

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ  
ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF MATHEMATICS

# MĚŘENÍ RYCHLOSTI OBJEKTU POMOCÍ FOURIEROVY TRANSFORMACE

OBJECT SPEED MEASURING USING THE FOURIER TRANSFORM

DIPLOMOVÁ PRÁCE  
DIPLOMA THESIS

AUTOR PRÁCE  
AUTHOR

TOMÁŠ ZIKMUND

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. PAVEL ŠTARHA, Ph.D.



**Abstrakt**

Práce se zabývá měřením rychlosti objektu pomocí obrazové analýzy. Čtenář se zde seznámí s matematickou teorií spojenou s tímto problémem. Je zde vysvětlen matematický postup, kterým lze rychlost daného objektu vypočítat. Dále jsou v práci uvedeny výsledky reálného měření, které jsme dostali pomocí námi vytvořeného programu.

**Abstract**

This work deals with object speed measuring using image analysis method. The reader will become familiar with a mathematical theory upon which the method is based. The mathematical algorithm for obtaining the speed of a given object is illustrated. Furthermore, an original computer program has been developed and the results of real measuring are shown.

**Klíčová slova**

Fourierova transformace, fázová korelace, měření rychlosti

**Keywords**

Fourier transform, Phase correlation, speed measuring

ZIKMUND, T. *Měření rychlosti objektu pomocí Fourierovy transformace*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2008. 53 s. Vedoucí diplomové práce Ing. Pavel Štarha, Ph.D.



Prohlašuji, že jsem diplomovou práci *Měření rychlosti objektu pomocí Fourierovy transformace* vypracoval samostatně pod vedením Ing. Pavla Štarhy, Ph.D.; s použitím materiálů uvedených v seznamu literatury.

Tomáš Zikmund



Děkuji svému školiteli Ing. Pavlu Štarhovi, Ph.D. za vedení mé diplomové práce.

Tomáš Zikmund





# Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>Digitalizace</b>	<b>5</b>
2.1	Snímání . . . . .	5
2.1.1	Obrazová matice . . . . .	6
2.1.2	Barvy v obraze . . . . .	6
2.2	Vzorkování obrazu . . . . .	8
2.2.1	Ideální vzorkování . . . . .	9
2.3	Kvantizace obrazu . . . . .	11
<b>3</b>	<b>Bodové operátory a histogramy</b>	<b>12</b>
3.1	Histogram . . . . .	13
3.2	Typy převodních charakteristik . . . . .	13
3.2.1	Lineární transformace . . . . .	13
3.2.2	Nelineární transformace . . . . .	14
3.3	Vyrovnaní histogramu . . . . .	14
<b>4</b>	<b>Interpolační metody</b>	<b>16</b>
4.1	Metoda nejbližšího souseda . . . . .	16
4.2	Bilineární interpolace . . . . .	16
<b>5</b>	<b>Konvoluce</b>	<b>18</b>
5.1	Definice . . . . .	18
5.2	Diskrétní konvoluce . . . . .	18
5.3	Vybrané konvoluční filtry . . . . .	20
5.3.1	Filtr dolní propust . . . . .	21
5.3.2	Filtr horní propust . . . . .	21
5.3.3	Filtr pásmové propusti . . . . .	22
<b>6</b>	<b>Fourierova transformace</b>	<b>23</b>
6.1	Diskrétní Fourierova transformace . . . . .	23
<b>7</b>	<b>Fázová korelace</b>	<b>27</b>
7.1	Fázová korelace . . . . .	27
7.2	Aplikace fázové korelace . . . . .	28
<b>8</b>	<b>Měření rychlosti objektu za využití Fourierovy transformace</b>	<b>31</b>
8.1	Metody měření rychlosti . . . . .	31
8.2	Metoda měření rychlosti za využití změny úhlových rozměrů . . . . .	32
8.3	Metoda měření rychlosti za využití Fourierovy transformace . . . . .	34
8.3.1	Načtení fotografií . . . . .	34
8.3.2	Fourierova transformace . . . . .	36
8.3.3	Logaritmicko polární souřadnice . . . . .	39
8.3.4	Fázová korelace . . . . .	43
8.3.5	Výpočet rychlosti . . . . .	46

*OBSAH*

<b>9 Praktické ověření</b>	<b>47</b>
<b>10 Jednouúčelová aplikace</b>	<b>50</b>
<b>11 Závěr</b>	<b>51</b>
<b>12 Seznam použitých zkratek a symbolů</b>	<b>53</b>



# 1. Úvod

Cílem diplomové práce bylo popsání postupu, jak změřit rychlost objektu ze dvou snímků, na kterých je objekt zachycen v pohybu. K tomuto účelu se využije matematických nástrojů Fourierovy transformace a fázové korelace. Tento systém měření rychlosti by šel použít na jakýkoliv pohybující se objekt. My jsme se v této práci zabývali měřením rychlosti vozidla z důvodu jednoduchého ověření výpočtu. Dalším cílem bylo vytvořit jednoúčelovou aplikaci pro měření rychlosti daného objektu. Otázkou také bylo, jak přesně bude metoda na měření rychlosti fungovat. Práce se dá rozdělit do 3 větších částí. V první části se zabýváme matematickou teorií potřebnou k realizaci programu na výpočet rychlosti. V druhé části vysvětlujeme postup výpočtu rychlosti. A poslední částí je praktické měření.

Část první:

V druhé kapitole se zabýváme, jakým způsobem se obraz zpracovává do digitální formy. Popisují se zde procesy snímání, vzorkování a kvantizace obrazu. Jsou zde definovány barevné modely obrazu, pojem pixelu, aliasing.

Třetí kapitola je věnována jasové charakteristice obrazu. Jsou zde popsány základní typy jasových transformací, pomocí nichž lze jasovou charakteristiku tvarovat.

Čtvrtá kapitola popisuje dvě metody potřebné pro výpočet nových hodnot obrazových bodů v rámci předem definované mřížky.

V páté kapitole definujeme operaci konvoluce. Jsou zde popsány vlastnosti konvoluce a především vztah konvoluce s Fourierovou transformací. Je zde popsán postup, jak diskretní konvoluci naprogramovat. A dále jsou ukázány konvoluční filtry.

Šestá kapitola je věnována diskretní Fourierově transformaci, interpretaci výsledků po transformaci a definici amplitudového spektra.

V sedmé kapitole je definována fázová korelace, obsahující Fourierovu transformaci.

Část druhá:

Osmá kapitola popisuje postup výpočtu rychlosti na konkrétních snímcích. Využívá informací předešlých kapitol. Mimo jiné je zde podrobněji popsány lineární a logaritmická transformace a logaritmicko-polární transformace.

Část třetí:

Devátá kapitola je věnována výsledkům praktického měření. V desáté kapitole je zmínka o prostředcích, které byly využity k vytvoření programu.

Zdrojové kódy, program a série fotek z praktického měření jsou uloženy na příloženém CD.

## 2. Digitalizace

Kapitola popisuje zpracování obrazu a jeho převod do digitální formy, který je vhodný pro další zpracování na počítači.

### 2.1. Snímání

Vstupní informací při snímání nemusí být vždy jen jas z kamery či scanneru, ale mohou jí být i jiné veličiny, jako jsou intenzita rentgenového záření, ultrazvuk či tepelné záření. Dále však budeme uvažovat, že vstupním signálem je jasová složka z kamery. Snímání obrazu je převod optické veličiny na elektrický signál, který je spojitý v čase i v úrovni. V současné době se u digitálních přístrojů (tj. fotoaparátů či kamer) vyskytují dva druhy světlocitlivých snímacích prvků (viz. [3]).

#### *CCD snímače*

CCD snímače (Charge-Coupled Device) jsou malé destičky o úhlopříčce různých rozměrů složené z polovodičových elementů, které jsou schopny absorbovat světlo. Před expozicí jsou tyto elementy prázdné podobně jako neosvětlené políčko klasického filmu, v němž jsou ovšem místo polovodičových elementů mikroskopická zrnka halogenidů stříbra. Z toho je vidět, že princip získání obrazu je stejný, pouze je jinak technicky realizován. Po stisknutí spouště dojde po dobu nastavenou k osvětlení (expoziční doba) k tomu, že objektivem dovnitř pronikají světelné paprsky. Světlo si můžeme představit jako proud mikroskopických částíček, nazývajících se fotony. Fotony dopadají na snímač a jednotlivé polovodičové elementy (či buňky) tohoto prvku získají elektrický náboj. Z míst, která jsou jasnější, přichází větší proud fotonů, než z míst tmavých. Odtud plyne, že každý element získá jiný elektrický náboj a ten jsme schopni změřit. Samotný CCD snímač zaznamenává analogové hodnoty. Teprve obvody za ním provádí sčítání a digitalizaci.

#### *CMOS snímače*

CMOS snímače (Complementary Metal Oxid Semiconductor) využívají polovodičové součástky řízené elektrickým polem a k provozu jim stačí jen jedno napájecí napětí. Proto je jejich spotřeba velmi malá. Navíc je jejich technologie výroby poměrně laciná, protože se podobně vyrábí většina integrovaných obvodů. Také tyto snímače se dělí na dva druhy:

Pasivní CMOS snímače (PPS - Passive Pixel Sensors), které generují elektrický náboj úměrný energii dopadajícího svazku světelných paprsků. Náboj pak jde přes zesilovač do AD převodníku (analogově digitální převodník), stejně jako u CCD. V praxi však pasivní CMOS dávají díky šumu špatný obraz.

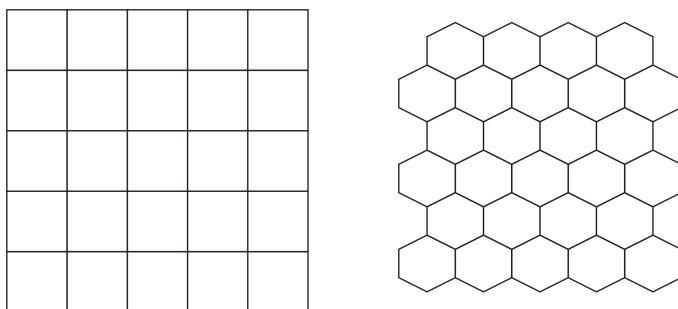
Aktivní CMOS snímače (APS - Active Pixel Sensors), u nichž je každý světlocitlivý element doplněn analytickým obvodem, který měří šum a eliminuje ho. Právě tyto snímače mají velkou budoucnost díky své nízké spotřebě energie a díky získávání podstatně většího barevného rozlišení obrazu.

## 2.1. SNÍMÁNÍ

### 2.1.1. Obrazová matice

Snímač zaznamenává analogové hodnoty. Proto je třeba zařídit převod spojité dvojrozměrné obrazové funkce do diskretní dvojrozměrné obrazové matice. Tento převod nazýváme digitalizací. Ta se skládá ze vzorkování (diskrétizace v ploše) a následné kvantizace (diskrétizace v úrovních).

Důležitou součástí digitalizace je volba vzorkovací mřížky. Nejčastěji používanými mřížkami jsou mřížky čtvercové a hexagonální. Jejich struktura je zobrazena na obrázku (2.2). Čtvercová mřížka vychází z konstrukce většiny snímacích prvků a je velmi snadno realizovatelná. Má však i své nevýhody týkající se především měření vzdáleností a spojitosti objektů. Hexagonální mřížka většinu těchto problémů řeší, ale není zase vhodná pro některé operace, jakými jsou např. Fourierova transformace. Dále budeme používat jen čtvercovou vzorkovací mřížku. Jeden prvek obrazové matice se nazývá obrazový bod. Často se nazývá pixel nebo pel. Poloha obrazového bodu je určena řádkovým  $j$  a sloupcovým  $k$ . Kapitola převzata z [4].

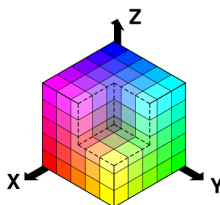


Obrázek 2.1: Obrazová matice

### 2.1.2. Barvy v obraze

Barevnost obrazu je důležitým nositelem informace. Pro určení přesného odstínu barvy jsou vytvořeny barevné modely. Ty se liší podle použitých složek (dle dokumentu [4]). Obrázky v této kapitole jsou převzaty z [11].

Model RGB

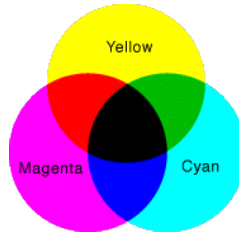


Obrázek 2.2: Obrazová matice

Tento popis barev je využíván v zobrazovacích zařízeních (monitor, TV), kdy jeden viditelný bod (pixel) tvoří tři velmi blízko umístěné body barevných složek Red, Green, Blue (červená, zelená a modrá). Jejich sloučením vzniká opticky jiná barva. Tento model barev je aditivní, tedy pro hodnoty 0,0,0 je výsledná barva černá. Se zvyšující se hodnotou se barvy přidávají až po maximální hodnotu 1,1,1, kdy je výsledná barva bílá. Platí,

že pokud jsou hodnoty jednotlivých složek stejné, jedná se o odstín šedi. V počítačové grafice se většinou uvažuje o rozsahu hodnot v celých číslech v rozmezí 0 – 255. V této souvislosti se hovoří i o hloubce barev, která právě udává číselné rozmezí pro jednotlivé složky. Barevná hloubka 8 bitů je tvořena výběrem (tabulkou) 256 definovaných barev. Výběr barev je buď definován (WEB safe colors – barvy využívané pro bezpečné zobrazení na internetu), nebo musí být definována k danému obrazu. U použití 24 bitové hloubky hovoříme již o TrueColor obrazech, obraz může obsahovat až 16 mil. barev, odpovídá tedy tomu, co je možno vidět v reálném světě.

### Model CMY a CMYK



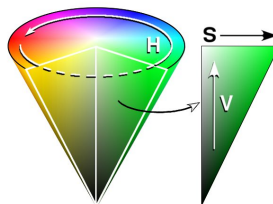
Tento barevný model je využíván pro tisk a při výrobě fotografií. Na rozdíl od RGB barevného systému se jedná o substraktivní systém. Barvy se neskládají, ale odečítají od původně bílé. Čím je tedy hodnota dané složky vyšší, tím se více blížíme k černé barvě. Vztah mezi RGB a CMY (Cyan, Magneta, Yellow – azurová, purpurová, žlutá):

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Platí tedy, že barva 1,0,0 v RGB je ekvivalentem 0,1,1 v CMY. Jinak řečeno RGB systém barvy vyzařuje, CMY pohlcuje.

V tiskárnách se většinou doplňuje CMY složkou K (blacK – černá), která se v tisku používá velmi často (pro černou se jinak musí nanést všechny tři barevné složky a výsledně složená barva je spíše šedá).

### Model HSI



Oproti předchozím modelům nejsou jeho složky tvořeny různými základními barvami, ale jejich třemi vlastnostmi: Hue, Saturation a Intensity mají význam: barva, sytost a jas.

- Barva je udávána jako úhel v rozmezí 0–360 (0 resp. 360 – červená, 120 – zelená, 240 – modrá). Barvy tvoří uzavřený kruh a jsou lineární. Lze tedy zadat libovolný úhel pro určení požadované barvy.

## 2.2. VZORKOVÁNÍ OBRAZU

- Sytost je závislá na množství přidané bílé složky.
- Jas určuje, kolik světla daná barva odrazí – jak jasná (zářivá) tedy bude.

Použití tohoto barevného modelu je asi nejlépe vidět v případě obrazů, kdy se používají barvy, do kterých se přidává černá a bílá pro vytvoření jejich odstínů.

### Model YUV

Tento model se v obrazovém zpracování nepoužívá. Jedná se o systém, který má uplatnění v televizní technice. Y představuje jas nebo šedivost a U a V jsou barevné složky.

### Šedá škála

V některých aplikacích postačí obraz v šedé škále. Většinou se využívá 256 odstínů, tedy 8 bitů hloubky. To je v rozmezí bílá – stupně šedi – černá.

## 2.2. Vzorkování obrazu

Vzorkování (anglicky *sampling*) je výraz pro získávání diskrétních vzorků ze spojitých dat prostoru. V počítačové grafice se tedy jedná o přechod analogického (spojitý, optický obraz) obrazu k digitálnímu. Obraz je v počítači reprezentován diskrétně. Je vždy omezen určitými parametry. Pro nás je to nejčastěji rozlišení monitoru, které způsobuje, že výsledný obraz je tvořen malinkými čtverci neboli vzorky obrazu.

Vzorkování obrazu si lze představit jako rozdělení obrazu na čtverce, kde část obrazu v jednom čtverci nahradíme jednou reprezentační hodnotou. Ta bude stejná pro všechny body čtverce. Tím však může dojít k omezení rozlišovací schopnosti. Při nesprávném vzorkování může dojít nejen k omezení rozlišovací schopnosti, ale i k naprostému znehodnocení obrazu (převzato včetně obrázků z [1]).

Jednou z nejčastějších chyb je **aliasing**. Tento jev znají snad všichni, neboť ho lze vidět na pohybu kol u rozjíždějícího se auta v televizi. Ze začátku se kola točí normálně pak chvíli stojí a pak se dokonce točí naopak. To vše jsou příčiny tohoto druhu aliasingu. Aliasing lze jednoduše nazvat jako podvzorkování. Pokud je vzorkovací frekvence nižší než je dvojnásobek maximální frekvence funkce obrazu (viz. Shannonův vzorkovací teorém), navzorkovaný výsledek neodpovídá skutečnosti.

