Image Segmentation

- Gray-level thresholding
- Supervised vs. unsupervised thresholding
- Binarization using Otsu's method
- Locally adaptive thresholding
- Maximally stable extremal regions
- Color-based segmentation
- Region labeling and counting
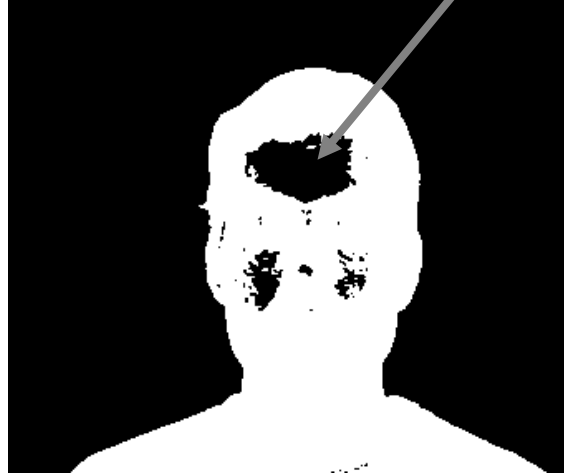- Region moments

Gray-level thresholding

How can holes be filled?

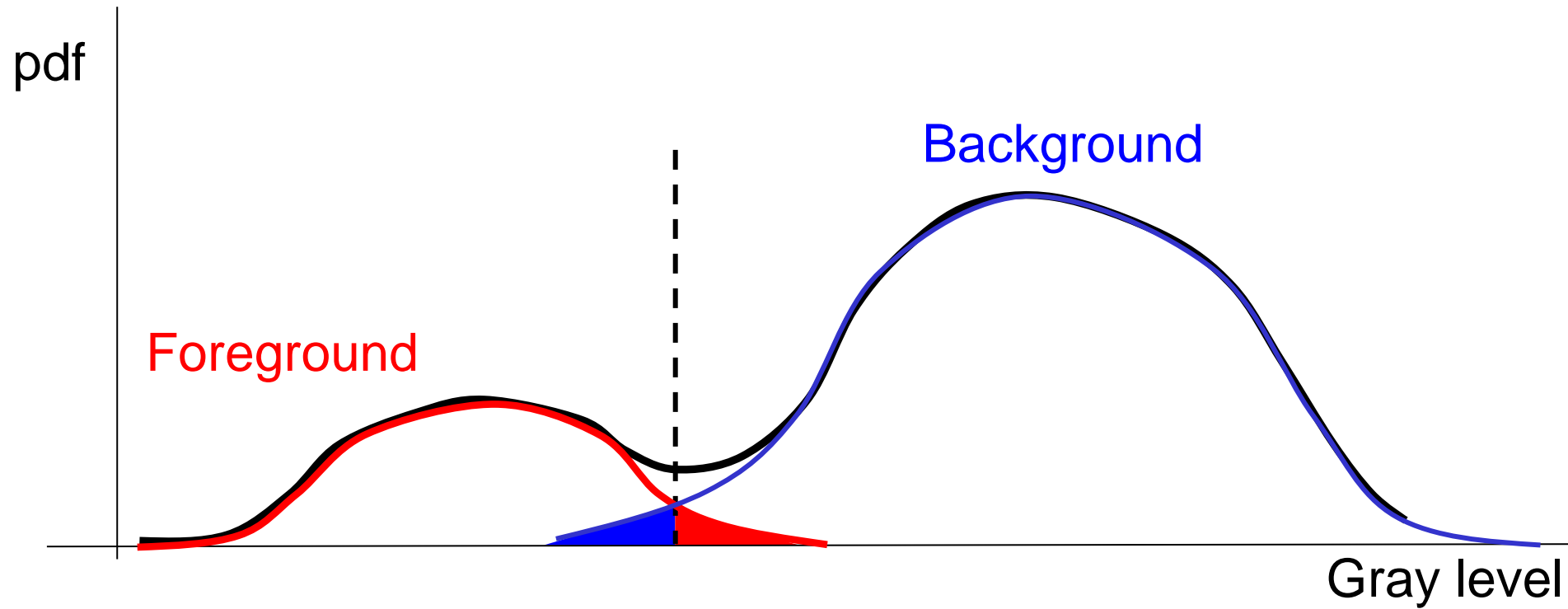Original image
Peter $f[x,y]$

Thresholded
Peter $m[x,y]$

$$f[x,y] \cdot m[x,y]$$

How to choose the threshold?

Unsupervised thresholding

- Idea: find threshold T that minimizes *within-class variance* of both foreground and background (same as k-means)

$$\sigma^2_{within}(T) = \frac{N_{Fgrnd}(T)}{N}\sigma^2_{Fgrnd}(T) + \frac{N_{Bgrnd}(T)}{N}\sigma^2_{Bgrnd}(T)$$

- Equivalently, maximize *between-class variance*

$$\sigma^2_{between}(T) = \sigma^2 - \sigma^2_{within}(T)$$

$$= \left(\frac{1}{N}\sum_{x,y}f^2[x,y] - \mu^2\right) - \frac{N_{Fgrd}}{N}\left(\frac{1}{N_{Fgrd}}\sum_{x,y\in Fgrnd}f^2[x,y] - \mu^2_{Fgrnd}\right) - \frac{N_{Bgrnd}}{N}\left(\frac{1}{N_{Bgrnd}}\sum_{x,y\in Bgrnd}f^2[x,y] - \mu^2_{Bgrnd}\right)$$

$$= -\mu^2 + \frac{N_{Fgrnd}}{N}\mu^2_{Fgrnd} + \frac{N_{Bgrnd}}{N}\mu^2_{Bgrnd} = \frac{N_{Fgrnd}}{N}\left(\mu_{Fgrnd} - \mu\right)^2 + \frac{N_{Bgrnd}}{N}\left(\mu_{Bgrnd} - \mu\right)^2$$

$$= \frac{N_{Fgrnd}(T)\cdot N_{Bgrnd}(T)}{N^2}\left(\mu_{Fgrnd}(T) - \mu_{Bgrnd}(T)\right)^2$$

[Otsu, 1979]

Unsupervised thresholding (cont.)
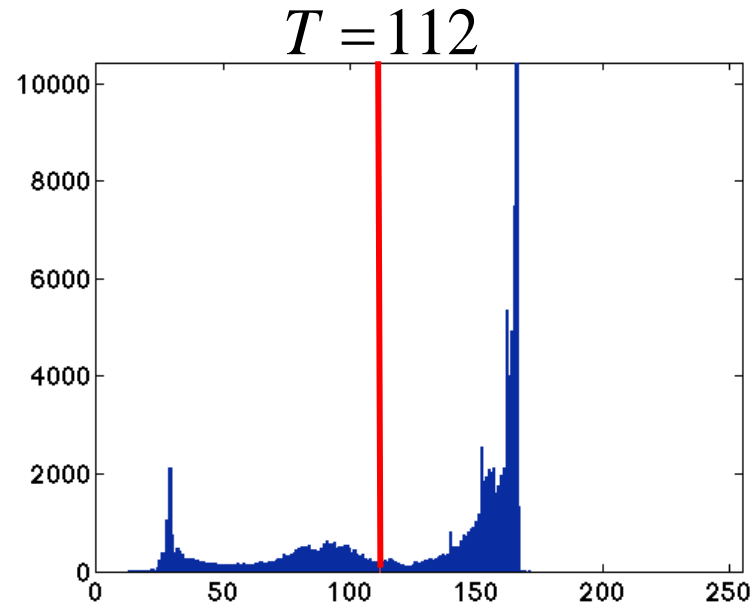
- Algorithm: Search for threshold T to maximize

$$\sigma_{between}^2(T) = \frac{N_{Fgrnd}(T) \cdot N_{Bgrnd}(T)}{N^2} \left(\mu_{Fgrnd}(T) - \mu_{Bgrnd}(T) \right)^2$$
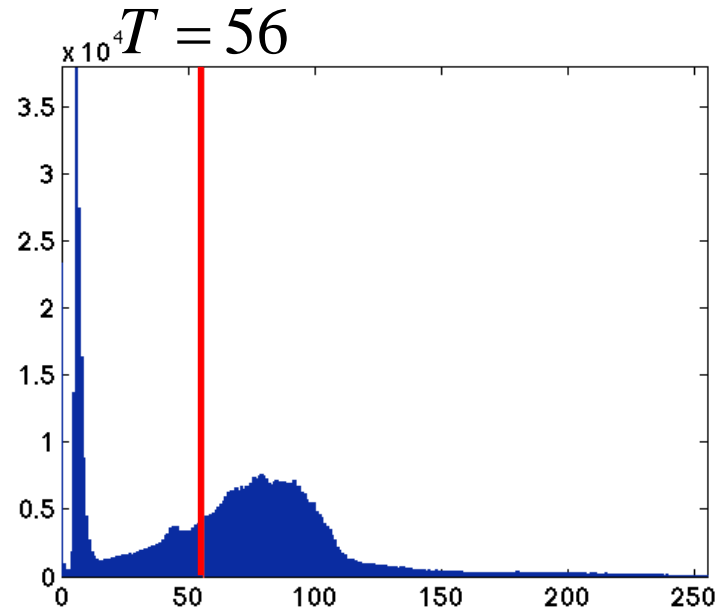
- Useful recursion for sweeping T across histogram:

$$N_{Fgrnd}(T+1) = N_{Fgrnd}(T) + n_T$$

$$N_{Bgrnd}(T+1) = N_{Bgrnd}(T) - n_T$$

$$\mu_{Fgrnd}(T+1) = \frac{\mu_{Fgrnd}(T) N_{Fgrnd}(T) + n_T T}{N_{Fgrnd}(T+1)}$$

$$\mu_{Bgrnd}(T+1) = \frac{\mu_{Bgrnd}(T) N_{Bgrnd}(T) - n_T T}{N_{Fgrnd}(T+1)}$$

[Otsu, 1979]

Unsupervised thresholding (cont.)

