Image analysis

Pre-processing, image enhancement, morphological operations
Binary operations
Classification and matching
Feature extraction

Segmentation

“Find the objects!”

Image

Data

Segmentation

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 background) to speed up the further processing.

Segmentation is often the most difficult problem to solve in image analysis; there is no universal solution!

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

Three types of segmentation:

Thresholding - Based on pixel intensities, often using the shape of the histogram for automation

Region-based – Group similar pixels: region growing, merge & split.

Edge-based - Search for discontinuities in the image, and try to connect objects or borders (often by a region based technique).

Thresholding

A threshold $T$ for pixel intensity classifies every pixel as belonging to objects (foreground) or background.

Fix thresholds: the same value is used in all images

Optimal thresholding: based on the shape of the current image histogram. Search 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.

Original image

Thresholded binary image, all pixel values are 0 (object) or 1 (background).
Labeling: identifying objects

- Once the objects are separated from each other, each separate object is given an ID number.
- Use a two-pass algorithm

It is important to choose a suitable neighborhood (lecture 3)

Edge-based Segmentation

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

2. "Threshold" edges to find pixels fulfilling some edge criteria
   - linking pixels with high gradient amplitude
   - Hough transform

3. Use ex. region growing or watershed to transform the edge image into a full segmentation.

The Hough transform

Transform local information into global information in parameter space and search for common "patterns"

Algorithm when searching for straight lines:
1. Detect edges and threshold on e.g. gradient magnitude.
2. Observe two foreground pixels (x1, y1) and (x2, y2) that may belong to the same straight line y=ax+b. A infinite number of straight lines, with varying parameters a and b, will cross each point in the image.
3. Observe parameter space b=ax-y. All pairs (a,b) for a given point (x,y) will end up on a straight line, and the parameter pair fitting a line through (x1,y1) and (x2,y2) will intersect in a point (a',b'). The more points that belong to this line, the more lines will cross at this point (a',b') in parameter space.
4. Detect multiple crossings in parameter space using accumulator cells.

Hough transform

Problems with vertical lines (a,b approaching infinity) are avoided by representing each point (x,y) with an angle t and a radius r.
We will then have xcos t+y= r.
and each pixel will be described by a curve in the r-t plane, where r can vary in the interval +/-90 degrees.
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.
Region based segmentation

region splitting and merging (top-down method)
1. Set up some criteria for what is a uniform area (e.g., mean, variance, bimodality of histogram, texture, etc).
2. Start with the full image and split it into 4 sub-images.
3. Check each sub-image. If not uniform, divide into 4 new sub-images.
4. After each iteration, compare touching regions with neighboring regions and merge if uniform.

The method is also called “quadtree” division.

region growing (bottom-up method)
1. Find starting points.
2. Include neighboring pixels with similar features (grayscale, texture, color).
3. Continue until all pixels have been included with one of the starting points.

Problems:
- Not trivial to find good starting points, difficult to automate.
- Need good criteria for similarity.

Ex. 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 an edge enhanced image or on a distance transformed image.

Example of watershed directly on a gray-level image (AFM image of wood fibers)

Example of watershed on a binary image with “round” objects

Seed watershed

Example: Every object contains 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 cytoplasm is easy to find.

Fully automatic analysis at the speed of 100 cells/second.
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

Extension to 3D
Same approach as for 2D, extending all 3x3 operators to 3x3x3.

Maximum intensity projection of 256x256x100 image
Segmentation result: Each cell (in 2D) has its own ID number and measurements can be made on the individual cells.