

Computational Photography



Dr. Irfan Essa

Professor

School of Interactive Computing

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



Image Processing and Filtering: Smoothing



Dr. Irfan Essa

Professor

School of Interactive Computing

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

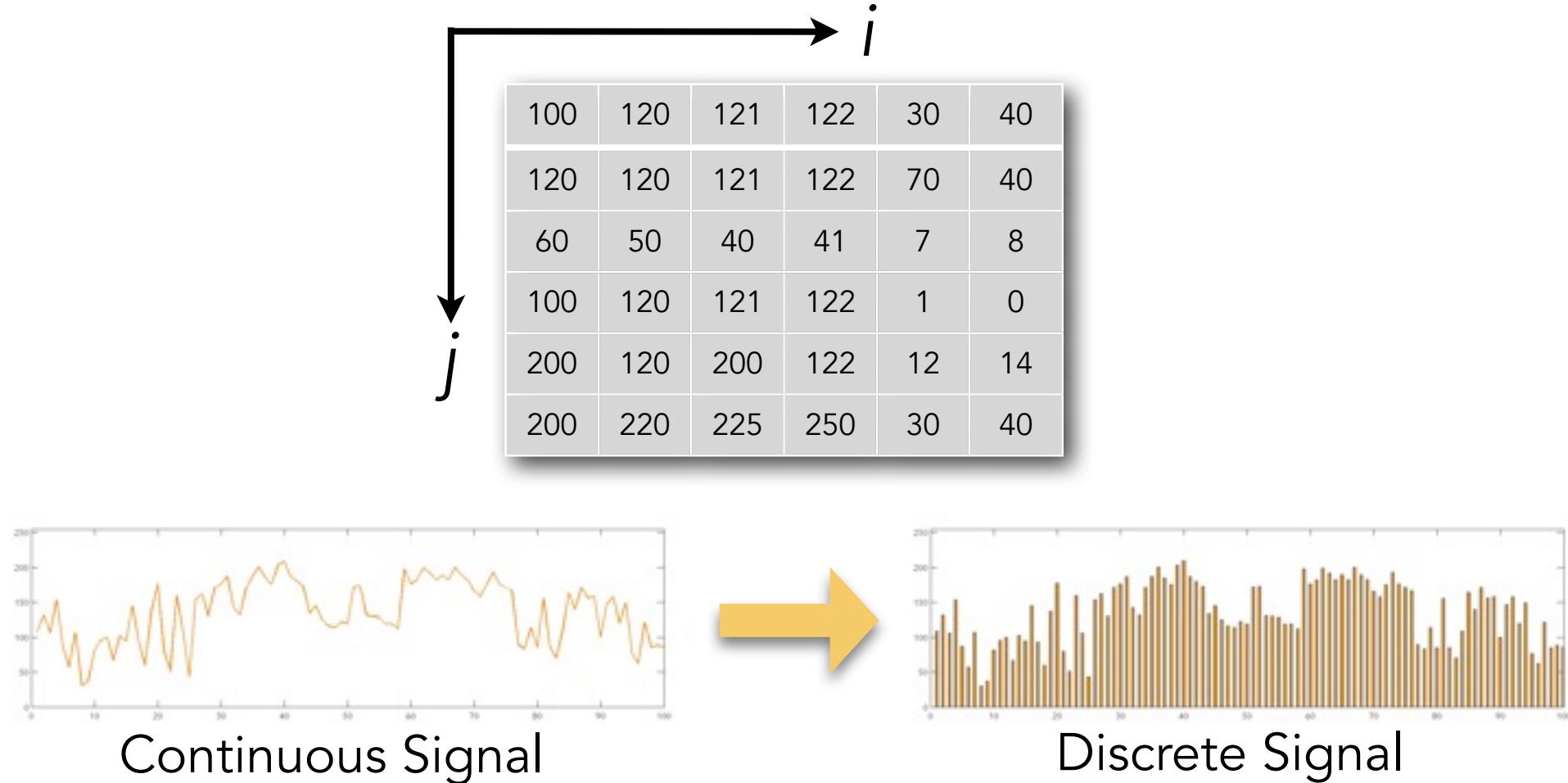
Lesson Objectives

- ★ Using the terms pixels, kernel, neighborhood of pixels, and averages, describe in your own words a process to smooth an image over a neighborhood of pixels.

- ★ Recall two (2) reasons for using median filtering as a special non-linear filtering and smoothing approach.



Review: Digital Image is a Function/Image



- ★ Typically, the functional operation requires discrete values
 - Sample the two-dimensional (2D) space on a regular grid
 - Quantize each sample (rounded to “nearest integer”)
- ★ Matrix of integer values (Range: 0-255)

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

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

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

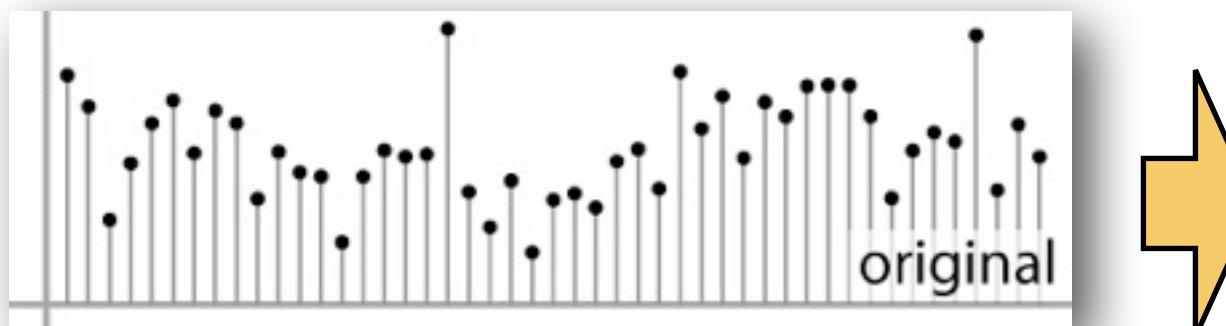
From Pixel/Point Operations to Groups of Pixels

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?

From Pixel/Point Operations to Groups of Pixels

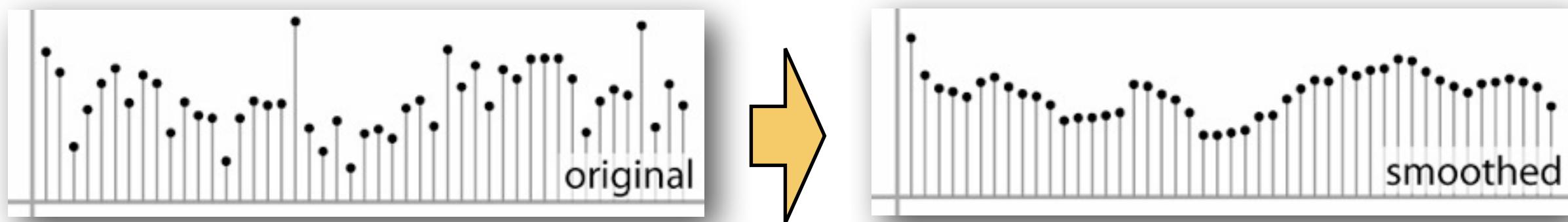
120	122	140	142	143	10
121	120	141	144	147	151
122	121	144	146	11	152
125	121	144	145	10	153
126	121	145	147	13	12
120	122	140	142	143	10



★ How to Smooth a Signal?

