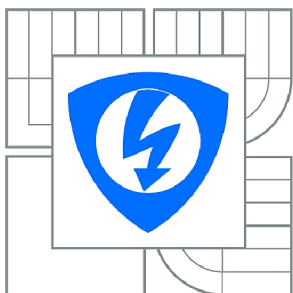


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SEGMENTACE OBRAZU ZALOŽENÁ NA RELIÉFU

IMAGE SEGMENTATION BASED ON RELIEF

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN GROS

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ALEŠ KŘUPKA

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Gros

ID: 125434

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Segmentace obrazu založená na reliéfu

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s technikou segmentace obrazu, u které je vymezení objektů v obraze založeno na principu zaplavování reliéfu krajiny (watershed metoda). Tuto metodu implementujte ve formě operátoru nástroje RapidMiner. Zabývejte se i modifikací metody zvyšující její použitelnost. Funkčnost implementované metody ověřte na testovacích obrázcích a zhodnotte dosažené výsledky.

DOPORUČENÁ LITERATURA:

[1] JAN, J. Medical Image Processing, Reconstruction and Restoration. Boca Raton: Taylor, 2006.

[2] VINCENT, L., SOILLE, P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. 1991, 583-598.

[3] VINCENT, L. Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. IEEE TRANSACTIONS ON IMAGE PROCESSING. 1993, 176-201.

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Aleš Křupka

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, 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

Cílem této práce je seznámit se s technikou segmentace obrazu pomocí zaplavování reliéfu krajiny (watershed metoda). Jsou zde přiblíženy některé metody segmentace obrazu a vysvětleny základní operace nad obrazy, z kterých vychází morfologické operace pro použití ve watershed segmentaci. Součástí práce je krátká pasáž, která se zabývá použitím některých obrazových filtrů k předzpracování obrazu. Výsledkem předchozích poznatků je prezentování segmentačního algoritmu vytvořeného Lee Vincentem a Pierre Soillem. V další části je popsána implementace algoritmu v jazyce Java, obsah a funkce jednotlivých tříd modulu a realizace pomocí nástroje RapidMiner. V poslední části jsou prezentovány výsledky segmentování vytvořeného modulu při použití na testovací obrazy bez a s použitím některých vybraných filtrů pro redukci přesegmentování a ke zlepšení výsledné segmentace.

KLÍČOVÁ SLOVA

segmentace obrazu, watershed segmentace, algoritmus zaplavování, nástroj RapidMiner, morfologické operace

ABSTRACT

The aim of this labor is to introduce the technique of image segmentation by flooding relief of landscape (watershed method). There are described some methods of image segmentation and explained the basic operations of the images, from which the morphological operations are used for watershed segmentation. Part of this labor is a short passage that deals with the use of some image filters for image preprocessing. The result of previous knowledge is presentation of segmentation algorithm designed by Lee Vincent and Pierre Soille. The next section describes the algorithm implementation in Java, contents and function of each class module and implementation using RapidMiner. The last section presents the results of the segmentation with created module using the test images without and with the use of some selected filters to reduce oversegmentation and improve the resulting segmentation.

KEYWORDS

image segmentation, watershed segmentation, flooding algorithm, RapidMiner tool, morphological operations

GROS, J. *Segmentace obrazu založená na reliéfu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 44 s. Vedoucí práce Ing. Aleš Křupka.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Segmentace obrazu založená na reliéfu“ 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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu práce panu Ing. Aleši Křupkovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

Brno

.....

(podpis autora)

OBSAH

Úvod	9
1 Techniky segmentace obrazu	10
1.1 Prahování	10
1.2 Segmentace pomocí detekce hran	10
1.3 Segmentace na principu hledání oblastí	11
1.3.1 Split and Merge	11
1.3.2 Watershed	12
1.3.3 Shluková analýza	12
2 Morfologické operace	13
2.1 Základy morfologie	13
2.1.1 Strukturní element	13
2.1.2 Dilatace	13
2.1.3 Eroze	14
2.1.4 Otevření	15
2.1.5 Uzavření	16
3 Watershed segmentace	17
3.1 Úvod do morfologické segmentace	17
3.1.1 Vztahy mezi pixely	18
3.2 Maximální kruh, kostra a konečná eroze	18
3.2.1 Kruhy	18
3.2.2 Kostra	19
3.2.3 Funkce quench	21
3.2.4 Konečná eroze a funkce vzdálenost	22
3.3 Geodetické transformace	24
3.3.1 Geodetické vzdálenosti	24
3.3.2 Geodetické dilatace a eroze	25
3.3.3 Geodetická zóna vlivu	26
3.4 Definice a výpočet watershedů	26
3.4.1 Definice z hlediska nejstrmějšího spádu	27
3.4.2 Definice pomocí ponoření	27
3.4.3 Metody proti přesegmentování	27
4 Implementace	29
4.1 RapidMiner	29
4.2 Proces segmentace	30

4.2.1	Moduly	30
4.2.2	Algoritmus	30
4.2.3	Třídy	31
4.3	Výsledky	32
5	Závěr	36
	Literatura	38
	Seznam symbolů, veličin a zkratk	39
	Seznam příloh	40
A	Watershed algoritmus	41
B	Obsah CD a manuál k programu	44

SEZNAM OBRÁZKŮ

1.1	Ukázka profilů jednotlivých hran: a) <i>line edge</i> ; b) <i>ramp edge</i> ; c) <i>step edge</i> ; d) <i>roof edge</i> ; e) <i>variance edge</i> ; f) <i>noisy edge</i>	11
1.2	Dělení na čtvrtinové oblasti. [4]	12
2.1	Ukázka strukturních elementů.	13
2.2	Binární dilatace.	14
2.3	Binární eroze.	15
2.4	Opening. [3]	16
2.5	Closing. [3]	16
3.1	Binární obrázek kávových zrn. [1]	18
3.2	Jednotkový kruh B v různé konektivitě. [1]	19
3.3	Koncept maximálního kruhu v eukleidovské rovině. [1]	19
3.4	Kostry několika jednoduchých tvarů. [1]	20
3.5	Ukázka koster dvou tečných kruhů ve spojitě rovině. [1]	20
3.6	Kávová zrna a jejich kostry pomocí maximálních kruhů. [1]	21
3.7	a) kompletní spojená kostra; b) ztenčená kostra. [1]	22
3.8	Kostra dvou překrytých objektů a jejich maxima. [1]	22
3.9	Proces iterativní eroze. [1]	23
3.10	Objekty kávových zrn s vlastními značkami. [1]	24
3.11	Geodetická vzdálenost.	25
3.12	Geodetická dilatace a eroze. [1]	25
3.13	Geodetická zóna vlivu spojené složky B_1 uvnitř množiny A . [2]	26
3.14	Minima, rozvodí a watershedy.	27
3.15	Budování hrází tam, kde se voda z různých rozvodí začne slévat.	28
4.1	Ukázka pracovní plochy programu RapidMiner.	29
4.2	Moduly pro segmentaci a jejich propojení.	31
4.3	a) originál; b) segmentace bez filtrů; c) segmentace s medián filtrem; d) segmentace s detektorem hran	33
4.4	a) originál; b) segmentace bez filtrů; c) segmentace s medián filtrem; d) segmentace s detektorem hran.	34
4.5	Levý obraz: originál. Pravý obraz: segmentace s detekcí hran.	34
4.6	a) originál; b) segmentace bez filtrů; c) segmentace s medián filtrem; d) segmentace s detektorem hran.	35
4.7	Originální syntetický obraz (nalevo) a jeho segmentace (napravo) bez použití filtrů.	35

ÚVOD

Tato práce se věnuje watershed segmentaci a morfologickým metodám, které se při této metodě používají. První kapitola shrnuje obecný přehled různých typů segmentací, jako je například hojně využívaná segmentace pomocí prahování. V přehledu je zmíněna i watershed segmentace, jako součást skupiny segmentací založených na principu hledání oblastí.

Další kapitola je věnována základním morfologickým operacím, které jsou součástí dalších sofistikovanějších metod používaných v segmentacích. Je zde představen pojem strukturního elementu a dva nejzákladnější pojmy, jako je dilatace a eroze.

Další sekce se zabývá watershed segmentací a s ní spojenými pojmy, jako jsou např. kruhy a kostra. Tyto pojmy následně souvisí s dalším pojmem geodetické transformace, mezi které patří geodetická vzdálenost, dilatace, eroze a zóna vlivu. Další teoretická část je věnována definici a výpočtu watershedů. Jsou zde nastíněny 2 rozdílné definice. Závěr kapitoly je věnován filtrům používaných k předzpracování obrazu určeného k segmentaci.

Následuje přiblížení implementovaného algoritmu a nastin funkcí jednotlivých tříd programu psaného v jazyce Java. Nechybí zde také seznámení s programem Rapidminer, pro který je tento modul vytvořen a také seznámení s jednotlivými moduly.

V závěru jsou prezentovány výsledky segmentovaných obrazů pomocí vytvořeného programu, zhodnocení dosažené segmentace a použitých filtrů. Nechybí zde srovnání časů segmentace pro různé velikosti obrazů.

1 TECHNIKY SEGMENTACE OBRAZU

V této kapitole bych rád zmínil několik technik segmentace obrazu.

Samotná segmentace obrazu je složitý proces. Tento proces lze rozdělit do několika kroků.

Obraz je většinou nutné určitým způsobem předzpracovat. Například je žádoucí odstranit šum, který je přítomný v každém obrazu nebo také provést odstranění malých detailů. Tyto detaily jsou většinou nepodstatné a mohly by mít vliv na samotnou segmentaci obrazu.

Dalším krokem již většinou bývá samotná segmentace. Tento krok však často nevytvoří plně použitelné výsledky, proto je nutné provést dodatečné zpracování.

1.1 Prahování

Prahování je v podstatě nejjednodušší a nejstarší metoda segmentace obrazu. Tato metoda má omezenou použitelnost, avšak je velmi oblíbená a hojně používaná. Z jednoduchosti vyplývají velmi malé nároky na výpočet, implementaci a čas. V mnoha případech se používá jako součást některých metod.

Segmentace je založená na principu objektů a pozadí. Tento princip spočívá v tom, že objekty a pozadí mají rozdílnou velikost intenzity. Na počátku je definován určitý práh a všechny pixely, které mají menší hodnotu než tato určená hodnota, jsou vyhodnoceny jako pozadí. Ostatní pixely jsou určeny jako objekt, či objekty. Uvedený postup se dá popsat následující rovnicí:

$$\begin{aligned} g(i, j) \geq T, \text{ pak } f(i, j) &= 1 \\ g(i, j) \leq T, \text{ pak } f(i, j) &= 0. \end{aligned} \tag{1.1}$$

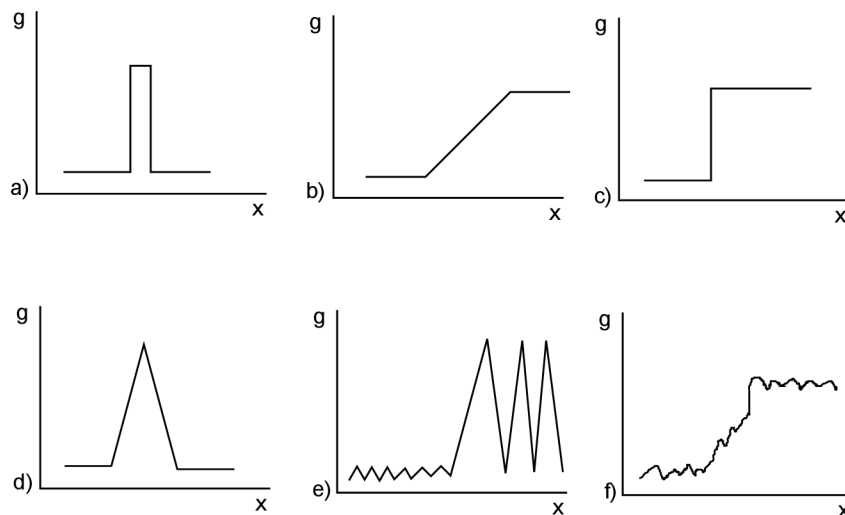
Výraz $g(i, j)$ je originální obraz, $f(i, j)$ je segmentovaný obraz a proměnné i, j jsou jejich souřadnice.

