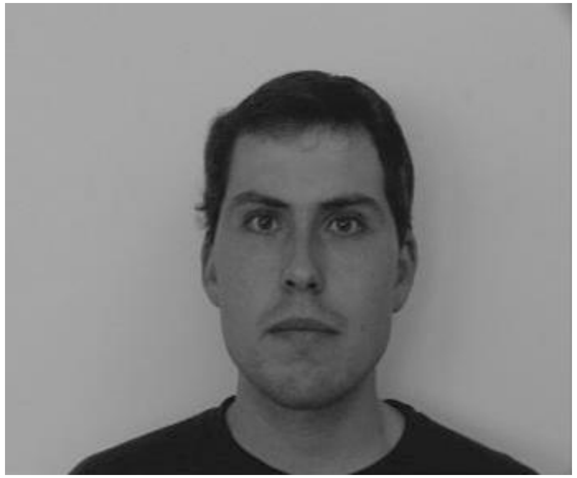
Image Segmentation

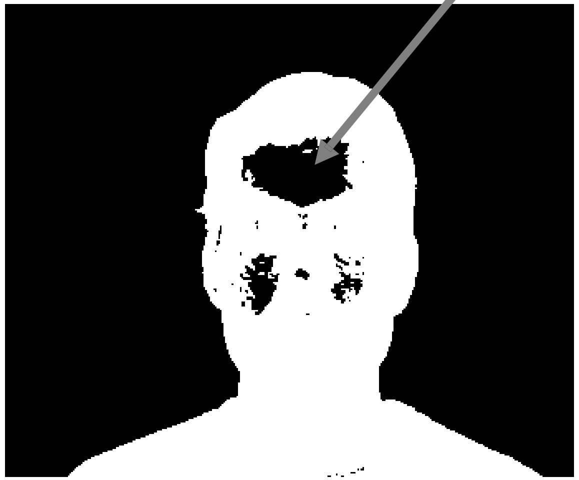
- Gray-level thresholding
- Supervised vs. unsupervised thresholding
- Binarization using Otsu’s method
- Locally adaptive thresholding
- Maximally stable extremal regions
- Color-based segmentation
- Region labeling and counting
- Region moments

Gray-level thresholding

How can holes be filled?



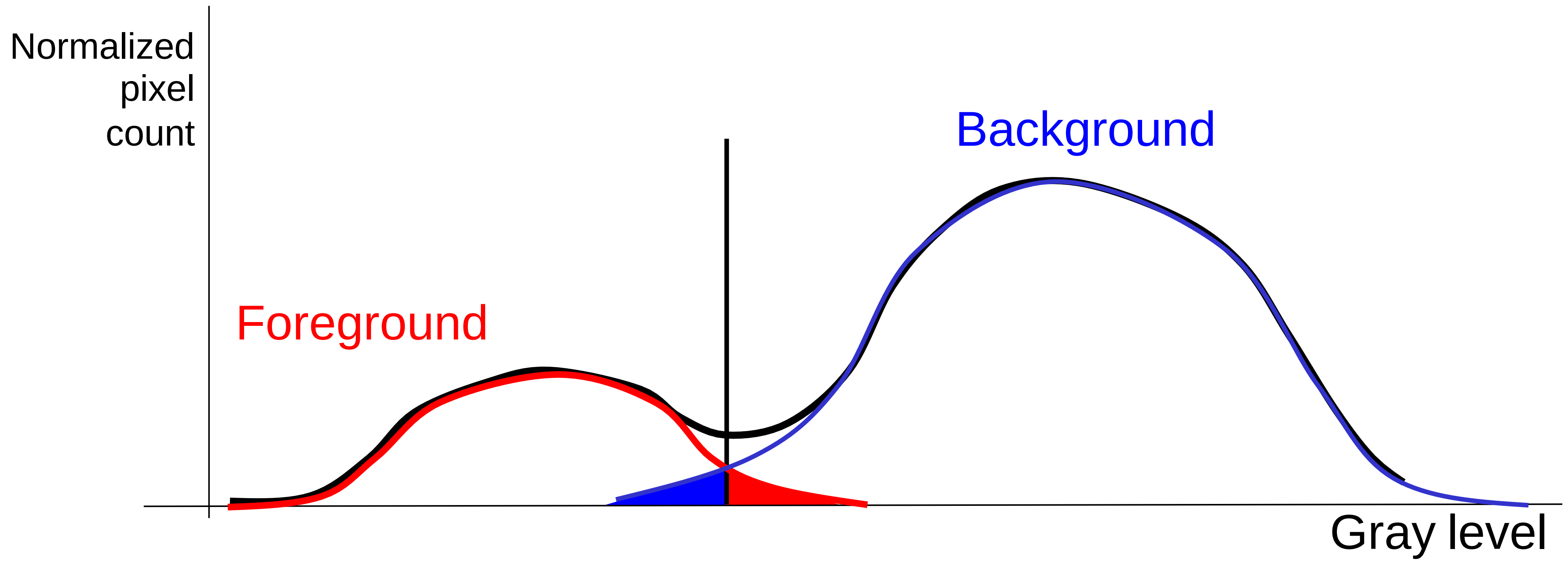
Original image
Peter $f[x,y]$



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)

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

- 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

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

- Useful recursion for sweeping T across histogram:

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

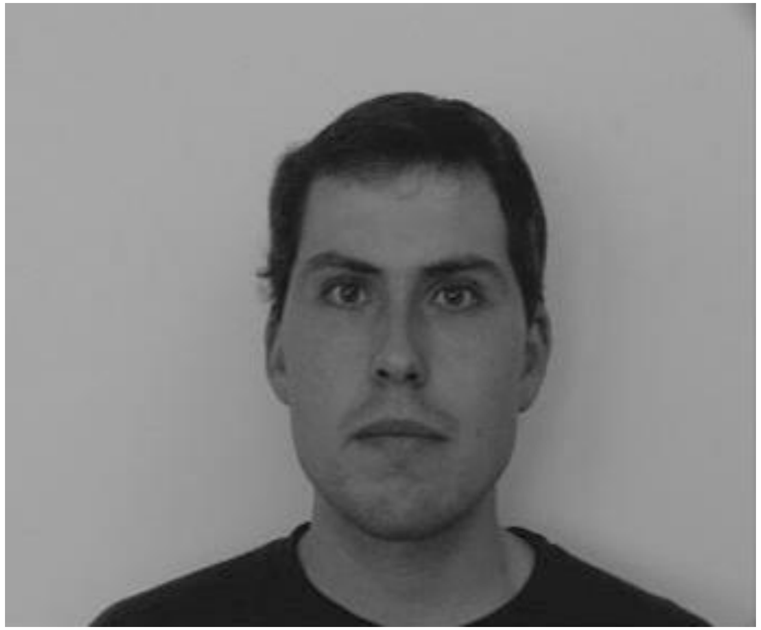
$$N_{Bgrnd}(T+1) = N_{Bgrnd}(T) - n_T$$

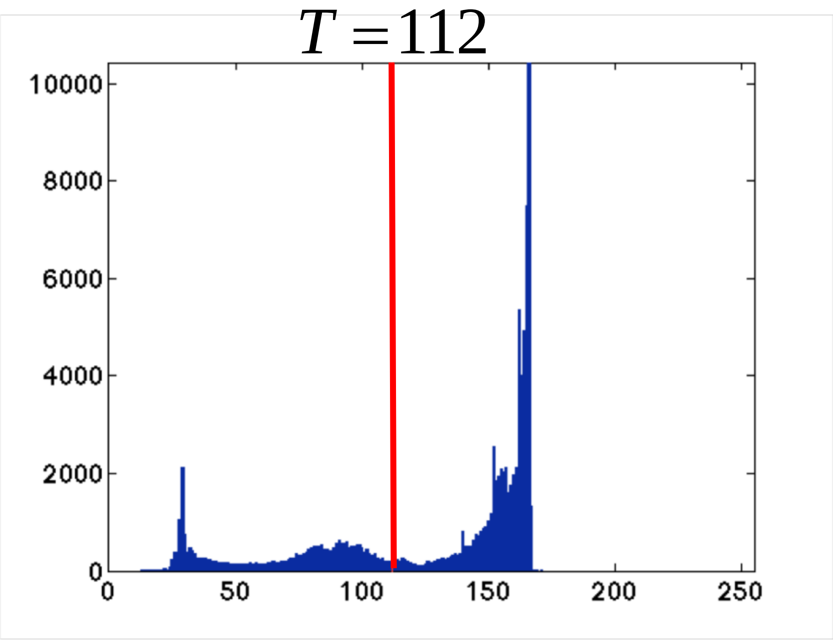
$$\mu_{Fgrnd}(T+1) = \frac{\mu_{Fgrnd}(T) N_{Fgrnd}(T) + n_T T}{N_{Fgrnd}(T+1)}$$

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

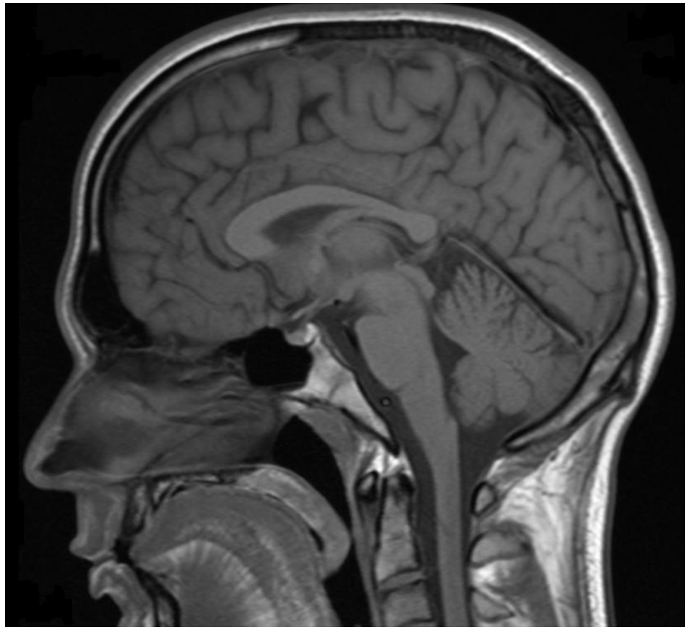
[Otsu, 1979]

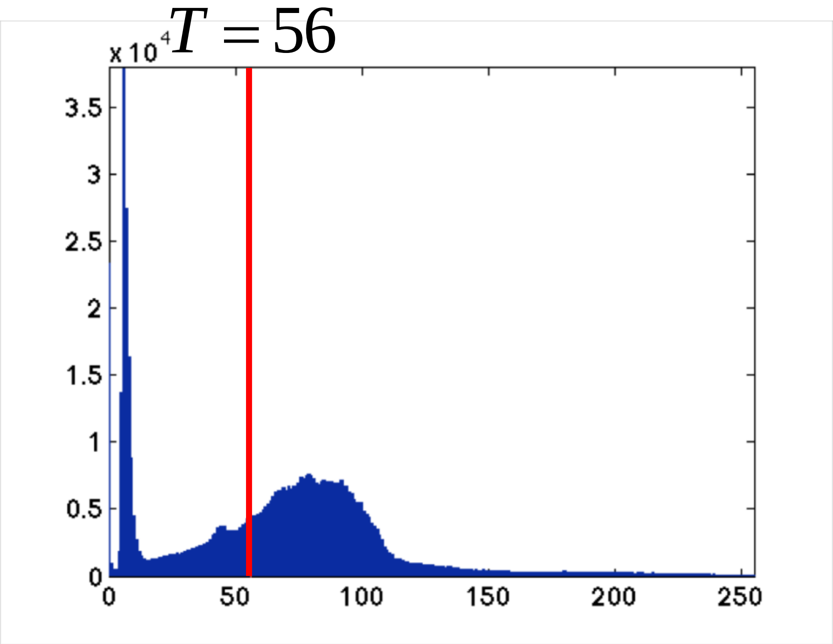
Unsupervised thresholding (cont.)





