



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY
FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

NUMERICKÉ METODY REGISTRACE OBRAZŮ S VYUŽITÍM NE- LINEÁRNÍ GEOMETRICKÉ TRANSFORMACE

NUMERICAL METHODS OF IMAGE REGISTRATION USING NONLINEAR GEOMETRIC TRANS-
FORMATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAEL RÁRA

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. PAVEL ŠTARHA, Ph.D.

BRNO 2019

Abstrakt

Cílem práce je pomocí vhodných matematických metod vytvořit jednoduchou aplikaci, která bude schopna zpracovat sadu digitálních obrazů postížených atmosferickým seeingem, takovým způsobem, aby se výstupní data podobala co nejvíce skutečnosti. Dalším výstupem budou mapy posuvů jednotlivých dat vzhledem k jejich zprůměrovanému obrazu.

Abstract

The goal of the thesis is creating simple software to modify entry data defected by atmospheric seeing and provide an output image, which is as much close to reality as possible. Another output is a group of images illustrating the move of every input image due to the average image of them.

klíčová slova

Atmosferický seeing, digitální obraz, digitální šum, interpolace, konvoluce, vyhlazování obrazu.

keywords

Astronomical seeing, digital image, digital noise, interpolation, convolution, image smoothing.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně dle pokynů vedoucího a za použití uvedené literatury.

Bc. Michael Rára

Děkuji panu doc. Ing. Pavlu Štarhovi, Ph.D., za vstřícný přístup, rady, poskytnutí literatury a knihovny *OpenTIFF*.

Bc. Michael Rára

Obsah

1	Lom světla	13
1.1	Atmosferický seeing	14
2	Problematika digitálního obrazu	16
2.1	Obrazová matice	16
2.2	Metrické a topologické vlastnosti obrazu	17
3	Reprezentace barev v počítačové grafice	23
3.1	Barevné prostory	23
3.1.1	High colour	24
3.1.2	True colour	24
4	Digitalizace signálu	24
4.1	Kvantování	25
4.2	Vzorkování	26
4.3	Fourierův obraz	26
4.4	Alias a antialiasing	29
4.5	Konvoluce	29
5	Digitální šum	32
5.1	Impulsní šum a způsoby jeho filtrace	32
5.2	Aditivní šum a způsoby jeho filtrace	34
5.3	Lokální vyhlazování obrazu	34
5.3.1	Lineární metody vyhlazování	35
5.3.2	Nelineární metody vyhlazování	37
6	Hledání hran	38
7	Zaostření obrazu	40
8	Vybrané matematické metody pro zpracování obrazu	41
8.1	Korelace digitálního obrazu	41
8.2	Interpolace	42
8.2.1	Lineární interpolace	42
8.2.2	Lineární interpolace funkcí dvou proměnných	43
8.2.3	Bilineární interpolace	45
9	Realizace tvorby softwaru pro zpracování dat postižených atmosferickým seeingem	45
9.1	Vstupní data	46
9.2	Postup řešení	46
9.3	Výstupní data	51
9.4	Statistické vyhodnocení posuvů	52
10	Popsání aplikace	53
10.1	Vývojový diagram softwaru.	57
10.2	Algoritmus pro vykreslení posuvů na černé pozadí	58

Přehled použitých symbolů

v_i – rychlost šíření světla v prostředí i .

n_i – index lomu světla v prostředí i .

c – rychlost světla ve vakuu.

I – spojitá dvojrozměrná obrazová funkce.

f_o – obrazová matice.

\mathbb{R} – množina reálných čísel.

\mathbb{Z} – množina celých čísel.

\mathbb{Z}_0^+ – množina kladných celých čísel s nulou (analogicky pro ostatní číselné množiny).

ρ – eukleidovská metrika.

ψ – neprázdná množina prvků.

(ψ, ρ) – metrický prostor.

ρ_M – manhattanská metrika.

ρ_S – šachovnicová metrika.

$N_4(\mathbf{x})$ – množina čtyřsousedných pixelů kolem pixelu \mathbf{x} .

$N_8(\mathbf{x})$ – množina osmisousedných pixelů kolem pixelu \mathbf{x} .

$N_D(\mathbf{x})$ – množina diagonálně sousedných pixelů kolem pixelu \mathbf{x} .

Υ – množina odstínů šedi.

Υ_1 – podmnožina Υ .

$\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$ digitální obrazy.

$a_{i,j}$ – pixel na souřadnicích i, j náležící obrazu A (analogicky pro obrazy značené jinými písmeny).

\emptyset – prázdná množina.

Ω – oblast v obraze.

Ω' – množinový doplněk k Ω .

Θ – pozadí obrazu.

Γ – množina děr v obraze.

Φ_i – podmnožina obrazu získaná segmentací obrazu.

t – čas.

T_p – délka periody.

ξ – funkce popisující antiperiodický signál.

Δt – časový interval.

u_t – vzorkovací frekvence.

$r(t)$ – funkce popisující obecný analogový signál.

$\tilde{r}(t)$ – diskrétní funkce získaná vzorkováním $r(t)$.

E – neprázdná množina.

S – σ -algebra.

(S, E) – měřitelný prostor.

$L^p(E)$ – prostor funkcí absolutně integrovatelných na E v p -té mocnině. Bývá uvedeno i bez množiny E např. L^p .

$|F|$ – mohutnost množiny F .

\aleph_0 – mohutnost množiny přirozených čísel.

$l^p(F)$ – prostor posloupností absolutně sumovatelných na $F \subset \mathbb{R}$ v p -té mocnině. Bývá uvedeno i bez množiny F např. l^p .

i – imaginární jednotka.

$G(u)$ – je Fourierovým obrazem funkce $g(x)$ (analogicky pro další písmena použitá k označení funkcí).

$g(x) \Leftrightarrow G(u)$ – funkce, které jsou si vzájemně přiřazeny Fourierovou dopřednou a zpětnou transformací.

$\tilde{g}(x)$ – diskrétní funkce.

\mathcal{F} – diskrétní dopředná Fourierova transformace.

\mathcal{F}^{-1} – diskrétní zpětná Fourierova transformace.

$\|G(u)\|$ – amplitudové spektrum funkce $G(u)$.

$Re^2(u)$ – reálná část Fourierova obrazu.

$Im^2(u)$ – imaginární část Fourierova obrazu.

$\phi(u)$ – fázové spektrum.

$arctg$ – funkce arkus tangens.

u_{max} – maximální frekvence.

$*$ – symbol konvoluce.

h – konvoluční jádro.

$f'_o(x, y)$ – obrazová matice získaná konvolucí.

$\delta(t)$ – Diracova funkce.

$s(t)$ – vzorkovací funkce.

\mathbf{A}^* , \mathbf{B}^* , ... – dokonalý digitální obraz.

$Z \sim A(p)$ – náhodná veličina s alternativním rozložením s parametrem p .

ε – konstanta udávající sílu kritéria.

$\check{c}_{i,j}$ – charakteristika polohy statistického souboru hodnot jasů v okolí pixelu $c_{i,j}$.

X, Y, N – náhodné veličiny.

σ – směrodatná odchylka.

\bar{x} – aritmetický průměr.

\mathbf{g}'_n – posloupnost upravených pixelů.

O – okolí pixelu.

\hat{p} – výsledek lineární kombinace hodnot jasů.

∇f – gradient funkce f .

∂ – parciální derivace.

$\nabla^2 f$ – Laplaceův operátor.

r – korelace mezi dvěma soubory dat.

$\bar{\mathbf{A}}^+$ – aritmetický průměr vstupních obrazů \mathbf{A}_i^+ , kde $i = 1, 2, 3, \dots$

\mathbf{A}_1^+ – obraz s potlačenými geometrickými deformacemi.

\mathbf{A}^{++} – výstupní obraz získaný pomocí vytvořeného softwaru.

Úvod

Obecně je známo, že pozorování vesmírných těles z povrchu planety, a tedy přes zemskou atmosféru, s sebou přináší řadu rozmanitých problémů, počínaje vysokou oblačností, která může zhatit plány mnoha astronomů, až k problematice ohýbání trajektorie světelných paprsků procházejících skrz atmosféru. Tyto problémy se dají samozřejmě vyřešit vysláním družice na oběžnou dráhu planety, ale jde o finančně velmi nákladné řešení.

V této práci se budeme zabývat problémem zpracování digitálních obrazů postížených atmosferickým seeingem, tedy jevem zemské atmosféry, který způsobuje geometrické deformace obrazu. Hned v první kapitole tento jev podrobně popíšeme a připomeneme i základní problematiku lomu světla, která je jeho součástí. Úkolem je vytvoření softwaru, který bude schopný zpracovat data zatížená atmosferickým seeingem a přiřadit ke vstupní množině dat jediný snímek znázorňující, jak by obraz měl vypadat, kdyby nebylo atmosferického seeingu. Dalšími důležitými výstupy jsou mapy posuvů, mapy seeingu a textový soubor *Posuv*, který obsahuje hodnoty posuvů každého pixelu v každém obrazu.

Následně se budeme věnovat problematice digitálního obrazu a definujeme pojem obrazová matice a další metrické a topologické vlastnosti obrazu potřebné pro jednoznačné pochopení problematiky zpracování digitálního obrazu. Poznamenejme, že některé pojmy, jako třeba jas, budeme chápat ve významu, jak jej známe z běžného života, neboť přesná definice těchto pojmů je komplikovaná. Zvědavého čtenáře odkazujeme na [6], kde je tato problematika světla a jeho vlastností kvalitně zpracována.

Další část práce se věnuje procesu digitalizace signálu, v níž stojí za pozornost zejména podkapitola zabývající se konvolucí. Za ní následuje rozsáhlá a důležitá kapitola o digitálním šumu, kde se věnujeme zejména šumu impulsnímu a aditivnímu, ale i metodám jejich filtrace.

Následují tři kratší kapitoly o detekci hran, zaostřování obrazu, metodě výpočtu fázové korelace a lineární, bilineární interpolaci, včetně interpolace funkce dvou proměnných pomocí interpolačních rovin.

Po této matematické části konečně čtenáře seznámíme se samotným algoritmem řešení zadaného problému, ukázkou vstupních dat a vyhodnocením získaných výsledků. Poslední kapitola se věnuje popsání okna vytvořené aplikace. Na zadních stránkách práce lze nalézt seznam příloh, které jsou tvořeny vývojovým diagramem algoritmu a algoritmem pro vykreslování map posuvů.

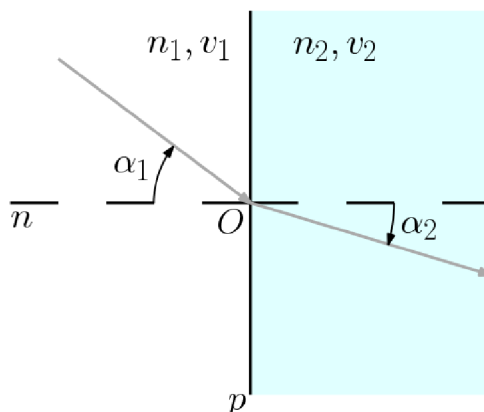
Stručně řečeno, předložená práce má tři části. V první se zabýváme problematikou atmosferického seeingu. Druhá část se zabývá matematickým přístupem k digitálnímu obrazu, a to už od jeho vzniku, tedy digitalizace, až po metody jeho zpracování. Poslední kapitoly se věnují popsání algoritmu řešení zadaného problému a popsání vytvořeného softwaru. Veškerá výstupní i vstupní data aplikace spolu se zdrojovým kódem a exe souborem, umístěným ve složce *Software k diplomove praci*, lze nalézt v elektronické příloze této práce. Získaným výsledkům se podrobně věnuje kapitola 9.

1 Lom světla

Prochází-li světlo hranicí dvou různých optických prostředí, dochází k jeho lomu, který je popsán Snellovým zákonem

$$\frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{v_1}{v_2} = \frac{n_2}{n_1},$$

kde v_1, v_2 jsou rychlosti šíření světla a n_1, n_2 jsou indexy lomu světla v prostředích 1 a 2 viz obrázek 1.



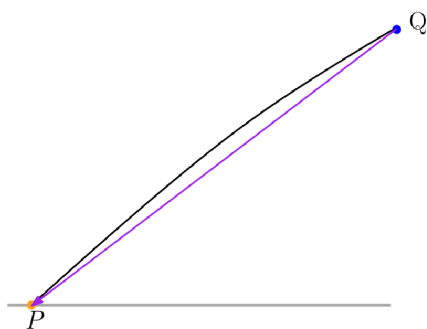
Obrázek 1: Rozhraní optických prostředí. Úhly se vždy měří od kolmice n k hranici prostředí p .

Index lomu světla popisuje zpomalení světla oproti jeho rychlosti ve vakuu a je dán vztahem

$$n = \frac{c}{v},$$

kde c je rychlost světla.

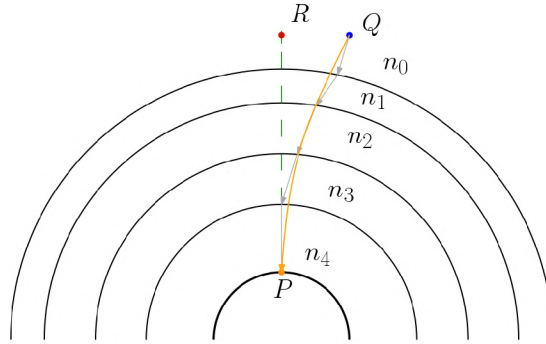
Máme-li prostředí 1 a 2 s indexy lomu $n_1 > n_2$, pak prostředí 1 nazýváme opticky hustším a prostředí 2 opticky řidším. Mění-li se index lomu v prostředí spojitě, pak původně přímá trajektorie světelného paprsku přechází v hladkou křivku. Tento jev je typický pro atmosféru a označuje se pojmem atmosferická refrakce, viz obrázek 2.



Obrázek 2: Atmosferická refrakce. Bod Q je zdrojem světelných paprsků, které vlivem spojitě se měnícího indexu lomu prostředí opisují trajektorii hladké křivky. Při neměnném indexu lomu prostředí by paprsky opisovaly trajektorii fialové úsečky.

Index lomu světla v zemské atmosféře je ovlivněn zejména tlakem¹, který klesá se vzdáleností od povrchu.

Obrázek 3 ilustruje reálný dopad atmosferické refrakce na vnímání polohy vesmírných těles. Paprsek vycházející z bodu Q se při průchodu atmosférou láme a dopadá do oka pozorovatele P pod změněným úhlem, díky čemuž lidské oko a mozek špatně vyhodnotí původní polohu světelného zdroje, neboť si dopadající paprsek prodlouží do bodu R , v němž vidí původní světelný zdroj.



Obrázek 3: Zkreslení polohy světelného zdroje vlivem atmosferické refrakce - obrázek ilustruje lom světla v různých úrovních atmosféry, tj. šedá trajektorie světelného paprsku, jedná se však jen o ilustraci, která má znázorňovat proměnlivost indexu lomu v atmosféře, přičemž $n_0 < n_1 < n_2 < n_3 < n_4$. V reálné situaci se index lomu světla mění plynule a původně přímý paprsek vlivem atmosferické refrakce přechází v hladkou, oranžově zbarvenou křivku, jak je znázorněno obrázkem 2.

Toto byl základní úvod k problematice pozorování vesmírných těles přes atmosféru. V dalším textu popíšeme jev zvaný atmosferický seeing, který vlivem proměnlivosti indexu lomu v čase způsobuje geometrické deformace obrazu.

1.1 Atmosferický seeing

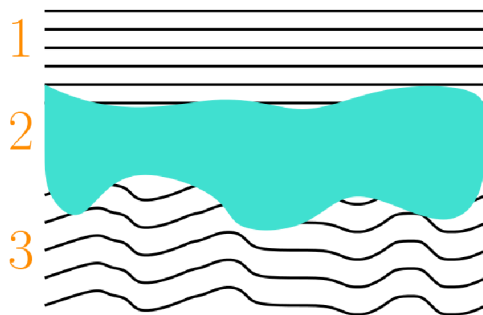
Atmosferický seeing je jedním z hlavních problémů při pozorování vesmírných těles z povrchu planety. Tímto pojmem máme na mysli relativní kvalitu optických vlastností zemské atmosféry. Kvalitní atmosferický seeing se vyznačuje stabilitou a nízkým zkreslením obrazu během pozorování. Nejvyšší seeing se projevuje geometrickými neměnnostmi oblastí v obraze, zatímco nejnižší seeing naopak způsobuje výrazné geometrické deformace a výrazné zkreslení obrazu.

Atmosferický seeing vzniká vlivem rychlého střídání teplot v různých vrstvách atmosféry. Obrázek 4 znázorňuje tento vliv atmosféry (zóna 2) na přicházející světelné vlny (zóna 1). Poslední zóna 3 ilustruje pokrivení světelných vln. Značný negativní vliv na kvalitu seeingu mohou mít i atmosferické aerosoly jako například vodní pára, prach nebo sopečný popel.

Nejčastější důsledek seeingu je rozmazání pozorovaných astronomických objektů nebo drobné změny jejich jasu. Dalším důsledkem, který se projevuje u dlouhých dob expozic, je rozmazání bodových objektů do oblasti zvané seeing disc².

¹Který je podle stavové rovnice plynu závislý na teplotě a obráceně. Lze tedy také mluvit o vlivu teploty vzduchu.

²Český ekvivalent není znám. Lze přeložit jako *kružnice seeingu*.



Obrázek 4: Vliv turbulentní vrstvy atmosféry na přicházející rovinné světelné vlny.

K měření atmosferického seeingu se používá řada různých metod a stupnic, které jej hodnotí podle zvolených kritérií. Tyto stupnice mohou být založeny jen na subjektivním názoru na celkovou kvalitu seeingu, nebo mohou upřednostňovat jediné kritérium. Jednoduchými ukázkami stupnic zohledňujícími subjektivní názor kvality seeingu jsou

The association of Lunar & Planetary Observers Scale³

- 1 Velmi nekvalitní obraz. Nelze rozlišit detaily ani obrys.
- 2–3 Téměř trvale špatný seeing. Příležitostné okamžiky dobrého seeingu.
- 4–6 Stále převládá špatný seeing, který je však prokládán krátkými intervaly kvalitního seeingu.
- 7–8 Převládá kvalitní seeing způsobující jemné zkreslení.
- 9–10 Dokonalý seeing s ustálenými snímky i při výrazném přiblížení obrazu.

The Andoniadi⁴ Scale.

- I. Dokonalý seeing bez chvění obrazu.
- II. Jemné chvění obrazu prostoupené několikavteřinovými momenty dokonalého seeingu.
- III. Průměrná úroveň seeingu s větším chvěním obrazu.
- IV. Špatná kvalita seeingu s neustálým chvěním obrazu.
- V. Vysoce nekvalitní seeing. Hrubé obrysy pozorovaných těles jsou obtížně rozeznatelné.

Existuje mnoho jiných stupnic, které jsou snadno dohledatelné na internetu. Zmíňme například *The Pickering/Douglass „Standard Scal“*, *The See/Cogshall Double Star Scale*, *The Environment Canada Scale*. V dalším textu však žádnou z nich potřebovat nebudeme, naopak si vystačíme se základním pochopením příčin a důsledků atmosferického seeingu. Podrobnější informace lze nalézt například v [10].

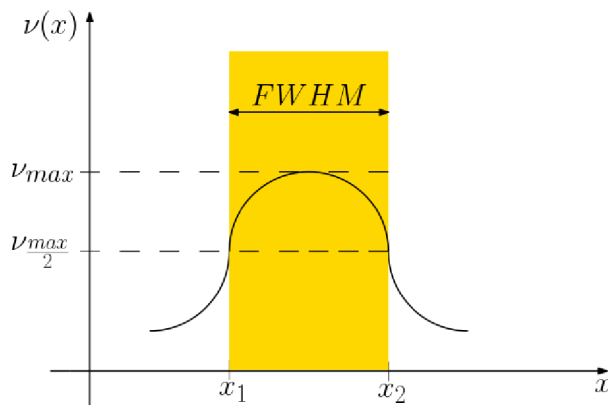
Nejčastějším způsobem k vyhodnocení kvality seeingu je však aplikování metody FWHM⁵ na seeing disc. Výstupem metody FWHM je vzdálenost dvou bodů, jejichž funkční hodnoty, tedy jas, jsou poloviční oproti hodnotě globálního maxima, viz obrázek 5. Aplikujeme-li FWHM na seeing disc, získáme jeho průměr, který je měřítkem kvality seeingu. Tento poloměr je obvykle uváděn v úhlových sekundách⁶. Je zřejmé, že by bylo užitečné mít objektivní kritéria pro rozhodování o kvalitě seeingu a k nim příslušnou stupnici. V pozdější části textu uvidíme, že vytvořený software nám tato kritéria poskytne ve formě aritmetického průměru a rozptylu hodnot posuvů pixelů.

