

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

DIPLOMOVÁ PRÁCE

Změny ve fotografii a způsob jejich detekce



2011

Bc. Roman Hájek

Anotace

Práce se zabývá metodami detekce změn v digitální fotografii. Blíže se věnuje metodám detekce změn provedených duplikováním částí obrazu. Druhá zkoumaná metoda využívá k odhalení změn inkonzistenci šumu v obraze.

Děkuji vedoucímu diplomové práce Ing. Michalovi Dobešovi, Ph.D. za trpělivé vedení práce, cenné rady, připomínky a čas strávený při konzultacích.

Obsah

1. Úvod	9
2. Metody detekce duplikovaných oblastí	10
2.1. Metody hodnocení podobnosti	10
2.1.1. Přesná shoda bloků	10
2.1.2. Porovnání pomocí MSE a PSNR	11
2.1.3. SSIM	12
2.1.4. Korelační koeficient	13
2.1.5. Autokorelace ve frekvenční doméně	14
2.2. Časová náročnost výpočtů	15
2.3. Analýza obrazu ve frekvenční doméně	16
2.3.1. Vektory posunutí	17
2.3.2. Koeficienty korelace	20
2.4. Analýza obrazu v prostorové doméně	21
2.4.1. Použití SSIM	22
2.4.2. Použití MSE	23
2.4.3. Použití korelačního koeficientu	23
2.5. Předcházení falešným detekcím	24
2.5.1. Plochy s nedostatkem detailů	24
2.5.2. Opakující se vzory	25
2.6. Ukázky	27
2.7. Optimalizace	29
2.7.1. Blokové operace	29
2.7.2. Optimalizace Fourierovy transformace	29
2.7.3. Procházení s větším krokem	30
3. Metody zkoumající inkonzistenci šumu	32
3.1. Šum v digitálním obraze	32
3.1.1. Typy digitálního šumu	32
3.1.2. Vlastnosti digitálního šumu	33
3.1.3. Model šumu	36
3.1.4. Uměle přidaný šum	37
3.2. Odhad šumu	38
3.2.1. Diskrétní waveletová transformace	38
3.2.2. Gaussův filtr	40
3.2.3. Bilaterální filtr	41
3.3. Měření úrovně šumu	43
3.3.1. Odstup signál-šum	43
3.3.2. Směrodatná odchylka	44
3.4. Nalezení oblastí s různými úrovněmi šumu	45
3.4.1. Detekce hran	46

3.5. Ukázky	48
3.5.1. Aplikace DWT	49
3.5.2. Aplikace Gaussova filtru	51
3.5.3. Aplikace bilaterálního filtru	52
3.6. Hodnocení metod	53
3.7. Srovnání metod	54
Závěr	56
Reference	57
A. Výsledky odhadu šumu	60
B. Srovnání metod detekce inkonzistence šumu	63
C. Obsah přiloženého CD	68

Seznam obrázků

1.	Vektory posunutí pro duplikované oblasti	17
2.	Armádní auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů	18
3.	Armádní auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů	19
4.	Armádní auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů	21
5.	Detekce s použitím SSIM	22
6.	Detekce s použitím MSE	23
7.	Detekce s použitím korelačního koeficientu v prostorové doméně	24
8.	Vyloučení ploch s nedostatkem detailů, maska 16×16 pixelů	25
9.	Obrázek s opakující se strukturou	26
10.	Zápalky - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 512×512 pixelů	27
11.	Auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 512×512 pixelů	28
12.	Armádní auta - maska 32×32 pixelů, velikost obrázku 256×256 pixelů	31
13.	Porovnání velikostí obrazových snímačů, převzato z [29]	33
14.	Bayerova maska, převzato z [28]	34
15.	Srovnání úrovně šumu pro jednotlivé barevné kanály	35
16.	Porovnání výřezů obrázků pořízených s různým nastavením ISO, fotoaparát Panasonic DMC-FZ7	35
17.	Porovnání výřezů obrázků pořízených fotoaparátů s různou velikostí senzoru při stejném nastavení ISO 400	36
18.	Lokální přidání šumu do obrazu v programu Photoshop	37
19.	Histogram aditivního bílého gaussovského šumu	38
20.	Princip 2D DWT, převzato z [17]	39
21.	Konvoluční jádro pro Gaussovův filtr 30×30 px, $\sigma = 5$	40
22.	Princip bilaterálního filtru, převzato z [13].	41
23.	Hodnoty SNR v kombinaci s aplikací Gaussova filtru	44
24.	Směrodatná odchylka šumu	45
25.	Postup při segmentaci obrazu metodou hledání hranice	46
26.	Testovací obrazy s ohraničenými zkopírovanými oblastmi.	48
27.	Výřezy z použitých testovacích obrazů.	49
28.	Detekce pomocí DWT pro obraz č.1.	50
29.	Detekce pomocí DWT pro obraz č.2.	50
30.	Detekce pomocí Gaussova filtru pro obraz č.1.	51
31.	Detekce pomocí Gaussova filtru pro obraz č.2.	51
32.	Detekce pomocí bilaterálního filtru pro obraz č.1.	52
33.	Detekce pomocí bilaterálního filtru pro obraz č.2.	53

34.	Grafické znázornění chyb	54
35.	Odhad šumu za použití diskrétní waveletové transformace	60
36.	Odhad šumu za použití Gaussova filtru.	61
37.	Odhad šumu za použití bilaterálního filtru.	62
38.	Obrázek „Letadlo“, velikost 1200×660 pixelů	63
39.	Obrázek „Dóm“, velikost 24880×3818 pixelů	64
40.	Obrázek „Severní pól“, velikost 1600×1200 pixelů	65
41.	Obrázek č.1, velikost 1000×1500 pixelů	66
42.	Obrázek č.2, velikost 1000×1500 pixelů	67

Seznam tabulek

1.	Prahové hodnoty koeficientů podobnosti	28
2.	Zrychlení operace výřezu	29

1. Úvod

V době, kdy digitální fotografie již téměř nahradila analogovou a vývoj softwaru pro editaci fotografií jde stále kupředu je manipulace fotografií a vytváření snímků nereálných situací stále snadnější. Vedle nástrojů pro editaci fotografií, které jsou k použití zdarma, jsou k mání na trhu komerční nástroje, za všechny zmíníme například populární Photoshop. S použitím těchto nástrojů lze do digitálních fotografií přidat libovolné objekty nebo je naopak odebrat. S klesající náročností těchto úprav stoupá potřeba tyto úpravy odhalit. Padělané fotografie lze zneužít například k pojistným podvodům, kdy dokládáme fotografickou dokumentaci. Dále k dezinformaci v médiích, kdy lze podvrhnout reportážní fotografie, k padělání důkazních fotografií v soudním řízení nebo například jen k úpravě fotografie v internetové seznamce.

Způsobů, jak fotografii upravit je několik, v této práci se budeme věnovat detekci dvou způsobů padělání fotografií. Prvním z nich bude detekce klonovaných oblastí ve fotografii, kdy duplikujeme část fotografie jiné místo v téže fotografii s cílem skrýt nějakou důležitou informaci. Druhým pak bude detekce oblastí, které byly do fotografie přidány z jiné fotografie. Tato metoda bude založena na zkoumání hladiny šumu a pro úspěšnou detekci předpokládá odlišné vlastnosti šumu v obou zdrojových fotografiích. U obou metod si představíme více přístupů a na závěr je porovnáme na testovacích fotografiích.

2. Metody detekce duplikovaných oblastí

V této kapitole se budeme věnovat metodám detekce fotomontáže, která byla vytvořena duplikováním určité části obrazu na jiné místo v témže obraze. Typickou technikou je použití nástroje „Klonovací razítko“. Tento nástroj je dostupný v mnoha editorech určených ke zpracování fotografií. Častá aplikace je například zkopírování pravidelného vzoru (textury) přes jinou oblast obrázku, kterou chceme například zakrýt. Lidské oko obtížně rozpozná, že byla nějaká manipulace provedena.

Při zpracování počítačem můžeme využít různých metod, jak manipulaci odhalit. Narážíme však na určitá omezení. Jelikož duplikované vzorky pocházejí ze stejného obrázku, mají stejnou šumovou složku, barevnou paletu i dynamický rozsah – všechny důležité parametry obrazu jsou homogenní a zkoumání jejich nesouvislosti nelze k odhalení fotomontáže použít.

Vhodné je hledat vztah mezi jedním segmentem z obrázku a jiným, který by mohl být zkopírován. Vzhledem k častému použití ztrátové JPEG komprese a dodatečným nastavením klonovacího nástroje v editoru fotografií se nemusejí bloky shodovat přesně, ale pouze přibližně. Základní požadavky na algoritmus jsou

- Rozpoznání přibližné shody malých segmentů v obraze
- Co nejmenší počet nesprávných detekcí
- Zohledňuje, že výsledná naklonovaná oblast je spíše větší a souvislá než velké množství malých oblastí

2.1. Metody hodnocení podobnosti

V této kapitole si z teoretického hlediska představíme několik metod, které lze použít k porovnávání podobnosti dvou obrazů, respektive jejich výřezů, jelikož při detekování klonovaných oblastí je nutno mezi sebou porovnávat oblasti, které jsou menší, než je předpokládaná velikost duplikované oblasti.

2.1.1. Přesná shoda bloků

Založeno na jednoduché myšlence, že definujeme velikost čtvercové masky, pod pod kterou budeme postupně pořizovat výřezy z obrazu. Začneme například z levého horního rohu a vždy se posuneme o jeden pixel doleva, po přečtení řádku posuneme masku opět na začátek a začneme o jeden pixel níže. Pro masku o velikosti $b \times b$ pixelů a obrázek o velikosti $m \times n$ pixelů získáme $(m - b + 1) \times (n - b + 1)$ výřezů.

Nyní mezi těmito vzorky budeme vyhledávat takové, které jsou stejné. Pro redukci složitosti výpočtu lze výřezy lexikograficky seřadit do tabulky a v této

tabulce pak hledat sousední záznamy, které budou buď velmi podobné nebo dokonce shodné. Z nalezených shod bychom pak mohli odhalit duplikované oblasti. Velkým nedostatkem této metody je její robustnost. Pokud by byla duplikovaná oblast jen nepatrně pozměněna například rozmazáním, přidáním šumu nebo aplikací JPEG transformace, nepovedlo by se nám ji odhalit. Jak jsme se zmínili, náš algoritmus by měl rozpoznat i pouze přibližnou shodu.

2.1.2. Porovnání pomocí MSE a PSNR

Pro porovnání výřezů lze použít taktéž metodu určení střední kvadratické chyby, neboli MSE¹ mezi těmito bloky. Index MSE [26] pro dva obrazy je dán vztahem

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{i,j} - K_{i,j})^2 \quad (1)$$

kde I, K jsou funkce jasu pro porovnávané obrazy o rozměrech $m \times n$ pixelů. Pro zpřehlednění budeme hodnotu MSE normovat do intervalu $< 0, 1 >$. Normování provedeme dle vztahu

$$MSE_{norm} = \frac{MSE}{(I_{max})^2} \quad (2)$$

kde I_{max} je maximální hodnota jasu v obraze. V našem případě, kdy používáme 8bitů na jeden barevný kanál bude $I_{max} = 255$.

MSE určuje podobnost dvou obrazů v tom smyslu, že počítá chybu mezi těmito dvěma obrazy. Čím je chyba menší, tím jsou si oba obrazy podobnější. Výhodou použití této metody je poměrně nízká výpočetní složitost.

Metodou, která vychází z MSE je výpočet odstupů signálu od šumu, neboli PSNR² [25]. Metoda je běžně používána například při hodnocení kvality ztrátové komprese JPEG. Signál zde reprezentuje původní data a šumem rozumíme chyby způsobené ztrátovou kompresí. Stejně dobře ji můžeme použít i při zkoumání podobnosti dvou bloků z obrazu. Vztah pro výpočet hodnoty PSNR je

$$PSNR = 20 \log_{10} \left(\frac{I_{max}}{\sqrt{MSE}} \right) \quad (3)$$

Jelikož PSNR přímo vychází z MSE, budeme při porovnávání používat pouze jednu z těchto metod. Otázkou zůstává stanovení prahové hodnoty, při které budeme považovat porovnávané bloky za shodné. Stanovení prahu se budeme věnovat kapitole zabývající se aplikací této metody.

¹Mean Square Error

²Peak Signal-to-Noise Ratio

2.1.3. SSIM

SSIM³ [4] zahrnuje do zkoumání podobnosti obrazů i jeho strukturu, která nese vzhledem k HVS⁴ větší část informace, než samotná hodnota jasu jednotlivých pixelů u kterých bychom neuvažovali jejich vzájemný vztah. Metoda je odolná vůči malým změnám v obraze, pokud není příliš změněna jeho struktura.

V indexu SSIM, který vyjadřuje podobnost dvou obrazů jsou zohledněny 3 nezávislé parametry porovnávaných obrazů. Jsou to jas, kontrast a samotná struktura obrazu. Definujeme tedy tři funkce, které nám budou tyto vztah těchto parametrů vyjadřovat a z těchto pak složíme jednu funkci pro výpočet samotného indexu.

Pro porovnání **jasu** použijeme funkci

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (4)$$

kde

- μ_x a μ_y jsou střední hodnoty náhodných veličin
- C_1 je konstanta blízká se nule

Konstanta C_1 je zahrnuta pro případ, že by součet $\mu_x^2 + \mu_y^2$ byl nulový. Pro porovnání **kontrastu** použijeme funkci

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (5)$$

kde

- σ_x a σ_y jsou rozptyly náhodných veličin
- C_2 je konstanta blízká se nule, opět ze stejného důvodu jako v předchozím vzorci

Konečně pro porovnání **struktury** použijeme funkci

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (6)$$

kde

- C_3 je opět malá konstanta

Výchozí hodnoty těchto konstant si uvedeme níže. Tato funkce má následující vlastnosti:

³Structural Similarity

⁴Human Visual System

1. $s(x, y) = s(y, x)$ tedy symetrii
2. $s(x, y) \leq 1$ tedy ohraničenost
3. $s(x, y) = 1 \Leftrightarrow x = y$

Nyní složíme uvedené tři funkce do jedné, která nám bude vyjadřovat podobnost mezi vstupními signály (obrazy):

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (7)$$

kde

- α, β, γ jsou hodnoty větší než 0 a určují nám váhu jednotlivé složky ve výsledném porovnání.

Pro jednoduchost položíme koeficienty α, β a γ rovny 1. Dále položíme $C_3 = C_2/2$. Pak po úpravách a dosazení dostáváme vzorec, který můžeme použít pro výpočet indexu podobnosti:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (8)$$

2.1.4. Korelační koeficient

Korelační koeficient [27] je důvěrně známý pojem ze statistiky. Vyjadřuje míru lineární závislosti mezi dvěma náhodnými veličinami. My jej využijeme k hodnocení podobnosti dvou výřezů z obrazu. Budeme tedy zjišťovat, nakolik jsou na sobě lineárně závislé. Čím větší koeficient nám vyjde, tím je pravděpodobnější, že jsme našli duplikované oblasti.

V manuálu k softwaru MATLAB [18], se kterým pracujeme, je uveden následující vztah pro výpočet korelačního koeficientu

$$r = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{i,j} - \bar{I})(K_{i,j} - \bar{K})}{\sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{i,j} - \bar{I})^2 \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (K_{i,j} - \bar{K})^2}} \quad (9)$$

kde

- I, K jsou funkce jasu pro porovnávané obrazy o rozměrech $m \times n$ pixelů
- \bar{I}, \bar{K} jsou střední hodnoty jasu

Korelační koeficient může nabývat hodnot z intervalu $\langle -1, 1 \rangle$, kde záporné hodnoty indikují nepřímou lineární závislost a kladné hodnoty indikují přímou lineární závislost. Podobnost dvou obrazů bude tím větší, čím více se bude hodnota korelačního koeficientu blížit horní hranici tohoto intervalu. Poznamenejme, že v našem případě nebudeme počítat koeficient pro celé obrazy, ale pouze pro jejich výřezy. Podrobněji se budeme aplikací korelačního koeficientu zabývat v kapitole 2.4.3.

2.1.5. Autokorelace ve frekvenční doméně

Fourierova transformace [1] je založena na myšlence, že každá periodicky se opakující funkce může být vyjádřena jako součet funkcí *sin* a *cos* o různých frekvencích a amplitudách. Tato myšlenka byla publikována již v roce 1822 francouzským matematikem J.B.J. Fourierem.

Jelikož se budeme zabývat zpracováním digitálního obrazu, který bývá reprezentován jako dvourozměrné pole reálných čísel, bude nás zajímat diskrétní Fourierova transformace (DFT). Ta se spočítá pro funkci dvou proměnných $f(x, y)$, která reprezentuje obraz o rozměrech $m \times n$ pixelů dle vztahu

$$F(u, v) = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y) e^{-j2\pi(ux/m + vy/n)} \quad (10)$$

Po substituci $e^{j\theta} = \cos \theta + j \sin \theta$ dostáváme vztah

$$F(u, v) = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y) [\cos(2\pi(ux/m + vy/n)) - j \sin(2\pi(ux/m + vy/n))] \quad (11)$$

Zpětná transformace (inverzní DFT) je pak dána vztahem

$$f(x, y) = \frac{1}{mn} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/m + vy/n)} \quad (12)$$

na který lze opět aplikovat substituci $e^{j\theta} = \cos \theta + j \sin \theta$

$$f(x, y) = \frac{1}{mn} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) [\cos(2\pi(ux/m + vy/n)) + j \sin(2\pi(ux/m + vy/n))] \quad (13)$$

Korelační teorém

Korelace nám při práci s digitálním obrazem slouží ke zkoumání podobnosti mezi obrazy nebo jejich částmi. V prostorové doméně se korelace pro dvě funkce

dvou proměnných $h(x, y)$ a $f(x, y)$, potažmo dva obrazy (o rozměrech $m \times n$ pixelů) počítá dle vztahu

$$f(x, y) \circ h(x, y) = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f^*(m, n) h(x + m, y + n) \quad (14)$$

Symbol f^* označuje komplexně sdružené číslo, my však obvykle budeme pracovat s reálnými čísly, můžeme proto položit $f^* = f$. Pro obrazy větších rozměrů může být vzhledem k počtu matematických operací výpočetně poměrně náročné spočítat korelaci. Z výpočetního hlediska je tedy vhodnější korelaci realizovat ve frekvenční doméně. Zde využijeme korelační teorém. Ten je dán vztahem

$$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v)H(u, v) \quad (15)$$

Tedy korelace v prostorové doméně je ekvivalentní násobení ve frekvenční doméně. V praxi to znamená, že jsme složitý výpočet zredukovali na pouhé vynásobení dvou matic. Samozřejmě musíme pro oba obrazy spočítat DFT a poté pro výsledek inverzní DFT, ale i tak je operace celkové méně výpočetně náročná.

2.2. Časová náročnost výpočtů

V kapitole 2. jsme si popsali několik metod, jak lze mezi sebou porovnávat bloky obrazu a zjišťovat tak jejich podobnost. Pokud budeme pracovat v prostorové doméně, nabízí se nám jednoduchý postup – porovnávat regiony v obraze každý s každým a hledat takové dvojice, které jsou téměř nebo úplně stejné. Zvolíme masku o určité velikosti, tu postupně posouváme po obraze a pro každý tento krok porovnáme obsah výřezu se všemi možnými výřezy, které získáme opět posouváním masky po obraze. Z hlediska implementace jde o dva vnořené cykly. Pokud uvážíme obrázek o velikosti $m \times n$ pixelů a masku o velikosti $b \times b$ pixelů, celkový počet porovnání dvou výřezů některou z uvedených metod pro prostorovou doménu bude

$$\frac{[(m - b + 1)(n - b + 1)]^2}{2} \quad (16)$$

za předpokladu, že výpočet koeficientu podobnosti dvou výřezů je symetrický, což jsou všechny uvedené v předchozích kapitolách.

