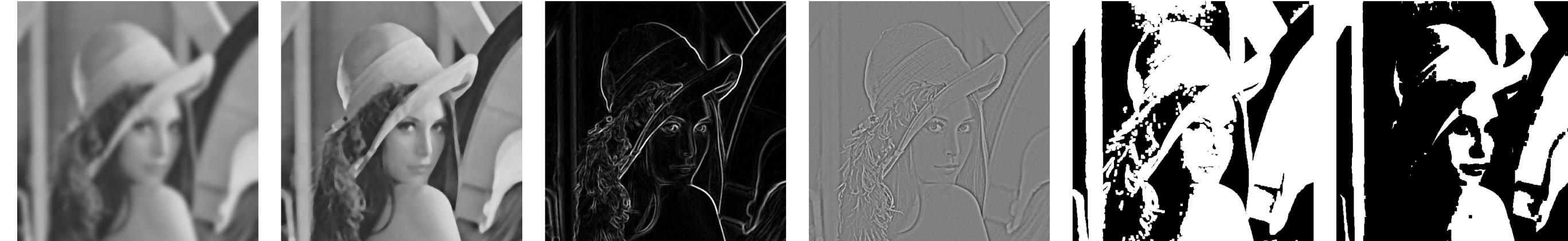




CPSC 425: Computer Vision



Lecture 4: Image Filtering (continued)

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

Menu for Today

Topics:

- Recap L3, **iClicker test**
- **Box, Gaussian, Pillbox** filters
- **Low/High Pass** Filters
- **Separability**

Readings:

- **Today's** Lecture: none
- **Next** Lecture: Forsyth & Ponce (2nd ed.) 4.4

Reminders:

- **Assignment 1:** Image Filtering and Hybrid Images is out, due **Sept 28th**

iClicker test

Please sign up for the iClicker course **CPSC 425 101 2023W1 Computer Vision**

See also the [UBC iClicker Student Guide](#), contact LT Hub if stuck

The screenshot shows a web browser window with the title bar "UBC iClicker Cloud Student Guide". The address bar contains the URL "https://lthub.ubc.ca/guides/iclicker-cloud-student-guide/". The page itself is the "iClicker Cloud Student Guide" from the Learning Technology Hub at UBC. It features the UBC logo and the text "THE UNIVERSITY OF BRITISH COLUMBIA". The main content area includes a blue header bar with "Learning Technology Hub" and a navigation menu with links like "Home", "Tool Finder", "Tool Guides", "Support", "Student Support", "Initiatives", "Governance", and "News". Below the header, there's a breadcrumb trail "Home » Tool Guides » iClicker Cloud Student Guide" and a "Contact Us" button. The main article starts with a large blue circular icon containing a white cloud-like shape. The text describes iClicker Cloud as an online student response system that allows individual responses to polls and quizzes using a computer or mobile device. It mentions that instructors receive instant results and can share them during lectures. A footer note states that iClicker Cloud has passed a UBC Privacy Impact Assessment. To the right, a section titled "What will I use it for?" lists activities like testing knowledge, supporting peer instruction, and discussing questions in small groups.

iClicker Cloud Student Guide



iClicker Cloud is an online student response system that allows you to respond individually to in-class polls and low-stakes quizzes, using your own computer or mobile device. Your instructor receives the responses instantly and may share these results and correct answers in the tool during the live lecture or afterward.

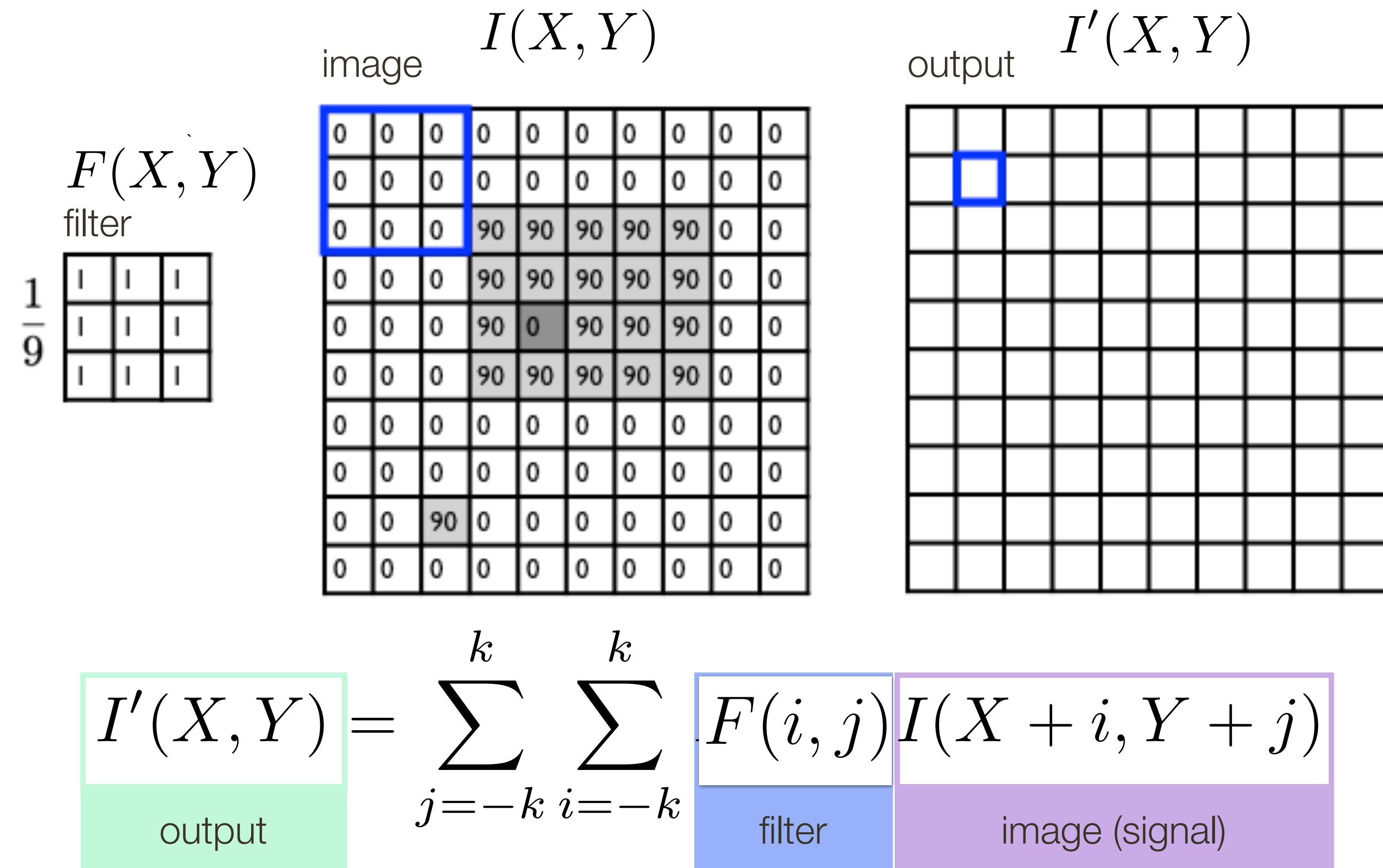
iClicker Cloud has passed a UBC [Privacy Impact Assessment](#), meaning it follows UBC and provincial privacy

What will I use it for?

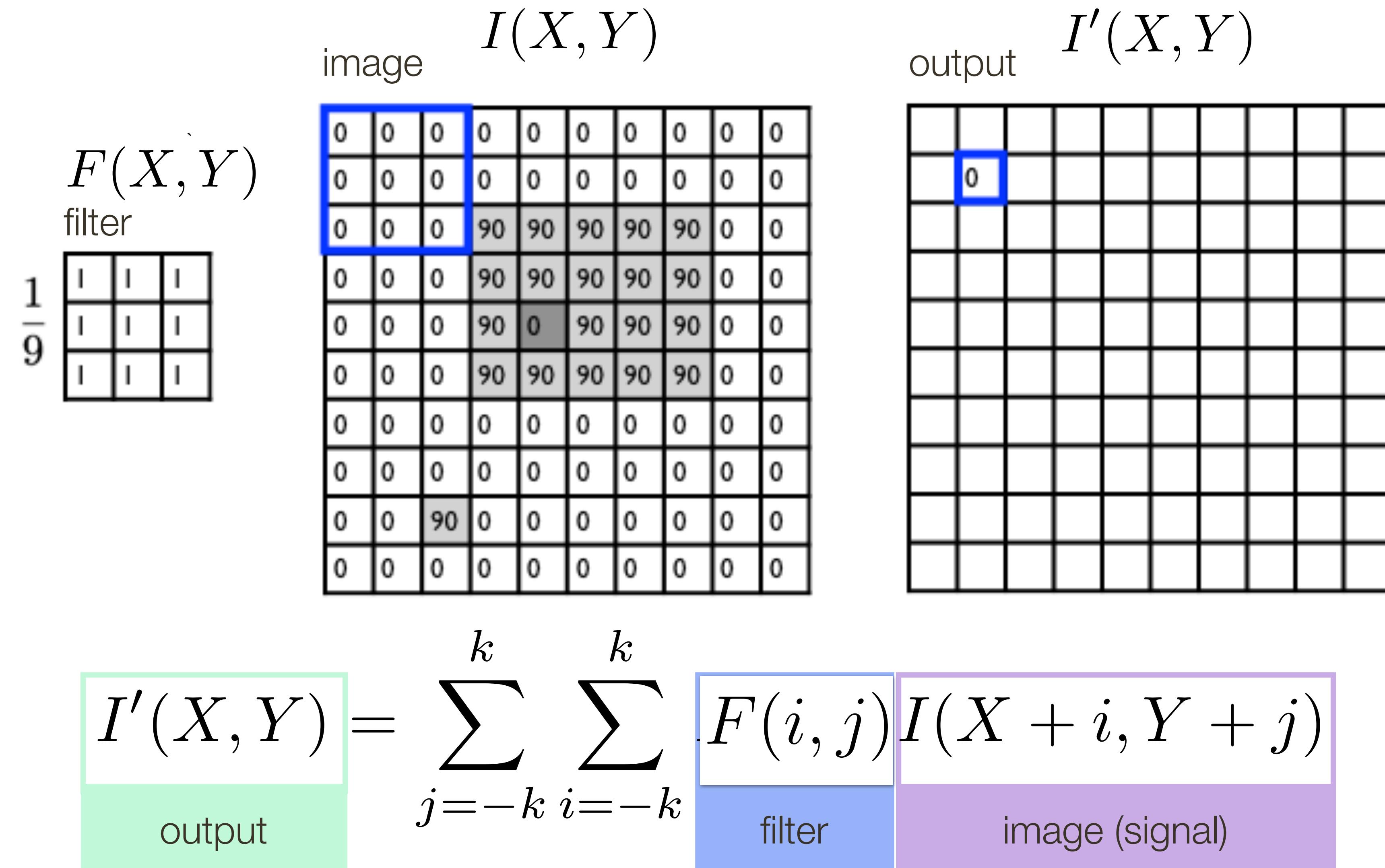
Your instructor may have you use iClicker for a variety of activities:

- Test your knowledge or opinions at different points in the class for marks
- Support peer instruction, wherein you answer a question, discuss in small

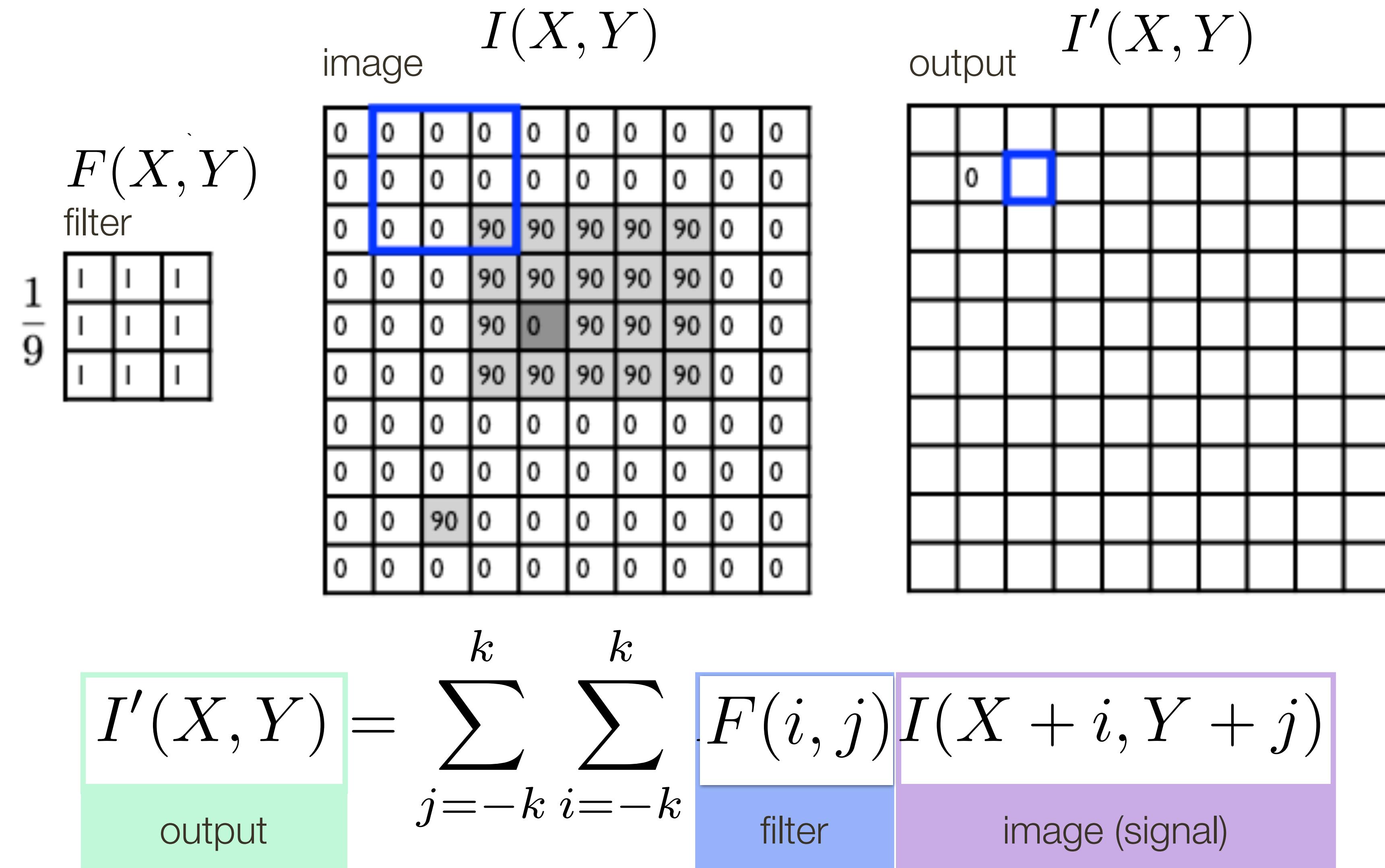
Linear Filter Example



Linear Filter Example



Linear Filter Example

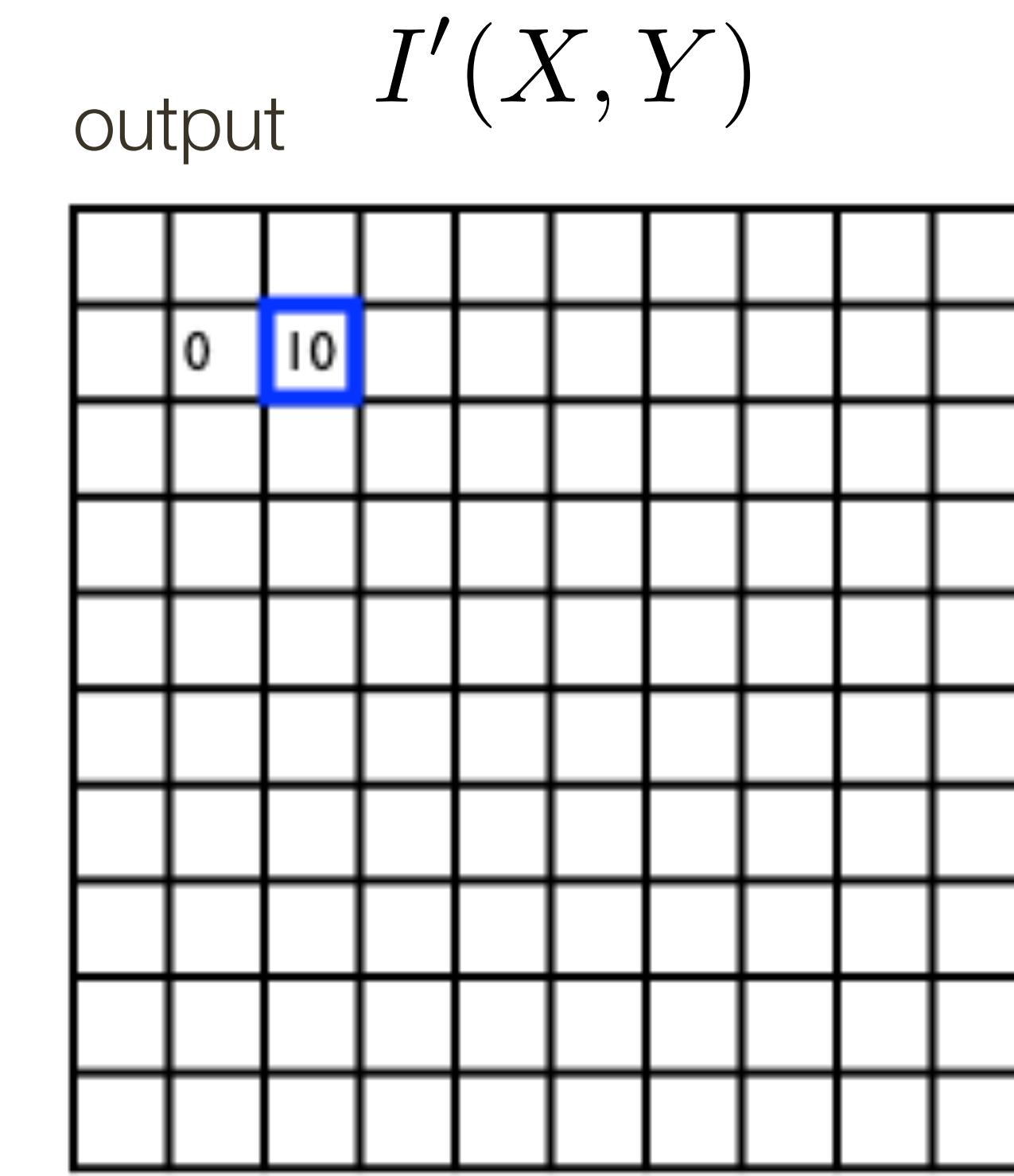
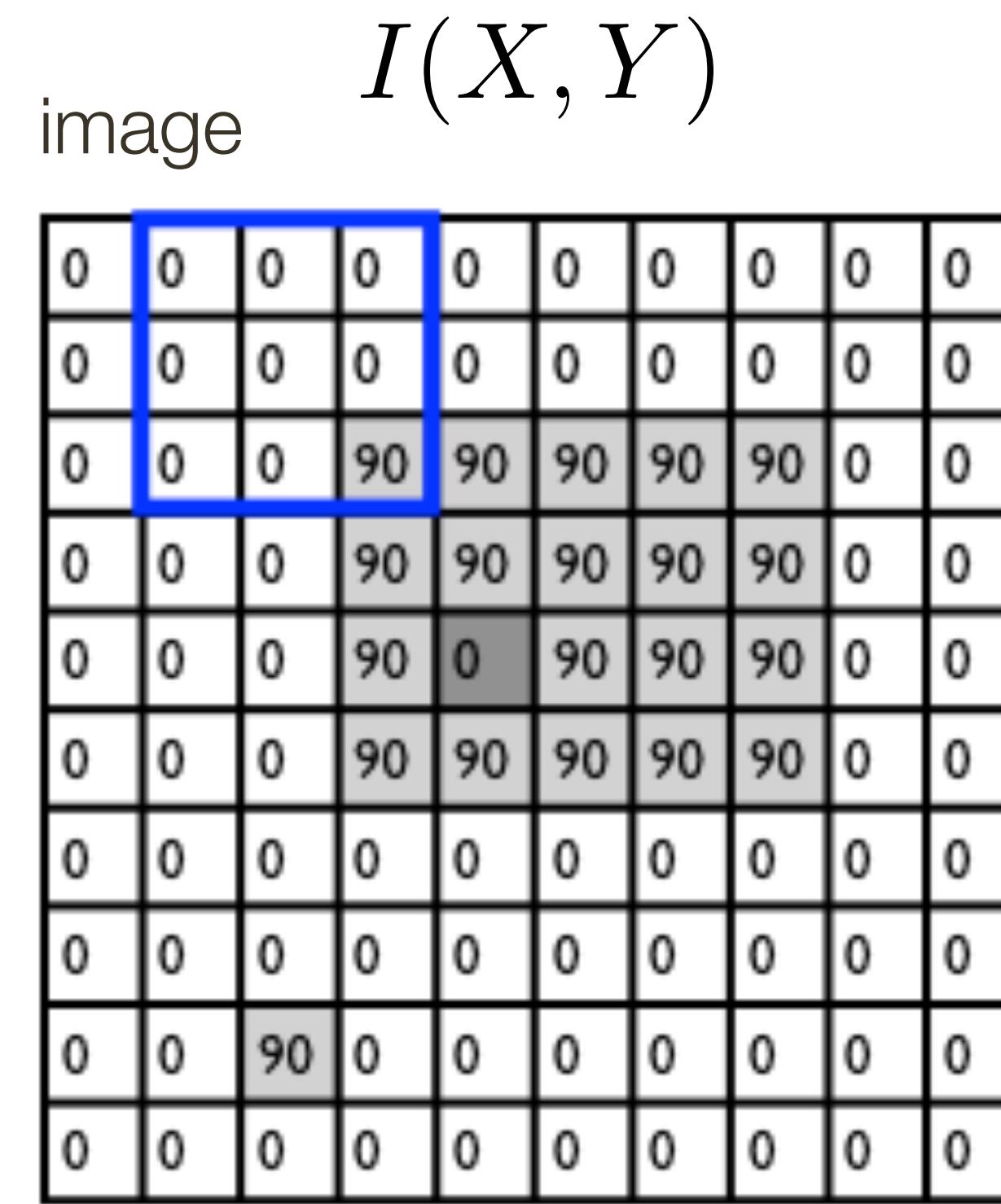


Linear Filter Example

$$F(X, Y)$$

filter

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



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

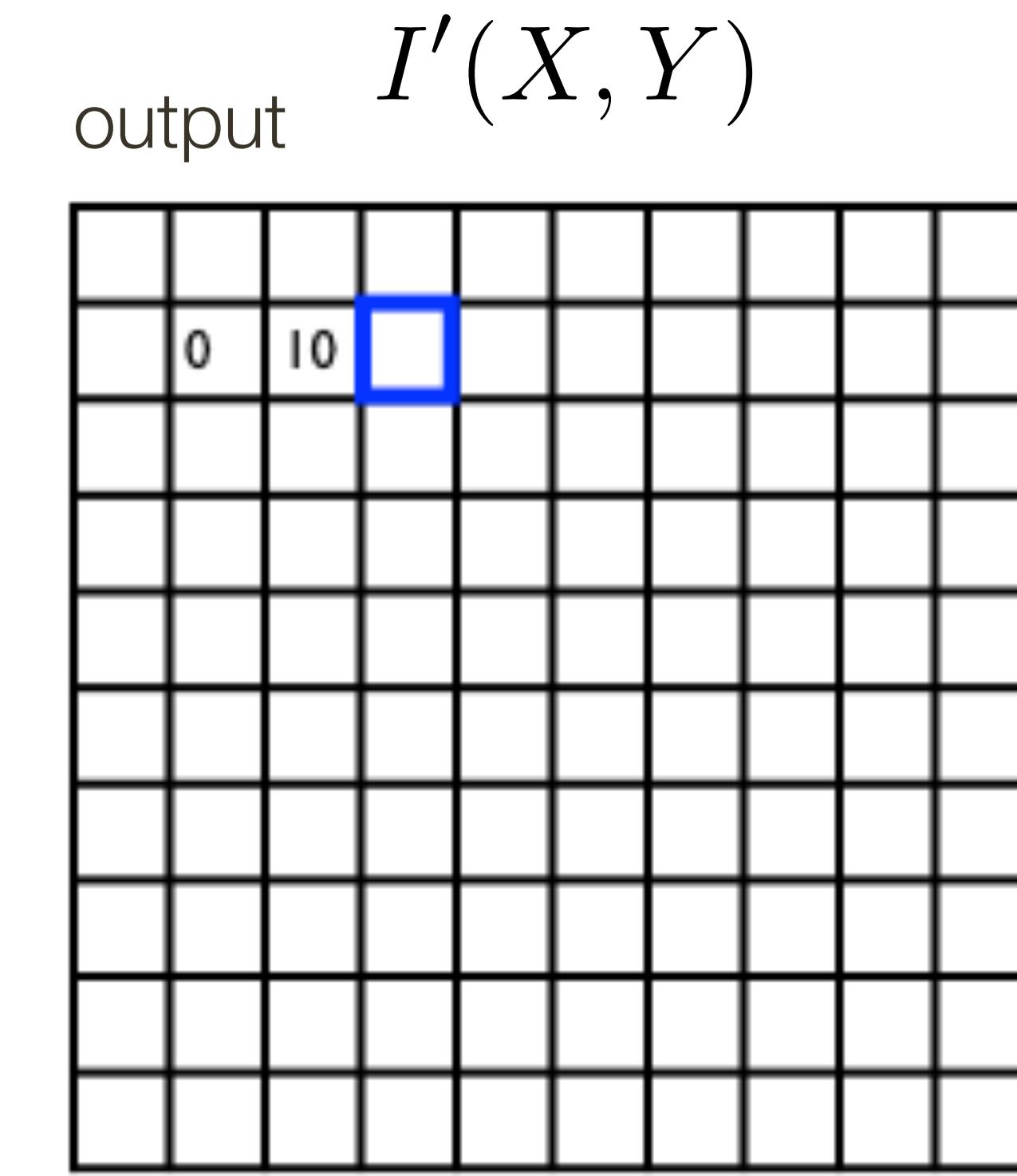
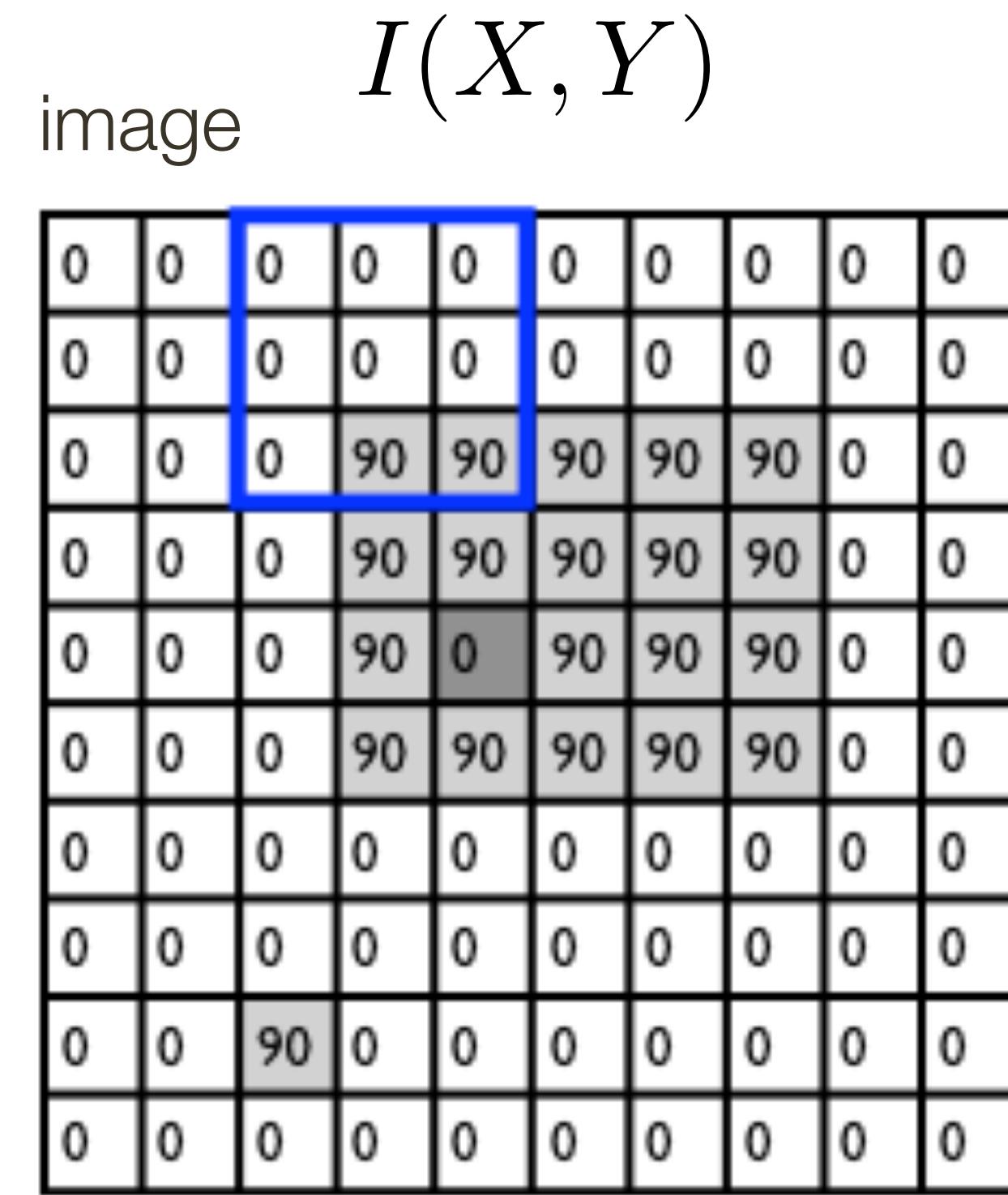
output

filter

image (signal)

Linear Filter Example

$$F(X, Y) \text{ filter}$$
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

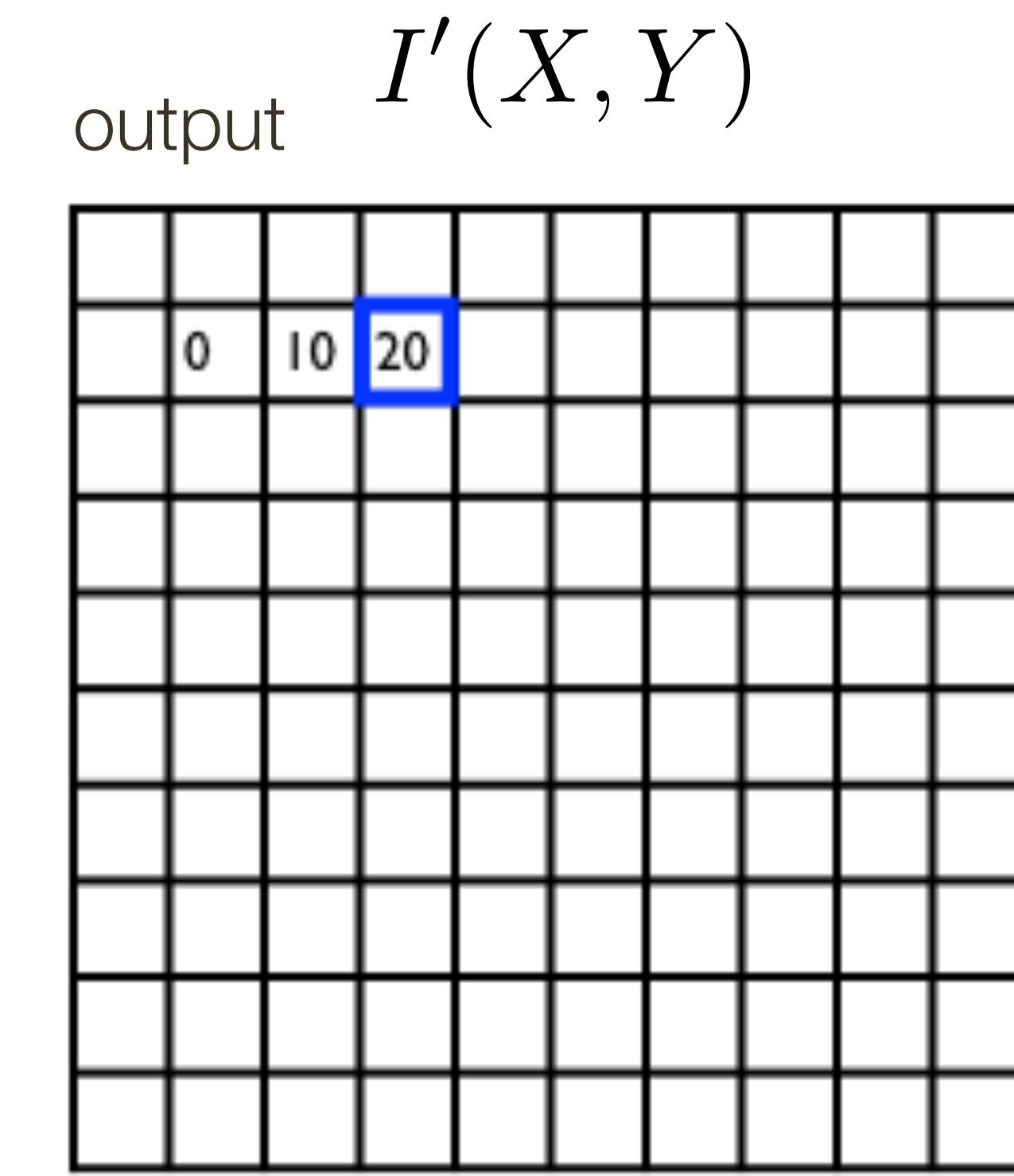
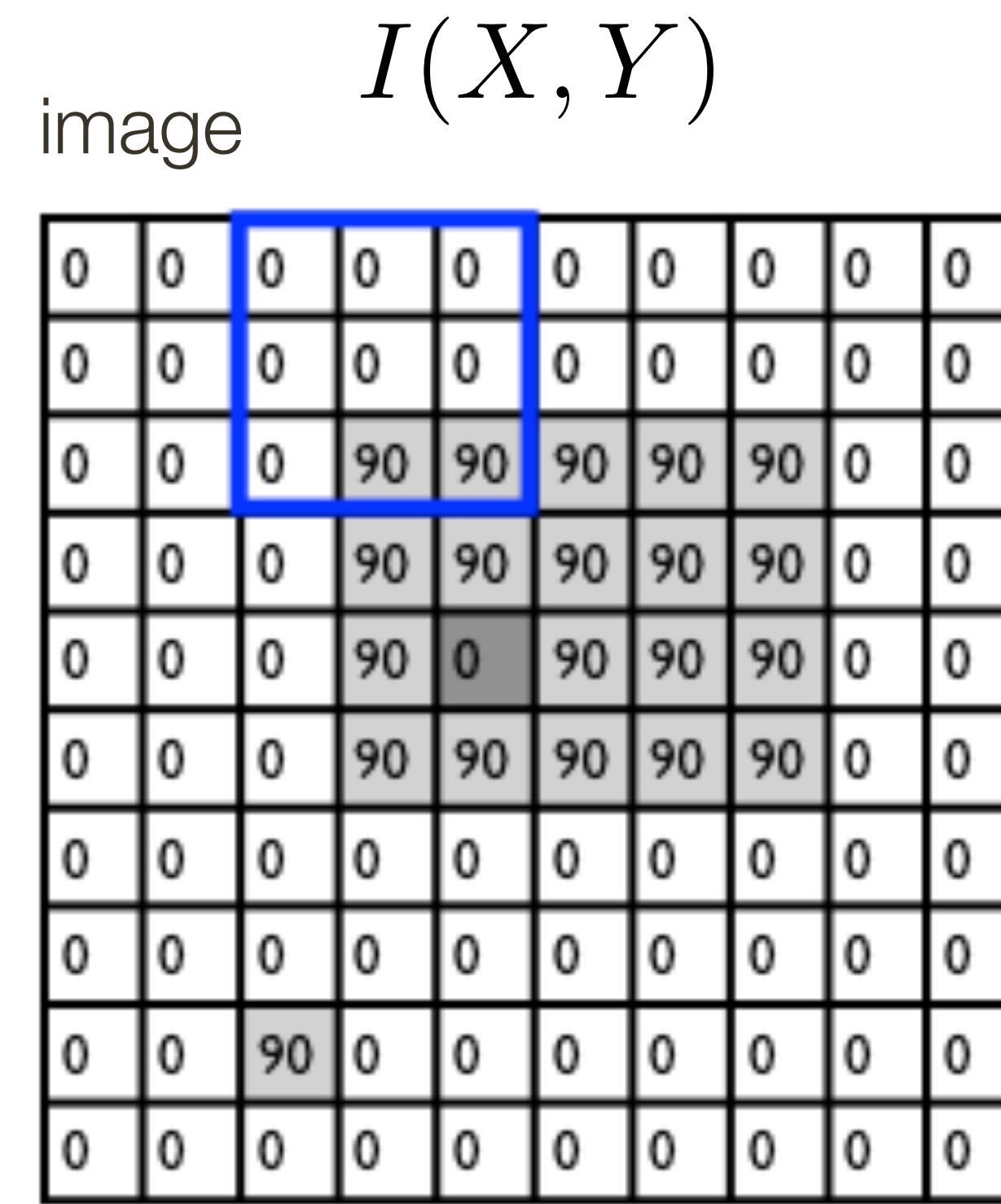


$$I'(X, Y) = \sum_{j=-k}^k \sum_{i=-k}^k F(i, j) I(X + i, Y + j)$$

output filter image (signal)