From Pixel/Point Operations to Groups of Pixels

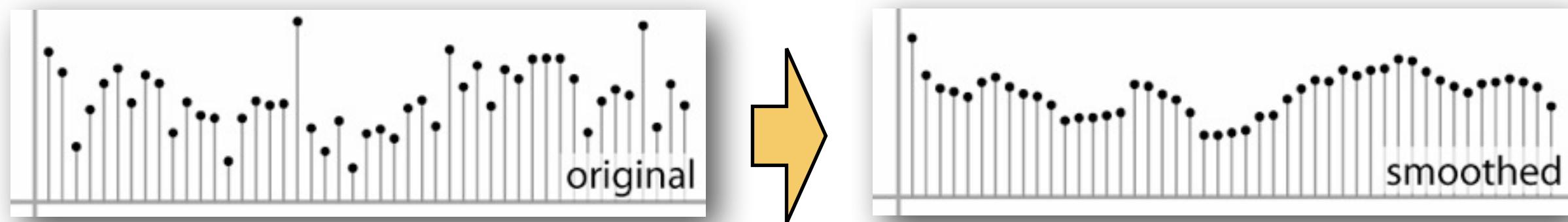
120	122	140	142	143	10
121	120	141	144	147	151
122	121	144	146	11	152
125	121	144	145	10	153
126	121	145	147	13	12
120	122	140	142	143	10



★ How to Smooth a Signal?

From Pixel/Point Operations to Groups of Pixels

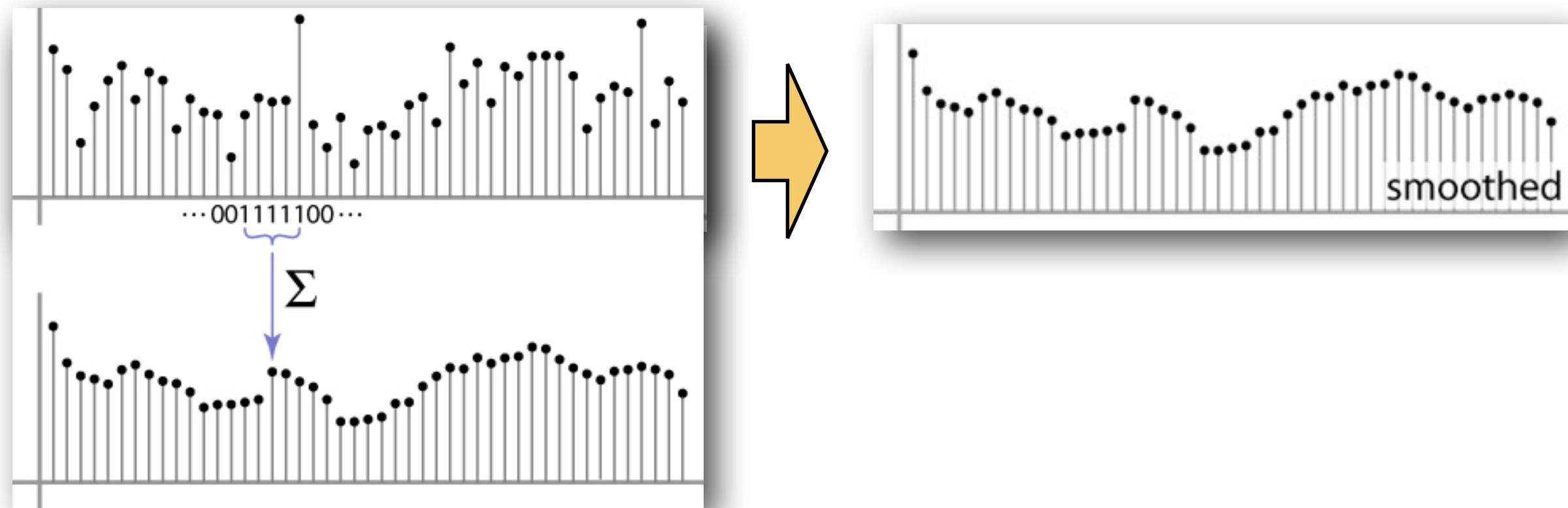
120	122	140	142	143	10
121	120	141	144	147	151
122	121	144	146	11	152
125	121	144	145	10	153
126	121	145	147	13	12
120	122	140	142	143	10



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

From Pixel/Point Operations to Groups of Pixels

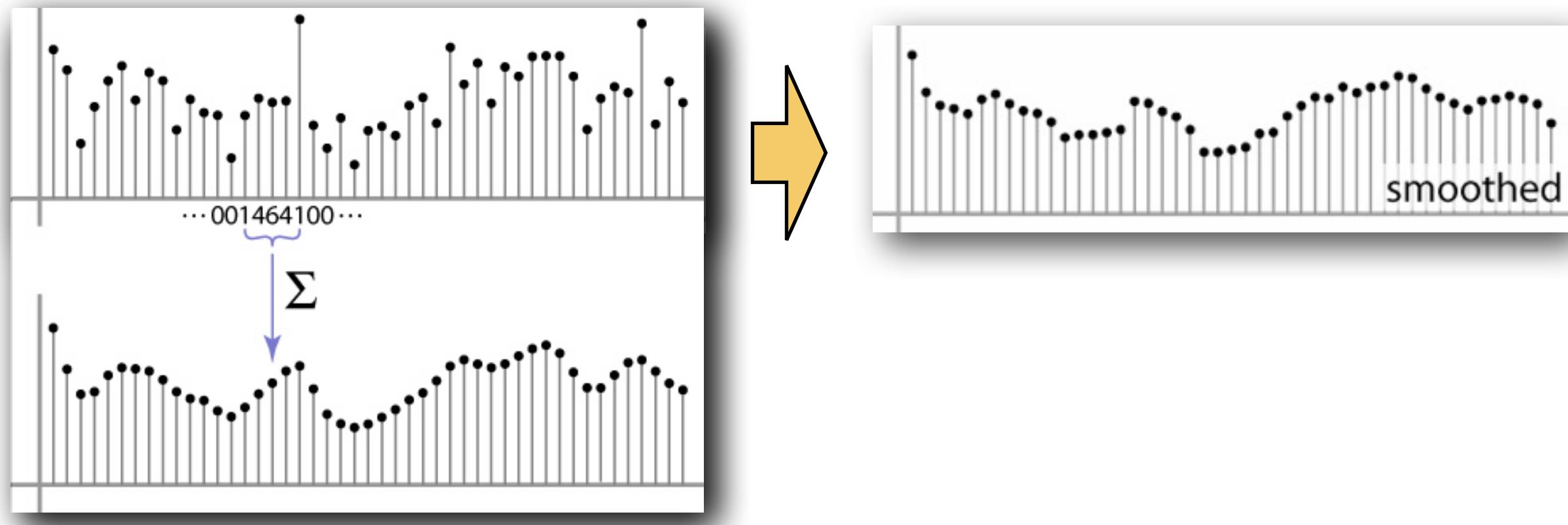
120	122	140	142	143	10
121	120	141	144	147	151
122	121	144	146	11	152
125	121	144	145	10	153
126	121	145	147	13	12
120	122	140	142	143	10