Unsupervised thresholding (cont.)





Unsupervised thresholding (cont.)

The Stanford Daily — Tuesday, September 18, 2012 ◆ 13

SIMON WARBY/The Stanford Daily

Stanford defensive lineman Josh Mauro put the pressure on USC's Matt Barkley. Mauro was relentleses in the second half as Stanford's defense completely shut down Barkley and his touted wide receivers.

Handing out the USC game balls

By SAM FISHER
FOOTBALL EDITOR

Stepfan Taylor: It all starts and ends with Stanford's workhorse. Taylor was everything you could ask for and more against USC. He provided the big plays with a game-tying touchdown on the ground, another score through the air and the consistent ground-and-pound to wear down the Trojans at the end. His 213 total yards of offense to go with a pair of TDs had fans on both sides forgetting about Andrew Luck.

Josh Mauro: The backup defensive end saw most of his action at nose tackle in the second half, where he completely took over the ball-game. Mauro dominated USC backup center Cyrus Hobbi all half to provide the key pressure up the middle from the defensive line that kept Matt Barkley rattled. The rest of the D-line played great in support, but Mauro went above and beyond the call of duty to help

Please see **AWARDS**, page 15

FOOTBALL

The winding road ahead

By SAM FISHER
FOOTBALL EDITOR

Andrew Luck may be gone, but with Saturday night's win over USC, the Stanford Cardinal put itself in position to achieve beyond the path paved by number 12. You heard right: though there's plenty of work left to do, this 2012 Stanford team showed that it is capable of playing at a national championship level.

Though Stanford survived one of its toughest tests in the gauntlet that is the BCS National Championship eliminator, the road to Miami 2013 is no walk in the park. The toughest challenges remaining on the schedule are games at Notre Dame, Oregon and UCLA, all of whom are currently ranked in the top 20. The next two games, at Washington and then at home against Arizona, are no pushovers either. And as Stanford has showed top-ranked opponents in years past, any team on the Cardinal's schedule has the potential for a magical upset.

From Stanford's current vantage point, there are three paths the rest of the season could take. Door Number One leads to The Promised Land, a berth in the BCS National Championship Game. In all likelihood, because Stanford is not named Alabama or LSU, the Cardinal will have to win out to earn a trip to South Beach, including wins at No. 3 Oregon and a potential rematch against USC for the Pac-12 title.

For Stanford to even think about returning to the site of its 2011 Orange Bowl beat down of Virginia Tech, Josh Nunes will have to build off his fourth-quarter success against USC to play at a higher level consistently. From the 12-yard scramble on third-and-10 on, Nunes was good enough to win a championship. USC's defense is nowhere near a pushover, so Jordan Williamson's 0-3 kicking night is also a major concern. With the way the Stanford defense played against USC over the last 41 minutes Saturday night, a national championship isn't completely out of the question, but it's still not the most likely ending to 2012.

Waiting behind Door Number Two for Stanford is a trip to a BCS bowl for the third straight season. Stanford has a bit more wiggle room to get back to the BCS, thanks to a strong strength of schedule. However, the path is anything but straightforward. One thing we do know is that if Stanford can win the Pac-12, the Rose Bowl becomes the worst-case scenario. To get there, Stanford almost certainly has to win at Oregon on Nov. 17, one of the toughest tasks in all of college football.

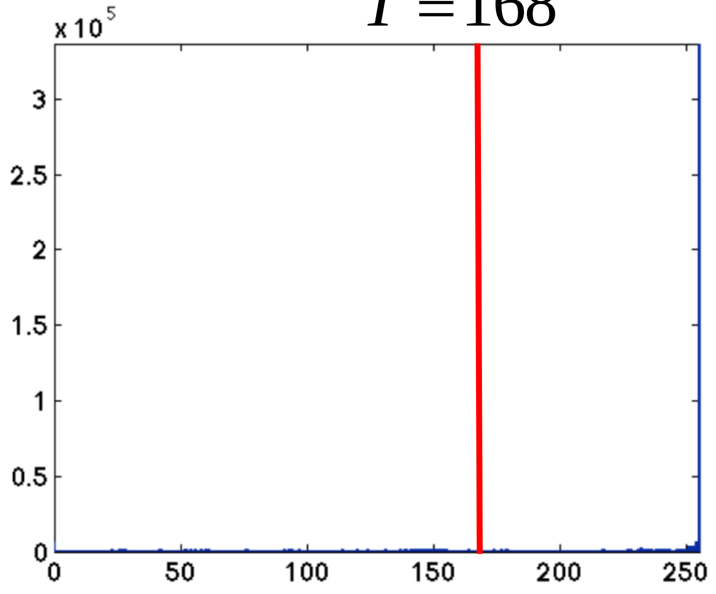
If Stanford can't beat the Ducks, a BCS bowl remains a legitimate possibility. Stanford is still struggling to shake the reputation of not having a big enough fan base to get bowls excited, so grabbing an at-large selection to the BCS won't be easy. Still, as Stanford showed in the last two seasons, beating everyone on the schedule except for Oregon is probably good enough to warrant a top-four BCS ranking and an automatic bid to a BCS game.

Door Number Three is the disappointment, the setback, the wasted opportunity. Stanford put itself in a remarkable position with its win over the second-ranked Trojans. However, if the Card regresses to San Jose State-game form at any point, it is primed to be upset a few times. It might not even take that bad of a performance, as Stanford has five teams left on its schedule in the AP Top 25, including road games at No. 3 Oregon, No. 11 Notre Dame and No. 19 UCLA. If Stanford loses more than two games, it will in all likelihood end up in a second-tier bowl game for the first time since the 2009 season, when Stanford lost a close game to Oklahoma in the Sun Bowl.

There is obviously tremendous uncertainty as to where the rest of the season heads, but it's truly remarkable for Stanford to be in the position it is in now. No Toby Gerhart, no Jim Harbaugh and no Andrew Luck; just Josh Nunes, Stepfan Taylor, a rabid defense and a whole bunch of guys showing a lot of heart. The superstar names might be gone, but the talent left behind is rising to the top, making Door Number One not so crazy to think about.

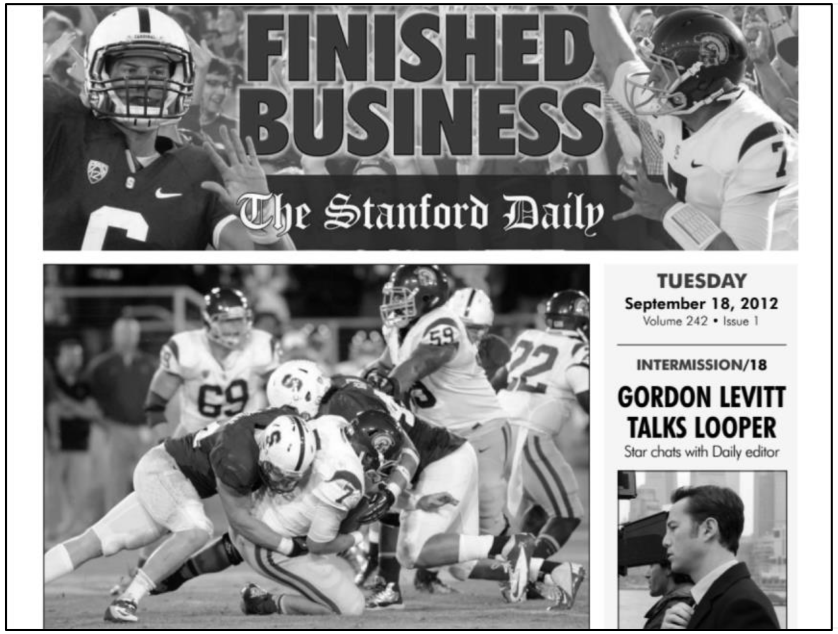
Contact Sam Fisher at safisher "at" stanford.edu.

$T = 168$

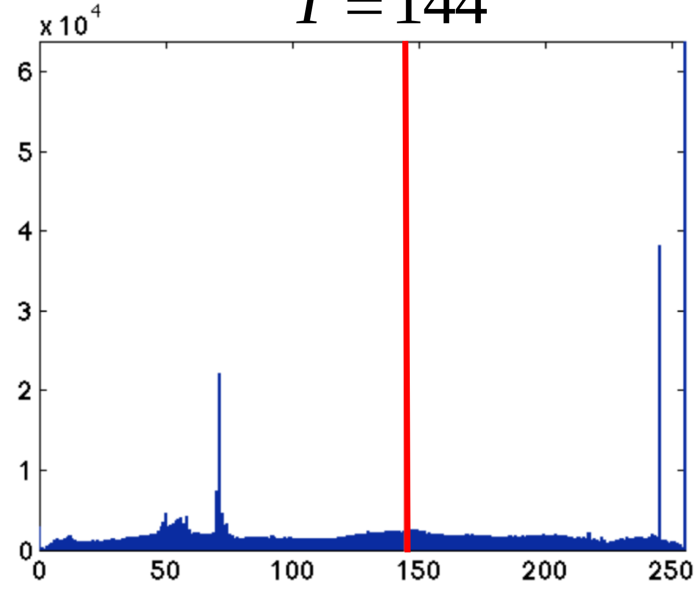


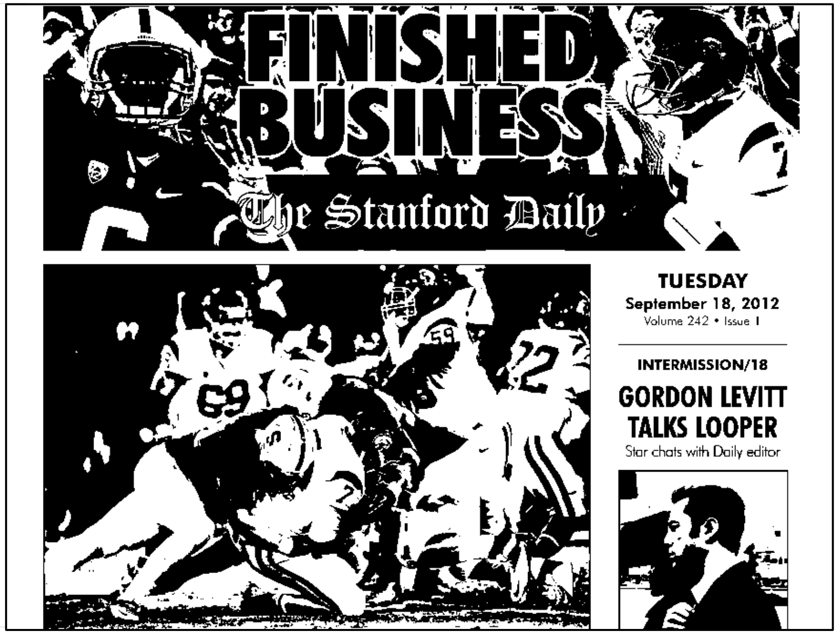
The Stanford Daily — Tuesday, September 18, 2012 ◆ 13
FOOTBALL
The winding road ahead
Handing out the USC game balls

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, Jiang Duan, Guoping Qiu

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

ABSTRACT

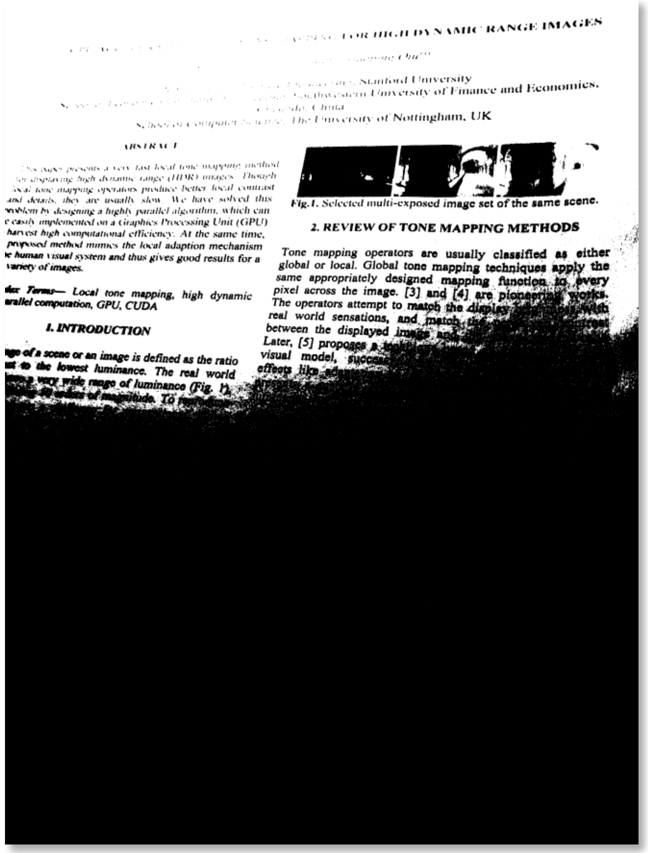
Fig.1. Selected multi-exposed image set of the same scene.

