

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

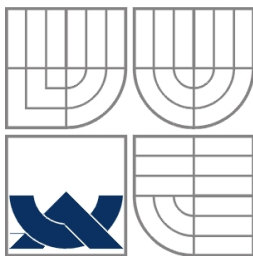
HIERARCHICKÉ METODY SEGMENTACE OBRAZU
HIERARCHICAL IMAGE SEGMENTATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

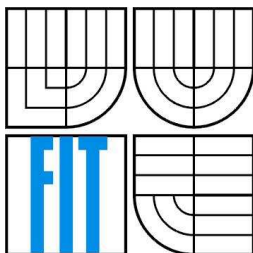
AUTOR PRÁCE
AUTHOR

STANISLAV STANĚK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

HIERARCHICKÉ METODY SEGMENTACE OBRAZU

HIERARCHICAL IMAGE SEGMENTATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

STANISLAV STANĚK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MICHAL ŠPANĚL

BRNO 2008

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Staněk Stanislav**

Obor: Informační technologie

Téma: **Hierarchické metody segmentace obrazu**

Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte základy zpracování obrazu. Zaměřte se zejména na problematiku reprezentace a segmentace obrazu.
2. Zorientujte se v současných metodách hierararchické segmentace obrazu.
3. Vyberte vhodné metody a experimentálně je implementujte.
4. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
5. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje. Zvažte další pokračování v rámci diplomové práce.
6. Vytvořte stručný plakát prezentující vaši bakalářskou práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

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

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

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

Vedoucí: **Španěl Michal, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Stanislav Staněk**
Id studenta: 84143
Bytem: Bratří Hlavců 92, 755 01 Vsetín
Narozen: 06. 10. 1984, Olomouc
(dále jen "autor")

a

2. Vysoké učení technické v Brně

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

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

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

Název VŠKP: Hierarchické metody segmentace obrazu
Vedoucí/školitel VŠKP: Španěl Michal, Ing.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

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

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

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnožení.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

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

V Brně dne:

.....
Nabyvatel

.....
Autor

Abstrakt

V mnoha aplikacích počítačového vidění je segmentace obrazu jedním z nekritičtějších kroků jeho analýzy. V této práci je prezentována segmentační metoda založená na shlukovacích algoritmech fuzzy c-means a k-means. Je použita hierarchická datová struktura společně se shlukovacími algoritmy pro segmentaci na každém stupni pyramidy. Výsledky ukazují, že výpočetní čas je mnohem menší, než v případě klasického shlukování.

Klíčová slova

Zpracování obrazu, segmentace, shlukování, k-means, fuzzy c-means, Gausova pyramida.

Abstract

In many vision applications image segmentation is one of the most critical steps of analysis, which has the objective of extracting information from an image. In this work a segmentation method based upon fuzzy c-means and k-means clustering is presented. A hierarchical data structure together with clustering algorithms for the segmentation in each level of the pyramid is used. The results show that the computation time is much less than that of a classical clustering.

Keywords

Image processing, segmentation, clustering, k-means, fuzzy c-means, Gaussian pyramid.

Citace

Staněk Stanislav: Hierarchické metody segmentace obrazu. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Hierarchické metody segmentace obrazu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením...

Další informace mi poskytli...

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
Datum

Poděkování

Děkuji vedoucímu své bakalářské práce za cenné rady, uvedení do problematiky a odborné vedení při zpracování této práce.

© Stanislav Staněk, 2008.

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

Obsah

Obsah	1
Úvod	2
1 Zpracování obrazu	3
1.1 Základní pojmy	4
1.1.1 Konvoluce	4
1.1.2 Šum a filtrace	4
1.2 Reprezentace obrazu	5
1.2.1 Prostorová oblast	5
1.2.2 Popis barev v obraze (barevné prostory)	5
2 Segmentace obrazu	7
2.1 Definice segmentace	7
2.2 Segmentační metody	8
2.2.1 Prahování a shlukování	8
2.2.2 Detekce hran	8
2.2.3 Detekce oblastí	9
2.3 Hierarchická segmentace	9
2.3.1 Hierarchická reprezentace dat	10
2.3.2 Hierarchické metody	11
3 Shlukování	13
3.1 K-means (metoda nejbližších středů)	13
3.2 Fuzzy c –means	14
4 Návrh algoritmu	16
5 Implementace	18
5.1 OpenCv	18
5.2 Vlastní implementace	18
6 Výsledky testování	20
6.1 Testování algoritmu k-means	20
6.2 Testování algoritmu fuzzy c-means	21
6.3 Testování v prostoru CILAB	21
6.4 Porovnání algoritmů	22
7 Závěr	23
8 Literatura	24

Úvod

Obraz, ve svém všeobecném smyslu může zahrnovat vše, co si dokážeme představit. Od obrázků, videa, animací, grafů, kreseb, až po samotný text. Z obrazu přijímáme největší množství informace z okolí, které nás obklopuje. Básník Ezra Pound jej kdysi definoval jako „vyjádření intelektuálního a emocionálního komplexu v časovém okamžiku“. Ve snaze lépe pochopit data společně s tím, co představují, byly proto vyvinuty rozdílné techniky. Ty nabyly na významu v mnoha oblastech jako je biometrie, komprese, rozpoznávání vzorů, nebo biomedické inženýrství. Všechny tyto techniky dnes můžeme zahrnout do tří oblastí: zpracování obrazu, obrazová analýza a rozpoznávání obrazu.

Smyslem této práce je představit jednu z nejkritičtějších částí analýzy obrazu, jíž je segmentace obrazu a především její hierarchický přístup.

V úvodu této práce se seznámíme s problematikou zpracování obrazu a jeho reprezentací. Popíšeme si reprezentaci obrazu v prostorové oblasti a následně se seznámíme s barevnými prostory použitými v této práci. Ve druhé kapitole se budeme hlouběji zabývat segmentací obrazu. Uvedeme si formální definici segmentace a možnou klasifikaci jednotlivých segmentačních metod. Kromě klasických segmentačních algoritmů nutných k pochopení jednotlivých segmentačních přístupů si představíme i některé hierarchické metody segmentace obrazu. Ve třetí kapitole se ponoříme do problematiky shlukové analýzy. Popíšeme si shlukovací algoritmy k-means a fuzzy c-means. U posledně jmenovaného si navíc uvedeme metodu pro určení optimálního počtu shluků. Čtvrtá kapitola se bude zabývat návrhem algoritmu použitého v této práci. V další části práce se dostaneme k implementaci výsledného programu. Uvedeme si knihovnu OpenCv a popíšeme si jednotlivé třídy a způsob jakým byly navrženy. V předposlední, šesté kapitole si v přehledné formě uvedeme výsledky testování. Zaměříme se především na celkovou časovou náročnost a rozdíly v segmentacích mezi klasickým a hierarchickým přístupem. Srovnáme taky výsledky mezi jednotlivými shlukovacími algoritmy. Zaměříme se i na výsledky segmentace v jednotlivých barevných prostorech. Na závěr shrneme celkový přínos práce a poznatky získané při jejím vypracování.

