Edges and Binary Image Analysis

April 9th, 2015

Yong Jae Lee

UC Davis

Previously

- Filters allow local image neighborhood to influence our description and features
  - Smoothing to reduce noise
  - Derivatives to locate contrast, gradient

- Seam carving application:
  - use image gradients to measure “interestingness” or “energy”
  - remove 8-connected seams so as to preserve image’s energy

Slide credit: Kristen Grauman

Today

- Edge detection and matching
  - process the image gradient to find curves/contours
  - comparing contours

- Binary image analysis
  - blobs and regions

Slide credit: Kristen Grauman

Edge detection

- **Goal**: map image from 2d array of pixels to a set of curves or line segments or contours.
- **Why?**

Figure from D. Lowe

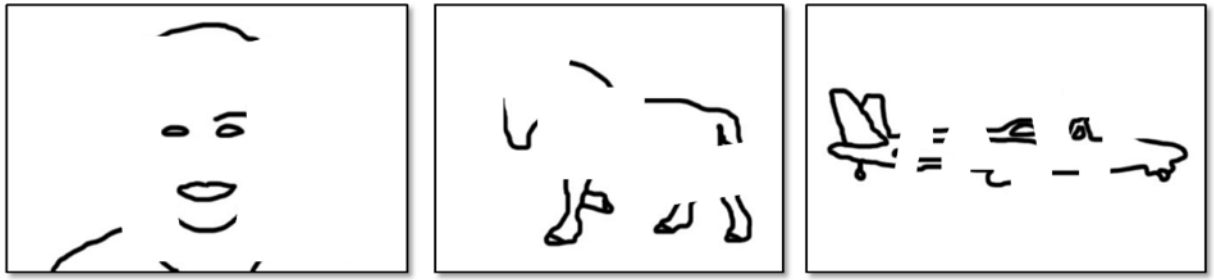


Figure from J. Shotton et al., PAMI 2007

- **Main idea**: look for strong **gradients**, post-process

Slide credit: Kristen Grauman

Gradients -> edges

Primary edge detection steps:

1. Smoothing: suppress noise
2. Edge enhancement: filter for contrast
3. Edge localization

   Determine which local maxima from filter output are actually edges vs. noise

   - Threshold, Thin

Slide credit: Kristen Grauman

Thresholding

- Choose a threshold value t
- Set any pixels less than t to zero (off)
- Set any pixels greater than or equal to t to one (on)

Slide credit: Kristen Grauman

Original image

Slide credit: Kristen Grauman

Gradient magnitude image

Thresholding gradient with a lower threshold

Slide credit: Kristen Grauman

Thresholding gradient with a higher threshold

Slide credit: Kristen Grauman

Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Non-maximum suppression**:
  - Thin wide “ridges” down to single pixel width
- **Linking and thresholding** (**hysteresis**):
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB: **`edge(image, ‘canny’);`**
- **`>>help edge`**

Slide credit: David Lowe, Fei-Fei Li

The Canny edge detector

original image (Lena)

Slide credit: Steve Seitz

The Canny edge detector

gradient magnitude

Slide credit: Kristen Grauman

Compute Gradients (DoG)

X-Derivative of Gaussian

Y-Derivative of Gaussian

Gradient Magnitude

Slide credit: Svetlana Lazebnik

The Canny edge detector

gradient magnitude

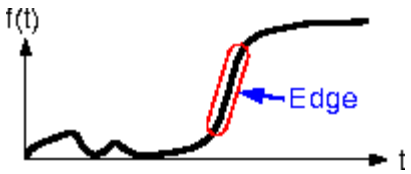
Slide credit: Kristen Grauman

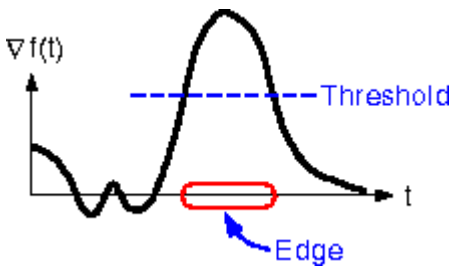
The Canny edge detector

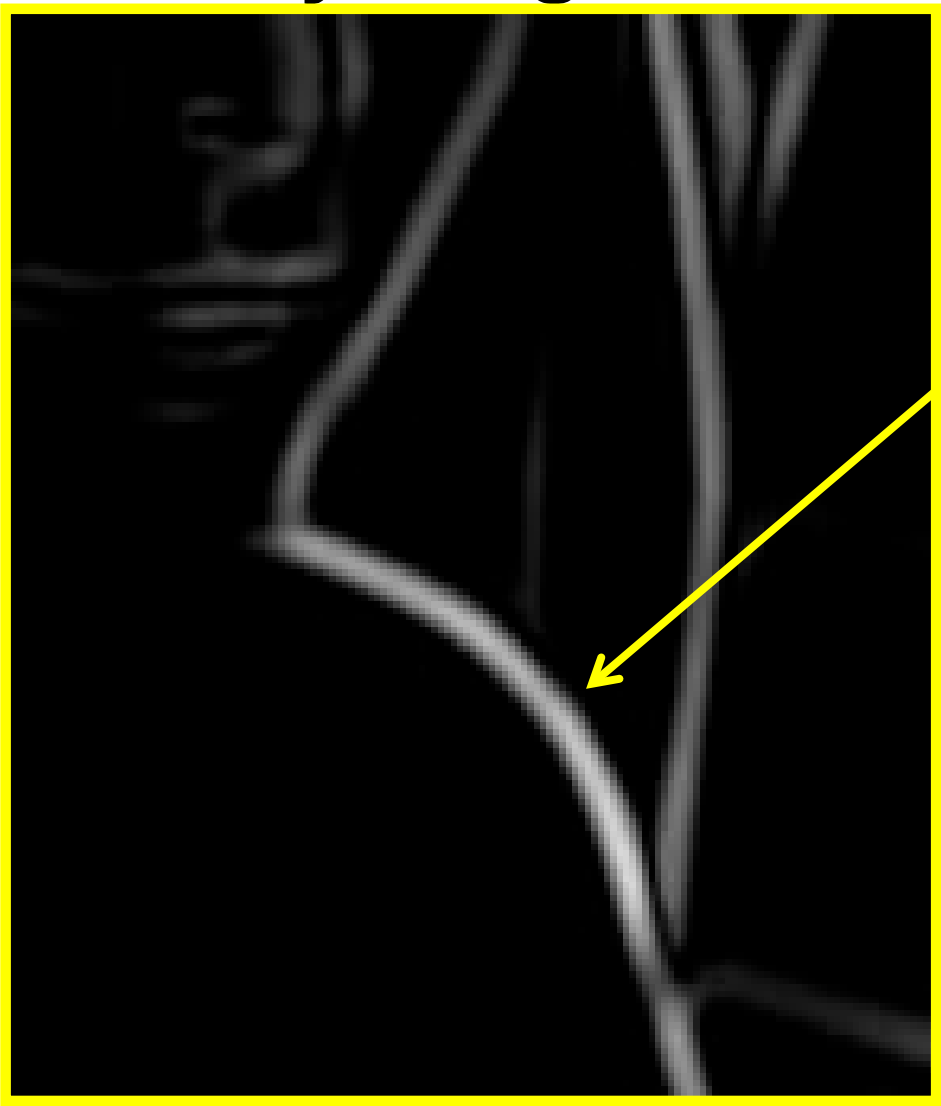
thresholding

Slide credit: Kristen Grauman

The Canny edge detector



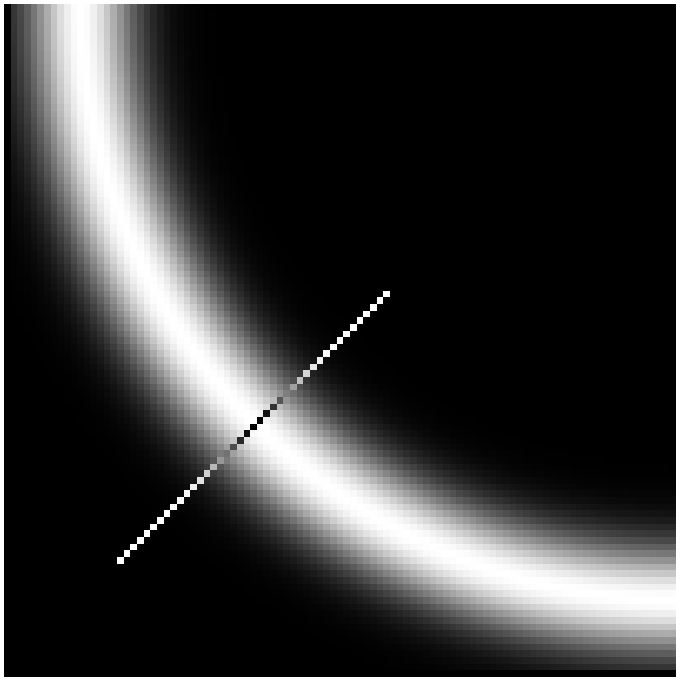


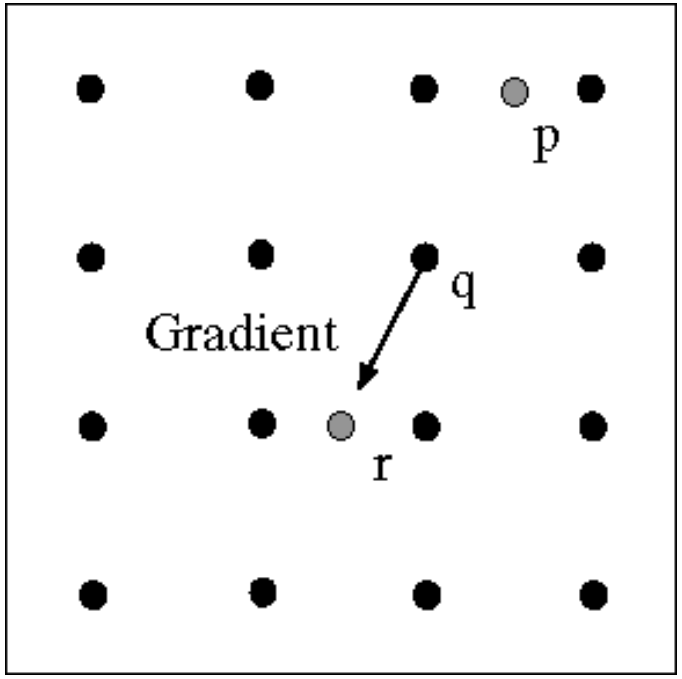


How to turn these thick regions of the gradient into curves?

Slide credit: Kristen Grauman

Non-maximum suppression





Check if pixel is local maximum along gradient direction

Select single max across width of the edge

Requires checking interpolated pixels p and r

Slide credit: Kristen Grauman

The Canny edge detector

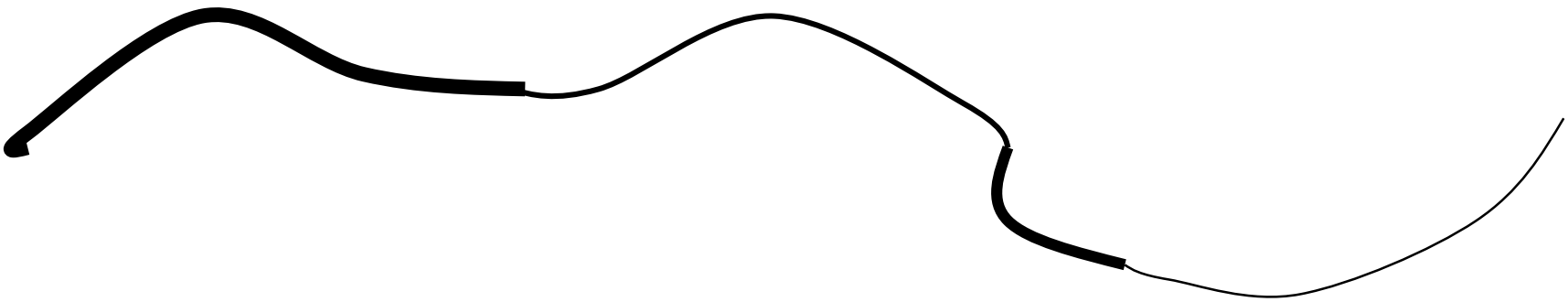
Problem: pixels along this edge didn't survive the thresholding

thinning
(non-maximum suppression)

Slide credit: Kristen Grauman

Hysteresis thresholding

- Use a high threshold to start edge curves, and a low threshold to continue them.



