



10th Astronomical Data Analysis Summer School (ADA X)

Fundamental of Image Processing

Effrosyni Doutsis, Ph.D.

Post-doctoral Researcher
Signal Processing Lab at FORTH-ICS

Who am I?

- ✓ **Post-doctoral Researcher**

- ✓ Signal Processing Lab (SPL), Institute of Computer Science (ICS),
Foundation for Research and Technology – Hellas (FORTH), Greece (since 2018).

- ✓ **Ph.D. in Computer Science**

I3S lab, CNRS, Université Côte d'Azur, 4G-TECHNOLOGY, France (2017).

- ✓ **MSc in Computational Biology and Biomedicine**

Université de Nice-Sophia Antipolis, France (2013).

- ✓ **BSc in Computer Science and Biomedical Informatics**

University of Thessaly, Greece (2012).

Teaching Activities

- ✓ **Visiting Professor**, Hellenic Air Force Academy, Greece (since 2022).
- ✓ **Visiting Professor**, University of West Attica (UNIWA), Greece
Department of Informatics and Computer Science, (2022-23).
- ✓ **Visiting Professor**, Technical University of Crete (TUC),
School of Electrical and Computer Engineering (2020-21, 2021-22).
- ✓ **Teaching assistant**, Polytech of Nice, France (2014 - 2016).
Master II in Signal for Health, Telecommunications, Image and Multimedia.
- ✓ **Teaching Assistant**, Université Côte d'Azur, France.
Master I Biomedical Engineering, (2014 - 2016).
- ✓ **Teaching Assistant**, Institut Universitaire de Technologie (IUT), France (2014-2016).

Professional Pathway

- ✓ **10.2021 - now :** **Principal Investigator / BrainSIM Project Signal Processing Lab (SPL),** Institute of Computer Science (ICS), Foundation for Research and Technology – Hellas (FORTH), Greece.
- ✓ **04.2021 - 04.2022 :** **Principal Investigator / BRIEFING Project MediaCoding Team** I3S laboratory, CNRS, Université Côte d'Azur, France .
- ✓ **06.2018 - 04.2021 :** **Associate Researcher / ARCHERS and KRIPIS II Signal Processing Lab / Inst. of Computer Science, Prof. Panagiotis Tsakalides,** Foundation for Research and Technology – Hellas, Greece.
- ✓ **03.2012 - 03.2017 :** **Research Assistant MediaCoding Team, Dr. Marc Antonini, Prof. Lionel Fillatre,** I3S laboratory, CNRS, Université Côte d'Azur, France.
- ✓ **07.2012 - 09.2012 :** **Research Assistant Software & Knowledge Engineering Lab. Dr. Karampiperis,** National Center for Scientific Research Demokritos, Greece
- ✓ **09.2011 - 06.2012 :** **Research Assistant / Dr. Euripides Markou,** Department of Computer Science & Biomedical Informatics, University of Central Greece (now University of Thessaly), Greece

My Research Interests...

- ✓ **Neuro-inspired image processing**

Goal: Utilize neuroscience models that approximate the structure and function of the visual system in order to design and implement efficient algorithms according to the visual perception.

- ✓ **Machine Learning-based image compression.**

Goal: Use Machine Learning algorithms to identify the Regions of Interests (RoI) within the visual scene to drive the bit-allocation with respect to the visual scene content.

- ✓ **Computational Biology**

Goal: Use Machine Learning algorithms to learn structural characteristics of biological molecules (e.g. proteins) and predict their 3D shape.

Course Outline

- ✓ Digital Image Fundamentals
- ✓ Image Acquisition
- ✓ Image Transforms
- ✓ Convolution Theorem
- ✓ Morphological Operations
- ✓ Multiresolution Analysis
- ✓ Image Coding

Image Perception

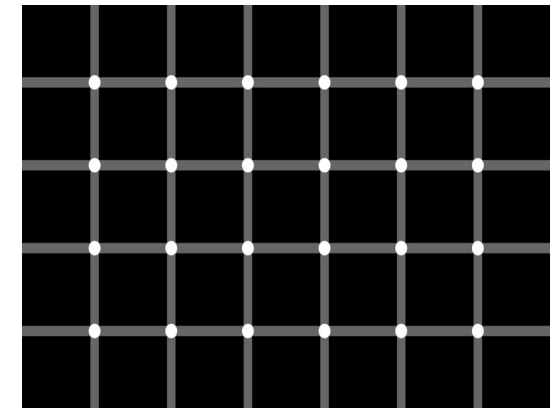
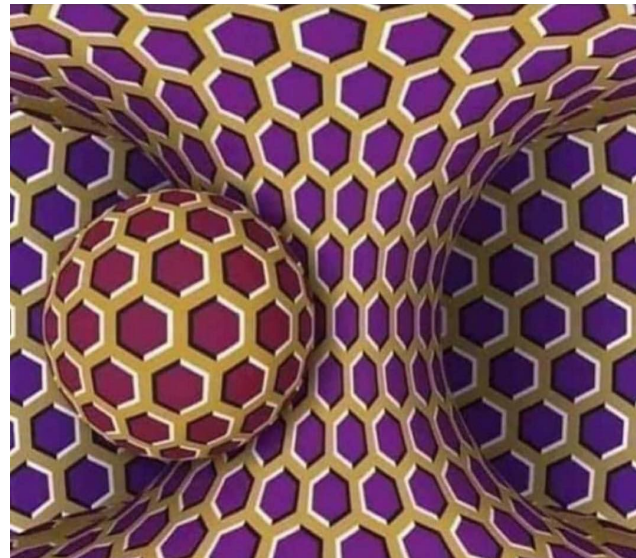
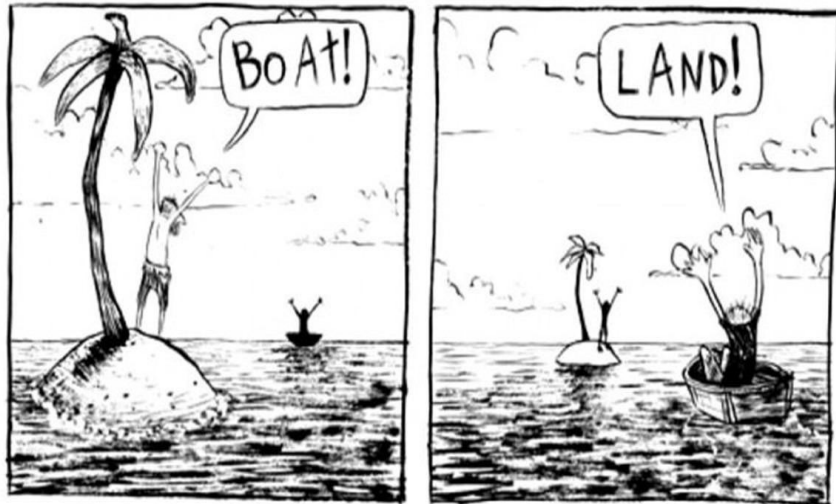
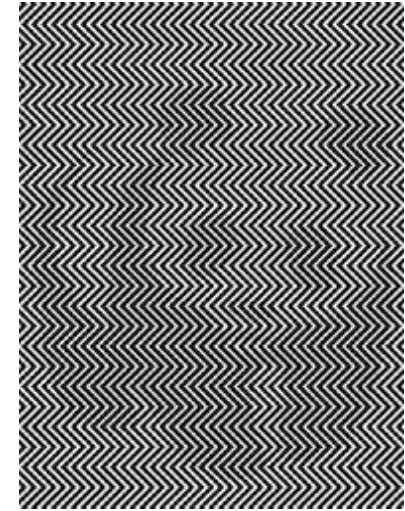
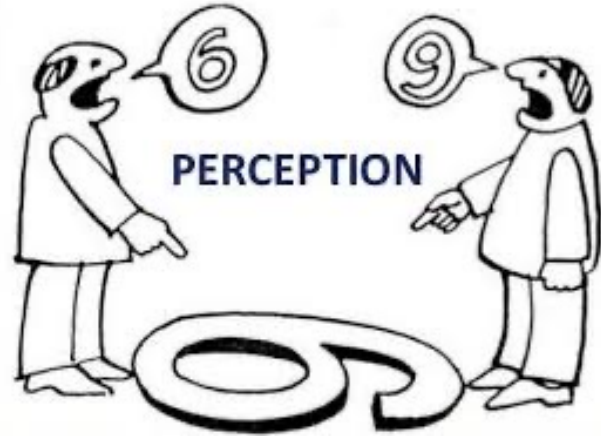


Image Transform

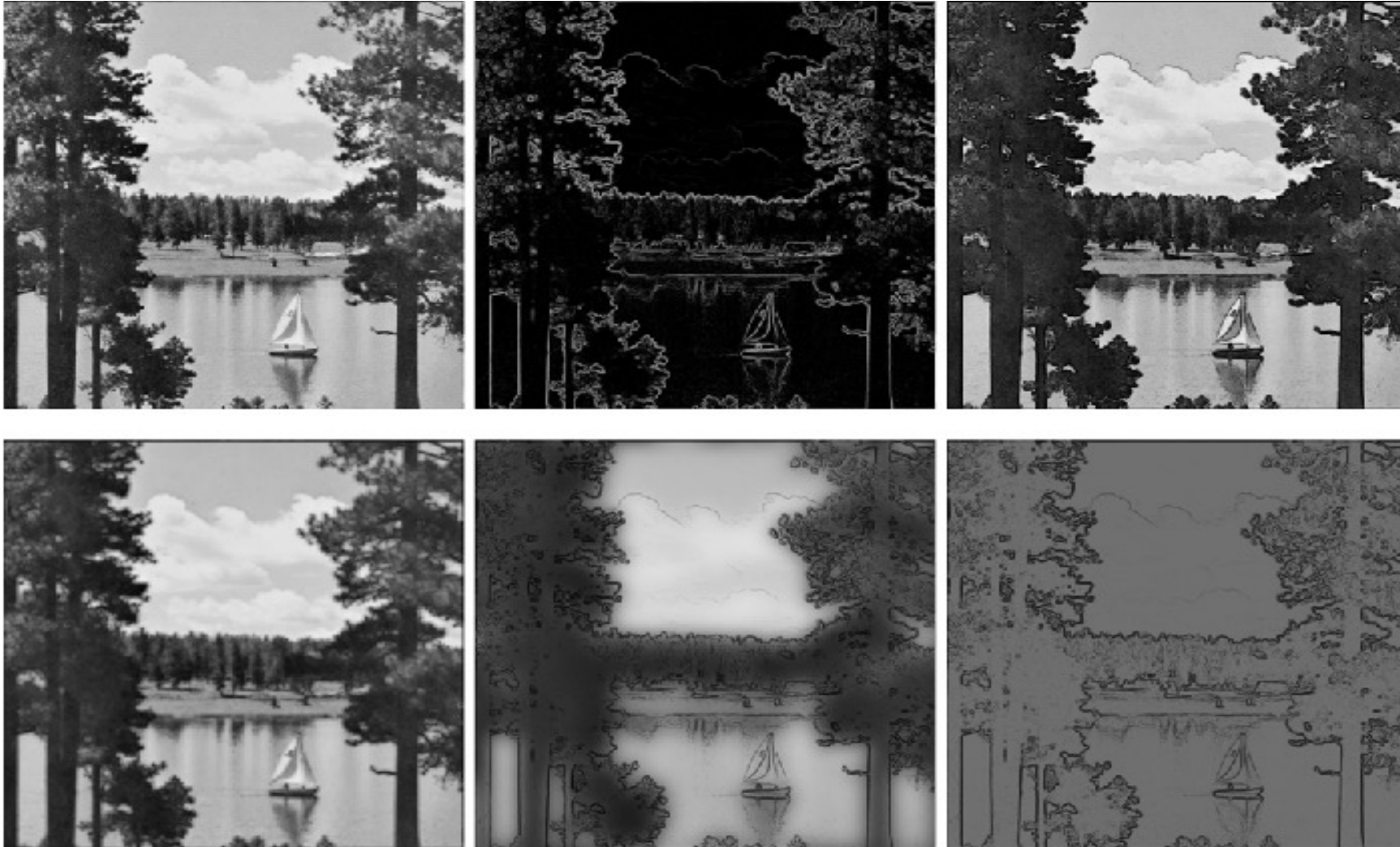


Image Enhancement

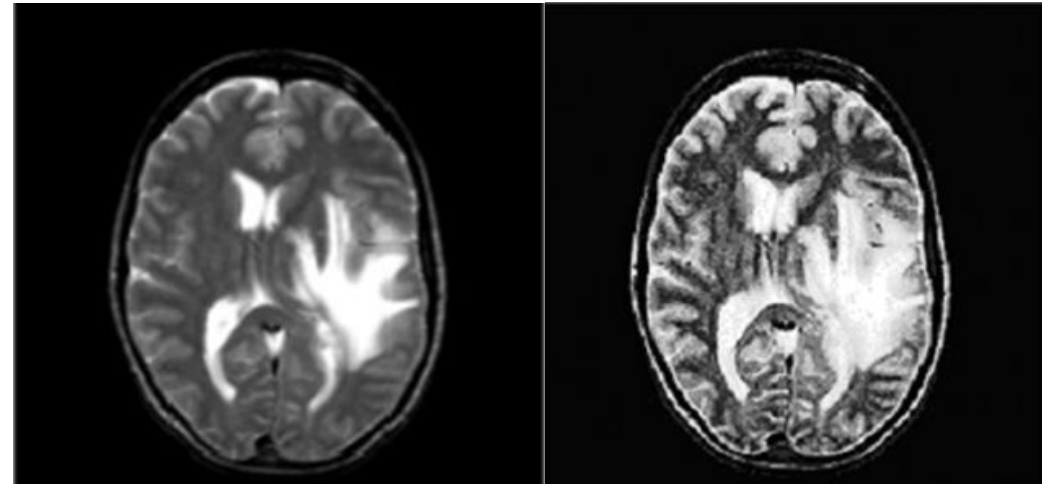


Image Restoration



Image Segmentation

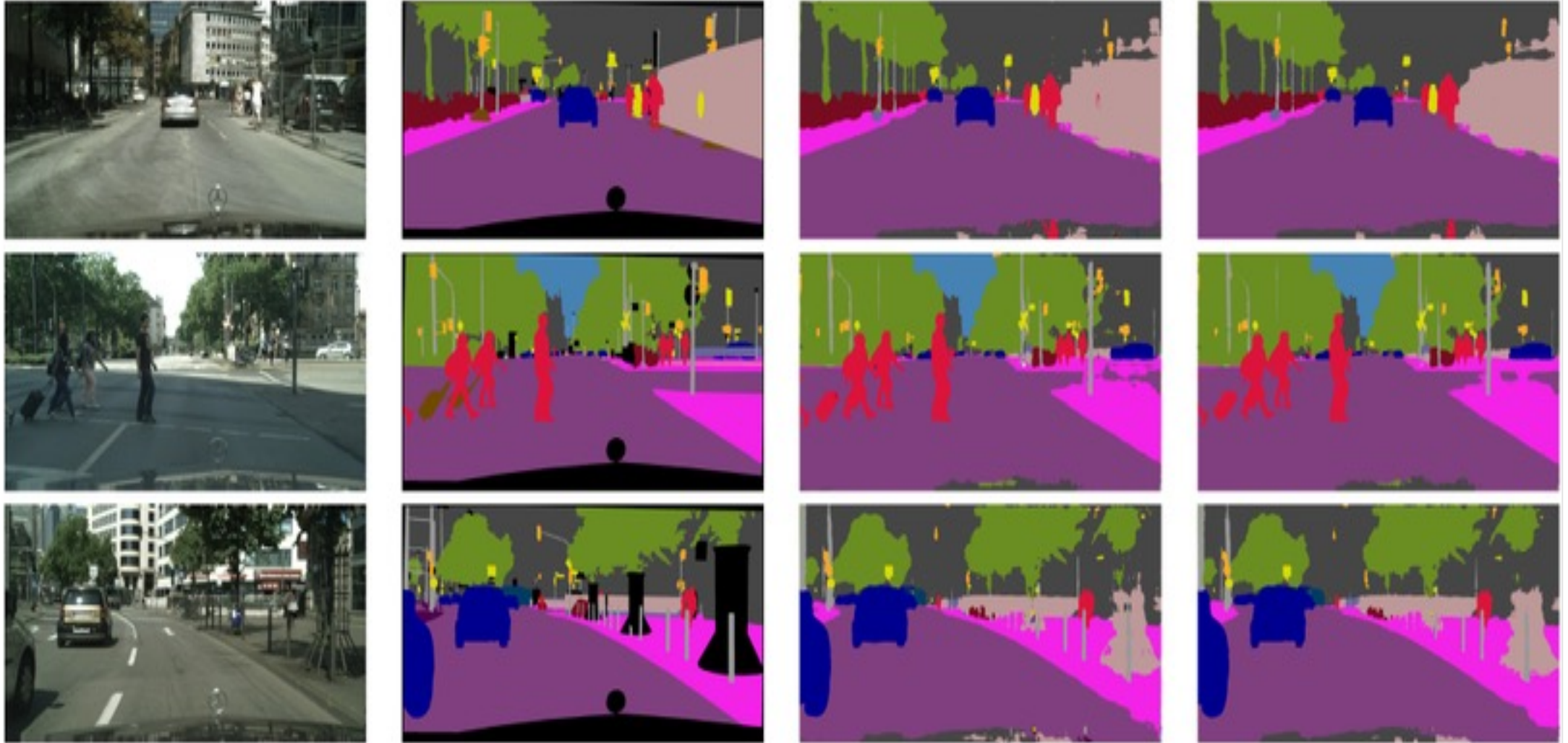


Image Description

Classification



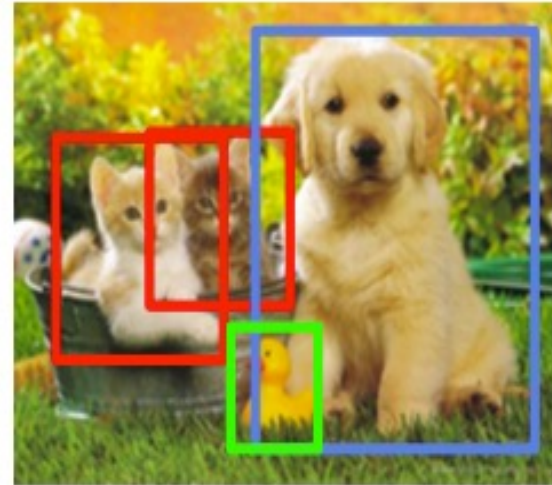
CAT

**Classification
+ Localization**



CAT

Object Detection



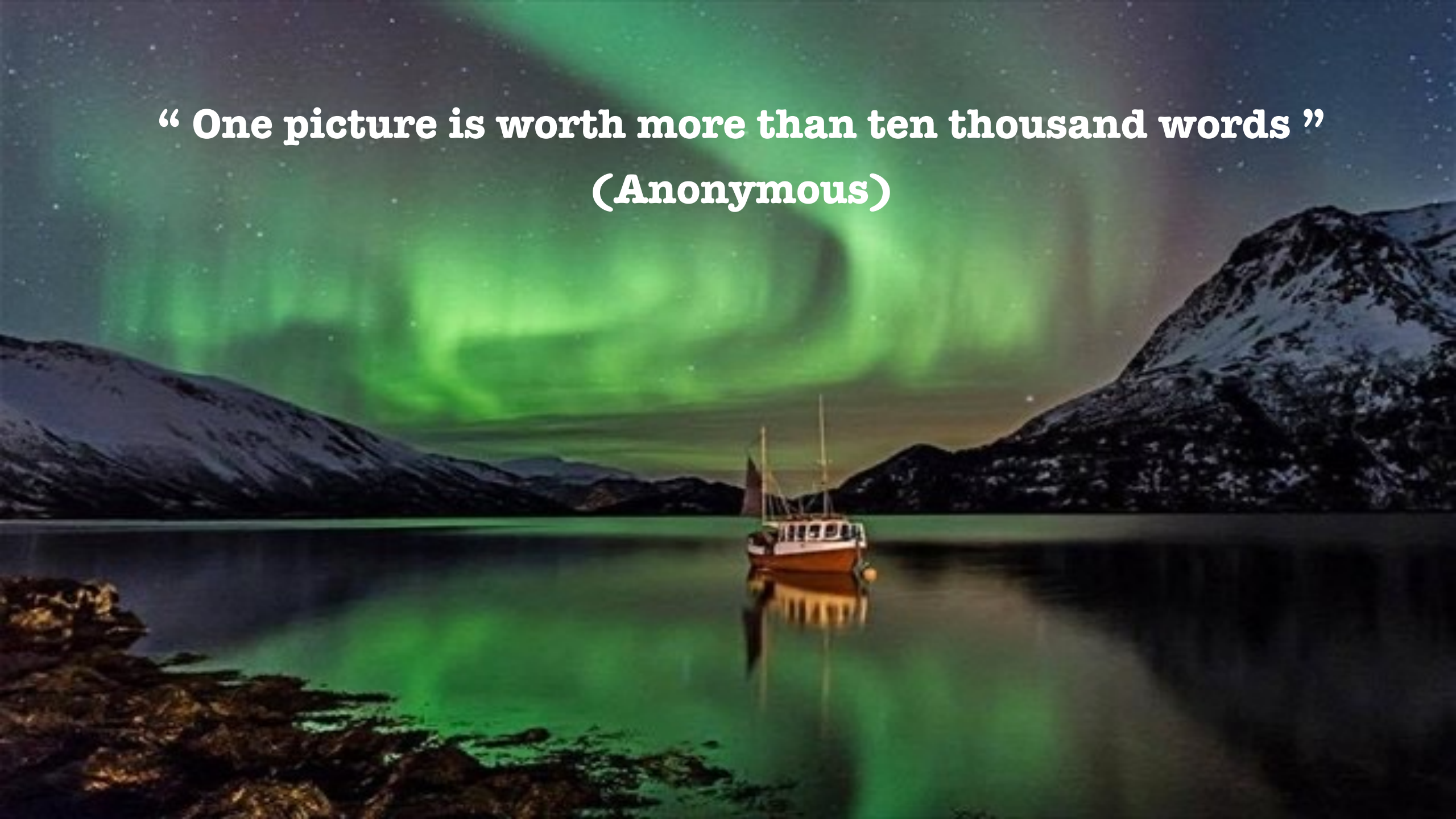
CAT, DOG, DUCK

**Instance
Segmentation**



CAT, DOG, DUCK

“ One picture is worth more than ten thousand words ”
(Anonymous)



Course Outline

- ✓ Digital Image Fundamentals
- ✓ Image Acquisition
- ✓ Image Transforms
- ✓ Convolution Theorem
- ✓ Morphological Operations
- ✓ Multiresolution Analysis
- ✓ Image Coding

How to capture an image?



Was it always
that easy to
capture an
image?



World's first photograph



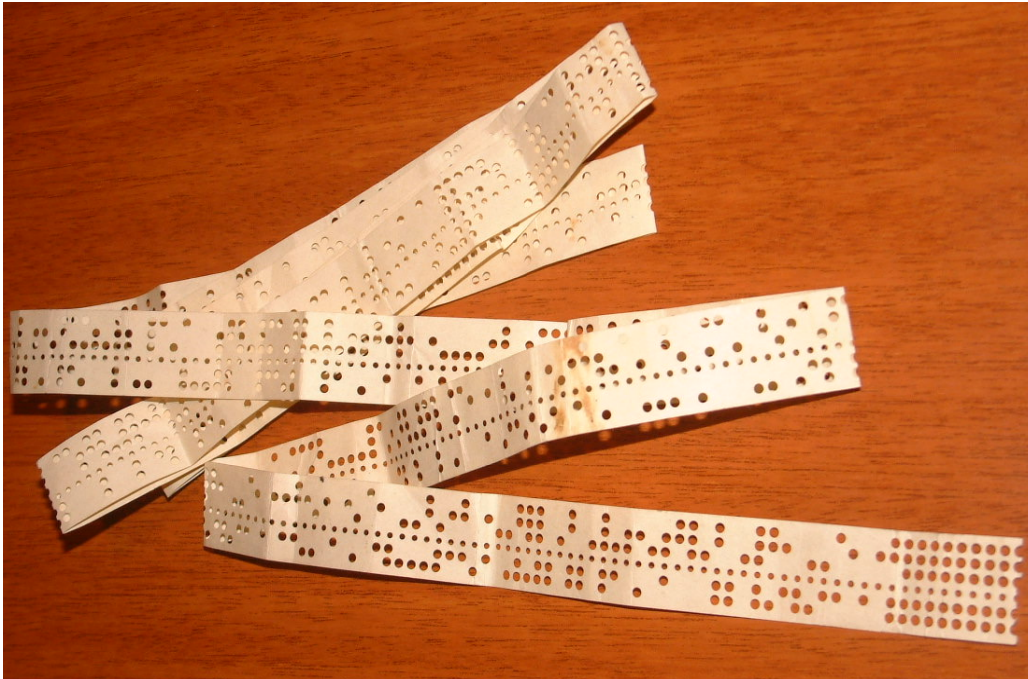
Joseph Nicéphore Niépce
(1765 - 1833) .

In 1826, Joseph Nicéphore Niépce took the world's first photograph with his a camera. The photograph was taken from the upstairs windows of Niépce's estate in the Burgundy region of France. This image was captured via a process known as **heliography**, which used Bitumen of Judea coated onto a piece of glass or metal; the Bitumen then hardened in proportion to the amount of light that hit it.



History

In early 1920's the Bartlane cable picture transmission system reduced the time required to transport a picture across the Atlantic from more than a **week** to less than three **hours**.



Baudot tape used to transmit pictures by Bartlane system.



Newspaper picture transmitted across the Atlantic.

The first Digital Image

In 1957, a team led by Russell A. Kirsch at the National Institute of Standards and Technology of US developed a **binary digital version** of an existing technology, so that alphanumeric characters, diagrams, photographs and other graphics could be transferred into digital computer memory. One of the first photographs scanned was a picture of Kirsch's infant son Walden.



Russell A. Kirsch
(1929 -2020)



First digital photo of Kirsch' son.

The first Moon picture

In 1964, the pictures of the **Moon** transmitted by Ranger 7 were processed by a computer at the jet Propulsion Laboratory (Pasadena, California) to correct various types of image distortion inherent in the on-board television camera.

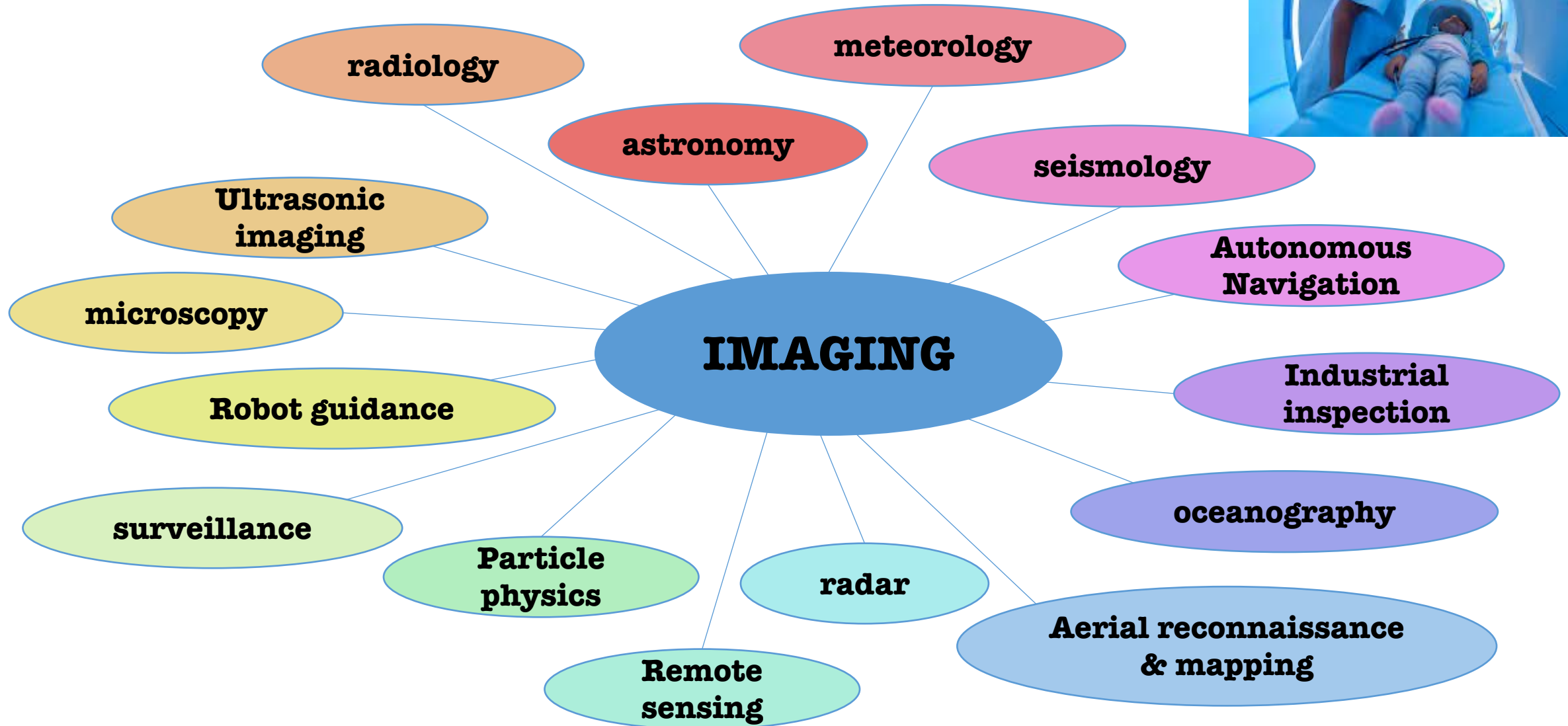


First Moon picture.

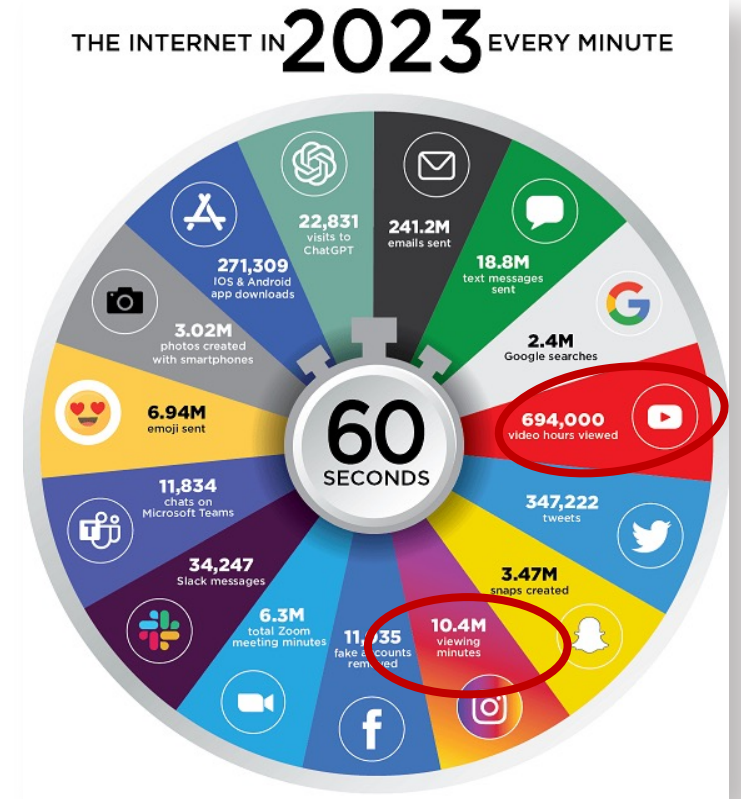
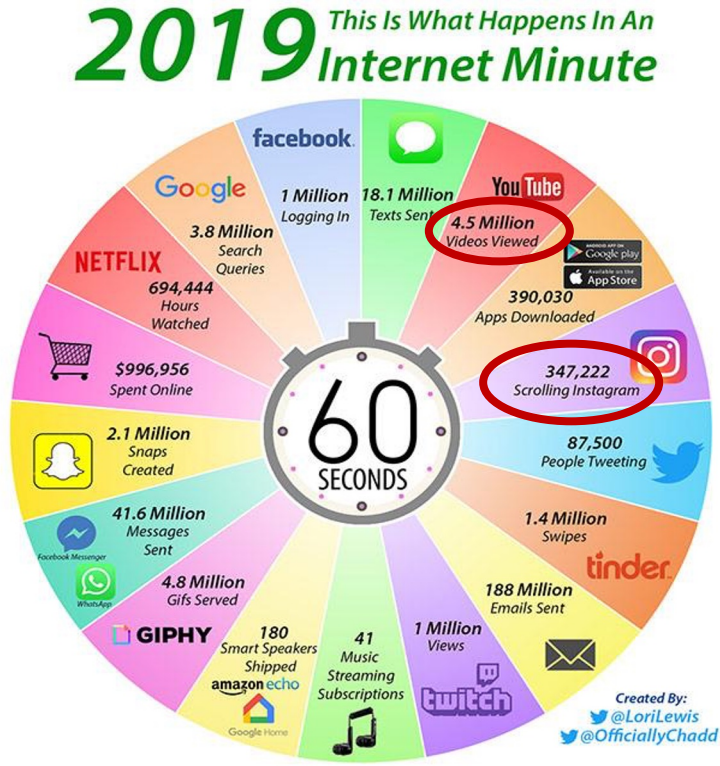


Drawing of the Ranger 7.

Applications



Big Data Era



[illegible]

What is an image?

Definition

- ✓ An **image** (from Latin: *imago*) is an artifact that depicts visual perception, such as a photograph or other two-dimensional **picture**, that resembles a subject—usually a physical object—and thus provides a depiction of it (*Wikipedia*).
- ✓ **Digital images** and **digital video** are, respectively, pictures and movies that have been converted into a computer-readable binary format consisting of logical 0s and 1s. Usually, by an image we mean a still picture that does not change with time, whereas a video evolves with time and generally contains moving and/or changing objects (Al Bovik, “Handbook of Image and Video Processing,” *Academic Press, 2000*).
- ✓ **Monochrome image** or simple image or still-image, refers to a two-dimensional light function $f(x,y)$ where x and y denote spatial coordinates and the value of f at any point (x,y) is proportional to the brightness of the image at that point. (R. Gonzalez, R. Woods, “Digital Image Processing,” *Addison-Wesley, 1983*).

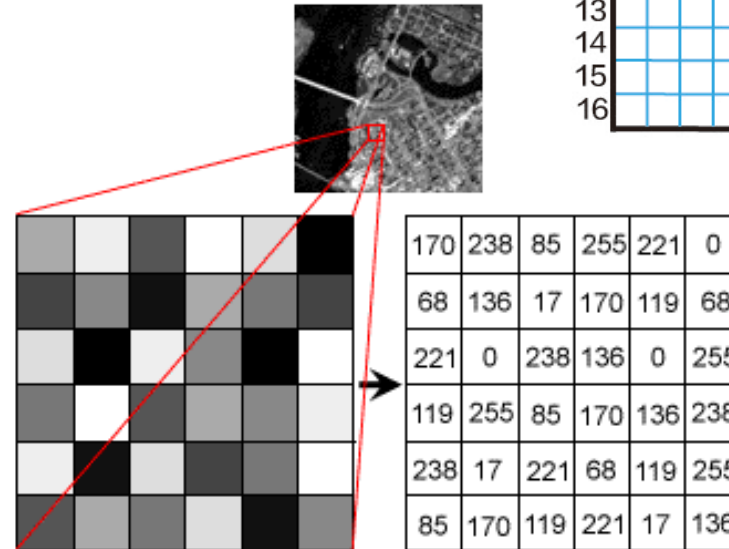
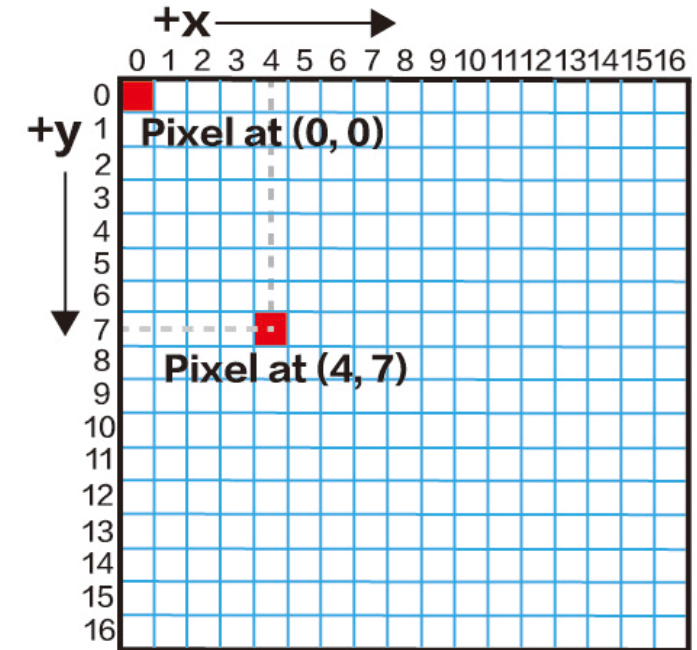
Digital Image Representation

- ✓ A **digital image** is an image $f(x,y)$ that has been discretized both in spatial coordinates and in brightness.
- ✓ You may consider a digital image as a **matrix** whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at that point.
- ✓ The elements of such a digital array are called *image elements*, *picture elements (pels)*, **pixels**.
- ✓ The discretized brightness values usually belong to the following interval

$$0 \leq f(x,y) \leq G - 1,$$

where $G = 2^m, m = 1, 2, \dots$ denotes the number of gray levels.

Cartesian Coordinates



Relationship between Pixels

- ✓ A pixel p at coordinates (x, y) has four **horizontal** and **vertical** neighbors whose coordinates are given by

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

- ✓ This set of pixels, called the **4-neighbors** of p , is denoted by $N_4(p)$.
- ✓ Each pixel is a unit distance from (x, y) , and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.
- ✓ The four **diagonal** neighbors of p have coordinates denoted by $N_D(p)$.

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

- ✓ These points, together with the 4-neighbors, are called the **8-neighbors** of p , denoted by $N_8(p)$

3	0	2
6	1	4
4	2	3

Gray Scale Images



Lena Image
Original image [here](#)



Lena Söderberg at ICIP Conference
in 2015

Gray Scale Images



- ✓ Let $I(i, j)$ where $i = 1, \dots, N$ and $j = 1, \dots, M$ be a discretized grayscale image of 2^m different gray levels.
- ✓ The number of **bits** required to store a digital image is given by

$$b = N \times M \times m.$$

- ✓ A group of 8-bits is commonly called 1 **Byte**. Thus, the number of Bytes required to store a digital image is given by

$$B = \frac{N \times M \times m}{8}.$$

- ✓ The **resolution** of an image is strongly dependent on N, M, m . The higher these parameters are increased, the closer the digitized image will approximate the original.

Image Resolution



Original Grayscale
Image
512x512



256x256



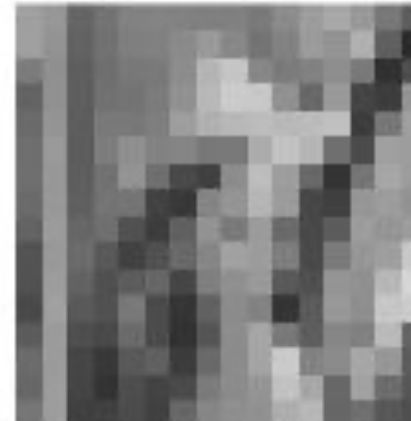
128x128



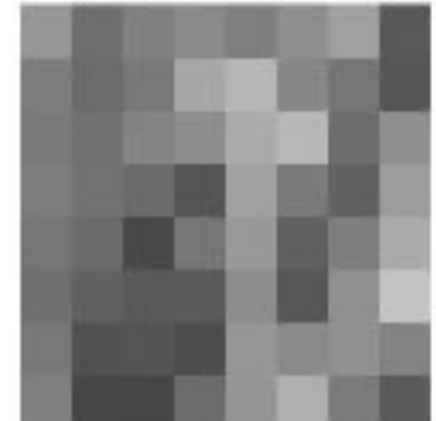
64x64



32x32

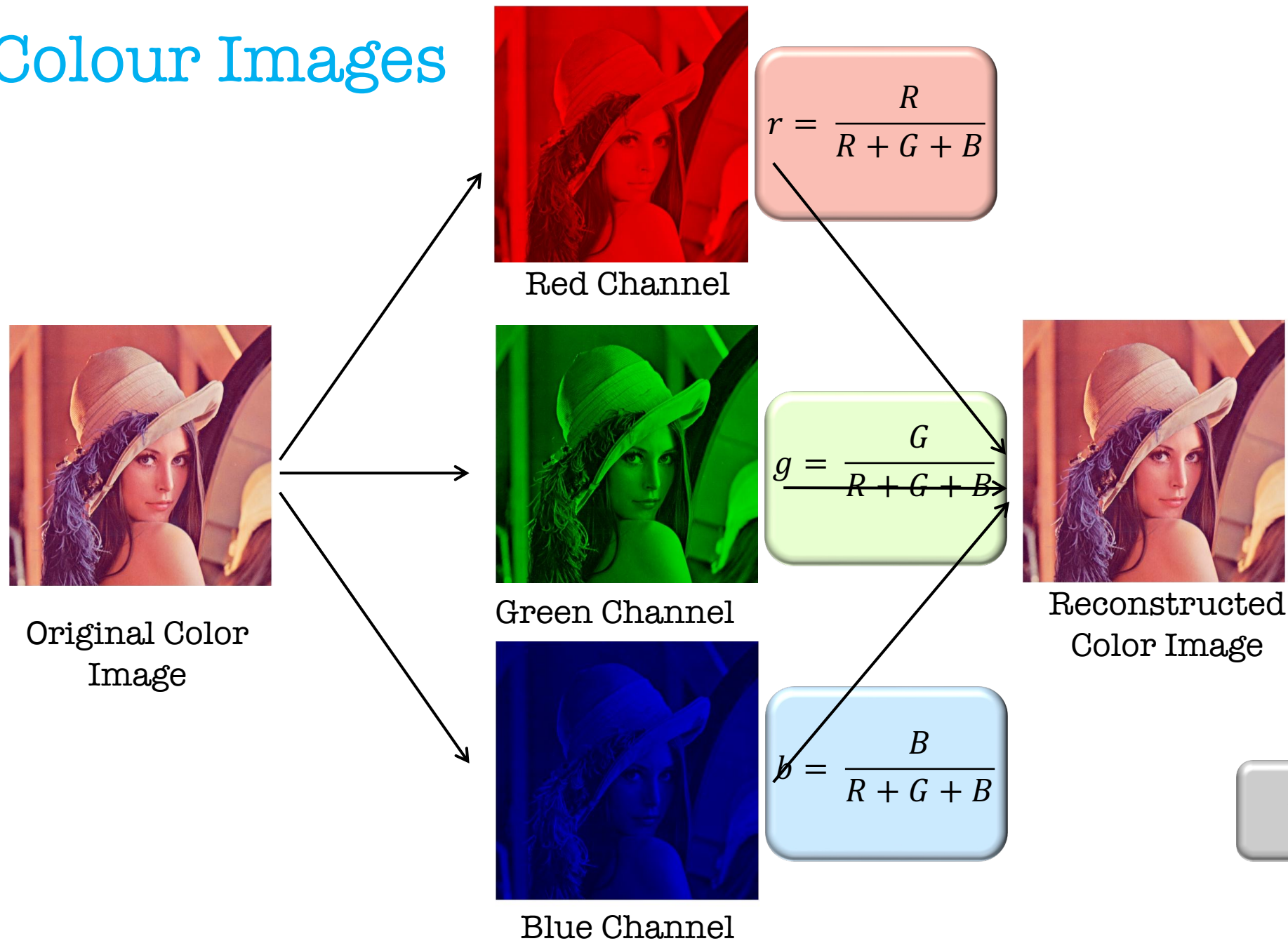


16x16



8x8

Colour Images



Resolution of Colour Images



Original Image
512x512



256x256



128x128



64x64



32x32



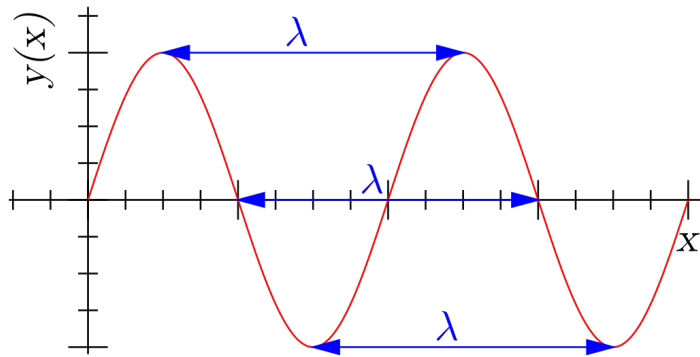
16x16



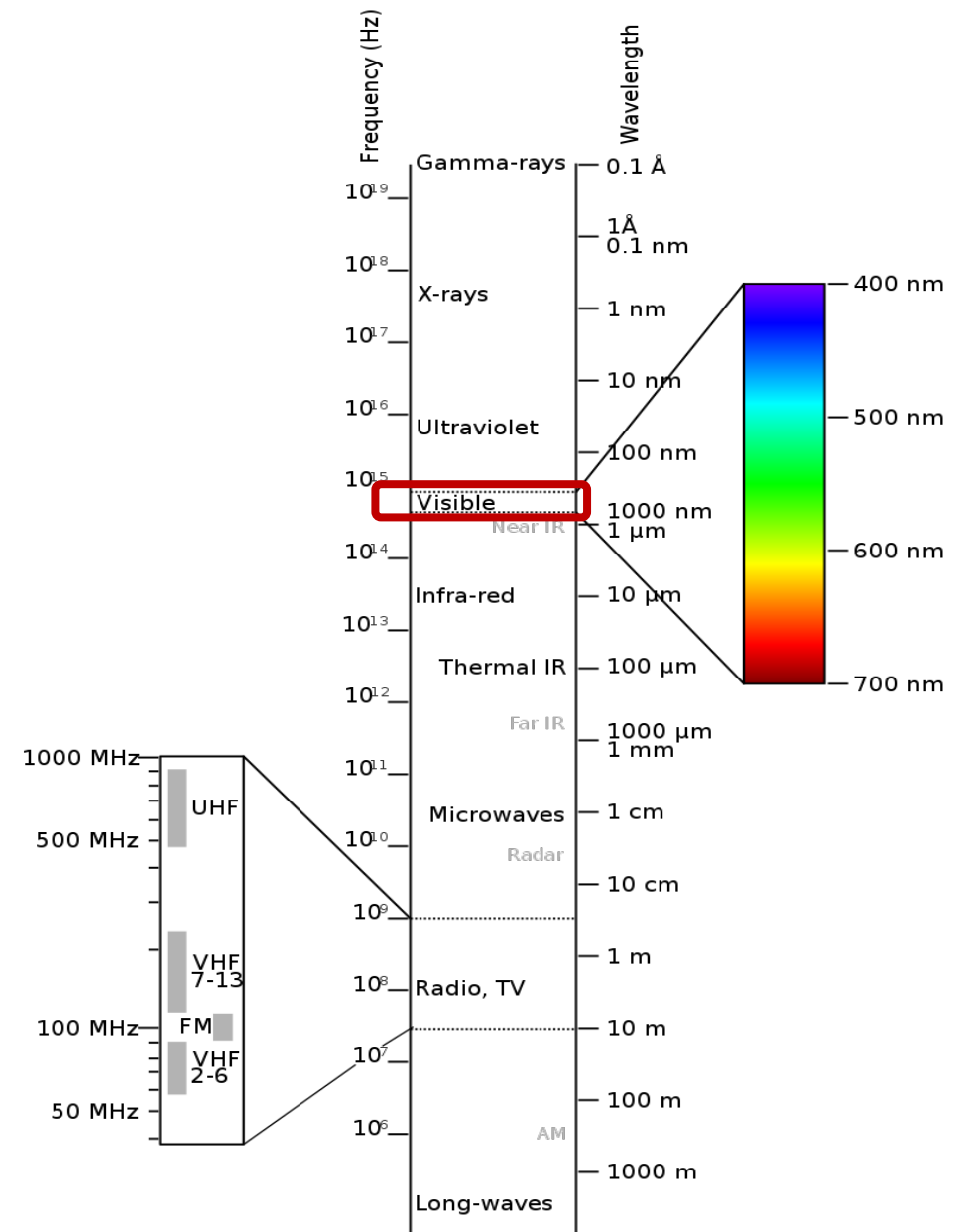
8x8

Electromagnetic Spectrum

- ✓ The electromagnetic spectrum covers electromagnetic waves with **frequencies** ranging from $1\text{-}10^{25}$ Hz, corresponding to **wavelengths** from 1000m to a fraction of the size of an atomic nucleus (1 Angstrom = 10^{-10} m = 0.1 nm).
- ✓ The frequency range is separated into bands; beginning at the low frequency / long-wavelength (radio, TV) until the high frequency / short wavelength (Gamma-rays).

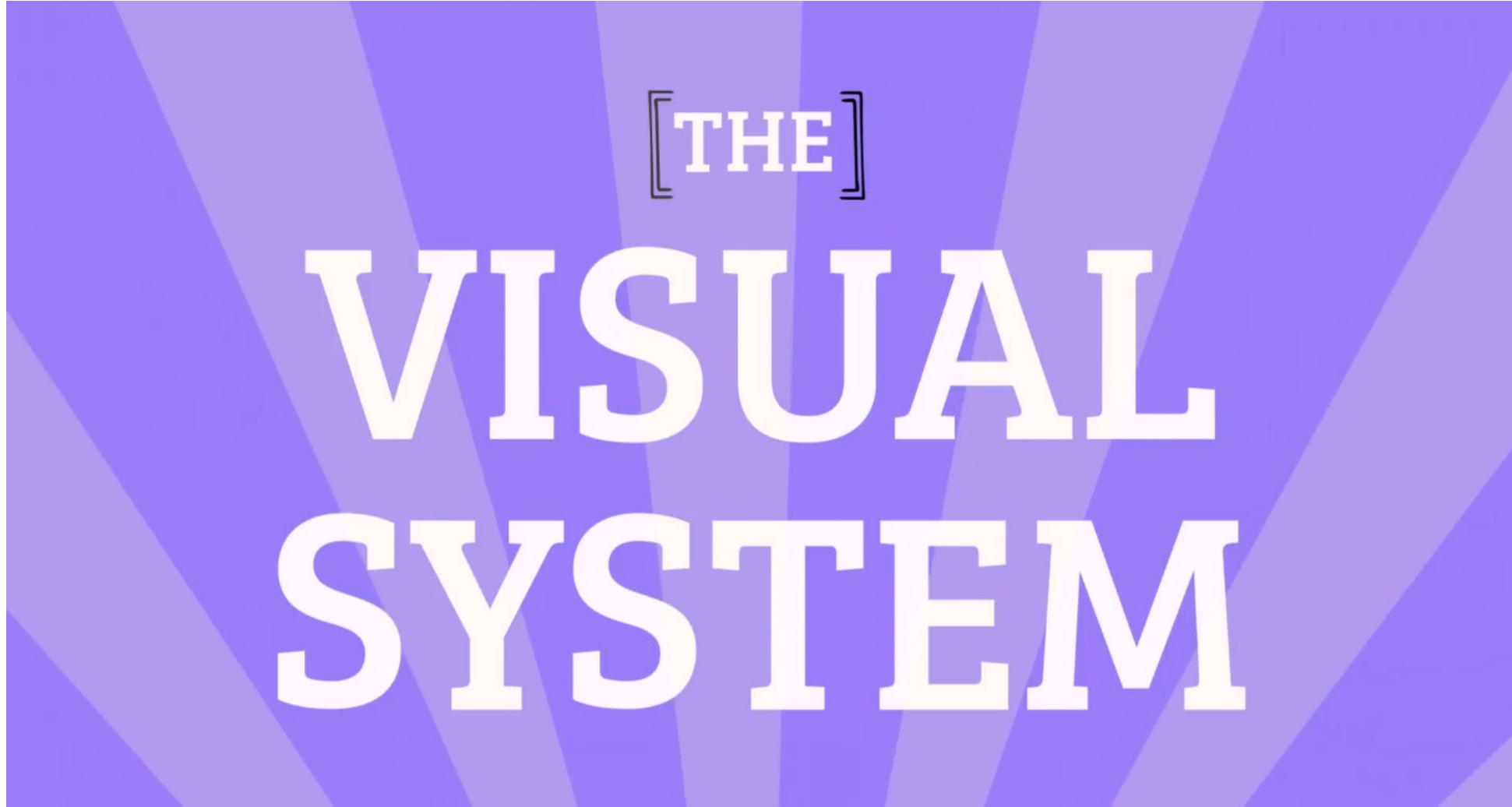


The wavelength of a sine wave.



Electromagnetic spectrum.

The Visual System



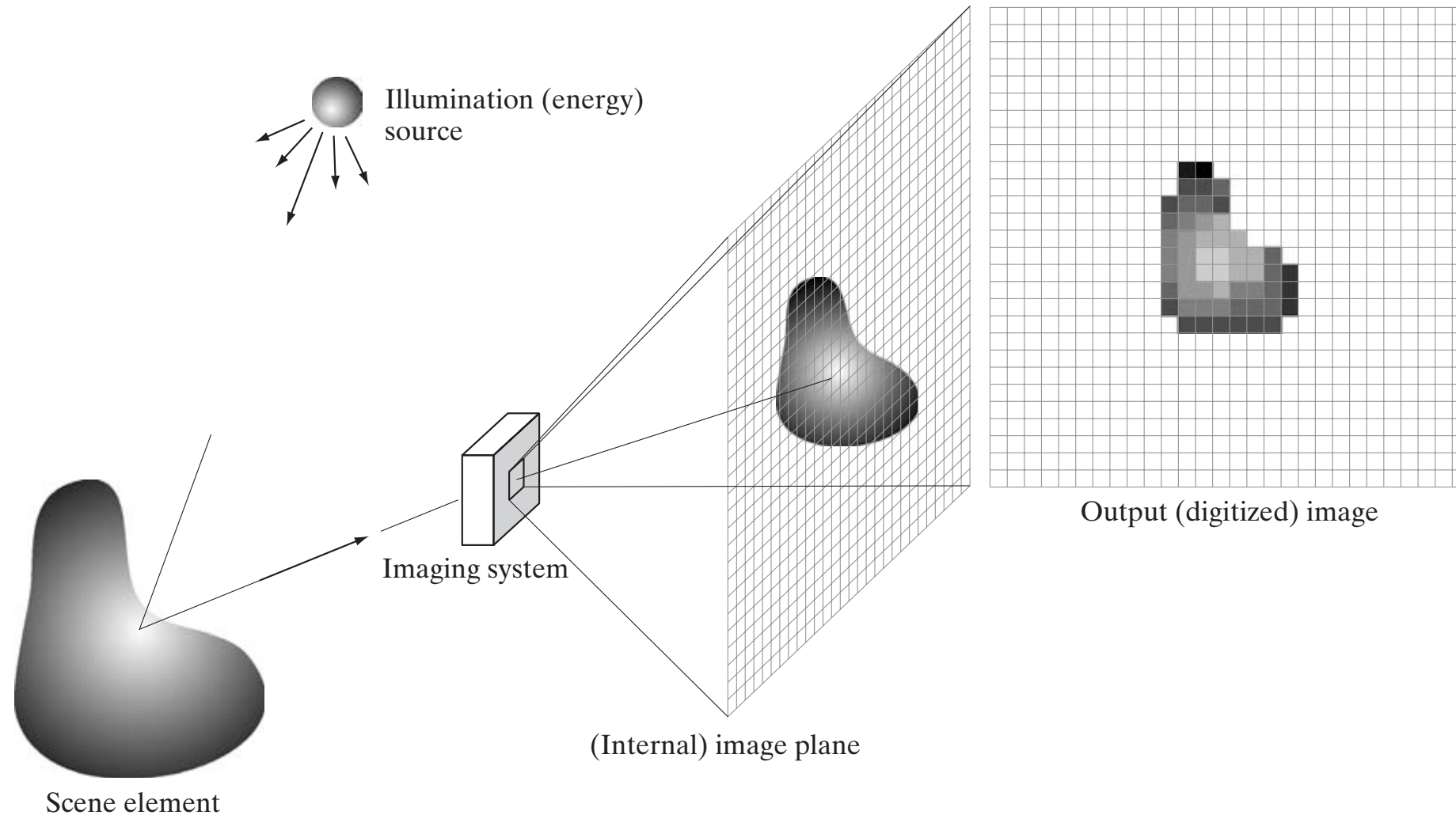
The Visual System

VISION

- ✓ Begins in the eye which receives the inputs, in the form of light.
- ✓ Finishes in the brain which interprets those inputs and gives us the information we need from the data we receive.



Example of the digital acquisition process



Sampling (statistics)

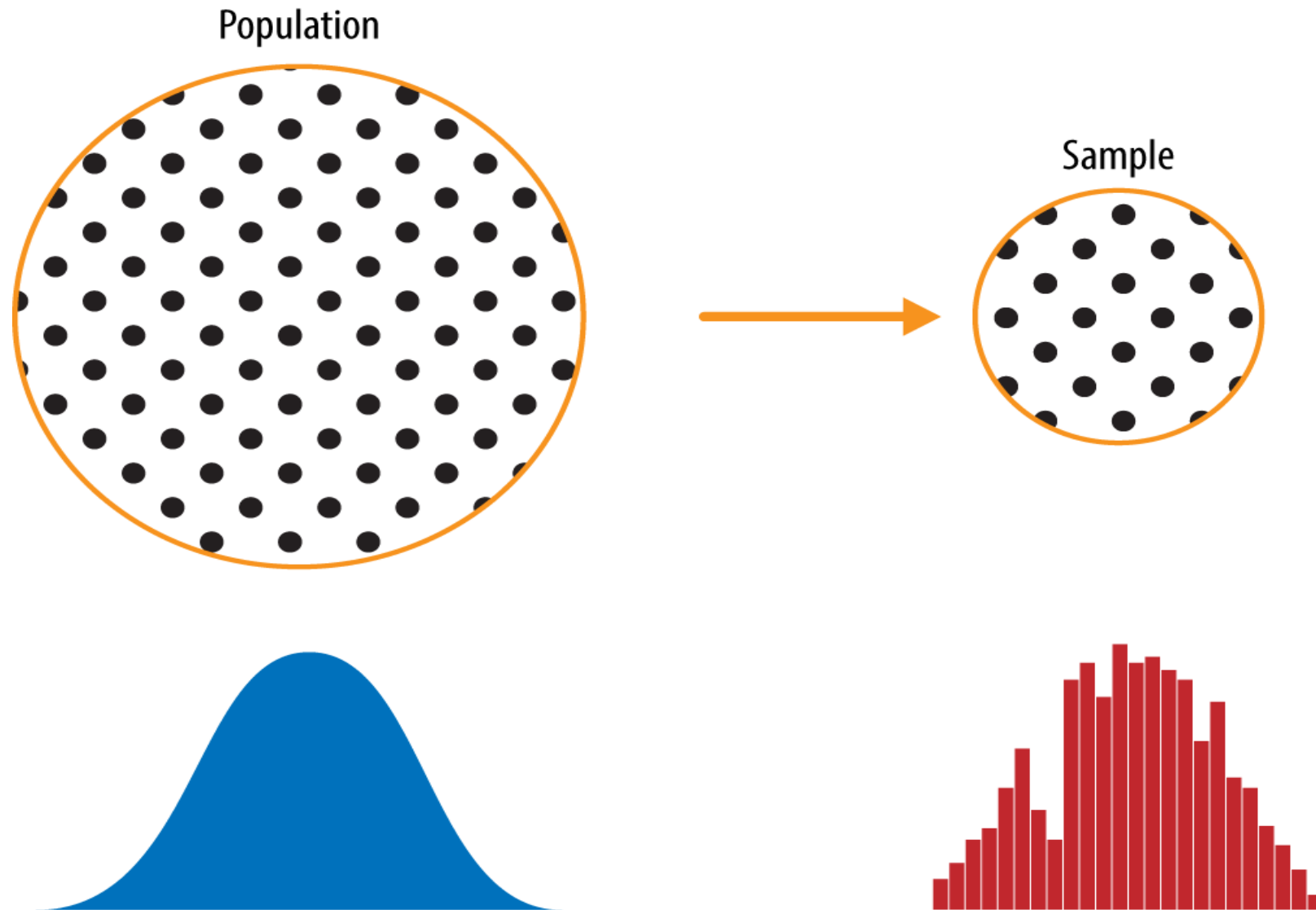
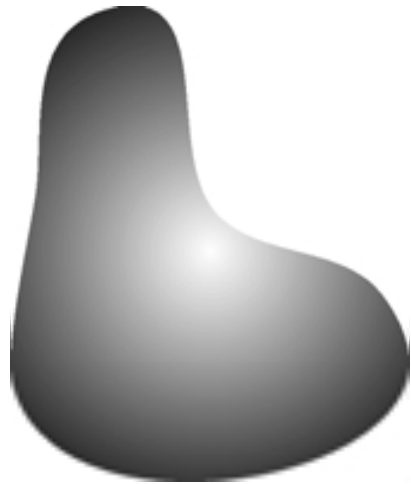
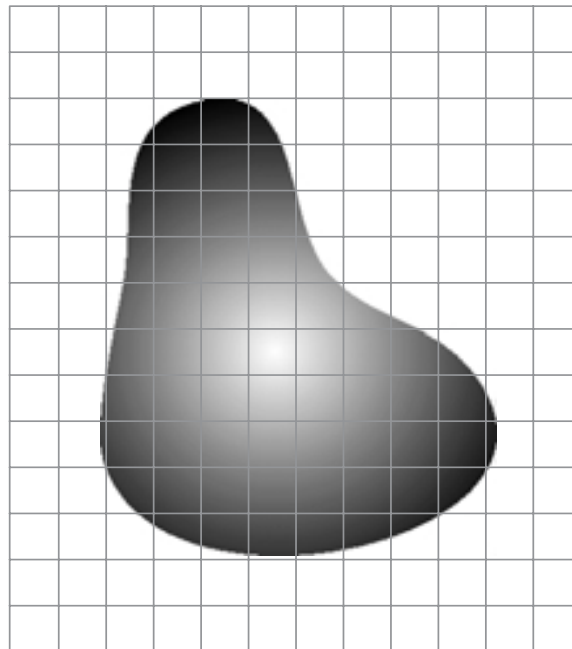


Image Sampling

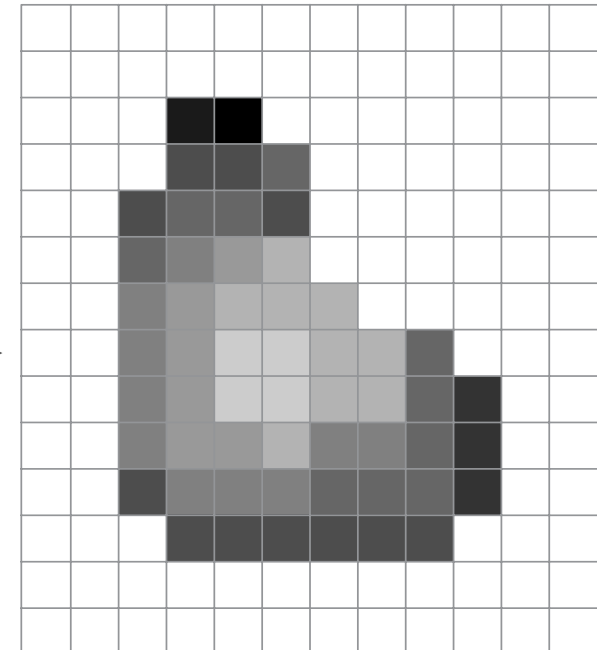
- ✓ An image may be continuous with respect to the x- and y-coordinates, and also in amplitude.
- ✓ To convert it to digital form, we have to sample the function in both coordinates and in amplitude.
- ✓ Digitizing the coordinate values is called sampling.
- ✓ Digitizing the amplitude values is called quantization.



