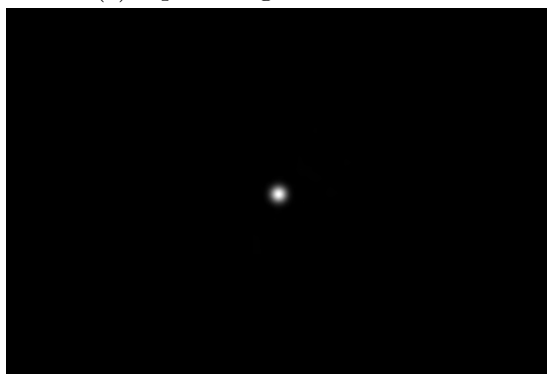
(b) Převedení do stupní šedi.



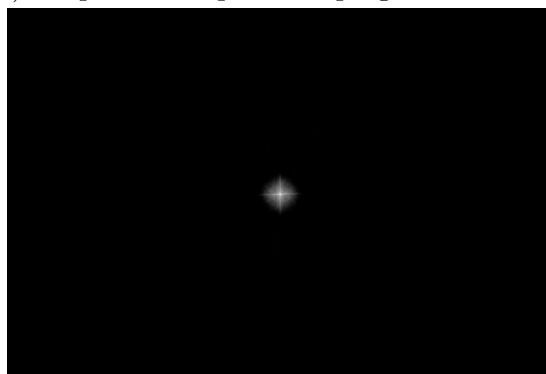
(c) Aplikace gamma korekce.



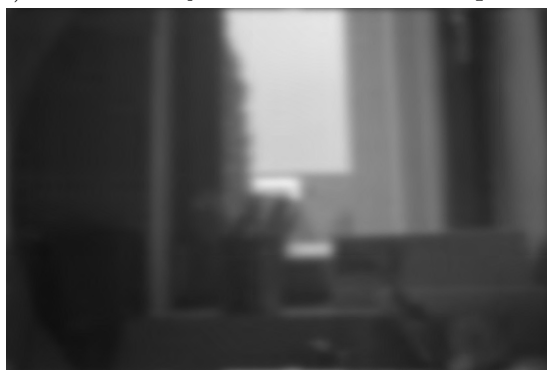
(d) Amplitudové spektrum po gamma korekci.



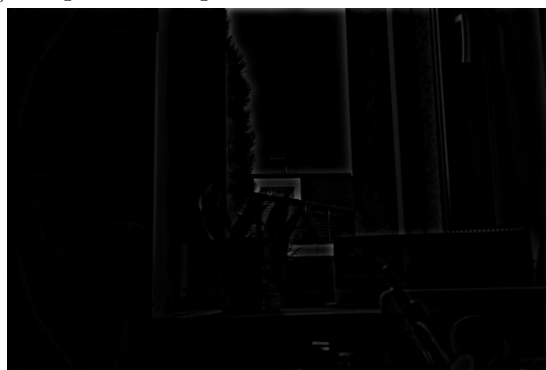
(e) Gaussovské jádro ve frekvenčním spektru.



(f) Amplitudové spektrum rozmazaného obrazu.



(g) Rozmazání obrazu.



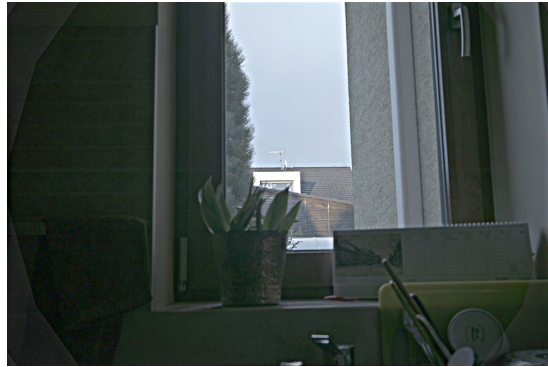
(h) Neostrá maska.

Obrázek 6.5: Mezikroky při úpravě obrazu.

