

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

PLZEŇ, 2015

Bc. Josef Ježek

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Josef JEŽEK**
Osobní číslo: **A13N0139P**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Řídicí a rozhodovací systémy**
Název tématu: **Segmentace obrazu pomocí Markovských náhodných polí**
Zadávající katedra: **Katedra kybernetiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s dostupnou literaturou týkající se Markovských náhodných polí.
2. Implementujte interaktivní segmentaci obrazu za pomoci výše zmíněné teorie.
3. Navrhněte vhodný automatický algoritmus pro segmentaci obrazu a výsledky porovnejte s interaktivní verzí a jinými automatickými metodami.
4. Diskutujte možnosti dalšího rozšíření Vámi navrženého postupu.
5. V závěru obecně zhodnoťte vhodnost Markovských náhodných polí a jejich reálné nasazení pro úlohu segmentace obrazu.

Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **40-50 stránek A4**
Forma zpracování diplomové práce: **tištěná**
Seznam odborné literatury:

1. **Milan Šonka, Václav Hlaváč, Roger Boyle: Image Processing, Analysis and Machine Vision**
2. **Stan Z. Li: Markov Random Field Modeling in Image Analysis**
3. **Andrew Blake, Pushmeet Kohli, Carsten Rother: Markov Random Fields for Vision and Image Processing**

Vedoucí diplomové práce: **Ing. Tomáš Ryba**
Nové technologie pro informační společnost

Datum zadání diplomové práce: **1. října 2014**
Termín odevzdání diplomové práce: **15. května 2015**



Doc. RNDr. Miroslav Lávička, Ph.D.
děkan



Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 1. října 2014

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne

.....

Abstrakt

Diplomová práce se zabývá segmentací obrazu pomocí teorie Markovských náhodných polí. Práce zkoumá segmentaci metodou Graph cut, zkoumá výhody i nevýhody této segmentační metody. Věnuje se i předzpracování vstupních obrazů takovým způsobem, aby výsledná segmentace dopadla co možná nejlépe.

Práce uvádí postupy automatického získání jasových modelů, ty jsou určeny střední hodnotou a rozptylem. Porovnáními ukazuje vliv vypočtené hodnoty rozptylu z různé velikosti okolí střední hodnoty na výsledné segmentace. Porovnává Graph cut s podobnými segmentačními metodami. Práce přináší výsledky praktických experimentů i zhodnocení experimentů.

Klíčová slova

Markovská náhodná pole, Graph cut, jasový model, histogram, Probabilistic Rand index, normální rozdělení, rozptyl

Abstract

This diploma thesis is focused on image segmentation based on theory of Markov Random Fields. Segmentation based on Graph cut method is studied in this thesis, advantages and disadvantages of this method are explored. Attention is paid also input image pre-processing in order to final segmentation was as good as possible.

The methods of automatic gaining brightness models are published, brightness models are determined by mean value and variance. The influence on resulting segmentations of computed variance value from different sizes of mean value environ is proved by comparisons. Graph cut is compared with similar segmentation methods. The results of practical experiments and the evaluation of the experiments are presented.

Keywords

Markov Random Fields, Graph cut, brightness model, histogram, Probabilistic Rand index, Normal distribution, variance

Obsah

1 Úvod.....	13
2 Základní pojmy	15
2.1 Histogram	15
2.2 Konvoluce	17
2.3 Vyhlazení histogramu	17
2.4 Střední hodnota	19
2.5 Rozptyl	19
2.6 Normální rozdělení.....	19
2.7 Segmentace.....	20
2.8 Rand index.....	21
2.9 Probabilistic Rand index	22
3 Markovská náhodná pole v počítačovém zpracování obrazu.....	24
3.1 Proč právě Markovská náhodná pole	24
3.2 Segmentace a zařazení pixelů do tříd	24
3.3 Přístup za pomoci pravděpodobnosti	26
3.4 Okolí.....	26
3.5 Definice Markovských náhodných polí	28
3.6 Hammersley-Cliffordův teorém a Gibbsova náhodná pole.....	29
3.7 Modely	30
3.8 Optimalizace.....	32
3.9 Graph cut	33
3.10 Maximální tok (minimální řez)	33
3.11 Large moves	34
3.11.1 α -expansion	35
3.11.2 α - β swap	36
4 Praktická část.....	38
4.1 Vyhlazení histogramu	38
4.2 Detekce maxim.....	38
4.3 Nevýhody funkce argrextrema	39
4.4 Automatické vytvoření jasových modelů.....	40
4.5 Skóre funkcí aic a bic[5]	40
4.6 Berkeley Segmentation Data Set[9][10]	41

4.7 Pokusy s PR indexem	42
4.8 Závěry k určování velikosti okolí.....	49
4.9 Porovnání	49
4.10 Felzenszwalbův algoritmus	50
4.10.1 Krátký popis algoritmu.....	50
4.10.2 Porovnání výsledků algoritmu s metodou Graph cut	52
4.11 Cut-normalized.....	55
4.11.1 Krátký popis algoritmu.....	55
4.11.2 Porovnání výsledků algoritmu s metodou Graph cut	56
4.12 Možná vylepšení	59
4.13 Vhodnost reálného nasazení metody Graph cut a teorie Markovských náhodných polí v praxi.....	60
4.14 Medicínské snímky	60
5 Závěr.....	65
6 Použité zdroje	67

Seznam obrázků

Obr. č. 2.1 – Vykreslený digitalizovaný obraz v šedotónové reprezentaci	16
Obr. č. 2.2 – Histogram šedotónového obrazu Lena (obr. č. 2.1)	16
Obr. č. 2.3 – Vyhlazený histogram šedotónového obrazu Lena (obr. č. 2.1) za použití jádra o velikosti 5	18
Obr. č. 2.4 - Vyhlazený histogram šedotónového obrazu Lena (obr. č. 2.1) za použití jádra o velikosti 3	18
Obr. č. 2.5 – Ilustrační ukázka segmentace metodou Graph cut, vlevo původní obraz, vpravo segmentovaný obraz, metoda segmentovala obraz do tří tříd na základě hodnot jasu.....	20
Obr. č. 2.6 – Ukázka příkladů manuálních segmentací původního obrázku „Man with hat“ (vlevo nahoře), obrázek pochází ze sady Berkeley Segmentation Data Set[9][10] .	21
Obr. č. 3.1 – Ilustrační ukázka čtyřokolí, uprostřed mřížky se nachází zpracováváný pixel, vybarvené čtverce v okolí představují pixely, se kterými se pracuje v případě čtyřokolí	27
Obr. č. 3.2 – Ilustrační ukázka osmiokolí, uprostřed mřížky se nachází zpracováváný pixel, vybarvené čtverce v okolí představují pixely, se kterými se pracuje v případě osmiokolí	27
Obr. č. 3.3 – Ilustrační ukázka dělení klik v závislosti na počtu uzlů a v závislosti na uvažovaném okolí[3]	30
Obr. č. 3.4 – Ilustrační ukázka sestaveného grafu pro hledání minimálního řezu grafem s naznačeným řezem grafem, S je uzel source a T je uzel sink[11]	33
Obr. č. 3.5 – Ilustrační ukázka práce algoritmů založených na large moves. Obrázek (a) ukazuje segmentovaný obraz. Obrázky (c) a (d) ukazují výsledky algoritmů α -expansion (c)) a α - β swap (d)), které v jednom kroku změny labely velkému množství pixelů. Algoritmy založené na standard moves změny v jednom kroku přiřazení pouze jednomu pixelu.[11]	35
Obr. č. 4.1 – Obrázek Mountains church ze sady Berkeley Segmentation Data Set[9] a lidmi vytvořené manuální segmentace	42
Obr. č. 4.2 – Obrázek Mountains church ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 30 (tmavě modrá), 60 (světle modrá), 100 (žlutá), 255 (tmavě červená)	43
Obr. č. 4.3 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Mountains Church	43

Obr. č. 4.4 – Obrázek Elephants ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 38 (modrá), 75 (zelená), 139 (tmavě červená)	44
Obr. č. 4.5 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Elephants	44
Obr. č. 4.6 – Obrázek Polar bears ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 13 (modrá), 125 (zelená), 182 (tmavě červená)	45
Obr. č. 4.7 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Polar Bears	45
Obr. č. 4.8 – Obrázek Church ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 97 (tmavě modrá), 123 (světle modrá), 155 (žlutá), 255 (tmavě červená)	46
Obr. č. 4.9 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Church	46
Obr. č. 4.10 – Obrázek Lady dots ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 158 (modrá), 195 (zelená), 238 (tmavě červená)	47
Obr. č. 4.11 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Lady Dots	47
Obr. č. 4.12 – Obrázek Moon ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 27 (modrá), 79 (zelená), 255 (tmavě červená)	48
Obr. č. 4.13 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Moon	48
Obr. č. 4.14 – Ukázka segmentace Felzenszwalbovým algoritmem (vpravo) vstupního obrazu Man with hat ze sady Berkeley Segmentation Data Set[9] (vlevo), parametry byly nastaveny na následující hodnoty: měřítko ovlivňující velikost shluků bylo zvoleno 1,	

sigma pro vyhlazení Gaussovským jádrem byla nastavena na hodnotu 5 a minimální velikost jedné složky byla nastavena na hodnotu 100.....	52
Obr. č. 4.15 – Porovnání segmentací vstupního obrázku Farm ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a Felzenszwalbovým algoritmem (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.450, hodnota PR indexu segmentace Felzenszwalbovým algoritmem byla 0.894	53
Obr. č. 4.16 - Porovnání segmentací vstupního obrázku Birds ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a Felzenszwalbovým algoritmem (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.966, hodnota PR indexu segmentace Felzenszwalbovým algoritmem byla 0.802	54
Obr. č. 4.17 - Porovnání segmentací vstupního obrázku Eskymo ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a Felzenszwalbovým algoritmem (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.668, hodnota PR indexu segmentace Felzenszwalbovým algoritmem byla 0.666	54
Obr. č. 4.18 - Ukázka segmentace algoritmem Normalized cut (vpravo) vstupního obrazu Man with hat ze sady Berkeley Segmentation Data Set[9] (vlevo), kromě parametru prahu, který byl nastaven na hodnotu 0.05, byly hodnoty ostatních parametru ponechány na výchozích hodnotách.....	56
Obr. č. 4.19 - Porovnání segmentací vstupního obrázku Mushroom ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a metodou Normalized cut (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.507, hodnota PR indexu segmentace metody Normalized cut byla 0.898.....	57
Obr. č. 4.20 - Porovnání segmentací vstupního obrázku Birds ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a metodou Normalized cut (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.966, hodnota PR indexu segmentace metody Normalized cut byla 0.529.....	58
Obr. č. 4.21 - Porovnání segmentací vstupního obrázku Deer ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a metodou Normalized cut (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.690, hodnota PR indexu segmentace metody Normalized cut byla 0.687.....	58
Obr. č. 4.22 – Názorné porovnání hodnoty PR indexu segmentací Graph cutem, N-Cutem a Felzenszwalbovým algoritmem na některých obrázcích sady Berkeley Segmentation Data Set[9] , pod číslem 23 se skrývá střední hodnota PR indexu metody.....	59

- Obr. 4.23 – Snímek z CT, v játrech (segmentována žlutě) je přítomen hyperdenzní nádor, ten je přiřazen do špatné třídy, segmentace získána výše navrženým způsobem, rozptyly jsou počítány na základě okolí velikosti 30 jasových úrovní, přítomno velké množství šumu⁶¹
- Obr. 4.24 – Stejný snímek jako na obrázku 4.23, segmentace získána výše navrženým způsobem s přenásobením korekčního koeficientu o velikosti 20, rozptyly jsou počítány na základě okolí velikosti 30 jasových úrovní, přenásobením korekčním koeficientem došlo k potlačení šumu 62
- Obr. 4.25 – Snímek z CT, v játrech (segmentována žlutě) je přítomen hypodenzní nádor, ten je přiřazen do špatné třídy, segmentace získána výše navrženým způsobem, rozptyly jsou počítány na základě okolí velikosti 30 jasových úrovní, přítomno velké množství šumu⁶²
- Obr. 4.26 - Stejný snímek jako na obrázku 4.25, segmentace získána výše navrženým způsobem s přenásobením korekčního koeficientu o velikosti 20, rozptyly jsou počítány na základě okolí velikosti 30 jasových úrovní, přenásobením korekčním koeficientem došlo k potlačení šumu, v tomto případě také ke správnému zařazení nádoru do třídy odpovídající nejbližšímu jasovému modelu 63
- Obr. 4.27 – Snímek z CT, v játrech (segmentována žlutě) je přítomen hyperdenzní nádor, který nebyl ve výsledné segmentaci detekován 64
- Obr. 4.28 - Snímek z CT, v játrech (segmentována žlutě) je přítomen hypodenzní nádor, v tomto případě došlo ve výsledné segmentaci k detekci nádoru, nádor by přiřazen správně do třídy odpovídající nejbližšímu jasovému modelu..... 64

1 Úvod

Pokrok nepřináší jen vývoj nových technologií. Pokroku lze dosáhnout též zdokonalováním již objevených a vynalezených technologií. Mnohdy zdokonalováním již objeveného objevíme nečekané věci, či prozkoumáme jevy, o kterých toho dosud není mnoho známo. V posledních letech se do popředí zájmu dostává počítačové zpracování obrazu a počítačové vidění. Mezi slabiny tohoto přístupu patří vysoká výpočetní náročnost, nicméně s dokonalejšími výpočetními prostředky je možné v této oblasti dosáhnout velmi uspokojivých výsledků.

Motivací pro práci bylo vylepšení modelů sloužících k segmentaci metodou Graph cut. Použití segmentovaných obrazů metodou Graph cut je široké, proto považuji práci na modelech sloužících k segmentaci touto metodou jako smysluplnou. Segmentaci Graph cutem je totiž možné použít v široké oblasti reálných úloh, například i k detekci nádorů v lékařských snímcích.

Tato práce se zabývá segmentací obrazů pomocí teorie Markovských náhodných polí. Segmentace v práci byly prováděny metodou Graph cut. Aby mohla segmentace proběhnout bylo nutné vytvořit jasové modely z každého vstupního obrazu.

Jasové modely byly vytvořeny automaticky na základě histogramu vstupního obrazu. Aby bylo možné jasové modely vytvořit, bylo nutné přistoupit k předzpracování histogramu. Histogram byl vyhlazen Gaussovským jádrem. Ve vyhlazeném histogramu pak byla detekována informačně nejvýznamnější maxima. Na jasové úrovni těchto maxim byly automaticky vytvořeny jasové modely. Jasové modely jsou Gaussovy křivky, střední hodnotu představuje jasová úroveň, rozptyl byl získán na základě charakteristiky histogramu v okolí střední hodnoty.

Pro zjištění nejlepších jasových modelů byla provedena srovnání výsledných segmentací provedených s těmito automaticky získanými modely. Segmentace byly porovnány za pomoci Probabilistic Rand Indexu (= PR index)[6]. Pro porovnání byly využity obrázky ze sady Berkeley Segmentation Data Set[9][10]. Porovnání se provádí vždy nad množinou sady lidmi segmentovaných obrázků. Cílem takového porovnání je zajištění smysluplnosti výsledných segmentací.

Byl očekáván určitý průběh hodnot PR indexu[6] segmentací, které k výpočtu rozptylu uvažují různé velikosti okolí střední hodnoty. Bylo očekáváno, že největší hodnota PR indexu bude při porovnávání získána přibližně u velikosti okolí 30 úrovní.

Cílem práce bylo i porovnání segmentací metodou Graph cut se segmentačními metodami, které obdobně jako Graph cut stojí na základě řezu grafem. Cílem bylo dokázat, že metoda Graph cut je použitelná v reálných úlohách počítačového zpracování obrazu.

Práce je rozdělena na dvě části. První částí je část teoretická. V teoretické části je čtenář obeznámen s pojmy, které byly použity při vypracování práce. Jedná se o pojmy z oblasti počítačového zpracování obrazu[1],[2] i statistiky[4]. Jeden díl teoretické části je věnován rozsáhlé oblasti teorie Markovských náhodných polí[5],[7],[8].

V praktické části jsou uvedeny získané výsledky a postupy, kterými bylo prezentovaných výsledků dosaženo. V praktické části jsou rovněž popsány experimenty, úspěšné i neúspěšné, a porovnání. Porovnání byla provedena za pomoci PR indexu[6], jednak sledováním vývoje hodnoty PR indexu i porovnáním hodnot PR indexu segmentací metodou Graph cut s dalšími dvěma podobnými segmentačními metodami.

2 Základní pojmy

2.1 Histogram

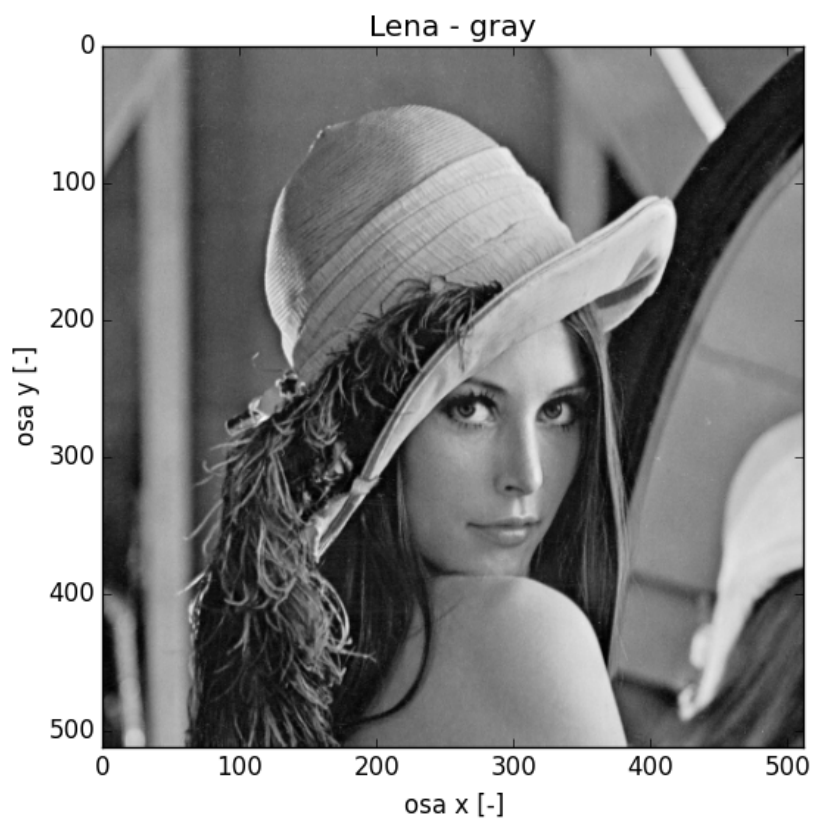
Histogram[1],[2] je grafickým vyjádřením četnosti dat v určitém souboru dat. V oblasti zpracování digitalizovaného obrazu vyjadřuje histogram četnost jasových úrovní v obraze o rozměrech $N \cdot M$ pixelů. Absolutní histogram sestavíme tak, že pro každou hodnotu jasu sečteme body s danou hodnotou jasu (viz obr. č. 2.2).

$$H(p) = \sum_x^N \sum_y^M f(x, y, p),$$
$$f(x, y, p) = 1: f(x, y) = p,$$
$$f(x, y, p) = 0: f(x, y) \neq p.$$
(2.1)

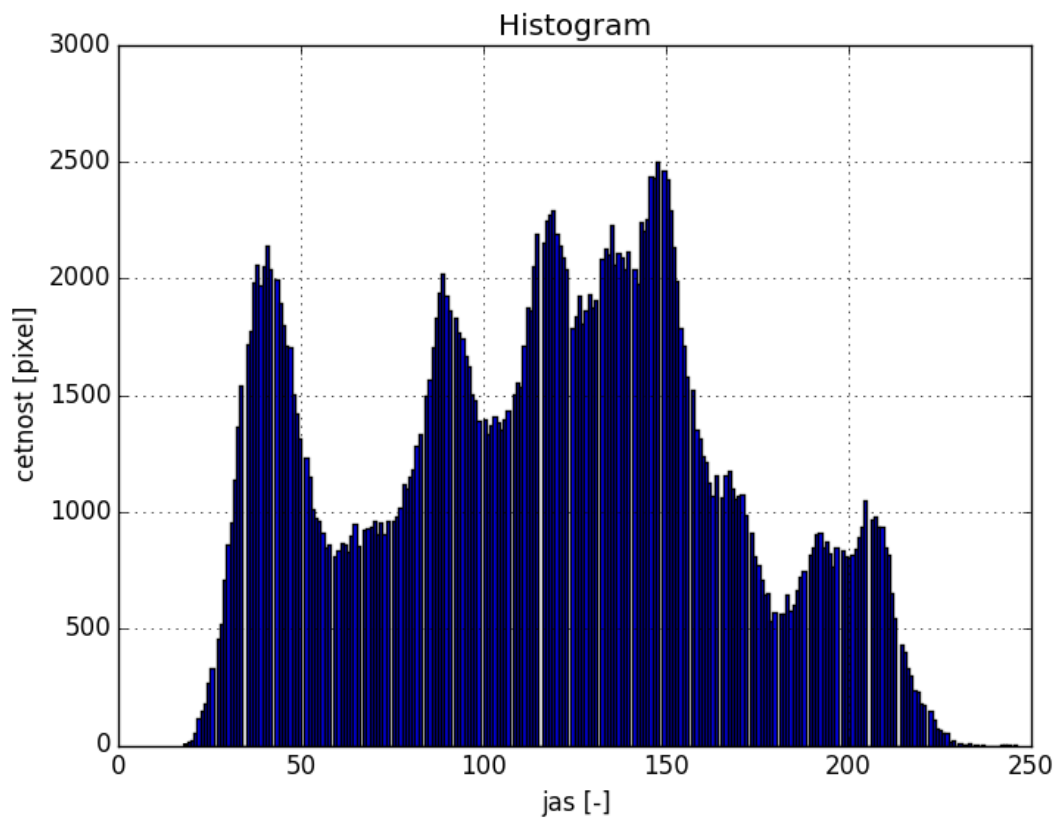
V oblasti zpracování digitalizovaného obrazu je často používán relativní histogram. Relativní histogram vyjadřuje pravděpodobnost výskytu dané jasové úrovně v obraze. Relativní histogram získáme vydělením četnosti každé jasové úrovně počtem pixelů v obraze.

$$H_r(p) = \frac{H(p)}{\sum_{p=0}^{255} H(p)},$$
$$\sum_{p=0}^{255} H_r(p) = 1.$$
(2.2)

Histogram poskytuje informaci o četnostech jasů v obraze. Histogram má i svoje nevýhody, např. pro různé obrazy můžeme získat stejný histogram[3].