Linear Filter Example

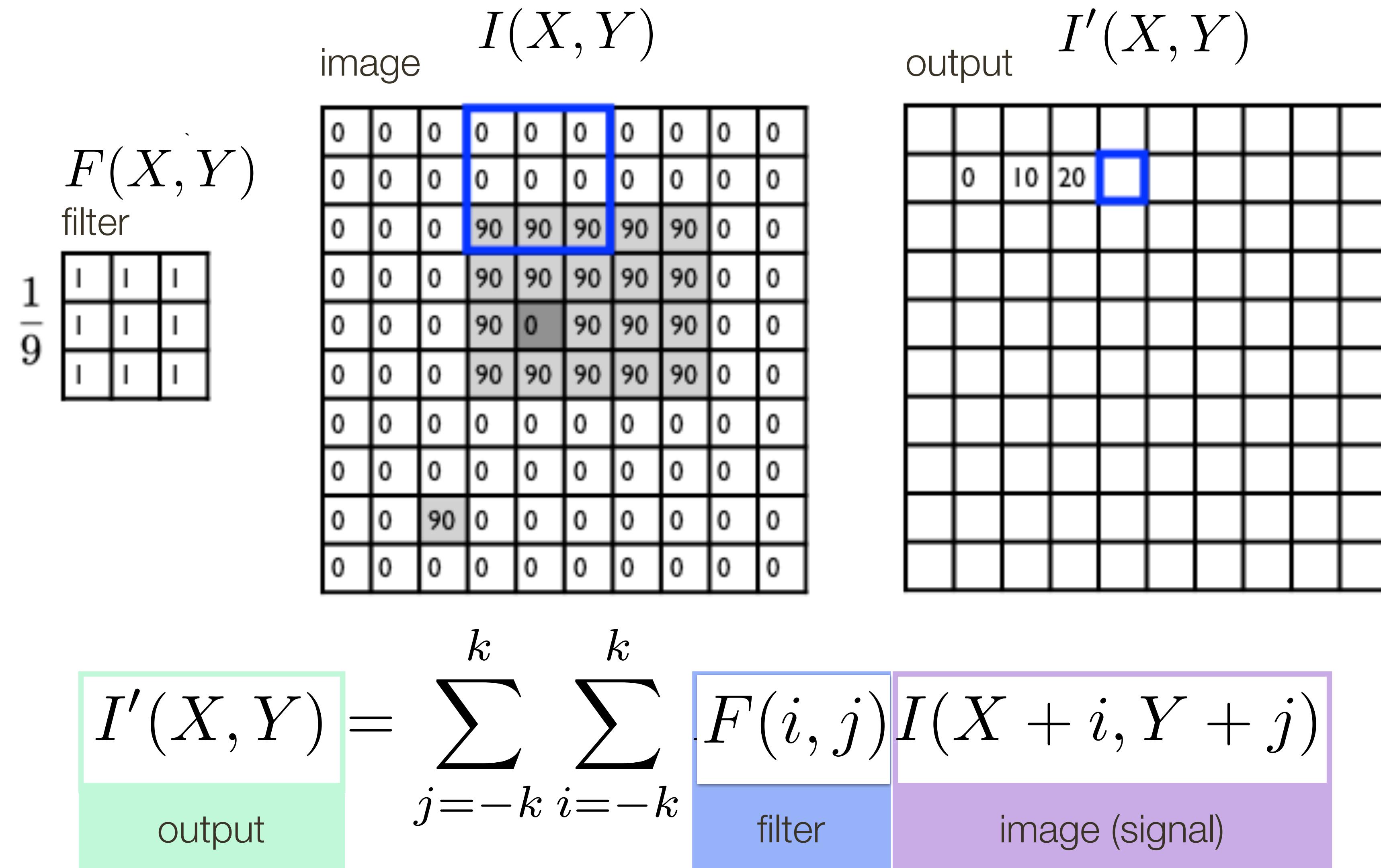
$$F(X, Y) \text{ filter}$$
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



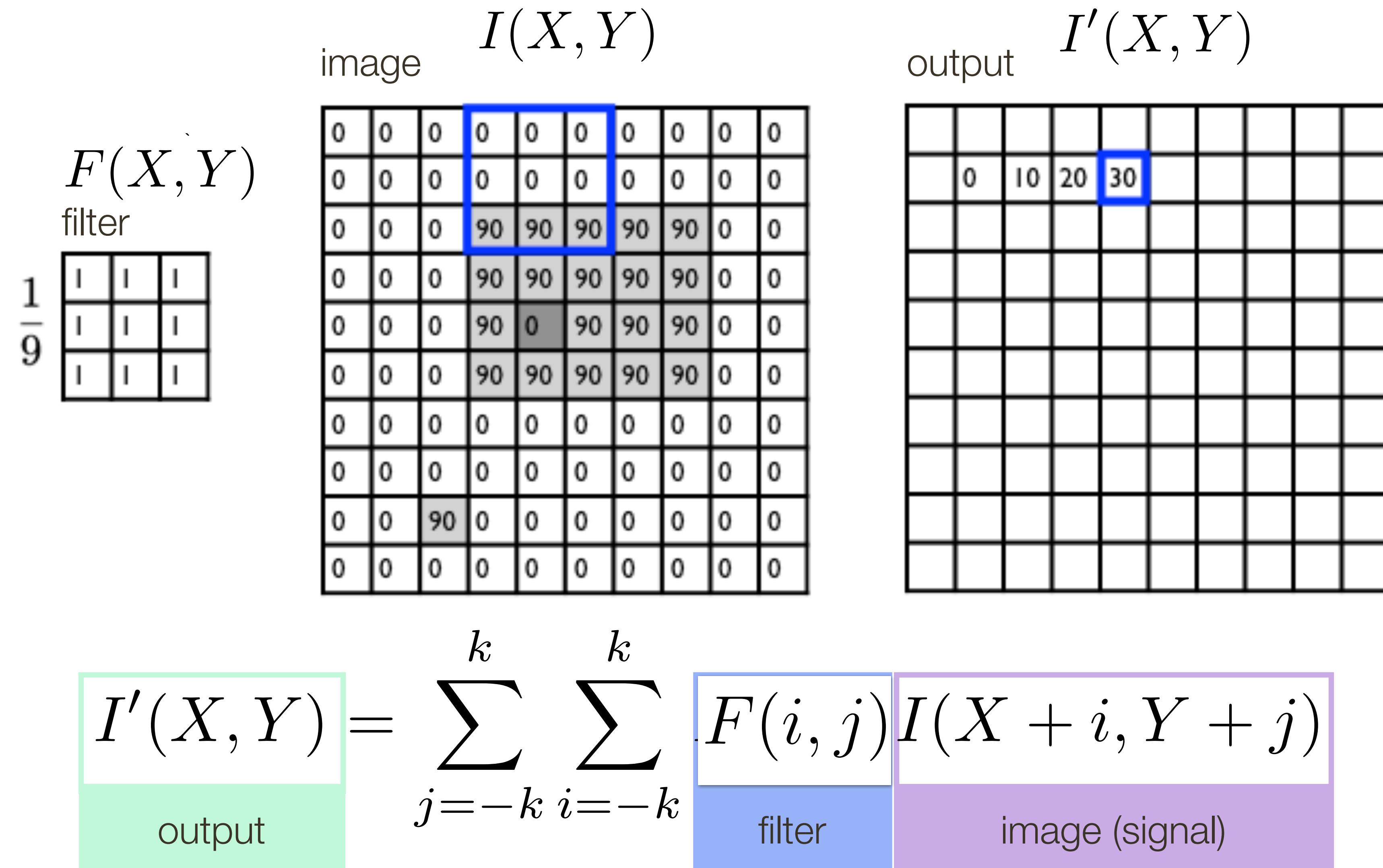
$$I'(X, Y) = \sum_{j=-k}^k \sum_{i=-k}^k F(i, j) I(X + i, Y + j)$$

output filter image (signal)

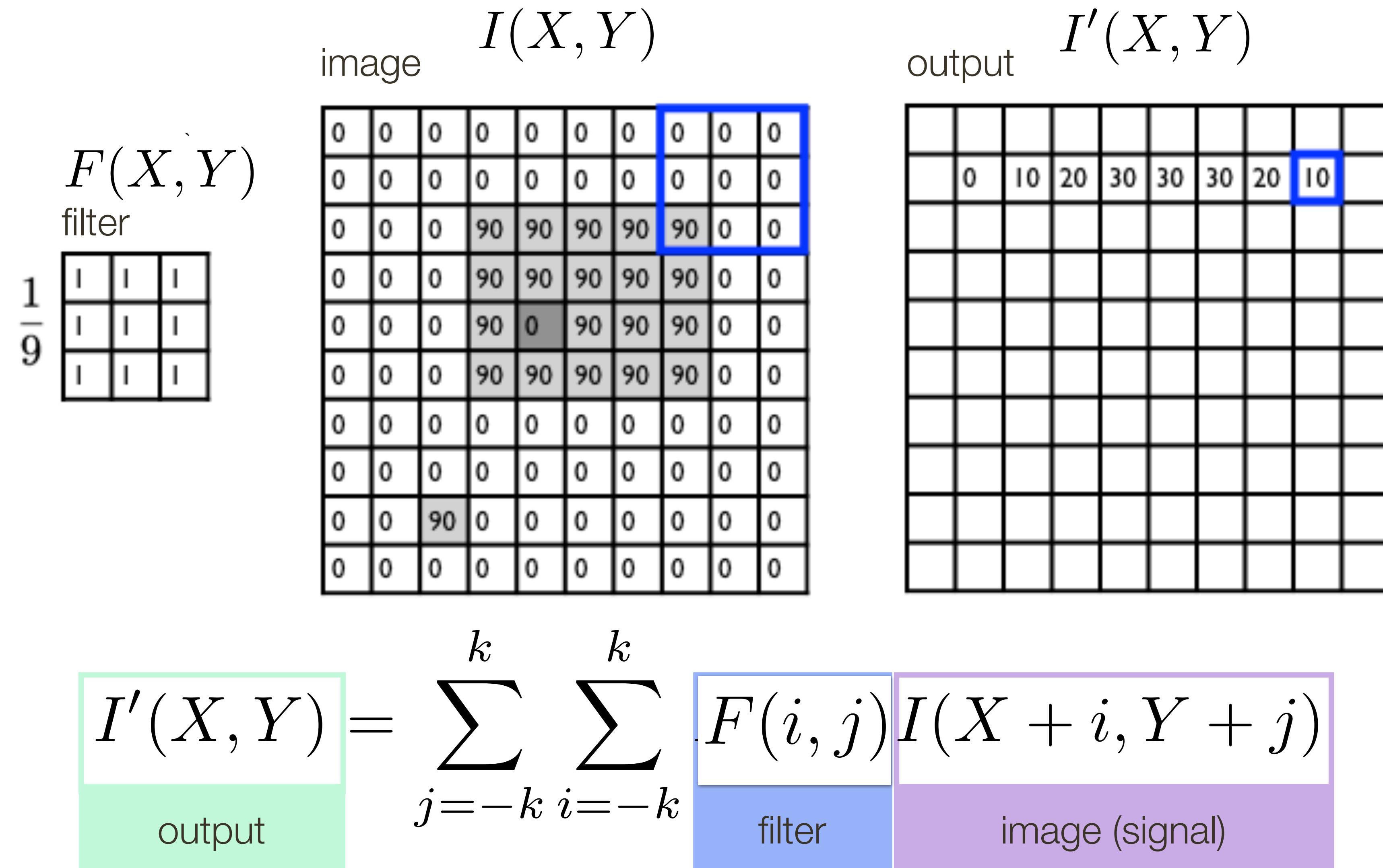
Linear Filter Example



Linear Filter Example

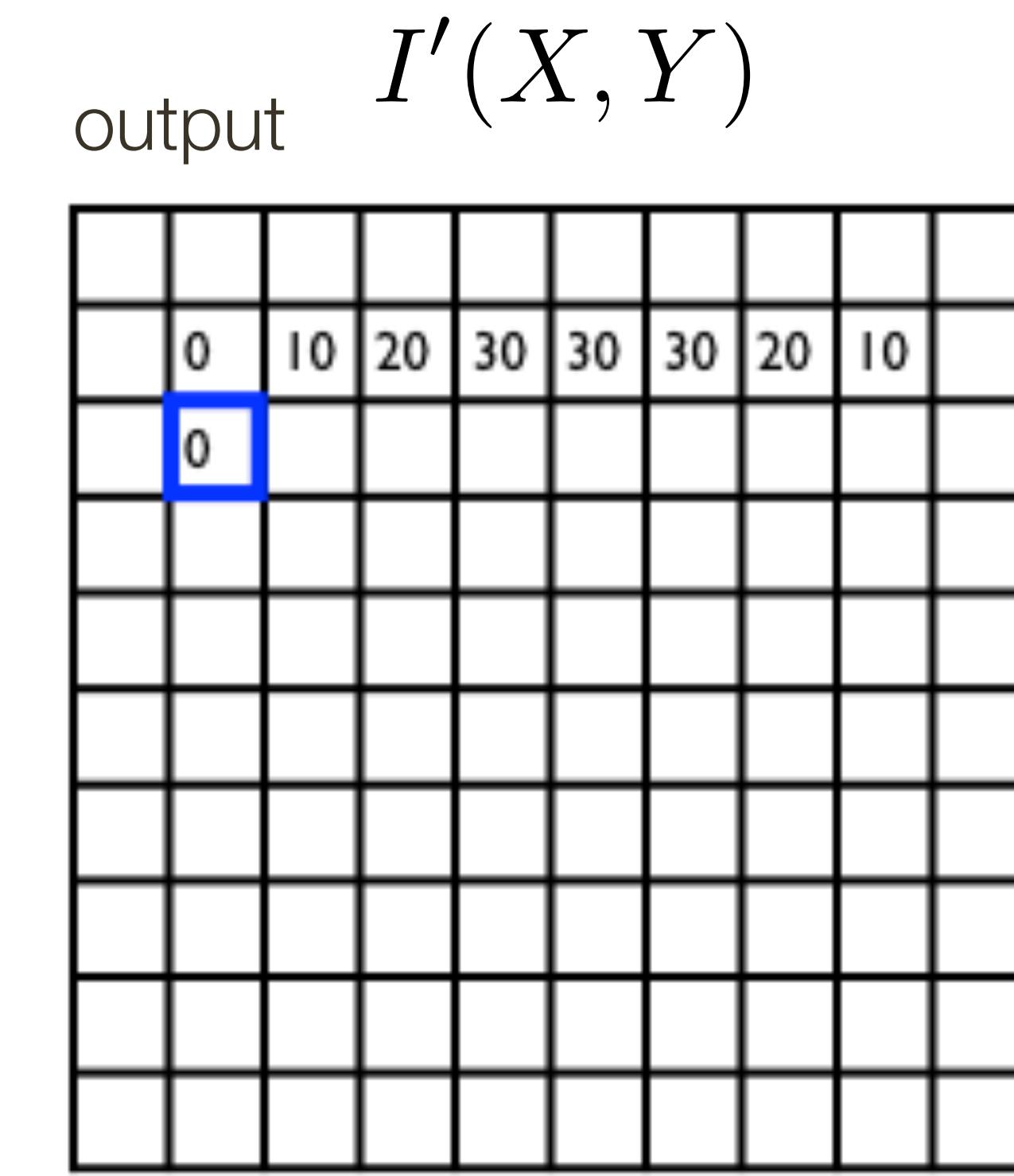
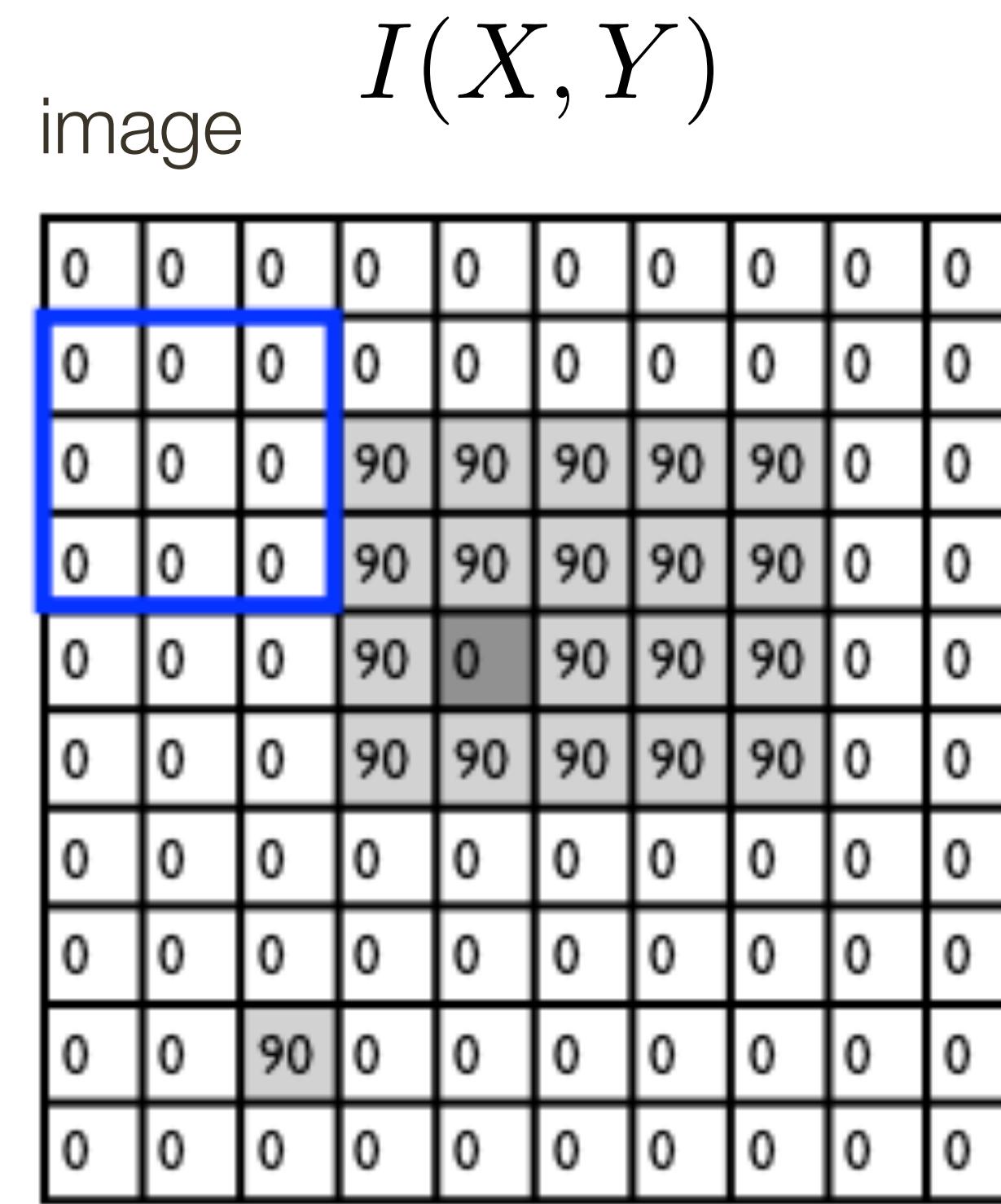


Linear Filter Example



Linear Filter Example

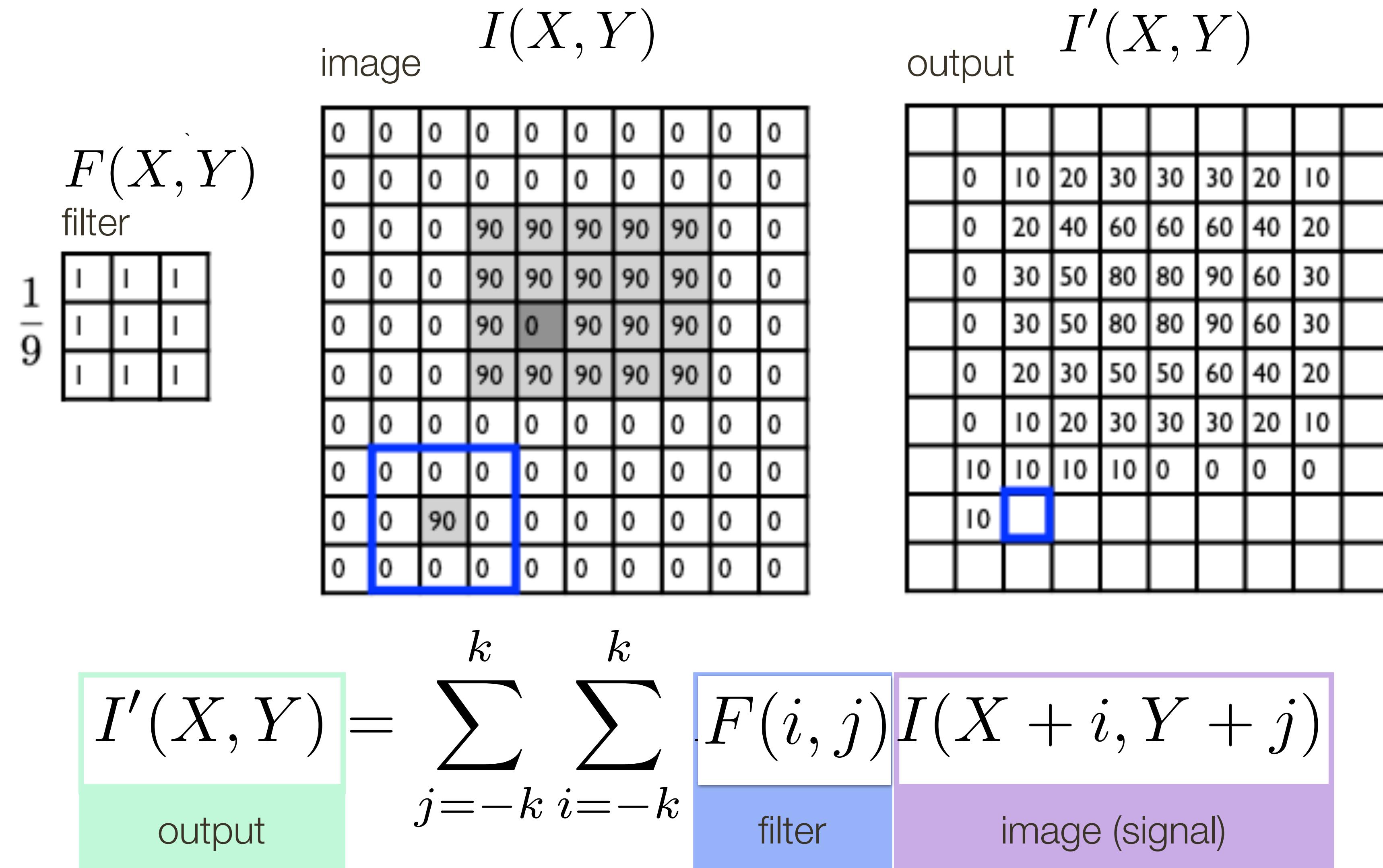
$$F(X, Y) \text{ filter}$$
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



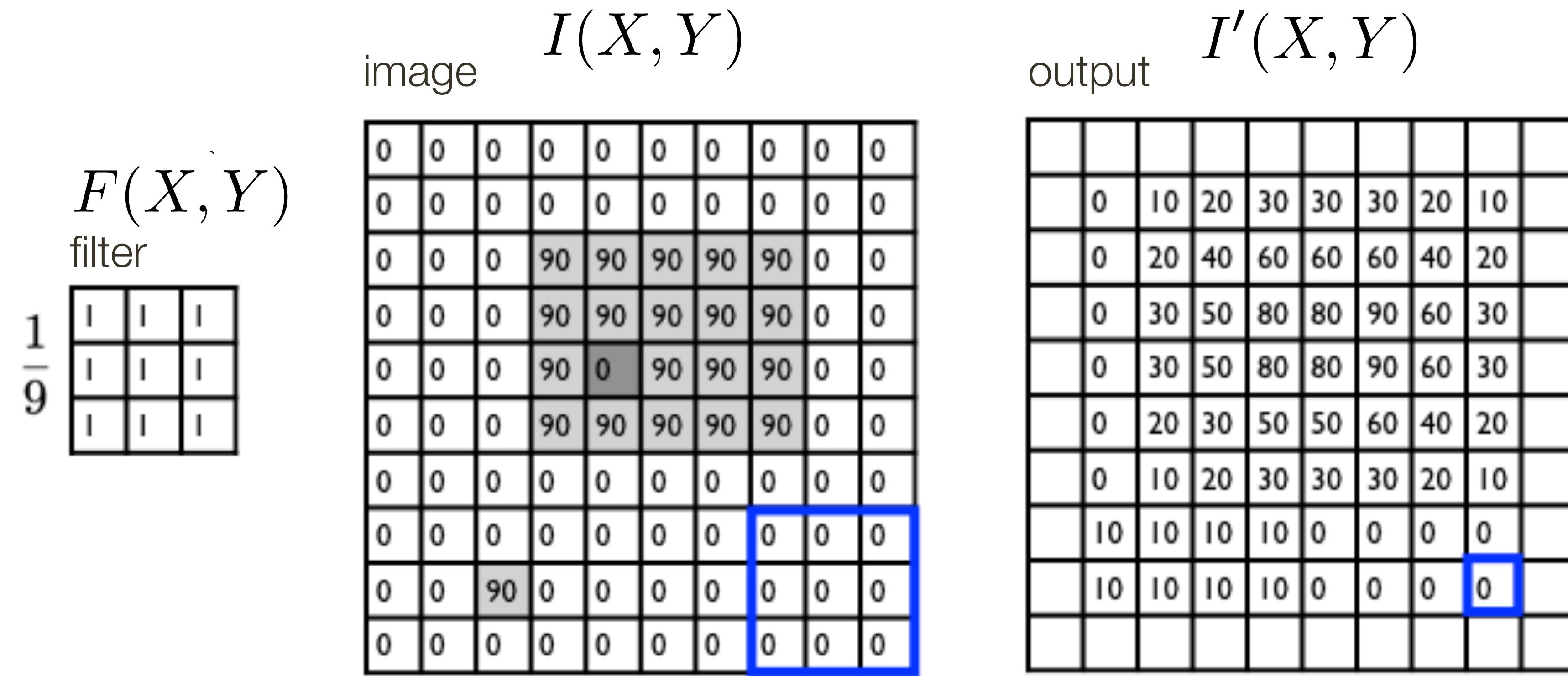
$$I'(X, Y) = \sum_{j=-k}^k \sum_{i=-k}^k F(i, j) I(X + i, Y + j)$$

output filter image (signal)

Linear Filter Example



Linear Filter Example



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

outputfilterimage (signal)

Linear Filter Example

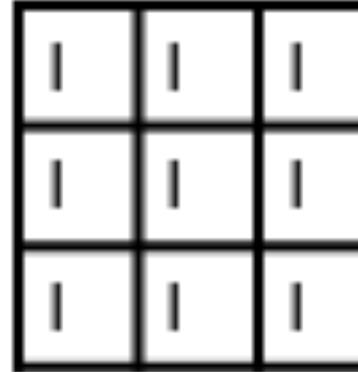
$F(X, Y)$
filter
 $\frac{1}{9}$


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	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)

Linear Filters: Properties



3.3

Convolution as matrix multiplication

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

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

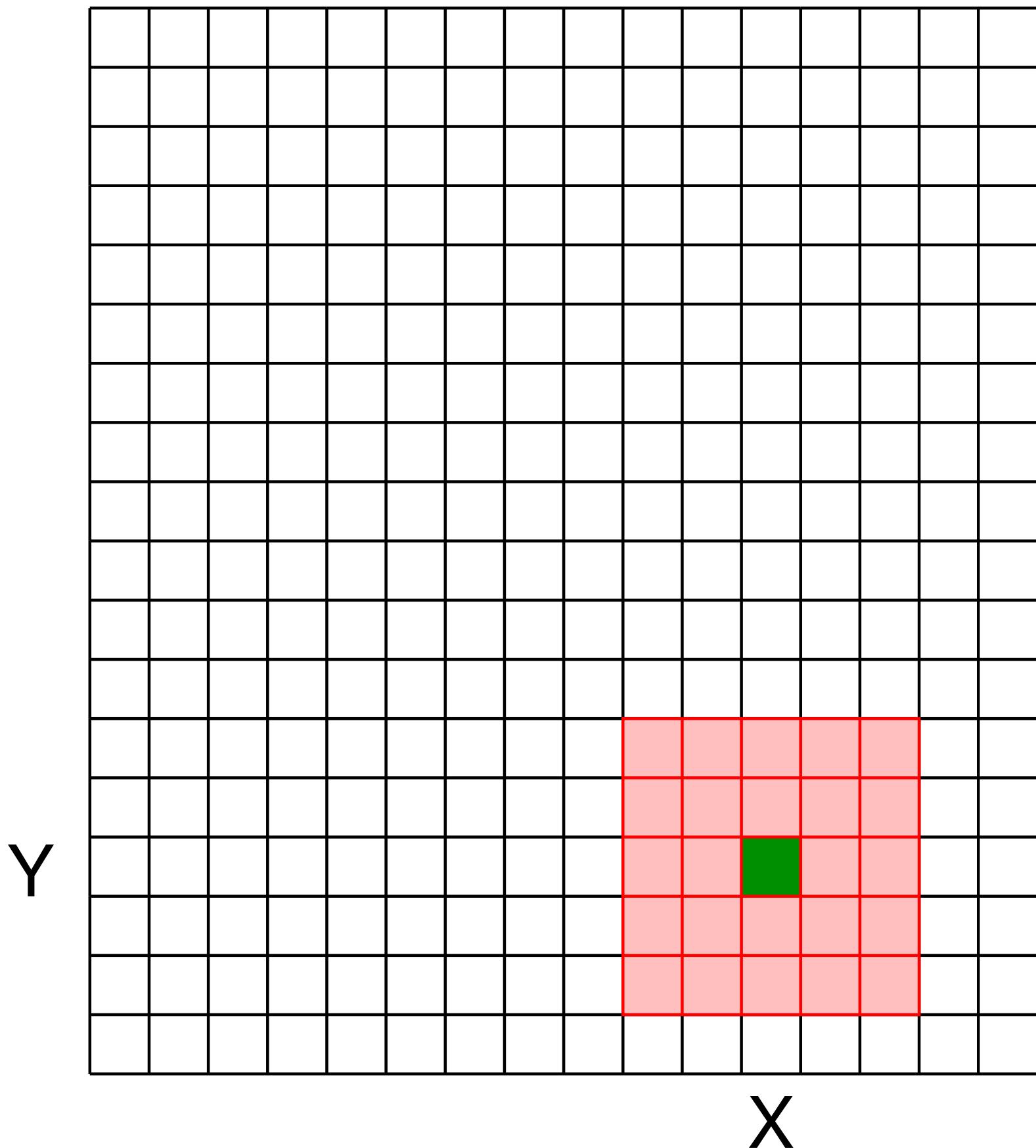
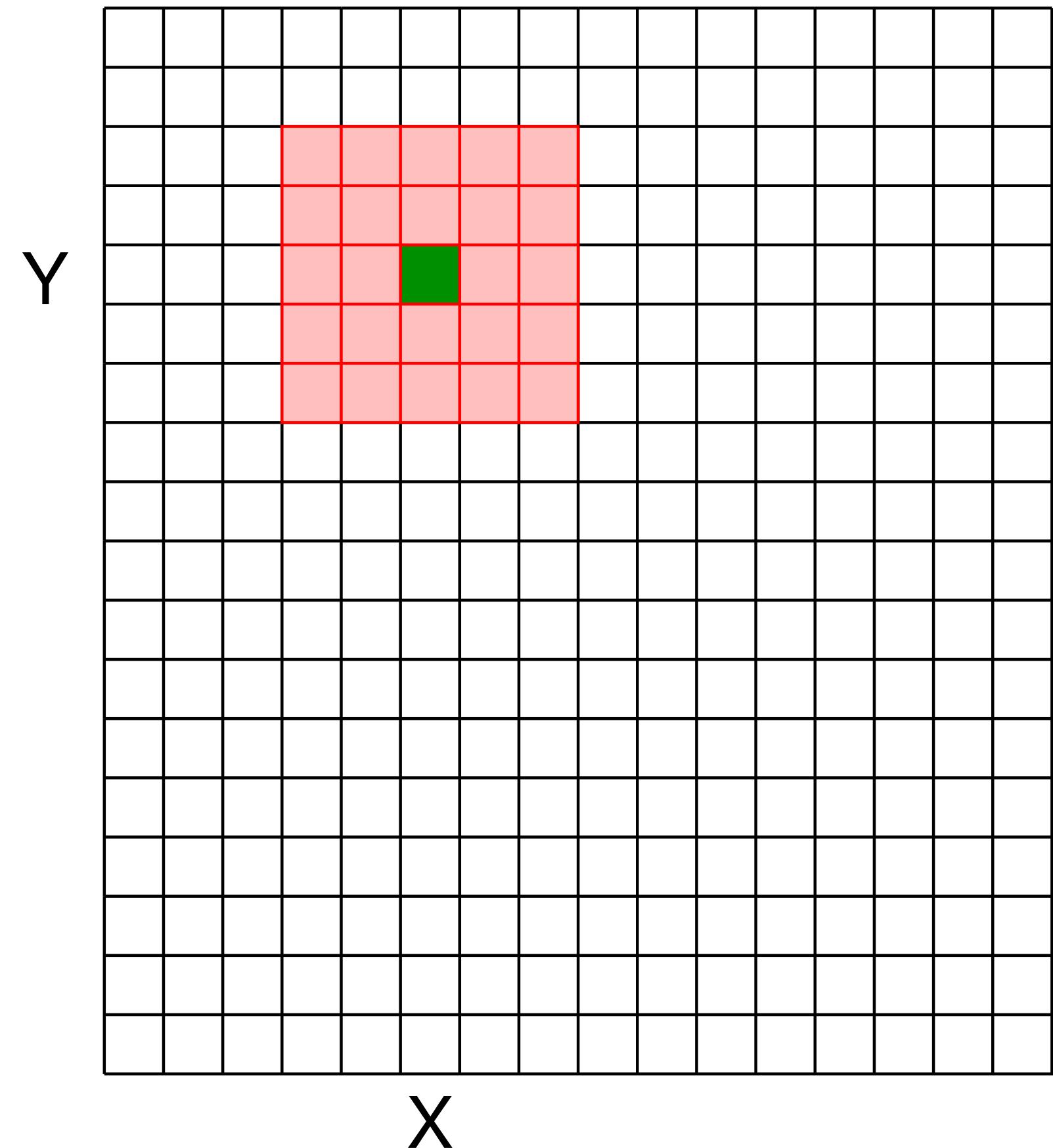
Scaling: Let F be digital filter and let k be a scalar

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

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

Linear Filters: Shift Invariance

Same linear operation is applied everywhere, no dependence on absolute position



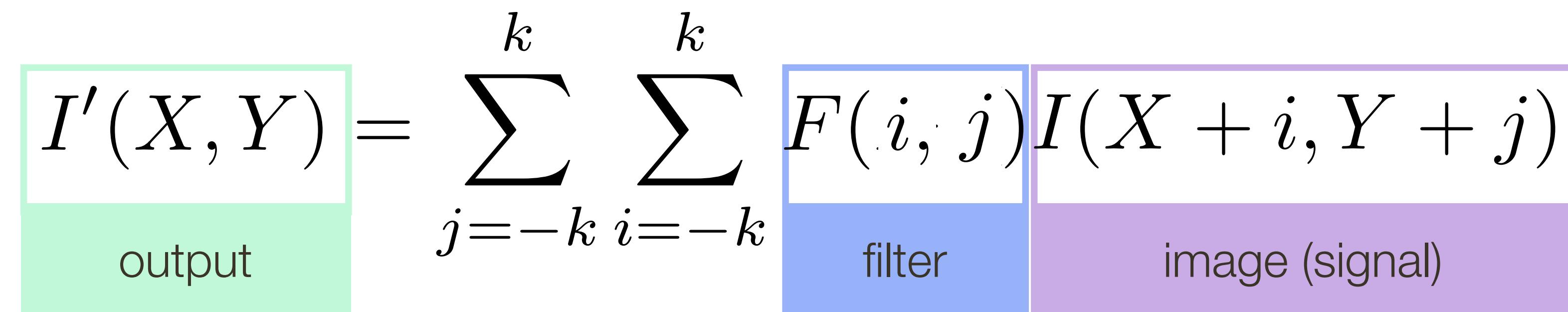
Linear Systems: Characterization Theorem

Any linear, shift invariant operation can be expressed as convolution

Lecture 3: Re-cap

- 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 rotated 180°
if $F(X, Y) = F(-X, -Y)$ then correlation = convolution.

