اصول پردازش تصویر

Principles of Image Processing

مصطفی کمالی تبریزی

۱۲ مهر ۱۳۹۹

جلسه پنجم

Image Enhancement (Spatial Filtering)

Neighborhood in Spatial Domain

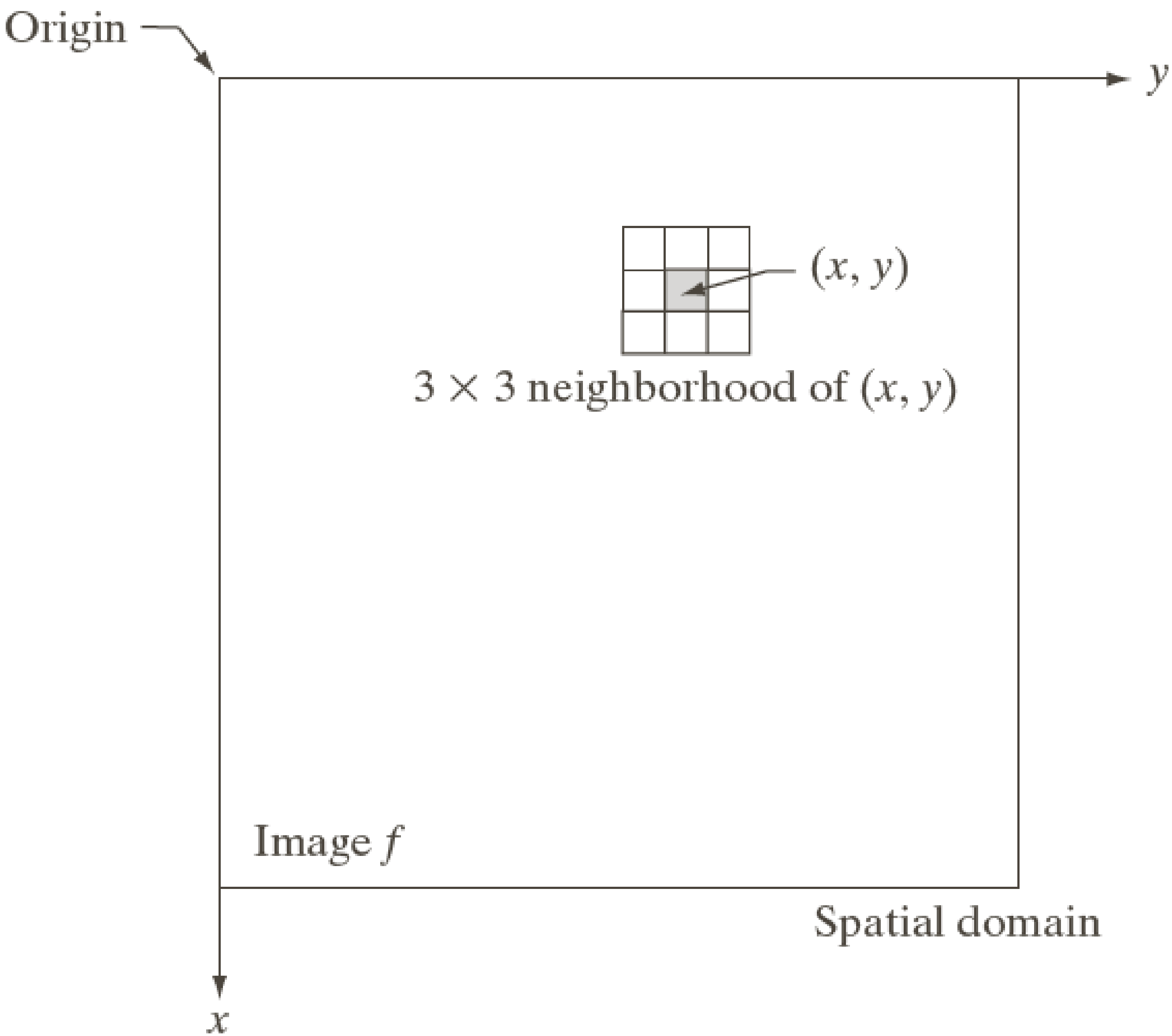
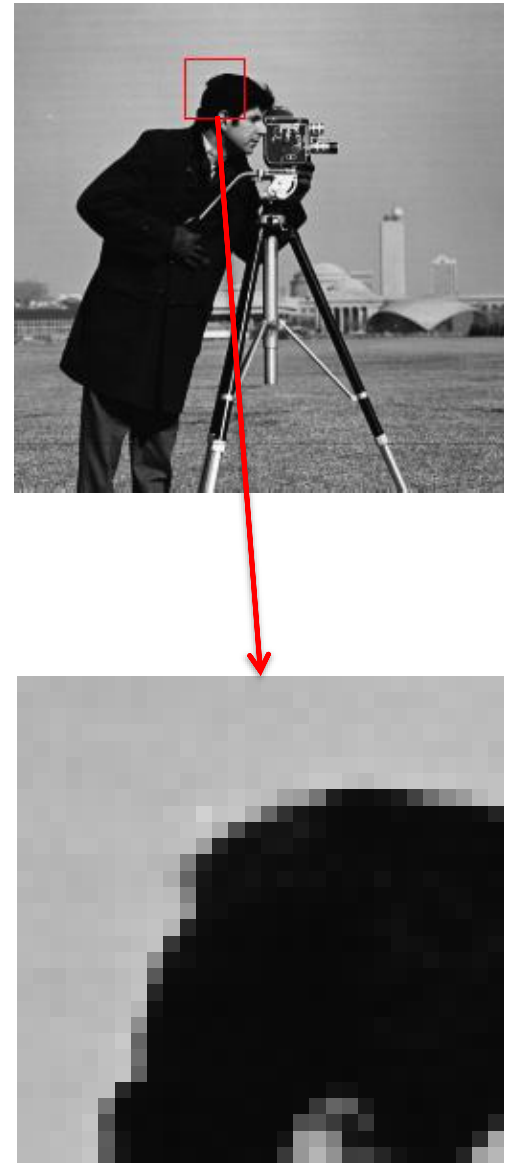


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	182	177	178	181	181	182	183	181	185	185	187	188	187	188	188
2	180	184	181	182	182	185	181	186	184	189	185	188	189	187	186
3	181	183	181	182	182	189	187	187	187	189	189	190	189	186	190
4	179	183	183	180	184	181	187	185	188	199	189	192	199	198	192
5	185	187	185	186	184	191	195	169	49	15	10	10	11	12	15
6	181	181	188	187	186	191	37	13	21	12	11	11	11	12	12
7	185	183	181	186	187	100	13	18	18	15	12	17	12	12	10
8	184	183	186	189	192	148	15	10	9	9	9	11	12	12	11
9	185	182	184	185	194	14	10	10	8	8	8	10	15	10	13
10	182	177	182	187	88	11	10	10	9	9	10	12	10	11	13
11	183	179	183	190	17	9	8	9	9	9	9	8	11	13	11
12	183	186	189	201	11	9	10	10	9	9	9	11	13	11	9
13	185	183	186	196	11	10	10	10	9	9	8	10	10	11	10
14	184	185	190	11	9	9	9	10	9	56	89	10	8	10	10
15	185	189	193	18	10	9	10	9	20	163	21	11	9	11	42



30´ 30

Every other row and column

Cross-Correlation

Image

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	182	177	178	181	181	182	183	181	185	185	187	188	187	188	188
2	180	184	181	182	182	185	181	186	184	189	185	188	189	187	186
3	181	183	181	182	182	189	187	187	187	189	189	190	189	186	190
4	179	183	183	180	184	181	187	185	188	199	189	192	199	198	192
5	185	187	185	186	184	191	195	169	49	15	10	10	11	12	15
6	181	181	188	187	186	191	37	13	21	12	11	11	11	12	12
7	185	183	181	186	187	100	13	18	18	15	12	17	12	12	10
8	184	183	186	189	192	148	15	10	9	9	9	11	12	12	11
9	185	182	184	185	194	14	10	10	8	8	8	10	15	10	13
10	182	177	182	187	88	11	10	10	9	9	10	12	10	11	13
11	183	179	183	190	17	9	8	9	9	9	9	8	11	13	11
12	183	186	189	201	11	9	10	10	9	9	9	11	13	11	9
13	185	183	186	196	11	10	10	10	9	9	8	10	10	11	10
14	184	185	190	11	9	9	9	10	9	56	89	10	8	10	10
15	185	189	193	18	10	9	10	9	20	163	21	11	9	11	42

