Image filtering
Recall: Image transformations

- What are different kinds of image transformations?
  - Range transformations or point processing
  - Image warping
  - Image filtering
Image filtering: Outline

- Linear filtering and its properties
- Gaussian filters and their properties
- Nonlinear filtering: Median filtering
- Fun filtering application: Hybrid images
Sliding window operations

- Let’s slide a fixed-size window over the image and perform the same simple computation at each window location.
- Example use case: how do we reduce image noise?
  - Let’s take the average of pixel values in each window.
  - More generally, we can take a weighted sum where the weights are given by a filter kernel.
Applying a linear filter

<table>
<thead>
<tr>
<th>Input</th>
<th>Filter</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_{11})</td>
<td>(f_{11})</td>
<td>(O_{11})</td>
</tr>
<tr>
<td>(I_{12})</td>
<td>(f_{12})</td>
<td>(O_{12})</td>
</tr>
<tr>
<td>(I_{13})</td>
<td>(f_{13})</td>
<td>(O_{13})</td>
</tr>
<tr>
<td>(I_{14})</td>
<td>(f_{21})</td>
<td>(O_{21})</td>
</tr>
<tr>
<td>(I_{15})</td>
<td>(f_{22})</td>
<td>(O_{22})</td>
</tr>
<tr>
<td>(I_{16})</td>
<td>(f_{23})</td>
<td>(O_{23})</td>
</tr>
<tr>
<td>(I_{21})</td>
<td>(f_{31})</td>
<td>(O_{31})</td>
</tr>
<tr>
<td>(I_{22})</td>
<td>(f_{32})</td>
<td>(O_{32})</td>
</tr>
<tr>
<td>(I_{23})</td>
<td>(f_{33})</td>
<td>(O_{33})</td>
</tr>
</tbody>
</table>

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

\[
O_{11} = I_{11} \cdot f_{11} + I_{12} \cdot f_{12} + I_{13} \cdot f_{13} + \ldots + I_{33} \cdot f_{33}
\]

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

\[
\begin{align*}
I_{11} & \quad I_{12} & \quad I_{13} & \quad I_{14} & \quad I_{15} & \quad I_{16} \\
I_{21} & \quad I_{22} & \quad I_{23} & \quad I_{24} & \quad I_{25} & \quad I_{26} \\
I_{31} & \quad I_{32} & \quad I_{33} & \quad I_{34} & \quad I_{35} & \quad I_{36} \\
I_{41} & \quad I_{42} & \quad I_{43} & \quad I_{44} & \quad I_{45} & \quad I_{46} \\
I_{51} & \quad I_{52} & \quad I_{53} & \quad I_{54} & \quad I_{55} & \quad I_{56}
\end{align*}
\]

\[
\begin{align*}
\text{Filter} & \quad \begin{array}{ccc}
f_{11} & f_{12} & f_{13} \\
f_{21} & f_{22} & f_{23} \\
f_{31} & f_{32} & f_{33}
\end{array} \\
\end{align*}
\]

\[
O_{12} = I_{12} \cdot f_{11} + I_{13} \cdot f_{12} + I_{14} \cdot f_{13} + \ldots + I_{34} \cdot f_{33}
\]

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

\[
O_{13} = I_{13} \cdot f_{11} + I_{14} \cdot f_{12} + I_{15} \cdot f_{13} + \ldots + I_{35} \cdot f_{33}
\]

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

\[ O_{14} = I_{14} \cdot f_{11} + I_{15} \cdot f_{12} + I_{16} \cdot f_{13} + \ldots + I_{36} \cdot f_{33} \]

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

\[
O_{21} = I_{21} \cdot f_{11} + I_{22} \cdot f_{12} + I_{23} \cdot f_{13} + \ldots + I_{43} \cdot f_{33}
\]

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

\[
\begin{array}{cccccc}
I_{11} & I_{12} & I_{13} & I_{14} & I_{15} & I_{16} \\
I_{21} & I_{22} & I_{23} & I_{24} & I_{25} & I_{26} \\
I_{31} & I_{32} & I_{33} & I_{34} & I_{35} & I_{36} \\
I_{41} & I_{42} & I_{43} & I_{44} & I_{45} & I_{46} \\
I_{51} & I_{52} & I_{53} & I_{54} & I_{55} & I_{56} \\
\end{array}
\]

\[
\begin{array}{ccc}
f_{11} & f_{12} & f_{13} \\
f_{21} & f_{22} & f_{23} \\
f_{31} & f_{32} & f_{33} \\
\end{array}
\]

\[
\begin{array}{cccc}
o_{11} & o_{12} & o_{13} & o_{14} \\
o_{21} & o_{22} & \ & \ \\
\end{array}
\]

\[
O_{22} = I_{22} \cdot f_{11} + I_{23} \cdot f_{12} + I_{24} \cdot f_{13} + \ldots + I_{44} \cdot f_{33}
\]

