

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

# Analýza obrazu pomocí metody Random Walker

Diplomová práce

Plzeň, 2015

Tomáš Filandr

# Prohlášení

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

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

V Plzni dne: .....

podpis: .....

# Poděkování

Rád bych poděkoval Ing. Tomáši Rybovi za odborné vedení diplomové práce a spoustu dobrých nápadů, rad a připomínek při její tvorbě. Dále bych rád poděkoval své rodině, která mne po celou dobu studia podporovala a udržovala na správné cestě.

# Anotace

Tématem práce je segmentace obrazu pomocí grafových metod. Zvolenými metodami jsou random walker, GrabCut, normalized cut a upravená metoda random walker pracující se superpixely, implementace této metody je součástí práce. Pro porovnání výsledků jednotlivých metod je použit Probabilistic Rand index. Problémem porovnávání segmentací je různý počet výsledných segmentů od různých metod. Probabilistic Rand index je v tomto případě výhodný tím, že porovnává příslušnosti k segmentům mezi dvojicemi bodů obrazu a tak nezáleží na počtu segmentů.

**Klíčová slova:** segmentace, grafové metody, random walk, superpixely, Probabilistic Rand index

# Abstract

This work focuses on image segmentation using graph methods. Methods chosen are random walker, GrabCut, normalized cut and a modified method of random walker working with superpixels, this method is implemented in the work. To compare the results of different methods Probabilistic Rand Index is used. Problem with comparing segmentations is a different number of segments resulting from the different methods. Advantage of Probabilistic Rand index is that it compares label relationship of a pixel pair and thus does not depend on the number of segments.

**Key words:** segmentation, graph algorithms, random walk, superpixels, Probabilistic Rand index

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Teoretická část</b>	<b>6</b>
2.1	Segmentační metody . . . . .	6
2.1.1	Úvod . . . . .	6
2.1.2	Random walker . . . . .	7
2.1.3	GrabCut . . . . .	10
2.1.4	Normalized cut . . . . .	12
2.2	Superpixely . . . . .	17
2.2.1	Úvod . . . . .	17
2.2.2	SLIC . . . . .	18
2.3	Ohodnocení segmentace . . . . .	20
2.3.1	Úvod . . . . .	20
2.3.2	Probabilistic Rand index . . . . .	21
<b>3</b>	<b>Praktická část</b>	<b>23</b>
3.1	Dílčí úlohy . . . . .	23
3.1.1	Úvod . . . . .	23
3.1.2	Obraz . . . . .	23
3.1.3	Superpixely . . . . .	24

3.1.4	Random walker . . . . .	25
3.1.5	Probabilistic Rand index . . . . .	28
3.2	Srovnávací test . . . . .	29
3.2.1	Úvod . . . . .	29
3.2.2	Postup . . . . .	30
3.2.3	Vyhodnocení . . . . .	32
<b>4</b>	<b>Závěr</b>	<b>35</b>
<b>A</b>	<b>Tabulky srovnávací test</b>	<b>41</b>
<b>B</b>	<b>Obrázky srovnávací test</b>	<b>49</b>

# Kapitola 1

## Úvod

Cílem této práce je porovnání grafových metod pro segmentaci obrazu, což je činnost, při které jsou automaticky či za pomoci člověka rozlišovány objekty v obraze. Segmentace obrazu je podstatným krokem nižší úrovně zpracování a její výsledky jsou předávány vyšší úrovni, která se obvykle zabývá rozpoznáváním obrazu, neboli počítačovým viděním. Pro správnou funkci metod vyšší úrovně zpracování je nutná dostatečná kvalita segmentace.

Hlavní metodou práce měl být algoritmus *Random walker*, ale po zjištění neuspokojivého chování, co se týče respektování oblastí spojených při tvorbě superpixelů, byla implementována vlastní verze algoritmu *Random walker* pracující se superpixely (*SPRW*), která tímto nedostatkem netrpí (obrázek č. 1.1). Dalším důvodem pro implementaci *SPRW* byl předpoklad, že zmenšením grafu s nímž metoda musí pracovat, by mělo dojít ke zrychlení výpočtu oproti klasické metodě *Random walker*.

Dále byly v práci použity další grafové metody a to *GrabCut* a *Normalized cut*. Tyto metody byly použity pro srovnání s implementovanou metodou *SPRW*. Pro tvorbu superpixelů byla použita metoda *SLIC0*, která měla nejlepší výsledky z testovaných metod. Pro účely srovnání výsledků segmentací

je implementován *Probabilistic Rand* index (*PR* index), výhodou této metody srovnání je, že nezáleží na kolik segmentů byl obraz rozdělen, jelikož je porovnávána příslušnost k segmentu mezi dvojicí pixelů.

Pro implementaci práce byl zvolen jazyk Python, který je hojně používán ve vědecké oblasti a proto je v něm k dispozici mnoho knihoven pro práci s obrazem. Navíc díky knihovně *NumPy*<sup>1</sup> jsou v tomto jazyce k dispozici funkce srovnatelné s matematickým nástrojem *MATLAB*®<sup>2</sup>.

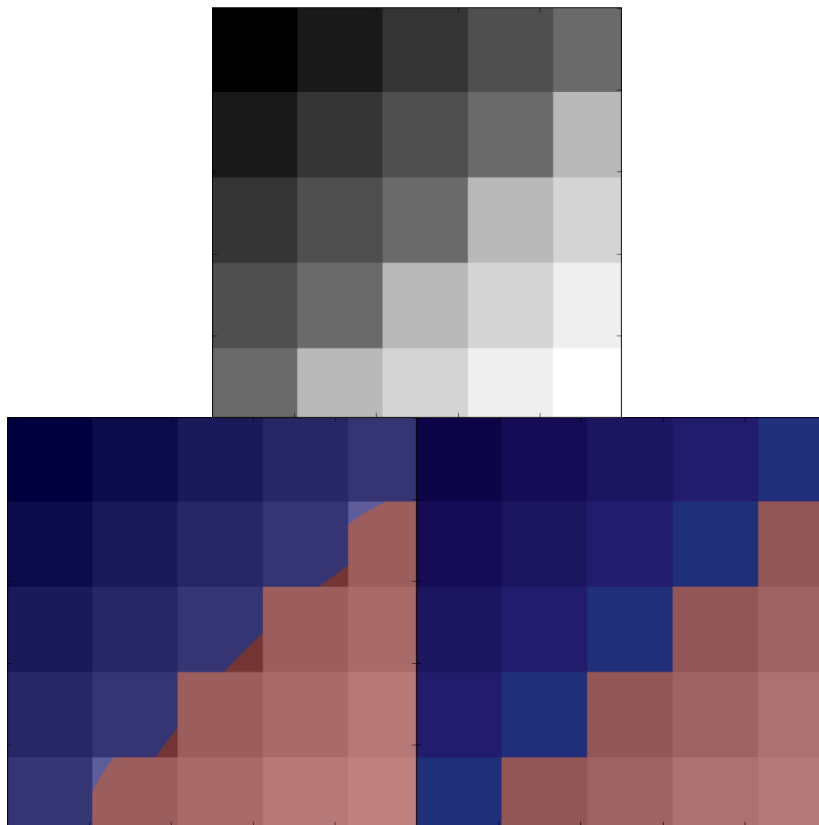
Práce se skládá z teoretické části, zde jsou popsány jednotlivé metody použité v práci. Praktická část se věnuje jednotlivým metodám tak, jak byly použity při implementaci *SPRW*. Následuje experiment, kde byly porovnávány segmentace jednotlivých metod na sadě obrazů, při použití různých vstupních dat od uživatele. V závěru práce jsou shrnuty získané poznatky.

---

<sup>1</sup><http://www.numpy.org/>

<sup>2</sup>*MATLAB*® je registrovaná obchodní značka The MathWorks.





Obrázek 1.1: Nahoře: originál, vlevo: segmentace random walker, vpravo: upravenou metodou.

# Kapitola 2

## Teoretická část

### 2.1 Segmentační metody

#### 2.1.1 Úvod

Segmentační metody jsou metody, jejichž cílem je rozdělení obrazu na logické části, neboli oddělení objektů v obraze se vyskytujícími od pozadí. Tyto logické celky vykazují jistou podobnost, ať už se jedná o podobnost v barvě, či intenzitě při šedotónovém obraze, nebo o podobnost vzoru. Segmentace je úkonem, prováděných v nižší úrovni zpracování obrazu. Před prováděním segmentace je obvykle nutné obraz předzpracovat, to zahrnuje digitalizaci obrazu, jeho vhodnou prezentaci v počítači, například matici, dále filtraci pokud se v obraze vyskytuje šum a další metody. Na segmentovaný obraz je poté použit jako vstup pro metody vyššího zpracování, jejich úkolem je obvykle rozpoznání jednotlivých objektů na obraze pomocí znalostních systémů. Aby tyto metody vyššího zpracování mohly být úspěšné, je nutné provést segmentaci dostatečně kvalitně.

Všechny segmentační metody, v této práci použité, patří do rodiny grafo-

vých metod. Tyto metody pohlížejí na segmentovaný obraz jako na graf, což je výhodné, ať už z pohledu přehlednosti takto prezentovaného obrazu, tak z pohledu metod jež nabízí teorie grafů. Obvyklým cílem grafových metod je rozdělení tohoto grafu na podgrafy tak, aby každý podgraf náležel jednomu objektu v obraze.

### 2.1.2 Random walker

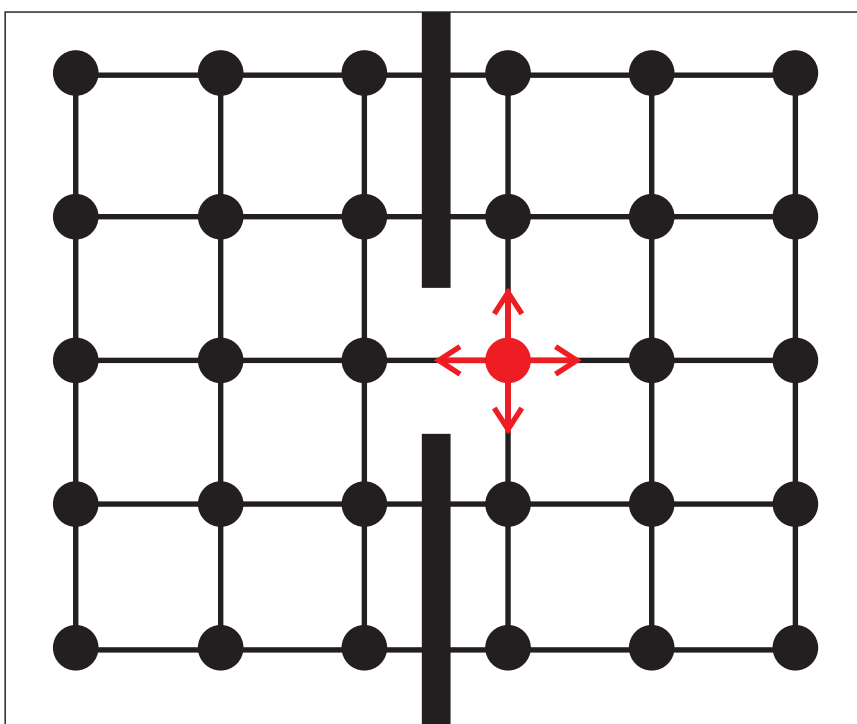
*Random walker* je grafová metoda používaná pro úplnou segmentaci obrazu. Algoritmus byl poprvé představen ve zkrácené formě ve sborníku [1]. Jeho úplné znění lze nalézt v článku [2]. Jedná se o interaktivní metodu, což znamená že při jejím používání jsou zapotřebí data zadaná uživatelem, konkrétně jde o zadání počátečních bodů (seedů). Pomocí těchto bodů se označí objekty v obraze, těch může být i více, dále se označí pozadí.

Algoritmus poté spočte pro každý neoznačený bod obrazu pravděpodobnost, že náhodná procházka vycházející z tohoto bodu dojde do jednoho z označených bodů (seedů). Tato pravděpodobnost je spočtena pro všechny dvojice neoznačený bod, seed. Výpočet je proveden vyřešením soustavy lineárních rovnic. Příslušnost každého neoznačeného bodu je určena podle maxima ze všech vypočtených pravděpodobností pro tento bod.

Hlavní přednost metody *Random walker* shledávám ve schopnosti správně segmentovat oblasti s neúplnými, či nezřetelnými hranami, jak je vidno na obrázku č. 2.1. Tento jev je způsoben náhlým pádem pravděpodobnosti, při průchodu oblastí neúplné hrany. Představme si, že náhodná procházka dojde na místo, kde je díra v hraně (obrázek č. 2.2), při pokračování z tohoto bodu je 75% šance, že zůstane na své straně hrany a pouze 25%, že překročí na druhou stranu. Stejně tak to platí pokud se bude blížit z druhé strany hrany. Tento jev nám zajistí kvalitní segmentaci, i když je hrana v obraze neúplná.



Obrázek 2.1: Segmentace metodou Random walker, respektování neúplných hran.



Obrázek 2.2: Metoda Random walker, respektování neúplných hran.

## Výpočet

Předpokládejme obraz reprezentovaný grafem, kde každý pixel obrazu tvoří uzel grafu  $v_i$ . Sousední pixely jsou spojeny hranou, kde hrana  $e_{i,j}$  označuje spojení uzlů  $v_i$  a  $v_j$ . Hrany jsou ohodnoceny vahou  $w_{i,j}$  určující podobnost propojených uzlů. Mějme intenzitu  $z_i$  uzlu  $v_i$  v šedotónové reprezentaci obrazu, váhu lze poté určit vztahem

$$w_{i,j} = e^{-\beta \cdot \text{abs}(z_i - z_j)} , \quad (2.1)$$

kde  $\beta$  je volný parametr nastavovaný při spuštění algoritmu.