³Často zmiňovaná pod zkratkou *ALPO*.

⁴Eugène-Michel Antoniadi – Řecký astronom. Tato stupnice je tradičně uváděna v římských číslicích.

⁵Full width and half maximum.

⁶Zkráceně arcsec, $1 \text{ arcsec} = \frac{1}{3600} \text{ deg}$.



Obrázek 5: Full width at half maximum.

2 Problematika digitálního obrazu

V této kapitole popíšeme možné způsoby digitální reprezentace obrazu a uvedeme problematiku metrických a topologických vlastností obrazu. Doplňkové informace k této kapitole lze nalézt v [4], [5], [6], [9].

2.1 Obrazová matice

Chceme-li na počítačích digitálně zpracovat obraz, musíme jej nejprve digitalizovat. Jde o proces, který se skládá ze dvou dílčích kroků, a to ze vzorkování a kvantování⁷. Účelem digitalizace obrazu je převod spojité dvojrozměrné informace do dvojrozměrné obrazové matice. Matematicky jej můžeme vyjádřit následovně

$$I(x, y) \xrightarrow{\text{Digitalizace}} f_o(i, j), x, y \in \mathbb{R}, i, j \in \mathbb{Z}_0^+. \quad (2.1)$$

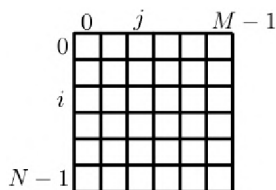
Funkce I je spojitá dvojrozměrná obrazová funkce, jejímž definičním oborem je dvojrozměrná oblast a oborem funkčních hodnot je fotometrická veličina, a sice jas. Symbolem $f_o(i, j)$ označujeme obrazovou matici, jejíž hodnoty jsou celá nezáporná čísla.

Tvar obrazové matice může být libovolný, přesto se v praxi nejčastěji používá čtvercová síť, viz obrázek 6. Mezi další často uváděné sítě⁸ patří síť hexagonální, obdelníková a síť rovnostranných trojúhelníků, viz obrázky 7, 8, 9. Jeden prvek obrazové matice se nazývá pixel⁹, zkráceně px. Později uvedeme, že ačkoliv je čtvercová a obdelníková síť pro práci nejjednodušší, zejména pro možnost popsání polohy každého pixelu pomocí sloupcového a řádkového indexu, přináší s sebou řadu obtíží, a to hlavně v definování topologických pojmů.

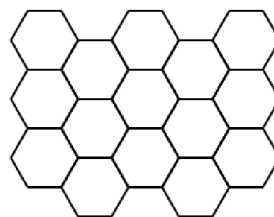
⁷Oba procesy budou podrobně popsány v kapitole 4.

⁸Také se používá termín vzorkovací mřížka.

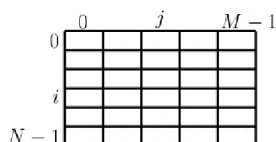
⁹Slovo vzniklé z anglických výrazů **p**icture a **e**lement. Ekvivalentním méně používaným výrazem je pel. Konkrétní podoba pixelu závisí na výstupním zařízení. Zdůrazněme, že na rozdíl od bodu se nejedná o bezrozměrný objekt.



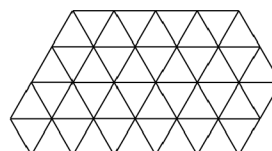
Obrázek 6: Čtvercová síť.



Obrázek 7: Hexagonální síť.



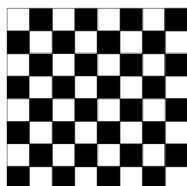
Obrázek 8: Obdélníková síť.



Obrázek 9: Trojúhelníková síť.

Významným procesem digitalizace je kvantování obrazu, při němž se funkční hodnoty obrazové funkce I převádí do diskrétního oboru hodnot obrazové matice. Je-li k reprezentaci informace o obrazovém elementu použito q bitů, je počet úrovní jasu $p = 2^q$. Obrazy, které reprezentují informaci o pixelu jediným bitem, se nazývají binární. Jednoduchým příkladem binárního obrazu je šachovnice, viz obrázek 10. Platí, že zcela černý pixel, tedy pixel bez jasu, je v paměti počítače reprezentován hodnotou $p = 0$. Naopak zcela bílý pixel, tedy pixel s maximální hodnotou jasu, je reprezentován nejvyšším možným číslem, které závisí na bitové hloubce obrazu, viz tabulka 1. Mezilehlé hodnoty odpovídají různým stupňům šedi.

Obrázek 10: Binární obraz – šachovnice.



Bitová hloubka q	Maximální jas ($2^q - 1$)	Poznámka
1	1	Černá a bílá barva
2	3	
4	63	
8	255	Běžně používané

Tabulka 1: Vybrané bitové hloubky.

2.2 Metrické a topologické vlastnosti obrazu

Bylo již řečeno, že vzorkovací mřížky mohou mít různé podoby. Nejčastěji se však používá síť čtvercová, a proto jí budeme dále věnovat zvýšenou pozornost.

Je zřejmé, že některé pojmy, které jsou ve spojité matematice běžně používané, jako třeba okolí, vzdálenost, souvislost, můžeme uvažovat i pro obrazovou matici a pixely.

Nejprve je však nutné tyto termíny definovat tak, aby nedocházelo k nedorozuměním. Začneme s definicí metriky.

Metrika a metrické prostory

Definice 2.1. Uvažujme neprázdnou množinu prvků Ψ a nezápornou funkci $\rho : \Psi \times \Psi \rightarrow \mathbb{R}_0^+$ ve které pro libovolné $\alpha, \beta, \gamma \in \Psi$ platí

1. $\rho(\alpha, \beta) \geq 0, \rho(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$. (axiom totožnosti)
2. $\rho(\alpha, \beta) = \rho(\beta, \alpha)$. (axiom symetrie)
3. $\rho(\alpha, \gamma) \leq \rho(\alpha, \beta) + \rho(\beta, \gamma)$. (trojúhelníková nerovnost)

Potom se funkce ρ nazývá metrika (vzdálenost) a dvojice (Ψ, ρ) metrickým prostorem.

V následujících příkladech budeme uvažovat metriku ρ a množinu $\Psi = \{\alpha = (\alpha_1, \dots, \alpha_n), \beta = (\beta_1, \dots, \beta_n) : \alpha_i, \beta_i \in \mathbb{R}, i = 1, 2, \dots, n\}$. Poznamenejme, že všechny tři níže uvedené metriky tvoří spolu s množinou Ψ metrický prostor.

Příklad 2.2. Nejnázornější metrika známá z všedního života je Euklidovská metrika ρ definovaná rovností

$$\rho(\alpha, \beta) = \sqrt{\sum_{i=1}^n (\alpha_i - \beta_i)^2}.$$

Dvojice (Ψ, ρ) je potom metrickým prostorem, kterému se říká *n-rozměrný euklidovský prostor*. Výhodou euklidovské metriky je její názornost a jednoduchost. Nevýhodou je složitější výpočet odmocniny a možnost neceločíselného výsledku, který neodpovídá celočíselným hodnotám v obrazové matici.

Příklad 2.3. Druhou možnou metrikou ve vzorkovací mřížce je metrika manhattanská¹⁰. Měří nejmenší počet polí, přes která musíme projít při cestě z počátečního do koncového bodu, je-li povolen pouze pohyb¹¹ ve směru vodorovném a svislém. Tato metrika je definovaná následovně

$$\rho_M(\alpha, \beta) = \sum_{i=1}^n |\alpha_i - \beta_i|.$$

Příklad 2.4. Třetí metrikou, kterou má smysl ve vzorkovací mřížce uvažovat je metrika šachovnicová. V této metrice připouštíme pohyb¹² vodorovný, svislý i diagonální. Výsledkem této metriky je nejmenší počet polí, přes která musíme projít při pohybu od počátečního do koncového bodu. Tato metrika je definovaná následovně

$$\rho_S(\alpha, \beta) = \max\{|\alpha_i - \beta_i| : i = 1, 2, \dots, n\}.$$

Metrika je tedy funkce, která podle zvoleného předpisu, splňujícím podmínky uvedené v definici 2.1, určuje vzdálenost dvou bodů. Přejdeme nyní k topologickým vlastnostem pixelů ve vzorkovací mřížce.

¹⁰Také známá jako newyorská metrika nebo vzdálenost v městských blocích.

¹¹Pohyb šachové věže.

¹²Pohyb šachového krále.

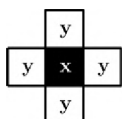
Topologické vlastnosti

Prvním nutným pojmem, který definujeme, je sousednost pixelů.

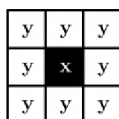
Definice 2.5. Řekneme, že dva pixely \mathbf{x}, \mathbf{y} jsou 4 – *sousedny* $\Leftrightarrow \rho_M(\mathbf{x}, \mathbf{y}) = 1$, viz obrázek 11. Tuto množinu bodů značíme $N_4(\mathbf{x})$ ¹³.

Definice 2.6. Řekneme, že dva pixely \mathbf{x}, \mathbf{y} jsou 8 – *sousedny* $\Leftrightarrow \rho_S(\mathbf{x}, \mathbf{y}) = 1$, viz obrázek 12. Tuto množinu bodů značíme $N_8(\mathbf{x})$ ¹⁴.

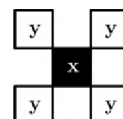
Definice 2.7. Řekneme, že dva pixely jsou diagonálně sousedné¹⁵, pokud patří do množiny $N_D(\mathbf{x}) = N_8(\mathbf{x}) \setminus N_4(\mathbf{x})$, viz obrázek 13.



Obrázek 11



Obrázek 12



Obrázek 13

Relace být sousedem je tedy závislá na zvolené metrice. Dále definujeme nezbytné pojmy pro zavedení důležitých termínů cesta a oblast.

Prvním z těchto pojmů bude spojitelnost pixelů, která je užitečná při určování hranic objektu. K definování tohoto pojmu budeme využívat sousednosti a pro větší názornost konkrétní množinu jednotlivých úrovní šedi $\Upsilon = \{0, 1, 2, 3, \dots, 255\}$ a její podmnožinu $\Upsilon_1 = \{32, 33, 34, \dots, 68\}$, viz tabulka 1.

Definice 2.8. Mají-li pixely \mathbf{x}, \mathbf{y} úrovně šedi v Υ_1 a zároveň platí $\mathbf{y} \in N_4(\mathbf{x})$, pak jsou tyto pixely 4-spojitelné¹⁶.

Definice 2.9. Mají-li pixely \mathbf{x}, \mathbf{y} úrovně šedi v Υ_1 a zároveň platí $\mathbf{y} \in N_8(\mathbf{x})$, pak jsou tyto pixely 8-spojitelné¹⁷.

Definice 2.10. Pixely \mathbf{x}, \mathbf{y} jsou m-spojitelné¹⁸, pokud tyto pixely mají úrovně šedi v Υ_1 a dále platí alespoň jedna z uvedených podmínek

- $\mathbf{y} \in N_4(\mathbf{x})$.
- $\mathbf{y} \in N_D(\mathbf{x}) \wedge N_4(\mathbf{x}) \cap N_4(\mathbf{y}) = \emptyset$.

Poznamenejme, že při m-spojitelosti nedochází k vícenásobným spojením a jedná se o modifikaci 8-spojitelnosti, viz obrázky 14, 15, kde jsou pixely s úrovní šedi patřící do Υ_1 označeny jako 1 a zbylé jako 0.

Definice 2.11. Jsou-li pixely \mathbf{x}, \mathbf{y} spojitelné, pak říkáme, že \mathbf{x} přiléhá k \mathbf{y} nebo opačně.

Je zřejmé, že pro různé spojitosti, lze definovat 4-, 8- nebo m-přiléhavost. Dále platí, že dvě podmnožiny obrazu \mathbf{A} jsou přiléhající, jestliže nějaký pixel v první podmnožině přiléhá k nějakému pixelu ve druhé.

Nyní s pomocí pojmu přiléhavosti definujeme cestu z jednoho pixelu do druhého.

¹³Anglický termín = 4-neighbours, také překládán jako čtyřsousednost.

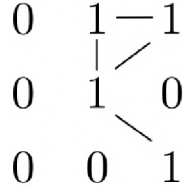
¹⁴Anglický termín = 8-neighbours, také překládán jako osmisousednost.

¹⁵Anglický termín = diagonal neighbours.

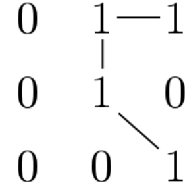
¹⁶Anglický termín = 4-connectivity.

¹⁷Anglický termín = 8-connectivity.

¹⁸Anglický termín = mixed connectivity.



Obrázek 14: 8-spojítost. Převzato z [5].



Obrázek 15: m-spojítost. Převzato z [5].

Definice 2.12. Cestu¹⁹ z pixelu \mathbf{x} do pixelu \mathbf{y} definujeme jako posloupnost pixelů $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, kde $\mathbf{x}_1 = \mathbf{x}, \mathbf{x}_n = \mathbf{y}$ a kde platí, že \mathbf{x}_{i+1} přiléhá²⁰ k $\mathbf{x}_i, i = 1, \dots, n - 1$, kde n je tzv. délka cesty.

Definice 2.13. Oblastí ve vzorkovací mřížce rozumíme množinu pixelů, mezi jejímiž každými dvěma body existuje cesta, která celá patří do oblasti.

Definice 2.14. Pokud mezi dvěma pixely v obraze existuje cesta, pak je nazýváme souvislými pixely.

Oblast lze také definovat jako množinu navzájem souvislých pixelů. Je zřejmé, že relace *být souvislý* je relací ekvivalence²¹, a proto určuje rozklad množiny, tj. digitálního obrazu, na třídy ekvivalence, v našem případě na oblasti.

Uvažujme digitální obraz na obrázku 16 a v něm manhattanskou metriku²². Oblasti $\Omega_i, i = 1, 2, 3, 4$ jsou tedy mezi sebou nesouvislé a představují třídy ekvivalence relace *být souvislý*. Dále zavedme oblast $\Omega = \bigcup_{i=1}^4 \Omega_i$ a Ω' , která je množinovým doplňkem oblasti Ω k obrazu. Platí, že $\Omega' = \Theta \cup \Gamma$, kde množina Θ , souvislá s okrajem obrazu, se označuje jako *pozadí* a Γ je tzv. *množina děr*. Poznamenejme, že množiny Θ a Γ mohou být složeny z vícero podmnožin, které jsou však navzájem disjunktní, tj.

$$\Theta = \bigcup_{i=1}^m \Theta_i; \Theta_i \cap \Theta_j = \emptyset, i \neq j.$$

Analogicky pro Γ , kde Γ_i jsou jednotlivé díry.

Definice 2.15. Oblast s k dírami se nazývá $k + 1$ -násobně souvislá oblast²³.

Některé vybrané oblasti obrazu se často označují jako objekty, například osoby na fotce. Proces, který obraz člení na jednotlivé objekty, se nazývá segmentace obrazu.

Definice 2.16. Segmentací obrazu rozumíme rozdělení obrazové matice $f_o(i, j)$ na množinu neprázdných podmnožin $\{\Phi_i\}_{i=1}^n$, pro kterou platí

1. $\bigcup_{i=1}^n \Phi_i = f_o(i, j)$.
2. $\Phi_i \cap \Phi_j = \emptyset, i \neq j$.
3. Každá podmnožina Φ_i má danou vlastnost.

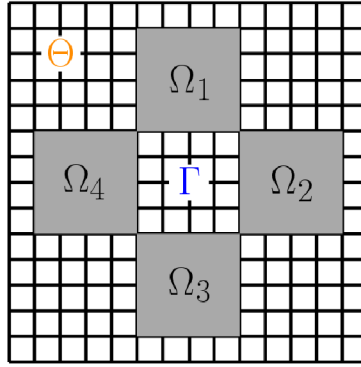
¹⁹Podle typu zvolené přiléhavosti, lze definovat 4-, 8- nebo m-násobnou cestu

²⁰S ohledem na zvolenou sousednost.

²¹Definici relace ekvivalence a pojmů s ní spojených lze nalézt v [7].

²²Pro případ šachové metriky by v obraze nebyla díra a oblasti Ω_i by byly spojitě.

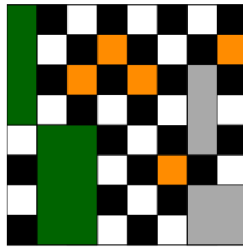
²³Pokud oblast díru neobsahuje, pak se používá označení *jednoduše souvislá oblast*.



Obrázek 16: Pozadí.

Z první a druhé podmínky vyplývá, že množina $\{\Phi_i\}_{i=1}^n$ tvoří rozklad množiny $f(i, j)$ podle jediné relace ekvivalence. Ono kritérium relace ekvivalence je obsaženo právě v poslední podmínce. Může jím být například vztah *stejná hodnota jasu pixelu*.

Názorný příklad segmentace obrazu je znázorněn na obrázku 17. Pro relaci *být stejné barvy* lze obraz rozčlenit na pět podmnožin obsahujících pixely stejné barvy. Je zřejmé, že tyto množiny splňují podmínky 1, 2.



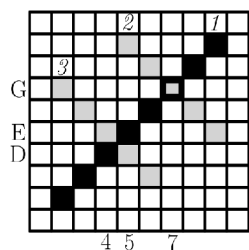
Obrázek 17: Segmentace obrazu.

Již dříve bylo řečeno, že čtvercová vzorkovací mřížka je pro zpracování obrazu nej-jednodušší, na druhou stranu jsou s ní spojené problémy v definování matematických pojmů. První paradox můžeme vidět na obrázku 18. Při uvažování manhattanské metriky jsou úsečky 1, 2, 3 v každém pixelu nesouvislé a pro průnik mohou nastat dvě situace. První možnost znázorňuje průnik úsečky 1, 2, tj. bod $[7, G]$. Druhá možnost průniku dvou úseček, tentokrát 1, 3, je znázorněna na souřadnicích $[4, D]$, $[4, E]$, $[5, D]$, $[5, E]$. Platí tedy, že průnikem dvou různoběžných přímk ve čtvercové síti může být prázdná množina.

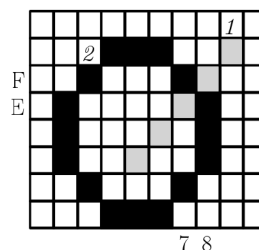
Další problém čtvercové sítě je znázorněn na obrázku 19. Zatímco v euklidovské geometrii dělí uzavřená křivka rovinu na dvě nesouvislé množiny, tak ve čtvercové síti, kde uvažujeme šachovou metriku, toto tvrzení nemusí platit. Z obrázku 19 vyplývá, že z vnitřku kružnice lze vést přímku do jejího vnějšího okolí bez toho, aniž by tato přímka kružnici protla. Mezi vnitřkem a vnějškem kružnice tedy existuje cesta, čili vnitřek i vnějšek kružnice tvoří jedinou oblast.

Je tedy nutné si uvědomit, že při použití čtvercové vzorkovací mřížky může docházet k různým paradoxům a pojmy, které jsou definované v problematice spojitě matematiky, je nutné vhodně upravit tak, aby se s nimi dalo pracovat i v diskrétních oblastech.

Posledními pojmy, které pro úplnost textu zavedeme, budou hranice a konvexní obal oblasti.



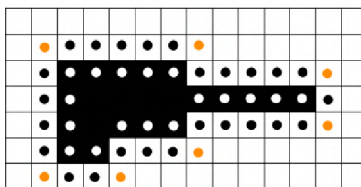
Obrázek 18: Paradox diskrétní úsečky.



Obrázek 19: Paradox diskrétní kružnice.

