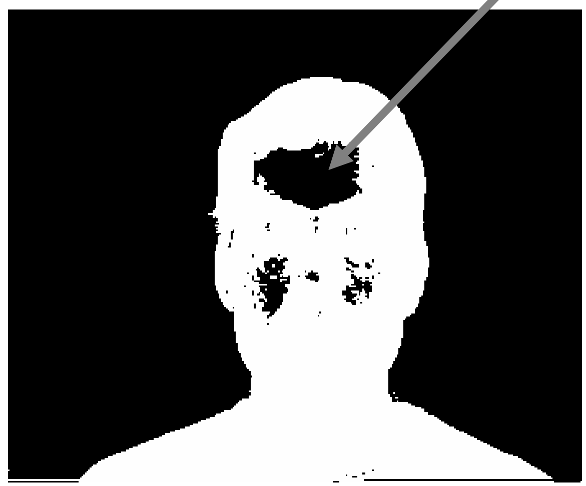
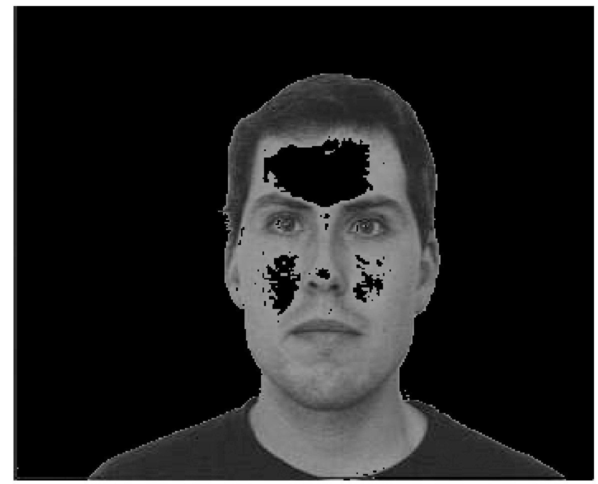
Luminance-based segmentation

How can holes be filled?

