

Segmentácia obrazu

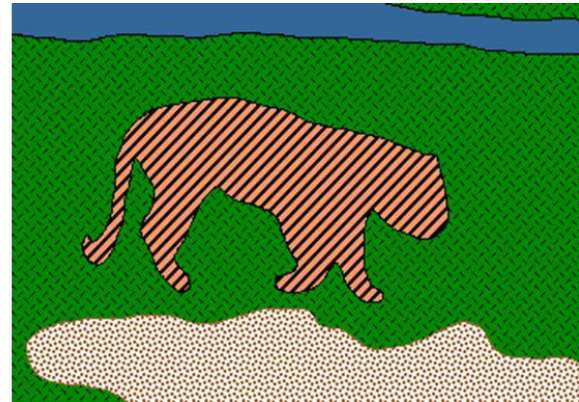
Cvičenia z Počítačového videnia 1

Segmentácia obrazu

- kľúčový krok k porozumeniu obrazu
- cieľ
 - rozdeliť obraz na časti, ktoré majú silnú koreláciu s objektami alebo oblasťami reálneho sveta zobrazenými v obraze
- segmentácia
 - úplná segmentácia (rozdelenie obrazu na disjunktné oblasti, ktoré zodpovedajú objektom)
 - čiastočná segmentácia (kde čiastočne zodpovedajú)

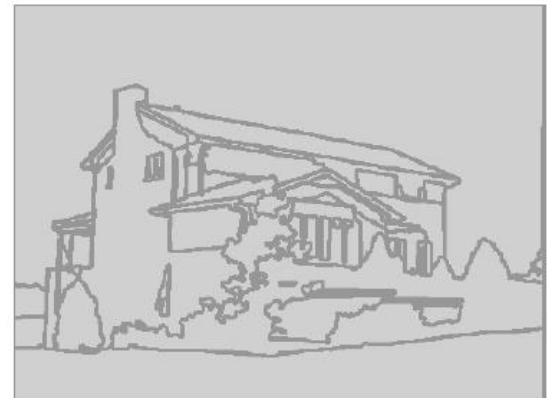
Úplná segmentácia

- čím viac apriórnej informácie je k dispozícii pri segmentačnom procese, tým lepšie výsledky pri segmentácii možno dosiahnuť



Čiastočná segmentácia

- cieľ
 - rozdeliť obraz na časti, ktoré sú homogénne z hľadiska vybranej vlastnosti, napr. jas, farby, textúry



Segmentácia obrazu - kritériá

- oblasti obrazu po segmentácii
 - pokrývajú celý obraz
 - neprekrývajú sa
 - sú homogénne
- nezlučovať príľahlé oblasti

Segmentačné metódy

- segmentácia založená na pixeloch (podobnosť)
 - histogram
 - zhlukovanie
- segmentácia založená na hranách (diskontinuita)
- segmentácia založená na oblastiach (podobnosť)
 - split and merge
 - region growing
- ...

segmentácia založená na pixeloch

- prahovanie
 - najjednoduchšia segmentačná technika
 - výpočtovo nenáročná a rýchla
 - globálne
 - jeden prah aplikovaný na celý obraz
 - lokálne
 - prah môže meniť v jednotlivých častiach obrazu

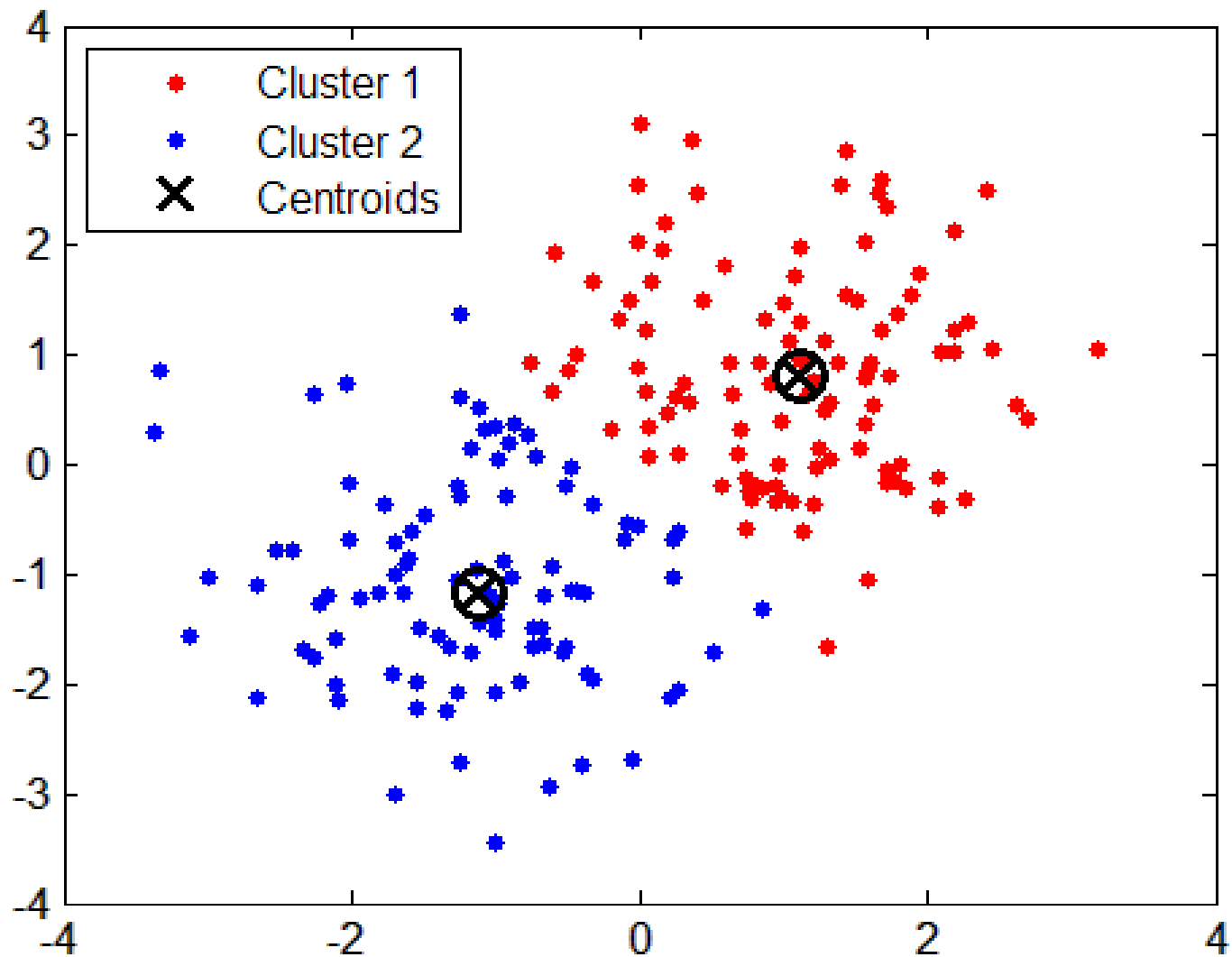
Segmentácia založená na zhľukovaní

- K-means
 - rozdelí obraz na **k** segmentov
 - maximalizuje podobnosť objektov v rámci segmentu a minimalizuje podobnosť medzi segmentmi navzájom
 - nevýhoda : náhodný výber počiatočných stredov