Filter / Kernel / Mask

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Cross-Correlation

Image

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	182	177	178	181	181	182	183	181	185	185	187	188	187	188	188
2	180	184	181	182	182	185	181	186	184	189	185	188	189	187	186
3	181	183	181	182	182	189	187	187	187	189	189	190	189	186	190
4	179	183	183	180	184	181	187	185	188	199	189	192	199	198	192
5	185	187	185	186	184	191	195	169	49	15	10	10	11	12	15
6	181	181	188	187	186	191	37	13	21	12	11	11	11	12	12
7	185	183	181	186	187	100	13	18	18	15	12	17	12	12	10
8	184	183	186	189	192	148	15	10	9	9	9	11	12	12	11
9	185	182	184	185	194	14	10	10	8	8	8	10	15	10	13
10	182	177	182	187	88	11	10	10	9	9	10	12	10	11	13
11	183	179	183	190	17	9	8	9	9	9	9	8	11	13	11
12	183	186	189	201	11	9	10	10	9	9	9	11	13	11	9
13	185	183	186	196	11	10	10	10	9	9	8	10	10	11	10
14	184	185	190	11	9	9	9	10	9	56	89	10	8	10	10
15	185	189	193	18	10	9	10	9	20	163	21	11	9	11	42

Filter / Kernel / Mask

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

195	169	49
37	13	21
13	18	18

$$\frac{195+169+49+37+13+21+13+18+18}{9} = 59.22$$

Cross-Correlation

Image

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3															
4															
5															
6								59							
7															
8															
9															
10															
11															
12															
13															
14															
15															

Filter / Kernel / Mask

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Cross-Correlation

Image

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	182	177	178	181	181	182	183	181	185	185	187	188	187	188	188
2	180	184	181	182	182	185	181	186	184	189	185	188	189	187	186
3	181	183	181	182	182	189	187	187	187	189	189	190	189	186	190
4	179	183	183	180	184	181	187	185	188	199	189	192	199	198	192
5	185	187	185	186	184	191	195	169	49	15	10	10	11	12	15
6	181	181	188	187	186	191	37	13	21	12	11	11	11	12	12
7	185	183	181	186	187	100	13	18	18	15	12	17	12	12	10
8	184	183	186	189	192	148	15	10	9	9	9	11	12	12	11
9	185	182	184	185	194	14	10	10	8	8	8	10	15	10	13
10	182	177	182	187	88	11	10	10	9	9	10	12	10	11	13
11	183	179	183	190	17	9	8	9	9	9	9	8	11	13	11
12	183	186	189	201	11	9	10	10	9	9	9	11	13	11	9
13	185	183	186	196	11	10	10	10	9	9	8	10	10	11	10
14	184	185	190	11	9	9	9	10	9	56	89	10	8	10	10
15	185	189	193	18	10	9	10	9	20	163	21	11	9	11	42

Filter / Kernel / Mask

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

9	9	8
9	9	11
9	8	10

$$\frac{9+9+8+9+9+11+9+8+10}{9} = 9.11$$

Cross-Correlation

Image

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3															
4															
5															
6								59							
7															
8															
9															
10															
11															
12											9				
13															
14															
15															

Filter / Kernel / Mask

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

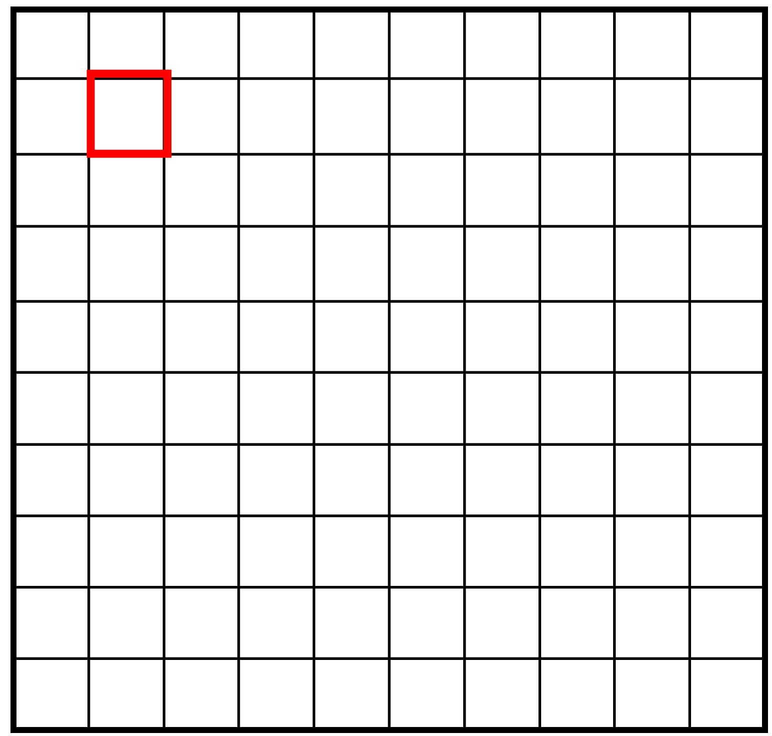
Image filtering
Smoothing, box filter

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$$f\,[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$



$$h[m,n]=\sum_{k,l} g[k,l]\ f\,[m+k,n+l]$$

Credit: S. Seitz

Image filtering
Smoothing, box filter

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10							

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering
Smoothing, box filter

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$

	0	10	20						

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering
Smoothing, box filter

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering
Smoothing, box filter

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering
Smoothing, box filter

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
				?					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering
Smoothing, box filter

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
						?			
				50					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering
Smoothing, box filter

$g[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

in Python:

```
cv2.filter2D
scipy.signal.convolve2D
```

Credit: S. Seitz

Smoothing (Box Filter)