Definice 2.17. Vnitřní hranicí²⁴ oblasti rozumíme množinu pixelů v ní ležících, takových, že každý z nich má alespoň jednoho souseda, který do této oblasti nepatří, viz obrázek 20.

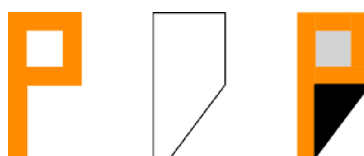
Definice 2.18. Vnější hranicí²⁵ oblasti rozumíme vnitřní hranici pozadí této oblasti.



Obrázek 20: Vnitřní a vnější hranice oblasti. Převzato z [4].

Definice 2.19. Konvexní obal, viz obrázek 21, objektu je taková množina, která je nejmenší nadmnožinou celého objektu, v níž každé dva body objektu lze spojit úsečkou, která celá leží v této nadmnožině.

Definice 2.20. Deficit²⁶ konvexního obalu je množina pixelů, které do objektu nepatří.



Obrázek 21: Konvexní obal. Převzato z [4].

Definovali jsme tedy základní vlastnosti digitálního obrazu.

Aritmetické a množinové operace mezi pixely

Základní aritmetické operace mezi pixely jsou stejné jako na množině reálných čísel. Hodnotami, s nimiž pracujeme jsou, vlastnosti pixelů, např. jas.

²⁴Pixely s bílou tečkou na obrázku 20.

²⁵Uvažujeme-li manhattanskou metriku, pak vnější hranici tvoří jen pixely s černou tečkou na obrázku 20. Pro případ šachové metriky tvoří vnější hranici černé i oranžové tečky.

²⁶Šedá oblast, tzv. jezero, a černá oblast, tzv. záliv, na obrázku 21.

Příklad 2.21. Necht' p_1 je hodnota jasu pixelu \mathbf{x} . Přičtením (resp. odečtením) hodnoty jasu p_2 změním původní jas pixelu \mathbf{x} na hodnotu $p_3 = p_1 + p_2$ (resp. $p_3 = p_1 - p_2$). Je však třeba mít na paměti následující

pro bitovou hloubku q , hodnoty jasu pixelu nabývají hodnot $\{0, \dots, 2^q - 1\}$, viz tabulka 1. Může tedy dojít k tzv. přetečení, kterému lze zamezit následujícími podmínkami.

- $p_1 + p_2 > 2^q - 1 \Rightarrow p_3 := 2^q - 1$ (analogicky pro násobení).
- $p_1 - p_2 < 0 \Rightarrow p_3 := 0$.

Zdůrazněme, že hodnoty jasu pixelu jsou celočíselné, a proto je případně nutné výsledky zaokrouhlit na celočíselný tvar.

Zavedli jsme tedy základní definice, které budeme spolu s vybranými poznatky z této kapitoly v dalším textu průběžně využívat.

3 Reprezentace barev v počítačové grafice

V této kapitole popíšeme základy reprezentace barev v počítačové grafice. Upozorníme, že pojmy jako barva, jas, intenzita jasu atd. budeme dále v textu chápat jen na intuitivní úrovni ve významu jak jej známe z běžného života, neboť problematika přesné definice je nad rámec tohoto textu. Podrobnější a kvalitní popis těchto pojmů lze nalézt v [6].

V počítačové grafice existuje vícero způsobů reprezentace barev. Nejjednodušším z nich je pomocí barevného modelu RGB, který každou barvu vytváří pomocí kombinace červené, zelené a modré barvy, přičemž každá z těchto barev nabývá hodnot ležících v množině $\{0, 1, \dots, 255\}$. Tuto trojici používáme z důvodu přítomnosti receptorů v lidském oku, které nejvíce vnímají zelené světlo, pak červené a nejméně modré. To je také důvodem používání koeficientů v rovnici 3.1. Jde o tzv. aditivní skládání barev. Tuto kombinaci lze zapsat pomocí vektoru (R, G, B) . Nulová hodnota značí, že příslušná barva v kombinaci není zahrnuta, naopak maximální hodnota indikuje, že příslušná barevná složka nabývá své největší intenzity.

Celkový jas barevného pixelu je vyjádřen rovnicí

$$Jas = 0,299R + 0,587G + 0,114B. \quad (3.1)$$

3.1 Barevné prostory

V předchozím textu lze nalézt tabulku 1, která stručně shrnuje základní informace o možných bitových hloubkách pro nebarevné obrazy. V této kapitole se budeme věnovat barevným prostorům *high colour* a *true colour*. Stejně jako u nebarevných obrazů, tak i v případě barevného obrazu určuje bitová hloubka počet bitů nutných pro reprezentaci barvy. Tímto číslem tedy můžeme vyjádřit pestrost barevné škály.

Bitová hloubka q	Počet možných barev ($2^q - 1$)	Název barevného prostoru
16	65 536	High colour
24	16 777 216	True colour

Tabulka 2: Vybrané bitové hloubky pro barevné prostory.

3.1.1 High colour

Než přejdeme k samotné kategorii *high colour*, seznámíme se nejprve s barevným prostorem využívajícím 15-bitovou hloubku. Tento barevný prostor pracuje s červenou, zelenou a modrou barvou, přičemž každému z těchto kanálů náleží 5 bitů, což znamená, že jsme schopni znázornit 32 odstínů každé jednotlivé komponenty a celkem 32 768 barev.

Přidáme-li předchozímu modelu jeden bit, dostaneme tzv. *high colour* model. Tento přidaný bit je většinou věnován zelenému barevnému kanálu, protože lidské oko tuto barvu vnímá nejvíce. Toto nám umožní znázornit 64 odstínů zelené a celkem 65 536 barev.

3.1.2 True colour

Moderní zobrazovací zařízení pracují s barevným prostorem *true color*, který umožňuje znázornit až 16 777 216 barev. Každému z barevných kanálů RGB je věnováno 8 bitů. Rozšíření tohoto prostoru se označuje jako *TrueColour RGBA*, která pracuje se 32 bity a přidaná složka *a* určuje průhlednost daného pixelu. Pomocí *true colour* můžeme vyjádřit mnohem více barev, než jich je schopné lidské oko rozlišit.

Následující obrázek 22 graficky znázorňuje barvy vytvořitelné RGB modelem.



Obrázek 22: Grafické znázornění prostoru *true colour*. Převzato z https://color.adobe.com/build01/resource/img/kuler/color_wheel.webp dne 12.3.2019.

4 Digitalizace signálu

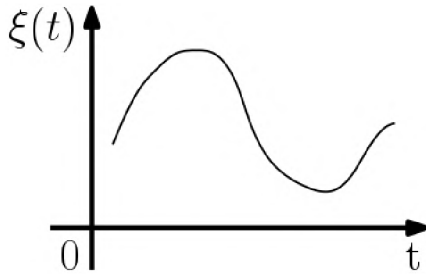
Zpracování signálu je rozsáhlou tematikou, která přesahuje zadání této práce, proto dále uvedeme jen nejnútnejší teorii. Podrobnosti lze nalézt například v [2], [3], [4], [5], [8], [9].

Jak již bylo řečeno v kapitole 2, digitalizace je proces, jehož účelem je převést spojité signály do diskretní podoby, viz rovnice 2.1. Samotné signály pak lze matematicky rozlišit na *signály deterministické* a *signály náhodné (stochastické)*. Deterministické signály dále dělíme na

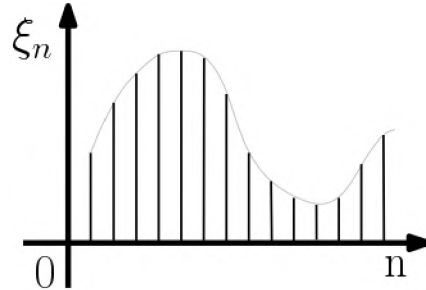
- *Periodické* – dále se dělí na signály *antiperiodické* a *multitónové*. Antiperiodické signály lze popsat funkcí $\xi(t + \frac{T_p}{2}) = -\xi(t)$, kde t obvykle značí čas a T_p délku periody.
- *Kvaziperiodické* – tyto signály jsou dány součtem harmonických signálů s frekvencemi, které nejsou celočíselným násobkem jediné frekvence.

- *Přechodné* – periodicky se neopakují, mají spojité frekvenční spektrum a v reálném případě trvají konečnou dobu (teoreticky i nekonečně dlouho).

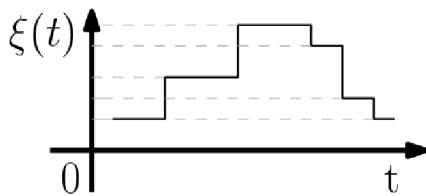
Platí, že všechny deterministické signály můžeme popsat matematicky, a to buď spojitou funkcí, obrázky 23, 25, nebo posloupností, obrázky, 24, 26.



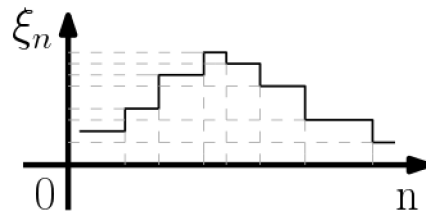
Obrázek 23: Analogový signál.



Obrázek 24: Diskrétní signál.



Obrázek 25: Víceúrovňový signál.



Obrázek 26: Vzorkované číslicové signály.

Chceme-li spojité signály, tj. analogové, a víceúrovňové, převést na diskrétní signály, tj. obrázky 24, 26, musíme provést tzv. *vzorkování* a splnit předpoklady vzorkovací věty²⁷. Nakonec poznamenejme, že náhodné signály lze popsat pouze statisticky a nebudeme se jimi dále zabývat. Další informace o této skupině signálů lze nalézt v [4], odkud byly obrázky 23, 25, 24, 26 převzaty.

Z kapitoly 2 víme, že samotná digitalizace se skládá ze dvou navazujících dílčích kroků a sice z *kvantování* a *vzorkování*. Oba procesy nyní podrobně rozebereme.

4.1 Kvantování

Během kvantování dochází k diskretizaci oboru hodnot funkce $I(x, y)$. Tento proces lze dále dělit na dvě podkategorie, a to kvantování *uniformní* a *neuniformní*. První podkategorie rozděluje obor hodnot funkce $I(x, y)$ na intervaly konstantní délky, zatímco druhá, méně používaná, nikoliv.

Hodnotou, kterou přiřazujeme jednotlivým intervalům oboru hodnot funkce $I(x, y)$, může být například medián, aritmetický nebo vážený průměr atd. z hodnot daného intervalu. Obecně platí, že kvantování může způsobit skokový přechod hodnot jasu, který se projevuje jako nově vytvořená hrana. Této chybě lze částečně předejít neuniformním kvantováním s vhodnou metodou výběru zástupné hodnoty intervalů.

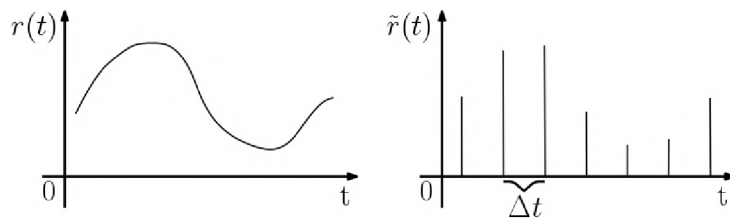
²⁷V podkapitole 4.2 oboje popíšeme.

4.2 Vzorkování

Znovu uvažujme spojitou funkci $I(x, y)$. Vzorkováním této funkce myslíme odečítání hodnot ve stejných intervalech, z kterých následně sestavíme obrazovou matici $f_o(i, j)$. Každou z odečtených hodnot pak přiřadíme pixelu, jehož střed je určen právě souřadnicemi i, j . Tento přístup vzorkování se označuje jako *bodové vzorkování*²⁸. Dalším příkladem vzorkování je *plošné vzorkování*, které je podrobně popsáno v [9].

Uvažujme nyní jednorozměrný signál popsany funkcí $r(t)$, z něhož budeme v daných časových intervalech odečítat hodnoty, jak je znázorněno obrázkem 27. Vzdálenost tohoto intervalu označíme Δt a vzorkovací frekvenci získáme podílem

$$u_t = \frac{1}{\Delta t}. \quad (4.1)$$



Obrázek 27: Vzorkování analogového signálu. Převzato z [9].

V následujícím textu budeme zkoumat otázku, zda je možné zvolit takovou vzorkovací frekvenci, aby signál spojitý v čase byl jednoznačně popsán posloupností vzorků odečítaných s touto frekvencí. K vyřešení této otázky nejprve uvedeme nezbytné pojmy a poznatky z Fourierovy analýzy.

4.3 Fourierův obraz

Dosud jsme používali pouze jednu reprezentaci obrazu, a sice diskrétní matici pixelů. Tuto reprezentaci budeme dále označovat pojmem *prostorová oblast*²⁹. Reprezentaci, která k popisu obrazu využívá obecně nekonečně mnoho sinusových signálů s různou amplitudou a fázovým posunem, nazýváme *Fourierovým obrazem* původního obrazu v tzv. *frekvenční oblasti*.

Pro názornost budeme dále uvažovat situace v jednorozměrném případě. Poznamenejme, že v této podkapitole budeme písmeno i používat pro označení imaginární jednotky.

Definice 4.1. Necht' E je neprázdná množina a $S \subset E$. Systém množin S nazýváme σ -algebrou, jestliže platí

- $\emptyset \in S$,
- $A \in S \Rightarrow E \setminus A \in S$,
- $A_1, A_2, A_3, \dots \in S \Rightarrow \bigcup_{j=1}^{\infty} A_j \in S$.

Definice 4.2. Necht' S je σ -algebrou na E , pak dvojici (S, E) nazýváme měřitelným prostorem a množiny $A \subset S$ se nazývají S -měřitelné.

²⁸Přívlastek „bodové“ se často vynechává a používá se jen termín *vzorkování*.

²⁹V teorii signálu se používá výraz *časová oblast*.

Definice 4.3. Necht' E je měřitelná množina. Řekneme, že funkce $f : E \rightarrow \mathbb{R}$ je měřitelná, právě když $\forall a \in \mathbb{R}$ je množina $\{x \in E : f(x) > a\}$ měřitelná.

Definice 4.4. Prostorem funkcí absolutně integrovatelných na E v p -té mocnině zn. $L^p(E)$ rozumíme množinu funkcí $\{f \mid f : E \rightarrow \mathbb{C} \text{ měřitelné, } E \subseteq \mathbb{R} \text{ měřitelná, } \int_E |f(t)|^p dt < \infty\}$, kde $1 \leq p < \infty$.

V dalším textu budeme kromě $L^p(E)$ potřebovat i prostor posloupností absolutně sumovatelných na $F \subset \mathbb{R}$ v p -té mocnině.

Definice 4.5. Prostorem posloupností absolutně sumovatelných na $F \subset \mathbb{R}$ v p -té mocnině rozumíme množinu $l^p(F) := \{\xi \mid \xi = \xi_n, |F| \leq \aleph_0, \sum_{n \in F} |\xi_n|^p < \infty \text{ pro } 1 \leq p < \infty\}$.

Nyní s pomocí pojmů *měřitelná množina* a $L^p(E)$ definujeme dopřednou Fourierovu transformaci.

Definice 4.6. Uvažujme funkci $g(x) \in L^1(E)$ definovanou na spojitém intervalu. Fourierovým obrazem této funkce je komplexní funkce $G(u)$ definovaná vztahem

$$G(u) = \int_{-\infty}^{\infty} g(x) e^{-i2\pi ux} dx, \quad (4.2)$$

kde i je imaginární jednotka. Samotná integrace dle rovnice 4.2 se nazývá dopředná Fourierova transformace a zřejmě se jedná o zobrazení přiřazující funkci jinou funkci.

Obdobně definujeme zpětnou Fourierovu transformaci.

Definice 4.7. Uvažujme funkci $G(u)$, která je Fourierovým obrazem funkce $g(x)$. Zpětnou Fourierovu transformaci, neboli přechod od frekvenční oblasti k prostorové, definujeme vztahem

$$g(x) = \int_{-\infty}^{\infty} G(u) e^{i2\pi ux} du. \quad (4.3)$$

Z těchto dvou definic vidíme, že funkce $g(x)$ a $G(u)$ jsou k sobě jednoznačně přiřazeny pomocí zpětné a dopředné Fourierovy transformace. Tuto skutečnost značíme následovně

$$g(x) \iff G(u).$$

Předchozí definice Fourierových transformací pracovaly se spojitou funkcí $g(x)$. Doplňme nyní tytéž definice pro Fourierovu transformaci diskrétní funkce $\tilde{g}(x)$.

Definice 4.8. Uvažujme diskrétní funkci $\tilde{g}(x) \in l^1(F)$, potom dopřednou Fourierovu transformaci zn. \mathcal{F} vypočteme dle vztahu

$$\tilde{G}(u) = \sum_{x=0}^{N-1} \tilde{g}(x) e^{-i2\pi \frac{ux}{N}} \quad (4.4)$$

a zpětnou Fourierovu transformaci zn. \mathcal{F}^{-1}

$$\tilde{g}(x) = \sum_{u=0}^{N-1} \tilde{G}(u) e^{i2\pi \frac{ux}{N}}, \quad (4.5)$$

přičemž značka $\tilde{g}(x) \iff \tilde{G}(u)$ má stále stejný význam. Fourierovou transformací získáme dekompozici vstupní funkce na funkce sinus a cosinus.

Upozorníme, že obecně neplatí, že inverzní Fourierovou transformací získáme původní vzor. Podrobněji je tato problematika rozepsána v [8].

Definice 4.9. Uvažujme nyní libovolně pevně zvolený bod u náležící Fourierově obrazu funkce $g(x)$. Amplitudové spektrum funkce $G(u)$ značíme $\|G(u)\|$ a je dáno vztahem

$$\|G(u)\| = \sqrt{Re^2(u) + Im^2(u)}$$

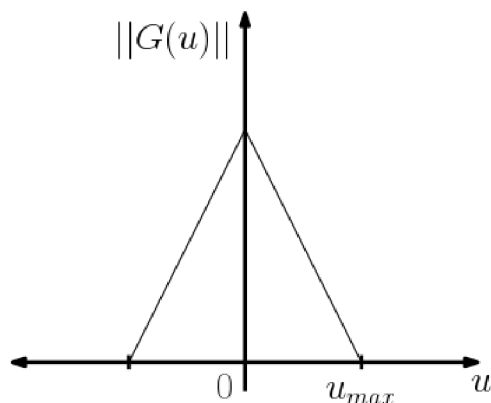
a fázové spektrum

$$\phi(u) = \arctg \frac{Re(u)}{Im(u)},$$

kde $Re(u)$ značí reálnou část Fourierova obrazu a $Im(u)$ jeho imaginární část. Význam jednotlivých spekter je zřejmý. Amplitudy jednotlivých frekvencí a jejich vliv na celkový obraz jsou dány amplitudovým spektrem. Fázové posuny dílčích sinusových a cosinusových složek udává fázové spektrum.

Zavedli jsme tedy základní aparát Fourierovy analýzy potřebný ke zpracování obrazu. Nyní jej využijeme k zodpovězení dříve položené otázky, a sice za jakých podmínek platí, že signál spojitý v čase je jednoznačně popsán posloupností vzorků odečítaných vzorkovací frekvencí.

Uvažujme nyní funkci, která má konečné amplitudové spektrum. Takové funkci se říká frekvenčně omezená funkce a platí pro ni, že v konečném amplitudovém spektru existuje frekvence u_{max} ³⁰, s vlastností, že pro všechny frekvence $u > u_{max}$ platí $\|G(u)\| = 0$, viz obrázek 28. Frekvence u_{max} je tedy nejvyšší frekvencí ve frekvenčním spektru frekvenčně



Obrázek 28: Frekvenčně omezená funkce. Převzato z [9].

omezené funkce. Uvedme nyní Shannonovu vzorkovací větu, která dává do souvislosti vzorkovací frekvenci u_t , a u_{max} .

Věta 4.10. *Nechť platí $u_t > 2u_{max}$, potom je signál spojitý v čase jednoznačně popsán posloupností vzorků odečítaných s frekvencí u_t .*

Praktickým důsledkem této věty je skutečnost, že ke správné jednoznačné diskretizaci každé frekvenčně omezené funkce je nutné použít vzorkovací frekvenci splňující podmínku Shannonovy věty. Při jejím nedodržení dochází k tzv. aliasingu.

