Introduction to Computer Vision

2. Basic Image Processing

UCLA – CS 188 – Fall 2019

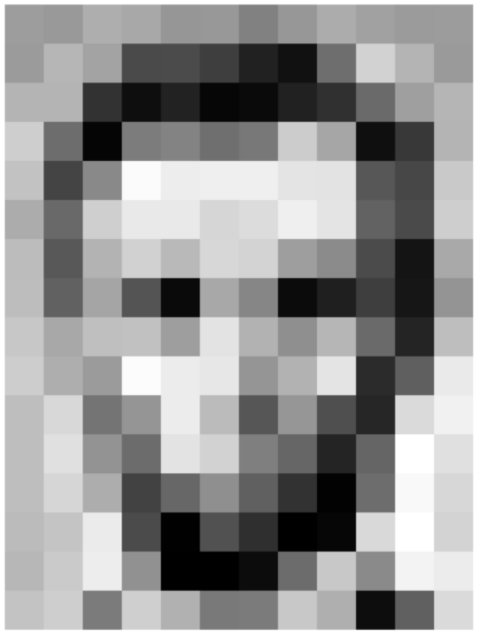
Fabien Scalzo, Ph.D.

Today: Basic Image Processing

Week 1			26-Sep	**Introduction**
Week 2	1-Oct	**Basic Image Processing**	3-Oct	**Feature Extraction and Classification**
Week 3	8-Oct	**Feature Tracking/Optical Flow**	10-Oct	**SVD, 2D camera model, projective plane**
Week 4	15-Oct	**2D Image transformations, RANSAC**	17-Oct	**Euclidean geometry, rigid body motion**
Week 5	22-Oct	**Epipolar Geometry**	24-Oct	**3D Cameras and processing**
Week 6	29-Oct	**Midterm**	31-Oct	**Statistical decision theory/Pattern Recognition**
Week 7	5-Nov	**Deep Learning**	7-Nov	**Deep Learning for Image Classification**
Week 8	12-Nov	**Object Detection**	14-Nov	**Generative models**
Week 9	19-Nov	**Medical Imaging**	21-Nov	**Autonomous Navigation**
Week 10	26-Nov	**Guest Lecture (Nikhil Naik)**	28-Nov	
Week 11	3-Dec	**Recursive 3D reconstruction and pose estimation**	5-Dec	**Recap**
Week 12	10-Dec		12-Dec	**Final**

Progressive Growing of GANs

Image representation (2D)

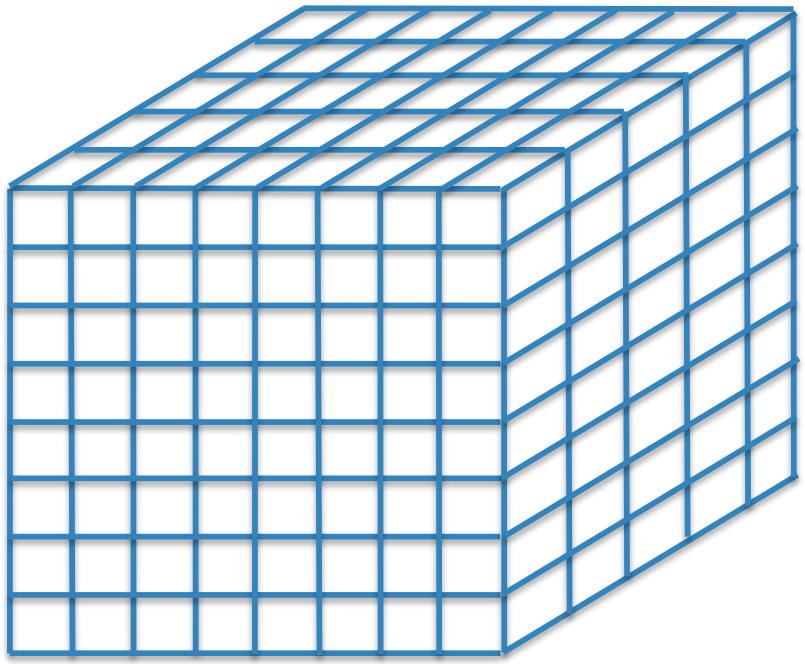


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

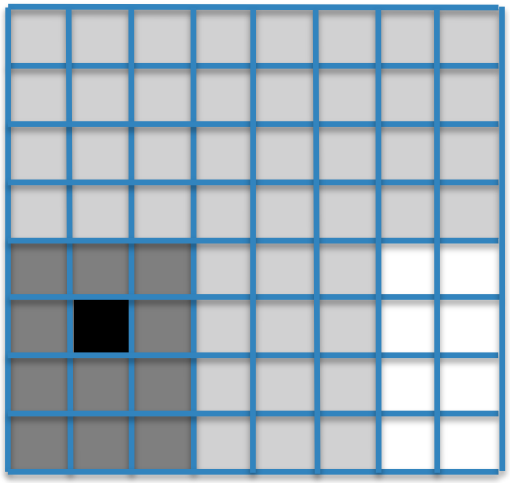
Array of pixels

Image representation (3D)



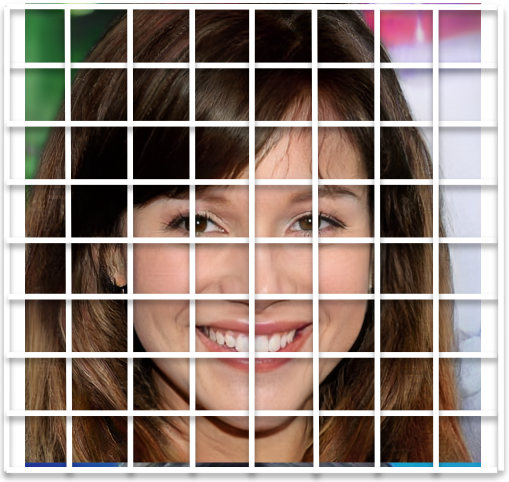
Array of voxels

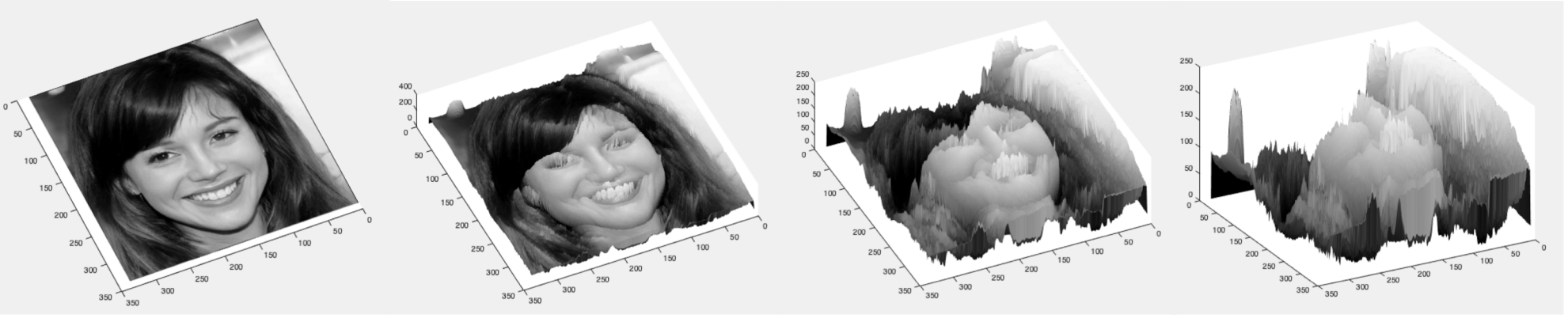
Image representation



Each pixel is described by a position (x,y) in the image and an intensity value I(x,y)

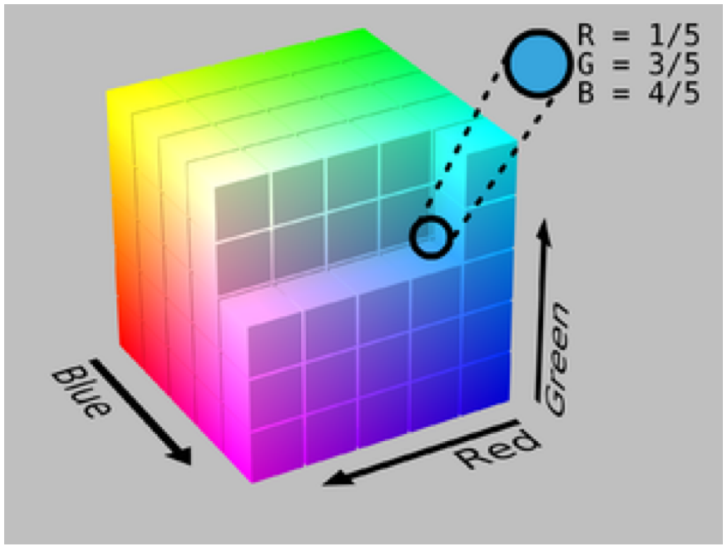
Image representation





RGB Image representation

Each pixel is described by a position (x,y) in the image and its RGB intensity values R(x,y), G(x,y), B(x,y)



red

green

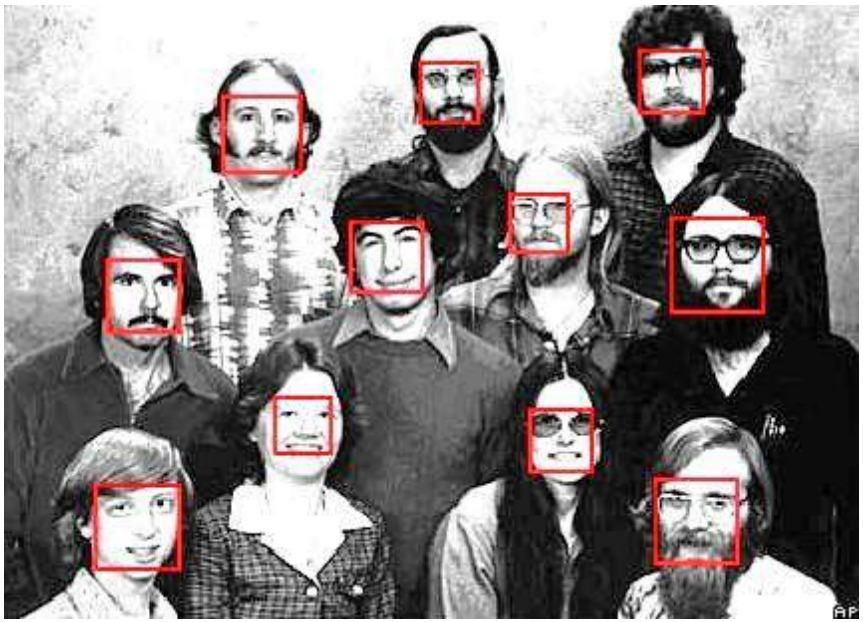
blue

Image filtering

- Purpose: image quality improvement, eliminate unwanted structures, emphasize structures of interest.

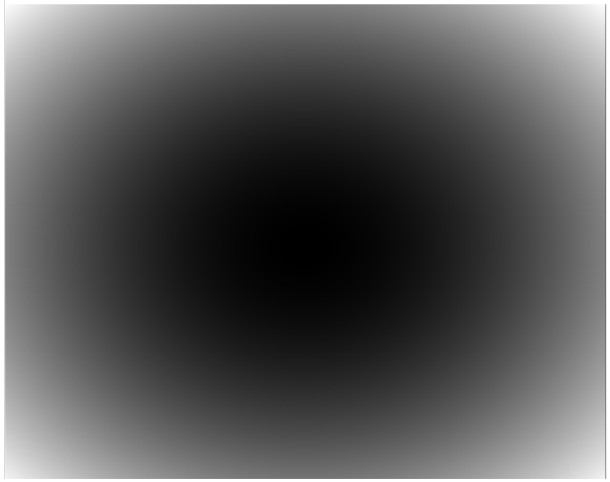
- In most of computer vision and imaging applications.

Feature correspondence, image registration, retrieval, categorization, face detection/recognition, …



We don’t “see” most information in image

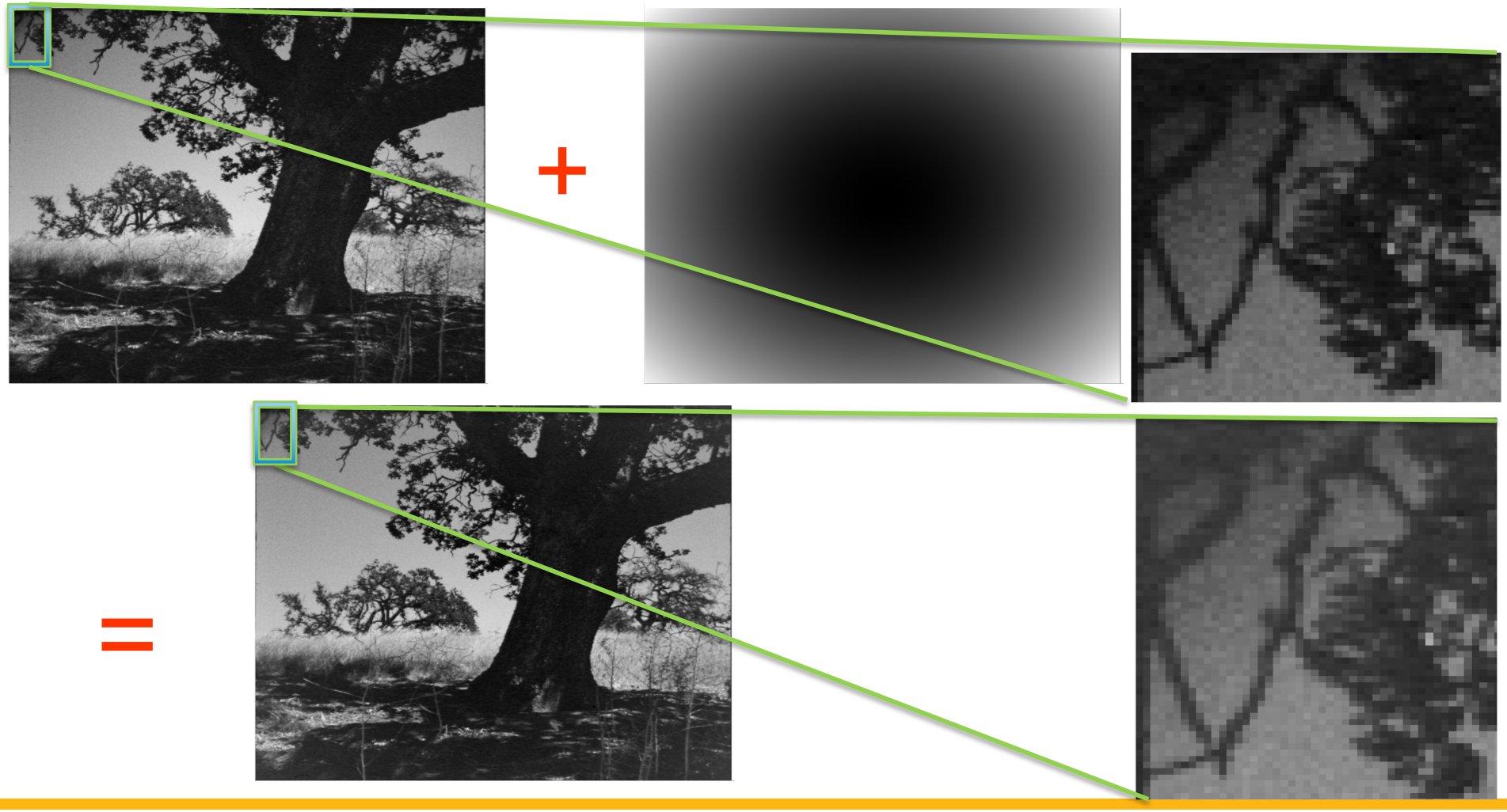
+



=

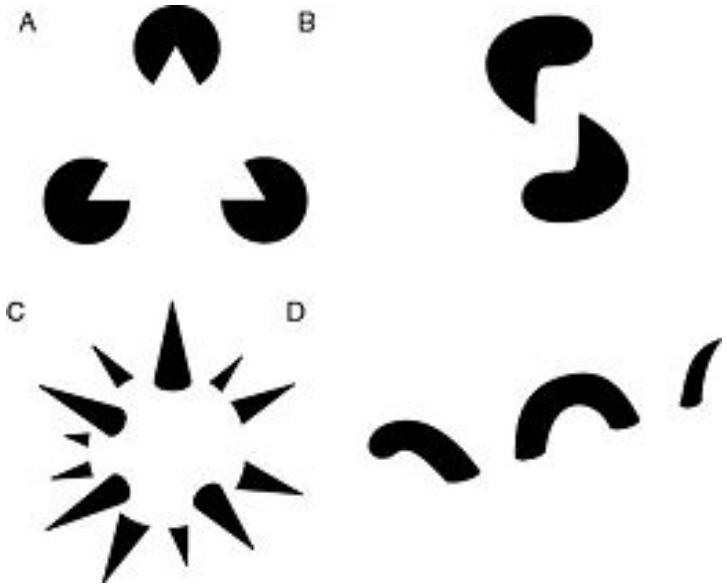
Can add invisible change

We don’t “see” most information in image



Images are complex to process

Human vision uses various tricks, such as **Proximity, Similarity, Continuity** to make processing more efficient.



see Gestalt psychology

The brain “filters” what matters,

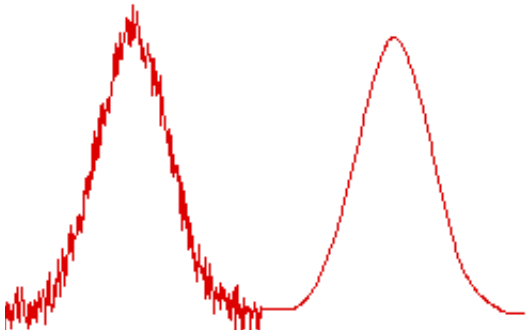
and what matters is purpose dependent

Filtering: the basics

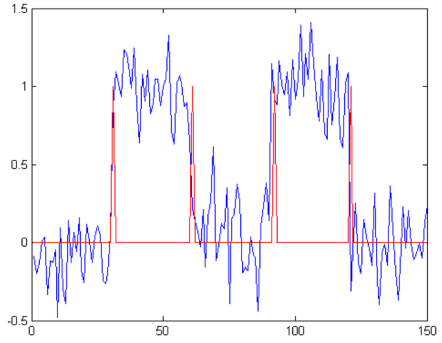
Performs mathematical operations to reduce or enhance certain aspects of a signal (1D, 2D, …)

A **low-pass** filter passes signals with a frequency lower than a certain cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency.