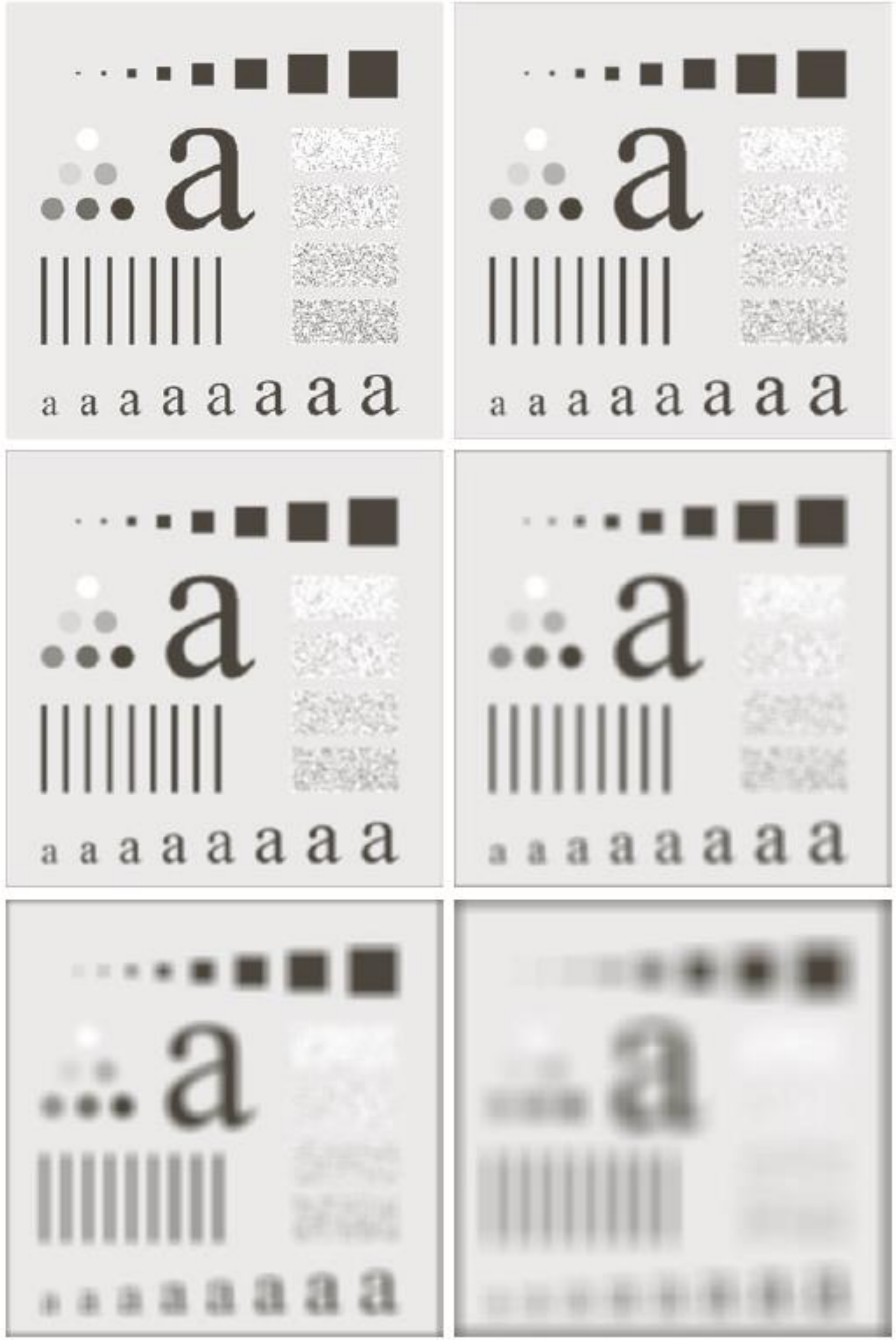
$$g[\cdot\,,\cdot\,]$$

What does it do?

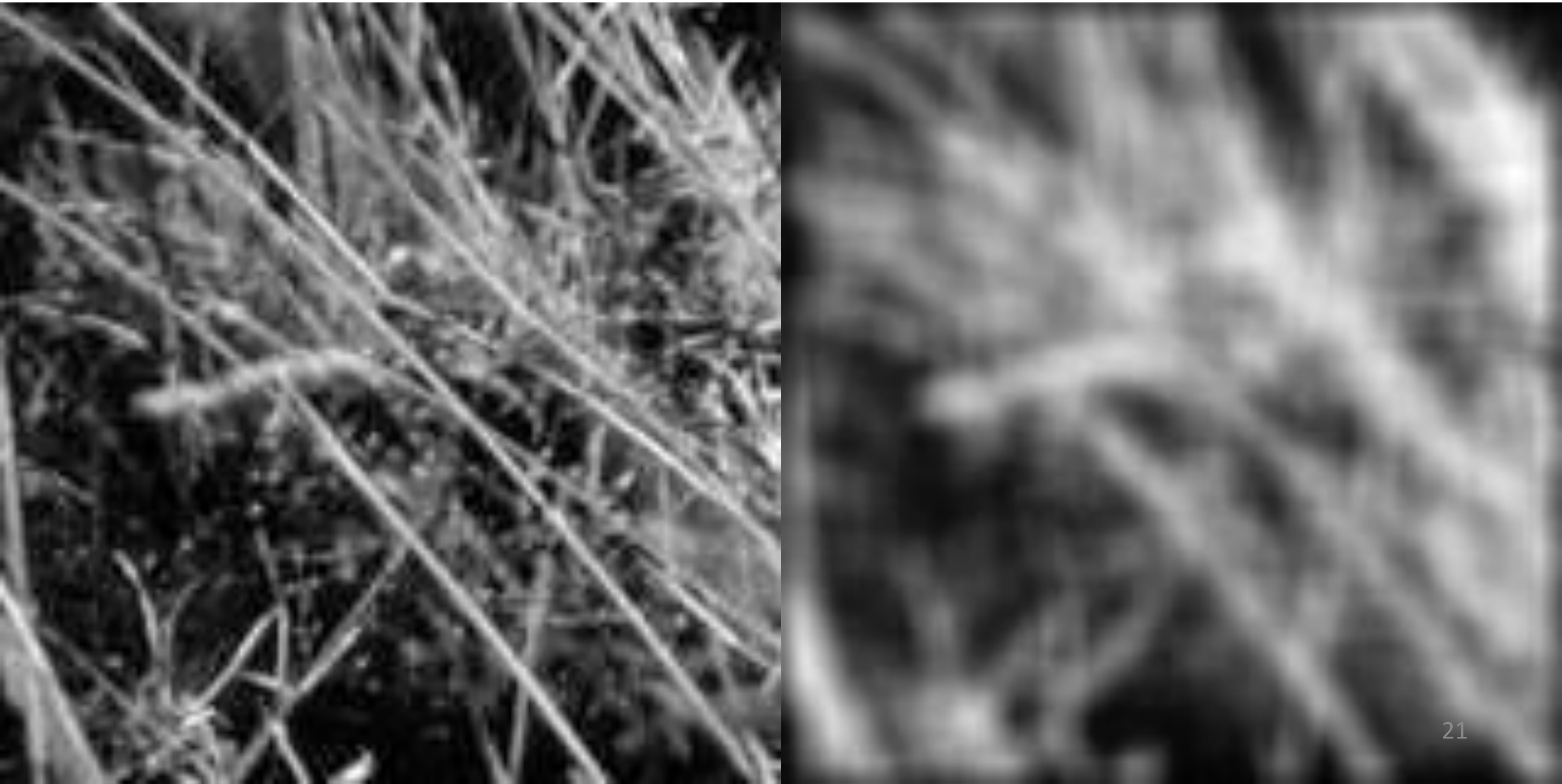
- Replaces each pixel with an average of its neighborhood
- Achieves smoothing effect (removes sharp features)

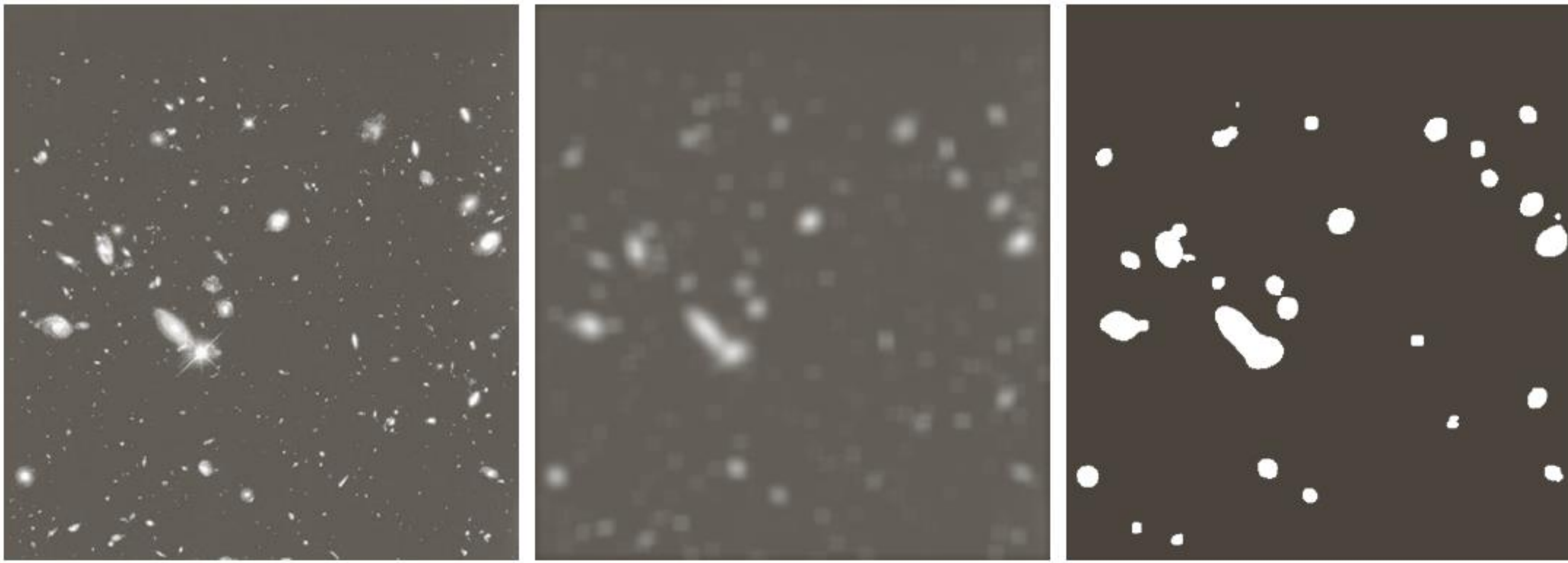
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Slide Credit: David Lowe (UBC)



Smoothing, Blurring





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.)