Original Signal



Analog Signal

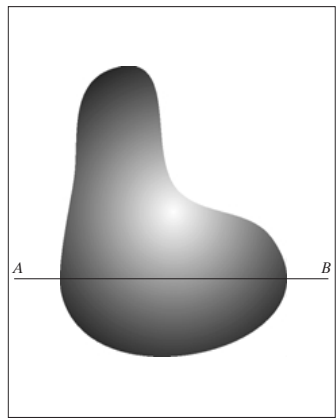


Digital Signal

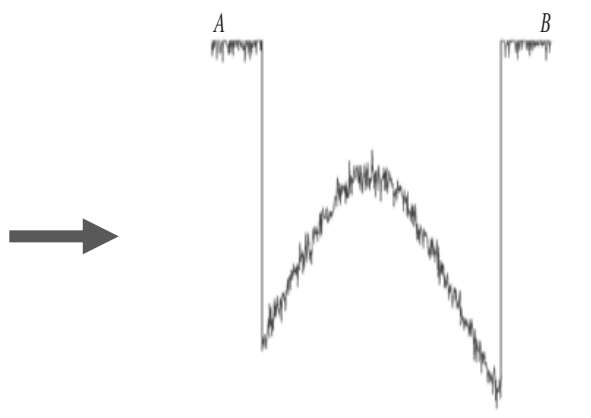
Image Sampling

We are interested in looking at the following questions:

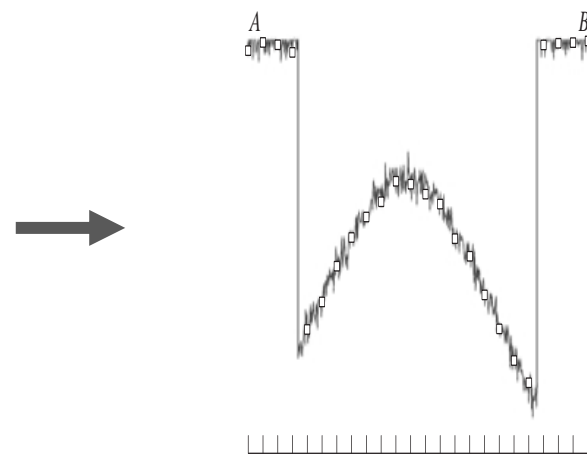
- ✓ How many samples should be taken so that no information is lost in the sampling process?
- ✓ Under what sampling conditions a continuous image can be fully recovered from a set of sampled values?



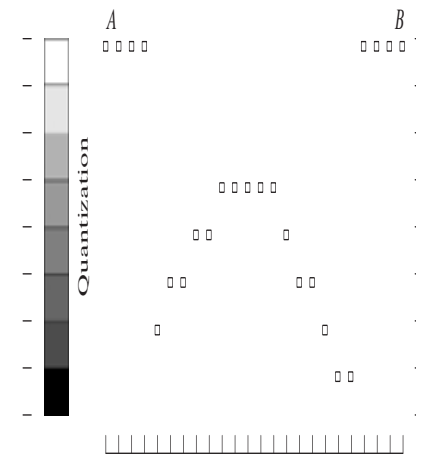
Original grayscale
Image



Illustrate the intensity
values of AB

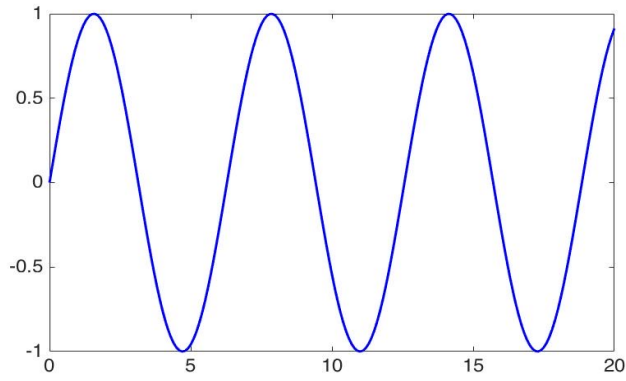


Sampling the
coordinate values

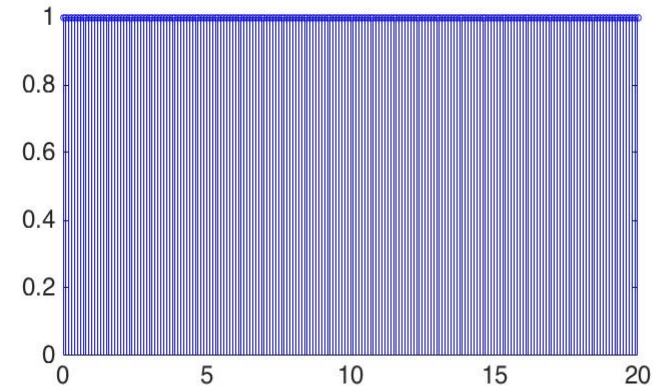


Digitize the
intensity values

Examples

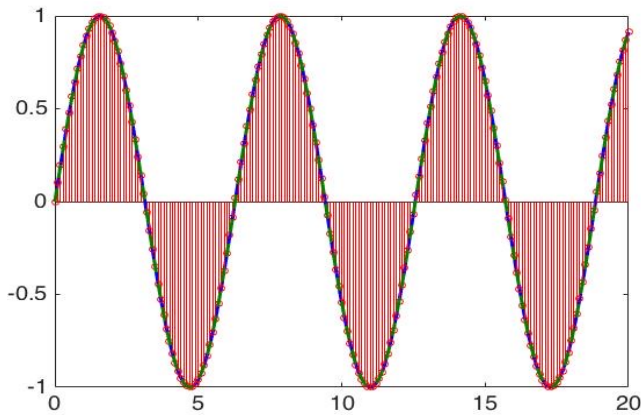


Input signal $f(x)$

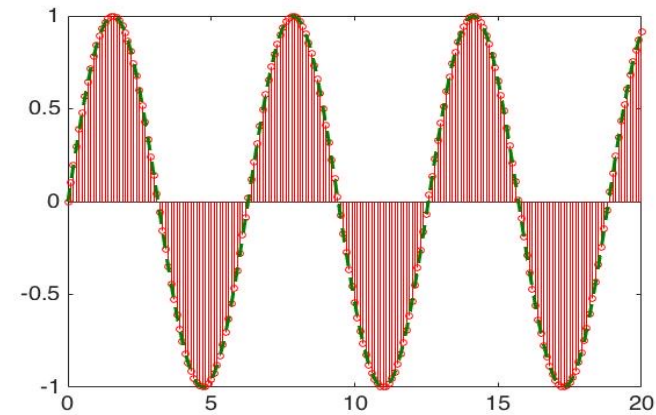


Sampling function $s(x)$

$\Delta x = 0.1$

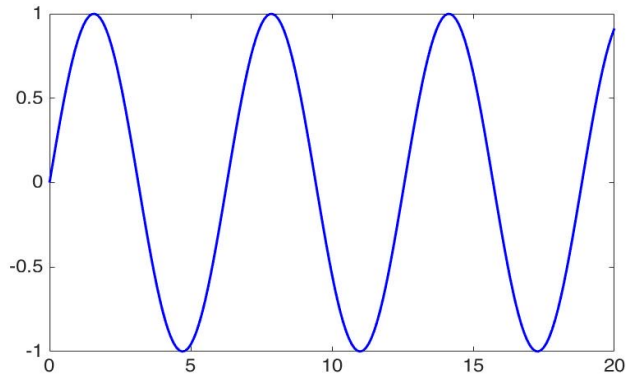


Superimpose $f(x), h(x)$

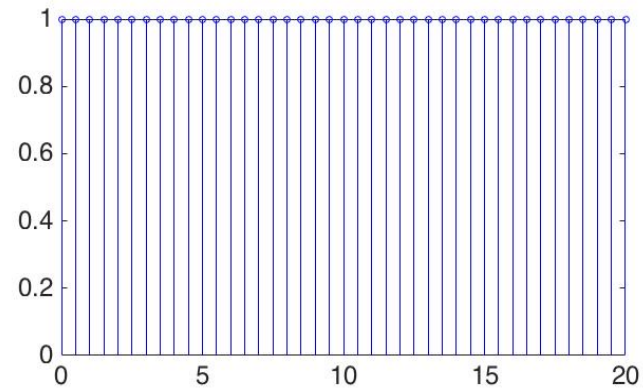


$h(x) = f(x)s(x)$

Examples

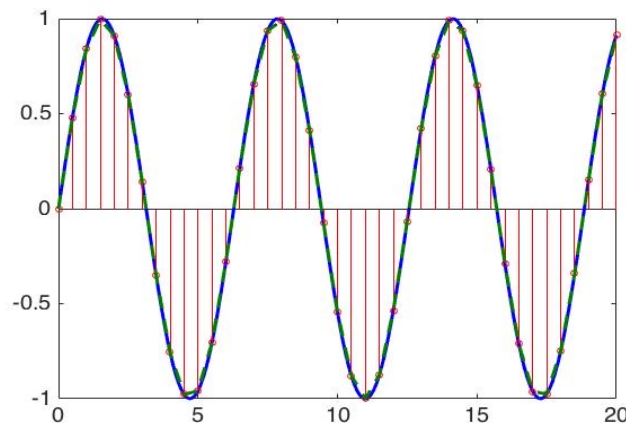


Input signal $f(x)$

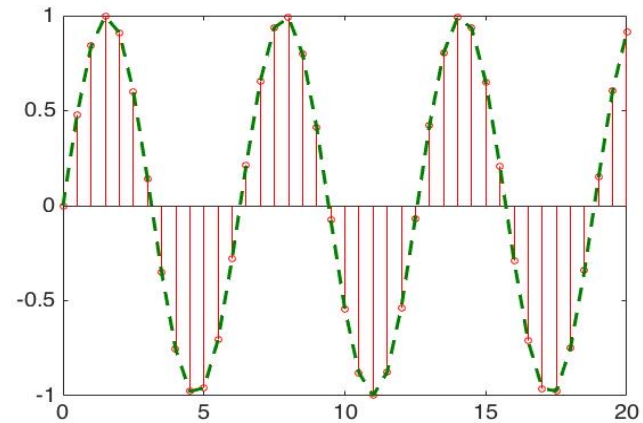


Sampling function $s(x)$

$\Delta x = 0.5$



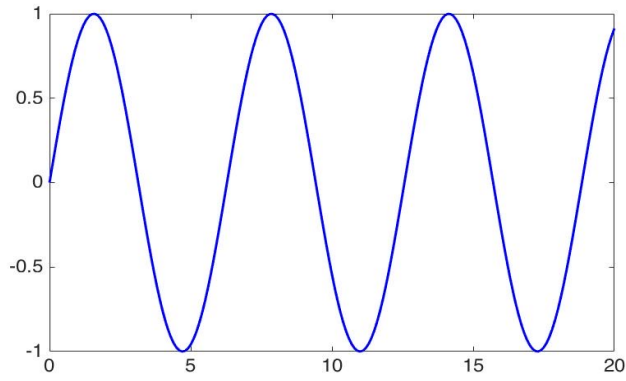
Superimpose $f(x), h(x)$



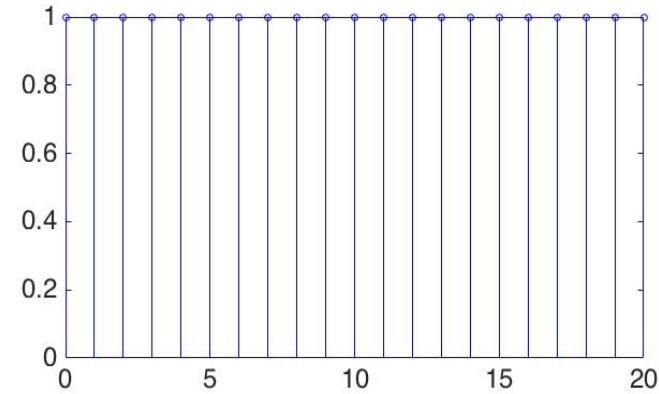
$h(x) = f(x)s(x)$



Examples

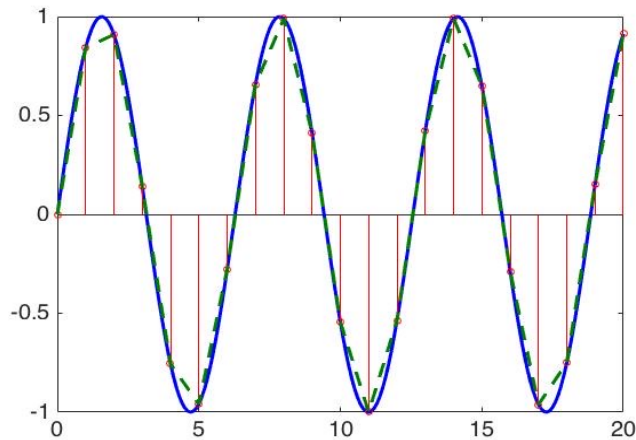


Input signal $f(x)$

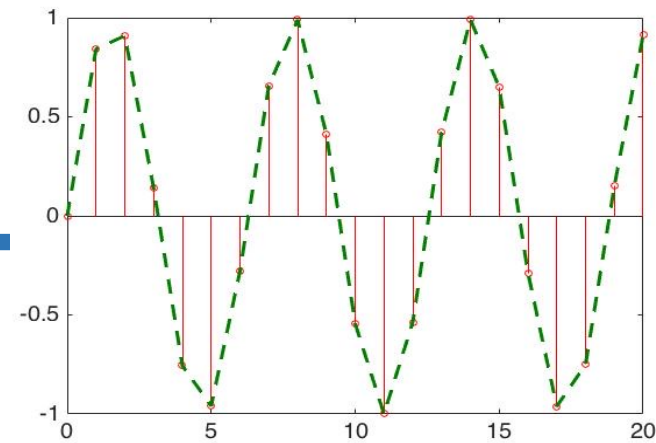


Sampling function $s(x)$

$$\Delta x = 1$$

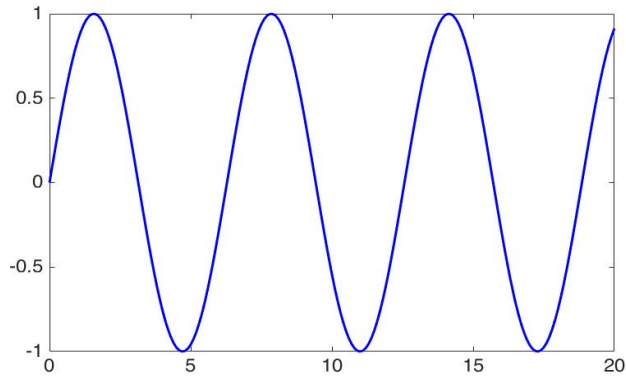


Superimpose $f(x), h(x)$

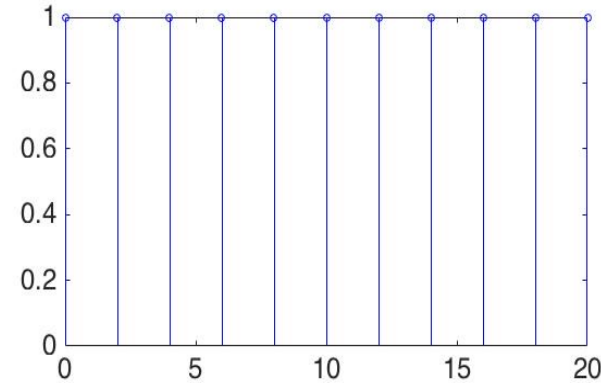


$$h(x) = f(x)s(x)$$

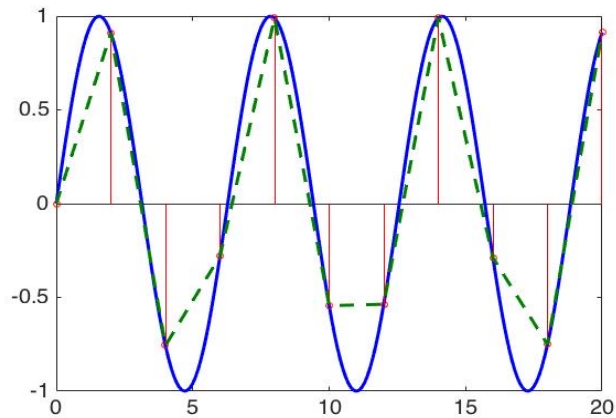
Examples



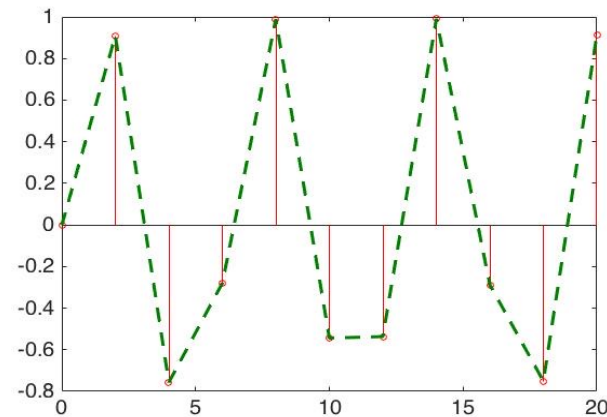
Input signal $f(x)$



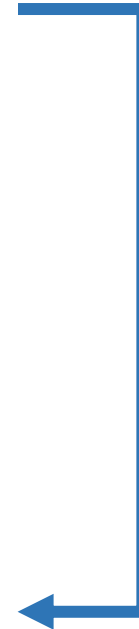
Sampling function $s(x)$
 $\Delta x = 2$



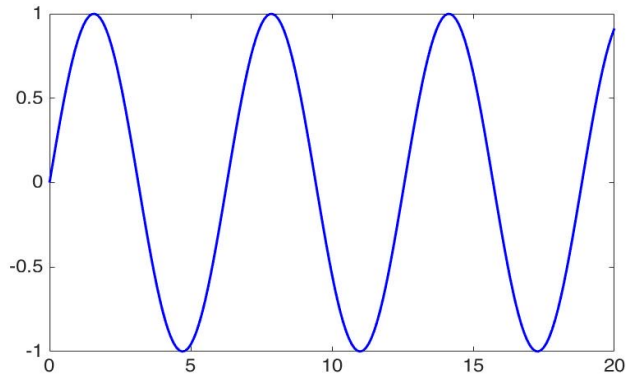
Superimpose $f(x), h(x)$



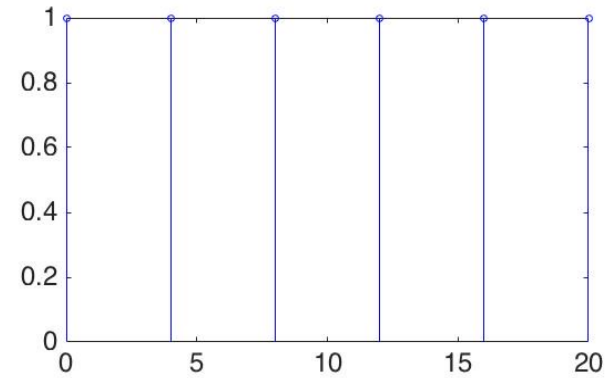
$h(x) = f(x)s(x)$



Examples

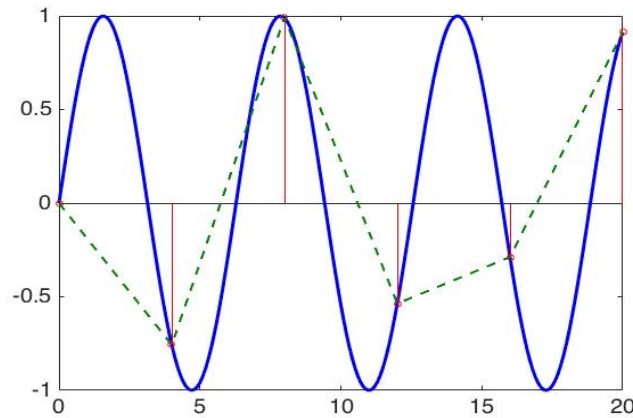


Input signal $f(x)$

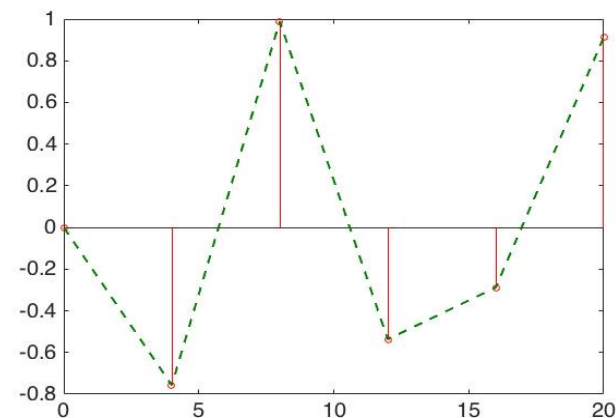


Sampling function $s(x)$

$$\Delta x = 4$$



Superimpose $f(x), h(x)$



$$h(x) = f(x)s(x)$$

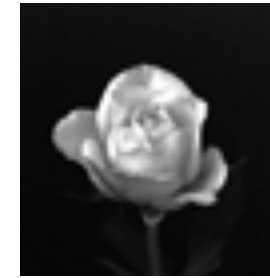
Image Subsampling (256 gray levels)



1024



512



256



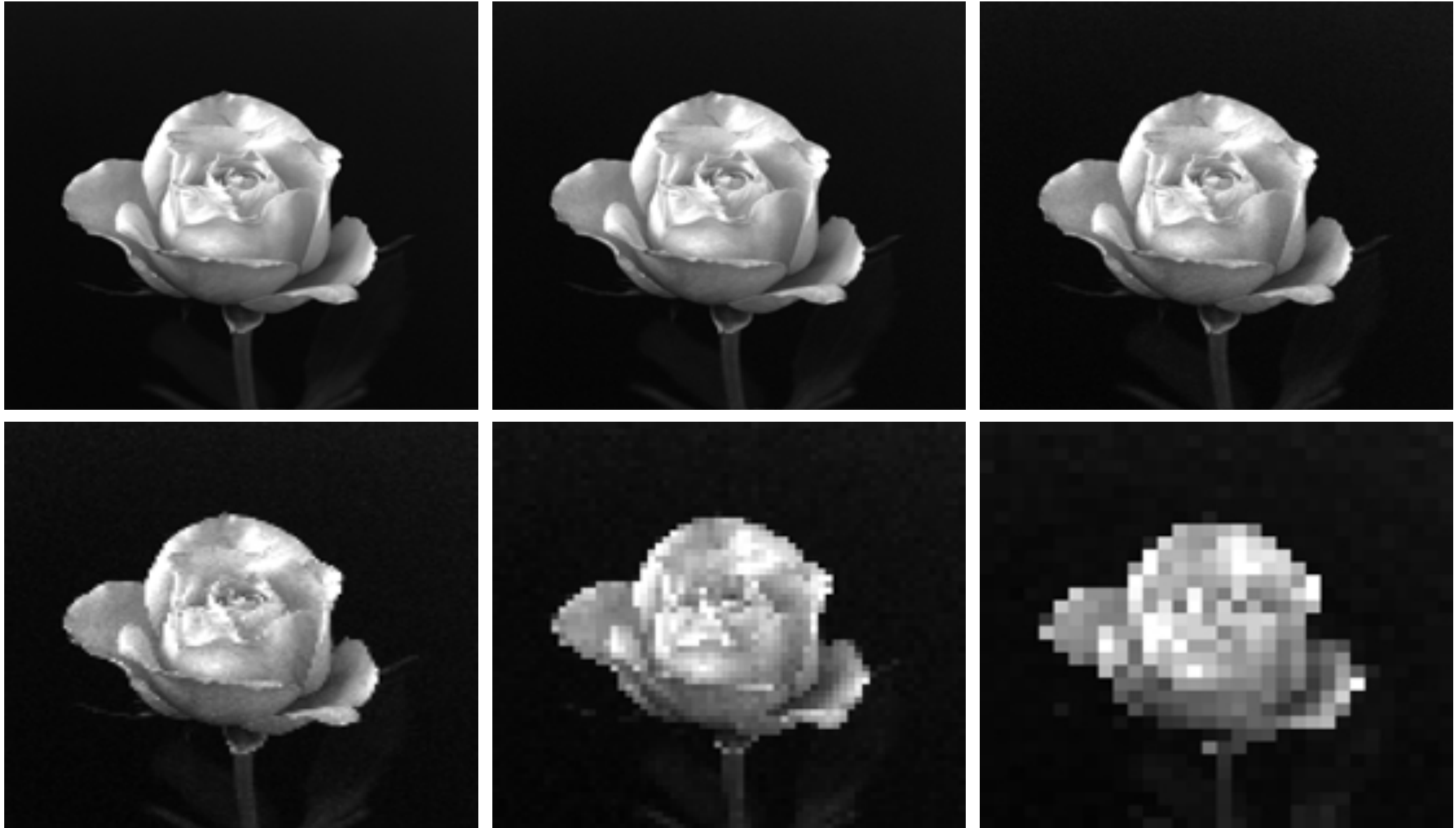
128



32

64

Image Resampling



Effect of Reducing the Spatial Resolution



1250 dpi



300 dpi



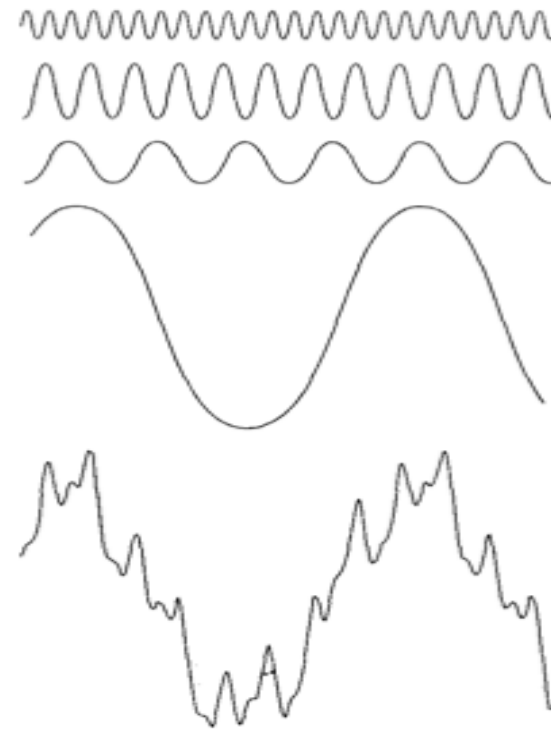
150 dpi



72 dpi

Aliasing

- ✓ Functions whose area under the curve is finite can be represented in terms of sines and cosines of various frequencies.
- ✓ The sine/cosine component with the highest frequency determines the highest “frequency content” of the function.
- ✓ Suppose that this highest frequency is finite and that the function is of unlimited duration (these functions are called band-limited functions).



Fourier J., “The Analytic Theory of Heat,” (1807).

Aliasing

Nyquist - Shannon Sampling Theorem

- ✓ If the function is sampled at a rate equal to or greater than twice its highest frequency, it is possible to recover completely the original function from its samples.
- ✓ If the function is undersampled, then a phenomenon called aliasing corrupts the sampled image.
- ✓ The corruption is in the form of additional frequency components being introduced into the sampled function. These are called aliased frequencies.

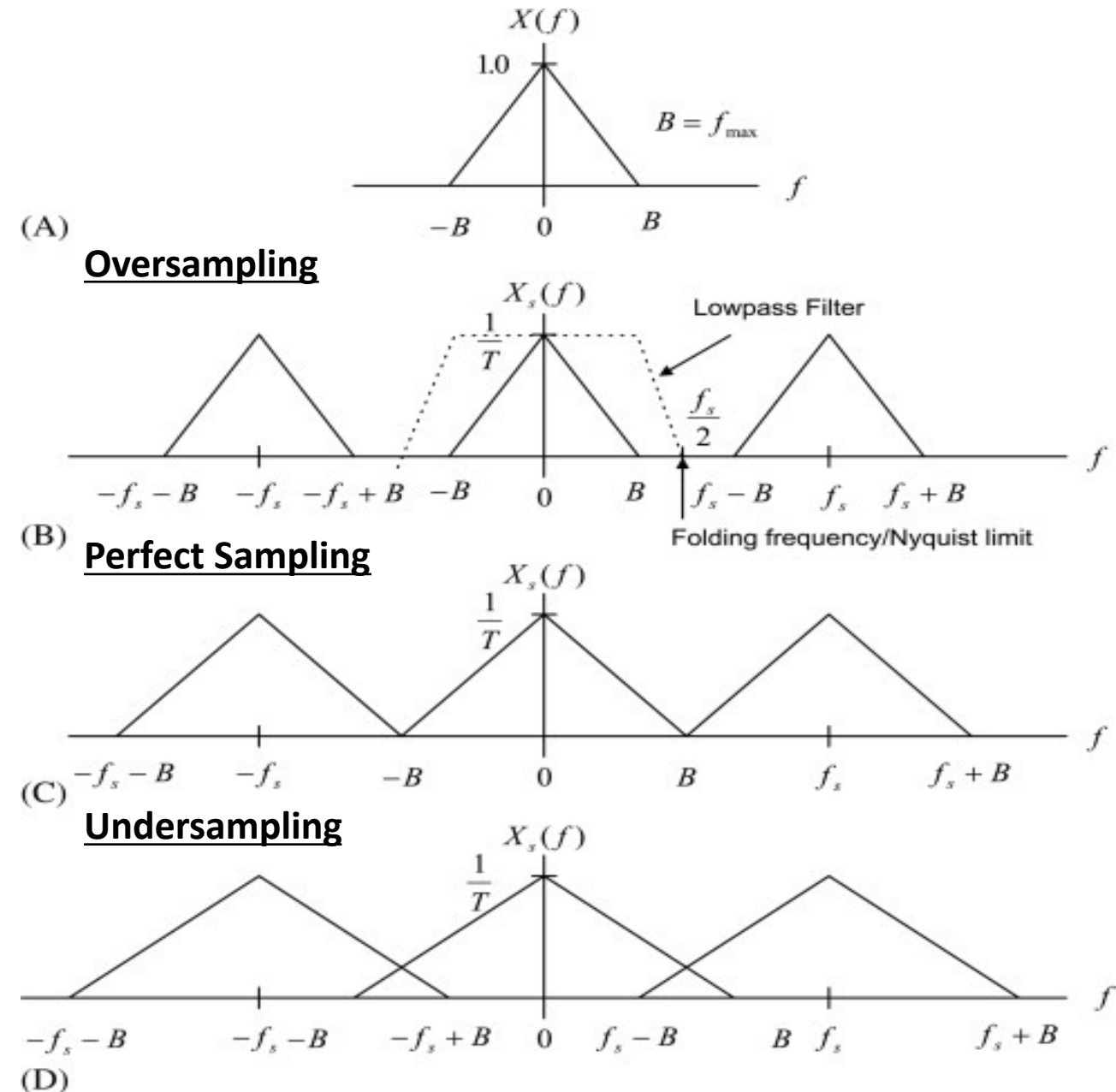
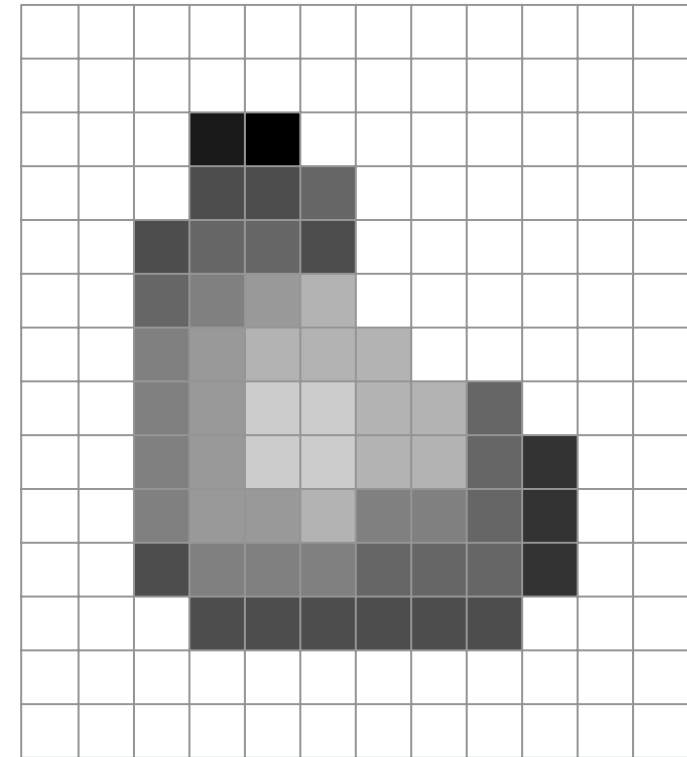


Image Quantization

- ✓ In order to form a digital image, the gray-level values also must be converted (quantized) into discrete quantities.
- ✓ The gray-level scale divided into L discrete levels, ranging from black to white
- ✓ The continuous gray levels are quantized simply by assigning one of the L discrete gray levels to each sample.
- ✓ Due to processing, storage, and sampling hardware considerations, the number of gray levels typically is an integer power of 2:

$$L = 2^k.$$



Number of storage bits

N/k	1 ($L = 2$)	2 ($L = 4$)	3 ($L = 8$)	4 ($L = 16$)	5 ($L = 32$)	6 ($L = 64$)	7 ($L = 128$)	8 ($L = 256$)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

N is the number of pixels

L is the number of the discrete intensity levels

Effect of Reducing the Intensity Levels



256 levels



128 levels



64 levels

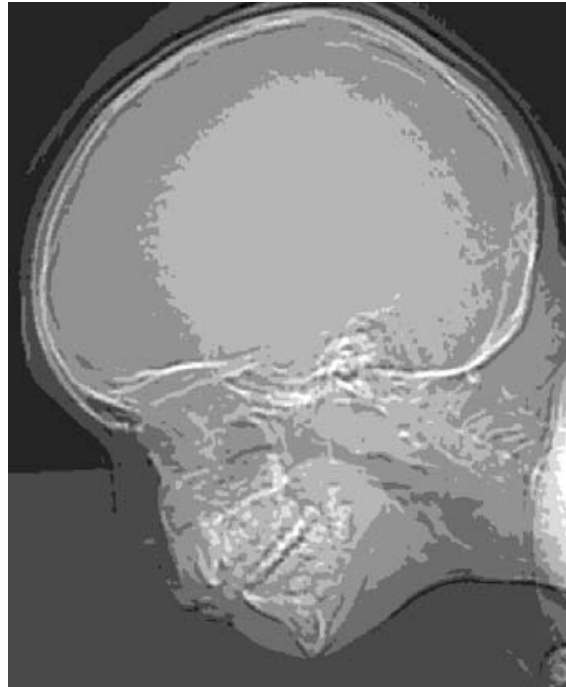


32 levels

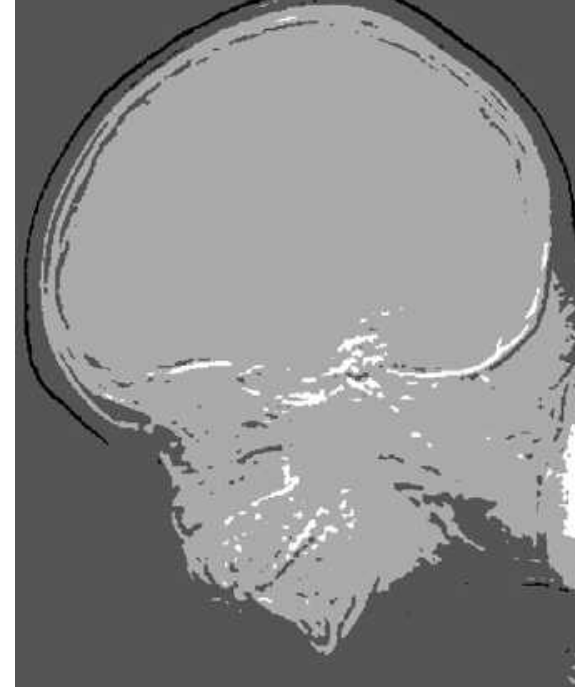
Effect of Reducing the Intensity Levels



16 levels



8 levels



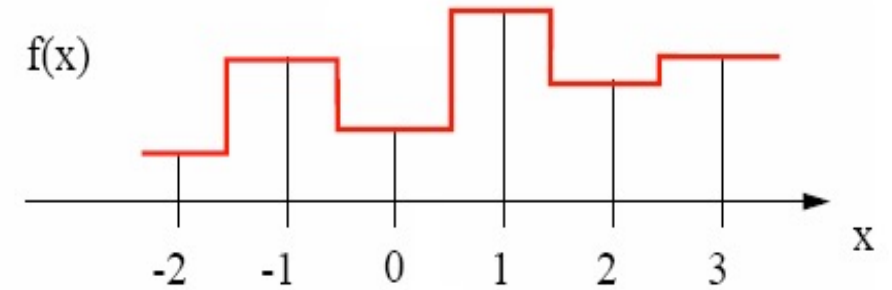
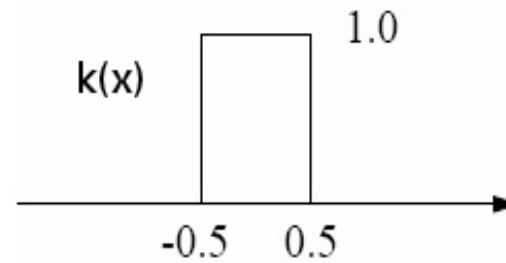
4 levels



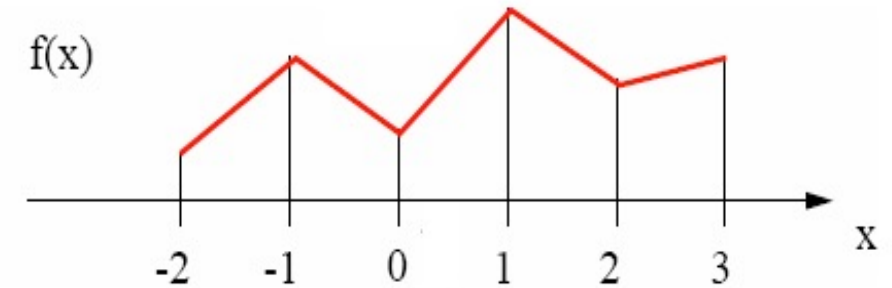
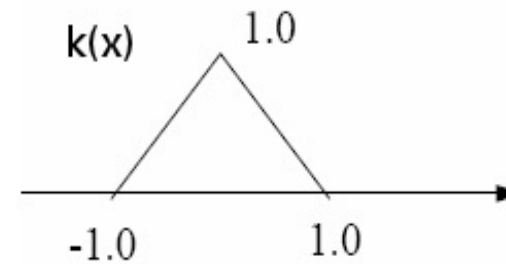
2 levels

Interpolation

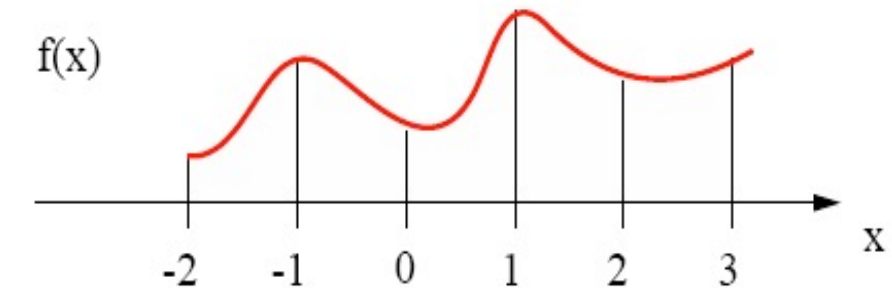
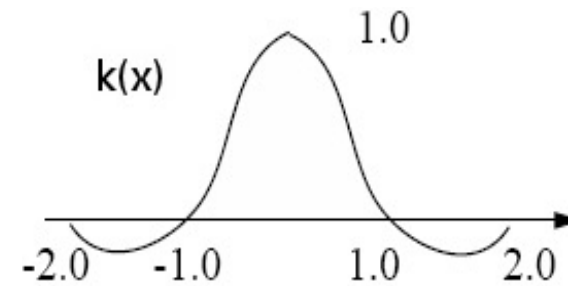
Nearest Neighbor
Resampling



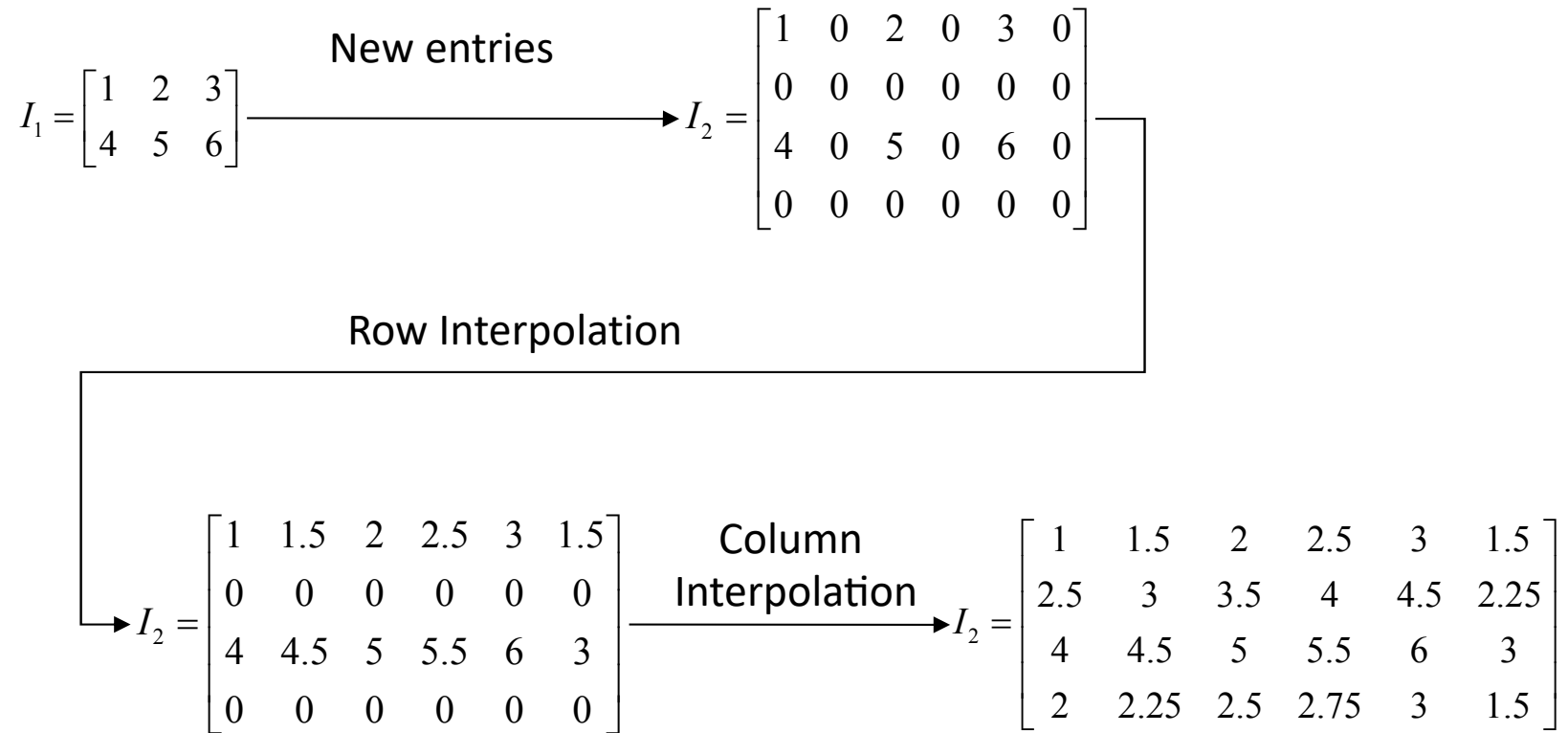
Bilinear
Resampling



Bicubic
Resampling

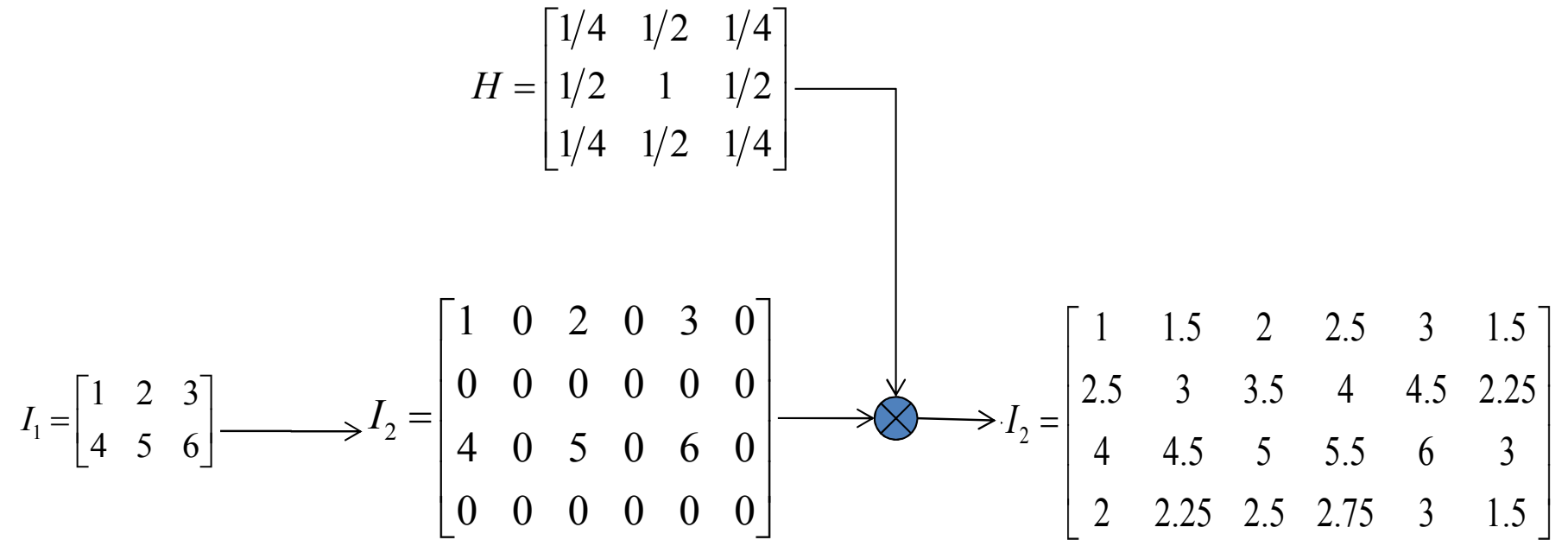


Linear Interpolation - Example



Linear Interpolation - Example

Row and Column Interpolation is equivalent to the convolution of the augmented matrix I_2 with the kernel H !!!!



Course Outline

- ✓ Digital Image Fundamentals
- ✓ Image Acquisition
- ✓ Image Transforms
- ✓ Convolution Theorem
- ✓ Morphological Operations
- ✓ Image Coding

Image Enhancement

- ✓ There is **no general theory** of image enhancement.
- ✓ The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application!!
- ✓ Establishing new image enhancement techniques is very much problem oriented e.g. a method which is quite useful for enhancing X-ray images may not be necessary for satellite images.
- ✓ The viewer is the ultimate assessor of how well an image enhancement technique works.

Image Enhancement Categories

- ✓ **Spatial Domain**

- ✓ This term refers to the image plane itself.
- ✓ Direct manipulations of pixels in an image.

- ✓ **Frequency Domain**

- ✓ Modifying the Fourier transform of an image.

Reminder: Joseph Fourier developed a theory by proving that complicated but periodic functions can be written as the sum of sines and cosines.

Spatial Domain

- ✓ It refers to the aggregate of pixels composing an image as most of the methods are procedures that operate directly on these pixels.
- ✓ It is denoted by the following expression

$$g(x, y) = T[f(x, y)],$$

where $f(x, y)$ is the input image, $g(x, y)$ is the new image and T is an operator on f defined over some neighbourhood.

- ✓ In addition, T can operate on a *set* of input images, such as performing the pixel-by-pixel sum of K images for noise reduction.

Pixel Neighborhood

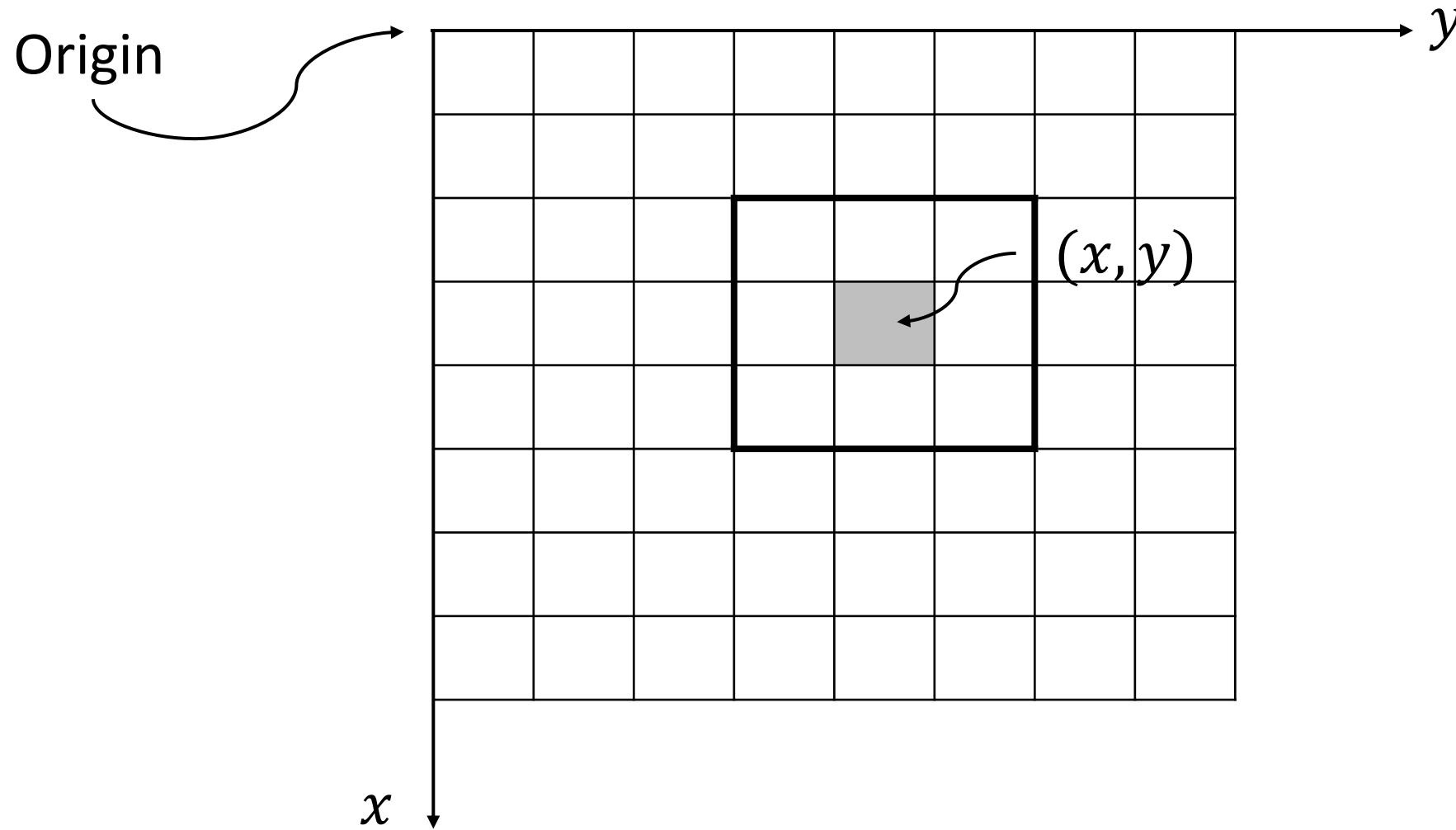


Fig. The 3x3 neighborhood of the pixel (x, y) in an 8x8 image.

Pixel Neighborhood

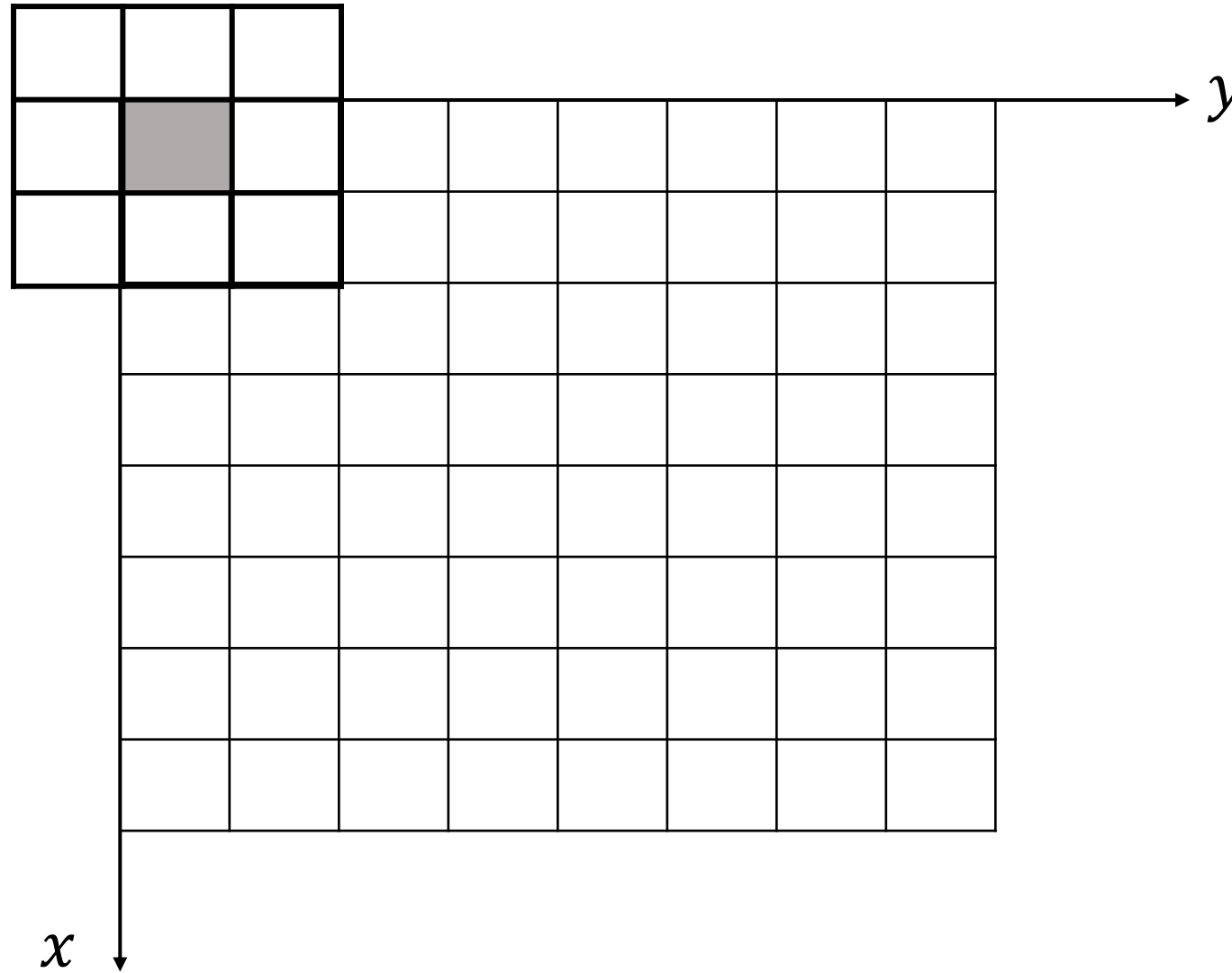


Fig. The 3x3 neighborhood of the pixel (x, y) in an 8x8 image.

Transformation function T

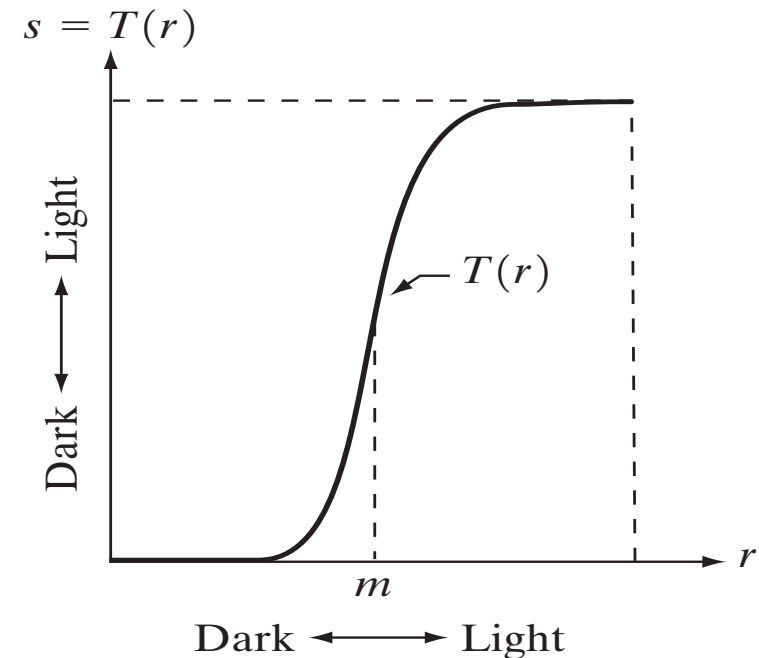
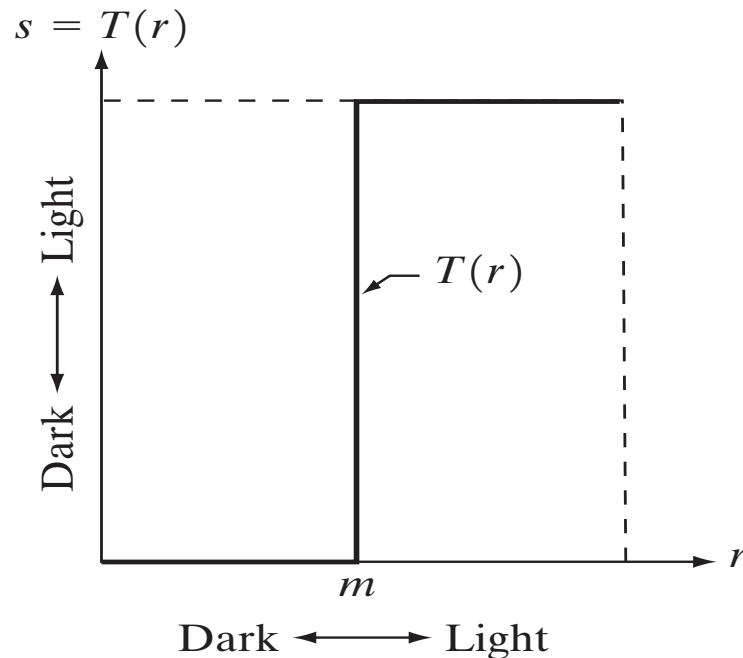
- ✓ The simplest form of T is when the neighborhood is of size 1×1 (a single pixel).
- ✓ In this case, g depends only on the value of f at (x, y) , and T becomes a *gray-level* (also called an *intensity* or *mapping*) *transformation function* of the form

$$s = T(r),$$

where, for simplicity in notation, r and s are variables denoting, respectively, the gray level of $f(x, y)$ and $g(x, y)$ at any point (x, y) .

