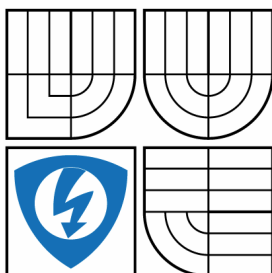


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PROJEKT 3 - POPELKA - TRÍDĚNÍ ČOČKY A RÝŽE

SEMESTRÁLNÍ PRÁCE PŘEDMĚTU POČÍTAČOVÉ VIDĚNÍ
(MPOV)

MASTER'S (BACHELOR'S) THESIS

AUTOR PRÁCE

AUTHOR

Bc. Adam Chromý

Bc. Jiří Hromek

VEDOUCÍ PRÁCE

SUPERVISOR

BRNO 2011

Ing. Pavel Číp

OBSAH

<u>Projekt 3 - Popelka - třídění čočky a rýže.....</u>	<u>1</u>
<u>Zadání projektu</u>	<u>3</u>
<u>Popis problému.....</u>	<u>4</u>
<u>Řešení.....</u>	<u>5</u>
<u>Shrnutí.....</u>	<u>11</u>

ZADÁNÍ PROJEKTU

Je dán obraz s náhodně rozmístěnými plodinami - omezme se pouze na čočku, hrách a rýži. Cílem projektu je nad daným snímkem provést vhodné předzpracování obrazu, separaci jednotlivých plodů a jejich správné zařazení do tříd - hrách, čočka či rýže. Výsledky vhodně reprezentujte - zakreslením okrajové linie plodiny barvou třídy dle schématu do původního obrazu s vyznačeným středem oblasti. Barvy zvolte takto:

- hrách - zelená
- čočka - žlutá
- rýže - modrá
- neurčité - červená

Snažte se o to, aby roztřídění plodin bylo co nejpřesnější a v položce neurčito pokud možno nula prvků. Berte v potaz pouze prvky, které jsou vidět celé a nedotýkají se okraje. Perspektivní a jiné zkruslení neuvažujte.

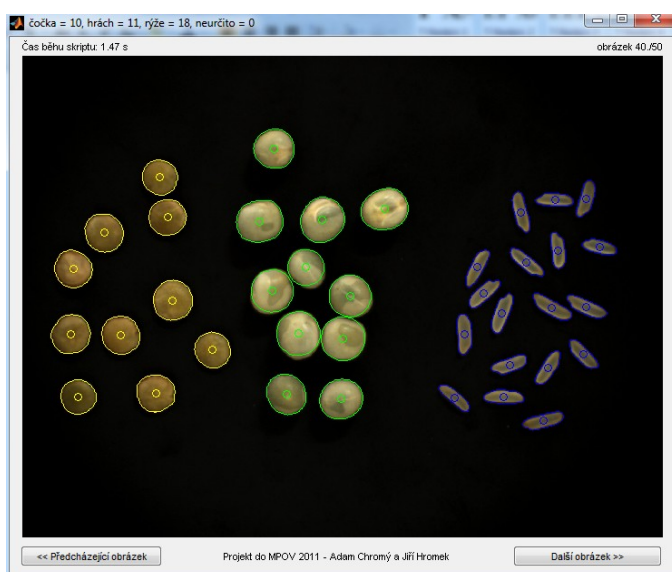
POPIS PROBLÉMU

Jak je zmíněno v zadání, máme k dispozici 50 snímků s náhodně rozmístěnými plodinami. Pro názornost je na obrázku 2.1 příklad vstupního obrazu s pořadovým číslem 40. Na jednotlivých snímcích narůstá s pořadovým číslem složitost vyhodnocení. Jednotlivé plodiny se mohou navzájem překrývat. Po provedení algoritmu je ukázka výsledku reprezentována na obrázku 2.2, kde je patrné zakreslení okrajové linie plodiny barvou třídy dle schématu v zadání do původního obrazu s vyznačeným středem oblasti.

Pro galerii vstupních snímků je shodné pozadí, které má barvu, jejíž jasová složka se pohybuje v hodnotách do 0,09.



