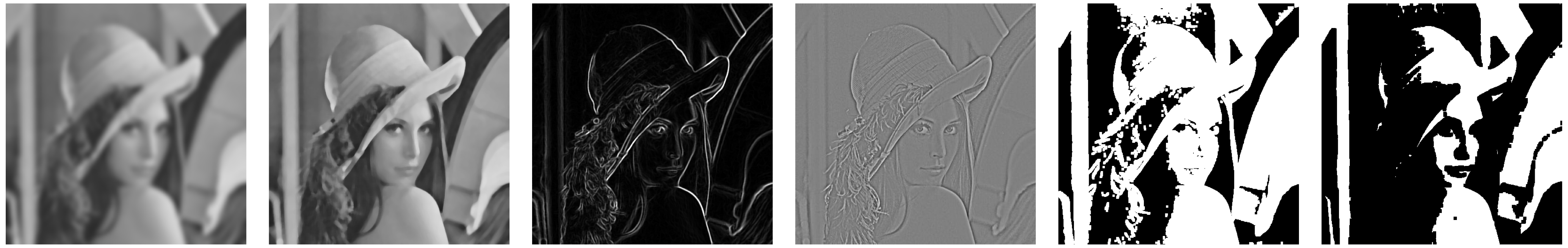
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 4: Image Filtering (continued)

(unless otherwise stated slides are taken or adopted from **Bob Woodham, Jim Little** and **Fred Tung**)

Menu for Today (**September 16, 2020**)

Topics: Image Filtering (also topic for next two classes)

— **Linear** filters
— **Correlation** / **Convolution**
— Filter **examples**: Box, Gaussian

Redings:

— **Today's** Lecture: Forsyth & Ponce (2nd ed.) 4.1, 4.5
— **Next** Lecture: none

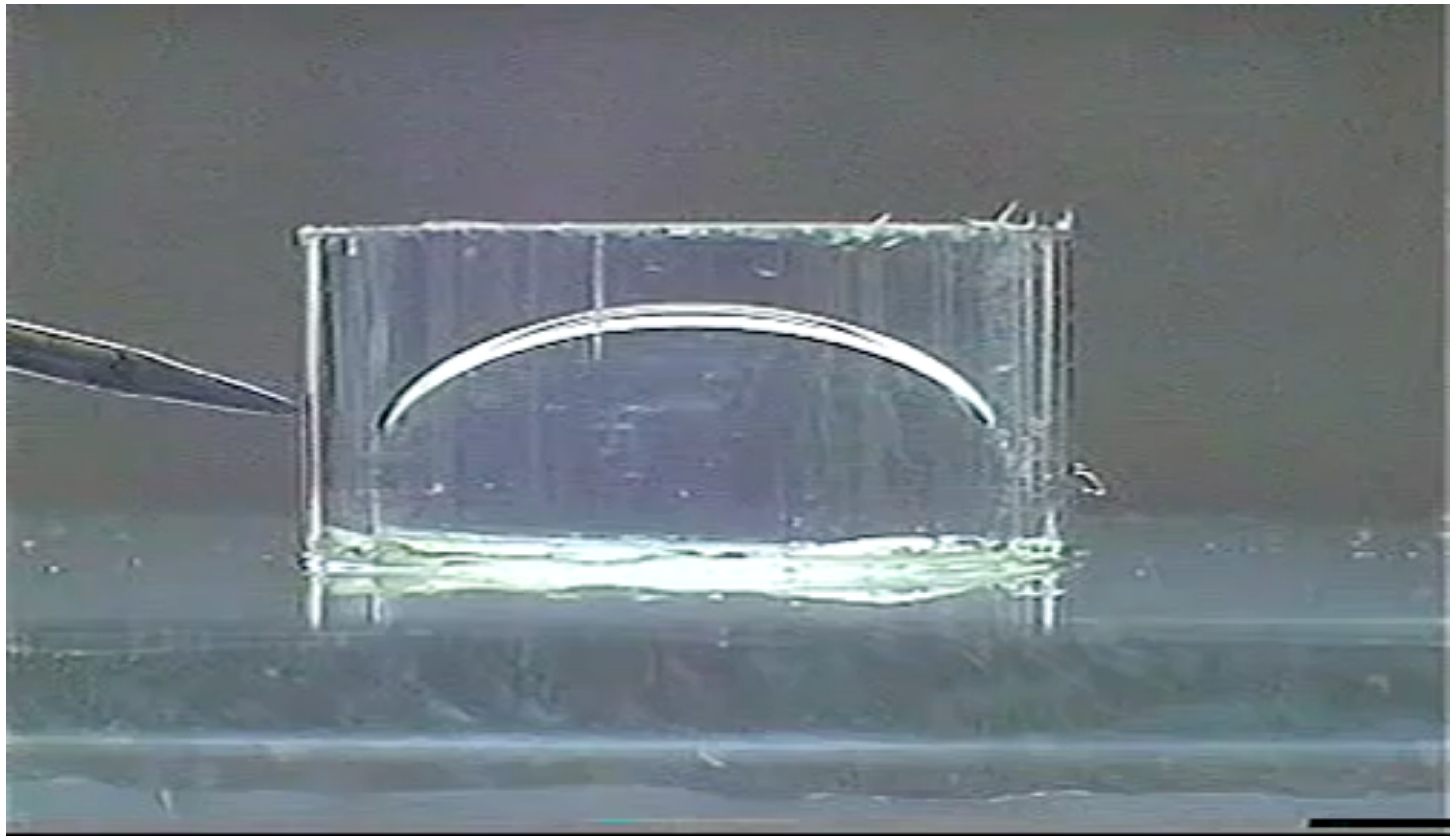
Reminders:

— **Assignment 1:** Image Filtering and Hybrid Images is out, due **September 30th**
— **Midterm** is scheduled for Week 7, **October 21st**

Today's "**fun**" Example:

Developed by the French company **Varioptic**, the lenses consist of an oil-based and a water-based fluid sandwiched between glass discs. Electric charge causes the boundary between oil and water to change shape, altering the lens geometry and therefore the lens focal length

The intended applications are: **auto-focus** and **image stabilization**. No moving parts. Fast response. Minimal power consumption.

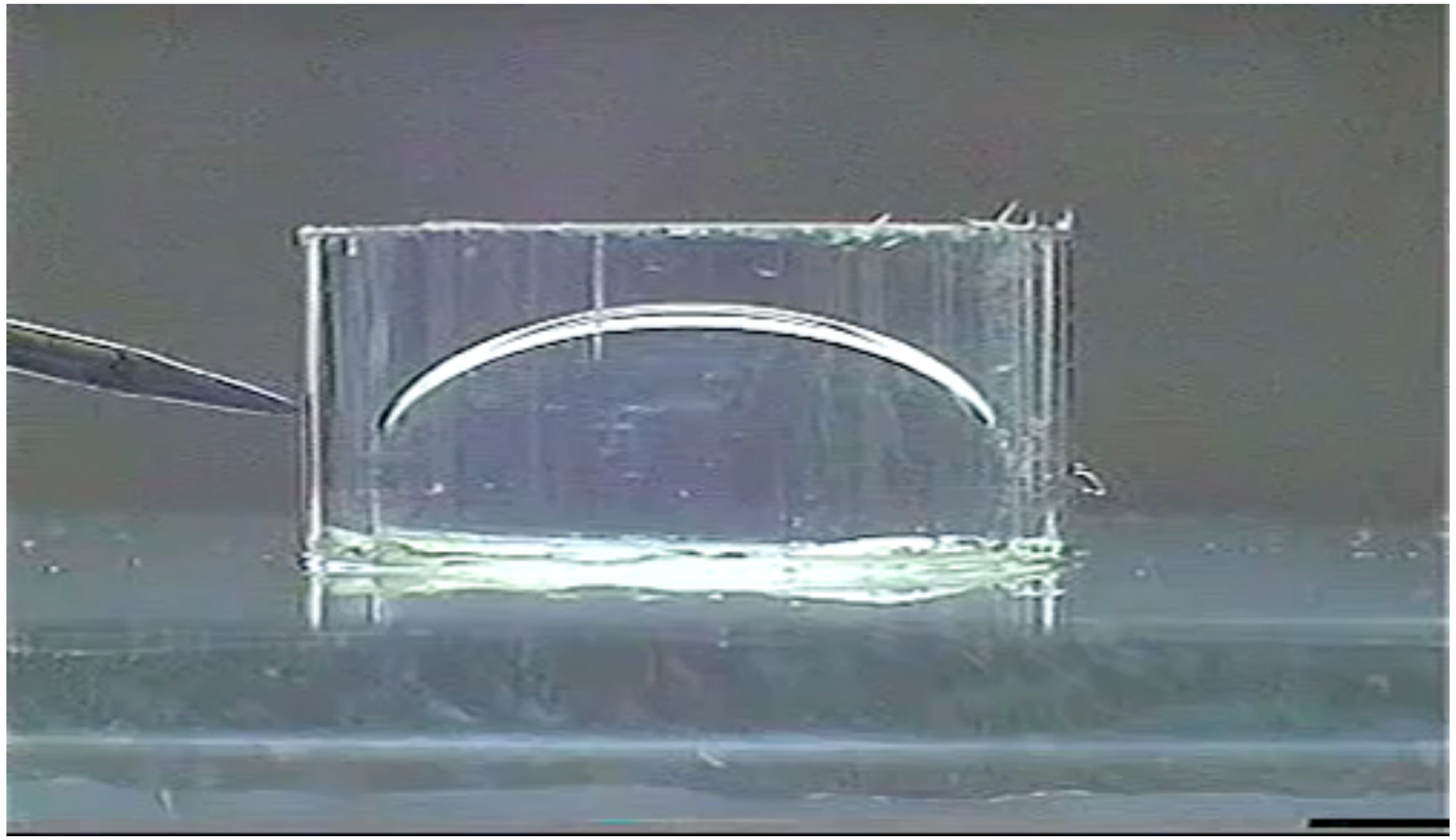


Video Source: https://www.youtube.com/watch?v=2c6lCdDFOY8

Today's "**fun**" Example:

Developed by the French company **Varioptic**, the lenses consist of an oil-based and a water-based fluid sandwiched between glass discs. Electric charge causes the boundary between oil and water to change shape, altering the lens geometry and therefore the lens focal length

The intended applications are: **auto-focus** and **image stabilization**. No moving parts. Fast response. Minimal power consumption.



Video Source: https://www.youtube.com/watch?v=2c6lCdDFOY8

Today’s “**fun**” Example:

Electrostatic field between the column of water and the electron (other side of power supply attached to the pipe) — see full video for complete explanation

Video Source: https://www.youtube.com/watch?v=NjLJ77IuBdM

Today's "**fun**" Example:

Electrostatic field between the column of water and the electron (other side of power supply attached to the pipe) — see full video for complete explanation

Video Source: https://www.youtube.com/watch?v=NjLJ77IuBdM

Today's "**fun**" Example:

As one example, in 2010, **Cognex** signed a license agreement with Varioptic to add auto-focus capability to it DataMan line of industrial ID readers (press release May 29, 2012)

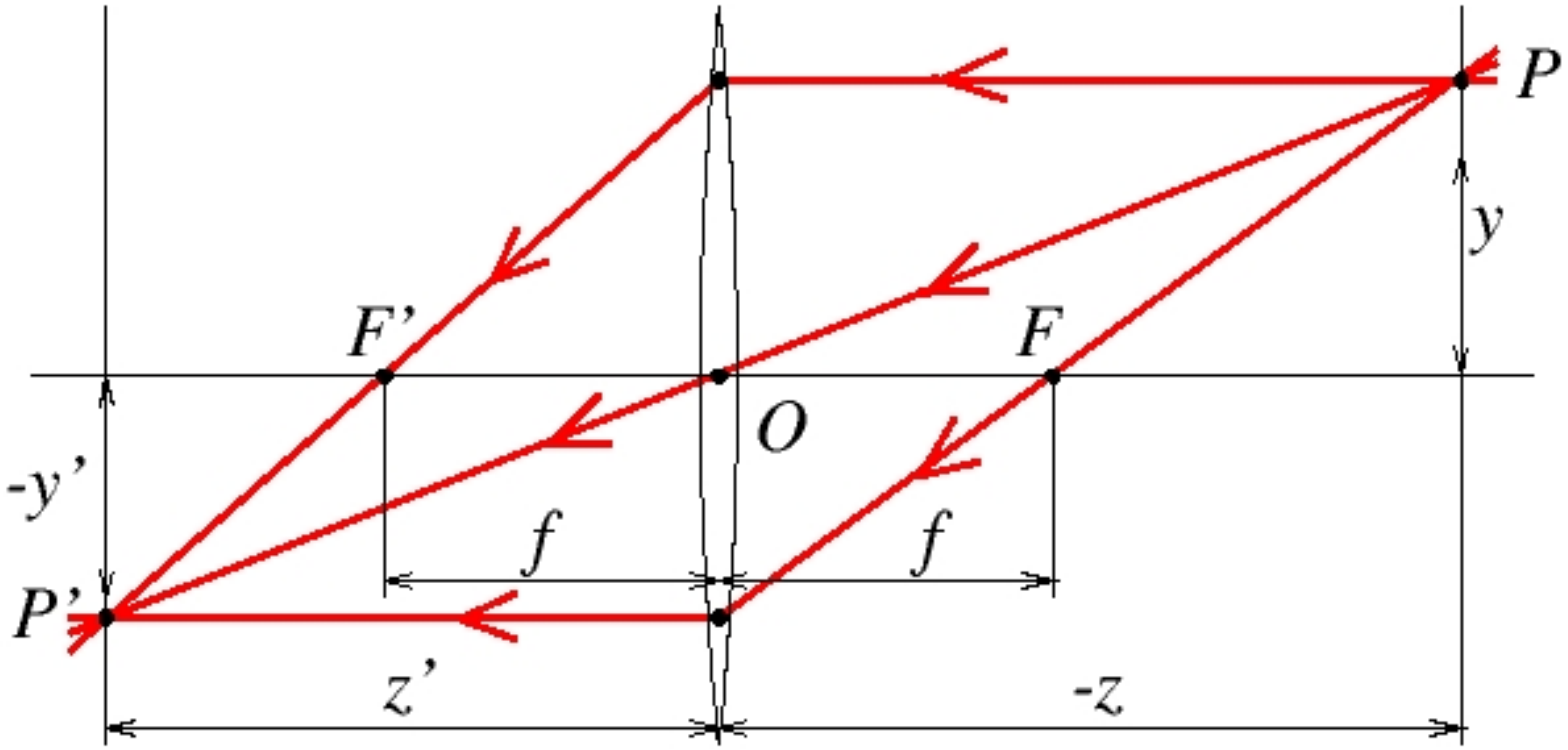
Video Source: https://www.youtube.com/watch?v=EU8LXxip1NM

Today's "**fun**" Example:

As one example, in 2010, **Cognex** signed a license agreement with Varioptic to add auto-focus capability to it DataMan line of industrial ID readers (press release May 29, 2012)

Video Source: https://www.youtube.com/watch?v=EU8LXxip1NM

Lecture 3: Re-cap Thin Lens Equation

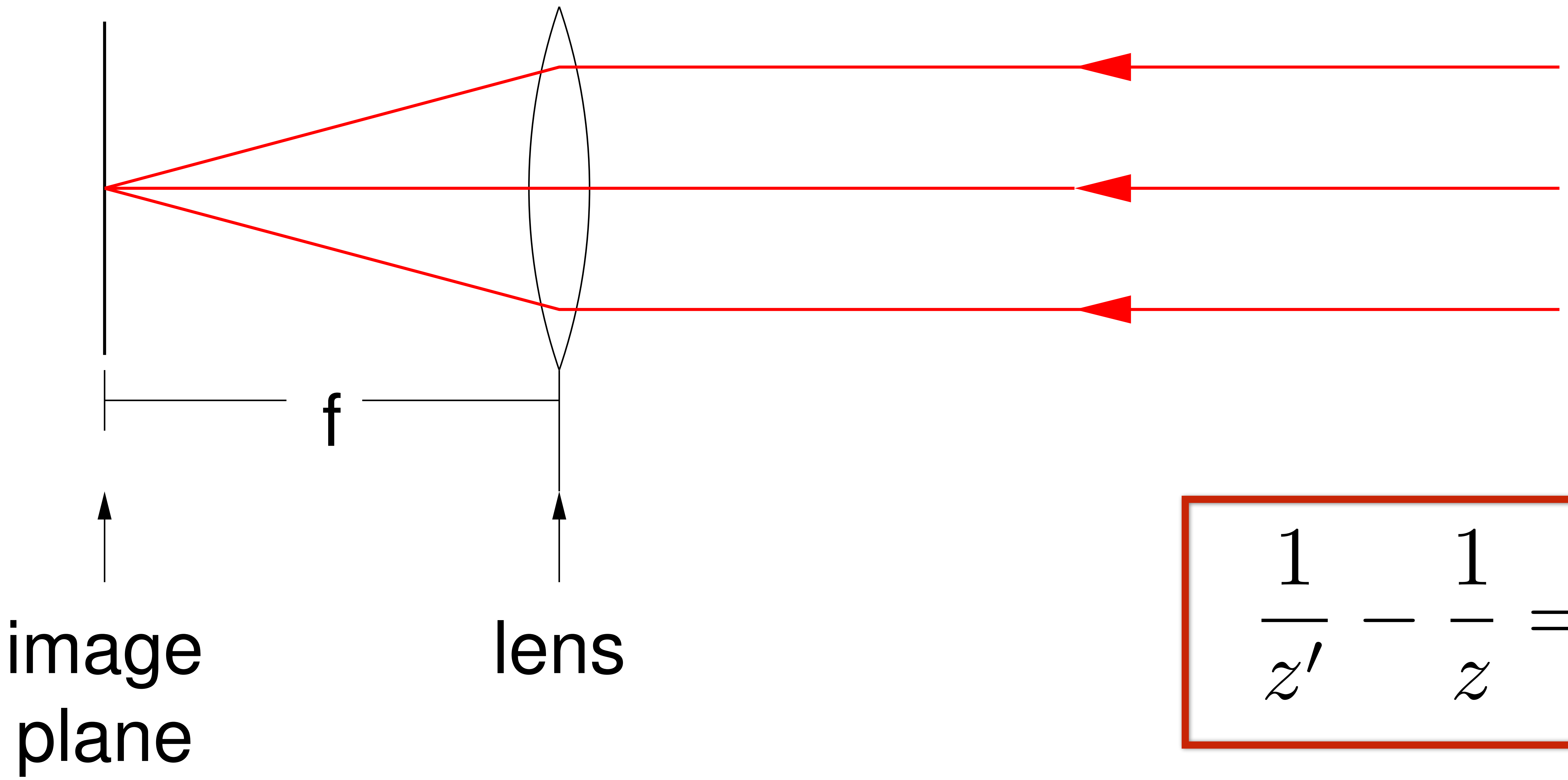


Forsyth & Ponce (1st ed.) Figure 1.9

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Lecture 3: Re-cap

Another way of looking at the **focal length** of a lens. The incoming rays, parallel to the optical axis, **converge to a single point a distance f behind the lens**. This is where we want to place the image plane.



$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

Let z go to -infinity

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

- -

https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$

- -

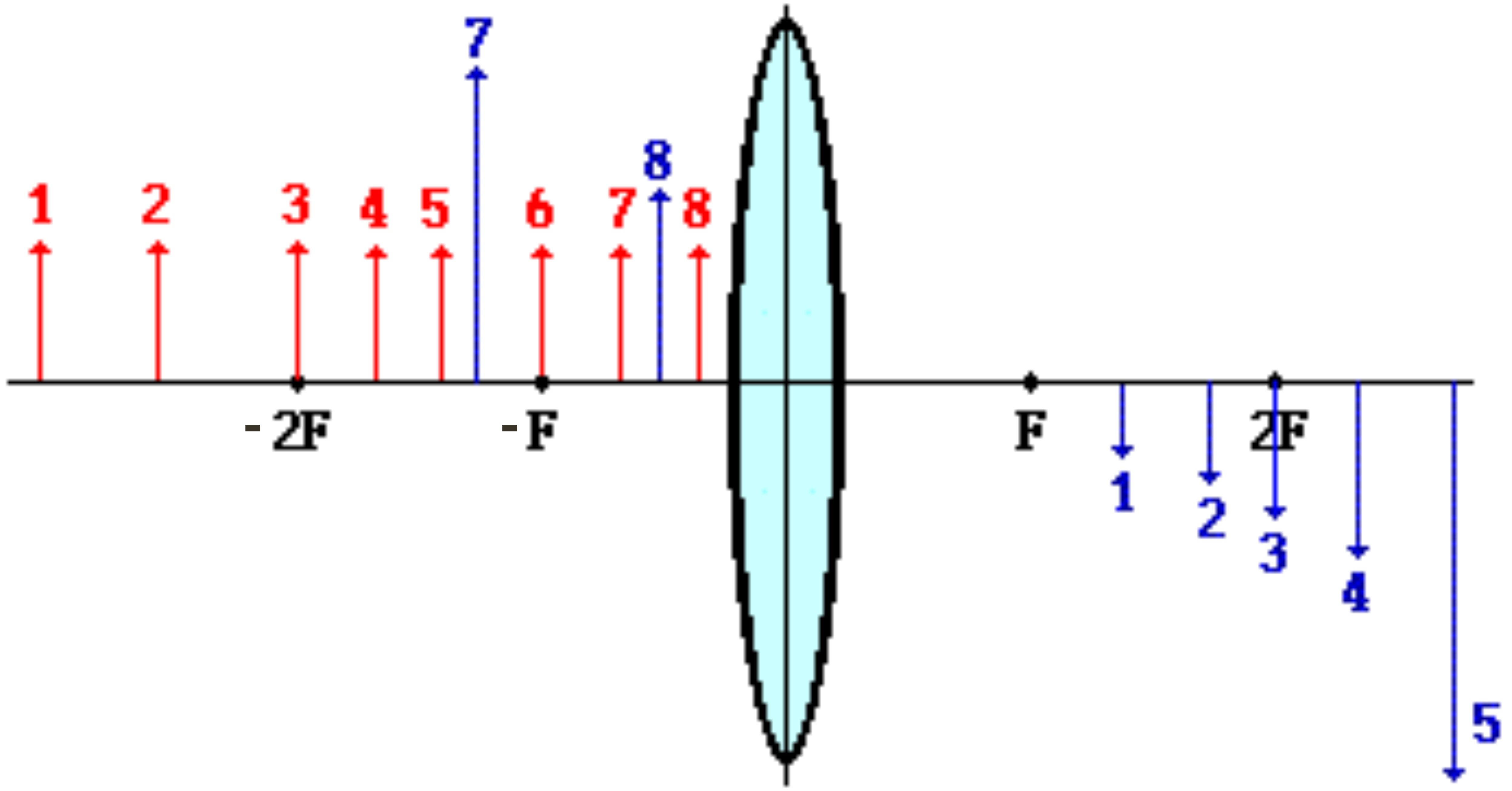
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



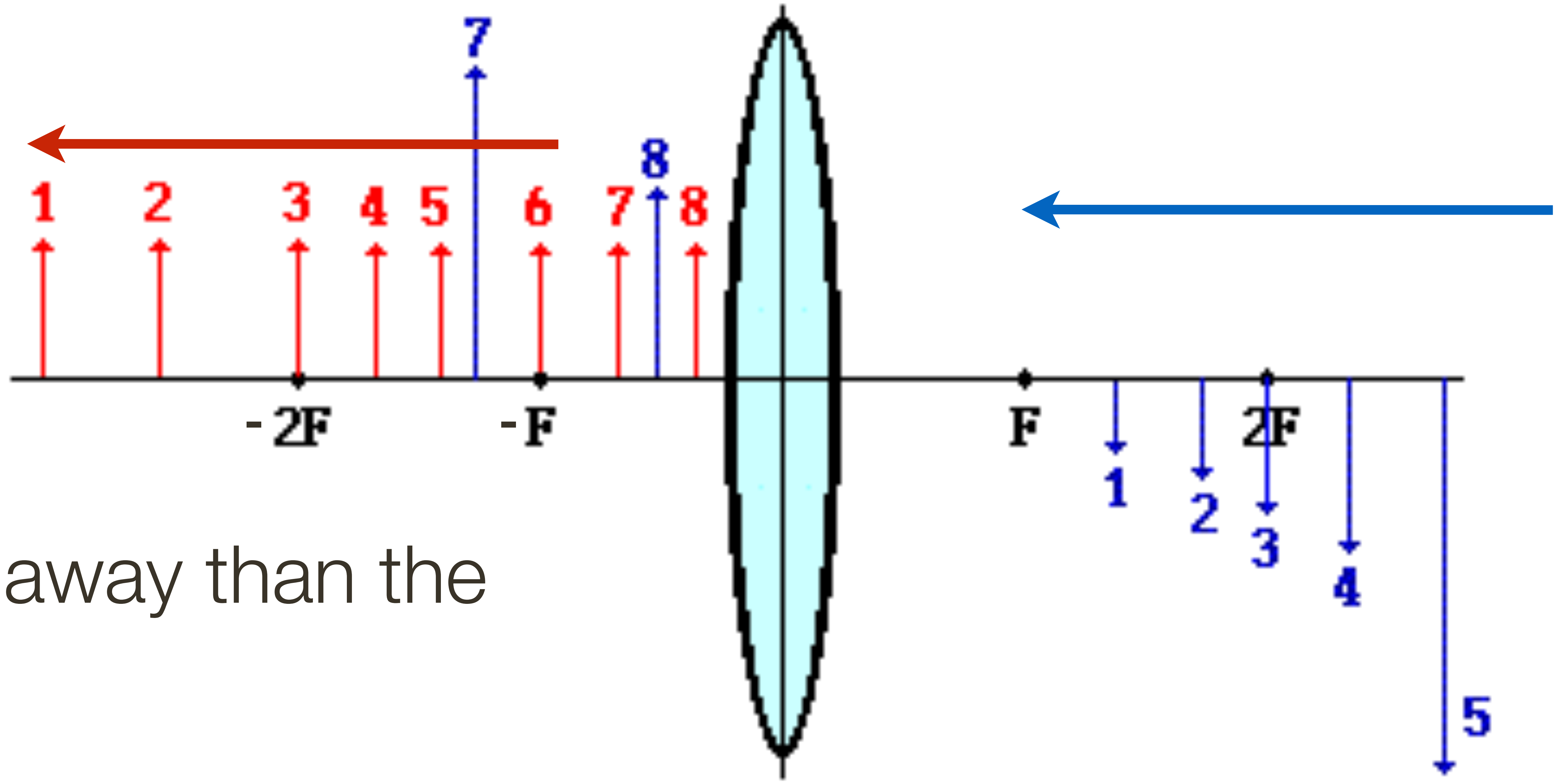
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects **further** away than the **focal length**

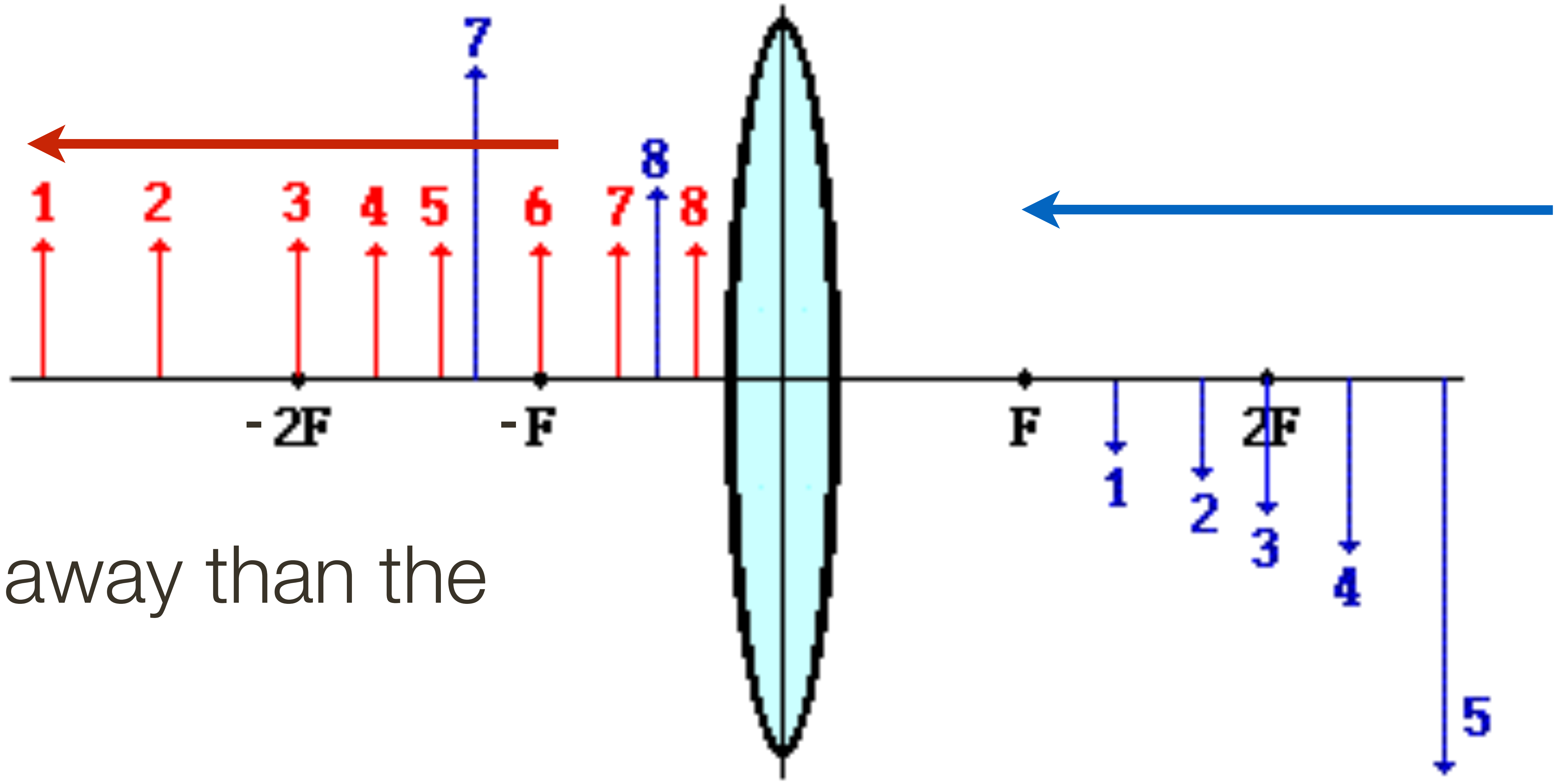
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



$$\lim_{z \to -\infty} \frac{zf}{z+f} = f$$

L'Hopital's Rule

Objects **further** away than the **focal length**

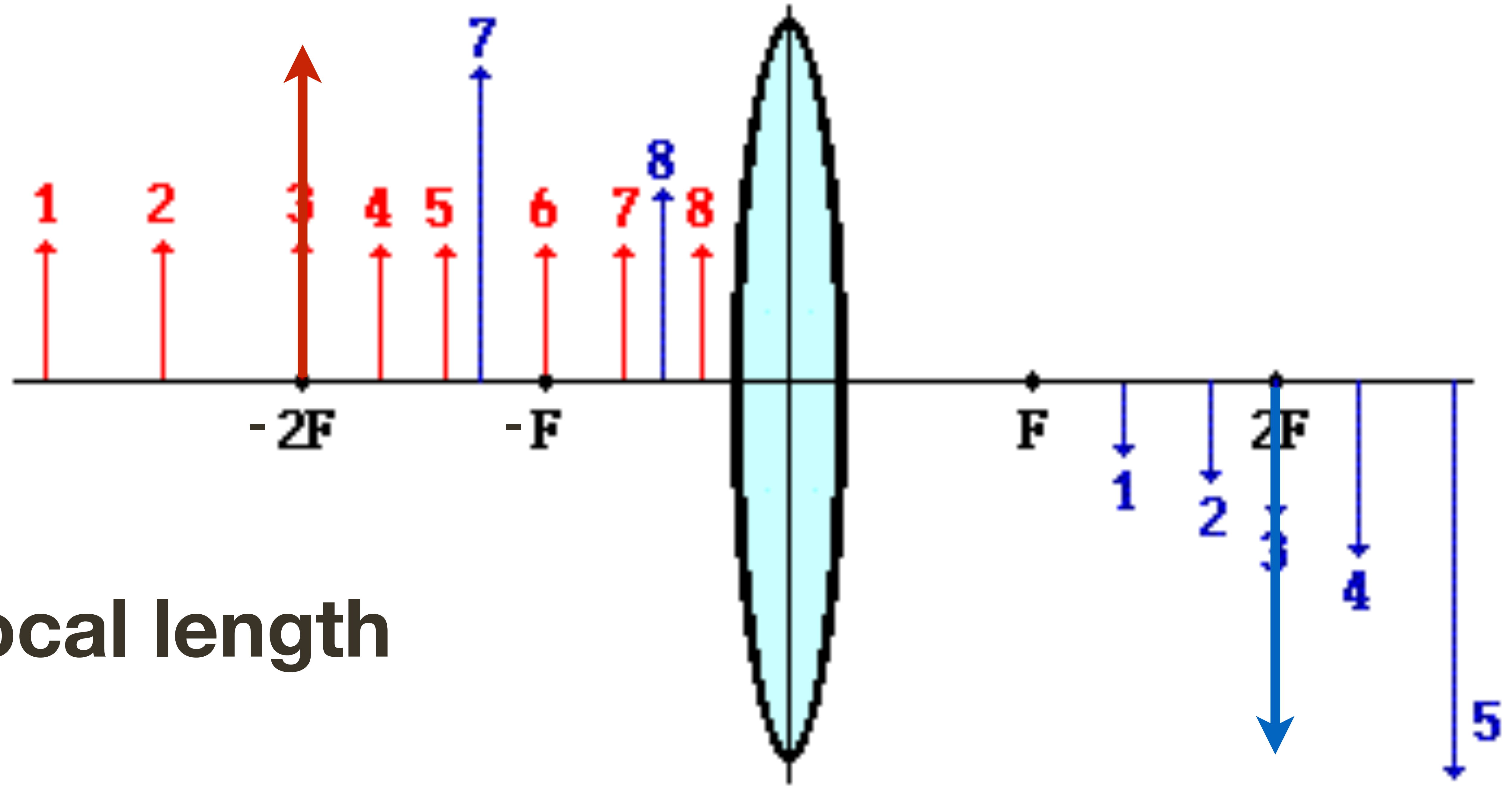
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects at 2 x **focal length**