K-means

- algoritmus náhodne zvolí **k** objektov - každý reprezentuje mean – stred
- ostatné objekty sú priradené ku segmentom na základe podobnosti určenej cez vzdialenosti medzi objektmi a centrami segmentov
- nanovo vypočítajú ich nové centrá na základe rozdelenia objektov do segmentov

K-means



K-means – jednoduchý příklad

```
img_rgb = imread('peppers.png');  
img_gray = im2double ( rgb2gray(img_rgb));  
[ny,nx] = size(img_gray);  
d_gray = reshape(img_gray, ny*nx, 1);  
[L C] = kmeans( d_gray, 2);  
L = reshape( L, ny, nx);  
imshow(L/max(max(L)));
```

L – výsledné rozdelenie do skupín na základe rozsegmentovania

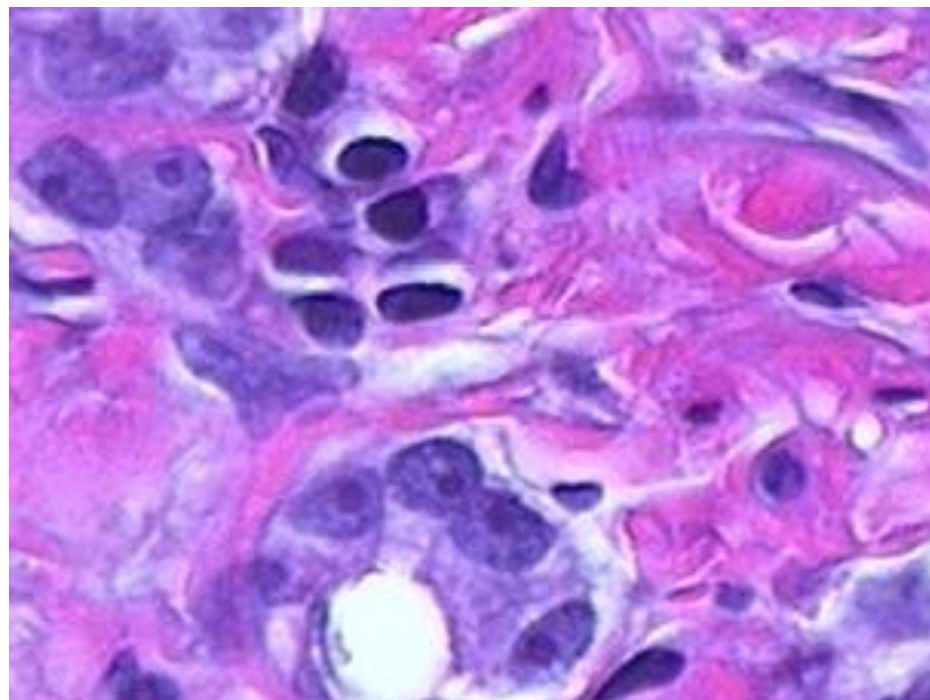
C - centroidy

K-means - príklad

<http://www.mathworks.com/products/image/examples.html?file=/products/demos/shipping/images/ipexhistology.html>

Vstup: obrázok tkaniva

Úloha: rozdelenie obrázka
na časti s rôznym typom
tkaniva a vysegmentovanie
tmavomodrých buniek



K-means - príklad

Načítanie obrázka

```
he = imread('hestain.png');  
figure, imshow(he), title('H&E image');
```

Konvertovanie obrázka do L*a*b* farebného priestoru kvôli lepšiemu rozlíšeniu dominantných farieb (biela, modrá, ružová)

```
cform = makecform('srgb2lab');  
lab_he = applycform(he,cform);  
figure, imshow(lab_he), title('H&E L*a*b* image');
```

K-means - príklad

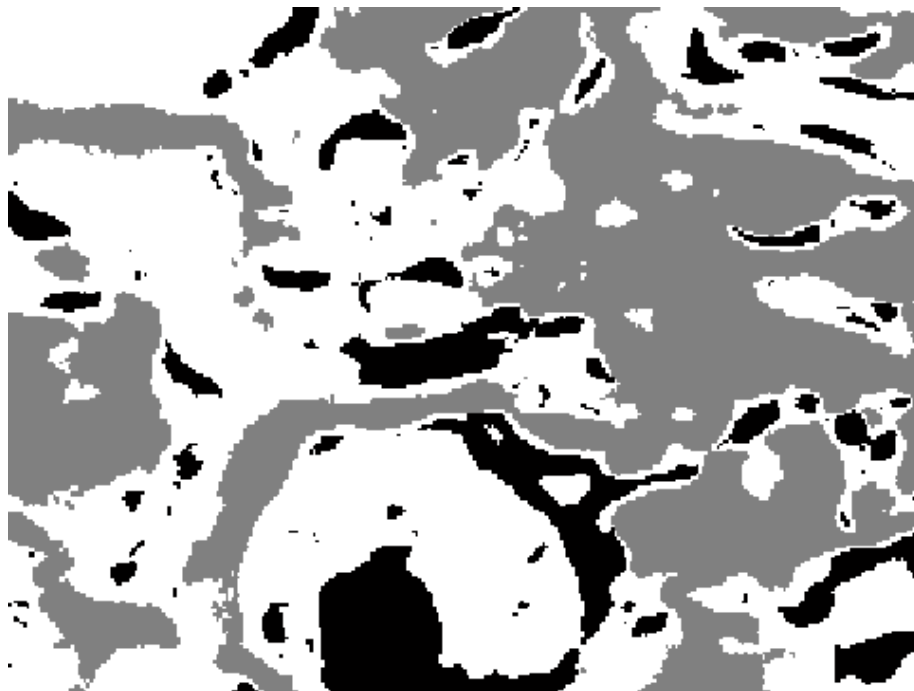
Klasifikácia farieb L*a*b modelu pomocou zhľukovania K-means

```
ab = double(lab_he(:,:,2:3));  
nrows = size(ab,1);  
ncols = size(ab,2);  
ab = reshape(ab,nrows*ncols,2);  
nColors = 3;  
% repeat the clustering 3 times to avoid local minima  
[cluster_idx cluster_center] =  
kmeans(ab,nColors,'distance','sqEuclidean', ...  
        'Replicates',3);
```