Obr. č. 2.1 – Vykreslený digitalizovaný obraz v šedotónové reprezentaci



Obr. č. 2.2 – Histogram šedotónového obrazu Lena (obr. č. 2.1)

2.2 Konvoluce

Konvoluce[3] je matematický postup, který slouží k filtraci signálů, často např. k vyhlazení signálů. Na vstupu jsou dvě funkce – funkce, kterou chceme filtrovat, a jádro. Výstupem konvoluce je již filtrovaná funkce. Konvoluci značíme hvězdičkou a je definována následovně:

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t - \tau)d\tau. \quad (2.3)$$

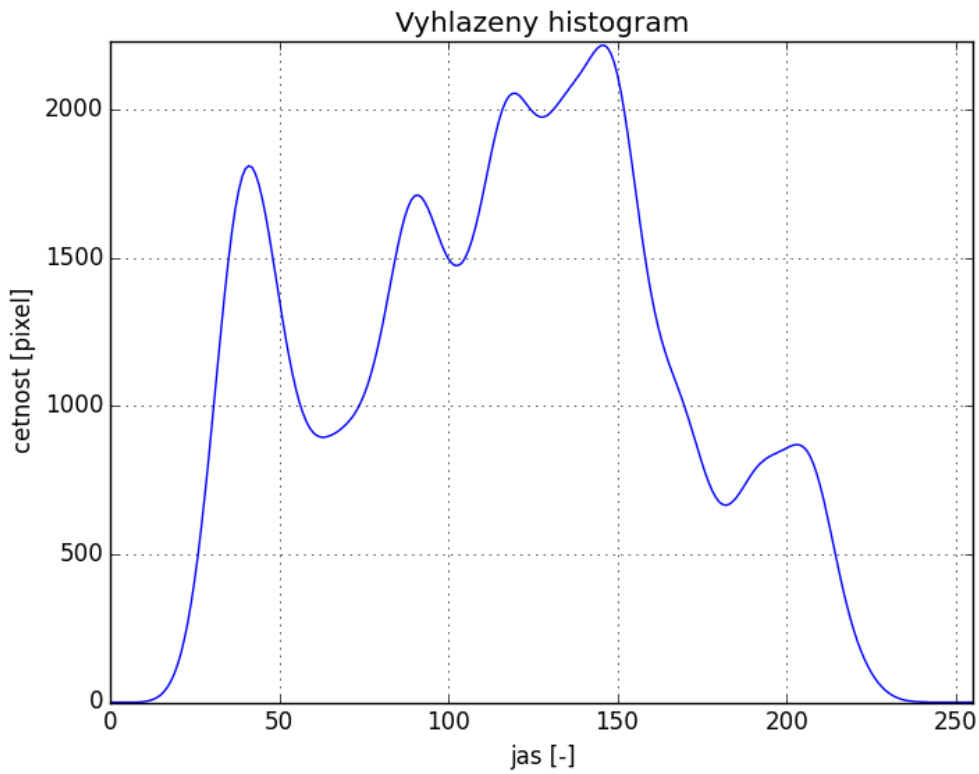
Konvoluce je počítána v bodě t . V oblasti počítačového zpracování obrazu nepracujeme se spojitými funkcemi, počítáme s diskrétními signály. Tyto signály bývají konečné. Diskrétní konvoluce je definována následovně:

$$(f * g)[n] = \sum_{m=-\infty}^{+\infty} f[m]g[n - m]. \quad (2.4)$$

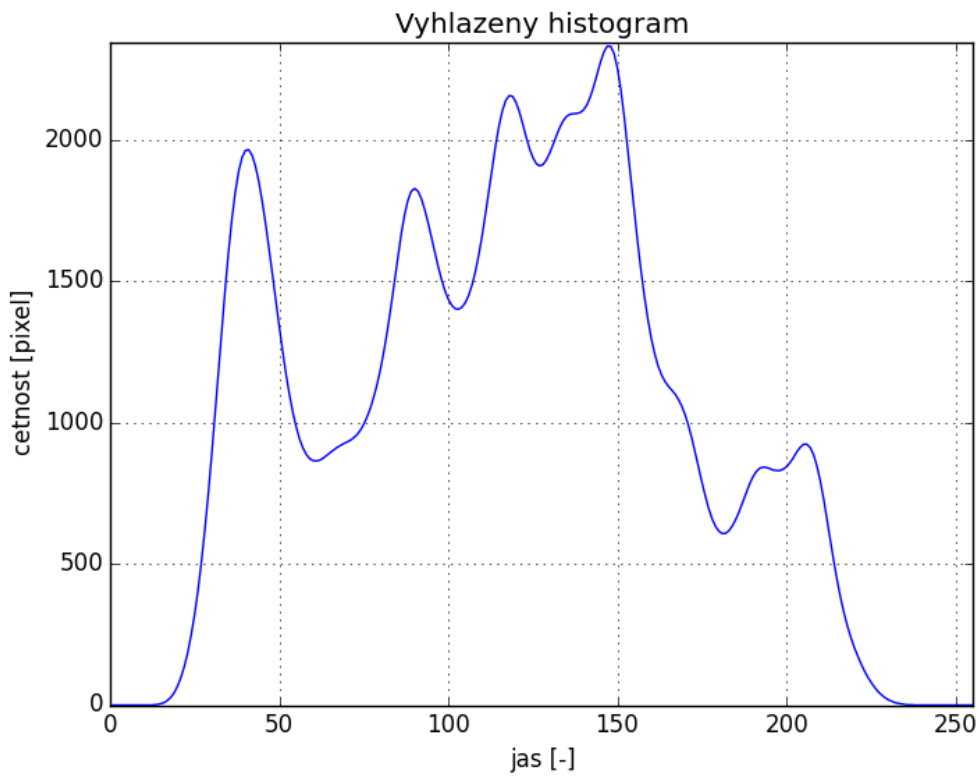
2.3 Vyhlazení histogramu

Histogram obsahuje mnoho nevýznamných lokálních extrémů, ty považujeme za šum a ztěžují další práci s histogramem, například při umístění prahu během prahování nebo během tvorby jasových modelů. Pro další práci s histogramem je užitečná znalost nejvýznamnějších lokálních extrémů včetně jejich polohy. Cílem vyhlazení histogramu je potlačení nevýznamných lokálních extrémů. Po vyhlazení je získán histogram obsahující pouze významné lokální extrémy.

Za účelem vyhlazení histogramu[1],[3] používáme diskrétní konvoluci. Vstupem je histogram obrazu a jádro. Jelikož nechceme změnit polohy lokálních extrémů, používáme taková jádra, která jsou symetrická a mají největší hodnotu uprostřed. Velikost jádra volíme s ohledem na řešený problém. Příliš malá jádra nepotlačí všechny nevýznamné lokální extrémy (viz obr. č. 2.4), naopak příliš velká jádra potlačí i významné lokální extrémy (obr. č. 2.3).



Obr. č. 2.3 – Vyhlazený histogram šedotónového obrazu Lena (obr. č. 2.1) za použití jádra o velikosti 5



Obr. č. 2.4 - Vyhlazený histogram šedotónového obrazu Lena (obr. č. 2.1) za použití jádra o velikosti 3

2.4 Střední hodnota

Střední hodnota[4] je ve statistice hojně používána. Střední hodnota je číslo, které znamená průměrnou hodnotu. Této průměrné hodnoty nabývá náhodná veličina X . Střední hodnota je ve spojitém případě definována:

$$E(X) = \int_{-\infty}^{+\infty} x f(x) dx . \quad (2.5)$$

V diskrétním případě se integrál mění na sumu a střední hodnota je definována:

$$E(X) = \sum_{i=1}^N x_i p_i . \quad (2.6)$$

2.5 Rozptyl

Rozptyl[4] je ve statistice často používán. Jedná se o druhý centrální moment náhodné veličiny a vyjadřuje variabilitu rozdělení hodnot souboru kolem střední hodnoty tohoto souboru. Rozptyl je definován jak pro spojitou, tak diskrétní náhodnou veličinu. Pro spojitou náhodnou veličinu je definován následovně:

$$\sigma^2 = \int_{-\infty}^{+\infty} (x_i - E(X))^2 f(x) dx . \quad (2.7)$$

Pro diskrétní náhodné veličiny je rozptyl definován takto:

$$\sigma^2 = \sum_{i=1}^N (x_i - E(X))^2 p_i . \quad (2.8)$$

2.6 Normální rozdělení

Normální, neboli Gaussovské, rozdělení[4] je určeno střední hodnotou μ a rozptylem σ^2 . Normální rozdělení je spojité a značíme jej:

$$N(\mu, \sigma^2). \quad (2.9)$$

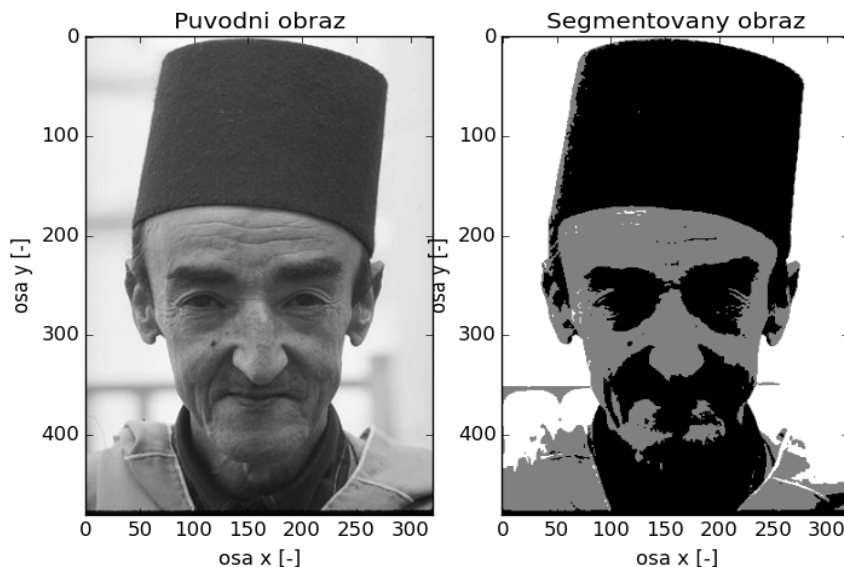
Hustota pravděpodobnosti normálního rozdělení $f(x)$ je dána vzorcem:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (2.10)$$

Ve statistice a pravděpodobnosti je normální rozdělení velmi často používáno, obzvláště při aproximacích.

2.7 Segmentace

Segmentací[1],[2],[3] v oblasti počítačového zpracování obrazu rozumíme rozdělení zpracovávaného obrazu na oblasti zájmu a na pozadí, viz obr. č. 2.5. Oblasti zájmu mohou být objekty reálného světa, které hledáme v obraze. Při segmentaci často používáme znalosti o řešeném problému za účelem zlepšení výsledku segmentace. Existuje mnoho různých technik segmentace. Mezi nejznámější patří prahování, srovnávání se vzorem, hledání hranic mezi oblastmi nebo v této práci použitá metoda Graph cut.

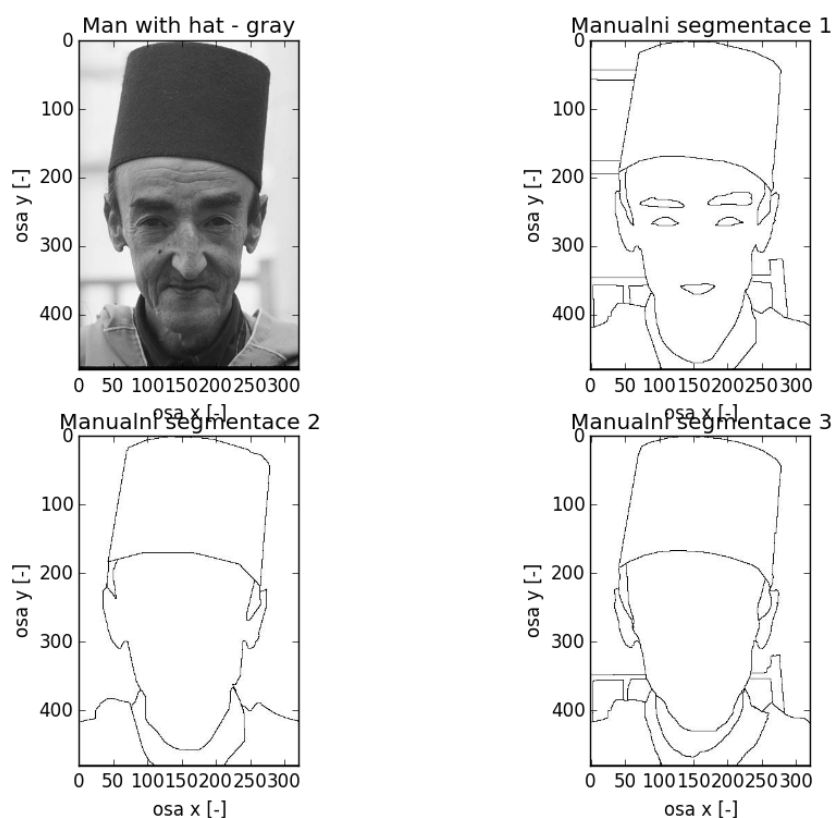


Obr. č. 2.5 – Ilustrační ukázka segmentace metodou Graph cut, vlevo původní obraz, vpravo segmentovaný obraz, metoda segmentovala obraz do tří tříd na základě hodnot jasu

2.8 Rand index

Spolu se segmentací je v oblasti počítačového zpracování obrazu řešena kvalita i hodnocení kvality segmentace. Pro hodnocení kvality segmentace nejsou jasně definovány základy ani pravidla. Neexistence základů a pravidel je způsobena nejednoznačným pohledem na segmentaci. Jinak obrázek automaticky segmentuje počítač, tentýž obrázek odlišně segmentuje člověk. V současné době je pro hodnocení kvality segmentace využíváno více přístupů. Velmi oblíbeným způsobem pro ohodnocení kvality segmentace se v poslední době stalo srovnávání automatické, počítačem vytvořené, segmentace s množinou člověkem segmentovaných vzorků obrazů.

Úmyslně zmiňuji množinu člověkem segmentovaných vzorků obrazů. Obrazy byly rozdány lidem, lidé tyto obrazy nasegmentovali. Každý účastník tohoto experimentu nasegmentoval obraz trochu jinak. Jeden oddělil objekt od pozadí, jiný rozdělil pozadí či objekt na více částí, viz obr. č. 2.6.



Obr. č. 2.6 – Ukázka příkladů manuálních segmentací původního obrázku „Man with hat“ (vlevo nahoře), obrázek pochází ze sady Berkeley Segmentation Data Set[9][10]

Jednou z metod ohodnocení kvality segmentace je metoda Rand index[6]. Metodu navrhl William Rand. Je hledána funkce podobnosti, srovnávané segmentované obrazy mohou mít i jiný počet tříd. Hodnota funkce je spočítána jako podíl čísel dvojic bodů, které byly segmentovány do stejné třídy ve dvou různých segmentacích S a S' .

Rand index je možné vypočítat:

$$R(S, S') = 1 - \frac{\left[\frac{1}{2} (\sum_u n_{u\bullet}^2 + \sum_v n_{\bullet v}^2) - \sum_{u,v} n_{uv}^2 \right]}{N(N-1)/2}, \quad (2.11)$$

kde S a S' jsou dvě různé segmentace jednoho obrazu. N vyjadřuje celkový počet pixelů v obraze, n_{uv} je počet pixelů, které patří do třídy u v segmentovaném obraze S a které patří do třídy v v segmentovaném obraze S' . Značením $n_{u\bullet}$ vyjadřujeme počet pixelů zařazených do třídy u v segmentovaném obraze S a značením $n_{\bullet v}$ vyjadřujeme počet pixelů zařazených do třídy v v segmentovaném obraze S' . Platí, že:

$$\begin{aligned} n_{u\bullet} &= \sum_v n_{uv}, \\ n_{\bullet v} &= \sum_u n_{uv}, \\ N &= \sum_u n_{u\bullet} = \sum_v n_{\bullet v}. \end{aligned} \quad (2.12)$$

Vypočtené hodnoty metodou Rand index se pohybují v intervalu mezi nulou a jedničkou. Nula znamená, že dvě odlišné segmentace jednoho obrazu nemají žádné shody, jednička znamená, že dvě odlišné segmentace jednoho obrazu jsou shodné.

2.9 Probabilistic Rand index

Probabilistic Rand index[6], neboli pravděpodobnostní Rand index, využívá statistické vlastnosti metody Rand index. Opět máme obrázek X a sadu lidí manuálně segmentovaných vzorků obrazu X , tuto sadu označíme písmenem A .

$$A = \{S_1, S_2, \dots, S_K\}. \quad (2.13)$$

Obrázek X má určité rozměry a obsahuje N pixelů. Dále mějme nějakou počítačem vytvořenou segmentaci obrazu X , tu nazveme S . Automatická segmentace je srovnávána s množinou lidmi manuálně segmentovaných vzorků obrazu X . Pixel x_i z obrazu X zařazený do nějaké třídy v počítačem vytvořené segmentaci S označíme l_i a $l_i^{(k)}$ bude označení stejného pixelu v manuální segmentaci S_k . Předpokládáme, že existuje množina správných zařazení pixelu x_i , takovou množinu označíme \hat{l}_i . Je nutné vzít v úvahu, že neexistuje jen jedna množina správných zařazení pixelu do tříd, takových množin existuje totiž celá řada. Cílem je srovnání automatické segmentace S s množinou A a získat hodnotu měření stálosti $d(S, S_{1...K})$. Díky sadě manuálních segmentací obrazu X můžeme vypočítat pravděpodobnost vztahu zařazení do třídy mezi pixely x_i a x_j následovně:

$$\begin{aligned}\hat{P}(\hat{l}_i = \hat{l}_j) &= \frac{1}{K} \sum_{k=1}^K \mathbf{I}(l_i^{(k)} = l_j^{(k)}), \\ \hat{P}(\hat{l}_i \neq \hat{l}_j) &= \frac{1}{K} \sum_{k=1}^K \mathbf{I}(l_i^{(k)} \neq l_j^{(k)}) = 1 - \hat{P}(\hat{l}_i = \hat{l}_j).\end{aligned}\tag{2.14}$$

Pak pravděpodobnostní Rand index (= PR index) vypočteme jako:

$$PR(S, S_{\{1...K\}}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i \neq j}} \left[\mathbf{I}(l_i = l_j) P(\hat{l}_i = \hat{l}_j) + \mathbf{I}(l_i \neq l_j) P(\hat{l}_i \neq \hat{l}_j) \right].\tag{2.15}$$

Výsledné hodnoty PR indexu se opět pohybují v intervalu od nuly do jedné včetně. Hodnotu nula získáme v případě, když automatická segmentace S a sada A lidmi segmentovaných vzorků obrazu X neobsahují žádnou shodu. To znamená, že automatická segmentace S obsahuje jen jednu třídu, zatímco sada A lidmi segmentovaných vzorků obrazu X obsahuje mnoho tříd – každá třída obsahuje jen jeden pixel. Stejnou hodnotu dostaneme i v opačném případě. Nejvyšší hodnotou PR indexu je hodnota jedna. Tu dostaneme v případě, že jsou všechny segmentace shodné.

