Analysis of Binary Images

Introduction to Computer Vision

CSE 152

Lecture 7

Announcements

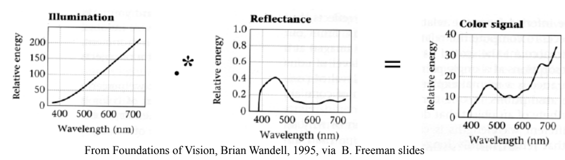
- HW0 returned
- HW1 due Thursday
- Wed 10:00, Discussion Section EBU3b 2217

The appearance of colors

- Color appearance is strongly affected by (at least):
  - Spectrum of lighting striking the retina
  - other nearby colors (space)
  - adaptation to previous views (time)
  - "state of mind"

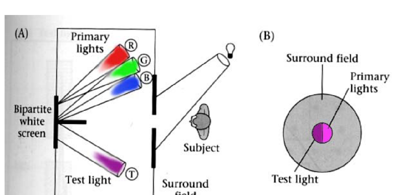
Color Reflectance

Measured color spectrum is a function of the spectrum of the illumination and reflectance



From Foundations of Vision, Brian Wandell, 1995, via B. Freeman slides

Color matching experiment



4.10 THE COLOR-MATCHING EXPERIMENT. The observer views a bipartite field and adjusts the intensities of the three primary lights to match the appearance of the test light. (A) A top view of the experimental apparatus. (B) The appearance of the stimuli to the observer. After Judd and Wyszecki, 1975.

Foundations of Vision, by Brian Wandell, Sinauer Assoc., 1995

slide from T. Darrel

Color matching functions

- Choose primaries, say $P_1(\lambda)$, $P_2(\lambda)$, $P_3(\lambda)$
- For monochromatic (single wavelength) energy function, what amounts of primaries will match it?
- i.e., For each wavelength λ, determine how much of A, of B, and of C is needed to match light of that wavelength alone.

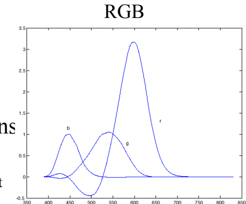
$$a(\lambda)$$

$$b(\lambda)$$

$$c(\lambda)$$

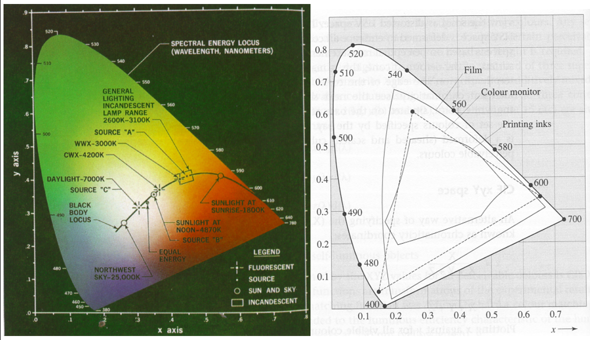
- These are color matching functions

RGB



primaries are monochromatic at 645.2nm, 526.3nm, 444.4nm

CIE xyY (Chromaticity Space)

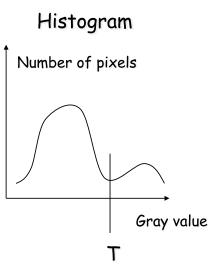


Basic Steps

1. Labeling pixels as foreground/background (0,1).
2. Morphological operators (sometimes)
3. Find pixels corresponding to a region
4. Compute properties of each region

Histogram-based Segmentation

Ex: bright object on dark background:



- Select threshold
- Create binary image:

  $I(x,y) < T \rightarrow O(x,y) = 0$

  $I(x,y) \geq T \rightarrow O(x,y) = 1$

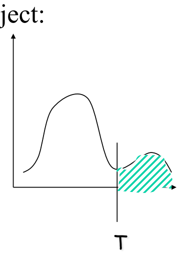
[From Octavia Camps]

How do we select a Threshold?