1.2 Segmentace pomocí detekce hran

Tato metoda je zastoupena mnoha různými segmentačními algoritmy. Základem těchto algoritmů jsou informace o hranách v obraze, ze kterých se pak v další fázi procesu určují jednotlivé segmenty obrazu. Ukázka jednotlivých typů hran je uvedena na obr. 1.1. Nejpoužívanější metody detekce hran:

- Metody založené na první derivaci.
- Metody založené na druhé derivaci.
- Cannyho detektor hran.

- Srovnání se vzorem, Template matching.
- Statistické metody.
- Scale space.



Obr. 1.1: Ukázka profilů jednotlivých hran: a) *line edge*; b) *ramp edge*; c) *step edge*; d) *roof edge*; e) *variance edge*; f) *noisy edge*.

1.3 Segmentace na principu hledání oblastí

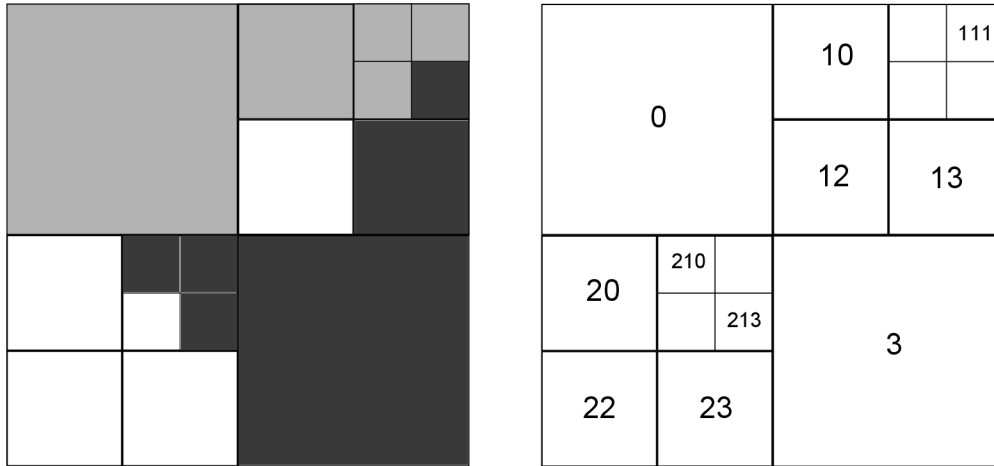
Tento princip nehledá hrany pro segmentování, ale objekty k segmentaci hledá přímo. Hlavní výhodou je zřejmě vysoká odolnost proti šumu, proto se těchto principů s velkým úspěchem používá ve více zašuměných obrazech, kde by segmentace hledající hrany neměly takový výsledek.

U všech těchto metod je nutné definovat určité kritérium homogenity. Toto kritérium může být například úroveň šedi nebo textura.

1.3.1 Split and Merge

Česky je tato metoda nazývána jako dělení a spojování oblastí. Vznikla spojením dvou segmentačních metod **region merging** a **region splitting**. Princip spočívá v dělení oblastí na menší a menší oblasti a sousední oblasti se spojují v případě, že splnily kritéria homogenity. Obraz je dělen do předem dané struktury.

Nejčastější strukturou je Stromová struktura, neboli **quadtree**. Zde se oblast vždy dělí na čtyři čtvrtinové oblasti jak ukazuje obrázek 1.2.



Obr. 1.2: Dělení na čtvrtinové oblasti. [4]

Dále je třeba definovat kritérium homogenity, které je téměř nejdůležitější vlastností dané metody. Toto kritérium je vhodné zvolit podle vzhledu a struktury obrazu a také podle výpočetní náročnosti jednotlivých ukazatelů.

1.3.2 Watershed

Watershed, česky též nazývána záplava, je segmentační metoda, která pohlíží na obraz, jako na reliéf krajiny, který je postupně zaplavován vodou.

Tato metoda je zde uvedena pouze jako jedna část v celkovém rozdělení jednotlivých metod a bude dále rozebrána v samostatné části tohoto dokumentu.

1.3.3 Shluková analýza

Shluková analýza je obecná statistická klasifikační metoda. Ve zpracování obrazu pracuje na principu shlukování pixelů podobných vlastností. Část obrazu nebo dokonce i každý pixel je určen vektorem vlastností $v = [v_1, v_2, \dots, v_n]$. Mohou to být například barevné komponenty pixelů, vlastnosti okolí, pozice atd. Je nutné tyto vlastnosti normalizovat a vybrat takové, které mají z jedné oblasti nejvíce podobné a z jiných oblastí rozdílné.

Na základě této metody pracují i další dvě segmentace:

- Mean-Shift segmentace obrazu.
- K-means segmentace obrazu.

2 MORFOLOGICKÉ OPERACE

V této kapitole bych rád rozebral některé morfologické operace a jejich základy, pro lepší pochopení textu a principů, které zde budou také probírány.

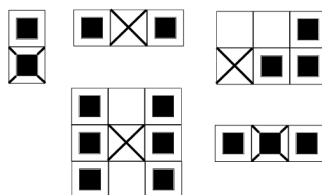
2.1 Základy morfologie

Matematická morfologie pracuje s obrazem zejména na bázi množin. V černobílých obrazech udávají jejich polohu souřadnice x , y (podle dohody buď černé nebo bílé body). V šedotónových obrazech se pro tyto body určuje kromě souřadnic x , y , také další hodnota, které určuje obvykle hodnotu jasu a často nabývá 256 hodnot (0 až 255).

Všechny následující operace budou prezentovány jako binární operace pro snadnější pochopení a vysvětlení základních principů těchto operací. Tyto principy lze změnit, aby se daly použít v jiných než binárních obrazech.

2.1.1 Strukturní element

Strukturní element (angl. structuring element) je v podstatě nějaký obrázek. Používá se hlavně v operacích nad obrázky např. při dilataci, erozi, otevření, zavření a dalších operacích. Výsledek operace je především určován tvarem a velikostí tohoto elementu. Při provádění určitých operací je nutné znát počáteční bod. Některé možné tvary a velikosti strukturních elementů jsou na obr. 2.1. Na každém elementu je křížkem vyznačen jeho počáteční bod pro použití v některých operacích.



Obr. 2.1: Ukázka strukturních elementů.

2.1.2 Dilatace

Dilatace spolu s erozí patří mezi základní operace (morfologie) nad obrázky. Velké množství dalších operací je složeno z těchto dvou základních.

Určíme si množiny X , B , všechny tyto množiny leží v \mathbb{Z}^2 . Poté lze definovat dilataci jako:

$$X \oplus B = \{p \in \mathbb{Z}^2 : p = x + b, x \in X, b \in B\}. \quad (2.1)$$

Příklad z obr. 2.2:

$$X = \{(1, 0), (1, 1), (1, 2), (2, 2), (0, 3), (0, 4)\}.$$

$$B = \{(0, 0), (1, 0)\}.$$

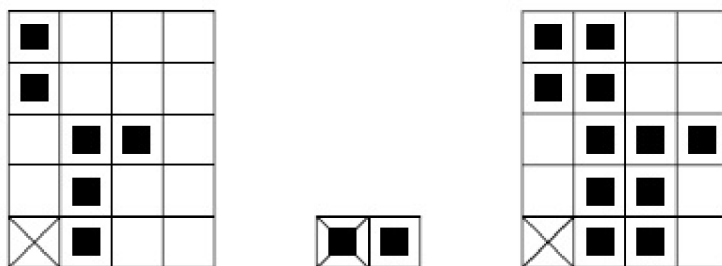
$$X \oplus B = \{(1, 0), (1, 1), (1, 2), (2, 2), (0, 3), (0, 4), (2, 0), (2, 1), (2, 2), (3, 2), (1, 3), (1, 4)\}.$$

Tato operace pracuje tak, že každý pixel z množiny X se sečte s každým pixelem množiny B . Tyto pixely jsou pak prvky výsledné množiny dilatace.

Dilatace se používá k zaplnění malých děr v objektech, zvětší se však původní velikost objektu. Má-li být zachována velikost, je nutné zkombinovat dilataci s erozí.

Pokud dilataci použijeme na šedotónový obraz, plynou z této skutečnosti dva důsledky:

- Výsledný obraz bude světlejší, pokud budou hodnoty strukturního elementu světlé.
- Temné detaily jsou redukovány nebo vyhlazeny, záleží na vlastnostech strukturního elementu.



Obr. 2.2: Binární dilatace.

2.1.3 Eroze

Běžné využití eroze je odstranění nadbytečných detailů, které se opět určují pomocí strukturního elementu. Stejně jako u dilatace je eroze definována takto:

$$X \ominus B = \{p \in \mathbb{Z}^2 : p = x + b \in X \text{ pro každé } b \in B\}. \quad (2.2)$$

Příklad z obr. 2.3:

$$X = \{(1, 0), (1, 1), (1, 2), (0, 3), (1, 3), (2, 3), (3, 3), (1, 4)\}.$$

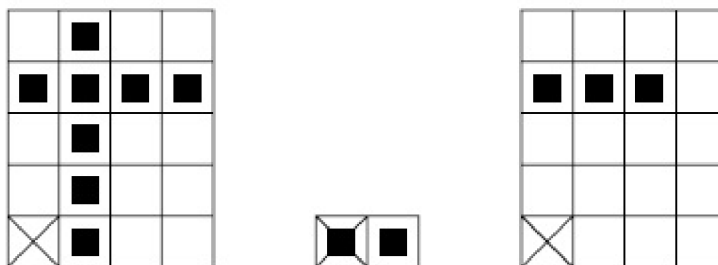
$$B = \{(0, 0), (1, 0)\}.$$

$$X \ominus B = \{(0, 3), (1, 3), (2, 3)\}.$$

Výpočet eroze lze jednoduše popsat tak, že každý pixel z množiny X sčítáme se součtem všech pixelů z množiny B . Každou novou hodnotu porovnáme s jednotlivými hodnotami v množině X , pokud se hodnoty shodují, tato hodnota je zahrnuta do výsledné množiny eroze.

Pokud erozi použijeme na šedotónový obraz, plynou z této skutečnosti dva důsledky [3]:

- Výsledný obraz bude tmavší, pokud budou hodnoty strukturního elementu světlé.
- Světlé oblasti ve vstupním obrázku, které jsou menší než strukturní element, jsou zmenšovány v závislosti na odstínu šedé v okolí těchto míst. Také jsou závislé na maximální hodnotě šedé ve strukturním elementu a jeho tvaru.



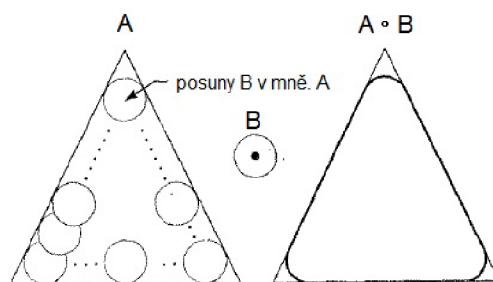
Obr. 2.3: Binární eroze.