- ★ How to Smooth a Signal?
- ★ Determine the Average of Neighboring Values
 - Eg. 1: Moving Average $[1 \ 1 \ 1 \ 1 \ 1] * 1/5$

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
 - Eg. 1: Moving Average $[1 \ 1 \ 1 \ 1 \ 1] * 1/5$
 - Eg. 2: Weighted Moving Average $[1 \ 4 \ 6 \ 4 \ 1] * 1/16$

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

output (i,j)

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

output (i,j)

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

output (i,j)

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

output (i,j)

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

output (i,j)

0										

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

output (i,j)

0										

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

output (i,j)

0	10									

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

output (i,j)

0	10									

Smoothing Process over an Image using Averages

input (i, j)

output (i, j)

Slide Adapted from Aaron Bobick and Steve Seitz

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

output (i,j)

	0	10	20							

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

output (i,j)

	0	10	20	30						

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

output (i,j)

	0	10	20	30						

Smoothing Process over an Image using Averages

input (i, j)

output (i, j)

Slide Adapted from Aaron Bobick and Steve Seitz

Smoothing Process over an Image using Averages

input (i,j)

output (i,j)

Slide Adapted from Aaron Bobick and Steve Seitz

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

output (i,j)

0	10	20	30	30	30					

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

output (i,j)

0	10	20	30	30	30	20				

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

output (i,j)

0	10	20	30	30	30	20						

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

output (i,j)

0	10	20	30	30	30	30	20	10				

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

output (i,j)

0	10	20	30	30	30	30	20	10		

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

output (i,j)

0	10	20	30	30	30	20	10			
0										

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

output (i,j)

0	10	20	30	30	30	20	10			
0										

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

output (i,j)

0	10	20	30	30	30	20	10			
0	20									

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

output (i,j)

0	10	20	30	30	30	20	10			
0	20									

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

output (i,j)

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

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

output (i,j)

0	10	20	30	30	30	20	10			
0	20	30								

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

output (i,j)

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

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	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	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						

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	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	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						

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

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

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

Smoothing Process over an Image using Averages

input (i,j)

Results in Shades of Gray

output (i,j)

Smoothing Process over an Image using Averages

<i>input (i,j)</i>										
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	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	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	0	0	0

Results in Shades of Gray

output (i,j)

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

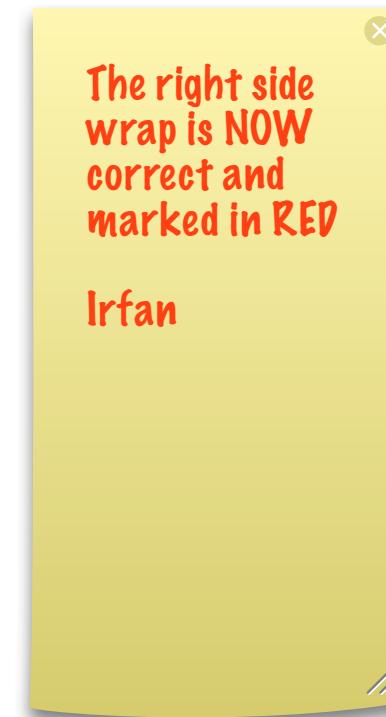
Results in Shades of Gray

output (i,j)

0	10	20	30	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	0		

Smoothing Process for the Edges of an Image

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



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



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

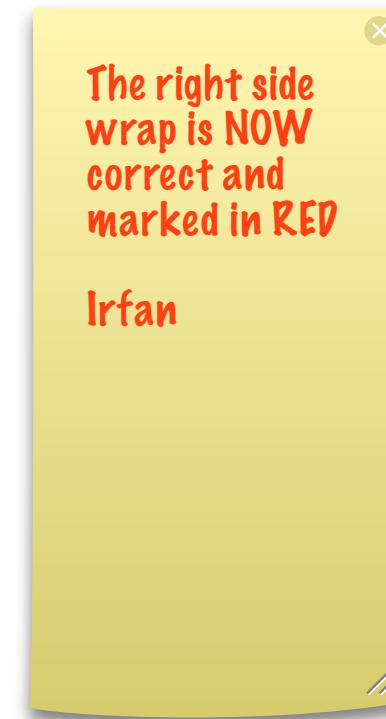


0	10	20	30	30	30	30	20	10	
0	20	30	50	50	60	40	20		
0	30	50	80	80	90	60	30		
0	30	80	80	80	90	60	30		
10	30	60	90	90	90	60	30		
10	30	40	60	60	50	40	20		
10	20	20	30	30	30	20	10		
10	10	0	0	0	0	0	0	0	

Gray values are for demonstration only, not accurate

Smoothing Process for the Edges of an Image

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



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



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



0	10	20	30	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	0			

Gray values are for demonstration only, not accurate

Smoothing Process for the Edges of an Image

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



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

The right side
wrap is NOW
correct and
marked in RED

Irfan

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



	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	13
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	13



	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	13
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	13



0	10	20	30	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	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	13
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	13



Some Options

- wrap around
- copy edge
- reflect across edge

0	10	20	30	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	0			

Gray values are for demonstration only, not accurate

Notes:

a	b	c
d	e	f
g	h	i

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

Notes:

- ★ A small 3x3 neighborhood of pixels was used to operate on the Entire Image

a	b	c
d	e	f
g	h	i

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

Notes:

- ★ A small 3x3 neighborhood of pixels was used to operate on the Entire Image
 - A small image was 'rubbed' over the bigger image.

a	b	c
d	e	f
g	h	i

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

Notes:

- ★ A small 3x3 neighborhood of pixels was used to operate on the Entire Image
 - A small image was 'rubbed' over the bigger image.

a	b	c
d	e	f
g	h	i

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

Notes:

- ★ A small 3x3 neighborhood of pixels was used to operate on the Entire Image
 - A small image was 'rubbed' over the bigger image.
 - The small image is identified as *kernel h(i,j)*.

a	b	c
d	e	f
g	h	i

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

Notes:

- ★ A small 3x3 neighborhood of pixels was used to operate on the Entire Image
 - A small image was 'rubbed' over the bigger image.
 - The small image is identified as *kernel h(i,j)*.
 - A 3x3 pixel image was moved around the original pixel (i.e., our neighborhood is size 1)

a	b	c
d	e	f
g	h	i

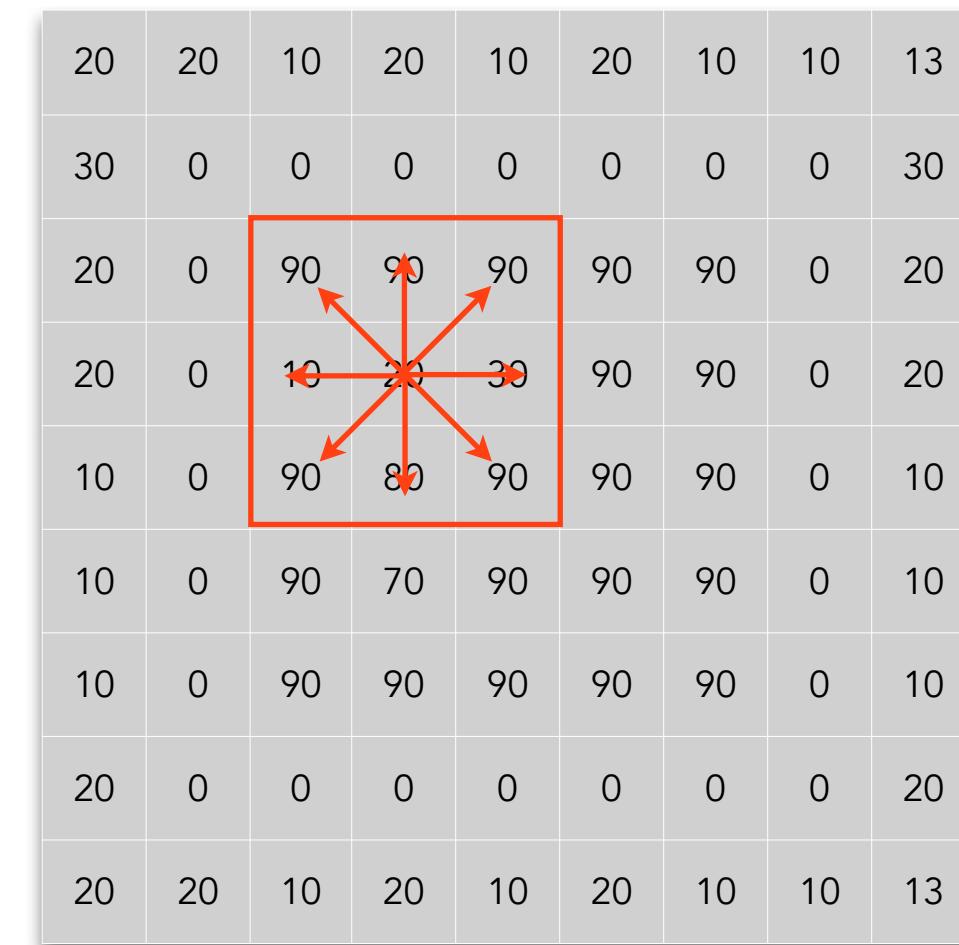
20	20	10	20	10	20	10	10	13
30	0	0	0	0	0	0	0	30
20	0	90	90	90	90	90	90	0
20	0	10	20	30	90	90	90	0
10	0	90	80	90	90	90	90	0
10	0	90	70	90	90	90	90	0
10	0	90	90	90	90	90	90	0
20	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	13

Notes:

- ★ A small 3×3 neighborhood of pixels was used to operate on the Entire Image
 - A small image was 'rubbed' over the bigger image.
 - The small image is identified as *kernel $h(i,j)$* .
 - A 3×3 pixel image was moved around the original pixel (i.e., our neighborhood is size 1)

a	b	c
d	e	f
g	h	i

20	20	10	20	10	20	20	10	10	13
30	0	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	0	20
20	20	10	20	10	20	20	10	10	13

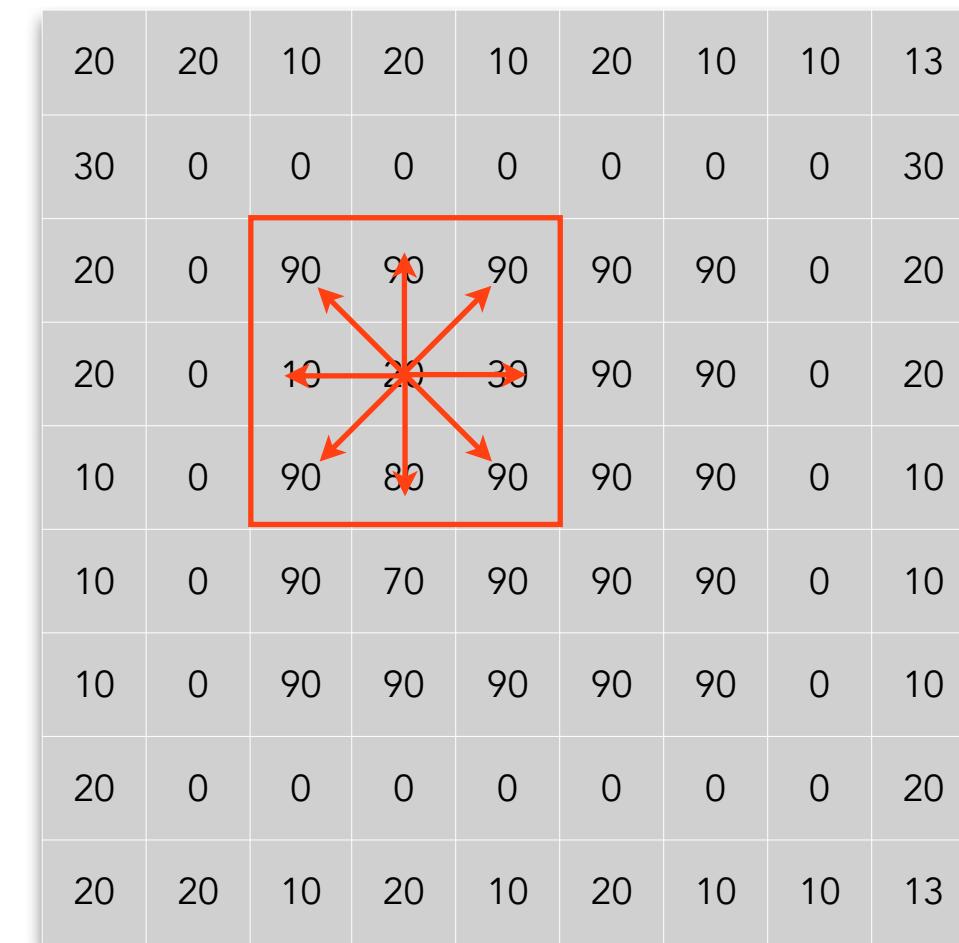


Notes:

- ★ A small 3×3 neighborhood of pixels was used to operate on the Entire Image
 - A small image was 'rubbed' over the bigger image.
 - The small image is identified as *kernel* $h(i,j)$.
 - A 3×3 pixel image was moved around the original pixel (i.e., our neighborhood is size 1)
 - The window size is $2k+1$, where $k=1$, therefore our window is 3×3

a	b	c
d	e	f
g	h	i

20	20	10	20	10	20	20	10	10	13
30	0	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	0	20
20	20	10	20	10	20	20	10	10	13