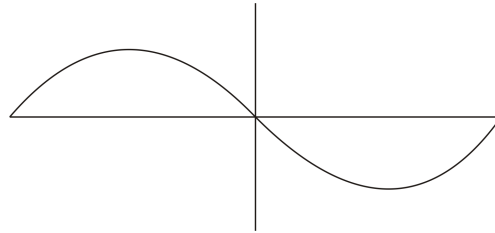
### *Shannonův teorém* (Nyquistův-Shannonův teorém)

Přesná rekonstrukce spojitého, frekvenčně omezeného, signálu z jeho vzorků je možná tehdy, pokud byl vzorkován frekvencí alespoň dvakrát vyšší než je maximální frekvence rekonstruovaného signálu. Poznamenejme, že hraniční vzorkovací frekvence se nazývá Nyquistova frekvence.

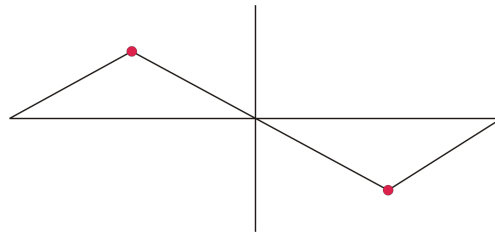
Ukažme si aliasing na jednoduchém příkladu. Vezměme si funkci  $\sin(x)$  a tu budeme chtít nakreslit. Pro kreslení v počítači ale musíme znát hodnotu funkce v každém bodě. Proto pustíme časovou osu a budeme odebírat vzorky (vzorkovat). Pokud bude  $\sin(x)$  probíhat s malým kmitočtem, tak se nám podaří zobrazit funkci téměř správně, ale pokud by kmitočet vzrostl tak, že stihneme vzít jen 2 vzorky, tak podle toho kdy vzorky uděláme,



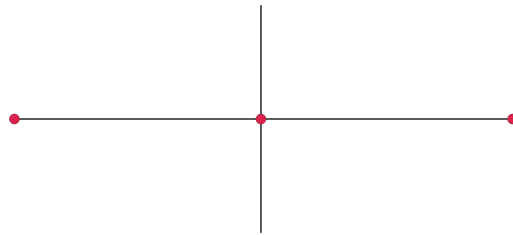
můžeme dostat např. dva trojúhelníky nebo jen rovnou čáru.



Obrázek 2.3: Při správném vzorkování



Obrázek 2.4: Při dvou vzorcích vhodně umístěných



Obrázek 2.5: Při vzorcích špatně umístěných

Na obrázcích (2.4), (2.5) jsou červeně označeny místa odebrání vzorku. Při dvou vzorcích, vhodně umístěných je vidět velké zkreslení. Při vzorcích, které jsou špatně umístěné, nelze již obraz rekonstruovat.

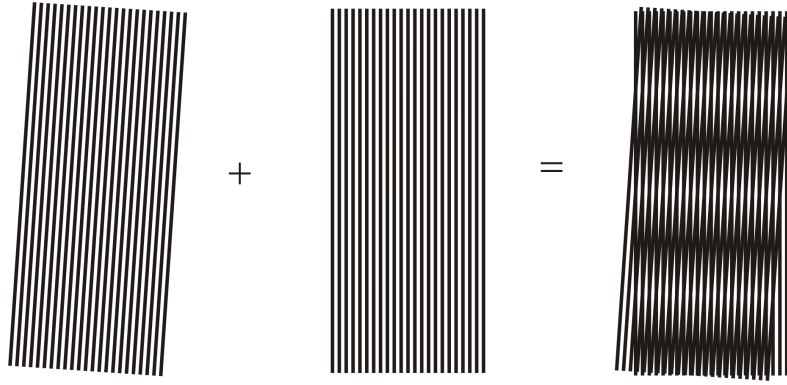
Moaré efekt se nazývá aliasing v obrazové technice. Může pokazit kvalitu snímků zřeznamenaných digitálním fotoaparátem. Vzorkujeme-li analogový obraz, který tvoří mírně šikmo položené pravidelné pruhy, periodickou vzorkovací strukturou o podobném kmitočtu, vytvoří se ve výsledném obrázku nová periodická struktura s dominantní periodou ve svislém směru (2.6). Metody, kterými se snažíme aliasing eliminovat se nazývají anti-aliasing.

### 2.2.1. Ideální vzorkování

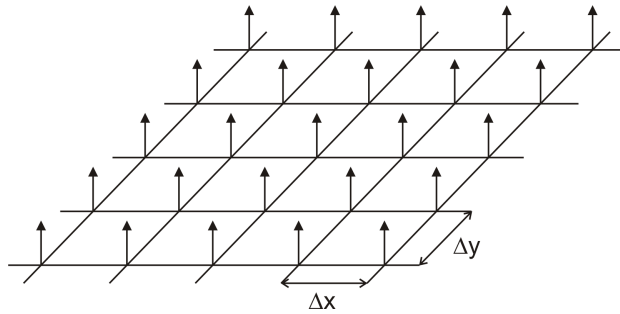
Kapitola převzata z [6]. Ideální vzorkovací funkci představuje nekonečné pole Diracových impulsů (obr. 2.7).

$$s(x, y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(x - j\Delta x, y - k\Delta y)$$

## 2.2. VZORKOVÁNÍ OBRAZU



Obrázek 2.6: Moaré efekt



Obrázek 2.7: Pole Diracových impulsů

Navzorkovaný obraz je dán součinem spojité obrazové funkce a vzorkovací funkce.

$$f_P(x, y) = f_I(x, y) \cdot s(x, y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f_I(j\Delta x, k\Delta y) \cdot \delta(x - j\Delta x, y - k\Delta y).$$

Z teorie integrální Fourierovy transformace (kapitola 6) víme, že součinu dvou funkcí v prostorové oblasti odpovídá konvoluce ve spektrální oblasti. Označíme-li spektrum spojité obrazové funkce  $F_I$  a spektrum vzorkovací funkce  $S$ , potom spektrum navzorkované obrazové funkce můžeme určit ze vztahu

$$f_P(\omega_x, \omega_y) = \frac{1}{2\pi} f_I(\omega_x, \omega_y) * S(\omega_x, \omega_y). \quad (2.1)$$

Spektrum nekonečného pole Diracových impulsů je opět nekonečné pole Diracových impulsů ve spektrální oblasti.

$$S(\omega_x, \omega_y) = \frac{2\pi}{\Delta x \Delta y} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(\omega_x - j\Omega_x, \omega_y - k\Omega_y),$$

kde  $\Omega_x = \frac{2\pi}{\Delta x}$  a  $\Omega_y = \frac{2\pi}{\Delta y}$  představují vzorkovací prostorové frekvence.

Dosažením do vztahu (2.1) a výpočtem konvoluce (kapitola 5) dostaneme vztah pro spektrum navzorkovaného obrazu

$$f_P(\omega_x, \omega_y) = \frac{1}{\Delta x \Delta y} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} F_I(\omega_x - j\Omega_x, \omega_y - k\Omega_y).$$

### 2.3. Kvantizace obrazu

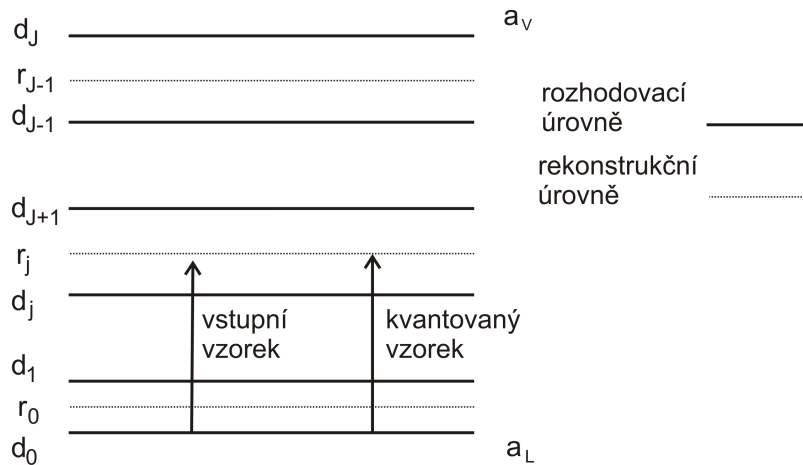
Dalším krokem k digitalizaci obrazu je kvantování. Toto je diskretizace v úrovních jasu, která má pouze jednorozměrný charakter oproti vzorkování. Bývá označována jako amplitudové kvantování (viz. [6]).

Amplitudovým kvantováním se rozumí reprezentace analogových plošně diskretizovaných vzorků s konečným počtem diskrétních úrovní. Postup je následující: Původní amplituda vzorku  $f$  se porovnává s hodnotami rozhodovacích úrovní. Jestliže amplituda vzorku leží mezi rozhodovacími úrovněmi  $d_i$  a  $d_{i+1}$ , pak se jí přiřadí rekonstrukční úroveň  $r_i$ .

Celý proces je znázorněn na (obr. 2.8), kde se předpokládá:

$$a_{min} < f < a_{max} ,$$

kde  $a_{min} = d_0$  je minimální amplituda vzorku a  $a_{max} = d_J$  je maximální amplituda vzorku.



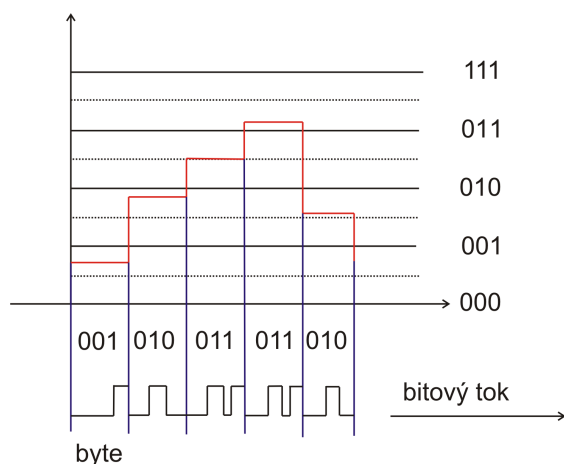
Obrázek 2.8: Princip kvantizace (převzato z [6])

Výšky vzorků se tedy dále nepřenášejí přesně, ale vždy jen s celou úrovní. Tím se do signálu zanáší chyba, která má vlastnosti šumu a nazývá se proto kvantizační šum. Další problém je nalezení kvantizační funkce, tj. stanovení počtu kvantizačních hladin a nalezení rozhodovacích a rekonstrukčních úrovní.

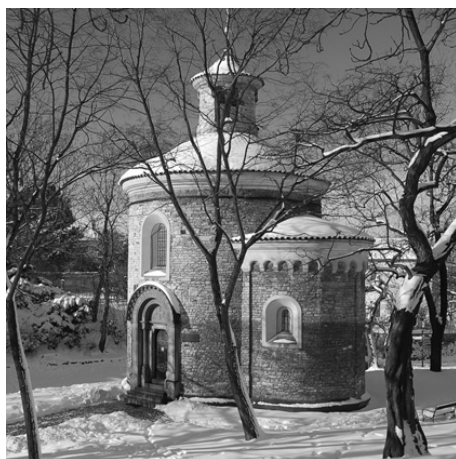
Pro jednoduchost se však v digitálních fotoaparátech používá rovnoměrná kvantizace, tzn., že všechny rozhodovací úrovně mají stejnou vzdálenost, a rekonstrukční úrovně jsou v polovině těchto intervalů. Počet kvantizačních úrovní je dán především kontrastní citlivostí zraku. Zrak lépe vyhodnotí změnu jasu sousedních vzorků, než jejich absolutní hodnotu.

Kvantovaný vzorek se dále binárně kóduje (viz. obrázek 2.9), tzn. jednotlivým kvantovacími úrovním se přiřadí binární číslo. Pokud je použito  $b$  bitů, potom je počet úrovní  $n = 2^b$ . V televizní a digitální technice se používá osmibitové kvantování na jeden obrazový element, což znamená  $2^8 = 256$  odstínů pro každý kanál R,G,B.

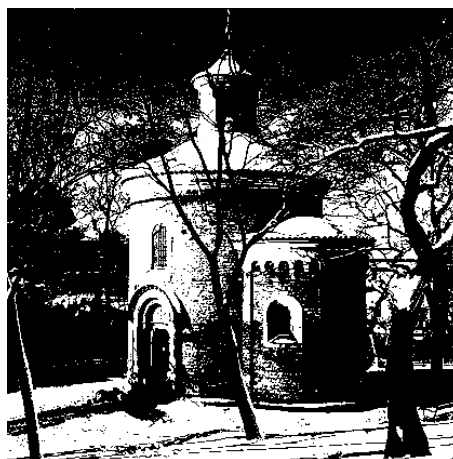
Jak se projeví na kvantovaném černobílém obrazu počet kvantovacích úrovní je ukázáno na obrázku (2.10). Obrázek vlevo je kvantován 8 bity, tedy 256 kvantizačními úrovněmi. U obrázku vpravo je použito kvantování 1 bitem, tedy 2 kvantizačními úrovněmi.



Obrázek 2.9: Princip kódování (převzato z [8])



(a) Obraz kvantován 8 bity



(b) Obraz kvantován 1 bitem

Obrázek 2.10: Ukázka rozdílného kódování

### 3. Bodové operátory a histogramy

Nejjednodušší operátory se týkají transformace hodnot jasu v příslušném obrazovém bodě bez ohledu na hodnoty jasu obrazových bodů v prostorovém nebo časovém okolí. Kapitola převzata včetně obrázků z [6].

Hodnota jasu  $a$  v obrazovém bodě o souřadnicích  $x, y$  je dána výrazem

$$a = a(x, y).$$

Bodový operátor je pak definován jednoduchou funkční závislostí

$$b(x, y) = f[a(x, y)],$$

kde  $b$  je hodnota jasu výstupního obrazu. Funkce  $f$  je zcela obecná a může být tedy lineární, nelineární, spojitá, nespojitá. Při vytváření převodní charakteristiky musíme však vždy zajistit správné využití dynamického rozsahu.

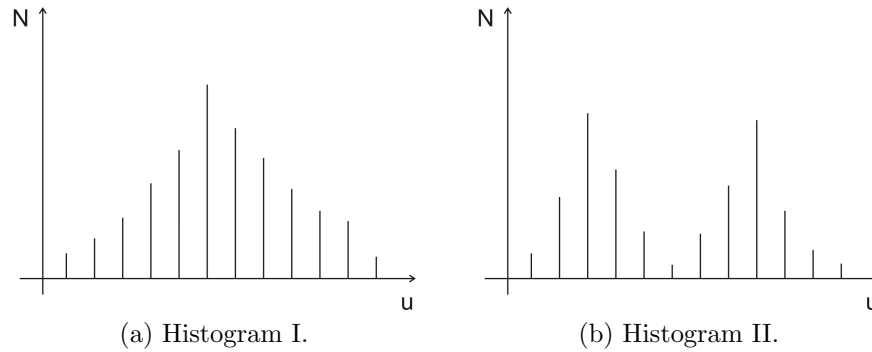
### 3.1. Histogram

Histogram, jinak také jasová charakteristika, je definován jako funkce četnosti výskytu určité úrovně

$$N = N(u), \quad (3.1)$$

kde  $u$  je příslušná úroveň,  $N$  je četnost výskytu úrovně  $u$ .

Histogram ukazuje, jak je využit dynamický rozsah a zda všechny úrovně jsou obsazeny se stejnou četností. Obvyklý tvar histogramu je uveden na obrázku (3.1.a).



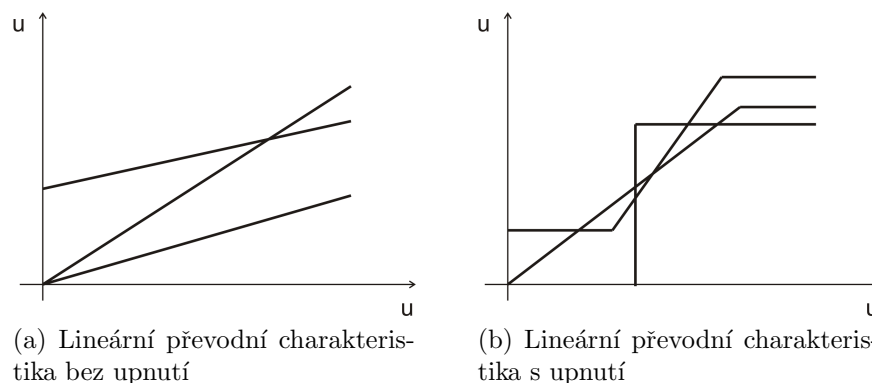
Obrázek 3.1: Histogramy

Velmi často se vyskytuje histogram se dvěma vrcholy (obr. 3.1.b), které odpovídají dvěma třídám objektů v obrazu lišících se jasnem, například buňky a pozadí. V případech nesprávné expozice se vrchol histogramu výrazně posouvá ze středu dynamického rozsahu směrem k bílému nebo černému okraji.

### 3.2. Typy převodních charakteristik

#### 3.2.1. Lineární transformace

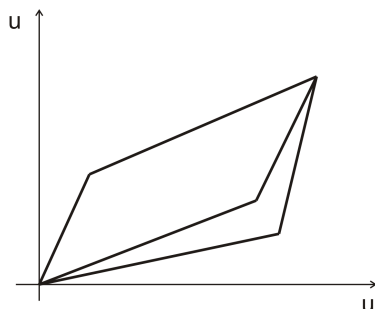
Rozlišujeme dva základní typy: bez upnutí (obr. 3.2.a) a s upnutím (obr. 3.2.b). Transformace sklonu bez upnutí je vhodná zejména tam, kde vstupní hodnoty nepokrývají plně dynamický rozsah. Často užívanou charakteristikou je charakteristika typu pryžová



Obrázek 3.2: Lineární převodní charakteristiky

páska, která zachovává úroveň bílé a úroveň černé při současné změně strmosti jednotlivých lineárních úseků (obr. 3.3).