A **high-pass** does the opposite and its output is proportional to rate of change of the input signal.

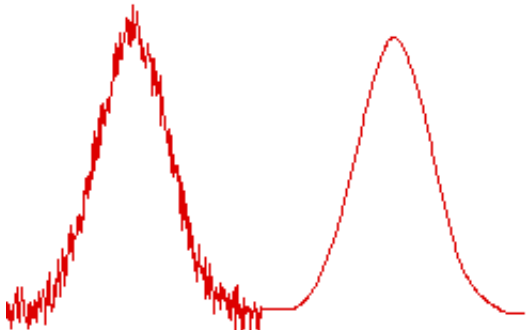


noise removal (low-pass)

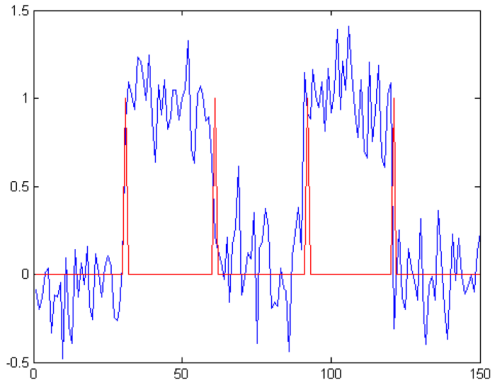


change detection (high-pass)

Filtering: the basics



noise removal (low-pass)



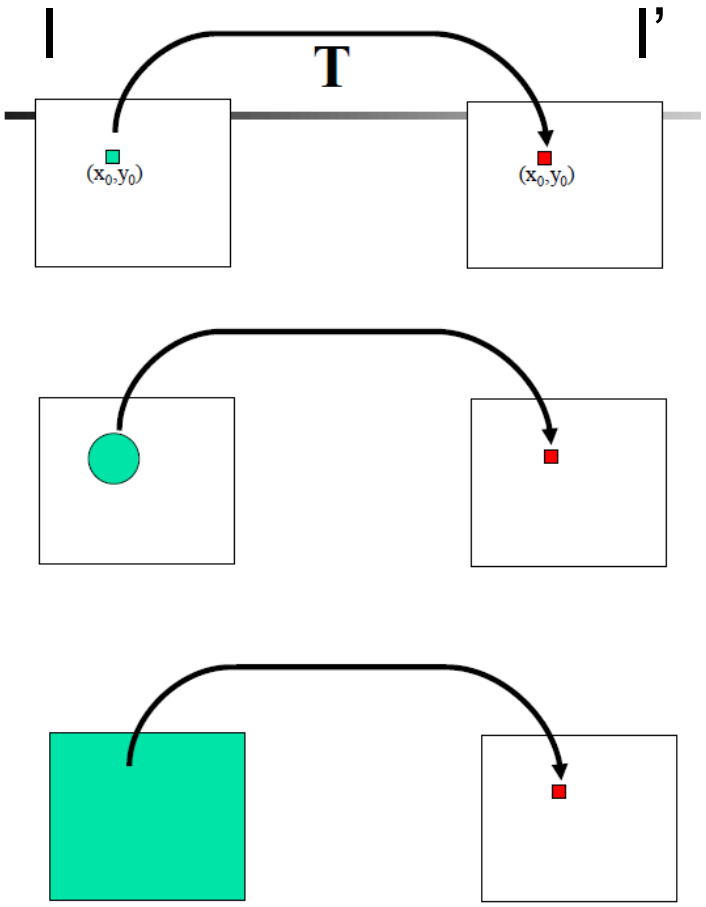
change detection (high-pass)

Filters can be applied either:

- In the frequency domain: apply Fourier transform and its inverse after application of the filter.
- **In the spatial domain**: convolve the image with the filter.