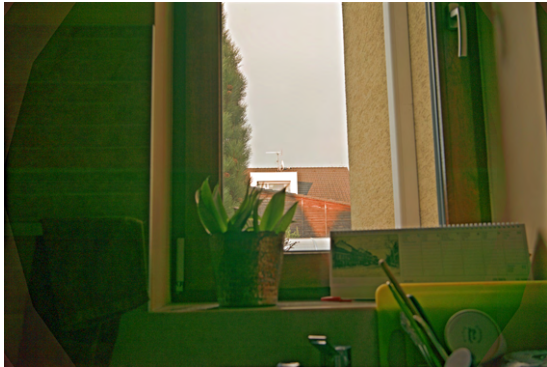
6.2. VLASTNÍ IMPLEMENTACE



(a) Výsledný obraz ve stupních šedi.

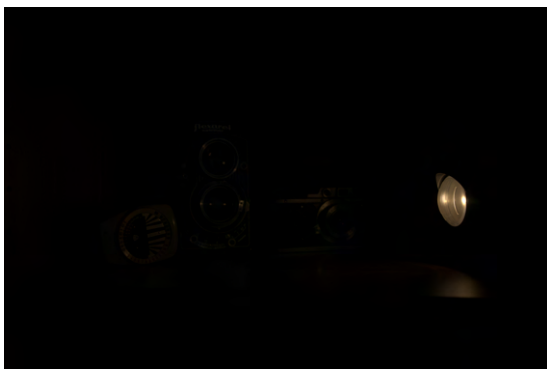


(b) Barevný výstup z `rgbHDR()`.

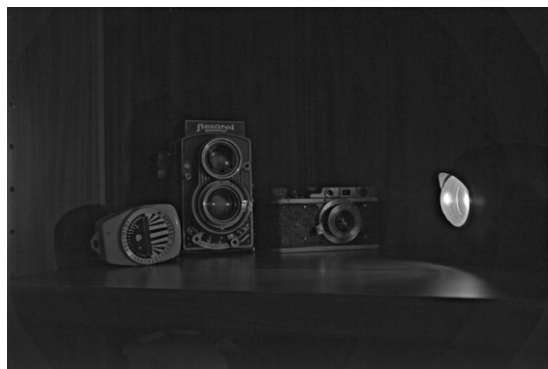


(c) Barevný výstup z `hlsHDR()`.

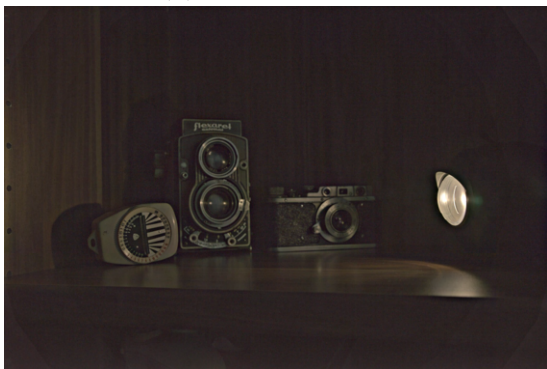
Obrázek 6.6: Výsledek zpracování scény.



(a) Vstupní obraz.



(b) Výsledek v stupních šedi.



(c) Výsledek RGB metodou.



(d) Výsledek HSL metodou.

Obrázek 6.7: Jablko.



(a) Vstupní obraz.



(b) Výsledek v stupních šedi.



(c) Výsledek RGB metodou.

