Computer Vision - Lecture 3
Linear Filters
27.10.2008

Bastian Leibe
RWTH Aachen
http://www.umic.rwth-aachen.de/multimedia
leibe@umic.rwth-aachen.de

Motivation
- Noise reduction/image restoration
- Structure extraction

Topics of This Lecture
- Linear filters
  - What are they? How are they applied?
  - Application: smoothing
  - Gaussian filter
  - What does it mean to filter an image?
- Nonlinear filters
  - Median filter
- Multi-Scale representations
  - How to properly rescale an image?
- Image derivatives
  - How to compute gradients robustly?

Common Types of Noise
- Salt & pepper noise
  - Random occurrences of black and white pixels
- Impulse noise
  - Random occurrences of white pixels
- Gaussian noise
  - Variations in intensity drawn from a Gaussian (“Normal”) distribution.
- Basic Assumption
  - Noise is i.i.d. (independent & identically distributed)

Gaussian Noise

Course Outline
- Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction
  - Color
- Segmentation
- Local Features & Matching
- Object Recognition and Categorization
- 3D Reconstruction
- Motion and Tracking

Nonlinear Filters
black and white pixels
First Attempt at a Solution

- **Assumptions:**
  - Expect pixels to be like their neighbors
  - Expect noise processes to be independent from pixel to pixel ("i.i.d. = independent, identically distributed")

- Let’s try to replace each pixel with an average of all the values in its neighborhood...

---

**Moving Average in 2D**

\[ F[x, y] \quad G[x, y] \]
Moving Average in 2D

\[
F[x, y] \quad G[x, y]
\]

Correlation Filtering

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

- This is called cross-correlation, denoted \( G = H \odot F \).
- Filtering an image:
  - Replace each pixel by a weighted combination of its neighbors.
  - The filter “kernel” or “mask” is the prescription for the weights in the linear combination.

Correlation vs. Convolution

- Correlation
  \[
  G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i + u, j + v]
  \]
  Matlab: `imfilter`

- Convolution
  \[
  G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i - u, j - v]
  \]
  Matlab: `conv2`

Note the difference!

Convolution

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

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

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: \( h \ast (k \cdot f) = k (h \ast f) \)
Properties of Convolution

- Linear & shift invariant
- Commutative: \( f \ast g = g \ast f \)
- Associative: \( (f \ast g) \ast h = f \ast (g \ast h) \)
  - Often apply several filters in sequence: \( ((f \ast b_1) \ast b_2) \ast b_3) \)
  - This is equivalent to applying one filter: \( a \ast (b_1 \ast b_2 \ast b_3) \)
- Identity: \( f \ast e = f \)
  - for unit impulse \( e = [\ldots, 0, 0, 1, 0, 0, \ldots] \).
- Differentiation: \( \frac{\partial}{\partial x} (f \ast g) = \frac{\partial f}{\partial x} \ast g \)

Averaging Filter

- What values belong in the kernel \( H[u, v] \) for the moving average example?

\[
\begin{align*}
F[x, y] \ast H[u, v] &= G[x, y] \\
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\end{align*}
\]

\( G = H \ast F \)

Smoothing by Averaging

- Depicts box filter, white = high value, black = low value

“Ringing” artifacts!

Smoothing with a Gaussian

- Original
- Filtered

Smoothing with a Gaussian - Comparison

- Original
- Filtered

Gaussian Smoothing

- Gaussian kernel
  
  \[
  G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}
  \]
- Rotationally symmetric
- Weights nearby pixels more than distant ones
  - This makes sense as ‘probabilistic’ inference about the signal
- A Gaussian gives a good model of a fuzzy blob
Gaussian Smoothing
- What parameters matter here?
- Variance $\sigma$ of Gaussian
  - Determines extent of smoothing

Gaussian Smoothing
- What parameters matter here?
- Size of kernel or mask
  - Gaussian function has infinite support, but discrete filters use finite kernels
  - Rule of thumb: set filter half-width to about $3\sigma$!

