Image Processing I

Introduction to Computational Photography:

EECS 395/495

Northwestern University

Image Processing I

Transform image to new one that is easier to manipulate.

Topics:

(1) Pixel Processing

(2) Convolution

(3) Linear Filtering

(4) Non-Linear Filtering

(5) Correlation

Lecture 1

Image Processing II

Transform image to new one that is easier to manipulate.

Topics:

(6) Frequency Representation of Signals

(7) Fourier Transform

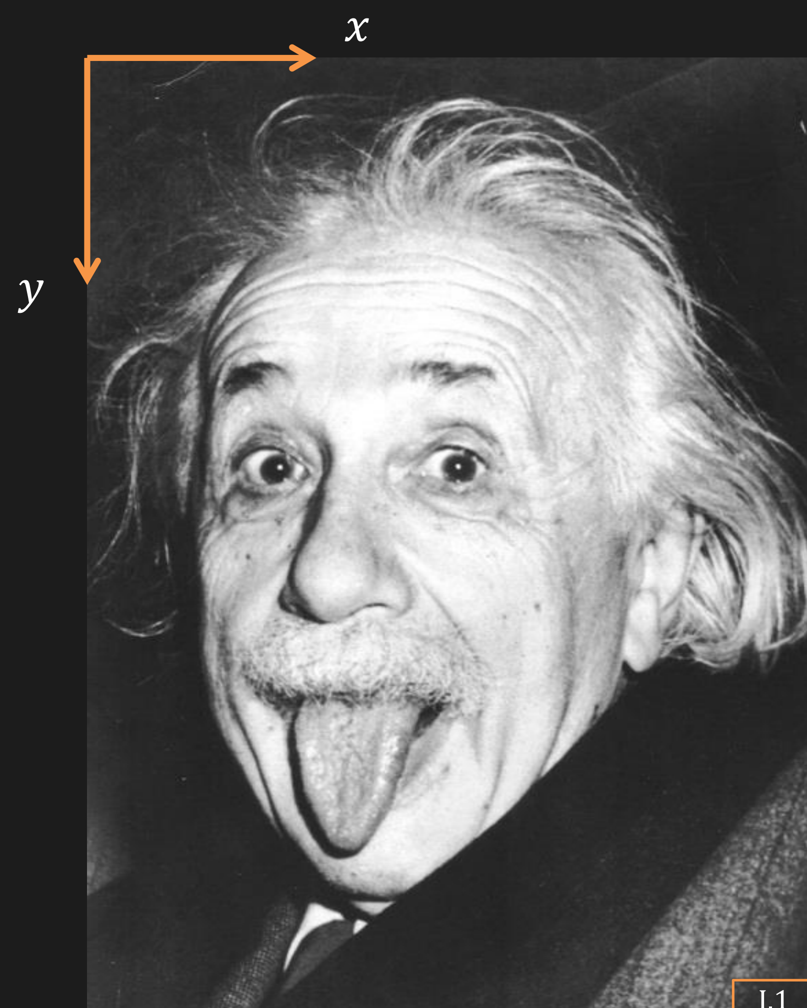
(8) Convolution $\leftrightarrow$ Fourier Transform

(9) Deconvolution in Frequency Domain

(10) Sampling Theory

Lecture 2

Image as a Function



$f(x, y)$ is the image intensity at position (x, y)

Image Processing

Transformation t of one image f to another image g

$$g(x, y) = t(f(x, y))$$

Point (Pixel) Processing

Darken $(f - 128)$

Original (f)

Lighten $(f + 128)$

Invert $(255 - f)$

Point (Pixel) Processing

Original (f)

Low Contrast $(f/2)$

High Contrast $(f * 2)$

Gray
$(0.3f_R + 0.6f_G + 0.1f_B)$

Linear Shift Invariant System

$$f(x) \longrightarrow \boxed{\text{LSIS}} \longrightarrow g(x)$$

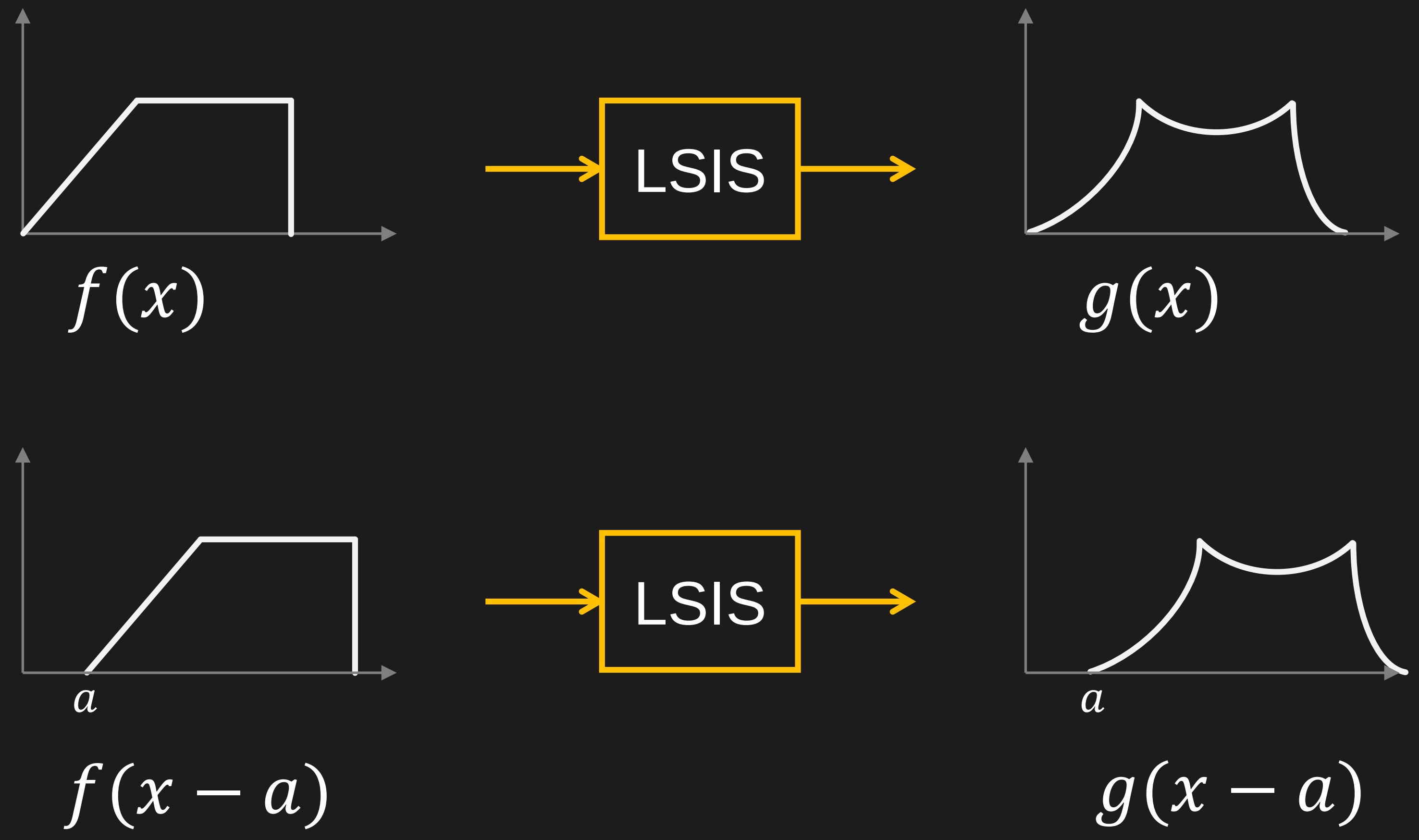
Study of Linear Shift Invariant Systems (LSIS) leads to useful image processing algorithms.

LSIS: Linearity

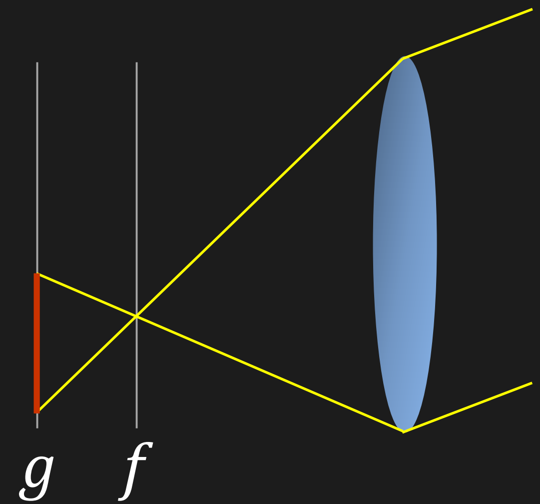
$$f_1 \longrightarrow \boxed{\text{LSIS}} \longrightarrow g_1 \qquad\qquad f_2 \longrightarrow \boxed{\text{LSIS}} \longrightarrow g_2$$

$$\alpha f_1 + \beta f_2 \longrightarrow \boxed{\text{LSIS}} \longrightarrow \alpha g_1 + \beta g_2$$

LSIS: Shift Invariance



Ideal Lens is an LSIS



Defocused Image (g): Processed version of Focused Image (f)

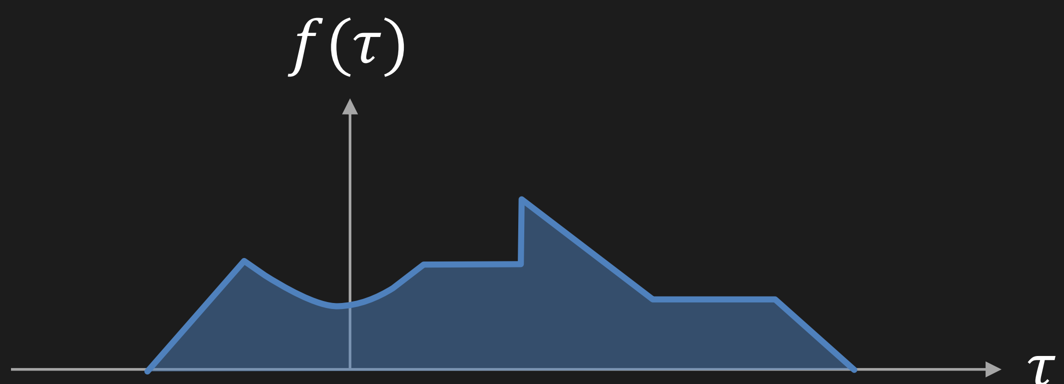
Linearity: Brightness variation

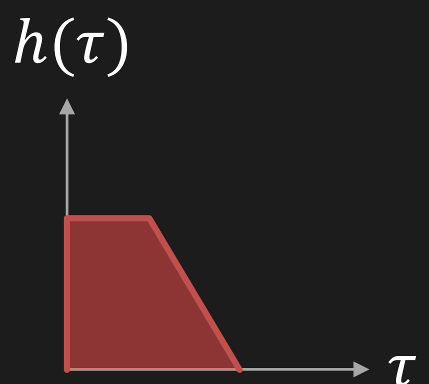
Shift invariance: Scene movement

Convolution (Important Concept)

Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

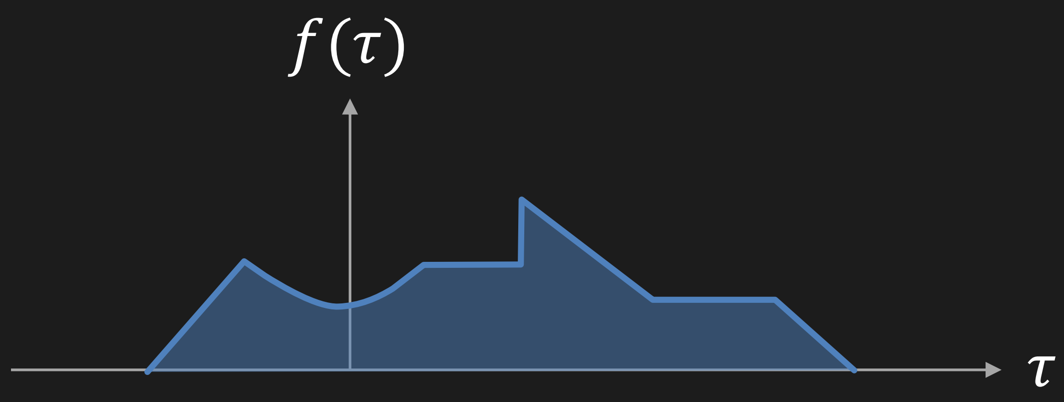


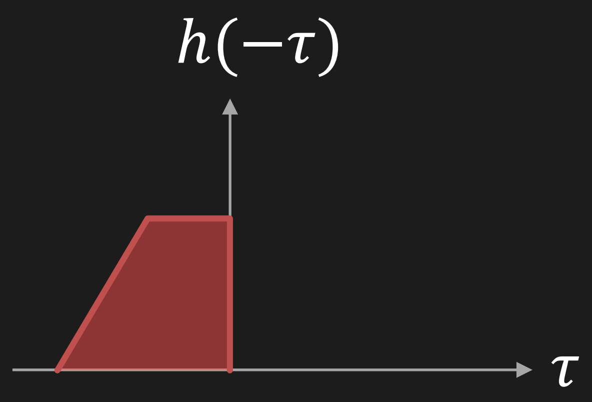


Convolution (Important Concept)

Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

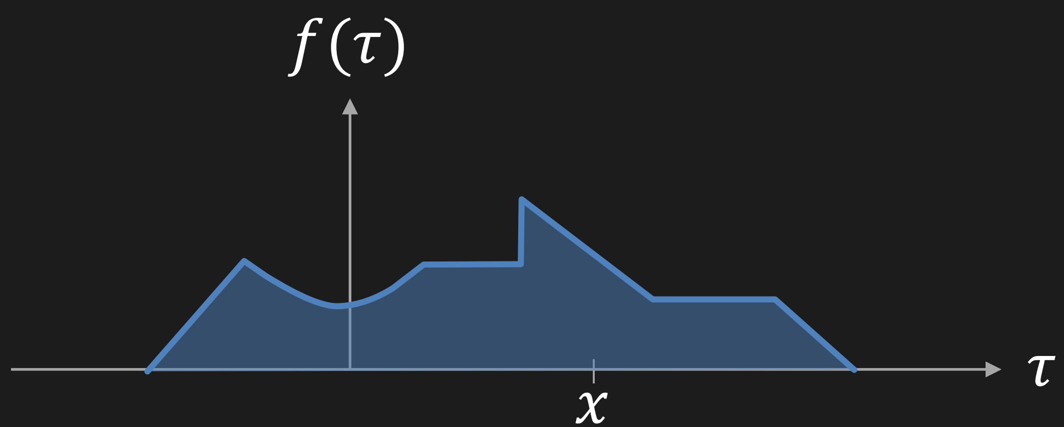


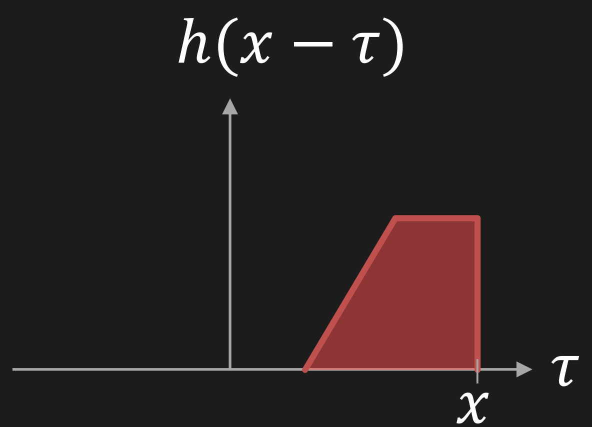


Convolution (Important Concept)

Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

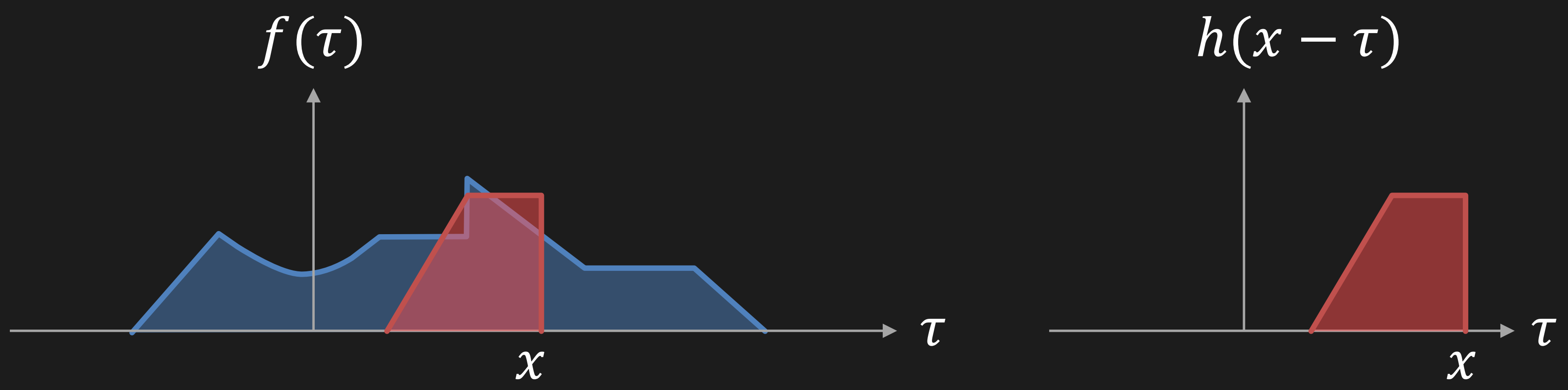




Convolution (Important Concept)

Convolution of two functions $f(x)$ and $h(x)$

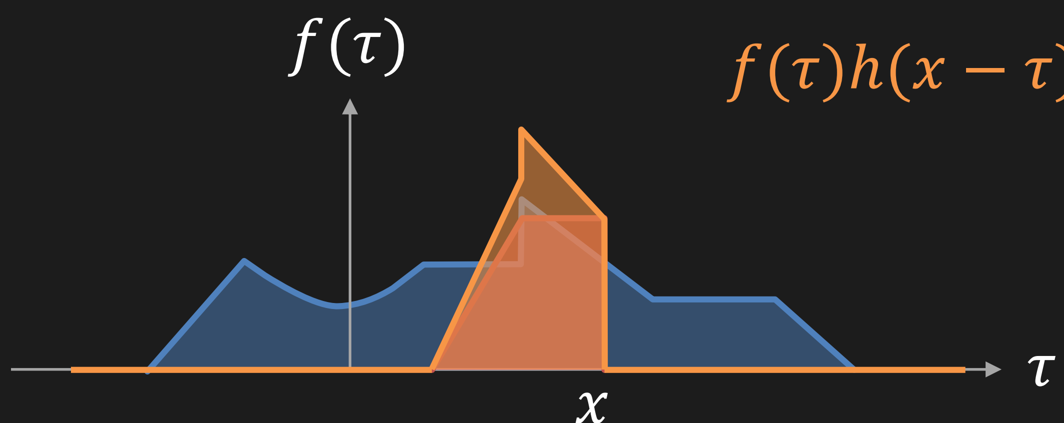
$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

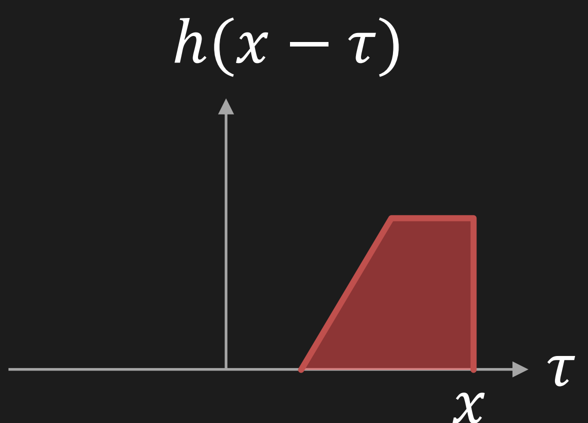


Convolution (Important Concept)

Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

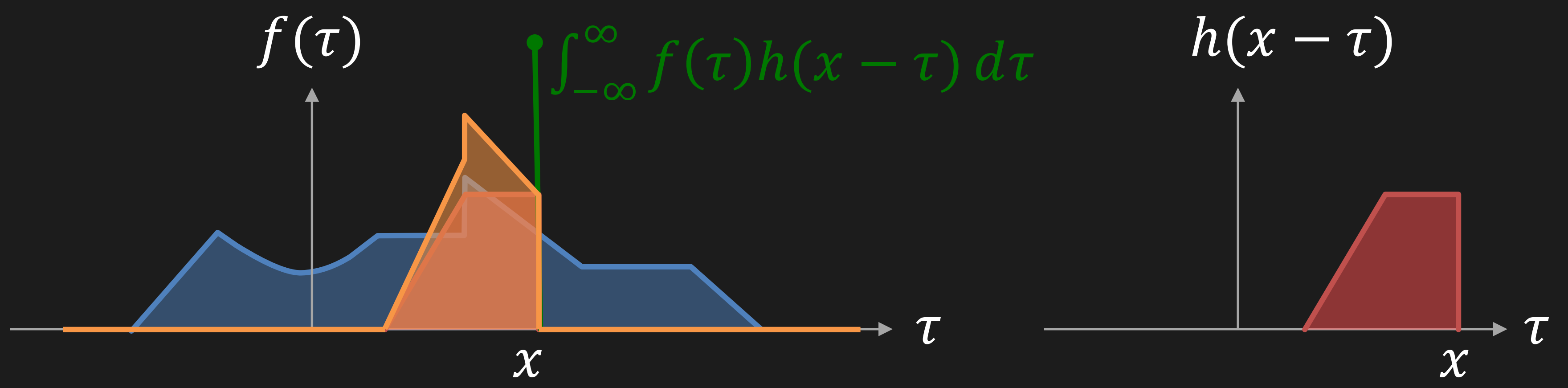




Convolution (Important Concept)

Convolution of two functions $f(x)$ and $h(x)$

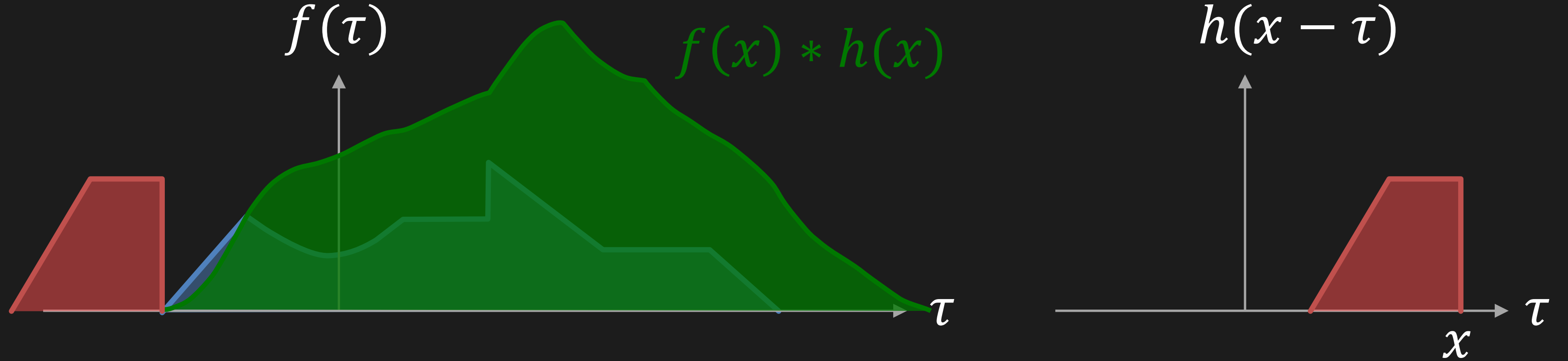
$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$



Convolution (Important Concept)