https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

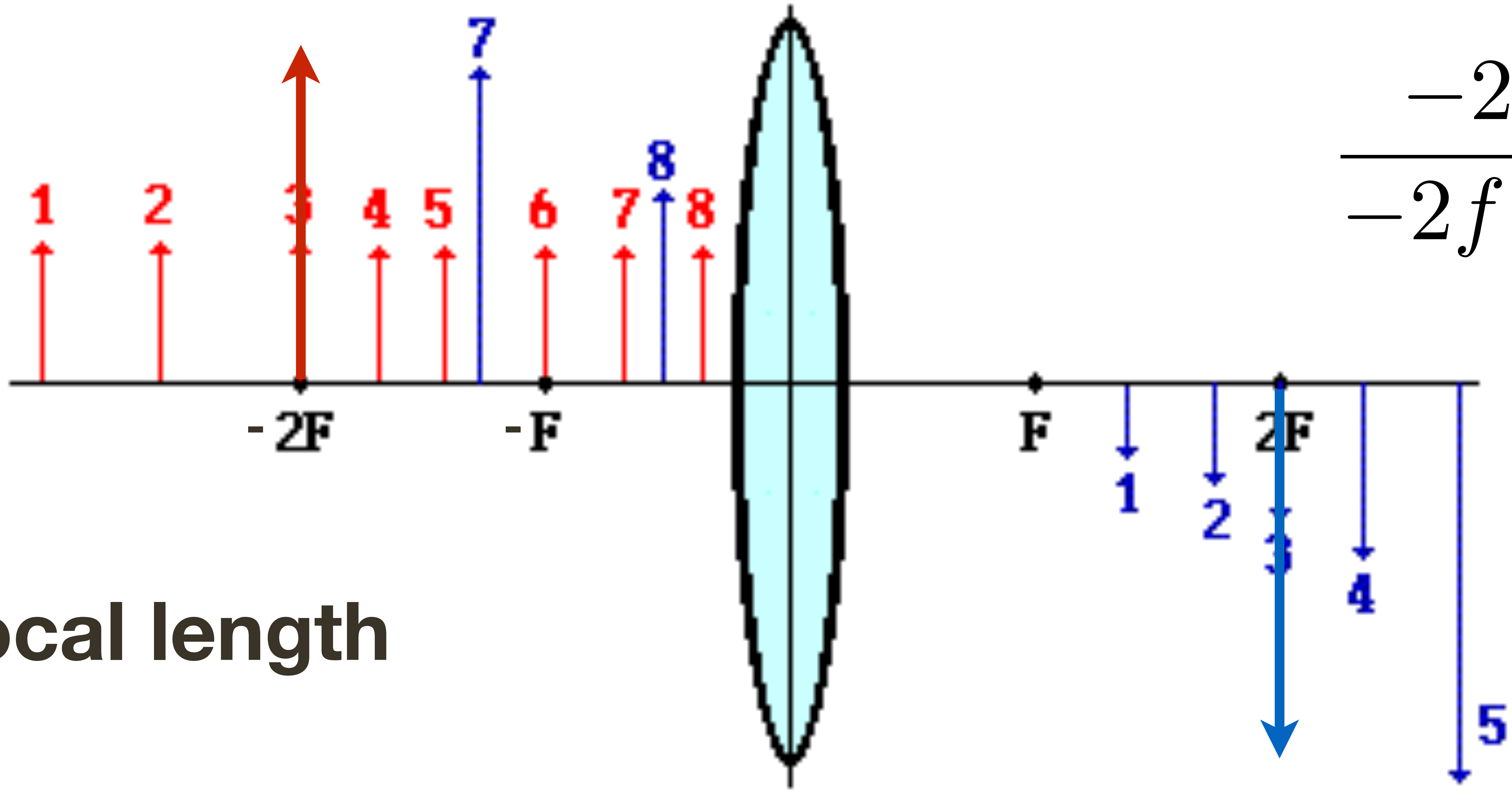
Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$

$$\frac{-2f^2}{-2f+f} = \frac{-2f^2}{-f} = 2f$$



Objects at 2 x **focal length**

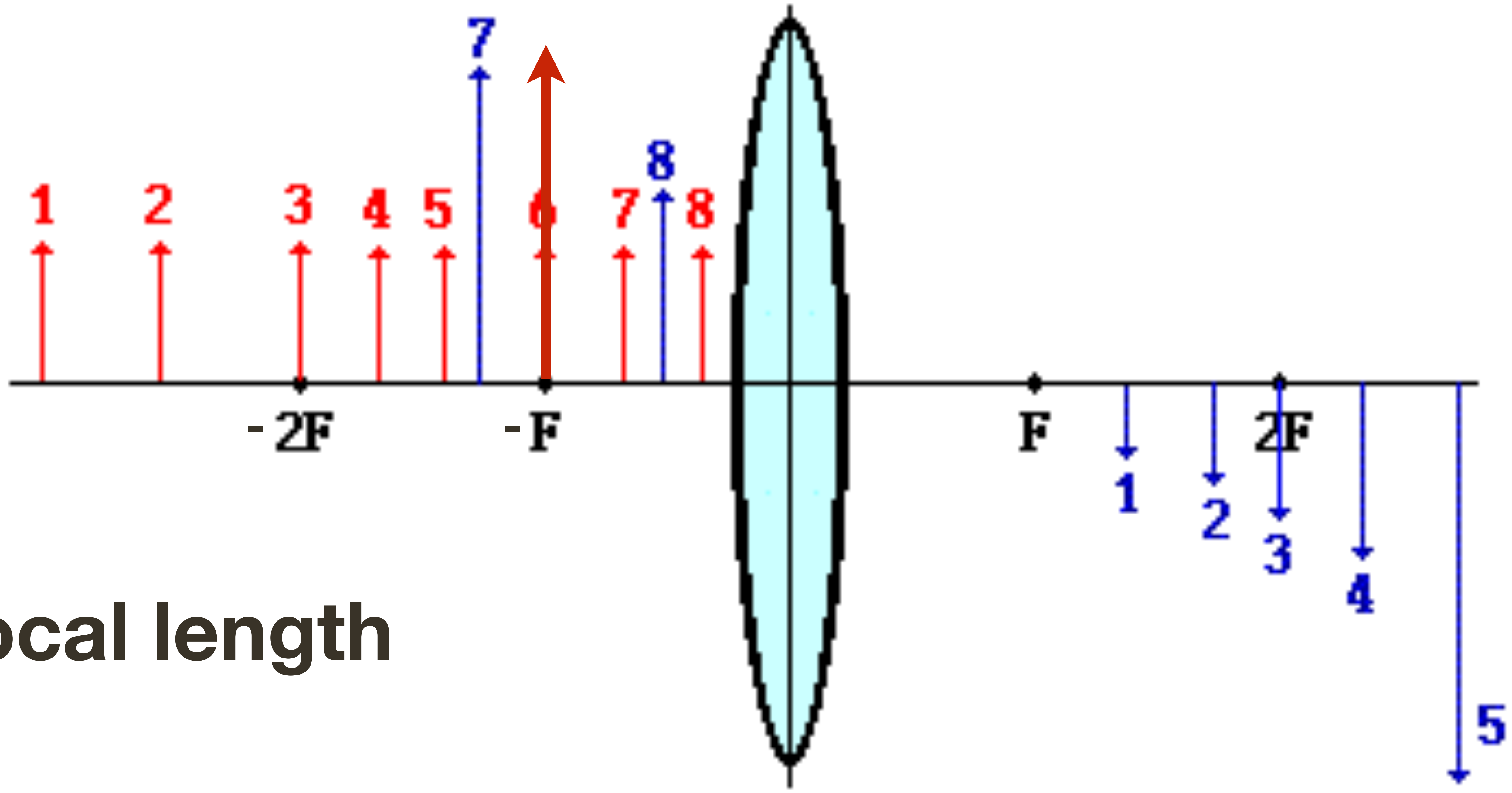
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects at the **focal length**

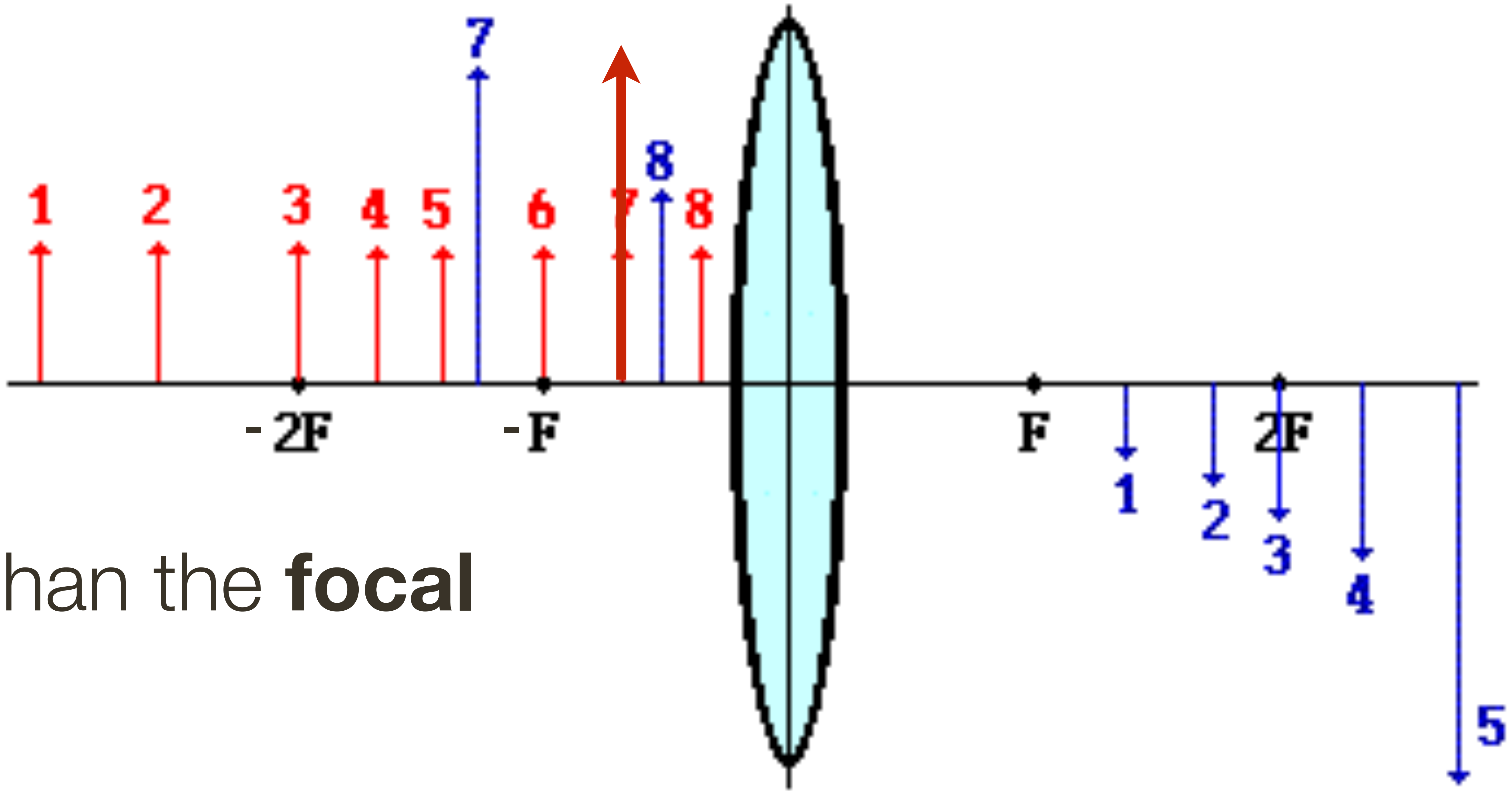
https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Thin Lens Equation

Is convergence projection point **directly** / **inversely** proportional to world position?

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

$$z' = \frac{zf}{z+f}$$



Objects **closer** than the **focal length**

https://www.physicsclassroom.com/class/refrn/Lesson-5/Converging-Lenses-Object-Image-Relations

Lecture 3: Re-cap Lens Imaging Artifacts

Chromatic **aberration**

— Index of refraction depends on wavelength, λ, of light

— Light of different colours follows different paths

— Therefore, not all colours can be in equal focus

Scattering at the lens surface

— Some light is reflected at each lens surface

There are other **geometric phenomena/distortions**

— pincushion distortion

— barrel distortion

— etc

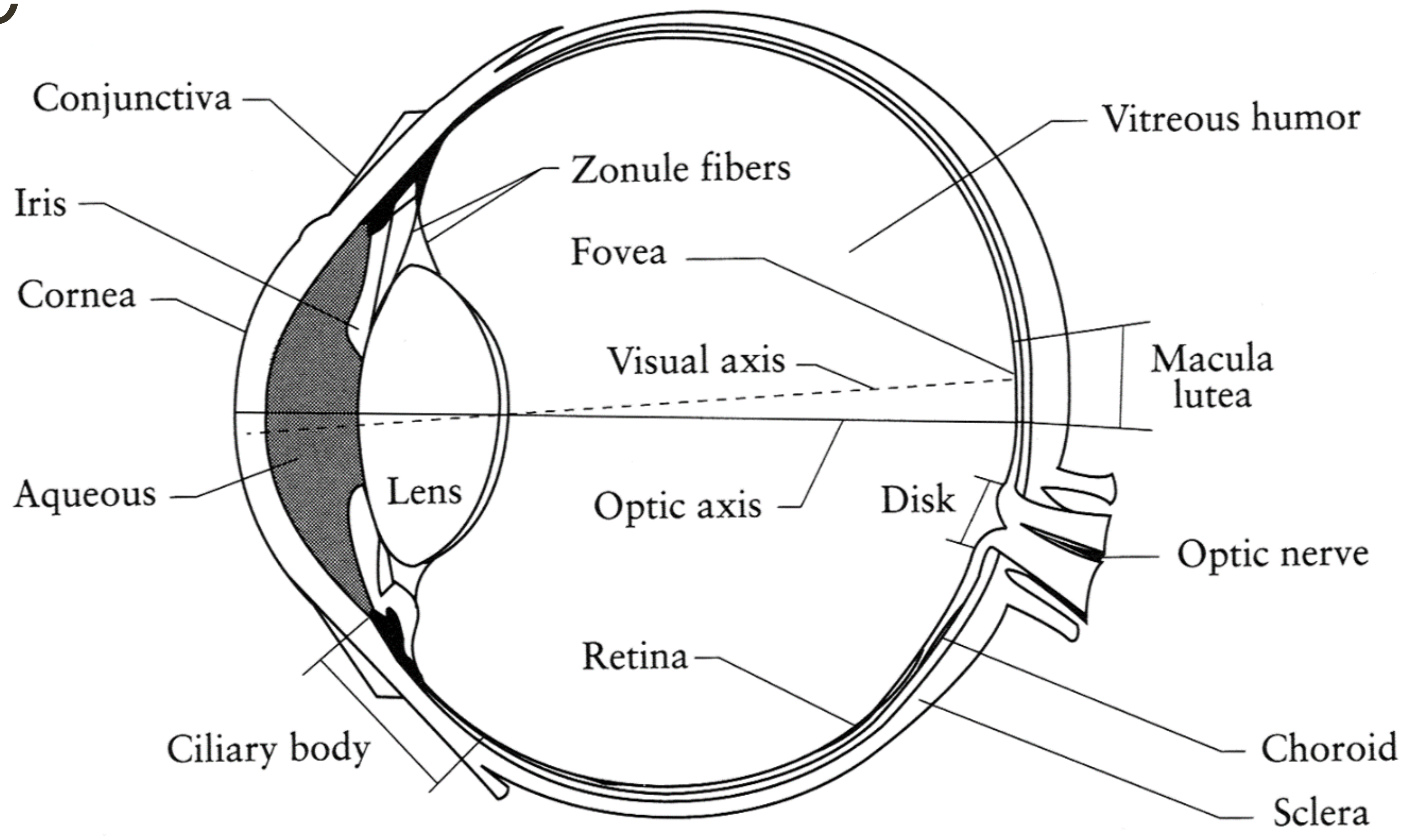
Lecture 3: Re-cap Human Eye

— The eye has an **iris** (like a camera)

— **Focusing** is done by changing shape of lens

— When the eye is properly focused, light from an object outside the eye is imaged on the **retina**

— The retina contains light receptors called **rods** and **cones**



pupil = pinhole / aperture

retina = film / digital sensor

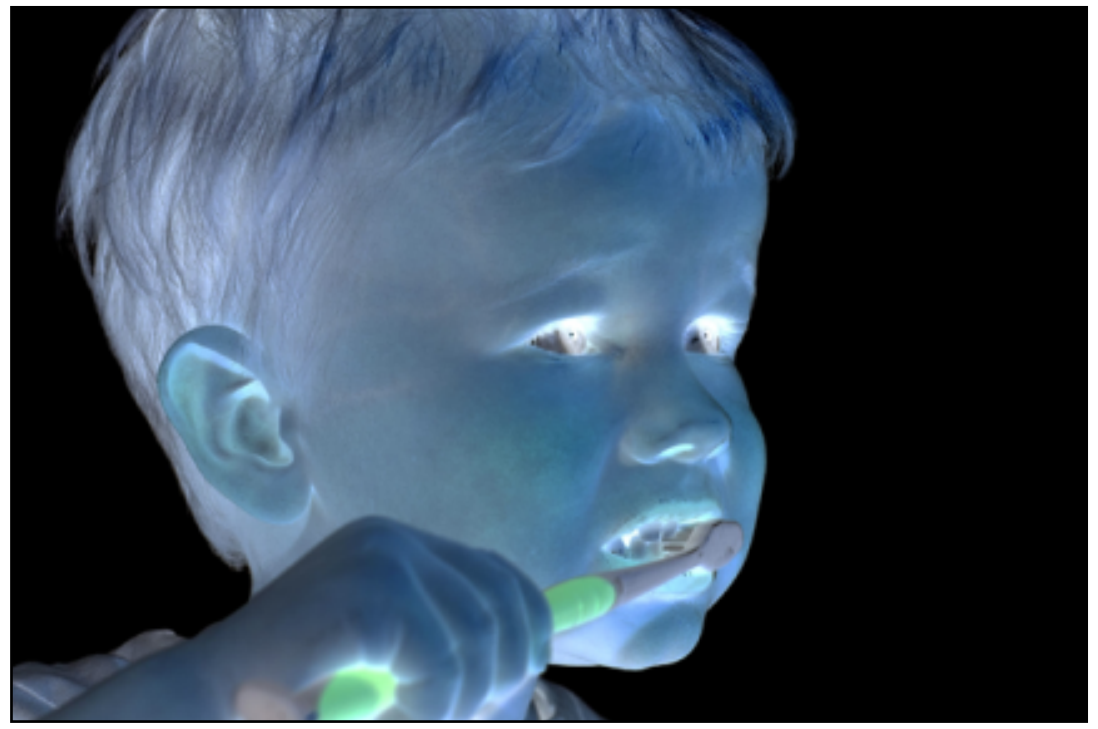
Slide adopted from: Steve Seitz

What types of **transformations** can we do?

$I(X, Y)$

Filtering

$I'(X, Y)$

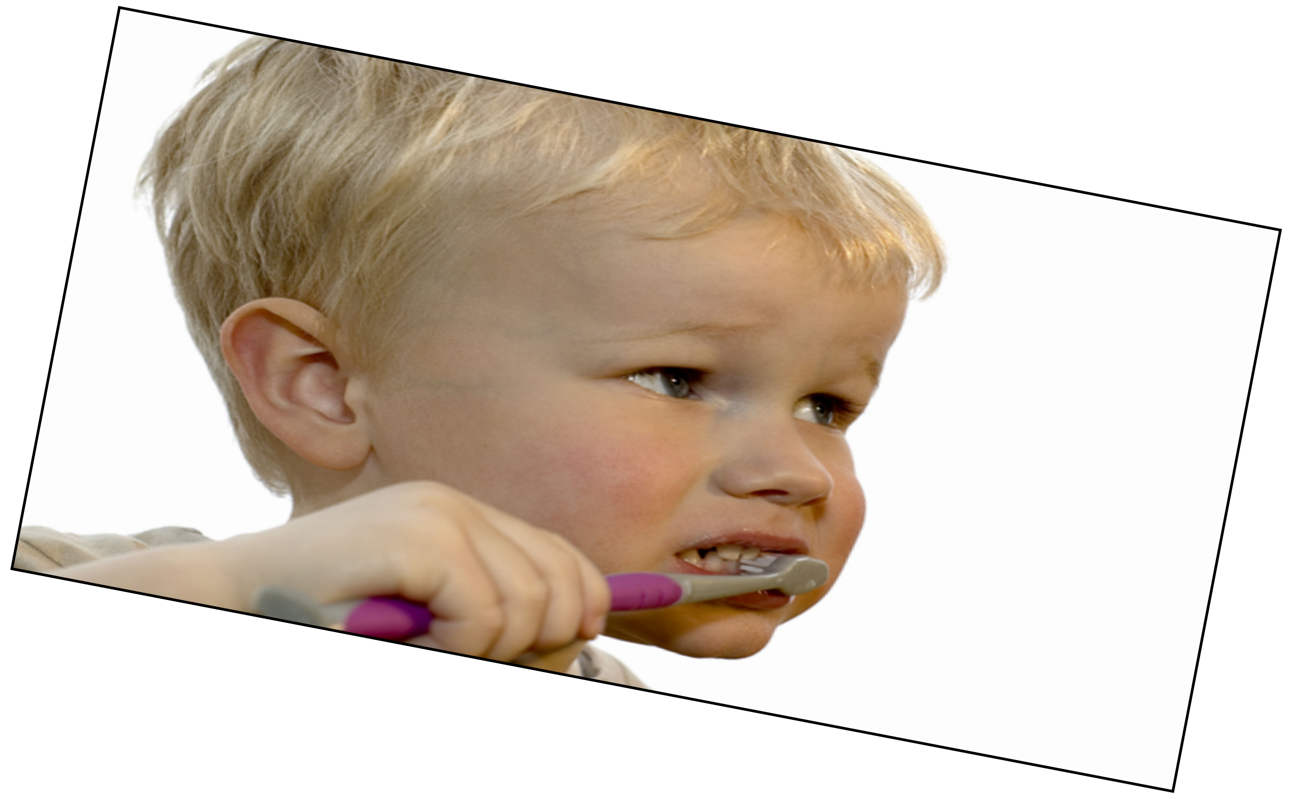


changes range of image function

$I(X, Y)$

Warping

$I'(X, Y)$

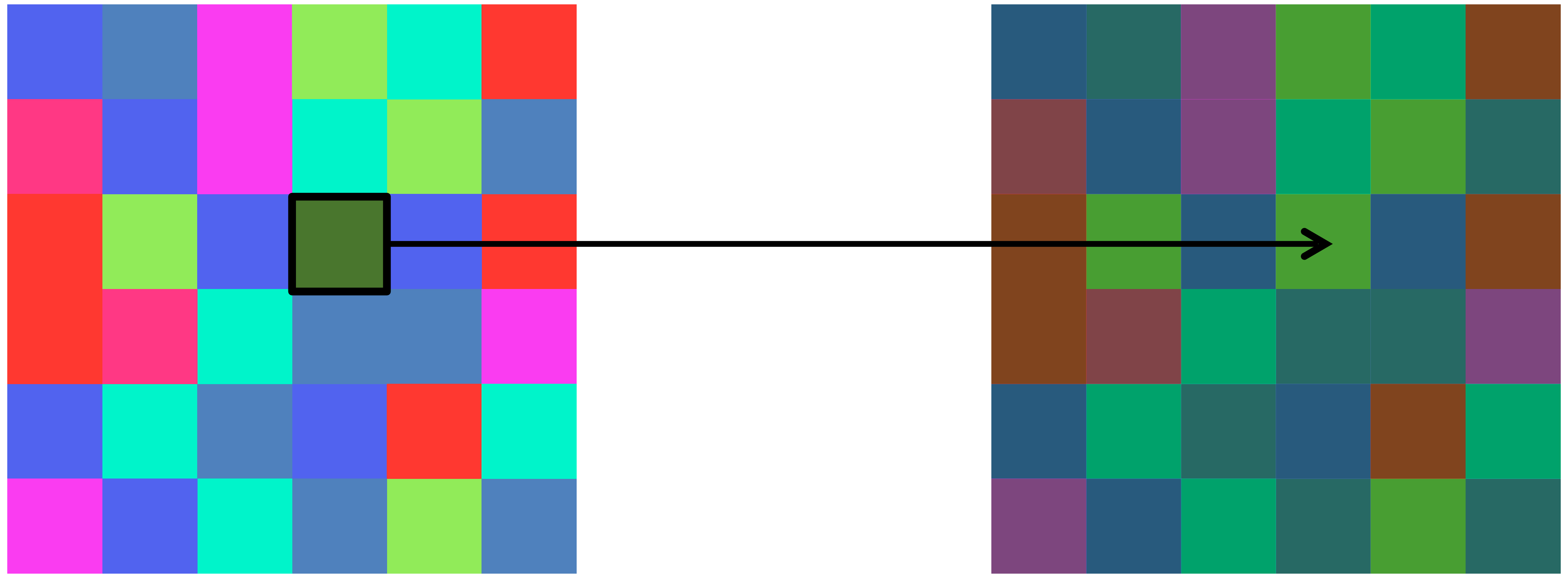


changes domain of image function

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

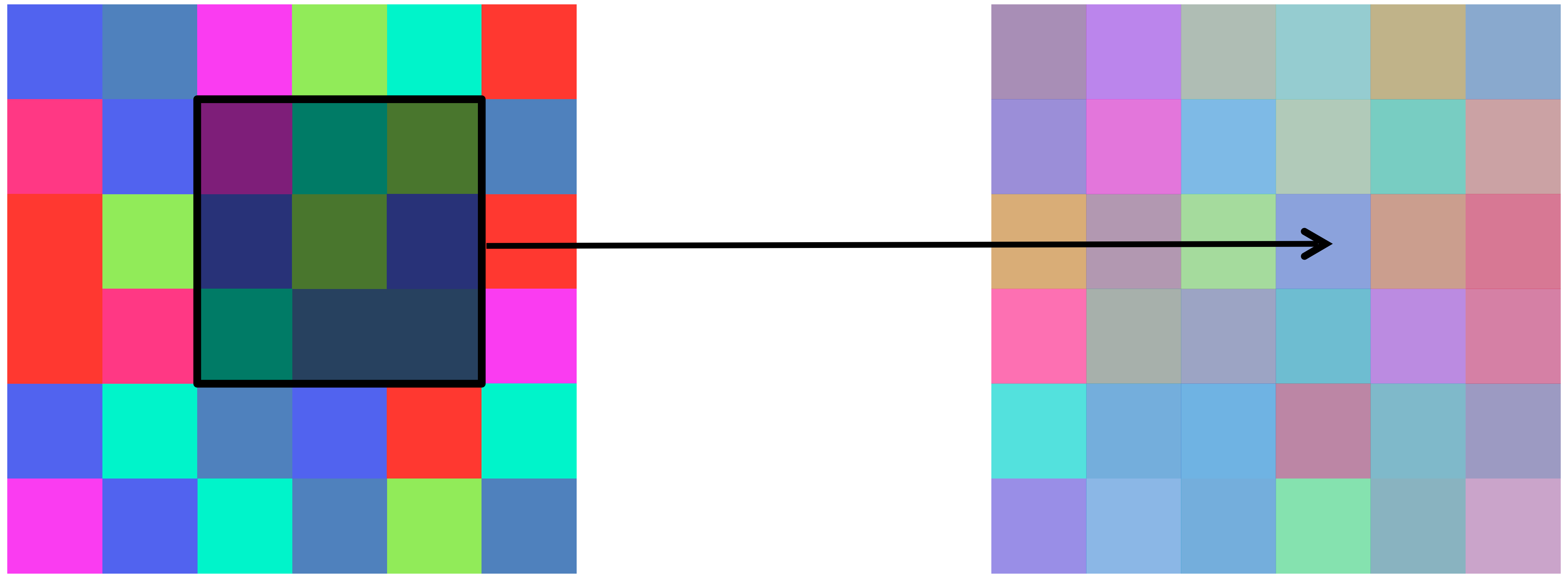
What types of **filtering** can we do?

Point Operation



point processing

Neighborhood Operation



"filtering"

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

darken

lower contrast

non-linear lower contrast

$I(X, Y)$

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$I(X, Y)$

darken

$I(X, Y) - 128$

lower contrast

non-linear lower contrast

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

darken

lower contrast

non-linear lower contrast

$I(X,Y)$

$I(X,Y) - 128$

$\frac{I(X,Y)}{2}$

invert

lighten

raise contrast

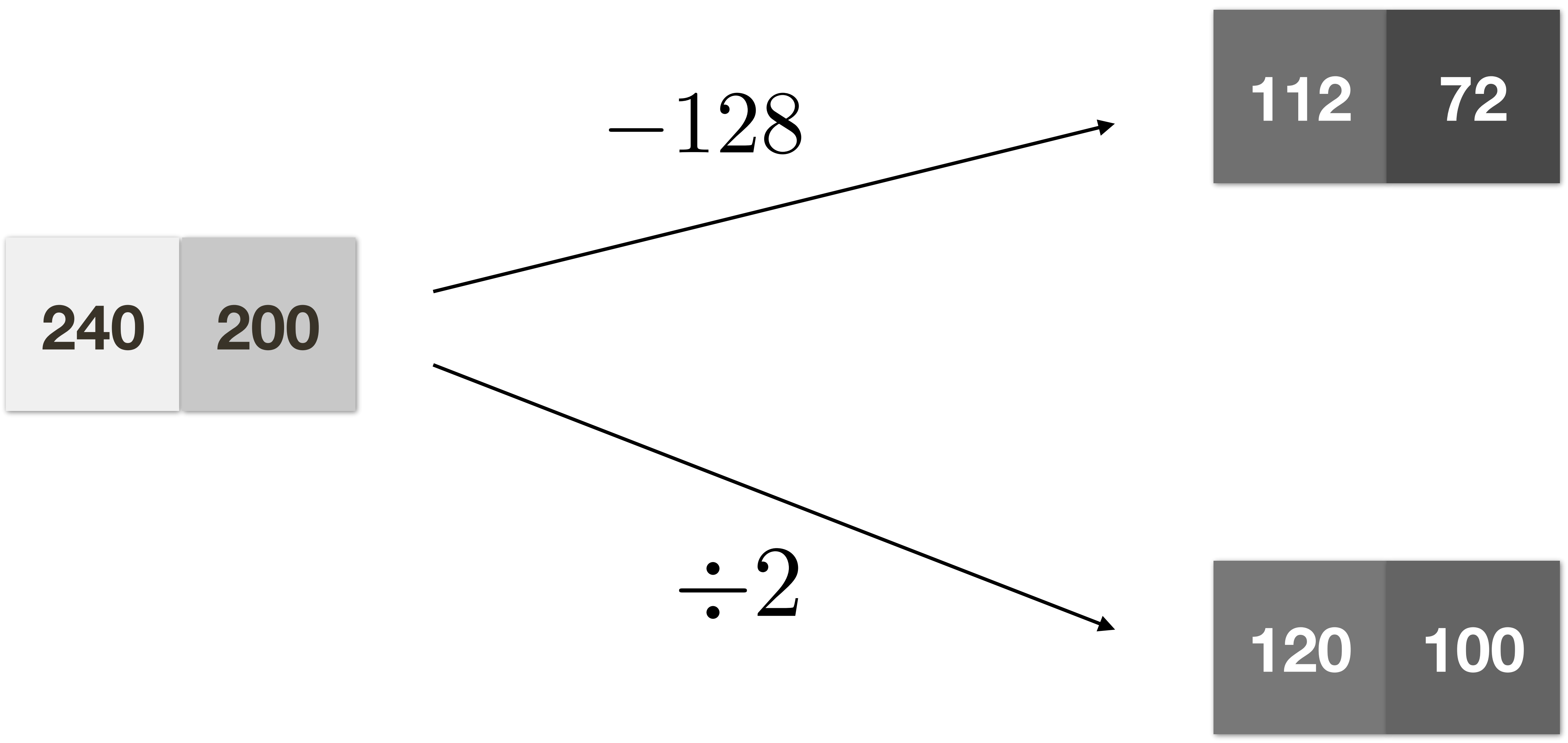
non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Darkening v.s. **Contrast**

Brightness: all pixels get lighter/darker, relative difference between pixel values stays the same

Contrast: relative difference between pixel values becomes higher / lower



Examples of **Point Processing**

original

$$I(X, Y)$$

darken

$$I(X, Y) - 128$$

lower contrast

$$\frac{I(X, Y)}{2}$$

non-linear lower contrast

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

$$255 - I(X,Y)$$

lighten

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

$$255 - I(X,Y)$$

lighten

$$I(X,Y) + 128$$

raise contrast

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$I(X,Y)$

darken

$I(X,Y) - 128$

lower contrast

$\frac{I(X,Y)}{2}$

non-linear lower contrast

$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$

invert

$255 - I(X,Y)$

lighten

$I(X,Y) + 128$

raise contrast

$I(X,Y) \times 2$

non-linear raise contrast

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert

$$255 - I(X,Y)$$

lighten

$$I(X,Y) + 128$$

raise contrast

$$I(X,Y) \times 2$$

non-linear raise contrast

$$\left(\frac{I(X,Y)}{255}\right)^{2} \times 255$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Examples of **Point Processing**

original

$$I(X,Y)$$

darken

$$I(X,Y) - 128$$

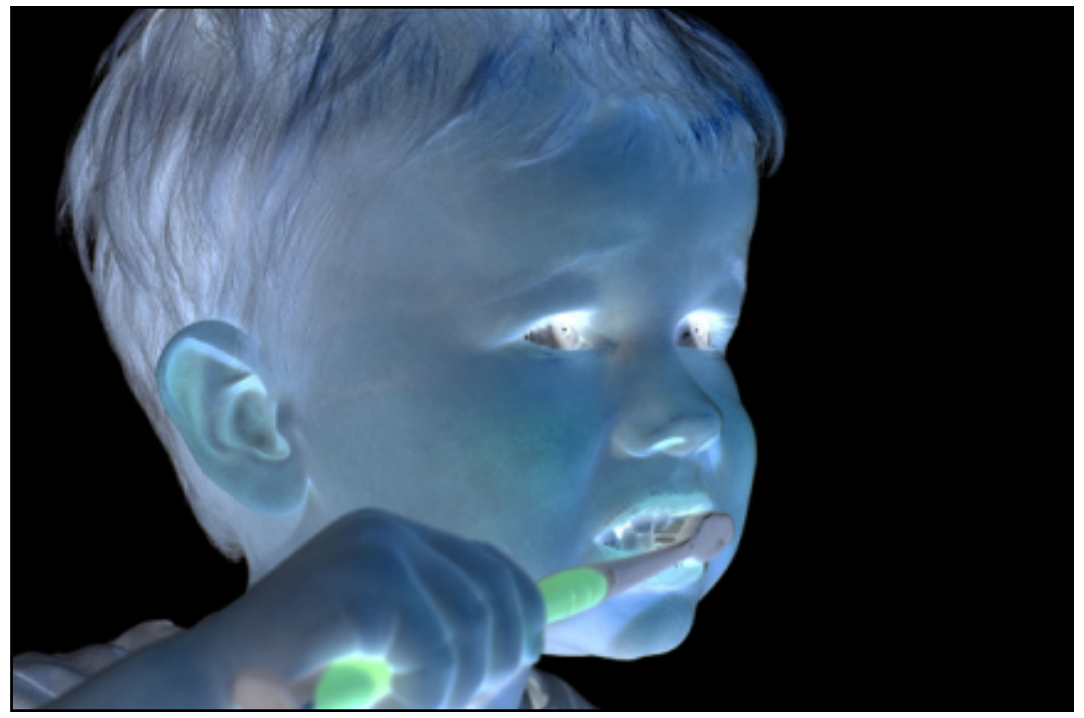
lower contrast

$$\frac{I(X,Y)}{2}$$

non-linear lower contrast

$$\left(\frac{I(X,Y)}{255}\right)^{1/3} \times 255$$

invert



$$255 - I(X,Y)$$

lighten

$$I(X,Y) + 128$$

raise contrast

$$I(X,Y) \times 2$$

non-linear raise contrast

$$\left(\frac{I(X,Y)}{255}\right)^{2} \times 255$$

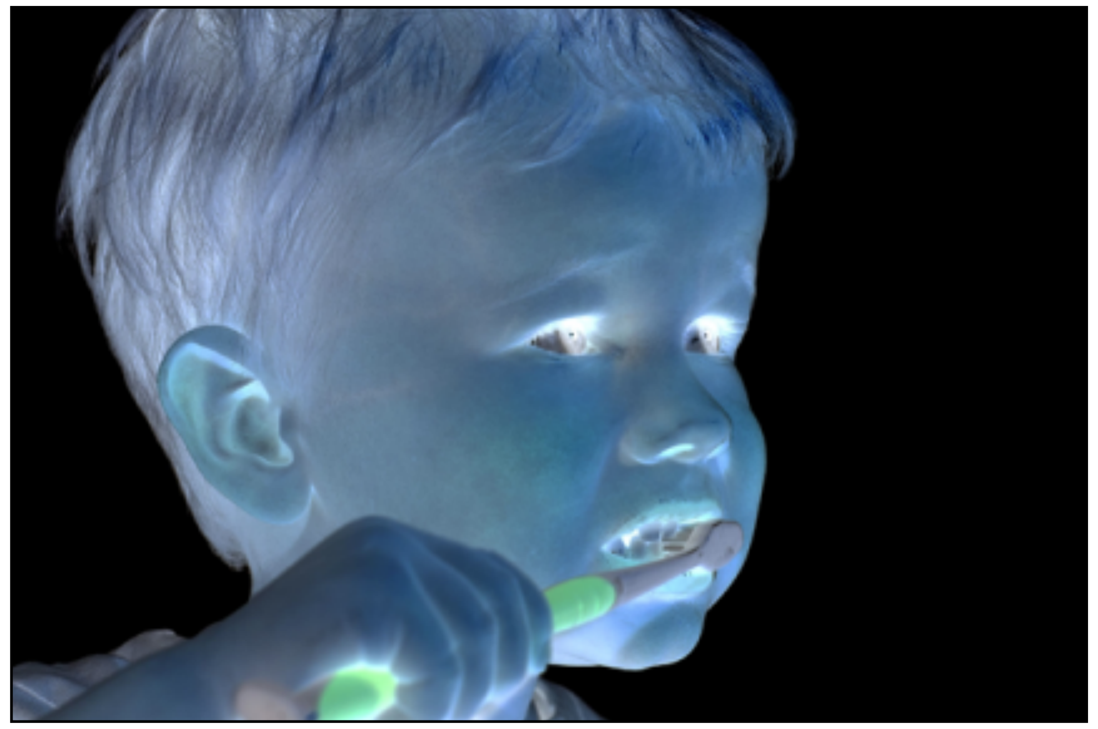
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

What types of **transformations** can we do?