Poté co uživatel označí některé uzly grafu jako náležící objektu či pozadí, lze celý soubor uzlů  $V$  rozdělit na označené uzly  $V_M$  a uzly neoznačené  $V_U$  a to takovým způsobem, že  $V_M \cup V_U = V$  a  $V_M \cap V_U = \emptyset$ . Cílem algoritmu je opatřit všechny uzly  $v_i \in V_U$  označením příslušnosti z množiny  $G = \{g^1, g^2, \dots, g^k\}$ . Uzly  $v_i \in V_M$  již toto označení mají zadáno uživatelem při startu algoritmu.

Pro určení označení je každému uzlu  $v_i \in V_U$  přidělena pravděpodobnost  $x_i^s$ , že náhodná procházka začínající v tomto uzlu, dojde jako první do označeného uzlu  $v_j \in V_M$ , jenž je označen příslušností  $g^s$ . U každého uzlu  $v_i \in V_U$  jsou spočteny pravděpodobnosti pro všechny cíle cesty, tedy všechny uzly  $v_j \in V_M$  a uzlu  $v_i \in V_U$  je poté přiděleno označení podle maximální z těchto pravděpodobností, tedy  $y_i = \max_s x_i^s$ . Na problém lze nahlížet jako na minimalizaci energie

$$E = x^{sT} L x^s , \quad (2.2)$$

kde  $x^s$  je vektor o rozměrech  $n \times 1$  kde  $n = |V|$ . Složky vektoru jsou pravděpodobnosti  $x_i^s$ , pokud náhodná procházka z uzlu  $v_i$  narazí na první označený uzel  $v_j \in V_M$  s označením  $g^s$  je  $x_j^s = 1$ . Naproti tomu dojde-li jako první do uzlu  $v_j \in V_M$  s označením  $g^{q \neq s}$  je  $x_j = 0$ .

$L$  je Laplaceova matice definovaná jako

$$L_{v_i v_j} = \begin{cases} d_{v_i} & \text{pro } i = j \\ -w_{ij} & \text{pokud mezi } v_i \text{ a } v_j \text{ je hrana} \\ 0 & \text{v ostatních případech ,} \end{cases} \quad (2.3)$$

kde  $d_{v_i}$  je definováno jako suma vah všech hran z uzlu  $v_i$  tedy  $d_{v_i} = \sum w(e_{ij})$ . Matici  $L$  lze rozdělit na část s označenými uzly  $L_M$  a část s neoznačenými uzly  $L_U$  takto

$$L = \begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix} . \quad (2.4)$$

Zavedením vektoru  $f^s$  o rozměrech  $|V_M| \times 1$  jako

$$f_j^s = \begin{cases} 1 & \text{pro } y_j = g^s \\ 0 & \text{pro } y_j \neq g^s , \end{cases} \quad (2.5)$$

lze minimalizaci rovnice 2.2 zajistit vyřešením soustavy rovnic

$$L_U x_U^s = -B f^s . \quad (2.6)$$

Jedná se o řídkou, symetrickou, pozitivně definitní soustavu lineárních rovnic což umožňuje rychlejší výpočet. Pro výpočet lze použít například metodu konjugovaných gradientů viz [3].

### 2.1.3 GrabCut

Metoda *GrabCut* byla představena ve článku [4]. Jedná se o vylepšenou verzi známého algoritmu *Graph cut* [5]. Segmentace metodou *Graph cut* patří mezi interaktivní metody. V případě této metody člověk určí popředí a pozadí obrazu. Algoritmus tyto body využije při konstrukci grafu. Z bodů popředí

udělá vrcholy zdroje a z bodů pozadí vrcholy spotřebiče nebo stoku. Takto vzniklý zdroj a spotřebič se napojí na všechny ostatní body v obraze. Cena vzniklých hran se určí podle rozdílu jasů a také je zde zohledněna vzdálenost mezi zdrojem popř. spotřebičem a vrcholem grafu. Úkolem algoritmu je poté nalézt řez grafem tak, že zdroj a spotřebič budou v rozdílných podgrafech a řez bude proveden hranami s nejmenším součtem vah.

## Výpočet

Algoritmus *Graph cut* v práci [5] pracuje s šedotónovým obrazem, kde obraz je převeden na vektor  $\mathbf{z} = \{z_1, z_2, \dots, z_N\}$ , segmentace je vyjádřena jako vektor  $\underline{\alpha} = \{\alpha_1, \dots, \alpha_N\}$ . Kde teoreticky  $0 \leq \alpha_n \leq 1$ , ale pro striktní segmentaci používáme pouze  $\alpha_n = \{0, 1\}$  kde 0 je pozadí a 1 je popředí. Dále je zaveden parametr  $\underline{\Theta}$  popisující šedotónovou distribuci v popředí a pozadí, skládá se z histogramů

$$\underline{\Theta} = \{h(z; \alpha), \alpha = 0, 1\} , \quad (2.7)$$

jednoho histogramu pro popředí a jednoho pro pozadí. Histogramy jsou sestaveny z dat zadaných uživatelem,  $T_B$  pro označení pozadí a  $T_F$  pro označení popředí. Histogramy jsou normovány. Úkolem segmentace je určit neznámé hodnoty proměnné  $\underline{\alpha}$  z dodaných dat  $\mathbf{z}$  a modelu  $\underline{\Theta}$ . Toho je dosaženo minimalizací energie

$$\mathbf{E}(\underline{\alpha}, \underline{\Theta}, \mathbf{z}) = U(\underline{\alpha}, \underline{\Theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}) . \quad (2.8)$$

Funkce  $U$  ohodnocuje vhodnost segmentace  $\underline{\alpha}$  k datům  $\mathbf{z}$  při známém modelu  $\underline{\Theta}$  a je definována jako

$$U(\underline{\alpha}, \underline{\Theta}, \mathbf{z}) = \sum_n -\log h(z_n; \alpha_n) . \quad (2.9)$$

Funkce  $V$  ohodnocuje hladkost segmentace

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in \mathbf{C}} dis(m, n)^{-1} [\alpha_n \neq \alpha_m] exp - \beta(z_m - z_n)^2, \quad (2.10)$$

kde  $[\alpha_n \neq \alpha_m]$  značí indikační funkci dosahující hodnot 0, nebo 1 v závislosti na pravdivosti výrazu uvnitř.  $\mathbf{C}$  je množina dvojic sousedících vrcholů a  $dis(.)$  je euklidovská vzdálenost. Při takto definovaném modelu energie lze segmentaci odhadnout jako globální minimum vztahu

$$\hat{\underline{\alpha}} = arg \min_{\underline{\alpha}} \mathbf{E}(\underline{\alpha}, \underline{\Theta}), \quad (2.11)$$

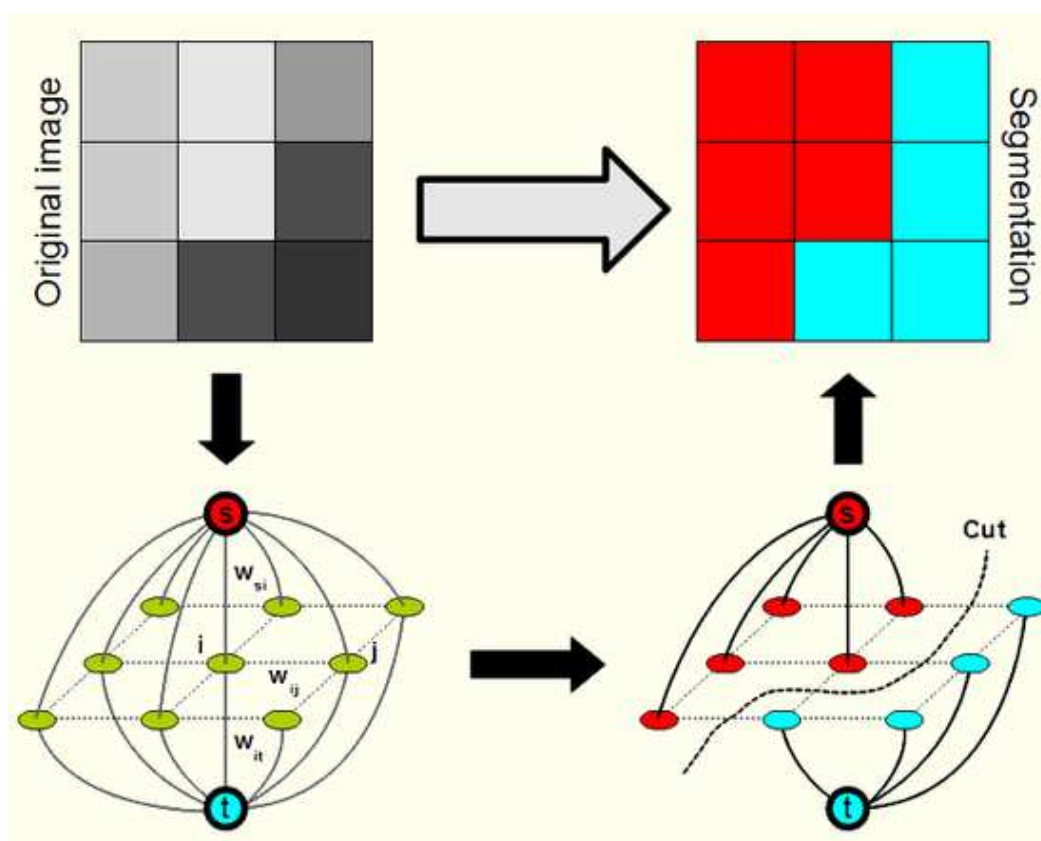
minimalizace je provedena použitím algoritmu minimálního řezu (obrázek č. 2.3) popsaného v [5].

Tento algoritmus se stal základem pro metodu *GrabCut*. Ta ho dále rozvíjí pro použití na barevných obrazech nahrazením histogramů modely gaussovských směsí. Jednoduchá minimalizace byla nahrazena iterativní verzí která se přepíná mezi hledáním minima a odhadem parametrů. Posledním zlepšením je zjednodušená interakce s člověkem kdy uživateli stačí označit obdélník kolem objektu na obraze a tím vlastně označí pouze pozadí. Já jsem bohužel tuto možnost nevyužil jelikož jsem z důvodů srovnání metod potřeboval shodné zásahy od uživatele pro všechny interaktivní metody které jsem používal.

#### 2.1.4 Normalized cut

Jedná se o plně automatickou segmentační metodu představenou v [6]. Metoda se soustředí na rozdělení grafu reprezentujícího obraz na několik podgrafů. Ty jsou tvořeny tak, aby podobnost mezi body v daném podgrafu byla co největší, zatímco podobnost mezi dvěma podgrafy byla co nejmenší.





Obrázek 2.3: Segmentace minimálním řezem grafu. Převzato z: <http://www.ondrej-danek.net/images/vyzkum/graph.png>

## Výpočet

Mějme graf  $G = (V, E)$ , kde  $V$  jsou vrcholy grafu a  $E$  jsou hrany. Tento graf chceme rozdělit na dva podgrafy  $A$  a  $B$  a to tak, aby  $A \cup B = V$  a  $A \cap B = \emptyset$ . Rozdělení provedeme odebráním hran spojující tyto podgrafy, toto se označuje jako řez grafem

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) . \quad (2.12)$$

Optimální řez je takový, který minimalizuje tento vztah. Avšak použití této míry vede k tvorbě osamocených vrcholů, kdy vrcholy které jsou vzdálenější od zbytku grafu jsou samostatně rozděleny do podgrafů. Toto chování není ideální a proto se v metodě *Normalized cut* tato míra upravuje tak aby vyjadřovala poměr mezi hranami které byly odstraněny a všemi hranami kterými je daný uzel napojen na zbytek grafu

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} , \quad (2.13)$$

kde  $assoc(A, V)$  je právě suma vah všech hran vedoucích z uzlů náležících oblasti  $A$  do ostatních uzlů grafu. Takto již nebudou oddělovány samostatné vrcholy, jelikož taková akce by vedla k odebrání 100% hran které tyto vrcholy spojují se zbytkem grafu.

Dále definujeme míru propojenosti mezi dvěma oblastmi grafu pomocí vztahu

$$Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)} , \quad (2.14)$$

kde  $assoc(A, A)$  a  $assoc(B, B)$  je suma vah všech hran uvnitř oblasti  $A$  resp.  $B$ . Tato míra ukazuje jak pevně jsou spojeny uzly uvnitř oblasti grafu.

Cílem metody *Normalized cut* je právě minimalizování vztahu  $Ncut$  pro prováděné řezy grafem a zároveň maximalizování míry  $Nassoc$ , protože platí

$$Ncut(A, B) = 2 - Nassoc(A, B) , \quad (2.15)$$

tak ideální rozdělení  $(A^*, B^*)$ , kterým minimalizujeme  $Ncut(A, B)$ , bude zároveň maximalizovat  $Nassoc(A, B)$ .

Pro nalezení tohoto rozdělení nejprve zavedme několik proměnných,

$$\mathbf{d}(i) = \sum_j w_{ij} , \quad (2.16)$$

jako sumu spojení uzlu  $i$  se všemi ostatními uzly grafu,  $\mathbf{x}$  nechť je vektor o délce  $N = |V|$ , kde  $x_i = 1$  pokud uzel  $i$  patří do podgrafu  $A$ , nebo  $x_i = -1$  v opačném případě. Dále definujme  $\mathbf{D}$  jako diagonální matici s prvky  $d(i)$  na diagonále a  $\mathbf{W}$  je symetrická matice kde  $\mathbf{W}_{ij} = w_{ij}$ ,

$$k = \frac{\sum_{x_i > 0} \mathbf{d}_i}{\sum_i \mathbf{d}_i} , \quad (2.17)$$

$\mathbf{1}$  je vektor velikosti  $N$  plný jedniček a nakonec  $b = \frac{k}{1-k}$ . Po matematických úpravách popsaných v [6] dojdeme ke vztahu pro minimalizaci  $Ncut$

$$\min_{\mathbf{x}} Ncut(\mathbf{x}) = \min_{\mathbf{y}} \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}} , \quad (2.18)$$

kde  $\mathbf{y} = (\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x})$ , na prvek  $\mathbf{y}$  jsou zároveň kladena omezení a to  $\mathbf{y}(i) \in \{1, -b\}$  a  $\mathbf{y}^T \mathbf{D} \mathbf{1} = 0$ . Pokud  $\mathbf{y}$  uvolníme do množiny reálných čísel, můžeme minimalizovat nalezením vlastních čísel pro systém

$$(\mathbf{D} - \mathbf{W}) \mathbf{y} = \lambda \mathbf{D} \mathbf{y} , \quad (2.19)$$

v práci [6] je ukázáno, že minimalizace dosáhneme vlastním vektorem s druhými nejmenšími vlastními čísly.