- Manually determine threshold experimentally.
  - Good when lighting is stable and high contrast.
- Automatic thresholding
  - P-tile method
  - Mode method
  - Peakiness detection
  - Iterative algorithm

P-Tile Method

- If the *size* of the object is approx. known, pick T s.t. the area under the histogram corresponds to the size of the object:



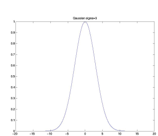
[From Octavia Camps]

Mode Method

- Model intensity in each region R_i as "constant" + $N(0,\sigma_i)$:

If $(x,y) \in R_i$ then, $I(x,y) = \mu_i + n_i(x,y)$

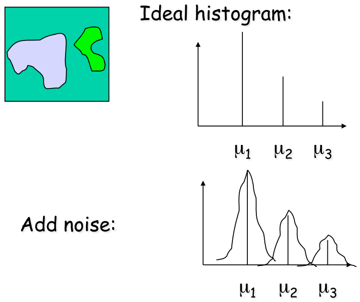
$$p(n_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\frac{n_i^2}{\sigma_i^2}}$$



$$E(n_i) = 0 \qquad E(n_i^2) = \sigma_i^2$$

[From Octavia Camps]

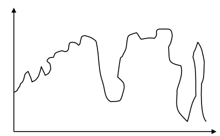
Example: Image with 3 regions



The valleys are good places for thresholding to separate regions.

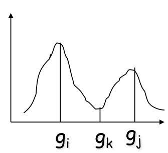
Finding the peaks and valleys

- It is a not trivial problem:



"Peakiness" Detection Algorithm

- Find the two **HIGHEST LOCAL MAXIMA** at a **MINIMUM DISTANCE APART**: g_i and g_j
- Find **lowest point** between them: g_k
- Measure "peakiness":
  - $\min(H(g_i),H(g_j))/H(g_k)$
- Find (g_i,g_j,g_k) with highest peakiness



Blob Tracking for Robot Control

Regions

What is a region?

- "Maximal connected set of points in the image with same brightness value" (e.g., 1)
- Two points are *connected* if there exists a continuous path joining them.
- Regions can be *simply connected* (For every pair of points in the region, all smooth paths can be smoothly and continuously deformed into each other). Otherwise, region is *multiply connected* (holes)

Connected Regions

		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										

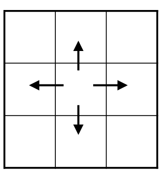
- What are the connected regions in this binary image?
- Which regions are contained within which region?

Connected Regions

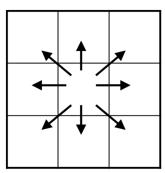
		1							1	1	1	1	1		
	1	1						1					1		
	1	1					1	1					1	1	
	1	1							1	1	1	1	1		

- What the connected regions in this binary image?
- Which regions are contained within which region?

Four & Eight Connectedness



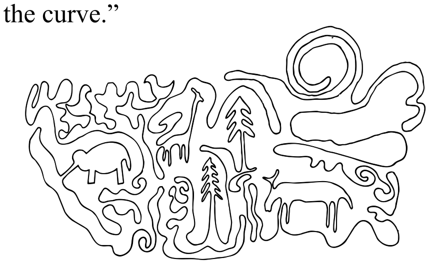
Four Connected



Eight Connected

Jordan Curve Theorem

- "Every closed curve in R^2 divides the plane into two region, the 'outside' and 'inside' of the curve."



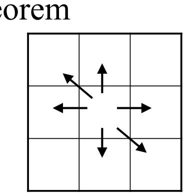
Problem of 4/8 Connectedness

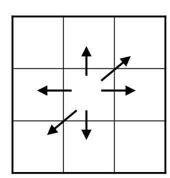
		1	1	1		
	1				1	
	1				1	
		1	1	1		

- 8 Connected:
  - 1's form a closed curve, but background only forms one region.
- 4 Connected
  - Background has two regions, but ones form four "open" curves (no closed curve)