Convolution of two functions $f(x)$ and $h(x)$

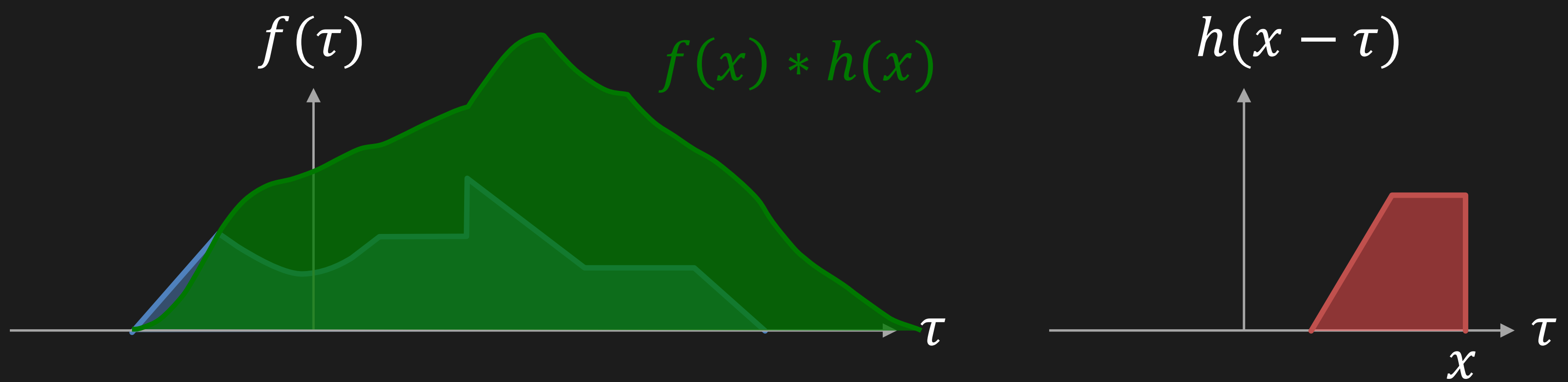
$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$



Convolution (Important Concept)

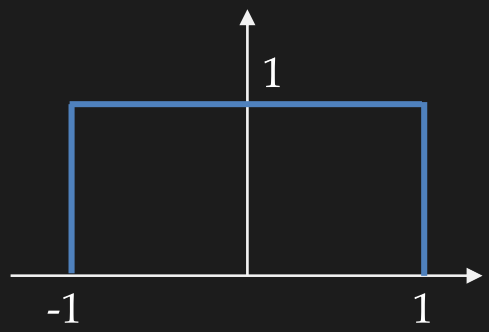
Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\, d\tau}$$

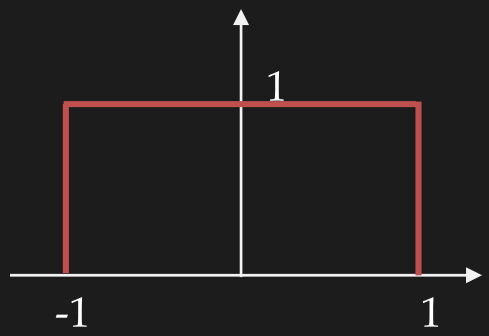


LSIS implies Convolution and Convolution implies LSIS

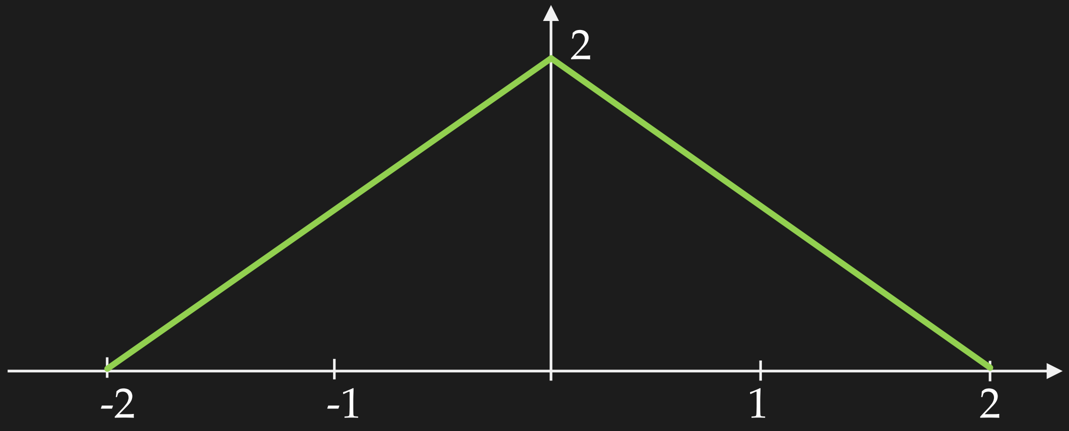
Convolution: Example



$f(x)$

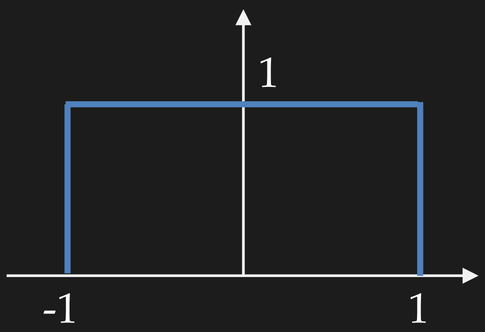


$h(x)$

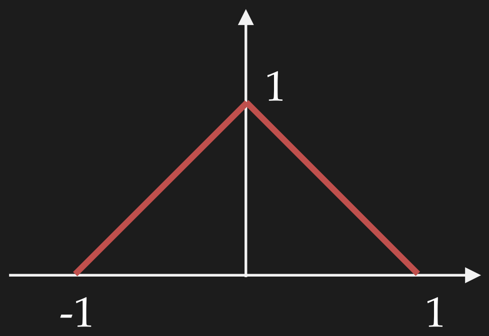


$f(x) * h(x)$

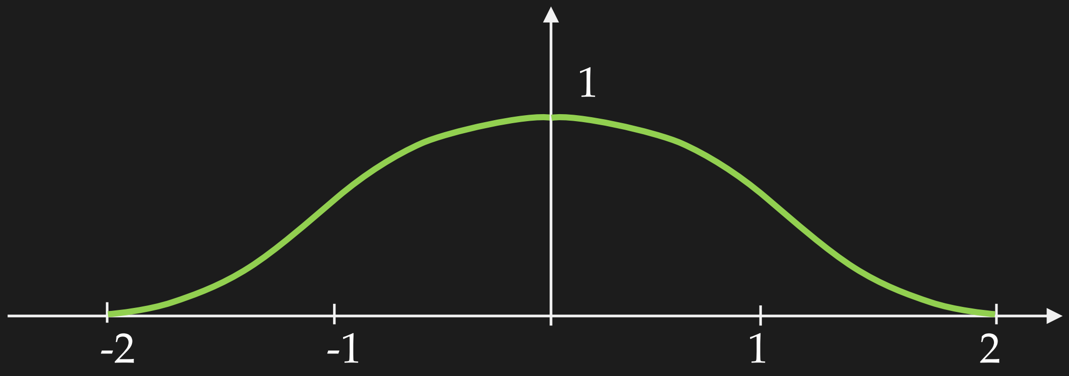
Convolution: Example



$f(x)$

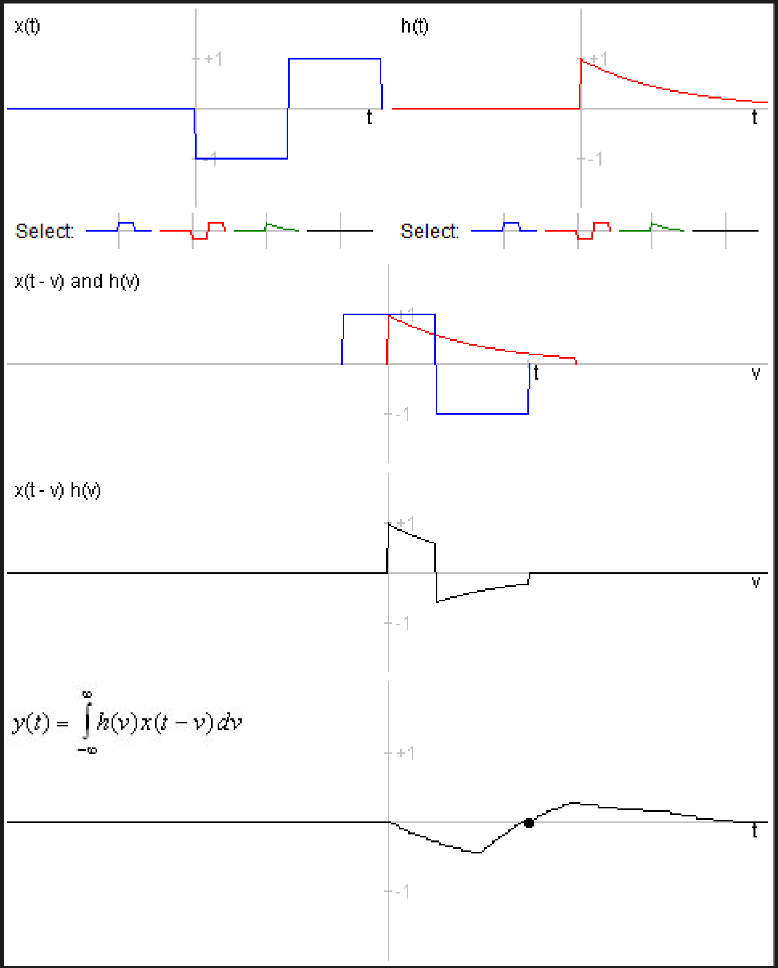


$h(x)$



$f(x) * h(x)$

Convolution: Online Demo



http://www.jhu.edu/signals/convolve/

Can we find h?

$$f \longrightarrow \boxed{h} \longrightarrow g \qquad g(x) = \int_{-\infty}^{\infty} f(\tau) h(x - \tau)\, d\tau$$

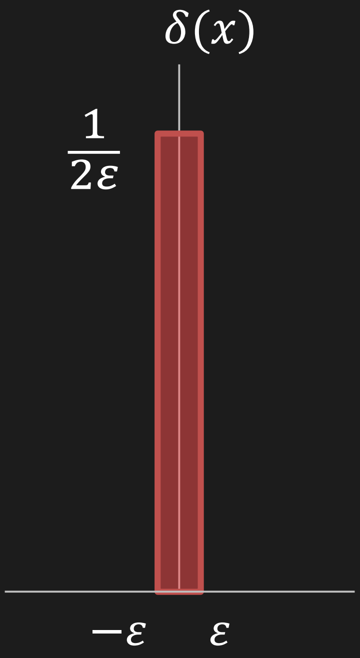
What input f will produce output $g = h$?

$$h(x) = \int_{-\infty}^{\infty} ?\,(\tau) h(x - \tau)\, d\tau$$

Unit Impulse Function

$$\delta(x) = \begin{cases} 1/2\varepsilon\,, & |x| \leq \varepsilon \\ 0, & |x| > \varepsilon \end{cases}$$

$$\varepsilon \to 0$$



$$\int_{-\infty}^{\infty} \delta(\tau)\, d\tau = \frac{1}{2\varepsilon} . 2\varepsilon = 1$$

$$\boxed{\int_{-\infty}^{\infty} \delta(\tau) b(x-\tau)\, d\tau = b(x)}$$

Sifting Property

Impulse Response

$$f \longrightarrow \boxed{h} \longrightarrow g$$

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

$$g(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$$

$$\delta \longrightarrow \boxed{h} \longrightarrow h$$

Unit Impulse

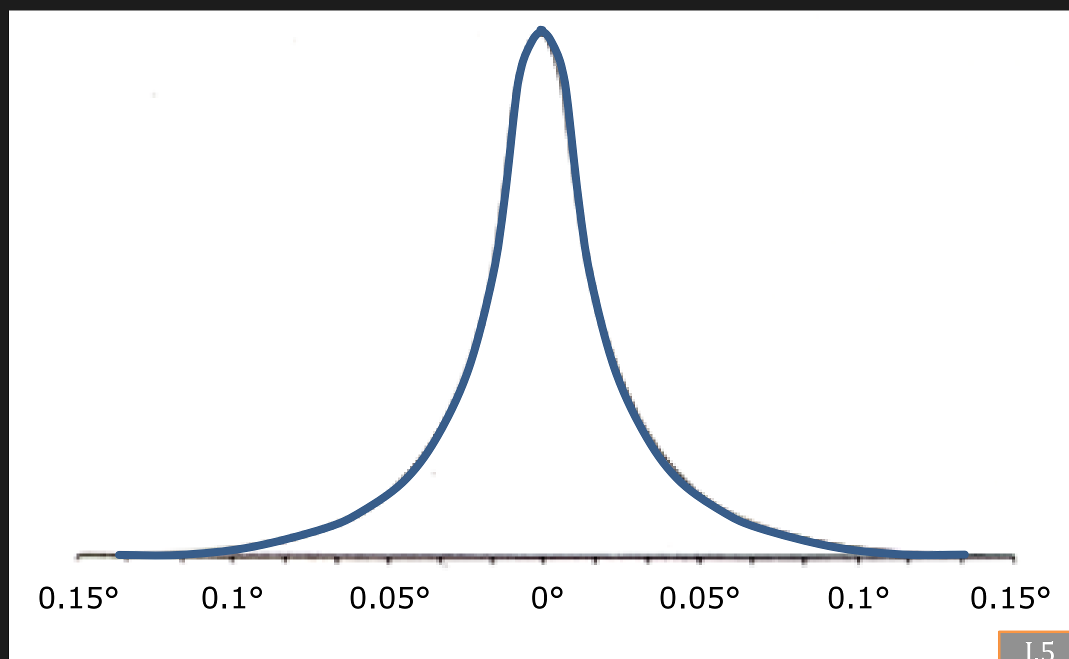
Impulse Response

$$h(x) = \delta(x) * h(x)$$

$$h(x) = \int_{-\infty}^{\infty} \delta(\tau)h(x-\tau)\,d\tau$$

Impulse Response of Human Eye

Distant Star $\delta(x, y)$ → Human Eye → Point Spread Function (PSF) $h(x, y)$



Human Eye PSF

Properties of Convolution

Commutative $a * b = b * a$

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

Cascaded System

$$f \longrightarrow \boxed{h_1} \longrightarrow \boxed{h_2} \longrightarrow g$$

$$\equiv \quad f \longrightarrow \boxed{h_1 * h_2} \longrightarrow g$$

$$\equiv \quad f \longrightarrow \boxed{h_2 * h_1} \longrightarrow g$$

2D Convolution

LSIS:

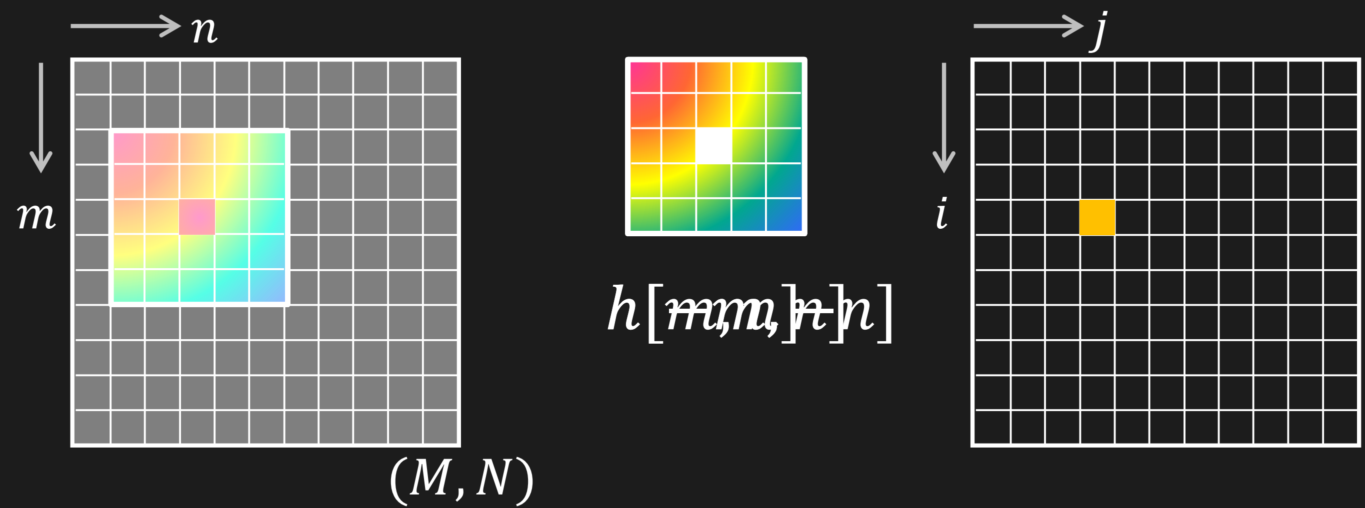
$$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

Convolution:

$$g(x,y) = \int_{-\infty}^{\infty} f(\tau,\mu)h(x-\tau, y-\mu)\, d\tau d\mu$$

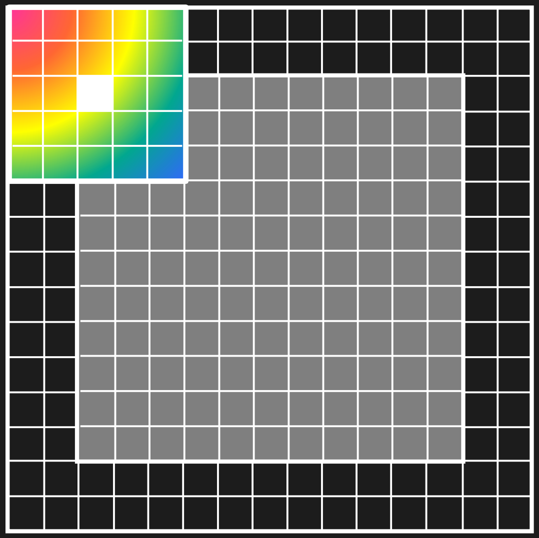
Convolution with Discrete Images

$$f[m,n] \longrightarrow \boxed{h[m,n]} \longrightarrow g[i,j]$$



$$g[i,j] = \sum_{m=1}^{M} \sum_{n=1}^{N} f[m,n]\,h[i-m, j-n]$$

Border Problem

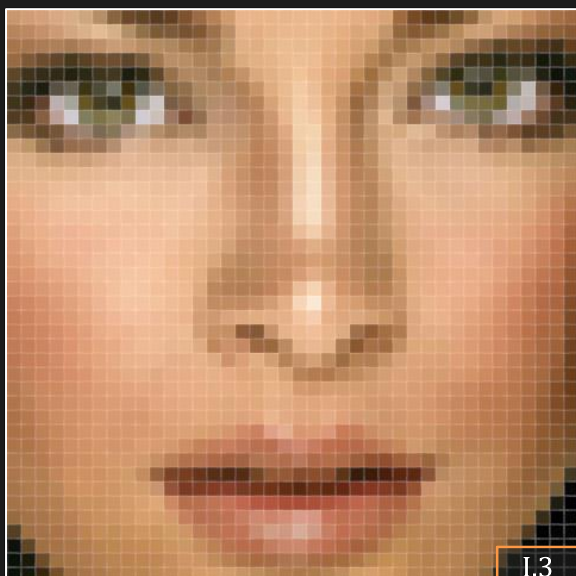


Solution:

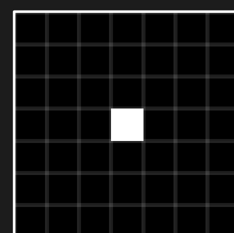
- Ignore Border
- Pad with Constant Value
- Pad with Reflection

Example: Impulse Filter

Input

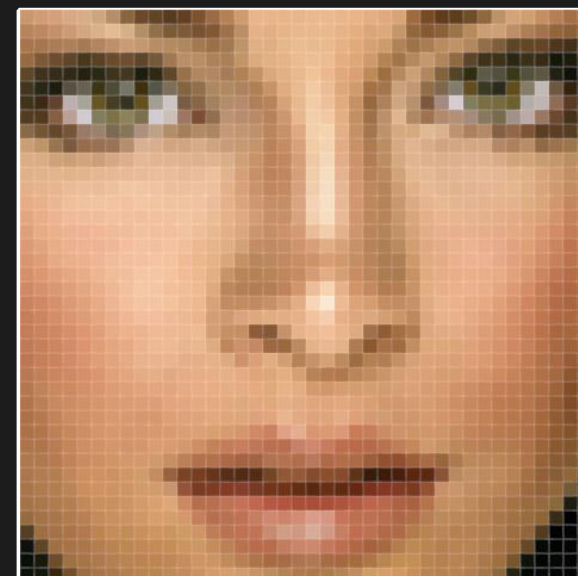


$*$



$=$

Output



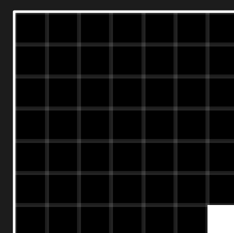
$f(x, y)$ $\qquad \delta(x, y)$ $\qquad f(x, y)$

Example: Image Shift

Input

$f(x, y)$

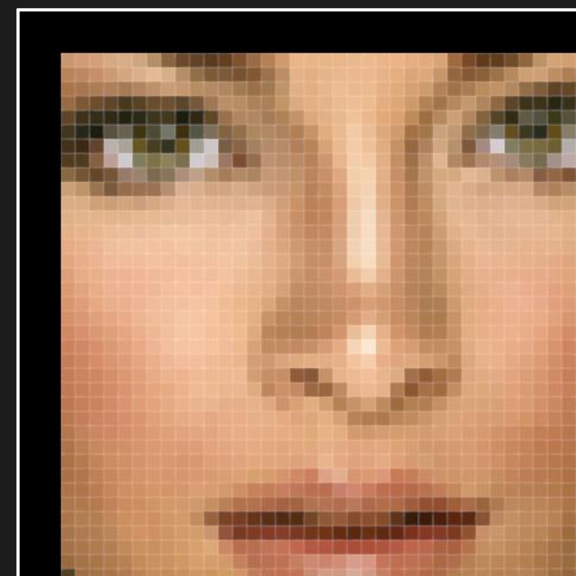
$*$



$\delta(x-u, y-v)$

$=$

Output

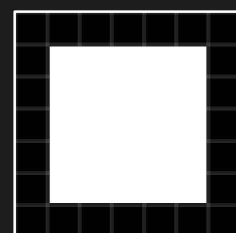


$f(x-u, y-v)$

Example: Averaging

Input

$*$

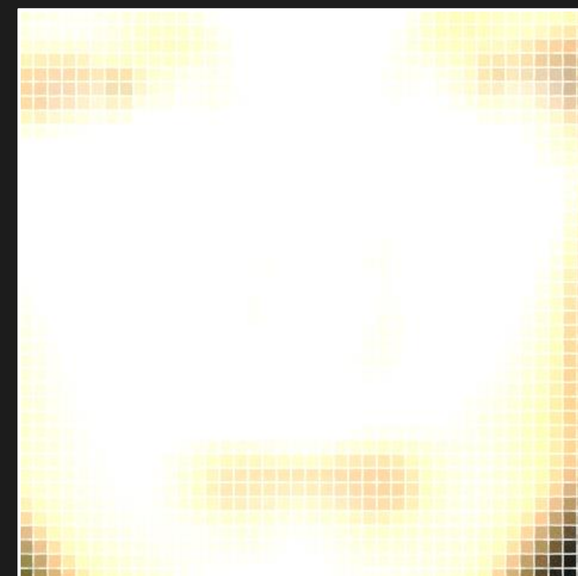


“Box Filter”

5 x 5

$=$

Output



$f(x,y)$ $a(x,y)$ $g(x,y)$

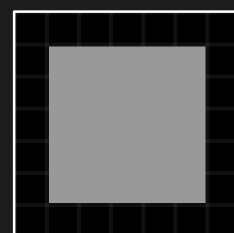
Result Image is Saturated. Why?

Example: Averaging

Input

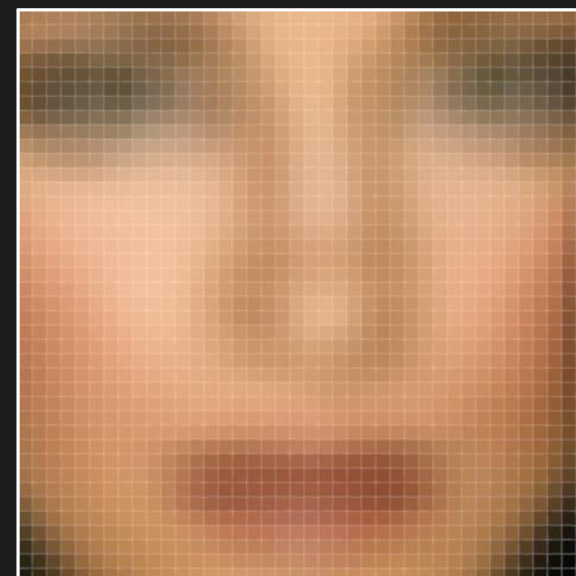
Output

$*$



"Box Filter"
5 x 5

$=$



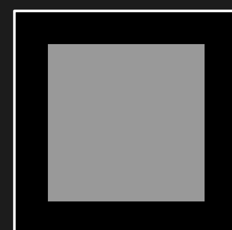
$f(x, y)$ $a(x, y)$ $g(x, y)$

Sum of all the Filter (Kernel) Weights should be 1.

