Filtering & Edges

Introduction to Computer Vision
CSE 152
Lecture 8

Announcements

• Assignment 2: Posted on web site, due 4/29
• See links on web page for reading on binary image processing (e-reserves)
• Text for filtering

Binary System Summary

1. Acquire images and binarize (thresholding, color labels, etc.).
2. Possibly clean up image using morphological operators.
3. Determine regions (blobs) using connected component exploration
4. Compute position, area, and orientation of each blob using Moments
5. Compute features that are translation, scale, and orientation invariant using Moments (e.g., Eigenvalues of Normalized Moments).

Other ideas

• Binarization of color images
• Blob Tracking
  – Binary regions
  – State (e.g., x,y,orientation, scale, etc.)
  – Prediction
  – Data Association

(From Bill Freeman)

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 neighbourhood
- Example: smoothing with a Gaussian
  - Form a weighted average of pixels in a neighbourhood
- Example: finding a derivative
  - Form a weighted average of pixels in a neighbourhood

What is image filtering?

• Modify the pixels in an image based on some function of a local neighborhood of the pixels.

(From Bill Freeman)
Convolution

Image (I) * Kernel (K)

Note: Typically Kernel is relatively small in vision applications.

Convolution: \( R = K \ast I \)

\[
R(i, j) = \sum_{h=-m}^{m} \sum_{k=-m}^{m} K(h, k) I(i-h, j-k)
\]

Kernel size is \( m+1 \) by \( m+1 \)

m=2

Kernel size is \( m+1 \) by \( m+1 \)

\[
R(i, j) = \sum_{h=-m}^{m} \sum_{k=-m}^{m} K(h, k) I(i-h, j-k)
\]

m=2

Kernel size is \( m+1 \) by \( m+1 \)

\[
R(i, j) = \sum_{h=-m}^{m} \sum_{k=-m}^{m} K(h, k) I(i-h, j-k)
\]

m=2

Kernel size is \( m+1 \) by \( m+1 \)
Convolution: \( R = K \ast I \)

Kernel size is \( m+1 \) by \( m+1 \)

\[
R(i, j) = \sum_{h=-m}^{m} \sum_{k=-m}^{m} K(h, k) I(i-h, j-k)
\]
Impulse Response

Linear filtering (warm-up slide)

Linear filtering (no change)

Linear filtering

shift
Blurring

original

coefficient

Pixel offset

Blurred (filter applied in both dimensions)

Blur examples

impulse

coefficient

Pixel offset

original

filtered

Blurred examples

impulse

coefficient

Pixel offset

original

filtered

Linear filtering (warm-up slide)

original

filtered

Linear filtering (no change)

original

Filtered (no change)

Linear filtering

original

filtered

?
Properties of convolution

Let \( f, g, h \) be images and \( * \) denote convolution

\[
f * g(x, y) = \int \int f(x-u, y-v) g(u, v) \, du \, dv
\]

- Commutative: \( f * g = g * f \)
- Associative: \( f * (g * h) = (f * g) * h \)
- Linear: for scalars \( a \) & \( b \) and images \( f, g, h \)
  \( (af + bg) * h = a(f * h) + b(g * h) \)
- Differentiation rule
  \[
  \frac{\partial}{\partial x} (f * g) = \frac{\partial f}{\partial x} * g = f * \frac{\partial g}{\partial x}
  \]

Smoothing by Averaging

**Kernel**

![Kernel](image)

**Sharpening example**

![Sharpening example](image)

(remember blurring)

![Original](image) \[ \rightarrow \] \![Blurred](image)

**Sharpening**

![Sharpened](image) \[ \rightarrow \] \![Original](image)

**Smoothing by Averaging**

![Before](image) \[ \rightarrow \] \![After](image)
Filtering to reduce noise

• Noise is what we’re not interested in.
  – We’ll discuss simple, low-level noise today:
    Light fluctuations; Sensor noise; Quantization
    effects; Finite precision
  – Not complex: shadows; extraneous objects.
• A pixel’s neighborhood contains
  information about its intensity.
• Averaging noise reduces its effect.

Noise

• Simplest noise model
  – independent stationary
  additive Gaussian noise
  \( I(i, j) = \hat{I}(i, j) + N(i, j) \)
  – \( N(i,j) \) is a Gaussian
    Random Variable
  – the noise value at each
    pixel is given by an
    independent draw from
    the same normal
    probability distribution
• Issues
  – this model allows noise values
    that could be greater than
    maximum camera output or
    less than zero
  – for small standard deviations,
    this isn’t too much of a
    problem - it’s a fairly good
    model
  – independence may not be
    justified (e.g. damage to lens)
  – may not be stationary (e.g.
    thermal gradients in the ccd)

Average (Box) Filter

• Mask with positive
  entries, that sum 1.
• Replaces each pixel
  with an average of
  its neighborhood.
• If all weights are
  equal, it is called a
  BOX filter.

Smoothing by Averaging

Kernel:

An Isotropic Gaussian

• The picture shows a
  smoothing kernel
  proportional to
  \( \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \)
  (which is a reasonable
  model of a circularly
  symmetric fuzzy blob)
Smoothing with a Gaussian Kernel:

The effects of smoothing:

Each row shows smoothing with Gaussians of different width; each column shows different realizations of an image of gaussian noise.

Efficient Implementation

Both, the BOX filter and the Gaussian filter are separable:
- First convolve each row with a 1-D filter
- Then convolve each column with a 1-D filter.

For Gaussian kernels \( g_1(x) \) and \( g_2(x) \),
- If \( g_1 \) & \( g_2 \) respectively have variance \( \sigma_1^2 \) & \( \sigma_2^2 \)
- Then \( g_1 * g_2 \) has variance \( \sigma_1^2 + \sigma_2^2 \)

Other Types of Noise

- Impulsive noise
  - randomly pick a pixel and randomly set to a value
  - saturated version is called salt and pepper noise
- Quantization effects
  - Often called noise although it is not statistical
- Unanticipated image structures
  - Also often called noise although it is a real repeatable signal.

Some other useful filtering techniques

- Median filter
- Anisotropic diffusion

Median filters: Principle

Method:
1. rank-order neighborhood intensities in a window
2. take middle value
- non-linear filter
- no new grey levels emerge...
Median filters: Example for window size of 3

1,1,7,1,1,1,1
↓
?,1,1,1,1,1,?

The advantage of this type of filter is that it
Eliminates spikes (salt & pepper noise).

Median filters: analysis

median completely discards the spike,
linear filter always responds to all aspects
median filter preserves discontinuities,
linear filter produces rounding-off effects
DON’T become all too optimistic

Median filters: Gauss revisited

Comparison with Gaussian:
e.g. upper lip smoother, eye better preserved

Example of median

10 times 3 X 3 median

patchy effect
important details lost (e.g. ear-ring)
Linear Filters are templates

- Applying a filter at some point can be seen as taking a dot-product between the image and some vector
- Filtering the image is a set of dot products

Insight
- Filters look like the effects they are intended to find
- Filters find effects they look like