Obrázek 2.2.1: Příklad vstupního obrazu

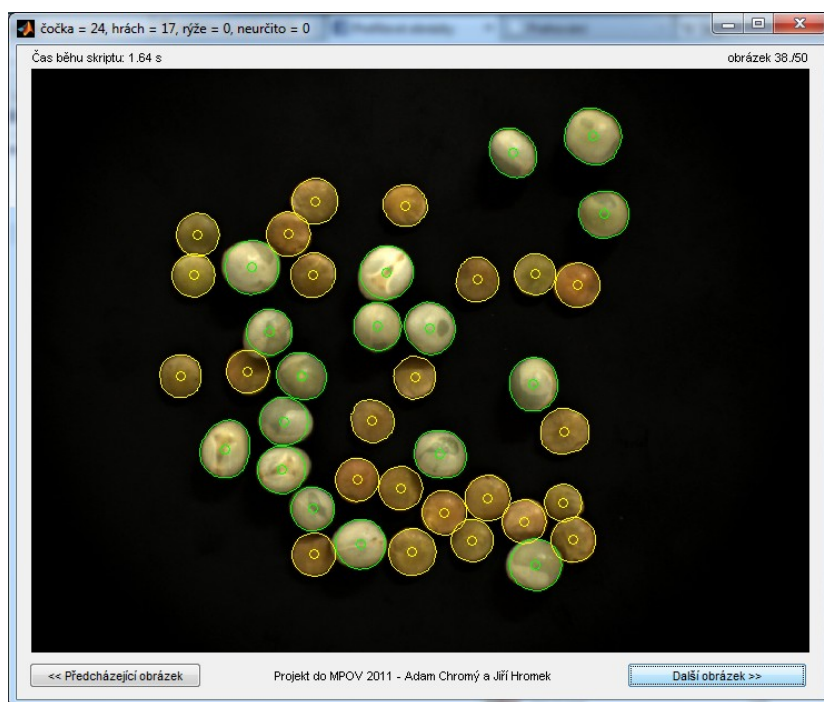


Obrázek 2.2.2: Příklad výstupu – detekce objektů: čočka = 10, hrách = 11, rýže = 18, neurčito = 0

ŘEŠENÍ

1.1 Program projekt.m

Jedná se o m-file, který je vygenerován MATLABem pomocí funkce GUI. Pomocí této funkce a návrhové modulu GUIDE jsme vytvořili jednoduché uživatelské rozhraní pro prohlížení galerie vstupních snímků, na které je vždy po zobrazení aplikován náš algoritmus. Ukázka rozhraní je zobrazena na obrázku 3.1. Jedná se zejména o 2 tlačítka pro posun snímku, v pravém horním rohu je zobrazena hodnota aktuálního snímku. V levém horním rohu je zobrazen čas, který je potřebný pro zpracování aktuálního snímku. V záhlaví je poté vyhodnocen počet jednotlivých plodin.



Obrázek 3.1: Ukázka ovládacího panelu se zobrazením aktuálního snímku společně s vyhodnocenými hodnotami

Pro potřebu správné funkce jednotlivých částí ovládní a zobrazení jsme doplnili některé funkce.

Funkce `performImageShow` je vytvořena pro ovládní výstupního okna. Zde je proveden výpočet času potřebného pro detekci jednotlivých objektů, posun snímku, zajištění stavu, kdy dojde k posunu z posledního snímku zpět na začátek a naopak. Funkce také zajišťuje aplikaci našeho algoritmu, který je uložen v souboru `algoritmus.m`.

Funkce `projekt_OpeningFcn` zajišťuje načtení prvního snímku z databáze, při spuštění programu.

1.2 Popis výstupního algoritmu – algoritmus.m

Tento soubor obsahuje algoritmus, pomocí kterého vyhodnocujeme zadanou úlohu. Zpracování probíhá v několika fázích:

1. **Segmentace samostatných objektů** – tato část algoritmu detekuje a identifikuje objekty, které se nedotýkají a nepřekrývají s jinými.
2. **Identifikace kruhů ve shlucích** – separace spojených kruhových objektů
3. **Segmentace rýže prahováním** – separace elips ve shlucích
4. **Segmentace rýže watershedem** – Separace elips ve shlucích, které nebyly identifikovány předchozí metodou.
5. **Segmentace rýže template-matchingem** - Separace elips ve shlucích, které nebyly identifikovány předchozími metodami.

