

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PŘEVOD ČERNOBÍLÝCH OBRAZŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

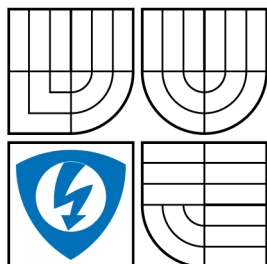
AUTOR PRÁCE
AUTHOR

LUKÁŠ BÁRNET

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PŘEVOD ČERNOBÍLÝCH OBRAZŮ BLACK AND WHITE DOCUMENTS CONVERSION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

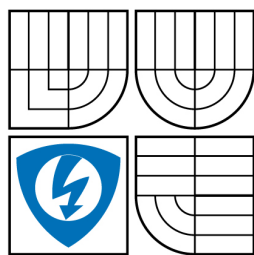
AUTOR PRÁCE
AUTHOR

LUKÁŠ BÁRNET

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MILOSLAV RICHTER, PH.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Lukáš Bárnét

ID: 106364

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Převod černobílých dokumentů

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku zpracování obrazu. Vytvořte program pro převod černobílých snímků (kopie textových dokumentů) na binární. Proveďte rozbor úlohy, navrhnete různé metody řešení a vytvořte testovací databázi. Ve snímku je nutné oddělit text s proměnným kontrastem od pozadí, jehož úroveň jasu je v ploše proměnná. Proveďte zhodnocení kvality jednotlivých metod na základě typu vstupního obrazu.

DOPORUČENÁ LITERATURA:

Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 1998, ISBN 80-251-0454-0

Hlaváč V., Šonka M.: Počítačové vidění, Grada, Praha 1992, ISBN 80-85424-67-3

Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

Kraus K.: Photogrammetrie 1 und 2, Ummeler / Bonn, 1996

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá zpracováním černobílých, šedotónových dokumentů, tedy převodem z šedotónového obrazu dokumentu na binární. Jsou v ní popsány postupy pro oddělení textu s proměnným jasnem od pozadí. V práci jsou použity různé metody pro oddělení popředí od pozadí, takzvaná segmentace obrazu. Jsou zde také porovnány výsledky různých metod.

KLÍČOVÁ SLOVA

programování, C++, obraz, černobílý, převod, aproximace pozadí, prahování

ABSTRACT

The Bachelor's paper is concerned with processing black-and-white, grey-toned documents, which means the conversion of grey-toned picture of the document into binary one. In the paper, the procedures of separating the variable-brightness text from the background are described. Diverse methods of detaching the foreground from the background, the so-called picture segmentation, are used. Moreover, the outcomes of different methods are compared.

KEYWORDS

programming, C++, picture, conversion, black-and-white, background approximation, thresholding

BÁRNET, L. Převod černobílých dokumentů. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 73 s. Vedoucí bakalářské práce Ing. Miloslav Richter, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Převod černobílých obrazů“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu své bakalářské práce Ing. Miloslavu Richterovi Ph.D. za velkou ochotu a trpělivost při konzultacích a za čas, který mi věnoval.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	12
1 Řešení studentské práce	13
1.1 Počítačové vidění	13
1.1.1 Snímání obrazu	14
1.1.2 Předzpracování obrazu	15
1.1.3 Segmentace	15
1.1.4 Popis nalezených objektů	16
1.1.5 Porozumění obsahu obrazu	16
1.2 Pořízení obrazu	17
1.3 Předzpracování obrazu	18
1.3.1 Gaussův filtr	18
1.3.2 Filtr průměrováním	20
1.3.3 Mediánový filtr	20
1.3.4 Filtr rotující maskou	22
1.3.5 Cannyho hranový detektor	23
1.3.6 Sobelův operátor	25
1.3.7 Shrnutí kapitoly	25
1.4 Segmentace obrazu	26
1.4.1 Procentní prahování	26
1.4.2 Poměrné procentní prahování	28
1.4.3 Aproximace pozadí	29
1.4.4 Segmentace rozvodím	29
1.4.5 Shrnutí kapitoly	30
1.5 Morfologické transformace	31
1.5.1 Dilatace	31
1.5.2 Eroze	32
1.5.3 Otevření	32
1.5.4 Uzavření	32

1.6	OpenCV	33
2	Výsledky studentské práce	34
2.0.1	Rozbor úlohy	34
2.1	Možné typy obrázků	34
2.2	Rozdělení obrazu na části	36
2.3	Metody předzpracování obrazu	37
2.3.1	Gaussův filtr	37
2.3.2	Filtr průměrováním	38
2.3.3	Mediánový filtr	39
2.3.4	Filtr rotující maskou	39
2.3.5	Shrnutí kapitoly	40
2.4	Metody segmentace obrazu	41
2.4.1	Segmentace procentním prahováním	41
2.4.2	Segmentace poměrným procentním prahováním	42
2.4.3	Aproximace pozadí pomocí funkce	43
2.4.4	Otsuho metoda	51
2.4.5	Metoda založená na narůstání oblastí	52
2.4.6	Shrnutí kapitoly	54
3	Závěr	56
	Reference	58
	Seznam příloh	60
A	Ukázky obrázků	61
B	Vývojové diagramy	65
C	Elektronická příloha	73

SEZNAM OBRÁZKŮ

1.1	Postup při zpracování obrazu.	14
1.2	Původní (nefiltrovaný) obrázek [7].	18
1.3	Původní obrázek znehodnocený šumem.	19
1.4	Obrázek, na kterém byl aplikován GF s maskou 3x3 px.	19
1.5	Obrázek, na kterém byl aplikován GF s maskou 7x7 px.	19
1.6	Obrázek, na kterém byl aplikován filtr průměrováním s maskou 3x3 px.	20
1.7	Obrázek, na kterém byl aplikován mediánový filtr s maskou 3x3 px.	21
1.8	Standardní maska 3x3 px a příklad nestandardní masky v okolí 5x5 px.	21
1.9	Na obrázek byl aplikován mediánový filtr s nestandardní maskou 5x5 px.	21
1.10	Funkce filtru rotující maskou v okolí 5x5 px s maskou 3x3 px.	22
1.11	Obrázek, na který byl aplikován filtr rotující maskou.	22
1.12	Výsledek aplikace Cannyho hranového detektoru.	24
1.13	Výsledný obraz po aplikaci Sobelova operátoru h_1	25
1.14	Originální obrázek před segmentací.	27
1.15	Obrázek po použití procentního prahování podle nejnižší hodnoty a nastavený práh.	27
1.16	Obrázek po použití procentního prahování podle nejvyšší hodnoty a nastavený práh.	28
1.17	Obrázek po použití poměrného procentního prahování a nastavený práh.	28
1.18	Výsledný obrázek a pozadí (hodnoty aproximační funkce) po aproximaci funkcí.	29
1.19	Původní obrázek a obrázek po dilataci.	31
1.20	Původní obrázek a obrázek po erozi.	32
2.1	Obrázek, který se postupně ztmavuje.	35
2.2	Obrázek, který byl prahován ve čtyřech oddělených částech a barvy prahů.	36
2.3	Dokument, na kterém byl použit Gaussův filtr.	38
2.4	Dokument, na kterém byl použit filtr průměrováním.	38
2.5	Dokument, na kterém byl použit mediánový filtr.	39

2.6	Dokument, na kterém byl použit filtr rotující maskou (na nalezené okolí byl použit Gaussův filtr).	40
2.7	Dokument, na kterém byla použita segmentace procentním prahováním. .	41
2.8	Dokument, na kterém byla použita segmentace poměrným procentním prahováním.	42
2.9	Výsledný obrázek a pozadí (hodnoty aproximační funkce) pro aproximaci funkcí 1. řádu.	45
2.10	Výsledný obrázek a pozadí (hodnoty aproximační funkce) pro aproximaci funkcí 2. řádu.	46
2.11	Výsledný obrázek a pozadí (hodnoty aproximační funkce) pro aproximaci funkcí 3. řádu.	47
2.12	Dokument, na kterém byla použita segmentace aproximační funkcí 3. řádu s dvěma průchody a hodnoty aproximační funkce.	48
2.13	Dokument, na kterém byla použita segmentace aproximační funkcí 3. řádu s třemi průchody a hodnoty aproximační funkce.	49
2.14	Dokument, na kterém byla použita segmentace aproximační funkcí 3. řádu s třemi průchody a překrýváním oblastí a hodnoty aproximační funkce. .	50
2.15	Dokument, na kterém byla použita segmentace Otsuho metodou.	52
2.16	Dokument, na kterém byla použita metoda založená na narůstání oblastí a barevně označené nalezené oblasti.	54
A.1	Obrázek, který má světlejší střed a tmavší rohy.	61
A.2	Obrázek, který má světlejší rohy a tmavší střed.	61
A.3	Obrázek, na kterém je ostrý stín.	62
A.4	Obrázek, který má vysokou hladinu šumu.	62
A.5	Obrázek s ostrým stínem, na kterém bylo použito procentní prahování. . .	62
A.6	Obrázek s ostrým stínem, na kterém bylo použito poměrné procentní prahování.	63
A.7	Obrázek s ostrým stínem, na kterém byla použita aproximace funkcí 3. řádu se třemi průchody a překrýváním oblastí a pozadí.	63
A.8	Obrázek s ostrým stínem, na kterém byla použita Otsuho metoda.	64

A.9	Obrázek s ostrým stínem, na kterém byla použita metoda založená na narůstání oblastí.	64
B.1	Vývojový diagram Gaussova filtru.	65
B.2	Vývojový diagram mediánového filtru.	66
B.3	Vývojový diagram filtru rotující maskou.	67
B.4	Vývojový diagram segmentace procentním prahováním.	68
B.5	Vývojový diagram segmentace poměrným procentním prahováním. . . .	69
B.6	Vývojový diagram segmentace aproximační funkcí.	70
B.7	Vývojový diagram segmentace Otsuho metodou.	71
B.8	Vývojový diagram metody založené na narůstání oblastí.	72

ÚVOD

Obrazové vjemy jsou jedny z nejdůležitějších informací, které člověk získává o svém okolí. Orientujeme se podle nich v prostoru, rozpoznáváme objekty atd.

Proto se člověk snaží naučit těmto dovednostem i stroje. Počítačové vidění je dnes využíváno ve výrobě pro kontrolu výrobků, pro navádění robotů a i k zjednodušení práce běžného uživatele na počítači.

Počítačové vidění pro uživatele na počítači se využívá hlavně pro OCR neboli Optické rozpoznávání znaků. Tyto programy fungují tak, že v naskenovaném obraze dokáží najít text a rozpoznat v něm znaky [1].

Tato práce se bude zabývat prvními kroky zpracování obrazu, tedy předzpracováním a segmentací obrazu. Budou probrány některé metody filtrace šumu v obraze, hranové detektory a metody segmentace.

V praktické části mé práce budou probrány metody řešení problémů zmiňované již v části teoretické. Budou zhodnoceny metody z první části a některé metody budou upraveny a/nebo vylepšeny pro účely zadání. Práce se proto nebude zabývat převáděním jakýchkoli obrázků, ale převodem vyfoceného/naskenovaného textu, který je zobrazen ve stupních šedi na binární. Bude se tedy jednat o počáteční kroky OCR programů.

Veškeré programy jsou psány v jazyce C++ při použití knihoven OpenCV¹.

¹Knihovny OpenCV budou probrány v jedné z následujících kapitol

1 ŘEŠENÍ STUDENTSKÉ PRÁCE

1.1 Počítačové vidění

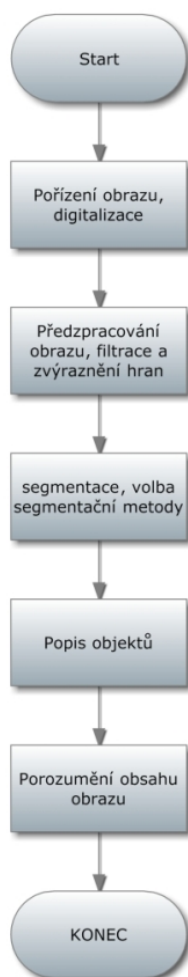
Počítačové vidění je obor, který se snaží technickými prostředky alespoň částečně napodobit lidské vidění. Lidé vnímají obrazové vjemy díky očím. Tyto obrazy potom vyhodnocují pomocí inteligence, která umožňuje reprezentovat nabyté znalosti. Díky tomu dokážeme pojmenovat věci, které vidíme.

Postupy, které se používají v počítačovém vidění jsou velmi složité. Jádrem postupů jsou znalostní systémy a umělá inteligence. Této části počítačového vidění se říká **vyšší úroveň**.

Další částí počítačového vidění je **nižší úroveň**. Cílem této úrovně je analyzovat vstupní obraz a najít informace potřebné pro vyšší úroveň.

Postup pro zpracování obrazů se dá rozložit do základních kroků:

Tato práce se bude zabývat hlavně předzpracováním obrazu a jeho segmentací.



Obrázek 1.1: Postup při zpracování obrazu.

1.1.1 Snímání obrazu

Snímání obrazu je prvním krokem ve zpracování obrazu. V této podkapitole budou popsány způsoby pořízení obrazu. Budou také vysvětleny pojmy jako kvantování a vzorkování.

Abychom obraz mohli uložit v číselné formě do počítače, musíme použít nějaký snímač, který převede obraz na signál, se kterým dokáže počítač pracovat.

Funkce snímače:

Snímačem je snímána vstupní, optická veličina. Tento snímač ji převádí na spojitý, elektrický signál. Tento signál je poté nakvantován a navzorkován.

Vzorkování je prvním krokem digitalizace obrazu. Vzorkování znamená, že celý obraz rozdělíme na velmi malé oblasti (počet oblastí je dán vzorkovací maticí $M \times N$) a každé oblasti přiřadíme pomocí **kvantování** hodnotu určité barvy.

Kvantování je převádění hodnot spojité funkce na digitální. Problémem digitalizace obrazů (či kterýchkoli veličin) je to, že vstupní obraz (veličina) je po celou dobu spojitý. V počítačové technice musíme ale vyjádřit každou hodnotu tak, aby s ní mohl počítač pracovat. Musíme proto vytvořit určité rozptyly barev, pro které budeme předpokládat, že barva je stejná, i když se ve skutečnosti nepatrně změnila. Tyto rozdíly jsou tak malé, že si jich lidské oko nevšimne.

Pro monochromatické snímání je potřebný jen jeden snímač (snímání jasu), ale pro barevné snímání potřebujeme minimálně tři snímače, které zajišťují snímání tří hlavních spektrálních složek (červená, zelená, modrá).

Nejmenším prvkem v obrazu je **pixel**. Místo tohoto pojmu se také používá termín bod. Rozdíl mezi pixelem a bodem je v tom, že pixel má vždy konečné rozměry, ale bod je nekonečně malý. Po vzorkování se z oblastí, které jsme určili při vzorkování, stávají pixely.

Tato podkapitola se zabývala digitalizací obrazu a jeho snímáním. Možnostmi snímání obrazu se zabývá kapitola 1.2.

