Image Filtering

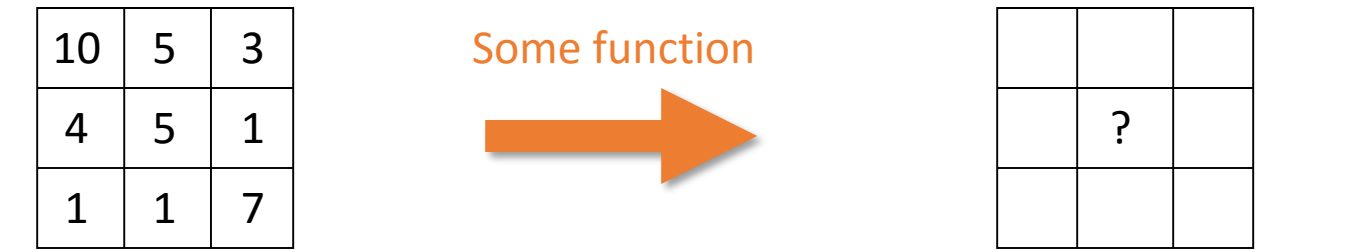
CPS592 – Visual Computing and Mixed Reality

Filter Operations

The name "filter" is used because these signal-processing elements typically "pass" or amplify certain frequency components of the signal, while they "stop" or attenuate others.

Image Filtering

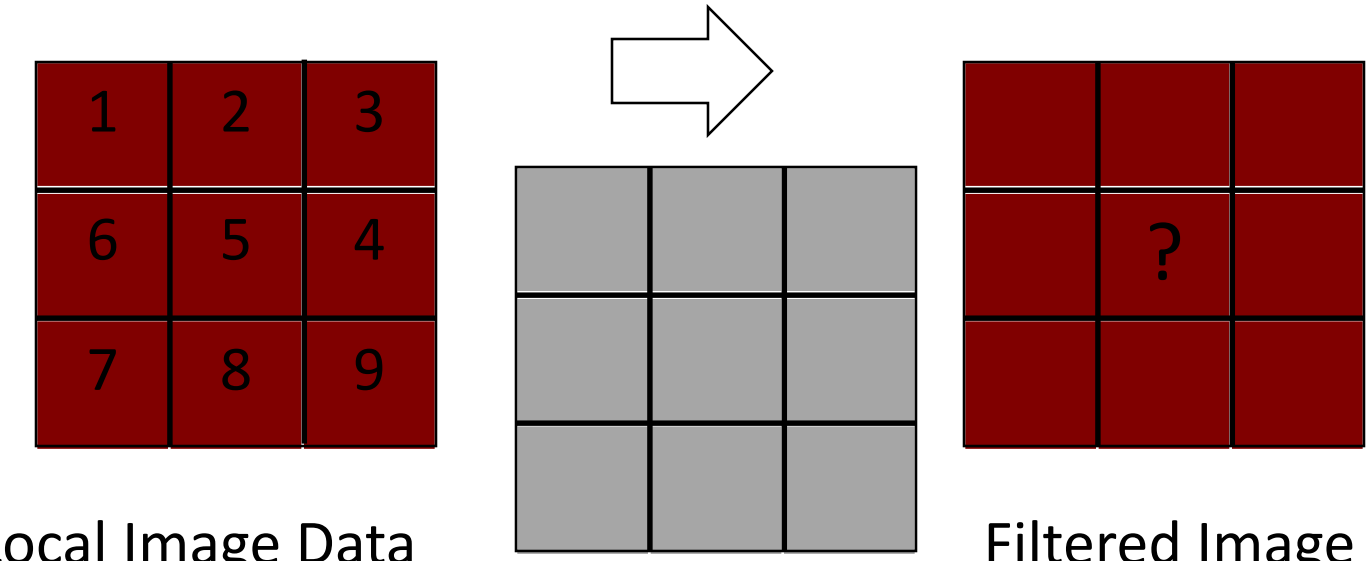
- Modify the pixels in an image based on some function of a local neighborhood of each pixel



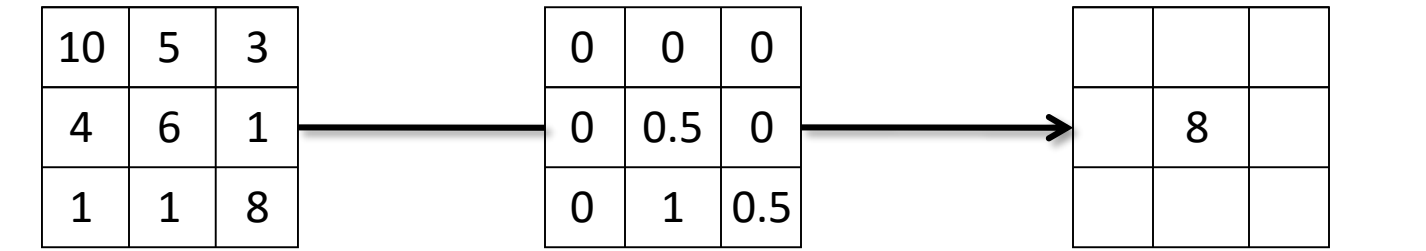
Source: L. Zhang

Convolution

- The prescription for the linear combination is called the “convolution kernel”.

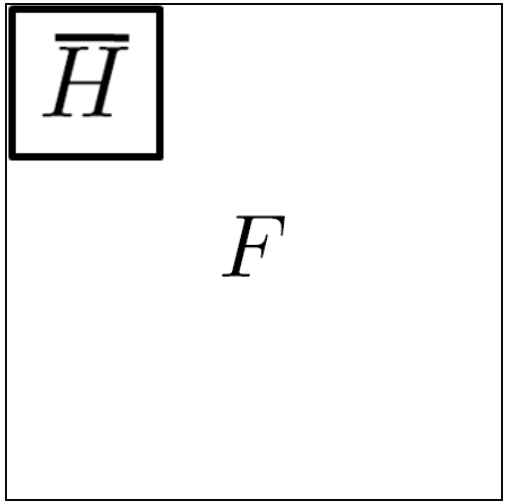


Example



Source: L. Zhang

Example



Adapted from F. Durand

Mean filtering

H

$*$

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

F

$=$

	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	

G

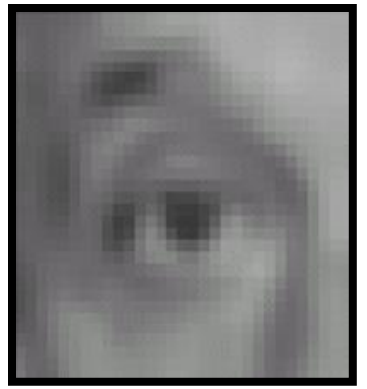
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$*$