### 3.3. VYROVNÁNÍ HISTOGRAMU



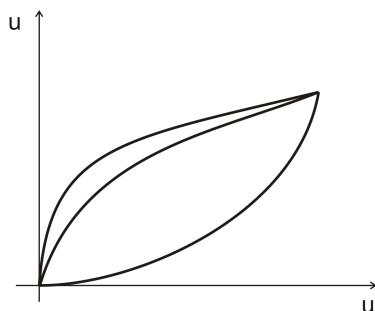
Obrázek 3.3: Převodní charakteristika typu pryžová páska

#### 3.2.2. Nelineární transformace

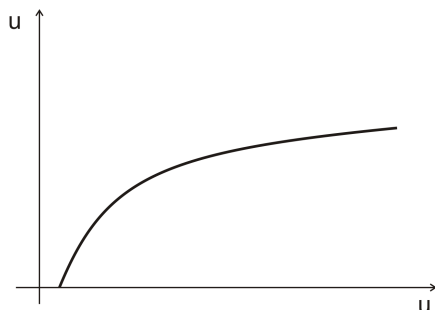
Z nelineárních převodních charakteristik patří k nejdůležitějším obecná mocnina (obr. 3.4) s různým exponentem  $a = 2, 1/2, 1/3$  a rovnicí

$$y = x^a.$$

Další významná charakteristika je logaritmická charakteristika (obr. 3.5).



Obrázek 3.4: Charakteristika obecné mocniny

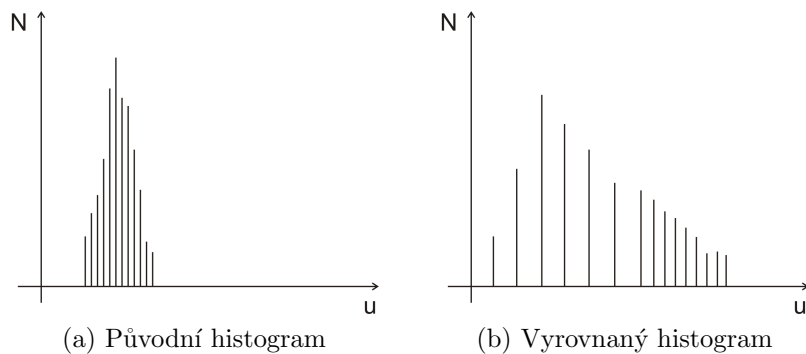


Obrázek 3.5: Logaritmická charakteristika

### 3.3. Vyrovnání histogramu

Vzhledem k tomu, že tvar převodní charakteristiky zásadním způsobem ovlivňuje tvar výsledného histogramu, lze převodní charakteristiku tvarovat tak, aby histogram získal požadovaný průběh. Základní myšlenkou vyrovnání histogramu je nelineární přerozdělení kvantizačních úrovní, které jsou v původním snímku obvykle rozděleny rovnoměrně v úrovních tak, aby jednotlivé nové kvantizační úrovně byly obsazeny pokud možno rovnoměrně s přibližně stejnou četností.

### 3. BODOVÉ OPERÁTORY A HISTOGRAMY



Obrázek 3.6: Vyrovnání histogramu

Vyrovnání histogramu používáme zejména u velmi úzkých průběhů histogramu, které zabírají jen velmi málo kvantizačních úrovní z celého dynamického rozsahu. Zde se nelineární převodní charakteristikou přerozdělí dílčí úrovně do sousedních tak, aby došlo k pokrytí celého dynamického rozsahu (obr. 3.6). Tento případ typicky nastává u podexponovaných nebo přeexponovaných snímků, kde je maximum hodnot obrazových bodů soustředěno v oblasti úrovně černé, resp. bílé.

## 4. Interpolační metody

Interpolační metody se používají pro výpočet nových hodnot obrazových bodů v rámci předem definovaného rastru (sít, mřížka). Toto je potřeba při změně geometrie obrazového pole, kde získáváme nové hodnoty obrazových bodů mimo souřadnice původního obrazového diskretizačního rastru. V literatuře se setkáváme s velkým počtem možných algoritmů, které jsou obvykle určeny pro specializované aplikace. Nyní uvedeme vybrané praktické příklady interpolačních postupů, které bychom mohli později využít. Kapitola převzata včetně obrázků z [6].

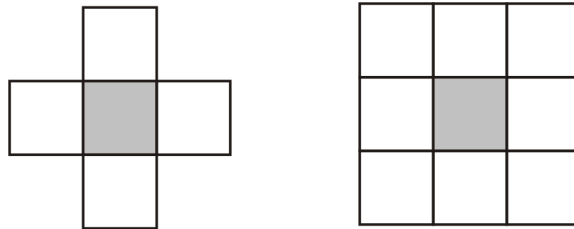
### 4.1. Metoda nejbližšího souseda

Tato metoda jednoduchým způsobem přiřazuje hledané hodnotě obrazového prvku hodnotu jeho nejbližšího souseda. Pojem nejbližší soused, tj. vzdálenost souseda, lze definovat různými způsoby a tím metodu modifikovat.

Vzdálenost dvou obrazových bodů udanými souřadnicemi  $(i,j)$  a  $(x,y)$  je obecně chápána jako Eukleidovská vzdálenost  $D_E$ , definovaná dle následujícího vztahu:

$$D_E = \sqrt{(x - i)^2 + (y - j)^2}.$$

Pokud je ale prostor diskretizován, tak definujeme nejprve sousedství bodu, a pak vzdálenost bodů v obraze za předpokladu daného sousedství. Při použití čtvercové vzorkovací mřížky můžeme uvažovat 4-sousedství nebo 8-sousedství.



Obrázek 4.1: 4-sousedství a 8-sousedství ve čtvercové mřížce

Vzdálenosti bodů jsou pro 4-sousedství je

$$D_4 = |x - i| + |y - j|.$$

Vzdálenosti bodů jsou pro 8-sousedství je

$$D_8 = \max\{|x - i|, |y - j|\}.$$

### 4.2. Bilineární interpolace

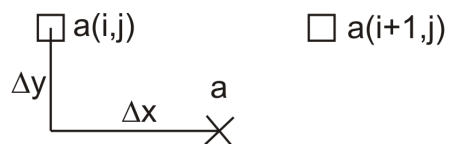
Jednou z nejčastěji používaných interpolačních metod je bilineární interpolace. Základní algoritmus vychází z výpočtu hodnoty mezilehlého obrazového bodu a na základě znalosti jeho čtyř nejbližších sousedů - bodů původního rastru.

Situace je znázorněna na obrázku (4.2), kde body  $a(i, j)$  jsou hodnoty sousedících pixelů. Hodnota v mezilehlém bodě  $a(x, y)$  je pak dána rovnicí (4.1).

$$a = (1 - \Delta x) [(1 - \Delta y)a(i, j) + \Delta y a(i, j + 1)] + \Delta x [(1 - \Delta y)a(i + 1, j) + \Delta y a(i + 1, j + 1)] \quad (4.1)$$



#### 4. INTERPOLAČNÍ METODY



Obrázek 4.2: Bilineární interpolace

Variantní interpolaci lze provést po náhradě bilineární funkce funkcí kubickou a výpočet hodnoty mezilehlého bodu z 16 sousedních bodů.

# 5. Konvoluce

V digitálním zpracování obrazu konvoluční filtrování hraje významnou roli v mnoha důležitých algoritmech v detekci okraje a příbuzných procesech. Kapitola převzata z [6].

## 5.1. Definice

Konvoluce je matematický operátor, který vezme dvě funkce  $f$  a  $g$  a produkuje třetí funkci. Konvoluce dvou funkcí  $f(t)$  a  $g(t)$  značíme symbolem  $*$  a je definována pomocí konvolučního integrálu

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau ,$$

který je funkcí jedné reálné proměnné. Pro funkce dvou reálných proměnných, kterými mohou být například právě obrazové funkce, je definice následující:

Konvoluce dvou reálných funkcí  $f(x, y)$  a  $g(x, y)$  je reálná funkce dvou proměnných definovaná integrálem

$$f(x, y) * g(x, y) = \iint_{\mathbb{R} \times \mathbb{R}} f(u, v)g(x - u, y - v)dudv.$$

Konvoluce má některé užitečné vlastnosti.

1.  $f * g = g * f$  komutativnost
2.  $C_1 f * C_2 g = C_1 C_2 (f * g)$  násobení konstantou
3.  $f * (g_1 + g_2) = f * g_1 + f * g_2$  distributivnost vůči sčítání
4.  $f_1 * [f_2 * f_3] = [f_1 * f_2] * f_3$  asociativnost

Hlavní význam konvoluce pro zpracování obrazu přináší její vztah k Fourierově transformaci. Je dán konvolučním teorémem o obrazu součinu dvou funkcí, resp. předmětu součinu dvou spekter. Nechť jsou dány dvě funkce a k nim přiřazeny Fourierovy obrazy (viz. kapitola 6)  $f(x, y) \sim F(\omega_x, \omega_y)$  a  $g(x, y) \sim G(\omega_x, \omega_y)$ . Platí:

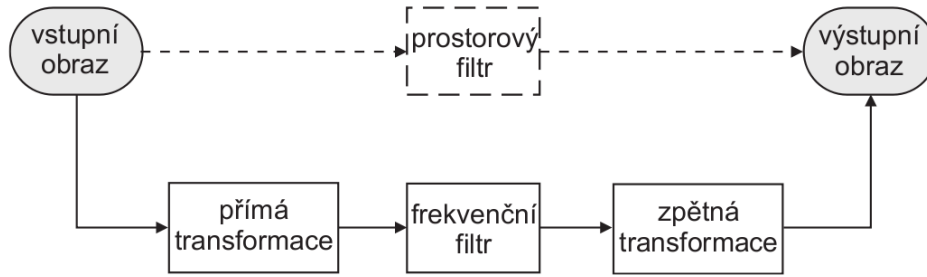
$$f(x, y) * g(x, y) \sim F(\omega_x, \omega_y)G(\omega_x, \omega_y) , \quad (5.1)$$

$$f(x, y)g(x, y) \sim \frac{1}{4\pi^2} F(\omega_x, \omega_y) * G(\omega_x, \omega_y) . \quad (5.2)$$

Tyto vlastnosti určují vztah mezi operacemi násobení a konvoluce v prostorové a spektrální oblasti. Konvoluci předmětů odpovídá součin jejich spekter a součinu předmětů odpovídá konvoluce spekter. Zvláště první vlastnost má široké použití v přechodu mezi spektrální a prostorovou oblastí. Běžná filtrace ve spektrální rovině totiž znamená vynásobení spektra filtrovaného obrazu kmitočtovou charakteristikou filtru.

## 5.2. Diskrétní konvoluce

Situace je zcela analogická v diskrétním případě. Jako řídicí proměnné používáme indexy v matici  $\mathbf{g}$ . Matice  $\mathbf{g}$  se nazývá konvoluční jádro, filtr nebo konvoluční maska. Situaci



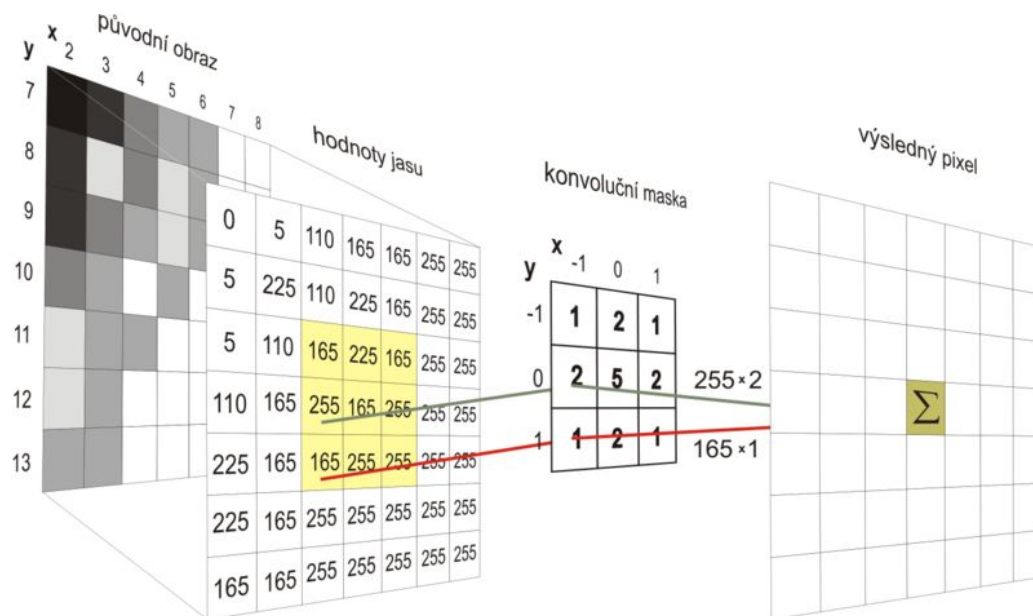
Obrázek 5.1: Filtrace dvourozměrných obrazků

zjednodušíme tím, že použijeme pouze konečnou čtvercovou vstupní matici. To je matice s konečným počtem prvků o rozměrech  $N \times N$ . Matici  $\mathbf{g}$  bude čtvercová  $L \times L$ .

$$f(x, y) * g(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k f(x+i, y+j)g(i, j), \quad (5.3)$$

kde  $1 \leq x \leq N$ ,  $1 \leq y \leq N$  a  $k = (L - 1)/2$ .

V případě diskrétní konvoluce lze jádro chápat jako tabulku, kterou položíme na příslušné místo obrazu. Každý pixel překrytý tabulkou vynásobíme koeficientem v příslušné buňce a provedeme součet všech těchto hodnot. Tím dostaneme jeden nový pixel. Tento postup je znázorněn na obrázku (5.2). Při programování operace konvoluce je třeba ošetřit



Obrázek 5.2: Diskrétní dvourozměrná konvoluce (převzato z [12])

situaci na okrajích obrazového pole, kde některé prvky matice  $g$  padnou při počítání mimo vstupní obrazové pole. Komplikovanému testování se lze vyhnout tím, že se zajistí, aby pro všechna  $(x, y)$  byly definovány všechny členy součtu. To znamená rozšířit vstupní pole o jakýsi obvodový rámeček pomocných bodů. Hodnoty v těchto bodech se nejčastěji volí rovny nule, pak se mluví o nulou rozšířené formě vstupního pole. Nebo se někdy volí hodnoty původního pole, které leží v polohách zrcadlově obrácených podle okraje obrázku, pak se mluví o zrcadlově rozšířené formě vstupního pole.

### 5.3. VYBRANÉ KONVOLUČNÍ FILTRY

Počátek indexů  $i, j$  leží ve středu matice  $\mathbf{g}$ , indexy  $i, j$  se pohybují v intervalech  $\langle -k, k \rangle$  a kryjí se středy vstupního a výstupního pole. Výstupní pole se rozšíří o rámeček kolem celého vstupního pole o šířce  $(L - 1)/2$  bodů. Tato definice konvoluce se nazývá centrovaná. Jinak řečeno, rozdíl spočívá v tom, do jakého místa se zapíše právě počítaný prvek výsledného pole. V centrované formě se zapisuje pod středový prvek matice  $\mathbf{g}$ . V tomto případě nastává menší nevýhoda záporných indexů matic. To lze odstranit pomocí vhodného posunutí indexů.

Koeficienty uvnitř konvoluční masky udávají vliv hodnoty pixelu pod nimi. Lze tak definovat velké množství operací. Tvar zadaného konvolučního jádra bývá

$$\frac{1}{K} \begin{bmatrix} g_{-k,-k} & g_{-k,-k+1} & \cdot & g_{-k,k} \\ g_{-k+1,-k} & \cdot & & \cdot \\ \cdot & & g_{0,0} & \cdot \\ \cdot & & \cdot & \cdot \\ g_{k,-k} & \cdot & \cdot & g_{kk} \end{bmatrix} = K^{-1}[g_{i,j}], \quad (5.4)$$

kde  $K^{-1}$  je konstanta úpravy dynamického rozsahu,  $L$  je dimenze jádra filtru.

Pro případ  $L = 3$  pak výpočet libovolného prvku  $b(x, y)$  výstupní matice je zapsán následujícím způsobem.

$$\begin{aligned} b(x, y) = & \frac{1}{K} f(x-1, y-1)g(-1, -1) + f(x, y-1)g(0, -1) + f(x+1, y-1)g(1, -1) + \\ & + f(x-1, y)g(-1, 0) + f(x, y)g(0, 0) + f(x+1, y)g(1, 0) + \\ & + f(x-1, y+1)g(-1, 1) + f(x, y+1)g(0, 1) + f(x+1, y+1)g(1, 1) \end{aligned}$$

Dimenze matice  $\mathbf{g}$  se liší podle požadovaných vlastností filtrů, nejčastěji se používají matice  $3 \times 3$ , velmi zřídka se používají větší než  $9 \times 9$ .

### 5.3. Vybrané konvoluční filtry

Nejdůležitější zásady při vytváření jádra konvoluce:

- Jádro by mělo mít nulové fázové spektrum (v opačném případě může měnit polohu i tvary objektu v obraze).
- Součet prvků konvoluční matice by měl být roven jedné (v opačném případě se mění jas obrazu).
- Zesílení vysokých frekvencí by nemělo překročit mez, při které začíná rušit aditivní šum.

V praktických případech se nenavrhují konkrétní filtry (konvoluční masky). Bylo by to příliš zdlouhavé. Operace často bývá zadána požadovaným subjektivním výsledkem (například zlepšení ostrosti), pro který nemusí existovat číselně vyjádřitelné jádro konvoluce. Proto se většinou používají již navržené a vyzkoušené filtry, ze kterých se subjektivním posouzením výsledku vybírá ten nejvhodnější.

Mezi nejčastěji používané filtry patří frekvenční filtry s charakteristikou horní, dolní nebo pásmové propusti. Jejich koeficienty jsou navrženy s ohledem na požadované propustné pásmo.