³⁰Této frekvenci se také říká Nyquistovo kritérium.

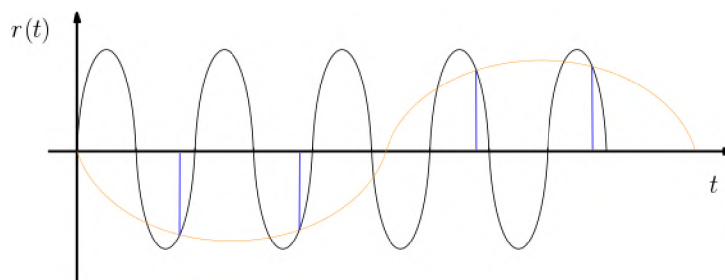
4.4 Alias a antialiasing

V minulé podkapitole jsme se seznámili s Shannovou větou, která udává nejmenší možnou vzorkovací frekvenci, takovou, aby navzorkovanému signálu bylo možné jednoznačně přiřadit jeho původní spojitou podobu. Nyní zmíníme případy tzv. *podvzorkování*. Tím máme na mysli vzorkování se vzorkovací frekvencí, která je menší než $2u_{max}$. Podvzorkování způsobuje tzv. *alias*.

Alias je nová, nežádoucí, nízkofrekvenční informace, která vzniká dvěma způsoby.

1. Původní funkce je frekvenčně neomezená, tzn. neexistuje konečné u_{max} .
2. Původní funkce je frekvenčně omezená, ale vzorkovací frekvence nesplňuje Shannovu větu.

Obrázek 29 ilustruje druhý, výše zmíněný případ. Nevhodně zvolenou vzorkovací frekvencí se z původního signálu (černá křivka) stala po jeho rekonstrukci jiná nízkofrekvenční funkce (oranžová křivka), kterou nazýváme alias.



Obrázek 29: Alias. Převzato z [9].

Metodu, která alias odstraňuje, nazýváme antialiasing. Více čtenář nalezne např. v [9]. Přejdeme nyní k dalšímu důležitému pojmu, kterým je konvoluce.

4.5 Konvoluce

Definice 4.11. Konvolucí neperiodické spojitě funkce rozumíme zobrazení $T_h : L^p \rightarrow L^p$, které je dáno vztahem $y := T_h(g)$, vyjádřeným $\forall t \in \mathbb{R}$ následovně

$$y(t) = \int_{-\infty}^{\infty} h(t - \alpha)g(\alpha)d\alpha, \quad (4.6)$$

kde funkce $h \in L^1$) je označována termínem *konvoluční jádro*. Konvoluci funkce g s konvolučním jádrem h značíme $g(x) * h(x)$.

Zřejmě platí

$$\begin{aligned} G(u) * H(u) &\iff g(x)h(x), \\ G(u)H(u) &\iff g(x) * h(x). \end{aligned}$$

Definujme dále lineární diskretní konvoluci, kterou následně zobecníme na dvojrozměrnou lineární diskretní konvoluci, často používanou pro práci s obrazovou maticí.

Definice 4.12. Lineární diskretní konvolucí pro neperiodické případy rozumíme zobrazení $T_h : l^p \rightarrow l^p$, které je dáno vztahem $\mu := T_h(\xi)$, předepsaným rovnicí

$$\mu_i = \sum_{j=-\infty}^{\infty} h_{i-j}\xi_j, \forall i \in \mathbb{Z},$$

kde $h \in l^1$ opět označujeme termínem *konvoluční jádro*. Symbol $*$ má analogický význam jako v definici 4.11.

Předchozí definici snadno rozšíříme pro dvoudimenzionální případ vztahem

$$f'_o(x, y) = f_o(x, y) * h(x, y) = \sum_{k=-M}^M \sum_{l=-N}^N f_o(x - k, y - l)h(k, l).$$

Při diskrétní dvourozměrné konvoluci se pracuje s obrazovou maticí $f_o(x, y)$ a konvolučním jádrem, tj. $h(x, y)$. Symbol $f'_o(x, y)$ označuje výslednou obrazovou matici.

Proces vzorkování je možné snadno popsat pomocí konvoluce a *Diracovy funkce* $\delta(t)$

$$\delta(t) = \begin{cases} \delta(t) = 0 & \Leftrightarrow t \neq t_0 \\ \delta(t) = 1 & \Leftrightarrow t = t_0 \end{cases},$$

přičemž je zřejmé, že platí

$$\int_{-\infty}^{\infty} \delta(t)dt = 1.$$

Vzorkovací funkci můžeme tedy popsat jako nekonečnou posloupnost Diracových impulsů ve vzájemné konstantní vzdálenosti Δt , tj.

$$s(t) = \sum_{j=-\infty}^{\infty} \delta(t - j\Delta t).$$

Fourierovým obrazem této funkce je

$$S(u) = \frac{1}{\Delta t} \sum_{j=-\infty}^{\infty} \delta\left(\frac{u - j}{\Delta t}\right).$$

Uvažujme nyní funkci $r(t)$ popisující analogový signál a vzorkovací funkci $s(t)$. Součin těchto dvou funkcí potom vyjadřuje funkce $\tilde{r}(t)$ a jejím Fourierovým obrazem je funkce,

$$\tilde{r}(t) = r(t)s(t) \iff \frac{1}{\Delta t} \sum_{j=-\infty}^{\infty} \tilde{R}\left(\frac{u - j}{\Delta t}\right),$$

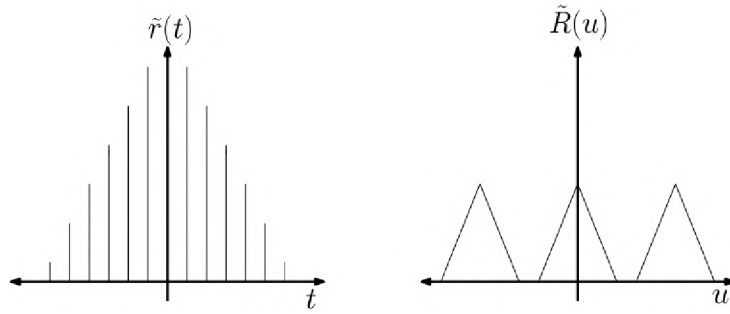
která je opakováním obrazu funkce $r(t)$ ve vzdálenosti Fourierových obrazů pulsů vzorkovací funkce, jak je znázorněno obrázkem 30, na němž je vzorkovací frekvence volena tak, aby se Fourierovy obrazy nepřekrývaly. Na obrázku 31 můžeme vidět, že při nevhodném zvolení vzorkovací frekvence se jednotlivé Fourierovy obrazy překrývají.

Chceme-li Fourierovým obrazům jednotlivých vzorků signálu přiřadit jejich vzorovou funkci, je nutné, aby se tyto obrazy vzájemně nepřekrývaly. Tato podmínka je automaticky splněna pro frekvenčně omezené funkce a pro vzorkovací frekvence splňující větu 4.10.

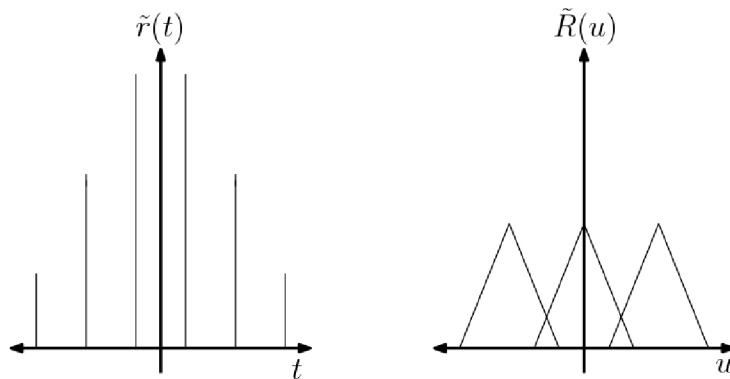
Pracujme dále s funkcí $\tilde{r}(t)$. Platí, že tuto funkci můžeme zrekonstruovat zpět na funkci $r(t)$, a to v časové i frekvenční oblasti.

Rekonstrukce ve frekvenční oblasti je dána vztahem

$$r(t) = \tilde{R}(u)S(u)w(u),$$



Obrázek 30: Vhodně navzorkovaný signál $r(t)$ a jeho amplitudové spektrum. Převzato z [9].



Obrázek 31: Nevhodně navzorkovaný signál $r(t)$ a jeho amplitudové spektrum. Převzato z [9].

kde

$$w(u) = \begin{cases} u = 1 & \Leftrightarrow u \in \langle -1, 1 \rangle \\ u = 0 & \Leftrightarrow u \notin \langle -1, 1 \rangle \end{cases}.$$

Rekonstrukci v časové oblasti získáme jako výsledek konvoluce

$$r(t) = \tilde{r}(t) * \text{sinc}(t),$$

kde

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}.$$

Pro názornost jsme dosud pracovali s jednorozměrnými případy, ale obrazová matice je dvojrozměrná, proto nyní ukážeme definici zpětné a dopředné Fourierovy transformace pro dvě proměnné. Všechny zbylé definice lze převést z jednorozměrného případu na dvojrozměrný zcela analogicky.

Definice 4.13. Nechť funkce $g(x, y) \in L^1(J)$, kde $J \subset \mathbb{R}^2$. Potom její dopřednou a zpětnou Fourierovu transformaci předepíšeme vztahy

$$G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-i2\pi(ux+vy)} dx dy$$

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(u, v) e^{i2\pi(ux+vy)} du dv.$$

Definice 4.14. Necht' diskrétní funkce $\tilde{g}(n, m) \in l^1(J)$, kde $J \subset \mathbb{R}^2$. Potom její dopřednou a zpětnou Fourierovu transformaci předepisujeme vztahy

$$\tilde{G}(u, v) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \tilde{g}(n, m) e^{-\frac{2\pi i}{N}(mu+nv)}$$

$$g(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \tilde{G}(u, v) e^{\frac{2\pi i}{N}(ux+vy)}.$$

Uvedli jsme tedy proces digitalizace signálu s využitím vhodného matematického aparátu. V další kapitole se budeme zabývat digitálním šumem a problematikou jeho odstraňování.

5 Digitální šum

Pojem digitální šum je označení pro náhodnou složku digitálního obrazu, které nejde zabránit a která vzniká náhodně, například vlivem poškození čipu snímacího zařízení, kvalitou světelných podmínek³¹, kompresí obrazu a mnoha dalšími faktory. Především je však nežádoucí, a proto vznikly metody na potlačení různých druhů šumu. Podrobnější informace k této kapitole lze nalézt v [2], [3], [4], [5], [9].

Digitální šum lze kategorizovat dle několika kritérií³²

1. Podle závislosti na obsahu obrazu

- *Šum závislý na obraze* - obtížně se odstraňuje, ale obvykle se nevyskytuje.
- *Šum nezávislý na obraze* - způsoben například dálkovým přenosem dat.

2. Podle velikosti a rozložení šumu

- *Impulsní šum*³³ - náhodným pixelům je přiřazena náhodná hodnota jasu.
- *Šedý šum* - postihuje všechny pixely, ale hodnoty jasu většinou nemění.

3. Podle vztahu obrazu k obsahu obrazu

- *Aditivní šum* - k původní hodnotě jasu pixelu je přičtena náhodná hodnota.
- *Multiplikativní šum* - původní hodnota jasu pixelu je vynásobena náhodnou hodnotou.
- *Jiné druhy*

4. Podle charakteru šumu ve frekvenční oblasti

- *Úzkopásmový šum* - vzniká kombinací více úzkopásmových nebo multiplikativních signálů.
- *Širokopásmový šum*

Toto jsou tedy ukázky jednotlivých kategorií digitálního šumu, z nichž za podrobnější pozornost stojí zejména šum aditivní a impulsní.

5.1 Impulsní šum a způsoby jeho filtrace

Impulsní šum se projevuje přítomností náhodně rozmístěných vadných pixelů, které je třeba vždy opravit, a lze jej definovat následovně

³¹Obecně platí, že tmavé části obrazu obsahují více šumu.

³²Převzato z [2].

³³Někdy též označován termínem salt & pepper.

Definice 5.1. Necht \mathbf{B}^* je dokonalý digitální obraz, $Z \sim A(p)$ je náhodná veličina s alternativním rozdělením s parametrem p a Y je náhodná veličina s libovolným rozložením, potom o obrazu \mathbf{C} definovaném následovně

$$\mathbf{c}_{i,j} = \begin{cases} \mathbf{b}_{i,j}^* & \Leftrightarrow Z = 0 \\ Y & \Leftrightarrow Z = 1 \end{cases}$$

říkáme, že je zatížen impulsním šumem.

Tato rušivá složka je způsobena například vadnými pixely na snímači, kosmickým zářením, dálkovým přenosem dat nebo prachovými částicemi.

Filtrace impulsního šumu se skládá ze dvou dílčích kroků, a sice detekce a korekce vadných pixelů. Věnujme se nejprve samotné detekci.

Pixely ovlivněné impulsním šumem můžeme detekovat například přímo ve snímači, hardwarově nebo matematicky. Dále se budeme zabývat výhradně matematickým přístupem, který vychází z testování statistické hypotézy poškození pixelu.

Řekneme, že pixel je vadný, pokud platí

$$|\mathbf{c}_{i,j} - \check{\mathbf{c}}_{i,j}| > \varepsilon,$$

kde $\mathbf{c}_{i,j}$ je hodnota jasu právě vyšetřovaného pixelu, $\check{\mathbf{c}}_{i,j}$ značí charakteristiku polohy statistického souboru³⁴ hodnot jasů pixelů ležících v okolí $\mathbf{c}_{i,j}$ a ε je pevně zvolená konstanta udávající sílu testu.

Nevýhodou tohoto přístupu je reálná možnost vzniku druhého druhu statistické chyby, tzn. oprava hodnot jasu správných pixelů.

Výše zmíněné testovací kritérium lze nahradit tzv. rankovým kritériem, tj.

$$\left| r - \frac{n}{2} \right| > \varepsilon.$$

Sílu testovacího kritéria opět nastavujeme pomocí libovolně pevně zvolené konstanty ε . Rank³⁵ testovaného pixelu značíme r a n je počet prvků statistického souboru pixelů ležících v okolí testovaného pixelu.

Po detekci poškozených pixelů následuje korekce jejich jasu, kterou lze provést několika způsoby.

1. První, nejjedodušší metodou je tzv. *jednoprůchodová korekce*. Ihned po detekci vadného pixelu, nahradí jeho hodnotu jasu např. aritmetickým průměrem, nebo mediánem hodnot jasů pixelů ležících ve zvoleném okolí tohoto vadného pixelu. Výhodou metody je její rychlost. Nevýhodou je reálná šance, že v okolí vadného pixelu leží další vadné pixely, které mohou výrazně ovlivnit výsledek.
2. Sofistikovanějším přístupem je *dvouprůchodová metoda*, která nejdříve najde všechny vadné pixely a teprve poté pracuje na jejich korekci, která probíhá stejně jako u *jednoprůchodové metody*, avšak s rozdílem, že vadné pixely se do výpočtu nové hodnoty jasu nezahrnují. Metoda je efektivnější, ale časově náročnější.
3. Další možnost korekce představuje *interpolační metoda*. Opět se jedná o dvouprůchodovou metodu, ale vadný pixel se opravuje pomocí interpolace hodnot okolních pixelů. Poznamenejme, že okolí interpolace může být jiné než okolí statistického sběru dat.

Metody filtrace impulsního šumu jsou použitelné, pokud v obraze je méně než 20 % pixelů postiženo impulsním šumem.

³⁴Nejvhodnější je používat medián, poté aritmetický průměr a modus.

³⁵Při seřazení n prvků statistického souboru od nejmenšího po největší bude mít nejmenší prvek rank 1 a největší rank n .

5.2 Aditivní šum a způsoby jeho filtrace

Uvažujme dokonalý digitální obraz \mathbf{A}^* , obraz s realizací náhodné veličiny X s normálním rozložením pravděpodobnosti o nulové střední hodnotě a nenulovým rozptylem a obraz \mathbf{B} , který vznikne sečtením obrazů \mathbf{A}^* , X , tzn. $\mathbf{B} = \mathbf{A}^* + X$. Této náhodné veličině X říkáme aditivní šum a o obrazu \mathbf{B} mluvíme jako o obrazu zatíženém aditivním šumem.

Mezi aditivním šumem a dokonalým obrazem existuje jedna z níže uvedených relací.

- Aditivní šum je nezávislý na obraze, tzn. šum nezávisí na obsahu fotografie.
- Aditivní šum je nekorelovaný s obrazem.
- Aditivní šum je závislý na obraze.

Pro CCD a CMOS čipy platí, že šum a obraz jsou buď nekorelované nebo nezávislé. Ve většině případů platí, že tmavé části obrazu jsou aditivním šumem zatíženy více než části světlé. Toto lze snadno odpozorovat vyšší zrnitostí tmavých oblastí, která je typickým znakem aditivního šumu. Charakteristickou vlastností aditivního šumu je směrodatná odchylka³⁶, tj.

$$\sigma(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\left(\frac{1}{n} \sum_{i=1}^n x_i^2\right) - \bar{x}^2}, \quad (5.1)$$

kde n je počet pixelů v obraze, x_i je náhodná hodnota jasu pro i -tý pixel a \bar{x} značí aritmetický průměr.

Zabývejme se nyní možnými způsoby filtrace aditivního šumu.

Průměrování více obrazů

Uvažujme, že jsme m -krát vyfotografovali stejnou statickou scénu a pořídili tedy m prvkový soubor obrazů $\mathbf{B}_1, \dots, \mathbf{B}_m$. Chceme-li potlačit aditivní šum pixelu $\mathbf{b}_{i,j}$, postačí využít vztahu

$$\mathbf{b}_{i,j} = \frac{\sum_{p=1}^m \mathbf{b}_{p,i,j}}{m}, \quad (5.2)$$

kde index p značí příslušnost pixelu k danému obrazu \mathbf{B}_p pro $p = 1, \dots, m$. Z tohoto vztahu plyne, že každý pixel $\mathbf{b}_{i,j}$ je zatížen náhodnou veličinou s nulovou střední hodnotou, ale menší směrodatnou odchylkou o velikosti $\frac{\sigma}{\sqrt{m}}$.

Tuto metodu využívá přiložený program, viz CD, a její nespornou výhodou je nerozmazání výsledného obrazu, ke kterému dochází u použití jiných metod, které nyní rozebereme.

5.3 Lokální vyhlazování obrazu

Metody, které budou uvedeny v této kapitole, budou pracovat pouze s jedním obrazem zatíženým aditivním šumem. Pro potlačení šumu daného pixelu budeme využívat hodnot pixelů, které leží v jeho pevně zvoleném okolí O . Takto založené metody se označují jako metody filtrace obrazu.

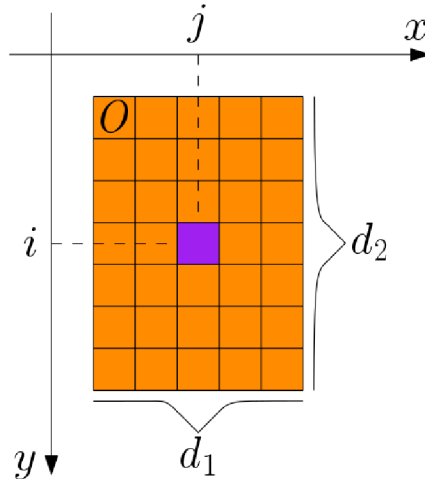
Tyto metody se podle účelu dělí na

1. *Vyhlazovací* – usilují o potlačení šumu.
2. *Detekce hran* – odhadují derivaci obrazové funkce v daném bodě pomocí hodnot pixelů ležících v okolí O tohoto bodu.

³⁶Lze využít i vztahu $\sigma(x) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\left(\frac{1}{n-1} \sum_{i=1}^n x_i^2\right) - \bar{x}^2}$.

Dalším možným tříděním těchto metod je podle matematických vlastností dané transformace. V takovém případě rozlišujeme metody *lineární* a *nelineární*.