Lecture 3: Re-cap

Ways to handle **boundaries**

- **Ignore/discard.** Make the computation undefined for top/bottom k rows and left/right-most k columns
- **Pad with zeros.** Return zero whenever a value of I is required beyond the image bounds
- **Assume periodicity.** Top row wraps around to the bottom row; leftmost column wraps around to rightmost column.

Simple **examples** of filtering:

- copy, shift, smoothing, sharpening

Linear filter **properties**:

- superposition, scaling, shift invariance

Characterization Theorem: Any linear, shift-invariant operation can be expressed as a convolution

iClicker test

Please sign up for the iClicker course **CPSC 425 101 2023W1 Computer Vision**

See also the [UBC iClicker Student Guide](#), contact LT Hub if stuck

The screenshot shows a web browser window with the title bar "UBC iClicker Cloud Student Guide". The address bar contains the URL "https://lthub.ubc.ca/guides/iclicker-cloud-student-guide/". The page itself is the "iClicker Cloud Student Guide" from the Learning Technology Hub at UBC. It features the UBC logo and the text "THE UNIVERSITY OF BRITISH COLUMBIA". A blue header bar includes the "Learning Technology Hub" logo. Below the header, there's a navigation menu with links to Home, Tool Finder, Tool Guides, Support, Student Support, Initiatives, Governance, and News. The main content area starts with the heading "iClicker Cloud Student Guide" and a large blue circular icon containing a white cloud shape. To the right, there's a section titled "What will I use it for?" with a list of bullet points about its use in class.

iClicker Cloud Student Guide



iClicker Cloud is an online student response system that allows you to respond individually to in-class polls and low-stakes quizzes, using your own computer or mobile device. Your instructor receives the responses instantly and may share these results and correct answers in the tool during the live lecture or afterward.

iClicker Cloud has passed a UBC [Privacy Impact Assessment](#), meaning it follows UBC and provincial privacy

What will I use it for?

Your instructor may have you use iClicker for a variety of activities:

- Test your knowledge or opinions at different points in the class for marks
- Support peer instruction, wherein you answer a question, discuss in small

Quiz 0, Question 1 (This is a **test** quiz, not for credit)

Python is:

- A) A large snake
- B) A programming language
- C) Both A) and B)
- D) I'll click a random answer and get 1/2 marks

Quiz 0, Question 1 (This is a **test** quiz, not for credit)

Python is:

- A) A large snake
- B) A programming language
- C) Both A) and B)
- D) I'll click a random answer and get 1/2 marks

Quiz 0, Question 2

Which of the following are true about a “Camera Obscura”?

- A) It is a compound lens design to reduce aberrations
- B) It is an unusual design of colour filter array
- C) It's what happens when someone stands in front of your picture
- D) It is a dark room
- E) A, B and C

Quiz 0, Question 2

Which of the following are true about a “Camera Obscura”?

- A) It is a compound lens design to reduce aberrations
- B) It is an unusual design of colour filter array
- C) It's what happens when someone stands in front of your picture
- D) It is a dark room
- E) A, B and C

Quiz 0, Question 3

The purpose of a lens in a camera is

- A) To focus rays on the pinhole
- B) To increase the amount of light entering
- C) To prevent chromatic aberration
- D) To enable orthographic projection
- E) C and D

Quiz 0, Question 3

The purpose of a lens in a camera is

- A) To focus rays on the pinhole
- B) To increase the amount of light entering
- C) To prevent chromatic aberration
- D) To enable orthographic projection
- E) C and D

Quiz 0, Question 4

A Lambertian surface

- A) Reflects light such that angle of incidence = angle of reflection
- B) Has a specular lobe around the reflection direction
- C) Reflects light equally in all directions
- D) A and C
- E) B and C

Quiz 0, Question 4

A Lambertian surface

- A) Reflects light such that angle of incidence = angle of reflection
- B) Has a specular lobe around the reflection direction
- C) Reflects light equally in all directions
- D) A and C
- E) B and C

Quiz 0, Question 5

Correlation is the same as convolution if

- A) The filter is rotated by 90 degrees
- B) The filter satisfies $F(X,Y)=F(-X,-Y)$
- C) The filter is rotated by 180 degrees
- D) A and B
- E) B and C

Quiz 0, Question 5

Correlation is the same as convolution if

- A) The filter is rotated by 90 degrees
- B) The filter satisfies $F(X,Y)=F(-X,-Y)$
- C) The filter is rotated by 180 degrees
- D) A and B
- E) B and C

Lecture Feedback

Quiz 0b, Question 1

Speed of the lectures is:

- A) Too fast!
- B) A little too fast
- C) Perfect
- D) A little too slow
- E) Too slow!

Quiz 0b, Question 2

Volume of the lectures is:

- A) Too loud!
- B) A little too loud
- C) Perfect
- D) A little too quiet
- E) Too quiet!

Quiz 0b, Question 3

Difficulty of the content is:

- A) Too hard!
- B) A little too hard
- C) Perfect
- D) A little too easy
- E) Too easy!

Lecture Feedback

Feel free to send other feedback via email or Piazza!

Smoothing with a **Box Filter**

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$



Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Filter has equal positive values that sum up to 1

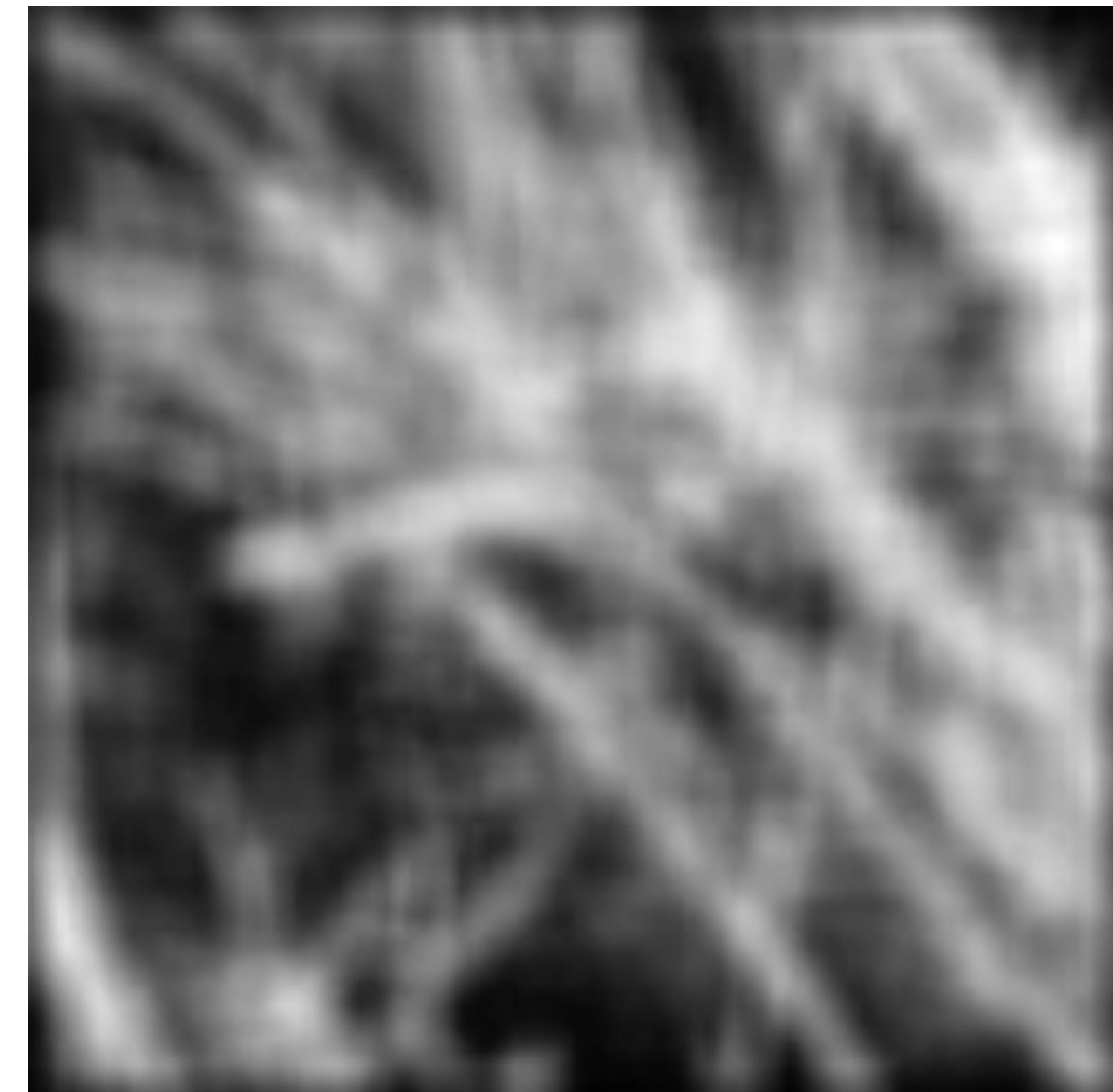
Replaces each pixel with the average of itself and its local neighborhood

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



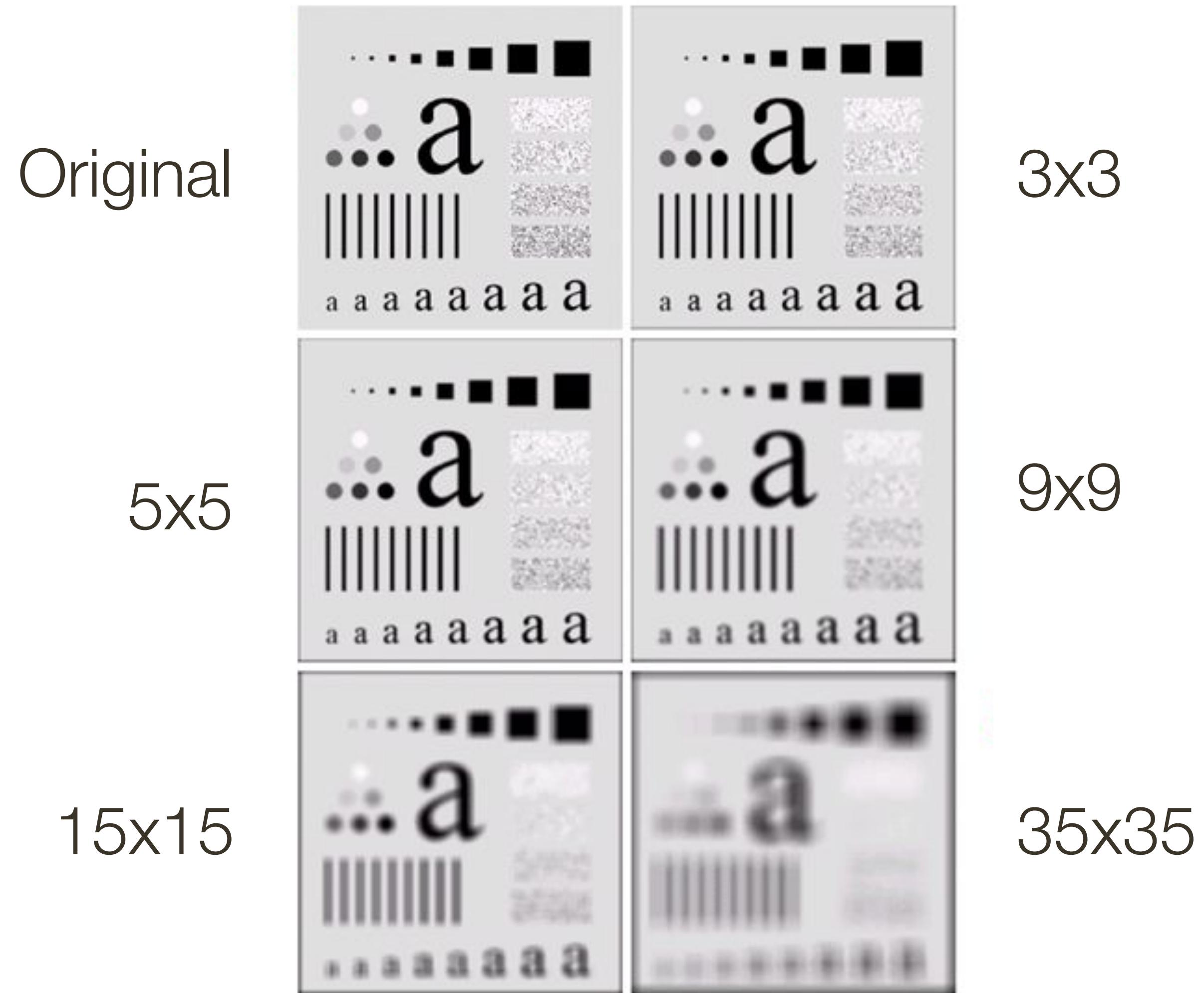
Why should the values sum to 1?

Smoothing with a **Box Filter**



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

Smoothing with a **Box Filter**



Gonzales & Woods (3rd ed.) Figure 3.3

Smoothing with a **Box Filter**

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

- Smoothing with a box filter depends on direction
- e.g., Image in which the center point is 1 and every other point is 0
- Point spread function is a box

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

1	1	1
1	1	1
1	1	1

Filter

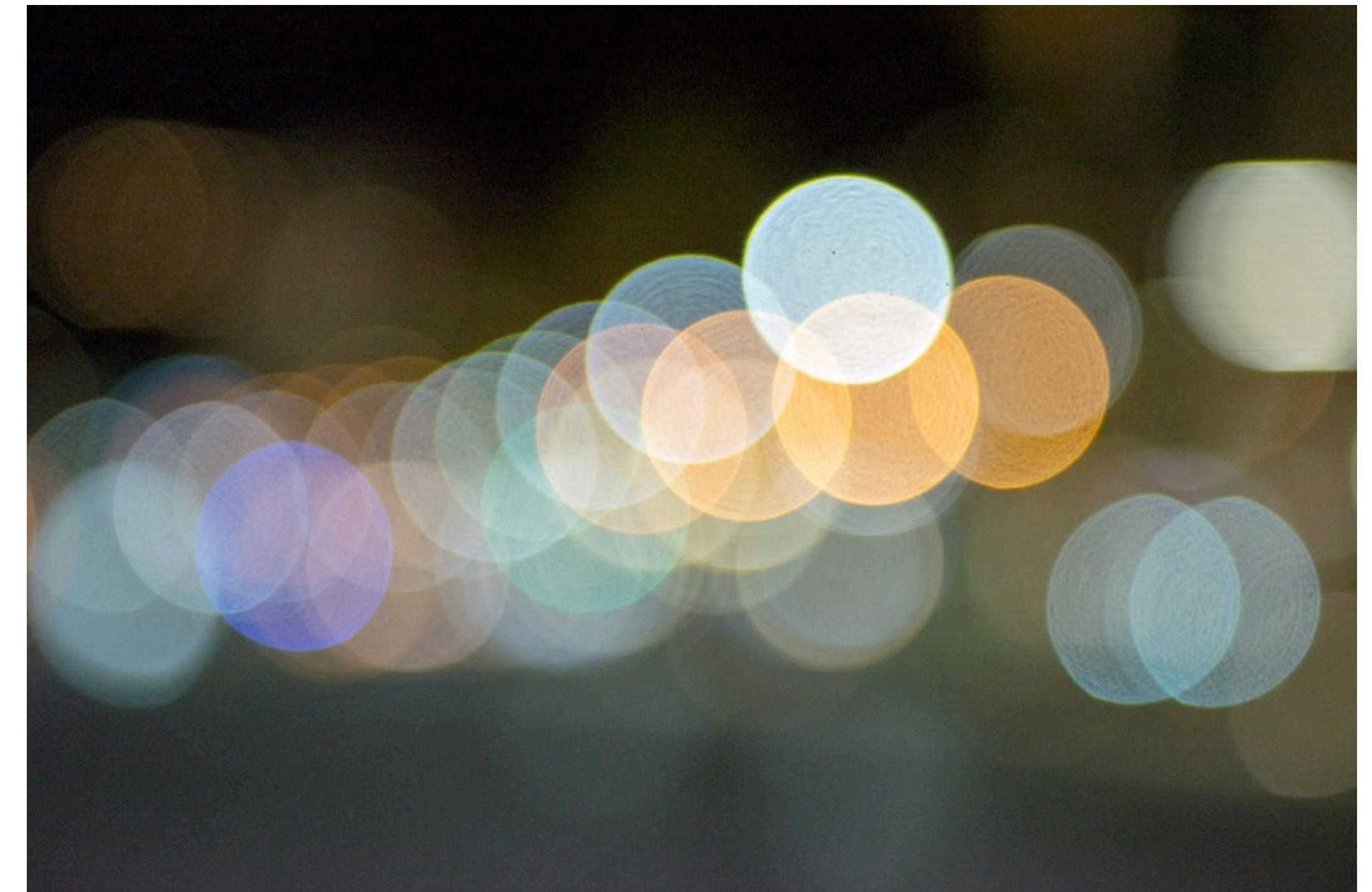
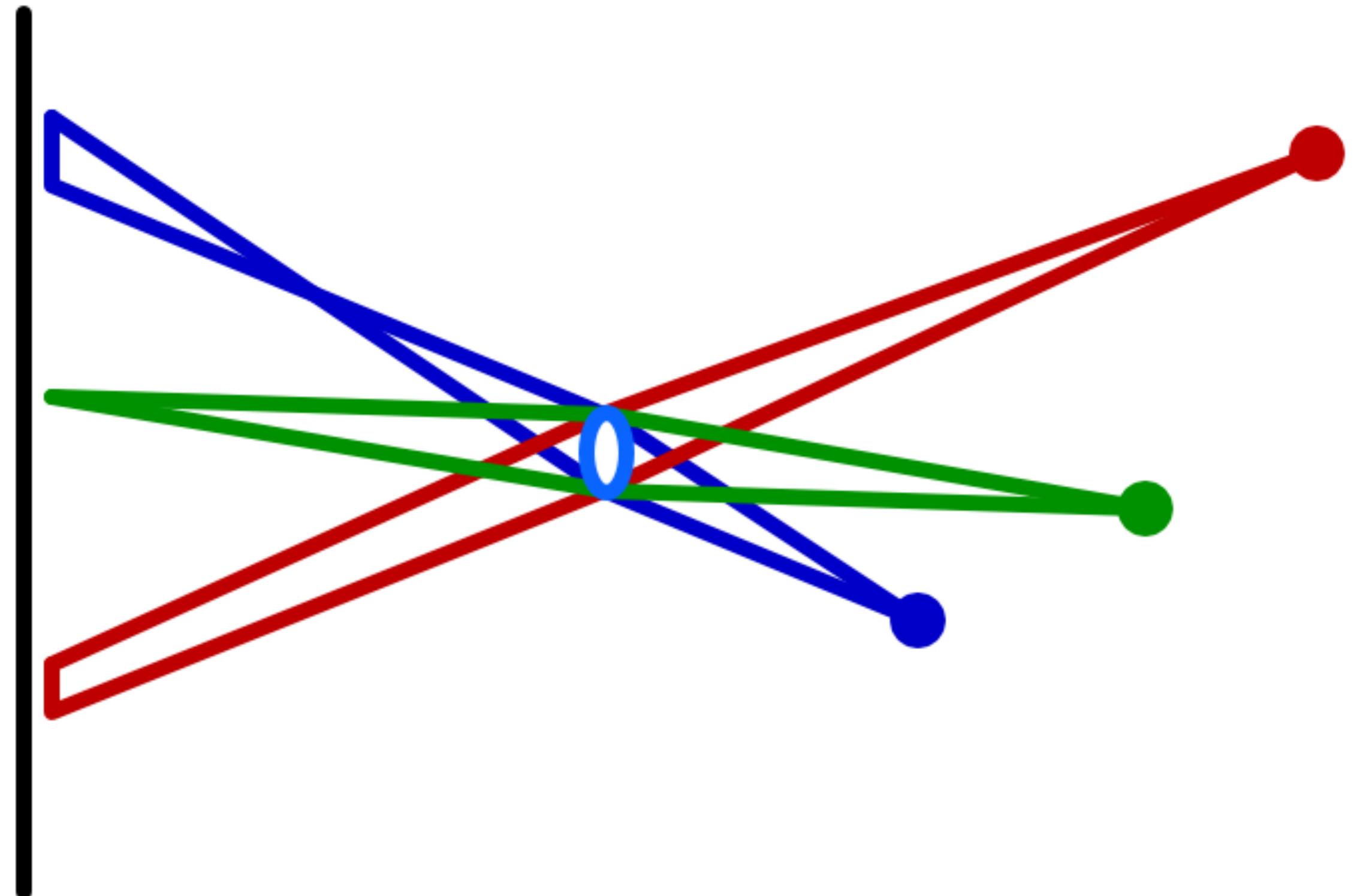
0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

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: Circular Kernel



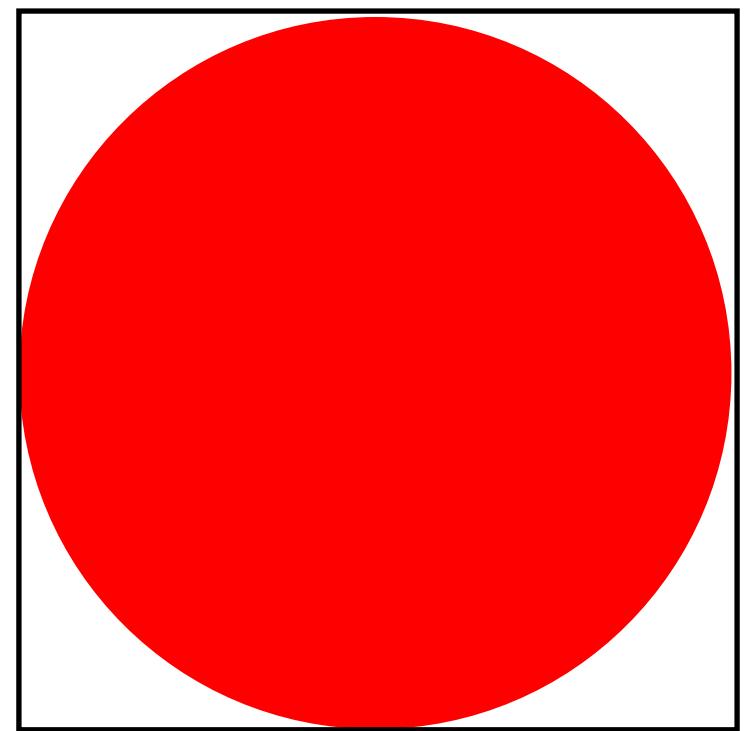
* image credit: <https://catlikecoding.com/unity/tutorials/advanced-rendering/depth-of-field/circle-of-confusion/lens-camera.png>

Pillbox Filter

Let the radius (i.e., half diameter) of the filter be r

In a contentious domain, a 2D (circular) pillbox filter, $f(x, y)$, is defined as:

$$f(x, y) = \frac{1}{\pi r^2} \begin{cases} 1 & \text{if } x^2 + y^2 \leq r^2 \\ 0 & \text{otherwise} \end{cases}$$



The scaling constant, $\frac{1}{\pi r^2}$, ensures that the area of the filter is one

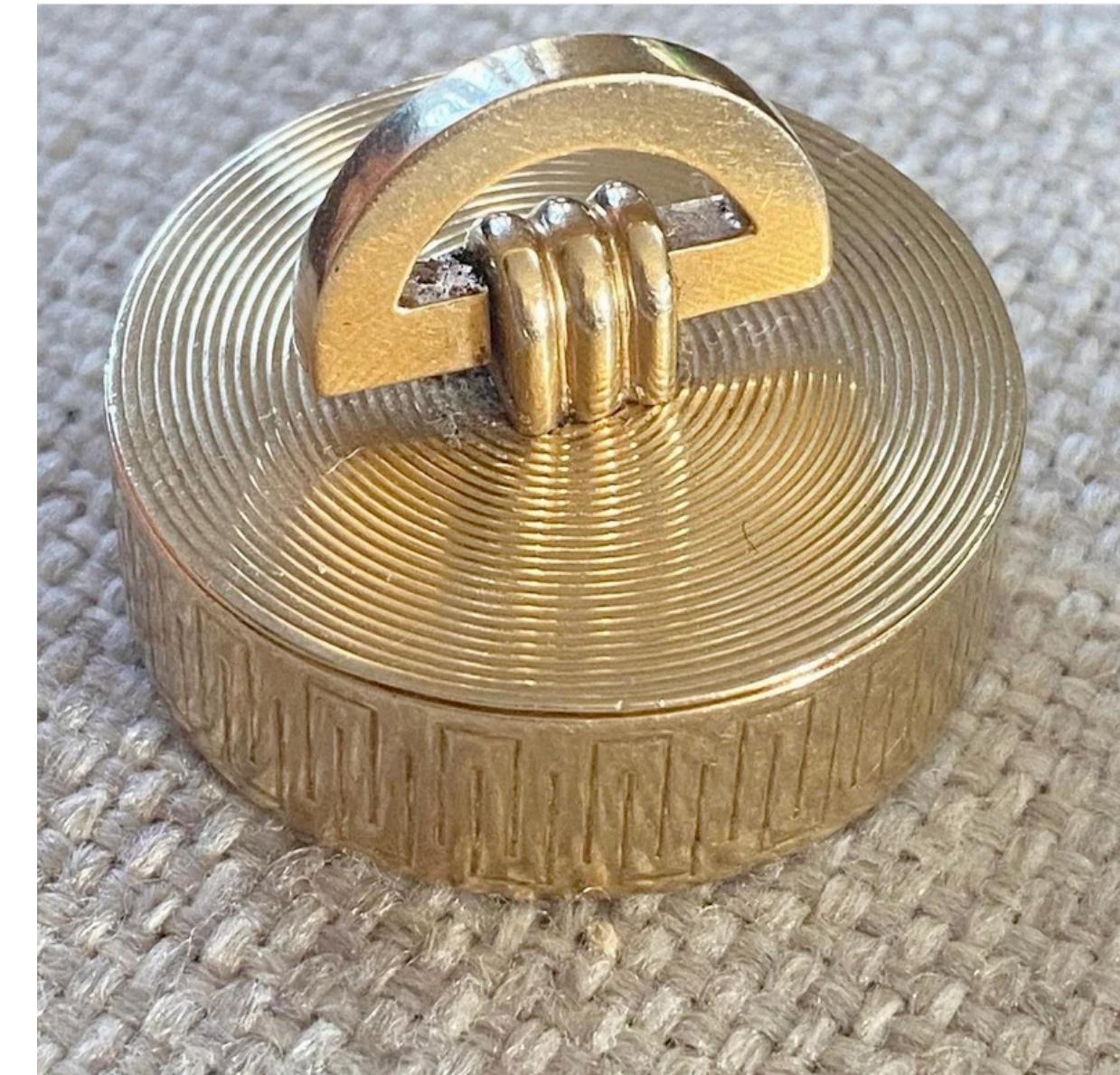
Aside: Q What do the following have in common



A hat



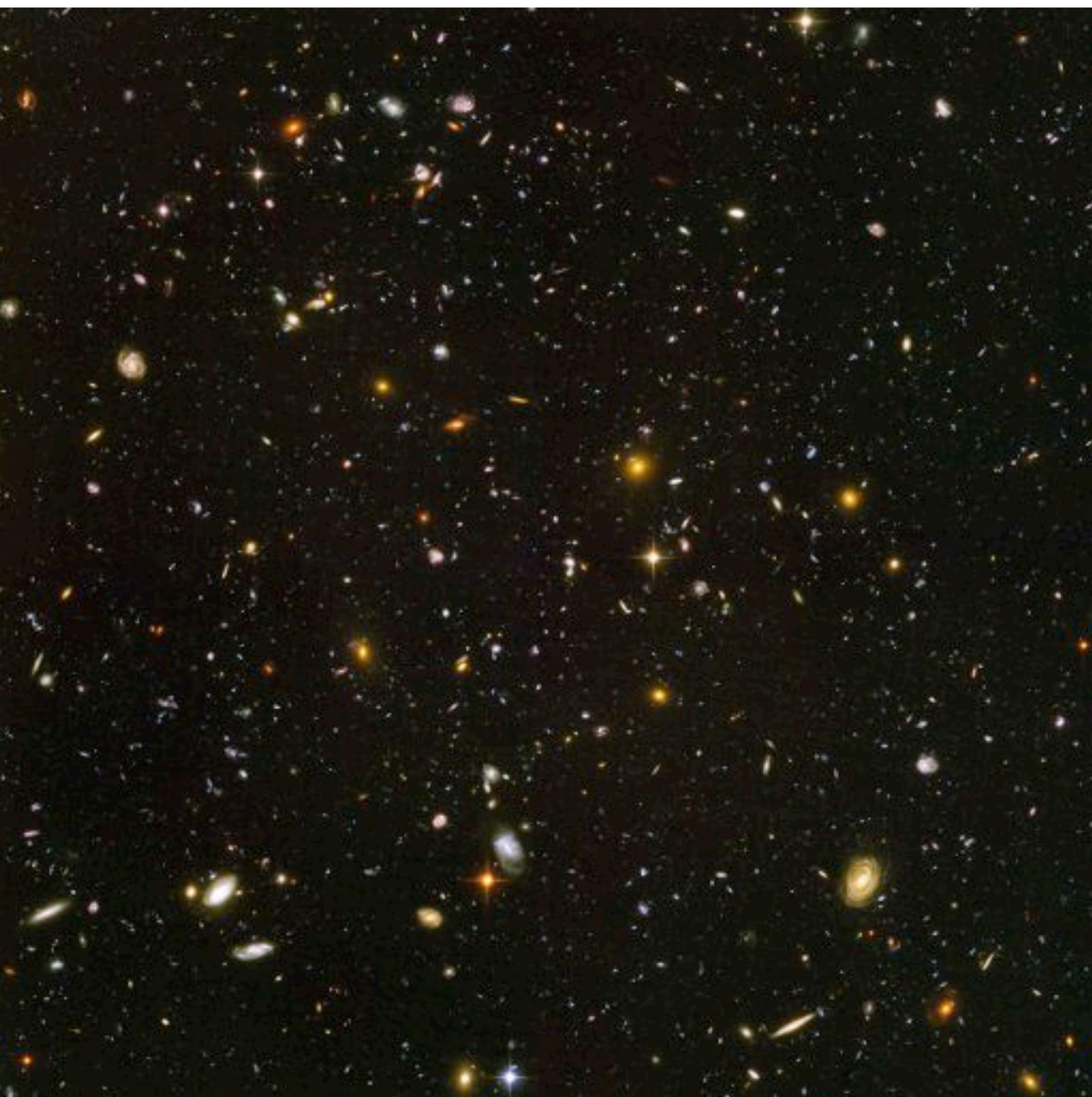
A military bunker



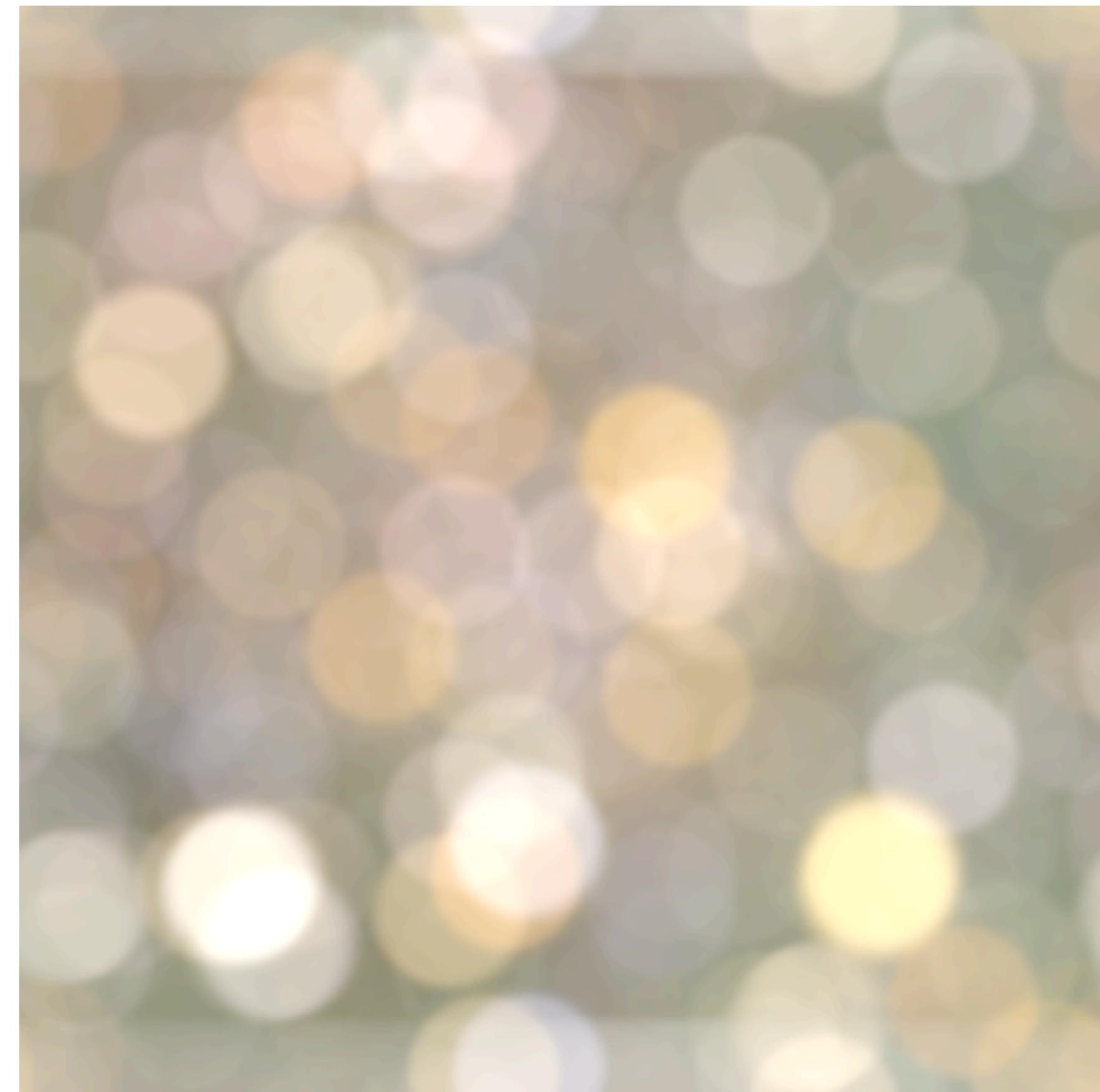
An old container

A They are all “Pillboxes”