### 5.3.1. Filtr dolní propust

Dolní propust je filtr odstraňující z obrazu složky vyšších prostorových frekvencí. Takový charakter mají především detaily a aditivní šum. Konvoluční jádro typu dolní propust je charakteristické tím, že obsahuje pouze nezáporné členy. Součet prvků musí být roven 1, jinak obraz bude tmavý, resp. světlý. Typická dolní propust je filtr (5.5), který představuje průměr hodnot.

$$\mathbf{g} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (5.5)$$

Tento filtr je ukázán na obrázku (5.3).



Obrázek 5.3: Filtr dolní propust

Tento filtr nahrazuje původní hodnotu obrazového bodu střední hodnotou bodů na okolí o velikosti masky filtru. Dalším filtrem dolní propust je gausovské jádro, které má prvky matice, vzniklé aproximací dvourozměrné gausovy funkce, přičemž maximum leží v centrálním prvku matice. Frekvenční charakteristika se ovládá pomocí rozptylu. Pokud zvolíme malý rozptyl, pak se zbavíme nejvyšších frekvencí v obraze. Pokud je zvolen velký rozptyl, pak ořezává výrazně i střední frekvence.

### 5.3.2. Filtr horní propust

Horní propust je filtr odstraňující z obrazu složky nízkých prostorových frekvencí a ponechává vysoké frekvence. Ukázka filtru (obr. 5.4).

$$\mathbf{g} = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (5.6)$$

Středový prvek filtru horní propust je vždy kladný a kolem něj jsou symetricky rozmístěny záporné prvky. Musí platit  $\sum g(i, j) = 0$

### 5.3. VYBRANÉ KONVOLUČNÍ FILTRY



Obrázek 5.4: Filtr horní propust

Dalším takovým typem je jádro  $H = D - G$ , kde  $D$  je Diracovo jádro, které má centrální prvek hodnotu 1 a ostatní prvky hodnotu 0. Matice  $G$  je Gausovské jádro, tj. otočená gausova dvourozměrná funkce.

Tento filtr se používá zejména k hledání hranic objektů a k vypreparování vysokofrekvenčních struktur.

#### 5.3.3. Filtr pásmové propusti

Pásmová propust propouští pouze vybrané složky prostorových frekvencí, zpravidla uprostřed frekvenčního rozsahu nebo jen v určitých izolovaných oblastech spektra. Je často používána k rekonstrukci obrazu, například k odstranění rušivých vertikálních interferencí vzniklých snímáním. Matice filtru mají rozměr nejméně  $5 \times 5$  a uprostřed je kladný koeficient obklopený symetricky rozmístěnými zápornými koeficienty, jejichž absolutní hodnota klesá směrem k okraji. Kapitola převzata z [6].

## 6. Fourierova transformace

Fourierova transformace, která se běžně používá při zpracování signálu, představuje ve své dvourozměrné verzi důležitý pomocný prostředek při digitálním zpracování obrazové informace. Obecně se obrazové transformace využívají na to, že obraz je převeden na tvar, který je pro dané zpracování podstatně výhodnější.

### 6.1. Diskrétní Fourierova transformace

Diskrétní dvourozměrná přímá Fourierova transformace periodické dvourozměrné obrazové funkce je definována vztahem

$$F(u, v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left\{ -j \frac{2\pi}{N} (ux + vy) \right\} \quad (6.1)$$

a zpětná transformace

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp \left\{ j \frac{2\pi}{N} (ux + vy) \right\}, \quad (6.2)$$

protože komplexní transformační jádro

$$W(x, y, u, v) = W_1^{xu} W_2^{yv},$$

kde

$$W_1 = W_2 = \exp \left( -j \frac{2\pi}{N} \right)$$

je separovatelné, dvourozměrná transformace může být vypočtena jako jednorozměrná transformace všech řádků obrazové matice následovaná jednorozměrnou transformací všech sloupců mezivýsledku podle vztahu

$$F(u, v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \left[ \sum_{y=0}^{N-1} f(x, y) \exp \left( -j \frac{2\pi}{N} vy \right) \right] \exp \left( -j \frac{2\pi}{N} ux \right) \quad (6.3)$$

V další části si uvedeme některé vlastnosti diskrétní dvourozměrné Fourierovy transformace, důležité zejména z hlediska jejího použití pro zpracování obrazu.

#### Střední hodnota

Spektrální složka pro  $u = 0, v = 0$

$$F(0, 0) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)$$

reprezentuje průměrnou hodnotu všech obrazových bodů a nazývá se stejnoměrná složka obrazu.

#### Periodicita

Substitucí  $u = u^* + mN, v = v^* + nN$ , kde  $m, n$  jsou celá čísla, dostaneme

$$F(u^* + mN, v^* + nN) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left( -j \frac{2\pi}{N} (u^*x + v^*y) \right) \exp (-j2\pi(mx + ny)),$$

## 6.1. DISKRÉTNÍ FOURIEROVA TRANSFORMACE

ze které je zřejmé, že druhý exponenciální výraz je jednotkový pro všechny celočíselné hodnoty  $m, n$ , a tudíž je spektrum periodické

$$F(u^* + mN, v^* + nN) = F(u^*, v^*), \quad m, n = 0, \pm 1, \pm 2, \dots$$

Z předpokladu Fourierova rozvoje vyplývá, že vstupní obraz musí mít rovněž periodický charakter. Obraz má vždy konečnou velikost, proto podmínku periodicity snadno v praxi splníme tím, že předpokládáme jeho opakování ve svislém i vodorovném směru.

### Teorém posunu

Říká, že posun objektu v předmětovém prostoru vede k fázovému posunu ve spektrální oblasti podle vztahu

$$f(m - k, n - l) \Leftrightarrow F(u, v) \exp \left[ -j \frac{2\pi}{N} (ku + lv) \right]. \quad (6.4)$$

Fourierova transformace převádí obraz z prostorové oblasti do spektrální oblasti, ve které je zřejmé, z jakých spektrálních složek se obraz skládá. Každá spektrální složka je určena svojí amplitudou a fází. Věta o posunu vyjadřuje skutečnost, že ve spektrální oblasti fázová složka obsahuje informaci o umístění objektu v obrazu a je možné si ověřit, že její zanedbání vede k podstatně větší degradaci obrazu než zanedbání některých amplitudových složek.

### Symetrie

V teorii funkcí hrají významnou úlohu liché, resp. sudé funkce, které jsou definovány vztahem

$$F(m, n) = \pm F(-m, -n),$$

přičemž kladné znaménko platí pro sudé a záporné pro liché funkce. Bod  $N/2, N/2$  tvoří střed symetrie. Fourierova transformace zachovává symetrii, tedy transformace liché (resp. sudé) funkce je opět lichou (resp. sudou) funkcí. Tato skutečnost vyplývá přímo z vlastnosti jádra, které lze rozložit do sudé reálné (kosinové funkce) a liché imaginární části (sinové funkce) podle známého vztahu

$$\exp j(ux) = \cos ux + j \sin ux.$$

Diskrétními lineárními ortogonálními transformacemi se počet koeficientů matice nezmění. V případě Fourierovy transformace je výstupem matice komplexních čísel a zdá se tedy, že obsahuje dvakrát tolik koeficientů. Vzhledem k tomu, že je obrazová matice vždy reálná, tak ze symetrie transformace reálné matice vyplývá, že komplexní spektrum je plně určeno hodnotami jedné poloviny transformovaných vzorků. Druhá polovina je redundantní a může být vždy určena zrcadlením podél středu symetrie.

### Konvoluční teorém

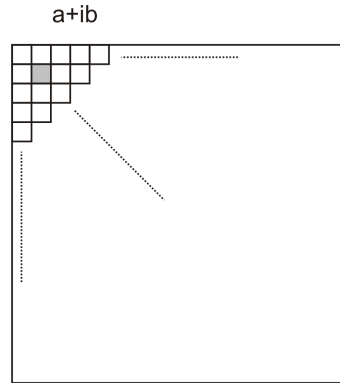
Umožňuje snadný výpočet konvoluce ve spektrální rovině, protože převádí diskretní konvoluci na prosté násobení matic (viz. kapitola 5).

Fourierova transformace je základní matematický nástroj ve zpracování digitálních obrazů. Používá se na detekci hran a segmentaci obrazu, k úpravě kvality obrazu, k rekonstrukci obrazu, ke kompresi obrazu a k detekci obrazu.



Fourierův obraz  $F(u, v)$  je funkce komplexní proměnné. Jinak řečeno výsledná matice po Fourierově transformaci se skládá z komplexních čísel (obr. 6.1). Nazýváme ho frekvenční spektrum, komplexní spektrum nebo jen spektrum. A lze ho je psát v maticovém zápisu

$$F(u, v) = R(u, v) + jI(u, v).$$



Obrázek 6.1: Obrazová matice po Fourierově transformaci

Předpokládejme, že jsme provedli jednorozměrnou Fourierovu transformaci a dostali jsme vektor komplexních čísel  $z_n = a_n + jb_n$ . Pak Fourierova řada je součtem funkcí  $\sin$  a  $\cos$ . Reálná část komplexního čísla je amplitudou u funkce  $\sin nx$  o frekvenci rovné pořadí čísla ve vektoru  $n$  a imaginární část je hodnota amplitudy funkce  $\cos nx$  o frekvenci rovné  $n$ .

$$a_0 + a_1 \cos x + b_1 \sin x + a_2 \cos 2x + b_2 \sin 2x + \dots + a_n \cos nx + b_n \sin nx + \dots$$

Použitím Moivreovy věty lze komplexní číslo přepsat na tvar

$$z_n = |z_n| \cos \varphi + j|z_n| \sin \varphi.$$

Pak lze každý člen Fourierovy řady přepsat následujícím způsobem:

$$a_n \cos nx + b_n \sin nx = |z_n| \cos \varphi \cos nx + |z_n| \sin \varphi \sin nx = |z_n| \cos(nx - \varphi).$$

Interpretace komplexního čísla ve dvourozměrném případě je už složitější.

Na obrázku (6.2) je znázorněno, jak frekvence u funkcí  $\sin$ ,  $\cos$  rostou.

Dále se ke zpracování obrazů používají různá spektra, definovaná následujícími vztahy:

### Amplitudové spektrum

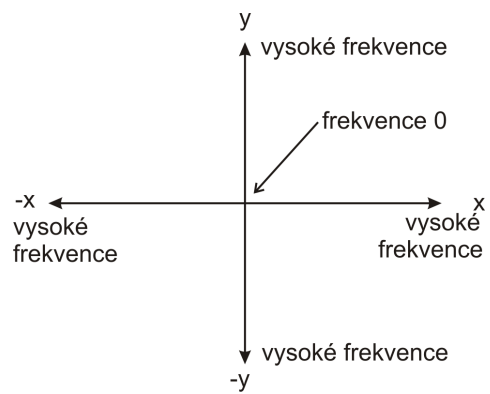
$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)} \quad (6.5)$$

Tento výraz znamená, že pro všechny prvky matice vypočítáme velikost komplexního čísla. Tím však dostáváme matici pouze reálných hodnot.

### Fázové spektrum

$$\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$$

## 6.1. DISKRÉTNÍ FOURIEROVA TRANSFORMACE



Obrázek 6.2: Frekvenční spektrum

### Výkonové spektrum

$$P(u, v) = |F(u, v)|^2 = R^2(x, y) + I^2(x, y)$$

## 7. Fázová korelace

Fázová korelace (phase correlation) patří mezi oblíbené metody registrace obrazu založené na Fourierově transformaci. Cílem metody je získat velikost posuvu mezi dvěma obrázky, které se částečně překrývají.

### 7.1. Fázová korelace

Myšlenka fázové korelace je založená na Fourierově posuvném teorému a také na skutečnosti, že dva obrázky s jistým stupněm podobnosti tvoří v jejich křížovém výkonovém spektru souvislý ostrý vrchol právě v místě registrace (v místě posunutí mezi obrazy). Narozdíl od toho se šum ve výkonovém spektru projeví náhodně rozložený v nesouvislých vrcholcích. Tuto metodu přesněji popíšeme následovně. Kapitola převzata z [5].

Nechť funkce  $f_1(x, y)$  a  $f_2(x, y)$  jsou definovány na  $\mathbb{R}^2$  a dále:

$$\mathfrak{F}\{f_1(x, y)\} = F_1(u, v), \quad (7.1)$$

$$\mathfrak{F}\{f_2(x, y)\} = F_2(u, v), \quad (7.2)$$

kde  $\mathfrak{F}$  označuje Fourierovu transformaci. Dále předpokládejme vzájemný posuv funkcí:

$$f_2(x, y) = f_1(x + \Delta x, y + \Delta y). \quad (7.3)$$

Na základě Fourierova posuvného teorému dostaneme vztah ve frekvenční oblasti:

$$F_2(u, v) = F_1(u, v) \exp j(u\Delta x + v\Delta y). \quad (7.4)$$

Nyní rovnici násobíme maticí  $\overline{F_1(u, v)}$ , což je matice, kde její prvky jsou komplexně sdružená čísla:

$$F_2(u, v) \overline{F_1(u, v)} = \exp j(u\Delta x + v\Delta y). \quad (7.5)$$

Tento obraz se nazývá křížové spektrum (cross spectrum) mezi  $F_1(u, v)$  a  $F_2(u, v)$ .

Dále získáme výsledné křížové výkonné spektrum, což znamená v anglickém překladu cross-phase spectrum (normalizované křížové spektrum):

$$\frac{F_2(u, v) \overline{F_1(u, v)}}{|F_2(u, v) \overline{F_1(u, v)}|} = \exp j(u\Delta x + v\Delta y), \quad (7.6)$$

V tento okamžik je již snadné odvodit relativní posuv  $\Delta x$  a  $\Delta y$ . Po provedení Fourierovy transformace získáme Dirakovu delta funkci se středem v  $(\Delta x, \Delta y)$ :

$$\mathfrak{F}^{-1} \left( \frac{F_2(u, v) \overline{F_1(u, v)}}{|F_2(u, v) \overline{F_1(u, v)}|} \right) = \exp j(u\Delta x + v\Delta y) = \mathfrak{F}^{-1}(\exp j(u\Delta x + v\Delta y)) = \delta(\Delta x, \Delta y). \quad (7.7)$$

Naznačený postup předpokládá reálné funkce s neomezeným definičním oborem hodnot. V praxi je třeba tento postup aplikovat na diskretní obrazové funkce konečné velikosti. Řešením vzniklého problému je použití diskretní verze Fourierovy transformace s předpokladem periodického rozšíření obrazových funkcí. Dirakova delta funkce je pak nahrazena jednotkovým impulsem. Je dokázáno, že i za těchto předpokladů výsledky stále platí. Výsledný vzájemný posuv obrázků se nyní jednoduše zjistí prohledáním křížového

## 7.2. APLIKACE FÁZOVÉ KORELACE

výkonového spektra v prostorové oblasti a nalezením maxima, které odpovídá výrazu  $\delta(\Delta x, \Delta y)$  z rovnice (7.7).

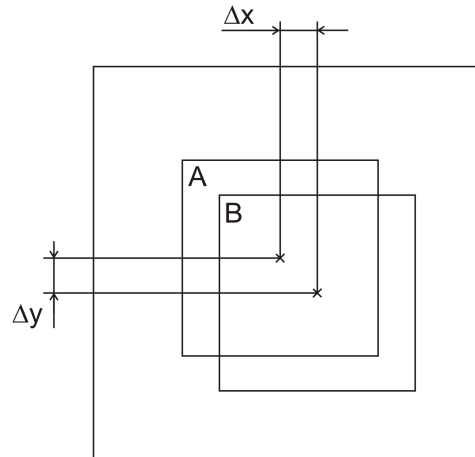
### Křížová korelace

Pokud aplikujeme zpětnou Fourierovu transformaci na obraz křížového spektra dostaneme křížovou korelaci

$$\mathfrak{F}^{-1} \left( F_2(u, v) \overline{F_1(u, v)} \right). \quad (7.8)$$

## 7.2. Aplikace fázové korelace

Cílem aplikace fázové korelace je získat velikost posuvu mezi dvěma obrázky. Abychom zajistili, že budou mít obrázky stejné hodnoty pixelu v části, kde se překrývají, vybrali jsme fotografii a vytvořili z ní dva rozměrově stejné obrazy, ovšem posunuté. A to pomocí freewarového bitmapového programu GIMP. Zapamatovali jsme si, o kolik jsme obrázky posunuli a fázovou korelaci jsme ověřili stejné hodnoty posunutí.



Obrázek 7.1: Vytvoření vzájemně posunutých obrazů z fotografie

V našem případě jsme vytvořili obrázky, které jsou posunuty o  $\Delta y = 60 px$  a  $\Delta x = 50 px$  ( $px$  je značka pixel). Dále je třeba dle vzorce pro fázovou korelaci (7.7) provést u každého z obrazů dvojrozměrnou diskretní Fourierovu transformaci. Potom násobit prvky spektra první matice s komplexně sdruženými hodnotami spektra druhé matice a výsledek dělit velikostí komplexního čísla, vzniklého násobením spekter.

Pak hledáme ve vzniklém obraze maximum, to znamená najít největší hodnotu v matici obrazu. A souřadnice toho maxima od středu matice udávají posunutí v ose  $y$  ve svislém směru a v ose  $x$  ve vodorovném směru. V našem případě bylo nalezeno maximum  $50 px$  od středu ve vodorovném směru a  $60 px$  od středu ve svislém směru. Nalezli jsme stejné hodnoty, o jaké jsme obrazy skutečně posunuli. Tím jsme ověřili, že v námi vytvořeném programu fázová korelace funguje správně.

Křížová korelace dle vzorce (7.8) je vidět na obrázku (7.5).