## Algoritmus

Průběh metody *Normalized cut* je shrnut ve čtyřech krocích v algoritmu č.

1. Výsledkem je automatická úplná segmentace obrazu jak je možno vidět na obrázku č. 2.4.

---

**Algorithm 1** algoritmus *Normalized cut*

---

- 1: Pro obraz sestrojme graf  $G = (V, E)$ , a určíme váhy hran mezi dvěma vrcholy tak, aby odpovídali podobnosti mezi těmito vrcholy.
  - 2: Vyřešme  $(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}$  pro vlastní vektory s nejmenšími vlastními čísly.
  - 3: Použijme vlastní vektor s druhými nejmenšími vlastními čísly k rozdělení grafu na dva podgrafy.
  - 4: Rozhodněme zda daný podgraf má být rozdělen a rekurzivně dělme segmenty je-li to nutné.
- 

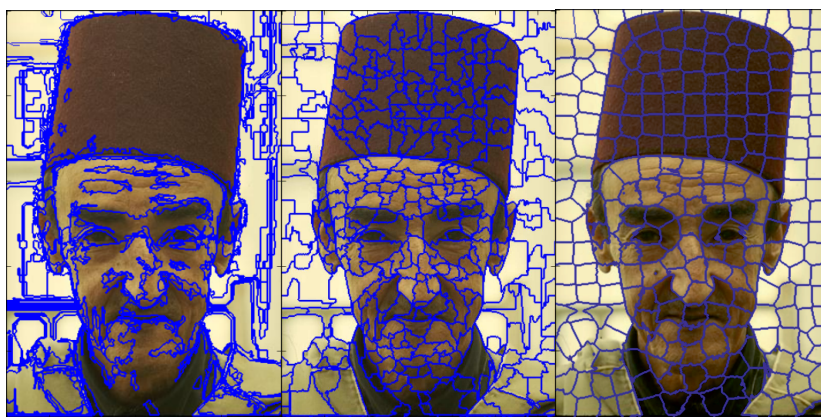


Obrázek 2.4: Segmentace metodou Normalized cut.

## 2.2 Superpixely

### 2.2.1 Úvod

Jelikož pro segmentaci využívám metodu pracující s grafem, je výhodné tento graf zmenšit co do počtu uzlů, pokud možno při zachování rozlišovací schopnosti. Pro tento účel lze použít princip superpixelů. Jedná se o shluky pixelů které jsou si podobné a jsou zároveň blízko sebe v  $x, y$  rovině. Pro účely tvorby superpixelů jsem otestoval tři metody (viz obrázek č. 2.5), *Felzenszwalb* se ukázal jako nevhodný jelikož vzniklé superpixely příliš nerespektovali hrany v obraze. Také byli natěsnány na sebe, což by vedlo k problémům při následné konstrukci grafu. Obě metody, *Quickshift* a *SLIC*, již správně respektují hrany v obraze, avšak pro další zpracování byla vybrána metoda *SLIC*, jelikož *Quickshift* vytváří příliš nepravidelné superpixely. Algoritmus *SLIC* (simple linear iterative clustering), je se vlastně upravené *k-means* shlukování. Vzniklé superpixely jsou přibližně stejně velké a mají kompaktní tvar viz obrázek č. 2.6.



Obrázek 2.5: Zleva doprava ukázka superpixelů vytvořených metodami Felzenszwalb, Quickshift a SLIC.



Obrázek 2.6: Ukázka superpixelů vytvořených metodou SLIC.

### 2.2.2 SLIC

*SLIC* je poměrně mladá metoda představená v technické zprávě [7]. *SLIC* metoda generuje shluky pixelů v závislosti na jejich barevné podobnosti a jejich vzájemné vzdálenosti. Tato vzájemná podobnost je počítána v pěti dimenzionálním prostoru  $[labxy]$  kde  $[lab]$  je barva daného pixelu v *CIELAB* barevném prostoru a  $[xy]$  jsou souřadnice v obraze. Jelikož je vzdálenost mezi dvěma barvami v *CIELAB* omezená, zatímco vzdálenost v  $xy$  rovině je závislá na velikosti obrazu, nelze ve vzniklém pěti dimenzionálním prostoru používat pouhou euklidovskou vzdálenost.

#### Měření vzdálenosti

Algoritmus *SLIC* se snaží vyprodukovat  $K$  superpixelů přibližně stejné velikosti. Jestliže má obraz  $N$  pixelů, vychází přibližně  $N/K$  pixelů na každý superpixel. Uvažujeme-li přibližně rovnoměrné rozdělení superpixelů v obraze, vychází nám středy superpixelů do mřížky s intervaly  $S = \sqrt{N/K}$ . V prvním kroku algoritmu jsou počáteční středy shluků  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  kde  $k = \langle 1, K \rangle$  umístěny do mřížky s intervaly  $S$ . Jelikož je prostorový rozsah

superpixelu přibližně  $S^2$ , lze předpokládat, že pixely příslušící danému střehu shluku se budou nalézat v  $2S \times 2S$  okolí kolem tohoto středu.

Z důvodu nepoměru mezi barevnou vzdáleností v *CIELAB* prostoru a vzdáleností v *xy* prostoru používá algoritmus místo euklidovské vzdálenosti, vzdálenost  $D_s$  definovanou takto

$$\begin{aligned} d_{lab} &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\ d_{xy} &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ D_s &= d_{lab} + \frac{m}{S} d_{xy}, \end{aligned} \quad (2.20)$$

kde  $D_s$  je suma euklidovských vzdáleností v *lab* barevném prostoru a *xy* prostoru, normovaná intervalem mřížky  $S$ . Proměnná  $m$  umožňuje ovlivňovat kompaktnost superpixelů. Její základní hodnota v algoritmu je  $m = 10$ .

## Algoritmus

Průběh algoritmu *SLIC* je shrnut pseudokódem v algoritmu č. 2. Algoritmus začíná umístěním  $K$  středů shluků do rovnoměrné mřížky. Následuje jejich posun do místa nejnižšího gradientu v  $3 \times 3$  okolí, tímto je zabráněno umístění středů na hrany v obraze a také je snížena šance že bude střed umístěn na zašumění pixel. Gradienty lze spočítat následovně

$$G(x, y) = \| \mathbf{I}(x + 1, y) - \mathbf{I}(x - 1, y) \|^2 + \| \mathbf{I}(x, y + 1) - \mathbf{I}(x, y - 1) \|^2 \quad (2.21)$$

kde  $\mathbf{I}(x, y)$  je *lab* vektor pixelu na pozici  $(x, y)$ , a  $\| \cdot \|$  je euklidovská norma. Takto je zahrnuta jak informace o barvě tak o intenzitě pixelu. Každý pixel na obraze je poté přiřazen k nejbližšímu středu shluku v jehož poli působnosti ( $2S \times 2S$ ) se nachází. Poté jsou přepočítány středy shluků jako průměr *labxy* vektorů všech pixelů příslušících těmto shlukům. Celý proces je iterativně opakován dokud se změny pozic středů shluků nedostanou pod předem

stanovenou mez. Na konci se může stát že některé pixely nejsou přímo napojeny na shluky k nimž přísluší, a proto se provede připojení těchto oblastí ke shlukům které jim jsou nejbližší, tím je zajištěna kompaktnost jednotlivých superpixelů.

---

**Algorithm 2** *SLIC* algoritmus

---

- 1: Inicializace středů shluků  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  do mřížky s intervaly  $S$ .
  - 2: Přesun středu do nejnižšího gradientu v  $n \times n$  okolí (rovnice. 2.21).
  - 3: **repeat**
  - 4:     **for** každý střed shluku  $C_k$  **do**
  - 5:         Připoj nejlépe se hodící pixely z  $2S \times 2S$  okolí okolo středu shluku k danému shluku vzhledem k normě (rovnice 2.20).
  - 6:     **end for**
  - 7:     Vypočti nové středy shluků a zbytkovou chybu  $E$  (  $L_1$  vzdálenost mezi předchozími a novými středy shluků )
  - 8: **until**  $E \leq$  zvolený práh
  - 9: Zajisti kompaktnost superpixelů
- 

## 2.3 Ohodnocení segmentace

### 2.3.1 Úvod

Po provedení segmentace je dobré ohodnotit kvalitu tohoto úkonu. Pro tento účel jsou zapotřebí další segmentace daného obrazu, které můžeme prohlásit za správné. K tomu se používají obvykle segmentace provedené člověkem. Avšak každý člověk rozdělí obraz na jiný počet segmentů, protože považuje za důležité jiné detaily na obraze. Z tohoto důvodu je nutné aby metoda



srovnávající segmentace nebyla náchylná na různý počet segmentů v obraze. Pro tento případ představil vhodnou metodu William Rand [8], jeho metoda převádí problém porovnávání dvou oddílů s možným rozdílným počtem tříd, na problém porovnávání jednotlivých párů a jejich příslušnosti k daným třídám. Tento postup je dále rozveden pro potřeby porovnání segmentací obrazu v práci [9], kde je také představena metoda *PR* index (*Probabilistic Rand* index).

### 2.3.2 Probabilistic Rand index

Předpokládejme množinu ručních segmentací  $\{S_1, S_2, \dots, S_K\}$  obrazu  $X = \{x_1, x_2, \dots, x_i, \dots, x_N\}$  indexy označují jeden z  $N$  pixelů obrazu. Necht  $S$  je segmentace jež má být porovnávána s množinou ručních segmentací. Dále označme příslušnost k segmentu u pixelu  $x_i$ , v případě segmentace  $S$  bude označení  $l_i$ , pokud se jedná o segmentaci z množiny ručních segmentací bude označení  $l_i^{(k)}$  pro segmentaci  $S_k$ . Nakonec zavedme pravděpodobnost závislosti mezi označeními příslušnosti k segmentu u páru pixelů  $x_i$  a  $x_j$

$$\hat{P}(\hat{l}_i = \hat{l}_j) = \frac{1}{K} \sum_{k=1}^K \mathbf{I}(l_i^{(k)} = l_j^{(k)}) \quad (2.22)$$

$$\hat{P}(\hat{l}_i \neq \hat{l}_j) = \frac{1}{K} \sum_{k=1}^K \mathbf{I}(l_i^{(k)} \neq l_j^{(k)}) = 1 - \hat{P}(\hat{l}_i = \hat{l}_j) . \quad (2.23)$$

Pravděpodobnost  $\hat{P}(\hat{l}_i = \hat{l}_j)$  lze považovat za pravděpodobnost toho, že pixely  $x_i$  a  $x_j$  mají opravdu patřit do stejného segmentu. Funkce  $\mathbf{I}(l_i^{(k)} = l_j^{(k)})$  vrací jedničku jestliže se označení příslušnosti  $l_i^{(k)}$  a  $l_j^{(k)}$  rovnají a nulu v opačném případě. *PR* index pro segmentaci  $S$  při porovnání s ručními segmentacemi  $\{S_1, S_2, \dots, S_K\}$  lze poté zavést ve tvaru

$$PR(S, S_{\{1 \dots K\}}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i \neq j}} [\mathbf{I}(l_i = l_j) \hat{P}(\hat{l}_i = \hat{l}_j) + \mathbf{I}(l_i \neq l_j) \hat{P}(\hat{l}_i \neq \hat{l}_j)] , \quad (2.24)$$

tato míra dosahuje hodnot mezi  $\langle 0, 1 \rangle$ . Kde 0 znamená že segmentace  $S$  a ruční segmentace  $\{S_1, S_2, \dots, S_K\}$  nemají žádnou podobnost, to nastane v případě že segmentace  $S$  rozdělila celý obraz pouze do jednoho segmentu a všechny z ručních segmentací  $\{S_1, S_2, \dots, S_K\}$  rozdělili obraz na segmenty tak, že každý pixel je ve vlastním segmentu. Naopak hodnota 1 značí že segmentace  $S$  a všechny ruční segmentace  $\{S_1, S_2, \dots, S_K\}$  jsou totožné.

# Kapitola 3

## Praktická část

### 3.1 Dílčí úlohy

#### 3.1.1 Úvod

V této části se budu věnovat jednotlivým krokům prováděným před samotnou segmentací. Jedná se o předzpracování obrazu, tato část má velký vliv na kvalitu samotné segmentace. Dále zde popíši mnou použitou upravenou metodu *SPRW*. Nakonec se budu věnovat ohodnocení kvality segmentace.

#### 3.1.2 Obraz

Obrazová data pro segmentaci jsem získal z on-line knihovny *Boundary Detection Benchmark*<sup>1</sup>, v této databázi jsou jak obrazy v rozlišení vhodném pro použití v metodách počítačového vidění, tak se zde pro účely segmentace nacházejí obrazy segmentované člověkem ke každému obrazu minimálně pět různých segmentací každá od jiného člověka. Toto je nezbytné pro použití *PR* indexu ohodnocujícího kvalitu segmentace. Prvním krokem při zpraco-

---

<sup>1</sup><http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

vání obrazu je jeho načtení do počítače, k tomuto účelu jsem zvolil knihovnu *OpenCV*<sup>2</sup>, která umožňuje rychlé načtení obrazových formátů jako jsou Windows bitmapy, JPEG, PNG, TIFF a další, do formátu dvourozměrných polí, která jsou výhodná pro další zpracování.

