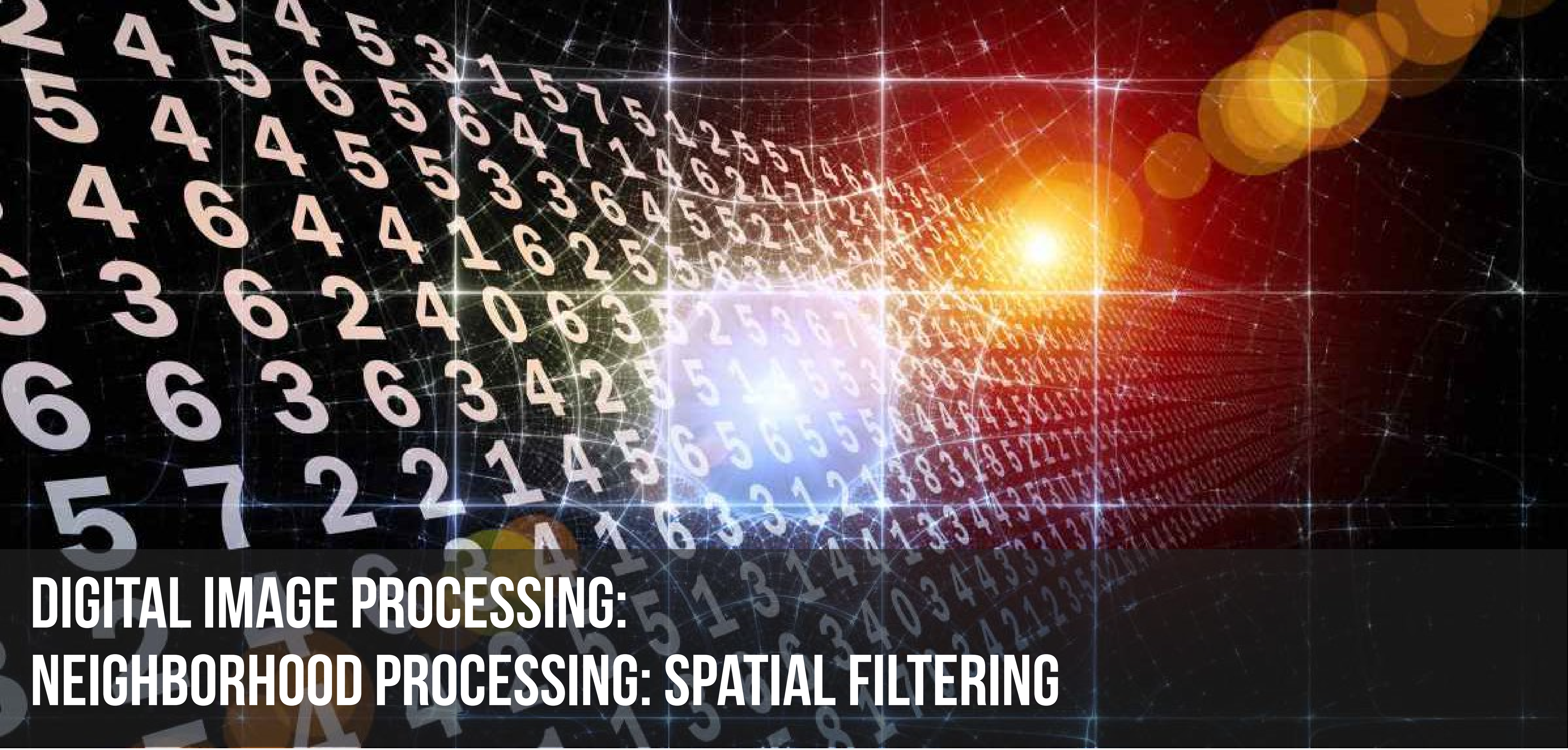
PENGOLAHAN SINYAL DIGITAL

Adhi Harmoko Saputro



DIGITAL IMAGE PROCESSING:
NEIGHBORHOOD PROCESSING: SPATIAL FILTERING

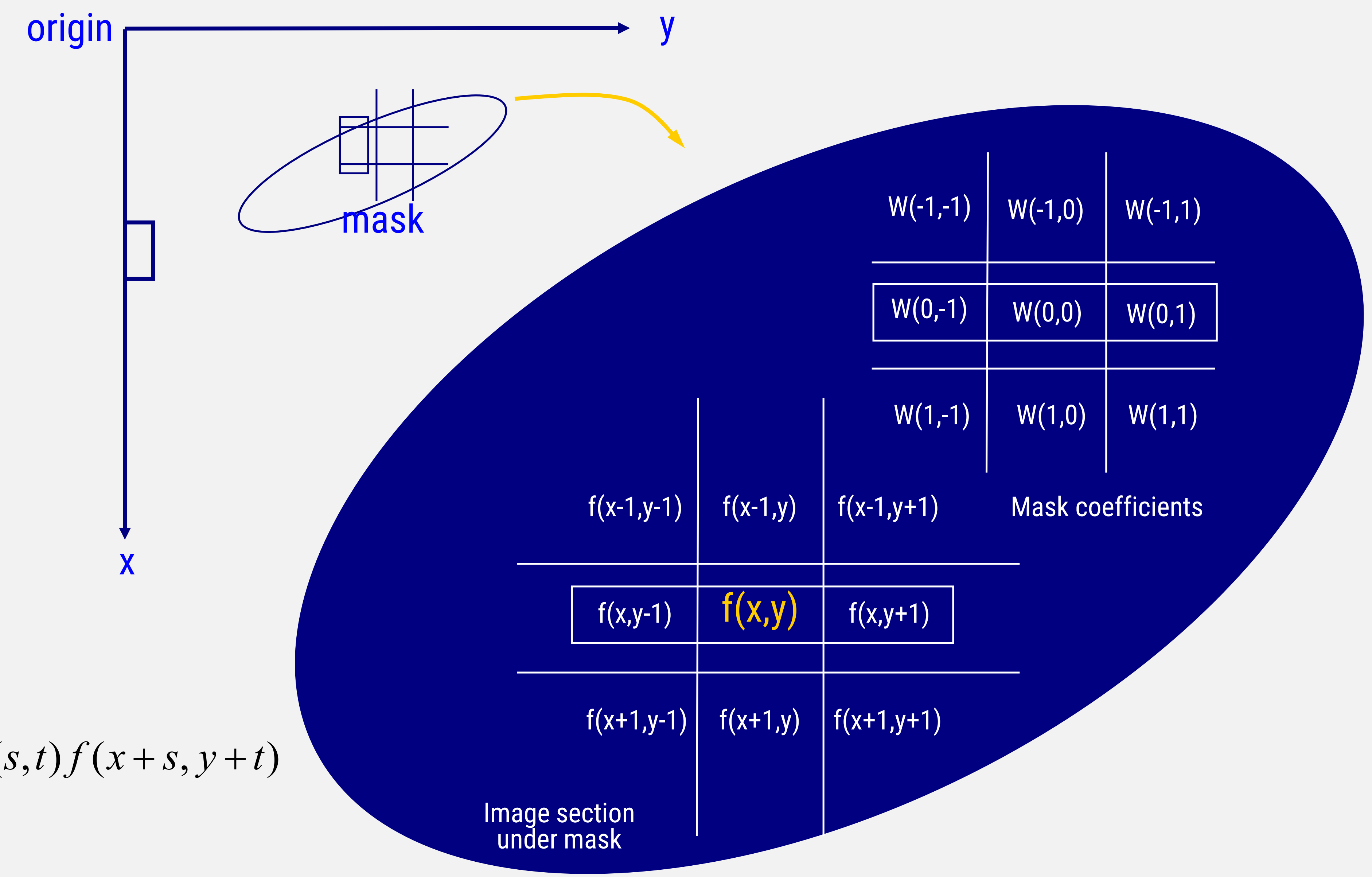
Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

SPATIAL FILTERING

- Definition: a process that moves a subimage from point to point in an image, with the response at each image point predefined.
- The subimage: filter, mask, kernel, template or window.
- The values in a filter subimage are called coefficients, not pixels.
- The process is also named convolution.
- Convolution g of 2D function f and w is denoted $f*w$ or $f \otimes w$

ILLUSTRATION



$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t)$$

EXAMPLE 1: AVERAGING

- One simple example is smoothing using a 3x3 mask.