Image Filtering

Modify the pixels in an image based on some function T



Point-wise

$I'(x_0,y_0)=T[I(x_0,y_0)]$

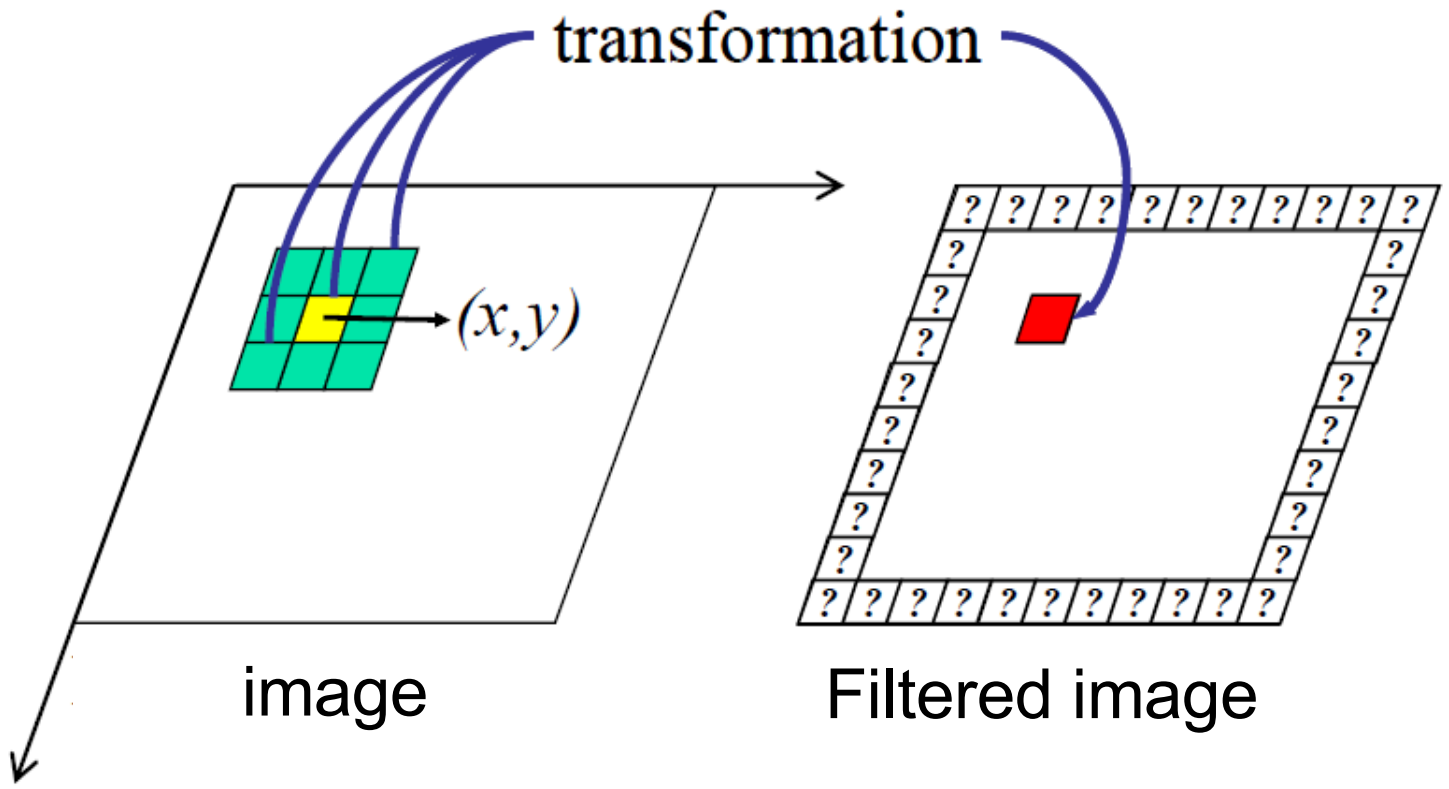
Local

$I'(x_0,y_0)=T[I(V)]$

Global

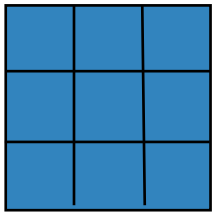
$I'(x_0,y_0)=T[I(x,y)]$

2D Local filter

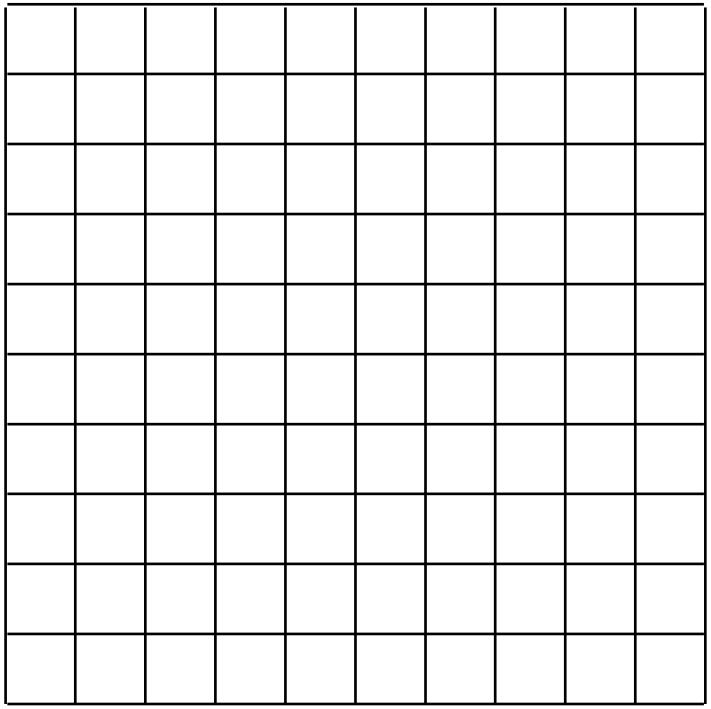


Examples: minimum, maximum, average

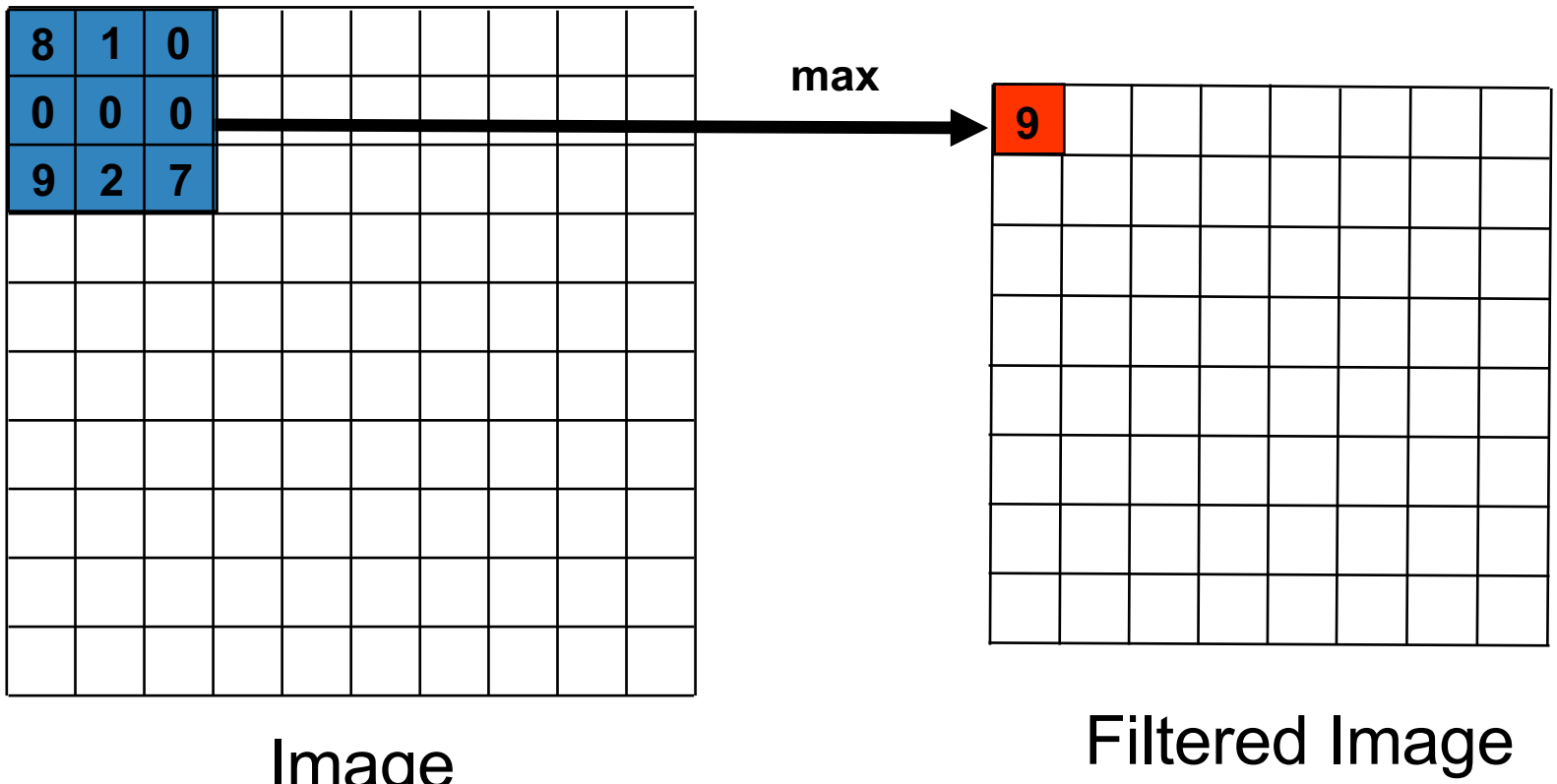
2D Local Filtering

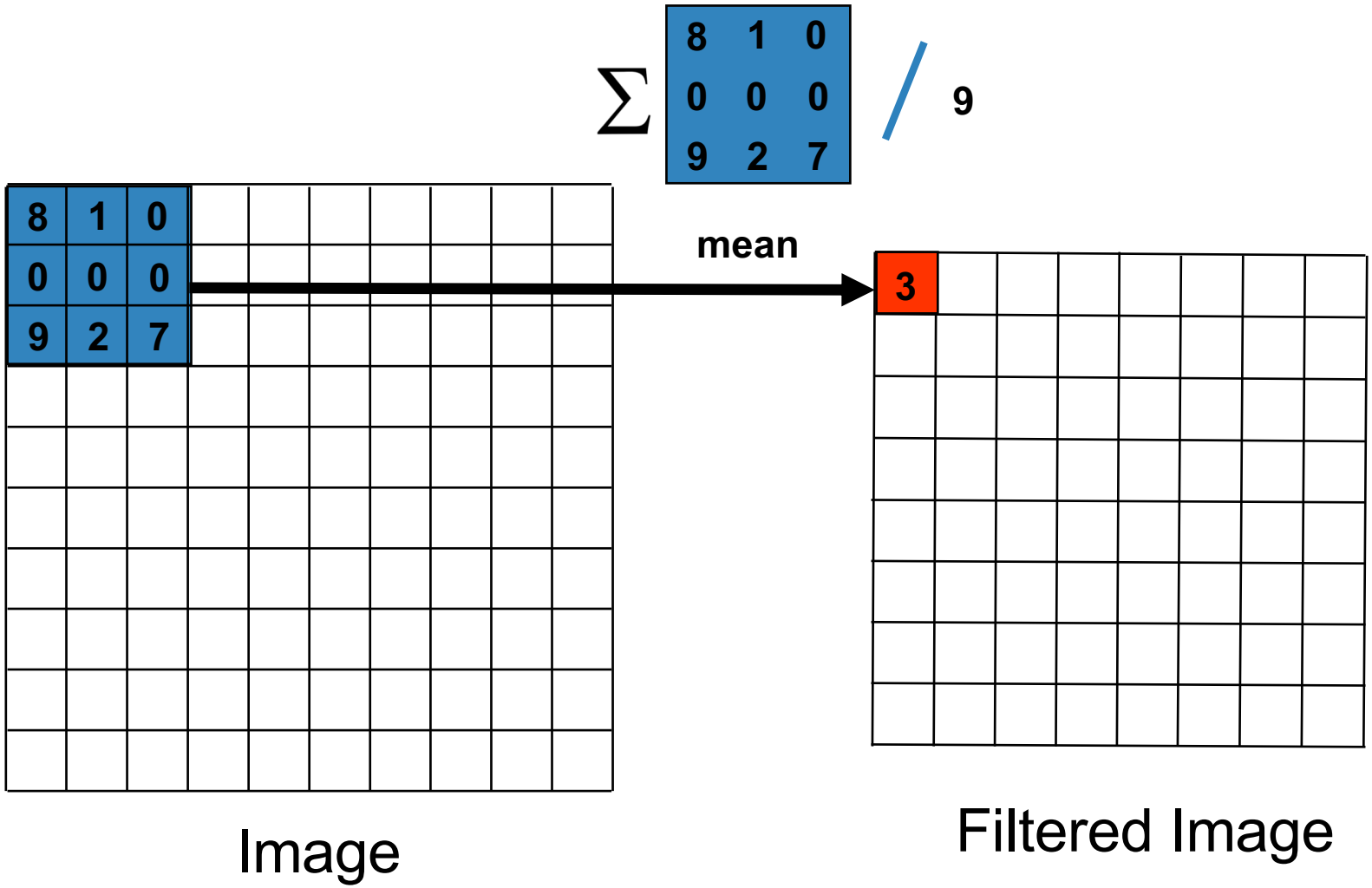


Filter



Image


8 1 0
0 0 0
9 2 7
max
9
Image
Filtered Image


8 1 0
0 0 0
9 2 7
/ 9
8 1 0
0 0 0
9 2 7
mean
3
Image
Filtered Image

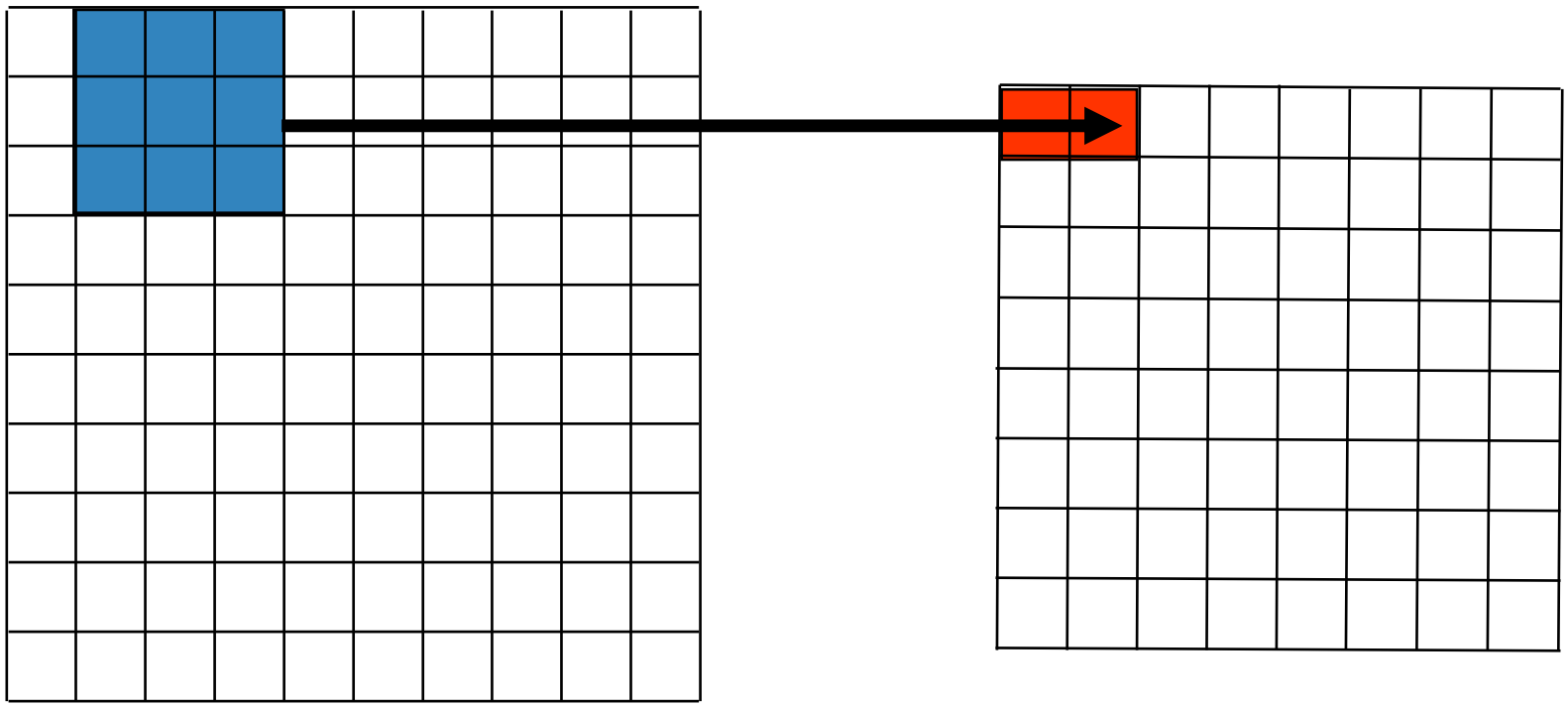


Image
Filtered Image

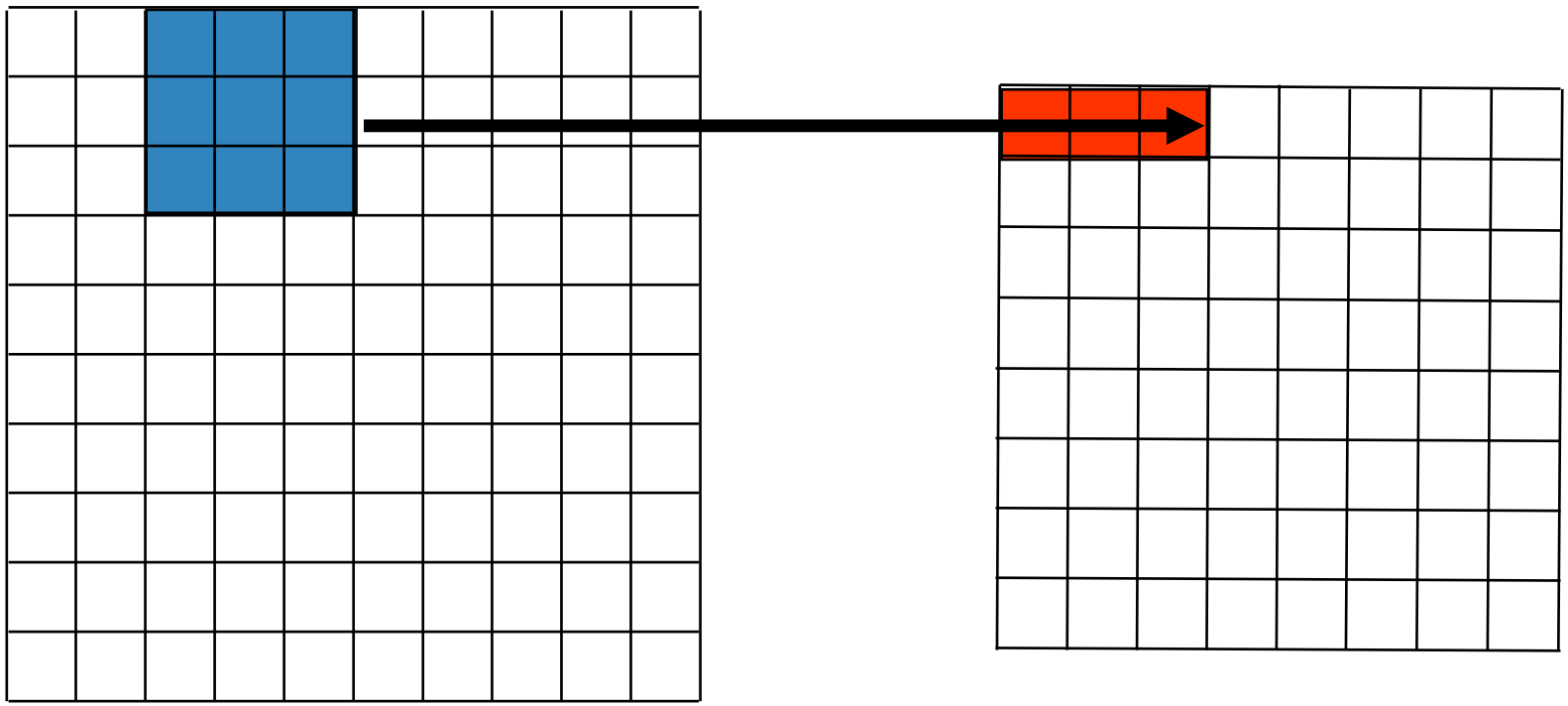


Image
Filtered Image

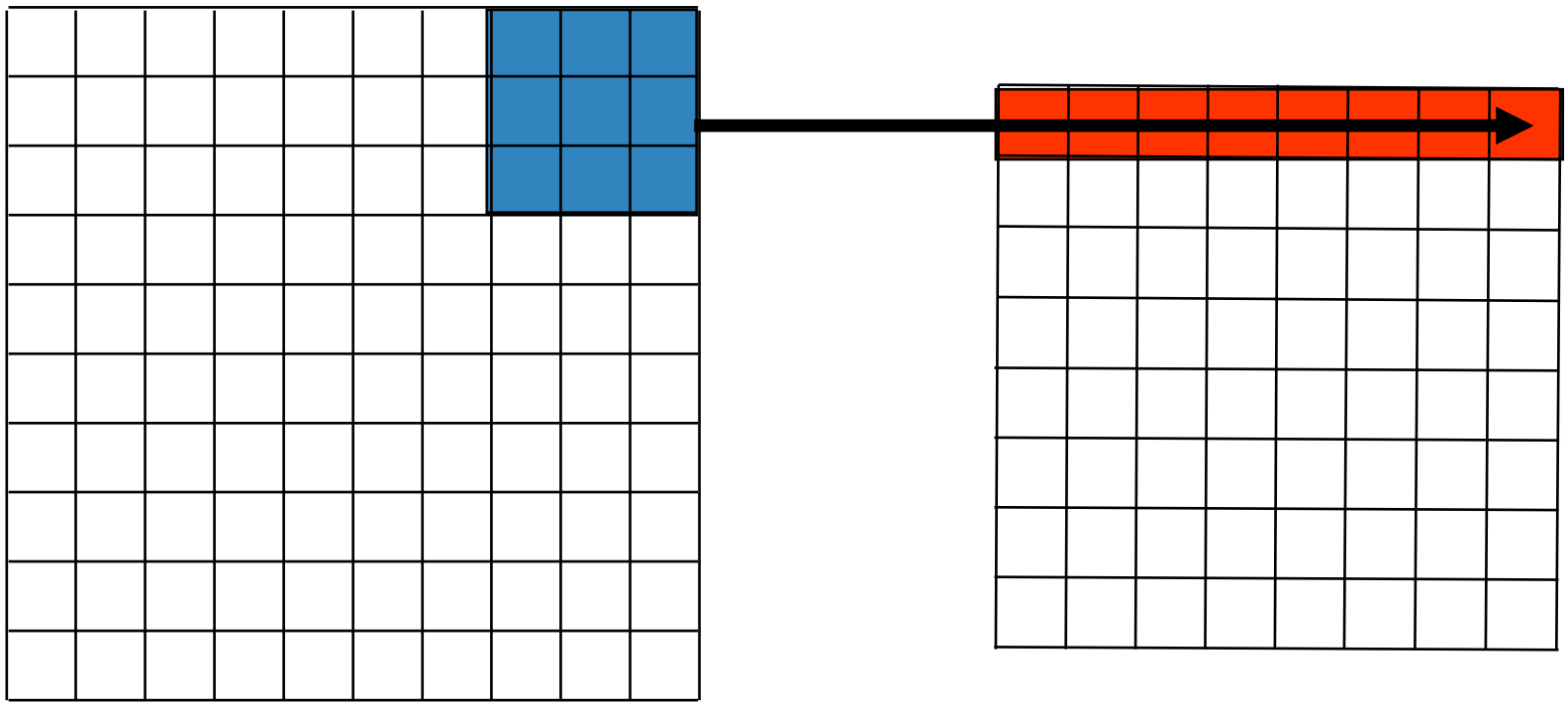


Image
Filtered Image

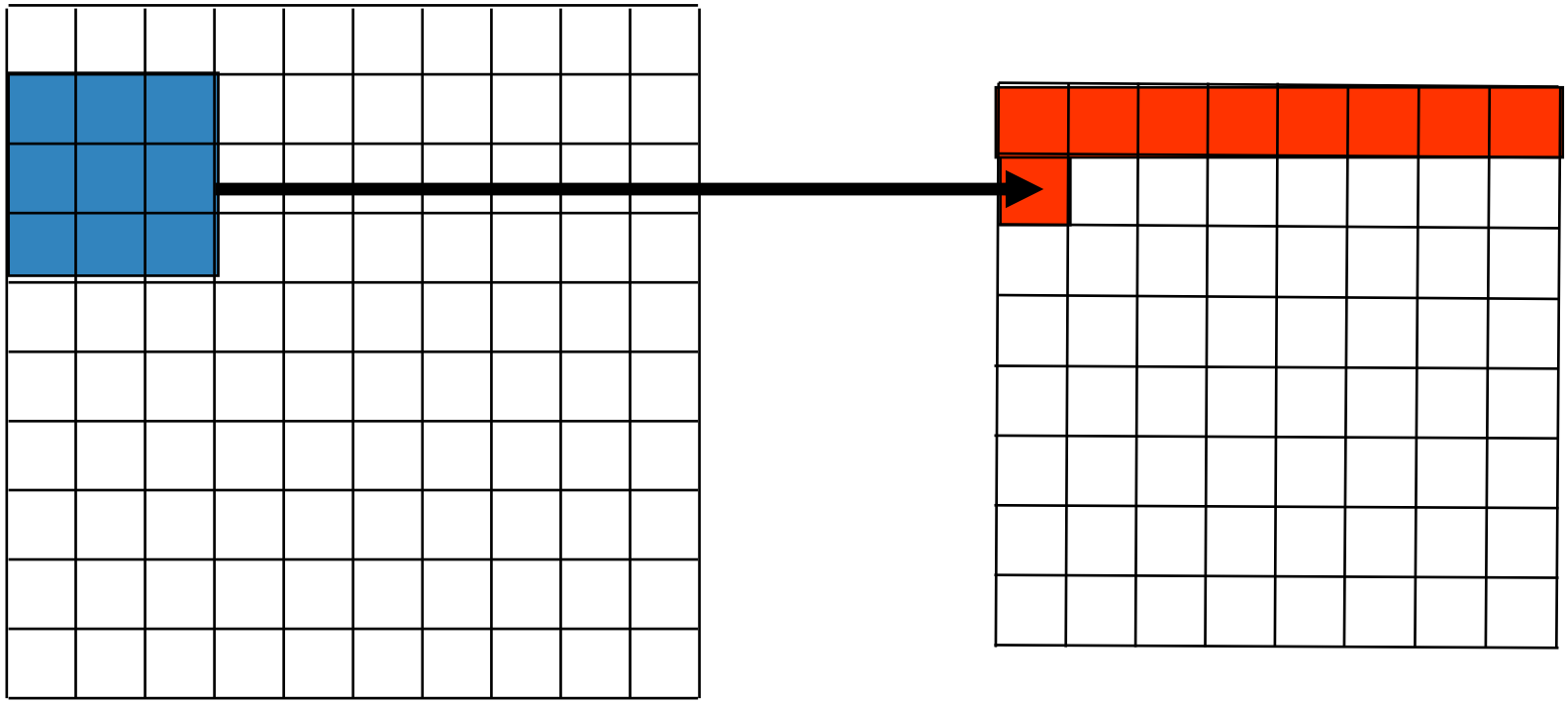


Image
Filtered Image

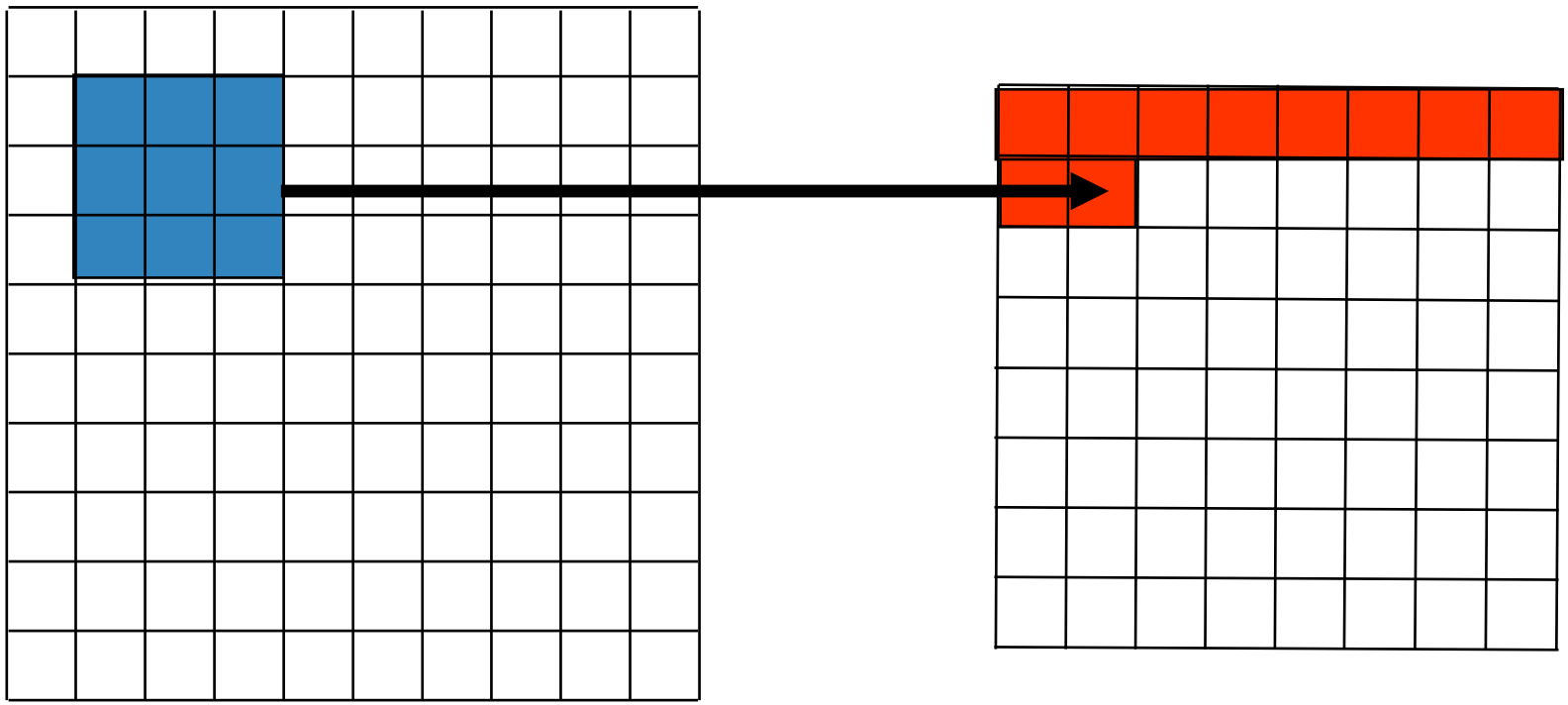


Image
Filtered Image

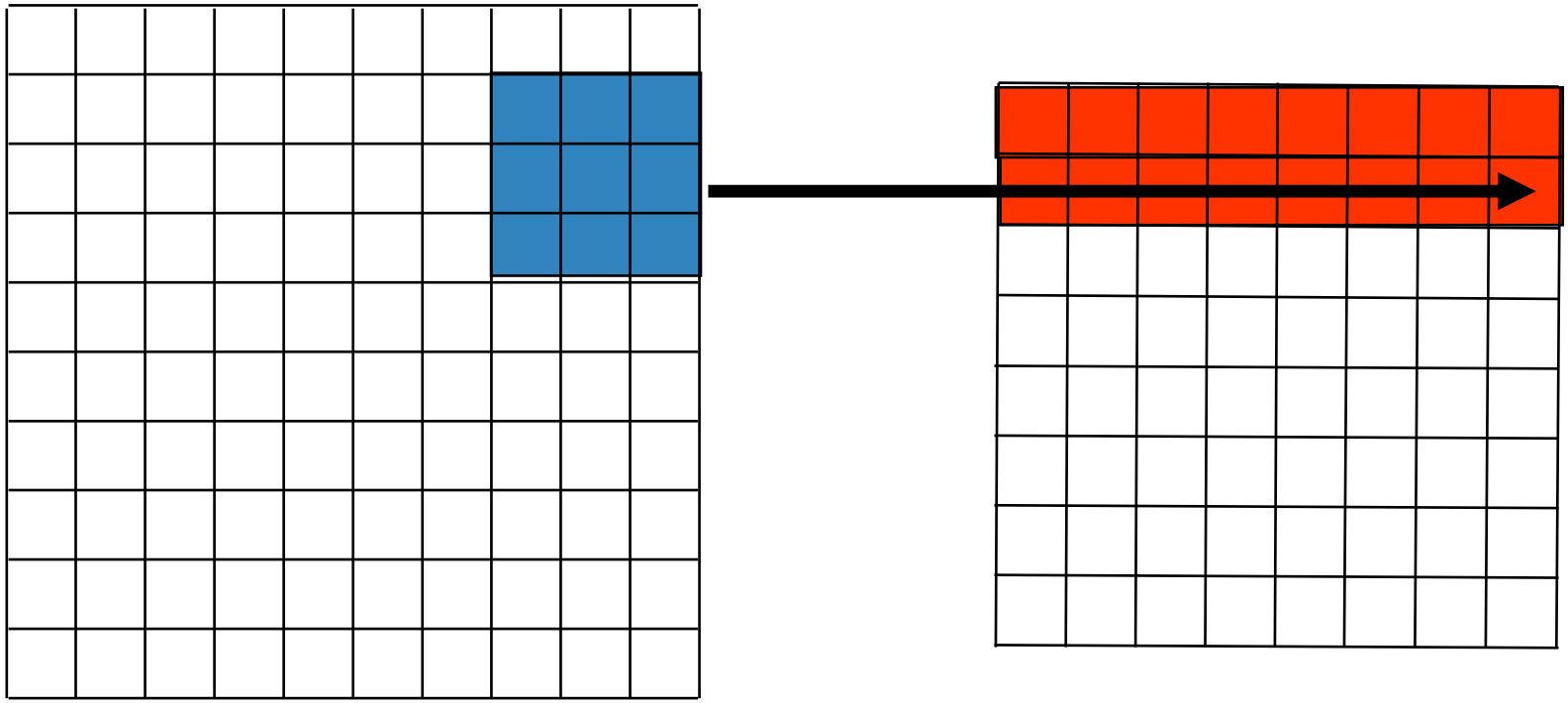


Image
Filtered Image

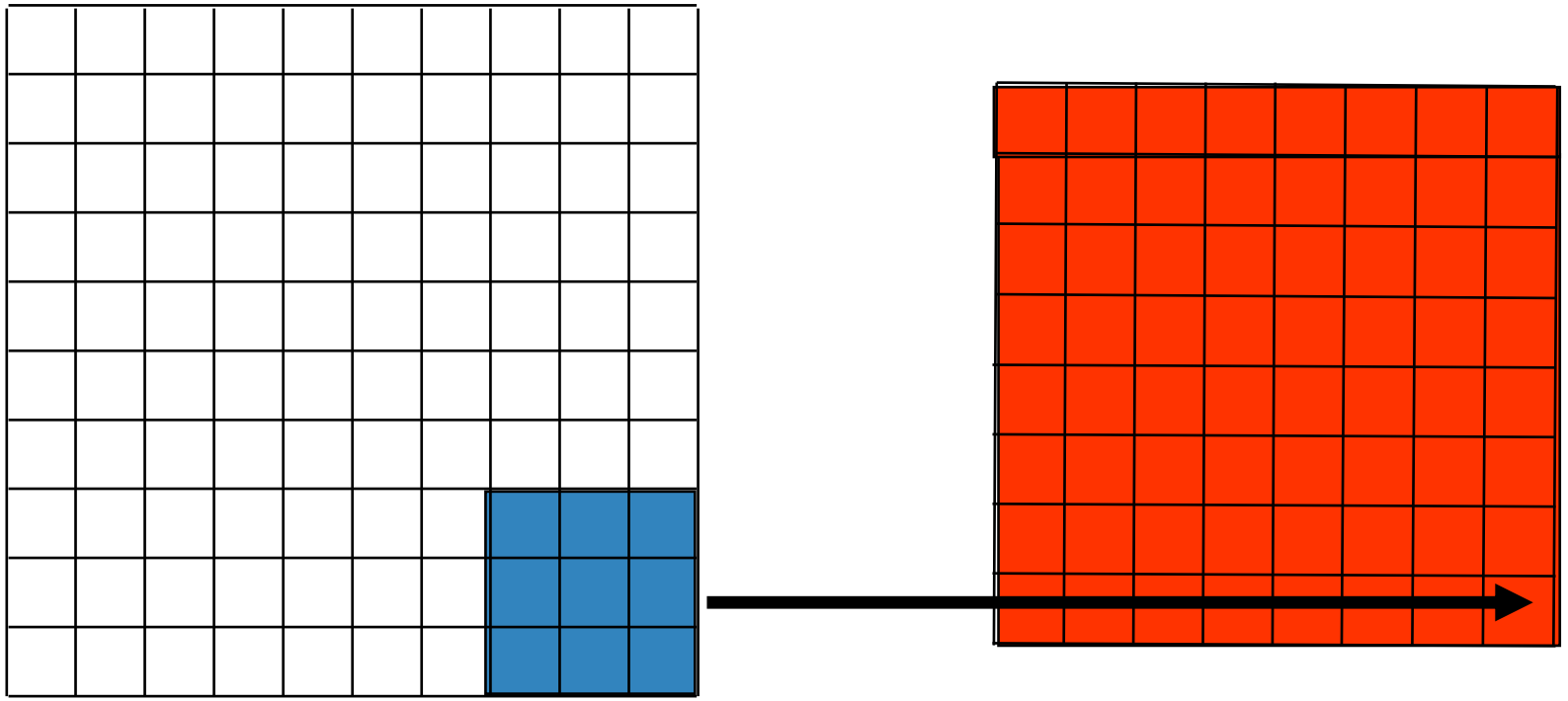
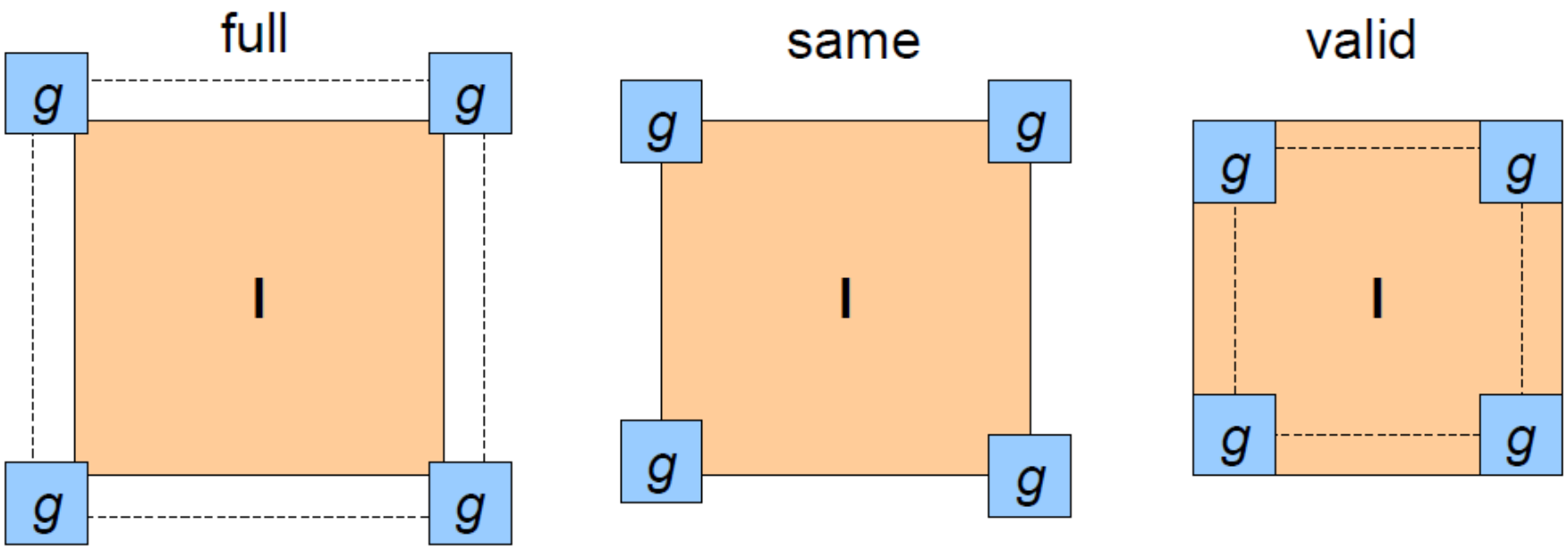


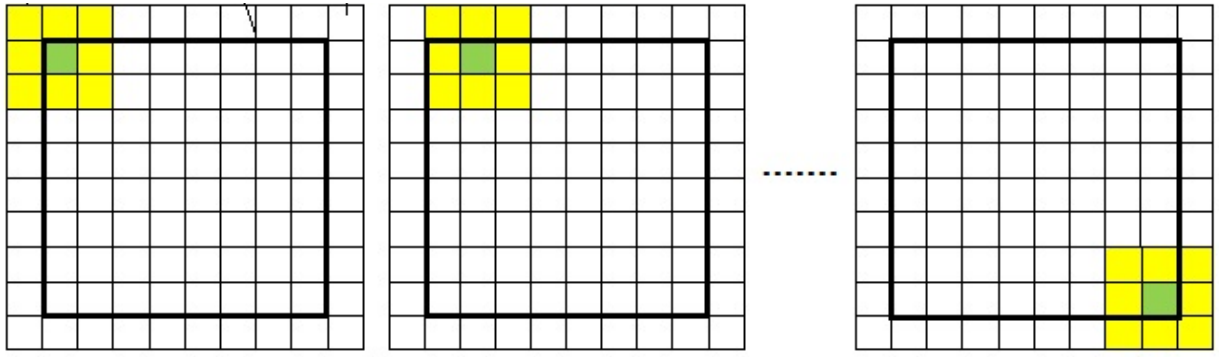
Image
Filtered Image