Thresholding Function

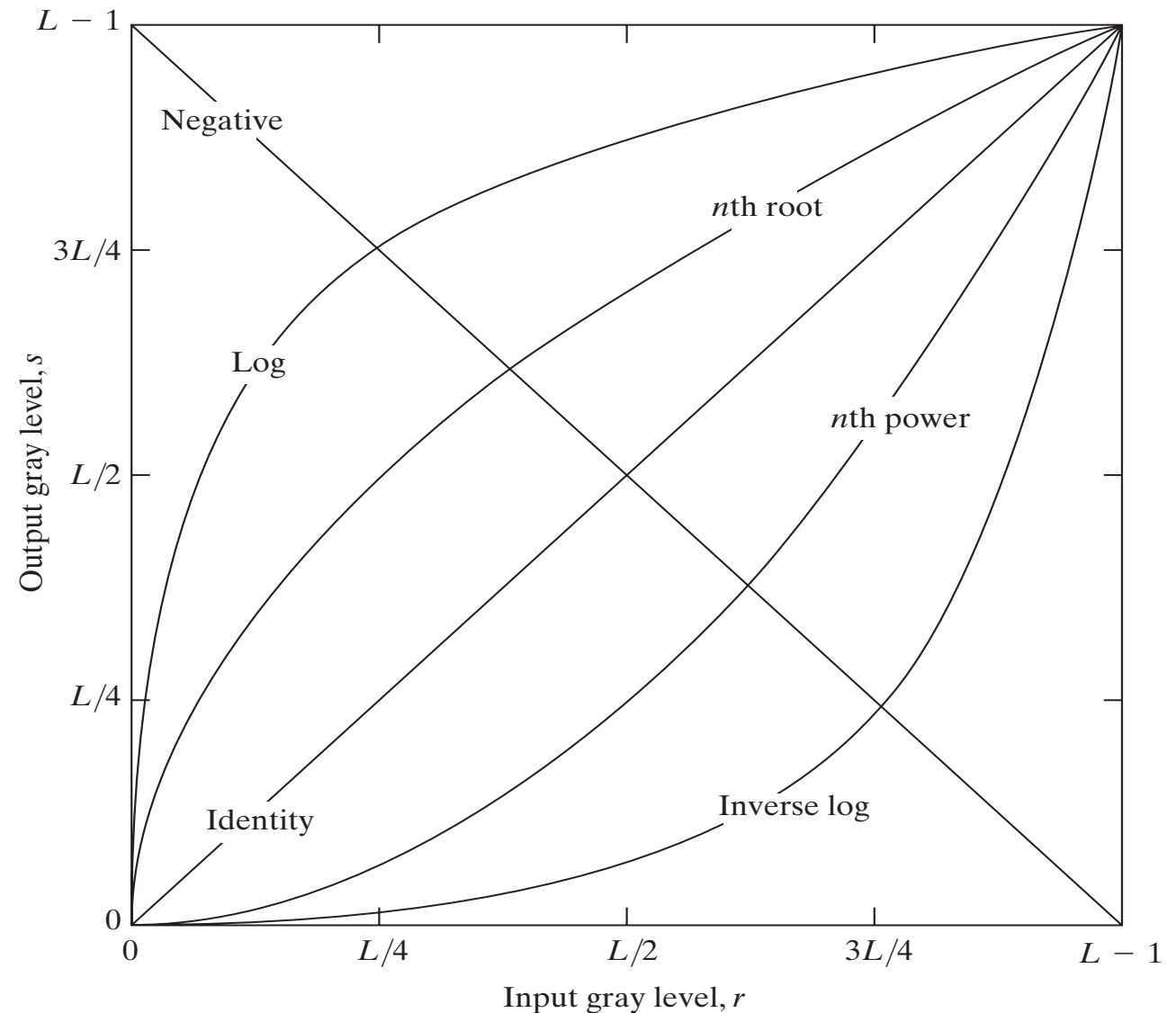
This transformation produces an image of **higher contrast** than the original by darkening the levels below a threshold value m and brightening the levels above the threshold in the original image. This technique is known as *contrast stretching*.



Basic Transformations

There are 3 basic types of functions used frequently for image enhancement:

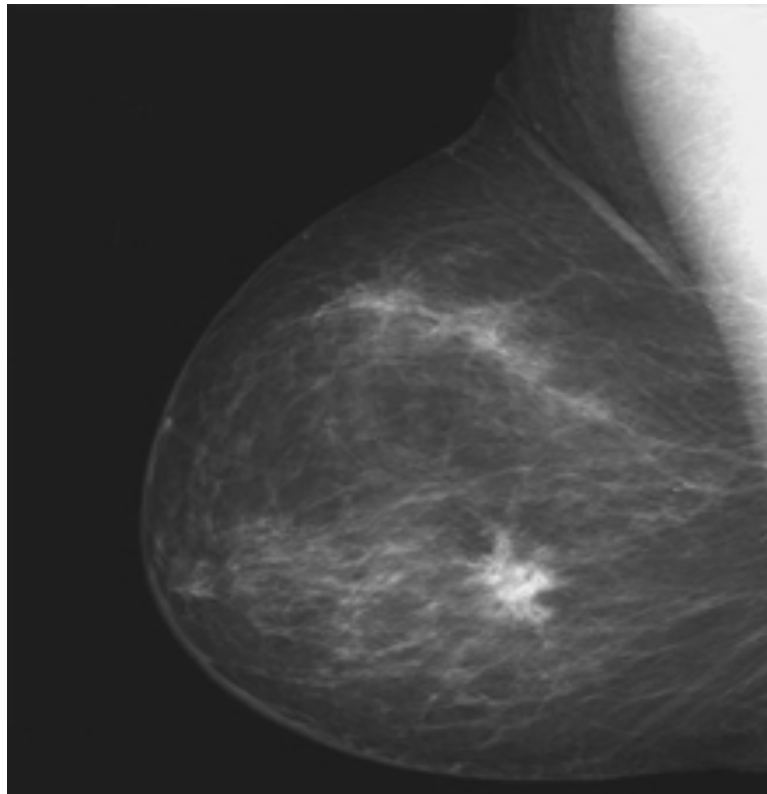
- ✓ linear (negative and identity transformations),
- ✓ logarithmic (log and inverse-log transformations),
- ✓ power-law (n th power and n th root transformations).



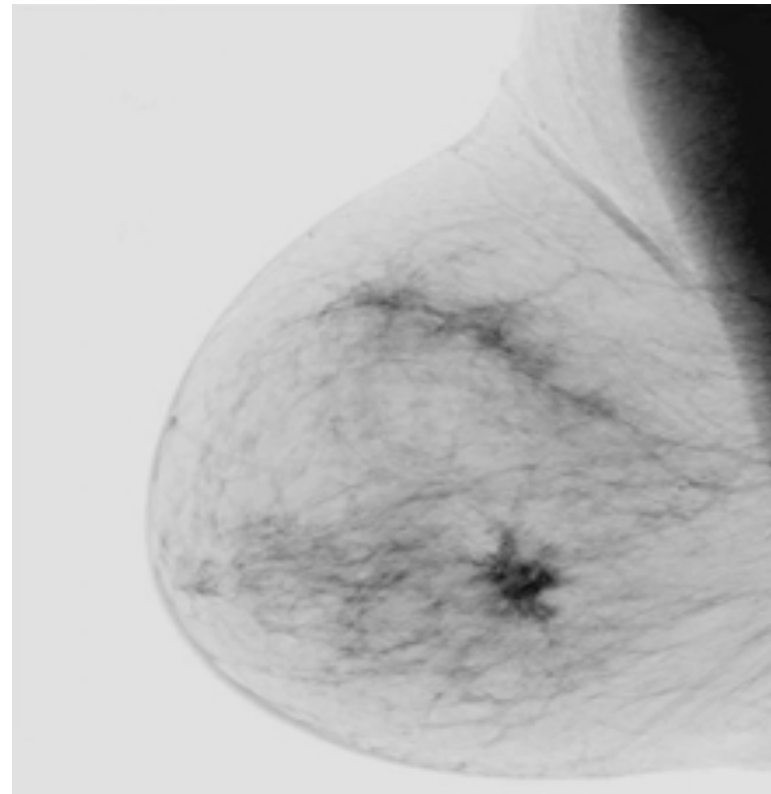
Linear Transformation

The negative of an image with gray levels in the range $[0, L - 1]$ is obtained by using the negative transformation, which is given by the expression

$$s = L - 1 - r.$$



(a) Original Image Mammography



(b) Negative Image

Logarithmic Transformation

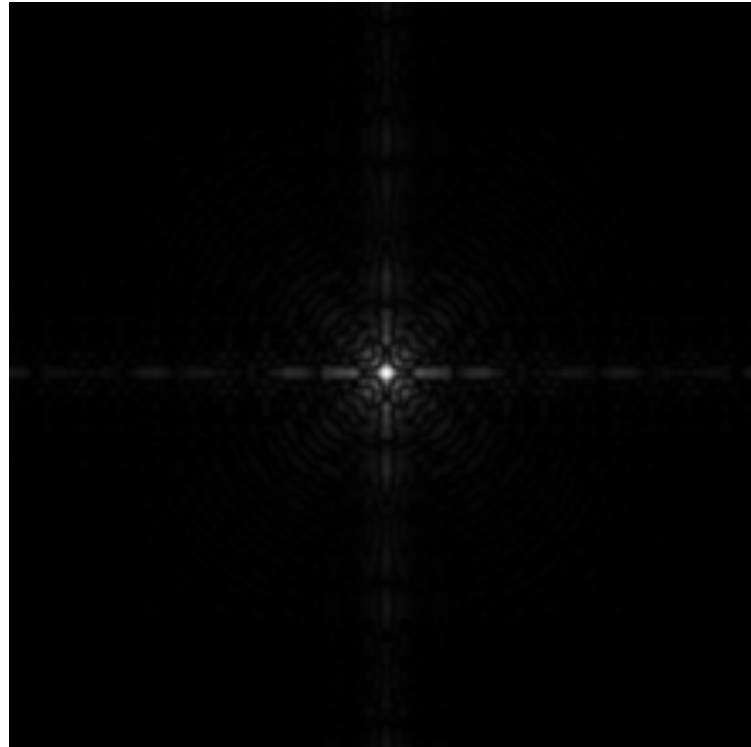
The general form of the log transformation is

$$s = c \log_b(1 + r),$$

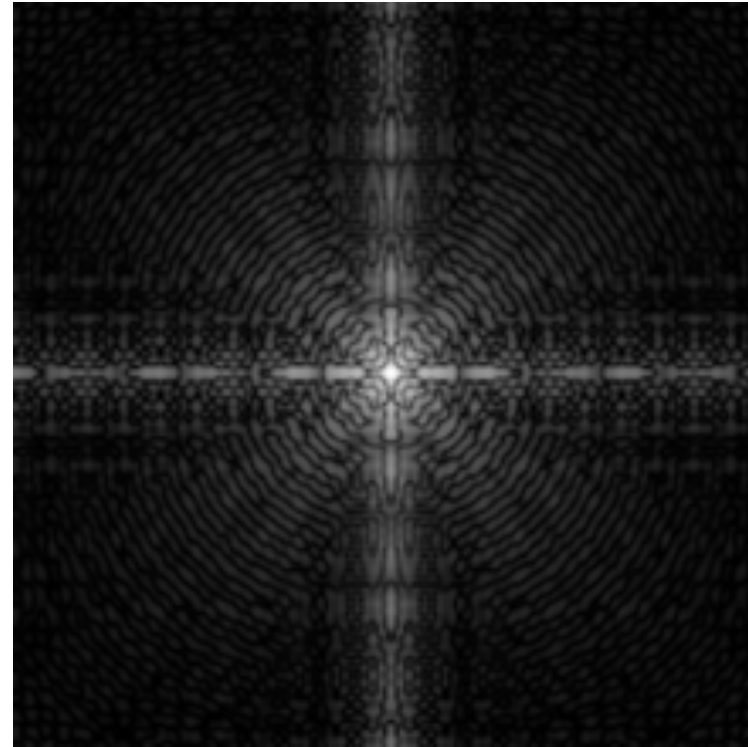
where c is a constant, b is the base of the logarithm and it is assumed that $r \geq 0$.

Reminder

$$\log_b(M) = N \iff b^N = M$$



(a) Fourier Spectrum
($r \in [0, 1.5 \times 10^6]$)



(b) Log Transform of Fourier Spectrum
($s \in [0, 6.2]$ for $c = 1$)

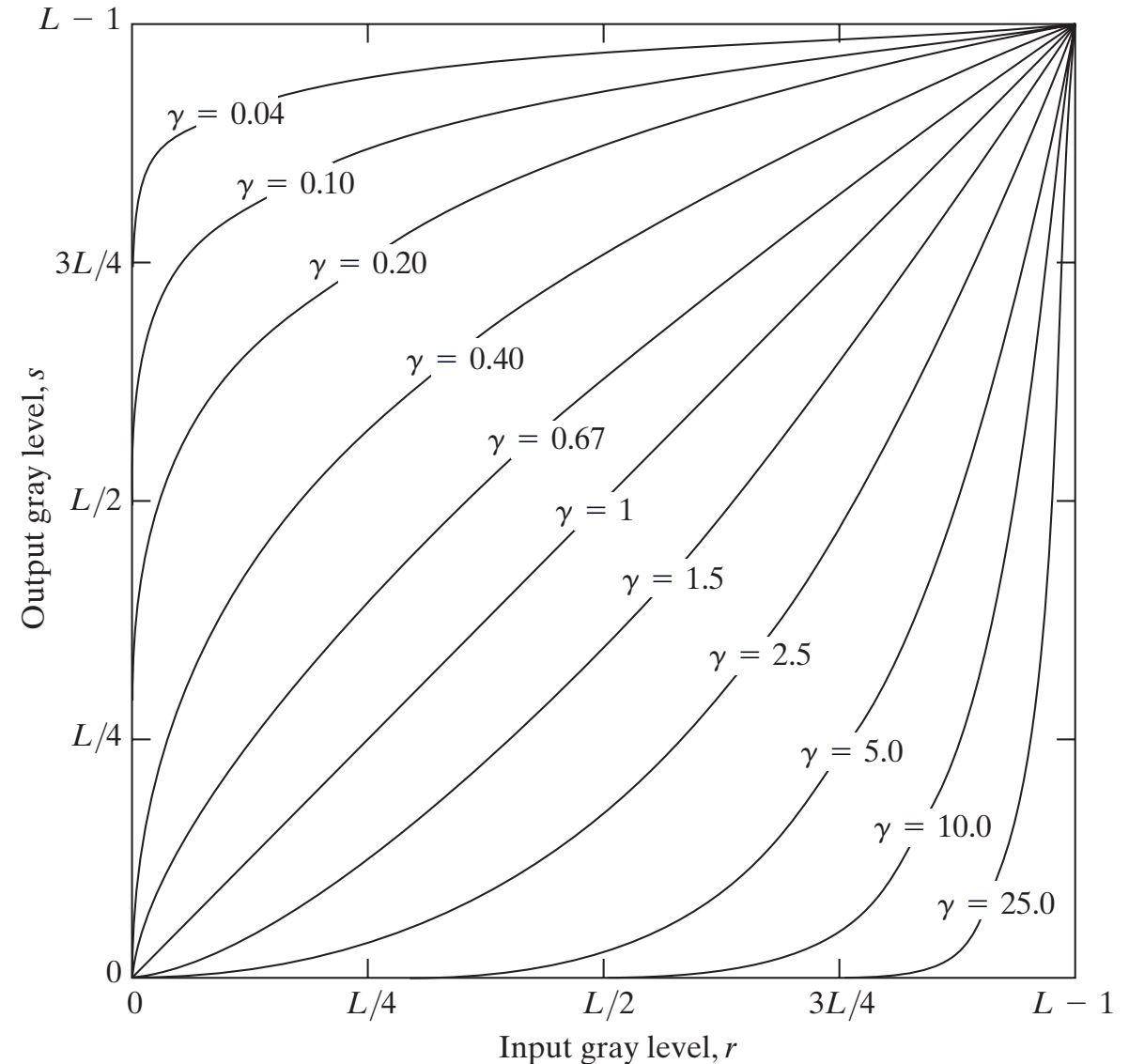
Power-law Transformation

- ✓ The basic form of the power-law transformation is given as follows

$$s = c(r + \varepsilon)^\gamma,$$

where γ and c are positive constants and ε is a small quantity that satisfies that the output exists even if the input is zero.

- ✓ This kind of transforms are also known as *Gamma Transforms* or *Gamma Correction*.



Power-law Transformation $\gamma < 1$



(a) Original MR Image of Human Spine



(b) Gamma Correction $\gamma = 0.6$



(c) Gamma Correction $\gamma = 0.4$



(d) Gamma Correction $\gamma = 0.3$

Power-law Transformation $\gamma > 1$



(a) Original Aerial images.



(b) Gamma Correction
 $\gamma = 3.0$



(c) Gamma Correction
 $\gamma = 4.0$



(d) Gamma Correction
 $\gamma = 5.0$

Piecewise-Linear Transformation

ADVANTAGES

- ✓ The form of piecewise functions can be arbitrarily complex.
- ✓ Important functions can be formulated only as piecewise function.

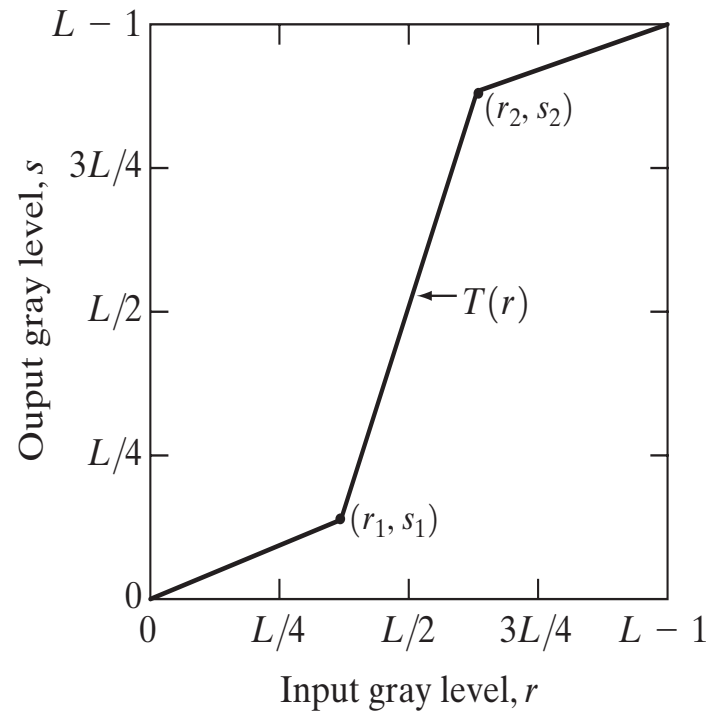
DISADVANTAGES

- ✓ They require considerably more user input.

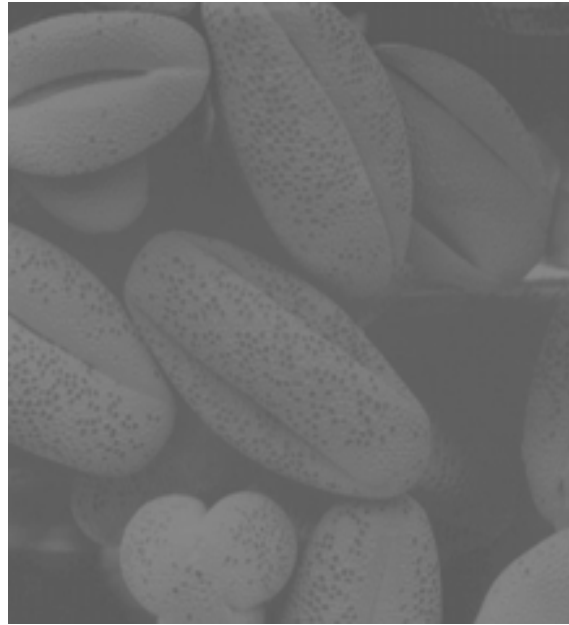
TYPES

1. Contrast Stretching
2. Gray-level Slicing
3. Bit-plane Slicing

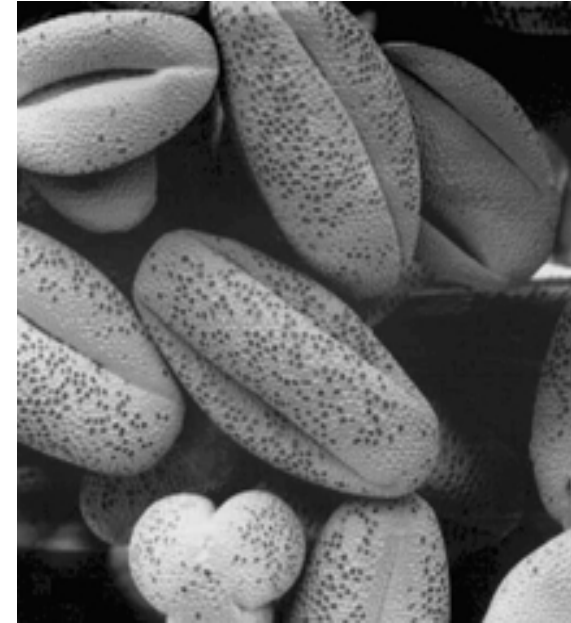
Contrast Stretching Transformation



(a) Contrast Stretching Function.



(a) Low contrast.

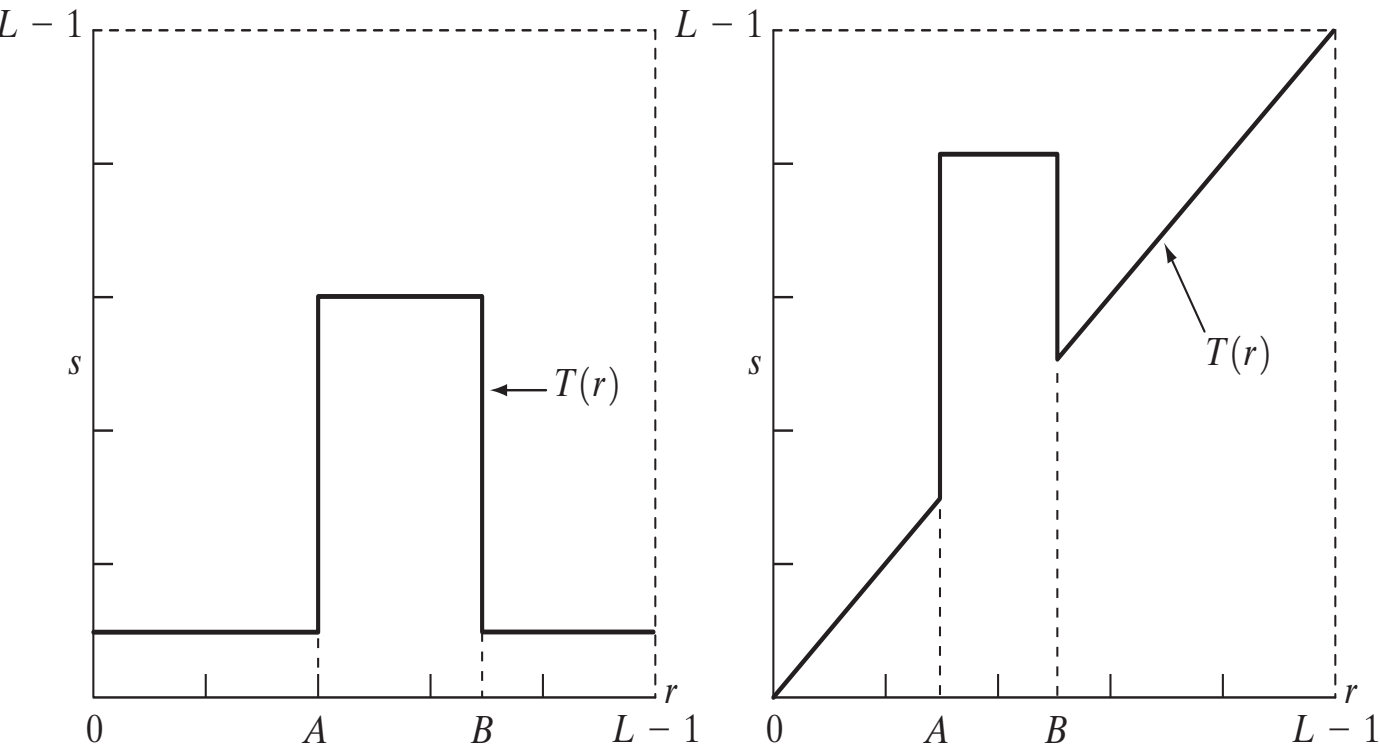


(b) Contrast Stretching.



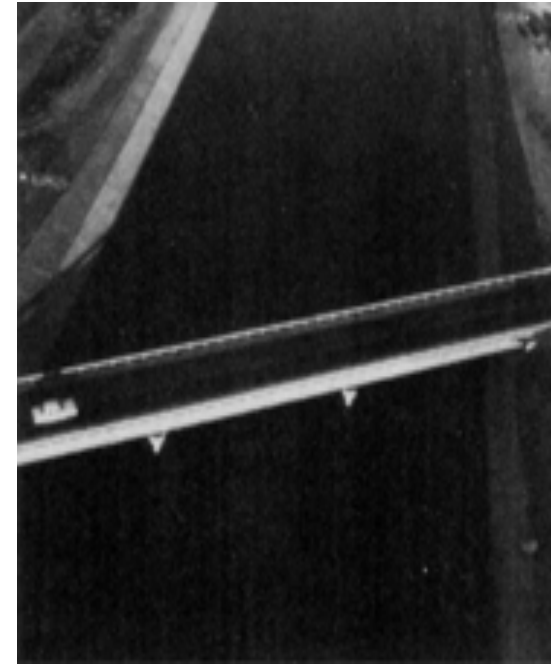
(c) Thresholding.

Gray-level Slicing Transformation

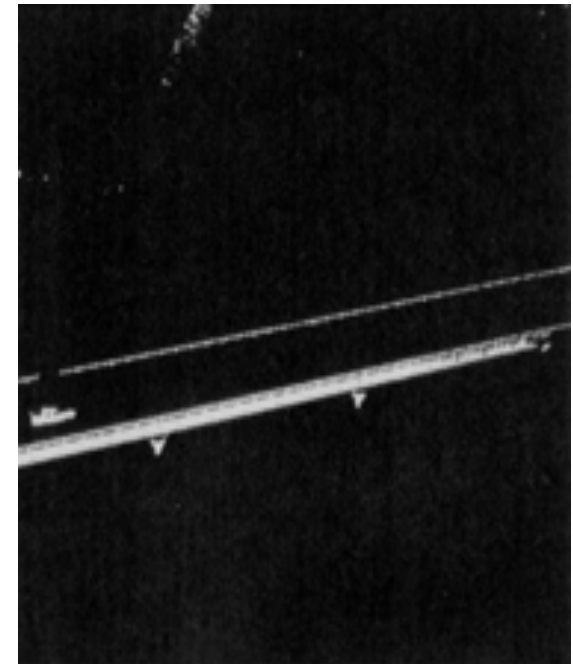


(a) Highlight values in $[A, B]$ and reduce all others.

(b) Highlight values in $[A, B]$ and preserve all others.



(c) Input image.



(d) Results of using the Transform (a).

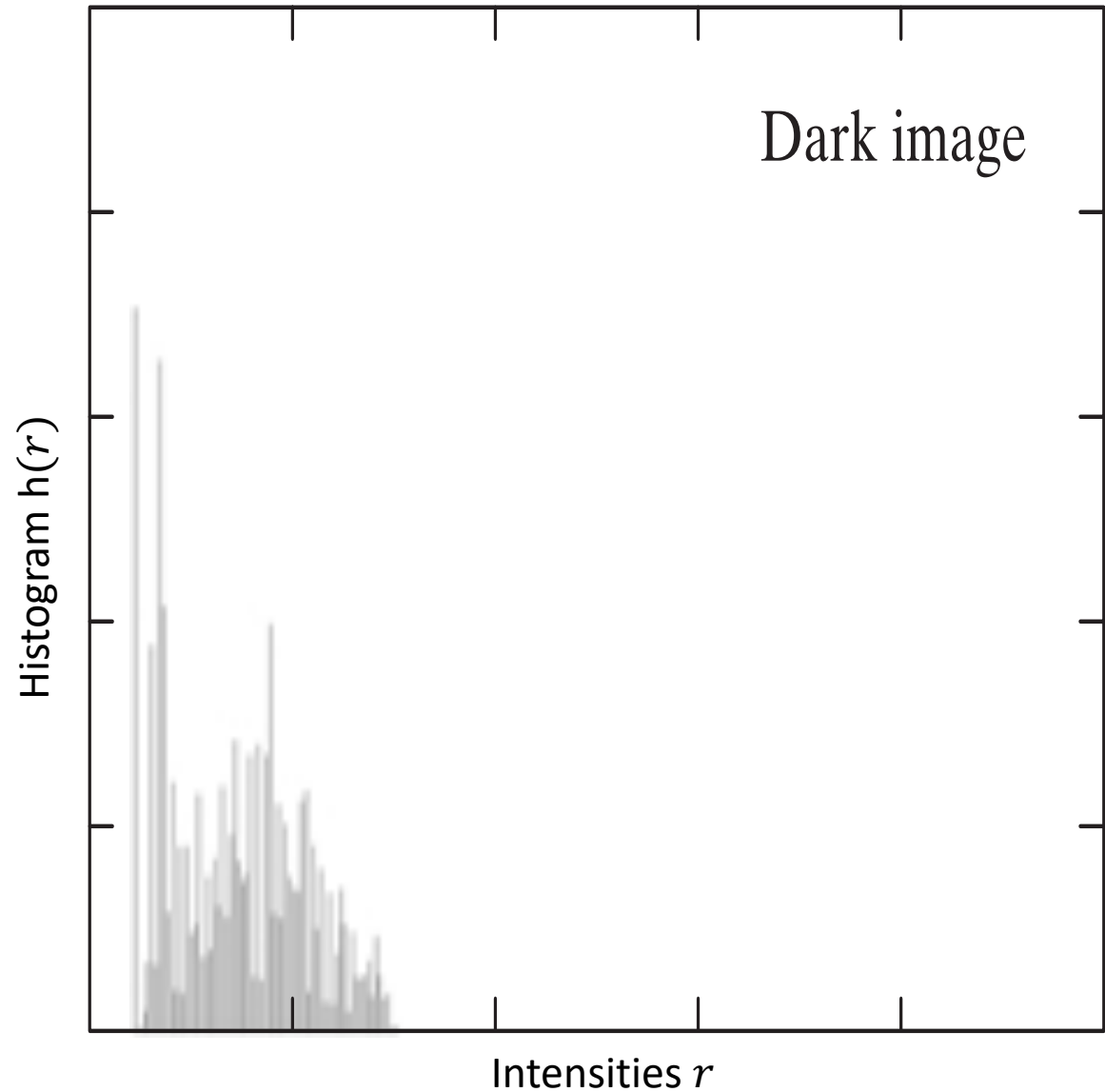
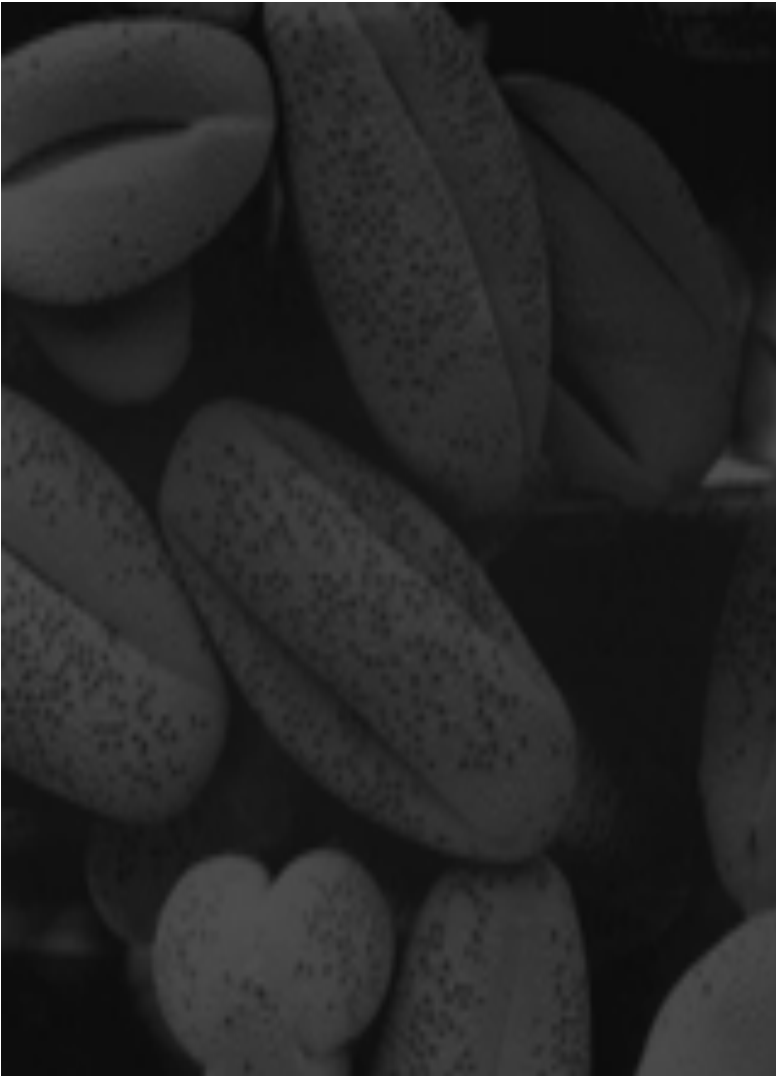
Histogram

- ✓ The **histogram** of an image with gray levels in range $[0, L - 1]$ is a discrete function $h(r_k) = n_k$, where r_k is the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k .
- ✓ Histogram **normalization** -> dividing each of its values by the total number of pixels in the image, denoted as N

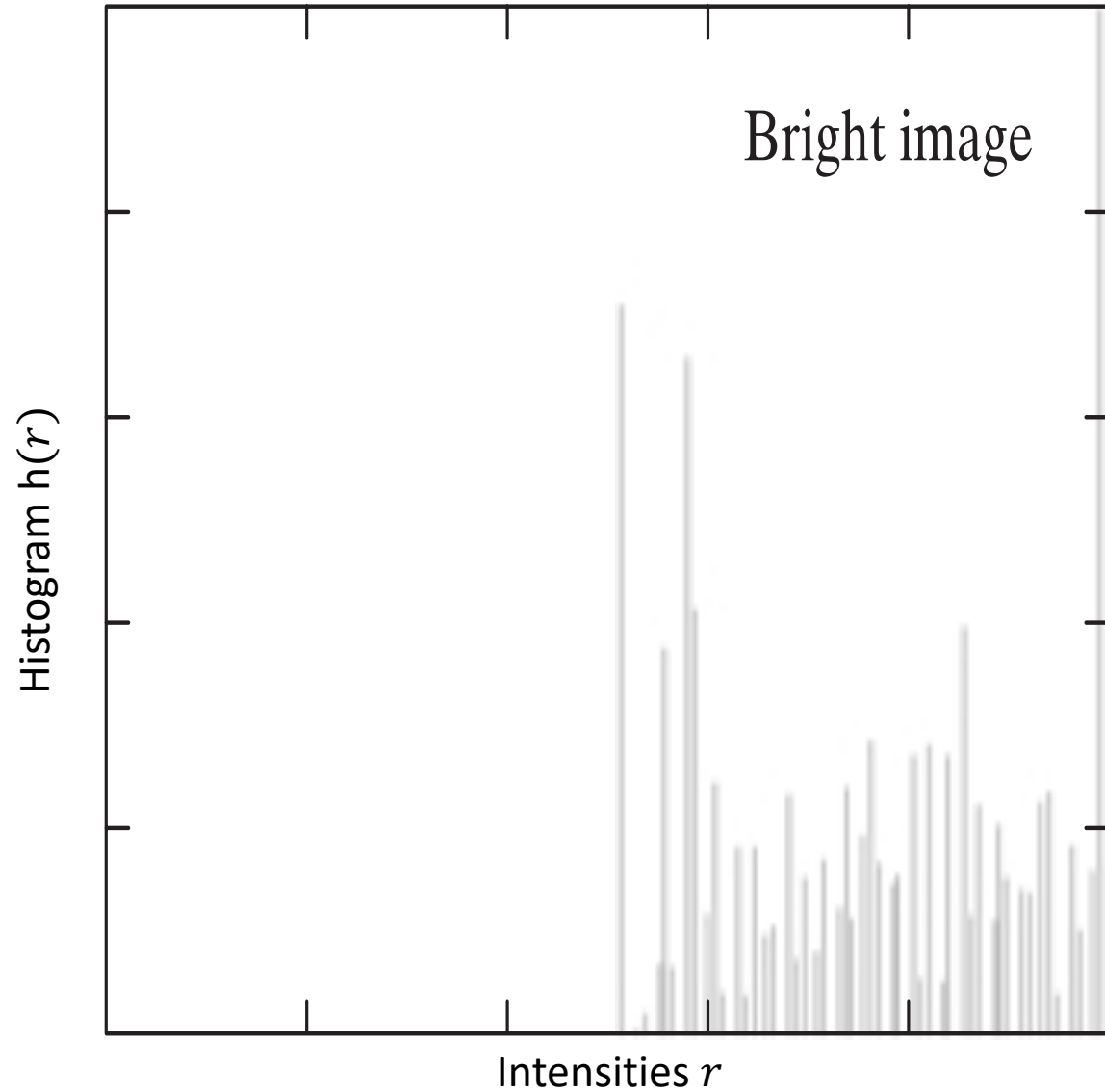
$$p(r_k) = \frac{n_k}{N}, \text{ for } k = 0, 1, \dots, L - 1.$$

- ✓ In other words, $p(r_k)$ gives the **probability** of occurrence of gray level r_k .
- ✓ The sum of all components of a normalized histogram is equal to 1.

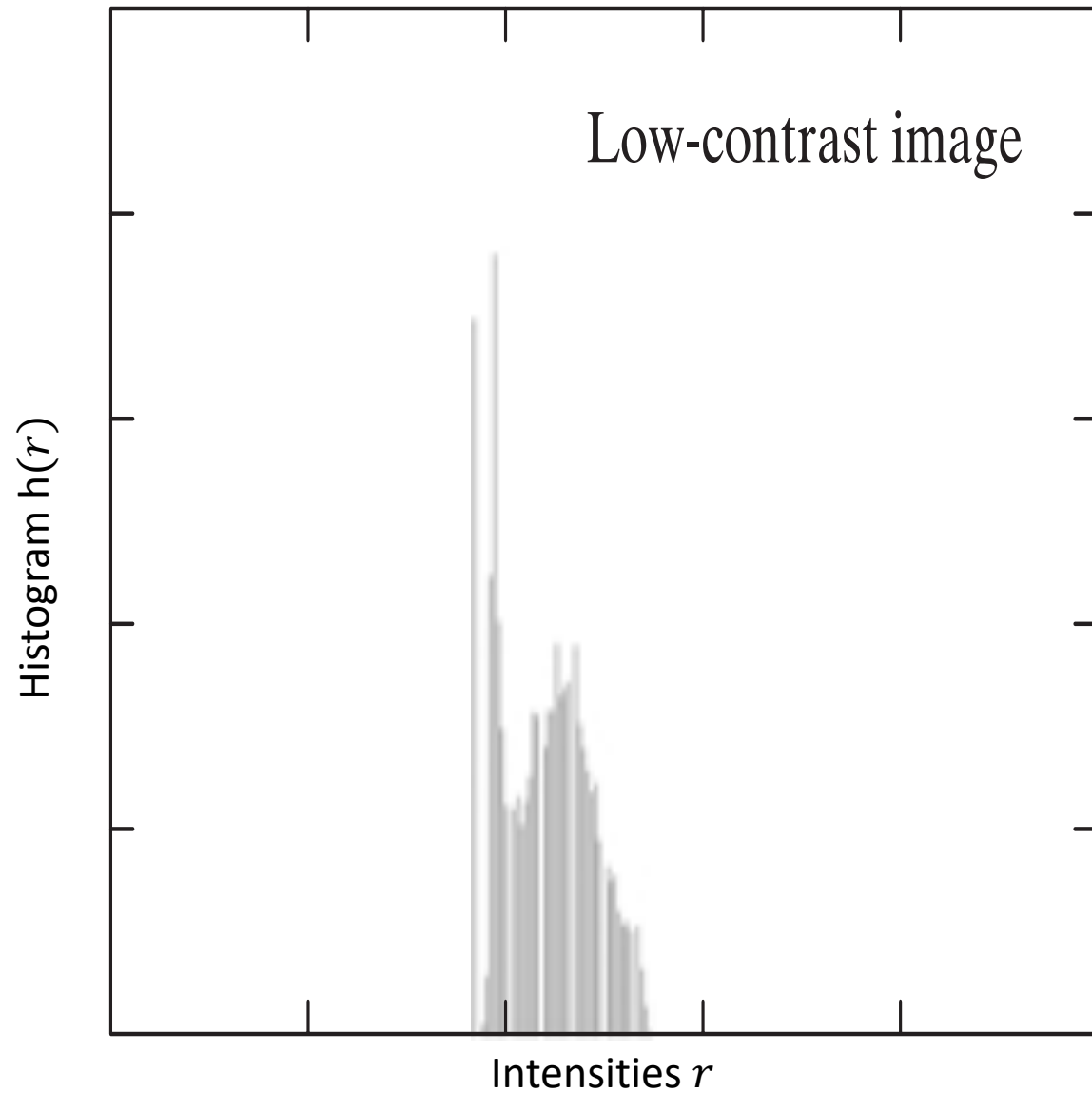
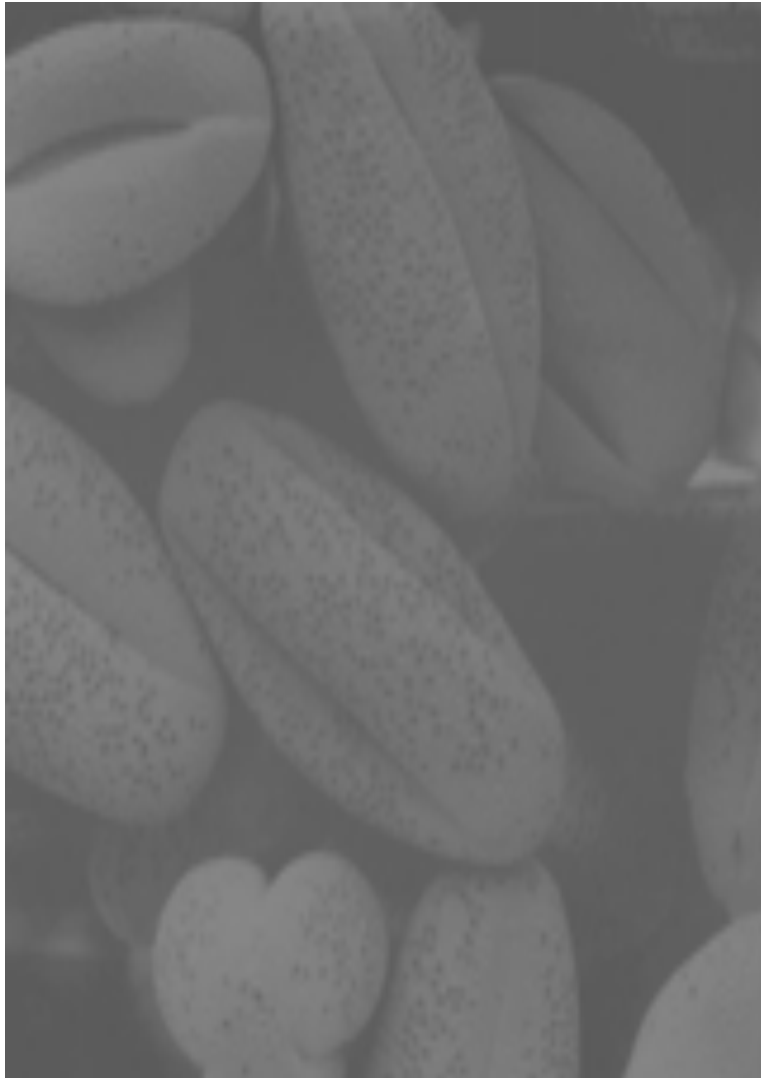
Examples



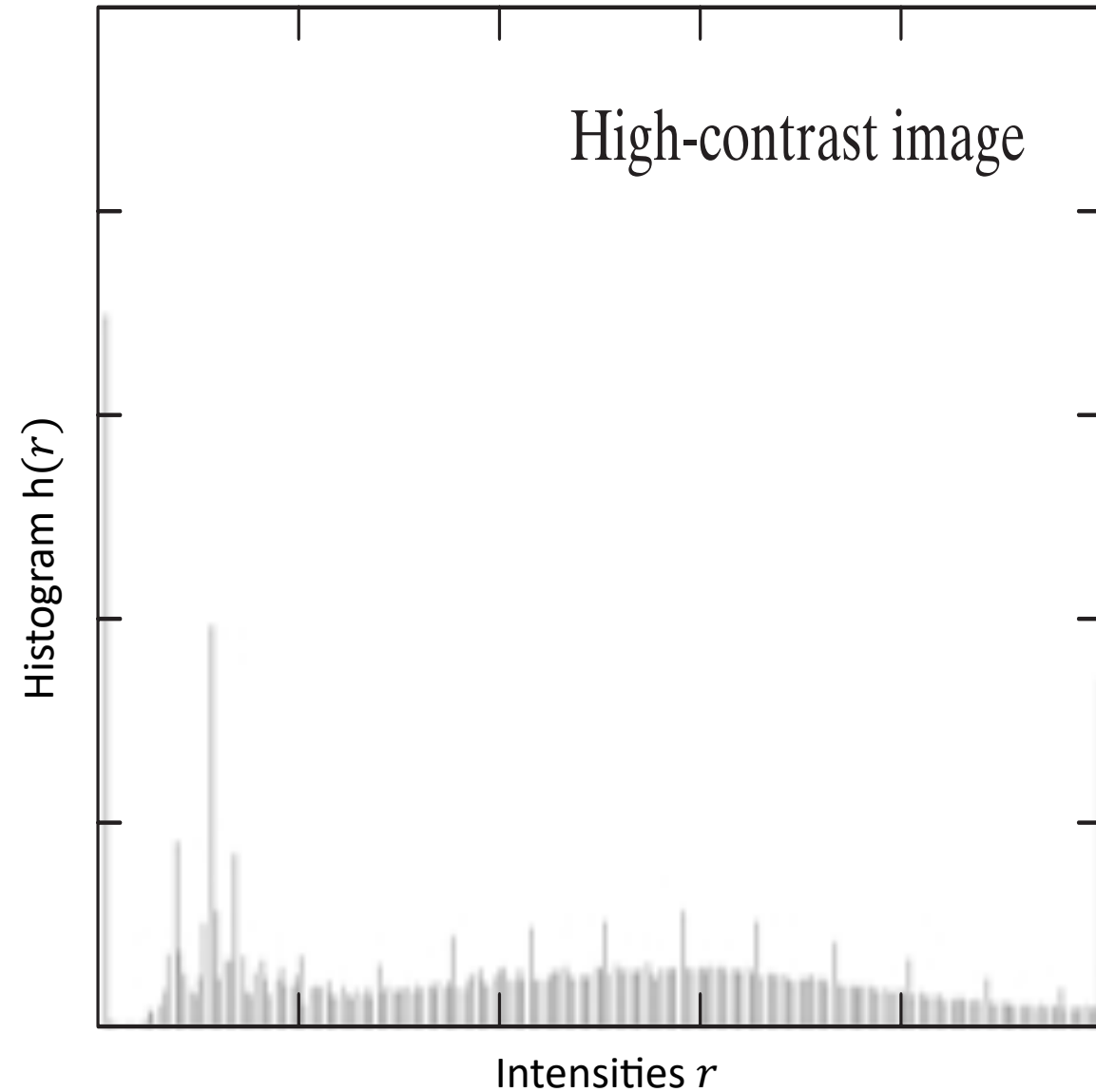
Examples



Examples



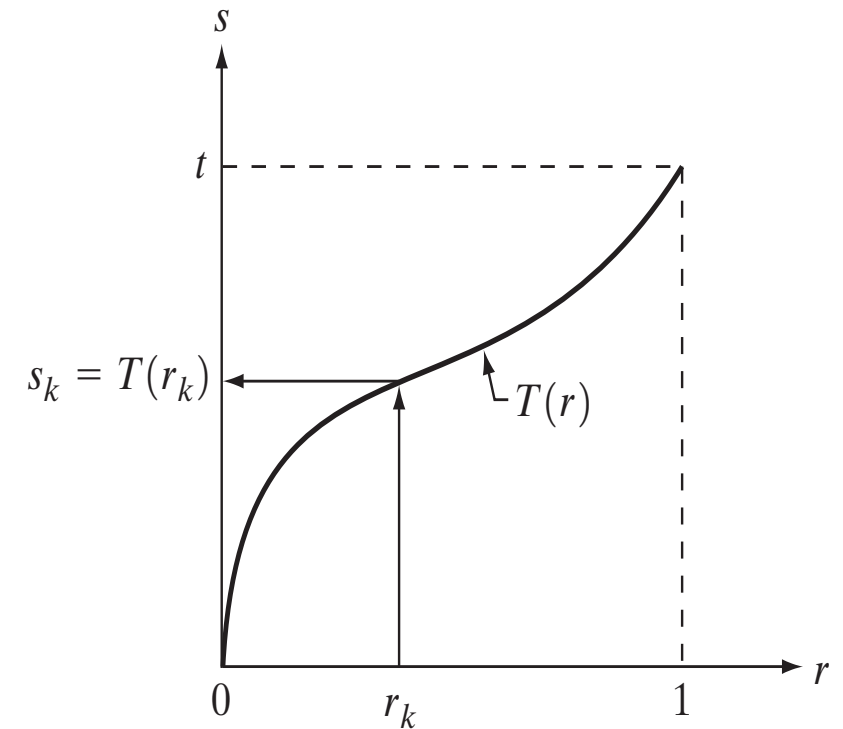
Examples



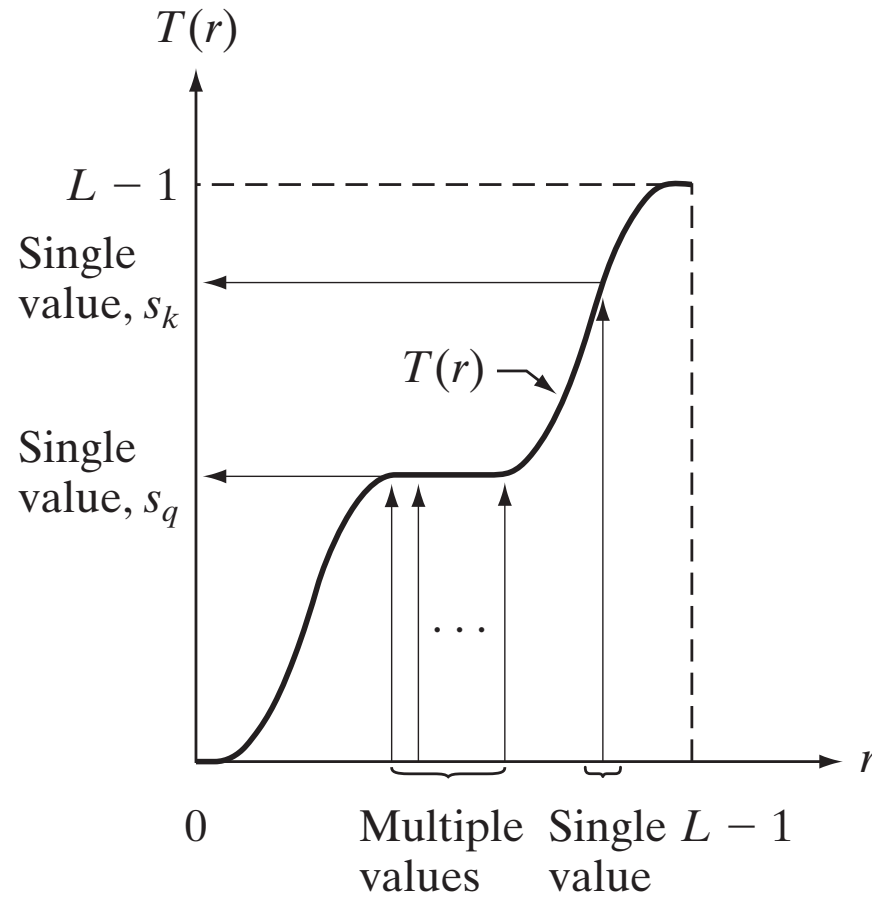
Histogram Equalization

- ✓ Every transformation is of the form $s = T(r)$ where $0 \leq r \leq 1$ when the values have been normalized
- ✓ We assume that the transformation function $T(r)$ satisfies the following conditions:
 - a) $T(r)$ is single-valued and monotonically increasing in the interval $0 \leq r \leq 1$ and
 - b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.
- ✓ The inverse transformation from s to r requires that $T(r)$ is a strictly monotonically increasing function and it is denoted

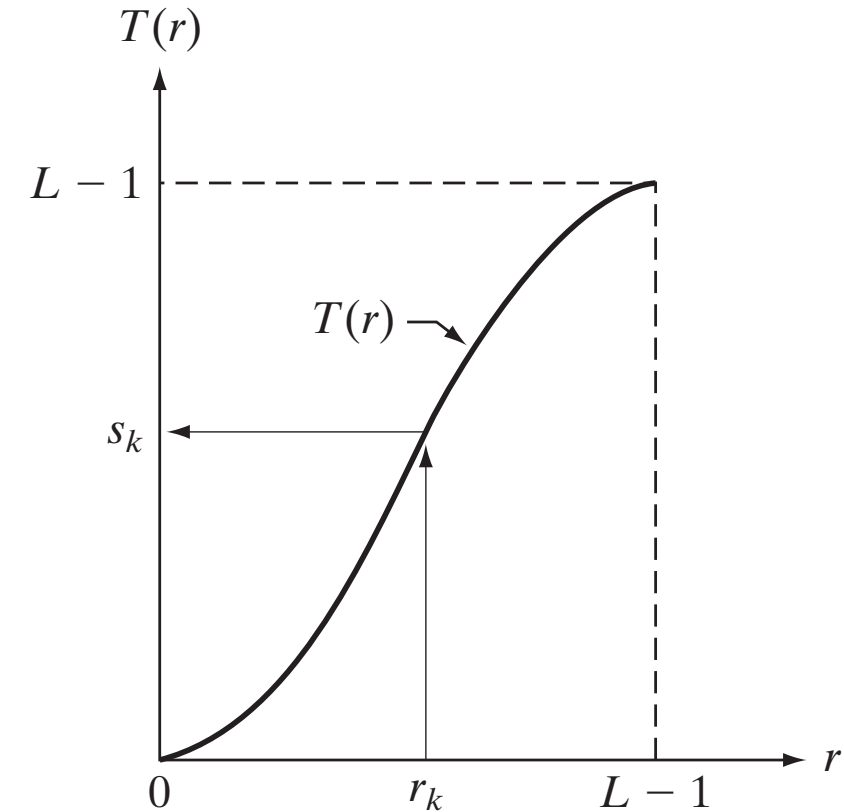
$$r = T^{-1}(s) \text{ for } 0 \leq s \leq 1.$$



Histogram Equalization

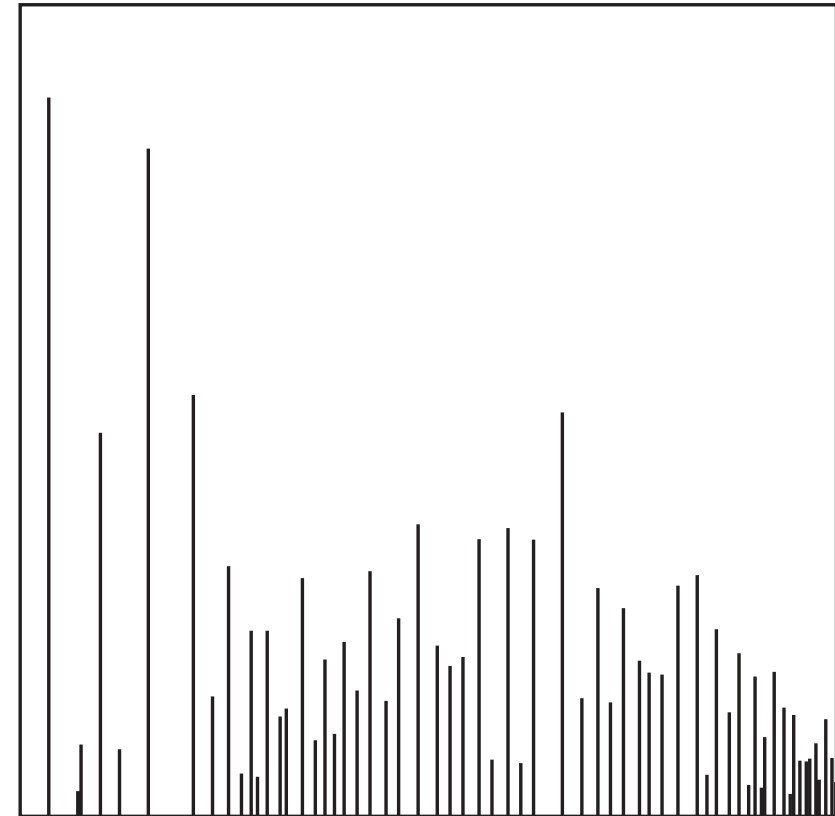
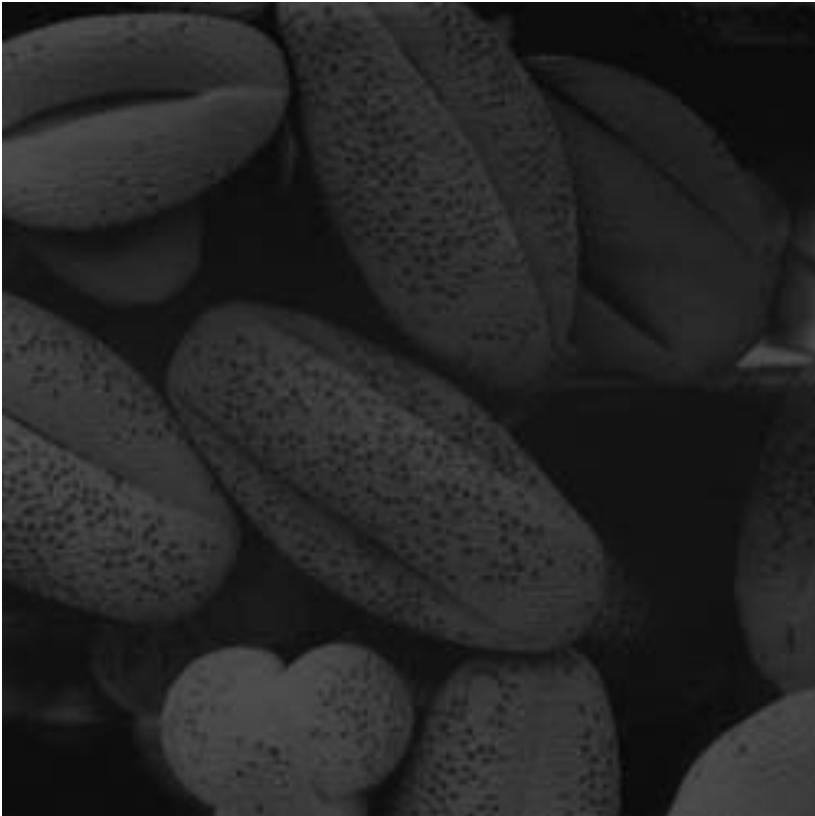