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y+1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

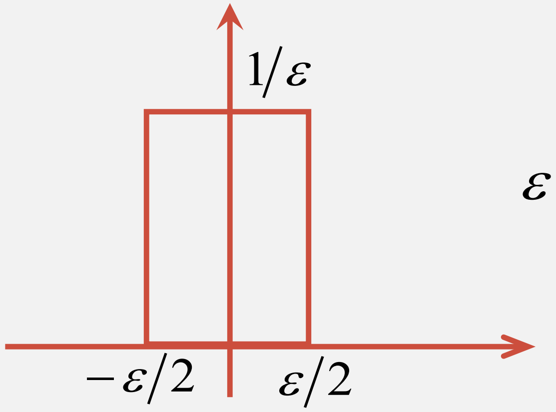
$$g(x,y)=\frac{1}{9}\left[f(x-1,y-1)+f(x-1,y)+f(x-1,y+1)+\ldots+f(x+1,y+1)\right]$$

EXAMPLE2: DELTA/IMPULSE FUNCTION

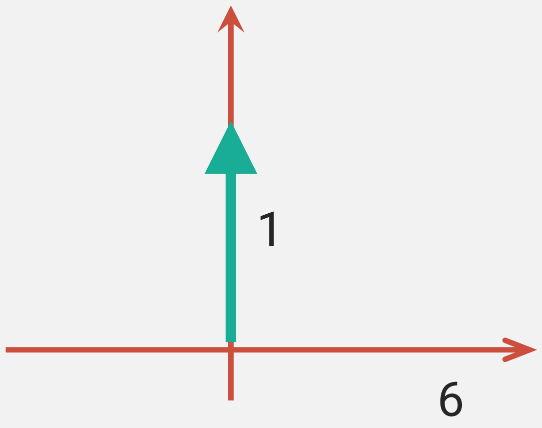
- An ideal impulse is defined using the Dirac distribution $\delta(x, y)$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x, y)dxdy = 1, \text{and } \delta(x, y) = 0 \forall x, y \neq 0$$

- To visualize in 1D, picture a rectangular pulse from $x - \varepsilon/2$ to $x + \varepsilon/2$ with a height of $1/\varepsilon$. As $\varepsilon \rightarrow 0$, the width tends to 0 and the height tends to infinity as the total area remains constant at 1.



$\varepsilon \rightarrow 0$



EXAMPLE2: DELTA/IMPULSE FUNCTION

- Sifting property: It provides the value of $f(x,y)$ at the point (a,b)

- Delta function as the mask.
  - $$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x-a,y-b)\,dx\,dy = f(a,b)$$
  - When moving across an image, it "copies" the value of image intensity.

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-s,y-t)\delta(s,t)\,ds\,dt = f\left(x,y\right)$$

GENERAL FORMULATION

- 2D Continuous space:

$$g(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-s,y-t)w(s,t)dsdt$$

- 2D discrete space:

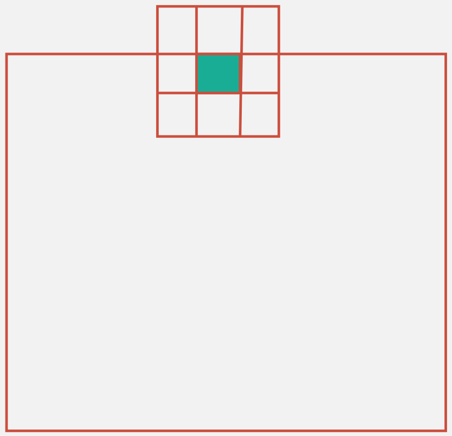
$$g[x,y]=\sum_{s=-\infty}^{\infty}\sum_{t=-\infty}^{\infty} f[x-s,y-t]w[s,t]$$

Convolution kernel/mask

- In digital image processing, we only deal with discrete space with a mask effective locally.
- Convolution is commutative, associative and distributive.

WHAT HAPPENS AT THE BORDERS?