Output size



Dealing with Borders

- the filter window falls off the edge of the image
- need to extrapolate
- methods
  - clip filter (black): Pad with zeros
  - wrap around: Pad with circular repetition of elements within the dimension, Assume periodicity of the image
  - copy edge: Pad by repeating border elements of image
  - reflect across edge: Pad with mirror reflection of image along dimension



Examples

Original image

Noise

Average 3 x 3

Min 3 x 3

Max 3 x 3

Median 3 x 3

Problems

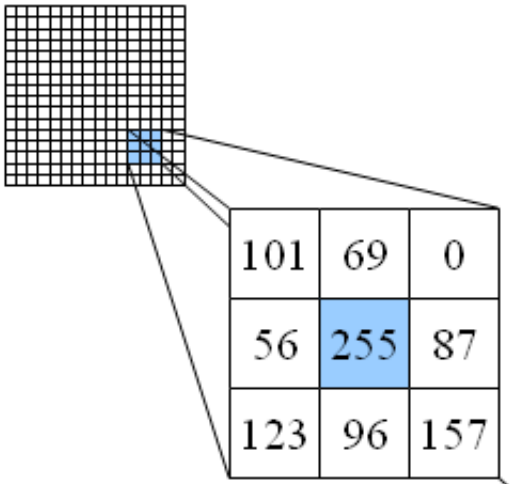
Median, 5pt

Mean, 5pt

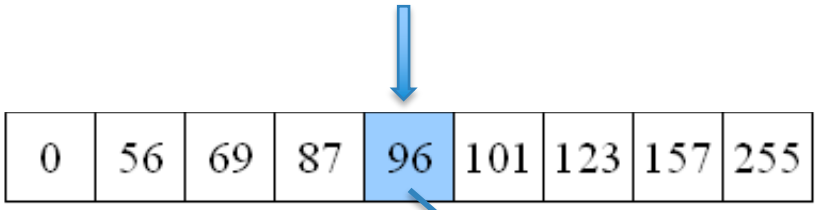
Min, 2pt

Max, 2pt

Median Filter

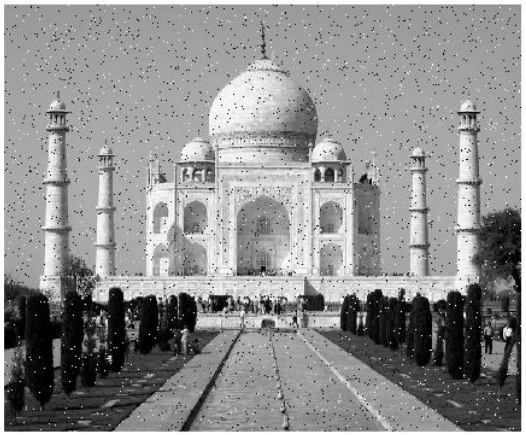


sort



new pixel value

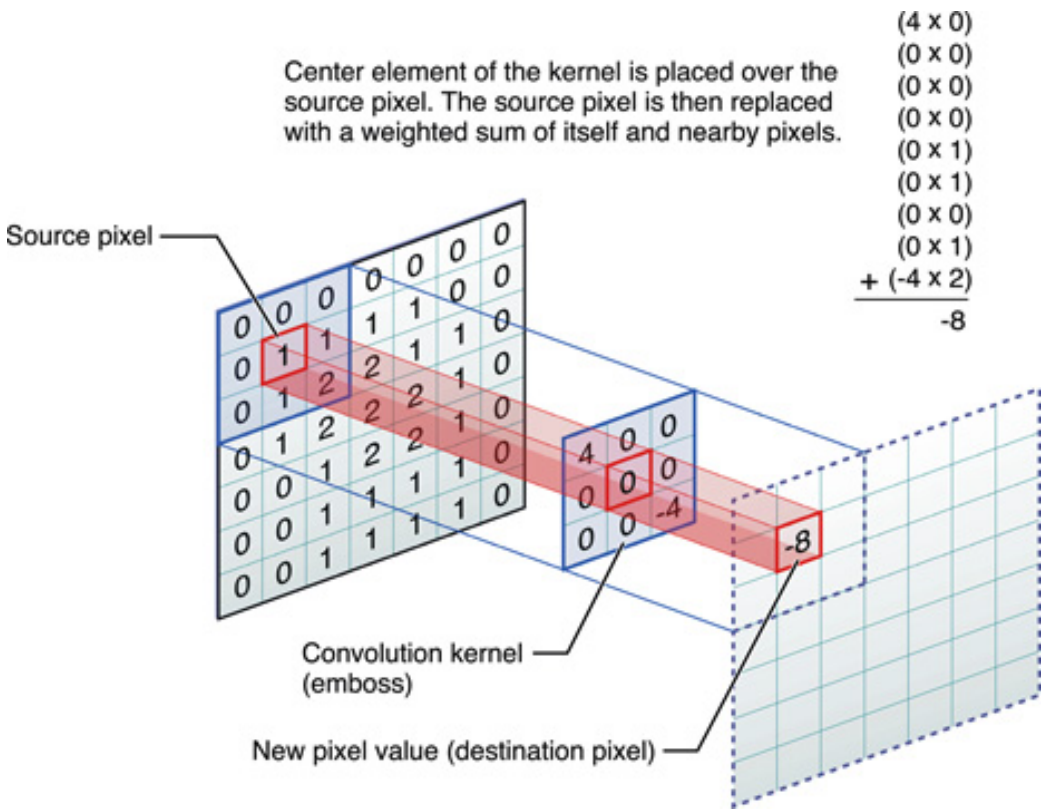
101	69	0
56	96	87
123	96	157



Convolution

$$I'(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(x+i, y+j) \cdot \text{filter}^{\text{flipped}}(i, j)$$

Use *odd sized* filters, symmetric index range [–N,N] so filter center at (0,0)

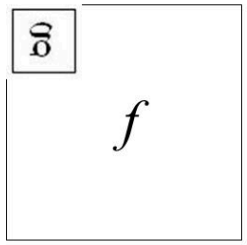


	-1	0	1
-1	-1	-2	-1
0	0	0	0
1	1	2	1

3 x 3 filter

(i.e. kernel)

Convention:
kernel is "flipped"

