Binary image processing
Big picture

Low-level
Edge/feature detection
(Signal-processing)

Mid-level
Grouping
(Graph algorithms)

High-level
Recognition
(Machine learning)

Input image
Edge image
$2^{1/2}$-D sketch
3-D model

1982

Tiger
Sand
Furry
Flat
Tiger
Sand
How do we generate binary images?
Edge/feature detection
Pixel-level models (1): statistical classifiers

Jones & Rehg “Statistical Color Models with Application to Skin Detection” IJCV 02

Color histogram on RGB. Why not Lab?
Pixel-level models (2): thresholding

http://en.wikipedia.org/wiki/Mooney_Face_Test
Pixel-level models (2): background subtraction

Show pixels that don’t match a background image

Mark pixel ‘i’ with the most-recent frame it was declared foreground

Motion History Images (Davis and Boback)
Outline

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

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

• Matching binary edge templates
Component Labeling

- Assign different labels to pixels belonging to different regions (components)
  - connected components
  - not necessary for images with one region
Connected components on arbitrary graphs
Defining connectedness

4-connected neighbors

8-connected neighbors
Two-pass sequential algorithm for estimating connected components

Iterate through each ‘on’ pixel from top to bottom, left to right:
1. If there are no neighbors, assign pixel new label
2. Otherwise
   1. Find the neighbor with the smallest label and assign it to the current pixel
   2. Store equivalence between neighboring labels

Iterate through pixels again, relabeling each with the lowest equivalent label
Dilation

Dilation: “I should turn on if any of my neighbors are turned on”

Neighborhood is defined by a “structuring element” (SE)
Example: finding borders of objects

Image

Dilated with 3x3 mask

Dilated minus Image
Grayscale dilation

$$F[x, y] = \max_{(u,v) \in SE} I[x + u, y + v]$$
Erosion

“I should turn off if any of my neighbors are off”

\[ F[x, y] = \min_{(u,v) \in SE} I[x + u, y + v] \]
Opening

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

Before opening

After opening
Closing

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

Before closing

After closing
Application: segmentation of a liver

Application: blob tracking

Absolute differences from frame to frame

Threshold

Erode

Application: blob tracking

• Background subtraction + blob tracking

Application: segmentation of a liver

Some useful features can be extracted once we have connected components, including

• Area
• Centroid
• Extremal points, bounding box
• Circularity
• Spatial moments

Region Filling

Extract Largest Region

Boundary Peeling

Application by Jie Zhu, Cornell University
MATLAB’s binary image functions

(Online help documentation sometimes better than help comments)

bwconncomp.m

bwmorph.m

regionprops.m

Can easily find the bounding box coordinates, area, best-fit ellipse, etc.. of each connected component
Outline

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

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

• Matching binary edge templates
Edge matching

Naive approach for matching binary template and image:
Count the number of matching edges (AND)
What’s wrong with this?
Chamfer matching

- Chamfer score is average nearest distance from template points to image points
- Nearest distances are readily obtained from a computationally inexpensive “distance transform” image
Distance transform

Let P be the set of ‘on’ pixels. For each location (x,y), find distance to nearest point (x’,y’) in B

\[ DT[i] = \min_{j \in B} dist(i, j) \]

\[ i = \begin{bmatrix} x \\ y \end{bmatrix} \]
Chamfer matching

T: set of template ‘on’ points \{(x,y)\}

B: set of ‘on’ point in binary image

\[ D(T, B) = \frac{1}{|T|} \sum_{i \in T} \min_{j \in B} \text{dist}(i, j) \]

What distance function to use?
Different Distance Measures

- **Euclidean distance (L₂ norm)**
  \[ \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ...} \]

- **City block distance (L₁ norm)**
  \[ |x_1 - y_1| + |x_2 - y_2| + ... \]

- **Chessboard distance (Lₘ norm)**
  \[ \max(|x_1 - y_1|, |x_2 - y_2|, ...) \]
L\(_1\) Distance Transform Algorithm

- Two pass O(n) algorithm for 1D L\(_1\) norm (just distance and not source point)

  1. **Initialize**: For all \( j \)
     \[ D[j] \leftarrow 1_p[j] \]

  2. **Forward**: For \( j \) from 1 up to \( n-1 \)
     \[ D[j] \leftarrow \min(D[j], D[j-1]+1) \]

  3. **Backward**: For \( j \) from \( n-2 \) down to 0
     \[ D[j] \leftarrow \min(D[j], D[j+1]+1) \]

\[ \begin{array}{cccccccc}
\infty & 0 & \infty & 0 & \infty & \infty & \infty & 0 & \infty \\
\infty & 0 & 1 & 0 & 1 & 2 & 3 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 \\
\end{array} \]
**L₁ Distance Transform**

- 2D case analogous to 1D
  - Initialization
  - Forward and backward pass
    - Forward pass adds one to closest above and to left, takes min with self
    - Backward pass analogous below and to right
L1 distance transform

Top-Left to Bottom-Right (Pass 1):

$$D[i, j] = \min(D[i, j], 1 + D[i - 1, j], 1 + D[i, j - 1])$$

Bottom-Right to Top-Left (Pass 2):

$$D[i, j] = \min(D[i, j], 1 + D[i + 1, j], 1 + D[i, j + 1])$$
L2 distance transform

Use 2-pass algorithm to keep track of lower envelope of 1-d quadratics

Felzenswalb and Huttenlocher
Distance transform

Let $P$ be the set of ‘on’ pixels. For each location $(x,y)$, find distance to nearest point $(x’,y’)$ in $P$

$$DT[x] = \min_{x’ \in P} \text{dist}(x, x’)$$

$x = \begin{bmatrix} x \\ y \end{bmatrix}$
Chamfer Matching

- Distance image provides a smooth cost function
- Efficient searching techniques can be used to find correct template
Chamfer Matching
Chamfer Matching
Chamfer Matching
Chamfer Matching
Chamfer Matching
Outline

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

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

• Matching binary edge templates