

BIZOD cvt04

Segmentace, k-means, binární obrazy, segmentace
silnice

jan.tesar@fbmi.cvut.cz

Segmentace

- ▶ Rozdělit obraz na určité části
 - ▶ Často pozadí od popředí
 - ▶ Oddělení jednotlivých objektů
- ▶ Pro člověka zcela intuitivní
 - ▶ Vychází ze zkušeností
 - ▶ Perspektiva,
 - ▶ Znalost objektů
 - ▶ Znalost stínů,...
- ▶ Pro počítač „složitá“
 - ▶ Proč?



Paní co krávu neviděla

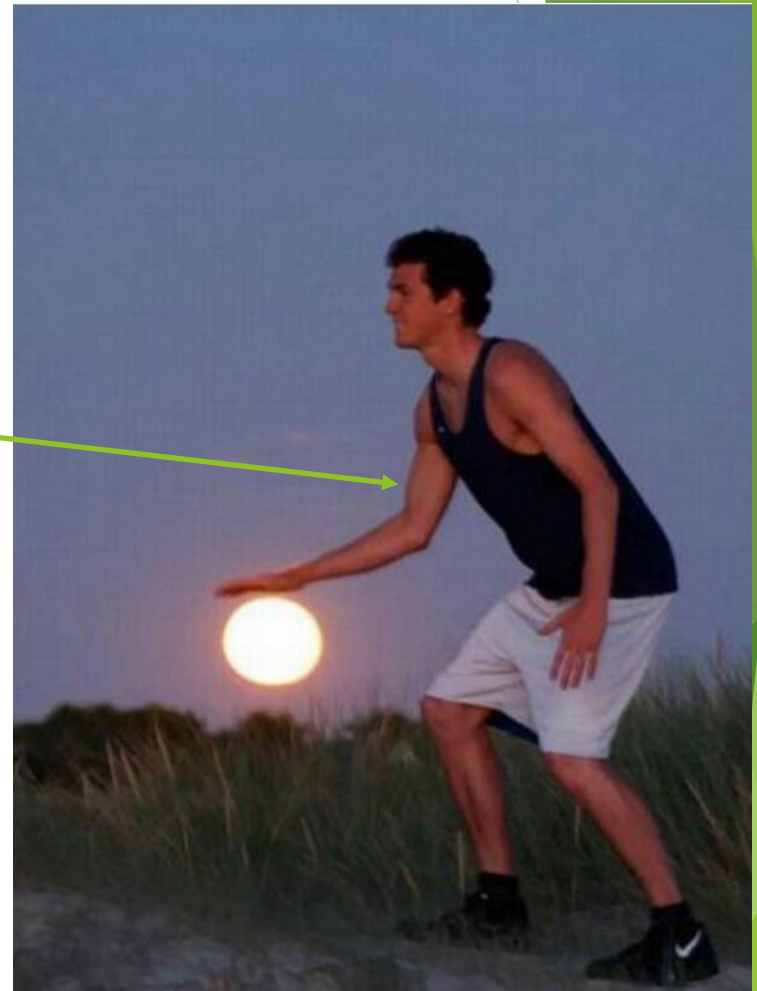
- ▶ Paní S. začala ve středním věku díky chirurgickému zákroku vidět. Dostavilo se však především zmatení. Mozek „nechápal“ příchozí informace. Po nějaké době se naučila rozpoznávat objekty.
- ▶ Při pohledu na louku viděla jen změt' černobílých bodů a kolem body zelené. Krávu rozpoznala až ve chvíli, kdy se natočila bokem - tak jak se ji naučila.
- ▶ Podobně jsou na tom počítače - vnímají body, nemají dlouholetou zkušenost.

Vidění po pixelech



Čemu odpovídá jaký pixel?

Vidění po pixelech

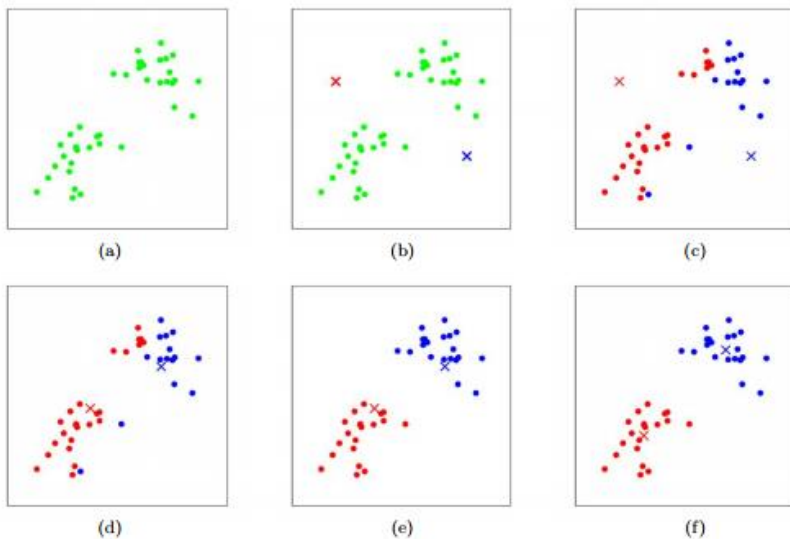


Segmentace kytky

- ▶ Převed'te obraz na binární `imbinarize(Ag,th)`
- ▶ Jakou hodnotu nastavíte pro práh?
- ▶ Otsu threshold - metoda pro získání „správného“ prahu z obrázku. Udělá stejnou věc jako následující k-means algoritmus
- ▶ `th = graythresh(A)`
- ▶ `Th = multithresh(A,N)` - pro více prahů
- ▶ Stále nevýhoda toho, že určitá barva bude vždy segmentována jako jedna kategorie (popředí - pozadí). To řeší pokročilejší metody
- ▶ Segmentujte kytku pomocí otsu metody