Circular Blur Kernel



Hubble Deep View



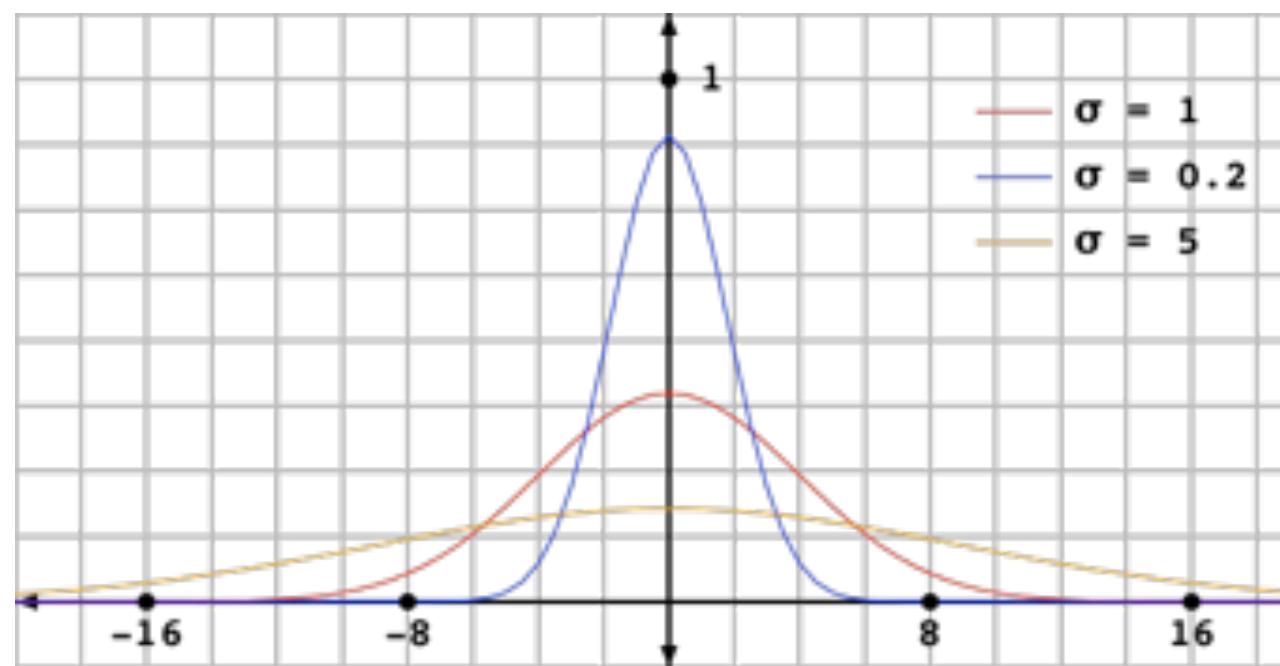
With Circular Blur

Images: yehar.com

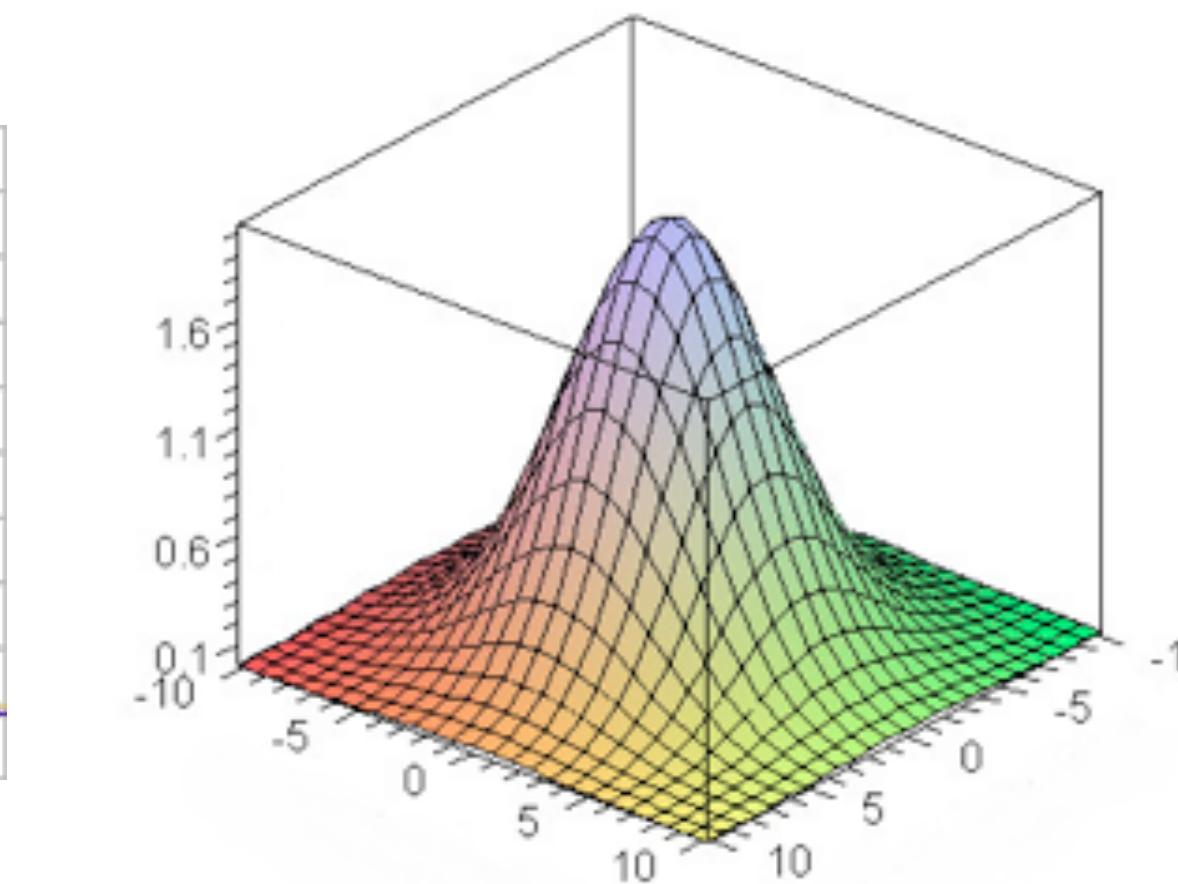
Gaussian Blur

- Gaussian kernels are often used for smoothing and resizing images

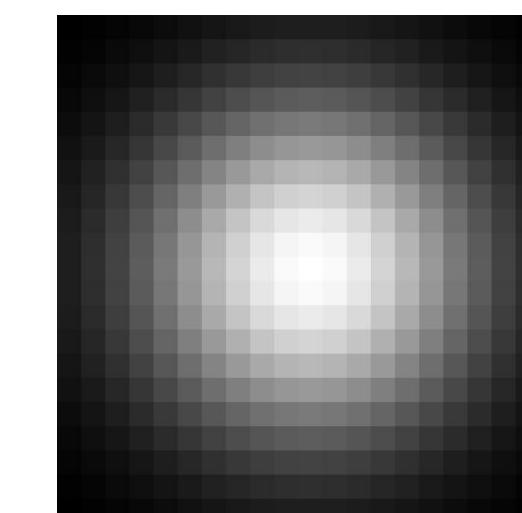
ID



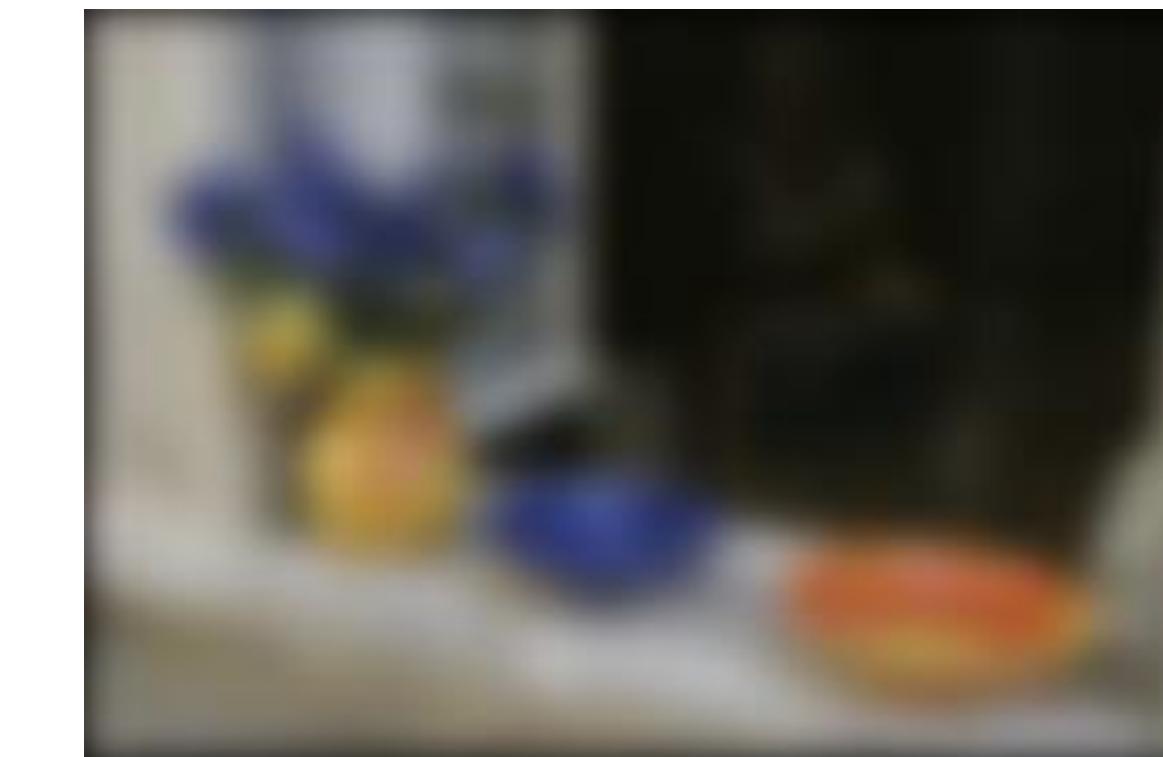
2D



*



=

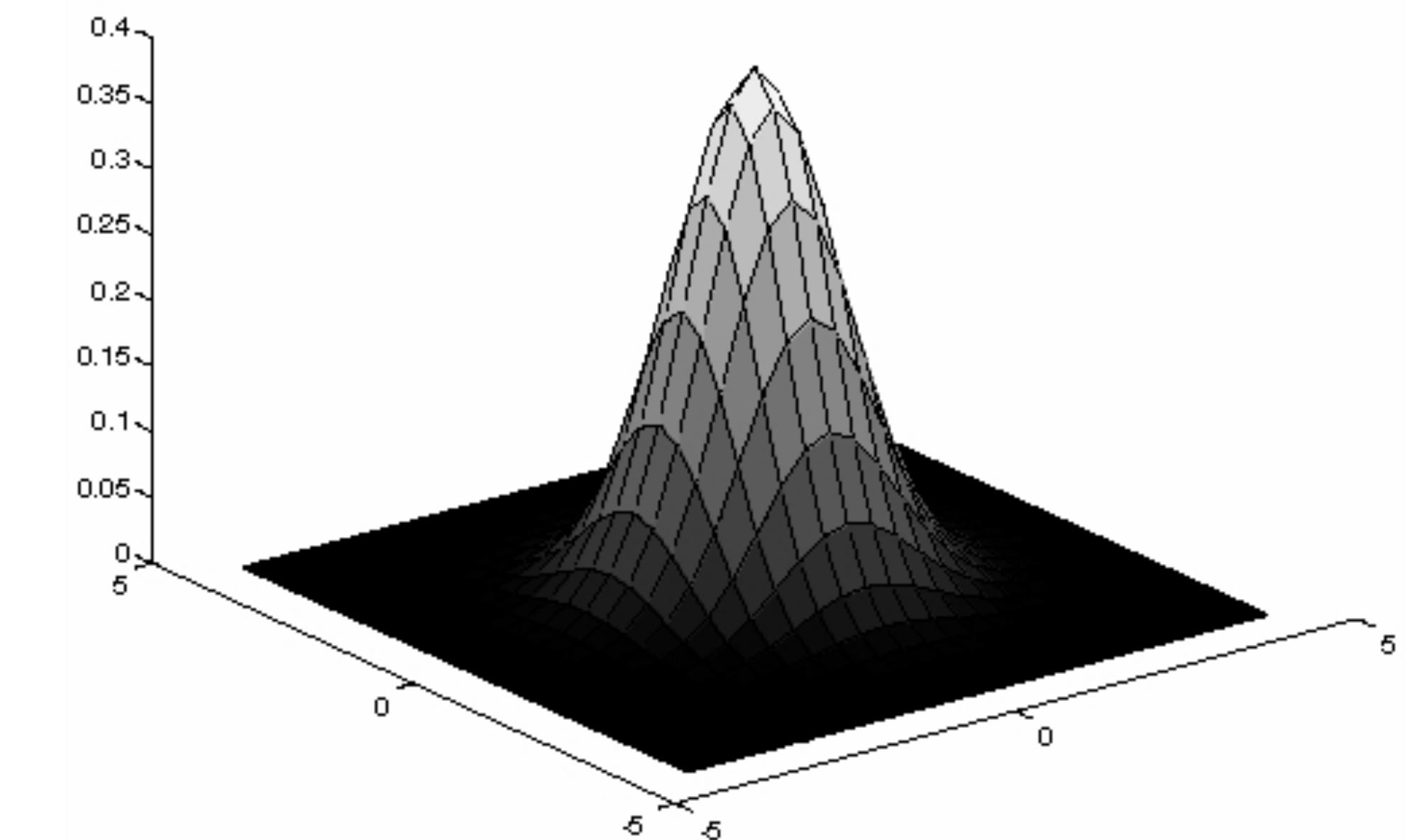


Smoothing with a **Gaussian**

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



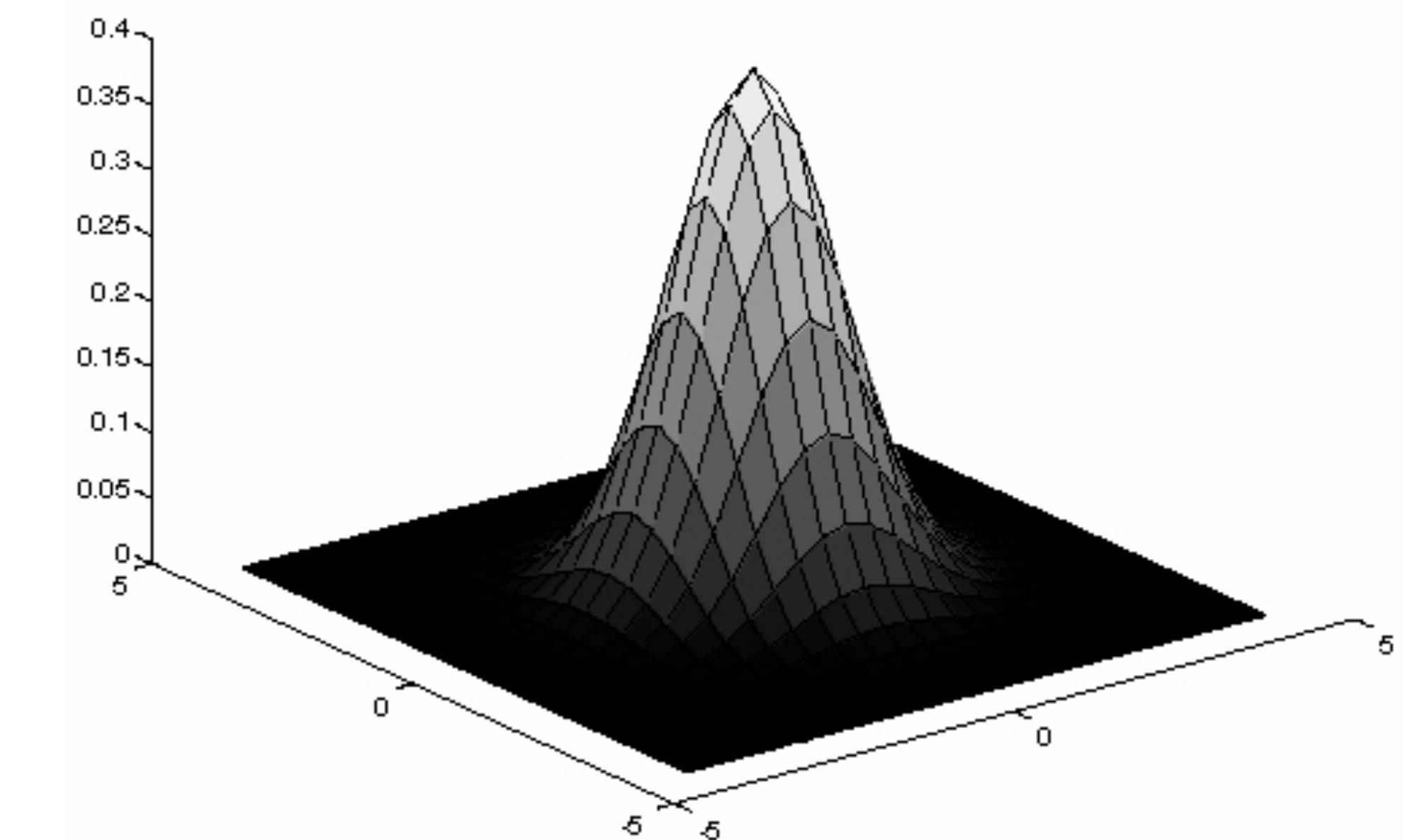
Example 6: Smoothing with a Gaussian

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:

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

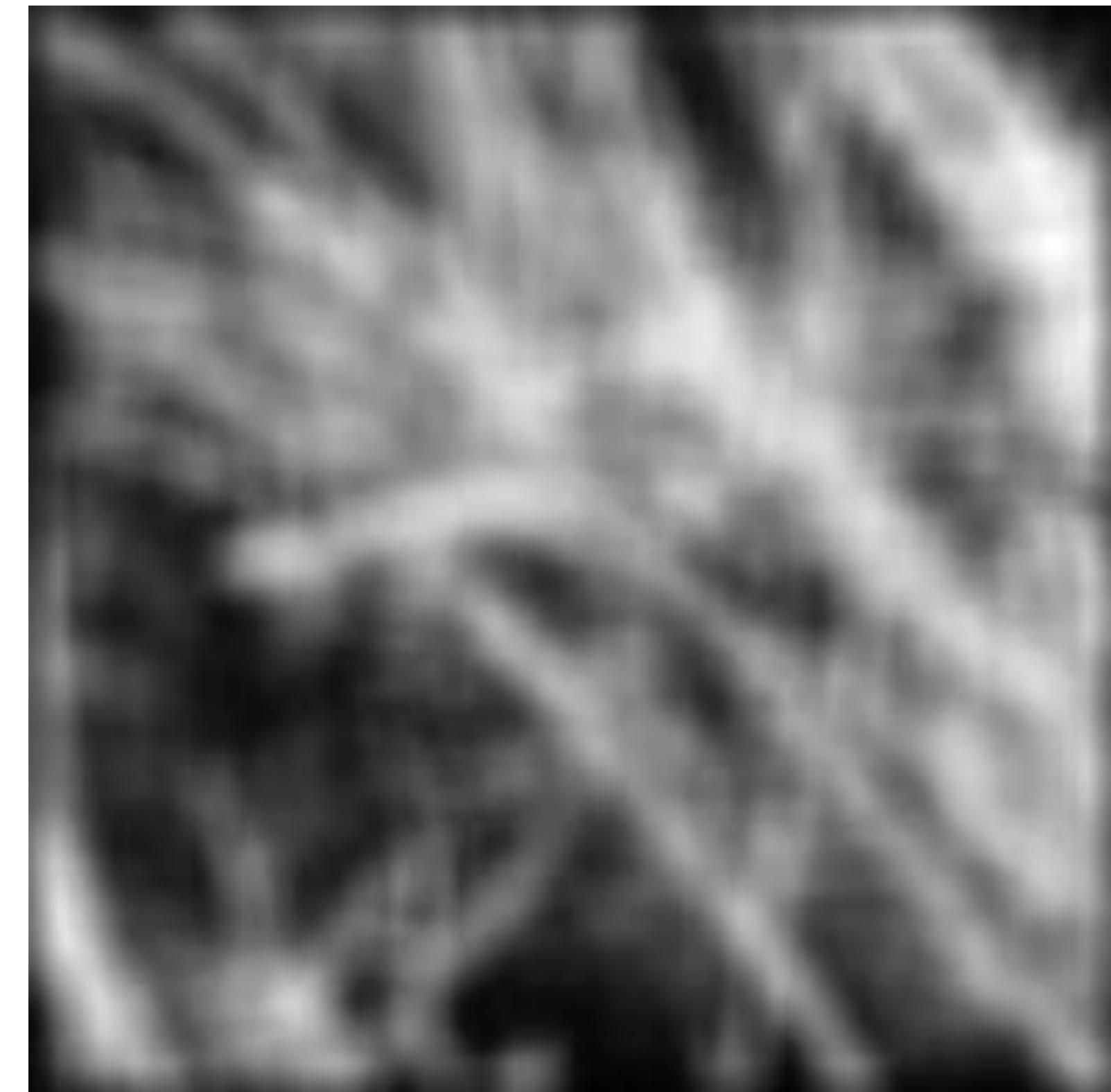
With $\sigma = 1$:

0.059	0.097	0.059
0.097	0.159	0.097
0.059	0.097	0.059

What happens if σ is larger?

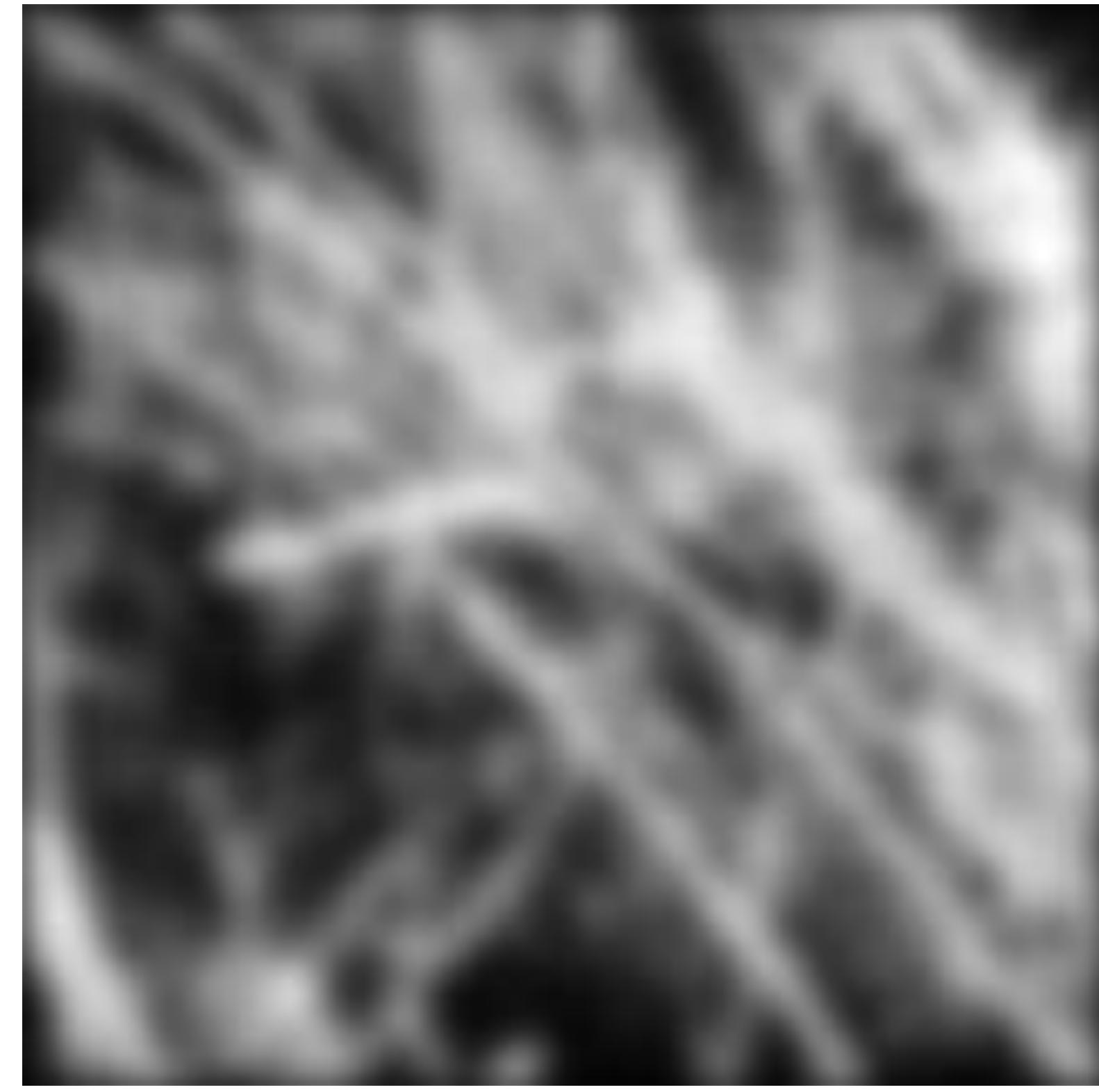
What happens if σ is smaller?

Smoothing with a **Box Filter**



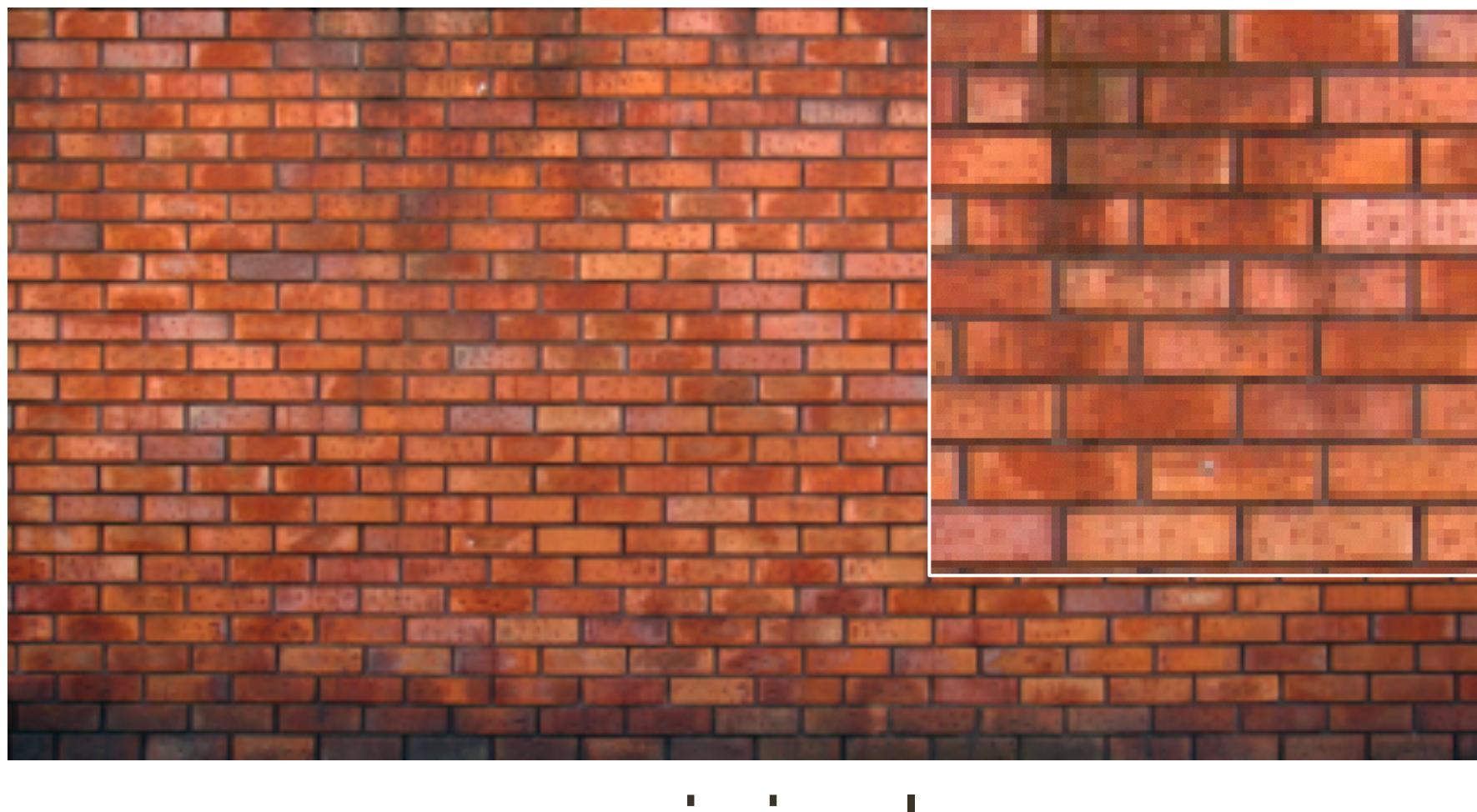
Forsyth & Ponce (2nd ed.) Figure 4.1 (left and middle)

Smoothing with a **Gaussian**

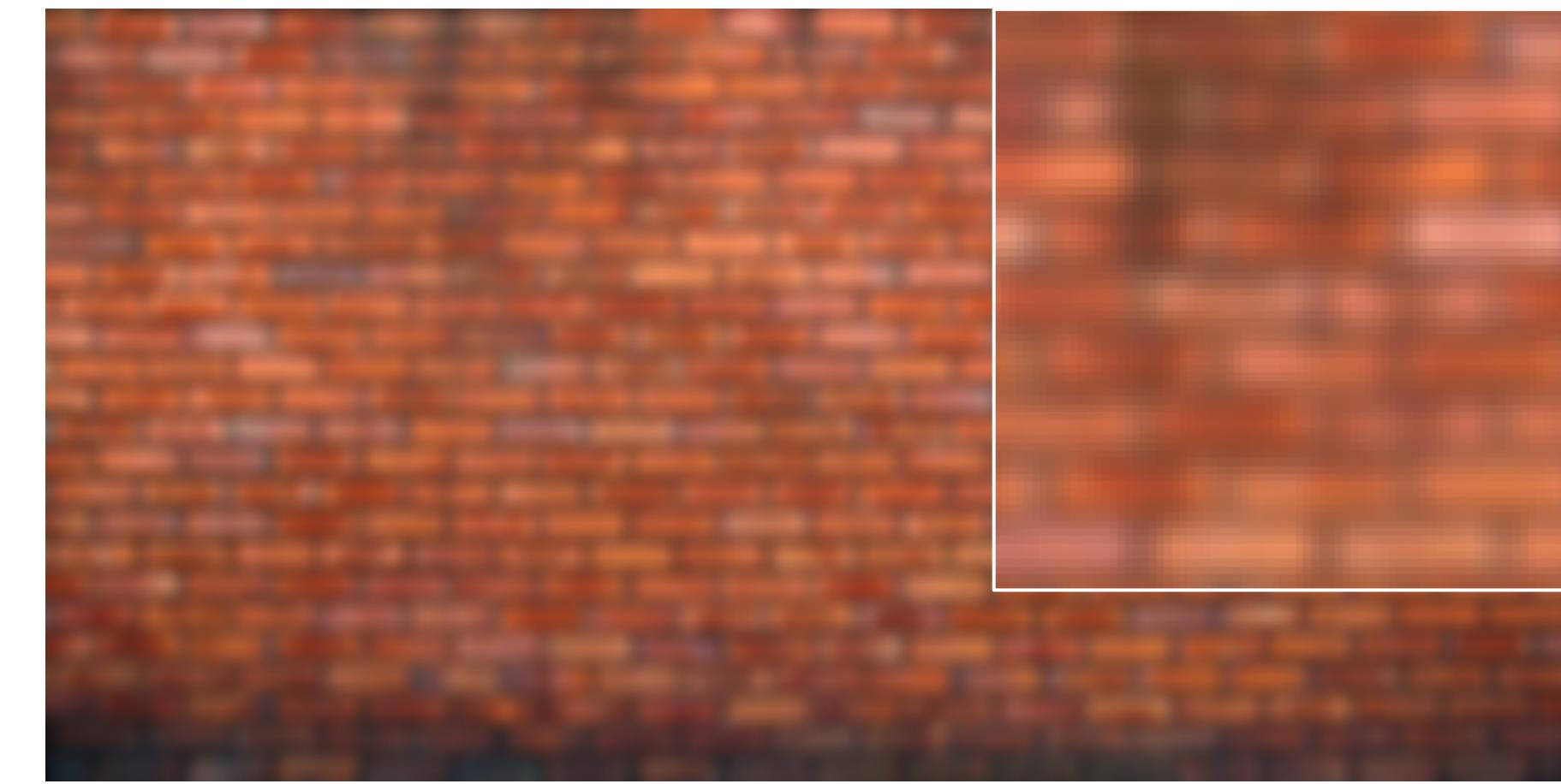


Forsyth & Ponce (2nd ed.) Figure 4.1 (left and right)

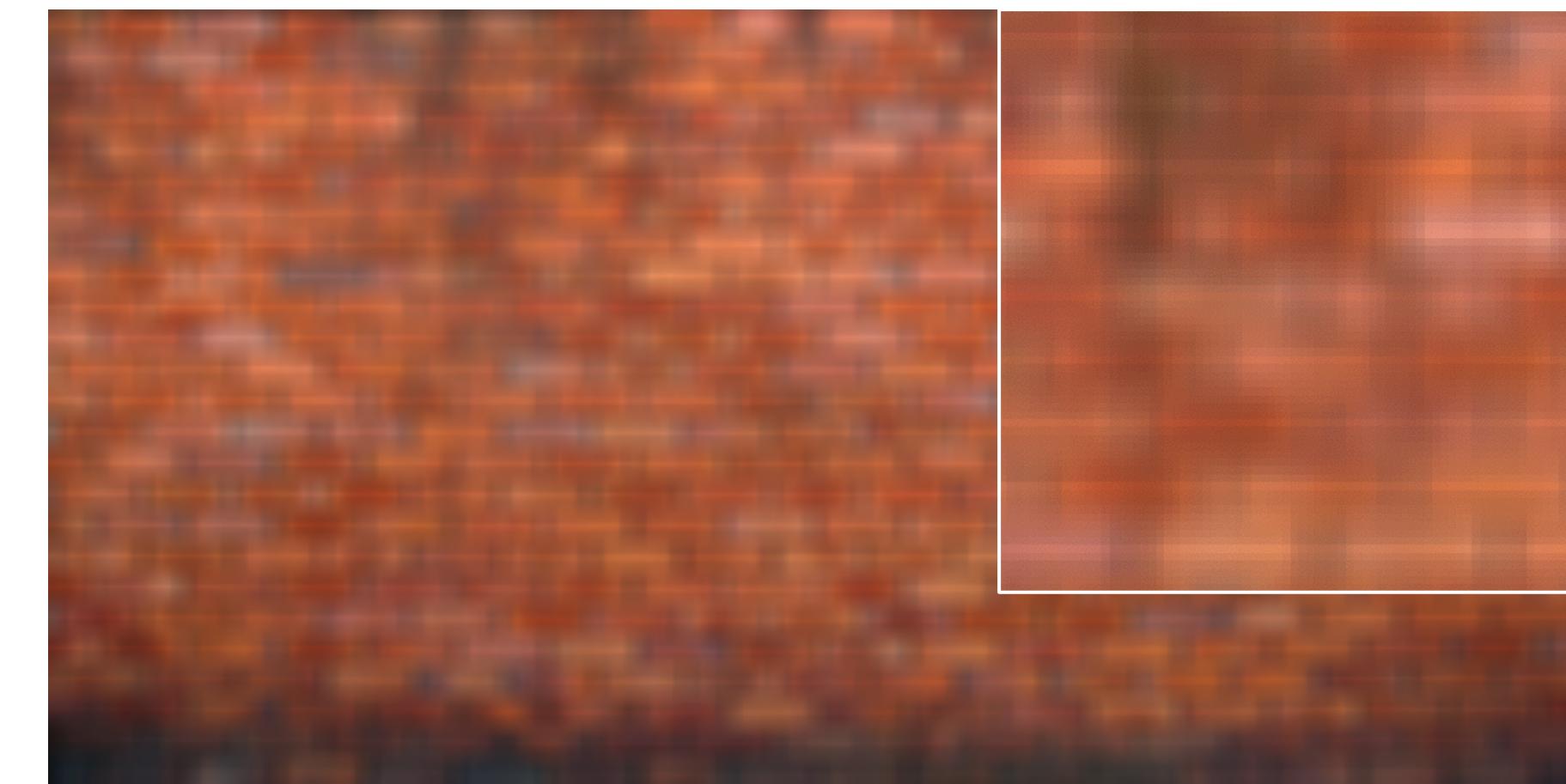
Box vs. Gaussian Filter



original



7x7 Gaussian



7x7 box

Fun: How to get shadow effect?

