# Spatial Filtering

Noise removal

# Vector representation of linear filtering

$$
\begin{aligned}
R &= w_1 z_1 + w_2 z_2 + \ \dots \ + w_{mn} z_{mn} \\
&= \sum_{k=1}^{mn} w_k z_k \\
&= \mathbf{w}^T \mathbf{z}
\end{aligned}
$$

# Vector representation of linear filtering

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \ \dots \ + w_9 z_9 \\ &= \sum_{k=1}^{9} w_k z_k \\ &= \mathbf{w}^T \mathbf{z} \end{aligned}$$

# Smoothing Linear Filter (Average filter)

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

# Average filter

$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)}$$

# Gaussian filter

$$h(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

# Average filter

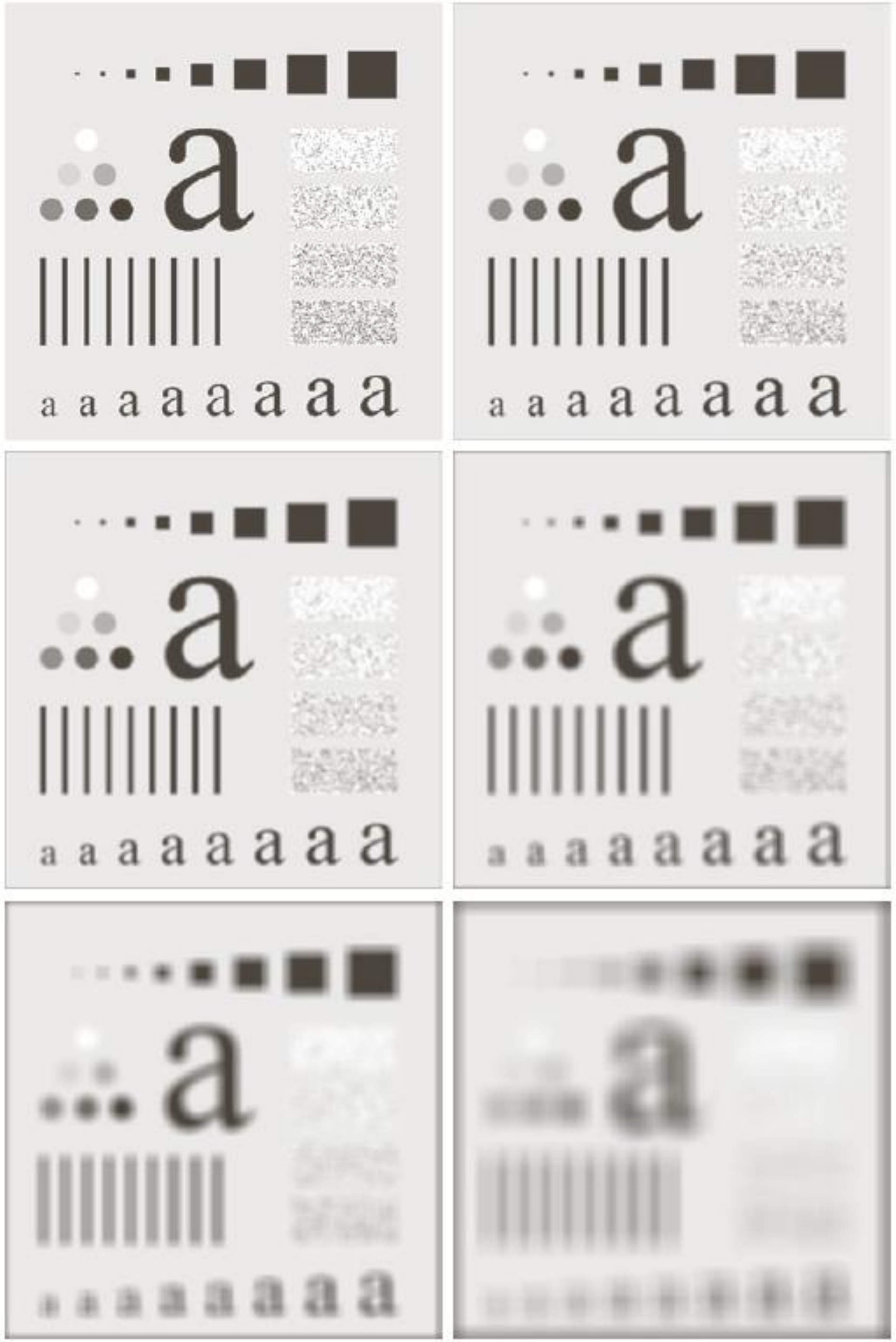
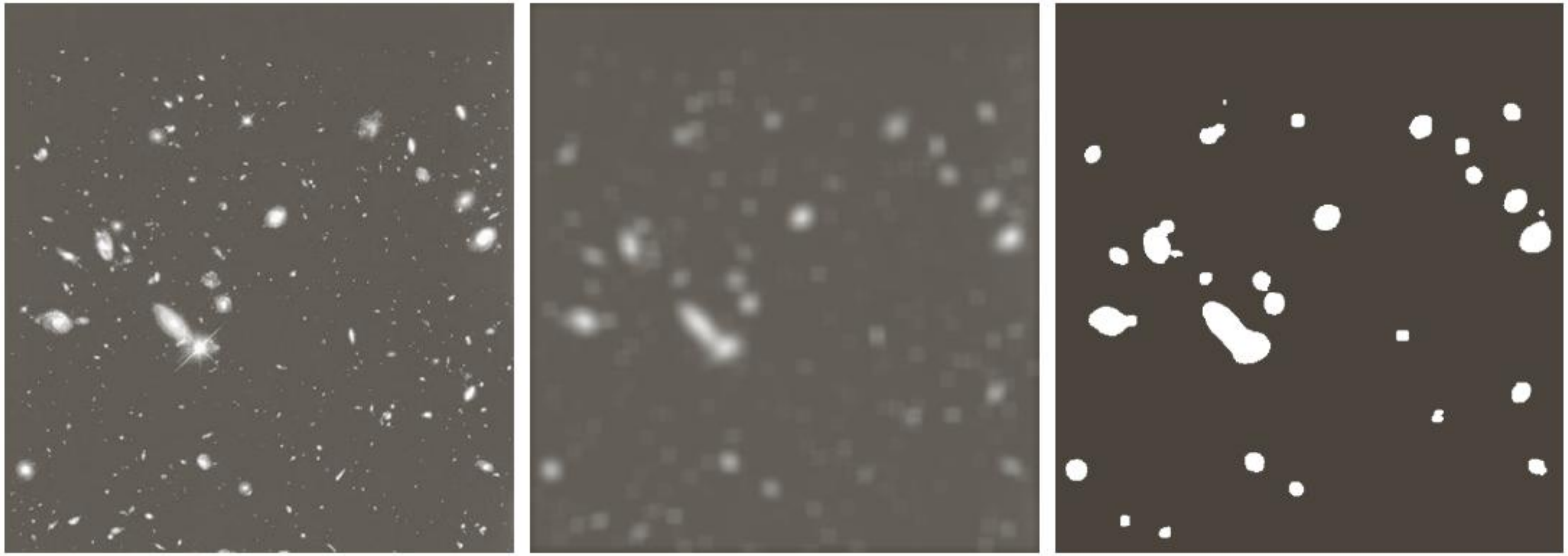