(a) Monotonically Increasing Function

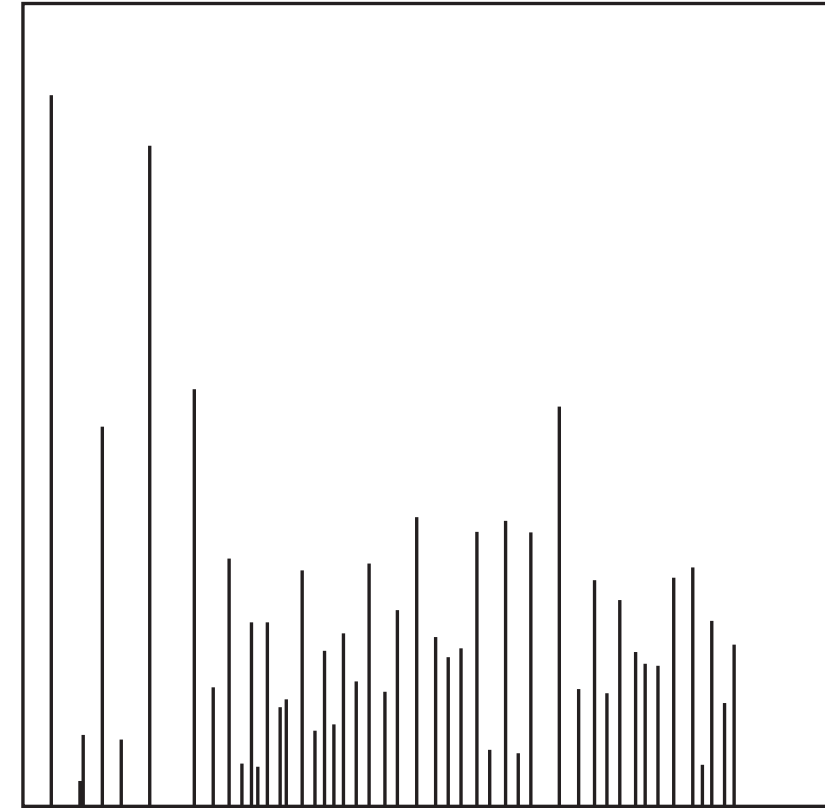


(b) Strictly Monotonically Increasing Function

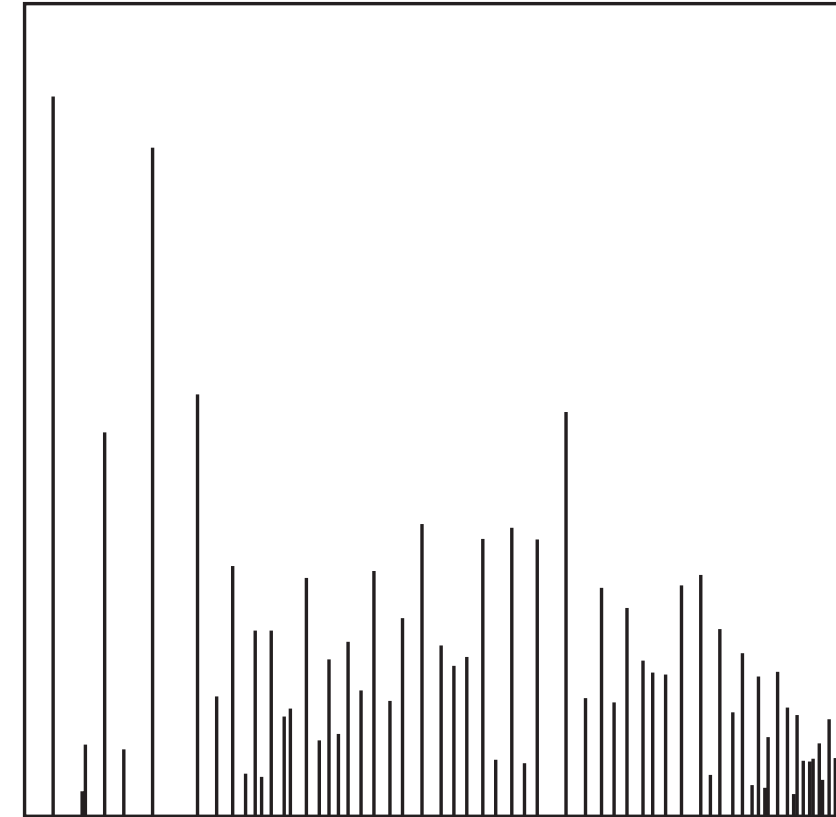
Histogram Equalization of a Dark Input



Histogram Equalization of a Bright Input



Histogram Equalization of a Low-Contrast Input



Histogram Equalization of a High-Contrast Input

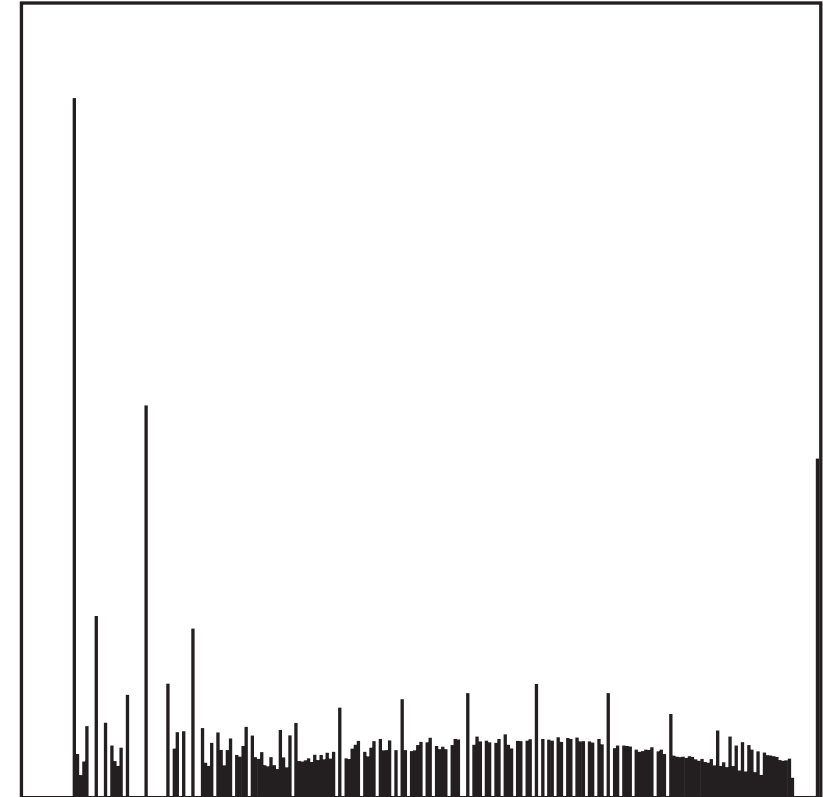
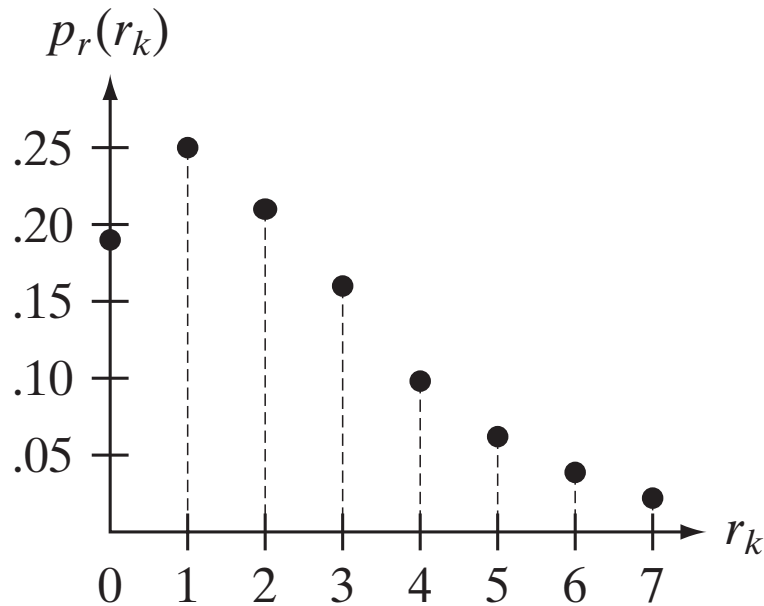


Illustration of a Histogram Equalization (1/2)

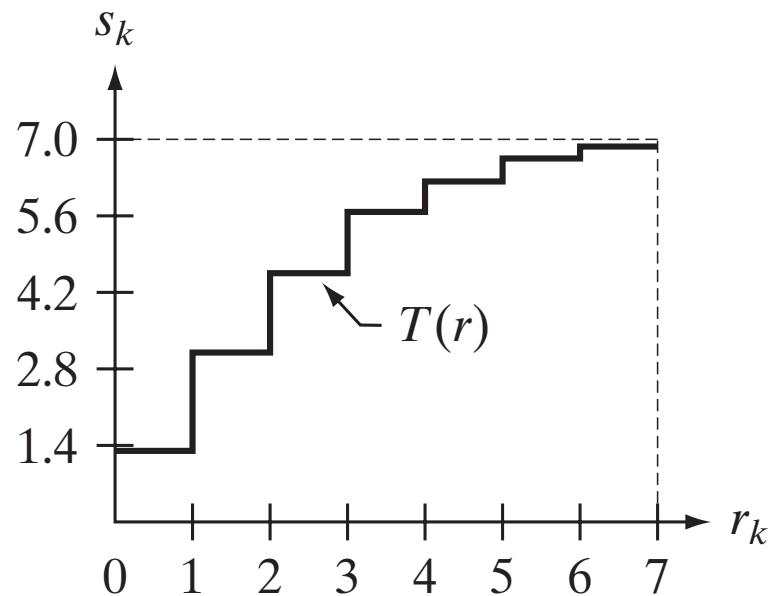
- ✓ Assume a 3-bit image ($L = 8$ levels) of size 64×64 pixels ($MN = 4096$).
- ✓ Assume that the intensity levels are integers in the range $[0, L - 1] = [0, 7]$.
- ✓ We are given the following data:

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

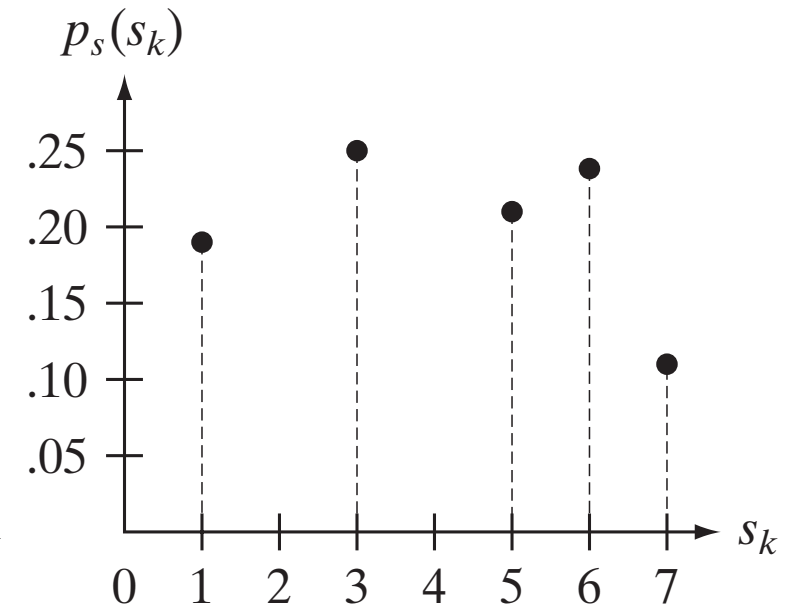
Illustration of a Histogram Equalization (2/2)



(a) Original Histogram of an 8-bit image .



(b) Transformation function.



(c) Equalized Histogram.

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j)$$

Enhancement Using Arithmetic Operations

- ✓ **Arithmetic** operations involving images are performed on a pixel-by-pixel basis between two or more images. E.g. subtraction of two images results in a new image whose pixel at coordinates (x, y) is the difference between the pixels in that same location in the two images being subtracted.
- ✓ The actual mechanics of implementing arithmetic operation can be done **sequentially**, one pixel at a time, **or in parallel**, where all operations are performed simultaneously.
- ✓ The four arithmetic operations are: **Addition**, **Subtraction**, **Multiplication** and **Division**.

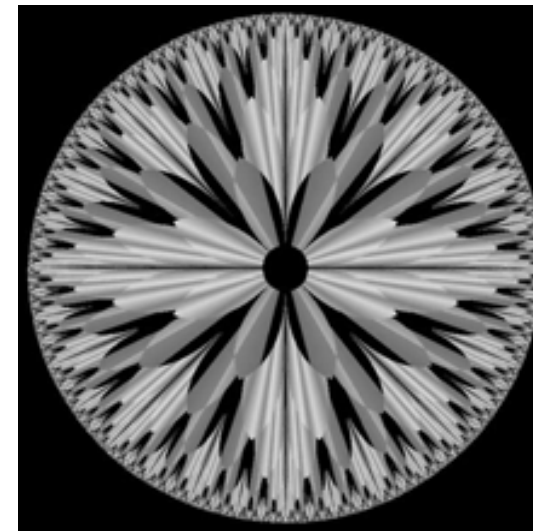
Image Subtraction

- ✓ The difference between two images $f(x, y)$ and $h(x, y)$, expressed as

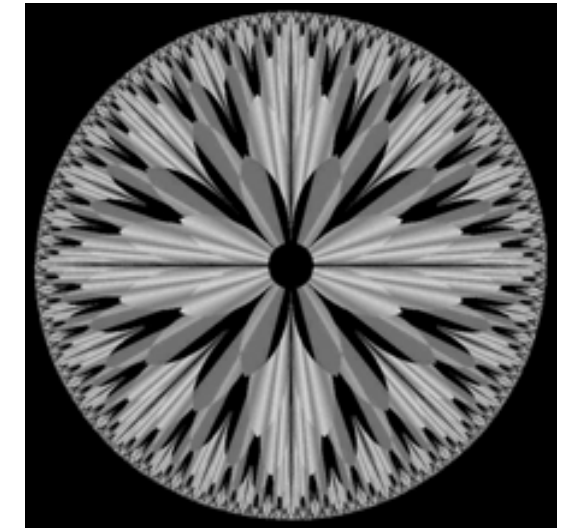
$$g(x, y) = f(x, y) - h(x, y)$$

is obtained by computing the difference between all pairs of corresponding pixels from f and h .

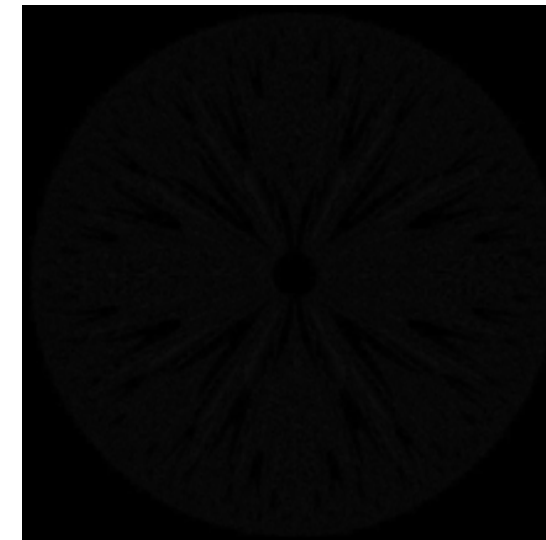
- ✓ The key usefulness of subtraction is the enhancement of *differences* between images.



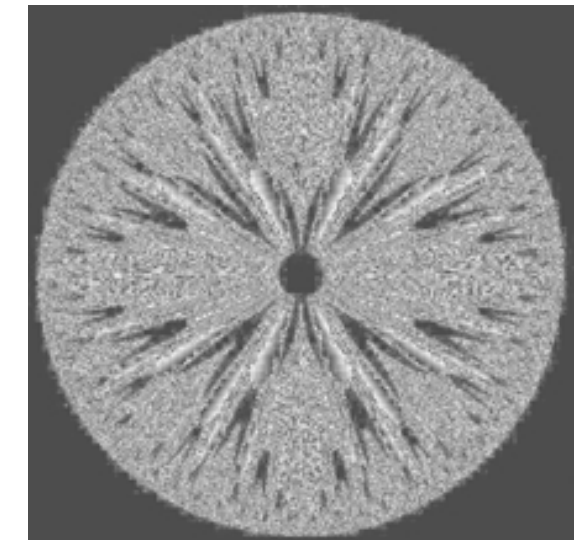
(a) Original Image



(b) Setting the Four lower-bit planes to zero



(c) Difference between (a) and (b)



(d) Histogram-equalized difference image

Image Averaging

- ✓ Consider a noisy image $g(x, y)$ formed by the addition of noise $h(x, y)$ to an original image $f(x, y)$; that is,

$$g(x, y) = f(x, y) + h(x, y)$$

where the assumption is that at every pair of coordinates (x, y) the **noise is un-correlated** and has zero average value.

- ✓ **Reminder:** The variance of a random variable x with mean m is defined as $E[(x - m)^2]$, where $E\{. \}$ is the expected value of the argument. The covariance of two random variables x_i and x_j is defined as $E[(x_i - m_i)(x_j - m_j)]$. If the variables are *uncorrelated*, their covariance is 0.

Image Averaging

- ✓ If the noise satisfies the constraints just stated, it can be shown that if an image $\bar{g}(x, y)$ is formed by averaging K different noisy images,

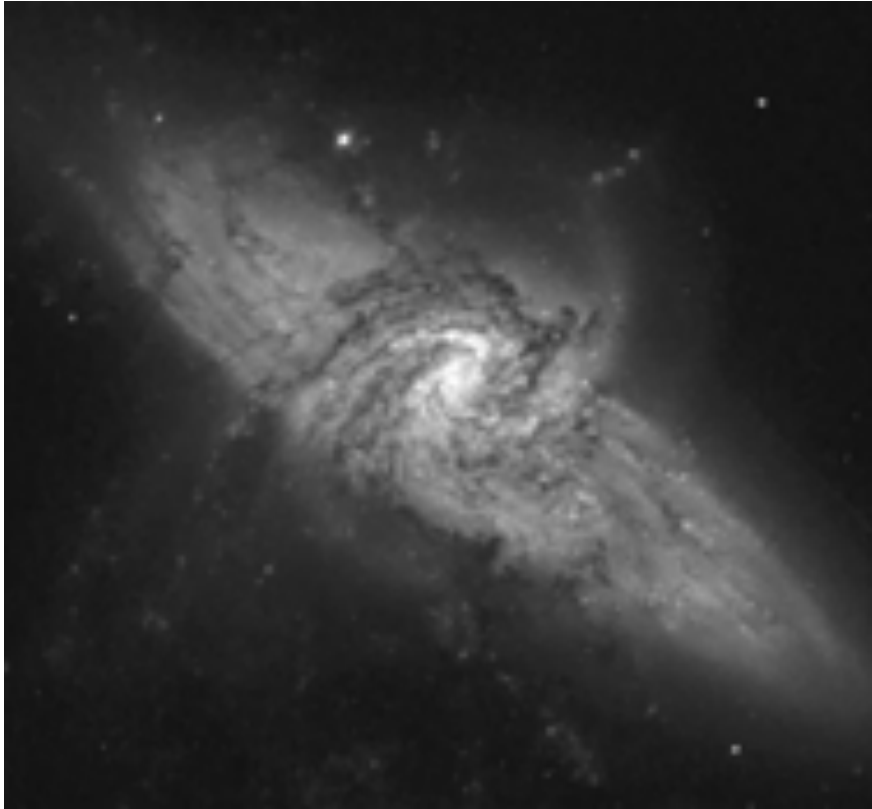
$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y).$$

- ✓ Then, it follows that

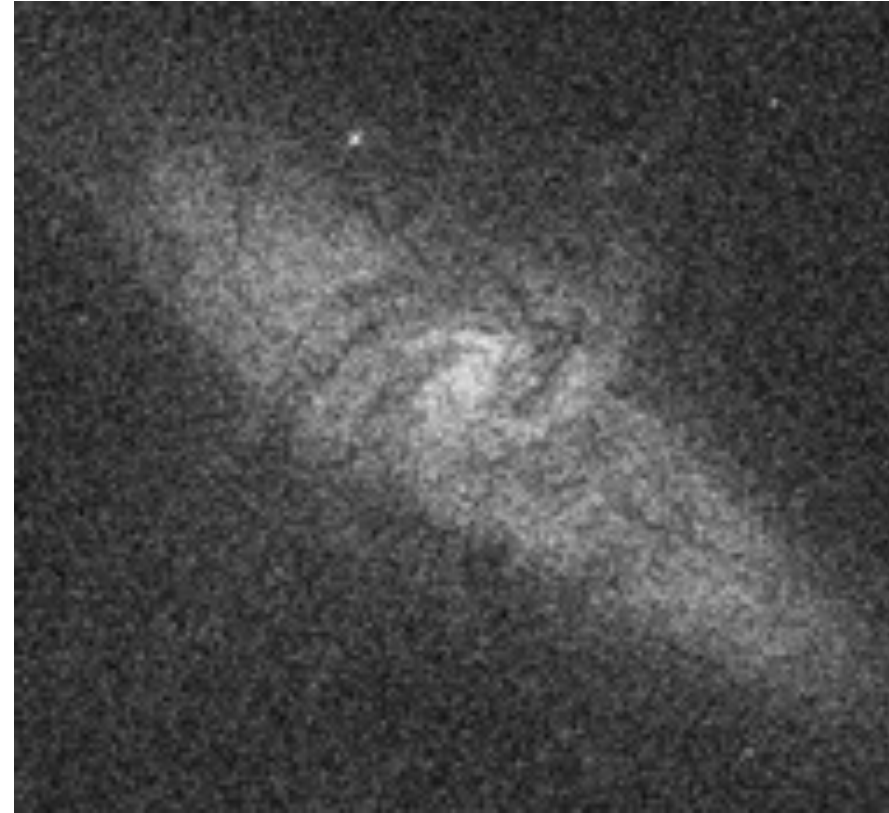
$$E\{\bar{g}(x, y)\} = f(x, y) \quad \text{and} \quad \sigma_{\bar{g}(x, y)}^2 = \frac{1}{\sqrt{K}} \sigma_{n(x, y)}^2,$$

where $E\{\bar{g}(x, y)\}$ is the expected value of \bar{g} , and $\sigma_{\bar{g}(x, y)}^2$ and $\sigma_{n(x, y)}^2$ are the variances of \bar{g} and n , all at coordinated (x, y) .

Applications

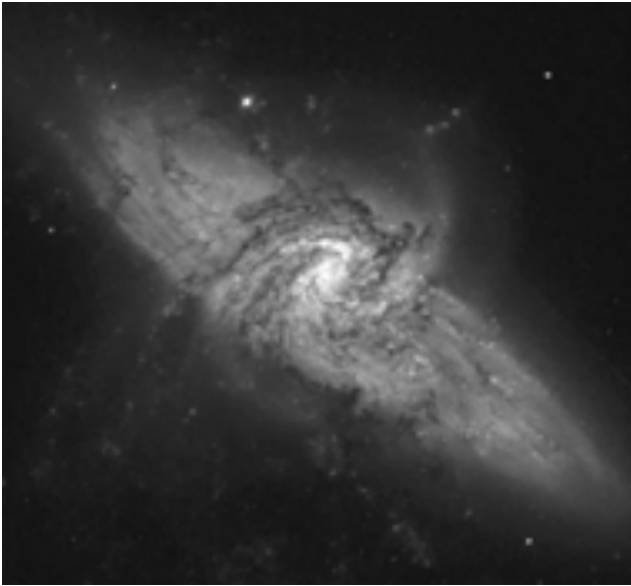


(a) Original Image

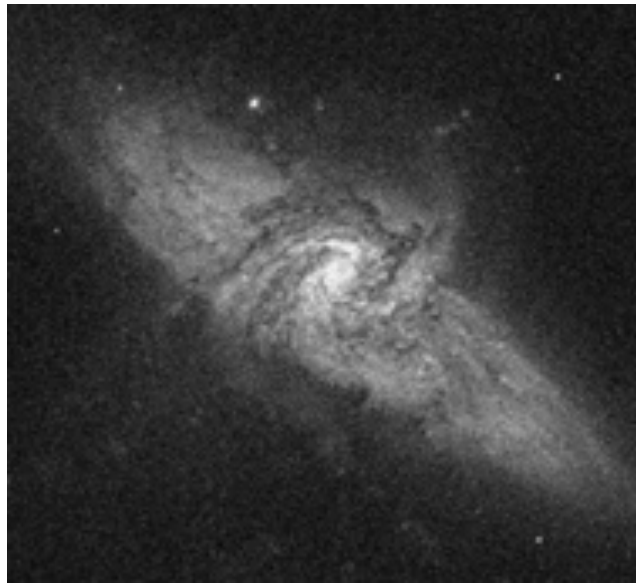


(b) Noisy Image

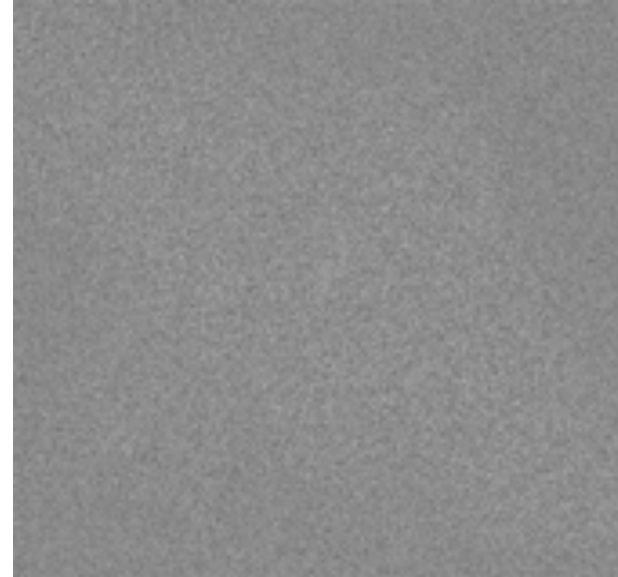
Applications



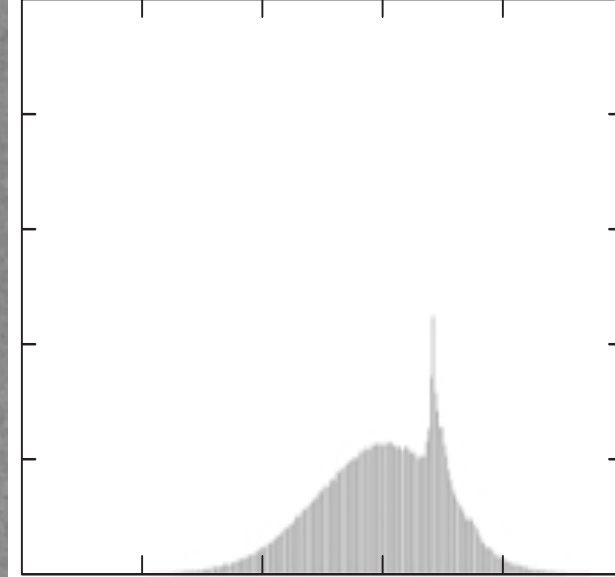
(a) Original Image



(b) Averaging $K = 8$ Images

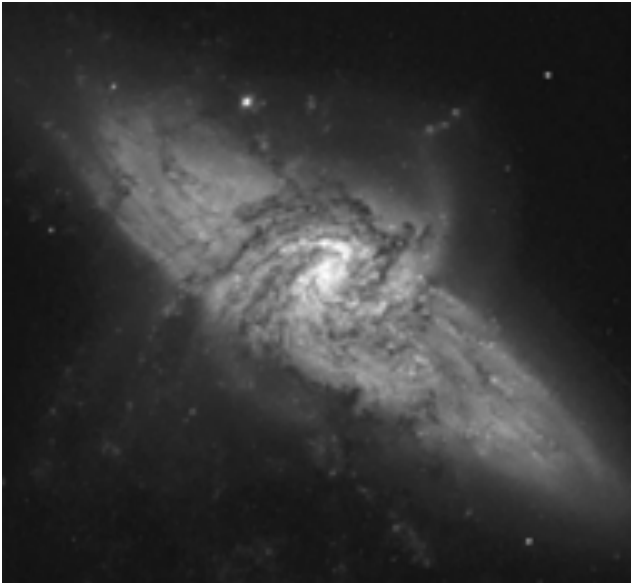


(c) Difference between
(a) and (b)

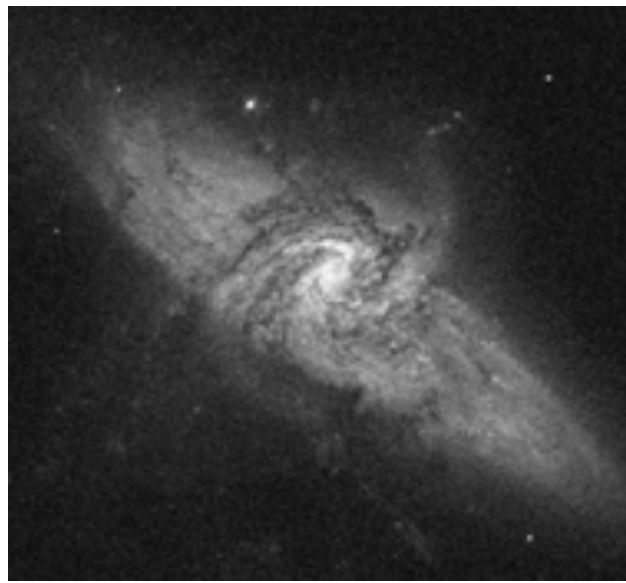


(d) Histogram of (c)

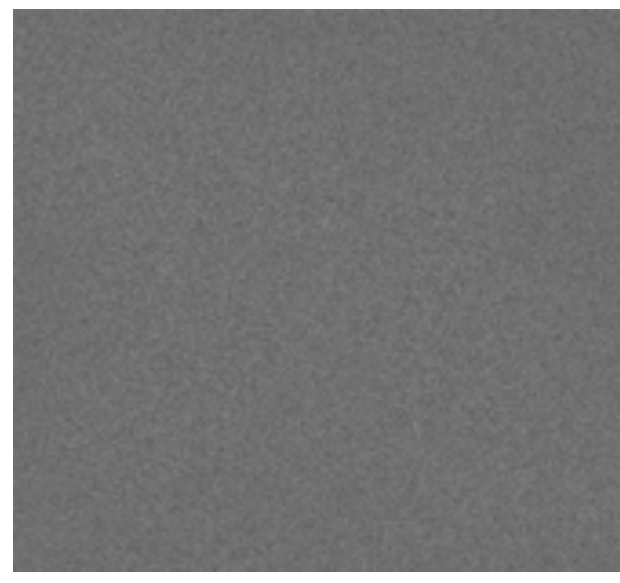
Applications



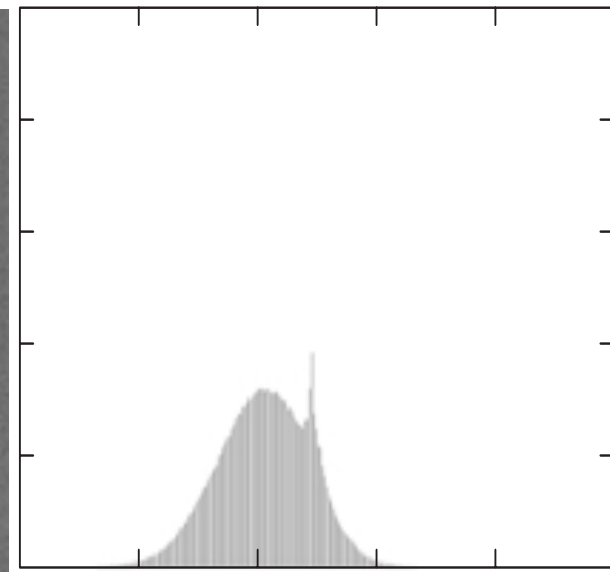
(a) Original Image



(b) Averaging $K = 16$ Images

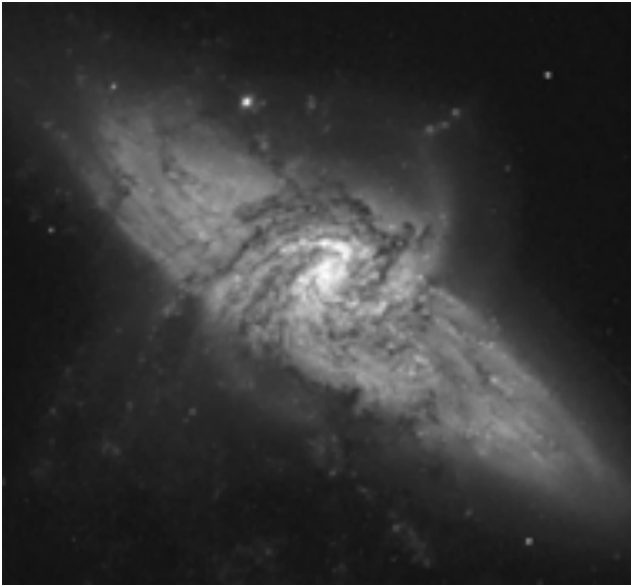


(c) Difference between
(a) and (b)

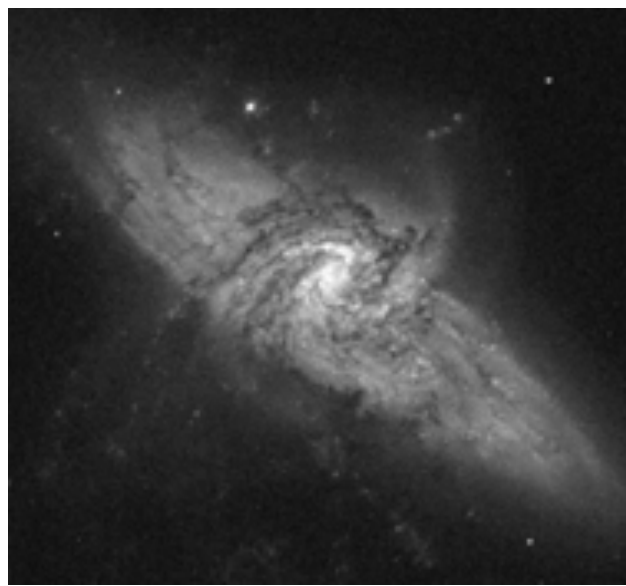


(d) Histogram of (c)

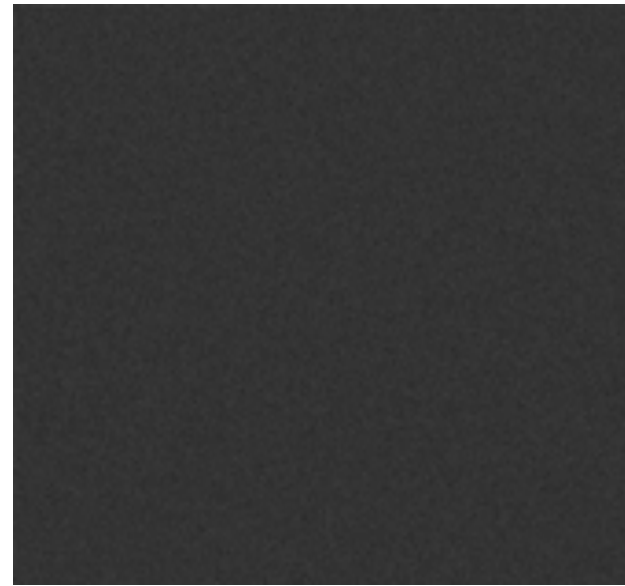
Applications



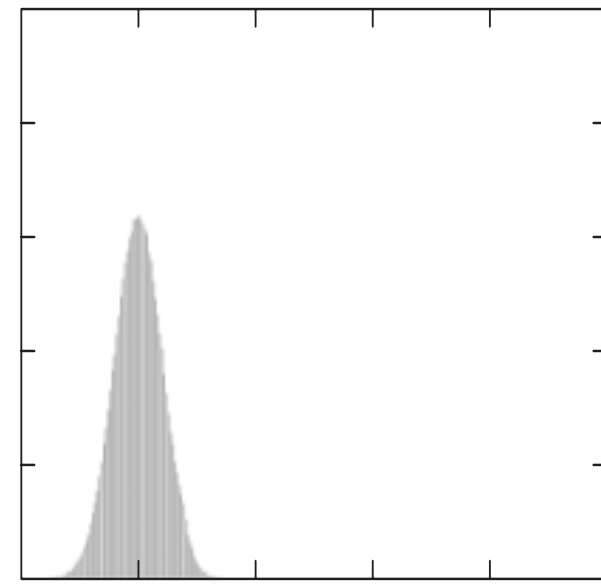
(a) Original Image



(b) Averaging $K = 64$ Images

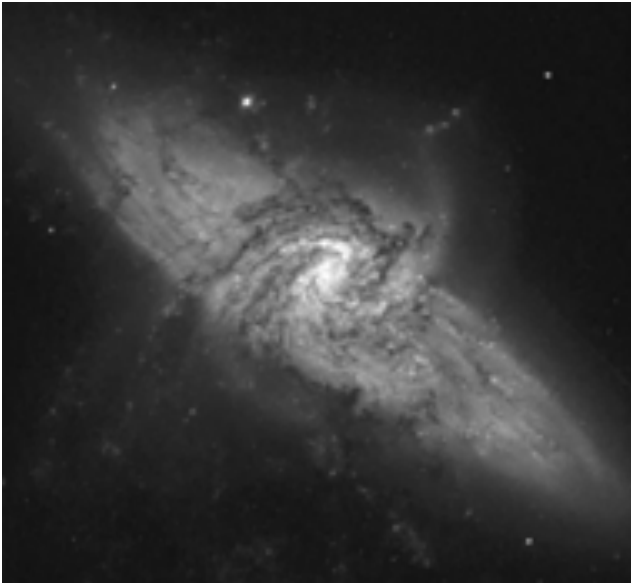


(c) Difference between
(a) and (b)

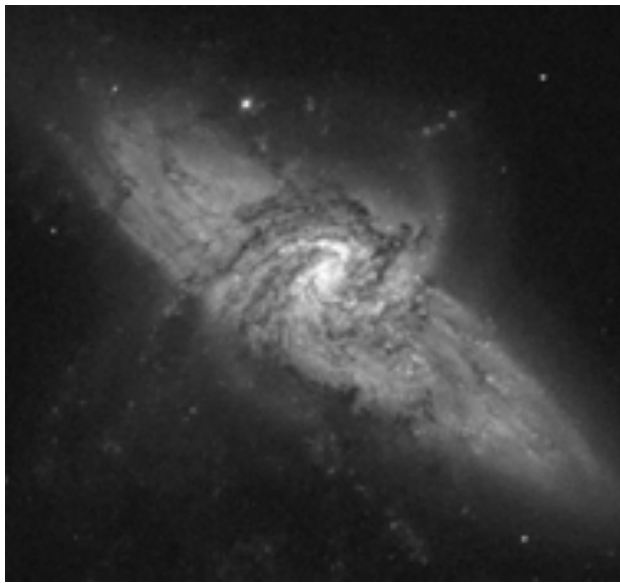


(d) Histogram of (c)

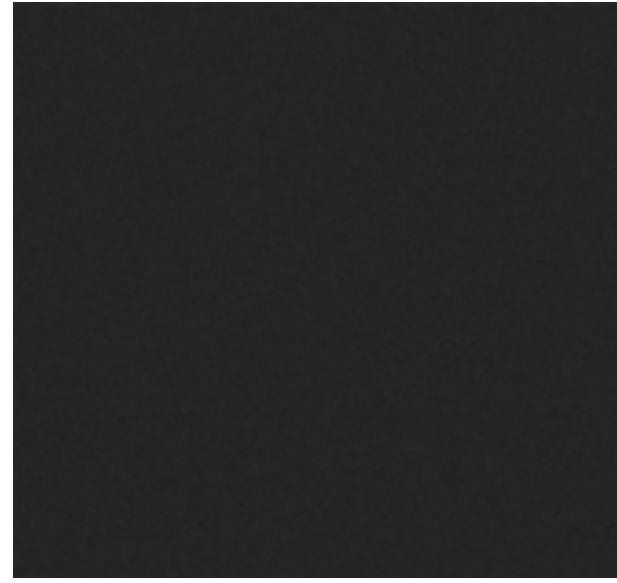
Applications



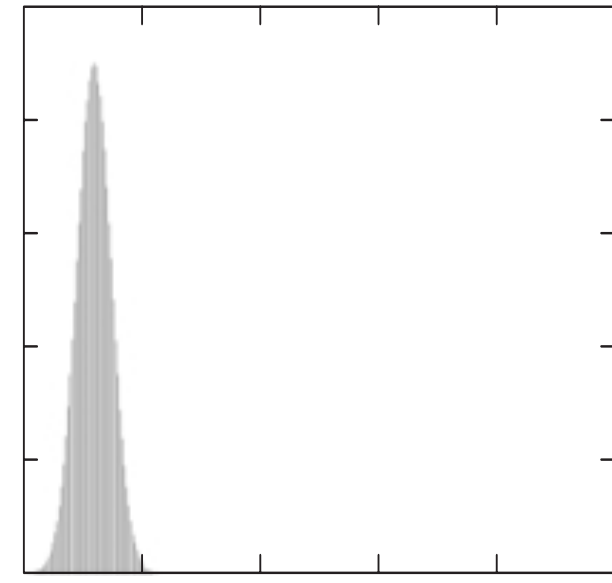
(a) Original Image



(b) Averaging $K = 128$ Images



(c) Difference between
(a) and (b)



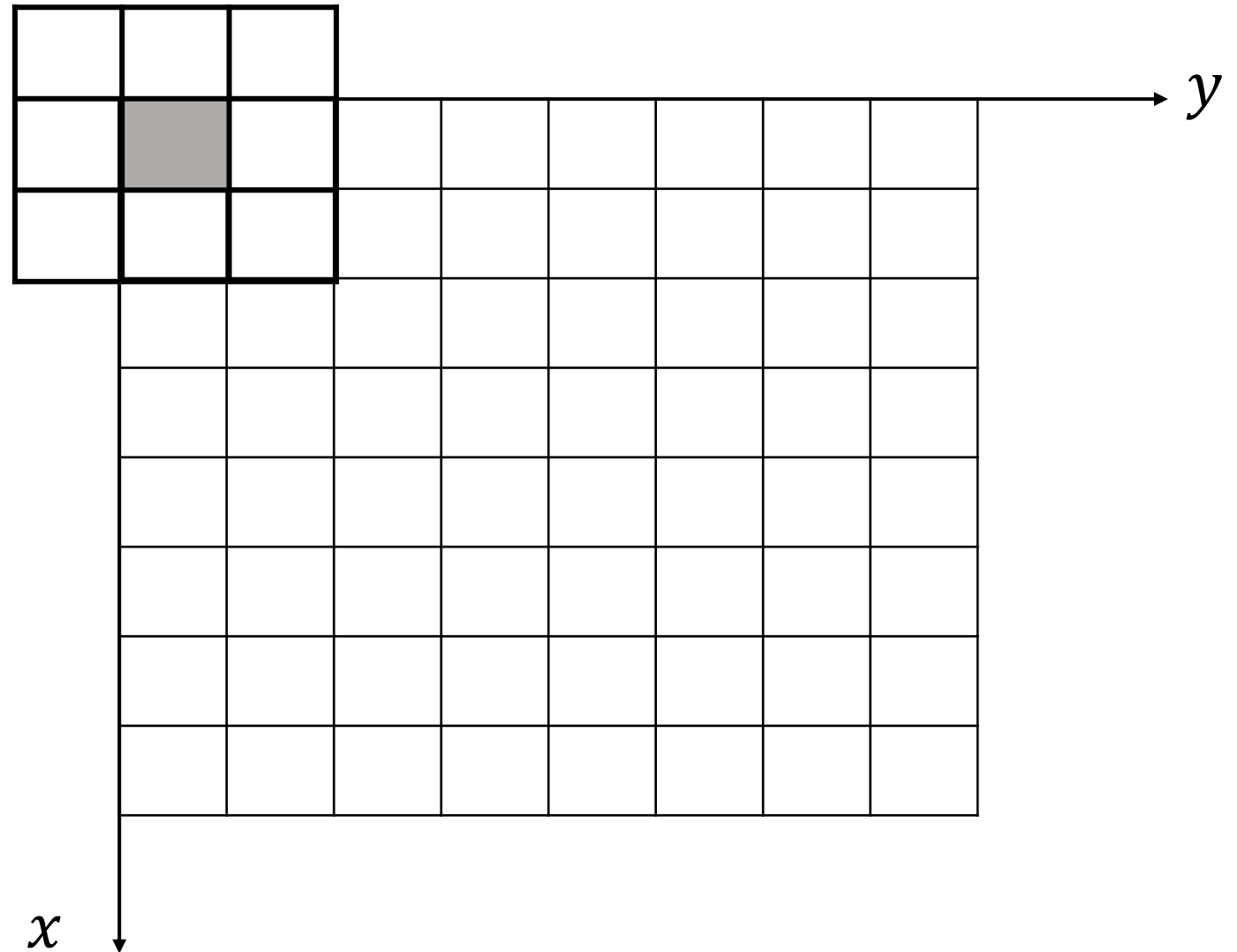
(d) Histogram of (c)

Spatial Filtering

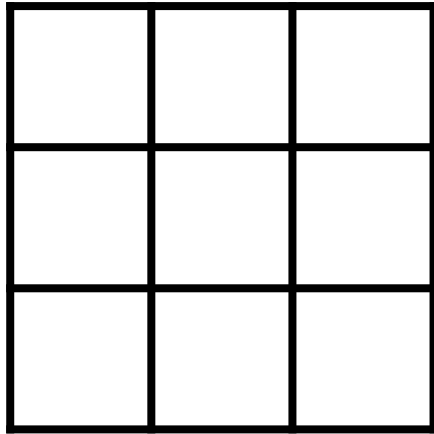
- ✓ Some neighborhood operations work with the values of the image pixels in the neighborhood and the corresponding values of a subimage that has the same dimensions as the neighborhood.
- ✓ The subimage is called a ***filter***, ***mask***, ***kernel***, *template*, or *window*.
- ✓ The values in a filter subimage are referred to as *coefficients*, rather than pixels.
- ✓ *Spatial filtering* is the filtering operations that are performed directly on the pixels of an image.

Spatial Filtering Mechanics

The process consists simply of moving the filter mask from point to point in an image.



Filtering Elements



Kernel g

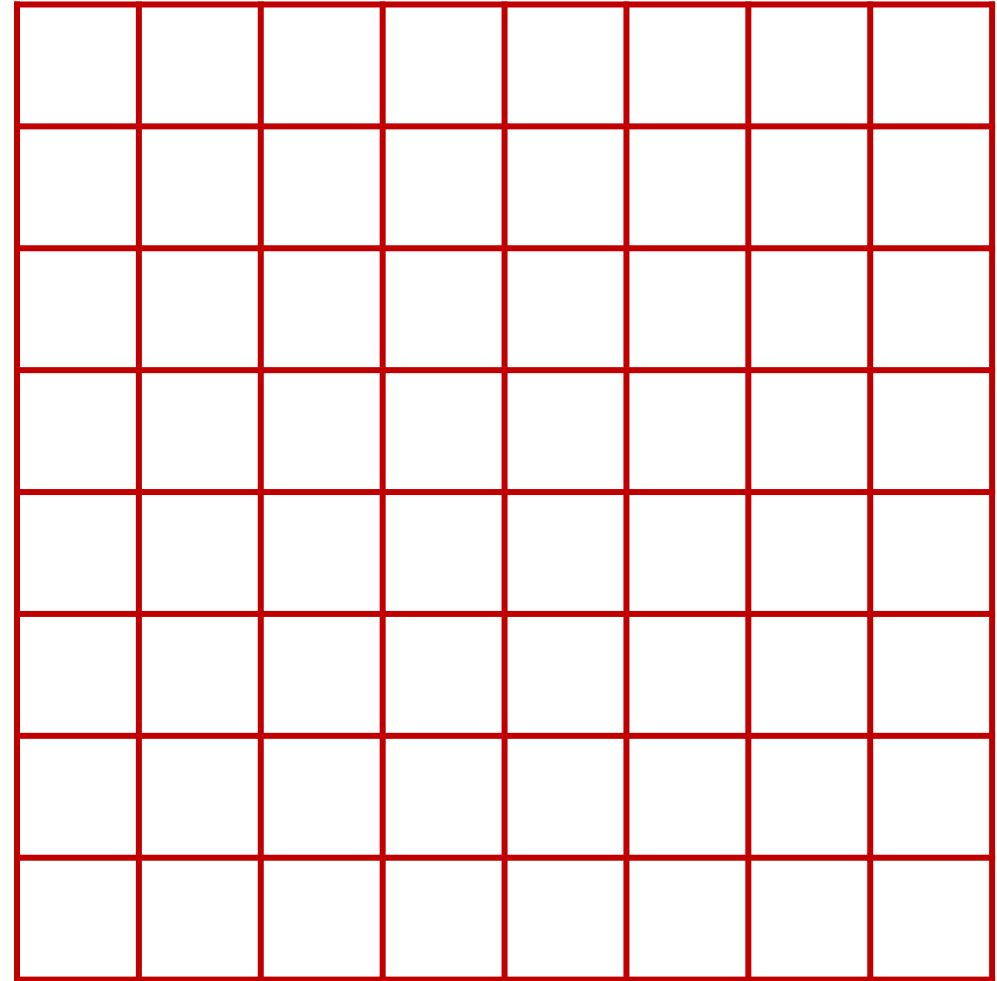


Image f

Position of the coefficients

$g(-1,-1)$	$g(-1,0)$	$g(-1,1)$
$g(0,-1)$	$g(0,0)$	$g(0,1)$
$g(1,-1)$	$g(1,0)$	$g(1,1)$

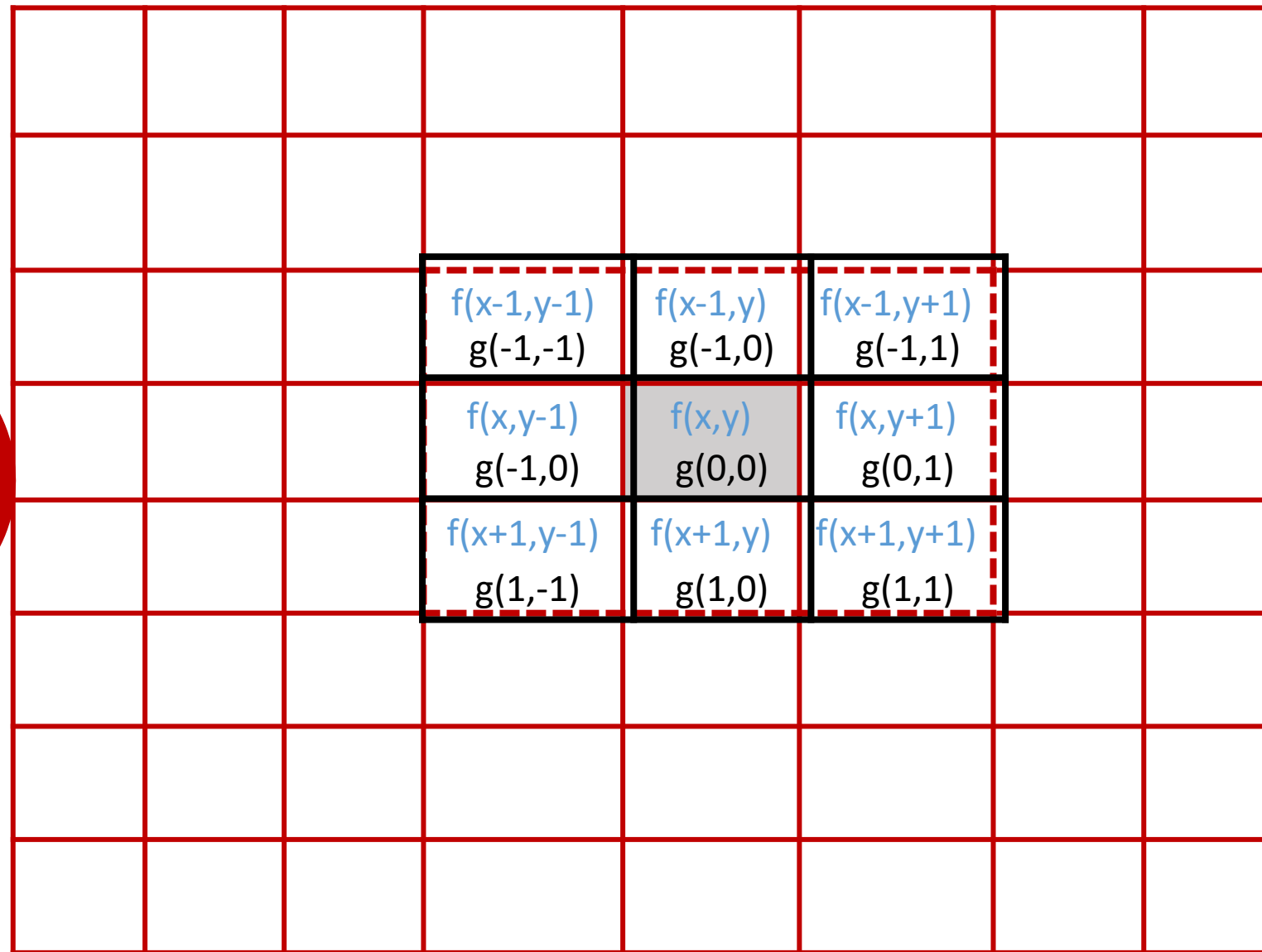
Kernel g

Image f

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

2D Correlation

At each point (x, y) , the response R of the filter at that point is calculated by a **sum of products** of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask



$$h = g \circ f = g(-1,-1)f(x-1,y-1) + g(-1,0)f(x-1,y) + \dots \\ + g(0,0)f(x,y) + \dots + g(1,0)f(x+1,y) + g(1,1)f(x+1,y+1).$$

2D Correlation

- ✓ Correlation is the mathematical operation on two functions f, g that produces a third function h .
- ✓ This new function expresses how the shape of function f is modified by the shape of function g . Correlation is denoted as follows:

$$h = f \circ g$$

- ✓ The mathematical 2D discrete correlation is given by:

$$h(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} g(i, j) f(x + i, y + j),$$

where f represents the input image to be correlated with the kernel g resulting in a new output image h . The indices x, y are concerned with the image matrices and the indices i, j deal with that of the kernel. If the size of the kernel involved in correlation is $N \times N$ then the indices i, j will range from $[-N/2]$ to $[N/2]$ where N is usually an odd number.

Numerical Example

1	2	1
2	4	2
1	2	1

Kernel

*

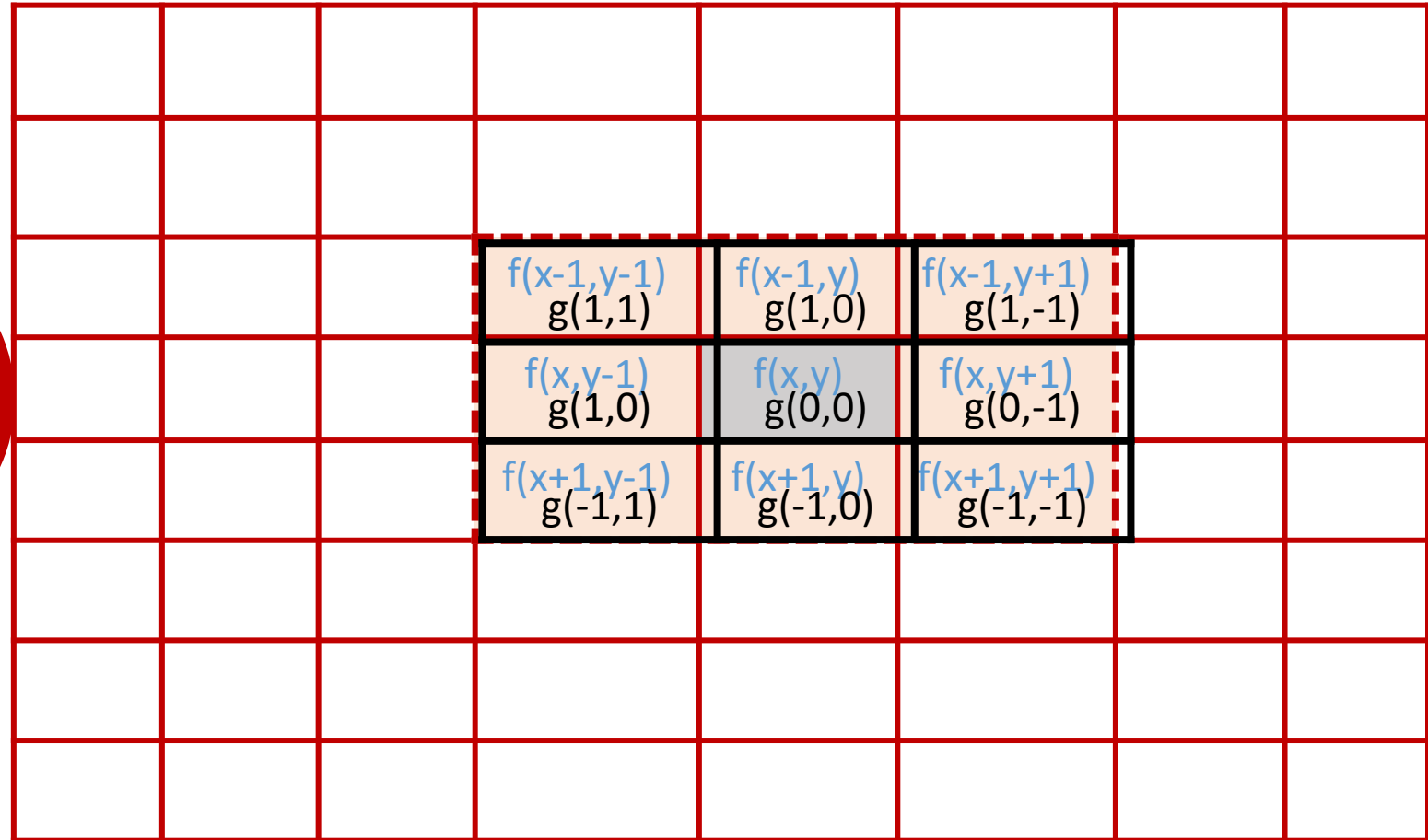
1	29	13	2	3	20	17	26
24	33	32	7	9	10	4	2
14	10	2	21	1	18	22	21
15	7	4	14	19	3	10	13
16	14	8	16	4	17	38	7
11	25	6	2	3	31	36	21
24	10	3	9	11	28	21	10
33	2	19	7	10	22	6	25

Image

$$h = g \circ f = 1 \times 21 + 2 \times 1 + 1 \times 18 + 2 \times 14 + 4 \times 19 + 2 \times 3 + 1 \times 16 + 2 \times 4 + 1 \times 17 = 192.$$

2D Convolution

At each point (x, y) , the response R of the filter at that point is calculated by a **sum of products** of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask



$$h = g * f = g(1,1)f(x-1,y-1) + g(1,0)f(x-1,y) + \dots \\ + g(0,0)f(x,y) + \dots + g(-1,0)f(x+1,y) + g(-1,-1)f(x+1,y+1).$$

2D Convolution

- ✓ Convolution is the mathematical operation on two functions f, g that produces a third function h .
- ✓ This new function expresses how the shape of function f is modified by the shape of function g . Convolution is denoted as follows:

$$h = f * g$$

- ✓ The mathematical 2D discrete convolution is given by:

$$h(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} g(i, j) f(x - i, y - j),$$

where f represents the input image to be convolved with the kernel g resulting in a new output image h . The indices x, y are concerned with the image matrices and the indices i, j deal with that of the kernel. If the size of the kernel involved in convolution is $N \times N$ then the indices i, j will range from $[-N/2]$ to $[N/2]$ where N is usually an odd number.