University of
British
Columbia

Fun: How to get shadow effect?

University of
British
Columbia

Blur with a Gaussian kernel, then compose the blurred image with the original
(with some offset)

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 is the problem with this filter?



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

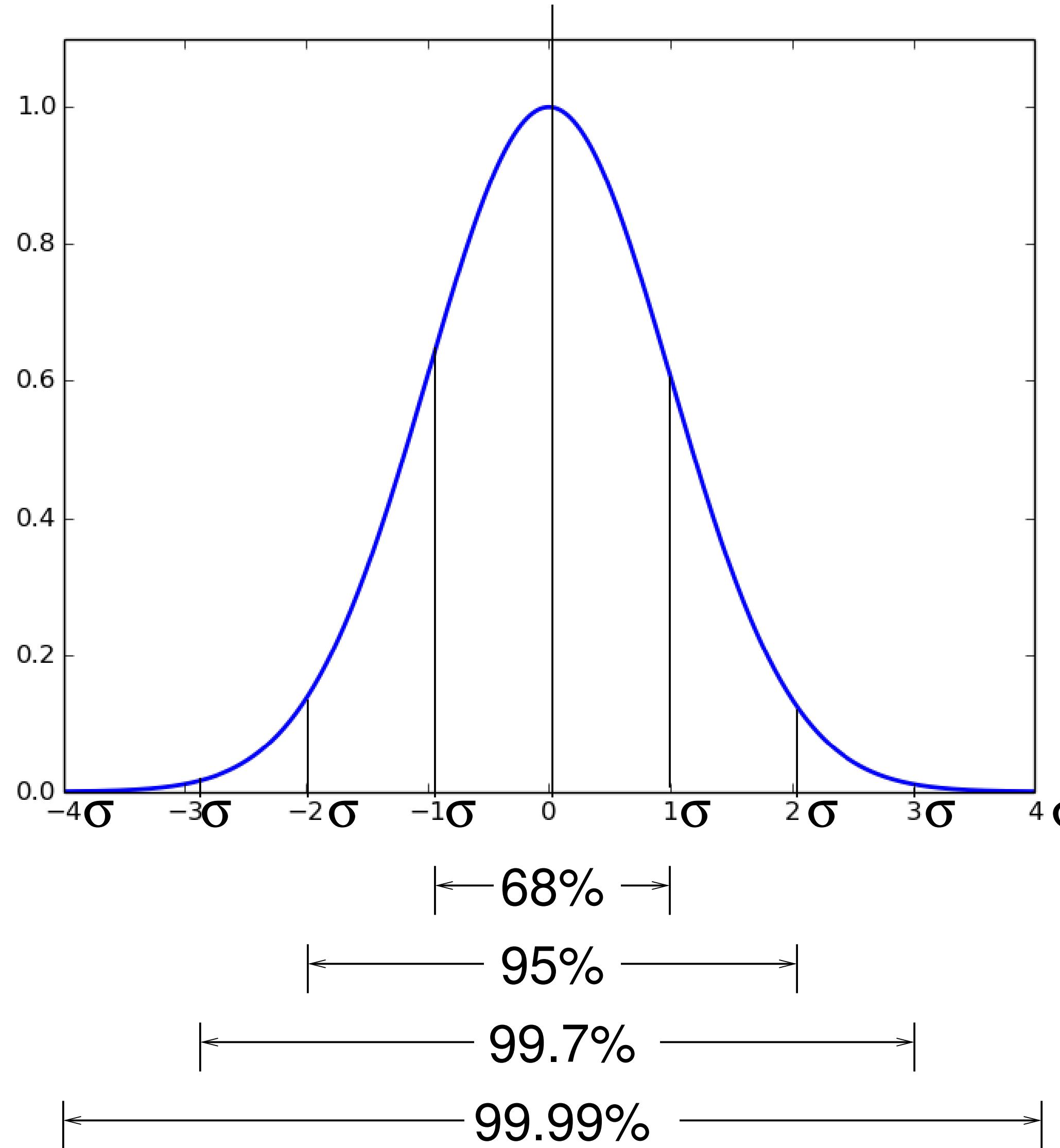
What is the problem with this filter?



does not sum to 1

truncated too much

Gaussian: Area Under the Curve



Example 6: Smoothing with a Gaussian

With $\sigma = 1$:

0.059	0.097	0.059
0.097	0.159	0.097
0.059	0.097	0.059

Better version of the Gaussian filter:

- sums to 1 (normalized)
- captures $\pm 2\sigma$

$$\frac{1}{273}$$

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

A good guideline for the Gaussian filter is to capture $\pm 3\sigma$, for $\sigma = 1 \Rightarrow 7 \times 7$ filter

Smoothing Summary

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

- Smoothing with a box filter depends on direction
- Point spread function is a box

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 (avg of many independent rvs → normal dist)

Lets talk about **efficiency**

Efficient Implementation: **Separability**

A 2D function of x and y is **separable** if it can be written as the product of two functions, one a function only of x and the other a function only of y

Both the **2D box filter** and the **2D Gaussian filter** are **separable**

Both can be implemented as two 1D convolutions:

- First, convolve each row with a 1D filter
- Then, convolve each column with a 1D filter
- Aside: or vice versa

The **2D Gaussian** is the only (non trivial) 2D function that is both separable and rotationally invariant.

Separability: Box Filter Example

Standard (3x3)

$$F(X, Y) = F(X)F(Y)$$

filter

1	1	1
1	1	1
1	1	1

Separability: Box Filter Example

Standard (3x3)

$$F(X, Y) = F(X)F(Y)$$

filter

1	1	1
1	1	1
1	1	1

$$\text{image} \quad I(X, Y)$$

Separable

$$F(X)$$

filter

$\frac{1}{3}$	1	1	1
---------------	---	---	---

Separability: Box Filter Example

Standard (3x3)

$$I(X, Y)$$

Separable

$$F(X)$$

filter

1	1
---	---

$$F(X, Y) = F(X)F(Y)$$

filter

1	1	1
1	1	1
1	1	1

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	30	60	90	90	90	60	30
0	30	60	90	90	90	60	30
0	30	30	60	60	90	60	30
0	30	60	90	90	90	60	30
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
30	30	30	30	0	0	0	0
0	0	0	0	0	0	0	0

$F(Y)$
filter

filter

1
1
1

$$\text{output} \quad I'(X, Y)$$

Separability: Proof

Convolution with $F(X, Y) = F(X)F(Y)$ can be performed as 2×1 D convolutions



4.2

Efficient Implementation: Separability

For example, recall the 2D **Gaussian**:

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

function of x function of y

The 2D Gaussian can be expressed as a product of two functions, one a function of x and another a function of y

In this case the two functions are (identical) 1D Gaussians

Gaussian Blur

- 2D Gaussian filter can be thought of as an **outer product** or **convolution** of row and column filters

$$\begin{matrix} \text{Row Filter} \\ \times \\ \text{Column Filter} \end{matrix} = \text{2D Filter}$$

The diagram illustrates the 2D Gaussian blur filter as an outer product of row and column filters. It shows a vertical column of grayscale blocks representing the Row Filter, followed by a horizontal row of grayscale blocks representing the Column Filter. An asterisk (*) indicates the multiplication (outer product) between the two filters. An equals sign (=) indicates the resulting 2D filter, which is a blurred version of the original input image.

Example: Separable Gaussian Filter

$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \otimes \frac{1}{16} \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Example: Separable Gaussian Filter

$$\frac{1}{16} \otimes \frac{1}{16} = \frac{1}{256}$$

The diagram illustrates the convolution of a 5x5 input matrix with a 1x5 filter. The input matrix is a 5x5 identity matrix. The filter is a 1x5 vector [1, 4, 6, 4, 1]. The result is a 5x5 output matrix where each element is the dot product of the corresponding row of the input and the filter, scaled by 1/256.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	4	6	4	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1
4
6
4
1

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

Efficient Implementation: Separability



4.3

Efficient Implementation: Separability

Naive implementation of 2D **Gaussian**:

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

Separable 2D **Gaussian**:

At each pixel, (X, Y) , there are $2m$ multiplications

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

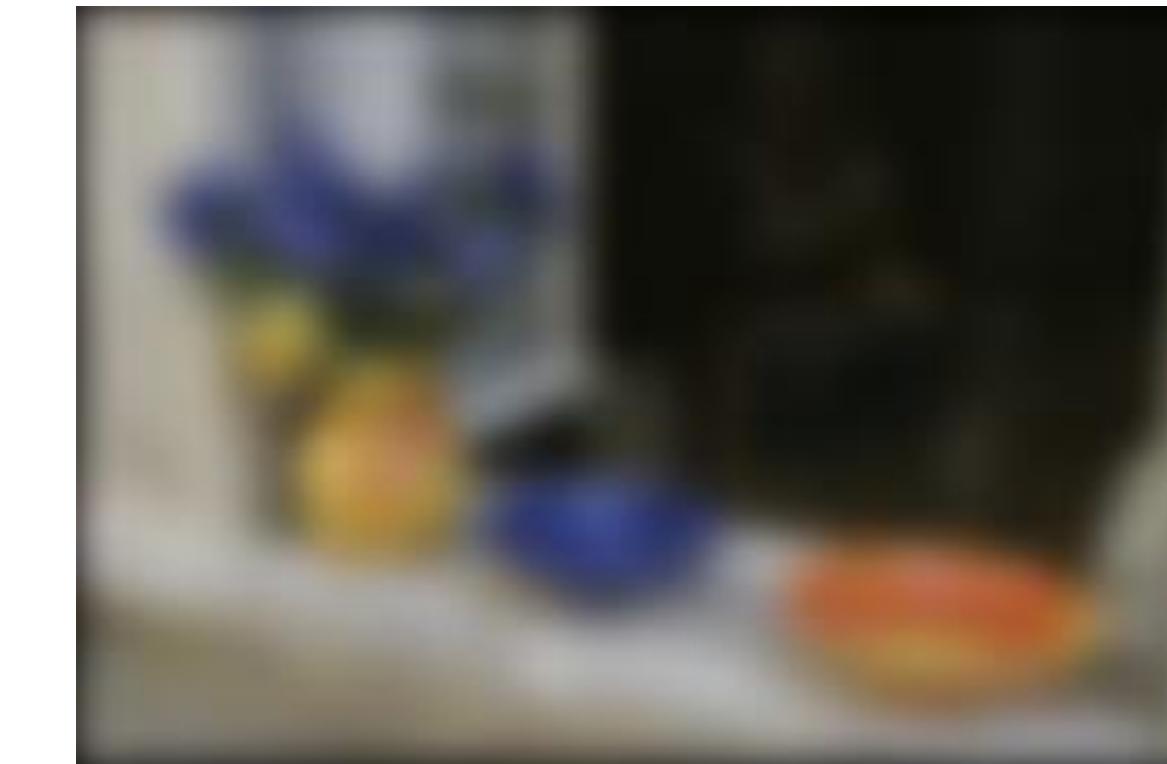
Total: $2m \times n^2$ multiplications

Separable Filtering

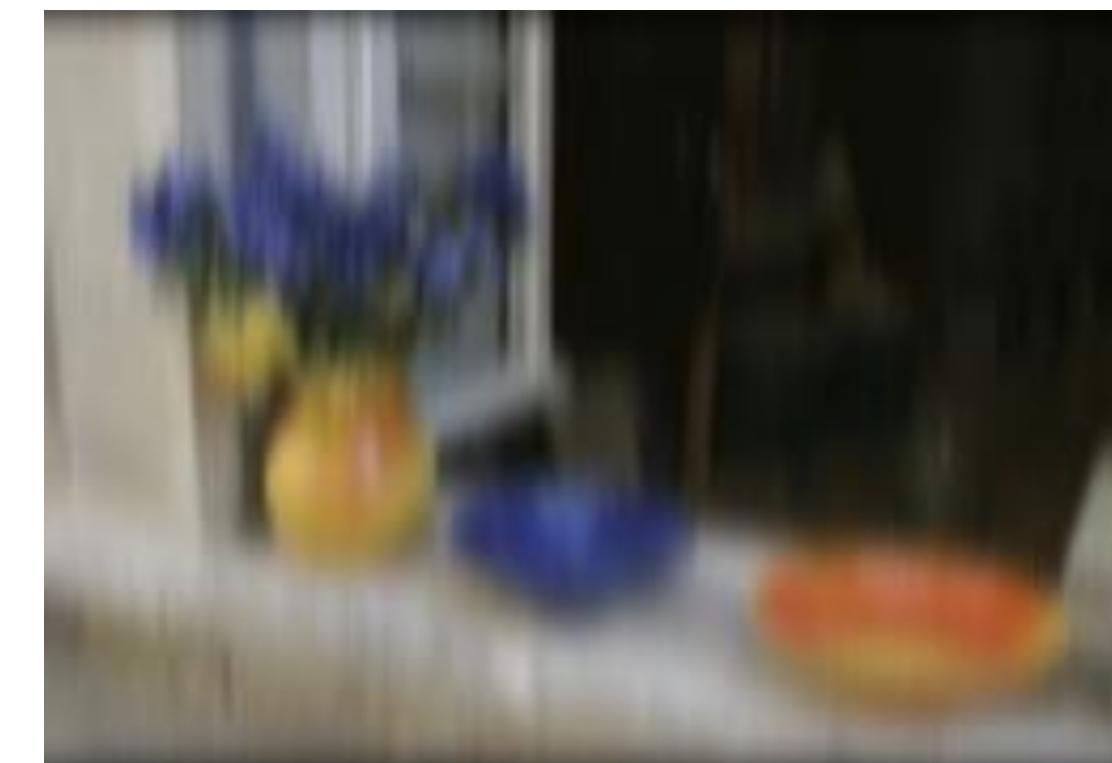
- 2D Gaussian blur by horizontal/vertical blur



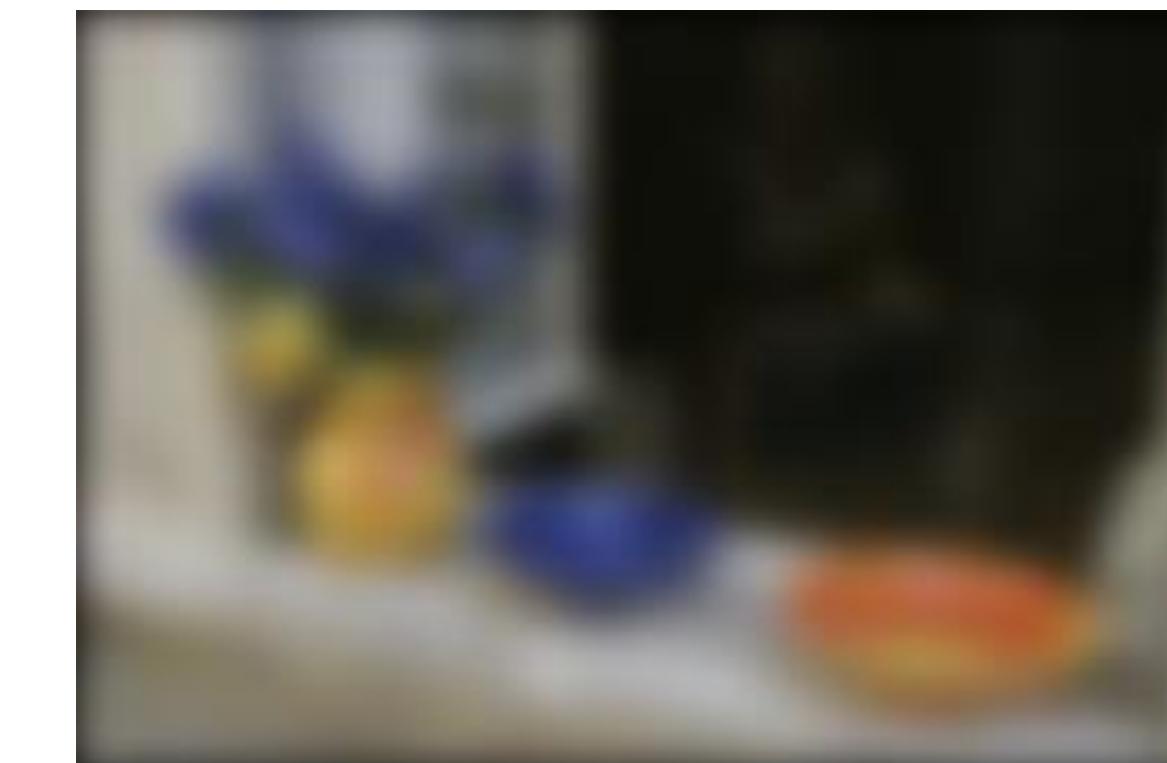
horizontal



vertical



vertical



horizontal

Separable Filtering

- Several useful filters can be applied as independent row and column operations

$$\frac{1}{K^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & 1 & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

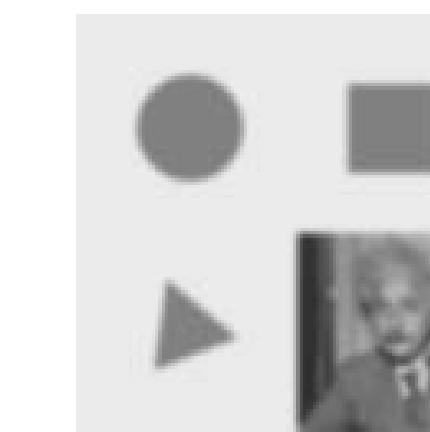
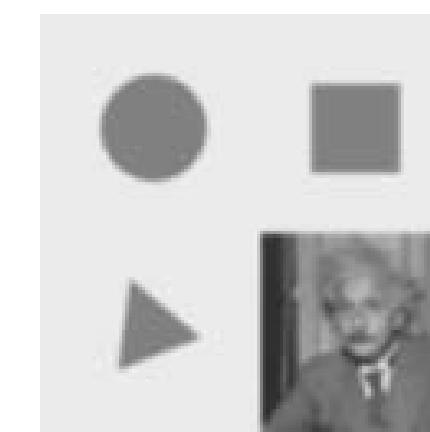
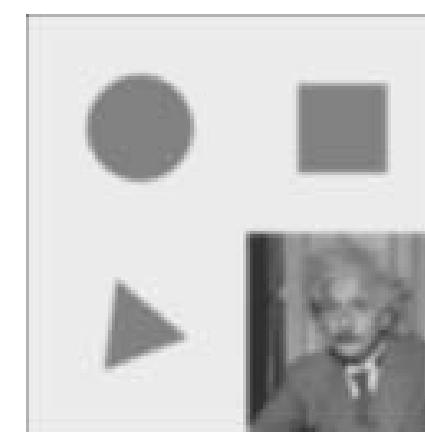
$$\frac{1}{K} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

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

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



(a) box, $K = 5$

(b) bilinear

(c) “Gaussian”

(d) Sobel

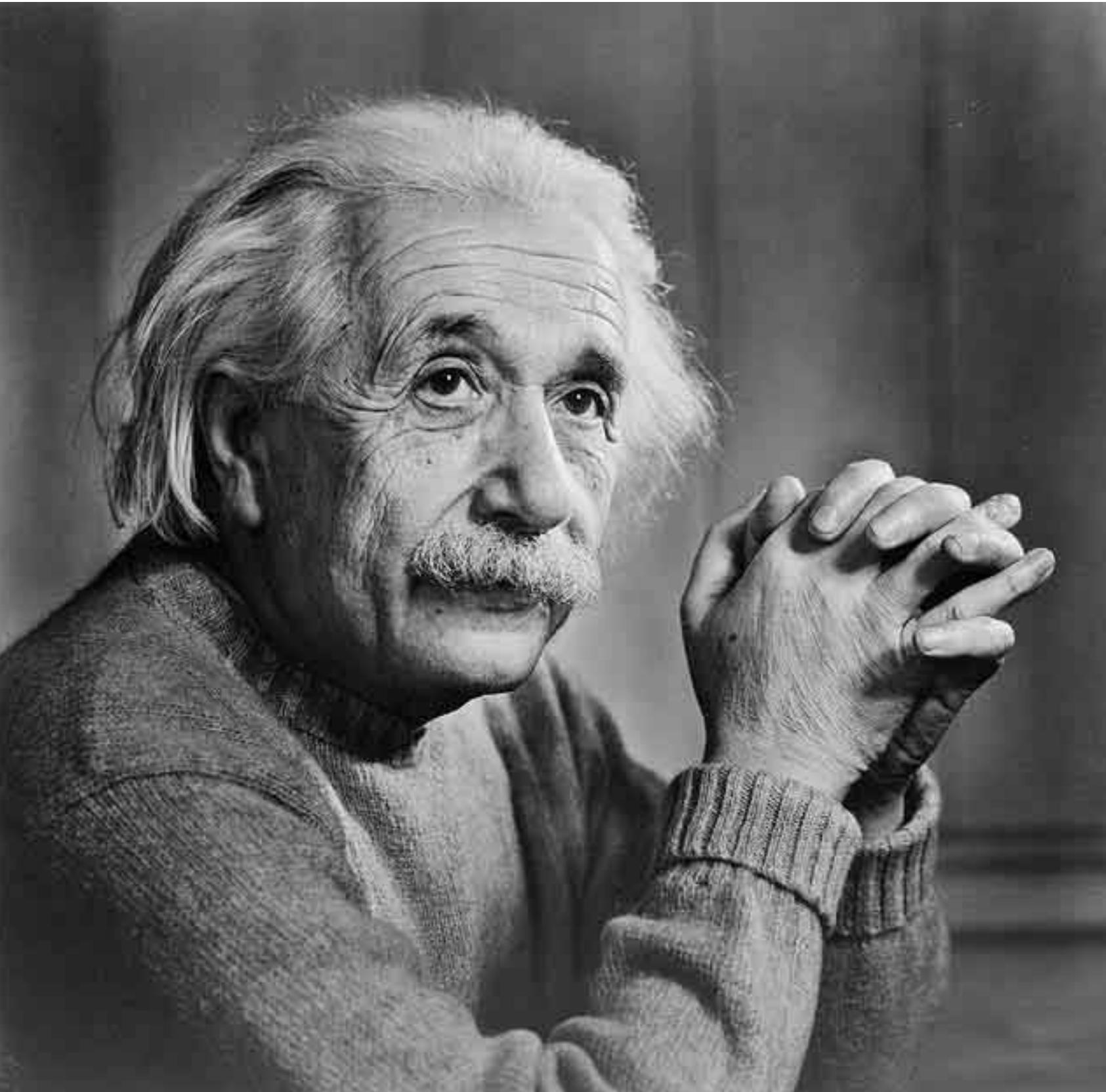
(e) corner

Example 7: Smoothing with a Pillbox

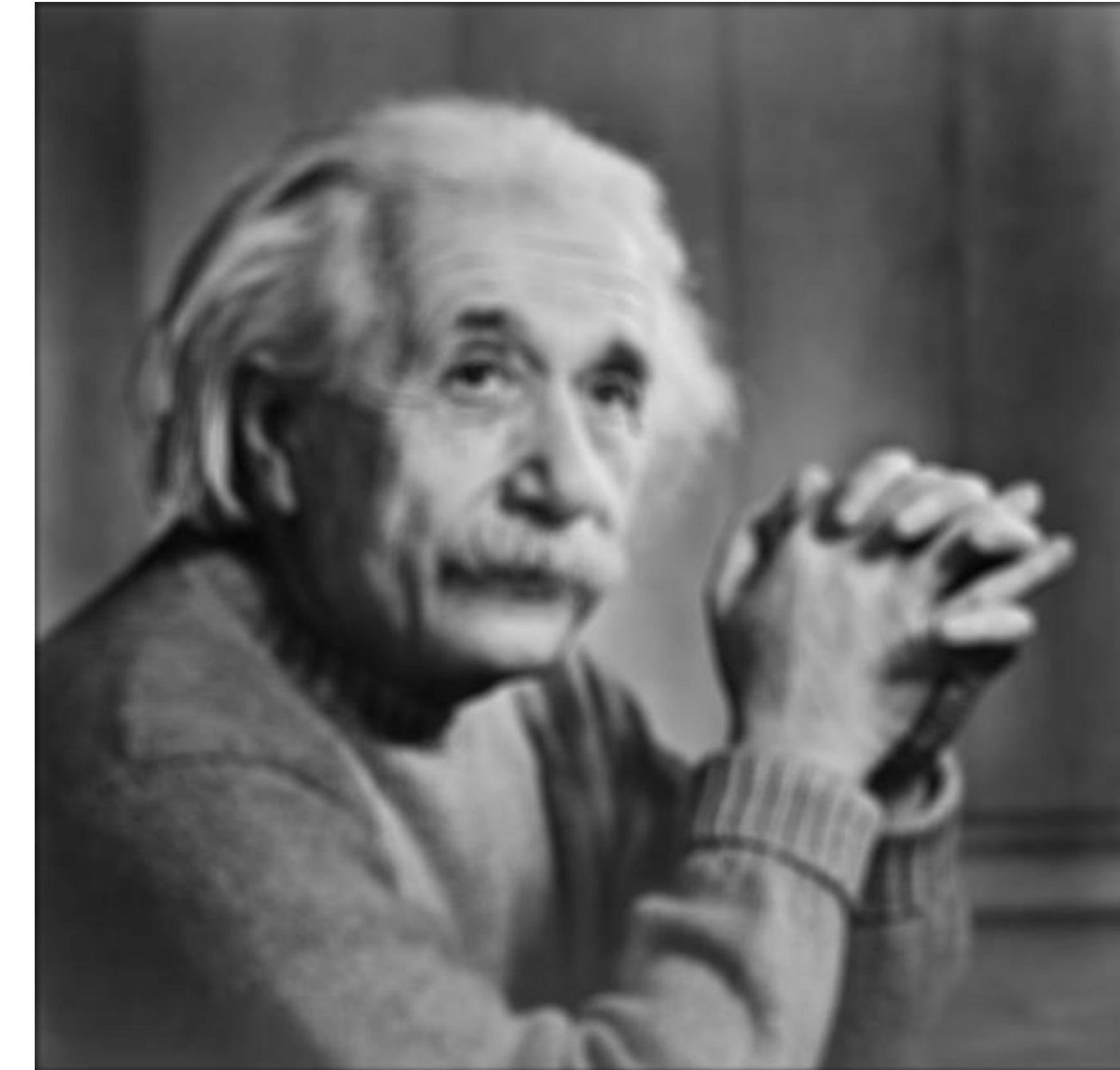
The 2D Gaussian is the only (non trivial) 2D function that is both **separable** and **rotationally invariant**.

A **2D pillbox** is rotationally invariant but **not separable**.

Example 7: Smoothing with a Pillbox



Original



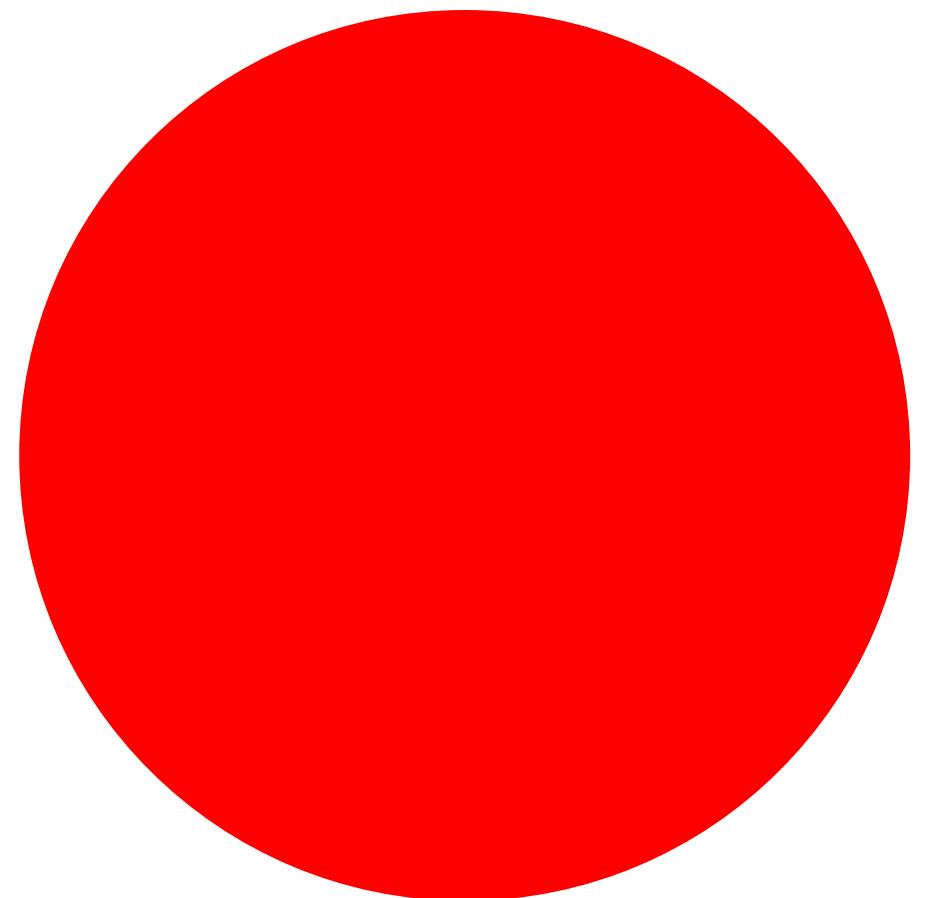
11 x 11 Pillbox

Low-pass Filtering = “Smoothing”

Box Filter

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Pillbox Filter



Gaussian Filter

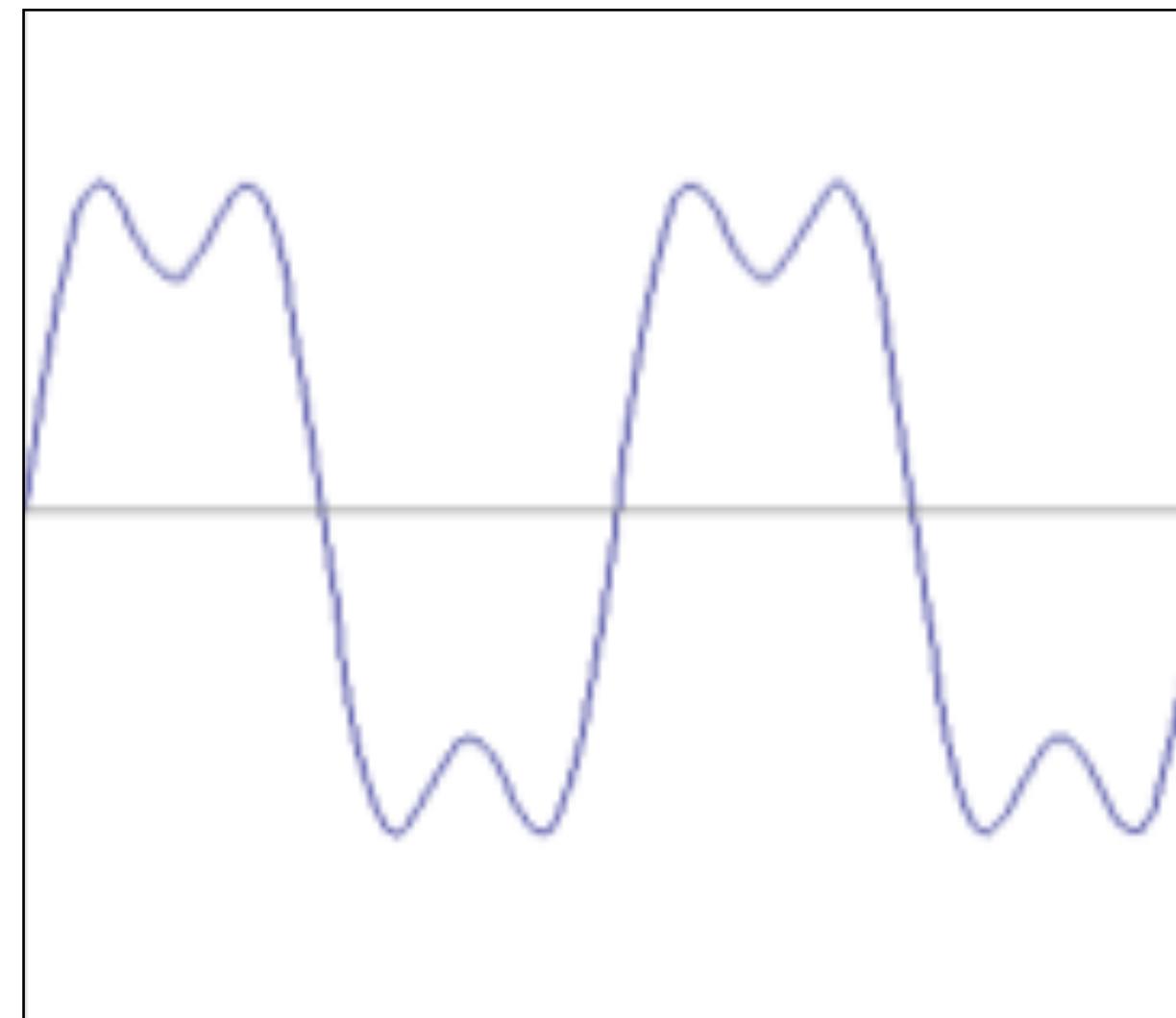
$$\frac{1}{256} \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

All of these filters are **Low-pass Filters**

Low-pass filter: Low pass filter filters out all of the high frequency content of the image, only low frequencies remain

Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?