Notes: (continued)

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

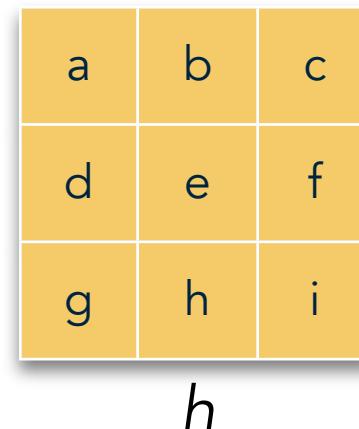
Notes: (continued)

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

20	20	10	20	10	20	10	10	10	13
30	0	0	0	0	0	0	0	0	30
20	0	90	90	90	90	90	90	0	20
20	0	90	0	90	90	90	90	0	20
10	0	90	90	90	90	90	90	0	10
10	0	90	90	90	90	90	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	10	13

Notes: (continued)

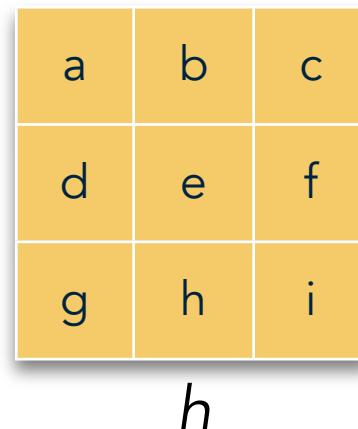
Let's refer to $F[i,j]$ as input and $G[i,j]$ as output,
 $h[i,j]$ as the kernel



20	20	10	20	10	20	10	10	10	13
30	0	0	0	0	0	0	0	0	30
20	0	90	90	90	90	90	90	0	20
20	0	90	0	90	90	90	90	0	20
10	0	90	90	90	90	90	90	0	10
10	0	90	90	90	90	90	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	10	13

Notes: (continued)

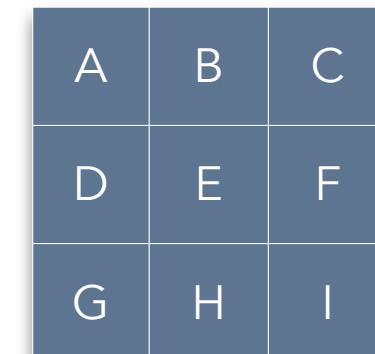
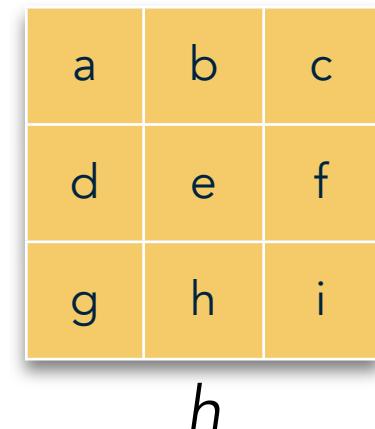
Let's refer to $F[i,j]$ as input and $G[i,j]$ as output,
 $h[i,j]$ as the kernel



20	20	10	20	10	20	10	10	10	13
30	0	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	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	10	13

Notes: (continued)

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

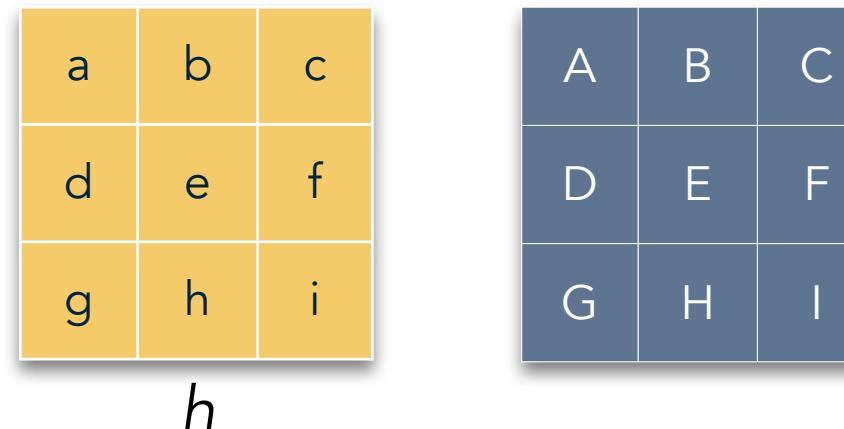


20	20	10	20	10	20	10	10	10	13
30	0	0	0	0	0	0	0	0	30
20	0	A	B	C	90	90	0	20	
20	0	D	E	F	90	90	0	20	
10	0	G	H	I	90	90	0	10	
10	0	90	90	90	90	90	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	10	13

Notes: (continued)

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

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

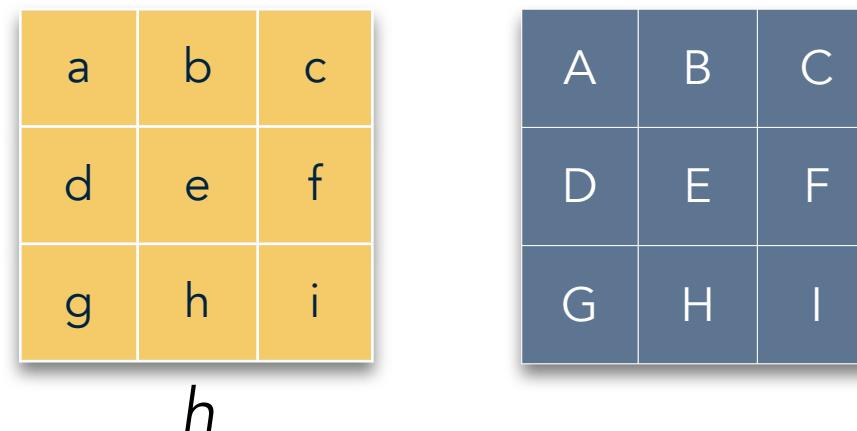


20	20	10	20	10	20	10	10	10	13
30	0	0	0	0	0	0	0	0	30
20	0	A	B	C	90	90	0	20	
20	0	D	E	F	90	90	0	20	
10	0	G	H	I	90	90	0	10	
10	0	90	90	90	90	90	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	10	13

Notes: (continued)

