

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

CANNYHO OPERÁTOR A DALŠÍ POUŽÍVANÉ  
HRANOVÉ DETEKTORY

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

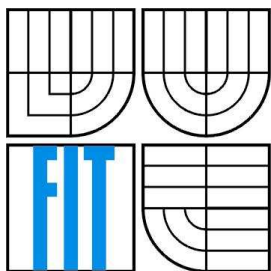
AUTOR PRÁCE  
AUTHOR

MILOŠ JANDA

BRNO 2008



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

# CANNYHO OPERÁTOR A DALŠÍ POUŽÍVANÉ HRANOVÉ DETEKTORY

CANNY'S OPERATOR AND OTHER USEFUL EDGE DETECTORS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MILOŠ JANDA

VEDOUCÍ PRÁCE

SUPERVISOR

ING. JIŘÍ VENERA

BRNO 2008

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Janda Miloš**

Obor: Informační technologie

Téma: **Cannyho operátor a další používané hranové detektory**

Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte problematiku předzpracování obrazu vhodnou pro následnou detekci hran.
2. Prostudujte algoritmy pro detekci hran, zaměřte se zejména na Cannyho operátor.
3. Implementujte některé algoritmy pro detekci hran (především Cannyho operátor) a jejich funkčnost demonstруйте na vhodných datech.
4. Prostudujte a implementujte algoritmy pro post-processing obrazu získaného aplikací hranových detektorů (prahování).
5. Porovnejte tyto algoritmy z hlediska časové složitosti, kvality detekce hran derivace a odolnosti proti šumu.
6. Srovnajte Vámi implementovaný přístup s jinými existujícími.

Literatura:

- Po domluvě na vyžádání u školitele.

Při obhajobě semestrální části projektu je požadováno:

- Body 1. a 2., započítí řešení bodu 3..

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 nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), 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 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
L.S.  
602 00 Brno, Rožstáňova 2



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: **Miloš Janda**  
Id studenta: 79710  
Bytem: Masarykova 79, 678 01 Blansko  
Narozen: 21. 07. 1986, Brno  
(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: Cannyho operátor a další používané hranové detektory  
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 tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....  
Nabyvatel

  
.....  
Autor

## **Abstrakt**

Tato práce poskytuje úvod do problematiky digitálního zpracování obrazu a definuje základní pojmy pro jeho úspěšné pochopení. Zabývá se několika vhodnými metodami předzpracování obrazu pro detekci hran, metodami pro detekci hran a následného zpracování těchto obrazů. Cílem práce je efektivní implementace a komplexní porovnání metod používaných pro detekci hran.

## **Klíčová slova**

šum, typy hran, konvoluce, průměrování, medián, Gaussův filtr, konzervativní vyhlazování, Robertsův operátor, Sobelův operátor, Prewittův operátor, Cannyho hranový detektor, Laplaceův operátor, statické prahování, hysterezní prahování, adaptivní prahování.

## **Abstract**

This work introduces main approaches for digital image processing and defines fundamental terms for successful understanding. Main aim is description of several suitable methods used in digital image pre-processing, methods for edge detection and consequent post-processing of these. The final goal of this work is effective implementation and complex comparison of methods for edge detection.

## **Keywords**

Noise, edge types, convolution, averaging, median, Gaussian filter, conservative smoothing, Roberts operator, Sobel operator, Prewitt operator, Canny edge detector, Laplacian operator, static thresholding, hysteresis thresholding, adaptive thresholding.

## **Citace**

Janda Miloš: Cannyho operátor a další používané hranové detektory. Brno, 2008, bakalářská práce, FIT VUT v Brně.

# Cannyho operátor a další používané hranové detektory

## 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.

.....  
Miloš Janda  
6. dubna 2008

## Poděkování

Tímto bych rád poděkoval vedoucímu práce za všechny informace poskytnuté při tvorbě mé práce,  
připomínky k vývoji programu a podmětné návrhy při celkovém řešení problémů.

© Miloš Janda, 2008.

*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
1 Úvod.....	2
2 Digitální zpracování obrazu .....	3
2.1 Digitalizace dat.....	3
2.2 Barevný prostor RGB.....	3
2.3 Barevný model CMY .....	4
2.4 Omezení barevného prostoru.....	5
3 Předzpracování obrazu .....	6
3.1 Co je to hrana .....	6
3.2 Vznik šumu a metody pro jeho odstranění.....	7
3.3 Obyčejné průměrování .....	8
3.4 Gaussův filtr .....	9
3.5 Medián.....	11
3.6 Konzervativní vyhlazování.....	11
4 Detekce hran v obraze.....	12
4.1 Výpočet gradientu ve dvou směrech .....	13
4.2 Výpočet gradientu ve více směrech.....	16
4.3 Operátor první derivace Gaussovské funkce.....	17
4.4 Laplaceův operátor.....	18
4.5 Laplacián Gausiánu (LoG).....	19
4.6 Diferenciál Gaussovské funkce (DoG).....	20
4.7 Cannyho hranový detektor .....	21
5 Post-processing obrazu .....	23
5.1 Globální prahování.....	23
5.2 Prahování s hysterezí.....	24
5.3 Adaptivní prahování.....	24
6 Implementace .....	26
6.1 Knihovna OpenCV .....	26
6.2 Implementované metody .....	26
7 Závěr .....	28
Literatura .....	29
Seznam příloh .....	30



# 1 Úvod

Se stále rostoucím rozvojem digitálních technologií v posledních letech, dochází k častějšímu využívání digitálního zpracování obrazu. Život bez používání digitálních fotoaparátů, digitálních kamer, digitální televize, scannerů a dalších digitálních zařízení si dnes asi ani nedovedeme představit. Digitální obraz má mnoho výhod, podobně jako má elektronický text mnoho výhod oproti psanému textu. Digitální obraz lze stejně jako elektronický text lehce upravovat, tisknout, sdílet a archivovat. V mnoha odvětvích, jako je průmysl, zdravotnictví, počítačová grafika i ve hrách a multimediích, se dnes úspěšně využívá počítačového zpracování obrazu.

Tato práce se zabývá jednou z částí zpracování obrazu a tím je problematika detekce hran v obraze. Jako příklad využití detekce hran v obraze může posloužit segmentace medicínských dat. Dále je to rozpoznávání objektů v obraze, které lze využít nejen v geografických systémech při analýze geografických snímků nebo map. Cílem práce je také vytvořit aplikaci v jazyce C, která bude provádět jednoduchou a efektivní detekci hran využitím několika různých metod pro detekci hran.

První úvodní kapitola je stručným uvedením do problematiky zpracování obrazu a detekce hran. Druhá kapitola je věnována teorii reprezentace digitálních dat, barevných modelům a omezení barevného prostoru. Třetí kapitola se zabývá reprezentací hran v obraze a předzpracováním obrazu pro následnou detekci hran. Čtvrtá stěžejní kapitola je rozbohem metod pro detekci hran a především Cannyho operátoru. Další kapitola je věnována převodu obrazu do binární podoby, tedy prahování. Následuje krátká kapitola o programovacím prostředí a implementovaných metodách. Poslední kapitolou je nezbytný závěr, kde jsou shrnuty všechny důležité poznatky této práce a možnosti dalšího vývoje.

## 2 Digitální zpracování obrazu

V této kapitole jsou uvedeny základní pojmy z oblasti počítačové grafiky, které dále poslouží pro lepší pochopení problematiky předzpracování obrazu a detekce hran v dalších kapitolách této práce.

### 2.1 Digitalizace dat

Hned na počátku můžeme rozdělit počítačovou grafiku na 2D a 3D počítačovou grafiku. Dále se dělí 2D grafika na grafiku rastrovou a grafiku vektorovou. V této práci se budeme zajímat pouze o 2D rastrovou grafiku, tedy budeme zpracovávat dvojrozměrný diskrétní obraz složený z jednotlivých bodů – pixelů.

Pokud tedy nemáme data již v této digitální podobě, tak dříve než začneme s jakoukoliv úpravou, je potřeba převést data do digitální podoby. K převodu slouží skenery, digitální fotoaparáty a kamery. Při převodu převádíme spojitý analogový signál do diskrétní podoby – signál je *vzorkován*. Vstupním analogovým signálem je spojitá funkce dvou proměnných – obrazová funkce [1]:

$$z = f(x, y) \quad (2.1)$$

Hodnotou obrazové funkce je typicky hodnota jasu nebo vektor intenzit základních barevných složek. Tento rozsah hodnot je nutno také převést do konečného počtu hodnot, které můžeme v počítačové architektuře uložit – hladina vstupního signálu je tedy *kvantována*.

Vzorkování a kvantování jsou kroky procesu zvaného *digitalizace*, tedy převodu teoreticky nekonečného analogového signálu s nekonečným počtem hodnot funkce do konečného rozměru s konečným počtem hodnot. Výsledkem je tedy matice pixelů, kde každý pixel obsahuje informaci o intenzitě každé ze základních složek barevného modelu RGB.

Pro detekci hran nám bude postačovat pouze šedotónový obraz, proto se dále podíváme na barevné modely používané v počítačové grafice a na omezení barevného prostoru.

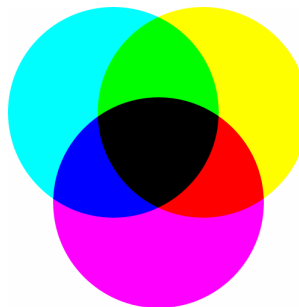
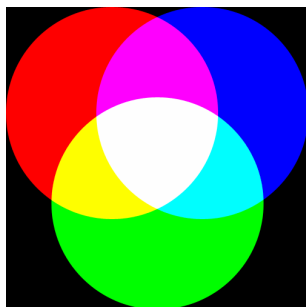
### 2.2 Barevný prostor RGB

Každý barevný prostor má několik základních barev, další barvy vznikají smícháním těchto základních barev. Podle způsobu míchání barev rozdělujeme barevné modely na modely s aditivním a subtraktivním mícháním barev (obr 2.1a, 2.1b). V této části si popíšeme barevný model RGB.

Barevný model RGB je nejběžnějším barevným modelem. Má tři základní barvy, kterými jsou červená (R, red), zelená (G, green) a modrá (B, blue). RGB model je aditivní, při smíchání maximální intenzity každé ze tří základních složek vznikne bílá barva, při použití nulových intenzit všech základních složek vznikne barva černá. Odstíny šedi lze získat smícháním stejných hodnot intenzit

základních složek. RGB model lze zakreslit jako jednotkovou krychli, kde každá ze tří os souřadného systému odpovídá jedné ze základních složek (obr 2.2).