1 Zpracování obrazu

Cílem zpracování a rozpoznávání obrazu je porozumění obsahu obrazové informace reálného světa, která do počítače vstupuje nejčastěji prostřednictvím kamery nebo skeneru. Postup zpracování obrazu můžeme obvykle rozdělit do několika základních kroků.

1. Snímání a digitalizace
2. Předzpracování
3. Segmentace obrazu
4. Popis objektů
5. Klasifikace

Prvním krokem ve zpracování obrazu je snímání obrazu pomocí senzorů např. optických. Dvou-dimenzionální signál získaný těmito senzory je převeden na elektrický signál spojitý v čase i úrovni. Vstupní informací může být například jas, intenzita červené barvy. Nejčastěji je to ale více hodnot (červená, zelená a modrá složka) při barevném snímání, nebo dokonce celé spektrum barev (např. data z počítačového tomografu, či spektrální měření svitu hvězd).

Dalším krokem je převod spojitého analogovaného signálu na signál diskrétní. Tento proces se nazývá digitalizace a odehrává se ve dvou nezávislých krocích, jimiž jsou kvantování a vzorkování.

Vzorkování (sampling) spojitě funkce je zaznamenání hodnot – vzorků v předem daných intervalech. S kratším vzorkovacím intervalem zaznamenáme více detailů z původní spojitě funkce a získáme obraz ve vyšším rozlišení. S delším vzorkovacím intervalem naopak informace o detailech v obraze ztrácíme. Kvantování probíhá v oboru hodnot obrazové funkce, který se rozdělí na intervaly, jimž je pak přidělena jediná zástupná hodnota. Výsledkem digitalizace je obraz v digitální podobě reprezentovaný maticí obrazových bodů – pixelů.

Po digitalizaci následuje předzpracování obrazu. Někdy je obraz zašumělý a degradovaný vlivem optického systému, špatného zaostření, nebo mlhy. Cílem předzpracování je filtrací, ostřením, jasovými a geometrickými transformacemi vylepšit obraz.

Třetím, nejtěžším krokem zpracování obrazu je segmentace. Jedná se o analýzu obrazu vedoucí k nalezení objektů zájmu v obraze. Ty jsou předány k dalšímu zpracování, jako je popis objektů, nebo klasifikace.

Popis objektů má dva způsoby. Na základě souboru číselných charakteristik – kvantitativně, nebo relacemi mezi objekty a jejich tvarovými vlastnostmi – kvalitativně. Ve většině případů je vstupní informací pro následnou klasifikaci.

Klasifikace je zařazení objektů nalezených v obraze do skupiny předem známých tříd. Jedná se o finální krok zpracování a rozpoznávání obrazu.

1.1 Základní pojmy

K nezbytnému pochopení celé problematiky segmentace si nyní uvedeme některé pojmy o které se tento proces opírá.

1.1.1 Konvoluce

Konvoluce je matematický operátor zpracovávající dvě funkce, který se často používá v algoritmech pro zpracování dvourozměrného diskrétního obrazu. Vzorec diskrétní konvoluce má tvar:

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

Funkci $h(i, j)$ se říká konvoluční jádro, které lze chápat jako tabulku (konvoluční masku) položenou na příslušné místo v obraze. Koefficienty uvnitř konvoluční masky udávají vliv hodnoty pixelu pod nimi. Každý pixel překrytý tabulkou vynásobíme koeficientem v příslušné buňce a provedeme součet všech hodnot. Výsledkem je pak nová hodnota pro pixel pod středem konvoluční masky.

1.1.2 Šum a filtrace

Šum můžeme obvykle definovat jako nežádoucí informaci, která byla do obrazu přidána v průběhu jeho zpracování. Často je znatelný jako různobarevné zrnění představované vysokými frekvencemi, které mohou značně ovlivnit celý proces segmentace. K jeho odstranění proto používáme filtraci. Ta kromě potlačení šumu a vyhlazování může být využita i pro zvýraznění kontrastu, nebo detekci hran. Filtrace pracuje s intenzitou bodu vázaného na své okolí. K jejímu provedení je tak často používána již zmiňovaná konvoluce.

1.2 Reprezentace obrazu

Informace obsažené v obraze můžeme reprezentovat zcela rozdílnými způsoby. Záleží především na způsobu jakým na výsledný obraz nahlížíme a charakteristikách, které nás zajímají. My se zde zaměříme na reprezentaci obrazu v prostorové oblasti a způsobu vyjádření hodnot, jichž nabývá.

1.2.1 Prostorová oblast

Při práci s obrazem v počítačové grafice máme jen zřídka k dispozici spojité obraz. Důvodem, který tomu zabráňuje je paměťové omezení dnešních počítačů. Protože počítače nedokážou obsloužit spojité signál, ale jen data v podobě sekvence digitálních hodnot, vznikla potřeba vyjádřit obraz jako dvou-dimenzionální pole bodů. Nebo-li, jako diskretní matici pixelů. Pixel pak představuje jeden svítící bod na příslušné pozici matice. V nejjednodušším případě je pixel umístěn na čtvercové mřížce bodů zvané "rastr". Pozice pixelu je dána běžnou notací pro matice. První index, m , určuje pozici řádku. Druhý index, n , určuje pozici sloupce.

Pokud se tedy obraz skládá z matice typu $M \times N$, pak M je počet řádků a N je počet sloupců matice.

Kromě reprezentace v podobě čtvercové mřížky, můžeme ve dvou-rozměrném prostoru použít další geometrická uspořádání a tvary v podobě hexagonální a trojúhelníkové mřížky.

S rostoucím výkonem počítačů se však předmětem zájmu staly i obrazy v dimenzích daleko vyšších např. v prostoru 3D je pixel přeměněn na objemový element s názvem voxel.

1.2.2 Popis barev v obraze (barevné prostory)

Existuje velké množství různých barevných prostorů. My si zde uvedeme prostory RGB a CIELAB použité v této práci.

1.2.2.1 RGB barevný prostor

Různé barvy, které se používají při vytváření obrazu, jsou tvořeny kombinací několika základních složek. Barevný prostor určuje ze kterých složek a jakým způsobem se výsledný obraz skládá. Na barevné obrazovce například vidíme barvu jako výsledek složení tří složek – červené (R, red), zelené (G, green), modré (B, blue). Každá výsledná barva je tvořena vektorem tří hodnot udávajících intenzitu jednotlivých složek. Ty mohou nabývat hodnot z intervalu $\langle 0, 1 \rangle$. Bývají však uváděny i v celočíselném rozsahu 0 – 255, což odpovídá kódování každé ze složek RGB v jednom bytu. Hodnota 0 znamená, že složka není zastoupena. Maximální hodnota 255 indikuje, že složka nabývá své největší intenzity. Barva je tak určena třemi bajty a počet zobrazitelných barev je $256^3 = 16\,777\,216$, někdy také značené jako true color.

