Lecture 3: Filtering II

Computer Assisted Image Analysis II, Spring 2016
Nataša Sladoje
natasa.sladoje@it.uu.se
Summary of previous lecture

- Gaussian filters for:
  - Smoothing
  - Derivatives
  - Laplace operator

- Non-linear filters for edge-preserving smoothing

- Smoothing filters for:
  - Noise reduction
  - Image simplification
  - Shading correction
  - Edge sharpening

- First order derivatives used for edge detection
- Second order derivatives used for line detection
Today’s lecture: Filtering II

Filtering can be used for analysis & detection

Special Applications of Filtering:
  - Scale spaces
  - Canny’s edge detector
  - Hough transform (for line detection)
  - Radon transform
  - Template matching
Scale space
Scale spaces

- Object appear differently depending on the scale of observation.
- Choosing the right scale in image processing is difficult, and problem dependent.
- **Scale-space representation** of an image considers the image at multiple (all) scales.
- Smoothing removes finer details and leaves bigger structures.
- Combine smoothing at many scales to detect and analyze structures at different scales.
- Scale is an additional dimension to an image.
Scale spaces

Scale space representation of an image:
one parameter family of smoothed images, with parameter being the size of a smoothing Gaussian kernel.

Scale parameter:
structures smaller than a certain size are smoothed away at a particular scale level.

Linear scale space:
Gaussian \( t = \sigma^2 \)
(has all desired properties!)
Scale pyramid

Signal is repeatedly smoothed and subsampled.

The sampling rate decreases at each pyramid level; resulting images are of smaller size.

The information content of the image is reduced.

Subsampling may create features which are not present in the original image.
Scale space representation of a signal

Parametric family of signals $f^t(x)$ where fine-scale information is successively attenuated

$\begin{align*}
  f'(x) \\
  f(x) = f^0(x)
\end{align*}$

Successive smoothing with a Gaussian filter

Zero-crossings of 2\textsuperscript{nd} derivative $f''(x)$
Fewer edges at coarser scales
Feature detection

- Characterize a feature of interest (blob, corner, edge, …) by some function of derivatives.
- Find maxima in scale space, to estimate size and approximate location of features.
- Use size information to refine the location of the feature.

Coarse to fine analysis in the scale space i.e., in the direction of decreasing value of sigma localizes large scale events. The structures (dis)appear with changing scale, but also - their locations change!
Edge detection
Edge detection

First order derivatives measure gradient. Gradient is strong at edges.

Zero crossing of second derivatives (Laplace) gives location of edge. Difficult to determine which zero-crossings are relevant.

Maximum of gradient magnitude, in direction of gradient, gives both location and direction of an edge

Canny’s edge detector is based on this.

By smoothing the image, we may perform scale-space analysis and choose an appropriate scale of the edges.
Canny’s edge detector

Optimal for step edges corrupted by white noise.

- **Detection** – important edges should not be missed and there should not be spurious responses.
- **Localization** – minimal distance between actual and located edges.
- **One response per edge** (both double responses, and noise corrupted responses are suppressed).
Canny’s edge detector

The algorithm in seven steps:

1. **Smoothing**: Blurring of the image to remove noise.
2. **Finding gradient magnitude and direction**: The edges are indicated by large gradient magnitudes.
3. **Non-maximum suppression**: Only local maxima in gradient directions are marked as edges.
4. **Double thresholding**: Potential edges are determined by thresholding. Strong edges and weak edges are found.
5. **Edge tracking by hysteresis**: Final edges are determined by suppressing all edges that are not connected to a very certain edge.
6. **Scale–space analysis**: Repeat the steps for increasing blur.
7. **Feature synthesis**: Combine information at multiple scales.
Canny’s edge detector

gradient magnitude

gradient direction
Canny’s edge detector

Non-maxima suppression
Non-maxima suppression

For each point $p$ check:

$$f(p) > a_1 \land f(p) \geq a_2$$

$$f(p) \geq a_1 \land f(p) > a_2$$

If condition not met, set $p$ to 0.

$a_1$ and $a_2$ computed by linear interpolation.
Canny’s edge detector

low threshold

hysteresis threshold

high threshold
Canny’s edge detector
Influence of the $\sigma$

$\sigma = 1$  
$\sigma = 2$  
$\sigma = 4$  
$\sigma = 8$

Use a scale space!
Hough transform
Hough transform

- Technique used for detecting imperfect instances of objects within a class of shapes by a voting procedure.
- Shapes should allow for parametric representation.
- Voting is performed in the parameter space.
- Shape detection becomes maxima detection.
- Originally, proposed for straight lines in 2D (2 parameters),
- Later, generalized to other shapes (more parameters)
2D Hough transform for lines

Each point in the parameter space represents one instance of the object in the image.
2D Hough transform for lines

Each point in the image gives evidence for all lines that go through that point.
2D Hough transform for lines

Each point in the image gives evidence for all lines that go through that point.
Hough transform example
2D Hough transform for lines

Each local maximum of the accumulator array defines a line defined by the corresponding parameters.

We may want a line segment, rather than a line across the whole image.

For each local maximum:
Look in the input image where there is support for this line.
Fill small gaps in the line
Is the resulting line long enough?
Hough transform example
Improving the Hough transform

Use additional knowledge about the points in the image, e.g. local gradient information
2D Hough transform for circles

Each point in the parameter space represents one instance of the shape in the image.
2D Hough transform for circles

Each point in the image gives evidence for all circles that go through that point. Accumulator is incremented for every possible centre point and every possible radius.
2D Hough transform for circles

Each point in the image gives evidence for all circles that go through that point. Votes accumulate for existing circles.
Improving the Hough transform

We can add information (e.g., gradient) and improve voting. Each point in the image gives evidence for most probable circles that go through that point.
Radon transform
The Radon transform
The Radon transform

For each point in the parameter space, examine how much evidence there is in the image.
The Radon transform

For each point in the parameter space, examine how much evidence there is in the image.

This can be done though one convolution for each value of the “position” and “distance” parameters.

This is identical to template matching.
Hough vs Radon

The resulting parameter space is identical.

Hough: efficient for binary images (sparse input)
Radon: efficient for grey-value images (few templates)

Hough: write paradigm
How a point in the source space maps onto data points in the destination space

Radon: read paradigm
How a point in destination space is obtained from the data in the source space

Radon transform: easier to define and sample the parameter space
Template matching
Template matching

Find positions in the image at which the template “fits best”.

Define matching criteria and optimize over different translations and rotations of a template.

Matching criteria (similarity or distance):
- Correlation
  (Normalized Correlation)
- Sum of squared differences
- Sum of absolute differences

\[
g(n) = \sum_k f(n+k)h(k)
\]
\[
g(n) = \frac{1}{N} \sum_k (f(n+k) - h(k))^2
\]
\[
g(n) = \frac{1}{N} \sum_k |f(n+k) - h(k)|
\]

Radon transform is equivalent to template matching.
Template matching

Image and template

1-NCC

Distance based on fuzzy set theory

SSD
Summary
Summary of today’s lecture

Special Applications of Filtering:

- Scale spaces
- Edge detection: Canny’s edge detector
- Hough transform
- Radon transform
- Template matching