Lecture 3: Binary image analysis

Thursday, Sept 6

• Sudheendra's office hours
  – Mon, Wed 1-2 pm
  – ENS 31NR

• Forsyth and Ponce book

Binary images

• Two pixel values
• Foreground and background
• Regions of interest

![Binary image example]

Constrained image capture setting

Documents, text

Medical, bio data
Intermediate low-level cues
- Edges
- Motion
- Orientation

Shape
- Silhouette
- Medial axis

Outline
- Thresholding
- Connected components
- Morphological operators
- Region properties
  - Spatial moments
  - Shape
- Distance transforms
  - Chamfer distance

Thresholding
- Grayscale -> binary mask
- Useful if object of interest’s intensity distribution is distinct from background
  
  \[ F(x, y) = \begin{cases} 
  1 & \text{if } F(x, y) \geq T \\
  0 & \text{otherwise}. 
  \end{cases} \]
  
  \[ F(x, y) = \begin{cases} 
  1 & \text{if } T_1 \leq F(x, y) \leq T_2 \\
  0 & \text{otherwise}. 
  \end{cases} \]
  
  \[ F(x, y) = \begin{cases} 
  1 & \text{if } F(x, y) = Z \\
  0 & \text{otherwise}. 
  \end{cases} \]

Selecting thresholds
- Partition a bimodal histogram
- Fit Gaussians
- Dynamic or local thresholds

A nice case: bimodal intensity histograms

A nice case: bimodal intensity histograms

- Example
- Thresholding a bimodal histogram

- Otsu method (1979): automatically select threshold by minimizing the weighted within-group variance of the two groups of pixels separated by the threshold.

Not so nice cases

- Threshold selection is an art, not a science

Connected components

- Identify distinct regions

Connected components

- Various algorithms to compute
  - Recursive (in memory)
  - Two rows at a time (image not necessarily in memory)
  - Parallel propagation strategy

Connectedness

- Which pixels are considered neighbors
Recursive connected components

- Find an unlabeled pixel, assign it a new label
- Search to find its neighbors, and recursively repeat to find their neighbors till there are no more
- Repeat

[Demo](http://www.cosc.canterbury.ac.nz/mukundan/covn/Label.html)

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

Morphological operators

- Dilation
- Erosion
- Open, close

Dilation

- Expands connected components
- Grow features
- Fill holes

Structuring elements

- Masks of varying shapes used to perform morphology
- Scan mask across foreground pixels to transform the binary image
Dilation / Erosion

- Dilation: if current pixel is foreground, set all pixels under S to foreground in output (OR)
- Erosion: if every pixel under S is foreground, leave as is; otherwise, set current pixel to background in output

Example for Dilation (1D)

\[ g(x) = f(x) \oplus SE \]

<table>
<thead>
<tr>
<th>Input image</th>
<th>1 0 0 0 1 1 1 0 1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structuring Element</td>
<td>1 1 1</td>
</tr>
<tr>
<td>Output Image</td>
<td>1 1 0 0 1 0 1 1 0 1 1</td>
</tr>
</tbody>
</table>

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

Adapted from T. Moeslund
Example for Dilation

Input image: 1 0 0 0 1 1 0 1
Structuring Element: 1 1 1
Output Image: 1 1 0 1 1 1 1 1

Example for Dilation

Input image: 1 0 0 0 1 1 0 1
Structuring Element: 1 1 1
Output Image: 1 1 0 1 1 1 1 1

The object gets bigger and holes are filled!

Example for Dilation

Input image: 1 0 0 0 1 1 0 1
Structuring Element: 1 1 1
Output Image: 1 1 0 1 1 1 1 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 0 0 0 1 1 1 1

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

Before erosion
After erosion
Example for Erosion (1D)

Input image: 1 0 0 1 1 1 0 1 1
Structuring Element: 1 1 1
Output Image: 0 0

$g(x) = f(x) \oslash SE$

Example for Erosion

Input image: 1 1 1 1 1 1 1 1 1
Structuring Element: 1 1 1
Output Image: 0 0 0

Example for Erosion

Input image: 1 0 0 1 1 1 0 1 1
Structuring Element: 1 1 1
Output Image: 0 0 0

Example for Erosion

Input image: 1 1 1 1 1 1 0 1 1
Structuring Element: 1 1 1
Output Image: 0 0 0 0

Example for Erosion

Input image: 1 0 0 1 1 1 0 1 1
Structuring Element: 1 1 1
Output Image: 0 0 0 0

Example for Erosion

Input image: 1 0 0 1 1 1 0 1 1
Structuring Element: 1 1 1
Output Image: 0 0 0 0 1 0
Example for Erosion