Celkový počet barev je dán rozsahem hodnot intenzit. Běžně se používá 24 bitů na 1 pixel, tedy 8bitů na každou ze 3 složek, to umožňuje získat 256 hodnot (0-255) jasu každé složky. Celkově lze tedy zobrazit  $256^3 = 16\,777\,216$  odstínů barev.

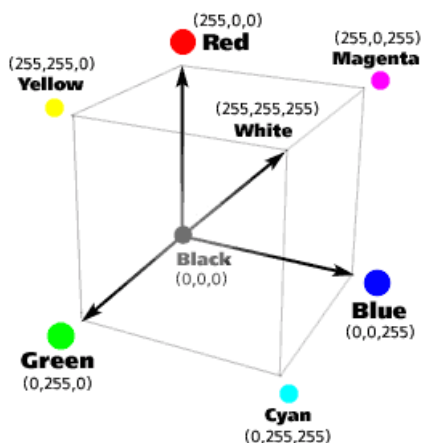


Obrázek 2.1a: Aditivní míchání barev

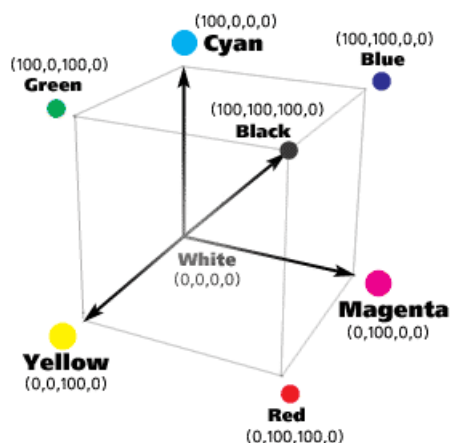
Obrázek 2.1b: Subtraktivní míchání barev

## 2.3 Barevný model CMY

CMY barevný model má také tři základní složky, kterými jsou azurová (C, cyan), purpurová (M, magenta) a žlutá (Y, yellow). Tento model využívá subtraktivního míchání barev, tedy opačného než model RGB. Zde při použití maximálních intenzit každé ze tří základních složek získáme černou barvu. Při smíchání nulových intenzit získáme barvu bílou. CMY model lze stejně jako RGB model zakreslit do jednotkové krychle viz. obr. 2.3. K modelu CMY se v praxi často přidává zvlášť černá barva (K, black) někdy označovaná jako (K, key), tím vznikne barevný model CMYK.



Obrázek 2.2: Barevný model RGB



Obrázek 2.3: Barevný model CMY(K)

Mezi další barevné prostory patří například modely HSV, HLS, YUV,  $Y_C B_C R_C$  a CIE. Více o těchto modelech lze nalézt v [1].

## 2.4 Omezení barevného prostoru

Jak již bylo uvedeno dříve, standardně se používá 24bitové kódování, kde lze získat přibližně celkem 16 milionů barev. Některá zařízení jako jsou mobilní telefony, projektory však tolik barevných odstínů nedokáží zobrazit, potřebujeme tedy omezit barvený prostor na méně barev. V některých případech jako je třeba detekce hran, je dokonce informace o barvě nedůležitá a stačí nám pouze informace o intenzitě jasu v jednotlivých bodech obrazu. Zde přichází na řadu převod do stupňů šedi, neboli omezení barevného prostoru.

Převod do odstínů šedi je realizován omezením původních tří barevných kanálů na obraz o jednom kanálu s rozsahem 256 stupňů šedi. Protože lidské oko není stejně citlivé na červenou, zelenou a modrou barvu, je při převodu do stupňů šedi (grayscale) použit empirický vztah viz. rovnice 2.2. Výsledná intenzita je získána jako součet jednotlivých složek, které jsou vynásobeny koeficienty  $r, g, b$ .

$$\begin{aligned} I &= r \cdot R + g \cdot G + b \cdot B \\ I &= 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \end{aligned} \quad (2.2)$$

Jak se z rovnice patrně nejvíce je zastoupena zelená složka a nejméně složka modrá, to vyplývá z faktu, že lidské oko je nejvíce citlivé na zelenou barvu a nejméně je citlivé na barvu modrou. Součet použitých konstant  $r, g, b$  se musí rovnat jedné.

Dalším typem omezení barevného prostoru je převod šedotónového (achromatického) obrazu na černobílý (monochromatický) neboli binární obraz. Tyto způsoby se liší podle účelu použití výsledného monochromatického obrazu. Existují zde dvě základní metody a těmi jsou:

- Rozptylování (dithering)
- Polotónování (halftoning)

Mezi metody rozptylování a polotónování patří náhodné rozptýlení, maticové rozptýlení, distribuce chyby a prahování (thresholding). Poslední jmenovanou metodu a její možné varianty si podrobně rozebereme v kapitole 5, která se zabývá závěrečnými úpravami obrazu po detekci hran.

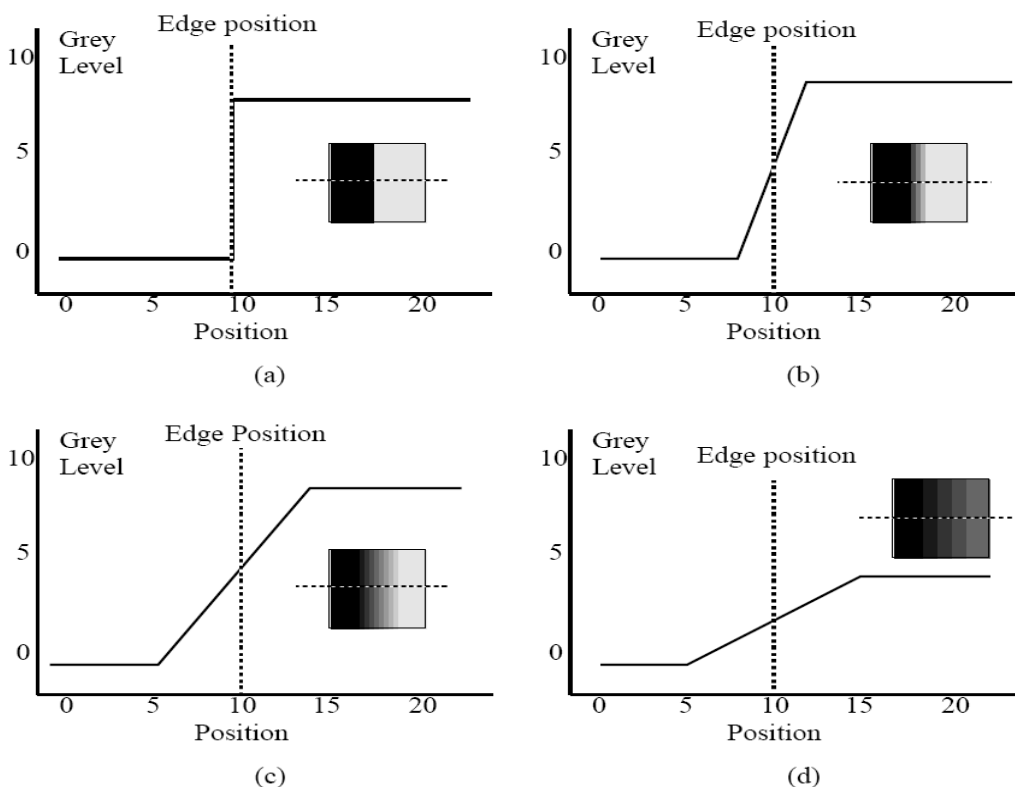
# 3 Předzpracování obrazu

Tato kapitola je věnována reprezentaci hran v obraze a předzpracování obrazu pro následnou detekci hran. Součástí předzpracování obrazu je odstranění šumu v obraze, proto v této kapitole budou probrány i jednotlivé druhy šumu a metody pro jejich odstranění.

## 3.1 Co je to hrana

Z pohledu fyzikálního je hrana vysokofrekvenční informace, například jako šum. Z matematického hlediska se hrana nachází v místech lokálního maxima derivace jasové funkce. V počítačové grafice se hrana vyjadřuje jako prudká změna jasu sousedních pixelů. Lidské oko detekuje hranu na okrajích (hranách) objektů nebo na hranici světla a stínu.

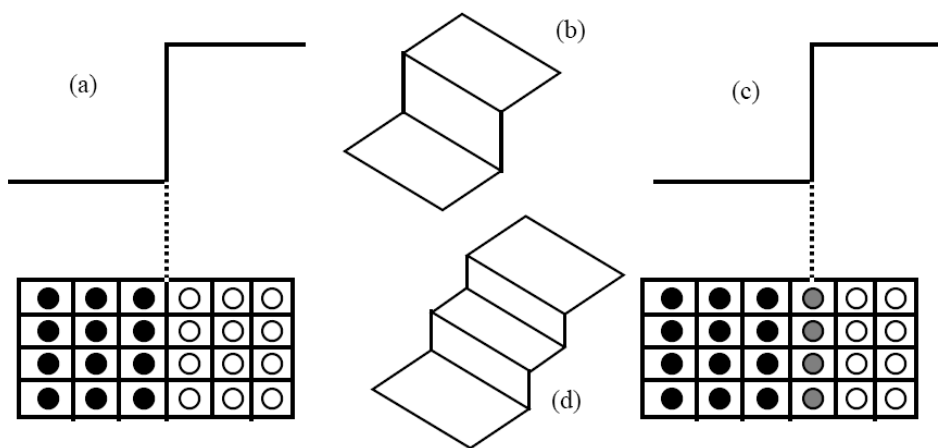
Ne vždy je však změna jasu sousedních pixelů tak prudká (skoková). V digitálním obraze se ideální skoková hrana vyskytuje jen zřídka, v obraze lze nalézt spíše šikmé hrany (ramp edges) viz obr. 3.1a-d.



Obrázek 3.1a-d: Šikmá (ramp) hrana.

Obrázek 3.1a znázorňuje ideální skokovou hranu. Ke změně jasu dochází přesně na pozici 10 pixelu. Na obrázku 3.1b stejná změna hodnoty jasu, v tomto případě ale změna probíhá přes 4 sousední pixely kolem středu na 10 pixelu. Schéma 3.2c ukazuje stejný případ, zde však změna probíhá v okruhu 10 pixelů. Poslední případ 3.1d znázorňuje poloviční změnu intenzity probíhající přes 10 pixelů, hrana se tak stává téměř nezřetelnou.