Practice with Linear Filters

Original

0	0	0
0	1	0
0	0	0

?

Source: D. Lowe

Practice with Linear Filters

Original

0	0	0
0	1	0
0	0	0

Filtered
(no change)

Source: D. Lowe

Practice with Linear Filters

Original

0	0	0
0	0	1
0	0	0

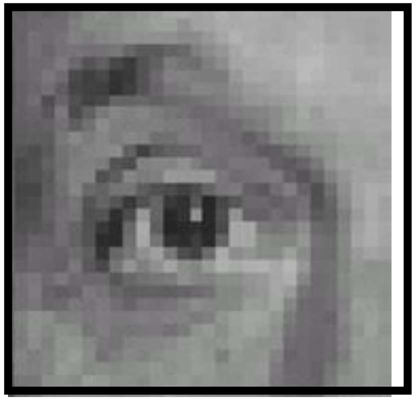
?

Source: D. Lowe

Practice with Linear Filters

Original

0	0	0
0	0	1
0	0	0



Shifted left
by 1 pixel

Source: D. Lowe

Practice with Linear Filters

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

?

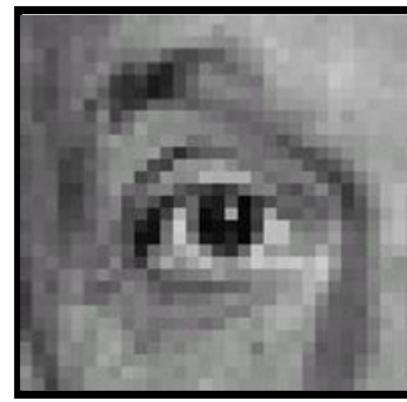
Source: D. Lowe

Practice with Linear Filters

Sharp

Original

$$\begin{bmatrix}0 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$



Sharpening filter

- Accentuates differences with local average

Source: D. Lowe

Cross-Correlation Filtering

Assume the averaging window is $(2k+1)´\ (2k+1)$

$$h[m,n] = \frac{1}{(2k+1)^2} \sum_{j=-k}^{k} \sum_{i=-k}^{k} f[m+i, n+j]$$

Generalization:

$$h[m,n] = \sum_{j} \sum_{i} g[i,j] f[m+i, n+j]$$

$$h[m,n] = \sum_{j=-k}^{k} \sum_{i=-k}^{k} g[k+1+i, k+1+j] f[m+i, n+j]$$

Cross-Correlation Operation: $H = G \otimes F$

filter, kernel, mask

Convolution

$$h[m,n] = \sum_{j=-k}^{k} \sum_{i=-k}^{k} g[k+1-i, k+1-j] f[m+i, n+j]$$

$$h[m,n] = \sum_{j} \sum_{i} g[-i,-j] f[m+i, n+j]$$

Convolution Operation: $H = G * F$

filter, kernel, mask

Convolution is nice!

Convolution is a multiplication-like operation:

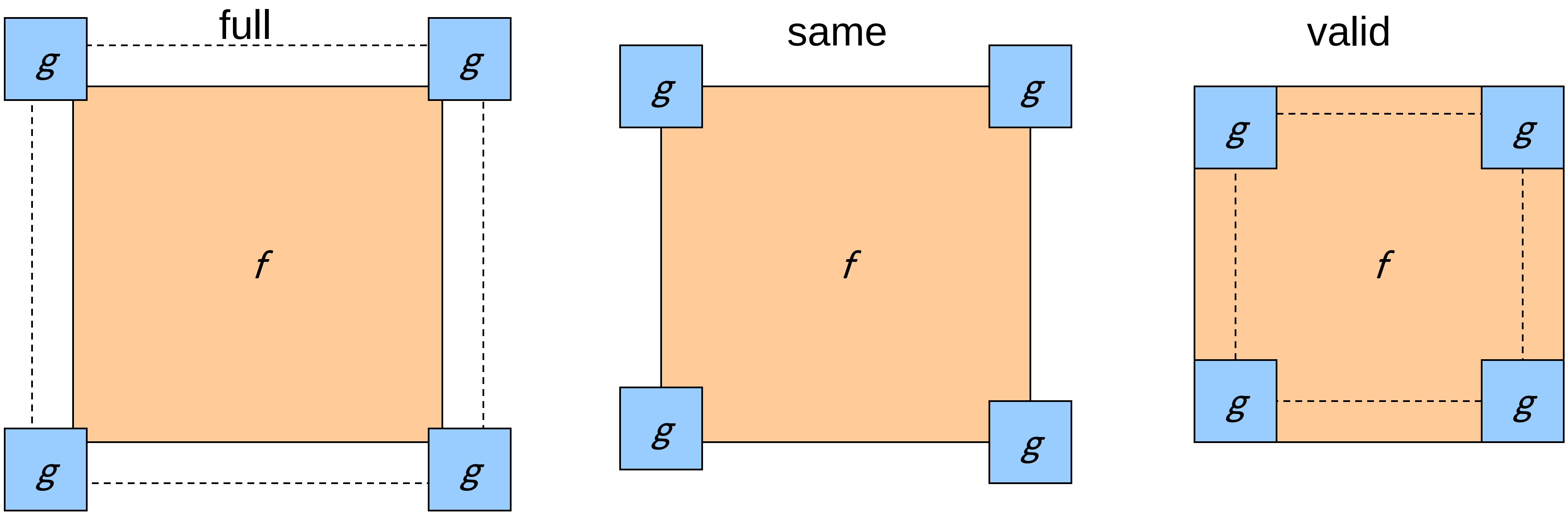
- Commutative: $f * g = g * f$
- Associative: $g_1 * (g_2 * f) = (g_1 * g_2) * f$
- Distributes over addition: $g * (f_1 + f_2) = (g * f_1) + (g * f_2)$
- Scalars factor out: $\alpha g * f = g * \alpha f = \alpha(g * f)$

Under proper conditions:

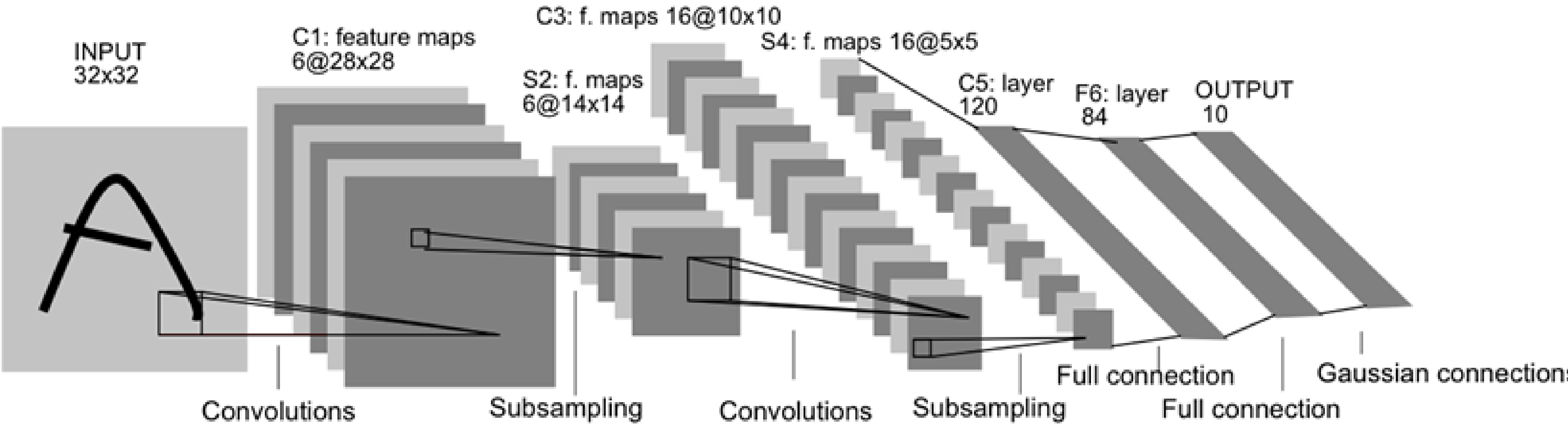
- Convolution theorem: $g * f \leftrightarrow G.F$ and $g.f \leftrightarrow G * F$
- Derivatives: $(g * f)' = g' * f = g * f'$

