Filtering and Edge Detection

Computer Vision I
CSE252A
Lecture 12

Announcement
• HW2 due Today
• HW3 assigned.

Image formation: Color Channel $k$

$$I_k = (D_k f_d + S_k f_s(\theta)) \hat{n} \cdot \hat{l}.$$ 

First row of $R$ is specular color $S$. Other rows are orthogonal to $S$.

Convolution: $R = K \ast I$

Kernel (K)

Note: Typically Kernel is relatively small in vision applications.

Data-dependent SUV Color Space

First row of $R$ is specular color $S$. Other rows are orthogonal to $S$.

Convolution: $R = K \ast I$

Kernel size is $m+1$ by $m+1$. 

$$R(i,j) = \sum_{h=0}^{m-1} \sum_{k=0}^{m-1} K(h,k) I(i-h, j-k)$$
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.

Properties of convolution

Let $f, g, h$ be images and $*$ denote convolution.

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

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

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.

Additive noise

- $I = S + N$. Noise doesn’t depend on signal.
- We’ll consider:
  \[ I_i = s_i + n_i \text{ with } E(n_i) = 0 \]
  $s_i$ deterministic. $n_i$ a random var.
  $n_i, n_j$ independent for $i \neq j$
  $n_i, n_j$ identically distributed.

Smoothing by Averaging

Kernel: $e^{-\frac{x^2 + y^2}{2\sigma^2}}$

An Isotropic Gaussian

- The picture shows a smoothing kernel proportional to
  \[ e^{-\frac{x^2 + y^2}{2\sigma^2}} \]
- (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 of input image $I$ with a 1-D row filter $R$ to produce an intermediate image $J$

$$J = R * I$$

– Then convolve each column of $J$ with a 1-D column filter $C$.$$

O = C * J
= C *(R * I)
= (C * R) * I$$
Fourier Transform

- 1-D transform (signal processing)
- 2-D transform (image processing)
- Consider 1-D
  - Time domain $\leftrightarrow$ Frequency Domain
  - Real $\leftrightarrow$ Complex
- Consider time domain signal to be expressed as weighted sum of sinusoid. A sinusoid $\cos(ut+\phi)$ is characterized by its phase $\phi$ and its frequency $u$
- The Fourier transform of the signal is a function giving the weights (and phase) as a function of frequency $u$.

Fourier basis element

$e^{i(ux+vy)}$

Transform is sum of orthogonal basis functions

Vector $(u,v)$
- Magnitude gives frequency
- Direction gives orientation.

Using Fourier Representations

Dominant Orientation

Limitations: not useful for local segmentation

Phase and Magnitude

$e^{i\theta} = \cos \theta + i \sin \theta$

- Fourier transform of a real function is complex
  - difficult to plot, visualize
  - instead, we can think of the phase and magnitude of the transform
- Phase is the phase of the complex transform
- Magnitude is the magnitude of the complex transform
- Curious fact
  - all natural images have about the same magnitude transform
  - hence, phase seems to matter, but magnitude largely doesn’t
- Demonstration
  - Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?
This is the magnitude transform of the cheetah pic

This is the phase transform of the cheetah pic

This is the magnitude transform of the zebra pic

This is the phase transform of the zebra pic
The Fourier Transform and Convolution

- If $H$ and $G$ are images, and $F(.)$ represents Fourier transform, then
  
  $$F(H*G) = F(H)F(G)$$

Or

$$H*G = F^{-1}(F(H)F(G))$$

- This is referred to as the Convolution Theorem.
- Fast Fourier Transform: complexity $O(n \log n)$ -> complexity of convolution is $O(n^2)$.
- Thus, one way of thinking about the properties of a convolution is by thinking of how it modifies the frequencies of the image to which it is applied.
- In particular, if we look at the power spectrum, then we see that convolving image $H$ by $G$ attenuates frequencies where $G$ has low power, and amplifies those which have high power.

Various Fourier Transform Pairs

- Important facts
  - scale function down $\iff$ scale transform up
    i.e. high frequency = small details
  - The FT of a Gaussian is a Gaussian.

Some other useful filtering techniques

- Median filter
- Anisotropic diffusion

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.
Median filters: principle

Method:
1. rank-order neighbourhood intensities
2. take middle value
• non-linear filter
• no new grey levels emerge...

Median filters: Example for window size of 3

1,1,1,7,1,1,1,1
↓
? ,1,1,1,1,1,?

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

Median filters: example

filters have width 5:

```
      +---+---+---+---+---+---+
      +---+---+---+---+---+
      +---+---+---+---+---+
      +---+---+---+---+---+
      +---+---+---+---+
```

INPUT  MEDIAN  MEAN

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 filter: images

3 x 3 median filter:

sharpens edges, destroys edge cusps
and protrusions

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)

On segmentation

Edges

Corners

Physical causes of edges

1. Object boundaries
2. Surface normal discontinuities
3. Reflectance (albedo) discontinuities
4. Lighting discontinuities

Object Boundaries
Surface normal discontinuities

Boundaries of materials properties

Boundaries of lighting

Profiles of image intensity edges

Noisy Step Edge

• Derivative is high everywhere.
• Must smooth before taking gradient.

Edge is Where Change Occurs: 1-D

• Change is measured by derivative in 1D
  • Biggest change, derivative has maximum magnitude
  • Or 2nd derivative is zero.
Numerical Derivatives

Take Taylor series expansion of \( f(x) \) about \( x_0 \)
\[
f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2} f''(x_0)(x-x_0)^2 + \ldots
\]
Consider samples taken at increments of \( h \) and first two terms of the expansion, we have
\[
f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{1}{2} f''(x_0)h^2
\]
\[
f(x_0 - h) = f(x_0) - f'(x_0)h + \frac{1}{2} f''(x_0)h^2
\]
Subtracting and adding \( f(x_0 + h) \) and \( f(x_0 - h) \) respectively yields
\[
f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h}
\]
\[
f''(x_0) = \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2}
\]
Convolve with
First Derivative: \([-1 0 1]\)
Second Derivative: \([1 -2 1]\)