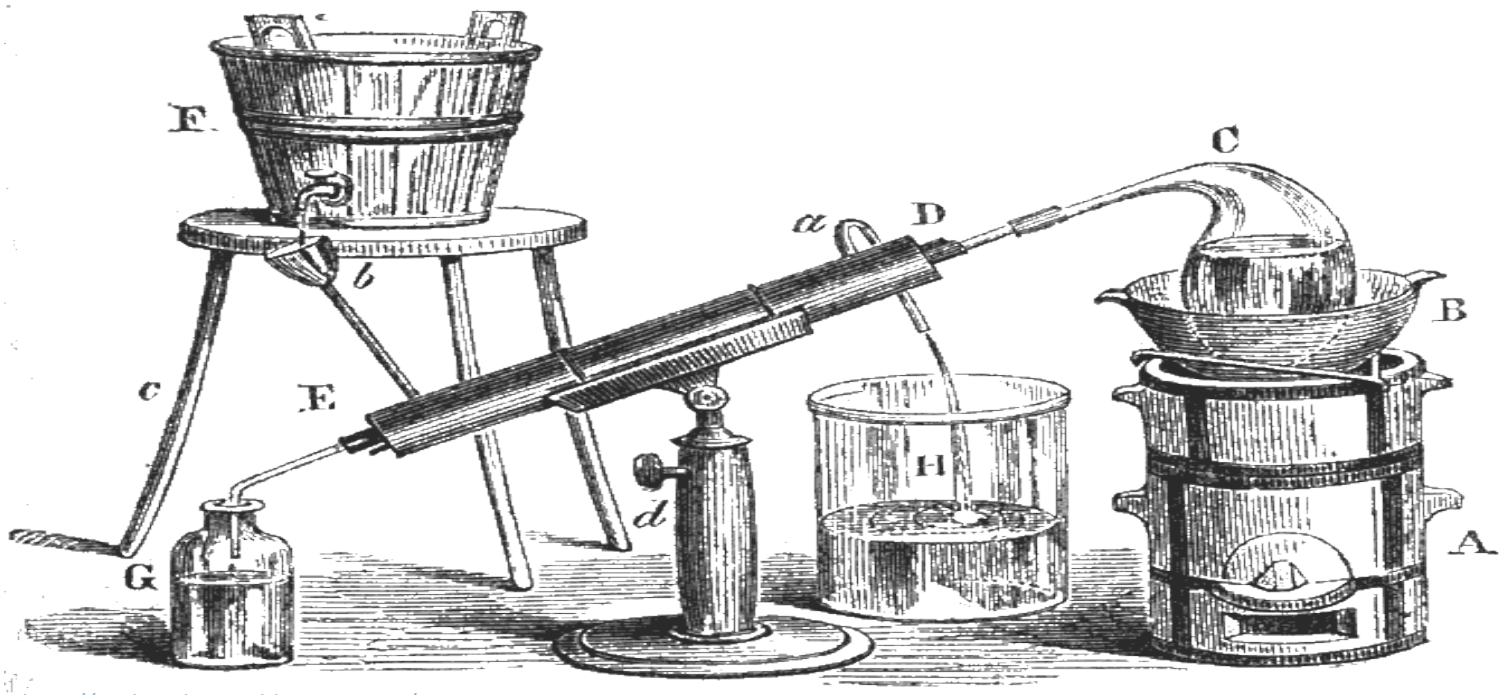
Filtering

Back to Basics



http://nodewebstore.blogspot.com/

Brent M. Dingle, Ph.D.
Game Design and Development Program
Mathematics, Statistics and Computer Science
University of Wisconsin - Stout

2015

Material in this presentation is largely based on/derived from presentation(s) and book: The Digital Image by Dr. Donald House at Texas A&M University

Previously

- Image Filtering
  - Examples
    - Low Pass
    - High Pass
    - Directional
    - Global
      - Normalization and Histogram Equalization
  - Relation to Image Enhancement
    - Spatial Domain
      - not so much the frequency domain

Outline

- Filters and Convolutions
  - General Definitions
    - Working around the Edges
  - Low Pass Examples
  - High Pass Examples

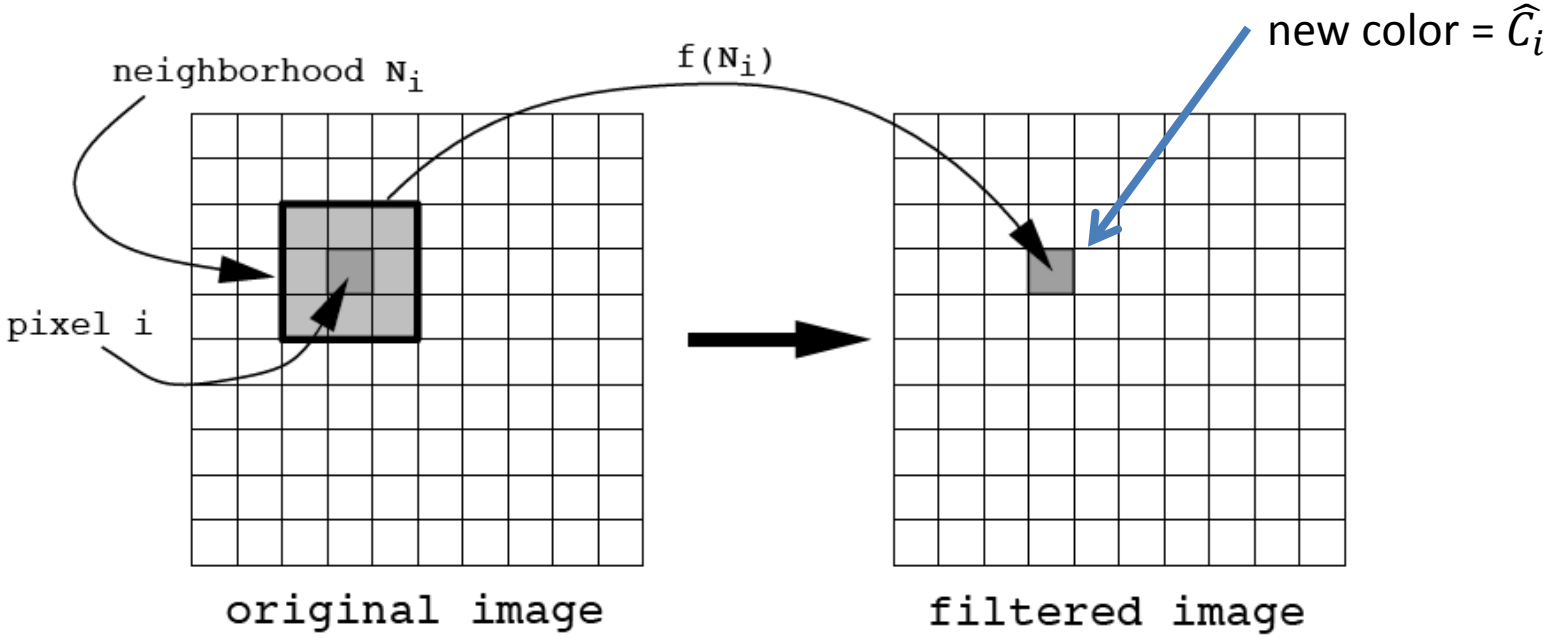
Filtering

- Filtering
  - modify an image based on image color content without any intentional change in image geometry
    - resulting image essentially has the same size and shape as the original