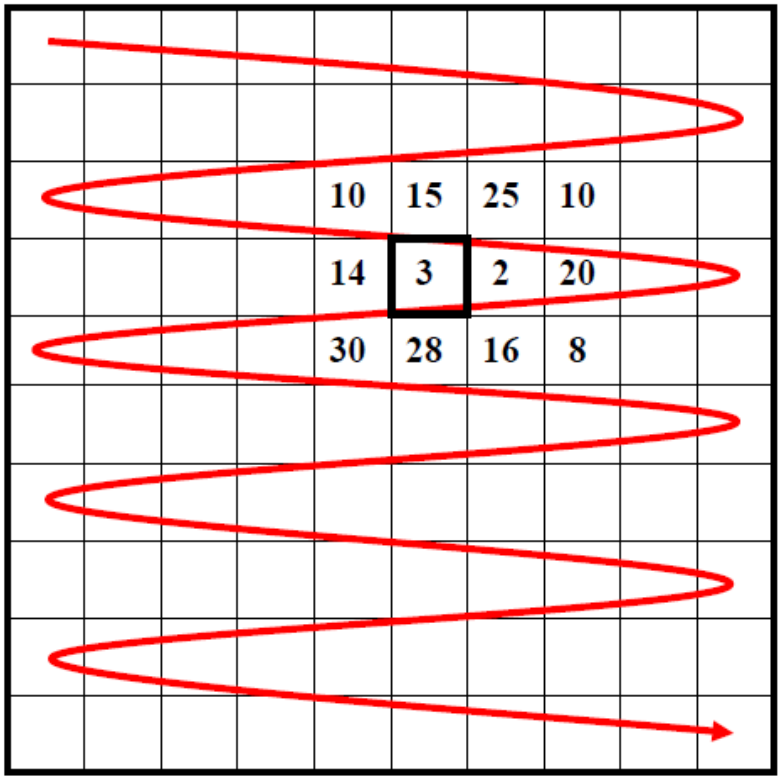


Average

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

Average 3 x 3 kernel

$\otimes$



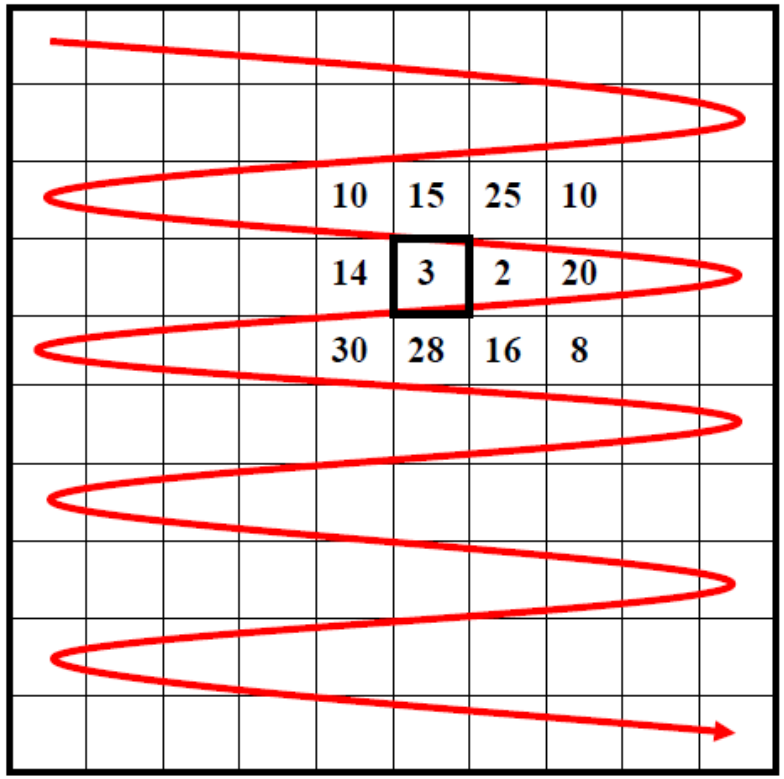
image

Average

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

Average 3 x 3 kernel

$\otimes$



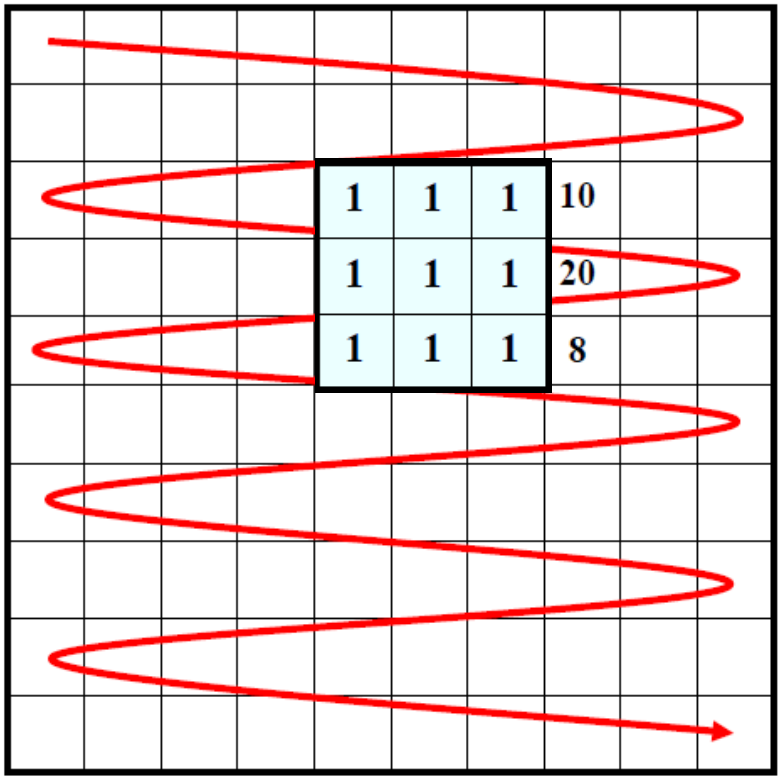
image

Average

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

Average 3 x 3 kernel

$\otimes$



image

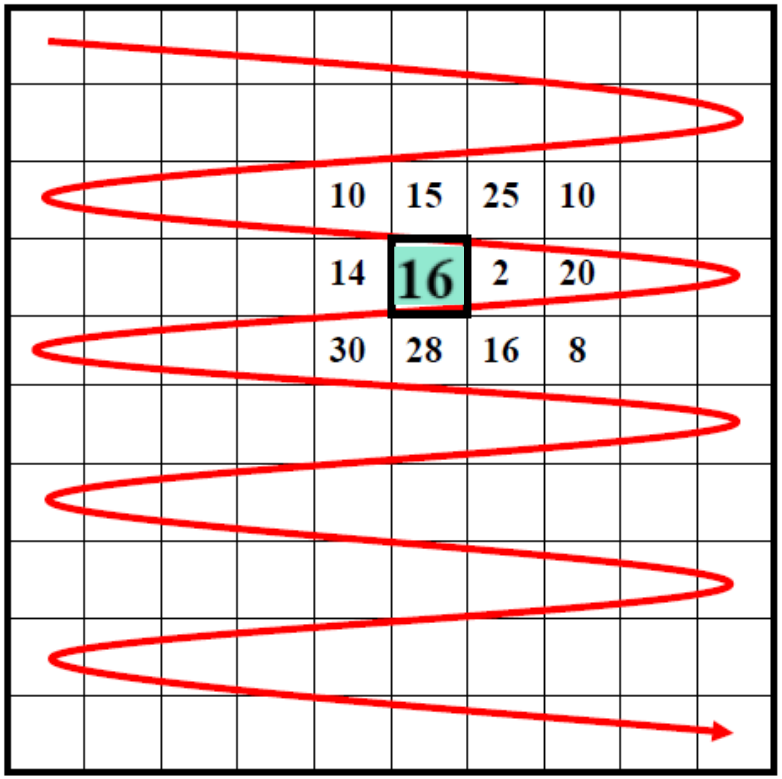
(10*1+15*1+25*1+14*1+3*1
+2*1+30*1+28*1+16*1) / 9
=143/9 ~ 16

Average

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

Average 3 x 3 kernel

$\otimes$



image

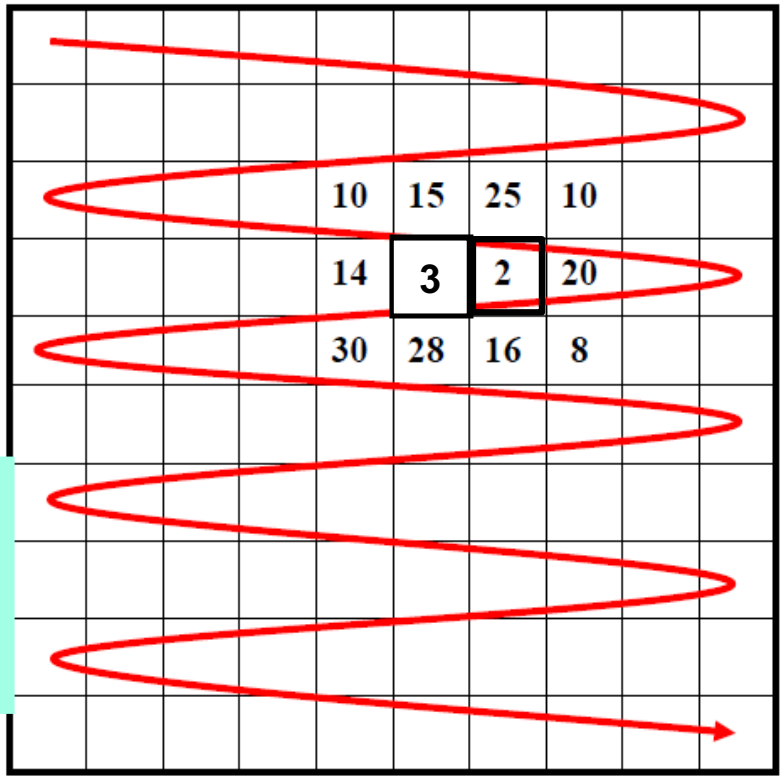
(10*1+15*1+25*1+14*1+3*1
+2*1+30*1+28*1+16*1) / 9
=143/9 ~ 16

Average

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

Average 3 x 3 kernel

$\otimes$



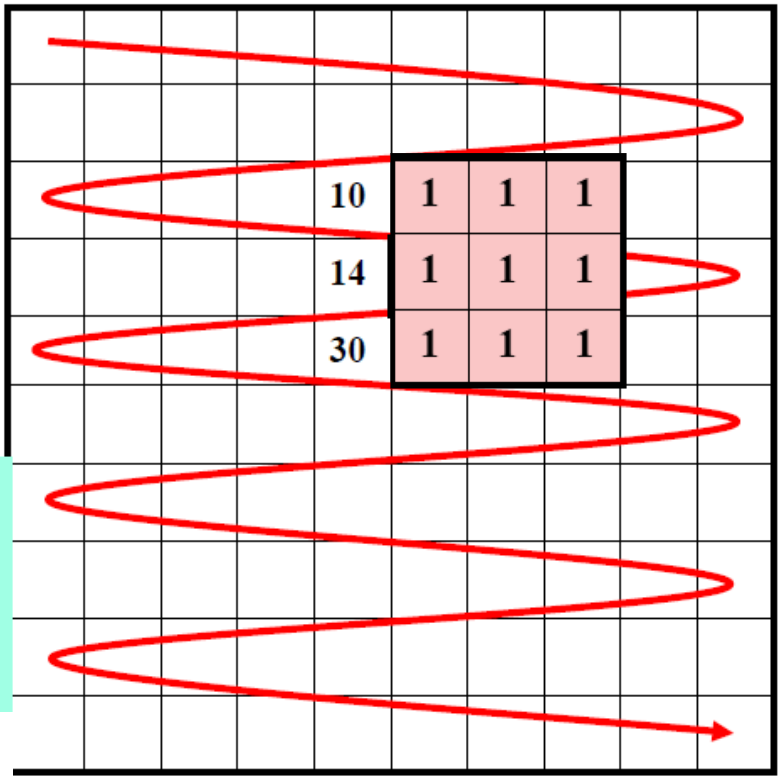
image

(10*1+15*1+25*1+14*1+3*1
+2*1+30*1+28*1+16*1) / 9
=143/9 ~ 16

Average

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

Average 3 x 3 kernel



image

(10*1+15*1+25*1+14*1+3*1 +2*1+30*1+28*1+16*1) / 9 =143/9 ~ 16

(15*1+25*1+10*1+3*1+2*1+ 20*1+28*1+16*1+8*1) / 9 =127/9 ~ 14

Main properties of convolution

- **Commutative**: $a * b = b * a$

  - no difference between filter and signal

- **Associative**: $a * (b * c) = (a * b) * c$

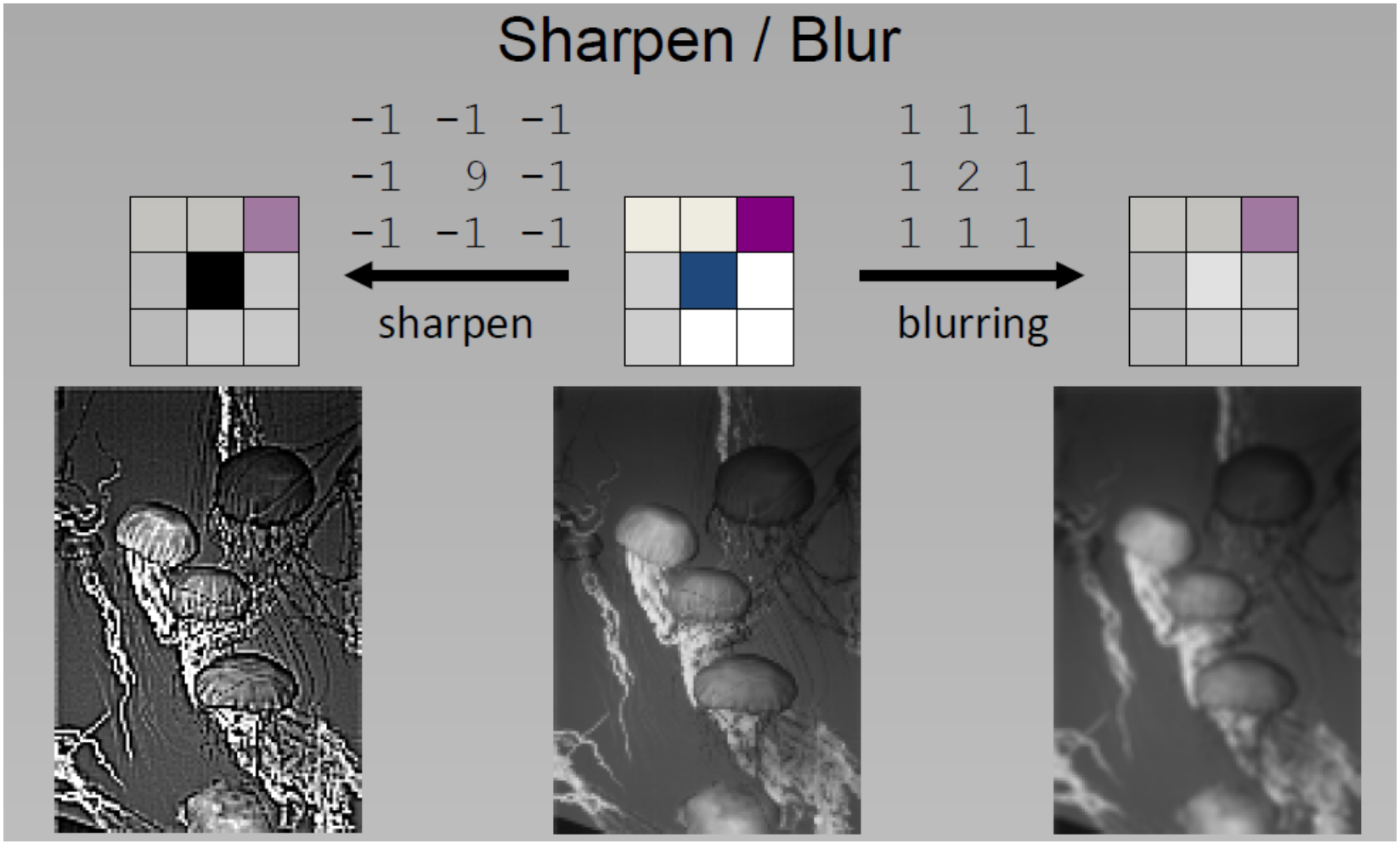
  - applying several filters one after another:

    $(((a * b1) * b2) * b3)$

is equivalent to applying one filter: $a * (b1 * b2 * b3)$

- **Distributive**: $a * (b + c) = (a * b) + (a * c)$

Enhancing vs Smoothing



Low-pass filters

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

$$\frac{1}{81}\times\begin{bmatrix}1&2&3&2&1\\2&4&6&4&2\\3&6&9&6&3\\2&4&6&4&2\\1&2&3&2&1\end{bmatrix}$$

pyramidal

$$\frac{1}{25}\times\begin{bmatrix}0&0&1&0&0\\0&2&2&2&0\\1&2&5&2&1\\0&2&2&2&0\\0&0&1&0&0\end{bmatrix}$$

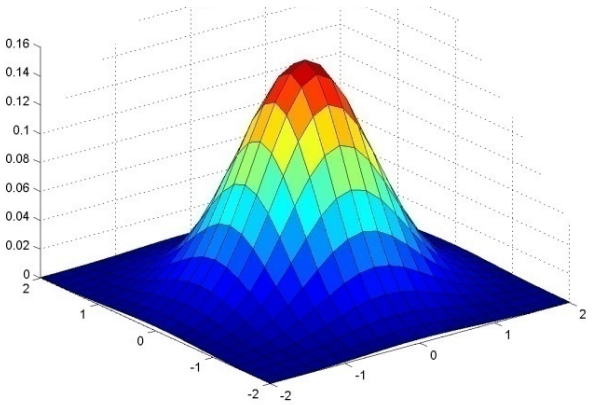
conical

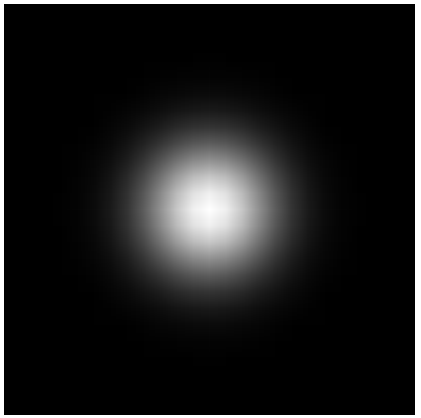
$$\frac{1}{273}\begin{bmatrix}1&4&7&4&1\\4&16&26&16&4\\7&26&41&26&7\\4&16&26&16&4\\1&4&7&4&1\end{bmatrix}$$