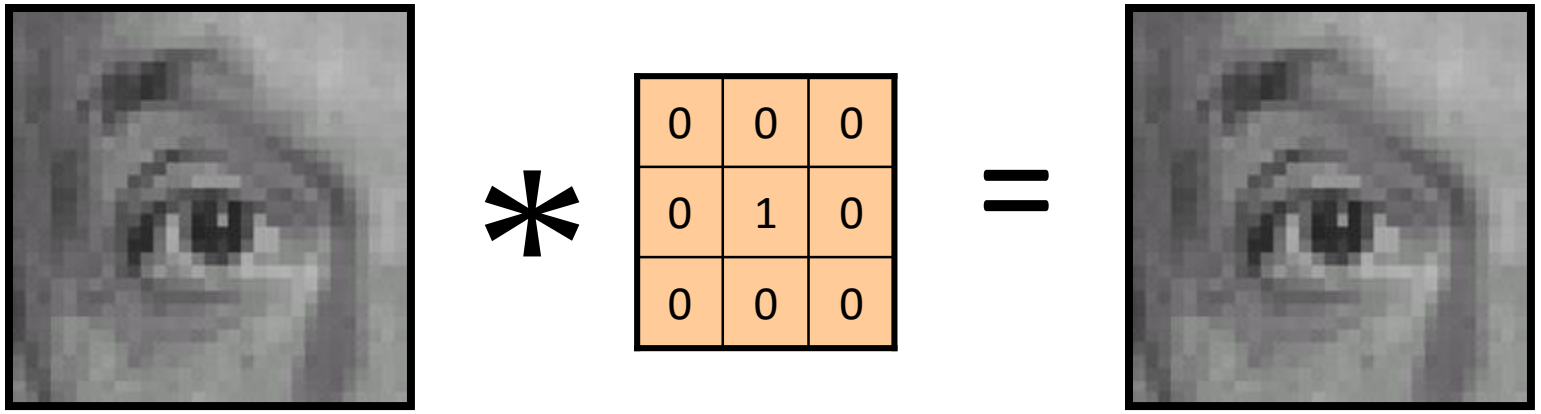
Original

$=$



Blur (with a mean filter)