Algoritmus postupně prochází těmito fázemi. V případě, že již není co detekovat, je algoritmus ukončen.

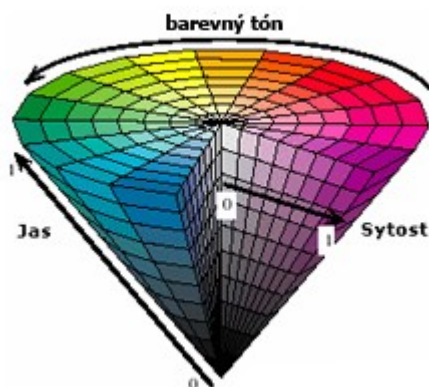
Na počátku určíme proměnné

clumps – seznam objektů co jsou shluky

Counts –matice pro ukládání počtu objektů

Následně provedeme převod do HSV modelu, který je pro zpracování výhodnější. Oproti RGB modelu, který je vhodný pro “míchání” barev, HSV model má tři základní parametry: barevný tón (Hue), sytost (Saturation) a jas (Value).[1]

Tato kombinace je v projektu výhodná vzhledem k velmi blízké hodnotě barvy čičky a hrachu v modelu RGB. Pomocí modelu HSV můžeme tyto barvy snadno odlišit. HSV model je zobrazen na obrázku 3.2.



Obrázek 3.2: Rozložení složek modelu HSV [1]

1.2.1 Identifikace samostatných objektů

V první části provedeme extrakci jasové složky a následně prahování. Tímto postupem odlišíme pozadí od objektů (plodin).

Následuje použití funkce *bwboundaries* - detekce obvodových hran objektu.

U prahování občas vznikají díry uvnitř objektů způsobené odlesky, šumem apod. Díky parametru *noholes* jsou díry (tj. většinou chyby prahování) ignorovány.

Dále užijeme funkce *regionprops* – výpočet parametrů, které jsou potřebné pro klasifikaci tvaru detekované oblasti a zobrazení středu. Zajímá nás zejména parametry:

Solidity – konvexnost, tedy pokud dosahuje konvexnost určité meze, lze rozhodnout, zda se jedná o samostatný objekt nebo shluk

MajorAxisLength – hlavní osa objektu

MinorAxisLength – vedlejší osa objektu

BoundingBox – slouží k vymezení výřezu z obrazu, ve kterém se nachází objekt

Image – binární obraz charakterizující plochu objektu

Centroid – souřadnice středu

Area – celková plocha detekovaného objektu v pixelech.

Abychom mohli dále identifikovat jednotlivé objekty, provedli jsme nejprve statistickou analýzu vybraných parametrů u jednotlivých tříd objektů. Výsledkem je následující tabulka.

Plodina	Rozsah Axis	Rozsah poměru MajorAxis/MinorAxis	Rozsah Area	Rozsah Solidity	Rozsah poměru avgS/avgH	Rozsah MinorAxis	Rozsah MajorAxis
rýže	23 - 92	2,14 - 3,52	1359 - 2318	0,96 - 0,99	-	23 - 35	64 - 92
čočka	52 – 84 *	1 - 1,41	3018 - 5174	0,98 - 1	5,21 - 8,84	-	-
hrách	50 – 102**	1 - 1,64	3077 - 7183	0,98 1	2,35 - 3,86	-	-

Tabulka 3.1: Rozsah prahových hodnot pro detekci objektů

* většina objektů typu „čočka“ má Axis 62 – 84, pouze jeden má hodnotu 52

** většina objektů typu „hrách“ má Axis 66 – 102, pouze dva objekty mají hodnotu kolem 50

Z této tabulky je patrné, že poměr hlavní a vedlejší osy se u rýže pohybuje v intervalu, který nemá žádný průnik s intervalem ostatních plodin. Stejně tak intervaly poměru průměrné S a průměrné H složky objektu pro čočku a hrách jsou disjunktní.

Postupně zpracováváme jednotlivé detekované oblasti a určíme souřadnice jejich souřadnici středu.

Následují 3 podmínky. První podmínka *if* zajišťuje minimální velikost detekovaného objektu. Podle tabulky 3.1 je patrné, že objekty nedosahují velikosti

menší než 1359. V algoritmu jsou zavedeny hodnoty detekovaného objektu: $Area > 1000$; $MajorAxisLength > 10$; $MinorAxisLength > 10$.