Slide credit: Steve Seitz

Hysteresis thresholding

original image

high threshold (strong edges)

low threshold (weak edges)

hysteresis threshold

Slide credit: Fei-Fei Li

Hysteresis thresholding

http://users.ecs.soton.ac.uk/msn/book/new_demo/thresholding/

Slide credit: Kristen Grauman

Recap: Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Non-maximum suppression**:
  - Thin wide “ridges” down to single pixel width
- **Linking and thresholding** (**hysteresis**):
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB: **`edge(image, ‘canny’);`**
- **`>>help edge`**

Slide credit: David Lowe, Fei-Fei Li

Low-level edges vs. perceived contours

Background

Texture

Shadows

Slide credit: Kristen Grauman

Low-level edges vs. perceived contours

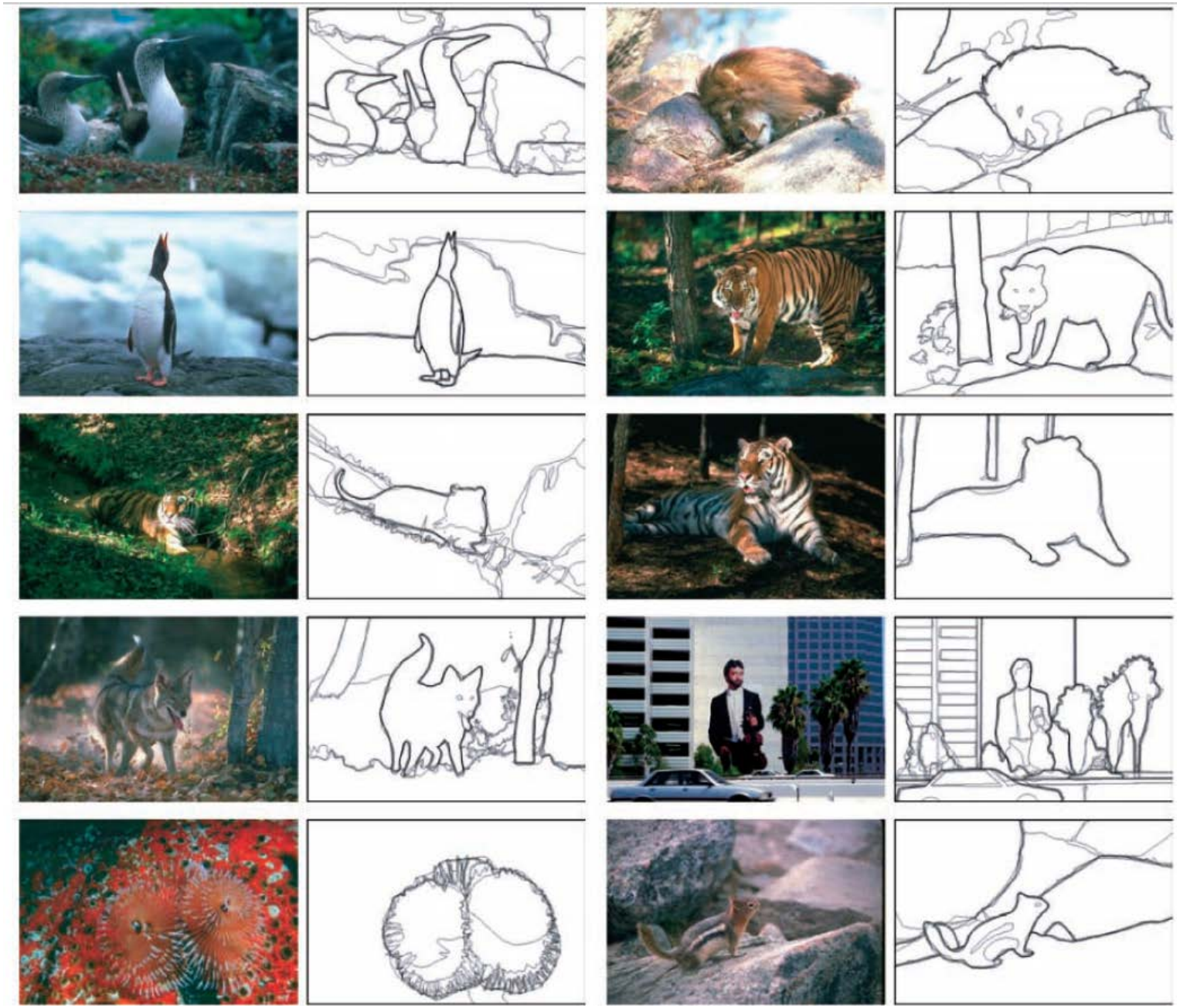
image **human segmentation** **gradient magnitude**

- Berkeley segmentation database: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Slide credit: Svetlana Lazebnik

Learn from humans which combination of features is most indicative of a “good” contour?

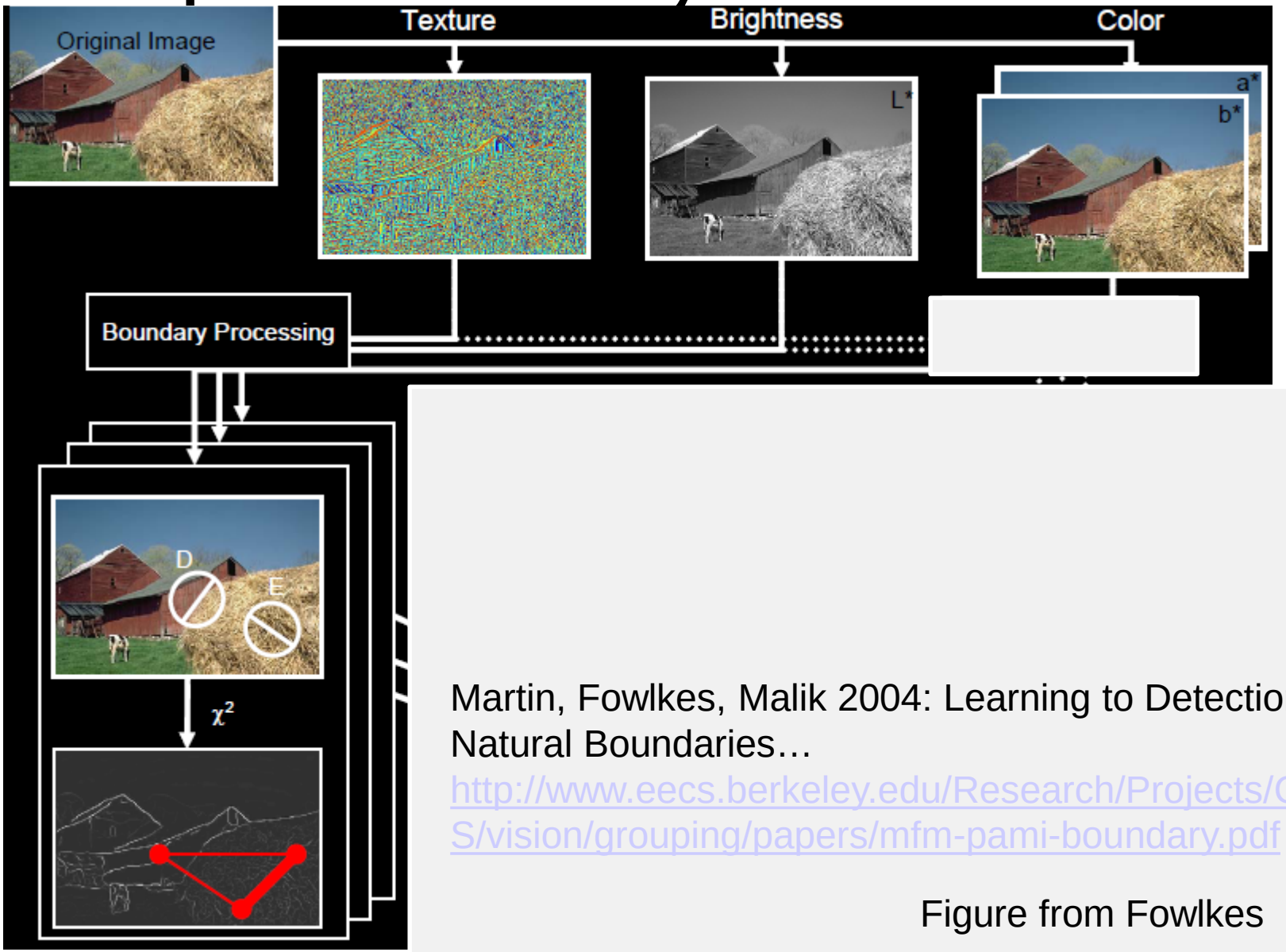
[D. Martin et al. PAMI 2004]



Human-marked segment boundaries

Slide credit: Kristen Grauman

pB boundary detector



Martin, Fowlkes, Malik 2004: Learning to Detection Natural Boundaries…
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/papers/mfm-pami-boundary.pdf

Figure from Fowlkes

pB Boundary Detector

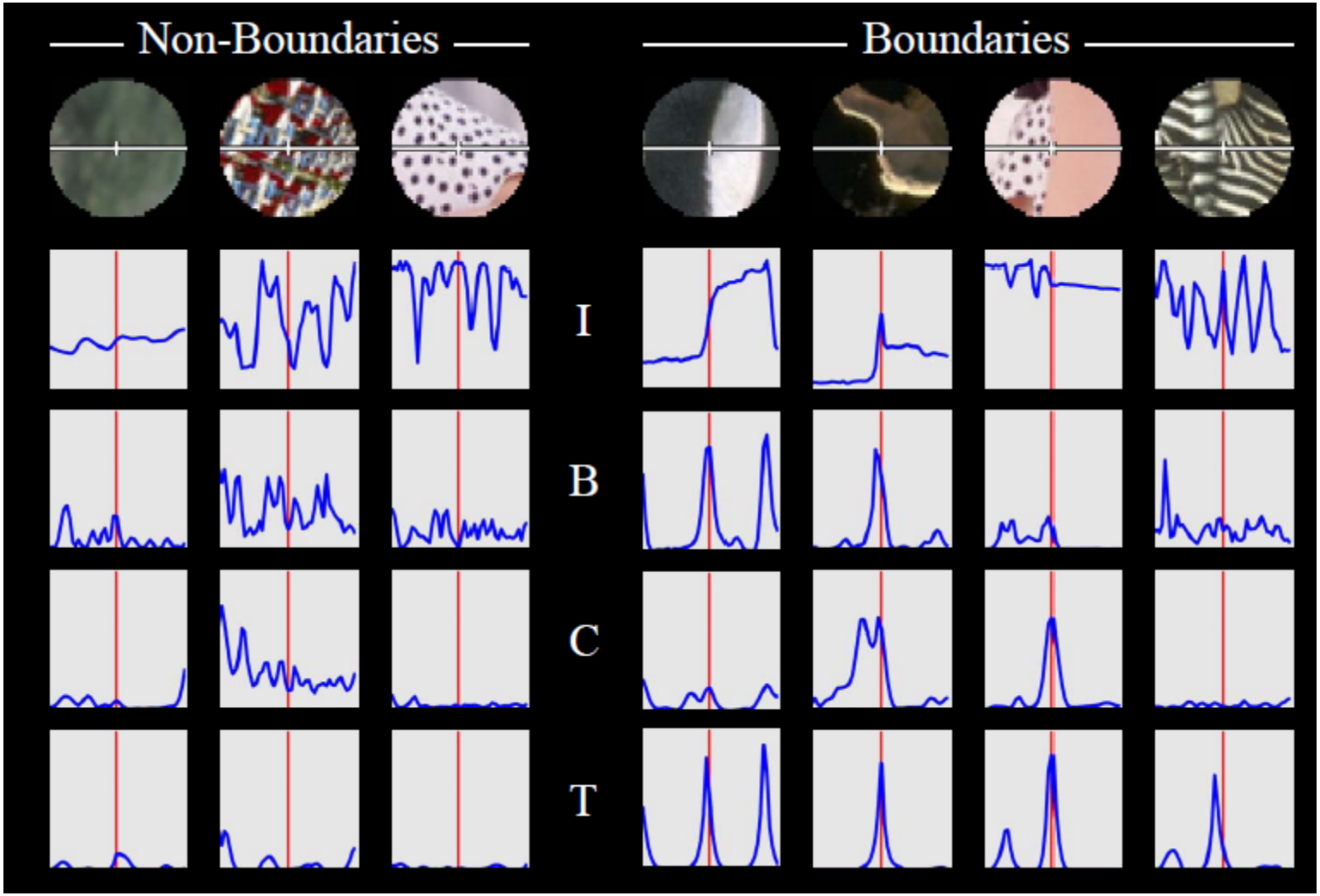
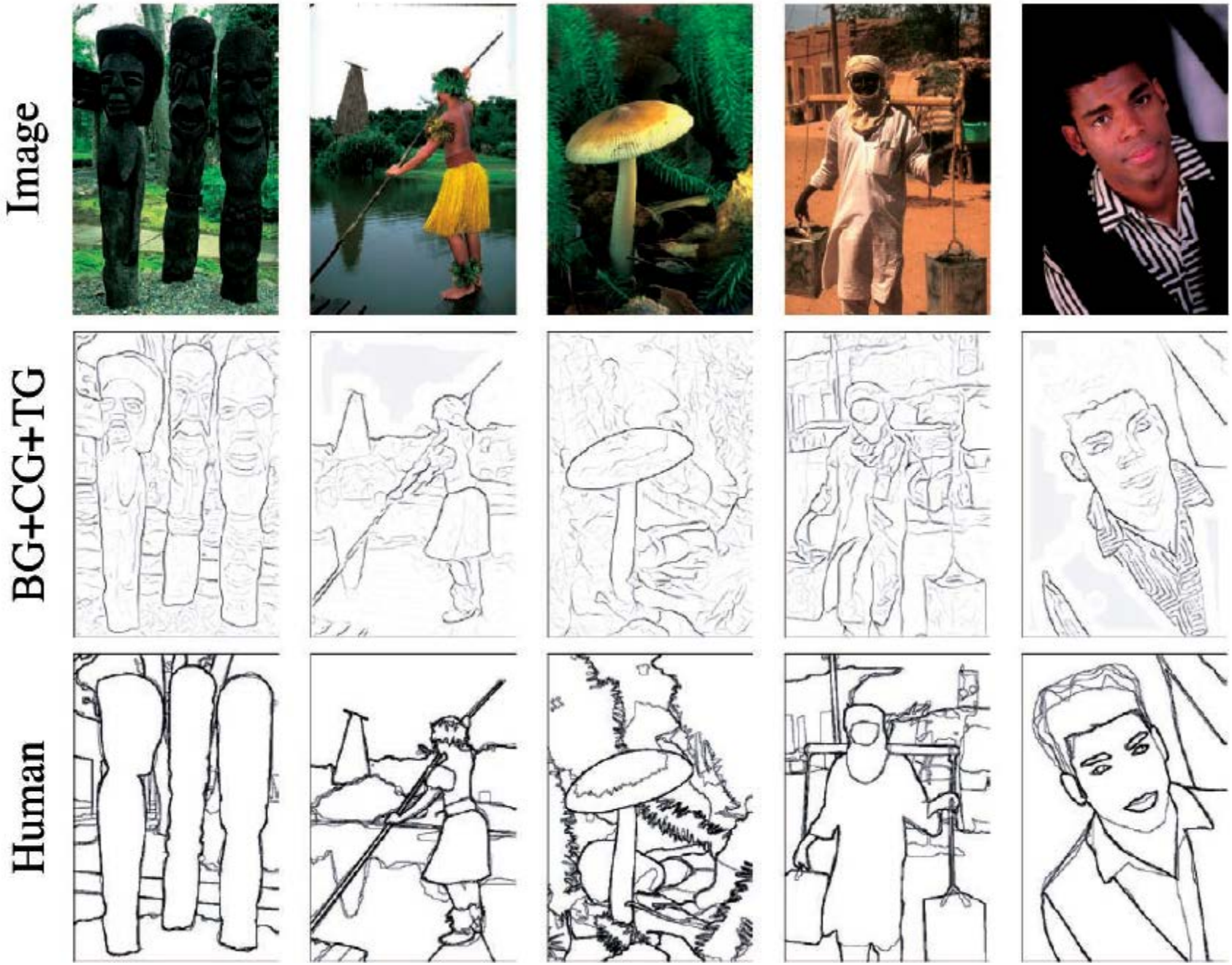