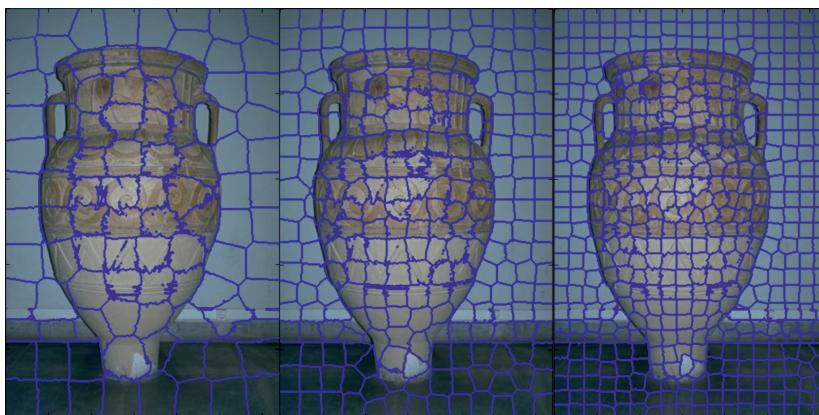
### 3.1.3 Superpixely

Superpixely usnadňují a urychlují práci s obrazem jelikož se zmenší počet bodů v obraze. Pro tvorbu superpixelů jsem použil metodu *SLIC* z knihovny *scikit-image*<sup>3</sup> konkrétně její verzi *SLIC0*, které má jediný vstupní parametr a to požadovaný přibližný počet superpixelů v obraze, u běžné metody *SLIC* je dalším parametrem kompaktnost výsledných superpixelů. V této verzi metody je kompaktnost nastavována automaticky pro každý superpixel podle členitosti v okolí prvotních středů shluků. Počet superpixelů v obraze jsem udržoval pro všechny testované obrazy na stejné hodnotě. Pokusy jsem došel k počtu 250 pixelů, což se jevílo dostatečné pro obrazy v rozlišení jaké je k dispozici v knihovně *Boundary Detection Benchmark*, jelikož s tímto počtem již byly hrany superpixelů, v drtivé většině případů umístěny na hranách objektů v obraze. Další zvyšování počtu superpixelů by vedlo pouze k většímu grafu a tím pádem pomalejšímu výpočtu. Srovnání počtu superixelů v obraze je vidět na obrázku č. 3.1. Pro další zpracování je nesmírně důležité aby hranice superpixelů ležely na hranách v obraze, jelikož pro segmentaci používám upravenou metodu *Random walker*, která segmentuje právě po celých superpixelech. Pokud byl obraz příliš členitý stalo se, že v některých částech obrazu neležely hranice superpixelů na hranách obrazu a to vedlo k chybám při segmentaci.

---

<sup>2</sup><http://docs.opencv.org/>

<sup>3</sup><http://scikit-image.org/>



Obrázek 3.1: Ukázka různého počtu superpixelů vytvořených metodou *SLIC0*.

### 3.1.4 Random walker

Jak jsem již psal metoda *Random walker* pracuje s obrazem jako s grafem, kde pixely představují vrcholy grafu a mezi nimi jsou umístěny hrany buď ve formě čtyřokolí nebo osmiokolí viz obrázek č. 3.3. Běžná metoda *Random walker* tak je implementovaná v knihovně *scikit-image* v modulu *segmentation* převede body obrazu na vrcholy grafu, tím ovšem vzniká velký graf který je poté výpočetně náročný. Proto jsem se rozhodl využít superpixely i pro tvorbu grafu, použitého pro výpočet náhodné procházky.

Při tvorbě grafu ze superpixelů již nelze hovořit o čtyř nebo osmiokolí jelikož superpixely nejsou umístěni v pravidelné mřížce jako pixely. Proto jsem pro nalezení sousedních superpixelů z důvodů vytvoření hrany mezi nimi použil dilataci. V cyklu jsem prošel každý superpixel vždy ho dilatoval a uložil si do jakých superpixelů zasahoval po dilataci, takto jsem vytvořil seznam sousedních superpixelů pro každý z nich, tento postup je zobrazen na obrázku č. 3.2. Vlevo je znázorněn superpixel jehož sousedy hledáme, uprostřed je od dilatovaného superpixelu odečten původní a tím je ponechána

část zasahující do sousedních superpixelů a nakonec vpravo jsou označeny sousední superpixely. Ze seznamu sousedních superpixelů jsem vytvořil matici hran a pro ní jsem vypočetl váhy jednotlivých hran. K tomuto účelu bylo nutné spočítat průměrnou intenzitu pixelů v každém superpixelu abych dostal jen jednu číselnou intenzitu každého superpixelu z čehož jsem vypočítal váhu jednotlivých hran pomocí vztahu

$$w_{ij} = e^{-\beta|z_i - z_j|} , \quad (3.1)$$

kde  $z_i$  a  $z_j$  je intenzita superpixelu  $i$  resp.  $j$ . proměnou  $\beta$  jsem volil s ohledem na barevnou členitost obrazu a tak jsem ji počítal pomocí vztahu

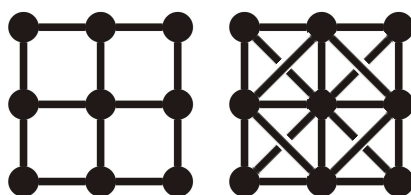
$$\beta = \frac{50}{10 * \sigma} , \quad (3.2)$$

kde  $\sigma$  je směrodatná odchylka intenzit všech pixelů v obraze. Čitatel v tomto vztahu jsem volil experimentálně různá nastavení jsou vidět na obrázku č. 3.4.

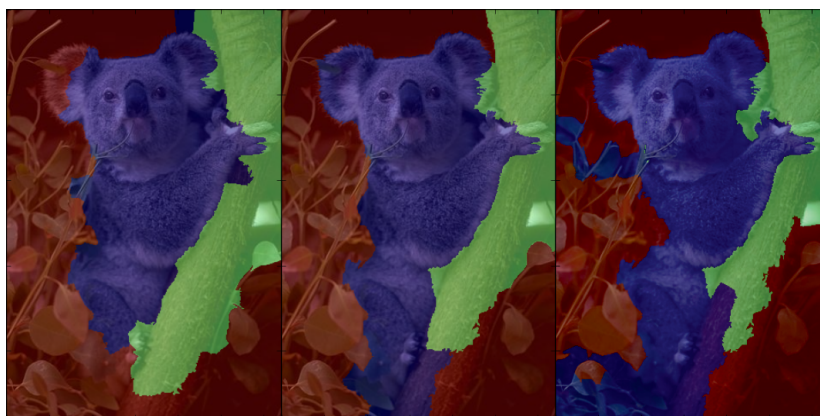


Obrázek 3.2: Hledání sousedu dilatací superpixelu. Vlevo superpixel jehož sousedy hledáme, uprostřed dilatace, vpravo sousední superpixely.

Jako další zlepšení se ukázalo použití superpixelů při označování počátečních bodů (seedů) před startem metody. Pokud tedy uživatel označí pixely znázorňující objekty v obraze, označí se i celé superpixely do kterých označené body patří, ukázka viz obrázek č. 3.5. Toto se ukázalo vhodné poté co



Obrázek 3.3: Ukázka čtyřokolí a osmiokolí používaného pro spojení vrcholů grafu hranami.



Obrázek 3.4: Ukázka vlivu volby čitatele vztahu 3.2. Zleva 10, 50, 100.

jsem experimentálně zjistil, že více známých bodů při startu metody *random walker* zvyšuje přesnost provedené segmentace, toto nastavení samozřejmě zlepšuje pouze neupravenou, původní metodu *Random walker* a také metodu *GrabCut*. Pro mnou implementovanou, upravenou metodu *Random walker*, toto rozšíření označení na celý superpixel, před startem metody nemá vliv, jelikož by se automaticky provedlo při tvorbě grafu ze superpixelů uvnitř metody.

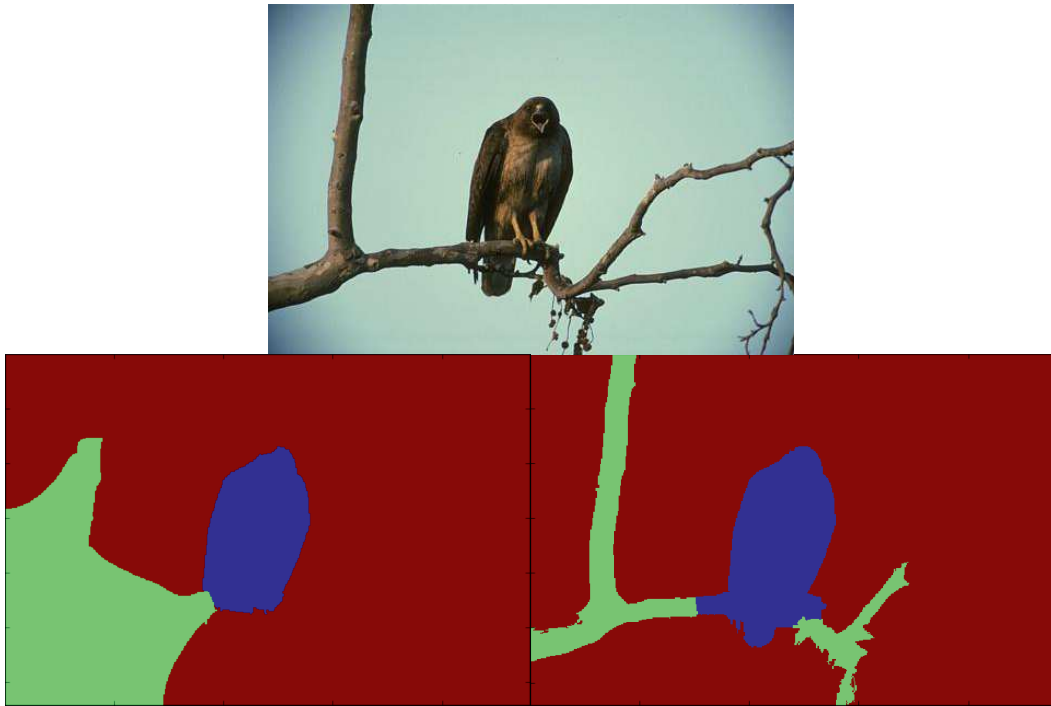


Obrázek 3.5: Ukázka prvotních označení (seedu) s použitím superpixelů.

### 3.1.5 Probabilistic Rand index

Pro výpočet PR indexu jsou zapotřebí segmentace, které můžeme považovat za správné, k tomuto účelu jsem ke každému obrázku z testovací sady získal na stránkách *Boundary Detection Benchmark* pět různých segmentací provedených lidmi. Tyto jsem použil při ohodnocení mnou použitých metod, kdy jsem výslednou segmentaci metody srovnával pomocí algoritmu pro výpočet PR indexu právě s těmito pěti “správnými” segmentacemi. výsledky byli někdy poněkud zvláštní viz obrázek č. 3.6. Segmentace vlevo dosáhla vyššího ohodnocení konkrétně  $PR = 0.573$  zatímco segmentace vpravo, kterou bych pohledem hodnotil jako přesnější je ohodnocena pouze  $PR = 0.504$ . Ještě zvláštnější se může zdát hodnocení u obrázku houby v příloze viz obrázek B.13.





Obrázek 3.6: nahoře originál, vlevo segmentace s  $PR = 0.573$ , vpravo segmentace  $PR = 0.504$ .

## 3.2 Srovnávací test

### 3.2.1 Úvod

Ve srovnávacím testu jsem porovnával kvalitu segmentace mnou upravené metody *Random walker* pracující se superpixely se segmentací provedenou postupně metodami *Random walker*, *GrabCut* a *Normalized cut*. Test jsem prováděl na sadě dvaceti obrázků z *Boundary Detection Benchmark*. Ohodnocení segmentace jsem poté prováděl srovnáním s pěti ručními segmentacemi.

### 3.2.2 Postup

Po načtení obrazu jsem rozdělil obraz na superpixely a poté jsem provedl počáteční označení objektů v obraze. Pro každý obraz z testovací sady jsem si uložil deset různých označení jako startovací data od uživatele. Označením je myšlena matice o rozměrech schodných s obrazem, v této matici jsou celá čísla, kde každá číslice značí objekt v obraze a číslice 0 značí oblast u které je teprve nutno určit příslušnost k objektu pomocí některé ze segmentačních metod. Pro test jsem zvolil tři možná označení tedy dva objekty a pozadí. Samozřejmě by šel použit větší počet, ale možnost označení dvou objektů se ukázala jako dostatečná pro testovací sadu obrazů. Navíc více možných označení zvyšuje náročnost kladenou na uživatele metody. Označení probíhá tahem myši při stisknutém levém tlačítku, číslo značky se volí tlačítkem na klávesnici. Výsledek takového označení je možno vidět na obrázku č. 3.7, kde vlevo jsou vidět čáry provedené uživatelem a kolem nich je výsledné označení za pomoci superpixelů, kdy je označen vždy celý superpixel abychom získali více dat pro segmentační metody.



Obrázek 3.7: Ukázka prvotního označení objektů v obrazech.

Takto uložená data byla použita pro metody *Random walker* a *Random*

*walker* se superpixely. Pro metodu *GrabCut* musela být data pozměněna co se hodnoty jednotlivých bodů týče. Maska metody *GrabCut* používá označení následovně

- 0 pro označení pozadí obrazu
- 1 pro označení popředí obrazu
- 2 pro označení spíše pozadí obrazu
- 3 pro označení spíše popředí obrazu ,

kde hodnoty 0 a 1 jsou neměnné zatímco hodnoty 2 a 3 jsou používány spíše jako neznámé a segmentační metoda pro ně určuje zda spadají do popředí či pozadí a tak jsem původní označení přeznačil tak že to co bylo označeno jako objekty 1 a 2 jsem přeznačil na hodnotu 1 čili popředí a to co bylo označeno třetí značkou kterou jsem používal pro pozadí jsem přeznačkoval na hodnotu 0 čili pozadí a neoznačený zbytek obrázku byl v masce označen jako spíše pozadí, hodnota 2. S tímto nastavením pracovala metoda *GrabCut* úspěšně.