Tím zabráníme chybné detekci plodiny v místě, kde se na snímku nachází nečistota, nebo drobná chyba vzniklá při zpracování obrazu v místě pozadí.

Pomocí druhé podmínky `if` zjišťujeme, zda se jedná o samostatný objekt, nebo „shluk“. Zde využíváme proměnné *solidity*. V případě, že můžeme obepnout detekovaný objekt konvexním obalem a ten obepíná objekt s přesností větší než 0,95 jedná se o samostatný objekt. V případě shluku jsou v konvexním obalu přítomny i části, které nejsou součástí objektu. V tom případě klesá hodnota *solidity* a daný objekt lze vyhodnotit jako shluk. Pokud je objekt vyhodnocen jako shluk, je pro další zpracování uložen do seznamu *clumps*.

Ve třetí podmínce zkoumáme, zda daný objekt je rýže. Využijeme poměr stran objektu $MajorAxisLength/MinorAxisLength > 2$. Z tabulky 3.1 je patrné, že v kladném případě hodnota poměru odpovídá elipsoidnímu tvaru, tedy rýži.

Při vyhodnocení objektu jako rýže, vykreslíme okraj modře a do proměnné `Counts(3)` přičteme jedničku.

Pokud objekt neodpovídá podmínkám splňujícím tvar rýže, jedná se o kruhový objekt, tedy jde o hrách nebo čočku. Pro rozhodnutí o jaký druh se jedná, využijeme posloupnosti následujících příkazů. Provedeme výřez obsahující pouze zpracovávaný objekt pomocí funkce `imcrop` a následně maskování okolí objektu v tomto výřezu.

Vypočteme stření hodnotu Hue (barevný tón) a Saturation (sytost) celého objektu. Těmto hodnotám odpovídají proměnné *avgH* a *avgS*.

V následujícím kroku porovnáme podíl *avgH* a *avgS* s hodnotou 4,5. Tato hodnota byla určena na základě hodnot vyplývajících z tabulky 3.1, kde vidíme hraniční rozdíl v tomto podílu právě na hodnotě v okolí bodu 4,5. Při kladném vyhodnocení podmínky zjistíme, že se jedná o čočku, volíme `type = 'y'` (okraj bude žlutý) a do proměnné `Counts(1)` přičteme hodnotu 1. V opačném případě se jedná o hrách a okraj bude zelený. V konečné fázi se nacházejí příkazy pro vykreslení okraje objektu barvou, na základě druhu detekované plodiny a vykreslení středu.

1.2.2 Separace a identifikace objektů ve „shluku“

Pro detekci plodin, které nebyly rozpoznány v první části algoritmu, se věnuje část kódu označená jako *Segmentace dotýkajících a překrývajících se objektů*. Tato část začíná cyklem `for`, pomocí kterého aplikuje následující postup na všechny shluky. Postupně jsou aplikovány různé metody na detekci plodin ve shluku. Tyto metody jsou systematicky rozloženy od těch jednodušších a časově méně náročných, až po ty náročnější, jejich použití na celý obraz by vedlo k neúnosně dlouhé výpočetní době.

Při detekci jednotlivých plodin nejprve vytvoříme výřez obsahující daný objekt (shluk plodin) a vymaskujeme jeho okolí. K dispozici tedy máme pouze detekovaný shluk, se kterým dále pracujeme. Tento obrázek je poté postupně „ořezáván“ o

identifikované objekty, pomocí systematicky prováděných metod. Jakmile nastane situace, že jsou všechny objekty ze shluku separovány, přejdeme na zpracování následujícího shluku, a tím přeskočíme časově náročnější algoritmy. Výsledkem toho je, že doba běhu algoritmu je závislá na složitosti snímku a pohybuje se v rozsahu od 1 do 14 s.