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.

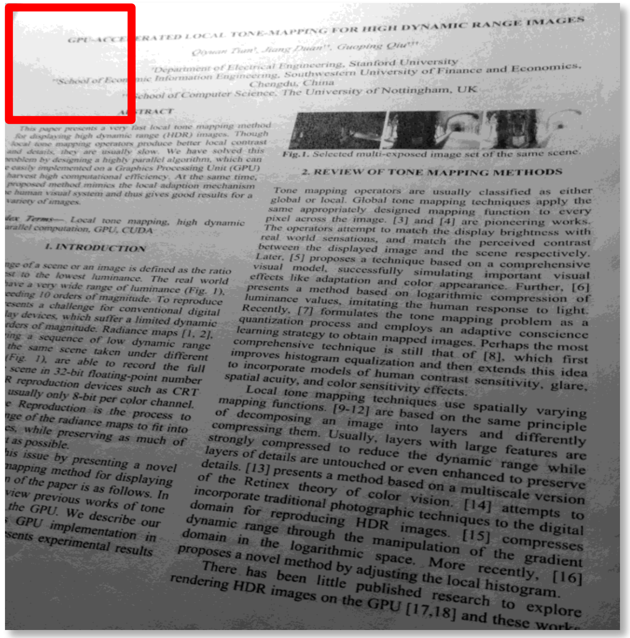
There has been little published research to explore rendering HDR images on the GPU [17,18] and these works



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 local histogram)
- Uniform areas: classify the entire area as foreground or background based on mean value

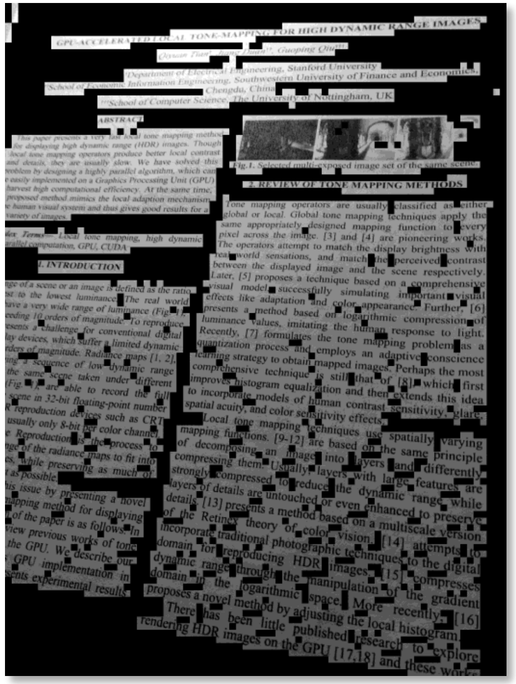

GPU-ACCELERATED LOCAL TONE-MAPPING FOR HIGH DYNAMIC RANGE IMAGES

1. INTRODUCTION

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

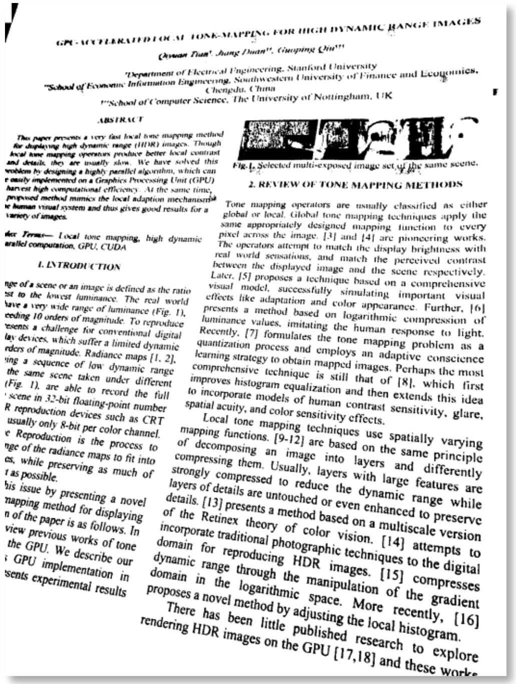
2. REVIEW OF TONE MAPPING METHODS

Locally adaptive thresholding (example)



