Image Segmentation

- Gray-level thresholding
- Supervised vs. unsupervised thresholding
- Binarization using Otsu’s method
- Locally adaptive thresholding
- Maximally stable extremal regions
- Color-based segmentation
- Region labeling and counting
- Region moments
Gray-level thresholding

Original image

$Peter f[x,y]$  

Thresholded

$Peter m[x,y]$  

$f[x,y] \cdot m[x,y]$  

How can holes be filled?
How to choose the threshold?

Normalized pixel count vs. Gray level

Foreground

Background
Unsupervised thresholding

- Idea: find threshold $T$ that minimizes *within-class variance* of both foreground and background (same as k-means)

$$\sigma_{\text{within}}^2(T) = \frac{N_{\text{Fgrnd}}}{N} \sigma_{\text{Fgrnd}}^2(T) + \frac{N_{\text{Bgrnd}}}{N} \sigma_{\text{Bgrnd}}^2(T)$$

- Equivalently, maximize *between-class variance*

$$\sigma_{\text{between}}^2(T) = \sigma^2 - \sigma_{\text{within}}^2(T)$$

$$= \left( \frac{1}{N} \sum_{x,y} f^2(x,y) - \mu^2 \right) - \frac{N_{\text{Fgrnd}}}{N} \left( \frac{1}{N_{\text{Fgrnd}}} \sum_{x,y \in \text{Fgrnd}} f^2(x,y) - \mu_{\text{Fgrnd}}^2 \right) - \frac{N_{\text{Bgrnd}}}{N} \left( \frac{1}{N_{\text{Bgrnd}}} \sum_{x,y \in \text{Bgrnd}} f^2(x,y) - \mu_{\text{Bgrnd}}^2 \right)$$

$$= -\mu^2 + \frac{N_{\text{Fgrnd}}}{N} \mu_{\text{Fgrnd}}^2 + \frac{N_{\text{Bgrnd}}}{N} \mu_{\text{Bgrnd}}^2 = \frac{N_{\text{Fgrnd}}}{N} \left( \mu_{\text{Fgrnd}} - \mu \right)^2 + \frac{N_{\text{Bgrnd}}}{N} \left( \mu_{\text{Bgrnd}} - \mu \right)^2$$

$$= \frac{N_{\text{Fgrnd}}(T) \cdot N_{\text{Bgrnd}}(T)}{N^2} \left( \mu_{\text{Fgrnd}}(T) - \mu_{\text{Bgrnd}}(T) \right)^2$$

[Otsu, 1979]
Unsupervised thresholding (cont.)

- **Algorithm:** Search for threshold $T$ to maximize

\[
\sigma_{\text{between}}^2(T) = \frac{N_{\text{Fgnd}}(T) \cdot N_{\text{Bgrnd}}(T)}{N^2} \left( \mu_{\text{Fgnd}}(T) - \mu_{\text{Bgrnd}}(T) \right)^2
\]

- Useful recursion for sweeping $T$ across histogram:

\[
\begin{align*}
N_{\text{Fgnd}}(T + 1) &= N_{\text{Fgnd}}(T) + n_T \\
N_{\text{Bgrnd}}(T + 1) &= N_{\text{Bgrnd}}(T) - n_T \\
\mu_{\text{Fgnd}}(T + 1) &= \frac{\mu_{\text{Fgnd}}(T) N_{\text{Fgnd}}(T) + n_T T}{N_{\text{Fgnd}}(T + 1)} \\
\mu_{\text{Bgrnd}}(T + 1) &= \frac{\mu_{\text{Bgrnd}}(T) N_{\text{Bgrnd}}(T) - n_T T}{N_{\text{Fgnd}}(T + 1)}
\end{align*}
\]

[Otsu, 1979]
Unsupervised thresholding (cont.)

\[ T = 112 \]
Unsupervised thresholding (cont.)

\[ T = 56 \]
Unsupervised thresholding (cont.)

$T = 168$
Unsupervised thresholding (cont.)

\[ T = 144 \]
Sometimes, a global threshold does not work

Original image

Thresholded with Otsu’s Method
Locally adaptive thresholding

- Slide a window over the image
- For each window position, decide whether to perform thresholding
  - Thresholding should not be performed in uniform areas
  - Use variance or other suitable criterion
- Non-uniform areas: apply Otsu’s method (based on local histogram)
- Uniform areas: classify the entire area as foreground or background based on mean value
Locally adaptive thresholding (example)

Non-uniform areas

Local threshold values

Locally thresholded result
Maximally stable extremal regions

- Extremal region: any connected region in an image with all pixel values above (or below) a threshold

- Observations:
  - Nested extremal regions result when the threshold is successively raised (or lowered).
  - The nested extremal regions form a “component tree.”