$T = 112$

Unsupervised thresholding (cont.)

$$T = 56$$

Unsupervised thresholding (cont.)

$$T = 168$$

Unsupervised thresholding (cont.)

$$T = 144$$

Sometimes, a global threshold does not work

Original image
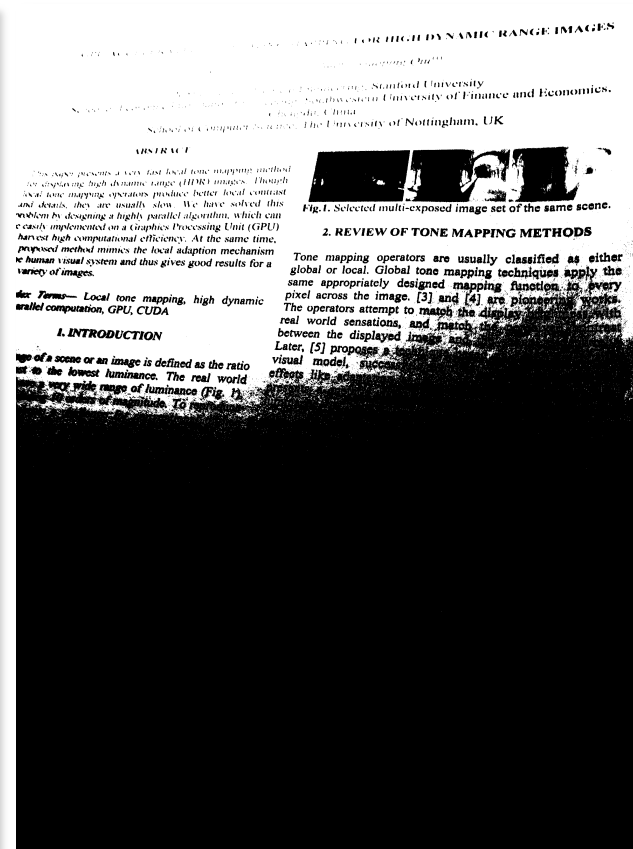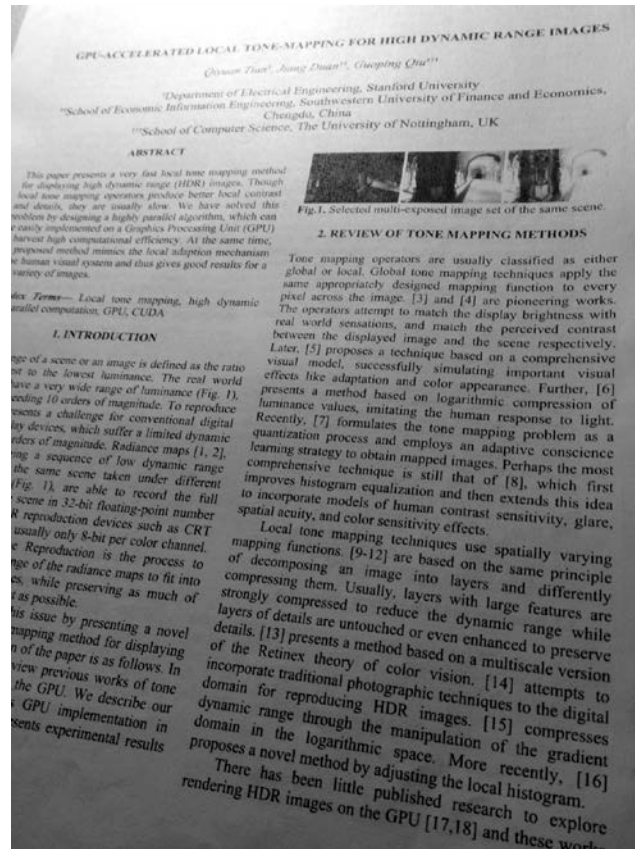
Thresholded with Otsu's Method

Locally adaptive thresholding

- Slide a window over the image

- For each window position, decide whether to perform thresholding
  - Thresholding should not be performed in uniform areas
  - Use variance or other suitable criterion

- Non-uniform areas: apply Otsu's method (based on local histogram)

- Uniform areas: classify the entire area as foreground or background based on mean value
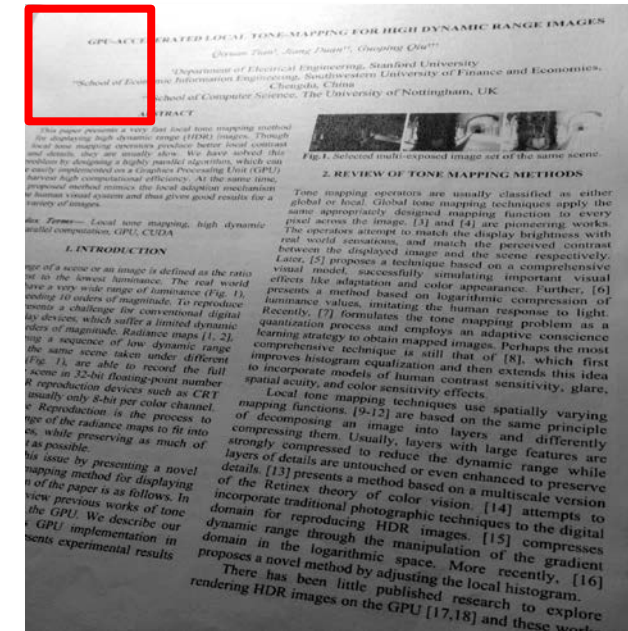
Locally adaptive thresholding (example)
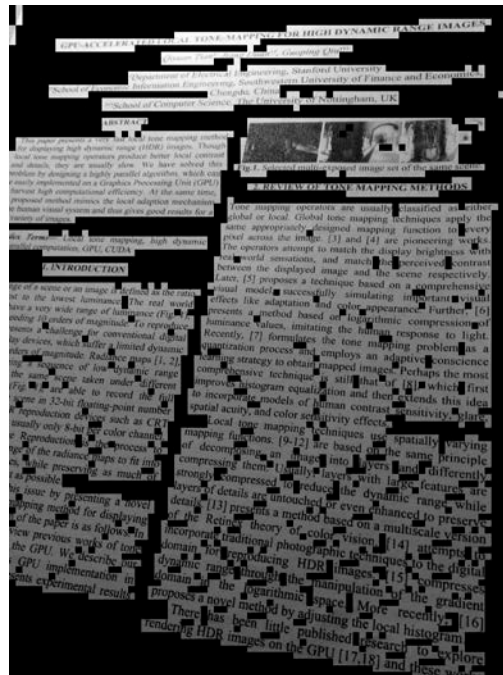
Non-uniform areas

Local threshold values
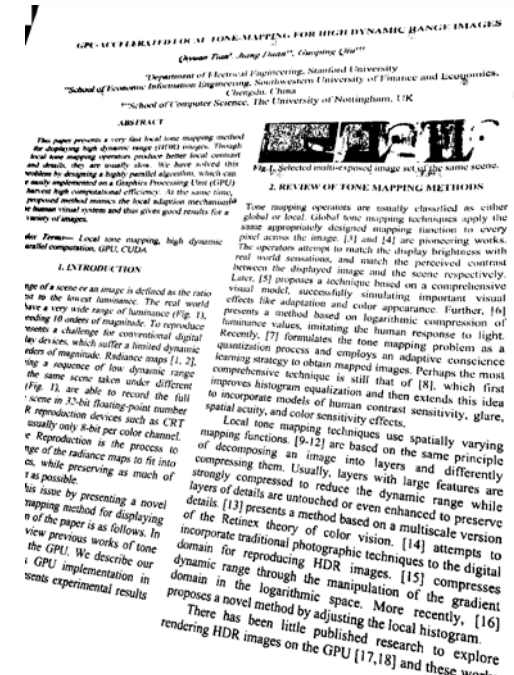
Locally thresholded result

Maximally stable extremal regions

- Extremal region: any connected region in an image with all pixel values above (or below) a threshold

- Observations:
  - Nested extremal regions result when the threshold is successively raised (or lowered).
  - The nested extremal regions form a "component tree."

- Key idea: choose thresholds θ such that the resulting bright (or dark) extremal regions are nearly constant when these thresholds are perturbed by $+/-\Delta$

$\rightarrow$ ***"maximally stable" extremal regions (MSER)***

[Matas, Chum, Urba, Pajdla, 2002]
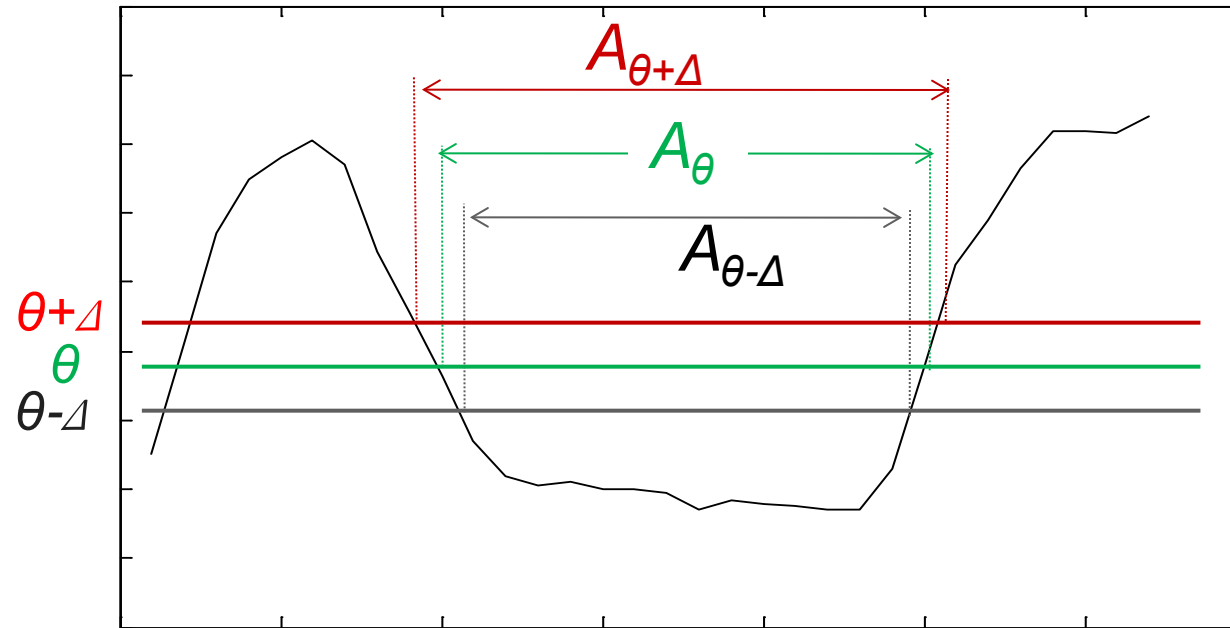
$$\text{Local minimum of } \left| \frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_{\theta}} \right| \rightarrow \text{ MSER}$$