3 Markovská náhodná pole v počítačovém zpracování obrazu

3.1 Proč právě Markovská náhodná pole

Markovská náhodná pole jsou v současné době používána v úlohách počítačového zpracování obrazu. Jsou nyní nejvíce využívána v nižší úrovni počítačového zpracování obrazu, například při segmentaci obrazu, restauraci obrazu, v úlohách syntézy obrazu a v dalších úlohách počítačového zpracování obrazu[3],[5].

Markovská náhodná pole[3],[5] představují jednu z možností, na základě které lze počítačově zpracovávat obraz. Oproti jiným metodám využívá kontextuální informaci, konkrétně pixely v okolí zpracovávaného pixelu a souvislosti mezi sousedními pixely. Ve světě kolem nás není nic uspořádáno jen tak náhodou, všude je možné najít určité souvislosti.

Využití kontextu je pro analýzu v počítačovém zpracování obrazu velmi užitečné. Kontext si v digitálním obrazu můžeme představit tak, že význam jednoho pixelu je nějak závislý na významech pixelů v okolí. Například pokud v obraze patří zpracovávaný pixel určité významové oblasti, je velmi pravděpodobné, že do stejné významové oblasti patří i pixely v okolí zpracovávaného pixelu[7],[8]. Nástrojem pro modelování kontextu je využití statistických kritérií, v případě Markovských náhodných polí je nejčastěji využíváno maximální aposteriorní pravděpodobnosti.

3.2 Segmentace a zařazení pixelů do tříd

Při počítačovém zpracování obrazu řešíme problém zařazení jednotlivých pixelů do definovaných tříd[3],[5],[7]. Každý digitální obraz o rozměrech $m \times n$ obsahuje právě N pixelů. Každý pixel má svoje souřadnice (i, j) , na kterých se nachází v obraze.

$$N = m \cdot n . \tag{3.1}$$

Množinu všech pixelů v obraze označíme písmenem S .

$$S = \{1, \dots, N\}. \quad (3.2)$$

Dále pro každý počítačově zpracovávaný obraz existuje množina labelů L . Label, nebo též štítek, poskytuje absolutní informaci o klasifikaci daného pixelu. V diskrétním případě nabývá množina labelů L určitou diskrétní hodnotu z množiny o M labelech.

$$L_d = \{\ell_1, \dots, \ell_M\}. \quad (3.3)$$

Vztah lze uvést i v jednodušší podobě:

$$L_d = \{1, \dots, M\}. \quad (3.4)$$

Pro jednoduchost si uvedeme příklad množiny L_d : $L_d = \{\text{hrana, nehrana}\}$ nebo $L_d = \{\text{objekt, pozadí}\}$. V teorii Markovských náhodných polí představuje label skrytou proměnnou. Během počítačového zpracování obrazu řešíme problém přiřazení určitého labelu z množiny L každému pixelu obrazu. Každý pixel p obrazu má svůj příznakový vektor \vec{f}_p . Všechny pixely v obraze mají množinu příznakových vektorů f .

$$f = \{f_1, \dots, f_m\}. \quad (3.5)$$

Když řešíme problém přiřazení nějakého labelu z množiny L každému pixelu obrazu, hledáme vlastně zobrazení z S do L :

$$f : S \rightarrow L. \quad (3.6)$$

V teorii Markovských náhodných polí je toto zobrazení označováno jako konfigurace pole. Obrázek o rozměrem $m \times n$ obsahuje právě $L^{m \cdot n}$ možných konfigurací. Cílem je vybrat takovou konfiguraci pole, která bude z určitého hlediska nejoptimálnější.

3.3 Přístup za pomoci pravděpodobnosti

V oblasti počítačového zpracování obrazu často používáme pravděpodobnost i statistiku[5],[8]. Nejčastěji jsou řešeny problémy nad určitou sadou dat, o kterých máme k dispozici určitou informaci. Definujeme pravděpodobnostní míru ω jako podmíněnou pravděpodobnost $P(\omega | f)$, která odpovídá pravděpodobnosti konfigurace pole daného vstupního obrazu f . V takových případech můžeme použít pro nalezení nejlepší konfigurace pole ω^* , tj. pro určení maximální aposteriori pravděpodobnosti, metodu maximální věrohodnosti[5].

$$\omega^{*MAP} = \arg \max_{\omega \in \Omega} P(\omega | f). \quad (3.7)$$

Pokud známe ještě funkci hustoty pravděpodobnosti, dosáhneme nejlepších výsledků za použití Bayesova pravidla.

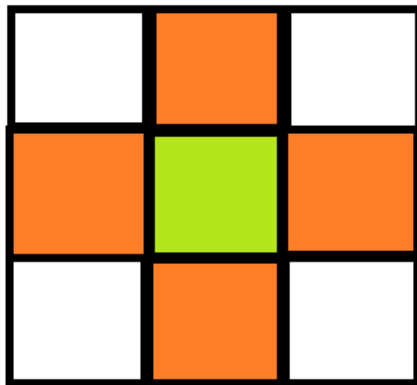
$$P(\omega | f) = \frac{p(f | \omega) P(\omega)}{p(f)}. \quad (3.8)$$

Bayesova teorie patří mezi základní teorie v oblasti rozhodování a odhadu, na problém rozhodování a odhadu pohlíží z pravděpodobnostního hlediska[5],[8]. Aposteriori pravděpodobnost je vypočtena dle vzorce (3.8). Pro výpočet aposteriori znalosti za použití Bayesova pravidla je nutná znalost hustoty pravděpodobnosti a pravděpodobnostní funkce. $P(\omega)$ označuje apriorní pravděpodobnost třídy ω , $p(f | \omega)$ je podmíněná hustotní funkce třídy ω a funkce hustoty pravděpodobnosti $p(f)$ vyjadřuje rozložení obrazů. Pokud je obraz dán, pak $p(f)$ je konstanta. Markovská náhodná pole se zabývají určením apriorní pravděpodobnosti $P(\omega)$ a určením podmíněné hustotní funkce $p(f | \omega)$.

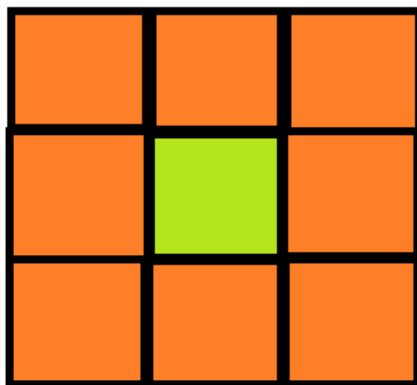
3.4 Okolí

Nejen Markovská náhodná pole používají pojem okolí. Okolím máme na mysli pixely, které sousedí se zpracovávaným pixelem. V oblasti počítačového zpracování obrazu uvažujeme nejčastěji dva typy okolí[7],[8]. Tyto typy se nazývají čtyřokolí a osmikolí.

Čtyřokolí (obr. č. 3.1) uvažuje pouze čtyři pixely sousedící se zpracovávaným pixelem. Jsou uvažovány sousedící pixely v horizontálním a vertikálním směru. Osmiokolí (obr. č. 3.2) uvažuje všechny pixely sousedící se zpracovávaným pixelem. Vedle sousedících pixelů v horizontálním a vertikálním směru uvažujeme také pixely ve všech diagonálních směrech.



Obr. č. 3.1 – Ilustrační ukázka čtyřokolí, uprostřed mřížky se nachází zpracovávaný pixel, vybarvené čtverce v okolí představují pixely, se kterými se pracuje v případě čtyřokolí



Obr. č. 3.2 – Ilustrační ukázka osmiokolí, uprostřed mřížky se nachází zpracovávaný pixel, vybarvené čtverce v okolí představují pixely, se kterými se pracuje v případě osmiokolí

3.5 Definice Markovských náhodných polí

Jestliže jsme definovali okolí zpracovávaného pixelu, můžeme definovat Markovské náhodné pole. Náhodné pole lze definovat jako graf následovně[3],[5]:

$$G \cong (V, E). \quad (3.9)$$

V je množina uzlů, $V = \{1, 2, \dots, N\}$, do které náleží pixely. E je množina hran mezi uzly. V množině hran mezi uzly E mohou být pouze takové hrany, které splňují podmínky okolí. Hrany definují sousednost mezi uzly. Každá hrana začíná u jednoho uzlu a končí u jiného uzlu. Pak platí, že:

$$(p, q) \in E \Leftrightarrow q \in N_p. \quad (3.10)$$

Každý uzel má přiřazenou jednu náhodnou proměnnou Ω_p . Tato náhodná proměnná nabývá hodnoty $\omega_p \in L$. N_p označuje množinu uzlů sousedících s uzlem p .

Abychom náhodné pole mohli nazvat Markovským náhodným polem, musí splňovat určité podmínky. Markovské náhodné pole musí splňovat dvě podmínky[5] – pozitivitu a Markovianitu.

Pozitivita:

$$P(\omega) > 0, \forall \omega \in \Omega, \quad (3.11)$$

kde Ω nazýváme prostor všech možných konfigurací pole. Jde o skupinu náhodných proměnných definovaných v rámci náhodného pole.

$$\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_m\}. \quad (3.12)$$

Každá náhodná proměnná Ω_p nabývá hodnoty ω_p z množiny možných konfigurací náhodného pole L . Pozitivita říká, že pravděpodobnost konfigurace $P(\omega)$ jakéhokoliv náhodného pole je jednoznačně určena podmíněnou pravděpodobností daného uzlu.

Markovianita:

$$P(\omega_p | \{\omega_q\}_{q \in V \setminus p}) = P(\omega_p | \{\omega_q\}_{q \in N_p}) = P(\omega_p | \omega_{N_p}). \quad (3.13)$$

Markovianita popisuje lokální charakteristiky náhodných proměnných Ω . Markovianita popisuje, že výsledná hodnota jednoho uzlu závisí jen na sousedních uzlech. V případě popisů závislostí vzdálených uzlů můžeme použít závislosti blízkých uzlů.

3.6 Hammersley-Cliffordův teorém a Gibbsova náhodná pole

V Markovských náhodných polích se snažíme určit pravděpodobnost $P(\omega)$. Určení této pravděpodobnosti je problémové. Řešení lze nalézt díky Hammersley-Cliffordově teorému[3],[5],[8].

Množinu Ω všech náhodných proměnných můžeme nazvat Gibbsovým náhodným polem[5] pouze v případě, že se řídí Gibbsovým rozložením. Gibbsovo rozložení je definováno:

$$P(\omega) = \frac{1}{Z} \times e^{-\frac{1}{T}E(\omega)}, \quad (3.14)$$

kde Z je normalizační konstanta, která nabývá hodnoty:

$$Z = \sum_{\omega \in \Omega} e^{-\frac{1}{T}E(\omega)}. \quad (3.15)$$

Ve vzorci se objevuje konstanta T . Konstantu T nazýváme teplotou a určuje, jak moc bude „špičaté“ Gibbsovo rozložení. Pokud je teplota T vysoká, jsou všechny konfigurace téměř rovnoměrně rozloženy. Pokud je teplota T naopak nízká, rozložení se pohybuje okolo minima energie. Funkci energie značíme $E(\omega)$. Hodnota energie závisí na lokální konfiguraci klíky c .

$$E(\omega) = \sum_{c \in C} V_c(\omega). \quad (3.16)$$







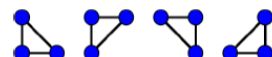

Vztah (3.16) jde rozepsat na více členů. Tyto členy odpovídají klikám jednotlivých řádů:

$$E(\omega) = \sum_{\{p\} \in C_1} V_1(\omega_p) + \sum_{\{p,q\} \in C_2} V_2(\omega_p, \omega_q) + \dots + \sum_{\{p,q,\dots\} \in C_n} V_n(\omega_p, \omega_q, \dots). \quad (3.17)$$

Klikou c v grafu $G(V, E)$ nazveme množinu, která je podmnožinou množiny V grafu G (vzorec (3.18)). Pro kliku platí, že uzel v této množině musí být sousedem zbylých uzlů této množiny.

$$\forall p, q \in c; p \neq q: p \in N_q. \quad (3.18)$$

Kliky jsou rozděleny podle počtu uzlů. Klika s jedním uzlem se nazývá singleton, klika se dvěma uzly se nazývá doubleton, klika se třemi uzly se nazývá tripleton... (viz obr. č. 3.3).

okolí	singletony	doubletony	tripletony	quadrupletony
 4-okolí			není definováno	není definováno
 8-okolí				

Obr. č. 3.3 – Ilustrační ukázka dělení klik v závislosti na počtu uzlů a v závislosti na uvažovaném okolí[3]

Vyjádření energie konfigurace pole $E(\omega)$ za pomoci potenciálů klik v grafu nám umožňuje hledat neoptimalnější konfiguraci Markovského náhodného pole. Proto používáme Gibbsovo rozložení. Problém maximalizace pravděpodobnosti $P(\omega)$ totiž přechází na problém minimalizace energie $E(\omega)$. Optimalizací problému se snažíme nalézt konfiguraci Markovského náhodného pole s co nejmenší energií.

3.7 Modely

Velmi často omezujeme uvažované okolí pouze na čtyřokolí. Děje se tak z několika důvodů. Hlavním důvodem je menší náročnost výpočtu. Omezením na čtyřokolí dojde

ke zjednodušení vzorců pro výpočet neoptimálnější konfigurace pole. V případě omezení na čtyřokolí jsou uvažovány pouze singletony a doubletony v horizontálním i vertikálním směru. Vzorec pro výpočet energie pomocí potenciálů klik v grafu se skládá pouze ze dvou členů, které odpovídají klikám prvního a druhého řádu:

$$\begin{aligned}
 E(\omega) &= \sum_{\{p\} \in C_1} V_1(\omega_p) + \sum_{\{p,q\} \in C_2} V_2(\omega_p, \omega_q), \\
 E(\omega) &= E_{data}(\omega) + E_{smoothness}(\omega).
 \end{aligned}
 \tag{3.19}$$

Jisté typy Markovských náhodných polí mohou být určeny pečlivým vybráním V_1 a V_2 . Vzorec výše si můžeme představit jako součet dvou sum. První suma, $E_{data}(\omega)$, představuje shodu konfigurace s daty. Za druhou sumou se skrývá kontext, je kladen důraz na homogenní oblasti. Existují dva druhy modelů – binární a víceúrovňový[3],[5].

Isingův model[3],[5] je příkladem binárního modelu vzhledem k počtu labelů $\rightarrow L = \{0,1\}$. Druhý člen ve vzorci, $V_2(\omega_p, \omega_q)$, ztvárňuje změny v označení labelů.

$$\begin{aligned}
 V_2(\omega_p, \omega_q) &= \beta |\omega_p - \omega_q| = \beta \delta(\omega_p, \omega_q), \\
 \delta(\omega_p, \omega_q) &= \begin{cases} 1 & \omega_p \neq \omega_q \\ 0 & \omega_p = \omega_q \end{cases},
 \end{aligned}
 \tag{3.20}$$

kde $\delta(\omega_p, \omega_q)$ je Kroneckerova delta. Parametr β určuje jak moc vysoká bude penalizace v případě změny labelů. Pokud je parametr β malý, nejsou detekované objekty příliš kompaktní. Ve výsledné segmentaci se objeví mnoho změn v označení labelů, bude zde i velký počet izolovaných ostrovů. Izolovanými ostrovy myslím jiné označení labelu než v okolí. Při zvětšujícím se parametru β budou objekty více celistvé. Můžeme říci, že parametr β dává přednost spojitým oblastem ve výsledné segmentaci.

Velkou nevýhodou Isingova modelu je právě jeho binárnost. Isingův model je velmi používaný, nicméně v případech s větším počtem různých labelů se nehodí. Pro větší počet různých labelů byl zaveden Pottsův model. Množina labelů v Pottsově modelu[3] může být velmi rozsáhlá:

$$L = \{1, 2, \dots, M\}.
 \tag{3.21}$$

Pottsův model[3] vyžaduje jinou definici členu $V_2(\omega_p, \omega_q)$, který modeluje změny v označení labelů. Člen $V_2(\omega_p, \omega_q)$ je definován následovně:

$$V_2(\omega_p, \omega_q) = \begin{cases} \beta & \omega_p \neq \omega \\ -\beta & \omega_p = \omega \end{cases} \quad (3.22)$$

Význam parametru β zůstává zachován. Parametr β opět určuje jak moc vysoká bude penalizace v případě změny labelů. Čím vyšší hodnota parametru β , tím spojitější získáme oblast ve výsledné segmentaci.

3.8 Optimalizace

Nyní máme řádně sestavený graf, vybrali jsme model a zbývá nalézt neoptimálnější konfigurace Markovského náhodného pole[5]. Hledání neoptimálnější konfigurace Markovského náhodného pole je složité, neboť funkce energie obsahuje velké množství lokálních minim. Nalezení globálního minima je výpočetně velmi náročné.

Pro nalezení neoptimálnější konfigurace pole, tj. pro hledání minim funkce energie, je k dispozici řada optimalizačních metod. Zmíním např. gradientní metody, simulované žíhání, genetické algoritmy, či v práci použitá metoda graph cut. V práci probíhaly veškeré segmentace pouze díky metodě graph cut. Proto tato metodu bude blíže popsána v dalším textu.

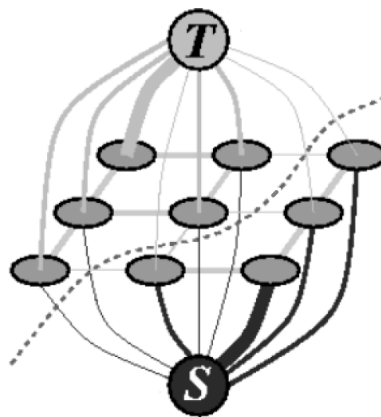
Některé metody, např. žíhání nebo náhodné hledání, prohledávají malý úsek funkce energie, jsou lokální. Je nutné ošetřit jeden významný problém – uváznutí v lokálním extrému. Tento problém lze vyřešit např. vícenásobnou inicializací metody. Hledání globálního extrémů je velmi výpočetně náročné. To samé platí i pro metody, které hledají globální minimum funkce energie. Například metoda graph cut využívající algoritmy α -expansion a α - β swap zaručuje, že nalezne takový extrém, který je v nejhorším případě pouze c -krát horší, než globální extrém. Konstanta c je předem známa.

3.9 Graph cut

Metoda graph cut[3],[5] je v současnosti hojně používána v oblasti počítačového zpracování obrazu. Je používána v úlohách segmentace obrazů, restaurace, syntézy či v úlohách stereovidení. V podstatě se jedná o skupinu algoritmů, které hledají minimální řez grafem. Existují dvě skupiny algoritmů. První skupina algoritmů je založena na hledání řezu grafem na základě maximálního toku. Druhá skupina algoritmů využívá velkých změn, tzv. large moves.

3.10 Maximální tok (minimální řez)

Při hledání minimálního řezu grafem[5] vyjdeme opět z grafu. Sestavíme orientovaný graf $G(V, E)$. E je množina hran, váhy všech hran jsou nezáporné. V představuje množinu uzlů. Množina uzlů V obsahuje v případě hledání řezu grafem dva další uzly. Tyto uzly se nazývají source (označení S) a sink (označení T) – viz obr. 3.4. Potom řez E_c grafem G je podmnožina množiny hran E , která splňuje následující podmínky. Podgraf grafu bez E_c , tj. $\tilde{G} = (V, E - E_c)$, je nespojitý. Pokud přidáme do podmnožiny grafu $\tilde{G} = (V, E - E_c)$ jakoukoliv hranu z E_c , tj. $\tilde{G} = (V, (E - E_c) \cup \{e\})$; ($\forall e \in E_c$), je tento graf souvislý. Celkové náklady na řez grafem E_c jsou sumou vah hran řezu E_c .