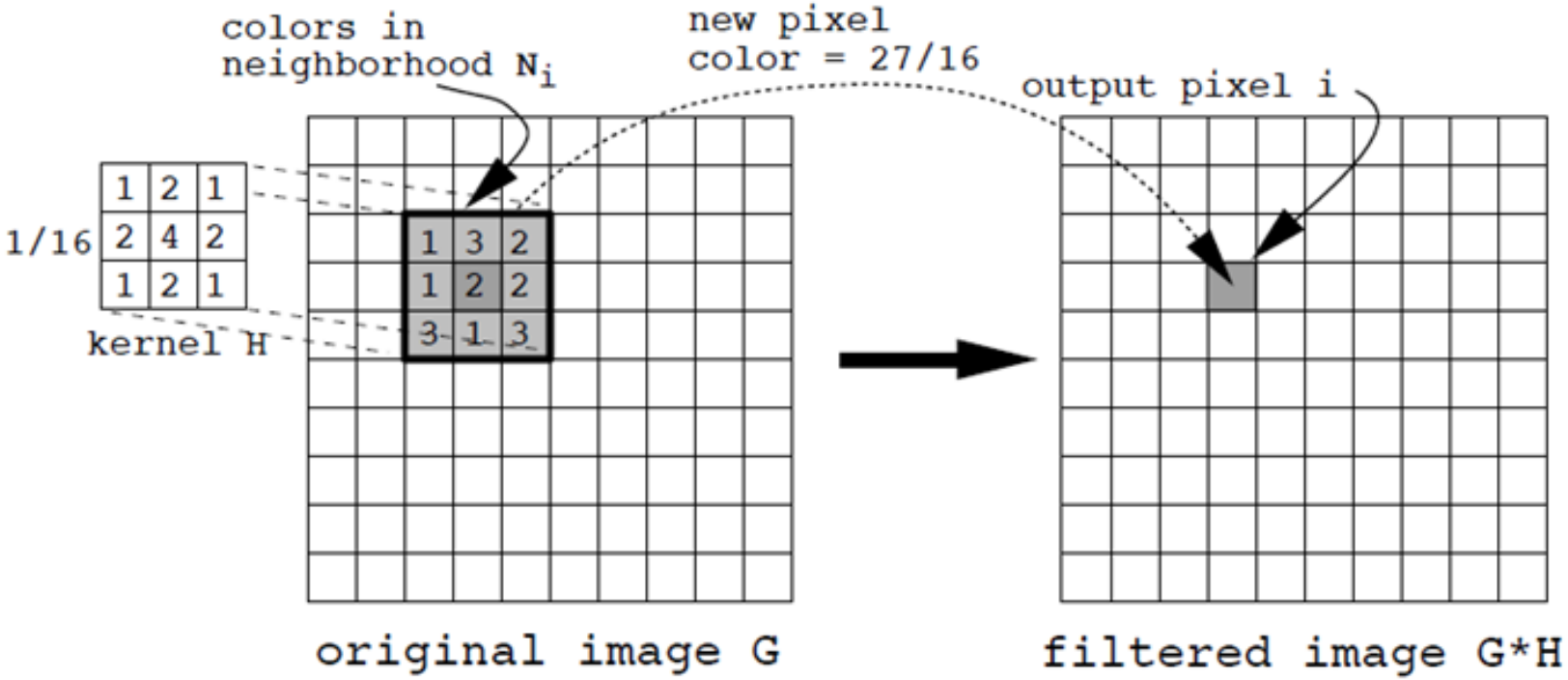
Image Filtering Operation

- Image Filtering Operation
  - Let $\boldsymbol{P_i}$ be the single ***input pixel*** with index *i* and color C_i.
  - *Let* $\widehat{\boldsymbol{P_i}}$ be the corresponding **output pixel** with color $\widehat{\boldsymbol{C_i}}$.
  - ***A Filtering Operation***
    - associates each pixel, $\boldsymbol{P_i}$,with a neighborhood set of pixels, $\boldsymbol{N_i}$, and determines an output pixel color via a ***filter function***, ***f***, such that:

$$\widehat{C_i} = f(N_i)$$



Convolution Filtering

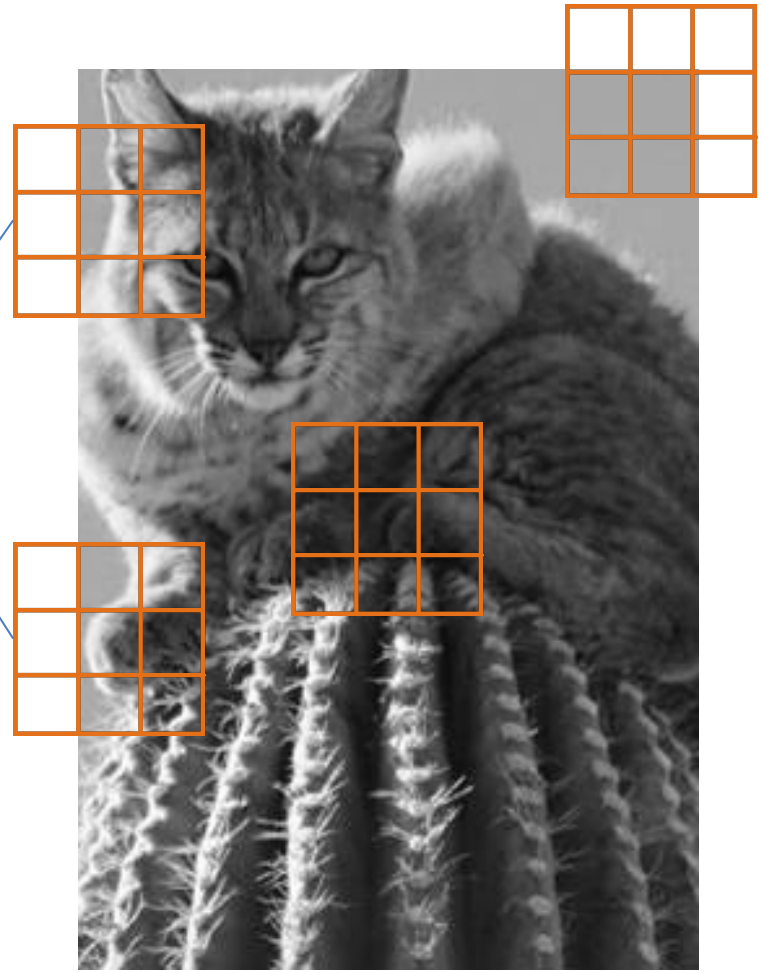


Effectively this is doing a weighted sum

Weighted Sum = ((1*1) + (2*3) + (1*2)
+ (2*1) + (4*2) + (2*2)
+ (1*3) + (2*1) + (1*3)) * (1/16) = 27/16

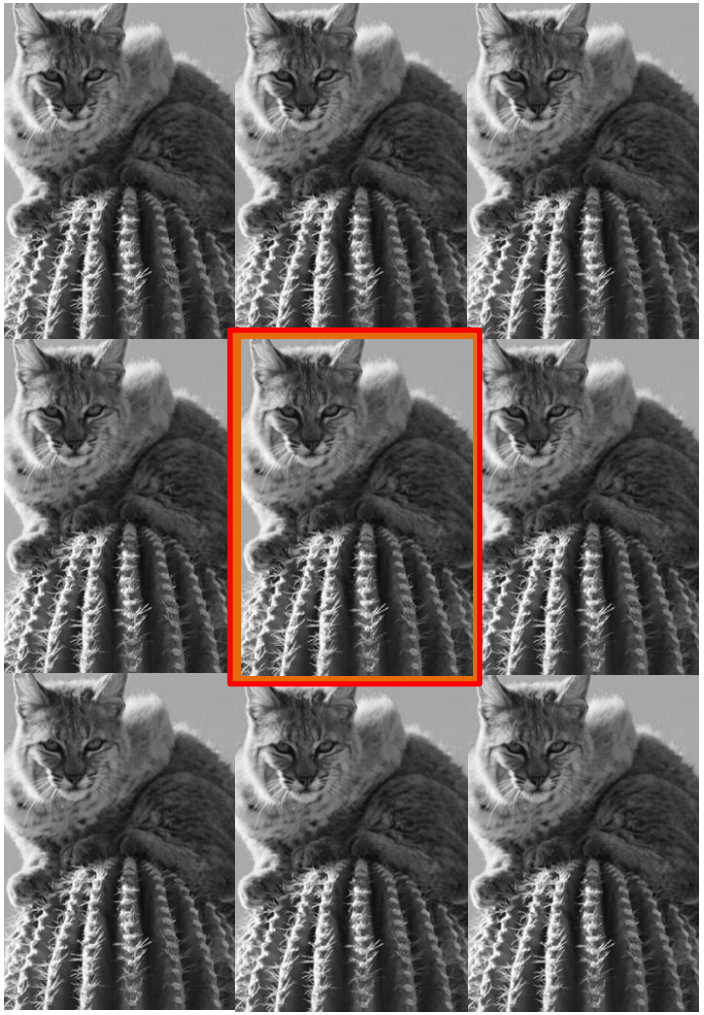
Tricky Around the Edges

Neighborhood
Needed
for some pixels
is out of the image



Many options… let's look at a good choice

Expand the Input Image: Tiling



This orientation may have problems

Do the pixels at the edges
"match" from one side to the other?

