Lecture 4: Image Filtering (continued)

( unless otherwise stated slides are taken or adopted from Bob Woodham, Jim Little and Fred Tung )
Menu for Today (January 16, 2020)

Topics:

- Gaussian and Pillbox filters
- Separability
- The Convolution Theorem
- Non-linear filters

Readings:

- Today’s Lecture: none
- Next Lecture: Forsyth & Ponce (2nd ed.) 4.4

Reminders:

- Assignment 1: Image Filtering and Hybrid Images due January 28-th
- Today my office hours will start at 3:30pm (not 3pm as posted)
Today’s “fun” Example: Rolling Shutter
Today’s “fun” Example: Rolling Shutter
Today’s “fun” Example: Rolling Shutter

Rolling shutter effect
Today’s “fun” Example: Rolling Shutter

Rolling shutter effect
I am in class today:

A) True
B) False
Lecture 3: Re-cap

— The **correlation** of $F(X, Y)$ and $I(X, Y)$ is:

$$I'(X, Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(I, J) I(X + i, Y + j)$$

— **Visual interpretation**: Superimpose the filter $F$ on the image $I$ at $(X, Y)$, perform an element-wise multiply, and sum up the values

— **Convolution** is like **correlation** except filter “flipped”

  if $F(X, Y) = F(-X, -Y)$ then correlation = convolution.
Lecture 3: Re-cap

Ways to handle **boundaries**
- **Ignore/discard.** Make the computation undefined for top/bottom k rows and left/right-most k columns
- **Pad with zeros.** Return zero whenever a value of I is required beyond the image bounds
- **Assume periodicity.** Top row wraps around to the bottom row; leftmost column wraps around to rightmost column.

Simple **examples** of filtering:
- copy, shift, smoothing, sharpening

Linear filter **properties:**
- superposition, scaling, shift invariance

**Characterization Theorem:** Any linear, shift-invariant operation can be expressed as a convolution
Example 5: Smoothing with a Box Filter

Gonzales & Woods (3rd ed.) Figure 3.3
Smoothing

Smoothing with a box doesn’t model lens defocus well

— Smoothing with a box filter depends on direction
— Image in which the center point is 1 and every other point is 0
Lecture 2: Re-cap

* image credit: https://catlikecoding.com/unity/tutorials/advanced-rendering/depth-of-field/circle-of-confusion/lens-camera.png
Smoothing with a box *doesn’t model lens defocus* well

- Smoothing with a box filter depends on direction
- Image in which the center point is 1 and every other point is 0
Smoothing

Smoothing with a box doesn’t model lens defocus well
— Smoothing with a box filter depends on direction
— Image in which the center point is 1 and every other point is 0

Filter

<table>
<thead>
<tr>
<th>1/9</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1/9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Image

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Result

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Smoothing

Smoothing with a box **doesn’t model lens defocus** well
— Smoothing with a box filter depends on direction
— Image in which the center point is 1 and every other point is 0

Smoothing with a (circular) **pillbox** is a better model for defocus (in geometric optics)

The **Gaussian** is a good general smoothing model
— for phenomena (that are the sum of other small effects)
— whenever the Central Limit Theorem applies
Example 6: Smoothing with a Gaussian

Idea: Weight contributions of pixels by spatial proximity (nearness)

2D Gaussian (continuous case):

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

— The **correlation** of $F(X, Y)$ and $I(X, Y)$ is:

$$ I'(X, Y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} F(i, j)I(X + i, Y + j) $$

— **Visual interpretation**: Superimpose the filter $F$ on the image $I$ at $(X, Y)$, perform an element-wise multiply, and sum up the values

— **Convolution** is like **correlation** except filter “flipped”

  if $F(X, Y) = F(-X, -Y)$ then correlation = convolution.

