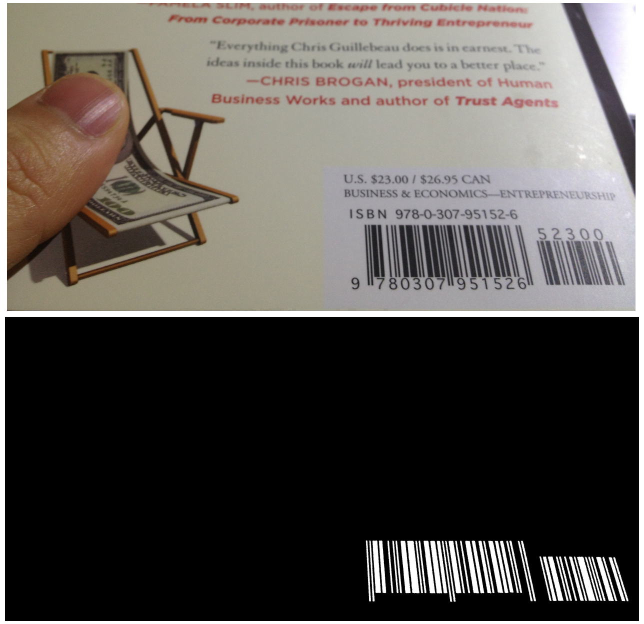
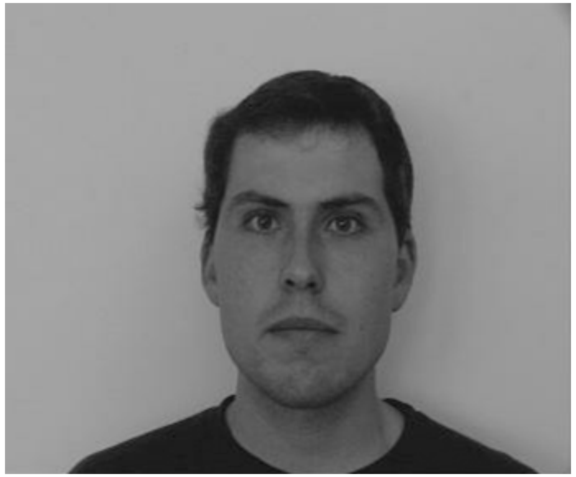
Image Segmentation

- 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

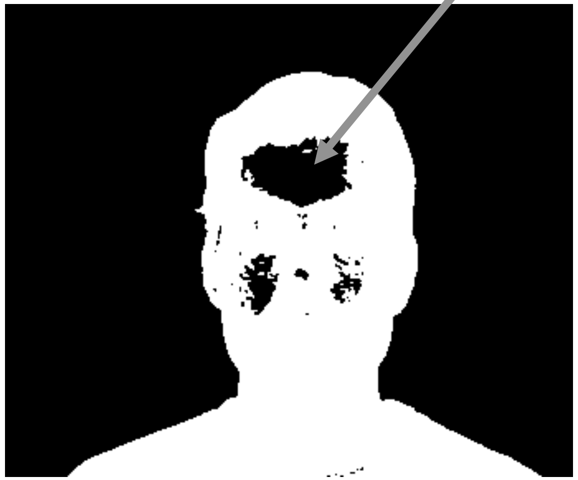


Gray-level thresholding

How can holes be filled?



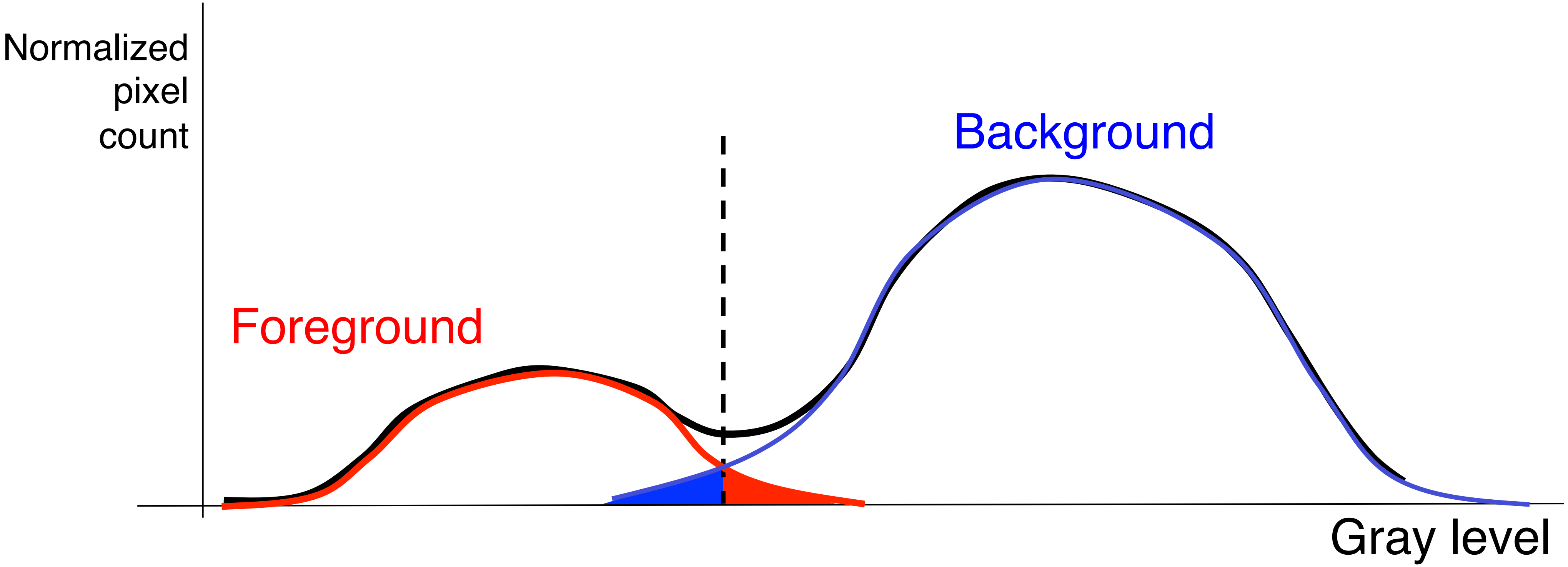
Original image
Peter $f[x,y]$



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)

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

- Equivalently, maximize *between-class variance*

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

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

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

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

[Otsu, 1979]

Unsupervised thresholding (cont.)

- Algorithm: Search for threshold T to maximize

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

- Useful recursion for sweeping T across histogram:

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

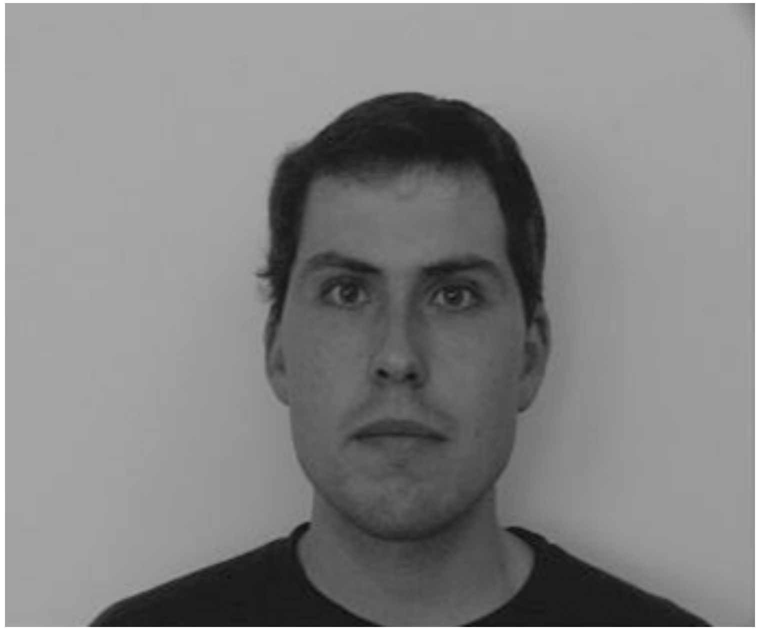
$$N_{Bgrnd}(T+1) = N_{Bgrnd}(T) - n_T$$

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

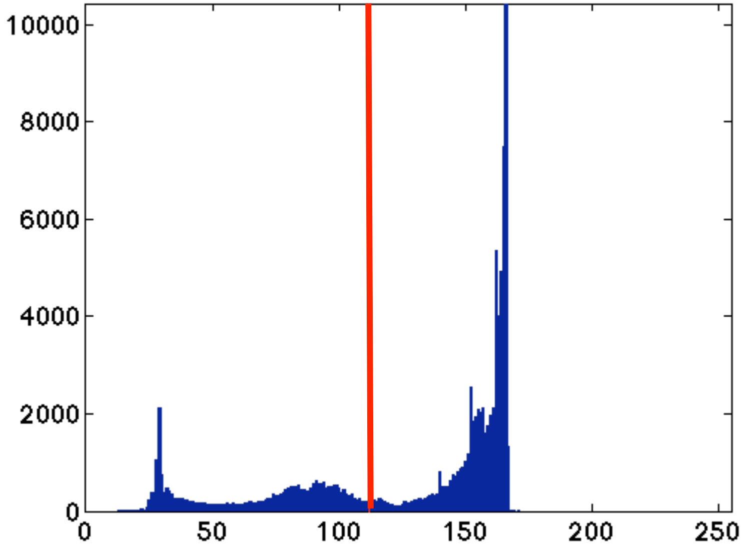
$$\mu_{Bgrnd}(T+1) = \frac{\mu_{Bgrnd}(T) N_{Bgrnd}(T) - n_T T}{N_{Fgrnd}(T+1)}$$

[Otsu, 1979]

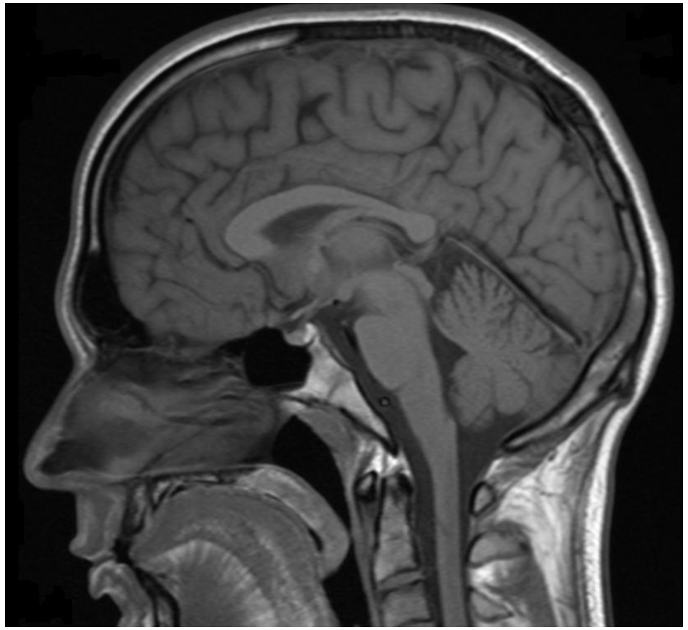
Unsupervised thresholding (cont.)