Non-uniform areas

Local threshold values



Locally thresholded result

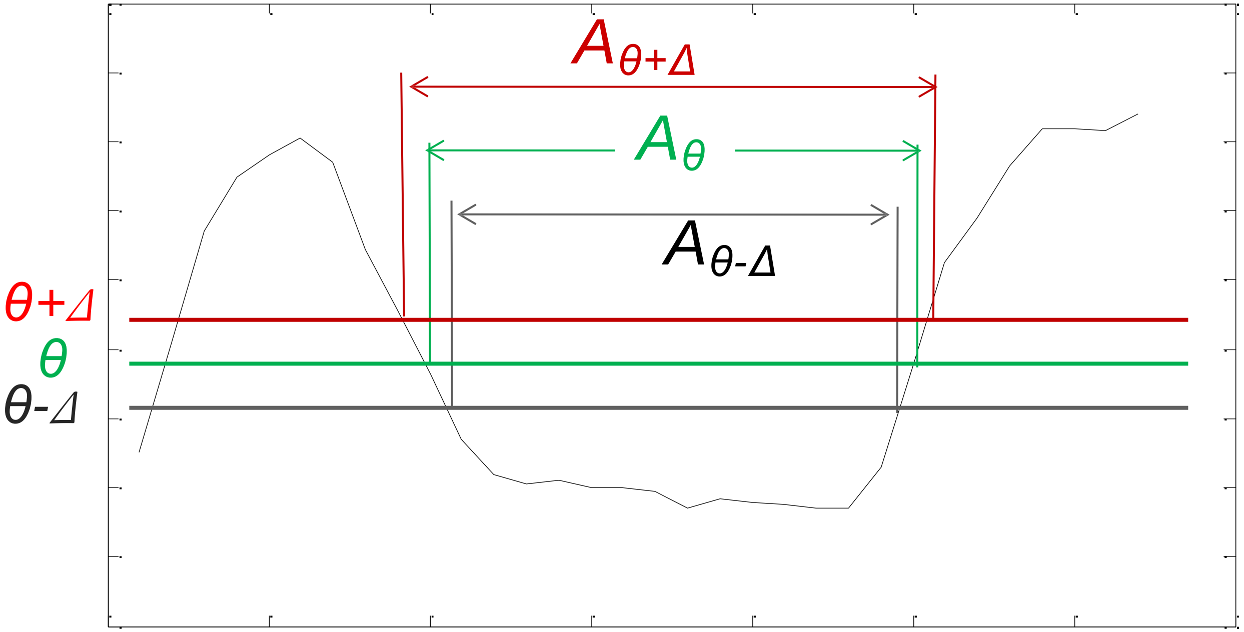
Maximally stable extremal regions

- 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

$f[x, y]$

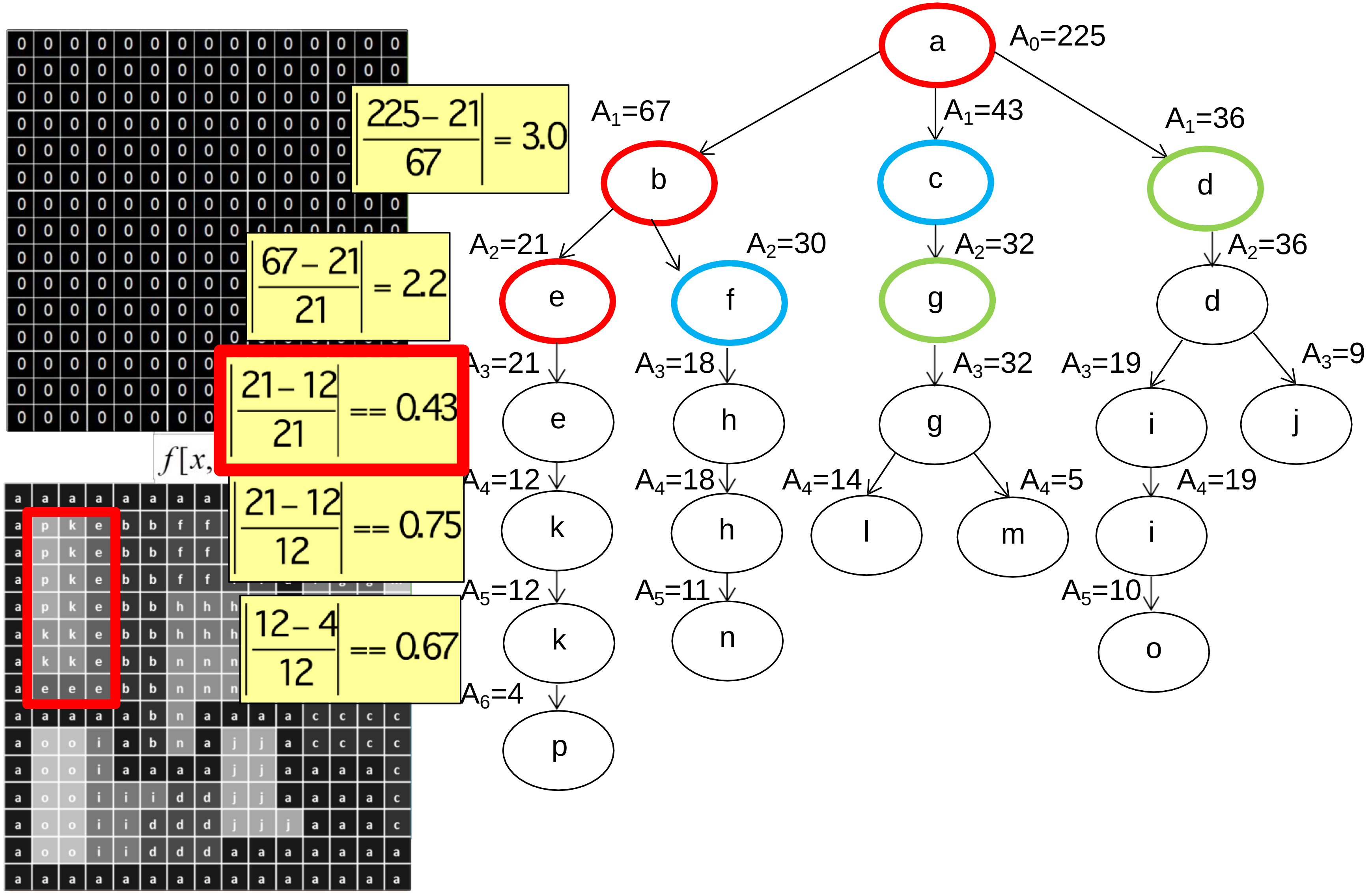
Image

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

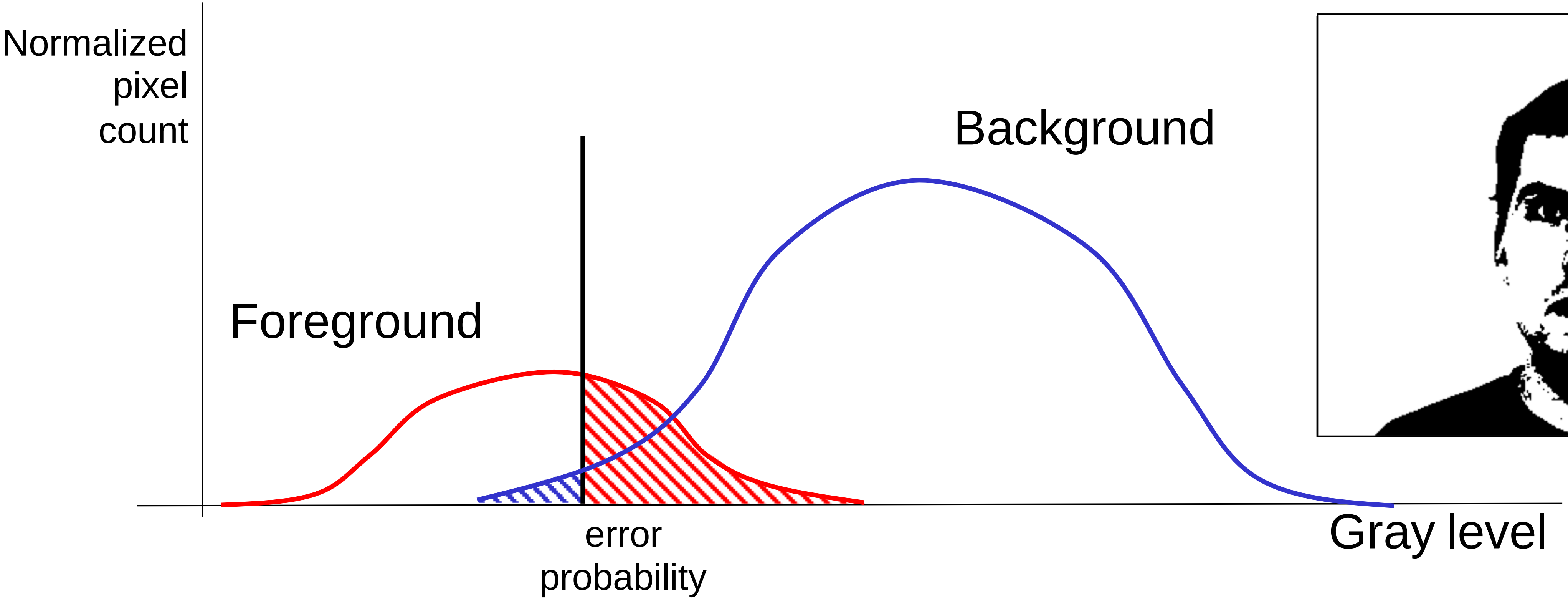
MSER: examples

Dark MSERs, Δ=15

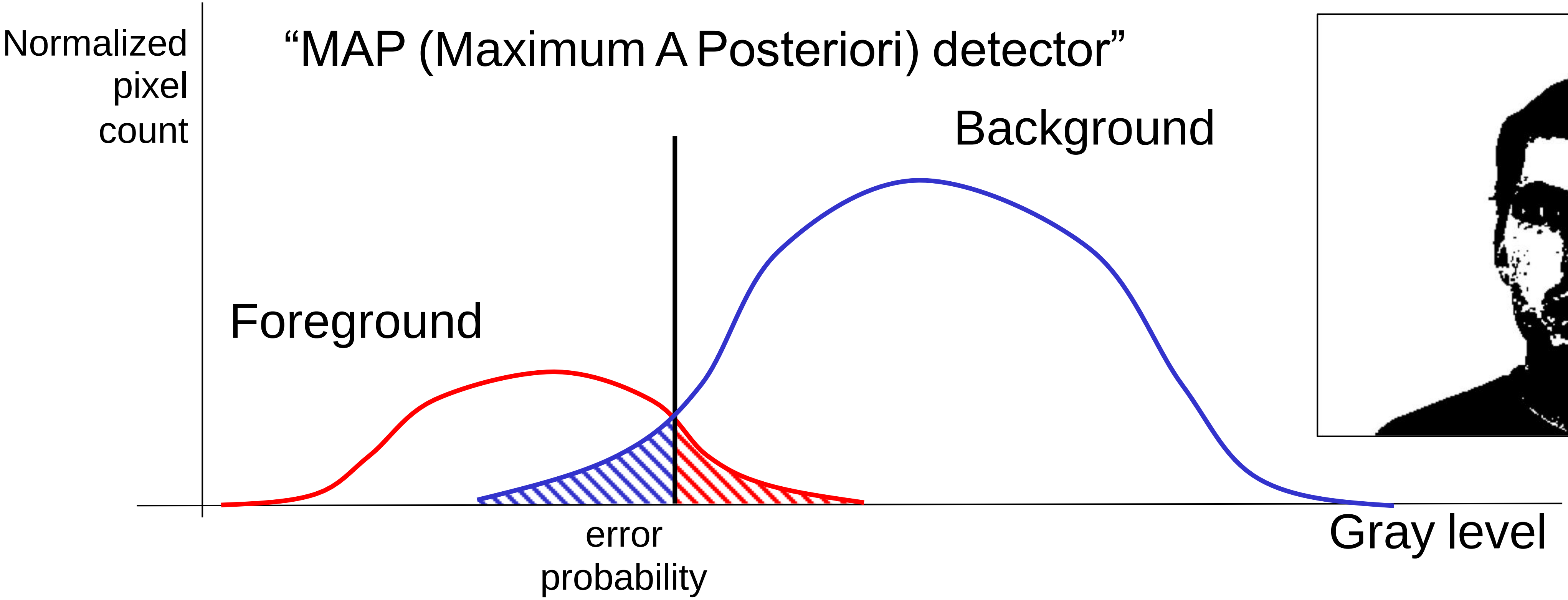
Original image

Bright MSERs, Δ=15

Supervised thresholding



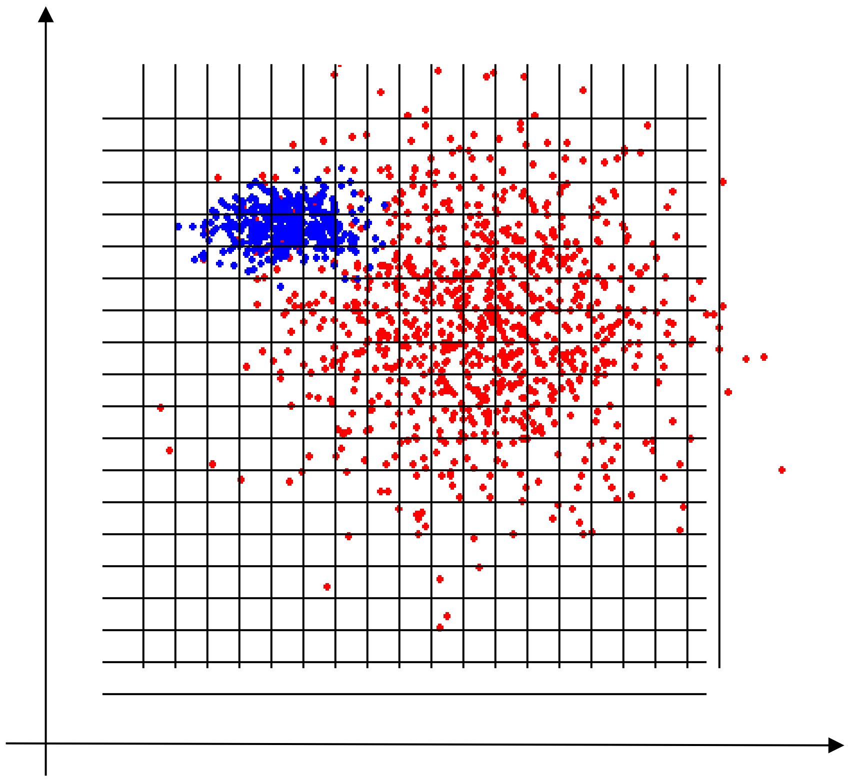
Supervised thresholding



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

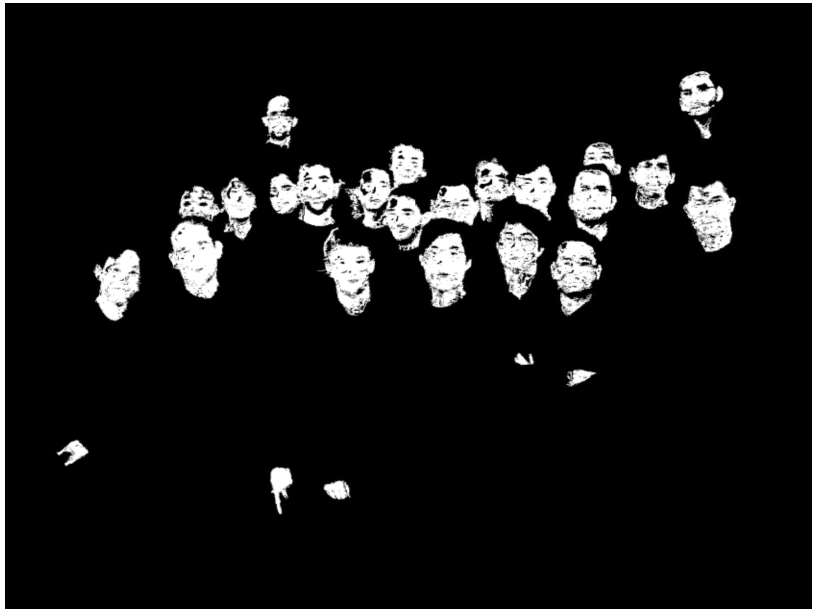
Multidimensional MAP detector

- 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