Figure from Fowlkes



[D. Martin et al. PAMI 2004]

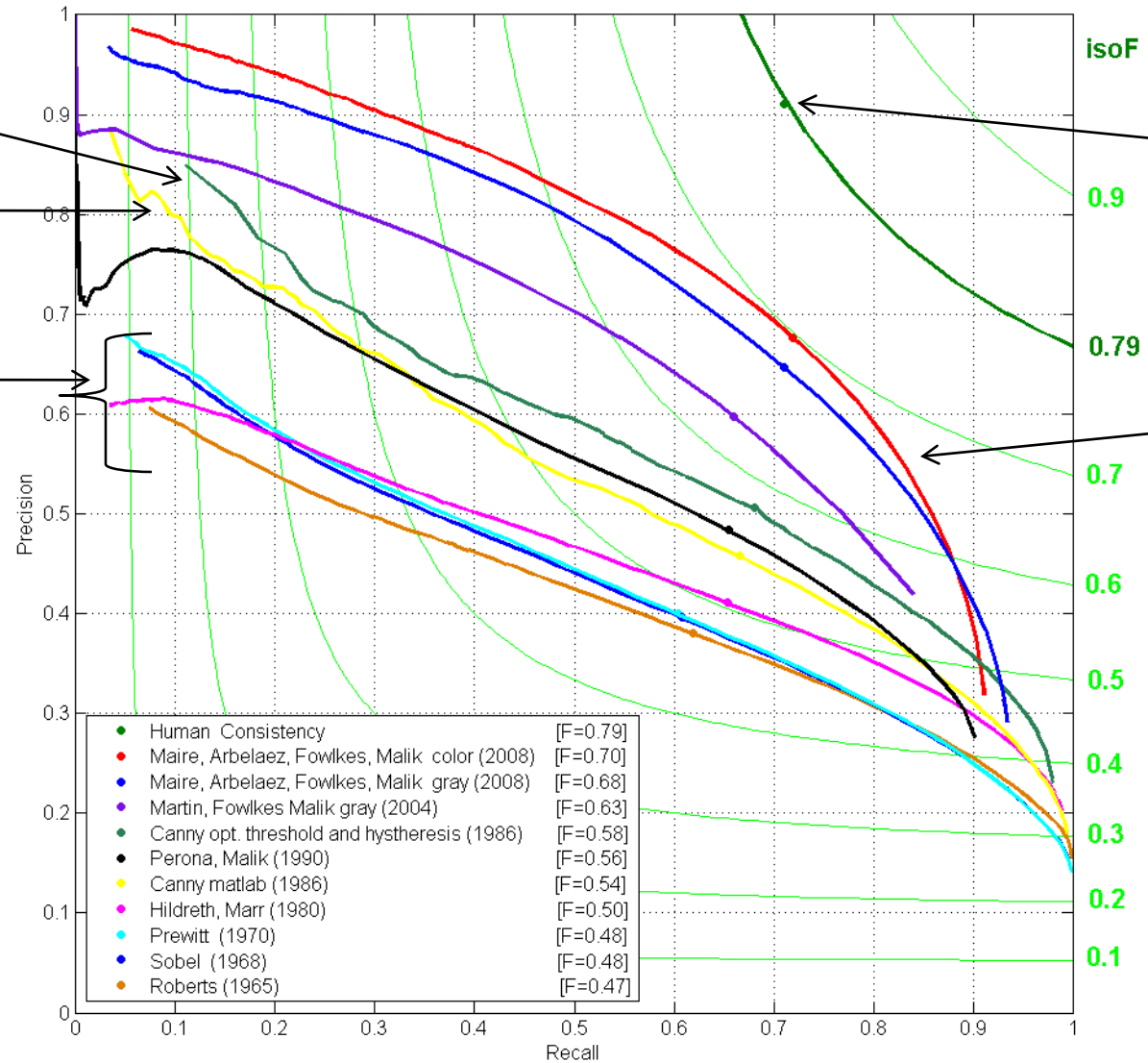
Slide credit: Kristen Grauman

State-of-the-Art in Contour Detection

Canny+opt thresholds

Canny

Prewitt, Sobel, Roberts



Human agreement

Learned with combined features

Source: Jitendra Malik: http://www.cs.berkeley.edu/~malik/malik-talks-ptrs.html

Slide credit: Adapted from Kristen Grauman

Global pB boundary detector

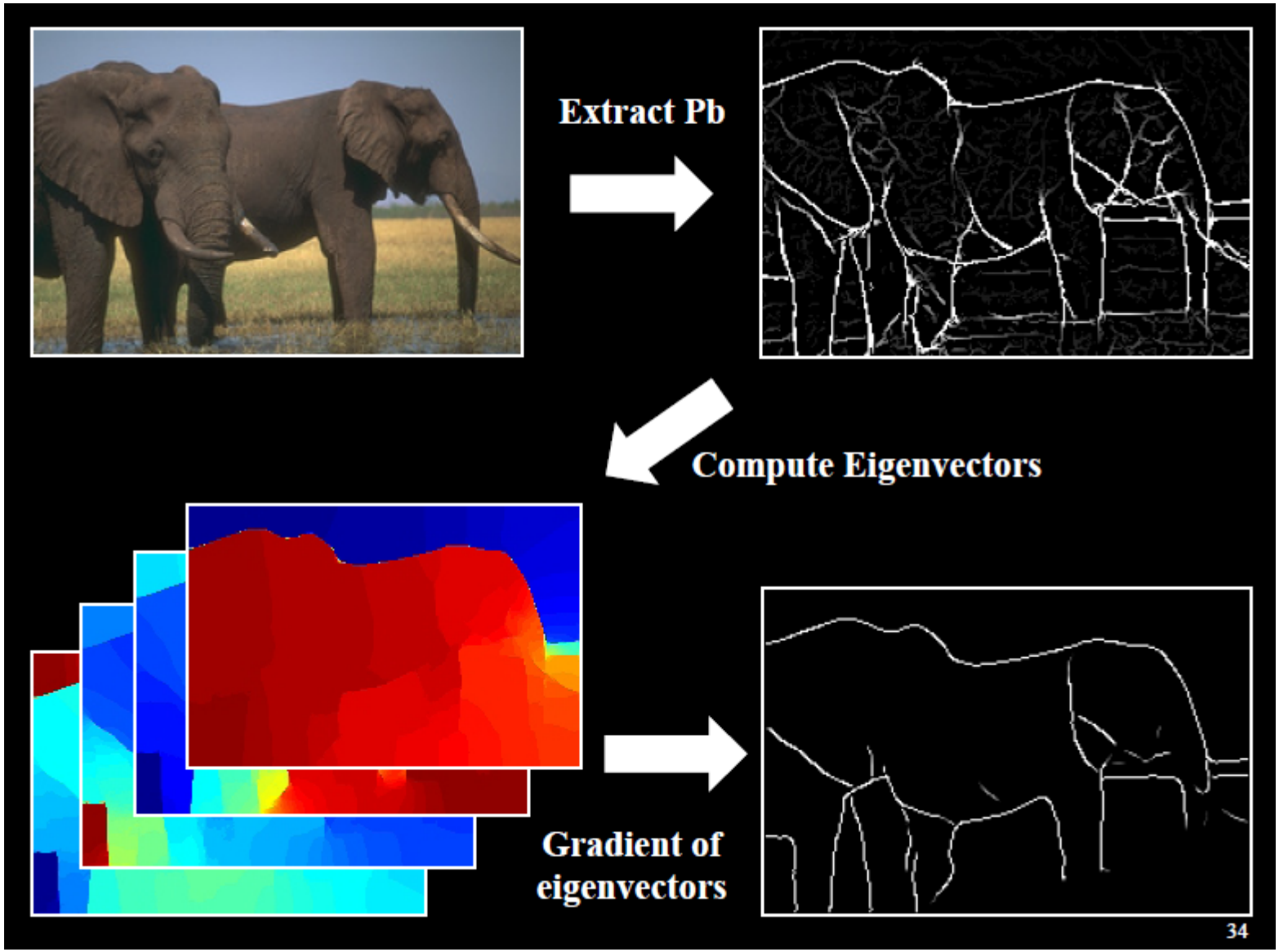
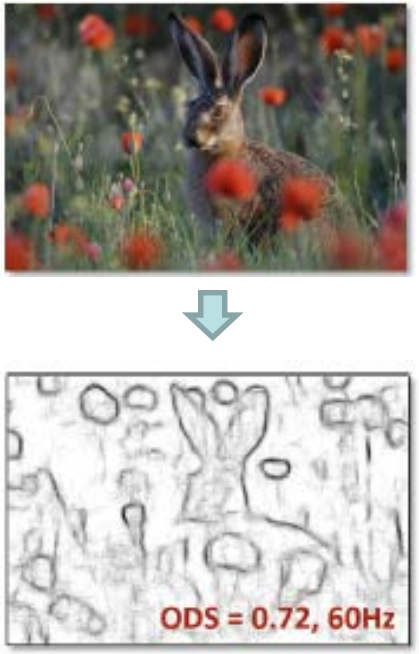
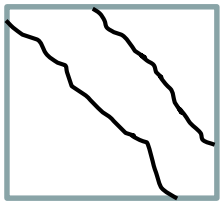


Figure from Fowlkes

Edge Detection with Structured Random Forests (Dollar Zitnick ICCV 2013)

- Goal: quickly predict whether each pixel is an edge
- Insights
  - Predictions can be learned from training data
  - Predictions for nearby pixels should not be independent
- Solution
  - Train structured random forests to split data into patches with similar boundaries based on features
  - Predict boundaries at patch level, rather than pixel level, and aggregate (average votes)





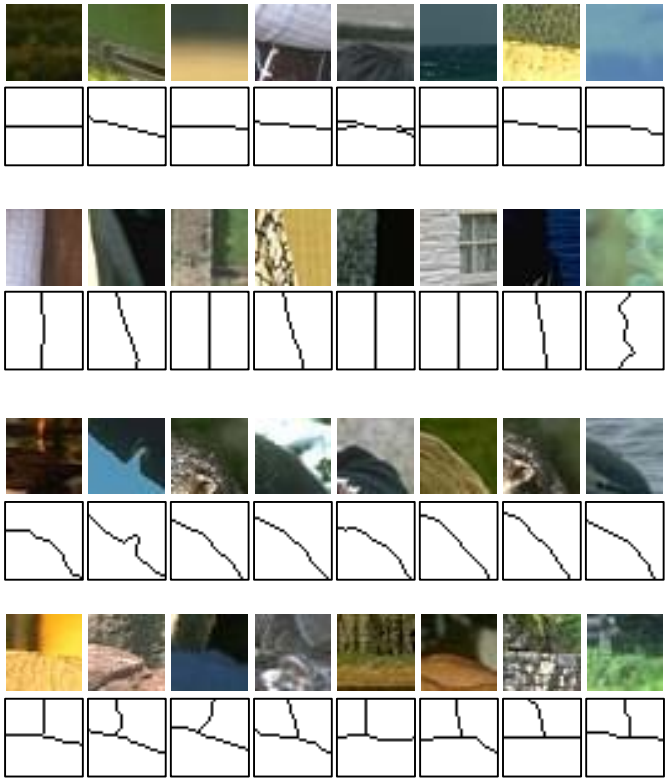
Boundaries in patch

Slide credit: Derek Hoiem

http://research.microsoft.com/pubs/202540/DollarICCV13edges.pdf

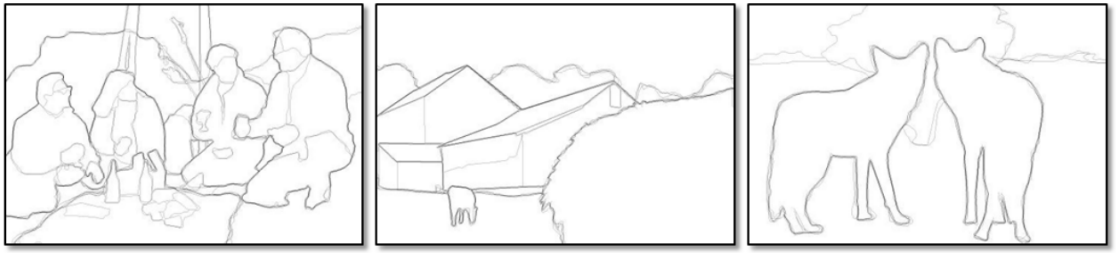
Edge Detection with Structured Random Forests

Training data



Input images

Ground truth



Output

P. Dollar and L. Zitnick, Structured forests for fast edge detection, ICCV 2013

Slide credit: Svetlana Lazebnik

State of edge detection

- Local edge detection is mostly solved
  - Intensity gradient, color, texture
- Work on RGB-D edge detection is currently more active
- Often used in combination with object detectors or region classifiers

Slide credit: Derek Hoiem

Today

- Edge detection and matching
  - process the image gradient to find curves/contours
  - comparing contours

- Binary image analysis
  - blobs and regions

Slide credit: Kristen Grauman

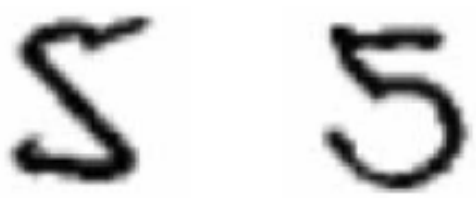


Fig. 1. Examples of two handwritten digits. In terms of pixel-to-pixel comparisons, these two images are quite different, but to the human observer, the shapes appear to be similar.