Složíme-li červenou, zelenou a modrou barvu ze složek o maximální intenzitě, získáme barvu bílou. Složením nulových intenzit, pak barvu černou. Podobně ze složek o stejné intenzitě získáme postupně šedi. Kdybychom však z různých důvodů chtěli převést barevný obraz na černobílý, nemůžeme jeho barvy nahradit odstíny šedi získanými prostým průměrem ze tří základních barev. Lidské oko vnímá různým způsobem intenzitu jednotlivých barevných složek. Pro výpočet jasu v černobílém obraze se proto používá následující empirický vztah.

$$I = 0.299 * R + 0.587 * G + 0.114 * B \quad (1.1)$$

Barevný rozsah prostoru RGB můžeme prostorově zobrazit jako jednotkovou krychli umístěnou v osách označených postupně r, g, b (obr. 1.1). Počátek souřadnic odpovídá černé barvě, zatímco vrchol o souřadnicích [1, 1, 1] odpovídá bílé. Vrcholy krychle, které leží na osách, představují základní barvy a zbývající vrcholy reprezentují doplňkové barvy ke každé ze základních barev.

Prostor RGB je jednoduchý na implementaci, ale je nelineární se smyslovým vnímáním barev člověkem. Přesto je velmi běžný a používaný skoro ve všech zařízeních jako je počítač, televize a video.

1.2.2.2 CIELAB barevný prostor

Prostor CIELAB je tří dimenzionální prostor, který je charakterizován třemi navzájem kolmými osami L^* , a^* , b^* . Osa L^* vyjadřuje jak je barva světlá. Souřadnice a^* , b^* definují barevný odstín. Osa a^* reprezentuje červeno-zelené odstíny, osa b^* žluto-modré. Charakterizace barev je tak podobná způsobu zpracování barevné informace okem. Ačkoliv jsou v oku tři druhy čípků citlivé ke krátkým, středním a dlouhým vlnovým délkám, nejsou do mozku přenášeny přímo signály R, G, B. V gangliových buňkách dochází k jejich zakódování do signálu červeno-zeleného, žluto-modrého a achromatického (jasového).

Často je označován přívlástkem uniformní, který vyjadřuje, že vzdálenost mezi dvěma body v prostoru CIELAB lépe odpovídá rozdílu intenzit mezi barevnými odstíny tak, jak je vnímá lidský vizuální systém. Rozdíly barev, které jsou člověkem vnímány jako stejně velké, jsou tak i při číselném vyjádření stejně velké. Tato vlastnost prostoru CIELAB ho tak činí obzvláště atraktivním v oblasti analýzy barev a shlukování.

2 Segmentace obrazu

Rostoucí potřeba po automatické analýze obrazu a jeho interpretaci v širokém spektru aplikací si vynutila vzniku řadě segmentačních algoritmů. Segmentace obrazu je často popisována jako proces, který podle vybraných vlastností rozdělí obraz do homogenních a smysluplných částí odpovídajících předmětům či oblastem reálného světa. Jestliže tyto části odpovídají objektům ve vstupním obraze, pak hovoříme o tzv. kompletní segmentaci. Pokud ale oblasti neodpovídají přesně objektům ve vstupním obraze, pak tuto segmentaci nazýváme částečnou. Kompletní segmentace obecně využívá vyšší úroveň zpracování, která je založena na znalostech řešeného problému. Částečná segmentace je založena na principu homogenity obrazových vlastností (např. jas, barva, textura) uvnitř segmentu.

2.1 Definice segmentace

Aby nedocházelo k záměně mezi částečnou a kompletní segmentací, uvedeme formální pravidla kompletní segmentace:

Kompletní segmentace obrazu R v sobě zahrnuje identifikaci konečného počtu neprázdných regionů R_i , $i = 1, 2, \dots, N$ takových, že:

1. $R = R_1 \cup R_2 \cup \dots \cup R_N$.
2. $R_i \cap R_j = \emptyset, \forall i \neq j$.
3. $P(R_i) = TRUE, \forall i$.
4. $P(R_i \cup R_j) = FALSE, \forall i \neq j$.

Kde $P(R_i)$ je predikát uniformity a homogenity pro všechny elementy R_i a kde \emptyset značí prázdnou množinu. Z výše uvedených podmínek tedy popořadě vyplývá:

1. Každý pixel obrazu by měl patřit do nějakého regionu.
2. Regiony by se neměly překrývat.
3. Pixely by měly být ve svém regionu přiřazeny podle určitých společných vlastností.
4. Pixely, které patří do různých regionů by měly mít rozdílné vlastnosti.

V některých textech zabývajících se segmentací obrazu se dále uvádí, aby hranice a vnitřek segmentů byly jednoduché s malým počtem děr uvnitř regionů.

2.2 Segmentační metody

S existencí velkého množství segmentačních metod vznikly různé typy klasifikací segmentačních algoritmů podle způsobu, jakým dosahují výsledné segmentace. Pohled na počet a typy jednotlivých skupin se v literatuře různě liší. Obecně však můžeme segmentační algoritmy rozdělit do tří základních skupin:

1. Prahování a shlukování.
2. Detekce hran
3. Detekce oblastí.

2.2.1 Prahování a shlukování

Prahování je jedna z nejjednodušších segmentačních metod, zahrnujících identifikaci skupiny hodnot, kterým říkáme práh. V nekomplikované podobě prahování spočívá v rozdělení jasové škály obrazu na dvě části a nahrazení každé z nich zástupnou hodnotou, kde hodnota 1 určuje nalezené objekty a hodnota 0 určuje pozadí obrazu. Výsledkem prahování je pak binární forma obrazu. Tato technika je však zastaralá a selhává i u velmi jednoduchých obrazů z důvodů nestejnomyšerného osvětlení vstupního obrazu. Proto se nejčastěji používá prahování adaptivní s proměnným počtem prahů, ve kterém je obraz rozdělen na několik částí, kde v každé části se hodnota prahu určí nezávisle na ostatních částech obrazu. Pokud v některé části obrazu hodnotu prahu nelze určit, můžeme práh v této části přibližně vyjádřit z hodnot prahů v jejím okolí. Dalším z možných přístupů je prahování na základě intervalu hodnot, kdy hodnoty z intervalu určují objekty a hodnoty mimo interval pozadí obrazu. Důležitým faktorem je také volba správného prahu. Ve složitějších metodách se nejčastěji detekuje vhodný práh z histogramu analýzou jeho tvaru, kde práh je určen jako minimum mezi dvěma maximy v histogramu.

Clustering (shluková analýza) patří mezi metody učení bez učitele. Jejím cílem je v dané množině objektů nalézt její podmnožiny, nebo-li shluky objektů tak, aby si členové shluku byli navzájem podobní a nebyli si příliš podobní od objektů mimo tento shluk. Shlukovou analýzou se budeme podrobněji zabývat v kapitole 3.

2.2.2 Detekce hran

Metody detekce hran jsou založené na náhlé lokální změně stupně intenzity v obraze, která se měří pomocí gradientu. Ten vypočítá jak změnu intenzit v obraze, tak i směr, ve kterém ke změně došlo. Protože obraz $f(x,y)$ je dvoudimenzionální funkce, jejím gradientem je vektor. Velikost vektoru pak určuje sílu hrany v obraze a úhel vektoru určuje úhel gradientu.

Alternativním přístupem pro výpočet hranových gradientů je konvoluce obrazu se skupinou vzorů hran, kde každý vzor odpovídá hranám v určitém směru. Nejčastěji je aplikován při použití hranových operátorů (Robertsův, Sobelův, Cannyho, Laplaceův atd). Ideální obrazový detektor by měl správně označit body, které patří hranám a měl by předcházet chybnému označení bodů v místech, kde hrany neexistují. Kritériem, které rozhodne o existenci hrany v obraze je tak prahová hodnota. Při velikosti gradientu větším než prahová hodnota pak odvodíme, že v daném bodě se vyskytuje bod hrany, jinak se v daném bodě hrana nevyskytuje.