Když jsem měl tedy uložena data od uživatele mohl jsem spustit jednotlivé metody pro segmentaci. Pro každou metodu jsem měřil čas provedení a jak výsledek segmentace tak časový údaj jsem si uložil pro použití při vyhodnocení.

Pro metodu *Random walker* jsem zvolil implementaci z knihovny *scikit-image* modulu *segmentation*. jako vstup byl použit šedotónový obraz, data od uživatele vytvořená v předchozím kroku. hodnota parametru pro výpočet vah  $\beta$  byla ponechána nezměněna tak jak je implementována. Pro vyřešení soustavy rovnic byla zvolena metoda '*cg\_mg*' která nejprve použije multigrdní řešení pro transformaci matice aby snížila její podmíněnost a poté použije pro řešení metodu konjugovaných gradientů [3].

U upravené verze *Random walker* se superpixely jsem použil stejné me-

tody pro řešení výsledné soustavy rovnic aby výsledky byli co nejlépe porovnatelné. jako vstup byl opět použit šedotónový obraz a uživatelská data připravená dříve.

Pro metodu *GrabCut* jsem použil implementaci z knihovny *OpenCV*, vstupem v tomto případě musí být tříkanálový barevný obraz, aby metoda neměla výhodu informace o barevných kanálech uložil jsem do každého kanálu barvy stejný šedotónový obraz, jaký byl použit v předchozích metodách. Počet iterací *Graph cut* algoritmu byl omezen na pět.

Metoda *Normalized cut* je implementována v knihovně *scikit-image*. Pro její použití je nutné vytvořit graf sousledností, k tomuto účelu je v této knihovně také implementován algoritmus. Jeho vstupem je barevný obraz, zde jsem použil stejný trik s šedotónovým obrazem jako o metodu výše, a matice se superpixely, na výstupu je graf použitelný pro metodu *Normalized cut*. Tvorba tohoto grafu nebyla zahrnuta do měření času chodu této metody, jelikož i tak je tato metoda nejpomalejší ze zde použitých metod a její zahrnutí vedlo v některých případech k zamrznutí programu.

### 3.2.3 Vyhodnocení

Porovnání výsledků jednotlivých metod je možné vidět v tabulkách a na obrázcích v příloze A resp. B. Jednotlivé segmentace jsou v koláži řazeny zleva doprava, odshora dolů v pořadí, originální obrázek, segmentace metodou *Random walker (RW)*, segmentace upravenou metodou *Random walker se superpixely (SPRW)*, segmentace metodou *GrabCut (GC)* a segmentace metodou *Normalized cut (NC)*. V popisu k obrázku jsou uvedeny hodnoty *PR* indexu pro zobrazené segmentace. Jelikož mám k dispozici deset sad segmentací pro každý obraz, vybral jsem pro ilustraci do této práce vždy tu sadu, ve které se nacházela nejlépe hodnocená segmentace ze všech metod

Metoda	PR index
SPRW	45 %
Random walker	20 %
Normalized cut	20 %
GrabCut	15 %

Tabulka 3.1: Procentuální vítězství metod v testu. Vztaženo k 20 testovacím obrazům.

z celé sady, k této nejlepší segmentaci pak jsou ukázány ostatní segmentace pořízené pomocí stejných dat od uživatele. U obrázků koně, klokana, totemů a dvou lidí bohužel není k dispozici výsledek segmentace metodou *Normalized cut*, jelikož tato metoda nekonvergovala ani po uplynutí nastaveného maximálního počtu iterací. Výsledné srovnání metod dopadlo následovně.

Z pohledu hodnocení *PR* indexem je při porovnání všech metod výsledek nejlepších metod uveden v tabulce č. 3.1, porovnáván byl průměrný *PR* index pro každý obraz. Rozptyly mezi hodnotami z jednotlivých průběhů pro různá uživatelská data jsou velmi malé, důvodem pro to pravděpodobně bude, že všechny sady uživatelských dat pocházejí od jednoho uživatele.

Z pohledu rychlosti metod je jasným vítězem původní algoritmus *Random walker* a to v ve všech 20 obrazech. Předpoklad, že použití superpixelů a tím pádem zmenšení grafu pro výpočet *Random walker*, bude mít pozitivní vliv na rychlost tohoto výpočtu, se ukázal jako lichý. Bohužel to co algoritmus ušetřil na výpočtu, nevyvážilo přidanou zátěž, která vznikla při počítání grafu ze superpixelů. Nejpomalejším článkem výpočtu se ukázalo počítání průměrné intenzity uvnitř každého superpixelu. Další pomalou částí bylo zjišťování sousedních superpixelů, jelikož se zde prováděla dilatace pro každý superpixel. Kdyby byl tento graf připraven předem a dodán jako vstupní data

metodě *Random walker*, bylo by možno pozorovat několikanásobné zrychlení výpočtu této metody. Kde tato rychlost není závislá na velikosti obrazu, ale pouze na počtu použitých superpixelů. Tento počet je nutno zvyšovat ne v závislosti s rozlišením obrazu ale s jeho členitostí či složitostí.

Při porovnání mezi původní a upravenou verzí metody *Random walker* je *PR* index vyšší ve 128 z 200 případů u upravené metody a ve dvou případech je index shodný. Avšak v případech kdy upravená metoda vyhrála, je rozdíl v *PR* indexech dvakrát vyšší, než v případech kdy vyhrála původní metoda. Samozřejmě že v časové oblasti je na tom mnohem lépe původní metoda s průměrným časem výpočtu napříč všemi obrazy  $t_{RW} = 1,509s$  zatímco upravená verze dosáhla průměrného času  $t_{SPRW} = 4,2372s$ , metoda *GrabCut* dosáhla průměrného času výpočtu  $t_{gc} = 4,1372s$  a metodě *Normalized cut* trval výpočet v průměru bezmála devět minut, ale doba se dosti různila obraz od obrazu, nejrychleji metoda proběhla za  $3,48s$  zatímco nejpomalejší průběh činí  $3455,13s$ .

# Kapitola 4

## Závěr

Hlavním cílem práce bylo zlepšit chování metody *Random walker* při použití superpixelů, toho mělo být dosaženo implementací upravené metody *Random walker*, která by pracovala s grafem tvořeným právě těmito superpixely. Pro správnou funkci této metody muselo být provedeno několik kroků.

Nejdříve by digitalizovaný obraz nahrán jako matice, aby s ním bylo možno pracovat. Dále byla použita metoda *SLIC0* pro tvorbu superpixelů, jejich počet na obraz byl experimentálně zvolen na 250 (viz obrázek č. 3.1). Tyto superpixely byli poté použity při získávání prvotních označení objektů a pozadí od uživatele, a to k rozšíření označení na celý superpixel. Tato označení byla uložena ve formátu matic, aby mohla být použita jako vstup segmentačních metod.

Při implementaci metody *Random walker* pracující se superpixely bylo nutno vyřešit tvorbu grafu ze superpixelů. Toto bylo provedeno dilatací jednotlivých superpixelů a následném určení sousedů nacházejících se v této dilatované oblasti (viz obrázek č. 3.2). Pro výpočet vah takto vzniklých hran byla použita průměrná intenzita všech pixelů v daném superpixelu.

Takto vzniklý graf skutečně urychlil výpočet soustavy lineárních rovnic

2.6. Bohužel čas potřebný pro tvorbu použitého grafu převýšil tento ušetřený čas na výpočtu.

S úspěšně implementovanou metodou *Random walker* pracující se superpixely byl proveden srovnávací test. V tomto testu byla srovnávána segmentace touto implementovanou metodou s originální metodou *Random walker*, dále s metodou *GrabCut* a nakonec byla zahrnuta i automatická metoda *Normalized cut*.

Segmentace byly prováděny na sadě dvaceti obrazů kde pro každý byla k dispozici desítka různých označení objektů od uživatele. Hodnocení úspěšnosti segmentací bylo provedeno metodou *Probabilistic Rand index*. Tato metoda srovnává testovanou segmentaci s několika segmentacemi provedenými lidmi. Výhodou této metody je nezávislost na počtu segmentů v obraze.

U devíti z dvaceti sad obrazů dosáhla implementovaná metoda nejvyššího skóre, tři a čtyři z dvaceti sad nejlépe segmentovali metody *GrabCut* a *Normalized cut* a u posledních čtyř byla nejlepší původní metoda *Random walker*. Tato metoda však byla nejrychlejší ve všech případech. Kompletní výsledky jsou uvedeny v tabulkách v příloze A. V příloze B jsou ukázky segmentací jednotlivými metodami, je zde vždy zobrazena jedna z deseti sad segmentací a to tak aby byla zobrazena sada s nejlépe hodnocenou segmentací pro daný obraz. V popise u obrázků jsou hodnoty *PR* indexu pro jednotlivé segmentace.

Vůbec nejlepšího výsledku dosáhla metoda *Normalized cut* v případě obrazu nazvaného Bird (obrázek B.1). Avšak lidskému oku to tak nemusí vůbec připadat, a tak i když je Probabilistic Rand index zajímavou metodou pro srovnání segmentací, je to stále pouze matematický nástroj a lidské vnímání se jen těžko převádí do roviny čísel.



# Literatura

- [1] Leo Grady and Gareth Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pages 230–245. Springer, 2004.
- [2] Leo Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, 2006.
- [3] Magnus Rudolph Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. 1952.
- [4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grab-cut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.
- [5] Yuri Y Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.
- [6] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

- [7] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, EPFL, 2010.
- [8] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.
- [9] Ranjith Unnikrishnan and Martial Hebert. Measures of similarity. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 394–394. IEEE, 2005.

# Seznam obrázků

1.1	Nahoře: originál, vlevo: segmentace random walker, vpravo: upravenou metodou. . . . .	5
2.1	Segmentace metodou Random walker, respektování neúplných hran. . . . .	8
2.2	Metoda Random walker, respektování neúplných hran. . . . .	8
2.3	Segmentace minimálním řezem grafu. Převzato z: <a href="http://www.ondrej-danek.net/images/vyzkum/graph.png">http://www.ondrej-danek.net/images/vyzkum/graph.png</a> . . . . .	13
2.4	Segmentace metodou Normalized cut. . . . .	16
2.5	Zleva doprava ukázka superpixelů vytvořených metodami Felzenszwalb, Quickshift a SLIC. . . . .	17
2.6	Ukázka superpixelů vytvořených metodou SLIC. . . . .	18
3.1	Ukázka různého počtu superpixelů vytvořených metodou <i>SLIC0</i> . . . . .	25
3.2	Hledání sousedu dilatací superpixelu. Vlevo superpixel jehož sousedy hledáme, uprostřed dilatace, vpravo sousední superpixely. . . . .	26
3.3	Ukázka čtyřokolí a osmiokolí používaného pro spojení vrcholů grafu hranami. . . . .	27
3.4	Ukázka vlivu volby čísel vztahu 3.2. Zleva 10, 50, 100. . . . .	27
3.5	Ukázka prvotních označení (seedu) s použitím superpixelů. . . . .	28

3.6	nahoře originál, vlevo segmentace s $PR = 0.573$ , vpravo segmentace $PR = 0.504$ . . . . .	29
3.7	Ukázka prvotního označení objektů v obrazech. . . . .	30
B.1	Bird. RW: 0.573, SPRW: 0.501, GC: 0.546, NC: 0.959 . . . . .	49
B.2	Boat. RW: 0.845, SPRW: 0.861, GC: 0.514, NC: 0.778 . . . . .	50
B.3	Cat. RW: 0.706, SPRW: 0.700, GC: 0.619, NC: 0.483 . . . . .	50
B.4	Cow. RW: 0.800, SPRW: 0.838, GC: 0.653, NC: 0.808 . . . . .	51
B.5	Croc. RW: 0.782, SPRW: 0.771, GC: 0.837, NC: 0.565 . . . . .	51
B.6	Deer. RW: 0.868, SPRW: 0.929, GC: 0.605, NC: 0.838 . . . . .	52
B.7	Elephant. RW: 0.772, SPRW: 0.799, GC: 0.666, NC: 0.824 . . . . .	52
B.8	Fireman. RW: 0.644, SPRW: 0.677, GC: 0.608, NC: 0.648 . . . . .	53
B.9	Horse. RW: 0.628, SPRW: 0.791, GC: 0.563 . . . . .	53
B.10	Kangaroo. RW: 0.641, SPRW: 0.612, GC: 0.565 . . . . .	54
B.11	Koala. RW: 0.781, SPRW: 0.776, GC: 0.619, NC: 0.781 . . . . .	54
B.12	Man with hat. RW: 0.791, SPRW: 0.882, GC: 0.721, NC: 0.837 . . . . .	55
B.13	Mushroom. RW: 0.510, SPRW: 0.399, GC: 0.421, NC: 0.726 . . . . .	56
B.14	Ostrich. RW: 0.510, SPRW: 0.655, GC: 0.816, NC: 0.594 . . . . .	56
B.15	Panther. RW: 0.716, SPRW: 0.834, GC: 0.785, NC: 0.423 . . . . .	57
B.16	Penguin. RW: 0.896, SPRW: 0.869, GC: 0.692, NC: 0.678 . . . . .	57
B.17	Soldier. RW: 0.860, SPRW: 0.867, GC: 0.619, NC: 0.783 . . . . .	58
B.18	Totems. RW: 0.447, SPRW: 0.511, GC: 0.590 . . . . .	58
B.19	Two people. RW: 0.719, SPRW: 0.736, GC: 0.605 . . . . .	59
B.20	Vase. RW: 0.837, SPRW: 0.879, GC: 0.730, NC: 0.552 . . . . .	59

