Today

Early vision on a single image

- Image filtering
- Features
Early Vision on One Image

Two important topics

• **Image noise**: intrinsic property of the sensor (CCD) and independent of scene
  • Intensity noise – quantization and sensor
  • Positional noise – spatial sampling

• **Features**: characterizing the shape/appearance of the objects in the image
  • Edges, lines, curves, corners etc.
  • Image statistics: mean, variance, histogram etc.
  • Complex features
  • Widely used in many problems of computer vision including camera calibration, stereo, object detection/tracking/recognition, etc.
Image Filtering

Modifying the pixels in an image based on some function of a local neighborhood of the pixels

\[ f(p) : \]
- Linear function
  - Correlation
  - Convolution
- Nonlinear function
  - Order statistic (median)
Linear Filters

General process:
• Form new image whose pixels are a weighted sum of original pixel values, using the same set of weights at each point.

Properties
• Output is a linear function of the input
• Output is a shift-invariant function of the input (i.e. shift the input image two pixels to the left, the output is shifted two pixels to the left)

Example: smoothing by averaging
• form the average of pixels in a neighborhood

Example: smoothing with a Gaussian
• form a weighted average of pixels in a neighborhood

Example: finding an edge
Blur examples

impulse

original

coefficient

0.3

Pixel offset

2.4

filtered

data

edge

original

coefficient

0.3

Pixel offset

8

filtered
Gaussian Filter

\[ G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right) \]

\[ H(i, j) = \frac{1}{2\pi\sigma^2} \exp \left( -\frac{(i-k-1)^2 + (j-k-1)^2}{2\sigma^2} \right) \]

where \( H(i, j) \) is \((2k + 1) \times (2k + 1)\) array
Linear Filtering (Gaussian Filter)
Gaussian vs Average

Gaussian Smoothing

Smoothing by Averaging
Gaussian vs Average

Ringing artifacts near sharp transitions by averaging
Gaussian vs Average

Averaging

Gaussian

Figure from “Digital Image Processing”, Gonzalez and Woods
Smoothing Reduces Noise

Generally expect pixels to “be like” their neighbors
  • surfaces turn slowly
  • relatively few reflectance changes

Generally expect noise processes to be independent from pixel to pixel

Implies that smoothing suppresses noise, for appropriate noise models

Scale
  • the parameter in the symmetric Gaussian
  • as this parameter goes up, more pixels are involved in the average
  • and the image gets more blurred
  • and noise is more effectively suppressed
The effects of smoothing

Each row shows smoothing with Gaussians of different width; each column shows results of an image of different level of Gaussian noise.

\( \sigma = 0.05 \)  \( \sigma = 0.1 \)  \( \sigma = 0.2 \)

No smoothing

\( \sigma_H = 1 \)

\( \sigma_H = 2 \)
Other Nonlinear Filters for Noise Reduction

- Geometric mean filter
- Harmonic mean filter
- Contraharmonic mean filter
- Order statistic filters
  - Median filter
  - Max/min filter

Continuous noise model

Impulse noise model
Reading Assignments

Image Features

Establishing correspondence
• 3D reconstruction
• Object tracking

Object recognition & classification

What are good features?

• Robust to variations, e.g., viewpoint and illumination
• Reliable
Image Features

- Gradient and edges
- Corners
- Line and curves
- Textures
- Advanced features
  - Human crafted features
    - Haar, Gabor, SIFT, HOG, etc.
  - Learned features
    - Sparse coding
    - Deep features – features learned by deep learning
Gradients and Edges

Points of sharp change in an image are interesting:
- change in reflectance
- change in object
- change in illumination
- noise

Sometimes called edge points

General strategy
- determine image gradient
- mark points where gradient magnitude is particularly large w.r.t neighbors (ideally, curves of such points).

Boundary to separate background

Surface reflectance discontinuity

Surface normal discontinuity

Illumination discontinuity
Basic Edge Detection

The first-order derivative
\[
\frac{\partial f}{\partial x} = f'(x) = f(x + 1) - f(x)
\]

The second-order derivative
\[
\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x + 1) - 2f(x) + f(x - 1)
\]

Figure from “Digital Image Processing”, Gonzalez and Woods
Some Observations

1. First-order derivatives generally produce thicker edges

2. Second-order derivatives have a stronger response to fine details, such as thin lines, isolated points, and noise

3. Second-order derivatives produce a double-edge response at ramp and step transitions in intensity

4. The sign of the second-order derivative can be used to determine whether a transition into an edge is from light to dark or dark to light

Figure from “Digital Image Processing”, Gonzalez and Woods
First-order Derivative based Edge Detection

Gradient
\[ \nabla f(x, y) = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \]

Edge strength
\[ M(x, y) = \sqrt{g_x^2 + g_y^2} \]

\[ \approx |g_x| + |g_y| \]

Gradient direction \( \alpha(x, y) = \tan^{-1}\left[ \frac{g_y}{g_x} \right] \)

Edge direction \( \alpha - 90^\circ \)

Figure from “Digital Image Processing”, Gonzalez and Woods
First-order Derivative

General strategy
- determine image gradient
- mark points where gradient magnitude is particularly large w.r.t neighbors
**Masks for Calculating the Gradient (3x3)**

<table>
<thead>
<tr>
<th>Gradient in vertical/horizontal</th>
<th>Gradient in diagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prewitt</td>
<td></td>
</tr>
<tr>
<td><img src="image1" alt="Prewitt Mask" /></td>
<td><img src="image2" alt="Prewitt Mask" /></td>
</tr>
<tr>
<td>Sobel</td>
<td></td>
</tr>
<tr>
<td><img src="image1" alt="Sobel Mask" /></td>
<td><img src="image2" alt="Sobel Mask" /></td>
</tr>
</tbody>
</table>

Sobel operator performs edge detection and smoothing simultaneously.
Smoothing and Differentiation

Issue: noise

1. Image Smoothing
2. Detecting edge points
3. Edge localization

Figure from “Digital Image Processing”, Gonzalez and Woods
Smoothing and Differentiation

Solution

- Smooth before differentiation
- Two convolutions for smooth and differentiate?
- Actually, no - we can use a derivative of Gaussian filter
  - because differentiation is convolution, and convolution is associative

\[
\frac{\partial G_\sigma}{\partial x}
\]

\[
\frac{\partial G_\sigma}{\partial y}
\]

Derivative of Gaussian
The scale of the smoothing filter affects derivative estimates, and also the semantics of the edges recovered.
There are three major issues:

1) The gradient magnitude at different scales is different; which should we choose?
2) The gradient magnitude is large along thick trail; how do we identify the significant points?
3) How do we link the relevant points up into curves?
Edge
- Mark points along the curve where the magnitude is biggest.
- Look for a maximum along a slice normal to the curve (non-
maximum suppression). These points should form a curve.

Two questions:
- How do we define the slice direction
- Where is the next one

Gradient direction