Filters, Features, Edges
Thursday, Sept 11

Last time
- Cross correlation
- Convolution
- Examples of smoothing filters
  - Box filter (averaging)
  - Gaussian

Convolution
- Convolution:
  - Flip the filter in both dimensions (bottom to top, right to left)
  - Then apply cross-correlation

\[ G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i - u, j - v] \]

\[ G = H \ast F \]

Notation for convolution operator

Predict the filtered outputs

Smoothing with a Gaussian
Parameter \( \sigma \) is the "scale" / "width" / "spread" of the Gaussian kernel, and controls the amount of smoothing.

```
for sigma=1:3:10
    h = fspecial('gaussian', fsize, sigma);
    out = imfilter(im, h);
    imshow(out);
    pause;
end
```

Practice with linear filters

Source: D. Lowe
Practice with linear filters

Original

Filtered (no change)

Source: D. Lowe

Practice with linear filters

Original

Shifted left by 1 pixel with correlation

Source: D. Lowe

Practice with linear filters

Original

Blur (with a box filter)

Source: D. Lowe

Practice with linear filters

Original

Source: D. Lowe
Practice with linear filters

Original

Sharpening filter
- Accentuates differences with local average

Source: D. Lowe

Filtering examples: sharpening

Before

After

Shift invariant linear system

- **Shift invariant:**
  - Operator behaves the same everywhere, i.e. the value of the output depends on the pattern in the image neighborhood, not the position of the neighborhood.
- **Linear:**
  - Superposition: \( h \ast (f_1 + f_2) = (h \ast f_1) + (h \ast f_2) \)
  - Scaling: \( h \ast (k \, f) = k \, (h \ast f) \)

Properties of convolution

- Linear & shift invariant
- Commutative:
  \( f \ast g = g \ast f \)
- Associative
  \( (f \ast g) \ast h = f \ast (g \ast h) \)
- Identity:
  \( \text{unit impulse } e = […], 0, 0, 1, 0, 0, […] \).
  \( f \ast e = f \)
- Differentiation:
  \( \frac{\partial}{\partial x} (f \ast g) = \frac{\partial f}{\partial x} \ast g \)

Separability

- In some cases, filter is separable, and we can factor into two steps:
  - Convolve all rows
  - Convolve all columns

Separability

- In some cases, filter is separable, and we can factor into two steps: e.g.,

\[
\begin{array}{ccc}
g & 1 & 2 & 1 \\
3 & 5 & 7 & 3 \\
8 & 4 & 6 & 8 \\
\end{array}
\]

What is the computational complexity advantage for a separable filter of size \( k \times k \), in terms of number of operations per output pixel?

\[
f \ast (g \ast h) = (f \ast g) \ast h
\]
Effect of smoothing filters

- **5x5**
  - Additive Gaussian noise
  - Salt and pepper noise

### Median filter

- No new pixel values introduced
- Removes spikes: good for impulse, salt & pepper noise
- Linear?

### Median filter

- Median filter is edge preserving

**Plots of a row of the image**

*Source: M. Hebert*

### Filters for features

- Previously, thinking of filtering as a way to remove or reduce noise
- Now, consider how filters will allow us to abstract higher-level “features”.
  - Map raw pixels to an intermediate representation that will be used for subsequent processing
  - Goal: reduce amount of data, discard redundancy, preserve what’s useful

### Template matching

- Filters as templates:
  - Note that filters look like the effects they are intended to find — “matched filters”
  - Use normalized cross-correlation score to find a given pattern (template) in the image.
  - Normalization needed to control for relative brightnesses.
Template matching

A toy example

Template matching

Where's Waldo?

Detected template
Correlation map

Where's Waldo?

Detected template
Correlation map
Template matching

What if the template is not identical to some subimage in the scene?

Template matching

Match can be meaningful, if scale, orientation, and general appearance is right.

Edge detection

- **Goal**: map image from 2d array of pixels to a set of curves or line segments or contours.
- **Why?**

  - **Main idea**: look for strong gradients, post-process

What can cause an edge?

- Reflectance change: appearance, information, texture
- Depth discontinuity: object boundary
- Change in surface orientation: shape
- Cast shadows

Contrast and invariance

Recall: Images as functions

Edges look like steep cliffs

Source: S. Seitz
Derivatives and edges

An edge is a place of **rapid change** in the image intensity function.

![Image intensity function](image)

First derivative

Edges correspond to extrema of derivative

Source: L. Lazebnik

Differentiation and convolution

For 2D function, \( f(x,y) \), the partial derivative is:

\[
\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon}
\]

For discrete data, we can approximate using finite differences:

\[
\frac{\partial f}{\partial x} \approx \frac{f(x+1, y) - f(x, y)}{1}
\]

To implement above as convolution, what would be the associated filter?

Partial derivatives of an image

\[
\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}
\]

Which shows changes with respect to \( x \)?