Gaussian Smoothing in Matlab

```
>> hsize = 10;
>> sigma = 5;
>> h = fspecial('gaussian', hsize, sigma);
>> mesh(h);
>> imagesc(h);
>> outim = imfilter(im, h);
>> imshow(outim);
```

Effect of Smoothing
- More noise
- Wider smoothing kernel

Efficient Implementation
- Both, the BOX filter and the Gaussian filter are separable:
  - First convolve each row with a 1D filter
    $g(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-x^2/(2\sigma^2))$
  - Then convolve each column with a 1D filter
    $g(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-y^2/(2\sigma^2))$
- Remember:
  - Convolution is linear - associative and commutative
    $g_x * g_y * I = g_y * (g_x * I) = (g_x * g_y) * I$

Filtering: Boundary Issues
- What is the size of the output?
- MATLAB: `filter2(g, f, shape)`
  - shape = ‘full’: output size is sum of sizes of f and g
  - shape = ‘same’: output size is same as f
  - shape = ‘valid’: output size is difference of sizes of f and g
Filtering: Boundary Issues

- How should the filter behave near the image boundary?
  - The filter window falls off the edge of the image
  - Need to extrapolate
- Methods:
  - Clip filter (black)
  - Wrap around
  - Copy edge
  - Reflect across edge

Methods (MATLAB):
- Clip filter (black): `imfilter(f,g,0)`
- Wrap around: `imfilter(f,g,'circular')`
- Copy edge: `imfilter(f,g,'replicate')`
- Reflect across edge: `imfilter(f,g,'symmetric')`

Source: S. Marschner

Topics of This Lecture

- Linear filters
  - What are they? How are they applied?
  - Application: smoothing
  - Gaussian filter
  - What does it mean to filter an image?
- Nonlinear Filters
  - Median filter
- Multi-Scale representations
  - How to properly rescale an image?
- Image derivatives
  - How to compute gradients robustly?

The Fourier Transform in Pictures

- A small excursion into the Fourier transform to talk about spatial frequencies...

\[
\begin{align*}
3 \cos(x) &= A \\
+ 1 \cos(3x) &= B \\
+ 0.8 \cos(5x) &= C \\
+ 0.4 \cos(7x) &= D \\
+ ... &= E
\end{align*}
\]

Frequency spectrum

Source: S. Chenney

Why Does This Work?

- A small excursion into the Fourier transform to talk about spatial frequencies...

\[
\begin{align*}
3 \cos(x) &= A \\
+ 1 \cos(3x) &= B \\
+ 0.8 \cos(5x) &= C \\
+ 0.4 \cos(7x) &= D \\
+ ... &= E
\end{align*}
\]

Source: Michal Irani

Fourier Transforms of Important Functions

- Sine and cosine transform to...

\[
\begin{align*}
\sin(x) &= F \\
\cos(x) &= G
\end{align*}
\]

Source: Michal Irani
Fourier Transforms of Important Functions

- Sine and cosine transform to “frequency spikes”

- A Gaussian transforms to...

- A box filter transforms to...

All of this is symmetric!

Duality

- The better a function is localized in one domain, the worse it is localized in the other.

- This is true for any function

Effect of Convolution

- Convolving two functions in the image domain corresponds to taking the product of their transformed versions in the frequency domain.

\[ f \ast g \rightarrow F \cdot G \]

- This gives us a tool to manipulate image spectra.
  - A filter attenuates or enhances certain frequencies through this effect.

