Computer Vision
for HCI

Region Extraction

1

Region Extraction

- Extract set of regions occupied by objects
  - Then perform region shape analysis/recognition
- Group those pixels that are similar in some region "*property*"
  - Intensity, color, texture, motion, etc.
- Topics
  - Background subtraction
  - Morphology
  - Region growing

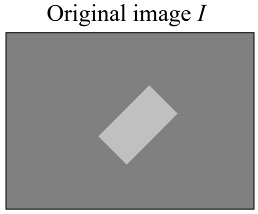
2

2

Simple Extraction

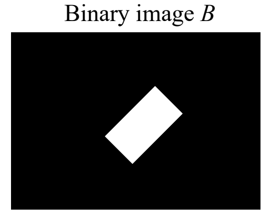
- If object pixels in grayscale image have higher (lower) intensity than known background intensity
  - Thresholding image yields binary image
    - Object: 1's, background: 0's

Original image I



$$B[x,y] = \begin{cases} 1 & \text{if } I[x,y] > T \\ 0 & \text{otherwise} \end{cases}$$

Binary image B

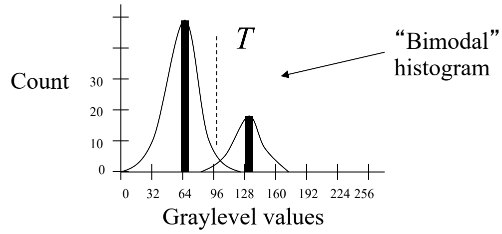


3

3

Thresholds and Histograms

- Distribution of graylevels can be used to learn/determine the binary threshold
- <u>Histogram</u> graphs number of pixels in the image with a particular graylevel, as a function of the possible graylevels
  - Find peaks and set threshold between peaks



4

4

Otsu's Method

- "A threshold selection method from graylevel histograms", IEEE Trans on Sys., Man, and Cyb., Vol 9, No 1, pp 62-66, 1979.
  - Basic idea: threshold is chosen such that the division in the histogram yields the largest reduction in standard deviation of the pixel intensities (black, white)
  - Matlab: graythresh()

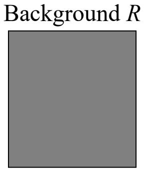
5

5

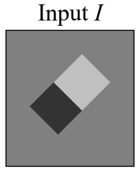
Background Subtraction I

- What if object is not uniformly brighter or darker than background?
- Obtain "background image" (no object present)
- Subtract object image from background image
  - Difference highlights the object
  - Use abs *or* square of difference! *****

Background R

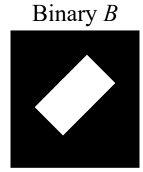


Input I



$$B[x,y] = \begin{cases} 1 & \text{if } \left| I[x,y] - R[x,y] \right| > T \\ 0 & \text{otherwise} \end{cases}$$

Binary B



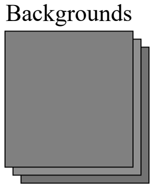
6

6

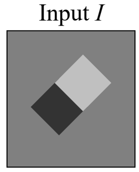
Background Subtraction II

- How choose threshold?
  - Want the difference to be above noise level in images
- Obtain sequence of background images
  - Compute intensity mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ for each pixel (across/through the background images)
- Check statistical distance (# stdev) of new pixel from background pixels (e.g., **Mahalanobis distance**)

Backgrounds

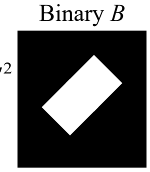


Input I



$$B[x,y] = \begin{cases} 1 & \text{if } \dfrac{\left(I[x,y]-\mu[x,y]\right)^2}{\sigma^2[x,y]} > T^2 \\ 0 & \text{otherwise} \end{cases}$$

Binary B



7

7

Background Subtraction III

- How deal with color?

$$\sqrt{\left(I_R[x,y]-R_R[x,y]\right)^2+\left(I_G[x,y]-R_G[x,y]\right)^2+\left(I_B[x,y]-R_B[x,y]\right)^2} > T$$

Euclidean distance in 3-D (R,G,B space)

$$\frac{\left(I_c[x,y]-\mu_c[x,y]\right)^2}{\sigma_c^2[x,y]} > T_c^2$$

AND/OR result for each color channel (c = R,G,B)
[assumes independence of colors]

8

8

Better Model

- Use the "covariance" of the color information

RGB vector: $C_{x,y} = \left[I_R[x,y], I_G[x,y], I_B[x,y]\right]^T$

Background model:

