Computer simulations in Physics

Segmentation

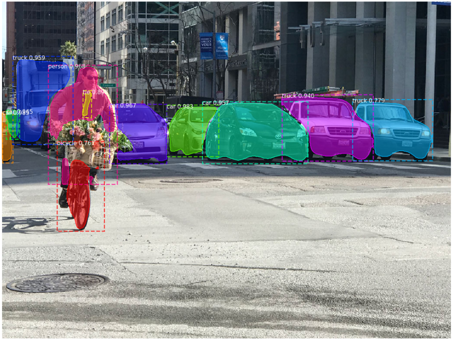
Janos Török

Department of Theoretical Physics

September 15, 2022

Segmentation

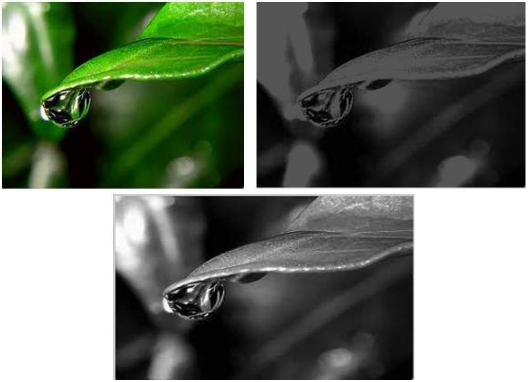
- Aim: Determine objects in an image
- Methods: Thresholding, clustering
- Methods (not covered): edge detection, convolution neural networks



Threshold based methods

- Problem: Image has three variables: red, green, blue
- Solution grayscale
- Formula:

$$gray = 0.3r + 0.59g + 0.11b$$



source: `https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm`

Threshold based methods

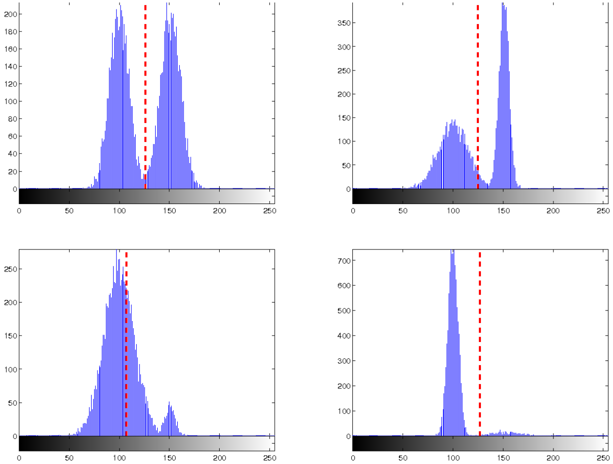
- Threshold T

$$g(x,y) = \begin{cases} 1 & \text{if} f(x,y) \geq T \\ 0 & \text{otherwise} \end{cases}$$

- How to get the threshold T
    - median
    - find a nice valley
    - Otsu's method

Otsu's method

- approximate the intensity distribution by the sum of two Gaussian distribution and minimize the *within-class* variance!



Otsu's method

- approximate the intensity distribution by the sum of two Gaussian distribution and minimize the *within-class* variance!

  $$\sigma_w^2(T) = n_a(T)\sigma_a^2(T) + n_b(T)\sigma_b^2(T),$$

  where $n_a, \ n_b$ is the number of pixels in group a, b, and σ_a, σ_b are the variances of the given group.
- Between variance:

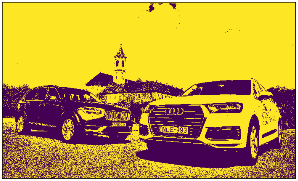
  $$\sigma_B^2(T) = \sigma^2 - \sigma_w^2(T) = n_a(T)(\mu_a(T) - \mu)^2 + n_b(T)(\mu_b(T) - \mu)^2$$

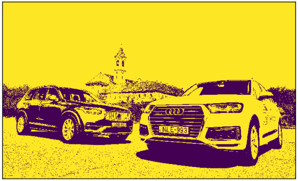
  where μ is the mean. So

  $$\sigma_B^2(T) = n_a(T)n_b(T)[\mu_a(T) - \mu_b(T)]^2$$
- One has to maximize $\sigma_B^2(T)$