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

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



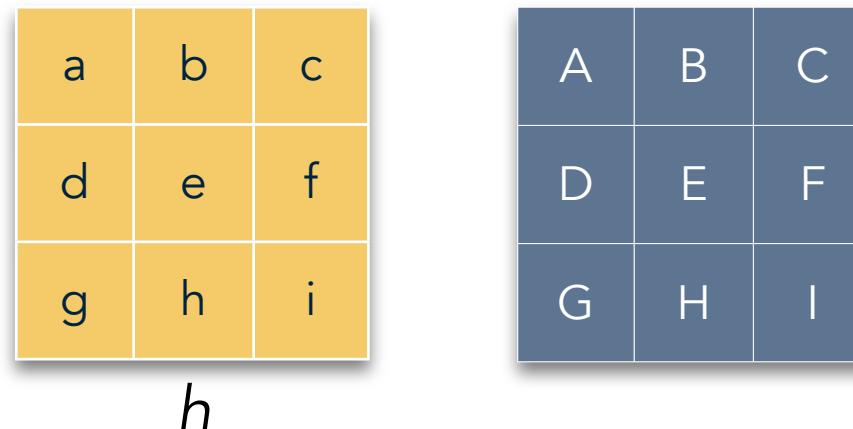
20	20	10	20	10	20	10	10	10	13
30	0	0	0	0	0	0	0	0	30
20	0	A	B	C	90	90	0	20	
20	0	D	E	F	90	90	0	20	
10	0	G	H	I	90	90	0	10	
10	0	90	90	90	90	90	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	10	13

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

Notes: (continued)

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

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



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

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

20	20	10	20	10	20	20	10	10	13
30	0	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	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	10	10	10	13

A Mathematical Representation for Smoothing

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

a	b	c
d	e	f
g	h	i

20	20	10	20	10	20	20	10	10	13
30	0	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	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	20	10	10	13

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	20	10	10	13
30	0	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	90	0	10
10	0	90	90	90	90	90	90	0	10
20	0	0	0	0	0	0	0	0	20
20	20	10	20	10	20	20	10	10	13

A Mathematical Representation for Smoothing

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

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

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

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	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

A Mathematical Representation for Smoothing

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

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

Attribute uniform weight to each pixel

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

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	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

A Mathematical Representation for Smoothing

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

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

Attribute uniform weight to each pixel

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

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	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,

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

A Mathematical Representation for Smoothing

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

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

Attribute uniform weight to each pixel

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

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	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,

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

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

A Mathematical Representation for Smoothing

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

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

Attribute uniform weight to each pixel

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

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	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,

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

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

Attribute non-uniform weights

A Mathematical Representation for Smoothing

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

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

Attribute uniform weight to each pixel

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

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	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,

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

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

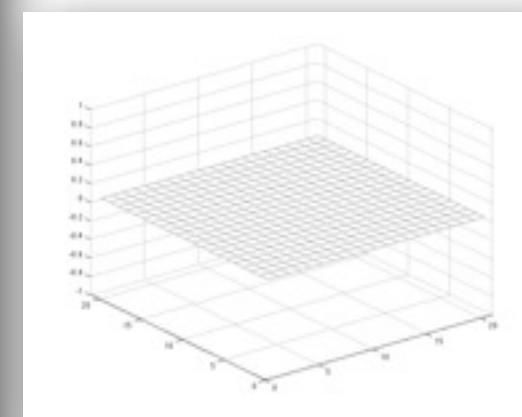
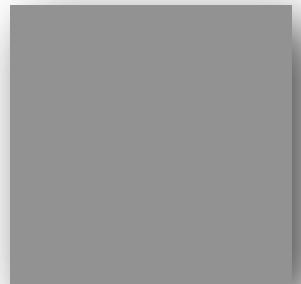
Attribute non-uniform weights

Referred to as Cross-correlation, which we will cover in next lecture.

Example: Box Filter (Averaging!) for Smoothing



Box/Average
Filter
 21×21



Special Case: Median Filtering

In the lecture, the average was WRONG, as the region I choose had interestingly both the median and average as 20. I have replaced the element from 10 to 0 and now it is correct. SORRY for the error.

Irfan

20	20	0	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

$$F[i,j]$$

- ★ Median filtering is a nonlinear operation often used in image processing to reduce noise and preserve edges (sharp lines!).
- ★ As opposed to take the average of all the pixels in the kernel, we take the median.

Special Case: Median Filtering

In the lecture, the average was WRONG, as the region I choose had interestingly both the median and average as 20. I have replaced the element from 10 to 0 and now it is correct. SORRY for the error.

Irfan

20	20	0	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

median(

20	20	0
20	0	0
20	0	90

)

$F[i,j]$

- ★ Median filtering is a nonlinear operation often used in image processing to reduce noise and preserve edges (sharp lines!).
- ★ As opposed to take the average of all the pixels in the kernel, we take the median.

Special Case: Median Filtering

In the lecture, the average was WRONG, as the region I choose had interestingly both the median and average as 20. I have replaced the element from 10 to 0 and now it is correct. SORRY for the error.

Irfan

20	20	0	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

$$\text{median}(\begin{array}{|c|c|c|}\hline 20 & 20 & 0 \\ \hline 20 & 0 & 0 \\ \hline 20 & 0 & 90 \\ \hline \end{array}) \quad \text{median}(\begin{array}{|c|c|c|c|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 20 & 20 & 20 & 20 & 90 \\ \hline \end{array}) = 20$$

$$F[i,j]$$

- ★ Median filtering is a nonlinear operation often used in image processing to reduce noise and preserve edges (sharp lines!).
- ★ As opposed to take the average of all the pixels in the kernel, we take the median.

Special Case: Median Filtering

In the lecture, the average was WRONG, as the region I choose had interestingly both the median and average as 20. I have replaced the element from 10 to 0 and now it is correct. SORRY for the error.

Irfan

20	20	0	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

$$\text{median}(\begin{array}{|c|c|c|} \hline 20 & 20 & 0 \\ \hline 20 & 0 & 0 \\ \hline 20 & 0 & 90 \\ \hline \end{array}) = 20$$

$$\text{average}(\begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 20 & 20 & 20 & 20 & 90 \\ \hline \end{array}) = 18.8$$

$F[i,j]$

- ★ Median filtering is a nonlinear operation often used in image processing to reduce noise and preserve edges (sharp lines!).
- ★ As opposed to take the average of all the pixels in the kernel, we take the median.

In the lecture, the average was WRONG, as the region I choose had interestingly both the median and average as 20. I have replaced the element from 10 to 0 and now it is correct. SORRY for the error.

Irfan

Special Case: Median Filtering

20	20	0	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

median(

20	20	0
20	0	0
20	0	90

) median(

0	0	0	0	20	20	20	20	90
---	---	---	---	----	----	----	----	----

) = 20

average(0 0 0 0 20 20 20 20 90) = 18.8

$F[i,j]$

Apply median filter to all 3x3 regions over the whole image (as in averaging case).

- ★ Median filtering is a nonlinear operation often used in image processing to reduce noise and preserve edges (sharp lines!).
- ★ As opposed to take the average of all the pixels in the kernel, we take the median.