Linear filters: examples

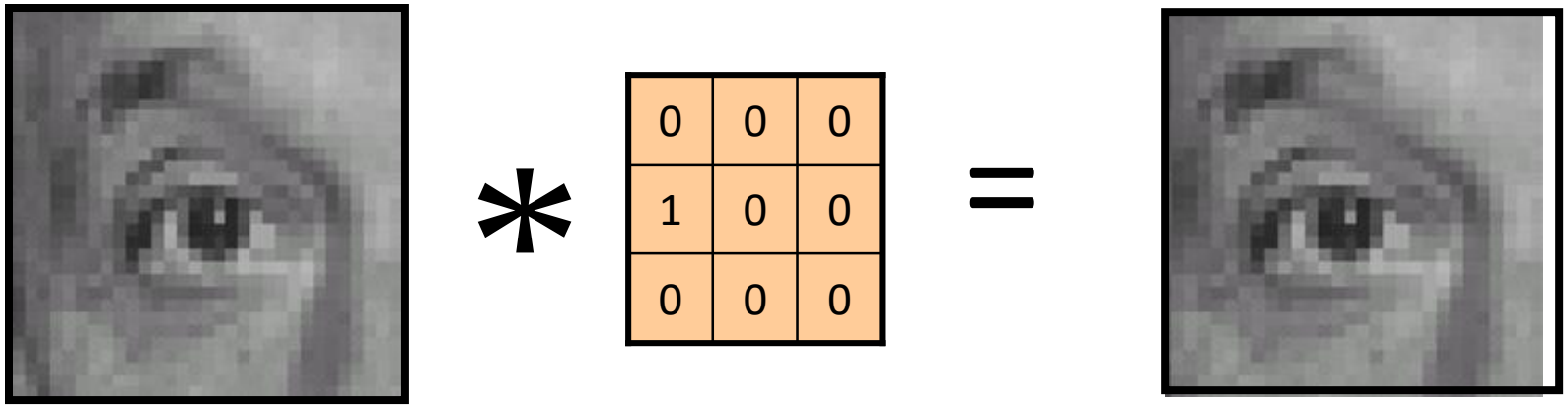


Original

Identical image

Source: D. Lowe

Linear filters: examples

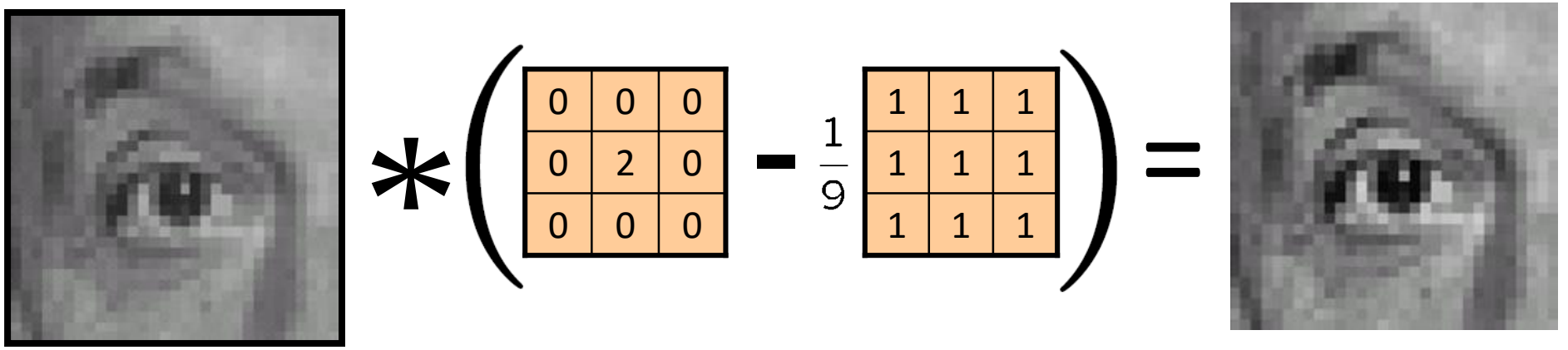


Original

Shifted left
By 1 pixel

Source: D. Lowe

Linear filters: examples

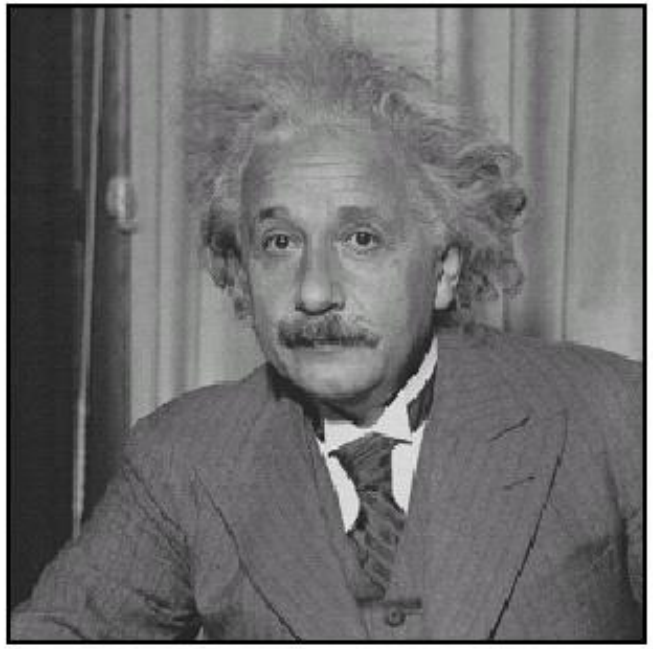


Original

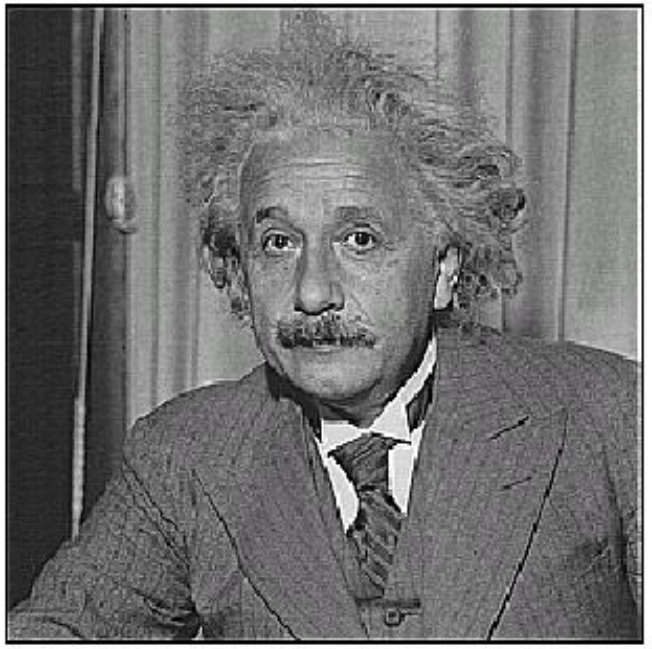
Sharpening filter
(accentuates edges)

Source: D. Lowe

Sharpening



before



after

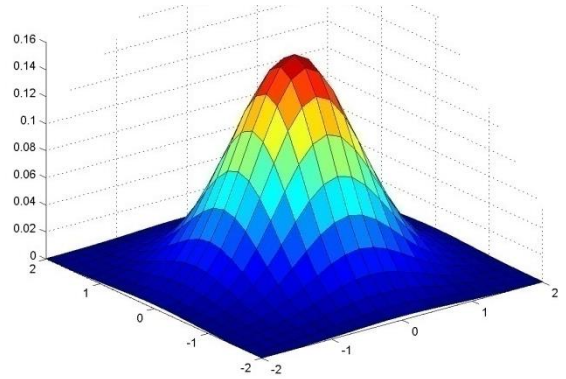
Source: D. Lowe

Gaussian filters

- A Gaussian kernel gives less weight to pixels further from the center of the window

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

$F[x, y]$



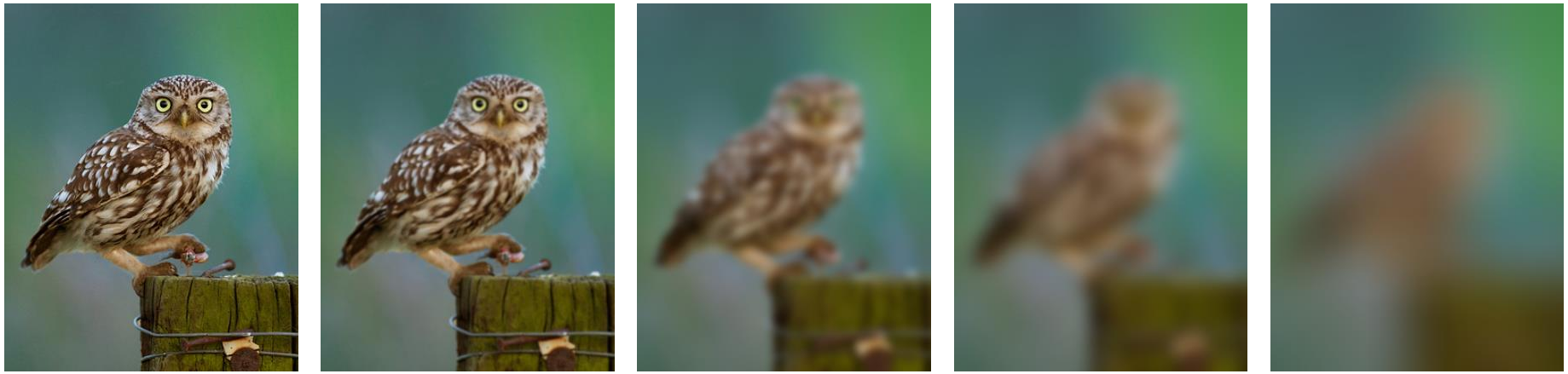
$$h(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