1.1.2 Předzpracování obrazu

Předzpracování obrazu má za úkol odstranit zkreslení a šum vzniklé při digitalizaci.

Při předzpracování můžeme také hledat rysy v obrazu, které jsou důležité pro další zpracování (například vyhledávání hran v obraze).

Problematikou předzpracování obrazu se zabývá kapitola 1.3.

1.1.3 Segmentace

Segmentace obrazu je skupina metod, které mají za úkol detekovat v obraze oblasti s podobnými vlastnostmi. Typickým cílem segmentace obrazu je identifikace popředí a určení oblastí v obraze, které odpovídají významnému prvku zachycené scény (v této práci se jedná o oddělení textu od pozadí) [2].

Tato práce se bude problematikou segmentace zabývat v kapitole 1.4.

1.1.4 Popis nalezených objektů

Objekty v obraze lze popsat několika způsoby. Hlavní dělení je na popis **kvantitativní** (pomocí číselných charakteristik) nebo **kvalitativní** (pomocí relací mezi objekty).

Za nejjednodušší popis lze považovat stanovení velikosti objektu (počtu jemu odpovídajících pixelů) [3].

1.1.5 Porozumění obsahu obrazu

Porozumění je posledním krokem zpracování obrazu. V nejjednodušším případě chápeme porozuměním zařazení objektů do jednotlivých, předem známých skupin (například podle velikosti nebo roztržďení objektů na hranaté a kulaté) [3].

Pomocí tohoto postupu by se dal zpracovat kterýkoli obraz. Tato práce se bude dále zabývat pouze druhým a třetím krokem, tedy předzpracováním a segmentací obrazu.

1.2 Pořízení obrazu

Abychom mohli obraz začít zpracovávat, musíme nejprve obraz pořídit nebo ho digitalizovat tak, aby se dal zpracovat.

Pořízení obrazu digitálním fotoaparátem

Obrazem může být fotografie, kterou z digitálního fotoaparátu můžeme pohodlně stáhnout přes kabel nebo pomocí čtečky paměťových karet. Pro některé případy může být tento způsob získání obrazu nevýhodný, protože fotografie má veliké rozlišení (v současné době není fotoaparát s 10 Mpx žádnou výjimkou, takže zpracování obrazu je potom programově náročnější a tedy delší. Další nevýhodou pořízení obrazu digitálním fotoaparátem je to, že většinou má obraz špatnou kvalitu – na papíru jsou vidět stíny, obraz je nedostatečně osvětlen atd.

Pořízení obrazu skenerem

Podle mého názoru je lepší přenášet již existující obrazy (nebo dokumenty) do počítače pomocí skeneru. Tato varianta je sice pro většinu lidí dražší (velká část populace má digitální fotoaparát doma, ale skener ne), ale obraz pořízený skenerem má lepší kvalitu.

Výhodou tohoto přenosu je konstantní osvětlení převáděného obrazu, skener svým víkem také částečně narovná ohyby atd. Tento způsob je výhodný i proto, že výstupní obrázek nebývá tak velký, a proto jeho pozdější zpracování netrvá tak dlouho. Nevýhodou je nepřenosnost skeneru, takže obrazy můžeme digitalizovat jen doma.

Tato kapitola se zabývala způsoby pořízení obrazu, jejich výhodami a nevýhodami. Pokud už máme obrázek uložený, můžeme ho začít zpracovávat.

1.3 Předzpracování obrazu

Prvním krokem, který musíme pro zpracování obrazu po digitalizaci udělat, je předzpracování. Nejčastěji si můžeme pod předzpracováním představit aplikaci některého z filtrů nebo hranových detektorů.

1.3.1 Gaussův filtr

Gaussův filtr (GF) je jedním z nejúčinnějších filtrů, které se používají. Filtr má (jak je patrné z názvu) tvar Gaussovy křivky. Ta je popsána funkcí 1.1 [4]:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1.1)$$

kde σ je směrodatná odchylka,

x, y jsou souřadnice bodu (bod, pro který je GF počítán má souřadnice $(x, y) = [0, 0]$).

Hlavní nevýhodou GF je fakt, že rozmazává hrany. To může být problémem při následné detekci hran, které jsou díky použití filtru znehodnoceny.

Gaussův filtr eliminuje šum lépe pro větší použité masky, ale na druhou stranu při větších maskách GF více rozmazává obraz, jak je vidět na obrázcích 1.3, 1.4 a 1.5 [3] [5][6]. Pro tuto kapitolu byl pro ukázkou funkce filtrů a hranových detektorů použit obrázek Leny [7]. Tento obrázek byl zvolen proto, že se stal jakýmsi standardem pro ukázkou funkce předzpracování obrazu.



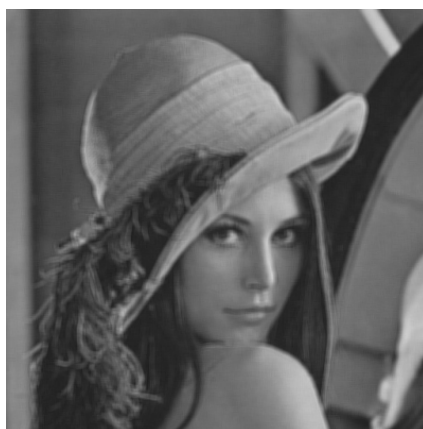
Obrázek 1.2: Původní (nefiltrovaný) obrázek [7].



Obrázek 1.3: Původní obrázek znehodnocený šumem.



Obrázek 1.4: Obrázek, na kterém byl aplikován GF s maskou 3x3 px.



Obrázek 1.5: Obrázek, na kterém byl aplikován GF s maskou 7x7 px.

1.3.2 Filtr průměrováním

Filtr vychází ze stejného vzorce jako GF (rovnice 1.1). Rozdíl mezi těmito filtry je v tom, že pro filtr průměrováním se $\sigma \rightarrow \infty$.

Tento filtr má samozřejmě klady i zápory jako GF – rozmazává tedy hrany.



Obrázek 1.6: Obrázek, na kterém byl aplikován filtr průměrováním s maskou 3x3 px.

1.3.3 Mediánový filtr

Medián je hodnota prvku ležícího uprostřed řady vzestupně (sestupně) řazených hodnot. Tedy leží vedle něj stejné množství menších (nebo rovných) prvků a větších (nebo rovných) prvků. Pro řady, které mají sudý počet členů se medián vypočítá jako aritmetický průměr dvou prostředních prvků ($\frac{n}{2}$ a $\frac{n+1}{2}$).

Mediánový filtr určí jas bodu tak, že vypočítá medián jasu bodů v okolí (například okolí 5x5) vstupního obrazu.

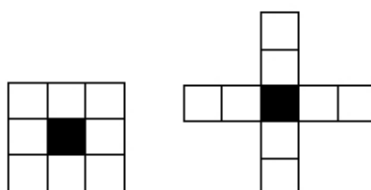
Tento filtr nerozmazává hrany tolik jako GF a dobře potlačuje impulsní šum, ale porušuje tenké čáry.



Obrázek 1.7: Obrázek, na kterém byl aplikován mediánový filtr s maskou 3x3 px.

Řešením tohoto problému je nepoužívat standardní masky, které zahrnují celé okolí, ale jen jeho části (obrázek 1.8) [3].

Výsledek po použití nestandardní masky je vidět na obrázku 1.9.



Obrázek 1.8: Standardní maska 3x3 px a příklad nestandardní masky v okolí 5x5 px.

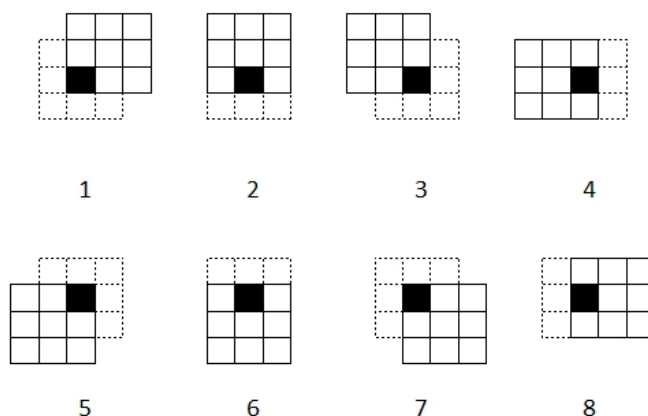


Obrázek 1.9: Na obrázek byl aplikován mediánový filtr s nestandardní maskou 5x5 px.

1.3.4 Filtr rotující maskou

Další filtr, který se pro předzpracování obrazu používá je **filtr rotující maskou**. Tento filtr funguje tak, že v okolí bodu 5x5 vyhledává takovou masku 3x3, která nejpravděpodobněji patří do stejného obrazce jako bod. Filtr tedy hledá masku, jejíž jasový průměr je nejblíže k hodnotě jasu hledaného bodu. Ukázka masek je na obr. 1.10. Masek, které filtr prozkoumává, je celkem devět. Osm z nich je na obr. 1.10 a devátou maskou je 3x3 okolí bodu, pro který je přepočítáván jas (na obrázku je vyznačen černě) – tato maska je na obrázku vyznačena tečkovaně.

Na obrázku 1.11 je vidět výsledek filtrace rotující maskou.



Obrázek 1.10: Funkce filtru rotující maskou v okolí 5x5 px s maskou 3x3 px.



Obrázek 1.11: Obrázek, na který byl aplikován filtr rotující maskou.

1.3.5 Cannyho hranový detektor

Jak bylo napsáno v předchozí kapitole, do předzpracování obrazu patří také detekce hran v obrazu. V této práci bude zmíněn pouze jeden hranový detektor – Cannyho.

Tento detektor je optimální hlavně pro hledání skokových hran vzhledem ke třem kritériím. První kritérium (detekční) zajišťuje, aby detektor neopomenul významné hrany, druhé (lokalizační) kritérium se stará o to, aby byl rozdíl mezi skutečnou a nalezenou pozicí hrany nejmenší a třetí kritérium zajišťuje, aby detektor nereagoval na jednu hranu vícenásobně.

Cannyho detektor pracuje s druhými derivacemi obrazové funkce a jeho výsledkem jsou velikosti i směry hran.

Funkce Cannyho hranového detektoru:

- eliminace šumu (např. Gaussovým filtrem – viz kapitola 1.3.1);
- určení gradientu (první derivace obrazové funkce);
- nalezení lokálních maxim;
- eliminace nevýznamných hran [8].

Prvním krokem práce Cannyho hranového detektoru je odstranit šum. Tímto problémem se tato práce zabývá výše.

Druhým krokem je určení gradientu, tedy detekce hran. Nejčastěji se na hledání hran používá Sobelův operátor (ten bude probrán níže). Výstupem Sobelova operátoru je nejen velikost, ale i směr gradientu.

Jako další krok provádí Cannyho detektor hledání lokálních maxim, díky čemuž ztenčí hrany. Detektor tedy hledá bod, který má po směru i proti směru gradientu menší sousedy. Tento bod je potom označen jako hrana.

Posledním krokem je eliminace nevýznamných hran. Ta probíhá pomocí prahování s hysterezí. Určíme tedy, jaká hodnota gradientu je významná. Zvolíme minimální (T1) a maximální (T2) hodnotu (prahy), mezi kterými může gradient kolísat. Pokud hodnota gradientu daného pixelu leží nad větším prahem T2, je přímo označen jako hranový. Pokud posuzujeme bod, jehož hodnota leží mezi T1 a T2, pak je označen jako hrana jedině pokud sousedí s bodem, který už byl jako hrana označen dříve [8].



Obrázek 1.12: Výsledek aplikace Cannyho hranového detektoru.

1.3.6 Sobelův operátor

Sobelův operátor je metoda, pomocí které můžeme najít hrany v obrázku. Je to gradientní metoda, která aproximuje první parciální derivace.

Sobelův operátor je směrově závislý (díky použití první derivace). Použití tento operátor můžeme pro různé rozměry masek (rovnice 1.2 je ukázka masek 3x3).

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad (1.2)$$

Ukázka použití Sobelova operátoru je na obrázku 1.13. Jak je vidět, jedna maska neobsáhne všechny hrany v obrázku. Proto se využívá Sobelův operátor s různými maskami, díky čemuž nalezneme všechny hrany.

Sobelův operátor je používán také proto, že je málo citlivý na šum [3].



Obrázek 1.13: Výsledný obraz po aplikaci Sobelova operátoru h_1 .

1.3.7 Shrnutí kapitoly

V této kapitole jsme tedy probrali některé ze způsobů předzpracování obrazu. Byly uvedeny způsoby odfiltrování šumu, z nichž některé byly v praktické části práce použity.

Dále byl popsán Cannyho hranový detektor a Sobelův operátor, jenž jsou velice dobře použitelné v praktických příkladech pro hledání hran.

1.4 Segmentace obrazu

V předchozí kapitole jsme probrali předzpracování obrazu, tudíž jsme z obrazu od-filtrovali šum a/nebo jsme našli v obrazu hrany. Dalším krokem pro zpracování obrazu je tedy jeho segmentace.

V této kapitole budou tedy probrány některé možnosti segmentace obrazu (samozřejmě nemohou být uvedeny všechny možnosti a způsoby řešení, a proto budou probrána jen některá řešení).

1.4.1 Procentní prahování

Segmentace obrazu procentním prahováním je pravděpodobně nejjednodušším způsobem. Procentní prahování může mít dvě řešení: prahování podle **nejnižší hodnoty** nebo prahování podle **nejvyšší hodnoty**.

Procentní prahování podle nejnižší hodnoty

Procentní prahování podle nejnižší hodnoty funguje tak, že najdeme nejnižší hodnotu jasu v obraze. V šedotónovém obrazu má černá barva jas #0 a bílá barva jas #255 (dekadicky nebo FF hexadecimálně). Tuto nejnižší hodnotu jasu vynásobíme číslem větším než 1. Díky tomu dostaneme hodnotu prahu.

Po nalezení prahu probíhá rozhodování – pokud má jas bodu vyšší hodnotu než je hodnota prahu, potom se jedná o bílou barvu (jas #255). Pokud má jas bodu nižší hodnotu, přiřadíme ji hodnotu černé (jas #0).

Výsledek prahování podle nejnižší hodnoty je na obrázku 1.15.

Základní obraz, který bude používán pro tuto kapitolu je na obrázku 1.14.



Obrázek 1.14: Originální obrázek před segmentací.



Obrázek 1.15: Obrázek po použití procentního prahování podle nejvyšší hodnoty a nastavený práh.

Procentní prahování podle nejvyšší hodnoty

Procentní prahování podle nejvyšší hodnoty funguje tak, že najdeme nejvyšší hodnotu jasu a poté ji vynásobíme číslem menším než 1. Díky tomu získáme práh pro rozhodování o výsledné barvě. Poté porovnáváme hodnotu v každém bodě s hodnotou prahu. Pokud je hodnota menší, potom přiřadíme bodu hodnotu 0 (černá), pokud je hodnota větší, přiřadíme bodu hodnotu 255 (bílá).

Výsledek procentního prahování podle nejvyšší hodnoty obrázku 1.14 je na obrázku 1.16.