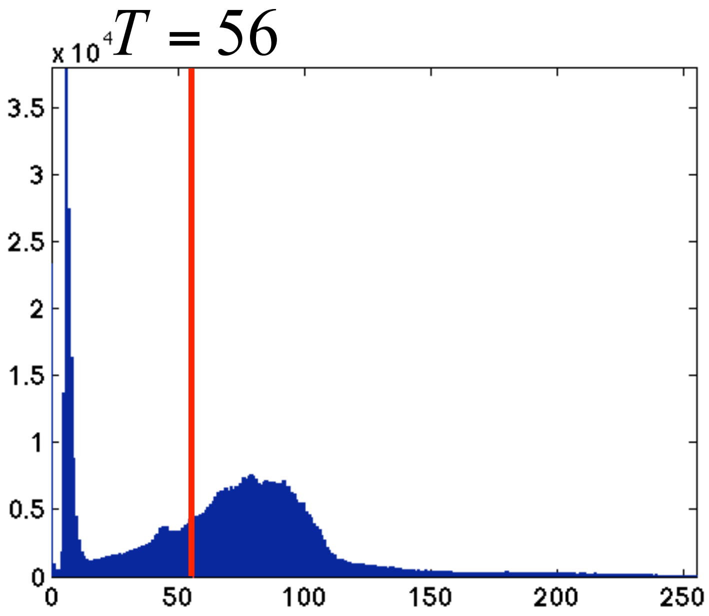


$T = 112$



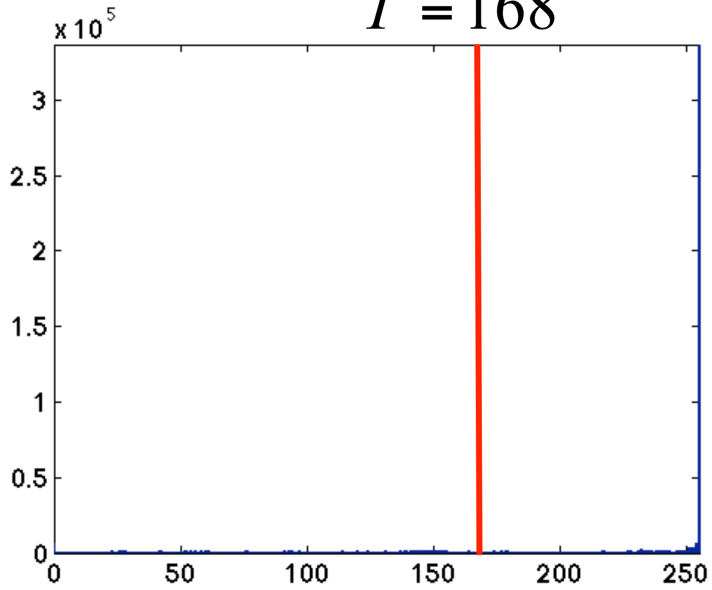
Unsupervised thresholding (cont.)



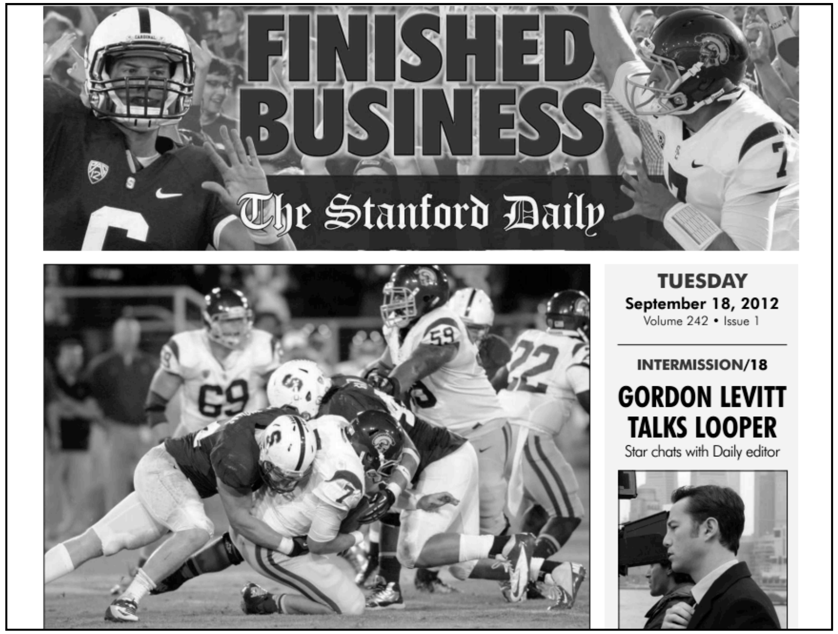


Unsupervised thresholding (cont.)

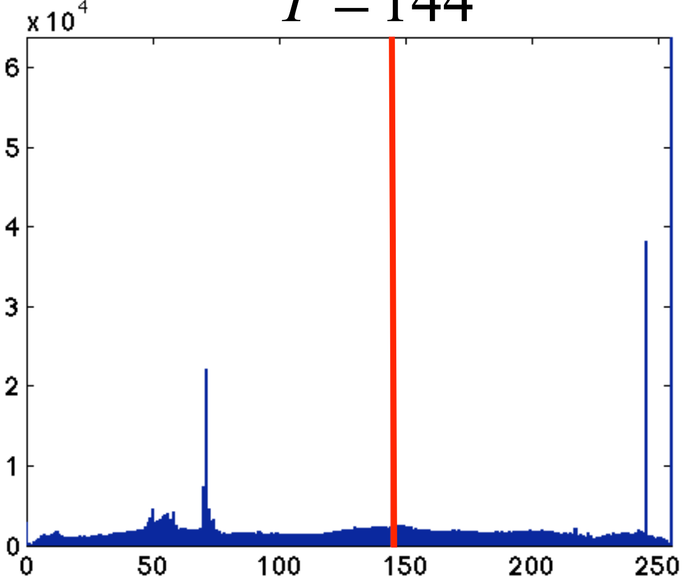
$T = 168$

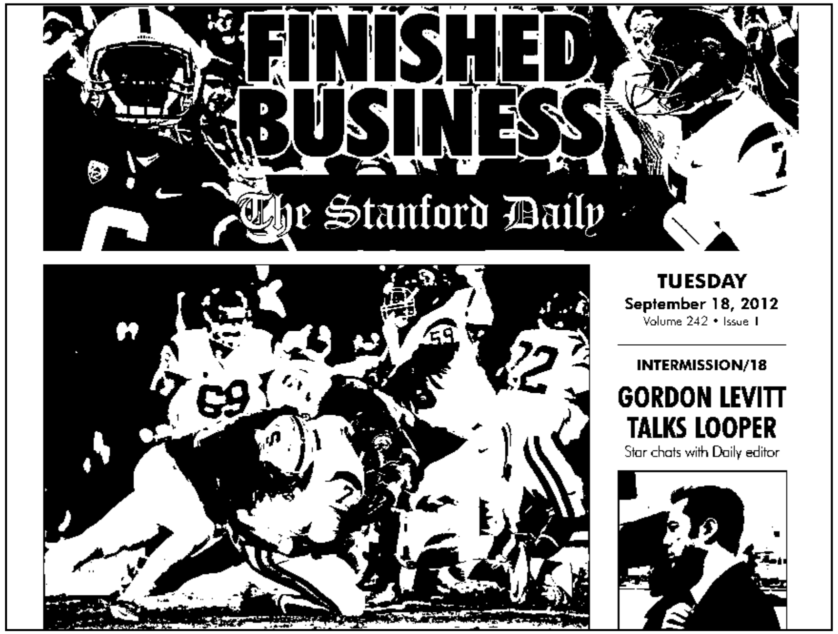


Unsupervised thresholding (cont.)



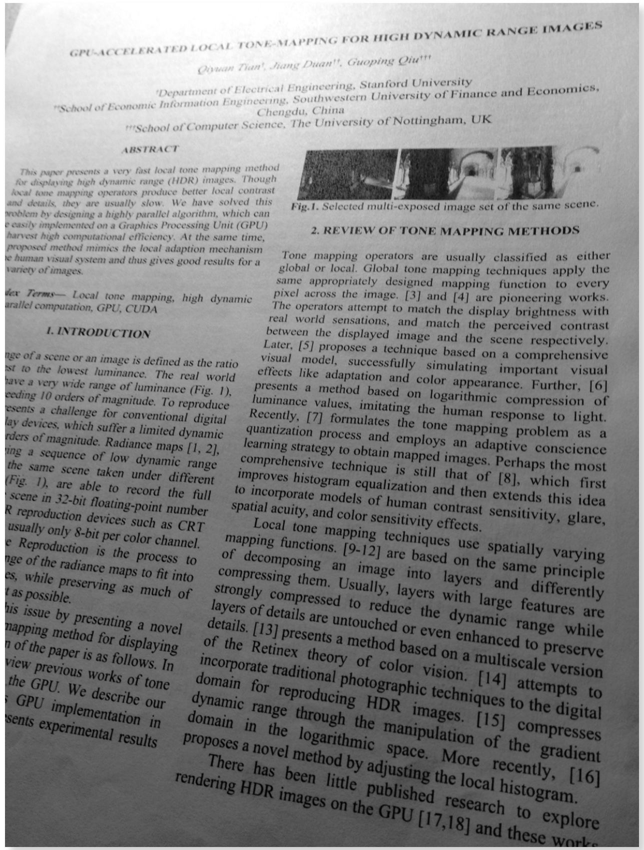
$T = 144$





Sometimes, a global threshold does not work

Original image



GPU-ACCELERATED LOCAL TONE-MAPPING FOR HIGH DYNAMIC RANGE IMAGES

Qiyuan Tian[1], Jiang Duan[1,2], Guoping Qiu[2,3]

[1]Department of Electrical Engineering, Stanford University
[2]School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, China
[3]School of Computer Science, The University of Nottingham, UK

ABSTRACT

This paper presents a very fast local tone mapping method for displaying high dynamic range (HDR) images. Though local tone mapping operators produce better local contrast and details, they are usually slow. We have solved this problem by designing a highly parallel algorithm, which can be easily implemented on a Graphics Processing Unit (GPU) to harvest high computational efficiency. At the same time, the proposed method mimics the local adaptation mechanism of the human visual system and thus gives good results for a variety of images.

Index Terms— Local tone mapping, high dynamic range, parallel computation, GPU, CUDA

1. INTRODUCTION

Fig.1. Selected multi-exposed image set of the same scene.

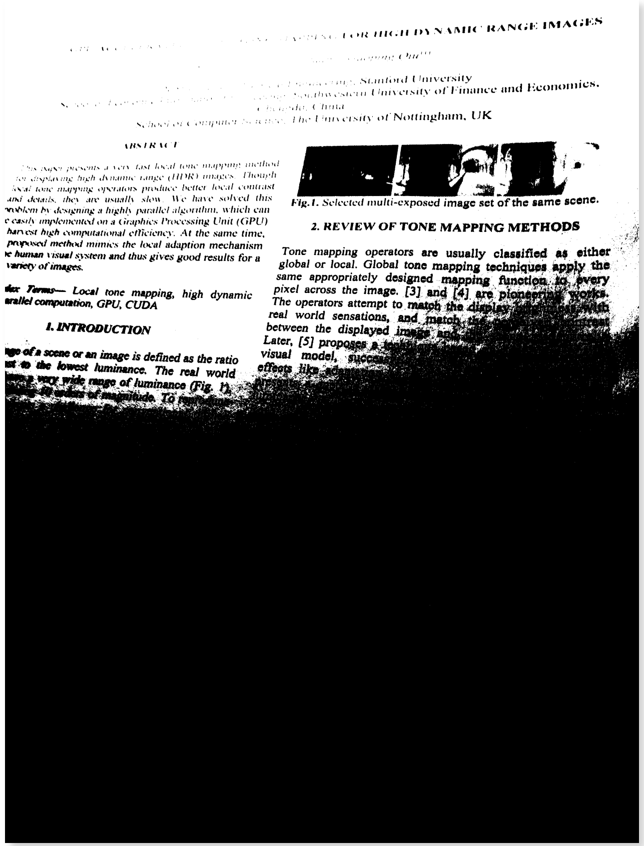
2. REVIEW OF TONE MAPPING METHODS

Tone mapping operators are usually classified as either global or local. Global tone mapping techniques apply the same appropriately designed mapping function to every pixel across the image. [3] and [4] are pioneering works. The operators attempt to match the display brightness with real world sensations, and match the perceived contrast between the displayed image and the scene respectively. Later, [5] proposes a technique based on a comprehensive visual model, successfully simulating important visual effects like adaptation and color appearance. Further, [6] presents a method based on logarithmic compression of luminance values, imitating the human response to light. Recently, [7] formulates the tone mapping problem as a quantization process and employs an adaptive conscience learning strategy to obtain mapped images. Perhaps the most comprehensive technique is still that of [8], which first improves histogram equalization and then extends this idea to incorporate models of human contrast sensitivity, glare, spatial acuity, and color sensitivity effects.

Local tone mapping techniques use spatially varying mapping functions. [9-12] are based on the same principle of decomposing an image into layers and differently compressing them. Usually, layers with large features are strongly compressed to reduce the dynamic range while layers of details are untouched or even enhanced to preserve details. [13] presents a method based on a multiscale version of the Retinex theory of color vision. [14] attempts to incorporate traditional photographic techniques to the digital domain for reproducing HDR images. [15] compresses dynamic range through the manipulation of the gradient domain in the logarithmic space. More recently, [16] proposes a novel method by adjusting the local histogram.