$I(X,Y)$

Filtering

$I'(X,Y)$

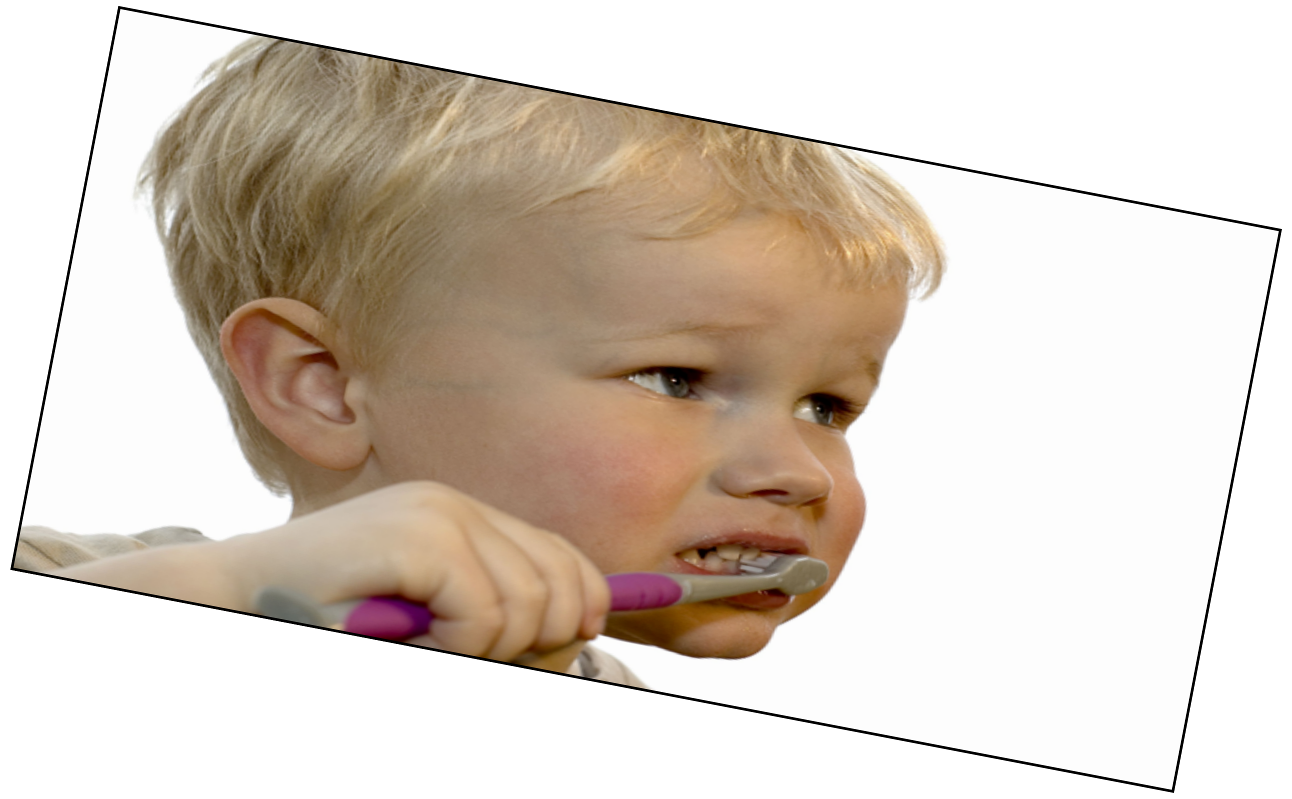


changes range of image function

$I(X,Y)$

Warping

$I'(X,Y)$

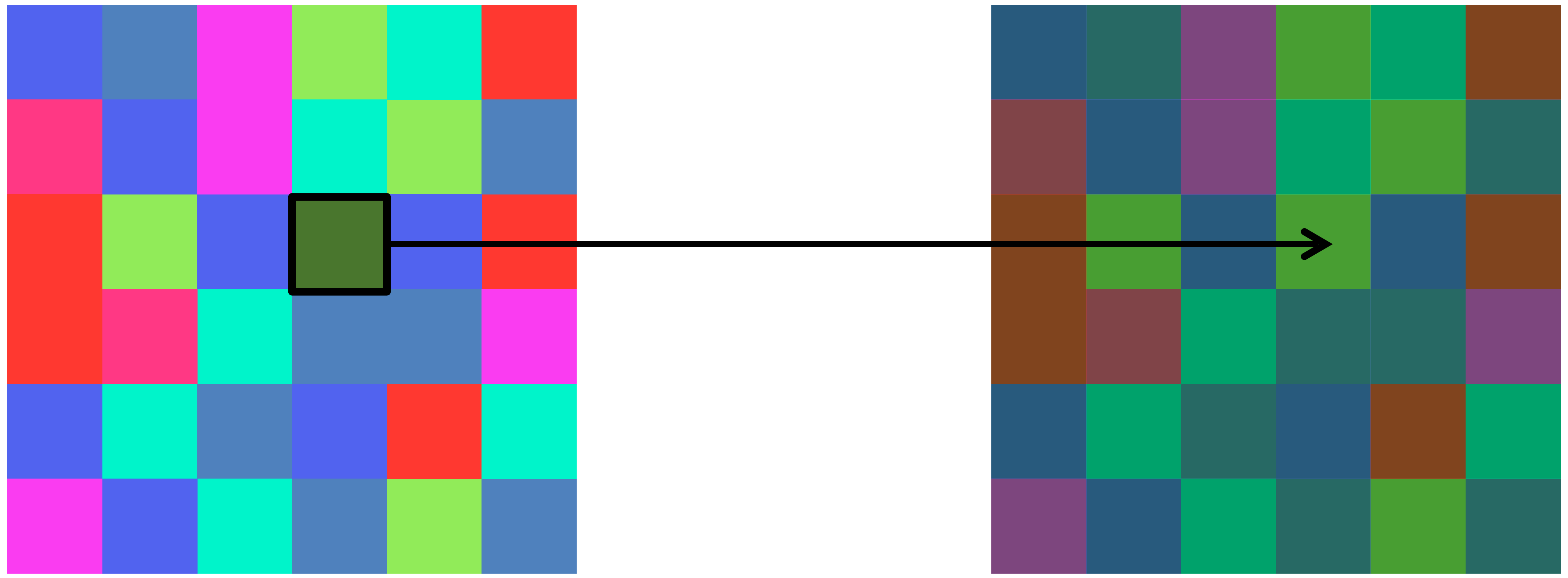


changes domain of image function

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

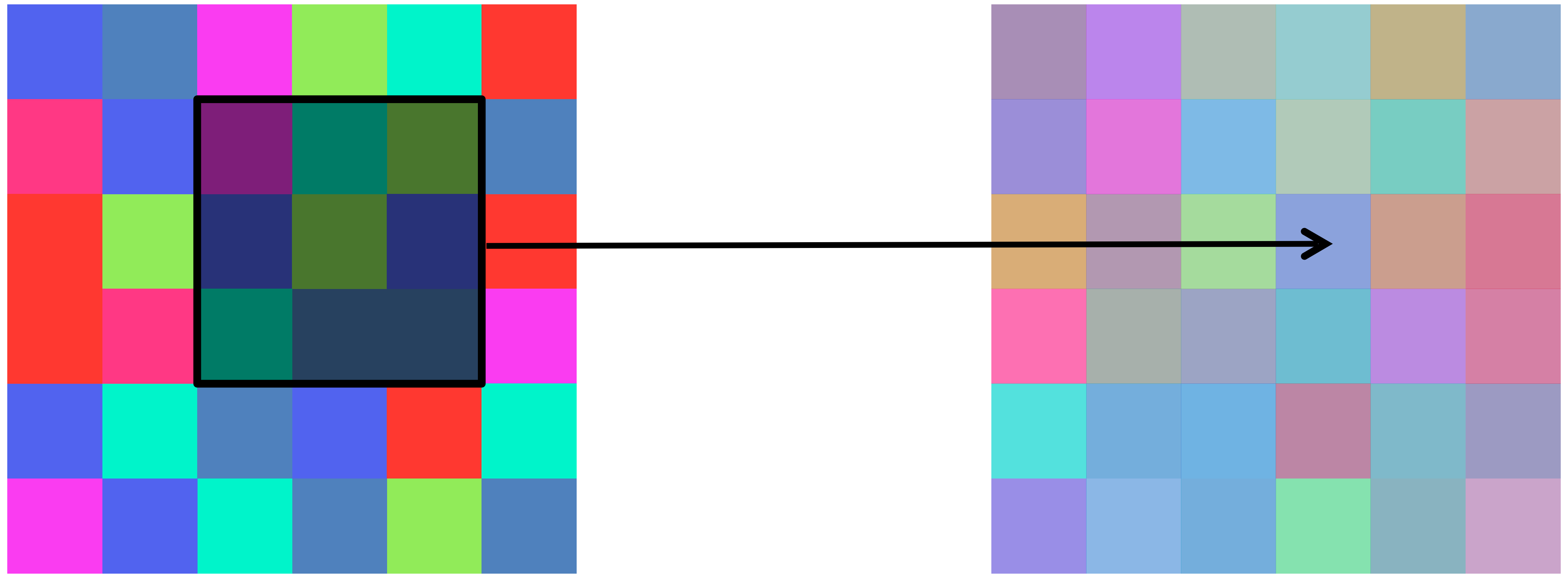
What types of **filtering** can we do?

Point Operation



point processing

Neighborhood Operation



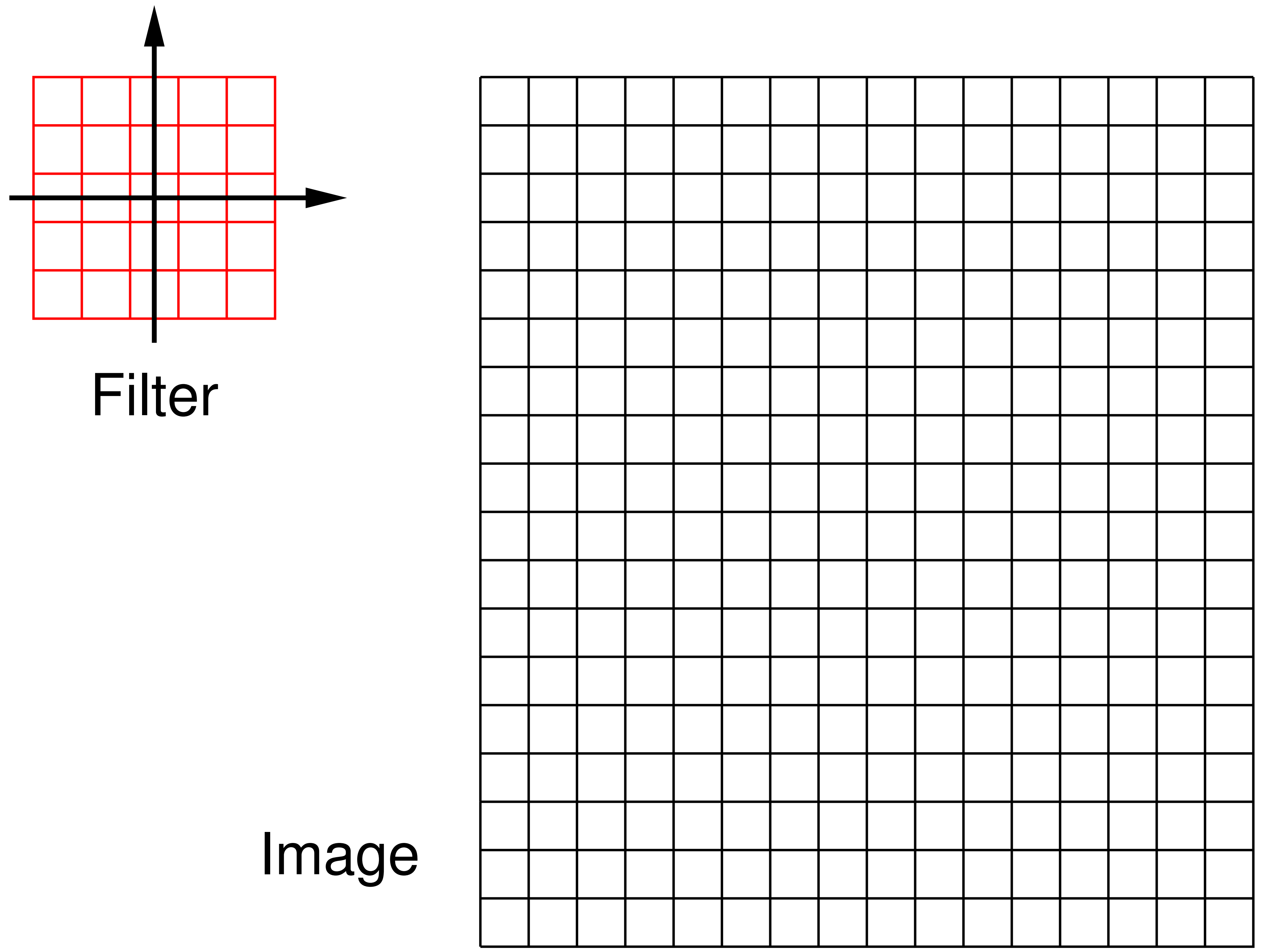
“filtering”

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear **Filters**

Let $I(X, Y)$ be an $n \times n$ digital image (for convenience we let width = height)

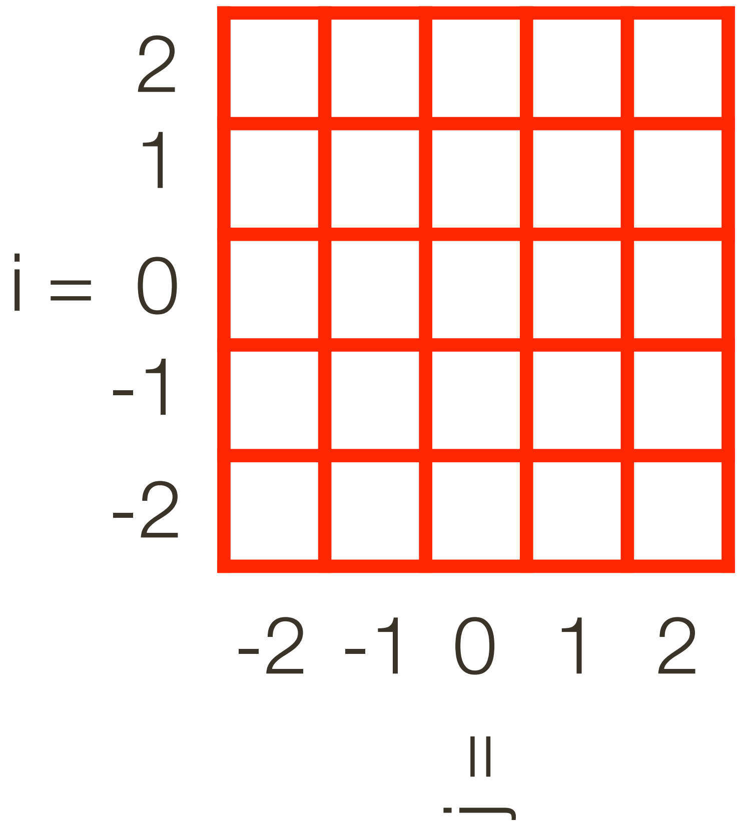
Let $F(X, Y)$ be another $m \times m$ digital image (our "**filter**" or "**kernel**")



For convenience we will assume m is odd. (Here, $m = 5$)

Linear **Filters**

Let $k = \left\lfloor \frac{m}{2} \right\rfloor$



Compute a new image, $I'(X,Y)$, as follows

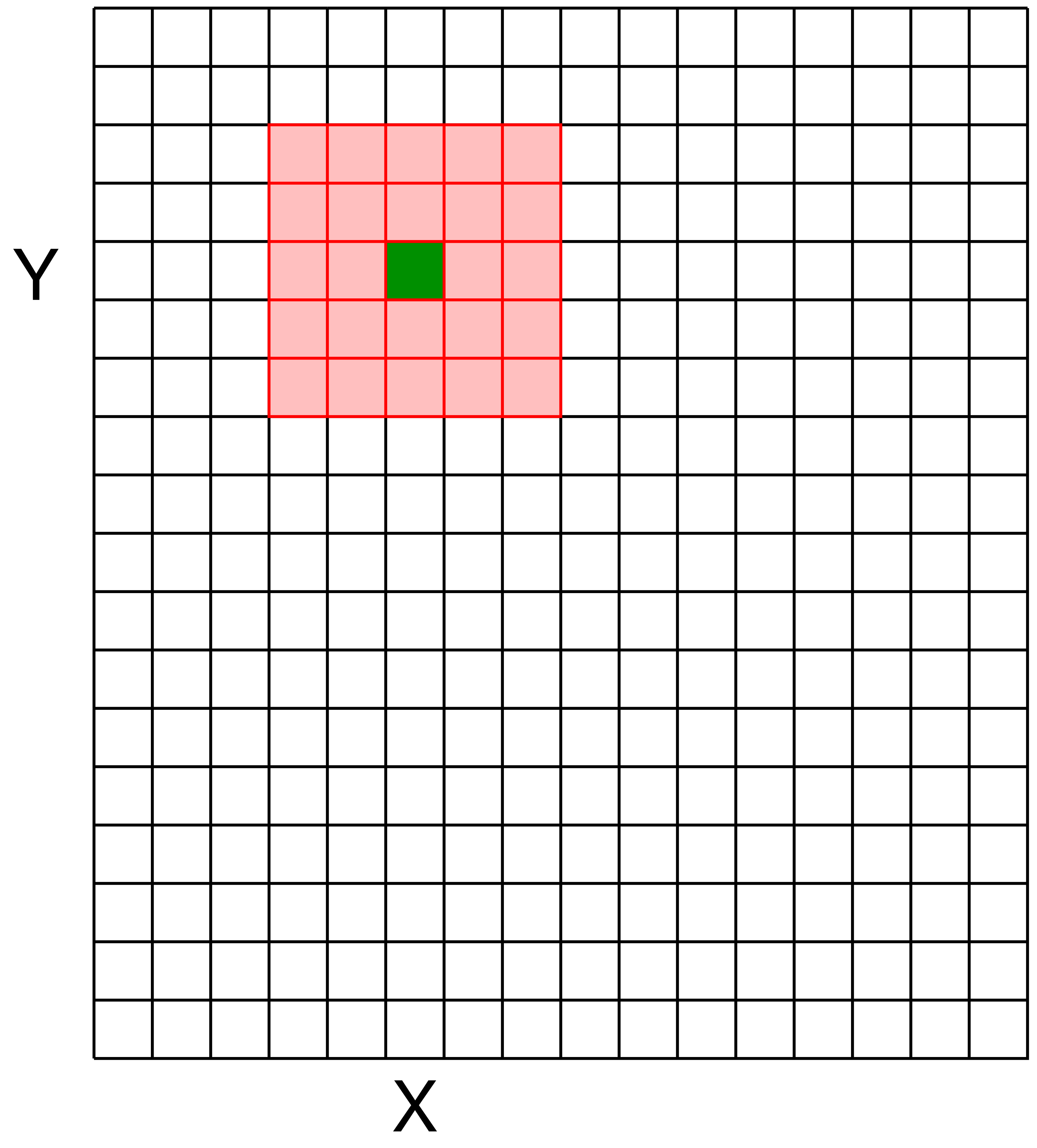
$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) \, I(X+i, Y+j)$$

output — filter — image (signal)

Intuition: each pixel in the output image is a linear combination of the same index pixel and its neighboring pixels in the original image

Linear **Filters**

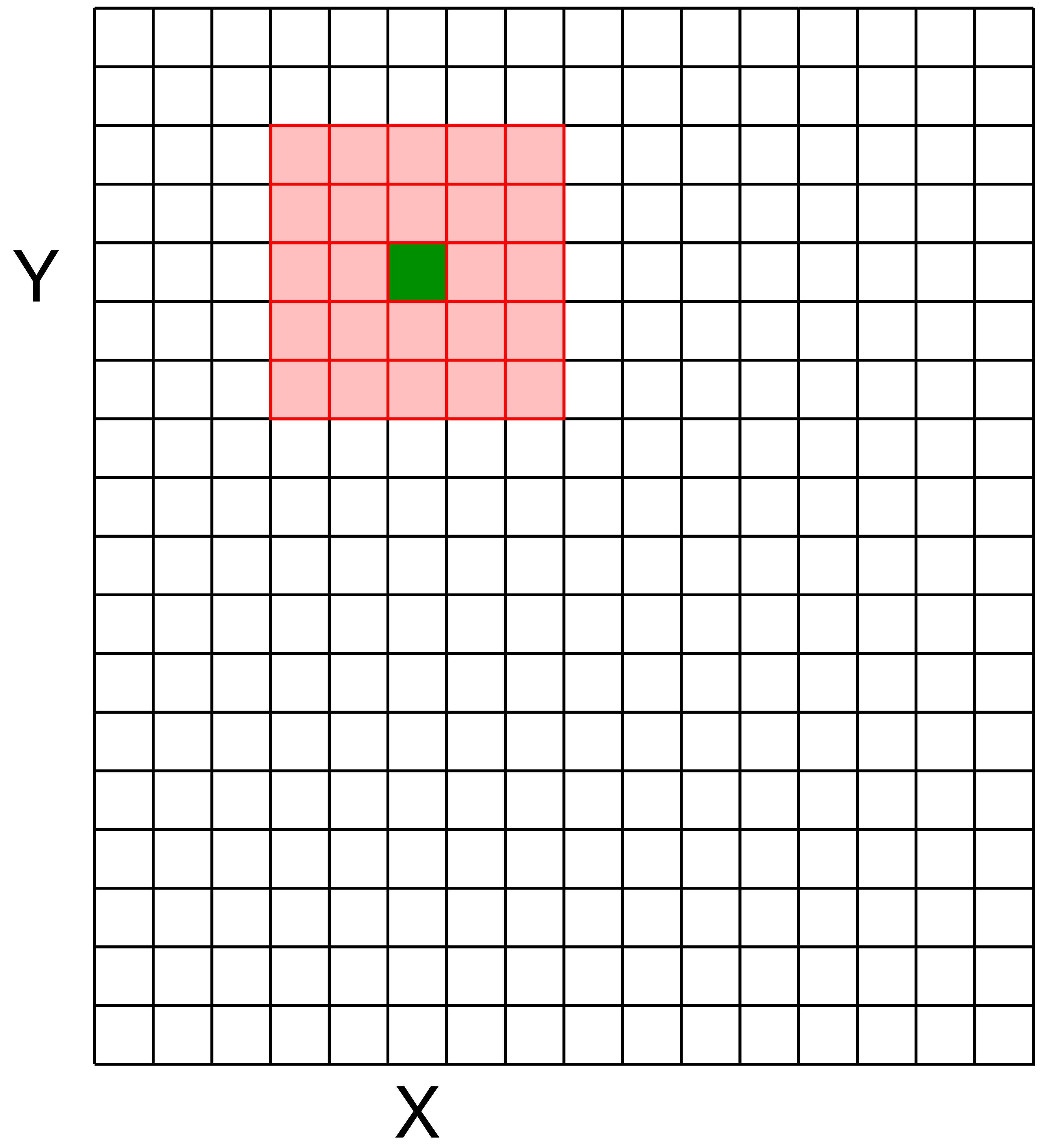
For a give X and Y, superimpose the filter on the image centered at (X, Y)



Linear **Filters**

For a give X and Y, superimpose the filter on the image centered at (X, Y)

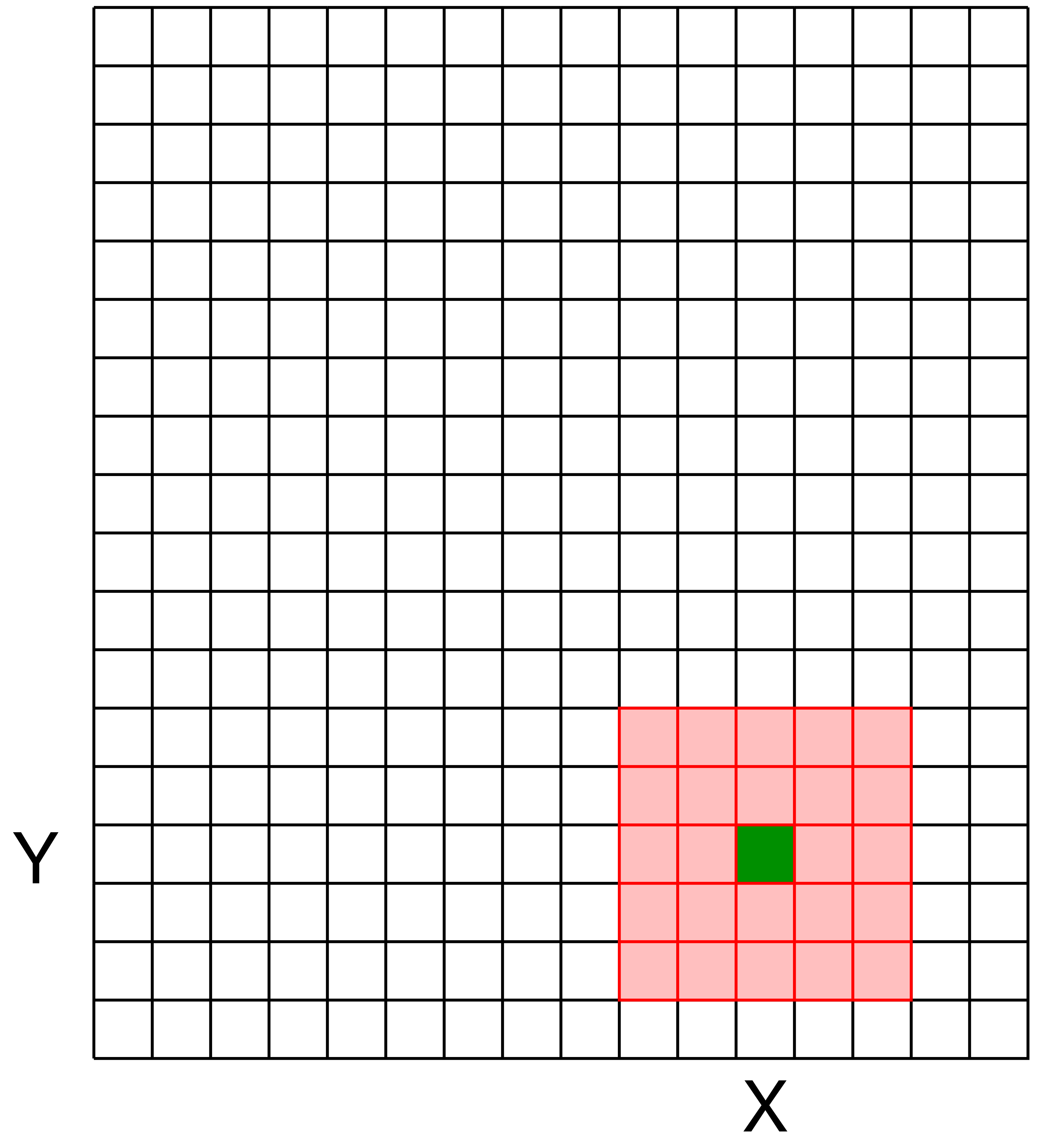
Compute the new pixel value, $I'(X, Y)$, as the sum of $m \times m$ values, where each value is the product of the original pixel value in $I(X, Y)$ and the corresponding values in the filter



Linear **Filters**

The computation is repeated for each

$$(X, Y)$$



Linear Filter **Example**

filter $F(X,Y)$

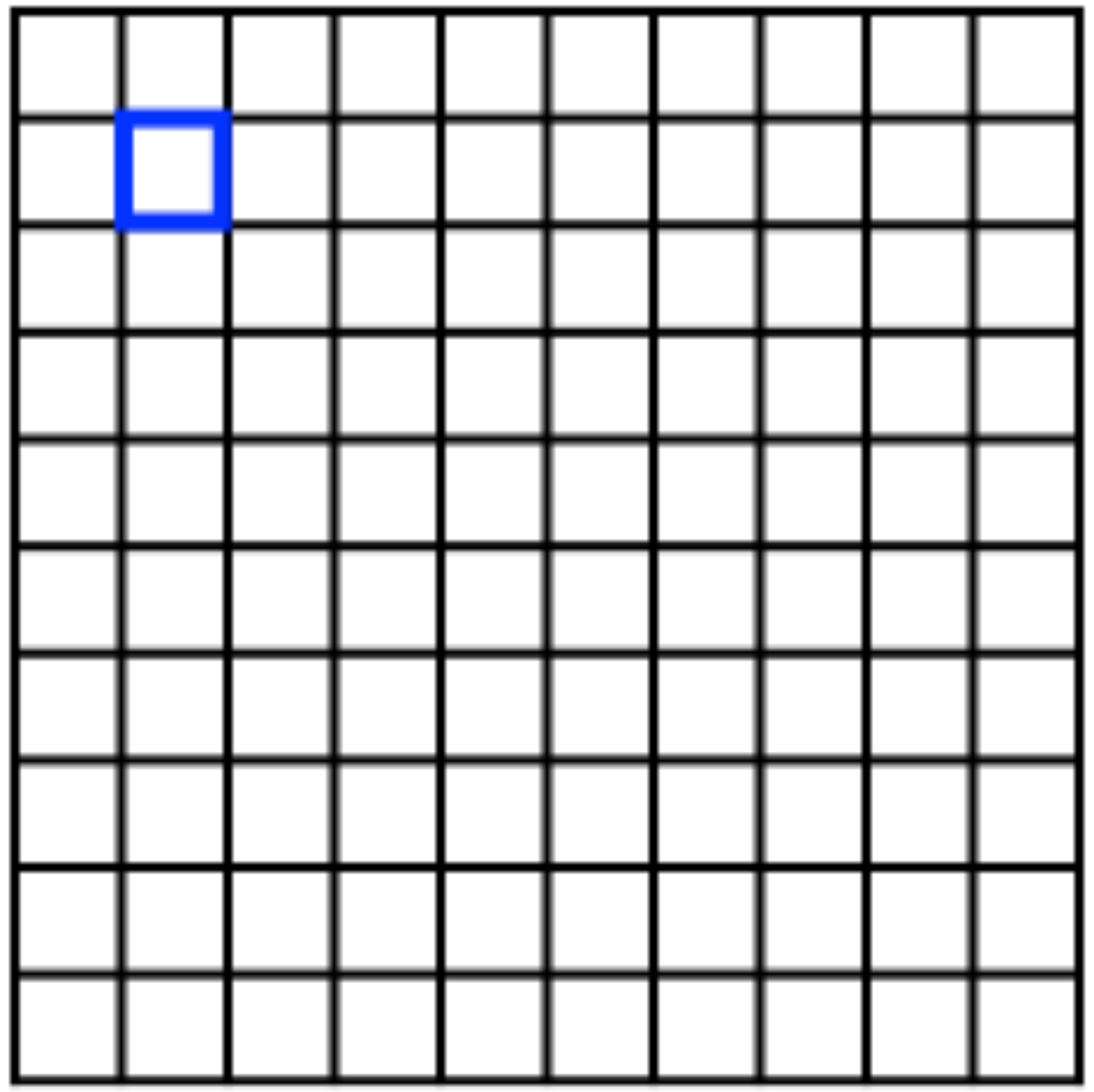
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$