To je poměrně mnoho operací, nabízí se tedy využití korelace ve frekvenční doméně, popsané v kapitole 2.1.5. Pomocí této metody si vytipujeme oblasti, které by mohly být potenciálně duplikované a tyto budeme dále zkoumat v prostorové doméně. Pro získání přehledu o tom, kde se vyskytují potenciálně duplikované oblasti nám bude stačit $(m - b)(n - b) + 1$ operací diskrétní Fourierovy transformace a $(m - b)(n - b)$ operací inverzní diskrétní Fourierovy transformace. Tyto transformace jsou náročnější, než porovnávací operace v prostorové doméně, ale neprovádíme jich tolik. V kapitole 2.7. se budeme zabývat postupy, jak výpočet optimalizovat, například dalším snížením počtu operací.

2.3. Analýza obrazu ve frekvenční doméně

Data pro další analýzu získáme pomocí výpočtu korelace ve frekvenční doméně. Budeme postupovat tak, že pro každý výřez určený aktuální polohou masky spočítáme Fourierovu transformaci, výsledek následně vynásobíme ve frekvenční doméně s Fourierovou transformací původního obrazu a převedeme zpět do prostorové domény pomocí inverzní Fourierovy transformace. Získáme tak obraz, ve kterém budou vysoké hodnoty jasu v místech, kde byl původní obraz podobný či shodný s použitým výřezem. Nejvyšší hodnota korelace bude v místě, odkud jsme pořídili výřez. Hodnoty pod maskou (sem patří i nejvyšší hodnota) budeme ignorovat a vezmeme hodnotu druhou největší, kterou si uložíme. Dále si uložíme vektor, o který je posunuto umístění této hodnoty v obraze oproti aktuální poloze masky. Jelikož se jedná o složitější postup, uveďme si algoritmus v pseudokódu:

Algoritmus 1 Analýza ve frekvenční doméně

Vstup: maska, obraz, krok

Výstup: vektoryPosunutí, hodnotyKorelace

```
(b,b)=size(maska);
(m,n)=size(obraz);
vektoryPosunutí=zeros(m,n);
hodnotyKorelace=zeros(m,n);
OBRAZ=fft2(obraz);
i=0, j=0;
while i < m do
    while j < n do
        vyrez=obraz(i:i+b,j:j+b);
        VYREZ=fft2(vyrez);
        K=VYREZ .* OBRAZ; {násobení matic prvek po prvku}
        k=ifft2(K);
        k(i:i+b,j:j+b)=0; {hodnoty pod maskou}
        hodnotyKorelace(i,j)=max(k);
        (poziceX,poziceY)=find(k,max(k));
        vektoryPosunutí(i,j)=(poziceX-i,poziceY-j);
        j+=krok;
    end while
    i+=krok;
end while
```

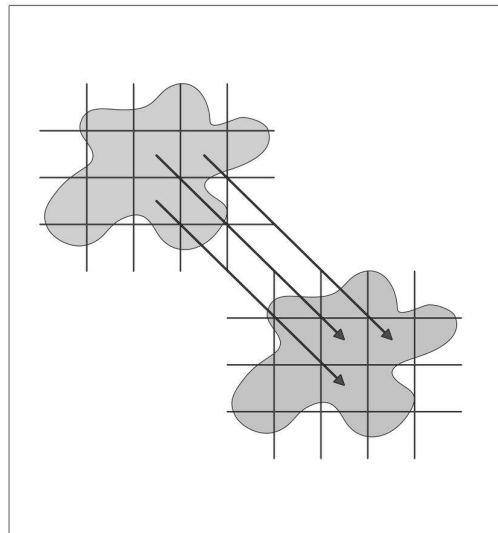
Poznamenejme, že názvy funkcí jsou analogií z MATLABu, v tomto prostředí jsme rovněž prováděli implementaci, viz příloha.

Pro každou polohu masky v obraze jsme tak pomocí vektoru posunutí získali informaci, kde leží „nejpodobnější“ region a pomocí hodnoty korelace údaj, „jak moc je podobný“. Pro další zkoumání, popsané v následujících kapitolách

máme k dispozici kromě zkoumaného obrazu navíc dvě dvojrozměrná pole, pracovně nazvaná `vektoryPosunuti` a `hodnotyKorelace`. Ta mají stejné rozměry, jako zkoumaný obraz. Mohly by být sice menší vzhledem k tomu, že parametr `krok` uvedeného algoritmu může být větší, než 1 pixel, ale pro zjednodušení ponecháváme uvedené rozměry a dále budeme pracovat s poli, kde nejsou využity všechny hodnoty.

2.3.1. Vektory posunutí

Nejprve si nastíníme myšlenku, na které je tato metoda založena. Ta je ilustrována na obrázku 1. Předpokládejme obraz, ve kterém jsme zduplikovali určitou jeho část na jiné místo. Obě tyto části nyní rozdělíme na určitý počet bloků. Pro každý pár sobě příslušejících bloků (první bude z originální oblasti, druhý z duplikované) spočítáme vektor o který je posunut druhý blok oproti prvnímu. Tyto vektory budou pro všechny páry bloků náležící do duplikované oblasti stejné.



Obrázek 1.: Vektory posunutí pro duplikované oblasti

Postup použitý u této metody spočívá v procházení dvourozměrného pole vektorů posunutí a hledání souvislých oblastí se stejnou hodnotou vektoru. Tato oblast bude představovat hledanou zkopírovanou oblast. Použijeme podobný postup jako v předchozí kapitole. Stanovíme si masku o velikosti $b \times b$ pixelů a s použitím této masky budeme postupně procházet mapu vektorů `vektoryPosunuti`. Pro každou podmnožinu vektorů V , danou aktuální polohou masky, spočítáme rozptyl dle vztahu 17. Rozptyl vypočítáme pro každou složku vektoru zvlášť, pracujeme se dvěma podmnožinami V_x a V_y . Ideálně najdeme pozice masky, kde rozptyl bude v obou složkách nulový, tedy všechny hodnoty jsou stejné.

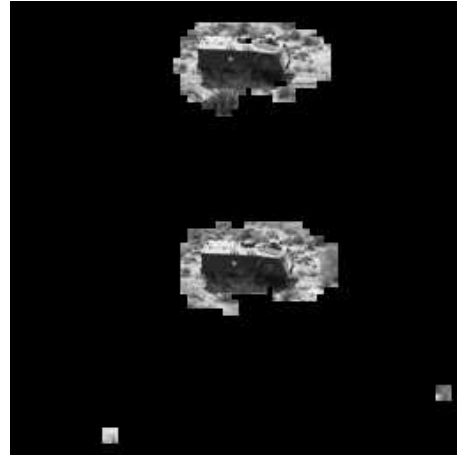
$$\text{var}(V_x) = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (V_x[i, j] - \bar{V}_x)^2 \text{ pro } \forall V_x[i, j] > 0 \quad (17)$$

Pro větší robustnost lze zavést toleranci a to tak, že vypustíme krajní hodnoty (nejmenší a největší) a poté vypočítáme rozptyl. Dále je nutno poznamenat, že mapa vektorů posunutí může být řídká vzhledem k tomu, že jsme při jejím výpočtu mohli postupovat s krokem větším, než 1 pixel. Z tohoto důvodu počítáme pouze s hodnotami většími, než nula. Nulová může být pouze hodnota, která není v poli využita, vzhledem k tomu, že v algoritmu 1 předem vylučujeme nejvyšší korelaci výřezu se samým sebou. V tomto případě by byl vektor posunutí $(0, 0)$.

Pokud se nám podaří najít větší souvislé oblasti v mapě vektorů posunutí, zkoumaný obraz pravděpodobně obsahuje duplikované oblasti. Jejich umístění, tvar a velikost nám prozradí přímo výsledky ze zkoumání vektorů posunutí. Výhodou tohoto přístupu je jednoznačně to, že vidíme, které dvojice umístění masky k sobě patří. Ty mají totiž oproti sobě stejný vektor posunutí až na orientaci. Znalost vektoru posunutí mezi těmito umístěními masky nám výrazně zjednoduší zkoumání dalšími metodami, které pracují v prostorové doméně a jsou založeny na porovnávání dvou regionů. Stačí nám v cyklu procházet podezřelé oblasti a pro každé umístění masky získáme potenciální duplikát snadno pomocí vektoru posunutí.



(a) Podvržený obrázek



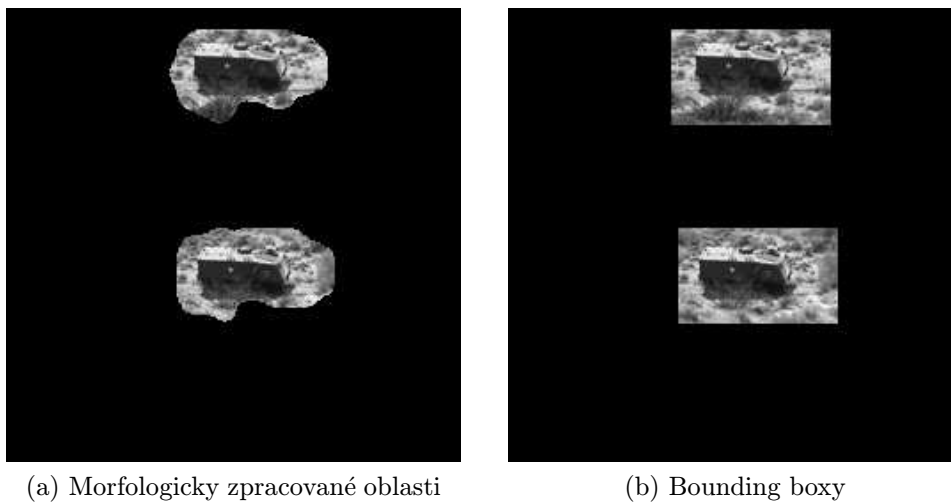
(b) Vytipované oblasti pomocí vektorů

Obrázek 2.: Armádni auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů

Problém může nastat, pokud se v obrázku vyskytuje větší počet malých oblastí, které jsou si hodně podobné. Pak výsledkem této metody bude mnoho dvojic oblastí, mezi kterými nás ale nebudou zajímat ty malé. Nabízí se poměrně jednoduché řešení a to takové, že si stanovíme minimální velikost oblasti, která by

mohla být potenciálně duplikovaná a oblasti, které jsou menší, vyřadíme. Velikost můžeme určitě na základě velikosti obrazu. Ve velkém obrazu budou pravděpodobně duplikovány větší oblasti, než v malém.

Ilustrujme si tento postup na ukázce. Budeme hledat oblast, která byla duplikována pomocí klonovacího razítka v obrázku 2.a. Pomocí výše popsaných postupů nalezneme oblasti podezřelé z duplikace, ty jsou vyznačeny na obrázku 2.b. Odhadem lze určit, že obsah větších oblastí si je nápadně podobný, což nemůžeme říct o dvojici menších oblastí. Zde zřejmě půjde o falešné detekce, které odstraníme morfologicky. Morfologické zpracování spočívá v aplikaci operace uzavření a následné aplikaci otevření se stejným strukturním elementem. Operace uzavření nám všechny detekované oblasti zmenší, přičemž malé a nežádoucí jsou zmenšeny natolik, že zmizí. Následnou aplikací opačné operace, tedy morfologického otevření dostanou zbylé oblasti tvar blížký původnímu tvaru. Bylo by samozřejmě možné dosáhnout i původního tvaru s použitím informací o oblastech před aplikací uzavření a otevření, ale to není nutné, jak si ujasníme později. Před aplikací morfologických operací je třeba ještě definovat strukturní element, se kterým operace provádíme. Se strukturním elementem lze libovolně experimentovat pro dosažení optimálního výsledku, nám se osvědčil kruhový strukturní element o průměru poloviny velikosti masky.



Obrázek 3.: Armádní auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů

Nyní, když jsme odstranili malé, nežádoucí oblasti, provedeme ještě normalizaci podezřelých oblastí tak, abychom si usnadnili další zkoumání v prostorové doméně. Na obrázku 3.a, který jsme získali z 2.b pomocí morfologických operací, jsou již jen dvě podezřelé oblasti. Pro každou oblast vypočítáme tzv. „bounding box“, což je minimální obdélník, do kterého lze danou oblast umístit. Následně přizpůsobíme velikost těchto obdélníků tak, aby byly oba stejně velké. Toto udě-

láme jednoduše tak, že jednu z oblastí posuneme o známý vektor tak, aby překrývala druhou oblast a vypočítáme jejich průnik. Ten pak bude vymezovat oblasti zájmu dalšího zkoumání v prostorové doméně. Výsledné obdélníky pro náš ilustrační příklad jsou na obrázku 3.b.

2.3.2. Koeficienty korelace

V předchozí kapitole jsme si popsali, jak pracovat s vektory posunutí, což byl první z výstupů 2.3.. Druhým z výstupů tohoto algoritmu je mapa korelačních koeficientů, označená `hodnotyKorelace`. Zpracováním mapy korelačních koeficientů se budeme zabývat v této kapitole. Jde o dvourozměrné pole o velikosti shodné s velikostí zkoumaného obrazu. Ne všechny prvky tohoto pole jsou obsazeny v závislosti na zvoleném kroku. Každý obsazený prvek obsahuje koeficient korelace mezi výřezem daným maskou na pozici tohoto prvku a jiným výřezem, který je umístěn na libovolném místě v obraze mimo masku a má nejvyšší koeficient korelace. Řečeno jednodušeji, obsahuje údaj o tom, moc je podobný nejpodobnější výřez ve zkoumaném obraze. Informaci o umístění tohoto nejpodobnějšího výřezu nese příslušný údaj v mapě vektorů posunutí.

Nyní se zaměříme na analýzu těchto koeficientů. Předpokládejme, že v obraze je nějaká duplikovaná oblast. Pak se budou v mapě korelačních koeficientů na příslušných místech vyskytovat velmi vysoké hodnoty. Nabízí se využít jednoduché metody a to prahování pro odhalení oblastí s podezřele vysokou hodnotou korelace. Stanovení globálního prahu může být problematické a málo robustní vzhledem k velkému rozptylu hodnot. Můžeme si ale pomoci výsledkem předchozí kapitoly, kde jsme si vytipovali podezřelé oblasti. Práh pak budeme stanovovat pouze v těchto vytipovaných oblastech.

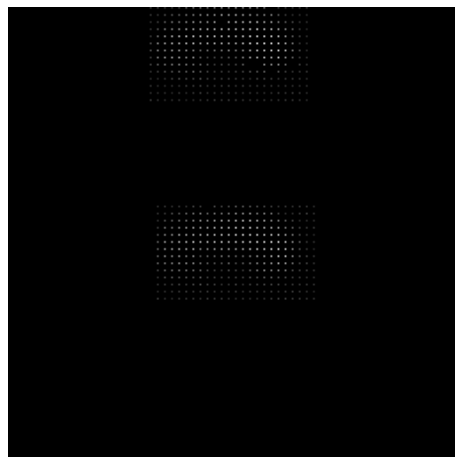
V těchto obdélníkových oblastech můžeme statisticky stanovit práh pro korelační koeficient. Oproti výpočtu na základě informací z celého obrázku zde dovedeme stanovit práh poměrně snadněji. Experimentálně jsme zjistili, že vhodným prahem je střední hodnota. To odpovídá intuitivnímu odhadu založeném na faktu, že oblasti, které jsou skutečně duplikované mají hodnotu korelačního koeficientu velmi vysokou, kdežto ty ostatní ji budou mít relativně k těmto hodnotám velmi malou. Tímto způsobem získáme upřesnění výsledku získaného metodou vektorů posunutí.

Uvedenou metodu si ilustrujme na obrázcích. Ke zkoumání zde máme obrázek 4.a, stejný jako v předchozí kapitole, který obsahuje jednu duplikovanou oblast. Obrázek 4.b znázorňuje korelační koeficienty v oblastech vytipovaných pomocí vektorů posunutí. Úroveň jasu zde odpovídá hodnotě koeficientu korelace, čím světlejší bod, tím vyšší hodnota. Je zde také vidět, že pole hodnot je řídké, tedy že hodnoty se vyskytují pouze na mřížce, to je dáno použitím kroku o velikosti 4 pixely. Na obrázku 4.c jsou již koeficienty po aplikaci prahování, zde nám zbyly pouze ty s hodnotou větší, než byla hodnota prahu. Nyní nám již jen zbývá převést tyto dvě mřížky bodů na souvislé oblasti. To provedeme tak, že na každý

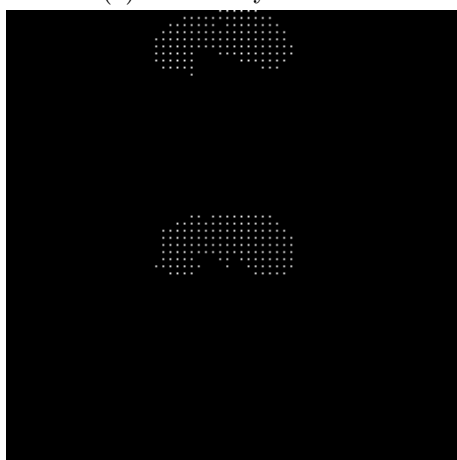
bod mřížky umístíme masku, o stejné velikosti se kterou jsme dosud pracovali. Každý bod mřížky odpovídá jedné pozici masky, resp. pozici jejího levého rohu. Do výsledného obrazu zkopírujeme výřez, který bychom dostali za pomoci této masky z původního obrazu. Na obrázku 4.d pak vidíme výsledek.



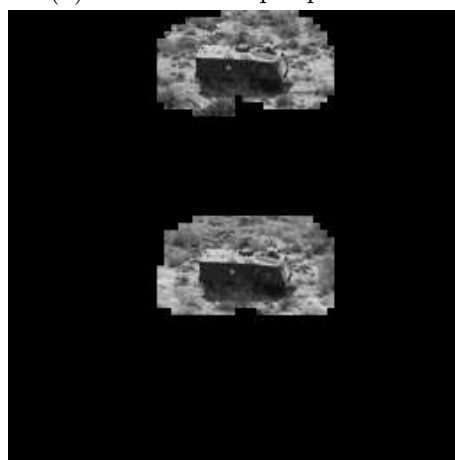
(a) Podvržený obrázek



(b) Určení lokalit pro prahování



(c) Lokálně prahované korelační koeficienty



(d) Určení duplikovaných oblastí

Obrázek 4.: Armádní auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů

2.4. Analýza obrazu v prostorové doméně

Jak jsme si odůvodnili v kapitole 2.2., hledání duplikovaných oblastí v obraze za pomoci výpočtů v prostorové doméně je poměrně výpočetně náročné. V této kapitole tedy využijeme výsledky analýzy ve frekvenční doméně, které se budeme snažit zpřesnit. Máme tedy již dopředu vytipované oblasti, které by mohly být potenciálně duplikované a pouze v těchto oblastech budeme dále aplikovat

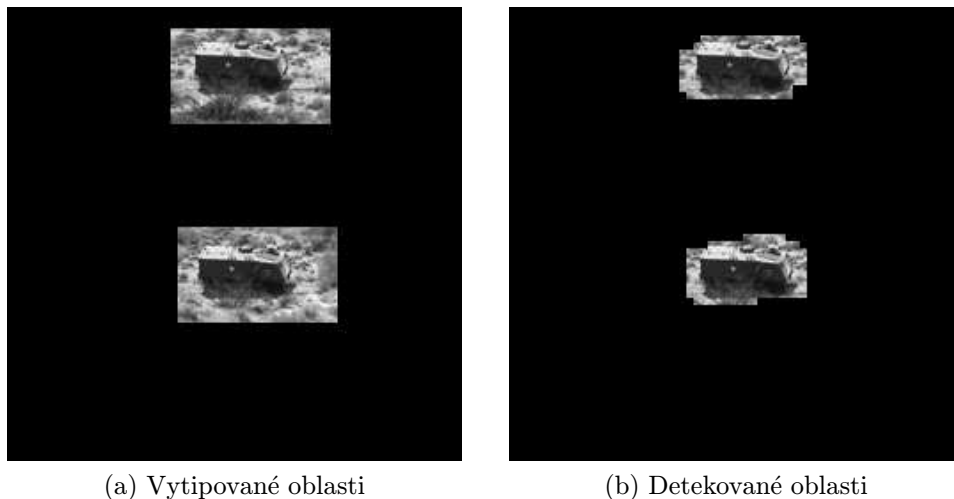
metody pracující v prostorové doméně.