Obrázek 1.16: Obrázek po použití procentního prahování podle nejvyšší hodnoty a nastavený práh.

1.4.2 Poměrné procentní prahování

Poměrné procentní prahování je založeno, jak je již z názvu zřejmé, na stejném principu jako předchozí kapitola (1.4.1).

Rozdíl mezi těmito metodami je v tom, že procentní prahování hledá pouze maximum (minimum) v obraze a podle toho se rozhoduje o velikosti prahu. Vylepšené procentní prahování najde maximum i minimum v obraze. Odečte minimum od maxima a poté pracuje s tímto rozdílem. Díky tomu se lépe nasegmentuje i velmi tmavý (světlý) obraz.

Ukázka obrázku, který byl zpracován pomocí vylepšeného procentního prahování je na obrázku 1.17.



Obrázek 1.17: Obrázek po použití poměrného procentního prahování a nastavený práh.

1.4.3 Aproximace pozadí

Dalším ze způsobů segmentace obrazu je aproximace pozadí. Jedná se o složitější metodu než je procentní prahování, ale na druhou stranu použitelnou v prakticky jakémkoli obraze, což neplatí pro procentní prahování.

Aproximace pozadí spočívá v tom, že si určíme řád dvojrozměrné funkce, který chceme použít. K této funkci poté vypočítáme pomocí jedné z numerických metod [9]. Po dopočítání koeficientů dosazujeme do rovnice souřadnice bodu. Vyřešíme rovnici a dostaneme hodnotu jasu, kterou by měl mít podle funkce bod. Pokud je skutečná hodnota jasu menší, potom se jedná o černou ($\#0$) a pokud je skutečná hodnota větší, jedná se o bílou ($\#255$). Obrázek 1.18 ukazuje segmentaci pomocí aproximace funkcí.



Obrázek 1.18: Výsledný obrázek a pozadí (hodnoty aproximační funkce) po aproximaci funkcí.

1.4.4 Segmentace rozvodím

Segmentace rozvodím je dalším způsobem pro segmentaci obrazu.

Metoda je založena na představě, že zaplavujeme vodou všechna místa, která leží níž než hladina vody. Hledáme tedy oblasti "zalité" vodou. Tyto oblasti jsou rozděleny skupinami pixelů, které mají vyšší hodnotu jasu.

1.4.5 Shrnutí kapitoly

Tato kapitola se zabývala problémem segmentace obrazu – oddělením pozadí od objektů, které pro nás mají nějakou vypovídající hodnotu.

Výpočetně nejnáročnější metoda pro segmentaci obrazu je **aproximace pozadí funkcí**. Náročnost metody spočívá v tom, že například rovnice třetího řádu má 16 koeficientů, které musíme vypočítat, což je výpočetně náročné. Řád funkce se hledá tak, že vyzkoušíme vyšší řád. Pokud se výsledek zlepší, zkusíme zvýšit řád znovu.

Naopak výpočetně nejméně náročná je segmentace obrazu **procentním prahováním**. Rychlost metody je její velkou výhodou. Nevýhodou je ale to, že pro obrázky, ve kterých se hodně mění jas, je prakticky nefunkční.

1.5 Morfologické transformace

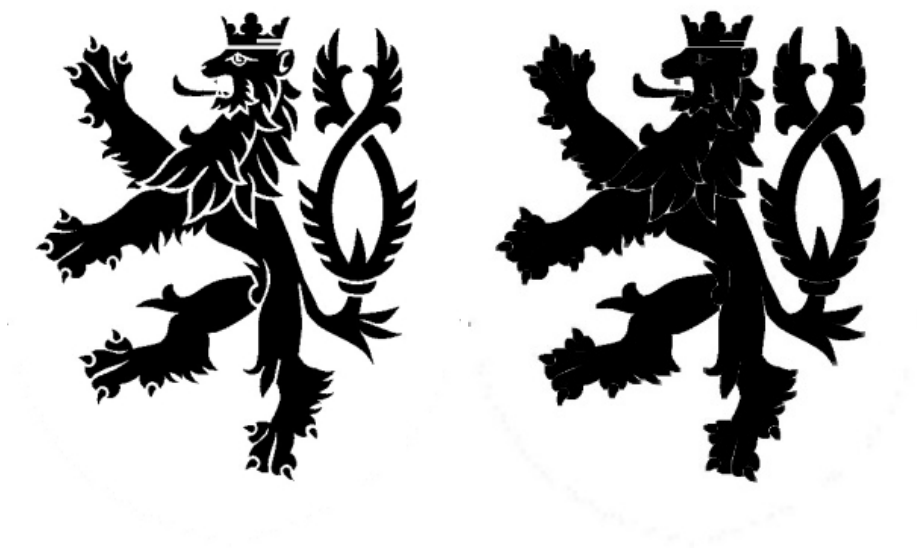
Morfologické transformace jsou postupy, při nichž se upravuje již vysegmentovaný obraz. Tyto transformace se provádějí až po segmentaci proto, že jejich použití pro šedotónový obraz by bylo velmi náročné.

Morfologické transformace se používají hlavně k zjednodušení tvaru objektů a zdůraznění jejich struktur.

1.5.1 Dilatace

Dilatace je operace, kdy k obrazu přičítáme jiný obraz. Musíme tedy chápat obraz jako matici čísel vyjadřujících barvu. K této matici přičítáme (Minkowského množinovým součtem) takzvanou **masku**, což je další matice, která má požadovaný rozměr (například jsou používány matice 2x1 nebo 3x3). Dilatace tedy zaplní díry (mezery v obraze), které mají šířku jednoho bodu a rozšíří celý obraz o jeden bod [3].

Výsledek dilatace je na obrázku 1.19. Pro ukázkou funkce dilatace a eroze byl použit Český lev.



Obrázek 1.19: Původní obrázek a obrázek po dilataci.

1.5.2 Eroze

Eroze je duální transformací k dilataci (není její inverzní transformací!). Jedná se o operaci, kdy od matice obrazu odečítáme Minkowského rozdílem matici masky. Eroze odstraní objekty, které mají tloušťku jednoho bodu a ztenčí celý obraz o celý bod.

Pomocí eroze je možné získat obrysy obrazu, a to tak, že od původního obrazu odečteme jeho erozi [3].

Výsledek eroze je na obrázku 1.20.



Obrázek 1.20: Původní obrázek a obrázek po erozi.

1.5.3 Otevření

Otevření je funkce, která je kombinací dilatace a eroze. Eroze, po které následuje dilatace, se nazývá **otevření**.

Otevření oddělí objekty, které jsou spojeny objektem o šířce 1 bodu. Tím dosáhneme zjednodušení struktury objektů [3].

1.5.4 Uzavření

Uzavření je funkce, která je kombinací dilatace a eroze. Dilatace, po které následuje eroze, se nazývá **uzavření**. Uzavření zaplní malé díry a také vyhladí obrys objektu [3].

1.6 OpenCV

Práce se v předešlých kapitolách zabývala metodami, jak zpracovat obraz. Pro programování metod probraných v předchozích kapitolách ale musíme použít některé knihovny pro zpracování obrazu.

Knihovny, které budou v práci probrány se nazývají **OpenCV**.

OpenCV (Open Computer Vision) jsou knihovny s otevřeným zdrojovým kódem, který spadá pod licenci **BSD** [10]. Licence BSD je jedna z nejmírnějších mezi otevřenými licencemi. Umožňuje šíření obsahu bez omezení pouze s tím, že musí být uveden autor a informace o licenci spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

Knihovny OpenCV jsou napsány v jazyce C a C++ a jsou funkční pod operačními systémy Linux, Windows¹ a Mac OS X.

Je zřejmé, že pokud jsou knihovny napsány v C a C++, budou fungovat v těchto jazycích. Navíc je zde podpora jazyků Python, Ruby, Matlab atd.

Knihovny jsou optimalizovány pro rychlé a efektivní zpracování obrazu v aplikacích reálného času.

Knihovny mají kolem 500 funkcí, mezi které patří načtení a vytvoření obrázku, dilatace a eroze, zobrazení obrázku atp. Díky tomu výrazně urychlily řešení mého projektu, protože načtení hodnot barev bez těchto knihoven by bylo velice obtížné, o zobrazení a vytvoření obrázku nemluvě [11].

¹Z mé vlastní zkušenosti je však nutno dodat, že pod Windows 7 a Visual Studiem 2008 knihovny nefungují. Je nutné doinstalovat několik rozšíření do Visual Studia a poté spustit ještě několik projektů vytvořených při instalaci OpenCV, a ani to není zárukou správné funkce. Je také možné, že ve verzi 2.0 již knihovny fungují.

2 VÝSLEDKY STUDENTSKÉ PRÁCE

V předchozí (teoretické) části práce byla probrána teoretická východiska k předzpracování a segmentaci obrazu. V této části se budu zabývat využíváním již probraných metod a jejich porovnáním na skutečných dokumentech. V práci jsem zpracoval 5 metod pro segmentaci obrazu: prahování, poměrné prahování, Otsuho metoda [12], aproximace pozadí funkcí a metoda založená na narůstání oblastí.

2.0.1 Rozbor úlohy

Prvním krokem, který musíme před řešením úlohy udělat, je rozbor úlohy. Musíme se rozhodnout, jakou metodu zvolíme a o jaký typ obrázku se jedná (černo-bílý nebo bílo-černý).

2.1 Možné typy obrázků

V této kapitole probereme možné typy obrázků, které mohou být vstupem coby obraz tištěného dokumentu.

Vstupní obrázek může mít:

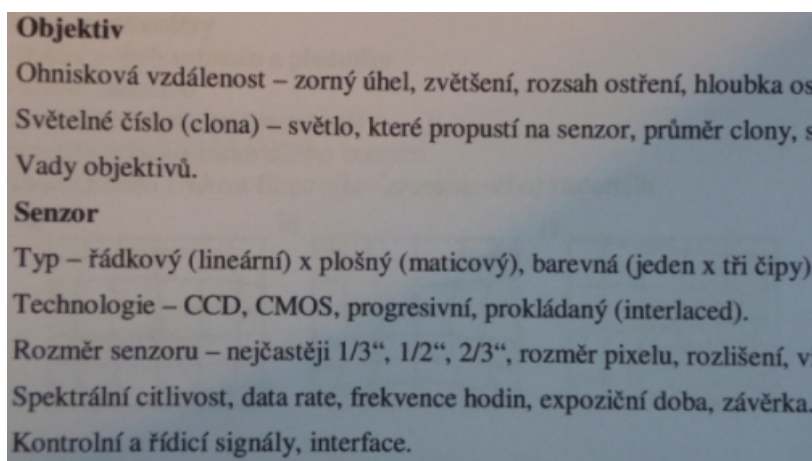
- tmavý text na světlém pozadí;
- světlý text na tmavém pozadí;
- světlé rohy, tmavý střed;
- světlý střed, tmavé rohy;
- ostrý stín;
- plynulé přechody tmavá – světlá.

Tyto vady obrázků se vyskytují hlavně při pořizování obrazu fotoaparátem a lze je prakticky libovolně kombinovat. Při pořizování skenerem bývají obrázky velice kvalitní.

Vady na obrázku mohou být způsobeny například zmačkáním papíru, pokud fotíme fotoaparátem a použijeme blesk, potom bývá střed obrázku přesvětlen a okraje proto bývají tmavší, naopak pokud osvětlujeme obrázek nějakým světelným zdrojem, bývá jeden okraj obrázku světlejší než ostatní atd.

Na obrázku 2.1 budu ukazovat výsledky jednotlivých metod. Další ukázky obrázků, které mají vady uvedené výše jsou v příloze (A.1, A.2, A.3, A.4).

Pro kapitolu předzpracování obrazu (kapitola 2.3) použiji obrázek A.4, protože na něm je nejvíce šumu. Obrázky jsou foceny z [6] digitálním fotoaparátem.



Obrázek 2.1: Obrázek, který se postupně ztmavuje.

2.2 Rozdělení obrazu na části

Rozdělení obrazu je často využívanou metodou při zpracování obrazu. V dnešní době, kdy nejsou problémem fotografie o rozlišení 10 Mpx by bylo zpracování celého obrazu velice náročné na paměť. Proto se obraz rozdělí na části, které se dále zpracovávají odděleně.

Díky tomu je zpracování úspornější na paměť a také přesnější, protože například určení prahu pro celý obrázek nemusí být vždy výhodné, protože hodnota jasu "černé" barvy v horním levém rohu může být stejná, jako hodnota jasu "bílé" barvy v pravém dolním rohu. Pokud bychom obraz nerozdělili, mohlo by díky tomu dojít ke špatné segmentaci obrazu a my bychom mohli ztratit důležité informace o části obrazu.

Velikost oblasti by se měla pohybovat v rozmezí 2 – 4 řádků textu, abychom měli v oblasti dostatečné množství písmen. Pokud bychom aproximovali obrázek pouze s pozadím, potom by mohly vznikat na výsledném obrázku chyby.

Na obrázku s klíči (obr. 1.14) si ukážeme, jak vypadá vysegmentovaný obrázek, pokud ho rozdělíme na 4 části a v každé z nich budeme počítat práh (použita byla metoda procentního prahování) – výsledek je na obrázku 2.2.

Jak je vidět, obrázek je vyprahován mnohem lépe, než tomu bylo u hledání prahu pro celý obraz. Proto v celé práci používám toto rozdělení obrazu na části. Dalším důvodem použití je již zmíněná úspora paměti a také problém, že při vytváření dvojrozměrného pole, které je větší než 500 x 500 pixelů, se objevuje chyba s alokací paměti.



Obrázek 2.2: Obrázek, který byl prahován ve čtyřech oddělených částech a barvy prahů.

2.3 Metody předzpracování obrazu

Teoretická východiska předzpracování obrazu jsme probrali v kapitole 1.3. V této kapitole zhodnotíme jednotlivé metody a jejich vhodnost pro segmentaci černobílých dokumentů.

Na rozdíl od teoretické části budou v této kapitole obrázky dokumentů – tedy naskenovaného textu.

2.3.1 Gaussův filtr

Gaussův filtr je jedním z nejlepších filtrů pro odstranění šumu.

Při zpracování obrázku se Gaussův filtr realizuje pomocí masky. Příklad masky je 2.1.

$$h_1 = \begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix} \quad (2.1)$$

Touto maskou vynásobíme okolí bodu, pro který masku počítáme. Tyto hodnoty sečteme a poté vydělíme součtem hodnot v masce (v případě masky 2.1 je jich 36) a získáme novou hodnotu bodu.

Jak je vidět, největší váhu má bod, pro který počítáme novou hodnotu. Naopak nejmenší váhu mají hodnoty na diagonálách.

Vývojový diagram pro aplikaci Gaussova filtru je na obrázku B.1. Výsledek po použití Gaussova filtru je na obrázku 2.3.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka ostření
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, světelná
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy), monochromatická
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
Rozměr senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, viditelnost
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.

Obrázek 2.3: Dokument, na kterém byl použit Gaussův filtr.