— **Characterization Theorem**: Any linear, spatially invariant operation can be expressed as a convolution
Example 6: Smoothing with a Gaussian

Idea: Weight contributions of pixels by spatial proximity (nearness)

2D **Gaussian** (continuous case):

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

Forsyth & Ponce (2nd ed.)
Figure 4.2
Example 6: Smoothing with a Gaussian

Idea: Weight contributions of pixels by spatial proximity (nearness)

2D **Gaussian** (continuous case):

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

Standard Deviation

Forsyth & Ponce (2nd ed.)
Figure 4.2
Example 6: Smoothing with a Gaussian

Quantized an truncated $3 \times 3$ Gaussian filter:

<table>
<thead>
<tr>
<th>$G_\sigma(-1, 1)$</th>
<th>$G_\sigma(0, 1)$</th>
<th>$G_\sigma(1, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\sigma(-1, 0)$</td>
<td>$G_\sigma(0, 0)$</td>
<td>$G_\sigma(1, 0)$</td>
</tr>
<tr>
<td>$G_\sigma(-1, -1)$</td>
<td>$G_\sigma(0, -1)$</td>
<td>$G_\sigma(1, -1)$</td>
</tr>
</tbody>
</table>
Example 6: Smoothing with a Gaussian

Quantized and truncated $3 \times 3$ Gaussian filter:

<table>
<thead>
<tr>
<th>$G_\sigma(-1, 1)$</th>
<th>$G_\sigma(0, 1)$</th>
<th>$G_\sigma(1, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\sigma(-1, 1) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
<td>$G_\sigma(0, 1) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(1, 1) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
</tr>
<tr>
<td>$G_\sigma(-1, 0) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(0, 0) = \frac{1}{2\pi \sigma^2}$</td>
<td>$G_\sigma(1, 0) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
</tr>
<tr>
<td>$G_\sigma(-1, -1) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
<td>$G_\sigma(0, -1) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(1, -1) = \frac{1}{2\pi \sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
</tr>
</tbody>
</table>
Example 6: Smoothing with a Gaussian

Quantized an truncated 3x3 Gaussian filter:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>( G_{\sigma}(x, y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>1</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2}{2\sigma^2}} )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>( \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} )</td>
</tr>
</tbody>
</table>

With \( \sigma = 1 \):

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.059</td>
<td>0.097</td>
<td>0.059</td>
</tr>
<tr>
<td>0.097</td>
<td>0.159</td>
<td>0.097</td>
</tr>
<tr>
<td>0.059</td>
<td>0.097</td>
<td>0.059</td>
</tr>
</tbody>
</table>
Example 6: Smoothing with a Gaussian

Quantized an truncated \textbf{3x3 Gaussian} filter:

\[
G_\sigma(-1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}} \quad G_\sigma(0, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} \quad G_\sigma(1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}
\]

\[
G_\sigma(-1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} \quad G_\sigma(0, 0) = \frac{1}{2\pi\sigma^2} \quad G_\sigma(1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}
\]

\[
G_\sigma(-1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}} \quad G_\sigma(0, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}} \quad G_\sigma(1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}
\]

With \(\sigma = 1\):

\[
\begin{array}{ccc}
0.059 & 0.097 & 0.059 \\
0.097 & 0.159 & 0.097 \\
0.059 & 0.097 & 0.059 \\
\end{array}
\]

What happens if \(\sigma\) is larger?
Example 6: Smoothing with a Gaussian

Quantized an truncated 3x3 Gaussian filter:

<table>
<thead>
<tr>
<th>$G_\sigma(-1, 1)$</th>
<th>$G_\sigma(0, 1)$</th>
<th>$G_\sigma(1, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{2}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{1}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{2}{2\sigma^2}}$</td>
</tr>
<tr>
<td>$G_\sigma(-1, 0)$</td>
<td>$G_\sigma(0, 0)$</td>
<td>$G_\sigma(1, 0)$</td>
</tr>
<tr>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{1}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{1}{2\sigma^2}}$</td>
</tr>
<tr>
<td>$G_\sigma(-1, -1)$</td>
<td>$G_\sigma(0, -1)$</td>
<td>$G_\sigma(1, -1)$</td>
</tr>
<tr>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{2}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{1}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp{-\frac{2}{2\sigma^2}}$</td>
</tr>
</tbody>
</table>

With $\sigma = 1$ :

```
  ↑  ↑  ↑
  ↑  ↓  ↑
  ↑  ↑  ↑
```

What happens if $\sigma$ is larger?