[Matas, Chum, Urba, Pajdla, 2002]

Level sets of an image

$$f[x, y]$$

Image

$$f[x, y] > 8$$

Level Set

Component tree of an image
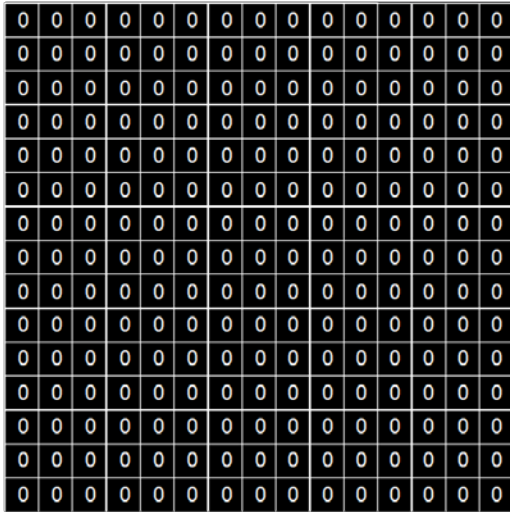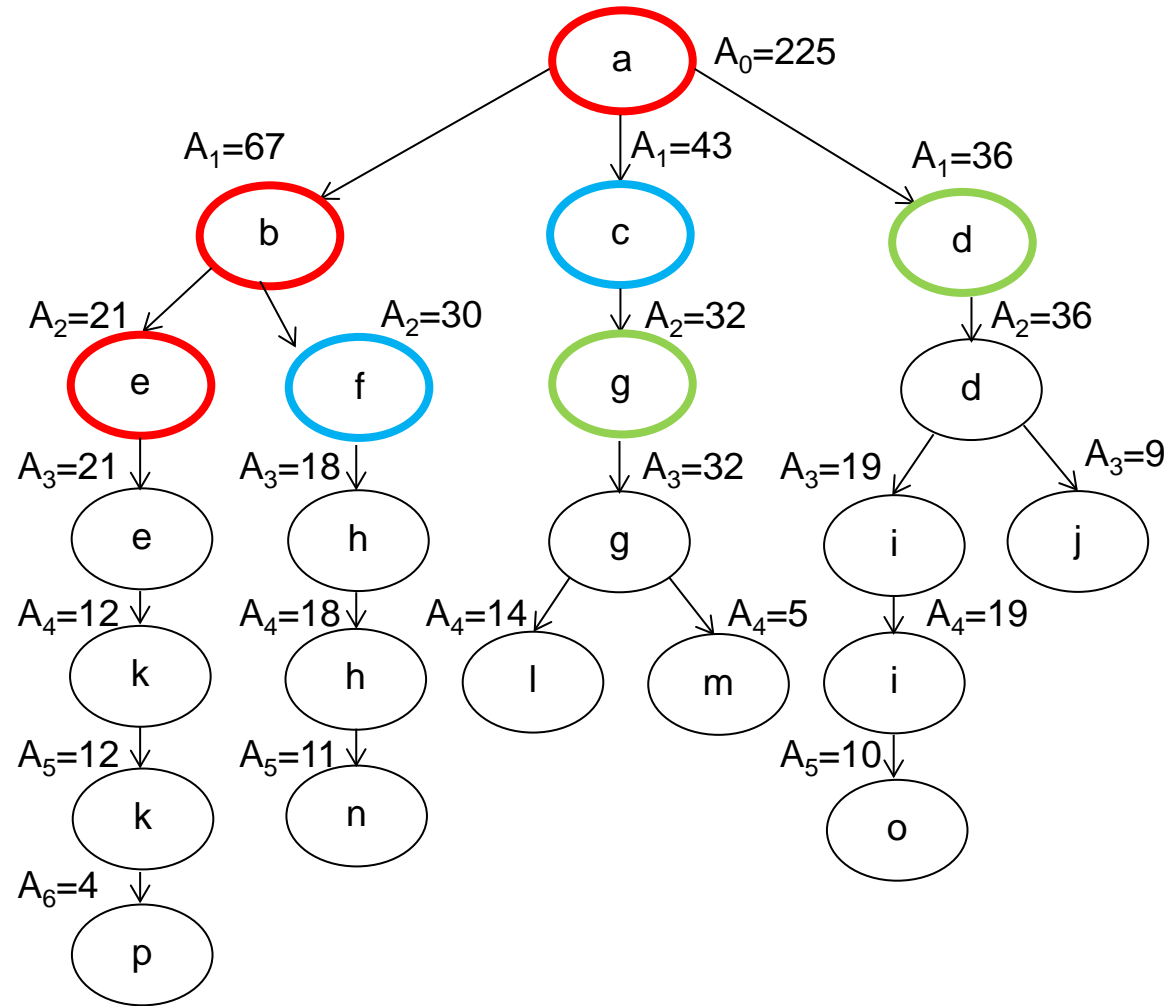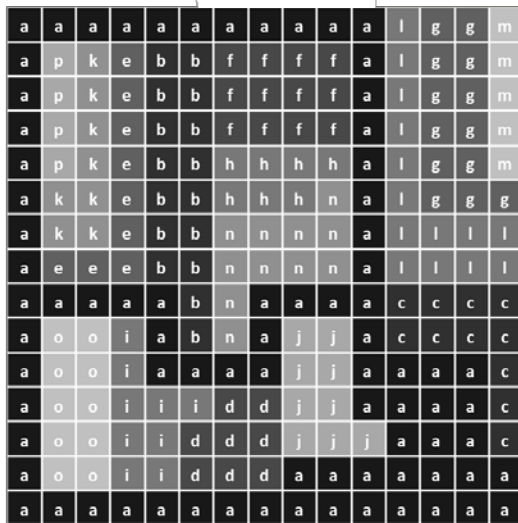
MSER: examples