2.1.4 Otevření

Otevření neboli opening (obr. 2.4) je v podstatě složením funkcí dilatace a eroze. Otevření vyhlazuje a zaobluje hrany daného předmětu a také maže malé předměty a čáry. Opakované použití otevření nemá již další vliv na výsledný obraz. Tato funkce by se s úspěchem dala použít k předzpracování obrazu některých segmentačních metod, které by přinášely horší výsledky, pokud by v obraze bylo mnoho malých předmětů a šumu. Otevření je definováno následujícím vztahem:

$$A \circ B = (A \ominus B) \oplus B. \quad (2.3)$$

Po rozšíření otevření na šedotónové obrazy se již nepracuje s množinami, ale s funkcemi, proto pro zápis vztahu používáme malá písmena, což je typické značení funkcí.

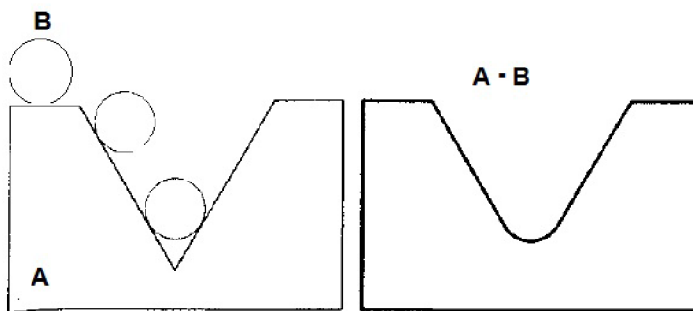


Obr. 2.4: Opening. [3]

2.1.5 Uzavření

Uzavření neboli closing je, stejně jako otevření, složením funkcí dilatace a eroze (obr. 2.5). Uzavření také zjemňuje hrany, ale narozdíl od otevření prodlužuje úzké linky, odstraňuje malé díry nebo je doplňuje, což však záleží na tvaru. Tato funkce by se opět dala použít k předzpracování v různých segmentačních algoritmech. Uzavření je definováno následujícím vztahem:

$$A \bullet B = (A \oplus B) \ominus B. \quad (2.4)$$



Obr. 2.5: Closing. [3]

Po rozšíření uzavření na šedotónové obrazy se již nepracuje s množinami, ale s funkcemi, proto pro zápis vztahu používáme malá písmena, což je typické značení funkcí.

3 WATERSHED SEGMENTACE

Tato kapitola pojednává o obrazové segmentaci watershed pomocí metod morfologického zpracování obrazu. Ukážeme si zde několik základních operací a definicí potřebných pro obecné pochopení principů a částí watershed segmentace.

Segmentace se dá rozdělit do dvou druhů. Je to segmentace textury obrazů a segmentace částí obrazů.

V segmentaci textury je obraz rozdělen do regionů a každý z nich je definován souborem vlastností mikrostruktury obsažené v obrazu. Typická aplikace je v segmentaci typů vegetace v leteckých snímcích, segmentace textu ve stránkách dokumentu a ve zdravotnických snímcích.

Druhý typ segmentace je založen na obrazech částí (nebo objektů), kde informace o textuře není daná nebo nemůže být použita jako diskriminující faktor. Úlohou segmentace je extrahovat části ze studovaného obrazu. Rozlišujeme mezi binární a šedotónovou částicovou segmentací. Úkolem binární segmentace je rozdělit překrývající se části. V šedotónové segmentaci je tento úkol ekvivalentní s problémem určení obrysu.

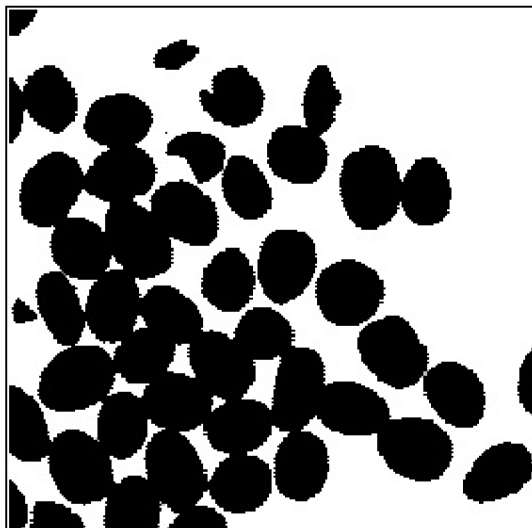
3.1 Úvod do morfologické segmentace

Segmentace poukazuje na úkol spočívající v určení objektů nebo částí z obrazu tak, jak nejlépe to bude možné.

Segmentace obrazu I je více než to, že spočítáme objekty v něm a ukážeme na ně (to také zahrnuje určení okrajů objektu). K určení obrysů je obvykle třeba označit si objekty k dalšímu zpracování. Značkou objektu nebo množiny X je obvykle myšlena množina M , která je obsažená v X . Také obvykle předpokládáme, že značky mají stejnou homotopii jako množina, která je označována. Pro příklad, značka spojených množin je obvykle sama spojena.

Tento krok určování značek je zřejmě nejdůležitějším krokem v mnoho komplexních segmentačních aplikacích. Intuitivně, toto značkování je algoritmickou simulací lidského chování. Když někoho požádáte, aby ukázal všechny objekty na obrázku, nezačne hned určovat obrysy objektů, ale nejdříve tyto objekty označí jeden po druhém. Toto hledání je v obrazové analýze obvykle méně náročné, než přímé hledání přesných obrysů objektů.

V binárním případě, segmentace poukazuje na určení spojených součástí reprezentující objekt zájmu, a také na segmentaci překrývajících se objektů. Jako příklad je uveden obr. 3.1 (kávové zrnka). Tyto kávová zrnka můžeme jednoduše spočítat pomocí oddělení spojené značky a spočítat jednotlivé oddělené značky. Toto však



Obr. 3.1: Binární obrázek kávových zrn. [1]

není zcela vhodné, ve většině případů je nutné tyto objekty oddělit a určit jejich hranice pro další použití.

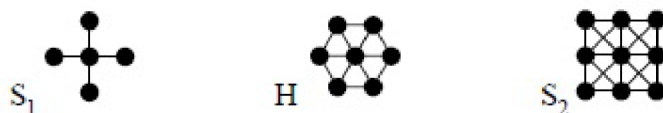
3.1.1 Vztahy mezi pixely

Do této doby jsme pracovali s konkrétními případy. Naše pracovní plocha je diskrétní rovina \mathbb{Z}^2 . V tomto případě *mřížka* G poskytuje sousedské vztahy mezi pixely. Obyčejně používanými mřížkami je čtvercová mřížka, pro kterou má pixel p 4 (4-konektivita) nebo 8 (8-konektivita) sousedů, také se používá hexagonální mřížka (6-konektivita). Dva sousední pixely p a q vytvářejí hranu G . Mřížka G vytváří diskrétní vzdálenost v \mathbb{Z}^2 , potom vzdálenost mezi dvěma pixely je nejmenší počet hran potřebných k tomu, aby se spojily.

3.2 Maximální kruh, kostra a konečná eroze

3.2.1 Kruhy

V rovině \mathbb{R}^2 s obvyklou eukleidovskou vzdáleností jsou kruhy standardní disky. V diskrétní rovině \mathbb{Z}^2 jsou tyto kruhy reprezentovány šestiúhelníky, pokud je použita hexagonální mřížka. Kruhy mohou být reprezentovány čtverci ve čtvercové mřížce. Jednotkový kruh B (kruh o poloměru 1) odpovídá buď S_1 , H nebo S_2 v závislosti na použité konektivě (4-, 6- nebo 8-konektivita), tyto konektivity jsou zobrazeny na obr. 3.2.

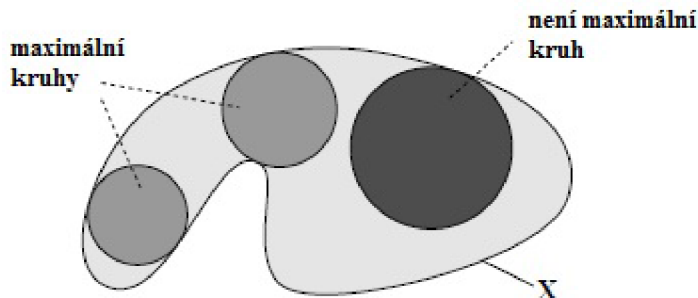


Obr. 3.2: Jednotkový kruh B v různé konektivitě. [1]

Definice maximálního kruhu B :

$$\forall B' \text{ kruhy, } B \subseteq B' \subseteq X \Rightarrow B' = B. \quad (3.1)$$

Tuto definici můžeme slovně vyjádřit takto: Kruh B obsažen v X je nazýván maximální, právě když neexistuje další kruh obsažen v X a obsahující B . Tento koncept přibližuje obr. 3.3



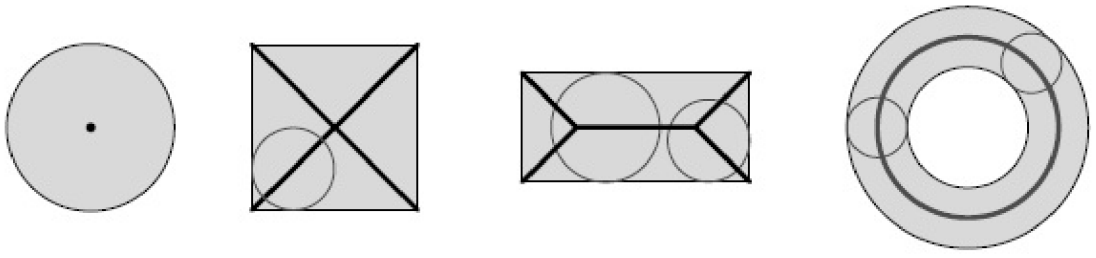
Obr. 3.3: Koncept maximálního kruhu v eukleidovské rovině. [1]

3.2.2 Kostra

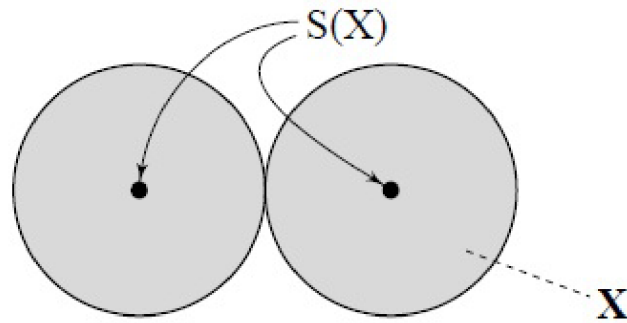
Definice kostry pomocí maximálních kruhů:

$$S(X) = \{p \in X \mid \exists r \geq 0, B(p, r) \text{ je maximální kruh } X\}. \quad (3.2)$$

Kostra je v podstatě intuitivní pojem, např. kostra kruhu je zmenšena do jeho středu. Několik příkladů koster jednoduchých tvarů je ukázáno na obr. 3.4. Problém nastává v situaci dvou tečných kruhů. Pokud si vezmeme dva tečné kruhy ve spojitě eukleidovské rovině \mathbb{R}^2 , tak jejich kostra je zredukována pouze na dva body, které leží ve středech tečných kruhů. V podstatě by ale měly být tyto středy spojeny přímkou linií směřující od jednoho středu tečného kruhu ke druhému. Jinými slovy, homotopie nebo konektivita původního celku není nezbytně zachována. Tuto situaci ilustruje obr. 3.5



Obr. 3.4: Kostry několika jednoduchých tvarů. [1]



Obr. 3.5: Ukázka koster dvou tečných kruhů ve spojitě rovině. [1]

Dále si definujeme nB jako kruh o poloměru n v uvažované konektivitě:

$$nB = \underbrace{B \oplus B \oplus B \oplus \dots \oplus B}_{n\text{-krát}}$$

s B srovnatelným s jedním z S_1 , H nebo S_2 . Kostra z maximálních kruhů může být získána pomocí následujícího vzorce:

$$S(X) = \bigcup_{n=0}^{+\infty} [(X \ominus nB) / ((X \ominus nB) \circ B)]. \quad (3.3)$$

Kostra pomocí maximálních kruhů může být získána jako sjednocení residuí otevření X ve všech měřítkách [1].