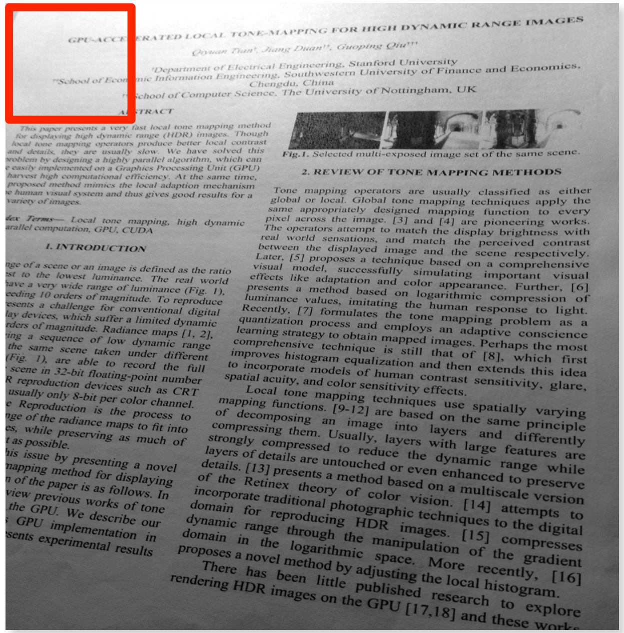
There has been little published research to explore rendering HDR images on the GPU [17,18] and these works

Thresholded with Otsu's Method

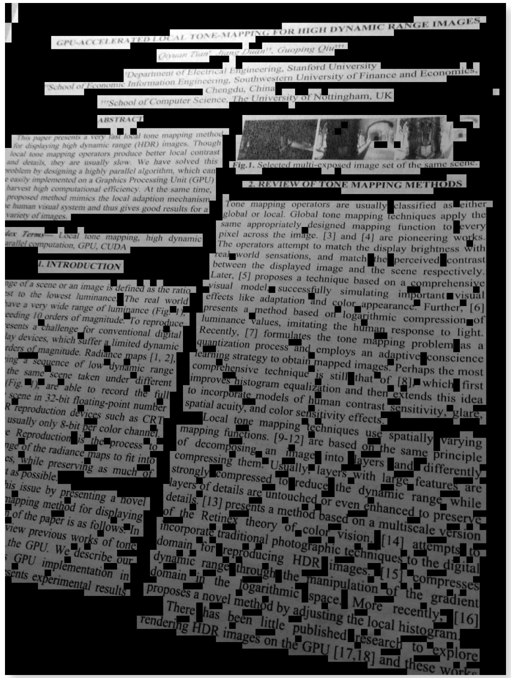


Locally adaptive thresholding

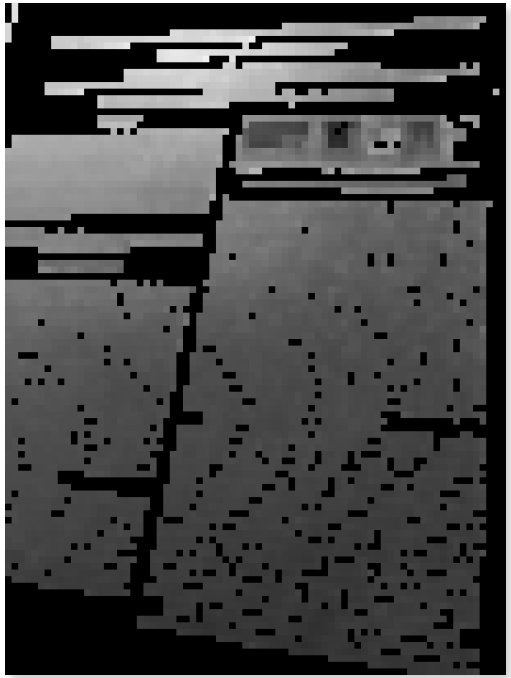
- Slide a window over the image
- For each window position, decide whether to perform thresholding
  - Thresholding should not be performed in uniform areas
  - Use variance or other suitable criterion
- Non-uniform areas: apply Otsu's method (based on <u>local</u> histogram)
- Uniform areas: classify the entire area as foreground or background based on mean value



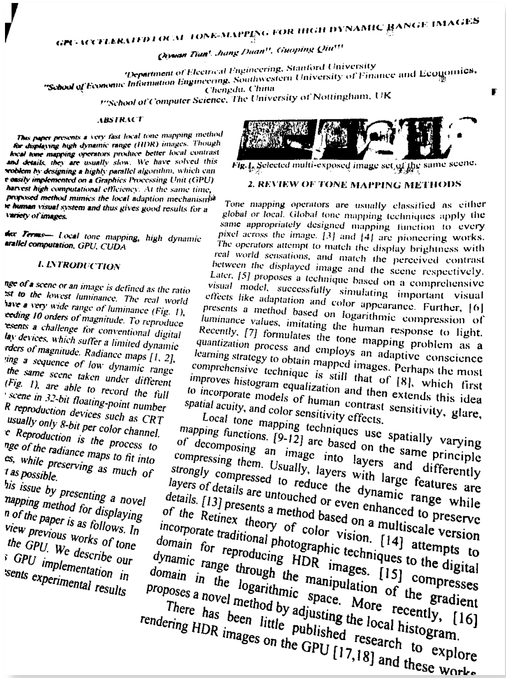
Locally adaptive thresholding (example)



Non-uniform areas



Local threshold values



Locally thresholded result

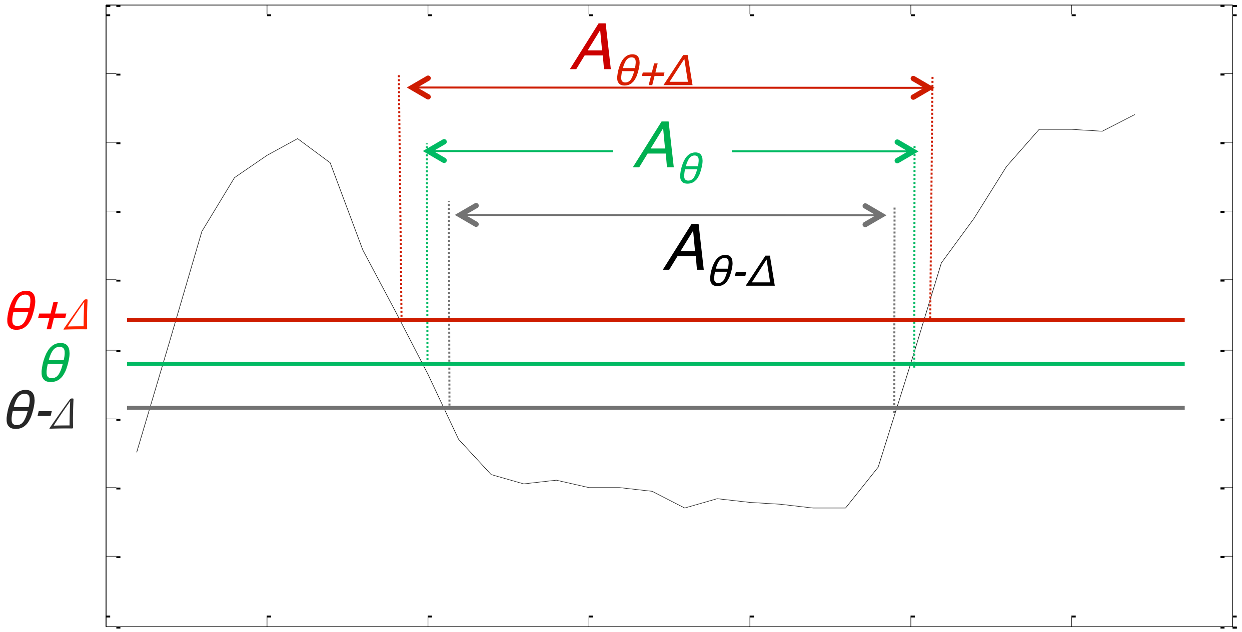
Maximally stable extremal regions

- 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 $+/-\Delta$

→ ***"maximally stable" extremal regions (MSER)***

[Matas, Chum, Urba, Pajdla, 2002]

MSERs: illustration



$$\text{Local minimum of } \left| \frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_{\theta}} \right| \rightarrow \text{MSER}$$

[Matas, Chum, Urba, Pajdla, 2002]

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

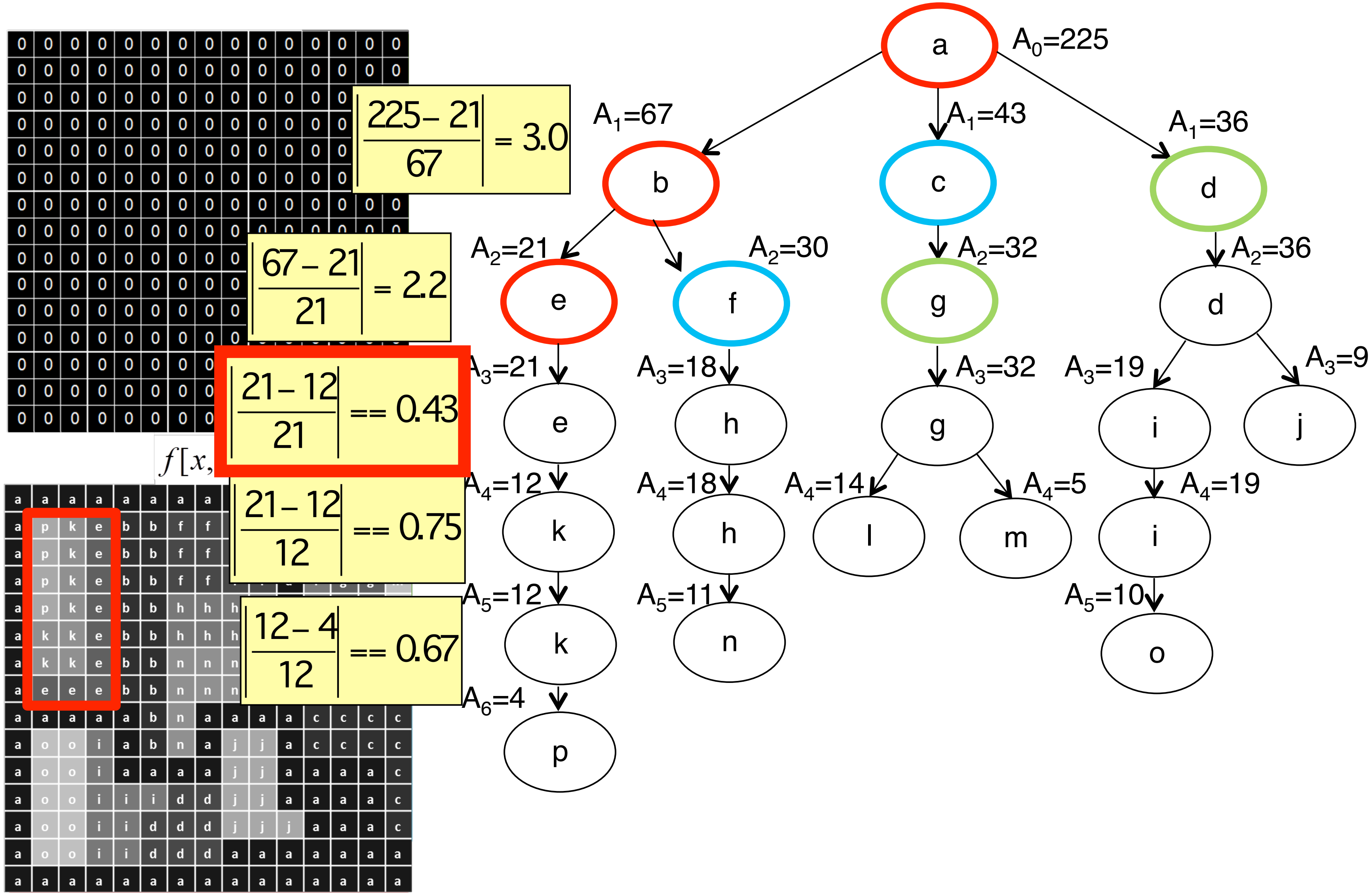
Image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 8$