Obr. č. 3.4 – Ilustrační ukázka sestaveného grafu pro hledání minimálního řezu grafem s naznačeným řezem grafem, S je uzel source a T je uzel sink[11]

Problém hledání minimálního řezu grafem spočívá v nalezení vhodného kritéria. Kritérium pro hledání řezu můžeme definovat jako:

$$C(L) = \lambda \cdot R(L) + B(L). \quad (3.23)$$

Kriteriální funkce obsahuje dvě části. Konstanta λ má funkci váhy. Váží vliv oblasti a okraje na výsledné kritérium. $R(L)$ představuje váhy oblasti.

$$R(L) = \sum_{p \in P} R_p(L_p). \quad (3.24)$$

$B(L)$ bere v potaz okraje segmentace. Penalizuje nespojitosti, největší penalizaci obdrží izolované pixely, tj. pixely s jiným label než je v okolí.

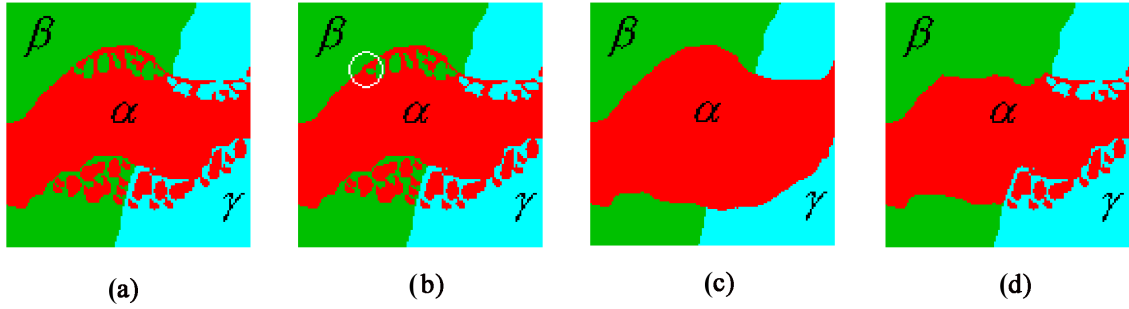
$$B(L) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(L_p, L_q),$$

$$\delta(L_p, L_q) = \begin{cases} 1 & L_p \neq L_q \\ 0 & L_p = L_q \end{cases}. \quad (3.25)$$

Cílem je najít takový řez, který bude mít co nejmenší náklady.

3.11 Large moves

Metodu Graph cut je možné použít i na problémy segmentace do více tříd. Cílem je opět najít co nejoptimálnější konfiguraci pole z pohledu energie. Mnohé algoritmy mění v jednom kroku labely pouze dvou pixelů. Tyto změny byly nazvány standard moves [11]. Algoritmy založené na velkých změnách, tzv. large moves[5],[11], dokáží v jednom kroku změnit labely velkého množství pixelů. Takové algoritmy dosahují v počítačovém zpracování obrazu lepších výsledků.



Obr. č. 3.5 – Ilustrační ukázka práce algoritmů založených na large moves. Obrázek (a) ukazuje segmentovaný obraz. Obrázky (c) a (d) ukazují výsledky algoritmů α -expansion (c) a α - β swap (d), které v jednom kroku změny labely velkému množství pixelů. Algoritmy založené na standard moves změny v jednom kroku přiřazení pouze jednomu pixelu.[11]

Vydeme ze vzorce pro energii potenciálů klik prvního a druhého řádu v grafu:

$$E(\omega) = \sum_{\{p\} \in C_1} V_1(\omega_p) + \sum_{\{p,q\} \in C_2} V_2(\omega_p, \omega_q). \quad (3.26)$$

Problém nalezení nejoptimálnější konfigurace pole řeší algoritmy α -expansion a α - β swap.

3.11.1 α -expansion

Algoritmus α -expansion[5],[11] předpokládá, že druhý člen výrazu energie $E_{smoothness}(\omega)$ představující hladkost oblastí je měřitelný. Pokud je tento člen měřitelný, pak musí splňovat určité podmínky.

$$V_2(\omega_p, \omega_q) \geq 0, \quad (3.27)$$

$$V_2(\omega_p, \omega_q) = 0 \Leftrightarrow \omega_p = \omega_q, \quad (3.28)$$

$$V_2(\omega_p, \omega_q) = V_2(\omega_q, \omega_p), \quad (3.29)$$

$$V_2(\omega_p, \omega_q) \leq V_2(\omega_p, \omega_r) + V_2(\omega_r, \omega_q). \quad (3.30)$$

V každé iteraci algoritmu existuje $\alpha \in L$ a množina labelů o dvou prvcích $\{\alpha, non\alpha\}$. V algoritmu je řešen problém přiřazení labelů metodou graph cut do dvou tříd. Algoritmus rozhoduje, zda pixel i , $i \in S$, $\omega_i \neq \alpha$, přeřadit z třídy α do třídy $non\alpha$. Algoritmus probíhá tak dlouho, dokud energie klesá. Algoritmus α -expansion je schopen najít takové minimum, které je nejhůře c -krát horší než globální minimum. Konstanta c je předem známa. Pseudokód algoritmu α -expansion[5]:

Begin

1. Initialize f with an arbitrary labeling;
2. Flag: = *false*;
3. For each label $\alpha \in L$;
 - a. Find $\hat{f} = \arg \min(E(f'))$ among f' within one α expansion of f ;
 - b. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and Flag: = *true*;
4. If Flag = *true* goto step 2
5. Return f ;

End

3.11.2 α - β swap

α - β swap[5],[11] je iterační algoritmus metody Graph cut. Algoritmus předpokládá, že druhý člen výrazu energie $E_{smoothness}(\omega)$ představující hladkost oblastí je semimetrický. To znamená, že platí následující podmínky.

$$V_2(\omega_p, \omega_q) \geq 0, \quad (3.31)$$

$$V_2(\omega_p, \omega_q) = 0 \Leftrightarrow \omega_p = \omega_q, \quad (3.32)$$

$$V_2(\omega_p, \omega_q) = V_2(\omega_q, \omega_p). \quad (3.33)$$

V každé iteraci algoritmu jsou vybrány dva labely, $\alpha, \beta \in L$. Dále definujeme dvě podmnožiny $S_\alpha = \{i \in S \mid \omega_i = \alpha\}$ a $S_\beta = \{i \in S \mid \omega_i = \beta\}$. Algoritmus určí dvě podmnožiny

$\hat{S}_\alpha \subset S_\alpha$ a $\hat{S}_\beta \subset S_\beta$, dále dojde k prohození labelů mezi těmito podmnožinami. Algoritmus skončí, pokud hodnota funkce energie neklesá.

Algoritmus α - β swap negarantuje nalezení minima, které je nejhůře c -krát horší než globální minimum jako v případě algoritmu α -expansion.

Pseudokód algoritmu α - β swap[5]:

Begin

1. Initialize f with an arbitrary labeling;
2. Flag: = *false*;
3. For each pair $\{\alpha, \beta\} \in L$;
 - a. Find $\hat{f} = \arg \min(E(f'))$ among f' within one $\alpha - \beta$ swap of f ;
 - b. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and Flag: = *true*;
4. If Flag = *true* goto step 2
5. Return f ;

End

4 Praktická část

4.1 Vyhlazení histogramu

Histogramy obecně obsahují velké množství šumu. Za šum v histogramu považujeme velké množství lokálních nevýznamných extrémů. Tyto lokální nevýznamné extrémy velmi znesnadňovaly další práci. V zašuměném histogramu bylo velmi obtížné stanovit informačně nejdůležitější lokální extrémy, na základě kterých byly určovány jasové modely pro segmentaci. Z důvodu velkého zašumění histogramu jsem přikročil k vyhlazení histogramu.

K vyhlazení histogramu jsem použil konvoluci histogramu s Gaussovským jádrem[3]. Pro vyhlazení histogramu je nutné brát v úvahu velikost jádra. Velikost jádra totiž velmi zásadně ovlivňuje výsledný vyhlazený histogram. Příliš malé jádro nepotlačí nevýznamné lokální extrémy, naopak příliš velká jádra signál příliš vyhladí. Z příliš vyhlazeného histogramu není možné určit informačně nejvýznamnější lokální extrémy. Tyto teoretické poznatky jsem si ověřil během volby velikosti Gaussovského jádra pro vyhlazení histogramu. Experimentálně jsem došel k následujícím závěrům. Při velikosti jádra menší než tři nedošlo k potlačení nevýznamných lokálních extrémů. Při velikosti jádra šest a více došlo k přílišnému vyhlazení signálu, informačně nejvýznamnější maxima byla již potlačena. Neoptimálnější velikost jádra pro pozdější vytváření jasových modelů byla čtyři, nebo pět. Při této velikosti jádra došlo k dostatečnému potlačení nevýznamných lokálních extrémů, informačně nejdůležitější lokální extrémy byly zachovány. Během experimentů jsem pracoval s velikostí jádra pět.

4.2 Detekce maxim

Po vyhlazení histogramu jsem přistoupil k automatické detekci maxim ve vyhlazeném histogramu. Cílem automatické detekce maxim ve vyhlazeném histogramu bylo nalézt všechna informačně významná maxima. Informačně významné maximum v histogramu nemusí být nutně maximum s největší četností jasů. Takové maximum s například třetinovou četností jasů oproti největšímu maximu v histogramu může nést také velké množství

informace potřebné pro vytvoření jasových modelů. Nalezená maxima dále sloužila k vytvoření jasových modelů pro segmentační metodu Graph cut.

Maxima v histogramu jsem detekoval za pomoci funkce `argrextrema` z balíku SciPy programovacího jazyka Python. Funkce detekuje extrém tak, že porovnává hodnoty dvou po sobě jdoucích prvků signálu. Argumentem `'<'` a `'<='` volíme detekci maxim. Pokud jsem zadal ve funkci parametr pro nalezení extrémů `'<'`, detekoval jsem pouze ostrá maxima. Částečné zlepšení poskytl argument funkce `'<='`. Při tomto argumentu byly detekovány i některé body, které nepatří mezi maxima, či více hodnot v okolí maxima. Podmínkami jsem ošetřil, aby byla detekována pouze maxima. Dalšími podmínkami jsem ošetřil detekci pouze jednoho maxima v určitém okolí. Více o nevýhodách funkce `argrextrema` v dalším odstavci.

4.3 Nevýhody funkce `argrextrema`

Funkce `argrextrema` se nachází v knihovně SciPy v balíku `signal` programovacího jazyka Python. Funkce jsem použil pro detekci lokálních maxim ve vyhlazeném signálu. Tato funkce má ovšem i svoje nevýhody. Funkce s argumentem pro detekci maxim dokáže najít pouze ostrá lokální maxima. Neostrá lokální maxima nebyla touto funkcí detekována. Neostrá lokální maxima přitom není možné vyloučit. Určité zlepšení situace jsem očekával při zadání argumentu `'<='`. Při zadání tohoto argumentu by mělo dojít k detekci ostrých i neostrých maxim. V některých případech proběhla detekce lokálních maxim bez problémů, v dalších případech byly detekovány body vedle lokálních minim.

Špatně detekované body bylo nutné vyloučit, čehož jsem dosáhl sepsáním vlastní metody. Podmínkami jsem ošetřil, aby funkce `argrextrema` s argumentem `'<='` detekovala pouze maxima. K detekování hodnot v okolí minim docházelo vždy v případě, že se třikrát opakovala stejná hodnota a za tímto opakováním následoval ještě pokles hodnot. Funkce `argrextrema` vyhodnocuje přítomnost maxima se zadanou podmínkou větší rovno přihlédnutím na jednu hodnotu před a za detekovaným bodem. Podmínkou jsem ošetřil, aby funkce zohledňovala hodnoty o dvě před i za detekovaným bodem. Rozdílem detekované hodnoty bodu a hodnoty bodu o dvě před i od dvě za detekovaným bodem jsem získal informaci, zda je na detekovaném bodu maximum. Pokud byl rozdíl aspoň na jedné straně záporný, vyloučil jsem takto detekovaný bod.

Podmínkami jsem ošetřil i případy, že došlo k vícerému detekování bodů v maximu a jeho okolí. Pokud bych tento případ neošetřil, zhoršil bych kvalitu výsledné segmentace metodou Graph cut. Okolo bodu, na kterém bylo detekováno maximum, jsem vytvořil okolí, ve kterém nesmí být detekováno další maximum. Pokud se tak stane, musí být toto další detekované maximum vyloučené. Velikost okolí jsem nastavoval na základě výsledků detekce maxim funkcí argrextrema s argumentem větší rovno. Pokud došlo k vícerému detekování bodů v okolí maxima, bylo to vždy do vzdálenosti pěti hodnot od maxima na obě strany. Proto jsem zvolil velikost okolí pět.

4.4 Automatické vytvoření jasových modelů

Po detekování maxim ve vyhlazeném histogramu jsem mohl přistoupit k automatickému vytvoření jasových modelů, které potřebuje metoda Graph cut při segmentaci. Modely byly vytvořeny aproximací Gaussovou funkcí v bodě, ve kterém bylo detekováno maximum. Gaussova funkce vyžaduje dva parametry – střední hodnotu a rozptyl.

Střední hodnotu Gaussovy funkce představuje bod, ve kterém bylo detekováno maximum. Větší problém pro správnou aproximaci Gaussovou funkcí představuje správné stanovení rozptylu. Je nutné zohlednit okolí maxima, např. pokud je na hodnotách v okolí detekovaného maxima v histogramu velké skoková změna. Rozptyl jsem určoval z hodnot histogramu v okolí maxima. Tento přístup při určení rozptylu se ukázal jako vhodný, Gaussova funkce přibližně kopírovala vzhled maxima ve vyhlazeném histogramu. Bylo nutné vyřešit problém určení správné velikosti okolí maxima. O řešení tohoto problému pojednávají další odstavce.

4.5 Skóre funkcí aic a bic[5]

Funkce aic a bic se nachází v knihovně scikit-learn v balíku mixture programovacího jazyka Python. Cílem použití obou těchto funkcí bylo ohodnocení velikosti okolí okolo střední hodnoty jasového modelu. Obě funkce vrací hodnotu skóre nad pixely dané sady jasových modelů. Skóre je vypočítáno na základě podobnosti dat sady jasových modelů se statisticky získaným modelem. Obě funkce jsou si velmi blízké, vzájemně se liší

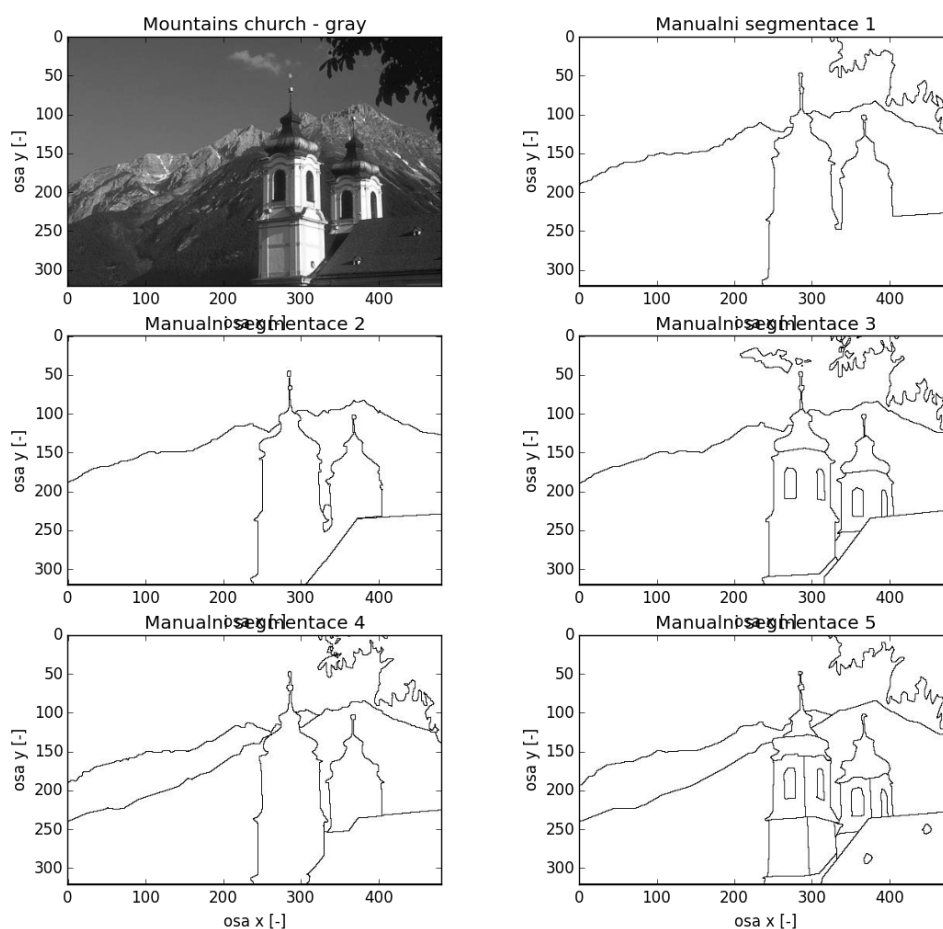
ohodnocením špatných případů. V případě funkce bic je penále za špatný případ větší. U hodnot skóre obou funkcí platí, že čím nižší je skóre, tím lepší je shoda.

Experimentálně jsem určoval skóre těchto funkcí pro sadu jasových modelů. Sledoval jsem změnu skóre v závislosti na velikosti okolí okolo střední hodnoty jasového modelu. Získal jsem hodnoty skóre funkcí aic i bic pro hodnoty okolí v intervalu pět až devadesát úrovní. Výsledky skóre obou funkcí se od sebe příliš nelišily. V obou případech však byly hodnoty skóre příliš velké. Hodnoty se pohybovaly v řáde desítek milionů až desítek miliard. Z výsledků jsem vyzoroval, že čím větší okolí střední hodnoty jasového modelu bereme v úvahu, tím nižší skóre. Z experimentů vyplynulo, že funkce aic a bic nejsou pro ohodnocení velikosti okolí střední hodnoty jasového modelu vhodné.

4.6 Berkeley Segmentation Data Set[9][10]

Pro stanovení rozptylu jasového modelu jsem vycházel z okolí střední hodnoty jasového modelu. Bylo nutné zjistit, jak velké okolí střední hodnoty je třeba uvažovat. Velikost okolí zásadně ovlivňuje výsledek segmentace. Při příliš malém okolí ztrácíme část informace, která je nutná pro jasový model, naopak při příliš velkém okolí uvažujeme i část informace jiného jasového modelu.

Pro stanovení neoptimálnější velikosti okolí bylo nutné experimentovat. Pro experimenty, obzvláště pro experimenty s metodou Probabilistic Rand Index[6], jsem využil sady obrázků pro segmentaci [9],[10], které jsou k dispozici na stránkách univerzity Berkeley v Kalifornii v USA. Jedná se o velmi různorodou sadu obrázků a vzorků lidmi manuálně segmentovaných obrázků (viz obr. 4.1). Na vzorcích manuálně segmentovaných obrazů můžeme názorně pozorovat, že každý vnímáme obraz trochu odlišně. Někteří účastníci experimentů nasegmentovali obraz jen na předmět zájmu a okolí, jiní účastníci experimentů zaběhli více do detailů.