=

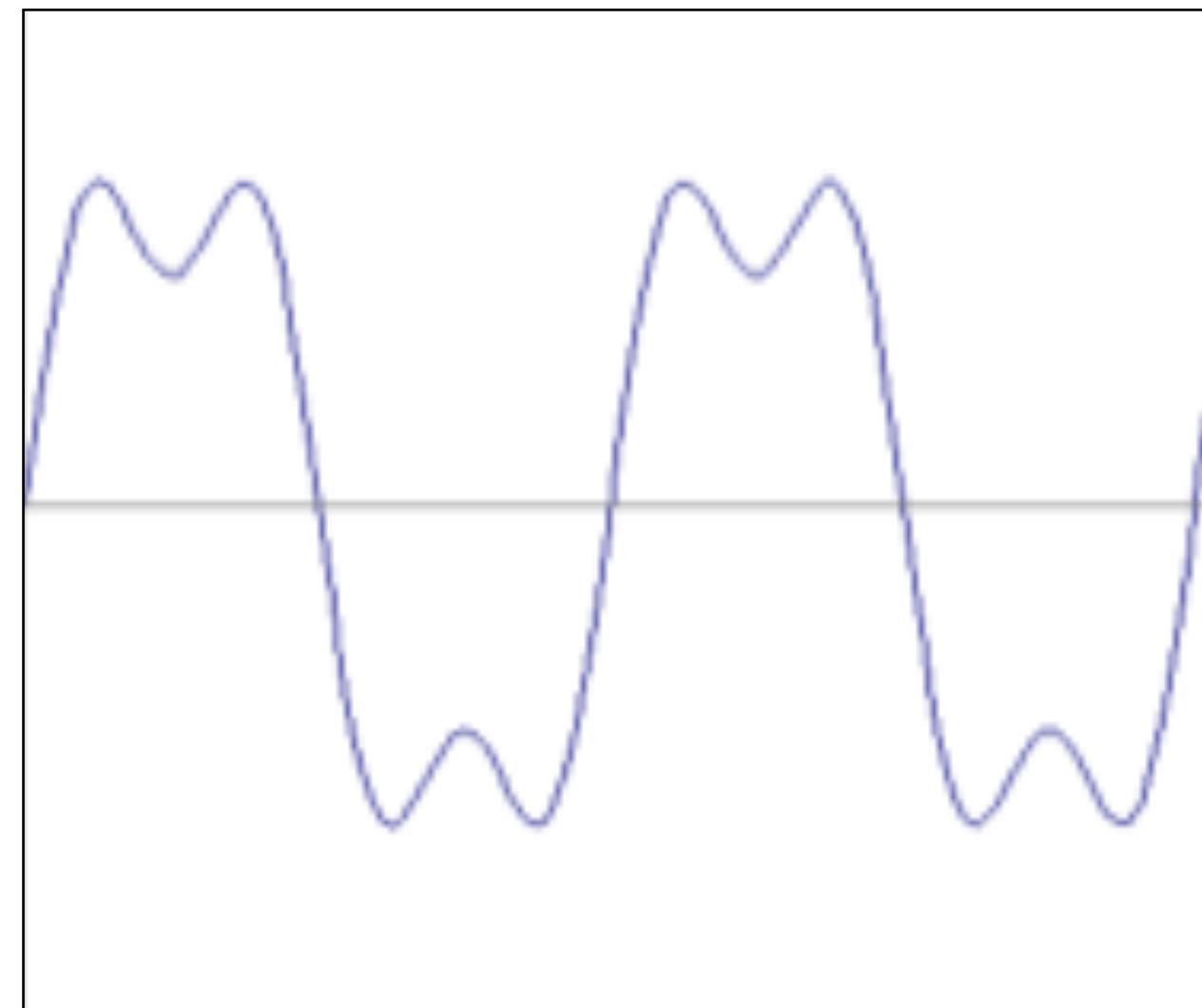
?

+

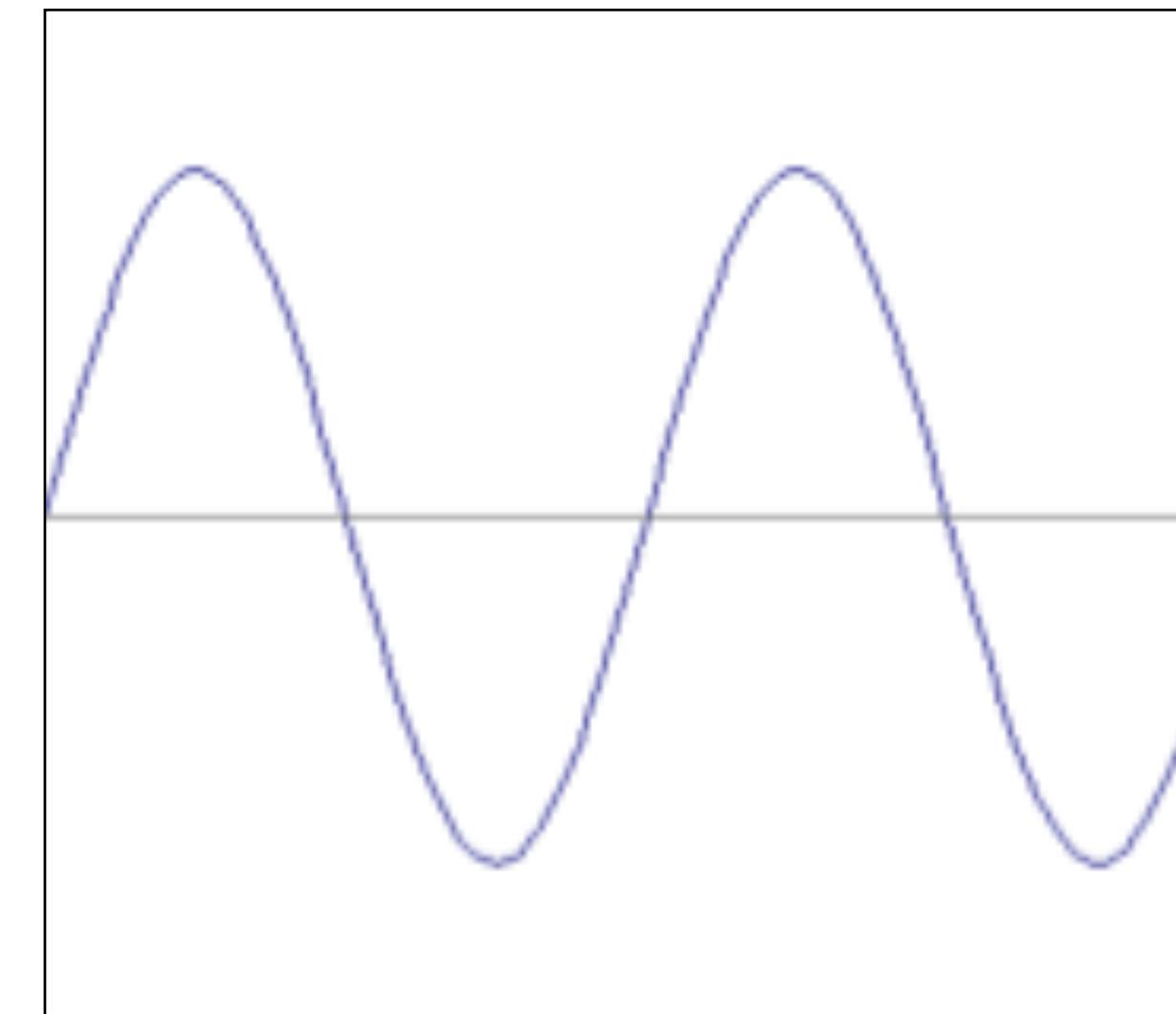
?

Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?



=



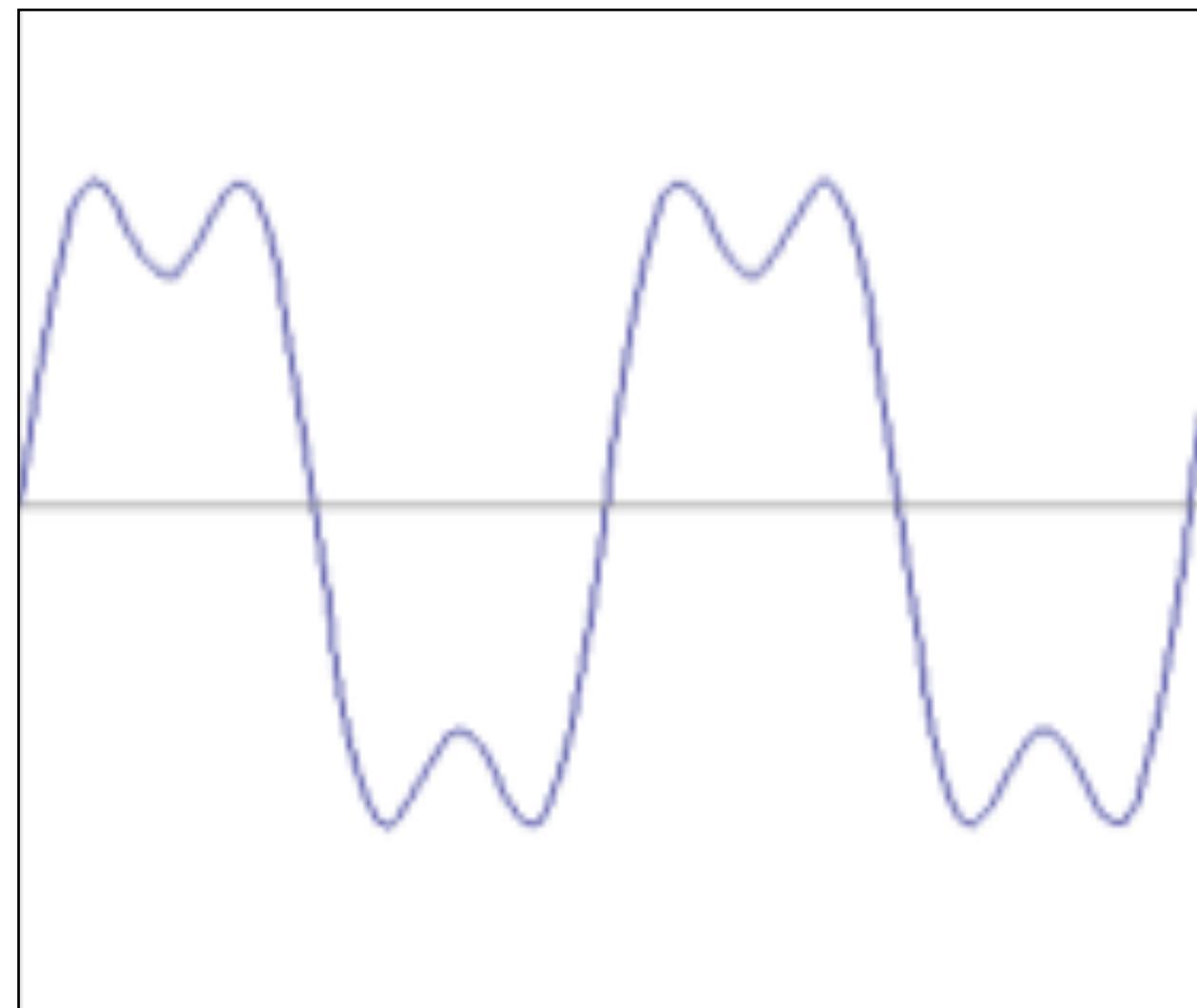
+

?

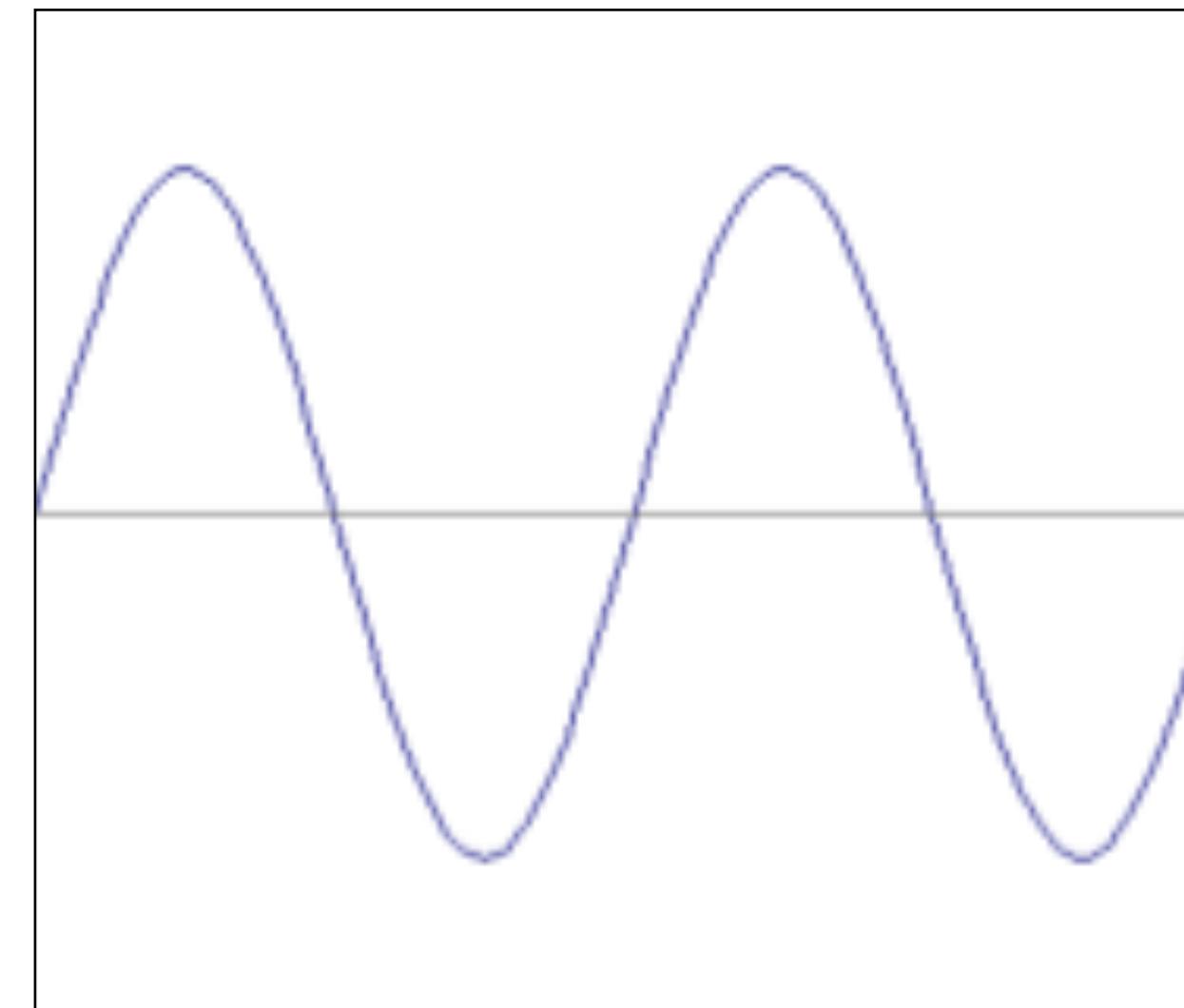
$$\sin(2\pi x)$$

Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?

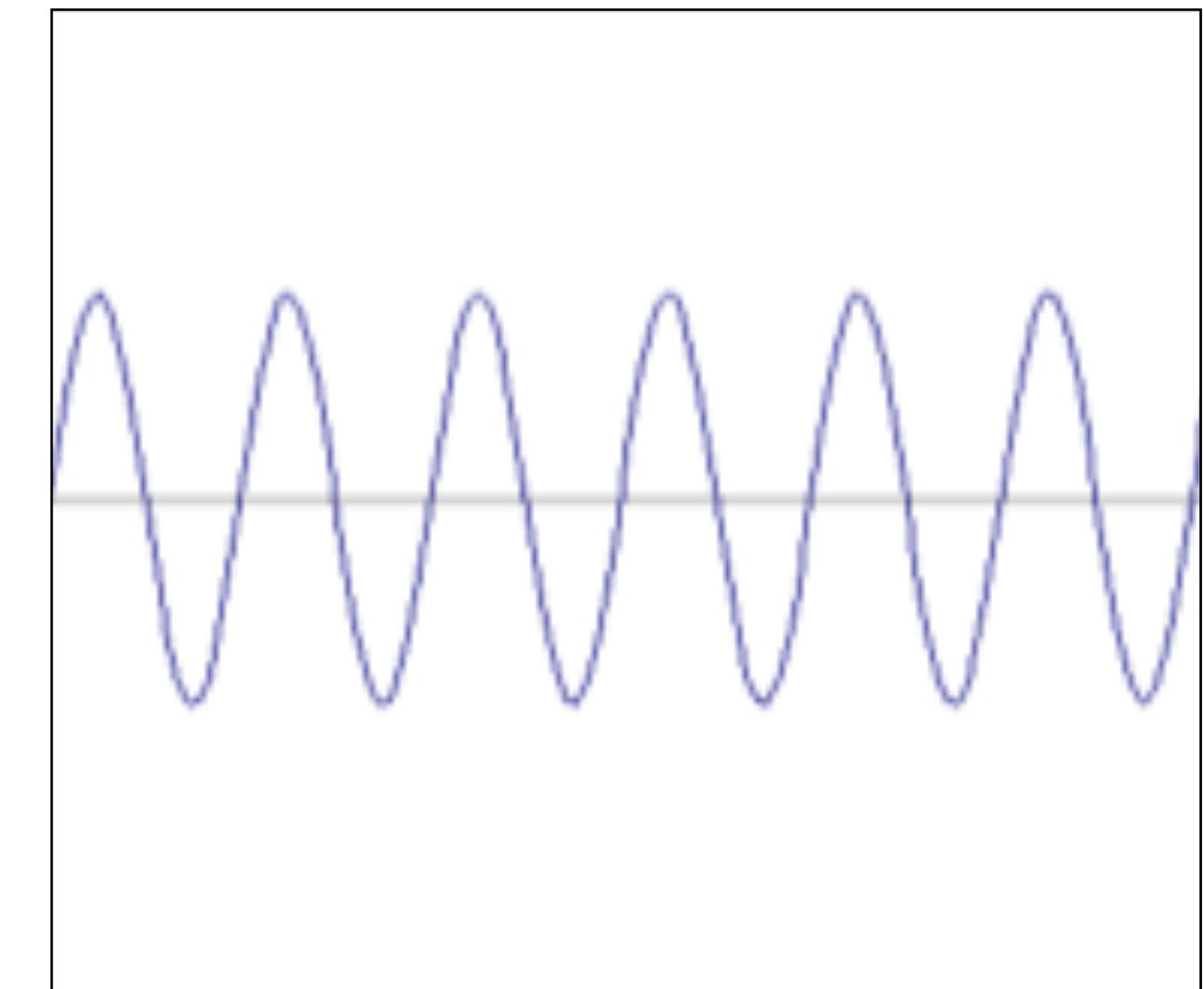


=



$$\sin(2\pi x)$$

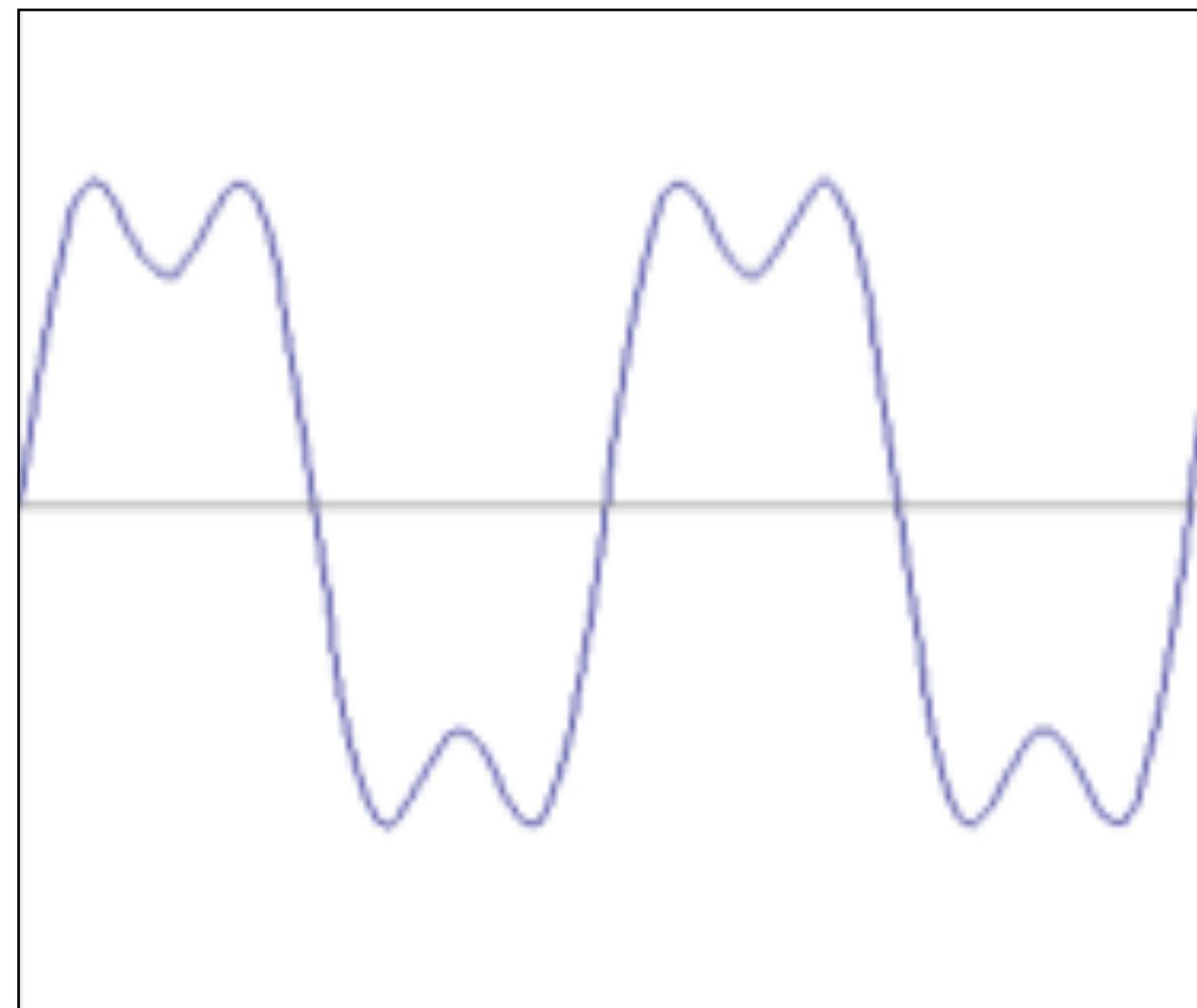
+



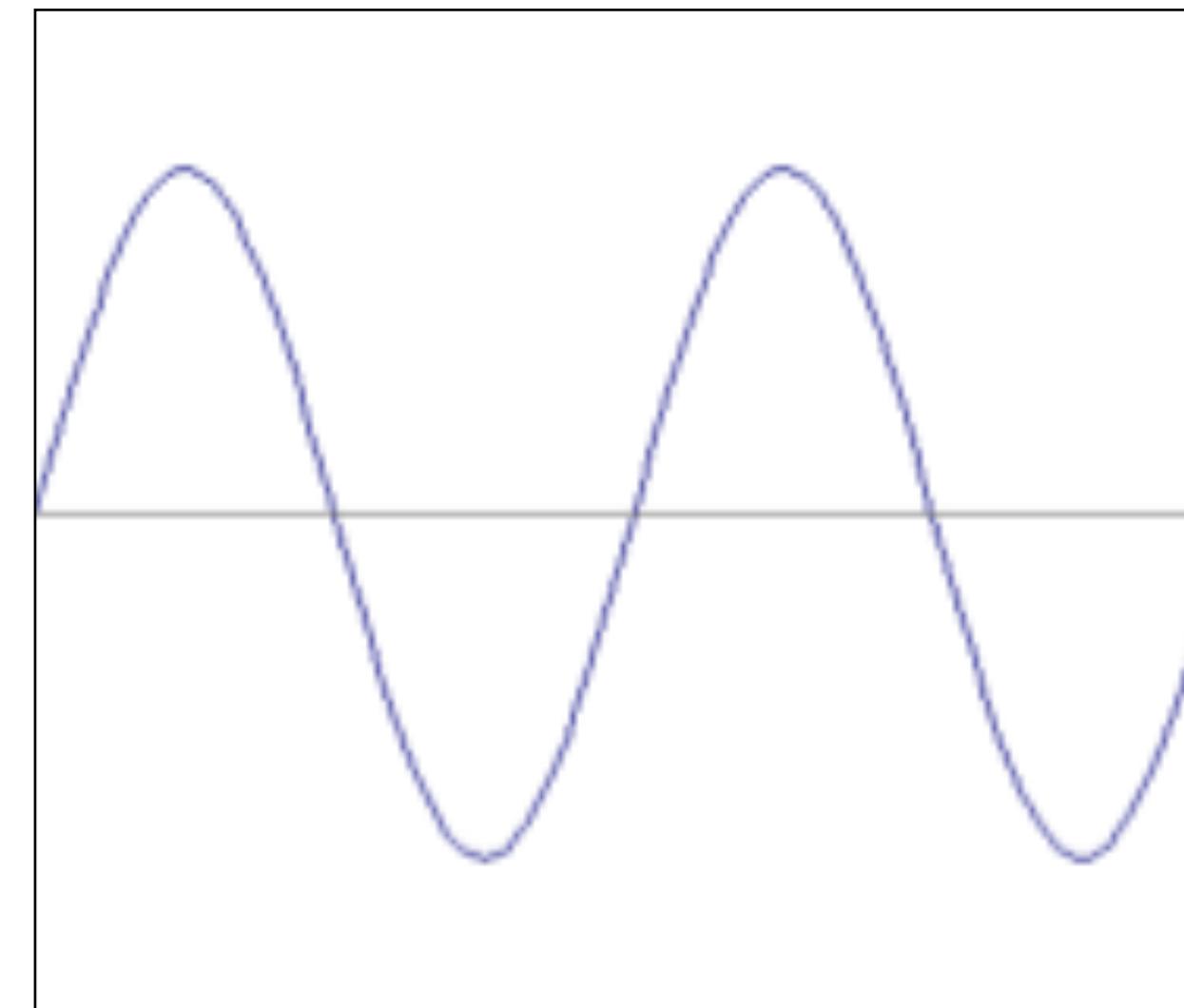
$$\frac{1}{3} \sin(2\pi 3x)$$

Fourier Transform (you will **NOT** be tested on this)

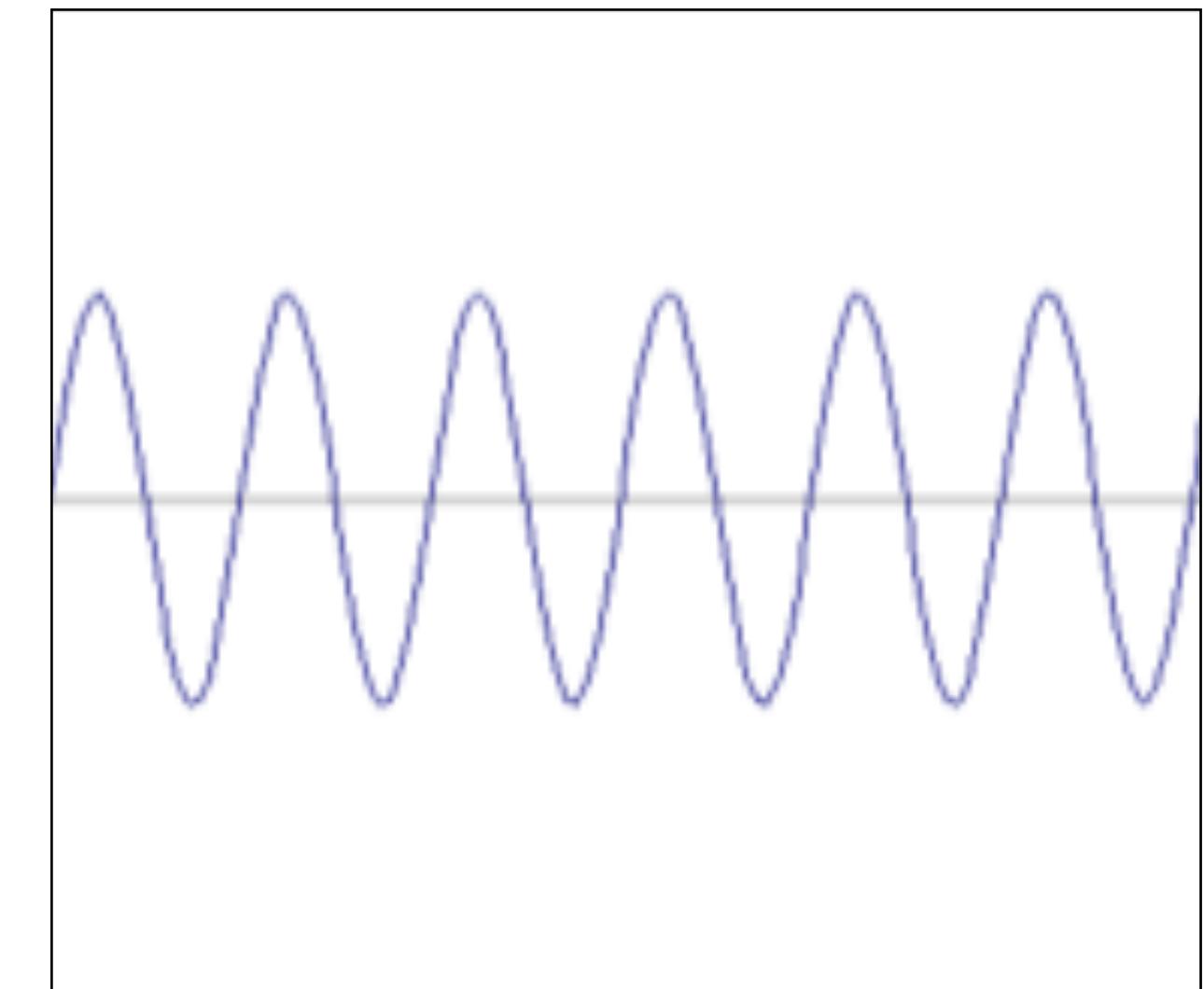
How would you generate this function?



=



+



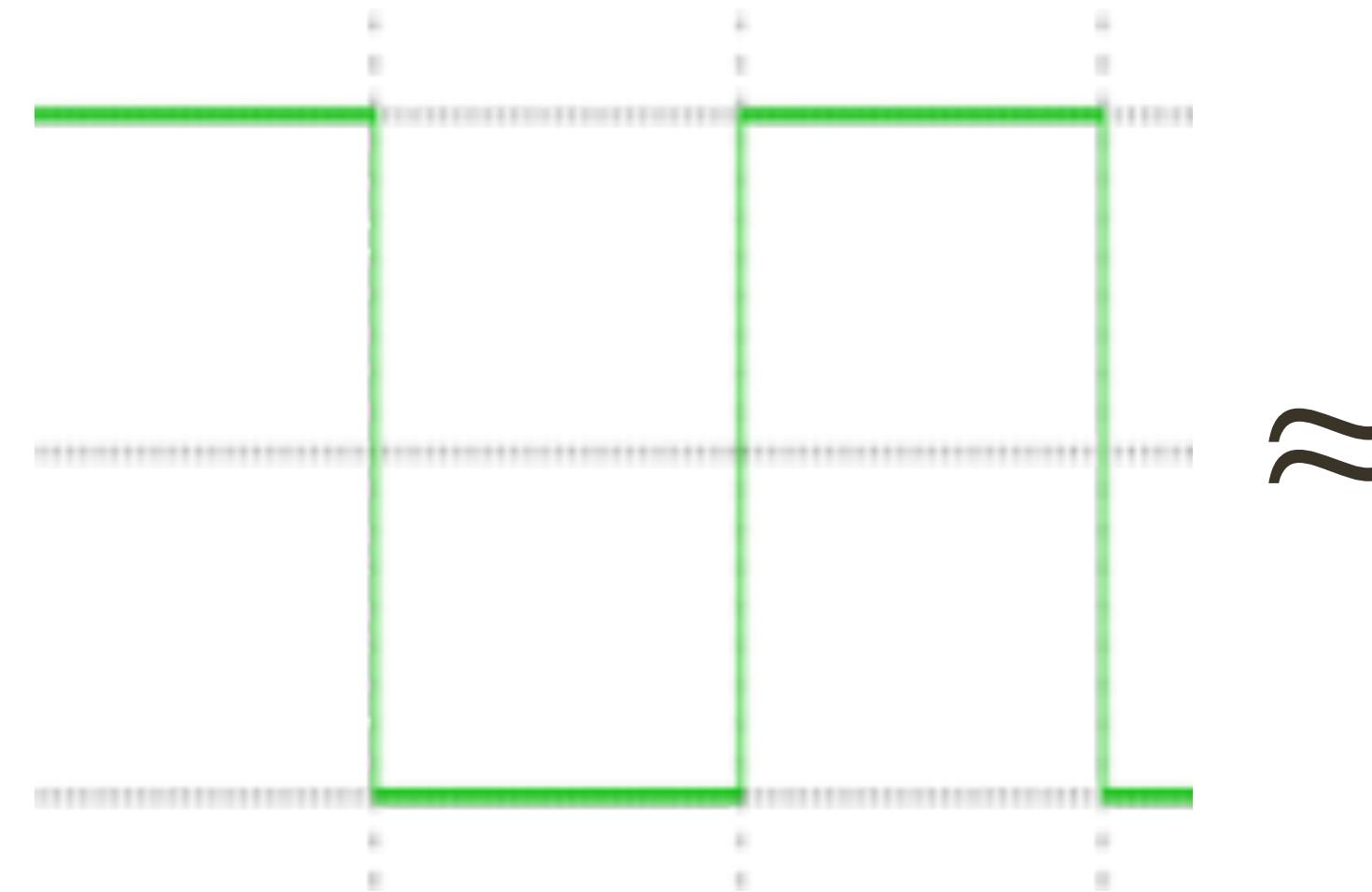
$$f(x) = \sin(2\pi x) + \frac{1}{3} \sin(2\pi 3x)$$

$$\sin(2\pi x)$$

$$\frac{1}{3} \sin(2\pi 3x)$$

Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?



