

# Pokročilé spracovanie obrazu - Fourierová transformácia

Ing. Viktor Kocur  
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

29.11.2017

# Obsah

- 1 Segmentácia
  - O čo ide
- 2 Watershed
  - Princíp
  - Postup
- 3 k-means clustering
  - Algoritmus
  - k-means v Matlabe
- 4 Graph Cut
  - Graph Cut

# Cieľ segmentácie

## Cieľ segmentácie

Cieľom segmentácie je rozdeliť obrázok na disjunktné oblasti, tak že jednotlivé oblasti zodpovedajú samostatným objektom/útvaram.

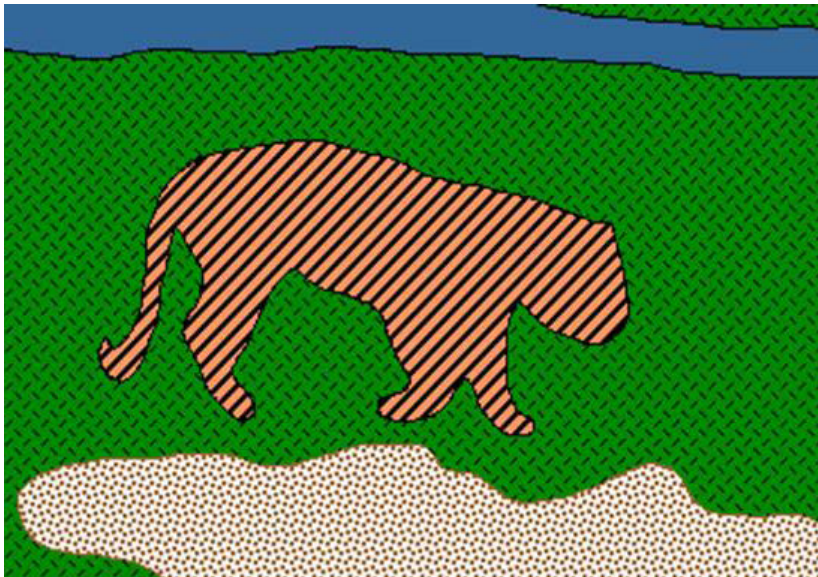
## Čiastočná segmentácia

Nie vždy je možné rozdeliť obraz na jednotlivé objekty/útvary. Niekedy nám však stačí čiastočná segmentácia. V takom prípade môže byť segmentovaný, len nejaký objekt, alebo skupina objektov.

# Pôvodný obrázok



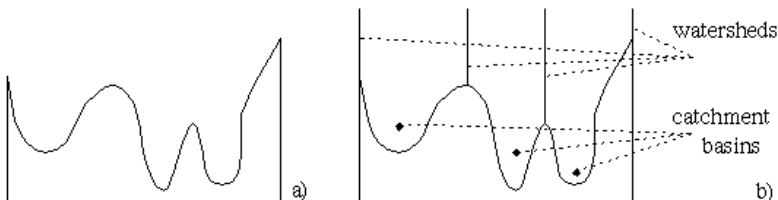
# Segmentovaný



# Princíp

## Watershed

Watershed vníma obraz ako topografickú mapu a oblasti rozdelí, podľa toho ako by boli rozdelené povodia na mape.



## Vstup pre watershed

Watershed je fajn, ale najprv potrebujeme vhodný obraz na získanie povodí.

# Princíp

## imgradient

imgradient(I) - vráti gradient šedotónového obrázka

## watershed

watershed(I) - vráti label maticu po aplikácii watershed algoritmu na šedotónový obraz

## label2rgb

label2rgb(L) - z label matice vytvorí obrázok, kde sú farby určené podľa labelov

## Úloha

Načítajte si obrázok pears.png, zobrazte si jeho gradient a watershed gradientu

# Postup

## Postup rozdelíme na časti

- Otvorenie a zatvorenie pomocou imread a imshow
- Nájdenie masky popredia
- Nájdenie hrebeňov v obraze
- Vynútenie miním na maske
- Watershed

## Link

Tento postup je aj na stránke:

<https://www.mathworks.com/help/images/marker-controlled-watershed-segmentation.html>



# Otvorenie a zatvorenie

## Kód

```
se = strel('disk',20);  
Ioc = imclose(imopen(I,se),se);  
Ie = imerode(I,se);  
Iobr = imreconstruct(Ie,I);  
Iobrd = imdilate(Iobr,se);  
Iobrcbr = imreconstruct(imcomplement(Iobrd), ...  
    imcomplement(Iobr));  
Iobrcbr = imcomplement(Iobrcbr);
```

## Čo sa udialo

Realizovali sme operáciu otvorenia a následného zatvorenia, ale za použitia markerov, tj pomocou rekonštrukcie.