Effect of Filtering

- Noise introduces high frequencies. To remove them, we want to apply a “low-pass” filter.
- The ideal filter shape in the frequency domain would be a box. But this transfers to a spatial sinc, which has infinite spatial support.
- A compact spatial box filter transfers to a frequency sinc, which creates artifacts.
- A Gaussian has compact support in both domains. This makes it a convenient choice for a low-pass filter.
**Low-Pass vs. High-Pass**

Original image

Low-pass filtered

High-pass filtered

Image Source: S. Chenney

**Quiz: What Effect Does This Filter Have?**

Source: D. Lowe

**Sharpening Filter**

Sharpening filter

−− Accentuates differences with local average

Original

before

after

Source: D. Lowe

**Application: High Frequency Emphasis**

High Frequency Emphasis

High Frequency Emphasis

State credit: Michael Irani

**Topics of This Lecture**

- **Linear Filters**
  - What are they? How are they applied?
  - Applications: smoothing
  - Gaussian filter
  - What does it mean to filter an image?

- **Nonlinear Filters**
  - Median filter

- **Multi-Scale representations**
  - How to properly rescale an image?

- **Image derivatives**
  - How to compute gradients robustly?
Non-Linear Filters: Median Filter

- Basic idea
  - Replace each pixel by the median of its neighbors.
  
  \[
  \begin{array}{c|c|c|c|c|c}
  10 & 15 & 20 & 23 & 90 & 27 \\
  33 & 31 & 30 & 31 & 33 & 90 \\
  \end{array}
  \]
  
  Sort
  
  Replace

- Properties
  - Doesn’t introduce new pixel values
  - Removes spikes: good for impulse, salt & pepper noise
  - Linear?

Median Filter

- The Median filter is edge preserving.

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

INPUT

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEDIAN

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEAN

Median vs. Gaussian Filtering

- The Median filter is edge preserving.

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

INPUT

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEDIAN

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEAN

Motivation: Fast Search Across Scales

- The Median filter is edge preserving.

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

INPUT

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEDIAN

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEAN

Motivation: Fast Search Across Scales

- The Median filter is edge preserving.

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

INPUT

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEDIAN

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEAN

Motivation: Fast Search Across Scales

- The Median filter is edge preserving.

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

INPUT

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEDIAN

\[
\begin{array}{c|c|c|c|c|c}
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

MEAN

Motivation: Fast Search Across Scales
**Image Pyramid**

Low resolution

High resolution

**How Should We Go About Resampling?**

Let's resample the checkerboard by taking one sample at each circle.

In the top left board, the new representation is reasonable. Top right also yields a reasonable representation.

Bottom left is all black (dubious) and bottom right has checks that are too big.

**Fourier Interpretation: Discrete Sampling**

- Sampling in the spatial domain is like multiplying with a spike function.

\[ \text{Spatial Domain} \times \text{Spike Function} \]

- Sampling in the frequency domain is like convolving with a spike function.

\[ \text{Frequency Domain} \ast \text{Spike Function} \]

**Sampling and Aliasing**

Nyquist theorem:

- In order to recover a certain frequency \( f \), we need to sample with at least \( 2f \).
- This corresponds to the point at which the transformed frequency spectra start to overlap.
**Sampling and Aliasing**

- Nyquist limit

**Aliasing in Graphics**

**Resampling with Prior Smoothing**

- Note: We cannot recover the high frequencies, but we can avoid artifacts by smoothing before resampling.

**The Gaussian Pyramid**

**Summary: Gaussian Pyramid**

- Construction: create each level from previous one
  - Smooth and sample
- Smooth with Gaussians, in part because
  - a Gaussian*Gaussian = another Gaussian
  - $G(\sigma_1) * G(\sigma_2) = G(\sqrt{\sigma_1^2 + \sigma_2^2})$
- Gaussians are low-pass filters, so the representation is redundant once smoothing has been performed.
  - There is no need to store smoothed images at the full original resolution.
The Laplacian Pyramid

Gaussian Pyramid

\[ L_0 = G_0 \]

\[ L_1 = G_1 = L_0 + \text{expand}(G_0) \]

\[ \ldots \]

\[ L_n = G_n = \text{expand}(L_{n-1}) \]

Laplacian Pyramid

\[ L_n - L_{n-1} = G_{n-1} \]

Why is this useful?

