FBSP: Image Processing Introduction

Nick Yao Larsen

nylarsen@cfin.au.dk

About me

Nick Yao Larsen, PhD

Email nylarsen@cfin.au.dk

- Masters in Neuroscience and Neuroimaging at SDC
- PhD from Aarhus University and and UCAS
  - Specialized in Microscope image processing

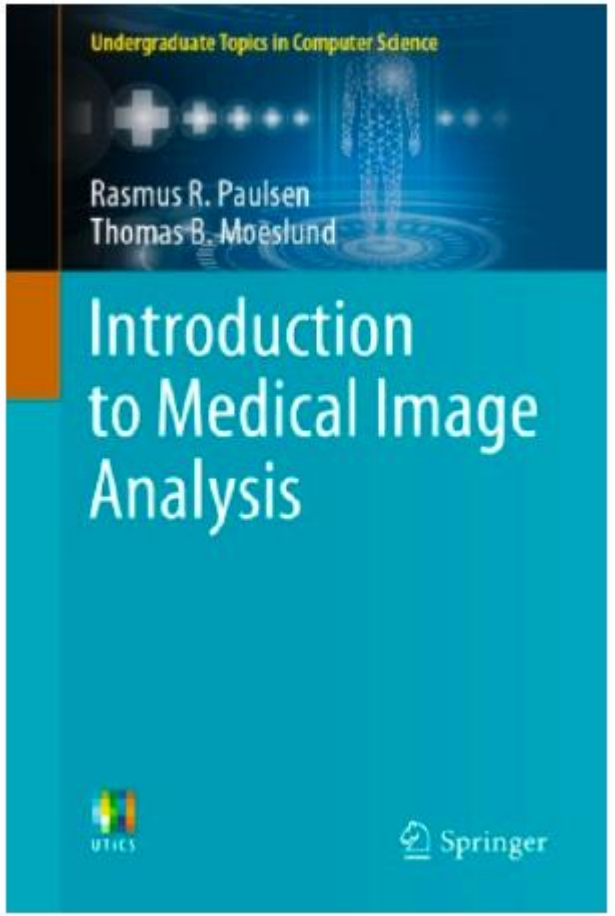
Lecture plan

3/11 13:30 - 15:10	Image Processing introduction Digital images	IMIA: p. 1-25	NYL
3/11 15:30 - 17:10	Colours	IMIA: p. 103-112	NYL
7/11 8:30 - 10:10	Point processing methods	IMIA: p. 33-55	NYL
7/11 10:30 - 12:10	Neighbourhood processing methods	IMIA: p. 57-73	NYL
8/11 13:30 - 15:10	Morphological operations	IMIA: p. 75-85	NYL
9/11 13:30 - 15:10	Spatial transformations and image registration	IMIA: p. 131-153	NYL
10/11 8:30 - 10:10	Mini-project	None	NYL
10/11 10:30 - 12:10	Mini-project+ Mini-project presentation	None	NYL

Materials

Literature

- “Introduction to Medical Image Analysis” (IMIA) by Rasmus R. Paulsen and Thomas B. Moeslund



Procedure

- Cover both theory and practical knowledge (i.e. how to apply the methods in Python)

Why do image processing?

Medical Imaging:

Purpose: Diagnostic purposes and treatment planning.

Enhancement of MRI or CT scans

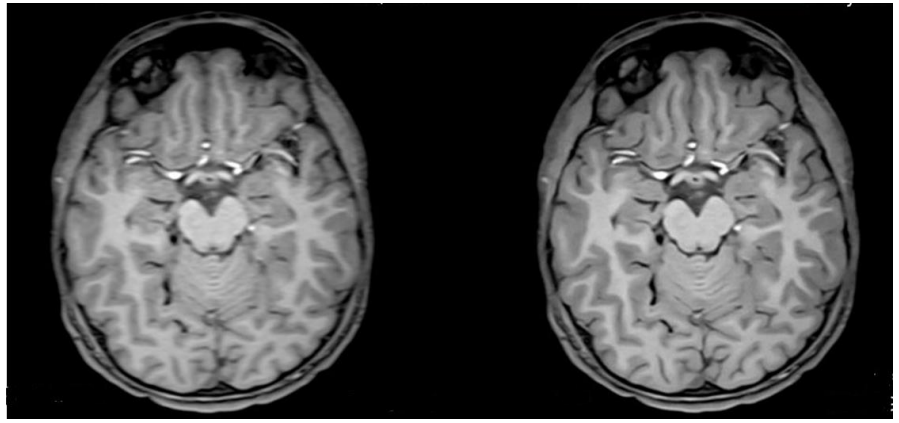
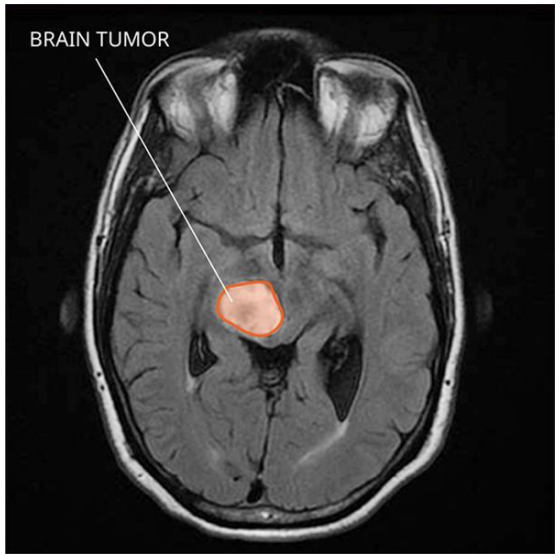
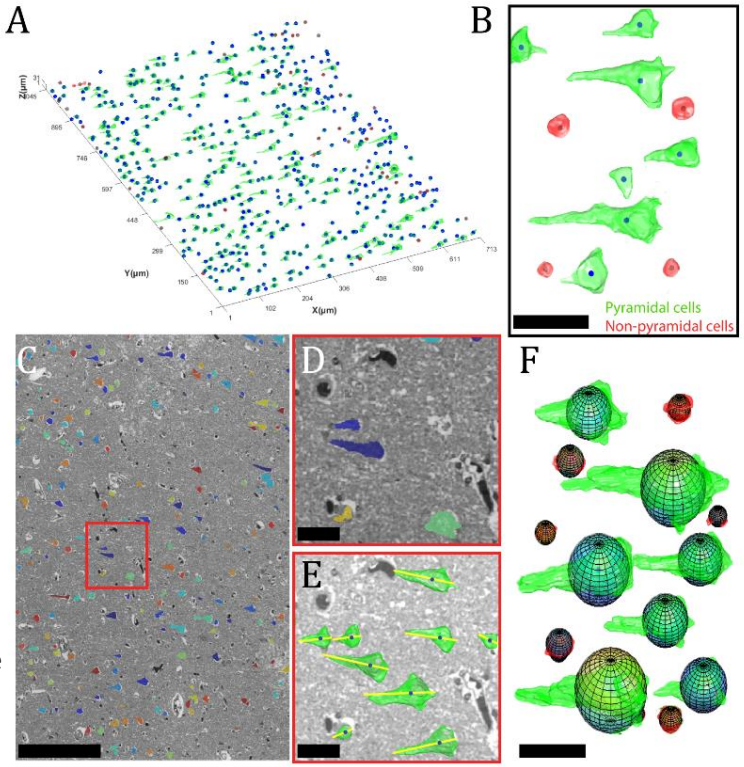