- The mask falls outside the edge.
- Solutions?
  - Ignore the edges
    - The resultant image is smaller than the original
  - Pad with zeros
    - Introducing unwanted artifacts



VALUES OUTSIDE THE RANGE

- Linear filtering might bring the intensity outside the display range.
- Solutions?
    - Clip values
$$y = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \le x \le 255 \\ 255 & \text{if } x > 255 \end{cases}$$
    - Scaling transformation
$$y = 255 \frac{x - g_L}{g_H - g_L}$$
New min

New max

        - Transform values in [gL,gH] to [0,255]

SEPARABLE FILTERS

- Some filters can be implemented by successive application of two simpler filters.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

- Separability results in great time saving.
- By how much?
  - For a mask of size n $\times$ n, for each image pixel
    - Originally, n^2 multiplications and n^2-1 additions
    - After separation, 2n multiplications and 2n-2 additions

EXAMPLE 1: AVERAGING

```
im = imread('onion.png');
h = [1 1 1; 1 1 1; 1 1 1]./9;
imFil = imfilter(im,h);
figure;
subplot(121); imshow(im); title('Original Image');
subplot(122); imshow(imFil); title('Averaging Image');
```

EXAMPLE 1: AVERAGING

Original Image

Averaging Image

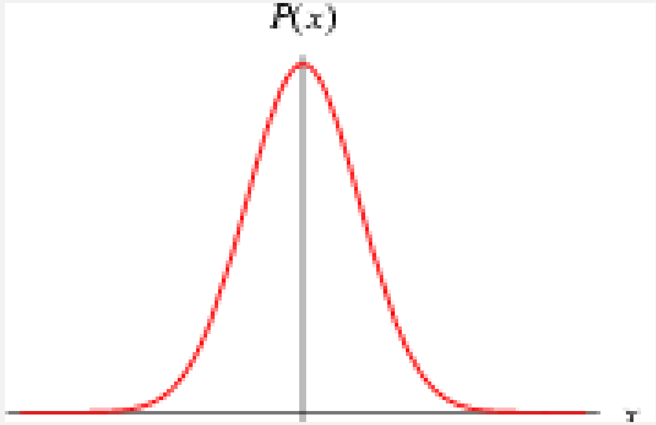
SOME TERMINOLOGY ON FREQUENCY…

- Frequencies: a measure of the amount by which gray values change with distance.
- High/low-frequency components: large/small changes in gray values over small distances.
- High/low-pass filter: passing high/low components, and reducing or eliminating low/high-frequency filter.

LOW-PASS FILTER

- Mostly for noise reduction/removal and smoothing
  - 3x3 averaging filter to blur edges
  - Gaussian filter,
    - based on Gaussian probability distribution function
    - a popular filter for smoothing
    - more later when we discuss image restoration

In 1D: $P(x) = e^{-x^2/2\sigma^2}$



HIGH-PASS FILTER

- The Laplacian, $\nabla^2 f(x, y)$, is a high-pass filter,

$$h_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{or} \quad h_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

- Sum of coefficients is zero
- Insight: in areas where the gray values are similar (low-frequency), application of the filter makes the gray values close to zero.

LAPLACIAN OPERATOR

- Laplacian is a derivative operator – 2nd order derivative

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\Rightarrow \quad h_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Highlighting gray level discontinuities.
- An edge detector
- isotropic in x- and y-direction.

Laplacian Operator (con'd)

- We may add the diagonal terms too => isotropic results for increments of 45°

$$h_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{or} \quad h_4 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- There are other variations/approximations:

$$h_1 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Separable filter

LAPLACIAN FILTER: DEMO

```
im = imread('moon.tif');
h = [1 1 1; 1 -8 1; 1 1 1];
imFil = imfilter(im,h);
figure;
subplot(121); imshow(im); title('Original Image');
subplot(122); imshow(imFil); title('Laplacian Image');
```