Original image
Peter $f(x,y)$

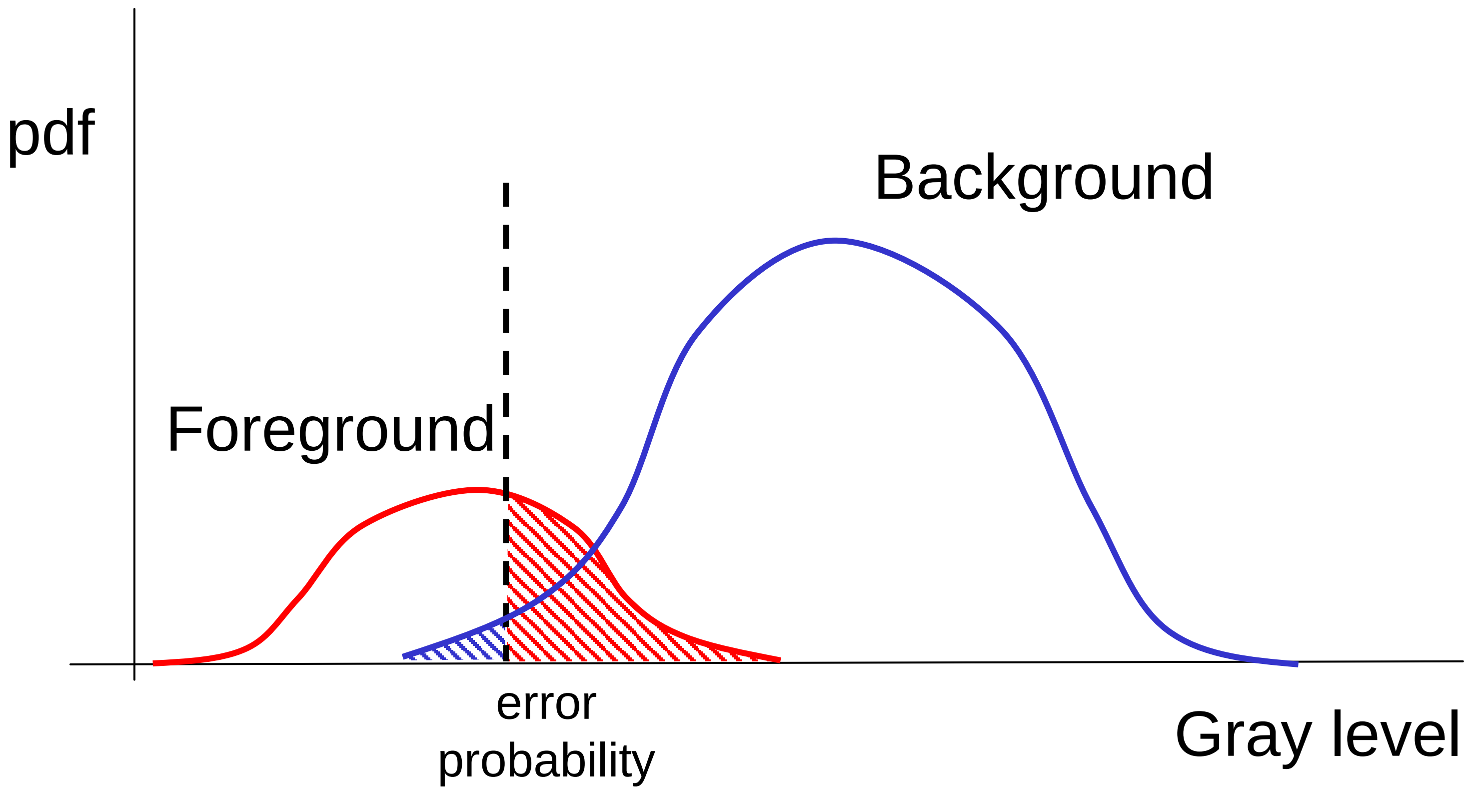


Thresholded
Peter $m(x,y)$

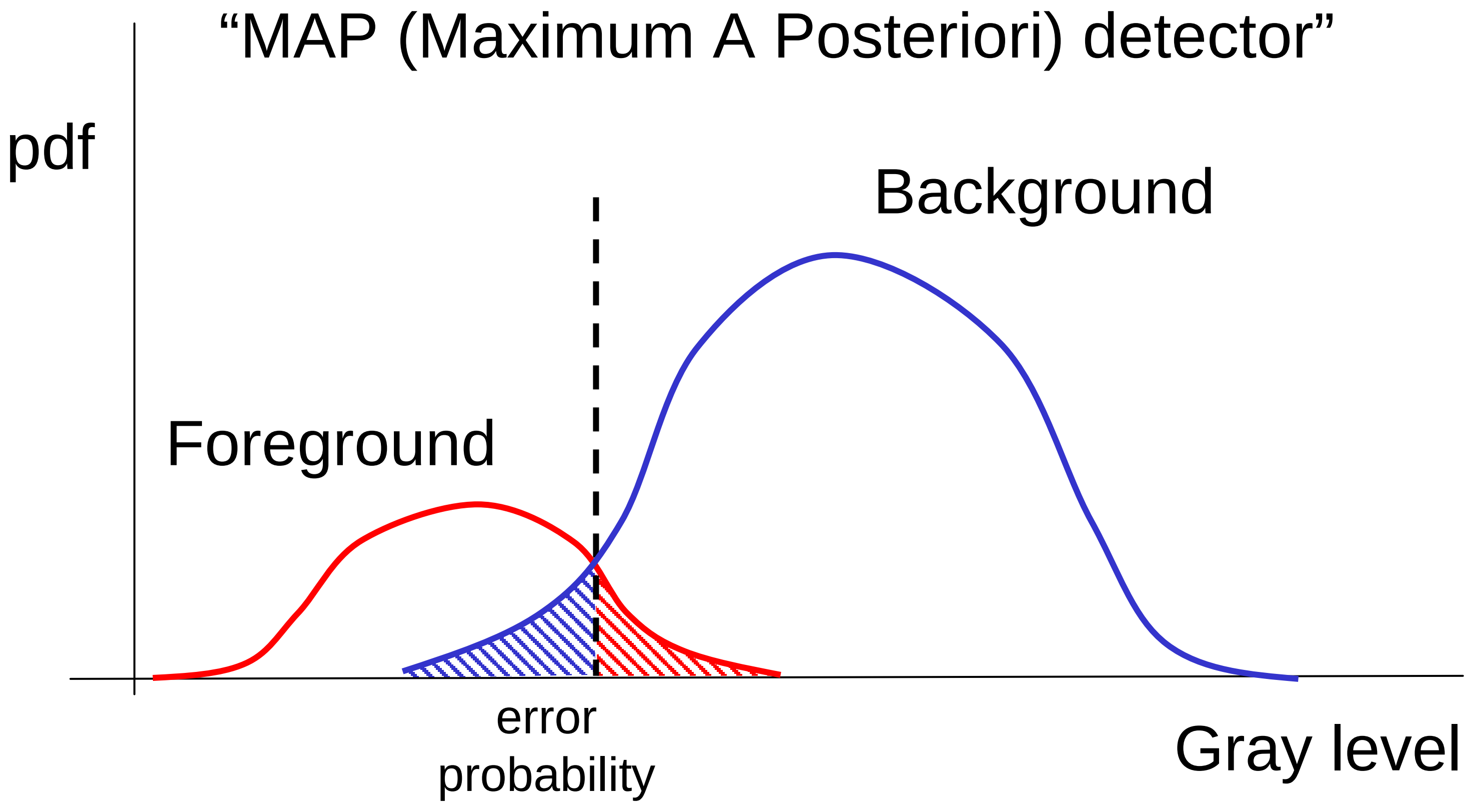


$const. \cdot f(x,y) \cdot m(x,y)$

Error probability for thresholding



Optimal supervised thresholding



If different outcomes are associated with different costs: more general "Bayes minimimum risk detector"

Unsupervised thresholding

- Idea: find threshold T that minimizes *within-class variance* of both foreground and background

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

- Equivalently, maximize *between-class variance*

$$\begin{aligned}\sigma^2_{between}(T) &= \sigma^2 - \sigma^2_{within}(T) \\ &= \left(\sum_{x,y} I^2(x,y) - \mu^2\right) - \frac{N_{Fgrd}}{N}\left(\sum_{x,y\in Fgrnd} I^2(x,y) - \mu^2_{Fgrnd}\right) - \frac{N_{Bgrnd}}{N}\left(\sum_{x,y\in Bgrnd} I^2(x,y) - \mu^2_{Bgrnd}\right) \\ &= -\mu^2 + \frac{N_{Fgrnd}}{N}\mu^2_{Fgrnd} + \frac{N_{Bgrnd}}{N}\mu^2_{Bgrnd} = \frac{N_{Fgrnd}}{N}\left(\mu_{Fgrnd} - \mu\right)^2 + \frac{N_{Bgrnd}}{N}\left(\mu_{Bgrnd} - \mu\right)^2 \\ &= \frac{N_{Fgrnd}(T)\cdot N_{Bgrnd}(T)}{N^2}\left(\mu_{Fgrnd}(T) - \mu_{Bgrnd}(T)\right)^2\end{aligned}$$

[Otsu, 1979]

Unsupervised thresholding (cont.)