image segmentation for tumor detection



3D reconstruction of medical images



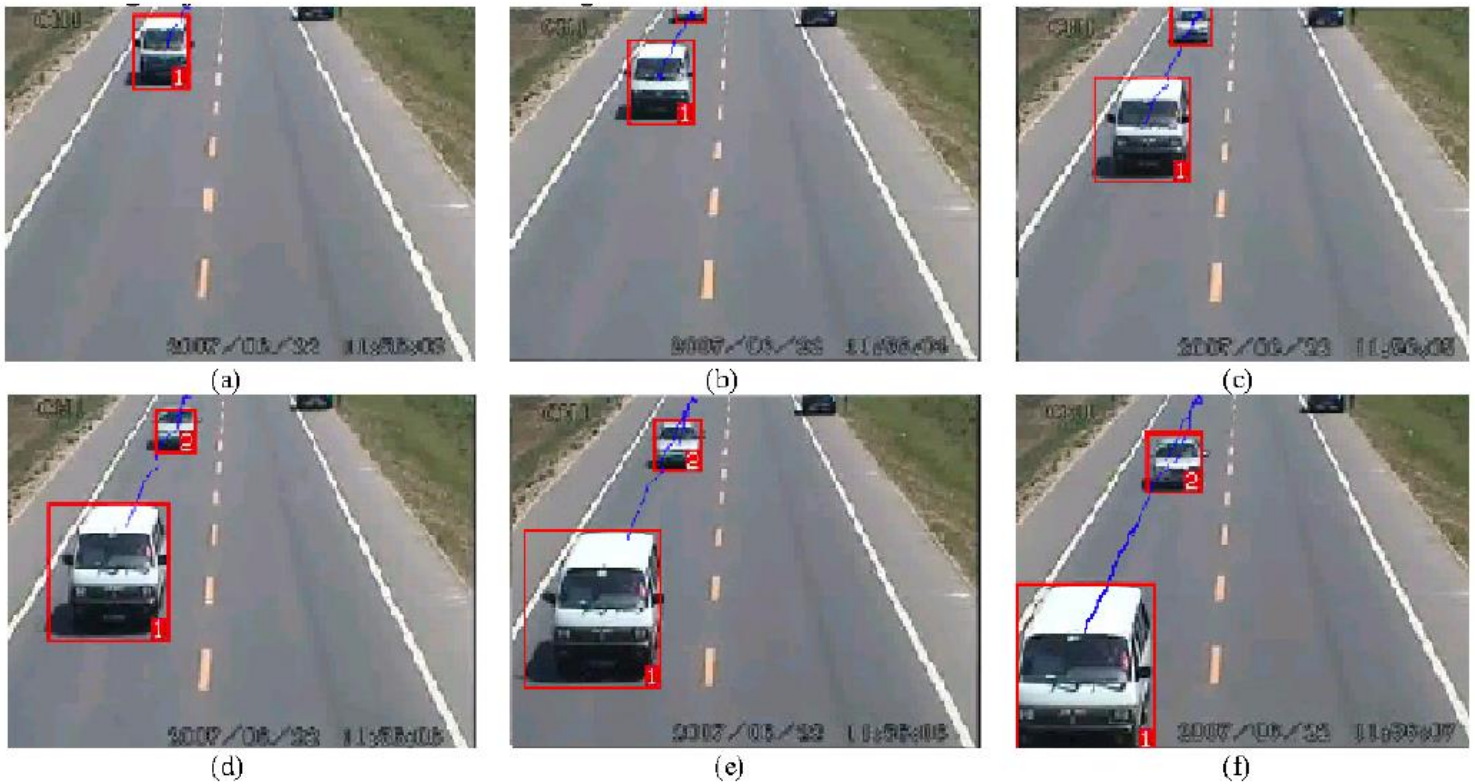
Larsen, N.Y. *et al.* Cellular 3D-reconstruction and analysis in the human cerebral cortex using automatic serial sections. *Commun Biol* **4**, 1030 (2021). https://doi.org/10.1038/s42003-021-02548-6

Why do image processing?

Computer Vision and Object Recognition:

Purpose: Recognizing and understanding visual information.

Object detection and tracking in video surveillance



Why do image processing?

Entertainment and Multimedia:

Purpose: Creating visually appealing and realistic experiences.

Image manipulation, Video editing and special effects in filmmaking etc.

Aim and expectation of this course

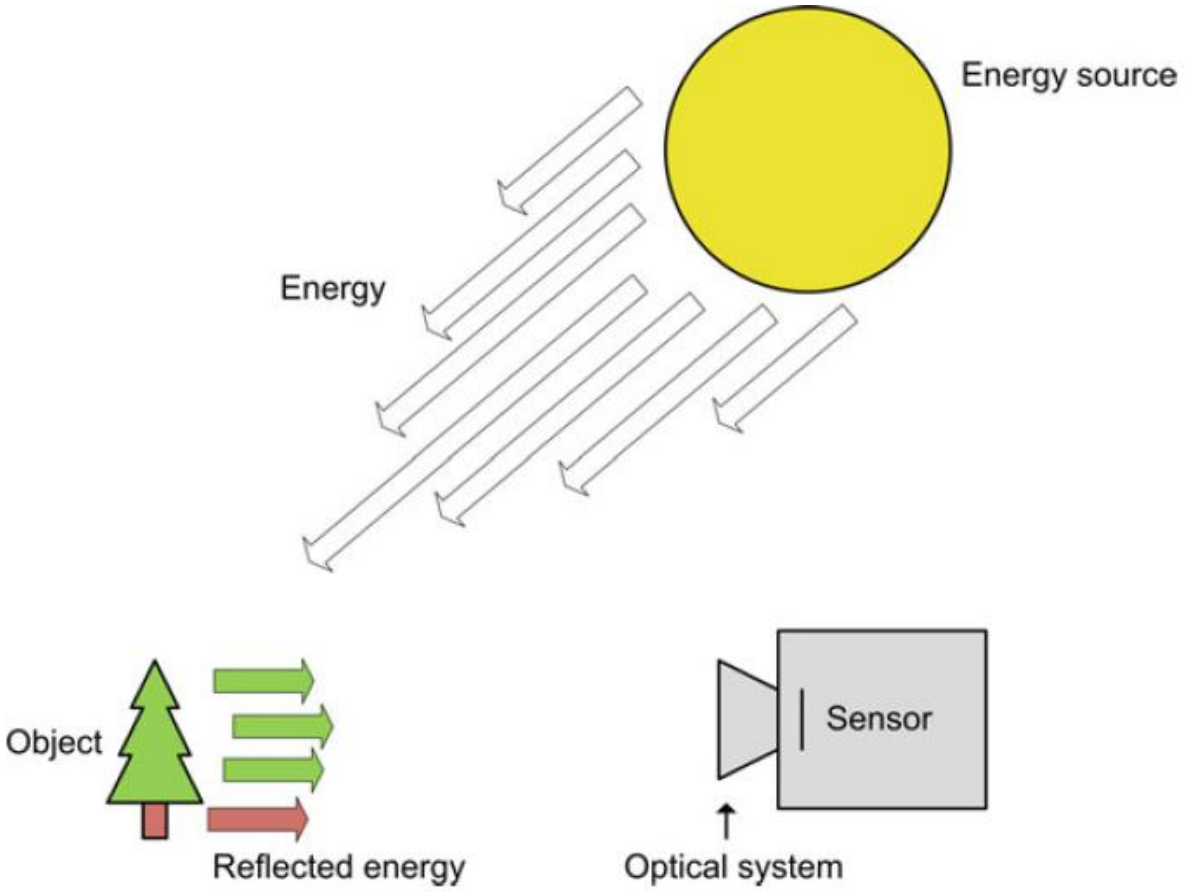
You will be able to:

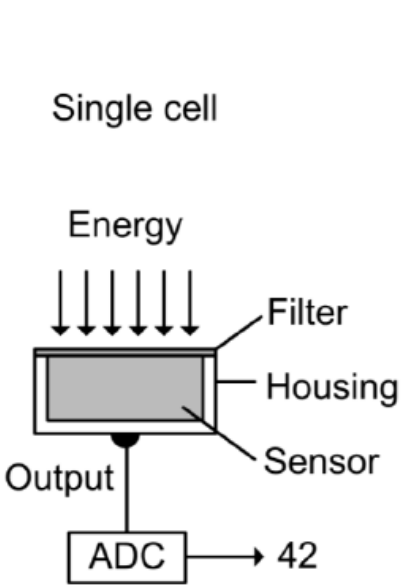
- Analyze images using basic image processing techniques.

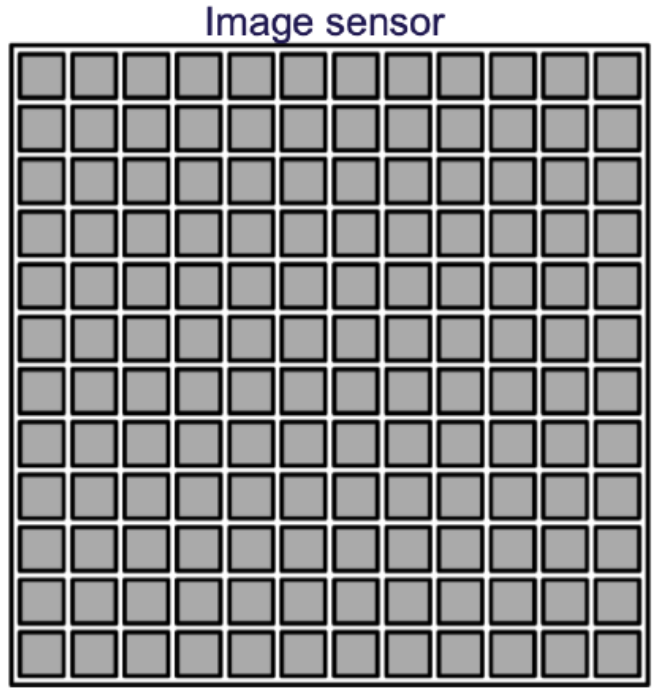
- Interpret and analyze digital signals and images