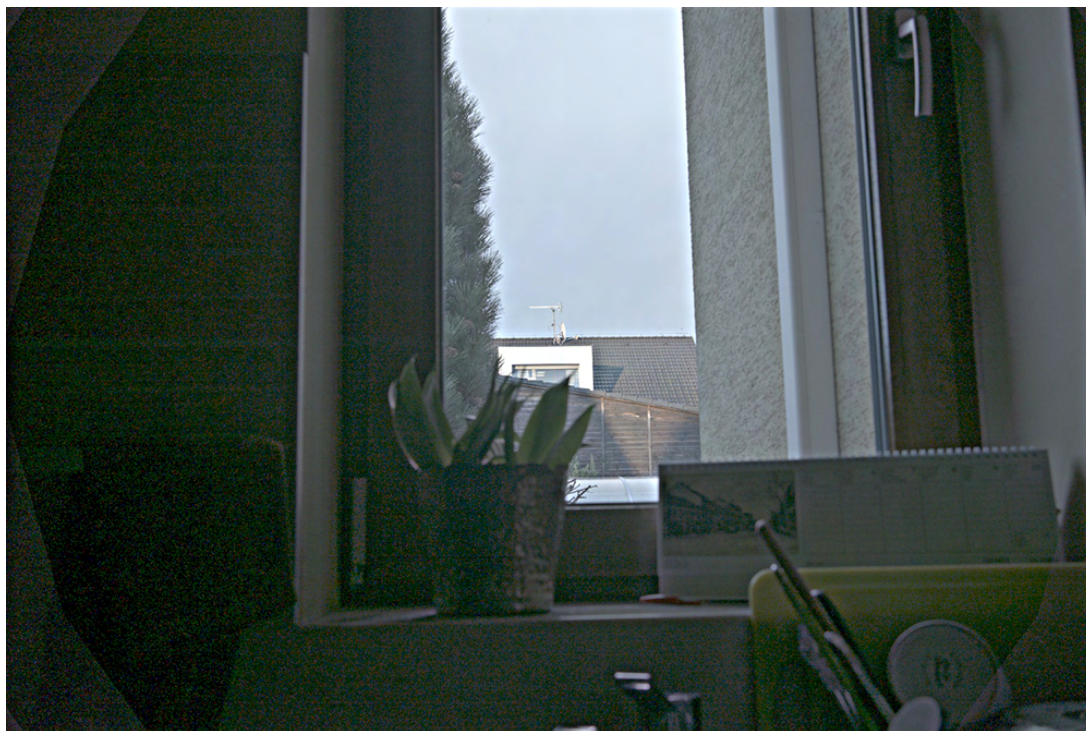


(d) Výsledek HLS metodou.

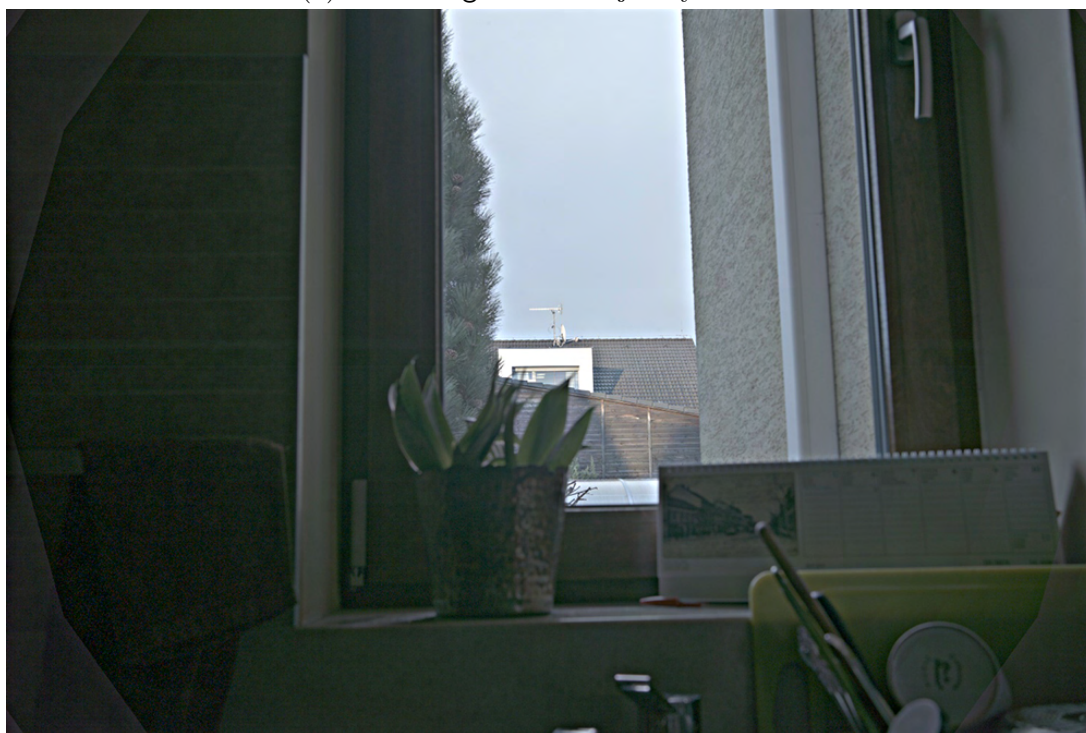
Obrázek 6.8: Okno.