Level Set

Component tree of an image



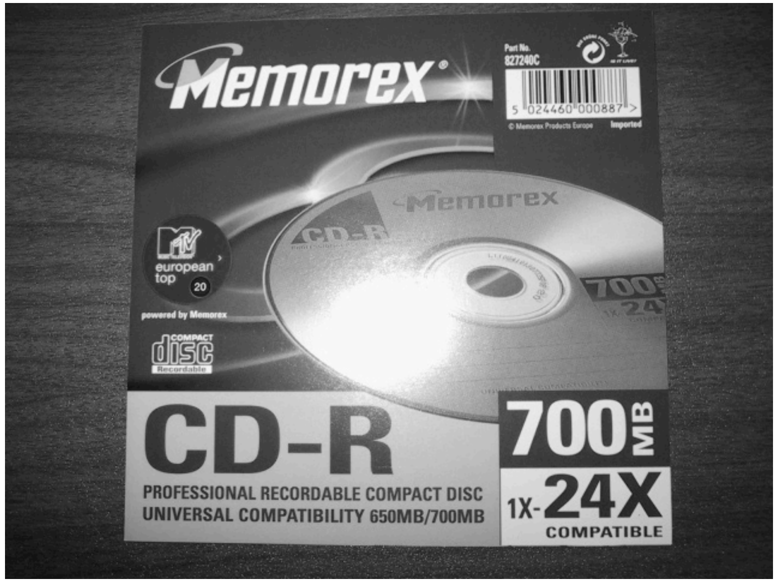
Local minima of sequence

$$\left|\frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_\theta}\right|$$

$\theta = \Delta,\ \Delta+1,\ \ldots \ \rightarrow$ MSERs

MSER: examples

Dark MSERs, Δ=15



Original image

Bright MSERs, Δ=15

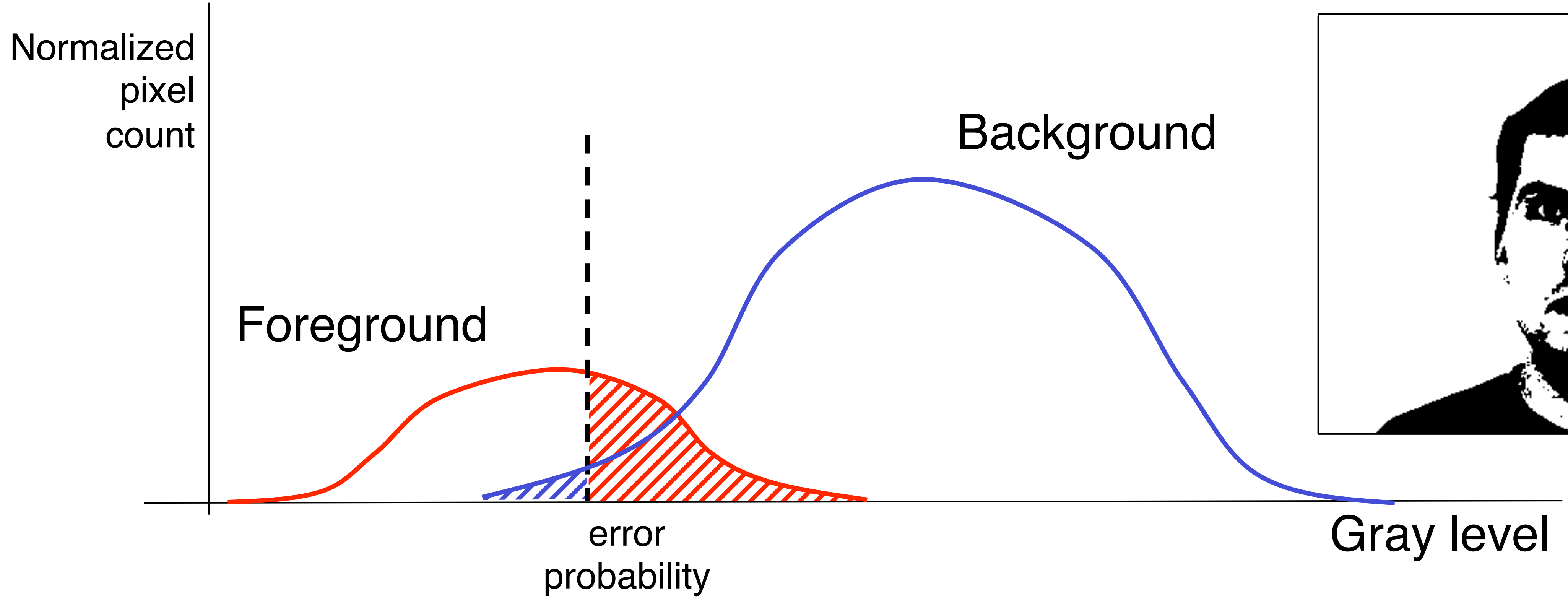
MSER: examples

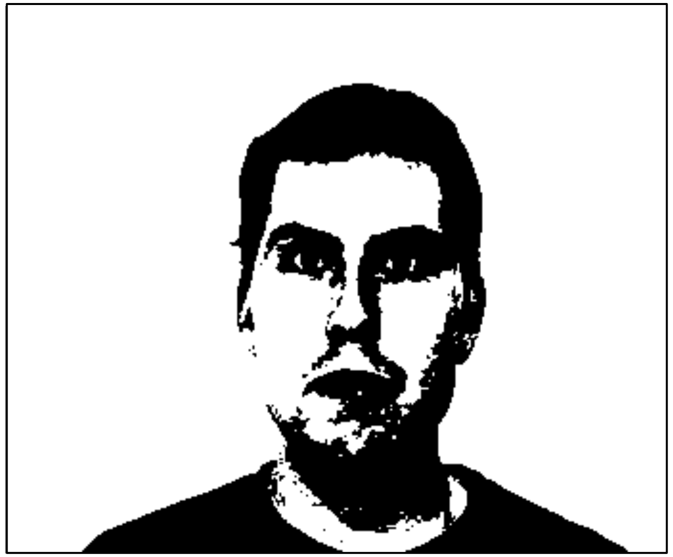
Dark MSERs, Δ=15

Original image

Bright MSERs, Δ=15

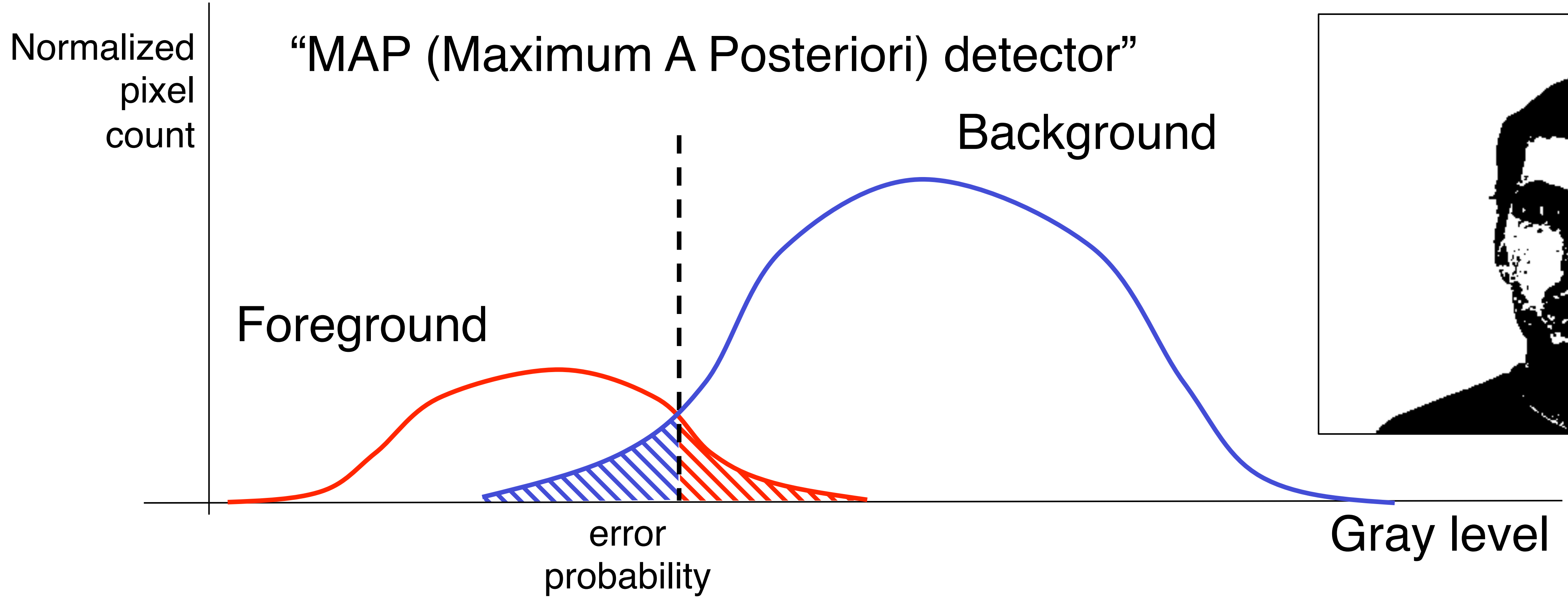
Supervised thresholding





Supervised thresholding

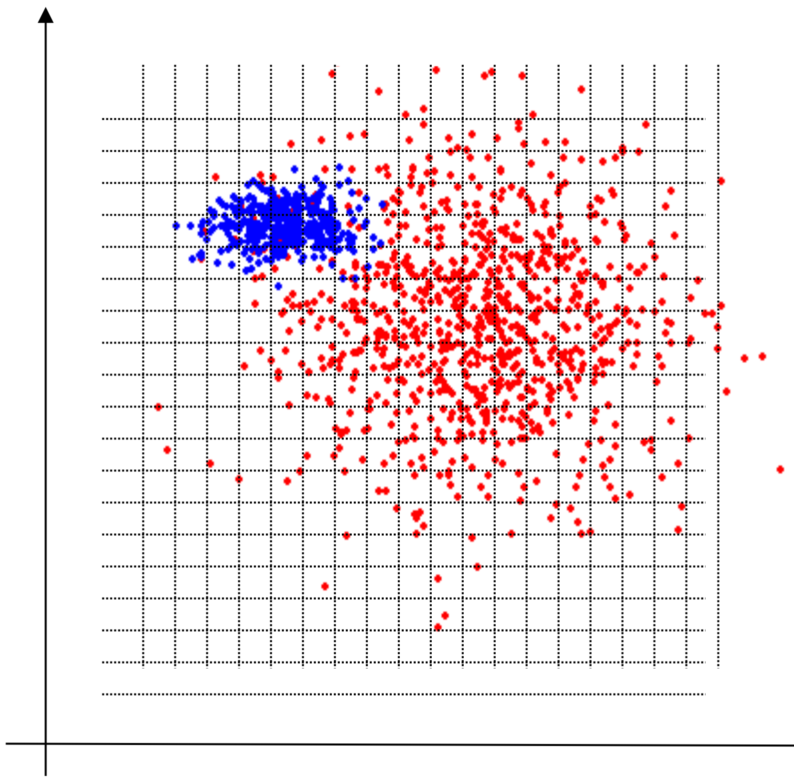
"MAP (Maximum A Posteriori) detector"



If errors BG➔FG and FG➔BG are associated with different costs:
"Bayes minimum risk detector" is optimal.