— More blur
Example 6: Smoothing with a Gaussian

Quantized an truncated 3x3 Gaussian filter:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\sigma(-1, 1)$</td>
<td>$G_\sigma(0, 1)$</td>
<td>$G_\sigma(1, 1)$</td>
</tr>
<tr>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\sigma(-1, 0)$</td>
<td>$G_\sigma(0, 0)$</td>
<td>$G_\sigma(1, 0)$</td>
</tr>
<tr>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\sigma(-1, -1)$</td>
<td>$G_\sigma(0, -1)$</td>
<td>$G_\sigma(1, -1)$</td>
</tr>
<tr>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$\frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
</tr>
</tbody>
</table>

With $\sigma = 1$:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.059</td>
<td>0.097</td>
<td>0.059</td>
</tr>
<tr>
<td>0.097</td>
<td>0.159</td>
<td>0.097</td>
</tr>
<tr>
<td>0.059</td>
<td>0.097</td>
<td>0.059</td>
</tr>
</tbody>
</table>

What happens if $\sigma$ is larger?

What happens if $\sigma$ is smaller?
Example 6: Smoothing with a Gaussian

Quantized and truncated 3x3 Gaussian filter:

<table>
<thead>
<tr>
<th>$G_\sigma(-1, 1)$</th>
<th>$G_\sigma(0, 1)$</th>
<th>$G_\sigma(1, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\sigma(-1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
<td>$G_\sigma(0, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
</tr>
<tr>
<td>$G_\sigma(-1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(0, 0) = \frac{1}{2\pi\sigma^2}$</td>
<td>$G_\sigma(1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
</tr>
<tr>
<td>$G_\sigma(-1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
<td>$G_\sigma(0, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
</tr>
</tbody>
</table>

With $\sigma = 1$ :

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>Down</td>
<td>Down</td>
</tr>
<tr>
<td>Down</td>
<td></td>
<td>Up</td>
</tr>
<tr>
<td>Down</td>
<td>Down</td>
<td>Down</td>
</tr>
</tbody>
</table>

What happens if $\sigma$ is larger?

What happens if $\sigma$ is smaller?

— **Less** blur
Example 6: Smoothing with a Gaussian

Forsyth & Ponce (2nd ed.) Figure 4.1 (left and right)
**Box vs. Gaussian Filter**

- **original**
- **7x7 Gaussian**
- **7x7 box**

*Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)*
Fun: How to get shadow effect?

University of British Columbia

Adopted from: Ioannis (Yannis) Gkioulekas (CMU)
Fun: How to get shadow effect?

University of British Columbia

Blur with a Gaussian kernel, then compose the blurred image with the original (with some offset)

Adopted from: Ioannis (Yannis) Gkioulekas (CMU)
Example 6: Smoothing with a Gaussian

Quantized an truncated 3x3 Gaussian filter:

\[
G_\sigma(-1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{\sigma^2}}
\]
\[
G_\sigma(0, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}
\]
\[
G_\sigma(1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}
\]
\[
G_\sigma(-1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}
\]
\[
G_\sigma(0, 0) = \frac{1}{2\pi\sigma^2}
\]
\[
G_\sigma(1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}
\]
\[
G_\sigma(-1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}
\]
\[
G_\sigma(0, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}
\]
\[
G_\sigma(1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}
\]

With \(\sigma = 1\):

<table>
<thead>
<tr>
<th>0.059</th>
<th>0.097</th>
<th>0.059</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.097</td>
<td>0.159</td>
<td>0.097</td>
</tr>
<tr>
<td>0.059</td>
<td>0.097</td>
<td>0.059</td>
</tr>
</tbody>
</table>

What is the problem with this filter?
Example 6: Smoothing with a Gaussian

Quantized and truncated 3x3 Gaussian filter:

<table>
<thead>
<tr>
<th>$G_\sigma(-1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</th>
<th>$G_\sigma(0, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</th>
<th>$G_\sigma(1, 1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_\sigma(-1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(0, 0) = \frac{1}{2\pi\sigma^2}$</td>
<td>$G_\sigma(1, 0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
</tr>
<tr>
<td>$G_\sigma(-1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
<td>$G_\sigma(0, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$</td>
<td>$G_\sigma(1, -1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$</td>
</tr>
</tbody>
</table>

With $\sigma = 1$ :

| 0.059 | 0.097 | 0.059 |
| 0.097 | 0.159 | 0.097 |
| 0.059 | 0.097 | 0.059 |

What is the problem with this filter?