Threshold based methods

gray, mean, Otsu



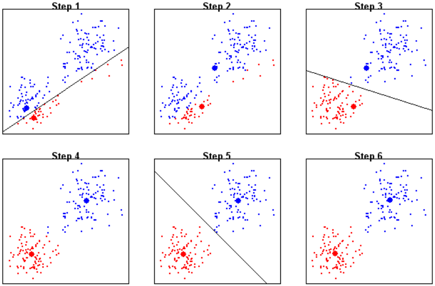


Other threshold methods

- Kapur (Graph. Models Image Process., 29 (1985), pp. 273-285)
- Rosin (Pattern Recognition, 34 (2001), pp. 2083-2096)
- Medina-Carnicer (Pattern Recognition, 41 (2008), pp. 2337-2346)

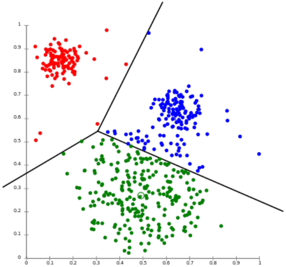
k-means clustering

- Cut the system into exactly k parts
- Let μ_i be the mean of each cluster (using a metric)
- The cluster i is the set of points which are closer to μ_i than to any other μ_j
- The result is a partitioning of the data space into Voronoi cells



k-means clustering, standard algorithm:

- Define a norm between nodes
- Give initial positions of the means m_i
- Assignment step: Assign each node to cluster whoose mean m_i is the closest to node.
- Update step: Calculate the new means of the clusters
- Go to Assignment step.



k-means clustering: Major usage

- Detection of connected parts in images
- Use the Red, Green, Blue value of each pixel
- Put them on a 3d space
- Find relevant clusters

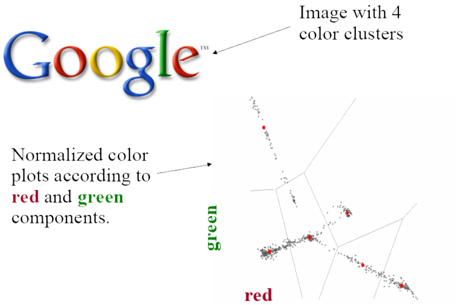
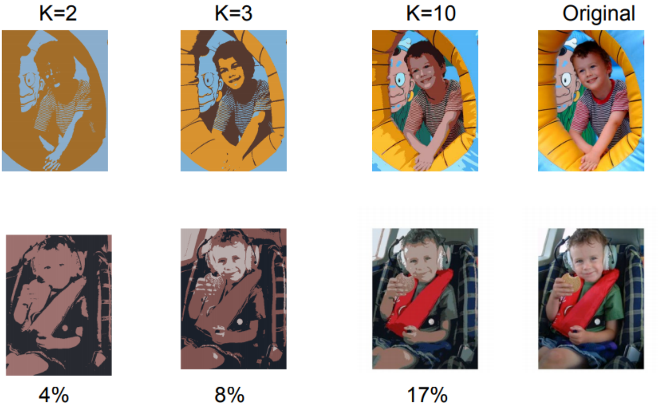


Image with 4 color clusters

Normalized color plots according to **red** and **green** components.

k-means clustering: Image color segmentation

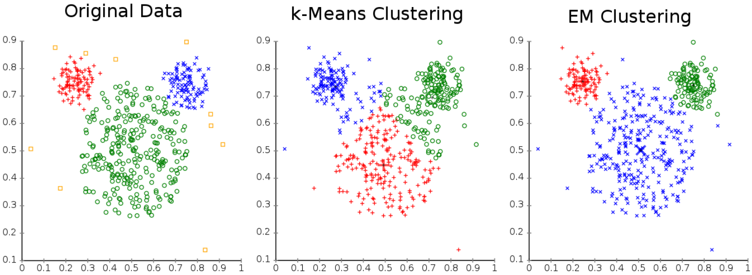
- Detection of connected parts in images
- Use the Red, Green, Blue value of each pixel
- Put them on a 3d space
- Find relevant clusters
- Use the center instead of each color
- Define connected clusters as objects on image



k-means clustering: Problems

- *k* has to fixed beforhand
- Fevorizes equal sized clusters:

Different cluster analysis results on "mouse" data set:

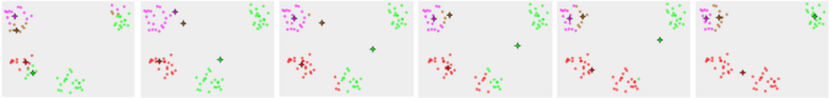


Original Data

k-Means Clustering

EM Clustering

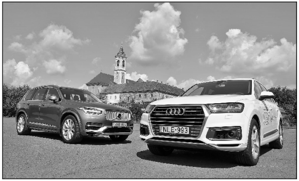
- Very sensitive on initial conditions:

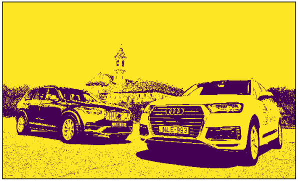


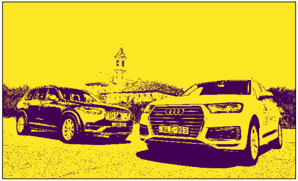
- No guarantee that it converges

K-means clustering

gray, Otsu, k-means

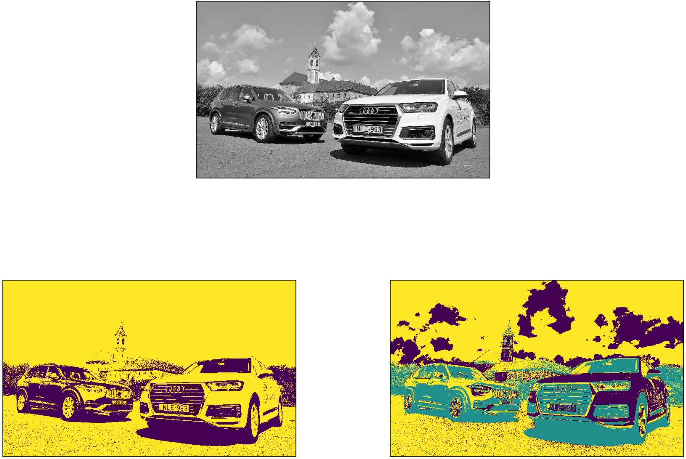






K-means clustering

gray, k-means 2, k-means 3



Connected components

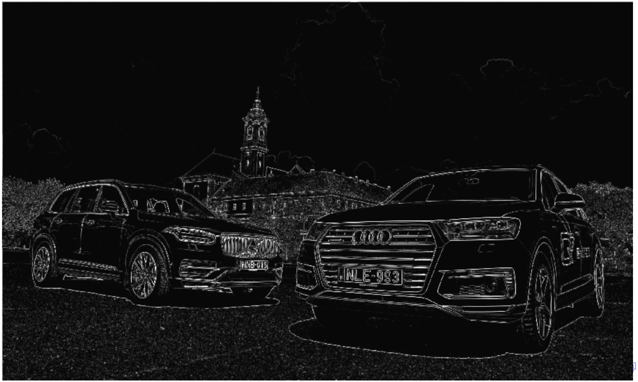
- We have a segmented image
- Find connected components → image segments
- Decide whether there is a background
- Zillion of algorithms
- Once the patch is found work with it:
  - rotate if necessary
  - pattern matching
  - feed it to neural network

Other methods

- complete thresholding with region growing
- Edge detection

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

- Find lines on edges
- Domain filling with edges



Pattern/template matching

- Have a pattern
- Size and orientation MUST match
- Sweep through the image and calculate correlations (f image, g template)
  - Correlation (problem with intensity

$$C_{xy} = \sum_{ij} f(x+i, y+j) g(i,j)$$

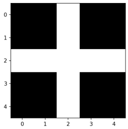
  - Correlation zero mean template

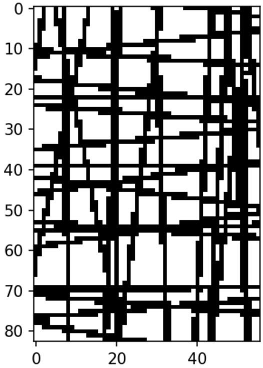
$$C_{xy} = \sum_{ij} (f(x+i, y+j) - \bar{f})(g(i,j) - \bar{g})$$

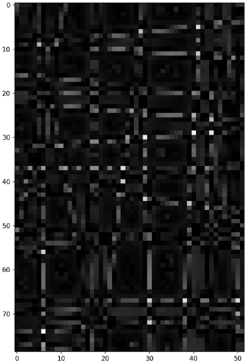
  - Sum of Squared Differences

$$C_{xy} = \sum_{ij} [f(x+i, y+j) - g(i,j)]^2$$

Template matching


0
1
2
3
4
0
1
2
3
4


0
10
20
30
40
50
60
70
80
0
20
40


0
10
20
30
40
50
60
70
0
10
20
30
40
50