$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$ filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

	0								

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

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

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

	0								

$$I'(X,Y) = \sum_{j=-k}^{k}\sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

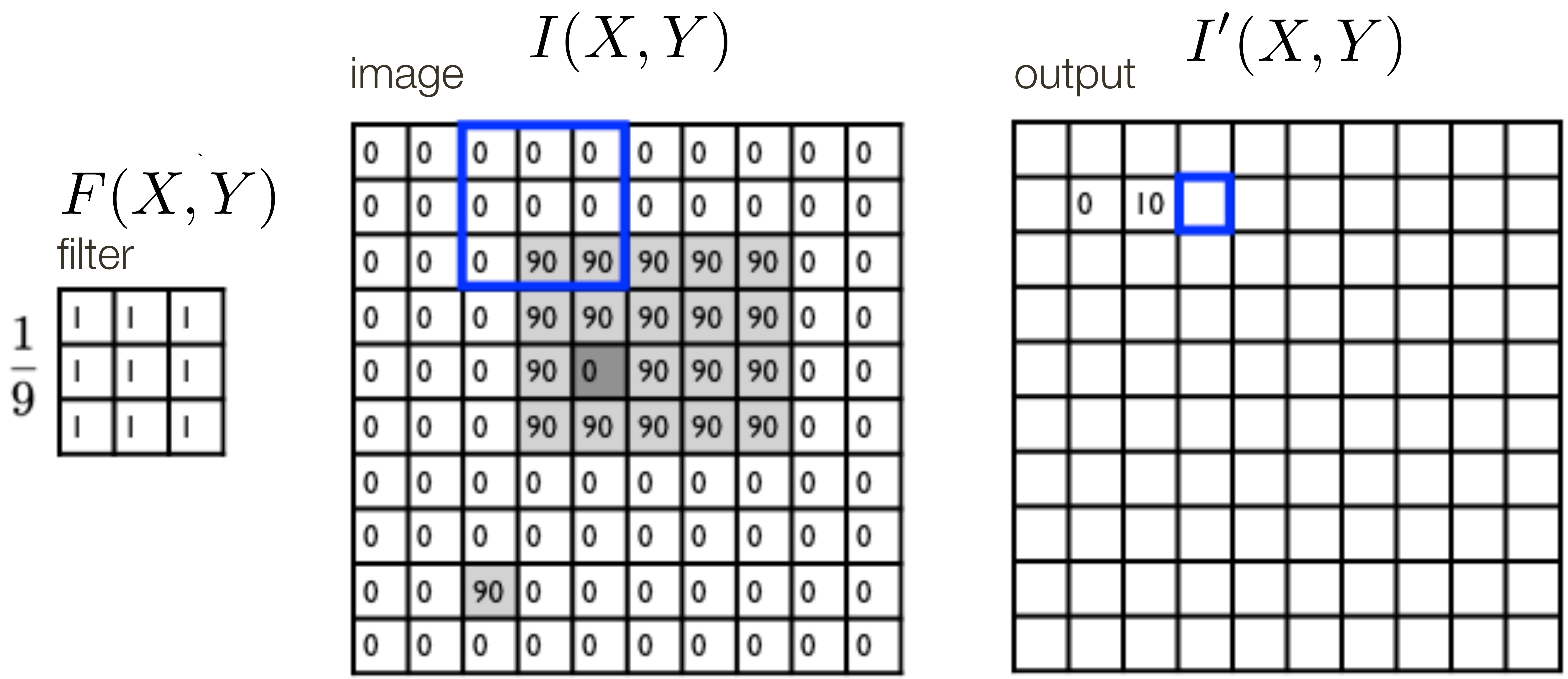
output $I'(X,Y)$

	0	10							

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



Filter $F(X,Y)$ ($\frac{1}{9}$ ×):

1	1	1
1	1	1
1	1	1

Image $I(X,Y)$:

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	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	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

Output $I'(X,Y)$:

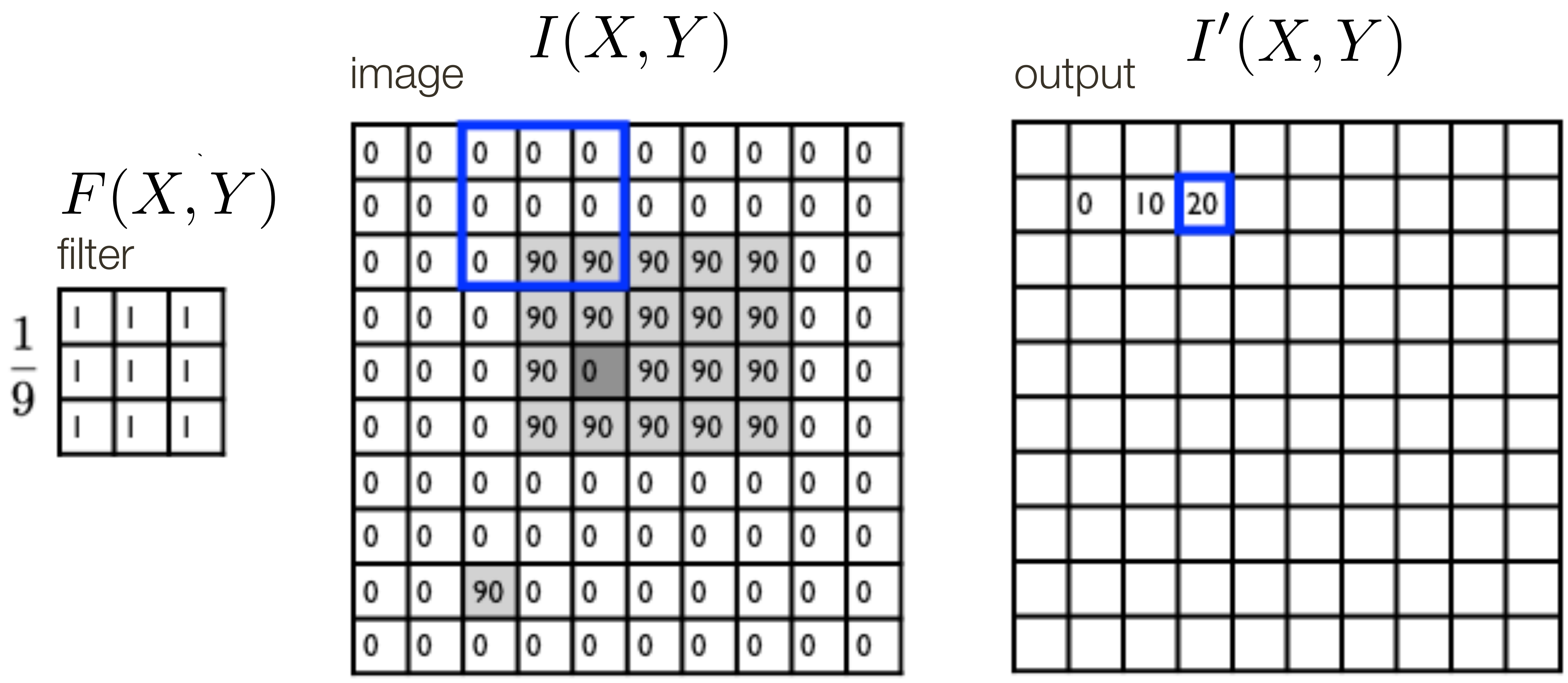
	0	10							

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output | filter | image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

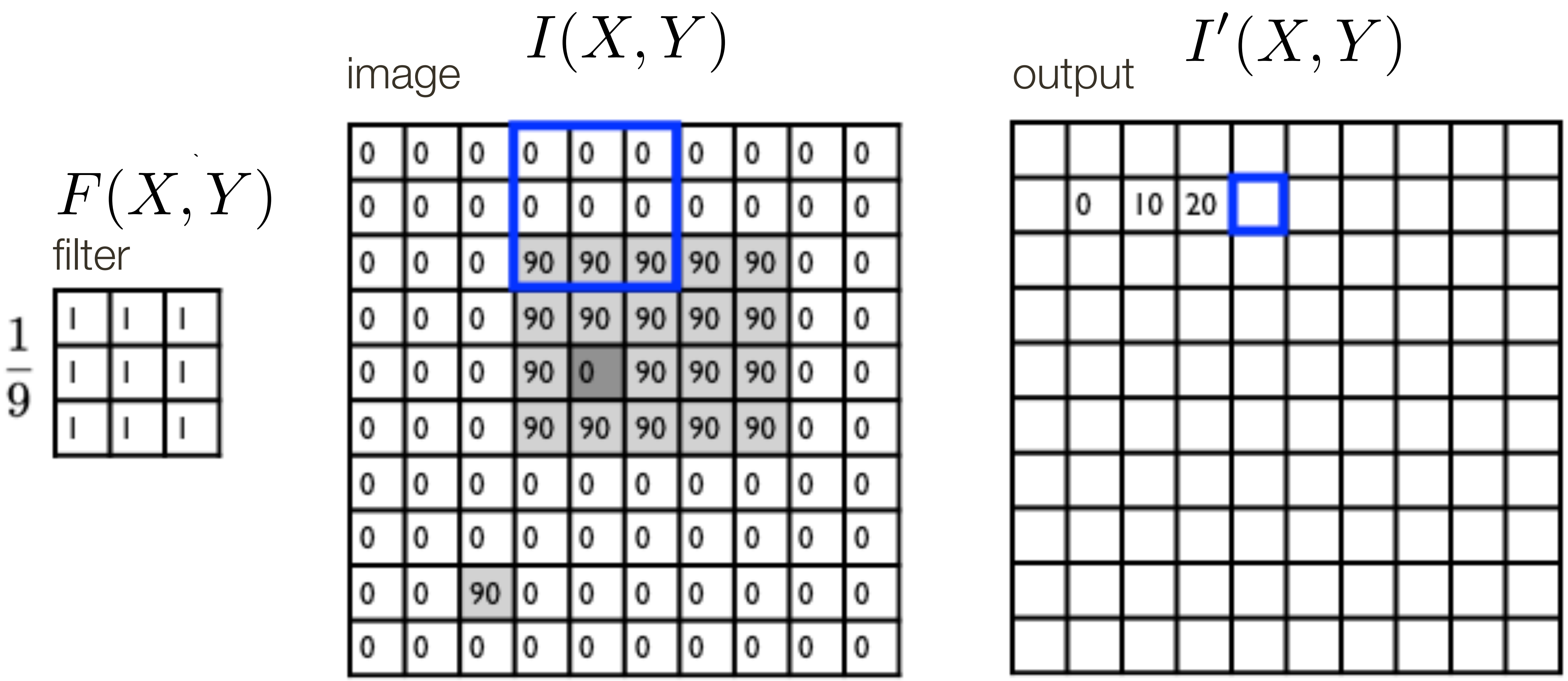
output

filter

image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output | filter | image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

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

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

	0	10	20	30					

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output · filter · image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

	0	10	20	30	30				

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output · filter · image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$ ×

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

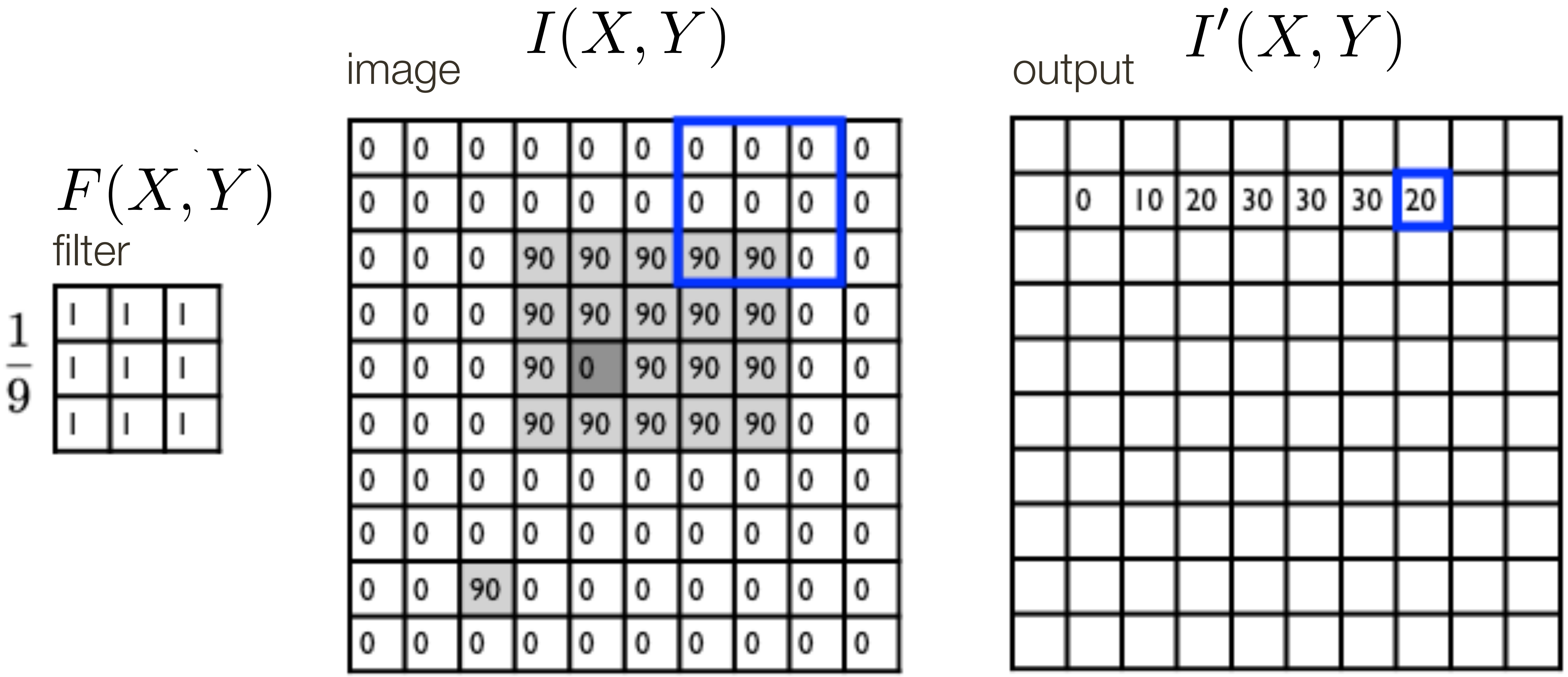
output $I'(X,Y)$

	0	10	20	30	30	30			

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k}\sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}}\, \underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

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

image $I(X,Y)$

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	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	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

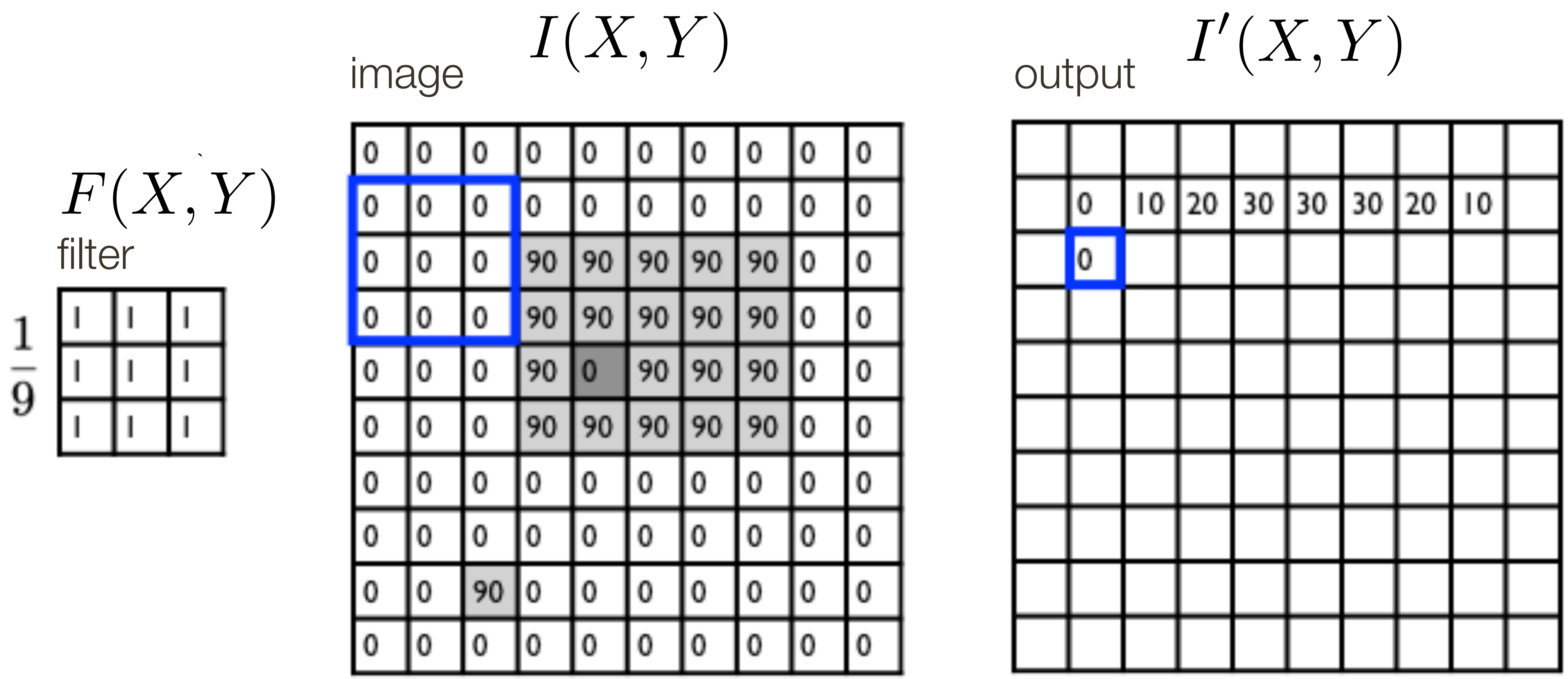
output $I'(X,Y)$

	0	10	20	30	30	30	20	10	

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k}\sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}}\, \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

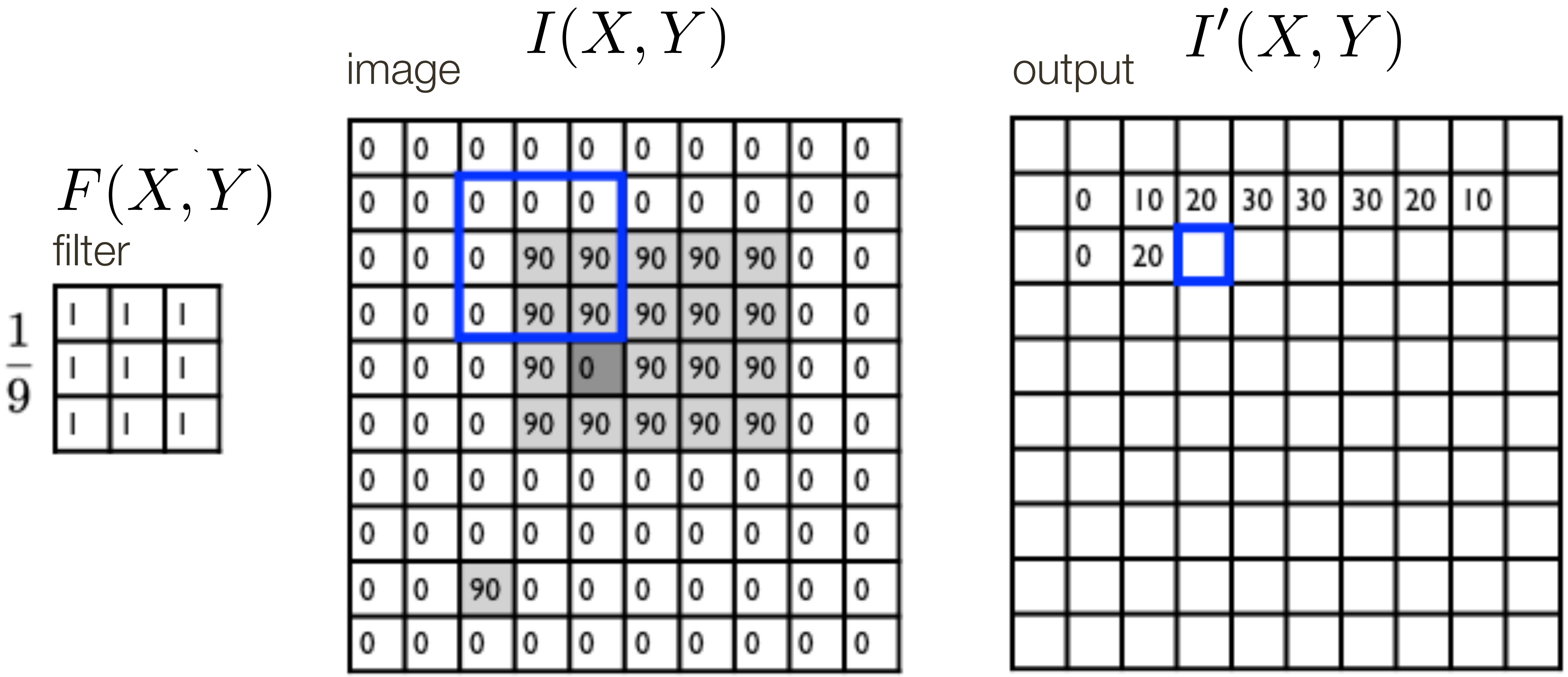
Linear Filter **Example**



$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**



$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k}\sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}}\,\underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40						

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) \, I(X+i, Y+j)$$

output | filter | image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$ ×

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0								

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k}\sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}}\,\underbrace{I(X+i,Y+j)}_{\text{image (signal)}}$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X, Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X, Y)$

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	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	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

output $I'(X, Y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30							

$$I'(X, Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i, j) \, I(X + i, Y + j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

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

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output | filter | image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

$F(X,Y)$

filter

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

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

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output — filter — image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filter **Example**

filter $F(X,Y)$

$\frac{1}{9}$ ×

1	1	1
1	1	1
1	1	1

image $I(X,Y)$

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	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	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

output $I'(X,Y)$

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

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

output · filter · image (signal)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear **Filters**

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

For a give X and Y, superimpose the filter on the image centered at (X,Y)

Compute the new pixel value, $I'(X,Y)$, as the sum of $m \times m$ values, where each value is the product of the original pixel value in $I(X,Y)$ and the corresponding values in the filter

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

At each pixel, (X,Y), there are $m \times m$ multiplications

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

At each pixel, (X, Y), there are $m \times m$ multiplications

There are $n \times n$ pixels in (X, Y)

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

At each pixel, (X,Y), there are $m \times m$ multiplications

There are $n \times n$ pixels in (X,Y)

Total: $m^2 \times n^2$ multiplications

Linear **Filters**

Let's do some accounting …

$$\underbrace{I'(X,Y)}_{\text{output}} = \sum_{j=-k}^{k} \sum_{i=-k}^{k} \underbrace{F(i,j)}_{\text{filter}} \underbrace{I(X+i, Y+j)}_{\text{image (signal)}}$$

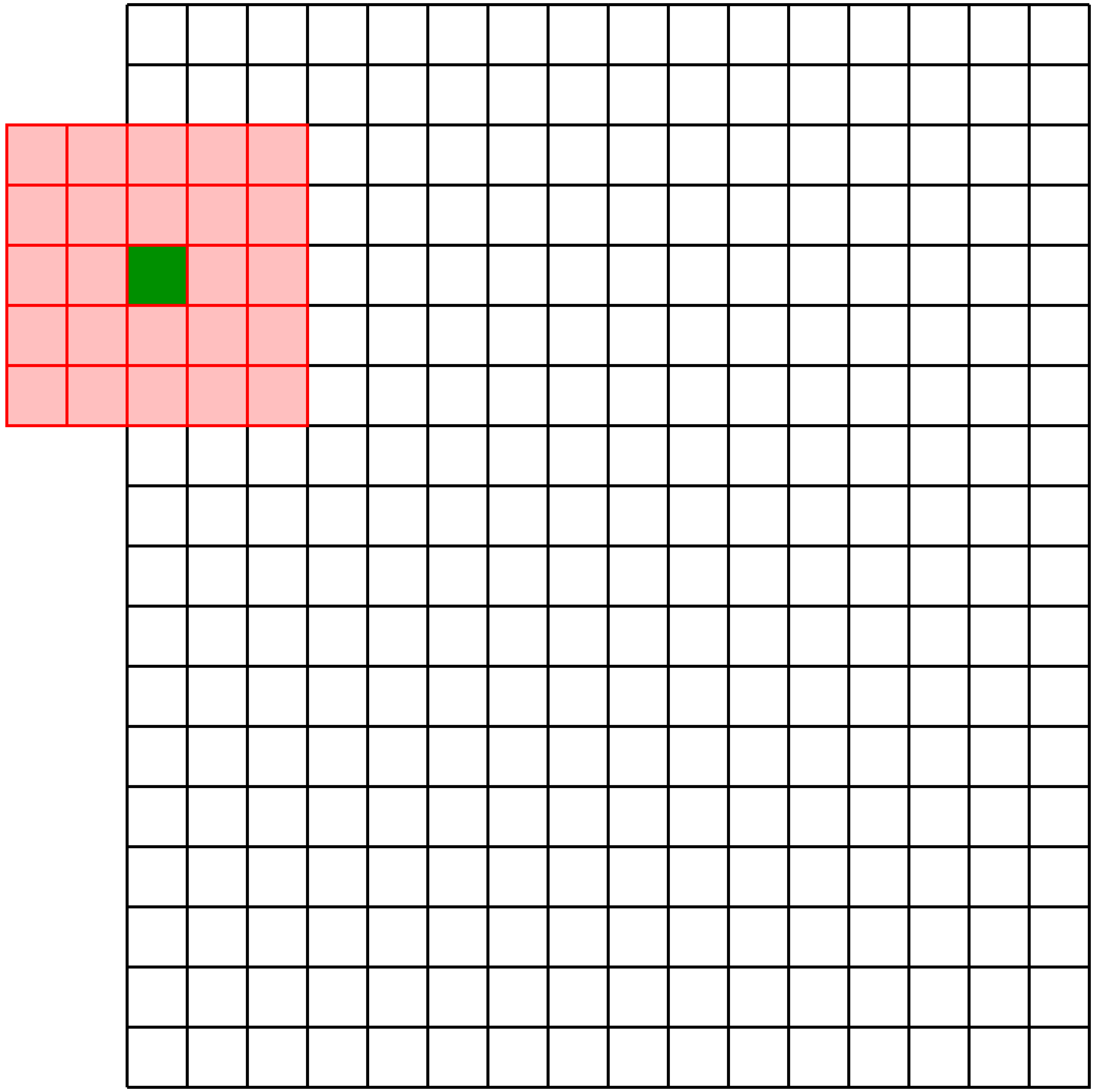
At each pixel, (X, Y), there are $m \times m$ multiplications

There are $n \times n$ pixels in (X, Y)

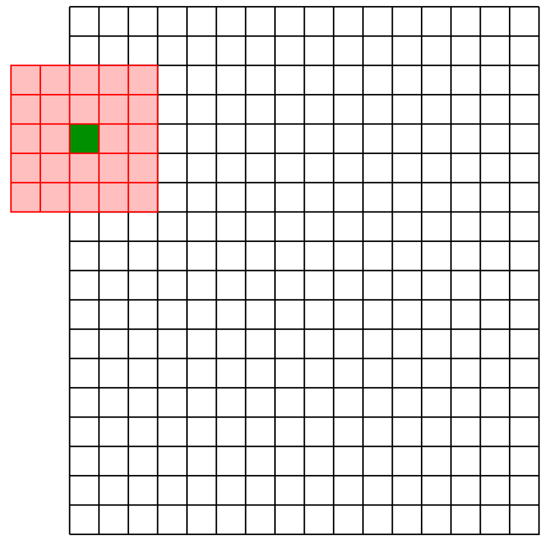
Total: $m^2 \times n^2$ multiplications

When m is fixed, small constant, this is $\mathcal{O}(n^2)$. But when $m \approx n$ this is $\mathcal{O}(m^4)$.

Linear Filters: **Boundary** Effects



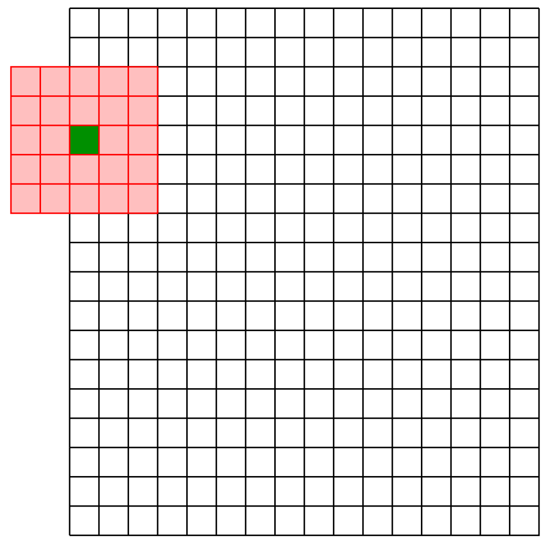
Linear Filters: **Boundary** Effects



Three standard ways to deal with boundaries:

1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

Linear Filters: **Boundary** Effects

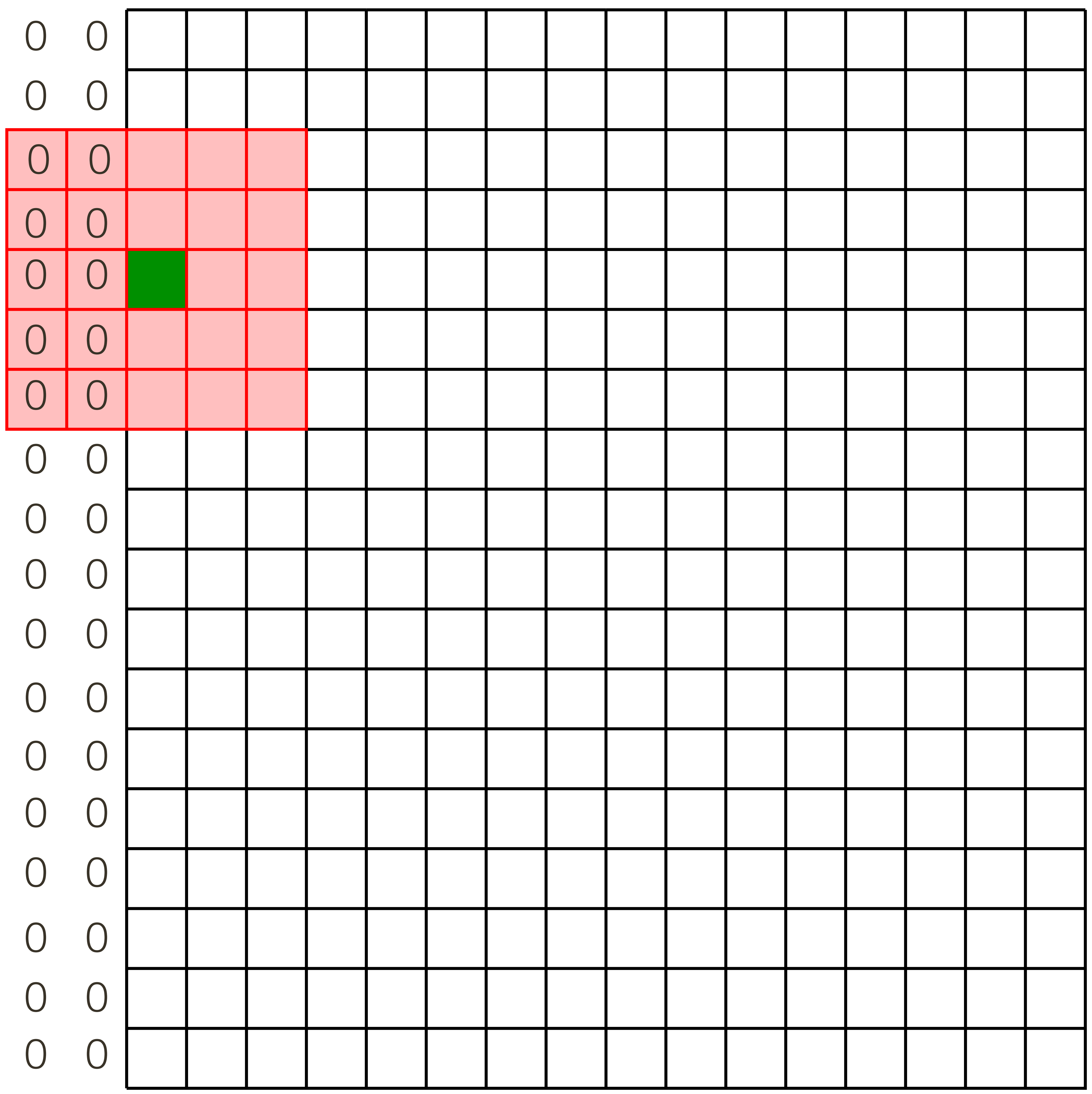


Three standard ways to deal with boundaries:

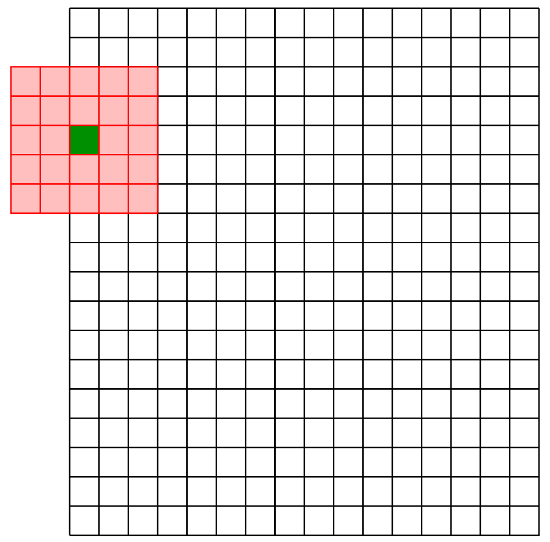
1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects



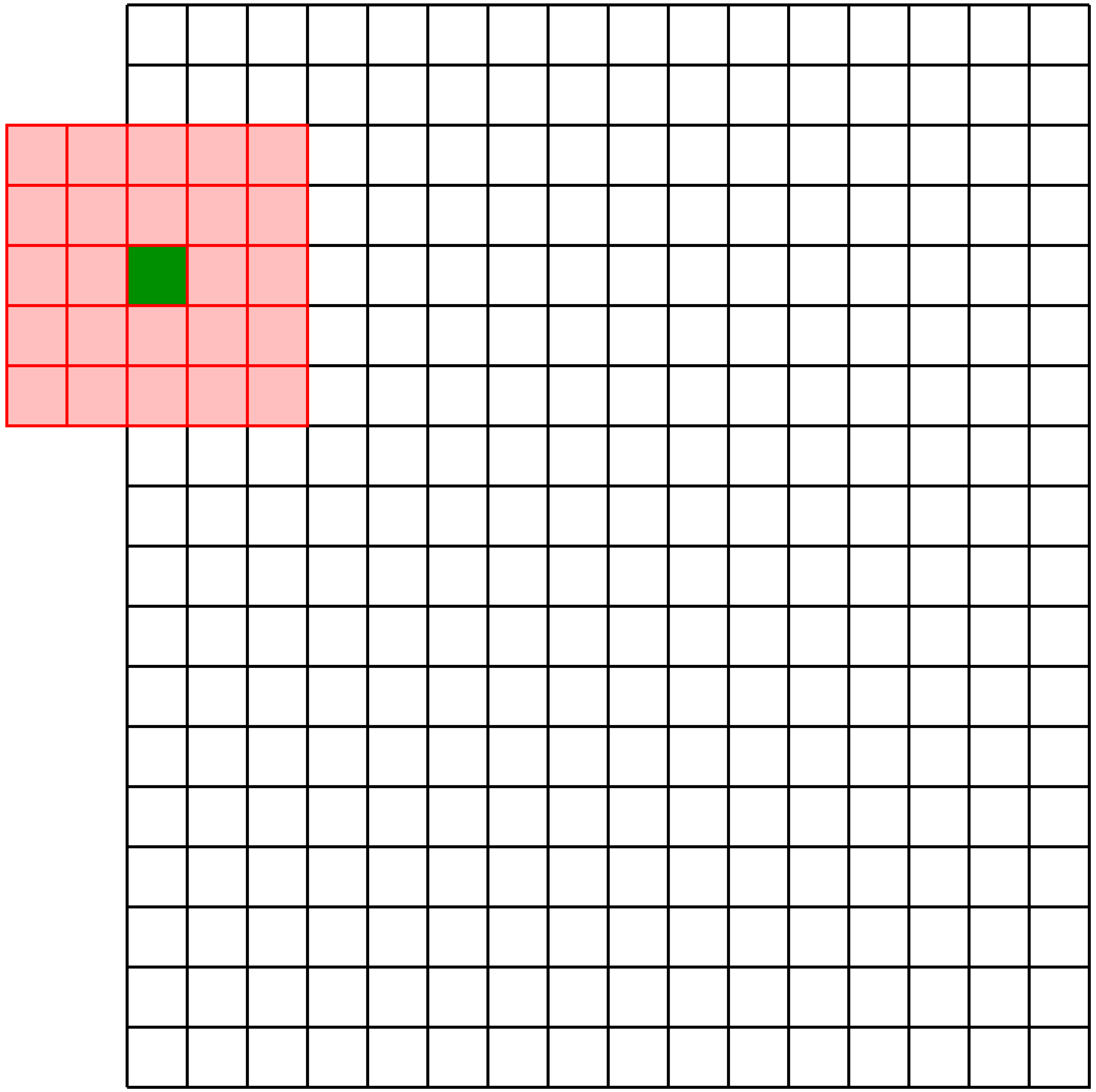
Four standard ways to deal with boundaries:

1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

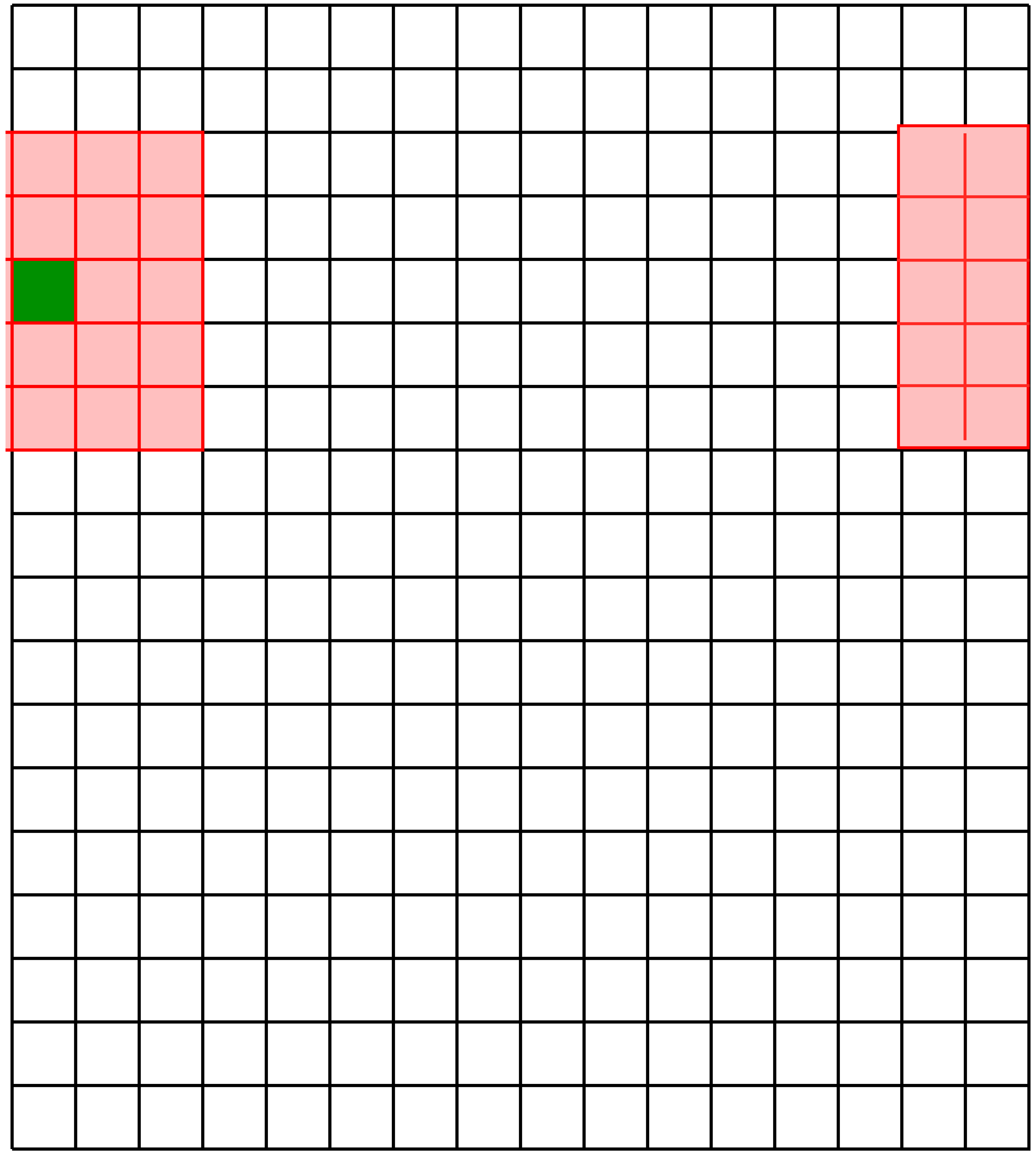
2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

3. **Assume periodicity**: The top row wraps around to the bottom row; the leftmost column wraps around to the rightmost column

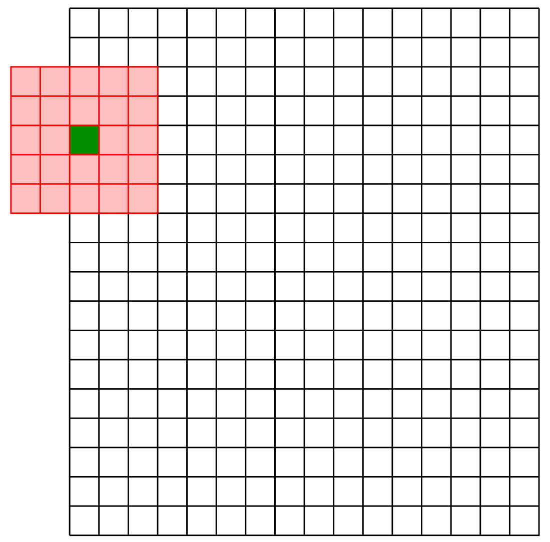
Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects



Four standard ways to deal with boundaries:

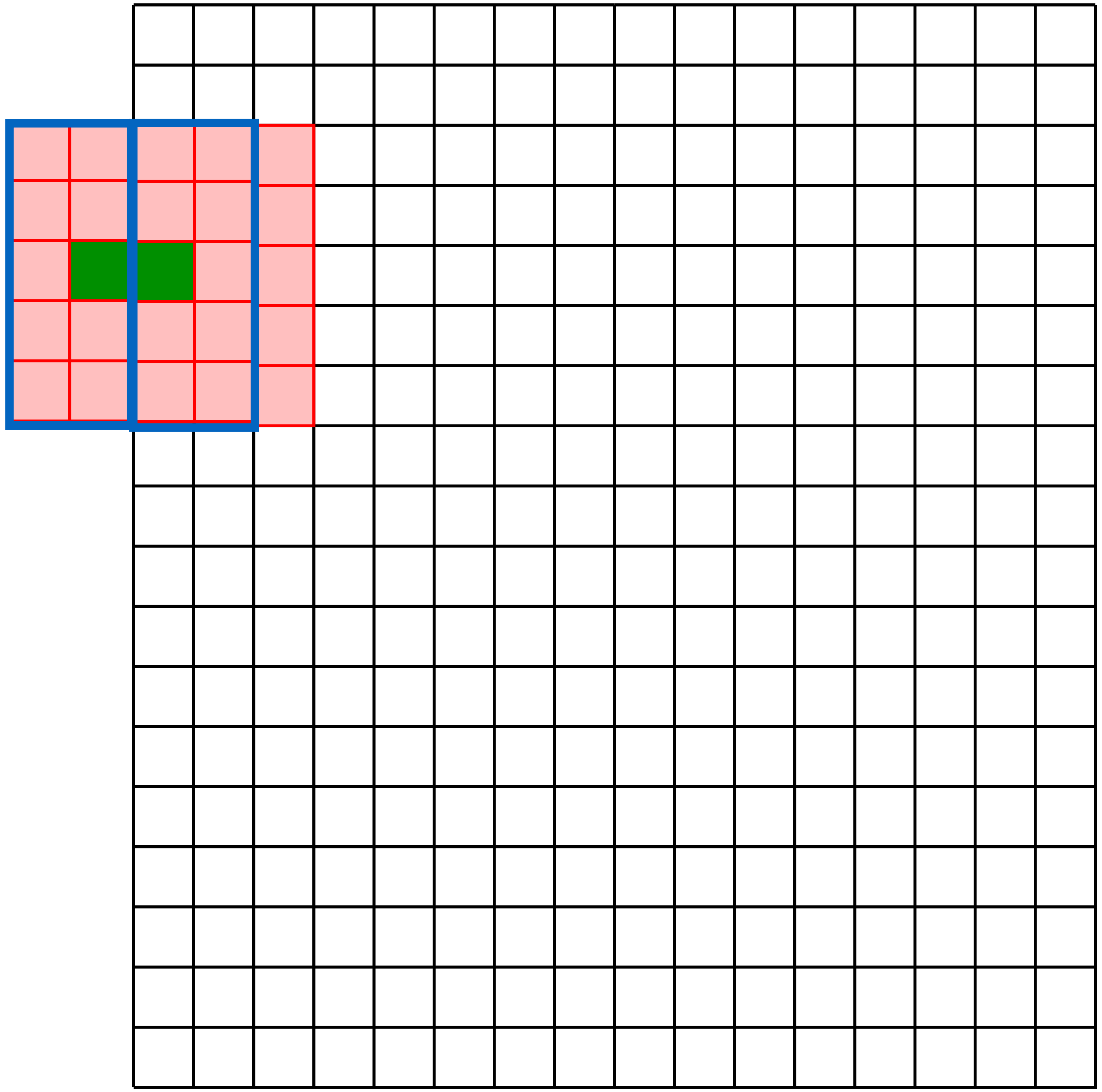
1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

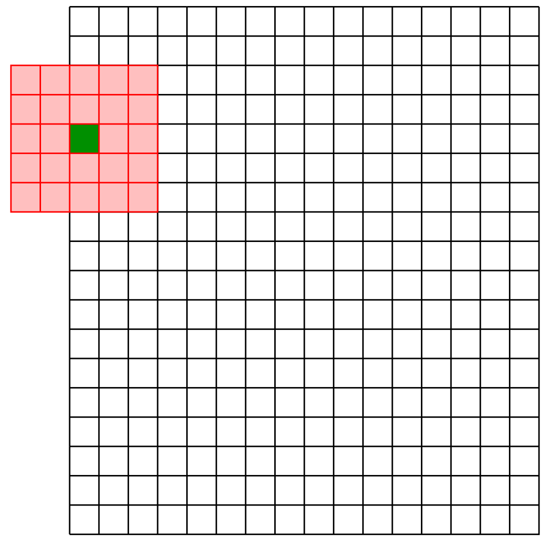
3. **Assume periodicity**: The top row wraps around to the bottom row; the leftmost column wraps around to the rightmost column

4. **Reflect boarder**: Copy rows/columns locally by reflecting over the edge

Linear Filters: **Boundary** Effects



Linear Filters: **Boundary** Effects



Four standard ways to deal with boundaries:

1. **Ignore these locations:** Make the computation undefined for the top and bottom k rows and the leftmost and rightmost k columns

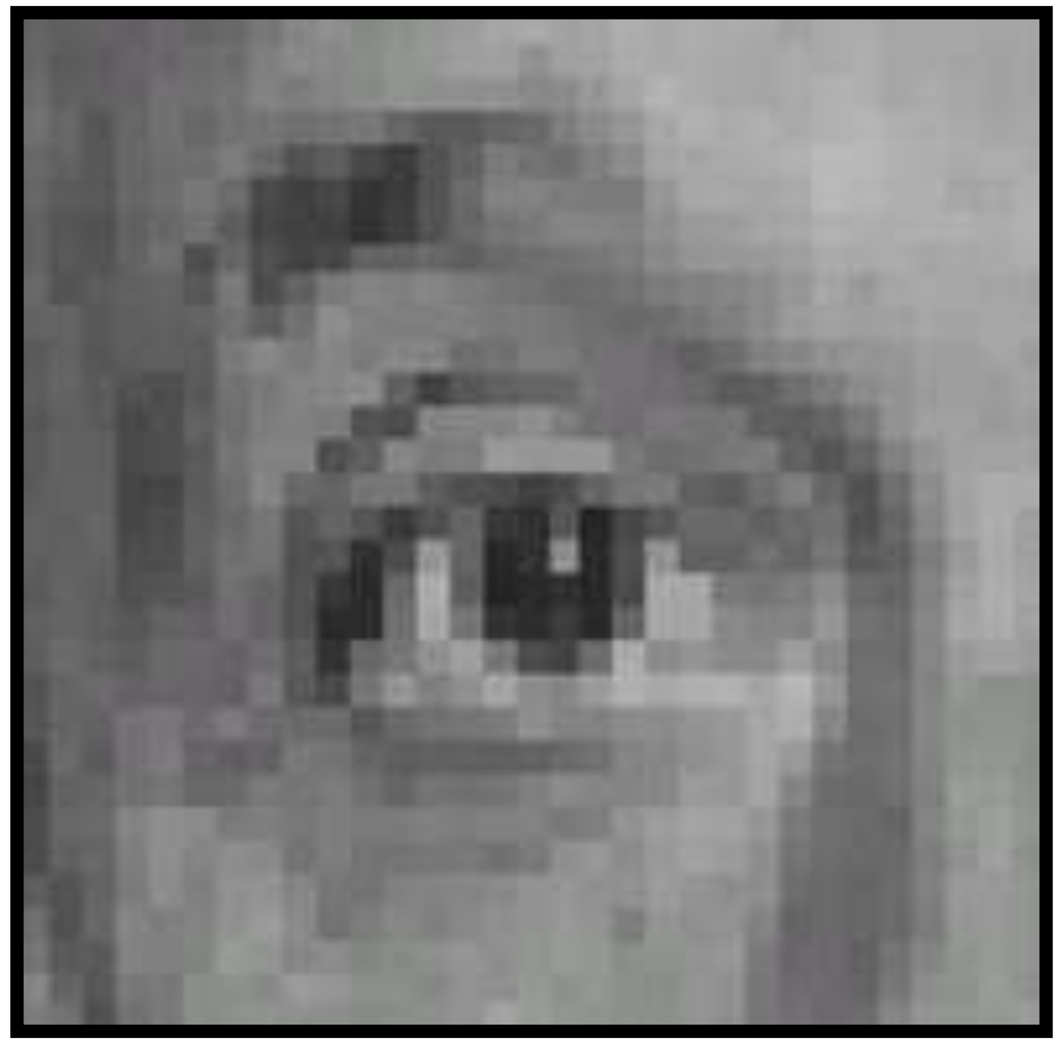
2. **Pad the image with zeros**: Return zero whenever a value of I is required at some position outside the defined limits of X and Y

3. **Assume periodicity**: The top row wraps around to the bottom row; the leftmost column wraps around to the rightmost column

4. **Reflect boarder**: Copy rows/columns locally by reflecting over the edge

A short exercise …

Example 1: Warm up



Original

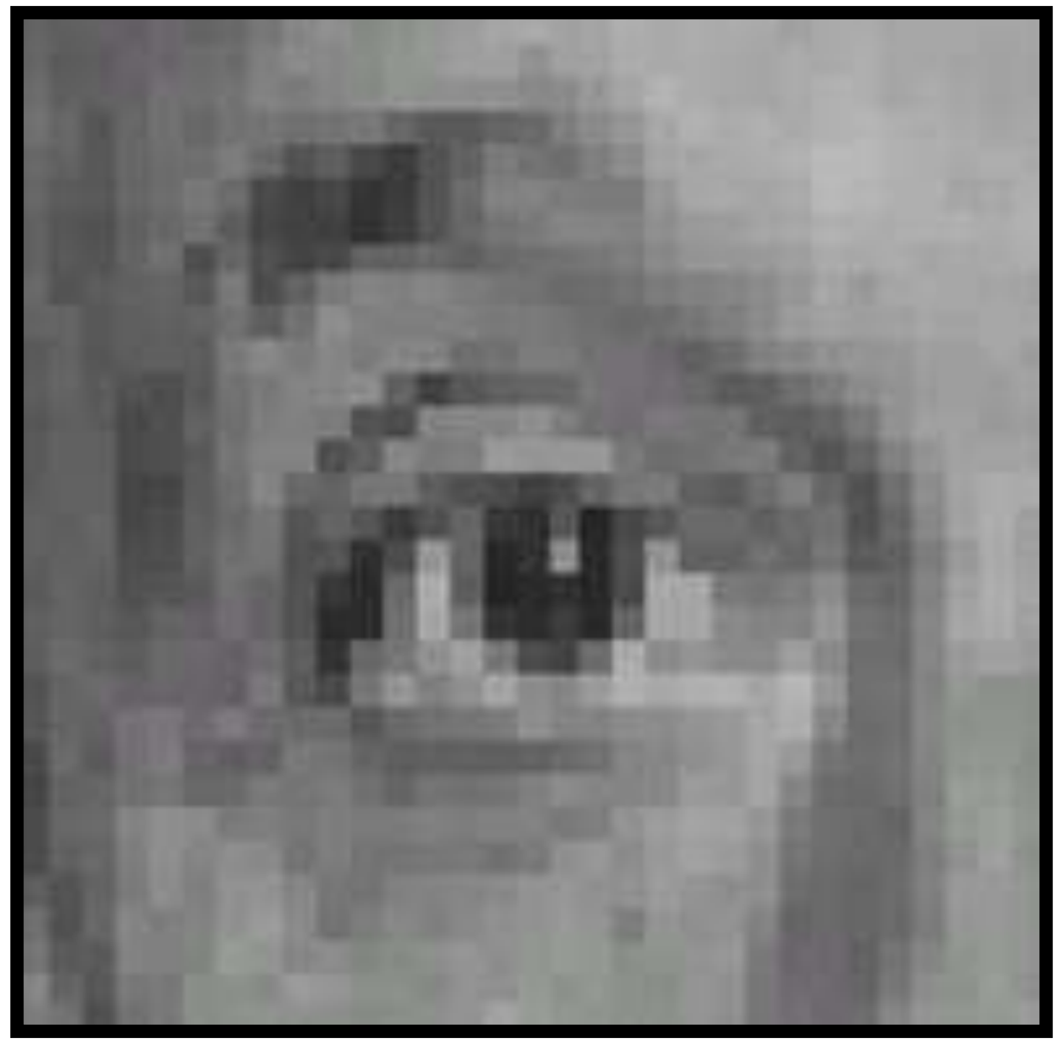
0	0	0
0	1	0
0	0	0

Filter

?

Result

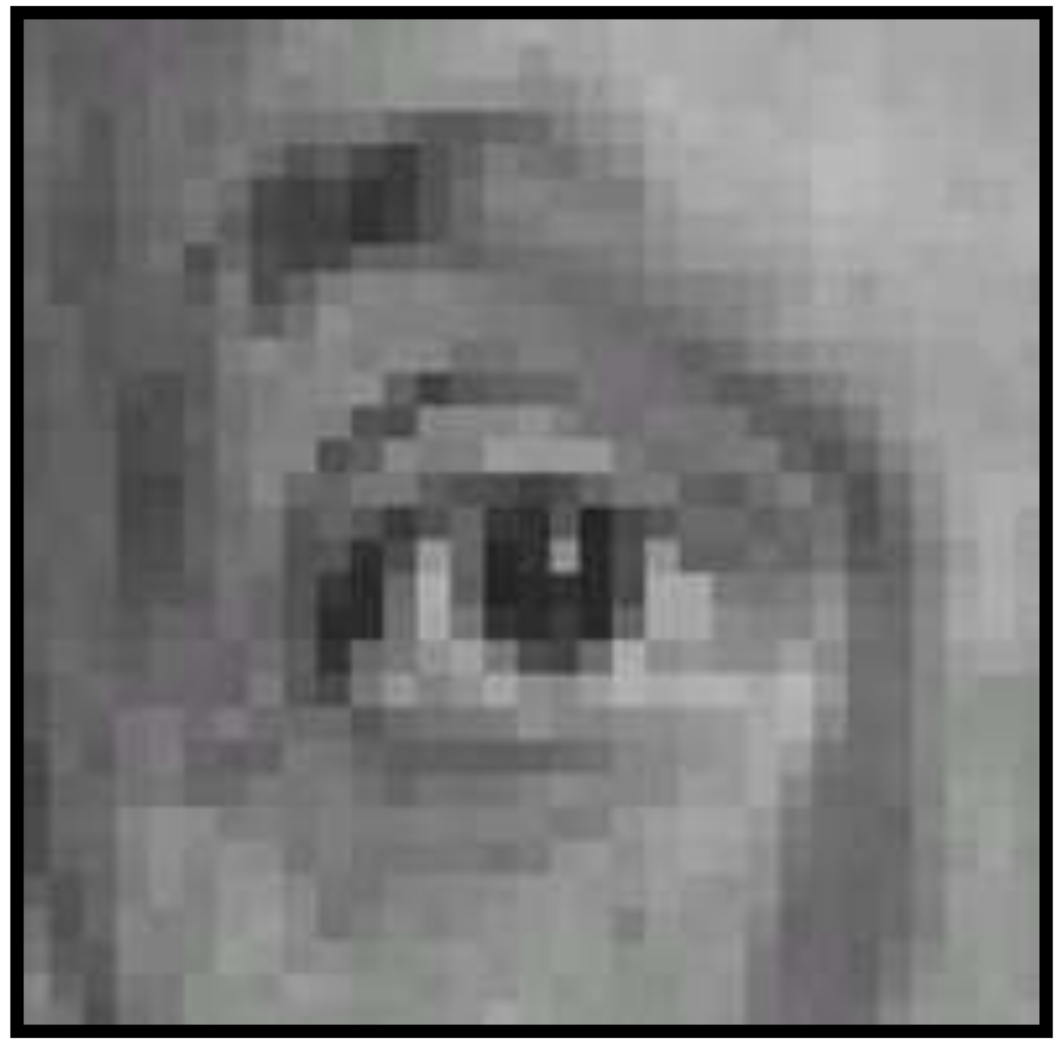
Example 1: Warm up



Original

0	0	0
0	1	0
0	0	0

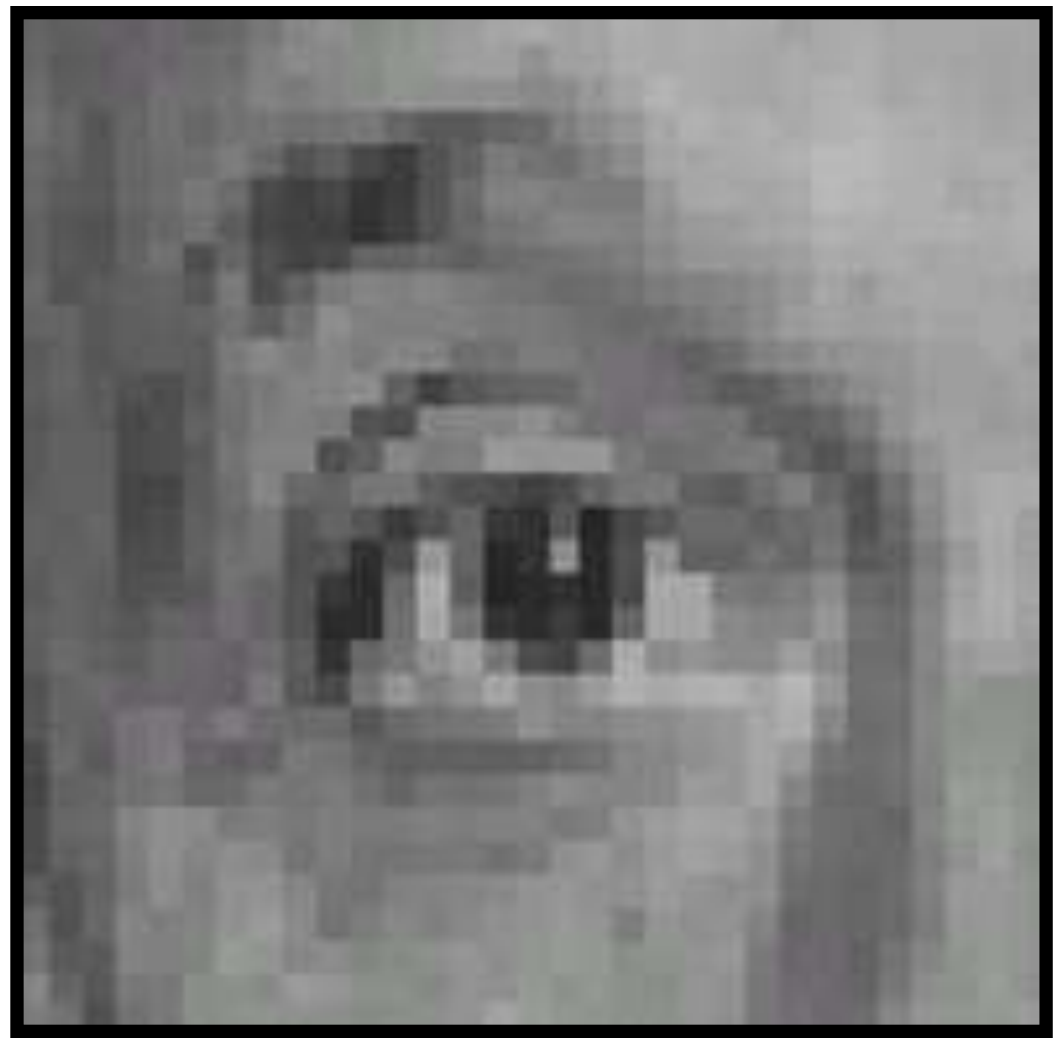
Filter



Result

(no change)

Example 2:



Original

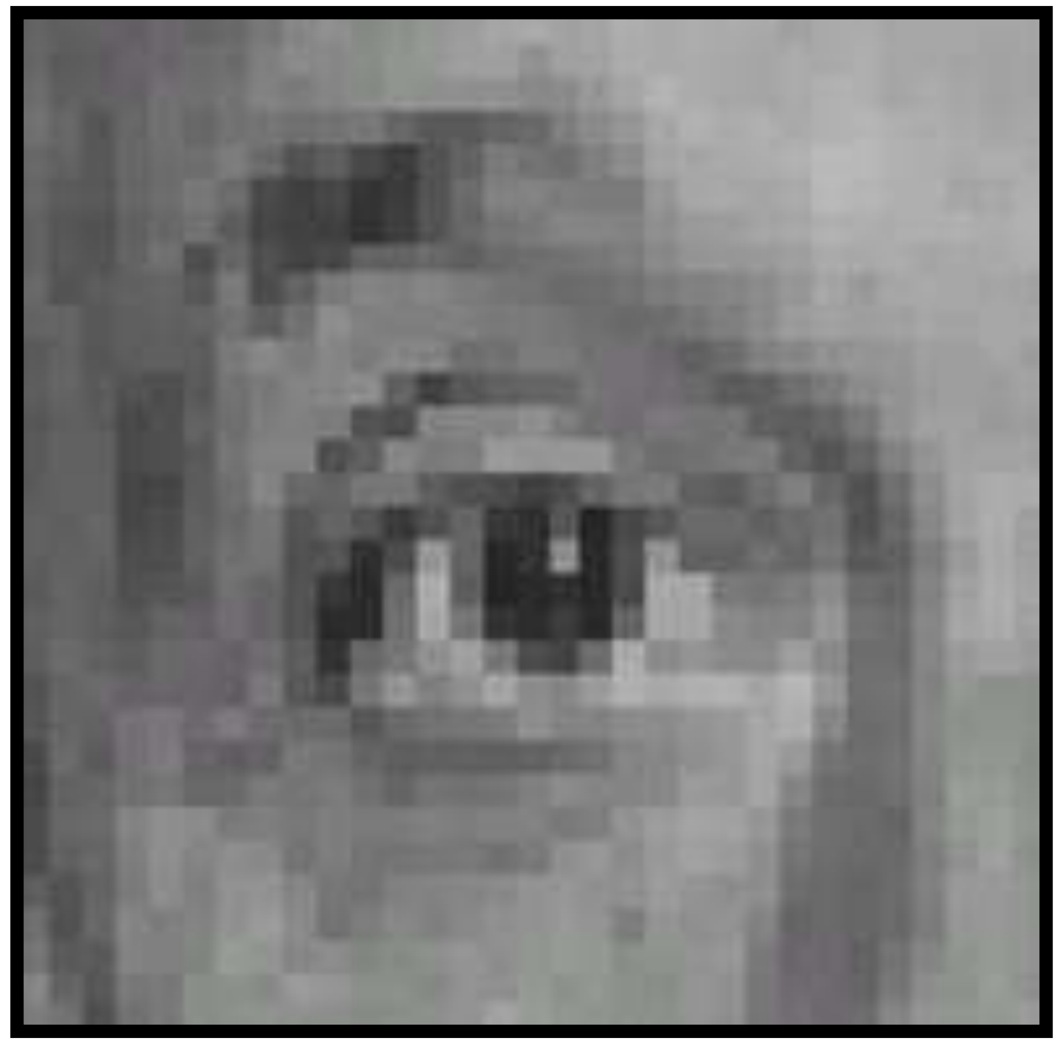
0	0	0
0	0	1
0	0	0

Filter

?