Slide credit: Kristen Grauman

Figure from Belongie et al.

Chamfer distance

- Average distance to nearest feature

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

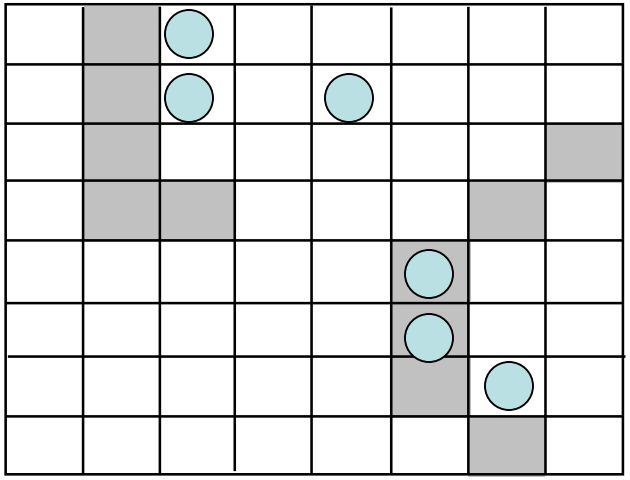
$\boldsymbol{I} =$ Set of points in image

$\boldsymbol{T} =$ Set of points on (shifted) template

$d_I(t) =$ Minimum distance between point t and some point in $\boldsymbol{I}$

Slide credit: Kristen Grauman

Chamfer distance



$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

Slide credit: Kristen Grauman

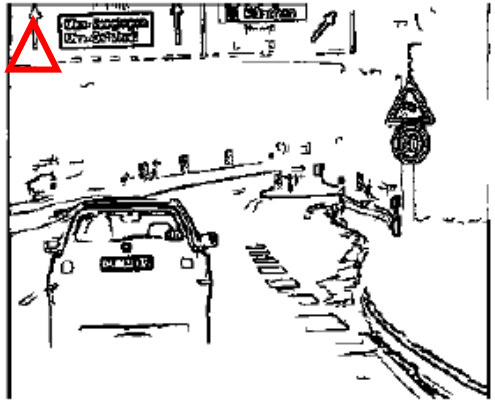
Chamfer distance

- Average distance to nearest feature

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

How is the measure different than just filtering with a mask having the shape points?

How expensive is a naïve implementation?

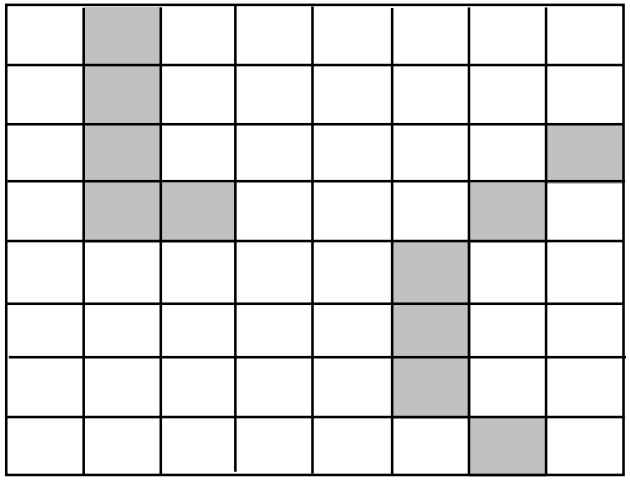


Edge image

Slide credit: Kristen Grauman

Distance transform

Image features (2D)



Distance Transform

1	0	1	2	3	4	3	2
1	0	1	2	3	3	2	1
1	0	1	2	3	2	1	0
1	0	0	1	2	1	0	1
2	1	1	2	1	0	1	2
3	2	2	2	1	0	1	2
4	3	3	2	1	0	1	2
5	4	4	3	2	1	0	1

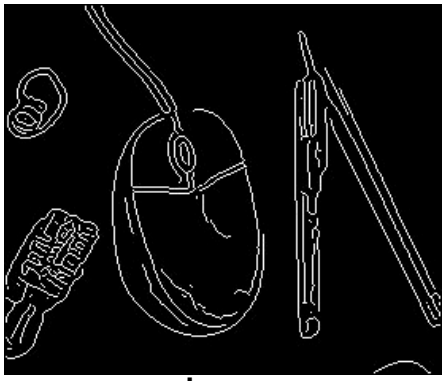
Distance Transform is a function $D(\cdot)$ that for each image pixel p assigns a non-negative number $D(p)$ corresponding to distance from p to the nearest feature in the image I

Features could be edge points, foreground points,…

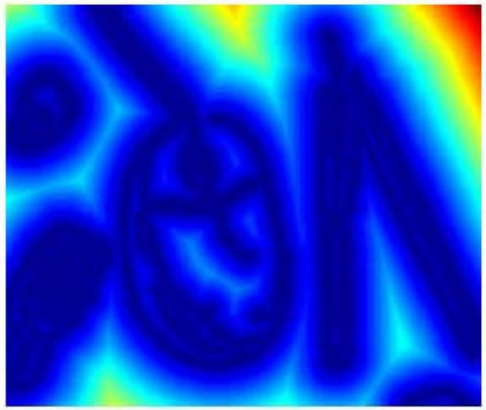
Slide credit: Yuri Boykov

Distance transform

original



edges



distance transform

Value at (x,y) tells how far that position is from the nearest edge point (or other binary mage structure)

```
>> help bwdist
```


Slide credit: Kristen Grauman

Distance transform (1D)

Two pass O(n) algorithm for 1D L_1 norm

1. <u>Initialize</u>: For all j
   $D[j] \leftarrow 1_{\mathbf{P}}[j]$ // 0 if j is in **P**, infinity otherwise

Image features (edges)

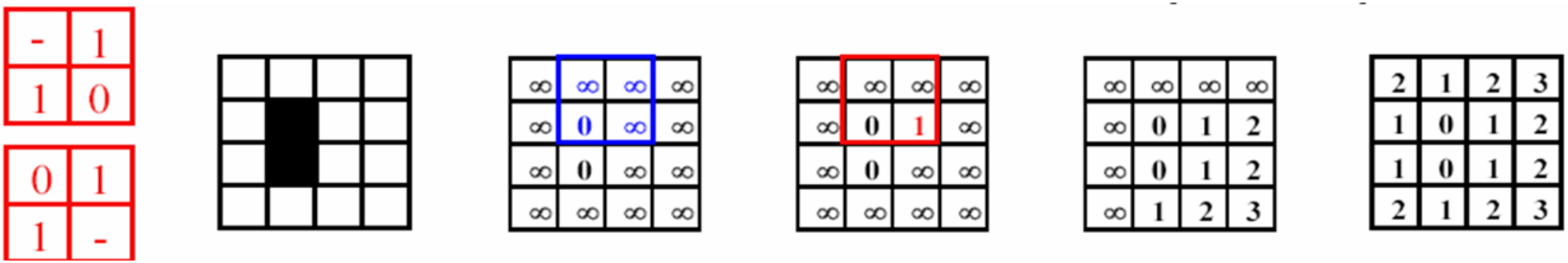
0	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**

Distance transform

Slide adapted from Dan Huttonlocher, Kristen Grauman

Distance Transform (2D)

- 2D case analogous to 1D
  - Initialization
  - Forward and backward pass
    - Fwd pass finds closest above and to left
    - Bwd pass finds closest below and to right

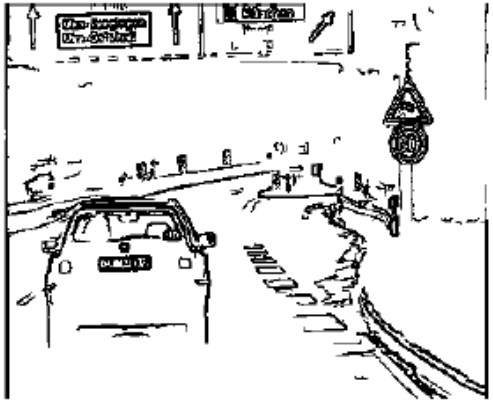


Slide credit: Adapted by Kristen Grauman from Dan Huttonlocher

Chamfer distance

- Average distance to nearest feature

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

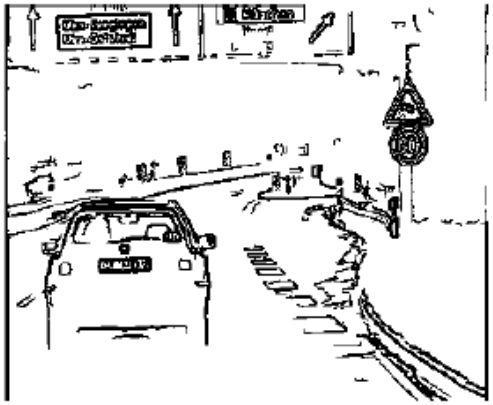


Edge image

Distance transform image

Slide credit: Kristen Grauman

Chamfer distance



Edge image **Distance transform image**

Slide credit: Kristen Grauman

Fig from D. Gavrila, DAGM 1999

Chamfer distance: properties

- Sensitive to scale and rotation
- Tolerant of small shape changes, clutter
- Need large number of template shapes
- Inexpensive way to match shapes

Slide credit: Kristen Grauman

Today

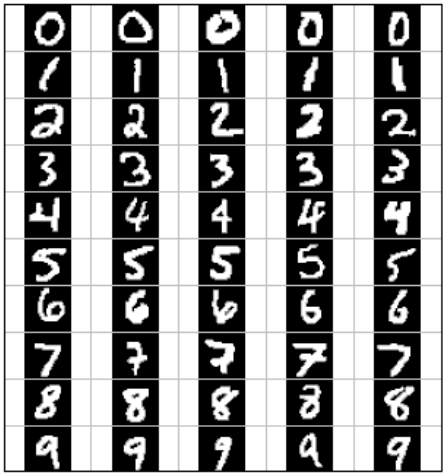
- Edge detection and matching
  - process the image gradient to find curves/contours
  - comparing contours

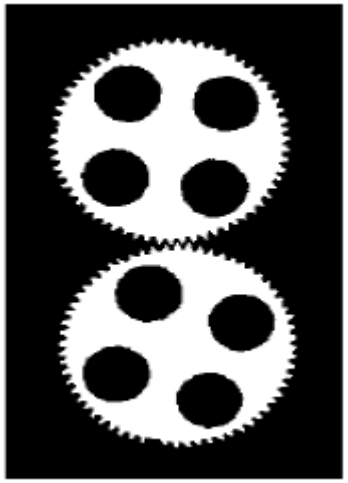
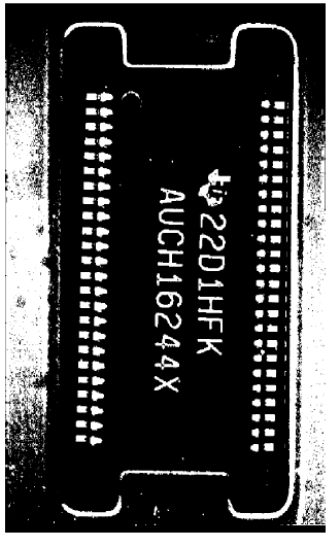
- Binary image analysis
  - blobs and regions

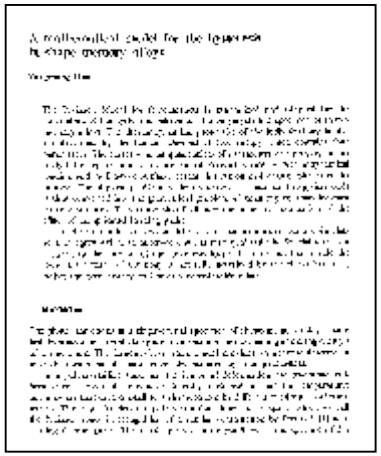
Slide credit: Kristen Grauman

Binary images





Slide credit: Kristen Grauman

Binary image analysis: basic steps

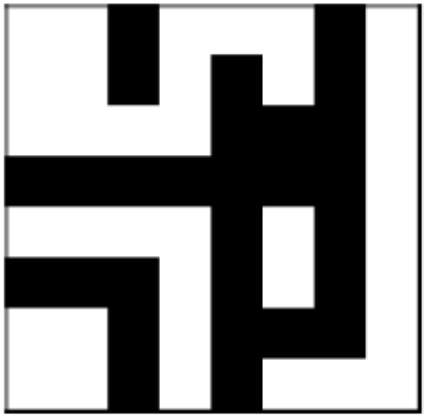
- Convert the image into binary form
  - Thresholding
- Clean up the thresholded image
  - Morphological operators
- Extract separate blobs
  - Connected components
- Describe the blobs with region properties

Slide credit: Kristen Grauman

Binary images

- Two pixel values
  - Foreground and background
  - Mark region(s) of interest

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1



Slide credit: Kristen Grauman

Thresholding

- Grayscale -> binary mask
- Useful if object of interest's intensity distribution is distinct from background

$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \geq T \\ 0 & otherwise. \end{cases}$$

$$F_T[i,j] = \begin{cases} 1 & \text{if } T_1 \leq F[i,j] \leq T_2 \\ 0 & otherwise. \end{cases}$$

$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \in Z \\ 0 & otherwise. \end{cases}$$

- Example
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FITZGIBBON/simplebinary.html

Slide credit: Kristen Grauman

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: edge detection

Gradient magnitude

```
fg_pix = find(gradient_mag > t);
```

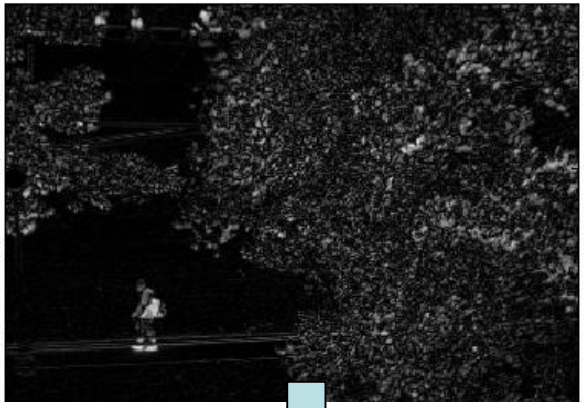
Looking for pixels where gradient is strong.