Výstupní obraz detekce hran však často nemůžeme použít jako konečný výsledek segmentace. Musí následovat doplňkové kroky, které spojí jednotlivé hrany do řetězců tak, aby v obraze zůstaly jen hrany korespondující s existujícími objekty v obraze.

Proto se většinou předzpracovává obraz filtrací, ve které se odstraňují vysoké frekvence obsahující šum.

2.2.3 Detekce oblastí

Detekce oblastí se zaměřuje na vyhledávání celých oblastí v obraze na základě zvoleného kritéria homogenity oblasti, jako je konstantní jas, barva, pokrytí stejnou texturou atd. Obecně je tato technika oproti detekci hran výhodná u zašumělých obrazů, ve kterých je obtížné detekovat hranici objektu.

Nevýhodou však může být stav, kdy v určitých případech může být obraz rozdělen do více regionů a dojde tak k přesegmentování obrazu. Někdy však může nastat i situace opačná, kdy je naopak ve výsledném obraze malý počet regionů a obraz je tak podsegmentován. Většinou je detekce oblastí založena na postupném spojování či dělení jednotlivých regionů a kombinaci obou těchto přístupů. Jeden z těchto přístupů nazvaný split and merge si popíšeme v následující kapitole.

2.3 Hierarchická segmentace

Rozdílné postupy segmentace obrazu vyžadují různý stupeň detailu segmentace. Přítomnost šumu v obraze může totiž výslednou segmentaci značně ovlivnit. Kromě způsobu, jakým segmentaci provádíme, navíc záleží i na rozlišení zpracovávaného obrazu a počtu sledovaných vlastností, např. započtených barevných kanálů. S jejich větším množstvím totiž roste celková doba výsledného zpracování dat. Do segmentace obrazu tak byly zavedeny prvky hierarchie, které se snaží tyto záporné vlastnosti potlačit.

2.3.1 Hierarchická reprezentace dat

Je mnoho způsobů jakými lze dosáhnout hierarchie v reprezentaci dat. My si zde představíme dva z nich: obrazové pyramidy a stromy.

Obrazové pyramidy:

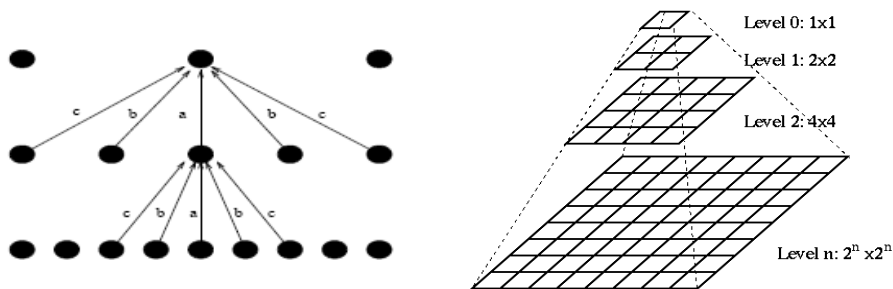
Pyramidou rozumíme hierarchii po sobě následujících obrazů o nižším rozlišení postavenou na originálním obraze. Základnou I_0 pyramidy je obraz v původním nejvyšším rozlišení. Obraz na dalším stupni I_{i+1} je vytvořen z obrazu I_i pomocí následujících dvou kroků:

1. Rozmazání obrazu pomocí filtru s dolní propustí.
2. Podvzorkování obrazu, které zmenší velikost obrazu na polovinu

Vrchol pyramidy pak tvoří obraz s nejnižším rozlišením. Protože pyramida roste směrem dolů od nejvyššího po nejnižší rozlišení, je výsledná pyramida převrácená. Pokud při vytváření pyramidy použijeme jako filtr gaussovo jádro

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right), \sigma = \text{standardní odchylka}$$

nazýváme pyramidu Gaussovou. Jádro Gaussova filtru hodnotí pixely v jeho centru mnohem silněji než v jeho okolí. Ohodnocení však závisí na velikosti standardní odchylky. Při její velmi nízké hodnotě bude mít filtr nepatrný vliv na celkový obraz, protože váhy pro všechny pixely mimo jádro jsou nízké. Naopak s nepřiměřeně velkou standardní odchylkou můžeme v obraze kromě šumu navíc přijít o některé detaily. Gaussova pyramida tak tvoří obrazy v několika stupních rozlišení, které jsou na jednotlivých vrstvách pyramidy rozmazané. Díky jeho vlastnostem jsou obrazy postupně na nižších vrstvách jeho opětovnou aplikací rovnoměrně rozmazány, kdy množství detailů v obraze klesá postupným průchodem pyramidou směrem k obrazům v nižším rozlišení.

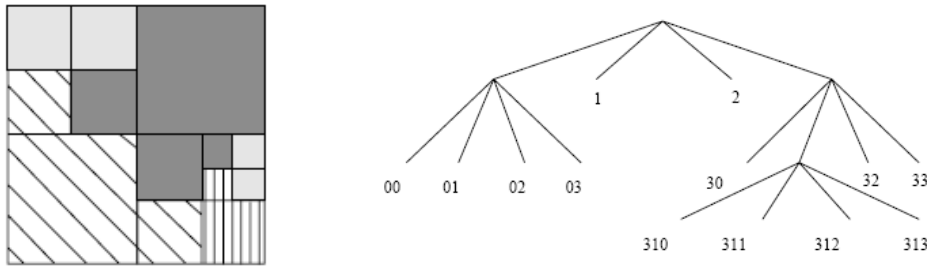


Obrázek 2.1. Tvorba nižší vrstvy a reprezentace výsledné pyramidy

Quadtree:

Quadtree využívá model „rozděl a panuj“ tak, že dělí obraz do čtyř stejně velkých kvadrantů. Heterogenní kvadranty jsou opět děleny do čtyř kvadrantů až do doby, kdy kvadrantu odpovídá homogenní oblast. Výsledný strom je tak stupeň 4, kde kořenem stromu je celý obraz. Uzly reprezentují heterogenní oblasti a listy oblasti homogenní.

Struktura Quadtree je používána pro kompresy dat, efektivní vyhledávání a jiné aplikace ve kterých je užitečné přerušit zpracování homogenních regionů.



Obrázek 2.2. Ukázka struktury Quadtree a její výsledné stromové reprezentace

2.3.2 Hierarchické metody

Pro lepší pochopení využití hierarchické reprezentace dat si popíšeme možné způsoby jejího použití v jednotlivých typech segmentačních metod tak, jak byly klasifikovány v kapitole 2.