- Key idea: choose thresholds $\theta$ such that the resulting bright (or dark) extremal regions are nearly constant when these thresholds are perturbed by $+/-\Delta$

$\rightarrow$ “maximally stable” extremal regions (MSER)

[Matas, Chum, Urba, Pajdla, 2002]
MSERs: illustration

Local minimum of \[ \frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_{\theta}} \] → MSER

[Matas, Chum, Urba, Pajdla, 2002]
Level sets of an image

\[ f[x, y] \]

Image

\[ f[x, y] > 8 \]

Level Set
Component tree of an image

Local minima of sequence

\[
\frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_\theta}
\]

\(\theta = \Delta, \Delta + 1, \ldots \rightarrow \text{MSERs}\)
**MSER: examples**

- Dark MSERs, $\Delta=15$
- Original image
- Bright MSERs, $\Delta=15$
MSER: examples

Dark MSERs, $\Delta=15$

Original image

Bright MSERs, $\Delta=15$
Supervised thresholding

Normalized pixel count vs. Gray level

Foreground

Background

Error probability

Normalized pixel count

Gray level
Supervised thresholding

If errors $\text{BG} \rightarrow \text{FG}$ and $\text{FG} \rightarrow \text{BG}$ are associated with different costs:

“Bayes minimum risk detector” is optimal.
Multidimensional MAP detector

- **Training**
  - Provide labelled set of training data
  - Subdivide n-dimensional space into small bins
  - Count frequency of occurrence for each bin and class in training set, label bin with most probable class
  - (Propagate class labels to empty bins)
- **For test data:** identify bin, look up the most probable class
MAP detector in RGB-space

Original image

Skin color detector

Five training images

Non-skin Samples

Skin Samples
Linear discriminant function

To segment image with $n$ components $f_i$, $i = 1, 2, \ldots, n$ into two classes, perform test

$$\sum_i w_i f_i + w_0 \geq 0 \quad ?$$

- Categories are separated by hyperplane in $n$-space
- Numerous techniques to determine weights $w_i$, $i = 0, 1, 2, \ldots, n$, see, e.g., [Duda, Hart, Stork, 2001]
- Can be extended to the intersection of several linear discriminant functions
- Can be extended to multiple classes
Chroma keying

Extract "blueness" for each pixel

\[ a \]

\[ a \]

\[ \sum \]

\[ 1 - a \]
Landsat image processing

Original Landsat image false color picture out of bands 4,5,6

Water area segmented and enhanced to show sediments

Region labeling and counting

- How many fish in this picture?

- Which pixels belong to the same object (region labeling)?
- How large is each object (region counting)?
Definition: a *region* is a set of pixels, where each pixel can be reached from any other pixel in the region by a finite number of steps, with each step starting at a pixel and ending in the neighborhood of the pixel.

Typically, either definition leads to the same regions, except when a region is only connected across diagonally adjacent pixels.
Region labeling algorithm (4-neighborhood)

- Loop through all pixels $f [x,y]$, left to right, top to bottom
- If $f [x,y] = 0$, do nothing.
- If $f [x,y] = 1$, distinguish 4 cases

- Second pass through image to replace equivalent label by the same label.
Region labeling example (4-neighborhood)

List of Region Labels

1 → 2 → 3

All three labels are equivalent, so merge into single label.
Region labeling example (4-neighborhood)

List of Region Labels

1
Example: region labeling

Thresholded image

20 labeled regions
Region counting algorithm

- Measures the size of each region
- Initialize $counter[label]=0$ for all label
- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$, increment $counter[label[x,y]]$
Small region removal

- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$ and $\text{counter}[\text{label}[x,y]]<S$, set $f[x,y]=0$

- Removes all regions smaller than $S$ pixels
Hole filling as dual to small region removal

Mask with holes

After NOT operation, (background) region labeling, small region removal, and second NOT operation
Region moments

- Raw moments

\[ M_{pq} = \sum_{x,y \in \text{Region}} x^p y^q \]

- Central moments

\[ \mu_{pq} = \sum_{x,y \in \text{Region}} (x - \bar{x})^p (y - \bar{y})^q \quad \text{with} \quad \bar{x} = \frac{M_{10}}{M_{00}} \quad \text{and} \quad \bar{y} = \frac{M_{01}}{M_{00}} \]

- Region orientation and eccentricity: calculate eigenvectors of covariance matrix

\[
\begin{bmatrix}
\mu_{20} & \mu_{11} \\
\mu_{11} & \mu_{02}
\end{bmatrix}
\]
Example: Detecting bar codes

Original Image
Example: Detecting bar codes

Locally adaptive thresholding
Example: Detecting bar codes

Locally adaptive thresholding

Filtering by eccentricity
Example: Detecting bar codes

Locally adaptive thresholding

Filtering by eccentricity

Filtering by major axis length
Example: Detecting bar codes

- Locally adaptive thresholding
- Filtering by eccentricity
- Filtering by major axis length
- Filtering by orientation