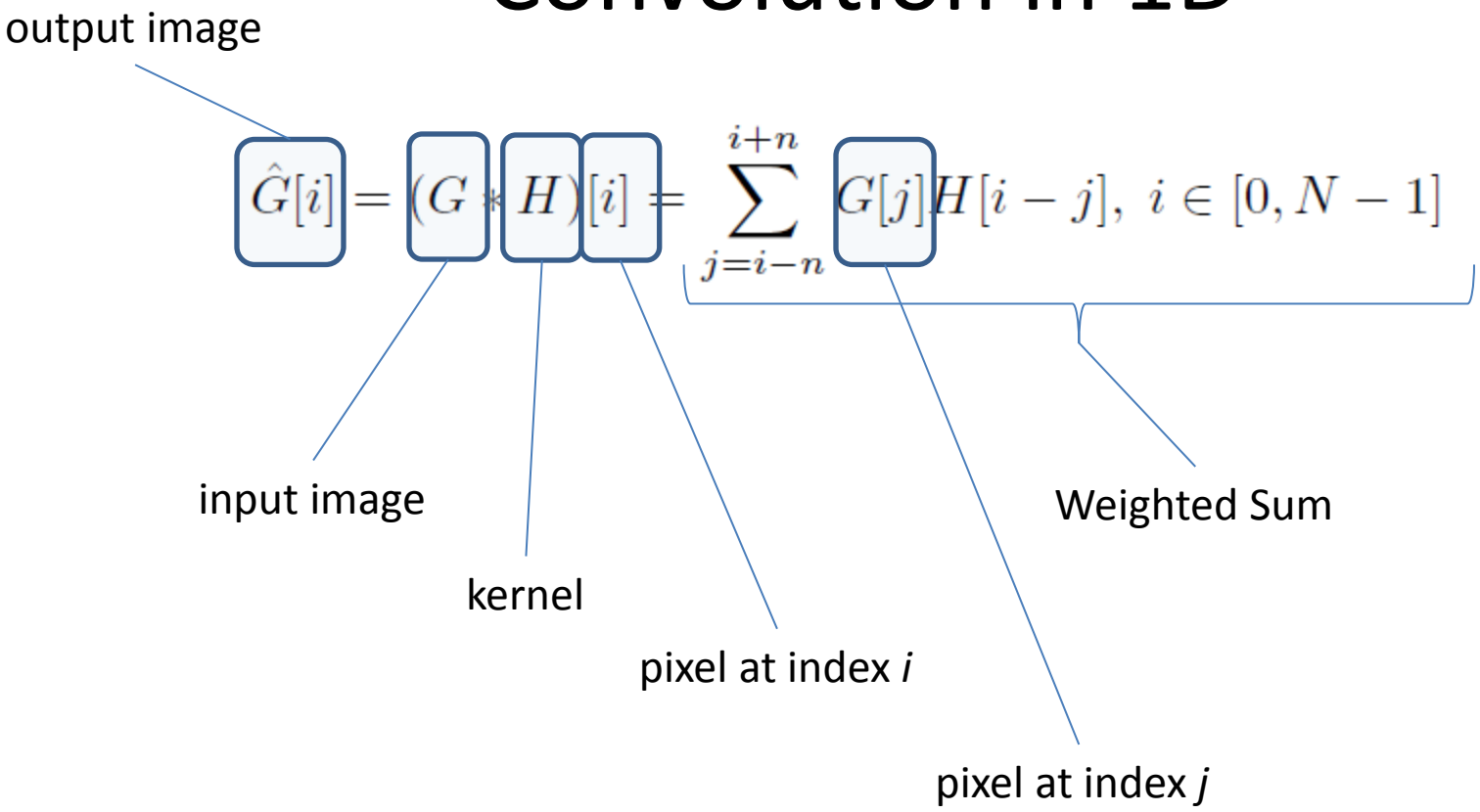
→ NO

Expand the Input Image: Tiling

Flipping/Reflecting the image
Then Tiling

Usually works
and
Is a GOOD Strategy

Convolution in 1D

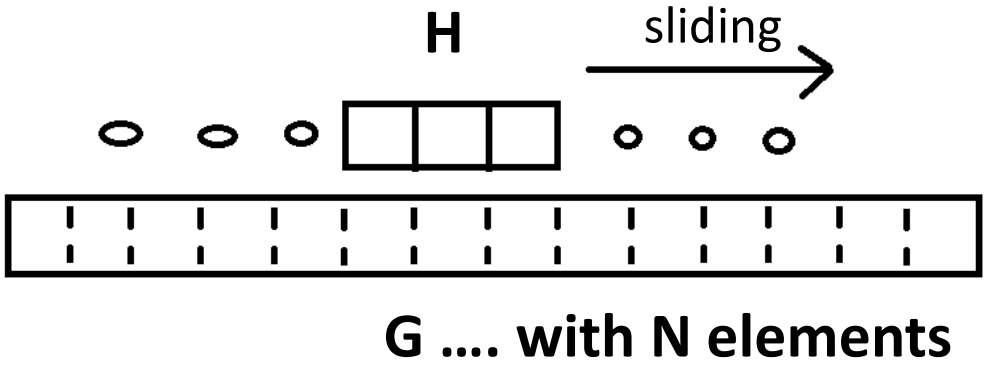


Convolution in 1D

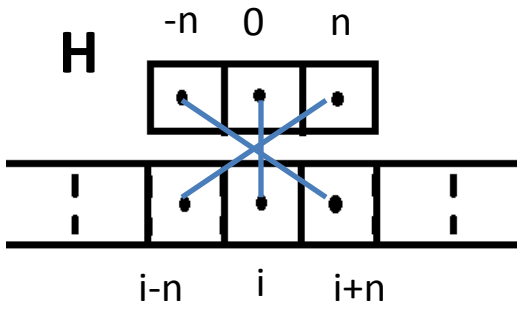
$$\hat{G}[i] = (G * H)[i] = \sum_{j=i-n}^{i+n} G[j]H[i-j],\ i \in [0, N-1]$$