2.3.2 Filtr průměrováním

Z Gaussova filtru vychází filtr průměrováním. Tento filtr funguje stejně jako Gaussův filtr, jen má masku ve tvaru:

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (2.2)$$

Problém tohoto filtru je, že rozmazává obraz ještě více než Gaussův filtr (viz obrázek 2.4).

Vývojový diagram je stejný jako pro Gaussův filtr (B.1).

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka ostření
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, světelná
Vady objektivů.

Senzor

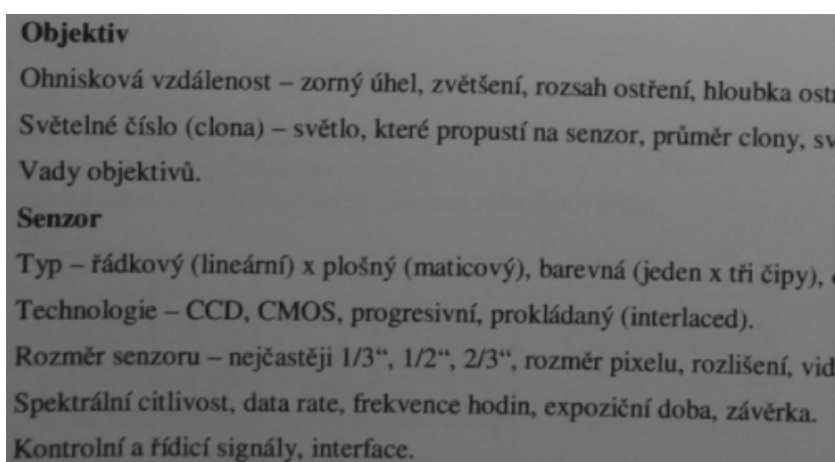
Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy), monochromatická
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
Rozměr senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, viditelnost
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.

Obrázek 2.4: Dokument, na kterém byl použit filtr průměrováním.

2.3.3 Mediánový filtr

Mediánový filtr pracuje tak, že seřadí hodnoty jasu bodů v okolí bodu, kterému budeme přiřazovat novou hodnotu, podle velikosti a poté vybere prvek, který leží ve středu seznamu a tuto hodnotu přiřadí. Filtr dobře potlačuje impulsní šum, ale jeho výsledky nejsou nejlepší. Některé větší chyby totiž filtr neodstraní. Výsledek mediánového filtru je na obrázku 2.5.

Vývojový diagram mediánového filtru je na obrázku B.2.



Obrázek 2.5: Dokument, na kterém byl použit mediánový filtr.

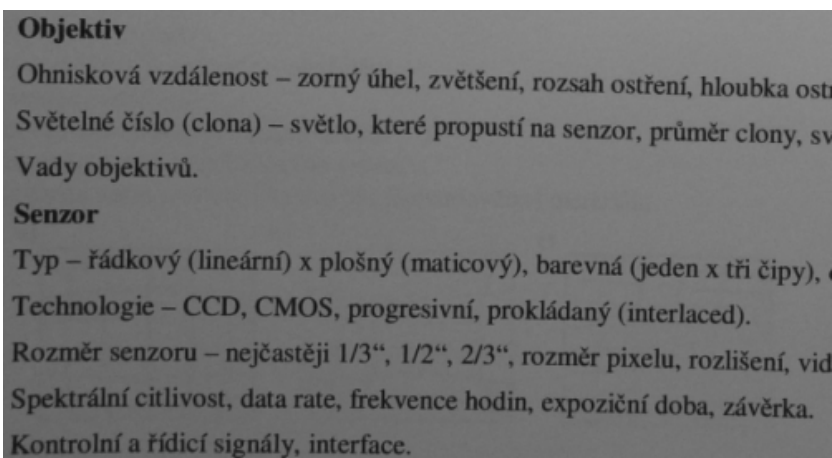
2.3.4 Filtr rotující maskou

Filtr rotující maskou je výpočetně náročnější než předchozí filtry, protože musí počítat více masek kvůli rozhodnutí, jakou z nich použije.

Hledáme tedy masku, jejíž průměr jasu je nejblíže k jasu bodu, na který aplikujeme filtr. Po nalezení masky, kterou budeme používat aplikujeme některý z filtrů (2.3.1, 2.3.2, 2.3.3).

Jako nejvhodnější filtr se mi jevil Gaussův filtr. Metoda rotující masky totiž částečně eliminuje jeho zápor – rozmazávání hran. Zároveň tento filtr dobře odstraňuje šum.

Výsledek filtru rotující maskou je na obrázku 2.6.



Obrázek 2.6: Dokument, na kterém byl použit filtr rotující maskou (na nalezené okolí byl použit Gaussův filtr).

Vývojový diagram filtru rotující maskou je na obrázku B.3.

2.3.5 Shrnutí kapitoly

V této kapitole jsme tedy probrali filtry pro odstranění šumu. Jako nejvhodnější se mi zdá filtr rotující maskou, který pro výsledný výpočet používá Gaussův filtr. Tato metoda nerozmazává hrany tolik, jako samotný Gaussův filtr, ale zachovává si jeho výhody – velice dobře odstraňuje šum.

Pokud bychom ale měli převádět obrázek, který má velice malá písmena, potom by byl pravděpodobně vhodnější mediánový filtr nebo filtr rotující maskou, kde byl pro výslednou masku použit mediánový filtr. (Použití rotující masky je v tomto případě podle mě zbytečné, protože výsledky jsou prakticky stejné.) Tento filtr totiž nerozmaže hrany písmen a výsledný text je potom lépe čitelný.

2.4 Metody segmentace obrazu

V této kapitole budou probrány postupy při programování segmentačních metod. Zhodnotíme také jednotlivé metody segmentace obrazu a jejich vhodnost pro jednotlivé typy obrázků.

2.4.1 Segmentace procentním prahováním

Segmentace procentním prahováním je nejjednodušší způsob určení prahu v obrazu. Jedná se o metodu, kdy najdeme maximální (minimální) hodnotu jasu v obraze (resp. v části obrazu, pro kterou právě hledáme práh) a tu vynásobíme číslem menším (větším) než jedna. Díky tomu získáme práh, podle kterého se rozhoduje, jestli se jedná o černý bod (jas $\#0$) a nebo o bílý bod (jas $\#255$).

Jak vypadá obrázek po segmentaci procentním prahováním je vidět na obrázku 2.7. Vývojový diagram procentního prahování je na obrázku B.4.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
Rozměr senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, v.
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.

Obrázek 2.7: Dokument, na kterém byla použita segmentace procentním prahováním.

Pro obrázky, které neobsahují žádné obrazové vady (viz obr. 2.1) je tato metoda dostačující. Metoda také vyžaduje, aby obraz neměl histogram posunut k jednomu extrému (například k bílé barvě (jas $\#255$) nebo k černé (jas $\#0$)).

2.4.2 Segmentace poměrným procentním prahováním

Segmentace poměrným procentním prahováním vychází – jak je z názvu patrné – z procentního prahování. Rozdíl je v tom, že nalezneme minimum i maximum. Z hodnoty maxima, podobně jako v procentním prahování, získáme hodnotu prahu a poté odečteme hodnotu minima od jasu aktuálního bodu. Hodnotu prahu porovnáme s tímto rozdílem

a podle toho rozhodneme, o jakou barvu se jedná.

Na obrázku 2.8 je vidět funkce segmentace poměrným prahováním.

Vývojový diagram poměrného procentního prahování je na obrázku B.5.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os

Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s

Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)

Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v

Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.

Kontrolní a řídicí signály, interface.



Obrázek 2.8: Dokument, na kterém byla použita segmentace poměrným procentním prahováním.

Metoda je univerzálnější než procentní prahování, protože se práh počítá z rozdílu minima a maxima, a proto metoda funguje i pro obrazy, které mají posunutý histogram k jednomu z extrémů. Černé body, které po segmentaci vznikají v "bílých" místech jsou způsobeny posunutím prahu velmi blízko k pozadí, a proto zde dochází k chybám. Řešením tohoto problému spočívá ve zvětšení zpracovávaných oblastí tak, aby nevycházelo "bílé" místo na celou oblast nebo zmenšení obrázku, což bude mít stejný efekt jako zvětšení oblasti.

2.4.3 Aproximace pozadí pomocí funkce

Aproximace pozadí pomocí funkce je účinnější metodou pro segmentaci obrazu než předchozí metody. Pozadí, které v oblasti aproximujeme totiž nemusí mít stále stejnou hodnotu jasu.

Aproximaci pozadí pomocí funkce probíhá podle kapitoly 1.4.3. V každé oblasti tedy vypočítáme koeficienty funkce, která potom nejlépe aproximuje průběh jasové složky oblasti.

Aproximační funkce prvního řádu

Tato funkce bude mít tvar:

$$z = axy + bx + cy + d, \quad (2.3)$$

kde z je hodnota jasu bodu, x, y jsou souřadnice bodu a a, b, c, d jsou koeficienty, které chceme spočítat.

Na této rovnici si ukážeme, jak budeme ve všech případech rovnici řešit, protože řešení této rovnice je nejjednodušší (má nejméně koeficientů). Neznámými v této rovnici jsou koeficienty a, b, c, d . Rovnici řešíme metodou nejmenších čtverců [9]. Je to jednoduchá metoda, která minimalizuje kvadratickou odchylku (po aproximaci bude rozdíl skutečného bodu od aproximovaného minimální).

Řešení rovnice tedy probíhá takto:

přepíšeme si rovnici tak, abychom vyjádřili kvadratickou odchylku

$$(axy + bx + cy + d - z)^2 = \min \quad (2.4)$$

Z této rovnice potom musíme vytvořit soustavu rovnic. Potřebujeme tedy stejný počet rovnic jako neznámých, aby měla soustava právě jedno řešení (podmínkou je, že rovnice nesmí být lineárně závislé, ale to je při této metodě splněno).

Z rovnice vytvoříme soustavu rovnic tak, že budeme celou rovnici postupně derivovat podle toho, co se vyskytuje jako proměnná u neznámé:

$$\begin{aligned}\frac{d((axy + bx + cy + d - z)^2)}{da} &= 0 \\ \frac{d((axy + bx + cy + d - z)^2)}{db} &= 0 \\ \frac{d((axy + bx + cy + d - z)^2)}{dc} &= 0 \\ \frac{d((axy + bx + cy + d - z)^2)}{dd} &= 0\end{aligned}\tag{2.5}$$

$$\begin{aligned}2(ax^2y^2 + bx^2y + cxy^2 + dxy - zxy) &= 0 \\ 2(ax^2y + bx^2 + cxy + dx - zx) &= 0 \\ 2(axy^2 + bxy + cy^2 + dy - zy) &= 0 \\ 2(axy + bx + cy + d - z) &= 0\end{aligned}\tag{2.6}$$

Aby metoda fungovala, musíme udělat součet všech proměnných. Výsledná soustava rovnic tedy vypadá takto:

$$\begin{aligned}a \sum x_i^2 y_j^2 + b \sum x_i^2 y_j + c \sum x_i y_j^2 + d \sum x_i y_j &= \sum z_{ij} x_i y_j \\ a \sum x_i^2 y_j + b \sum x_i^2 + c \sum x_i y_j + d \sum x_i &= \sum z_{ij} x_i \\ a \sum x_i y_j^2 + b \sum x_i y_j + c \sum y_j^2 + d \sum y_j &= \sum z_{ij} y_j \\ a \sum x_i y_j + b \sum x_i + c \sum y_j + d n &= \sum z_{ij}\end{aligned}\tag{2.7}$$

Řešením této soustavy rovnic jsou koeficienty a, b, c, d . Tyto koeficienty následně dosadíme do rovnice 2.3. Poté jsou proměnnými x, y a výslednou hodnotu pozadí dostaneme vypočítáním neznámé z .

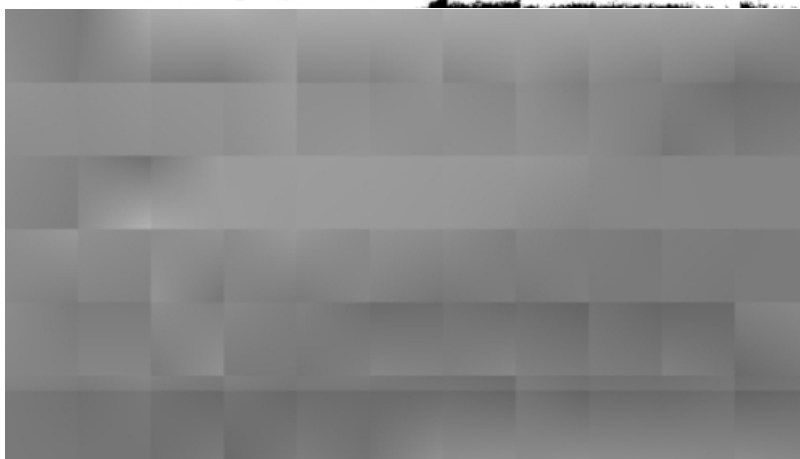
Po použití aproximační funkce prvního řádu bude výsledek vypadat takto: Jak je z

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
 Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
 Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)
 Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
 Rozměr senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, v
 Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
 Kontrolní a řídicí signály, interface.



Obrázek 2.9: Výsledný obrázek a pozadí (hodnoty aproximační funkce) pro aproximaci funkcí 1. řádu.

rovnice 2.3 vidět, tato funkce dokáže aproximovat pouze lineární závislost změny pozadí na změně pozice. Proto se nám tato rovnice pro aproximaci pozadí příliš nehodí. Pozadí totiž není vždy lineární.

Aproximační funkce druhého řádu

Tato funkce bude mít tvar:

$$z = ax^2y^2 + bx^2y + cy^2x + dx^2 + ey^2 + fxy + gx + hy + i \quad (2.8)$$

Koeficienty rovnice a, b, c, d, e, f, g, h vypočítáme stejným způsobem jako v předchozí kapitole a dosadíme do rovnice 2.8.

Po použití aproximační funkce druhého řádu bude výsledek vypadat takto:

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
 Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
 Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)

Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v

Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.

Kontrolní a řídicí signály, interface.



Obrázek 2.10: Výsledný obrázek a pozadí (hodnoty aproximační funkce) pro aproximaci funkcí 2. řádu.

Jak je vidět z rovnice 2.8, funkce dokáže aproximovat složitější pozadí než aproximační funkce prvního řádu. Zlepšila se tedy schopnost přizpůsobit se složitějším pozadím, ale zdvojnásobil se nám počet koeficientů, a tím také počet rovnic v soustavě, kterou musíme vyřešit.

Aproximační funkce třetího řádu

Tato funkce bude mít tvar:

$$z = ax_i^3 y_j^3 + bx_i^3 y_j^2 + cy_j^3 x_i^2 + dx_i^3 y_i + ey_j^3 x_i + fx_i^2 y_j^2 + \\ + gx_i^2 y_j + hy_j^2 x_i + ix_i^2 + jy_j^2 + kx_i y_j + lx_i + my_j + n \quad (2.9)$$

Postup je stejný jako v předchozích dvou případech. Z této rovnice odvodíme derivacemi soustavu rovnic, ze které vypočítáme koeficienty. Tyto koeficienty dosadíme do rovnice 2.9 a vypočítáme hodnotu aproximovaného pozadí.

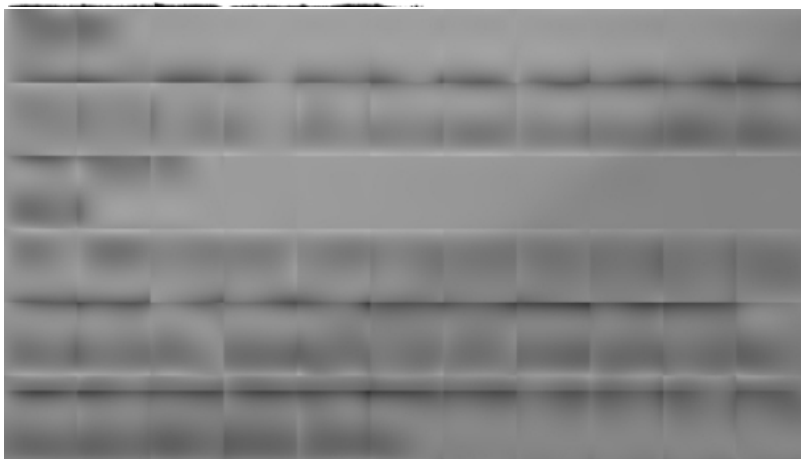
Po použití aproximační funkce třetího řádu bude výsledek vypadat takto:

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
Vaďy objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.



Obrázek 2.11: Výsledný obrázek a pozadí (hodnoty aproximační funkce) pro aproximaci funkcí 3. řádu.

Z předchozích poznatků je jasné, že čím vyšší řád funkce, tím lepší je aproximace pozadí. Toto zlepšení je opět vyváжено zvětšením počtu koeficientů. Rovnice jsou díky nim složitější a je jich více, takže se zvyšuje čas výpočtu.

Zpřesňování aproximace opakováním průchodů

Protože po prvním spočítání aproximační funkce nemusí být výsledek ideální a pozadí nemusí být aproximováno správně (například písmeno nemusí být celé atd.), používám pro aproximaci pozadí více průchodů.

V každém průchodu – kromě prvního – se váhuje aktuální hodnota aproximovaného pozadí podle rozdílu od skutečné hodnoty v obrázku. Váhování probíhá tak, že čím blíže leží hodnota aproximace skutečnému bodu, tím větší má váhu. Body ležící pod křivkou, kterou jsme získali aproximací (mají nižší hodnotu jasu – jsou tedy tmavší, a proto se může potencionálně jednat o černé písmeno), mají poloviční váhu. Zavedení poloviční váhy bylo nutné z důvodu ovlivňování tvaru funkce, která potom nekopíruje tvar pozadí, ale je "snižována" tmavými body, které tvoří písmena.

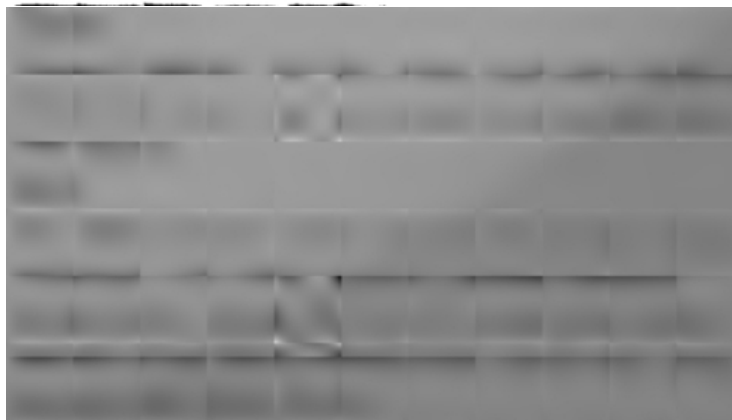
Obrázky 2.12 a 2.13 ukazují, jak ovlivní pozadí pokud oblast projdeme vícekrát.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.



Obrázek 2.12: Dokument, na kterém byla použita segmentace aproximační funkcí 3. řádu s dvěma průchody a hodnoty aproximační funkce.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os

Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s

Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)

Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v

Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.

Kontrolní a řídicí signály, interface.



Obrázek 2.13: Dokument, na kterém byla použita segmentace aproximační funkcí 3. řádu s třemi průchody a hodnoty aproximační funkce.

Vývojový diagram je na obrázku B.6.

Překrývání oblastí

Jelikož na krajích oblastí, které aproximujeme vzniká největší odchylka aproximační funkce od skutečných hodnot jasu v obraze, je výhodné využívat jen střední části oblastí, pro které jsme funkci počítali.

Vypočítám tedy funkci pro celou oblast, ale při dosazování hodnot do funkce nedosazuji všechny hodnoty. Na každém okraji oblasti, který neleží na kraji obrázku, vynechávám prvních 10 procent bodů (pro oblast 200 x 200 bodů tedy dosazuji za body $\langle 20 - 180 \rangle \times \langle 20 - 180 \rangle$). Pokud oblast obsahuje krajní body obrázku, potom tyto krajní body dosazuji (nevynechávám zmíněných 10 procent krajních bodů).

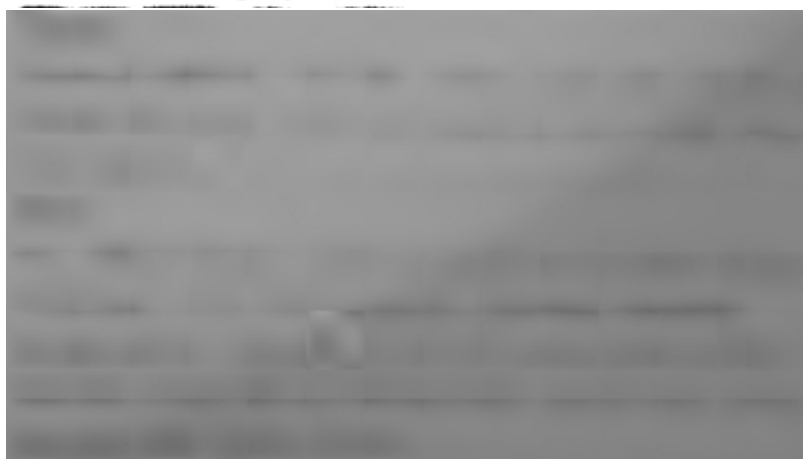
Na obrázku 2.14 je výsledek po překrývání oblastí.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.



Obrázek 2.14: Dokument, na kterém byla použita segmentace aproximační funkcí 3. řádu s třemi průchody a překrýváním oblastí a hodnoty aproximační funkce.

2.4.4 Otsuho metoda

Otsuho metoda [12] je metoda založena na statistických datech z obrazu. Metoda vychází ze vzorce:

$$\sigma_b^2 = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2, \quad (2.10)$$

kde ω_i je prvek součtového histogramu a μ_i je prvek součtového histogramu vynásobený polohou hodnoty v histogramu.

Postup při prahování Otsuho metodou:

- vytvoření histogramu;
- vytvoření součtového histogramu;
- vytvoření váhovaného součtového histogramu;
- nalezení maxima váhového součtového histogramu;
- vypočítání σ_b^2 ;
- určení nejvyšší hodnoty σ_b^2 a jasů, kterému odpovídá;
- jas, který odpovídá nejvyšší hodnotě σ_b^2 je prahem pro následné prahování.

Metoda vykazuje lepší výsledky než metody procentního prahování a poměrného procentního prahování. Přitom není o moc složitější.

Výsledek segmentace Otsuho metodou je na obrázku 2.15. Vývojový diagram je na obrázku B.7.

Tečky v oblasti, kde by mělo být pouze pozadí jsou opět způsobeny tím, že hodnoty jasů v obrazu jsou velice blízko u sebe, a proto je práh nastaven velice blízko k hodnotám pozadí. Tento problém by se dal vyřešit (jako v kapitole 2.4.2) tím, že bychom zvětšili zpracovávané oblasti tak, že by každá obsahovala alespoň část znaku, nebo zmenšením zpracovaného obrazu, čímž bychom docílili stejného výsledku jako zvětšením oblastí.

Další možností je sledovat velikost rozptylu jasů bodů a pokud by byl tento rozptyl malý, nepočítali bychom práh a okamžitě označili oblast za pozadí.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).
Rozměr senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, v
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.

Obrázek 2.15: Dokument, na kterém byla použita segmentace Otsuho metodou.

2.4.5 Metoda založená na narůstání oblastí

Metoda, kterou v této kapitole popíšu je výpočetně méně náročná než aproximace pozadí funkcí, ale mnohem složitější než prahování.

Metoda je založená na narůstání oblastí [6] v kombinaci s vyhledáváním hran.

Postup metody

Prvním krokem metody je rozdělení obrázku na oblasti. V každé oblasti najdu minimální a maximální hodnotu. Díky tomu získám maximální rozptyl hodnot. Tento rozptyl vydělím koeficientem. Díky tomu získám práh pro zpracovávanou oblast obrázku:

$$prah = \frac{max - min}{koeficient}. \quad (2.11)$$

Jako koeficient můžeme dosadit jakékoli reálné číslo větší než nula. Čím větší číslo dosadíme, tím více dostaneme hran. Na druhou stranu při dosazení malého čísla nedostaneme některé důležité hrany.

Hrany hledáme ve směru osy x i ve směru osy y. Tyto výsledky následně sloučíme a dostáváme obrysy písmen. Tyto obrysy jsou ale zatíženy šumem.

Po nalezení hran aplikujeme na obrázek dilataci a erozi – aplikujeme tedy **uzavření** 1.5.

Máme obrázek s nalezenými hranami po funkci otevření. V tuto chvíli musíme obrázek načíst celý – tedy ne po oblastech jako v prvním kroku. V obrázku začneme hledat po řádcích oblasti¹ patřící k sobě (nejsou odděleny černou barvou) a označíme je čísly – černá barva (hrany) mají číslo nula a poté každá nová oblast dostává své vlastní číslo. Získáme tedy mnoho oblastí. Oblasti, které se dotýkají – nejsou odděleny – spojíme, přidělíme jim tedy stejné číslo.

K nalezeným oblastem vytvoříme histogram – určíme počet pixelů, které daná oblast obsahuje. Hodnoty histogramu poté seřadíme sestupně (oblast nula vynecháváme, protože počet černých bodů nás nezajímá). Oblast, která má největší počet pixelů označíme jako pozadí.

Nyní si opět obraz rozdělíme na části a budeme je zpracovávat zvlášť. Oblast, kterou jsme označili jako pozadí ihned vykreslíme do výsledného obrázku bílou barvou – hodnotou jasu #255. Poté vypočítáme průměrnou hodnotu jasu této oblasti v původním obrázku a opět vypočítáme rozptyl hodnot z minima a maxima zpracovávané části. Koeficient, kterým budeme rozptyl dělit bude ale jiný (měl by být větší, protože pokud bychom zvolili menší koeficient, potom by byly i vnitřní části písmen označeny bíle)

$$hystereze = \frac{max - min}{koeficient}. \quad (2.12)$$

Dále musíme vypočítat průměrnou hodnotu oblasti o jejíž barvě rozhodujeme. Aby oblast byla ve výsledku bílá, musí platit:

$$prumer_{oblasti} - prumer_{pozadi} \in < -hystereze; hystereze > . \quad (2.13)$$

Pokud tento vztah neplatí, je oblast označena černě – jas #0.

Předposledním krokem je kontrola oblastí s číslem nula. Pokud byla oblast označena číslem nula, zkontrolujeme, zda je hodnota pixelu v původním obrázku ve stejných mezích jako v rovnici 2.13. Pokud ano, přepíšeme bod na bílou barvu.

Tímto krokem se odstraní většina slitků mezi písmeny či mezi písmenem a jeho diakritikou.

Nakonec ještě použijeme kombinaci eroze-dilatace-eroze, čímž zúžíme a částečně vyhladíme písmena.

Výsledek metody je na obrázku 2.16. Vývojový diagram je na obrázku B.8.

¹Nejedná se o stejné oblasti, na které dělíme obrázek při hledání hran.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)

Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v

Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.

Kontrolní a řídicí signály, interface.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka os

Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, s

Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy)

Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Rozměr senzoru – nejčastěji 1/3", 1/2", 2/3", rozměr pixelu, rozlišení, v

Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.

Kontrolní a řídicí signály, interface.

Obrázek 2.16: Dokument, na kterém byla použita metoda založená na narůstání oblastí a barevně označené nalezené oblasti.

2.4.6 Shrnutí kapitoly

V této kapitole jsem vysvětlil postup při programování jednotlivých metod segmentace obrazu.

Jako nejuniverzálnější metoda se jeví aproximace pozadí pomocí funkce, která je tak výpočetně náročná, že v praxi je pravděpodobně nepoužitelná². Její výhodou je, že dokáže aproximovat i velmi rychlé přechody (obrázek A.7) s dobrými výsledky.

Za v praxi nejpoužitelnější považuji prahování Otsuho metodou. Je univerzální, rychlá a velice účinná. Bohužel nedokáže aproximovat ostré přechody, protože nastavuje pevný práh pro celou oblast, takže i při rozdělení oblasti na části nám v obrázku vzniknou poměrně velké chyby v oblasti přechodu (obrázek A.8).

²Například obrázek o velikosti 2200 x 1100 pixelů při velikosti oblastí 200 x 200 pixelů, třemi průchody a překrýváním oblastí jsem na 2 GHz procesoru zpracovával 15 min.

Dalšími metodami použitelnými v praxi jsou procentní prahování a poměrné procentní prahování. Při správném nastavení, které může být pro každý obrázek jiné, mohou vykazovat stejné vlastnosti jako Otsuho metoda. Jsou ale rychlejší. Jejich nevýhodou je, že pro správnou funkci bychom museli měnit koeficient pro výpočet prahové hodnoty v každém obrázku.

Poslední zmíněnou metodou je metoda založená na rozrůstání oblastí (a semínkovém vyplňování). Metoda vykazuje poměrně dobré výsledky srovnatelné s Otsuho metodou a dalšími metodami prahování. Je ale výpočetně mnohem náročnější. V porovnání s aproximací pomocí funkce je metoda přibližně třikrát rychlejší. Obrázky, na kterých je ostrý přechod, nedokáže zpracovat správně (stejně jako jednodušší metody) a výsledek obsahuje velké chyby (obrázek A.9).

3 ZÁVĚR

V této práci jsem nastínil funkci některých metod pro předzpracování a segmentaci obrazu, které jsou dostupné v odborných literaturách. Některé z nich se používají běžně, jiné – pro svou náročnost – jen velmi zřídka.

V první části jsem se zabýval teoretickými východisky pro předzpracování a segmentaci obrazu – Gaussůvým filtrem, filtrem průměrováním, mediánovým filtrem, filtrem rotující maskou, Cannyho hranovým detektorem a Sobelovým operátorem, který slouží k detekci hran.

Pro segmentaci obrazu jsem zmínil její základní metody – procentní prahování (podle nejmenší i největší hodnoty), poměrné procentní prahování, aproximace funkcí a segmentace rozvodím.

Posledním teoretickým východiskem pro zpracování obrazu, které jsem zde probral jsou morfologické transformace – dilatace, eroze, otevření, uzavření. Tyto transformace usnadňují identifikaci obrázků, zjednodušují jeho tvar nebo jsou vhodné pro odstranění malých chyb v obrazu.

Také jsem zmínil knihovny OpenCV – knihovny pro programování, které umožňují jednoduchou práci s obrazy.

Druhou část jsem věnoval praktickému řešení některých zmíněných metod. Nejprve jsem uvedl některé typy obrázků, které mohou být vstupem.

Dále byla vysvětlena volba velikosti oblasti – oblast by měla být veliká asi jako 2 – 4 řádky textu.

Poté jsem znovu probral filtry zmíněné v první části s tím rozdílem, že nyní jsem se zabýval problematikou programování těchto filtrů. Jako nejlepší filtr se mi jeví filtr rotující maskou, který nerozmazává hrany a který velice dobře eliminuje šum. Dalším dobře použitelným filtrem je podle mého názoru Gaussův filtr, který při správném nastavení parametrů rozmazává hrany jen minimálně. Naopak mediánový filtr není příliš účinný a filtr průměrováním hodně rozmazává hrany.

Poslední částí jsou metody segmentace obrazu. Probral jsem metody procentního prahování, poměrného procentního prahování, aproximace pozadí pomocí funkce, Otsuho metodu prahování a metodu založenou na narůstání oblastí. Největší univerzálnost vykazuje metoda aproximace pozadí pomocí funkce, která je výpočetně velmi náročná, takže je v praxi nepoužitelná. Okraje oblastí, ve kterých pozadí aproximují jsou také zatíženy chybou a aproximace je nejpřesnější pouze uprostřed oblasti.

Otsuho metoda se jeví jako nejlepší použitelná metoda v praxi. Není tak univerzální jako aproximace funkcí, ale je rychlá a poměrně přesná. V oblastech, kde je pouze pozadí, metoda nastavuje práh tak, že jsou tyto oblasti znehodnoceny šumem.

Podobné výsledky jako Otsuho metoda vykazují při správném nastavení i procentní prahování a poměrné procentní prahování. Pro každý obrázek bychom ale museli měnit koeficient, pomocí kterého počítáme práh.

Poslední je metoda založená na narůstání oblastí, která je výpočetně mnohem náročnější než Otsuho metoda, i obě metody prahování, při prakticky stejných výsledcích. Není ale tak náročná jako aproximace pozadí pomocí funkce.

Pro běžné využití segmentace obrazu dokumentu bych tedy využil Otsuho metodu. Před jejím použitím bych navíc použil filtr rotující maskou pro eliminaci šumu v obrazu. Tato metoda by pro běžné fotografie nebo obrázky ze skeneru měla vykazovat velice dobré výsledky a výstup těchto metod by se dal využít jako vstup pro případný OCR program.

REFERENCE

- [1] *OCR* [online]. 2010 [citováno 2010-4-6].
Dostupný z WWW: <<http://cs.wikipedia.org/wiki/OCR>>.
- [2] *Segmentace obrazu* [online]. 2010 [citováno 2010-4-1].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/Segmentace_obrazu>.
- [3] HLAVÁČ, V., ŠONKA, M. *Počítačové vidění*. GRADA, 1992. 272 s.
ISBN 80-85424-67-3.
- [4] *Odstranění šumu* [online]. 2010 [citováno 2010-4-2].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/Odstranění_šumu>.
- [5] ŽÁRA, J., BENEŠ, B., SOCHOR, J., FELKEL, P. *Moderní počítačová grafika*.
Computer press, 2004. 628 s.
ISBN 80-251-0454-0.
- [6] HORÁK, K., KALOVÁ, I., PETYOVSKÝ, P., RICHTER, M. *Počítačové vidění*.
FEKT VUT Brno, 2007. 132 s.
- [7] *Playboy 20*.
Chicago, Ill.: HMH Pub. Co., 1972, č. 11, Obrázek Leny Söderberg,
ISSN 0032-1478.
- [8] *Cannyho hranový detektor* [online]. 2010 [citováno 2010-4-3].
Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Canny>>.
- [9] FAJMON, B., RŮŽIČKOVÁ, I. *Matematika 3*. FEKT VUT Brno, 2003. 257 s.
Identifikační číslo v IS FEKT MAT103.
- [10] *BSD licence* [online]. 2010 [citováno 2010-4-12].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/BSD_licence>.
- [11] BRADSKI, G., KAEHLER, A. *Learning OpenCV*
Computer Vision with the OpenCV Library. O'Reilly Media, 2008. 576 s.
ISBN 978-0-596-51613-0.

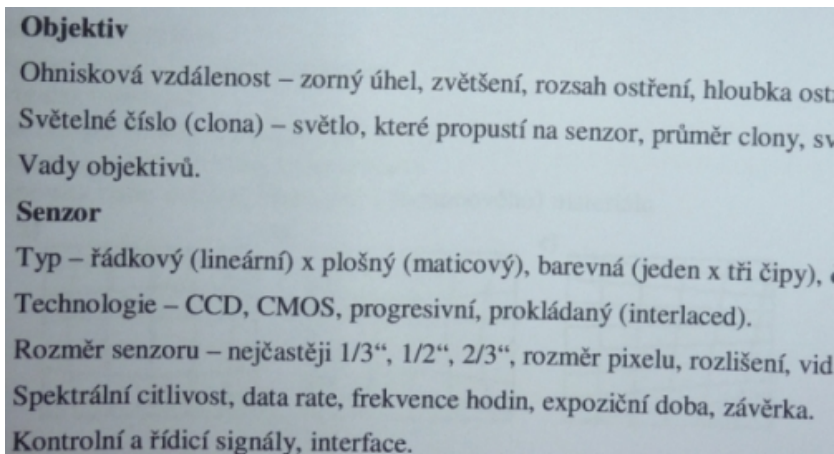
[12] *Otsu's method* [online]. 2010 [citováno 2010-4-30].

Dostupný z WWW: <http://en.wikipedia.org/wiki/Otsu's_method>.

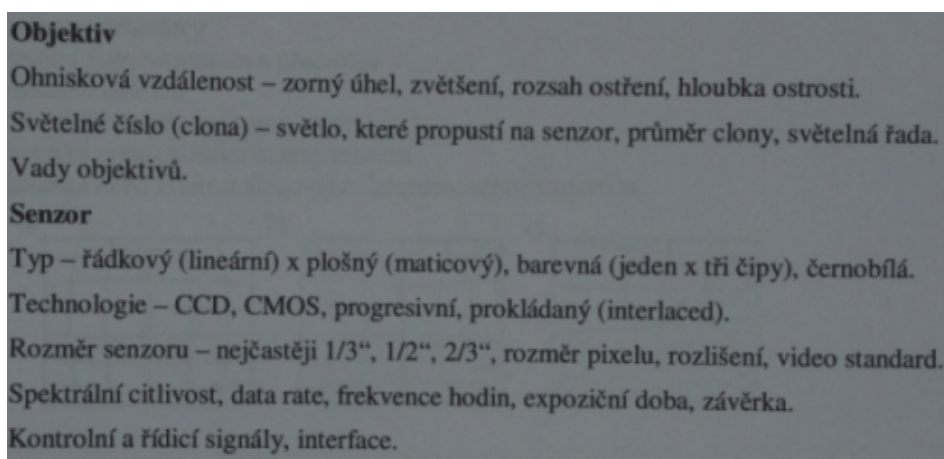
SEZNAM PŘÍLOH

A Ukázky obrázků	61
B Vývojové diagramy	65
C Elektronická příloha	73

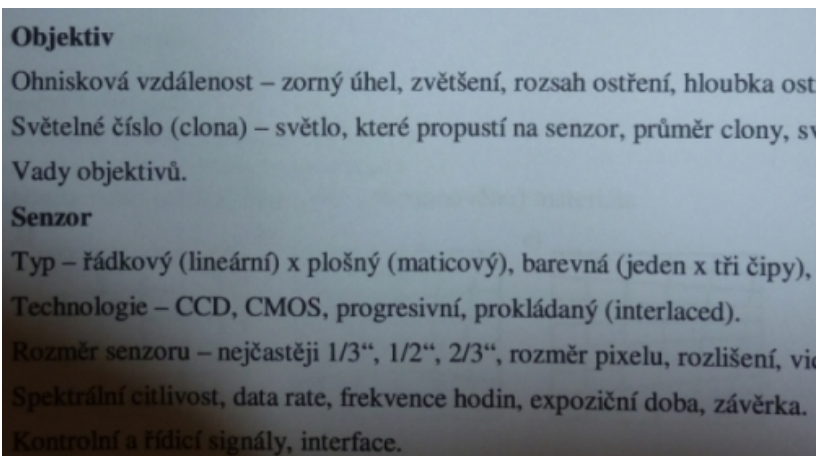
A UKÁZKY OBRÁZKŮ



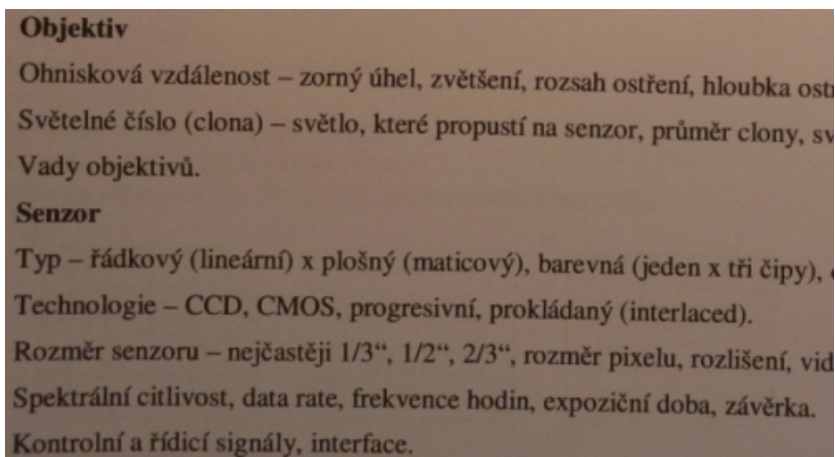
Obrázek A.1: Obrázek, který má světlejší střed a tmavší rohy.



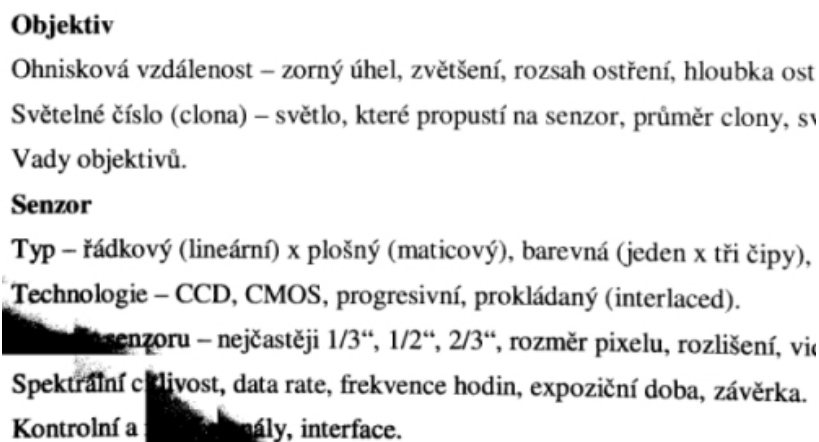
Obrázek A.2: Obrázek, který má světlejší rohy a tmavší střed.



Obrázek A.3: Obrázek, na kterém je ostrý stín.



Obrázek A.4: Obrázek, který má vysokou hladinu šumu.



Obrázek A.5: Obrázek s ostrým stínem, na kterém bylo použito procentní prahování.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka ost
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, sv
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy),
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Rozměr senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, vi
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.

Obrázek A.6: Obrázek s ostrým stínem, na kterém bylo použito poměrné procentní prahování.

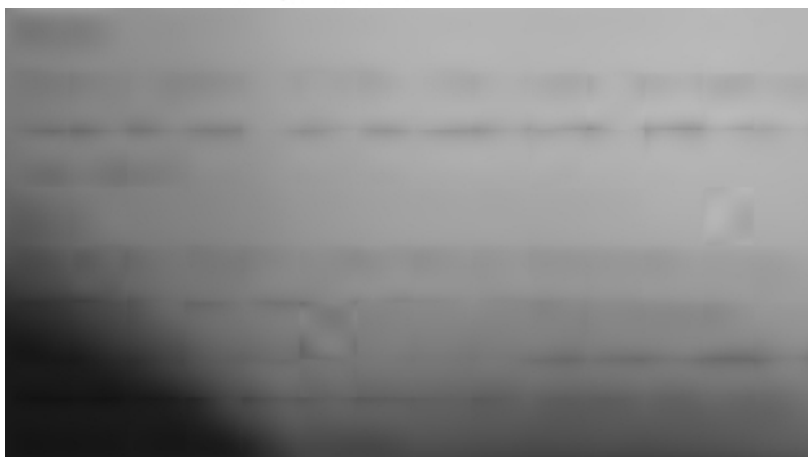
Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka ost
Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, sv
Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy),
Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Rozměr senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, vi
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.
Kontrolní a řídicí signály, interface.



Obrázek A.7: Obrázek s ostrým stínem, na kterém byla použita aproximace funkcí 3. řádu se třemi průchody a překrýváním oblastí a pozadí.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka ost

Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, sv

Vady objektivů.

Senzor

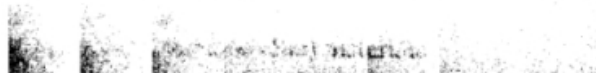
Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy),

Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Velikost senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, vi

Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.

Kontrolní a řídicí signály, interface.



Obrázek A.8: Obrázek s ostrým stínem, na kterém byla použita Otsuho metoda.

Objektiv

Ohnisková vzdálenost – zorný úhel, zvětšení, rozsah ostření, hloubka ost

Světelné číslo (clona) – světlo, které propustí na senzor, průměr clony, sv

Vady objektivů.

Senzor

Typ – řádkový (lineární) x plošný (maticový), barevná (jeden x tři čipy),

Technologie – CCD, CMOS, progresivní, prokládaný (interlaced).