Example: Median Filtering for Smoothing



Median
Filter
 11×11



Example: Median Filtering for Smoothing



Median
Filter
 11×11

Average
Filter
 11×11



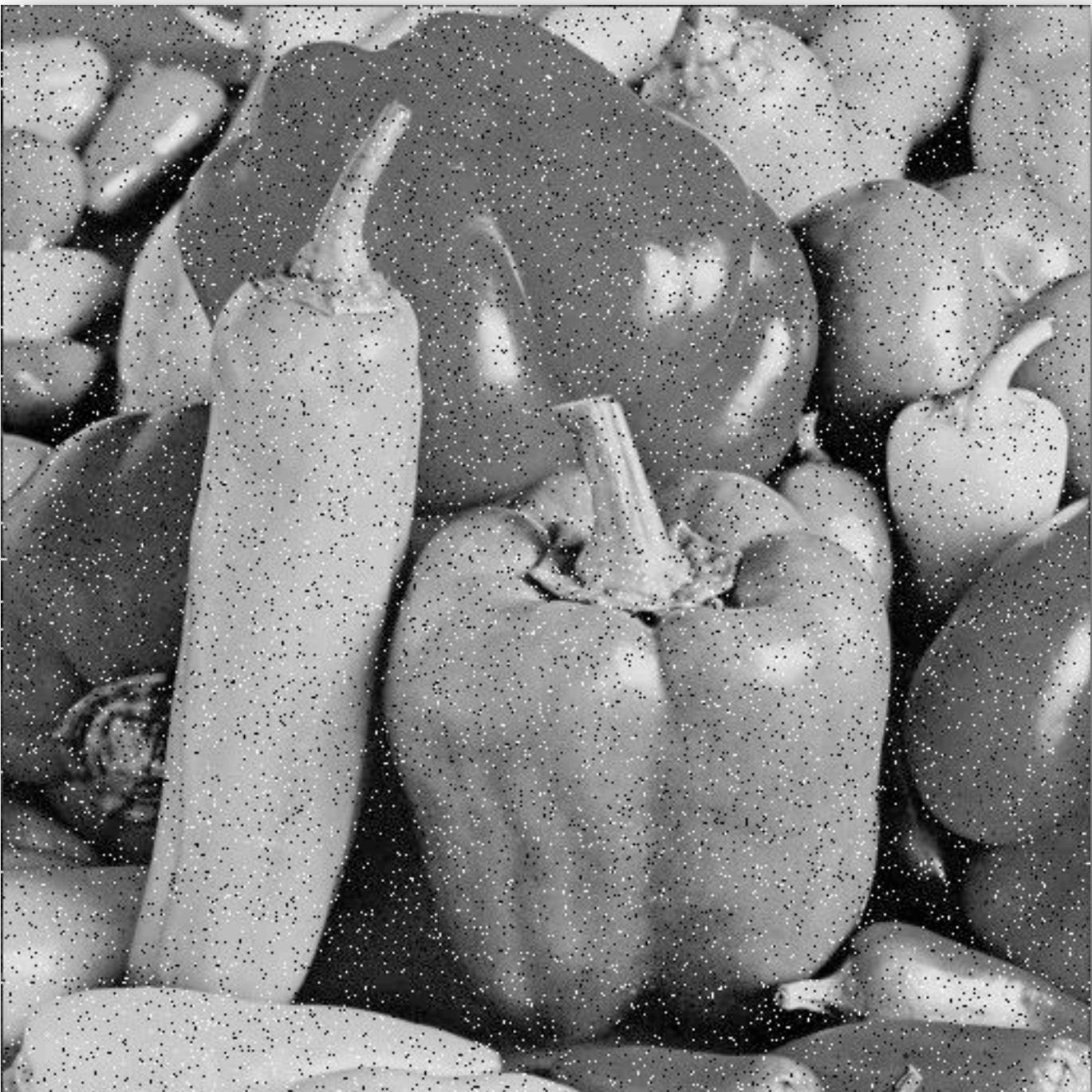
Example: Median Filtering for Smoothing



Median
Filter
 11×11



Example: Median Filtering for Noise Removal

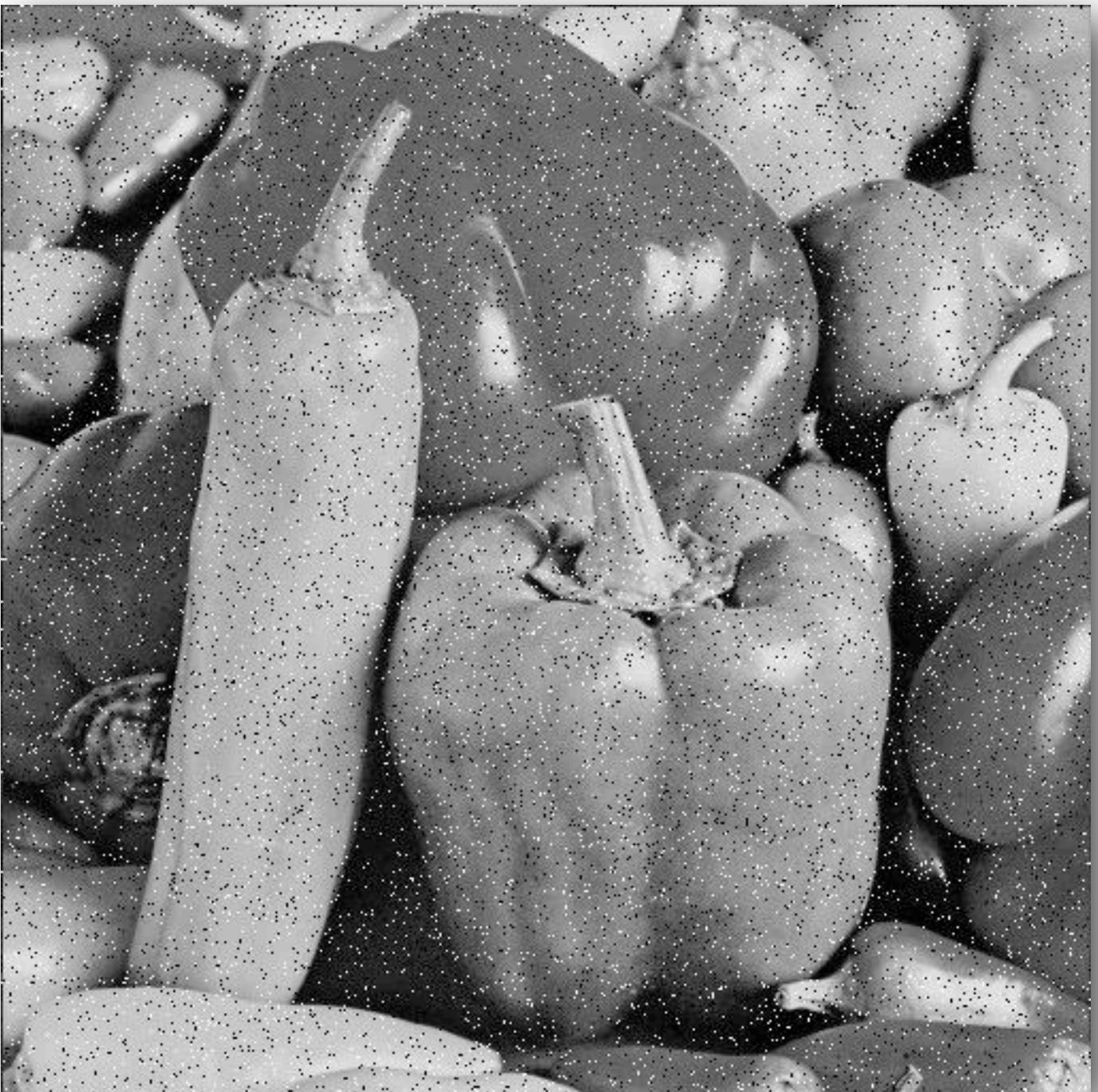


'Salt & Pepper'
Noise

Median
Filter
 3×3



Example: Median Filtering for Noise Removal



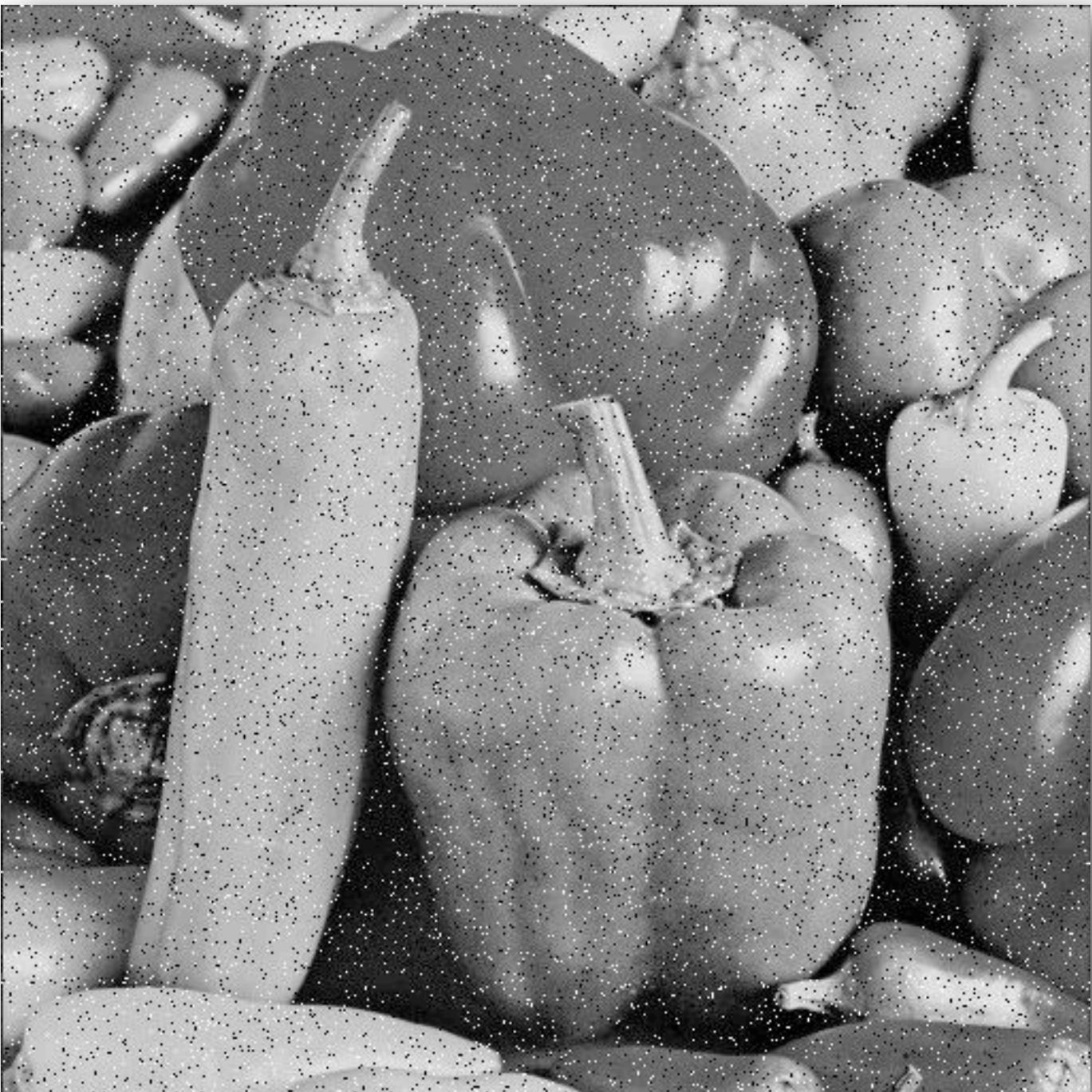
'Salt & Pepper'
Noise

Median
Filter
 3×3

Average
Filter
 3×3