# Příloha A

## Tabulky srovnávací test

Tabulka A.1: Bird

Sada	BIRD											E(x)	var(x)
Značkování		1	2	3	4	5	6	7	8	9	10		
Random walker	PR index	0,539	0,546	0,543	0,547	0,518	0,518	0,533	0,573	0,532	0,48	0,5329	5,9566e-4
	čas	1,43	1,54	1,46	1,63	1,44	1,61	1,64	1,55	1,58	1,38	1,526	0,0085
SPRW	PR index	0,504	0,504	0,502	0,504	0,501	0,5	0,504	0,501	0,504	0,5	0,5024	3,1556e-6
	čas	2,77	2,77	2,78	2,75	2,76	2,84	2,8	2,76	2,76	2,77	2,776	6,9333e-4
GrabCut	PR index	0,548	0,548	0,548	0,548	0,543	0,546	0,546	0,546	0,548	0,546	0,5467	2,6778e-6
	čas	1,49	1,59	1,57	1,63	1,64	1,63	1,65	1,65	1,55	1,58	1,598	0,0027
Normalized cut	PR index	0,959											
	čas	8,86											

Tabulka A.2: Boat

Sada	BOAT												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,845	0,852	0,856	0,841	0,849	0,805	0,851	0,849	0,836	0,841	0,8425	2,0983e-4
	čas	1,44	1,36	1,58	1,53	1,52	1,52	1,46	1,37	1,45	1,48	1,471	0,0049
SPRW	PR index	0,861	0,855	0,85	0,858	0,857	0,856	0,851	0,851	0,846	0,846	0,8531	2,5878e-5
	čas	3,36	3,37	3,42	3,35	3,4	3,38	3,4	3,4	3,5	3,38	3,396	0,0018
GrabCut	PR index	0,514	0,483	0,518	0,53	0,515	0,53	0,485	0,522	0,517	0,518	0,5132	2,6818e-4
	čas	2,48	2,42	2,64	2,12	2,46	2,29	2,38	2,68	2,66	2,5	2,463	0,0306
Normalized cut	PR index	0,778											
	čas	112,49											

Tabulka A.3: Cat

Sada	CAT												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,69	0,703	0,635	0,657	0,675	0,706	0,675	0,631	0,696	0,626	0,6694	9,2871e-4
	čas	1,11	1,14	1,44	1,37	1,39	1,48	1,58	1,66	1,56	1,61	1,434	0,0354
SPRW	PR index	0,668	0,652	0,578	0,648	0,61	0,7	0,65	0,604	0,666	0,62	0,6396	0,0013
	čas	4,01	4,01	4,06	3,96	3,95	3,98	4,02	4,01	4,04	4	4,004	0,0011
GrabCut	PR index	0,643	0,665	0,559	0,616	0,629	0,619	0,571	0,59	0,633	0,507	0,6032	0,0022
	čas	4,19	3,66	5,24	5,69	6,3	7,09	8,85	9,38	9,81	7,85	6,8060	4,6325
Normalized cut	PR index	0,483											
	čas	611,17											

Tabulka A.4: Cow

Sada	COW												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,774	0,766	0,8	0,805	0,823	0,779	0,786	0,796	0,782	0,816	0,7927	3,429e-4
	čas	1,28	1,39	1,4	1,39	1,34	1,4	1,39	1,39	1,43	1,36	1,377	0,0017
SPRW	PR index	0,811	0,823	0,838	0,82	0,822	0,819	0,828	0,841	0,818	0,833	0,8253	9,0678e-5
	čas	3,69	3,82	3,76	3,76	3,76	3,78	3,78	3,76	3,81	3,8	3,772	0,0013
GrabCut	PR index	0,692	0,692	0,653	0,688	0,689	0,688	0,689	0,691	0,688	0,685	0,6855	1,3494e-4
	čas	2,91	3,01	3,57	3,54	3,23	3,31	3,63	3,47	4,16	3,32	3,415	0,1241
Normalized cut	PR index	0,808											
	čas	36,89											

Tabulka A.5: Croc

Sada	CROC												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,768	0,782	0,715	0,646	0,783	0,778	0,71	0,766	0,719	0,715	0,7382	0,002
	čas	1,16	1,26	1,46	1,36	1,19	1,38	1,36	1,32	1,31	1,29	1,309	0,008
SPRW	PR index	0,691	0,771	0,68	0,678	0,786	0,749	0,686	0,776	0,766	0,754	0,7337	0,002
	čas	4,78	4,76	4,73	4,69	4,77	4,77	4,7	4,7	4,77	4,72	4,739	0,0012
GrabCut	PR index	0,724	0,837	0,807	0,55	0,787	0,541	0,695	0,802	0,689	0,751	0,7183	0,0107
	čas	3,37	6,79	9,05	6,48	4,12	6,73	4,6	8,02	4,74	4,9	5,88	3,315
Normalized cut	PR index	0,565											
	čas	938,2											

Tabulka A.6: Deer

Sada	DEER												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,868	0,851	0,869	0,738	0,778	0,826	0,837	0,825	0,655	0,792	0,8039	0,0044
	čas	1,41	1,41	1,36	1,54	1,49	1,55	1,46	1,68	1,57	1,42	1,4890	0,0093
SPRW	PR index	0,929	0,929	0,926	0,928	0,919	0,918	0,929	0,923	0,924	0,919	0,9244	2,0044e-5
	čas	2,94	2,96	3,03	2,95	2,98	2,99	2,97	3,07	2,98	3,15	3,002	0,0042
GrabCut	PR index	0,605	0,605	0,602	0,594	0,605	0,605	0,601	0,605	0,599	0,605	0,6026	1,3822e-5
	čas	2,97	3,06	3,62	3,36	3,32	3,17	3,96	3,21	3,34	3,39	3,34	0,0808
Normalized cut	PR index	0,838											
	čas	5,38											

Tabulka A.7: Elephant

Sada	ELEPHANT												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,772	0,72	0,719	0,738	0,761	0,703	0,745	0,77	0,772	0,753	0,7453	6,129e-4
	čas	1,03	1,25	1,39	1,39	1,2	1,4	1,42	1,41	1,44	1,29	1,322	0,017
SPRW	PR index	0,799	0,782	0,772	0,796	0,793	0,792	0,776	0,799	0,797	0,79	0,7896	9,36e-5
	čas	4,39	4,41	4,47	4,37	4,39	4,39	4,43	4,42	4,4	4,43	4,41	8,2222e-4
GrabCut	PR index	0,666	0,652	0,677	0,67	0,67	0,677	0,677	0,667	0,652	0,678	0,6686	9,6044e-5
	čas	1,57	1,81	1,8	2,07	1,7	1,91	1,78	1,82	1,87	1,69	1,802	0,0184
Normalized cut	PR index	0,824											
	čas	379,82											

Tabulka A.8: Fireman

Sada	FIREMAN												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,652	0,665	0,644	0,66	0,677	0,652	0,671	0,652	0,638	0,65	0,6561	1,461e-4
	čas	1,62	1,71	2,08	1,96	1,93	1,57	1,91	1,67	1,63	1,5	1,758	0,0383
SPRW	PR index	0,636	0,666	0,677	0,672	0,66	0,644	0,638	0,646	0,627	0,632	0,6498	3,1262e-4
	čas	4,25	4,26	4,26	4,26	4,25	4,32	4,3	4,28	4,26	4,28	4,272	5,2889e-4
GrabCut	PR index	0,641	0,629	0,608	0,601	0,581	0,631	0,596	0,63	0,609	0,628	0,6154	3,6871e-4
	čas	3,8	4,74	3,67	5,99	2,94	4,17	3,93	4,19	3,99	3,72	4,114	0,6443
Normalized cut	PR index	0,648											
	čas	624,75											

Tabulka A.9: Horse

Sada	HORSE												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,736	0,689	0,668	0,632	0,542	0,686	0,628	0,613	0,64	0,732	0,6566	0,0034
	čas	2,06	2,36	2,76	2,52	3,25	2,54	2,62	2,58	2,52	2,17	2,538	0,1068
SPRW	PR index	0,77	0,741	0,734	0,77	0,784	0,789	0,791	0,786	0,788	0,763	0,7716	4,1538e-4
	čas	6,52	6,56	6,57	6,6	6,53	6,57	6,57	6,52	6,47	6,53	6,544	0,0014
GrabCut	PR index	0,573	0,573	0,563	0,563	0,563	0,563	0,563	0,528	0,563	0,572	0,5624	1,6693e-4
	čas	4,91	7,54	9,89	9,53	14,89	9,52	9,03	10,34	8,65	3,29	8,759	9,8708
Normalized cut	PR index	NE											
	čas	NE											

Tabulka A.10: Kangaroo

Sada	KANGAROO												
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,594	0,629	0,631	0,562	0,581	0,598	0,641	0,56	0,61	0,587	0,5993	7,9912e-4
	čas	1,13	1,19	1,2	1,03	0,96	1,11	1,18	1,24	1,33	1,09	1,146	0,0113
SPRW	PR index	0,546	0,637	0,579	0,544	0,521	0,591	0,612	0,511	0,488	0,566	0,5595	0,0022
	čas	4,96	5,26	4,96	4,93	4,98	4,96	4,97	4,95	4,95	4,96	4,988	0,0093
GrabCut	PR index	0,57	0,61	0,546	0,559	0,554	0,597	0,565	0,555	0,403	0,61	0,5569	0,0035
	čas	6,29	5,75	4,95	4,66	5,54	6,5	3,82	7,91	4,76	4,63	5,481	1,4031
Normalized cut	PR index	NE											
	čas	NE											



Tabulka A.11: Koala

Sada	KOALA													
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)	
Random walker	PR index	0,769	0,764	0,738	0,742	0,748	0,712	0,665	0,727	0,74	0,781	0,7386	0,0011	
	čas	1,56	1,72	1,48	1,29	1,51	1,52	1,55	1,45	1,58	1,18	1,484	0,0231	
SPRW	PR index	0,769	0,771	0,728	0,746	0,732	0,762	0,709	0,756	0,713	0,776	0,7462	5,9862e-4	
	čas	4,53	4,62	4,59	4,57	4,54	4,57	4,53	4,52	4,49	4,5	4,456	0,0017	
GrabCut	PR index	0,621	0,588	0,557	0,639	0,544	0,547	0,556	0,599	0,559	0,619	0,5829	0,0012	
	čas	5,78	4,16	5,74	5,18	4,44	4,48	4,89	4,59	5,95	3,83	4,904	0,5375	
Normalized cut	PR index	0,781												
	čas	404,15												

Tabulka A.12: Man with hat

Sada	MAN_WITH_HAT													
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)	
Random walker	PR index	0,791	0,797	0,782	0,724	0,783	0,752	0,69	0,767	0,741	0,699	0,7526	0,0015	
	čas	1,47	1,53	1,33	1,54	1,55	1,57	1,73	1,73	1,75	1,79	1,599	0,0216	
SPRW	PR index	0,882	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,8784	1,6e-6	
	čas	3,07	3,05	3,1	3,11	3,11	3,1	3,06	3,17	3,09	3,17	3,103	0,0017	
GrabCut	PR index	0,721	0,722	0,719	0,719	0,726	0,726	0,611	0,719	0,61	0,717	0,699	0,0022	
	čas	1,61	2,11	2,7	3,04	2,4	2,25	4,92	2,87	2,61	3,35	2,786	0,8065	
Normalized cut	PR index	0,837												
	čas	3,48												

Tabulka A.13: Mushroom

Sada	MUSHROOM													
Značkování		1	2	3	4	5	6	7	8	9	10	E(x)	var(x)	
Random walker	PR index	0,436	0,396	0,408	0,51	0,416	0,453	0,464	0,414	0,392	0,487	0,4376	0,0016	
	čas	1,35	1,3	1,34	1,34	1,37	1,32	1,27	1,35	1,36	1,32	1,332	9,0667e-4	
SPRW	PR index	0,394	0,394	0,399	0,399	0,394	0,394	0,399	0,404	0,394	0,394	0,3965	1,25e-5	
	čas	3,04	3,07	3,02	3,04	3,02	3,04	3,16	3,04	3,03	3,06	3,052	0,0017	
GrabCut	PR index	0,403	0,41	0,421	0,421	0,41	0,403	0,421	0,424	0,397	0,403	0,4113	9,5344e-5	
	čas	1,7	1,71	1,8	1,76	1,8	1,74	1,72	1,81	1,78	1,69	1,751	0,002	
Normalized cut	PR index	0,726												
	čas	53,72												

Tabulka A.14: Ostrich

Sada	OSTRICH											E(x)	var(x)	
Značkování		1	2	3	4	5	6	7	8	9	10			
Random walker	PR index	0,704	0,598	0,556	0,592	0,677	0,619	0,774	0,629	0,51	0,742	0,6401	0,0069	
	čas	1,51	1,6	1,35	1,51	1,4	1,59	1,49	1,48	1,47	1,55	1,495	0,0061	
SPRW	PR index	0,706	0,683	0,628	0,753	0,756	0,74	0,74	0,738	0,655	0,729	0,7128	0,0019	
	čas	3,58	3,62	3,58	3,65	3,59	3,6	3,65	3,63	3,59	3,55	3,604	0,0011	
GrabCut	PR index	0,753	0,668	0,731	0,701	0,774	0,64	0,754	0,796	0,816	0,702	0,7335	0,0031	
	čas	4,76	4,48	5,75	4,48	6,14	4,16	5,55	5,04	14,45	5,58	6,039	9,146	
Normalized cut	PR index	0,594												
	čas	236,15												

Tabulka A.15: Panter

Sada	PANTER											E(x)	var(x)	
Značkování		1	2	3	4	5	6	7	8	9	10			
Random walker	PR index	0,757	0,714	0,696	0,671	0,646	0,716	0,723	0,777	0,735	0,759	0,7194	0,0017	
	čas	1,4	1,66	1,75	1,79	1,76	1,53	1,83	1,62	1,69	1,74	1,677	0,0172	
SPRW	PR index	0,775	0,767	0,753	0,8	0,627	0,834	0,813	0,828	0,826	0,821	0,7844	0,0038	
	čas	4,68	4,64	4,7	4,65	4,66	4,67	4,66	4,64	4,65	4,66	4,661	3,4333e-4	
GrabCut	PR index	0,796	0,786	0,823	0,795	0,784	0,785	0,79	0,823	0,83	0,825	0,8037	3,6268e-4	
	čas	4,08	3,63	3,9	3,79	3,57	3,53	3,22	4,48	3,85	4,27	3,832	0,1395	
Normalized cut	PR index	0,423												
	čas	1615,67												