Zpřesňování spočívá například ve zmenšení kroku, o který posouváme masku při pořizování výřezů. Ke snížení počtu operací Fourierovy transformace, která je výpočetně náročná jsme při analýze ve frekvenční doméně posouvali masku o více než 1 pixel. Nyní, vzhledem k tomu, že se oblast dalšího zkoumání výrazně zmenší, můžeme si dovolit posouvat masku právě o 1 pixel v každém kroku.

2.4.1. Použití SSIM

Pro stanovení podobnosti dvou obrazů slouží koeficient SSIM. My jej nepoužijeme pro stanovení podobnosti dvou obrazů, ale postupně budeme počítat podobnost výřezů ze zkoumaného obrazu. Princip, na kterém je SSIM založen a jeho výpočet jsme si popsali v kapitole 2.1.3.

Porovnávání výřezů budeme provádět následovně. Předpokládejme, že máme dvě vytipované oblasti, které jsou podezřelé ze vzájemné duplicity. Jsou to obdélníkové oblasti tak, jak jsme je určili v kapitole 2.3.1.. Zvolíme jednu z nich a tu budeme postupně procházet s použitím masky o rozměrech $n \times n$ pixelů a s krokem 1 pixel. V každém tomto kroku získáme jeden výřez. Druhý výřez pro porovnání získáme tak, že posuneme masku o známý vektor posunutí a pořídíme výřez na této pozici. Pro tyto dva výřezy pak spočítáme index SSIM a uložíme na odpovídající pozici. Velkou výhodou je zde fakt, že již neporovnáváme výřezy „každý s každým“. Pro obdélníkovou oblast o rozměrech $m \times n$ pixelů a velikost masky $b \times b$ pixelů tak budeme potřebovat pouze $(m - b + 1)(n - b + 1)$ výpočtů, pokud budeme procházet s avizovaným krokem 1 pixel.



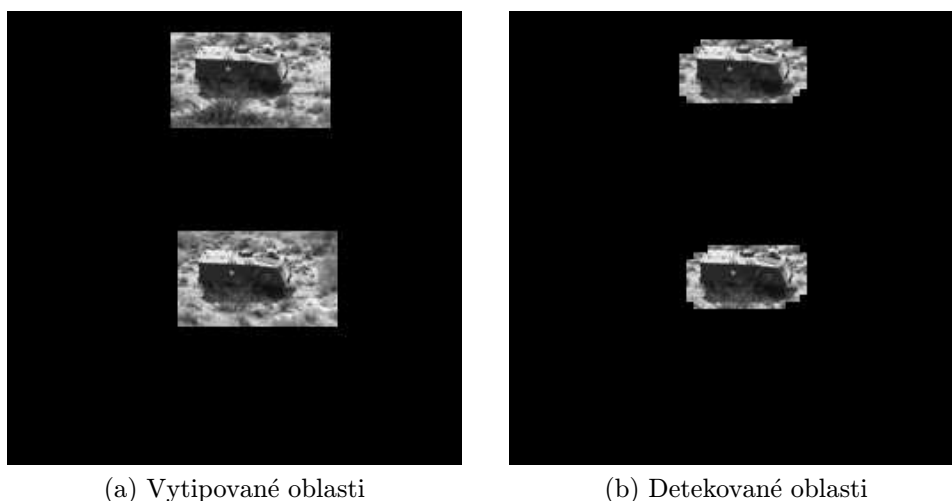
Obrázek 5.: Detekce s použitím SSIM

Podívejme se nyní na ukázkou aplikace této metody. Na obrázku 5. jsme experimentálně stanovili práh $SSIM_{min} = 0.9999$, od této hodnoty nahoru považujeme

oblasti za duplikované. Ač je to poměrně vysoká hodnota, vidíme, že jsme stále detekovali poměrně velké oblasti. Parametry výpočtu: maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů.

2.4.2. Použití MSE

Výpočet střední kvadratické chyby jsme si popsali v kapitole 2.1.2. Tuto metodu jsme si zde uvedli jako výpočetně jednoduchou variantu. Postup procházení je stejný jako u předchozí metody, popsané v kapitole 2.4.1.. Rozdíl je ten, že pro každé dva výřezy, které chceme porovnat, vypočítáme právě MSE. Poté se pomocí vhodného nastavení prahové hodnoty budeme snažit nalézt duplikované oblasti. Poznamenejme, že zde je opačný trend než u SSIM, tedy čím je hodnota nižší, tím jsou si porovnávané výřezy více podobné. Výsledek detekce je podobný jako u předchozí metody 2.4.1.

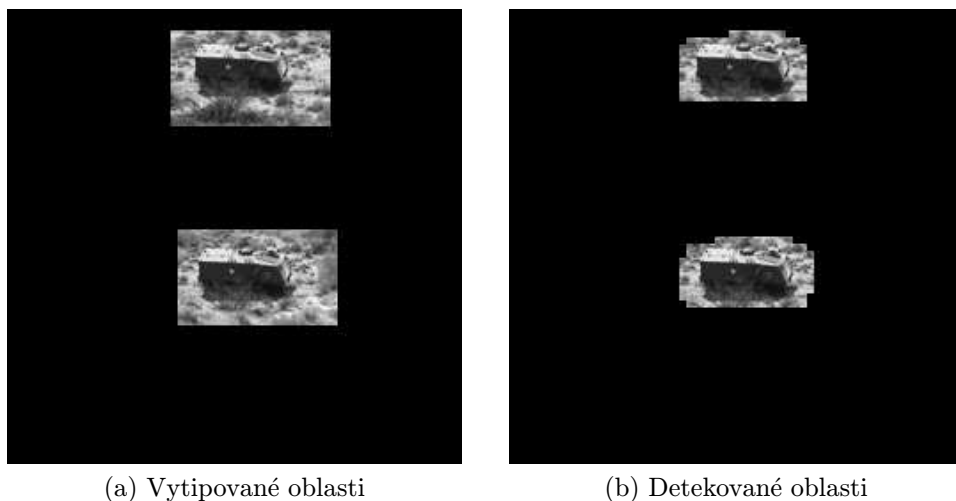


Obrázek 6.: Detekce s použitím MSE

Na obrázku 6. jsme stanovili práh $MSE_{norm} = 0.001$, od této hodnoty hodnoty níže považujeme oblasti za duplikované. Parametry výpočtu: maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů.

2.4.3. Použití korelačního koeficientu

Korelační koeficient, jehož výpočet jsme si popsali v kapitole 2.1.4. použijeme podobně jako index SSIM. Jak jsme se již zmínili, korelační koeficient může nabývat hodnot z intervalu $\langle -1, 1 \rangle$. Při výpočtech s obrazovými výřezy získáváme pro sobě podobné výřezy hodnotu blížíící se 1. Pro výpočet korelačního koeficientu jsme v MATLABu použili funkci `corr2`. Podívejme se na ukázkou.



Obrázek 7.: Detekce s použitím korelačního koeficientu v prostorové doméně

Na obrázku 7. jsme experimentálně stanovili práh korelačního koeficientu $r_{min} = 0.99$, od této hodnoty korelace nahoru považujeme oblasti za duplikované. Přestože i zde máme prahovou hodnotu poměrně vysokou, detekujeme úspěšně duplikované oblasti. Parametry výpočtu: maska 24×24 pixelů, krok 4 pixely, velikost obrázku 256×256 pixelů.

2.5. Předcházení falešným detekcím

V obraze hledáme oblasti, které jsou si do jisté míry podobné, jsou tedy na základě jedné nebo více uvedených metod jako duplikované. Takové oblasti se však mohou v obraze objevit přirozeně a přitom nemusí jít o naklonované oblasti. Uvedme si příklady, kdy k takové situaci může dojít a jak předejít falešným detekcím v případě, že budeme zkoumat obraz, kde se tento příklad vyskytuje.

2.5.1. Plochy s nedostatkem detailů

Uvažujme například fotografii, na které je kus oblohy. Pokud bude obloha bezmračná, máme zde plochu, ve které bychom jistě našli mnoho podobných výřezů a přitom se nemusí jednat o podvrženou fotografii. Abychom předešli falešným detekcím, bude se hodit, pokud takovéto výřezy vyloučíme z dalšího zpracování. Je pravda, že tímto se vystavujeme riziku, že pokud by tyto plochy byly použity k padělení fotografie, dopředu je zamítáme a padělek neodhalíme. Na druhou stranu, pokud máme velké množství falešných detekcí a pokud by se mezi nimi vyskytla oblast označená právem za duplikovanou, těžko s jistotou určíme, která z nich to je. Připustíme tedy malý nedostatek našeho dalšího zkoumání a to ten, že pokud bude provedena duplikace s použitím oblasti, jež trpí absencí detailů, nenajdeme ji.

Popišme si jeden z možných přístupů, jak vyloučení zmíněných ploch provést. Můžeme použít podobného principu, jako v kapitolách předchozích a to procházení obrazu s pomocí masky o určité velikosti. Pro každý výřez daný masku spočítáme rozptyl hodnot jasu, dle vztahu 18 Na základě hodnoty rozptylu jasu lze rozhodnout, zda budeme tento blok používat. Otázkou je, jak stanovit práh rozptylu, při kterém již nebudeme chtít blok dále zkoumat. Pokud připustíme jistou interaktivitu uživatele během výpočtu, lze provést přizpůsobení ručně.

$$\text{var}(B) = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (B[i, j] - E(B))^2 \quad (18)$$

kde

- B je soubor hodnot jasu pro výřez o rozměrech $m \times n$ pixelů
- $E(B)$ je střední hodnota pro soubor hodnot jasu B

Ukázka této problematiky je na obrázku 8. Zde lze za oblast bez detailů považovat modrou oblohu. Na obrázek jsme aplikovali prahování s využitím hodnoty rozptylu ve výřezech o velikosti 16×16 pixelů. Plochy vyznačené černě zůstávají pro další výpočty, plochy označené bíle jsme vyloučili pro nedostatek detailů.



(a) Vzorový obrázek



(b) Vyloučené plochy

Obrázek 8.: Vyloučení ploch s nedostatkem detailů, maska 16×16 pixelů

2.5.2. Opakující se vzory

Nyní uvažujme obraz, kde se určitá struktura přirozeně opakuje. Zde jistě můžeme pořídit výřezy, které vyhodnotíme jako velmi podobné, avšak nebude se jednat o podvrh. Pro lepší představu si ukažme hned na začátek ilustrační obrázek. Na obrázku 9. jsou červenými obdélníky vyznačeny oblasti, o kterých se

zmiňujeme. Pokud bychom vzali výřez z některé z červeně vyznačených oblastí a dále maskou pohybovali po hraně střechy, jistě pořídíme velké množství výřezů, které budou podobné.



Obrázek 9.: Obrázek s opakující se strukturou

Aniž bychom tento fakt zdůraznili, metoda využívající vektorů posunutí, kterou jsme si popsali v kapitole 2.3.1. je vůči tomuto problému poměrně odolná. Zde totiž pomocí morfologických operací eliminujeme malé detekované oblasti, které považujeme za nezajímavé. Právě takové oblasti vznikají u zmíněného typu fotografií. Pokud tedy budeme při aplikaci ostatních metod vycházet z metody využívající vektorů posunutí, což z důvodu výpočetní náročnosti děláme, s tímto problémem se příliš nebudeme potýkat.

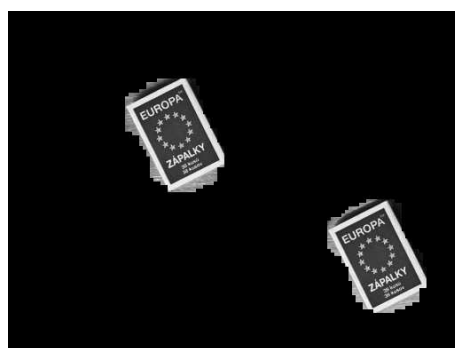
2.6. Ukázky

Na závěr kapitoly o detekci fotomontáže, která byla vytvořena prostřednictvím duplikace části obrazu si uvedeme ukázky více podvržených obrázků. Ty jsme vytvořili pomocí nástroje Photoshop. V každé z níže uvedených čtveřic obrázků je první vždy podvržený obrázek, u kterého jsme duplikovali nějakou jeho část. Dále jsou zde výsledky detekce pomocí korelačního koeficientu, SSIM a MSE tak, jak jsme si je popsali v příslušných kapitolách. V popisu obrázku jsou pak uvedeny parametry zkoumaného obrazu a parametry algoritmu, kterým jsme detekci prováděli. O nastavení prahových hodnot pro koeficienty podobnosti se zmíníme jednotlivě, jelikož je nutno je nastavovat pro každý obraz individuálně.

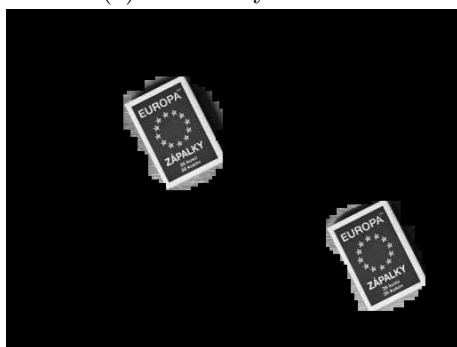
Na první sadě obrázků 10.a – 10.d jsme využili vlastní fotografii. Zde jsme duplikovali krabičku zápalek. Před výpočtem byl obrázek převeden na šedotónový. Výsledky jednotlivých metod jsou si podobné. Duplikovaná oblast byla úspěšně detekována. Prahové hodnoty pro jednotlivé koeficienty podobnosti jsou uvedeny v tabulce 1.



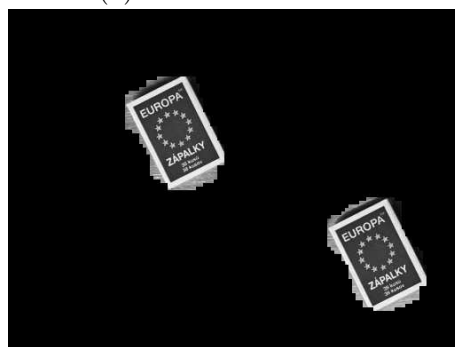
(a) Podvržený obrázek



(b) Korelační koeficient



(c) SSIM

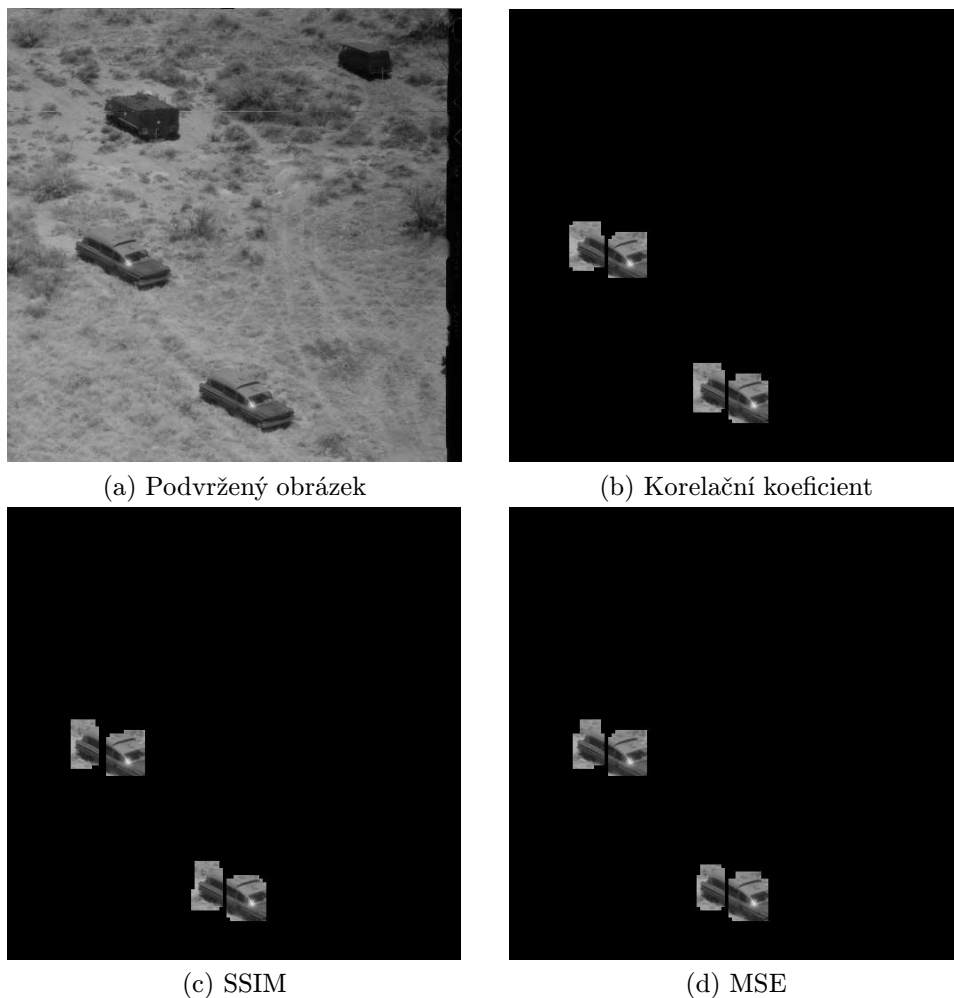


(d) MSE

Obrázek 10.: Zápalky - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 512×512 pixelů

V druhé sadě obrázků 11.a – 11.d jsme využili obraz z databáze [19] Zde jsme duplikovali automobil v dolní levé části obrazu. I zde jsou výsledky jednotlivých metod velice podobné a mají jeden nedostatek – nenalezly oblast souvislou, ale

dvě oddělené oblasti. Podstatnější ale je, že detekce proběhla úspěšně. Prahové hodnoty pro jednotlivé koeficienty podobnosti jsou uvedeny v tabulce 1..



Obrázek 11.: Auta - maska 24×24 pixelů, krok 4 pixely, velikost obrázku 512×512 pixelů

Koeficient / Obrázek	Auta	Zápalky
Korelační koeficient	0,99	0,90
SSIM	0,9999	0,99
MSE	0.01	0.02

Tabulka 1.: Prahové hodnoty koeficientů podobnosti

2.7. Optimalizace

Jelikož celý proces hledání duplikovaných oblastí je poměrně výpočetně náročný, je vhodné se zaměřit na optimalizaci celého výpočtu. Podstatné přitom bude omezení počtu diskretních Fourierových transformací a inverzních diskretních Fourierových transformací, což jsou v našem algoritmu 1 nejnáročnější operace.