Smoothing With Box Filter

Input

$*$



“Box Filter”
21 x 21

$=$

Output

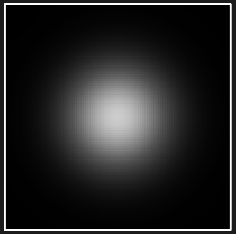
$f(x,y)$ $\quad$ $a(x,y)$ $\quad$ $g(x,y)$

Image smoothed with a box filter does not look “natural.”
Has blocky artifacts.

Smoothing With “Fuzzy” Filter

Input

$*$



“Fuzzy Filter”
21 x 21

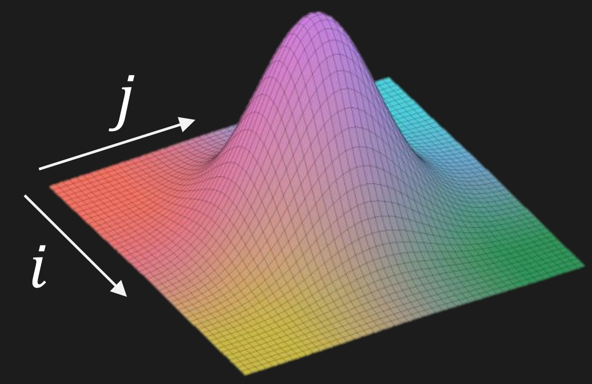
$=$

Output

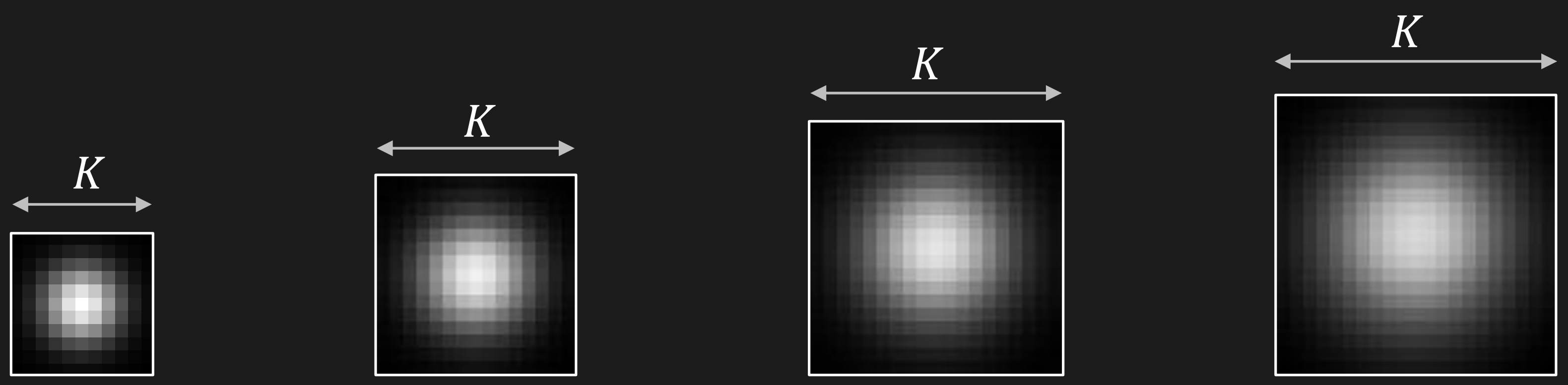
$f(x,y)$ $b(x,y)$ $g(x,y)$

Gaussian Kernel: A Fuzzy Filter

$$n_\sigma[i,j] = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left(\frac{i^2+j^2}{\sigma^2}\right)}$$



σ^2: Variance



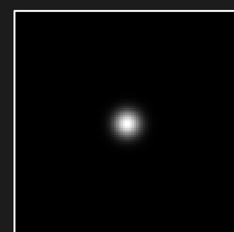
$\sigma = 2$ $\sigma = 3$ $\sigma = 4$ $\sigma = 5$

Rule of Thumb: Set Kernel Size $K \approx 2\pi\sigma$

Gaussian Smoothing

Input

$*$



$\sigma = 4$

$=$

Output

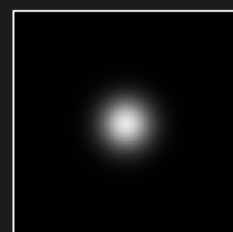
$f(x, y)$ $\quad$ $n_4(x, y)$ $\quad$ $g(x, y)$

Larger the Kernel (or σ), More the Blurring

Gaussian Smoothing

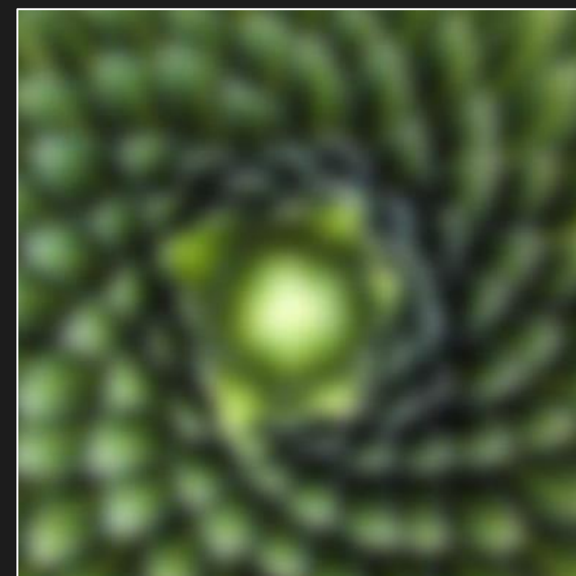
Input

$*$



$=$

Output



$\sigma = 8$

$f(x,y)$ $\quad$ $n_8(x,y)$ $\quad$ $g(x,y)$

Larger the Kernel (or σ), More the Blurring

Gaussian Smoothing

Input

Output

$*$

$=$

$\sigma = 16$

$f(x, y)$

$n_{16}(x, y)$

$g(x, y)$

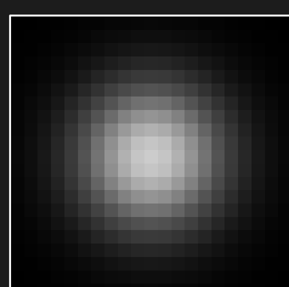
Larger the Kernel (or σ), More the Blurring

Gaussian Smoothing is Separable

$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)} f[i-m, j-n]$$

$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2}{\sigma^2}\right)} \cdot \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{n^2}{\sigma^2}\right)} f[i-m, j-n]$$

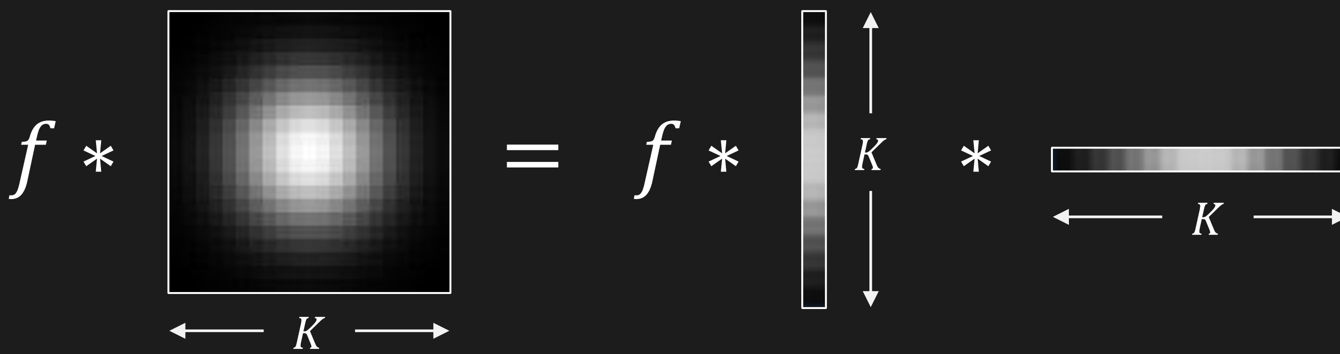
Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters

$f\ *$  $=\ f\ *$  K $*$ 

K K

Gaussian Smoothing is Separable

Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters



Which one is faster? Why?

K^2 Multiplications	$2K$ Multiplications
$K^2 - 1$ Additions	$2(K-1)$ Additions

Smoothing to Remove Image Noise

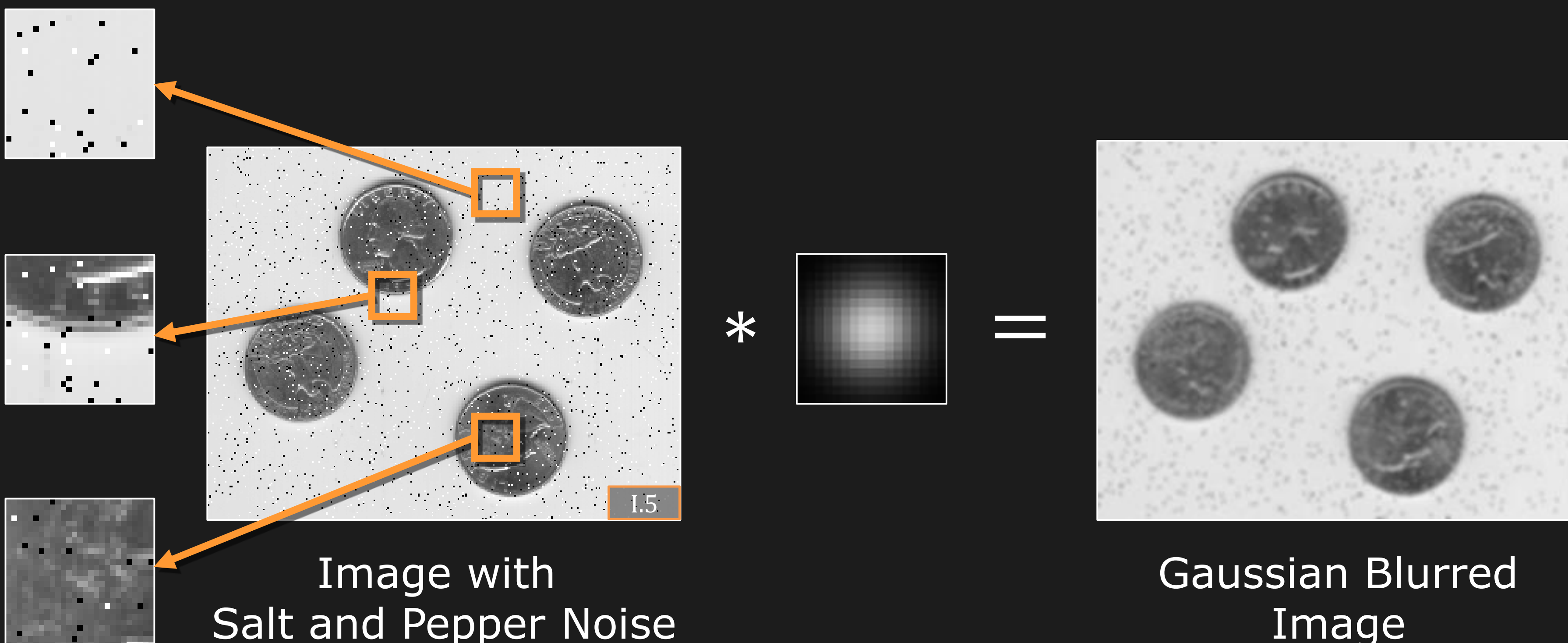


Image with
Salt and Pepper Noise

Gaussian Blurred
Image

Problem with Smoothing:

- Sensitive to Outliers (Noise)
- Smoothens Edges (Blur)

Median Filtering

1. Sort the K^2 values in window centered at the pixel
2. Assign the Middle value (Median) to pixel

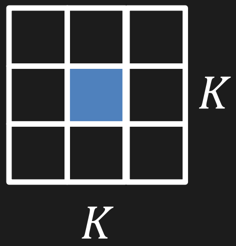


Image with
Salt and Pepper Noise

Median Filtered
Image ($K = 3$)

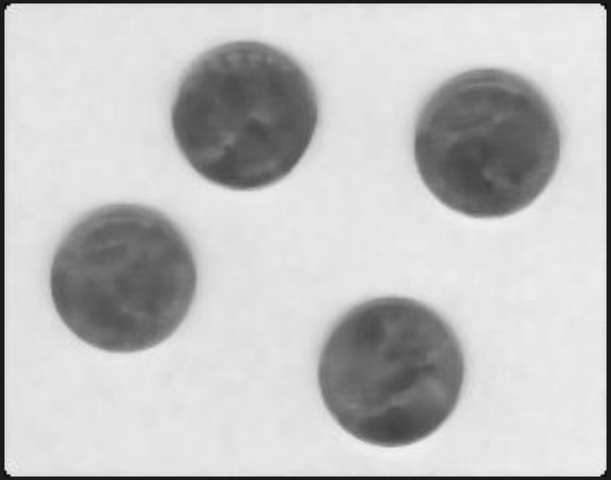
Non-linear Operation

(Cannot be implemented using Convolution)

Median Filtering

Not Effective when Image Noise is not a Simple Salt and Pepper Noise.