Numerical Examples

1	2	1
2	4	2
1	2	1

Kernel

*

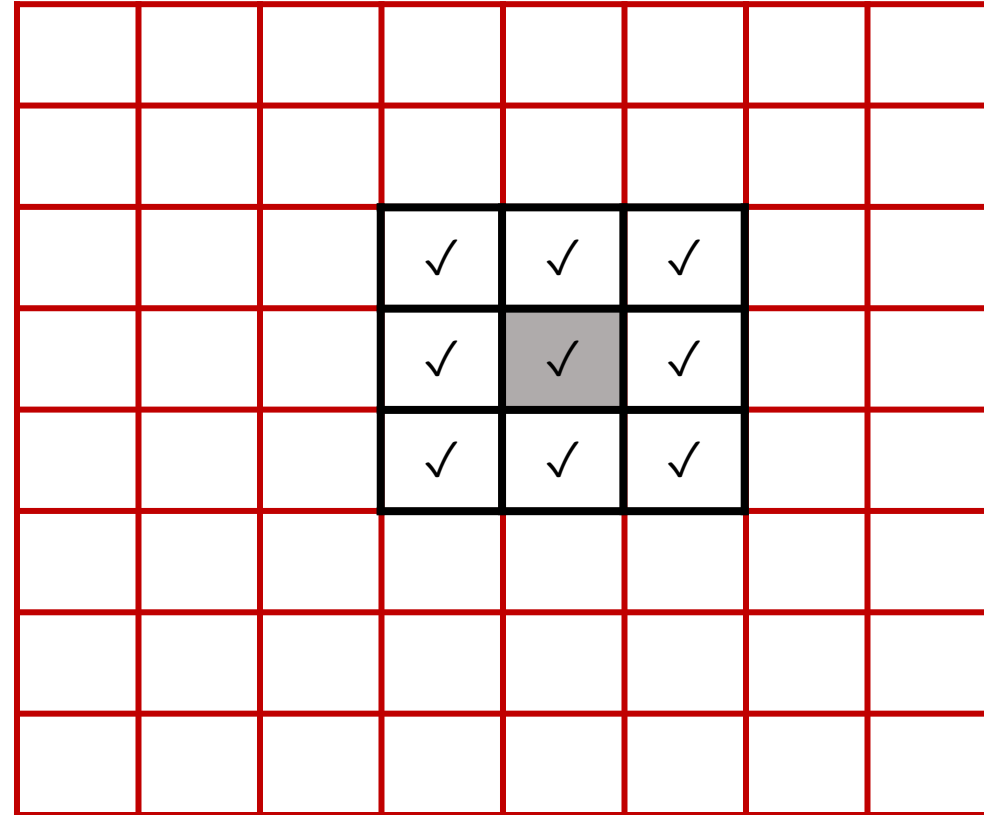
1	29	13	2	3	20	17	26
24	33	32	7	9	10	4	2
14	10	2	21	1	18	22	21
15	7	4	14	19	3	10	13
16	14	8	16	4	17	38	7
11	25	6	2	3	31	36	21
24	10	3	9	11	28	21	10
33	2	19	7	10	22	6	25

Image

$$h = g * f = 1 \times 17 + 2 \times 4 + 1 \times 16 + 2 \times 3 + 4 \times 19 + 2 \times 14 + 1 \times 18 + 2 \times 1 + 1 \times 21 = 192.$$

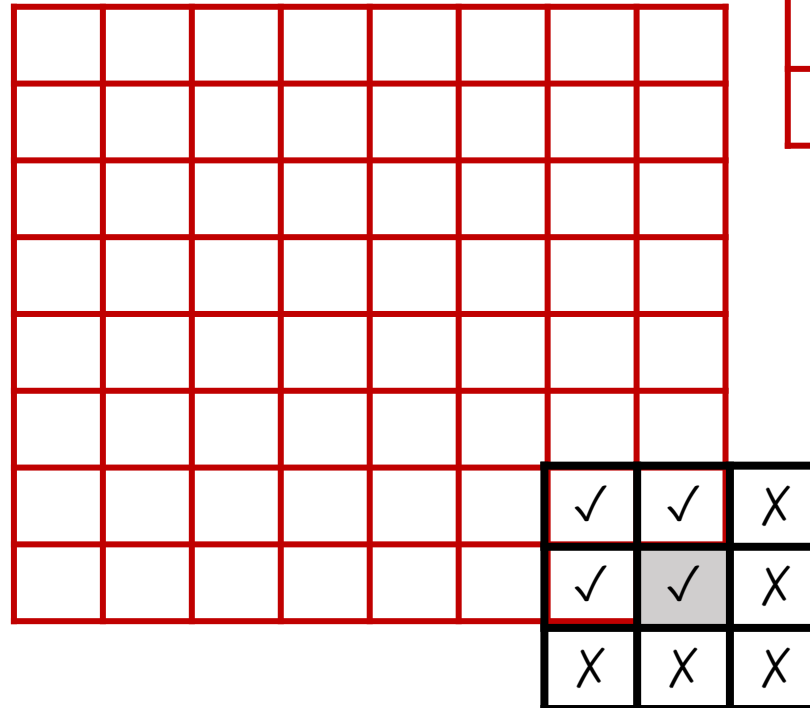
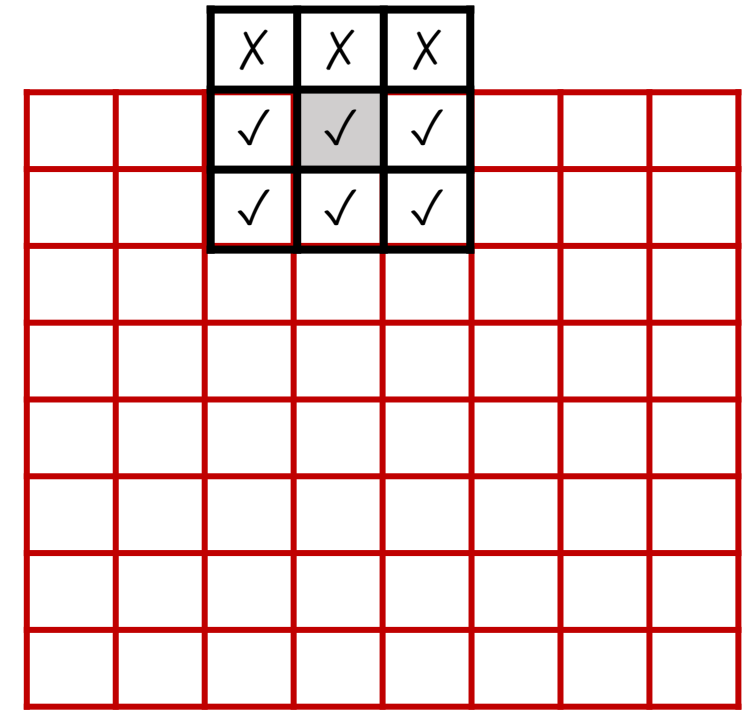
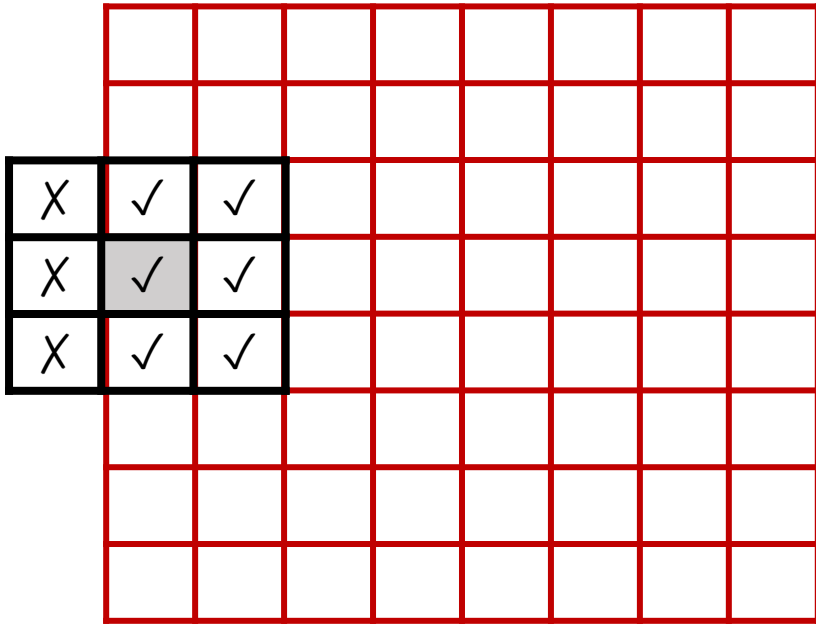
2D Convolution

At each point (x, y) , the response R of the filter at that point is calculated by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask



$$h(x, y) = g(x, y) * f(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} g(i, j) f(x - i, y - j),$$

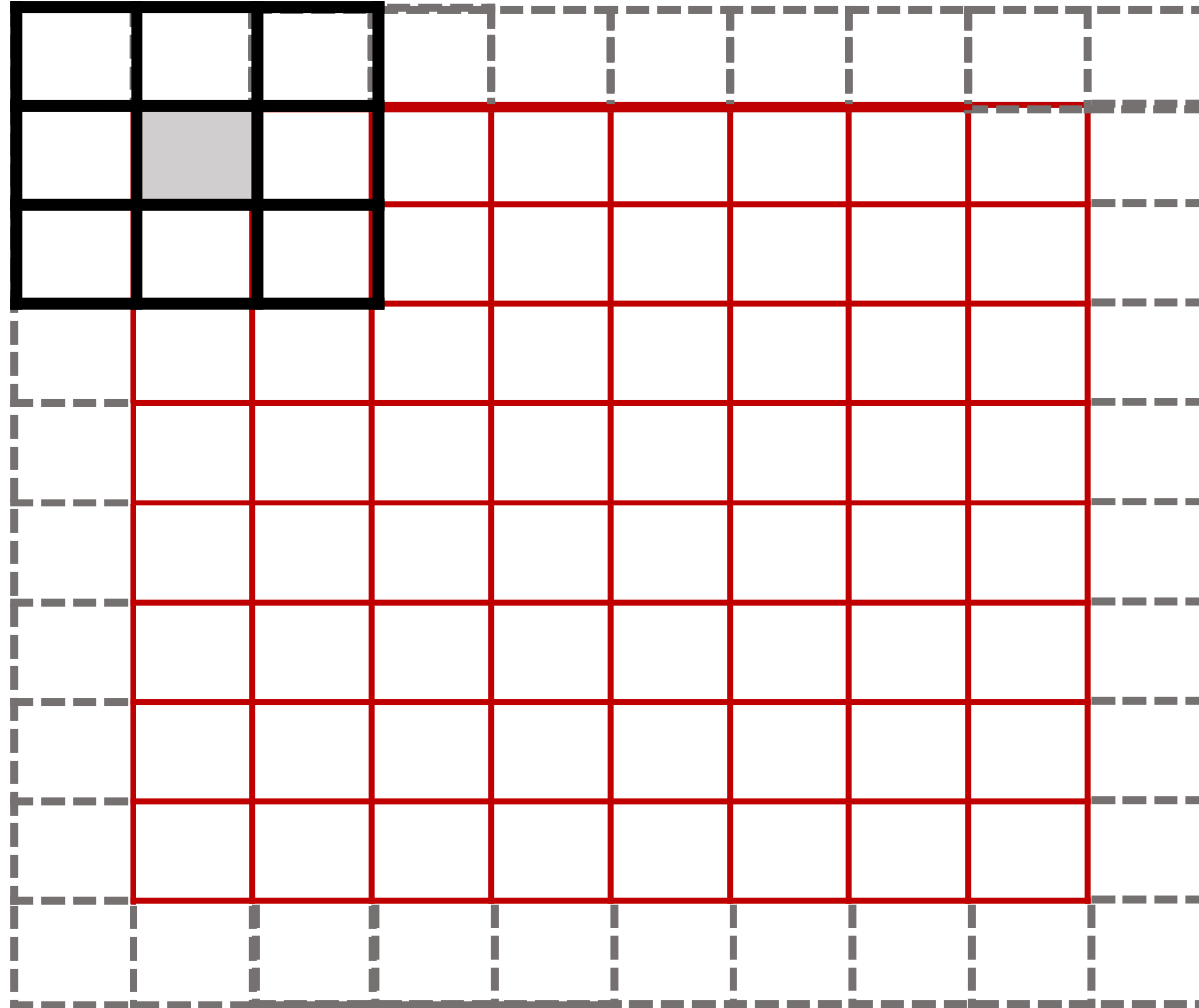
Boundary Conditions



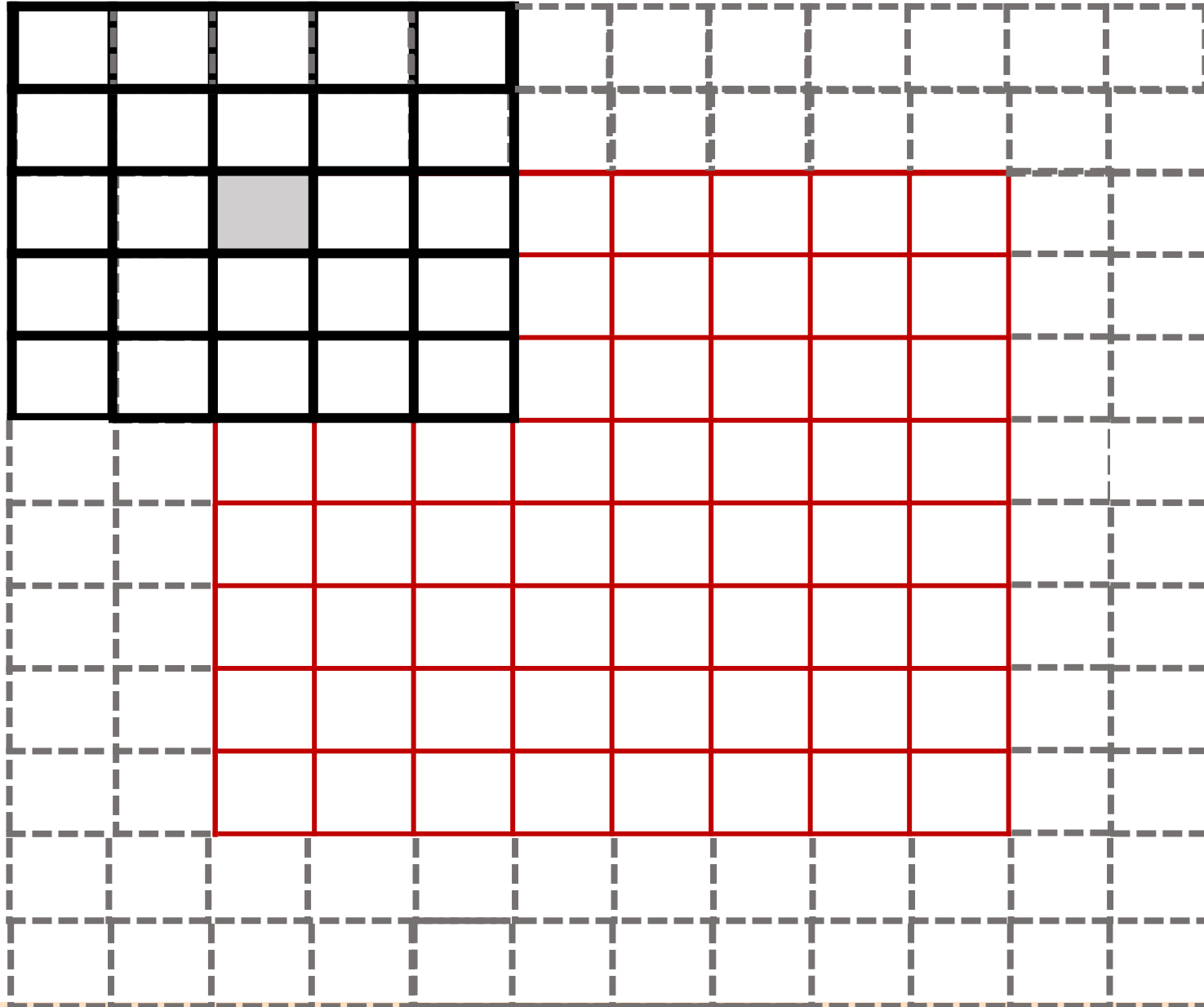
Padding Approaches

- ✓ Zero
- ✓ Symmetric
- ✓ Circular
- ✓ Replicate

Solution → Padding for a 3x3 kernel



Solution → Padding for a 5x5 kernel



Zero padding

0	0	0	0	0	0	0	0	0	0
0	1	29	13	2	3	20	17	26	0
0	24	33	32	7	9	10	4	2	0
0	14	10	2	21	1	18	22	21	0
0	15	7	4	14	19	3	10	13	0
0	16	14	8	16	4	17	38	7	0
0	11	25	6	2	3	31	36	21	0
0	24	10	3	9	11	28	21	10	0
0	33	2	19	7	10	22	6	25	0
0	0	0	0	0	0	0	0	0	0

Replicate padding

1	1	29	13	2	3	20	17	26	26
1	1	29	13	2	3	20	17	26	26
24	24	33	32	7	9	10	4	2	2
14	14	10	2	21	1	18	22	21	21
15	15	7	4	14	19	3	10	13	13
16	16	14	8	16	4	17	38	7	7
11	11	25	6	2	3	31	36	21	21
24	24	10	3	9	11	28	21	10	10
33	33	2	19	7	10	22	6	25	25
33	33	2	19	7	10	22	6	25	25

Symmetric padding

Attention!
Replicate
≠
Symmetric

1	1	29	13	2	3	20	17	26	26
1	1	29	13	2	3	20	17	26	26
24	24	33	32	7	9	10	4	2	2
14	14	10	2	21	1	18	22	21	21
15	15	7	4	14	19	3	10	13	13
16	16	14	8	16	4	17	38	7	7
11	11	25	6	2	3	31	36	21	21
24	24	10	3	9	11	28	21	10	10
33	33	2	19	7	10	22	6	25	25
33	33	2	19	7	10	22	6	25	25

Replicate padding

1	1	1	29	13	2	3	20	17	26	26	26
1	1	1	29	13	2	3	20	17	26	26	26
1	1	1	29	13	2	3	20	17	26	26	26
24	24	24	33	32	7	9	10	4	2	2	2
14	14	14	10	2	21	1	18	22	21	21	21
15	15	15	7	4	14	19	3	10	13	13	13
16	16	16	14	8	16	4	17	38	7	7	7
11	11	11	25	6	2	3	31	36	21	21	21
24	24	24	10	3	9	11	28	21	10	10	10
33	33	33	2	19	7	10	22	6	25	25	25
33	33	33	2	19	7	10	22	6	25	25	25
33	33	33	2	19	7	10	22	6	25	25	25

Symmetric padding

33	24	24	33	32	7	9	10	4	2	2	4
29	1	1	29	13	2	3	20	17	26	26	17
29	1	1	29	13	2	3	20	17	26	26	17
33	24	24	33	32	7	9	10	4	2	2	4
10	14	14	10	2	21	1	18	22	21	21	22
7	15	15	7	4	14	19	3	10	13	13	10
14	16	16	14	8	16	4	17	38	7	7	38
25	11	11	25	6	2	3	31	36	21	21	36
10	24	24	10	3	9	11	28	21	10	10	21
2	33	33	2	19	7	10	22	6	25	25	6
2	33	33	2	19	7	10	22	6	25	25	6
10	24	24	10	3	9	11	28	21	10	10	21

Lowpass Filters

- ✓ The response of a smoothing, **linear spatial filter** is simply the average of the pixels contained in the neighborhood of the filter mask.
- ✓ These filters sometimes are called ***averaging filters***.
- ✓ By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood
- ✓ this process results in an image with reduced “sharp” transitions in gray levels.

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

1	1	1
1	1	1
1	1	1

Mean Filter

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

Gaussian Filter

Applications

- ✓ **Noise:** The most obvious application of smoothing is noise reduction, because random noise typically consists of sharp transitions in gray levels.
- ✓ **Edges:** The sharp transitions in gray levels characterize the edges (which almost always are desirable features of an image), so averaging filters have the undesirable side effect that they blur edges.
- ✓ **Contours:** The smoothing of false contours that result from using an insufficient number of gray levels.

Standard Average Filter

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

1	1	1
1	1	1
1	1	1

Mean Filter

- ✓ The standard average filter can best be seen by substituting the coefficients of the mask.

$$R = \frac{1}{9} \sum_{i=1}^9 z_i$$

which is the average of the gray levels of the pixels in the 3×3 neighborhood defined by the mask.

- ✓ An $m \times n$ mask would have a normalizing constant equal to $\frac{1}{mn}$.
- ✓ A spatial averaging filter in which all coefficients are equal is sometimes called a **box filter**.

Weighted Average Filter

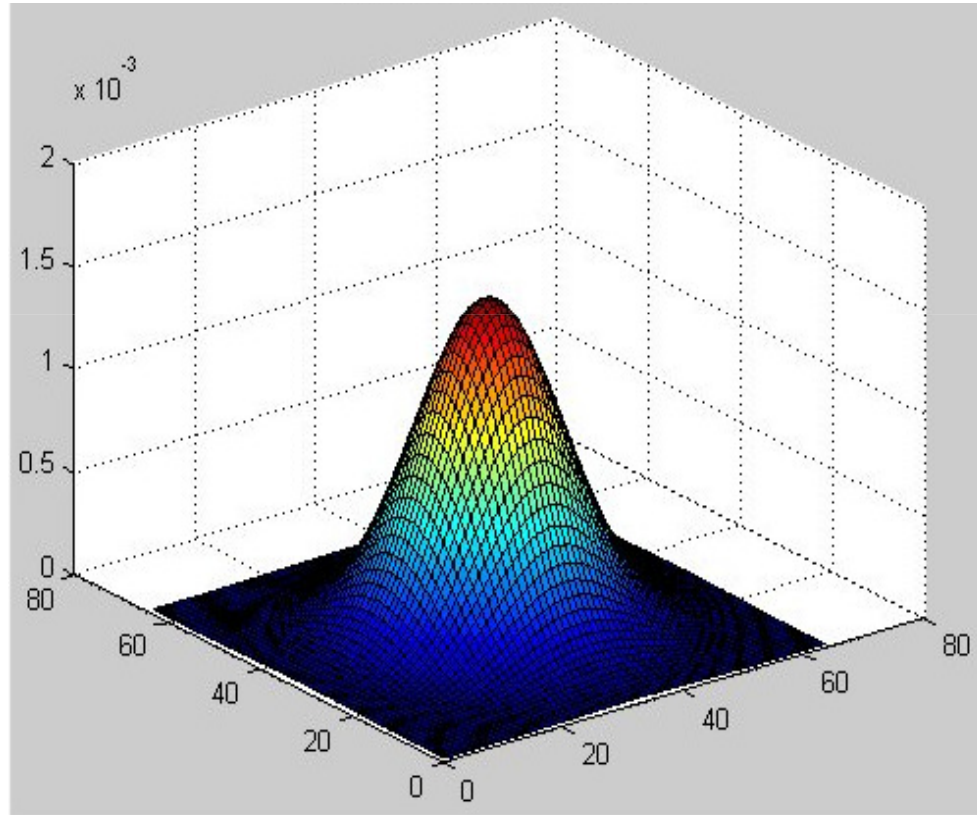
$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

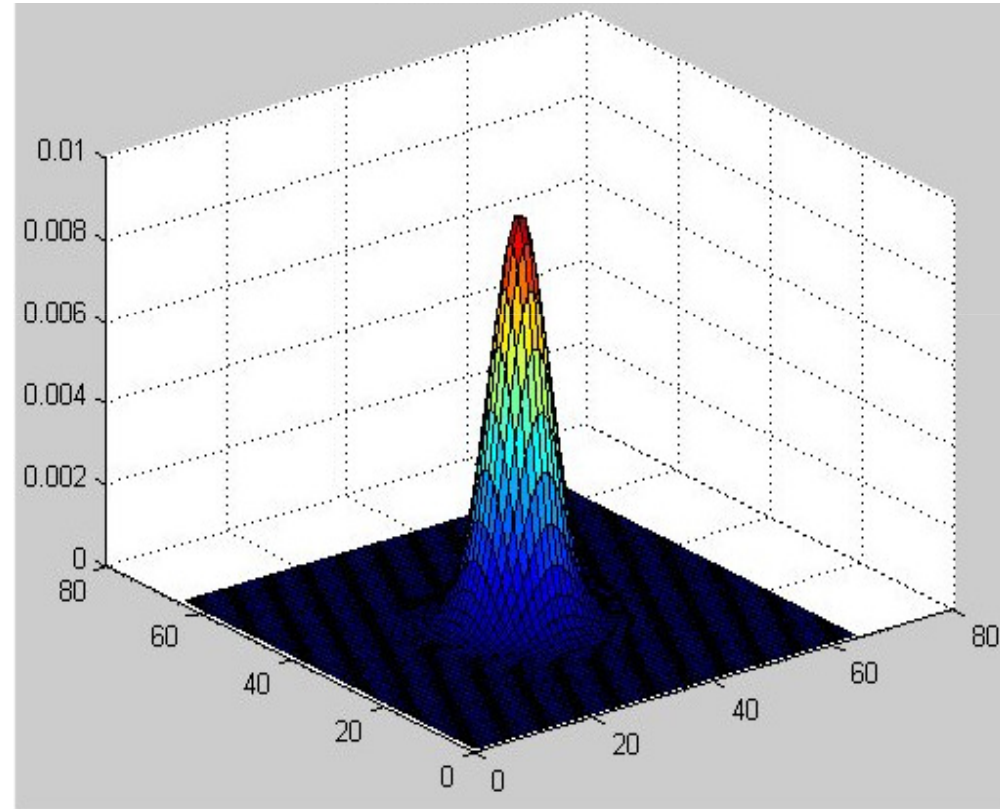
Gaussian Filter

- ✓ This terminology is used to indicate that pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others.
- ✓ The basic strategy is simply an attempt to reduce blurring in the smoothing process as follows
 - ✓ the center point is weighted by the highest value
 - ✓ the value of the coefficients as a function of increasing distance from the origin.

Gaussian Filter



Gaussian 64x64 with $\sigma = 10$.



Gaussian 64x64 with $\sigma = 4$.

$$G(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right).$$

1st Example (1/2)



(a) Original



(b) Average Filter $n = 3$

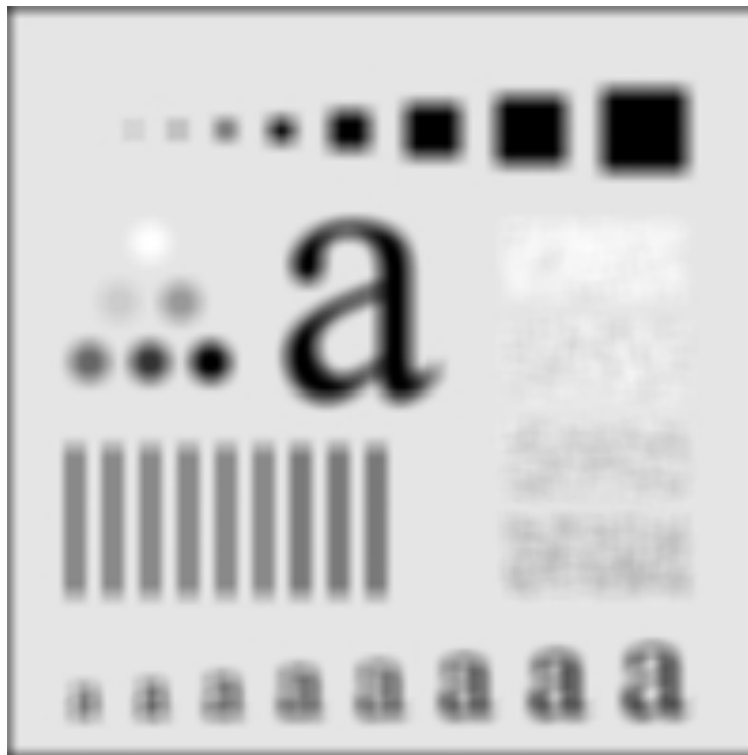


(c) Average Filter $n = 5$

1st Example (2/2)



(d) Average Filter $n = 9$

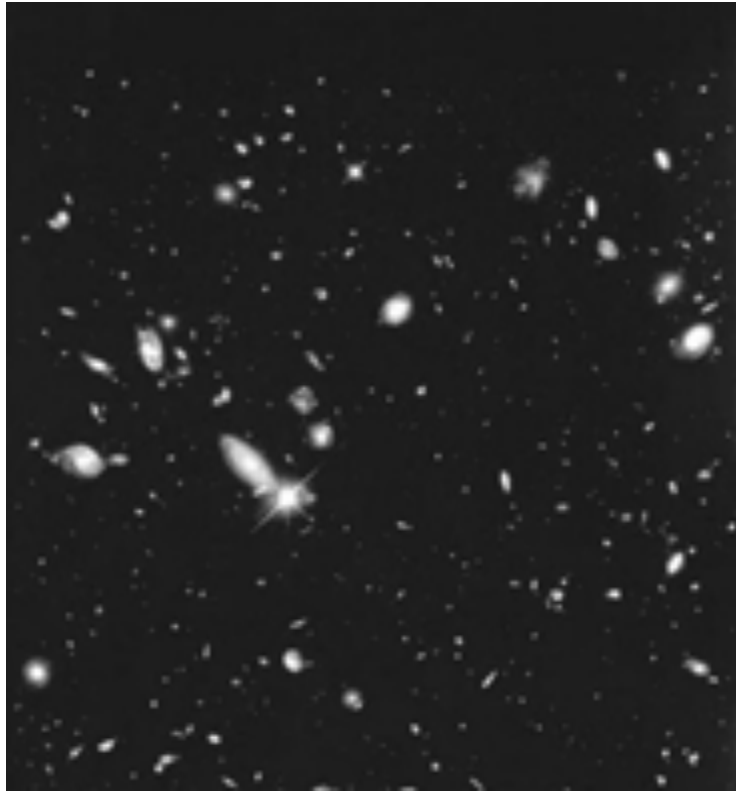


(e) Average Filter $n = 15$

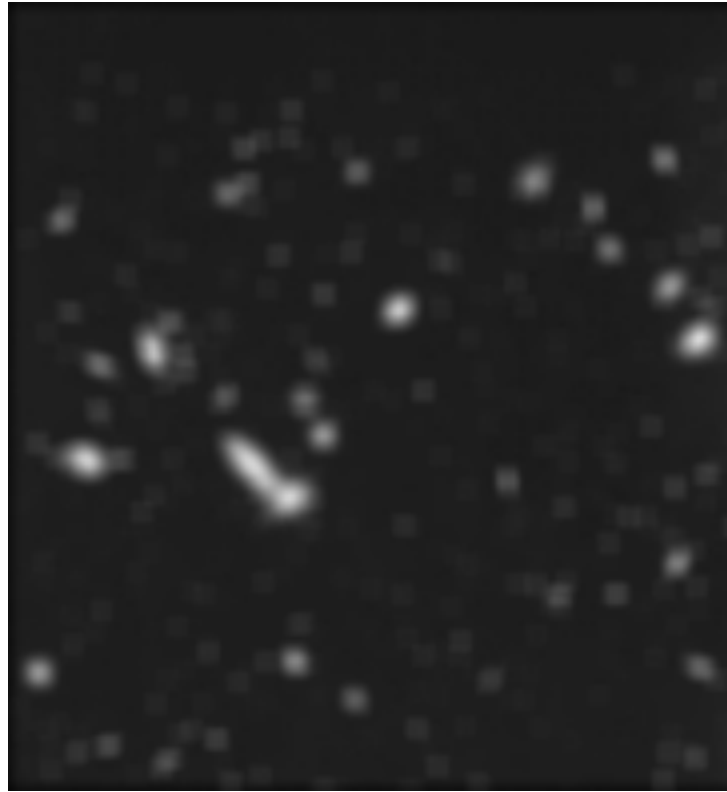


(f) Average Filter $n = 35$

2nd Example



(a) Original NASA image



(b) Average Filter 15x15



(c) Thresholding

Order-statistics Filters

- ✓ Order-statistics filters are **nonlinear spatial filters** whose response is based on
 - ✓ ordering (ranking) the pixels contained in the image area encompassed by the filter,
 - ✓ replacing the value of the center pixel with the value determined by the ranking result.
- ✓ The best-known example in this category is the ***median filter***.
- ✓ Median filters provide excellent **noise-reduction** capabilities, with considerably less blurring than linear smoothing filters of similar size, the presence of *impulse noise*, also called *salt-and-pepper noise*

Median Filter

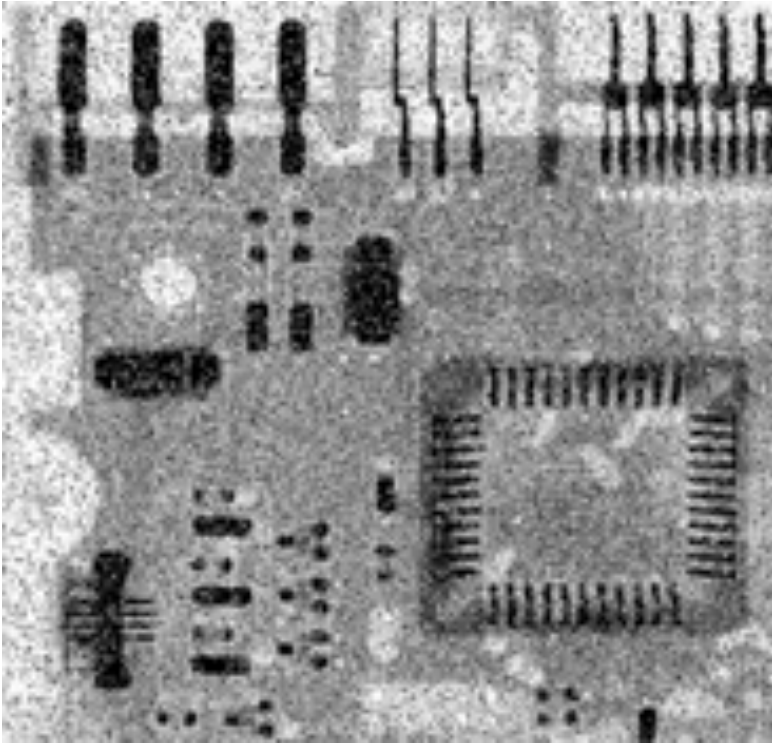
33	32	7	9	10	4
10	2	21	1	18	22
7	4	14	19	3	10
14	8	16	4	17	38
25	6	2	3	31	36
10	3	9	11	28	21

33	32	7	9	10	4
10	2	21	1	18	22
7	4	14	16	3	10
14	8	16	4	17	38
25	6	2	3	31	36
10	3	9	11	28	21

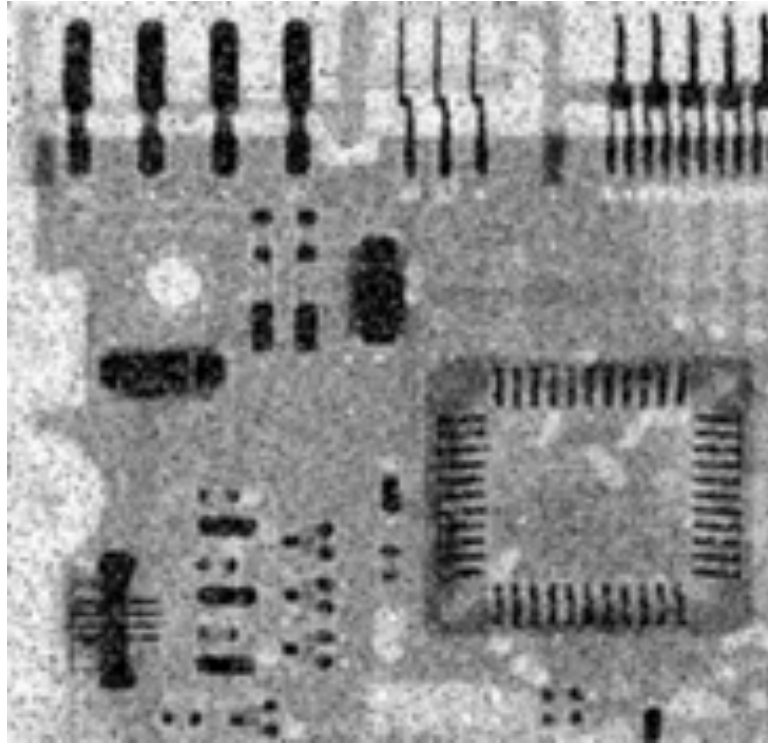
21	1	18	14	19	3	16	4	17
----	---	----	----	----	---	----	---	----

1	3	4	14	16	17	18	19	21
---	---	---	----	----	----	----	----	----

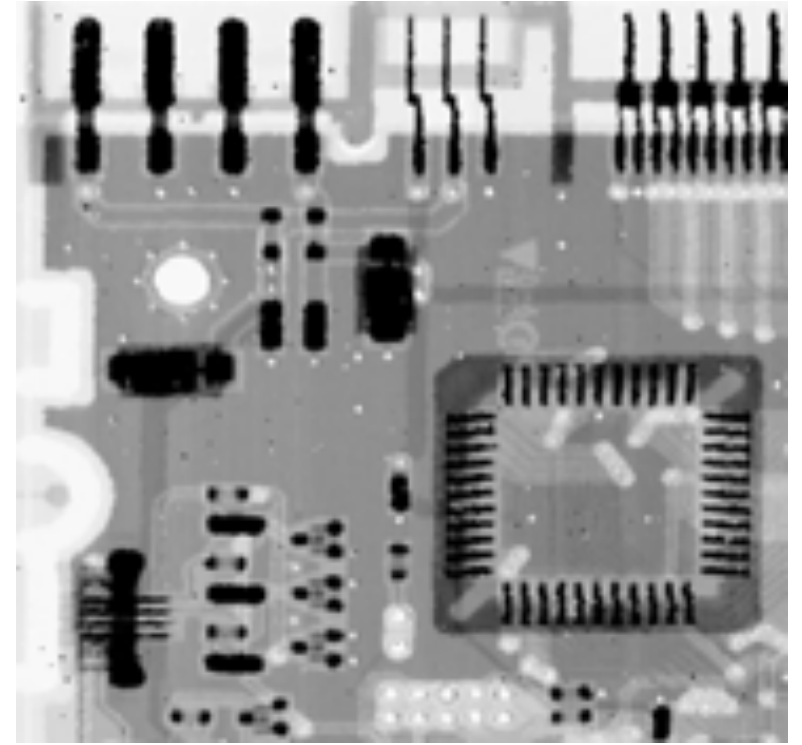
Example



(a) X-ray image with salt and pepper noise.



(b) Average Filter 3x3



(c) Median Filter 3x3

Min Filter

33	32	7	9	10	4
10	2	21	1	18	22
7	4	14	19	3	10
14	8	16	4	17	38
25	6	2	3	31	36
10	3	9	11	28	21

33	32	7	9	10	4
10	2	21	1	18	22
7	4	14	1	3	10
14	8	16	4	17	38
25	6	2	3	31	36
10	3	9	11	28	21

21	1	18	14	19	3	16	4	17
----	---	----	----	----	---	----	---	----

Max Filter

33	32	7	9	10	4
10	2	21	1	18	22
7	4	14	19	3	10
14	8	16	4	17	38
25	6	2	3	31	36
10	3	9	11	28	21

33	32	7	9	10	4
10	2	21	1	18	22
7	4	14	21	3	10
14	8	16	4	17	38
25	6	2	3	31	36
10	3	9	11	28	21

21	1	18	14	19	3	16	4	17
----	---	----	----	----	---	----	---	----

Sharpening Spatial Filters

- ✓ The principal objective of sharpening is to highlight **fine details** in an image or to enhance detail that has been blurred, The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred.
- ✓ Sharpening could be accomplished by spatial differentiation
- ✓ The strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.
- ✓ Image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray-level values.

Sharpening Spatial Filters

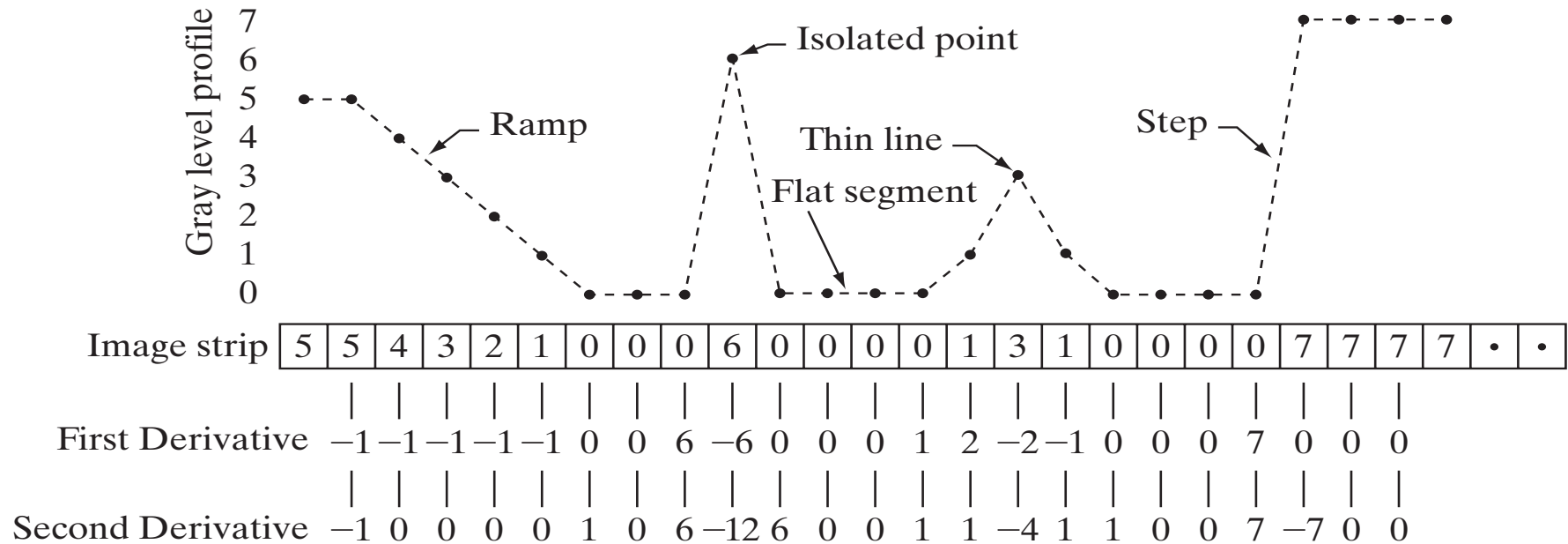
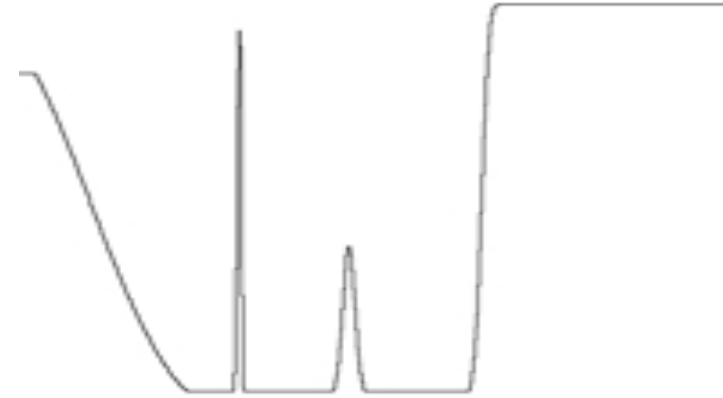
- ✓ A basic definition of the first-order derivative of a one-dimensional function $f(x)$ is the difference

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x).$$

- ✓ Similarly, we define a second-order derivative as the difference

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

Example



Second Derivatives for Enhancement

- ✓ The approach basically consists of defining a discrete formulation of the second-order derivative and then constructing a filter mask based on that formulation.
- ✓ We are interested in ***isotropic*** filters, whose response is independent of the direction of the discontinuities in the image to which the filter is applied.
- ✓ Isotropic filters are ***rotation invariant***, in the sense that rotating the image and then applying the filter gives the same result as applying the filter to the image first and then rotating the result.

The Laplacian

- ✓ the *Laplacian* is the simplest isotropic derivative operator, which, for a function (image) $f(x, y)$ of two variables, is defined as

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

- ✓ The partial second-order derivative in the x -direction and y -direction :

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

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

- ✓ The digital implementation of the two-dimensional Laplacian is obtained by summing these two components:

$$\nabla^2 f = [f(x + 1, y) + f(x - 1, y) - 2f(x, y) + f(x, y + 1) + f(x, y - 1) - 2f(x, y)]$$

Example

0	1	0
1	-4	1
0	1	0

(a) Laplacian mask

1	1	1
1	-8	1
1	1	1

(b) Extension of (a)
that include diagonal
neighbors

0	-1	0
-1	4	-1
0	-1	0

(c) Other Laplacian mask

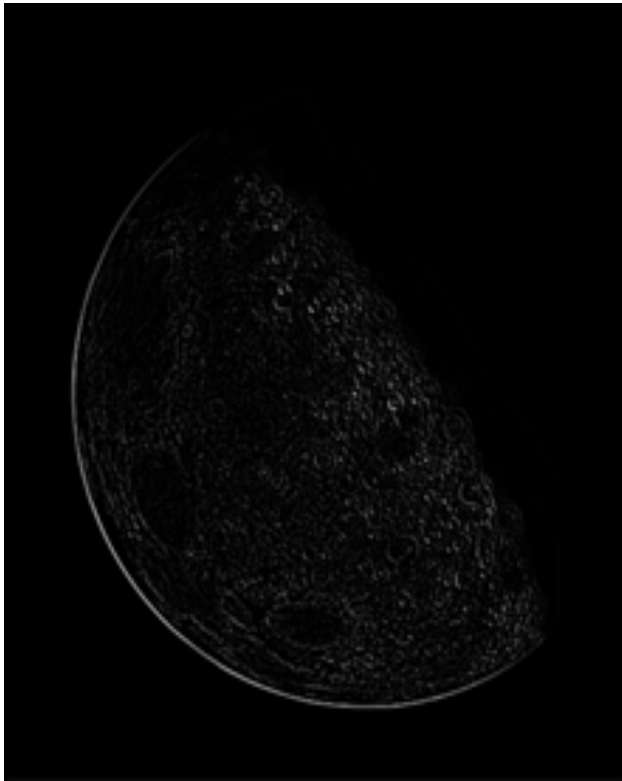
-1	-1	-1
-1	8	-1
-1	-1	-1

(d) Extension of (c)
that include diagonal
neighbors

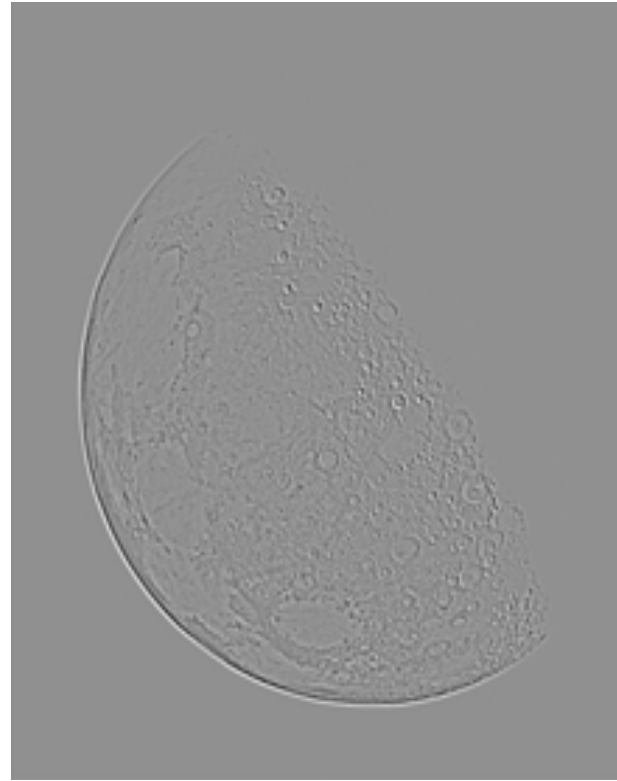
Example



(a) Original NASA image



(b) Laplacian Filtered Image



(c) Laplacian scaled filtered image



(d) Image enhanced

Edge detection filters

0	0	-1
0	1	0
0	0	0

(a) Roberts columns

-1	0	1
-1	0	1
-1	0	1

(b) Prewitt columns

-1	0	1
-2	0	2
-1	0	1

(c) Sobel columns

-1	0	1
-2	0	2
-1	0	1

(d) Robinson columns

-1	0	0
0	1	0
0	0	0

(a) Roberts rows

1	1	1
0	0	0
-1	-1	-1

(b) Prewitt rows

1	2	1
0	0	0
-1	-2	-1

(c) Sobel rows

1	2	1
0	0	0
-1	-2	-1

(d) Robinson rows

Sobel filter

Initial Image								
	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	1	1	1	1	0	0
4	0	0	1	1	1	1	0	0
5	0	0	1	1	1	1	0	0
6	0	0	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

+

Sobel row			
	1	2	3
1	1	2	1
2	0	0	0
3	-1	-2	-1

Sobel column			
	1	2	3
1	1	0	-1
2	2	0	-2
3	1	0	-1

Convolution with Sobel row								
	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	1	3	4	4	3	1	0
3	0	1	3	4	4	3	1	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	-1	-3	-4	-4	-3	-1	0
7	0	-1	-3	-4	-4	-3	-1	0
8	0	0	0	0	0	0	0	0

=

Convolution with Sobel column								
	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	1	3	0	0	-1	-1	0
3	0	1	3	0	0	-3	-3	0
4	0	4	4	0	0	-4	-4	0
5	0	4	4	0	0	-4	-4	0
6	0	-1	-3	0	0	-3	-1	0
7	0	-1	-3	0	0	-3	-1	0
8	0	0	0	0	0	0	0	0

$abs(Sr * f) + abs(Sc * f)$								
	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	2	6	4	4	4	2	0
3	0	2	6	4	4	6	4	0
4	0	4	4	0	0	4	4	0
5	0	4	4	0	0	4	4	0
6	0	2	6	4	4	6	2	0
7	0	2	6	4	4	6	2	0
8	0	0	0	0	0	0	0	0

Applications: Feature Extraction



Original



Sobel



Canny



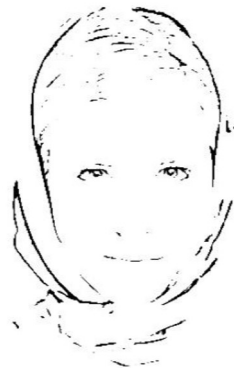
LoG



Thresholded
DoG



XDoG



WDoG
t = 26ms



WDoG
t = 31ms



WDoG
t = 36ms



WDoG
t = 41ms

Course Outline

- ✓ Digital Image Fundamentals
- ✓ Image Acquisition
- ✓ Image Transforms
- ✓ Convolution Theorem
- ✓ Morphological Operations
- ✓ Multiresolution Analysis
- ✓ Image Coding

Fourier Transform

- ✓ The **Fourier Transform** of a **continuous** function $f(t)$ of a **continuous** variable t denoted as $\mathfrak{F}\{f(t)\}$ is defined by

$$\mathfrak{F}\{f(t)\} = \int_{-\infty}^{+\infty} f(t)e^{-j\theta t} dt = \int_{-\infty}^{+\infty} f(t)e^{-j2\pi\mu t} dt,$$

where $\theta = 2\pi\mu$ is the angle and μ is the frequency.

- ✓ The Fourier Transform is a **function of frequency** $\mathfrak{F}\{f(t)\} = F(\mu)$
- ✓ Using the Euler's formula the Fourier Transform can be also written as

$$F(\mu) = \int_{-\infty}^{+\infty} f(t)[\cos(2\pi\mu t) - j \sin(2\pi\mu t)] dt$$

Inverse Fourier Transform

- ✓ The **Inverse Fourier Transform** of a **continuous** function $f(t)$ of a **continuous** variable t denoted is defined as

$$f(t) = \mathfrak{F}^{-1}\{F(\mu)\} = \int_{-\infty}^{+\infty} F(\mu) e^{j\theta t} d\mu = \int_{-\infty}^{+\infty} F(\mu) e^{j2\pi\mu t} d\mu,$$

where $\theta = 2\pi\mu$ is the angle and μ is the frequency.

- ✓ The Fourier Transform and the inverse Fourier Transform are so-called **Fourier transform pair**.
- ✓ We also call the Fourier Domain as **Frequency Domain** because the only variable left after the integration is **frequency**.

Convolution Theorem

- ✓ The convolution in space domain between two continuous function $f(t)$ and $h(t)$ of one continuous variable t is analogous to multiplication in frequency domain of the Fourier Transform of these functions

$$f(t) * h(t) \Leftrightarrow F(\mu)H(\mu)$$

- ✓ In addition the convolution in frequency domain equals the multiplication in the spatial domain

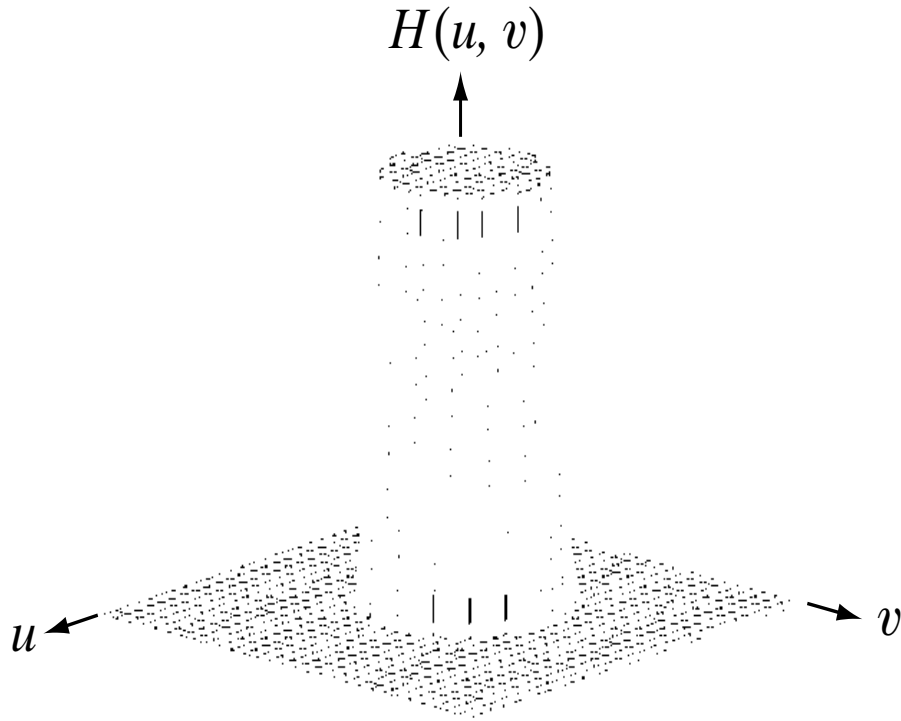
$$F(\mu) * H(\mu) \Leftrightarrow f(t)h(t)$$

NOTE! The double arrow is used to indicate the the expression on the right is obtained by taking the Fourier Transform of the expression on the left, while the expression on the left is obtained by taking the inverse Fourier transform of the expression on the right.

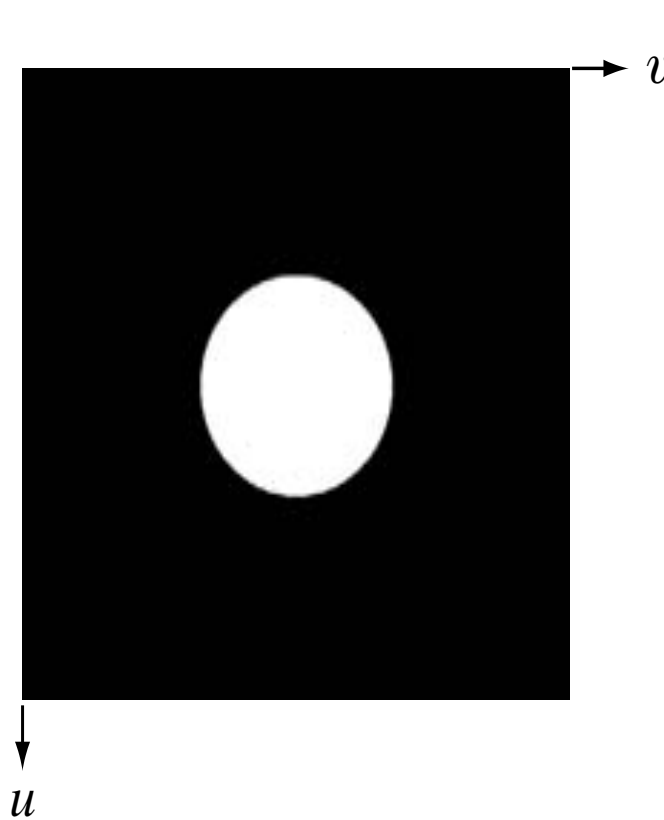
Lowpass Filtering

- Smoothing (blurring) is achieved in the frequency domain by high-frequency attenuation; by low pass filtering
 - The *Ideal filter* is a very sharp filtering
 - The *Gaussian filter* is a very smooth filtering
 - And the *Butterworth filtering* has a parameter called the filter order. It approaches the ideal filter for high order values and the Gaussian filter for low order values.

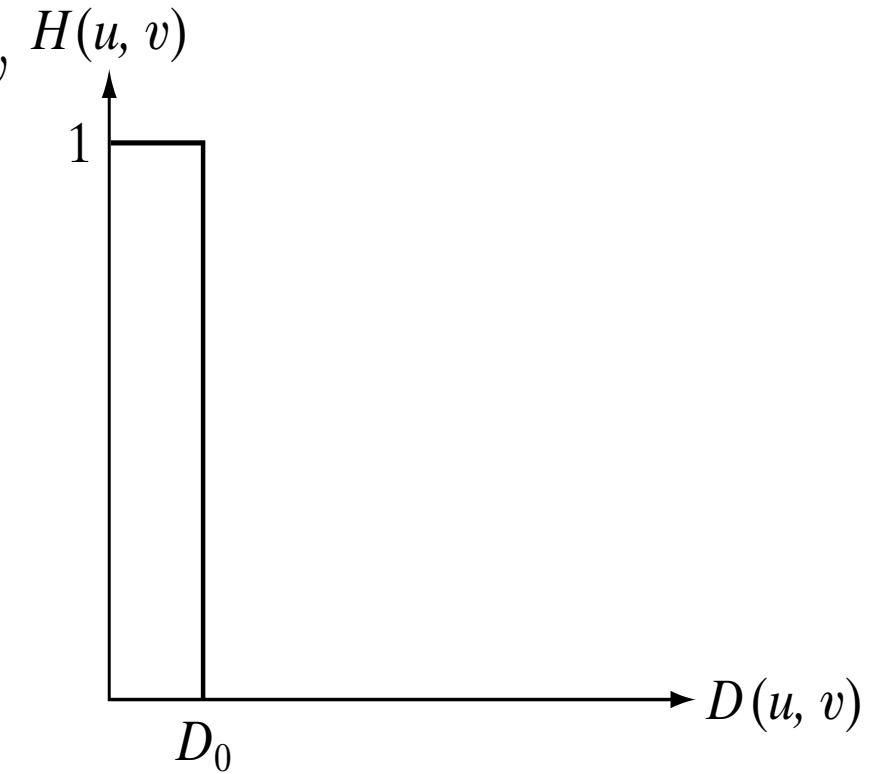
Ideal Lowpass Filtering



Ideal Lowpass Filter



Ideal Lowpass Filter
illustrated as an image



Filter radial cross section.

Definition

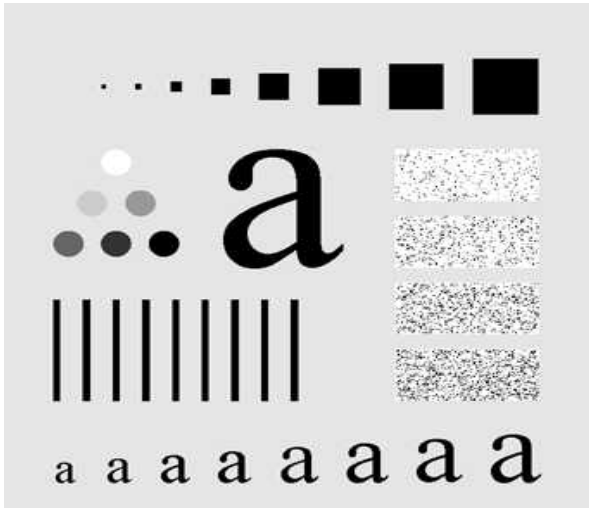
- A 2D lowpass filter that passes without attenuation all frequencies within a circle of radius D_0 from the origin and “cuts off” all frequencies outside this circle is called an **ideal lowpass filter (ILPF)**

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

where D_0 is a positive constant and $D(u, v)$ is the distance between a point (u, v) in the frequency domain and the center of the frequency rectangle

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$

Example



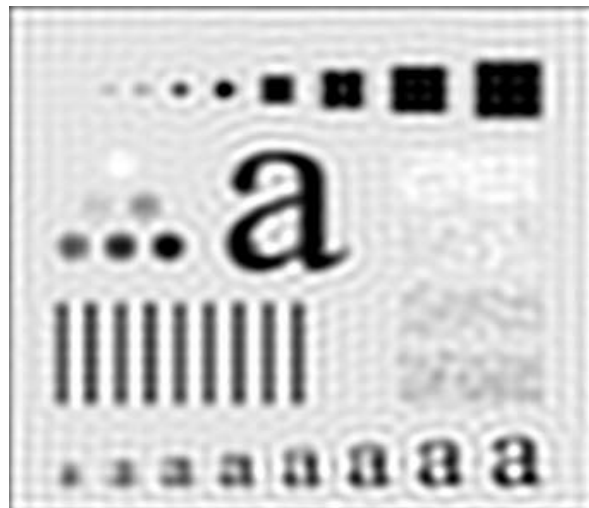
Original Image



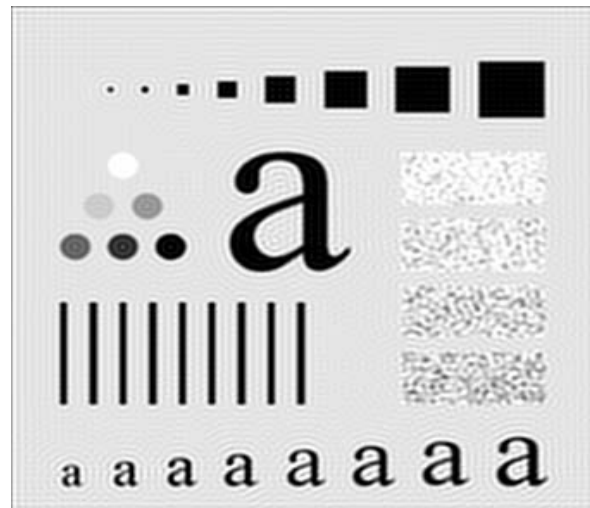
Cut-off frequency 10



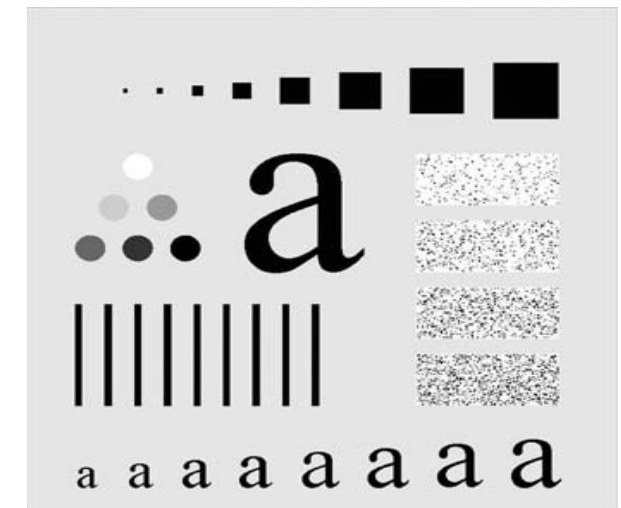
Cut-off frequency 30



Cut-off frequency 60

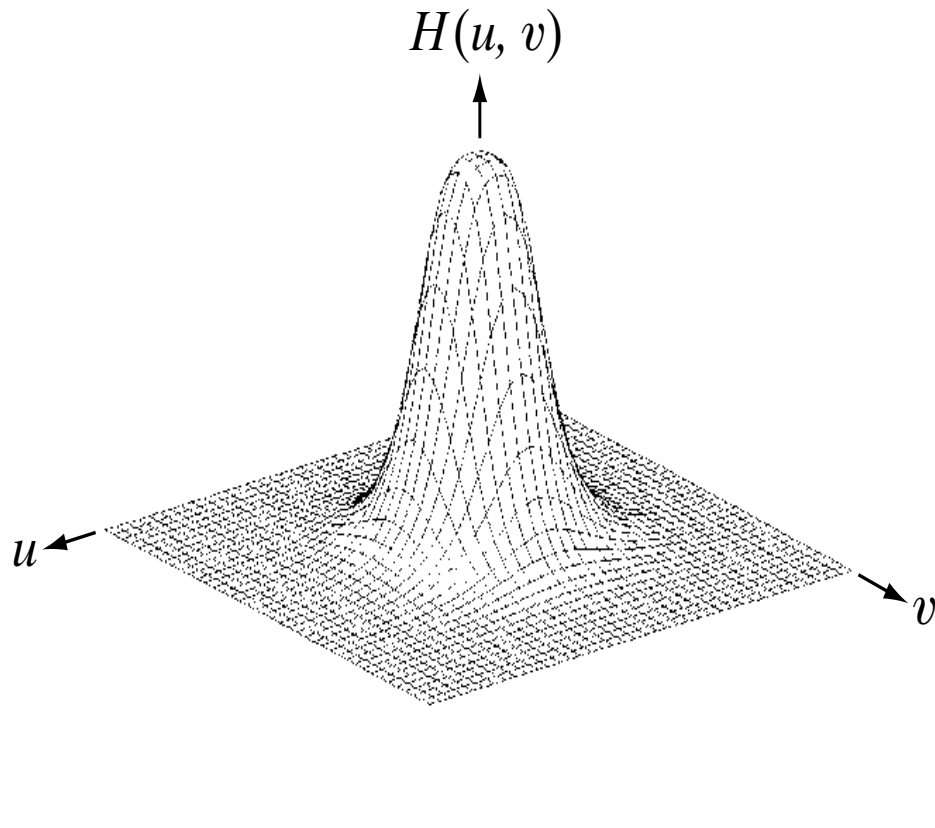


Cut-off frequency 160

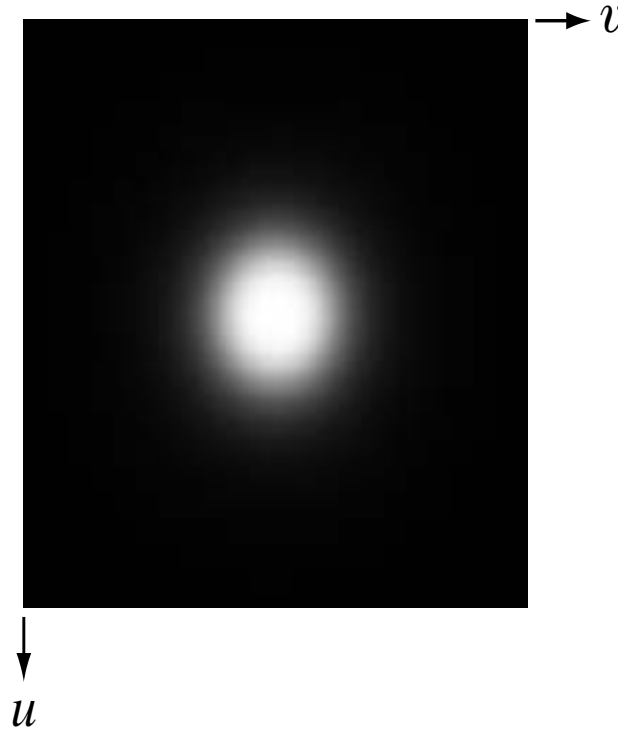


Cut-off frequency 460

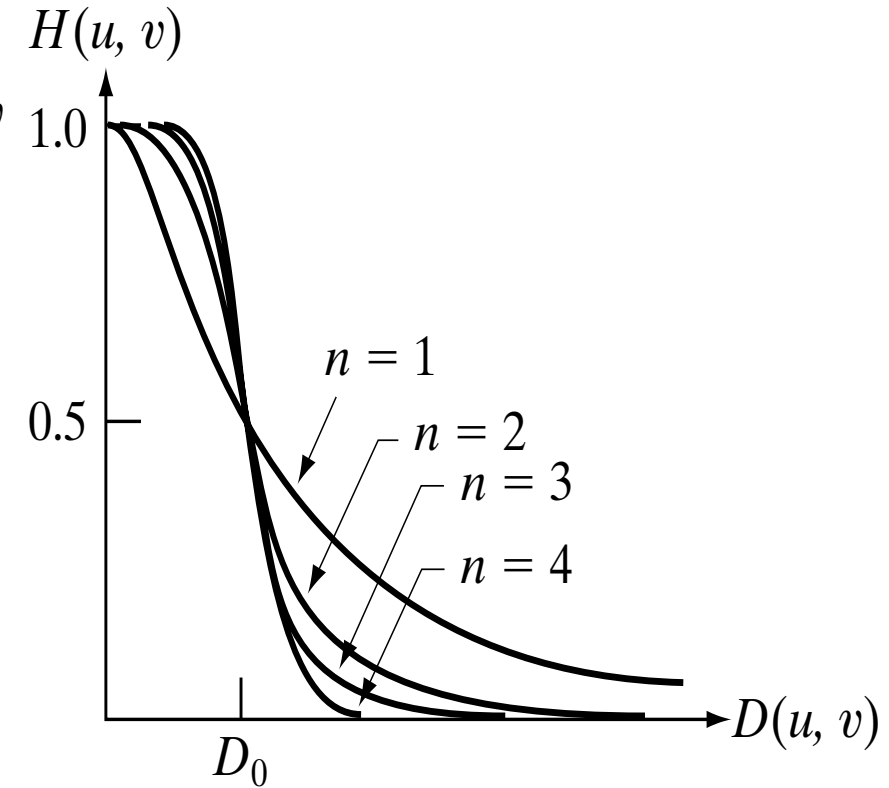
Butterworth Lowpass Filtering



Butterworth Lowpass Filter



Butterworth Lowpass Filter
illustrated as an image



Filter radial cross section.

Definition

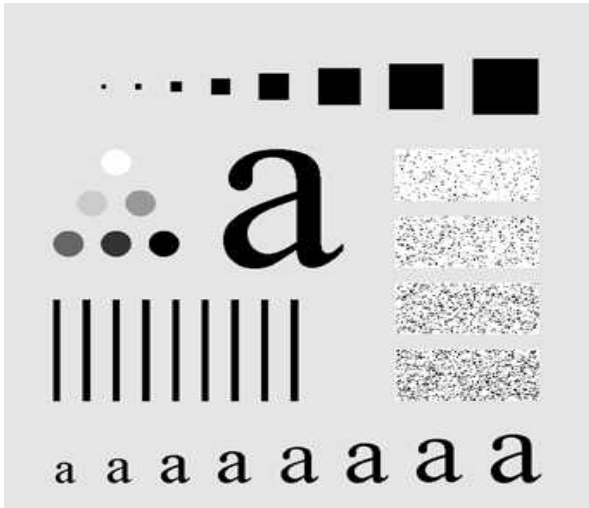
- A transfer function of a **Butterworth lowpass filter (BLPF)** of order n and with cutoff frequency at a distance D_0 from the origin is defined as

$$H(u, v) = \frac{1}{1 + \left[D(u, v) / D_0 \right]^{2n}}$$

where $D(u, v)$ is given by

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$

Example



Original Image



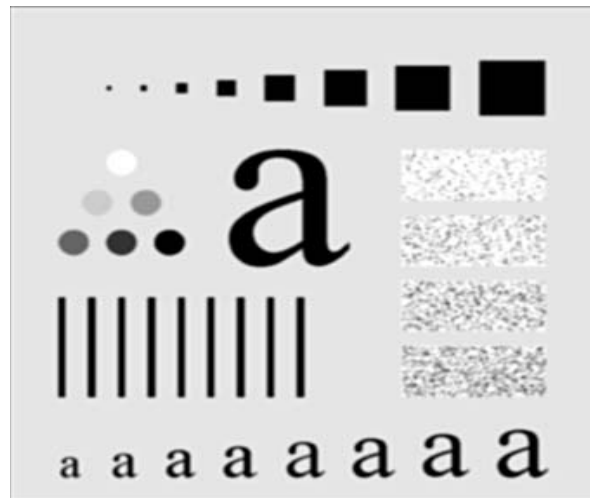
Cut-off frequency 10



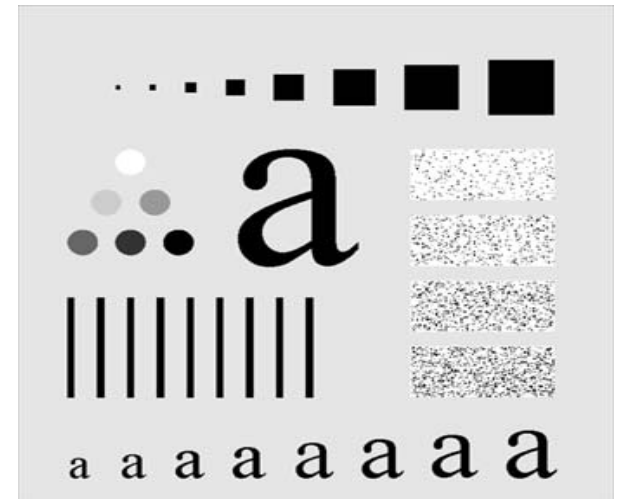
Cut-off frequency 30



Cut-off frequency 60

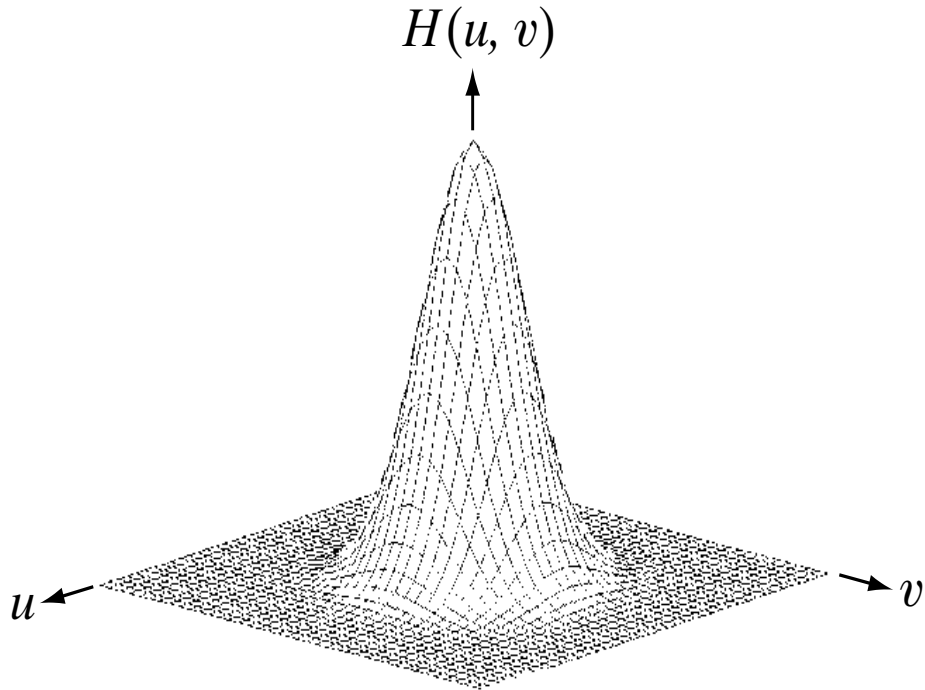


Cut-off frequency 160

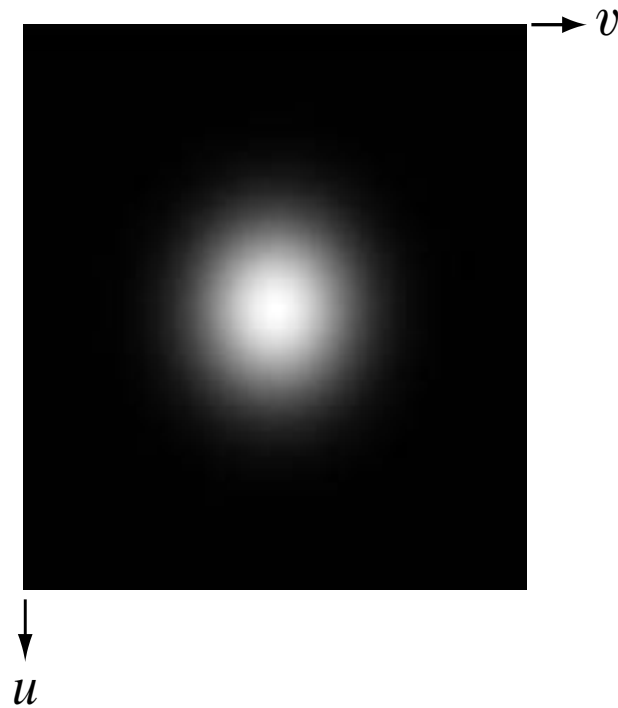


Cut-off frequency 460

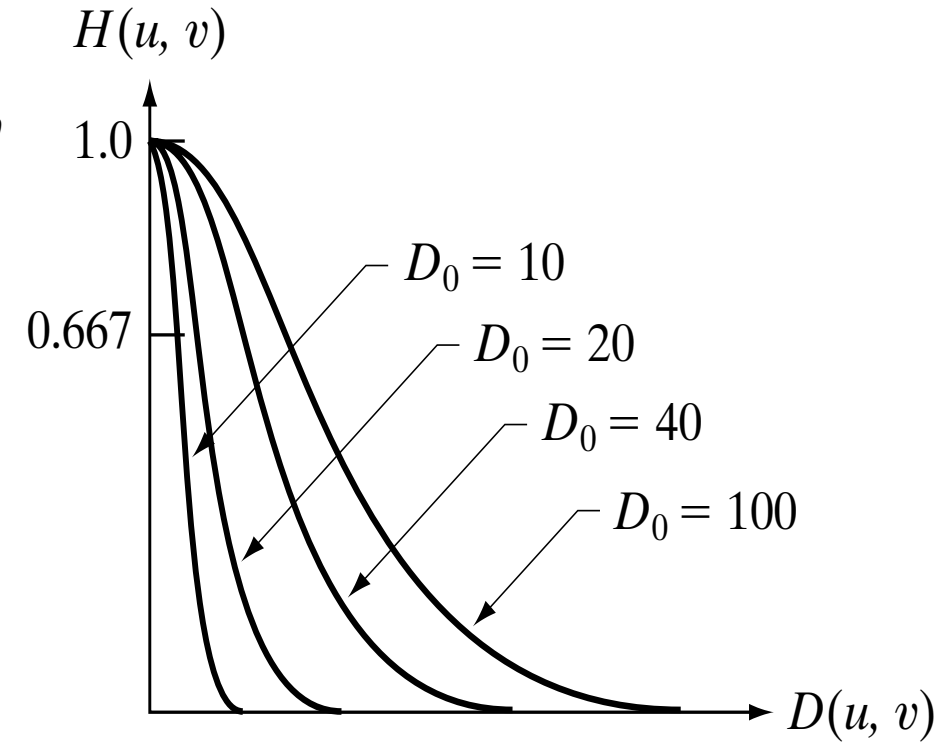
Gaussian Lowpass Filtering



Gaussian Lowpass Filter



Gaussian Lowpass Filter
illustrated as an image



Filter radial cross section.

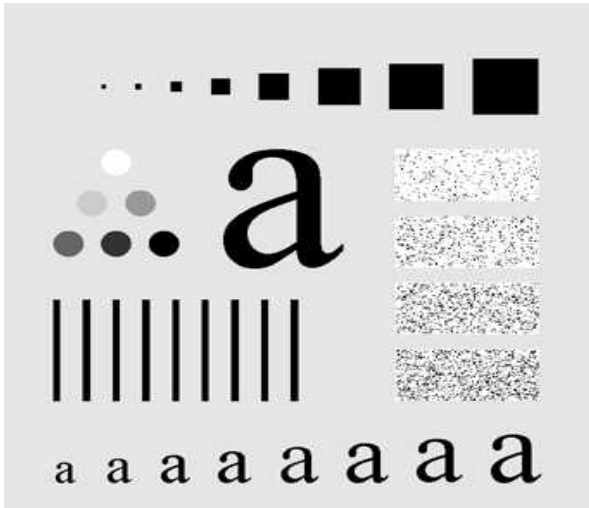
Definition

- A transfer function of a **Gaussian lowpass filter (GLPF)** in two dimensions is given by

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

where D_0 is the cutoff frequency. When $D(u, v) = D_0$ the GLPF is down to 0.607 of its maximum value.

Example



Original Image



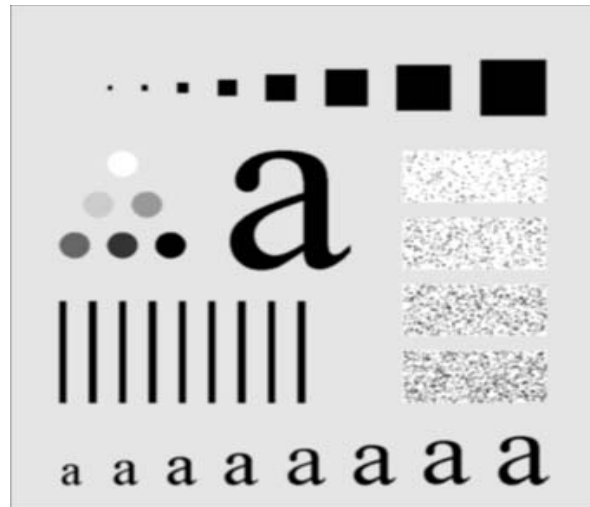
Cut-off frequency 10



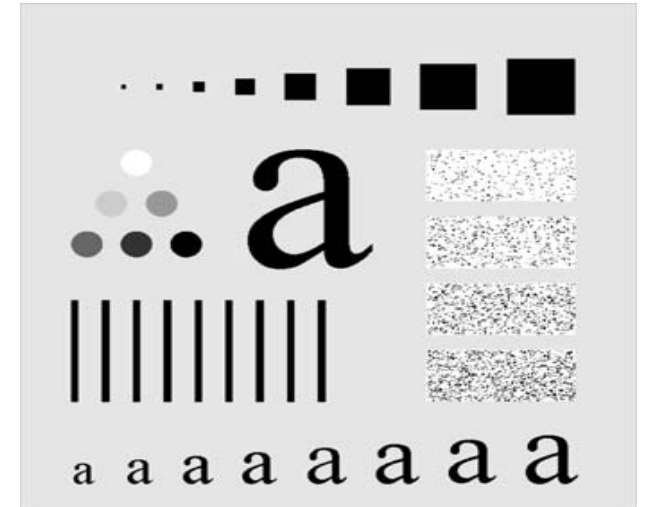
Cut-off frequency 30



Cut-off frequency 60

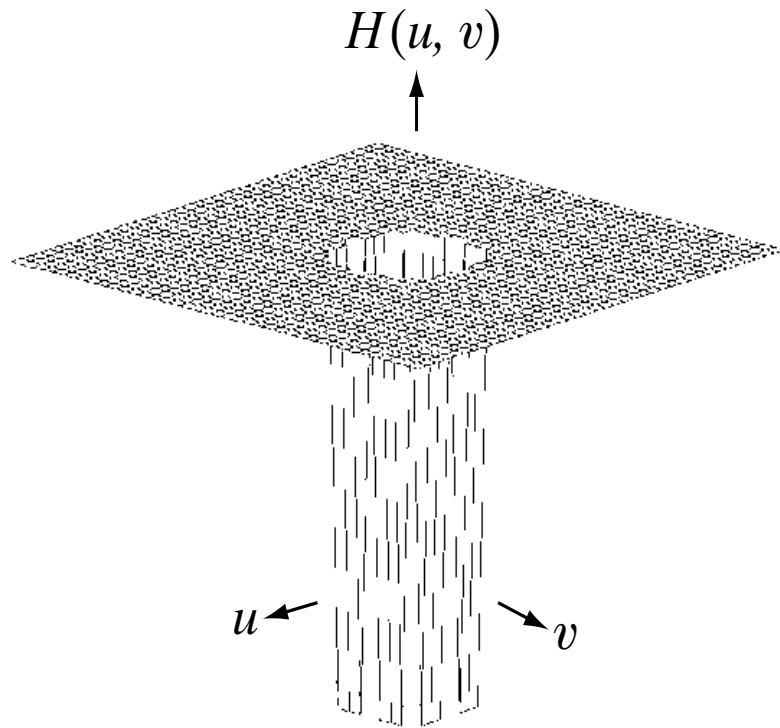


Cut-off frequency 160

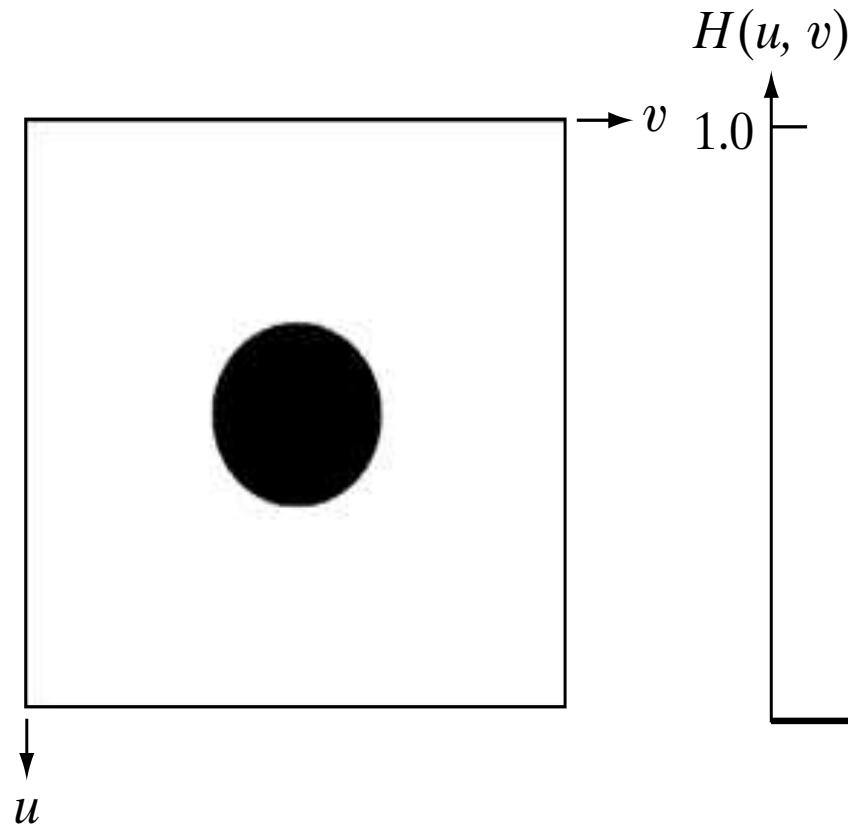


Cut-off frequency 460

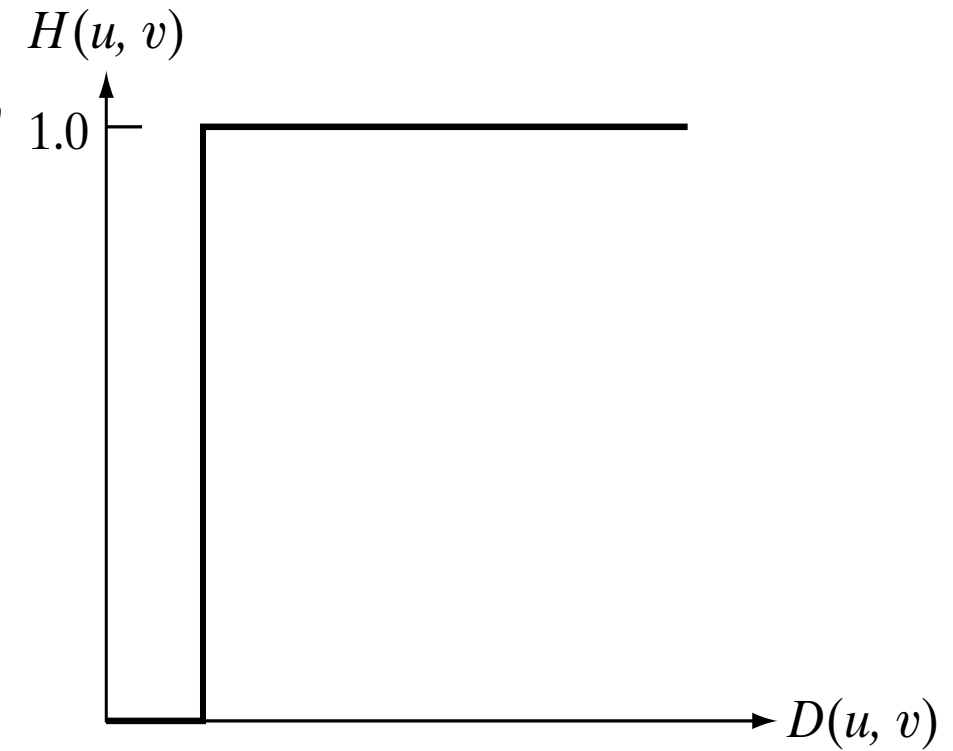
Ideal Highpass Filtering



Ideal Highpass Filter



Ideal Highpass Filter
illustrated as an image



Filter radial cross section.

Definition

- A 2D highpass filter that passes without attenuation all frequencies outside a circle of radius D_0 from the origin and “cuts off” all frequencies inside this circle is called an **ideal highpass filter (IHPF)**

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$

where D_0 is a positive constant and $D(u, v)$ is the distance between a point (u, v) in the frequency domain and the center of the frequency rectangle

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$

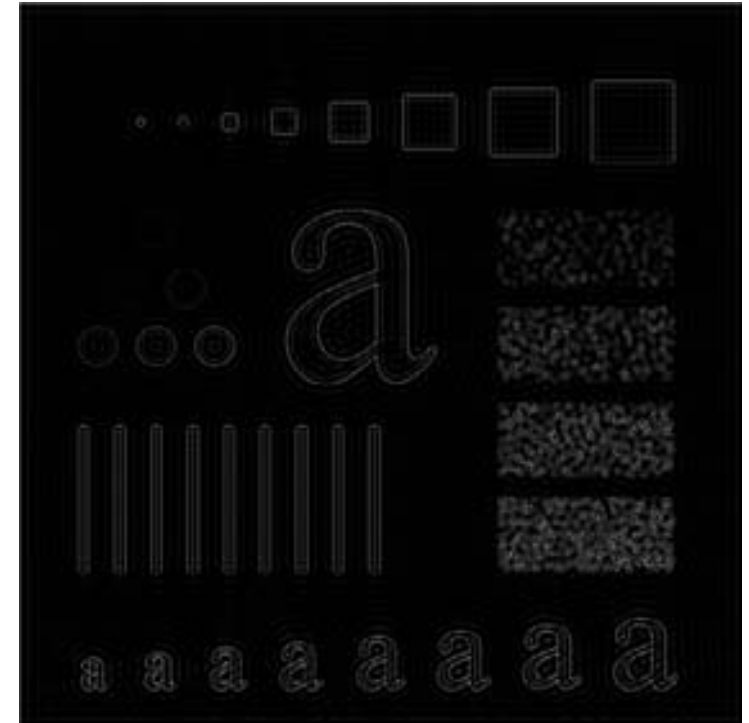
Example



Ideal Highpass Filter with
 $D_0 = 30$

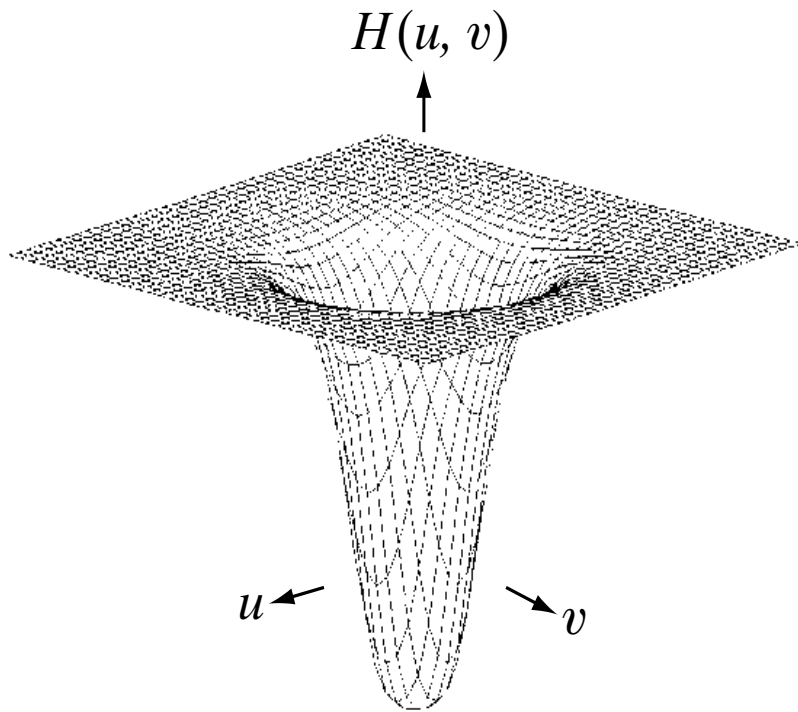


Ideal Highpass Filter with
 $D_0 = 60$

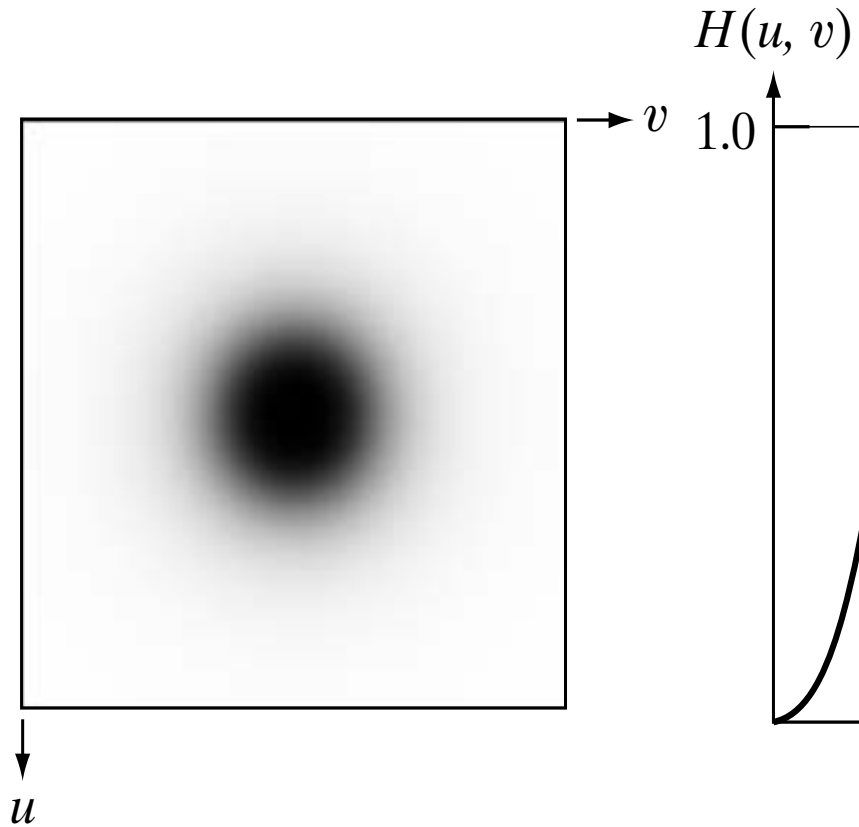


Ideal Highpass Filter with
 $D_0 = 160$

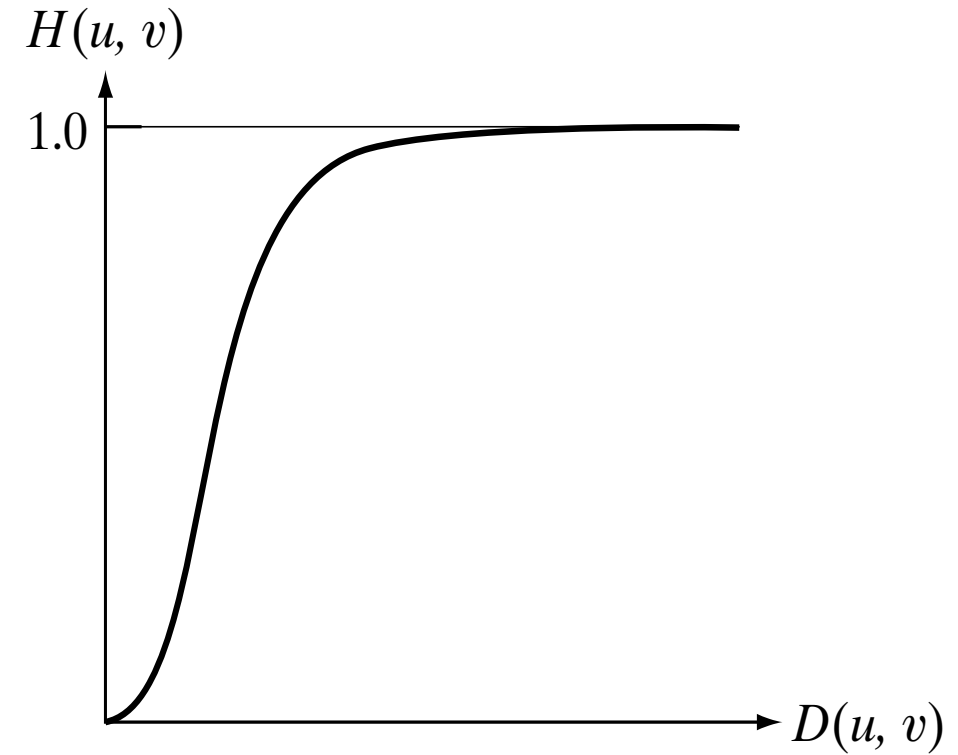
Butterworth Highpass Filtering



Butterworth highpass
Filter



Butterworth highpass
Filter illustrated as an
image



Filter radial cross section.

Definition

- A transfer function of a **Butterworth highpass filter (BHPF)** of order n and with cutoff frequency at a distance D_0 from the origin is defined as

$$H(u, v) = \frac{1}{1 + \left[D_0 / D(u, v) \right]^{2n}}$$

where $D(u, v)$ is given by

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$

Example



Butterworth Highpass
Filter with $D_0 = 30$

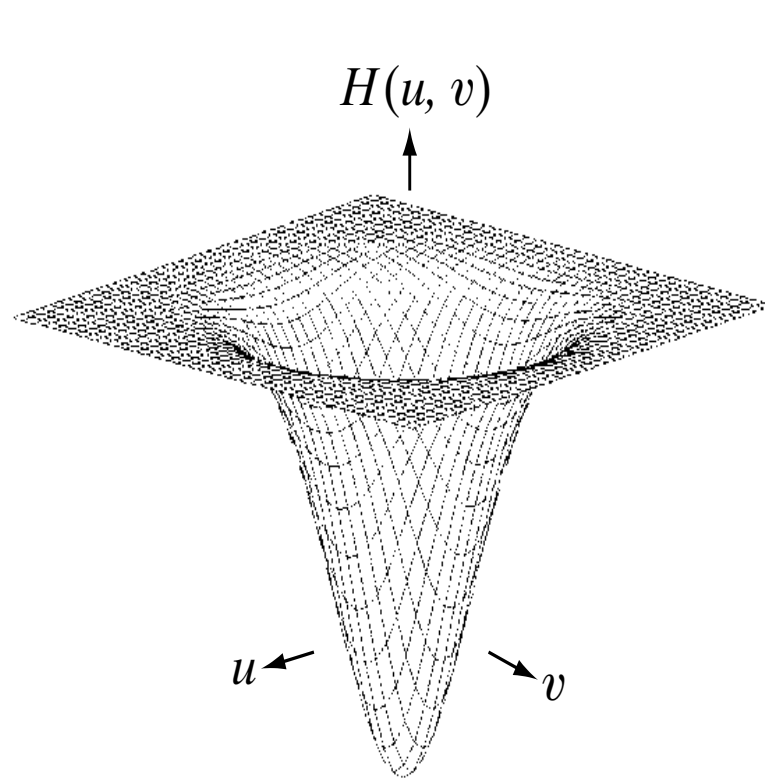


Butterworth Highpass
Filter with $D_0 = 60$

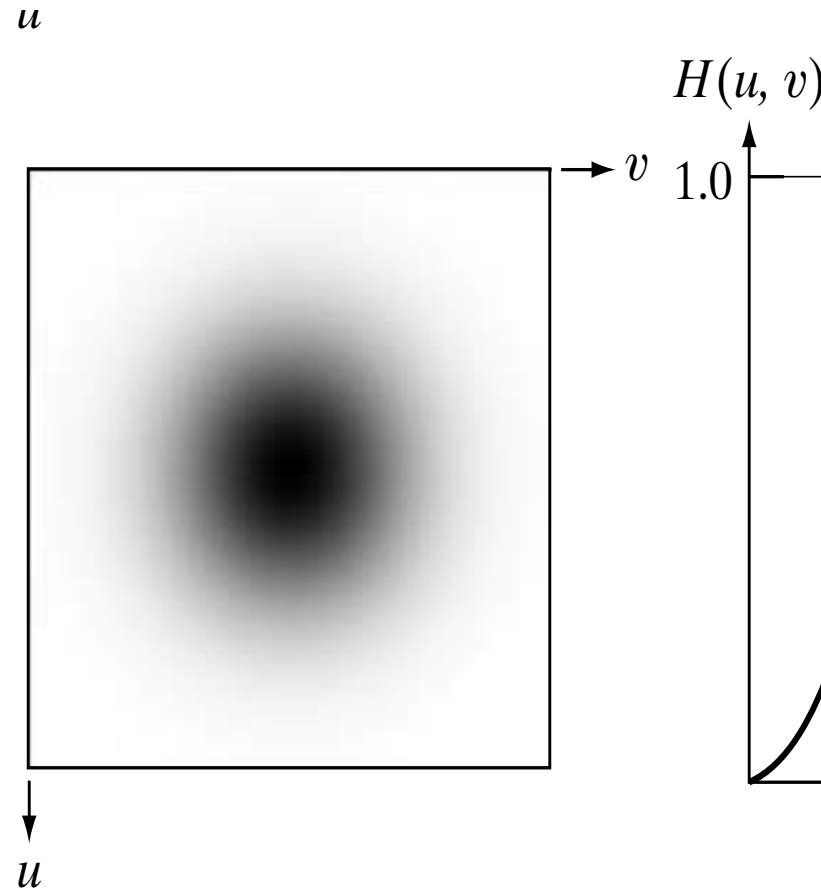


Butterworth Highpass
Filter with $D_0 = 160$

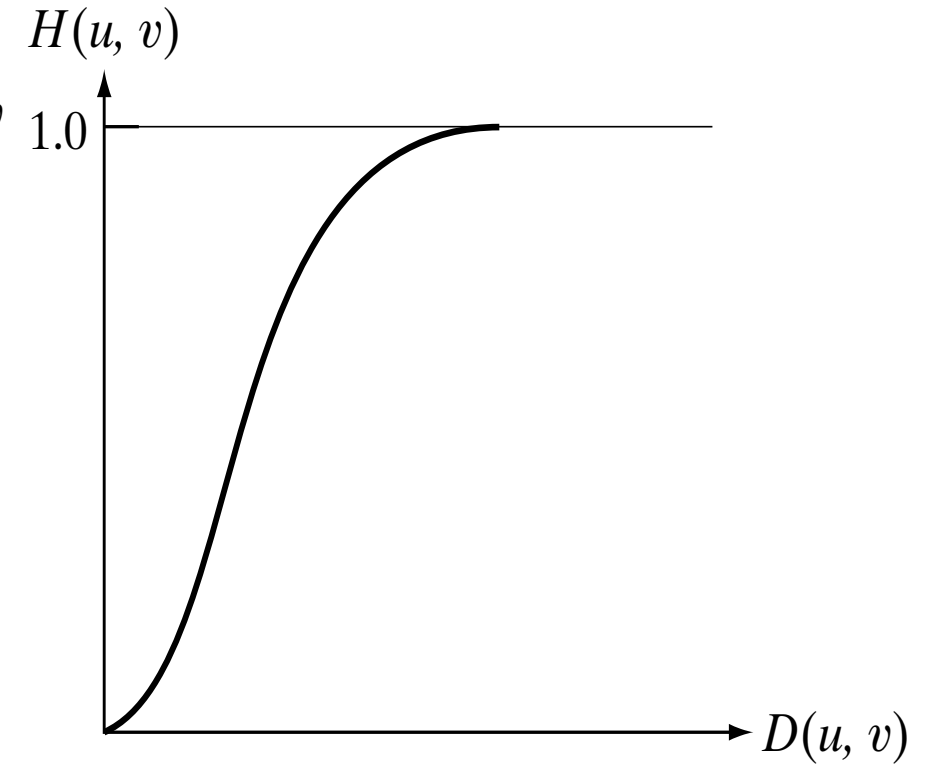
Gaussian Highpass Filtering



Gaussian Highpass Filter



Gaussian Highpass Filter
illustrated as an image



Filter radial cross section.

Example



Gaussian Highpass Filter
with $D_0 = 30$



Gaussian Highpass Filter
with $D_0 = 60$



Gaussian Highpass Filter
with $D_0 = 160$

Definition

- A transfer function of a **Gaussian highpass filter (GHPF)** in two dimensions is given by

$$H(u, v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

where D_0 is the cutoff frequency.

Course Outline

- ✓ Digital Image Fundamentals
- ✓ Image Acquisition
- ✓ Image Transforms
- ✓ Convolution Theorem
- ✓ **Morphological Operations**
- ✓ Multiresolution Analysis
- ✓ Image Coding

Morphological Processing

- ✓ The work ***morphology*** commonly denotes a branch of biology that deals with the form and structure of animals and plants.
- ✓ ***Mathematical Morphology*** is a tool that enables the extraction of image components that are useful in the representation and description of region shapes such as boundaries, skeletons and convex hull.
- ✓ ***Morphological techniques*** might be pre- or post-processing i.e. filtering, thinning and pruning.
- ✓ The input of a morphological method is an image and the output might be either an image or some useful ***attributes*** extracted from the input images.

Reminder

- ✓ A set is a collection of objects, which are the elements of the set.
- ✓ If S is a set and x is an element of S , we write $x \in S$.
- ✓ If x is not an element of S , we write $x \notin S$.
- ✓ A set can have no elements, in which case it is called the **empty set**, denoted by \emptyset .
- ✓ If S contains a finite number of elements, say x_1, x_2, \dots, x_n , we write it as a list of the elements, in braces:

$$S = \{x_1, x_2, \dots, x_n\}$$

Reminder

- ✓ If S contains infinitely many elements x_1, x_2, \dots , which can be enumerated in a list, we say that S **countably infinite** and we write

$$S = \{x_1, x_2, \dots\}$$

- ✓ Alternatively, we can consider the set of all x that have a certain property P and denote it by

$$\{x \mid x \text{ satisfies } P\}$$

- ✓ A set of all scalars x in the interval $[0,1]$ can be written as $\{x \mid 0 \leq x \leq 1\}$. The elements of this set cannot be written down in a list; such a set is said to be **uncountable**.

Set operations

- ✓ The **complement** of a set S with respect to the universe Ω , is the set of all elements of Ω that do not belong to S and is denoted as S^c .

$$\{x \in \Omega \mid x \notin S\}$$

- ✓ The **union** of two sets S and T is the set of all elements that belong to S or T (or both), and is denoted by $S \cup T$.

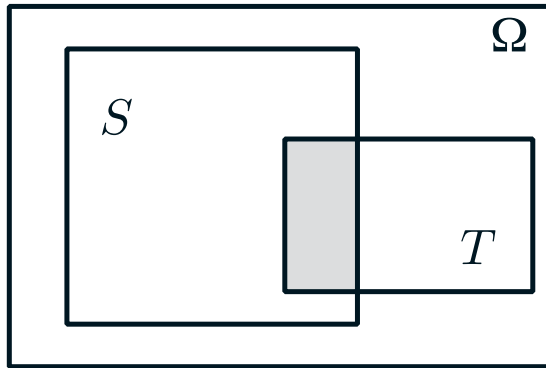
$$S \cup T \{x \in S \text{ or } x \in T\}$$

- ✓ The **intersection** of two sets S and T is the set of all elements that belong to both S and T , and it is denoted by $S \cap T$.

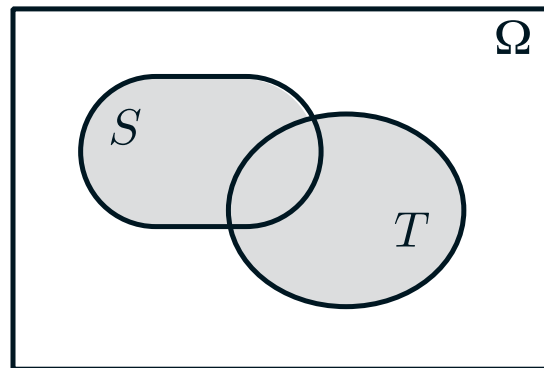
$$S \cap T \{x \in S \text{ and } x \in T\}$$

- ✓ Two sets are said to be disjoint if their intersection is empty. In other words, two sets are called **disjoint** if no two of them have a common element.

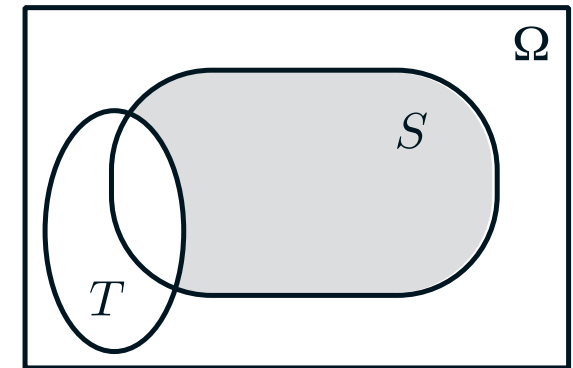
Examples



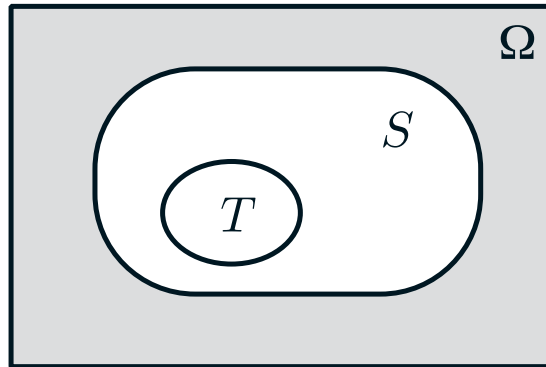
$$S \cap T$$



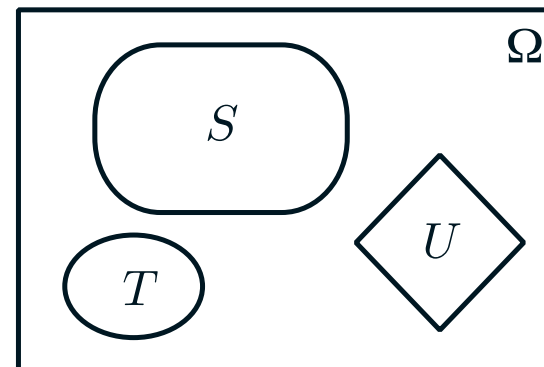
$$S \cup T$$



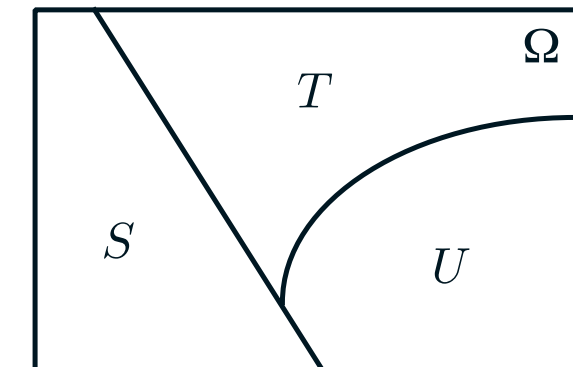
$$S \cup T^c$$



$T \subset S$ and the shaded region is
The complement of S



T, S, U are disjoint



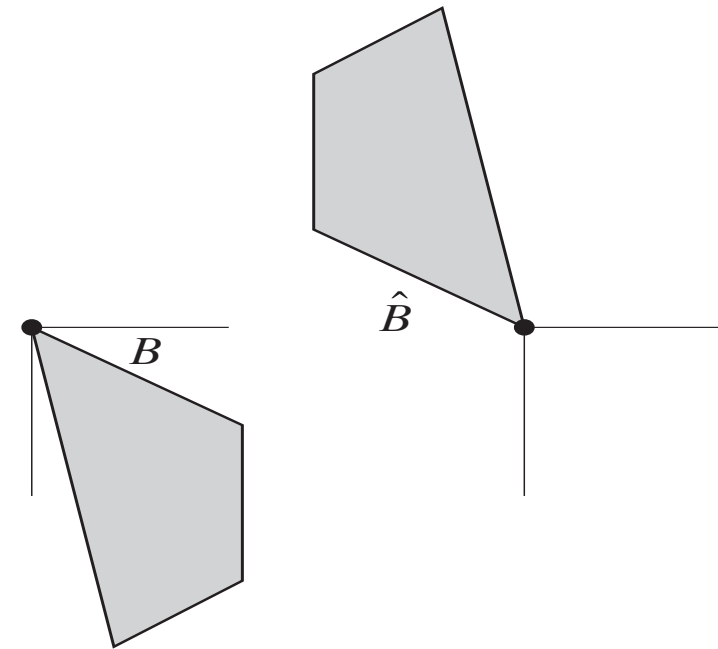
T, S, U are the partition of Ω

Basic Set Definitions

- ✓ **Reflection** of a set B , denoted as \hat{B} , is defined as

$$\hat{B} = \{w | w = -b, \text{ for } b \in B\}.$$

If B is the set of pixels (2D points) representing an object in an image, then \hat{B} is simply the set of points in B whose (x, y) coordinates have been replaced by $(-x, -y)$.

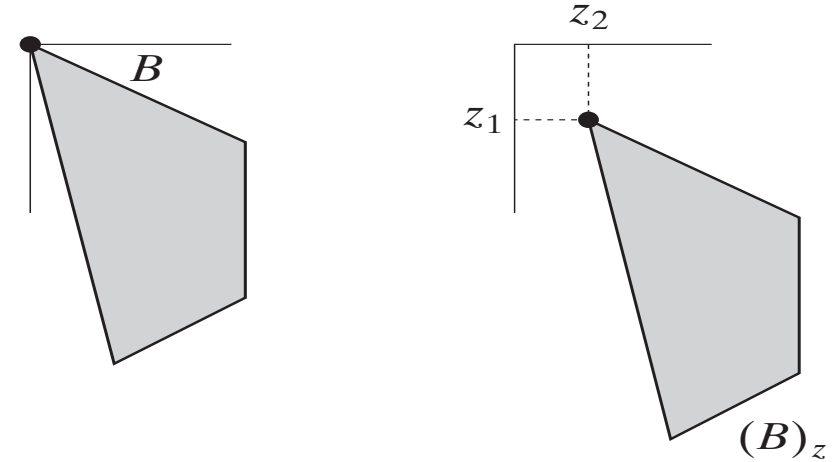


Basic Set Definitions

- ✓ **Translation** of a set B , denoted as $(B)_z$, is defined as

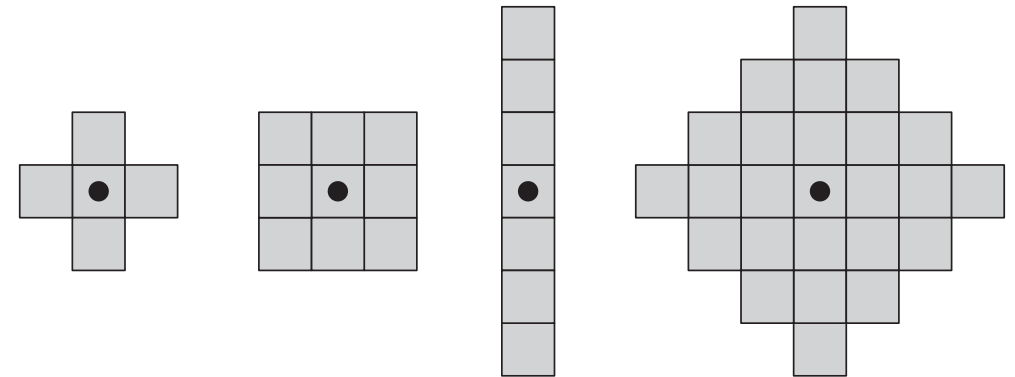
$$(B)_z = \{w \mid w = b + z, \text{ for } b \in B\}.$$

- ✓ If B is the set of pixels (2D points) representing an object in an image, then $(B)_z$ is simply the set of points in B whose (x, y) coordinates have been replaced by $(x + z_1, y + z_2)$.



Structuring Elements (SEs)

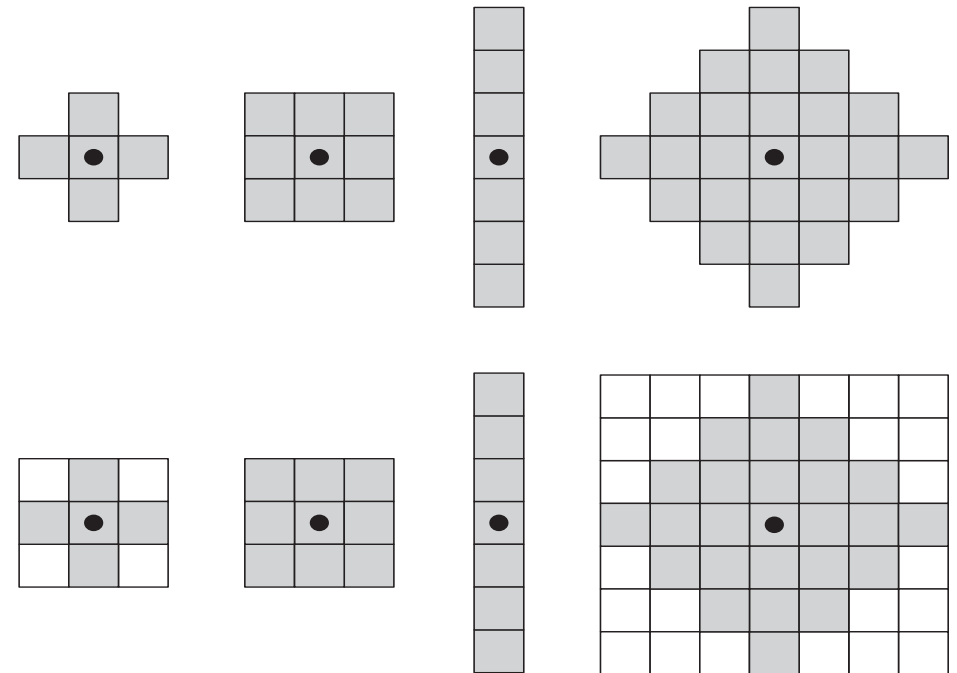
- ✓ **Set reflection and translation** are employed extensively in morphology to formulate operations based on structural elements (SEs).
- ✓ Structuring Elements are small sets or sub-images used to probe an image under study for **properties of interest**.
- ✓ Structural Elements might have several different **shapes**, such as square, line, diamond, ball, disk, ect.
- ✓ The **origin** of a structuring element also must be specified (black dot).



Examples of structuring elements.

Structuring Elements (SEs)

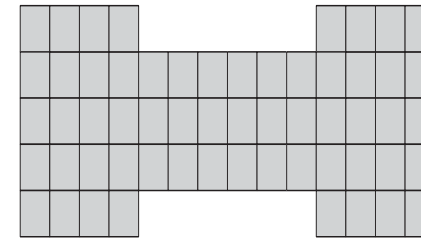
- ✓ In image processing it is required that structuring elements be rectangular arrays.
- ✓ This is accomplished by appending the smallest possible number of background.



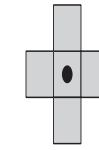
Structuring elements converted to rectangular arrays.

Example

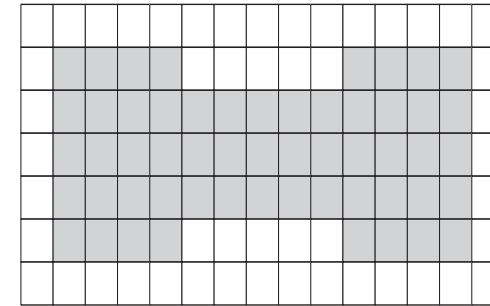
- ✓ Suppose that we are interested in creating a new set by running a set B over a set A so that the origin of B visits every element of A .
- ✓ At each location of the origin of B , if B is completely in A then the pixel is shaded otherwise it is not shaded.
- ✓ The result is that the boundary of the initial set is **eroded**.



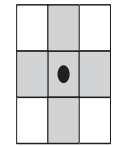
Step 1: A set



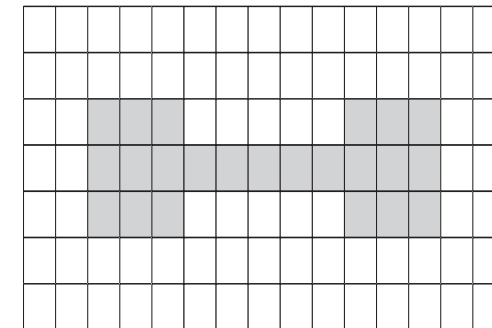
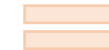
Step 2:
Structuring
Element



Step 3: The set padded with
background elements.



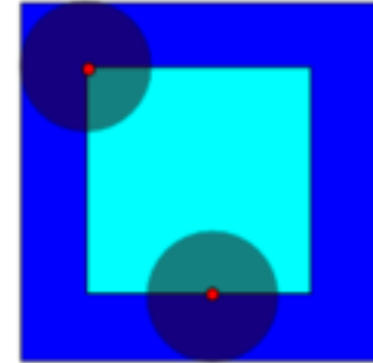
Step 4: The structuring
element as a
rectangular array.



