Computer Vision - Lecture 3

Linear Filters

26.10.2009

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

Motivation

• Noise reduction/image restoration

• Structure extraction

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. (independently and identically distributed)

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

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?

• Filters as templates
  - Correlation as template matching

Gaussian Noise

\[
\text{Gaussian Noise:}\quad \mu = 0, \quad \sigma = \text{sigma}
\]

\[
\text{Output:}\quad y = x + \text{noise} \quad \text{or}\quad y = \text{im} + \text{noise}
\]

Noise credit: Kristin Lenz

Image Source: Matt Horgan
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…
**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 \( G = H \circ F \).

- 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.

**Convolution**

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

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

\[ G = H * F \]

**Correlation vs. Convolution**

- Correlation

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

- Convolution

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

**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: \( h \ast (kf) = k(h \ast f) \)

**Non-uniform weights**

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

\[ G = H \circ F \]

**Moving Average in 2D**

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

\[ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \]

\[ 0 \quad 0 \quad 0 \quad 90 \quad 90 \quad 90 \quad 90 \quad 90 \quad 0 \quad 0 \]

\[ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \]

**Shift Credit:** Kristen Grauman
Properties of Convolution

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

Averaging Filter

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

\[
F[x, y] \otimes H[u, v] = G[x, y]
\]

\[
\begin{array}{cccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

“box filter”

\[
G = H \otimes F
\]

Smoothing by Averaging

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

“Ringing” artifacts!

Smoothing with a Gaussian

- 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

Original

Filtered

Original

Filtered
Gaussian Smoothing

- What parameters matter here?
- **Variance** $\sigma$ of Gaussian
  - Determines extent of smoothing

$\sigma = 2$ with 30x30 kernel
$\sigma = 5$ with 30x30 kernel

B. Leibe
Slide credit: Kristen Grauman

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$\sigma$!

$\sigma = 5$ with 10x10 kernel
$\sigma = 5$ with 30x30 kernel

B. Leibe
Slide credit: Kristen Grauman

Gaussian Smoothing in Matlab

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

B. Leibe
Slide credit: Kristen Grauman

Effect of Smoothing

More noise $ightarrow$

B. Leibe
Slide credit: Svetlana Lazebnik

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
    $$g_x \ast g_y \ast I = g_x \ast (g_y \ast I) = (g_x \ast g_y) \ast I$$

B. Leibe
Slide credit: Bernt Schiele

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?
- Filters as templates
  - Correlation as template matching

Why Does This Work?

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

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

\( A + B + C + D = \) ...

Source: Michal Irani

The Fourier Transform in Pictures

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

Frequency spectrum

Sine and cosine transform to...

Source: Michal Irani
Fourier Transforms of Important Functions

- Sine and cosine transform to “frequency spikes”

- A Gaussian transforms to...

- A box filter transforms to...

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.

\[ f \ast g \rightarrow \mathcal{F} \cdot \mathcal{G} \]

- 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**

Original

Sharpening filter

Accentuates differences with local average

Source: D. Lowe

**Application: High Frequency Emphasis**

Original

High pass Filter

High Frequency Emphasis

High Frequency Emphasis

Histogram Equalization

Slide credit: Michal 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.

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

Median Filter

- The Median filter is edge preserving.

Median vs. Gaussian Filtering

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?

- **Non-linear filters**
  - Median filter

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

- **Filters as templates**
  - Correlation as template matching
**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.
- Sampling in the frequency domain is like convolving with a 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**

- **Signal**
- **Sample**
- **Nyquist limit**
- **Cropped and MRI**
- **Resampling**

**Aliasing in Graphics**

- Disintegrating textures

**Resampling with Prior Smoothing**

- **Gaussian**
  - $\sigma = 1$
  - $\sigma = 2$

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

**The Gaussian Pyramid**

- $G_0 = G_0 \ast$ blur
- $G_1 = (G_0 \ast$ blur$) \ast$ down-sample
- $G_2 = (G_1 \ast$ blur$) \ast$ down-sample

**Summary: Gaussian Pyramid**

- **Construction**: create each level from previous one
  - Smooth and sample
- **Smooth with Gaussians**, in part because:
  - $G(\sigma) \ast G(\sigma) = G(\sqrt{\sigma^2 + \sigma^2})$
- **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

Laplacian ~ Difference of Gaussian

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?
- Filters as templates
  - Correlation as template matching

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.

Where’s Waldo?

Slide credit: Kristen Grauman
Where’s Waldo?

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
    \[ a \cdot b = \|a\| \|b\| \cos \theta \]
    \[ \cos \theta = \frac{a \cdot b}{\|a\| \|b\|} \]
  - 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
- 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
  - D. Forsyth, J. Ponce,
    *Computer Vision: A Modern Approach.*
    Prentice Hall, 2003