Topics of This Lecture

- Linear filters
  - What are they? How are they applied?
  - Applications: smoothing
  - Gaussian filter
  - What does it mean to filter an image?

- Nonlinear Filters
  - Median filter

- Multi-Scale representations
  - How to properly rescale an image?

- Image derivatives
  - How to compute gradients robustly?

Derivatives and Edges

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

Edges and Derivatives...

1st derivative

- “zero crossings” of second derivative

2nd derivative

\[ \text{Minima of first derivative} \]

Differentiation and Convolution

- For the 2D function \( f(x, y) \), the partial derivative is:
  \[
  \frac{\partial f(x, y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon}
  \]

- For discrete data, we can approximate this using finite differences:
  \[
  \frac{\partial f(x, y)}{\partial x} \approx \frac{f(x + 1, y) - f(x, y)}{1}
  \]

- To implement the above as convolution, what would be the associated filter?
Partial Derivatives of an Image

\[ \frac{\partial f(x, y)}{\partial x} \]

\[ \frac{\partial f(x, y)}{\partial y} \]

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

-1 1

-1 1 or 1 -1?

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 intensity change.
  \[ \nabla f = [\frac{\partial f}{\partial x}, 0] \]

- The gradient direction (orientation of edge normal) is given by:
  \[ \theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right) \]

- The edge strength is given by the gradient magnitude:
  \[ \| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \]

Assorted Finite Difference Filters

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

- Sobel: \( M_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad M_y = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \)

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

\[ \text{My} = \text{fspecial('sobel')}; \]
\[ \text{outim} = \text{imfilter(double(im), My)}; \]
\[ \text{imagesc(outim)}; \]
\[ \text{colormap gray}; \]

Effect of Noise

- Consider a single row or column of the image.
  - Plotting intensity as a function of position gives a signal.

\[ f(x) \]

Where is the edge?

Solution: Smooth First

- Where is the edge?
- Look for peaks in \( \frac{\partial}{\partial x}(h \ast f) \)

Derivative Theorem of Convolution

- Differentiation property of convolution.

\[ \frac{\partial}{\partial x}(h \ast f) = (\frac{\partial h}{\partial x}) \ast f \]
Derivative of Gaussian Filter

\[(I \ast g) \ast h = I \ast (g \ast h)\]

\[
\begin{bmatrix}
0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\
0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\
0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\
0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\
0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030
\end{bmatrix} \ast \begin{bmatrix} 1 \\ -1 \end{bmatrix}
\]

Why is this preferable?

Laplacian of Gaussian (LoG)

- Consider \( \frac{\partial^2}{\partial x^2} (h \ast f) \)

\[
\text{Source: Svetlana Lazebnik}
\]

Where is the edge? Zero-crossings of bottom graph

Summary: 2D Edge Detection Filters

- \( \nabla^2 \) is the Laplacian operator:

\[
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]

Note: 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.
Correlation as Template Matching

- Think of filters as a dot product of the filter vector with the image region.
  - Now measure the angle between the vectors
    \[ a \cdot b = \frac{||a|| \cdot ||b|| \cdot \cos \theta}{||a|| \cdot ||b||} \]
- Angle (similarity) between vectors can be measured by normalizing length of each vector to 1.

Summary: 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

Summary Linear Filters

- Linear filtering:
  - Form a new image whose pixels are a weighted sum of original pixel values
- Properties
  - Output is a shift-invariant function of the input (same at each image location)
- Examples:
  - Smoothing with a box filter
  - Smoothing with a Gaussian
  - Finding a derivative
  - Searching for a template
- Pyramid representations
  - Important for describing and searching an image at all scales

References and Further Reading

- Background information on linear filters and their connection with the Fourier transform can be found in Chapters 7 and 8 of