Tabulka A.16: Penguin

Sada	PENGUIN											E(x)	var(x)	
Značkování		1	2	3	4	5	6	7	8	9	10			
Random walker	PR index	0,805	0,839	0,843	0,818	0,873	0,875	0,862	0,863	0,869	0,896	0,8543	7,7534e-4	
	čas	1,61	1,33	1,41	1,44	1,4	1,36	1,26	1,54	1,68	1,45	1,448	0,0166	
SPRW	PR index	0,838	0,851	0,869	0,862	0,861	0,868	0,87	0,872	0,843	0,869	0,8603	1,4757e-4	
	čas	3,48	3,49	3,63	3,49	3,54	3,47	3,47	3,51	3,56	3,47	3,511	0,0027	
GrabCut	PR index	0,742	0,612	0,758	0,592	0,587	0,579	0,724	0,775	0,769	0,692	0,683	0,0067	
	čas	4,38	4,23	3,68	4,81	4,69	5,06	2,84	3,47	3,32	3,18	3,966	0,5895	
Normalized cut	PR index	0,678												
	čas	18,34												

Tabulka A.17: Soldier

Sada	SOLDIER											E(x)	var(x)
Značkování		1	2	3	4	5	6	7	8	9	10		
Random walker	PR index	0,86	0,712	0,799	0,813	0,768	0,807	0,814	0,794	0,821	0,835	0,8023	0,0016
	čas	1,37	1,52	1,53	1,54	1,59	1,76	1,59	1,6	1,68	1,63	1,581	0,0108
SPRW	PR index	0,867	0,763	0,787	0,839	0,808	0,779	0,853	0,834	0,812	0,8	0,8142	0,0011
	čas	4,84	4,85	4,83	4,86	4,84	4,8	4,83	4,81	4,79	4,82	4,827	4,9e-4
GrabCut	PR index	0,619	0,617	0,576	0,595	0,581	0,577	0,561	0,571	0,634	0,542	0,5873	8,2334e-4
	čas	3,34	3,77	5,94	3,98	5,44	4,6	5,58	4,04	5,74	5,1	4,753	0,8613
Normalized cut	PR index	0,783											
	čas	3455,13											

Tabulka A.18: Totems

Sada	TOTEMS											E(x)	var(x)
Značkování		1	2	3	4	5	6	7	8	9	10		
Random walker	PR index	0,512	0,34	0,487	0,426	0,495	0,518	0,447	0,511	0,522	0,413	0,4671	0,0035
	čas	1,4	1,31	1,38	1,43	1,36	1,43	1,3	1,2	1,35	1,29	1,345	0,0051
SPRW	PR index	0,543	0,444	0,483	0,491	0,563	0,474	0,511	0,506	0,553	0,478	0,5046	0,0015
	čas	6,73	6,65	6,67	6,55	6,69	6,58	6,52	6,57	6,52	6,5	6,598	0,0066
GrabCut	PR index	0,544	0,522	0,581	0,561	0,579	0,539	0,59	0,58	0,581	0,564	0,5641	5,0366e-4
	čas	3,62	2,29	2,92	2,79	2,86	3,51	2,79	2,52	3,25	3,34	2,989	0,1856
Normalized cut	PR index	NE											
	čas	NE											

Tabulka A.19: Two people

Sada	TWO_PEOPLE											E(x)	var(x)
Značkování		1	2	3	4	5	6	7	8	9	10		
Random walker	PR index	0,724	0,728	0,72	0,719	0,713	0,706	0,719	0,708	0,714	0,715	0,7166	4,6267e-5
	čas	1,22	1,39	1,5	1,64	1,54	1,5	1,53	1,74	1,7	1,49	1,525	0,0227
SPRW	PR index	0,727	0,73	0,722	0,736	0,729	0,727	0,729	0,718	0,711	0,729	0,7258	4,9956e-5
	čas	4,79	4,73	4,79	4,78	4,8	4,88	4,82	4,94	4,85	4,86	4,824	0,0036
GrabCut	PR index	0,609	0,608	0,605	0,605	0,604	0,598	0,605	0,587	0,591	0,608	0,602	5,7111e-5
	čas	2,7	3,72	3,37	3,61	3,55	3,65	3,81	4,65	3,91	3,47	3,644	0,2358
Normalized cut	PR index	NE											
	čas	NE											

Tabulka A.20: Vase

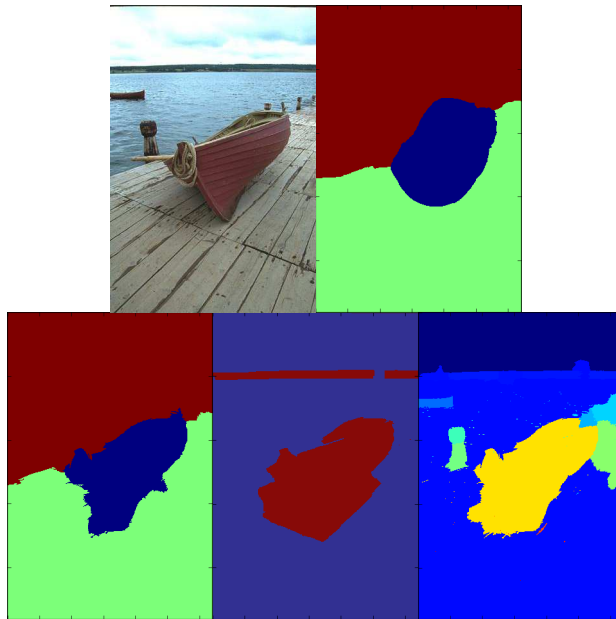
Sada	VASE	1	2	3	4	5	6	7	8	9	10	E(x)	var(x)
Random walker	PR index	0,835	0,811	0,798	0,837	0,752	0,803	0,808	0,843	0,829	0,75	0,8066	0,0011
	čas	1,19	1,34	1,15	1,19	1,3	1,46	1,46	1,38	1,3	1,46	1,323	0,014
SPRW	PR index	0,866	0,872	0,875	0,879	0,876	0,869	0,875	0,861	0,87	0,872	0,8715	2,7833e-5
	čas	4,06	4,23	4,09	4,09	4,12	4,07	4,19	4,1	4,04	4,17	4,116	0,0038
GrabCut	PR index	0,728	0,73	0,728	0,73	0,73	0,73	0,73	0,73	0,73	0,73	0,7296	7,1111e-7
	čas	3,39	5,11	3,75	3,45	4,69	5,26	4,96	4,58	4,23	4,8	4,422	0,467
Normalized cut	PR index	0,552											
	čas	20,1											

## Příloha B

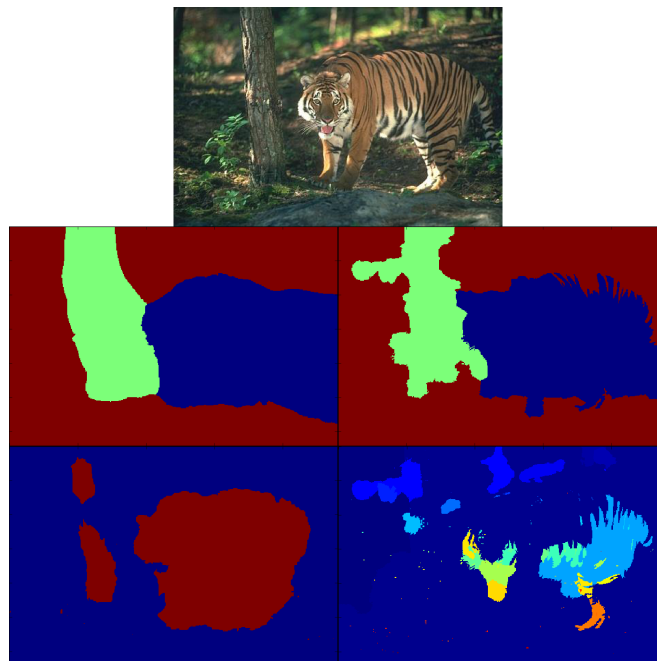
### Obrázky srovnávací test



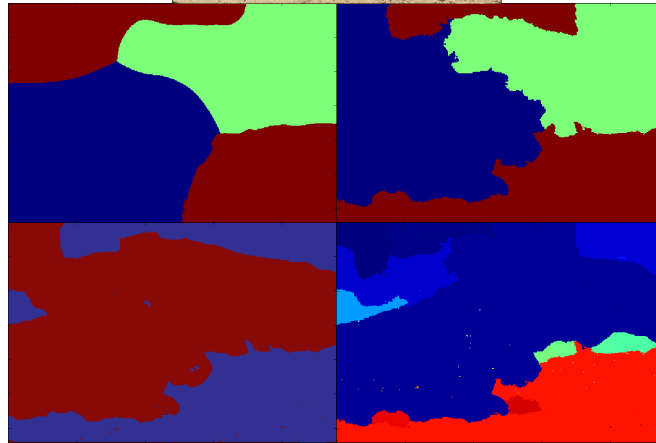
Obrázek B.1: Bird. RW: 0.573, SPRW: 0.501, GC: 0.546, NC: 0.959



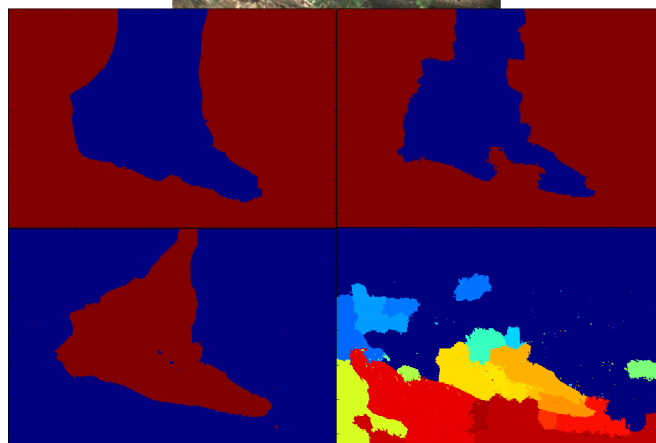
Obrázek B.2: Boat. RW: 0.845, SPRW: 0.861, GC: 0.514, NC: 0.778



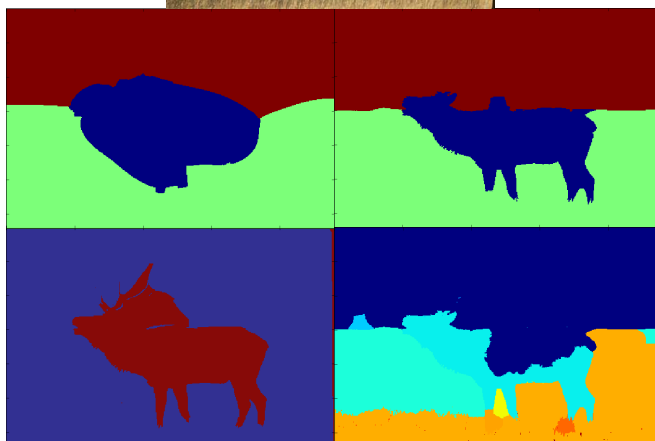
Obrázek B.3: Cat. RW: 0.706, SPRW: 0.700, GC: 0.619, NC: 0.483



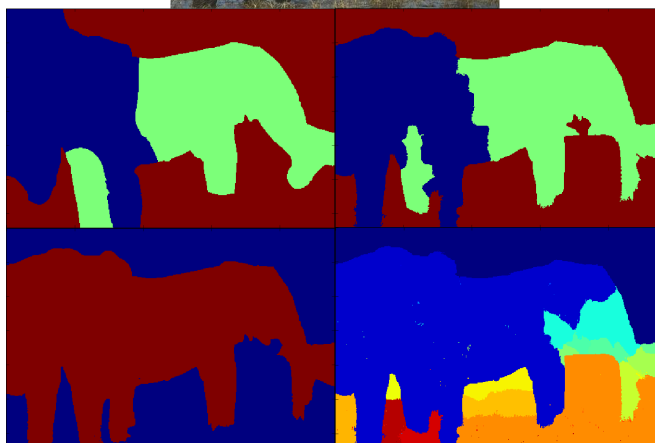
Obrázek B.4: Cow. RW: 0.800, SPRW: 0.838, GC: 0.653, NC: 0.808