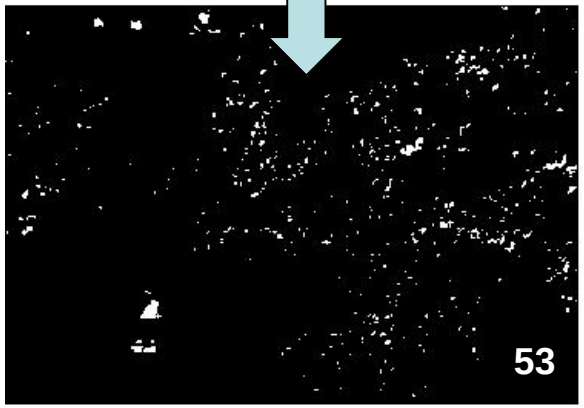
Slide adapted from Kristen Grauman

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: background subtraction





Looking for pixels that differ significantly from the “empty” background.

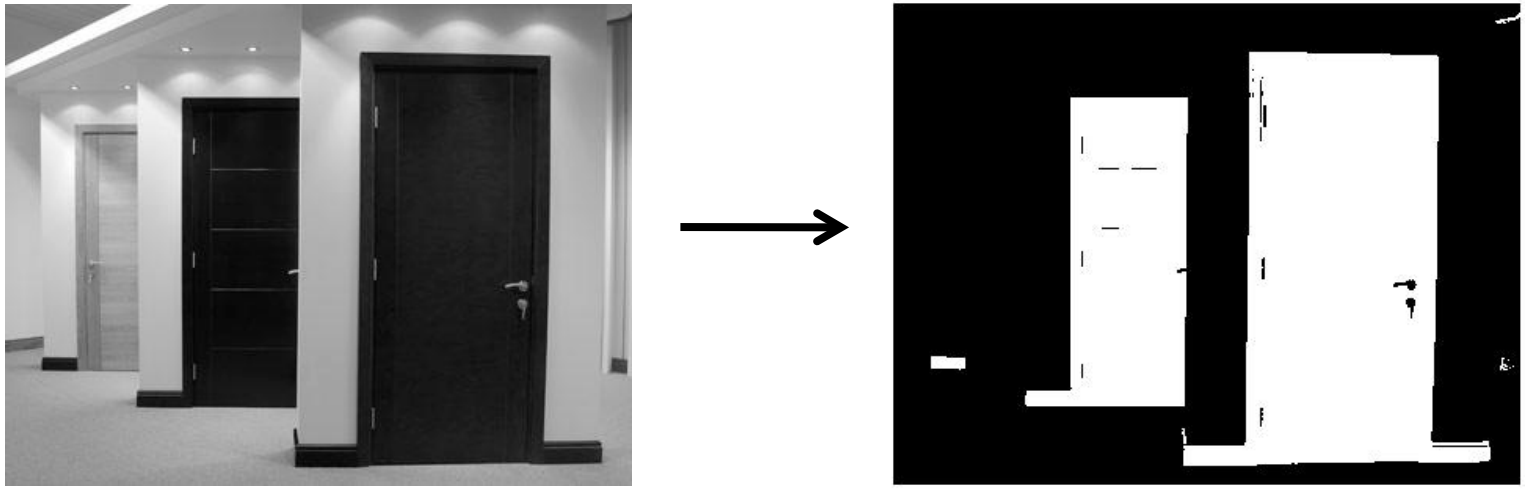
```
fg_pix = find(diff > t);
```

Slide credit: Kristen Grauman

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: intensity-based detection



```
fg_pix = find(im < 65);
```

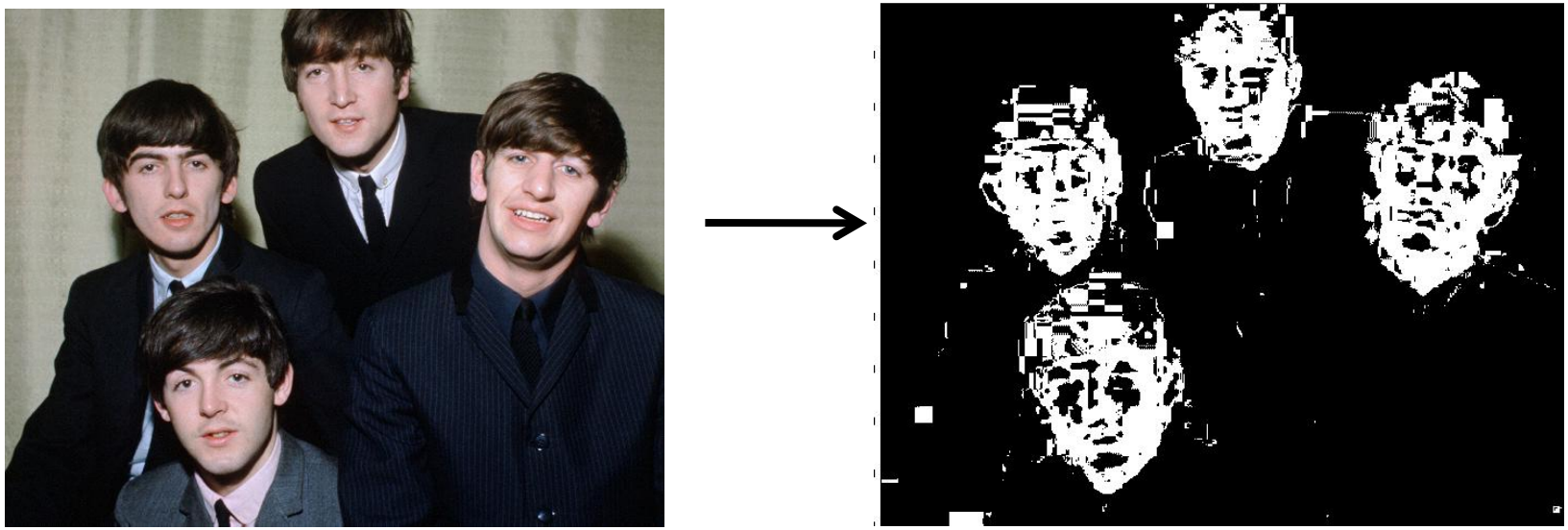
Looking for dark pixels

Slide credit: Kristen Grauman

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: color-based detection

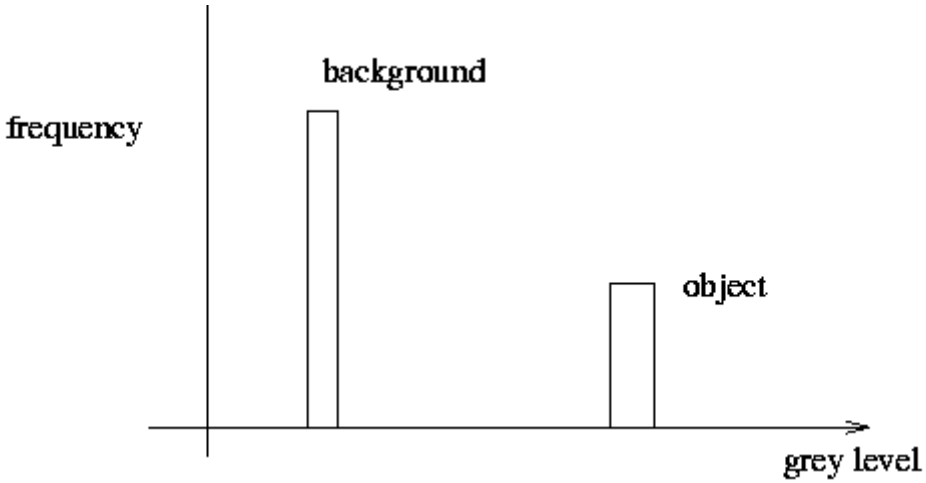


```
fg_pix = find(hue > t1 & hue < t2);
```

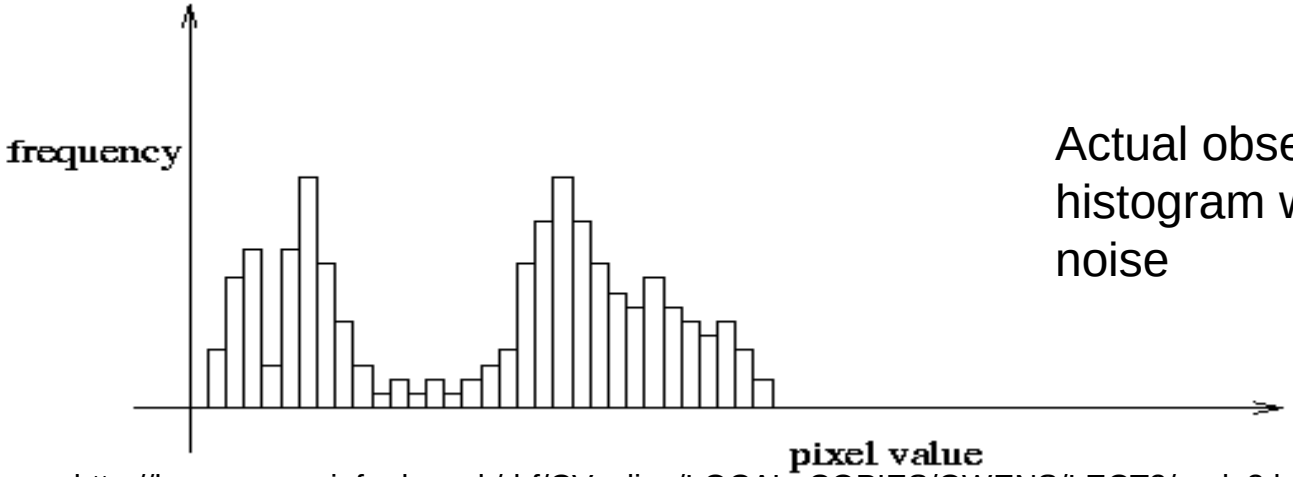
Looking for pixels within a certain hue range.