- Expect you have some basic knowledge about Python
  - Matplotlib
  - numpy

Formation of digital images

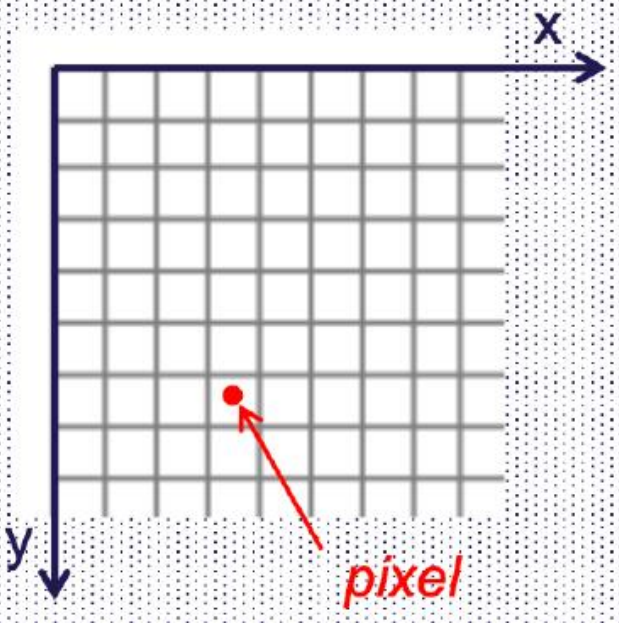






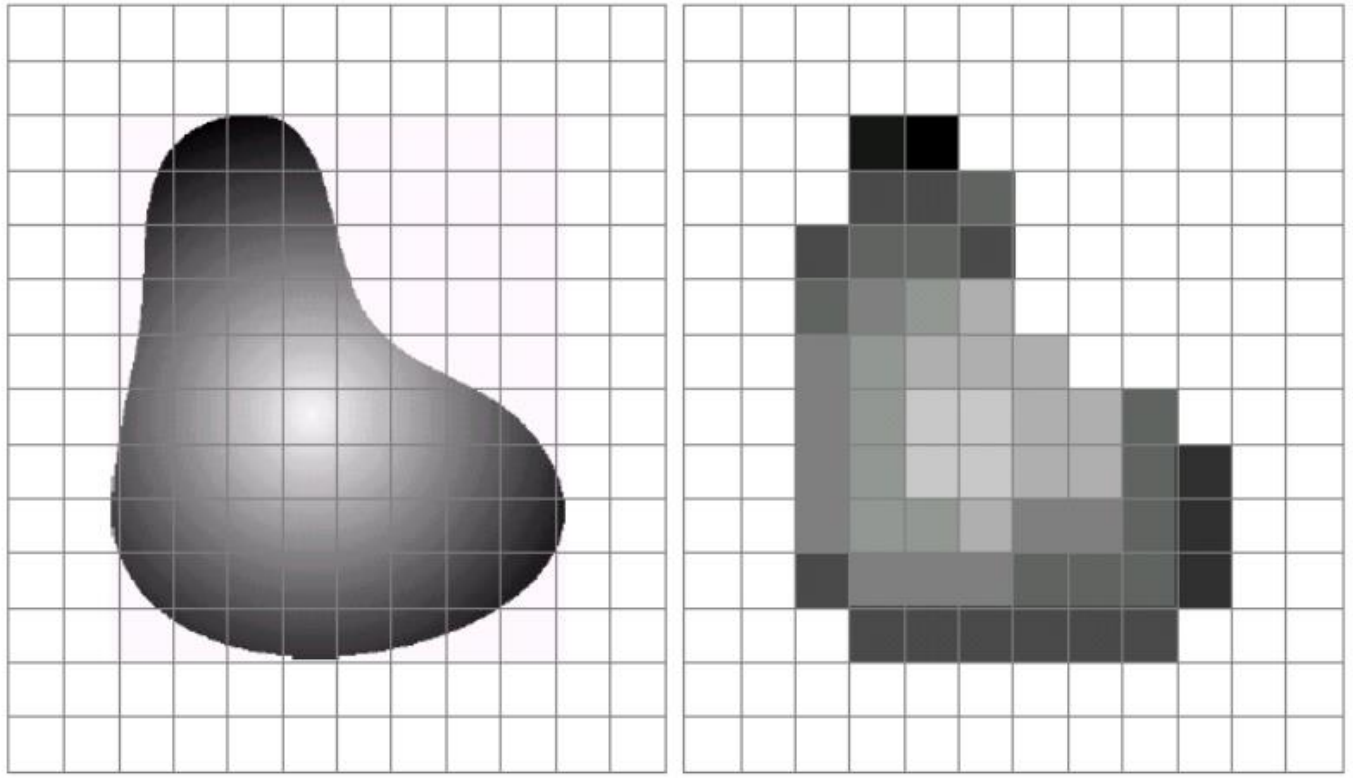
Digital images

- An image *f(x,y)* is represented as a 2D Array, matrix
- An image is seen as a discrete function *f(x,y)* as opposed to a continuous function
- Each dot/cell in an image is called a pixel
- *x* and *y* cannot take on any value!
  They must be within the bounds of the image.
  - Width = number of pixels in x-direction
  - Height = number of pixels in y-direction
  - Size = (width x height)



Pixels

- **Each dot/cell in an image is called a pixel**
- Each cell has a value
- This value is a pixel intensity



Pixels

- Each dot/cell in an image is called a pixel
- **Each cell has a value**
- This value is a pixel intensity

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Pixels

- Each dot/cell in an image is called a pixel
- Each cell has a value
- **This value is a pixel intensity**