Identifikace kruhových objektů ve shlucích

První metoda je v kódu označena jako *Středně rychlá detekce – detekce kruhových objektů ve shlucích*. Nejprve do proměnné `distmap` vygenerujeme `distmap` celého shluku. Jedná se o tzv. vzdálenostní mapu, kdy má bod hodnotu úměrnou vzdálenosti k nejbližšímu okraji objektu. V krajní pozici je hodnota nižší (tmavší) a ke středu objektu se hodnota zvyšuje. U kruhových objektů nám tedy vzniknou maxima v místě středů a jejich hodnota bude rovna poloměru kružnice. Hledáme tedy maximum v `distmap`, což odpovídá středu. Jeho hodnota nám rovnou udává poloměr, jehož velikost porovnáme s hodnotou 30, která je vybrána na základě velikostí kruhových objektů (hrachu a čočky) z tabulky 3.1. Pokud je hodnota maxima menší, nejedná se o objekt čočky nebo hrachu.

Vytvoříme masku, pomocí které vybereme pouze tento kruh. Při výběru kruhu pomocí masky je nutné zmínit zvýšení poloměru `radius` o 4. Toto je nutná podmínka, protože v opačném případě by mohlo dojít k výběru kruhové oblasti a na místě by zůstal jakýsi prstenec, který by mohl být detekován jako další objekt.

Následuje část vyhodnocení, zda se jedná o hrach nebo čočku. Tento zápis se objevil již v kapitole 3.1.1 popisující detekci samostatných objektů. Pro zakreslení na detekovaného objektu na správnou pozici je vytvořena funkce `drawEllipse`. Následně aplikujeme masku, která vybere vše, kromě tohoto identifikovaného kruhu a aplikujeme metodu znovu. To provádíme dokud se v obrázku vyskytují maxima s hodnotou vyšší než 30.

Po předchozí metodě již v obrázku zbývá pouze rýže. Použijeme námi vytvořenou funkci `detectResidualsShape`. Tato funkce je v programu použita po této i každé následující metodě a provádí eliminaci zbytků objektů na základě plochy (všechny objekty s plochou menší než 1000 jsou odstraněny). Současně identifikuje samostatné objekty typu rýže, které se samy vysegmentovaly odstraněním předchozích objektů.

Segmentace elips přísnějším prahováním

Tato metoda využívá prahování s přísnější hodnotou. Tímto způsobem můžeme detekovat objekty rýže, které se dotýkají pouze minimálně. Přísnější práh je použit až v tomto místě, protože kdybychom jej použili hned na začátku, způsobil by chybnou segmentaci kruhových objektů. Nyní jsou však již všechny kruhové objekty odebrány, takže můžeme přísnější prahování použít. Detekujeme okraje pomocí funkce `bwboundaries` s parametrem `noholes`. `Noholes` je použito z důvodu, že rýže je uvnitř tmavší než na okraji a vznikaly by díry, které jsou tímto způsobem eliminovány. Pokud objekt splňuje podmínku `Solidity > 0,93`, je identifikován jako rýže vykreslíme okraje rýže a střed modře a do položky `Counts(3)`, která odpovídá celkovému počtu rýže, přičteme 1. Následně vymaskujeme objekt z obrázku.

Na konci této metody je opět volána funkce `detectResidualsShape`, která je popsána již v předchozí metodě.

Segmentace elips pomocí watershedu

Segmentace rozvodím (*angl. watershed segmentation*) je metoda založená na názorné představě zaplavování obrazu vodou. Jas pixelu je přitom chápán jako jakási nadmořská výška. Jeden segment je pak vlastně oblastí vyplněnou vodou, hranice segmentů je místem, kde se stýkají dvě zaplavené oblasti oddělené souvislou skupinou pixelů s vyšší hodnotou jasu.[2]

Pro použití této metody vytvoříme zase „distmapu“. Tuto mapu budeme dále filtrovat, protože u rýže nemůže být vzdálenost větší než 15. Takové body jsou jistě dvou dotýkajících se objektů takže je potlačíme. Poté provedeme úpravy pro aplikaci watershedu.

Při postupném „zaplavování“ může nastat detekce mnoha segmentů namísto jednoho požadovaného. Tento jev je znám jako *oversegmentation* a pro jeho potlačení tedy vyhledáme drobná minima pomocí funkce `Irmin`.