Dále jsme vyzkoušeli modifikovanou fázovou korelaci dle vzorce (7.9), kterou budeme později potřebovat.

$$\mathfrak{F}^{-1} \left( \frac{F_2(u, v) \overline{F_1(u, v)}}{(|F_2(u, v)| + n_1) (|\overline{F_1(u, v)}| + n_2)} \right), \quad (7.9)$$

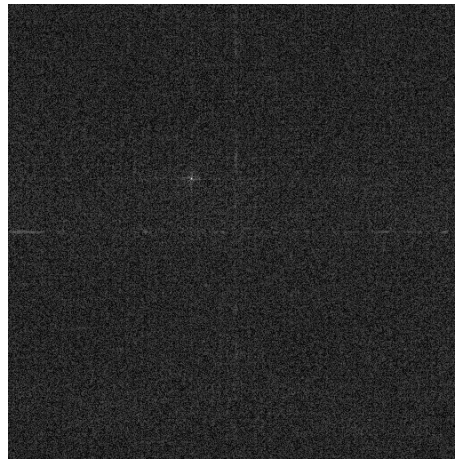


(a) Obraz A

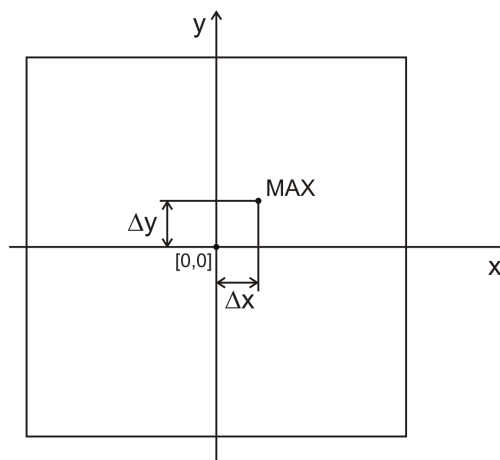


(b) Obraz B

Obrázek 7.2: Posunuté obrazy



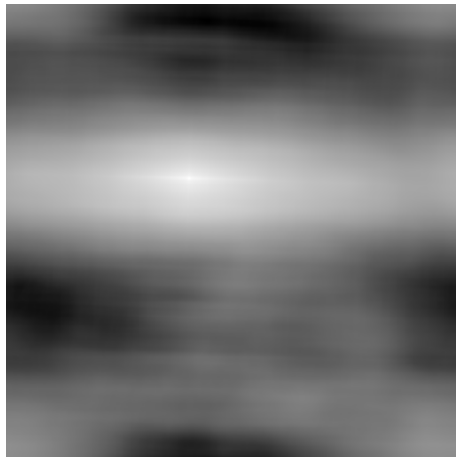
Obrázek 7.3: Obraz po fázové korelaci



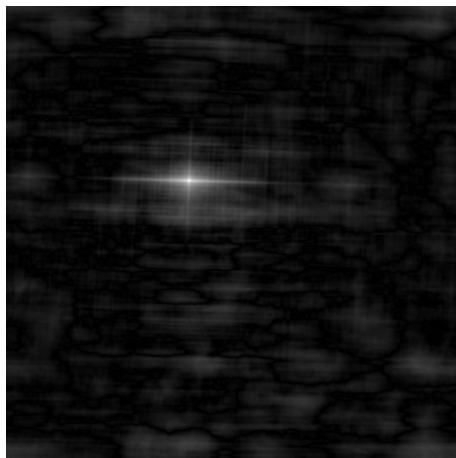
Obrázek 7.4: Interpretace výsledku po fázové korelaci

kde  $n_1$  a  $n_2$  jsou váhy. Když se  $n_1, n_2$  blíží  $\infty$ , pak se tato modifikovaná korelace stává křížovou. Pokud se naopak  $n_1, n_2$  přiblíží nulovým hodnotám má, tato korelace vlastnosti fázové korelace.

## 7.2. APLIKACE FÁZOVÉ KORELACE



Obrázek 7.5: Obraz po křížové korelaci



Obrázek 7.6: Obraz po modifikované fázové korelaci pro zvolené hodnoty  $n_1 = 10000$ ,  $n_2 = 10000$

I u těchto typů korelace vyšlo hledané maximum v obraze stejně jako v obraze fázové korelace. Pokud bychom za vstupní obrazy vzali identické obrazy, pak by je fázová korelace sesadila na sebe. Dostali bychom tím nulový posun a maximální hodnota (nesvětlejší místo v obraze) by byla přesně uprostřed obrazu.

## 8. Měření rychlosti objektu za využití Fourierovy transformace

Rychlost je charakteristika pohybu, která nám sděluje, jakým způsobem se mění poloha tělesa v čase. Měření rychlosti tak probíhá nepřímou, tedy z jiných veličin, které dokážeme měřit. Jakými různými metodami měřit rychlost, lze vidět i na našich vozovkách, kde se jich využívá ke kontrole povolené rychlosti vozidel (viz. [9]).

### 8.1. Metody měření rychlosti

#### RADAR

Prvním takovým měřicím systémem je RADAR (RADio Detection And Ranging), český výraz je rádiová detekce a měření vzdálenosti, což je v případě měření rychlosti poněkud zavádějící. Metoda využívá Dopplerova jevu, tedy změny kmitočtu signálu přijímaného od pohybujícího se objektu vůči kmitočtu signálu původního. Signál akustický nebo světelný, se v homogenním prostředí pohybuje konstantní rychlostí, tedy nezrychluje, ani nezpomaluje. Signál světelný se pohybuje rychlostí světla, signál akustický rychlostí zvuku. Pokud se objekt a pozorovatel vůči sobě nepohybují, respektive jejich vzájemná vzdálenost se nemění, musí signál mezi oběma body urazit vždy stejnou vzdálenost. Cesta od objektu k pozorovateli bude trvat vždy stále stejnou dobu a dorazený signál bude přesnou kopií signálu vyslaného. Pokud se objekt bude přibližovat a měříme-li ho z pevného stanoviště, stačí nám zjistit absolutní hodnotu rozdílu kmitočtů mezi signálem přijatým a vysílaným, a přes známou hodnotu rychlosti světla se tak velmi snadno dostaneme k rychlosti měřeného vozidla. Výhodou této metody je principiální jednoduchost a s tím spojené nízké finanční náklady na výrobu. Další výhodou je měření rychlosti bez ohledu na povětrnostní vlivy a taktéž měření na značnou vzdálenost. Nevýhodou je poměrně široký kužel paprsku, který tak dává možnost detekce vhodným přijímačem a včasného varování. Další nevýhodou je použitelnost pouze v případě, kdy měřící stanoviště je v ose pohybu vozidla, jinak prudce klesá přesnost měření.

#### LIDAR

LIDAR (LASer Detection And Ranging) je další metodou měření ujeté vzdálenosti za jednotku času. Metoda využívá měření pomocí světelného paprsku a to nejčastěji polovodičového laseru v infračerveném pásmu. V intervalu  $67\text{ ms}$  je vyslána série pulzů. Pokud je přijat jejich odraz od objektu, je změřena prodleva příjmu od odeslání, a tím je zjištěna série hodnot udávajících vzdálenost měřeného objektu od měřidla. Z rozdílu jednotlivých změřených vzdáleností a ze známého intervalu mezi měřícími pulzy je vypočtena rychlost pohybu objektu. Výhodou je minimální rozptyl paprsku. Díky tomu je možno změřit rychlost zvoleného jednotlivého vozidla mezi ostatními vozidly. Opět nezávisí na povětrnostních podmínkách. Mezi další výhody patří poměrně velký dosah, uvádí se až dva kilometry. Nevýhodou je opět závislost přesnosti měření na úhlu svíraném osou paprsku s osou pohybu.

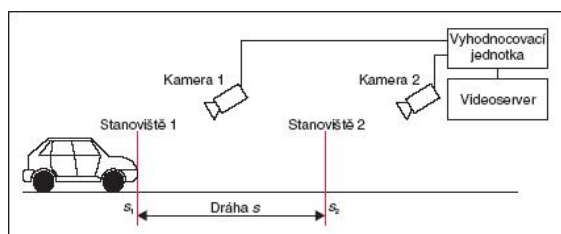
Jelikož právní systém ČR vyžaduje subjektivní provinění, tedy jednoznačnou identifikaci přestupníka, je třeba vozidlo zastavit při zachování přímé viditelnosti zastavujícím

## 8.2. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ ZMĚNY ÚHLOVÝCH ROZMĚRŮ

policistou na celém úseku mezi měřícím stanovištěm a místem zastavení. Nebo je třeba vozidlo vyfotit při měření rychlosti.

### Úsekové měření rychlosti

Díky poklesu nákladu na výpočetní výkon zažívají rozmach kamerové systémy ve spojení s technologií rozpoznávání obrazu. Jednou z metod, která se již používá, je právě úsekové měření rychlosti. To funguje tak, že kamery zaznamenají každé vozidlo projíždějící přes kontrastní značku na vozovce a získaný obraz předávají počítači. Ten každý snímek opatří časovou značkou a uloží. V uloženém obraze je pak softwarově vyhledána a přečtena registrační značka vozidla a uložena spolu s obrazem jako datová informace. Systém tedy ví v jakou dobu vozidlo projelo kolem každé kamery, a jelikož jsou v něm zaneseny i polohy kamer, a tedy i jejich vzájemné vzdálenosti, je počítač schopen jednoduše vypočítat průměrné rychlosti všech zaznamenaných vozidel na jednotlivých úsecích. Výhodou je, že je to bezobslužný systém, který nelze detekovat antiradary. Nevýhodou však je, že u těchto systémů se nastavuje dost vysoká tolerance, typicky 10 až 20 *km/h*. Další nevýhodou je snadná ovlivnitelnost řidičem. Pokud si řidič uvědomí, že vjel do měřeného úseku příliš rychle, stačí po zbytek cesty k následujícím kamerám projet sníženou rychlostí.



Obrázek 8.1: Schéma úsekového měření rychlosti

### Měření rychlosti pomocí Fourierovy transformace

je další alternativou, jak by se dala měřit rychlost pomocí nástroje obrazové analýzy. K tomuto systému, který v následné kapitole podrobněji popíšeme, bychom mohli jako výhody oproti předcházejícím systémům považovat menší variabilitu přesnosti měření na úhlu svíraném osou paprsku s osou pohybu objektu a nemožnost ovlivnění antiradary. Výhoda oproti úsekovému měření je, že by k měření stačilo pouze jedno snímácí zařízení (kamera), tedy jednodušší a finančně méně náročnější měřicí systém. Další výhodou oproti předešlým systémům je, že nemusíme měřit rychlost v reálném čase. Mohli bychom objekt nasnímat pomocí kamery nebo fotoaparátu a později si nechat rychlost vypočítat.

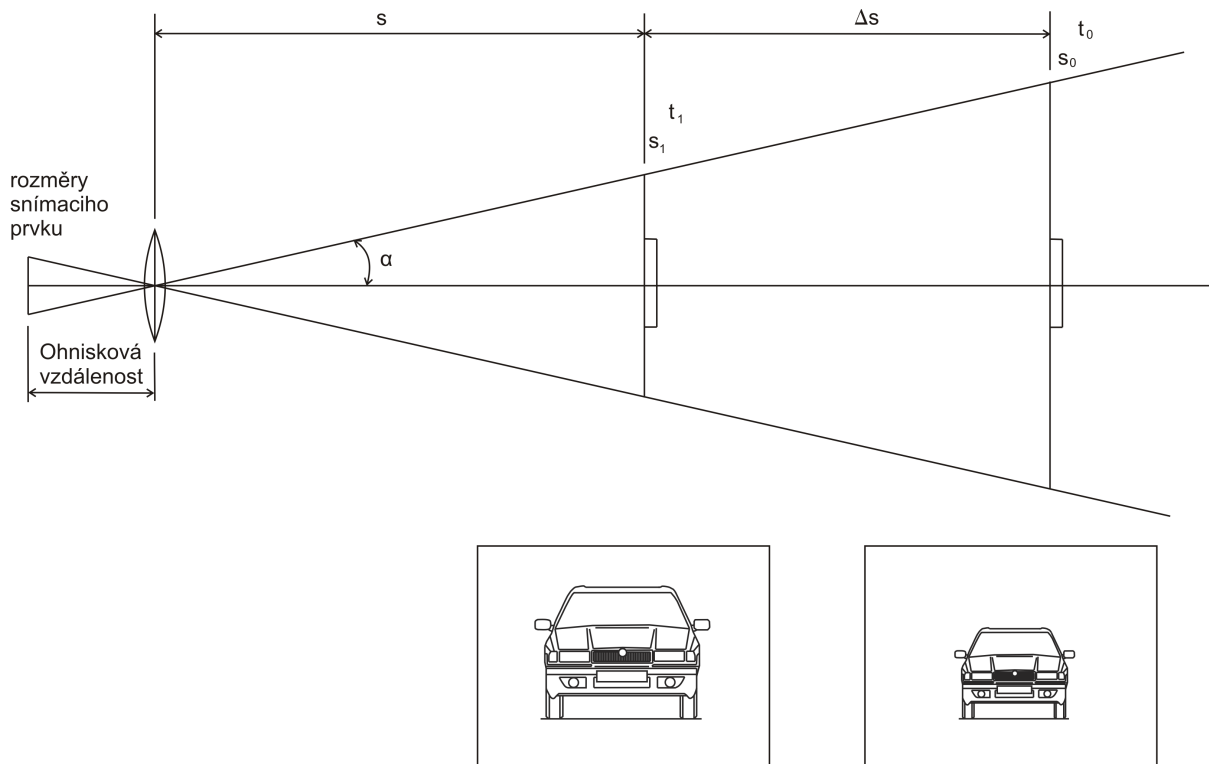
## 8.2. Metoda měření rychlosti za využití změny úhlových rozměrů

K výpočtu rychlosti objektu potřebujeme ze známého vzorečku délku uraženého úseku a čas, za který daný úsek urazil. Ze známých parametrů optiky snímácího zařízení a jedné reálné míry objektu lze ze dvou nebo více snímků zjistit délku, kterou objekt urazil. Dále budeme znát v jakých intervalech snímácí zařízení pořizuje snímky, respektive budeme znát dobu za jakou je pořízen druhý snímek od prvního. Tím máme dostatek hodnot k výpočtu rychlosti. Známými parametry je myšlena ohnisková vzdálenost a rozměry



## 8. MĚŘENÍ RYCHLOSTI OBJEKTU ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

snímacího prvku (například CCD-čipu). Za reálnou míru vezmeme délku značky, protože ji lze považovat za prvek na objektu, který má vždy stejný tvar a rozměr, alespoň pro jednotlivé typy automobilů (tj. osobní, nákladní, jednostopé).



Obrázek 8.2: Schéma zdánlivé změny velikosti objektu

Vzdálenost objektu vypočítáme tak, že pro každý snímek zjistíme vzdálenost objektu od snímacího zařízení a z rozdílu těchto vzdáleností získáme délku uraženého úseku. Vzdálenost snímku od snímacího zařízení zjistíme tak, že vypočítáme poměr mezi rozměrem fotografie a naměřenou délkou značky v obraze. Tento poměr musí být stejný i mezi rozměrem snímané roviny v zorném poli a skutečnou délkou značky. Tím snadno vypočítáme rozměr snímané roviny v metrech. A vzdálenost nyní dostaneme z pomyslného pravoúhlého trojúhelníku za pomoci funkce tangens:

$$h_s = \frac{h_f \cdot d_s}{d_f} \quad \left[ m = \frac{px \cdot m}{px} \right],$$

kde  $h_s$  je rozměr snímané roviny,  $h_f$  je rozměr fotografie,  $d_s$  je skutečná délka značky,  $d_f$  je délka značky v obraze.

Vzdálenost pak dostaneme

$$s = \frac{h_s/2}{\tan(\alpha)},$$

kde  $s$  je vzdálenost objektu,  $h_s$  je rozměr snímané roviny,  $\alpha$  je poloviční zorný úhel.

$$\alpha = \arctan\left(\frac{r/2}{o}\right),$$

kde  $r$  je rozměr snímacího čipu,  $o$  je ohnisková vzdálenost.

### 8.3. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

Pro tento výpočet však musíme naměřit délku značky v obraze. Nevýhodou je velká nepřesnost v měření rozměru značky u vzdálenějšího objektu. Čím dále objekt je, tím méně pixelů poskytuje informaci o délce značky a tím méně přesné měření je. Toto pak ovlivňuje i výpočet rychlosti. Tento problém lze vyřešit tak, že vzdálenost druhého objektu zjistíme metodou určení změny velikosti objektu za využití Fourierovy transformace. Využijeme tak několikanásobný počet pixelů, protože budeme pracovat z celou plochou objektu a ne jen s délkovým rozměrem značky.

Zde je nutno poznamenat, že parametry snímacího systému nelze přesně zjistit, a proto je nutná kalibrace snímacího zařízení. A to takovým způsobem, že objekt vyfotíme ve známé vzdálenosti a v této pořízené fotografii (tzv. referenčním snímku) změříme rozměr poznávací značky. Když pak potřebujeme znát vzdálenost objektu v libovolném obraze, stačí změřit rozměr poznávací značky a přes jednoduchý výpočet zjistit vzdálenost objektu od snímacího zařízení.

$$s = \frac{s_r \cdot d_r}{d}, \quad (8.1)$$

kde  $s$  je vzdálenost objektu od snímacího zařízení,  $s_r$  je vzdálenost objektu od snímacího zařízení při pořizování referenčního snímku,  $d_r$  je naměřený rozměr poznávací značky v referenčním snímku,  $d$  je rozměr poznávací značky ve snímku objektu, u kterého chceme znát vzdálenost od snímacího zařízení.