k-means

- Obecná metoda na shlukování.
 - Rozdělí data do předem daného počtu shluků.
- Funguje pro jakýkoliv počet parametrů
 - Co parametr, to rozměr, zde 2 parametry (dobrá vizualizace)

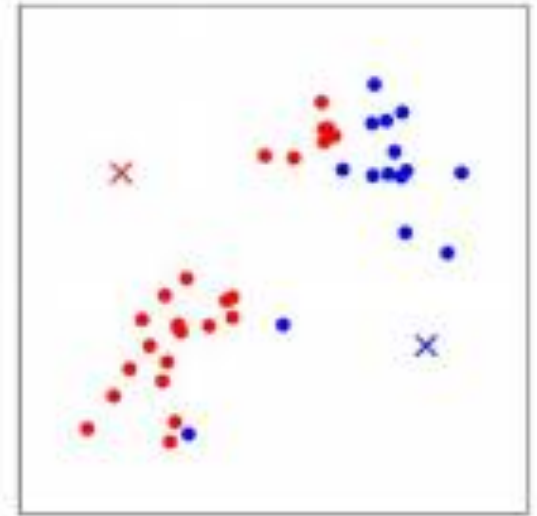




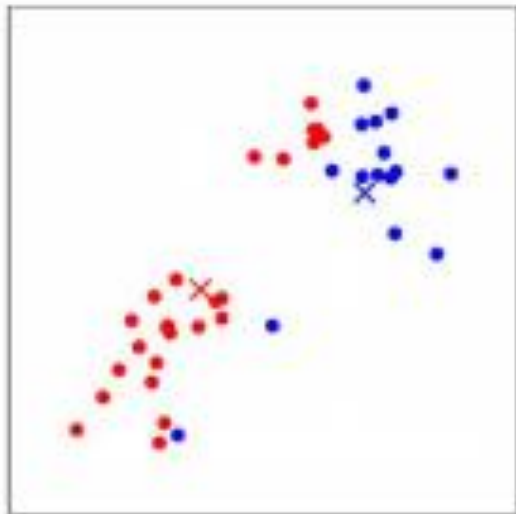
(a)



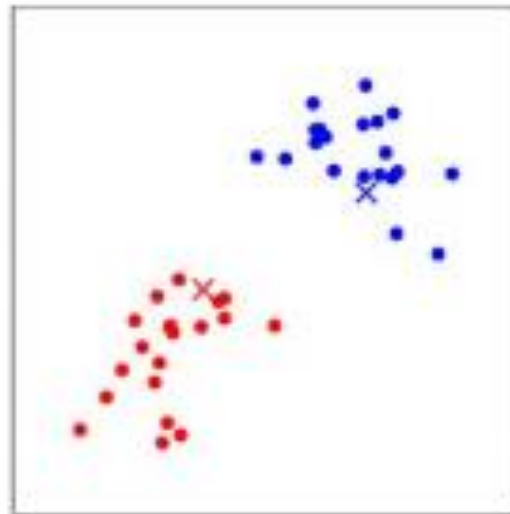
(b)



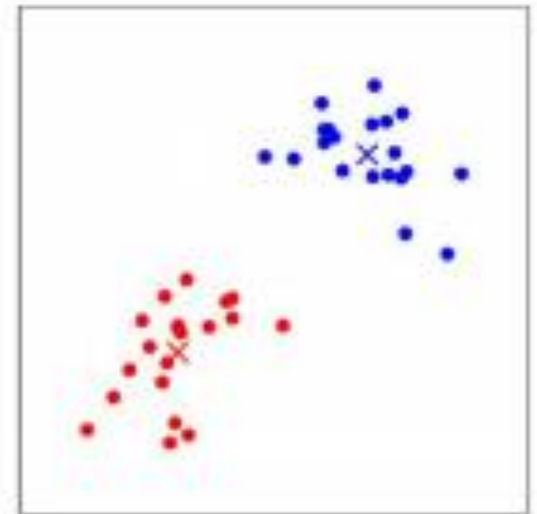
(c)



(d)



(e)



(f)