Na závěr si uvedme srovnání vlivu kalibrace na snížení šumu ve výsledném obraze. Vezmeme jediný .raw soubor a uměle jej konvertujeme do 32bitového .tif souboru (například "kalibrací" v DeepSkyStackeru). Obrázek 6.9a je výsledkem zpracování jediného light frameu. Ve srovnání s 6.9b trpí obraz výraznou přítomností šumu (je nutné si uvědomit, že obrázky uvedené v této práci jsou zmenšené a tudíž vliv šumu je menší, než na obrázku plného rozlišení). I přesto si obraz zachovává věrné barvy a odhaluje struktury na levé části obrazu. Porovnejme si tento výsledek s RAW fotkou upravenou v 6.2, výsledek naší metodou má mnohem přirozenější barvy. Také jsme na tento obrázek neaplikovali žádnou lokální redukci šumu, která by jej mohla ještě dále zlepšit. Z tohoto srovnání pozorujeme, že i z jediného snímku lze získat přijatelné výsledky, tato metoda by tedy mohla najít uplatnění například i ve zpracování videa, kde nemáme možnost kalibrovat každý snímek.

6.2. VLASTNÍ IMPLEMENTACE



(a) Použití `rgbHDR()` na jediný snímek.



(b) Použití `rgbHDR()` na zkalibrovaný snímek.

Obrázek 6.9: Srovnání přínosu kalibrace.

7. Závěr

Tato práce se zabývala problematikou vizualizace obrazů s vysokým dynamickým rozsahem. Byl v ní popsán kompletní proces, jak s takovými obrazy lze pracovat, od získávání dat, přes jejich kalibraci, až po samotnou vizualizaci. Teoretická část dále představuje matematický aparát Fourierovy transformace a konvoluce, pomocí kterého lze provádět lineární filtraci obrazu. Tento aparát dokáže rozdělit obraz do jednotlivých frekvencí. Tohoto využívá metoda neostré masky, která zvýrazňuje vysoké frekvence, které obsahují detaily vnímané okem, a potlačuje nízké frekvence, čímž dokáže efektivně snížit potřebný dynamický rozsah. Tato metoda, která respektuje vnímání obrazu lidským okem, dokáže vytvořit přirozenější vizualizaci HDR obrazů. V praktické části je metoda implementována jako program napsaný v C++ s využitím knihovny OpenCV. V práci jsou demonstrace situací, kdy se nelze obejít bez této metody, chceme-li zachovat detaily ve všech částech obrazu. V příloze lze nalézt tento program spolu s vytvořenými testovacími 32bitovými obrázky. Program lze velmi dobře dále rozšiřovat, jak DeepSkyStacker, tak OpenCV jsou napsané v C++ a jejich kód je otevřený, lze proto například sjednotit proces kalibrace a zpracování obrazu do jednoho programu s vysokou mírou optimalizace, kterou jazyk umožňuje.