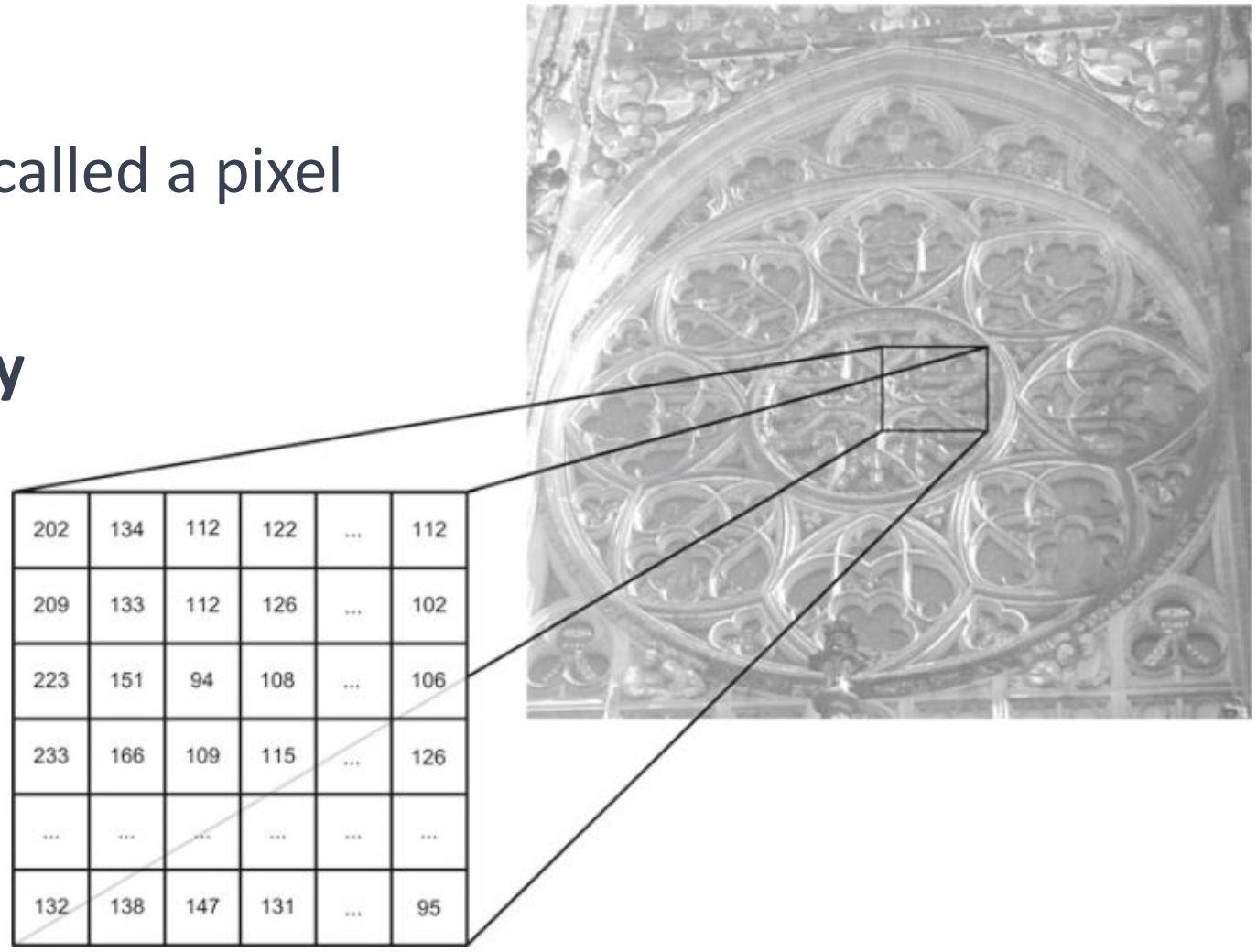
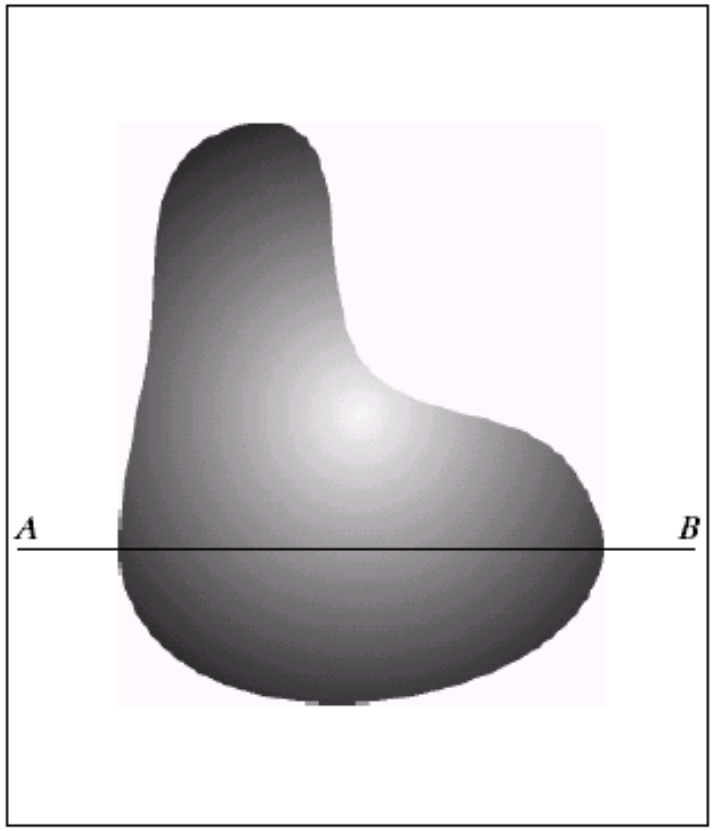
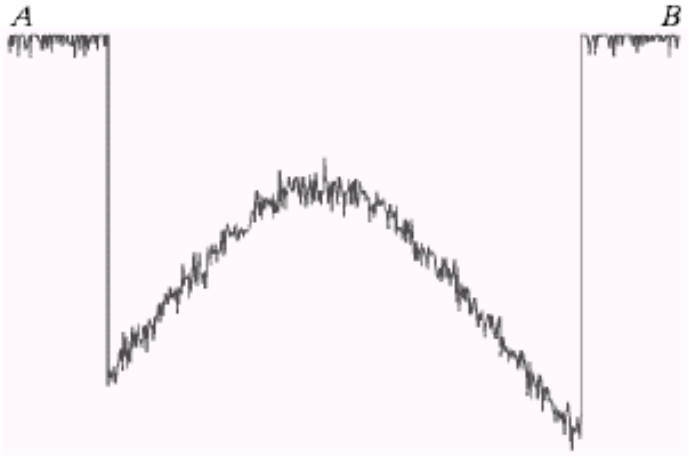


Fig. 2.20 A gray-scale image and part of the image described as a 2D array, where the cells represent pixels and the value in a cell represents the intensity of that pixel

Pixels

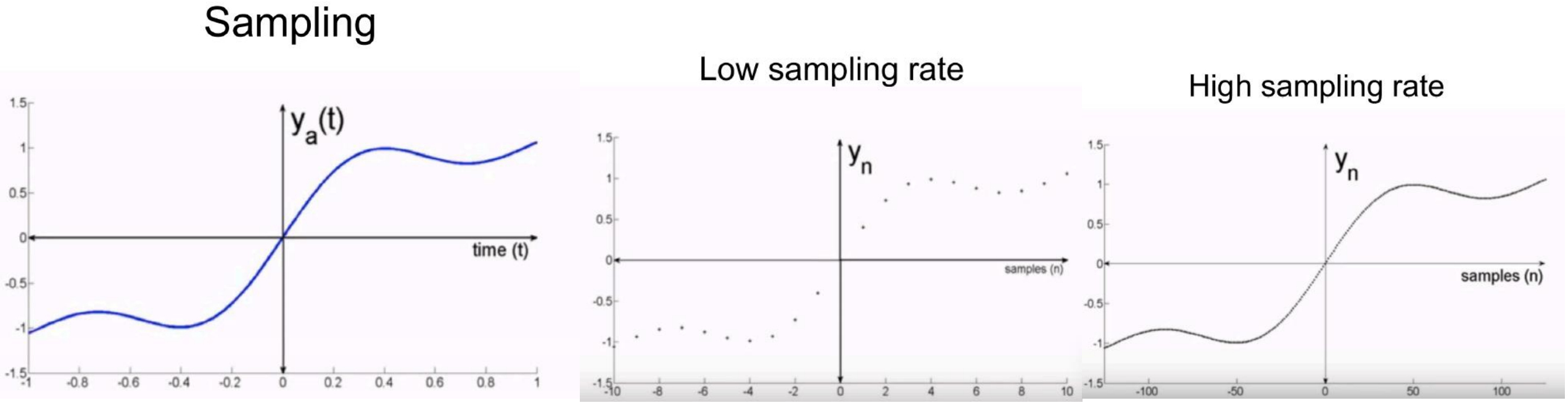
- Let's look at one line of pixels.
  How do we digitize them?
- We use two things:
  - **Sampling**
  - **Quantization**





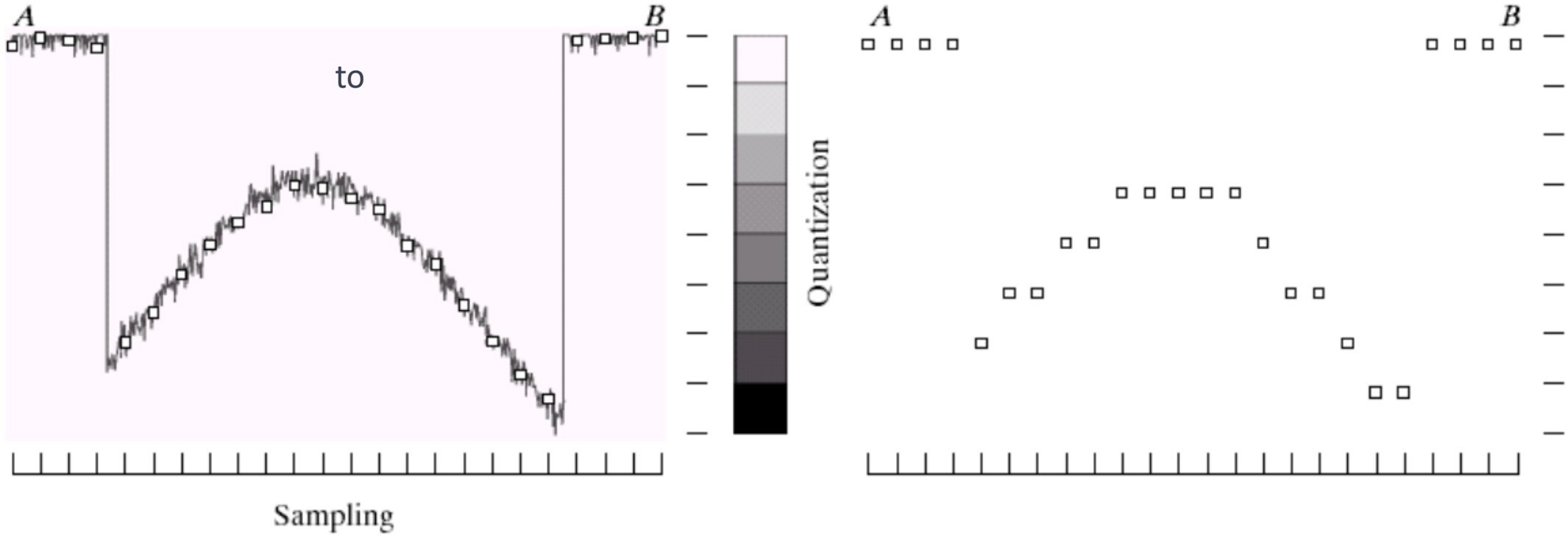
Sampling

- Sampling relates to resolution
  - how many pixels do we want to have?