- Algorithm: Search for threshold T to maximize

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

- Efficient recursive computation:

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

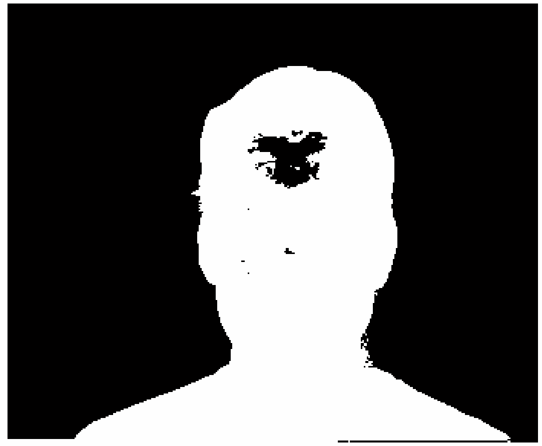
$$N_{Bgrnd}(T+1) = N_{Bgrnd}(T) - n_T$$

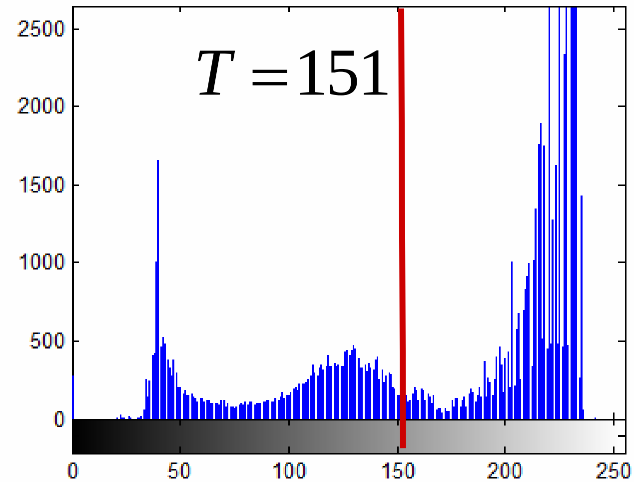
$$\mu_{Fgrnd}(T+1) = \frac{\mu_{Fgrnd}(T)\,N_{Fgrnd}(T) + n_T T}{N_{Fgrnd}(T+1)}$$

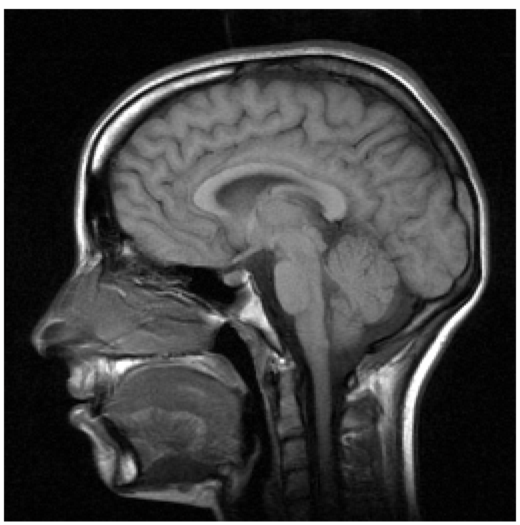
$$\mu_{Bgrnd}(T+1) = \frac{\mu_{Bgrnd}(T)\,N_{Bgrnd}(T) - n_T T}{N_{Fgrnd}(T+1)}$$

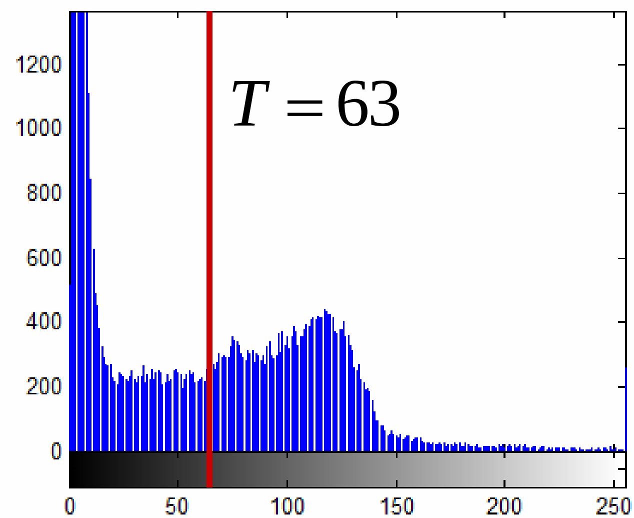
[Otsu, 1979]

Unsupervised thresholding (cont.)