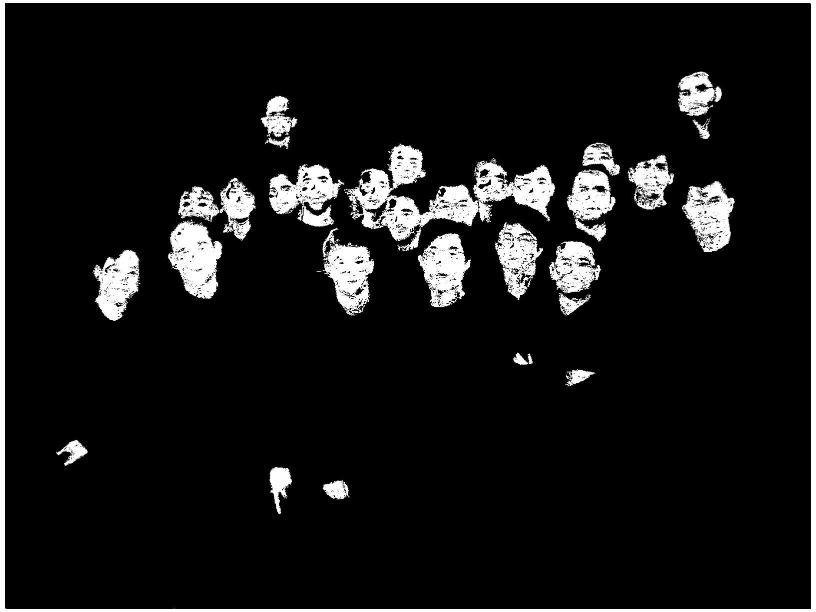
Multidimensional MAP detector

- 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

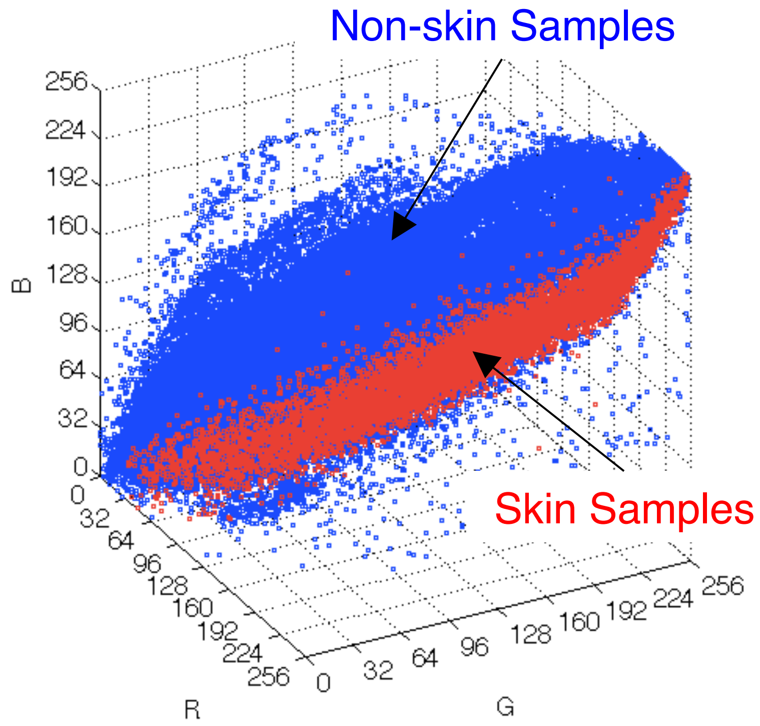


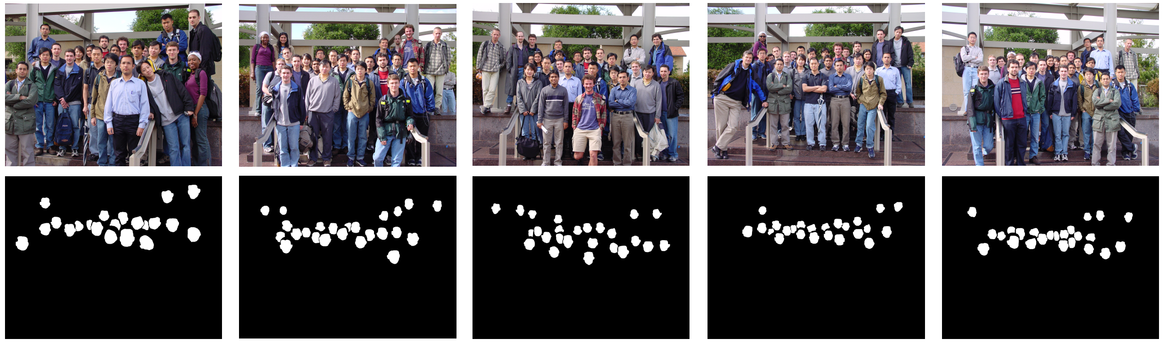
MAP detector in RGB-space

Original image



Skin color detector



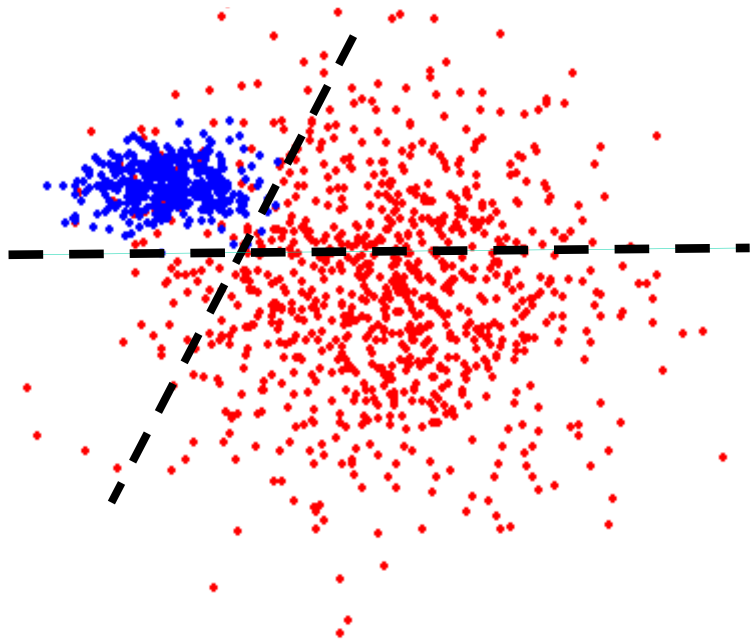


Five training images

Linear discriminant function

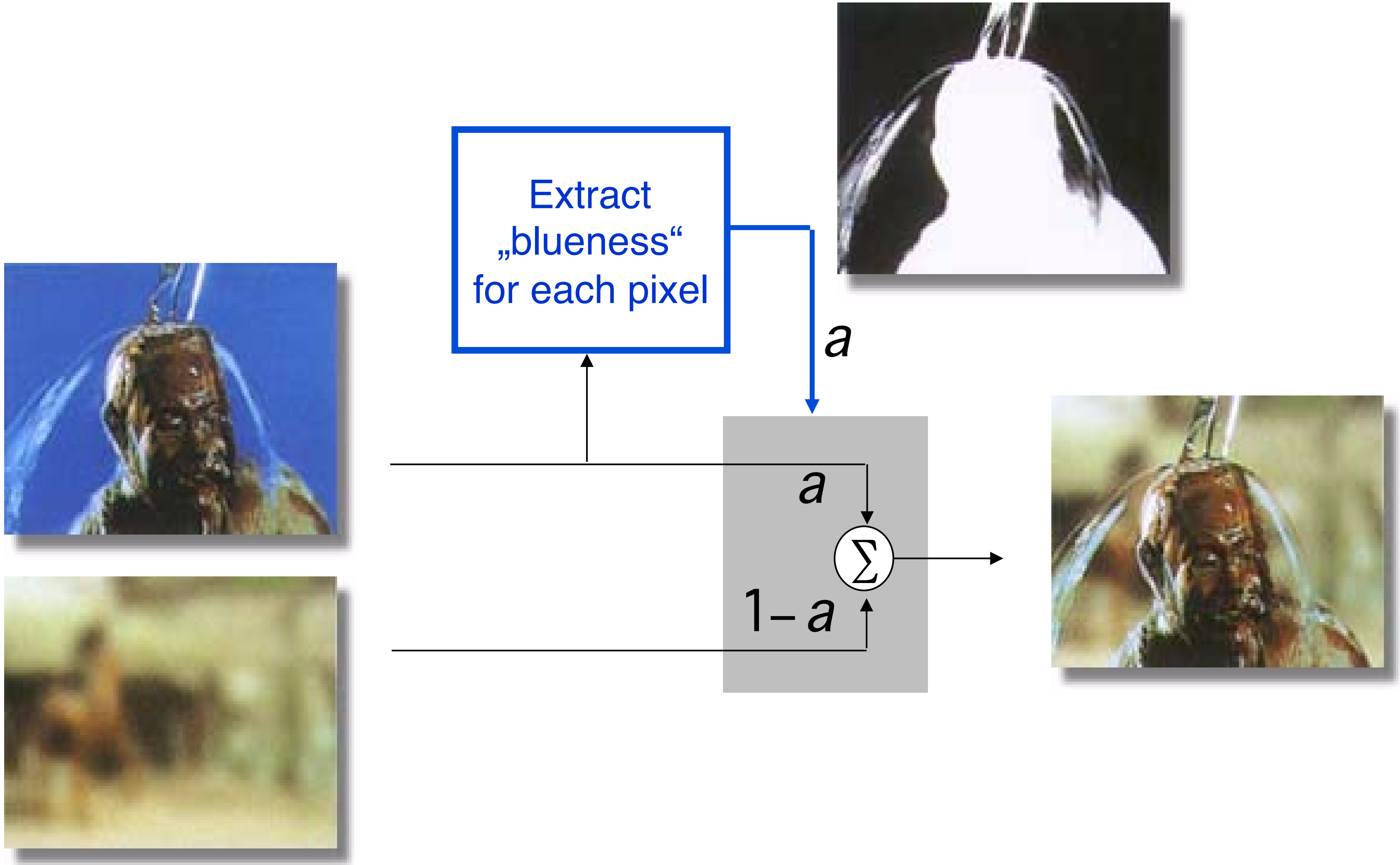
- To segment image with n components f_i, $i=1,2,\ldots,n$ into two classes, perform test

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

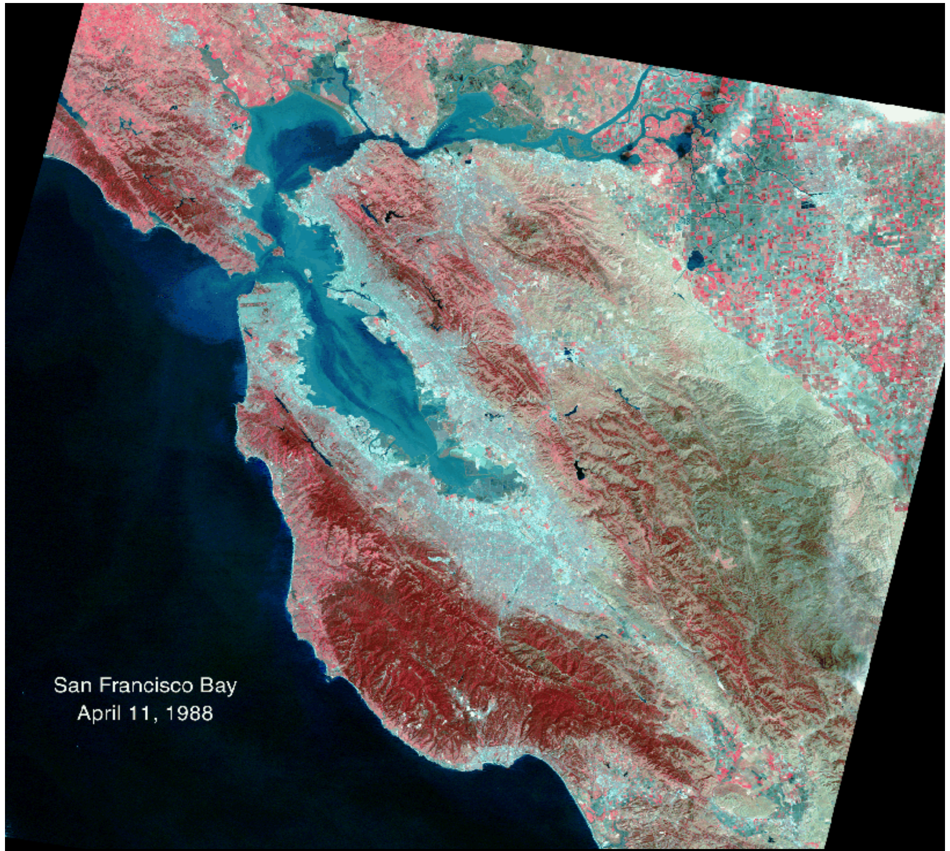


- Categories are separated by hyperplane in n-space
- Numerous techniques to determine weights w_i, $i=0,1,2,\ldots,n$, see, e.g., *[Duda, Hart, Stork, 2001]*
- Can be extended to the intersection of several linear discriminant functions
- Can be extended to multiple classes