![Partial derivatives](partial_derivatives)

Assorted finite difference filters

Prewitt: \( M_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & 1 \end{bmatrix} \)

Sobel: \( M_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & 1 \end{bmatrix} \)

Roberts: \( M_x = \begin{bmatrix} -1 & 0 \end{bmatrix} \quad M_y = \begin{bmatrix} 0 & 1 \end{bmatrix} \)

Image gradient

The gradient of an image:

\[
\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}
\]

The gradient points in the direction of most rapid change in intensity

\[
\nabla f = \begin{bmatrix} 0 & 1 \end{bmatrix}
\]

The gradient direction (orientation of edge normal) is given by:

\[
\theta = \tan^{-1} \left( \frac{\partial f}{\partial x} / \frac{\partial f}{\partial y} \right)
\]

The edge strength is given by the gradient magnitude

\[
\| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}
\]

Effects of noise

Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal

![Signal](signal)

Where is the edge?
Solution: smooth first

Derivative theorem of convolution

$$\frac{\partial}{\partial x} (h * f) = (\frac{\partial h}{\partial x} * f)$$

Differentiation property of convolution.

Derivative of Gaussian filter

$$(I \otimes g) \otimes h = I \otimes (g \otimes h)$$

Why is this preferable?

Derivative of Gaussian filters

2D edge detection filters

$\nabla^2$ is the Laplacian operator:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$
Mask properties

- **Smoothing**
  - Values positive
  - Sum to 1 → constant regions same as input
  - Amount of smoothing proportional to mask size
  - Remove "high-frequency" components; "low-pass" filter

- **Derivatives**
  - Opposite signs used to get high response in regions of high contrast
  - Sum to 0 → no response in constant regions
  - High absolute value at points of high contrast

- **Filters act as templates**
  - Highest response for regions that "look the most like the filter"
  - Dot product as correlation

Gradients -> edges

Primary edge detection steps:
1. Smoothing: suppress noise
2. Edge enhancement: filter for contrast
3. Edge localization
   Determine which local maxima from filter output are actually edges vs. noise
   • Threshold, Thin

Smoothing with a Gaussian

Recall: parameter $\sigma$ is the "scale" / "width" / "spread" of the Gaussian kernel, and controls the amount of smoothing.

![Smoothing with a Gaussian](image)

Effect of $\sigma$ on derivatives

The apparent structures differ depending on Gaussian's scale parameter.

- Larger values: larger scale edges detected
- Smaller values: finer features detected

So, what scale to choose?

It depends what we're looking for.

- Too fine of a scale...can't see the forest for the trees.
- Too coarse of a scale...can't tell the maple grain from the cherry.

Thresholding

- Choose a threshold value $t$
- Set any pixels less than $t$ to zero (off)
- Set any pixels greater than or equal to $t$ to one (on)
Canny edge detector

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- **Non-maximum suppression:**
  - Thin multi-pixel wide “ridges” down to single pixel width
- Linking and thresholding (**hysteresis**):
  - Define two thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB: `edge(image, 'canny');`
- `>>help edge`

Source: D. Lowe, L. Fei-Fei

The Canny edge detector

original image (Lena)
The Canny edge detector

norm of the gradient

thresholding

The Canny edge detector

How to turn these thick regions of the gradient into curves?

Non-maximum suppression

Check if pixel is local maximum along gradient direction, select single max across width of the edge
• requires checking interpolated pixels p and r

The Canny edge detector

Problem: pixels along this edge didn’t survive the thresholding

Thinning (non-maximum suppression)

Hysteresis thresholding

• Check that maximum value of gradient value is sufficiently large
  – drop-outs? use hysteresis
  • use a high threshold to start edge curves and a low threshold to continue them.

Source: S. Seitz
Hysteresis thresholding

original image

high threshold (strong edges)  low threshold (weak edges)  hysteresis threshold

Source: L. Fei-Fei

Object boundaries vs. edges

Background  Texture  Shadows

Edge detection is just the beginning…

Image  human segmentation  gradient magnitude

Berkeley segmentation database:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Source: L. Lazebnik

What features are responsible for perceived edges?

Feature profiles (oriented energy, brightness, color, and texture gradients) along the patch’s horizontal diameter

[D. Martin et al. PAMI 2004]
Summary

- Filters allow local image neighborhood to influence our description and features
  - Smoothing to reduce noise
  - Derivatives to locate contrast, gradient
- Filters have highest response on neighborhoods that "look like" it; can be thought of as template matching.
- Convolution properties will influence the efficiency with which we can process images.
  - Associative
  - Filter separability
- Edge detection processes the image gradient to find curves, or chains of edgels.

Next

- Tues 9/16 binary images
- Reminder: Pset 1 due Sept 18.

Seam Carving

- Energy function: \( e_1(I) = |\frac{\partial}{\partial x} I| + |\frac{\partial}{\partial y} I| \)
- Want to remove or insert seams where they won’t be very noticeable
- Choose seam based on minimum total energy path across image.