Dark MSERs, Δ=15

Original image

Bright MSERs, Δ=15

MSER: examples

Dark MSERs, Δ=15

Original image

Bright MSERs, Δ=15

Supervised thresholding

pdf

Background

Foreground

error
probability

Gray level
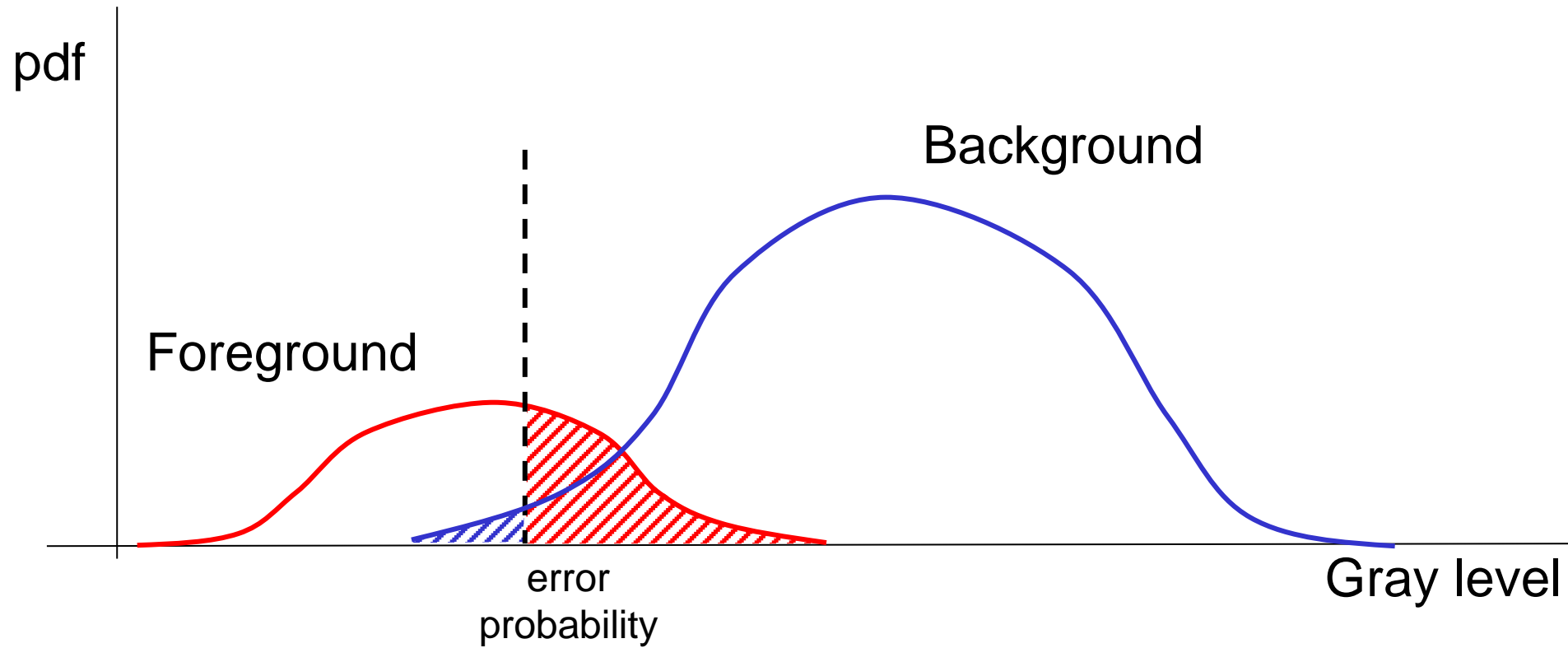
Supervised thresholding
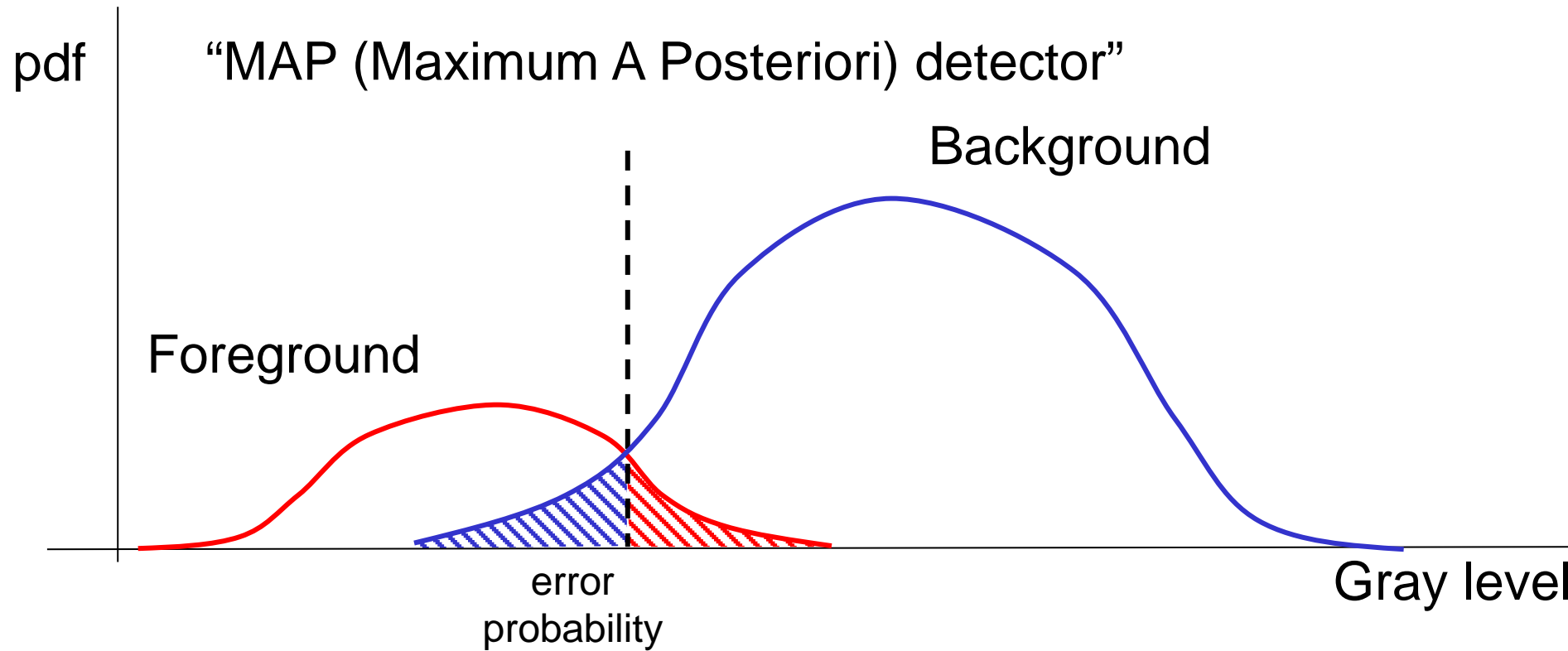
Multidimensional MAP detector

- Training
  - Provide labelled set of training data
  - Subdivide n-dimensional space into small bins
  - Count frequency of occurrence for each bin and class in training set, label bin with most probable class
  - (Propagate class labels to empty bins)
- For test data: identify bin, look up the most probable class
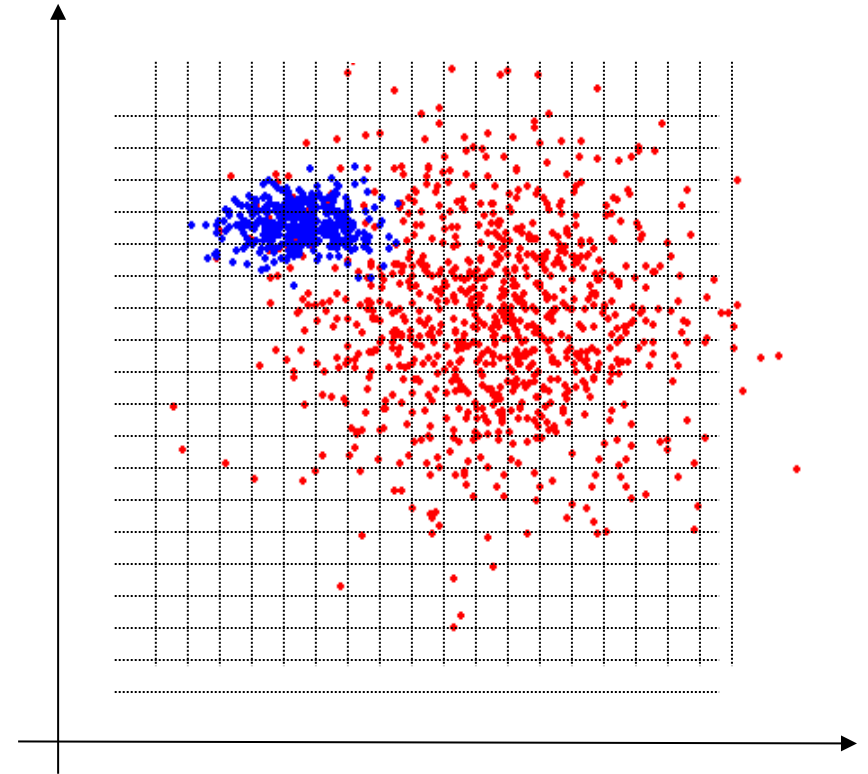
MAP detector in RGB-space

Original image

Skin color detector

Non-skin Samples

Skin Samples
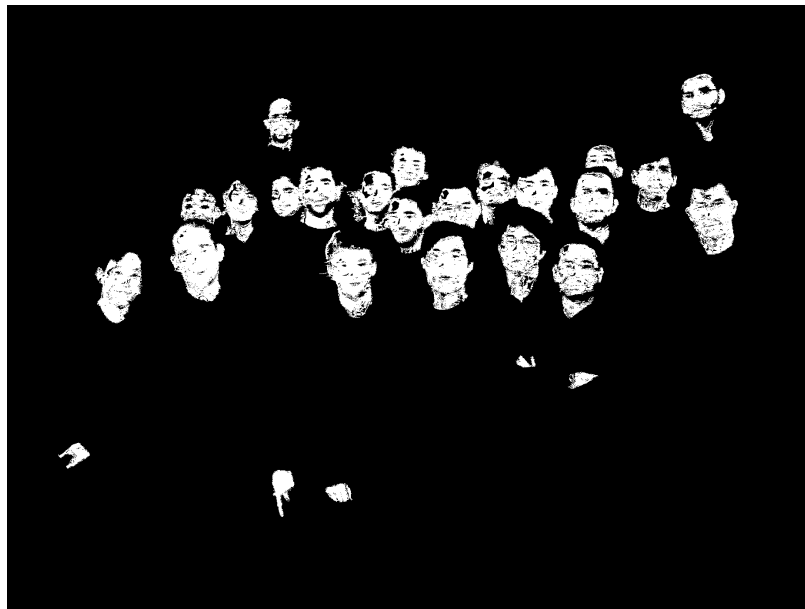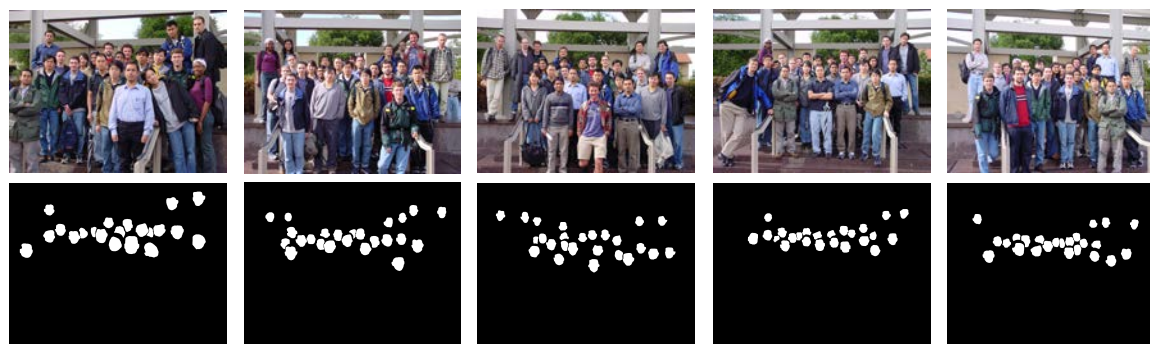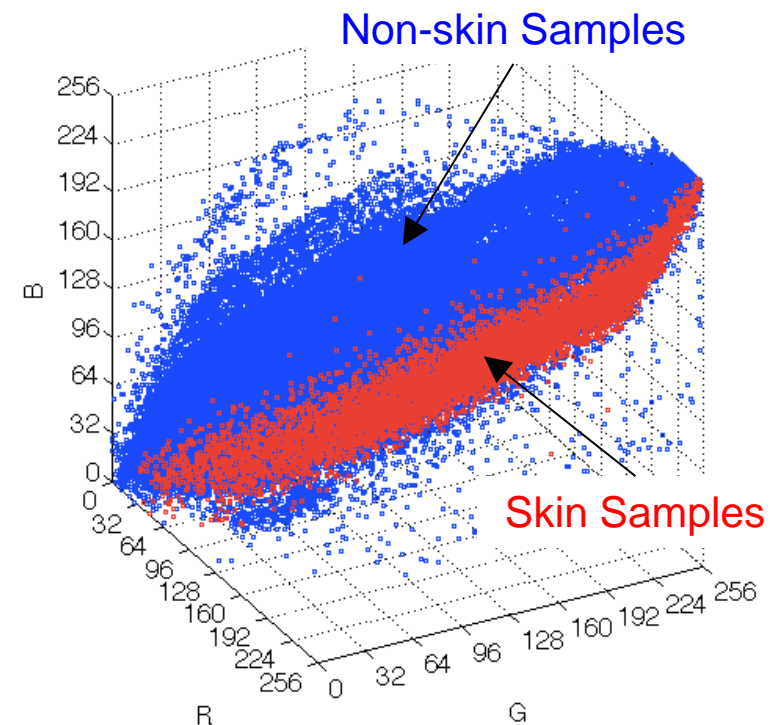
Five training images