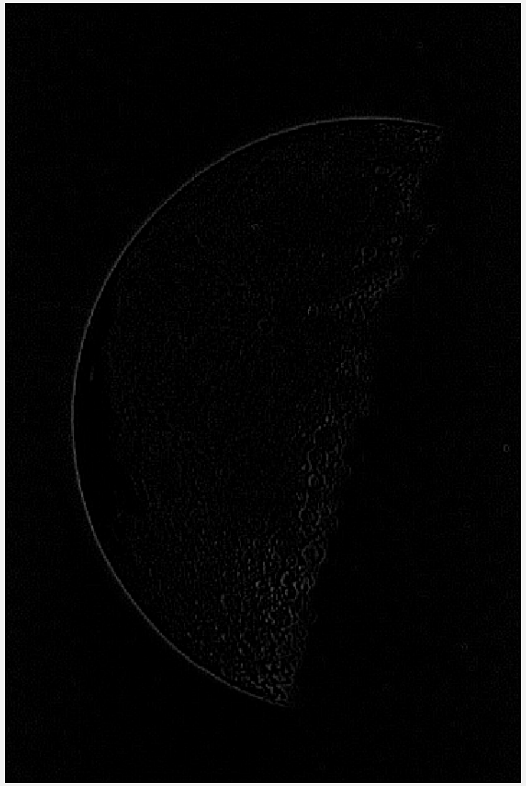
LAPLACIAN FILTER: DEMO

```
im = imread('moon.tif');
h = fspecial('laplacian',0.5);
imFil = imfilter(im,h);
figure;
subplot(121); imshow(im); title('Original Image');
subplot(122); imshow(imFil); title('Laplacian Image');
```

LAPLACIAN FILTER: DEMO

Original Image

Laplacian Image



LAPLACIAN: APPLICATION

- Laplacian highlights discontinuities, and turn featureless regions into dark background.
- How can we make good use of the property?
  - One example is edge enhancement

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y), \text{ using } h_1 \\ f(x,y)+\nabla^2 f(x,y), \text{ using } h_2 \end{cases}$$

original

Laplacian enhanced

MORE FILTERS ...

- What feature does each mask highlight?

$$h_1 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad h_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

EDGE SHARPING

- To make edges slightly sharper and crisper.
- This operation is referred to as edge enhancement, edge crispening or unsharp masking.
- A very popular practice in industry.
- Subtracting a blurred version of an image from the image itself.

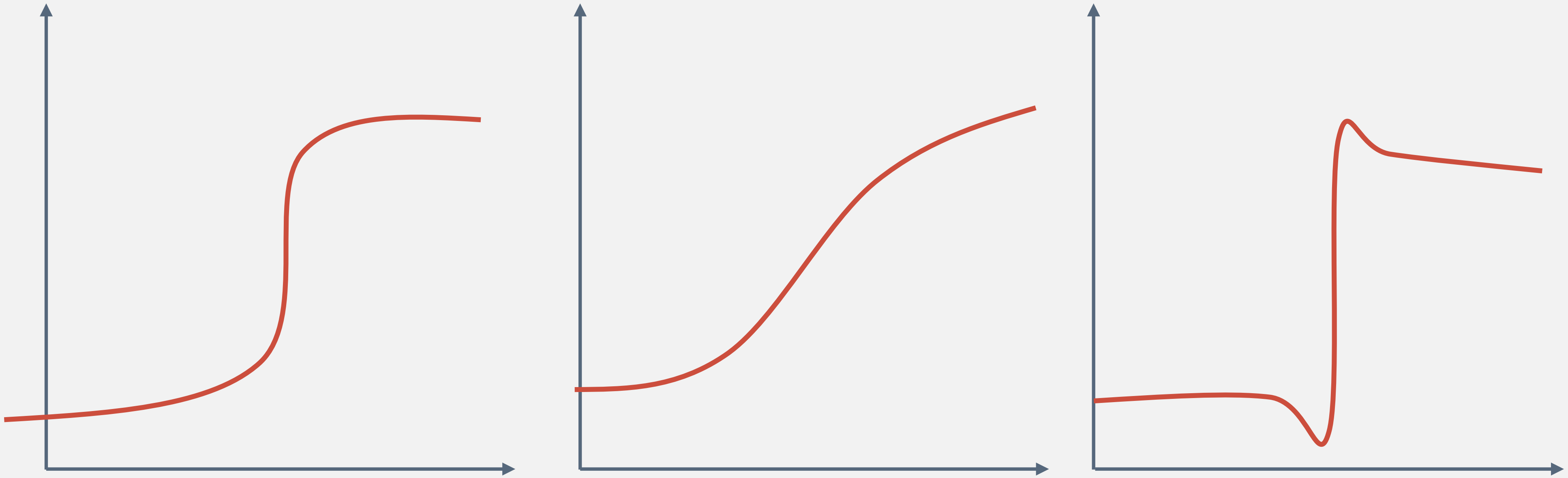
$$f_s(x,y) = f(x,y) - \overline{f}(x,y)$$