set j from (i-n) to (i+n)
H from n to -n

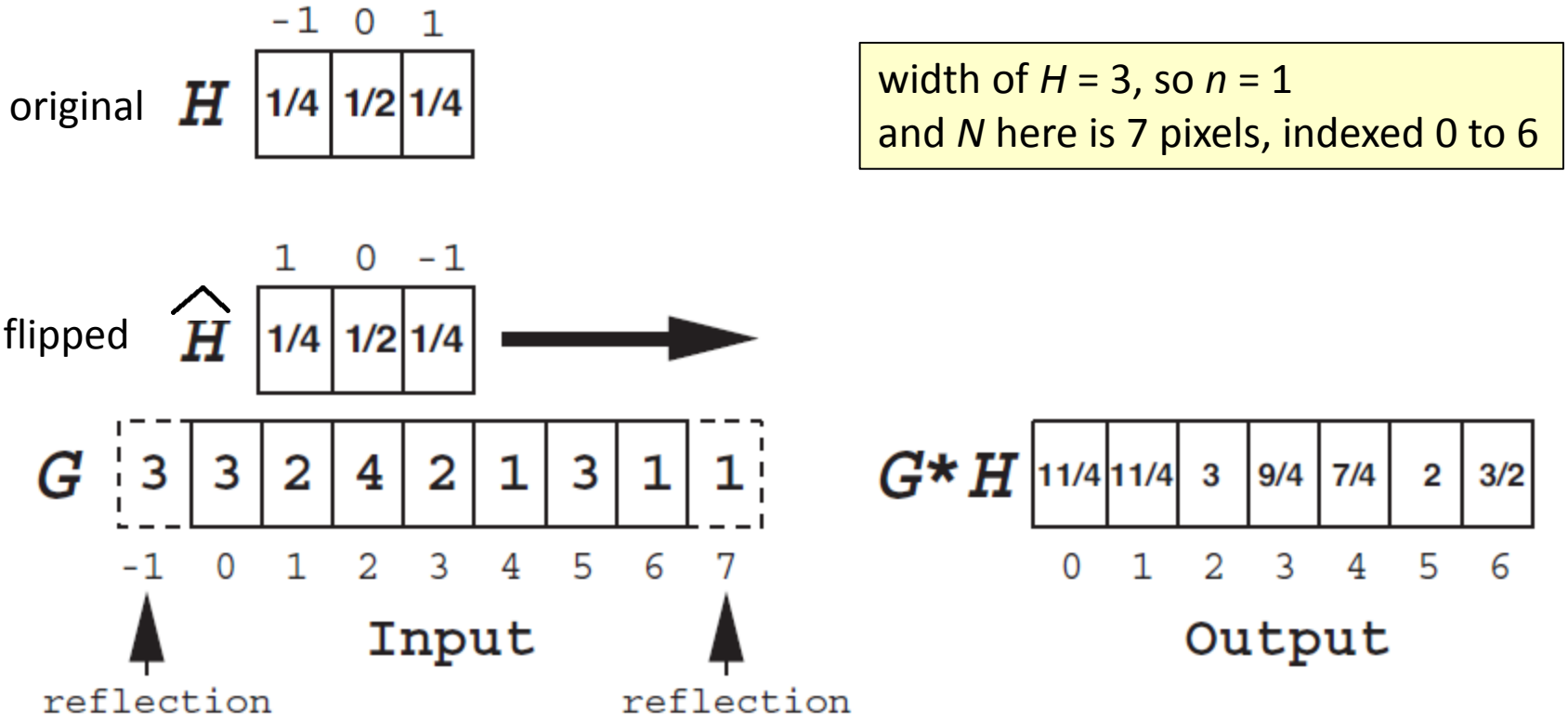
reflected or flipped



As drawn kernel H has a width, *w = 3* so *n = 1*

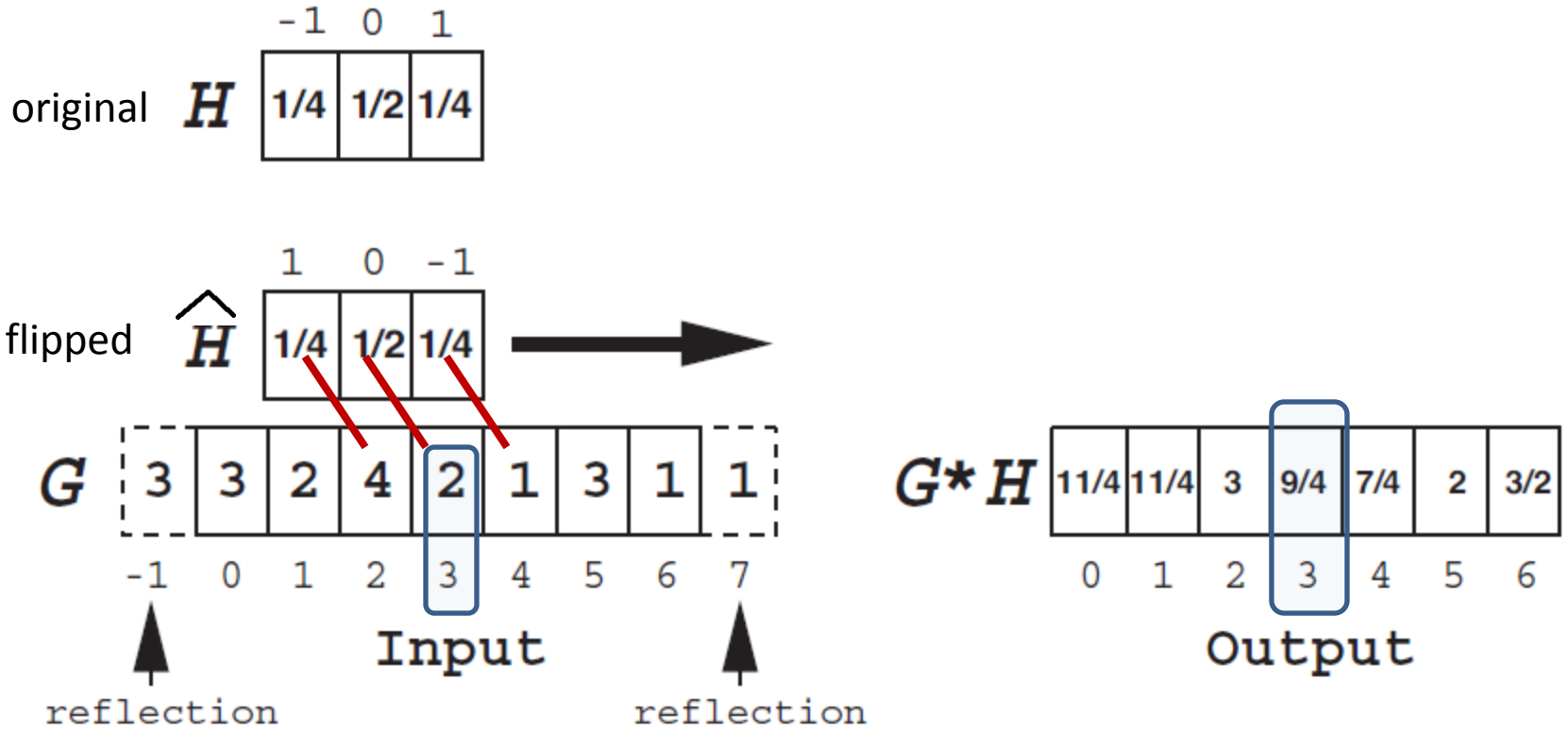


1D Example: Convolution *G*H*



Example from: The Digital Image by Dr. Donald House at Texas A&M University, Figure 8.11

1D Example: Convolution *G*H*



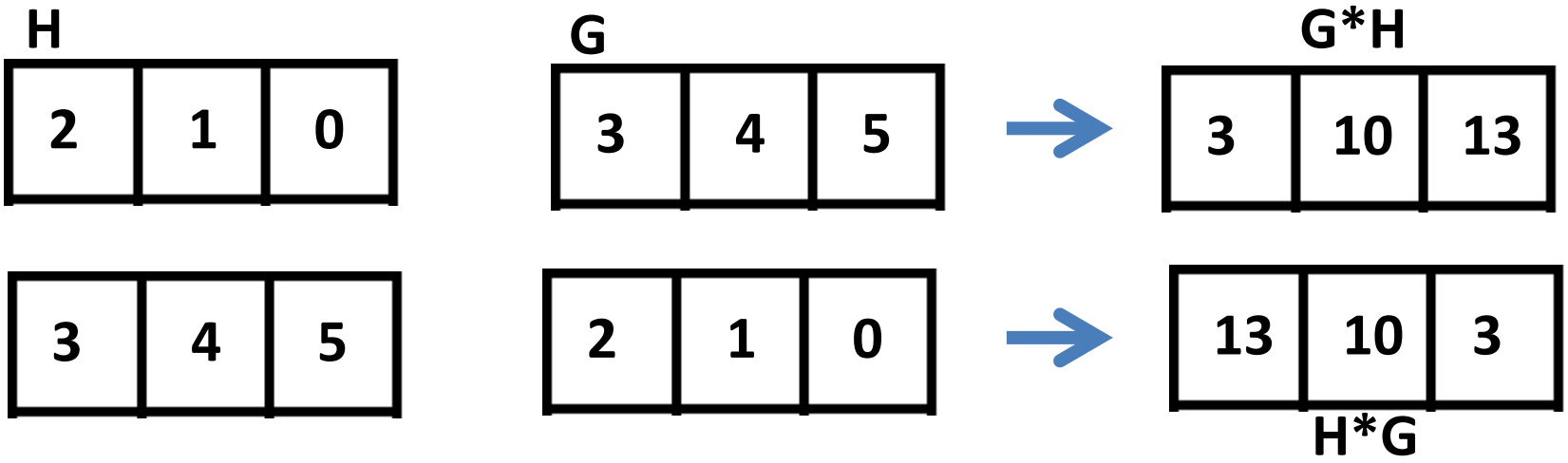
Example for $i = 3$

$\sum_{j=3-1}^{3+1} G(j)H(3-j) =$ G(2)H(1) + G(3)H(0) + G(4)H(-1)
= (4*1/4) + (2*1/2) + (1*1/4)
= 9/4

Example from: The Digital Image by Dr. Donald House at Texas A&M University, Figure 8.11

Why Flip the Kernel?

Example with **no flip**:

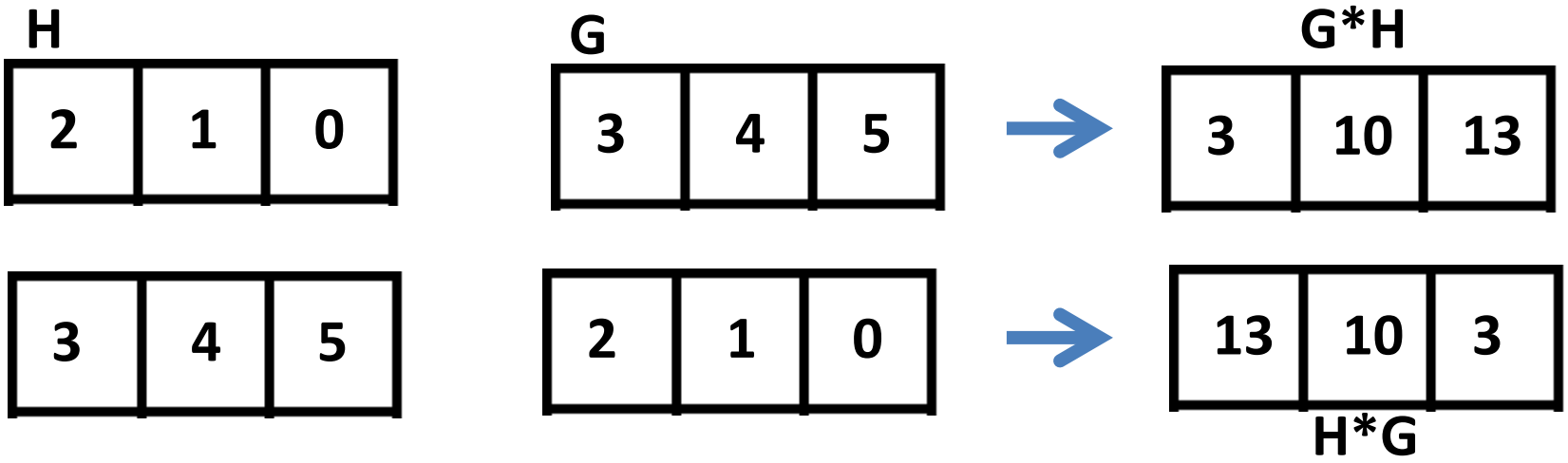


G*H does NOT equal H*G

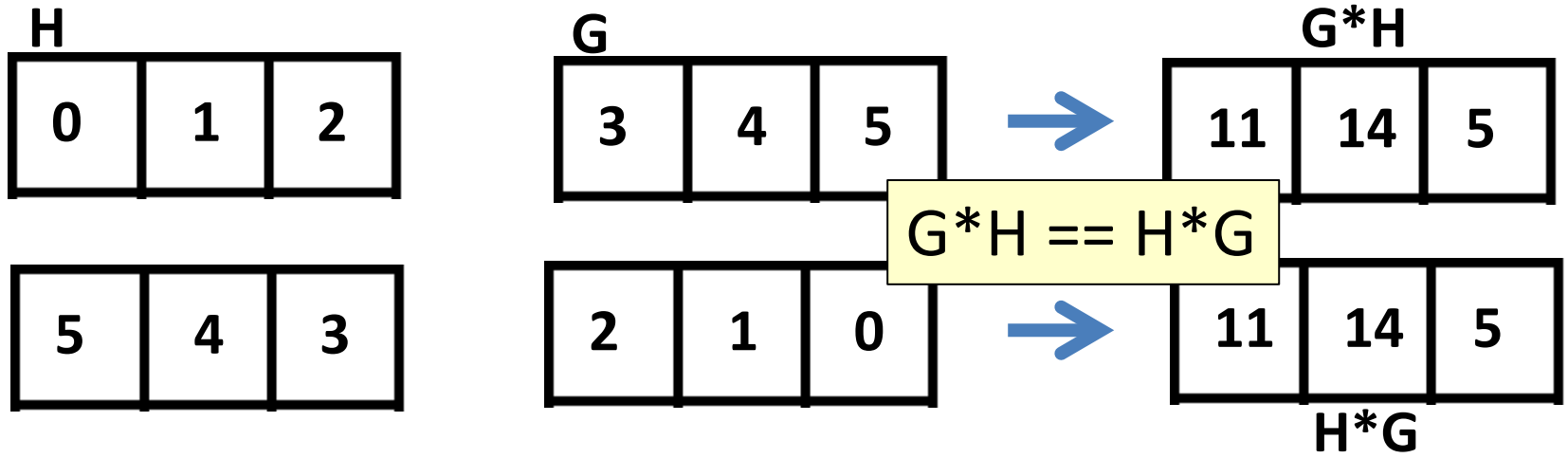
Example from: The Digital Image by Dr. Donald House at Texas A&M University, Figure 8.11

Why Flip the Kernel?

Example with **no flip**:



Example with **flipped kernel**:



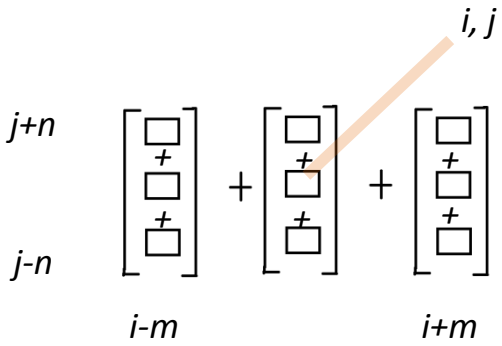
Example from: The Digital Image by Dr. Donald House at Texas A&M University, Figure 8.11

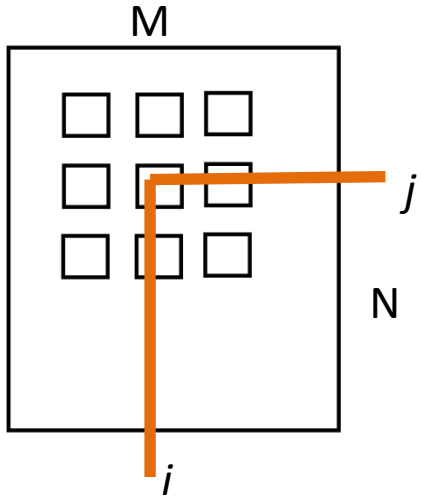
Convolution: 1D to 2D

- Convolution extends to 2D in straightforward fashion
  - Assume kernel H is rectangular array of
    width, ***w=2n+1***
    and height, ***h=2m+1***
  - Assume image G is of width M and height N
  - Then…

$$\hat{G}[i,j] = (G*H)[i,j] = \sum_{l=i-m}^{i+m} \sum_{k=j-n}^{j+n} G[l,k]H[i-l,j-k],\ i \in [0, M-1],\ j \in [0, N-1]$$

i is column
j is row

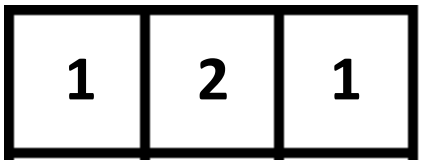


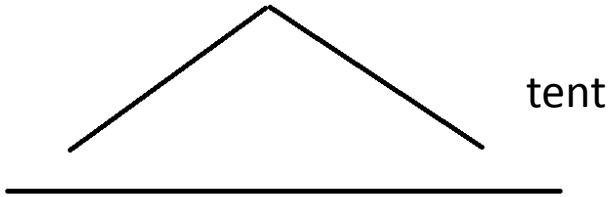


As drawn
w = 3 and h = 3
Thus n = 1 and m = 1

2D kernel from a 1D kernel

$H_2 = H_1 H^t_1$





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

2D kernel

that yields results
similar to 1D kernel

2D Kernel Separability

- 2D kernels can be separated
  - Apply filtering twice using the 1D filter
    - once “horizontally”
    - once “vertically”
  - Get same results as applying the 2D filter once

- Efficiency increase when W large
  - multiplications per pixel = $2W < W^2$
    - e.g. $2*3 < 3^2$.

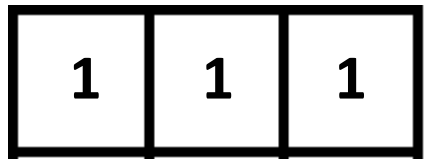
Outline

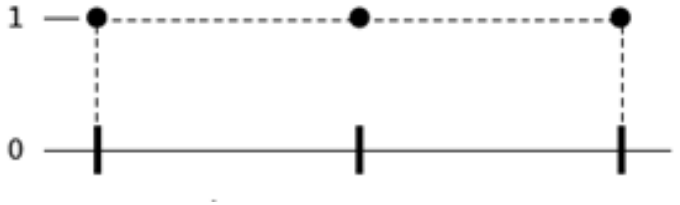
- Filters and Convolutions
  - General Definitions
    - Working around the Edges
  - Low Pass Examples
  - High Pass Examples

Low Pass Filters

- A filter that retains low frequencies and reduces high frequencies
  - Example: box, or pulse, kernel

1D:



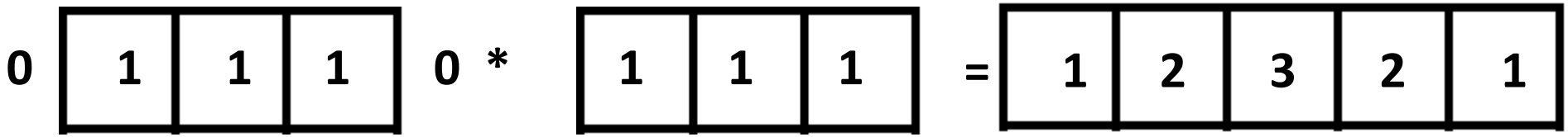


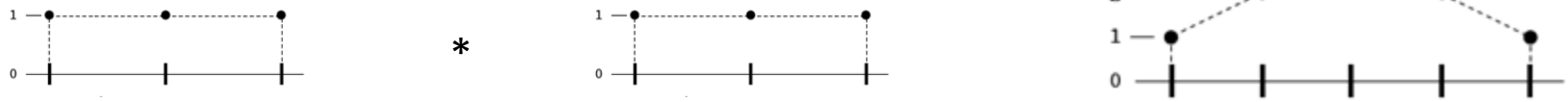
2D:

$$\begin{bmatrix}1\\1\\1\end{bmatrix}\begin{bmatrix}1 & 1 & 1\end{bmatrix}=\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