Příčinou vzniku šikmých hran je digitalizace obrazu a vznik šumu v obraze. Při procesu vzorkování ne všechny hrany připadnou přesně na okraj mezi dva sousední pixely (obr. 3.2a-b) a zde pak dojde ke schodovité změně hodnot jasu přes několik pixelů (obr. 3.2c-d).



Obrázek 3.2a-d: Rozdíl mezi ideální a schodovitou hranou vzniklou v důsledku vzorkování obrazu.

## 3.2 Vznik šumu a metody pro jeho odstranění

Šum je stejně jako hrana vysokofrekvenční informace. Podle způsobu jakým byl do obrazu přidán to může být šum aditivní nebo multiplikatívni. Šum lze přidat do obrazu i uměle, toho se ale příliš nevyužívá, snad jen pro účely demonstrace druhů šumů a metod pro odstranění šumu v obraze. V počítačové grafice rozlišujeme několik druhů šumů, kterými jsou:

- Bílý šum
- Gaussův šum
- Impulsní šum

První zmíněný bílý šum se vyznačuje tím, že má ve frekvenčním spektru rovnoměrně zastoupeny všechny frekvence. To ovšem platí jen pro ideální bílý šum, protože kdyby byl určitý výkon na všech frekvencích, celkový výkon takového signálu by byl nekonečný. Proto je v praxi za

bílý šum považován signál frekvenčně omezený. Gaussův šum je šum, který má pravděpodobnost výskytu danou Gaussovým (normálním) rozložením (viz rovnice 3.1). Posledním druhem je šum impulsní, ten se vyznačuje krátkou dobou trvání a vysokou frekvencí. Často bývá nazýván jako varianta sůl a pepř (salt and pepper). Impulsní šum způsobuje náhodnou změnu intenzity pixelů na černou nebo bílou barvu, odtud také pochází název této varianty.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.1)$$

Pro oddělení šumu v obraze je třeba znát původní funkci, kterou však ve většině případů neznáme. Obraz totiž máme většinou zadán jako množinu bodů, nikoli jako spojitou funkci. Další možností je použití více obrazů téže scény, kde jsou pak porovnány hodnoty pixelů v jednotlivých obrazech na stejných pozicích. Výslednou hodnotou pak může být průměr nebo medián z těchto hodnot.

V případě, že nemáme více obrazů stejné scény je nutné použít techniky, které analyzují okolí bodu a výslednou hodnotu aproximují. Podle způsobu výpočtu se rozlišuje několik druhů filtrů, které si zde později jednotlivě projdeme. Tento způsob odstranění šumu se spoléhá na skutečnost, že sousední pixely mají většinou stejnou nebo podobnou hodnotu jasu. Jelikož jde o potlačení vysokých frekvencí, výsledkem těchto filtrací obrazu je převážně rozmazání obrazu v místech ostrých změn jasu.

### 3.3 Obyčejné průměrování

První a nejjednodušší metodou je obyčejné průměrování, kdy výsledná hodnota je dána průměrem hodnot okolních pixelů. Jde tedy o konvoluci s konvoluční maskou  $h_1$ . Pro odstranění Gaussova šumu lze použít mírně pozměněné masky  $h_2$  a  $h_3$ , které zohledňují větší váhu pixelů blíže středu masky.

$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Výsledkem konvoluce je obraz, kde jsou potlačeny vysoké frekvence z frekvenčního spektra. Opakovanou aplikací nebo aplikací masky s větším rozměrem je efekt potlačení šumu silnější a dojde také k většímu rozmazání obrazu. Limitně vede opakované použití této techniky k jednobarevnému obrazu, jehož barva je dána průměrem hodnot ze všech pixelů obrazu. Velikost masky je libovolná, používá se však obvykle čtvercová maska s lichým počtem řádků a sloupců, aby se dal jednoznačně určit středový (reprezentativní) bod. Hodnota masky musí být rovna jedné, menší hodnota by měla za

následek tmavší výsledný obraz, naopak hodnota větší než jedna by znamenala světlejší výsledný obraz. Příkladem rovnice pro výpočet s použitím jádra o velikosti 7x7 je rovnice 3.2.

$$I'(i, j) = \frac{1}{49} \sum_{k=-3}^3 \sum_{l=-3}^3 I(i+k, j+l) \quad (3.2)$$



Obrázek 3.3: Obyčejné průměrování - (horní řada zleva) zašuměný originál, jádro 3x3, jádro 5x5 (dolní řada zleva) jádro 7x7, jádro 9x9

### 3.4 Gaussův filtr

Gaussův filtr je metoda založená na konvoluci s vhodným konvolučním jádrem, stejně jako obyčejné průměrování. Zde je však rozdíl ve výpočtu hodnot konvolučního jádra, které se vypočítají podle rovnice Gaussovy funkce. Rovnice 3.3a a 3.3b jsou rovnice Gaussovy funkce v 1D a 2D. Gaussův filtr je většinou využíván při odstranění Gaussova šumu.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.3ab)$$



Jediným parametrem pro výpočet hodnot konvolučního jádra pomocí Gaussovy funkce je směrodatná odchylka  $\sigma$ , která určuje poloměr rozostření v pixelech. Opakované použití konvoluční masky má stejný účinek jako použití jedné masky s větším poloměrem rozostření. Poloměr rozostření této masky je pak roven odmocnině součtu druhých mocnin poloměrů rozostření použitých konvolučních masek. Použijeme-li tedy dvě konvoluce s jádry o velikosti 3x3 a 4x4, tak stejný účinek bude mít použití jediné konvoluce s jádrem o velikosti 5x5. Příkladem konvolučního jádra o velikosti 5x5 s  $\sigma=1$  je vzorec 3.4. Stejně tak jako u obyčejného průměrování hodnota konvolučního jádra musí být rovna jedné.

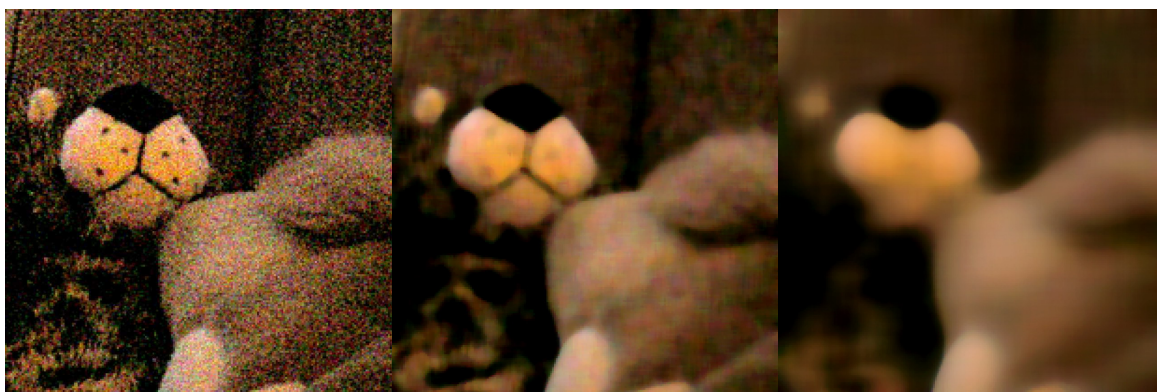
$$h = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (3.4)$$



Obrázek 3.4: Gaussův filtr - (horní řada zleva) zašuměný originál, jádro 3x3, jádro 5x5 (dolní řada zleva) jádro 7x7, jádro 9x9

## 3.5 Medián

Metoda filtrace pomocí mediánu pracuje tak, že hodnoty jasu lokálního okolí se vzestupně/sestupně seřadí a jako výsledná hodnota se vybere prostřední hodnota této posloupnosti neboli *medián*. V praxi se používá okolí s lichým počtem hodnot, aby bylo možno přesně určit prostřední prvek z posloupnosti. Jedinou komplikací a této metody je nutnost seřadit pole hodnot, což také zvyšuje časovou složitost této metody. Metoda filtrace pomocí mediánu je nelineární metodou a často se používá při odstranění impulsního šumu.



Obrázek 3.5: Medián - (zleva) zašuměný originál, okolí 3px, okolí 10px

## 3.6 Konzervativní vyhlazování

Poslední zde zmíněnou metodou je metoda konzervativního vyhlazování. Stejně jako předchozí metoda pracuje na principu lokální statistiky okolí a je to také nelineární metoda. Tato metoda nejprve zjistí minimální a maximální hodnotu jasu z okolí právě zpracovávaného bodu. Výsledná hodnota jasu se nemění, pokud leží uvnitř intervalu určeného minimální a maximální hodnotou z okolí bodu. Pokud je hodnota intenzity zpracovávaného pixelu větší než maximum, je nahrazen hodnotou tohoto maxima. Naopak pokud je hodnota pixelu menší než minimum, je výsledná intenzita rovna minimální hodnotě. Filtrování touto metodou je velice rychlé a používá se při odstranění impulsního šumu. Velikým kladem této metody oproti ostatním metodám je, že nejlépe zachovává hrany v obraze. To je velice výhodné právě v našem případě, kdy nejdříve potřebujeme odstranit šum a následně vyhledáváme hrany v obraze.

## 4 Detekce hran v obraze

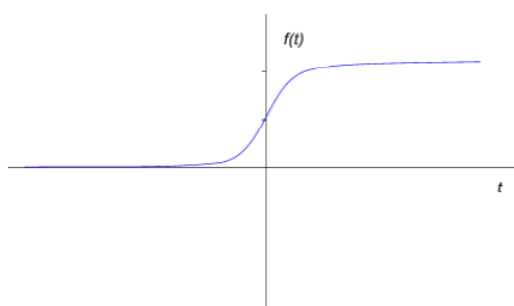
V části 3.1 (Co je to hrana), byla vysvětlena reprezentace hran v obraze, byla popsána ideální skoková hrana a bylo řečeno, že tato hrana se vyskytuje velmi zřídka. Na obrázku 4.1 jsou znázorněny jednotlivé typy hran: (a) Skoková hrana (step), (b) Dvě skokové hrany vedle sebe neboli čára (line), (c) Šikmá postupná hrana (ramp), (d) Dvojice šikmých hran vedle sebe neboli střecha (roof).



