Image Segmentation
Lecture 05

Milan Gavrilovic
milan@cb.uu.se

Centre for Image Analysis
Uppsala University

Computer Assisted Image Analysis
2008-04-14
Chapter 10.1 – 10.2.5, and 10.3 – 10.5 in Gonzales-Woods.
Previous Lectures

Image processing

- Point processing (Spatial domain, pixel-wise)
- Local neighborhoods (Spatial domain, filtering)
- Fourier transform (Frequency domain, filtering)

Image analysis

- Segmentation (Easier said, than done!)
Fundamental steps in problem solving using digital image analysis

Problem

Image acquisition

Preprocessing

Segmentation

Representation and Description

Classification, recognition, interpretation

Solution
Segmentation
Divide the image contents into its constituent regions

- **Full segmentation**: Individual objects are separated from the background and given individual ID numbers (labels).
- **Partial segmentation**: The amount of data is reduced (usually by separating objects from the background) to speed up further processing.

The problem can be made much easier if solved in cooperation with the constructor of the imaging system (choice of sensors, illumination, background, etc.).

... is often the most difficult problem to solve in image analysis; there is no universal solution!
Four types of segmentation

- **Thresholding**: Based on pixel intensities, often using the shape of the histogram for automation.
- **Edge-based**: Search for discontinuities in the image, and try to connect objects or borders (often by a region-based technique).
- **Region-based**: Group similar pixels: region growing, merge & split.
- **Match-based**: Comparison to a template (more in lecture 9).
Segmentation
based on two basic properties of intensity values

- **Similarity**: thresholding, region growing, split and merge.
- **Discontinuity**: Edges.
Thresholding

A threshold $T$, a gray level intensity, classifies every pixel as belonging to objects (foreground) or background.

- **Fixed thresholds**: The same value is used for all images.
- **Optimal thresholding**: Based on the shape of the current image histogram. Searching for valleys, Gaussian distributions, etc.
- **Local (or dynamic) thresholding**: The image is divided into overlapping sections which are thresholded one by one.

Lighting conditions are extremely important, and it will only work under very controlled circumstances.
Global thresholding

- (a) Original image.
- (b) Image histogram.
- (c) Result of global thresholding with $T$ midway between the maximum and minimum gray levels.

Thresholded binary image, all pixel values are 0 (object) or 1 (background).
Example method

1. Choose initial threshold $T_0$.
2. Then $f(x, y) < T_0$ is background, and $f(x, y) \geq T_0$.
3. Calculate mean for background $\mu_{bg}$, and foreground $\mu_{fg}$.
4. Set next threshold $T_i = 0.5 \times (\mu_{bg} + \mu_{fg})$.
5. Repeat 2 – 4 until stopping criterion, $T_i = T_{i-1}$, is fulfilled.
Threshold examples
Local thresholding

Original image.

Image segmented by local thresholding.
Once the objects are separated from the background and from each other, each separate object is given an ID number.

How many objects are there in the image?
Labeling - the two-pass algorithm

Set background pixels to zero and object pixels to infinity.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>∞</td>
<td>0</td>
<td>0</td>
<td>∞</td>
</tr>
<tr>
<td>0</td>
<td>∞</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Mask for 4-adjacent pixels

Mask for 8-adjacent pixels
Labeling - the two-pass algorithm

0. Set background pixels to zero and object pixels to infinity (or maximum intensity, e.g., 255).

1. The first pass: For each pixel $p$, from the top-left to the bottom-right pixel in the image:
   - If ($p$ is not labeled) and (neighbours are not labeled) then (assign new label to $p$)
   - If ($p$ is not labeled) and (one of the neighbours is labeled) then (assign old label to $p$)
   - If (neighbours have different labels) then (add values to equality table)

2. Update equality table and, in the second pass, update the labels.
Labeling - the two-pass algorithm
An example for 4-adjacent pixels

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Equality table:
- 2-3
- 3-4
- 4-5