Prahování

Prahování v hierarchické struktuře dat je většinou založeno na lokálních metodách. Namísto jedné pevné prahové hodnoty pro celý obraz tak dovolíme, aby se tato hodnota v průběhu zpracování lišila. Cílem je najít v obraze o nižším rozlišení regiony, které budou následně doladěny segmentací obrazu ve vyšším rozlišení. Algoritmus začíná na vrcholu pyramidy, kde se nachází obraz s nejnižším rozlišením, na který je aplikována segmentace. Na dalším stupni je pak segmentace upřesněna. Pixely, které jsou blízko hranicím objektů jsou rozsegmentovány a označeny jako body objektů, nebo pozadí. Toto upřesnění je pak opakováno na každém stupni pyramidy až k obrazu v originální rozlišení. Velkou výhodou tohoto postupu je menší dopad šumu v obraze na výslednou segmentaci. Segmentace v nižším rozlišení je totiž prováděna na rozmazaném obraze, ve kterém je šum potlačen. Nepřesné hranice objektů, vzniklé rozmazáním obrazu jsou pak opětovnou segmentací obrazu na dalším stupni pyramidy opraveny.

Detekce oblastí

Mezi nejjednodušší algoritmy hierarchické detekce oblastí můžeme zahrnout algoritmus Split and Merge. V tomto algoritmu jsou regiony na základě předem daného kritéria homogenity postupně děleny a spojovány. Na začátku je celý obraz označen jako jeden region. Jestliže tento region není

homogenní, tak je obraz rozdělen na čtyři podregiony. Pokud se na některém stupni pyramidy nachází čtyři regiony s podobnými vlastnostmi, jsou následně spojeny do jednoho celistvého regionu v nadřazeném stupni pyramidy. Algoritmus pokračuje až do doby, kdy již není co spojit, nebo rozdělit. Segmentaci tak můžeme chápat jako konstrukci struktury Quadtree, kde každý list reprezentuje homogenní region.

Detekce hran

Většina detektorů hran vymezuje prostor, na kterém jsou jednotlivé hrany detekovány. Ten je omezen velikostí konvolučního jádra a poskytuje tak jen lokální informace v obraze. S větším konvolučním jádrem navíc roste výpočetní náročnost jednotlivých algoritmů. Detekce hran pomocí obrazu v různých stupních rozlišení je tak často používaným procesem zpřesnění výsledku hranových detektorů. Jedním z možných přístupů je nalezení správného stupně pyramidy, na kterém je úspěšně potlačen šum, ale hlavní hrany obrazu jsou zachovány. U každého pixelu je pomocí prahové hodnoty detekováno s jak velkou mírou vyjadřuje hranu na příslušném stupni pyramidy. Pokud je pixel označen jako hranový, opakujeme stejný postup na jeho čtyři následníky. Algoritmus je ukončen při nalezení detailních hran bez přítomnosti šumu.

3 Shlukování

Shlukování, nebo-li shluková analýza je společný název pro celou řadu metod, jejichž cílem je využití analýzy vícerozměrných dat k rozřídění množiny objektů do tříd (shluků). Tyto metody jsou úspěšné především v situacích, kdy objekty mají tendenci se přirozeně seskupovat. Obecně jsou známé dva přístupy shlukování dat. První je založen na principu dělení, kdy data v obraze představují jeden velký shluk. Ten je rekurzivně dělen tak, aby se dosáhlo dobrého shlukování. Ve druhém případě je naopak každý datový prvek vyjádřen jedním shlukem, které jsou postupně spojovány. Kvalita s jakou jsou jednotlivé shluky vytvořeny závisí ve značné míře na vstupních datech. Často se ve shlukovacích algoritmech musí zadat množství cílových segmentačních regionů. To ale většinou neznáme předem. Proto byly vytvořeny různé algoritmy, které se snaží určit množství výsledných shluků, jako např. adaptivní shlukování. Většinou je však stále v těchto metodách dána určitá prahová hodnota, podle které se nakonec obraz rozsegmentuje. Analýza optimálního množství shluku se tak v procesu shlukování stala velmi komplexní záležitostí, kde výsledné množství shluků nakonec s velkou mírou závisí na vstupních datech a vlastnostech obrazu, které pozorujeme. My si zde představíme dva shlukovací algoritmy k-means a fuzzy c-means, které jsou představiteli skupiny těchto shlukovacích algoritmů.

3.1 K-means (metoda nejbližších středů)

Princip metody K-means spočívá v rozdělení n objektů o m proměných do k shluků tak, že mezishluková suma čtverců je přitom minimalizována. Metoda se jeví užitečnou především při vytváření malého počtu shluků z velkého počtu objektů. Vyžaduje spojitě proměnné, a především bez odlehlých hodnot. Diskrétní data mohou být rovněž analyzována, ale mohou způsobit problémy. Algoritmus nalezne spíše optimum lokální než globální. Je to takové uspořádání shluků, kdy již přemístění objektu z jednoho shluku do druhého nezpůsobí snížení sumy čtverců. Algoritmus pracuje opakovaně, startuje vždy u jiného počátečního uspořádání. Nakonec vybere optimální řešení ze všech možných dosažených uspořádání shluků.

Uživatel zadává počet shluků, jež mají být nalezeny. Pak jsou vytvořeny prostorové shluky nalezením souboru středů shluků tak, že každý objekt je přiřazen do jednoho shluku, načež jsou určeny nové shluky a celý proces se opakuje.

V této práci jsem použil verzi algoritmu, která začíná inicializací center. Celý algoritmus můžeme popsat následovně:

1. Inicializuj matici středů shluků V použitím generátoru náhodných čísel. Nastav iterační krok $k = 0$.
 2. Vypočítej matici příslušnosti U
 3. Aktualizuj matici center V
 4. Pokud $\|V^k - V^{k-1}\| < \varepsilon$, ukonči algoritmus. Jinak nastav $k = k+1$ a pokračuj od bodu 2.
- za podmínek:

$\| * \|$ určuje normu. V algoritmu je normou $\max\{(j_1-j_2)^2 + (k_1-k_2)^2\}$ bodů (j,k) .

3.2 Fuzzy c –means

Základní a patrně nejpoužívanější shlukovací metodou je fuzzy c-means. Tato metoda umožňuje shlukování do více než jednoho shluku, zatímco v běžném shlukování je každý objekt členem pouze jednoho shluku. Tím je lépe charakterizováno rozložení objektů a snadno se dají určit objekty ležící na hranicích různých shluků. V případě, že u každého objektu je jedna z hodnot účasti do shluku rovna jedné a ostatní jsou nulové, přechází výsledek na pevné shlukování. Naopak, pokud u všech objektů je hodnota příslušnosti k libovolnému shluku rovna převrácené hodnotě počtu shluků, jsou jednotlivé shluky neurčené. Tento algoritmus může být formulován následovně:

$$\text{Minimalizuj } J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m d^2(x_k, v_i)$$

za platnosti podmínek:

$$\sum_{j=1}^c u_{ik} = 1, \forall i$$

$$u_{ik} \in [0,1] \text{ pro } \forall i, j \text{ a } 1 \leq j \leq c.$$

kde n je počet vstupních dat a c je počet center shluků. $X = \{x_1, x_2, \dots, x_n\} \subseteq R^p$ je matice vstupních dat. U je matice příslušnosti obsahující hodnotu příslušnosti u_{ik} objektu x_k v i -tém shluku a V je matice středů shluků.