Sampling

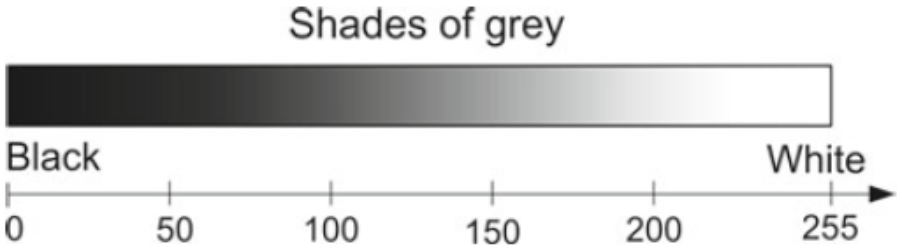
- Left - Sampling (we have sampled the line with a low frequency)
- Right - Quantization (Need to assign values to each point/pixel)

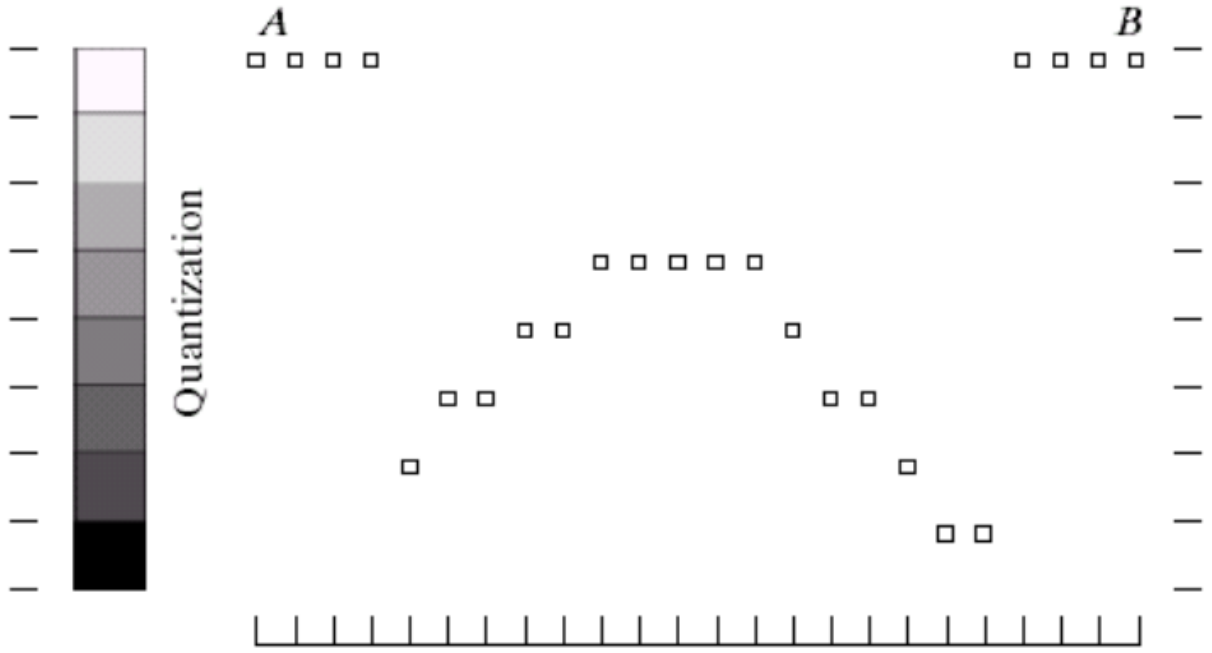


Quantization

- Quantization relates to the range of values we can assign to a pixel to describe its value (intensity)

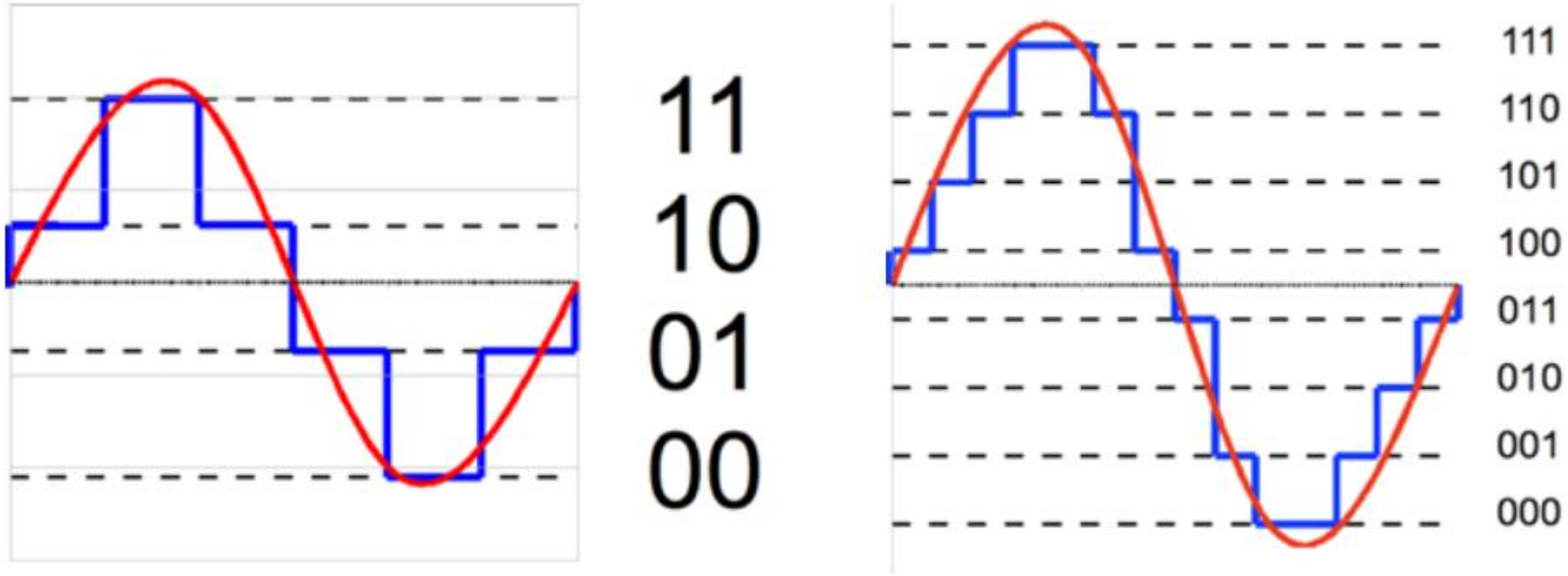
3 bits = 2^3 = 8 different colors





Quantization

- Digital signals are represented as binary numbers, the bit length of binary numbers used determines the quantization level
- The same is true for pixels