Mean vector: $M_{x,y} = \dfrac{1}{N}\sum_i C_{x,y}^i$

Covariance matrix: $K_{x,y} = \dfrac{1}{N}\sum_i (C_{x,y}^i - M_{x,y})(C_{x,y}^i - M_{x,y})^T$

$$B[x,y] = \begin{cases} 1 & \text{if } [C_{x,y} - M_{x,y}]^T K_{x,y}^{-1} [C_{x,y} - M_{x,y}] > T^2 \\ 0 & \text{otherwise} \end{cases}$$

Multi-dimensional Mahalanobis Distance

9

9

Shadows

- May not want to select shadows in image as part of the object
  - Shadows due to change (drop) in intensity
- Normalize brightness of RGB space

$$R^n = \frac{R}{R+G+B} \qquad G^n = \frac{G}{R+G+B} \qquad B^n = \frac{B}{R+G+B}$$

- Or convert RGB to YUV, HSI, or YIQ
  - Intensity and chrominance are <u>decoupled</u>
  - Perform previous methods using chrominance channels

10

10

RGB to YIQ

- YIQ colorspace
  - Decouples intensity and color
    - Y component provides all monochrome/intensity information
    - I and Q components carry the color information
    - More bandwidth to Y than I,Q (based on human perceptual sensitivity ranges)

Recall the NTSC conversion formula!

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

11

11

Blue/Green Screening

(Chroma-Keying)

- A subject is photographed in front of an evenly lit, bright, pure blue/green background
- The compositing process, whether photographic or electronic, replaces all the blue/green in the picture with another image, known as the background plate

More info at
https://www.seanet.com/~bradford/blue_green_screen_visual_effects_1.html

12

12

Why Green?

- Video cameras are most sensitive to green channel
  - Green channel is cleanest channel in most digital cameras today
  - Recall Bayer Pattern filtering
    - Pixel array records twice as many green pixels as red or blue (recording resolution is double)

13

13

Green-Screens

14

14

Multi-Modal Background Modeling

- Static single distribution background models often too simplistic for long duration videos
- Need to account for dynamic factors
  - Illumination changes
  - Moving scene elements (e.g., swaying trees)
  - Objects that are stationary for significant durations
- Need to adapt over time
- Use temporally-adaptive **Gaussian Mixture Model** (GMM) for each pixel (Stauffer and Grimson 1998)

15

First-Down Line

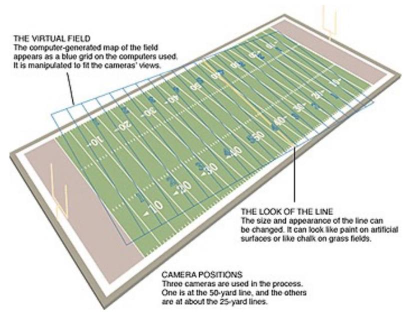
http://www.howstuffworks.com/first-down-line.htm

16

16

Model-Based Backsub

The system determines which pixels to change based on very precise information about the camera's view, a 3D model of the field, which camera is on-air, and a palette of colors for the field and another palette for players.



17

17

Binary Image Morphology

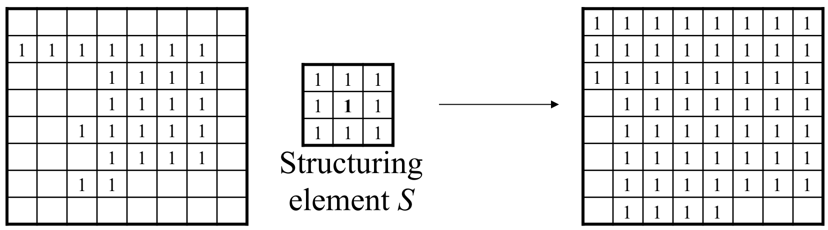
- Useful once extract binary regions
  - e.g, from background subtraction
- Basic operations of binary morphology
  - Dilation
  - Erosion
  - Closing
  - Opening
- Binary image *B*
- Structuring element *S*
  - Smaller binary image or mask