Slide credit: Kristen Grauman

A nice case: bimodal intensity histograms



Ideal histogram, light object on dark background

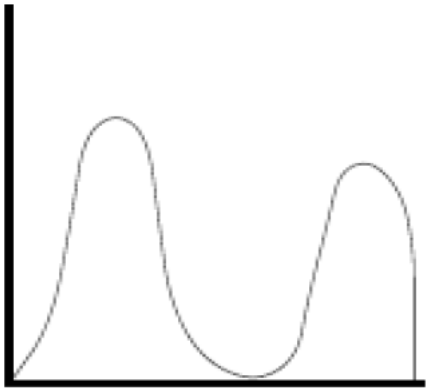


Actual observed histogram with noise

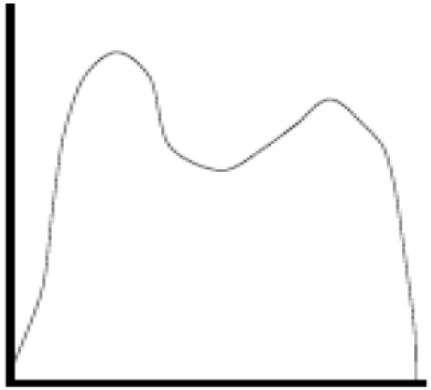
Images: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT2/node3.html

Slide credit: Kristen Grauman

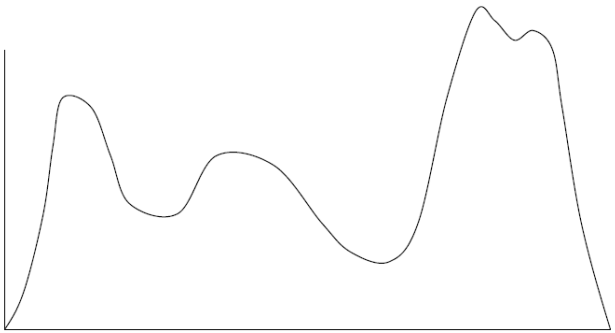
Not so nice cases



Two distinct modes



Overlapped modes

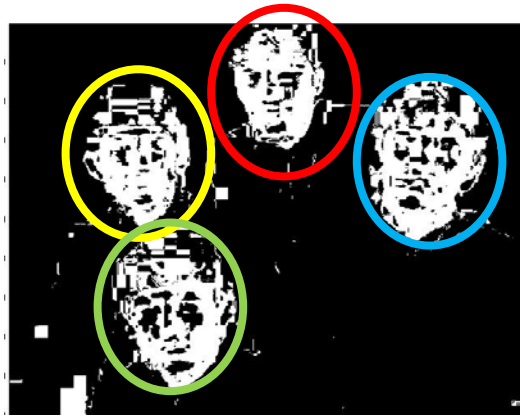


Slide credit: Shapiro and Stockman

Issues

- What to do with “noisy” binary outputs?
  - Holes
  - Extra small fragments

- How to demarcate multiple regions of interest?
  - Count objects
  - Compute further features per object



Slide credit: Kristen Grauman

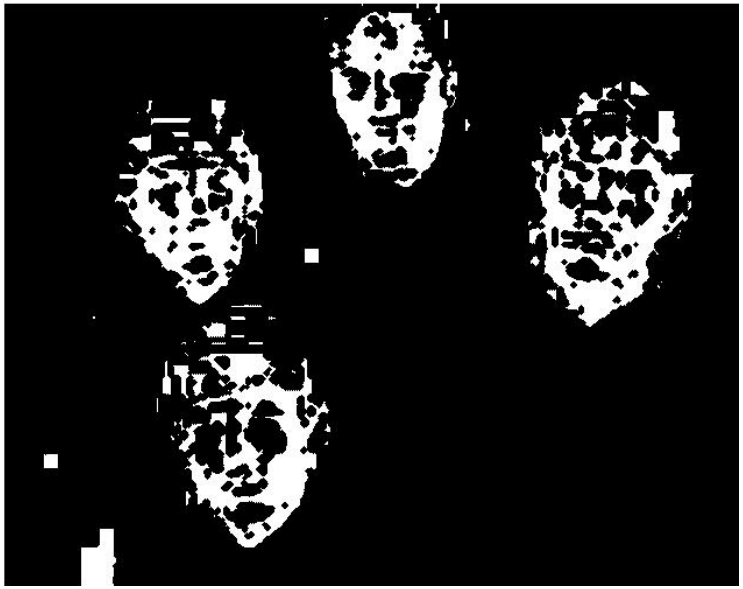
Morphological operators

- Change the shape of the foreground regions via intersection/union operations between a scanning structuring element and binary image
- Useful to clean up result from thresholding
- Basic operators are:
  - Dilation
  - Erosion

Slide credit: Kristen Grauman

Dilation

- Expands connected components
- Grow features
- Fill holes



Before dilation

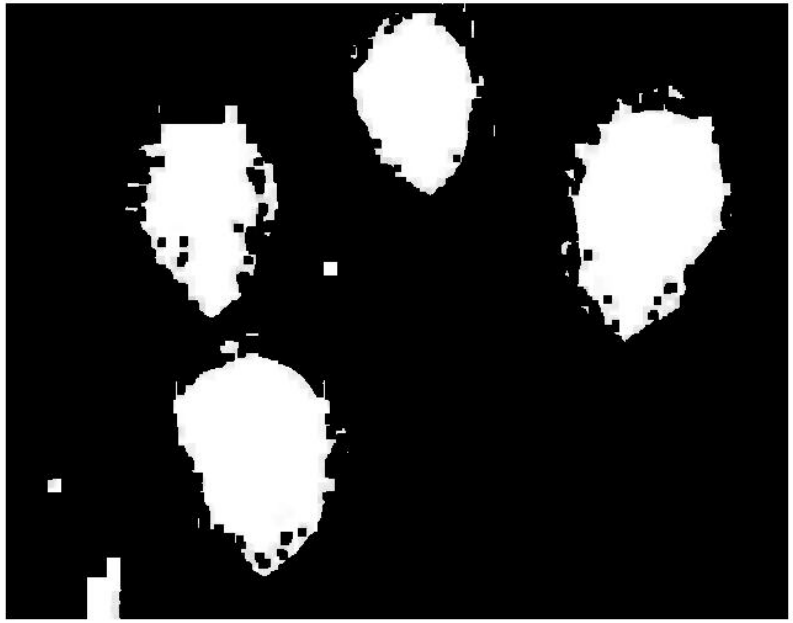
After dilation

Slide credit: Kristen Grauman

Erosion

- Erode connected components
- Shrink features
- Remove bridges, branches, noise

Before erosion

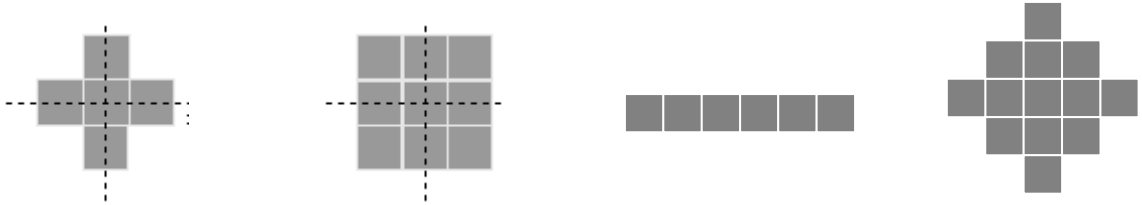


After erosion

Slide credit: Kristen Grauman

Structuring elements

- **Masks** of varying shapes and sizes used to perform morphology, for example:



- Scan mask across foreground pixels to transform the binary image

```
>> help strel
```


Slide credit: Kristen Grauman

Dilation vs. Erosion

At each position:

- **Dilation**: if **current pixel** is 1, then set all the output pixels corresponding to structuring element to 1.

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

1	1								

Slide credit: Adapted by Kristen Grauman from T. Moeslund

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1								

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0							

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0	0						

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0	1	1	1				

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0	1	1	1	1			

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0	1	1	1	1	1		

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0	1	1	1	1	1		

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0	1	1	1	1	1	1	1

Note that the object gets bigger and holes are filled.

```
>> help imdilate
```


Slide credit: Kristen Grauman

2D example for dilation

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

(*a*) Binary image **B**

1	1	1
1	1	1
1	1	1

(*b*) Structuring element **S**

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1			

(*c*) Dilation $\mathbf{B} \oplus \mathbf{S}$

Slide credit: Shapiro & Stockman

Dilation vs. Erosion

At each position:

- **Dilation**: if **current pixel** is 1, then set all the output pixels corresponding to structuring element to 1.

- **Erosion**: if **every pixel** under the structuring element is 1, then set the output pixel corresponding to the current pixel to 1.

Example for Erosion (1D)

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0									

Slide credit: Kristen Grauman

Example for Erosion (1D)

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0								

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0							

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0	0						

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0	0	0					

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0	0	0	1				

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0	0	0	1	0			

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0	0	0	1	0	0		

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0	0	0	1	0	0	0	

Slide credit: Kristen Grauman

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

0	0	0	0	0	1	0	0	0	1

Note that the object gets smaller

```
>> help imerode
```


Slide credit: Kristen Grauman

2D example for erosion

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

(*a*) Binary image **B**

1	1	1
1	1	1
1	1	1

(*b*) Structuring element **S**

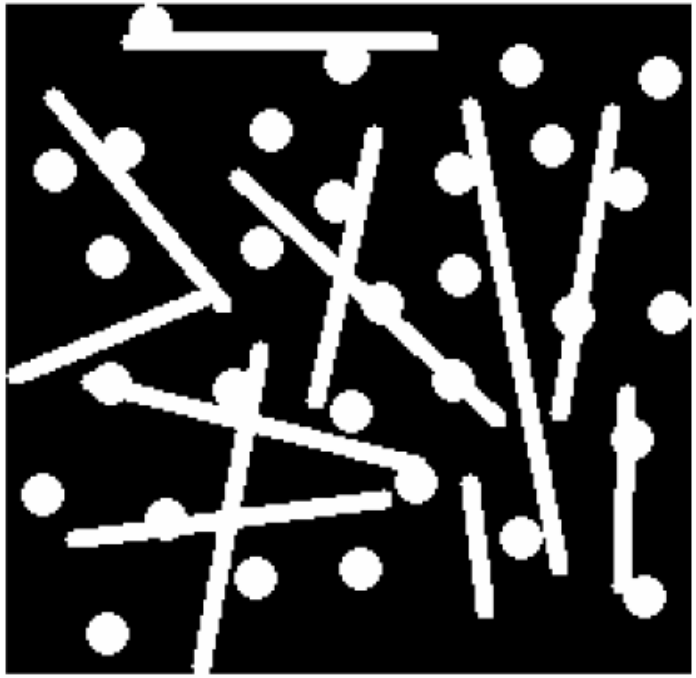
				1	1		
				1	1		
				1	1		

(*d*) Erosion **B** ⊖ **S**

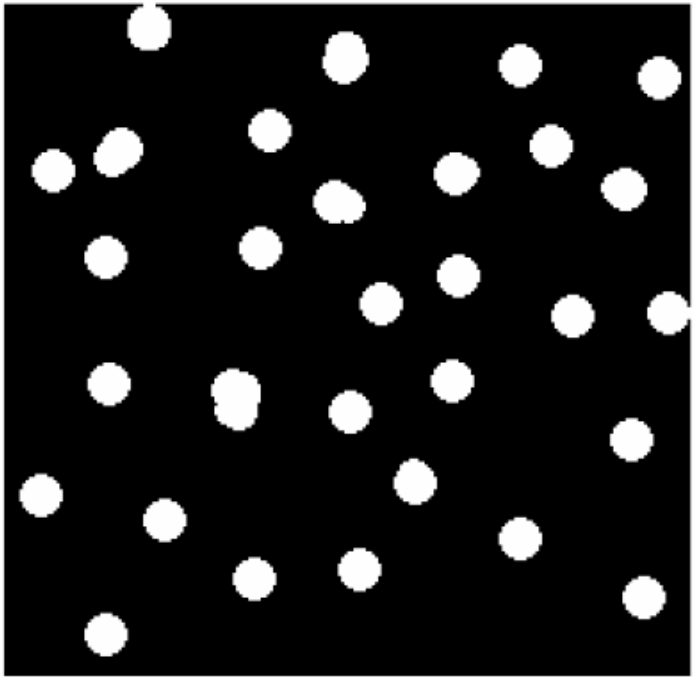
Slide credit: Shapiro & Stockman

Opening

- Erode, then dilate
- Remove small objects, keep original shape



Before opening

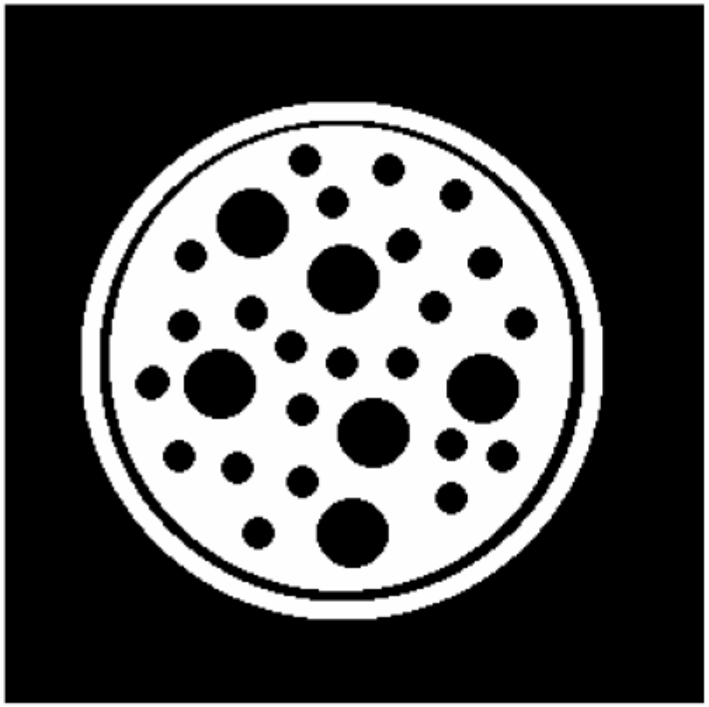


After opening

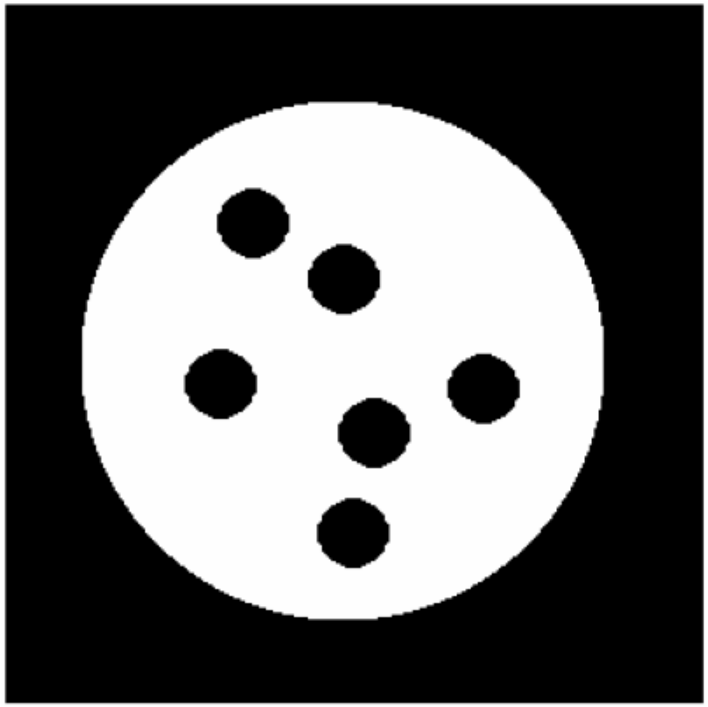
Slide credit: Kristen Grauman

Closing

- Dilate, then erode
- Fill holes, but keep original shape



Before closing



After closing

Applet: http://bigwww.epfl.ch/demo/jmorpho/start.php

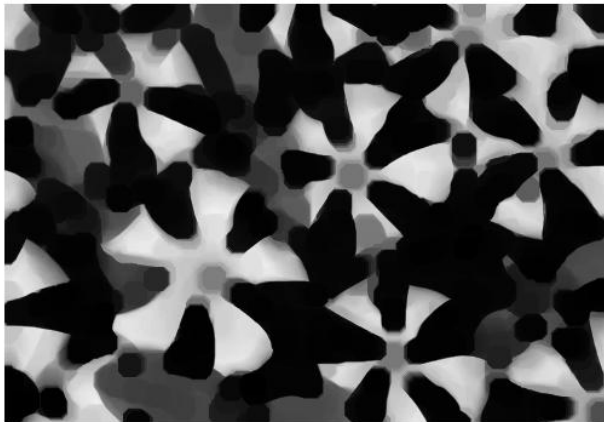
Slide credit: Kristen Grauman

Morphology operators on grayscale images

- Dilation and erosion typically performed on binary images.
- If image is grayscale: for dilation take the neighborhood **max**, for erosion take the **min**.