K-means - príklad

Rozdelenie pixlov obrázka podľa výsledkov K-means

```
pixel_labels = reshape(cluster_idx,nrows,ncols);  
figure, imshow(pixel_labels,[]), title('image labeled by  
cluster index');
```



K-means - príklad

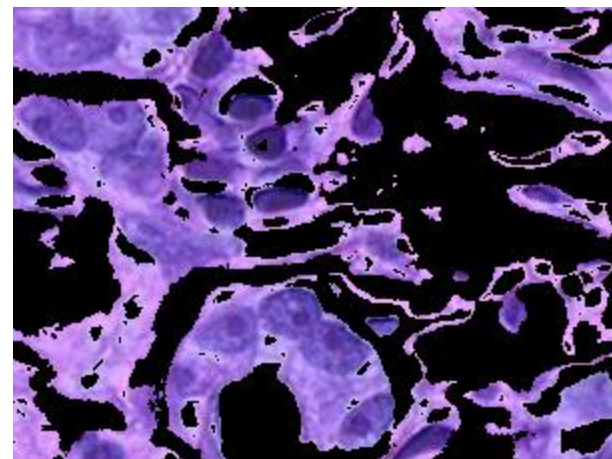
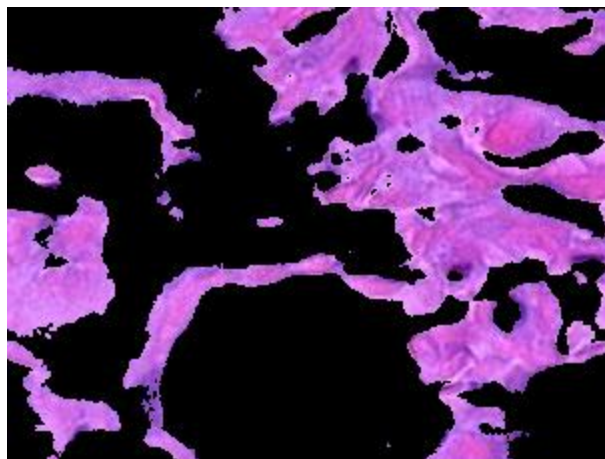
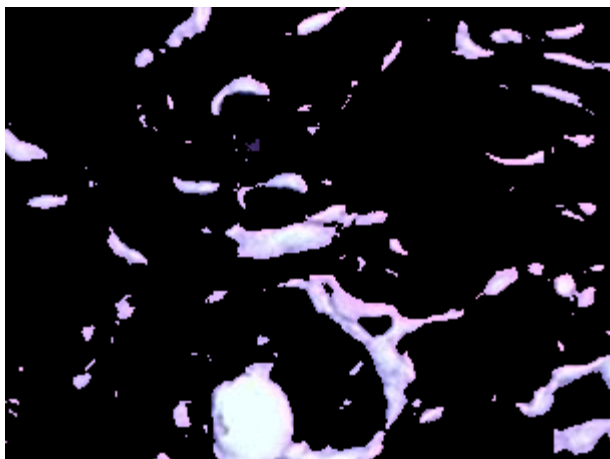
Zobrazenie častí obrazu podľa rozsegmentovania

```
segmented_images = cell(1,3);  
rgb_label = repmat(pixel_labels,[1 1 3]);  
  
for k = 1:nColors  
    color = he;  
    color(rgb_label ~= k) = 0;  
    segmented_images{k} = color;  
end
```

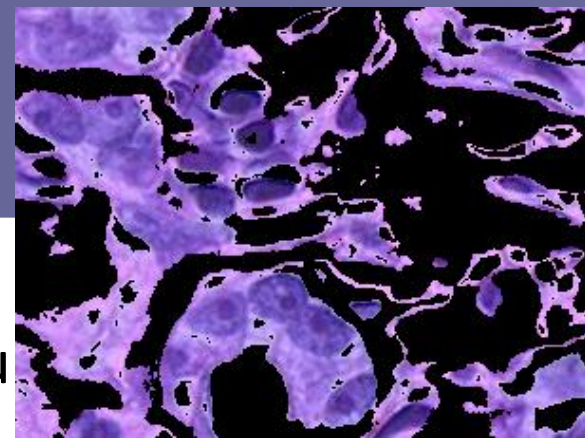

K-means - príklad

Zobrazenie častí obrázu podľa rozsegmentovania

```
figure, imshow(segmented_images{1}), title('objects in cluster 1');  
figure, imshow(segmented_images{2}), title('objects in cluster 2');  
figure, imshow(segmented_images{3}), title('objects in cluster 3');
```



K-means - príklad



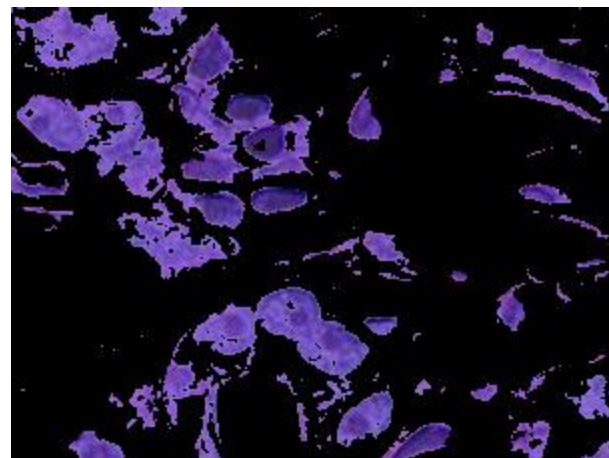
Vysegmentovanie modrých buniek pomocou
L kanála

```
mean_cluster_value = mean(cluster_center,2);  
[tmp, idx] = sort(mean_cluster_value);  
blue_cluster_num = idx(1); % modrý klaster  
L = lab_he(:,:,1);  
blue_idx = find(pixel_labels == blue_cluster_num);  
L_blue = L(blue_idx);  
is_light_blue = im2bw(L_blue,graythresh(L_blue));
```

K-means - príklad

Použitím masky `is_light_blue` určiť ktoré bunky sú tie, ktoré chceme

```
nuclei_labels = repmat(uint8(0),[nrows ncols]);  
nuclei_labels(blue_idx(is_light_blue==false)) = 1;  
nuclei_labels = repmat(nuclei_labels,[1 1 3]);  
blue_nuclei = he;  
blue_nuclei(nuclei_labels ~= 1) = 0;  
figure, imshow(blue_nuclei),  
title('blue nuclei');
```