To achieve consistency w.r.t. Jordan Curve Theorem

1. Treat background as 4-connected and foreground as 8-connected.
2. Use 6-connectedness





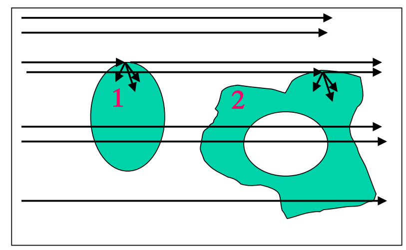
Recursive Labeling Connected Component Exploration

```
Procedure Label (Pixel)
BEGIN
  Mark(Pixel) <- Marker;
  FOR neighbor in Neighbors(Pixel) DO
     IF Image (neighbor) = 1 AND Mark(neighbor)=NIL THEN
           Label(neighbor)
END

BEGIN Main
  Marker <- 0;
  FOR Pixel in Image DO
     IF Image(Pixel) = 1 AND Mark(Pixel)=NIL THEN
           BEGIN
              Marker <- Marker + 1;
              Label(Pixel);
           END;
END
```

Globals:
Marker: integer
Mark: Matrix same size as Image, initialized to NIL

Recursive Labeling Connected Component Exploration



Some notes

- Once labeled, you know how many regions (the value of Marker)
- From Mark matrix, you can identify all pixels that are part of each region (and compute area)
- How deep does stack go?
- Iterative algorithms (See reading from Horn)
- Parallel algorithms

Recursive Labeling Connected Component Exploration

```
Procedure Label (Pixel)
BEGIN
  Mark(Pixel) <- Marker;
  FOR neighbor in Neighbors(Pixel) DO
     IF Image (neighbor) = 1 AND Mark(neighbor)=nil THEN
           Label(neighbor)
END

BEGIN Main
  Marker <- 0;
  FOR Pixel in Image DO
     IF Image(Pixel) = 1 AND Mark(Pixel)=nil THEN
           BEGIN
              Marker <- Marker + 1;
              Label(Pixel);
           END;
END
```

Globals:
Marker: integer
Mark: Matrix same size as Image, initialized to NIL

Some notes

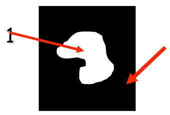
- How deep does stack go?
- Iterative algorithms (See reading from Horn)
- Parallel algorithms

Properties extracted from binary image

- A tree showing containment of regions
- Properties of a region
  1. Genus – number of holes
  2. Centroid
  3. Area
  4. Perimeter
  5. Moments (e.g., measure of elongation)
  6. Number of “extrema” (indentations, bulges)
  7. Skeleton

Moments

B(x,y)



The region S is defined as:

$$S = \{(x,y) \mid B(x,y) = 1\}$$

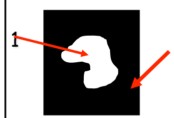
Given a pair of non-negative integers (j,k) the *discrete* (j,k)[th] *moment* of S is defined as:

$$M_{jk}(S) = \sum_{(x,y)\in S} x^j y^k$$

$$M_{j,k} = \sum_{x=1}^{n}\sum_{y=1}^{m} B(x,y)x^j y^k$$

- Fast way to implement computation over n by m image or window
- One object

Area: Moment M_{00}



$$S = \{(x,y) \mid f(x,y) = 1\}$$

$$M_{jk}(S) = \sum_{(x,y)\in S} x^j y^k$$

Example:

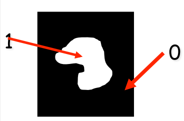
$$M_{oo}(S) = \sum_{(x,y)\in S} x^0 y^0 = \sum_{(x,y)\in S} 1 = \#(S)$$

Area of S !!

$$M_{0,0} = \sum_{x=1}^{n}\sum_{y=1}^{m} B(x,y)$$

- Fast way to implement computation over n by m image or window
- One object

Moments: Centroid



$$S = \{(x,y) \mid f(x,y) = 1\}$$

$$M_{jk}(S) = \sum_{(x,y)\in S} x^j y^k$$

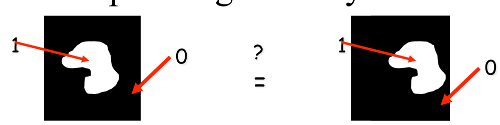
Example:

$$M_{10}(S) = \sum_{(x,y)\in S} x^1 y^0 = \sum_{(x,y)\in S} x \qquad M_{01}(S) = \sum_{(x,y)\in S} x^0 y^1 = \sum_{(x,y)\in S} y$$

$$\frac{M_{10}(S)}{M_{oo}(S)} = \frac{\sum_{(x,y)\in S} x}{\#(S)} = \bar{x} \qquad \frac{M_{01}(S)}{M_{oo}(S)} = \frac{\sum_{(x,y)\in S} y}{\#(S)} = \bar{y}$$

Center of gravity (Centroid) of S !!