U lineárních metod je hodnota pixelu výstupního obrazu na souřadnicích i, j počítána jako lineární kombinace hodnot pixelů ve vstupním obrazu ležících v daném okolí O , o rozměrech³⁷ d_1, d_2 , viz obrázek 32, pixelu na souřadnicích i, j . Uvažujme nyní $M = d_1$



Obrázek 32: Okolí O pixelu, u něhož chceme potlačit šum.

$div\ 2$, $N = d_2\ div\ 2$. Platí, že jednotlivé pixely v tomto okolí mají různé důležitosti, a proto je při výpočtu opatřujeme váženými koeficienty h podle

$$\bar{f}_o(i, j) = \sum_{k=-M}^M \sum_{l=-N}^N f_o(i - k, j - l)h(k, l), \quad (5.3)$$

kde $\bar{f}_o(i, j)$ značí hodnotu jasu pixelu výstupního obrazu a $f_o(i - k, j - l)$ hodnotu jasu pixelu zpracovávaného obrazu. Samotná rovnice 5.3 vyjadřuje diskétní konvoluci s jádrem h ³⁸. Konvoluční filtry (filtrace) se velmi snadno používají a jejich uplatnění nalezneme například při vyhlazování obrazu nebo detekci hran.

Jak již bylo řečeno v podkapitole 5.2 nejvýhodnější metodou k potlačení šumu je průměrování obrazů, neboť nedochází k rozmazání obrazu. Máme-li však k dispozici pouze jeden obraz musíme se spolehnout na jiné metody, které využívají poznatku, že sousední pixely mají velmi blízkou hodnotu jasu. Hodnotu pixelů zatížených šumem nahradíme reprezentativní hodnotou, např. aritmetickým průměrem, mediánem atd., spočtenou z hodnot sousedních pixelů.

5.3.1 Lineární metody vyhlazování

Lineární metody přiřadí nové hodnotě pixelu výsledek lineární kombinace hodnot pixelů ležících ve zkoumaném okolí. Tuto lineární kombinaci lze vyjádřit diskrétní konvolucí, viz rovnice 5.3. V dalším textu se postupně seznámíme s různými lineárními filtry, které se od sebe liší konvolučním jádrem.

Poznamenejme, že lineární filtry dělíme na dvě skupiny, a to na skupinu *lineárních filtrů* a *lineárně prostorových invariantních filtrů*, viz obrázky 33, 34, kde $h(i, j)$ je konvoluční maska a $f_o(i, j)$ s $\bar{f}_o(i, j)$ po řadě značí vstupní a výstupní obrazovou matici. Vztah 5.3 je

³⁷Často se uvažuje čtvercové okolí, tedy $d_1 = d_2$.

³⁸Též se používá sousloví *konvoluční maska*.

$$f_o(i, j) \longrightarrow \boxed{h(i, j)} \longrightarrow \bar{f}_o(i, j)$$

Obrázek 33: Lineární filtr.

$$f_o(i - k, j - l) \longrightarrow \boxed{h(i, j)} \longrightarrow \bar{f}_o(i - k, j - l)$$

Obrázek 34: Lineárně prostorově invariantní filtr.

příkladem lineárního prostorově invariantního filtru. Přirozeně vzniká problém v situacích, kdy konvoluční maska „vyčnívá“ z hranic obrazové matice. Tato situace nastává například u pixelů na hranicích obrazové matice. Tento problém lze vyřešit neaplikováním konvoluční masky na tyto pixely, nebo vhodným pozměněním jejich hodnot a tvaru.

Jednoduchou metodou na potlačení šumu je *obyčejné průměrování*. Hodnota jasu každého pixelu je dána aritmetickým průměrem původních hodnot jasu v okolí tohoto pixelu. Konvoluční maska pro okolí 3×3 je

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

Poznamenejme, že prvky v konvolučním jádru indexujeme následovně

$$\begin{bmatrix} h_{1,1} & h_{1,0} & h_{1,-1} \\ h_{0,1} & h_{0,0} & h_{0,-1} \\ h_{-1,1} & h_{-1,0} & h_{-1,-1} \end{bmatrix}.$$

V situacích, kdy vybraným pixelům přikládáme větší váhu, je nutné tuto masku vhodně upravit. Následující dvě konvoluční jádra po řadě zdůrazňují vliv středového pixelu nebo jeho 4-sousedů

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Masky pro větší okolí se vytvářejí zcela analogicky, vždy je nutné, aby se součet prvků konvolučního jádra rovnal 1, tedy

$$\sum_{i=-k}^k \sum_{j=-l}^l h_{i,j} = 1,$$

kde $2k + 1, 2l + 1$ jsou rozměry konvoluční masky.

Výhodou této metody je její jednoduchost, která je ale vykoupená rozmazáním hran v obraze.

Výpočet rovnice 5.3 lze urychlit pomocí tzv. *separabilních filtrů*, tj. takových lineárních filtrů, jejichž konvoluční masku můžeme rozložit na součin jednorozměrných masek. Platí, že každá konvoluční maska s hodnotami 1 je separabilní.

Příklad 5.2. Uvažujme konvoluční jádro h , které rozložíme na součin jednorozměrných masek h_1, h_2

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

Následující tvrzení využívá vlastnost separabilních filtrů k přepisu rovnice 5.3 do výpočetně rychlejšího tvaru.

Tvrzení 5.1. *Uvažujme separabilní filtr se čtvercovým konvolučním jádrem o rozměrech $2N + 1$. Potom lze rovnici 5.3 přepsat do tvaru*

$$g(i, j) = \sum_{k=-N}^N \sum_{l=-N}^N h(k, l) f_o(i - k, j - l) = \sum_{k=-N}^N h_1(k) \sum_{l=-N}^N h_2(l) f_o(i - k, j - l).$$

Další skupinou prostorově invariantních filtrů jsou *rekurzivní filtry*, tj. takové filtry, které pro výpočet nových hodnot pixelů využívají hodnoty již dříve napočítaných pixelů. Ukázkou rekurzivního filtru pro 1D případ ilustruje následující příklad.

Příklad 5.3. Necht' $\mathbf{g}_n, n \in \mathbb{Z}_0^+$ je posloupnost pixelů a $h = (h_0, \dots, h_r)$ je maska filtru. Rekurzivní 1D filtr je dán rovnicí

$$\mathbf{g}'_n = h_0 \mathbf{g}_n + \sum_{i=0}^r h_i \mathbf{g}'_{n-i},$$

kde \mathbf{g}'_n je posloupnost upravených pixelů.

Rekurzivní filtry převyšují nerekurzivní filtry ve výpočetní efektivitě, ale pro 2D obrazy se nepoužívají kvůli řadě netriviálních problémů viz [4].

Popsali jsme tedy lineární metody vyhlazování obrazu. V další kapitole se budeme věnovat metodám nelineárním.

5.3.2 Nelineární metody vyhlazování

Z minulé kapitoly víme, že lineární vyhlazovací metody způsobují rozmazávání hran. Tento neduh lze částečně vyřešit používáním nelineárních vyhlazovacích metod. Společným rysem těchto metod je, že z analyzovaného okolí O vybírají jen ty pixely, jejichž hodnota jsou je blízká hodnotě jasu reprezentativního pixelu. Právě tyto pixely jsou použity k výpočtu³⁹ jasu zkoumaného pixelu.

Jednoduchou ukázkou vyhlazovací metody, která k výpočtu jasu využívá nejhodnější oblast, je metoda *rotující masky*. Kolem reprezentativního bodu rotuje malá maska, viz obrázek 35, v jejíž každé poloze se spočte rozptyl jasů pixelů v ní ležících a k výpočtu jasu se nakonec použijí pixely, které leží v masce, v níž byl tento rozptyl nejmenší. Metodu rotující



Obrázek 35: Ilustrace rotující masky o velikosti 3×3 na okolí 5×5 . Reprezentativní pixel je vybarven oranžově, zbytek masky pak modře.

masky lze využít iterativně, přičemž hlavní faktory ovlivňující rychlost konvergence⁴⁰

³⁹Opět se může jednat například o aritmetický průměr.

⁴⁰Je dokázáno, že tento proces vždy konverguje do stabilního stavu.

jsou tvar a velikost masky. Mezi velikostí masky a počtem nutných iterací existuje přímá úměrnost.

Metoda *obyčejného průměrování s omezením změn jasu* je další používaná nelineární metoda, která je založena na principu obyčejného průměrování, avšak částečně brání rozmáznutí hran. Opět se počítá lineární kombinace⁴¹ hodnot jasů v okolí reprezentativního pixelu, ale výsledná hodnota se použije pouze, pokud je splněna podmínka,

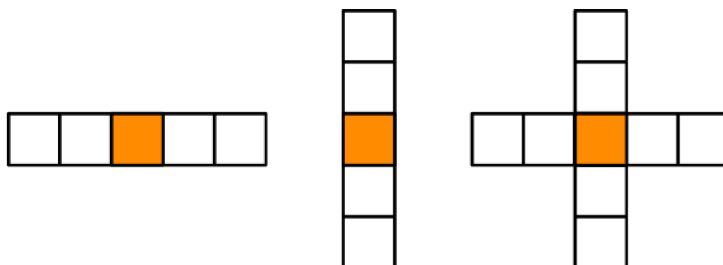
$$\left| \mathbf{p}_{i,j} - \hat{\mathbf{p}}_{i,j} \right| < \varepsilon,$$

kde $\varepsilon \in \mathbb{N}$ je libovolně pevně zvolená hodnota, $\mathbf{p}_{i,j}$ je hodnota jasu reprezentativního pixelu a $\hat{\mathbf{p}}_{i,j}$ je výsledek zvolené lineární kombinace hodnot jasů pixelů ležících v okolí O pixelu $\mathbf{p}_{i,j}$. V případě, kdy podmínka splněna není, se pixelu ponechá jeho původní hodnota jasu.

Všechny předchozí metody využívaly k nalezení nové hodnoty jasu všechny pixely v příslušném okolí. Největší slabinou této myšlenky je skutečnost, že v takovém okolí může ležet pixel, jehož hodnota jasu⁴² se výrazně liší od hodnot ostatních pixelů, což značným způsobem ovlivní výsledek příslušné lineární kombinace. Z tohoto důvodu se využívají metody robustní statistiky, které dokáží v daném souboru hodnot najít takové hodnoty, které jsou příliš vychýlené, vyloučit je z uvažování a zbylé hodnoty nahradit jinou typickou hodnotou.

Myšlenku robustní statistiky ztělesňujeme pomocí výběrových kvantilů, nejčastěji mediánu z hodnot jasů bodů v okolí reprezentativního bodu vstupního obrazu. Pomocí této mediánové metody docílíme potlačení impulsního šumu.

Svou roli při použití této metody hraje i tvar zvoleného okolí. Obdelníkové okolí není vhodné, neboť porušuje tenké čáry a ostré rohy v obraze. Naštěstí lze tento nedostatek snadno obejít užitím jiného tvaru okolí viz obrázek 36.



Obrázek 36: Ukázky vhodných tvarů okolí typických při mediánové filtraci pro zachování svislých a vodorovných čar. Převzato z [4].

Ukázali jsme tedy základní metody k potlačení aditivního šumu. V následující kapitole se budeme zabývat problematikou hledání hran.

6 Hledání hran

Oblasti obrazu, v kterých se náhle mění hodnoty jasů, nazýváme hranami. Hrana je určena rychlostí změn hodnot jasů pixelů v obrazové matici.

⁴¹Nejčastěji se používá jedna z výše uvedených konvolučních masek k průměrování obrazu.

⁴²Tato situace je typická například pro impulsní šum.

Připomeňme, že se jedná o diskrétní funkci dvou proměnných, a proto budeme pro detekci hran využívat aproximace parciální derivace a gradient této funkce. Velikost gradientu a jeho směr jsou po řadě dány vztahy

$$\|\nabla f_o(i, j)\| = \sqrt{\left(\frac{\partial f_o(i, j)}{\partial i}\right)^2 + \left(\frac{\partial f_o(i, j)}{\partial j}\right)^2},$$

$$\psi = \operatorname{argtn}\left(\frac{\frac{\partial f_o(i, j)}{\partial j}}{\frac{\partial f_o(i, j)}{\partial i}}\right).$$

Platí, že směr hrany je kolmý na směr gradientu. Výpočet parciální derivace na hranicích obrazové matice lze aproximovat následovně (analogicky pro $\frac{\partial f_o(i, j)}{\partial j}$)

$$\frac{\partial f_o(i, j)}{\partial i} \approx f_o(i, j) - f_o(i - 1, j),$$

$$\frac{\partial f_o(i, j)}{\partial i} \approx f_o(i + 1, j) - f_o(i, j)$$

a pro výpočet parciální derivace v pixelu neležícím na hranici lze využít aproximaci

$$\frac{\partial f_o(i, j)}{\partial i} \approx \frac{f_o(i + 1, j) - f_o(i - 1, j)}{2}.$$

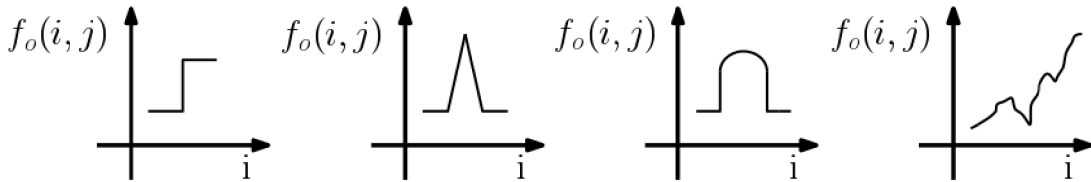
Další možnosti numerického výpočtu parciálních derivací lze nalézt v [1].

V situacích, kdy nás nezajímají směrové vlastnosti hran, používáme Laplaceův operátor definovaný vztahem

$$\nabla^2 f_o(i, j) = \frac{\partial^2 f_o(i, j)}{\partial i^2} + \frac{\partial^2 f_o(i, j)}{\partial j^2}.$$

Tento operátor je roven nulové hodnotě v bodech, v nichž je velikost gradientu maximální.

Je zřejmé, že hrany obvykle nejsou určeny skokovou změnou jasu, ale jeho plynulým přechodem, viz obrázek 37, který ilustruje možné přechody jasu na hranách objektu ve svislé ose. První tři obrázky jsou idealizované. Poslední profil odpovídá reálným datům, která jsou zatížena šumem.



Obrázek 37: Profily jasu na hranách objektů. Převzato z [4].

Poznamenejme, že gradientní operátory lze využít i pro ostření obrazu, a to podle vztahu

$$\bar{f}_o(i, j) = f_o(i, j) - \kappa \rho(i, j),$$

kde $f_o(i, j)$ je původní obrazová matice, κ je kladný parametr určující sílu ostření a $\rho(i, j)$ je velikost operátoru udávajícího strmost změny jasu v daném bodě. Tímto operátorem může být například gradient nebo Laplacián.

Dále se budeme věnovat konvolučním maskám, kterými lze aproximovat derivace obrazové funkce. Následující konvoluční masky po řadě pro 4-sousedství a 8-sousedství aproximují Laplaceův operátor v okolí 3×3

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Tyto masky lze upravit, aby pixely blíže k reprezentativnímu bodu měly větší váhy.

$$h = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}, \quad h = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}.$$

Mezi hlavní nevýhody tohoto operátoru patří velká citlivost na šum.

Další informace o metodách detekujících hrany čtenář nalezne v [4].

7 Zaostření obrazu

K zaostřování obrazu se používá diskretní dvojrozměrná konvoluce. Uvažujme obraz \mathbf{A} a náhodnou veličinu N s nulovou střední hodnotou, nenulovým rozptylem σ a normálním rozdělením pravděpodobnosti. Chceme-li z obrazu \mathbf{A} odfiltrovat vysoké frekvence, použijeme *filtr dolní propust*, naopak pro odstranění nízkých frekvencí používáme *filtr horní propust*.

Filtr typu dolní propust získáme jako konvoluci obrazu \mathbf{A} s Gaussovou funkcí s normálním rozdělením $N(0, \sigma)$. Zmíňme, že obrazem Fourierovy transformace Gaussovy funkce pro normální rozdělení s parametry $N(0, \sigma)$ je opět Gaussova funkce, ale s parametry pro normální rozdělení $N(0, \sigma^{-1})$, zn. $G_{\frac{1}{\sigma}}$. Upozorníme, že $G_{\frac{1}{\sigma}}$ představuje diskretní dvourozměrné rozložení. Samotný filtr dolní propust lze vypočítat pomocí jedné z následujících rovnic

$$\begin{aligned} \mathbf{B} &= \mathbf{A} * G_{\frac{1}{\sigma}}, \\ \mathbf{B} &= \mathcal{F}^{-1}(\mathcal{F}(\mathbf{A}) \cdot G_{\sigma}). \end{aligned}$$

Výsledkem tohoto filtru je obraz \mathbf{B} , v němž je potlačen šum, ale za cenu snížení jeho ostrosti.

Opakem filtru dolní propust je filtr typu horní propust, který propouští aditivní šum, ale lze jej využít k zaostření původního obrazu. Pro tento filtr se definuje tzv. Diracovo jádro zn. D , což je matice, která má všechny prvky nulové s výjimkou centrálního, jehož hodnota je rovna jedné. Filtr horní propust je předepsán vztahem

$$\mathbf{A} - \mathbf{A} * G_{\frac{1}{\sigma}} = \mathbf{A} * D - \mathbf{A} * G_{\frac{1}{\sigma}} = \mathbf{A} * (D - G_{\frac{1}{\sigma}}).$$

Pomocí substituce $S : H_{\frac{1}{\sigma}} = D - G_{\frac{1}{\sigma}}$ získáme

$$\mathbf{A} * H_{\frac{1}{\sigma}}.$$

Samotné doostření původního obrazu \mathbf{A} pomocí filtru horní propust pak provedeme přičtením výsledku filtru horní propust k původnímu obrazu, tj.

$$\mathbf{A} + \mathbf{A} * H_{\frac{1}{\sigma}}.$$

Na závěr kapitoly uvedeme možná konvoluční jádra, která lze použít pro doostření obrazu. Zaostření obrazu pomocí konvolučního jádra je nesmírně jednoduchou metodou. Tato jednoduchost je však vykoupená faktem zvýšení šumu v zaostřeném obrazu.

Níže jsou po řadě uvedena konvoluční doostřovací jádra pro 4-sousedství a 8-sousedství

$$h = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

Výpočet provádíme standardně dle rovnice 5.3.

Probrali jsme tedy důležitou a obsáhlou kapitolu o problematice digitálního šumu a jeho odstraňování doplněnou o metodu detekce hran a zaostřování obrazu. V další kapitole stručně shrneme poslední nezbytný matematický aparát použitý k vytvoření algoritmu pro zpracování dat zatížených atmosferickým seeingem.

8 Vybrané matematické metody pro zpracování obrazu

V této kapitole uvedeme poslední nezbytné matematické postupy, které následně využijeme pro potlačení seeingu. Budou jimi korelace digitálního obrazu, interpolace a bilineární interpolace.

8.1 Korelace digitálního obrazu

Pomocí korelace lze vyhodnotit relativní posuvy mezi dvěma stejnými obrazy, které jsou však zatíženy například geometrickou deformací oblastí vlivem seeingu.

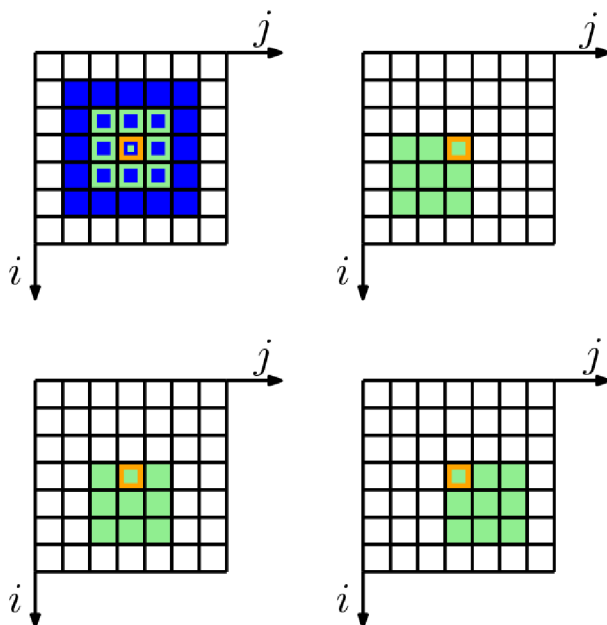
Než přejdeme k samotné metodě, připomeňme vztah pro výpočet korelace mezi dvěma soubory dat X, Y

$$r = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sigma(x)\sigma(y)},$$

kde n je počet prvků statistického souboru, \bar{x}, \bar{y} jsou aritmetické průměry a $\sigma(x)$ značí směrodatnou odchylku (analogicky pro $\sigma(y)$), která je dána vztahem 5.1.

