Computational Photography

- Study the basics of computation and its impact on the entire workflow of photography, from capturing, manipulating and collaborating on, and sharing photographs.

Georgia Tech College of Computing

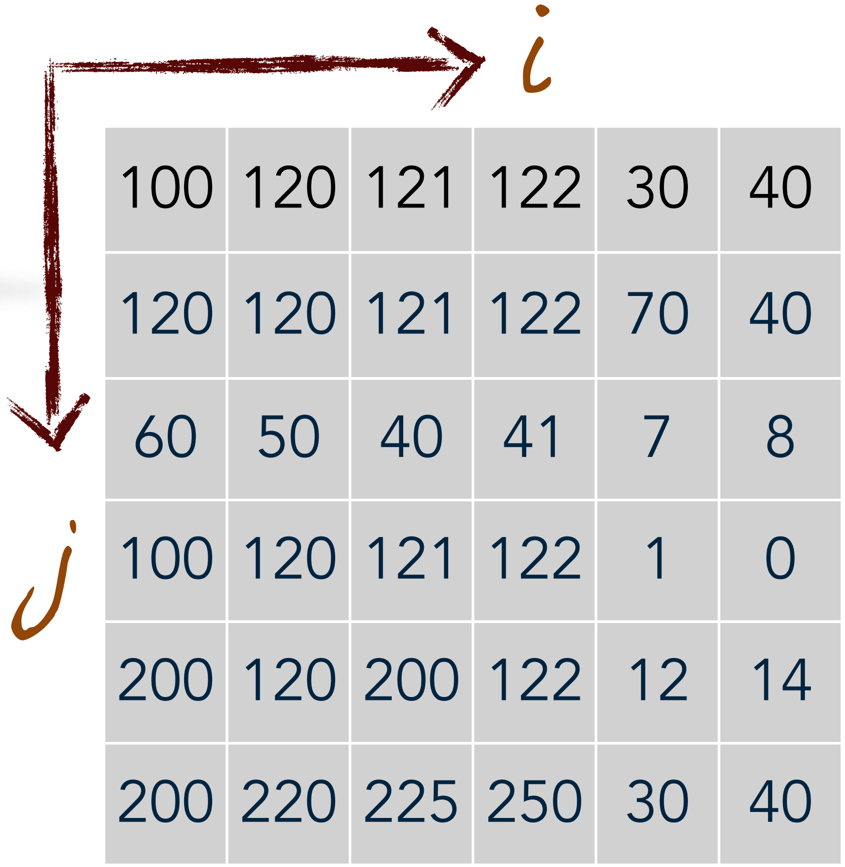
Image Processing and Filtering: Smoothing

- Point-process and Neighboring Pixels Computations on an Image for Image Smoothing

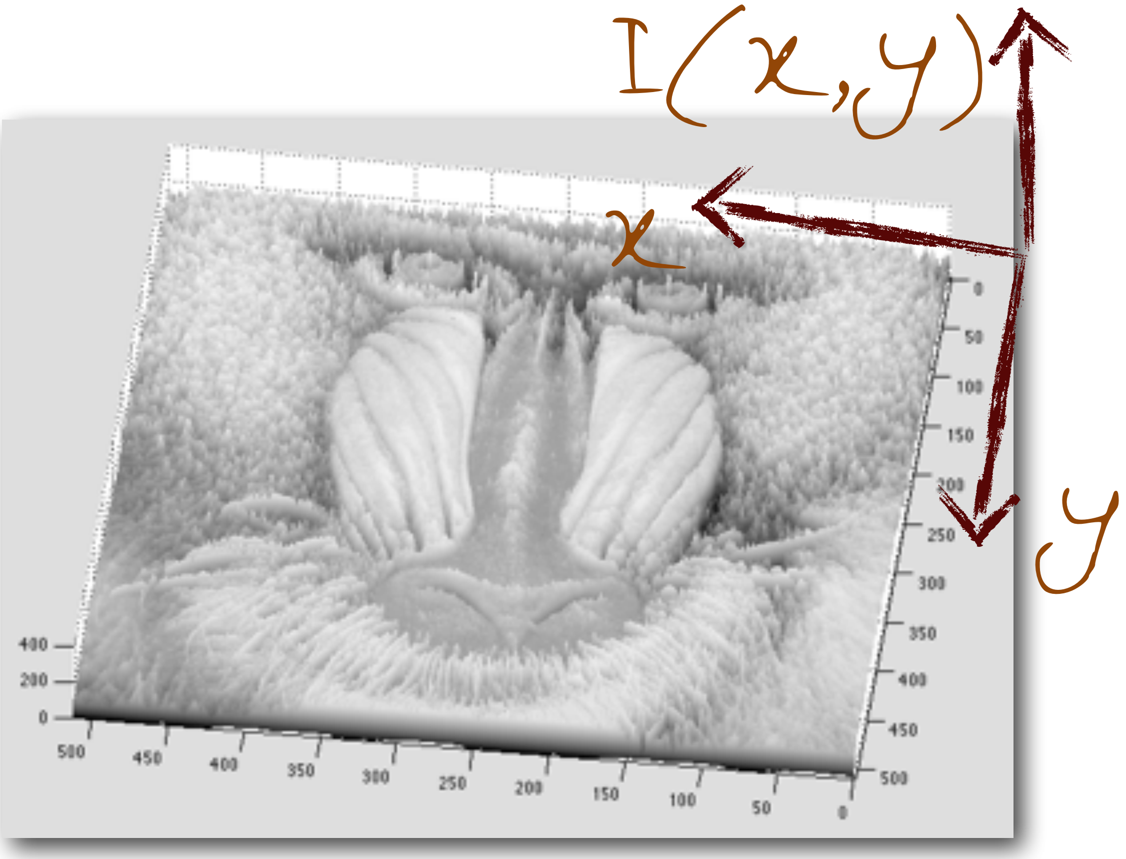
Georgia Tech College of Computing

1. Smooth an image over a neighborhood of pixels
2. median filtering as a special non-linear filtering and smoothing approach

Recall: Digital Image is a Function



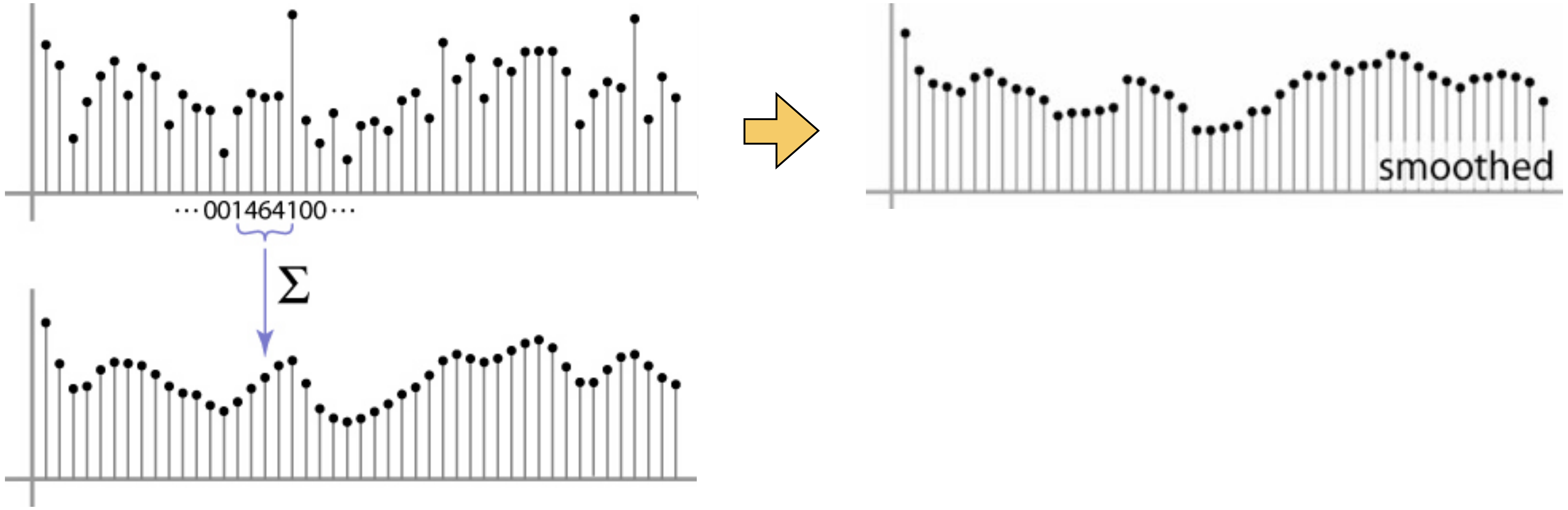
100	120	121	122	30	40
120	120	121	122	70	40
60	50	40	41	7	8
100	120	121	122	1	0
200	120	200	122	12	14
200	220	225	250	30	40



Slide adapted from Steve Seitz and Aaron Bobick

From Pixel/Point Operations to Groups of Pixels

120	122	140	142	143	10
121	120	141	144	147	151
122	121	144	146	11	152
125	121	144	145	10	153
126	121	145	147	13	12
120	122	140	142	143	10



- How to Smooth a Signal?
- Determine the Average of Neighboring Values

1. Moving Average [1 1 1 1 1] X 1/5
2. Weighted Moving Average [1 4 6 4 1] X 1/16

Slide adapted from Aaron Bobick and Steve Marschner

Smoothing Process over an Image using Averages

input (i,j)

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	0	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	90	90	90	90	0	0
0	90	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

output (i,j)

Slide adapted from Aaron Bobick and Steve marschner

Smoothing Process over an Image using Averages

input (i,j)

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	0	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	90	90	90	90	0	0
0	90	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

output (i,j)

	0	10	20	30	30	30	20	10	
	0	20	30	50					

Slide adapted from Steve Seitz and Aaron Bobick

Smoothing Process over an Image using Averages

input (i,j)

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	0	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	90	90	90	90	0	0
0	90	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

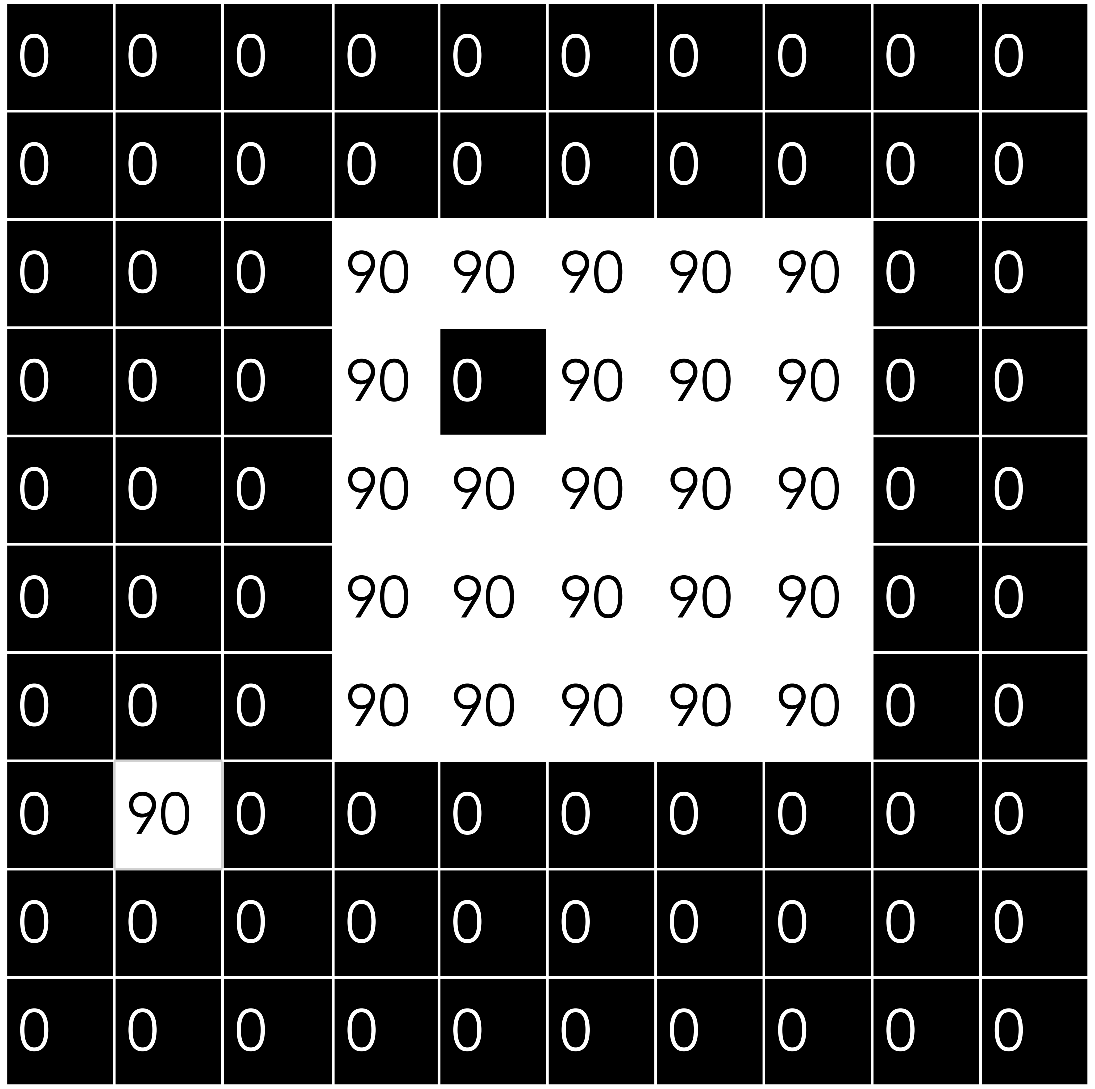
output (i,j)

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

Slide adapted from Steve Seitz and Aaron Bobick

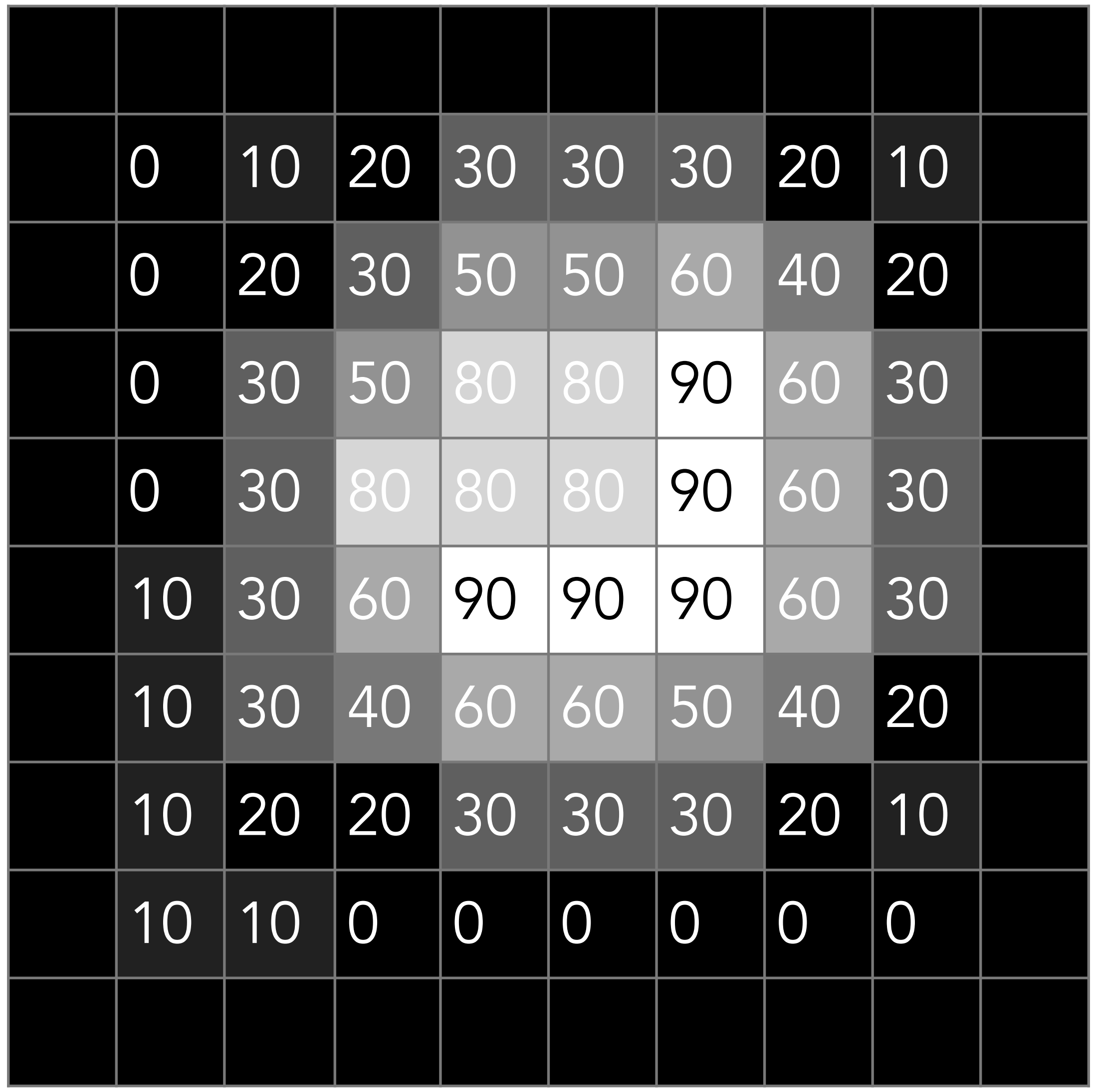
Smoothing Process over an Image using Averages

input (i,j)



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	0	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	90	90	90	90	0	0
0	90	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

output (i,j)



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

Results in Shades of Gray

Slide adapted from Aaron Bobick and Steve marschner

Smoothing Process for the Edges of an Image

20	20	0	20	10	20	10	20	10	0	10	10
20	20	0	20	10	20	10	20	10	0	10	10
30	30	0	0	0	0	0	0	0	0	10	10
20	20	0	0	90	90	90	90	90	0	20	20
20	20	0	0	90	0	90	90	90	0	10	10
10	10	0	0	90	90	90	90	90	0	20	20
10	10	0	0	90	90	90	90	90	0	10	10
10	10	0	0	90	90	90	90	90	0	10	10
20	20	90	0	0	0	0	0	0	0	10	10
20	20	0	0	0	0	0	0	0	0	10	10
20	20	0	20	10	20	10	20	10	0	10	10
20	20	0	20	10	20	10	20	10	0	10	10

Some Options

1. wrap around
2. copy edge
3. reflect across edge

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

Gray values are for demonstration only, not accurate

Smoothing Process for the Edges of an Image

20	20	0	20	10	20	10	20	10	0	10	10
20	20	0	20	10	20	10	20	10	0	10	10
30	30	0	0	0	0	0	0	0	0	10	10
20	20	0	0	90	90	90	90	90	0	20	20
20	20	0	0	90	0	90	90	90	0	10	10
10	10	0	0	90	90	90	90	90	0	20	20
10	10	0	0	90	90	90	90	90	0	10	10
10	10	0	0	90	90	90	90	90	0	10	10
20	20	90	0	0	0	0	0	0	0	10	10
20	20	0	0	0	0	0	0	0	0	10	10
20	20	0	20	10	20	10	20	10	0	10	10
20	20	0	20	10	20	10	20	10	0	10	10

Some Options

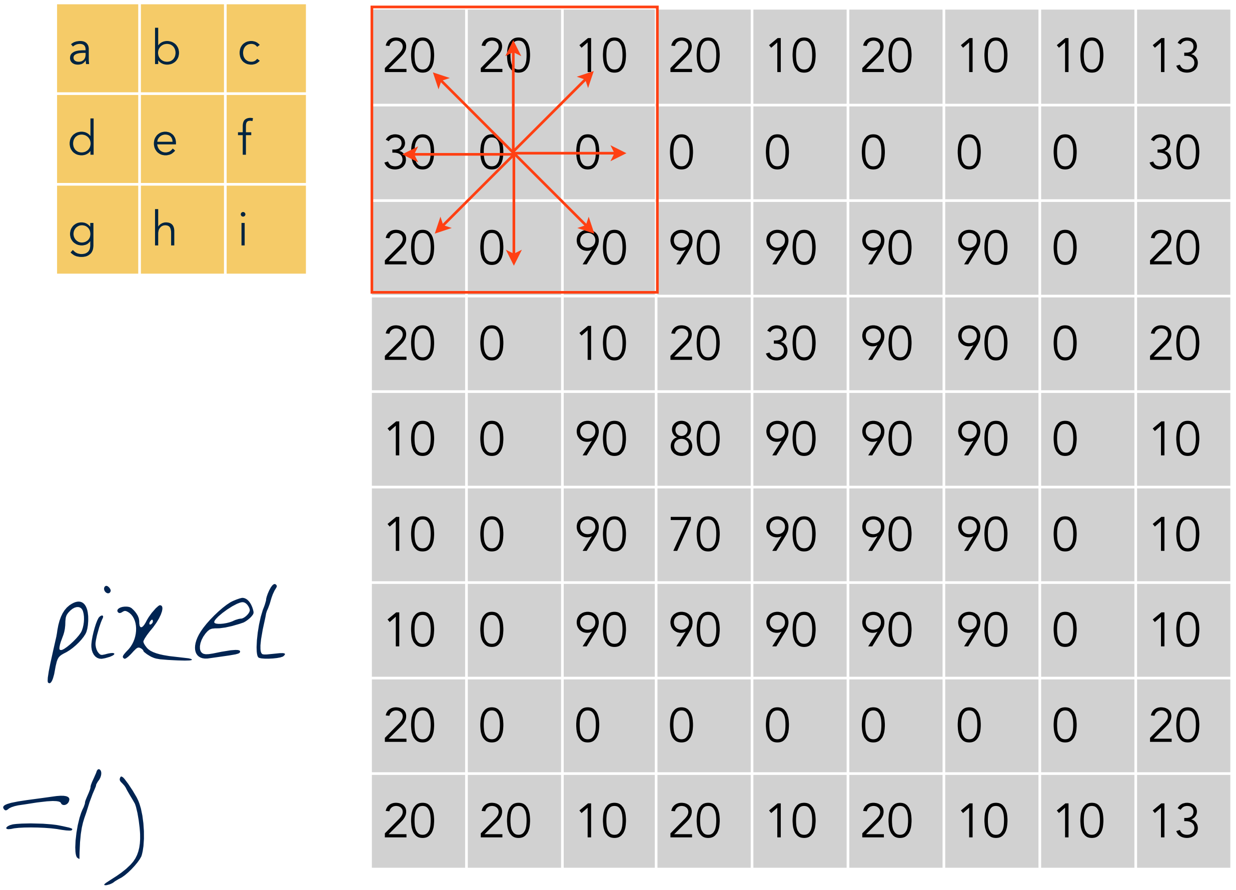
1. wrap around
2. copy edge
3. reflect across edge

Gray values are for demonstration only, not accurate

Some observations ...

- Small image "rubbed" over bigger image

kernel h(i,j)



a	b	c
d	e	f
g	h	i

20	20	10	20	10	20	10	10	13
30	0	0	0	0	0	0	0	30
20	0	90	90	90	90	90	0	20
20	0	10	20	30	90	90	0	20
10	0	90	80	90	90	90	0	10
10	0	90	70	90	90	90	0	10
10	0	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	13

- 3x3 area around each original pixel used (i.e., neighborhood size k=1)
- Window size is 2k+1, therefore our window is 3x3

Observations (continued)

Let's refer to F[i,j] as input and G[i,j] as output, h[i,j] as the kernel

$$G[3,3] = a*A + b*B + c*C + d*D + e*E + f*F + h*H + i*I$$

a	b	c
d	e	f
g	h	i

A	B	C
D	E	F
G	H	I

20	20	10	20	10	20	10	10	13
30	0	0	0	0	0	0	0	30
20	0	A	B	C	90	90	0	20
20	0	D	E	F	90	90	0	20
10	0	G	H	I	90	90	0	10
10	0	90	90	90	90	90	0	10
10	0	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	13

$$G[3,3] = \frac{1}{9}(A + B + C + D + E + F + G + H + I)$$

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

A Mathematical Representation for Smoothing

$$G[3,3] = \frac{1}{9}(A+B+C+D+E+F+G+H+I)$$

$$G[i,j] = \frac{1}{(2k+1)^2} \sum_{u=-k}^{k} \sum_{v=-k}^{k} F[i+u, j+v]$$

a	b	c
d	e	f
g	h	i

20	20	10	20	10	20	10	10	13
30	0	0	0	0	0	0	0	30
20	0	A	B	C	90	90	0	20
20	0	D	E	F	90	90	0	20
10	0	G	H	I	90	90	0	10
10	0	90	90	90	90	90	0	10
10	0	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	13

Attribute uniform weight to each pixel

Loop over all pixels in neighborhood around image pixel F[i,j]

A Mathematical Representation for Smoothing

$$G[3,3] = \frac{1}{9}(A + B + C + D + E + F + G + H + I)$$

$$G[i,j] = \frac{1}{(2k+1)^2} \sum_{u=-k}^{k} \sum_{v=-k}^{k} F[i+u, j+v]$$

a	b	c
d	e	f
g	h	i

20	20	10	20	10	20	10	10	13
30	0	0	0	0	0	0	0	30
20	0	A	B	C	90	90	0	20
20	0	D	E	F	90	90	0	20
10	0	G	H	I	90	90	0	10
10	0	90	90	90	90	90	0	10
10	0	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	13

More Generally,

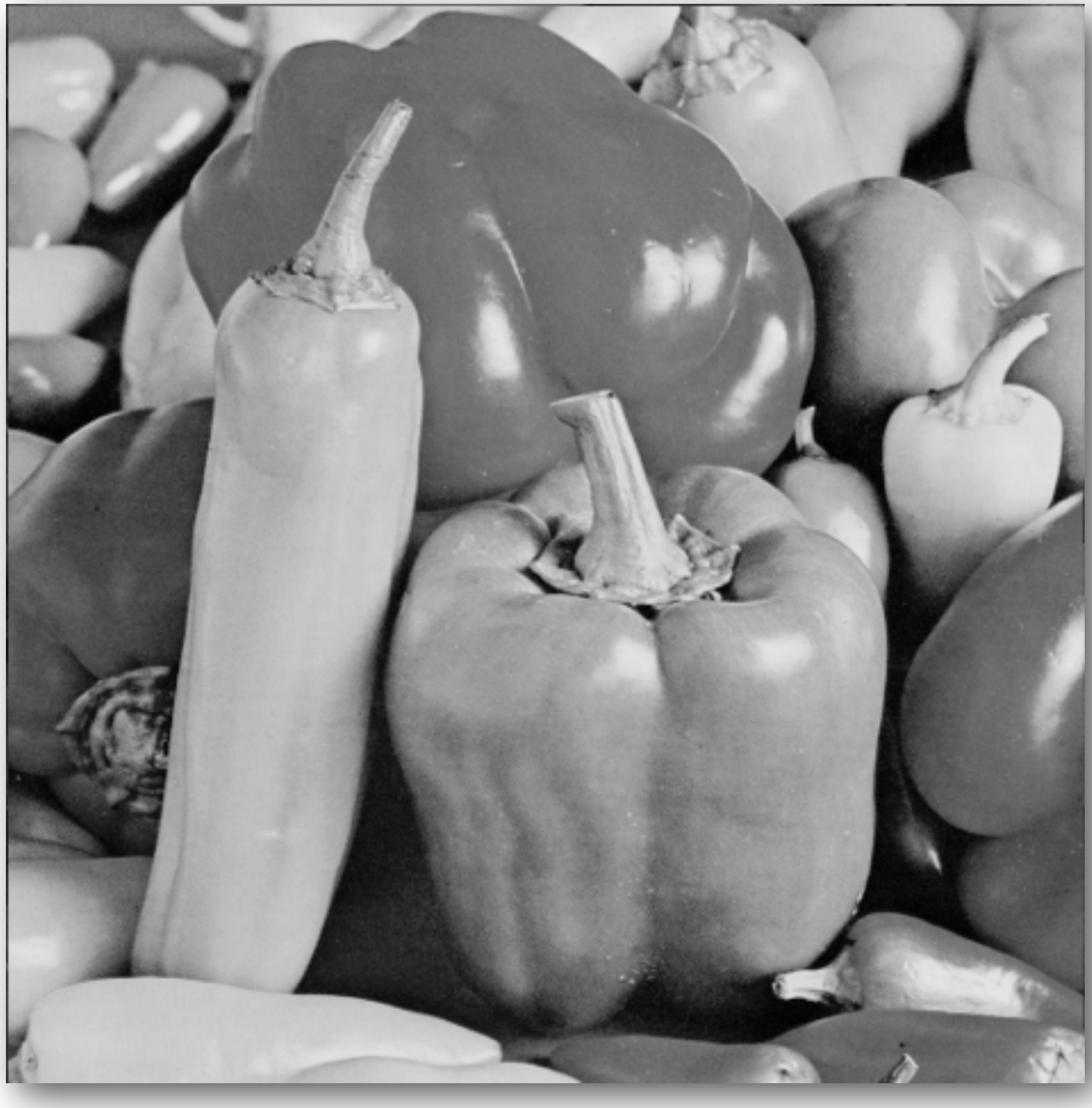
$$G[3,3] = a*A + b*B + c*C + d*D + e*E + f*F + g*G + h*H + i*I$$

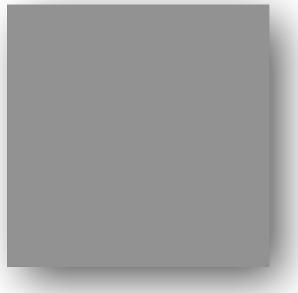
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} \boxed{h[u,v]} F[i+u, j+v]$$

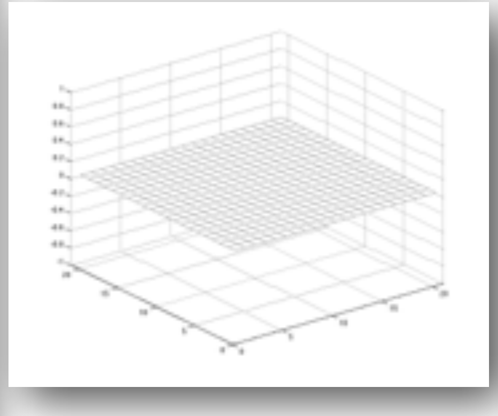
Attribute non-uniform weights

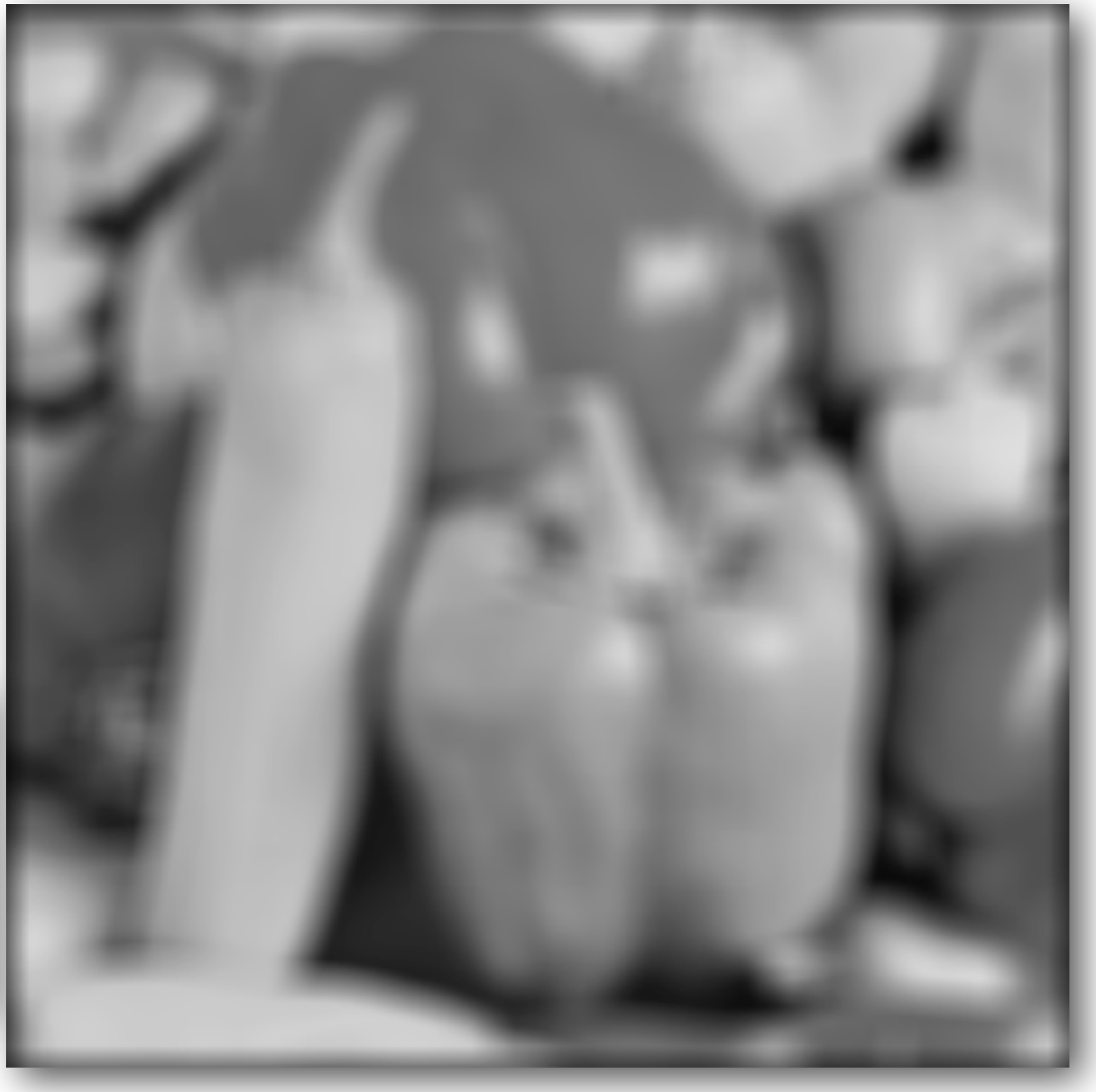
Referred to as Cross-correlation, which we will cover later

Example: Box Filter (Averaging!) for Smoothing









Box/Average Filter - 21 x 21

Special Case: Median Filtering

20	20	1	20	10	20	10	13	13
20	0	0	0	0	0	0	20	20
20	0	90	90	90	90	90	20	20
20	0	90	0	90	90	90	20	20
10	0	90	90	90	90	90	10	10
10	0	90	90	90	90	90	10	10
10	0	90	90	90	90	90	10	10
20	0	0	0	0	0	0	20	20
20	20	10	20	10	20	10	13	13

F [i,j]

$$\text{median}\left(\begin{array}{ccc}20 & 20 & 1\\ 20 & 0 & 0\\ 20 & 0 & 90\end{array}\right)$$

$$\text{median}(\,0\ 0\ 0\ 1\ 20\ 20\ 20\ 20\ 90\,) = 20$$

$$\text{average}(\,0\ 0\ 0\ 1\ 20\ 20\ 20\ 20\ 90\,) = 19$$

Apply median filter to all 3x3 regions over the whole image (as we would for averaging)

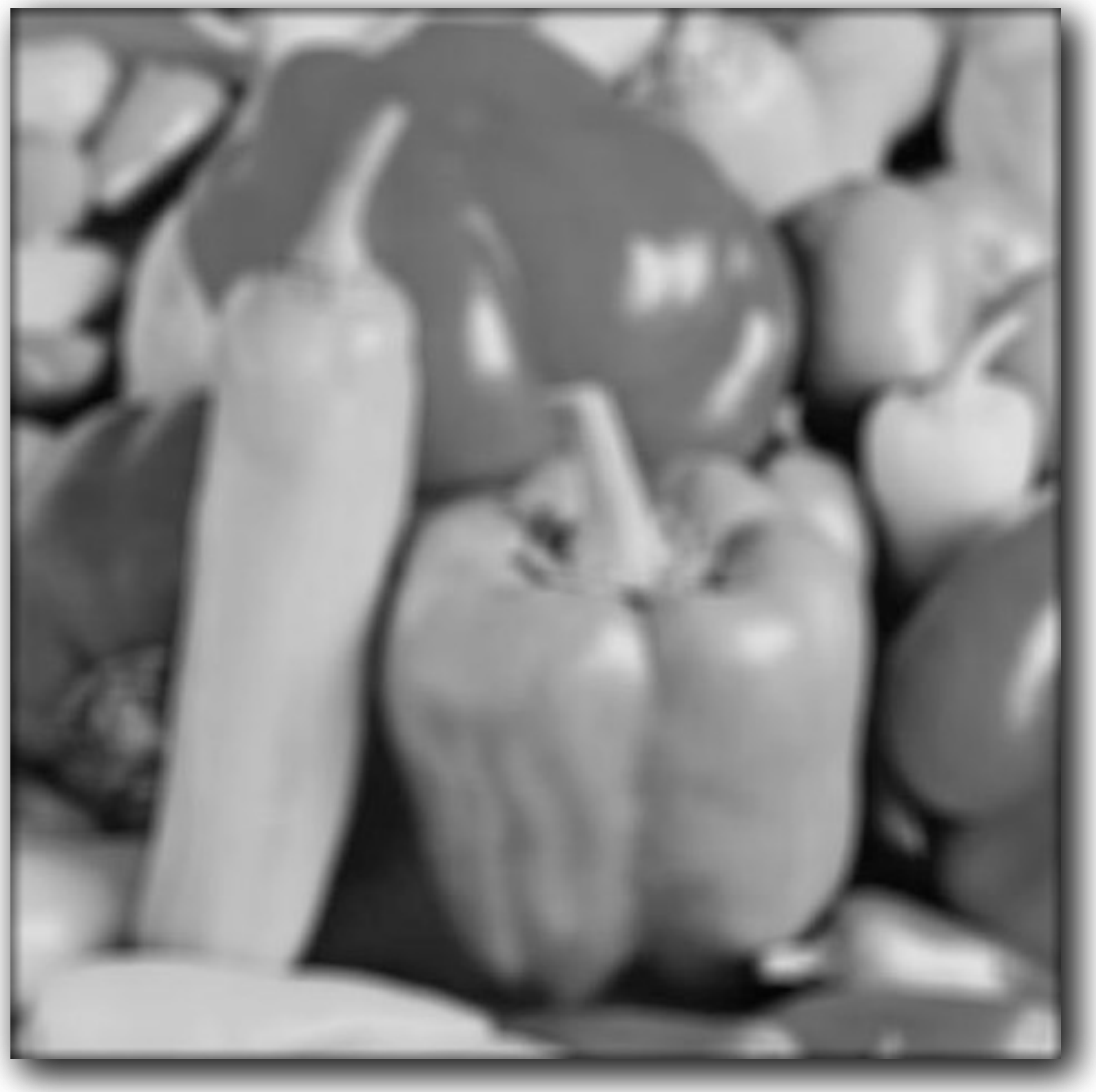
Special Case: Median Filtering

median filtering: Nonlinear operation often used in image processing

- reduces noise, but,
- preserves edges (sharp lines!)

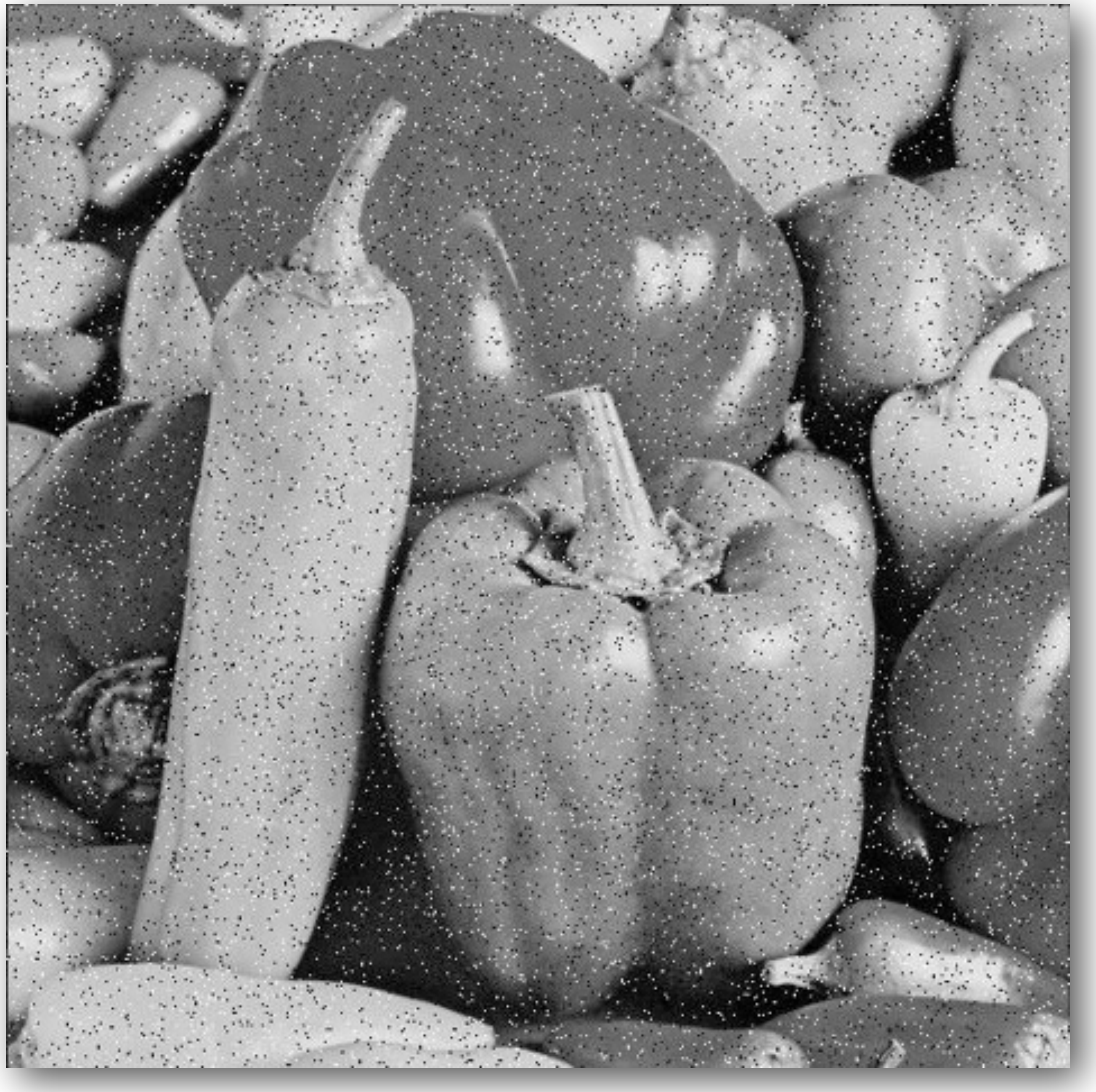
main idea: Use median of all pixels in kernel area, instead of mean

Example: Median Filtering for Smoothing Images

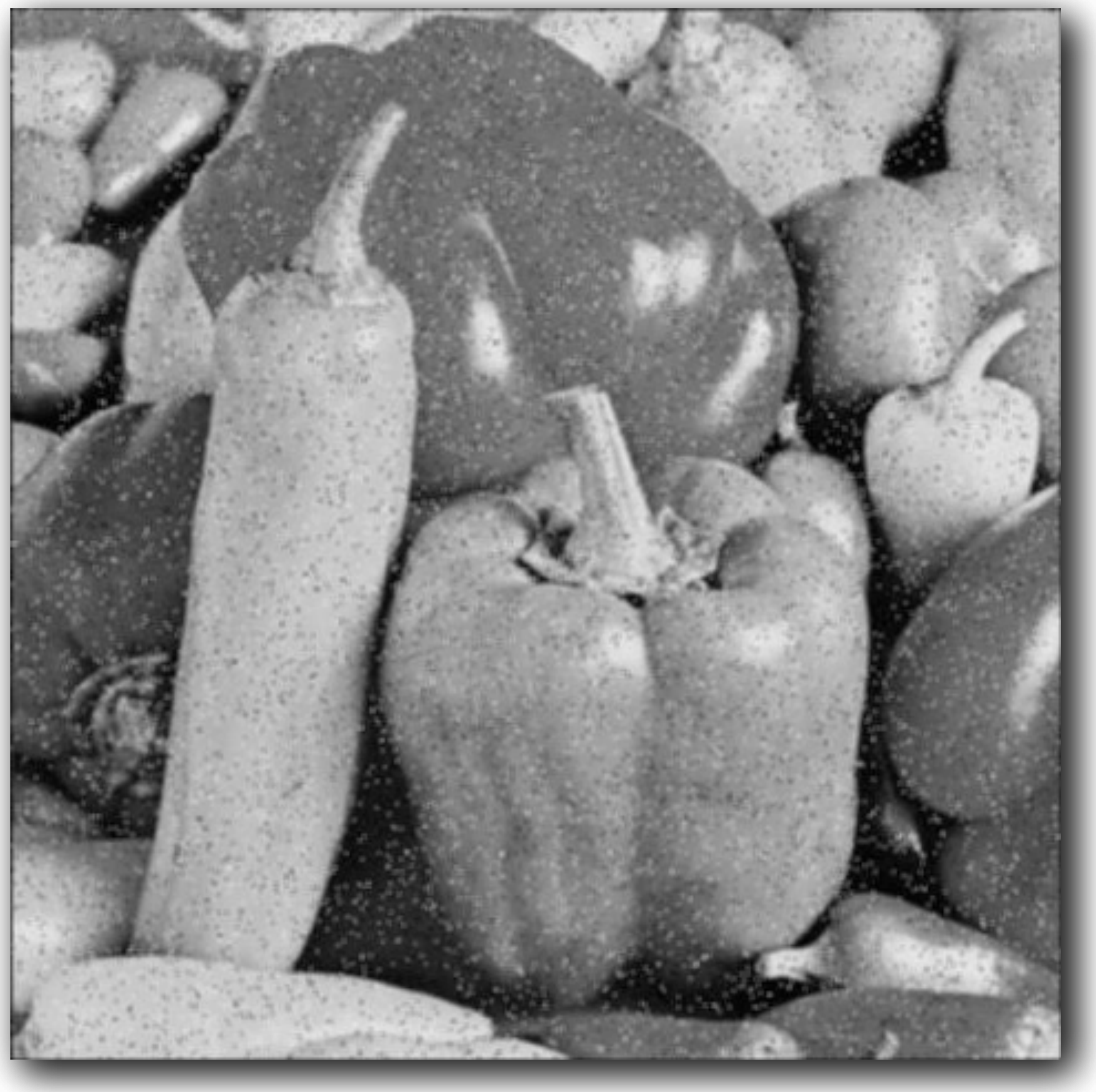


Average Filter (11 X 11)

Example: Median Filtering for Noise Removal



'Salt & Pepper' Noise



Average Filter (3 X 3)

Summary

1. Image smoothing
2. Applying a kernel to smooth an image
3. Averaging and median filtering

Next Class

Image Analysis: Cross-correlation and Convolution

Credits

- Matlab software by Mathworks Inc.
- Some Slides adapted from Aaron Bobick, Steve Seitz, Steve Marschner.
- Images used from USC's Signal and Image Processing Institute's Image Database

Computational Photography

- Study the basics of computation and its impact on the entire workflow of photography, from capturing, manipulating and collaborating on, and sharing photographs.

Georgia Tech College of Computing