# Otvorenie a zatvorenie

## Kód

```
fgm = imregionalmax(Iobrcbr);  
se2 = strel(ones(5,5));  
fgm2 = imclose(fgm,se2);  
fgm3 = imerode(fgm2,se2);  
fgm4 = bwareaopen(fgm3,20);
```

## Čo sa udialo

Vytvorili sme masku popredia z oblastí ktoré sú lokálnymi maximami. Potom sme ju pomocou morfológie upravili a nakoniec sme odstránili oblasti s obsahom menším ako 20 pixelov.

# Nájdenie hrebeňov pozadia

## Kód

```
dist = bwdist(imbinarize(Iobrcbr));  
ws1 = watershed(dist);  
bgm = ws1 == 0;
```

## Čo sa udialo

Zobrali sme širšie popredie a pomocou funkcie `bwdistance`, sme spočítali pre každý pixel vzdialenosť od najbližšieho pixelu popredia. Na tieto vzdialenosti sme aplikovali `watershed` a získali sme tak oblasti ktoré sú hrebeňmi.

# Vynútenie minima a watershed

## Kód

```
grad = imgradient(I);  
grad2 = imimposemin(grad, bgm | fgm4);  
L = watershed(grad2);
```

## Čo sa udialo

Spojili sme masku miest kde chcme mať 'kotliny' a vynútili sme aby na gradientnom obraze boli lokálne minimá len tam. Týmto spôsobom sa samostatne segmentujú objekty a zároveň aj pozadie bude mať svoje povodie. Nakoniec aplikujeme watershed.

# Krajšie zobrazenie obrázka

## Kód

```
figure
imshow(I)
hold on
himage = imshow(Lrgb);
himage.AlphaData = 0.3;
```

## Čo sa udialo

Len sme vykreslil obrázok pekným spôsobom.

# k-means clustering

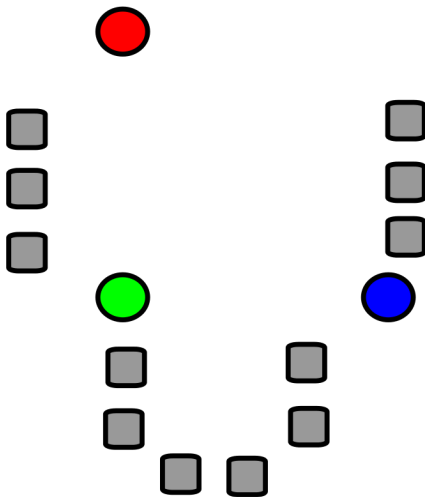
## k-means

k-means clustering je metóda, ktorá zo súboru vektorov vytvorí k skupín, ktoré predstavujú jeden zhluk.

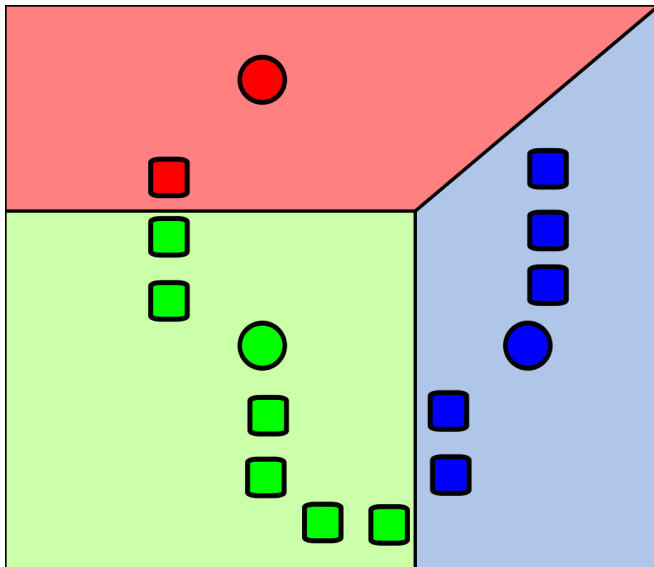
## Postup

- V požadovanom vektorovom priestore náhodne rozmiestnim k 'centroidov'
- Každý vektor priradím centroidu ktorému je najbližšie
- Centroidy posunieme, tak ich nové pozície budú ťažiská ich priradených vektorov
- Opakujeme 2. a 3. bod kým sa centroidy posúvajú