- does not sum to 1
- truncated too much
Gaussian: Area Under the Curve
Example 6: Smoothing with a Gaussian

With $\sigma = 1$:

\[
\begin{array}{ccc}
0.059 & 0.097 & 0.059 \\
0.097 & 0.159 & 0.097 \\
0.059 & 0.097 & 0.059 \\
\end{array}
\]

Better version of the Gaussian filter:

- sums to 1 (normalized)
- captures $\pm 2\sigma$

\[
\begin{array}{cccccc}
1 & 4 & 7 & 4 & 1 \\
4 & 16 & 26 & 16 & 4 \\
7 & 26 & 41 & 26 & 7 \\
4 & 16 & 26 & 16 & 4 \\
1 & 4 & 7 & 4 & 1 \\
\end{array}
\]

\[
\frac{1}{273}
\]

In general, you want the Gaussian filter to capture $\pm 3\sigma$, for $\sigma = 1 \Rightarrow 7\times7$ filter
Lets talk about efficiency
Efficient Implementation: **Separability**

A 2D function of \( x \) and \( y \) is **separable** if it can be written as the product of two functions, one a function only of \( x \) and the other a function only of \( y \).

Both the **2D box filter** and the **2D Gaussian filter** are **separable**.

Both can be implemented as two 1D convolutions:
- First, convolve each row with a 1D filter
- Then, convolve each column with a 1D filter
- Aside: or vice versa

The **2D Gaussian** is the only (non trivial) 2D function that is both separable and rotationally invariant.
Separability: Box Filter Example

\[ F(X, Y) = F(X)F(Y) \]

filter

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

Standard (3x3)
Separability: Box Filter Example

$F(X, Y) = F(X)F(Y)$

$I(X,Y)$

$F(X)$

Separable filter

Standard filter
Separability: Box Filter Example

$$F(X, Y) = F(X)F(Y)$$

$$F(X)$$ filter

$$F(Y)$$ filter

$$I'(X, Y)$$ output
Efficient Implementation: **Separability**

For example, recall the 2D **Gaussian**:

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

The 2D Gaussian can be expressed as a product of two functions, one a function of \(x\) and another a function of \(y\).
Efficient Implementation: **Separability**

For example, recall the 2D **Gaussian**:

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

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

The 2D Gaussian can be expressed as a product of two functions, one a function of x and another a function of y.
Efficient Implementation: **Separability**

For example, recall the 2D **Gaussian**:

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

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

function of x function of y

The 2D Gaussian can be expressed as a product of two functions, one a function of x and another a function of y.

In this case the two functions are (identical) 1D Gaussians.
Efficient Implementation: *Separability*

Naive implementation of 2D *Gaussian*:

At each pixel, \((X, Y)\), there are \(m \times m\) multiplications

There are \(n \times n\) pixels in \((X, Y)\)

**Total:** \(m^2 \times n^2\) multiplications
Efficient Implementation: **Separability**

Naive implementation of 2D **Gaussian**:

At each pixel, \((X, Y)\), there are \(m \times m\) multiplications

There are \(n \times n\) pixels in \((X, Y)\)

**Total:** \(m^2 \times n^2\) multiplications

Separable 2D **Gaussian**:
Efficient Implementation: **Separability**

Naive implementation of 2D **Gaussian**:

At each pixel, \((X, Y)\), there are \(m \times m\) multiplications

There are \(n \times n\) pixels in \((X, Y)\)

**Total**: \(m^2 \times n^2\) multiplications

Separable 2D **Gaussian**:

At each pixel, \((X, Y)\), there are \(2m\) multiplications

There are \(n \times n\) pixels in \((X, Y)\)

**Total**: \(2m \times n^2\) multiplications
Example 7: Smoothing with a Pillbox

Let the radius (i.e., half diameter) of the filter be \( r \)

In a contentious domain, a 2D (circular) pillbox filter, \( f(x, y) \), is defined as:

\[
f(x, y) = \frac{1}{\pi r^2} \begin{cases} 
1 & \text{if } x^2 + y^2 \leq r^2 \\
0 & \text{otherwise}
\end{cases}
\]

The scaling constant, \( \frac{1}{\pi r^2} \), ensures that the area of the filter is one
Recall that the 2D Gaussian is the only (non trivial) 2D function that is both separable and rotationally invariant.