**FIGURE 3.33** (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f

# Average filter



a b c

**FIGURE 3.34** (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

# Image blurring Algo

1. Define the type and size of filter mask
2. Image Padding depending on the size of filter mask
3. Apply the filter mask on input image using correlation or convolution technique
4. Crop the output image to make it of same size as input image

# Image blurring in OpenCV

```
Blur(image, smoothed_image, Size(3, 3));

GaussianBlur(image, smoothed_image, Size(5, 5), 1.5);
```

# Inserting Gaussian Noise

```
import cv2
import numpy as np
# Load the image
img = cv2.imread('test_image.jpg')
# Generate random Gaussian noise
mean = 0
stddev = 180
noise = np.zeros(img.shape, np.uint8)
cv2.randn(noise, mean, stddev)
# Add noise to image
noisy_img = cv2.add(img, noise)
# Save noisy image
cv2.imwrite('noisy_img.jpg', noisy_img)
```

# Food for thought!

1. What is the purpose of smoothing filters in image processing?
2. What is an average filter and how does it work?
3. How is a Gaussian filter different from an average filter?
4. Why is image padding required before applying a filter mask?
5. What are the basic steps involved in an image blurring algorithm?

# Programming assignment

- Implement smoothing filters to reduce noise in images and analyze the difference between average and Gaussian filtering techniques.
- **Concepts Used**
  - Spatial smoothing
  - Average filter
  - Gaussian filter
  - Filter mask and padding
  - Correlation / Convolution
  - Noise reduction
- **Tasks**
  - Read a grayscale image.
  - Add Gaussian noise to the image.
  - Apply a 3×3 average filter to smooth the noisy image.
  - Apply a Gaussian filter (e.g., 5×5 mask) and generate the output.
  - Display the original, noisy, and filtered images.
  - Compare the results and briefly comment on which filter preserves image details better.

# AI supported self-learning (Prompts compatible with ChatGPT)

Active Learners (Learning by Doing)

1. Provide a small grayscale matrix and a 3×3 average filter. Ask me to manually compute the filtered output, then verify my answer step by step.
2. Guide me in writing a Python/OpenCV program that adds Gaussian noise to an image and removes it using both average and Gaussian filters.

Reflective Learners (Learning by Thinking)

1. Explain why smoothing filters reduce noise and summarize the mathematical reasoning behind weighted averaging.
2. Conceptually compare average and Gaussian filters and explain why Gaussian filtering typically preserves details better.

Sensing Learners (Concrete & Practical)

1. Use real pixel values to demonstrate how an average filter blurs an image and reduces noise.
2. Provide a practical example where Gaussian filtering is preferred over average filtering.

Intuitive Learners (Concepts & Patterns)

1. Explain Gaussian filtering as a weighted spatial operation based on the normal distribution.
2. Why does increasing filter size increase blurring? Explain the mathematical trade-off between noise reduction and detail preservation.

Visual Learners (Diagrams & Structure)

1. Show an original image, a noisy version, and outputs after average and Gaussian filtering, and visually explain the differences.
2. Illustrate how padding works and why it is necessary before applying a filter mask.

Verbal Learners (Words & Explanation)

1. Explain smoothing filters using an analogy such as averaging neighboring opinions to reduce extreme values.
2. Describe the difference between average and Gaussian filtering in simple teaching language.

Sequential Learners (Step-by-Step Logic)

1. Break down the image blurring algorithm into ordered steps, including padding, mask application, and cropping.
2. Explain step by step how convolution is applied when performing Gaussian smoothing.

Global Learners (Big Picture First)

1. Explain the overall purpose of smoothing filters in the image enhancement pipeline before discussing implementation.
2. Provide a big-picture comparison of noise reduction techniques in spatial filtering.