Image with Noise



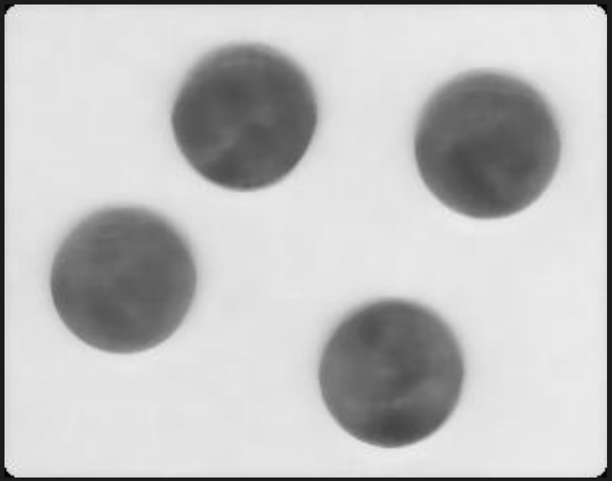
Median Filtered Image ($K = 7$)

Larger K causes Blurring of Image Detail

Median Filtering

Not Effective when Image Noise is not a Simple Salt and Pepper Noise.

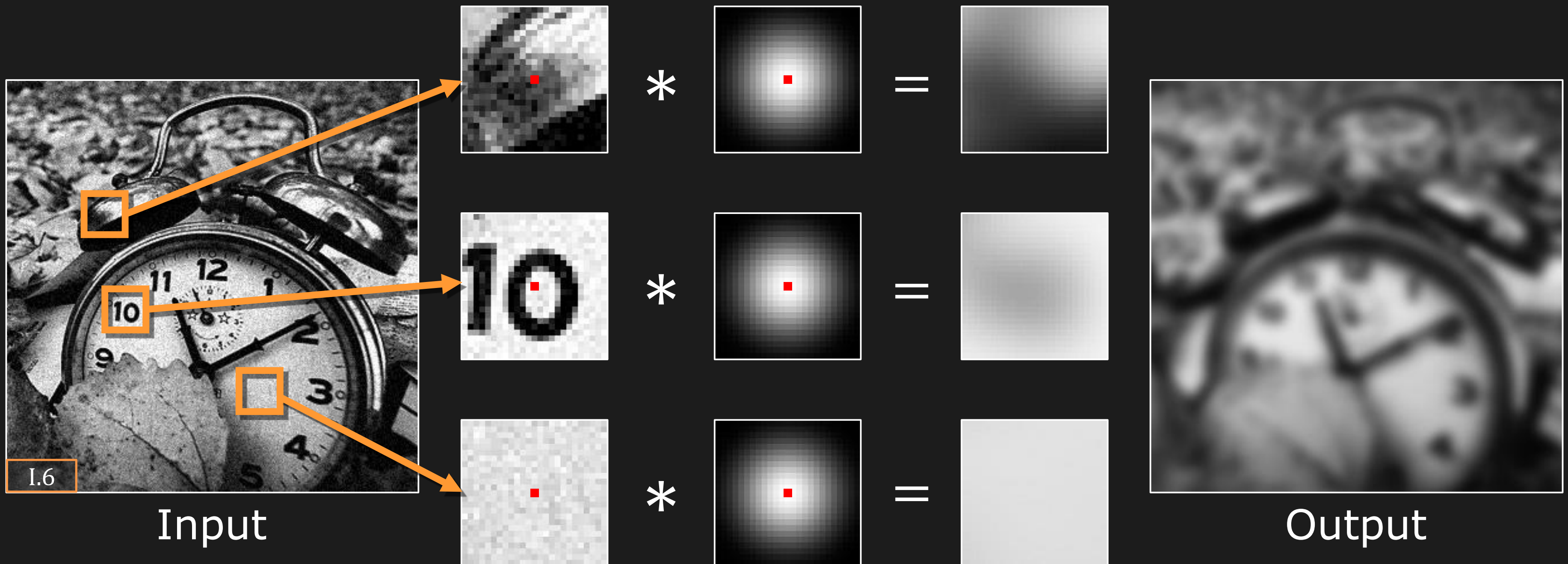
Image with Noise



Median Filtered Image ($K = 11$)

Larger K causes Blurring of Image Detail

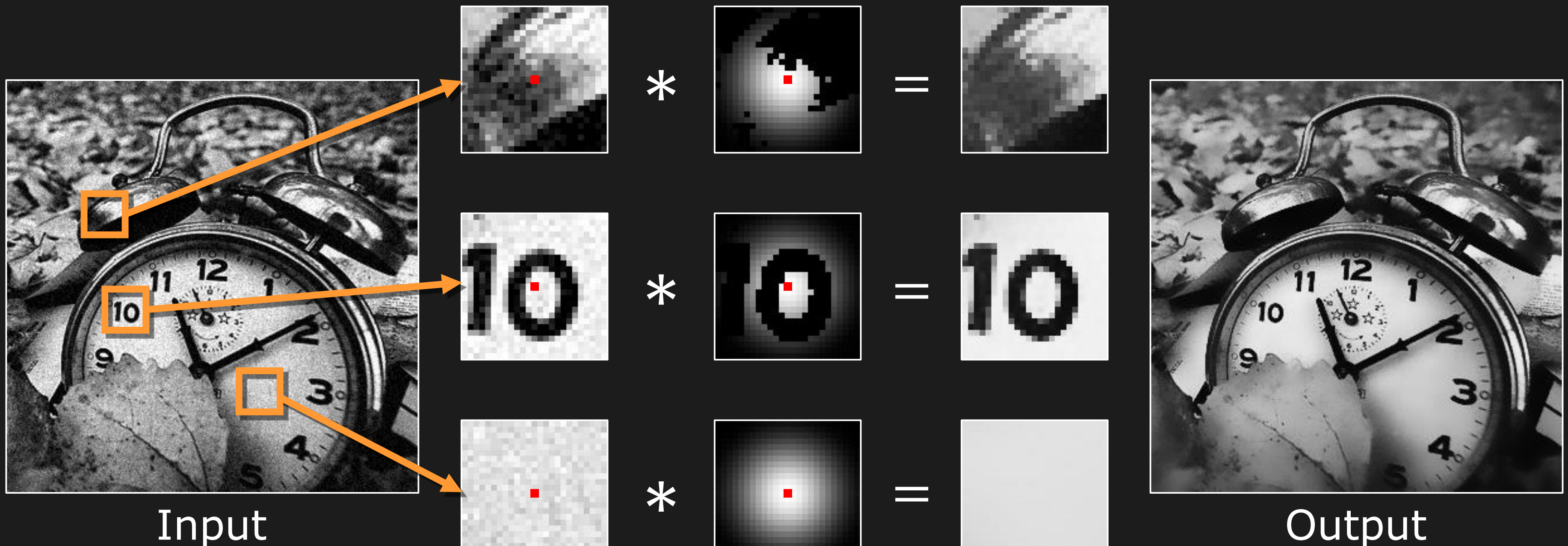
Revisiting Gaussian Smoothing



Same Gaussian Kernel is used Everywhere

Blurs Across Edges

Blur Similar Pixels Only

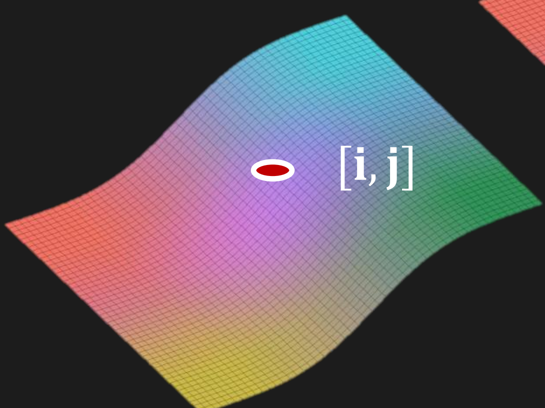


“Bias” Gaussian Kernel such that pixels not similar in intensity to the center pixel receive a lower weight.

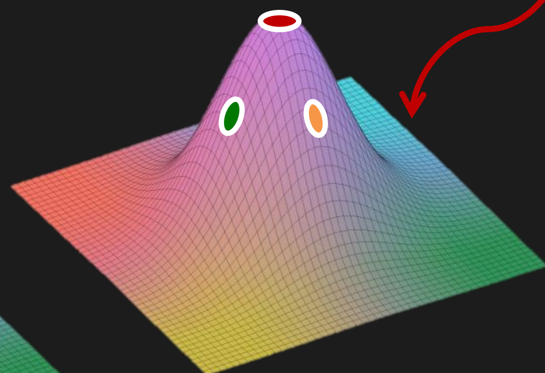
Bilateral Filter: Start With Gaussian

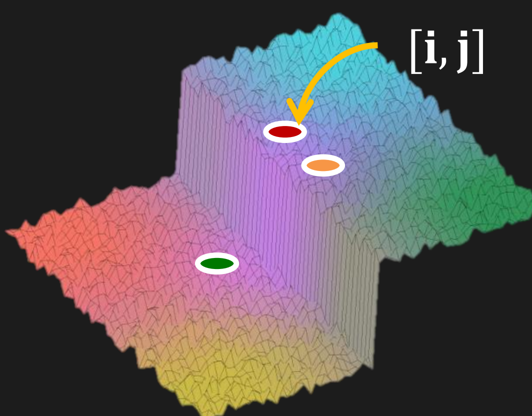
Spatial Gaussian

$$g[i,j] = \frac{1}{W_s} \sum_m \sum_n f[m,n] \, n_{\sigma_s}[i-m, j-n]$$



Gaussian Smoothed Output (g)



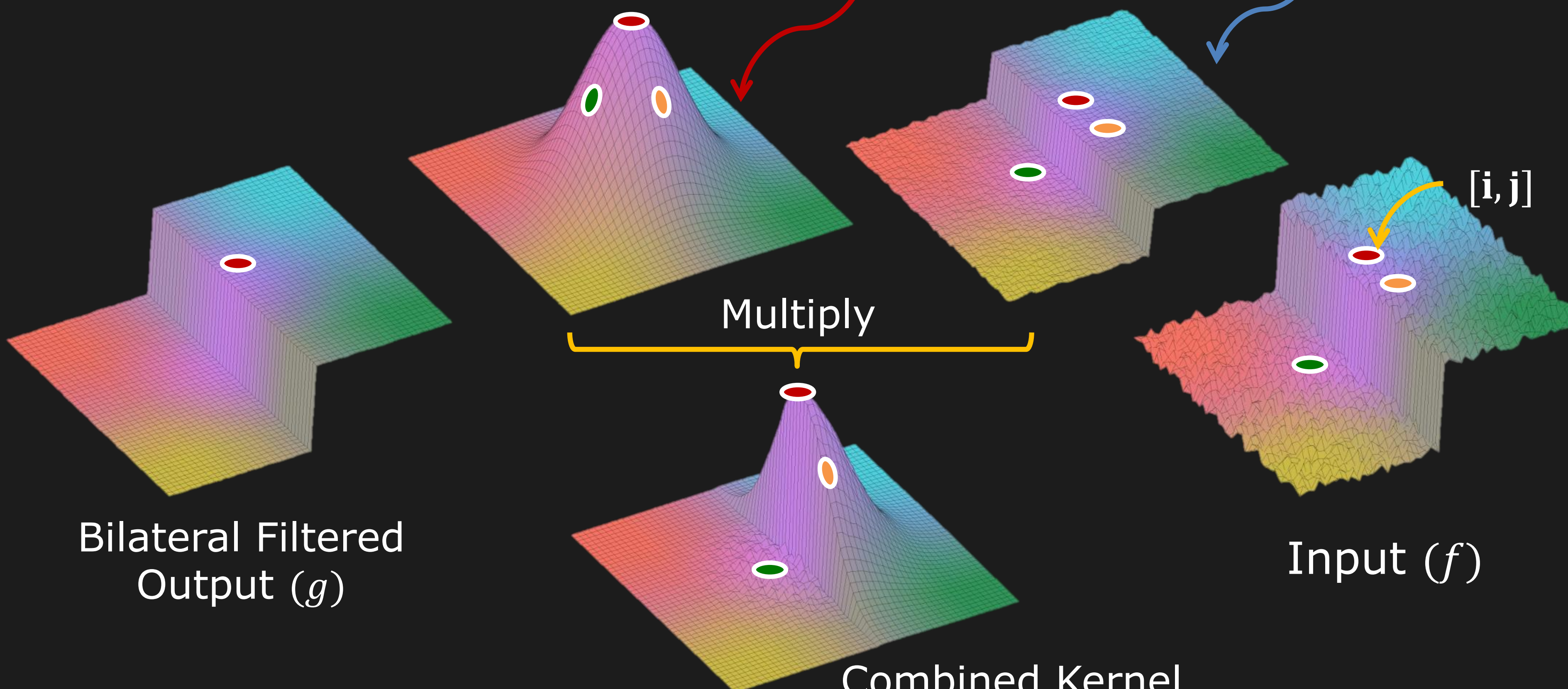


Input (f)