Sharpened image

Original image

Blurred image

UNSHARP MASKING: WHY IT WORKS



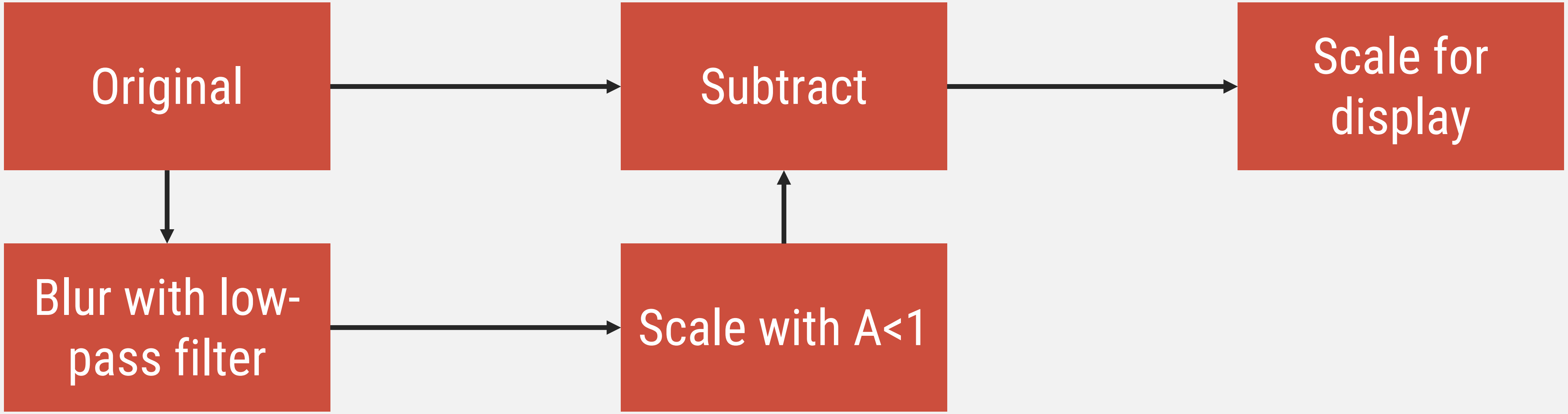
f(x,y) blurred image Sharpened

FILTER FOR UNSHARP MASKING

- Combining filtering and subtracting in one filter.

$$h = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{A} \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \approx A \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

- Schematically



HIGH-BOOST FILTERING

- Generalization of unsharp masking $f_{hb}(x,y) = Af(x,y) - \overline{f}(x,y)$
- Here A is called boost coefficient, and $A \geq 1$
- We rewrite the equation as

$$
\begin{aligned}
f_{hb}(x,y) &= (A-1)f(x,y) + f(x,y) - \overline{f}(x,y) \\
&= (A-1)f(x,y) + f_s(x,y)
\end{aligned}
$$