Velikost senzoru – nejčastěji 1/3“, 1/2“, 2/3“, rozměr pixelu, rozlišení, vi

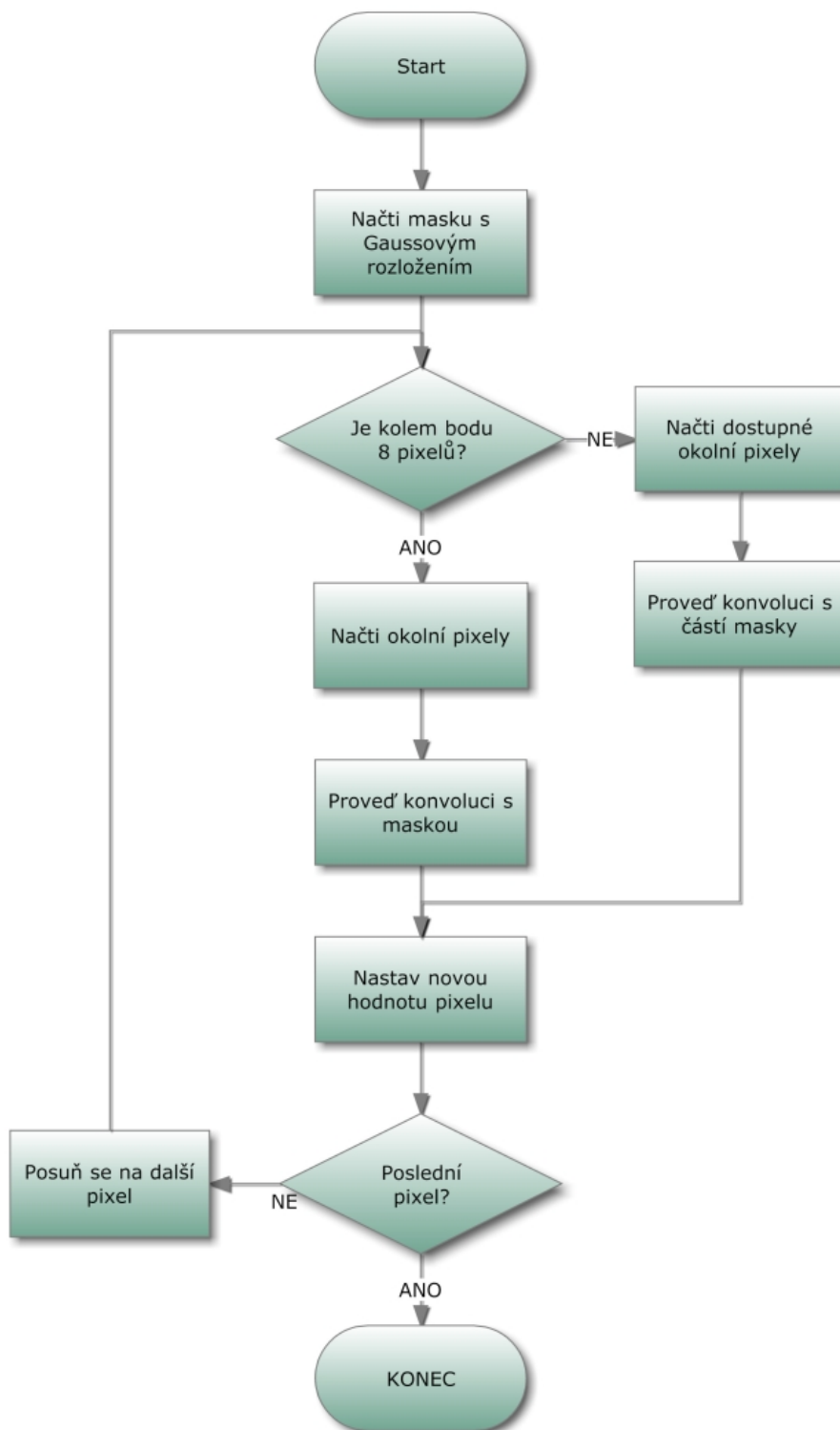
Spektrální citlivost, data rate, frekvence hodin, expoziční doba, závěrka.

Kontrolní a řídicí signály, interface.

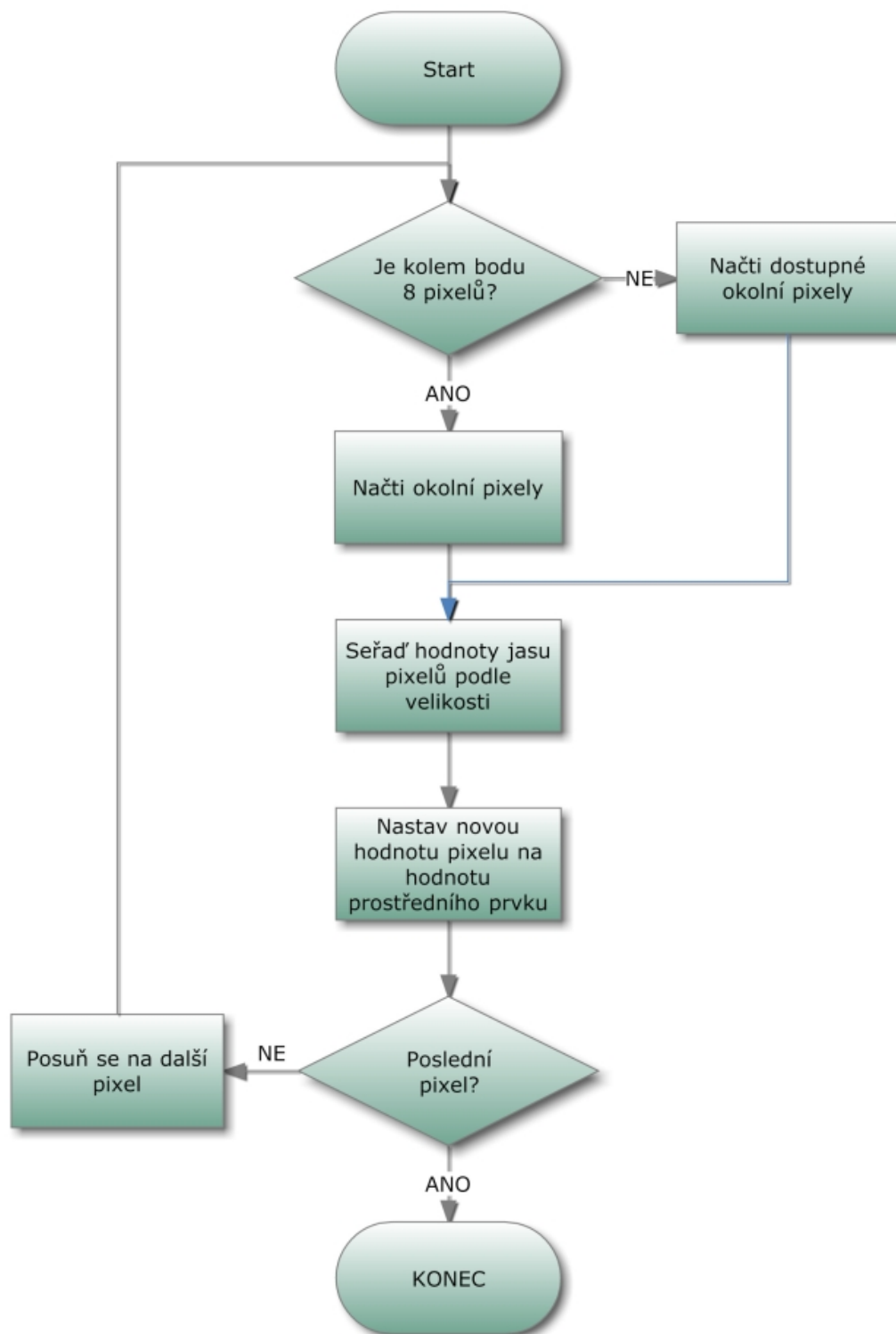


Obrázek A.9: Obrázek s ostrým stínem, na kterém byla použita metoda založená na narůstání oblastí.

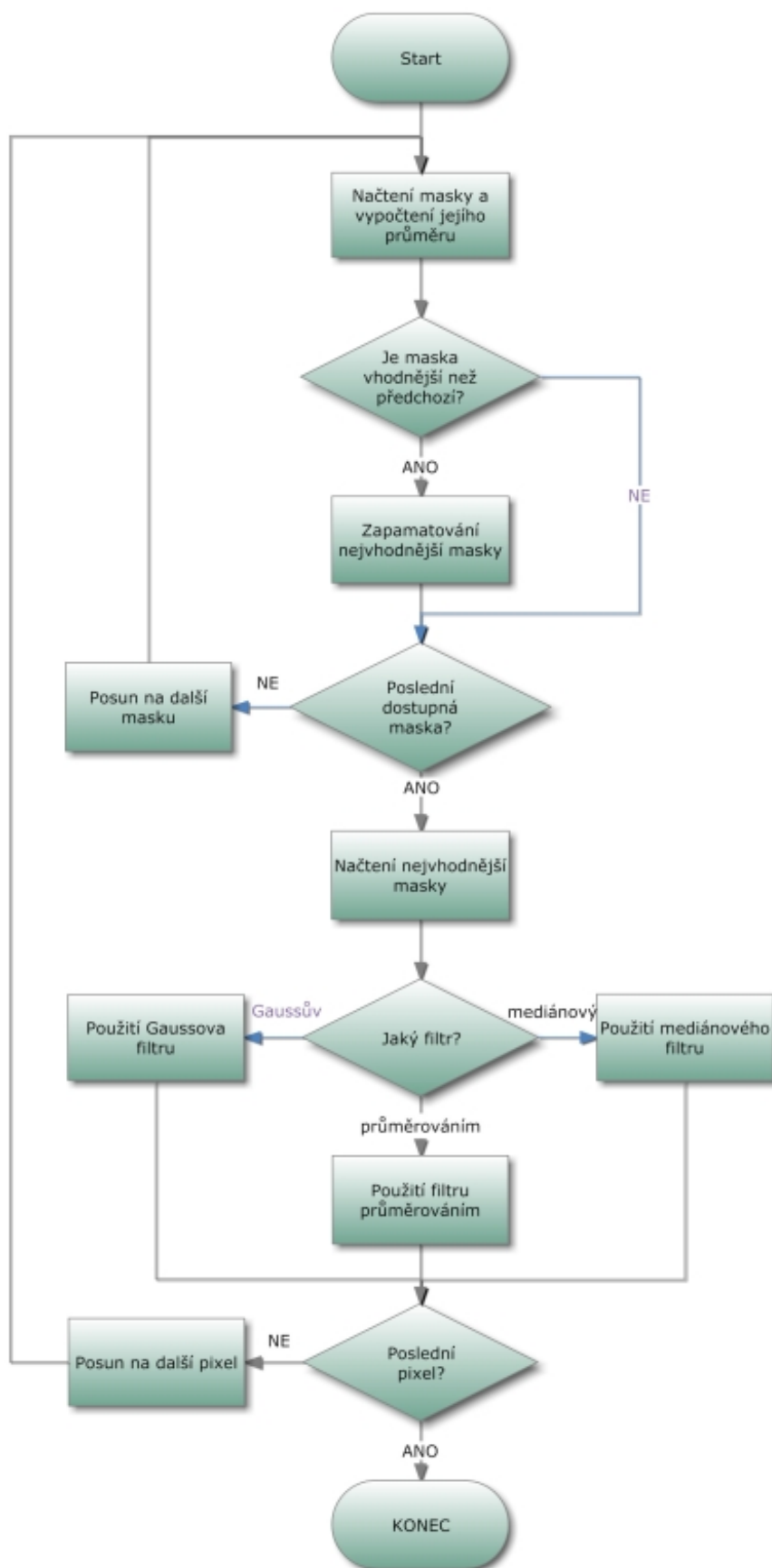
B VÝVOJOVÉ DIAGRAMY



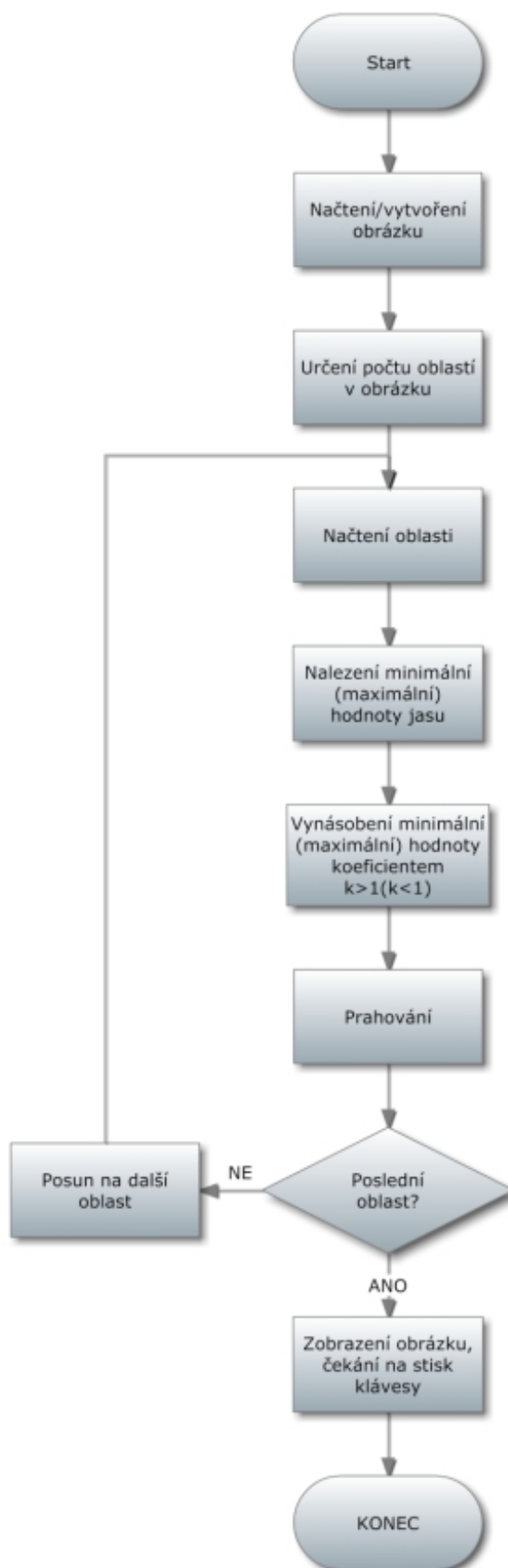
Obrázek B.1: Vývojový diagram Gaussova filtru.