Example: Median Filtering for Noise Removal



'Salt & Pepper'
Noise

Median
Filter
 3×3



Summary

- ★ Studied how image smoothing can be achieved.
- ★ Modeled the process of a kernel being applied to an image to smooth the image.
- ★ Studied specifics of averaging and median filtering over a neighborhood of pixels.



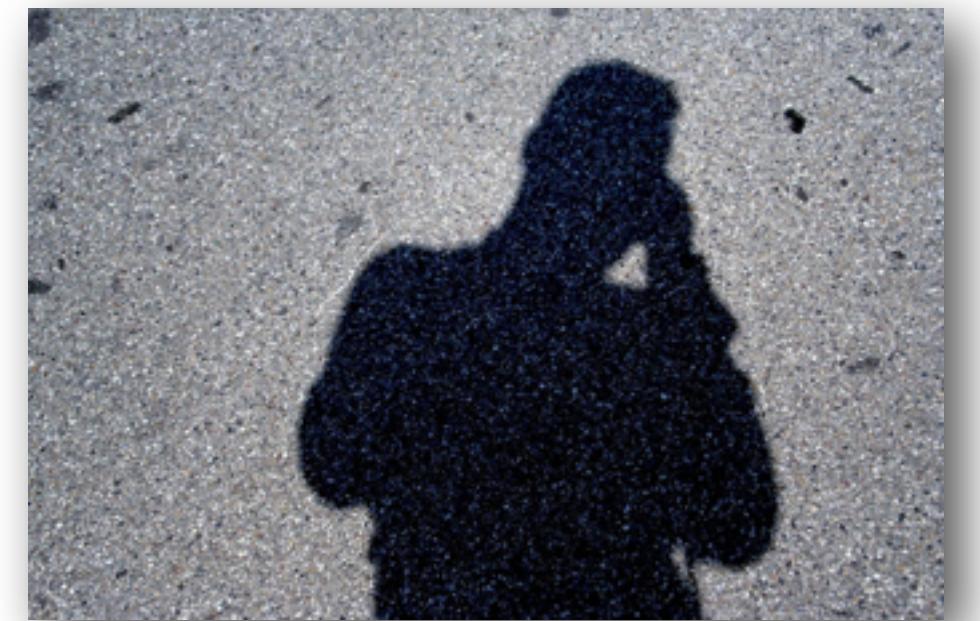
Next Class

★ 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



Dr. Irfan Essa

Professor

School of Interactive Computing



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