A 2D pillbox is rotationally invariant but not separable.

There are occasions when we want to convolve an image with a 2D pillbox. Thus, it worth exploring possibilities for efficient implementation.
Example 7: Smoothing with a Pillbox

A 2D box filter can be expressed as the sum of a 2D pillbox and some “extra corner bits”
Example 7: Smoothing with a Pillbox

Therefore, a 2D pillbox filter can be expressed as the difference of a 2D box filter and those same “extra corner bits”
Implementing convolution with a 2D pillbox filter as the difference between convolution with a box filter and convolution with the “extra corner bits” filter allows us to take advantage of the separability of a box filter.

Further, we can postpone scaling the output to a single, final step so that convolution involves filters containing all 0’s and 1’s. — This means the required convolutions can be implemented without any multiplication at all.
Example 7: Smoothing with a Pillbox

Original

11 x 11 Pillbox
Speeding Up Convolution (The Convolution Theorem)

Let $z$ be the product of two numbers, $x$ and $y$, that is,

$$z = xy$$
Speeding Up **Convolution** (The Convolution Theorem)

Let \( z \) be the product of two numbers, \( x \) and \( y \), that is,

\[
z = xy
\]

Taking logarithms of both sides, one obtains

\[
\ln z = \ln x + \ln y
\]
Speeding Up **Convolution** (The Convolution Theorem)

Let $z$ be the product of two numbers, $x$ and $y$, that is,

$$z = xy$$

Taking logarithms of both sides, one obtains

$$\ln z = \ln x + \ln y$$

Therefore.

$$z = \exp^{\ln z} = \exp^{(\ln x+\ln y)}$$
Speeding Up **Convolution** (The Convolution Theorem)

Let $z$ be the product of two numbers, $x$ and $y$, that is,

$$z = xy$$

Taking logarithms of both sides, one obtains

$$\ln z = \ln x + \ln y$$

Therefore.

$$z = \exp^{\ln z} = \exp^{(\ln x + \ln y)}$$

**Interpretation:** At the expense of two $\ln()$ and one $\exp()$ computations, multiplication is reduced to admission
Speeding Up Rotation

Another analogy: 2D rotation of a point by an angle $\alpha$ about the origin.

The standard approach, in Euclidean coordinates, involves a matrix multiplication:

$$
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} =
\begin{bmatrix}
  \cos \alpha & -\sin \alpha \\
  \sin \alpha & \cos \alpha
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
$$

Suppose we transform to polar coordinates:

$$(x, y) \rightarrow (\rho, \theta) \rightarrow (\rho, \theta + \alpha) \rightarrow (x', y')$$

Rotation becomes addition, at expense of one polar coordinate transform and one inverse polar coordinate transform.
Speeding Up **Convolution** (The Convolution Theorem)

Similarly, some image processing operations become cheaper in a transform domain

Gonzales & Woods (3rd ed.) Figure 2.39
Speeding Up Convolution (The Convolution Theorem)

Convolution Theorem:

Let \( i'(x, y) = f(x, y) \otimes i(x, y) \)

then  \( I'(w_x, w_y) = F(w_x, w_y) I(w_x, w_y) \)

where  \( I'(w_x, w_y), F(w_x, w_y), \) and \( I(w_x, w_y) \) are Fourier transforms of \( i'(x, y), f(x, y) \) and \( i(x, y) \)

At the expense of two Fourier transforms and one inverse Fourier transform, convolution can be reduced to (complex) multiplication
Let's take a detour ...
What follows is for fun
(you will NOT be tested on this)
Fourier Transform (you will NOT be tested on this)

Basic building block:

\[ A \sin(\omega x + \phi) \]

Fourier’s claim: Add enough of these to get any periodic signal you want!
Fourier Transform (you will NOT be tested on this)

Basic building block:

\[ A \sin(\omega x + \phi) \]

Fourier’s claim: Add enough of these to get any periodic signal you want!