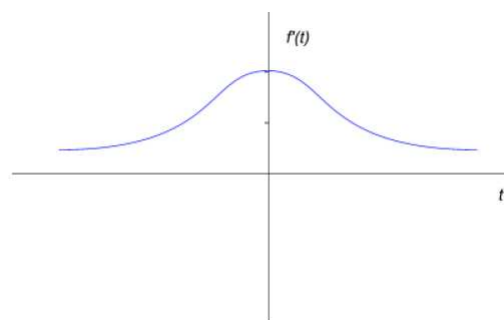
Obrázek 4.1a-d: Typy hran

Proces detekce hran lze rozdělit na tři fáze: filtrování, diferenciaci a detekci [2]. První fází je filtrace, kterou jsme si probrali v předchozí kapitole o odstranění šumu. V druhé fázi jsou zvýrazněny oblasti, kde je výrazná změna intenzity jasu v obraze. Tato část procesu detekce hran bude detailně popsána v této kapitole. Poslední fáze je detekce, kdy jsou lokalizovány body s největší změnou intenzity. I tato fáze bude podrobněji popsána a to v kapitole 5.

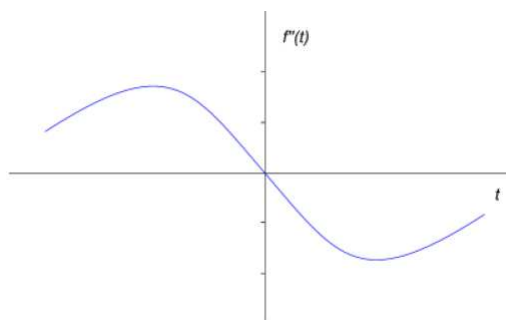
V předchozí části jsme hranu definovali jako velkou změnu jasové funkce, pak je hrana v místě největší hodnoty derivace jasové funkce. Toto využívá první typ hranových detektorů, které jsou založeny na první derivaci. Druhý typ jsou metody založené na druhé derivaci, zde se nehledá lokální maximum, nýbrž průchod nulou. Nalézt průchod nulou je obecně jednodušší než nalezení extrému funkce, to je důvod využití metod, které aproximují právě druhou derivaci. Na obrázku 4.2a je ukázána funkce intenzity jasu, hrana je v bodě  $t = 0$ . Na obrázku 4.2b je znázorněna první derivace funkce jasu, hrana je v místě nejvyšší hodnoty funkce. Na posledním obrázku 4.2c je druhá derivace funkce a její průchod nulou v místě hrany.



Obrázek 4.2a: Jednorozměrná funkce jasu



Obrázek 4.2b: Derivace funkce z obr. 4.2a



Obrázek 4.2c: Druhá derivace funkce z obr. 4.2a

Nejdříve se obecně podíváme na metody založené na první derivaci. Dále pak budou podrobněji probrány jednotlivé metody založené na první i druhé derivaci. Poslední část v této kapitole bude také věnována Cannyho operátoru, jednomu z hlavních bodů této práce. Algoritmus detekce hran pomocí Cannyho operátoru vyžaduje několik kroků a musí splňovat určité podmínky detekce hran. Více o Cannyho operátoru lze nalézt v podkapitole 4.7.

Metody založené na první derivaci hledají hrany v místě s vysokou hodnotou derivace jasové funkce. Výpočet hodnoty derivace (gradientu) aproximuje tato první skupina metod pomocí konvoluce s vhodným jádrem. Hrana je pak určena hodnotou tohoto gradientu, tj. velikostí a směrem. Maximální hodnota derivace (gradientu) je ve směru kolmo na hranu a tento směr lze vektorově popsat operátorem nabla  $\nabla$ .

$$\nabla f(x, y) = \left( \frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right) \quad (4.1)$$

Velikost gradientu neboli „sílu“ hrany lze určit jako délku vektoru 4.1: [1]

$$|\nabla f(x, y)| = \sqrt{\left( \frac{\partial f(x, y)}{\partial x} \right)^2 + \left( \frac{\partial f(x, y)}{\partial y} \right)^2} \quad (4.2)$$

Tyto uvedené vzorce platí pro spojité funkce. V diskrétním obraze gradient získáme jako rozdíl okolních pixelů v obraze. Okolní pixely nemusí být v některých případech přímo pixely sousední. V praxi se detekce hran provádí ve dvou, resp. ve čtyřech směrech.

## 4.1 Výpočet gradientu ve dvou směrech

Výpočet gradientu ve dvou směrech se provádí v horizontálním (X) a vertikálním (Y) směru (tedy pro řádky a sloupce) zvlášť. Výsledný gradient se pak spočítá jako

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (4.3)$$

pro zjednodušení výpočtu se často výpočet zjednodušuje na

$$G(x, y) = |G_x(x, y)| + |G_y(x, y)| \quad (4.4)$$

Nevýhodou této aproximace je možné zhoršení výsledku v některých případech.

Směr (orientaci) gradientu vzhledem k vodorovnému směru získáme výpočtem podle rovnice 4.5

$$\theta(x, y) = \tan^{-1} \frac{G_y(x, y)}{G_x(x, y)} \quad (4.5)$$

Jak bylo uvedeno výše, výsledný gradientní obraz můžeme určit pomocí konvolučního filtrování původního obrazu s vhodným konvolučním jádrem. Jednotlivé hranové detektory se pak liší svými konvolučními jádry (operátory). Jádro filtru udává, které body se pro výpočet gradientu použijí a jaké budou mít váhy. Vlastnosti filtru, jeho velikost a hodnoty významně ovlivňují výsledné vlastnosti hranového detektoru. [2] Většina významných filtrů, jak je vidět na obr. 4.3, má velikost masky 3x3. Vzhledem k tomu, že šum i filtr jsou vysokofrekvenční informace a detekce hran je citlivá na šum, používají se i masky, které pracují s větším okolím bodu. Proto lze nalézt i další varianty významných operátorů s velikostí 4x4, 5x5, 7x7 i větší.

Operátor	Řádkový gradient	Sloupcový gradient
Rozdíl sousedních pixelů (Pixel difference)	$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Rozdíl krajních pixelů (Separated pixel difference)	$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
Robertův operátor (Roberts operator)	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$
Prewittův operátor (Prewitt operator)	$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobelův operátor (Sobel operator)	$\frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Frei-Chenův operátor (Frei-Chen operator)	$\frac{1}{2+\sqrt{2}} \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	$\frac{1}{2+\sqrt{2}} \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

Obrázek 4.3: Impulsní charakteristiky významných filtrů

Příkladem větších rozměrů jádra může být Prewittové operátor 4x4 a 5x5,

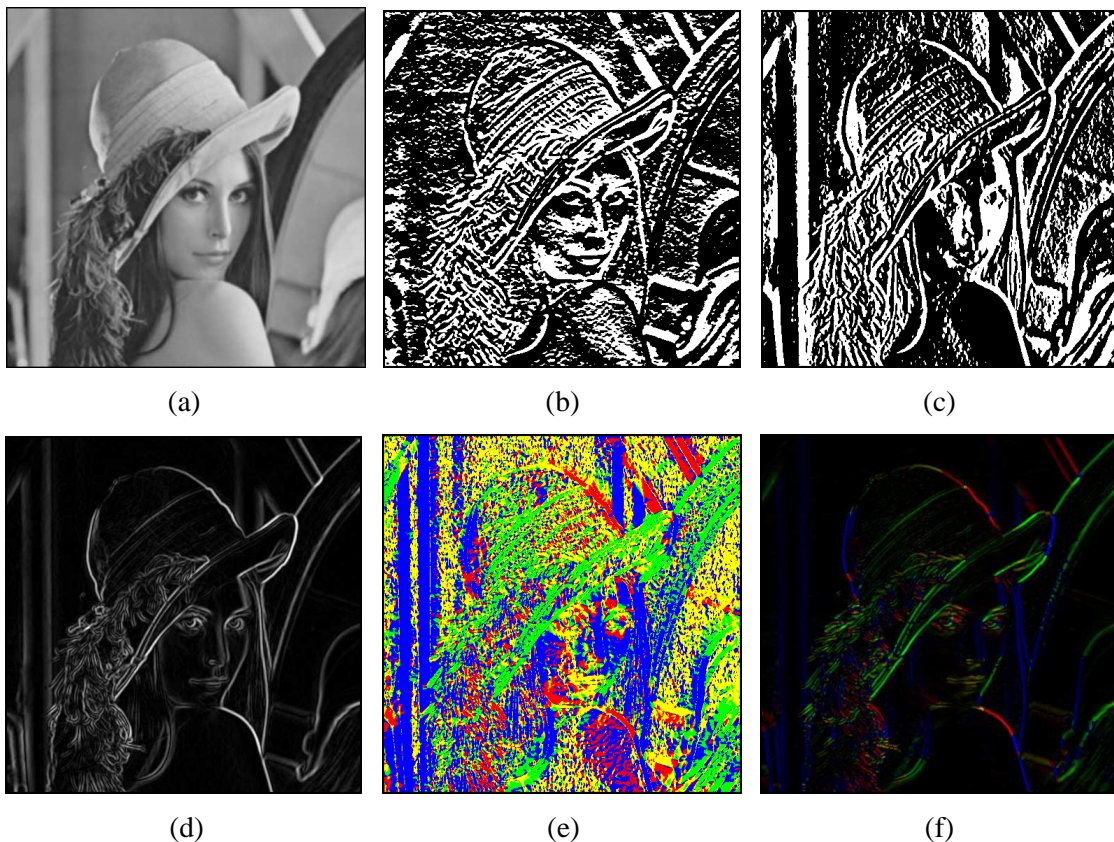
$$h_4 = \begin{bmatrix} -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \end{bmatrix} \quad \bar{h}_4 = \begin{bmatrix} -3 & -3 & -3 & -3 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 \end{bmatrix} \quad h_5 = \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix} \quad \bar{h}_5 = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

případně Sobelův operátor velikosti 5x5.

$$h_5 = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix} \quad \bar{h}_5 = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

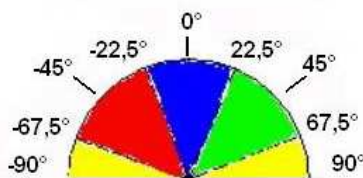
Protože zjišťujeme gradient ve dvou kolmých směrech, musíme použít dvě různá konvoluční jádra. Jedno jádro pro horizontální (vertikální) směr a pro druhý směr se použije konvoluční jádro o 90° otočené.