Updated eq. table:
- 2-3
- 2-4
- 2-5

After the first pass:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

After the second pass:
Edge-based segmentation

1. First apply an edge/point detector
   - gradient operator (Sobel)
   - second derivative (Laplace)

2. Second “Threshold” edges and link and segment by:
   - local processing or
   - Hough transform (global)

3. Or... use, e.g., region growing or watershed to transform the edge image into a full segmentation.
Edge-based segmentation

Find edges of a given thickness in a given direction using a designed filter. E.g., Sobel $-45^\circ$

Find edges of a given thickness in a given direction using a designed filter. E.g., Sobel $-45^\circ$

\[
\begin{array}{ccc}
0 & -1 & -2 \\
1 & 0 & -1 \\
2 & 1 & 0 \\
\end{array}
\]

Result after filtering with Sobel filter for $-45^\circ$ and thresholding on gradient magnitude (can be further processed by, e.g., removal of single points).
The Hough transform

After edges are extracted from image, the Hough transform can be used for transforming local information into global information in parameter space and search for common “patterns”. Brief description of the algorithm:

- Represent all edge points \((x_i, y_i)\) as lines in a \((a, b)\) parameter space (where \(a\), and \(b\) comes from the equation for a straight line \(y = ax + b\)).
- Look for points \((a', b')\) were many lines meet = edge points from a line.

Other parameter spaces can be used to detect, e.g., the position of circles with a given radius. The Hough transform can also be applied locally to find curves that are only locally straight.
The Hough transform
What is it good for?

- When the patterns searched for are sparsely digitized, have holes and/or the images are noisy.
- Finding objects described with a small number of parameters (i.e. generally low complexity)
- Finding straight lines
- Finding circles
Region based segmentation
Top-down approach

Region splitting and merging

1. Set up some criteria for what is a uniform area (e.g., mean, variance, bi-modality of histogram, texture, etc.).
2. Start with the full image and \textit{split} it into four sub-images.
3. Check each sub-image. If not uniform, divide into four new sub-images.
4. Compare regions with neighboring regions and \textit{merge} if uniform.

Repeat 2 – 4 until nothing more happens. The method is also called “quad-tree” division.
Region based segmentation

Bottom-up approach

Region growing

1. Find starting points.
2. Include neighboring pixels with similar features (gray level, texture, color, etc.).
3. Continue until all pixels have been included with one of the starting points.

Problems: Non trivial to find good starting points, difficult to automate, need good criteria for similarity. A method is *watershed*. 
Watershed
A *kind* of region growing

Think of the gray level image as a landscape. Let water rise from the bottom of each valley (the water from each valley is given its own label). As soon as the water from two valleys meet, build a dam, or watershed. These watersheds will then define the borders between different regions.

Can be used directly on the image, on a edge enhanced image or on a distance transformed image.
Watershed

Example of watershed applied directly on gray level image
**Watershed**

Example of watershed applied to a binary image with “round” objects

![Original Image](image1.png)

![Distance Transformed Image](image2.png)

![Distance Coded as Intensity](image3.png)

![Inverse](image4.png)

![Intensity Landscape](image5.png)

![Result of Watershed Segmentation](image6.png)
The model is that every object contain a “seed”, e.g., every cell has a cell nucleus which can be detected by thresholding and shape-based watershed segmentation. Using the nuclei as seeds, the cytoplasms are easy to find. Fully automatic analysis at the speed of 100 cells/second.
Watershed
Find individual nuclei in (2D & 3D) tissue samples

Model: A nucleus is a connected region that is ...

1. Brighter than the local background ...
2. Surrounded by a sharp edge ...
3. Convex
Watershed
Extension to 3D

Same approach as for 2D, but extending all $3 \times 3$ operators to $3 \times 3 \times 3$.

Maximum intensity projection (MIP) of a $256 \times 256 \times 100$ volume image.

Segmentation result: Each cell (in 3D) has its own label (ID number) and measurements can be made on the individual cells.
Match-based segmentation

Compare a “template” to the underlying image to find objects with a certain intensity distribution or shape (detailed in a later lecture).
Reading Instructions
Chapters for tomorrow's lecture

- Chapter 9.1 – 9.5.4 in Gonzales-Woods.