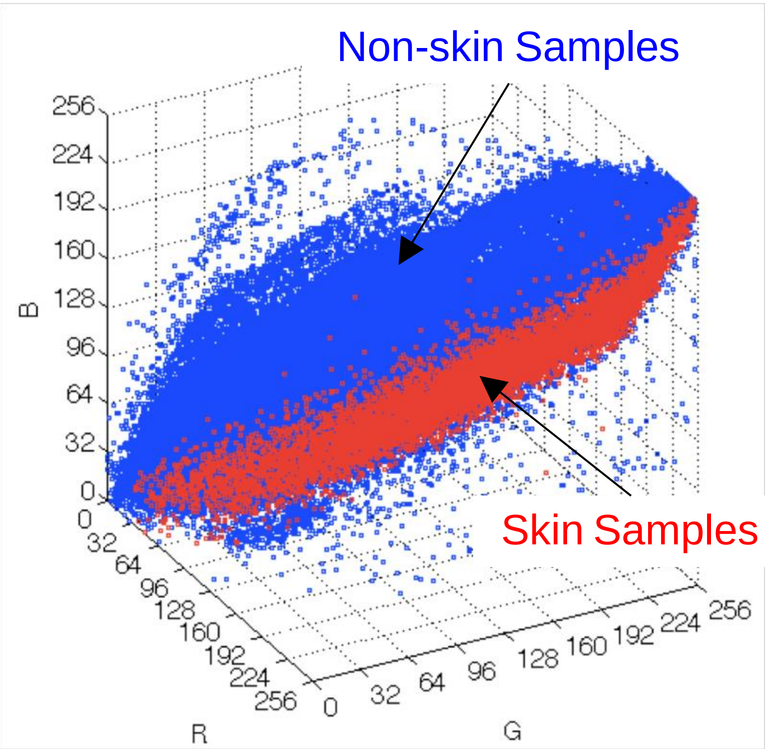


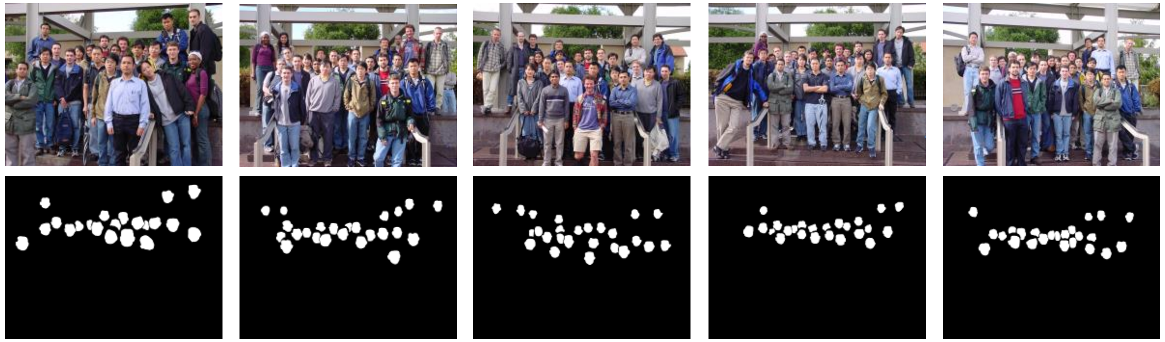
MAP detector in RGB-space

Original image



Skin color detector



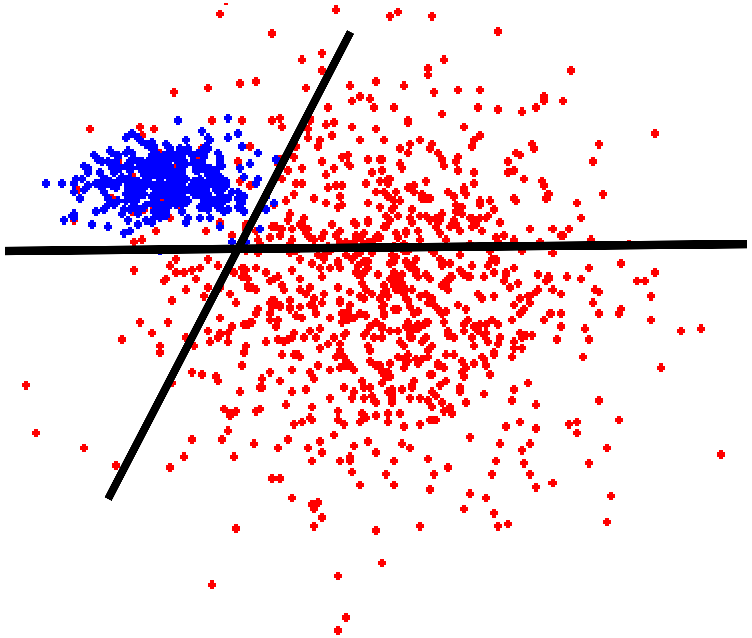


Five training images

Linear discriminant function

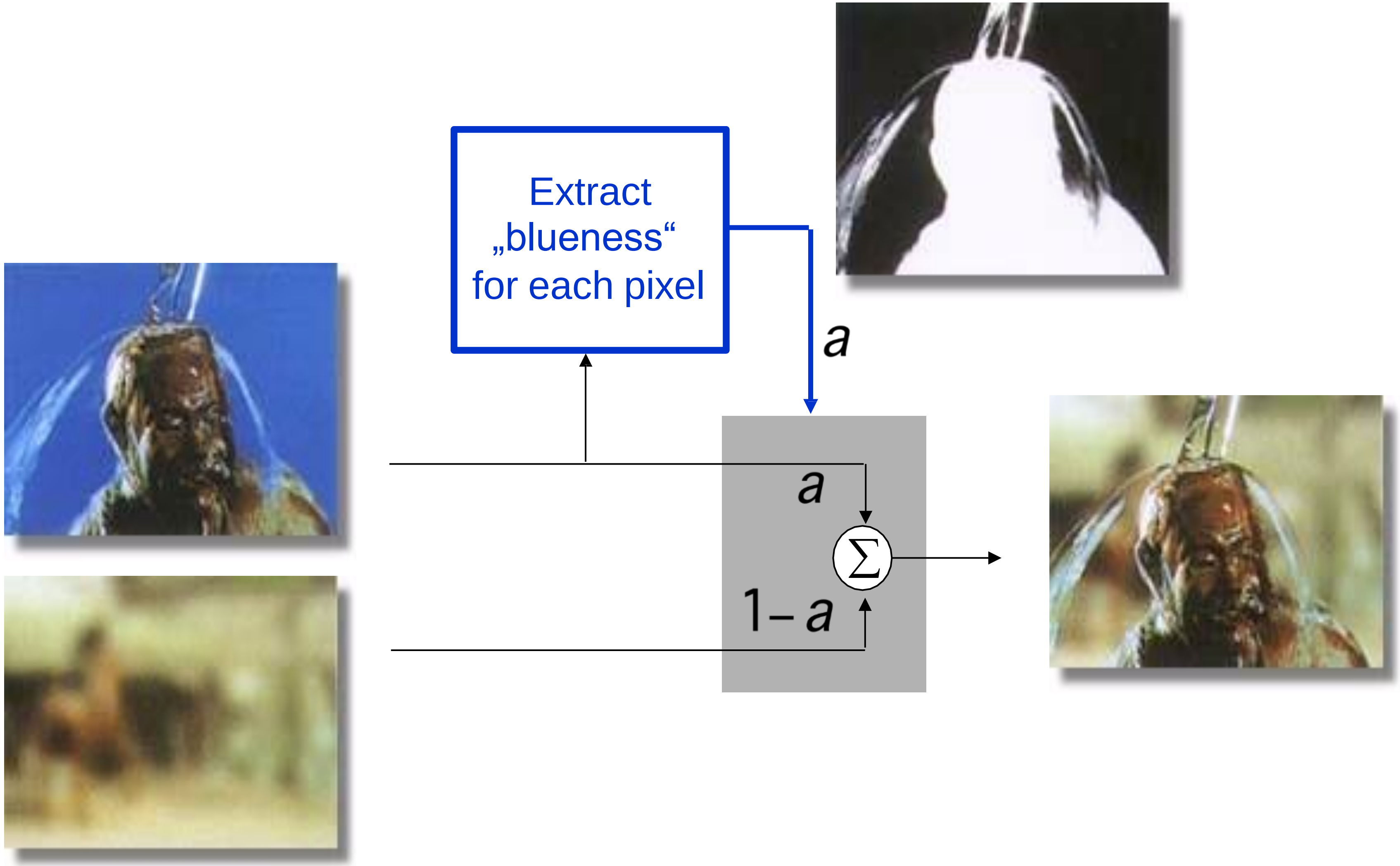
- To segment image with n components f_i, $i=1,2,\ldots,n$ into two classes, perform test

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

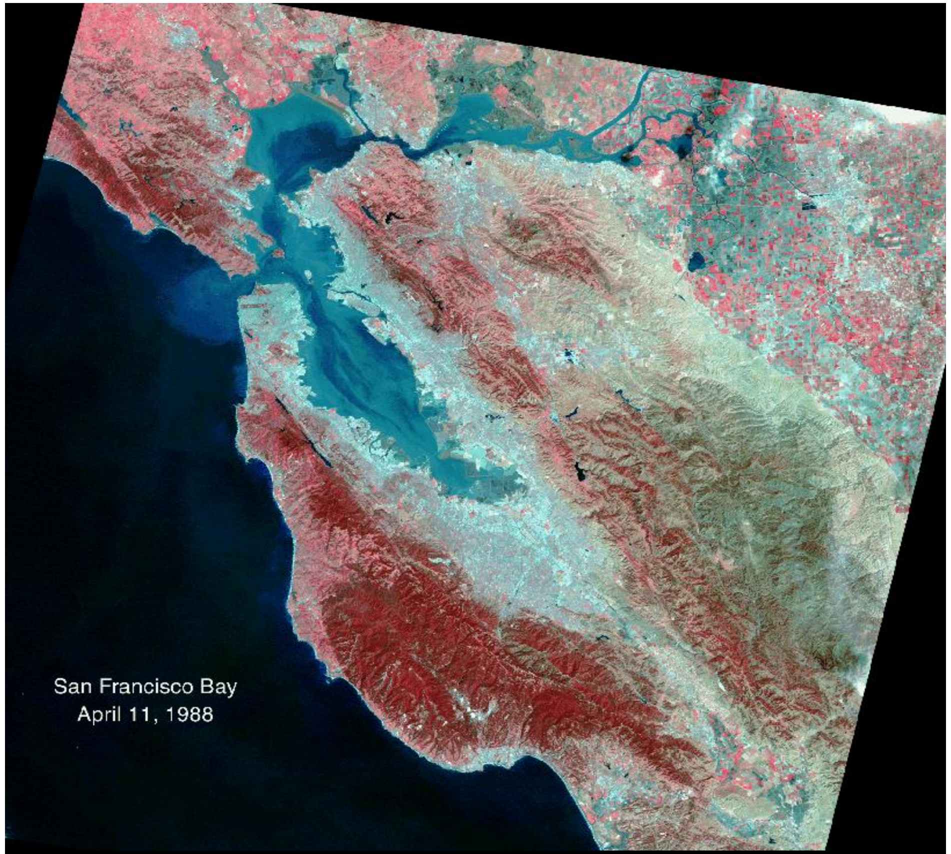


- Categories are separated by hyperplane in n-space
- Numerous techniques to determine weights w_i, $i=0,1,2,\ldots,n$, see, e.g., [Duda, Hart, Stork, 2001]
- Can be extended to the intersection of several linear discriminant functions
- Can be extended to multiple classes