Gaussian

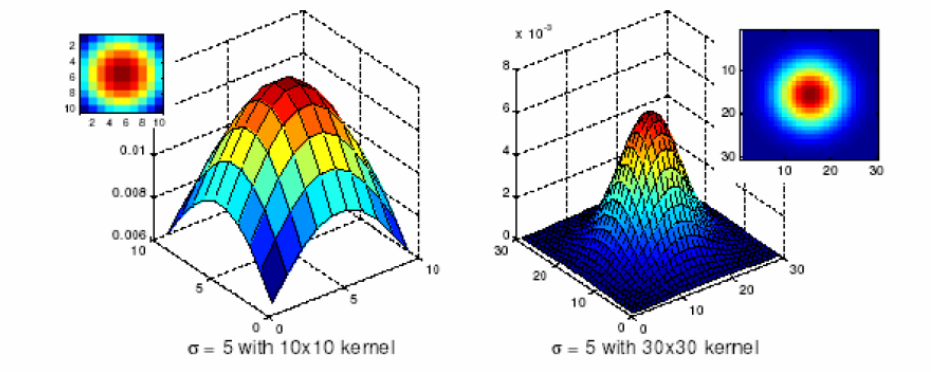
Gaussian Kernel

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$





- Gaussian filters have infinite support, but discrete filters use finite kernels

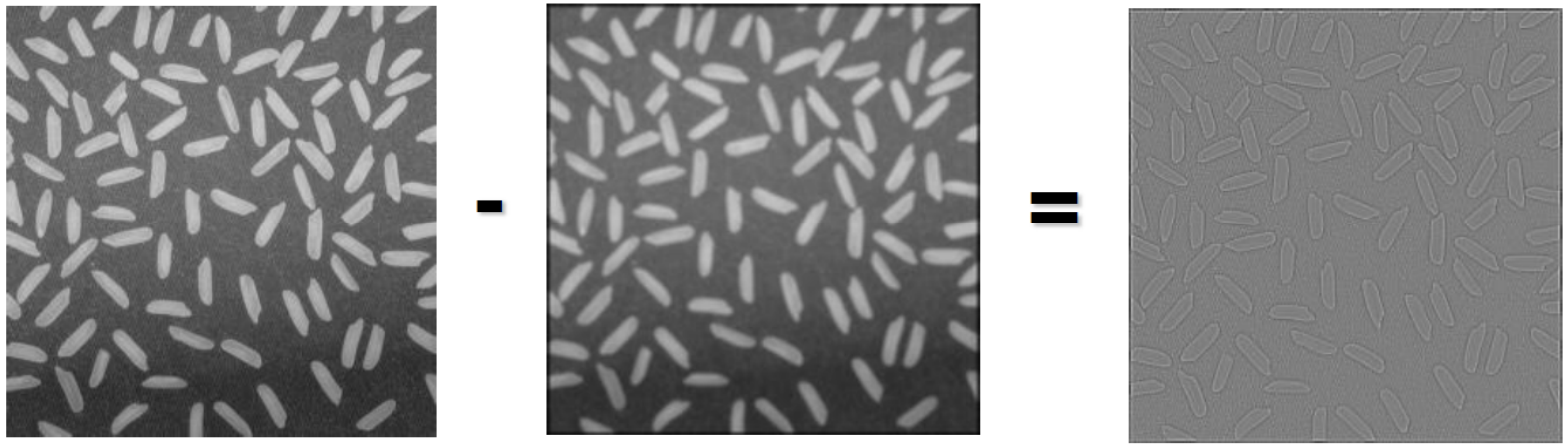


Rule of thumb: Use kernel with each side $5 \text{ or } 6\sigma$

Gaussian Blur

High-pass filters

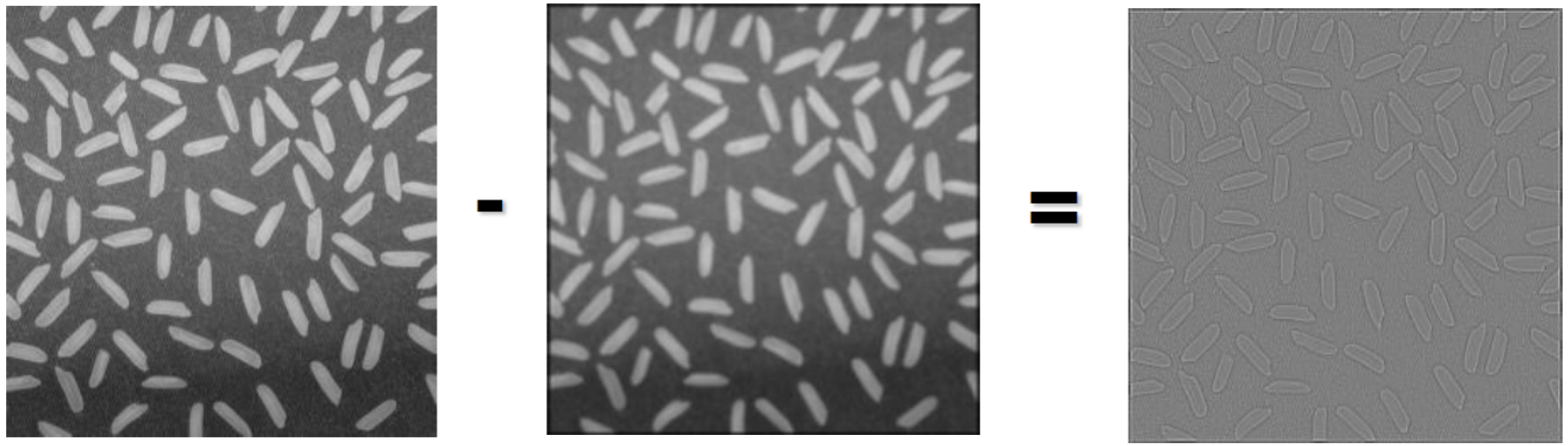
Example of edge detection using low-pass filter:
HighPassImage = SourceImage – LowPassImage



Source Image Low-pass High-pass

High-pass filters

Example of edge detection using low-pass filter:
HighPassImage = SourceImage – LowPassImage



0	0	0
0	1	0
0	0	0

Source Image

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Low-pass

$\frac{1}{9}$

-1	-1	-1
-1	8	-1
-1	-1	-1

High-pass

–

=

Let’s add it back:

+

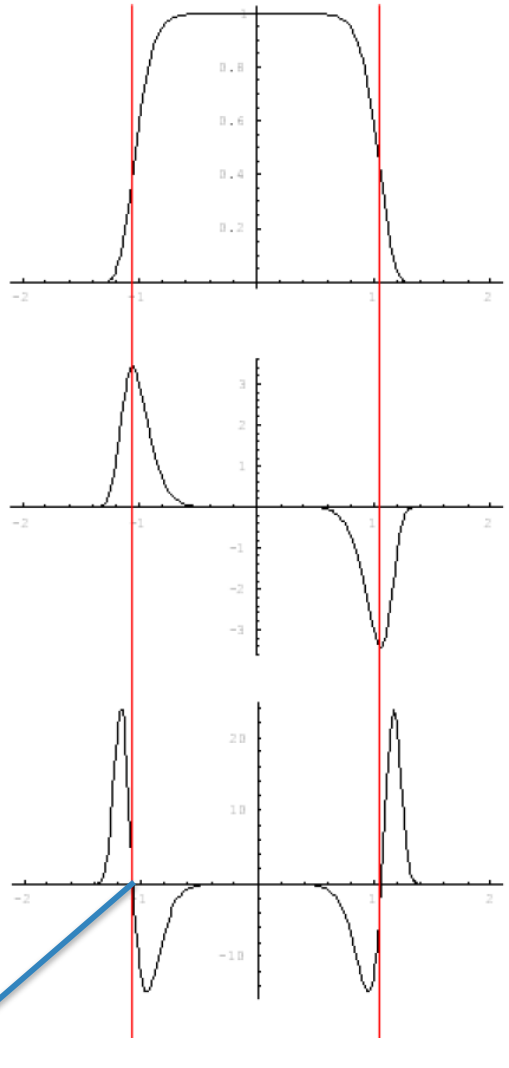
=

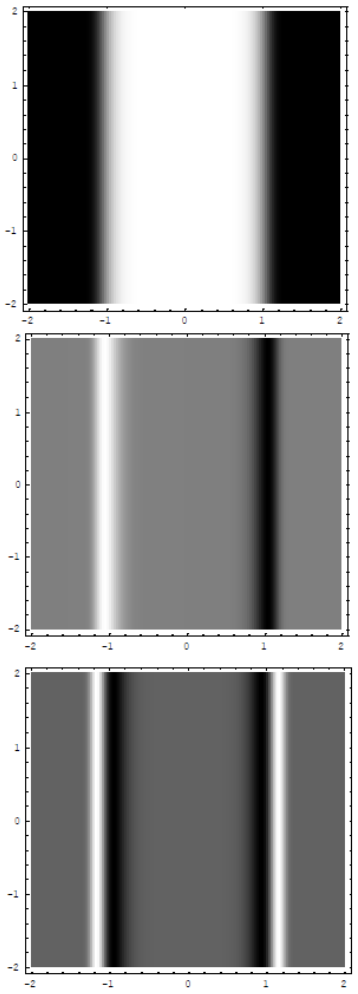
Signal Derivatives

$$I(x,y) = e^{-x^{10}/2}$$

$$\frac{\partial I}{\partial x}$$

$$\frac{\partial^2 I}{\partial x^2}$$





1st

2nd

Zero crossing

Gradient

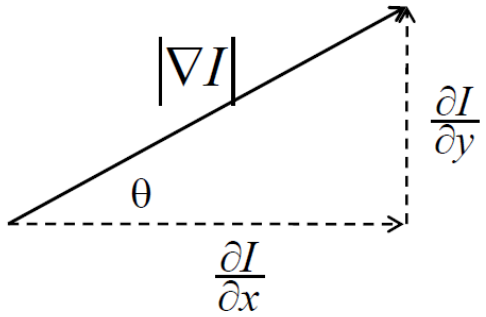
$$\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

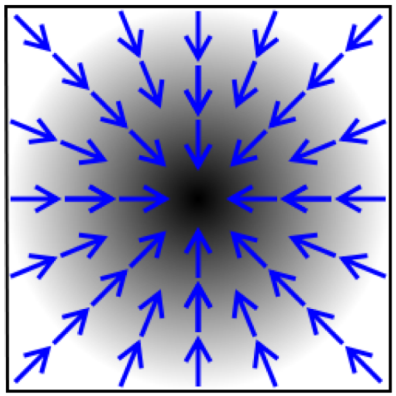
Magnitude

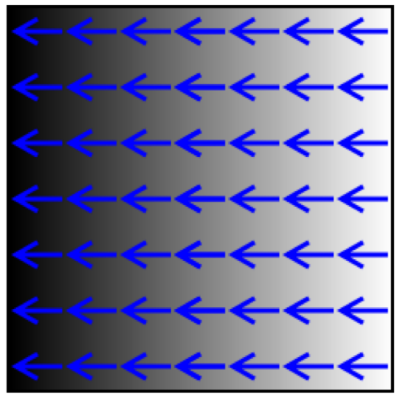
$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \approx \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|$$

Direction

$$\theta = \tan^{-1} \frac{\left(\frac{\partial I}{\partial y}\right)}{\left(\frac{\partial I}{\partial x}\right)}$$



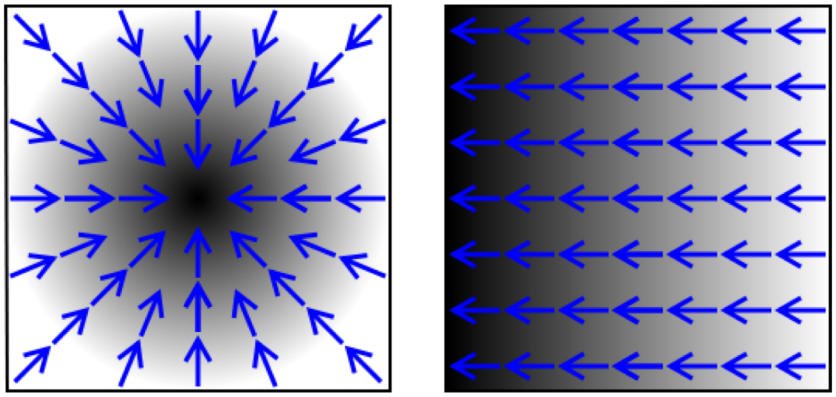




Gradient

Discrete approximation

$$\frac{\partial I}{\partial x} \approx \frac{I(x+1, y) - I(x-1, y)}{2}$$



Gradient

Image

I_1	I_2	I_3
I_4	I_5	I_6
I_7	I_8	I_9

Filter

0	-1	0
0	0	0
0	1	0

0	0	0
-1	0	1
0	0	0

$$\frac{\partial I}{\partial y} = [I_8 - I_2]$$

$$\frac{\partial I}{\partial x} = [I_6 - I_4]$$

Filters

Prewitt:

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Sobel:

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

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

Prof. Judith M.S. Prewitt

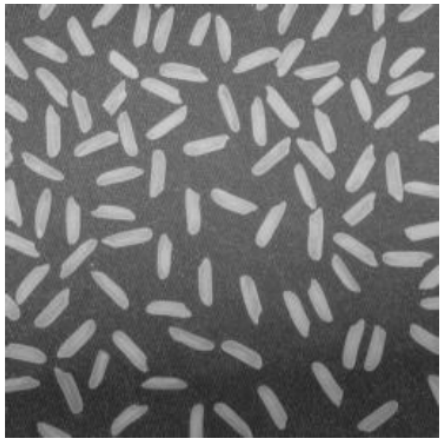
Mathematical Methods Applied to Image Processing in Medicine

Chapter Dec 1980

J. M. S. Prewitt

Image processing is a mathematical and engineering discipline which has as its domain digitized pictures, that is, pictures which have been spatially sampled and photometrically quantitized by an electro-optical scanning device and have been recorded on a medium such as magnetic tape, magnetic disc or computer memory core. Each spatially discrete image element is called a pixel and with each pixel is associated a photometric quantity such as brightness, transmittance, extinction or optical density. Processing refers to mathematical and logical operations performed on these image pixels in order to extract desired information. Medical images of interest arise from blood smears showing white cells and showing red cells, Papanicolaou cervical smears, radiographs, electron micrographs, scintigrams, etc.