Další důležitou vlastností konvolučního jádra je, že součet všech hodnot musí být roven nule. To zaručí, že v oblastech s konstantní hodnotou bude i odezva konvoluce nulová. Za zmínku stojí i to, že u konvolučního jádra pro filtrování šumu je situace jiná a součet hodnot musí být roven jedné.



Obrázek 4.4: Jednotlivé fáze detekce hran ve dvou směrech

Na obrázku 4.4 lze vidět obrázky z jednotlivých fází detekce hran. První obraz (a) je filtrovaný Gaussovým filtrem podle konvolučního jádra ze vzorce 3.6. Další dva obrázky (b,c) jsou řádkové a sloupcové gradientní obrazy (použit byl Sobelův operátor). Výsledný gradient spočítaný podle vzorce 4.3 je na obr. 4.4d. Další obraz zobrazuje směry hran vypočítané podle rovnice 4.5. Barvy jsou určeny podle vzoru na obr. 4.5. Poslední obrázek 4.4f je obraz spočtený jako součin obrazu 4.4d a 4.4e.



Obrázek 4.5: Vzor barev pro jednotlivé úhly

## 4.2 Výpočet gradientu ve více směrech

Další možností, jak zlepšit detekci hran, je počítání gradientu ve více směrech. Zatím jsem počítali jen gradient ve vodorovném a svislém směru. Měli jsme jednu masku a druhou jsme získali otočením o 90°. Pokud použijeme 8 různých šablon jedné konvoluční masky, počítáme potom gradient v osmi různých směrech. Jednotlivé šablony masky získáme otočením předchozí masky o 45°. Výsledný gradient je pak roven největší odezvě na jednu z osmi masek, tak jak to vyjadřuje rovnice 4.6.

$$G(x, y) = \text{MAX} \{ |G_1(x, y)|, \dots, |G_m(x, y)| \} \quad (4.6)$$

Výhodou je lepší detekce hran prováděná ve více směrech, nevýhodou je nutnost vypočítat celkem osm konvolucí s původním obrazem. Nutné je ještě dodat, že směr hrany je určen směrem největšího gradientu. Zatímco u dvou masek pro dva kolmé směry spočítáme směr přesněji podle rovnice 4.5, tak zde získáme pouze úhly, které jsou násobky 45°. Na obrázku 4.6 je zobrazeno všech osm šablon tří významných operátorů – Robinsonova, Kirschova a jádro Prewittové používané pro detekci ve více směrech.

Směr gradientu	Robinson	Kirsch	Prewitt
H <sub>1</sub> 180°	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
H <sub>2</sub> 225°	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$
H <sub>3</sub> 270°	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

H <sub>4</sub> 315°	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$
H <sub>5</sub> 0°	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$
H <sub>6</sub> 45°	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$
H <sub>7</sub> 90°	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
H <sub>8</sub> 135°	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$
Koeficient	$\frac{1}{5}$	$\frac{1}{15}$	$\frac{1}{3}$

Obrázek 4.6: Šablony pro jednotlivé směry hranových operátorů

## 4.3 Operátor první derivace Gaussovské funkce

V předchozí části jsme obraz nejdříve filtrovali od šumu za použití Gaussova filtru. Prováděli jsme tedy konvoluci s jádrem Gaussova filtru. Následně jsme aproximovali první derivaci jasové funkce pomocí vhodného konvolučního jádra. Prováděli jsme tedy dvě konvoluce, nejdříve s gaussovským jádrem a poté s jádrem některého z hranových detektorů. Protože derivace i konvoluce jsou lineární operace, lze pořadí jejich výpočtu zaměnit. Tento postup je vyjádřen v rovnici 4.7, kde  $f$  je vstupní obraz a  $h$  je vyhlazovací funkce.

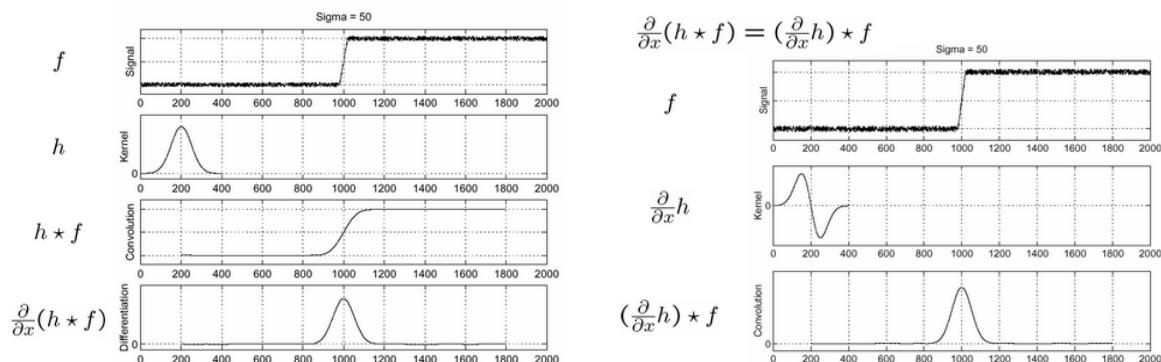
$$\frac{\partial}{\partial x}(h * f) = h * \frac{\partial f}{\partial x} = \frac{\partial h}{\partial x} * f \quad (4.7)$$

Tato metoda tedy provádí výpočet gradientního obrazu tak, že nejprve vypočítá potřebnou derivaci funkce  $h$  a ta je poté použita pro konvoluci se vstupním obrazem. V praxi se volí jako vyhlazovací funkce gausián. V tomto případě je počítána první derivace Gaussovské funkce a ta je dále použita pro konvoluci se vstupním obrazem. Získáme tak vyhlazení (odstranění šumu) a výpočet gradientního obrazu v jednom kroku výpočtem jedné konvoluce. Rovnice 4.8a je rovnice Gaussova filtru a výpočet derivace s provádí podle rovnice 4.8b. [3]

$$G(x) = e^{-\frac{x^2}{2\sigma^2}} \quad G'(x) = \left(-\frac{x}{\sigma^2}\right) \cdot e^{-\left(\frac{x^2}{2\sigma^2}\right)} \quad (4.8ab)$$



Rozdíly v postupu výpočtu lze vidět na obrázku 4.7. Obrázek vlevo, je ukázka filtrování obrazu  $f$  pomocí filtrační funkce  $h$ , v dalším kroku je ukázán filtrovaný signál a posledním krokem je aproximována první derivace. Obrázek vpravo demonstruje výpočet pomocí operátoru první derivace Gaussovské funkce. Máme zde opět vstupní signál  $f$ , který však konvolujeme s první derivací gaussovy funkce. Výsledek konvoluce je pak totožný jako v předchozím případě. Hrana se v obou případech nachází v bodě největší hodnoty poslední funkce.



Obrázek 4.7: Demonstrace rozdílu postupu ve dvou předchozích metodách detekce hran

V praxi je tedy výpočet následující: rádkový, resp. sloupcový gradientní obraz se spočítá jako konvoluce vstupního obrazu s vektorem hodnot vypočítaných podle rovnice 4.8b. Konvoluce se provádí po řádcích, resp. po sloupcích. Výsledný gradient je pak vypočítán stejně jako v předcházejících případech podle rovnice 4.3 popř. 4.4.

Nutné je už pouze dodat, že zde končí výčet nejčastěji používaných metod založených na první derivaci. Další metody jsou již metody založené na druhé derivaci, vyjma Cannyho operátoru, který je popsán na konci kapitoly.

## 4.4 Laplaceův operátor

Změny gradientu se u metod založených na první derivaci projeví jako lokální extrém první derivace. Jelikož extrém první derivace odpovídají bodům, kde druhá derivace prochází nulou, všechny dále zmíněné operátory aproximují druhou derivaci jasové funkce. Tyto operátory pak označujeme jako operátory průchodu nulou (anglický termín je označuje jako zero-crossing). Využíváme toho, že nalézt průchod nulou je snazší než nalézt extrém funkce, velikou nevýhodou je však skutečnost, že druhá derivace je ještě více citlivější na šum než derivace první.

Metody založené na druhé derivaci pracují často s Laplaceovým operátorem (viz. rov. 4.9). Laplaceův operátor, zkráceně Laplacián má stejné vlastnosti ve všech směrech a díky tomu je invariantní vůči rotaci.

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial^2 x} + \frac{\partial^2 f(x, y)}{\partial^2 y} \quad (4.9)$$

Nejjednodušší způsob, jak vypočítat výslednou odezvu za použití Laplaciánu, je použití vhodných konvolučních masek. Konvoluční masky Laplaciánu se 4-okolím, 8-okolím a separabilním 8-okolím jsou uvedeny v obrázku 4.8.

$$h_1 = \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad h_2 = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad h_3 = \frac{1}{8} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

Obrázek 4.8: Normalizované impulsní odezvy Laplaciánu.

Výsledný obraz získáme konvolucí s jedním z těchto jader. Výsledkem mohou být jak kladné tak i záporné hodnoty intenzity jasu. Proto se v praxi záporné hodnoty oříznou - celý rozsah se převede do kladných hodnot (většinou přičtením hodnoty 128, tedy střední šedé hodnoty). Druhou častější možností je použití absolutní hodnoty. Jelikož má však Laplaceův operátor obecně pozitivní a negativní odezvu na každou hranu, výpočet absolutní hodnoty zvýrazní tuto dvojitou odezvu.

## 4.5 Laplacián Gaussiánu (LoG)

Z důvodů větší citlivosti druhé derivace na šum, je vhodné její výpočet kombinovat s vyhlazením, které odstraní maximální množství šumu a zachová co nejvíce hran. Nejčastěji používaným filtrem je Gaussův filtr (viz. 3.4) a proto se v praxi využívá Laplacián Gaussiánu (Laplacian of Gaussian - LoG). Druhá derivace je stejně jako první derivace a konvoluce lineární operace, proto je možno zaměnit postup výpočtu stejně jako u operátoru první derivace Gaussovské funkce (viz. 4.3). Proto filtrování Gaussovým filtrem a následný výpočet Laplaciánu má stejný výsledek jako výpočet LoG a jeho aplikace na vstupní obraz. Tato skutečnost je vyjádřena rovnicí 4.10.

$$\Delta[G_\sigma(x, y) * f(x, y)] = [\Delta G_\sigma(x, y)] * f(x, y) = LoG * f(x, y) \quad (4.10)$$

Postup výpočtu Laplaciánu Gaussiánu demonstrují rovnice 4.11 a 4.12

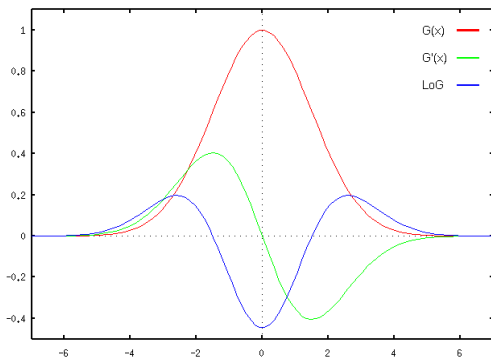
$$\begin{aligned} \frac{\partial}{\partial x} G_\sigma(x, y) &= \frac{\partial}{\partial x} e^{-\frac{(x^2+y^2)}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \\ \frac{\partial^2}{\partial^2 x} G_\sigma(x, y) &= \frac{x^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} - \frac{1}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} = \frac{x^2 - \sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \end{aligned} \quad (4.11 - 4.12)$$

podobně získáme  $\frac{\partial^2}{\partial^2 y} G_\sigma(x, y)$  jako  $\frac{y^2 - \sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$

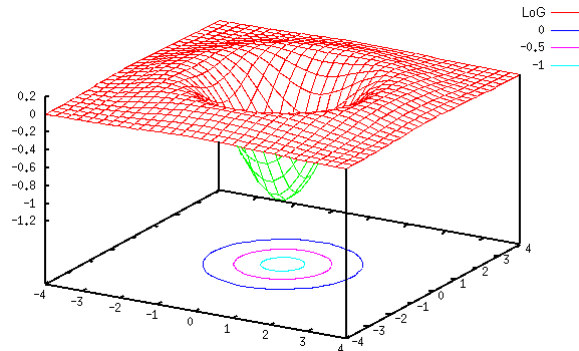
Výsledné hodnoty konvolučního jádra LoG pak získáme z rovnice 4.13, kde  $r^2 = x^2 + y^2$

$$LoG = \Delta G_{\sigma}(x, y) = \frac{\partial^2}{\partial^2 x} G_{\sigma}(x, y) + \frac{\partial^2}{\partial^2 y} G_{\sigma}(x, y) = \frac{r^2 - 2\sigma^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}} \quad (4.13)$$

V praxi se používá maska vynásobená -1, výsledná maska pak obsahuje kladné hodnoty ve svém středu, na které navazují hodnoty záporné, ty pak postupně přecházejí v nulu. Pro lepší představu jak vypadá LoG filtr následuje obrázek 4.10 a 4.11. Operátor -LoG se díky tvaru funkce ve 3D často označuje jako Mexican hat kernel nebo také jako sombrero filter. Na obrázku 4.10 je červeně zakreslena Gaussova funkce, zeleně její první derivace a modře funkce LoG.



Obrázek 4.10: Gausián, der. G(x), LoG



Obrázek 4.11: Laplacián Gausiánu ve 3D

## 4.6 Diferenciál Gaussovské funkce (DoG)

Funkce Laplaciánu Gausiánu, použitá v předchozí části, lze také aproximovat pomocí diferenciálu Gaussovské funkce, jak je naznačeno ve vzorci 4.14.

$$F(x, y) = -\nabla^2[G(x, y, \sigma)] \approx G(x, y, \sigma_1) - G(x, y, \sigma_2) \quad (4.14)$$

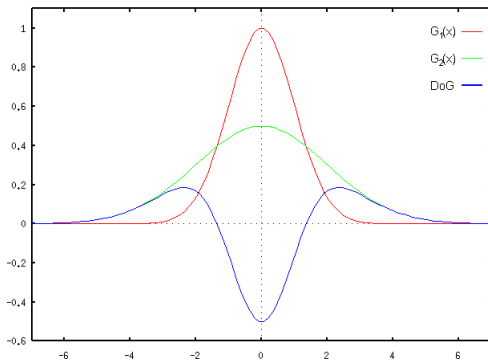
To znamená, že pokud použijeme dvakrát Gaussův filtr, pokaždé s jiným parametrem  $\sigma$  a tyto dva obrazy od sebe odečteme, získáme přibližně stejný výsledek jako po aplikaci Laplaciánu Gausiánu (LoG). V praxi se nejdříve vypočítá konvoluční jádro, které odpovídá diferenciálu Gaussovské funkce – Difference of Gaussian (DoG) a to se následně použije při konvoluci se vstupním obrazem. Postup je vyjádřen v rovnici 4.15.

$$G_{\sigma_1} * f(x, y) - G_{\sigma_2} * f(x, y) = (G_{\sigma_1} - G_{\sigma_2}) * f(x, y) = DoG * f(x, y) \quad (4.15)$$

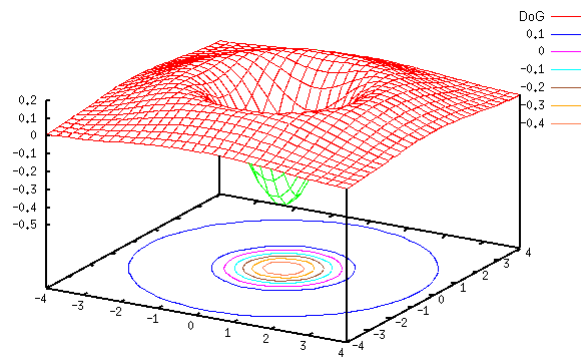
Hodnoty konvoluční jádra operátoru diferenciálu Gaussovské funkce (DoG) pak získáme podle rovnice 4.16.

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}} \left[ \frac{1}{\sigma_1} e^{-\left(\frac{x^2+y^2}{2\sigma_1^2}\right)} - \frac{1}{\sigma_2} e^{-\left(\frac{x^2+y^2}{2\sigma_2^2}\right)} \right] \quad (4.16)$$

Obrázky 4.12 a 4.13 vyobrazují grafy DoG filtru ve 2D a 3D. Na obrázku 4.12 je červeně a zeleně zobrazena Gaussova funkce (každá s jiným parametrem sigma) a modře je zobrazen jejich rozdíl.



Obrázek 4.12: Diferenciál Gaussiánu ve 2D



Obrázek 4.13: Diferenciál Gaussiánu ve 3D

Nakonec je ještě potřebné dodat, že abychom získali kladné hodnoty ve středu konvolučního jádra je třeba, aby v rovnici 4.16 byl parametr  $\sigma_1 < \sigma_2$ . Pro dobrou detekci hran je také nutné, aby oba parametry byly v určitém poměru vůči sobě. [3] Poslední zmínkou je, že diferenciál Gaussiánu (DoG), popř. Laplacián Gaussiánu (LoG) se v literatuře často označují jako operátor Marr-Hildreth.

Tímto jsme vyčerpali již všechny často používané metody založené na druhé derivaci. Zbývá tedy jen zmínit Cannyho hranový detektor, momentálně jeden z nejlepších algoritmů pro detekci hran založený na první derivaci.

## 4.7 Cannyho hranový detektor

Cannyho algoritmus pro detekci hran je často označován jako optimální hranový detektor. John F. Canny navrhnul v roce 1986 svůj operátor tak, aby splňoval tři základní kritéria, která dřívější operátory nedokázali zajistit. Mezi hlavní kritéria Cannyho hranového detektoru patří: [4]

- **Minimální chybovost** – poměr mezi hranami a šumem musí být co největší. Musí být detekovány všechny hrany. Nesmí být detekovány místa, která nejsou hranou.
- **Lokalizace** – poloha detekované hrany musí být co nejbližší její skutečné poloze v obraze
- **Jednoznačná odezva** – operátor může detekovat pouze jednu odezvu na každou hranu (nesmí docházet k dvojité odezvě na hranu)

Aby byli dodrženy všechny tyto podmínky, je Cannyho algoritmus realizován v několika krocích, mezi základní kroky patří:

1. **Odstranění šumu** – ve vstupní obrazě je odstraněn šum. Většinou je tato operace realizována Gaussovým filtrem (viz. 3.4)
2. **Určení gradientu** (aproximace první derivace) – výsledný gradient je většinou vypočten pomocí konvoluce se Sobelovým operátorem (viz. 4.1) nebo pomocí operátoru první derivace Gaussovské funkce (viz. 4.3)
3. **Nalezení lokálních maxim** – v gradientním obrazě jsou potlačeny hodnoty, které nejsou lokální maxima (ang. výraz označuje tuto část jako *NonMaxima Suppression*). Dojde tak ke ztenčení hran a je získána jednoznační odezva na hranu.
4. **Eliminace nevýznamných hran** – gradientní obraz po ztenčení hran je převeden do binární podoby. Většinou se používá globální prahování (global thresholding) nebo lépe prahování s hysterezí (hysteresis thresholding).

Odstraněním šumu zaručíme splnění první podmínky, tedy co nejmenšího počtu chyb při detekci hran. Druhá podmínka je splněna použitím vhodného operátoru a poslední kritérium je naplněno díky ztenčení detekovaných hran - tedy potlačení hodnot, která nejsou lokálními extrémami. První dva kroky Cannyho algoritmu již byli popsány v předchozích kapitolách, zde se jen stručně zmíním jak realizovat krok ztenčení hran (thinning). Poslední krok bude popsán v následující kapitole, která je přehledem různých metod prahování.

U Cannyho hranového detektoru je fáze ztenčení hran (thinning) označována jako NonMaxima Suppression. Tento algoritmus pracuje tak, že dojde k výběru pouze lokálních maxim z hodnot gradientu vypočítaného pomocí konvoluce s vhodným hranovým operátorem. Hodnoty, které nejsou lokálními extrémami jsou potlačeny. Prakticky jsou tedy vybrány pouze pixely jejichž okolí je ve směru a proti směru gradientu nižší, v opačném případě je pixel označen jako nehranový a je vymazán. Směr hrany je známý, je vypočítán podle rovnice 4.5 v kroku určení gradientu. Demonstrace funkce ztenčení je na obrázku 4.14ab



Obrázek 4.14ab: (a) Gradientní obraz, (b) Gradientní obraz po ztenčení

## 5 Post-processing obrazu

Další kapitola se bude zabývat, jak už z názvu vyplývá, závěrečnou úpravou obrazu po detekci hran. Už v části o omezení barevného prostoru, přesněji u metod pro převod do binárního obrazu jsem zmínil prahování (thresholding) a to je přesně metoda, kterou využijeme pro převod šedotónového obrazu na obraz černobílý. Projdu zde jednotlivé metody prahování jako je prahování s pevným prahem (statické prahování), prahování adaptivní a prahování s hysterezí.