## 8.3. Metoda měření rychlosti za využití Fourierovy transformace

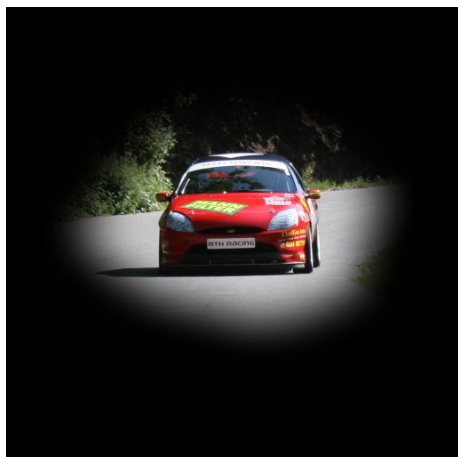
Tato metoda měření rychlosti se dá využít na jakýkoliv pohybující se objekt a my jsme si vybrali pohybující se vozidlo, protože lze snadno ověřit náš výpočet a protože u všech těchto objektů je stejný rozměr poznávací značky. Naše situace je nyní taková, že dokážeme relativně přesně zjistit vzdálenost objektu na prvním snímku, který je blíže snímacímu zařízení. A chceme dostat hodnotu zvětšení vzdálenějšího objektu od bližšího. K tomu nám pomůže již dříve zmiňovaná fázová korelace (kapitola 7), neboť pro Fourierovy obrazy v logaritmicko-polárních souřadnicích platí, že vzájemné posunutí v jedné ose znamená změnu velikosti a v druhé rotaci objektu. Nás bude pouze zajímat změna velikosti, neboť vzájemnou rotaci nepředpokládáme. Teď budeme popisovat postup jak se dostat k hodnotě zvětšení objektu.

### 8.3.1. Načtení fotografií

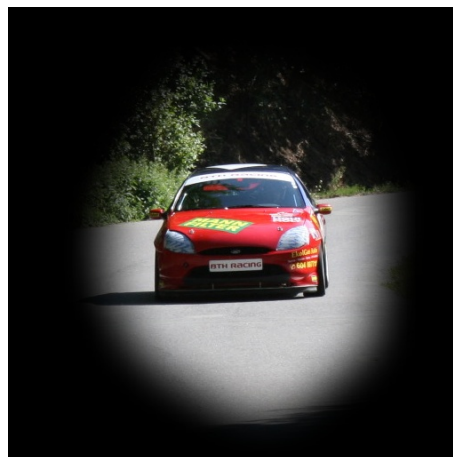
Během výpočtu zvětšení objektu provádíme na obraze více Fourierových transformací. Aby byl čas výpočtu malý, musí být obrazy čtvercové a menších rozměrů. V celém výpočtu tedy používáme fotografie o rozměrech  $512 \times 512$ . Před zahájením výpočtu je třeba okolí objektu v obrázku upravit. Úpravu provedeme tak, že přiřadíme nulové hodnoty pixelům v okolí objektu. Je důležité, aby toto okolí mělo u každého z načtených obrázků jiný tvar. Dále je třeba zajistit plynulý přechod mezi hodnotami obrazu a okolím.

Barevný obraz je popsán barvami modelu RGB, hodnoty jsou kódované ve 24 bitové hloubce (kapitola 2.1.2). Pro naše výpočty potřebujeme černobílé fotografie. Z barevné fotografie je snadno možné vyrobít černobílou fotografii tak, že převedeme všechny barvy

## 8. MĚŘENÍ RYCHLOSTI OBJEKTU ZA VYUŽITÍ FOURIEROVY TRANSFORMACE



(a) 1.Obraz



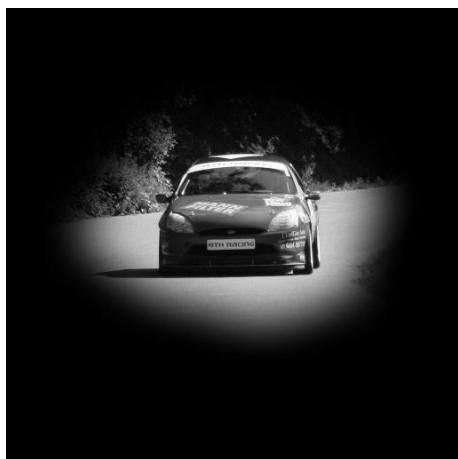
(b) 2.Obraz

Obrázek 8.3: Načtené fotografie

na stupně šedé, přičemž šedé (barevně neutrální) jsou ty barvy, které mají všechny tři RGB složky stejné. Tím převedeme RGB ze 24 bitové hloubky na plnohodnotný RGB využívající 8 bitové hloubky.

$$\text{Jas (nová hodnota R, G, B)} = \frac{(R + G + B)}{3},$$

kde R je hodnota intenzity červené barvy, G hodnota intenzity zelené barvy a B intenzita modré barvy.



(a) 1.Obraz



(b) 2.Obraz

Obrázek 8.4: Černobílé fotografie

V našem programu, který je psán v jazyce C++ je pro převod napsána funkce s názvem `TransferBlackAWhite`. Ta načte hodnoty z vloženého obrazu, vypočítá stupně šedi a uloží hodnoty do pole `pArray`.

## 8.3. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

```
void Matrix::TransferBlackAWhite(wxImage image)
{ /*Prevede data z image na cernobilou fotografii a ulozi do pole pointeru formatu Ipp32fc */
  pArray = new Ipp32fc [image.GetHeight() * image.GetWidth()];
  for (int i=0;i<image.GetHeight();i++)
  {
    for (int j=0;j<image.GetWidth();j++)
    {
      float r = (image.GetBlue(j,i)+image.GetGreen(j,i)+image.GetRed(j,i))/3;
      pArray[image.GetWidth()*i+j].re=r;
      pArray[image.GetWidth()*i+j].im=0;
    }
  }
  return ;
}
```

### 8.3.2. Fourierova transformace

U každého takto upraveného a načteného obrazu provedeme Fourierovu transformaci. Abychom operaci nemuseli programovat využili jsme knihovny: Intel® Integrated Performance Primitives for Intel® Architecture Volume 2: Image and Video Processing. Z knihovny jsme využili funkci pro Fourierovu transformaci.

Funkce FFTFwd tedy provede na daném poli pArray Fourierovu transformaci a využije těchto knihovnických funkcí:

```
ippiFFTInitAlloc_C_32fc , ippiFFTFwd_CToC_32fc_C1IR, ippiFFTFree_C_32fc
```

Takto vypadá funkce FFTFwd:

```
IppStatus Matrix::FFTFwd()
{//dopredna f transformace,ulozena na pole pointeru pArray
  IppiFFTSpec_C_32fc *spec;
  IppStatus status;
  int powerH = int(log(size.GetHeight())/log(2));
  int powerW = int(log(size.GetWidth())/log(2));

  ippiFFTInitAlloc_C_32fc ( &spec , powerW , powerH , IPP_FFT_DIV_INV_BY_N , ippAlgHintAccurate );
  status=ippiFFTFwd_CToC_32fc_C1IR ( pArray , size.GetWidth()*sizeof(Ipp32fc) , spec , 0 );
  ippiFFTFree_C_32fc ( spec );

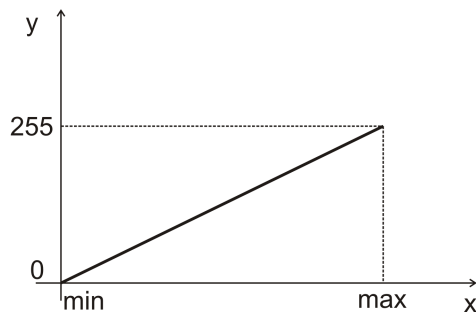
  return status;
}
```

Fourierova transformace převede matici na matici komplexních čísel o stejných rozměrech. Dále však pracujeme s amplitudovým spektrem (vzorec 6.5).

```
void Matrix::TransferSpectrumWithoutNewArray(Ipp32fc *pointer)
{
  int Count = size.GetHeight() * size.GetWidth();
  for (int i=0;i<Count;i++)
  {
    pointer[i].re = sqrt( pow(pointer[i].re,2) + pow(pointer[i].im,2) );
    pointer[i].im = 0;
  }
  return;
}
```

Abychom mohli amplitudové spektrum zobrazit na obrazovku, musíme hodnoty převést na zobrazitelné hodnoty. To lze provést pomocí lineární transformace (kapitola 3.2). Hodnoty amplitudového spektra jsou kladná reálná čísla a my je potřebujeme převést na zobrazitelný interval  $\langle 0, 255 \rangle$ , vše je vidět na obrázku (8.5).

## 8. MĚŘENÍ RYCHLOSTI OBJEKTU ZA VYUŽITÍ FOURIEROVY TRANSFORMACE



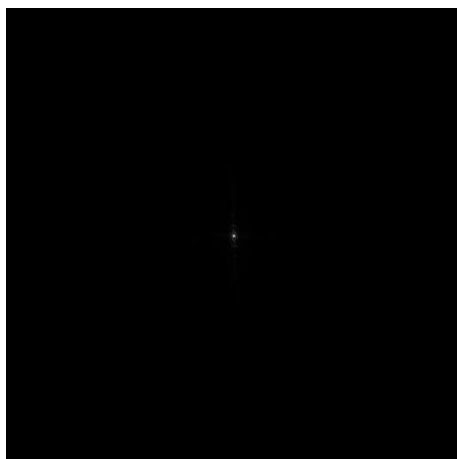
Obrázek 8.5: Lineární transformace

Spektrum transformujeme pomocí vzorečku (8.2)

$$y = \frac{255}{max} \cdot x, \quad (8.2)$$

kde  $x$  jsou hodnoty původního spektra a  $y$  transformovaného spektra. K výpočtu však potřebujeme najít maximum ze všech hodnot amplitudového spektra. K tomu nám v programu slouží funkce:

```
void Matrix::FindMaxMin(Ipp32fc *pointer)
{
    maximum=pointer[0].re;
    minimum=pointer[0].re;
    int h = size.GetHeight();
    int w = size.GetWidth();
    for(int i=0;i<h;i++)
        for(int j=0;j<w;j++)
            {
                if ( maximum < pointer[w*i+j].re )
                {
                    maximum = pointer[w*i+j].re;
                    xmax = j;
                    ymax = i;
                }
                if ( minimum > pointer[w*i+j].re )
                    minimum = pointer[w*i+j].re;
            }
    return;
}
```



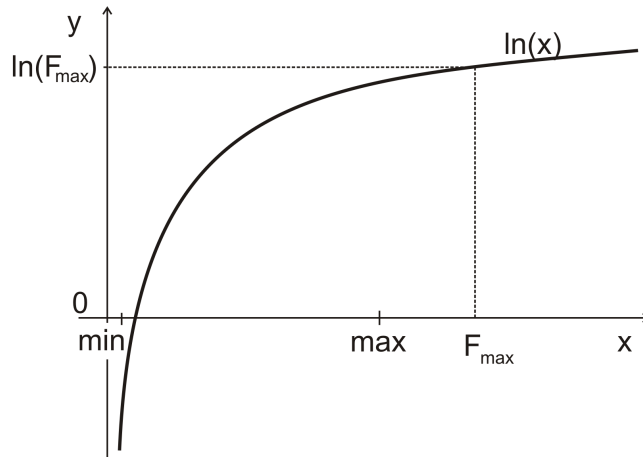
Obrázek 8.6: Amplitudové spektrum po lineární transformaci

### 8.3. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

K vykreslení obrázku na plochu používáme funkci `PutImage`, která z reálných čísel udělá celá čísla a nastaví ke všem složkám barev stejnou hodnotu.

```
wxImage Matrix::PutImage(Ipp32fc *pointer)
{
    wxImage imageOut;
    imageOut.Create(size.GetWidth(),size.GetHeight());
    for (int i=0;i<size.GetHeight();i++)
    {
        for (int j=0;j< size.GetWidth();j++)
        {
            int color = (int)pointer[size.GetWidth()*i+j].re;
            imageOut.SetRGB(j,i,color, color, color);
        }
    }
    return imageOut;
}
```

Na obrázku (8.6) je již zobrazené amplitudové spektrum po lineární transformaci. Z vykresleného spektra je vidět, že jen okolo středu obrázku jsou velké hodnoty a ve zbytku obrazu jsou malé hodnoty, které se projevily černou barvou. Abychom mohli lépe s informacemi spektra pracovat, potřebujeme malé hodnoty zesílit. Toho docílíme použitím nelineární transformace hodnot.



Obrázek 8.7: Logaritmičká transformace hodnot

Logaritmičká transformace:

$$y = c \cdot \ln \left( \frac{v - \min}{\max - \min} \cdot (F_{\max} - 1) + 1 \right), \quad (8.3)$$

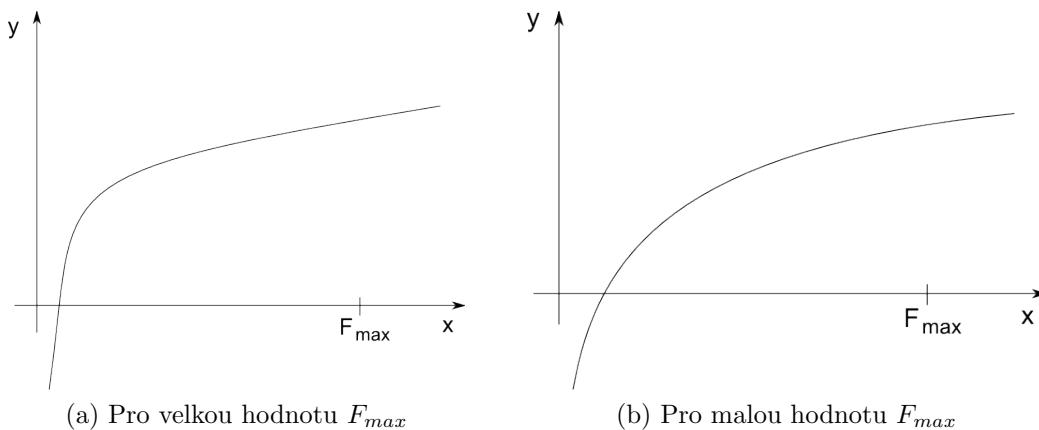
kde  $v \in \langle \min, \max \rangle$ . Interval  $\langle \min, \max \rangle$  roztahujeme na interval  $\langle 0, F_{\max} \rangle$ . Hodnotou  $F_{\max}$  regulujeme jakou část logaritmu využijeme. Jaký vliv má velikost  $F_{\max}$  na transformaci hodnot, je vidět na obrázku (8.8). A jelikož se nám interval ztransformuje na interval  $\langle 0, \ln(F_{\max}) \rangle$ , je nutno tento interval ještě lineárně transformovat na zobrazitelný interval  $\langle 0, 255 \rangle$  tak, že  $c$  vypočítáme:

$$c = \frac{255}{\ln(F_{\max})}.$$

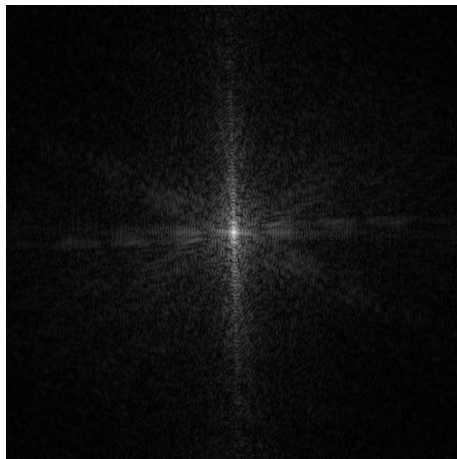
Na obrázku (8.9) je vidět amplitudové spektrum po logaritmičké transformaci hodnot. V programu nám k této transformaci slouží funkce `TransformLogCoordinate`

## 8. MĚŘENÍ RYCHLOSTI OBJEKTU ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

```
void Matrix::TransformLogCoordinate (Ipp32fc *pointer,int F)
{
FindMaxMin(pointer);
int Count = size.GetHeight() * size.GetWidth();
if (minimum>=0)
{
for (int i=0;i<Count;i++)
{
pointer[i].re = (255 * log( (((pointer[i].re-minimum)*(F-1))/(maximum-minimum)) +1 ) )/log(F);
}
}
return;
}
```



Obrázek 8.8: Využitá část logaritmu

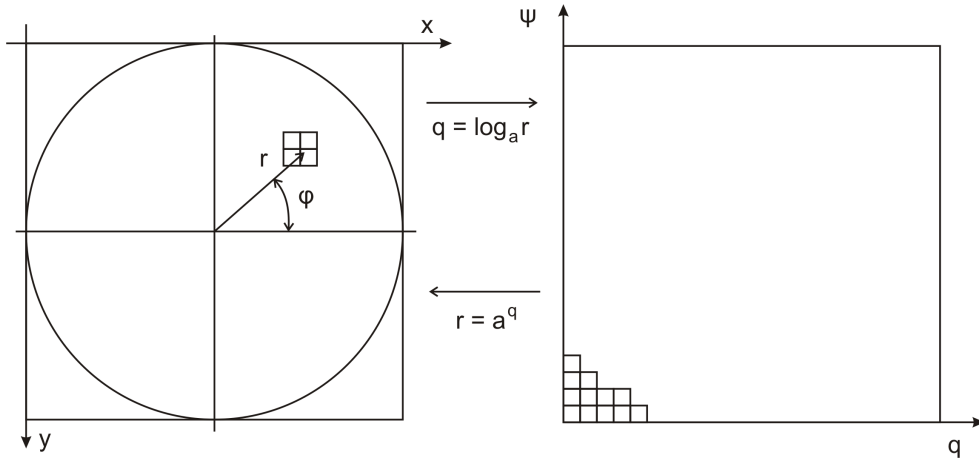


Obrázek 8.9: Amplitudové spektrum po logaritmické transformaci hodnot

### 8.3.3. Logaritmicko polární souřadnice

K získání hodnoty zvětšení objektu v obraze se musí do fázové korelace dosadit obrazy amplitudového spektra v logaritmicko-polárních souřadnicích. K převodu souřadnic použijeme logaritmicko-polární transformaci (Log-pol transformaci). Transformaci uvedeme pro náš konkrétní případ, tedy obraz o rozměrech  $512 \times 512$ .