Segmentácia založená na zhľukovaní

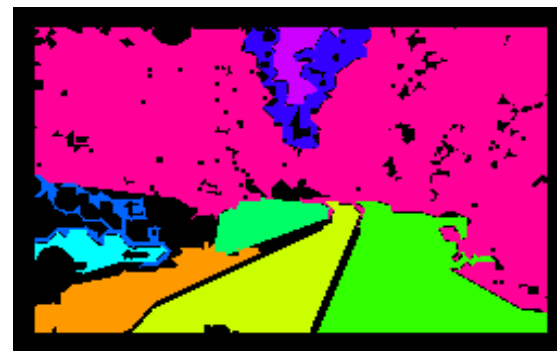
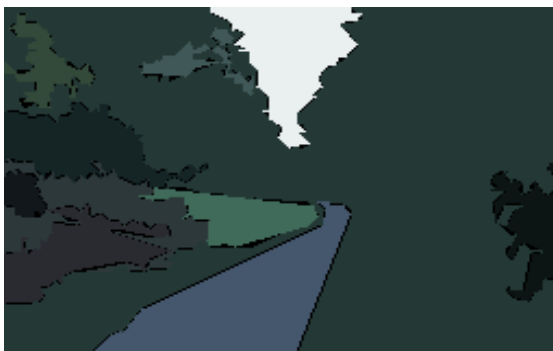
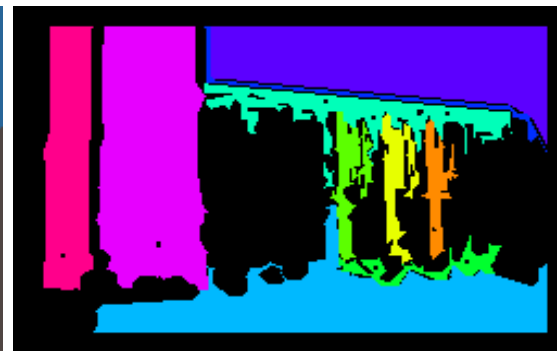
Original Images



Color Regions



Texture Regions



Split and merge

- top-down prístup
- predpokladá, že obraz je homogénny, ak nie, delí ho na 4 časti
- rekurzívna procedúra
- reprezentácia stromom - quadtree

Split, Split and merge

- **Split**
 - nevýhoda: vytvára oblasti, ktoré môžu byť podobné a susedné, ale nespojí ich
- **Split and merge**
 - iteratívny algoritmus, ktorý obsahuje rozdeľovanie aj spájanie
 - ak sú susedné oblasti dostatočne podobné a homogénne, spojí ich
 - vytvára kompaktnejšie oblasti, ako split

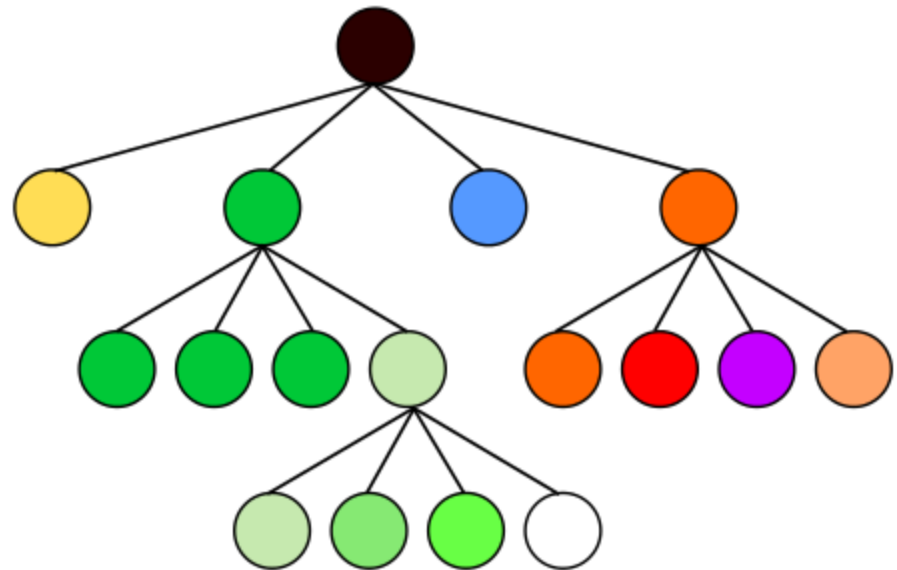
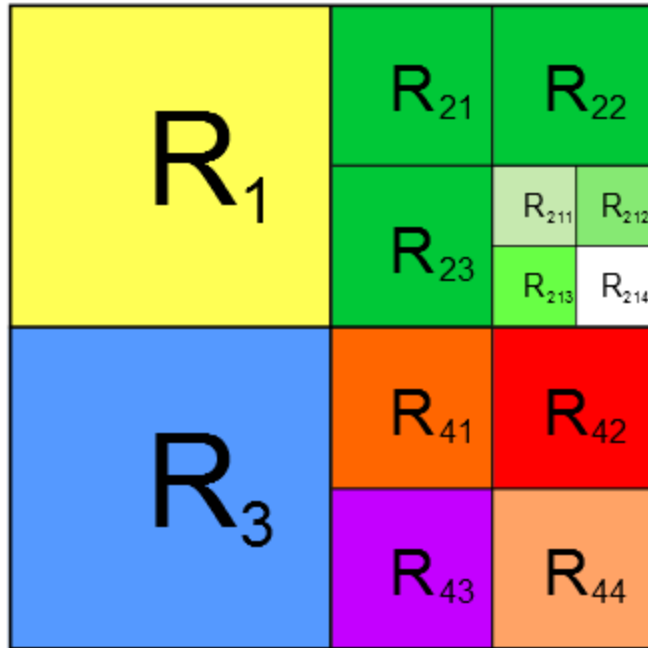
Split and merge

- iteratívny algoritmus, ktorý obsahuje rozdeľovanie aj spájanie
- aksú susedné oblasti dostatočne podobné a homogénne, spojí ich
- vytvára kompaktnejšie oblasti, ako split
- problém: šum
- aplikácie:
 - medicína MRI
 - 3D – Imaging

Split and merge - algoritmus

1. vstup: pôvodný obraz
2. ak je variancia príliš vysoká rozdeľ na kvadranty
3. spoj všetky oblasti, ktoré sú dostatočne podobné
4. opakuj 2 a 3, pokiaľ sa oblasti rozdeľujú/spájajú