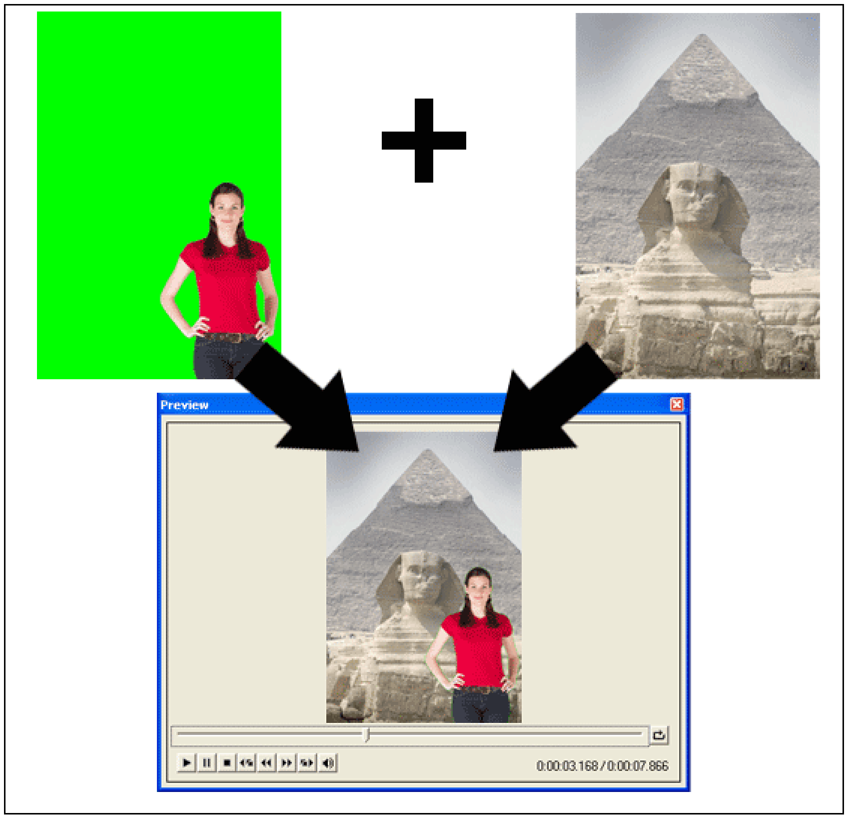


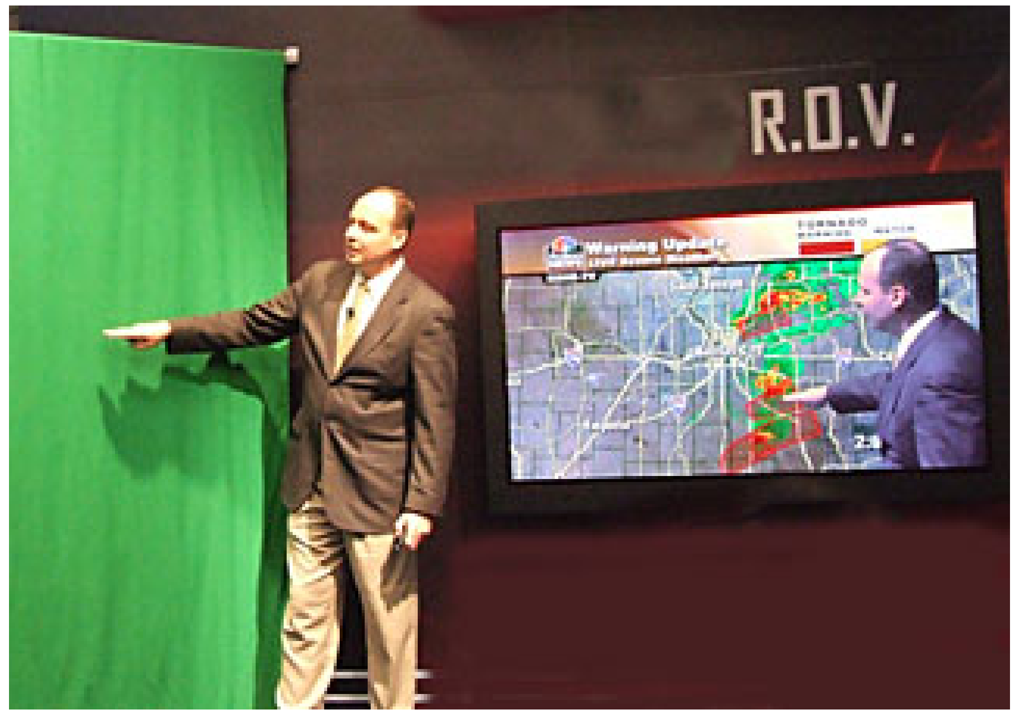




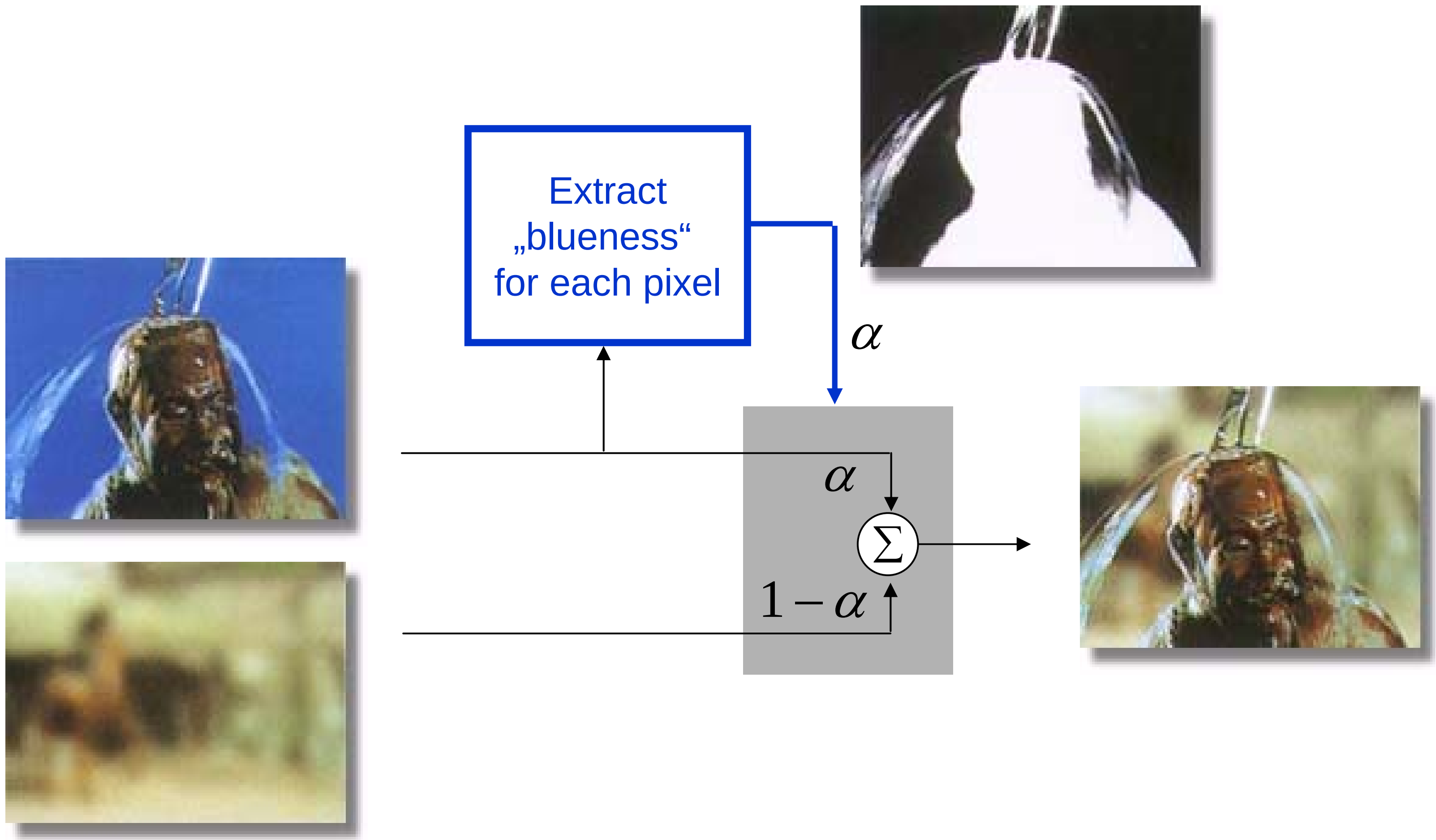
Chroma keying

- Color is more powerful for pixel-wise segmentation: 3-d vs. 1-d space
- Take picture in front of a blue screen (or green, or orange)