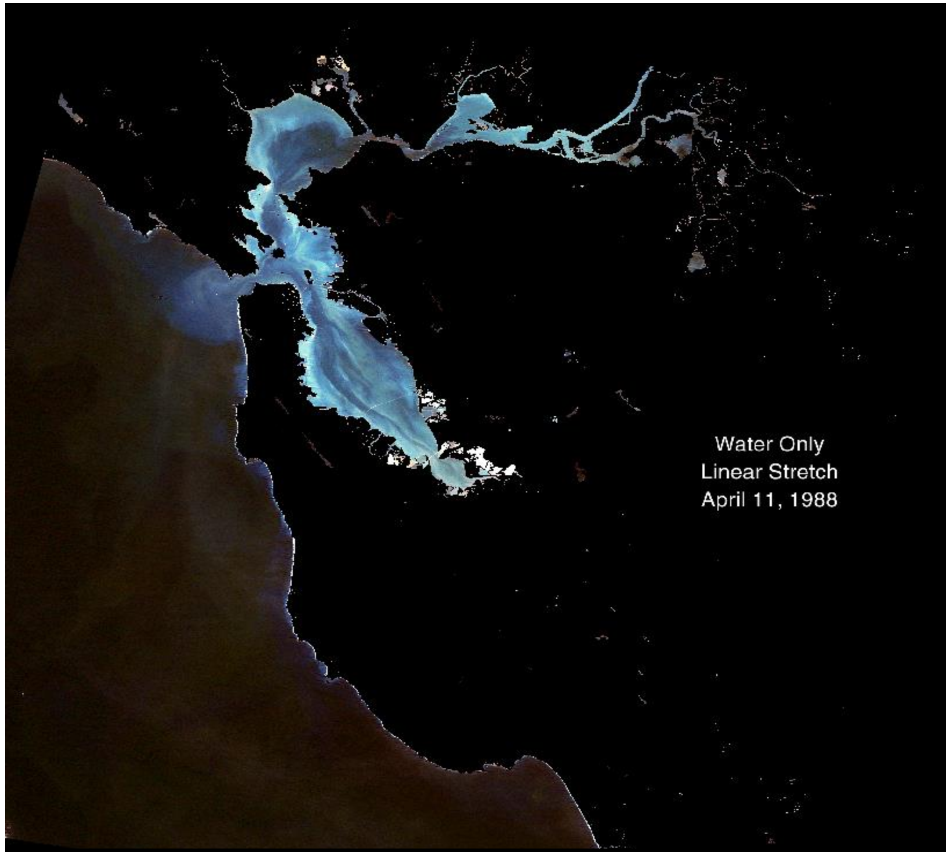
Chroma keying



Landsat image processing



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

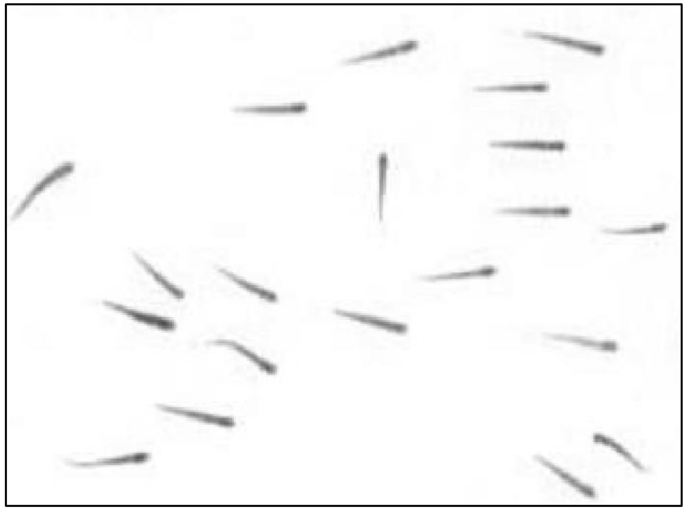


Water area segmented and enhanced to show sediments

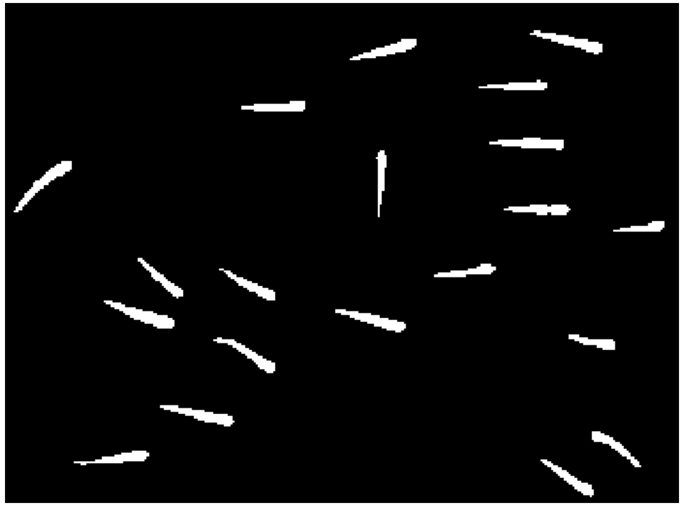
Source: US Geological Survey USGS, http://sfbay.wr.usgs.gov/

Region labeling and counting

- How many fish in this picture?



Original *Fish* image



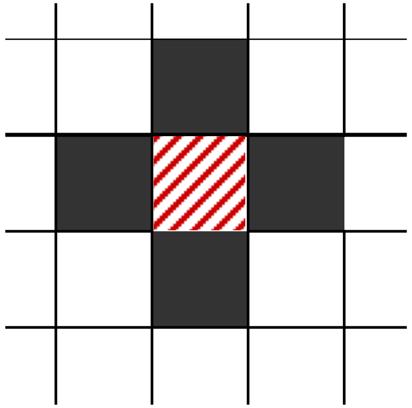
after thresholding

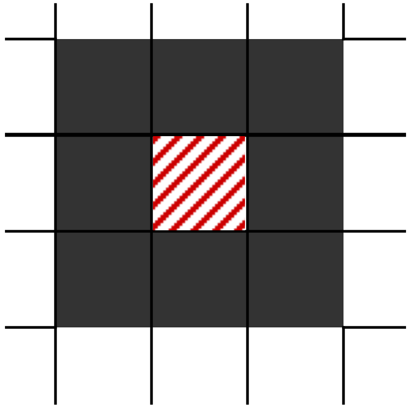
- Which pixels belong to the same object (region labeling)?
- How large is each object (region counting)?

4-connected and 8-connected neighborhoods

- Definition: 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



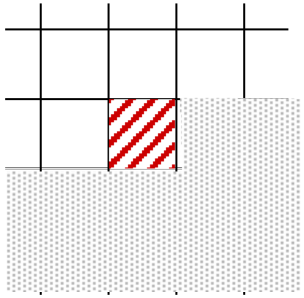


8-neighborhood

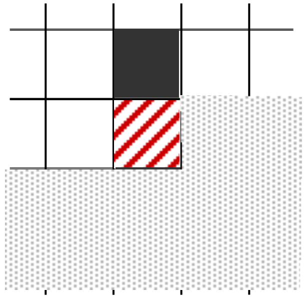
- Typically, either definition leads to the same regions, except when a region is only connected across diagonally adjacent pixels.

Region labeling algorithm (4-neighborhood)

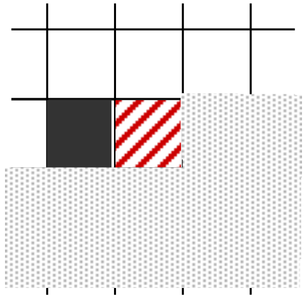
- Loop through all pixels $f[x,y]$, left to right, top to bottom
- If $f[x,y]=0$, do nothing.
- If $f[x,y]=1$, distinguish 4 cases



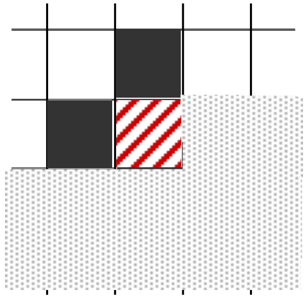
Generate new region label



Copy label from above



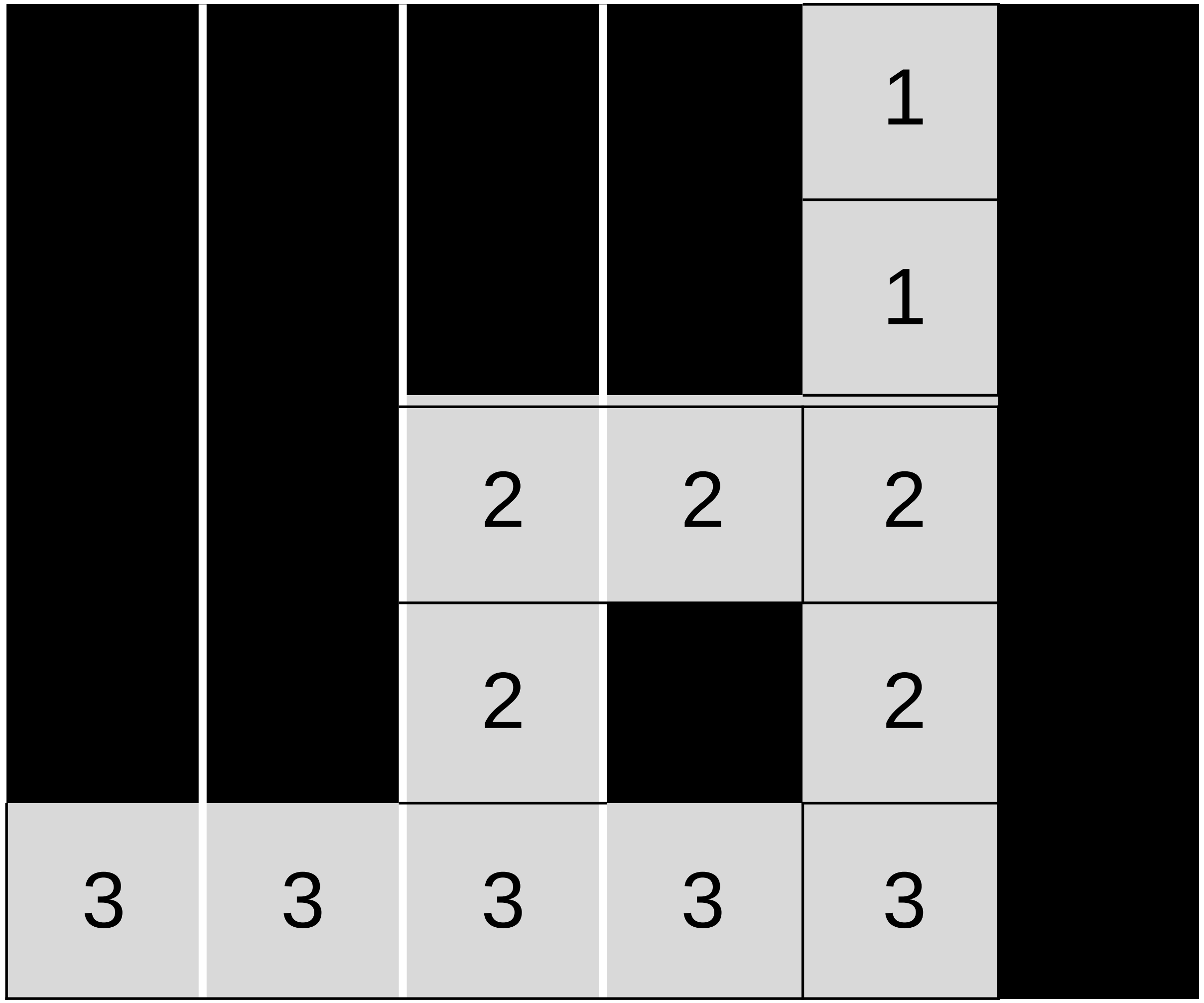
Copy label from the left



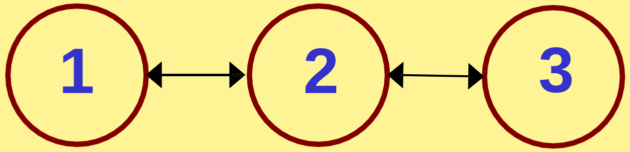
Copy label from the left. If labels above and to the left are different, store equivalence.

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

Region labeling example (4-neighborhood)