Slide Credit: Ioannis (Yannis) Gkioulakas (CMU)
Fourier Transform (you will NOT be tested on this)

How would you generate this function?

\[
\text{Fourier Transform} \quad = \quad ? \quad + \quad ?
\]

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?

\[ \sin(2\pi x) = \text{?} \]
Fourier Transform (you will NOT be tested on this)

How would you generate this function?

\[
\sin(2\pi x) + \frac{1}{3}\sin(2\pi 3x)
\]

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?

\[ f(x) = \sin(2\pi x) + \frac{1}{3} \sin(2\pi 3x) \]

\[ \begin{align*}
\text{=} & \quad \sin(2\pi x) \\
\text{+} & \quad \frac{1}{3} \sin(2\pi 3x)
\end{align*} \]
Fourier Transform (you will NOT be tested on this)

How would you generate this function?

\[ \text{square wave} \approx ? + ? \]
**Fourier Transform** (you will **NOT** be tested on this)

How would you generate this function?

- A square wave
- Fourier Transform (you will **NOT** be tested on this)
Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
How would you generate this function?

\[ \text{square wave} \approx \text{Fourier Transform} \]

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Fourier Transform (you will NOT be tested on this)

How would you generate this function?

How would you express this mathematically?

Slide Credit: Ioannis (Yannis) Gkioulkekas (CMU)
**Fourier Transform** (you will **NOT** be tested on this)

How would you generate this function?

\[
\text{square wave} = A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k x)
\]

infinite sum of sine waves

*Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)*
Fourier Transform (you will NOT be tested on this)

Basic building block:

\[ A \sin(\omega x + \phi) \]

Fourier’s claim: Add enough of these to get any periodic signal you want!
Fourier Transform (you will **NOT** be tested on this)

Image from: Numerical Simulation and Fractal Analysis of Mesoscopic Scale Failure in Shale Using Digital Images
Fourier Transform (you will NOT be tested on this)

Forsyth & Ponce (2nd ed.) Figure 4.6
Fourier Transform (you will NOT be tested on this)

Forsyth & Ponce (2nd ed.) Figure 4.6

cheetah phase with zebra amplitude

zebra phase with cheetah amplitude
Fourier Transform (you will NOT be tested on this)

**Experiment:** Where do you see the stripes?
Fourier Transform (you will NOT be tested on this)

Campbell-Robson contrast sensitivity curve

Our eyes are sensitive to mid-range frequencies

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
What preceded was for fun
(you will **NOT** be tested on it)
Fourier Transform

Preview of Part 3 of your homework

Gala Contemplating the Mediterranean Sea Which at Twenty Meters Becomes the Portrait of Abraham Lincoln (Homage to Rothko)

Salvador Dali, 1976
Fourier Transform

Preview of Part 3 of your homework

Low-pass filtered version

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Fourier Transform

Preview of Part 3 of your homework

High-pass filtered version
Low-pass Filtering = “Smoothing”

Are all of these low-pass filters?
Low-pass Filtering = “Smoothing”

Are all of these low-pass filters?

**Low-pass filter**: Low pass filter filters out all of the high frequency content of the image, only low frequencies remain.
**Low-pass Filtering = “Smoothing”**

Are all of these **low-pass** filters?

**Low-pass filter**: Low pass filter filters out all of the high frequency content of the image, only low frequencies remain.
After long detour …

lets go back to efficiency
Speeding Up **Convolution** (The Convolution Theorem)

**Convolution Theorem:**

Let \( i'(x, y) = f(x, y) \otimes i(x, y) \)

then \( \mathcal{I}'(w_x, w_y) = \mathcal{F}(w_x, w_y) \mathcal{I}(w_x, w_y) \)

where \( \mathcal{I}'(w_x, w_y), \mathcal{F}(w_x, w_y), \) and \( \mathcal{I}(w_x, w_y) \) are Fourier transforms of \( i'(x, y), f(x, y) \) and \( i(x, y) \)

At the expense of two **Fourier** transforms and one inverse Fourier transform, convolution can be reduced to (complex) multiplication.
Speeding Up **Convolution** (The Convolution Theorem)

**General** implementation of convolution:

At each pixel, \((X, Y)\), there are \(m \times m\) multiplications.

There are \(n \times n\) pixels in \((X, Y)\).