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

\[ O_{23} = I_{23} \cdot f_{11} + I_{24} \cdot f_{12} + I_{25} \cdot f_{13} + \ldots + I_{45} \cdot f_{33} \]

Adapted from D. Fouhey and J. Johnson
Applying a linear filter

<table>
<thead>
<tr>
<th>Input</th>
<th>Filter</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{11}$</td>
<td>$f_{11}$</td>
<td>$O_{11}$</td>
</tr>
<tr>
<td>$I_{12}$</td>
<td>$f_{12}$</td>
<td>$O_{12}$</td>
</tr>
<tr>
<td>$I_{13}$</td>
<td>$f_{13}$</td>
<td>$O_{13}$</td>
</tr>
<tr>
<td>$I_{14}$</td>
<td>$f_{21}$</td>
<td>$O_{14}$</td>
</tr>
<tr>
<td>$I_{15}$</td>
<td>$f_{22}$</td>
<td>$O_{22}$</td>
</tr>
<tr>
<td>$I_{16}$</td>
<td>$f_{23}$</td>
<td>$O_{23}$</td>
</tr>
<tr>
<td>$I_{21}$</td>
<td>$f_{31}$</td>
<td>$O_{24}$</td>
</tr>
<tr>
<td>$I_{22}$</td>
<td>$f_{32}$</td>
<td></td>
</tr>
<tr>
<td>$I_{23}$</td>
<td>$f_{14}$</td>
<td></td>
</tr>
<tr>
<td>$I_{24}$</td>
<td>$f_{15}$</td>
<td></td>
</tr>
<tr>
<td>$I_{25}$</td>
<td>$f_{16}$</td>
<td></td>
</tr>
<tr>
<td>$I_{26}$</td>
<td>$f_{24}$</td>
<td></td>
</tr>
<tr>
<td>$I_{31}$</td>
<td>$f_{25}$</td>
<td></td>
</tr>
<tr>
<td>$I_{32}$</td>
<td>$f_{33}$</td>
<td></td>
</tr>
<tr>
<td>$I_{33}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{34}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{35}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{36}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{41}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{42}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{43}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{44}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{45}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{46}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{51}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{52}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{53}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{54}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{55}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{56}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What filter values should we use to find the average in a 3×3 window?

Adapted from D. Fouhey and J. Johnson
Practical details: Dealing with edges

- To control the size of the output, we need to use *padding*
Practical details: Dealing with edges

- To control the size of the output, we need to use *padding*
- What values should we pad the image with?
Practical details: Dealing with edges

- To control the size of the output, we need to use *padding*
- What values should we pad the image with?
  - Zero pad (or clip filter)
  - Wrap around
  - Copy edge
  - Reflect across edge

Source: S. Marschner
Properties: Linearity

\[ \text{filter}(I, f_1 + f_2) = \text{filter}(I, f_1) + \text{filter}(I, f_2) \]

Adapted from D. Fouhey and J. Johnson
Properties: Linearity

\[ \text{filter}(I, f_1 + f_2) = \text{filter}(I, f_1) + \text{filter}(I, f_2) \]

Also:

\[ \text{filter}(I_1 + I_2, f) = \text{filter}(I_1, f) + \text{filter}(I_2, f) \]
\[ \text{filter}(kI, f) = k \text{ filter}(I, f) \]
\[ \text{filter}(I, kf) = k \text{ filter}(I, f) \]

Adapted from D. Fouhey and J. Johnson
Properties: Shift-invariance

\[ \text{filter(shift}(I, f)) = \text{shift(filter}(I, f)) \]

Adapted from D. Fouhey and J. Johnson
More linear filtering properties

• Commutativity: \( f \ast g = g \ast f \)
  • For infinite signals, no difference between filter and signal

• Associativity: \( f \ast (g \ast h) = (f \ast g) \ast h \)
  • Convolving several filters one after another is equivalent to convolving with one combined filter:
    \[
    \left( (g \ast f_1) \ast f_2 \right) \ast f_3 = g \ast (f_1 \ast f_2 \ast f_3)
    \]

• Identity: for unit impulse \( e \), \( f \ast e = f \)
Note: Filtering vs. “convolution”

- In classical signal processing terminology, convolution is filtering with a flipped kernel, and filtering with an upright kernel is known as cross-correlation
- Check convention of filtering function you plan to use!

Filtering or “cross-correlation”  
(Kernel in original orientation)

“Convolution”  
(Kernel flipped in x and y)

Adapted from D. Fouhey and J. Johnson
Practice with linear filters

Original

Source: D. Lowe
Practice with linear filters

Original

One surrounded by zeros is the *identity filter*

Filtered (no change)

Source: D. Lowe
Practice with linear filters

Original

Source: D. Lowe
Practice with linear filters

Original

Shifted left
By one pixel

Source: D. Lowe
Practice with linear filters

Original

\[ \frac{1}{9} \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \]

? 