$\frac{1}{9}$

1	2	1
2	4	2
1	2	1

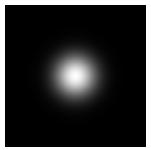
$H[u, v]$

Gaussian filters

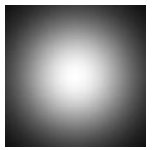


σ = 1 pixel

σ = 5 pixels



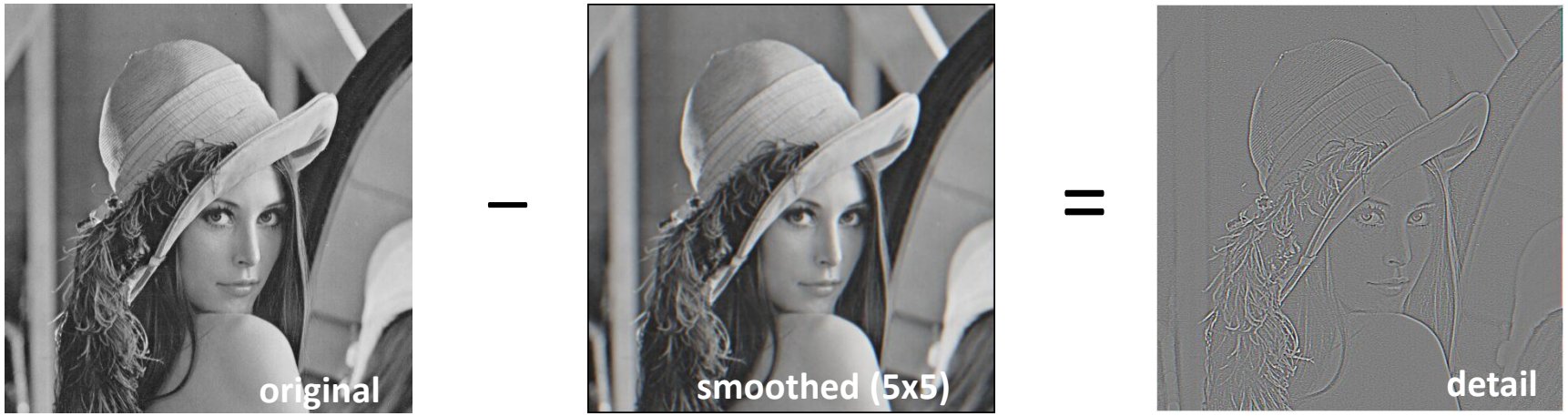
σ = 10 pixels



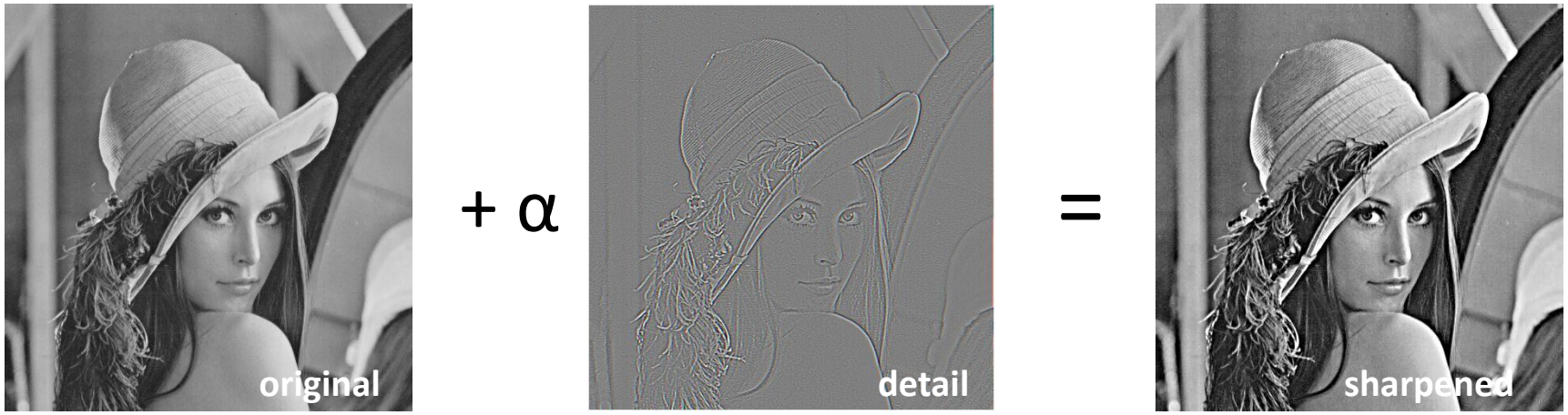
σ = 30 pixels

Sharpening revisited

- What does blurring take away?

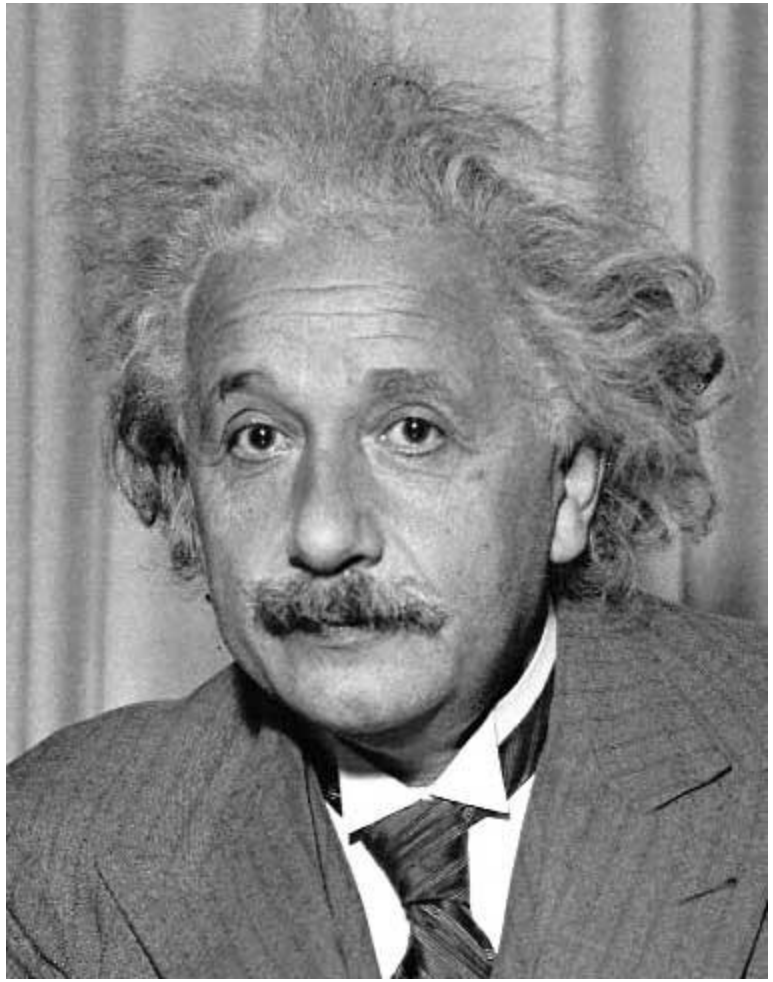


Let's add it back:



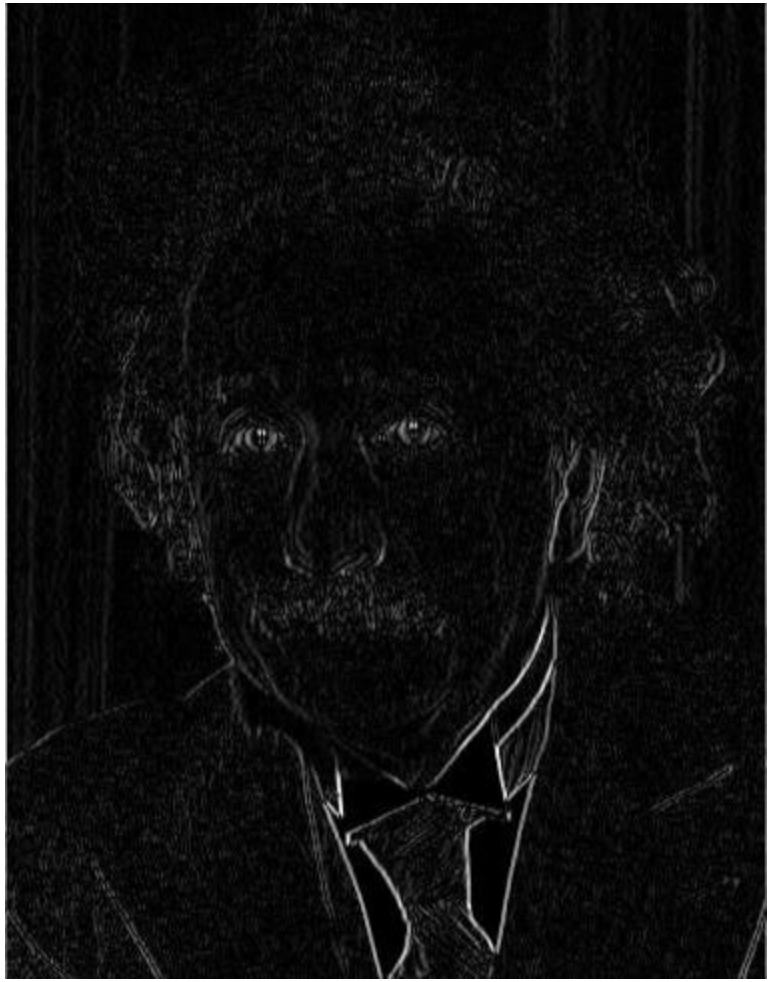
Source: S. Lazebnik

Other filters



1	0	-1
2	0	-2
1	0	-1

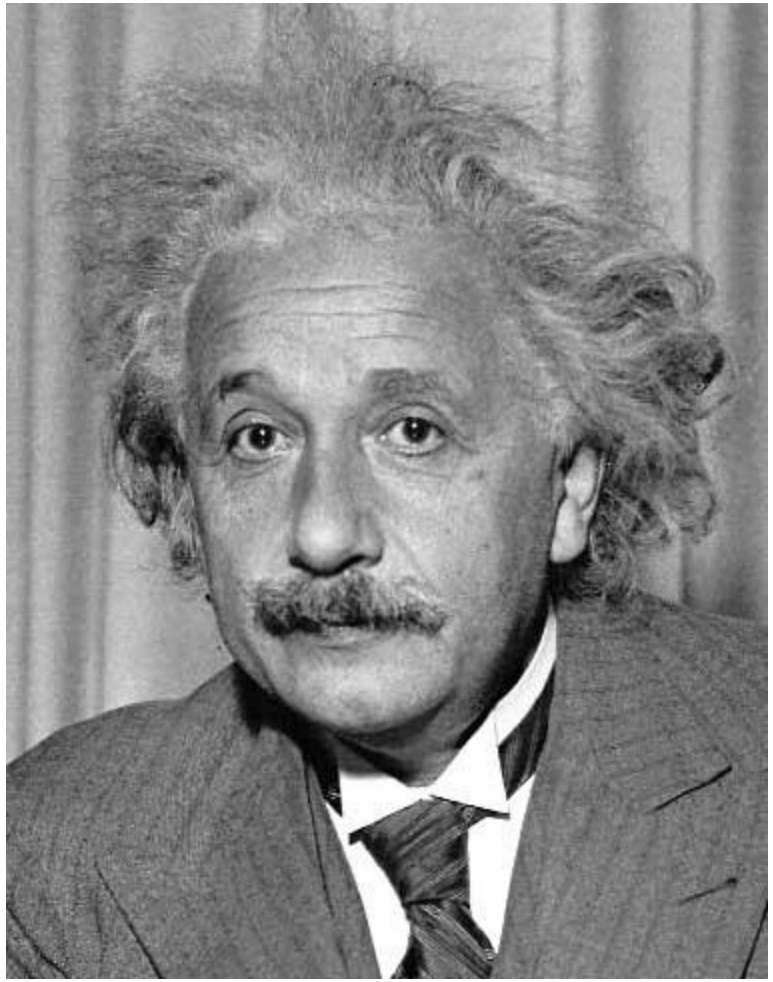
Sobel



Vertical Edge
(absolute value)

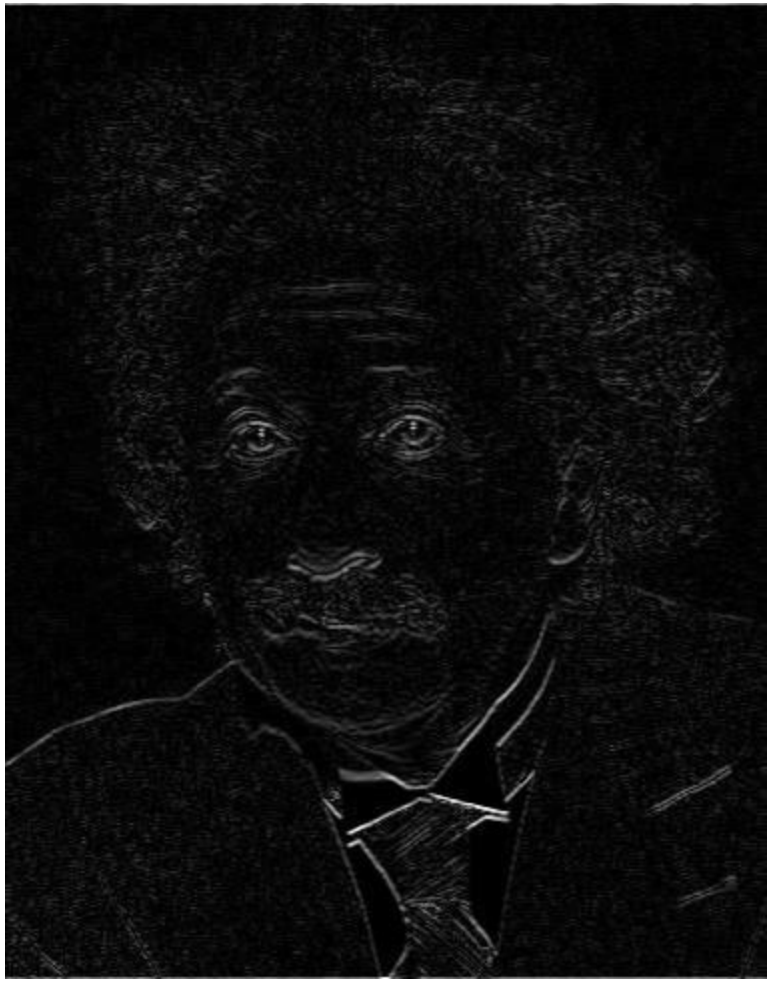
Source: D. Lowe

Other filters



1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

Source: D. Lowe

Linear vs. Non-linear

- “Convolution”/Linear Filters
  - Linear operation
  - Have corresponding frequency domain filter
- Non-linear Filters
  - Mask used to determine the proper substitution of a “good” pixel value
  - Examine neighbors using various orderings
  - Often use Rank or Order Statistics
  - Harder to interpret effect in frequency domain