Linear discriminant function

- To segment image with n components $f_i,\ i=1,2,\ldots,n$ into two classes, perform test

$$\sum_i w_i f_i + w_0 \geq 0 \quad ?$$

- Categories are separated by hyperplane in n-space
- Numerous techniques to determine weights $w_i,\ i=0,1,2,\ldots,n$, see, e.g., *[Duda, Hart, Stork, 2001]*
- Can be extended to the intersection of several linear discrimant functions
- Can be extended to multiple classes
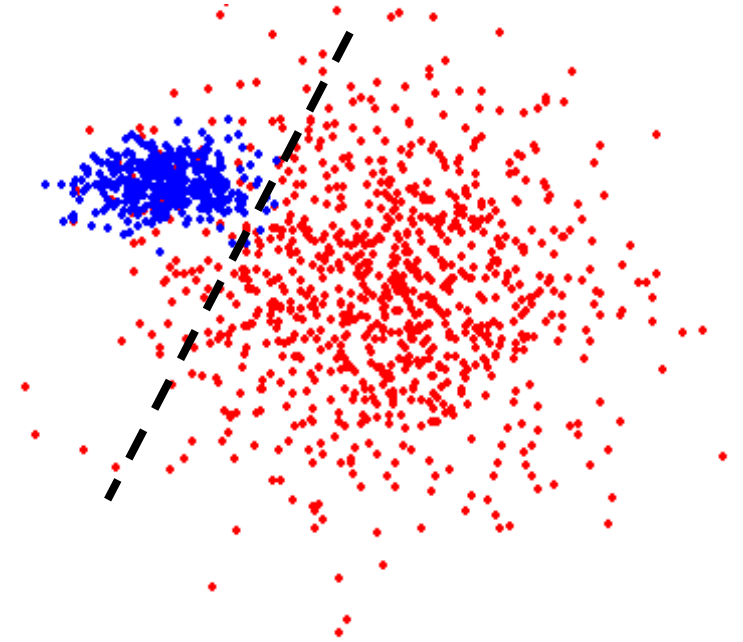
Chroma keying

Landsat image processing

Original Landsat image false color picture out of bands 4,5,6

San Francisco Bay
April 11, 1988

Water area segmented and enhanced to show sediments

Water Only
Linear Stretch
April 11, 1988
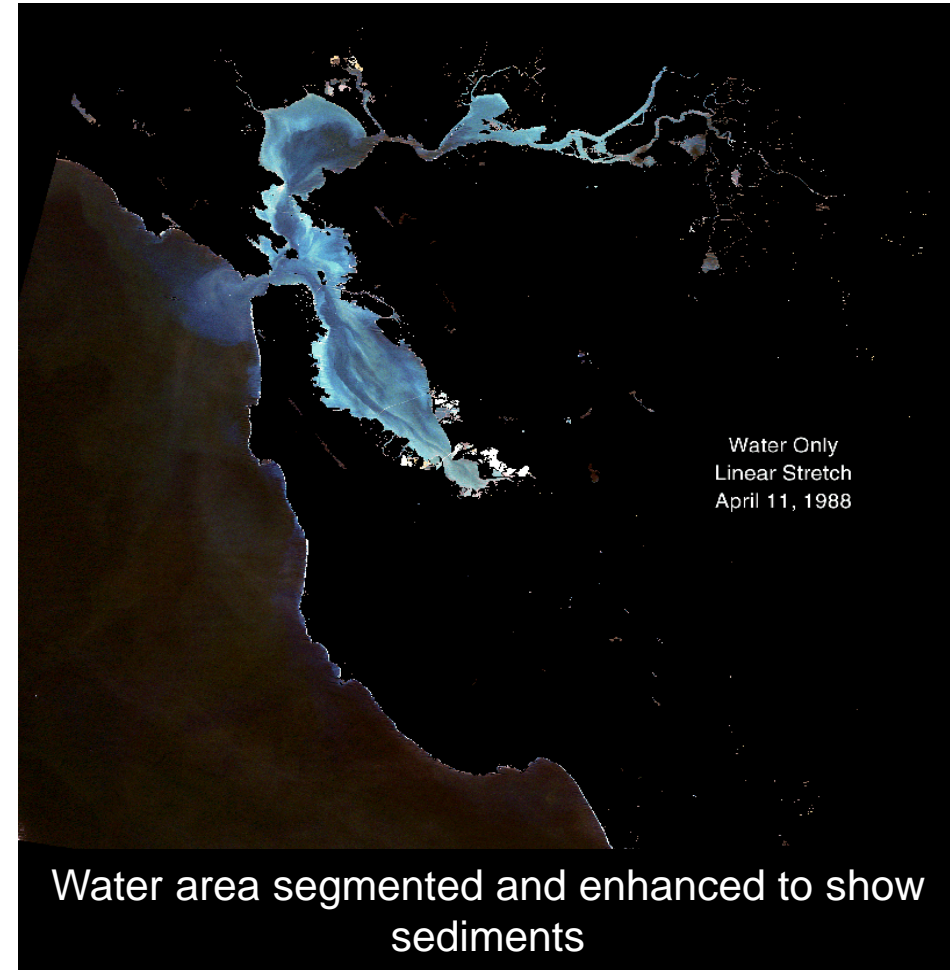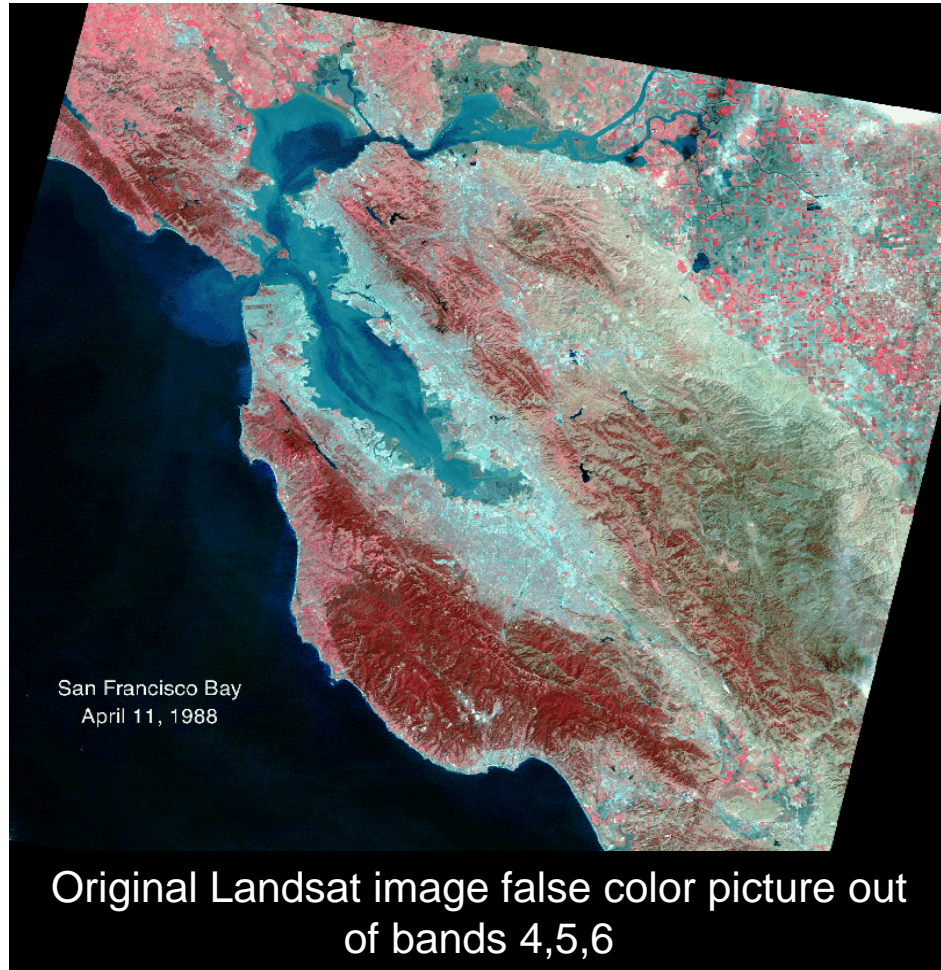
Region labeling and counting