První krok v optimalizaci bylo již vyloučení některých bloků popsané v kapitole 2.5.1. Tento dílčí výpočet sice zabere určitý čas, ale vzhledem k celkové časové náročnosti celého procesu je výhodné jej provést, jelikož po dokončení tohoto kroku vyloučíme jistou část bloků, pro kterou další náročnější výpočty nebudeme provádět.

2.7.1. Blokové operace

Jelikož výpočet v MATLABu je optimalizován na operace s bloky dat, je vhodné toto zohlednit ve zdrojovém kódu. Mírného zrychlení výpočtu jsme dosáhli vypuštěním funkce `imcrop`, která slouží k získání výřezu z obrazu. Jelikož pracujeme s obrazem jako s velkým dvourozměrným číselným polem, můžeme použít příslušný operátor pro výběr části pole. Ve zdrojovém kódu tedy nahradíme řádek

```
region = imcrop(f,[x y mask_size-1 mask_size-1]);
```

za řádek

```
region = f(y:y+mask_size-1,x:x+mask_size-1);
```

Provedli jsme jednoduchý test, jak velké je zrychlení při výpočtu. Výsledky jsou obsaženy v následující tabulce.

Velikost [px]	1024x1024	512x512	256x256
Pomocí <code>imcrop</code> [s]	133,10	31,75	6,78
Pomocí operátoru [s]	8,99	2,02	0,46
Zrychlení	14,80x	15,71x	14,74x

Tabulka 2.: Zrychlení operace výřezu

Z tabulky je vidět, že zrychlení je nezávisle na velikosti obrazu asi patnáctinásobné. Rovněž vidíme úsporu času, která je znát zejména u obrazu větších rozměrů.

2.7.2. Optimalizace Fourierovy transformace

Při výpočtech, které jsou popsány v kapitolách 2.3.2. a 2.3.1. provádíme mimo ostatních operací pro každý výřez z obrazu diskretní Fourierovu transformaci a

poté inverzní diskrétní Fourierovu transformaci. V MATLABu se tyto operace provádí pomocí funkcí `fft2` a `ifft2`. Dle popisu těchto funkcí v [18] lze dosáhnout markantního zrychlení ve speciálních případech. Je zde uvedeno, že nejrychleji se výpočet provede, pokud je velikost transformovaného obrazu mocninou dvojky. Dále je také rychlost výrazně vyšší pro rozměry, které mají v prvočíselném rozkladu malé prvky. Těchto vlastností lze využít tak, že pokud obdržíme na vstupu obraz o nevhodných rozměrech, provedeme doplnění na přijatelnější velikost.

2.7.3. Procházení s větším krokem

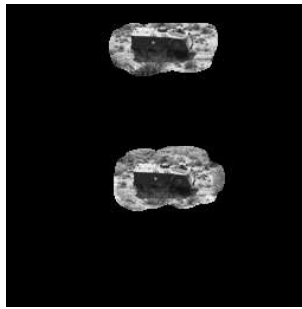
Dalším postupem při optimalizaci může být zvětšení kroku, se kterým posouváme masku. Pokud zvětšíme krok dvakrát na obou osách, pak obraz, který zkoumáme může mít čtyřikrát větší plochu při zachování přibližně stejného počtu volání `fft2` a `ifft2`. Krok bychom měli volit v závislosti na velikosti obrazu a také v závislosti na tom, jaká je předpokládaná velikost duplikované oblasti.

Provedme experiment, který ilustrujeme na obrázku 12. Využijeme metodu popsanou v kapitole 2.3.1. Budeme postupně zvětšovat velikost proměnné `krok`, neboli velikost posunu masky v pixelech. Přitom budeme sledovat zrychlení celého výpočtu (čas t) a zároveň hodnotit zda je pro nás výsledek přijatelný.

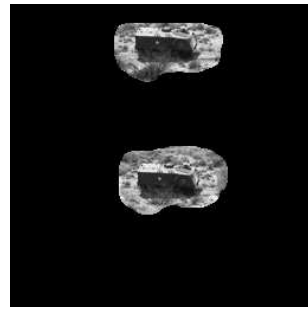
Velikost duplikované oblasti je cca 80×40 pixelů. Vidíme, že i když postupně zvětšujeme krok posunu masky, dostáváme jsme přibližně stejný výsledek ale v mnohem kratším čase. Výrazná změna nastává až v posledním uvedeném případě, kdy jsou rovněž identifikovány duplikované oblasti (bloky jsou stejně obarveny), ale již se zde objevuje velký počet falešných detekcí, což je nežádoucí. Navíc zlepšení časové složitosti zde již není tak markantní.



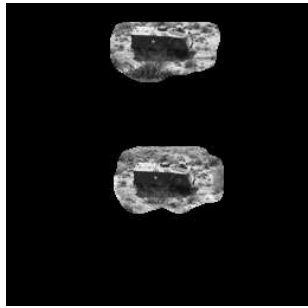
(a) Zkoumaný obrázek



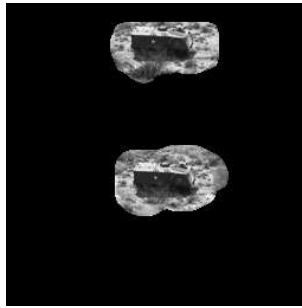
(b) krok=1px t=2371s



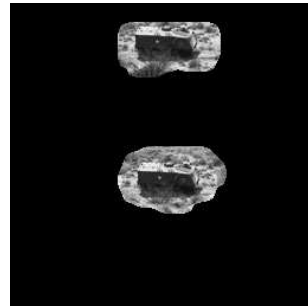
(c) krok=2px t=751s



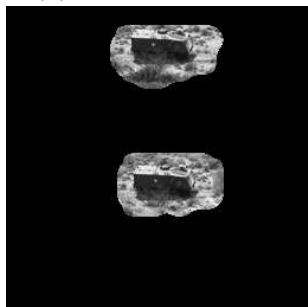
(d) krok=3px t=322s



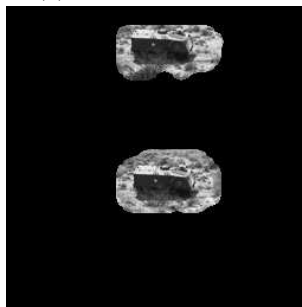
(e) krok=4px t=191s



(f) krok=5px t=114s



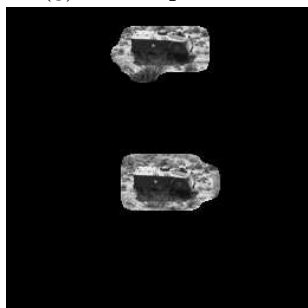
(g) krok=6px t=88s



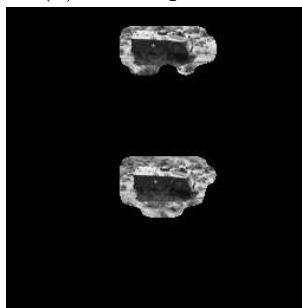
(h) krok=7px t=62s



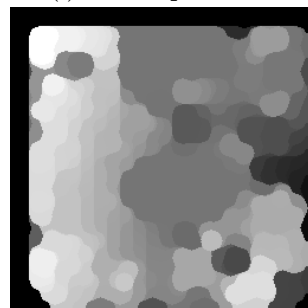
(i) krok=8px t=48s



(j) krok=9px t=40s



(k) krok=10px t=30s



(l) krok=11px t=25s

Obrázek 12.: Armádní auta - maska 32×32 pixelů, velikost obrázku 256×256 pixelů

3. Metody zkoumající inkonzistenci šumu

V této kapitole si uvedeme metody, které využívají pro detekci fotomontáže zkoumání struktury šumu. Nejprve si uvedeme, co je vlastně šum v digitálních fotografiích, jak vzniká a jaké má charakteristiky. Poté se budeme věnovat metodám odhadu šumu, což nám poslouží k dalšímu zkoumání, konkrétně zkoumání inkonzistencí.

3.1. Šum v digitálním obraze

Šum v digitálním obraze jsou náhodné změny hodnot jasu jednotlivých pixelů. Pojmenování pochází z analogie zvukového záznamu, kde vyvstává podobný problém a výsledkem je slyšitelný šum. Šum je samozřejmě nechtěný jev, který je s modernizací fotografické techniky stále více a lépe potlačován. Objasněme si, jaké typy šumu se v digitálním obraze vyskytují a jak vznikají.

3.1.1. Typy digitálního šumu

Do procesu vzniku digitální fotografie zasahuje mnoho jevů, které jsou zdrojem šumu ve výsledném obraze. Všimněme si významných zdrojů šumu a přibližme si jejich charakteristiky. [12],[14]

Prvním typem šumu, který vstupuje do zpracovávaného signálu je tzv. **dark current noise**. Je způsoben nedokonalostí fotodiody, která je základním konstrukčním prvkem CCD senzoru. Ta propouští malý proud i v případě, že není osvětlena, tedy v situaci, kdy by měl být proud nulový. Tento typ šumu je přímo úměrný teplotě a délce expozice.

Dalším typem šumu je tzv. **fixed pattern noise**. Je zapříčiněn tím, že každý pixel má mírně odlišnou citlivost na světlo. Ve výsledném obraze se mohou objevovat světlé a tmavé pixely. Projevuje se zejména při delších expozičních časech, kdy se na snímku objeví body s vyšší intenzitou jasu, než by měla být.

Shot noise vzniká, pokud na světlocitlivé součástce, vzniknou měřitelné fluktuace napětí zapříčiněné kolísáním počtu fotonů, které dopadly na tuto součástku. Vzhledem k tomu, že ke kolísání počtu fotonů dochází vždy, projeví se fluktuace napětí pouze při malém počtu celkově zachycených fotonů. Při velkém počtu fotonů již přestanou být odchylky měřitelné. Z toho vyplývá, že shot noise se projeví zejména při slabém osvětlení, kdy se snažíme měřit malé hodnoty napětí odpovídající nízkému počtu zachycených fotonů.

Napětí na fotocitlivé součástce musí být před měřením a kvantizací zesíleno, zde dochází ke vzniku tzv. **amplifier noise**. Standardně je tento šum charakterizován jako aditivní gaussovský šum, nekorelovaný se signálem. Je zapříčiněn Johnson–Nyquistovým šumem, který vzniká v elektronickém obvodu zesilovače.

Kvantizační šum vzniká při převodu analogového signálu na digitální, tzv.

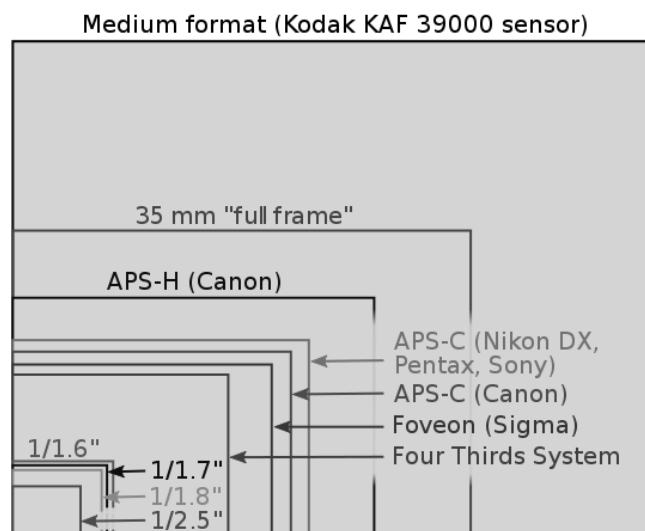
kvantizaci. Při převodu na diskrétní stupnici dochází k mírnému zkreslení skutečně zachycených hodnot.

První dva popsané typy šumu lze ve fotografickém přístroji redukovat pomocí tzv. flat-field correction [15], kdy se pořídí korekční snímky na základě kterých je šum z výsledných snímků odstraněn. Pro další zkoumání se jeví jako vhodní kandidáti shot noise a amplifier noise, kterými se budeme zabývat dále.

3.1.2. Vlastnosti digitálního šumu

Na základě vlastností digitálního šumu popsaných výše můžeme usuzovat, za jakých okolností se bude intenzita šumu měnit. Je zřejmé, že jeho velikost se bude zvyšovat se zvyšující se teplotou. Ta má vliv jak na snímač, tak na zesilovač, který produkuje šum.

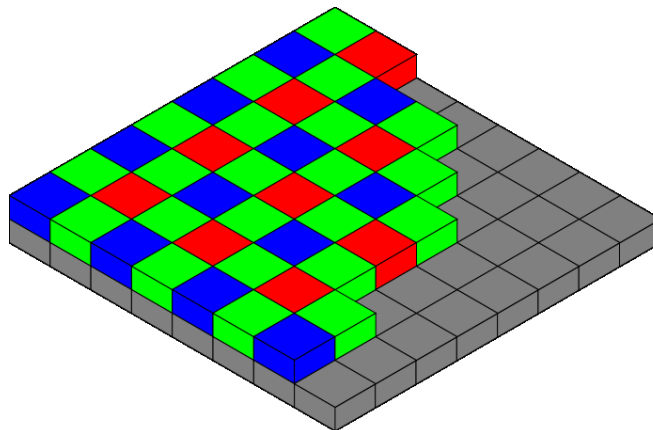
Z principu shot noise je zřejmé, že čím méně fotonů zachytíme, tím více se tento šum projeví. Počet zachycených fotonů je závislý na intenzitě osvětlení a na velikosti plochy, na kterou fotony zachycujeme. Hladina šumu tedy bude nepřímo úměrná intenzitě dopadajícího světla. Co se týče plochy, zde zjistíme významné rozdíly mezi jednotlivými typy fotoaparátů a také mezi barevnými kanály. Velikost snímače se pohybuje od nejmenších, které jsou instalovány v mobilních telefonech, přes větší, které používají kompaktní fotoaparáty až po kvalitní tzv. full frame velké snímače instalované do zrcadlovek.



Obrázek 13.: Porovnání velikostí obrazových snímačů, převzato z [29]

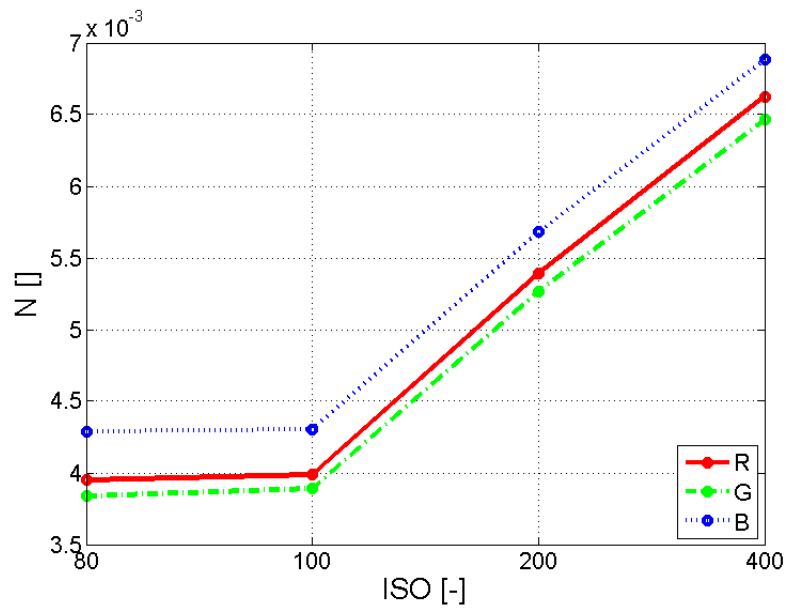
Na obrázku 13. jsou znázorněny velikosti běžně používaných formátů obrazových snímačů. Na obrázku 17. vidíme rozdíl při použití dvou odlišných velikostí snímače. Testovali jsme fotoaparát Panasonic DMC-FZ77, který má snímač o velikosti 5.75mm × 4.31mm (na obrázku 13. je to ten s nejmenší velikostí,

tedy 1/2.5"). Pro porovnání jsme pořídili snímek stejné scény při stejném nastavení ISO, ale digitální zrcadlovkou Canon EOS 400D, ve které je instalován senzor o velikosti 22.2mm × 14.8mm (na obrázku 13. je to ten s označením APS-C Canon). Rozdíl je jasně viditelný, při použití menšího senzoru je ve snímku obsaženo více šumu.



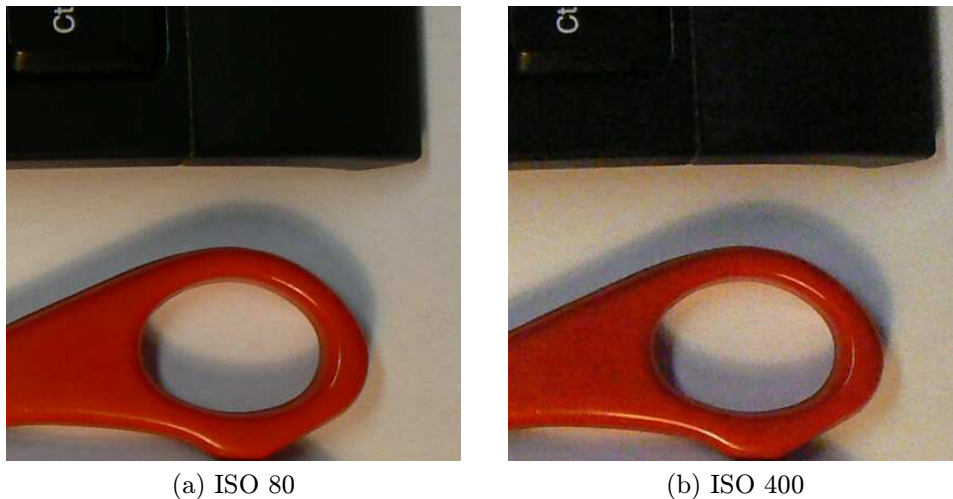
Obrázek 14.: Bayerova maska, převzato z [28]

Rozdíly napříč barevnými kanály jsou důsledkem konstrukce snímače, kde je použita tzv. Bayerova maska, znázorněná na obrázku 14. – pro snímání zelené barevné složky slouží dvakrát více bodů než pro červenou a modrou složku. Tím máme pro zelenou složku dvakrát větší snímací plochu, což znamená menší šum. Na obrázku 15. vidíme srovnání hladin šumu pro jednotlivé barevné složky. Zde jsme pořídili 4 snímky stejné scény fotoaparátem Panasonic DMC-FZ7 při použití čtyř hodnot ISO, které lze nastavit a to 80,100,200 a 400. Je patrné nejen, že šum roste s vyšší hodnotou ISO, ale také že nejvíce se šum projevuje v modré složce, naopak v zelené je šumu nejméně. Během experimentu jsme použili metodu 3.2.1. pro odhad šumu a metodu 3.3.2. pro měření úrovně šumu.



Obrázek 15.: Srovnání úrovně šumu pro jednotlivé barevné kanály

Hladina šumu je rovněž závislá na nastavené citlivosti snímače, ta je vyjadřována parametrem ISO. Moderní fotoaparáty umožňují nastavovat poměrně vysoké hodnoty, přičemž u vyšších hodnot je již šum ve fotografii jasně zřetelný. Čím vyšší citlivost nastavíme, tím méně fotonů (menší intenzitu světla) potřebujeme pro požadovanou expozici. Tím pádem se více projeví fluktuace popsané u šumu typu shot noise. Na obrázku 16. je jasně viditelný šum při použití vyššího ISO, rovněž z obrázku 15. je zřejmý nárůst šumu při zvyšování ISO.



Obrázek 16.: Porovnání výřezů obrázků pořízených s různým nastavením ISO, fotoaparát Panasonic DMC-FZ7



(a) Canon EOS 400D, velikost senzoru APS-C (22.2×14.8 mm)



(b) Panasonic DMC-FZ7, velikost senzoru 1/2.5" (5.75×4.31 mm)

Obrázek 17.: Porovnání výřezů obrázků pořízených fotoaparáty s různou velikostí senzoru při stejném nastavení ISO 400

Na základě uvedených tvrzení, která jsou známa z praxe a která jsme si ilustrovali jednoduchým experimentem můžeme usoudit, že pokud chceme detekovat fotomontáž na základě inkonzistencí šumu, budeme tohoto schopni zejména pokud byly použity části fotografie pořízené fotoaparátem s různými velikostmi snímače, při různých hodnotách ISO nebo při vhodné kombinaci těchto rozdílů. Vhodnou kombinací máme na mysli takovou, kdy se hodnoty úrovně šumu rozcházejí. Nevhodnou kombinací by mohlo být například: jedna část je pořízena velkým snímačem s velkou hodnotou ISO, druhá část je pořízena malým snímačem s malou hodnotou ISO. Hladiny šumu se pak mohou rozcházet minimálně.

3.1.3. Model šumu

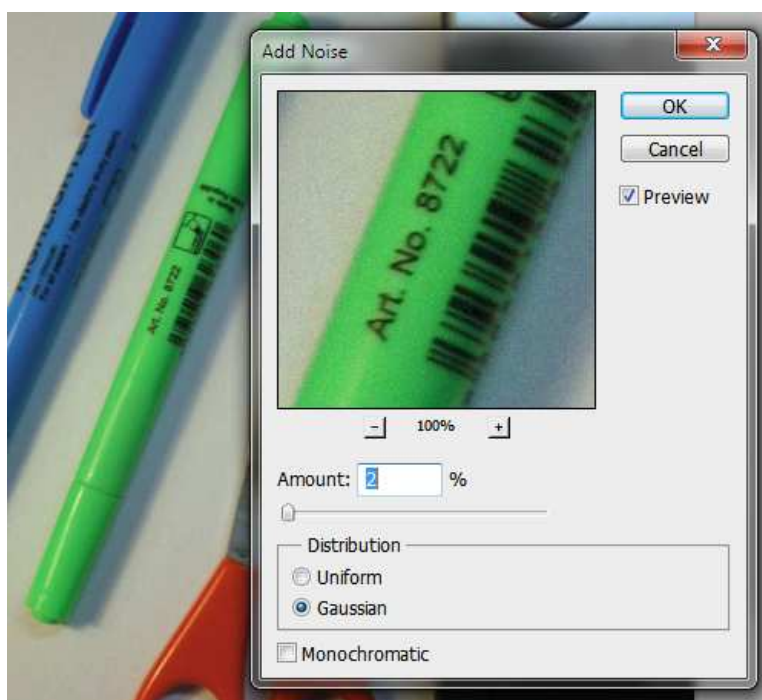
Pro zkoumání případné inkonzistence šumu budeme nejdříve potřebovat nějakým způsobem odhadnout šum v obraze. V kapitole 3.1.1. jsme si uvedli několik zdrojů šumu. Některé se projeví více (shot noise, amplifier noise), některé méně, přičemž každý má odlišné vlastnosti. Výsledkem této kombinace bude šum, který bude vhodné charakterizovat pomocí jistého modelu. Běžně používaným modelem pro šum v digitálním obraze je aditivní bílý gaussovský šum. Tento model je podrobně popsán v [1] a [32]. Šum je způsoben náhodnými fluktuacemi intenzity signálu, u kterých předpokládáme:

- mají normální rozdělení
- mají nulovou střední hodnotu
- nejsou korelovány s intenzitou signálu

Výhodnou vlastností tohoto modelu je aditivnost. Vzhledem k této vlastnosti můžeme odhad šumu získat z obrazu tak, že na něj aplikujeme některý ze známých algoritmů pro redukci šumu a výsledek odečteme od původního obrazu, čímž získáme odhad šumu. Metod používaných na redukci šumu je poměrně mnoho, my se zaměříme na tři z nich, jimž se budeme podrobněji věnovat v kapitolách 3.2.1., 3.2.2. a 3.2.3.

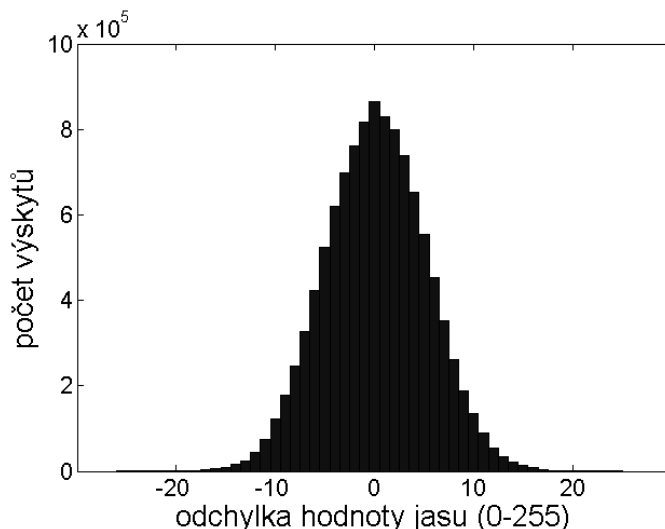
3.1.4. Uměle přidaný šum

Pro hodnocení metod, které si uvedeme dále a jako ilustrační příklad se nám bude hodit referenční obraz, který bude obsahovat šum přesně odpovídající modelu popsanému v kapitole 3.1.3. Ten lze ve Photoshopu přidat pomocí jednoho z filtrů. Použití tohoto filtru ilustrujeme na obrázku 18. V modifikovaném obraze budeme hledat oblast, kam jsme šum přidali aniž bychom použili informaci o poloze a velikosti této oblasti.



Obrázek 18.: Lokální přidání šumu do obrazu v programu Photoshop

Pro ověření charakteristik provedeme globální aplikaci šumu, přičemž od výsledku odečteme původní obraz a získáme tak pouze šum. Zde můžeme provést analýzu, abychom si ověřili, že šum skutečně má parametry, které jsme si pro náš model uvedli. Na obrázku 19. vidíme histogram pro přidaný šum. Lze odhadnout, že histogram má normální rozdělení a že střední hodnota bude blízká nule.



Obrázek 19.: Histogram aditivního bílého gaussovského šumu

3.2. Odhad šumu

Na rozdíl od případu, který jsme si popsali v kapitole 3.1.4., při zpracování obrazů se skutečným šumem způsobeným nedokonalostmi fotografického zařízení nebudeme schopni získat identický obraz, kde se šum nevyskytuje. Zde budeme pracovat pouze s odhadem. Pro získání odhadu šumu nám poslouží několik metod, které si popíšeme dále. V kapitole 3.2.1. použijeme diskrétní waveletové transformace, v kapitolách 3.2.3. a 3.2.2. pak využijeme filtry pro redukci šumu.

3.2.1. Diskrétní waveletová transformace

Diskrétní waveletová transformace [7], označovaná také jako DWT⁵, nalézá vedle širokého využití i uplatnění při potlačování šumu v obrazu. Proces transformace je dekompozicí obrazu, kdy z původního obrazu vzniká sada aproximačních a detailních koeficientů. Opačný postup se nazývá rekonstrukce, kdy jsme schopni z těchto koeficientů zpět sestavit původní obraz. Potlačení šumu bývá realizováno prahováním detailních koeficientů.

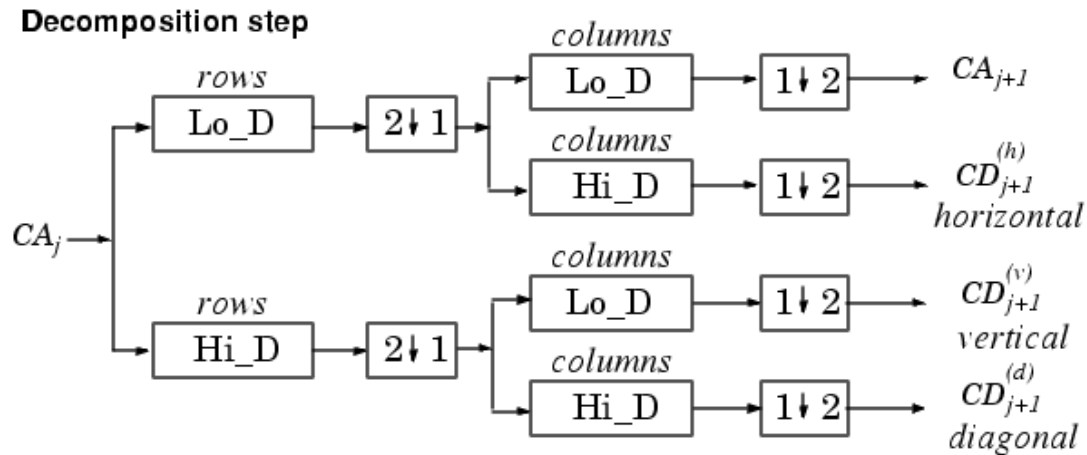
Poznamenejme předem, že nebudeme provádět prahování detailních koeficientů a následnou zpětnou transformaci, protože pro náš účel nepotřebujeme obraz s redukovaným šumem. Naším cílem je najít inkonzistence v šumu, k dalšímu zkoumání nám tedy postačí zmíněné detailní koeficienty.

Důležitou technikou, kterou budeme potřebovat při zkoumání obrazu na různých úrovních rozlišení je subband kódování [1]. Při tomto kódování je obraz rozdělen na několik komponent s omezeným frekvenčním rozsahem, které se na-

⁵Discrete Wavelet Transformation

zývají subbands. Z těchto komponent, pak může být obraz zpětně bezeztrátově zrekonstruován.

Na obrázku 20. je ilustrován jeden krok algoritmu dvourozměrné diskrétní waveletové dekompozice. Na jeho vstupu jsou aproximační koeficienty pro úroveň j (při startu algoritmu je to originální vstupní obraz) a na výstupu jsou pak 4 komponenty: aproximace pro úroveň $j + 1$ a koeficienty detailů ve třech směrech: horizontálním, vertikálním a diagonálním. Každá z těchto komponent má poloviční rozměry než původní obraz. Toto zmenšení je provedeno na základě Shannonova teorému [1], jelikož jsme pomocí filtru odstranili polovinu frekvencí, lze odstranit i polovinu vzorků.



Obrázek 20.: Princip 2D DWT, převzato z [17]

Vysvětleme si činnost tohoto algoritmu. Využita je dvojice filtrů - dolnopro-pustní Lo_D a hornopropustní Hi_D . Tyto dva filtry jsou vzájemně komplementární, tzv. kvadraturně zrcadlové a umožňují přesnou rekonstrukci signálu.[30],[1] Oba tyto filtry aplikujeme na řádky obrazu. U filtrovaných obrazů poté provedeme podvzorkování sloupců (operace $2 \downarrow 1$) a to tak, že vynecháme každý druhý sloupec. Nyní máme dvojici obrazů, jeden po průchodu filtrem Lo_D , druhý po průchodu filtrem Hi_D , přičemž oba mají poloviční počet sloupců než původní obraz. Na každý z nich aplikujeme nyní podobný postup s tím rozdílem, že zaměníme řádky a sloupce. Provedeme filtrování dvojicí filtrů Lo_D a Hi_D , tentokrát na sloupcích a poté podvzorkování, tentokrát na řádcích (operace $1 \downarrow 2$). Výstupem jsou 4 obrazy o polovičních velikostech stran.

Nás bude zajímat výsledná komponenta prvního stupně dekompozice v obrázku označená jako $cD_{j+1}^{(d)}$, která obsahuje výstupy z hornopropustních filtrů, tedy koeficienty detailů v obou směrech. Předpokládáme, že tento bude nejvíce vystihovat obsažený šum. O tom se snadno přesvědčíme z obrázků 35.b–35.d které představují právě tuto komponentu (každá pro jeden ze tří barevných kanálů). Jsou výsledkem aplikace diskrétní waveletové transformace na testovací obrázek

popsaný v kapitole 3.1.4. Na všech třech je zřejmá elipsovité oblast uprostřed, kam jsme lokálně přidali gaussovský šum.

3.2.2. Gaussův filtr

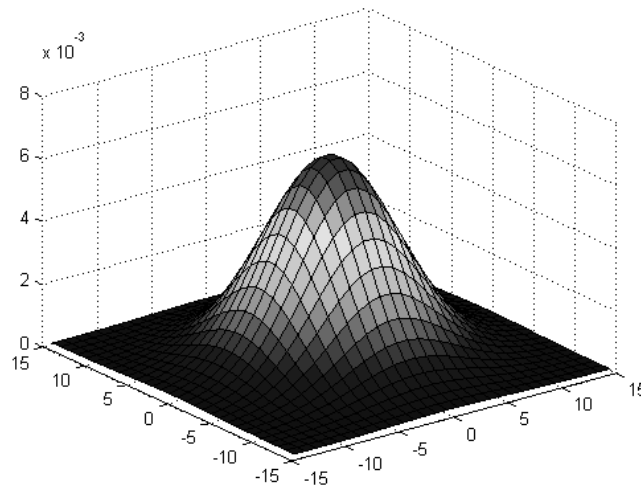
Gaussův filtr je lineární filtr, který je vhodný k potlačení gaussovského šumu. V našem případě se jedná o dolnoproputní filtr, který eliminuje vysoké frekvence, tedy šum, ale i jemné detaily. Způsobí rozmazání obrazu. Impulzní odezva filtru je Gaussova funkce, která je dána vztahem

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\bar{x})^2+(y-\bar{y})^2}{2\sigma^2}} \quad (19)$$

kde

- x je vzdálenost od středu jádra v horizontálním směru.
- y je vzdálenost od středu jádra ve vertikálním směru.
- \bar{x}, \bar{y} jsou střední hodnoty
- σ je směrodatná odchylka.

Z této funkce odvodíme koeficienty konvolučního jádra, se kterým pak aplikujeme konvoluci na obraz, který chceme filtrovat. Hodnoty \bar{x}, \bar{y} budou v našem případě rovny nule. Na ilustračním obrázku 21. je velikost konvolučního jádra 30×30 pixelů a směrodatná odchylka $\sigma = 5$. Při naší aplikaci použijeme menší konvoluční jádro. Funkční hodnoty v tomto grafu vyjadřují váhy, pomocí kterých průměrujeme hodnoty okolních pixelů. Součet těchto vah je roven 1.



Obrázek 21.: Konvoluční jádro pro Gaussův filtr 30×30 px, $\sigma = 5$

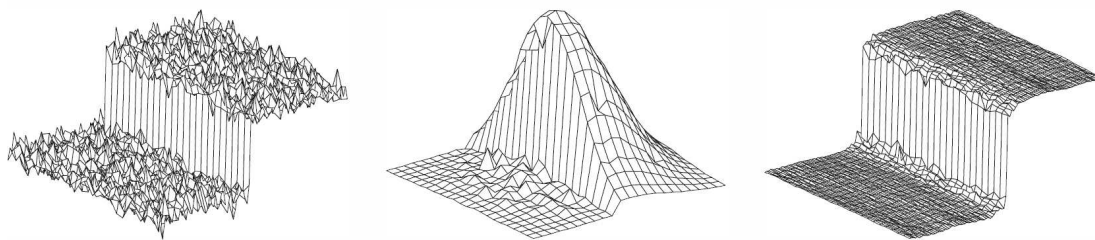
Ukažme si použití filtru v praxi, podobně jako v předchozí kapitole. Použijeme opět náš testovací obraz s přidaným Gaussovým šumem, který jsme si zavedli v kapitole 3.1.4. Na ten aplikujeme Gaussov filtr, jehož konvoluční jádro má rozměry 3×3 body a směrodatnou odchylku $\sigma = 0,5$, což jsou standardní hodnoty Gaussova filtru v Matlabu. Konvoluční jádro je číselně vyjádřeno maticí 20.

$$\begin{pmatrix} 0,0113 & 0,0838 & 0,0113 \\ 0,083 & 0,6193 & 0,0838 \\ 0,0113 & 0,0838 & 0,0113 \end{pmatrix} \quad (20)$$

Aplikací filtru získáme vyhlazený obraz, který odečteme od původního obrazu a získáme tak odhad šumu. Vyhlazováním jsme kromě redukce šumu dosáhli i odstranění jemných detailů a rozostření hran. Na obrázcích 36.b–36.d jsou odhady šumu pro jednotlivé barevné kanály. Kromě oblasti s přidaným šumem jsou zde zřetelné rovněž hrany objektů.

3.2.3. Bilaterální filtr

Bilaterální filtr [13] je filtr, který rozmazává obraz a redukuje šum, přičemž ale zachovává ostrost hran. V tom je rozdíl oproti Gaussově filtru popsanému v předchozí kapitole. Z Gaussova filtru vyjdeme i při popisu bilaterálního filtru, jelikož mají mnoho společného. U bilaterálního filtru je rovněž jasová hodnota každého pixelu nahrazena váženým průměrem jasových hodnot okolních pixelů. Váhy jsou přitom založeny na Gaussově funkci, kterou je definována hustota pravděpodobnosti normálního rozdělení. Čím je pixel vzdálenější od nahrazovaného pixelu (středu konvolučního jádra), tím je jeho váha menší. Podstatný rozdíl je, že váhy nezávisí pouze na eukleidovské vzdálenosti od středu jádra, ale také na fotometrické podobnosti s okolím. Filtr zohledňuje, zda nepracuje v oblasti výskytu nějaké hrany.



Obrázek 22.: Princip bilaterálního filtru, převzato z [13].

Na obrázku 22. je ilustrován princip bilaterálního filtru. Vstupem je zašumělý signál, který obsahuje velký skok ve funkčních hodnotách, ten představuje hranu

v obraze. Na prostředním obrázku je filtr, klasická Gaussova funkce která je však modifikována v místě odpovídajícímu hraně. Funkční hodnoty, tedy váhy jsou přizpůsobeny intenzitě vstupního signálu tak, aby do průměru byly s vyšší váhou započítány pixely s podobnou hodnotou jasu. Výsledkem je signál na obrázku vpravo, kde je šum eliminován, hrana je však zachována. Kdyby byl použit klasický Gaussův filtr, nebyla by hrana ve výsledku tak strmá.

Matematicky lze filtr vyjádřit následujícím vztahem:

$$k(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, \mathbf{x}) s(\xi, \mathbf{x}) d\xi \quad (21)$$

$$c(\xi, \mathbf{x}) = e^{-\frac{1}{2} \left(\frac{d(\xi, \mathbf{x})}{\sigma_d} \right)^2}$$

$$s(\xi, \mathbf{x}) = e^{-\frac{1}{2} \left(\frac{\delta(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))}{\sigma_\delta} \right)^2}$$

kde

- \mathbf{x} je střed filtru
- ξ je bod z okolí středu filtru
- \mathbf{f} je funkce jasu
- $d(\xi, \mathbf{x})$ je Eukleidovská vzdálenost bodů \mathbf{x} a ξ
- $\delta(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))$ je rozdíl jasu bodů \mathbf{x} a ξ

Důležitou vlastností tohoto filtru je, že u barevných fotografií pracuje se všemi barevnými kanály zároveň, nefiltruje každý barevný kanál zvlášť. Toho je dosaženo převedením obrazu do barevného modelu CIE-LAB [22], ve kterém se provede filtrace. Obraz je poté převeden zpět do modelu RGB. Pokud se filtruje každý kanál zvlášť bez informace o ostatních kanálech, vznikají pak na hranách aury, což je dáno tím, že kontrast této hrany není ve všech barevných kanálech stejný.

V Matlabu není bilaterální filtr přímo implementován, použili jsme však externí zdroj [22], který obsahuje implementaci tohoto filtru. Výsledek aplikace na náš testovací obraz, zavedený v kapitole 3.1.4. je na obrázcích 37.b–37.d. Zde dostáváme oproti předchozím dvěma metodám na první pohled nejhorší výsledek, oblast s přidaným šumem je sice zřejmá, ale vystupují zde poměrně výrazně hrany, které mohou narušovat další zpracování.

Podrobnosti ohledně výpočtu filtru, konkrétně pak algoritmus pro výpočet vah na základě jasové odlišnosti je podrobněji popsán v souboru `bfilter2.m` na příloženém CD.

3.3. Měření úrovně šumu

Výsledkem aplikace procedur, které jsme si popsali v kapitole 3.2. je odhad šumu v obraze. Nyní potřebujeme měřit hladinu šumu v rámci malé oblasti abychom mohli posoudit, zda se ve zkoumaném obraze vyskytují oblasti, kde je tato hladina výrazně odlišná. Budeme postupovat tak, že obraz, který obsahuje odhad šumu rozdělíme na disjunktní regiony o určité velikosti a pro tyto regiony pak vypočteme hodnotu, která nám bude charakterizovat hladinu šumu v tomto regionu. Velikost regionů je na zvážení, bude záviset na velikosti zkoumaného obrazu. Určitě by ale velikost regionu měla být menší, než je předpokládaná velikost podvržené oblasti. Uveďme si metody, které můžeme použít pro výpočet hodnoty charakterizující hladinu šumu v regionu.

3.3.1. Odstup signál-šum

Tato veličina, často označovaná jako SNR je běžně používána k vyjádření míry poškození signálu šumem. Je definována následujícím vztahem [24]:

$$SNR = 20 \log_{10} \frac{P_s}{P_n} = 20 \log_{10} \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |F_{i,j} - N_{i,j}|}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |N_{i,j}|} \quad (22)$$

kde

- P_s je intenzita signálu
- P_n je intenzita šumu
- F je funkce jasu ve zkoumaném obraze o rozměrech $m \times n$
- N je funkce jasu v odhadu šumu o rozměrech $m \times n$

Vzhledem k tomu, že hodnoty podílu signál/šum vycházejí obvykle velké, používá se logaritmické stupnice a odstup signál-šum je vyjadřován v decibelech (dB). Pokud je například intenzita signálu desetkrát větší, než intenzita šumu, pak je SNR rovno 20dB.

Měření odstupu signál-šum jsme pro ukázkou provedli na obraze z kapitoly 3.1.4. Odhad šumu jsme provedli pomocí Gaussova filtru v modré složce obrazu (obrázek 36.d). Velikost regionu jsme zvolili 16×16 pixelů. Výsledek je na obrázku 23. Hodnota jasu zde odpovídá odstupu odstupu signál-šum. Čím je odstup větší, tím je oblast světlejší. Oblasti, které měly v původním obraze nízkou hodnotu jasu mají i nízký odstup signál od šumu, jeví se zde proto tmavé. Naopak oblasti, které měly původně vyšší jas jsou i zde zobrazeny jasněji, mají vyšší

odstup signálu od šumu. Tmavší elipsovité oblasti uprostřed odpovídá oblasti s přidaným šumem. Zde je odstup signál od šumu menší, jeví se proto tmavší.



Obrázek 23.: Hodnoty SNR v kombinaci s aplikací Gaussova filtru

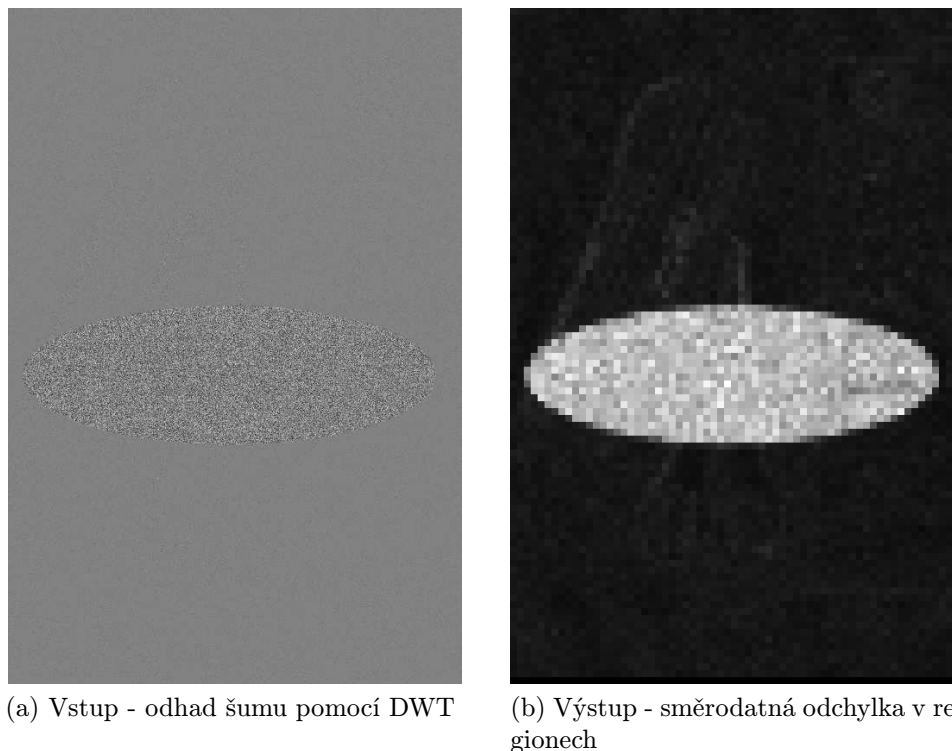
3.3.2. Směrodatná odchylka

Ne vždy bude výhodné popisovat šum pomocí hodnoty SNR. Bude to například v případě, kdy použijeme metodu diskretní waveletové transformace. Při použití této metody, popsané v kapitole 3.2.1. získáme komponentu, ze které zkoumáme šum. Abychom nemuseli provádět redukci šumu a počítat zpětnou transformaci, bude výhodné, pokud spočítáme charakteristiku šumu přímo ze získané komponenty. Zdroj [6] uvádí pro výpočet vztah

$$\hat{\sigma} = \frac{\text{med}(|region|)}{0.6745} \quad (23)$$

kde *region* je soubor hodnot šumu o velikosti $n \times n$. Připomeňme, že pojem „region“ pochází z úvodu kapitoly 3.3., kde jsme deklarovali rozdělení zkoumaného odhadu šumu na disjunktní regiony. Nyní se podívejme na význam tohoto vztahu. Jedná se o **odhad** směrodatné odchylky šumu z absolutní hodnoty mediánu. Zde se vychází z **předpokladu**, že hodnoty, se kterými počítáme mají normální rozdělení a střední hodnotu rovnou nule, blíže je vztah 23 vysvětlen v [11].

Směrodatná odchylka je míra variability a v našem případě vypovídá o tom, jak moc se od sebe liší hodnoty šumu. Je-li malá, jsou si hodnoty šumu ve zkoumaném regionu hodně podobné, pokud je velká, signalizuje velké odlišnosti v hodnotách šumu. Pomocí směrodatné odchylky lze, jak si ověříme později, charakterizovat šum v určitém regionu a odlišit od sebe regiony více zašumělé a méně zašumělé.



Obrázek 24.: Směrodatná odchylka šumu

Na obrázku 24.b je výsledek aplikace této metody na odhad šumu 24.a získaný v kapitole 3.2.1.. Hodnoty jsme počítali pro regiony o velikosti 12×12 pixelů. Na výstupu je oblast s odlišnou hladinou šumu více zřejmá, než na vstupu.

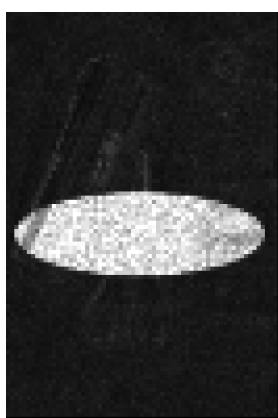
3.4. Nalezení oblastí s různými úrovněmi šumu

Naším úkolem je v obraze, potažmo v odhadu šumu najít souvislé oblasti, ve kterých je úroveň šumu s jistou tolerancí stejná. Navážeme na postup uvedený v předchozí kapitole, kde je výstupem popsanych metod obraz obsahující disjunktní regiony, přičemž hodnota jasu každého regionu odpovídá úrovni šumu v dané oblasti obrazu. Potřebujeme provést sloučení těchto regionů do větších souvislých oblastí. K tomuto účelu lze použít několik metod, my využijeme metodu založenou na hledání hran, kterou si popíšeme v následující kapitole.

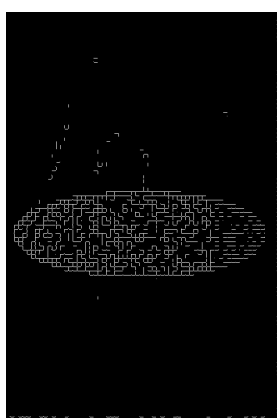
3.4.1. Detekce hran

Metoda je založena na detekci hran a na morfologických operacích [21]. S podobnými metodami se lze setkat pod pojmem segmentace obrazu, protože se v podstatě jedná o rozdělení obrazu do segmentů na základě určitého kriteriia. Pokud se v obraze (v našem případě v odhadu jeho šumové složky) nachází objekt, který je v určitém kontrastu s okolím, dokážeme najít jeho hranice. Čím je kontrast větší, tím je tato detekce snadnější a spolehlivější.

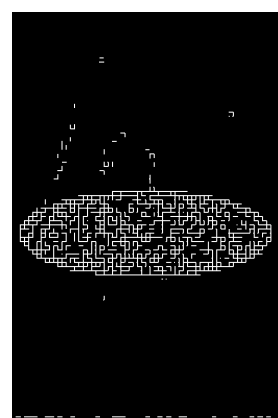
Popišme si podrobně jednotlivé kroky, které vedou k nalezení hranice. Na obrázcích 25.a–25.f si zároveň budeme tyto kroky ilustrovat. Metodu aplikujeme na mapu regionů s charakteristikami šumu, viz obrázek 25.a. Pro účely zobrazení (nikoliv však pro výpočet) zobrazení jsme na obrázek aplikovali metodu roztažení kontrastu, která je podrobněji popsána v [20].



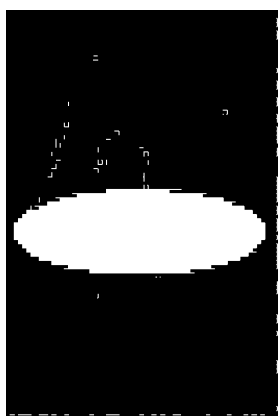
(a) Odhady šumu v jednotlivých regionech.



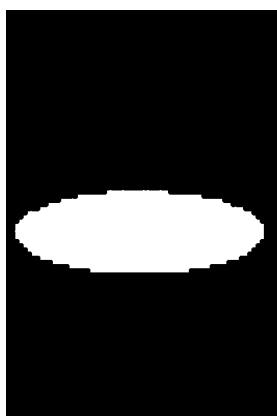
(b) Hrany vypočítané pomocí sobelova hranového detektoru.



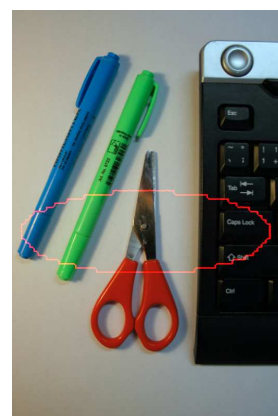
(c) Hrany po aplikaci dilatace.



(d) Segmenty po aplikaci algoritmu na vyplnění děr.



(e) Segmenty po aplikaci eroze.



(f) Původní obraz s vyznačenou hranicí vypočítaného segmentu.

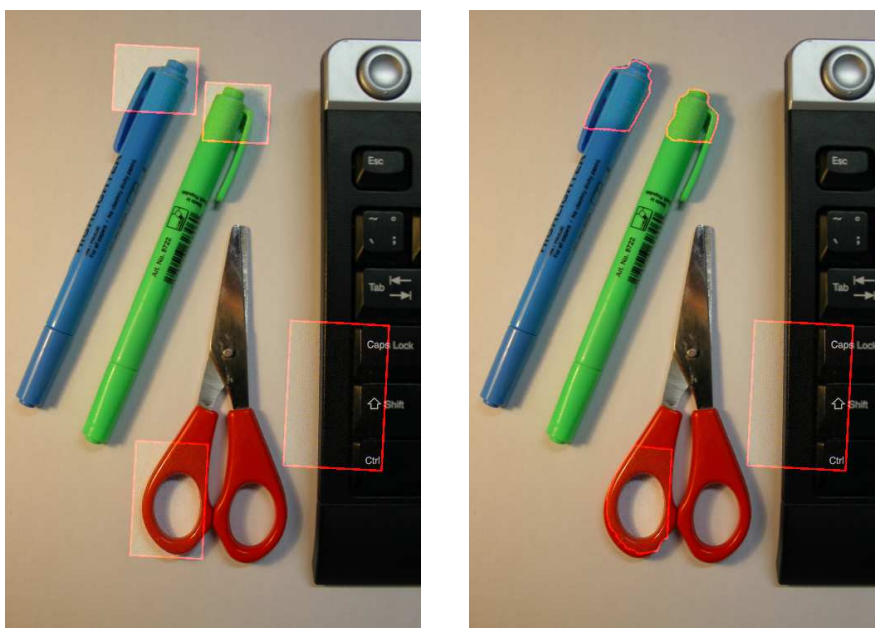
Obrázek 25.: Postup při segmentaci obrazu metodou hledání hranice

Prvním krokem, viz obrázek 25.b je nalezení hran hledaných objektů. K tomu lze použít některý z hranových detektorů. Osvědčil se Sobelův operátor, u kterého provádíme regulaci prahové hodnoty vynásobením prahu stanoveného funkcí `edge` z MATLABu. Podrobněji se regulaci prahu věnujeme v kapitole 3.5. a v kapitole 3.7. si uvedeme návaznost stanovení násobitele na úspěšnost detekce. Na obrázku je patrné, že uprostřed se vyskytuje elipsovitý objekt, nicméně jeho hranice není spojitá. Dalším krokem zpracování je aplikace dilatace, čímž odstraníme drobné nesouvislosti hran a objekt tak získá souvislou uzavřenou hranici, viz obrázek 25.c. Uzavřená hranice objektu je podstatná pro aplikaci dalšího kroku, vyplnění děr. Dírou se rozumí podmnožina pixelů pozadí, která nemůže být dosažena vyplněním pozadí začínajícím na okraji obrazu. Přeneseno do praxe, na obrázku 25.d je oproti předchozímu kroku vyplněna plocha uvnitř elipsy.

Na obrázku 25.d však zbývají ještě hrany, které jsme úspěšně detekovali, nicméně neohraničují žádný objekt. Pomocí dilatace se jich zbavíme a výsledkem je pouze jediný segment, viz obrázek 25.e. Z tohoto segmentu můžeme vzít hranici a přidat ji do původního obrazu, čímž vyznačíme oblast, ve které jsme našli odlišnou charakteristiku šumu, než ve zbytku obrazu. Výsledek je na obrázku 25.f. Zde jsme provedli experiment pouze na červené složce obrazu. Výsledky aplikace na ostatní barevné složky jsou podobné a kombinací těchto výsledků můžeme dosáhnout větší přesnosti a lepší robustnosti detekce. V našem testu, který je podrobně popsán v kapitole 3.5. provádíme logický součet výsledků pro jednotlivé barevné kanály. K úspěšné detekci bloku tedy stačí, aby byl detekován alespoň v jedné barevné složce. Důvodem toho postupu je, že například pokud je podvržena jednobarevná plocha, nepodaří se nám ji detekovat ve všech barevných kanálech naráz, ale mnohdy pouze v jednom z nich.

3.5. Ukázky

V předchozích kapitolách jsme uvedené metody aplikovali na obraz, kde jsme lokálně přidali šum ve Photoshopu. To nám zajistilo, že šum v oblasti inkonzistence, kterou jsme hledali měl přesně takové parametry, jaké jsme předpokládali. Navíc jsme použili takový šum, který jsme byli schopni detekovat, aby ilustrace metod byla názorná. Nyní budeme zkoumat obrazy, kde šum pochází přímo z fotografického zařízení a je způsoben vlivy, které jsme si popsali v kapitolách 3.1.1. a 3.1.2..



(a) Obraz č.1

(b) Obraz č.2

Obrázek 26.: Testovací obrazy s ohraničenými zkopírovanými oblastmi.

Pro účely této kapitoly jsme vytvořili dva obrazy kde kombinujeme části dvou různých snímků. Jako základ je použit snímek z digitálního fotoaparátu Canon EOS 400D při citlivosti ISO 100. Dále jsme pořídili snímek stejné scény fotoaparátem Panasonic DMC-FZ7 při citlivosti ISO 400. Z druhého snímku jsme vzali výřezy určitých částí a umístili je do prvního snímku, čímž vznikl obraz č.1. Tyto výřezy jsou na první pohled poměrně zřejmé i kdybychom nepoužili žádného strojového zpracování zkoumající šum. Nicméně chceme detekovat tyto oblasti strojově. V obraze č.2 jsme přidané oblasti upravili, aby nebyly na první pohled tolik zřejmé. Výřezy ze zmíněných obrazů jsou na obrázcích 27.a–27.d, na obrázcích 26.a a 26.b jsou pak testovací obrazy s ohraničenými oblastmi, které jsme zkopírovali. Tyto obrázky nám poslouží pro vizuální hodnocení výsledků. Pro znázornění výsledků jednotlivých metod budeme používat stejné ohraničení.



(a) Detail prvního testovacího obrazu.



(b) Detail prvního testovacího obrazu.



(c) Detail druhého testovacího obrazu.



(d) Detail druhého testovacího obrazu.

Obrázek 27.: Výřezy z použitých testovacích obrazů.

Nyní na oba obrazy budeme aplikovat metody popsané v kapitole 3.2. přičemž se pokusíme detekovat regiony zkopírované z jiné fotografie.

3.5.1. Aplikace DWT

Při detekci s odhadem šumu pomocí diskretní waveletové transformace, popsané v kapitole 3.2.1., jsme využili Daubechies wavelet D8. Zvolili jsme právě tento, jelikož v literatuře je k tomuto účelu hojně používán a dává zde dobré výsledky, viz [8],[9],[6]. Na odhad šumu jsme poté aplikovali regionální měření šumu 3.3.2., přičemž velikost regionu byla 12×12 pixelů. Poté jsme provedli segmentaci pomocí metody detekce hran popsané v kapitole 3.4.1. a to jak přímo na odhad šumu, tak na odhad šumu rozdělený do regionů. Během této segmentace jsme upravovali práh pro hranový detektor tak, abychom dosáhli optimálního výsledku. Podrobněji se regulaci prahu věnujeme v kapitole 3.7.. Detekované bloky jsou orámovány červeně na obrázcích 28.a, 28.b, 29.a a 29.b.