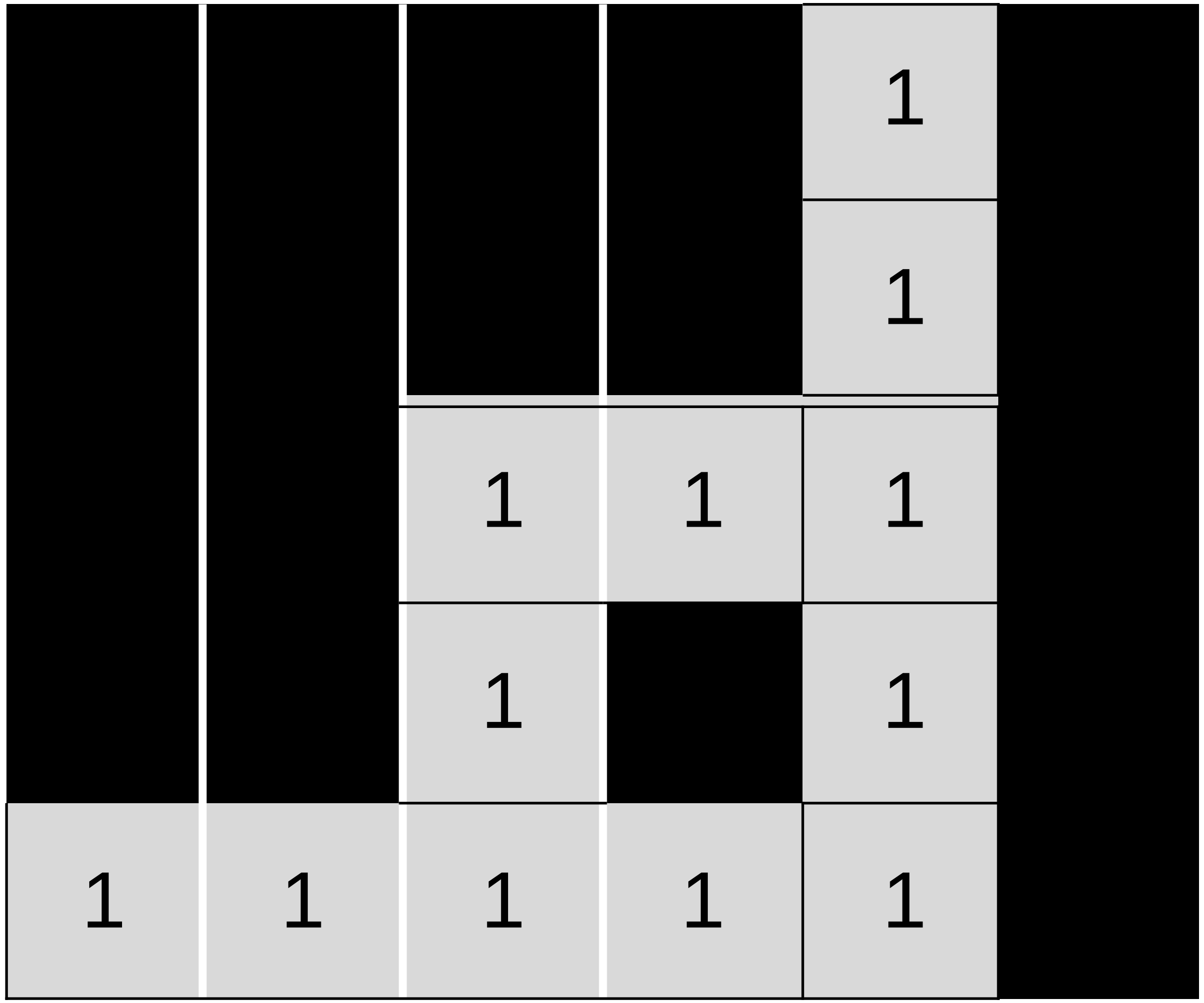


List of Region Labels



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

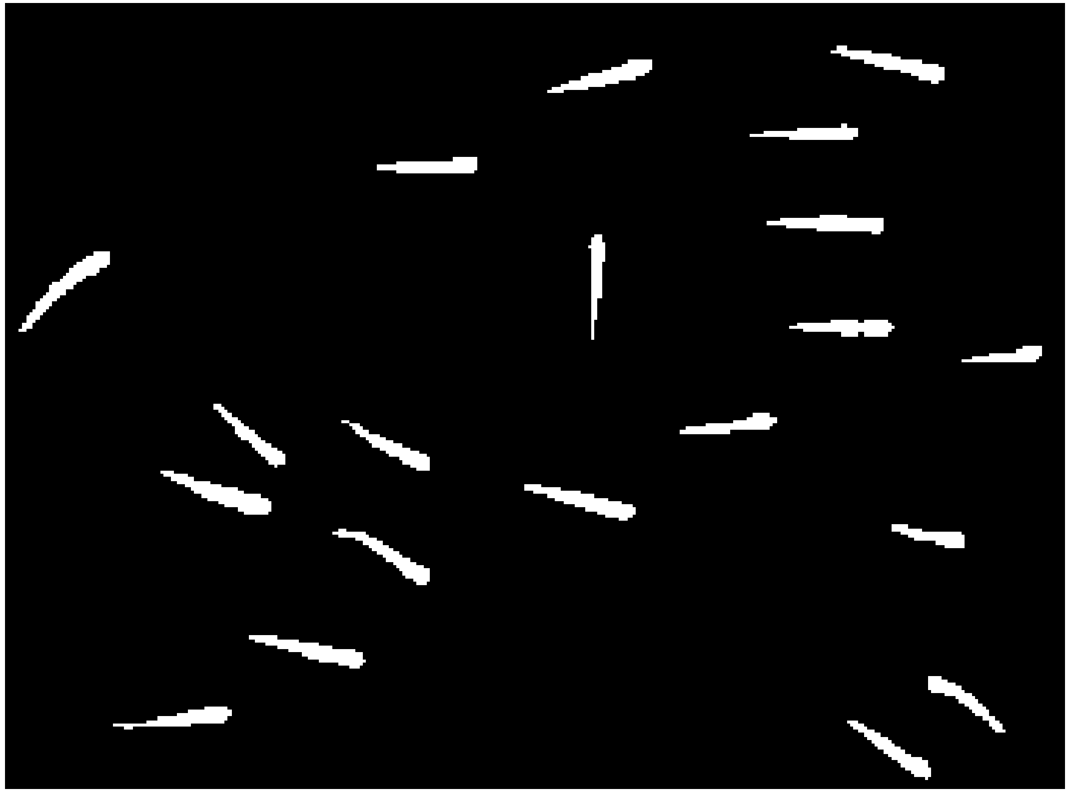
Region labeling example (4-neighborhood)



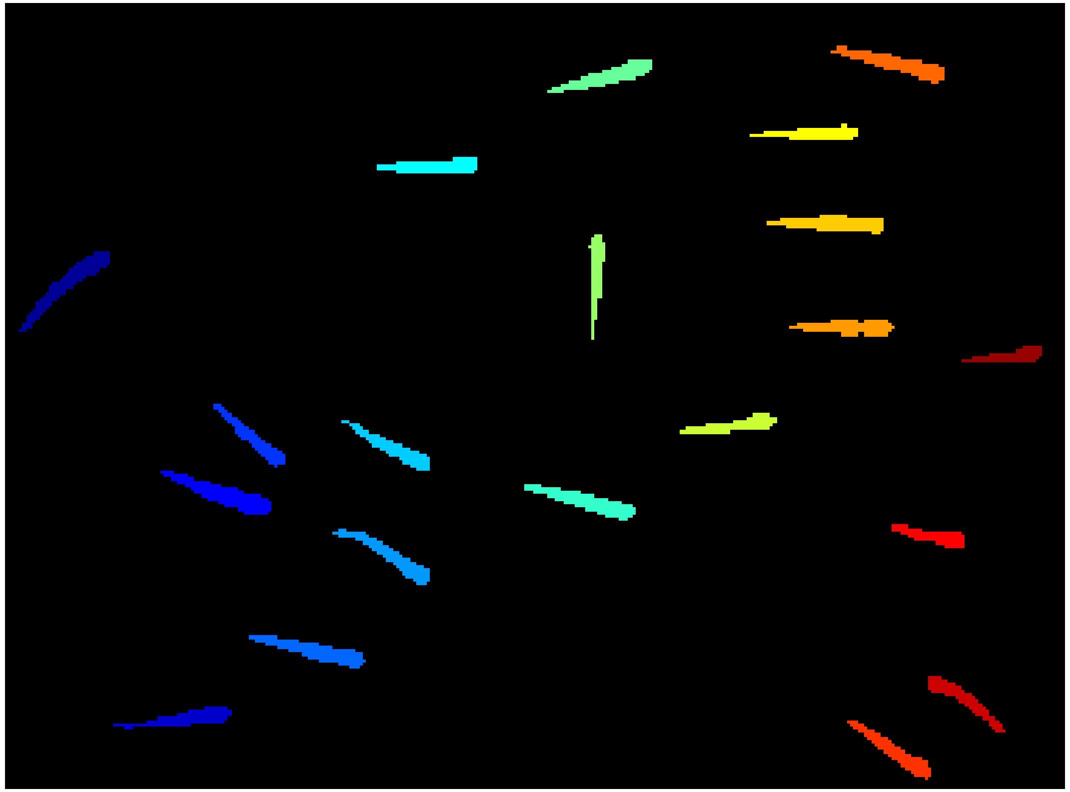
List of Region Labels

1

Example: region labeling



Thresholded image



20 labeled regions

Region counting algorithm

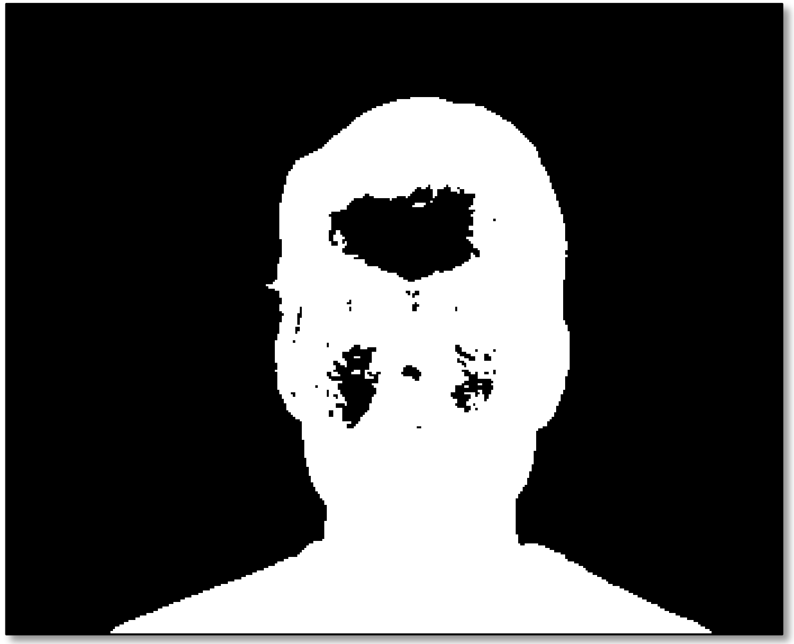
- Measures the size of each region
- Initialize $counter[label]=0$ for all $label$
- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$, increment $counter[label[x,y]]$

Small region removal

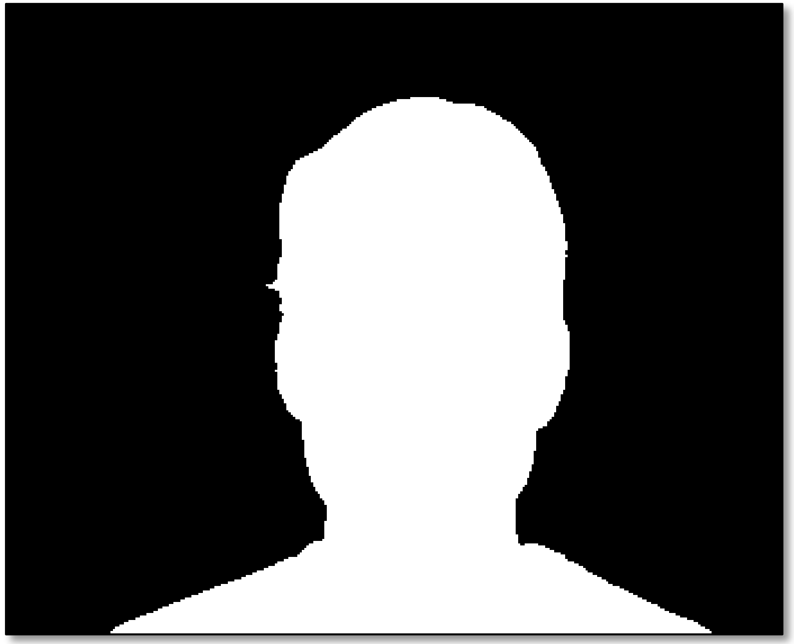
- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$ and $counter[label[x,y]]<S$, set $f[x,y]=0$
- Removes all regions smaller than S pixels