Uvažujme nyní dva obrazy \mathbf{A}, \mathbf{B} , které se od sebe liší například v geometrických deformacích oblastí. Chceme-li zjistit posunutí pixelu $\mathbf{a}_{i,j}$ vůči obrazu \mathbf{B} , pak postupujeme následovně.

1. Určíme velikost okolí pixelu $\mathbf{a}_{i,j}$ (oranžový pixel na obrázku 38) statistického sběru dat (zelená oblast, dále jen *korelační maska*) a velikost okolí (modrá oblast), v němž budeme vyšetřovat korelaci pixelů mezi obrazy.
2. Postupně spočteme hodnotu korelace z dat získaných z korelační masky se středem v pixelu $\mathbf{a}_{i,j}$ obrázku \mathbf{A} a z dat ležících v korelační masce na obrázku \mathbf{B} se středem v každém z modrých pixelů, jak je znázorněno na obrázkových maticích na obrázku 38.
3. Leží-li bod s maximální hodnotou korelace na hranici okolí, pak toto okolí rozšíříme do všech směrů o jeden pixel a znovu postupujeme od bodu 2.
4. Nakonec vybereme takový bod, v němž je hodnota korelace největší a vektor posuvu spočteme jako rozdíl tohoto bodu s bodem $\mathbf{a}_{i,j}$.



Obrázek 38: Vyšetřování posuvu pixelu mezi dvěma obrazy.

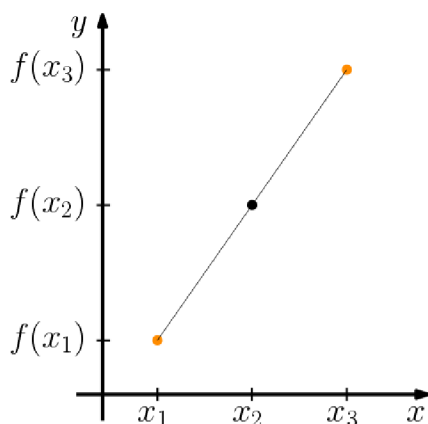
8.2 Interpolace

Smyslem interpolace je aproximovat danou funkci $g(x)$ funkcí $h(x)$, která v daných bodech nabývá stejných funkčních hodnot, případně má stejné hodnoty derivace a je jí pokud možno co nejpodobnější. Jestliže funkce $h(x)$ aproximuje funkci $g(x)$, píšeme $g(x) \approx h(x)$.

Více informací o této problematice lze nalézt v [1]. V dalším textu se budeme věnovat pouze interpolaci dvou funkčních hodnot po přímce, tzv. lineární interpolace, kterou následně rozšíříme na interpolaci funkce dvou proměnných s využitím interpolační roviny.

8.2.1 Lineární interpolace

Jedná se o nejjednodušší případ interpolace dvou hodnot, při němž jsou známé funkční hodnoty v daných bodech proloženy přímkou. Uvažujme tedy interval $\langle x_1, x_3 \rangle$, bod $x_2 \in \langle x_1, x_3 \rangle$ a funkční hodnoty na krajích tohoto intervalu, viz obrázek 39. Funkční



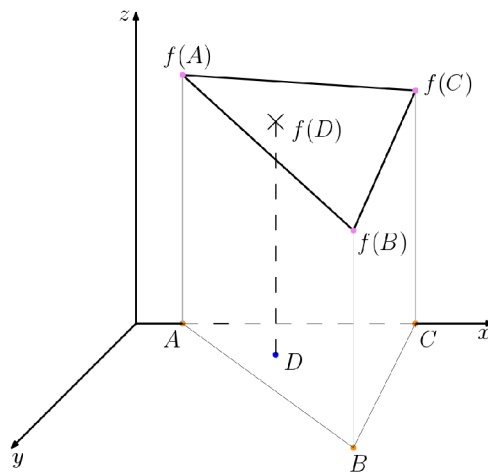
Obrázek 39: Lineární interpolace funkce jedné proměnné.

hodnotu v bodě x_2 pak snadno spočítáme dle rovnice

$$f(x_2) = f(x_1) + \frac{f(x_3) - f(x_1)}{x_3 - x_1}(x_2 - x_1).$$

8.2.2 Lineární interpolace funkcí dvou proměnných

Uvažujme nyní funkci dvou proměnných $g(x, y)$ definovanou na oblasti Ω , u níž známe funkční hodnoty v předepsaných bodech. Chceme-li interpolovat funkční hodnoty bodů v této oblasti, pak ke každému z nich najdeme tři nejbližší body se zadanou funkční hodnotou, takové, aby interpolovaný bod ležel na hranici nebo uvnitř trojúhelníku daném těmito třemi body, viz obrázek 40, kde je zmíněný trojúhelník dán body $A = [x_A, y_A]$, $B = [x_B, y_B]$, $C = [x_C, y_C]$, jejichž funkční hodnoty známe.



Obrázek 40: Lineární interpolace funkce dvou proměnných.

Funkční hodnotu v bodě $D = [x_D, y_D]$ pak snadno spočteme z rovnice roviny procházející body A, B, C vyjádřenou v následujícím maticovém zápisu

$$\begin{vmatrix} x_D - x_A & y_D - y_A & f(D) - f(A) \\ x_B - x_A & y_B - y_A & f(B) - f(A) \\ x_C - x_A & y_C - y_A & f(C) - f(A) \end{vmatrix} = 0. \quad (8.1)$$

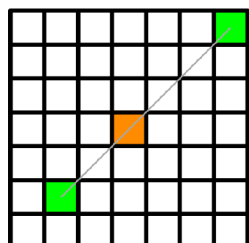
Je zřejmé, že pokud bod D leží na hraně trojúhelníku, postačí nám použít lineární interpolaci.

Samotnou trojici bodů k vytvoření interpolační roviny lze snadno najít následujícím postupem.

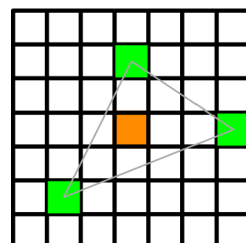
1. Určíme největší možný poloměr čtvercového okolí pixelu r_4 , jehož funkční hodnotu chceme interpolovat, tj. oranžový pixel na obrázcích 41, 42, 43. V tomto okolí hledáme dva nebo tři nejbližší body pomocí nichž jsme schopni provést interpolaci, tj. zelené pixely na obrázcích 41, 42, 43.
2. Pokud jsem našli první dva nejbližší body, pak ověříme, jestli vyšetřovaný pixel leží na jejich spojnici. Pokud leží, tak provedeme lineární interpolaci z těchto dvou hodnot, obrázek 41, v opačném případě pokračujeme v hledání třetího nejbližšího bodu.
3. V případě, že jsme našli trojici pixelů pomocí kterých lze aproximovat funkční hodnotu, obrázek 42, zkoumaného pixelu, pak postupujeme dle rovnice 8.1.

4. V situaci, kdy se nedaří nalézt třetí nejbližší bod ležící v okolí vytvořeném v bodě 1 pro vytvoření interpolační roviny, opakujeme postup od bodu 2, jen s tím rozdílem, že hledáme trojici prvního, třetího a čtvrtého nejbližšího pixelu, z nichž lze aproximovat funkční hodnotu v daném pixelu.

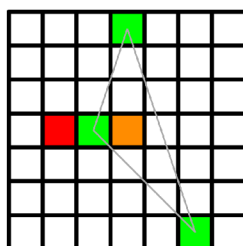
Poznamenejme, že pokud nevyhovuje trojice složená z prvního, třetího a čtvrtého nejbližšího bodu, tak postup opakujeme a hledáme první, čtvrtý a pátý nejbližší bod. Tato situace může nastat vícekrát a další postup je analogický postupu popsanému výše.



Obrázek 41: Lineární interpolace pomocí dvou nejbližších hodnot.



Obrázek 42: Lineární interpolace pomocí tří nejbližších hodnot.



Obrázek 43: Interpolační rovina je vytvořena z prvního, třetího a čtvrtého nejbližšího bodu. Červený pixel značí druhý nejbližší pixel, který však pro jeho nevhodnou polohu nelze k interpolaci použít.

Poznamenejme, že je důležité zvolit dostatečně velký poloměr, neboť v malém okolí nemusí ležet dvojice, nebo trojice bodů, které lze pro interpolaci použít. Zvolíme-li však příliš velké okolí, zvýší se časové nároky chodu programu. Je tedy nutné volit okolí „rozumné“ velikosti.

Odpověděli jsme tedy na otázku jak nalézt nevhodnější body k interpolaci. V poslední části této kapitoly odpovíme na otázku jak zjistit, zda bod D leží uvnitř trojúhelníku, jak je znázorněno obrázkem 40.

Nejdříve vypočteme následující rovnice a výsledky označíme písmeny ν_1, ν_2, ν_3

$$\begin{aligned}\nu_1 &:= (x_B - x_A) * (y_D - y_A) - (x_D - x_A) * (y_B - y_A), \\ \nu_2 &:= (x_C - x_B) * (y_D - y_B) - (x_D - x_B) * (y_C - y_B), \\ \nu_3 &:= (x_C - x_A) * (y_D - y_A) - (x_D - x_A) * (y_C - y_A).\end{aligned}\tag{8.2}$$

Pokud hodnoty ν_1, ν_2, ν_3 mají stejné znaménko, pak bod D uvnitř trojúhelníku leží.

Výpočet lze zjednodušit v případě, že jedna z hodnot ν_1, ν_2, ν_3 je rovna nule. V takovém případě to znamená, že bod D leží na spojnici dvou bodů, jimiž je trojúhelník určen. Například pro ν_1 za podmínek $x_A < x_B, y_A < y_B$ poté stačí ověřit podmínky

$$\begin{aligned}x_A &< x_D < x_B, \\ y_A &< y_D < y_B.\end{aligned}\tag{8.3}$$

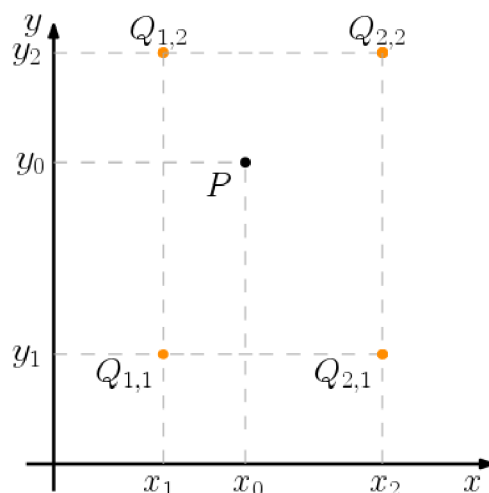
Je-li toto splněno, pak lze rovnou nasadit lineární interpolaci po přímce. Pro ν_2, ν_3 se postupuje analogicky.

Existuje mnoho jiných metod pro zjištění, zda bod leží uvnitř trojúhelníku a jsou snadno dohledatelné na internetu.

8.2.3 Bilineární interpolace

Jde o rozšíření lineární interpolace pro funkci dvou proměnných, přičemž body se zadanou funkční hodnotou tvoří pravidelnou obdelníkovou mříž. Hlavní myšlenkou této metody je provést nejprve lineární interpolaci v jednom směru a poté ve druhém.

Uvažujme body $Q_{1,1}, Q_{2,1}, Q_{1,2}, Q_{2,2}$, v nichž známe funkční hodnotu, se souřadnicemi dle obrázku 44.



Obrázek 44: Bilineární interpolace.

Přibližnou funkční hodnotu bodu P pak lze spočítat z rovnice

$$f(P) = f(Q_{1,1}) \frac{(x_2 - x_0)(y_2 - y_0)}{(x_2 - x_1)(y_2 - y_1)} + f(Q_{2,1}) \frac{(x_0 - x_1)(y_2 - y_0)}{(x_2 - x_1)(y_2 - y_1)} + f(Q_{1,2}) \frac{(x_2 - x_0)(y_0 - y_1)}{(x_2 - x_1)(y_2 - y_1)} + f(Q_{2,2}) \frac{(x_0 - x_1)(y_0 - y_1)}{(x_2 - x_1)(y_2 - y_1)}. \quad (8.4)$$

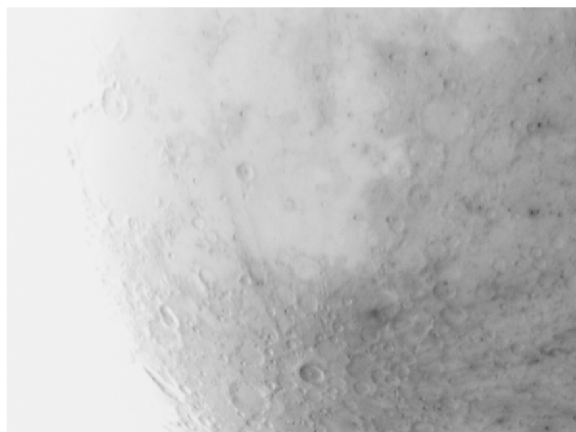
Toto byly poslední ukázky matematického aparátu, který budeme využívat ke zpracování poskytnutých dat. V další části textu bude popsán algoritmus pro zpracování dat zatížených seeingem.

9 Realizace tvorby softwaru pro zpracování dat po- stižených atmosferickým seeingem

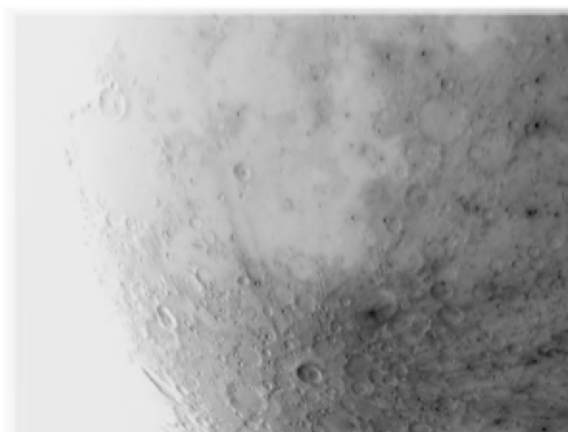
V první části této kapitoly nejdříve ilustrujeme vstupní data, která mají být programem zpracována. V druhé fázi bude popsán algoritmus řešení problému a celou kapitolu zakončíme vyhodnocením výsledků získaných z výstupu programu.

9.1 Vstupní data

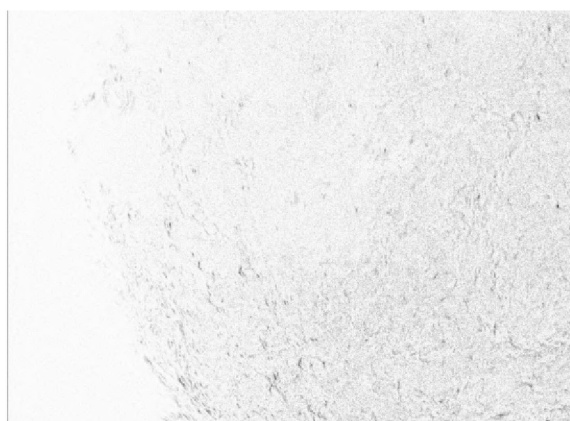
Vstupními daty, viz složka *Original data* v elektronické příloze práce, je šestnáct digitálních fotografií Měsíce, které jsou však znehodnocené vlivem geometrických deformací způsobených atmosferickým seeingem a aditivním šumem. Následující dva obrázky 45, 46 jsou negativy originálních fotografií a třetí obrázek 47 znázorňuje jejich rozdíl.



Obrázek 45: Originální snímek č. 156.



Obrázek 46: Originální snímek č. 500.



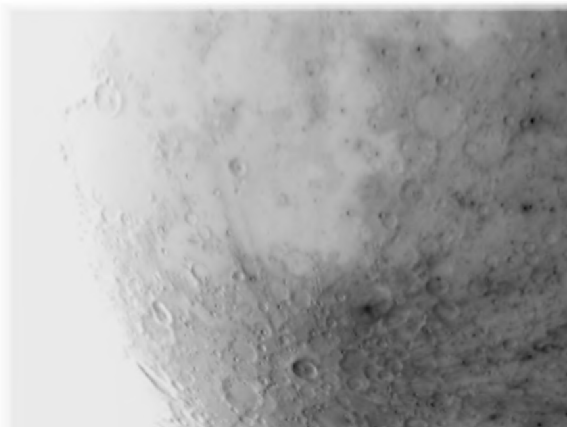
Obrázek 47: Vizualizace jejich rozdílů.

9.2 Postup řešení

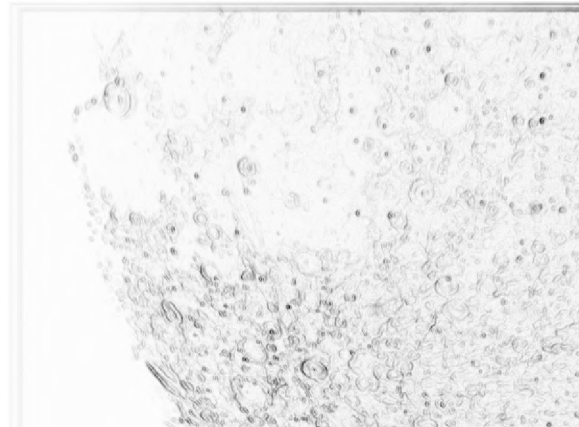
Algoritmus řešení je ilustrován diagramem na obrázku 66 v příloze 10.1. Každý z kroků nyní popíšeme.

Nejprve dle rovnice 5.2 vygenerujeme obraz $\bar{\mathbf{A}}$, který bude dán aritmetickým průměrem vstupních obrazů $\mathbf{A}_1, \dots, \mathbf{A}_n$, viz obrázek 48. Následně pomocí gradientního algoritmu detekce hran z kapitoly 6 vytvoříme obraz jeho hran \mathbf{B} , tj. obrázek 49.

Zvolíme kladné celočíselné parametry k , r a $MinJas$. Vybereme každý k -tý pixel ve vodorovném i svislém směru obrazu a prohledáme jeho čtvercové okolí o poloměru r . První nalezený pixel s hodnotou jasu větší, než je $MinJas$, uložíme do nového obrazu \mathbf{C} , viz obrázek 50. Máme tedy síť pixelů vytvořenou z hran zprůměrovaného obrazu. Nyní napočítáme posuvy těchto pixelů vzhledem k původním obrazům.



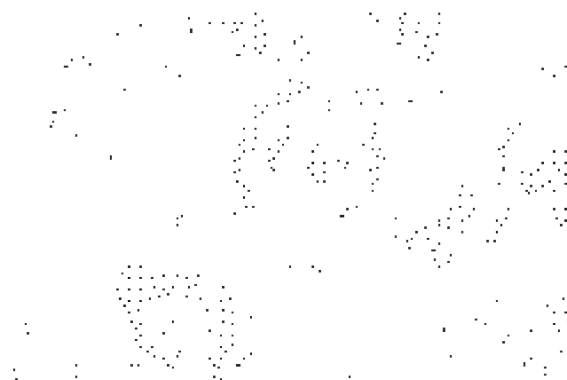
Obrázek 48: Zprůměrovaný obraz.



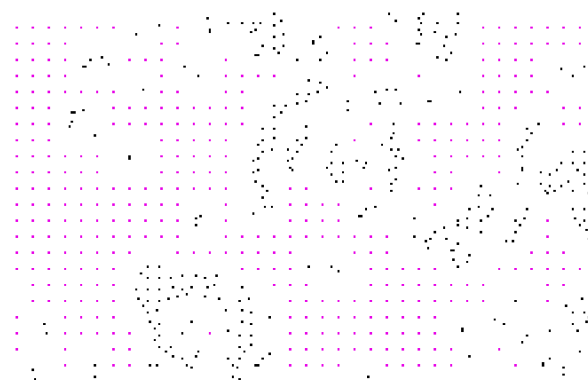
Obrázek 49: Hrany ve zprůměrovaném obrazu.