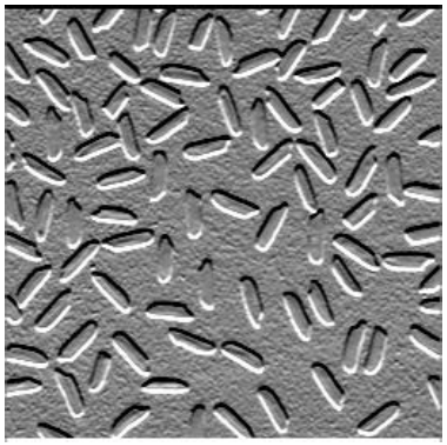
Sobel



$$|\nabla I| = \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|$$

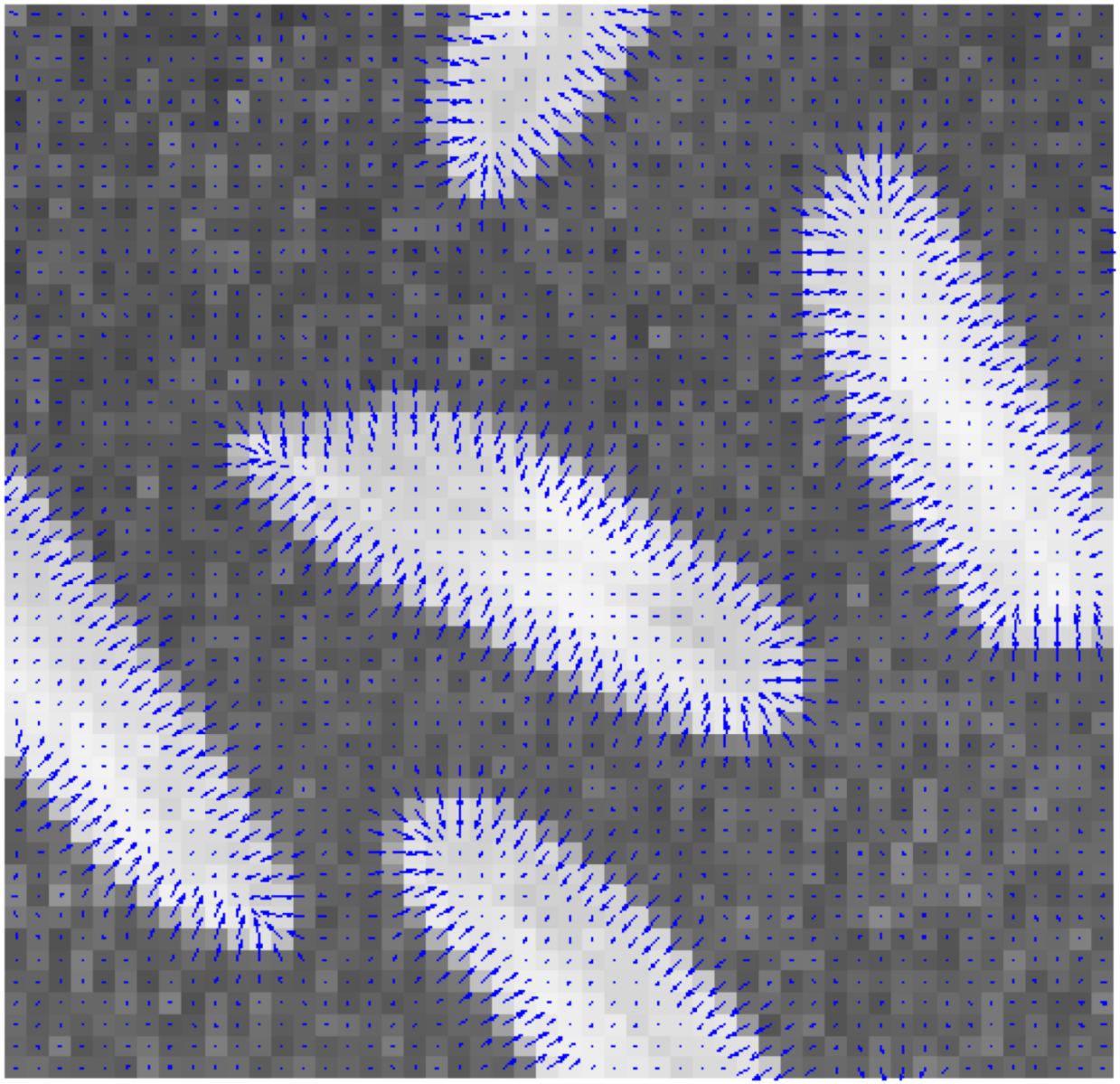
$$\frac{\partial I}{\partial x}$$

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



$$\frac{\partial I}{\partial y}$$

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



Laplacian: 2nd derivative

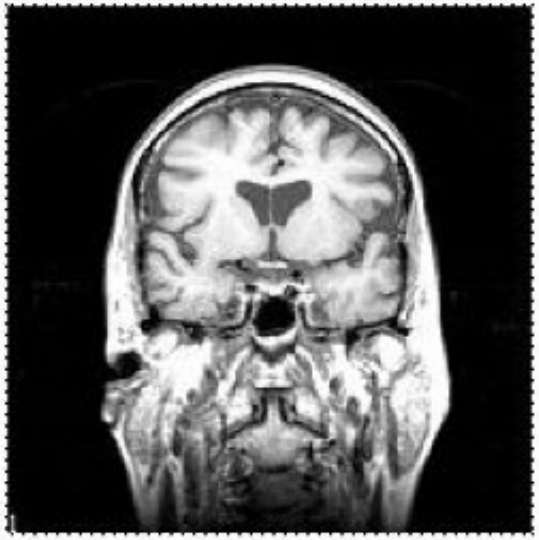
$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Discrete:

0	-1	0
-1	4	-1
0	-1	0

Laplacian

Laplacian: 2nd derivative

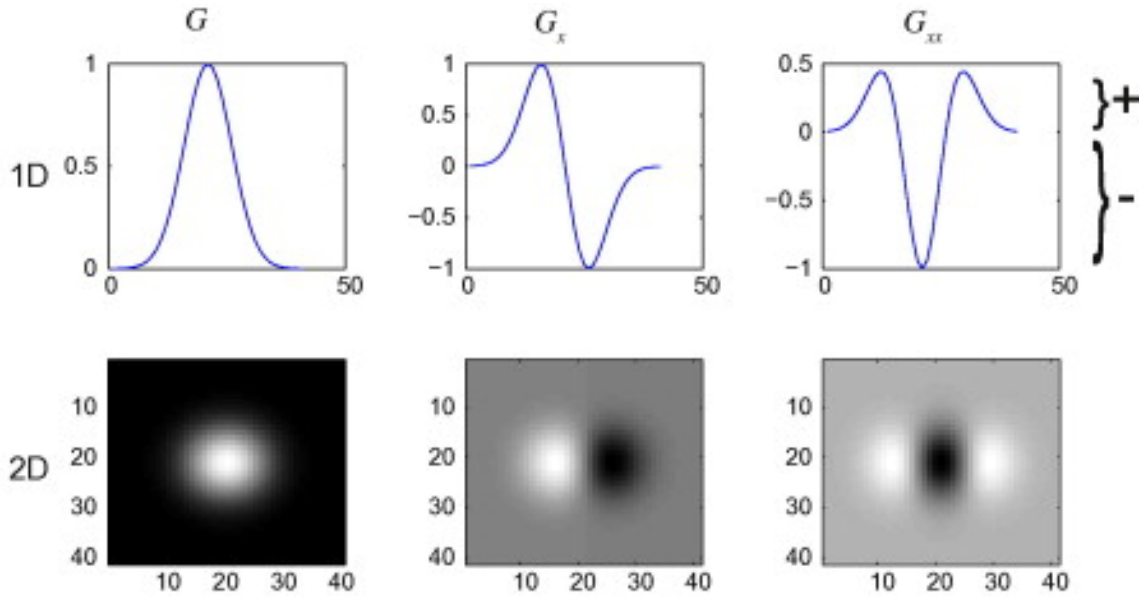


(A) Original MR image

(B) Laplacian results

(C) Extraction of the zero crossing of the Laplacian (object edges)

Gaussian Derivatives: 1D vs 2D



Along x

prewitt

laplacian

Separability

- A two-dimensional filter kernel is ***separable*** if it can be expressed as the outer product of two vectors.
- Example: Sobel kernel

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

Separable Convolution

- **Associativity of convolution:**

$$f * (g * h) = (f * g) * h$$

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

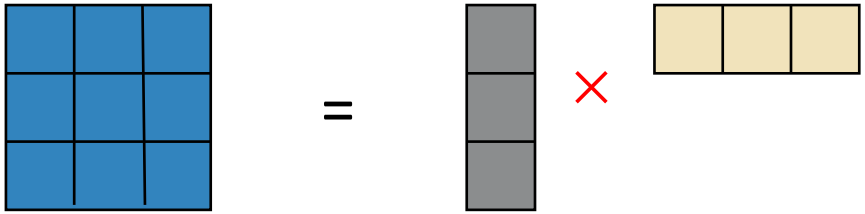
K = g * h

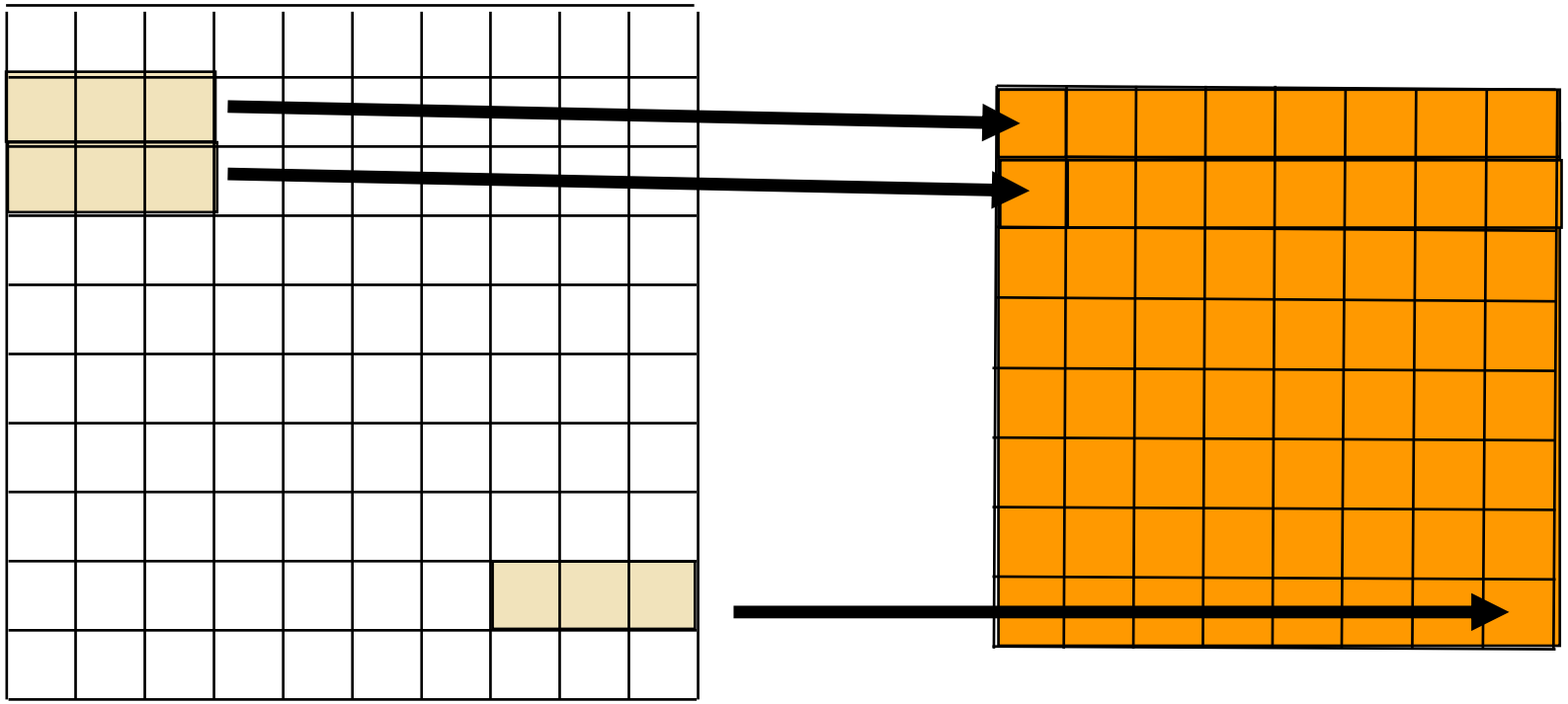
$$Dest[i,j] = \sum_{l=0}^{Kh} \sum_{k=0}^{Kw} Kernel[k,l]\, Source[i+k, j+l]$$

$$Dest[i,j] = \sum_{l=0}^{Kh} Vert[l] \sum_{k=0}^{Kw} Horiz[k]\, Source[i+k, l+j]$$

The Gaussian kernel is separable

Separable Filtering

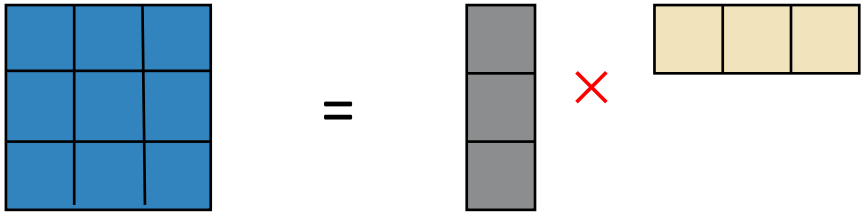


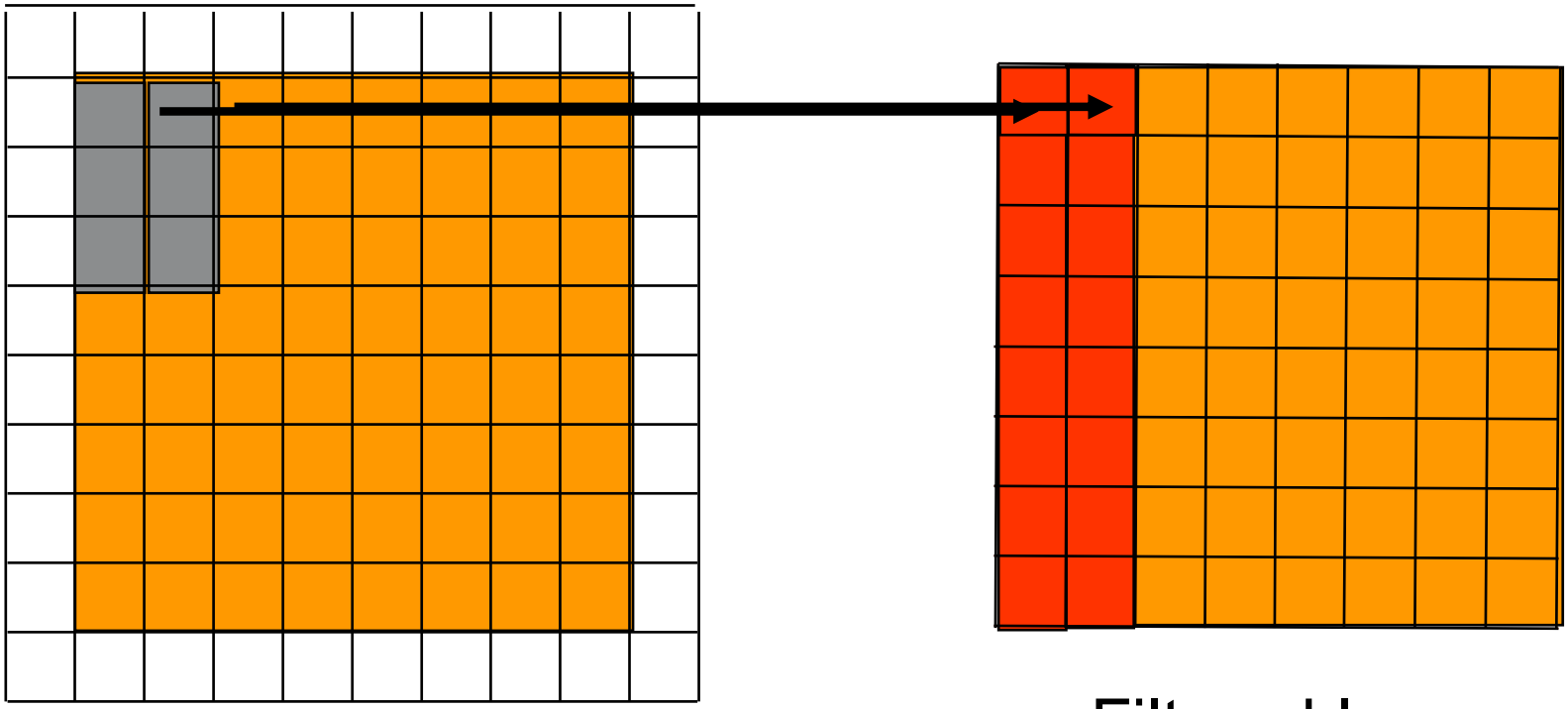


Image

Filtered Image

Separable Filtering





Image

Filtered Image

Separable Filtering

What is the number of operations for the 3×5 kernel H?
Answer: $15wh$

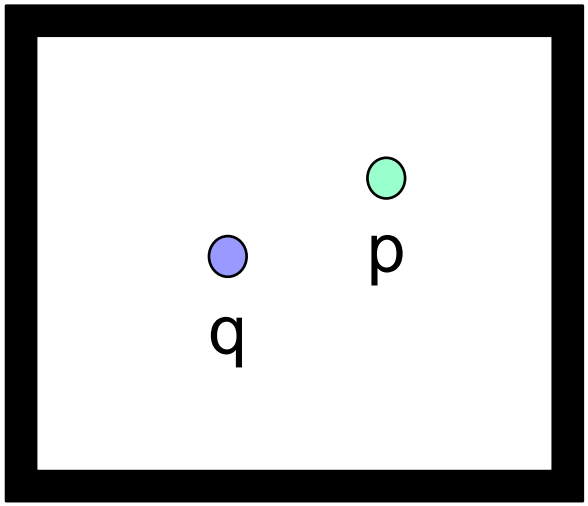
What is the number of operations for H_x followed by H_y?
Answer: $3wh + 5wh = 8wh$

What about the case of a $M \times M$ kernel?
Answer:
$O(M^2)$ – no separability (M^2wh operations)
$O(M)$ – with separability ($2Mwh$ operations)

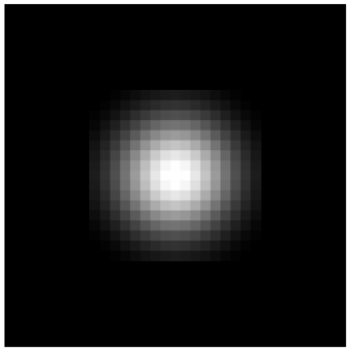
Properties of Gaussian Filter

Same idea: **weighted average of pixels**.

$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q} \in S} G_{\sigma}(\|\mathbf{p}-\mathbf{q}\|) I_{\mathbf{q}}$$



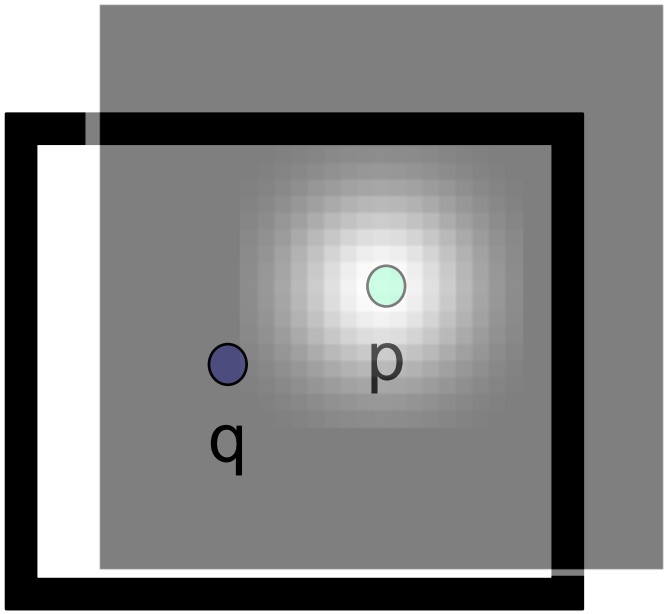
normalized
Gaussian function



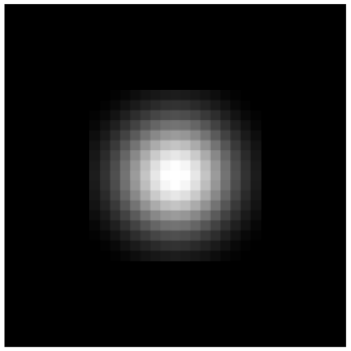
Properties of Gaussian Filter

Same idea: **weighted average of pixels**.