Gaussian Blurs Across Edges

Bilateral Filter: Add Bias to Gaussian

$$g[i,j] = \frac{1}{W_{sb}} \sum_m \sum_n f[m,n]\, \underbrace{n_{\sigma_s}[i-m, j-n]}_{\text{Spatial Gaussian}}\, \underbrace{n_{\sigma_b}(|f[m,n] - f[i,j]|)}_{\text{Brightness Gaussian}}$$



Bilateral Filter: Summary

$$g[i,j] = \frac{1}{W_{sb}} \sum_m \sum_n f[m,n]\, n_{\sigma_s}[i-m, j-n] n_{\sigma_b}(|f[m,n] - f[i,j]|)$$

Where:

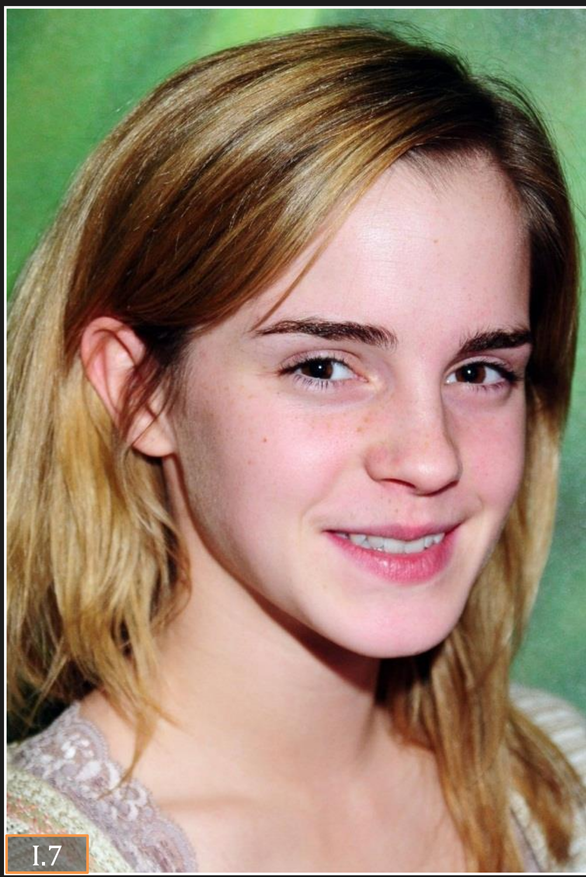
$$n_{\sigma_s}[m,n] = \frac{1}{2\pi{\sigma_s}^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma_s^2}\right)} \qquad n_{\sigma_b}[k] = \frac{1}{2\pi{\sigma_b}^2} e^{-\frac{1}{2}\left(\frac{k^2}{\sigma_b^2}\right)}$$

$$W_{sb} = \sum_m \sum_n n_{\sigma_s}[i-m, j-n] n_{\sigma_b}(|f[m,n] - f[i,j]|)$$

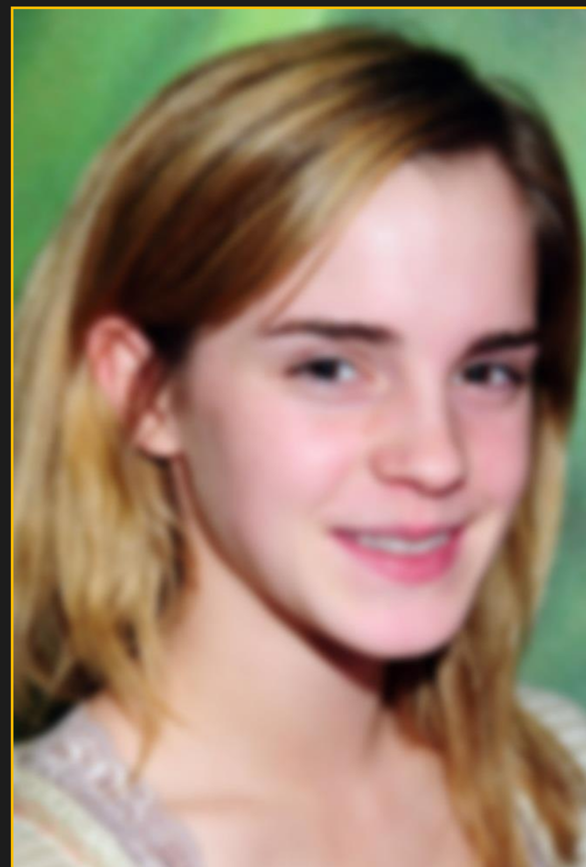
Non-linear Operation

(Cannot be implemented using Convolution)