Přímá aplikace této metody vede ke kompletně nesouvislým kostrám. Tuto skutečnost můžeme vidět na obr. 3.6, kde jsou kostry jednotlivých zrn, avšak tyto kostry jsou kompletně nesouvislé. Metody k jejich tvorbě nejsou tak přímočaré. V současnosti existuje mnoho algoritmů, které řeší tuto problematiku. Na obr. 3.7a jsou spojené kostry po použití sofistikovanějších algoritmů. Na obr. 3.7b vidíme kostry, které jsou v podstatě ztenčené na minimum, pokud je to nutné pro další zpracování.



Obr. 3.6: Kávová zrna a jejich kostry pomocí maximálních kruhů. [1]

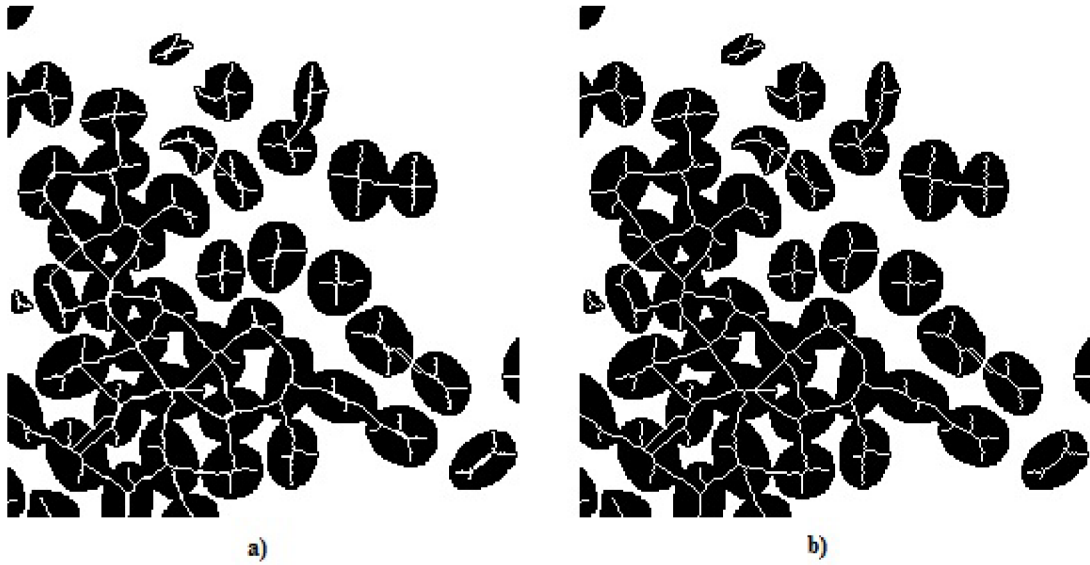
3.2.3 Funkce quench

Každému pixelu p v kostře náleží maximální kruh. Určíme si $q_x(p)$ jako poloměr tohoto kruhu. Pak funkce quench asociuje s každým pixelem $p \in S(X)$ poloměr příslušného maximálního kruhu. Jedním z nejdůležitějších výsledků je údaj dostatečný k rekonstrukci původní množiny. Množina X je rovna sjednocení příslušných maximálních kruhů [1]:

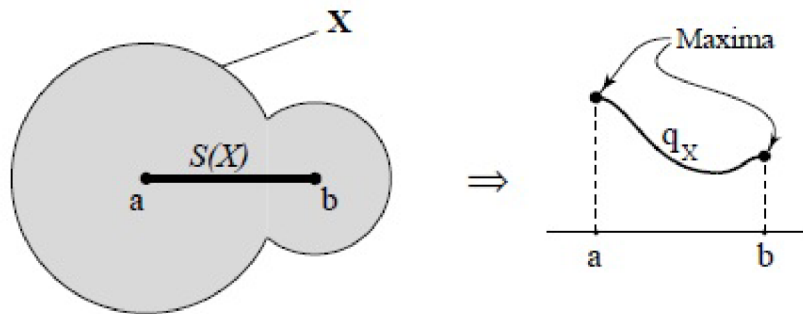
$$X = \bigcup_{p \in S(X)} (p + q_x(p)B). \quad (3.4)$$

Další důležitou částí této funkce je definice konečné eroze. Množina X se dá také vysvětlit jako sjednocení všech maximálních kruhů množiny. Jak je vidět na obrázku 3.8, jsou zde dva spojené objekty a jejich kostra. Napravo je vidět soubor poloměrů všech maximálních kruhů. V obou bodech a a b jsou určeny maxima. Tyto maxima definují značky jednotlivých objektů (i spojených) a jejich množina určuje konečnou erozi, kterou si nyní i s dalšími pojmy definujeme podle [1]:

- **Konečná eroze:** Konečná eroze množiny X , určená $Ult(X)$, je množina regionálních maxim funkce quench.
- **Regionální maximum:** Regionální maximum šedotónového obrazu je spojení pixelů s určenou hodnotou h (výška), sousedství má hodnotu striktně menší než h .
- **Lokální maximum:** Pixel p šedotónového obrazu I je lokální maximum právě tehdy, když každý pixel q je sousedící s pixelem p , potom platí $I(p) \geq I(q)$.



Obr. 3.7: a) kompletní spojená kostra; b) ztenčená kostra. [1]

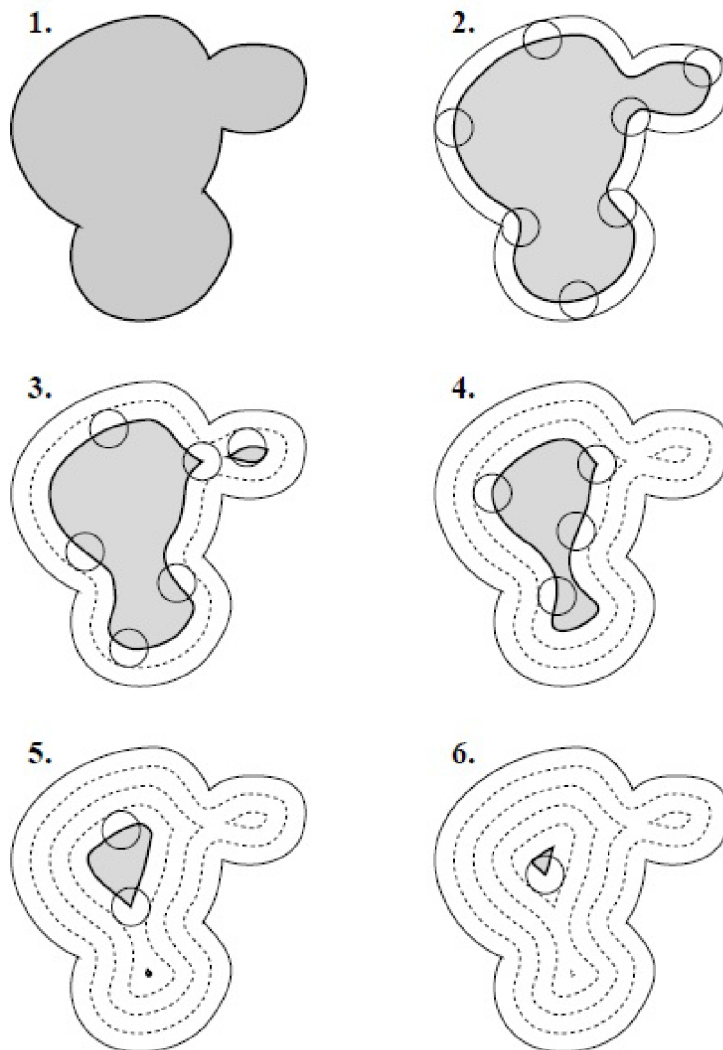


Obr. 3.8: Kostra dvou překrytých objektů a jejich maxima. [1]

3.2.4 Konečná eroze a funkce vzdálenost

V této části se budeme věnovat jiné interpretaci konečné eroze. Určíme si X jako dva překrývající se objekty. Na tyto objekty budeme aplikovat iterativní (postupnou) erozi pomocí jednotkového kruhu B . Tyto objekty jsou zmenšeny a odděleny od celku předtím, než je kompletně odstraní proces eroze. Tento proces je předveden na obr. 3.9.

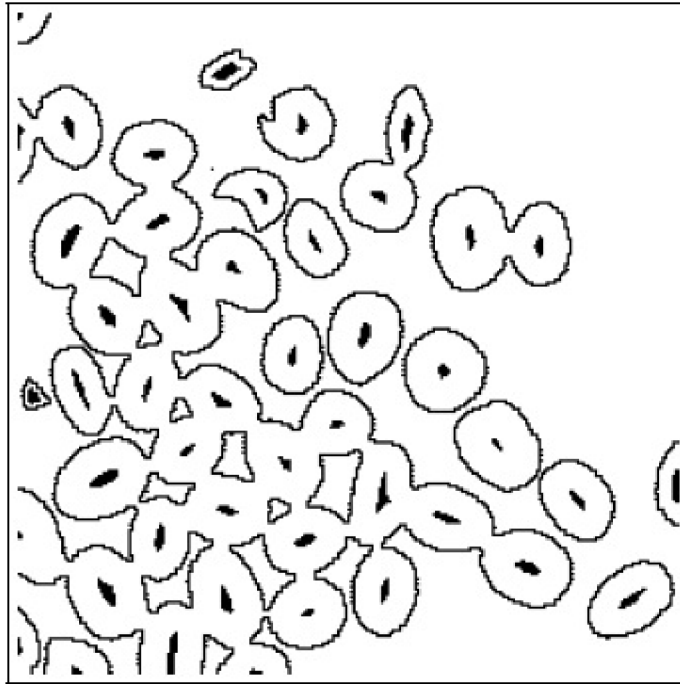
Funkce vzdálenosti je morfologická transformace, která sdružuje všechny informace postupné eroze množiny X . Vzdálenost $dist_x(p)$ je vzdálenost mezi každým pixelem p a pozadím. Regionální maxima jsou maxima pro které platí, že výška sousedních pixelů je jedinečně nižší než výška maxim h . Tato skutečnost je uvedena v [1]. Tyto maxima tedy patří do konečné eroze množiny X a můžeme říct, že konečná



Obr. 3.9: Proces iterativní eroze. [1]

eroze je rovna sjednocení regionálních maxim vzdálenostní funkce množiny X .

Touto funkcí vzniknou značky objektů (např. značky jednotlivých kávových zrn), ale není zaručeno, že každý objekt bude mít jen jednu značku. Obvykle se stává, že při použití vzdálenostní funkce je značka roztrhána na více značek. Korekce však můžeme dosáhnout, pokud odečteme jedničku od vzdálenostní funkce v místě všech komponentů konečné eroze. Pak dostáváme modifikovanou funkci vzdálenosti, jejíž maxima jsou přesně naše požadované značky. Výsledek této funkce si můžeme prohlédnout na obr. 3.10.



Obr. 3.10: Objekty kávových zrn s vlastními značkami. [1]

3.3 Geodetické transformace

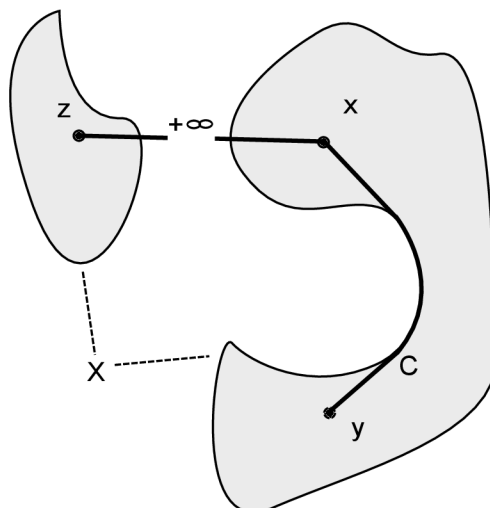
3.3.1 Geodetické vzdálenosti

Myšlenka je určit každý objekt (v našem příkladu kávové zrno) jako region obrazu vystředěný kolem jeho značky. Záměrem je nechat „vyrůst“ tyto značky zpátky. Pro tento účel si definujeme geodetické operátory založené na představě geodetické vzdálenosti, které nebudou operovat v celém prostoru, ale jen v konečné množině X nazývané jako maska.