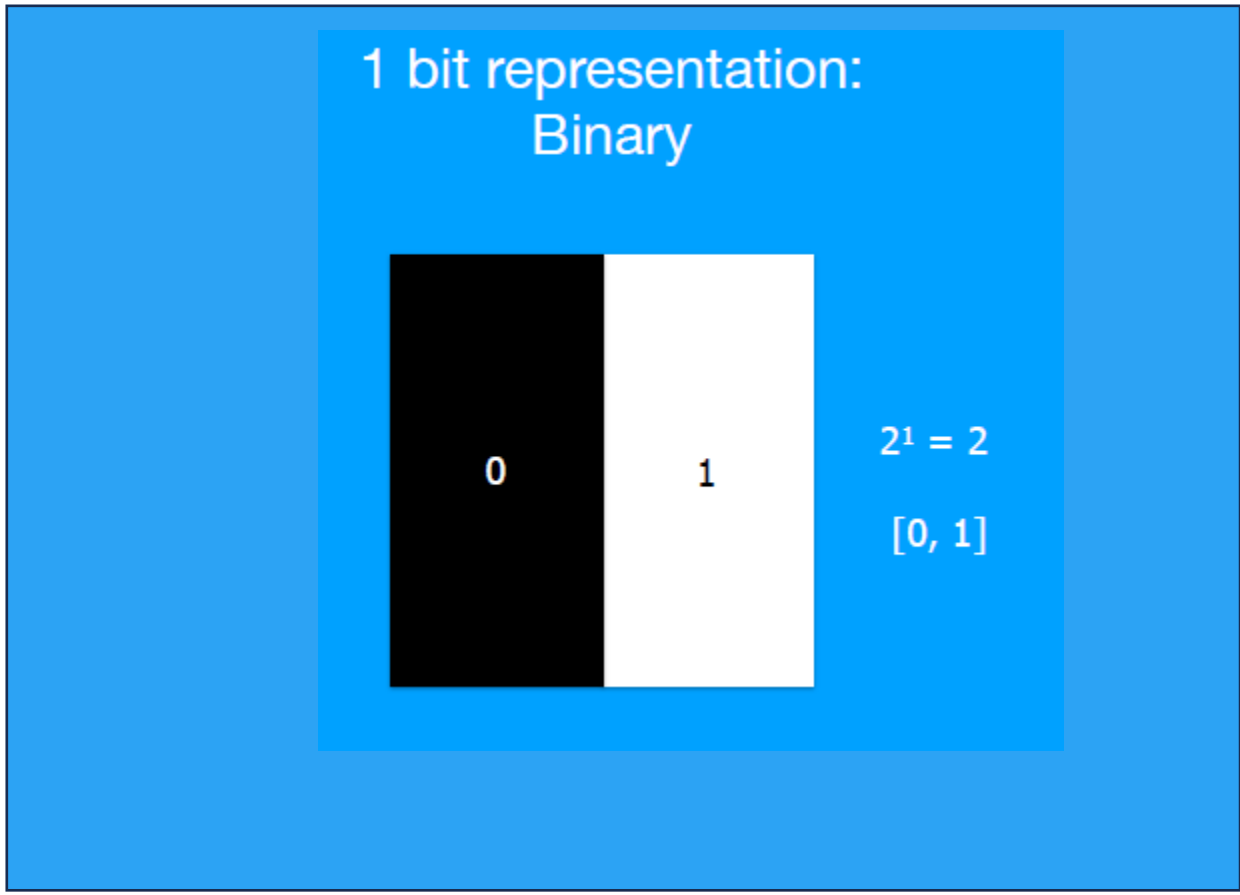
Quantization - Bits

Why are bits interesting?

- The human visual system cannot detect more than 256 different gray levels in an image available.
- Often this quantization results in a representation of one Byte (8 bits), since one byte corresponds to the way memory is organized inside a computer 0 and a high charge quantized to 255

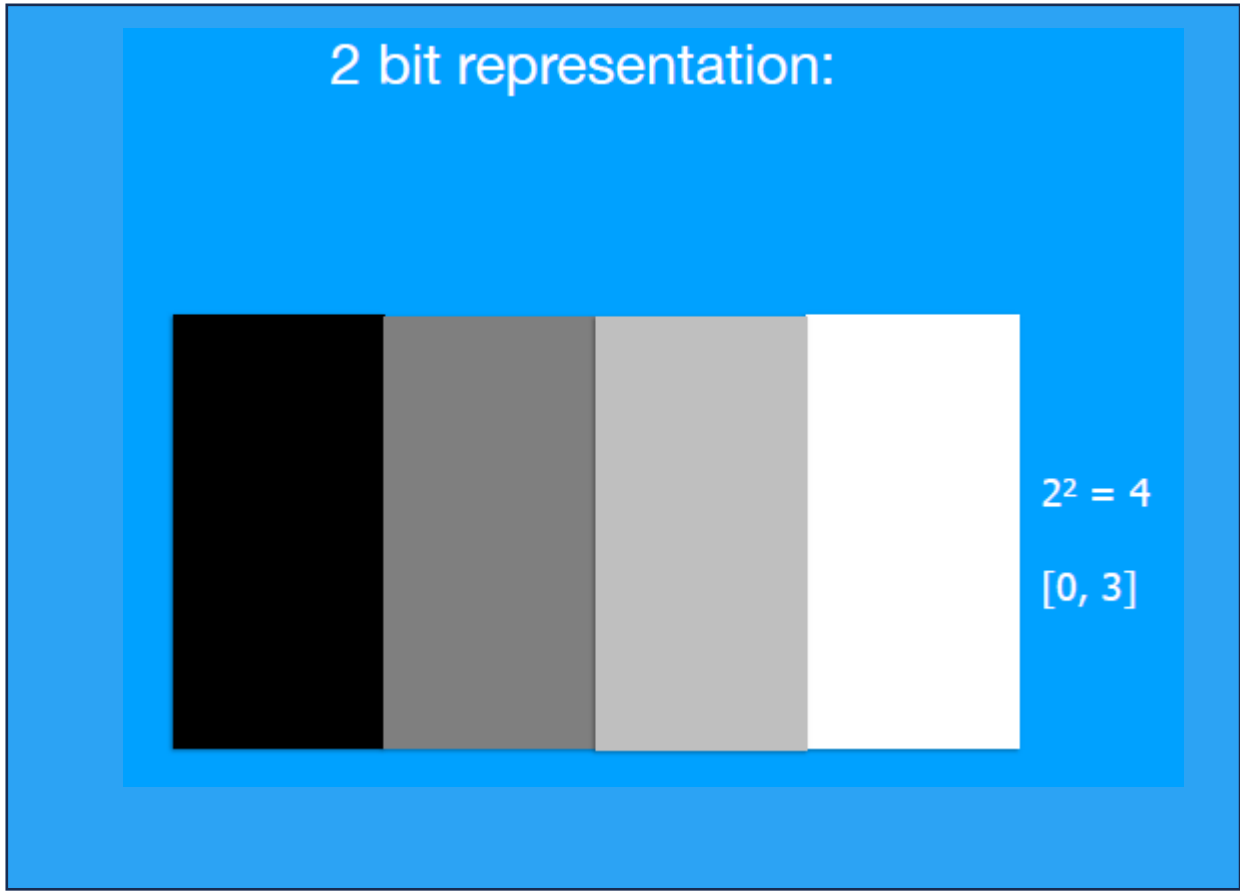
Quantization – Intensity level

- The number of intensity levels:
- $L = 2^k$
- Intensity levels are equally spaced
- They are integers in the interval [0, L-1]



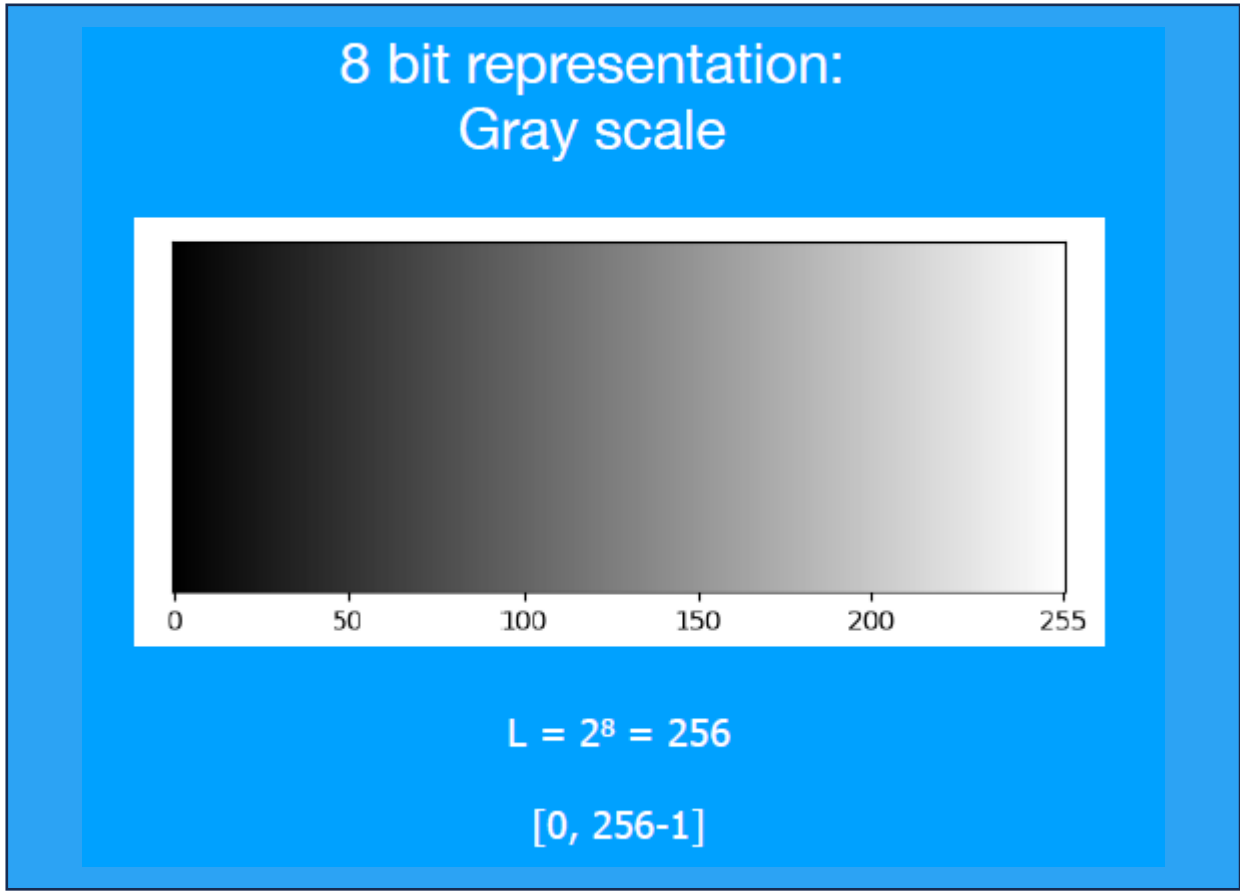
Quantization – Intensity level

- The number of intensity levels:
- $L = 2^k$
- Intensity levels are equally spaced
- They are integers in the interval [0, L-1]



Quantization – Intensity level

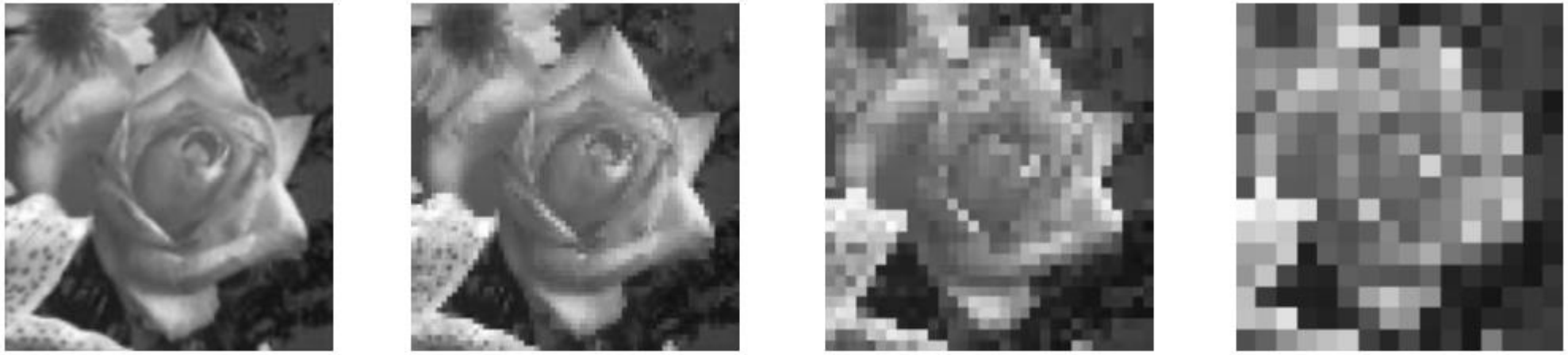
- The number of intensity levels:
- $L = 2^k$
- Intensity levels are equally spaced
- They are integers in the interval [0, L-1]



Sampling Examples

Sampling Examples

- The size of an area in a scene that is represented by the number of pixels in an image
- Different resolutions are possible (4000x300…256x256…16x16)



- Lower resolution leads to data reduction!

Quantization Examples

Summary of data formats

Format	Bytes per pixel	Black	White
uint8	1	0	255
uint16	2	0	65535
double	8	0.0	1.0

Image classes

There are four different image classes or encoding schemes:

- "**Binary**" images: number is either 0 or 1 indicating black or white. These are logical arrays.
- "**Grayscale/Intensity**" images: number indicates grayscale intensity (0 to 1, or 0 to 255).
- "**Indexed/labeled image**" images: pixel number is pointer (i.e., index) to an RGB color map. (Permits easy change of colorization schemes.)
- "**RGB**" images: number indicates intensity of either red, blue or green (0 to 1, or 0 to 255). Three separate arrays are required.

Question

How do you convert the **Grayscale/Intensity** images: number indicates grayscale intensity from 0-255 to 0-1?

Image types - grayscale

- Example of grayscale image of double format

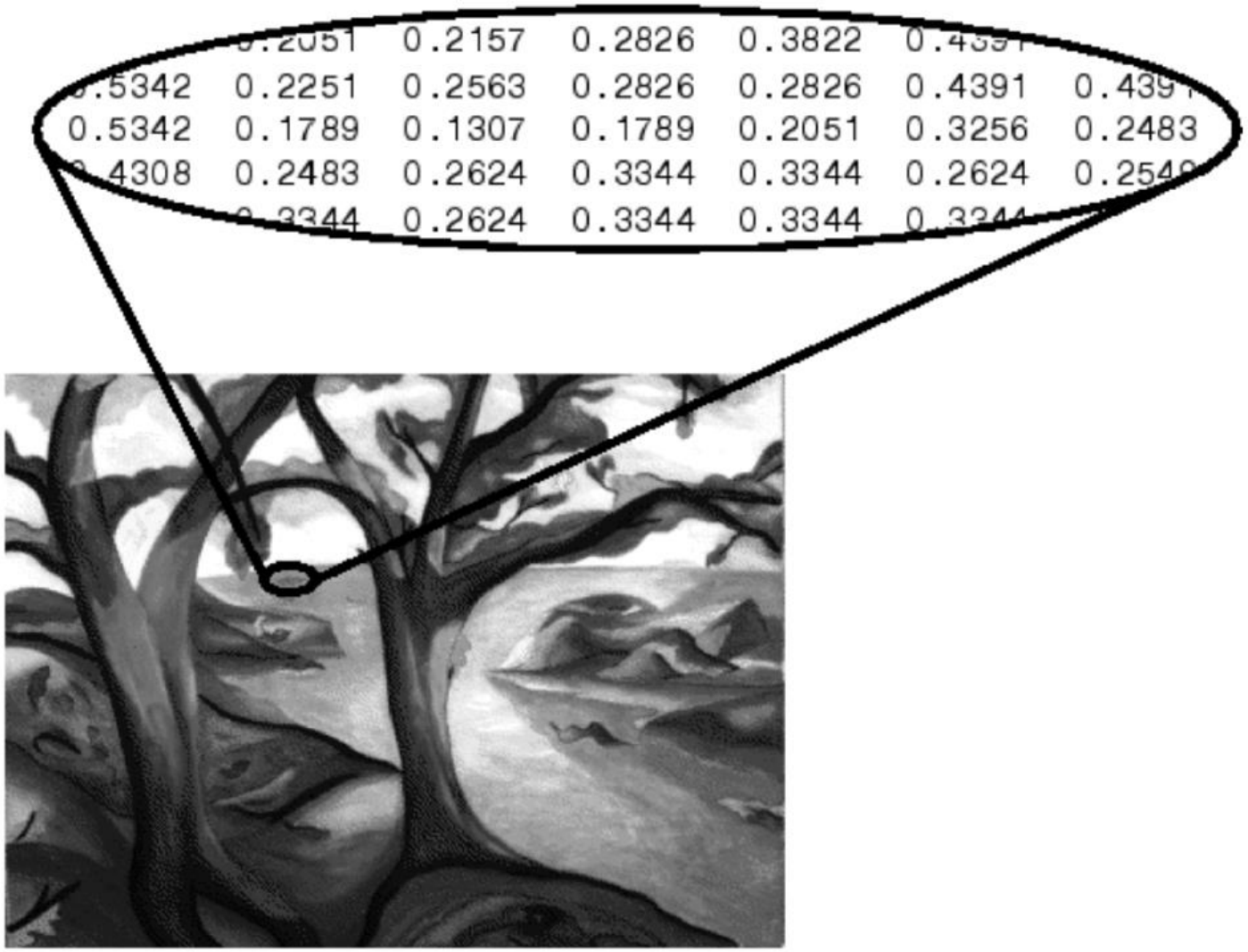
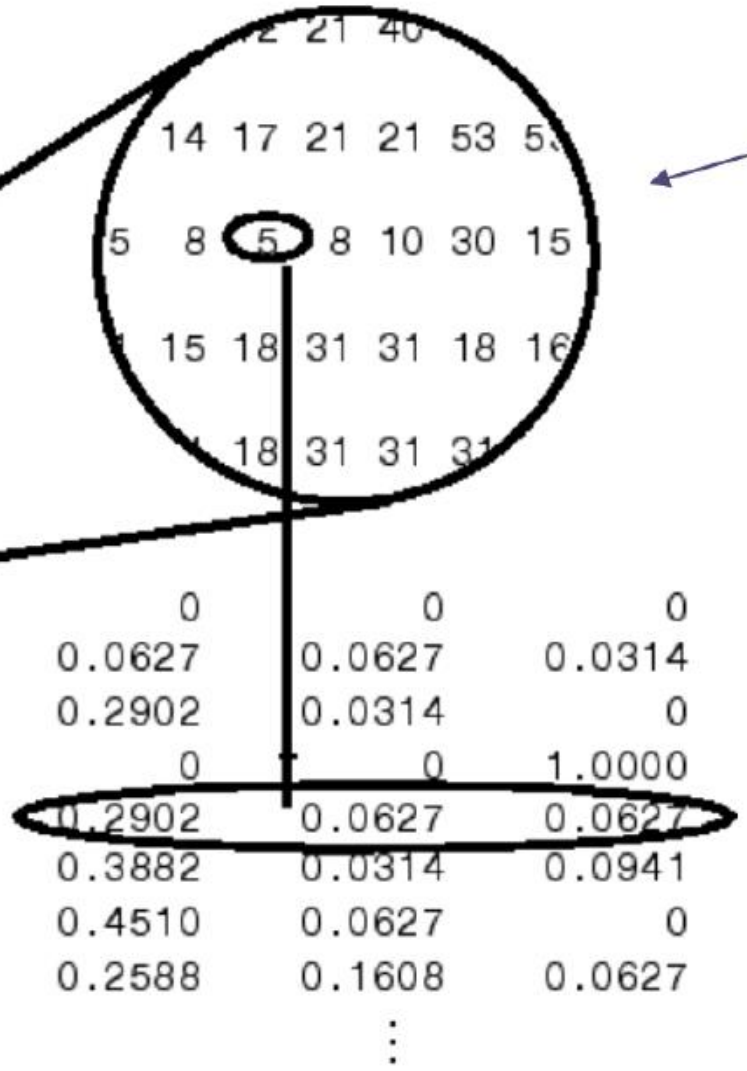