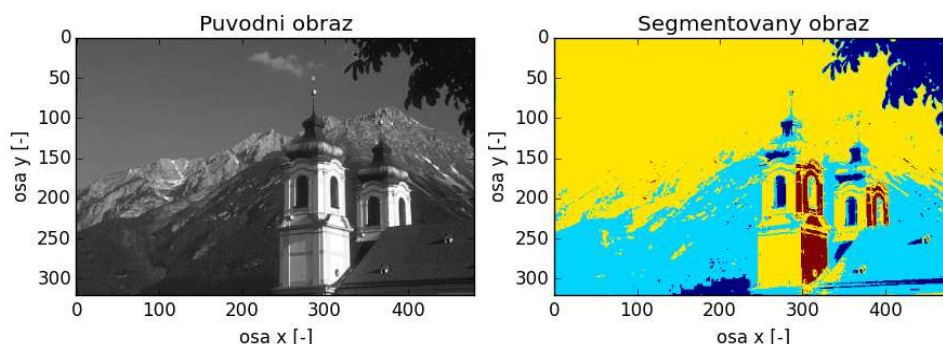
Obr. č. 4.1 – Obrázek Mountains church ze sady Berkeley Segmentation Data Set[9] a lidmi vytvořené manuální segmentace

4.7 Pokusy s PR indexem

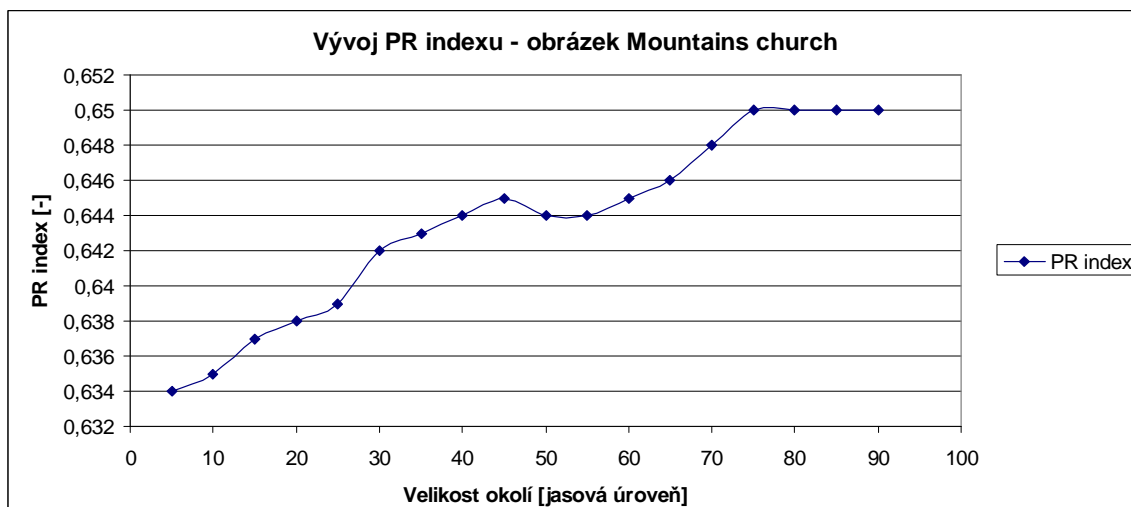
K pokusům s PR indexem[6] jsem přikročil z důvodu stanovení neoptimálnější okolí pro práci s jasovými modely. Experimenty jsem prováděl nad sadou obrázků Berkeley Segmentation Data Set[9]. Na části sady obrázků jsem počítal hodnotu PR indexu. Pro výpočet PR indexu byly využity lidmi manuálně nasegmentované vzorky obrázků. Tyto manuálně nasegmentované obrázky byly srovnávány s automatickou segmentací téhož obrázku metodou Graph Cut. Byl sledován vliv velikosti okolí střední hodnoty jasového modelu na výsledné hodnoty PR indexu. Předpokládal jsem, že při malém okolí bude hodnota PR indexu malá. Se zvětšujícím se okolím bude hodnota PR indexu narůstat. Očekával jsem,

že nejvyšších hodnot dosáhne hodnota PR indexu mezi hodnotami velikosti okolí 30 a 40. Větší hodnoty okolí jasového modelu uvažují i jasové úrovně jiných jasových modelů a dojde tak k zanesení chyby. Při větším okolí bude hodnota PR indexu klesat. Experimenty jsem došel k následujícím závěrům.

Nejčastějším jevem byla zvětšující se hodnota PR indexu v závislosti na zvětšujícím se okolí střední hodnoty daného jasového modelu. Názorně tento případ vývoje PR indexu ukazují na obrázku Mountains church (obr. č. 4.2 a 4.3). Pro tento případ vývoje hodnoty PR indexu je nutné zmínit, že v případě obrázku Man With Hat došlo ke skokové změně hodnoty PR indexu. Skoková změna nastala mezi okolím o velikosti 35 a okolím o velikost 40 úrovní. Hodnota PR indexu se skokově zvýšila z hodnoty 0,398 na hodnoty 0,693.

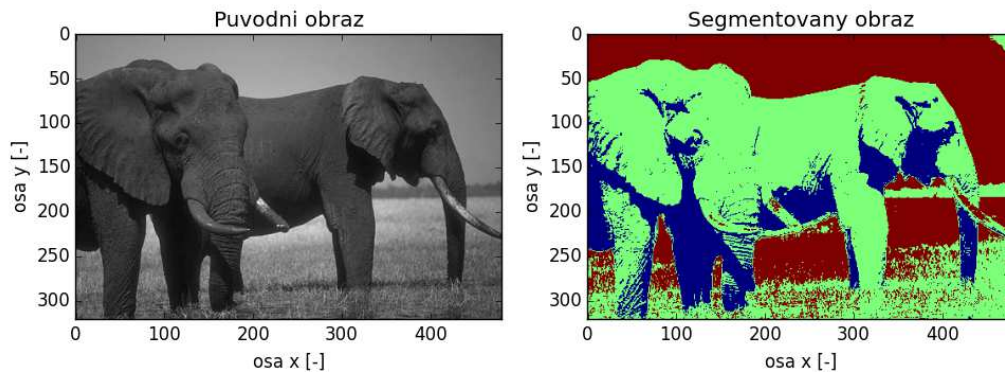


Obr. č. 4.2 – Obrázek Mountains church ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 30 (tmavě modrá), 60 (světle modrá), 100 (žlutá), 255 (tmavě červená)

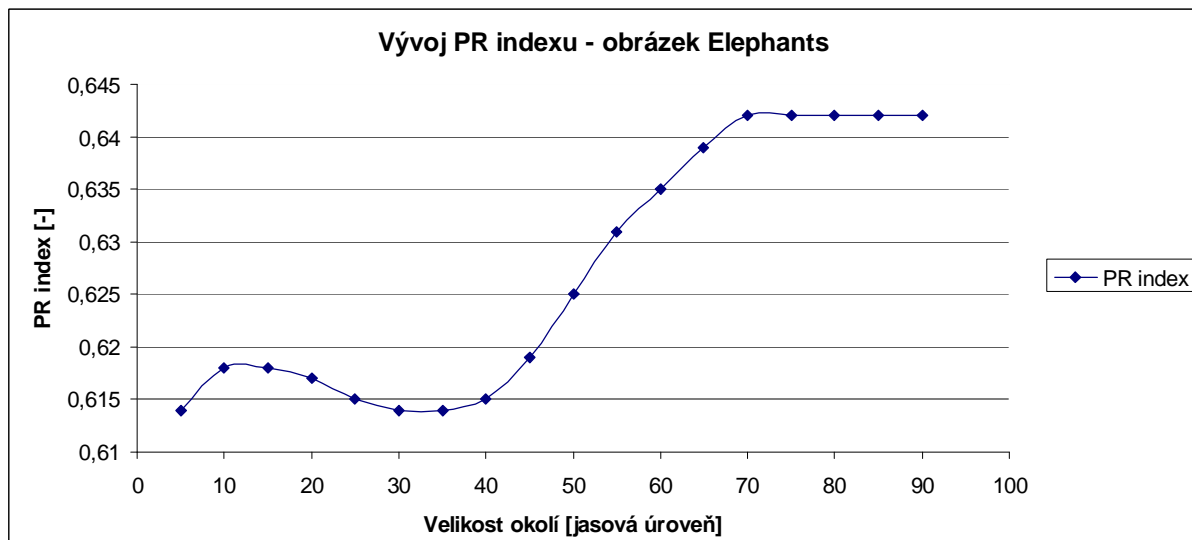


Obr. č. 4.3 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Mountains Church

V části případů hodnota PR indexu narůstala se zvětšujícím se okolím s vytvořením jednoho malého lokálního maxima hodnoty PR indexu. Při velikosti okolí od 15 do 25 docházelo k výskytu menšího lokálního maxima hodnoty PR indexu, s dále se zvětšujícím okolím hodnota PR indexu neustále stoupala. Jako názornou ukázkou tohoto případu ukazují vývoj hodnoty PR indexu na obrázku Elephants (viz obr. č. 4.4 a 4.5).



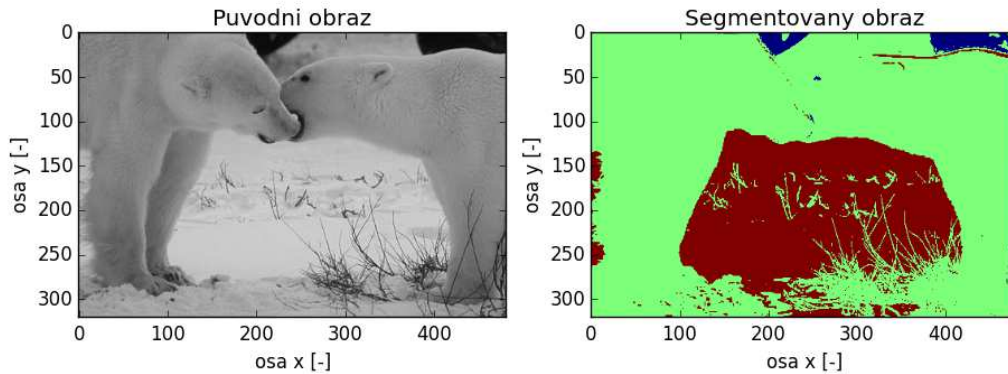
Obr. č. 4.4 – Obrázek Elephants ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 38 (modrá), 75 (zelená), 139 (tmavě červená)



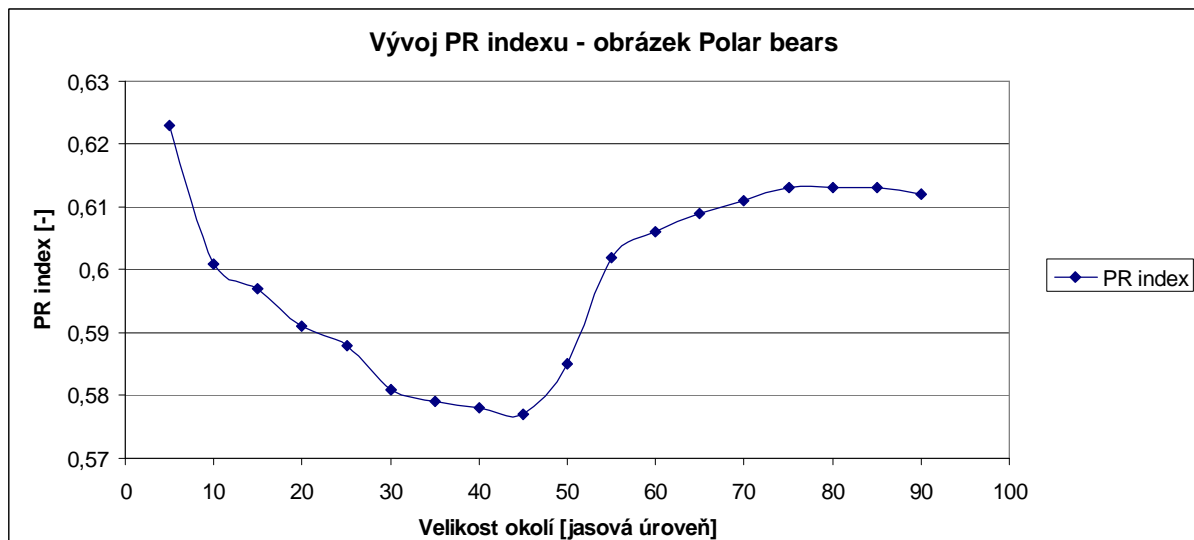
Obr. č. 4.5 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Elephants

Časté byly i případy, ve kterých byl průběh hodnot PR indexu v závislosti na okolí následující. Při velmi malém okolí (okolí 5 úrovní) byla hodnota PR velká, při zvětšujícím se

okolí nejdříve hodnota PR indexu klesala. V některých případech došlo k opětovnému nárůstu hodnoty PR indexu při okolí 50 úrovní a větším. Názorně tento případ dokumentuje obrázek Polar Bears (obr. 4.6 a 4.7).

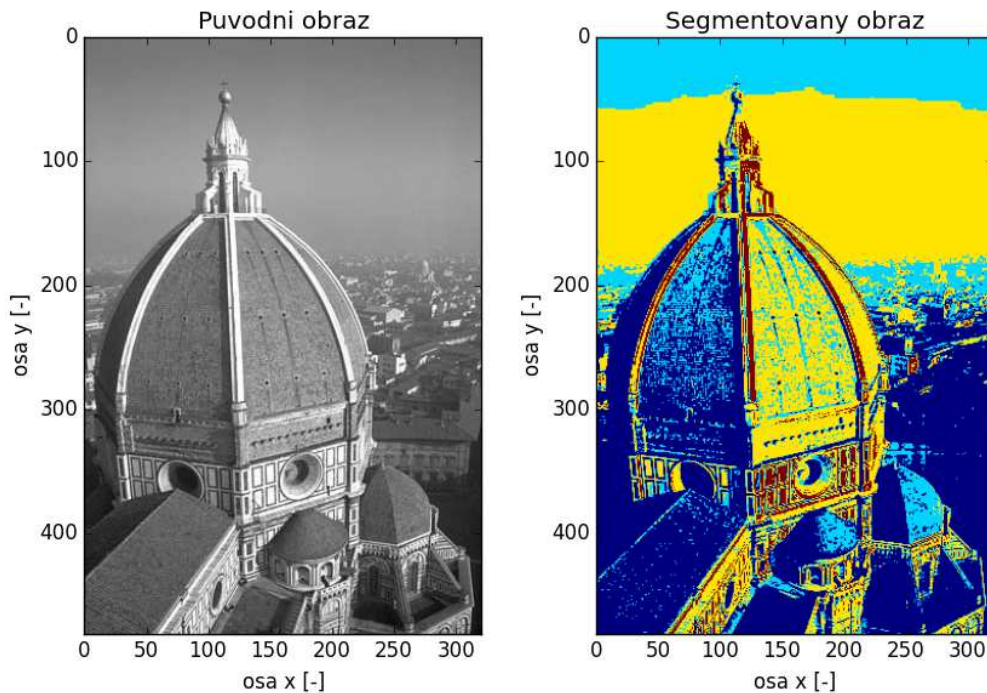


Obr. č. 4.6 – Obrázek Polar bears ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 13 (modrá), 125 (zelená), 182 (tmavě červená)

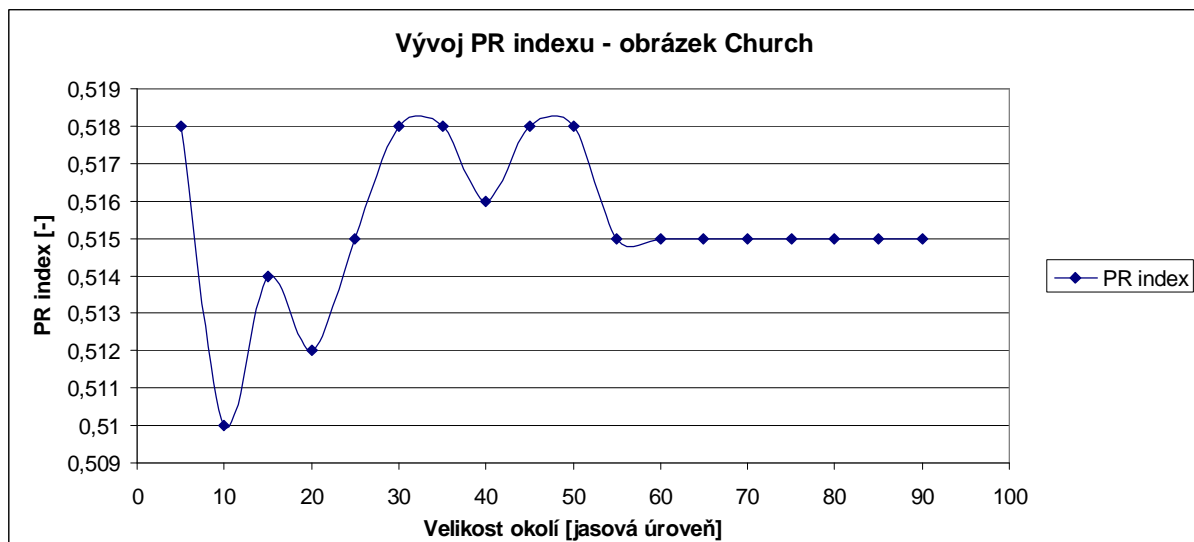


Obr. č. 4.7 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Polar Bears

Předpokládaný vývoj hodnoty PR indexu jsem pozoroval v minimálním počtu případů. Důkazem toho je obrázek Church (viz obr. 4.8 a 4.9), kde k předpokládanému vývoji hodnoty PR indexu došlo od velikosti okolí 15 úrovní.



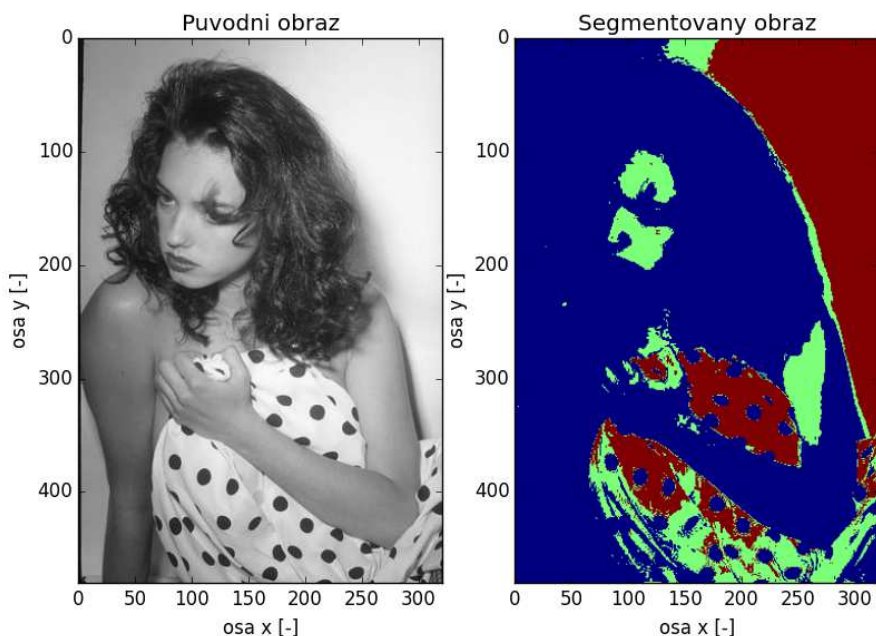
Obr. č. 4.8 – Obrázek Church ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 97 (tmavě modrá), 123 (světle modrá), 155 (žlutá), 255 (tmavě červená)



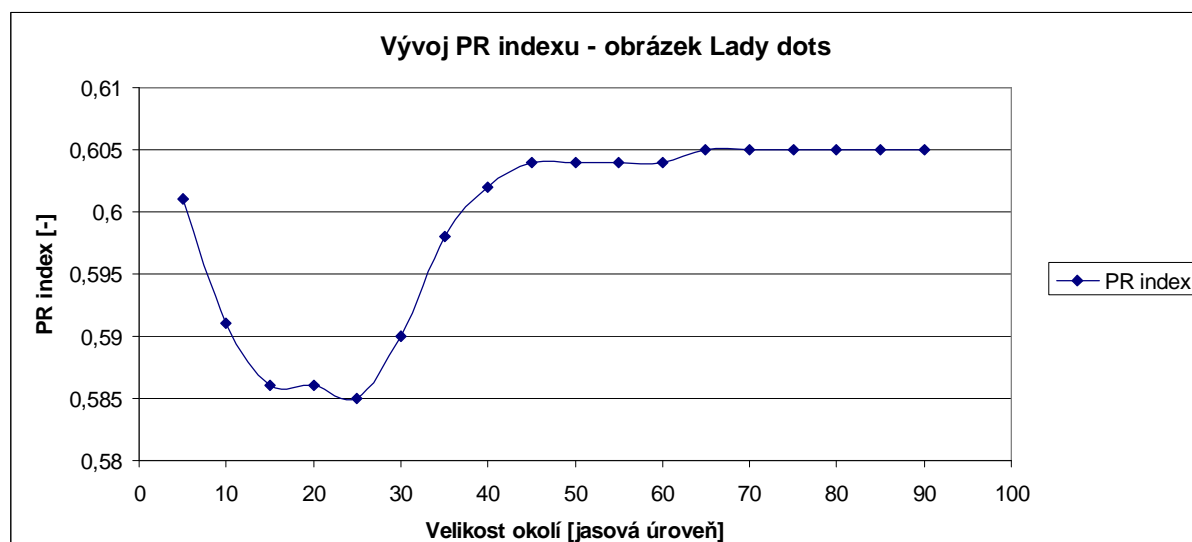
Obr. č. 4.9 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Church

Za zmínku stojí i následující dva případy. V minimálním počtu případů byl zaznamenán vyrovnaný průběh hodnoty PR indexu v závislosti na okolí. Hodnota PR indexu se s různou velikostí okolí měnila jen nepatrně. Tento jev jsem pozoroval v případě obrázku

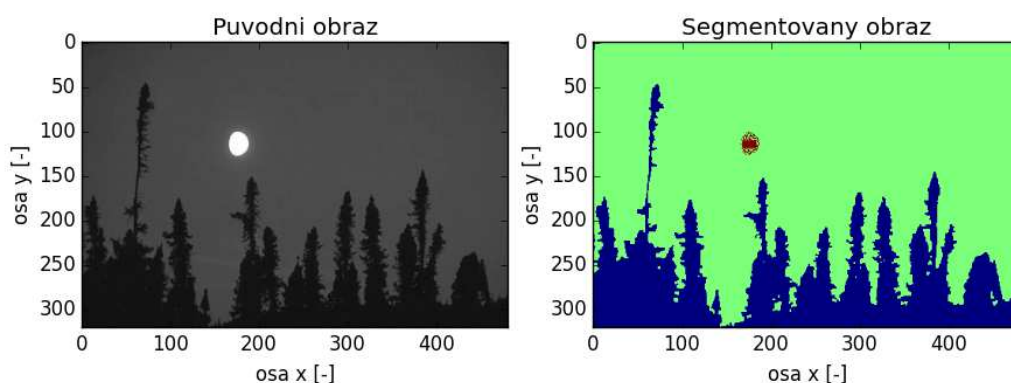
Lady Dots (obr. 4.10 a 4.11). Zajímavý byl i další průběh hodnoty PR indexu v závislosti na okolí. Tento průběh jsem zaznamenal jen v jednom případě. Jednalo se o obrázek Moon (obr. 4.12 a 4.13). Hodnota PR indexu se při zvětšujícím se okolí skoro pořád klesala. Nejvyšší hodnotu PR indexu v závislosti na okolí jsem v tomto případě zaznamenal při velikosti 5 úrovní.