Source: D. Lowe
Practice with linear filters

Original

Blur (with a box filter)

Source: D. Lowe
Practice with linear filters

Original

\[
\begin{array}{ccc}
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 \\
\end{array}
\quad - \quad \frac{1}{9}
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

Source: D. Lowe
Practice with linear filters

Original

Sharpening filter:
Accentuates differences
with local average
(Note that filter sums to 1)

Sharpened

Source: D. Lowe
Sharpening

before

after

Source: D. Lowe
Sharpening

Original - Smoothed = Detail

Original + Detail = Sharpened

Source: S. Gupta
Image filtering: Outline

• Linear filtering and its properties
• Gaussian filters and their properties
Smoothing with box filter revisited

• What’s wrong with this picture?
• What’s the solution?

Source: D. Forsyth
Smoothing with box filter revisited

• What’s wrong with this picture?
• What’s the solution?
  • To eliminate edge effects, weight contribution of neighborhood pixels according to their closeness to the center

\[
G(x, y) \propto \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right)
\]

“proportional to”
(renormalize values to sum to 1)

standard deviation
(determines size of “blob”)

Adapted from D. Fouhey and J. Johnson
Gaussian vs. box filtering
Applying Gaussian filters

Input image
(no filter)

Source: D. Fouhey and J. Johnson
Applying Gaussian filters

\[ \sigma = 1 \]

Source: D. Fouhey and J. Johnson
Applying Gaussian filters

\[ \sigma = 2 \]

Source: D. Fouhey and J. Johnson
Applying Gaussian filters

$\sigma = 4$

Source: D. Fouhey and J. Johnson
Applying Gaussian filters

\[ \sigma = 8 \]
Choosing filter size

- Rule of thumb: set filter width to about $6\sigma$ (captures 99.7% of the energy)

\[
\sigma = 8 \\
\text{Width} = 21
\]

Too small!

\[
\sigma = 8 \\
\text{Width} = 43
\]

A bit small (might be OK)

Adapted from D. Fouhey and J. Johnson
Gaussian filters: Properties

- Gaussian is a low-pass filter: it removes high-frequency components from the image (more on this soon)
- Convolution with self is another Gaussian
  - So we can smooth with small-\(\sigma\) kernel, repeat, and get same result as larger-\(\sigma\) kernel would have
  - Convolving \textit{two times} with Gaussian kernel with std. dev. \(\sigma\) is the same as convolving \textit{once} with kernel with std. dev. \(\sigma \sqrt{2}\)
- Gaussian kernel is \textit{separable}: it factors into product of two 1D Gaussians

Source: K. Grauman
Separability of the Gaussian filter

\[
\frac{1}{2\pi\sigma^2} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{x^2}{2\sigma^2} \right) \cdot \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{y^2}{2\sigma^2} \right)
\]

Adapted from D. Fouhey and J. Johnson
Why is separability useful?

- Separability means that a 2D convolution can be reduced to two 1D convolutions (one along rows and one along columns).
- What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?
  - $O(n^2 m^2)$
- What if the kernel is separable?
  - $O(n^2 m)$
Image filtering: Outline

• Linear filtering and its properties
• Gaussian filters and their properties
• Nonlinear filtering: Median filtering
Different types of noise

- Gaussian filtering is appropriate for additive, zero-mean noise (assuming nearby pixels share the same value)

Adapted from D. Fouhey and J. Johnson
Different types of noise

- What about *impulse* or *shot noise*, i.e., when some pixels are arbitrarily replaced by spurious values?

Adapted from D. Fouhey and J. Johnson
Where Gaussian filtering fails

\[ \sigma = 1 \]

Adapted from D. Fouhey and J. Johnson
Alternative idea: Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window.

```
10 15 20
23 90 27
33 31 30
```

1. Sort the values:
   - Sorted values: 10, 15, 20, 23, 27, 30, 31, 33, 90

2. Replace the center value with the median:
   - Median value: 27
   - Replaced values: 10, 15, 20, 23, **27**, 30, 31, 33, 90
```
Applying median filter

Input image
(no filter)

Adapted from D. Fouhey and J. Johnson
Applying median filter

median filter
(width = 3)

Adapted from D. Fouhey and J. Johnson
Applying median filter

median filter
(width = 7)
Is median filtering linear?
Is median filtering linear?

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}$$

median filter

$1 + 0 \neq 2$

Adapted from D. Fouhey and J. Johnson
Image filtering: Outline

• Linear filtering and its properties
• Gaussian filters and their properties
• Nonlinear filtering: Median filtering
• Fun filtering application: Hybrid images
Application: Hybrid images

Recall: Sharpening

Original - Smoothed = Detail

Original + \( \alpha \) = Sharpened

Source: S. Gupta
“Detail” filter

\[ I - I \ast g = I \ast (e - g) \]
Application: Hybrid images

Gaussian filter

Laplacian filter
Application: Hybrid images

Changing expression

Sad ——— Surprised

SIGGRAPH2006