Split and merge



Porovnanie

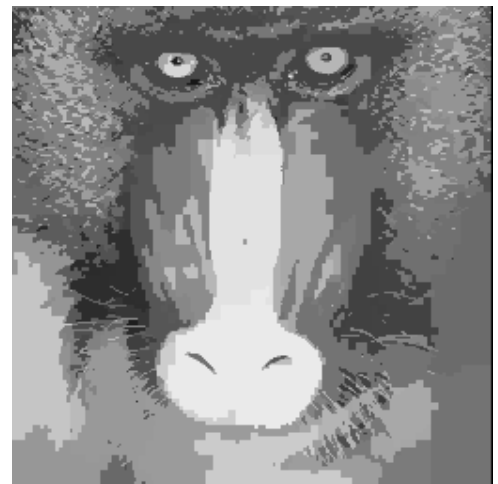
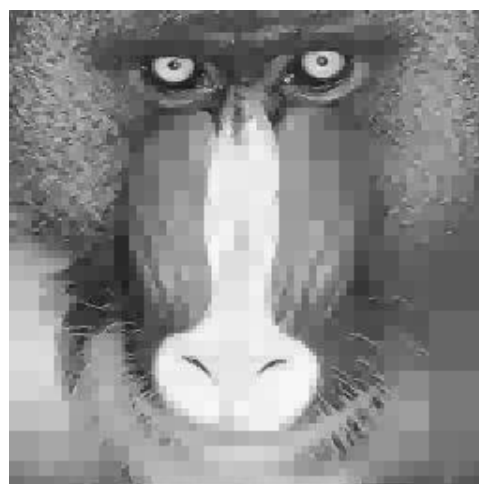
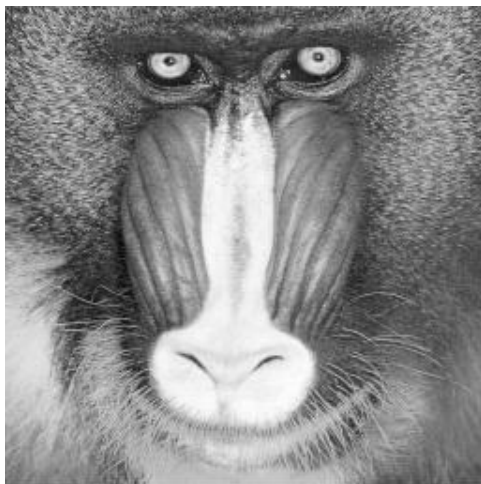
Pôvodný obraz



Split



Split and Merge

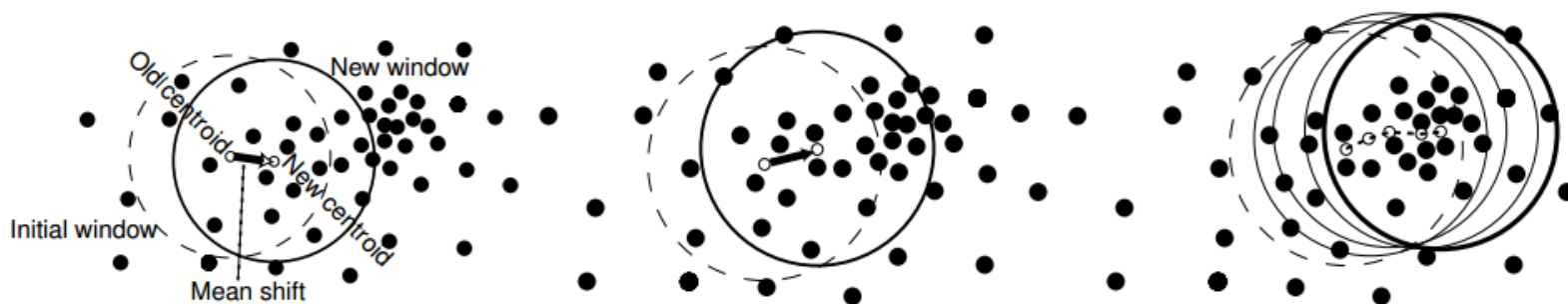


Mean shift

- neparametrická intuitívna metóda
- zhlučuje body obrazu (pixely) na základe podobnosti ich vzhľadu a blízkosti ich pozície pomocou konvergenzie do lokálnych maxím spojeného súradnicového a intenzitného priestoru
- použitie:
 - filtrácia, segmentácia, object tracking

Mean shift

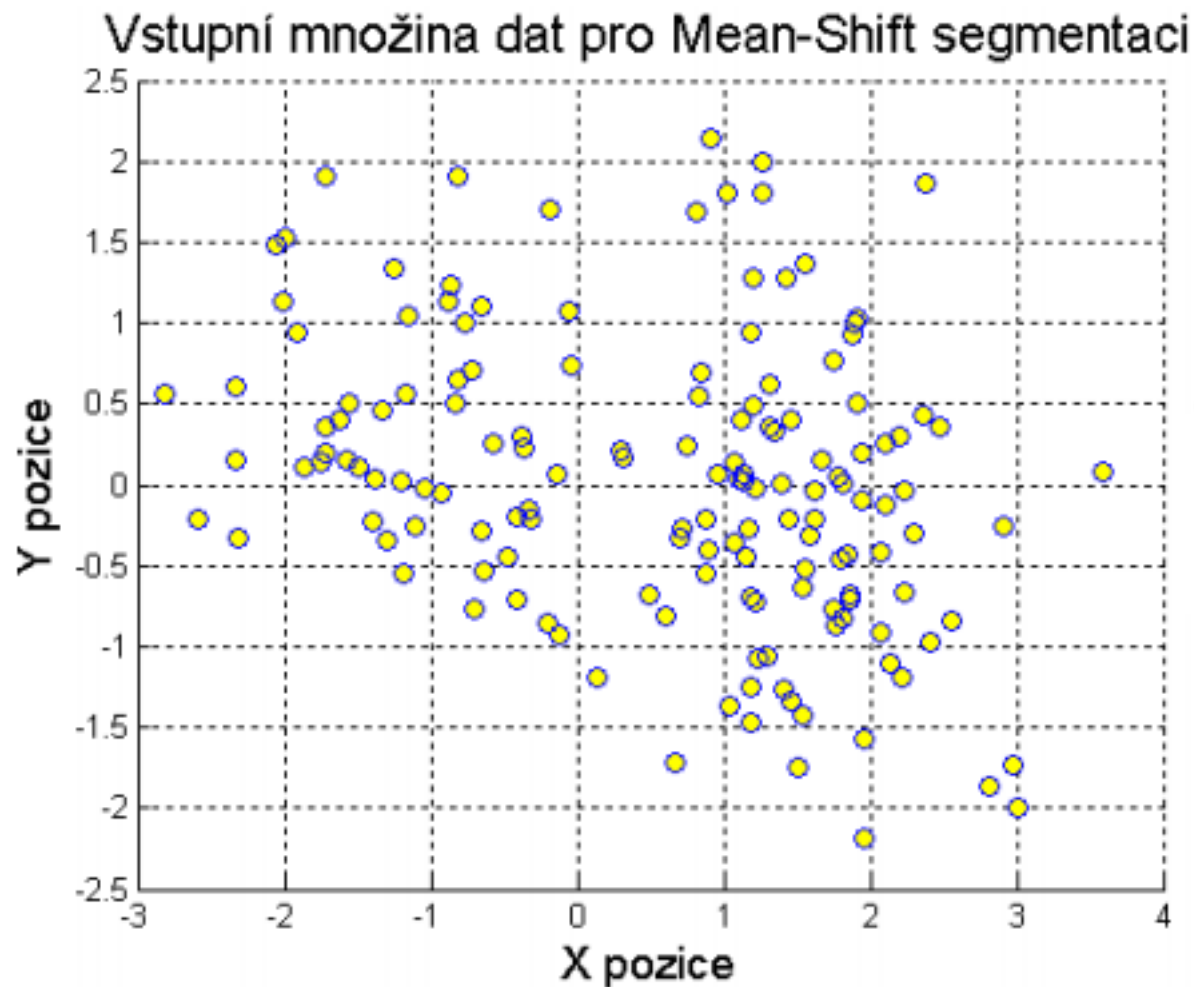
- Postup pri hľadaní lokálneho maxima:
 - začni v oblasti záujmu
 - urči centroid dát
 - posuň sa na pozíciu nového centroidu
 - opakuj