Result

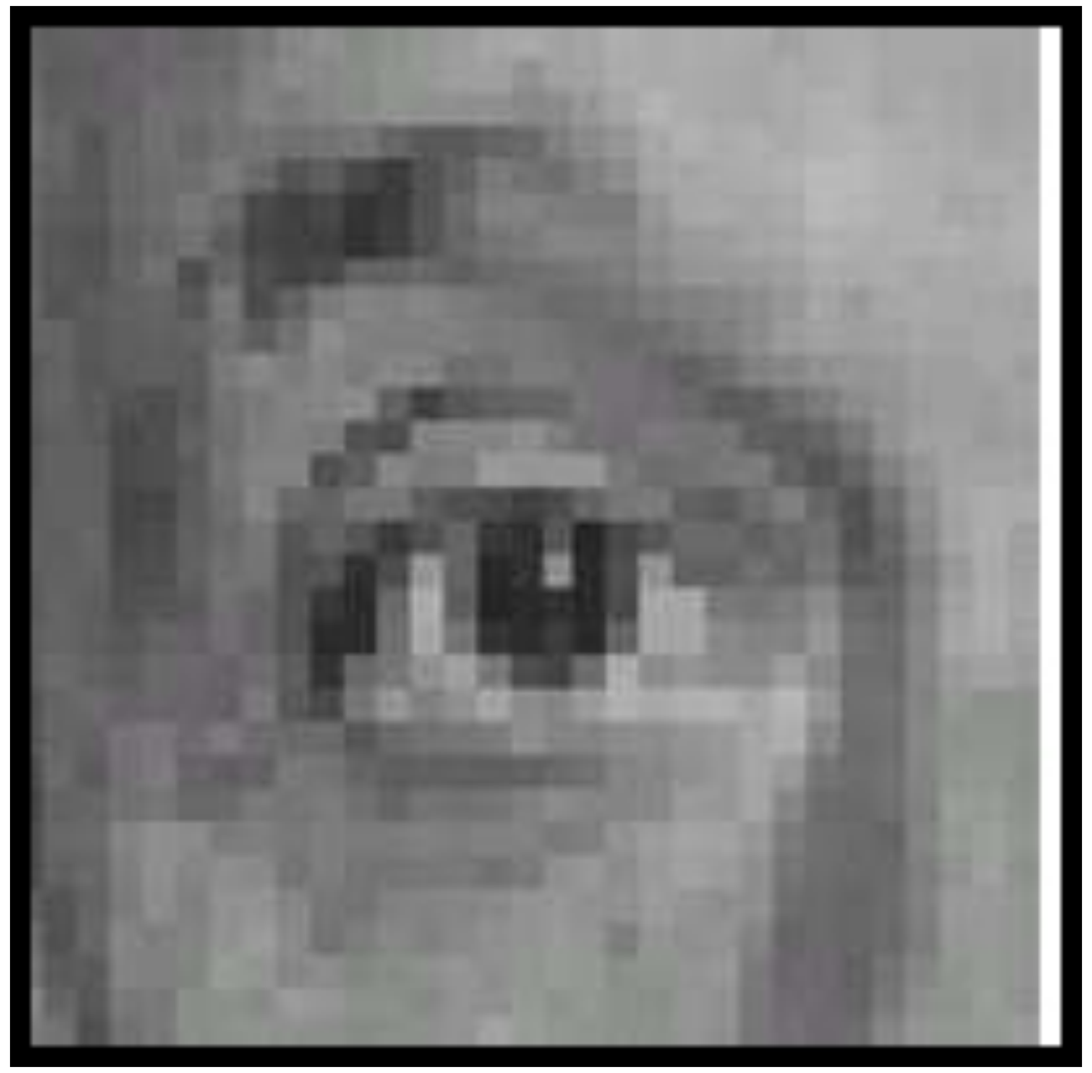
Example 2:



Original

0	0	0
0	0	1
0	0	0

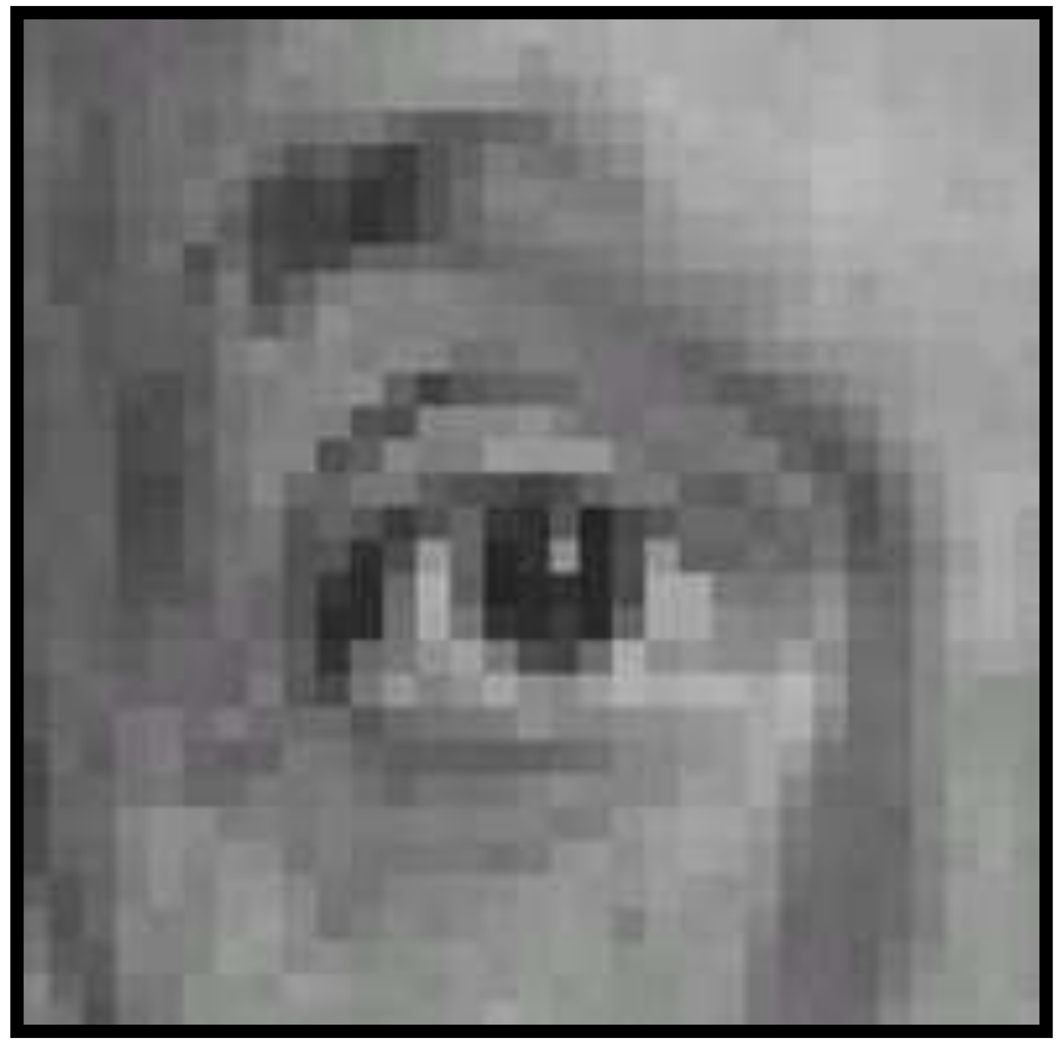
Filter



Result

(sift left by 1 pixel)

Example 3:



Original

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

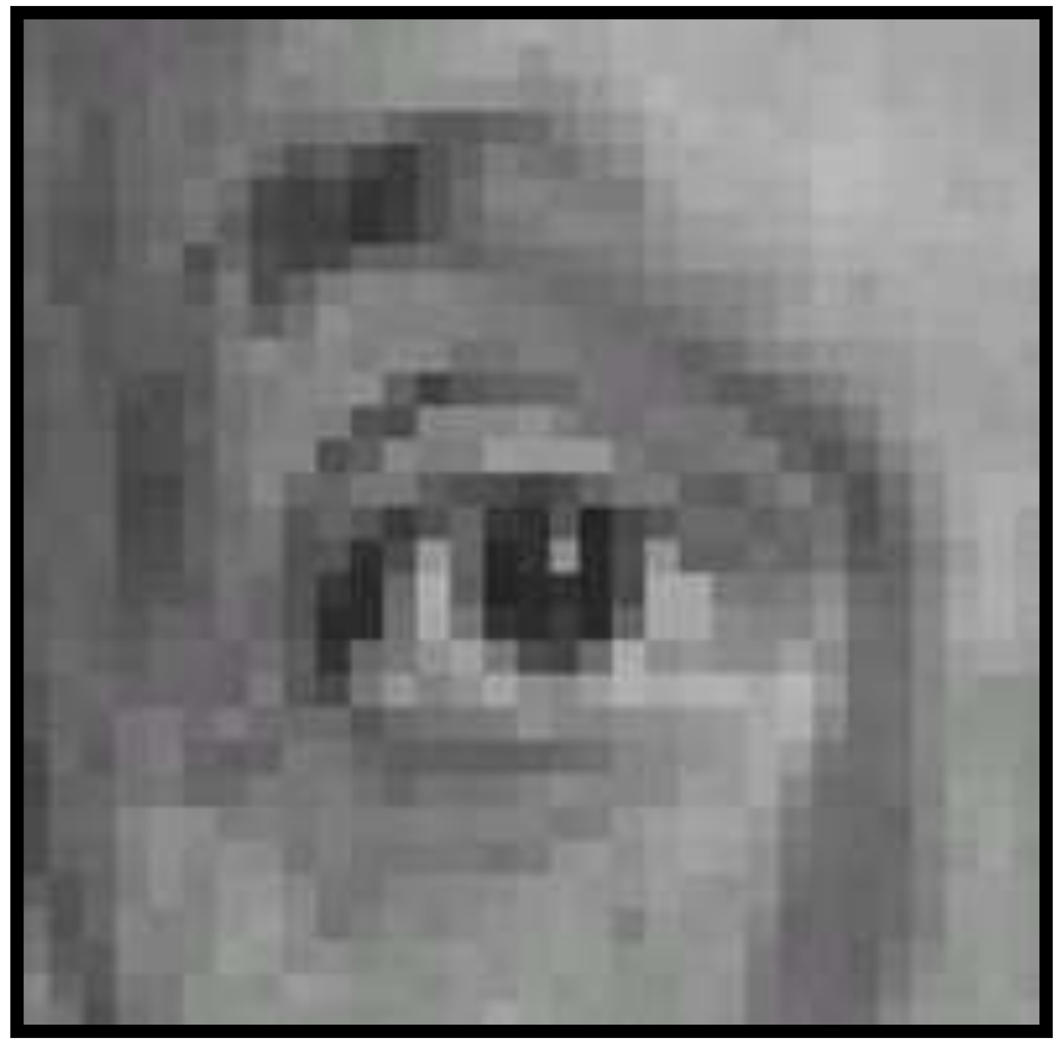
Filter

(filter sums to 1)

?

Result

Example 3:

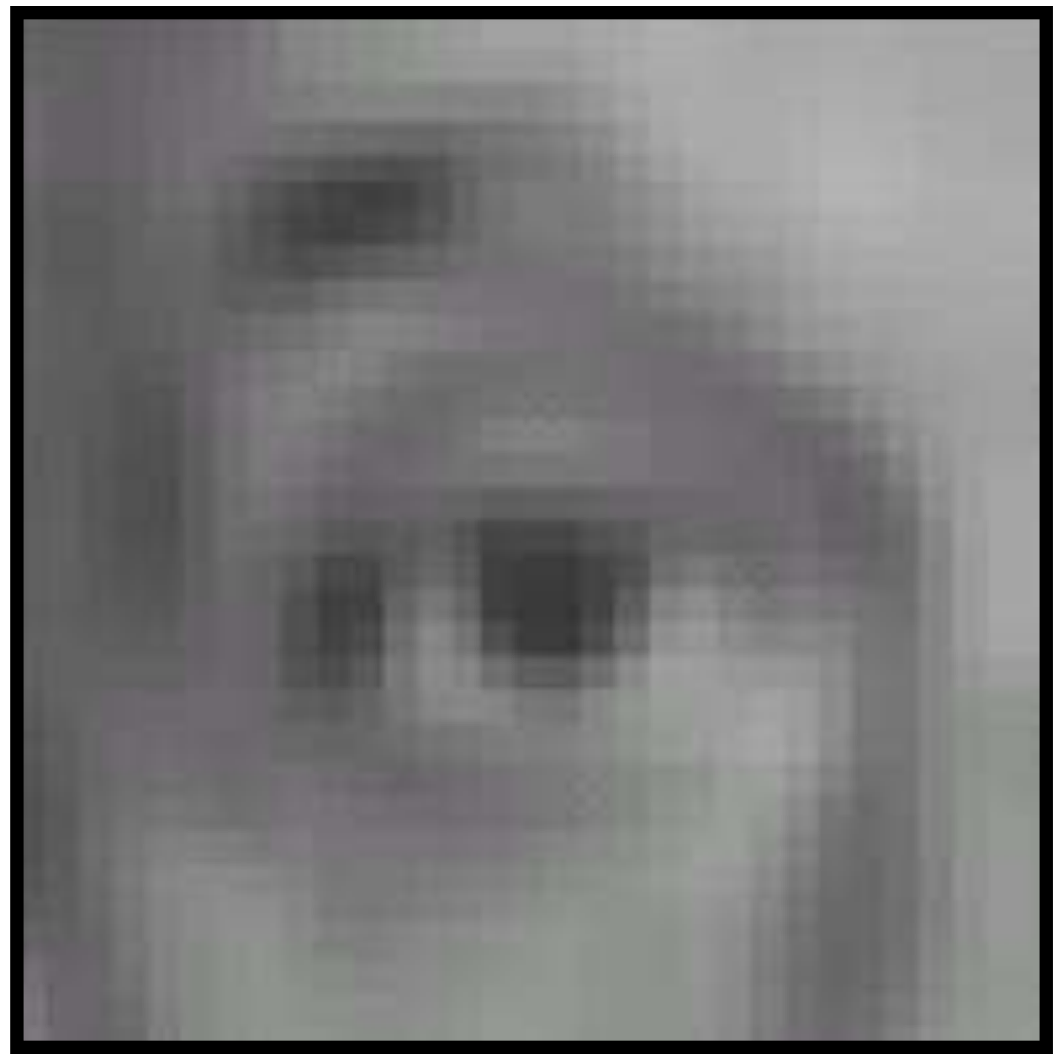


Original

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

Filter

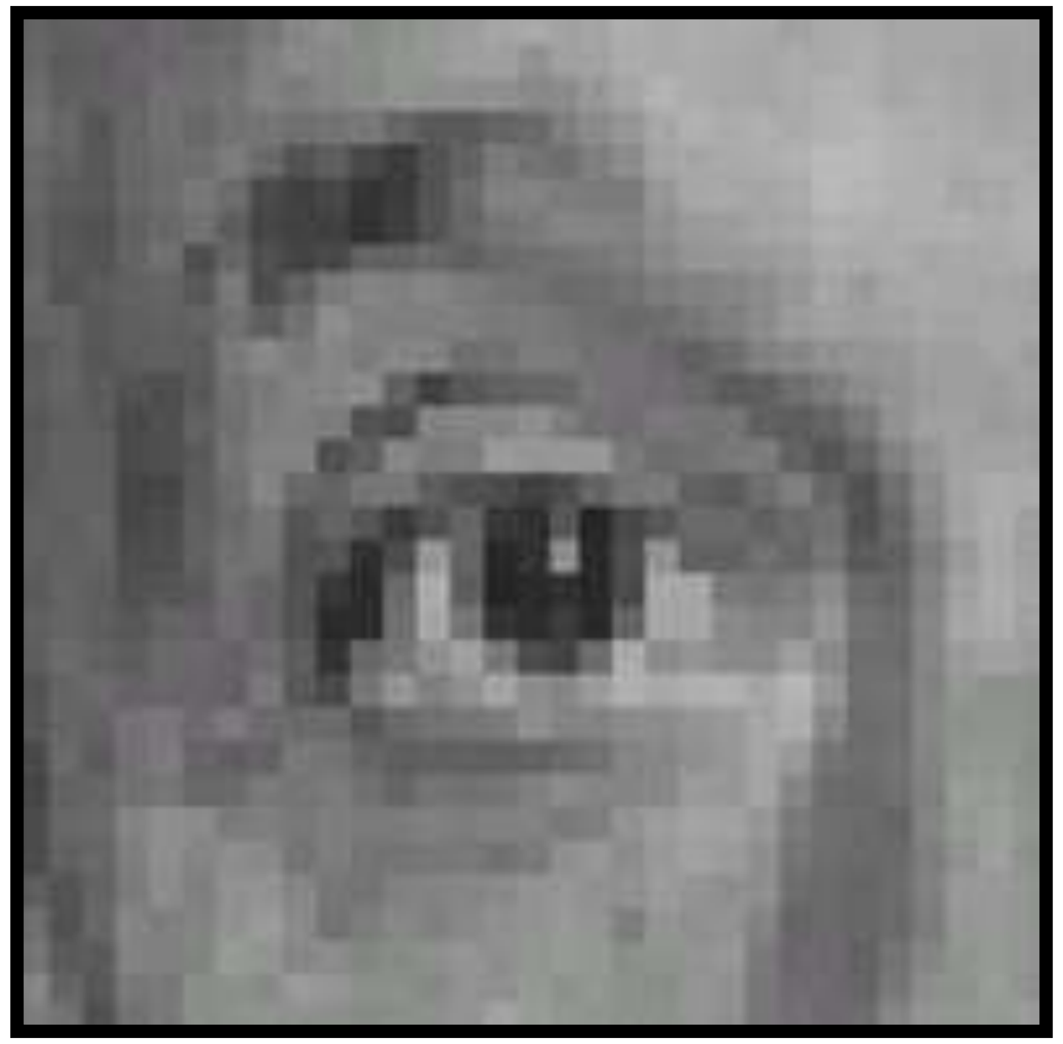
(filter sums to 1)



Result

(blur with a box filter)

Example 4:



Original

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

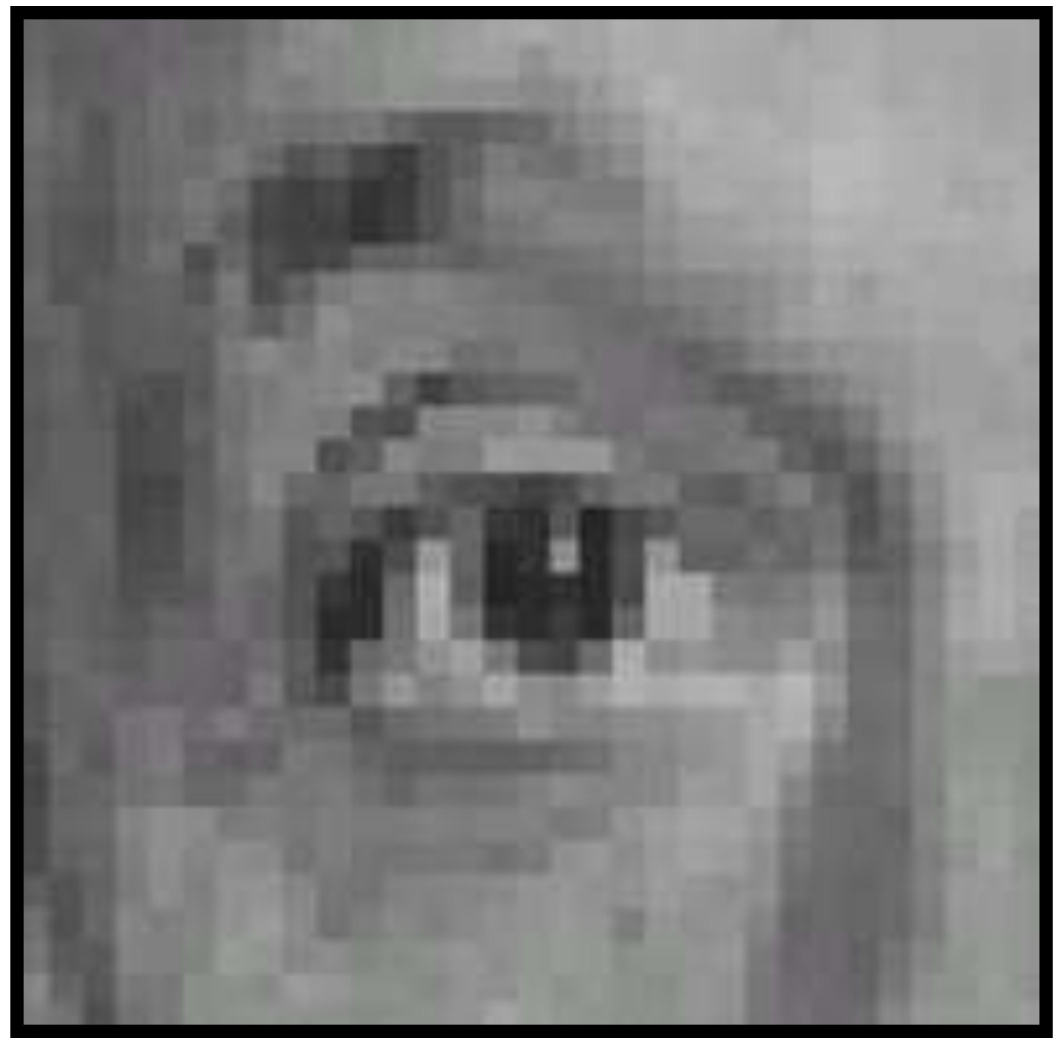
Filter

(filter sums to 1)

?

Result

Example 4:



Original

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

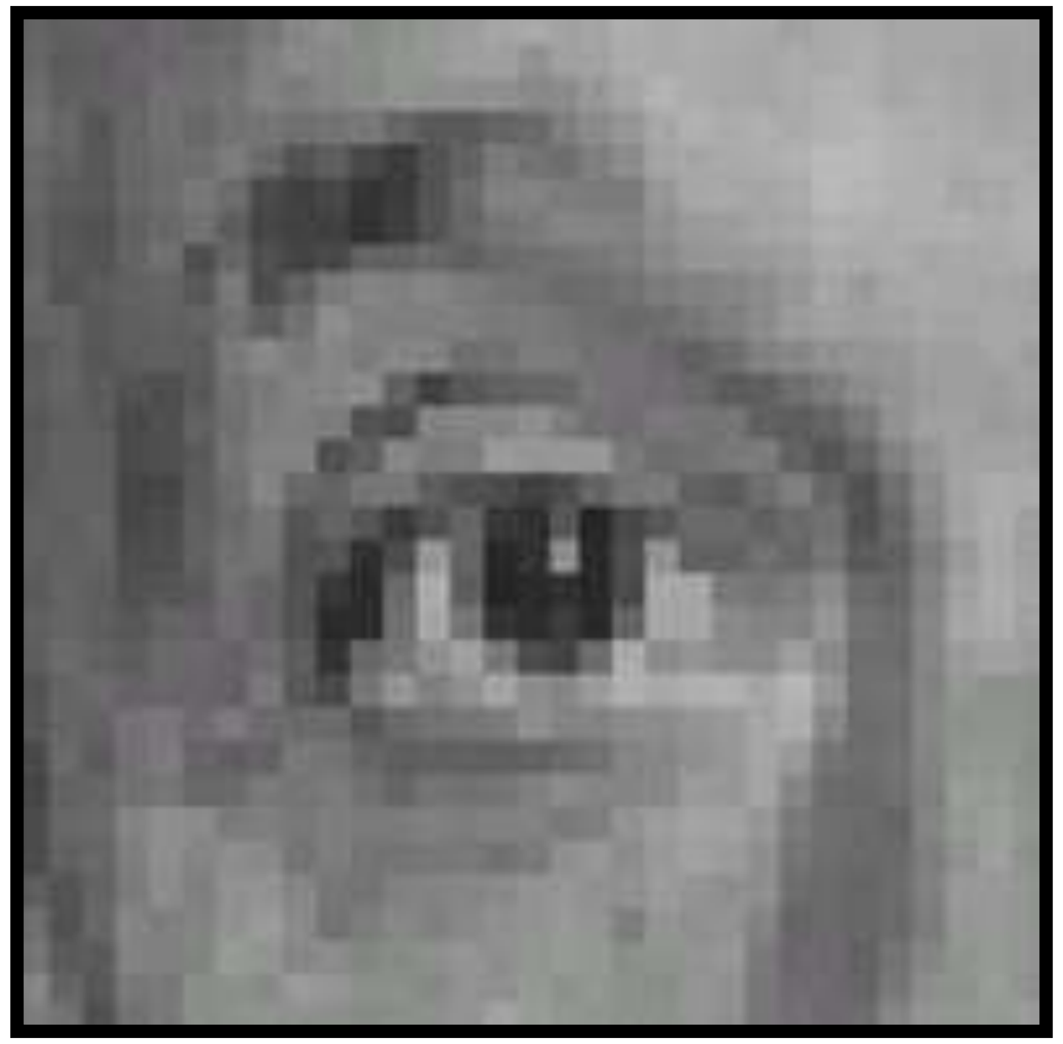
Filter

(filter sums to 1)

Result

(sharpening)

Example 4:



Original

(Scaled)
Image Itself

Blurred Version

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

Filter

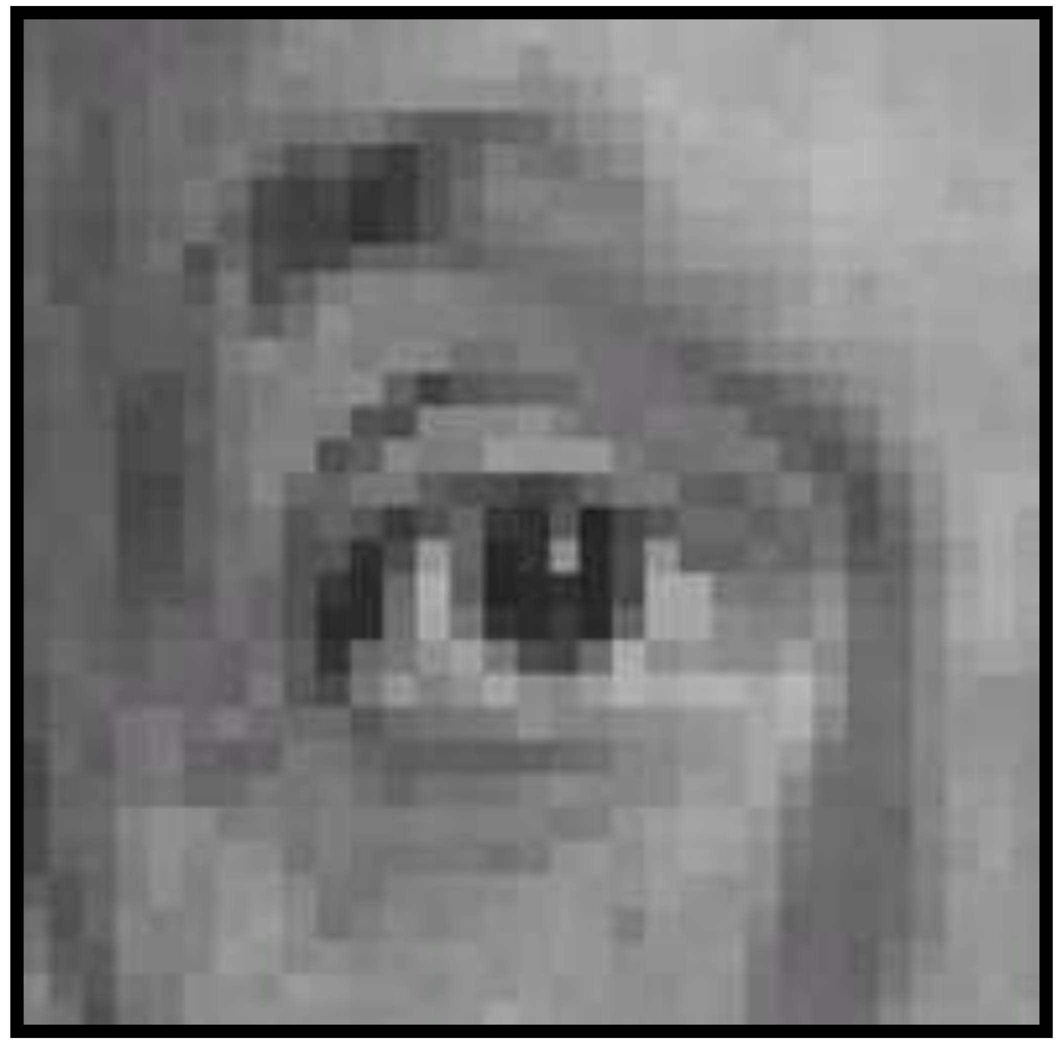
(filter sums to 1)

Result

(sharpening)

Example 4:

Why have filters sum up to 1?



Original

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

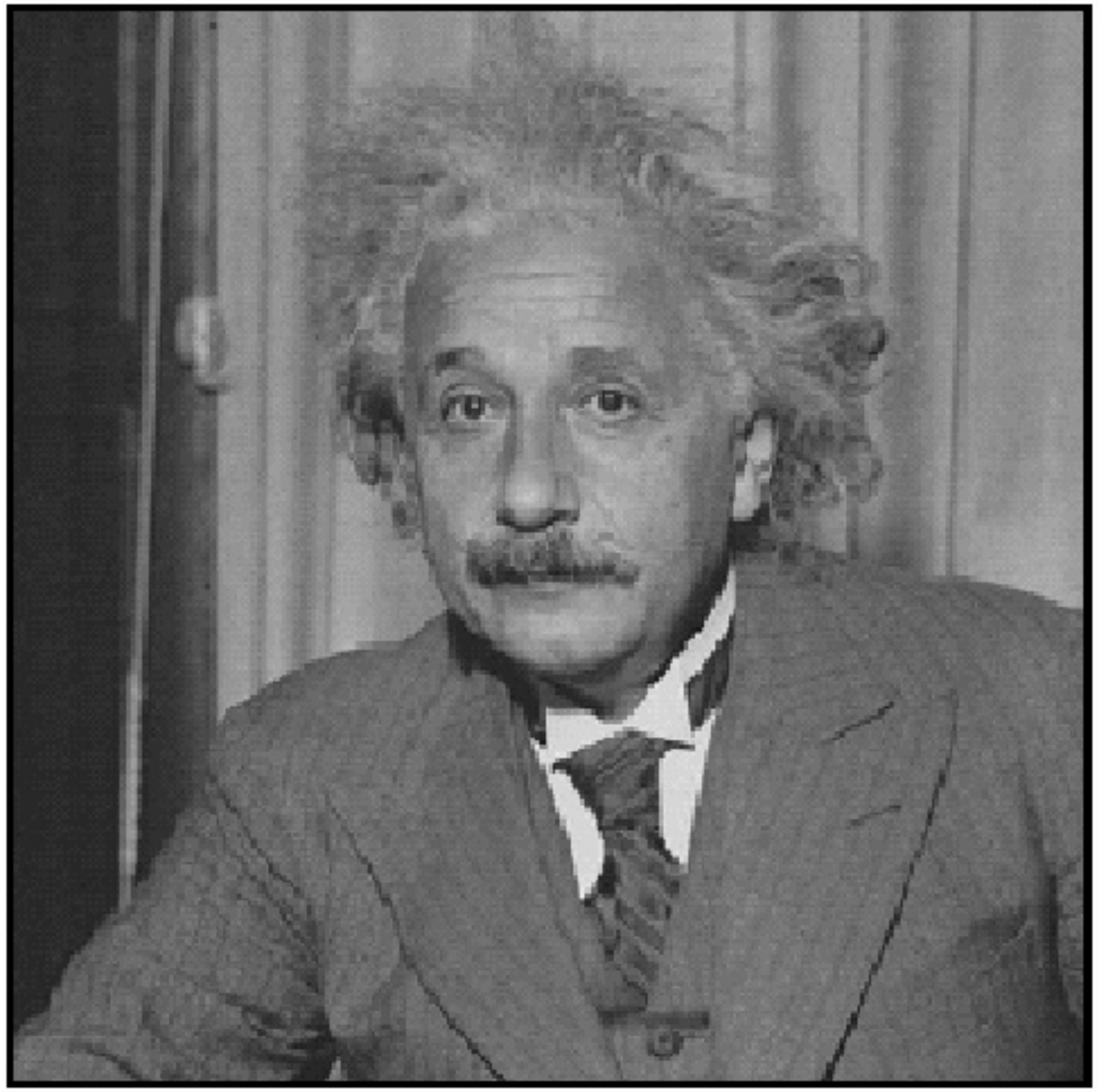
Filter

(filter sums to 1)

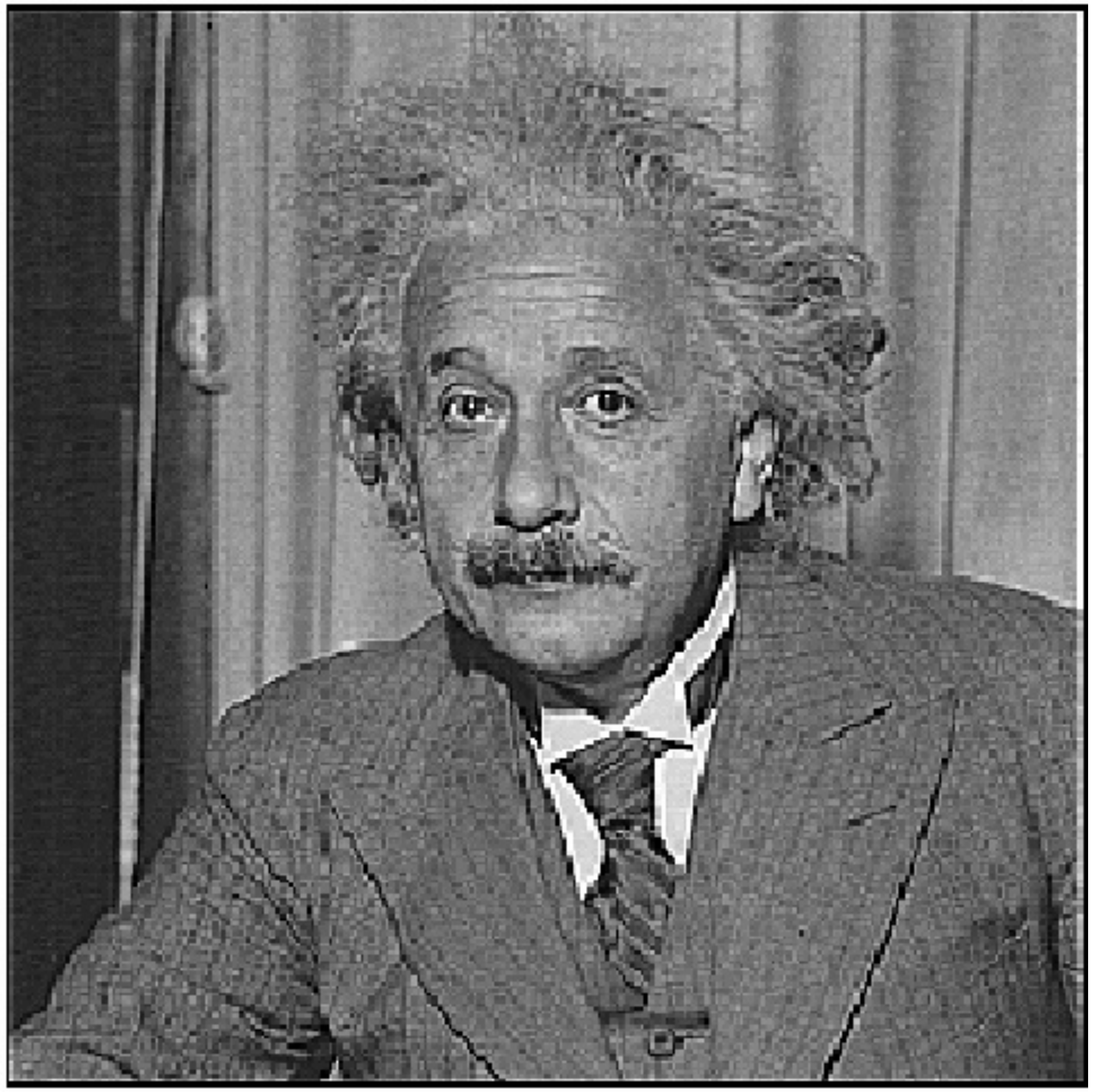
Result

(sharpening)