Input image
1 0 0 0 1 1 1 0 1
Structuring Element
1 1 1
Output Image
0 0 0 0 1 0 0 0 1

The object gets smaller

Example for Erosion

Input image
1 0 0 0 1 1 1 0 1
Structuring Element
1 1 1
Output Image
0 0 0 0 0 1 0 0 0 1

Example for Erosion

Input image
1 0 0 0 1 1 1 0 1
Structuring Element
1 1 1
Output Image
0 0 0 0 0 1 0 0 0 1

Example for Erosion

Input image
1 0 0 0 1 1 0 1 1
Structuring Element
1 0 0 1
Output Image
1 1

Dilation / Erosion

• Dilation: if current pixel is foreground, set all pixels under S to foreground in output (OR)
• Erosion: if every pixel under S is foreground, leave as is; otherwise, set current pixel to background in output

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: blob tracking

Absolute differences from frame to frame

Threshold

Erode

Application: blob tracking

• Background subtraction + blob tracking

Region properties

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

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

- We denote the set of pixels in a region by \( R \).
- Assuming square pixels:
  area: \[ A = \sum_{p \in R} 1 \]
  centroid: \[ \bar{x} = \frac{1}{A} \sum_{p \in R} x \]
  \[ \bar{y} = \frac{1}{A} \sum_{p \in R} y \]
- \((\bar{x}, \bar{y})\) is generally not a pair of integers.

Circularity

A second measure uses variation off a circle centrality (2):

\[ C_R = \frac{m_2}{m_0} \]

where \( m_0 \) and \( m_2 \) are the mean and variance of the distance from the centroid of the shape to the boundary pixels \((r, \theta)\).

Mean radial distance:

\[ m_0 = \frac{1}{N} \sum_{(r, \theta) \in S} \| (x, y) - (r, \theta) \| \]

Variance of radial distance:

\[ m_2 = \frac{1}{N} \sum_{(r, \theta) \in S} \| (x, y) - (r, \theta) \|^2 \]

Invariant descriptors

Often want features independent of position, orientation, scale.

Central moments

- 2nd central moment: variance
- 3rd central moment: skewness
- 4th central moment: kurtosis

Central moments

- Invariant to translation of \( S \).

Axis of least second moment

- Invariance to orientation?
- Need a common alignment

Axis for which the squared distance to 2d object points is minimized.
Distance transform

- Image reflecting distance to nearest point in point set (e.g., foreground pixels).

Distance transform (1D)

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

1. Initialize: For all $j$
   \[ D[1] \leftarrow 1 \]
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
   \[ D[j] \leftarrow \min(D[j], D[j+1]+1) \]

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
- Note nothing depends on $0, \infty$ form of initialization
  - Can “distance transform” arbitrary array

Chamfer distance

- Average distance to nearest feature

\[ D_{\text{chamfer}}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t) \]
Generalized distance transforms

- Same forward/backward algorithm applicable with different initialization
- Initialize with function values $F(x,y)$:

$$D(p) = \min_{q \neq l} (\|p - q\| + F(q))$$

Distance Transform vs. Generalized Distance Transform

- For general $F(p)$
  $$D(p) = \min_q (\|p - q\| + F(q))$$

  is Generalized Distance Transform of $F(p)$

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

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);
- [D,L] = bwdist(BW,METHOD);

- Everything is matrix

Tutorial adapted from W. Freeman, MIT 6.896
Matrix index

```matlab
A = magic(4);
>> A
```

Manipulate matrices

```matlab
>> whos
```

Scripts and functions

- Scripts are m-files containing MATLAB statements
- Functions are like any other m-file, but they accept arguments
- Name the function file the same as the function name

```matlab
function y = myfunction(x)
% Function of one argument with one return value
y = x + 1;
end
```

Try to code in matrix ways

```matlab
a = [1 2 3; 4 5 6; 7 8 9];
>> a
```

Manipulate matrices

```matlab
>> whos
```

Try to code in matrix ways

```matlab
a = [1 2 3; 4 5 6; 7 8 9];
>> a
```

• whos
• help
• lookfor
• clear / clear x
• save
• load