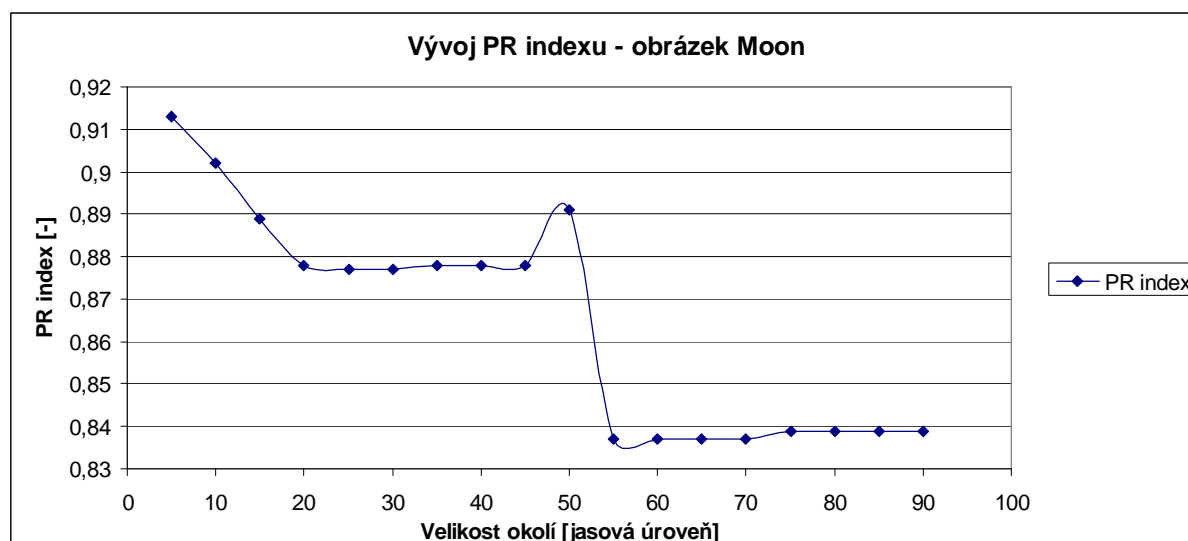
Obr. č. 4.10 – Obrázek Lady dots ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 158 (modrá), 195 (zelená), 238 (tmavě červená)



Obr. č. 4.11 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Lady Dots



Obr. č. 4.12 – Obrázek Moon ze sady Berkeley Segmentation Data Set[9] vlevo, vpravo segmentace obrázku metodou Graph cut, uvažovaná velikost okolí je 30, $\alpha, \beta = 1$, střední hodnoty jasových modelů se nachází na jasových úrovních 27 (modrá), 79 (zelená), 255 (tmavě červená)



Obr. č. 4.13 – Hodnoty PR indexu segmentace metodou Graph Cut v závislosti na velikosti okolí, obrázek Moon

Během experimentů byly zjištěny různé hodnoty PR indexu. Zaznamenal jsem hodnoty PR indexů od 0,300 po 0,980. Tyto hodnoty PR indexu značí, jak moc významově odpovídala automatická segmentace metodou Graph Cut manuální segmentacím. Ve větším počtu případů byly zaznamenány výsledky nad 0,600, což lze brát jako velmi dobrý výsledek. V menším počtu případů byly zaznamenány horší výsledky.

4.8 Závěry k určování velikosti okolí

Experimentálně jsem zjistil, že určení optimálního okolí maxima pro stanovení rozptylu je skoro nemožné. Správné určení velikosti okolí maxima velmi závisí na vstupních datech. Liší se případ od případu.

Velikost uvažovaného okolí maxima velmi ovlivňuje hodnotu rozptylu. Pokud budeme uvažovat pouze malé okolí maxima, ztratíme velkou část informace potřebné k vytvoření jasového modelu. Při malém nebo žádném uvažovaném okolí maxima nejsme schopni určit správně hodnotu rozptylu Gaussovy funkce tak, aby výsledná Gaussova funkce správně aproximovala vzhled maxima a jeho okolí ve vyhlazeném histogramu. V opačném případě – pokud budeme uvažovat příliš velké okolí maxima, zaneseme do hodnoty rozptylu šum. Za šum zde považuji, že v hodnotě rozptylu se promítnou i vzdálenější hodnoty. Tyto vzdálenější hodnoty způsobí, že výsledná hodnota rozptylu bude mít takový vliv na Gaussovu funkci, že výsledná Gaussova funkce nebude aproximovat skutečný vzhled maxima ve vyhlazeném histogramu. Tím pádem bude Gaussova funkce zkreslená.

4.9 Porovnání

Metodu Graph cut jsem porovnal s obdobnými algoritmy, které k problému segmentace přistupují na základě konstrukce grafu a následného rozdělení sestrojeného grafu. Graph cut jsem porovnal s Felzenszwalbovým algoritmem[12] a s algoritmem Normalized cut[13]. Jako výchozí parametry metody Graph cut při porovnání s podobnými metodami jsem zvolil velikost okolí 30, parametry α i β byly rovny jedné. Výsledky všech metod jsem porovnával za pomoci PR indexu[6] na obrázcích sady Berkeley Segmentation Data Set[9].

4.10 Felzenszwalbův algoritmus

4.10.1 Krátký popis algoritmu

Felzenszwalbův algoritmus[12] přistupuje k problému segmentaci za pomoci konstrukce grafu a následného rozdělení zkonstruovaného grafu do oblastí. Nejdříve je zkonstruován neorientovaný graf $G = (V, E)$. V je množina uzlů grafu představující pixely obrazu, E je množina hran grafu. Hrany jsou uvažovány vždy jen mezi dvěma sousedícími uzly. Každá hrana z množiny E je ohodnocena vahou w , která je větší než nula. Cílem segmentace S je rozdělení grafu G do určitých hranami propojených oblastí $G' = (V, E')$, $E' \subseteq E$. Dále je zavedena funkce D , která vyhodnocuje, zda má v daném místě dojít k vytvoření hranice mezi oblastmi v segmentovaném obrazu.

Definujme vnitřní rozdíl podgrafu $C \subseteq V$ jako největší váhu v minimálně se větvcím stromu podgrafu. Minimálně se větvcí strom podgrafu, tj. strom, který spojí všechny uzly v podgrafu tak, aby celková váha hran byla minimální, označme jako $MST(C, E)$. Potom vnitřní rozdíl podgrafu je definován:

$$Int(C) = \max_{e \in MST(C, E)} w(e). \quad (4.1)$$

Dále definujeme rozdíl mezi dvěma podgrafy $C_1, C_2 \in V$ jako minimální váhu hrany spojující dva podgrafy.

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j). \quad (4.2)$$

Pokud neexistuje spojení mezi podgrafy C_1 a C_2 , je rozdíl mezi dvěma podgrafy rovný nekonečnu, $Dif(C_1, C_2) = \infty$. Rozdíl mezi dvěma podgrafy $Dif(C_1, C_2)$ je vždy relativně větší než vnitřní rozdíl podgrafu, ať už rozdíl vnitřního podgrafu $Int(C_1)$, nebo vnitřní rozdíl druhého podgrafu $Int(C_2)$. Funkce D je binární a nabývá hodnot:

$$D(C_1, C_2) = \begin{cases} true & \text{pokud } Dif(C_1, C_2) > MInt(C_1, C_2) \\ false & \text{jinak} \end{cases}, \quad (4.3)$$

kde $MInt(C_1, C_2)$ je minimální vnitřní rozdíl.

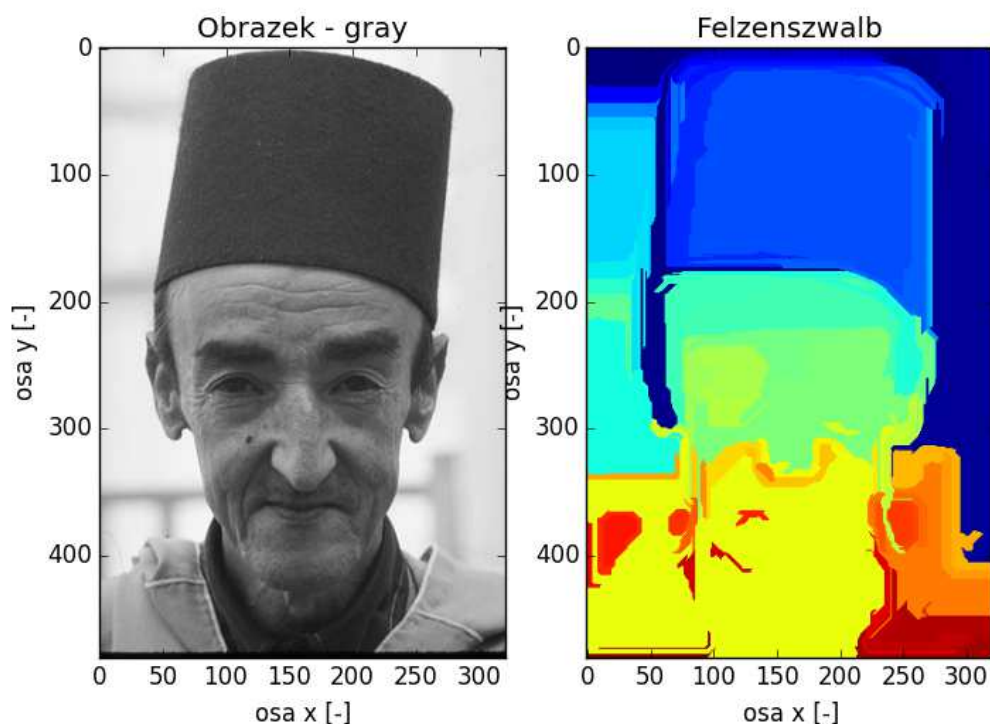
$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)), \quad (4.4)$$

kde τ plní funkci prahu. Tato prahová funkce kontroluje a porovnává velikosti vnitřních rozdílů podgrafů s rozdílem mezi podgrafy tak, aby mohla být zřízena nová hranice mezi oblastmi v segmentovaném obrazu. Prahovou funkci můžeme vyjádřit jako:

$$\tau(C) = \frac{k}{|C|}, \quad (4.5)$$

kde $|C|$ je velikosti podgrafu C , k je volená konstanta. Pokud zvolíme k malé, požadujeme větší důraz na vytváření nových hranic mezi podgrafy, ve výsledné segmentaci bude méně oblastí.

Pro samotný algoritmus předpokládáme, že vždy existuje nějaká segmentace vstupního obrazu. Algoritmus probíhá v několika krocích. Nejdříve je načten vstupní obrázek a zkonstruován graf $G = (V, E)$ o n uzlech a m hranách. Graf $G = (V, E)$ je vstupem algoritmu, výstupem algoritmu je segmentace S , $S = (C_1, \dots, C_r)$ množiny V . Dále jsou vzestupně uspořádány prvky množiny hran E podle váhy, $\pi = (o_1, \dots, o_m)$. V dalším kroku předpokládáme segmentaci S^0 , kde každý uzel množiny V představuje jeden podgraf. Následující krok je opakován, dokud není nalezena neoptimálnější segmentace. Vytvoř segmentaci S^q , která vychází se segmentace S^{q-1} následovně. Zvolme uzly $v_i, v_j \in V$, které jsou spojeny q -tou hranou z uspořádané množiny hran podle velikosti vah $o_q = (v_i, v_j)$. Pokud jsou uzly $v_i, v_j \in V$ v rozdílných podgrafech v segmentaci S^{q-1} a vnitřní rozdíl obou podgrafů je malý, spoj tyto podgrafy. Pokud tomu tak není, nic nedělej. Toto lze napsat i jinak – necht' podgraf C_i^{q-1} je složkou segmentace S^{q-1} obsahující uzel v_i a necht' podgraf C_j^{q-1} je podgrafem stejné segmentace a obsahuje uzel v_j . Pokud $C_i^{q-1} \neq C_j^{q-1}$ a $w(o_q) \leq MInt(C_i^{q-1}, C_j^{q-1})$, pak segmentace S^q vznikne sloučením podgrafů C_i^{q-1} a C_j^{q-1} . Pokud tomu tak není, platí $S^q = S^{q-1}$. Na závěr vrať nejhodnější segmentaci, $S = S^*$.



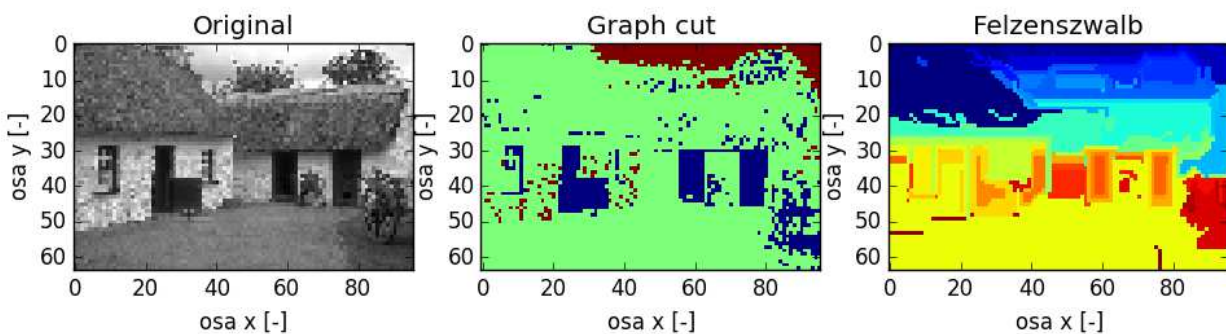
Obr. č. 4.14 – Ukázka segmentace Felzenszwalbovým algoritmem (vpravo) vstupního obrazu Man with hat ze sady Berkeley Segmentation Data Set[9] (vlevo), parametry byly nastaveny na následující hodnoty: měřítko ovlivňující velikost shluků bylo zvoleno 1, sigma pro vyhlazení Gaussovským jádrem byla nastavena na hodnotu 5 a minimální velikost jedné složky byla nastavena na hodnotu 100

4.10.2 Porovnání výsledků algoritmu s metodou Graph cut

Felzenszwalbův algoritmus jsem použil v programovacím jazyce Python za pomoci balíčku scikit-image. V balíčku scikit-image se nachází funkce felzenszwalb, která je implementací Felzenszwalbova algoritmu. Funkce vyžaduje čtyři vstupní parametry. Prvním parametrem je obrázek k segmentaci, další parametry představují měřítko, velikost Gaussovského jádra při operaci předzpracování a minimální velikost podgrafu. Velikost Gaussovského jádra jsem nastavil 5, to znamená, že jsem nastavil stejnou hodnotu jako při vyhlazování histogramu. Další parametry jsem nastavil empiricky. Velikost měřítka jsem nastavil na hodnotu 20 a minimální velikost složky jsem nastavil na hodnotu 100. Čím vyšší

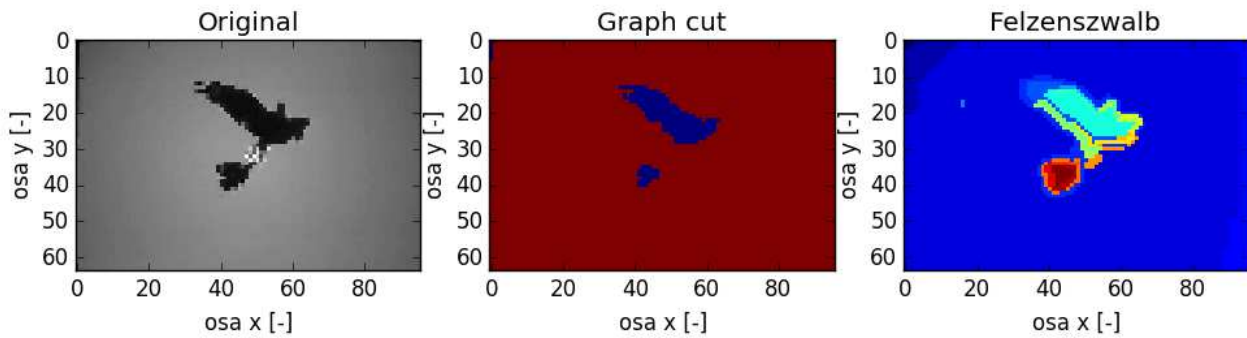
hodnoty těchto parametrů nastavíme, tím větší a kompaktnější získáme výsledné shluky. Ukázka segmentace felzenszwalbovým algoritmem je na obrázku číslo 4.14.

Hodnoty PR indexu metody Graph cut byly ve většině případů nižší než hodnoty PR indexu Felzenszwalbova algoritmu, viz obr. č. 4.22. Hodnoty PR indexu metody Graph cut byly často o 15% až 20% horší než hodnoty PR indexu výsledných segmentací Felzenszwalbova algoritmu, např. obr. č. 4.15. Horší výsledky segmentací metodou Graph cut z pohledu PR indexu byly získány díky tomu, že Graph cut segmentoval do značně menšího počtu složek než Felzenszwalbov algoritmus. Na druhou stranu jsou výsledné segmentace metodou Graph cut o něco srozumitelnější. Výsledné segmentované oblasti jsou kompaktnější, lze z nich vyčíst, co je objekt a co pozadí. Výsledné segmentace Felzenszwalbovým algoritmem jsou přesegmentované. Ve výsledné segmentaci je příliš mnoho složek, složky významově neodpovídají reálným předmětům ve vstupním obrazu. Tato skutečnost poukazuje, že hodnocení segmentací obrázků na základě PR indexu nemusí odpovídat skutečnému významu a srozumitelnosti segmentace.