Ordered Statistic Filters

- Also called “rank” filters

2	3	8
3	4	10
4	2	9

Consider a neighborhood about a pixel.
Rank (sort) the pixels.
{2, 2, 3, 3, 4, 4, 8, 9, 10}

Rank Filters: Median Filter

- One of the most popular non-linear filter
- Find the median of the window
- Preserves edges
- Removes impulse noise, avoids excessive smoothing

2	3	8
3	4	10
4	2	9

pixel values about (x,y) window 3x3

neighbor sort = {2,2,3,3,4,4,8,9,10}

f(x,y) = median

Rank Filters: Min/Max Filter

- Find the min or max of the neighborhood
- Not as “mainstream” as median filter
- Has various uses, will talk about these more later.

2	3	8
3	4	10
4	2	9

pixel values about (x,y) window 3x3

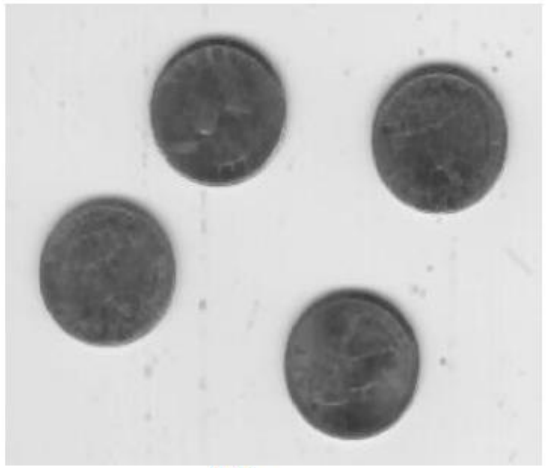
neighbor sort = {2,2,3,3,4,4,8,9,10}

f(x,y) = min **f(x,y) = max**

Examples

Original Image

Median



Min

Max

Sobel Filter

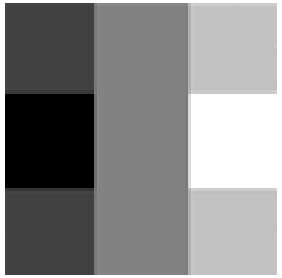
Kernel = [-1 0 1;

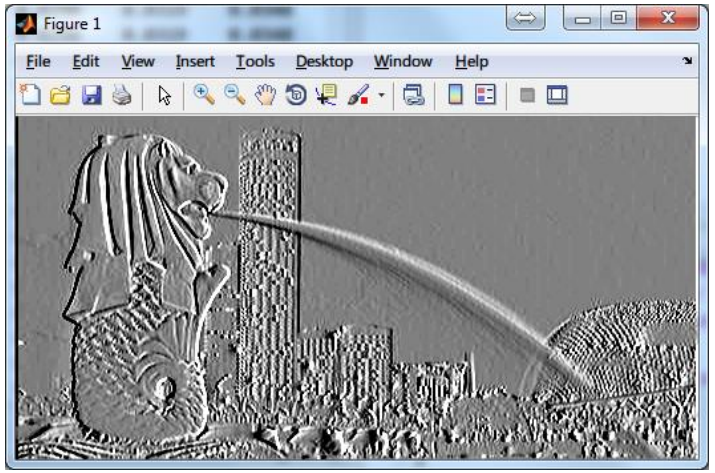
-2 0 2;

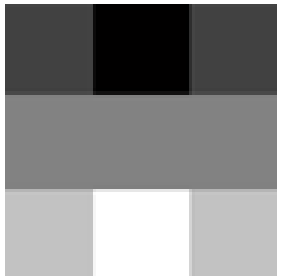
-1 0 1];

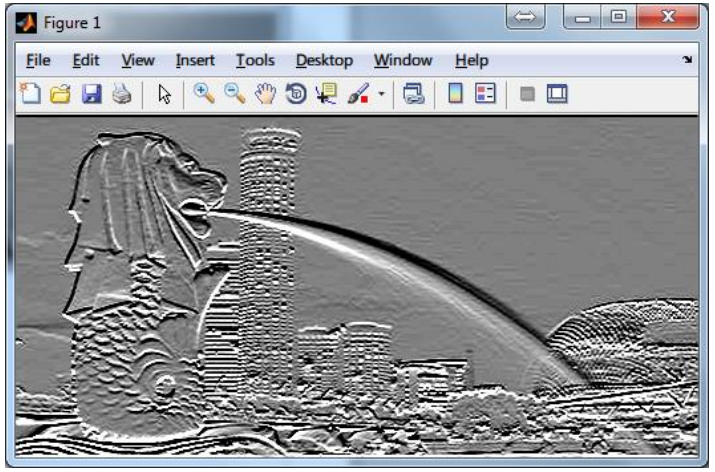
J1 = imfilter(I, kernel, 'replicate');

J2 = imfilter(I, kernel', 'replicate');







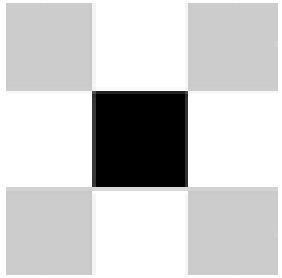


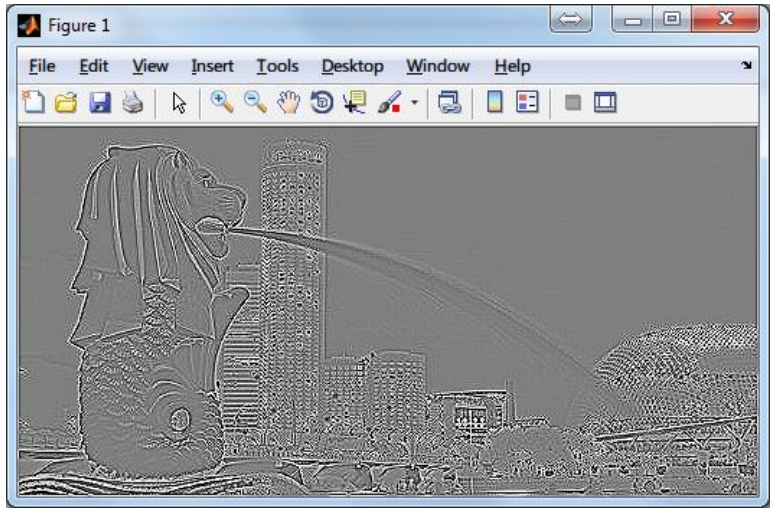
Laplacian Filter

Kernel = [0 1 0;