Dále zde budou uvedeny 2 definice z dokumentu [1]:

- **Geodetická vzdálenost** $d_X(x, y)$: Geodetická vzdálenost mezi dvěma body x a y z X je infimum délky všech cest mezi x a y v X , pokud nějaká cesta C existuje. Pokud žádná cesta neexistuje, pak $d_X(x, y) = +\infty$.
- **Geodetický kruh** $B_X(p, n)$: Geodetickým kruhem nazýváme kruh s poloměrem n a středem $p \in X$.

Princip geodetické vzdálenosti je na obr. 3.11.



Obr. 3.11: Geodetická vzdálenost.

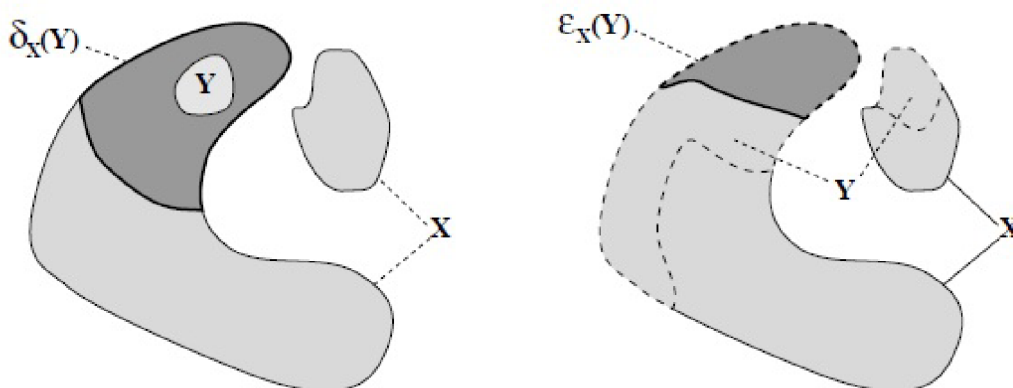
3.3.2 Geodetické dilatace a eroze

Pokud každý pixel p , který leží v X , přiřadíme geodetickému kruhu $B_x(p, n)$ s poloměrem n a se středem v p , pak si můžeme definovat geodetickou dilataci $\delta_X^{(n)}(Y)$ a erozi $\epsilon_X^{(n)}(Y)$. Tyto definice vychází z dokumentu [1] a vypadají následovně:

$$\delta_X^{(n)}(Y) = \bigcup_{p \in Y} B_X(p, n) = \{p \in X, \exists p' \in Y, d_X(p', p) \leq n\}, \quad (3.5)$$

$$\epsilon_X^{(n)}(Y) = \{p \in Y \mid B_X(p, n) \subseteq Y\} = \{p \in Y \mid \forall p' \in X/Y, d_X(p, p') > n\}. \quad (3.6)$$

Příklady geodetické dilatace a eroze jsou na obr. 3.12.



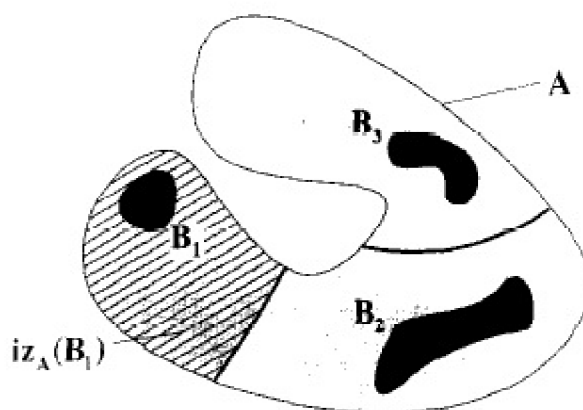
Obr. 3.12: Geodetická dilatace a eroze. [1]

Jak je vidět, tak výsledek geodetických operací na množině $Y \subseteq X$ opět leží v množině X . Pokud si zvolíme n (velikost) geodetické dilatace, pak tato dilatace

může být uskutečněna tolika jednotkovými dilatacemi (velikost 1), jako je velikost výsledné dilatace. Totéž lze určit i pro geodetické eroze.

3.3.3 Geodetická zóna vlivu

Můžeme si definovat geodetickou zónu vlivu $iz_A(B_i)$. Tato zóna je soubor všech bodů (pixelů) v A , jejichž geodetická vzdálenost do B_i je menší než geodetická vzdálenost do ostatních B [2]. Tuto situaci ilustruje obr. 3.13.



Obr. 3.13: Geodetická zóna vlivu spojené složky B_1 uvnitř množiny A . [2]

V geodetické zóně vlivu se dají určit body (pixely), které nepatří ani do jedné zóny vlivu. Tohoto využívá *kostra zónami vlivu (SKIZ)*. Tyto body, které nepatří do žádné zóny vlivu vytvoří SKIZ v množině A a měly by vytvořit souvislou čáru. Ve většině případů nejsou tyto čáry souvislé a neoddělují tak jednotlivé zóny. Toto je způsobeno body, které jsou ve stejné vzdálenosti od obou spojených komponentů.

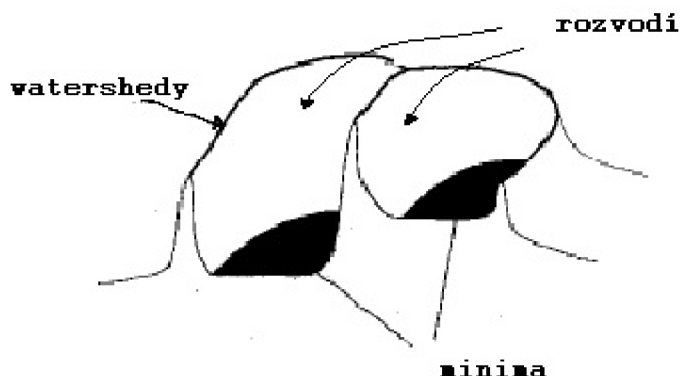
3.4 Definice a výpočet watershedů

Pro další práci je nutno si definovat *minimum* M , což je vlastně plošina pixelů výšky h , z které není možno dosáhnout nižší výšky bez toho, aniž bychom museli vylézt (zvýšit výšku) [2]. S tímto je spojováno lokální a regionální minimum, což se dá považovat za obrácenou analogii k již zmiňovaným maximům. V této kapitole uvedeme dva pohledy na definice watershedů.

3.4.1 Definice z hlediska nejstrmějšího spádu

Nejprve zde zmíníme pojem *rozvodí* $C(M)$ přidružené k minimu M . Je to vlastně množina pixelů oblasti D_I obrazu I taková, že padající kapka vody stéká dolů po reliéfu a sleduje určitou klesající cestu a nakonec dosáhne M [2].

Linie, které oddělují jednotlivá rozvodí se nazývají *watershedy* obrazu I . Všechny tyto skutečnosti jsou velmi dobře viditelné na obr. 3.14. Rozvodí se v podstatě



Obr. 3.14: Minima, rozvodí a watershedy.

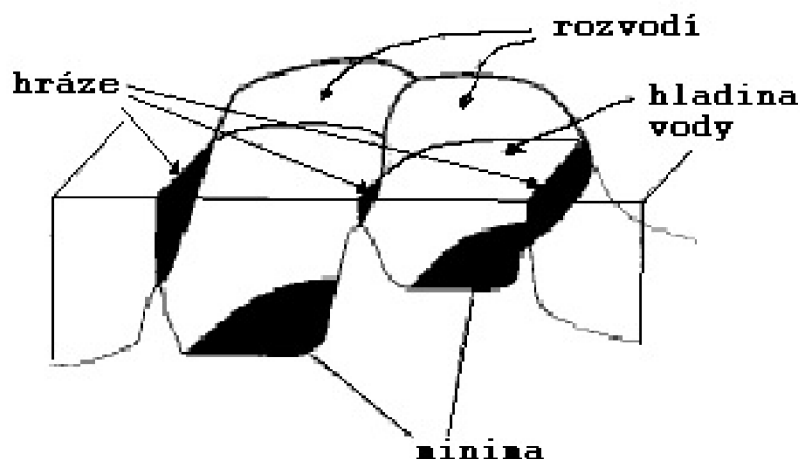
shoduje s minimem, které bylo vytvořené *zónami vlivu*. Z předešlého se dá vytušit, že je zde úzké spojení s funkcí *SKIZ* a watershedy.

3.4.2 Definice pomocí ponoření

Tato část definice představující watershedy je více vhodná pro praktické implementace. Obraz I si můžeme představit jako plochu, v jejíchž minimech vytvoříme díry, kterými může pronikat voda [2]. Nyní začneme tuto plochu postupně ponořovat do jezera. Voda od nejnižších minim začne postupně zaplavovat různé rozvodí obrazu I . V místě, kde se voda při postupném zaplavování začne slévat vybudujeme tzv. *hráz*, kterou můžeme vidět na obr. 3.15. Na konci zaplavení jsou všechny minima obklopeny hrázemi, které přesně vymezují přidružené rozvodí. Tyto hráze se shodují s watershedy obrazu I .

3.4.3 Metody proti přesegmentování

Pokud použijeme watershed segmentaci na neupravený obraz, často dochází k přesegmentování obrazu, vzniká mnoho segmentů a snižuje se jeho použitelnost. Nejlepším způsobem k odstranění tohoto problému je použití obrazového filtru nebo kombinace filtrů před výsledným segmentováním. Většina těchto filtrů pracuje na principu



Obr. 3.15: Budování hrází tam, kde se voda z různých rozvodí začne slévat.

odstranění šumu z obrazu, z čehož vyplývá i snížení počtu minim, z kterých dochází k záplavě obrazu.

Zde je výčet několika známých filtrů:

- **Gaussův filtr:** Jedná se o lineární filtr, který pracuje na principu konvoluce obrazu s maskou, která se skládá z prvků určených Gaussovou funkcí. Použití filtru vede k rozmazání obrazu.
- **Průměrování:** Další filtr ze skupiny lineárních filtrů. Výsledná hodnota každého pixelu je určena průměrem daného pixelu a pixelů sousedních. Jedná se o speciální případ Gaussova filtru.
- **Dolní propust:** Lineární filtr, který propustí pouze nízké frekvence (šum většinou vysokofrekvenční). Možnost aplikace pomocí konvoluce s malou konvoluční maticí.
- **Mediánový filtr:** Ze skupiny nelineárních filtrů. Filtr vezme pro daný pixel pixely z jeho okolí a z těchto všech pixelů vytvoří medián, který je hodnotou nového pixelu. Je vhodný k odstranění náhodného šumu.

Pro lepší výsledky se v kombinaci s některými filtry používá detekce hran, která je více popsána v kapitole 1.2. Po aplikaci dojde k vystoupení hran z tmavšího pozadí.