K-means při segmentaci obrazů

- ▶ Shlukování pixelů podobných vlastností
 - ▶ Často pouze odstín -> 1 parametr -> 1D-> pohyb po histogramu
- ▶ V matlabu implementovaná funkce kmeans
 - ▶ Vstup: vektor hodnot (odstíny pixelů)
počet shluků (K)
 - ▶ Výstup: index shluku pro každý pixel
 - ▶ Nutné „zvektorovat a zmaticovat“ obraz (:,:), reshape)
- ▶ Využijte nastavení colormap (hot, jet,...)
- ▶ Segmentujte kytku
- ▶ Zajímavá úloha - více než 2 třídy, dalším parametrem je souřadnice x a y

Segmentace v hsv



- ▶ Hue (odstín „duhy“), Saturation (mezi barvou a černobílým odstínem), Value (mezi černou a bílou)
 - ▶ Také 3 matice, ale nejsou v nich jednotlivé barvy
- ▶ `rgb2hsv` - trojrozměrná matice (odstín, saturace a hodnota)
- ▶ Proved'te segmentaci s pomocí odstínu
- ▶ Segmentaci vizualizujte zašedotónováním pozadí.

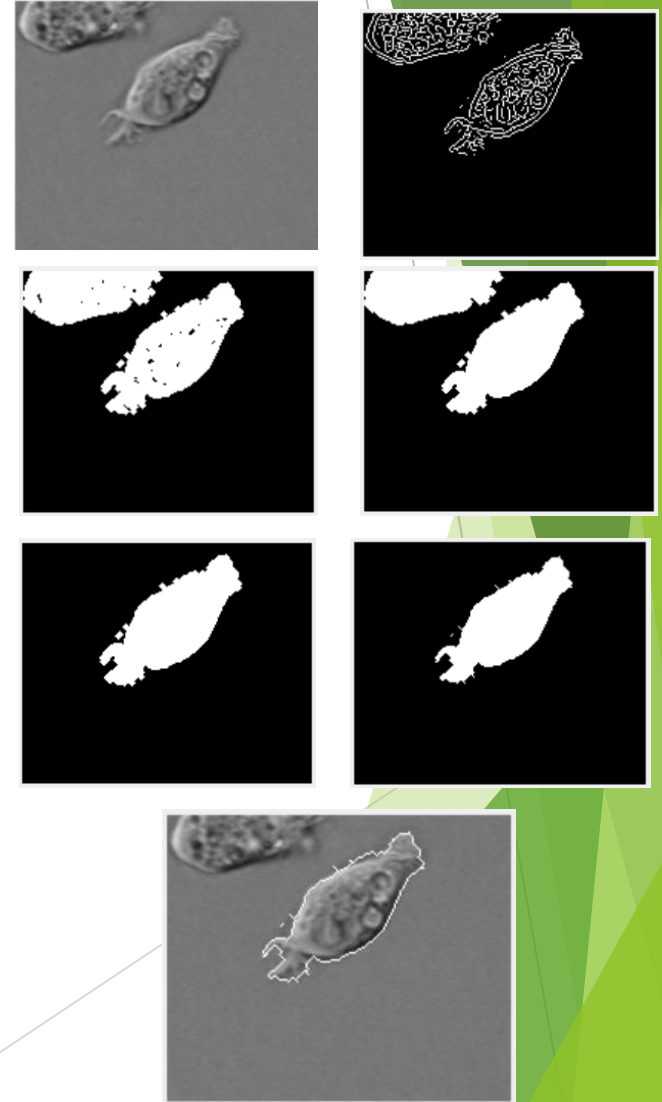


colorThresholder

- ▶ Užitečná Matlab aplikace pro prahování obrázků
- ▶ Různé barevné modely
- ▶ Export výsledku, či kódu
- ▶ Apps -> Image Processing -> Color Thresholder

Segmentace pomocí hran a binární matematické morfologie

- ▶ Načtěte obrázek cell.tif
- ▶ Detekujte hrany (edge)
 - ▶ Zvolte vhodný práh
- ▶ Dilatujte obraz (imdilate)
 - ▶ Přidá a pospojuje pixely
 - ▶ Zvolte vhodný strukturní element (strel)
- ▶ Zaplňte obraz (imfill)
- ▶ X obj. na hraně (imclearborder)
- ▶ Erodujte obraz (imerode)
 - ▶ Vyhladí
 - ▶ Proveďte i víckrát
- ▶ Zobrazte segmentaci (bwperim)



Hledání silnice

- Pomocí podobných postupů jako v předchozí úloze naleznete středovou čáru z čelního výhledu



Hledání silnice 2



- ▶ Funkce které zřejmě budete potřebovat:
 - ▶ `im2bw`
 - ▶ `bwareaopen` - bin. otevření ~ odstranění malých objektů
 - ▶ `imclearborder` - odstranění objektů na hraně obrazu
 - ▶ `bwboundaries` - tvorba regionů
 - ▶ `label2rgb` - obarvení regionů
 - ▶ `regionprops` - vlastnosti regionů
 - ▶ Vlastnost regionů, která se zřejmě bude hodit je „kolečkovost“ (Eccentricity), (od dokonalého kruhu až po čáru)
- ▶ Povšimněte si toho, že „algoritmus“ musíte hodně poladit pro tuto konkrétní úlohu, tento konkrétní obrázek
 - ▶ Snažíme se vytvářet postupy, které fungují v co nejširším záběru (pro různé obrázky silnic)