Box * Box = Tent

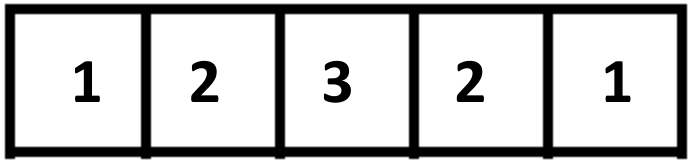
- Convolution of box kernel with itself
  - Gives a tent filter

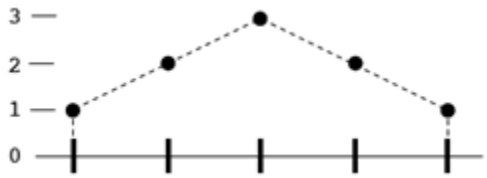




Low Pass Filter: 2D Tent

1D:



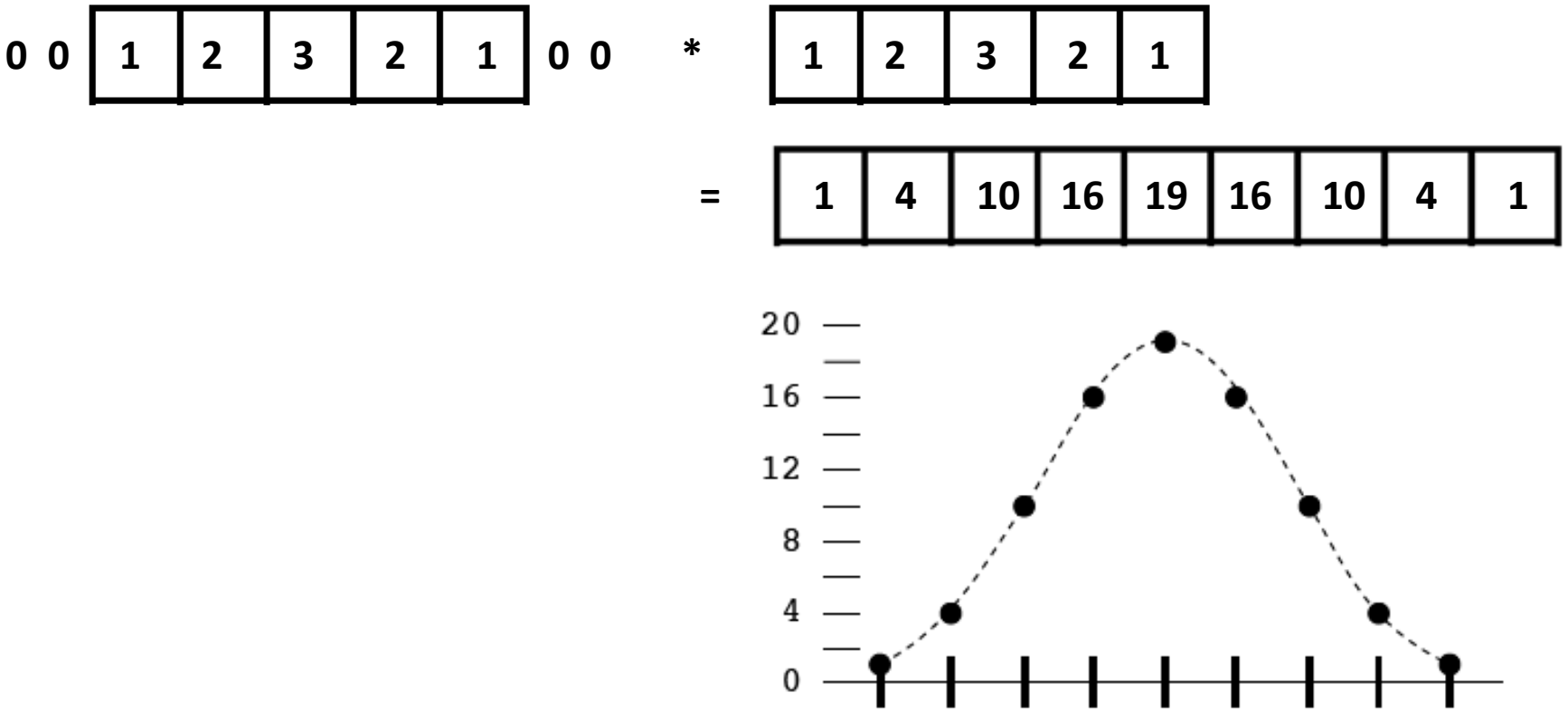


2D:

$$\begin{bmatrix}1\\2\\3\\2\\1\end{bmatrix}\begin{bmatrix}1 & 2 & 3 & 2 & 1\end{bmatrix} = \begin{bmatrix}1 & 2 & 3 & 2 & 1\\2 & 4 & 6 & 4 & 2\\3 & 6 & 9 & 6 & 3\\2 & 4 & 6 & 4 & 2\\1 & 2 & 3 & 2 & 1\end{bmatrix}$$

Tent * Tent = Bell

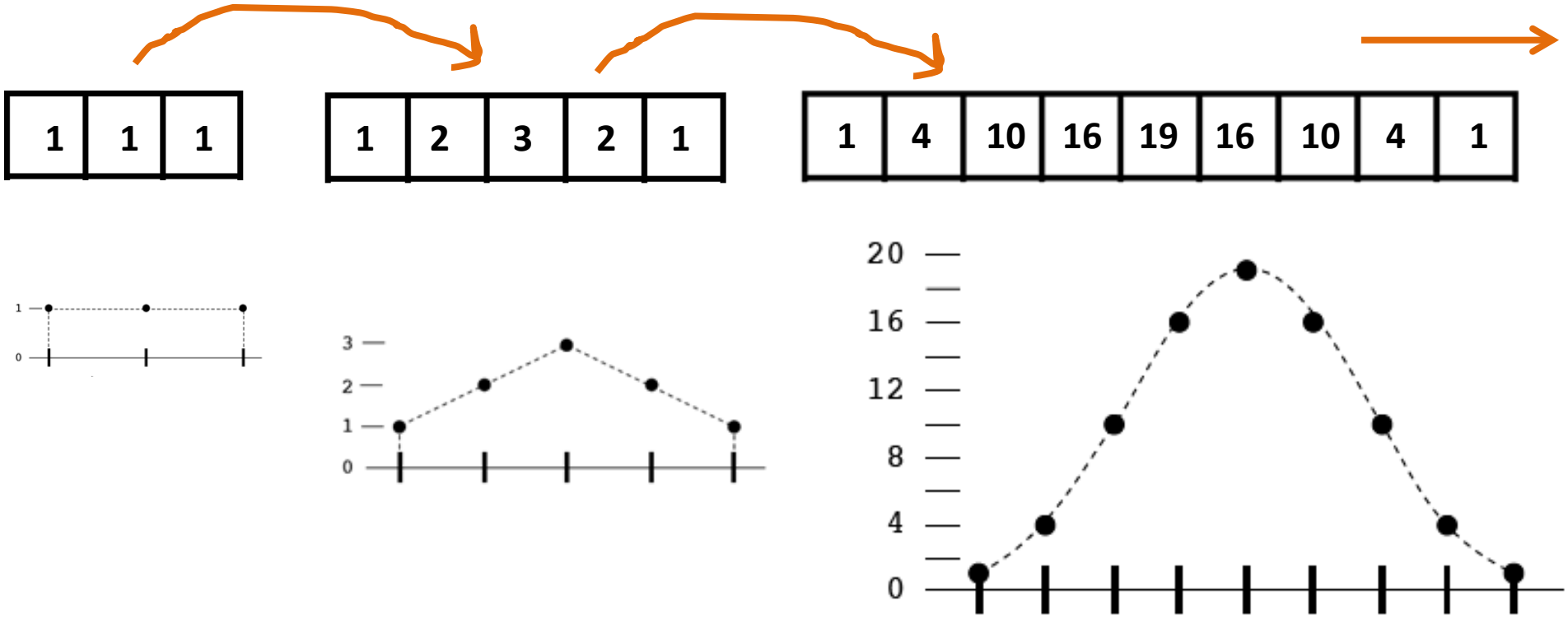
- Convolution of tent kernel with itself
  - Gives a Bell filter