Obrázek B.5: Croc. RW: 0.782, SPRW: 0.771, GC: 0.837, NC: 0.565

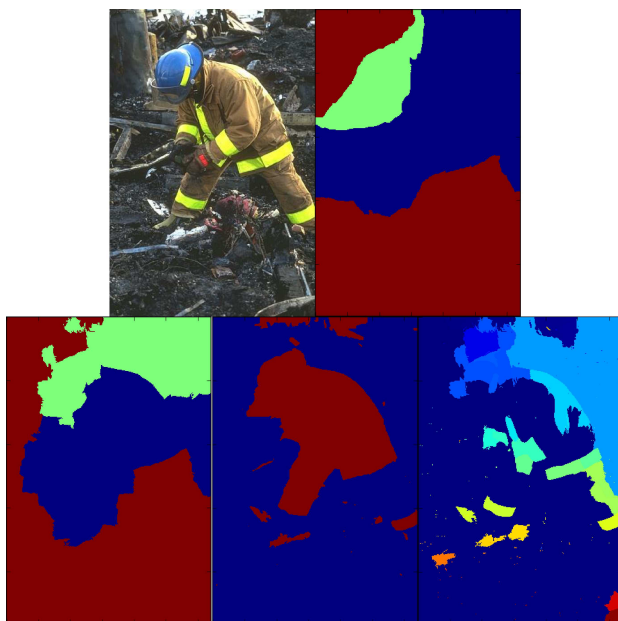


Obrázek B.6: Deer. RW: 0.868, SPRW: 0.929, GC: 0.605, NC: 0.838

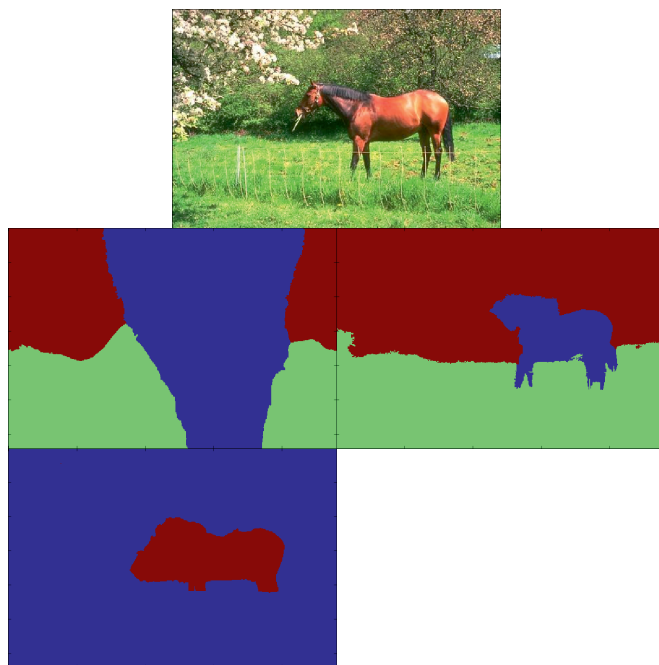


Obrázek B.7: Elephant. RW: 0.772, SPRW: 0.799, GC: 0.666, NC: 0.824

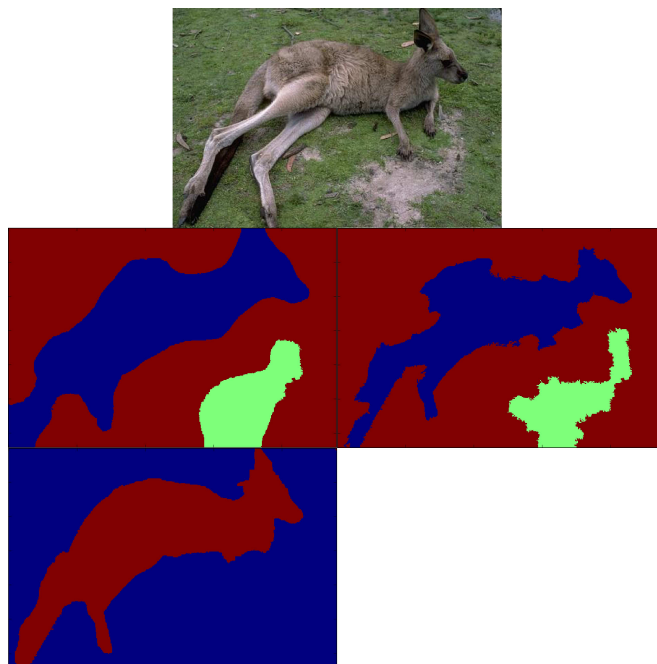




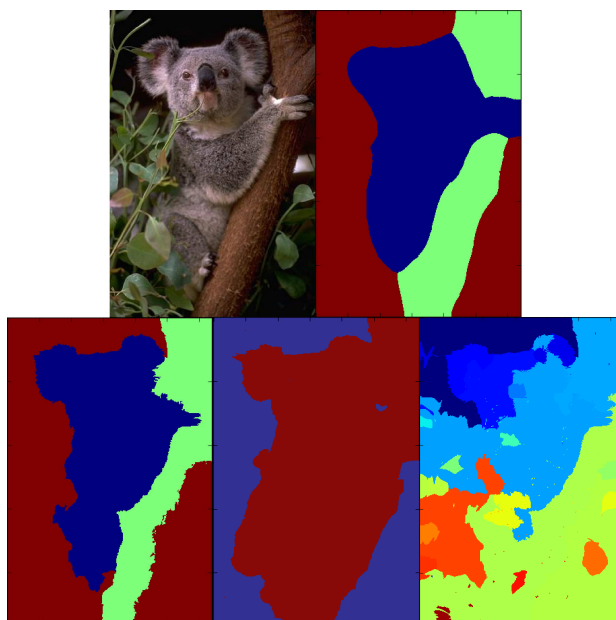
Obrázek B.8: Fireman. RW: 0.644, SPRW: 0.677, GC: 0.608, NC: 0.648



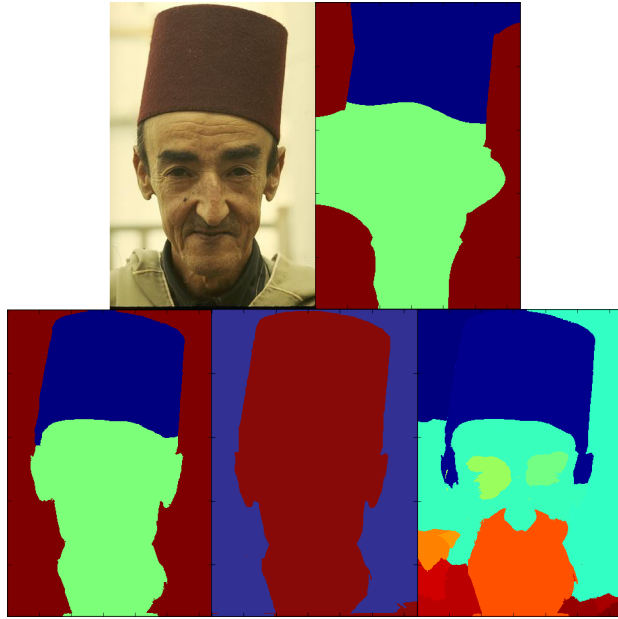
Obrázek B.9: Horse. RW: 0.628, SPRW: 0.791, GC: 0.563



Obrázek B.10: Kangaroo. RW: 0.641, SPRW: 0.612, GC: 0.565



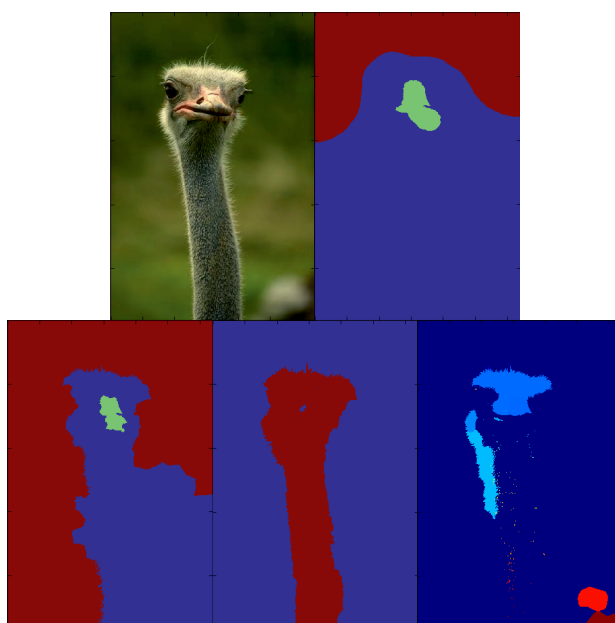
Obrázek B.11: Koala. RW: 0.781, SPRW: 0.776, GC: 0.619, NC: 0.781



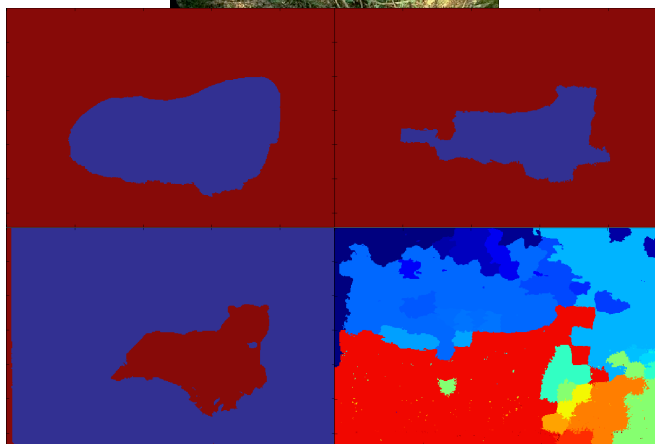
Obrázek B.12: Man with hat. RW: 0.791, SPRW: 0.882, GC: 0.721, NC: 0.837



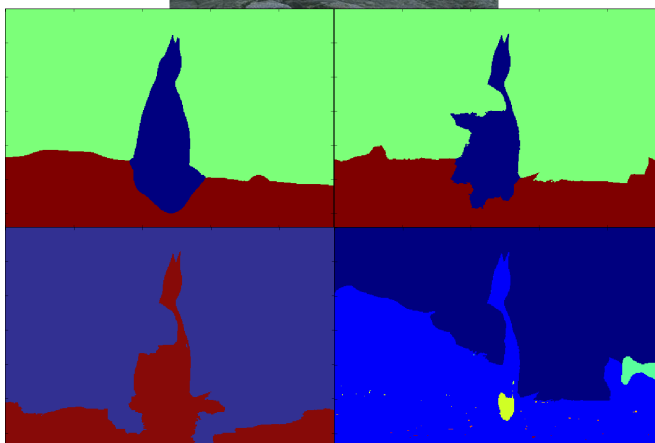
Obrázek B.13: Mushroom. RW: 0.510, SPRW: 0.399, GC: 0.421, NC: 0.726



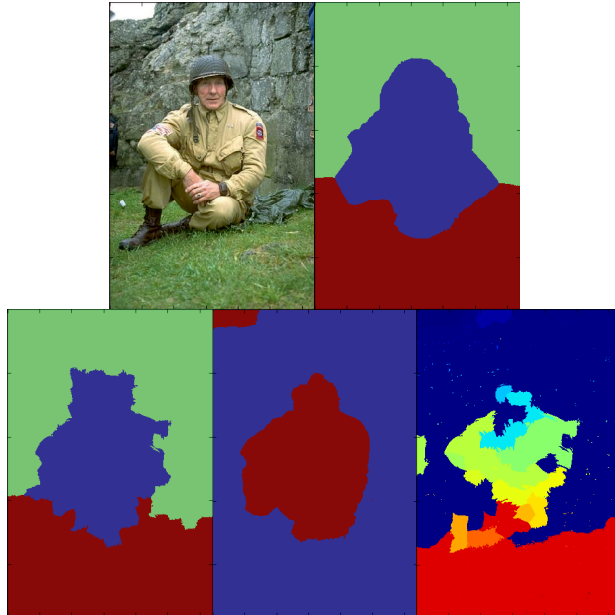
Obrázek B.14: Ostrich. RW: 0.510, SPRW: 0.655, GC: 0.816, NC: 0.594



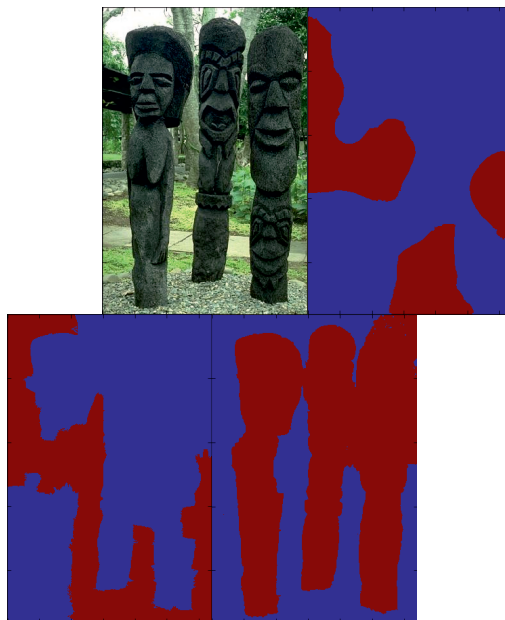
Obrázek B.15: Panter. RW: 0.716, SPRW: 0.834, GC: 0.785, NC: 0.423



Obrázek B.16: Penguin. RW: 0.896, SPRW: 0.869, GC: 0.692, NC: 0.678



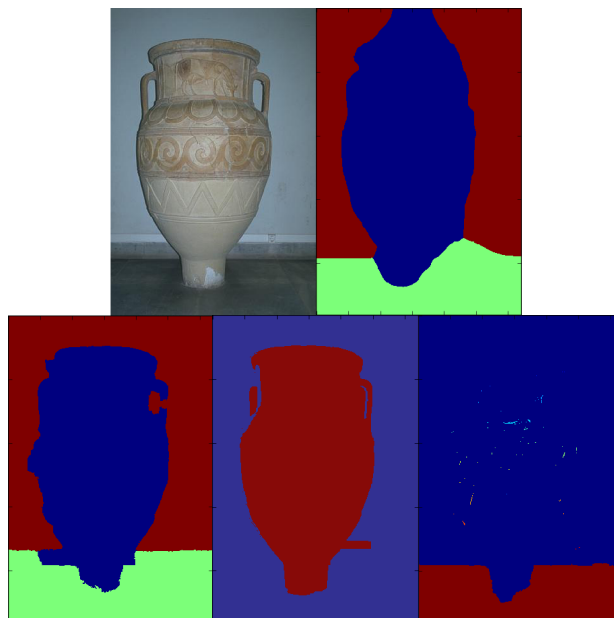
Obrázek B.17: Soldier. RW: 0.860, SPRW: 0.867, GC: 0.619, NC: 0.783



Obrázek B.18: Totems. RW: 0.447, SPRW: 0.511, GC: 0.590



Obrázek B.19: Two people. RW: 0.719, SPRW: 0.736, GC: 0.605



Obrázek B.20: Vase. RW: 0.837, SPRW: 0.879, GC: 0.730, NC: 0.552