Literatura

- [1] ASADZADEH, M.: *Lecture Notes in Fourier Analysis* [online]. [cit 1.5.2021, 2010]. Dostupné z http://www.math.chalmers.se/~mohammad/teaching/Fourier/LectureNotes_A3/draft_1.pdf
- [2] BEZVODA, V., et al.: *Dvojměrná diskrétní Fourierova transformace a její použití –I.: Teorie a obecné užití*. První vydání. Praha: Státní pedagogické nakladatelství, n. p., 1988. 181 p. ISBN 17-135-88.
- [3] COOLEY J. W. a TUKEY, J.: An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, volume 19. 1965. Strany 297–301. Dostupné z: <http://www.ams.org/journals/mcom/1965-19-090/S0025-5718-1965-0178586-1/S0025-5718-1965-0178586-1.pdf>.
- [4] DRUCKMÜLLEROVÁ, H.: *Phase correlation: The mathematical background and application to image registration*. Brno: LAP LAMBERT Academic Publishing, 2011. ISBN 3846550914.
- [5] *DeepSkyStacker*, [online]. Dostupné z: <http://deepskystacker.free.fr/english/index.html>
- [6] HAMEY, L. G. C.: A Functional Approach to Border Handling in Image Processing, 2015 *International Conference on Digital Image Computing: Techniques and Applications*. DICTA, 2015, doi:10.1109/DICTA.2015.7371214.
- [7] IAS, Park City Mathematics Institute: *An Introduction to Mathematical Image Processing*. [online]. 2010 [Cit. 12.5.2021]. Dostupné z http://www.cse.iitm.ac.in/~vplab/courses/optimization/Maths_IM_DEBLUR_ENH_SD_EDT_2016.pdf.
- [8] INDYK, P. a KAPRALOV, M.: *Sample-Optimal Fourier Sampling in Any Constant Dimension*, 2014. [Online]. Dostupné z: <https://groups.csail.mit.edu/netmit/sFFT/ft-hd-part1.pdf>.
- [9] JACOBS, A.: *Open window with armchair and manequin* [foto]. 2006. In: Wikipedia [online]. [Cit. 16.5.2021]. Dostupné z: [https://en.wikipedia.org/wiki/High_Dynamic_Range_\(photography_technique\)](https://en.wikipedia.org/wiki/High_Dynamic_Range_(photography_technique))
- [10] LERCH, A.: *An Introduction to Audio Content Analysis. Applications in Signal Processing and Music Informatics*. Wiley-IEEE Press 2012. strany 181-183. ISBN 9781118266823. Dostupné z <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118393550.app1>.
- [11] McCOY, K.: *The Jefferson National Expansion Memorial, including the Gateway Arch and Old Courthouse*, in St Louis, MO, USA [foto]. St Louis, 2008. In: Wikipedia [online]. [Cit. 16.5.2021]. Dostupné z: [https://en.wikipedia.org/wiki/High_Dynamic_Range_\(photography_technique\)](https://en.wikipedia.org/wiki/High_Dynamic_Range_(photography_technique))
- [12] McNULTY, I.: The future of X-ray holography. *Nucl. Instr. and Meth. In Phys. Res. A*, 347, 1994, p.170-176. ISSN 0168-9002. Dostupné z [https://doi.org/10.1016/0168-9002\(94\)91874-0](https://doi.org/10.1016/0168-9002(94)91874-0)

- [13] KIMPTE, T. a TUYTSCHAEVER, T.: Journal of digital imaging. *Increasing the number of gray shades in medical display systems—how much is enough?*. Dostupné z <https://doi.org/10.1007/s10278-006-1052-3>
- [14] MINOIA, A. [Foto]. In: Nightskypix [online]. [Cit. 13.5.2021]. Dostupné z: <https://nightskypix.com/calibration-frames/>.
- [15] PLOTKE, M. [Foto], 2013. [online][Cit. 12.5.2021]. Dostupné z [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)).
- [16] STEIN, E. M. a WEISS, G.: *Introduction to Fourier Analysis on Euclidean Spaces*. První vydání. Princeton (New Jersey, U.S.A.): Princeton University Press, 1971. 297 stran. ISBN 0-691-08078-X.
- [17] VANÍČKOVÁ, L.: *Matematické modely v biometrii otisků lidských prstů*. [Diplomová práce.] Brno, MUNI, PF, 2017. 70 s.
- [18] VYVLEČKA, M.: *Charakterizace spektrografu s CCD kamerou*. 2015. Bakalářská práce. Univerzita Karlova, Matematicko-fyzikální fakulta, Katedra chemické fyziky a optiky. Vedoucí práce Němec, Petr.
- [19] WALACH, J.: *Aplikace Fourierovy analýzy na rozpoznávání kvality samohlásek podle jejich formantů*. [Diplomová práce.] Olomouc: UPOL, 2013. 45 s.

8. Seznam příloh

- Aplikace - řešení ve VisualStudiu. Je potřeba nalinkovat knihovnu OpenCV.
- Zdrojové obrázky "apple.tif", "kitchen.tif" a "windowsunset.tif". Jde o kalibrované 32bitové obrázky s redukováním rozlišením.