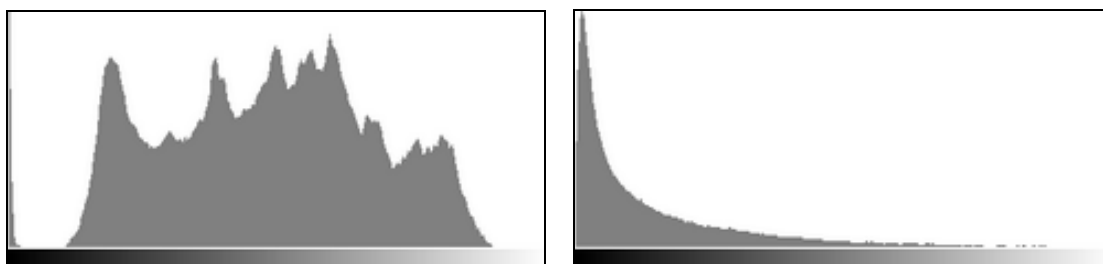
V předchozí kapitole o detekci hran jsme zjistili, že výstupem algoritmů založených na první derivaci je šedotónový (gradientní) obraz, kde maxima odpovídají hranám. Úkolem v této kapitole je tedy získat binární (černobílý obraz), kde maximální intenzita odpovídá hranám. Postup prahování lze vyjádřit rovnicí 5.1

$$f'(x, y) = \begin{cases} 255, & \text{když } f(x, y) > T \\ 0, & \text{když } f(x, y) \leq T \end{cases} \quad (5.1)$$

### 5.1 Globální prahování

První metodou prahování je globální prahování (global thresholding). Globální prahování používá pouze jeden parametr – globální prahovou hodnotu. Většinou je tato hodnota značená jako  $T$  (threshold). Určení správné hodnoty prahu není ve většině případů jednoduché. Se správným výběrem prahu nám však může pomoci histogram. Ukázka histogramu obr. 4.4a a 4.4d je na obrázku 5.1.

*Histogram kvantifikuje množství a frekvenci barev obsažených v obraze. Matematicky je histogram vektor absolutních četností hodnot zastoupených v obraze. Jinými slovy řečeno, hodnota histogramu  $H$  pro index  $i$  říká, kolik pixelů v obraze má intenzitu  $i$ . Pokud je původní obraz složen z jasových složek, je histogram jednorozměrný vektor, v případě složek RGB je složen ze tří vektorů, atp. Obrazy, které reprezentují barvu pomocí více barevných kanálů mají pro každý kanál zvláštní histogram. Tmavé odstíny se nacházejí blízko nulové hodnoty a s rostoucí hodnotou ve směru vodorovné osy roste i intenzita. Zcela vlevo je tedy barva černá a zcela vpravo maximální intenzita, obvykle odpovídající bílé barvě.[1]*



Obrázek 5.1: Ukázka možných histogramů

## 5.2 Prahování s hysterezí

Hysterezí prahování, neboli prahování se dvěma prahy, jak už z názvu vyplývá, umožňuje určit dvě prahové hodnoty – horní prahovou hodnotu  $H_1$  (high) a dolní prahovou hodnotu  $H_2$  (low). Postup výpočtu binárního obrazu je pak následující. Pokud intenzita zpracovávaného pixelu je vyšší než hodnota horního prahu  $H_1$ , pak nová hodnota je rovna maximální hodnotě intenzity (většinou hodnotě 255). Naopak pokud je intenzita menší než dolní prah  $H_2$  je pixel považován za nevýznamný a jeho intenzita je ohodnocena nulou. Pixely s intenzitou která je v intervalu  $\langle H_1, H_2 \rangle$  jsou označeny jako významné jen pokud sousedí alespoň s jedním pixelem, jehož hodnota je vyšší než prah  $H_1$ , jinak je označen jako nevýznamný a jeho nová hodnota je rovna nule. V praxi se tak pro hysterezí prahování využívá rekurzivní funkce. V praxi je často dolní hodnota prahu určena jako polovina horního prahu. Ukázka, jak vypadá hysterezní prahování v porovnání s globálním prahováním je na obr. 5.2ab.



Obrázek 5.2ab: Prahování obr. 4.14b – (a) globální s hodnotou 96, (b) hysterezní hodnotami 128 a 64.

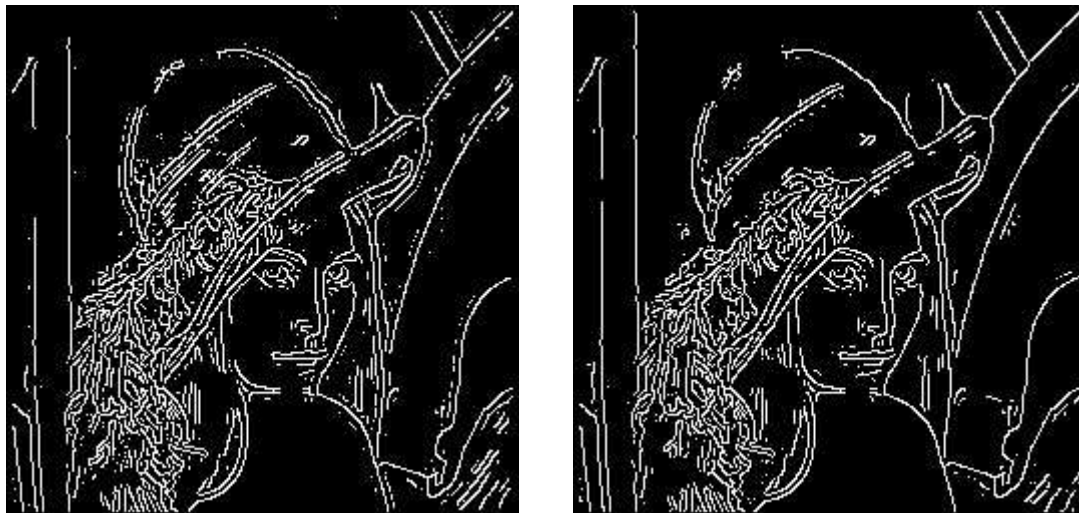
## 5.3 Adaptivní prahování

Prahovou hodnotu občas není možno určit globálně a v některých případech nedosahuje dobrých výsledků ani prahování s hysterezí. Tento případ nastává u obrazů, které nemají konstantní hodnotu pozadí a v obraze se vyskytuje šum a různé stíny. U detekce hran tento problém nenastává, protože pozadí má konstantní hodnotu a je vždy černé (v případě inverzního obrazu bílé) a hrany jsou určeny hodnotou gradientu. Pro úplnost však zde popíšeme i tuto metodu.

Při použití adaptivního prahování, není určena hodnota prahu globálně, ale je určena pro různé části obrazu zvlášť. Parametrem adaptivního prahování je pak tedy jen velikost okolí, ze kterého se prahová hodnota bude počítat. Většinou se hodnota počítá jako polovina rozdílu maximální a minimální hodnoty v okolí, nebo jako medián hodnot okolí, popř. jako polovina z průměru okolních

hodnot intenzit. Někdy může být vypočítaná hodnota „posunutá“ a je dobré použít druhý parametr - offset, tedy hodnotu, která se přičte (odečte) k vypočtené hodnotě.

Dále následuje už jen ukázka adaptivního prahování (obr. 5.3), které bylo provedeno na ztenčeném gradientním obraze 4.14b.



Obrázek 5.3ab: Adaptivní prahování

Obrázek 5.3a je adaptivní prahování, kde okolí bylo nastaveno na 7 pixelů, byl počítán průměr z okolí a byla připočtena hodnota offsetu -15. U obrázku 5.3b bylo okolí také nastaveno na 7 pixelů, byla počítána polovina z rozdílu max. a min hodnoty okolí a byla připočtena hodnota offsetu -45.



## 6 Implementace

Tato kapitola se zabývá vlastní implementací aplikace pro detekci hran v obraze. Bude zde stručně představena knihovna OpenCV, která byla použita při implementaci. Dále zde také uvedu, které výše zmíněné metody pro detekci hran byly prakticky implementovány.

### 6.1 Knihovna OpenCV

*OpenCV (Open Computer Vision library)* je volně dostupná grafická knihovna vyvinutá firmou Intel. Její výhodou, je množství funkcí pro práci s grafikou a grafickými soubory.

V našem případě využijeme pouze funkce pro otevření obrazu, zobrazení obrazu v okně a případně také uložení obrazu. Pro snadnější uživatelskou práci s programem použijeme také základní komponenty pro grafické uživatelské rozhraní (GUI), které knihovna OpenCV nabízí. Všechny ostatní funkce, které realizují metody odstranění šumu, detekci hran a prahování byly nově implementovány. Knihovna sice obsahuje mnoho již implementovaných funkcí pro práci s obrazem, přesněji i funkce pro detekci hran. V knihovně tak lze nalézt i funkce jako `cvSobel`, `cvCanny`, `cvLaplace`, jejichž název je plně vypovídající o tom, které metody zmíněné v této práci odpovídají těmto funkcím.

V OpenCV vlastně najdeme téměř všechny výše zmíněné metody a to nejen pro detekci hran, ale i pro odstranění šumu a prahování. Zde jen zmíním funkce jako `cvSmooth`, `cvThreshold` a `cvAdaptiveThreshold`. Pokud si pro zajímavost přiblížíme funkci `cvSmooth`, tak zjistíme, že jedním z parametrů je parametr `smoothtype`, jeho hodnoty pak mohou být `CV_BLUR`, `CV_GAUSSIAN`, nebo také `CV_MEDIAN`.

Praktická část této práce, pak tedy mohla být implementována jen za pomoci všech těchto funkcí, ale to by byla práce příliš snadná a také by neodpovídala zadání, jehož jednou z částí byl úkol tyto metody implementovat. Pro bližší seznámení s knihovnou OpenCV, lze na přiloženém CD nalézt manuál a další dokumenty. Součástí instalace grafické knihovny jsou pak i ukázkové příklady a zdrojové kódy těchto příkladů.