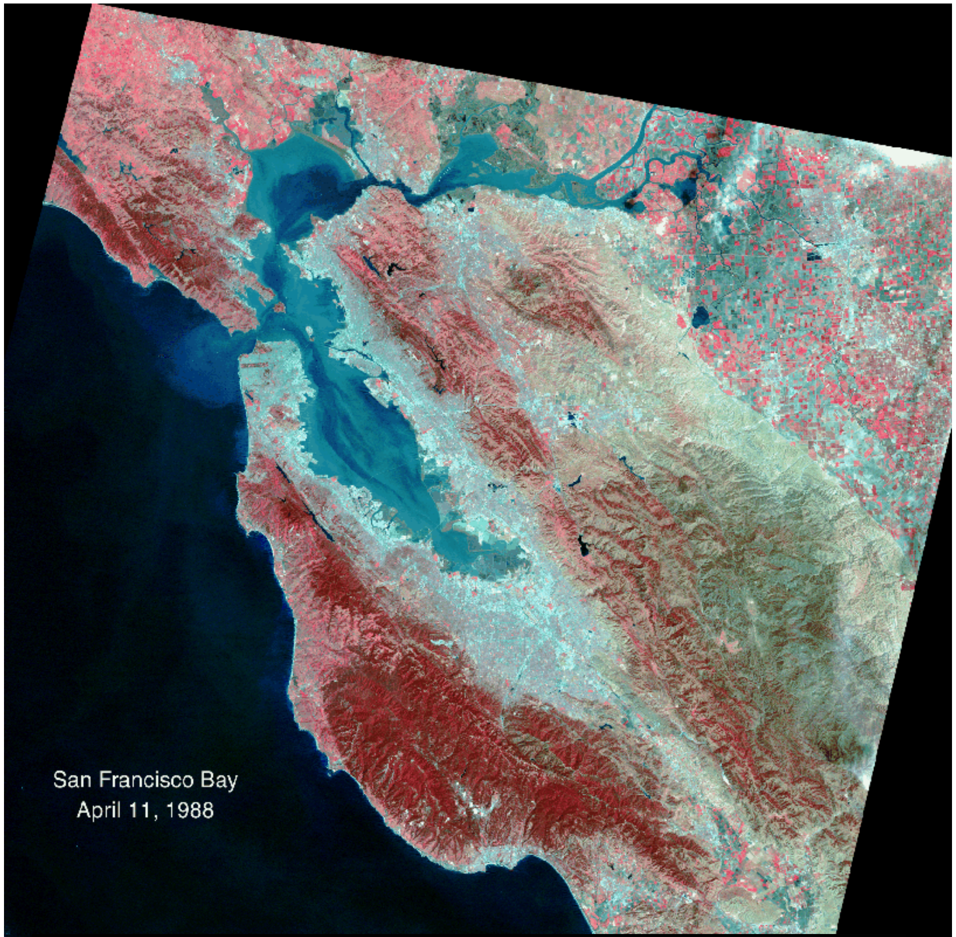




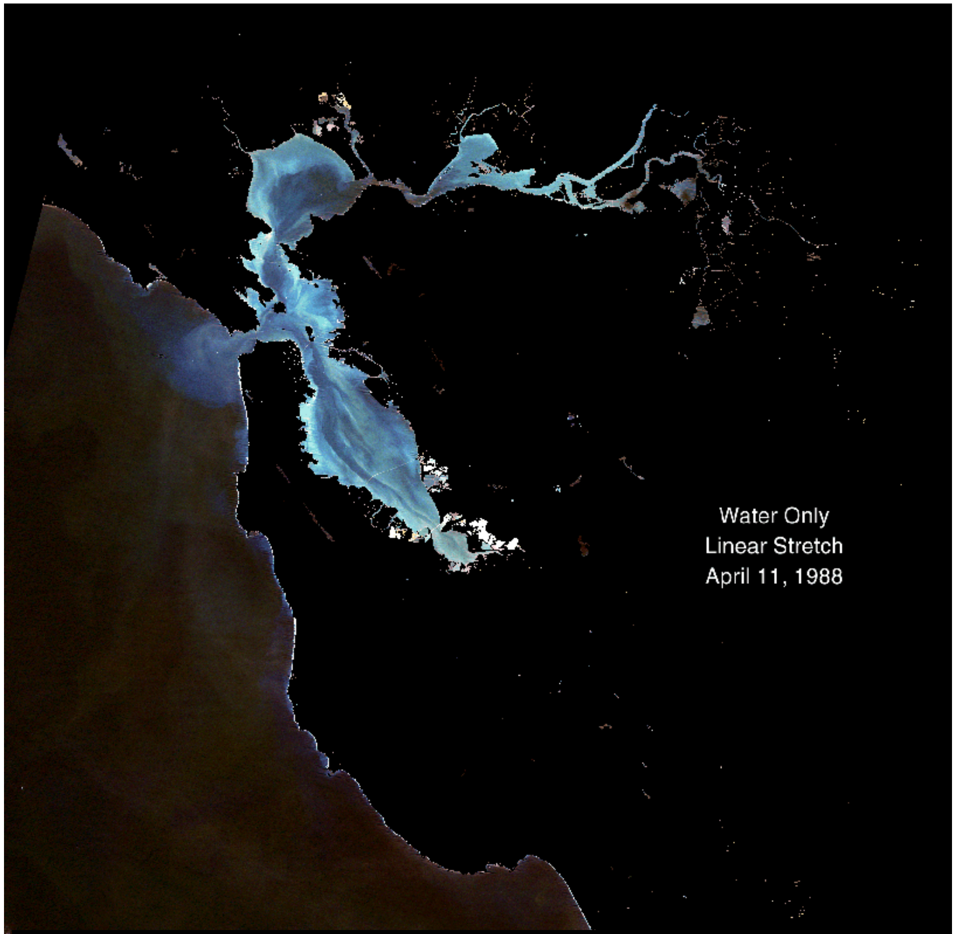
Soft chroma keying



Landsat image processing



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

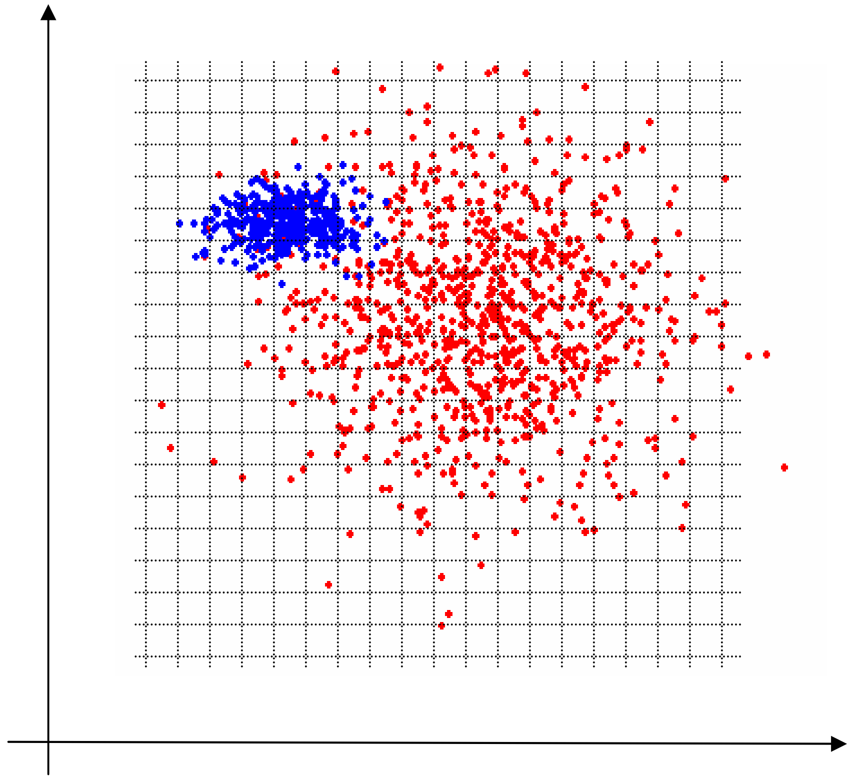


Water area segmented and enhanced to show sediments

Source: US Geological Survey USGS, http://sfbay.wr.usgs.gov/

Multidimensional MAP detector