1 -4 1;

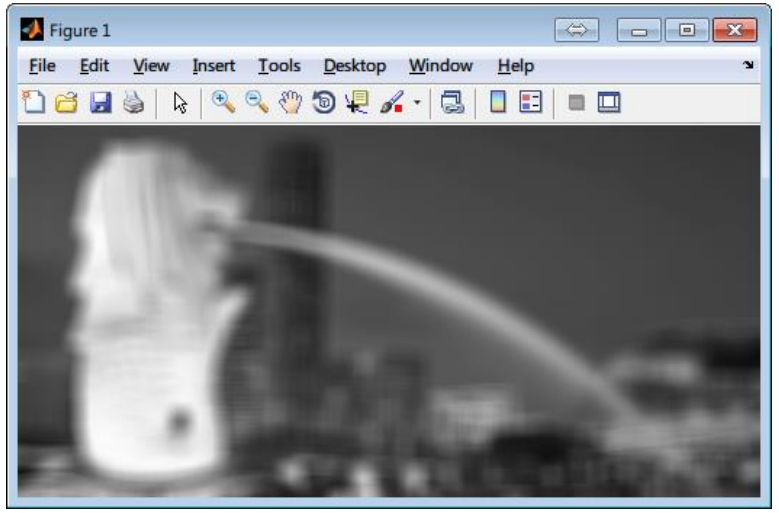
0 1 0];





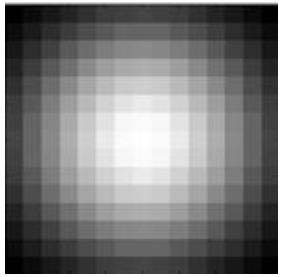
Mean Filter

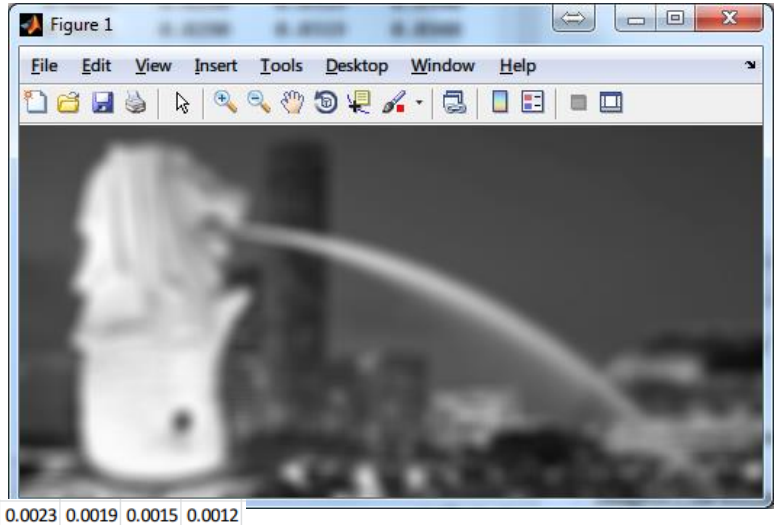
Kernel = ones(15,15) / 15^2;



Gaussian Filter

Kernel = fspecial('gaussian', [15 15], 5);