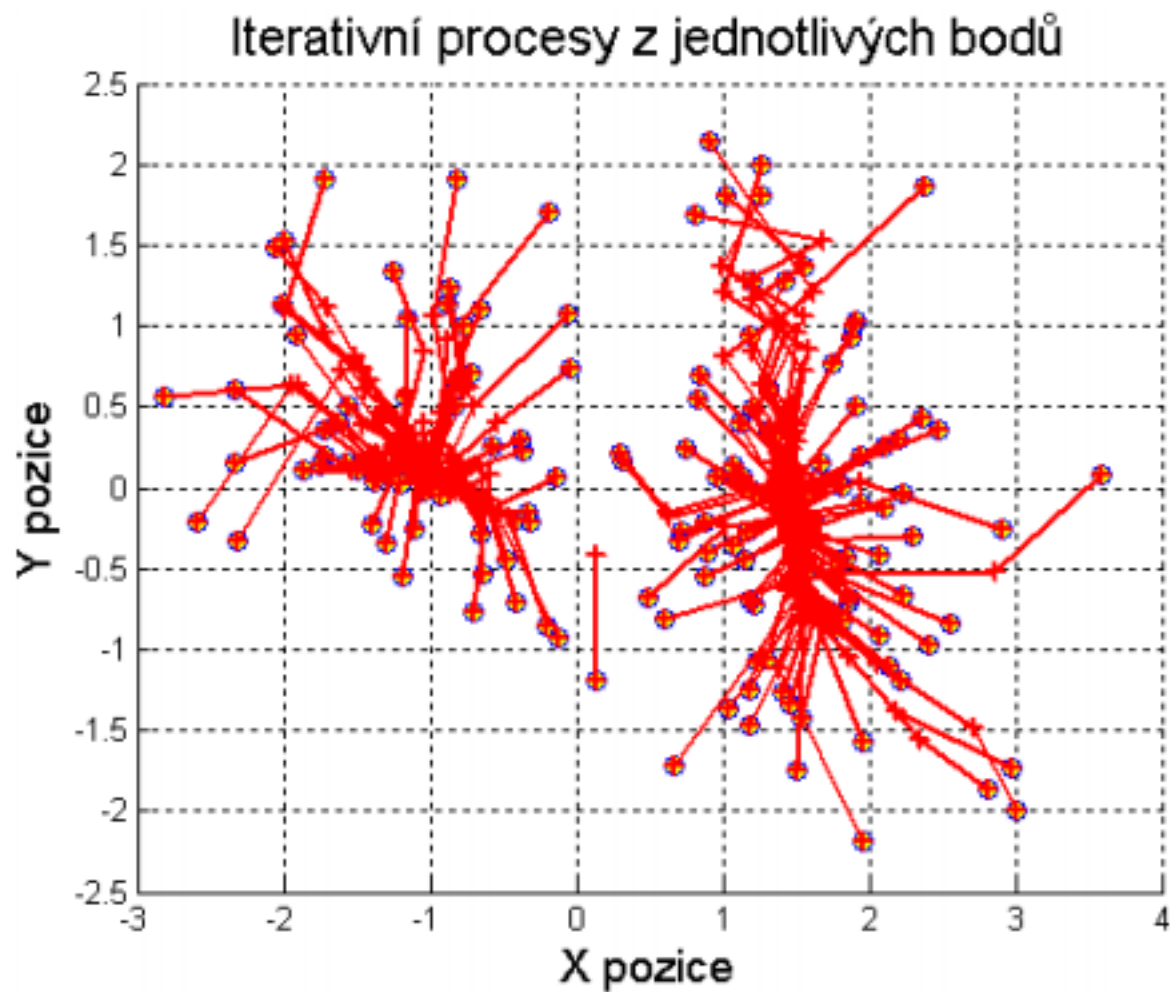
Mean shift

- v každej iterácii pre určitú pozíciu počíta gradient hustoty bodov v blízkom okolí a pohybuje sa v smere gradientu, až kým nedosiahne lokálne maximum
- tento proces je iniciovaný v každom bode
- výsledok
 - súradnice lokálneho maxima pre každý bod
 - zhukovanie obrazových bodov (pixlov), ktoré dokonvergovali do rovnakého maxima s určitou toleranciou