Example 4: Sharpening



Before



After

Example 4: Sharpening

Before

After

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Filters: Correlation vs. Convolution

Definition: **Correlation**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

Linear Filters: Correlation vs. Convolution

Definition: **Correlation**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

Definition: **Convolution**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X-i, Y-j)$$

Linear Filters: Correlation vs. Convolution

Definition: **Correlation**

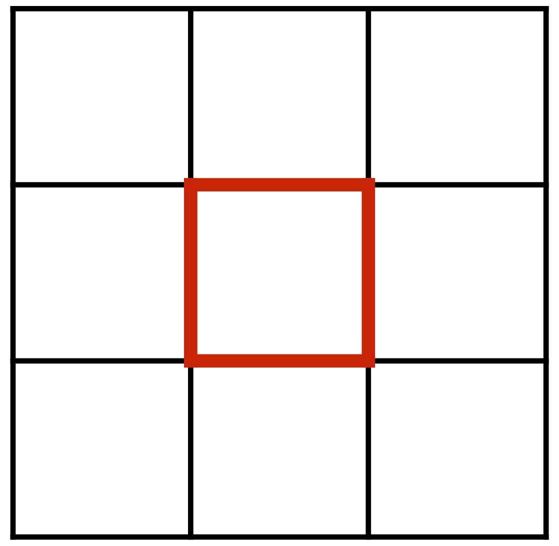
$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

a	b	c
d	e	f
g	h	i

Filter

1	2	3
4	5	6
7	8	9

Image



Output

Linear Filters: Correlation vs. Convolution

Definition: **Correlation**

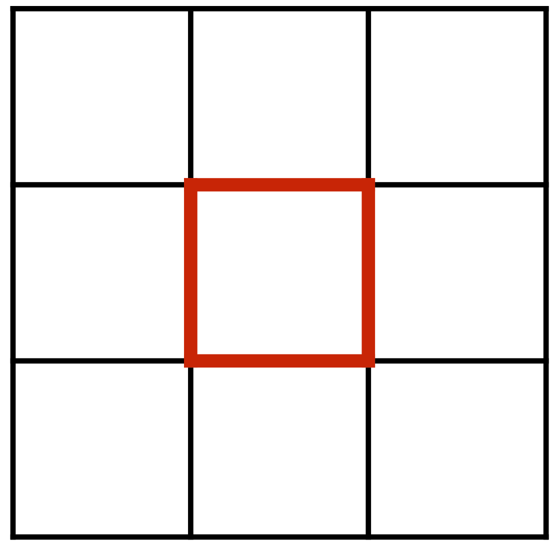
$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

a	b	c
d	e	f
g	h	i

Filter

1	2	3
4	5	6
7	8	9

Image



Output

= 1a + 2b + 3c
+ 4d + 5e + 6f
+ 7g + 8h + 9i

Linear Filters: Correlation vs. Convolution

Definition: **Correlation**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

Definition: **Convolution**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X-i, Y-j)$$

a	b	c
d	e	f
g	h	i

Filter

1	2	3
4	5	6
7	8	9

Image

Output

Linear Filters: Correlation vs. Convolution

Definition: **Correlation**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

Definition: **Convolution**

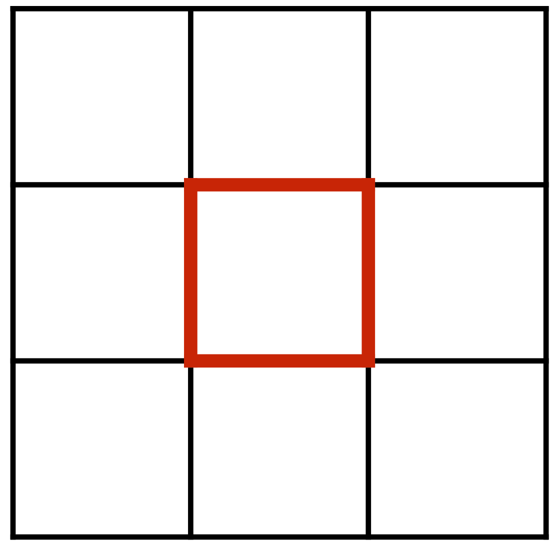
$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X-i, Y-j)$$

a	b	c
d	e	f
g	h	i

Filter

1	2	3
4	5	6
7	8	9

Image



Output

= 9a + 8b + 7c
+ 6d + 5e + 4f
+ 3g + 2h + 1i

Linear Filters: Correlation vs. Convolution

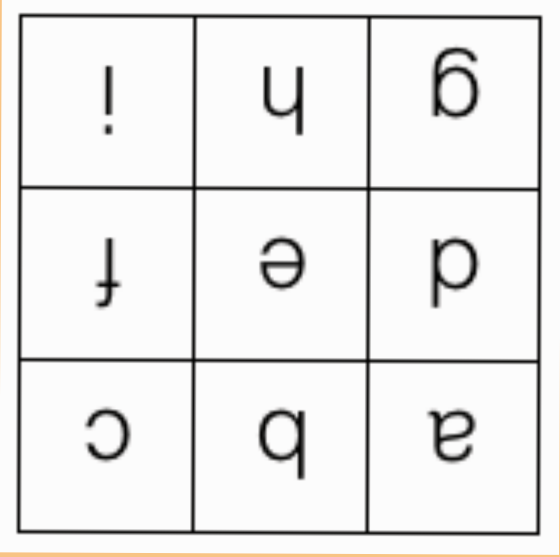
Definition: **Correlation**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

Definition: **Convolution**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X-i, Y-j)$$

Filter
(rotated by 180)

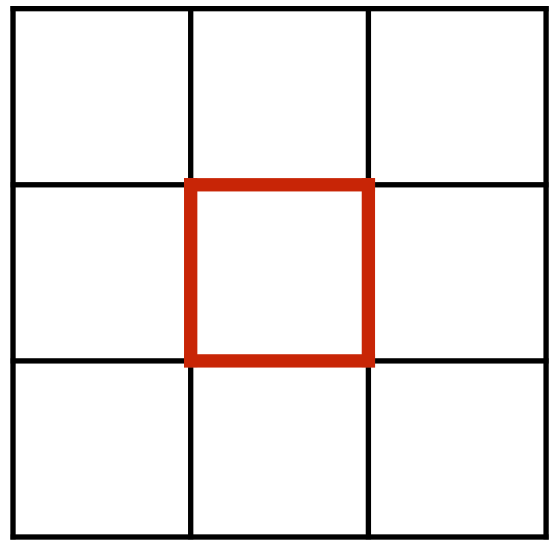


a	b	c
d	e	f
g	h	i

Filter

1	2	3
4	5	6
7	8	9

Image



Output

= 9a + 8b + 7c
+ 6d + 5e + 4f
+ 3g + 2h + 1i

Linear Filters: Correlation vs. Convolution

Definition: **Correlation**

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

Definition: **Convolution**

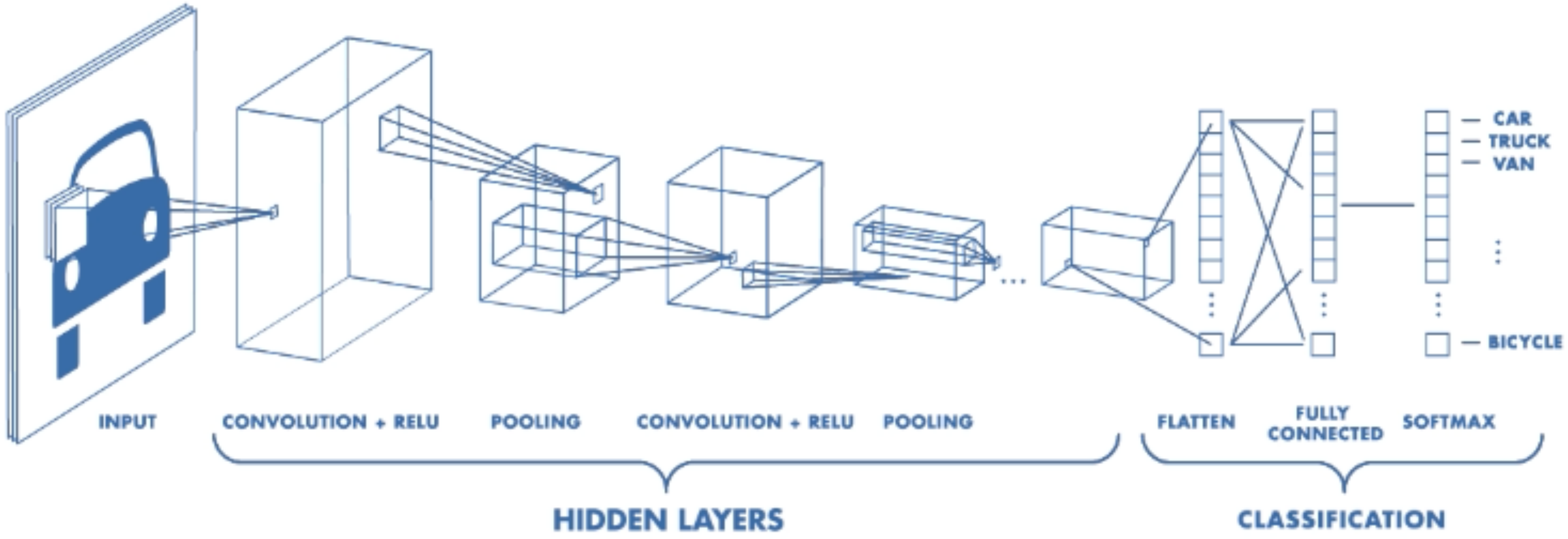
$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X-i, Y-j)$$

$$= \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(-i,-j) I(X+i, Y+j)$$

Note: if $F(X,Y) = F(-X,-Y)$ then correlation = convolution.

Preview: Why convolutions are important?

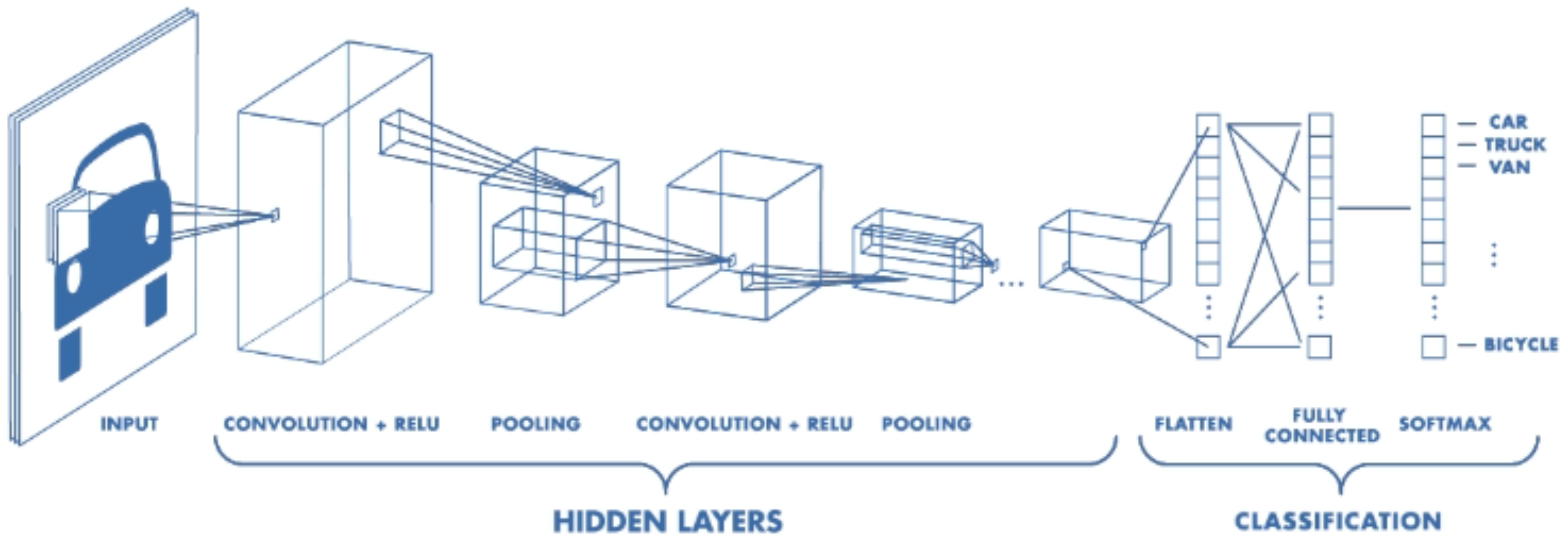
Who has heard of **Convolutional Neural Networks** (CNNs)?



Preview: Why convolutions are important?

Who has heard of **Convolutional Neural Networks** (CNNs)?

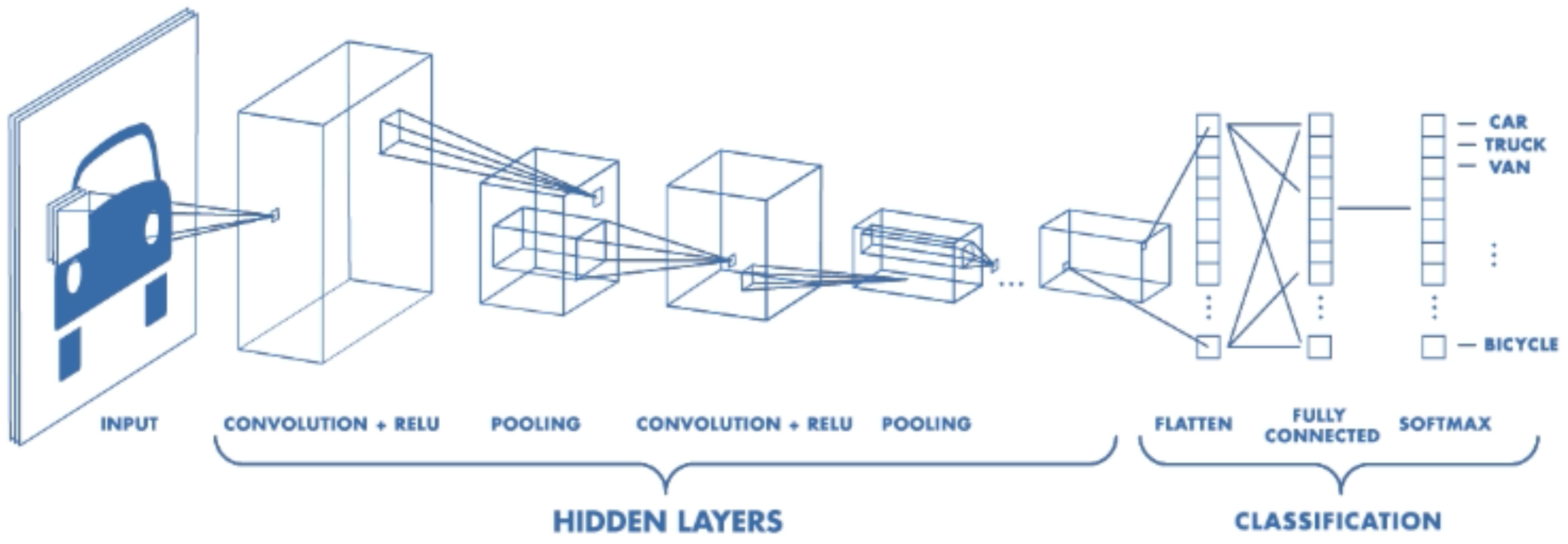
What about **Deep Learning**?



Preview: Why convolutions are important?

Who has heard of **Convolutional Neural Networks** (CNNs)?

What about **Deep Learning**?



Basic operations in CNNs are convolutions (with learned linear filters) followed by non-linear functions.

Note: This results in non-linear filters.

Linear Filters: **Properties**

Let $\otimes$ denote convolution. Let $I(X,Y)$ be a digital image

Superposition: Let F_1 and F_2 be digital filters

$$(F_1 + F_2) \otimes I(X,Y) = F_1 \otimes I(X,Y) + F_2 \otimes I(X,Y)$$

Linear Filters: **Properties**

Let $\otimes$ denote convolution. Let $I(X, Y)$ be a digital image

Superposition: Let F_1 and F_2 be digital filters

$$(F_1 + F_2) \otimes I(X, Y) = F_1 \otimes I(X, Y) + F_2 \otimes I(X, Y)$$

Scaling: Let F be digital filter and let k be a scalar

$$(kF) \otimes I(X, Y) = F \otimes (kI(X, Y)) = k(F \otimes I(X, Y))$$

Linear Filters: **Properties**

Let $\otimes$ denote convolution. Let $I(X,Y)$ be a digital image

Superposition: Let F_1 and F_2 be digital filters

$$(F_1 + F_2) \otimes I(X,Y) = F_1 \otimes I(X,Y) + F_2 \otimes I(X,Y)$$

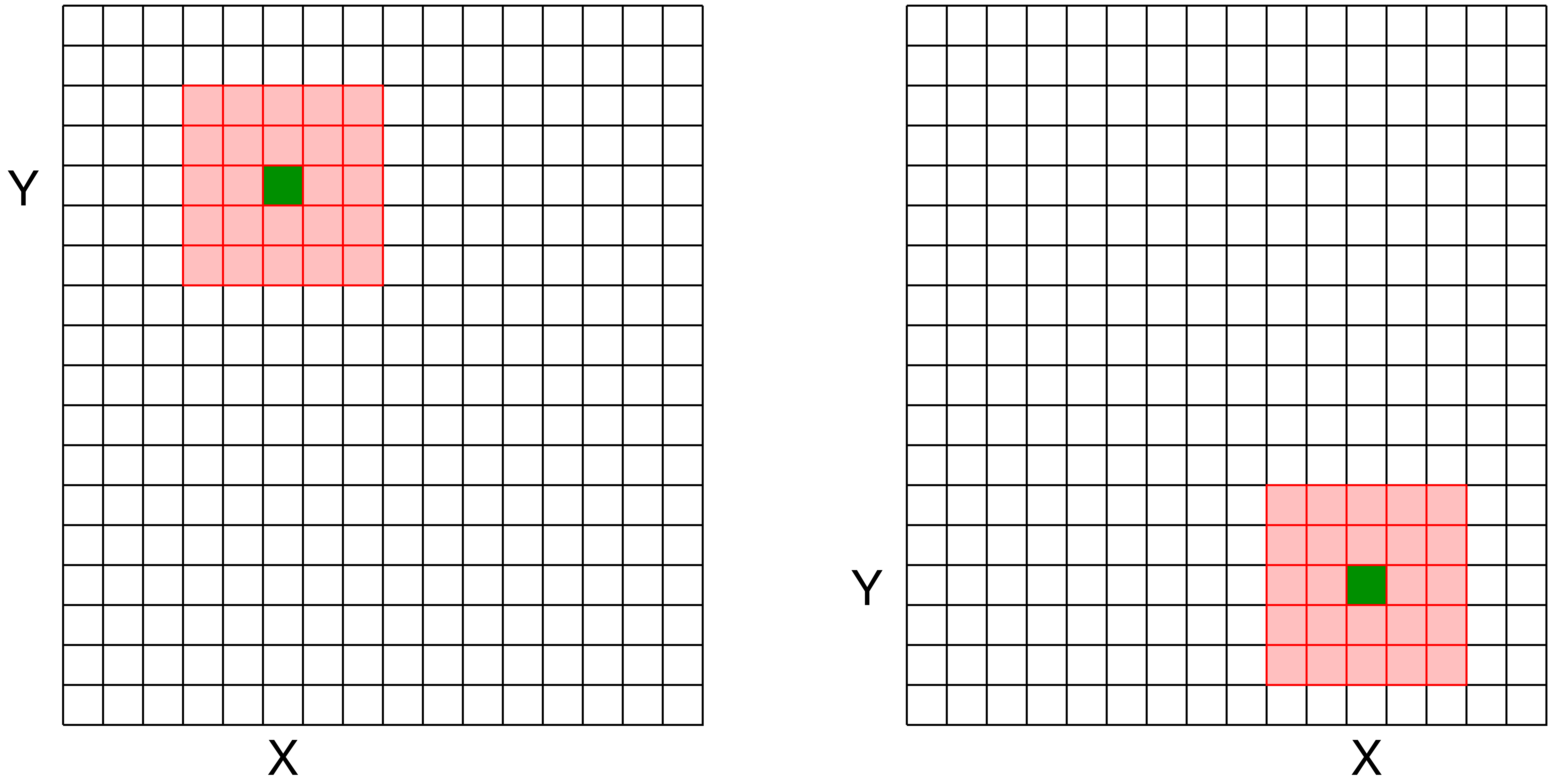
Scaling: Let F be digital filter and let k be a scalar

$$(kF) \otimes I(X,Y) = F \otimes (kI(X,Y)) = k(F \otimes I(X,Y))$$

Shift Invariance: Output is local (i.e., no dependence on absolute position)

Linear Filters: Shift Invariance

Output does **not** depend on absolute position



Linear Filters: **Properties**

Let $\otimes$ denote convolution. Let $I(X, Y)$ be a digital image

Superposition: Let F_1 and F_2 be digital filters

$$(F_1 + F_2) \otimes I(X, Y) = F_1 \otimes I(X, Y) + F_2 \otimes I(X, Y)$$

Scaling: Let F be digital filter and let k be a scalar

$$(kF) \otimes I(X, Y) = F \otimes (kI(X, Y)) = k(F \otimes I(X, Y))$$

Shift Invariance: Output is local (i.e., no dependence on absolute position)

An operation is **linear** if it satisfies both **superposition** and **scaling**

Linear Systems: Characterization Theorem

Any linear, shift invariant operation can be expressed as convolution

Example 5: Smoothing with a Box Filter

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

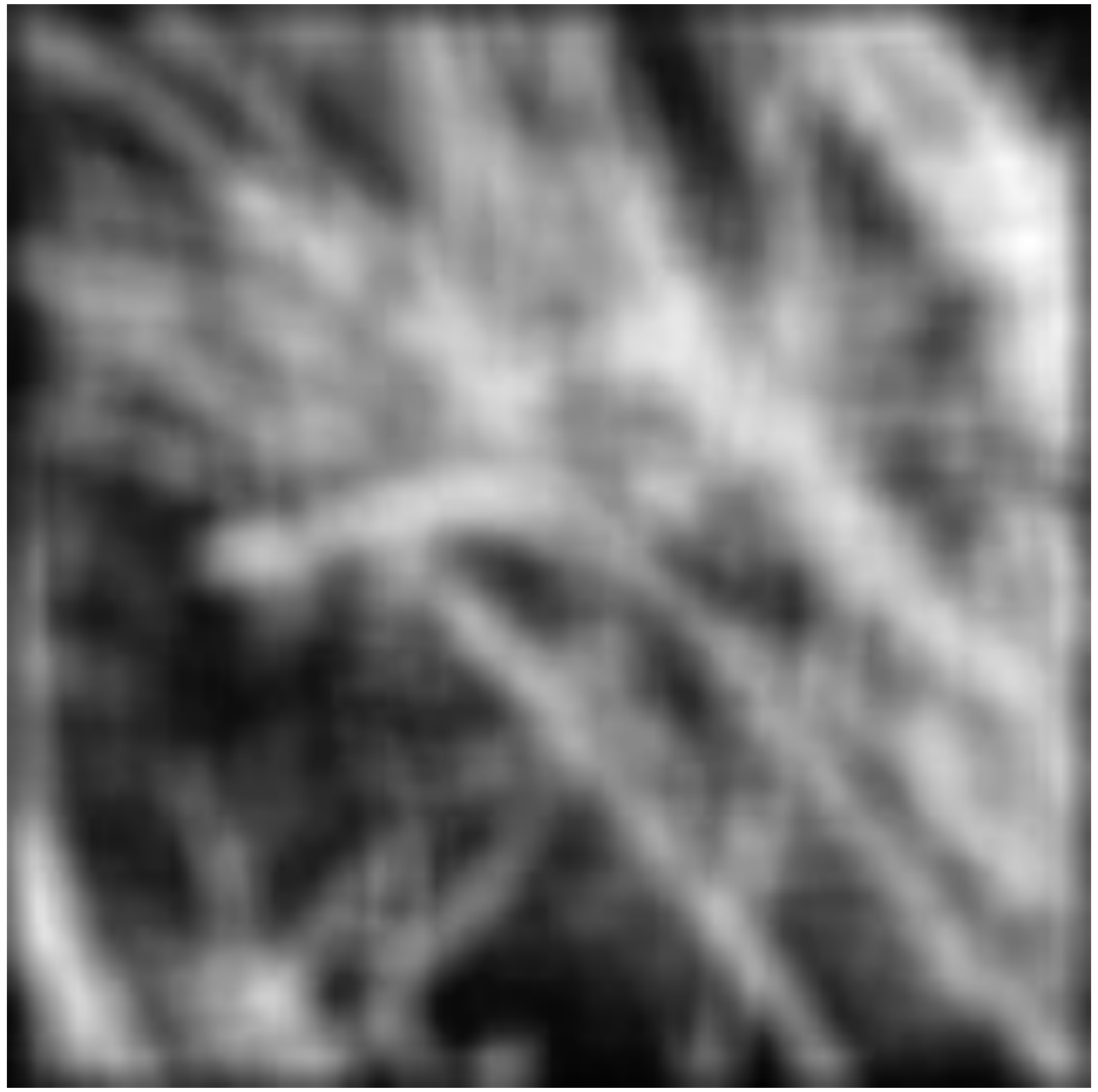
Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Filter has equal positive values that some up to 1

Replaces each pixel with the average of itself and its local neighborhood

— Box filter is also referred to as **average filter** or **mean filter**

Example 5: Smoothing with a Box Filter

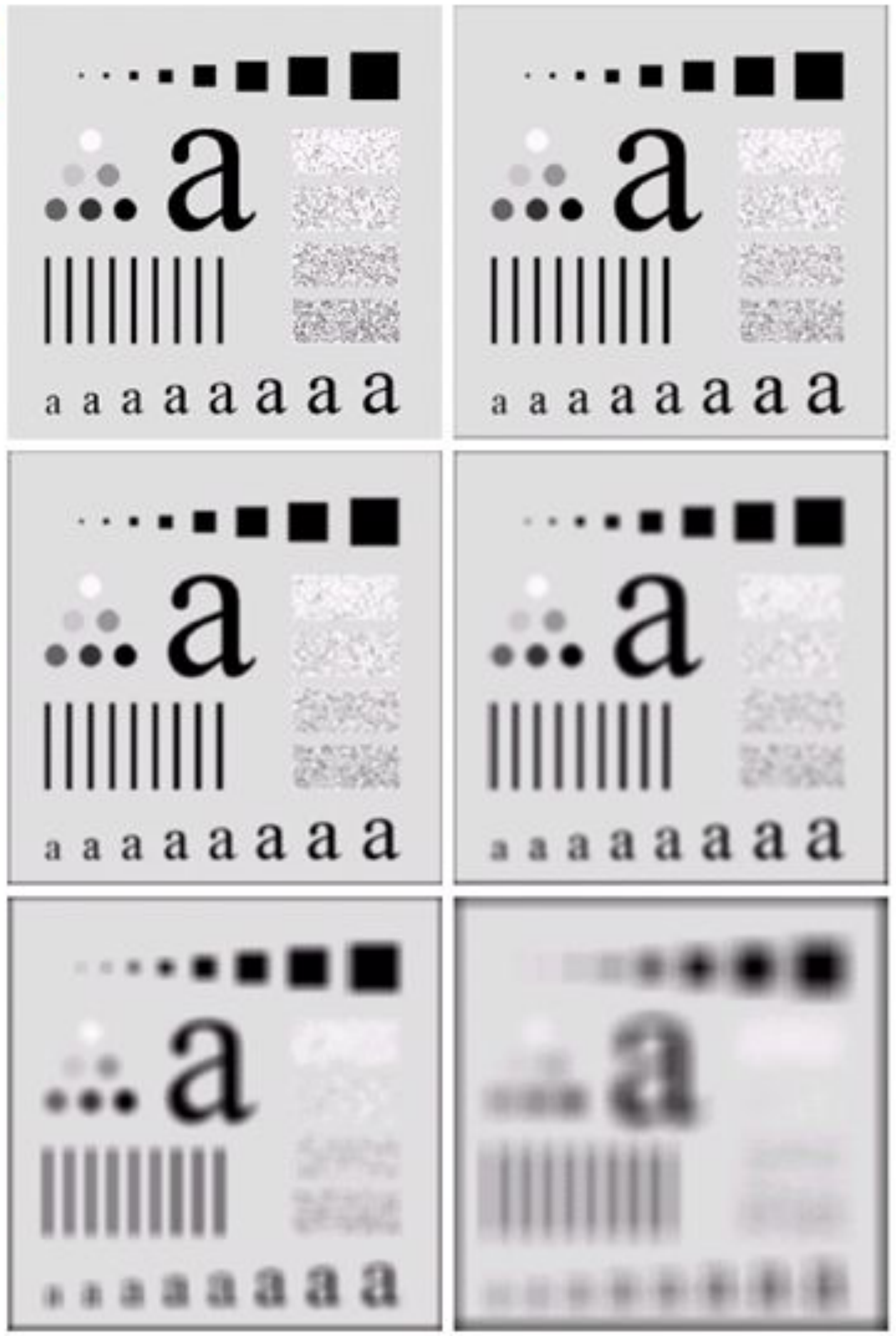


Forsyth & Ponce (2nd ed.) Figure 4.1 (left and middle)

Example 5: Smoothing with a Box Filter

What happens if we increase the width (size) of the box filter?