18

18

Dilation

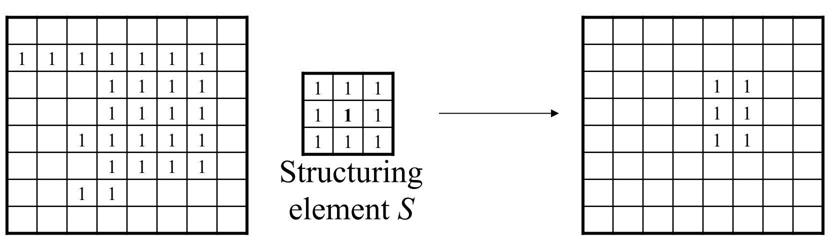
- Dilation operation <u>enlarges</u> a binary region
- Convolve *S* throughout *B*
  - If <u>center</u> of *S* touches binary 1-pixel in *B*, then *S* is OR-ed with region in *B* and placed into output image



Binary image *B* — Dilation of *B*

Erosion

- Erosion operation <u>decreases</u> size of binary region
- Convolve *S* throughout *B*
  - If <u>every</u> 1-pixel of S touches a binary 1-pixel in *B*, then <u>center pixel</u> of *S* is OR-ed with corresponding pixel in *B* and placed into output image



Binary image *B* — Erosion of *B*

Closing

- Closing operation closes up internal holes and eliminates “bays” along boundary
- Perform dilation followed by 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				

Binary image *B*

1	1	1
1	**1**	1
1	1	1

Structuring element *S*

$\longrightarrow$

	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				

Closing of *B*

Opening

- Opening operation gets rid of small portions of the binary region that jut out from boundary into background region
- Perform erosion followed by 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				

Binary image *B*

1	1	1
1	**1**	1
1	1	1

Structuring element *S*

$\longrightarrow$

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

Opening of *B*

Counting Objects

- So far, no notion of a distinct/separate object region
  - No selection of a region of pixels
- Segment binary image (after morphology)
  - Connected components
  - Region growing

23

23

Recursive Connected Components

- Steps
  - Scan binary image left-to-right, top-to-bottom
  - If unlabeled 1-pixel encountered, assign new label number (new seed pixel)
  - Recursively check neighboring 1-pixels and assign same label
    - 4 or 8 connected
  - Stop when all 1-pixels have been labeled
- Advantages
  - Easy to implement, similar to flood-fill algorithm
- Problems
  - Can run into **stack overflow for large images**

24

24

Sequential Connected Components

- Steps (2-pass algorithm)
  - Scan binary image left-to-right, top-to-bottom
  - If unlabeled 1-pixel encountered, assign new label number according to rules:

$$\begin{array}{cc} & 0 \\ 0 & 1 \end{array} \rightarrow \begin{array}{cc} & 0 \\ 0 & L \end{array} \qquad \begin{array}{cc} & 0 \\ L & 1 \end{array} \rightarrow \begin{array}{cc} & 0 \\ L & L \end{array}$$

$$\begin{array}{cc} & L \\ 0 & 1 \end{array} \rightarrow \begin{array}{cc} & L \\ 0 & L \end{array} \qquad \begin{array}{cc} & L \\ M & 1 \end{array} \rightarrow \begin{array}{cc} & L \\ M & L \end{array} \text{ (Set L = M)}$$

  - Determine equivalent classes of labels
  - In second pass, assign same label to all elements in equivalent class

25

25

Example

Binary image

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

After Pass 1

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

Equivalence classes:
(1,2)
(3,7)

After Pass 2

1	1		**1**	**1**	**1**		3
1	1		**1**		**1**		3
1	1	1	1				3
							3
4	4	4	4		5		3
			4		5		3
6	6		4				3
6	6		4		**3**	**3**	3

26

26

Removing Small Regions

- Some noise regions may appear, or may only want largest object in scene, etc.
- Count number of pixels belonging to regions during recursion
- If region-size < T, then recursively "remove" binary pixels
  - Recursively check neighboring 1-pixels for same label
    - Delete (make 0-pixel) in binary image
    - Delete label
  - Stop when all 1-pixels in region have been removed

<u>Or</u> use Matlab with

bwareaopen(bwIm, 20, 8)

"Removes all regions fewer than 20 pixels (8-connected)"

27

27

Summary

- Region extraction
- Background subtraction
  - Simple differences
  - Statistical distance
  - Multi-Modal
- Morphology
  - Dilation to enlarge region
  - Erosion to reduce region
  - Closing (dilation + erosion)
  - Opening (erosion + dilation)
- Region growing
  - Connected components
    - Recursive
    - Sequential
- Matlab
  - bwmorph(): dilate, erode, open, close, etc.
  - bwlabel(): label connected components in binary image
  - bwareaopen(): remove small regions

28

28