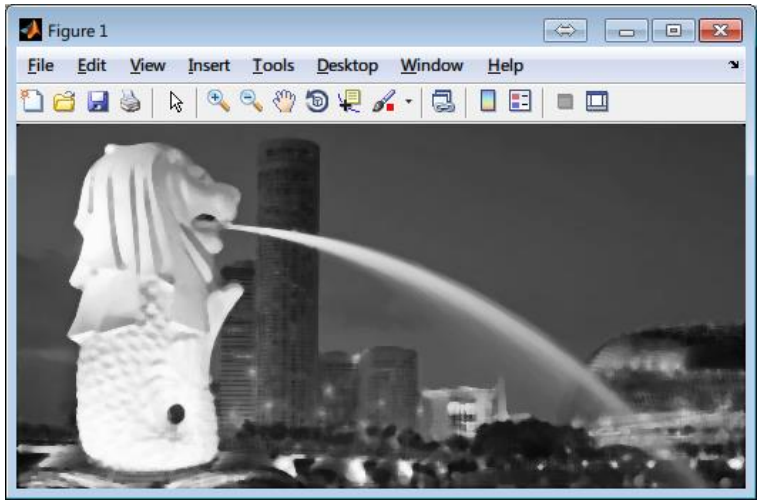




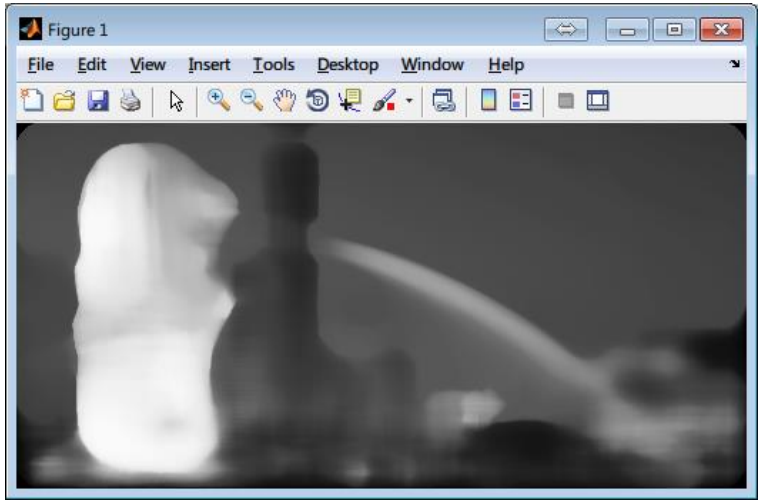
0.0012	0.0015	0.0019	0.0023	0.0027	0.0029	0.0031	0.0032	0.0031	0.0029	0.0027	0.0023	0.0019	0.0015	0.0012
0.0015	0.0020	0.0025	0.0030	0.0034	0.0038	0.0040	0.0041	0.0040	0.0038	0.0034	0.0030	0.0025	0.0020	0.0015
0.0019	0.0025	0.0031	0.0037	0.0043	0.0047	0.0050	0.0051	0.0050	0.0047	0.0043	0.0037	0.0031	0.0025	0.0019
0.0023	0.0030	0.0037	0.0045	0.0051	0.0057	0.0060	0.0061	0.0060	0.0057	0.0051	0.0045	0.0037	0.0030	0.0023
0.0027	0.0034	0.0043	0.0051	0.0059	0.0065	0.0069	0.0071	0.0069	0.0065	0.0059	0.0051	0.0043	0.0034	0.0027
0.0029	0.0038	0.0047	0.0057	0.0065	0.0072	0.0077	0.0078	0.0077	0.0072	0.0065	0.0057	0.0047	0.0038	0.0029
0.0031	0.0040	0.0050	0.0060	0.0069	0.0077	0.0081	0.0083	0.0081	0.0077	0.0069	0.0060	0.0050	0.0040	0.0031
0.0032	0.0041	0.0051	0.0061	0.0071	0.0078	0.0083	0.0085	0.0083	0.0078	0.0071	0.0061	0.0051	0.0041	0.0032
0.0031	0.0040	0.0050	0.0060	0.0069	0.0077	0.0081	0.0083	0.0081	0.0077	0.0069	0.0060	0.0050	0.0040	0.0031
0.0029	0.0038	0.0047	0.0057	0.0065	0.0072	0.0077	0.0078	0.0077	0.0072	0.0065	0.0057	0.0047	0.0038	0.0029
0.0027	0.0034	0.0043	0.0051	0.0059	0.0065	0.0069	0.0071	0.0069	0.0065	0.0059	0.0051	0.0043	0.0034	0.0027
0.0023	0.0030	0.0037	0.0045	0.0051	0.0057	0.0060	0.0061	0.0060	0.0057	0.0051	0.0045	0.0037	0.0030	0.0023
0.0019	0.0025	0.0031	0.0037	0.0043	0.0047	0.0050	0.0051	0.0050	0.0047	0.0043	0.0037	0.0031	0.0025	0.0019
0.0015	0.0020	0.0025	0.0030	0.0034	0.0038	0.0040	0.0041	0.0040	0.0038	0.0034	0.0030	0.0025	0.0020	0.0015
0.0012	0.0015	0.0019	0.0023	0.0027	0.0029	0.0031	0.0032	0.0031	0.0029	0.0027	0.0023	0.0019	0.0015	0.0012

Median Filter

5x5



25x25



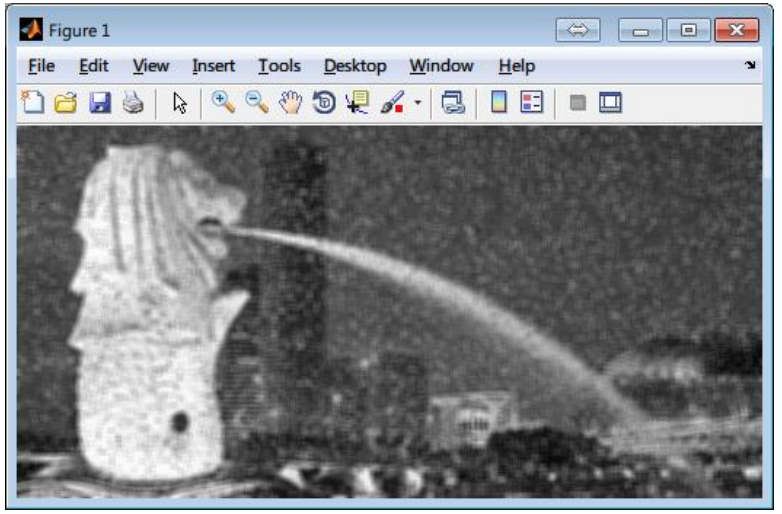
Sharpening


x 2 -

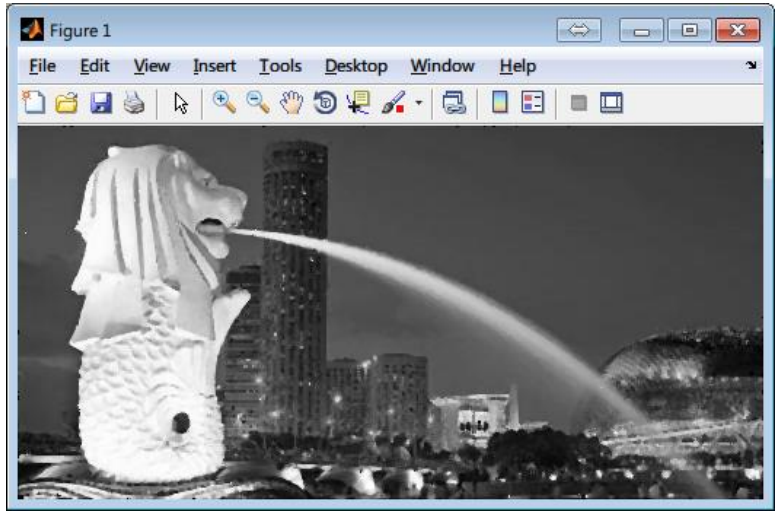
=


Denoising

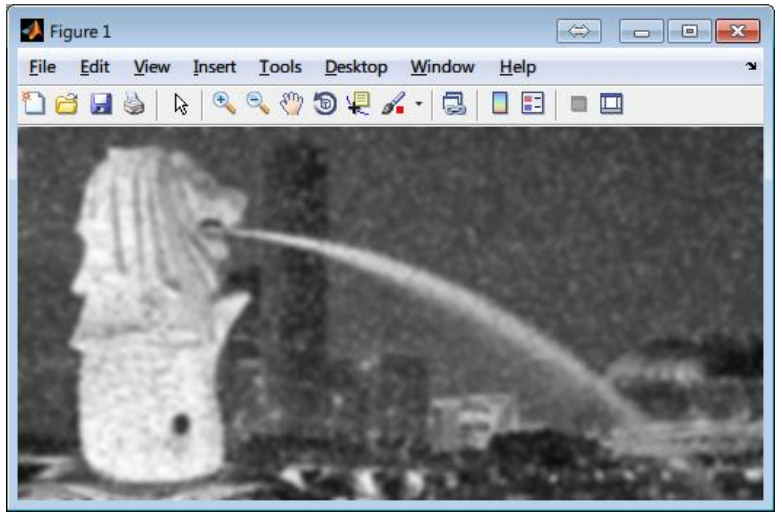
Mean



Median



Gaussian



Q&A