Example 5: Smoothing with a Box Filter



Original | 3x3

5x5 | 9x9

15x15 | 35x35

Gonzales & Woods (3rd ed.) Figure 3.3

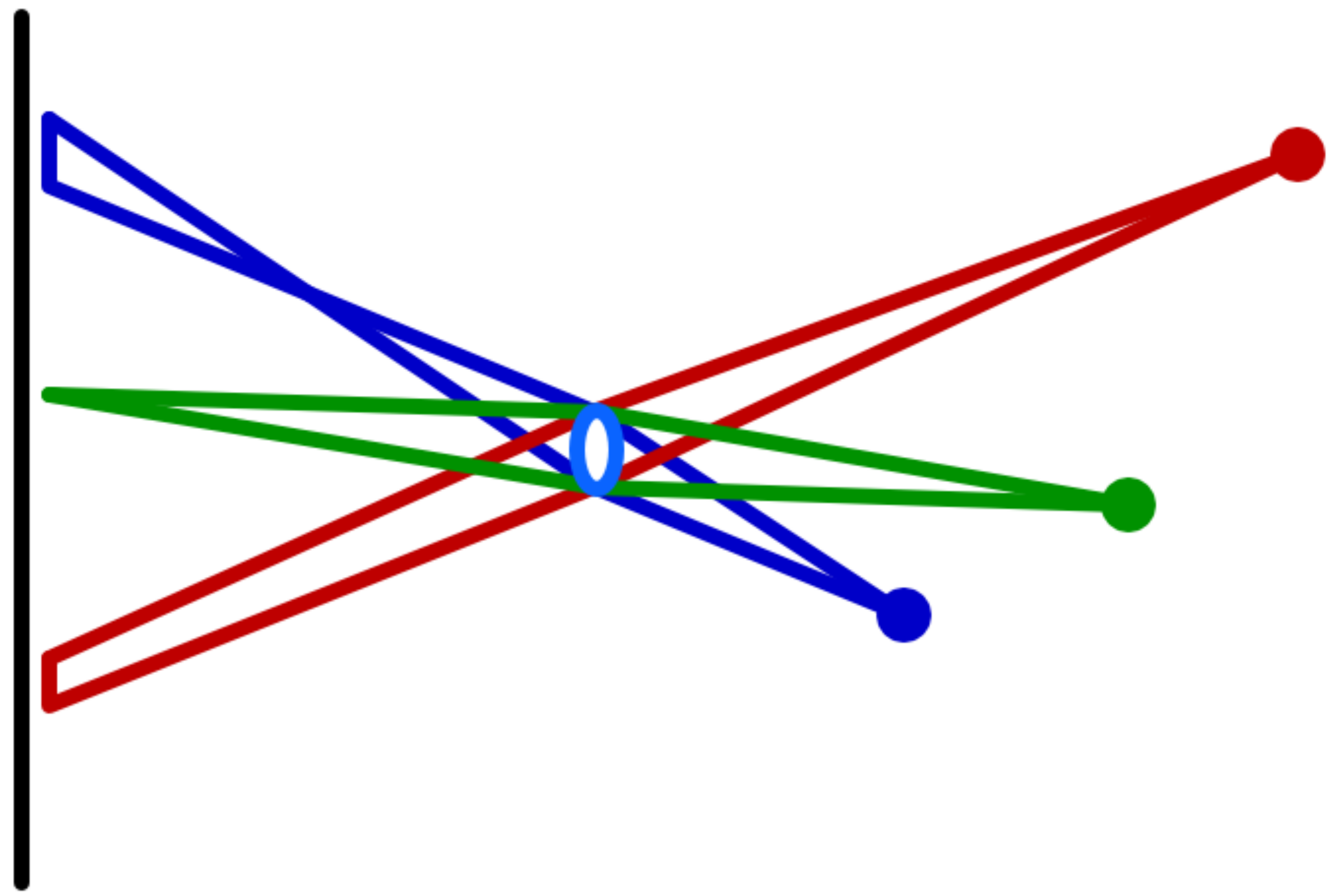
Smoothing

Smoothing with a box **doesn't model lens defocus** well

— Smoothing with a box filter depends on direction

— Image in which the center point is 1 and every other point is 0

Lecture 2: Re-cap



* image credit: https://catlikecoding.com/unity/tutorials/advanced-rendering/depth-of-field/circle-of-confusion/lens-camera.png

Smoothing

Smoothing with a box **doesn't model lens defocus** well

— Smoothing with a box filter depends on direction

— Image in which the center point is 1 and every other point is 0

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Filter

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Image

Smoothing

Smoothing with a box **doesn’t model lens defocus** well

— Smoothing with a box filter depends on direction

— Image in which the center point is 1 and every other point is 0

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Filter

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Image

0	0	0	0	0
0	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	0
0	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	0
0	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	0
0	0	0	0	0

Result

Smoothing

Smoothing with a box **doesn’t model lens defocus** well

— Smoothing with a box filter depends on direction

— Image in which the center point is 1 and every other point is 0

Smoothing with a (circular) **pillbox** is a better model for defocus (in geometric optics)

The **Gaussian** is a good general smoothing model

— for phenomena (that are the sum of other small effects)

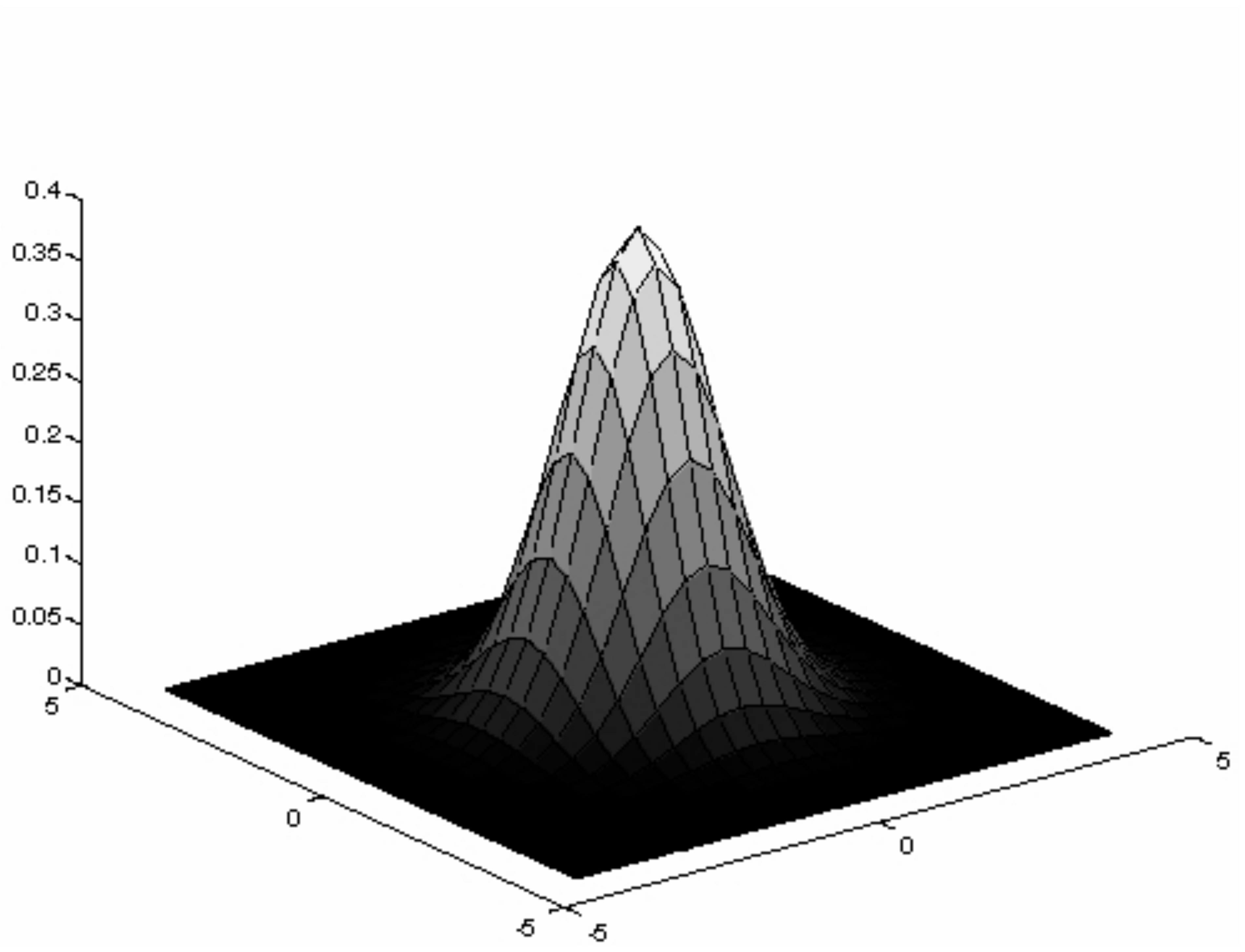
— whenever the Central Limit Theorem applies

Example 6: Smoothing with a Gaussian

Idea: Weight contributions of pixels by spatial proximity (nearness)

2D **Gaussian** (continuous case):

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



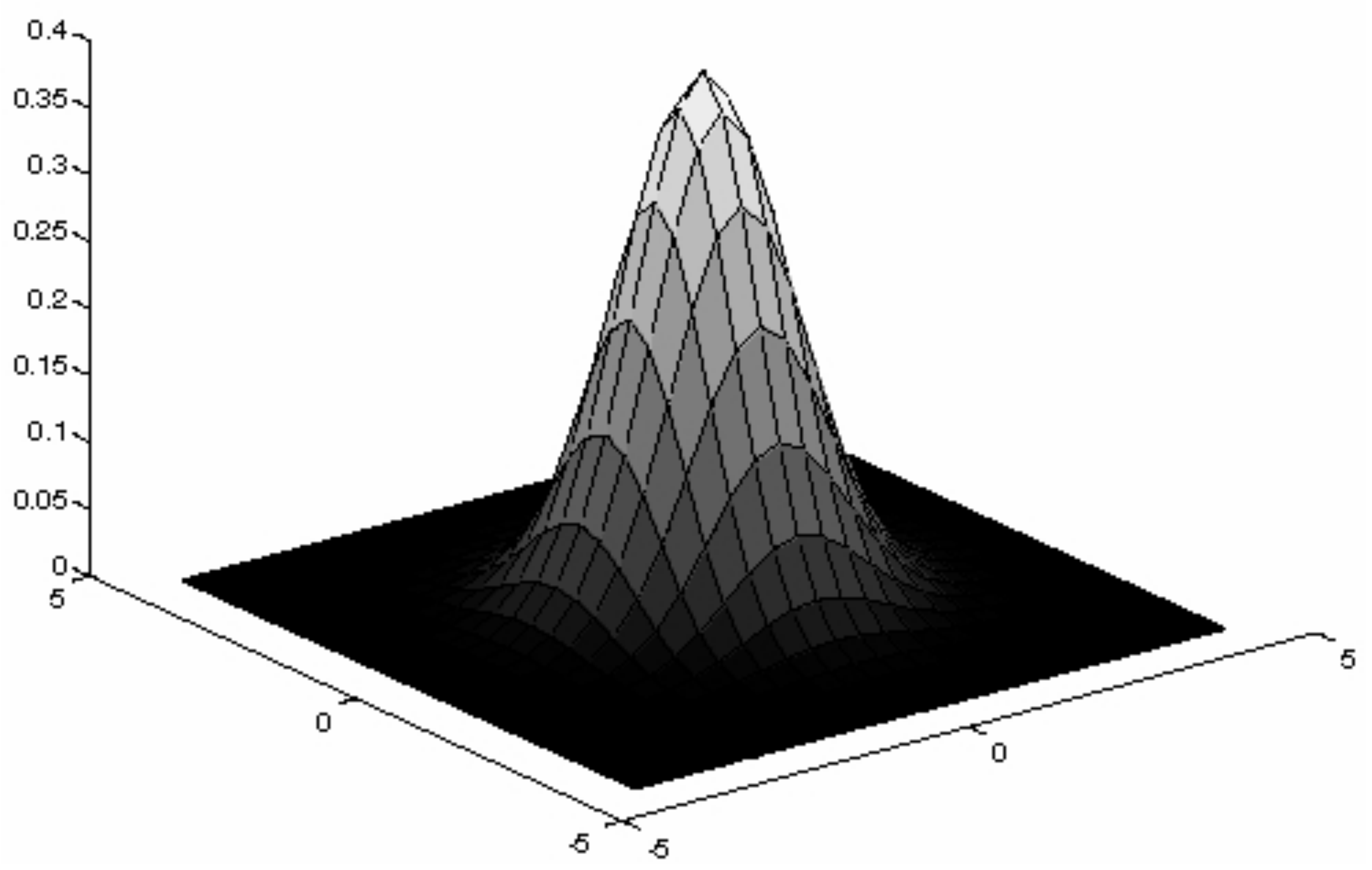
Forsyth & Ponce (2nd ed.)
Figure 4.2

Example 6: Smoothing with a Gaussian

Idea: Weight contributions of pixels by spatial proximity (nearness)

2D **Gaussian** (continuous case):

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



Forsyth & Ponce (2nd ed.)
Figure 4.2

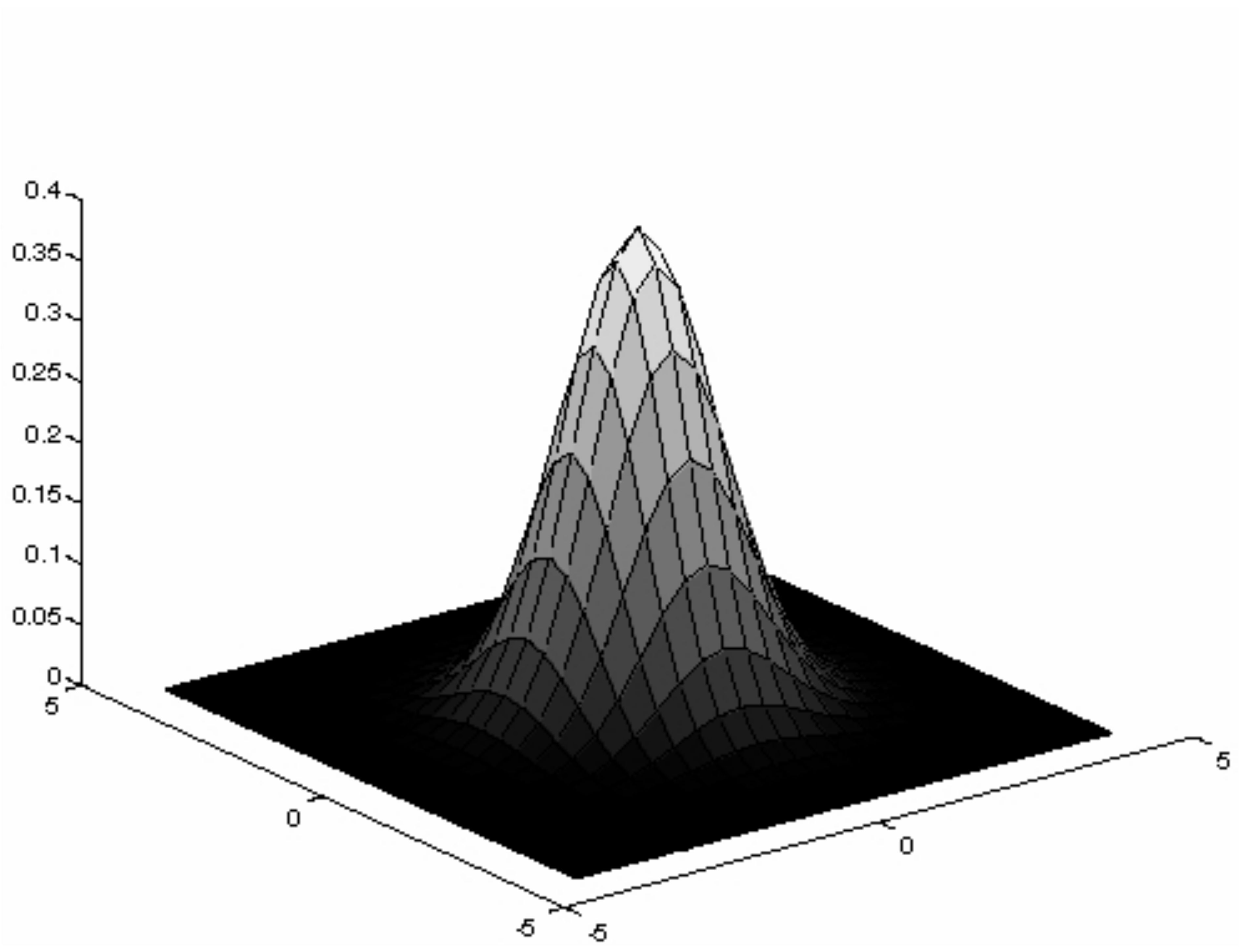
Example 6: Smoothing with a Gaussian

Idea: Weight contributions of pixels by spatial proximity (nearness)

2D **Gaussian** (continuous case):

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

Standard Deviation



Forsyth & Ponce (2nd ed.)
Figure 4.2

Example 6: Smoothing with a Gaussian

Quantized an truncated **3x3 Gaussian** filter:

$G_\sigma(-1,1)$	$G_\sigma(0,1)$	$G_\sigma(1,1)$
$G_\sigma(-1,0)$	$G_\sigma(0,0)$	$G_\sigma(1,0)$
$G_\sigma(-1,-1)$	$G_\sigma(0,-1)$	$G_\sigma(1,-1)$

Example 6: Smoothing with a Gaussian

Quantized an truncated **3x3 Gaussian** filter:

$G_\sigma(-1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$
$G_\sigma(-1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(0,0) = \frac{1}{2\pi\sigma^2}$	$G_\sigma(1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$
$G_\sigma(-1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$

Example 6: Smoothing with a Gaussian

Quantized an truncated **3x3 Gaussian** filter:

$G_\sigma(-1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$
$G_\sigma(-1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(0,0) = \frac{1}{2\pi\sigma^2}$	$G_\sigma(1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$
$G_\sigma(-1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$

With $\sigma = 1$:

0.059	0.097	0.059
0.097	0.159	0.097
0.059	0.097	0.059

Example 6: Smoothing with a Gaussian

Quantized an truncated **3x3 Gaussian** filter:

$G_\sigma(-1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$
$G_\sigma(-1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(0,0) = \frac{1}{2\pi\sigma^2}$	$G_\sigma(1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$
$G_\sigma(-1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$

With $\sigma = 1$:

0.059	0.097	0.059
0.097	0.159	0.097
0.059	0.097	0.059

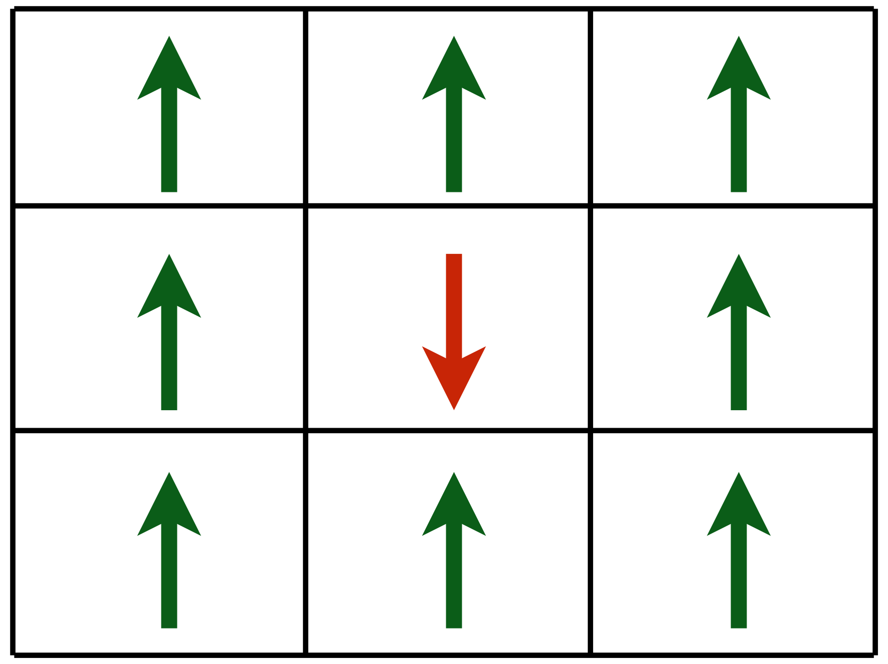
What happens if σ is larger?

Example 6: Smoothing with a Gaussian

Quantized an truncated **3x3 Gaussian** filter:

$G_\sigma(-1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$
$G_\sigma(-1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(0,0) = \frac{1}{2\pi\sigma^2}$	$G_\sigma(1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$
$G_\sigma(-1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$

With $\sigma = 1$:



What happens if σ is larger?

— **More** blur

Example 6: Smoothing with a Gaussian

Quantized an truncated **3x3 Gaussian** filter:

$G_\sigma(-1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$
$G_\sigma(-1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(0,0) = \frac{1}{2\pi\sigma^2}$	$G_\sigma(1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$
$G_\sigma(-1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$

With $\sigma = 1$:

0.059	0.097	0.059
0.097	0.159	0.097
0.059	0.097	0.059

What happens if σ is larger?

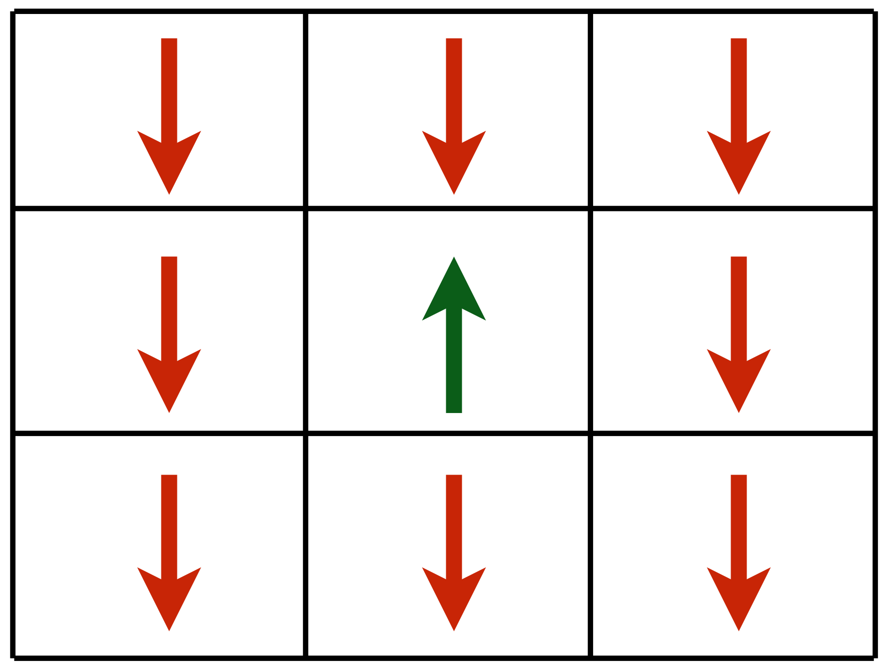
What happens if σ is smaller?

Example 6: Smoothing with a Gaussian

Quantized an truncated **3x3 Gaussian** filter:

$G_\sigma(-1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$
$G_\sigma(-1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(0,0) = \frac{1}{2\pi\sigma^2}$	$G_\sigma(1,0) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$
$G_\sigma(-1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$	$G_\sigma(0,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2\sigma^2}}$	$G_\sigma(1,-1) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{2}{2\sigma^2}}$

With $\sigma = 1$:

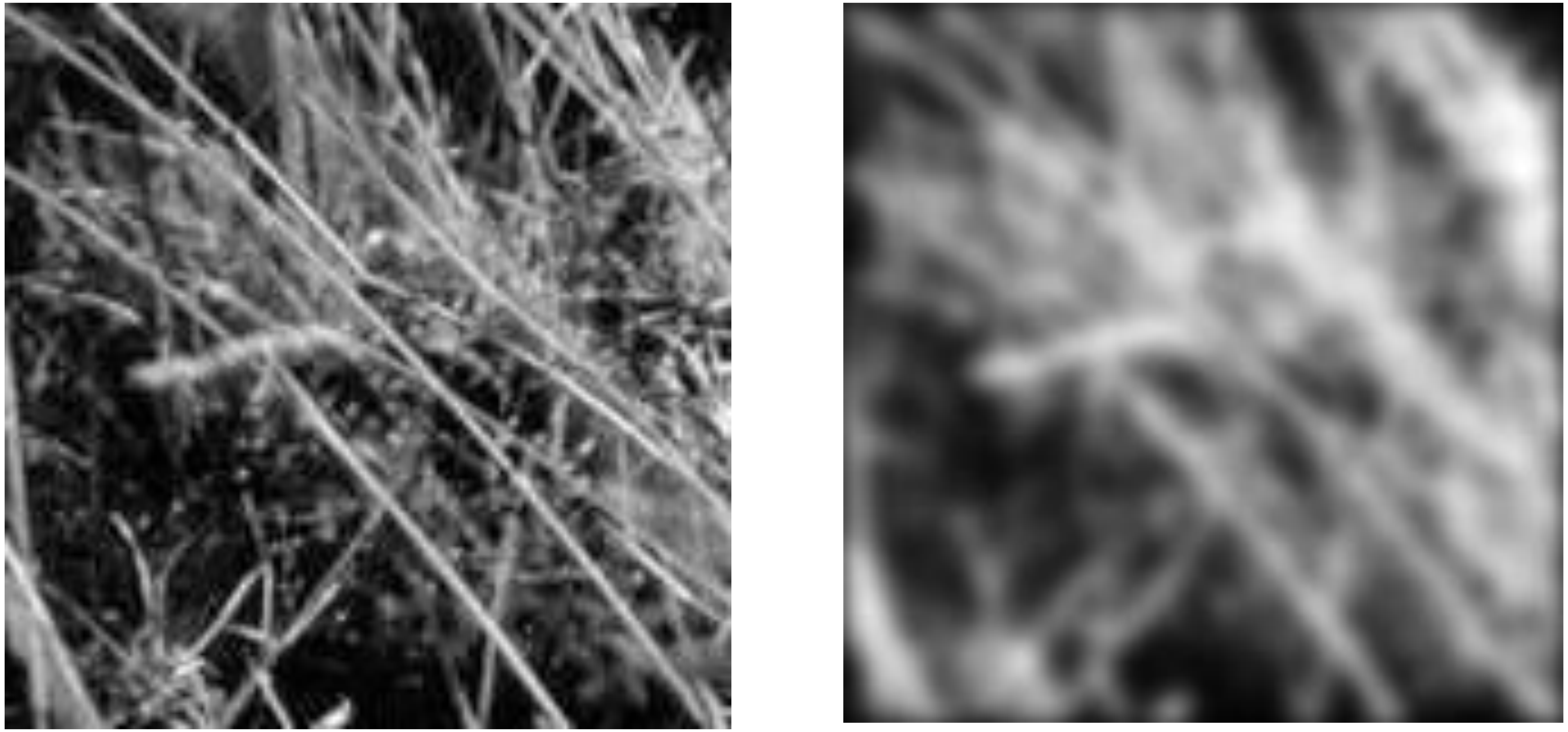


What happens if σ is larger?

What happens if σ is smaller?

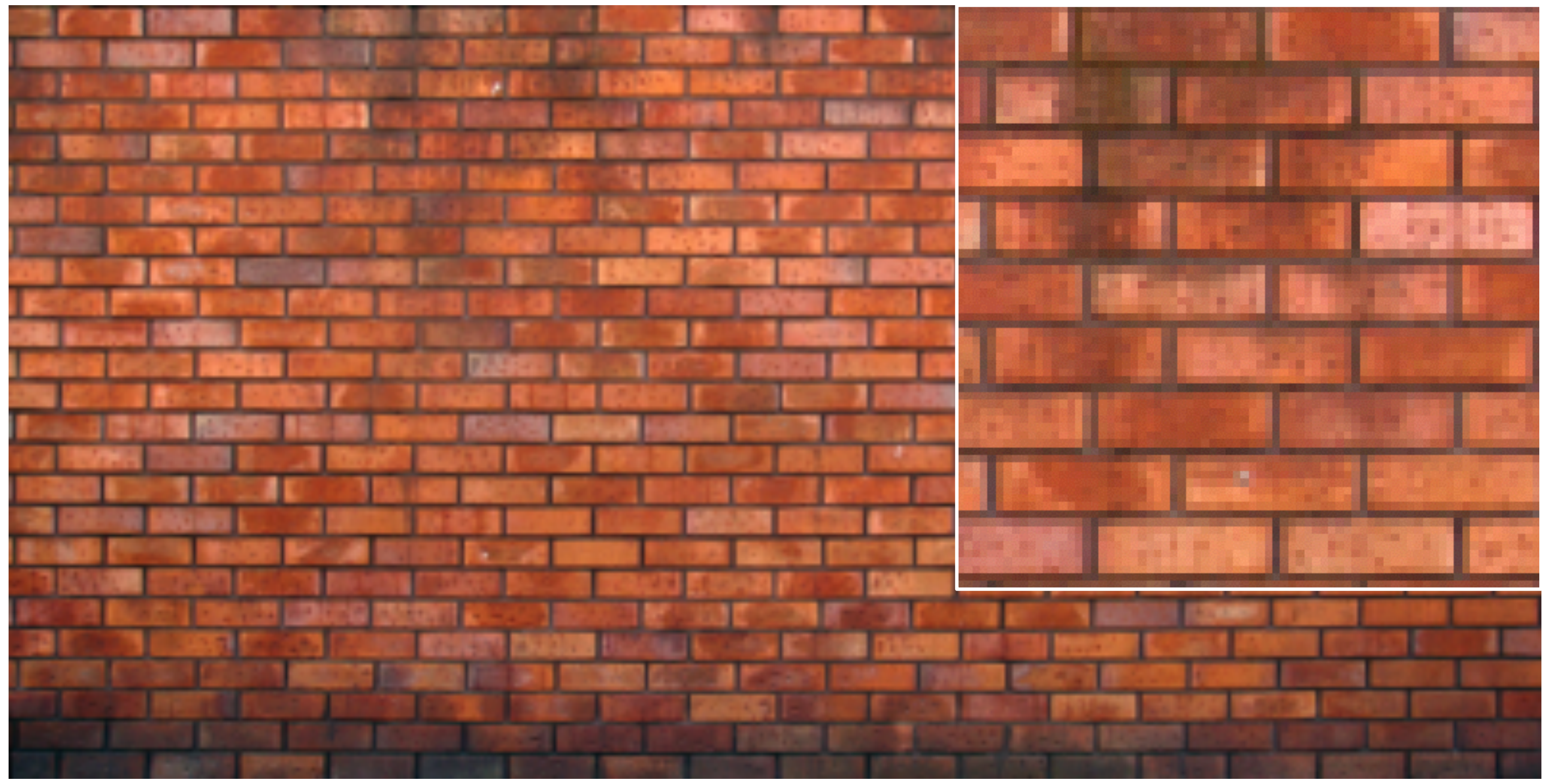
— **Less** blur

Example 6: Smoothing with a Gaussian

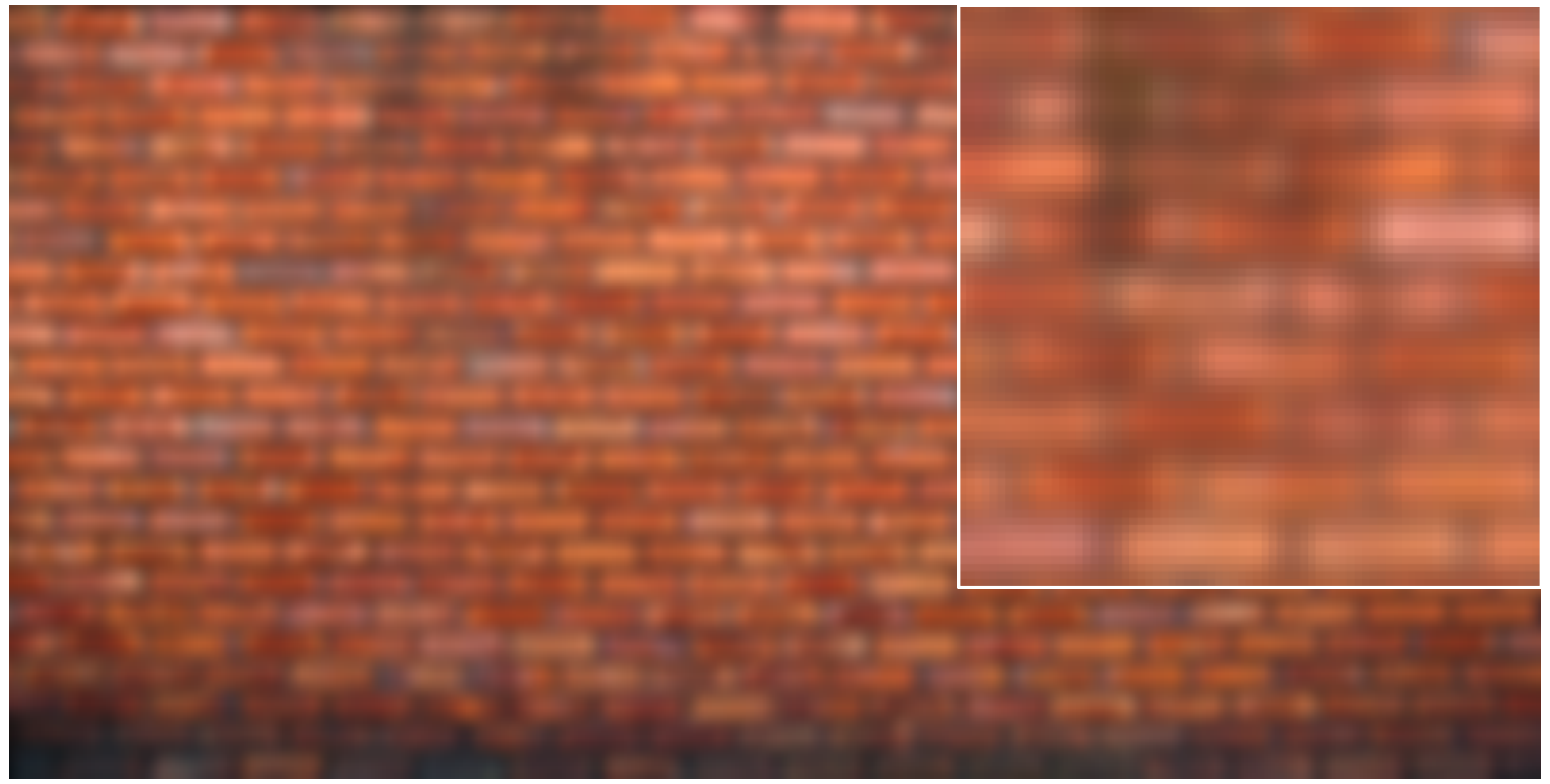


Forsyth & Ponce (2nd ed.) Figure 4.1 (left and right)

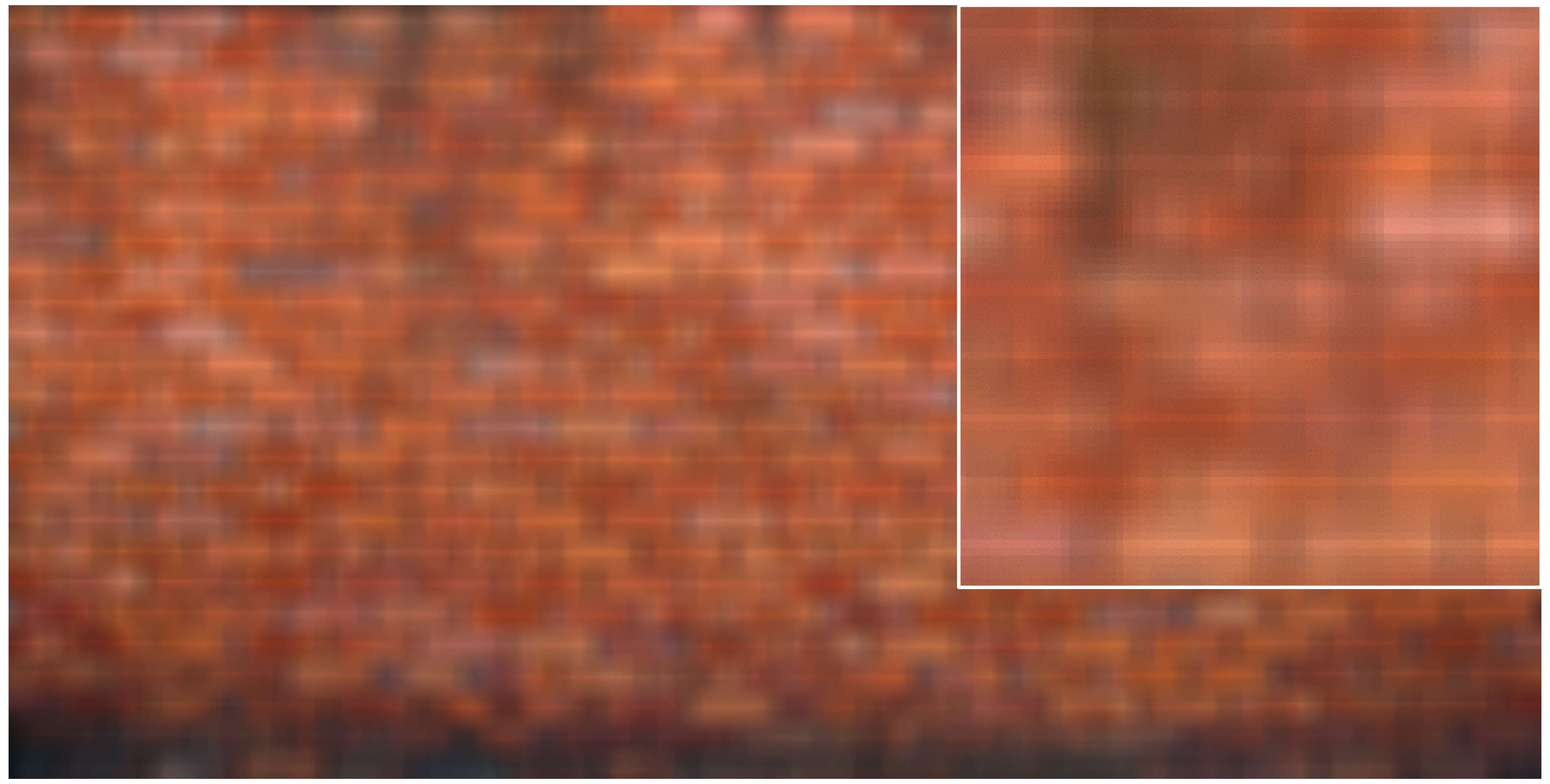
Box vs. **Gaussian** Filter



original



7x7 Gaussian



7x7 box

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Summary

— The **correlation** of $F(X,Y)$ and $I(X,Y)$ is:

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(X+i, Y+j)$$

— **Visual interpretation**: Superimpose the filter F on the image I at (X,Y), perform an element-wise multiply, and sum up the values

— **Convolution** is like **correlation** except filter "flipped"

if $F(X,Y) = F(-X,-Y)$ then correlation = convolution.

— **Characterization Theorem**: Any linear, spatially invariant operation can be expressed as a convolution