Hole filling as dual to small region removal

Mask with holes



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

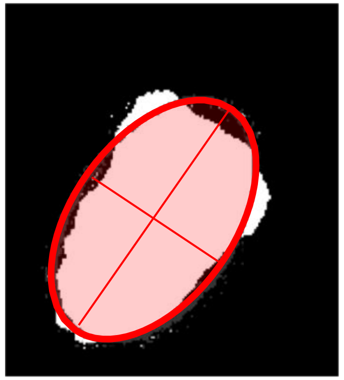


Region moments

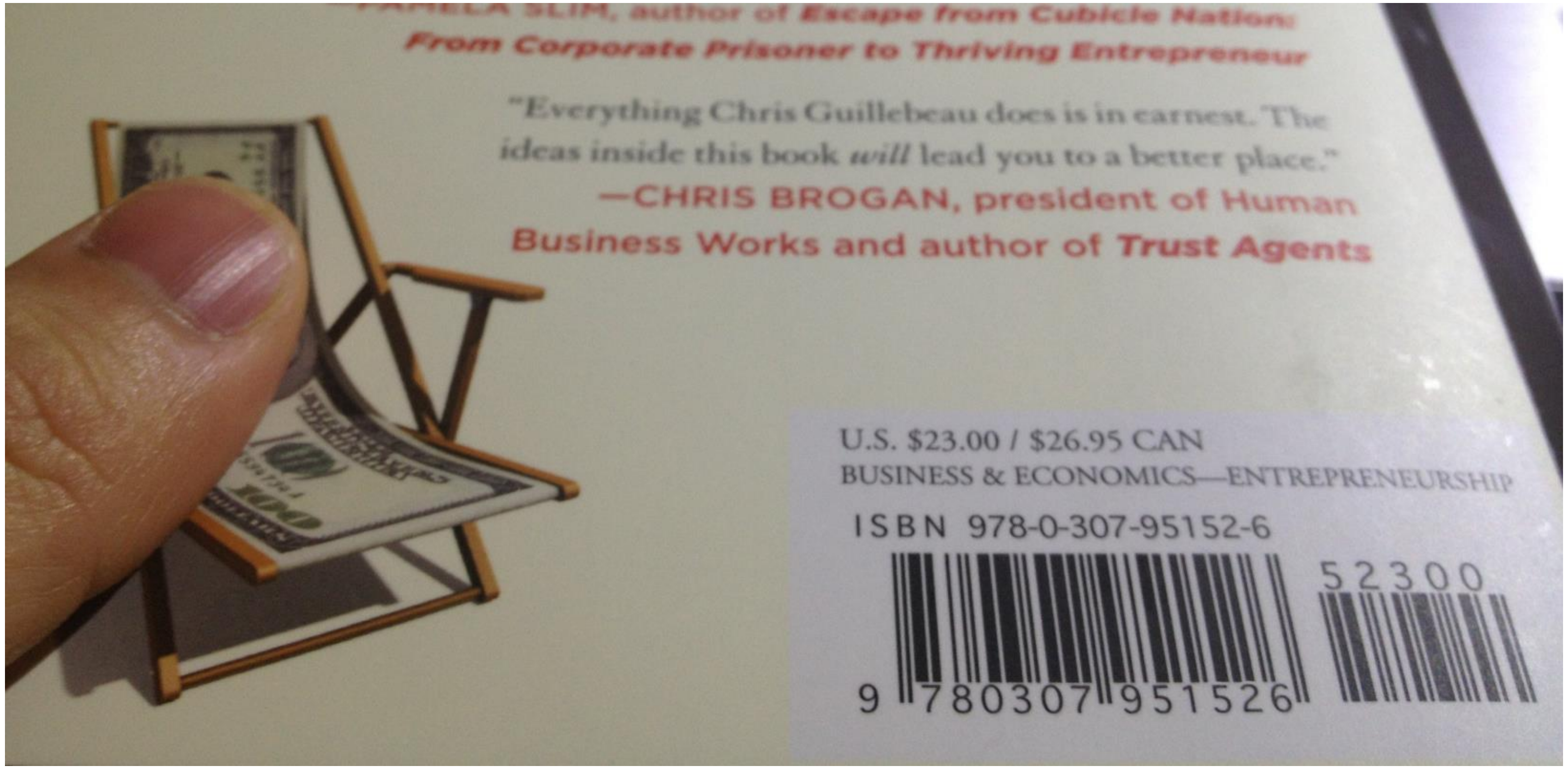
- Raw moments $$M_{pq} = \sum_{x,y \in \text{Region}} x^p y^q$$
- Central moments

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

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



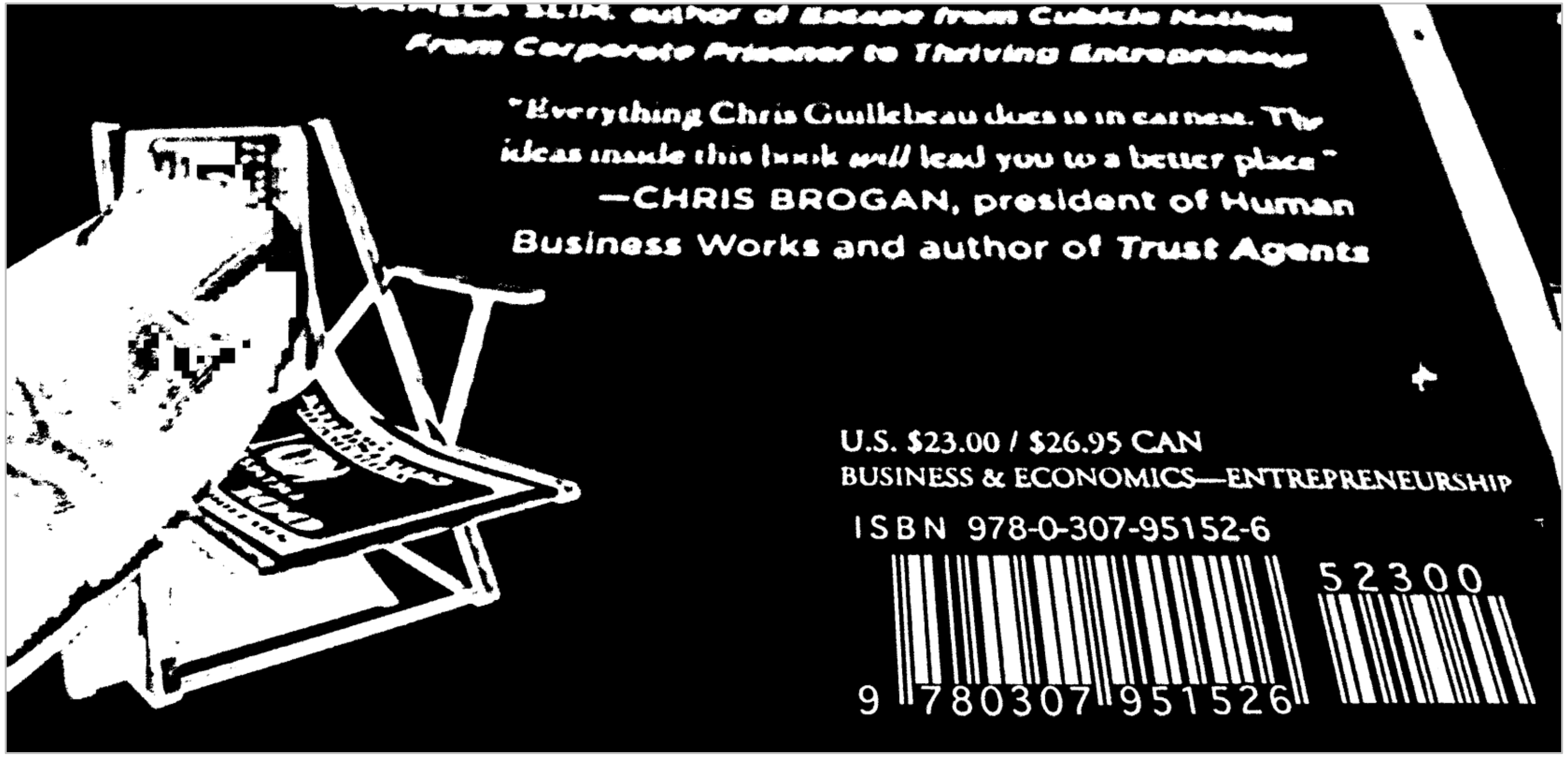
Example: Detecting bar codes



Original Image

Example: Detecting bar codes

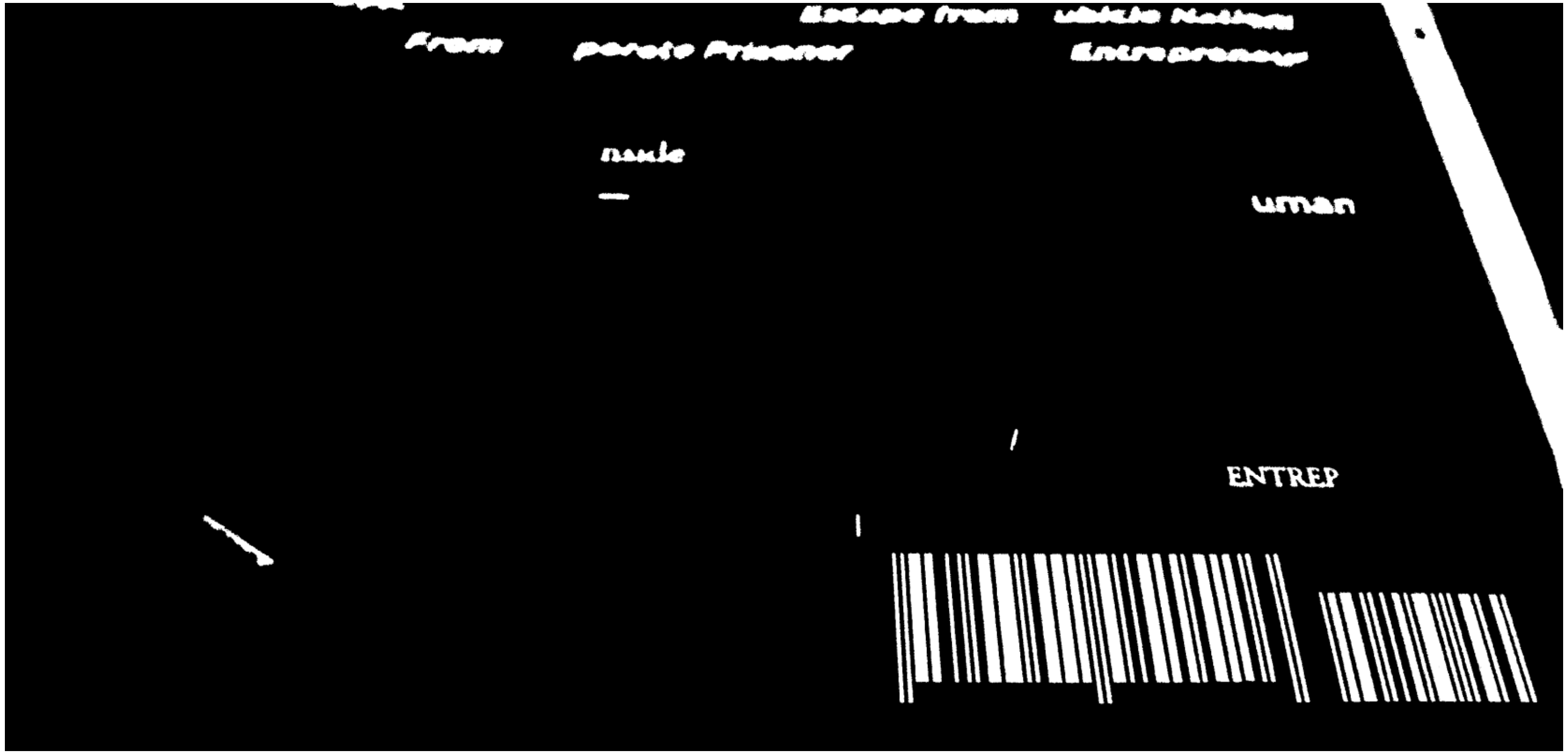
Locally adaptive thresholding



Example: Detecting bar codes

Locally adaptive thresholding

Filtering by eccentricity

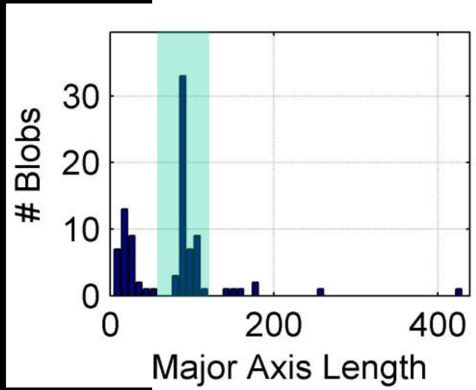


Example: Detecting bar codes

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