### 6.2 Implementované metody

Praktickou částí této práce je aplikace **Edge**, která provádí detekci hran v obraze. Program byl implementován v Microsoft Visual Studio 2005 s využitím již zmíněné knihovny OpenCV. Aplikaci lze spouštět v operačním systému Windows i Linux (na CD lze nalézt přiložený Makefile pro překlad a spuštění).

Program implementuje několik metod pro předzpracování obrazu, detekci hran i post-processing obrazu. Dále obsahuje různé pomocné funkce pro výpočet konvoluce obrazu s konvolučním jádrem, funkce pro výpis matic, funkce pro ovládání základního GUI a další.

Výpis jednotlivých metod, které jsou implementovány v rámci programu:

- Metody pro odstranění šumu
  - Obyčejné průměrování
  - Gaussův filtr
  - Konzervativní vyhlazování
- Metody pro detekci hran
  1. Sobelův operátor
  2. Prewittové operátor
  3. Cannyho algoritmus (používající operátor první derivace Gaussovské funkce)
  4. Laplacián Gausiánu (LoG)
- Metody pro post-processing obrazu
  1. Globální prahování
  2. Hysterezní prahování

Požadované metody lze vybrat zadáním správných parametrů při spuštění programu. Popis jednotlivých parametrů a stručný návod pro ovládání programu, jsou popsány v příloze 1.

# 7 Závěr

Cílem mé bakalářské práce bylo prostudovat Cannyho hranový detektor a další často používané algoritmy pro detekci hran. Dalším úkolem bylo Cannyho algoritmus s několika dalšími hranovými detektory přímo implementovat.

Protože z detekcí hran úzce souvisí i předzpracování obrazu, byly první kapitoly této práce právě věnovány této problematice. První kapitola poskytla úvod k této práci, další kapitoly se již prakticky zabývaly předzpracováním obrazu, tedy převodem vstupního barevného obrazu na šedotónový (kapitola 2) a odstraněním šumu (kapitola 3). Ve čtvrté, nejpodstatnější kapitole této práce, jsem pak popsal jednotlivé metody pro detekci hran a kapitola byla završena jednou z hlavních náplní této práce Cannyho operátorem. Jelikož obraz po detekci hran je potřeba dále upravit, nejlépe převést do binární podoby, byla v další kapitole diskutována problematika post-processingu obrazu. V předposlední kapitole byla také představena grafická knihovna OpenCV a byly uvedeny metody, které jsem přímo implementoval.

Dle mého názoru je hlavním přínosem této práce představení několika hranových detektorů a spojení těchto metod s metodami pre-processingu a post-processingu obrazu, které úzce souvisí s efektivní detekcí hran. Výsledná aplikace pak implementuje všechny tyto kroky procesu detekce hran a dává uživateli možnost testovat a porovnávat kvalitu detekce hran při použití různých algoritmů. Velké množství parametrů, které lze zadat při spuštění programu, pak umožňuje kombinovat různé metody odstranění šumu s různými metodami detekce hran a dále zvolit jednu z metod prahování (viz. Příloha 1 – Uživ. příručka k programu).

Během vývoje aplikace a psaní této práce jsem se přiučil mnoha věcem o zpracování obrazu a hlavně zajímavé problematice detekce hran. Při vývoji aplikace jsem se pak naučil pracovat s knihovnou OpenCV, která umožňuje efektivní práci s obrazem. Já bohužel většinu z již implementovaných funkcí použít nemohl, ale odměnou za vlastní implementaci funkcí mi byla celkem vysoká kvalita detekce hran se kterou jsem velice spokojen.

Menší nevýhodu knihovny OpenCV vidím v absenci většího počtu prvků pro grafické rozhraní a tím i omezení GUI mé aplikace. Dalším krokem při vývoji by tak mohla být grafická nadstavba mé aplikace, která by usnadnila práci s obrazy, zadávání parametrů a volbu metod. Implementované algoritmy jsou optimalizovány na maximální možnou kvalitu na úkor časové a výpočetní složitosti, zde se tak nachází další možnost pro vylepšení aplikace.

Metod pro detekci hran existuje mnoho, v této práci byl představen hlavně Cannyho hranový detektor s dalšími často používanými operátory. Existuje tak i nadále možnost rozšířit mojí práci o další metody a tyto metody pak prakticky implementovat.

# Literatura

- [1] Žára, J., Beneš, B., Sochor, P., Felkel, P. *Moderní počítačová grafika*.  
Brno, Computer Press, 2004. ISBN 80-251-0454-0.
  
- [2] Španěl, M., Beran, V. *Obrazové segmentační techniky - Přehled existujících metod* [online].  
Poslední aktualizace 2006-01-19. [cit. 2008-04-06].  
Dostupné na URL: < <http://www.fit.vutbr.cz/~spanel/segmentace/> >.
  
- [3] Shapiro, Linda G, Stockman, George C. *Computer Vision*.  
Prentice-Hall Inc., New Persey, 2001. ISBN 0-13-030796-3.
  
- [4] Parker, James R. *Algorithms for Image Processing and Computer Vision*.  
Wiley Computer Publishing, 1997. ISBN 0-471-14056-2.

# Seznam příloh

Příloha 1. Uživatelská příručka k programu

Příloha 2. CD obsahující zdrojové soubory projektu, spustitelnou aplikaci s testovacími obrázky, technickou zprávou, manuál a dokumentaci k programu ve formátu HTML atd.

Adresářová struktura CD:

- `\Edge` - obsahuje projekt pro Visual Studio 2005
- `\src` - zdrojové soubory programu, Makefile pro systém Linux
- `\build` - spustitelná verze aplikace pro MS Windows (soubor „Edge.exe“)
- `\img` - obrázky použitelné pro testování programu
- `\doc` - dokumentace (generována aplikací Doxygen) a manuál k programu
- `\OpenCV_install` - instalační soubory knihovny OpenCV pro MS Windows a Linux
- `\OpenCV_manual` - manuál knihovny OpenCV a další návody pro práci s knihovnou
- `\thesis` - text technické zprávy v elektronické podobě ve formátu .doc a .pdf
- `\thesis_img` - obrázky použité v technické zprávě (obrázky ve větším rozlišení)

# Příloha 1

## Uživatelská příručka k programu

Zde bych chtěl čtenáře instruovat, jak pracovat s programem **Edge**, jaké parametry lze použít a s jakými formáty vstupních dat lze pracovat.

### *Spuštění programu*

Program je spustitelný přímo z CD z adresáře \build, kde se nachází soubor Edge.exe. Jelikož však soubor ukládá obrázky z jednotlivých fází detekce hran, nebude při spuštění programu z CD možné tyto obrázky uložit a dojde k vypsání chybového hlášení knihovny OpenCV. Je tedy doporučeno spouštět aplikaci z disku, kde lze tyto obrázky uložit (obrázky jsou uloženy do souboru Output[N].jpg, kde N je číslo od 0 do 4).

### *Parametry spuštění*

Parametrů je velké množství vzhledem k množství metod, které lze pro jednotlivé fáze detekce hran zvolit.

Jediným a to povinným parametrem je první parametr určující vstupní soubor obrazu, pokud vstupní obraz není zvolen, je na standardní výstup vypsána nápověda.

OpenCV podporuje tyto formáty - BMP, DIB, JPEG, JPG, JPE, PNG, PBM, PGM, PPM, SR, RAS, TIFF, TIF. Aplikace byla testována s obrázky ve formátu JPG (JPEG), BMP a PNG, které lze nalézt v adresáři \img. Pokud zvolený vstupní soubor nelze otevřít, dojde k vypsání chyby.

Jednotlivé algoritmy detekce hran, lze volit druhým parametrem při spuštění programu.

- **Sobelův operátor** `Edge.exe test.bmp -s`
- **Prewittové operátor** `Edge.exe test.bmp -p`

U sobelova a prewittové operátoru lze volitelně zadat další 2 parametry, které určují metodu odstranění šumu a metodu prahování.

Volby pro odstranění šumu jsou následující:

- `-average`  
Pro redukci šumu obyčejným průměrováním (nastaveno okolí 5px)
- `-gauss3 (gauss5, gauss7, gauss9)`  
Gaussův filtr (číslo udává rozměr konvolučního jádra)

- `-conservative`

Pro konzervativní vyhlazování (nastaveno okolí 5px)

Volby metody prahování jsou:

- `-global`                      Globální prahování
- `-hysteresis`                  Hysterezní prahování
- **Cannyho operátor**                      `Edge.exe test.bmp -c`

I u Cannyho hranového detektoru lze určit další dva nepovinné parametry. Jedním z nich je hodnota parametru SIGMA. Další je parametr určující metodu prahování, zde jsou možnosti stejné jako u Sobela a Prewittové – lze tedy zvolit globální nebo hysterezní prahování.

- **Algoritmus Marr-Hildreth**              `Edge.exe test.bmp -m`

U tohoto algoritmu lze zadat pouze jediný další parametr a to hodnotu parametru SIGMA (stejně jako u Cannyho operátoru).

Hodnota parametru SIGMA musí být v rozsahu od 1.0 do 3.0, v případě zadání hodnoty mimo tento rozsah je použita přednastavená hodnota (1.5).

Pokud není zadán některý z volitelných parametrů, jsou automaticky použity přednastavené hodnoty. Stejně tak pokud není zvolen operátor detekce hran, je defaultně nastaven Sobelův operátor.

Defaultní nastavené metody a hodnoty:

Detekce hran:	Sobelův operátor
Odstranění šumu:	Gaussův filtr (velikost jádra 5x5)
Prahování:	Hysterezní prahování s dvěma prahy
Sigma:	1.50

Při spuštění programu jsou také nastaveny výchozí hodnoty prahů u metod jednotlivých metod prahování (tyto prahové hodnoty však lze měnit během programu pomocí Trackbaru):

Globální prahování:	<code>threshold = 80</code>
Hysterezní prahování:	<code>high = 80, low = 40</code>

Shrnutí všech parametrů spuštění:

`Edge.exe test.bmp [-s|-p][-average|-gauss3|-gauss5|-gauss7|-gauss9]-conservative][-global]-hysteresis]`

`Edge.exe test.bmp [-c][sigma][-global]-hysteresis]`

`Edge.exe test.bmp [-m][sigma]`