Organizational Remarks

- Presenting today
  - István Sárándi (sarandi@vision.rwth-aachen.de)
- No lecture tomorrow
  - Next lecture: Tue, 23.04.

Course Outline

- Image Processing Basics
  - Image Formation
  - Linear Filters
  - Edge & Structure Extraction
  - Color
- Segmentation
- Local Features & Matching
- Object Recognition and Categorization
- Deep Learning
- 3D Reconstruction

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?
- Filters as templates
  - Correlation as template matching

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

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
Moving Average in 2D

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

Correlation Filtering

- Say the averaging window size is \( 2k+1 \times 2k+1 \):
  \[
  G[i, j] = \frac{1}{(2k+1)^2} \sum_{u=-k}^{k} \sum_{v=-k}^{k} F[i + u, j + v]
  \]
  Attribute uniform weight to each pixel
  Loop over all pixels in neighborhood around image pixel \( F[i,j] \)
- Now generalize to allow different weights depending on neighboring pixel's relative position:
  \[
  G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i + u, j + v]
  \]
  Non-uniform weights

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

Correlation vs. Convolution

- Correlation
  \[
  G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i + u, j + v]
  \]
  \[ G = H \otimes F \]
  Note the difference!
- Convolution
  \[
  G[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u, v] F[i - u, j - v]
  \]
  \[ G = H \ast F \]
- Note
  - If \( H[-u,-v] = H[u,v] \), then correlation = convolution.
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: $h \star (f_1 + f_2) = (h \star f_1) + (h \star f_2)$
  - Scaling: $h \star (k f) = k (h \star f)$

Properties of Convolution

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

Averaging Filter

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

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

\[
G = H \otimes F
\]

Smoothing by Averaging

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

Smoothing with a Gaussian

- Comparison
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 \( \sigma^2 \) 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

\( \sigma = 0.05 \) vs. \( \sigma = 0.2 \)

- Notice the difference in smoothness
- Increasing \( \sigma \) reduces noise

Effect of Smoothing

More noise

\( \sigma = 0.05 \) vs. \( \sigma = 0.2 \)

- Notice the difference in smoothness
- Increasing \( \sigma \) reduces noise
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
  -\[ g_x \ast g_y \ast I = g_x \ast (g_y \ast I) = (g_x \ast g_y) \ast I \]

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

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...
  - $3 \cos(x) + 1 \cos(3x) + 0.8 \cos(5x) + 0.4 \cos(7x) + \ldots$

The Fourier Transform in Cartoons

- A small excursion into the Fourier transform to talk about spatial frequencies...
  - $A + B + C + D$
  - Frequency coefficients
Fourier Transforms of Important Functions

- Sine and cosine transform to “frequency spikes”
- A Gaussian transforms to a Gaussian
- A box filter transforms to a sinc

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

- Low-pass filtered
- High-pass filtered

---

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

- Original
- 2.0
- 0.33
- ?

---

**Sharpen Filter**

- Original
- 2.0
- 0.33
- Sharpening filter
- Accentuates differences with local average

---

**Application: High Frequency Emphasis**

- Original
- High pass Filter
- High Frequency Emphasis
- Histogram Equalization
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?

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

<table>
<thead>
<tr>
<th>Filter Size</th>
<th>Gaussian</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td><img src="image" alt="Gaussian 3x3" /></td>
<td><img src="image" alt="Median 3x3" /></td>
</tr>
<tr>
<td>5x5</td>
<td><img src="image" alt="Gaussian 5x5" /></td>
<td><img src="image" alt="Median 5x5" /></td>
</tr>
<tr>
<td>7x7</td>
<td><img src="image" alt="Gaussian 7x7" /></td>
<td><img src="image" alt="Median 7x7" /></td>
</tr>
</tbody>
</table>

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
Motivation: Fast Search Across Scales

How Should We Go About Resampling?

Fourier Interpretation: Discrete Sampling

Sampling and Aliasing
**Nyquist Theorem:**

- 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 (the Nyquist limit).

---

**Aliasing in Graphics**

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

---

**The Gaussian Pyramid**

- $G_i = (G_{i-1} \ast \text{gaussian}) \downarrow 2$
- $G_0 = (G_{0} \ast \text{gaussian}) \downarrow 2$
- $G_1 = (G_{1} \ast \text{gaussian}) \downarrow 2$
- $G_2 = (G_{2} \ast \text{gaussian}) \downarrow 2$

---

**Gaussian Pyramid – Stored Information**

- Illustrates the storage of information at different resolutions using the Gaussian pyramid.
Summary: Gaussian Pyramid

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

\[ L_0 = G_0, \quad L_i = G_i - \text{DoG}_i = G_i - \text{expand}(G_{i+1}) \]

Why is this useful?

Laplacian ~ Difference of Gaussian

\[ \text{DoG} = \text{Difference of Gaussians} \]

Cheap approximation – no derivatives needed.

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

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?
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
    \[ a \cdot b = \|a\| \|b\| \cos \theta \quad \text{and} \quad \cos \theta = \frac{a \cdot b}{\|a\| \|b\|} \]
  - 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)
  - Important for describing and searching an image at all scales
- 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 Chapter 3 of the Szeliski book or Chapters 7 and 8 of Forsyth & Ponce.