4 IMPLEMENTACE

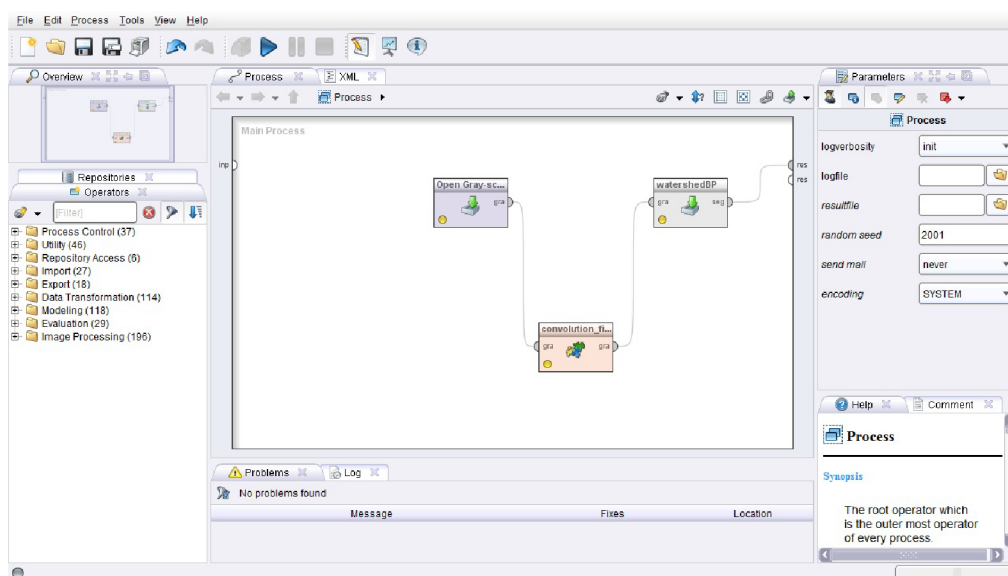
4.1 RapidMiner

V této kapitole je uvedeno pouze hrubé seznámení s programem RapidMiner, do kterého je implementovaný vytvářený modul pro watershed segmentaci.

RapidMiner je open-source systém pro získávání různých informací. Je kompletně napsán v jazyce Java, což umožňuje program spouštět téměř na všech operačních systémech, je však nutné mít nainstalován Java Runtime Environment (JRE).

Po spuštění programu se nám zobrazí úvodní obrazovka, kde si můžeme zvolit z již vytvořeného projektu nebo si vytvořit projekt nový. V novém projektu je vidět pracovní plocha a na levé straně zásuvné moduly, které při přetáhnutí vytvoří na ploše grafický objekt. Tento objekt má určité vstupy a výstupy (podle druhu modulu). Jednotlivé moduly se dají spojovat čarami. Tímto lze spojit výstup jednotlivého modulu se vstupem dalšího nebo zobrazit konečný výstup.

Napravo se pak zobrazují další možnosti (je to např. cesta k souboru) modulu. Na horní liště lze spustit výpočet modrým trojúhelníkem. Okno s touto pracovní plochou můžeme vidět na obr. 4.1. Na konci výpočtu se v nové záložce zobrazí výsledky.



Obr. 4.1: Ukázka pracovní plochy programu RapidMiner.

4.2 Proces segmentace

Celý watershed modul pro program RapidMiner je psán v jazyce Java. V této části je vysvětlen použitý algoritmus, který obsahuje třída „WAlgorithm.java“. Dále jsou zde popsány další pomocné třídy a jednotlivé moduly procesu segmentace v programu RapidMiner.

4.2.1 Moduly

Vytvořený proces segmentace je tvořen několika moduly, které jsou vzájemně propojeny. Konečným modulem je modul „watershedBP“, jehož vstupem je šedotónový obraz typu `ImagePlusGSIObject`. Výstupem jsou 3 nová okna. V prvním okně je zobrazen výsledný obraz s černými watershedy, druhé okno obsahuje obraz s bílými watershedy a poslední obraz je pouze černý s bílými watershedy. Tyto způsoby zobrazení jsou zvoleny z důvodu lepší viditelnosti jednotlivých segmentů, kdy jsou výsledné obrazy někdy tmavější a někdy světlejší povahy.

Na počátku celého řetězce je modul „Open Gray-scale Image“, který načítá zvolený obraz a na výstup předává upravený šedý obraz.

Tento výstup je spojen se vstupem modulu „Median Filter“, jehož výstupem je obraz upravený pomocí filtru typu medián, což je v podstatě jiný způsob průměrování obrazu. Tento filtr snižuje počet minim v obraze, tudíž nedochází k velkému přesegmentování obrazu. Je zde také možnost zadání počtu iterací (průchodů). Pomocí testování bylo zjištěno, že nejlepší počet iterací je mezi 15 až 20. Vyšší počet iterací již zvyšuje počet segmentů nebo je rozdíl mezi segmentacemi pouze nepatrný.

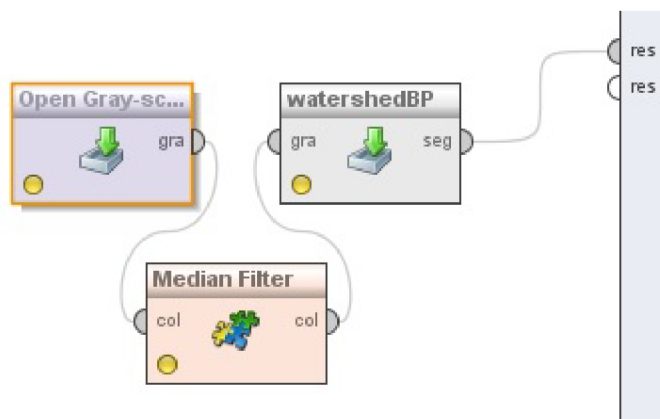
Na CD jsou přiloženy 2 scénáře, jeden je s mediánovým filtrem a druhý s detektorem hran, který má nastavené jádro na „Laplacian of Gaussian 9x9“. Při použití tohoto filtru dochází v některých případech k lepší segmentaci s méně segmenty, které více kopírují skutečný tvar objektů v obraze.

Výsledný filtrovaný obraz je poslán na vstup posledního modulu. Celý systém modulů a jejich propojení lze vidět na obr. 4.2

4.2.2 Algoritmus

Pro implementaci byl vybrán algoritmus, který sestavil Lee Vincent a Pierre Soille v roce 1991. Zdrojový kód je k vidění v pseudokódu v příloze A. Jedná se o metodu, která využívá principu ponoření (immersion), který byl popsán v kapitole 3.4.2. Dále také pracuje s funkcí SKIZ, která je popsána v kapitole 3.3.3. Tento algoritmus je efektivní a ve velké míře se využívá dodnes.

V algoritmu jsou dva základní kroky. V první řadě se jedná o řazení pixelů podle jejich narůstající hodnoty šedé pro přímý přístup k pixelům na určité úrovni šedi.



Obr. 4.2: Moduly pro segmentaci a jejich propojení.

Dalším krokem je zaplavování procházející úroveň po úrovni a začínající z minim. Implementace používá frontu pixelů FIFO. Ta pracuje na principu datové struktury, v které první položka, která vstupuje dovnitř, také vystupuje jako první ven. Na této struktuře mohou být prováděny následující operace: `fifo_add(p,queue)` přidá pixel na konec fronty, `fifo_remove(queue)` odebere první položku z fronty a vrátí její hodnotu, `fifo_init(queue)` inicializuje prázdnou frontu a `fifo_empty(queue)` je testovací funkce, která vrací hodnotu `true`, pokud je fronta prázdná. Hodnota `false` je vrácena v případě, že fronta obsahuje alespoň jeden prvek.

Algoritmus přiřazuje rozdílnou značku `lab[]` každému minimu a jeho přidruženému povodí pomocí iterativních záplav obrazu. V zaplavovacím kroku jsou všechny pixely se stupněm šedi `h` označeny značkou `MASK`. Pak všechny pixely, které mají označeny všechny sousedy z předcházející iterace, jsou vloženy do fronty a z těchto pixelů jsou geodetické zóny vlivu šířeny uvnitř množiny maskovaných pixelů. Jestliže pixel náleží do dvou nebo více rozdílných povodí, pak je označen jako watershed pixel se značkou `WSHED`. Pokud může být pixel dosažen z pixelů se stejnou značkou, pak je pixel sloučen s odpovídajícím povodím. Pixely, které na konci stále mají značku `MASK`, patří do množiny nových minim na úrovni `h`, jejíž spojené součásti dostanou novou značku.

4.2.3 Třídy

Pro správný chod algoritmu jsou použity další třídy. V první řadě je to „WatershedOperator.java“. Jedná se o hlavní třídu, která vytváří modul pro segmentaci v RapidMiner. Jsou zde definovány vstupy a výstupy a také volání watershed segmentace.

Další třídou je „WFIFO.java“. Je zde vytvořen abstrakt FIFO zásobníku, který je použit v segmentačním algoritmu pro zaplavování a práci s pixely. Pro datovou strukturu byl použit komponent `LinkedList` z knihovny `java.util` který umožňuje použít všechny funkce, které jsou popsány v sekci 4.2.2.

Třída „WPixel.java“ reprezentuje jeden pixel, který zpracovává watershed algoritmus. Jedná se o strukturu do níž se ukládá například hodnota pixelu a pole sousedů také ve formátu `WPixel`. Jsou zde definovány funkce pro označení pixelu značkami, aby bylo možné identifikovat například zamaskování pixelů nebo pixely, které tvoří watershed linie.

Dále je to „WStructure.java“ třída, která vytváří strukturu všech pixelů obrazu. Tyto pixely jsou seřazeny podle jejich hodnoty šedé od nejnižší (černá) po nejvyšší (bílá). Pro řazení prvků a jejich vkládání byl použit komponent `Vector` (stejně jako v předešlé třídě), který umožňuje vkládání prvků (typu `WPixel`) za sebou a následně jejich řazení.

4.3 Výsledky

Pro zhodnocení výsledků byl vybrán obraz o velikosti 1600x1200 pixelů, ten byl zmenšen na 800x600 pixelů a 400x300 pixelů. Tyto 3 obrazy byly podrobeny segmentování bez filtru, s mediánovým filtrem (15 iterací) a filtrem s detekcí hran. Výsledky jsou zobrazeny v tabulce 4.1. Vyhodnocené časy segmentování jsou vypočítány od počátku segmentování, do konce segmentování. Nejsou zde zahrnuty časy potřebné k filtraci a časy potřebné ke zobrazení výsledků.

Velikost	rychlost [ms]		
	Bez filtru	Medián	Detekce hran
400x300	143	155	158
800x600	691	799	735
1600x1200	3403	3065	3092

Tab. 4.1: Rychlosti výpočtů watershedů obrazů.

Pro srovnání jsou zde uvedeny příklady různých obrazů 4.3, 4.4 a jejich segmentací bez filtru, s mediánovým filtrem a filtrem s detekcí hran.