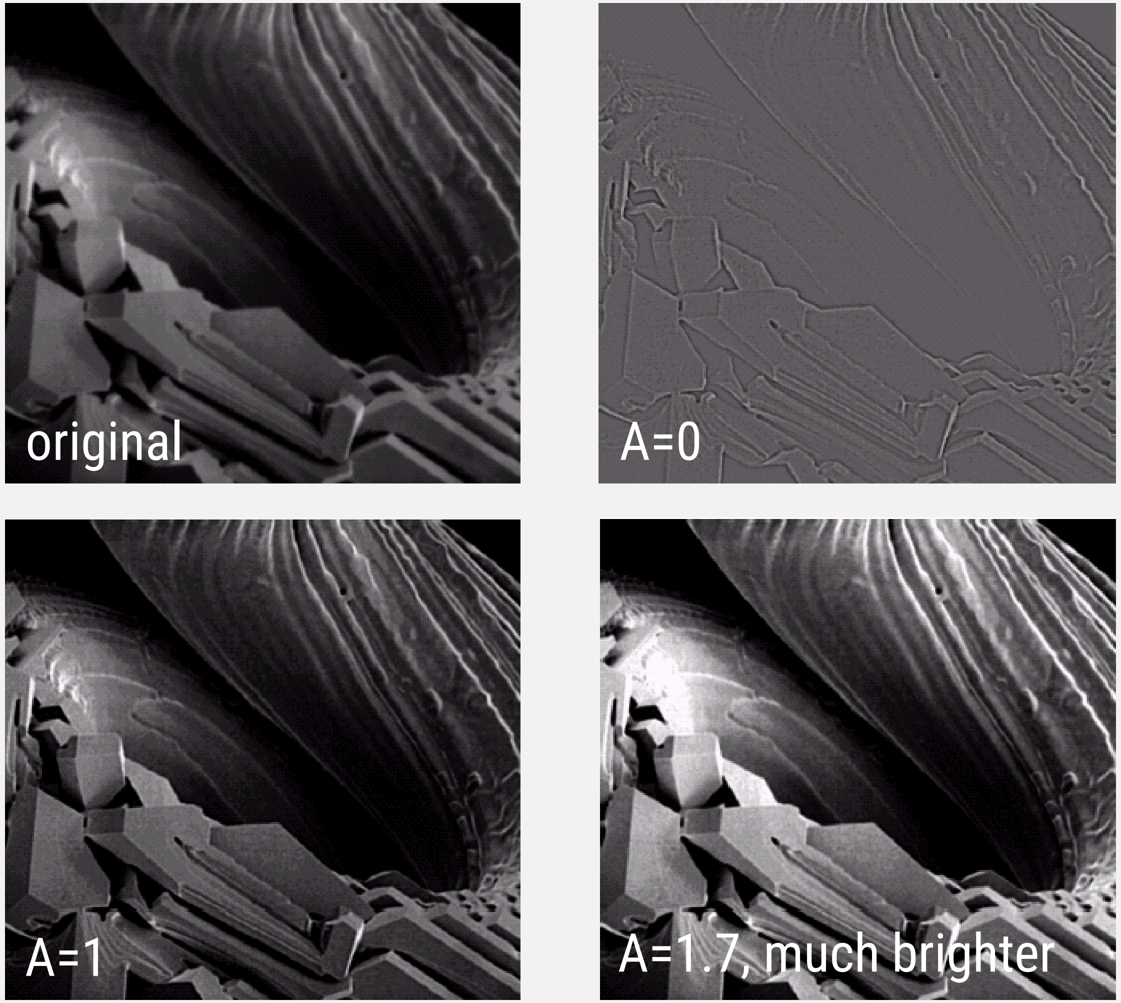
- Applicable to any sharpening operation
  - f_s can be $f \pm \nabla^2 f$
  - The filter for f_{hb} becomes

$$
h = \begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix}
$$

HIGH-BOOST FILTERING: DEMO

$$h = \begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

- By varying A, better overall brightness can be improved.



NONLINEAR FILTERS

- Will discuss some of them in more detail later for the purpose of image restoration.
- Maximum filter: the output the maximum value under the mask
- Minimum filter: the output the minimum value under the mask
- Rank-order filter:
  - Elements under the mask are ordered, and a particular value is returned.
  - Both maximum and minimum filter are instances of rank-order
  - Another popular instance is median filter

MORE NONLINEAR FILTERS

- Alpha-trimmed mean filter:
  - Order the values under the mask
  - Trim off elements at either end of the ordered list
  - Take the mean of the remainder
  - E.g. assuming a 3x3 mask and the ordered list

$$x_1 \le x_2 \le x_3 \le \ldots \le x_9$$

  - Trimming of two elements at either end, the result of the filter is

$$\left(x_3 + x_4 + x_5 + x_6 + x_7\right)/5$$

SUMMARY

- We introduced the concept of convolution
- We briefly discussed spatial filters of
  - Low-pass filter for smoothing
  - High-pass filter for edge sharpening
  - Nonlinear
- We'll come back to most of the filters under the appropriate topics.

TERIMA KASIH

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia