VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

TECHNIKY POUŽÍVANÉ PRO OSTŘENÍ A ROZMAZÁVÁNÍ OBRAZU

TECHNIQUES USED FOR IMAGE SMOOTHING, BLURRING AND SHARPENING

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE JIŘÍ KUBÍNEK
AUTHOR

VEDOUCÍ PRÁCE Ing. JIŘÍ VENERA
SUPERVISOR

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Kubínek Jiří**
Obor: Informační technologie
Téma: **Techniky používané pro ostření a rozmazávání obrazu**
Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte algoritmy používané pro ostření, rozmazávání a vyhlazování obrazu.
2. Pro každou uvedenou techniku implementujte více než jeden algoritmus. Výsledky demonstrujte na vhodných datech.
3. Porovnejte tyto algoritmy z hlediska časové složitosti.
4. Zhodnoťte Vámi dosažené výsledky a porovnejte s jinými existujícími přístupy.

Literatura:

- Po domluvě na vyžádání u školitele.

Při obhajobě semestrální části projektu je požadováno:

- Bod 1., započetí řešení bodu 2..

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Venera Jiří, Ing.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Jiří Kubínek**
Id studenta: 84289
Bytem: Pod Čechem 1409, 739 34 Šenov
Narozen: 15. 10. 1984, Frýdek-Místek
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   bakalářská práce

Název VŠKP: Techniky používané pro ostření a rozmazávání obrazu
Vedoucí/školitel VŠKP: Venera Jiří, Ing.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☒ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....	.....
Nabyvatel	Autor

Abstrakt

Práce je věnovaná metodám pro úpravu digitálního obrazu. Definuje nezbytné základní pojmy v této problematice, jako je barevný prostor či šum. Především však rozebírá samotné metody umožňující ostření a rozmazávání obrazu. Popisuje několik nejznámějších postupů jak z hlediska teorie, tak následně i nastiňuje jejich implementaci v programovacím jazyce C. Algoritmy jsou dále porovnávány z hlediska časové složitosti. Cílem této práce je seznámení s problematikou filtrace obrazu a vysvětlení základních postupů užívaných k jejich realizaci.

Klíčová slova

Šum, barevný prostor, RGB, HLS, ostření, hrana, Laplaceův operátor, konvoluce, Unsharp masking, Sobelův operátor, Robinsonův operátor, Kirschův operátor, Cannyho hranový detektor, rozmazávání, průměrování, Bilaterální filtr, Gaussův filtr, mediánový filtr, konzervativní filtr

Abstract

This work is dedicated to methods used for digital image editing. It defines fundamental terms for this work as color space or noise. Above all, it analyses methods allowing image sharpening and blurring. It describes some of the most known algorithms from the theoretical point of view, but also introduces their implementation in C programming language. There are compared according to time complexity. The purpose of this work is to introduce digital image filtering and demonstrate elementary procedures used for their implementation.

Keywords

noise, color space, RGB, HLS, sharpening, edge, Laplacian operator, convolution, Unsharp masking, Sobel operator, Robinson operator, Kirsch operator, Canny edge detector, blurring, avraging, Bilateral filter, Gaussian blur, median filter, conservative filter

Citace

Jiří Kubínek: Techniky používané pro ostření a rozmazávání obrazu, bakalářská práce, Brno, FIT VUT v Brně, 2007

Techniky používané pro ostření a rozmazávání obrazu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jiřího Venery. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jiří Kubínek
Datum

Poděkování

Tímto bych rád poděkoval vedoucímu mé práce, panu Ing. Jiřímu Venerovi za vedení mé práce, cenné rady a podnětné návrhy, které mi vždy pomohly k pochopení problematiky a navedly mne na správnou cestu.

© Jiří Kubínek, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah 1
Úvod 2
1 Šumy 3
1.1 Vznik a dělení šumů 3
1.2 Odstranění šumů 3
2 Barevné prostory 4
2.1 RGB 4
2.2 HLS 5
3 Ostření obrazu 7
3.1 Úvodem 7
3.2 Laplaceův operátor 8
3.3 Unsharp masking 10
3.4 Další metody 13
4 Rozmazávání obrazu 14
4.1 Úvodem 14
4.2 Obyčejné průměrování 14
4.3 Bilaterální filtr 16
4.4 Další metody 17
5 Implementace 18
5.1 Laplaceův operátor 18
5.2 Unsharp masking 19
5.3 Obyčejné průměrování 19
6 Časová náročnost 20
Závěr 21
Literatura 22
Seznam příloh 23

Úvod

Úpravy digitálního obrazu patří mezi užitečné a často využívané operace. Může se jednat o zvýraznění detailů na fotografii či potlačení šumů vzniklých nedokonalou snímací soustavou. Tyto metody však najdou své uplatnění také jako předstupeň algoritmů pro zpracování obrazu, např. pro detekci pohybu, tváří či poznávacích značek vozů. Smyslem této práce je implementace algoritmů, které nám umožní tyto operace provádět, a to nezávisle na použitém programovacím jazyce či vývojovém prostředí – tedy ve formě knihovny.

V první kapitole bude vysvětlena problematika šumů jako nežádoucí součásti obrazu, jejich dělení a možnosti jeho odstranění. Druhá kapitola si klade za cíl vysvětlit pojem barevný prostor a dva v této práci použité v ní popsat. Jsou to asi nejznámější prostor RGB a praktický prostor HLS. Třetí kapitola charakterizuje metody užívané k ostření obrazu, věnuje se především Laplaceově operátoru a metodě nazvané Unsharp masking. Následující kapitola je věnována rozmazávání obrazu, opět se věnuje především dvěma metodám, obyčejnému průměrování a bilaterálnímu filtru. Kapitola pátá s názvem Implementace popisuje algoritmy použité pro implementaci filtrů. Poslední kapitola se věnuje použitým algoritmům z hlediska časové složitosti.

1 Šumy

Jedním z důvodů, proč provádíme úpravy uvedené v dalších kapitolách, je odstranění šumů. Jak vznikají, jak je dělíme a jak je můžeme potlačit? Odpovědi na tyto otázky by měla poskytnout tato kapitola [1].

1.1 Vznik a dělení šumů

Šum obvykle definujeme jako novou informaci, která byla přidána během pořízení či během transportu obrazu. Může být buď aditivní, nebo multiplikativní v závislosti na způsobu, jakým byl do obrazu přidán. Takto vzniklý šum je svázán s původní informací. Abychom jej mohli oddělit, potřebujeme znát původní funkci, což však není častý případ. Nejčastěji máme obraz definovaný jako množinu bodů, nikoliv jako spojitou funkci. Algoritmy sloužící k odstranění šumu s ním nakládají jako s jakoukoliv jinou vysokofrekvenční informací, což kromě šumu může být také hrana či textura s velkým rozdílem hodnot blízkých bodů.

Pro popis šumů se využívá teorie pravděpodobnosti. Pokud je pravděpodobnost výskytu šumu pro všechny frekvence stejná, jedná se o bílý šum. Ideální bílý šum má ve frekvenčním spektru zastoupeny všechny frekvence. Za bílý šum však považujeme i funkce frekvenčně omezené. Gaussův šum má pravděpodobnost výskytu danou Gaussovým rozložením. Poslední typ, který zde zmíním, je impulsní šum. Ten se vyznačuje vysokou energií a krátkým trváním. V počítačové grafice se objevuje zejména jeho varianta nazvaná sůl a pepř. Jejím projevem je náhodná změna intenzity některých bodů na bílou nebo černou barvu.

1.2 Odstranění šumů

Volba správného postupu pro redukci či odstranění šumu je závislá na jeho charakteristice a na způsobu, jakým byl obraz pořízen. Nejjednodušším způsobem je opakované pořízení obrazu a porovnání hodnot bodů na stejných souřadnicích. K tomu však potřebujeme znát původní funkci a navíc pořizovací zařízení nesmí trpět systémovou chybou. Příliš odlišná hodnota od ostatních s velkou pravděpodobností znamená chybu. Postupů, jak vypočítat novou hodnotu, se nabízí hned několik. Můžeme volit průměr hodnot, nejčastější hodnotu z množiny nebo použít medián. Tyto postupy jsou však spíše výjimkou, obvykle máme k dispozici pouze jediný exemplář a k odstranění šumu musíme volit algoritmy pracující na základě charakteristiky okolí každého bodu. Těm se věnuje samostatná kapitola nazvaná Rozmazávání obrazu.

2 Barevné prostory

Pro pochopení dále popisovaných algoritmů je nezbytné mít základní znalosti o barevných prostorech. Tato kapitola se bude věnovat dvěma, které jsou použity v práci: RGB a HLS. Ostatní barevné prostory vynechám, jsou však popsány například v [1] nebo [2].

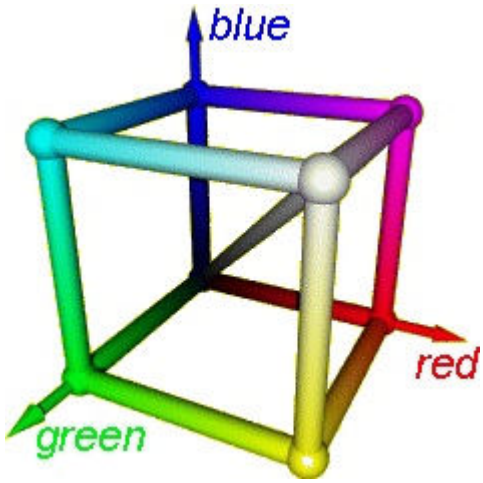
2.1 RGB

Jakákoliv barva zobrazitelná například na barevném monitoru počítače je tvořena kombinací několika základních složek. Barevný prostor nám říká jakých složek, kolik jich je a jakým způsobem se z nich tvoří výsledná barva. Na již zmíněném barevném monitoru vidíme výslednou barvu jako složení tří základních složek. Jsou to červená (R, red), zelená (G, green) a modrá (B, blue), v závorce používané písmeno vždy vycházející z anglického názvu barvy. Každá výsledná barva je tvořena vektorem tří hodnot udávajících intensitu jednotlivých složek. Celkový počet zobrazitelných barev je odvozen od rozsahu intensit. Běžně používané kódování využívá jednoho bajtu na každou složku, tedy nabývá hodnot 0 – 255. Hodnota 0 značí nulové zastoupení dané složky ve výsledné barvě, hodnota 255 pak maximální intensitu. Barva je takto určena třemi bajty a počet zobrazitelných barev je roven 256^3 = 16 777 216. Samozřejmě se využívá i jiných kódování, například 12 nebo 16 bitů na složku, přičemž jednotlivé složky nemusí být kódovány stejným počtem bitů (např. RGB 565).

Jak již z předchozího odstavce vyplývá, složíme-li barvu ze složek maximálních intensit, získáme bílou barvu. Podobně získáme barvu černou, a to složením nulových intensit. A jakýkoliv jiný odstín šedi lze získat složením intensit shodných hodnot. Pokud bychom však potřebovali převést barevný obraz na černobílý, nemůžeme použít prostého průměru tří základních barev. Lidské oko je odlišně citlivé na intensitu jednotlivých barevných složek. Pro výpočet jasu, a tedy i hodnoty bodu v černobílém obrazu ze tří složek využíváme následující vztah

$$I = 0.299*R + 0.587*G + 0.114*B. \tag{2.1}$$

Barevný rozsah prostoru RGB lze vyobrazit prostorově jako jednotkovou krychli umístěnou v souřadné soustavě s osami r, g a b (obr 2.1). Počátek souřadnic odpovídá černé barvě, bod [1,1,1] barvě bílé. Vrcholy krychle ležící na osách odpovídají základním barvám, ostatní vrcholy reprezentují doplňkové barvy k základním.

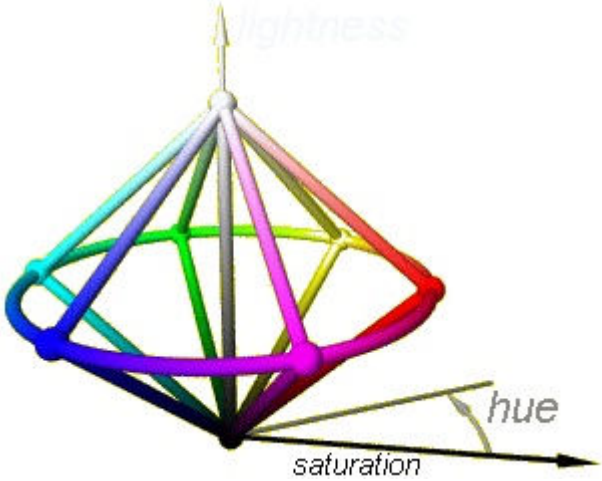


Obrázek 2.1: Barevný prostor RGB

Popis barev pomocí tří složek R, G a B je dán technickým provedením monitorů, ve kterých se výsledná barva získává sčítáním jednotlivých složek. Jeho základní vlastností je součtové, nebo-li aditivní skládání barev. Čím vyšší jsou hodnoty složek, tím světlejší je bod. Opačným případem je subtraktivní skládání barev, které vychází z praxe. Přidáváním barev se odstín stává tmavším. Kombinace maximálních intensit tak dává černou barvu a podobně bílou získáme kombinací nulových intensit. Tohoto skládání barev využívá například barevný prostor CMYK.

2.2 HLS

I tento prostor definuje barvu pomocí tří složek. Nejsou to však základní barvy, jako v případě prostoru RGB, ale barevný tón (H, hue), světlost (L, lightness) a sytost (S, saturation). V prostoru je znázorněn dvojicí kuželů (obr. 2.2). Barevný tón reprezentuje úhel s rozsahem 0° - 360°, světlost nabývá hodnot 0 (černá) až 1 (bílá) a sytost je rovna jedné na povrchu kuželů a klesá směrem k ose.



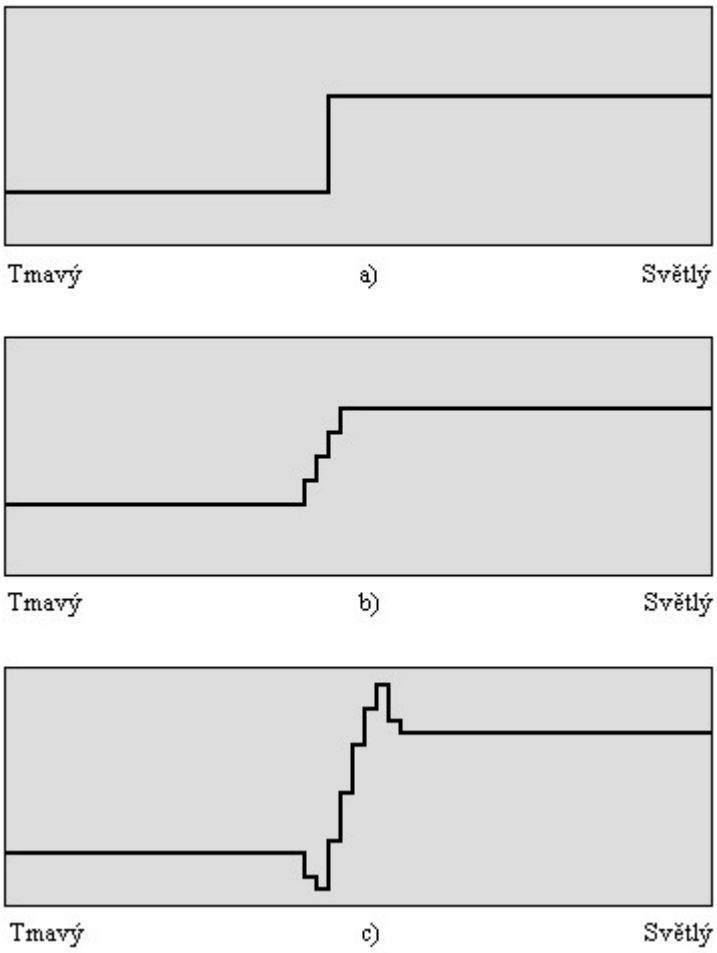
Obrázek 2.2: Barevný prostor HLS

Tvar prostoru HLS odpovídá skutečnosti, že nejvíce odstínů barev jsme schopni rozlišit při střední světlosti. Při nadměrné i nedostatečné světlosti naše rozlišovací schopnost klesá. Další výhodou je možnost měnit barevné charakteristiky, aniž bychom tím narušili jiné vlastnosti barvy. Prostor HLS se používá také jako vzorník barev pomocí řezu kužely s různou hodnotou barevného tónu. Pro úplnost je vhodné doplnit, že pro barvy ležící na ose kuželů není hodnota barevného tónu jednoznačně určena. Pro tyto případy je zavedena hodnota nedefinováno. Ta se také využívá při velikosti s = 0, kdy složka h musí být nedefinována.

3 Ostření obrazu

3.1 Úvodem

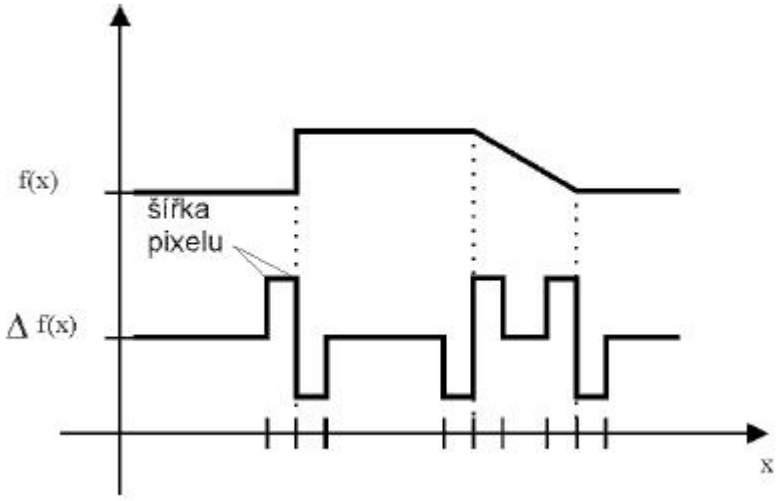
Lidské oko je citlivé na hrany. Rozpoznávání osob a věcí v obraze je založeno na detekci hran. Na základě tohoto poznatku se nám nabízí možnost, jak zajistit, abychom vnímali obraz jako ostřejší. Tou je zvýraznit hrany v něm obsažené. Hrana je místo, kde se výrazně mění hodnota sousedních bodů. Hranu lze také chápat jako vysokofrekvenční informaci. Jejím zvýrazněním dochází k ostření, ale také k zvýraznění případného šumu, je to tedy operace inverzní k odstranění šumu. Zjednodušeně lze říci, že ostření obrazu způsobuje světlou část hrany světlejší a tmavší část ještě tmavší (obr. 3.1). Na prvním obrázku lze vidět hranu tak, jak ji vnímá oko. Na druhém obrázku je tatáž hrana zachycena digitálním snímacím zařízením. A konečně na třetím je zobrazena po provedeném ostření. Další informace vztahující se k této kapitole lze nalézt například v [1], [3], [4].



Obrázek 3.1: Zvýraznění hrany

3.2 Laplaceův operátor

Tato metoda aproximuje druhou derivaci obrazu [1]. Její použití má za následek zvýraznění oblastí s velkými změnami intensit a používá se tak často pro detekci hran. Jak již bylo uvedeno, je proto vhodná pro náš účel. Podobně jako jiné metody pro ostření je však náchylná na přítomnost šumů v obraze. Proto může být někdy vhodné před její aplikací použít nějakou metodu rozmazávání a šum potlačit. Druhou možností je použít větší konvoluční jádro. Hrana může mít různou šířku. Na obr. 3.2 je znázorněna reakce laplaceova operátoru na různé typy hran.



Obrázek 3.2: Úzká a ostrá pozitivní hrana, široká negativní hrana a reakce

V našem případě, kdy máme obraz definovaný jako množinu bodů, musíme najít diskrétní konvoluční jádro, které bude aproximovat druhou derivaci obrazu. Dvě často používaná jádra (obr. 3.3) pro čtyřokolí a osmiokolí jsou použity i v mé práci. Jak lze z jejich podoby vidět, jsou invariantní k otáčení o násobky 45°.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Obrázek 3.3: Dvě běžně používané diskrétní aproximace Laplaceova filtru

Protože je pro pochopení principu Laplaceova operátoru potřebná jen diskrétní konvoluce, definici spojité konvoluce zde vynechám. Vzorec diskrétní konvoluce (označované hvězdičkou) má tvar:

$$f(x,y) * h(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(x-i, y-j) \cdot h(i,j) \tag{3.1}$$

Ve vyobrazeném vztahu je f(x,y) dvourozměrné pole vstupního obrazu a h(x,y) je konvoluční jádro. Pro výpočet si lze představit jádro jako tabulku, kterou přiložíme na příslušné místo obrazu. Každý bod překrytý tabulkou vynásobíme hodnotou jádra na shodné pozici. Následně tyto hodnoty sečteme a získáme novou hodnotu bodu. Tento postup aplikujeme pro všechny body vstupního obrazu a máme vypočtenou konvoluci. Hodnoty v jednotlivých buňkách jádra udávají, jaký vliv má bod pod nimi. Pokud tedy například chceme získat průměr devíti bodů (což, jak je uvedeno v následující kapitole, je jedna z metod rozmazávání), stačí provést konvoluci s jádrem o velikosti 3x3 body s hodnotou 1/9 ve všech devíti buňkách.

Samotné ostření probíhá ve dvou krocích. Pokud je obraz barevný, rozdělí se na tři kanály podle RGB barev, jedna barva na jeden kanál. Pokud je obraz černobílý, zůstává beze změn. Poté se aplikuje konvoluce. V případě černobílého obrazu přímo na něj, je-li obraz barevný, aplikuje se konvoluce na každý kanál samostatně. Výsledek konvoluce se přičte k původnímu obrazu a případně se tři kanály spojí zpět v jeden RGB obraz. Aby nedošlo ke ztrátě informace, provádí se tento proces na větším datovém rozsahu.

Druhý krok je volitelný a spočívá v úpravě histogramu. Cílem je roztáhnutí rozsahu u zostřeného obrazu do hranic. V případě černobílého obrazu je postup snadný. V množině bodů zostřeného obrazu se naleznou extrémy, tedy minimum a maximum. Na jejich základě se roztáhnou hodnoty bodů v obraze tak, aby vyplnily celý rozsah. U barevného obrazu se pro úpravu využívá prostoru HLS. Postup je uveden v kapitole 5.

Obrázek 3.4: Originál obrazu

Obrázky 3.5 a 3.6 prezentují výsledky této metody. Fotografie na obrázku 3.4 je originál pořízený digitálním fotoaparátem.

Obrázek 3.5: Výsledek ostření s použitím jádra pro čtyřokolí

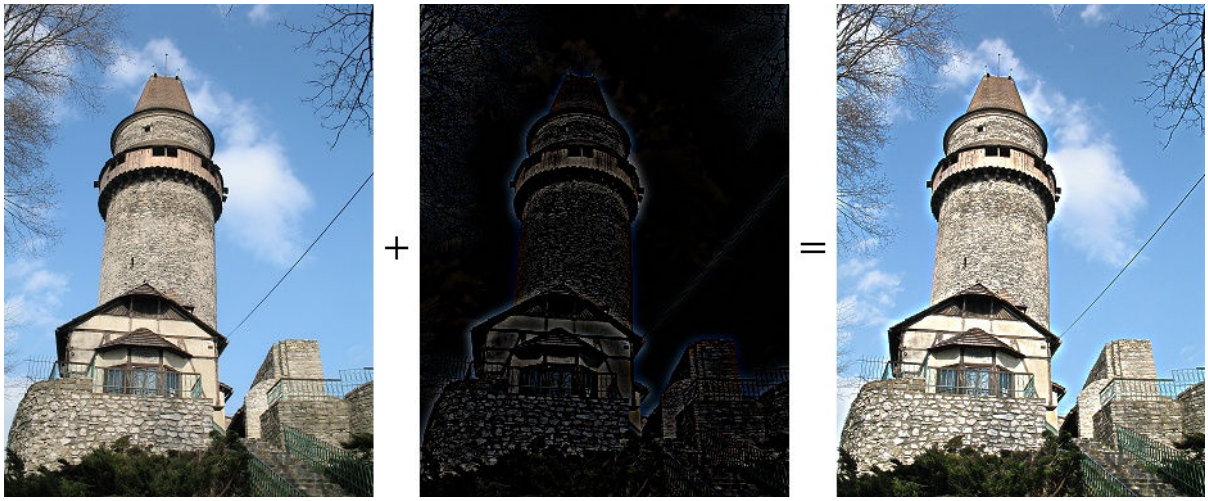
Na obrázku 3.5 lze pozorovat výsledek laplaceova ostření s použitím konvolučního jádra pro čtyřokolí (viz obr. 3.2.2). Použití stejné metody, avšak s konvolučním jádrem pro osmiokolí je znázorněno na obrázku 3.6

Obrázek 3.6: Výsledek ostření s použitím jádra pro osmiokolí

3.3 Unsharp masking

Požití této metody sahá ještě do dob, kdy editace fotografií na osobních počítačích neexistovala a fotografie se vyvolávaly v temných komorách. Postup byl takový, že fotograf na základě negativu zhotovil další, uměle rozmazaný. Následně složil rozmazaný negativ s originálem a zhotovil fotografii se zvýšenou ostrostí.

Tato metoda postupuje v zásadě stejně, jen digitální cestou, tedy rychleji a levněji [4]. Jakým způsobem tato metoda docílí vyšší ostrosti? Na rozdíl od předchozí metody Laplaceova operátoru nedetekuje hrany, ale oblasti s vysokým kontrastem. K tomu využívá popsaného principu. Vytvoří kopii obrazu a použitím některé z metod (často používanou je Gaussův filtr) ji rozmaže. Následně pro každý bod obrazu spočte rozdíl mezi původním a rozmazaným obrazem a tento rozdíl přičte do výsledného obrazu. V případě barevného obrazu se tento postup aplikuje pro každý kanál zvlášť. Princip metody je znázorněn na obrázku 3.7. První obrázek je fotografie zachycena digitálním fotoaparátem. Uprostřed je maska vytvořená odečtením rozmazaného obrázku (pomocí Gaussova filtru s jádrem o velikosti 25x25) a poslední obrázek je výsledek jako součet originálu s maskou [4].



Obrázek 3.7: Princip metody Unsharp masking, originál (vlevo) se sečte s maskou (Gaussův filtr s jádrem 25x25, uprostřed), výsledkem je zaostřený obraz (vpravo)

V praxi mívá tento algoritmus implementovaný v nejznámějších programech pro úpravu obrazu 3 parametry. Jsou to míra (amount), poloměr (radius) a práh (treshold). První parametr definuje sílu doostření, udává se v procentech. Poloměr definuje počet bodů, které budou doostřeny okolo nalezené hrany a konečně práh určuje, jaký musí být minimální rozdíl hodnot mezi originálním a rozmazaným bodem, aby byl tento bod doostřen. Pokud je rozdíl menší než zadaný, bod se ponechá beze změny. Můžeme ho použít třeba v případě, kdy se chceme vyhnout zesílení šumu v obraze.

Obrázek 3.8: Výsledek ostření s použitím Gaussova filtru s jádrem o velikost 3x3

Efekt této metody můžeme vidět na obrázku 3.8 a 3.9. Originál (obr. 3.4) je stejný, jako v případě Laplaceova ostření, můžeme tedy porovnat výsledky. Horní obrázek je ostřený za pomocí Gaussova filtru s jádrem o velikosti 9x9 (obr. 3.8), spodní s jádrem o velikosti 25x25 (obr. 3.9).

Obrázek 3.9: Výsledek ostření s použitím Gaussova filtru s jádrem o velikost 25x25

3.4 Další metody

V předcházejících kapitolách byly podrobněji popsány 2 metody ostření, které jsou implementovány v práci. Samozřejmě však nejsou jediné a v této kapitole stručně zmíním několik dalších, které by neměly zůstat opomenuty [1]. Podobně jako 2 již představené, i tyto metody slouží k detekci hran a je tak možné je použít k ostření obrazu.

Stejně jako Laplaceův operátor je na konvoluci založen také Sobelův operátor. Na rozdíl od něj však aproximuje první derivaci a je směrově orientovaný. Skládá se vždy z dvojice komplementárních konvolučních masek. Komplementární maska se získá z původní masky rotací o devadesát stupňů kolem středu. Na obrázku 3.10 vlevo je vyobrazena varianta konvolučního jádra pro detekci hran ve vodorovném směru. Pro další směry se jádro získá rotací o 90°. Dále z této skupiny metod už jen zmíním Robinsonův a Kirschův operátor. Příslušná konvoluční jádra jsou na obrázku 3.7.

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{pmatrix}$$

Obrázek 3.10: Konvoluční jádra Sobelova, Robinsonova a Kirschova operátoru (zleva)

V současnosti asi nejlepší hranový detektor je Cannyho. Jeho cílem je splnění tří kritérií. Detekční kritérium požaduje, aby nebyly přehlédnuty žádné významné hrany a aby na žádnou z nich nebyla vícenásobná odezva. Lokalizační kritérium požaduje, aby rozdíl mezi skutečnou a detekovanou polohou hrany byl minimální. A konečně požadavek jedné odezvy zajišťuje, aby detektor nereagoval na jednu hranu vícenásobně. Částečně se tedy shoduje s prvním kritériem, zaměřuje se však zejména na nehladké a zašumělé hrany, což první požadavek nesplňuje. Doporučený postup lze shrnout do několika bodů. Nejprve je zapotřebí eliminovat šum, například pomocí Gaussova filtru. Následně se v obraze detekují hrany, jako nejvhodnější způsob se jeví použití Sobelova operátoru, který nám vrátí nejen velikost, ale i směr gradientů. Třetím krokem je ztenčení. Zde hledáme lokální maxima z hodnot získaných v předešlém kroku. Cílem tohoto kroku je odebrat body, které nejsou maximem. Poslední krok je nazván eliminace nevýznamných hran. Již známe polohu hran, ale neznáme jejich význam. Máme označeny všechny hrany, protože i ty nejmenší mají své lokální maximum. Provedeme tedy prahování. Není vhodné určit jediný globální práh, nad kterým budeme považovat gradient za významný, protože hodnota může například vlivem šumu kolísat. Určíme tedy minimální a maximální práh. Pokud je hodnota gradientu daného bodu větší, než maximální práh, je rovnou označen za hranový. Pokud hodnota leží mezi těmito prahy, je označen za hranový pouze tehdy, sousedí-li s bodem již označeným za hranový.

4 Rozmazávání obrazu

4.1 Úvodem

Základní informace o šumech jsou obsaženy v první kapitole. Co je to šum, jak ho dělíme, jakým způsobem vzniká a proč se jej snažíme potlačit již tedy víme. Přejdu tedy rovnou k popisu některých metod. Jejich společným rysem je práce nad jediným obrazem, kdy nemáme k dispozici původní funkci. Odstraňování šumu v takovém případě se nazývá filtrace obrazu. Podrobnější informace jsou k nalezení například v [1].

4.2 Obyčejné průměrování

Jedná se zřejmě o vůbec nejjednodušší metodu. Hodnota nového bodu se spočítá jako průměr z jeho okolí [1]. Výsledkem je potlačení vysokých frekvencí ze spektra. Ostré změny v obraze, například hrany či šum, se rozmažou. Opakovanou aplikací filtru je rozmazání silnější. Limitně vede tato operace k jednobarevnému obrazu, jehož barva je průměrem ze všech hodnot v obraze. Pro výpočet se opět využívá konvoluce, kdy velikost konvolučního jádra je volitelná (jedná se o parametr metody, s narůstajícími rozměry jádra se zesiluje efekt rozmazání) a všechny hodnoty jsou rovny jedné. Rovnice pro výpočet za použití jádra o velikosti 3x3:

$$p(i,j) = \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f(i+k, j+l) \tag{4.2}$$

Pro ukázku je fotografie pořízena digitálním fotoaparátem uměle zatížena šumem (obr. 4.1).

Obrázek 4.1: Originál obrazu s uměle přidaným šumem

Obrázek 4.2: Obraz po aplikaci obyčejného průměrování s jádrem o velikosti 3x3

Na obrázku 4.2 je fotografie po aplikaci průměrování za pomocí jádra o velikosti 3x3. Obrázek 4.3 je výsledkem průměrování za pomocí jádra o velikosti 7x7. Jak je z obrázků patrné, zvláště s použitím většího jádra se šum vytrácí, avšak za cenu poměrně výrazné ztráty detailu.

Obrázek 4.3: Obraz po aplikaci obyčejného průměrování s jádrem o velikosti 7x7

4.3 Bilaterální filtr

Tato poměrně nedávno publikovaná metoda (Tomasi a Manduchi, 1998) [5] vykazuje velmi dobré výsledky na úkor velké časové složitosti. Potlačuje šum na souvislých plochách a přitom zachovává ostré hrany. Jedná se o nelineární filtr. Narozdíl od filtrů lineárních tedy nelze kombinovat více filtrů do jednoho konvolučního jádra a provést operace najednou. Dalším rozdílem je, že záleží na pořadí jejich provedení. Za určitých podmínek lze také změny provedené lineárními filtry vrátit zpět pomocí vhodného inverzního konvolučního jádra.

Pro pochopení principu bilaterálního filtru je nutné rozlišit dva pojmy. Dva body mohou být blízké, což znamená, že leží v obrazu blízko sebe. Pokud je označíme za obdobné, máme tím na mysli, že mají obdobnou hodnotu. Filtr nahrazuje bod na každé pozici průměrem z hodnot blízkých a obdobných bodů. Ve vyhlazených oblastech jsou hodnoty bodů v malém okolí obdobné, a filtr pracuje obdobně jako jiné filtry. Upravuje rozdíly mezi hodnotami jednotlivých bodů. Jakmile se ale filtr zaměří na řekněme světlý bod ležící na ostré hraně, započte do průměru pouze světlé body z okolí, tmavé ignoruje. Podobně, pokud se zaměří na tmavý bod, ignorují se v průměru světlé body. Díky tomu se filtrují souvislé plochy a zároveň se zachovávají ostré hrany. Tento filtr tedy uvažuje jak prostorové vlastnosti (tedy blízkost ve smyslu barevného prostoru) tak i frekvenční vlastnosti sousedních bodů, čili pracuje v obou oblastech a od tohoto je odvozen název této techniky bilaterální.

Je vhodné poznamenat, že na rozdíl od většiny filtrů je nutný odlišný přístup v případě barevných obrazů. Nestačí jen zpracovat samostatně tři kanály, protože by docházelo k barevným přechodům. Například na hranici modré a červené se tvoří fialový přechod, který by se vlivem Bilaterálního filtru ještě zesílil. Obraz je tedy nutné převést do barevného prostoru CIE-LAB. Na základě vlastností tohoto prostoru se jeví bilaterální filtrování v tomto prostoru jako nepřirozenější filtrování barevných obrazů. Průměrovány jsou pouze barvy pro lidské vnímání obdobné, zachovány jsou pouze důležité hrany.

Obrázek 4.4: Ukázka bilaterálního filtru, originál (vlevo), po filtraci (vpravo)

Obrázek 4.4 ilustruje výsledek bilaterálního filtru. Ztráta textury je zřetelná, přesto jsou hrany stejně ostré, jako u originálu. Podrobnější informace lze získat například v [5].

4.4 Další metody

Podobně jako u ostření i zde existuje řada dalších metod, které stojí za zmínku. Následuje alespoň stručný přehled několika z nich.

Již několikrát zmíněný Gaussův filtr je také variantou průměrování, avšak váženého. Konvoluční jádro je diskrétní aproximací Gaussova rozložení. Jedná se o hojně využívanou metodu pro potlačení šumu jako předstupeň dalších operací (například Unsharp masking). Filtr zohledňuje ve vyšší míře frekvenční vlastnosti obrazu.

Ve filtraci impulsního šumu vyniká mediánový filtr. Jeho výhodou je také lepší zachovávání hran oproti obyčejnému průměrování. Má však i své nevýhody. Není lineární, a navíc narušuje tenké čáry. Také je poměrně náročný na výpočetní výkon. Pro každý bod je nutné seřadit posloupnost hodnot, což je náročná operace. Je tak nutné volit vhodný algoritmus třídění. Metoda pracuje na bázi lokální statistiky okolí bodu. Hodnota nového bodu se spočítá jako medián z jeho okolí, nejedná se tedy o konvoluci.

Další metodou je konzervativní vyhlazování. Opět pracuje s okolím bodu. Tentokrát však nehledá medián z hodnot, ale hledá maximum a minimum. Následně je s nimi porovnán bod ve středu. Pokud leží jeho hodnota mezi minimem a maximem, zůstane nezměněna. Pokud je větší, než maximum, změní se na maximum. Podobně, je-li menší než minimum, nahradí se jeho hodnota minimem. Tento filtr v porovnání s obyčejným i mediánovým filtrem zachovává nejlépe hrany.

5 Implementace

Pro implementaci algoritmů jsem využil knihovny OpenCV (Open Source Computer Vision Library, http://www.intel.com/technology/computing/opencv). Tato knihovna byla pro můj účel velmi vhodná, protože obsahuje některé potřebné matematické operace a také umí načítat a ukládat obraz v mnoha formátech. Mohl jsem se tak soustředit na implementaci algoritmů na již připravených datech. Knihovna je napsána v jazyce C, měla by být přenositelná na jakoukoliv platformu podporovanou OpenCV. Při návrhu algoritmů byl kladen důraz na správnost, přesnost ale také na co nejnižší časovou náročnost. Nakolik se tyto cíle podařilo splnit mohou odpovědět kapitoly 3 a 4 s ukázkami výsledku a kapitola 6 obsahující zhodnocení časové náročnosti implementovaných algoritmů.

Veškeré algoritmy podporují černobílé i barevné obrazy. Implementována je i podpora pro COI a ROI. Obě jsou to vlastnosti obrazu v knihovně OpenCV, COI (channel of interest) určuje aktivní barevný kanál, lze tak nastavit jediný kanál, na kterém se mají úpravy provádět, ostatní kanály zůstanou netknuty. Smysl má samozřejmě pouze v případě barevného obrazu. Vlastnost ROI (area of interest) určuje aktivní oblast. Operace nad obrazem budou v případě nastavení této vlastnosti prováděny pouze ve vymezené oblasti.

5.1 Laplaceův operátor

Při návrhu této metody jsem vycházel z postupů uvedených v [1] a [3]. Při zavolání této funkce se zjistí, zda je nastaven COI. Pokud nastaven je, zavolá se funkce znovu s obrazem obsahující pouze aktivní kanál a zaručí se tak změna pouze tohoto kanálu. Následně se připraví konvoluční jádro. Na základě parametrů může být použito jedno ze dvou předvolených, nebo může být dodáno vlastní jádro. Následuje nezbytná konverze datového rozsahu na datový typ float. Poté už následuje konvoluce. Získáme tak pro každý kanál masku, kterou později přičteme. Ještě před tím se však hodnoty masky upraví do rozsahu 8 bitů, přičemž jsou povoleny i záporné hodnoty. Při přičítaní kontrolujeme hranice, výsledná hodnota nesmí nabývat záporných hodnot ani hodnot větších než je 255. Posledním krokem je spojení kanálů a konverze do původního datového rozsahu.

Volitelným krokem je roztáhnutí histogramu. Pokud má obraz jediný kanál, jedná se o prosté roztáhnutí hodnot do rozsahu 0..255. Pokud je obraz v RGB, převede se nejdříve do prostoru HLS. Následně se roztáhne kanál L do rozsahu 0..255 a obraz se převede zpět do RGB. Tato operace, pokud je zvolena, se provádí po spojení kanálů a před konverzí do původního datového rozsahu.

Je-li zvoleno jedno z předvolených konvolučních jader, je to jediný parametr funkce. Pokud požadujeme použití vlastního jádra, je nutné poskytnout tři parametry. Dva z nich představují rozměry jádra a třetí parametr je pole hodnot konvolučního jádra.

5.2 Unsharp masking

Implementace této metody je poměrně jednoduchá. Pokud v obrazu není nastaven COI, aplikuje se na něj v prvním kroku Gaussův filtr s parametry získanými při volání funkce. Následně se odečte originální obraz od rozmazaného (bod po bodu v rámci celého obrazu). Posledním krokem je přičtení takto vzniklé masky k originálu (opět bod po bodu). Sílu ostření tak určují dva parametry ovlivňující sílu Gaussova filtru. Oba tyto parametry musí být liché.

Postup pro obraz s nastaveným COI je obdobný. Vytvoří se kopie obrazu obsahující pouze vybraný kanál. S tímto obrazem se provedou operace popsané v předcházejícím odstavci a upravený aktivní kanál se zkopíruje zpět do obrazu.

V algoritmu není implementováno prahování, což by bylo jedním z vhodných rozšíření mé práce. S touto vlastností by bylo možné nastavit minimální rozdíl hodnot mezi originálním a rozmazaným obrazem, aby se provedl součet. Je to jedna z možností, jak zabránit zvýrazňování šumu.

5.3 Obyčejné průměrování

Nejjednodušším přístupem pro implementaci průměrování získáme jednoduchý algoritmus, který však bude nepřiměřeně náročný. Můžeme procházet všechny body a pro každý z nich spočítat součet hodnot bodů v jeho okolí a posléze tento součet podělit počtem bodů v okolí. Velikost okolí určují dva parametry funkce. Mohou být rozdílné, liché i sudé, pouze musí být kladné.

V mém algoritmu je však obsažena optimalizace. Pokud postupujeme po bodech na řádku, přibývá vždy jen jeden nový sloupec a jeden sloupec ze součtu mizí. Po počáteční inicializaci tak stačí pro každý bod na řádku cyklus s počtem průchodů dle výšky jádra. Pokud postupujeme body doprava, přičteme při každém průchodu bod z nového pravého sloupce a odečteme bod ze starého levého sloupce. Podobně by bylo možné optimalizovat přechod mezi řádky, přínos by již ovšem nebyl tak vysoký. Algoritmus tedy na každém řádku inicializuje součet od nuly a optimalizuje se až počínaje druhým bodem v řádku.