- Label categories in training set by hand
- Subdivide n-dimensional space into small bins
- Count frequency of occurrence for each bin and class in training set
- For test data: identify bin, detect the more probable category



MAP detector in RGB-space

Original image

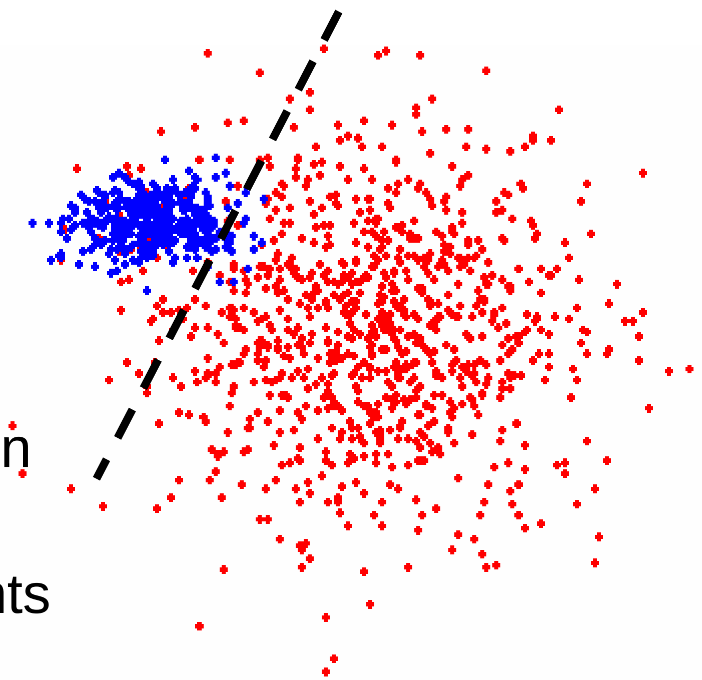
Skin color detector

[S. Leahy, EE368 class project, 2003]