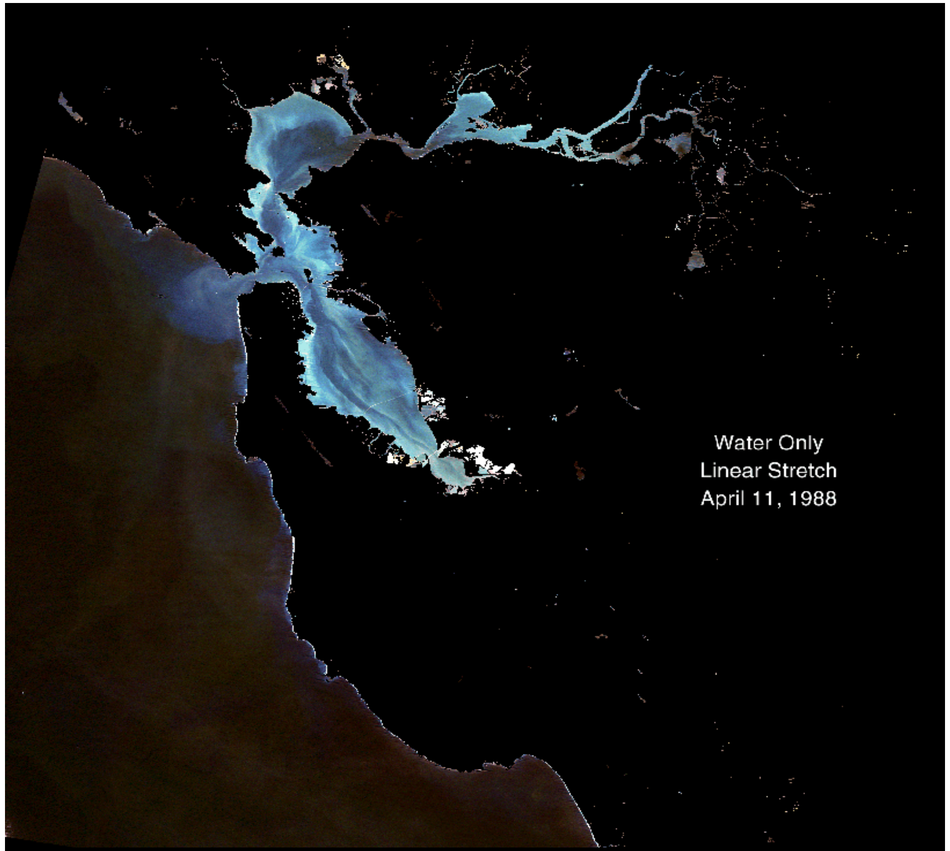
Chroma keying



Landsat image processing



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

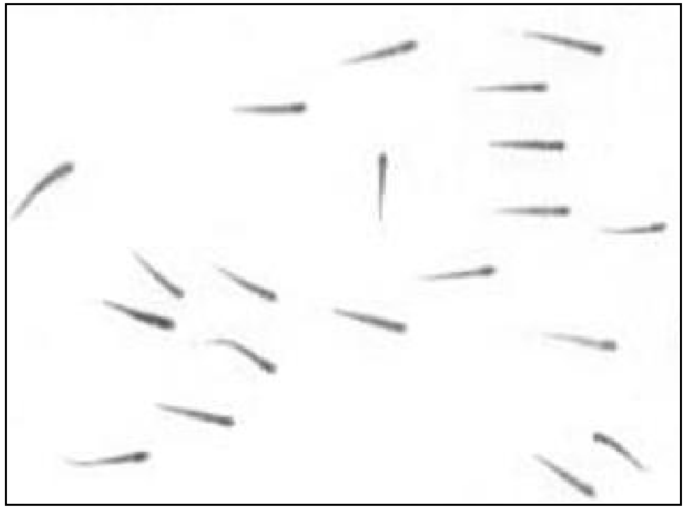


Water area segmented and enhanced to show sediments

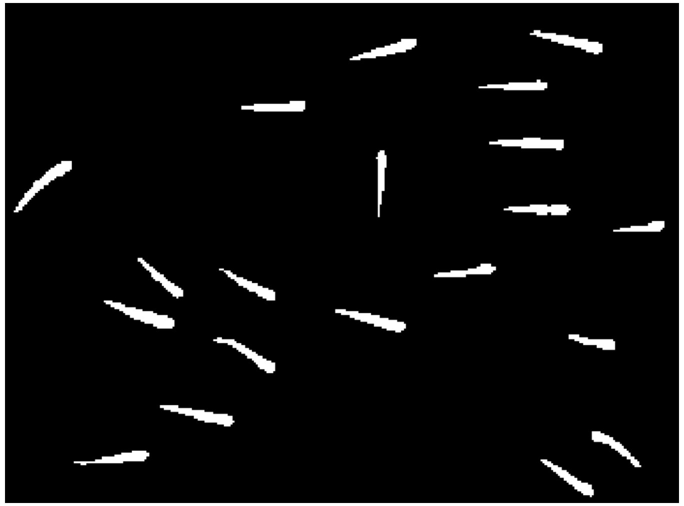
Source: US Geological Survey USGS, http://sfbay.wr.usgs.gov/

Region labeling and counting

- How many fish in this picture?



Original *Fish* image



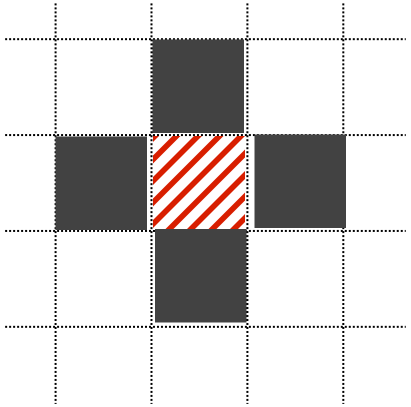
after thresholding

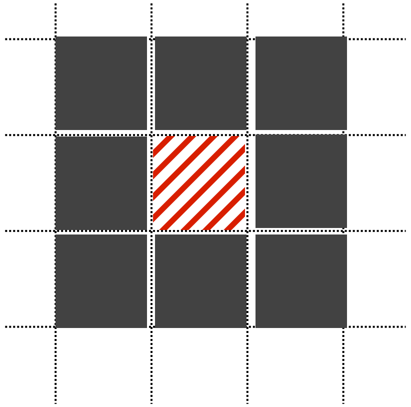
- Which pixels belong to the same object (region labeling)?
- How large is each object (region counting)?

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



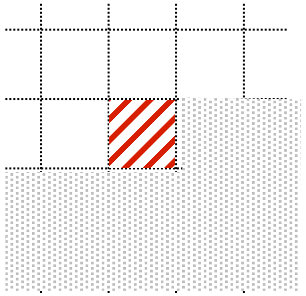


8-neighborhood

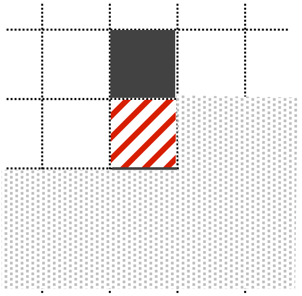
- Typically, either definition leads to the same regions, except when a region is only connected across diagonally adjacent pixels.

Region labeling algorithm (4-neighborhood)

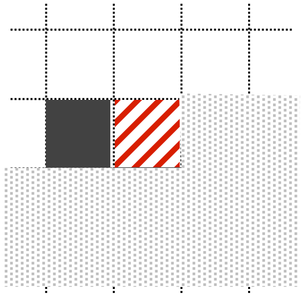
- Loop through all pixels $f[x,y]$, left to right, top to bottom
- If $f[x,y]=0$, do nothing.
- If $f[x,y]=1$, distinguish 4 cases



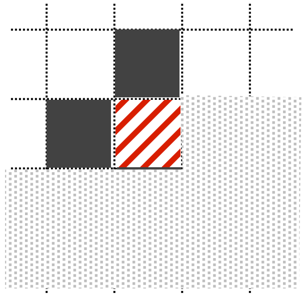
Generate new region label



Copy label from above



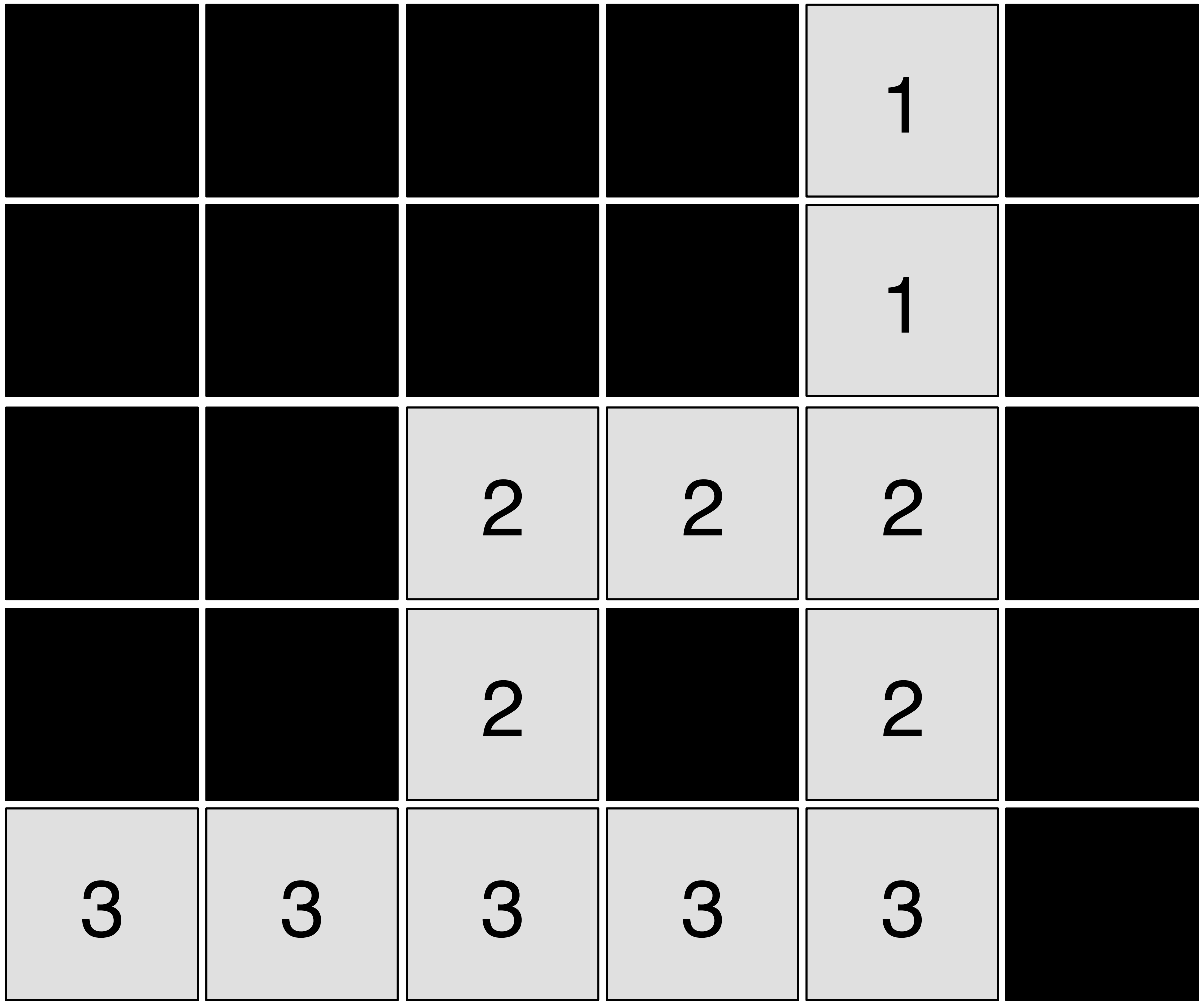
Copy label from the left

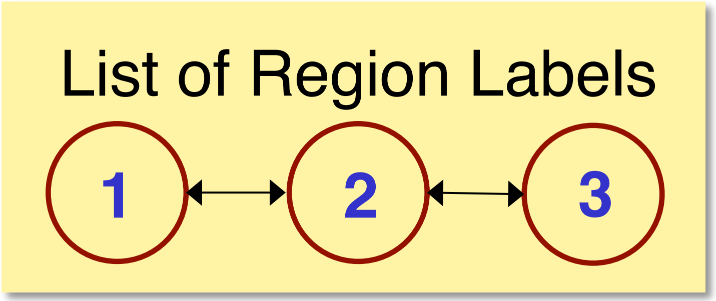


Copy label from the left. If labels above and to the left are different, store equivalence.

