Computer Vision (ZDO) - Segmentation

Introduction, Image thresholding

Zdeněk Krňoul, Ph.D.

Department of Cybernetics
FACULTY OF APPLIED SCIENCES
University of West Bohemia

DEPARTMENT OF CYBERNETICS

Content:

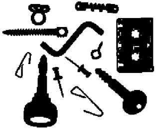
- Introduction to image segmentation
- Basic segmentation techniques
- Image thresholding
- Searching a threshold value

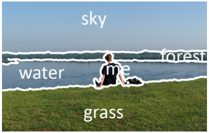
DEPARTMENT OF CYBERNETICS

Image Segmentation

INPUT: INTENSITY IMAGE

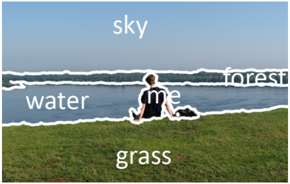
OUTPUT: IMAGE DIVIDED INTO THE WORLD-RELATED PARTS

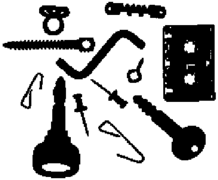




- The next level of image processing (does the following levels processing easier);
- Reduces the complexity by assigning a label to each pixel.

Complete segmentation:





- The segmented areas correspond to the objects in the input image;
- For contrasting backgrounds and objects with constant brightness - we achieve good results at this level of image processing.;
- In general, however, collaboration with a higher level of processing is required, i.e. using knowledge about the problem.

Partial segmentation:

- Segmented areas do not correspond to objects on the input image
- We get only the parts with semantic meaning;
- Segmented areas are homogeneous with respect to certain properties (brightness, color, texture, set of edges, etc.)
- Segmented areas may generally overlap;
- Additional processing needs to be applied - object description and detection.

Examples: text segmentation, blood cells, back-light illuminated details inspected in industry - screw counting

Image segmentation - what method to use?

The more knowledge we have, the better. The options are:

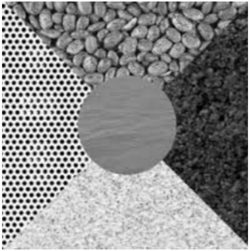
- Required shape and appearance;
- Given pose (translation, scale, orientation)
- The start and end points of the boundary;
- The relationship of the area to other areas with the given properties.

Examples:

- *search for ships at sea*
- *geometric properties of railways, motorways (eg maximum curvature, etc.)*
- *the rivers do not intersect*

Main methods of segmentation:

- brightness analysis - image thresholding
- creating areas by boundaries (Active contours, ...)
- creating areas by processing regions (Split and merge; MRF; Graph-cut)
- template matching (keypoint matching)
- texture segmentation (Gabor filters, ...)
- segmentation with the model of shape and appearance (ASM, AAM, ...)
- various combinations



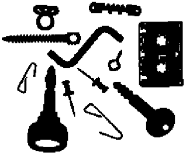
DEPARTMENT OF CYBERNETICS

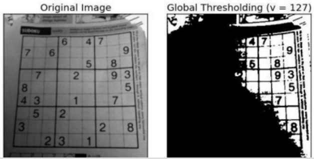
Image thresholding

$$g(i,j) = \begin{cases} 1 & \text{for } f(i,j) \geq T \\ 0 & \text{for } f(i,j) < T \end{cases} \qquad (1)$$

T – threshold value – constant scalar (vector)

- ☺ the simplest and most commonly used;
- ☺ low hardware requirements ... fast - can be computed in real-time
- ☹ determining the threshold - very difficult to do automatically
- ☹ for images, where objects and backgrounds are distinguishable in brightness





DEPARTMENT OF CYBERNETICS

Variant 2: image thresholding by a set of known image brightnesses

$$g(i,j) = \begin{cases} 0 & \text{for } f(i,j) \in D \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

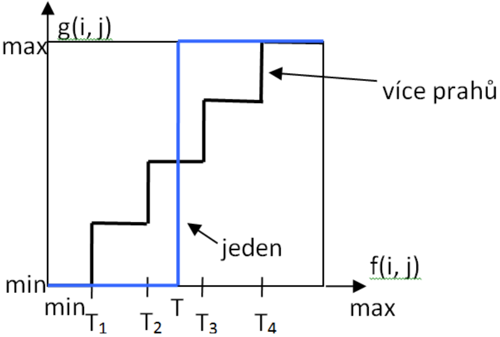
where D is the set of image pixel values corresponding to the background

E.g. human face (skin color) or blood cell images - the cytoplasm appears in a certain range of brightness, the background is lighter, the nucleus is darker

Variant 3: image thresholding with multiple thresholds

$$g(i,j) = \begin{cases} 1 & \text{for } f(i,j) \in D_1 \\ 2 & \text{for } f(i,j) \in D_2 \\ \vdots & \\ n & \text{for } f(i,j) \in D_n \\ 0 & \text{other} \end{cases} \quad (3)$$

where $D_i \cap \ D_j = 0 \ i \neq j$



variant 4: partial image thresholding

$$g(i,j) = \begin{cases} f(i,j) & \text{pro } f(i,j) \in D \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where D is a set of brightnesses corresponding to multiple regions, for example, several objects

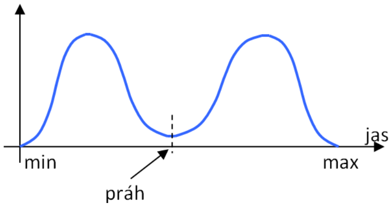
- We remove the background but keep the brightness in the areas. For example, it is used in human visual assessment of results.
- $f(i,j)$ **doesn't have to be** just a brightness function (e.g. gradient value, local texture properties, depth map, color - RGB, hue, saturation, etc.).