original

dilated



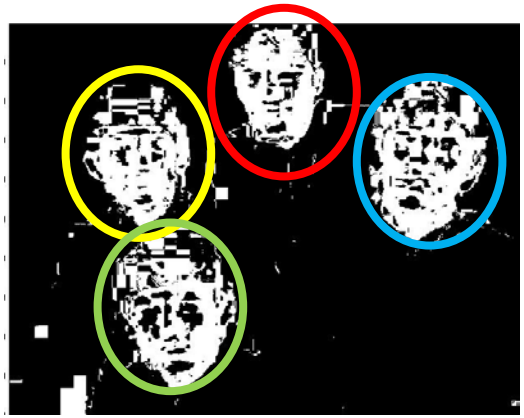
eroded

Slide credit: Kristen Grauman

Issues

- What to do with “noisy” binary outputs?
  - Holes
  - Extra small fragments

- How to demarcate multiple regions of interest?
  - Count objects
  - Compute further features per object



Slide credit: Kristen Grauman

Connected components

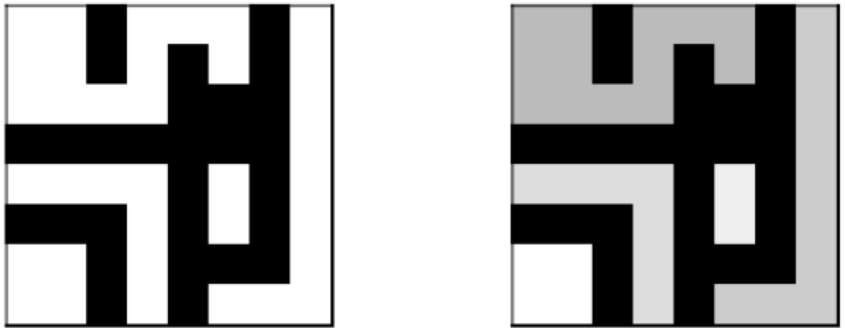
- Identify distinct regions of “connected pixels”

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

a) binary image

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

b) connected components labeling

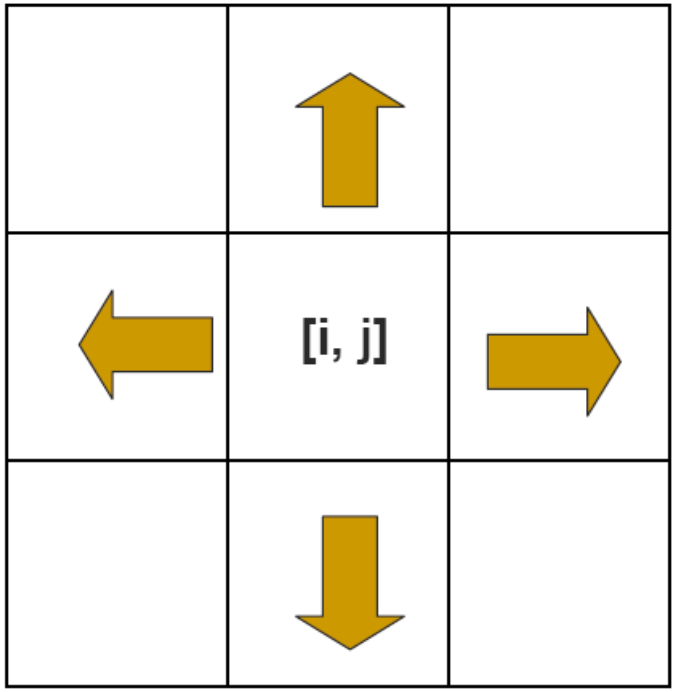


c) binary image and labeling, expanded for viewing

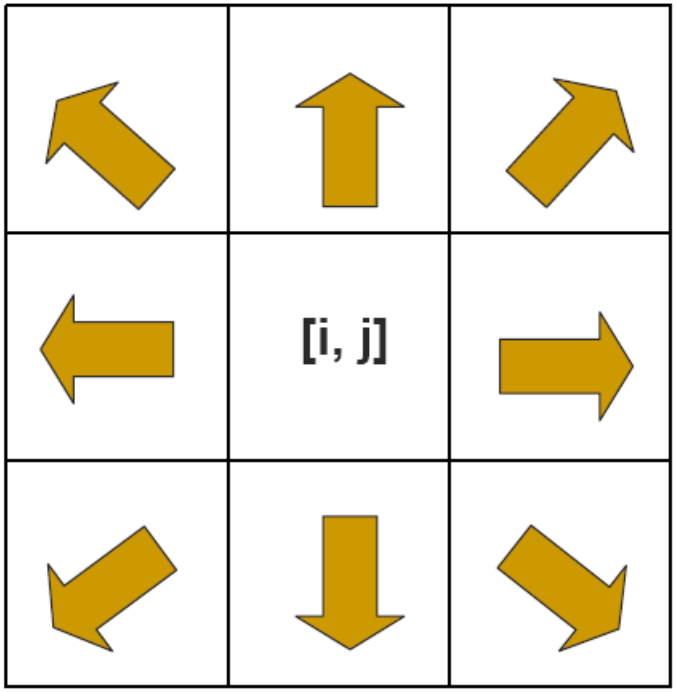
Slide credit: Shapiro and Stockman

Connectedness

- Defining which pixels are considered neighbors



4-connected



8-connected

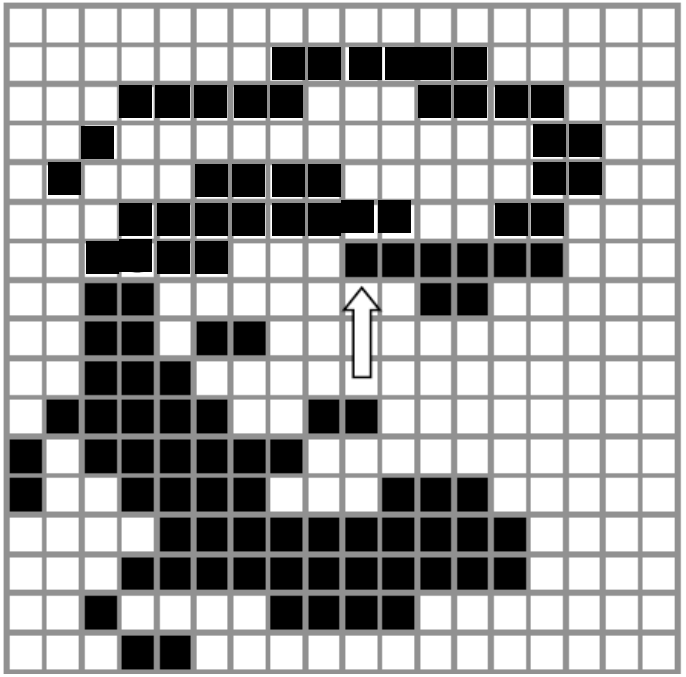
Slide credit: Chaitanya Chandra

Connected components

- We'll consider a sequential algorithm that requires only 2 passes over the image.

- **Input**: binary image
- **Output**: "label" image, where pixels are numbered per their component

- Note: foreground here is denoted with black pixels.

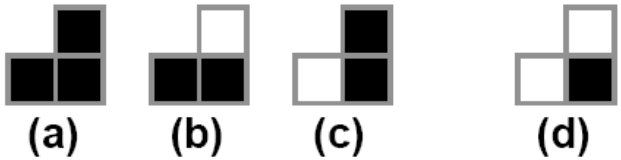


Slide credit: Kristen Grauman

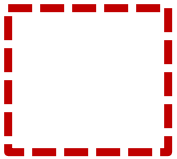
Sequential connected components

- Labeling a pixel only requires to consider its prior and superior neighbors.
- It depends on the type of connectivity used for foreground (4-connectivity here).

Same object | New object



What happens in these cases?

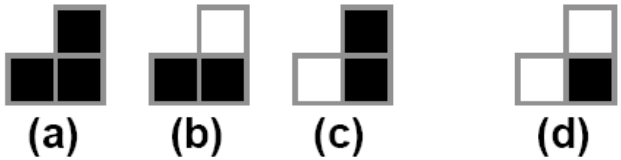


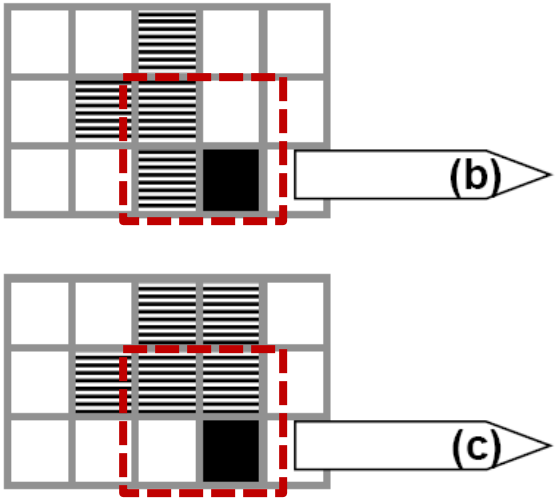
Slide credit: Adapted by Kristen Grauman from J. Neira

Sequential connected components

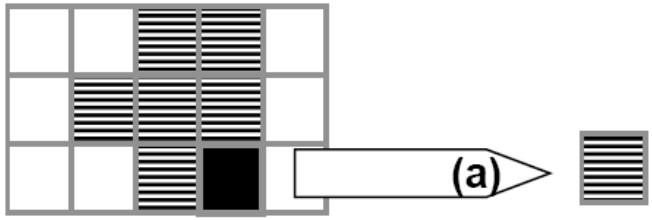
- Labeling a pixel only requires to consider its prior and superior neighbors.
- It depends on the type of connectivity used for foreground (4-connectivity here).

Same object | New object





What happens in these cases?

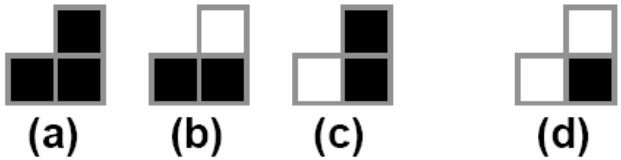


Slide credit: Kristen Grauman

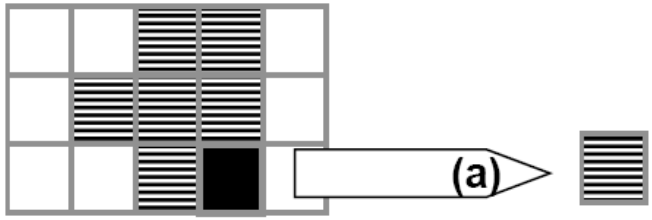
Sequential connected components

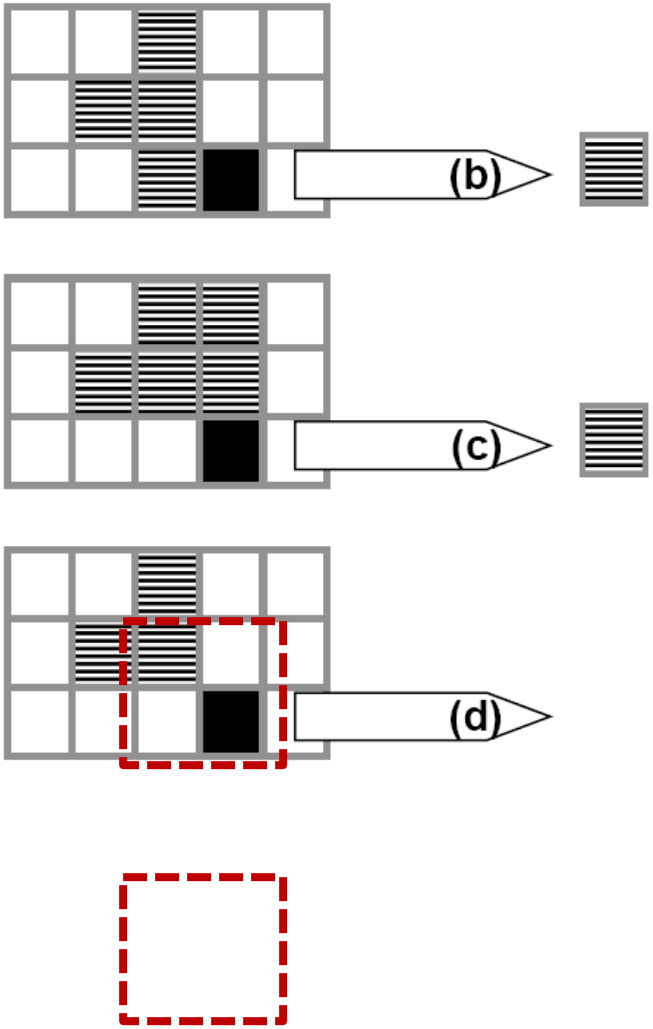
- Labeling a pixel only requires to consider its prior and superior neighbors.
- It depends on the type of connectivity used for foreground (4-connectivity here).