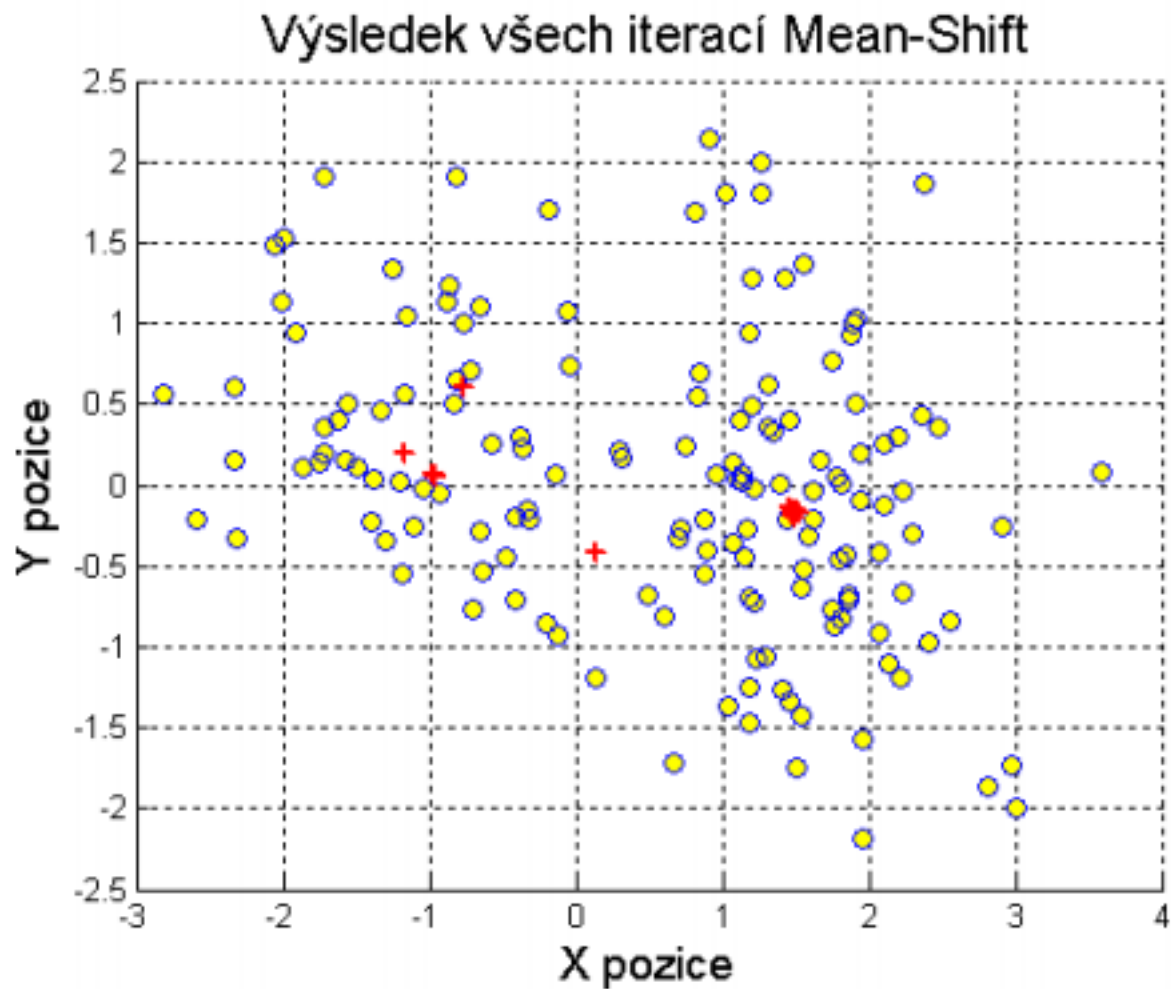
Mean shift



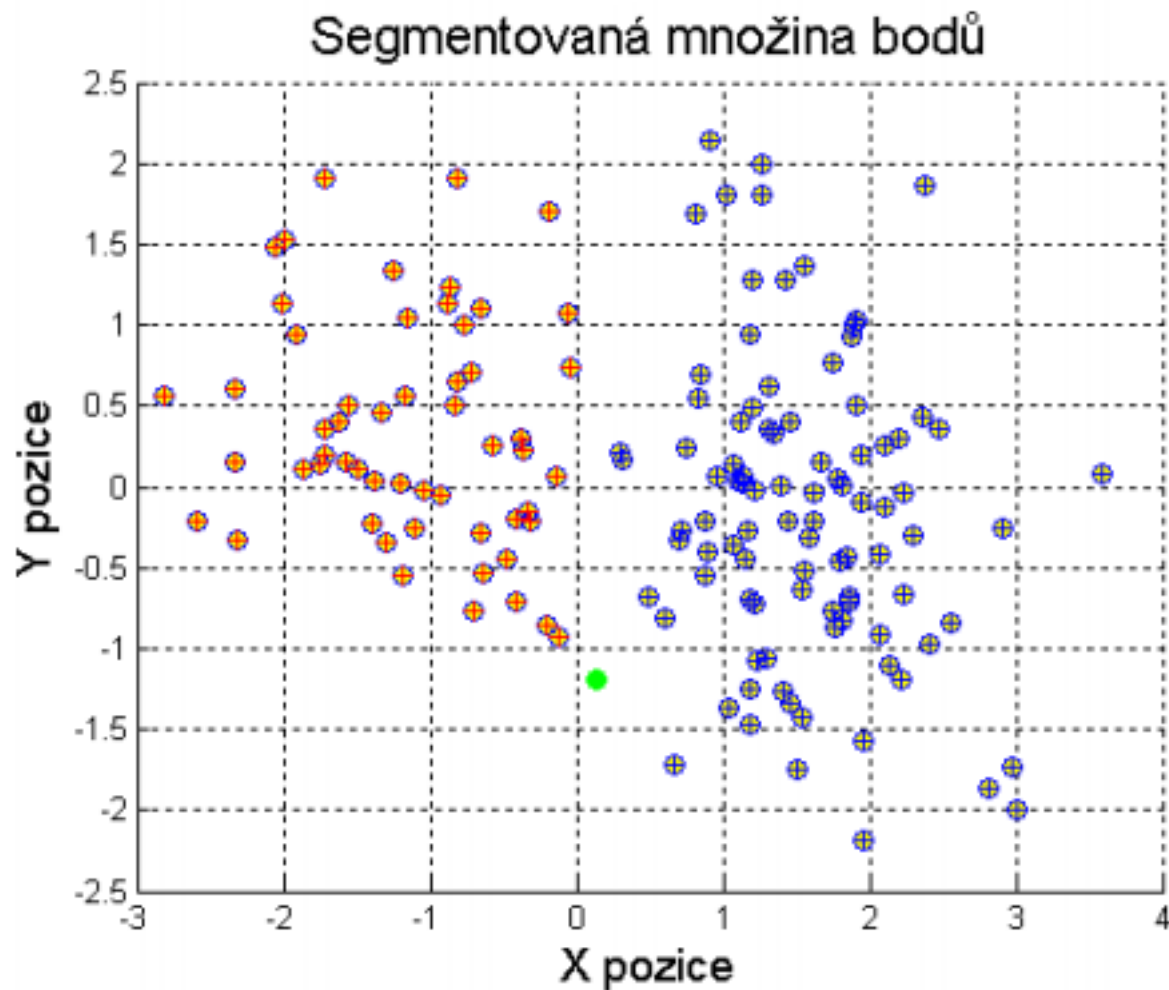
Mean shift



Mean shift

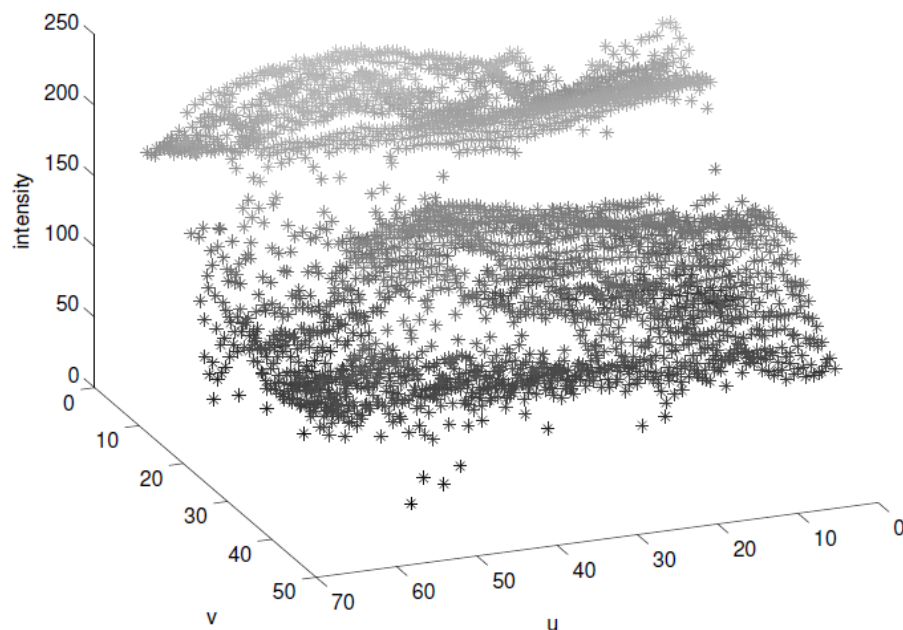
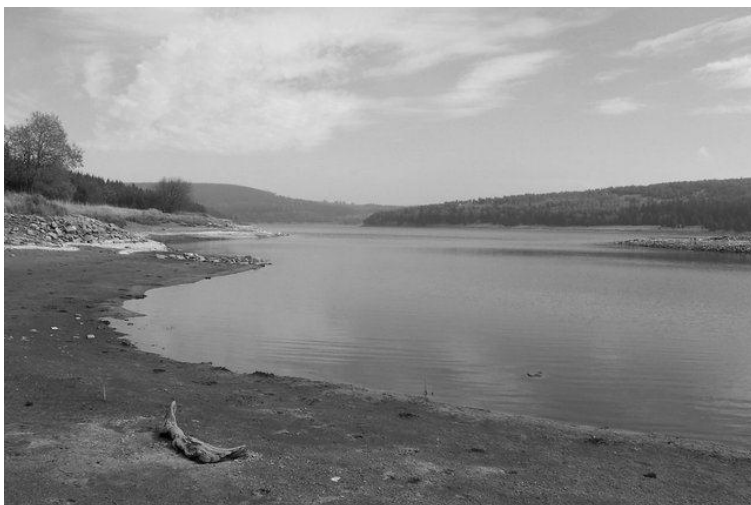


Mean shift



Mean shift - príklad

- vstup:
 - šedotónový obraz
 - každý pixel má súradnice $[u,v]$ a intenzitu $i \Rightarrow$ možná reprezentácia v 3D



Mean shift - príklad

- blízke pixle s podobnou intenzitou budú vytvárať zhluky
- susedné pixle s výrazne rozdielnou intenzitou budú v priestore vzdialené a preto budú patriť do rôznych zhlukov
- => segmentácia prevedená na *zhlukovanie* bodov v d -rozmernom priestore
 - šedotónový obraz: $d=3$ (u,v, intenzita)
 - farebný obraz: $d=5$ (u,v,r,g,b)

Mean shift

- Mean Shift Clustering
 - <http://machinethatsees.blogspot.sk/2012/01/mean-shift-clustering-segmentation-in.html>
- Image segmentation with mean-shift algorithms
 - <http://faculty.ucmerced.edu/mcarreira-perpinan/papers/icml06-applet/ImgSeg-MS-Embedded.html>

Aktívne kontúry

- čiary, body, obdĺžniky... :
 - bežné objekty vo svete vytvorené človekom
- bežný svet, príroda:
 - krivky, ktoré korešpondujú s okrajmi objektov
- aktívne kontúry:
 - prístup na detekovanie kriviek ohraničujúcich objekty

Aktívne kontúry

- snakes
 - 2D spline krivka, ktorá minimalizuje energiu
 - pohybuje sa k príznakom v obraze, ako sú napríklad hrany