Existuje zde ještě jedna komplikace, a tou jsou hranice obrazu. Pokud se má například určit bod na souřadnicích [1,1] a spočítat průměr pro okolí 3x3 body, dostáváme se za hranice obrazu. Souřadnice je tak nutné hlídat, aby tyto hranice nepřekročily. Pokud se tak stane, započítává se do součtu hraniční bod. Podobně jako ostatní algoritmy, i tento v případě barevného obrazu prochází jednotlivé kanály zvlášť.

6 Časová náročnost

Poslední kapitola rozebere algoritmy z hlediska časové náročnosti. Ve všech případech je zdrojem dat barevná fotografie o rozlišení 1536x1152 bodů. Jedná se o fotografii pořízenou digitálním fotoaparátem. Její jedinou úpravou bylo zmenšení na poloviční velikost. Kopie je uložena na přiloženém CD. Testovací počítač má následující konfiguraci:

Model: NB Sony Vaio PCG-F104K

CPU: Pentium III 650MHz (jádro Coppermine)

RAM: 256MB DIMM 100MHz

OS: Microsoft Windows XP SP2

Měření délky trvání metod bylo realizováno rozdílem mezi časy spuštění dané funkce a návratem z ní. K tomuto měření bylo využito funkce clock() ze standardní knihovny time. Každé měření proběhlo 15x a do tabulky byly vloženy výsledky posledních 10 naměřených hodnot a jejich průměr. V tabulce 6.1 jsou uvedeny naměřené hodnoty pro každý algoritmus ve dvou variantách.

metoda / čas (ms)	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	průměr
LO - čtyřokolí	4526	4447	4436	4457	4436	4446	4437	4446	4446	4457	**4453**
LO - osmiokolí	5197	5188	5177	5188	5177	5177	5168	5187	5168	5197	**5182**
Unsharp 3x3	771	811	751	751	762	771	771	771	741	751	**765**
Unsharp 15x15	4816	4817	4827	4837	4837	4867	4857	4827	4827	4837	**4835**
Průměr 3x3	1743	1742	1743	1742	1803	1732	1743	1742	1743	1742	**1748**
Průměr 15x15	7591	7581	7581	7571	7591	7560	7561	7711	7591	7541	**7588**

Tabulka 6.1: Délka trvání implementovaných metod v milisekundách, LO - Laplaceův operátor, Unsharp – Unsharp masking, Průměr – obyčejné průměrování

Jak je z tabulky patrné, ve srovnání metod pro ostření vychází jako rychlejší Unsharp masking. Zejména s menším jádrem Gaussova filtru je metoda velmi rychlá. U metody obyčejného průměrování je zřetelná optimalizace. Náročnost stoupá lineárně (s 5x větším rozměrem stoupá potřebná doba k výpočtu přibližně 4.3x), a nikoliv kvadraticky, přestože počet hodnot konvolučního jádra kvadraticky roste. I tak je však doba potřebná k výpočtu vyšší, než v případě metody Unsharp masking, přestože by měla být menší. Určitě je zde tedy prostor pro další optimalizace. Naměřené výsledky jsou na dnešní dobu poměrně vysoké, což je však dáno již poměrně zastaralým testovacím počítačem. Na druhou stranu jsou alespoň lépe vidět rozdíly mezi metodami.

Závěr

V práci jsem se věnoval metodám pro filtraci obrazu. Seznámil jsem se s knihovnou OpenCV, která byla pro velkým přínosem. Obsahuje mnoho vestavěných funkcí, které mi umožňují věnovat veškeré své úsilí implementaci algoritmů, aniž by bylo nutné podrobně studovat obrazové formáty, kterých existuje mnoho. Knihovna sama zajišťuje načtení, zobrazení a uložení obrazu, mým úkolem tak jen bylo zpracovat připravené pole bodů. Veškerý programový kód jsem psal, ladil i testoval ve vývojovém prostředí Microsoft Visual Studio 2005 (www.microsoft.com). Přestože se nejednalo o mé první seznámení s tímto prostředím, naučil jsem se mnoho nového. Při své práci jsem ocenil zejména vestavěný systém ladění, který urychlil hledání chyb v mém kódu.

Ačkoliv je dle mého soudu knihovna plnohodnotně použitelná, pro její zlepšení existuje hned několik cest. Implementované metody pracují na základě mého testování správně, avšak určitě je zde prostor pro další optimalizace, ať už časové nebo prostorové. Další cestou je rozšířit nabídku metod. Především by bylo vhodné implementovat bilaterální filtrování pro barevné obrazy a její urychlené modifikace. Tato vykazuje výborné výsledky v porovnání výstupního obrazu s běžnými metodami rozmazávání (například obyčejné průměrování), na druhou stranu je podstatně časově náročnější. Dalším krokem by mohlo být vytvoření uživatelsky přívětivého prostředí, které by využívalo knihovny a pomocí intuitivních ovládacích prvků by uživateli umožnilo filtraci obrazu s průběžným náhledem. Naměřené výsledky v řádu sekund však tuto potřebu nemohou uspokojit, bylo by proto nutné zvýšit efektivitu některých algoritmů.

Literatura

[1] Žára, J., Beneš, B., Sochor, P., Felkel, P. *Moderní počítačová grafika*. Brno, Computer Press 2004.

[2] Doležal, J. *Základní barevné systémy* [on-line]. Katedra matematiky a deskriptivní geometrie. [květen 2007]. Dostupný z WWW: < http://mdg.vsb.cz/jdolezal/Pgrafika/Prednaska/ZaklBarevneSys.html >.

[3] Fanning, David W. *Image Sharpening with a Laplacian Kernel* [on-line]. Fanning Consulting. [květen 2007]. Dostupný z WWW: < http://www.dfanning.com/ip_tips/sharpen.html>.

[4] Bigelow, R. *Sharpening – Part I* [on-line]. Ron Bigelow Photography. [květen 2007]. Dostupný z WWW: <http://www.ronbigelow.com/articles/sharpen1/sharpen1.htm>.

[5] Manduchi, M. *Bilateral Filtering for Gray and Color Images* [on-line]. The University of Edinburgh, School of Iformatics. [květen 2007]. Dostupný z WWW: <http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html>.

Seznam příloh

Příloha 1. CD obsahující zdrojové texty knihovny, technickou zprávu v elektronické verzi, manuál ke knihovně a testová fotografie použitá k měření časové náročnosti