Same object | New object



What happens in these cases?



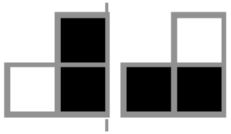


Slide credit: Kristen Grauman

Sequential connected components

- Process the image from left to right, top to bottom.

1. If the next pixel to process is 1-pixel:



1. If only one of its neighbors (<u>superior</u> or <u>left</u>) is 1-pixel, copy its label.

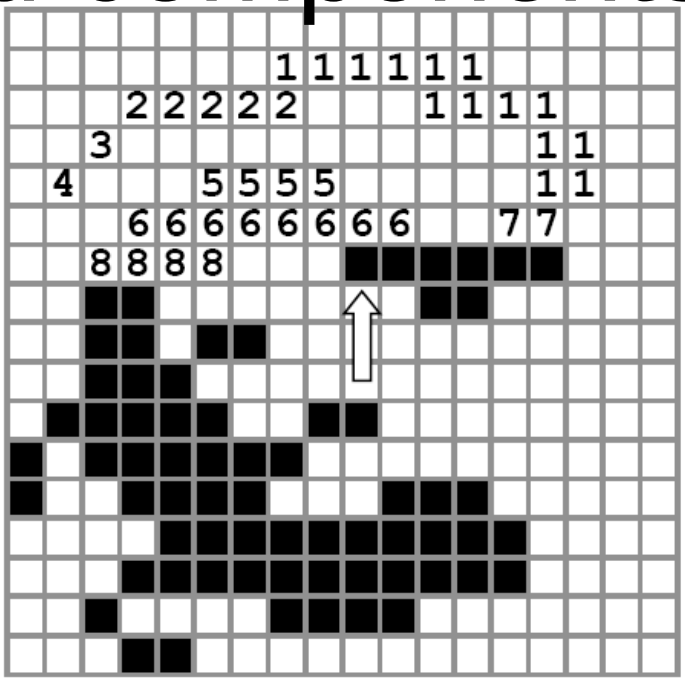
2. If both are, and have the same label, copy it.

3. If they have different labels:

   1. Copy the label from the <u>prior</u>.
   2. Reflect the change in the table of equivalences.

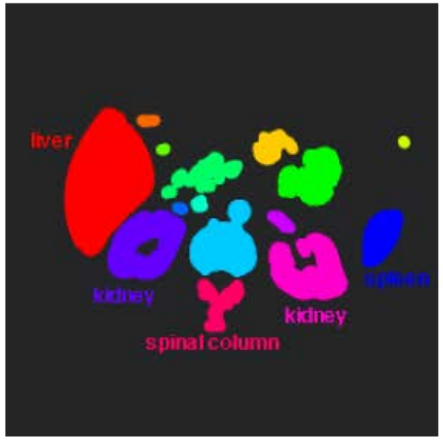
4. Otw, assign a new label.

2. More pixels? Go to step 1.

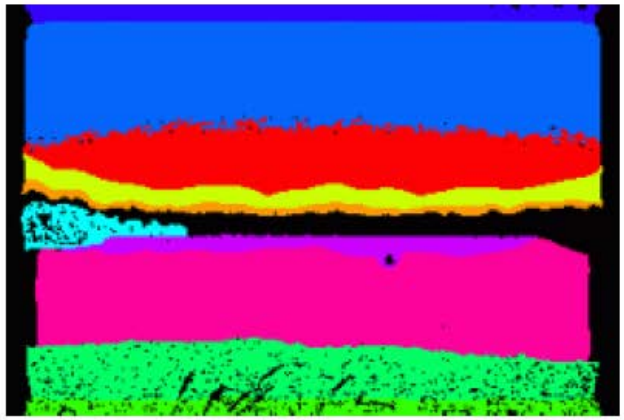


Slide credit: Kristen Grauman

Connected components



connected components of 1's from thresholded image

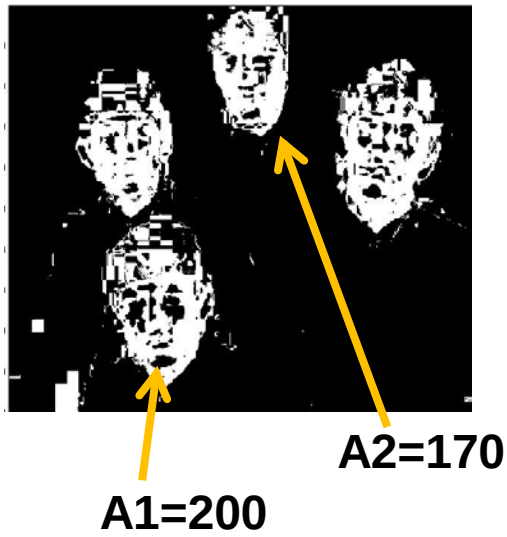


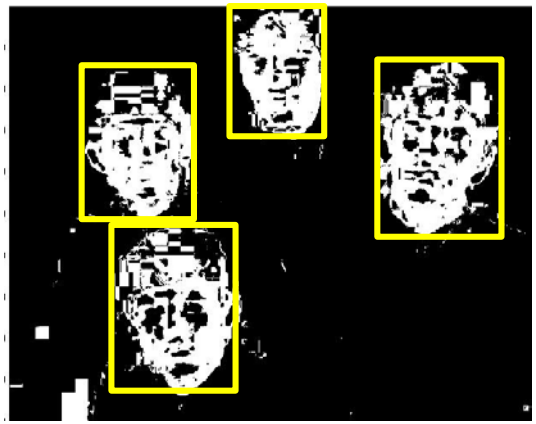
connected components of cluster labels

Slide credit: Pinar Duygulu

Region properties

- Given connected components, can compute simple features per blob, such as:
  - Area (num pixels in the region)
  - Centroid (average x and y position of pixels in the region)
  - Bounding box (min and max coordinates)
  - Circularity (ratio of mean dist. to centroid over std)





Slide credit: Kristen Grauman

Binary image analysis: basic steps (recap)

- Convert the image into binary form
  - Thresholding
- Clean up the thresholded image
  - Morphological operators
- Extract separate blobs
  - Connected components
- Describe the blobs with region properties

Slide credit: Kristen Grauman

Matlab

- **N = hist(Y,M)**
- **L = bwlabel (BW,N);**
- **STATS = regionprops(L,PROPERTIES) ;**
  - **'Area'**
  - **'Centroid'**
  - **'BoundingBox'**
  - **'Orientation', …**
- **IM2 = imerode(IM,SE);**
- **IM2 = imdilate(IM,SE);**
- **IM2 = imclose(IM, SE);**
- **IM2 = imopen(IM, SE);**

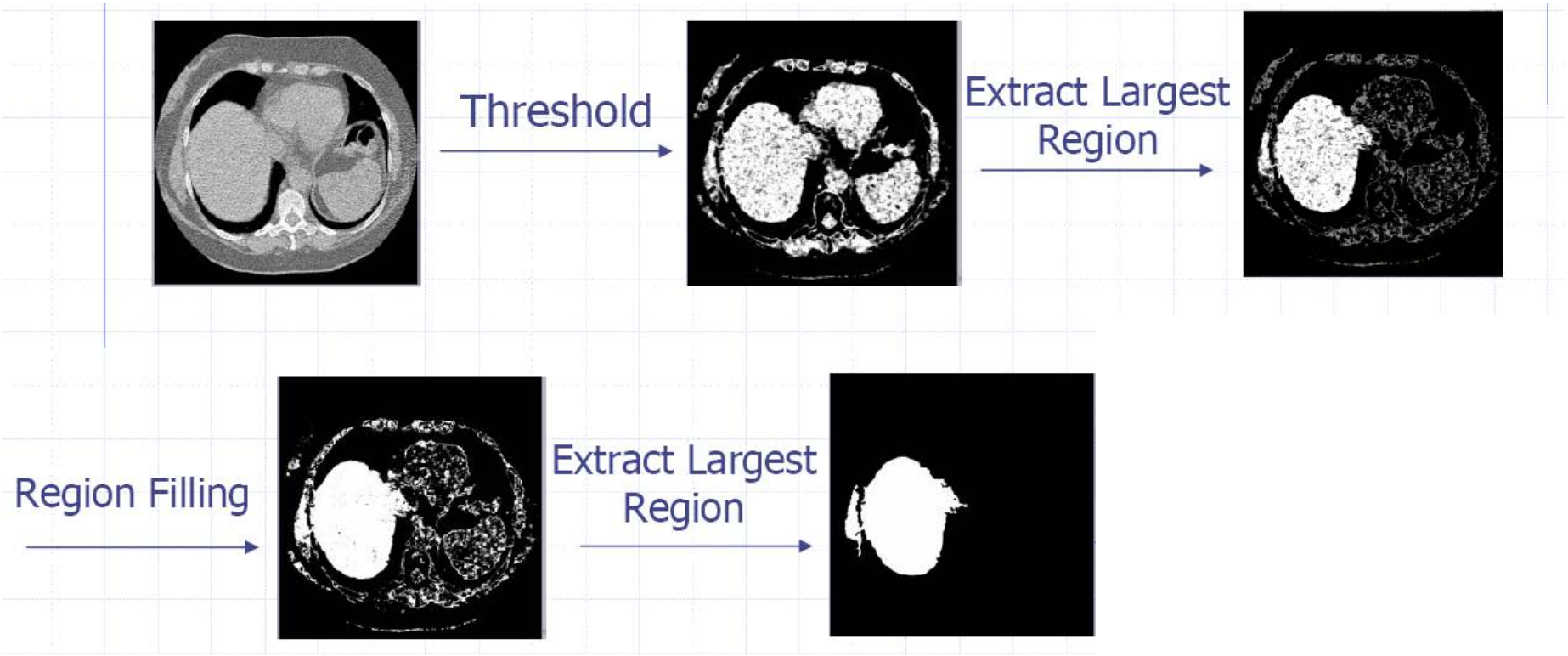
Slide credit: Kristen Grauman

Example using binary image analysis: OCR

[Luis von Ahn et al. http://recaptcha.net/learnmore.html]

Slide credit: Kristen Grauman

Example using binary image analysis: segmentation of a liver



Slide credit: Li Shen

Application by Jie Zhu, Cornell University

Binary images

- Pros
  - Can be fast to compute, easy to store
  - Simple processing techniques available
  - Lead to some useful compact shape descriptors

- Cons
  - Hard to get “clean” silhouettes
  - Noise common in realistic scenarios
  - Can be too coarse of a representation
  - Not 3d

Slide credit: Kristen Grauman

Summary

- Operations, tools

Derivative filters

Smoothing, morphology

Thresholding

Connected components

Matched filters

Histograms

- Features, representations

Edges, gradients

Blobs/regions

Local patterns

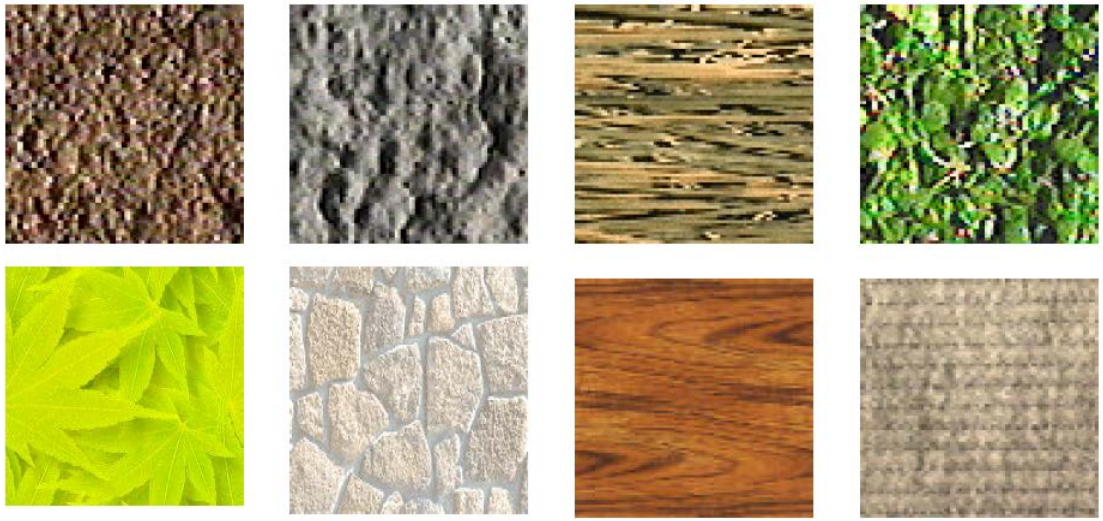
Textures (next)

Color distributions

Slide credit: Kristen Grauman

Coming up

- Texture
  - Read Szeliski 10.5



Questions?

See you Tuesday!