# k-means

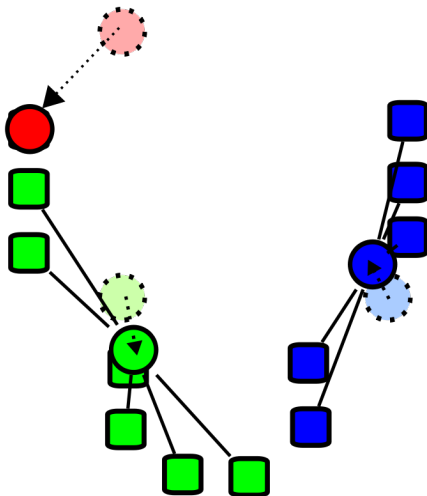


## k-means

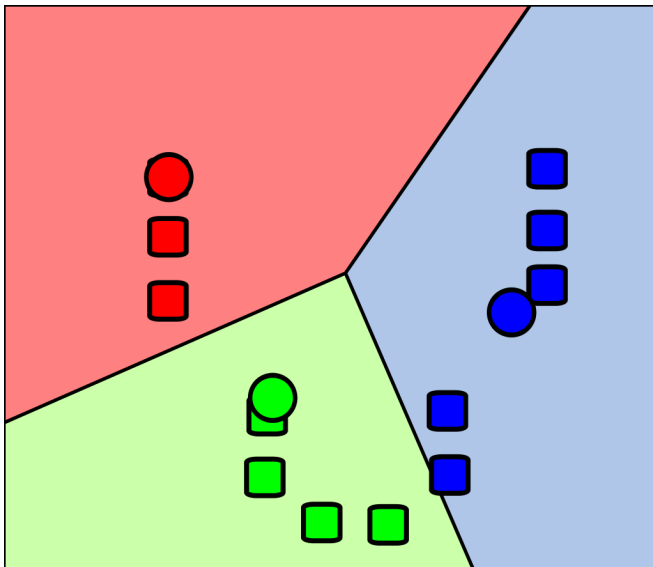




# k-means



## k-means



# k-means v matlabe

## kmeans

`kmeans(A, k)` - pre maticu s  $n$  riadkami, z ktorých každý predstavuje jeden vektor vráti vektor dĺžky  $n$  ktorého prvky sú hodnoty od 1 po  $k$ , podľa toho do ktorého klusteru daný vektor patrí

## Vektory pre segmentáciu obrazu

Pre obrazy môžeme napríklad segmentovať jednotlivé pixely. Ako ich vektory môžeme zobrať

# k-means v matlabe

## Úloha

Použite k-means na obrázok zátišia. Ako vektory vezmite farby v Lab priestore.

## Úloha

Ako vektory použite farbu, ale aj x-ové a y-ové súradnice. Nazabudnite jednotlivé zložky normalizovať.

## meshgrid

$[X, Y] = \text{meshgrid}(1:c, 1:r)$  - vytvorí dve matice rozmerov  $r \times c$ . X obsahuje x-ové súradnice v tejto matici, Y obsahuje y-ové súradnice.

# Graph Cut

## Graph Cut

Graph Cut je metóda ktorá využíva užívateľský vstup na segmentáciu popredia. Užívateľ označí nejaké pixely ako popredie a pozadie.

## Algoritmus

Zo všetkých pixelov sa zostrojí graf, každý pixel je spojený so susednými s váhou, ktorá zodpovedá podobnosti pixelov. V grafe sú ešte dva vrcholy jeden reprezentuje popredie a druhý pozadie. Tieto sú prepojené s pixelmi pomocou pravdepodobnosti, že sú z popredia resp. z pozadia. Túto pravdepodobnosť získame pomocou distribúcie farieb v užívateľom označenými pixelmi. Nakoniec použijeme algoritmus ktorý urobí rez grafom tak, aby minimalizoval energiu, teda váhy hrán ktoré vedú od vrchola popredia k vrcholu pozadia.

# Graph Cut

## Matlab

Graph Cut sa dá použiť aj v matlabe. V záložke apps si nájdite image segmnenter.