(a) Bez rozdělení do regionů.



(b) S rozdělením do regionů.

Obrázek 28.: Detekce pomocí DWT pro obraz č.1.



(a) Bez aplikace segmentace.



(b) S aplikací segmentace.

Obrázek 29.: Detekce pomocí DWT pro obraz č.2.

3.5.2. Aplikace Gaussova filtru



(a) Bez rozdělení do regionů.



(b) S rozdělením do regionů.

Obrázek 30.: Detekce pomocí Gaussova filtru pro obraz č.1.



(a) Bez rozdělení do regionů.



(b) S rozdělením do regionů.

Obrázek 31.: Detekce pomocí Gaussova filtru pro obraz č.2.

Při detekci s odhadem šumu pomocí Gaussova filtru jsme pro filtrování využili konvoluční jádro o rozměrech 5×5 pixelů a směrodatnou odchylkou $\sigma = 0.8$. Dále jsme postupovali stejně jako v předchozím případě. Aplikovali jsme regionální měření šumu pro bloky o velikosti 12×12 pixelů a následně provedli segmentaci metodou detekce hran jak přímo na odhad šumu, tak na odhad šumu rozdělený do regionů. Detekované oblasti s odlišnou úrovní šumu jsou vyznačeny červeně na obrázcích 30.a, 30.b, 31.a a 31.b.

3.5.3. Aplikace bilaterálního filtru

Při detekci s odhadem šumu pomocí bilaterálního filtru jsme nastavili následující parametry filtru: poloměr 3 pixely, směrodatné odchylky jsme ponechali výchozí, viz implementace v [22]. Dále jsme postupovali stejně jako v předchozím případě. Aplikovali jsme regionální měření šumu pro bloky o velikosti 12×12 pixelů a následně provedli segmentaci metodou detekce hran přímo na odhad šumu a na odhad šumu rozdělený do regionů. I zde jsme ručně ladili práh pro hrannový detektor tak, abychom dosáhli co nejlepšího výsledku. Detekované oblasti s odlišnou úrovní šumu jsou vyznačeny červeně na obrázcích 32.a, 32.b, 33.a a 33.b.



(a) Bez rozdělení do regionů.

(b) S rozdělením do regionů.

Obrázek 32.: Detekce pomoc bilaterálního filtru pro obraz č.1.



(a) Bez rozdělení do regionů.

(b) S rozdělením do regionů.

Obrázek 33.: Detekce pomocí bilaterálního filtru pro obraz č.2.

3.6. Hodnocení metod

V předchozích kapitolách jsme měli možnost porovnat výsledky detekce pomocí jednotlivých metod přímo na testovacích obrazech tak, že jsme si vyznačili hranice detekovaných oblastí a porovnali je s oblastmi, které byly skutečně zkopírovány z jiné fotografie. V této kapitole se budeme věnovat hodnocení metod pomocí statistického testování hypotéz [23].

Nejdříve je třeba zformulovat dvě hypotézy. Nulovou hypotézu H_0 , o jejíž platnosti chceme rozhodnout a alternativní hypotézu H_1 , která vymezuje situaci, do které se dostaneme, pokud neplatí hypotéza H_0 . Nepřipouštíme přitom, že by existovala jiná možnost, vždy platí buď H_0 nebo H_1 . Chybou 1.druhu α rozumíme případ, kdy zamítneme hypotézu H_0 přestože platí. Chybou 2.druhu β pak rozumíme případ, kdy nezamítneme H_0 pokud platí H_1 .

Aplikujme tuto teorii na náš případ testování úspěšnosti detekce podvržených oblastí. Mějme dvě podmnožiny pixelů zkoumaného obrazu F :

- S_f je podmnožina pixelů, která je podvržená
- S_d je podmnožina pixelů, která je detekována jako podvržená

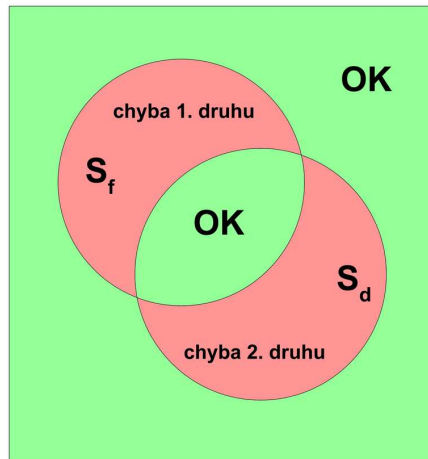
S použitím těchto množin formulujeme hypotézy:

- H_0 : Pixel v S_d je podvržený.
- H_1 : Pixel v S_d není podvržený.

Na základě těchto formulací vyjádříme chyby 1. a 2. druhu:

$$\alpha = \frac{|S_f - S_d|}{|F|} \quad (24)$$

$$\beta = \frac{|S_d - S_f|}{|F|} \quad (25)$$



Obrázek 34.: Grafické znázornění chyb

3.7. Srovnání metod

V této kapitole si uvedeme grafické porovnání metod. Pro zkrácení názvů a lepší přehlednost si zavedeme následující označení:

- WN – použití DWT bez rozdělení do regionů
- WS – použití DWT s rozdělením do regionů
- GN – použití Gaussova filtru bez rozdělení do regionů
- GS – použití Gaussova filtru s rozdělením do regionů
- BN – použití bilaterálního filtru rozdělení do regionů
- BS – použití bilaterálního filtru s rozdělením do regionů

Pokud zde mluvíme o použití metody s rozdělením do regionů, detekujeme podvržené oblasti v odhadu šumu, na který byla aplikována metoda 3.3.2. Jinak detekujeme podvržené oblasti přímo v odhadu šumu. Grafické výsledky porovnání metod pro pět zkušebních fotografií jsou na obrázcích 41. – 40. U každého

obrázku je uveden náhled zkoumané fotografie, dále maska, kde jsou bílou barvou vyznačeny zkopírované oblasti. Následují grafy chyb 1. a 2.druhu pro každou z metod v seznamu. Interpretaci těchto chyb jsme si formálně popsali v kapitole 3.6. Výskyt chyby 1. druhu znamená neodhalení zkopírované oblasti. Výskyt chyby 2. druhu znamená mylné označení oblasti jako zkopírované. Chyby jsou v grafech uvedeny jako funkce prahu, tímto prahem rozumíme násobitele, pomocí kterého upravujeme prahovou hodnotu popsanou v kapitole 3.4.1.

Na základě uvedených experimentálních výsledků lze vyhodnotit jako nejlepší metodu využívající diskrétní waveletovou transformaci. Chyba 1. druhu, tedy neúspěšné detekce, se oproti ostatním metodám obvykle pohybuje v menších hodnotách při současném zachování nízkých hodnot chyby 2. druhu, což jsou detekce falešné. Metoda využívající bilaterální filtr dává v některých případech, například u obrázku 38. dobrý výsledek, ale je ze všech uvedených výpočetně nejnáročnější, což je značná nevýhoda. Metodu aplikující Gaussův filtr bychom zařadili co se týče výsledků mezi tyto dvě uvedené. Obvykle nepodává tak dobré výsledky, jako metoda využívající DWT, ale je významně rychlejší, než metoda využívající bilaterální filtr.

Závěr

V první části práce jsme zkoumali metody odhalování manipulace digitální fotografie, která je provedena duplikováním části obrazu. Uvedli jsme si, jakým způsobem tento typ manipulace detekovat. Blíže jsme si představili tři srovnávací kritéria, která lze aplikovat v prostorové doméně. Pro zrychlení výpočtu jsme zavedli analýzu obrazu ve frekvenční doméně. Výsledky jsme poté upřesňovali v prostorové doméně. Uvedli jsme si ukázky detekcí a porovnali použité metody.

V druhé části práce jsme se věnovali odhalování fotomontáže realizované složením dvou různých fotografií. Popsali jsme celkem tři možné postupy, jak tento druh manipulace odhalit. Všechny tři byly založeny na předpokladu rozdílných vlastností šumu zdrojových obrázků. Provedli jsme ukázkovou aplikaci těchto metod a představili si způsob, jak metody testovat a porovnávat. Oproti metodě popsané v článku [6], ze které jsme vycházeli, jsme navíc provedli aplikace na reálné snímky, kde je hledání manipulovaných oblastí obtížnější, než při přidání silného šumu v grafickém editoru. Na závěr jsme metody vzájemně porovnali na základě dosažených výsledků.

Dalším pokračováním této práce by mohlo být zdokonalování uvedených metod a zkoumání dalších, které by se zaměřily na jiné techniky manipulace s digitální fotografií. Kromě dvou uvedených technik existuje ještě řada dalších. Skloubením s metodami odhalujícími i jiné typy manipulace bychom dosáhli spolehlivějšího ověřování pravosti fotografií.

Reference

- [1] GONZALES, R. C., WOODS, R. E.
Digital Image Processing.
Prentice Hall, 2002. ISBN 0-13-094650-8.
- [2] GONZALES, R. C., WOODS, R. E., EDDINS, S. L.
Digital Image processing using MATLAB.
Pearson Prentice Hall, 2004. ISBN 0130085197, 9780130085191.
- [3] FRIDRICH, Jessica, SOUKAL, David, LUKÁŠ, Jan. D
etection of Copy-Move Forgery in Digital Images. [Internet] [4.1.2010]
<http://www.ws.binghamton.edu/fridrich/Research/copymove.pdf>
- [4] WANG, Zhou, BOVIK, Alan Conrad , SHEIKH, Hamid Rahim, SIMON-CELLI, Eero P.
Image Quality Assessment: From Error Visibility to Structural Similarity.
[Internet] [4.1.2010]
<http://www.cns.nyu.edu/ftp/lcv/wang03-preprint.pdf>
- [5] POPESCU, Alin C, FARIDY, Hany.
Exposing Digital Forgeries by Detecting Duplicated Image Regions. [Internet]
[4.1.2010]
<http://www.ists.dartmouth.edu/library/102.pdf>
- [6] MAHDIAN, Babak, SAIC, Stanislav.
Using noise inconsistencies for blind image forensics.
- [7] MĚŘÍNSKÝ, Zdeněk, GAVLASOVÁ, Andrea, HOŠŤÁLKOVÁ Eva.
Užití Diskrétní wavelet transformace v potlačování rušivých složek biomedicínských obrazů. [Internet] [25.3.2011]
http://dsp.vscht.cz/konference_matlab/MATLAB08/prispevky/071_merinsky.pdf
- [8] SUDHA, S., SURESH, G.R., SUKANESH, R.
Speckle Noise Reduction in Ultrasound Images by Wavelet Thresholding based on Weighted Variance. [Internet] [25.3.2011]
<http://www.ijcte.org/papers/002.pdf>
- [9] K LAPETEK, Petr.
Wavelet denoising. [Internet] [25.3.2011]
<http://klapetek.cz/wdenoise.html>
- [10] SIMANEK, Donald E.
Distribution of measurements [Internet] [25.3.2011]
<http://www.lhup.edu/~dsimaneke/scenario/errorman/distrib.htm>

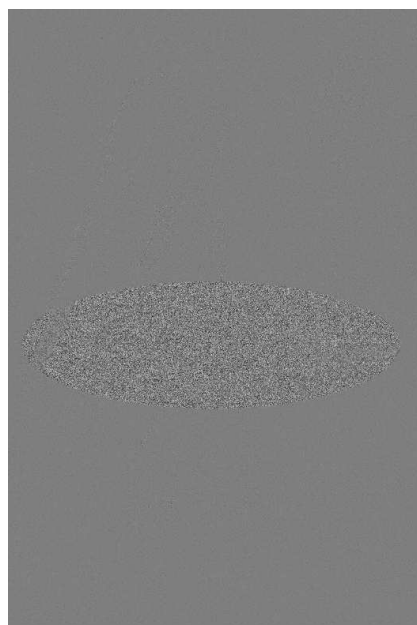
- [11] GROSS, Jürgen.
Linear regression.
Berlin : Springer, 2003.
- [12] LIU, Ce, FREEMAN, William T. , SZELISKI, Richard, KANG, Sing Bing
Noise Estimation from a Single Image. [Internet] [13.2.2011]
<http://people.csail.mit.edu/ce-liu/denoise/estnoise/noise.pdf>
- [13] TOMASI, Carlo, MANDUCHI, Roberto
Bilateral Filtering for Gray and Color Images. [Internet] [15.2.2011]
<http://users.soe.ucsc.edu/~manduchi/Papers/ICCV98.pdf>
- [14] QUANTUM SCIENTIFIC IMAGING, INC.
Understanding CCD Read Noise . [Internet] [13.2.2011]
http://www.qsimaging.com/ccd_noise.html
- [15] WWW.CELLMIGRATION.COM – CMC ACTIVITY CENTER.
Flat Field Correction. [Internet] [13.2.2011]
http://www.cellmigration.org/resource/biosensors/ppt-pdf/2flat_field.pdf
- [16] ACKLAM, Peter J.
MATLAB array manipulation tips and tricks. [Internet] [5.2.2010]
<http://cseweb.ucsd.edu/~datorres/docs/MatlabTip.pdf>
- [17] THE MATHWORKS, INC. *Wavelet Toolbox - dwt2*. [Internet] [14.2.2011]
<http://matlab.izmiran.ru/help/toolbox/wavelet/dwt2.html>
- [18] *MATLAB - online documentation*
<http://www.mathworks.com/access/helpdesk/help/techdoc/>
- [19] *The USC-SIPI Image Database*. [Internet] [24.1.2010]
<http://sipi.usc.edu/database/>
- [20] FISHER R., PERKINS S., WALKER A., WOLFART E.
Image processing learning resources. [Internet] [27.2.2011]
http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm
- [21] MATHWORKS – IMAGE PROCESSING TOOLBOX OVERVIEW
Detecting a Cell Using Image Segmentation. [Internet] [28.2.2011]
<http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexcell.html>
- [22] LANMAN, Douglas
Bilateral Filtering. [Internet] [28.2.2011]
<http://www.mathworks.com/matlabcentral/fileexchange/12191>

- [23] ŠEDIVÁ, Blanka
Parametrické testy. [Internet] [8.4.2011]
<http://home.zcu.cz/~sediva/stav/kap07.pdf>
- [24] *Wikipedia - The Free Encyklopedia*
<http://en.wikipedia.org>
- [25] *Wikipedia - PSNR*
http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
- [26] *Wikipedia - MSE*
http://en.wikipedia.org/wiki/Mean_squared_error
- [27] *Wikipedia - Correlation and dependence*
http://en.wikipedia.org/wiki/Correlation_and_dependence
- [28] *Wikipedia - The Free Encyklopedia*
http://en.wikipedia.org/wiki/Bayer_filter
- [29] *Wikipedia - The Free Encyklopedia*
<http://cs.wikipedia.org/wiki/APS-C>
- [30] *Wikipedia - The Free Encyklopedia*
http://en.wikipedia.org/wiki/Discrete_wavelet_transform
- [31] *Lenna - Wikipedia, The Free Encyklopedia*
<http://en.wikipedia.org/wiki/Lenna>
- [32] *AWGN - Wikipedia, The Free Encyklopedia*
http://en.wikipedia.org/wiki/Additive_white_Gaussian_noise
- [33] *Scale space implementation - Wikipedia, The Free Encyklopedia*
http://en.wikipedia.org/wiki/Scale_space_implementation

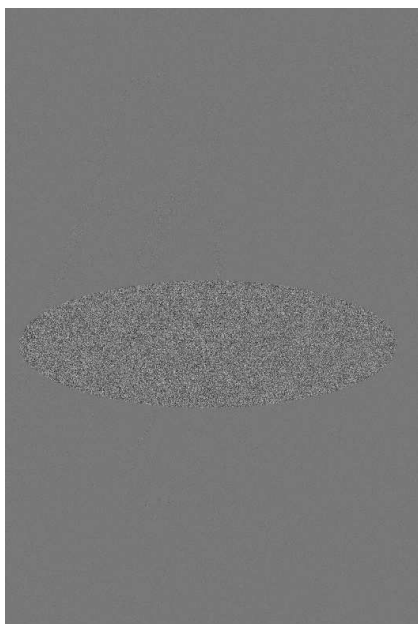
A. Výsledky odhadu šumu



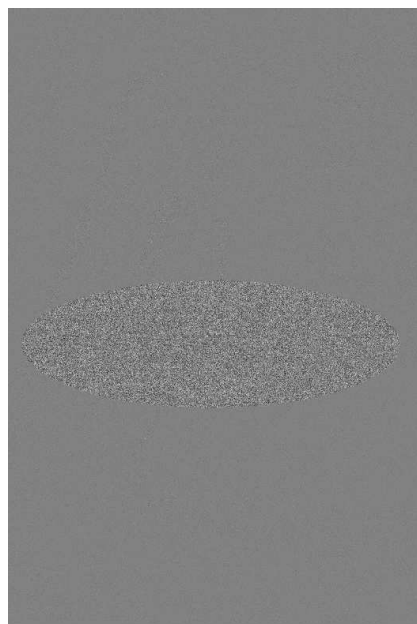
(a) Obraz s lokálně přidaným šumem.



(b) Komponenta $cD_1^{(d)}$ pro červený kanál.



(c) Komponenta $cD_1^{(d)}$ pro zelený kanál.

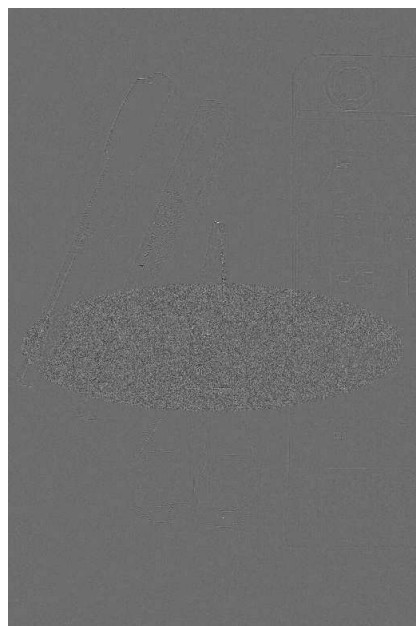


(d) Komponenta $cD_1^{(d)}$ pro modrý kanál.

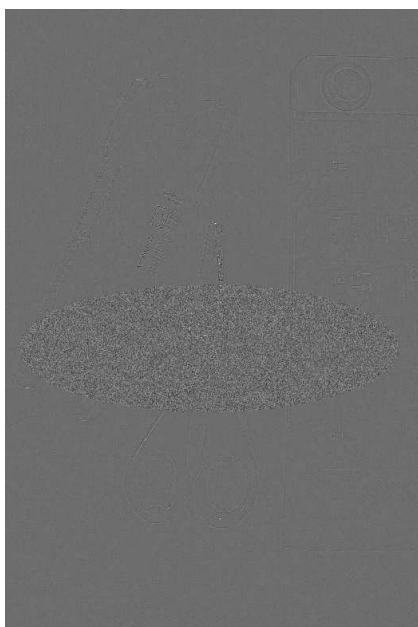
Obrázek 35.: Odhad šumu za použití diskrétní waveletové transformace



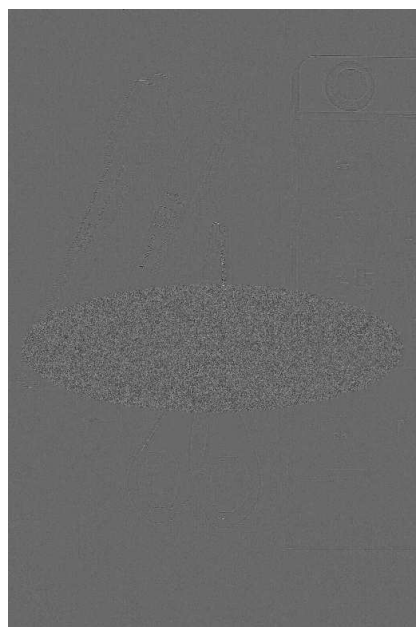
(a) Obraz s lokálně přidaným šumem.