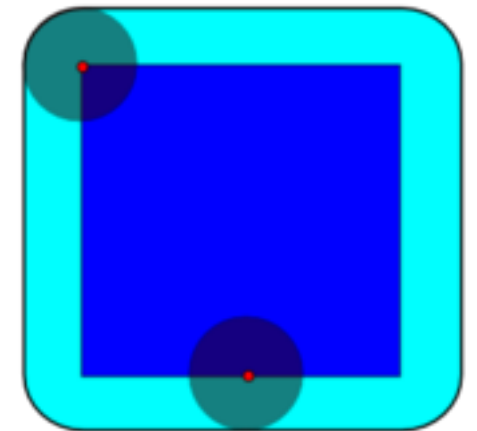
Step 5: Set proposed by the structuring
element.

Erosion and Dilation

- ✓ The most fundamental morphological operations are called *erosion* and *dilation*.
- ✓ **Erosion:** is an operation that usually uses a structuring element for probing and reducing the shapes contained in the input image.
- ✓ **Dilation:** is an operation that usually uses a structuring element for probing and expanding the shapes contained in the input image.
- ✓ Erosion \neq Dilation



Erosion of the dark-blue square by a disk.



Dilation of the dark-blue square by a disk.

Erosion

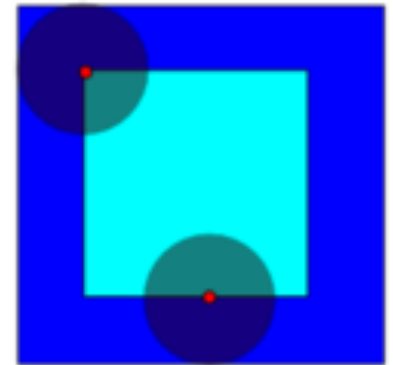
- ✓ If we consider two sets A and B in \mathbb{Z}^2 , the erosion of A by B , denoted $A \ominus B$ is defined as

$$A \ominus B = \{z | (B)_z \subseteq A\}$$

- ✓ The above equation indicates that the erosion of A by B is the set of all points z such that B , translated by z , is contained in A .
- ✓ The erosion can be expressed also in the following form:

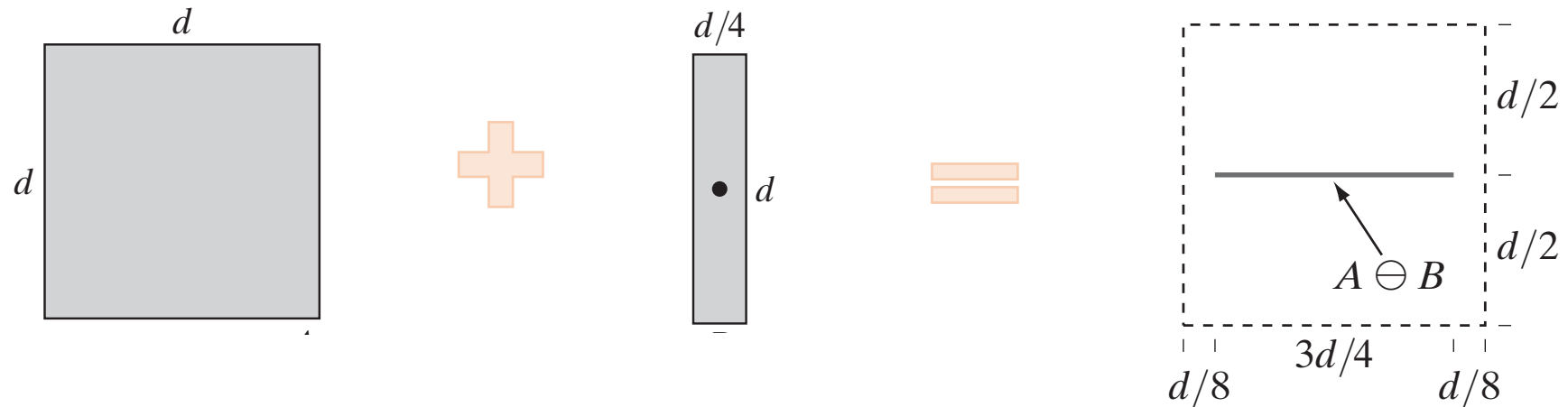
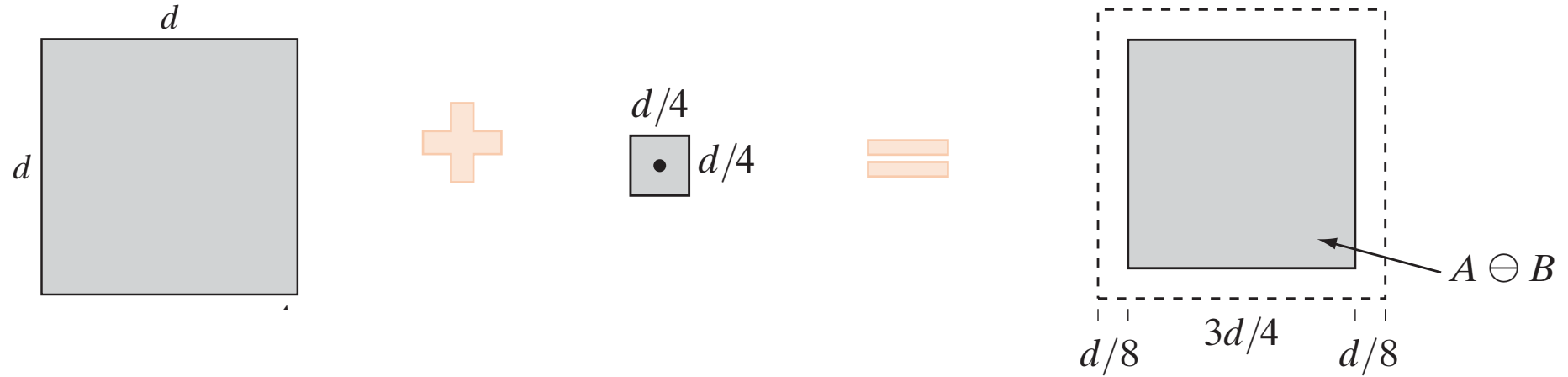
$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\}$$

where A^c is the complement of A and \emptyset is the empty set.



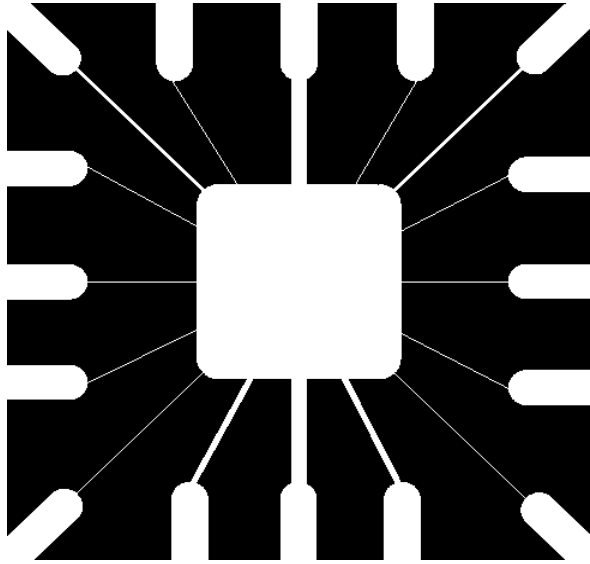
Erosion of the dark-blue square by a disk.

Examples of Erosion

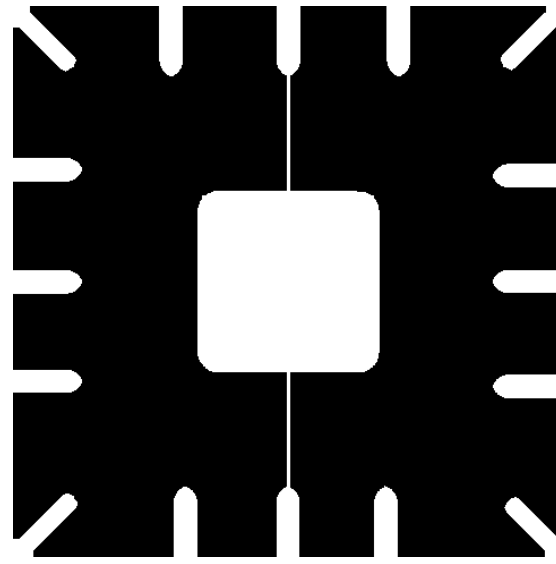


Examples of Erosion

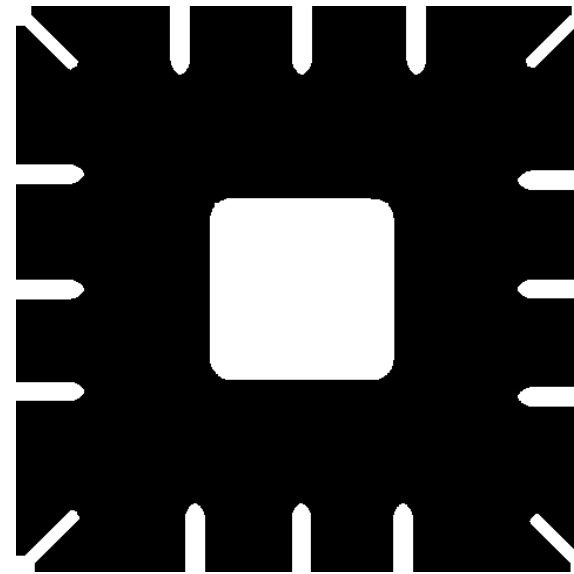
The components of the square structuring element are all 1s!!!



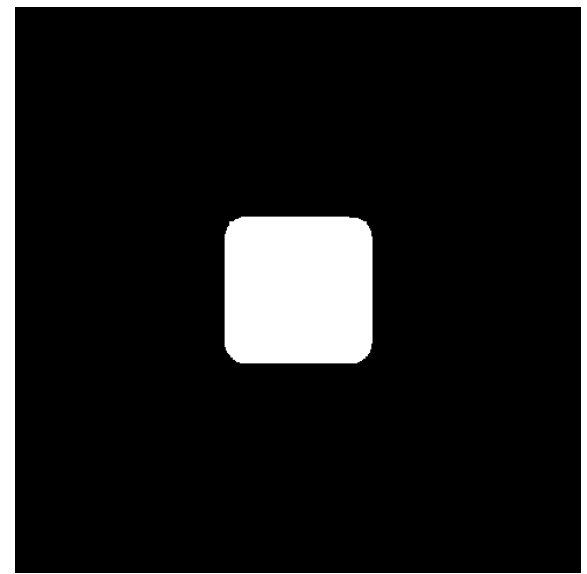
Input Image.



Erosion using a 11x11 square.



Erosion using a 15x15 square.



Erosion using a 45x45 square.

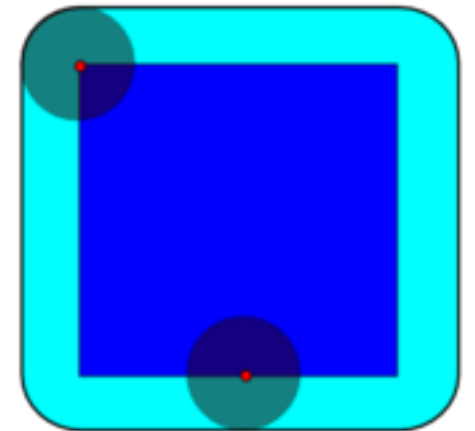
Dilation

- ✓ If we consider to sets A and B in \mathbb{Z}^2 , the dilation of A by B , denoted $A \oplus B$ is defined as

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

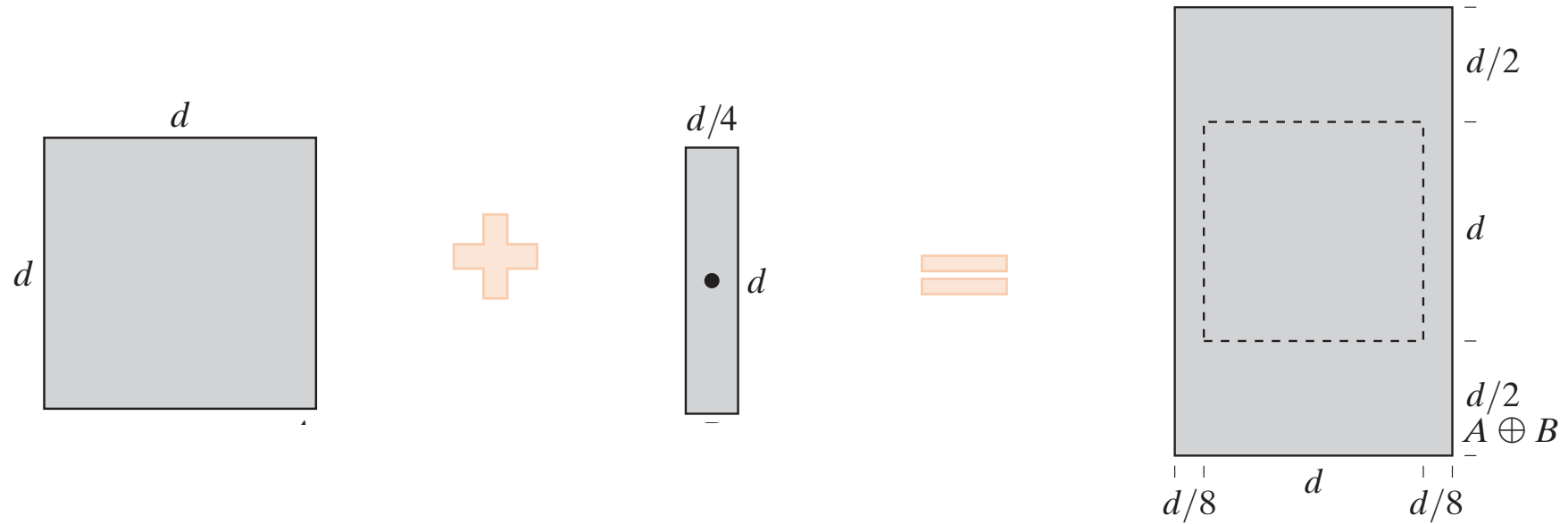
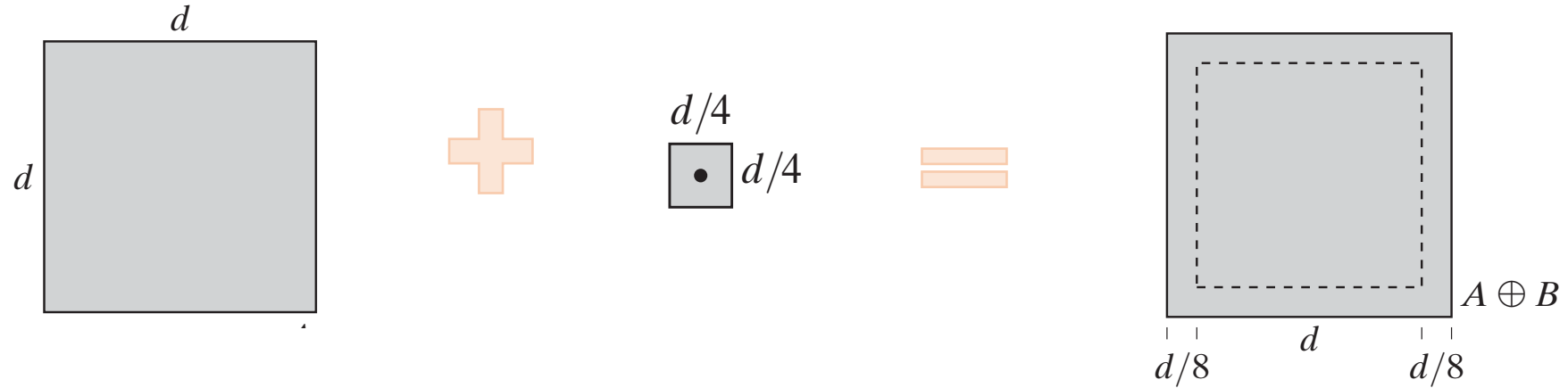
- ✓ The above equation indicates that the dilation of A by B is the set of all points z such that \hat{B} and A overlap by at least for one element.
- ✓ The dilation can be expressed also in the following form:

$$A \oplus B = \{z | [(\hat{B})_z \cap A] \subseteq A\}$$



Dilation of the dark-blue square by a disk.

Examples of Dilation

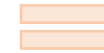


Examples of Dilation

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



0	1	0
1	1	1
0	1	0



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Duality

- ✓ Erosion and dilation are **duals** of each other with respect to set complementation and reflection. That is

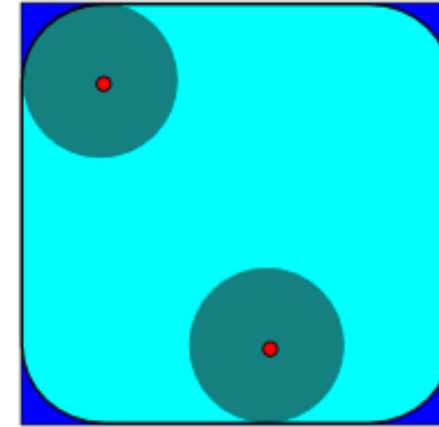
$$(A \ominus B)^c = A^c \oplus \hat{B}$$

$$(A \oplus B)^c = A^c \ominus \hat{B}$$

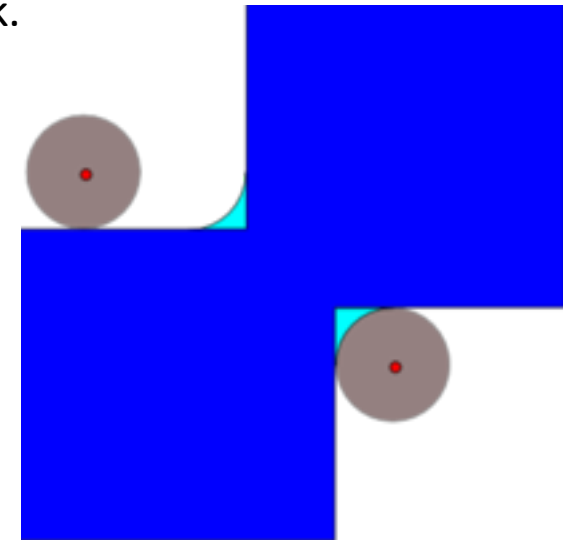
- ✓ Erosion of A by B is the complement of the dilation of A^c by \hat{B} , and vice versa.
- ✓ The duality is a very useful property particularly when the structuring element is **symmetric** with respect to its origin, so that $\hat{B} = B$.

Opening and Closing

- ✓ **Opening:** is an operation that generally smoothes the contour of an object, breaks narrow isthmuses and eliminated thin protrusions.
- ✓ **Closing:** is an operation that also tends to smooth sections of contours, but as apposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminated small holes and fills gaps in the contour.



Opening of the dark-blue square by a disk.



Closing of the dark-blue square by a disk.

Opening

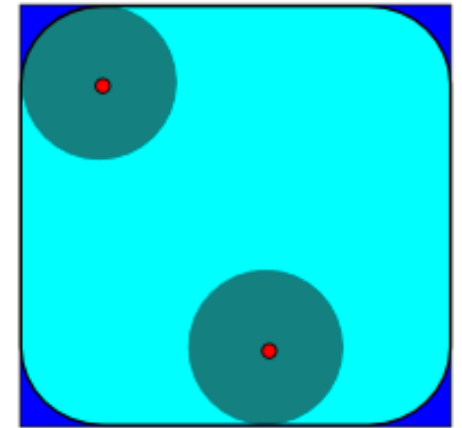
- ✓ The opening A by B is the erosion of A by B , followed by a dilation of the result by B

$$A \circ B = (A \ominus B) \oplus B$$

- ✓ The opening can be also expressed as a fitting process such as

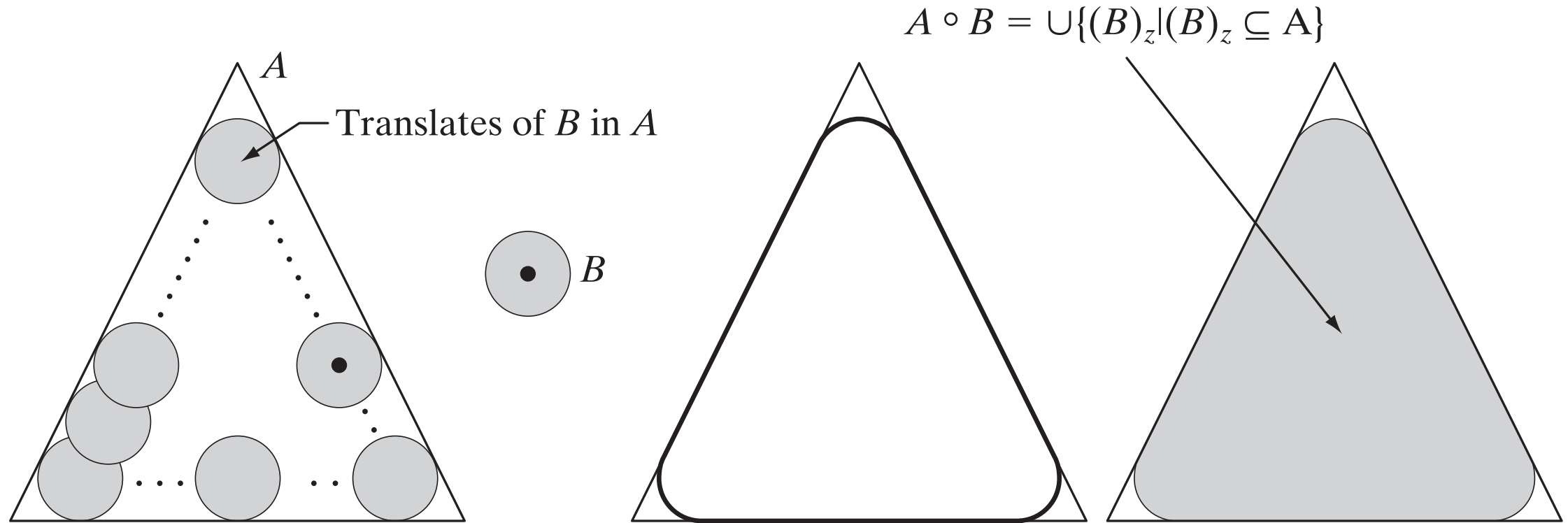
$$A \circ B = \bigcup \{ (B)_z \mid (B)_z \subseteq A \}$$

where $\bigcup \{.\}$ denoted the union of all the sets inside the braces.



Opening of the dark-blue square by a disk.

Examples of Opening



Closing

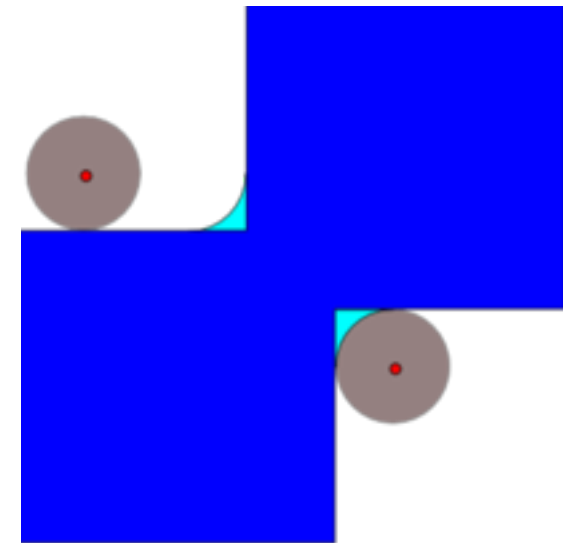
- ✓ The closing of set A by the structuring element B is the dilation of A by B , followed by an erosion of the result by B

$$A \cdot B = (A \oplus B) \ominus B$$

- ✓ The closing is dual of opening thus a point w is an element of closing if and only if

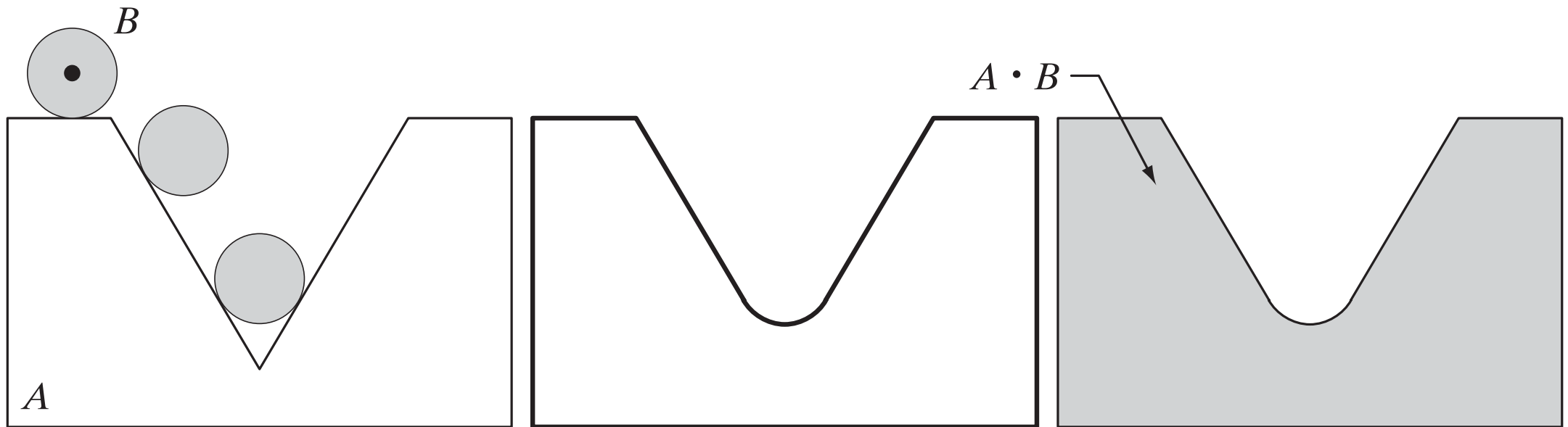
$$A \cdot B = (B)_z \cap A \neq \emptyset$$

where $\cup\{.\}$ denoted the union of all the sets inside the braces.



Closing of the dark-blue square by a disk.

Examples of Opening



Morphological Operations vs Spatial Transforms

- ✓ Morphological operations can be used to construct filters similar in concept to the spatial filters related to:
 - ✓ Image denoising
 - ✓ Edge detection
- ✓ **Noise** and its effects can be eliminated if we apply for example the opening followed by the closing operation.
- ✓ **Edges** can be detected if we apply the dilation operation, then we apply the erosion operation and then we compute the difference of these to results.

Example: Image denoising (Step 1)



1	1	1
1	1	1
1	1	1

$$A \oplus B$$




Example: Image denoising (Step 2)



1	1	1
1	1	1
1	1	1

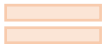
$$(A \ominus B) \oplus B = A \circ B$$



Example: Image denoising (Step 3)



1	1	1
1	1	1
1	1	1

$$\widehat{(A \circ B)} \oplus B$$




Example: Image denoising (Step 4)



1	1	1
1	1	1
1	1	1

$$[(A \circ B) \oplus B] \ominus B = (A \circ B) \cdot B$$



Course Outline

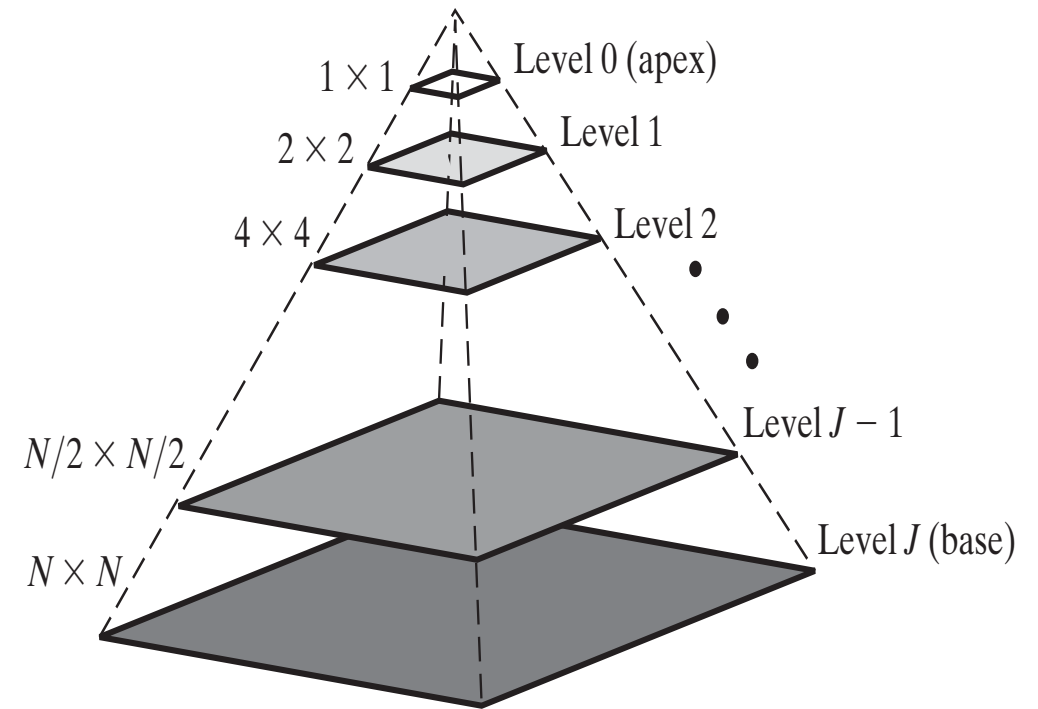
- ✓ Digital Image Fundamentals
- ✓ Image Acquisition
- ✓ Image Transforms
- ✓ Convolution Theorem
- ✓ Morphological Operations
- ✓ Multiresolution Analysis
- ✓ Image Coding

Multiresolution Analysis

- In 1950s a novel transformation, called **wavelet transform**, was introduced as an alternative of the Fourier transform.
- Unlike the Fourier Transform, whose basis functions are sinusoids, wavelet transforms are based on small waves, called **wavelets**, of varying frequency and limited duration.
- In 1987, wavelets were first shown to be the foundation of a powerful new approach to signal processing and analysis, called **multiresolution theory** (Mallat 1987).
- **Multiresolution Advantages:** Features that might be undetected at one resolution may be easy to detect at another.

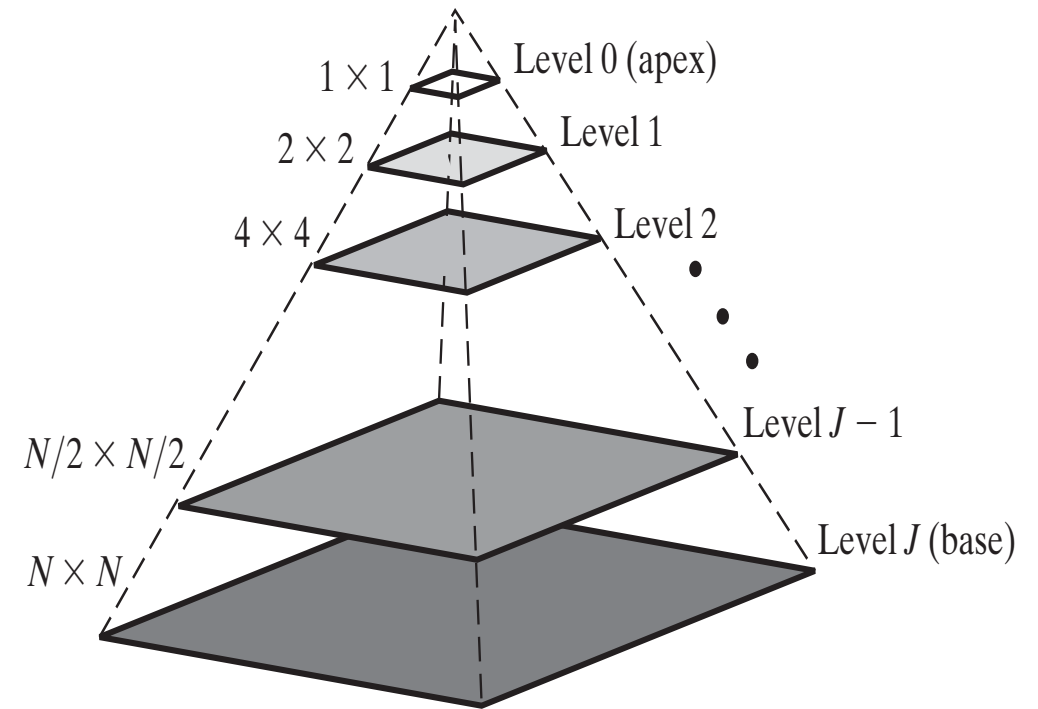
Image Pyramids (Burt and Adelson, 1983)

- Pyramids are powerful, yet conceptually simple structure for representing images at more than one resolution.
- Base level J is of size $2^J \times 2^J$ or $N \times N$
- Top level 0 is of size $2^0 \times 2^0$ or 1×1 .
- The pyramid is composed of $J + 1$ resolution levels from $2^J \times 2^J$ to $2^0 \times 2^0$.

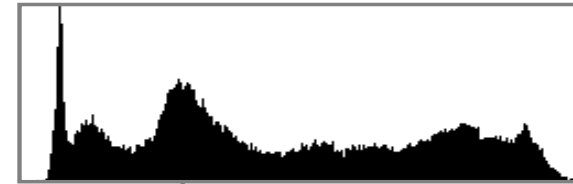


Laplacian Pyramid Algorithm

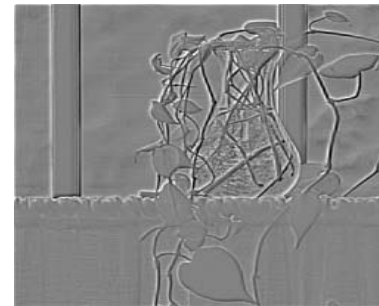
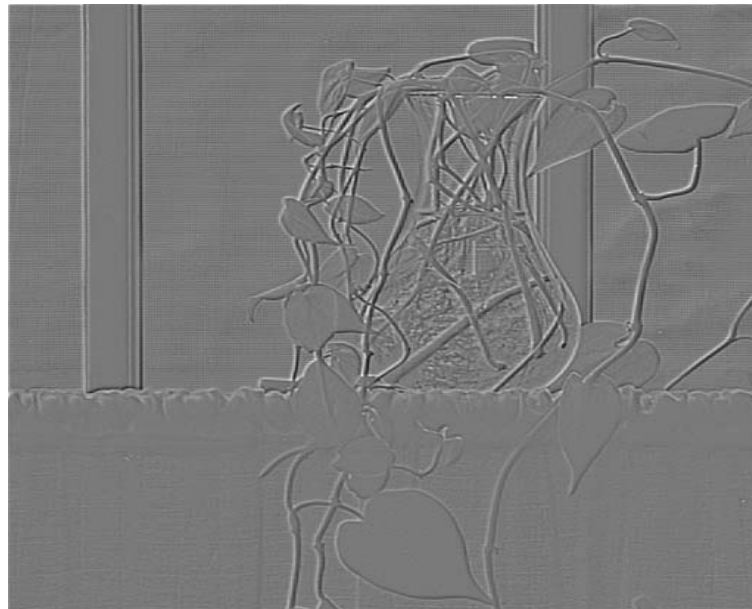
- **Step 1:** Filter and downsample by a factor of 2 the level J input image.
- **Step 2:** Upsample and filter the generated approximation in step 1 generating an image of the same dimensions as the level J input image.
- **Step 3:** Compute the difference between the prediction image of step 2 and the input to step 1.



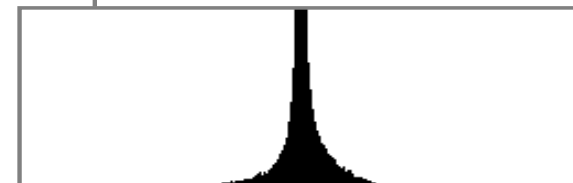
Example



Step 1



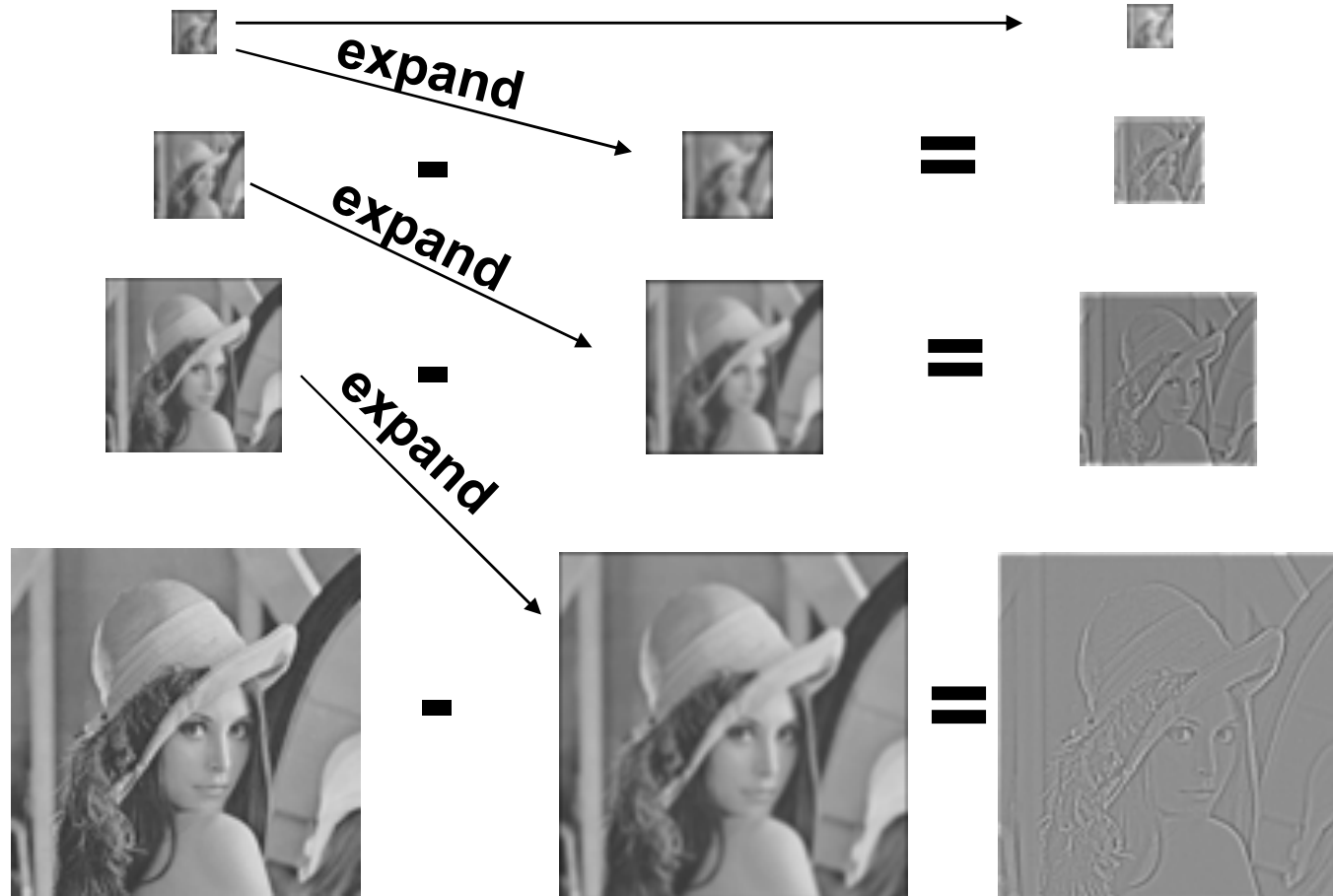
Step 2



Example

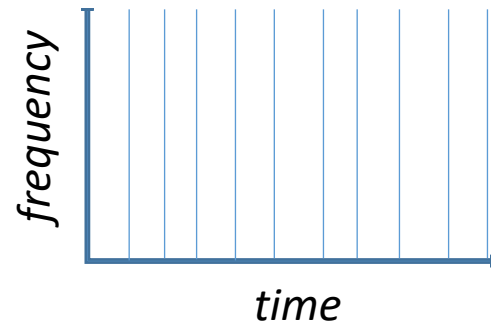
**Gaussian
Pyramid**

**Laplacian
Pyramid**

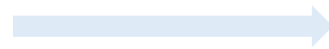


Why Wavelets are important?

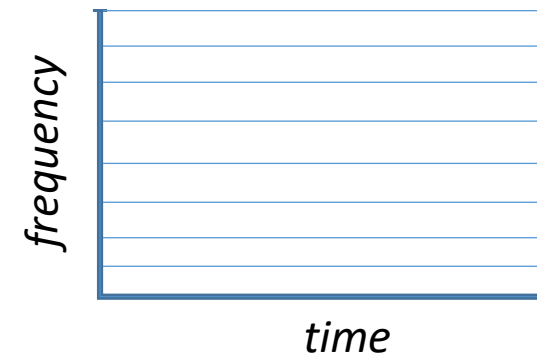
Time-series



Fourier Transform



Fourier of Time-series



Time-series

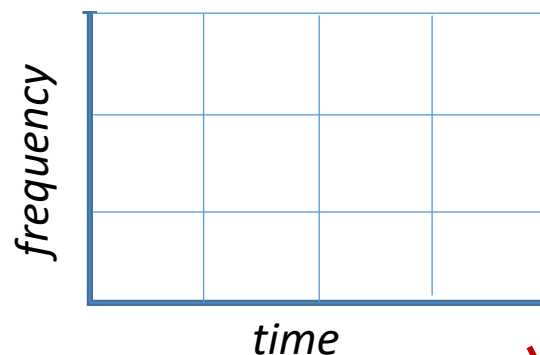
- High temporal resolution
- Uncertainty in the frequency

Fourier Transform

- High frequency resolution
- Uncertainty of the exact time

Why Wavelets are important?

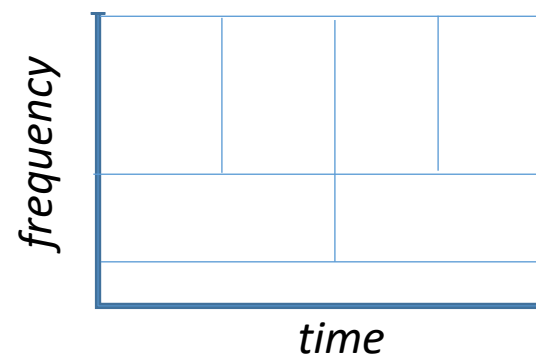
Spectrogram / Gabor filter



Spectrogram

- A grid of equally weighted time and frequency
- We know when specific frequencies exist in time

Multiresolution Analysis /
Wavelets



Wavelets

- Multiple scales in time and frequency
- Hierarchical grading of time and frequency information

What does the Wavelet look like?

- There is a **mother wavelet** $\psi(t)$
- From this mother wavelet we derive all the smaller wavelets

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$

- a is the scale parameter
- b is the translation parameter

Different scales

Large values of a



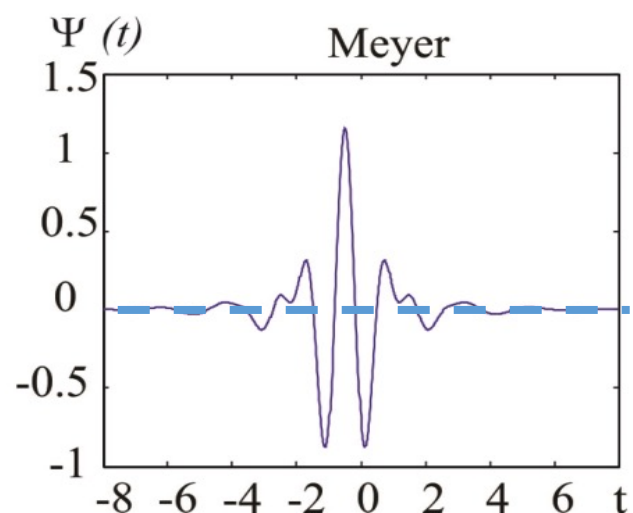
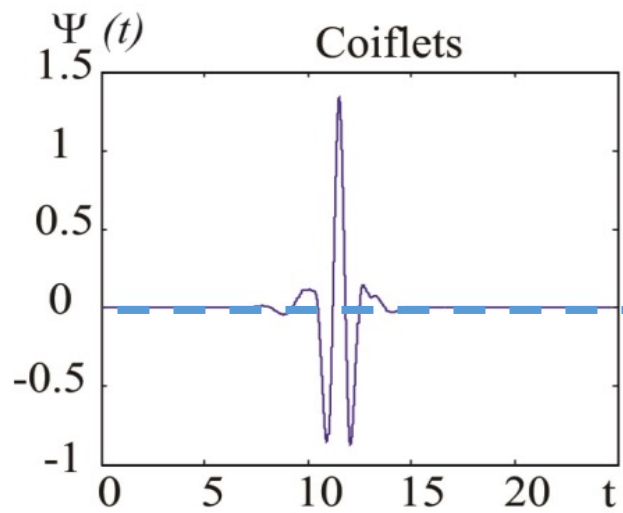
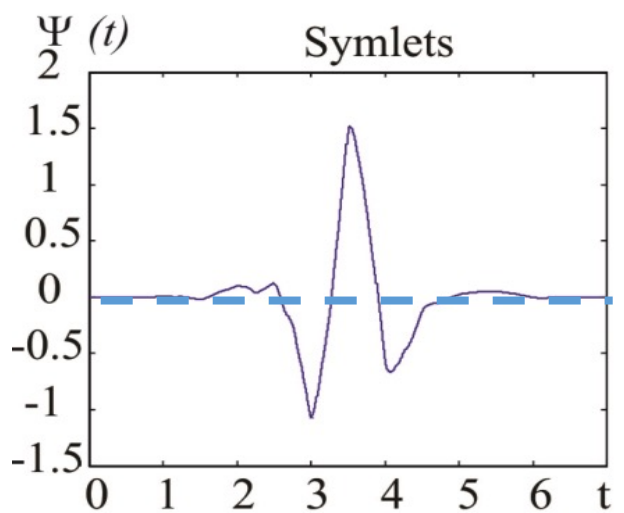
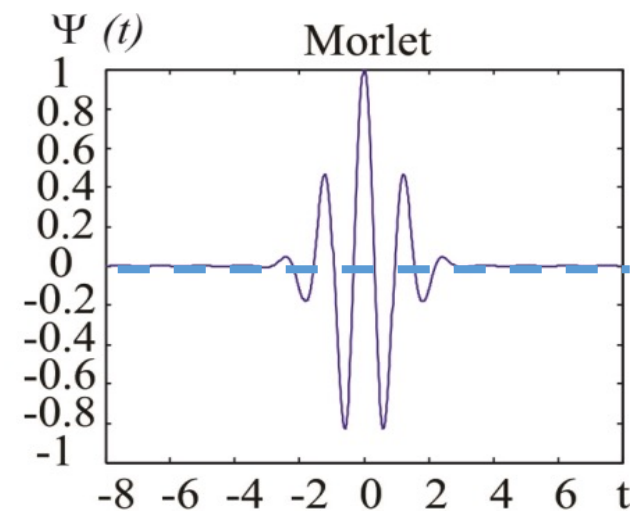
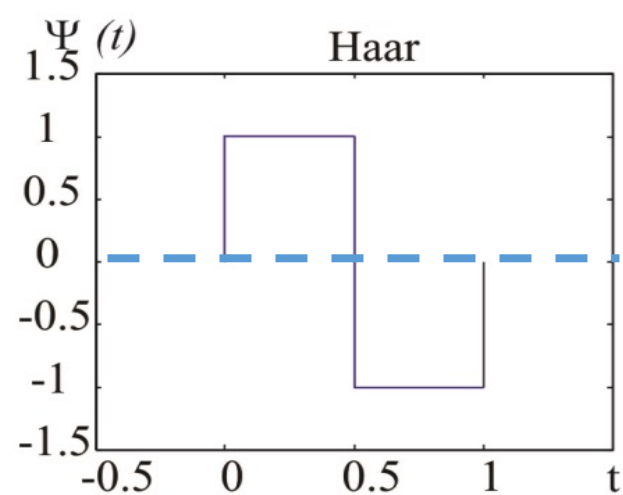
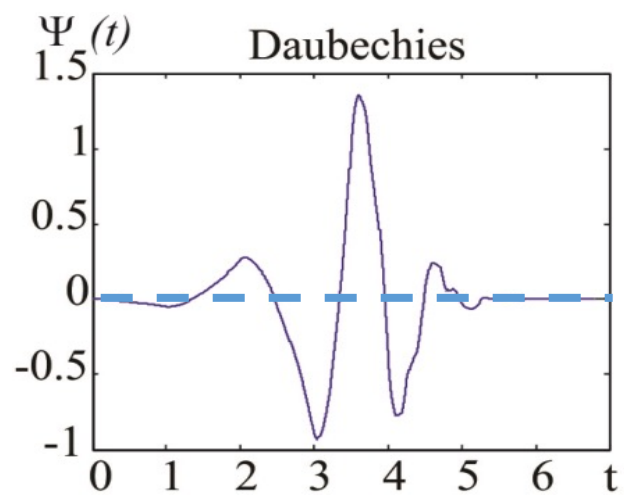
Expanded wavelet - Low frequency components with bad time resolution

Small values of a



Shrunk Wavelet - High frequency components with good time resolution

Different Wavelet types



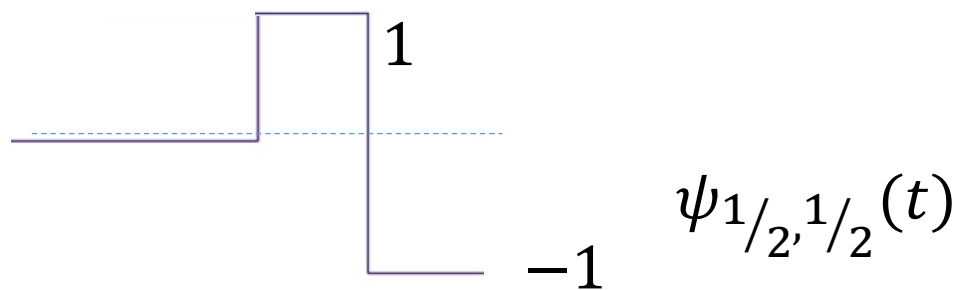
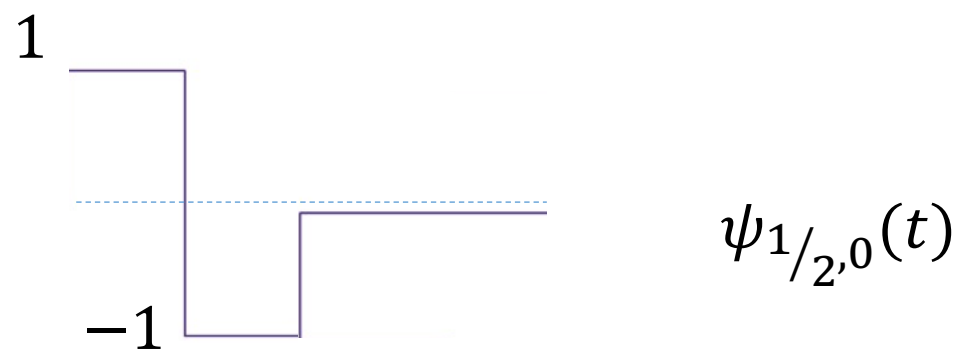
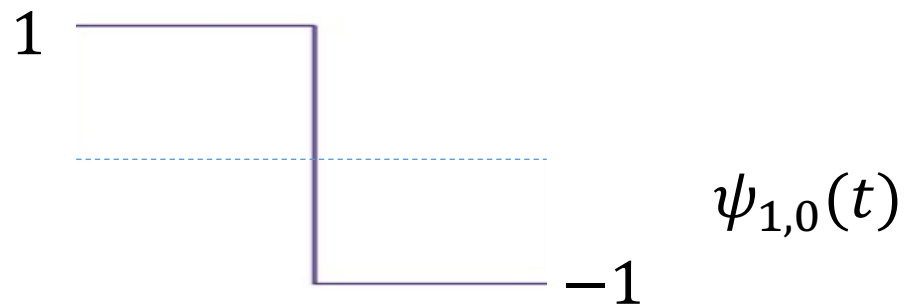
Introduction to Haar Transform

- Haar Transform is the oldest and simplest known **orthonormal wavelets**.
- Haar Transform is expressed in the following matrix form

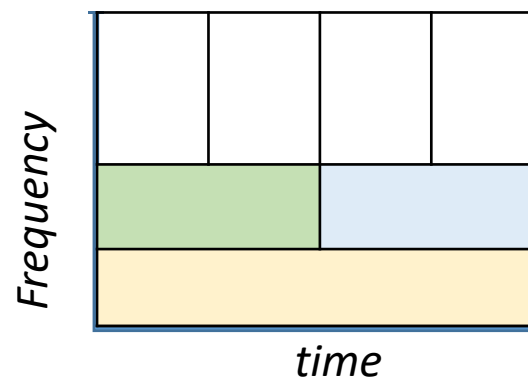
$$T = HFH^T$$

where F is an $N \times N$ image matrix, H is an $N \times N$ Haar transformation matrix and T is the resulting $N \times N$ transform. H is not symmetric thus we need to compute its transpose H^T .