Váhový exponent ($m > 0$) určuje míru překrývání jednotlivých shluků. $d^2(x_k, v_i)$ udává vzdálenost mezi objektem x_k a centrem shluku v_i , kterou můžeme získat pomocí vzorce:

$$v_i^{(b)} = \frac{\sum_{k=1}^n (u_{ik}^{(b)})^m x_k}{\sum_{k=1}^n (u_{ik}^{(b)})^m}$$

Matici příslušnosti U vypočítáme následovně:

$$u_{ik}^{(b+1)} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}}$$

Dále musí platit $d_{ik} = \|x_i - v_i\| > 0, \forall i, k$. Necht' $d_{ik} = 0$, potom $u_{ik} = 1$ a $u_{jk} = 0, \forall j \neq i$

Kostra algoritmu použitého v této práci:

1. Inicializuj matici středů shluků V použitím generátoru náhodných čísel. Nastav iterační krok $b = 0$ a fuzzy koeficient m na hodnotu 2 ;
2. Vypočítej matici příslušnosti U podle vzorce (3.3).
3. Aktualizuj matici center V podle vzorce (3.2) s využitím aktuální matice U .
4. Pokud $\|V^b - V^{b-1}\| < \varepsilon$, ukonči algoritmus. Jinak nastav $k = k+1$ a pokračuj od bodu 2.

Kde $\|*\|$ je euclidovská vzdálenost vektorů.

Míra věrohodnosti:

Jedním z nejobtížnějších úkolů ve shlukové analýze je nalezení vhodného počtu shluků. Velikost "fuzzifikace" v řešení se dá změřit Dunnovým rozdělovacím koeficientem, který představuje míru, jak těsně padne fuzzy řešení na odpovídající pevné shluky. Za pevné shluky budeme považovat klasifikaci každého objektu do shluku, který má největší účast. Dunnův rozdělovací koeficient se dá vyjádřit vzorcem :

$$F(U) = \frac{1}{N} \sum_{k=1}^c \sum_{i=1}^n m_{ik}^2$$

Koeficient leží v intervalu od $1/c$ do 1 . Tento koeficient nabývá hodnoty $1/c$ pro maximálně nejasný rozklad. Hodnota $F(U) = 1$ platí, když pro každý objekt je účast jednotková a zbytek je roven nule.

4 Návrh algoritmu

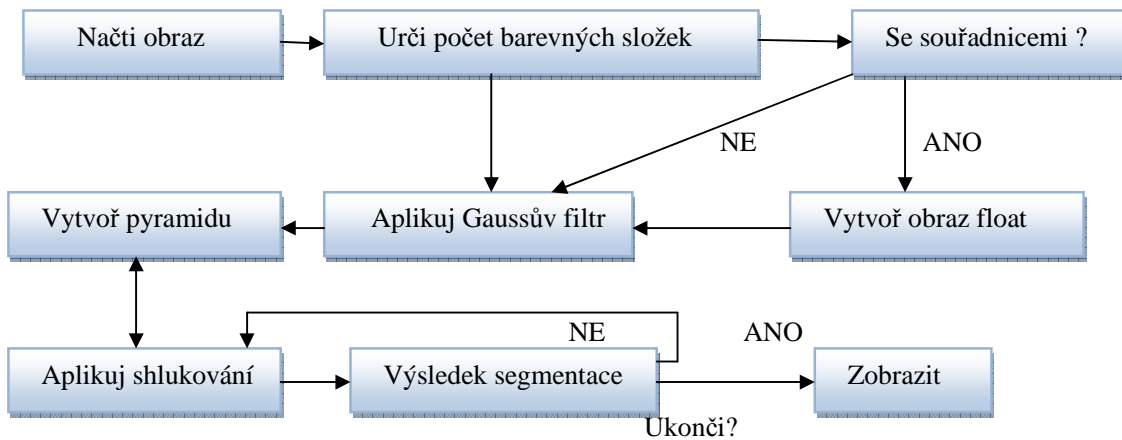
Shlukovací algoritmy jsou široce využívány k segmentaci barevného obrazu v aplikacích jako je rozpoznání tváře a otisku prstů, rozpoznání rakovinových buněk od zdravých a průmyslová automatizace. Pro obrazy ve vyšším rozlišení však rostou nároky na jejich výpočetní výkon. Využijeme proto strategie, kdy vytvoříme pyramidální reprezentaci obrazu, na kterou aplikujeme vybraný shlukovací algoritmus. Začneme postupně od dna pyramidy, kde se nachází obraz v nejnižším rozlišení. Ten rozsegmentujeme a výsledek středů center shluků uschováme a použijeme jako nová počáteční centra pro obraz s vyšším rozlišením na následujícím stupni pyramidy. Takto pokračujeme až k vrcholu pyramidy, kde se nachází původní obraz s nejvyšším rozlišením. Přitom předpokládáme rychlejší konvergenci algoritmu, kdy s průchodem jednotlivými stupni pyramidy by měl klesat počet iterací. Zrychlení se tak dosáhne výpočtem center na nižších stupních pyramidy, pro které procházíme méně obrazových bodů.

Algoritmus tedy můžeme popsat následovně.

1. Nastavíme počet center shluků K , kde $2 \leq K < N = M_1^{(b)} \times M_2^{(b)}$, b odpovídá nejnižšímu stupni pyramidy.
2. Nastavíme iterační krok na $k = 0$.
3. Nastav centra shluků $V^{(k)}$.
4. Vypočítáme novou matici $U^{(k)}$ příslušnosti ke shlukům.
5. Vypočítáme nová centra $V^{(k+1)}$.
6. Pokud $\|V_i^{(k+1)} - V_i^{(k)}\| < \varepsilon$ pro všechna i , jdi na krok 7, jinak nastav $k = k+1$ a jdi na krok 3.
7. Jestliže je $b = 0$, ukonči algoritmus, jinak pokračuj.
8. $M_1^{(b-1)} \times M_2^{(b-1)}$ je obraz na dalším stupni. Nastav iterační krok na $k = 0$.
9. Jestliže $k = 0$, použij centra $V^{(k+1)}$ z předchozího stupně, jinak použij centra $V^{(k)}$.
10. Vypočítáme novou matici $U^{(k)}$ příslušnosti ke shlukům.
11. Vypočítáme nová centra $V^{(k+1)}$.
12. $\|V_i^{(k+1)} - V_i^{(k)}\| < \varepsilon$ pro všechna i , jdi na krok 7, jinak nastav $k = k+1$ a jdi na krok 8.

$\| * \|$ určuje normu. Pro algoritmus fuzzy c-means je normou Euclidova vzdálenost vektorů. V algoritmu K-means je normou maximální hodnota sumy čtverců vzdáleností.

Celý návrh postupu si vyjádříme v následujícím diagramu:



Obrázek 4.1. Postup při hierarchické segmentaci obrazu

Kroky algoritmu si tedy dají shrnout několika body:

1. Načtení obrazu
2. Určení počtu jeho barevných kanálů.
3. V případě započítání souřadnic vytvoření nového obrazu.
4. Použití Gassova filtru a vytvoření pyramidy
5. Opětovná aplikace shlukování
6. Zobrazení výsledku segmentace.