**Total:** \(m^2 \times n^2\) multiplications.

**Convolution** if FFT space:

Cost of FFT/IFFT for image: \(O(n^2 \log n)\)

Cost of FFT/IFFT for filter: \(O(m^2 \log m)\)

Cost of convolution: \(O(n^2)\)  
**Note:** not a function of filter size !!!
Linear Filters: Properties (recall Lecture 3)

Let $\otimes$ denote convolution. Let $I(X,Y)$ be a digital image

**Superposition:** Let $F_1$ and $F_2$ be digital filters

$$(F_1 + F_2) \otimes I(X,Y) = F_1 \otimes I(X,Y) + F_2 \otimes I(X,Y)$$

**Scaling:** Let $F$ be digital filter and let $k$ be a scalar

$$(kF) \otimes I(X,Y) = F \otimes (kI(X,Y)) = k(F \otimes I(X,Y))$$

**Shift Invariance:** Output is local (i.e., no dependence on absolute position)

An operation is **linear** if it satisfies both **superposition** and **scaling**
Linear Filters: Additional Properties

Let $\otimes$ denote convolution. Let $I(X, Y)$ be a digital image. Let $F$ and $G$ be digital filters

— Convolution is **associative**. That is,

$$G \otimes (F \otimes I(X, Y)) = (G \otimes F) \otimes I(X, Y)$$

— Convolution is **symmetric**. That is,

$$(G \otimes F) \otimes I(X, Y) = (F \otimes G) \otimes I(X, Y)$$

Convolving $I(X, Y)$ with filter $F$ and then convolving the result with filter $G$ can be achieved in single step, namely convolving $I(X, Y)$ with filter $G \otimes F = F \otimes G$

**Note**: Correlation, in general, is **not associative**.
Example: Two Box Filters

```python
filter = boxfilter(3)
signal.correlate2d(filter, filter, 'full')
```

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

\[
\begin{array}{cccc}
1 & 2 & 3 & 2 & 1 \\
2 & 4 & 6 & 4 & 2 \\
3 & 6 & 9 & 6 & 3 \\
2 & 4 & 6 & 4 & 2 \\
1 & 2 & 3 & 2 & 1 \\
\end{array}
\]

3x3 Box 3x3 Box 3x3 Box
Example: Two Box Filters

Treat one filter as padded “image”

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\times
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
= \frac{1}{81}
\]
**Example**: Two Box Filters

Treat one filter as padded “image”

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

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

\[
\frac{1}{9} \times \frac{1}{9} = \frac{1}{81}
\]

3x3 Box

Output
Example: Two Box Filters

Treat one filter as padded “image”

\[
\begin{array}{ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

\[\frac{1}{9} \times \frac{1}{9} = \frac{1}{81}\]

3x3 Box

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

Output
Example: Two Box Filters

Treat one filter as padded “image”

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

3x3 Box

\[\times\]

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

3x3 Box

\[\frac{1}{9}\]

\[\frac{1}{9}\times\frac{1}{9}\times\frac{1}{9} = \frac{1}{81}\]

Output
Example: Two Box Filters

Treat one filter as padded “image”

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\times\frac{1}{9}
\begin{array}{cccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\times\frac{1}{9}
\begin{array}{cccccccc}
1 & 2 & 3 & 2 & 1 \\
2 & 4 & 6 & 4 & 2 \\
3 & 6 & 9 & 6 & 3 \\
2 & 4 & 6 & 4 & 2 \\
1 & 2 & 3 & 2 & 1 \\
\end{array}
=\frac{1}{81}
\]
Example: Two Box Filters

Treat one filter as padded “image”

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\times
\begin{array}{c}
1 \\
9 \\
1 \\
9 \\
\end{array}
= \begin{array}{cccc}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\times
\begin{array}{c}
1 \\
9 \\
1 \\
9 \\
\end{array}
= \begin{array}{cccc}
1 & 2 & 3 & 2 \\
1 & 2 & 3 & 2 \\
\end{array}
\begin{array}{cccc}
1 & 2 & 3 & 2 \\
2 & 4 & 6 & 4 \\
3 & 6 & 9 & 6 \\
2 & 4 & 6 & 4 \\
1 & 2 & 3 & 2 \\
\end{array}
= \begin{array}{cccc}
1 & 2 & 3 & 2 \\
2 & 4 & 6 & 4 \\
3 & 6 & 9 & 6 \\
2 & 4 & 6 & 4 \\
1 & 2 & 3 & 2 \\
\end{array}
= \frac{1}{81}
\]
Example: Two Box Filters