Zvolíme další kladné celočíselné parametry r_1, r_2 , které po řadě značí poloměr korelační masky a poloměr okolí, v němž hledáme body posunutí, viz podkapitola 8.1. V dalším kroku spočteme korelaci mezi obrazy $(\bar{\mathbf{A}}, \mathbf{A}_1), \dots, (\bar{\mathbf{A}}, \mathbf{A}_n)$ tak, jak je popsána v podkapitole 8.1, čímž získáme hodnoty posuvů jednotlivých pixelů. Poznamenejme, že v obraze $\bar{\mathbf{A}}$ pracujeme s pixely, jejichž souřadnice se shodují se souřadnicemi pixelů sítě v obraze \mathbf{C} .

Dalším krokem je interpolace hodnot posuvů pro zbývající pixely. Nejprve do sítě na obrázku 50 přidáme pomocné body, v nichž bude hodnota posuvů nulová, čímž získáme interpolační síť, jaká je na obrázku 51. Postup je snadný. Nejprve zvolíme kladné celočíselné parametry k_1 a r_3 a dále vyšetříme, zda okolí poloměru r_3 každého k_1 -ního pixelu ve vodorovném i svislém směru obsahuje bod původní sítě. Pokud ano, pak přejdeme k dalšímu bodu, v opačném případě takový pixel nahrajeme do interpolační sítě a přiřadíme mu nulovou hodnotu posuvu ve svislém i vodorovném směru. Poté přejdeme k samotné lineární interpolaci hodnot posuvů pro pixely s neznámou hodnotou vektoru posuvu, viz podkapitola 8.2.2. Obraz interpolační sítě označme symbolem \mathbf{C}^+ .



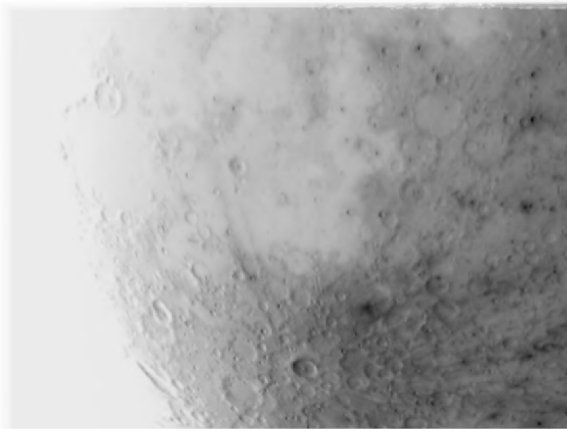
Obrázek 50: Detail sítě nejvýraznějších pixelů.



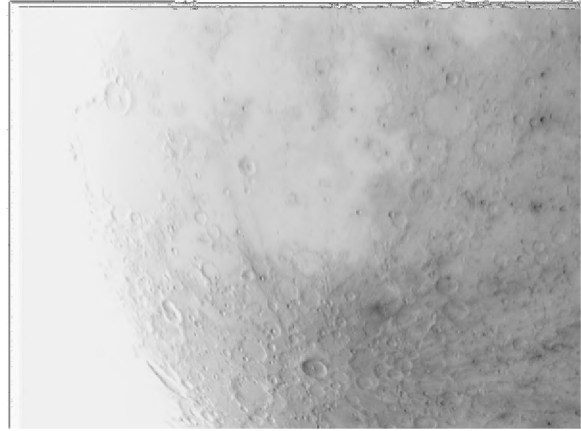
Obrázek 51: Detail interpolační sítě pixelů.

Posledním nutným krokem k získání konečného výsledku je provést bilineární interpolaci hodnot jasů z původních obrazů $\mathbf{A}_1, \dots, \mathbf{A}_n$. Hodnotu jasů pixelu získáme pomocí rovnice 8.4, kde souřadnice bodu P jsou dány součtem vektoru posunutí a souřadnic interpolovaného pixelu. Souřadnice zbylých bodů pak získáme vhodným zaokrouhlením souřadnic bodu P . Výstupem této bilineární interpolace je soubor obrazů $\mathbf{A}_1^+, \dots, \mathbf{A}_n^+$, z nichž následně znovu pomocí rovnice 5.2 vytvoříme zprůměrovaný obraz $\bar{\mathbf{A}}^+$ vykreslený na obrázku 52. Tento obraz následně zaostříme pomocí rovnice 5.3 s libovolným konvolučním jádrem

z kapitoly 7, čímž získáme výstupní obraz \mathbf{A}^{++} , viz obrázek 53. Nakonec je k tomuto obrazu vytvořen obraz jeho hran \mathbf{B}^{++} , tj. obrázek 54.



Obrázek 52: Zprůměrovaný upravený obraz.



Obrázek 53: Zaostrění obrazu 52.

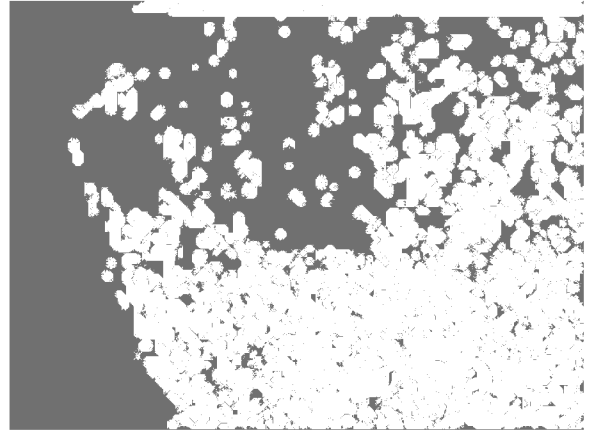
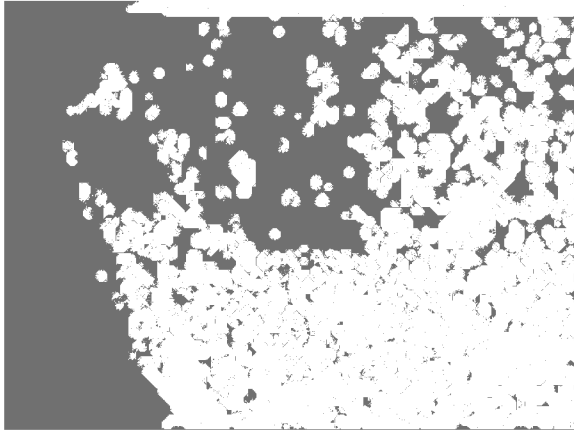


Obrázek 54: Vizualizace hran z obrázku 53.

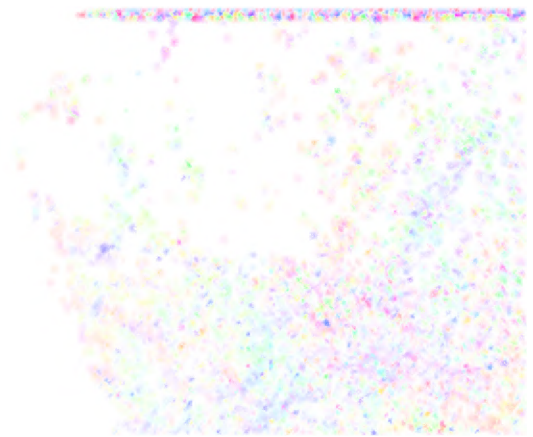
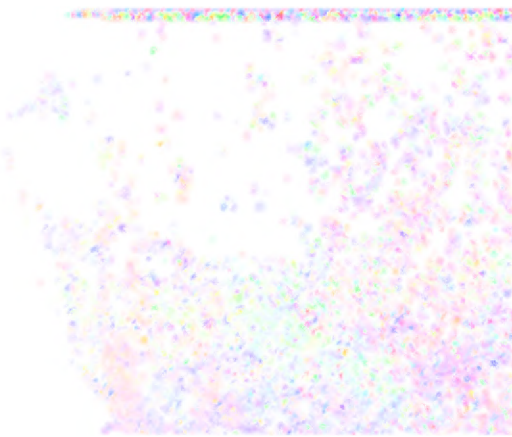
Před výpočtem bilineární interpolace lze po softwaru požadovat vykreslení mapy posuvů pixelů vstupních obrazů $\mathbf{A}_1, \dots, \mathbf{A}_n$ vzhledem k obrazu $\bar{\mathbf{A}}$. Na obrázcích 55, 56 jsou znázorněny pixely, v nichž nelze vyhodnotit hodnoty posunutí, tj. šedé pixely, zbylé pixely jsou znázorněny bíle. Jedná se o tzv. mapy seeingu a uživatel tyto obrázky nalezne ve složce *MapySeeingu*.

Software nám však také vrátí obrázky využívající barevný prostor *true color* k vizualizaci směru a velikosti posuvu pro každý pixel.

Tato mapa udává, v jakém směru a v jaké vzdálenosti leží původní pixel, např. obrazu \mathbf{A}_1 od obrazu $\bar{\mathbf{A}}$. Ukázkou mohou být obrázky 57, 58. Pro větší názornost přikládáme obrázky 59, 60, které znázorňují posuvy v oblasti na obrázku 51.



Obrázek 55: Nenulové posuvy pro obraz 45. Obrázek 56: Nenulové posuvy pro obraz 46.



Obrázek 57: Mapa posuvů pro obraz 45.

Obrázek 58: Mapa posuvů pro obraz 46.



Obrázek 59: Detail mapy posuvů obrazu 57. Obrázek 60: Detail mapy posuvů obrazu 58.

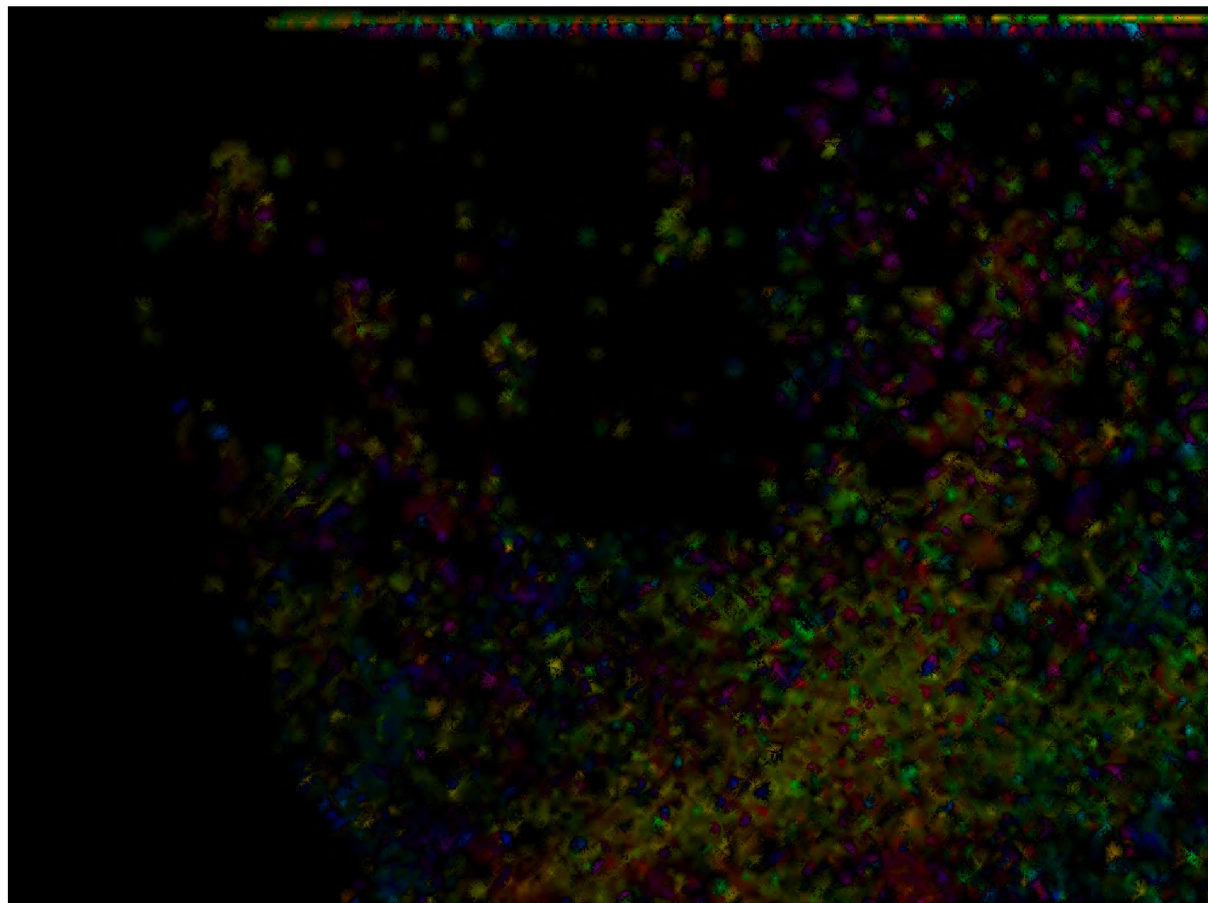
Postup tvorby těchto map posuvů je snadný a využívá rozložení barev znázorněné obrázkem 22. Nejprve spočteme normu vektoru posuvu každého pixelu a vybereme největší z těchto hodnot, označme ji N_{max} .

Dále procházíme každý pixel v obraze a počítáme normu vektoru jeho posuvu N_{pixel} a velikost kladně orientovaného úhlu mezi tímto vektorem a vodorovnou osou orientovanou zleva doprava, označme arg . Poté napočítáme⁴³ jednotlivé barvy RGB modelu pro daný pixel a vykreslíme jej. Poznamenejme, že pixely s největším posuvem budou vyznačeny barvou maximálního jasu.

Nelze si nevšimnout, že mapy posuvů jsou vykresleny na bílou plochu. Software nám

⁴³Algoritmus je popsán v příloze textu.

však vrátí⁴⁴ tytéž mapy posuvů jen s tím rozdílem, že jsou vykresleny na černý podklad. K této možnosti jsem se rozhodl z důvodu lepší zřetelnosti jednotlivých barev na monitoru. Pro tisk se však jeví jako lepší varianta používat bílé pozadí. Přesto pro názornost uvedme i možnost černého podkladu. Ukázkový výstup mapy posuvů, kterou software vrátí je na obrázku 61.



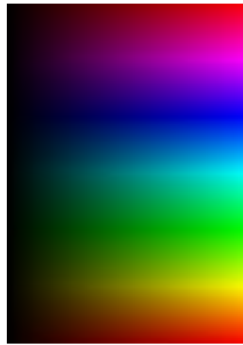
Obrázek 61: Mapa posuvů na černém pozadí.

Na obrázku 62 vidíme měřítko mapy posuvů. Software nám k tomuto měřítku vygeneruje i txt soubor s níže uvedeným popisem tohoto měřítku.

- Počátek je umístěn v levém dolním rohu obrázku.
- Vodorovná osa je orientovaná zleva doprava, přičemž maximální hodnota na této ose je rovna 255.
- Hodnoty na této ose znázorňují jas pixelu. Platí, že nejjasnější pixely jsou ty, které dosahují maximální hodnotu posunutí ve smyslu normy vektoru.
- Svislá osa je orientována od počátku směrem vzhůru, přičemž maximální hodnota je rovna 359.
- Hodnoty na této ose znázorňují kladně orientovaný úhel mezi vodorovnou osou a vektorem posunutí daného pixelu.
- Pomocí hodnoty na svislé ose a hodnoty na vodorovné ose, tedy lze určit posunutí daného pixelu.

⁴⁴Toto je standartní nastavení softwaru. Pokud uživatel požaduje vykreslení posuvů na bílé pozadí, postačí když potvrdí příslušný checkbox, viz kapitola 10.

- Maximální hodnotu posuvu nalezneme v souboru StatistickéVyhodnocení.txt.



Obrázek 62: Měřítko mapy posuvů.

Dalšími výstupy programu jsou textové soubory StatistickéVyhodnocení, Posuv, ParametryInfo. První zmíněný soubor obsahuje vybrané statistické údaje vztahující se k hodnotám posuvů pixelů. Uživatel zde nalezne hodnotu největšího posuvu, která je nutná k vyhodnocení map posuvů.

V úvodní části této práce jsme se seznámili s některými stupnicemi pro měření atmosferického seeingu. Společným rysem těchto stupnic byl fakt, že jsou sestaveny na subjektivním názoru pozorovatele a nedávají tak objektivní informaci. Pro sestavení nové stupnice bychom potřebovali objektivní kritéria, kterými mohou být například aritmetický průměr a rozptyl hodnot posuvů. Pomocí těchto dvou hodnot získáváme tedy objektivní informaci, kterou můžeme použít k vytvoření zcela nové exaktní stupnice pro měření atmosferického seeingu.

Upozorníme, že software pro výpočet těchto dvou hodnot používá hodnoty posuvů všech pixelů, tedy i těch, kde seeing nebyl detekován a je zde tedy nulová hodnota posuvů. Chce-li tedy uživatel zjistit průměrnou hodnotu a rozptyl posuvů je nutné, aby použil data bez pozadí a zároveň bylo v každém bodě možné detekovat seeing. Jde tedy zřejmě o náročný vstupní požadavek, a proto software vygeneruje textový soubor Posuv, který obsahuje složky vektoru posunutí každého pixelu z každého vstupního obrazu. Díky tomu má uživatel možnost odstranit nulové hodnoty posuvů, které jsou zřejmě způsobeny nedetekováním seeingu. Po této filtraci dat si uživatel snadno spočte aritmetický průměr a rozptyl hodnot posuvů, které vypovídají o stavu atmosferického seeingu⁴⁵. Toto je možno provést pomocí adekvátního statistického softwaru např. Minitab nebo MS Excel.

Poslední vygenerovaný textový soubor ParametryInfo nabízí přehled o zadaných vstupních parametrech.

Připomeňme, že uživatel má také možnost nechat si vykreslit měřítko pro mapu posuvů. Podrobný popis tohoto měřítka čtenář nalezne v současně vygenerovaném souboru PopisMěřítkoPosuvů.

9.3 Výstupní data

V předchozích dvou podkapitolách jsme se seznámili se vstupními daty a metodou řešení problému, přičemž jsme použili spoustu symbolických zkratk k označení jednotlivých výsledků např. $\bar{\mathbf{A}}$, \mathbf{A}^{++} atd. V této části přiřadíme jednotlivým symbolickým označením

⁴⁵Platí, že hodnoty posuvů jsou přímo úměrné kvalitě atmosferického seeingu. Velké posuvy vypovídají o velkém atmosferickém seeingu a opačně.

jejich textové názvy jakými jsou výstupní data pojmenována v elektronické příloze.

A_1, \dots, A_n – vstupní data, která lze nalézt ve složce *Original data*.

\bar{A} – *ZprumerovanyObraz*.

B – *Hrany_ZprumerovanyObraz*.

C – *SítZHran_Krok_7_polomer_4_PH_550*.

C^+ – *SítTriangulace_Krok_7_Polomer_4_PH_550_KrokPomSit_10_PolPomSit_7*.

A_1^+, \dots, A_n^+ – Srovnaná vstupní data uložená ve složce *Upravena data*.

\bar{A}^+ – *ZprumerovanyObrazUpraveny*.