(b) Odhad šumu v červené složce.



(c) Odhad šumu v zelené složce.

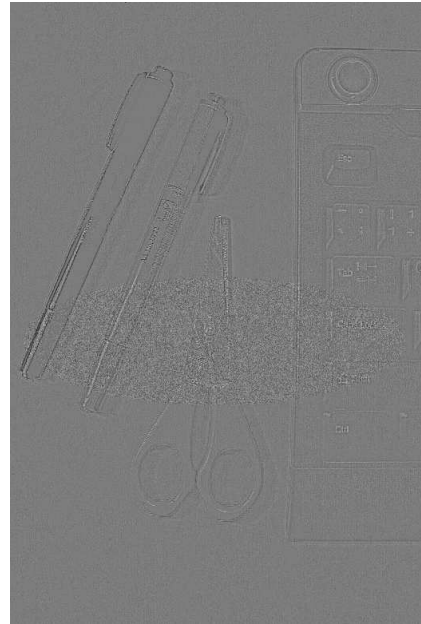


(d) Odhad šumu v modré složce.

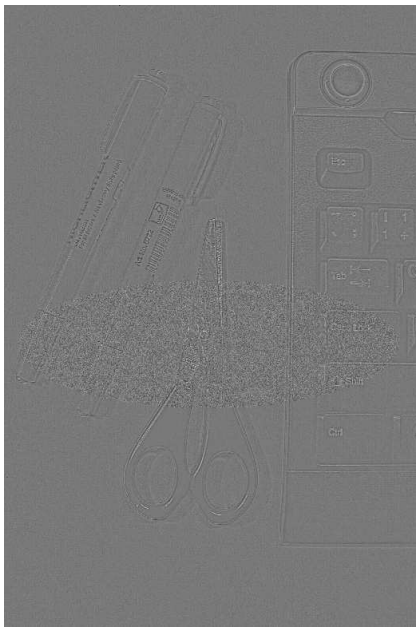
Obrázek 36.: Odhad šumu za použití Gaussova filtru.



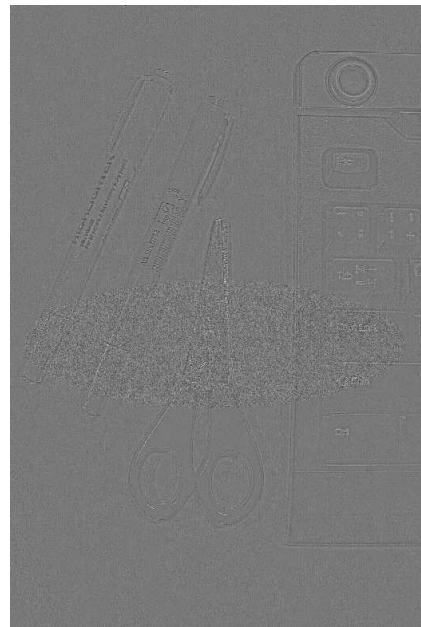
(a) Obraz s lokálně přidaným šumem.



(b) Odhad šumu v červené složce.



(c) Odhad šumu v zelené složce.



(d) Odhad šumu v modré složce.

Obrázek 37.: Odhad šumu za použití bilaterálního filtru.

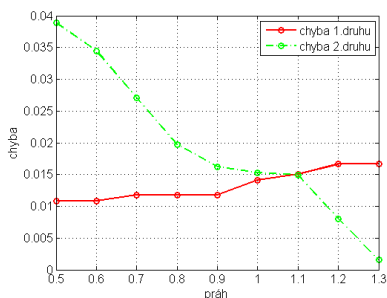
B. Srovnání metod detekce inkonzistence šumu



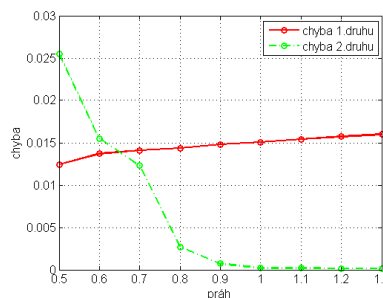
(a) Zkoumaný obrázek



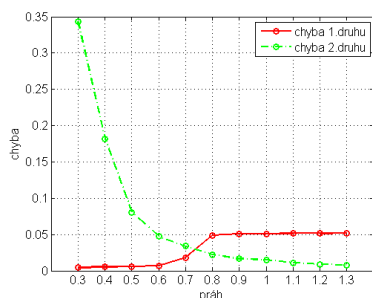
(b) Maska



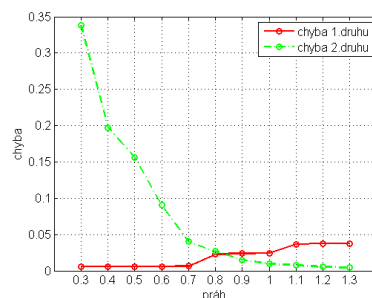
(c) WS



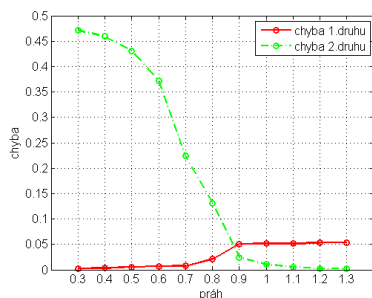
(d) WN



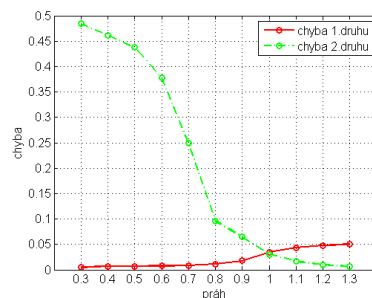
(e) GS



(f) GN

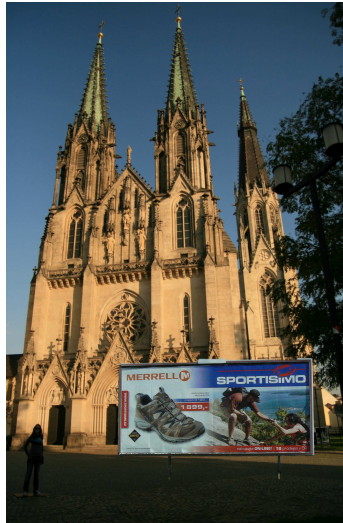


(g) BS



(h) BN

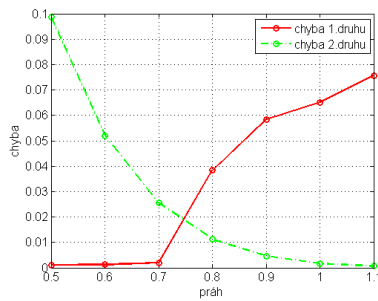
Obrázek 38.: Obrázek „Letadlo“, velikost 1200×660 pixelů



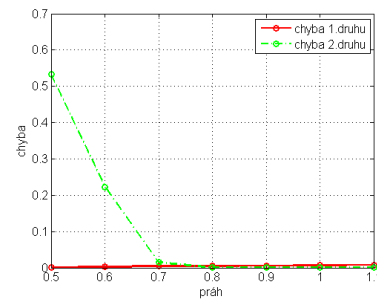
(a) Zkoumaný obrázek



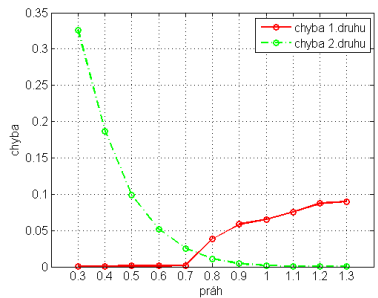
(b) Maska



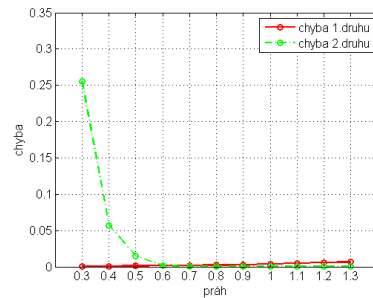
(c) WS



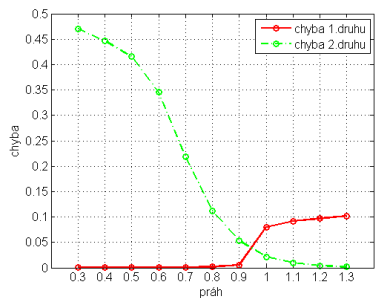
(d) WN



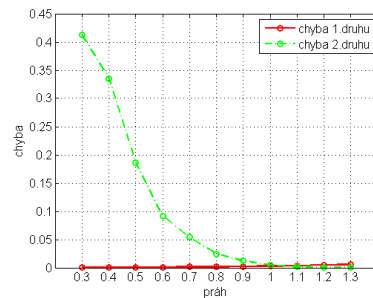
(e) GS



(f) GN



(g) BS

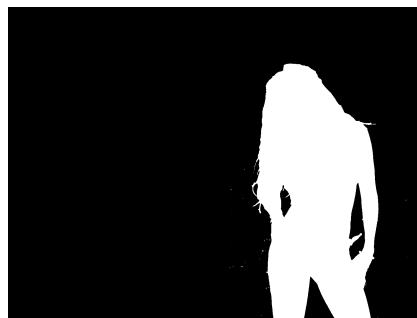


(h) BN

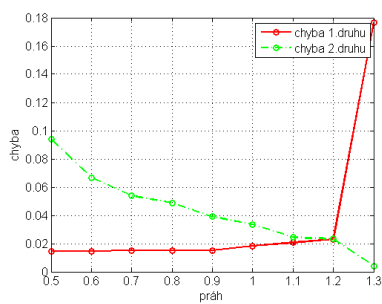
Obrázek 39.: Obrázek „Dóm“, velikost 24880 × 3818 pixelů



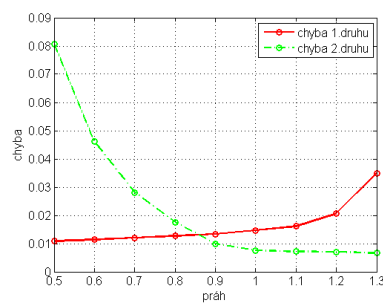
(a) Zkoumaný obrázek



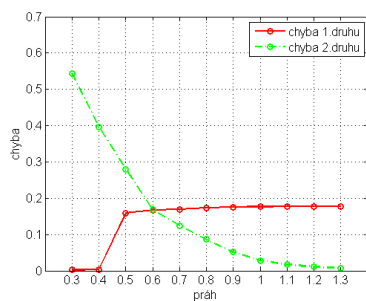
(b) Maska



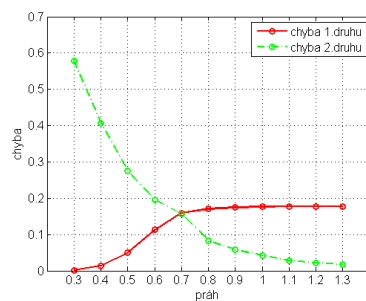
(c) WS



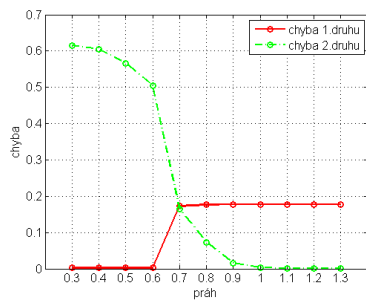
(d) WN



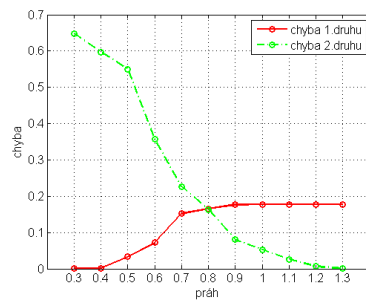
(e) GS



(f) GN



(g) BS

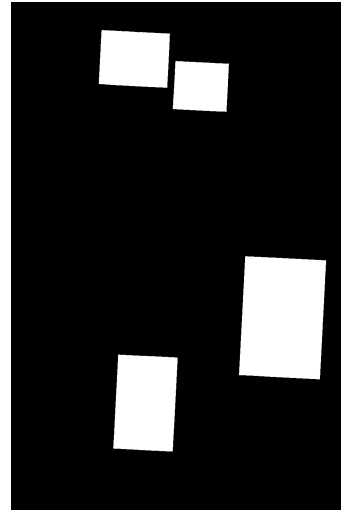


(h) BN

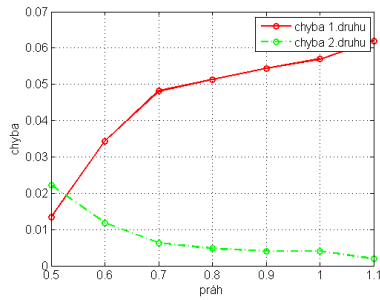
Obrázek 40.: Obrázek „Severní pól“, velikost 1600×1200 pixelů



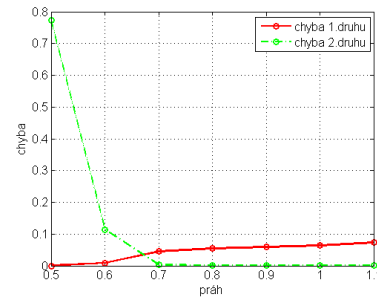
(a) Zkoumaný obrázek



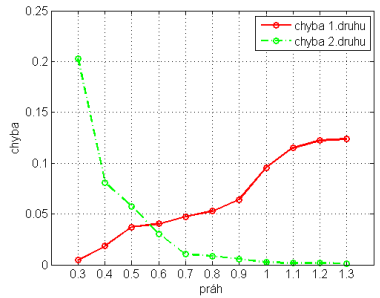
(b) Maska



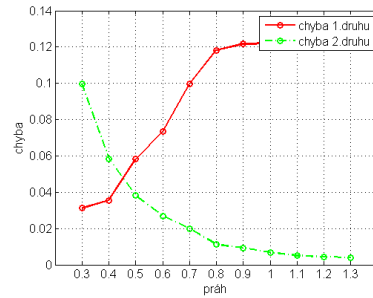
(c) WS



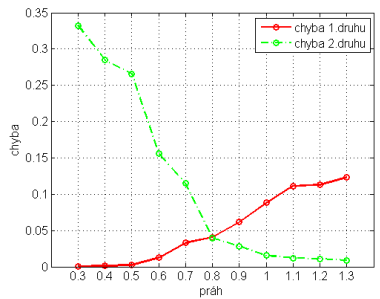
(d) WN



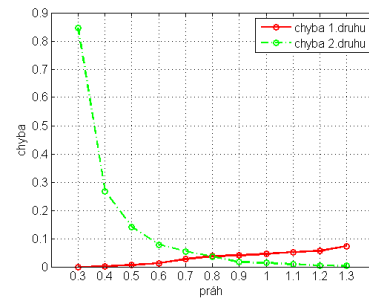
(e) GS



(f) GN



(g) BS

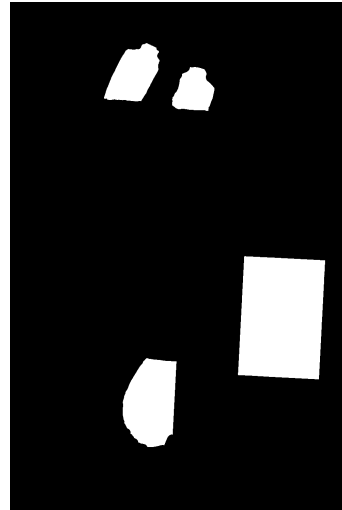


(h) BN

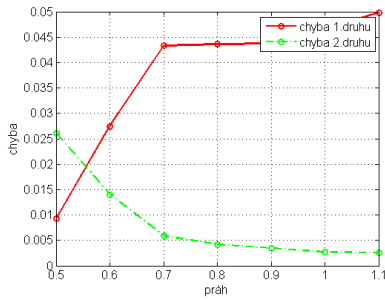
Obrázek 41.: Obrázek č.1, velikost 1000×1500 pixelů



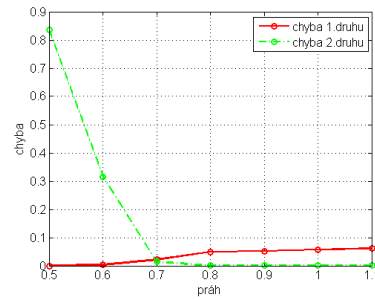
(a) Zkoumaný obrázek



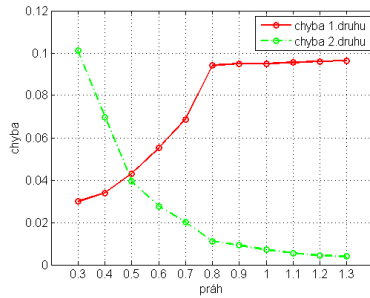
(b) Maska



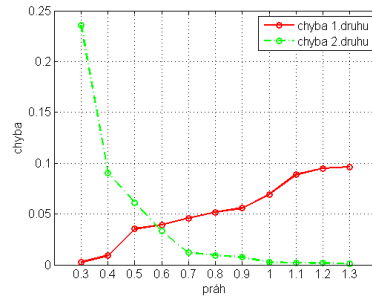
(c) WS



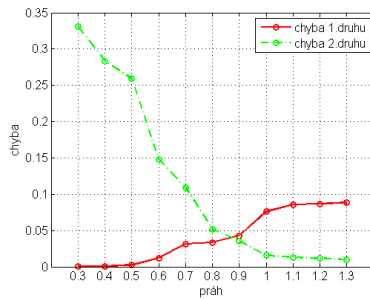
(d) WN



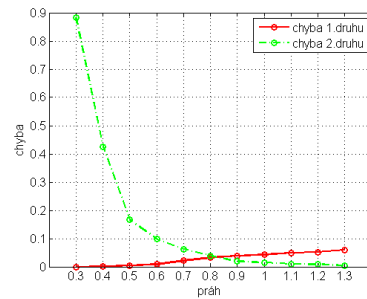
(e) GS



(f) GN



(g) BS



(h) BN

Obrázek 42.: Obrázek č.2, velikost 1000×1500 pixelů

C. Obsah příloženého CD

V samotném závěru práce je uveden stručný popis obsahu příloženého CD/DVD, tj. závazné adresářové struktury, důležitých souborů apod.

`doc/`

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

`src/`

Kompletní zdrojové texty uvedených metod interpretovatelné v programu MATLAB. Zejména jsou to soubory

- `duplicity.m` – implementace metod popsaných v kapitole 2.
- `noise_wavelets.m` – implementace metody popsané v kapitole 3.5.1.
- `noise_bilateral.m` – implementace metody popsané v kapitole 3.5.3.
- `noise_gauss.m` – implementace metody popsané v kapitole 3.5.2.

Ostatní soubory obsahují implementace funkcí volaných z těchto souborů.

`readme.txt`

Instrukce pro interpretaci zdrojových souborů.

Navíc CD/DVD obsahuje:

`data/`

Ukázková a testovací data použitá v práci a pro potřeby obhajoby práce.

`literature/`

Některé položky literatury odkazované z dokumentace práce.

U veškerých odjinud převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovoluují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru `readme.txt`.