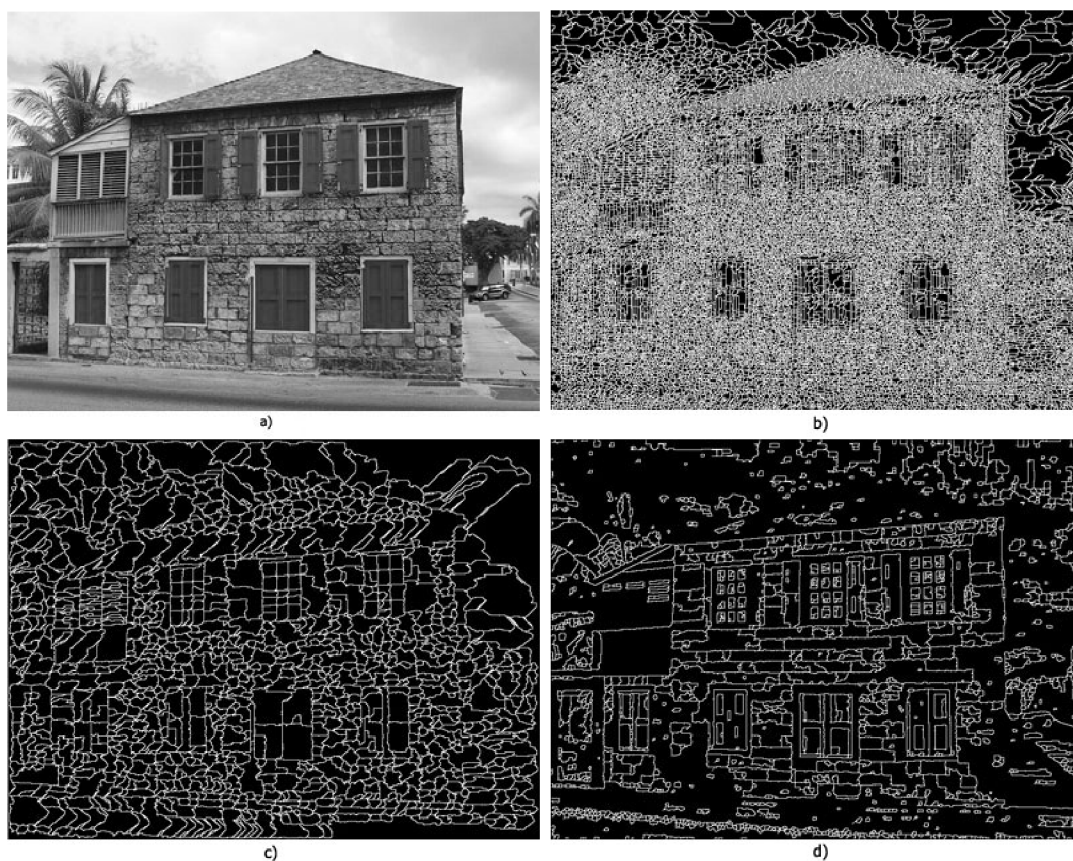
Na obr. 4.3 je vidět, že při nepoužití žádného filtru dochází k velkému přesegmentování obrazu. S mediánovým filtrem dosahuje segmentace lepších výsledků, stejně jako s filtrem, který detekuje hrany. Ovšem segmentace s mediánem je zde zřejmě ve výsledku lepší, než s detekcí hran.

V případě obr. 4.4 platí totéž s tím, že segmentace s detekcí hran velmi dobře detekuje středy jednotlivých květin v obraze. Segmentace s mediánem dobře segmentuje celé květy.

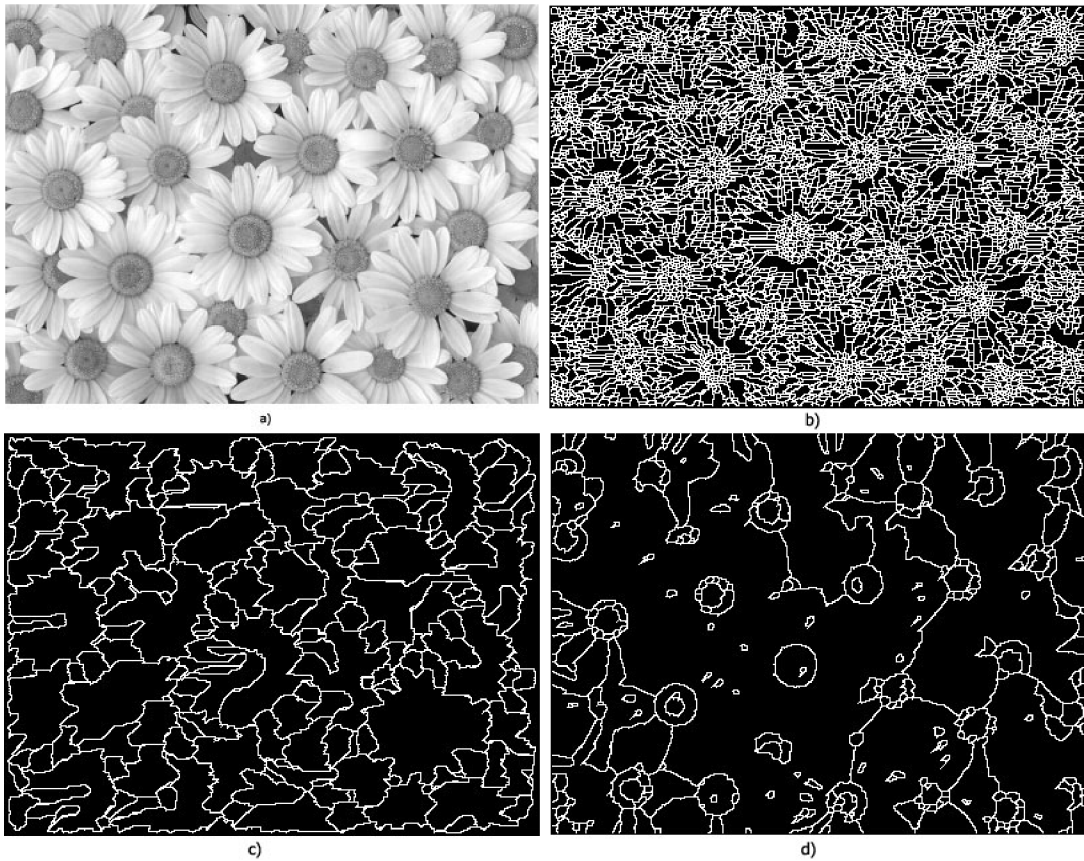
Na obr. 4.5 byla použita segmentace s filtrem s detekcí hran. Na obraze jsou jasně zřetelné segmenty jednotlivých jablek.

Dále byl pro segmentaci použit ve své podstatě binární obraz, kde na bílém pozadí jsou zaneseny černé objekty různých tvarů. Na tomto obraze je demonstrována použitelnost segmentace a její výsledky pro binární obraz, které jsou na obr. 4.6.

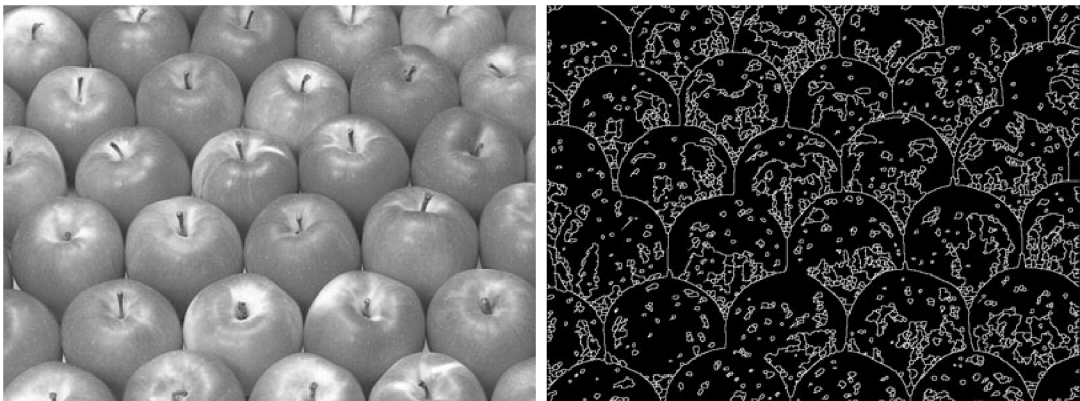
Na posledním obraze 4.7 je demonstrována základní funkce segmentačního algoritmu. Tento obraz je syntetický a byl vytvořen uměle za účelem jednoduchého zobrazení několika rozvodí s příslušnými minimy, od kterých začíná zaplavování reliéfu obrazu.



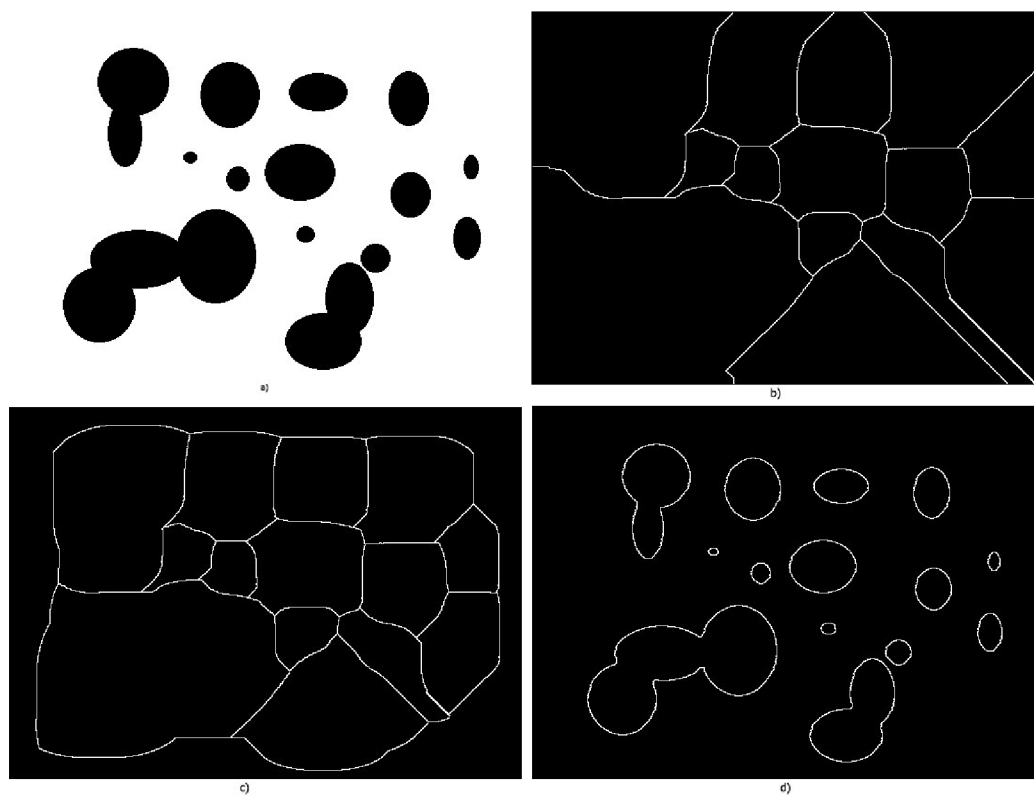
Obr. 4.3: a) originál; b) segmentace bez filtrů; c) segmentace s medián filtrem; d) segmentace s detektorem hran



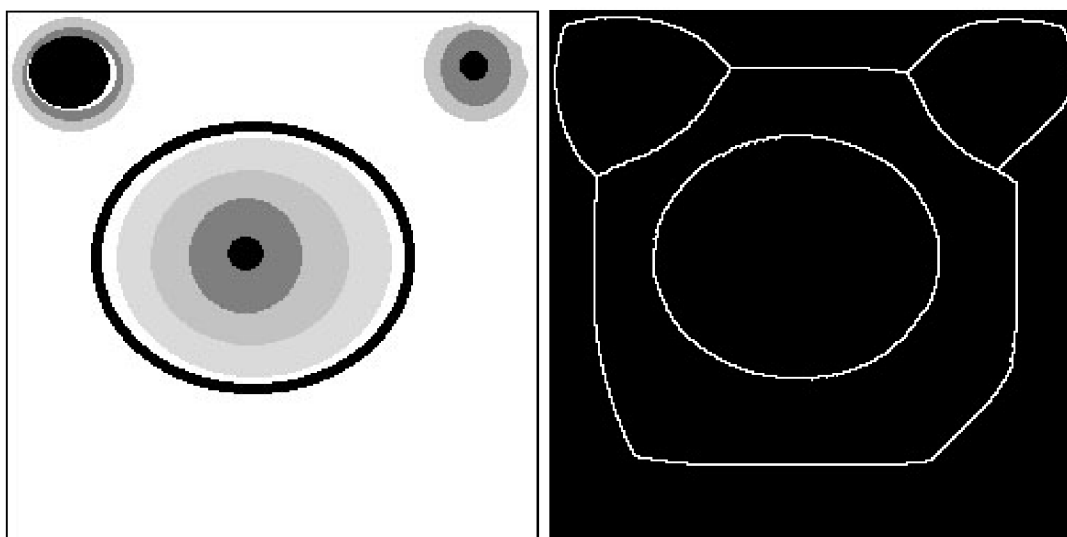
Obr. 4.4: a) originál; b) segmentace bez filtrů; c) segmentace s medián filtrem; d) segmentace s detektorem hran.