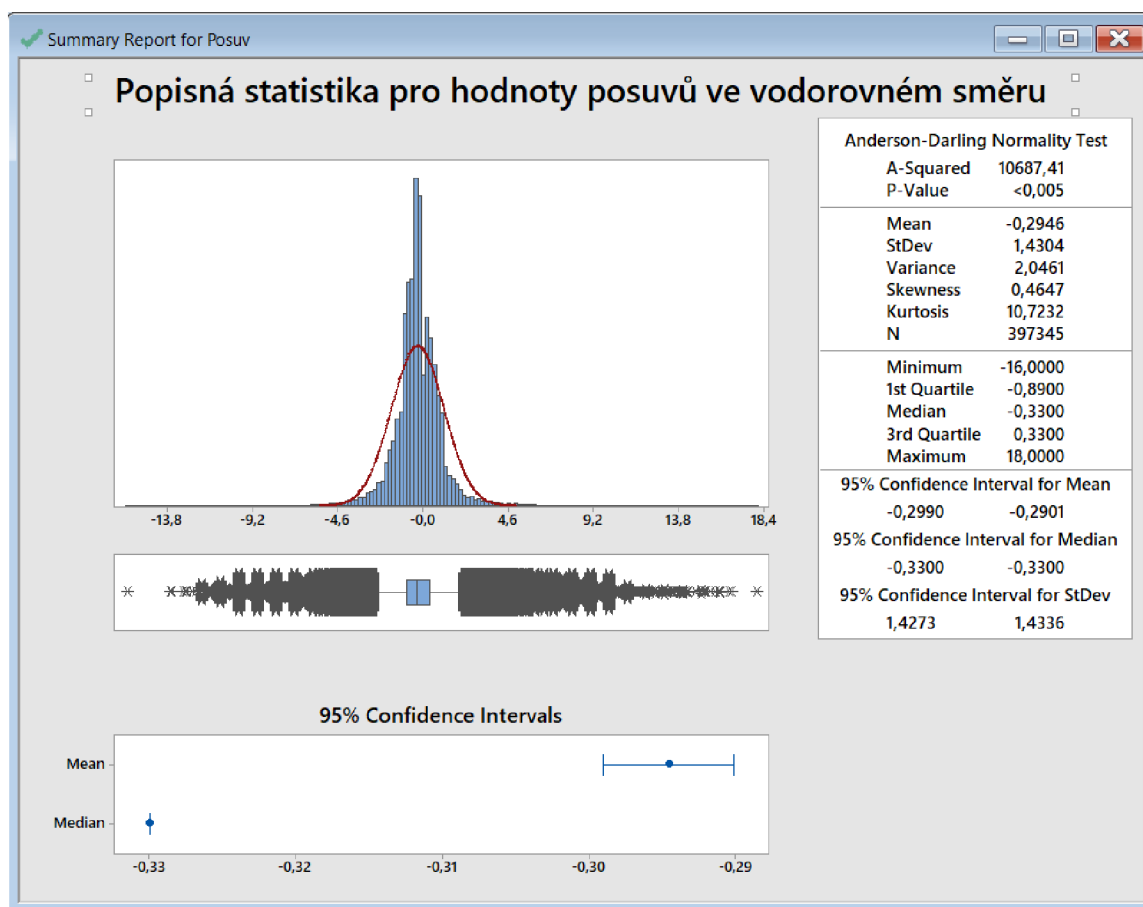
A^{++} – *ZprumerovanyObrazUpravenyZaostreny*.

B^{++} – *Hrany_ZprumerovanyObrazUpravenyZaostreny*.

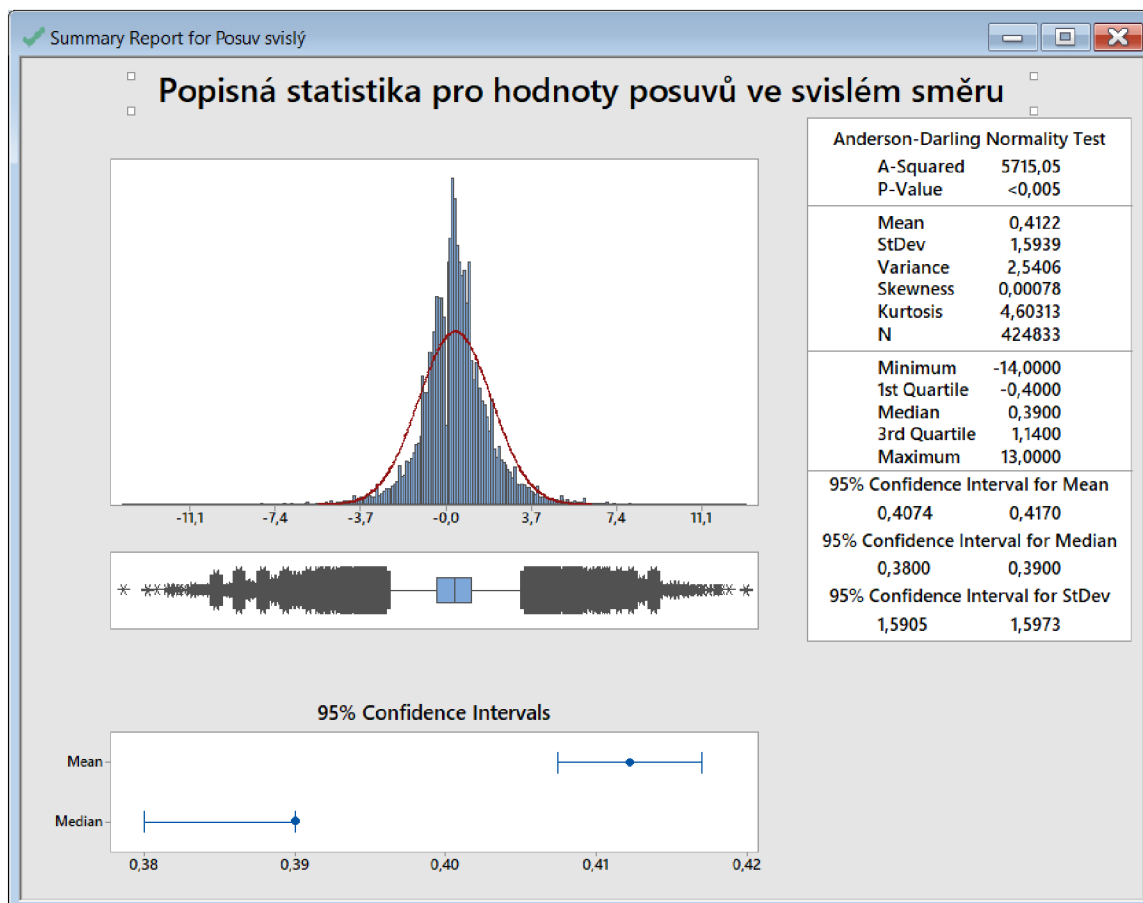
Mapy posuvů mezi obrazy $(\bar{A}, A_1), \dots, (\bar{A}, A_n)$ lze nalézt ve složce *Mapy posuvu*. Obrázky zvýrazňující místa, v nichž byl seeing úspěšně detekován jsou uložena ve složce *Mapy seeingu*.

9.4 Statistické vyhodnocení posuvů

Jak již bylo řečeno výše, textový soubor *Posuv* obsahuje hodnoty posuvu každého pixelu v každém obraze. Množství těchto dat přímo vybízí ke statistické analýze. Uveďme alespoň základní popisnou statistiku pro jeden vstupní obraz, kterou jsme získali pomocí softwaru Minitab, viz obrázky 63 a 64 .



Obrázek 63: Popisná statistika pro hodnoty posuvů ve vodorovném směru.



Obrázek 64: Popisná statistika pro hodnoty posuvů ve svislém směru.

Z obou p – hodnot plyne, že posuvy ve svislém ani vodorovném směru nemají normální rozdělení, přičemž pracujeme s 95% intervalem spolehlivosti

10 Popsání aplikace

V poslední řadě se věnujme popsání grafického rozhraní vytvořené aplikace.

V levé horní části okna se nachází položka *Tfilelistbox*, v níž označíme data, která chceme zpracovat. Vstupní obrázky musejí být ve formátu .tif. Uprostřed horní poloviny okna jsou rozmístěna pole pro zadání vstupních parametrů, kterými jsou přiřozená čísla. Význam jednotlivých vstupů byl popsán v předchozí kapitole, viz tabulka 3.

Parametry pro tvorbu sítě	Parametry pro tvorbu pomocné sítě	Parametry k výpočtu korelace
Krok k	Krok k_1	Poloměr korelační masky r_1
Poloměr r	Poloměr r_3	Poloměr okolí r_2
Prahová hodnota $MinJas$	Maximální poloměr okolí interpolace r_4	

Tabulka 3: Značení vstupních parametrů dle předchozí kapitoly.

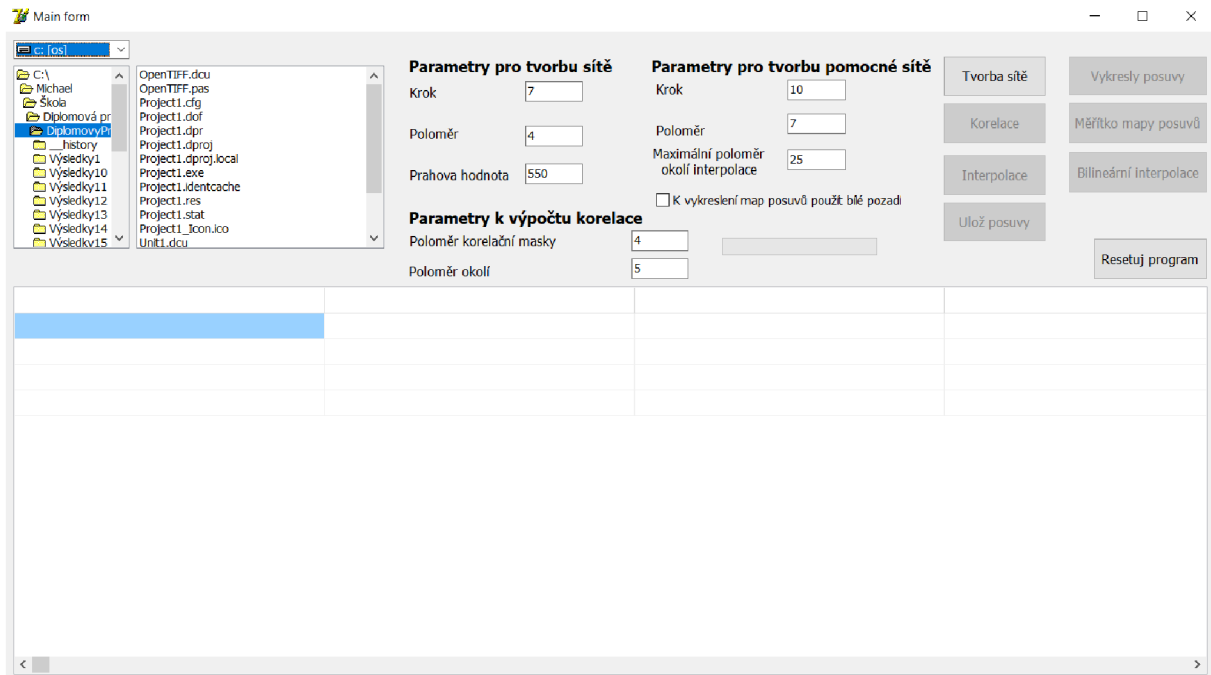
Po označení vstupních údajů a vyplnění vstupních parametrů spustíme výpočet postupným spouštěním akcí pod tlačítky *Tvorba sítě*, *Korelace*, *Interpolace*, *Bilineární interpolace*. Po stisknutí *Bilineární interpolace* se provede poslední nezbytný výpočet, uloží se výsledná data a celý program se zresetuje zpět do původní podoby.

Po stisknutí tlačítka *Interpolace* se odemknou tlačítka *Ulož posuvy*, *Vykresli posuvy*, *Měřítko mapy posuvů*, která dávají výstup v podobě textového souboru s uloženými posuvy každého pixelu v každém obrazu, map posuvů, map seeingu a měřítko mapy posuvů. Pomocí potvrzení checkboxu *K vykreslení map posuvů použít bílé pozadí* lze požadovat, aby se mapy posuvů vykreslily na bílé pozadí.

Posledním tlačítkem je *Resetuj program*, které po stisknutí celý program resetuje a nastaví parametry na původní hodnoty.

Spodní polovinu okna programu tvoří tabulka, do níž se po stisknutí *Vykresly posuvy* nahrají statistická data jako např. aritmetický průměr posuvů ve vodorovném a svislém směru, maximální hodnota posuvu pixelu v každém obrazu atd.

Všechny výsledky jsou uloženy do složky *Výsledky1, ..., VýsledkyN* obsažené v adresáři, z něhož je program spuštěn. K získání výsledků z předchozí kapitoly byly použity vstupní parametry z obrázku 65.



Obrázek 65: Software pro zpracování dat postižených atmosférickým seeingem.

Závěr

Diplomová práce se tedy zabývá problematikou zpracování obrazů postižených atmosférickým seeingem, s jehož problematikou byl čtenář seznámen hned v úvodu. Text dále pokračuje matematickým popisem obrazové matice a digitálního obrazu od jeho digitalizace až k metodám jeho zpracování jako například filtrace šumu, hledání hran a zaostření obrazu. V poslední části je vysvětlen algoritmus řešení problému a popsán vytvořený software, spolu s ukázkami výstupních obrazů.

Výstupem softwaru je tedy několik obrazů, ale za nejdůležitější lze považovat obraz s názvem *ZprumerovanyObrazUpravenyZaostreny.png*, který je výsledkem zpracování vstupních dat a znázorňuje, jak by měl obraz vypadat, kdyby nedocházelo ke znehodnocení dat vlivem atmosférického seeingu.

Dalším zajímavým výstupem jsou mapy posuvů znázorňující posuvy jednotlivých pixelů mezi originálními daty a obrazem z nich zprůměrovaným. Tyto obrazy čtenář nalezne ve složce *MapyPosuvů* v elektronické příloze. Za připomenutí stojí i složka *MapySeeingu*, která obsahuje obrázky znázorňující oblasti, v nichž byl atmosférický seeing detekován.

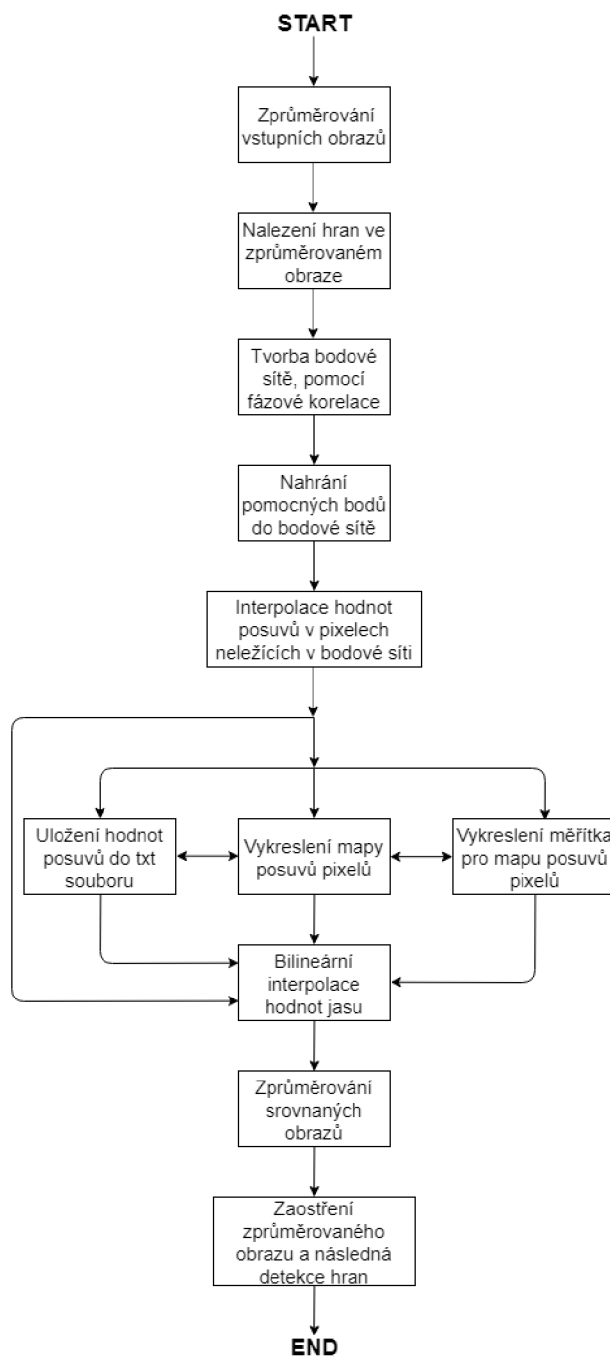
Stejně tak důležitými a zajímavými výstupy softwaru jsou textové soubory *Statisticke-Vyhodnoceni*, obsahující statistické údaje o vektorech posuvu jednotlivých pixelů, a *Posuv*, v němž uživatel nalezne vektor posuvu každého pixelu každého obrazu. Tento soubor dat lze využít k dalšímu statistickému zpracování dat a pomocí aritmetického průměru a rozptylu posuvů k sestavení nové objektivní stupnice pro měření atmosférického seeingu.

Reference

- [1] ČERMÁK, Libor a Rudolf HLAVIČKA. *Numerické metody*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2008. ISBN 978-80-214-3752-4.
- [2] ČIŠECKÝ, Roman. *Metody pro odstranění šumu z digitálních obrazů* [online]. Brno, 2012 [cit. 2018-07-28]. Dostupné z https://www.vutbr.cz/studenti/zav-prace?zp_id=52067. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Petr Číka.
- [3] DRUCKMÜLLEROVÁ, Hana. *Registrace obrazů pomocí fázové korelace* [online]. Brno, 2010 [cit. 2018-08-11]. Dostupné z https://www.vutbr.cz/studenti/zav-prace?zp_id=26464. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Mgr. Jana Procházková, Ph.D.
- [4] HLAVÁČ, Václav. a Miloš. SEDLÁČEK. *Zpracování signálů a obrazů*. 2. přeprac. vyd. Praha: ČVUT, 2007. ISBN 978-80-01-03110-0.
- [5] KLÍMA, Miloš. *Zpracování obrazové informace*. V Praze: České vysoké učení technické, 1996. ISBN 80-01-01436-3.
- [6] MARTIŠEK, Dalibor. *Matematické principy grafických systémů*. Brno: Littera, 2002. ISBN 80-85763-19-2.
- [7] ŠLAPAL, Josef. *Metody diskrétní matematiky*. Vyd. 2., upr. Brno: Akademické nakladatelství CERM, 2004. ISBN 80-214-2785-x.
- [8] VESELÝ, Vítěslav. *Funkcionální analýza II*.
- [9] ŽÁRA, Jiří, Bedřich BENEŠ a Petr FELKEL. *Moderní počítačová grafika*. Praha: Computer Press, 1998. ISBN 80-7226-049-9.
- [10] *Astronomical Seeing: Seeing Measurement Methods* [online]. 2013 [cit. 2019-03-12]. Dostupné z: <https://www.handprint.com/ASTRO/seeing2.html>

Přílohy

10.1 Vývojový diagram softwaru.



Obrázek 66: Vývojový diagram vytvořeného algoritmu.

10.2 Algoritmus pro vykreslení posuvů na černé pozadí

Písmenem N_{pixel} značíme normu vektoru posuvu pro aktuální pixel. Hodnotou N_{max} rozumíme největší normu vektoru posuvu v obraze. Velikost kladně orientovaného úhlu mezi vektorem posuvu a vodorovnou osou směřující zleva doprava symbolizuje arg . Písmena R, G, B pak značí červenou, zelenou a modrou složku RGB modelu z barevného prostoru *true colour*.

```
 $x \leftarrow \frac{N_{pixel}}{N_{max}} 255$   
R složka  
if  $arg \leq 60$  or  $arg \geq 300$  then  
     $R \leftarrow round(x)$   
else if  $(arg > 60)$  and  $(arg < 120)$  then  
     $R \leftarrow round(-\frac{x}{60}arg + 2x)$   
else if  $(arg > 240)$  and  $(arg < 300)$  then  
     $R \leftarrow round(\frac{x}{60}arg - 4x)$   
else  
     $R \leftarrow 0$   
end if  
G složka  
if  $arg \geq 0$  and  $arg < 60$  then  
     $G \leftarrow round(\frac{x}{60}arg)$   
else if  $(arg > 180)$  and  $(arg < 240)$  then  
     $G \leftarrow round(-\frac{x}{60}arg + 4x)$   
else if  $(arg \geq 60)$  and  $(arg \leq 180)$  then  
     $G \leftarrow round(x)$   
else  
     $G \leftarrow 0$   
end if  
B složka  
if  $arg > 120$  and  $arg < 180$  then  
     $B \leftarrow round(\frac{x}{60}arg - 2x)$   
else if  $(arg > 300)$  and  $(arg \leq 359)$  then  
     $B \leftarrow round(-\frac{x}{60}arg + 6x)$   
else if  $(arg \geq 180)$  and  $(arg \leq 300)$  then  
     $B \leftarrow round(x)$   
else  
     $B \leftarrow 0$   
end if
```