\approx

square wave

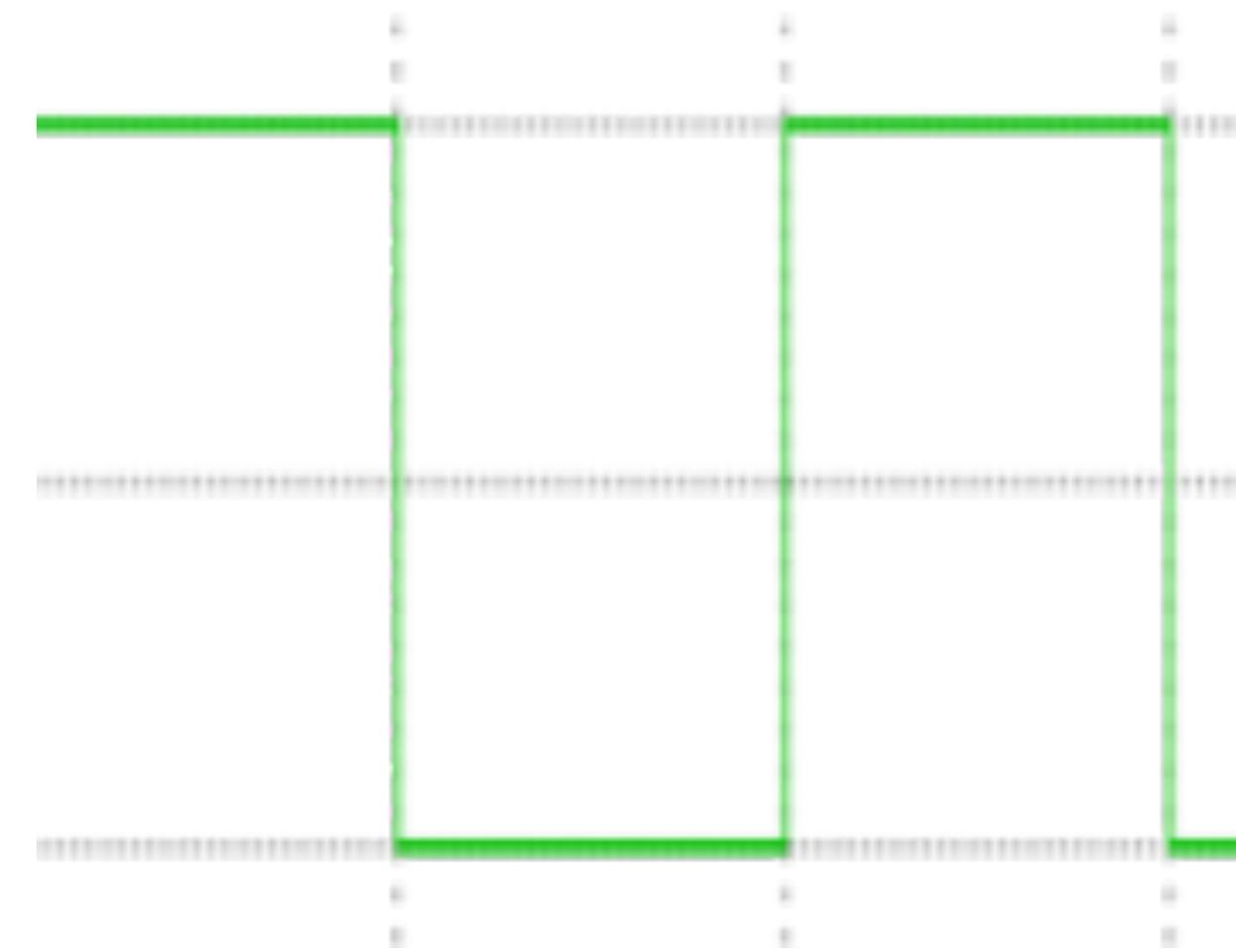
?

+

?

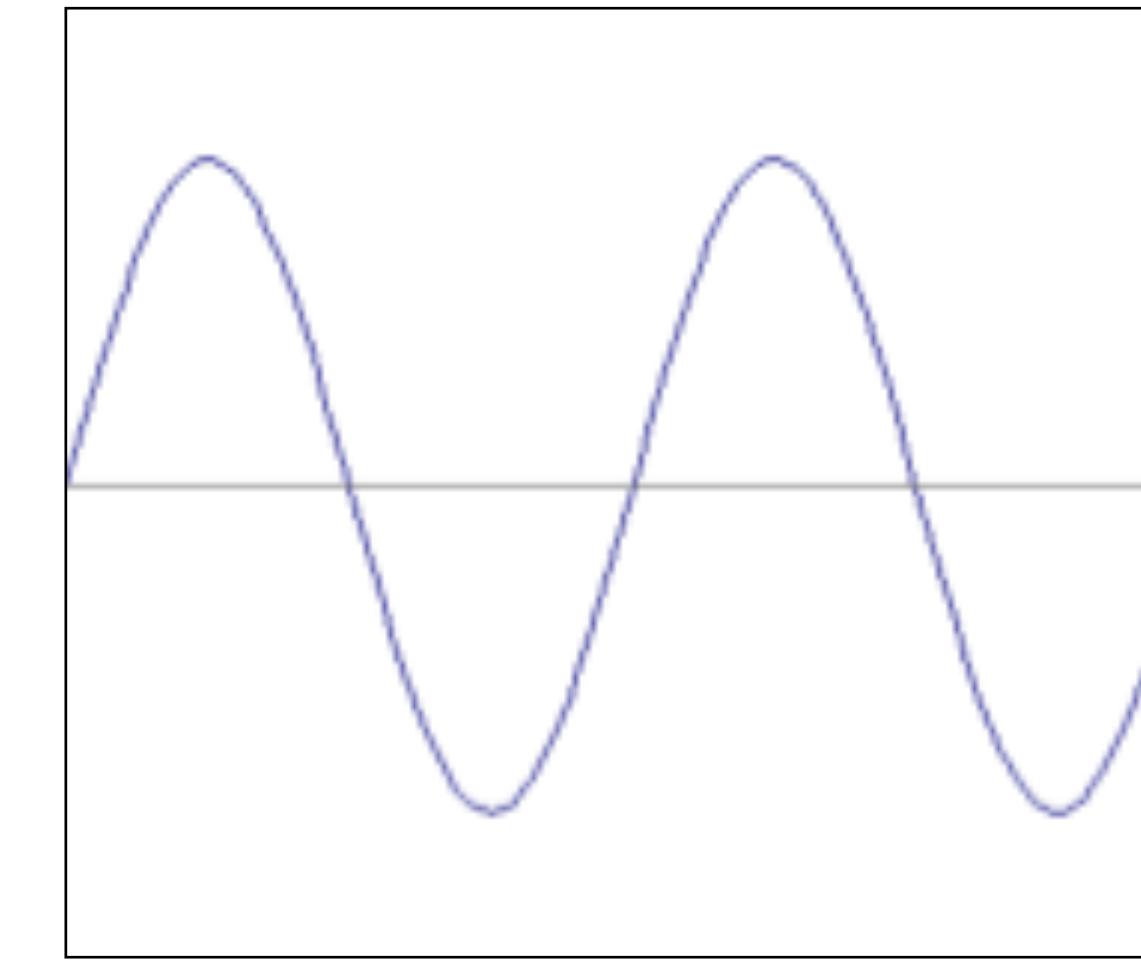
Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?

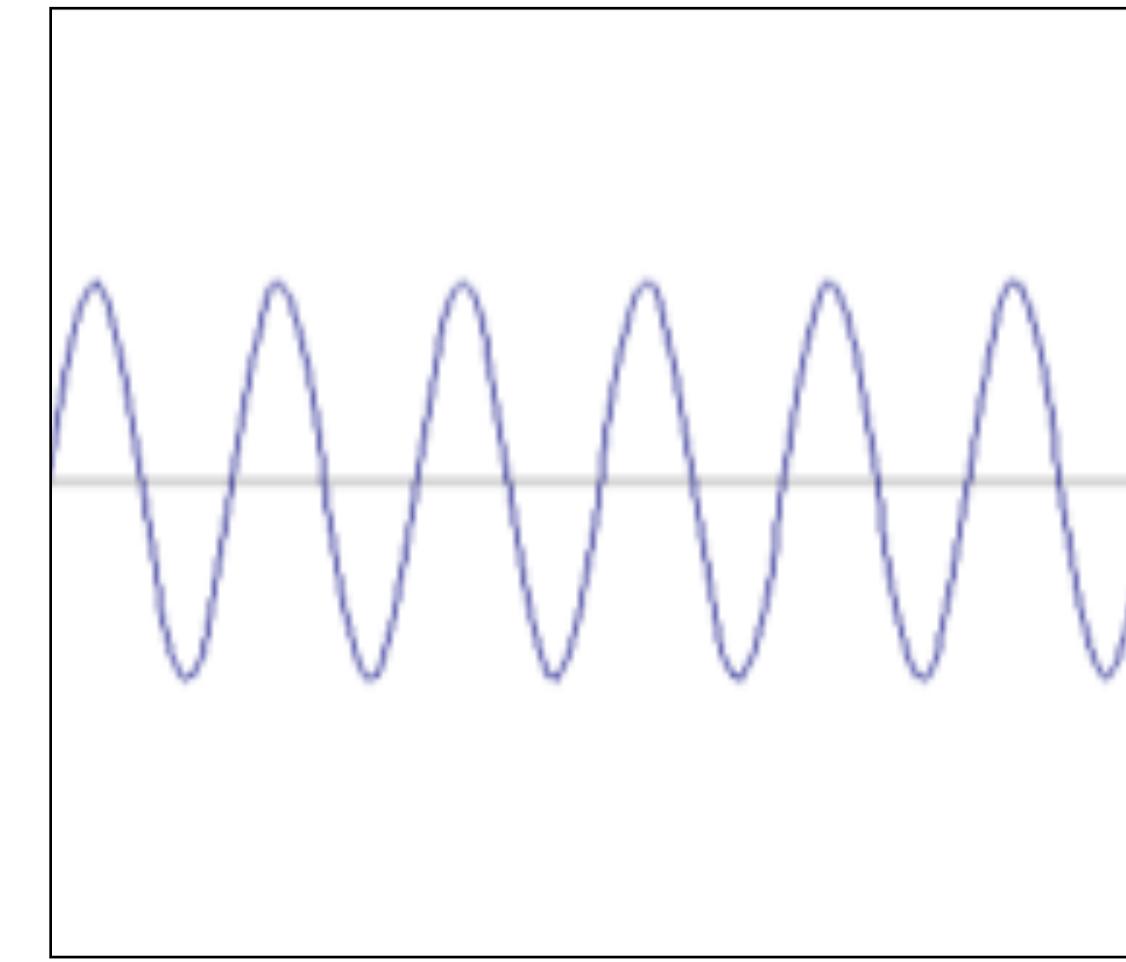


square wave

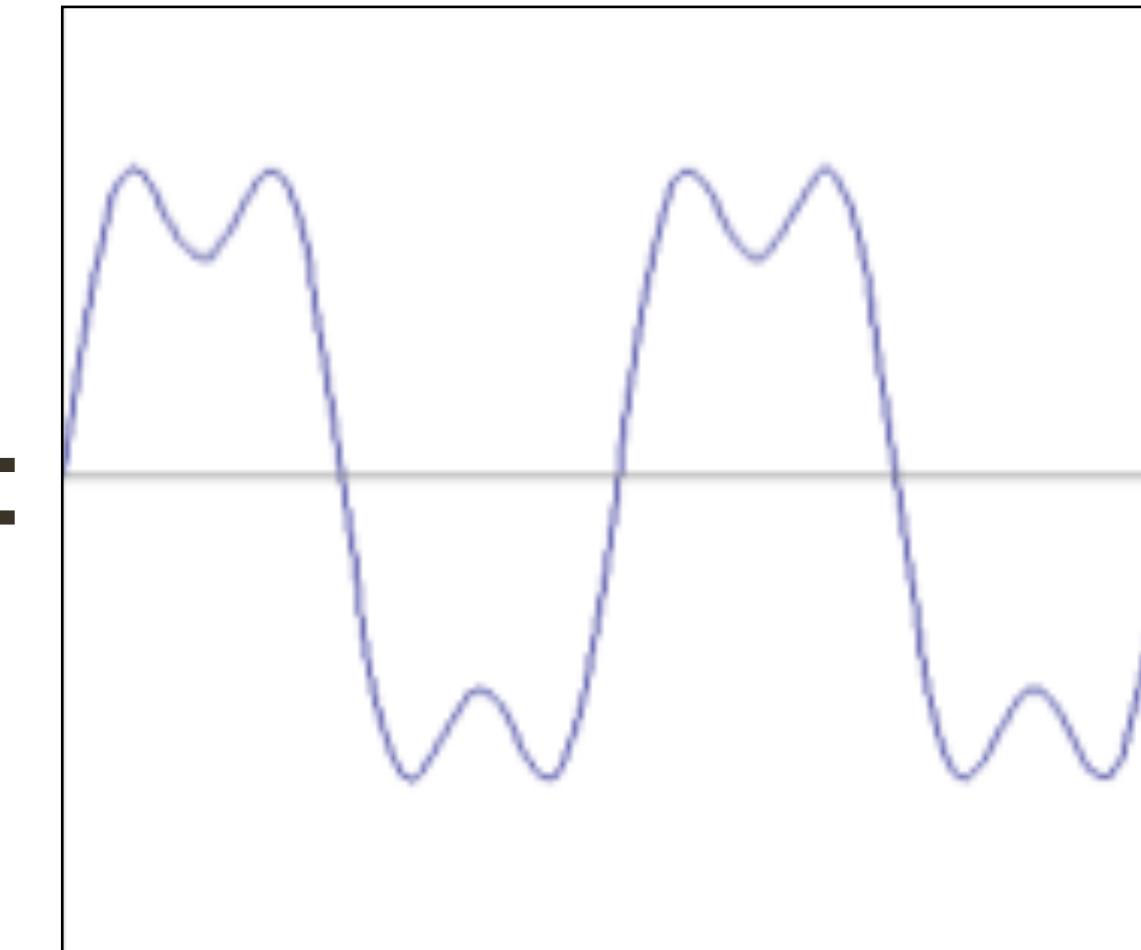
\approx



+

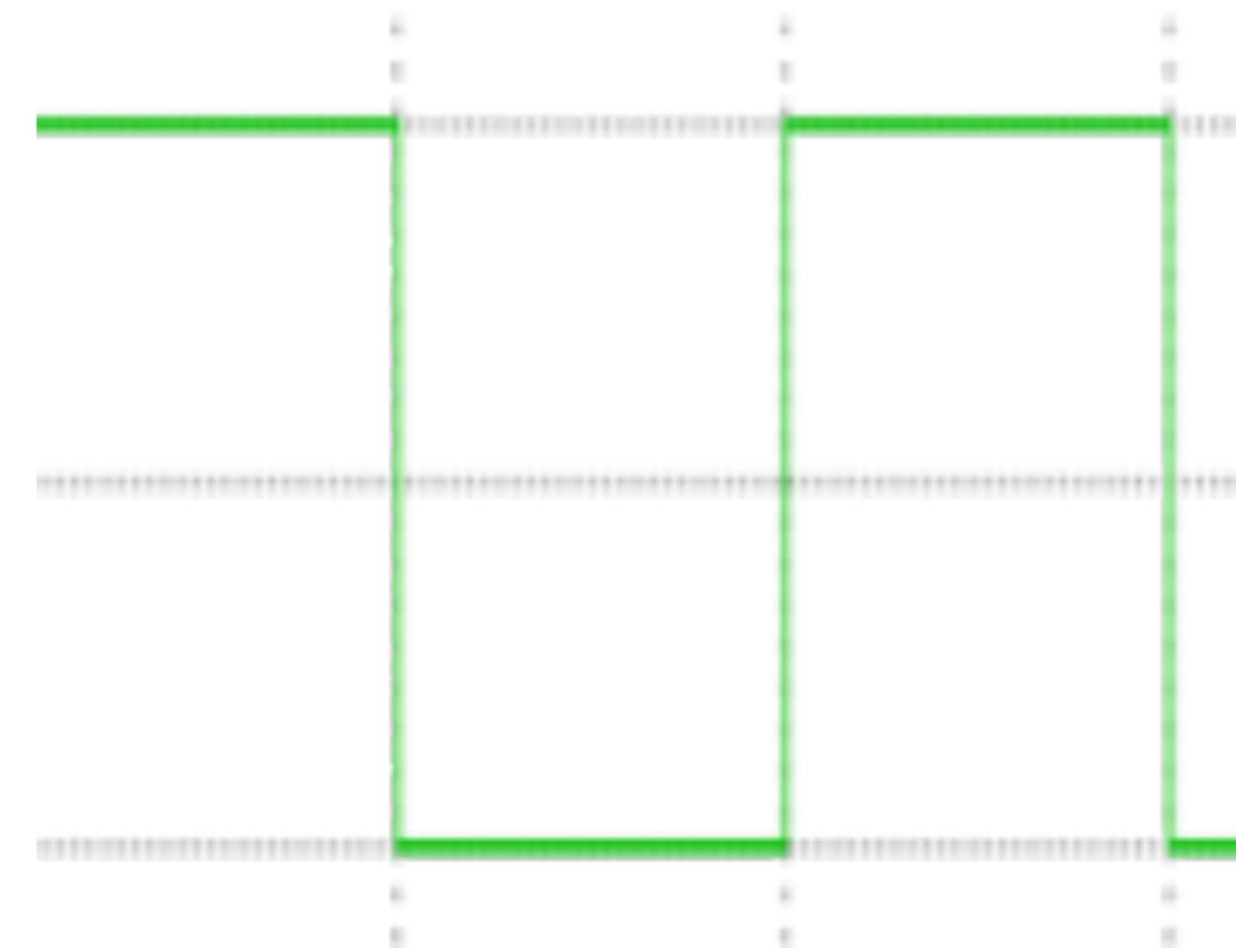


=

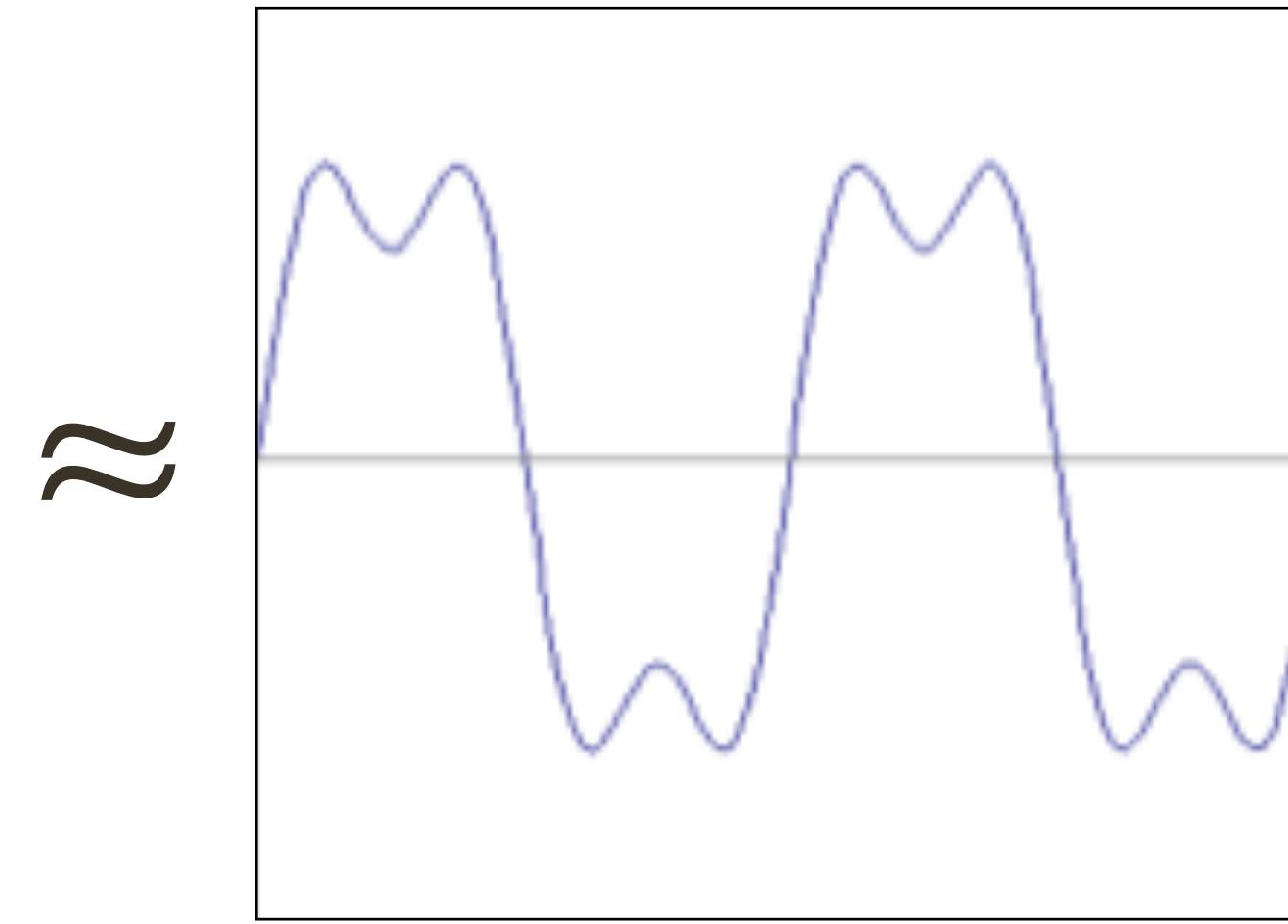


Fourier Transform (you will **NOT** be tested on this)

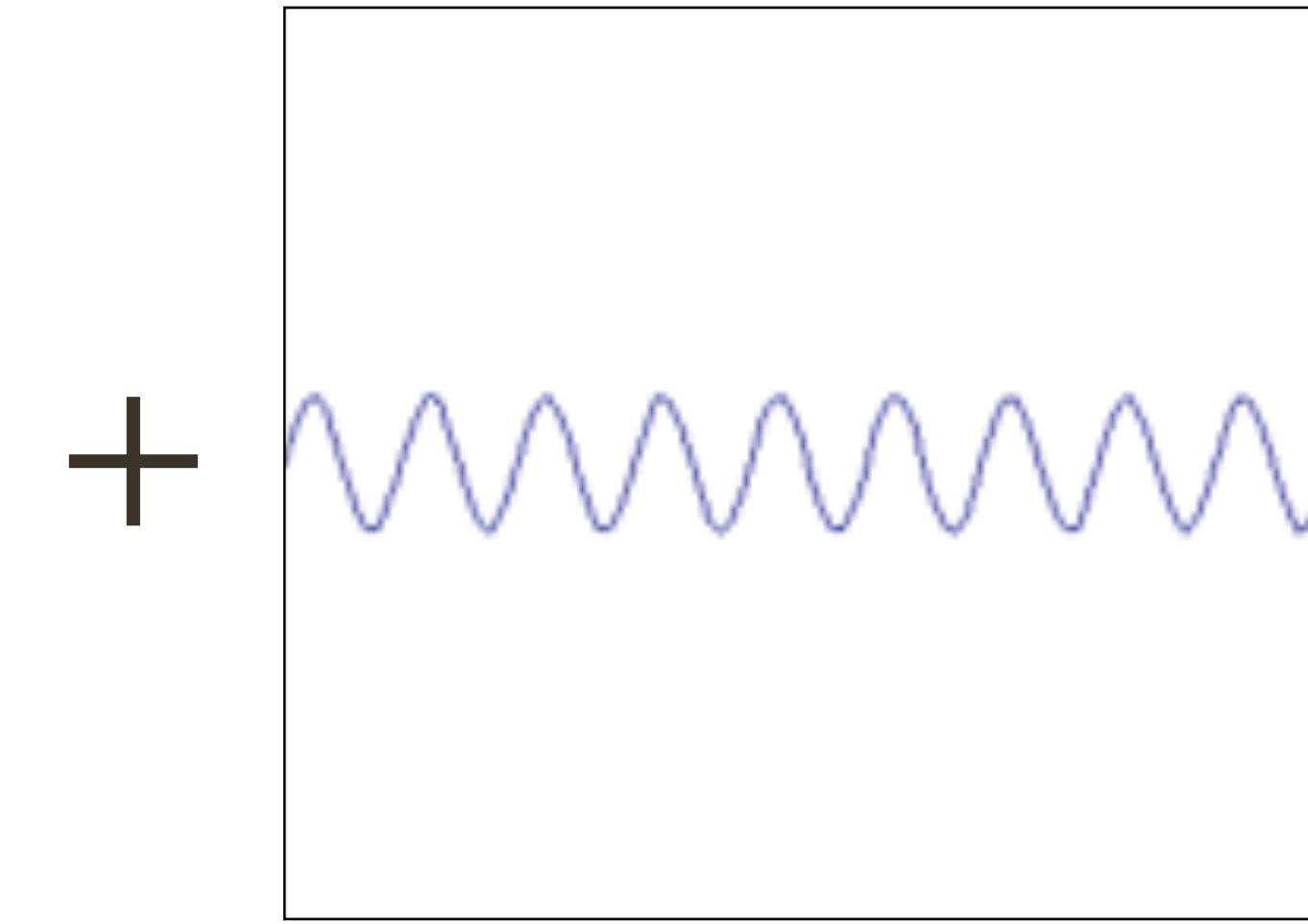
How would you generate this function?



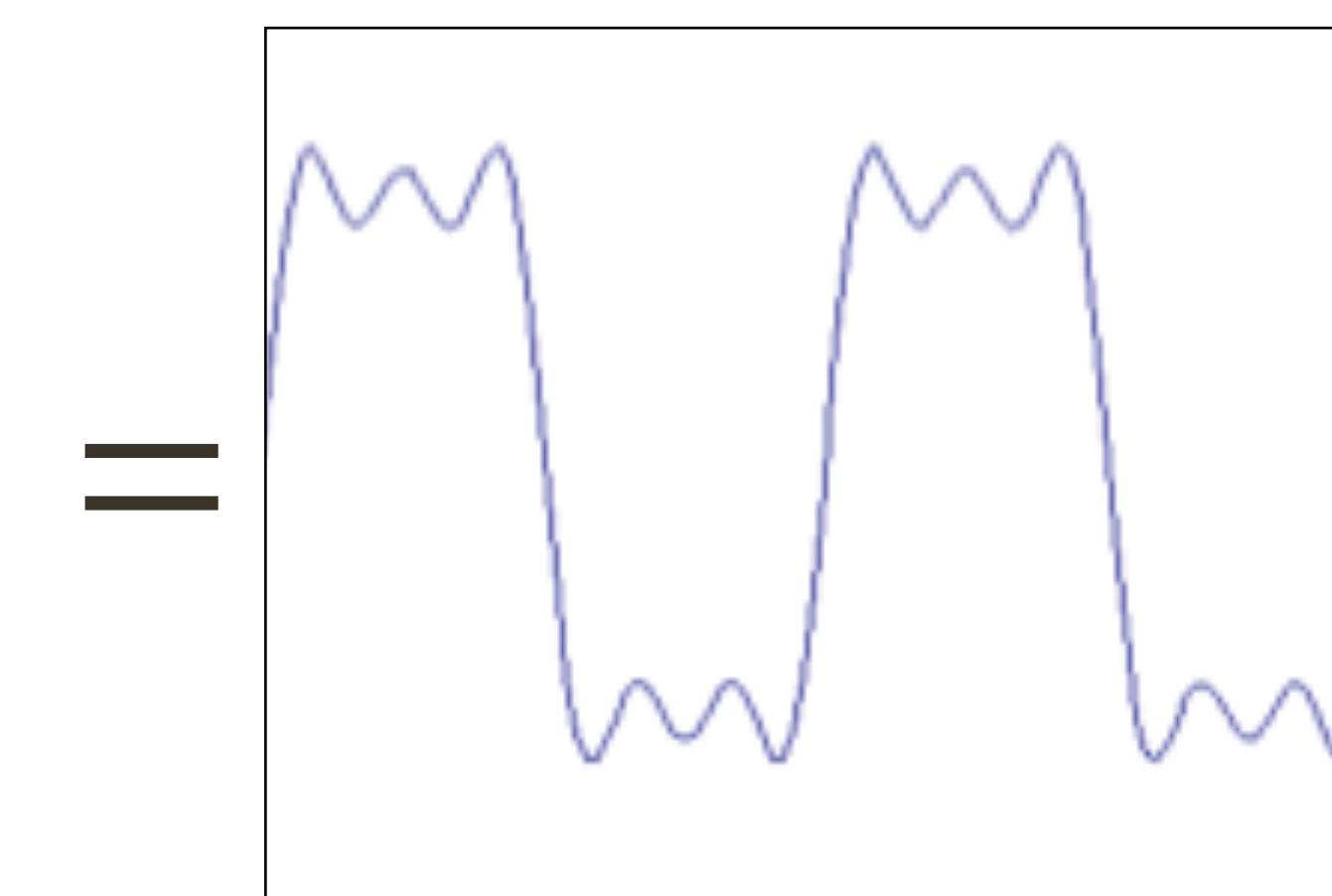
square wave



\approx



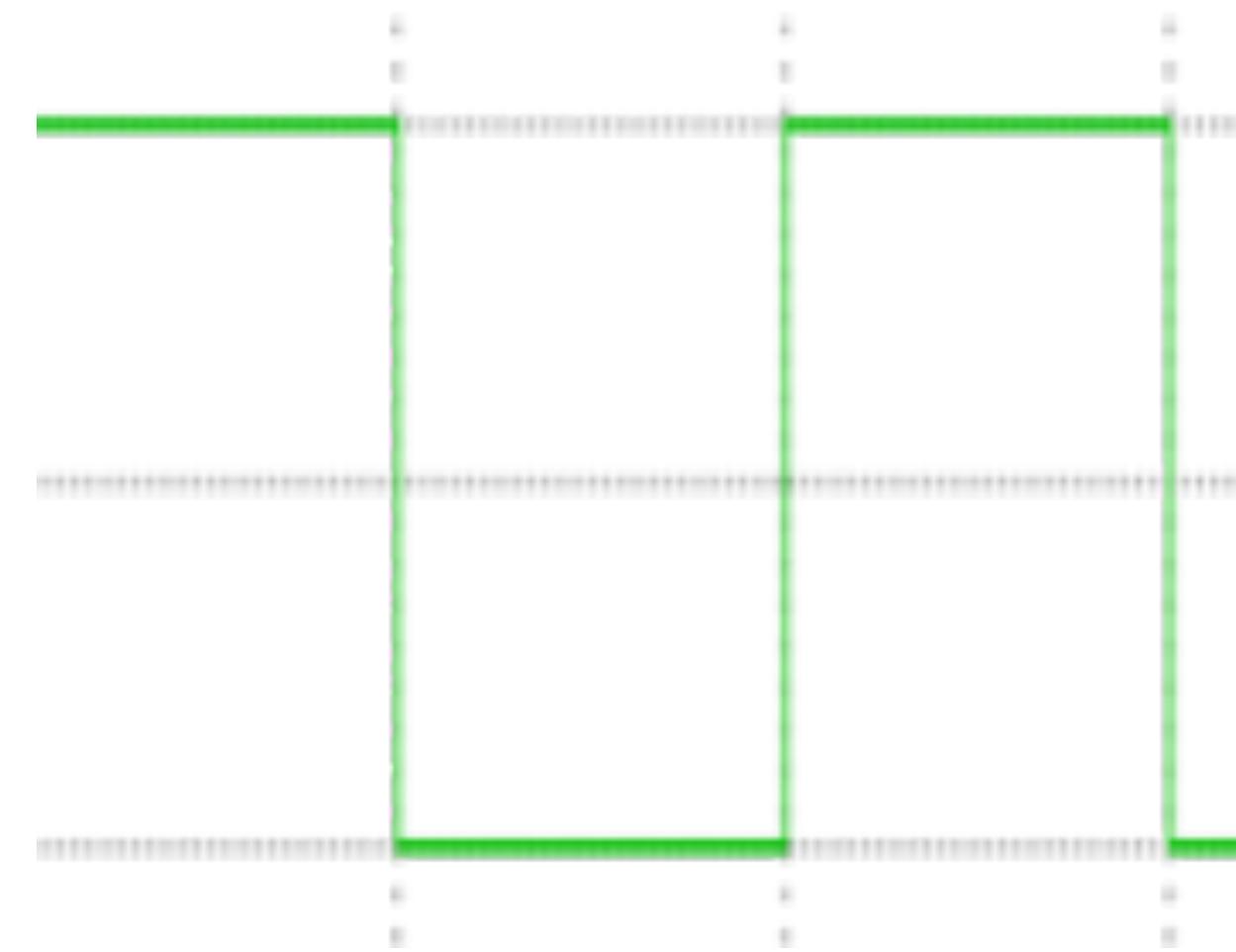
+



=

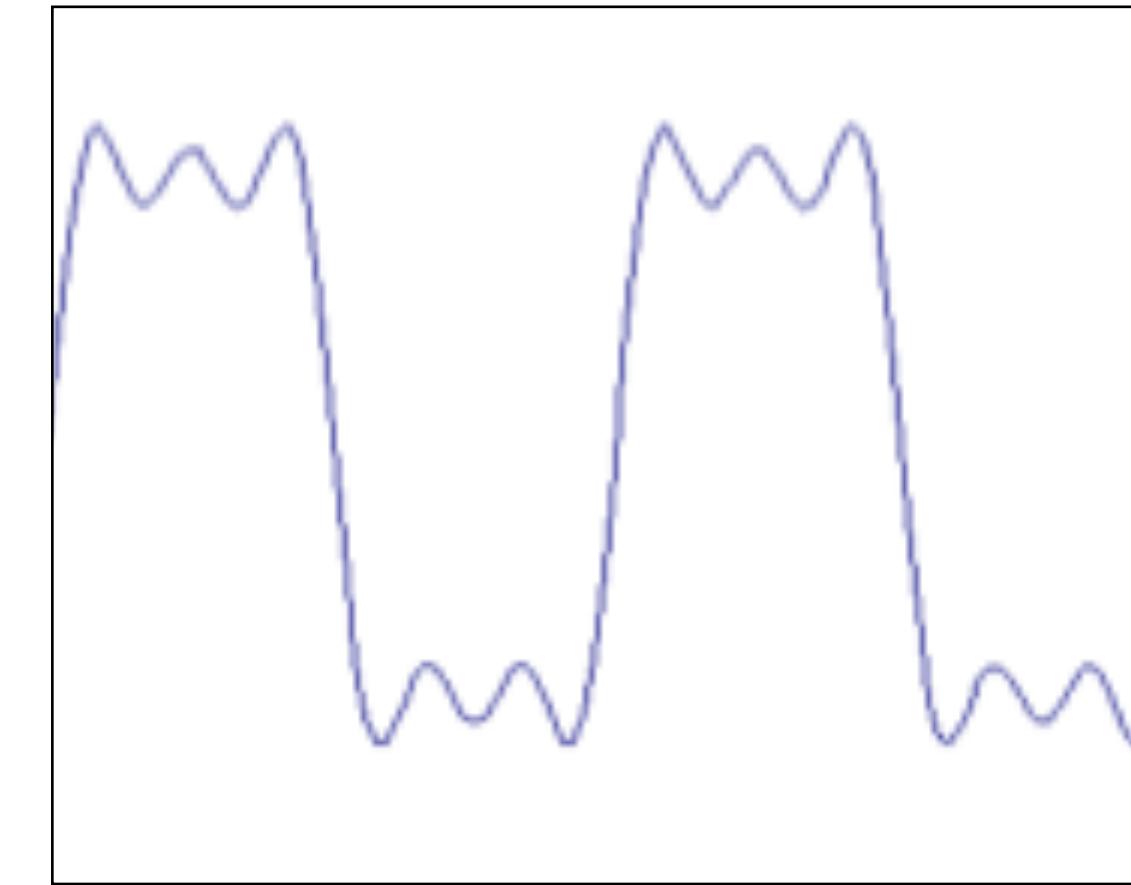
Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?