Linear discriminant function

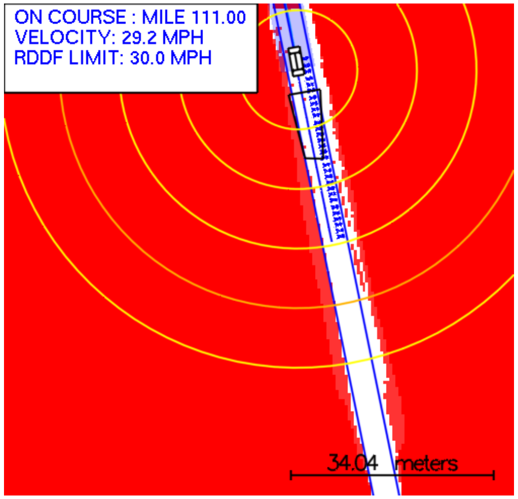
- To segment image with n components f_i, $i=1,2,\ldots,n$ into two categories, perform test

$$\boxed{\sum_i w_i f_i + w_0 \geq 0 \ \ ?}$$



- 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 discriminant functions
- Can be extended to multiple categories

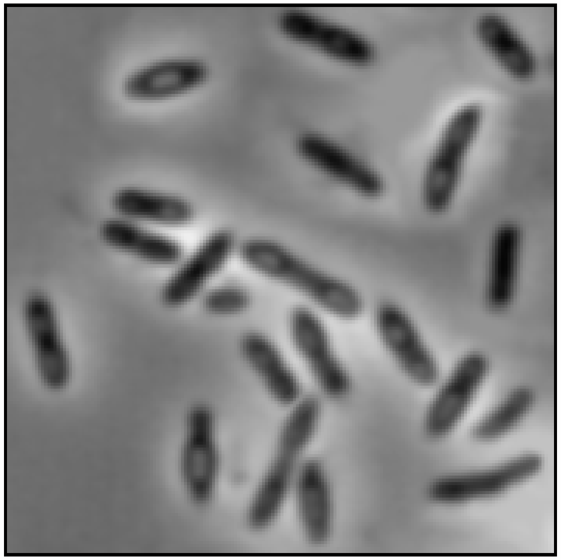
Self-Supervised Road Detection



[courtesy: H. Dahlkamp, S. Thrun, 2005]

Region labeling and counting

- How many bacteria in this picture?



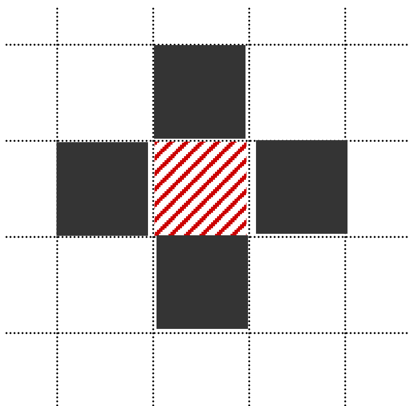
Original *Bacteria* image

after thresholding

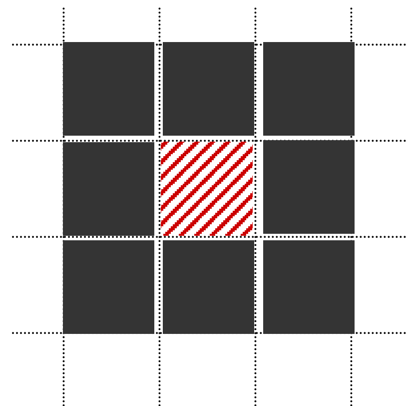
- Which pixels belong to the same object (region labeling)?
- How large is each object (region counting)?

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, where each step starts at a pixel and ends in the neighborhood of the pixel.



4-neighborhood

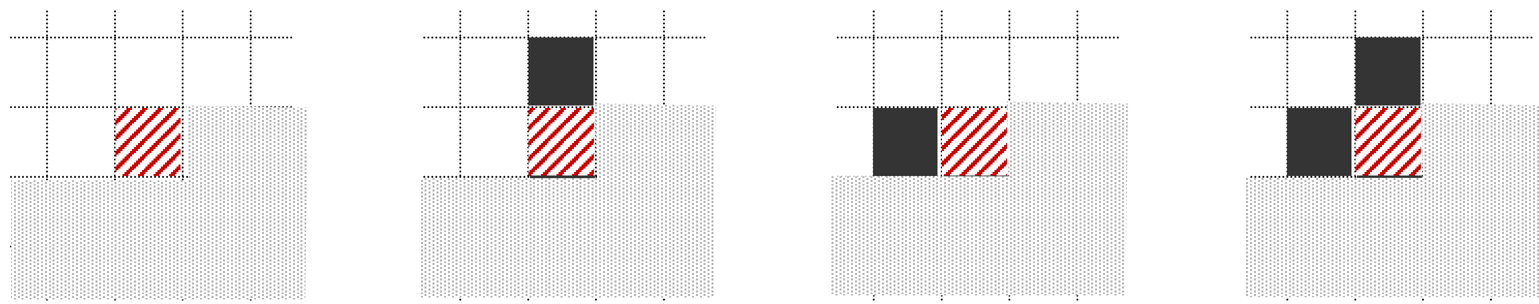


8-neighborhood

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

Region labeling algorithm (4-neighborhood)