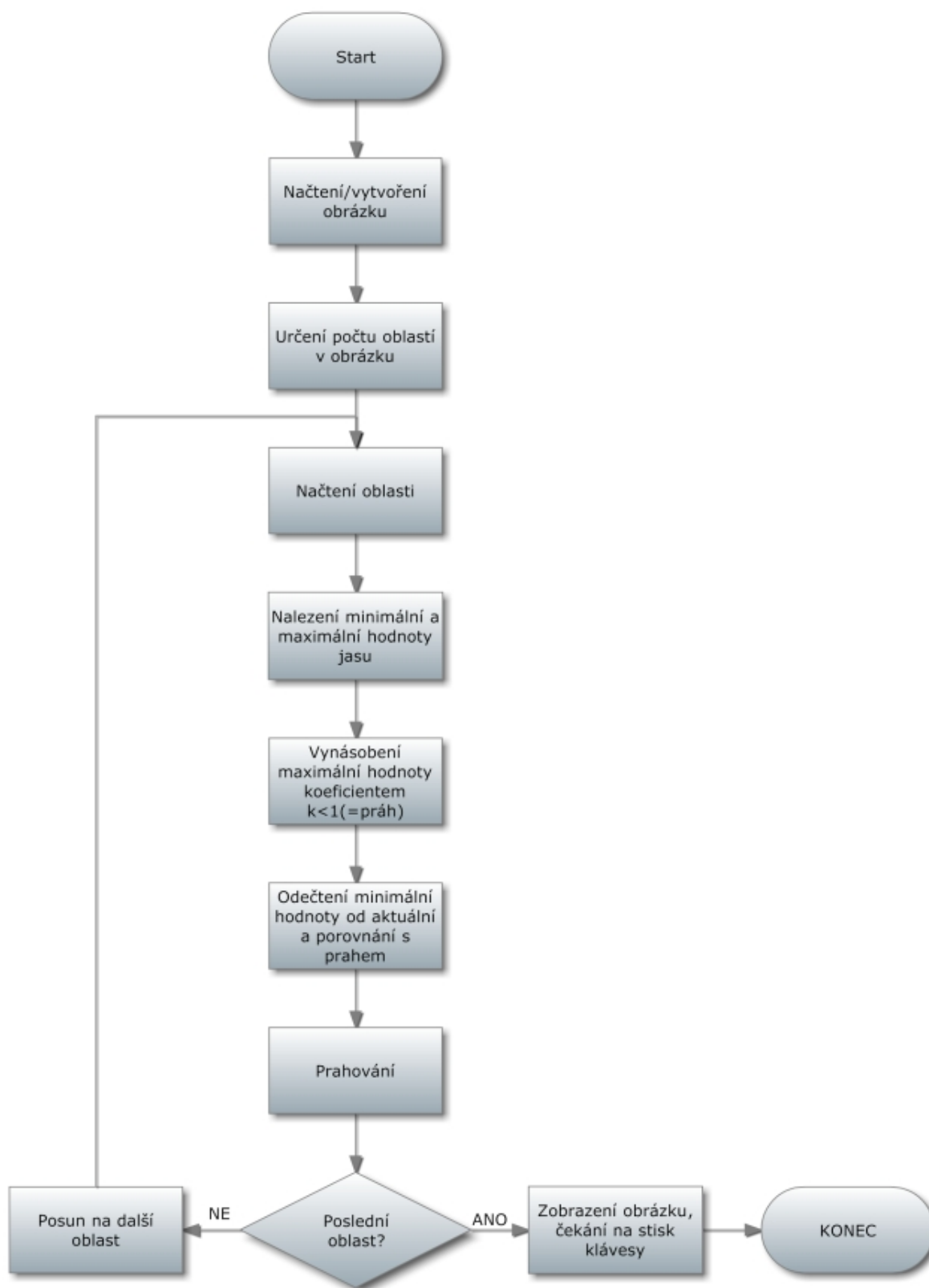
Obrázek B.2: Vývojový diagram mediánového filtru.



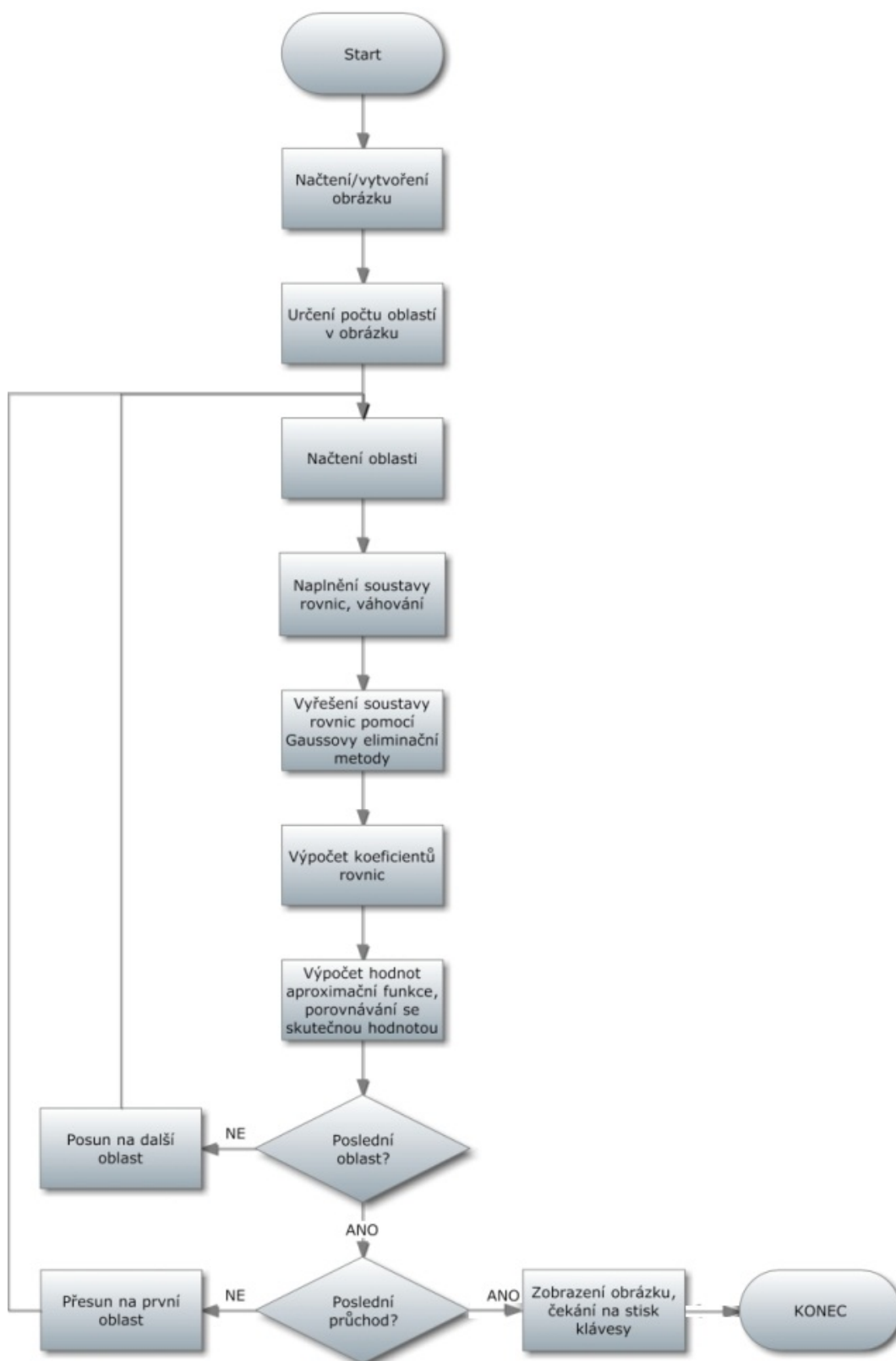
Obrázek B.3: Vývojový diagram filtru rotující maskou.



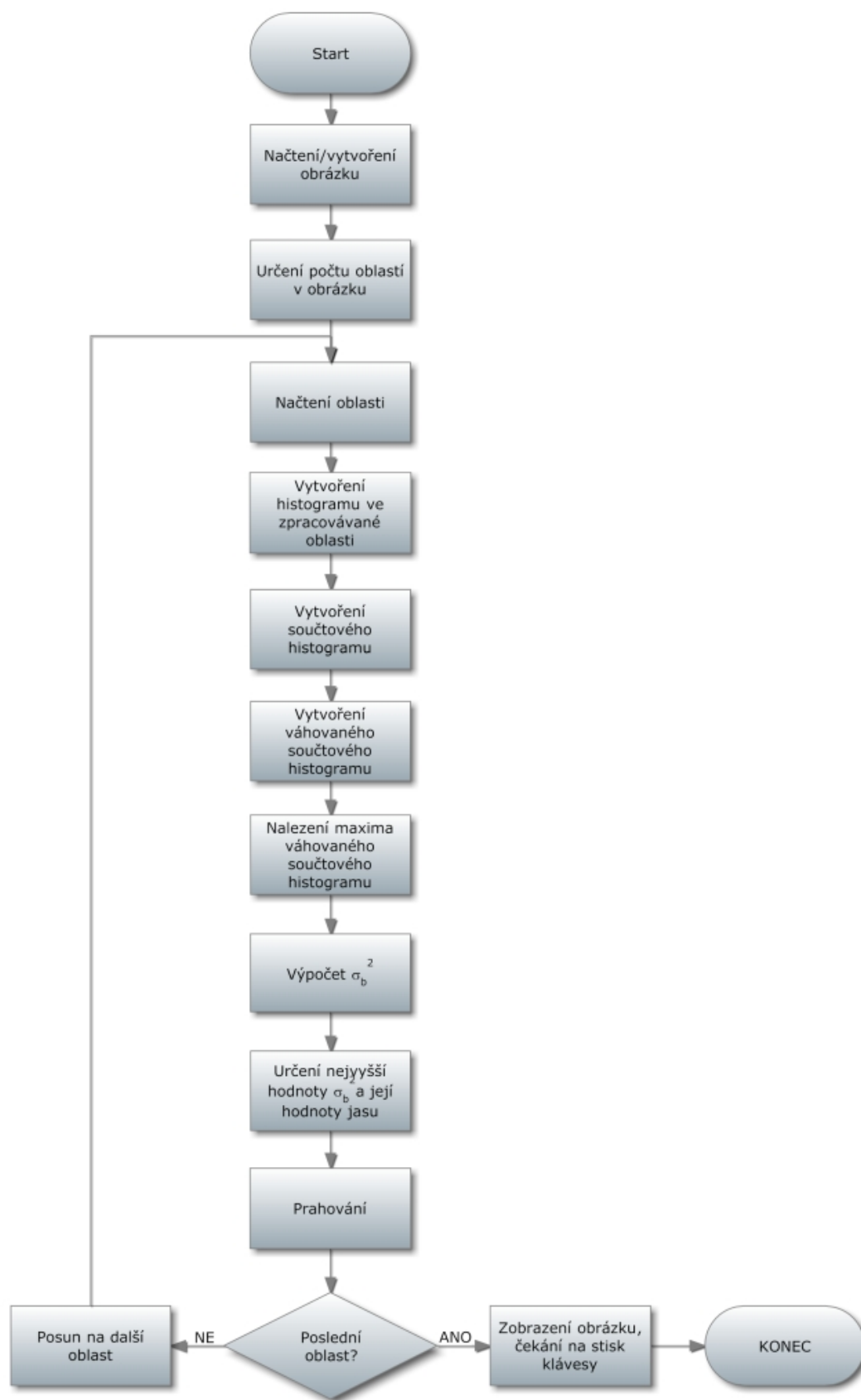
Obrázek B.4: Vývojový diagram segmentace procentním prahováním.



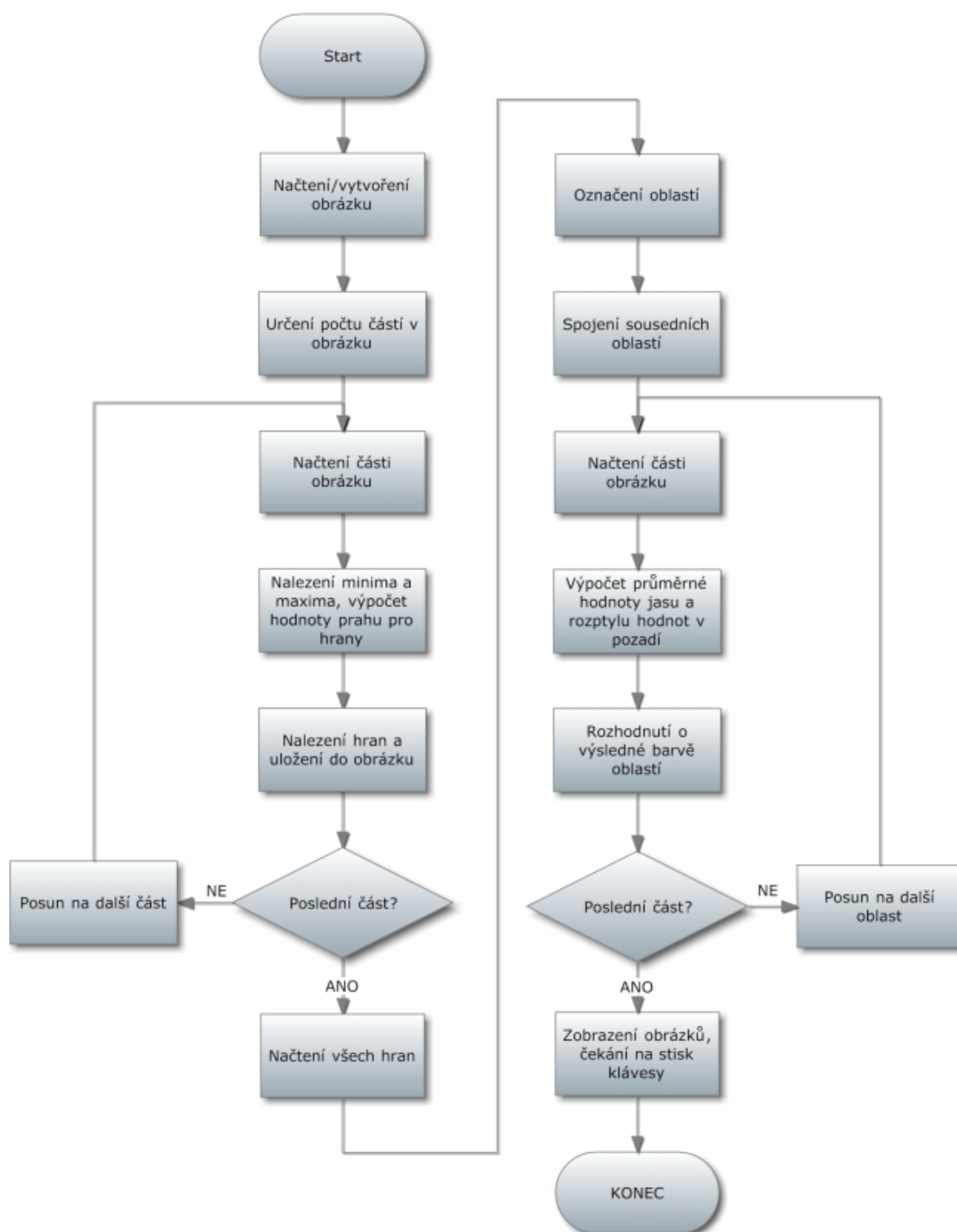
Obrázek B.5: Vývojový diagram segmentace poměrným procentním prahováním.



Obrázek B.6: Vývojový diagram segmentace aproximační funkcí.



Obrázek B.7: Vývojový diagram segmentace Otsuho metodou.



Obrázek B.8: Vývojový diagram metody založené na narůstání oblastí.

C ELEKTRONICKÁ PŘÍLOHA

Elektronická příloha (CD) obsahuje:

- zdrojové kódy všech metod segmentace a předzpracování obrazu;
- instalaci knihoven OpenCV pro Windows;
- vstupy i výstupy programu v plné velikosti;
- tento dokument ve formátu PDF.