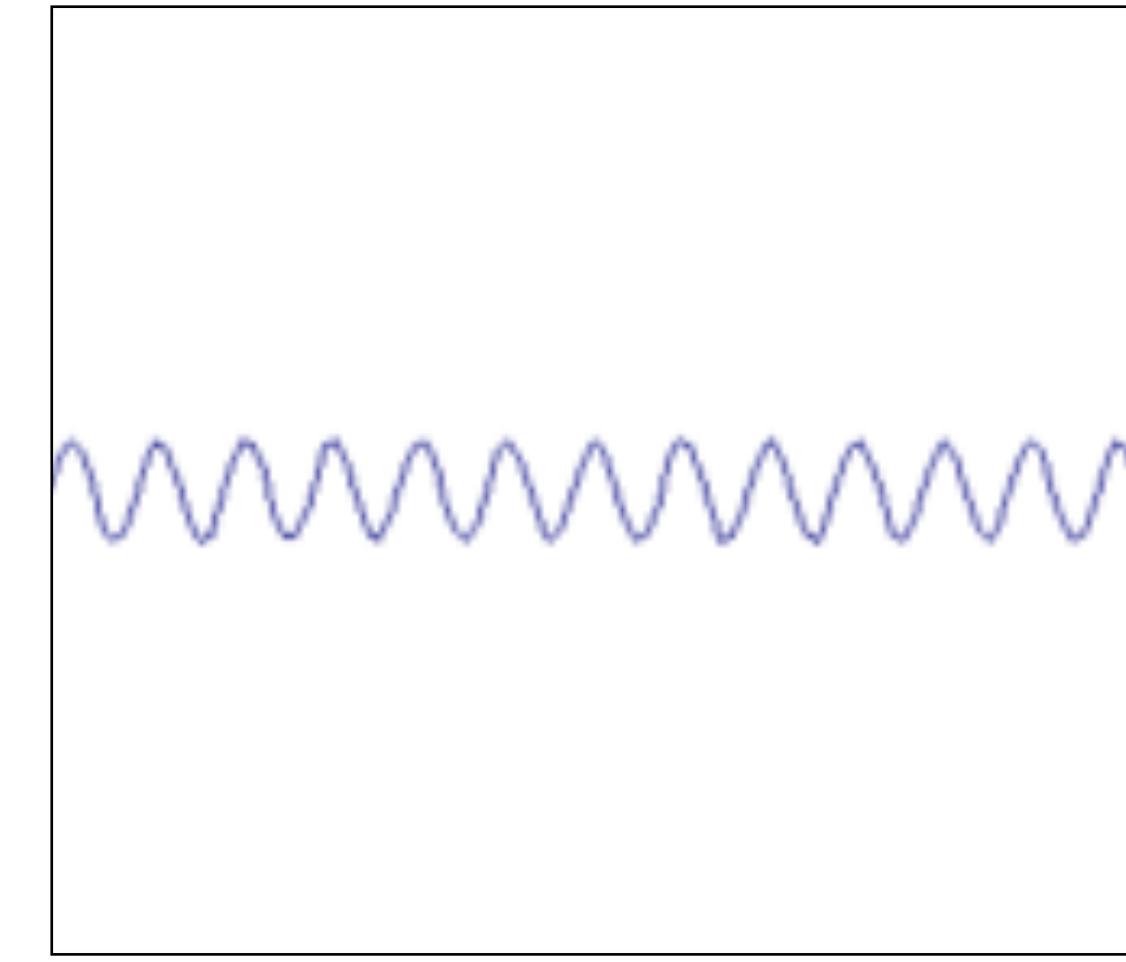


square wave

\approx



+

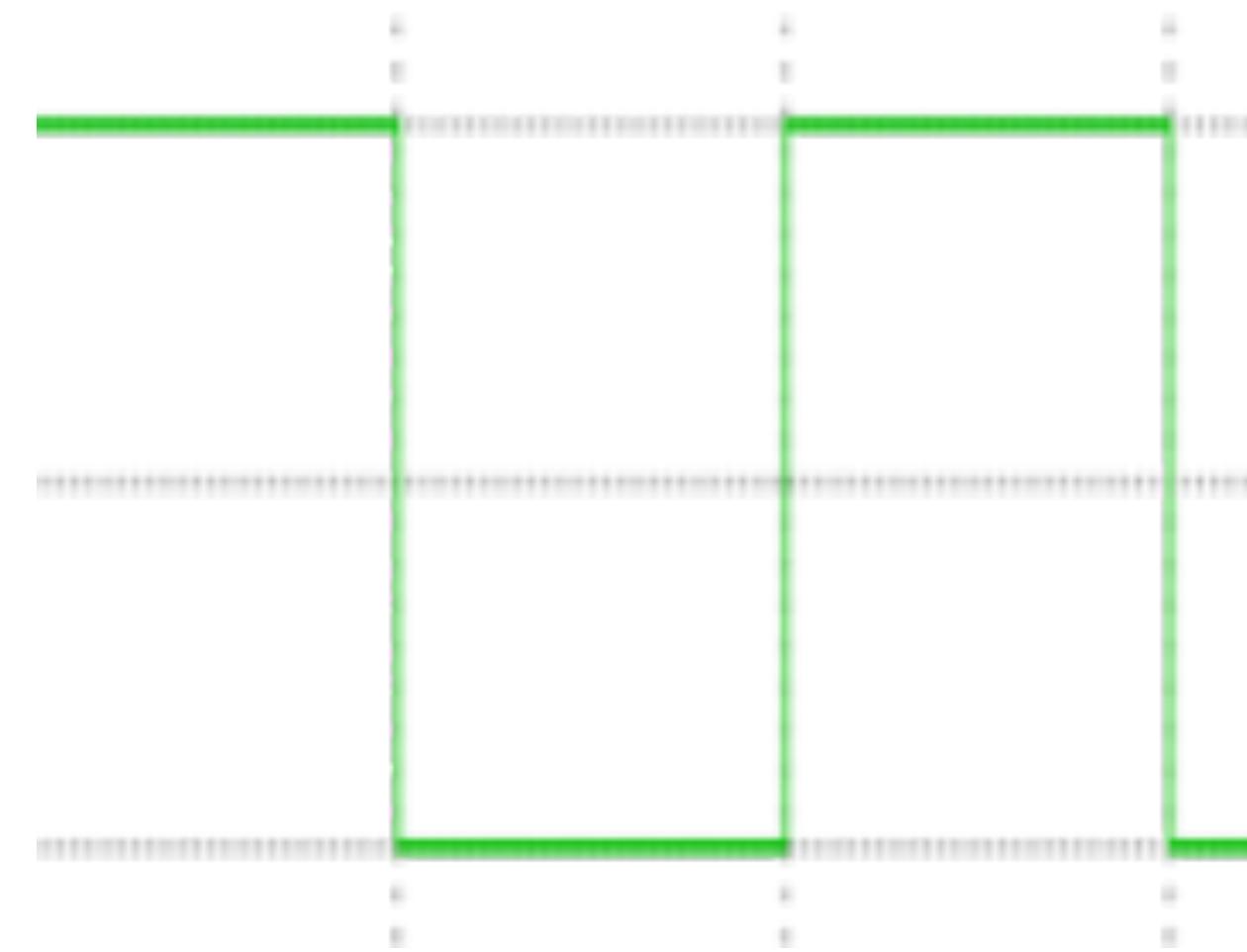


=



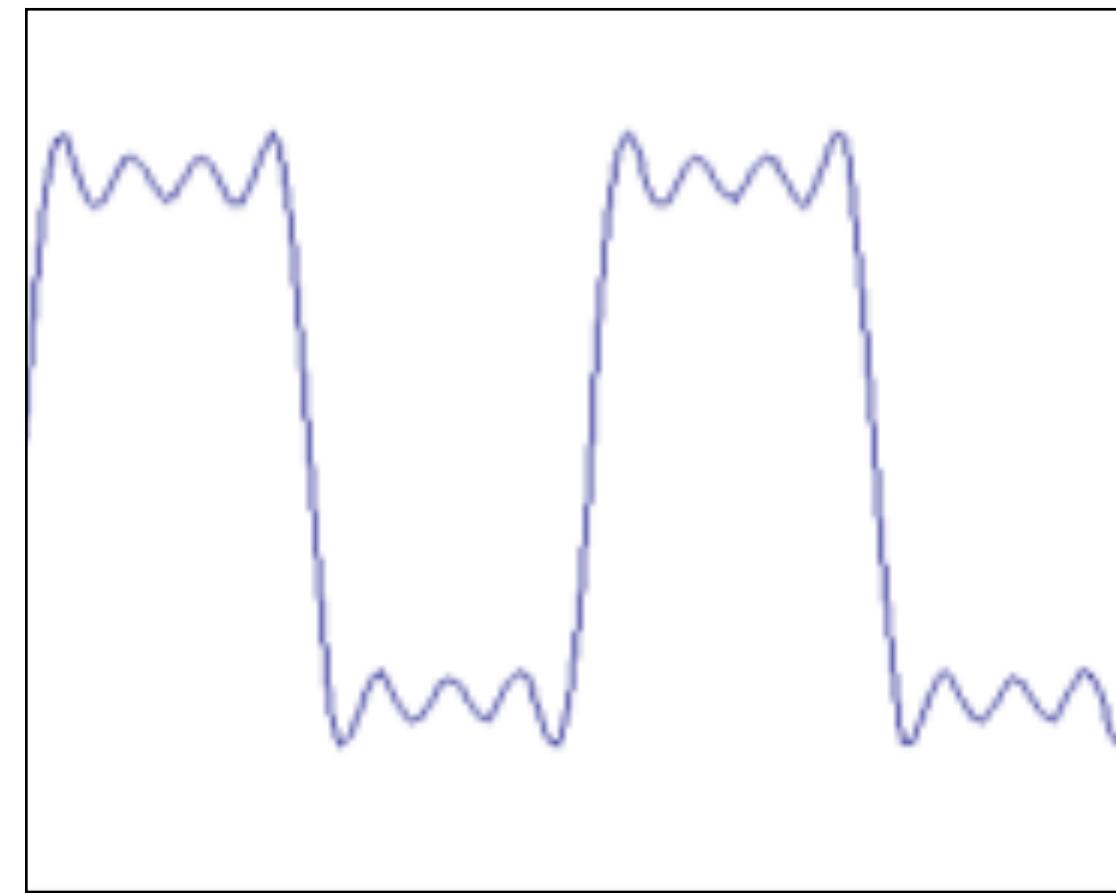
Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?

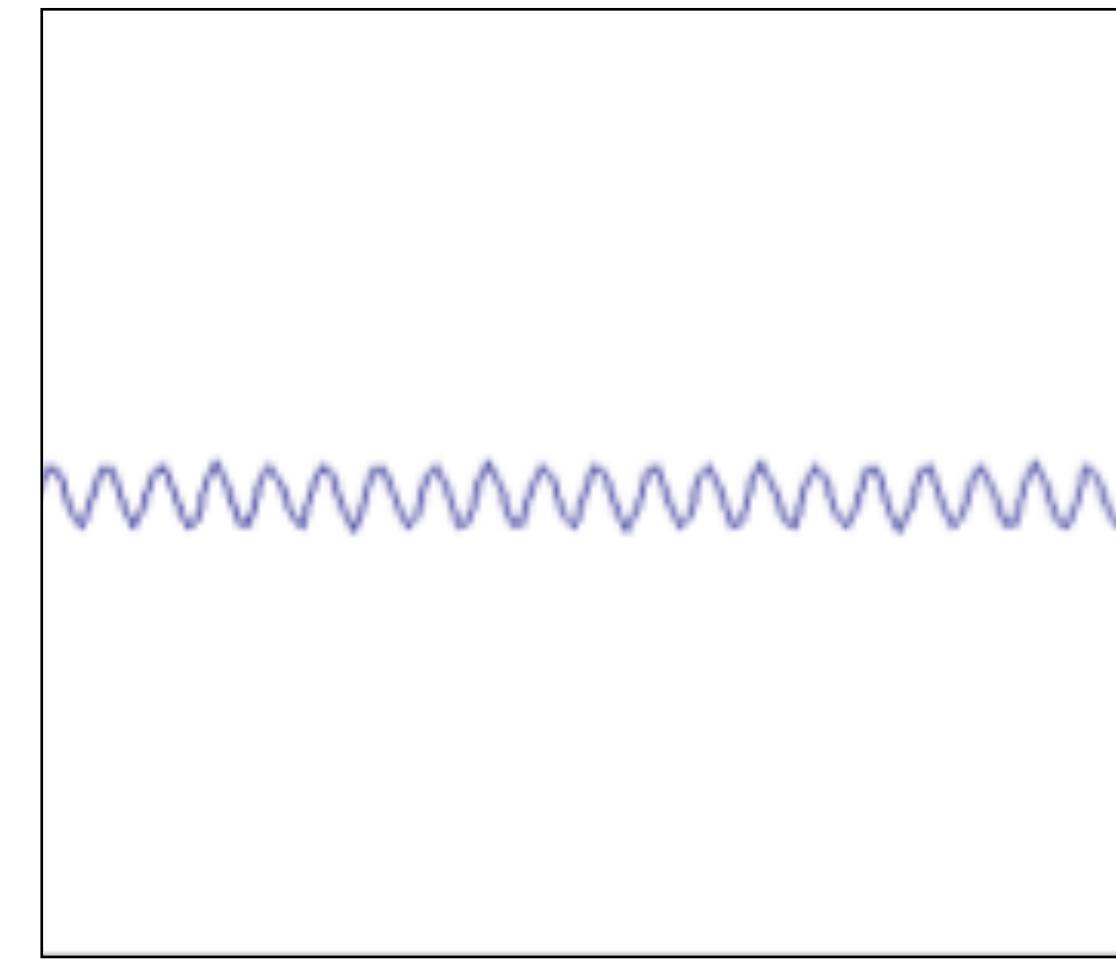


square wave

\approx



+



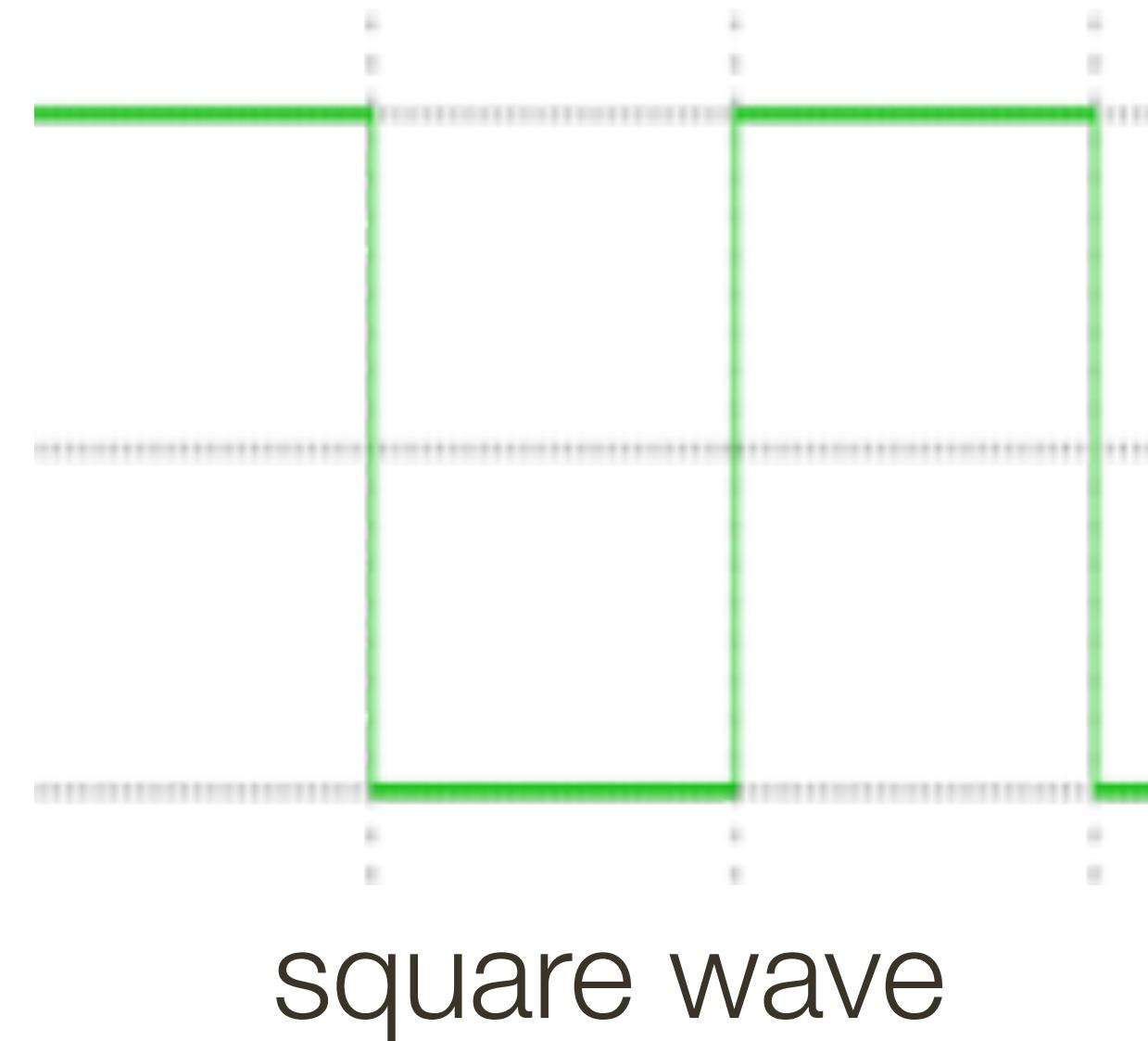
=



How would you
express this
mathematically?

Fourier Transform (you will **NOT** be tested on this)

How would you generate this function?



$$= A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi kx)$$

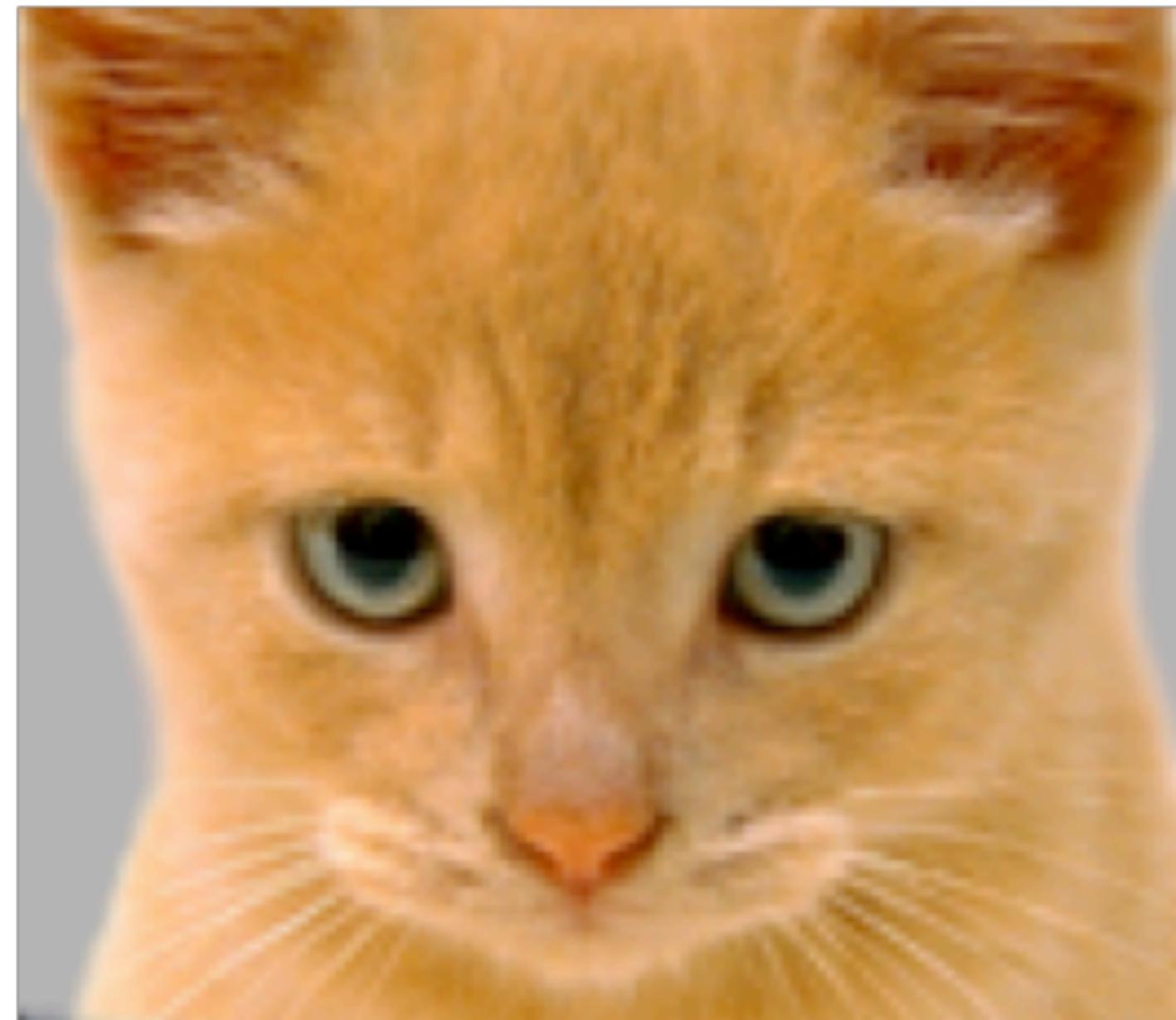
infinite sum of sine waves

Low-Pass Filtering in 1D



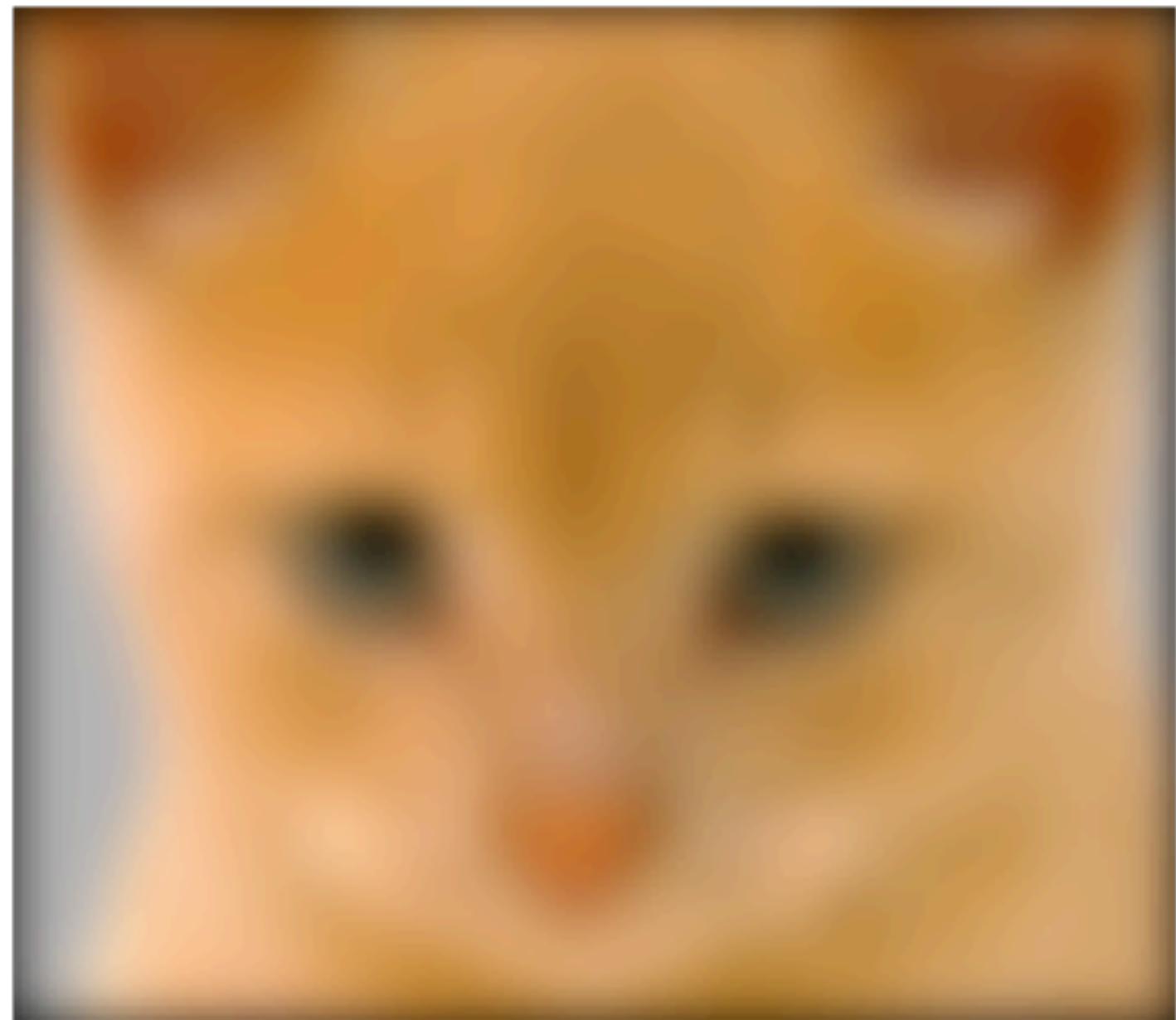
4.4

Assignment 1: Low/High Pass Filtering



Original

$$I(x, y)$$



Low-Pass Filter

$$I(x, y) * g(x, y)$$



High-Pass Filter

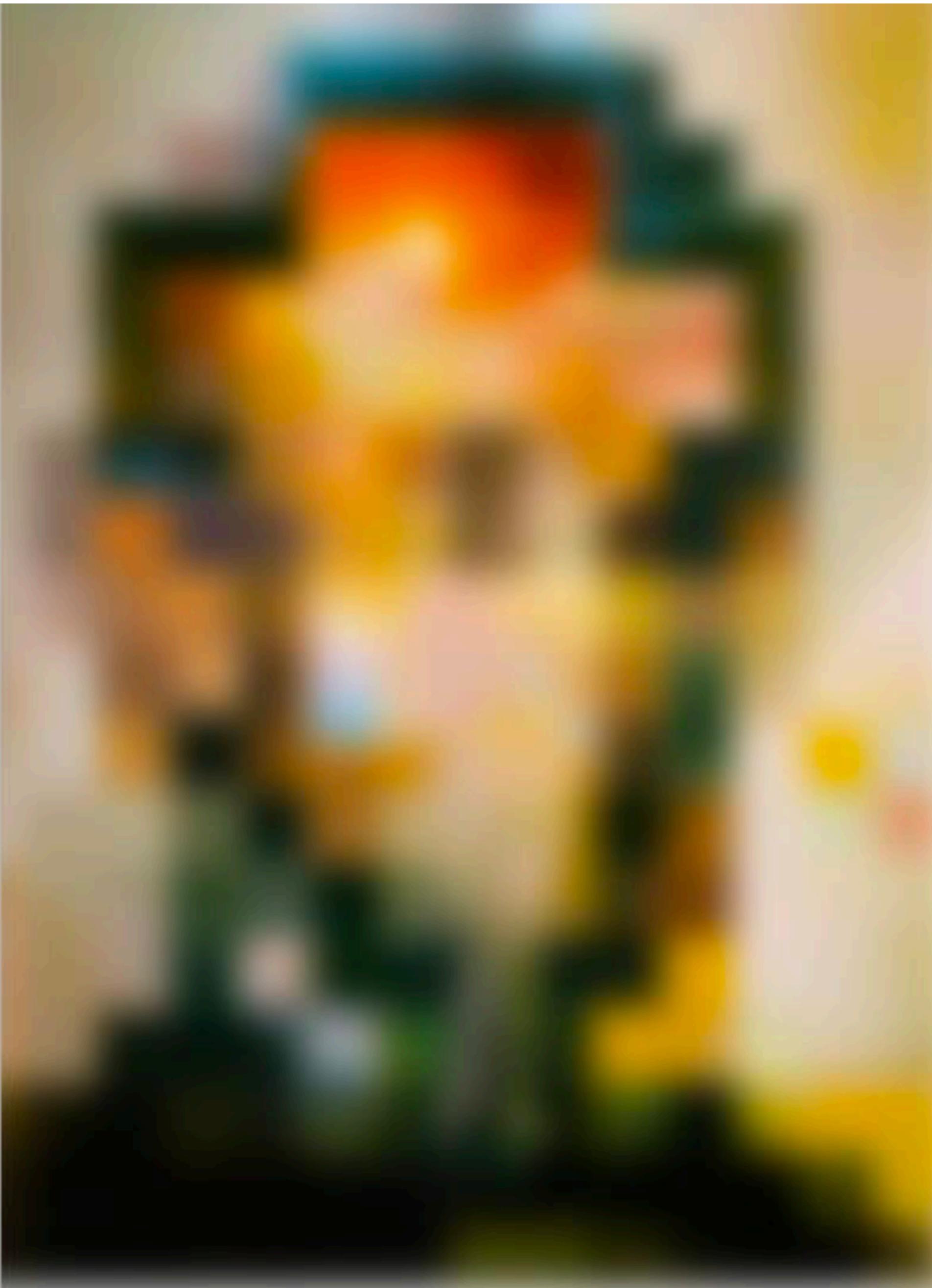
$$I(x, y) - I(x, y) * g(x, y)$$



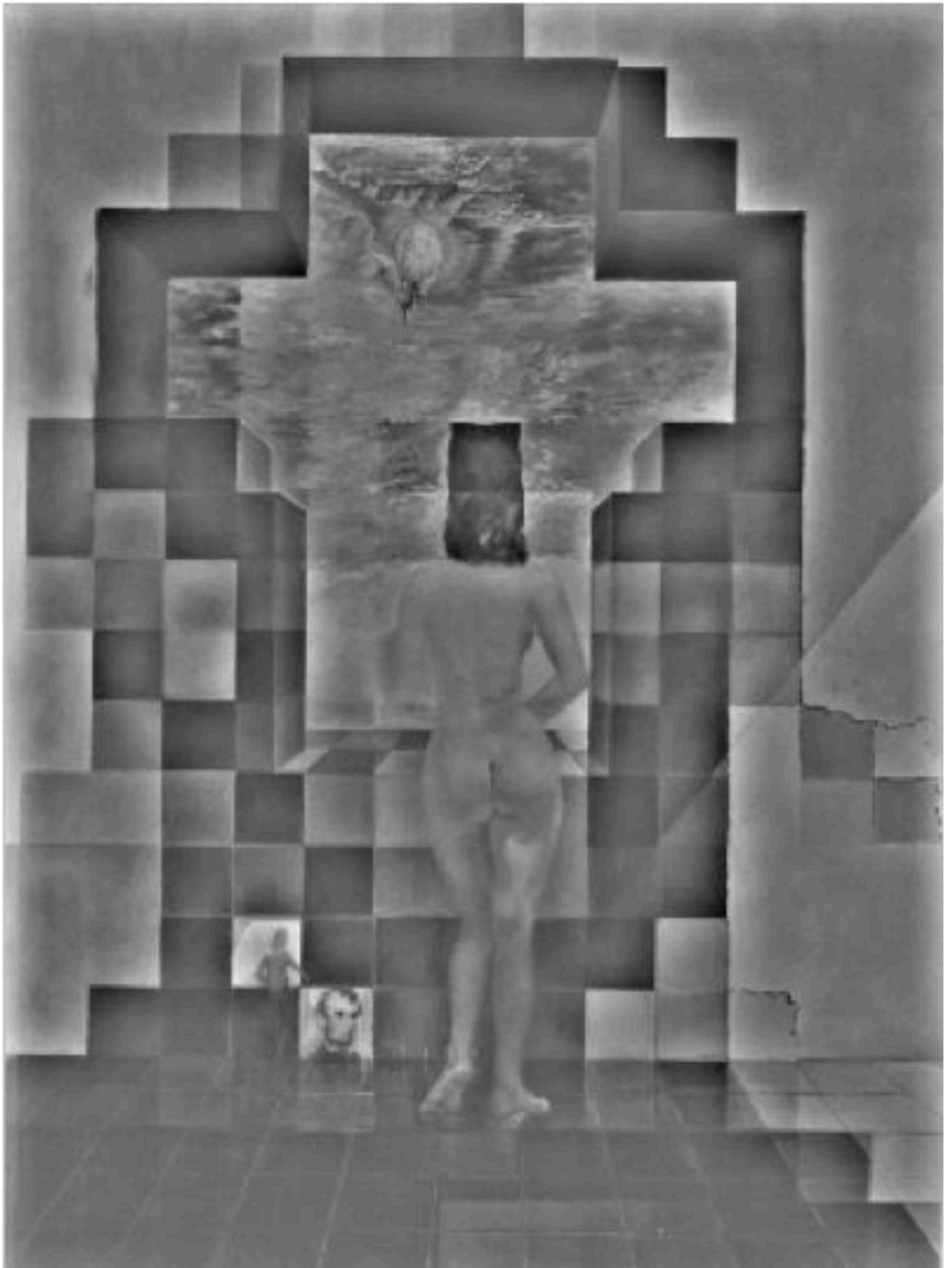
*Gala Contemplating the Mediterranean
Sea Which at Twenty Meters Becomes
the Portrait of Abraham Lincoln
(Homage to Rothko)*

Salvador Dali, 1976

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)



Low-pass filtered version



High-pass filtered version

Menu for Today

Topics:

- Recap L3, **iClicker test**
- **Box, Gaussian, Pillbox** filters
- **Low/High Pass** Filters
- **Separability**

Readings:

- **Today's** Lecture: none
- **Next** Lecture: Forsyth & Ponce (2nd ed.) 4.4

Reminders:

- **Assignment 1:** Image Filtering and Hybrid Images is out, due **Sept 28th**