### 8.3. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ FOURIEROVY TRANSFORMACE



Obrázek 8.10: Log-polární transformace

Na obrázku (8.10) vpravo je nová obrazová matice, která má stejné rozměry, ale v log-polárních souřadnicích. Ve svislém směru je úhel  $\psi$ , který bereme z intervalu  $\langle 0, 511 \rangle$ . Ve vodorovném směru je poloměr  $q$ , který bereme z intervalu  $\langle 0, 511 \rangle$ . Technicky se transformace provádí tak, že se postupně berou souřadnice prvků z nově vytvářené obrazové matice, ty se převedou pomocí vztahů (8.4) na úhel  $\varphi$  a poloměr  $r$ . S těmito parametry jdeme pro hodnoty bodu do původní obrazové matice o souřadnicích  $x, y$ . Úhel  $\varphi$  nabývá hodnot pouze  $\langle 0, \pi \rangle$ , protože pole je symetrické. Poloměr  $r$  nabývá hodnot z intervalu  $\langle 1, 256 \rangle$ , protože počátek souřadného systému je ve středu obrazu.

Převodní vztahy jsou:

$$\begin{aligned}\varphi &= \frac{\psi}{511} \cdot \pi, \\ r &= a^q,\end{aligned}\tag{8.4}$$

kde  $a$  je vhodný základ logaritmu.

Vhodný základ logaritmu se vypočítá z podmínek:

- $\log_a 256 = 511$
- $\log_a 1 = 0$

Výpočet základu plyne z první rovnice, protože druhá podmínka je splněna pro všechna  $a > 1$ . K výpočtu vhodného základu využijeme vlastnosti logaritmu:

$$\log_a x = \frac{\log_b x}{\log_b a}.$$

Základ pak dostaneme výpočtem:

$$a = \exp\left(\frac{\ln 256}{511}\right).$$

Po převedení poloměru a úhlu si bereme hodnoty ležící na souřadnicích vypočtených jako průměty poloměru do os  $x, y$ :

$$r_x = r \cos \varphi\tag{8.5}$$

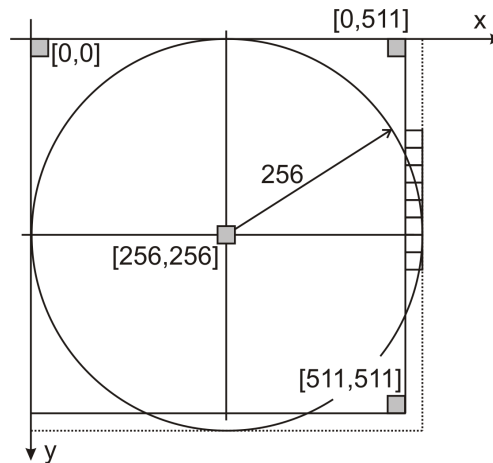
$$r_y = r \sin \varphi\tag{8.6}$$



## 8. MĚŘENÍ RYCHLOSTI OBJEKTU ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

Výpočtem průmětů poloměrů však získáváme nové hodnoty obrazových bodů mimo souřadnice původního obrazového rastru. K výpočtu hodnoty mezilehlého obrazového bodu použijeme bilineární interpolaci (kapitola 4.2).

Obrazová matice, z které bereme hodnoty, se také musí doplnit o jeden sloupec hodnot na pravém okraji, protože obraz má rozměr sudého čísla. Vezmeme-li střed v bodě  $[256, 256]$ , pak se poloměrem  $r$  dostaneme na pravém okraji mimo obrazové hodnoty. Z předpokladu Fourierovy transformace víme, že obrazové pole po Fourierově transformaci je periodické, proto můžeme doplňovaný sloupec vyplnit hodnotami z prvního sloupce obrazové matice (obr. 8.11).



Obrázek 8.11: Obrazová matice s doplněným sloupcem

V programu na logaritmickou transformaci máme funkci `LogPolTransform`, která přijímá jako parametr pomocné pole kam si dočasně uloží původní hodnoty, z kterých následně bere hodnoty pro výpočty.

```
void Matrix::LogPolTransform(Matrix *pointerHelp)
{
    for (int i=0; i<size.GetHeight(); i++)
    {
        for (int j=0; j<size.GetWidth(); j++)
        {
            pointerHelp->SetpArray(j, i, pArray[size.GetWidth()*i+j]);
        }
    }

    double fi, ro, radiusNew, radiusOld, radiusOldx, radiusOldy;
    float h1, h2, h3, h4;
    Ipp32fc value;
    int h = size.GetHeight();
    int w = size.GetWidth();
    double base;
    base = exp(log(256)/511);
    for (int i=0; i<h; i++)
    {
        for (int j=0; j<w; j++)
        {
            radiusNew = j;
            ro = i;
            fi = ro/511*PI;
            radiusOld = pow(base, radiusNew);

            radiusOldx = fmod ( (-radiusOld * cos(fi) + 256), 511);
            radiusOldy = (-radiusOld * sin(fi) + 256);

            int hrx = (int) floor(radiusOldx);
```

### 8.3. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

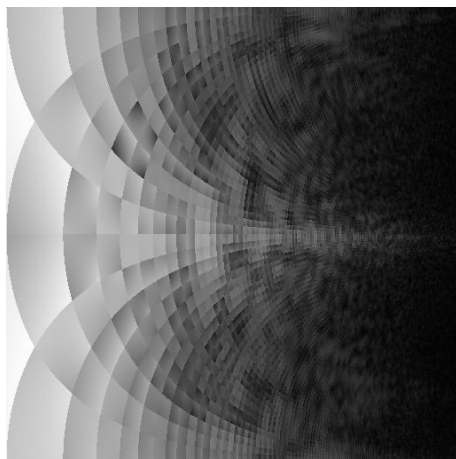
```

int hry = (int) floor(radiusOldy);

float restx = radiusOldx - hrx;
float resty = radiusOldy - hry;
h1 = pointerHelp->GetpArray(hrx,hry).re;
h2 = pointerHelp->GetpArray(hrx,hry+1).re; //nasledujici pixel ve smeru y
h3 = pointerHelp->GetpArray(hrx+1,hry).re; //nasledujici ve smeru x
h4 = pointerHelp->GetpArray(hrx+1,hry+1).re; //nasledujici ve smeru x i y
//h1 h2 h3 h4 musi byt presne takto oznaceny kvuli bilineartransform
value.im = 0;
value.re = BilinearInterpolation(restx,resty,h1,h2,h3,h4);
SetpArray(j,511-i,value); // (511-i) kvuli zmene pocatku souradnych os
}
}
}

```

Na obrázku (8.12) je amplitudové spektrum zobrazeno v logaritmicko-polárních souřadnicích.

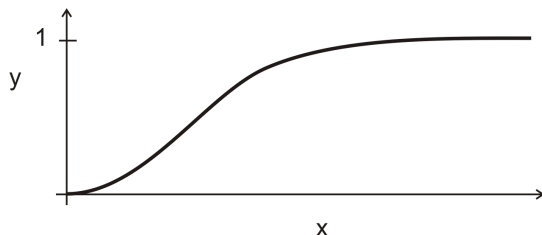


Obrázek 8.12: Obraz v log-polárních souřadnicích

Dále musíme potlačit nízké frekvence v amplitudovém spektru. Ty se nachází, jak už víme, na souřadnicích kolem středu obrazové matice. Proto vynásobíme obrazovou matici obrácenou Gaussovou dvourozměrnou funkcí (rovnice 8.7).

$$h(x, y) = 1 - \exp\left(-\frac{(x)^2 + (y)^2}{\sigma^2}\right) \quad (8.7)$$

Pokud bychom tak neudělali, fázová korelace by následně obrazy sesadila na sebe. Jelikož však máme amplitudové spektrum v logaritmicko-polárních souřadnicích, stačí vzít obrácenou jednorozměrnou Gaussovou funkci (obrázek 8.13) a násobit s ní každý řádek obrazové matice. K potlačení nízkých frekvencí máme v programu funkci GaussTransform.

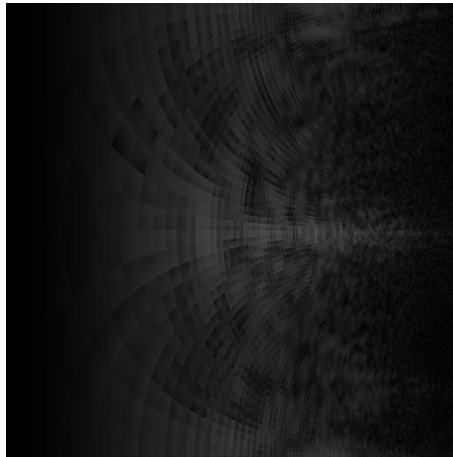


Obrázek 8.13: Jednorozměrná Gaussova funkce odečtená od 1

## 8. MĚŘENÍ RYCHLOSTI OBJEKTU ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

```
void Matrix::GaussTransform( Ipp32fc *pointer,int variability )
{//pronasobi hodnoty obrazu gausovou funkci, pouze po radcich
//predpoklada se ze je pred tim udelana logpol transformace
for (int i=0;i<size.GetHeight();i++)
{
for (int j=0;j<size.GetWidth();j++)
{
float value = 1-exp(-pow(float(j),float(2))/variability);
value =pointer[size.GetWidth()*i+j].re*value;
pointer[size.GetWidth()*i+j].re=value;
}
}
}
```

Výsledkem je obrázek (8.14). Zde je vidět ztmavení levé strany obrazu, což je okolí středu obrazu v souřadnicích  $x, y$ .



Obrázek 8.14: Amplitudové spektrum v log-polárních souřadnicích vynásobené obrácenou Gaussovou funkcí

### 8.3.4. Fázová korelace

Předcházejícím postupem je třeba upravit oba vstupní obrazy. Na obrazy nyní použijeme fázovou korelaci (kapitola 7). Aby nám výpočet neztroskotal při dělení nulou, použijeme modifikovanou fázovou korelaci, která je při vhodné volbě vah podobná fázové korelaci.

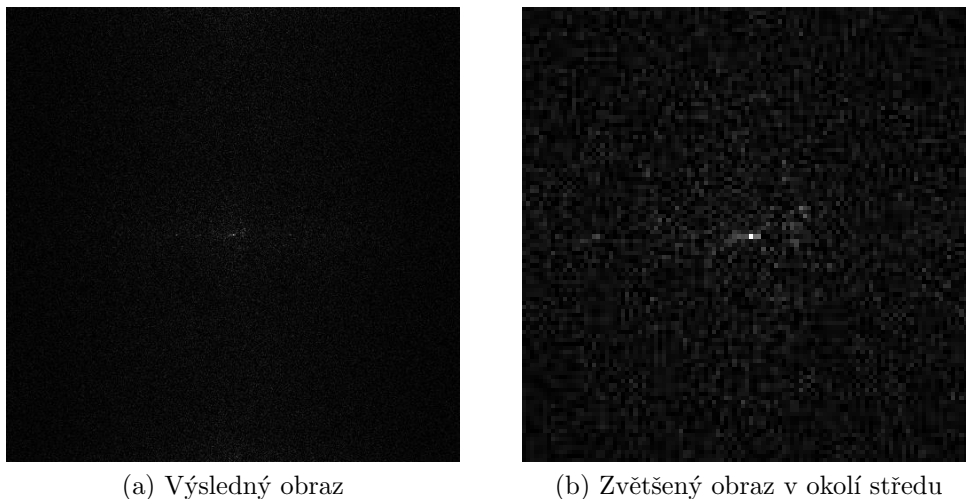
$$\mathfrak{F}^{-1} \left( \frac{F_2(u, v) \overline{F_1(u, v)}}{(|F_2(u, v)| + 1) (|\overline{F_1(u, v)}| + 1)} \right),$$

kde váhy  $n_1 = 1, n_2 = 2$ .

Po fázové korelaci je obraz zobrazen na obrázku (8.15.a). Ovšem výsledky z něj moc čitelné zatím nejsou. Maximální hodnotu, kterou bychom měli nyní hledat, bychom našli ve středu obrazu.

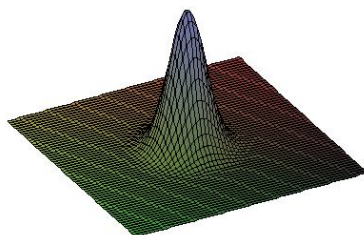
Ve zvětšeném detailu (8.15.b) obrazu po fázové korelaci je vpravo od středu vidět hlouček světlejších bodů. Dá se tedy předpokládat, že vhodným filtrem eliminujeme osamocený světlý bod uprostřed a zvýrazníme oblasti, kde je takových světlých bodů více. Zlepšení nám přinese odfiltrování vysokých frekvencí. Jelikož máme obraz před zpětnou

### 8.3. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

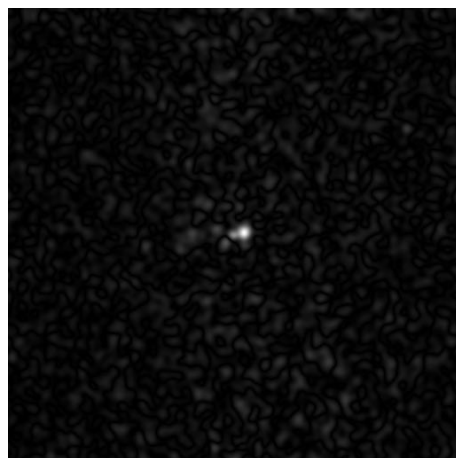


Obrázek 8.15: Obraz po fázové korelaci

Fourierovou transformací ve frekvenčním spektru, využijeme vztah (5.1) Fourierovy transformace a konvoluce a provedeme konvoluci na frekvenčním spektru. Ta se realizuje násobením obrazové matice Gaussovou dvourozměrnou funkcí.



Obrázek 8.16: Gaussova dvourozměrná funkce

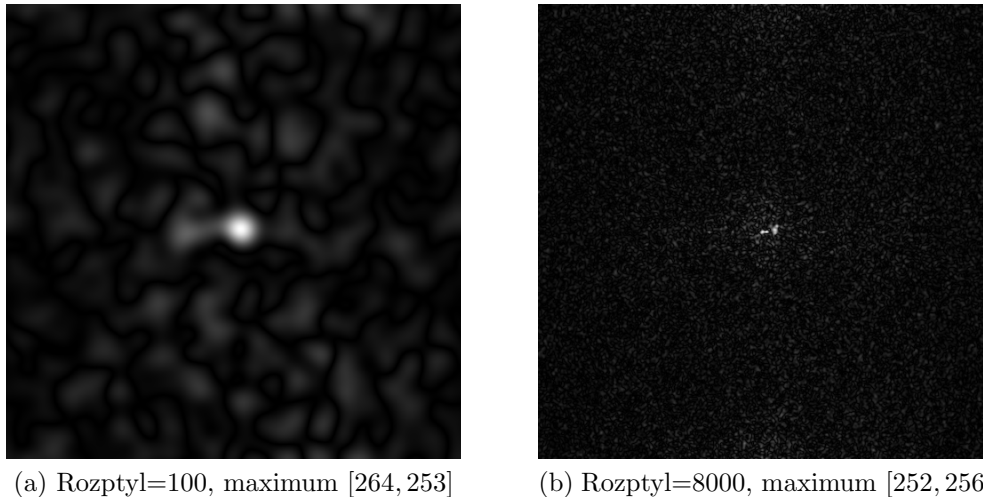


Obrázek 8.17: Obraz fázové korelace po odfiltrování

Výsledek korelace s odfiltrováním vysokých frekvencí je na obrázku (8.17). Výsledek je však velice citlivý vzhledem k použitému rozptylu Gaussovy funkce. Při volbě rozptylu

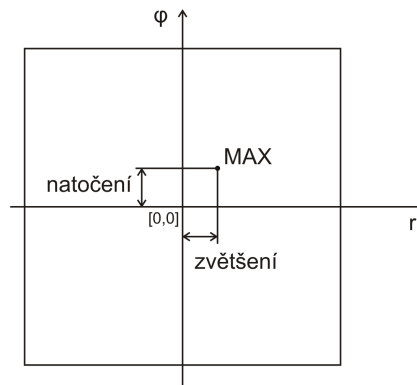
## 8. MĚŘENÍ RYCHLOSTI OBJEKTU ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

z intervalu  $\langle 600, 6000 \rangle$  pro náš případ vychází maximum na souřadnicích  $[266, 253]$ . Při volbě hodnoty rozptylu mimo tento interval se maximum v obraze posouvá. Pro jiné rozptyly je obraz ukázán na obrázku (8.18).



Obrázek 8.18: Výsledné obrazy po fázové korelaci s filtrováním

Nyní hledáme v obraze (8.17) maximum (kapitola 7.2). Souřadnice maxima ve svislém směru od středu nám určuje natočení původních obrazů vůči sobě. Souřadnice ve vodorovném směru od středu obrazu pak udává hodnotu zvětšení, kterou je nutno ještě odlogaritmovat, neboť máme obraz v log-polárních souřadnicích (obrázek 8.19).