- intelligent scissors
 - pomáha užívateľovi real-time načrtnúť krivku, ktorá sa „drží“ hraníc objektu
- level set
 - tvorí krivku, ako nulovú množinu z charakteristickej funkcie, čo umožňuje jednoducho meniť topológiu oblasti

Snakes

- pokročilá metóda segmentácie obrazu
- metóda postupného tvarovania kontúr až ku hrane objektu v obraze
- predpokladá, že obraz je tvorený dvoma oblasťami – objektom a pozadím
- vstup: uzavretá krivka, ktorá približne ohraničuje daný objekt

Snakes

- 2D zovšeobecnenie 1D splajnov, ktoré minimalizujú energiu, z ktorej môžeme vytvoriť diskretnú funkciu na získanie vnútornej energie E_{int}
- okrem vnútornej energie, snake súčasne minimalizuje vonkajšie obrazové a obmedzujúce sily

Snakes

- 2 typy síl
 - vonkajšie
 - tlačia snake do oblastí podľa informácií z obrazu
 - vnútorné
 - snažia sa držať snake-a zaobleného a bez veľkých uhlov
 - kontrolujú hladkosť priebehu

Snakes

- obrazové sily
 - smerujú tvarovanie kontúry smerom k hrane objektu

$$E_{\text{image}} = w_{\text{line}} E_{\text{line}} + w_{\text{edge}} E_{\text{edge}} + w_{\text{term}} E_{\text{term}}$$

- line
 - priťahujú snake k tmavým oblastiam
- edge
 - priťahujú snake k oblastiam so silným gradientom
- term
 - priťahuje snake ku koncom čiar

Aktívne kontúry

- <http://www.shawnlankton.com/2008/04/active-contour-matlab-code-demo/>