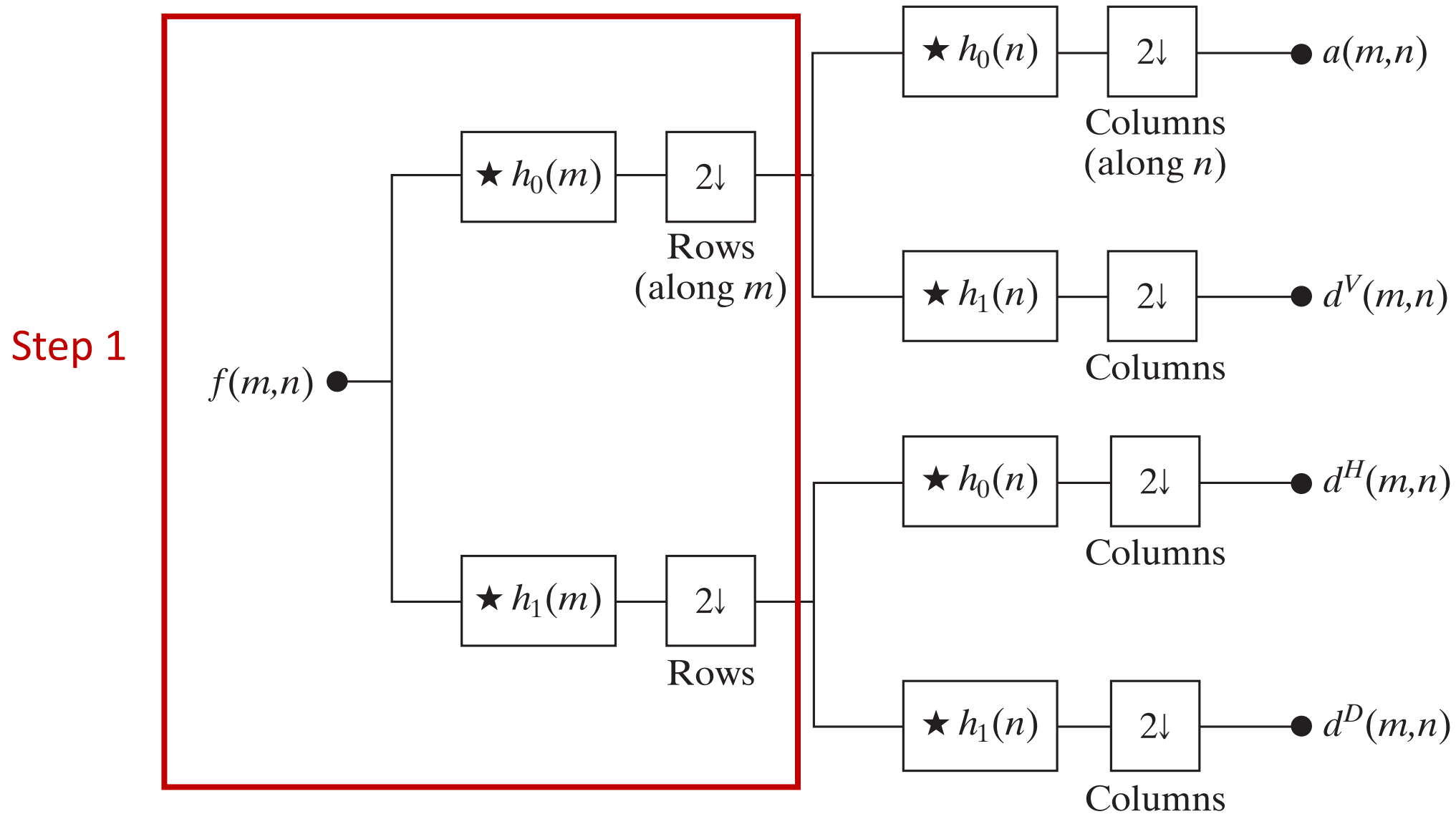
Haar Wavelet



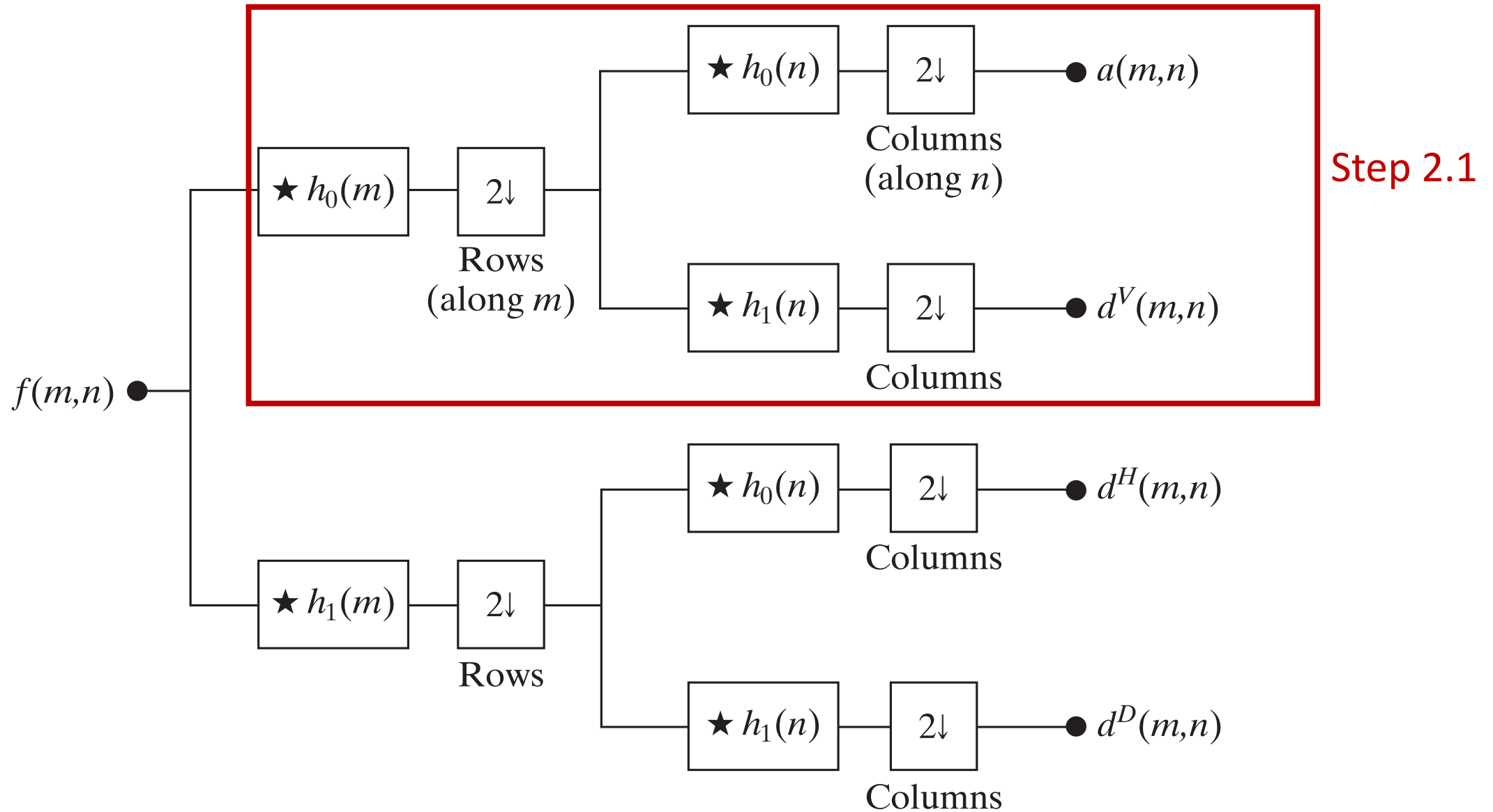
$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$



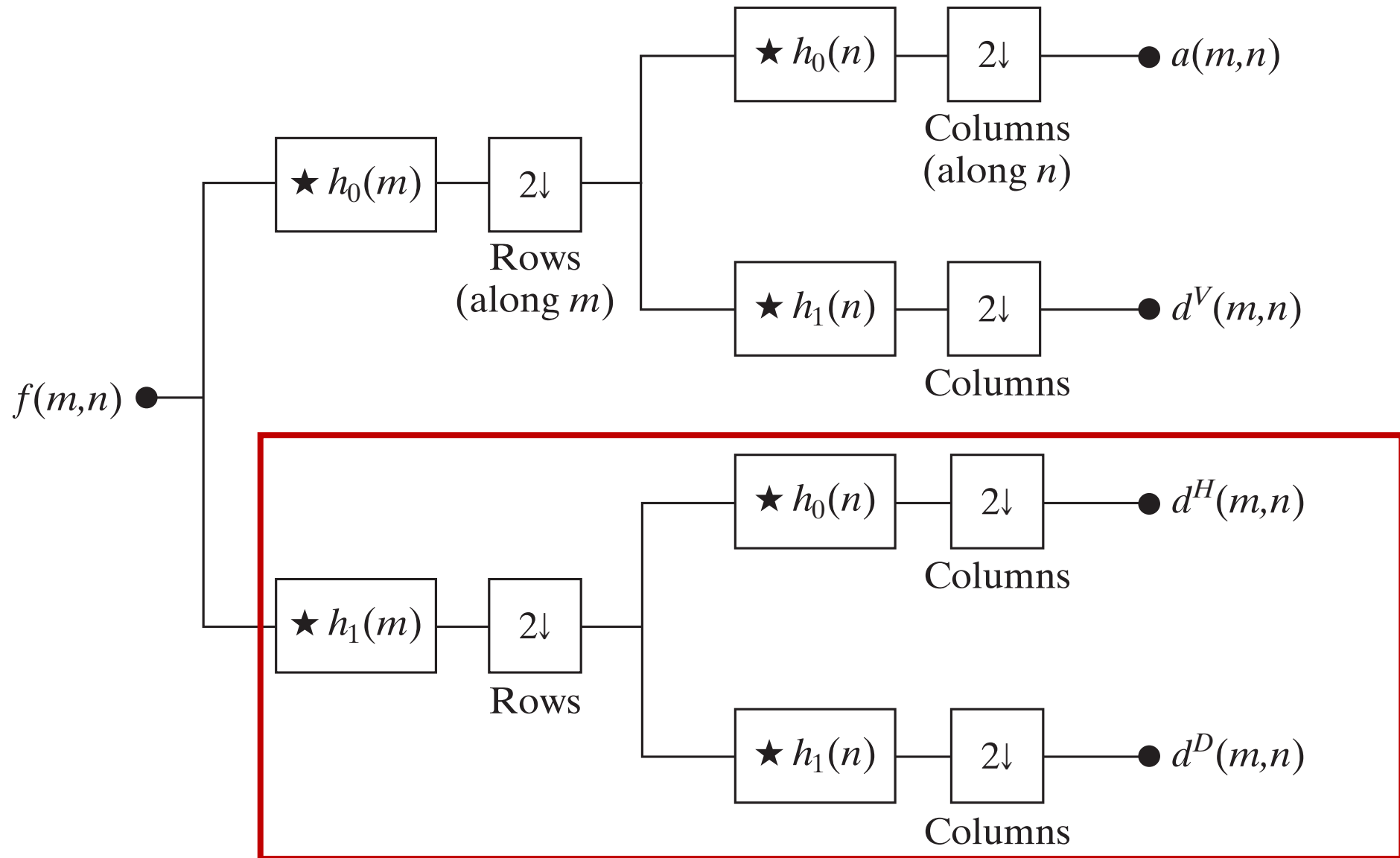
2D 4-band filter bank



2D 4-band filter bank



2D 4-band filter bank



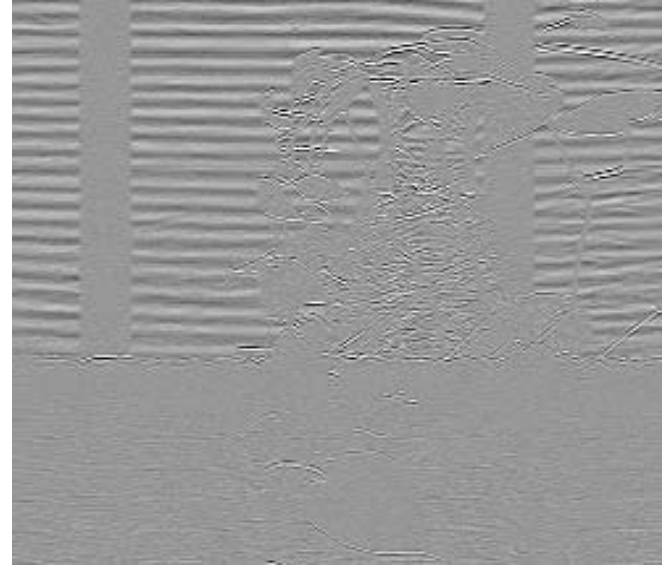
Step 2.2

2D Haar Filter - 1 Level - Example

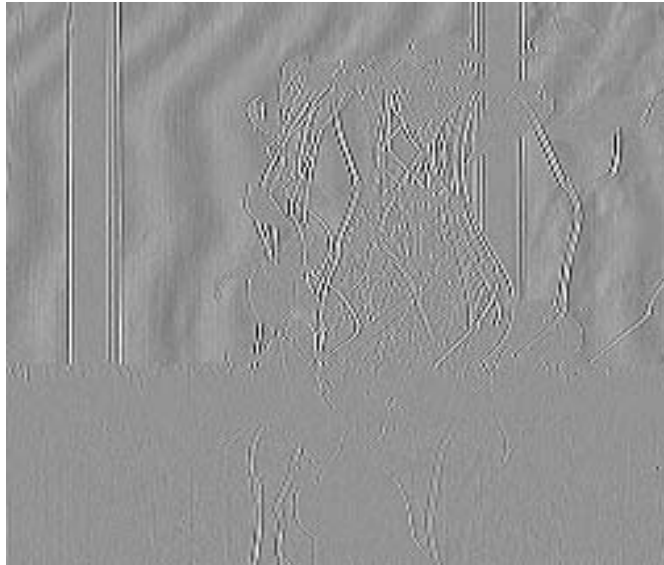
LL



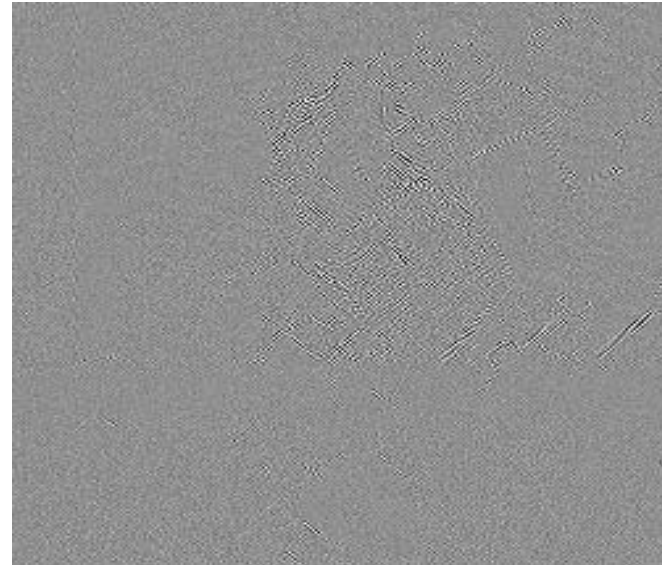
HL



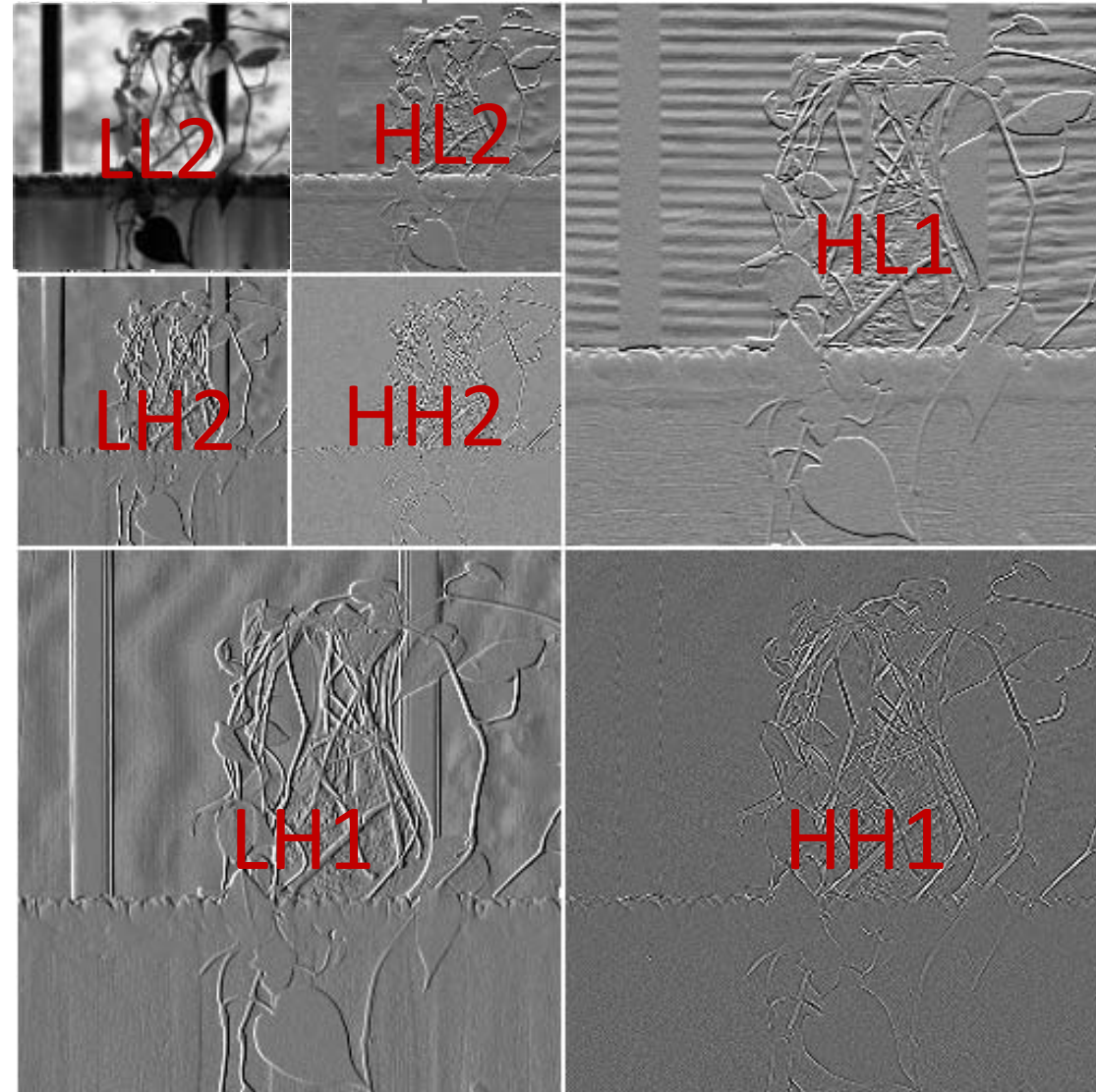
LH



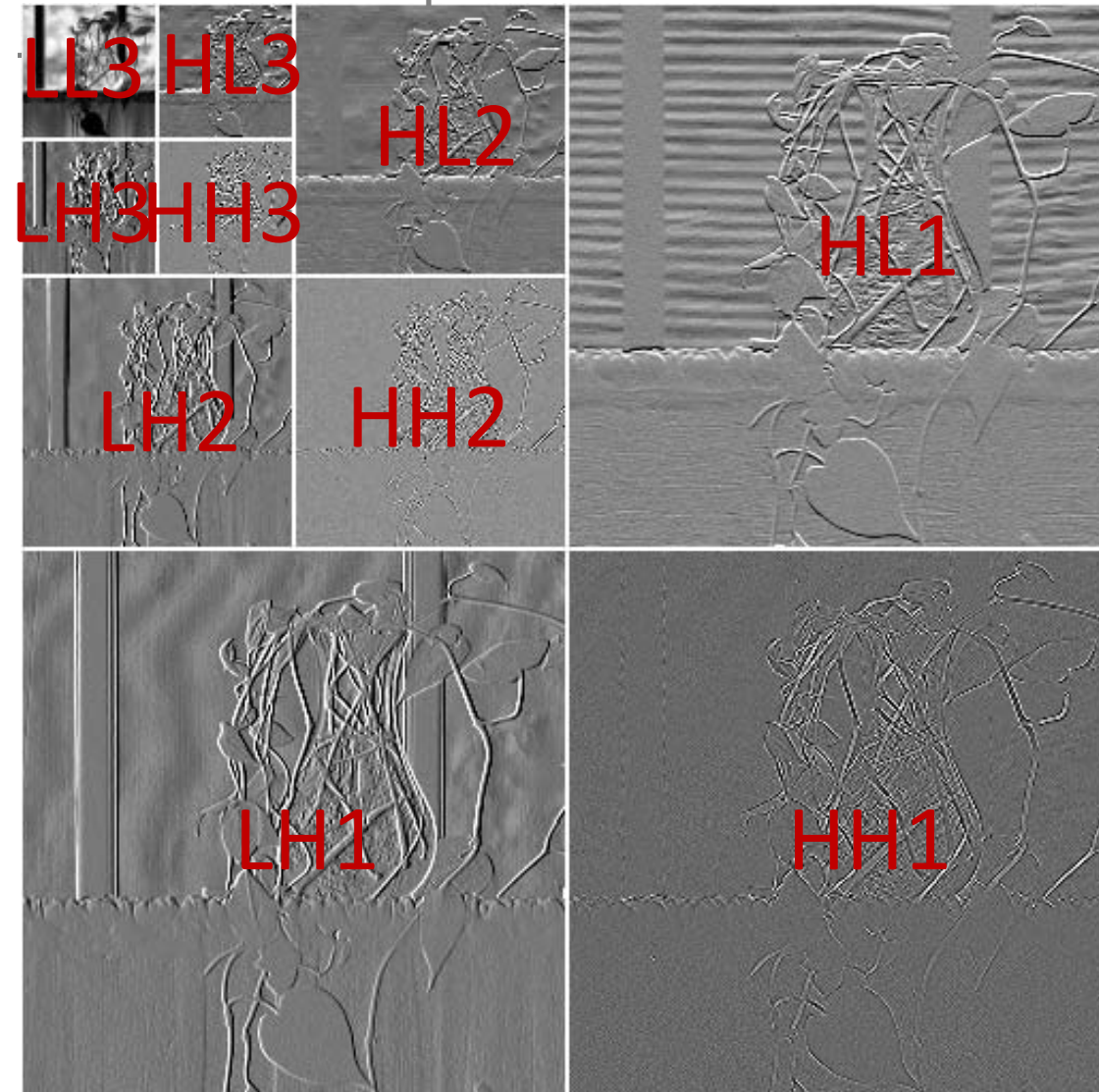
HH



2D Haar Filter - 2 Levels - Example



2D Haar Filter - 3 Levels - Example



Course Outline

- ✓ Digital Image Fundamentals
- ✓ Image Acquisition
- ✓ Image Transforms
- ✓ Convolution Theorem
- ✓ Morphological Operations
- ✓ Image Coding

Image Compression



- ❖ Easier in processing
- ❖ Suited for audio and video
- ❖ Low cost
- ❖ Less bandwidth
- ❖ Accurate representation of a sound

- ❖ Easily compressed
- ❖ Less expensive and more common equipment
- ❖ Multiple editing tools
- ❖ Easy to transmit over networks

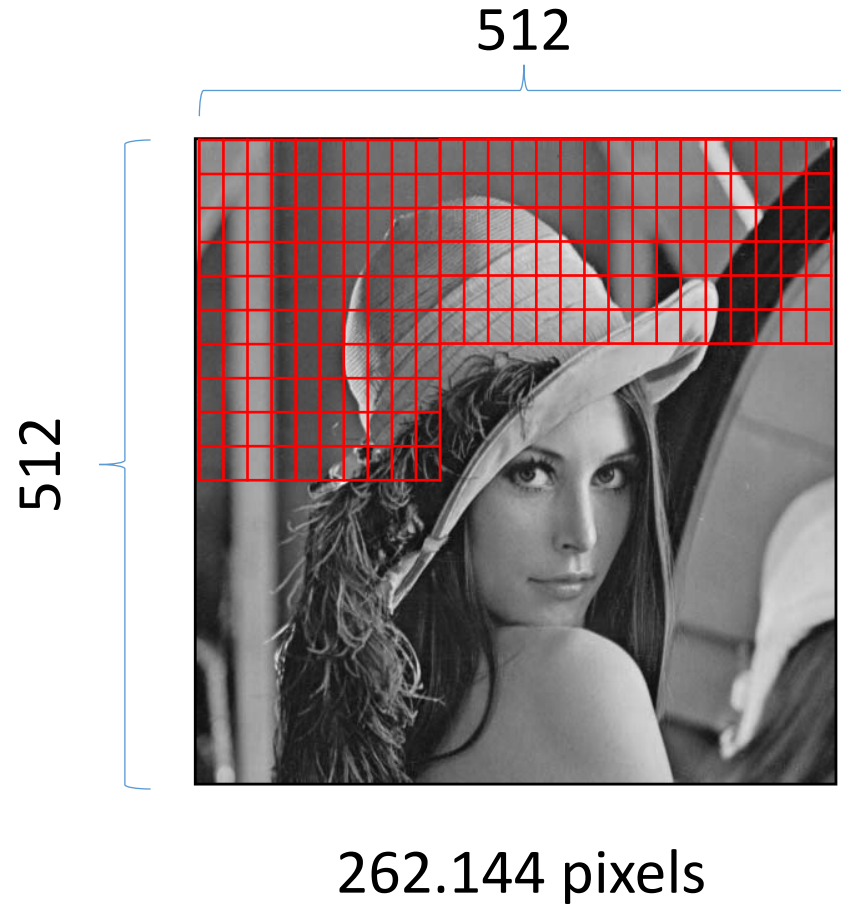
Image Compression



- ❖ Low quality signals
- ❖ Cables are sensitive
- ❖ Difficult to synchronize analog signals
- ❖ Limitations in editing
- ❖ Data can be corrupted

- ❖ Sampling/quantization might cause loss of information
- ❖ A/D and D/A demand mixed-signal hardware
- ❖ Requires great bandwidth
- ❖ Systems/processing is more complex

Digital Images



2.097.152 bits!!

Example

- Image with 256 different gray levels
- The range of these scales is between 0 (black) and 255 (white).
- Each level is represented by 8 binary elements (0 or 1) = 8bits/pixel.

i.e. 00000101 = ?

$$0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 1 = 5$$

Color Digital Images

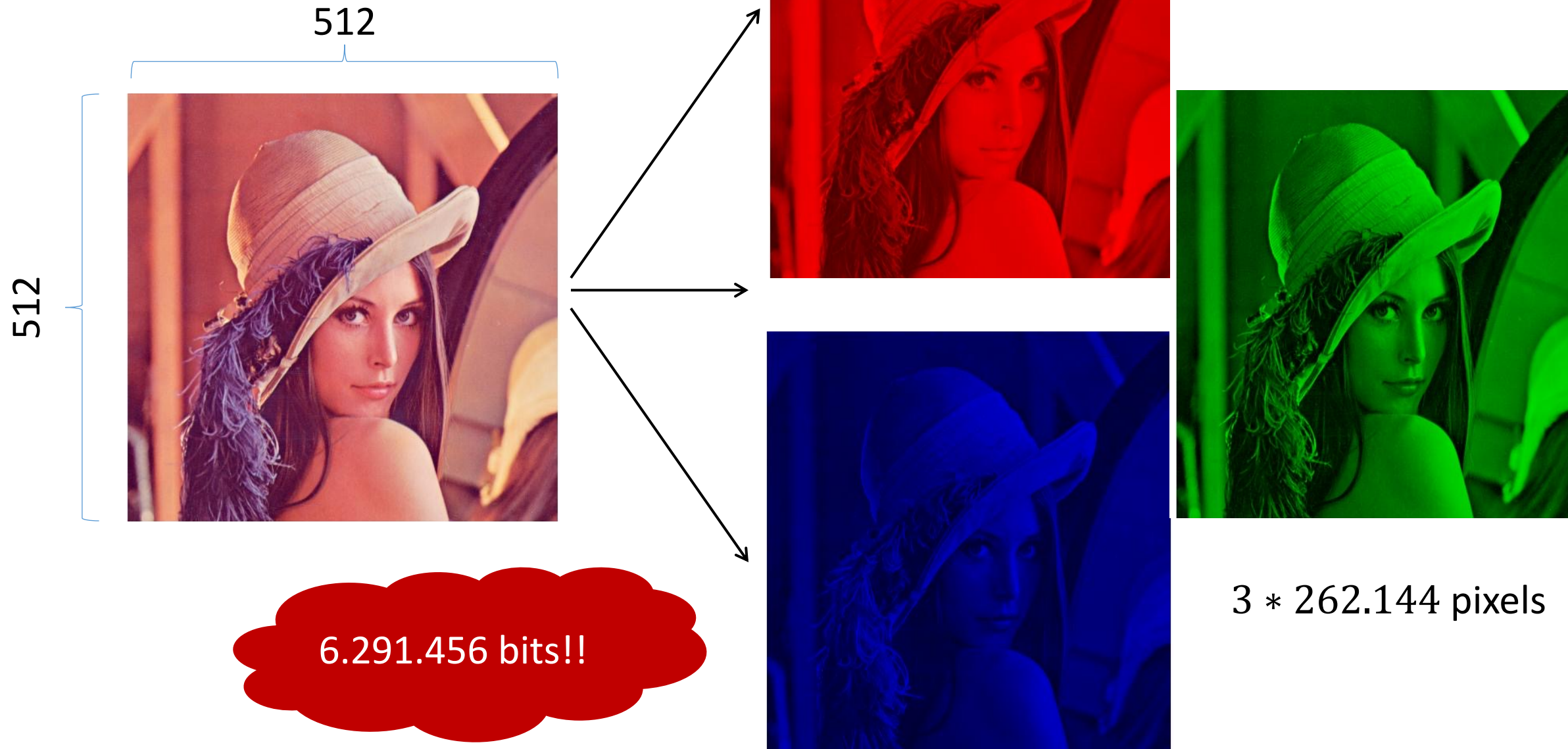


Image Resolution

≈ 99 MBytes!!

7680×4320

3840×2160

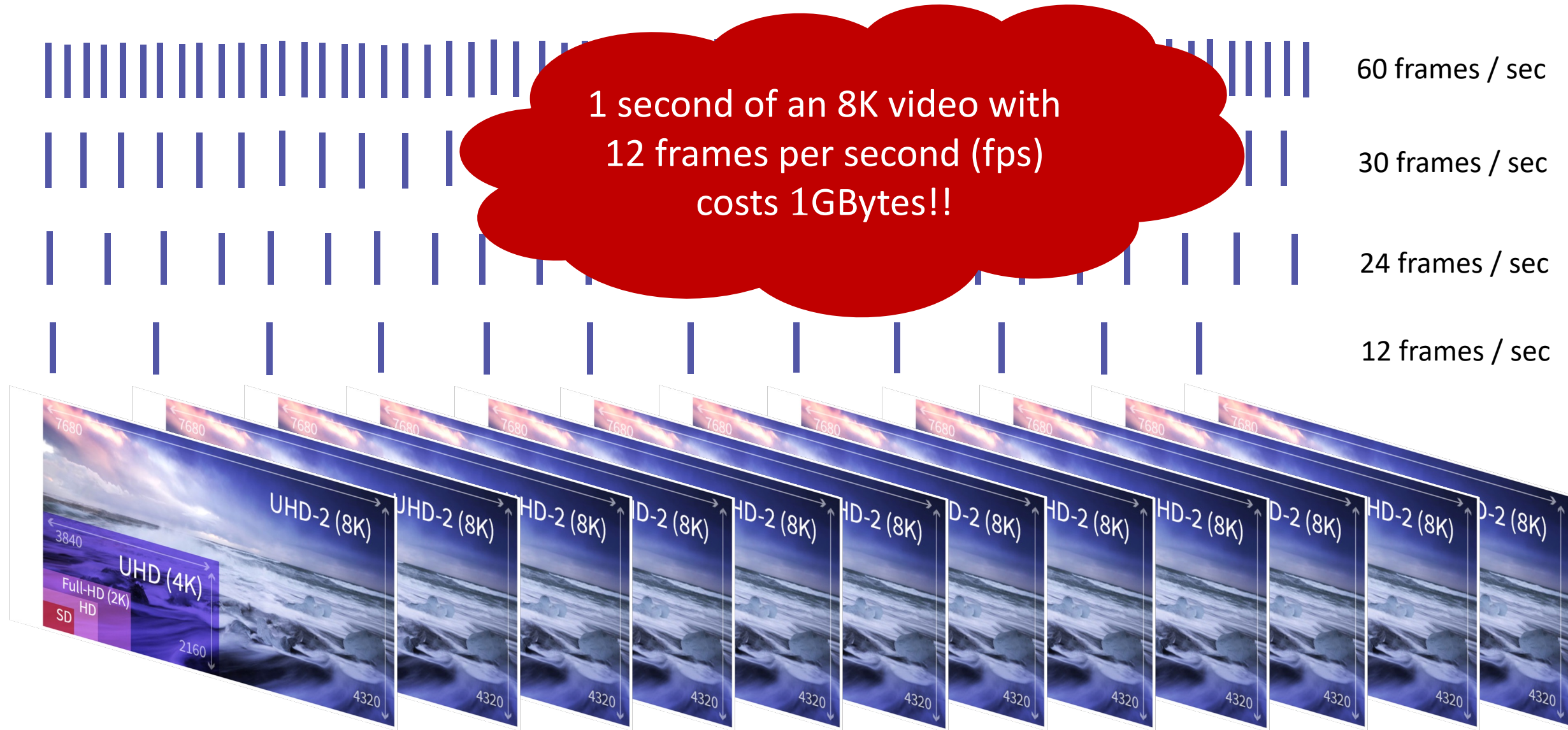
1920×1080

1280×720

640×480



Video Framerate



Coding Redundancy (1/2)

- ❖ Let r_k a discrete random variable in the interval $[0, L - 1]$ which is used to represent the intensities of an $M \times N$ image
- ❖ Each variable occurs with probability $p_r(r_k)$ such that

$$p_r(r_k) = \frac{n_k}{MN} \text{ for } k = 0, 1, 2, 3, \dots, L - 1$$

where L is the number of intensity values and n_k the number of times the k -th intensity appears in the image.

Coding Redundancy (2/2)

- ❖ Let $l(r_k)$ the number of bits used to represent each value r_k , then the average number of bits required to represent each pixels is

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

- ❖ The total number of bits required to represent an $M \times N$ image is

$$L_{total} = MNL_{avg}$$

Shannon Entropy

- ❖ Given a source of statistically independent random events form a discrete set of possible events $\{r_1, r_2, \dots, r_L\}$ with associated probabilities $\{p(r_1), p(r_2), \dots, p(r_L)\}$, the average information per source output, called **entropy** of the source, is

$$H = - \sum_{k=0}^{L-1} p(r_k) \log p(r_k)$$

Example

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8		
$r_{128} = 128$	0.47	10000000	8		
$r_{186} = 186$	0.25	11000100	8		
$r_{255} = 255$	0.03	11111111	8		
r_k for $k \neq 87, 128, 186, 255$	0	—	8		

Code 1: $L_{avg} = 0.25 \times 8 + 0.47 \times 8 + 0.25 \times 8 + 0.03 \times 8 = 8$ bits /pixel

Shannon: $H = -\{0.25 \log 0.25 + 0.47 \log 0.47 + 0.25 \log 0.25 + 0.03 \log 0.03\}$
 $= 1.6614$ bits/pixel

Shannon 1st Theorem

- ❖ Shannon's first theorem also called noiseless coding theorem assures that a source can be represented by the minimum number of bits.

$$\lim_{n \rightarrow \infty} \left[\frac{L_{avg,n}}{n} \right] = H$$

Compression Ratio – Redundancy

- ❖ If data contain irrelevant or repeated information are said to be redundant

$$R = 1 - \frac{1}{C}$$

where C is called compression ratio and it is defined as

$$C = \frac{b}{\hat{b}}$$

- ❖ If $C = 10$ then there is a 90% of redundancy.

Example

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Code 1: $L_{avg} = 0.25 \times 8 + 0.47 \times 8 + 0.25 \times 8 + 0.03 \times 8 = 8$ bits

Code 2: $L_{avg} = 0.25 \times 2 + 0.47 \times 1 + 0.25 \times 3 + 0.03 \times 3 = 1.81$ bits

$$C = \frac{256 \times 256 \times 8}{118.621} = \frac{8}{1.81} \approx 4.42$$

$$R = 1 - \frac{1}{4.42} = 0.774$$

Compression - Motivation

- ❖ **Objective:** Transmission of big data (files, images, videos, etc.)
- ❖ **Applications:**
 - Data base
 - Videoconference
 - Transmission of interactive videos
 - Telemedicine
 - High Definition TV (HDTV)
 - Digital Cinema

Compression for Transmission...

... **through networks**

- ❖ Internet
(text files, images, sounds, video, etc.)
- ❖ Telephone
(digital voice, etc.)
- ❖ Satellites
(space probes, high definition television, etc.)
- ❖ Radio-mobiles
(3rd generation GSM, UMTS, GSM, etc.)

Compression for Storage...

... to support different types

- ❖ Hard-disks, floppy disks (files)
- ❖ CD (sound, images)
- ❖ DVD (videos)
- ❖ BLUE RAY (2 hours of HDTV)
- ❖ USB key (files, sound, video, images)

Applications and Constraints (1/2)

❖ **Real Time**

- ❖ Telephone, video

(Rapid compression/decompression)

❖ **Deferred time**

- ❖ Disk storage (CD, CD ROM, DVD, USB key)

Slow compression / Rapid Decompression

- ❖ Satellite Images

Rapid Compression / Slow Decompression

Applications and Constraints (2/2)

- ❖ **Medicine:** No artifacts (wrong diagnosis)
- ❖ **Military:** Keep the details (target detection),
Movement aspect (mobile tracking)
- ❖ **Public videos:** Eye / face masks (space and time)
- ❖ **Computer Vision:** Object detection, Classification, etc.

Problem Statement

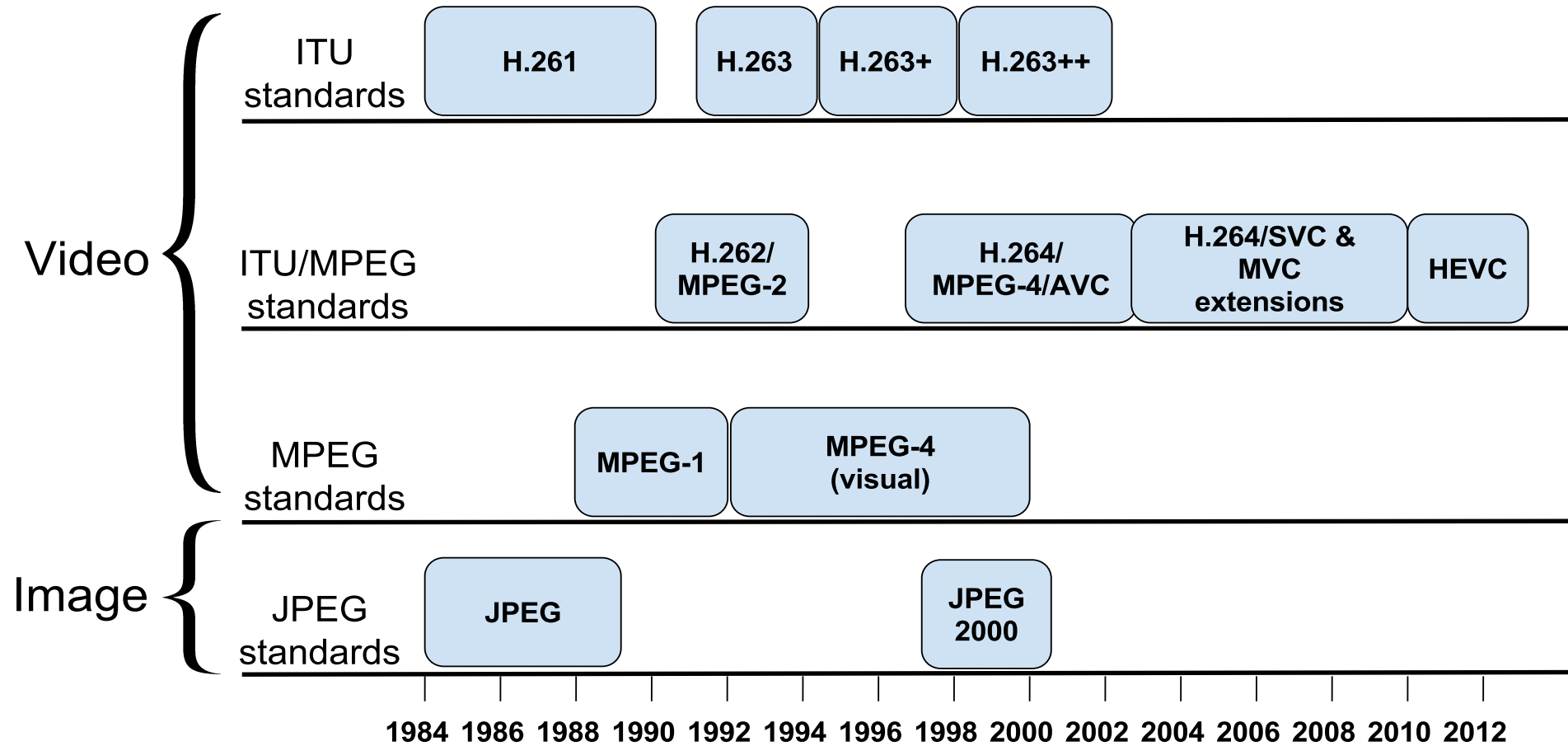


The performance of a compression system depends on:

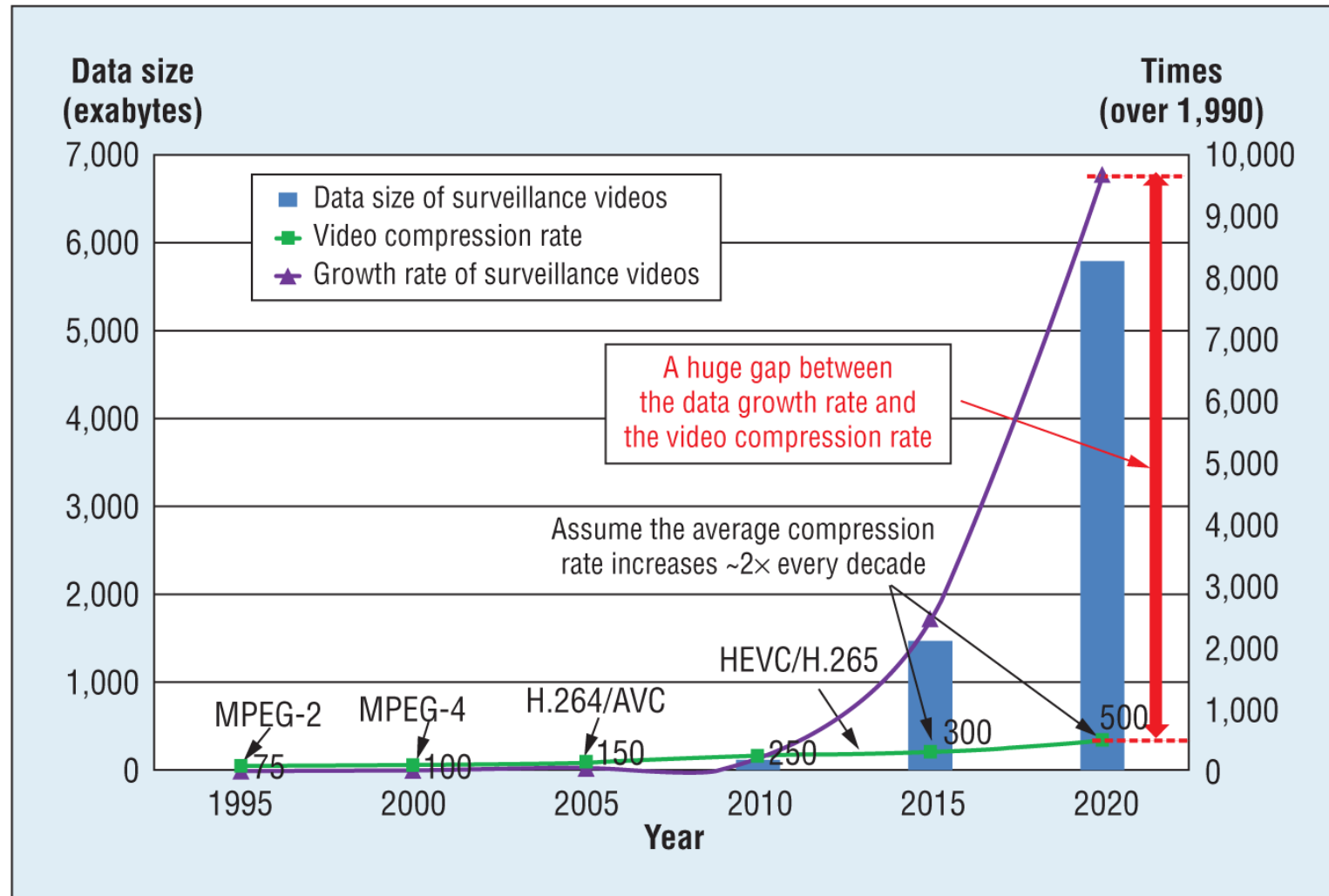
- ❖ The compression **ratio**
(initial #bits/#bits after compression)
- ❖ The **quality** of the compressed signal
(subjective/objective criteria)
- ❖ The **complexity** of the system
(computational/memory cost)

PROBLEM: How to optimize each of the 3 factors at the same time?

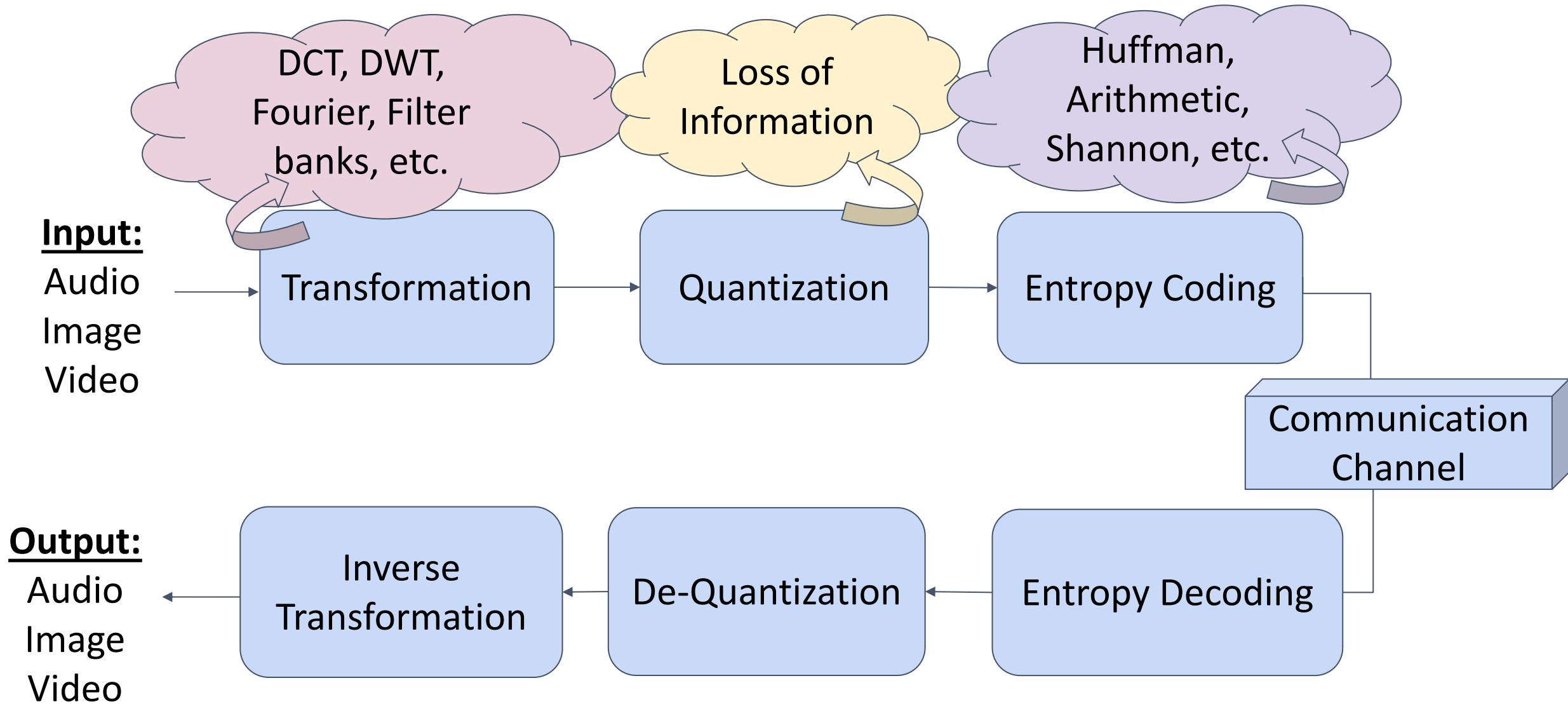
Standardization History



Progress of Compression Standards



Compression Architecture

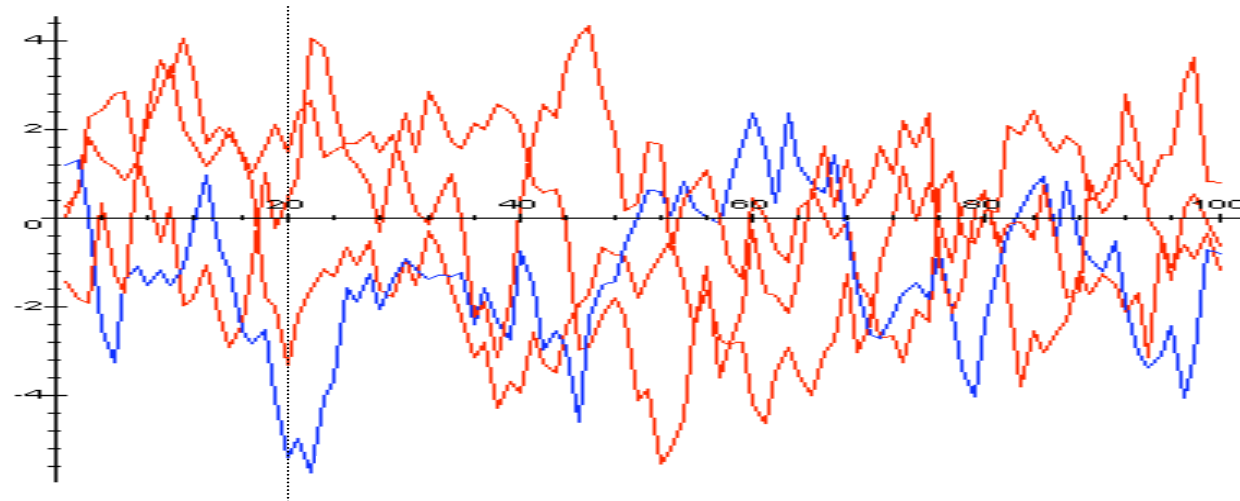


Digital Image

Image: two-dimensional random process $f(x, y)$ where (x, y) are the spatial coordinates of a pixel.

Pixel Intensities: random variable

1D Example



Quantization Principle

- ❖ **Quantization** is the fundamental operation of a compression system. Its purpose is to select for a given input value, the closest neighbor belonging to a predetermined finite set of digital values.
- ❖ A **scalar quantizer** of size L is a map Q of R in a finite set C , also called a dictionary (or codebook), containing L scalar symbols

$$Q: R \rightarrow C \text{ with } C = \{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_L\}$$

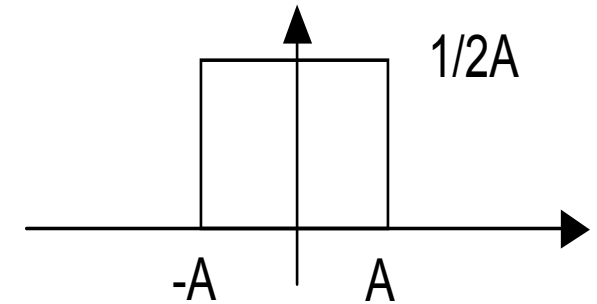
- ❖ We note $\hat{x} = Q(x)$ the quantization of x .

Uniform Scalar Quantizer

- ❖ Consider a discrete in time signal $f(x)$ that belongs to the interval $[-A, A]$ with a uniform distribution

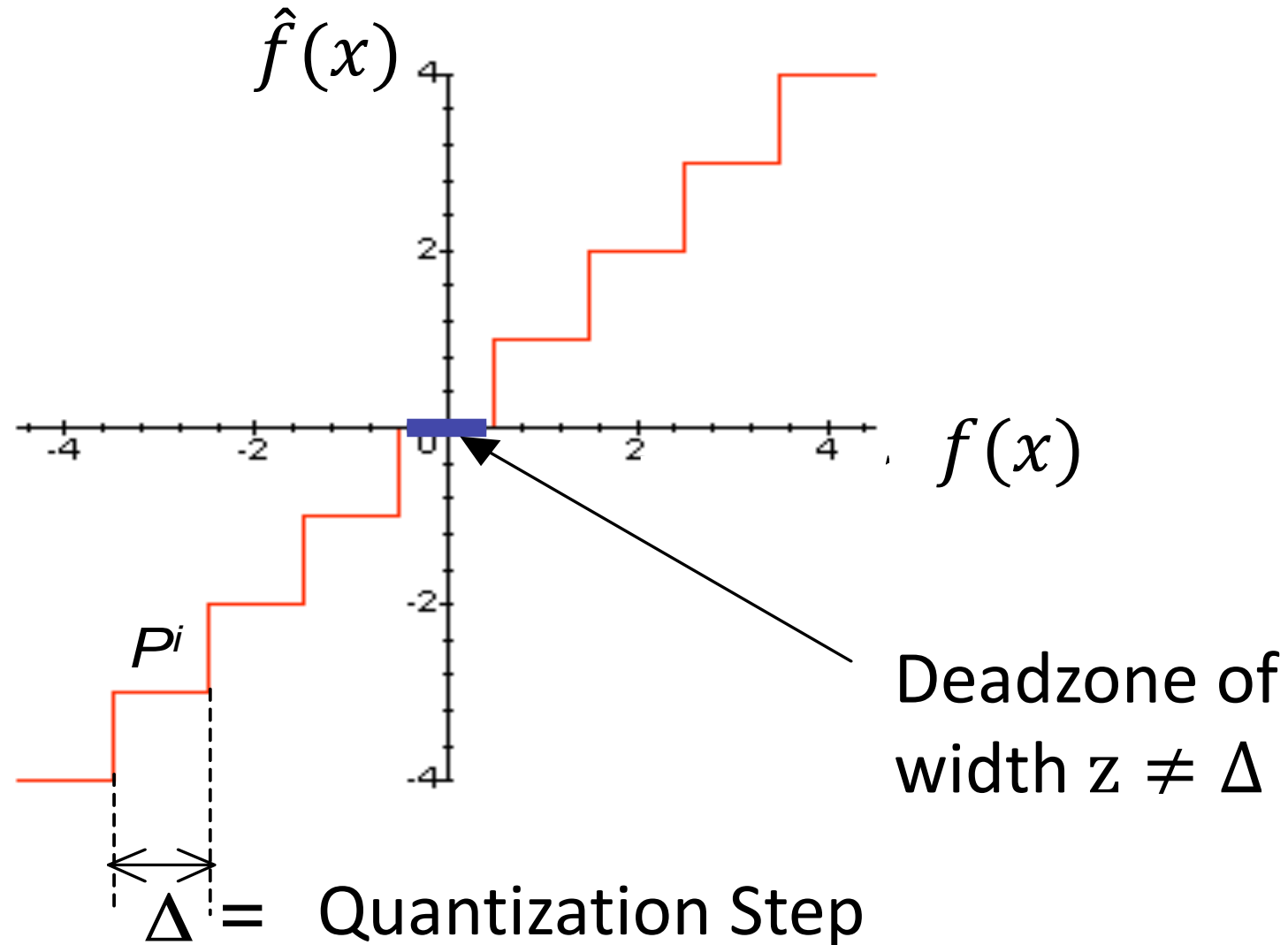
$$f_k(x) = \frac{1}{2A} \quad \forall k \in [-A, A]$$

- ❖ Divide the interval in $L = 2^R$ different intervals $\{P^1, \dots, P^L\}$ of the same length $\Delta = \frac{2A}{2^R}$
- ❖ Enumerate each interval
- ❖ Define a representative value for each interval.



Characteristic Function

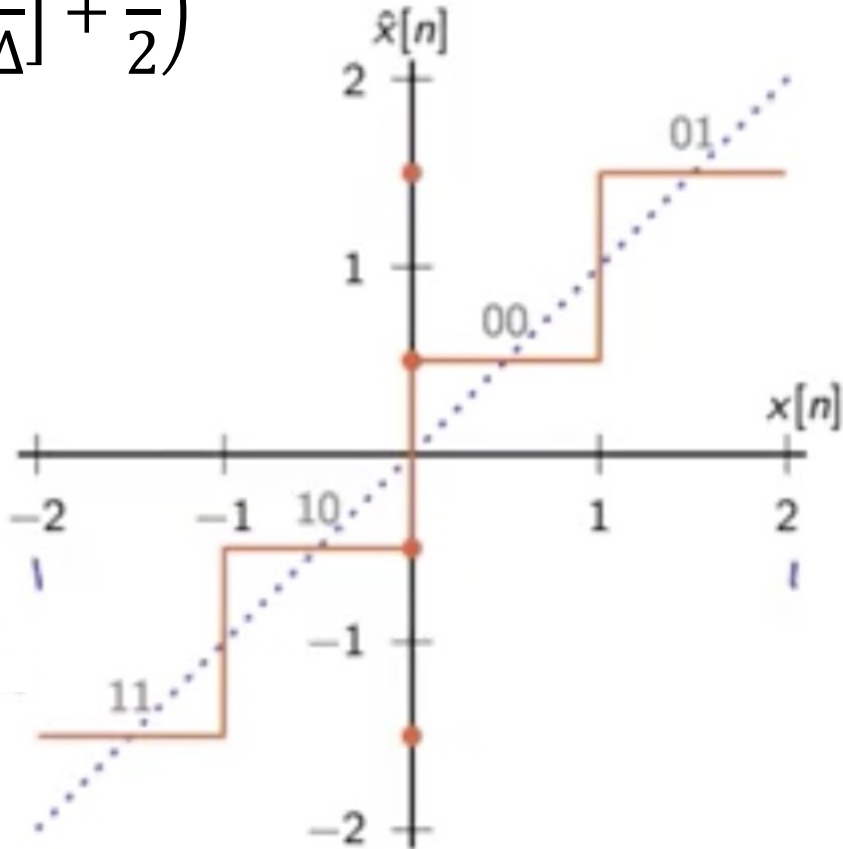
P^i represent the different classes.



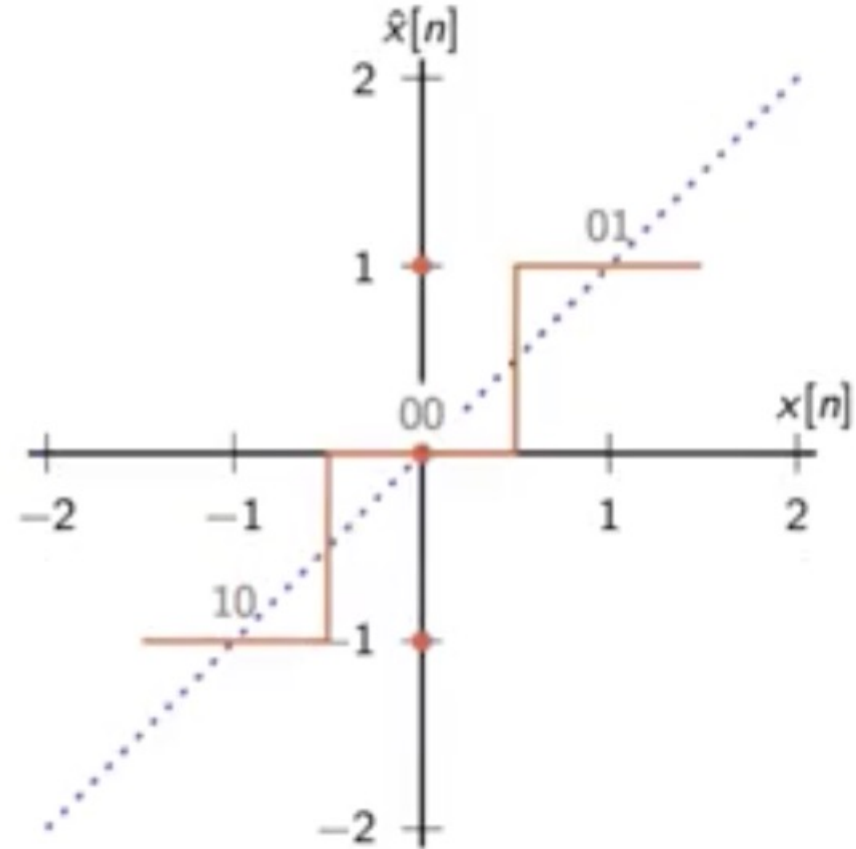
Mid-rise vs Mid-tread Quantizer

$$\hat{x} = \Delta \left(\left\lfloor \frac{x}{\Delta} \right\rfloor + \frac{1}{2} \right)$$

$$\hat{x} = \Delta \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor$$



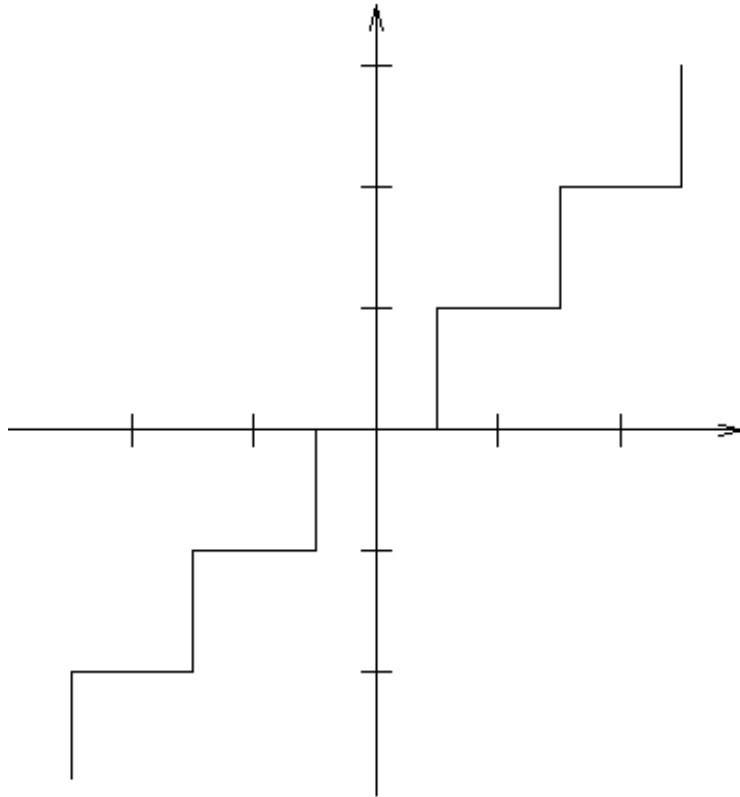
Mid-rise quantizer



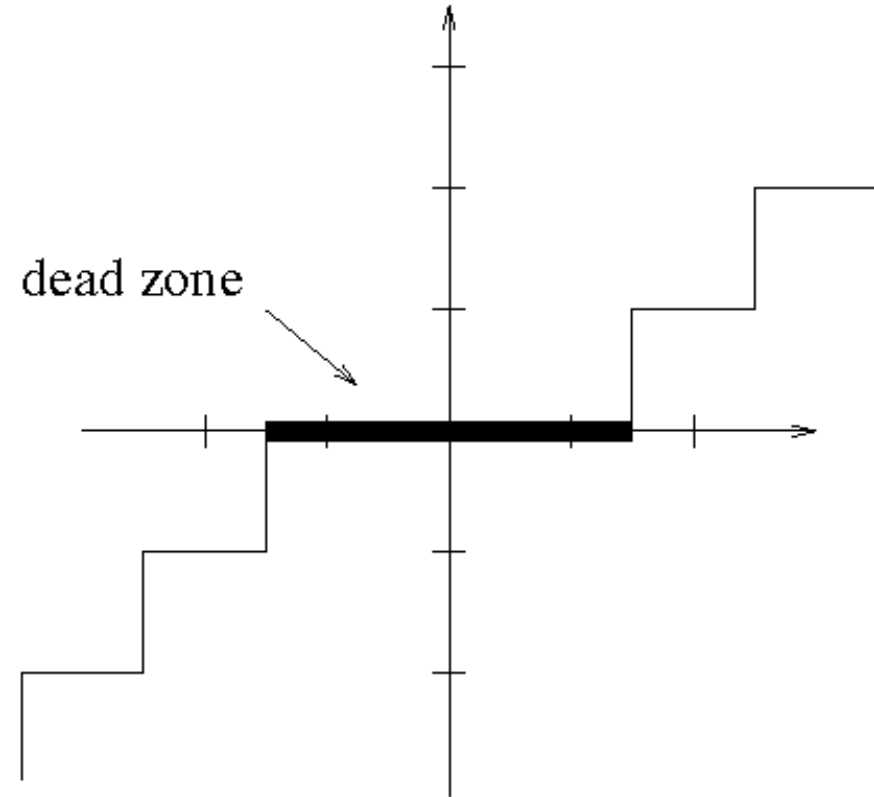
Mid-tread quantizer

Deadzone Quantizer

$$\hat{x} = \Delta \times \max \left(0, \left\lfloor \frac{x - z/2}{\Delta} + 1 \right\rfloor \right)$$



Mid-tread quantizer



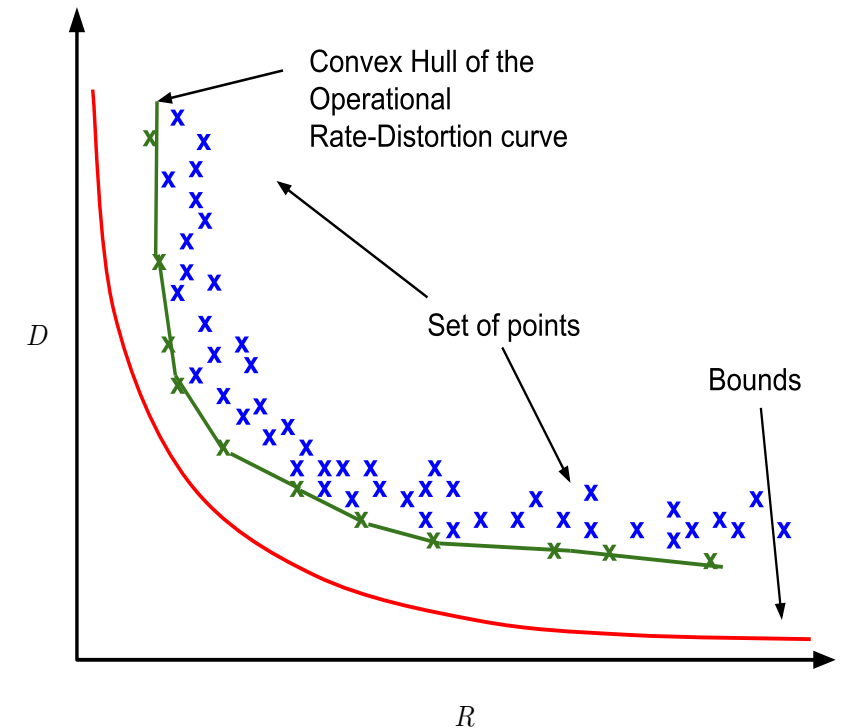
Deadzone quantizer

Rate-Distortion Optimality

- ❖ The RD optimization requires to minimize the distortion D under the constraint of $R < R_{max}$, where R_{max} is a maximum bitrate bound.
- ❖ A very well known method which has been used to seek for the operational rate-distortion curve is the Lagrangian optimization algorithm.
- ❖ The optimization algorithm of the RD curve for biorthogonal sources could be described by a cost function J depending on the distortion D and the rate R which need to be optimized

$$J_{\mu} = D + \mu R$$

where μ is the Lagrange multiplier.



In this course...

- ❖ We discussed about how we perceive the visual information
- ❖ Acquisition challenges (sampling theorem, aliasing, artifacts)
- ❖ Pixel-by-pixels transforms as well as filters applied to bigger areas of an image
- ❖ The Convolution Theorem
- ❖ Morphological Analysis of an image
- ❖ Multiresolution analysis that allows us to extract different kind of information at each level
- ❖ Basic principles in image coding

Thank
you