Obrázek 8.19: Interpretace výsledku

V programu je k tomuto určena funkce `GeneralCorrelation`, která v sobě zahrnuje funkci `MultiplicationGauss`, která provádí již zmiňovanou konvoluci na frekvenčním spektru.

```
void Matrix::GeneralCorrelation(Matrix *p, Matrix *p1, Matrix *p2, double n1, double n2, int variability)
{
// konvoluce se provede, ovsem na spektrální oblasti(tj. v tomto
// pripade jeste pred zpetnou furrier transformaci)
p1->FFTFwd();
p2->FFTFwd();
p1->TransferSpectrum(p1->GetpArray());
p2->TransferSpectrum(p2->GetpArray());
p2->ComplexConjugate(p2->GetpArray());
Ipp32fc Number;
for (int i=0; i<p->GetSize().GetHeight(); i++)
```

### 8.3. METODA MĚŘENÍ RYCHLOSTI ZA VYUŽITÍ FOURIEROVY TRANSFORMACE

```

{
for (int j=0;j<p->GetSize().GetWidth();j++)
{
float a = p1->GetpArray(j,i).re;
float b = p1->GetpArray(j,i).im;
float c = p2->GetpArray(j,i).re;
float d = p2->GetpArray(j,i).im;
Number.re = (a*c - b*d)/((p1->GetpArraySpectrum(j,i).re+n1)*(p2->GetpArraySpectrum(j,i).re+n2));
Number.im = (a*d + b*c)/((p1->GetpArraySpectrum(j,i).re+n1)*(p2->GetpArraySpectrum(j,i).re+n2));
p->SetpArray(j,i,Number);
}
}
//preskladam pole --- provedu pronasobeni hodnot gauss funkci --- a preskladam zpět
// preskladání zaradí přímo metoda MultiplicationGauss
p->MultiplicationGauss(p->GetpArray(),variability);
////////////////////////////////////
p->FFTInv();
p->Reshuffle(p->GetpArray());
p->TransferSpectrum(p->GetpArray());
p->FindMaxMin(p->GetpArraySpectrum());
return;
}

```

#### 8.3.5. Výpočet rychlosti

Hodnota zvětšení pro náš případ vyšla 1,1146. K výpočtu rychlosti potřebujeme délku úseku, kterou objekt urazil. To je rozdíl vzdáleností  $s_0 - s_1$ . Vzdálenost  $s_1$  je určena z velikosti poznávací značky,  $s_0$  dostaneme jako násobek zvětšení a vzdálenosti  $s_1$ . Změna vzdálenosti je přímo úměrná zvětšení objektu. To znamená, že o kolik se zvětší objekt, o tolik se zvětší vzdálenost objektu.

$$\Delta s = s_0 - s_1 = s_0 - s_1 = k \cdot s_1 - s_1,$$

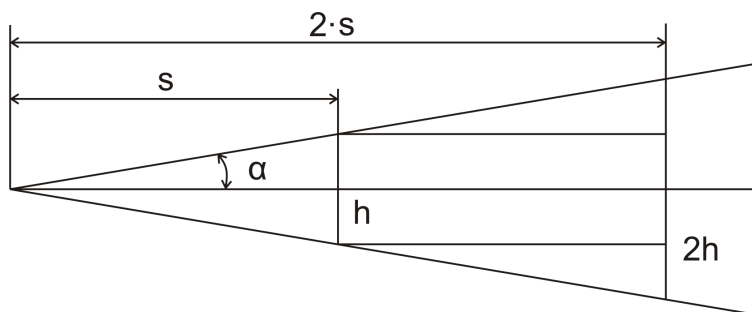
kde  $k$  je koeficient zvětšení.

Geometrické znázornění je na obrázku (8.20). Při načtení vstupních obrázků v opačném pořadí, můžeme dostat koeficient  $k$  menší než 1. Jedná se pak o koeficient zmenšení objektu. V tomto případě změním výpočet  $\Delta s$  a nebo koeficient zvětšení dostaneme převrácenou hodnotou koeficientu zmenšení.

Pak lze rychlost vypočítat dle vzorce (8.8).

$$v = \frac{s(k - 1)}{t}, \quad (8.8)$$

kde  $k$  je koeficient zvětšení a  $s$  je vzdálenost nejbližší naměřeného objektu.



Obrázek 8.20: Geometrické znázornění zvětšení

## 9. Praktické ověření

K ověření, zda nám výpočty dávají věrohodné výsledky jsme fotili vozidlo pohybující se od měřicího stanoviště. Snímky byly pořizeny digitálním fotoaparátem, který snímá objekt rychlostí 3 snímky za sekundu. Rychlost vozidla jsme měřili pomocí navigačního systému GPS.



Obrázek 9.1: Referenční snímek

Nejprve jsme provedli kalibraci snímacího zařízení (popsáno v kapitole 8.2), v našem případě to byla digitální zrcadlovka Canon EOS 30D + objektiv s ohniskovou vzdáleností 90 mm . Pořídili jsme referenční snímek, pro který bylo vozidlo ve vzdálenosti 42m od měřicího stanoviště, a v tomto obraze jsme naměřili délku poznávací značky 168 pixelů.

K výpočtu rychlosti potřebujeme v obraze změřit rozměr poznávací značky, která slouží pro prvotní určení vzdálenosti objektu. Pro jednoduchost provádíme měření ručně. Rozměr značky měříme ve snímku, ve kterém je objekt zachycen co nejbližší k měřicímu stanovišti. Většinou značku naměříme kolem 250 pixelů, proto chyba 5 pixelů má za následek pouze 2% chybu ve výpočtu.

### Ověření při rychlosti 23 km/h

První sérii fotek jsme pořizovali při rychlosti vozidla 23km/h (dle GPS). Snímek v čase



Obrázek 9.2: Snímek v čase  $t = 0$



(a) Snímek v čase  $t = 1\text{ s}$



(b) Snímek v čase  $t = 2\text{ s}$

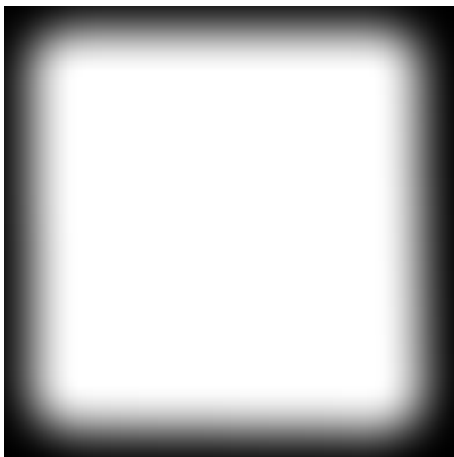
Obrázek 9.3: Snímky vzdálenějšího objektu

$t = 0\text{ s}$  je na obrázku (9.2). V tomto obraze jsme naměřili délku poznávací značky 254 px. Díky informacím z referenčního snímku jsme vypočetali nepřímou úměrou (vzorec 8.1) vzdálenost objektu 27,7m. To je potřebná hodnota pro výpočet rychlosti. Všechny snímky jsou upraveny na rozměr  $512 \times 512$ . Je jim přidělena maska, která ořeže okolí objektu (viz. kapitola 8.3.1). To je nutné vykonat před vložením snímků do programu. Různé masky jsou vidět na obrázku (9.4).

Snímky v časech  $t=1\text{ s}$  a  $t=2\text{ s}$  jsou na obrázku (9.3).

Koeficient zvětšení mezi snímkem v čase  $t = 0\text{ s}$  a snímkem v čase  $t = 1\text{ s}$  měl dle našeho programu hodnotu  $k = 1,242$ . Průměrná rychlost vozidla na úseku 6,8m pak vyšla  $v = 24\text{ km/h}$ .

Koeficient zvětšení mezi snímkem v čase  $t = 0\text{ s}$  a snímkem v čase  $t = 2\text{ s}$  měl dle našeho programu hodnotu  $k = 1,494$ . Průměrná rychlost vozidla na úseku 13,7m pak vyšla  $v = 24,7\text{ km/h}$ .



(a) 1.Maska



(b) 2.Maska

Obrázek 9.4: Různé tvary masek



**Ověření při rychlosti  $50\text{ km/h}$** 

Druhou sérii fotek jsme pořizovali při rychlosti vozidla  $50\text{ km/h}$  (dle GPS). Na snímku v čase  $t = 0\text{ s}$  jsme naměřili délku poznávací značky 250 pixelů. Díky informacím z referenčního snímku jsme vypočítali, že objekt byl v čase  $t = 0\text{ s}$  vzdálen  $28,2\text{ m}$  od stanoviště snímacího zařízení.

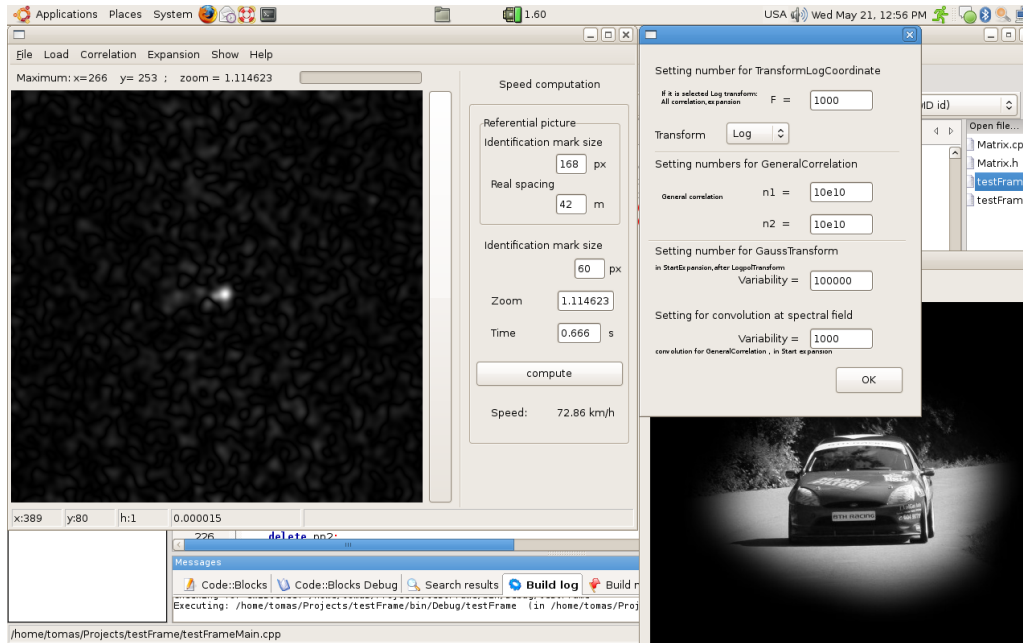
Koeficient zvětšení mezi snímkem v čase  $t = 0\text{ s}$  a snímkem v čase  $t = 0,667\text{ s}$  měl dle našeho programu hodnotu  $k = 1,355$ . Průměrná rychlost vozidla na úseku  $10\text{ m}$  pak vyšla  $v = 54,1\text{ km/h}$ .

**Ověření při rychlosti  $80\text{ km/h}$** 

Třetí sérii fotek jsme pořizovali při rychlosti vozidla  $80\text{ km/h}$  (dle GPS). Na snímku v čase  $t = 0\text{ s}$  jsme naměřili délku poznávací značky 229 pixelů. Díky informacím z referenčního snímku jsme vypočítali, že objekt byl v čase  $t = 0\text{ s}$  vzdálen  $30,8\text{ m}$  od stanoviště snímacího zařízení.

Koeficient zvětšení mezi snímkem v čase  $t = 0\text{ s}$  a snímkem v čase  $t = 0,333\text{ s}$  měl dle našeho programu hodnotu  $k = 1,242$ . Průměrná rychlost vozidla na úseku  $7,4\text{ m}$  pak vyšla  $v = 80,67\text{ km/h}$ .

# 10. Jednoúčelová aplikace



Obrázek 10.1: Ukázka programu počítajícího zvětšení

K naprogramování aplikace jsme použili těchto prostředků:

Ubuntu

je komunitní operační systém na bázi Linuxu.

C++ programovací jazyk

je rozšíření jazyka C. C++ podporuje několik programovacích stylů jako je procedurální programování, objektově orientované programování a generické programování.

Code::Blocks

je svobodné multiplatformní vývojové prostředí zaměřené na programovací jazyky C a C++. Podporuje využití více překladačů. Je vyvíjeno v C++ s využitím wxWidgets toolkitu.

wxWidgets

je free software/open source multiplatformní widget toolkit. Je to knihovna základních elementů pro tvorbu grafického uživatelského rozhraní (GUI). Knihovna je implementována v C++, ale její používání je možné v mnoha běžně používaných programovacích jazycích.

GCC

jsou kompilátory jazyka C i C++, které najdeme v mnoha operačních systémech. V Linuxu se standardně používá GCC (The GNU Compiler Collection).

Intel Integrated Performance Primitives for Intel

je knihovna od intelu napsaná v C++ a je pod operačním systémem Linux volně ke stažení. Z této knihovny jsme využili funkci FFT.

# 11. Závěr

Podářilo se nám vytvořit program pro výpočet rychlosti objektu ze dvou snímků o rozměrech  $512 \times 512$ . Program je schopen načíst a uložit snímky ve formátech JPG a PNG. Po načtení dvou snímků lze v programu provést výpočet na koeficient zvětšení. A po naměření rozměru značky ve snímku bližšího objektu vypočítá rychlost.

Z výsledků praktického měření (kapitola 9) je vidět, že vypočtená rychlost je velmi podobná rychlosti naměřené pomocí navigačního systému GPS. Zde je otázkou, jak moc přesně systém GPS dokáže rychlost měřit. V první a třetí sérii snímků je rychlost téměř totožná, liší se o desetinu hodnoty. U druhé série snímků při rychlosti  $50 \text{ km/h}$  se vypočtená hodnota liší od naměřené asi o 8%. To mohlo být způsobeno nerovnoměrnou rychlostí jízdy vozidla. Lze tedy říct, že metoda měření rychlosti objektu pomocí Fourierovy transformace funguje spolehlivě.

K výpočtu rychlosti potřebujeme v obraze změřit rozměr značky. Měřením poznávací značky vnášíme chybu do výpočtu rychlosti, protože měření závisí na subjektivním posouzení. Měření by šlo zpřesnit automatickou identifikací značky a změřením její délky.

Bylo by výhodné snímat objekt po častějších časových intervalech a počítat tak vývoj rychlosti v závislosti na čase, protože však počítáme rychlost pouze ze dvou snímků, dostaneme pouze průměrnou rychlost objektu v daném časovém intervalu.

Program na výpočet rychlosti by se mohl dále vyvíjet tak, aby automaticky rozpoznal poznávací značku a změřil její rozměr. Dále by program mohl umět automaticky přiřadit masku načítaným obrázkům, která je důležitá pro správnou registraci obrazů. Tyto doplňky jsou již nad rámec této diplomové práce.

Tato metoda je univerzální metodou měření rychlosti v ose pohybu daného objektu, u kterého stačí znát jeho geometrické vlastnosti.

# Literatura

- [1] BARTOŇ, M. Úvod, seznámení. *Aliasing a antialiasing* [online]. 2002, [cit.2008-15-05]. Dostupný z WWW: <file:///home/tomas/Desktop/Diplomova prace/Podklady/digitalizace/antialias1.html>
- [2] BEZVADA, V.,JEŽEK, J.,SAIC, S.,SEGETH, K. *Dvojměrná diskrétní Fourierova transformace a její použití I.Státní pedagogické nakladatelství, n. p. Praha 1, 1988*
- [3] DVOŘÁK, D. Světlocitlivé snímací prvky. *Digitální fotoaparát V* [online]. 2003, [cit.2008-15-05]. Dostupný z WWW: <http://www.digimanie.cz/art\_doc-D80A41BB4E82A7AFC1256CC60031E152.html>
- [4] FIŘT, J.,HOLOTA, R. *Digitalizace a zpracování obrazu* [online]. [cit.2008-15-05]. Dostupný z WWW: <http://home.zcu.cz/~holota5/publ/DigZprO.pdf>
- [5] HUMPOLÍČEK, J.: *Vyhodnocování snímků z mamogramu*. Praha, 2004. 70 s. Diplomová práce na Matematicko-fyzikální fakultě Univerzity Karlovy na katedře softwaru a výuky informatiky. Vedoucí diplomové práce RNDr. Ing. Jiřina Marcel, Ph.D. Dostupný na WWW: <http://www.ms.mff.cuni.cz/~jhum8111/docs/mammo.pdf>
- [6] KLÍMA, M.,BERNAS, M.,HOZMAN, J.,DVOŘÁK, P. *Zpracování obrazové informace. ČVUT 1996*
- [7] KOLMOGOROV, A. N.. FOMIN, S.V. *Základy teorie funkcí a funkcionální analýzy.SNTL Praha 1975*
- [8] RADOVAN. Digitalizace obrazu. *Graficky orientované IS* [online]. 2006, [cit.2008-15-05]. Dostupný z WWW: <http://radovan.blogger.cz/it/graficky-orientovane-IS/digitalizace-obrazu>
- [9] STANĚK, M. Řidiči versus radary. *VTM Science*, říjen 2006, roč. 60, č 10, s. 45-46
- [10] TAKITA, K. High-Accuracy Subpixel Image Registration Based on Phase-Only Correlation *Special Section on Digital Signal Processing* [online]. 2003, [cit.2008-15-05]. Dostupný na WWW: <http://www.aoki.ecei.tohoku.ac.jp/research/docs/e86-a\_8\_1925.pdf>
- [11] WIKIPEDIA, The Free Encyclopedia. *Barevný model* [online]. Poslední revize 18.4.2008 [cit.2008-15-05]. Dostupný z WWW:<http://cs.wikipedia.org/wiki/Barevný\_model>
- [12] WIKIPEDIA, The Free Encyclopedia. *Konvoluce* [online]. Poslední revize 15.2.2008 [cit.2008-15-05]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Konvoluce>

## 12. Seznam použitých zkratek a symbolů

$F$	frekvenční spektrum obrazu
$\bar{F}$	frekvenční spektrum obrazu, jehož prvky jsou komplexně sdužená čísla
$\mathfrak{F}$	přímá Fourierova transformace
$\mathfrak{F}^{-1}$	zpětná Fourierova transformace
$f(x, y)$	dvourozměrné pole obrazu
$\mathbf{g}$	konvoluční jádro
$\delta$	Dirakov impuls
$N$	rozměr obrazové matice
$px$	jednotka pixel
GPS	(Global Positioning System) vojenský polohový družicový systém