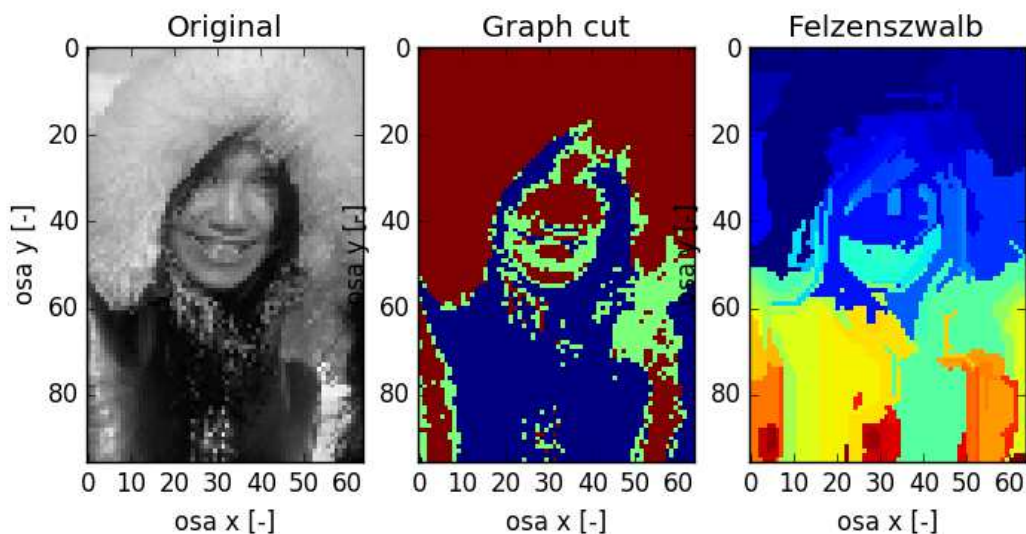
Obr. č. 4.15 – Porovnání segmentací vstupního obrázku Farm ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a Felzenszwalbovým algoritmem (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.450, hodnota PR indexu segmentace Felzenszwalbovým algoritmem byla 0.894

V menším počtu případů byla segmentace metodou Graph cut ohodnocena vyšší hodnotou PR indexu než výsledná segmentace Felzenszwalbovým algoritmem, např. 4.16. Dělo se tak v případech, kdy ve vstupním obraze byl pouze objekt a pozadí. Lepší hodnotou PR indexu byla ohodnocena ta segmentace, ve které bylo méně segmentovaných složek.



Obr. č. 4.16 - Porovnání segmentací vstupního obrázku Birds ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a Felzenszwalbovým algoritmem (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.966, hodnota PR indexu segmentace Felzenszwalbovým algoritmem byla 0.802

V několika případech nastala situace, kdy výsledné segmentace byly ohodnoceny skoro stejnou hodnotou PR indexu, např. obr. č. 4.17. Tato situace nastala v případech, kdy ve vstupním obraze bylo více objektů a pozadí.



Obr. č. 4.17 - Porovnání segmentací vstupního obrázku Eskymo ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a Felzenszwalbovým algoritmem (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.668, hodnota PR indexu segmentace Felzenszwalbovým algoritmem byla 0.666

4.11 Cut-normalized

4.11.1 Krátký popis algoritmu

Algoritmus N-Cut[13] přistupuje k segmentaci obrazu na základě konstrukce grafu a následného rozdělení grafu do oblastí. Graf $G = (V, E)$ je neorientovaný, V je množina uzlů představující pixely a E je množina hran. Každá e hrana množiny E má váhu w na určenou na základě podobnosti mezi uzly v_i a v_j . Segmentace je proces, během kterého hledáme neoptimálnější rozdělení množiny V do podmnožin V_1, V_2, \dots, V_m . Podobnost uzlů různých podmnožin je malá, každý uzel je obsažen pouze v jedné podmnožině.

Graf $G = (V, E)$ může být jednoduše rozdělen do podmnožin A, B ; $A \cup B = V$; $A \cap B = \emptyset$ pouhým odebráním jednotlivých hran. Hodnota nestejnosti, neboli řez, může být spočítána jako suma vah odebraných hran.

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v). \quad (4.6)$$

Vzhledem k obrovskému počtu možností rozdělení grafu je nalezení neoptimálnějšího řezu velmi výpočetně náročné. Algoritmus N-Cut používá jinou míru pro rozdělení grafu na podmnožiny. Tato míra zohledňuje náklady řezu jako části všech hran spojujících uzly grafu. Tato míra je nazvána normalized cut a je vypočítána:

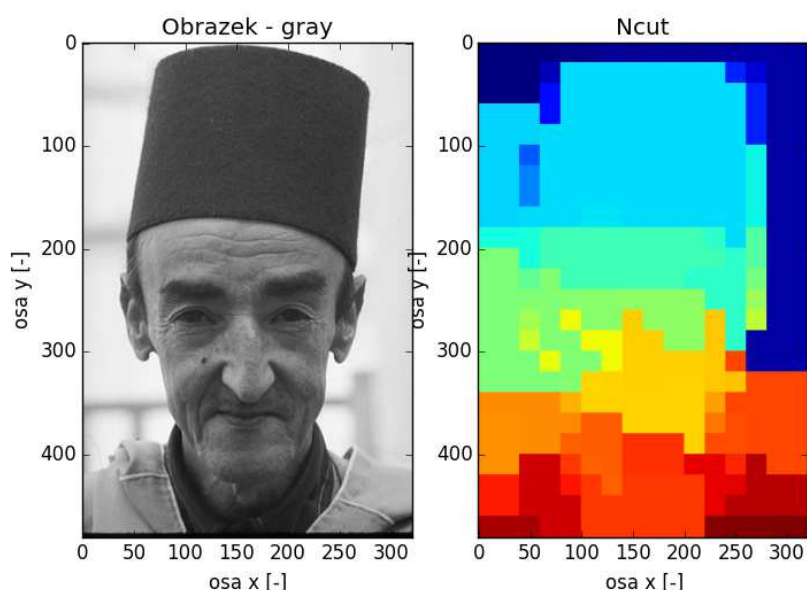
$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \quad (4.7)$$

$$\begin{aligned} assoc(A, V) &= \sum_{u \in A, t \in V} w(u, t), \\ assoc(B, V) &= \sum_{v \in B, t \in V} w(v, t). \end{aligned} \quad (4.8)$$

Hodnota $assoc(A, V)$ je vypočítána jako suma vah všech hran vedoucích z A ke všem uzlům grafu, podobně je spočtena i hodnota $assoc(B, V)$.

Algoritmus se skládá z několika kroků. Nejdříve načteme vstupní obraz a vytvoříme neorientovaný graf $G = (V, E)$. Ohodnotíme váhou w každou hranu z množiny E spojující dva uzly množiny V tak, aby váha hrany představovala míru podobnosti mezi dvěma uzly grafu. V dalším kroku řešíme soustavu rovnic $(D - W)x = \lambda Dx$. Hledáme vlastní vektor

s nejmenším vlastním číslem. D je diagonální matice o rozměrech $N \times N$ s $d(i) = \sum_j w(i, j)$ na diagonále. Výraz $d(i) = \sum_j w(i, j)$ představuje sumu vah hran spojující uzel se všemi ostatními uzly grafu. Matice W má také rozměry $N \times N$ a je symetrická, obsahuje prvky $W(i, j) = w_{ij}$. V dalším kroku algoritmu vybereme vlastní vektor s druhým nejmenším vlastním číslem a rozdělíme graf na dvě části. Nakonec algoritmus rozhoduje, zda by stávající dělení mohlo být dále děleno. Na základě kritéria je rozhodnuto, zda dojde k rozdělení již rozdělených částí, či nikoliv.



Obr. č. 4.18 - Ukázka segmentace algoritmem Normalized cut (vpravo) vstupního obrazu Man with hat ze sady Berkeley Segmentation Data Set[9] (vlevo), kromě parametru prahu, který byl nastaven na hodnotu 0.05, byly hodnoty ostatních parametru ponechány na výchozích hodnotách

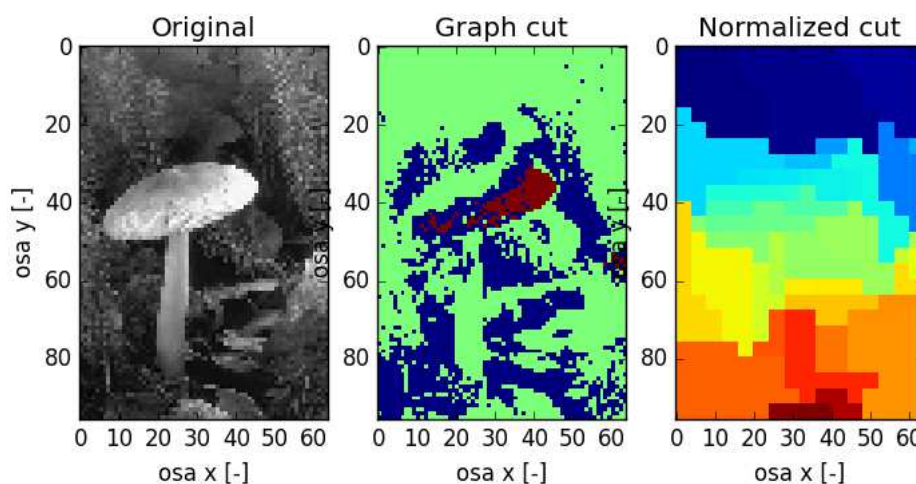
4.11.2 Porovnání výsledků algoritmu s metodou Graph cut

Algoritmus normalized cut jsem použil v programovacím jazyce Python za pomoci funkce `cut_normalized` z balíčku `scikit-image`. Funkce vyžaduje dva parametry – množinu labelů a graf. Ve funkci je možné ještě upravit volitelné parametry. Mezi volitelné parametry patří práh, počet provedených řezů před nalezením optimálního řezu i maximální velikost

hrany mezi dvěma rozdílnými oblastmi. Parametry jsem nastavil empiricky. Jediný parametr, který měl výrazný vliv na výslednou segmentaci, byla hodnota prahu. Po mnohých pokusech jsem nastavil hodnotu z výchozí hodnoty 0.001 na hodnotu 0.5. Při výchozí hodnotě nebyl schopen algoritmus segmentovat základní předměty, při větší hodnotě parametru, než je nastavená hodnota, docházelo k segmentování obrazu na příliš mnoho nevýznamných oblastí. Ukázka segmentace algoritmem N-Cut je na obrázku č. 4.18.

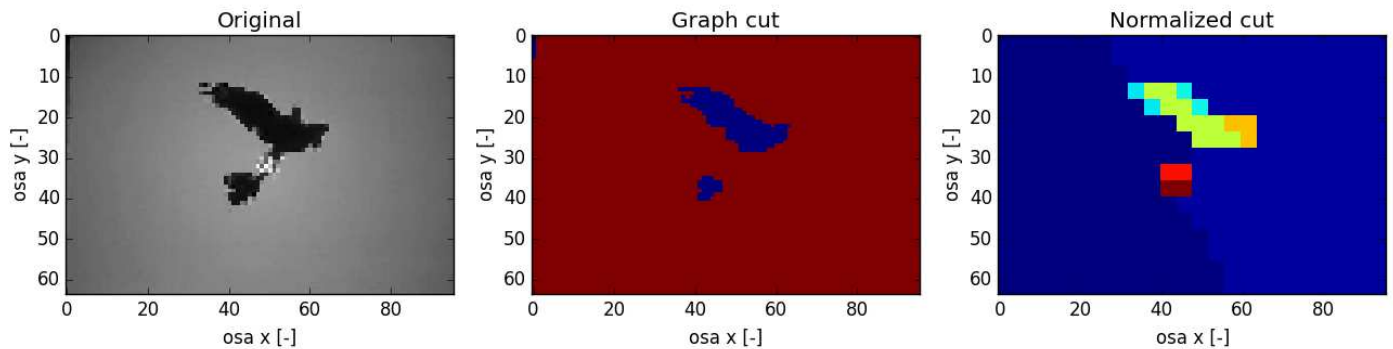
Porovnání výsledných segmentací metod Graph cut a Normalized cut za pomoci PR indexu dopadlo hůře pro metodu Graph cut, viz obr. č. 4.22. Výsledné segmentace metody Graph cut obsahují malý počet segmentovaných oblastí, počet oblastí závisí na počtu maxim v histogramu vstupního obrazu. Metoda Normalized cut segmentuje do mnoha oblastí. Zatímco ze segmentací metodou Graph cut lze rychle vyčíst co je objekt a co pozadí, v segmentacích metodou Normalized cut toto možné není. Výsledné segmentace obsahují velký počet oblastí, které neodpovídají reálným objektům v obraze. Segmentované oblasti metodou Normalized cut jsou často malé a hranaté. Segmentované oblasti metodou Graph cut jsou oproti tomu větší, spojitější, nejsou tolik hranaté a více odpovídají skutečným předmětům v obraze.

V největším počtu případů porovnání byly hodnoty PR indexu segmentací metodou Graph cut cca. o 20% až 25% horší než segmentace stejných obrazů metodou Normalized cut (viz obr. 4.19). Bylo to dáno především tím, že obrazy obsahovaly větší počet různých objektů.



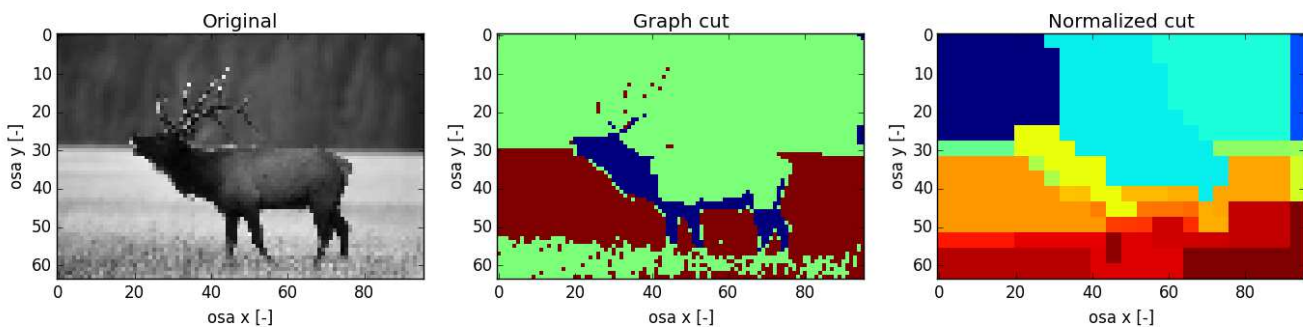
Obr. č. 4.19 - Porovnání segmentací vstupního obrázku Mushroom ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a metodou Normalized cut (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.507, hodnota PR indexu segmentace metody Normalized cut byla 0.898

V některých případech byly segmentace metodou Graph cut ohodnoceny vyšší hodnotou PR indexu než segmentace metodou Normalized cut, např. obr. 4.20. Stalo se tak v případech, kdy ve vstupních obrazech bylo malé množství objektů, nebo jen objekt a pozadí. Metoda Normalized cut obraz rozsegmentovala do více oblastí, což snížilo hodnotu PR indexu.

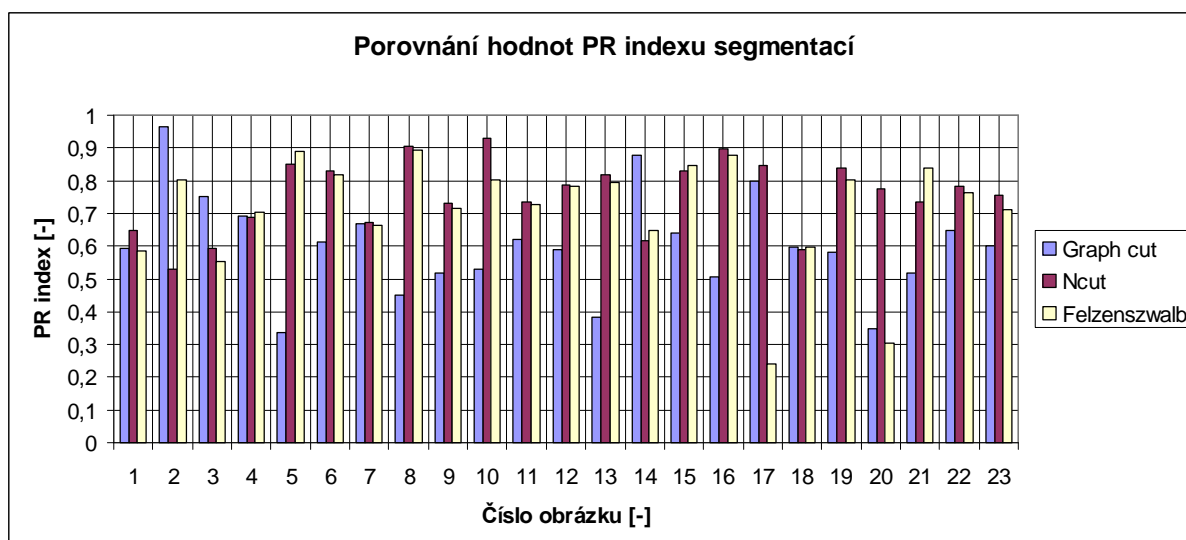


Obr. č. 4.20 - Porovnání segmentací vstupního obrázku Birds ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a metodou Normalized cut (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.966, hodnota PR indexu segmentace metody Normalized cut byla 0.529

V několika málo případech se stalo, že obě segmentace byly ohodnoceny podobnou hodnotou PR indexu, např. obr. č. 4.21. V těchto případech byl na obrázku opět menší počet objektů a pozadí.



Obr. č. 4.21 - Porovnání segmentací vstupního obrázku Deer ze sady Berkeley Segmentation Data Set[9] metodou Graph cut (uprostřed) a metodou Normalized cut (vpravo), hodnota PR indexu u metody Graph cut byla v tomto případě 0.690, hodnota PR indexu segmentace metody Normalized cut byla 0.687



Obr. č. 4.22 – Názorné porovnání hodnoty PR indexu segmentací Graph cutem, N-Cutem a Felzenszwalbovým algoritmem na některých obrázcích sady Berkeley Segmentation Data Set[9], pod číslem 23 se skrývá střední hodnota PR indexu metody

4.12 Možná vylepšení

Navržený postup segmentace obrazů je možné dále vylepšit. Možná vylepšení patří v několika směrech.

Jisté vylepšení navrhovaného přístupu by jistě nastalo vyladěním parametrů metody Graph cut. Ladění parametrů metody Graph cut nebylo cílem této práce, nicméně si myslím, že lze tímto způsobem dosáhnout lepších výsledků. Nastavení takto získaných parametrů by nebylo totožné pro všechny obrázky. Hodnoty parametrů by závisely na vlastnostech jednotlivých vstupních obrazů.

Další vylepšení se nabízí v detekování většího množství extrémů v histogramu, bylo by možné vytvořit větší množství jasových modelů. Výsledný obraz by byl segmentován do většího množství složek. Bylo by nutné najít takové optimální množství složek, aby nedošlo k přesegmentování obrazu – aby výsledná segmentace neobsahovala příliš velké množství složek.

V úvahu přichází i jiný přístup než detekce extrémů v histogramu. Kromě jasu by bylo vhodné segmentovat objekty na základě jejich souřadnic v obraze. Objekty se stejným nebo podobným jasem ale s odlišnou polohou by byly segmentovány do jiných tříd.

4.13 Vhodnost reálného nasazení metody Graph cut a teorie Markovských náhodných polí v praxi

Použití metody Graph cut a teorie Markovských náhodných polí[5],[7],[8],[11] v reálných úlohách počítačového zpracování obrazu považuji za vhodné. Na základě teorie Markovských náhodných polí segmentuje metoda Graph cut vstupní obrazy do menšího počtu složek, nedochází k přesegmentování obrazu na mnoho malých, až nevýznamných, složek. Ze segmentovaného obrazu je možné jasně rozeznat původní obrázek, segmentací získané objekty jsou spojitě. Segmentované složky odpovídají objektům skutečného světa.

Nejlepších výsledků bylo dosaženo na vstupních obrázcích s jedním velmi výrazným maximem v histogramu, dále v případech se dvěma výraznými avšak velmi rozdílnými maximy v histogramech vstupních obrázků.

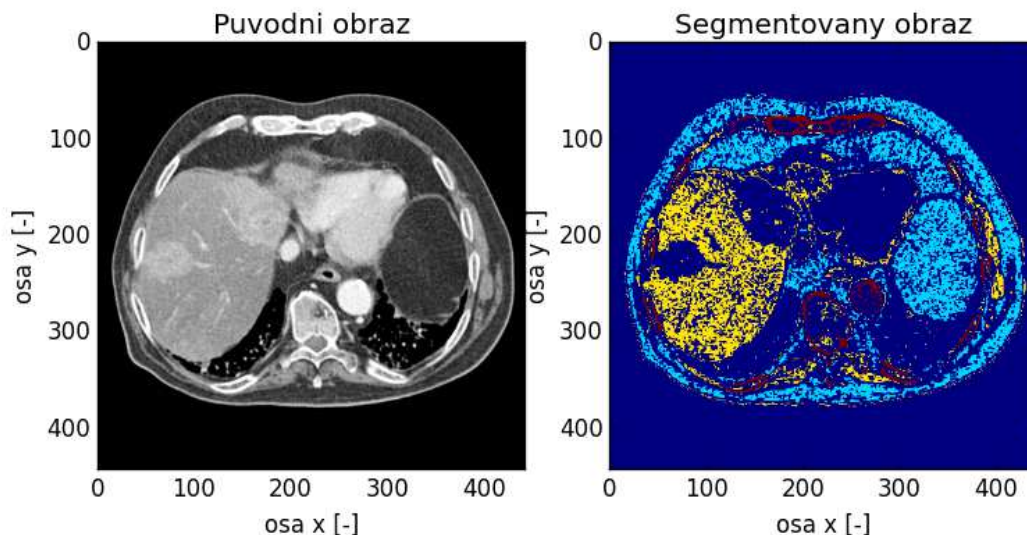
Za nevýhodu výsledných segmentací získaných na základě teorie Markovských náhodných polí metodou Graph cut je možné považovat malý počet segmentovaných složek. Segmentované složky jsou spojitě a při malých změnách skoro nedošlo ke změnám ve výsledných segmentacích. Tento případ by mohl záporně ovlivnit výsledky segmentace nad vstupní množinou obrázků s malým kontrastem, či v případech hledání malého nevýrazného objektu uvnitř většího objektu.