- How many fish in this picture?
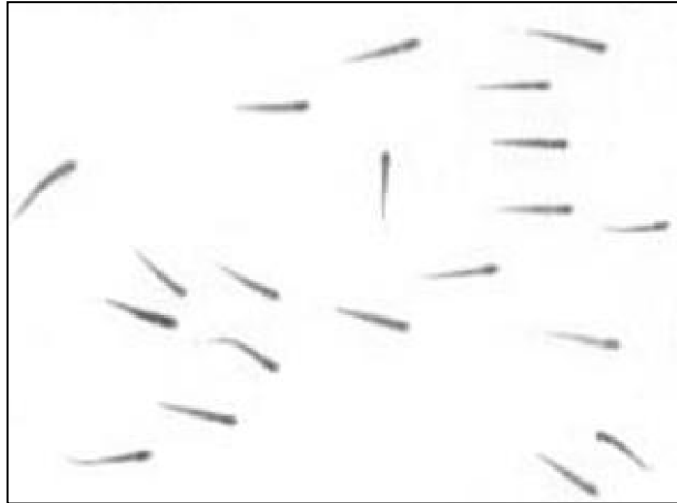
Original *Fish* image   after thresholding
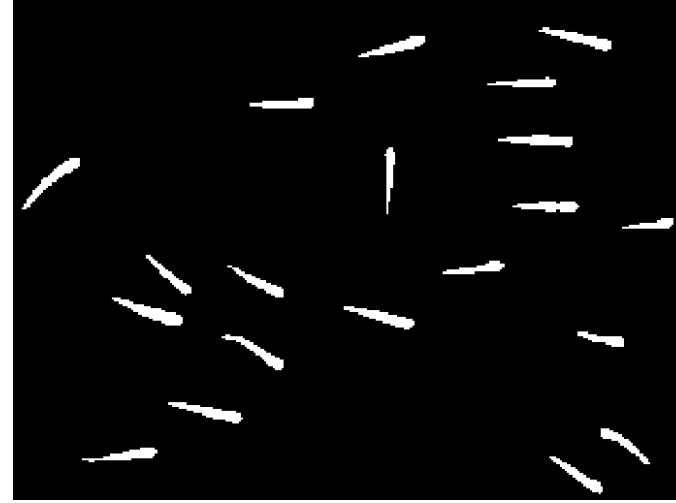
- Which pixels belong to the same object (region labeling)?
- How large is each object (region counting)?

4-connected and 8-connected neighborhoods

- <u>Definition:</u> a ***region*** is a set of pixels, where each pixel can be reached from any other pixel in the region by a finite number of steps, with each step starting at a pixel and ending in the neighborhood of the pixel.

4-neighborhood 8-neighborhood
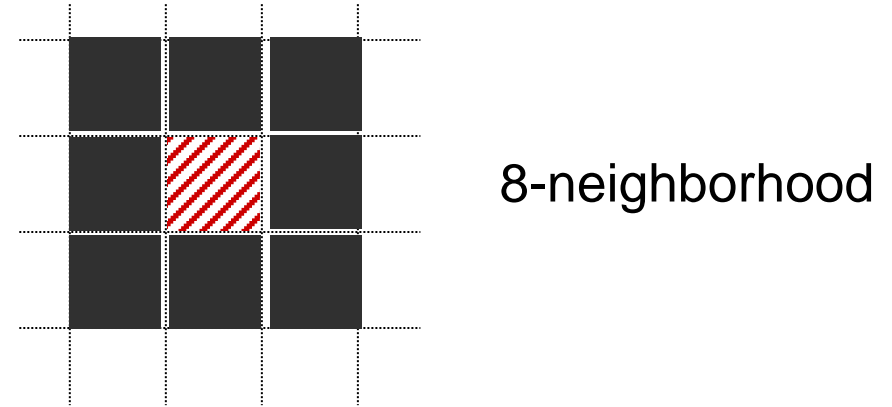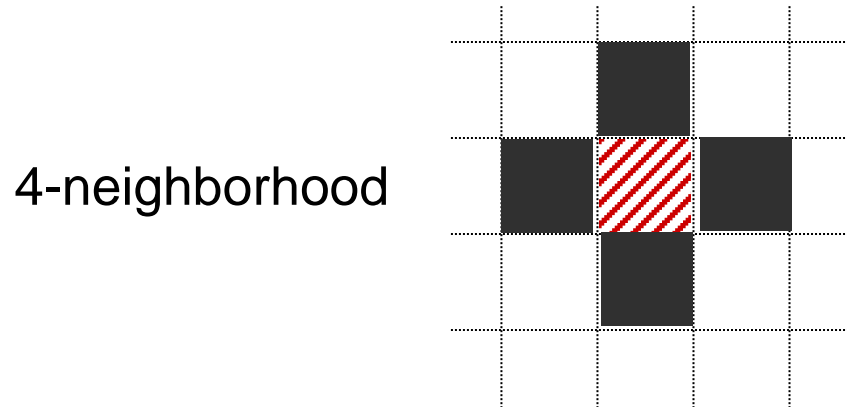
- Typically, either definition leads to the same regions, except when a region is only connected across diagonally adjacent pixels.

Region labeling algorithm (4-neighborhood)

- Loop through all pixels $f[x,y]$, left to right, top to bottom
- If $f[x,y]=0$, do nothing.
- If $f[x,y]=1$, distinguish 4 cases
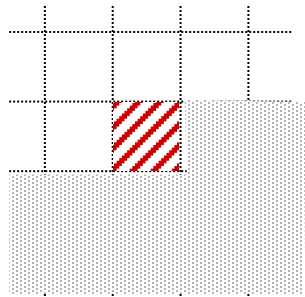
Generate new region label
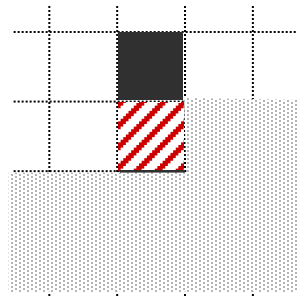
Copy label from above

Copy label from the left

Copy label from the left. If labels above and to the left are different, store equivalence.

- Second pass through image to replace equivalent label by the same label.

Region labeling example (4-neighborhood)

Region labeling example (4-neighborhood)
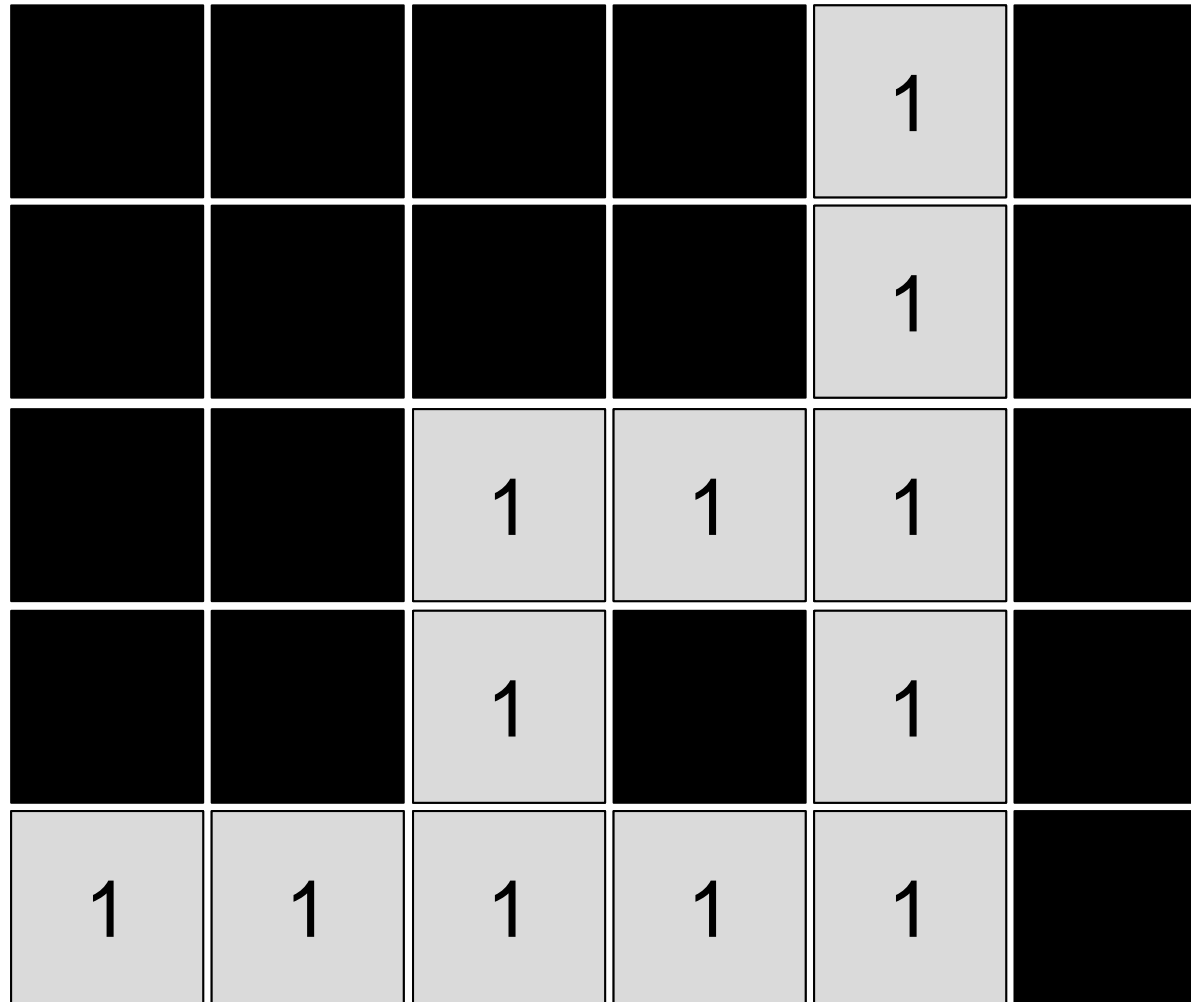
List of Region Labels

1

Example: region labeling
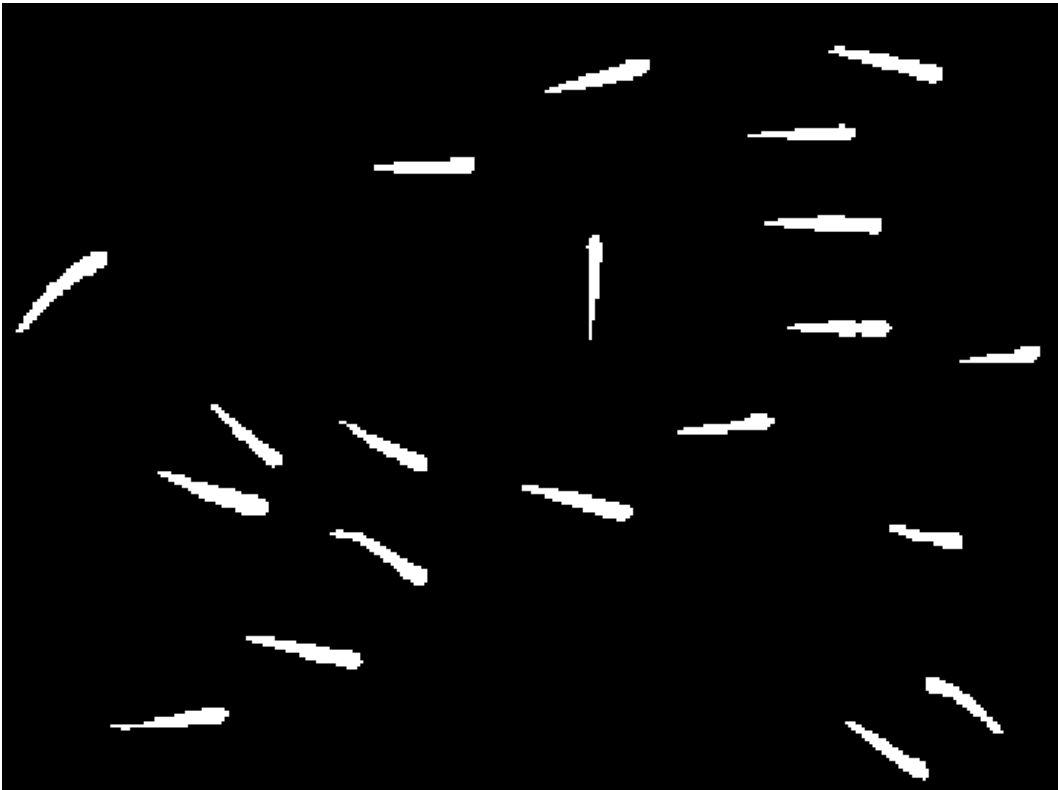
Thresholded image
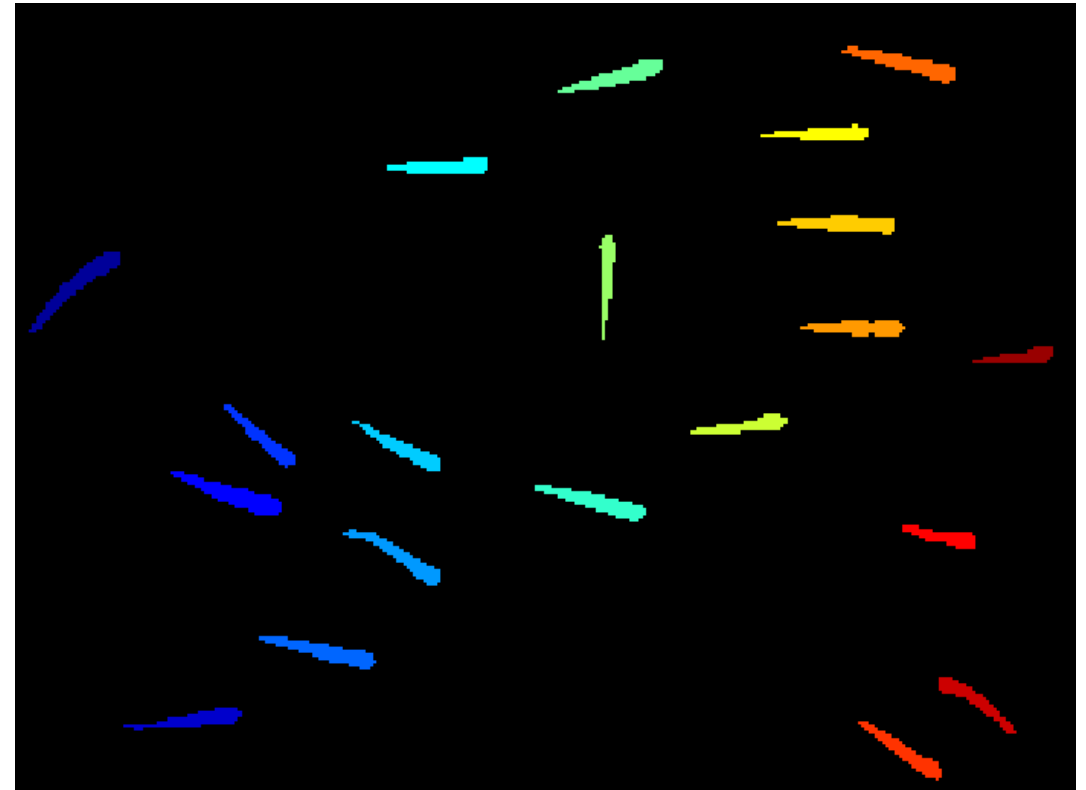
20 labeled regions

Region counting algorithm

- Measures the size of each region

- Initialize $counter[label]=0$ for all $label$
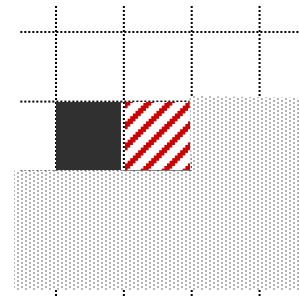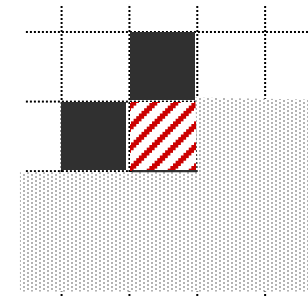
- Loop through all pixels $f[x,y]$, left to right, top to bottom

  - If $f[x,y]=0$, do nothing.

  - If $f[x,y]=1$, increment $counter[label[x,y]]$

Small region removal

- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$ and $counter[label[x,y]]<S$, set $f[x,y]=0$
- Removes all regions smaller than S pixels

Hole filling as dual to small region removal

Mask with holes

After NOT operation, (background)
region labeling, small region removal,
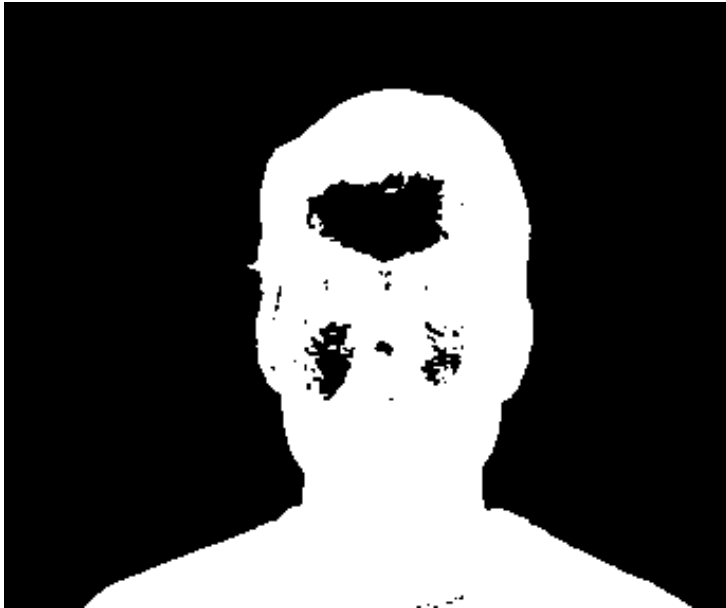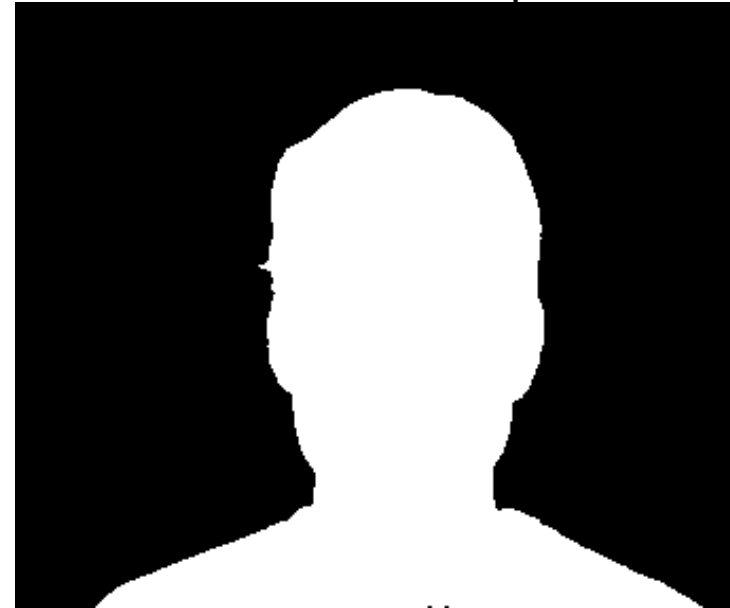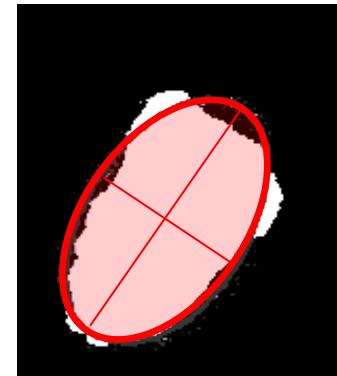and second NOT operation