5 Implementace

Celý program byl implementován v jazyce C++, který nabízí jak možnost objektového, tak procedurálního programování. Stejně, jako v použité knihovně OpenCv bylo pro výslednou aplikaci zvoleno programování procedurální. Jednotlivé typy segmentačních algoritmů jsou tak rozčleněny do celků, které obsahují všechny potřebné funkce pro běh daného algoritmu. Tyto celky však tvoří objekty ale statické třídy. Nabízejí tak pouze operace, ale nikoliv instance či vlastnosti, které jsou typické u objektového přístupu. Celý projekt byl vyvíjen v prostředí Microsoft Visual Studio 2008.

5.1 OpenCv

Pro implementaci algoritmů jsem se rozhodl pro knihovnu OpenCV (Open Source Computer Vision Library), dostupnou zdarma na adrese <http://sourceforge.net/projects/opencvlibrary>. Tato knihovna obsahuje řadu struktur a optimalizovaných algoritmů z oblasti zpracování obrazu a počítačového vidění.

V této práci jsem použil funkce z knihovny OpenCv pro načítání a ukládání obrazu. Ty podporují většinu obrazových formátů. Mohl jsem tak pracovat na již připravených datech a odpadla tak režije nutná pro implementaci těchto algoritmů. K vytvoření obrazové pyramidy využívám již zabudovanou funkci pro rozmazání obrazu pomocí Gaussového filtru a jeho následné podvzorkování. Pro reprezentaci dat v paměti jsem použil připravené struktury CvMat pro uchování matic a IplImage pro uchování načteného obrazu.

5.2 Vlastní implementace

Funkce pro segmentaci obrazu jsou dvojího druhu: klasické a hierarchické. Oba druhy očekávají jako vstup obrázek typu IplImage v 8-bitovém uchar formátu. Formát uchar byl zvolen pro zminimalizování paměťových nároků obou segmentačních algoritmů. Navíc je implicitním formátem funkce pro načtení obrazu, dostupné v knihovně OpenCV. Pro reprezentaci obrazu s pozicemi pixelů byl zvolen typ float. Při rozlišení obrazu větším než 256 pixelů je totiž rozsah datových hodnot typu uchar nevyhovující. Dalším důvodem je omezení vstupního datového typu zabudované funkce pro vytvoření Gaussovy pyramidy.

Segmentace probíhá následujícím způsobem. Na začátku každého algoritmu získáme počet barevných kanálů. U algoritmů se započítáním souřadnic je tento počet navýšen o souřadnice x a y .

Pokud byl zvolen pyramidální algoritmus, zavoláme funkci `createPyramid` pro vytvoření pyramidy. V případě volby pyramidálního algoritmu se započtenými souřadnicemi vytvoříme zavoláním funkce `createFloatXYimg`, obraz o jedné barevné složce vynásobený počtem sledovaných vlastností (počet barevných složek + počet souřadnic).

Poté zavoláme funkci `refreshUV(xy)` pro výpočet nových středů a matice příslušnosti vektoru ke shluku (hodnota xy značí funkce pro práci s obrazem se započtenými souřadnicemi). Tato funkce využívá informace o počtu barevných kanálů k segmentaci barevných složek vstupního obrazu v jednom průchodu. Algoritmy by tak měly podporovat segmentaci barevných i černobílých obrazů. Po ukončení algoritmu je vrácena matice příslušnosti. U algoritmu k -means představuje jeden bod matice vítězný shluk. Naopak v algoritmu fuzzy c -means představuje bod matice vektor příslušností ke shlukům. Pokud dojde podle stanovených kritérií k ukončení segmentačního algoritmu, zavoláme funkci pro obarvení výsledných shluků v originálním obraze.

6 Výsledky testování

Naimplementované algoritmy prošly několika základními testy, ve kterých byla sledována časová a paměťová náročnost a počet iterací nad jednotlivými stupni pyramidy.

Zdrojem dat z důvodu snadného porovnání s výsledky jiných segmentačních algoritmů je barevná standardizační fotografie peppers.jpg o rozlišení 512 x 512 pixelů. Testovací počítač má následující konfiguraci:

1. Model: Notebook HP compaq nx9020
2. CPU: Celeron M 1400MHz.
3. RAM: 736MB , OS: Microsoft Windows XP Professional version 2002, SP2.

6.1 Testování algoritmu k-means

Čas [s]	Počet shluků							
	2	3	4	5	6	7	8	9
k-means	0.200	1.474	1.843	1.733	2.904	7.631	7.771	7.230
Pyr. k-means	0.190	0.270	0.480	2.103	0.751	1.282	0.771	0.841
k-means XY	0.801	1.071	2.553	3.505	3.905	11.567	10.215	19.017
Pyr. k-means XY	1.202	1.162	2.594	7.951	7.151	11.206	10.455	23.975

Tabulka 6.1: maximum iterací: 100. Prostor: RGB

Z výsledků testů vidíme zkrácení doby nutné pro výpočet zejména u pyramidálního k-means, kdy se algoritmus chová podle očekávání. Došlo ke zkrácení doby trvání segmentace při zachování stejných vizuálních výsledků, jako u algoritmu bez použití pyramidy.

V postupu se započtením souřadnic došlo ke zpomalení výpočtu. To je oproti klasickému k-means dáno především nutností zpracovávat větší objem dat. Výsledné shluky jsou souvislejší bez přítomnosti děr v jednotlivých regionech. Segmentační hranice však přesně neodpovídají hranicím objektů ve vstupním obraze. U pyramidálního algoritmu se započtením souřadnic nedopadly výsledky podle očekávání. Rychlost segmentace zde byla snížena díky aplikaci Gaussova filtru. Pozice pixelů u obrazů ve vyšším rozlišení nabývají oproti barevným složkám intenzivnější hodnoty jasu, protože rozsah datových hodnot pro jednotlivé barvy je menší, než rozsah hodnot pro pozice pixelů v obraze. Gaussův filtr však tyto vysoké hodnoty jasu postupně přes jednotlivé stupně pyramidy odstraní. Centra shluků v obrazech o nižším rozlišení jsou tak určena špatně a jejich následné přepočítání na vyšším rozlišení zvyšuje počet iterací mezi jednotlivými stupni pyramidy, a tím taky dobu výsledné segmentace.

6.2 Testování algoritmu fuzzy c-means

Čas [s]	Počet shluků							
	2	3	4	5	6	7	8	9
Fuzzyc - means	6.22	27.47	39.15	40.39	86.22	113.13	142.55	230.68
Pyr. fuzzy c - means	3.64	17.7	15.85	26.98	29.37	35.27	37.74	47.308
Fuzzy c - means XY	18.54	270.5	70.4	79.30	158.07	665.08	294.35	870.96
Pyr. fuzzy c - means XY	55.84	157.06	119.01	392.44	175.06	370.08	725.69	1019.5

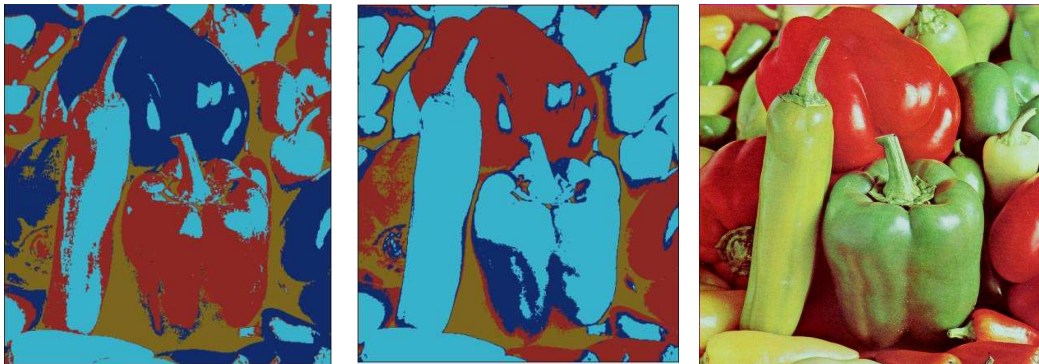
Tabulka 6.2: maximum iterací: 100. Prostor: RGB

Segmentace pomocí algoritmu fuzzy-c means se zachovala obdobně, jako tomu bylo u k-means.