Practical matters

- What is the size of the output?
- Python: `convolve2D(g, f, mode)`
  - *mode* = ‘full’: output size is sum of sizes of f and g
  - *mode* = ‘same’: output size is same as f
  - *mode* = ‘valid’: output size is difference of sizes of f and g



Convolutional Neural Networks (CNNs)



An early (Le-Net5) Convolutional Neural Network design, LeNet-5, used for recognition of digits

- *Padding*
- *Stride*
- *Kernel Size*

Correlation vs. Convolution

- 2d correlation

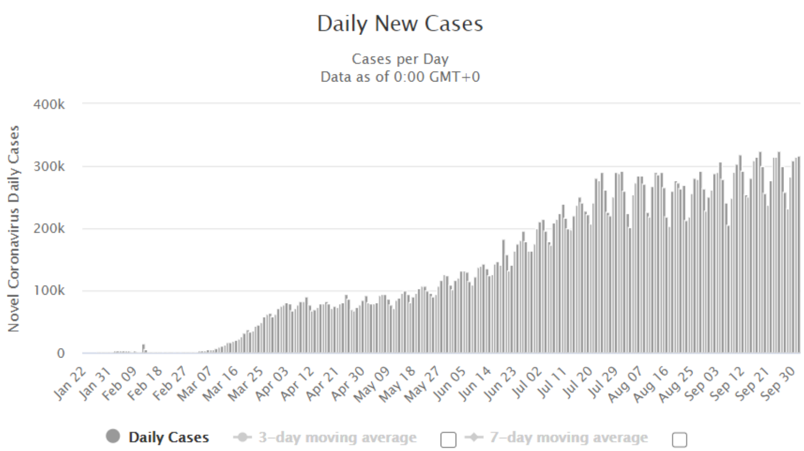
```
im_fil = cv2.filter2D(im, -1, fil)
```

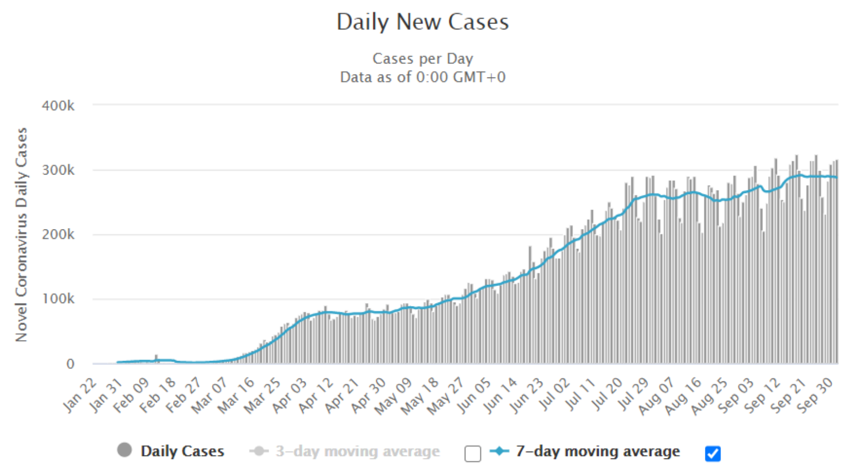
- 2d convolution

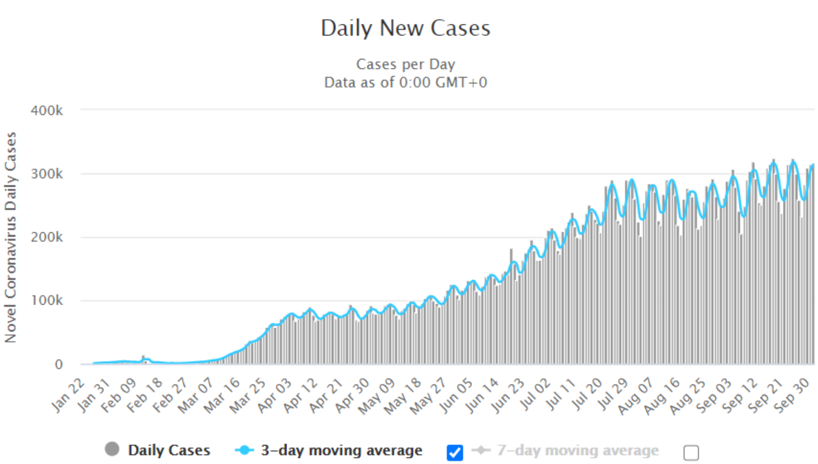
```
im_fil = scipy.signal.convolve2D(im, fil, [opts])
```

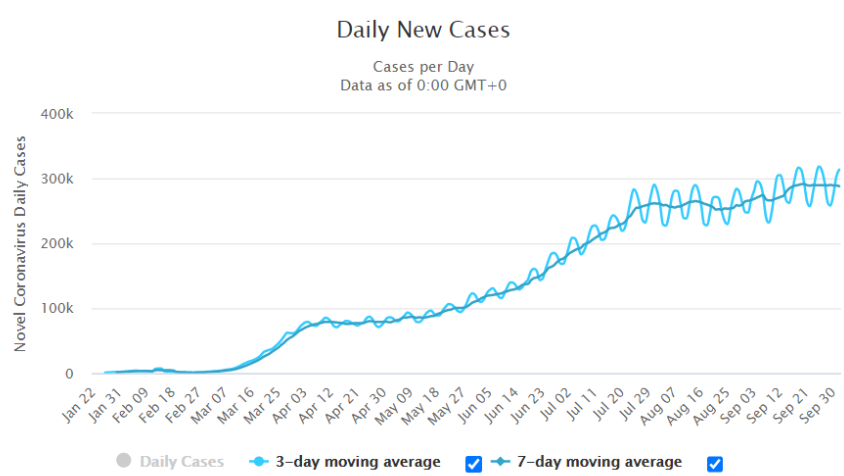
- “convolve” mirrors the kernel, while “filter” doesn’t

`cv2.filter2d(im, -1, cv2.flip(fil,-1))` same as
`signal.convolve2d(im, fil, mode=‘same’, boundary=‘symm’)`


Daily New Cases
Cases per Day
Data as of 0:00 GMT+0
Novel Coronavirus Daily Cases
400k
300k
200k
100k
0
Jan 22 Jan 31 Feb 09 Feb 18 Feb 27 Mar 07 Mar 16 Mar 25 Apr 03 Apr 12 Apr 21 Apr 30 May 09 May 18 May 27 Jun 05 Jun 14 Jun 23 Jul 02 Jul 11 Jul 20 Jul 29 Aug 07 Aug 16 Aug 25 Sep 03 Sep 12 Sep 21 Sep 30
Daily Cases
3-day moving average
7-day moving average


Daily New Cases
Cases per Day
Data as of 0:00 GMT+0
Novel Coronavirus Daily Cases
400k
300k
200k
100k
0
Jan 22 Jan 31 Feb 09 Feb 18 Feb 27 Mar 07 Mar 16 Mar 25 Apr 03 Apr 12 Apr 21 Apr 30 May 09 May 18 May 27 Jun 05 Jun 14 Jun 23 Jul 02 Jul 11 Jul 20 Jul 29 Aug 07 Aug 16 Aug 25 Sep 03 Sep 12 Sep 21 Sep 30
Daily Cases
3-day moving average
7-day moving average


Daily New Cases
Cases per Day
Data as of 0:00 GMT+0
Novel Coronavirus Daily Cases
400k
300k
200k
100k
0
Jan 22 Jan 31 Feb 09 Feb 18 Feb 27 Mar 07 Mar 16 Mar 25 Apr 03 Apr 12 Apr 21 Apr 30 May 09 May 18 May 27 Jun 05 Jun 14 Jun 23 Jul 02 Jul 11 Jul 20 Jul 29 Aug 07 Aug 16 Aug 25 Sep 03 Sep 12 Sep 21 Sep 30
Daily Cases
3-day moving average
7-day moving average


Daily New Cases
Cases per Day
Data as of 0:00 GMT+0
Novel Coronavirus Daily Cases
400k
300k
200k
100k
0
Jan 22 Jan 31 Feb 09 Feb 18 Feb 27 Mar 07 Mar 16 Mar 25 Apr 03 Apr 12 Apr 21 Apr 30 May 09 May 18 May 27 Jun 05 Jun 14 Jun 23 Jul 02 Jul 11 Jul 20 Jul 29 Aug 07 Aug 16 Aug 25 Sep 03 Sep 12 Sep 21 Sep 30
Daily Cases
3-day moving average
7-day moving average

Weighted Averaging

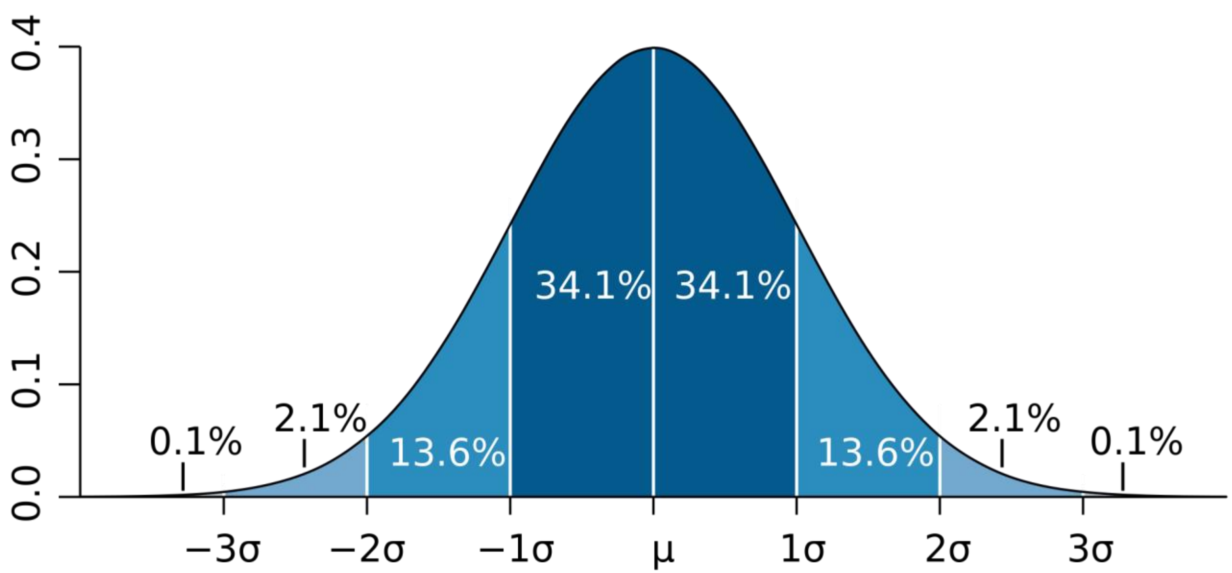
$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array} \qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

Gaussian (Normal) Distribution (Probability Density Function)

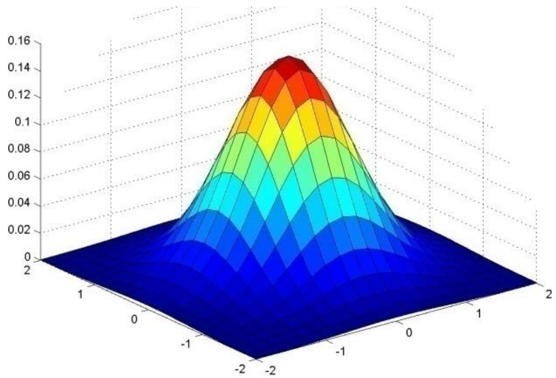
$$f(x, m, s) = \frac{1}{s\sqrt{2p}} e^{-\frac{(x-m)^2}{2s^2}}$$

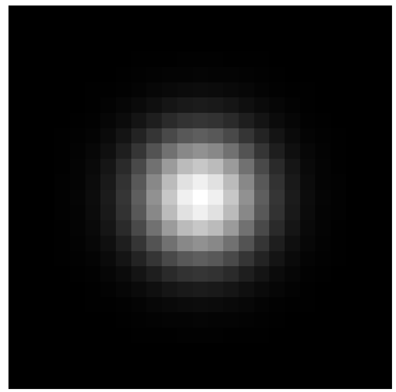


99.6%

Important Filter: Gaussian

- Spatially-weighted average





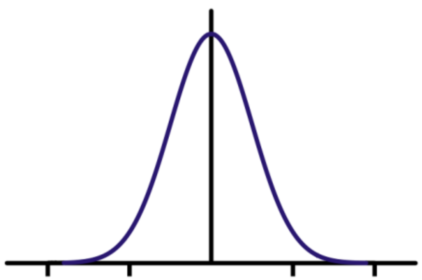
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, σ = 1

$$G_s = \frac{1}{2ps^2} e^{-\frac{(x^2+y^2)}{2s^2}}$$

Slide Credit: Christopher Rasmussen

Gaussian Filter

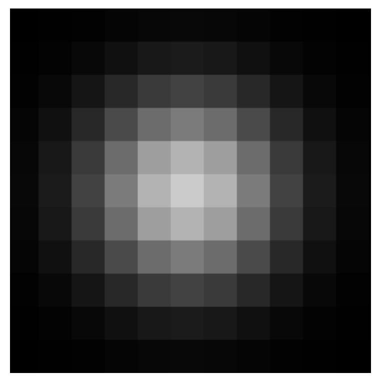


$$G_s(x,y) = \frac{1}{Z} e^{-\frac{(x^2+y^2)}{2s^2}}$$

Compute empirically

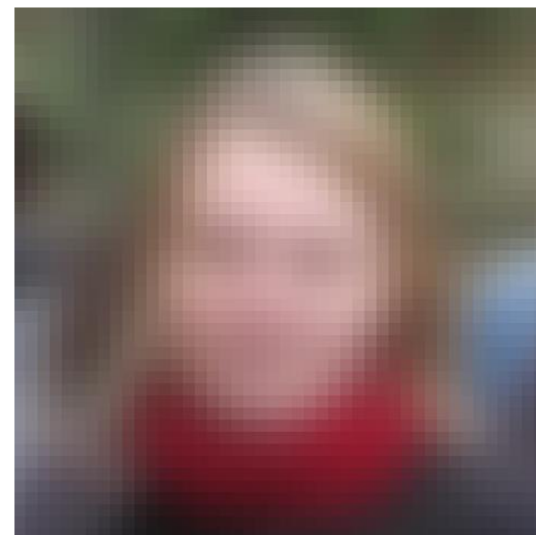
Input image f

$*$



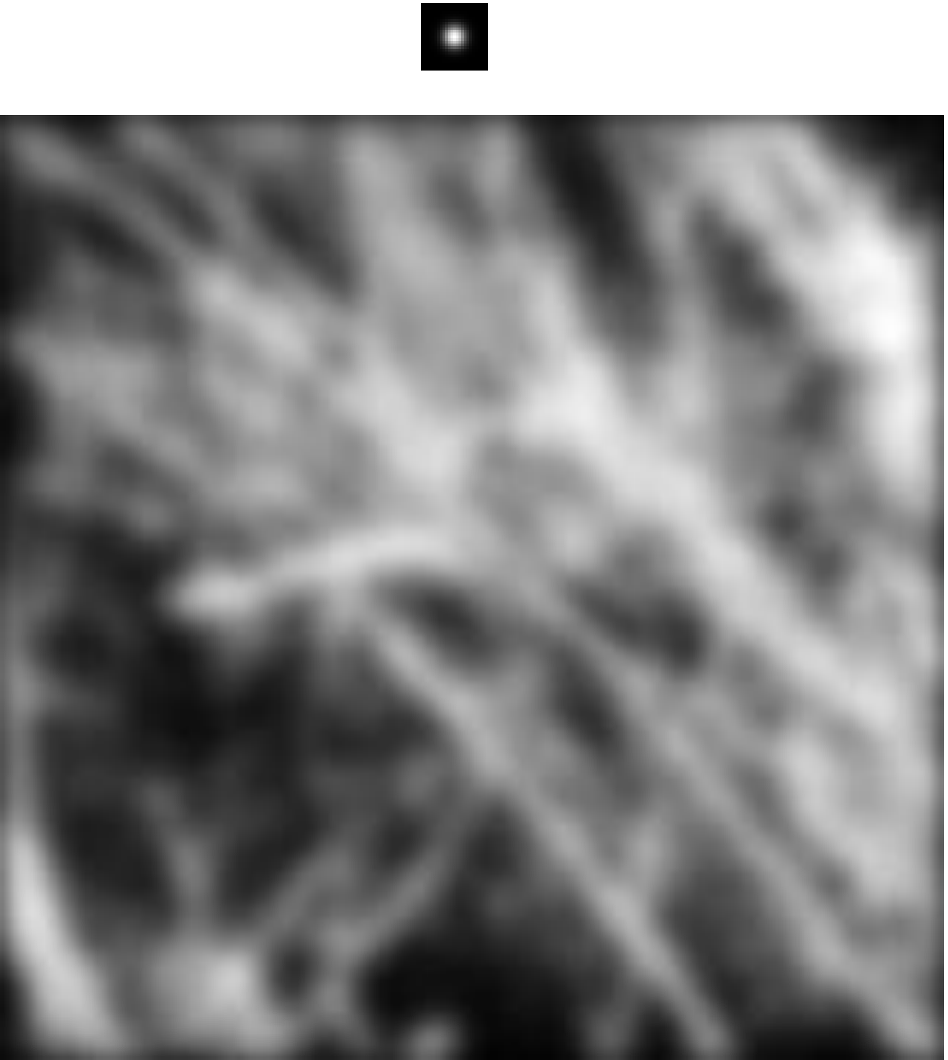
Filter h

$=$



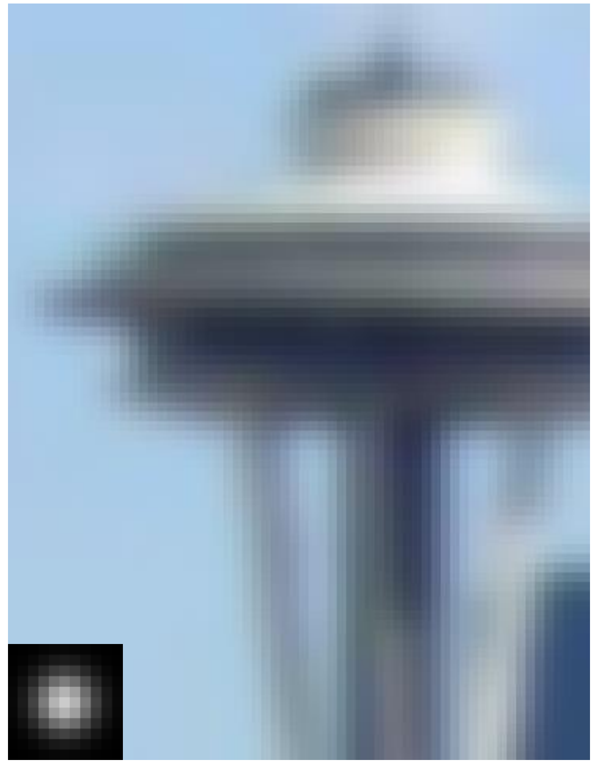
Output image g

Gaussian Filter

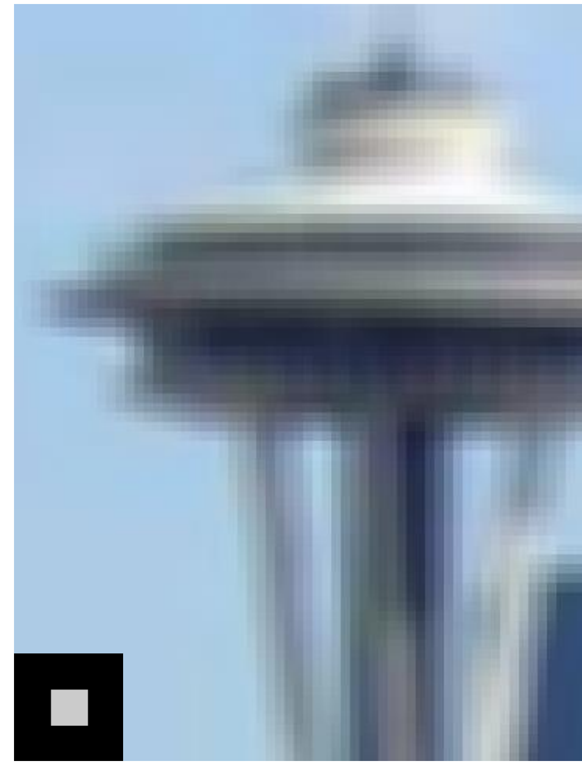


Box Filter

Gaussian vs. Mean Filters



Gaussian

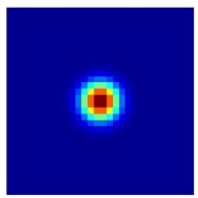


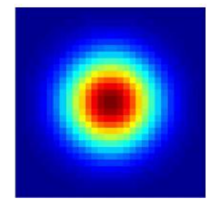
Box filter

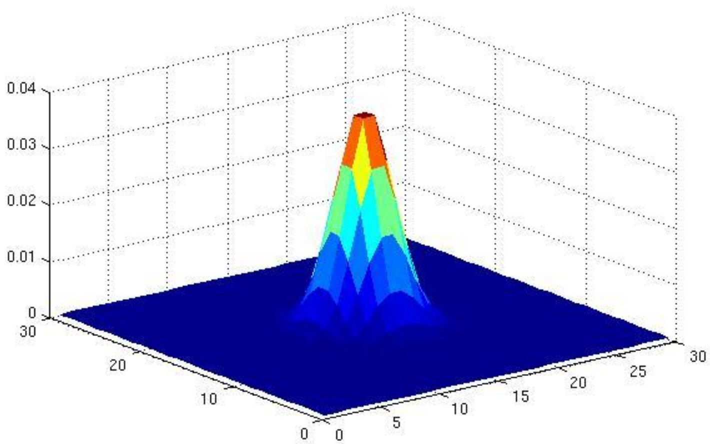
What does real blur look like?