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

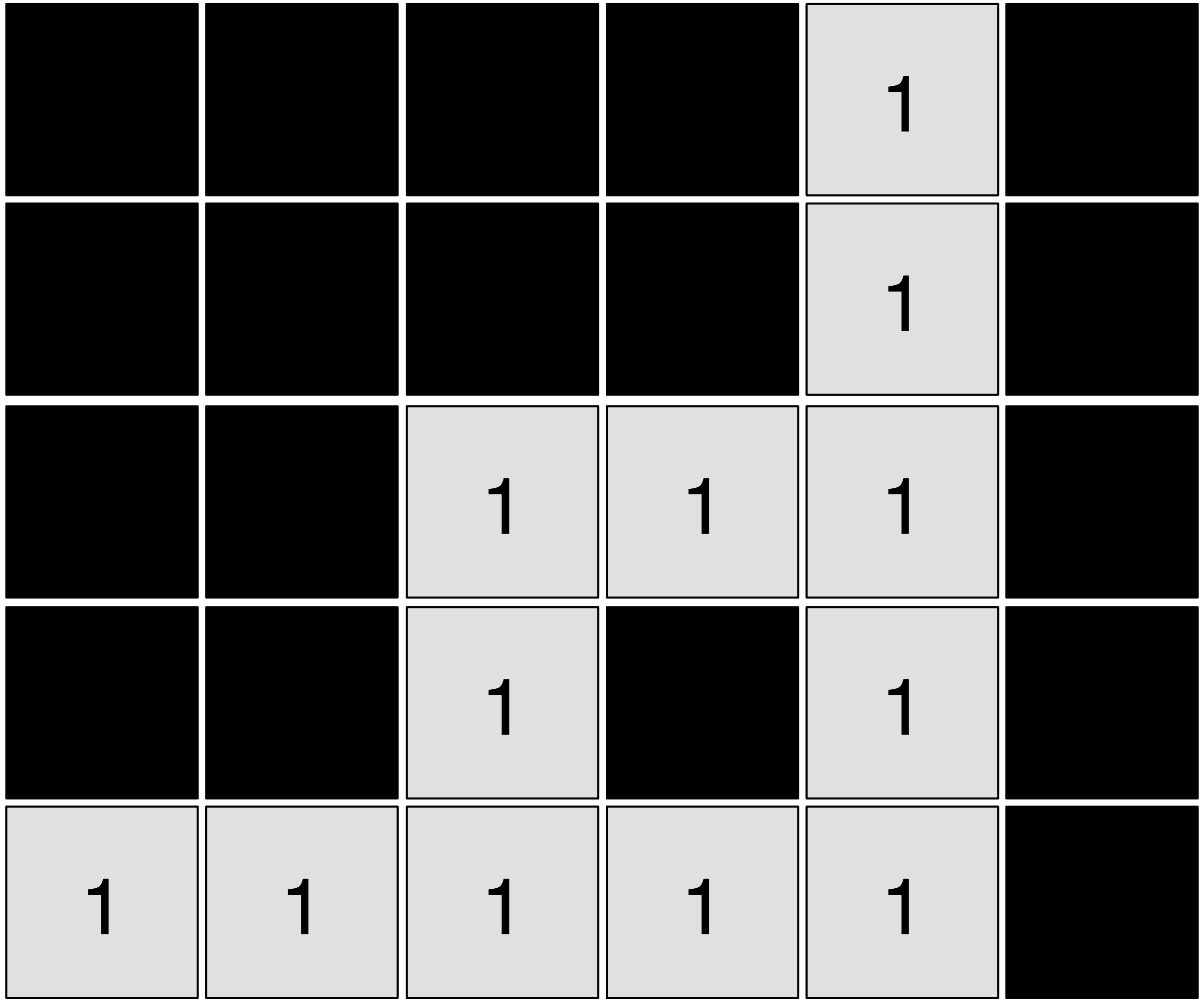
Region labeling example (4-neighborhood)





All three labels are equivalent, so merge into single label.

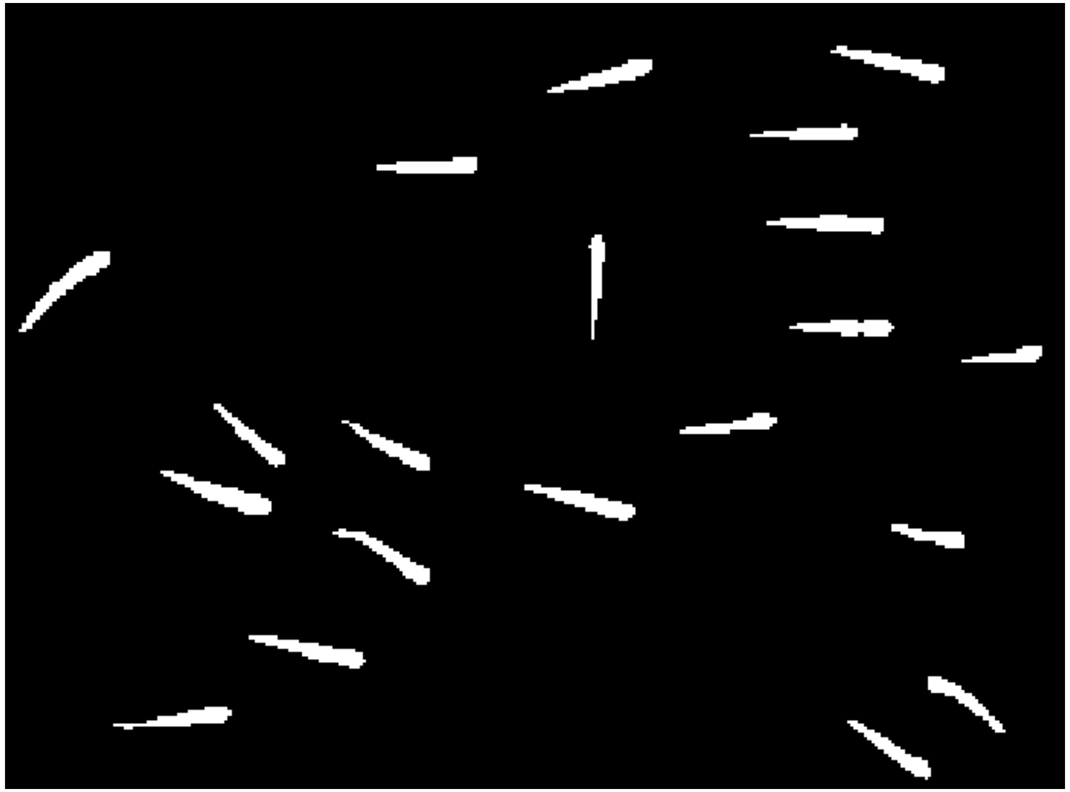
Region labeling example (4-neighborhood)



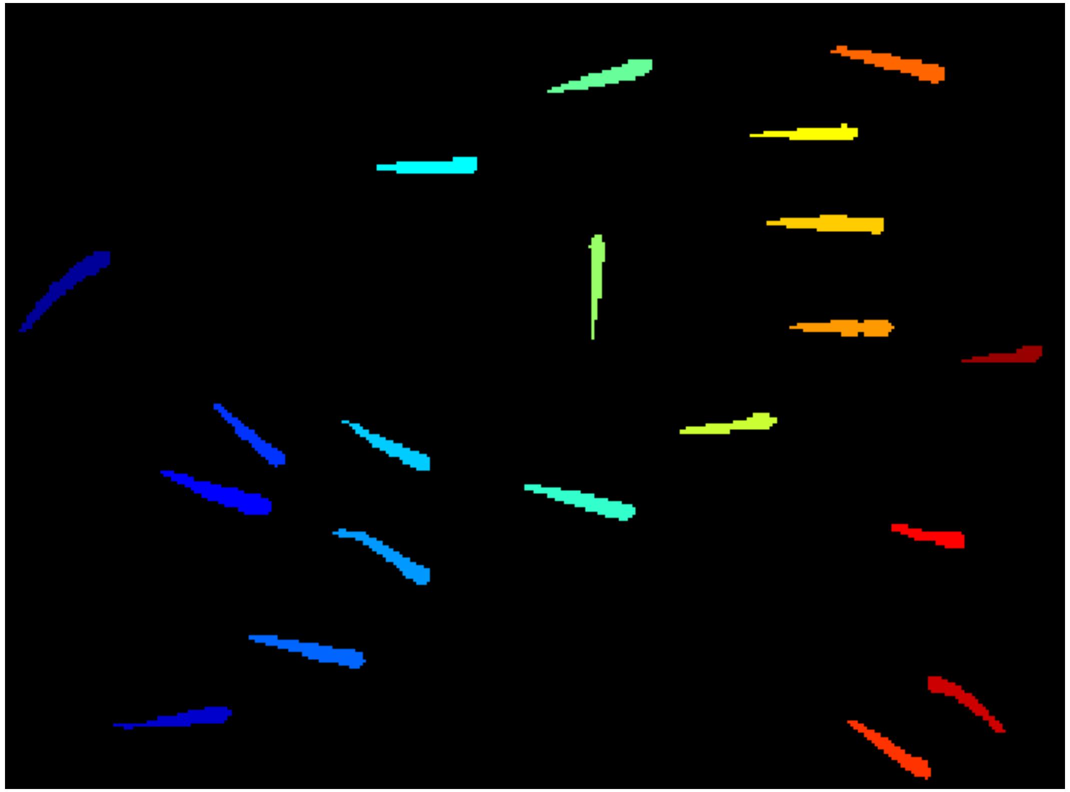
List of Region Labels

1

Example: region labeling



Thresholded image



20 labeled regions

Region counting algorithm

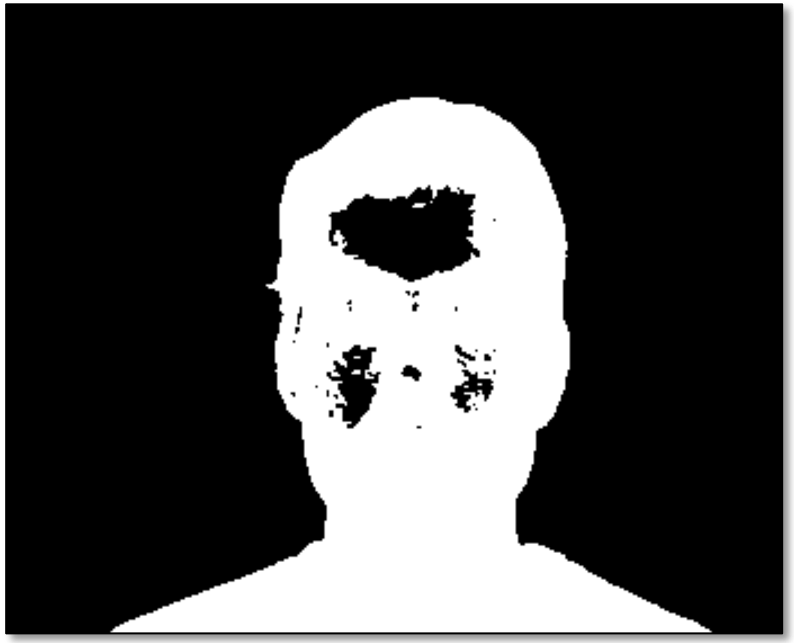
- Measures the size of each region
- Initialize $counter[label]=0$ for all $label$
- Loop through all pixels $f\,[x,y]$, left to right, top to bottom
  - If $f\,[x,y]=0$, do nothing.
  - If $f\,[x,y]=1$, increment $counter[label[x,y]]$

Small region removal

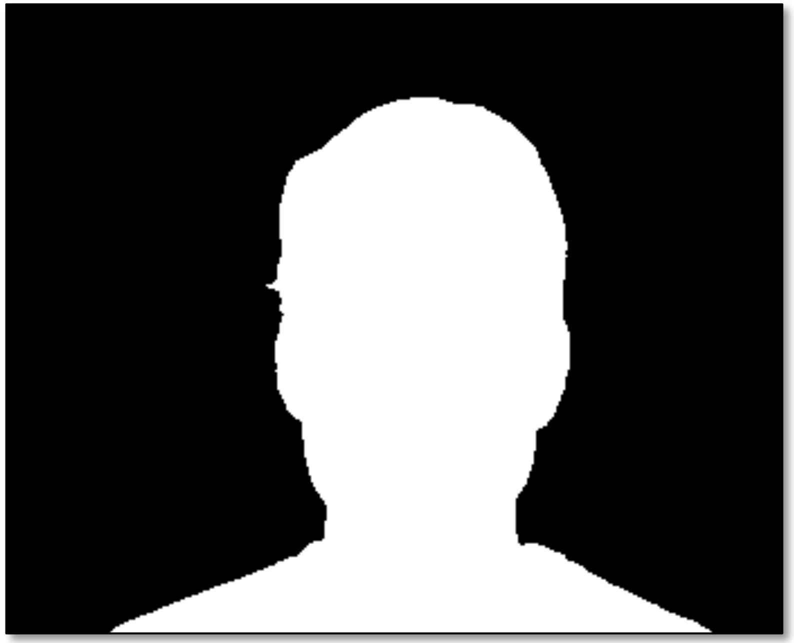
- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$ and $counter[label[x,y]]<S$, set $f[x,y]=0$
- Removes all regions smaller than S pixels