$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q} \in S} G_{\sigma}(\|\mathbf{p}-\mathbf{q}\|) I_{\mathbf{q}}$$



normalized
Gaussian function



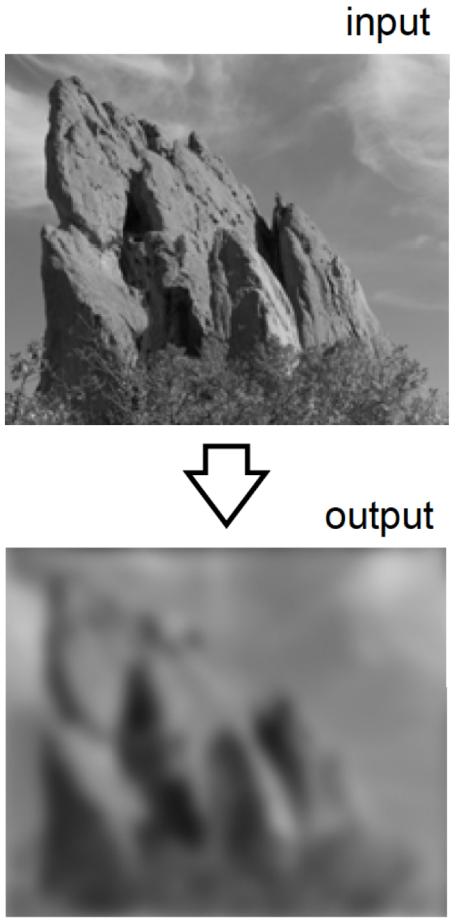
S

Properties of Gaussian Filter

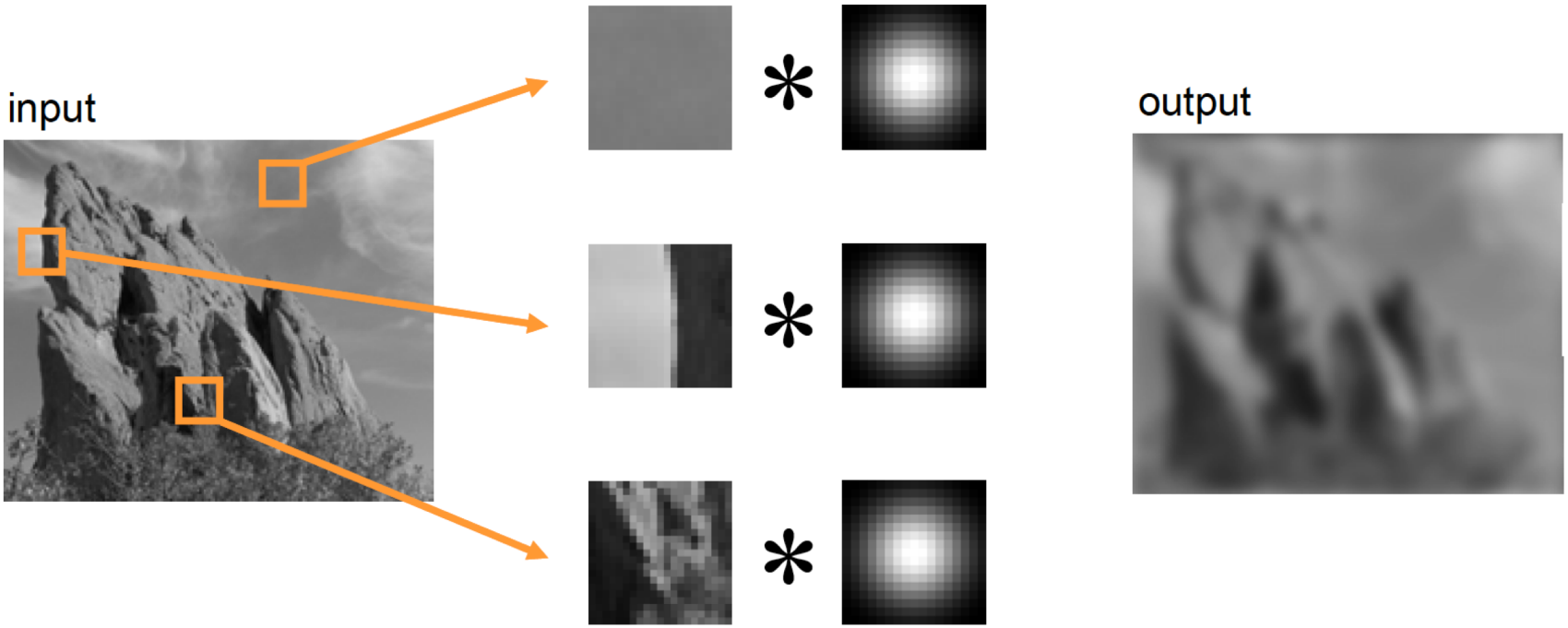
- Does smooth images
- But smoothes too much: **edges are blurred.**
  - Only spatial distance matters
  - No edge term

$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q} \in S} G_{\sigma}(\|\mathbf{p}-\mathbf{q}\|) I_{\mathbf{q}}$$

space

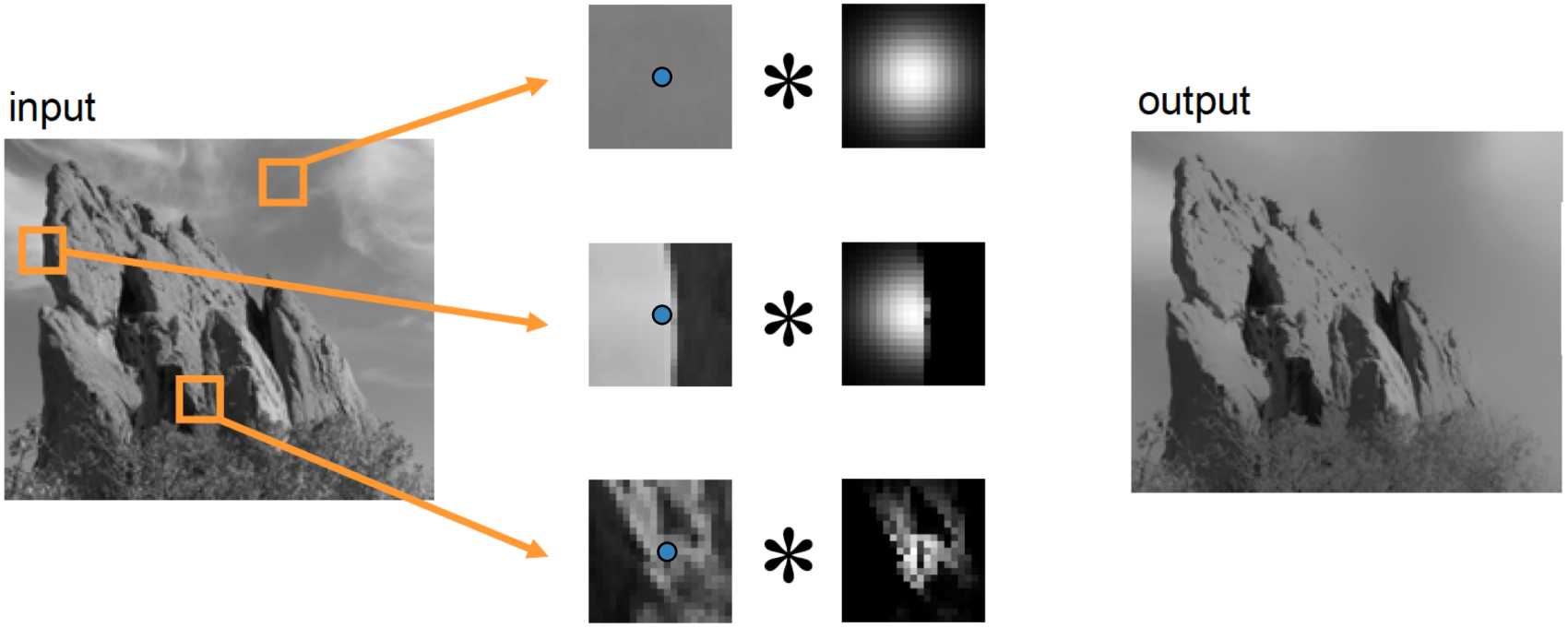


Properties of Gaussian Filter



Same Gaussian kernel everywhere.

Bilateral Filter



The kernel shape depends on the image content.

Bilateral Filtering

Bilateral Filtering

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)\, I_{\mathbf{q}}$$

new

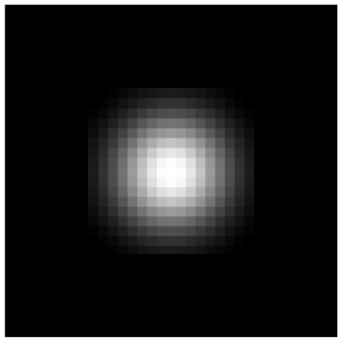
not new

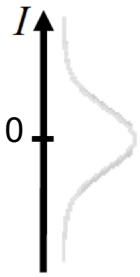
new

normalization factor

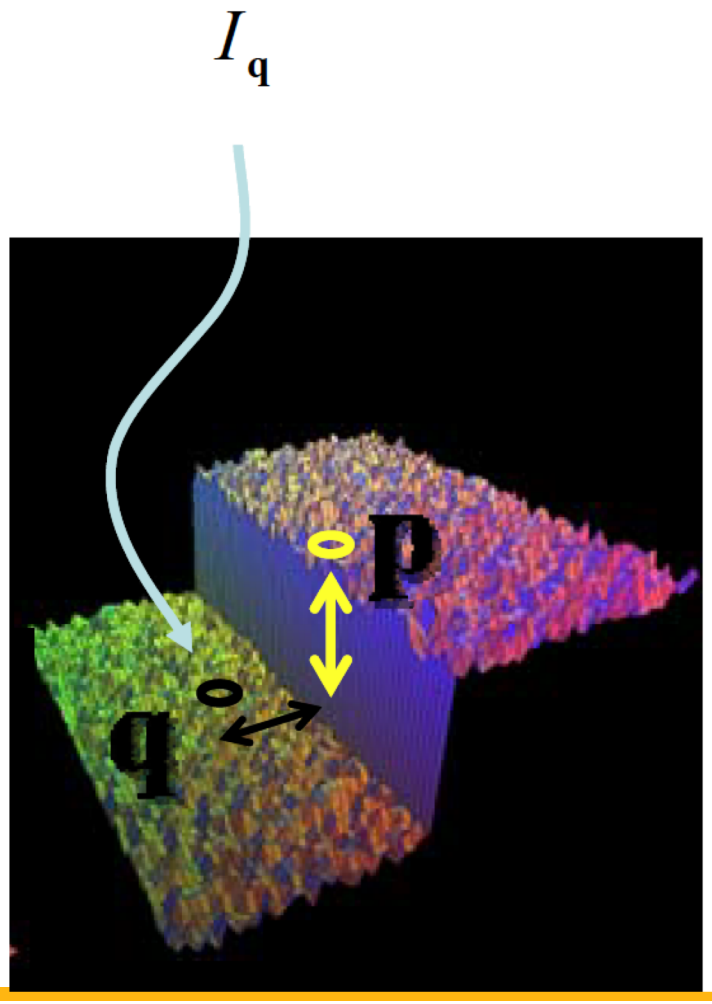
space weight

range weight



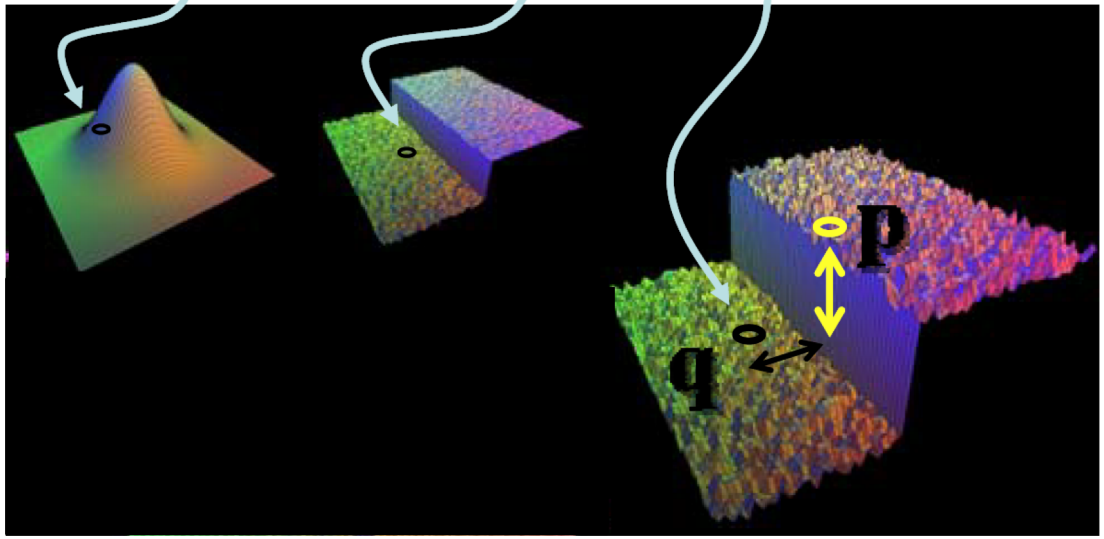


Bilateral Filtering



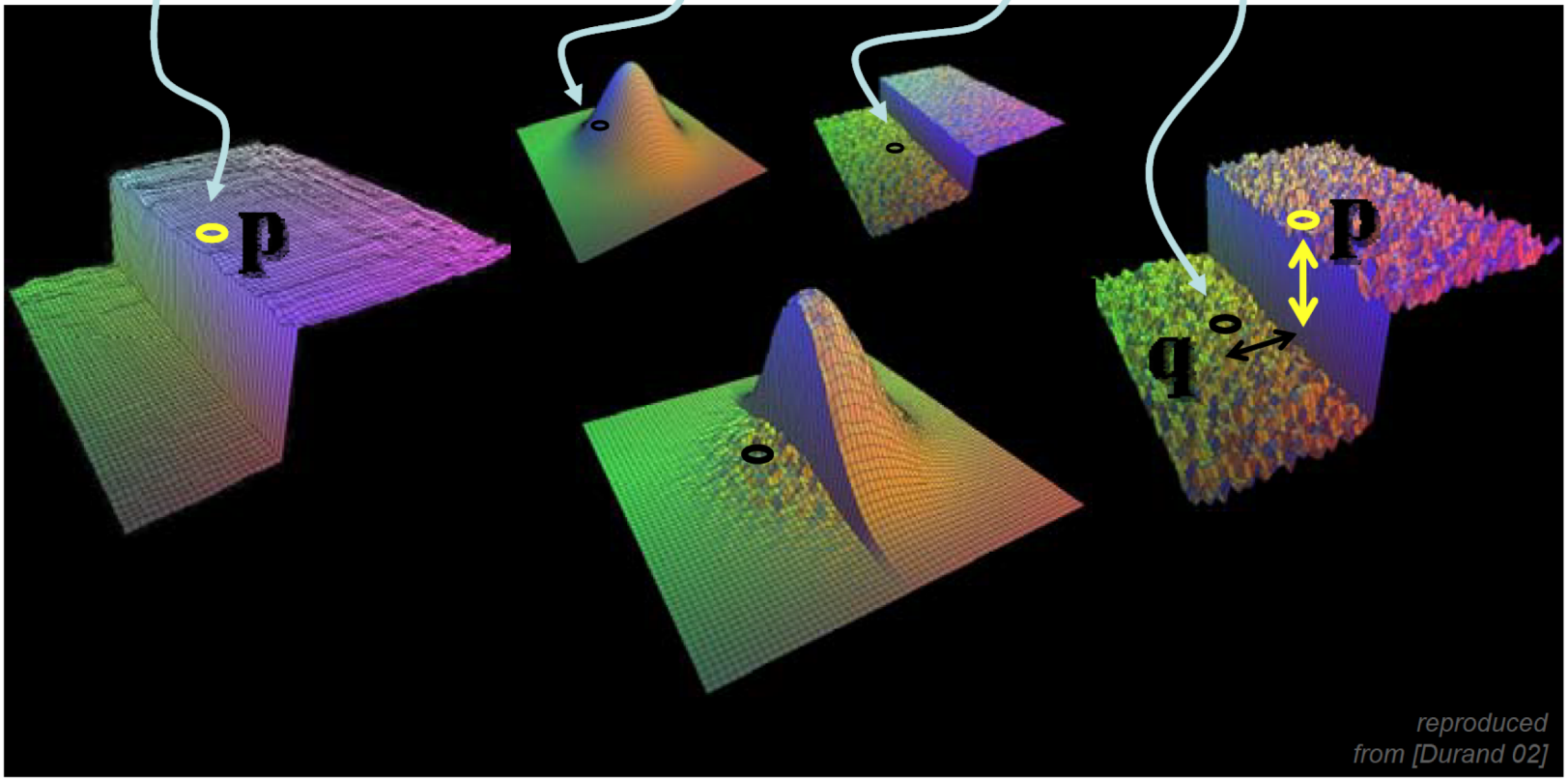
Bilateral Filtering

$$\sum_{S} G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)\; I_{\mathbf{q}}$$



Bilateral Filtering

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\| \mathbf{p} - \mathbf{q} \|) \; G_{\sigma_r}(| I_{\mathbf{p}} - I_{\mathbf{q}} |) \; I_{\mathbf{q}}$$



reproduced from [Durand 02]

Bilateral filtering: parameters

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\| \mathbf{p} - \mathbf{q} \|) G_{\sigma_r}(| I_{\mathbf{p}} - I_{\mathbf{q}} |) I_{\mathbf{q}}$$

- space σ_s : spatial extent of the kernel, size of the considered neighborhood.

- range σ_r : “minimum” amplitude of an edge

Bilateral filtering: parameters