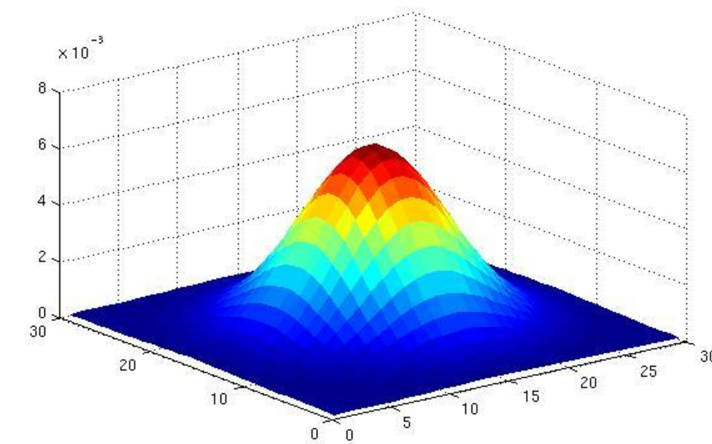
Gaussian Filters

- What parameters matter here?
- **Variance** of Gaussian: determines extent of smoothing





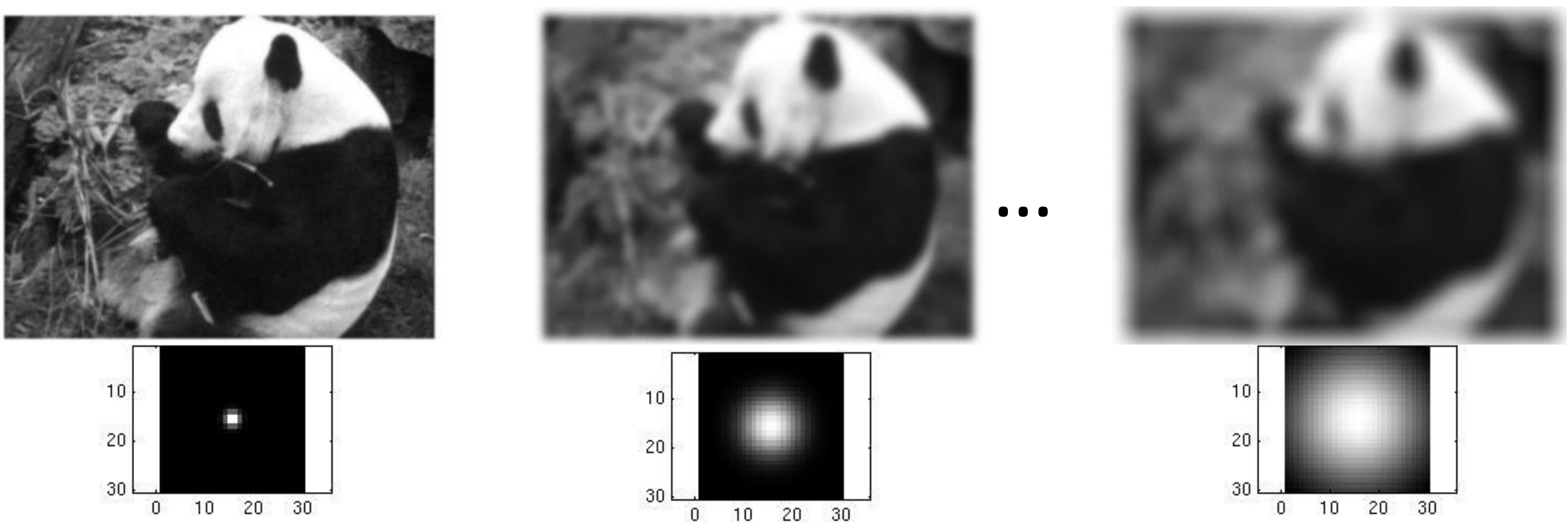




Source: K. Grauman

Smoothing with a Gaussian

Parameter σ is the "scale" / "width" / "spread" of the Gaussian kernel, and controls the amount of smoothing.

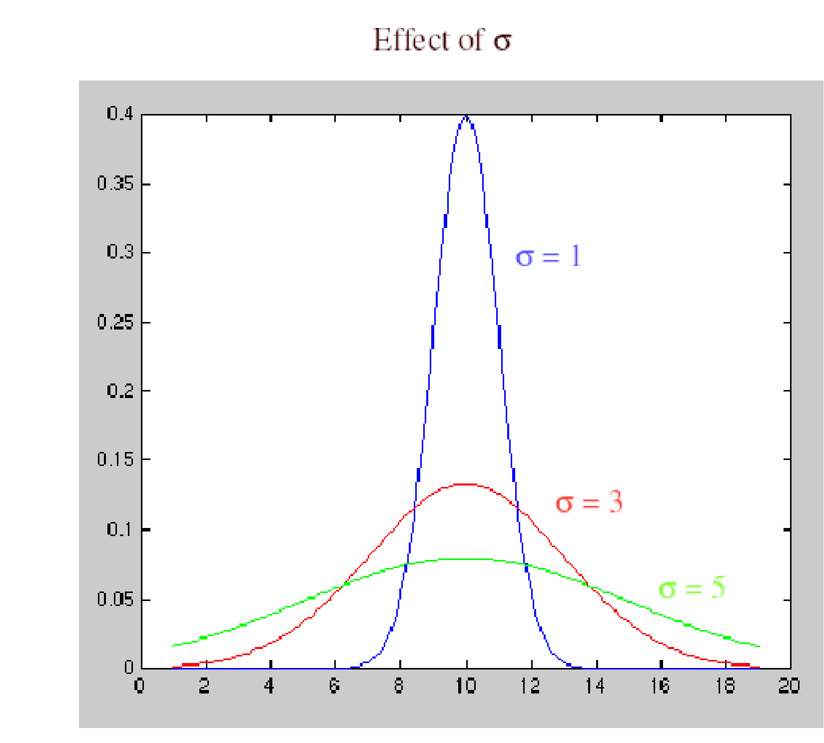


Source: K. Grauman

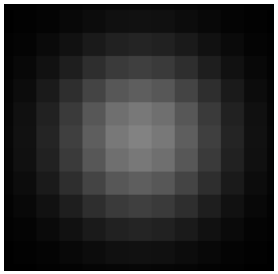
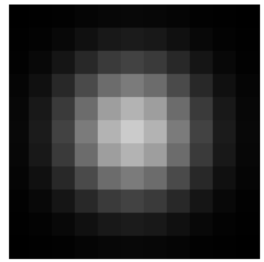
Practical matters

How big should the filter be?

- Values at edges should be near zero
- Rule of thumb for Gaussian: set kernel half-width to >= 3 σ



Combining Gaussian Filters

 $*$  $=$?

$$(f * g_s) * g_{s'} = f * (g_s * g_{s'}) = f * g_{s''}$$

$$s'' = \sqrt{s^2 + s'^2}$$

More blur than either individually (but less than $s + s'$)

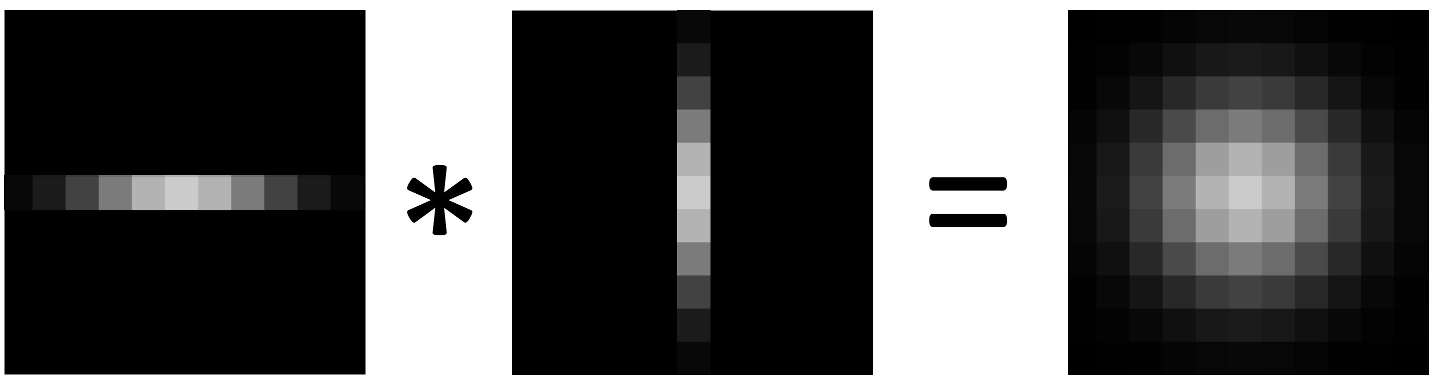
Separable Filters

Much faster!

$$G_s = G_s^x * G_s^y$$

Compute Gaussian in horizontal direction, followed by the vertical direction.

$$G_s^x(x,y) = \frac{1}{Z_y} e^{-\frac{x^2}{2s^2}} \qquad G_s^y(x,y) = \frac{1}{Z_x} e^{-\frac{y^2}{2s^2}}$$



Not all filters are separable.

Freeman and Adelson, 1991

References

- Weighted Averaging
  Gonzalez, Section 3.4 & 3.5
  Szeliski, Section 3.2