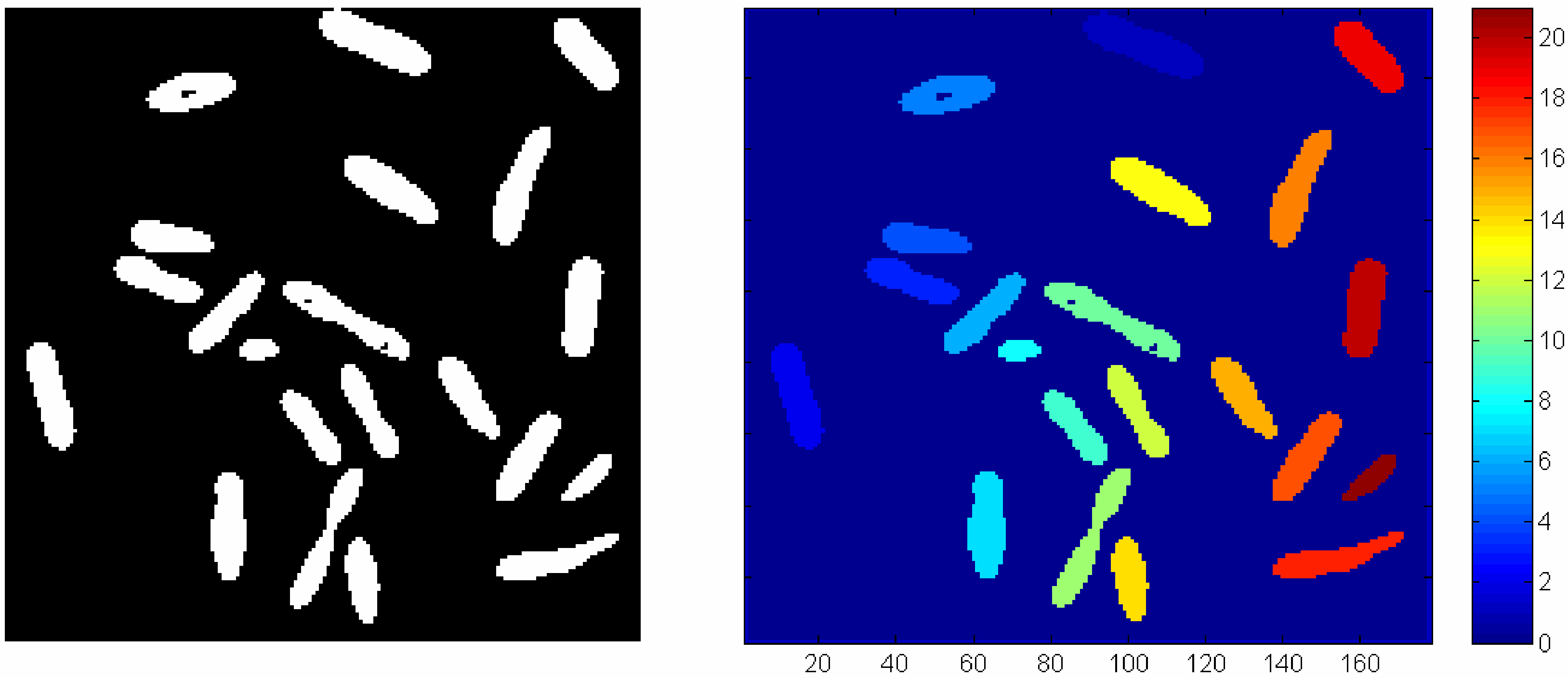
- Loop through all pixels $f(x,y)$, left to right, top to bottom
- If $f(x,y)=0$, do nothing.
- If $f(x,y)=1$, distinguish 4 cases



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.

Example: region labeling



Region counting algorithm

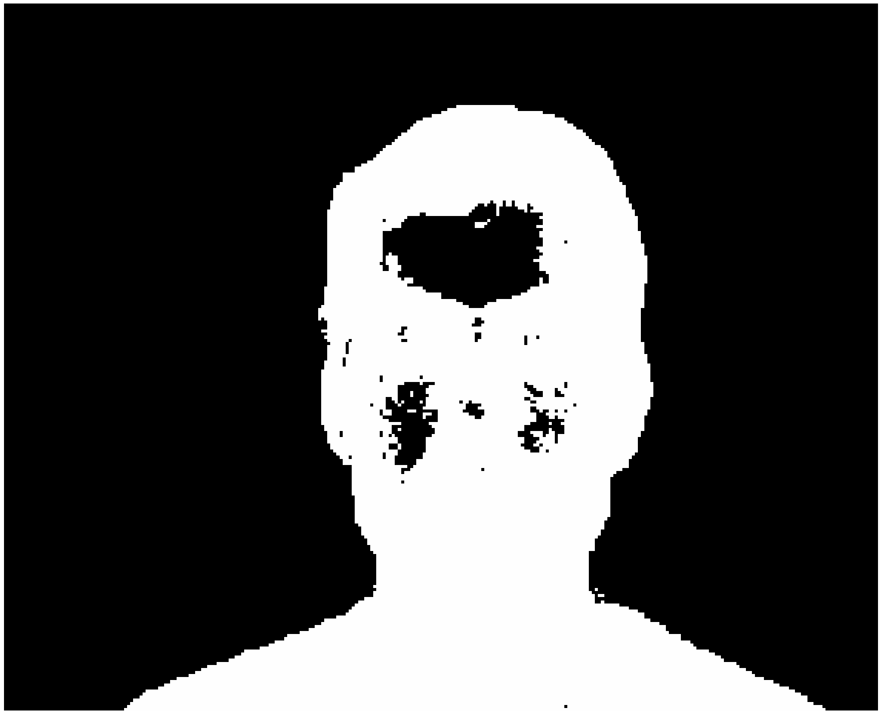
- Measures the size of each region
- Initialize $counter(label)=0$ for all $label$
- Loop through all pixels $f(x,y)$, left to right, top to bottom
  - If $f(x,y)=0$, do nothing.
  - If $f(x,y)=1$, increment $counter(label(x,y))$

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 $counter(label(x,y))<S$, then 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