Adaptive thresholding

- One global threshold value for the image may product segmentation errors;
- **Problem**: the image has different lighting conditions in different places;
- In this case, adaptive thresholding calculates the threshold for small regions of the image (sliding window)

Searching for a threshold value

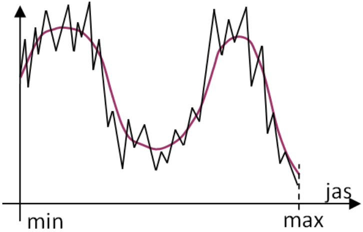
- Usually thresholding uses a threshold value as an input parameter (is known);
- How to determine this threshold **automatically**?; we can use the ”trial and error”technique;
- We can use the histogram:



- a bimodal histogram (2 maxima far enough apart) is a good case;
- in this case the threshold can be identified as the value between the two hills;

Histogram smoothing:

- We are looking for a local minimum between the two largest and sufficiently distant local maxima
- but often the decision **is not robust**



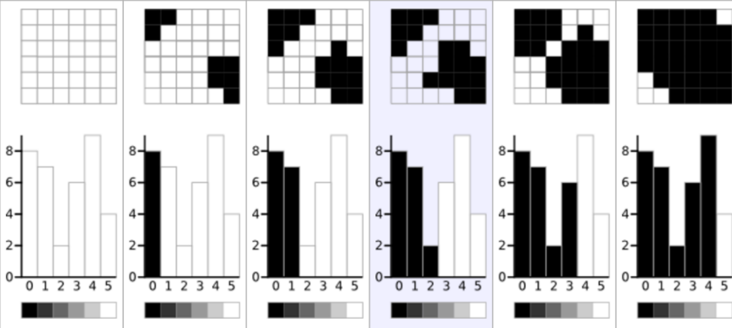
- Histogram smoothing suppresses local extremes and ideally provides a bimodal histogram (local averaging - e.g. Gaussian window or median filtering etc.).

Otsu's Binarization

- **automatic** threshold search method (Nobuyuki Otsu)
- Principle:
  - discrimination of two classes (histogram hills)
  - provides the optimal threshold value T
- Algorithm **minimizes** the weighted variance σ_W^2 of two brightness classes - ie background class b and foreground class f
- $\sigma_W^2(t) = W_b(t)\sigma_b^2(t) + W_f(t)\sigma_f^2(t)$

Otsu's Binarization - Algorithm

- $\sigma_W^2(t) = W_b(t)\sigma_b^2(t) + W_f(t)\sigma_f^2(t)$
- where:
  $W_b(t) = \sum_{i=1}^{t} P(i)$
  $W_f(t) = \sum_{i=t+1}^{I} P(i)$
- $M_b(t) = \sum_{i=1}^{t} \frac{iP(i)}{W_b(t)}$ $M_f(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{W_f(t)}$
- $\sigma_b^2(t) = \sum_{i=1}^{t}[i - M_b(t)]^2 \frac{P(i)}{W_b(t)}$
  $\sigma_f^2(t) = \sum_{i=t+1}^{I}[i - M_f(t)]^2 \frac{P(i)}{W_f(t)}$
- optimal threshold value $T = argmin_{\forall t}\sigma_W^2(t)$ so the overall variance of both hills is minimal



Threshold	T=0	T=1	T=2	T=3	T=4	T=5
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2{}_b = 0$	$\sigma^2{}_b = 0$	$\sigma^2{}_b = 0.2489$	$\sigma^2{}_b = 0.4637$	$\sigma^2{}_b = 1.4102$	$\sigma^2{}_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2{}_f = 3.1196$	$\sigma^2{}_f = 1.9639$	$\sigma^2{}_f = 0.7755$	$\sigma^2{}_f = 0.5152$	$\sigma^2{}_f = 0.2130$	$\sigma^2{}_f = 0$
Within Class Variance	$\sigma^2{}_W = 3.1196$	$\sigma^2{}_W = 1.5268$	$\sigma^2{}_W = 0.5561$	$\boldsymbol{\sigma^2{}_W = 0.4909}$	$\sigma^2{}_W = 0.9779$	$\sigma^2{}_W = 2.2491$

Obrázek: source: http://computervisionwithvaibhav.blogspot.cz

Percentage thresholding

WE have a priori knowledge of what **percent** of the image area is covered by objects;For example, the text in a printed page is around 5 %We set the threshold value so that just as many percent of the pixels have the color of the objects, the rest the background color;See fig. **cumulative histogram** for 2x different lighted scenes 70 %