Low Pass Filters

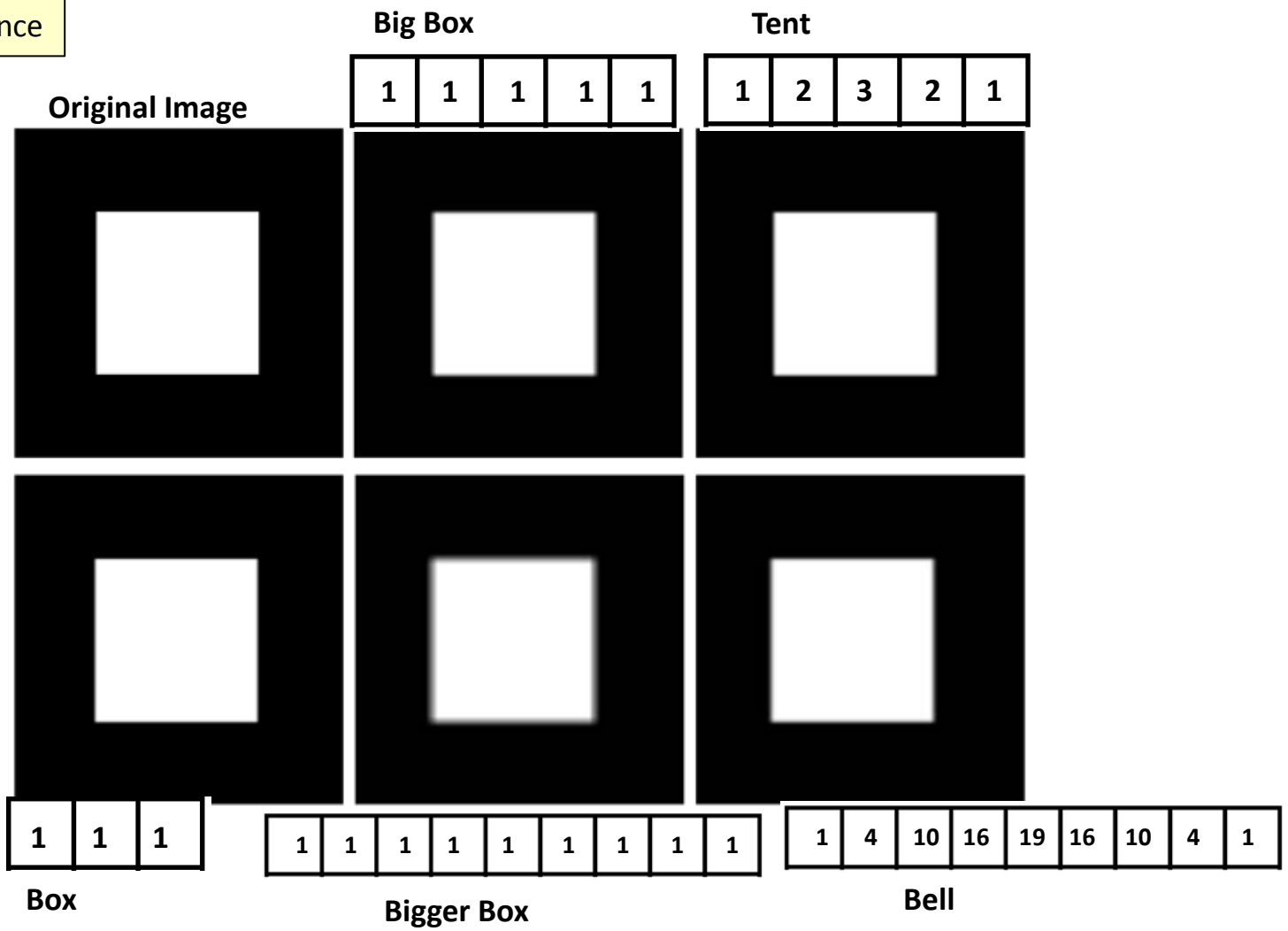
- The more convolutions with self
  - The better smoothing you achieve

continue towards Gaussian



Note:
Tent and Bell
have better quality
than just a “bigger” box
They more cleanly
blur the edges…
…less muddy in appearance

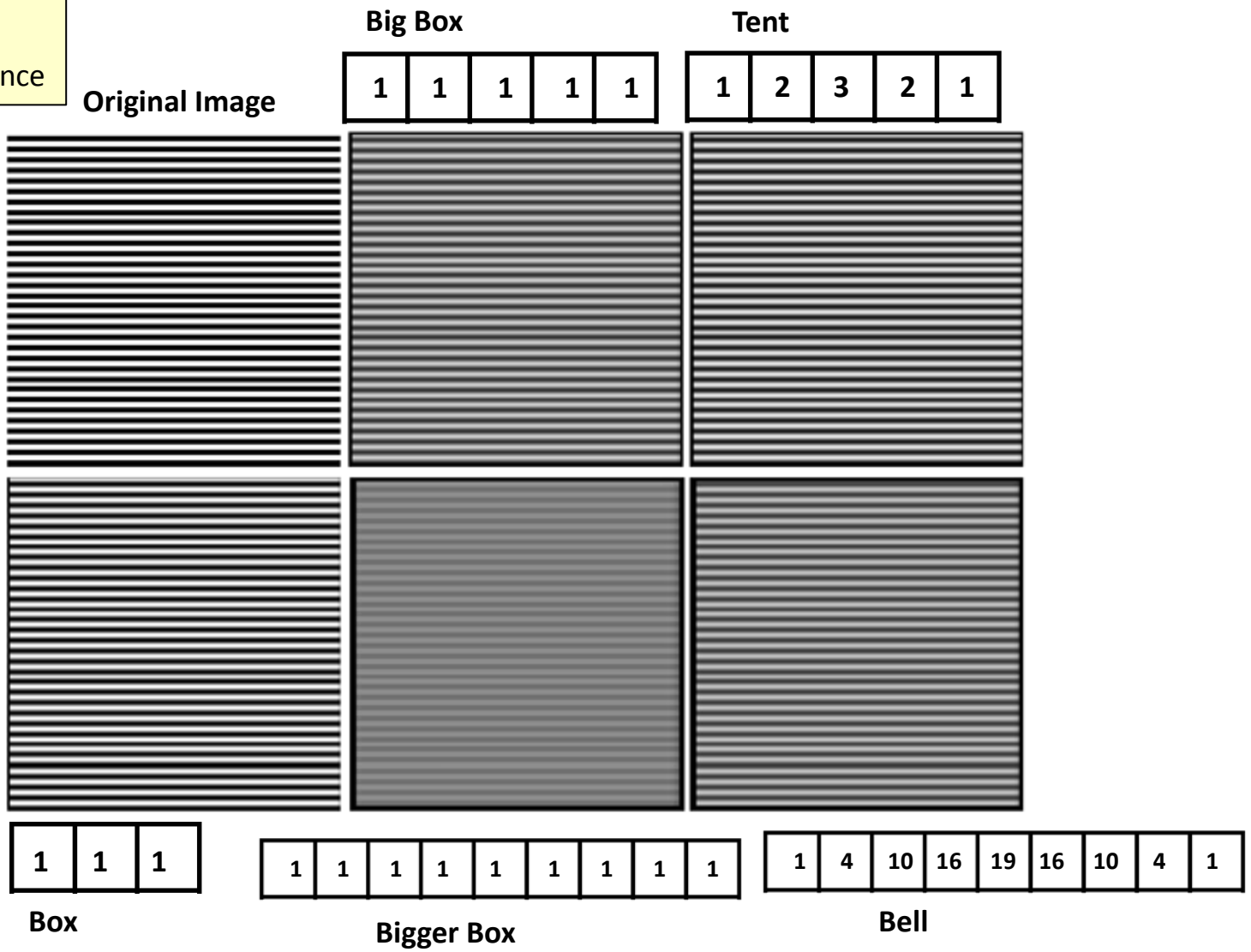
Example Results



Example from: The Digital Image by Dr. Donald House at Texas A&M University, Figure 8.11

Note:
Tent and Bell
have better quality
than just a “bigger” box
They more cleanly
blur the edges…
…less muddy in appearance

Example Results



Example from: The Digital Image by Dr. Donald House at Texas A&M University, Figure 8.11

Outline

- Filters and Convolutions
  - General Definitions
    - Working around the Edges
  - Low Pass Examples
  - High Pass Examples

High Pass Filters

- A filter that retains high frequencies and reduces low frequencies

Example:

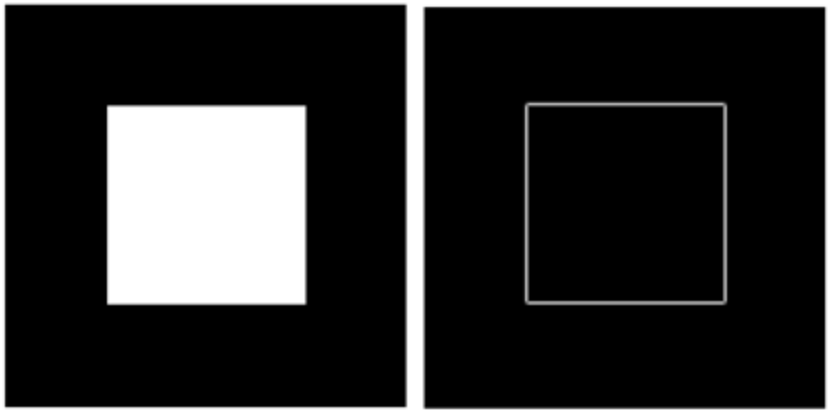
1	1	1
1	-8	1
1	1	1

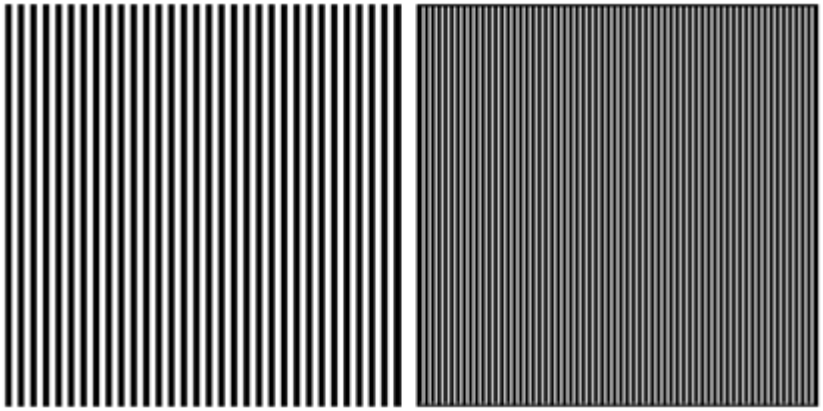
This example suppresses large areas of uniform or smoothly varying color
It will emphasize edges (and other steep changes in color)

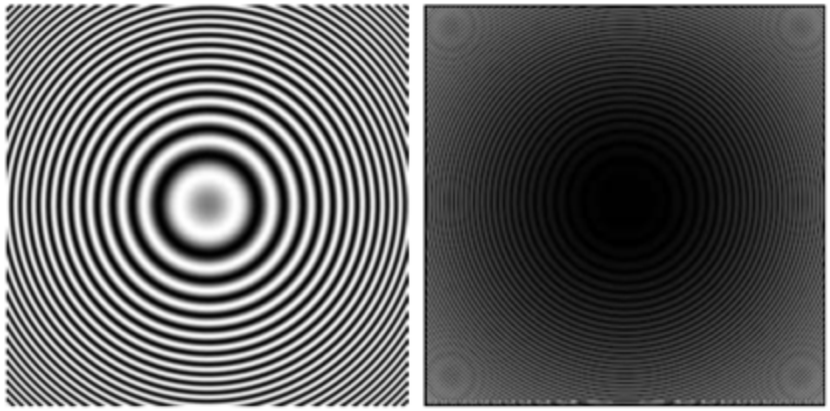
Why?

The sum of the kernel weights is zero
So an area of ‘constant’ color will result in a filtered output of zero

Example Results

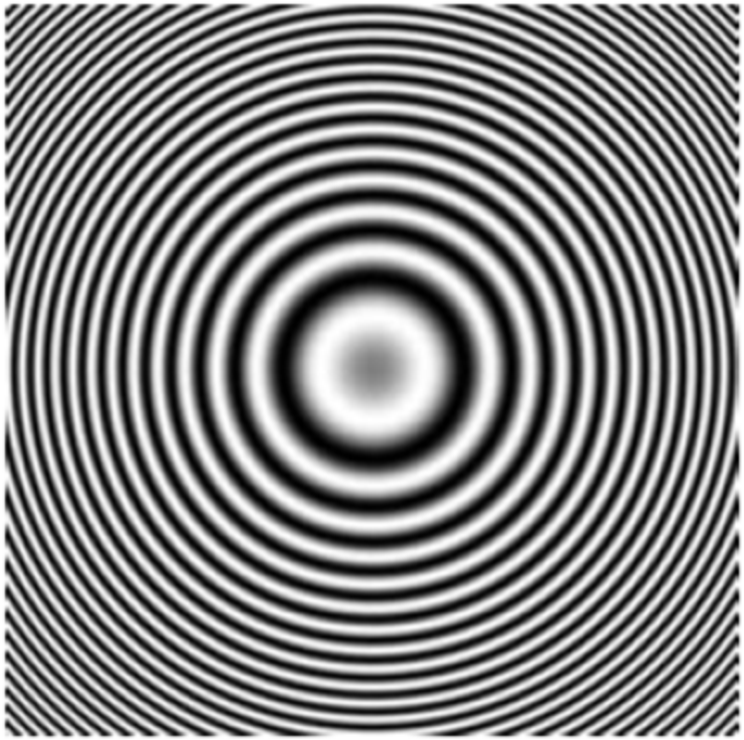


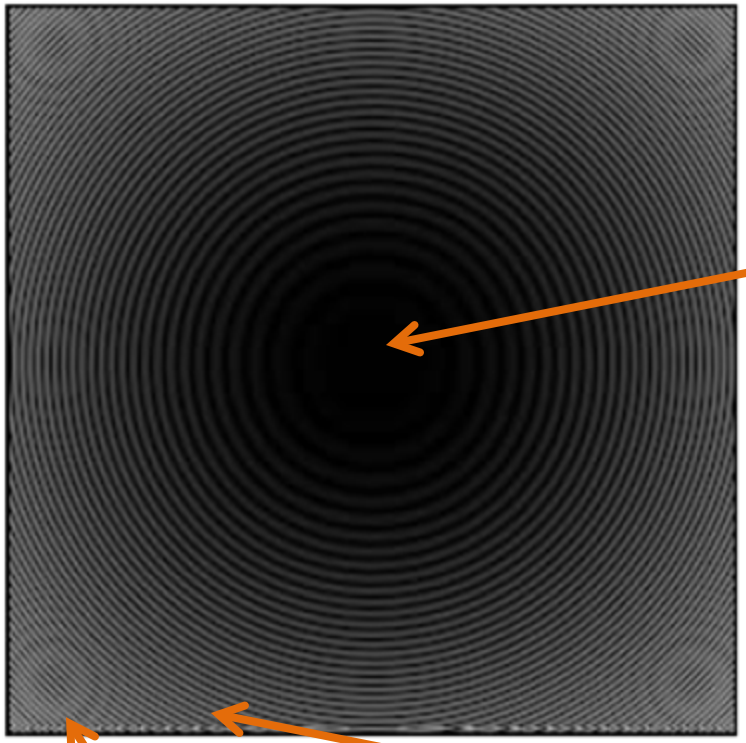




low frequency suppressed – yet high frequency remains

High Pass: Examples Continue





low frequency removed

high frequency remains

Yes you are seeing smaller circles

Why? Remember Moire Patterns?
Aliasing Artifacts?

Summary

- More Details provided on
  - Filters and Convolutions
    - General Definitions
      - Working around the Edges
    - Low Pass Examples
    - High Pass Examples

- Should help in implementation and detailed understanding

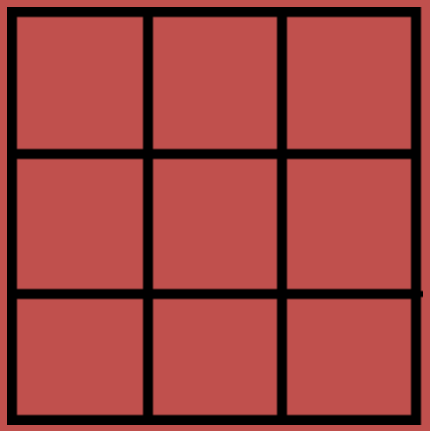
- Go try doing these things! =)

Questions?

- Beyond D2L
  - Examples and information can be found online at:
    - *http://docdingle.com/teaching/cs.html*

    - *Continue to more stuff as needed*

Extra Reference Stuff Follows



Credits

- Much of the content derived/based on slides for use with the book:
  - *Digital Image Processing,* Gonzalez and Woods

- Some layout and presentation style derived/based on presentations by
  - Donald House, Texas A&M University, 1999
  - Bernd Girod, Stanford University, 2007
  - Shreekanth Mandayam, Rowan University, 2009
  - Igor Aizenberg, TAMUT, 2013
  - Xin Li, WVU, 2014
  - George Wolberg, City College of New York, 2015
  - Yao Wang and Zhu Liu, NYU-Poly, 2015
  - Sinisa Todorovic, Oregon State, 2015