Gaussian vs. Bilateral Filtering: Example



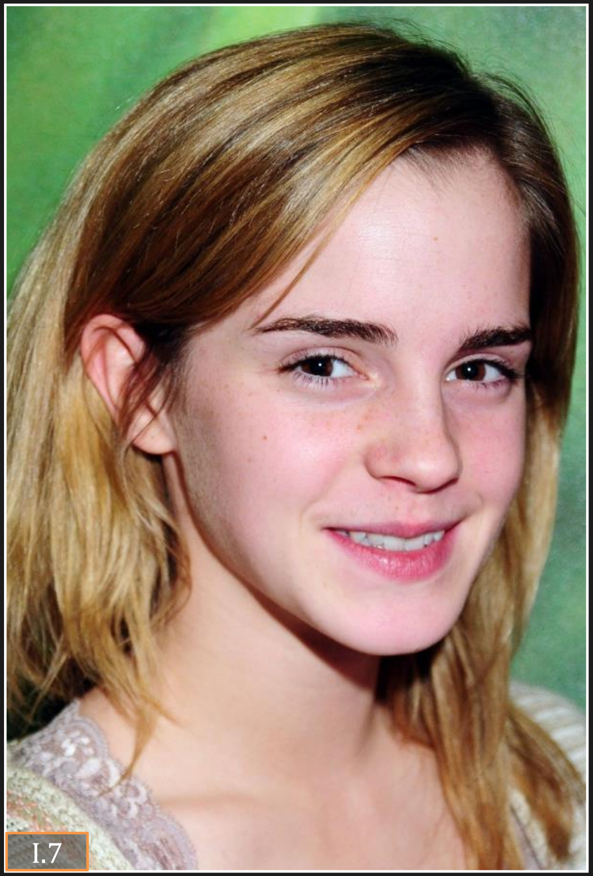
Original



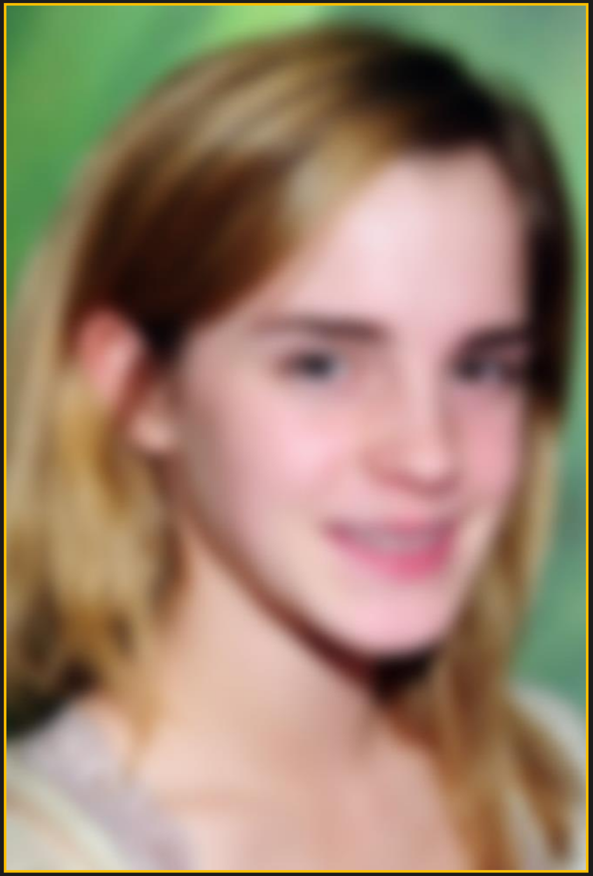
Gaussian
$\sigma_s = 2$

Bilateral
$\sigma_s = 2, \sigma_b = 10$

Gaussian vs. Bilateral Filtering: Example



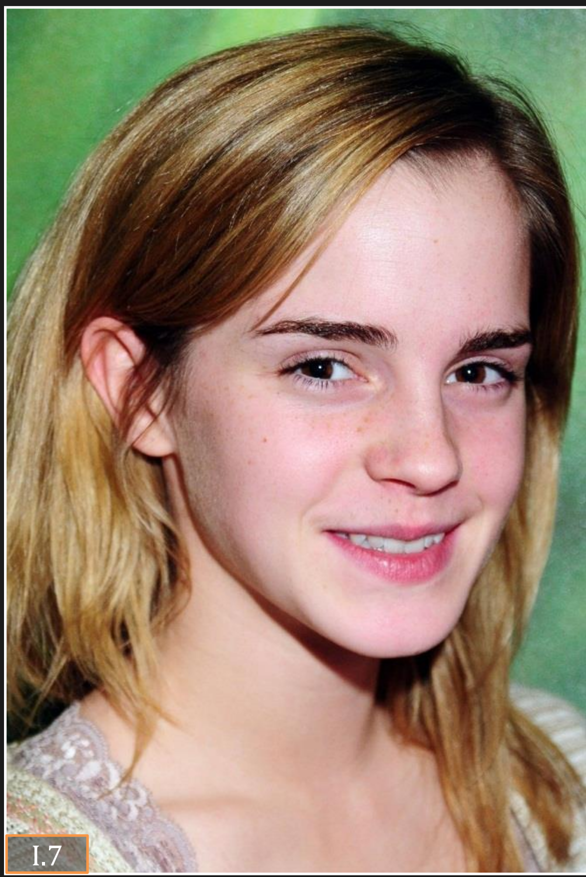
Original



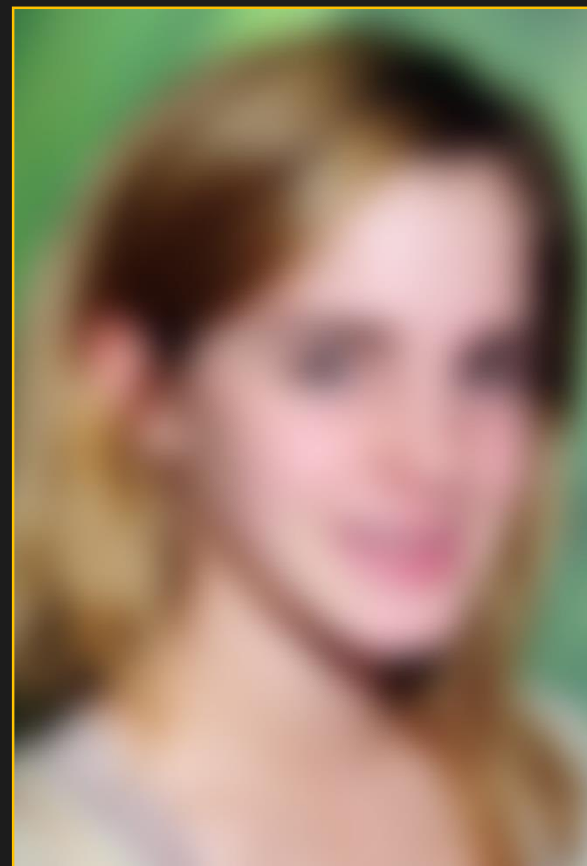
Gaussian
$\sigma_s = 4$

Bilateral
$\sigma_s = 4, \sigma_b = 10$

Gaussian vs. Bilateral Filtering: Example



Original

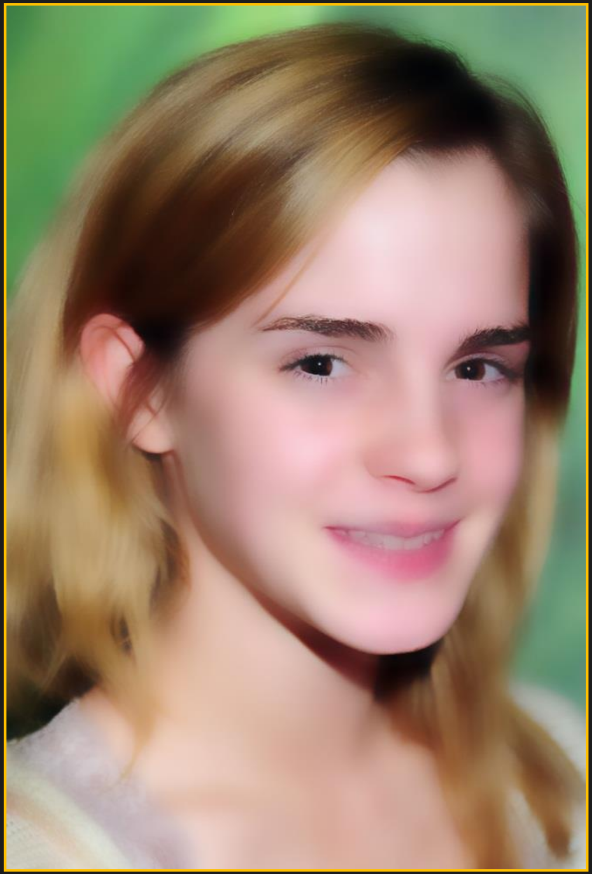


Gaussian
$\sigma_s = 8$

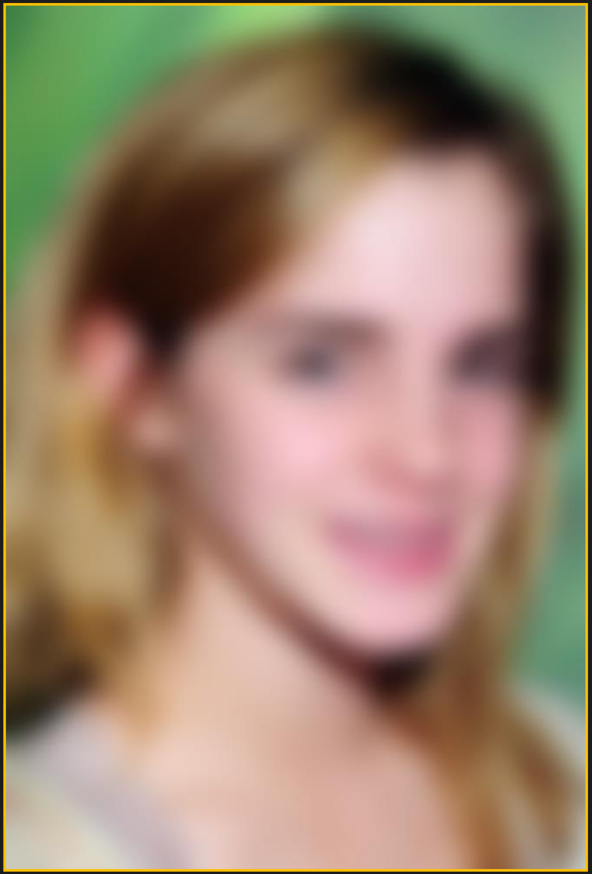
Bilateral
$\sigma_s = 8, \sigma_b = 10$

Bilateral Filtering: Changing σ_b

Bilateral
$\sigma_s = 6, \sigma_r = 10$



Bilateral
$\sigma_s = 6, \sigma_r = 20$



Bilateral
$\sigma_s = 6, \sigma_b = \infty$
(Gaussian Smoothing)

Template Matching

Template

How do we locate the template in the image?

Minimize:

$$E[i,j] = \sum_{m}\sum_{n}(f[m,n] - t[m-i,n-j])^2$$

$$E[i,j] = \sum_{m}\sum_{n}(f^2[m,n] + t^2[m-i,n-j] \underbrace{- 2f[m,n]t[m-i,n-j]}_{\text{Maximize}})$$

Template Matching

Template

How do we locate the template in the image?

Maximize:

$$R_{tf}[i,j] = \sum_{m}\sum_{n} f[m,n]\,t[m-i,n-j] = t \otimes f$$

(Cross-Correlation)

Convolution vs. Correlation

Convolution:

$$g[i,j] = \sum_{m}\sum_{n} f[m,n]\underline{t[i-m,j-n]} = t * f$$

Correlation:

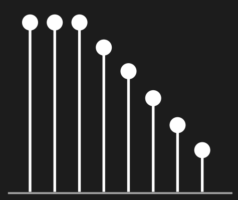
$$R_{tf}[i,j] = \sum_{m}\sum_{n} f[m,n]\underline{t[m-i,n-j]} = t \otimes f$$

No Flipping in Correlation

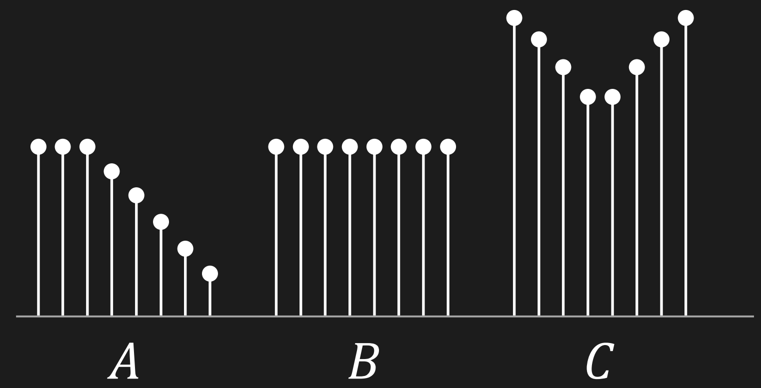
Problem with Cross-Correlation

$$R_{tf}[i,j] = \sum_m \sum_n f[m,n]\,t[m-i,n-j] = t \otimes f$$

t:



f:



$$R_{tf}(C) > R_{tf}(B) > R_{tf}(A)$$

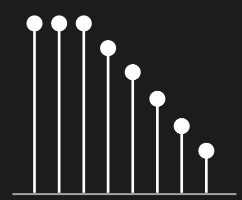
We need $R_{tf}(A)$ to be the maximum!

Normalized Cross-Correlation

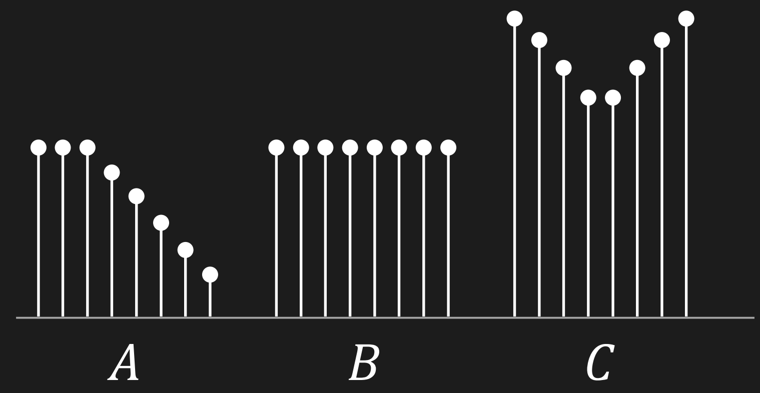
Account for energy differences

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]\,t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]}\,\sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$

t:



f:

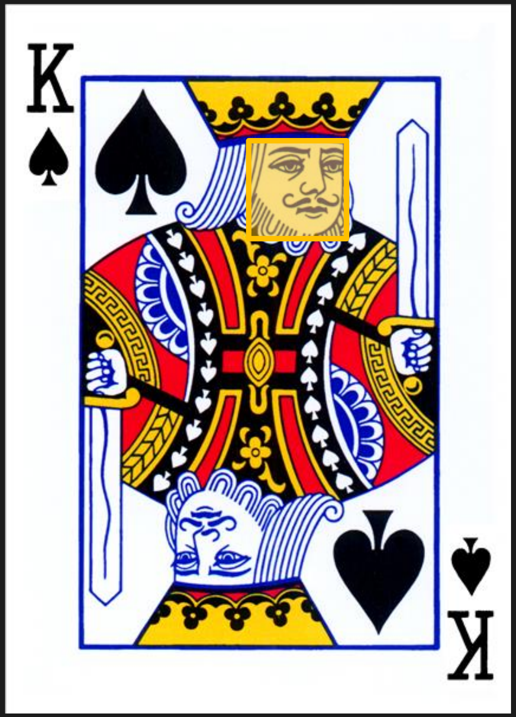


$$R_{tf}(A) > R_{tf}(B) > R_{tf}(C)$$

Normalized Cross-Correlation

Account for energy differences

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]\,t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]}\;\sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$



$\otimes$

$=$

Correlation: Issues

- Problem at borders
- Sensitive to object pose, scale and rotation
- Not good for general object recognition
- Good for feature detection
- Can be computationally expensive

Convolution is LSIS: Linearity

Let: $g_1(x) = \int_{-\infty}^{\infty} f_1(\tau)h(x-\tau)\,d\tau$ and $g_2(x) = \int_{-\infty}^{\infty} f_2(\tau)h(x-\tau)\,d\tau$

Then:

$$\int_{-\infty}^{\infty} (\alpha f_1(\tau) + \beta f_2(\tau))h(x-\tau)\,d\tau$$

$$= \alpha \int_{-\infty}^{\infty} f_1(\tau)h(x-\tau)\,d\tau + \beta \int_{-\infty}^{\infty} f_2(\tau)h(x-\tau)\,d\tau$$

$$= \alpha g_1(x) + \beta g_2(x)$$

Convolution is LSIS: Shift Invariance

Let: $g(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$

Then:

$$\int_{-\infty}^{\infty} f(\tau - a)h(x-\tau)\,d\tau$$

$$= \int_{-\infty}^{\infty} f(\mu)h(x-a-\mu)\,d\mu$$

(Substituting $\mu = \tau - a$)

$$= g(x-a)$$

References

Textbooks:

Digital Image Processing (Chapter 3)
González, R and Woods, R., Prentice Hall

Robot Vision (Chapter 6 and 7)
Horn, B. K. P., MIT Press

Computer Vision: A Modern Approach (Chapter 7)
Forsyth, D and Ponce, J., Prentice Hall

Papers:

[Tomasi 1998] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in Proceedings of the IEEE International Conference on Computer Vision, 1998.

Image Credits

I.1 Sasse, Arthur. Einstein © Ullstein Bilderdienst, Berlin.

I.2 http://www.flickr.com/photos/byspice/4577634277

I.3 http://www.toptenz.net/wp-content/uploads/2008/06/kristin-kreuk.jpg

I.4 http://images.nationalgeographic.com/wpf/media-live/photos/000/093/cache/giant-lobelia-plant_9372_600x450.jpg

I.5 Matlab Demo Image

I.6 Adapted from http://1x.com/photo/24072

I.7 http://cdn04.cdn.socialitelife.com/wp-content/uploads/2009/11/stars-without-makeup-photos-11062009-07.jpg