filter = boxfilter(3)
temp = signal.correlate2d(filter, filter, 'full')
signal.correlate2d(filter, temp, 'full')
Example: Separable Gaussian Filter

\[
\begin{array}{cccc}
\frac{1}{16} & 1 & 4 & 6 \\
\frac{1}{16} & 4 & 6 & 4 \\
\frac{1}{16} & 4 & 1 & 1
\end{array} \otimes \begin{array}{cccc}
1 & 4 & 6 & 4 \\
4 & 16 & 24 & 16 \\
6 & 24 & 36 & 24 \\
4 & 16 & 24 & 16 \\
1 & 4 & 6 & 4 \\
\end{array} = \frac{1}{256}
\]
Example: Separable Gaussian Filter

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & 4 & 6 & 4 & 1 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix} \times \frac{1}{16} \times \frac{1}{16} = \frac{1}{256}
\]
Example: Separable Gaussian Filter
Example: Separable Gaussian Filter

\[
\begin{array}{cccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\frac{1}{16} & 1 & 4 & 6 & 4 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 4 & 6 & 4 & 1 \\
\frac{1}{16} & 6 & 4 & 1 \\
\hline
\end{array}
\times
\begin{array}{cccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 4 & 6 & 4 & 1 \\
4 & 16 & 24 & 16 & 4 \\
6 & 24 & 36 & 24 & 6 \\
4 & 16 & 24 & 16 & 4 \\
1 & 4 & 6 & 4 & 1 \\
\hline
\end{array}
= \frac{1}{256}
\]
**Example: Separable Gaussian Filter**

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & 4 & 6 & 4 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{array}
\times
\begin{array}{c}
1 \\
4 \\
6 \\
4 \\
1
\end{array}
= \frac{1}{256} 
\begin{array}{cccccc}
1 & 4 & 6 & 4 & 1 \\
4 & 16 & 24 & 16 & 4 \\
6 & 24 & 36 & 24 & 6 \\
4 & 16 & 24 & 16 & 4 \\
1 & 4 & 6 & 4 & 1
\end{array}
Pre-Convolving Filters

Convolving two filters of size $m \times m$ and $n \times n$ results in filter of size:

$$\left( n + 2 \left\lfloor \frac{m}{2} \right\rfloor \right) \times \left( n + 2 \left\lfloor \frac{m}{2} \right\rfloor \right)$$

More broadly for a set of $K$ filters of sizes $m_k \times m_k$ the resulting filter will have size:

$$\left( m_1 + 2 \sum_{k=2}^{K} \left\lfloor \frac{m_k}{2} \right\rfloor \right) \times \left( m_1 + 2 \sum_{k=2}^{K} \left\lfloor \frac{m_k}{2} \right\rfloor \right)$$
Gaussian: An Additional Property

Let \( \otimes \) denote convolution. Let \( G_{\sigma_1}(x) \) and \( G_{\sigma_2}(x) \) be two 1D Gaussians

\[
G_{\sigma_1}(x) \otimes G_{\sigma_2}(x) = G_{\sqrt{\sigma_1^2 + \sigma_2^2}}(x)
\]

Convolution of two Gaussians is another Gaussian

**Special case:** Convolving with \( G_{\sigma}(x) \) twice is equivalent to \( G_{\sqrt{2\sigma}}(x) \)
Summary

We covered two additional linear filters: **Gaussian, pillbox**

**Separability** (of a 2D filter) allows for more efficient implementation (as two 1D filters)

The Convolution Theorem: In **Fourier** space, convolution can be reduced to (complex) multiplication
Menu for Today (January 16, 2020)

Topics:

— Gaussian and Pillbox filters
— Separability
— The Convolution Theorem
— Non-linear filters

Readings:

— Today’s Lecture: none
— Next Lecture: Forsyth & Ponce (2nd ed.) 4.4

Reminders:

— Assignment 1: Image Filtering and Hybrid Images due January 28-th
— Today my office hours will start at 3:30pm (not 3pm as posted)