Shape recognition by Moments

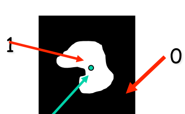


Recognition could be done by comparing moments

However, moments M_{jk} are not invariant under:

- Translation
- Scaling
- Rotation
- Skewing

Central Moments



$(\bar{x}, \bar{y})$

$$S = \{(x,y) \mid f(x,y) = 1\}$$

$$\bar{x} = \frac{M_{10}(S)}{M_{oo}(S)} \qquad \bar{y} = \frac{M_{01}(S)}{M_{oo}(S)}$$

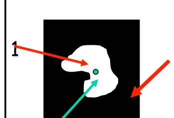
Given a pair of non-negative integers (j,k) the <u>*central*</u> (j,k)[th] <u>*moment*</u> of S is given by:

$$\mu_{jk}(S) = \sum_{(x,y)\in S} (x-\bar{x})^j (y-\bar{y})^k$$

Central Moments

$$S = \{(x,y) \mid f(x,y) = 1\}$$

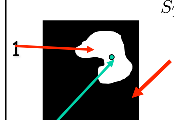
$$\mu_{jk}(S) = \sum_{(x,y)\in S} (x-\bar{x})^j (y-\bar{y})^k$$



$(\bar{x}, \bar{y})$

Translation by T = (a,b) :

$$S_T = \{(x^*, y^*) \mid x^* = x + a, y^* = y + b, (x,y) \in S\}$$



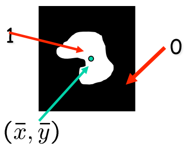
$(\bar{x^*}, \bar{y^*})$

$$\bar{x^*} = \frac{M_{10}(S_T)}{M_{oo}(S_T)} = \bar{x} + a \qquad \bar{y^*} = \frac{M_{01}(S_T)}{M_{oo}(S_T)} = \bar{y} + b$$

$$\mu_{jk}(S_T) = \mu_{jk}(S)$$

Translation INVARIANT!

Normalized Moments



$$S = \{(x,y) | f(x,y) = 1\}$$

$$\mu_{jk}(S) = \sum_{(x,y)\in S} (x-\bar{x})^j (y-\bar{y})^k$$

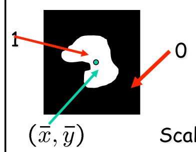
$$\sigma_x = \sqrt{\frac{\mu_{20}(S)}{M_{oo}(S)}} \qquad \sigma_y = \sqrt{\frac{\mu_{02}(S)}{M_{oo}(S)}}$$

Given a pair of non-negative integers (j,k) the *__normalized__* (j,k)th *__moment__* of S is given by:

$$m_{jk}(S) = \sum_{(x,y)\in S} \left(\frac{x-\bar{x}}{\sigma_x}\right)^j \left(\frac{y-\bar{y}}{\sigma_y}\right)^k$$

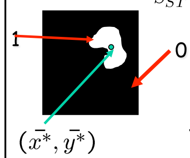
Normalized Moments

$$S = \{(x,y) | f(x,y) = 1\}$$



Scaling by (a,c) and translating by T = (b,d) :

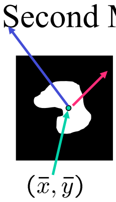
$$S_{ST} = \{(x^*, y^*) | x^* = ax+b, y^* = cy+d, (x,y) \in S\}$$



$$m_{jk}(S_{ST}) = m_{jk}(S)$$

Scaling and translation INVARIANT!

Region Orientation from Second Moment Matrix



- Second Centralized Moment Matrix

$$\begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix}$$

- Eigenvectors of Moment Matrix give orientation