4.14 Medicínské snímky

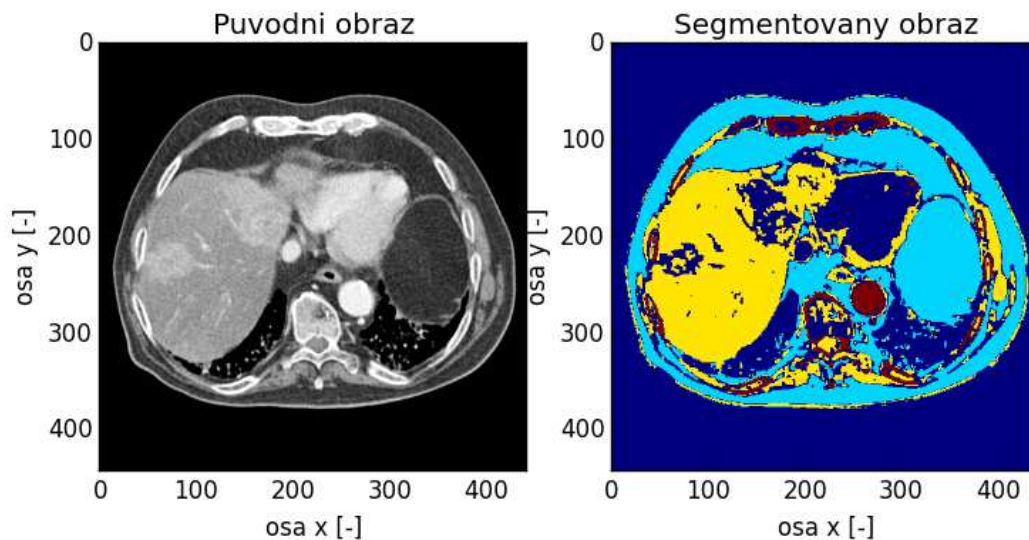
Výše uvedené postupy byly odzkoušeny také na medicínských snímcích. Postupy jsem vyzkoušel na CT snímcích. Cílem pokusů bylo v CT snímku detekovat nádor v játrech. V játrech existuje více typů nádorů, které je možné na snímku odlišit jasně. V případě hyperdenzních typů nádorů je nádor světlejší než okolí, v případě hypodenzních nádorů je nádor na CT snímku tmavší než okolí. Detekci nádoru znesnadňují určité okolnosti. Jedním z problémů detekce je, že nádor detekujeme v celém CT snímku, nemáme k dispozici jen snímek s výřezem samotných jater. Za další problém lze považovat hodnoty jasu nádorů, které se příliš neliší od okolí, snímek je v těchto místech málo kontrastní. Při detekci hyperdenzního nádoru je nutné počítat s tím, že podobný jas jako nádor mají i cévy v játrech.

Během pokusů o detekci nádoru způsobem, během kterého byly rozptýly jasových modelů získány na základě četnosti jasů v okolí jsem dospěl k ne zcela optimálním výsledkům. Bez problémů proběhla detekce maxim v histogramu, rozptýly byly vypočteny na základě velikosti okolí 30 úrovní. Ve výsledné segmentaci došlo k detekování nádorů, avšak nádory byly přiřazeny do třídy odpovídající prvnímu jasovému modelu vypočteného na základě maxima a jasů na nejnižších jasových úrovních (viz obr. 4.23 a 4.25). Stalo se tak díky hodnotě rozptýlů, první jasový model má oproti ostatním jasovým modelům velký rozptýl, křivka normálního rozdělení získaná na základě vypočteného rozptýlu je široká a pohlíjí jasové úrovně mezi normálami ostatních jasových modelů. Děje se tak z důvodu, že pravděpodobnost prvního jasového modelu je na těchto jasových úrovních větší než pravděpodobnost ostatních jasových modelů. Z našeho pohledu jde o chybné zařazení, nádor by měl být zařazen do třídy odpovídající jasově nejbližšímu jasovému modelu. Ve výsledné segmentaci bylo navíc přítomno velké množství šumu.

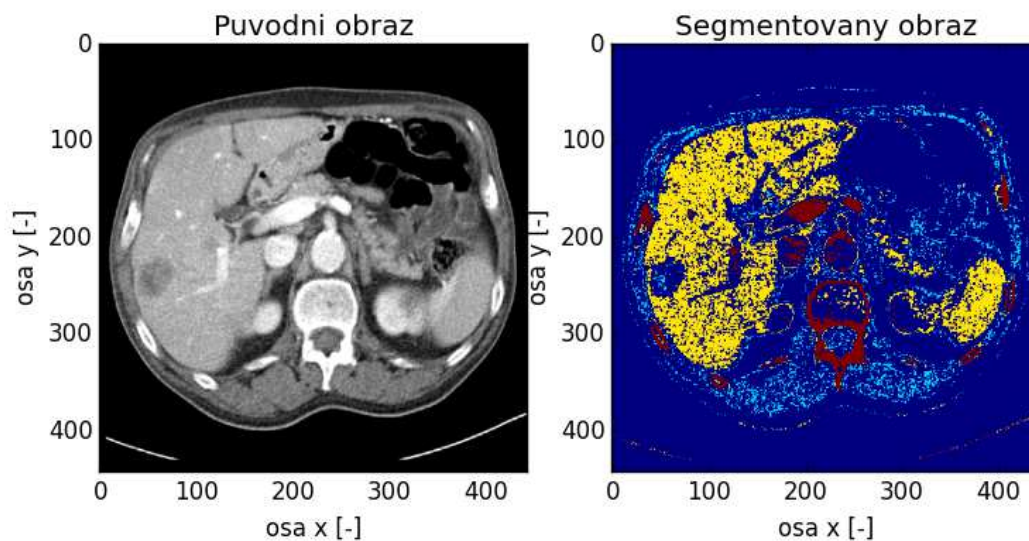
Určitého potlačení šumu jsem dosáhl přenásobením výsledných hodnot rozptýlů korekčním koeficientem (viz obr. 4.24 a 4.26). Přenásobením došlo ke zvětšení rozptýlů a potlačení šumu. Špatné zařazení nádorů do tříd však tímto způsobem zcela vyřešeno nebylo.



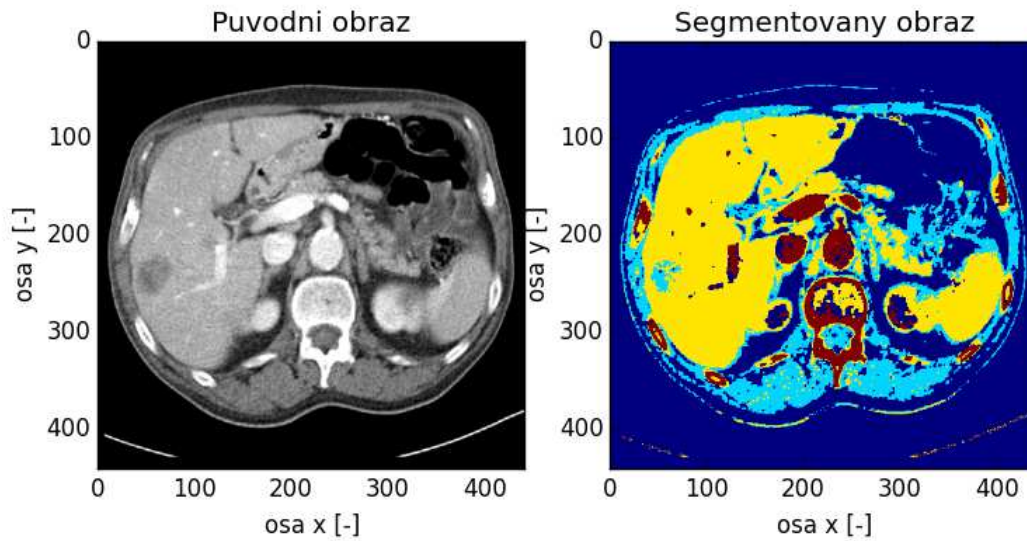
Obr. 4.23 – Snímek z CT, v játrech (segmentována žlutě) je přítomen hyperdenzní nádor, ten je přiřazen do špatné třídy, segmentace získána výše navrženým způsobem, rozptýly jsou počítány na základě okolí velikosti 30 jasových úrovní, přítomno velké množství šumu



Obr. 4.24 – Stejný snímek jako na obrázku 4.23, segmentace získána výše navrženým způsobem s přenásobením korekčního koeficientu o velikosti 20, rozptyly jsou počítány na základě okolí velikosti 30 jasových úrovní, přenásobením korekčním koeficientem došlo k potlačení šumu

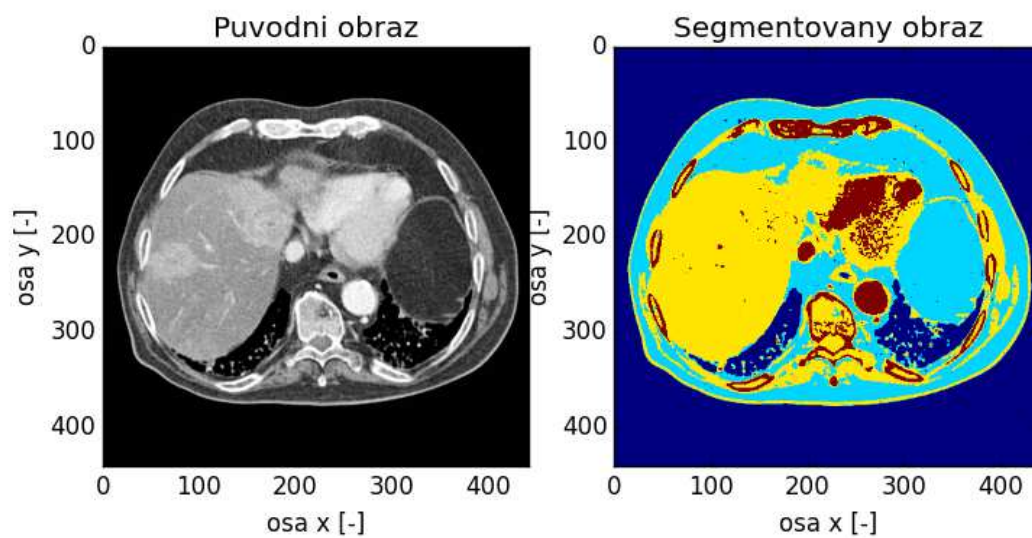


Obr. 4.25 – Snímek z CT, v játrech (segmentována žlutě) je přítomen hypodenzní nádor, ten je přiřazen do špatné třídy, segmentace získána výše navrženým způsobem, rozptyly jsou počítány na základě okolí velikosti 30 jasových úrovní, přítomno velké množství šumu

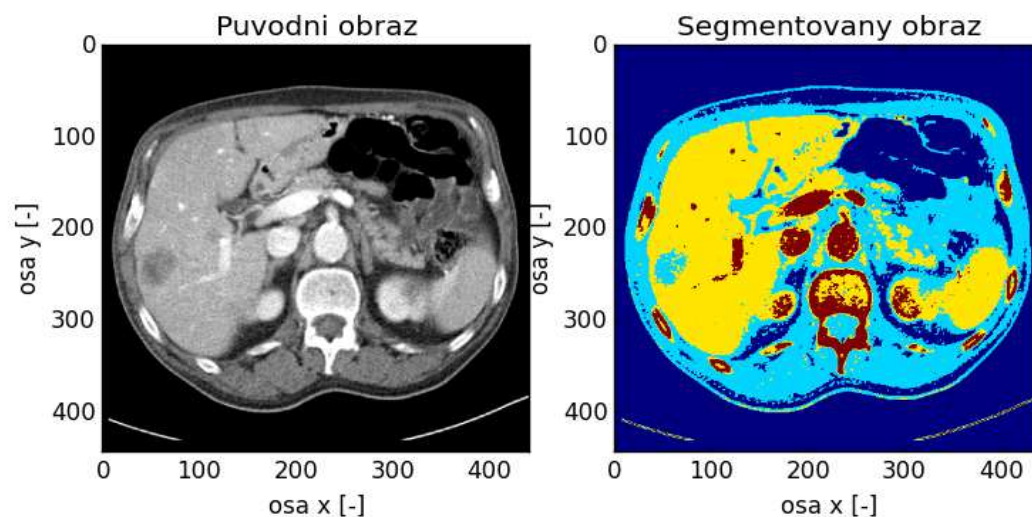


Obr. 4.26 - Stejný snímek jako na obrázku 4.25, segmentace získána výše navrženým způsobem s přenásobením korekčního koeficientu o velikosti 20, rozptyly jsou počítány na základě okolí velikosti 30 jasových úrovní, přenásobením korekčním koeficientem došlo k potlačení šumu, v tomto případě také ke správnému zařazení nádoru do třídy odpovídající nejbližšímu jasovému modelu

Určitého zlepšení situace jsem dosáhl částečným opuštěním způsobu výpočtu rozptylu na základě okolí střední hodnoty v histogramu. Pro získání jasových modelů byla opět detekována maxima v histogramu, rozptyly však byly zadány fixně, nebyl uvažován výpočet rozptylů na základě okolí maxim v histogramu. Při tomto postupu došlo v některých případech ke správnému zařazení nádorů do tříd nejbližších jasových modelů (viz obr. 4.27 a 4.28). Avšak ani v tomto případě nebyly výsledky zcela optimální. V případě snímků s hyperdenzními nádory nedošlo ve výsledné segmentaci k detekci nádorů v játrech. Výhodu tohoto postupu lze spatřit v nepřítomnosti šumu ve výsledných segmentacích.



Obr. 4.27 – Snímek z CT, v játrech (segmentována žlutě) je přítomen hyperdenzní nádor, který nebyl ve výsledné segmentaci detekován



Obr. 4.28 - Snímek z CT, v játrech (segmentována žlutě) je přítomen hypodenzní nádor, v tomto případě došlo ve výsledné segmentaci k detekci nádoru, nádor by přiřazen správně do třídy odpovídající nejbližšímu jasovému modelu.

5 Závěr

Hlavním cílem práce byla segmentace obrazu pomocí Markovských náhodných polí a automatické vytvoření jasových modelů, na základě kterých proběhne segmentace vstupního obrazu.

Teoretická část seznamuje čtenáře s pojmy, bez kterých se nebylo možné při vypracování práce obejít. V práci byly použity pojmy ze dvou oblastí. První oblastí jsou základní pojmy týkající se počítačového zpracování obrazu[1],[2] a statistiky[4], do druhé oblasti patří seznámení se s teorií Markovských náhodných polí[3],[5],[7],[8],[11].

Mezi základní pojmy patří například histogram a jeho vyhlazení, konvoluce, střední hodnota, rozptyl, normální rozdělení, segmentace obrazu, či ohodnocení segmentovaných obrazů pomocí Rand indexu a Probabilistic Rand indexu. Aplikací těchto pojmů byl předzpracován histogram a dále byly automaticky vytvořeny jasové modely.

Do druhé oblasti teoretické části práce patří teorie Markovských náhodných polí. Teorie Markovských náhodných polí je velmi obsáhlá a složitá. Zjednodušeně se dá napsat, že v případě použití teorie k segmentaci obrazů pojednává o co nejoptimálnějším rozdělení objektů na podobjekty na základě grafu. Na teorii Markovských náhodných polí stojí segmentační metoda Graph cut, další metody jako simulované žíhání, genetické algoritmy, či gradientní algoritmy mohou teorie Markovských náhodných polí využít. Cílem práce bylo pouze použití metody Graph cut.

Praktická část pojednává o přístupu automatického vytváření jasových modelů pro účely segmentace obrazů metodou Graph cut. Jasové modely byly vytvářeny automaticky na základě histogramu. Histogram bylo nutné nejdříve vyhladit, poté byla v histogramu detekována informačně nejvýznamnější maxima. Automatické jasové modely byly vytvořeny za pomoci Gaussovy křivky, jejíž parametry se odvíjely od charakteristik histogramu daného obrazu. Uvažované okolí střední hodnoty jasového modelu velmi ovlivňuje výslednou kvalitu jasového modelu. Pro ohodnocení kvality jasových modelů s různou velikostí uvažovaného okolí bylo využito porovnání PR indexem[6]. Předpokládaný vývoj PR indexu se zvětšujícím se okolím nebyl porovnáními PR indexem[6] potvrzen.

Výsledné segmentace metodou Graph cut byly porovnány s podobnými přístupy k segmentaci. Konkrétně byly výsledné segmentace porovnány se segmentacemi vytvořenými Felzenszwalbovým algoritmem a metodou Normalized cut. Tyto metody přistupují

k segmentaci také za pomoci dělení grafu. Segmentace byly porovnány opět za pomoci PR indexu[6].

Na práci je možné dále navázat. V histogramu je možné detekovat více extrémů, díky čemuž dojde k automatickému vytvoření většího množství jasových modelů a lepší segmentaci. Bude však nutné najít optimální počet složek v segmentaci tak, aby nedošlo k přesegmentování obrazu. Jistá vylepšení je možné nalézt i při vyladění parametrů metody Graph cut. Ladění parametrů metody Graph cut však nebylo cílem této práce. Výrazného zlepšení by jistě šlo dosáhnout, pokud by při segmentaci byl uvažován nejen jas objektu v obraze, ale i jeho poloha v obraze.

6 Použité zdroje

- [1] Šonka, Miloš, Hlaváč, Václav, Boyle, Roger. Image Processing Analysis and Machine Vision. Thomson Learning 2008. ISBN 978-0-495-24428-7.
- [2] Železný Miloš. Přednášky k předmětu ZDO.
- [3] Hružík Marek, Krňoul Zdeněk. Přednášky k předmětu MPV.
- [4] Friesl, Michal. Pravděpodobnost a statistika HYPERTEXTOVĚ [online]. 2014 [cit. 2015-03-17]. Dostupné z: <http://home.zcu.cz/~friesl/hpsb/tit.html>
- [5] Li, S. Markov random field modeling in image analysis. 3rd ed. London: Springer, 2009, xxiii, 357 p. ISBN 1848002793.
- [6] Ranjith, Unnikrishnan a Herbert Martial. Measures of Similarity [online]. Pittsburgh, 2005 [cit. 2015-03-17]. Dostupné z: https://www.ri.cmu.edu/pub_files/pub4/unnikrishnan_ranjith_2005_1/unnikrishnan_ranjith_2005_1.pdf. Článek. Carnegie Mellon University.
- [7] Gould, Stephen. Markov Random Fields for Computer Vision (Part 1) [online]. 2011 [cit. 2015-03-17]. Dostupné z: <http://users.cecs.anu.edu.au/~sgould/papers/part1-MLSS-2011.pdf>
- [8] Kato, Zoltan. Markov Random Fields in Image Segmentation [online]. 2008 [cit. 2015-03-17]. Dostupné z: https://inf.u-szeged.hu/~SSIP/2008/presentations2/Kato_Ssip2008.pdf
- [9] Martin, David, Fowlkes, Charless, Tal Doron, Malik, Jitendra. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. ICCV. 2001.
- [10] Arbelaez, Pablo, Maire, Michael, Fowlkes, Charless, Malik, Jitedendra. Contour Detection and Hierarchical Image Segmentation. IEEE TPAMI. 2011, Vol. 33, No. 5, pp. 898-916.

- [11] Boykov, Juri, Veksler, Olga. Graph Cut in Vision and Graphics: Theoris and Applications. Handbook of Mathematical Models in Computer Vision, Springer, 2006, pp. 79-86.
- [12] Felzenszwalb, Pedro F., and Daniel P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision. 2004, Vol. 59, No. 2, pp. 167-181.
- [13] Shi, Jianbo, Malik, Jitendra. Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence. IEEE Transactions on. 2000, Vol. 22, No. 8, pp. 888-905.