I zde pyramidální reprezentace snížila časovou náročnost algoritmu se zachováním vizuálních výsledků. Doba celkové segmentace je oproti předchozímu algoritmu vyšší z důvodů rozdílného zpracování dat.

6.3 Testování v prostoru CILAB

Časová náročnost v prostoru CILAB byla obdobná časové náročnosti v prostoru RGB. Algoritmy tedy projdou v průměru podobným počtem iterací. Segmentace v prostoru L*a*b však podává lepší výsledky, kdy regiony jsou celistvější s menším počtem děr. Potvrdily se vlastnosti toho prostoru, kdy jednotlivé segmentační regiony jsou souvislejší (viz. obrázek 6.2).



Obrázek 6.2. Nalevo segmentace v prostoru RGB, uprostřed prostor CIELAB, napravo originál.

6.4 Porovnání algoritmů

V algoritmu fuzzy c-means jsme při použití pyramidální struktury dosáhli podobného zrychlení jako tomu bylo u k-means. Výsledky na testovaných obrázcích jsou u obou algoritmů velmi podobné. U fuzzy c-means jsou někdy výsledky celistvější s menším počtem děr. Záleží hlavně na nastavení kritéria ukončení a na velikosti fuzzy koeficientu. Jako optimální hodnota fuzzy koeficientu se osvědčila doporučená hodnota 2. V obou testovaných algoritmech při použití pyramidy záleží rychlost výsledku na jejím stupni. S příliš velkým stupněm pyramidy ztrácíme některé detaily v obraze a výsledné shluky na nižších stupních pyramidy pak nemusí být optimální. Nadměrným rozmazáním obrazu na nižších stupních pyramidy může docházet k přesegmentování obrazu. Takto přesegmentovaný obraz pak má vliv na celkový počet iterací mezi stupni pyramidy. Ve výsledku se tak celý proces segmentace zpomalí. Na celkový výsledek segmentace má taky vliv počáteční inicializace center shluků. Optimální počet center jsem se snažil zjistit pomocí Dunnova koeficientu. V mnoha případech však tato hodnota neodpovídala reálným výstupům segmentace, a proto nebyla k určení počtu shluků v algoritmu využita.

7 Závěr

Cílem této práce bylo prostudovat vybrané hierarchické metody segmentace obrazu. V současné době je k dispozici jen málo materiálů v českém jazyce zabývajících se touto problematikou. Proto jsem se snažil vytvořit ucelený přehled použití základních hierarchických segmentačních principů spolu s popisem, jakým způsobem využívají hierarchických datových struktur k segmentaci obrazu.

V implementační části jsem se zaměřil na použití obrazových pyramid společně se shlukovacími algoritmy fuzzy c-means a k-means. Podle výsledků testů můžeme říci, že začlenění pyramid do těchto algoritmů bylo úspěšné a došlo k nárůstu výkonu segmentace. Oba algoritmy však stále potřebují zadat optimální počet shluků. Tento problém jsem se snažil vyřešit pomocí Dunnova rozdělovacího indexu. V průběhu testování jsem sledoval jeho hodnoty, které by však na výsledek segmentace neměly žádný vliv, a tak bylo od jeho začlenění do segmentačního procesu a získání optimálního počtu shluků upuštěno. Výsledek segmentace jsem se dál snažil vylepšit použitím barevného prostoru CIELAB. Můžeme říct, že oproti prostoru RGB podává zpravidla lepší výsledky.

V naimplementovaných algoritmech určitě ještě existuje prostor k optimalizaci. Proto bych u shlukovacích segmentačních algoritmů doporučil se dále zabývat problémem určení optimálního počtu shluků a ukončovacího kritéria. Jako další námět k pozorování doporučuji zkoumat vliv počtu stupňů v pyramidální struktuře a jeho optimální určení. Dalším krokem by mohlo být vytvoření uživatelsky přívětivého prostředí, které by uživateli umožnilo intuitivně ovládat jednotlivé metody.

8 Literatura

- [1] Fiřt, J., Holota, L.: Digitalizace a zpracování obrazu
Dokument dostupný na URL:
<http://home.zcu.cz/~holota5/publ/DigZprO.pdf> (květen 2008)
- [2] Yu-Jin Zhang. Advances in Image and Video Segmentation, IGI 2006, s 3-4, 6.
Dokument dostupný na URL:
<http://books.google.cz/books?id=VMWrAh2Di2QC> (květen 2008)
- [3] Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision, PWS, Boston, 1999. Dokument dostupný na URL:
<http://cmp.felk.cvut.cz/~hlavac/Public/Pu/33PVRleto2003/ch05d.pdf> (květen 2008)
- [4] Žára, J., Beneš, B., Sochor, J.: Moderní počítačová grafika, Computer Press, Brno, 2004
- [5] Vyhlazování (filtrace) [online], Plzeň: ZČU v Plzni, fakulta aplikovaných věd, katedra kybernetiky. Dokument dostupný na URL:
<http://www.kky.zcu.cz/cs/courses/zdo/lesson3> (květen 2008)
- [6] Popis barev a základy barvových prostorů[online],AZ reprodukce barev.
Dokument dostupný na URL:
<http://www.reprodukce-barev.org/?menu=3> (květen 2008)
- [7] Kebeľ, J., Šilhán, D., Shluková analýza.
Dokument dostupný na URL:
<http://gerstner.felk.cvut.cz/biolab/X33BMI/slides/KMeans.pdf> (květen 2008)
- [8] Bryan S.Morse: Data Structures for Image Analysis, Brigham Young University, 1998 – 2000
- [9] Meloun, M.: Analýza shluků CLU, Univerzita Pardubice.
Dokument dostupný na URL:
<http://meloun.upce.cz/kapitoly/4gmetody.pdf> (květen 2008)
- [10] Dong, L., Ogunbona, P., Wanghing Li: A fast algorithm for color image segmentation, University of Wollongong, Australia Shenyan University, Northeastern University, China.

- [11] Tinka Acharya, Ajoy K. Ray.:Image processing principles and applications, Wiley-Interscience, New Jersey 2005.

Příloha:

V kořenovém adresáři se nachází text práce ve formátu pdf. V adresáři source se nachází zdrojové kódy programu. V adresáři input je používán testovací obrázek. V adresáři output jsou výsledky testů.