Region moments

- Raw moments

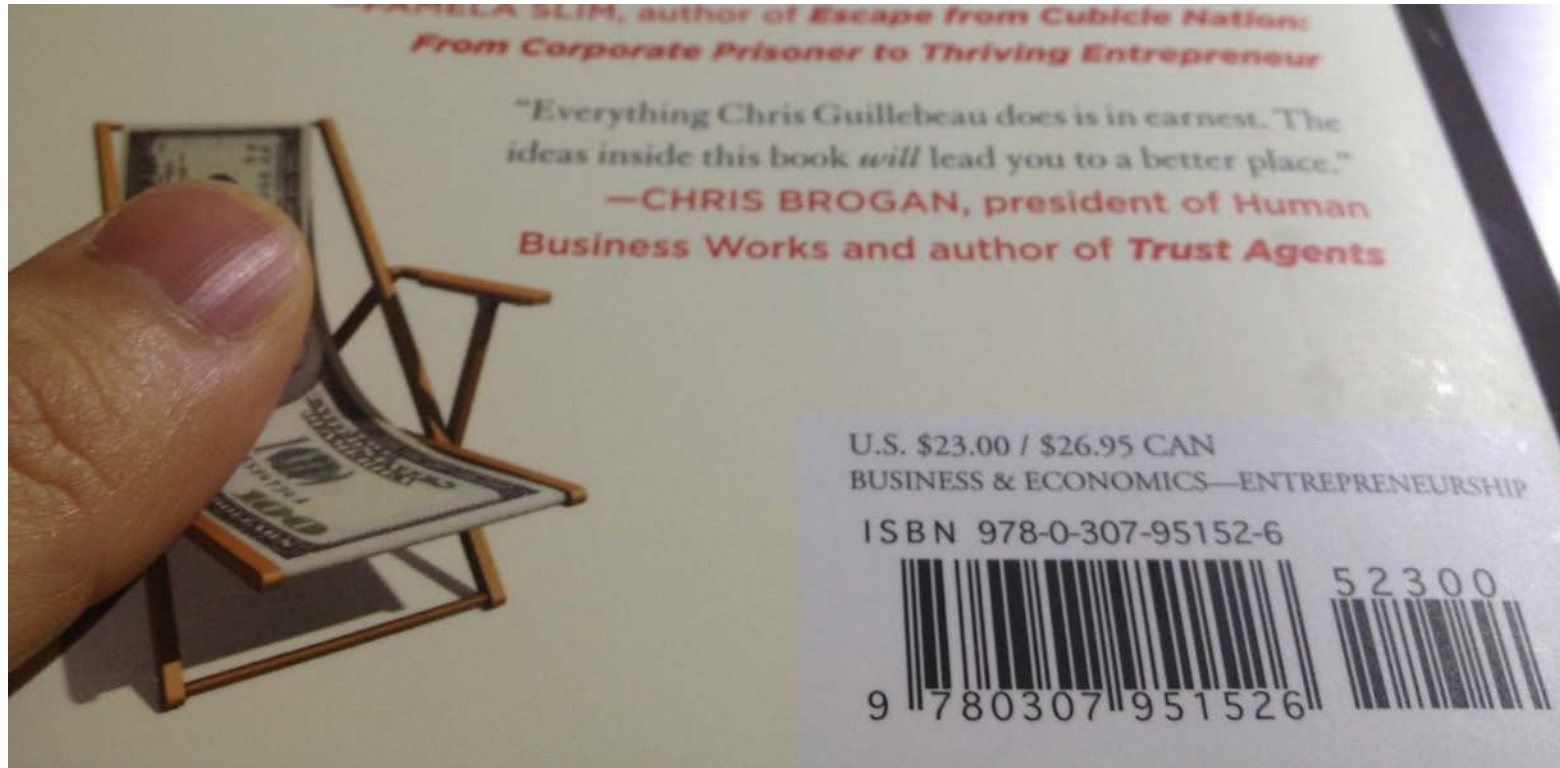$$M_{pq} = \sum_{x,y \in \text{Region}} x^p y^q$$

- Central moments

$$\mu_{pq} = \sum_{x,y \in \text{Region}} \left(x - \overline{x}\right)^p \left(y - \overline{y}\right)^q \quad \text{with } \overline{x} = \frac{M_{10}}{M_{00}} \text{ and } \overline{y} = \frac{M_{01}}{M_{00}}$$

- Region orientation and eccentricity: calculate eigenvectors of covariance matrix $\begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix}$
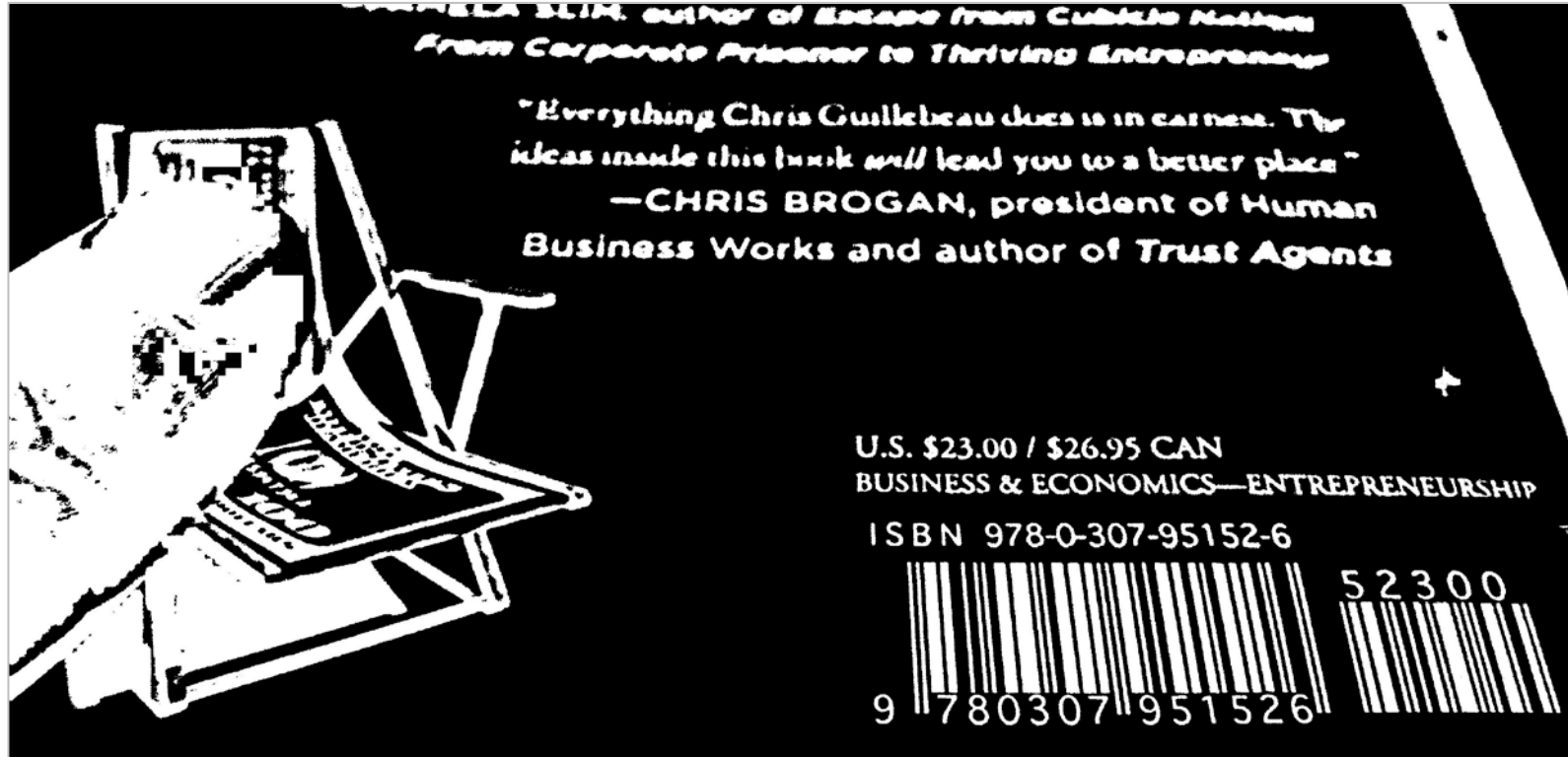
Example: Detecting bar codes

Original Image

Locally adaptive
thresholding

Example: Detecting bar codes
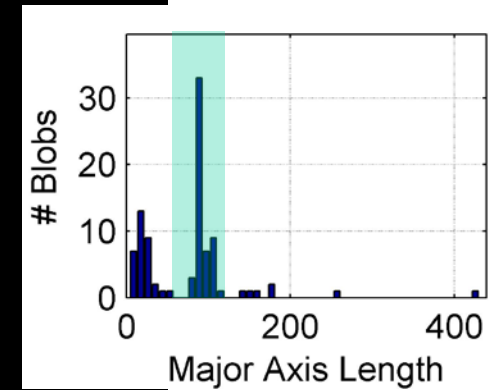
Locally adaptive thresholding

Filtering by eccentricity

Locally adaptive thresholding

Filtering by eccentricity

Filtering by major axis length

Locally adaptive thresholding

Filtering by eccentricity

Filtering by major axis length

Filtering by orientation