Hole filling as dual to small region removal

Mask with holes



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

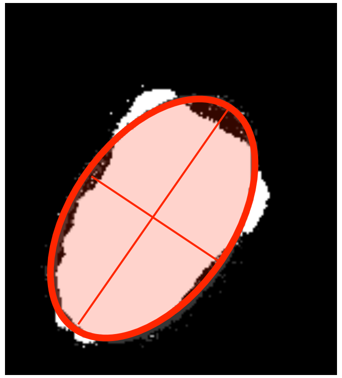


Region moments

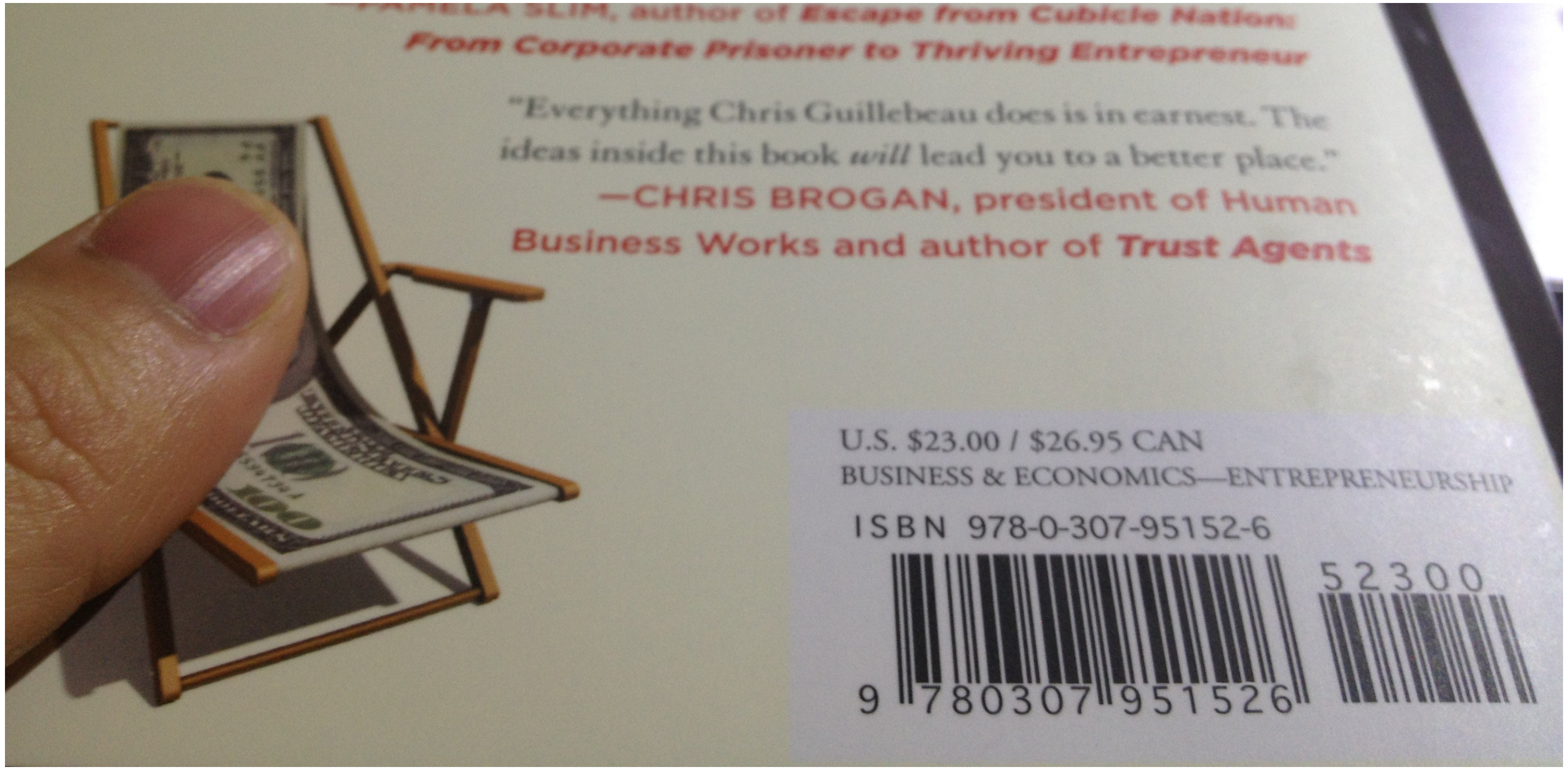
- Raw moments $$M_{pq} = \sum_{x,y \in \text{Region}} x^p y^q$$
- Central moments

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

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



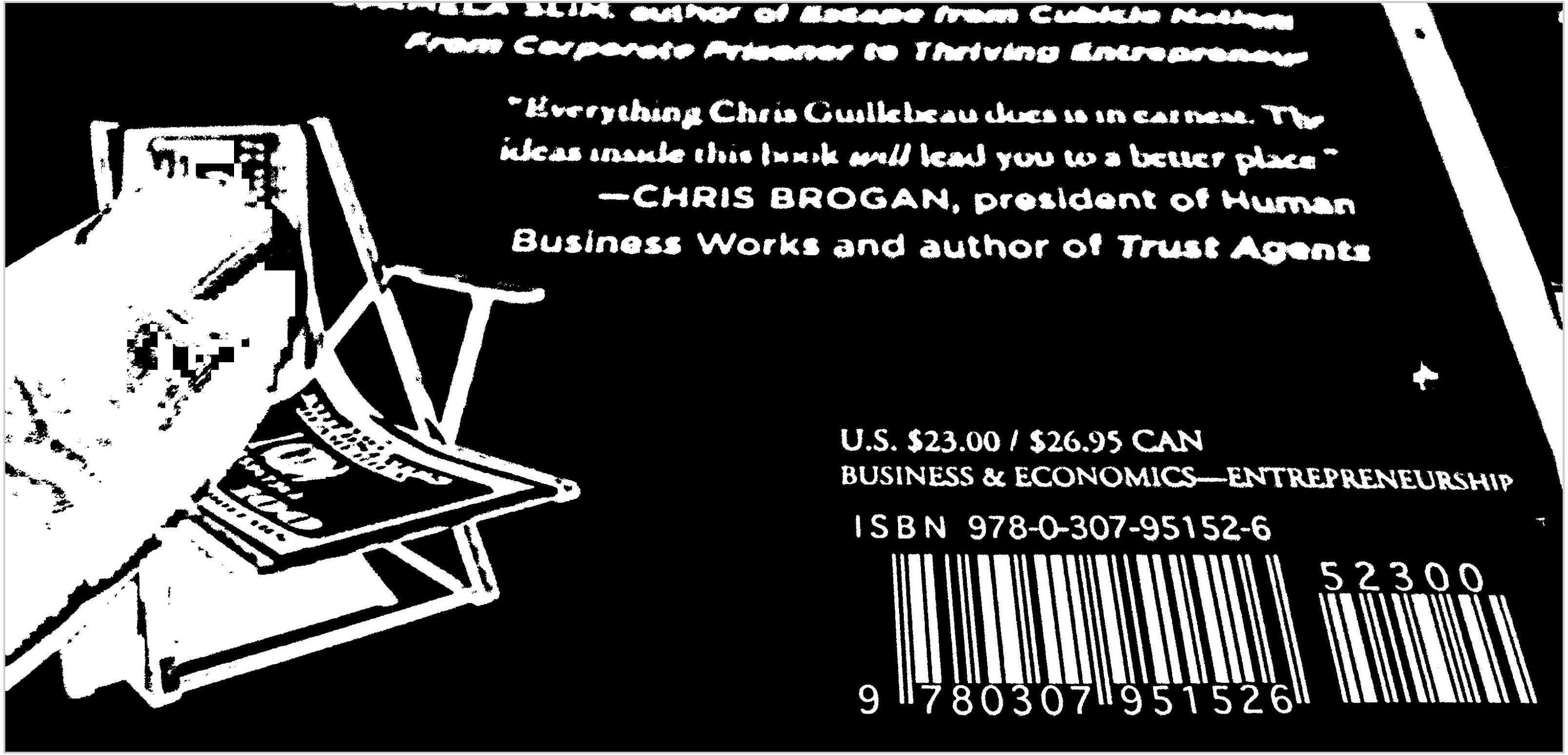
Example: Detecting bar codes



Original Image

Example: Detecting bar codes

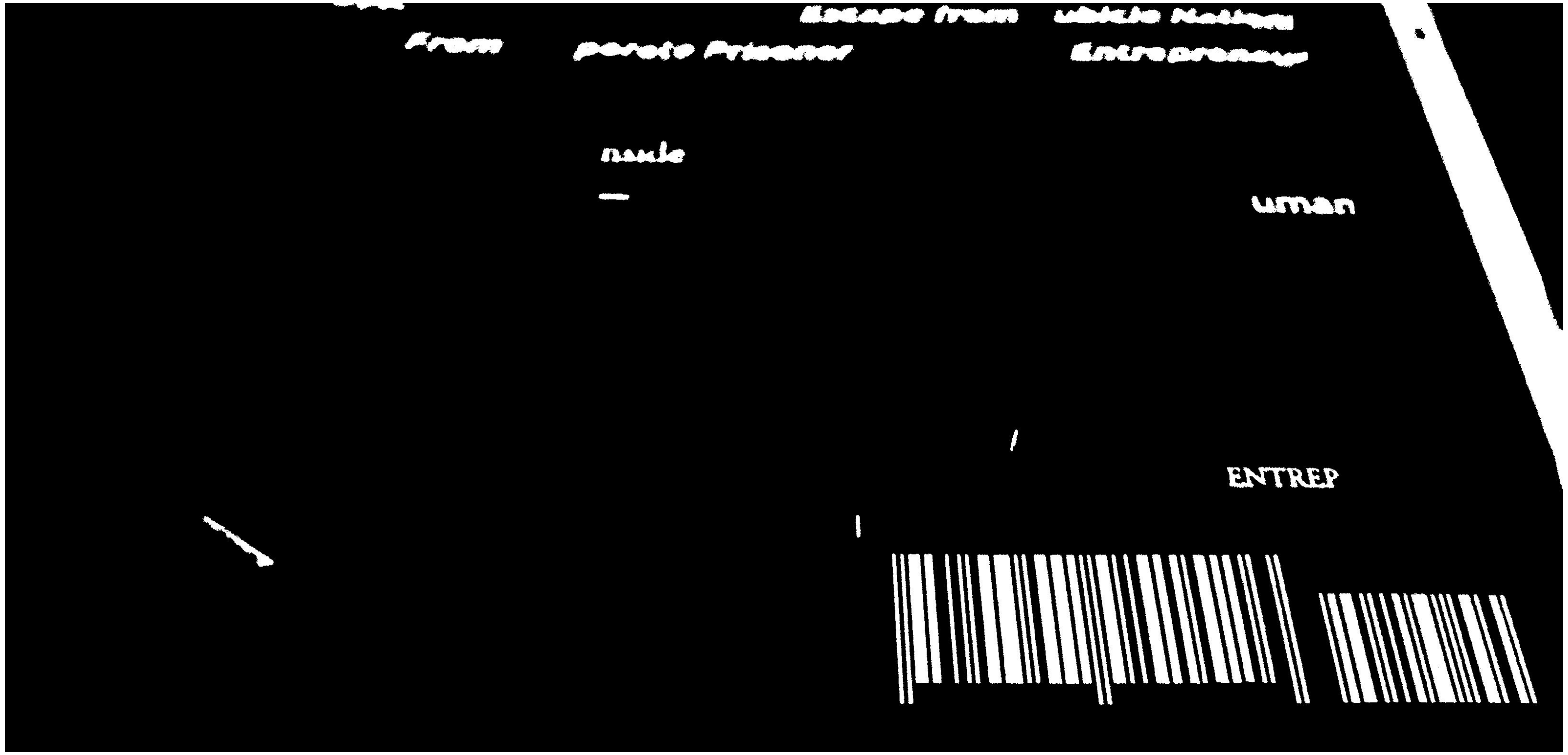
Locally adaptive thresholding



Example: Detecting bar codes

Locally adaptive thresholding

Filtering by eccentricity

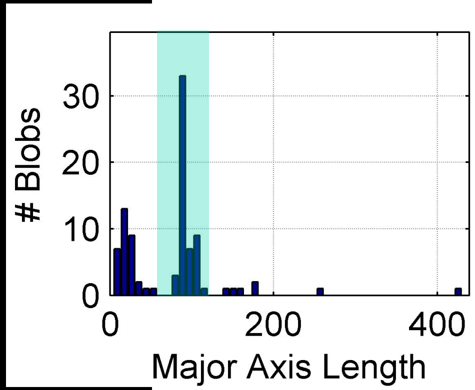


Example: Detecting bar codes

Locally adaptive thresholding

Filtering by eccentricity

Filtering by major axis length



Example: Detecting bar codes

Locally adaptive thresholding

Filtering by eccentricity

Filtering by major axis length

Filtering by orientation