Obr. 4.5: Levý obraz: originál. Pravý obraz: segmentace s detekcí hran.



Obr. 4.6: a) originál; b) segmentace bez filtrů; c) segmentace s medián filtrem; d) segmentace s detektorem hran.



Obr. 4.7: Originální syntetický obraz (nalevo) a jeho segmentace (napravo) bez použití filtrů.

5 ZÁVĚR

Na začátek zde byly nastíněny některé běžné metody segmentace obrazu, z nichž asi nepoužívanější je metoda prahování. Je to z důvodu její jednoduchosti zpracování a implementace. Poslední kategorie byla segmentace na principu hledání oblastí.

Další kapitola byla věnována základním morfologickým operacím a základům morfologie. Všechny zde rozebírané metody byly vysvětleny na binárním obraze. Mezi takoveto základní operace patří dilatace a eroze. Pomocí dilatace lze s úspěchem zaplňovat malé díry v objektu, avšak zvětší se jeho velikost. Eroze je v podstatě duální funkcí k dilataci, kdy dochází ke zmenšování objektu.

Všechny předchozí kapitoly byly věnované pouze obecným metodám a postupům, které nám měly více přiblížit další probírané pojmy a skutečnosti. Kapitola Watershed segmentace se zabývá vztahy mezi pixely, kde jsou zavedené různé způsoby konektivity obecně používané v různých segmentacích. V návaznosti na tyto konektivity byl zaveden pojem kruh, který je reprezentován různými tvary, které souvisí s konektivitou, ve které jsou vytvářeny.

Tyto kruhy jsou v další části užívány k tvorbě koster jednotlivých objektů, které se nachází v obraze. Setkáváme se zde s pojmem maximální kruhy, což jsou kruhy, které jsou tečné ve dvou bodech k obrysu předmětu a jsou celé obsaženy v tomto objektu. Obdobou tohoto je funkce quench, která zaznamenává velikost poloměru těchto kruhů, podle nichž se určuje množina maxim. V souvislosti s tímhle je zaveden pojem lokální a regionální maximum a pojem konečná eroze.

S konečnou erozí je spojena funkce vzdálenost. Je to morfologická transformace, která sdružuje všechny informace postupné eroze. Jsou vytvářeny určité vrstevnice a tyto vrstevnice jsou analogií k reliéfu rozvodí. Pomocí těchto funkcí lze určit značky objektů pro úspěšnou segmentaci obrazu. S těmito značkami koresponduje i definice rozvodí, kdy tyto značky jsou v podstatě minima objektu a kolem nich vznikají rozvodí.

Poslední část se zabývá implementací segmentačního algoritmu a použitím filtrů k předzpracování obrazu. Tento algoritmus je napsán v jazyce Java, což v mnoha ohledech usnadňuje návrh programu.

Ve výsledcích byl algoritmus podroben zkoušce časové náročnosti výpočtu na jednotlivých testovacích obrazech. Z měření vyplývá, že s narůstající velikostí obrazu exponenciálně roste i čas potřebný k výpočtu segmentů. U nejvyšší velikosti obrazu se doba pohybovala kolem 3 sekund. Jak je dále vidět, na rychlost výpočtu nemá vliv v podstatě žádný z použitých filtrů.

V závěru je zobrazen přehled některých obrazů a jejich segmentací při použití různých filtrů. Z obrazů vyplývá, že ne vždy je typ filtru vhodný pro předzpracování obrazu určeného k segmentaci.

Ačkoliv je výsledkem segmentace často přesegmentovaný obraz, je tento typ segmentace v oblasti zpracování obrazu hojně využíván a bývá součástí komplexních procesů zpracování obrazu, které se liší v závislosti na konkrétní aplikaci. Z tohoto pohledu byl pro nástroj RapidMiner vytvořen samostatný funkční blok watershed segmentace, který tak lze dále využívat v různých úlohách zpracování obrazu.

LITERATURA

- [1] VINCENT, Luc. *Lecture notes on granulometries, segmentation, and morphological algorithms*. Zakopane (Polsko), 1995. Dostupné z: <http://www.vincent-net.com/luc/papers/95zakopane_course.pdf>.
- [2] VINCENT, L. a P. SOILLE. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1991, roč. 13, č. 6, s. 583-598. ISSN 01628828. DOI: 10.1109/34.87344. Dostupné z: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=87344>>.
- [3] PEŘINKA, Michal. *Morfologie a algoritmy počítačového vidění* [online]. 2011 [cit. 2012-12-12]. Bakalářská práce. UNIVERZITA PALACKÉHO V OLOMOUCI, Přírodovědecká fakulta. Vedoucí práce Michal Dobeš. Dostupné z: <<http://theses.cz/id/5dufr2/>>.
- [4] STRAKA, Stanislav. *Segmentace obrazu* [online]. 2009 [cit. 2012-12-12]. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Radka Pospíšilová. Dostupné z: <http://is.muni.cz/th/72784/fi_m/>.
- [5] VINCENT, L. Morphological Grayscale Reconstructuion in Image Analysis: Applications and Efficient Algorithms. *IEEE Transactions on Image Processing*. 1993, 176-201.
- [6] ROERDINK a Arnold MEIJSTER. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*. 2001, č. 41, s. 187-228. Dostupné z: <<http://www.cs.rug.nl/~roe/publications/parwshed.pdf>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

\oplus dilatace

\ominus eroze

T práh

\circ otevření

\bullet uzavření

X, B, Y množiny

I množina obrazu

M množina minima

nB kruh o poloměru n

$d_X(x, y)$ geodetická vzdálenost mezi dvěma body

$B_X(p, n)$ geodetický kruh

p pixel

n poloměr

$\delta_X^n(Y)$ geodetická dilatace

$\epsilon_X^n(Y)$ geodetická eroze

$iz_A(B_i)$ geodetická zóna vlivu

SKIZ kostra zónami vlivu

$C(M)$ rozvodí

D_I oblast obrazu

JRE Java Runtime Environment

SEZNAM PŘÍLOH

A Watershed algoritmus	41
B Obsah CD a manuál k programu	44

A WATERSHED ALGORITHMUS

```
1: procedura Watershed-by-Immersion
2: VSTUP: digitální šedý obraz  $G = (D, E, im)$ .
3: VYSTUP: segmentovaný obraz  $lab$  v  $D$ .
4: #define INIT -1 (*počáteční hodnota  $lab$  obrazu *)
5: #define MASK -2 (*počáteční hodnota na každém stupni *)
6: #define WSHED 0 (*značka watershed pixelu*)
7: #define FICTITIOUS (-1,-1) (*fiktivní pixel  $\notin D$ )
8: curlab  $\leftarrow$  0 (*curlab je aktuální značka*)
9: fifo_init(queue)
10: for all  $p \in 2D$  do
11:      $lab[p] \leftarrow$  INIT;  $dist[p] \leftarrow$  0
        (*dist je pracovní obraz vzdáleností*)
12: end for
13: Třídí pixely podle narůstající hodnoty šedé
        (minimum  $h_{min}$ , maximum  $h_{max}$ )
14:
15: (*Začátek zaplavování*)
16: for  $h = h_{min}$  to  $h_{max}$  do
        (* Geodetický SKIZ úrovně  $h - 1$  uvnitř úrovně  $h$ *)
17:     for all  $p \in D$  with  $im[p] = h$  do
        (*zamaskuje pixely na úrovni  $h$ *)
18:     (*jsou přímo přístupné kvůli kroku řazení*)
19:          $lab[p] \leftarrow$  MASK
20:         if  $p$  has a neighbour  $q$  with ( $lab[q] > 0$  or
             $lab[q] = WSHED$ ) then
21:     (*Načte frontu se sousedy na úrovni  $h$  aktuálních povodí
        nebo watershedů*)
22:          $dist[p] \leftarrow$  1 ; fifo_add( $p, queue$ )
23:     end if
24: end for
25:  $curdist \leftarrow$  1; fifo_add(fictitious, queue)
26: loop (*rozšíří povodí*)
27:      $p \leftarrow$  fifo_remove(queue)
28:     if  $p = FICTITIOUS$  then
29:         if fifo_empty(queue) then
30:             BREAK
31:         else
```

```

32:         fifo_add(FICTITIOUS, queue);   curdist ← curdist + 1;
33:         p ← fifo_remove(queue)
34:     end if
35: end if
36: for all q ∈ N_G(p) do
    (*značení p pomocí kontroly sousedů*)
37:     if dist[q] < curdist and (lab[q] > 0 or
        lab[q] = WSHED) then
38: (*q patří do existujícího povodí nebo watershedů*)
39:         if lab[q] > 0 then
40:             if lab[p] = MASK or lab[p] = WSHED then
41:                 lab[p] ← lab[q]
42:             else if lab[p] ≠ lab[q] then
43:                 lab[p] ← WSHED
44:             end if
45:         else if lab[p] = MASK then
46:             lab[p] ← WSHED
47:         end if
48:     else if lab[q] = MASK and dist[q] = 0 then
49:         dist[q] ← curdist + 1; fifo_add(q, queue)
50:     end if
51: end for
52: end loop
53: (*zjistí a zpracuje nová minima na úrovni h*)
54: for all p ∈ D with im[p] = h do
55:     dist[p] ← 0 (*obnoví vzdálenost na 0*)
56:     if lab[p] = MASK then (*p je uvnitř nového minima*)
57:         curlab ← curlab + 1; (*vytvoří novou značku*)
58:         fifo_add(p, queue); lab[p] ← curlab
59:         while not fifo_empty(queue) do
60:             q ← fifo_remove(queue)
61:             for all r ∈ N_G(q) do(*kontrola sousedů q*)
62:                 if lab[r] = MASK then
63:                     fifo_add(r, queue); lab[r] ← curlab
64:                 end if
65:             end for
66:         end while
67:     end if
68: end for

```

69: end for

70: (*Konec zaplavování*)

B OBSAH CD A MANUÁL K PROGRAMU

CD obsahuje 3 scénáře. Jeden s mediánovým filtrem, druhý s filtrem s detekcí hran a poslední bez filtrů. Tyto scénáře jsou obsaženy ve složce „scénáře“. V další složce „testovací obrázky“ jsou umístěny obrázky pro testování programu. Ve složce „RapidMiner_Unuk“ je obsažen projekt programu RapidMiner s již vloženými potřebnými moduly. Tento projekt je třeba importovat do vývojového prostředí eclipse a celý projekt spustit jako RapidMinerGUI. Poté již stačí jen importovat do programu jednotlivé scénáře a v prvním modulu zvolit cestu k testovacímu obrazu. Po provedení výpočtu se zobrazí 3 okna. První s bílými watershedy v obraze, druhý s černými a poslední s bílými na černém pozadí.

Obsah CD:

- **scénáře:** Složka obsahující 3 scénáře s použitými filtry.
- **testovací obrázky:** Složka obsahující několik testovacích obrázků, které byly použity k segmentaci v práci.
- **RapidMiner_Unuk:** Složka obsahující projekt pro spuštění programu RapidMiner s příloženými zásuvnými moduly a modulem použitým v práci.
- **třídy:** Složka obsahující všechny použité třídy k vytvoření segmentačního modulu.

Dále je zde elektronická verze této práce a soubor „Návod.docx“ s návodem na použití modulů a zobrazení výsledků.