Následně provedeme velmi podobnou sekvenci příkazů, která je obsažena ve funkci `detectResidualsShape` ovšem při vyhodnocení, zda se jedná o rýži použijeme přísnějších kritérií. Tedy `MajorAxisLength` je v rozmezí 55 – 95 a `MinorAxisLength` je v rozmezí 19 – 40. Proměnná `Solidity` musí být větší než 0,90. Kritéria jsou přísnější, protože metoda watershed může detekovat i spojené rýže nebo naopak jejich kusy. Na konci metody je opět použita funkce `detectResidualsShape`.

Template matching

Pro aplikaci metody Template matching jsme užili funkce, která je vytvořena D. Kroon University of Twente (February 2011) a je volně použitelná.

Metoda spočívá v porovnávání vzoru s oblastí detekovaných shluků. V našem případě používáme jako vzor obraz rýže, který má průměrné hodnoty co se týče velikosti, tvaru, jasu. Tento obraz je v programu reprezentován proměnnou `T`, do které je vzor uložen. Shodu vzoru s oblastí shluku hledáme v šedotónovém obraze. Porovnávání bude probíhat v jednotlivých polohách, co se týče úhlu natočení obrazu a vzoru. Vytvoříme pole `accum`, které bude sloužit pro ukládání míry shody pro jednotlivá natočení vzoru. V cyklu postupně natáčíme vzor po 10° v intervalu $\langle 0, 180 \rangle$. Pro každé natočení vzoru určíme nejlepší shodu a zapíšeme ji do `accum`. Pomocí funkce `find` poté stačí nalézt maximální hodnotu, tedy hodnotu, kdy má natočení objektu nejlepší shodu. Stejně jako u předchozích funkcí vymaskujeme detekované elipsy z obrázku a pomocí funkce `detectResidualsShape` odstraníme zbytky po ořezu předchozích objektů a zkontrolujeme, zda nezůstávají oddělené rýže.

Nedetekovatelné oblasti

V případě, že shluk nebyl rozdělen na objekty žádnou z předchozích metod, v tomto místě jej obtáhneme červeným okrajem. Aby se opravdu jednalo o objekt, musí splňovat velikost `Area` větší než 1000. Na trénovacích obrázcích k tomu nikdy nedošlo, algoritmus byl vždy schopen rozdělit všechny shluky.

V úplném závěru výstupního algoritmu je výpis textu do záhlaví, který vypíše množství jednotlivých plodin, které jsou detekovány na vstupním obrazu.

SHRNUTÍ

Námi navržený algoritmus dokáže na všech zadaných obrázcích bezchybně nalézt a identifikovat všechny objekty. V žádném ze snímků nenastane případ, že bude vykreslena a započtena neurčitá oblast. Pouze na snímku 46 dochází k určení počtu rýže, kde je počet o jednu rýži nižší. Tato skutečnost je způsobena tím, že metoda Template matching vyhodnotila trojici rýží jako dvojici v důsledku netypické velikosti jedné z rýží vzhledem ke vzoru. Celkově však můžeme konstatovat, že algoritmus pracuje podle představ a zadání.

Díky postupné aplikaci pěti různých metod v pořadí od časově nejméně náročných závisí doba zpracování obrazu na jeho složitosti. V případě obrazů bez překryvu objektů (nejjednodušší případ) dosahuje náš algoritmus výpočetní doby okolo 1s, v nejsložitějších případech se pak doba zpracování blíží ke 14 s. Tento výpočetní čas by bylo nejspíše možné ještě mírně snížit optimalizací kódu, nebo jeho převodem do EXE souboru, ale i v současném stavu považujeme časovou náročnost výpočtu za přiměřenou složitosti problému.

Literatura

- [1] KRIŠTOF, Martin. *Výukový modul do EPO - Digitální fotografie* [online]. 2007 [cit. 2011-12-12]. Barevný model. Dostupné z WWW: <<http://www.dmp.spsei.cz/digi/model.php>>.
- [2] *Wikipedie* [online]. 2011 [cit. 2011-12-12]. Segmentace obrazu. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Segmentace_obrazu>.
- [3] KALOVÁ, Ilona. *Segmentace a detekce geometrických primitiv*. materiály k přednáškám, VUT Brno, 2011.
- [4] KALOVÁ, Ilona. *Regionální segmentace a shlukování*. materiály k přednáškám, VUT Brno, 2011.