Image types – indexed image



Pixel values refer to index in colormap

Colormap

Image types – indexed image

- An indexed image needs an accompanying colormap
- For pixel operations, you need to convert to grayscale or RGB
  You may use predefined colormaps for visualization

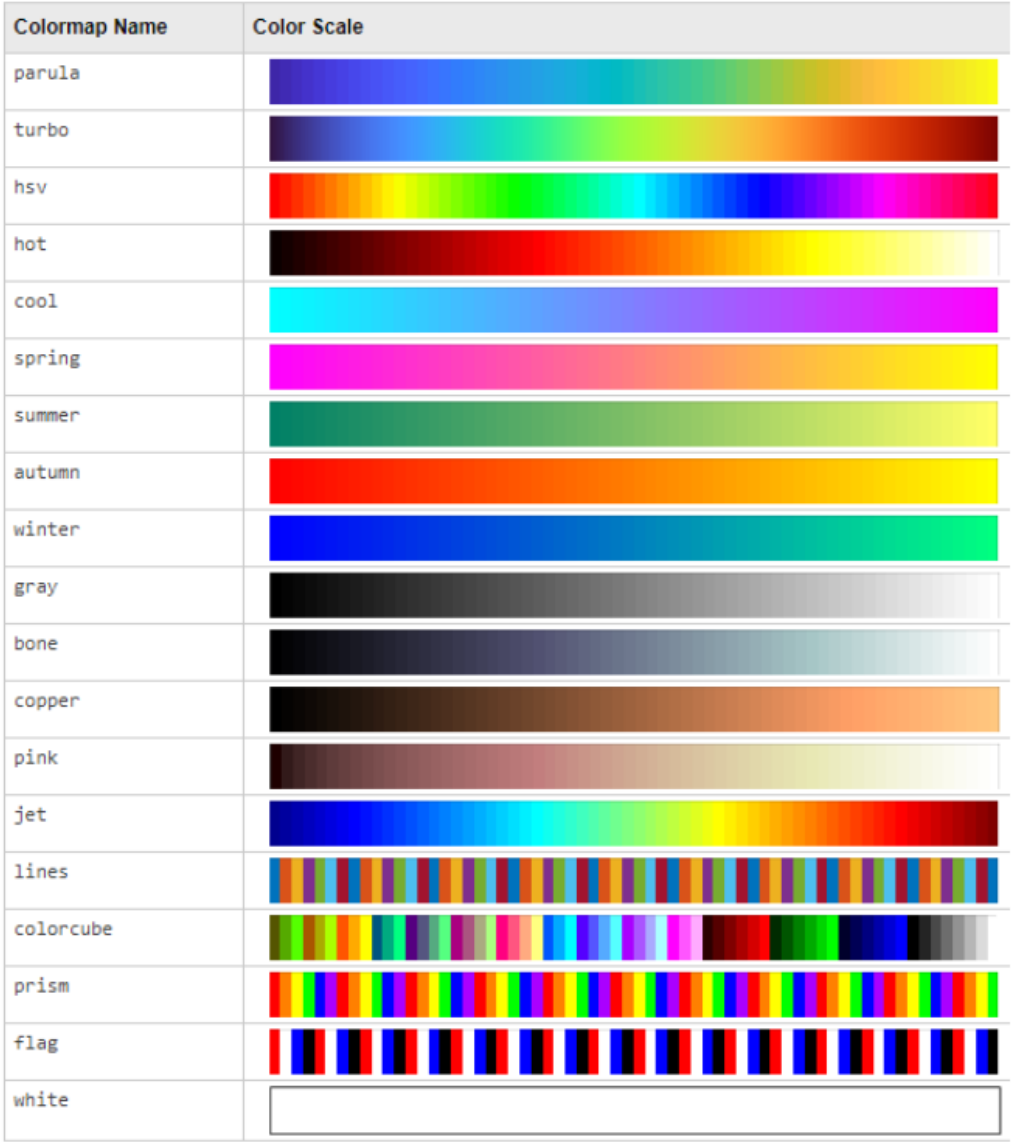


Image types – Binary

- Binary: Pure black and white pixels only
- They are often used as image "**masks**," (i.e., to isolate selected segments of an image.)
- Binary image variables consist of a '**logical**' array containing either 0's (black) or 1's (white).

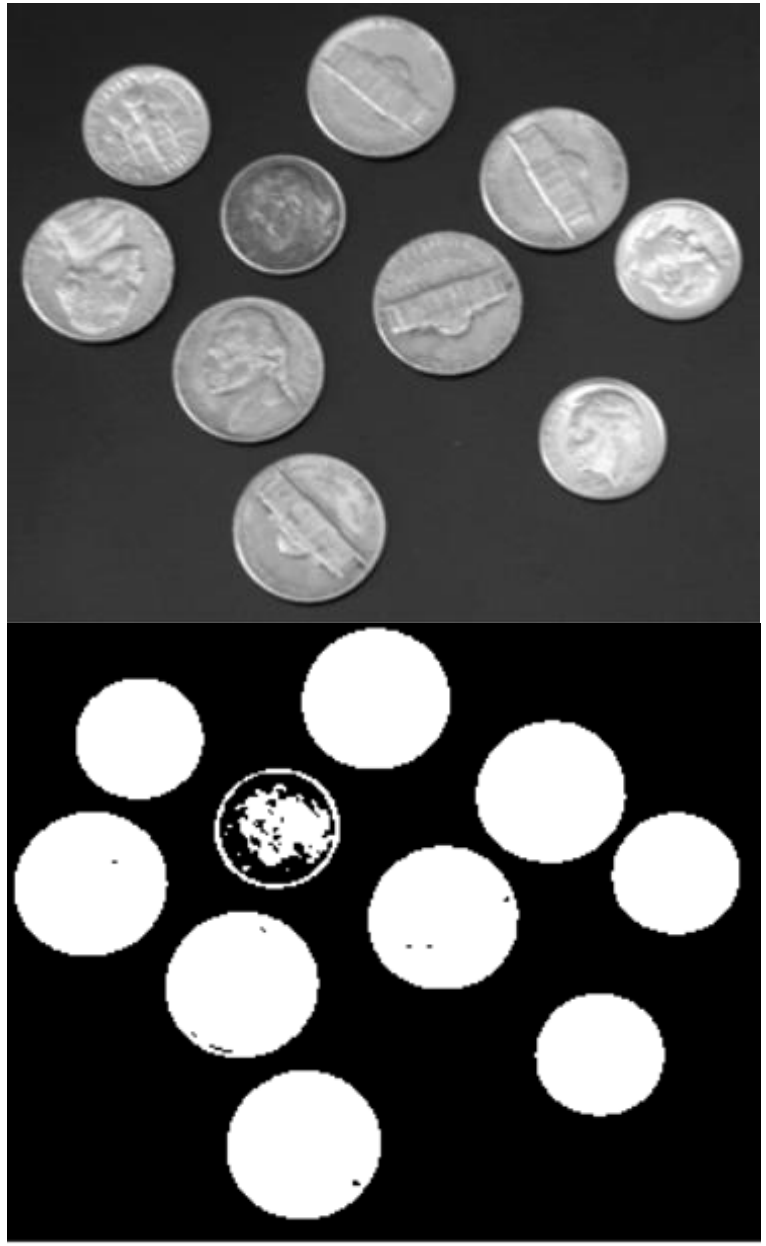
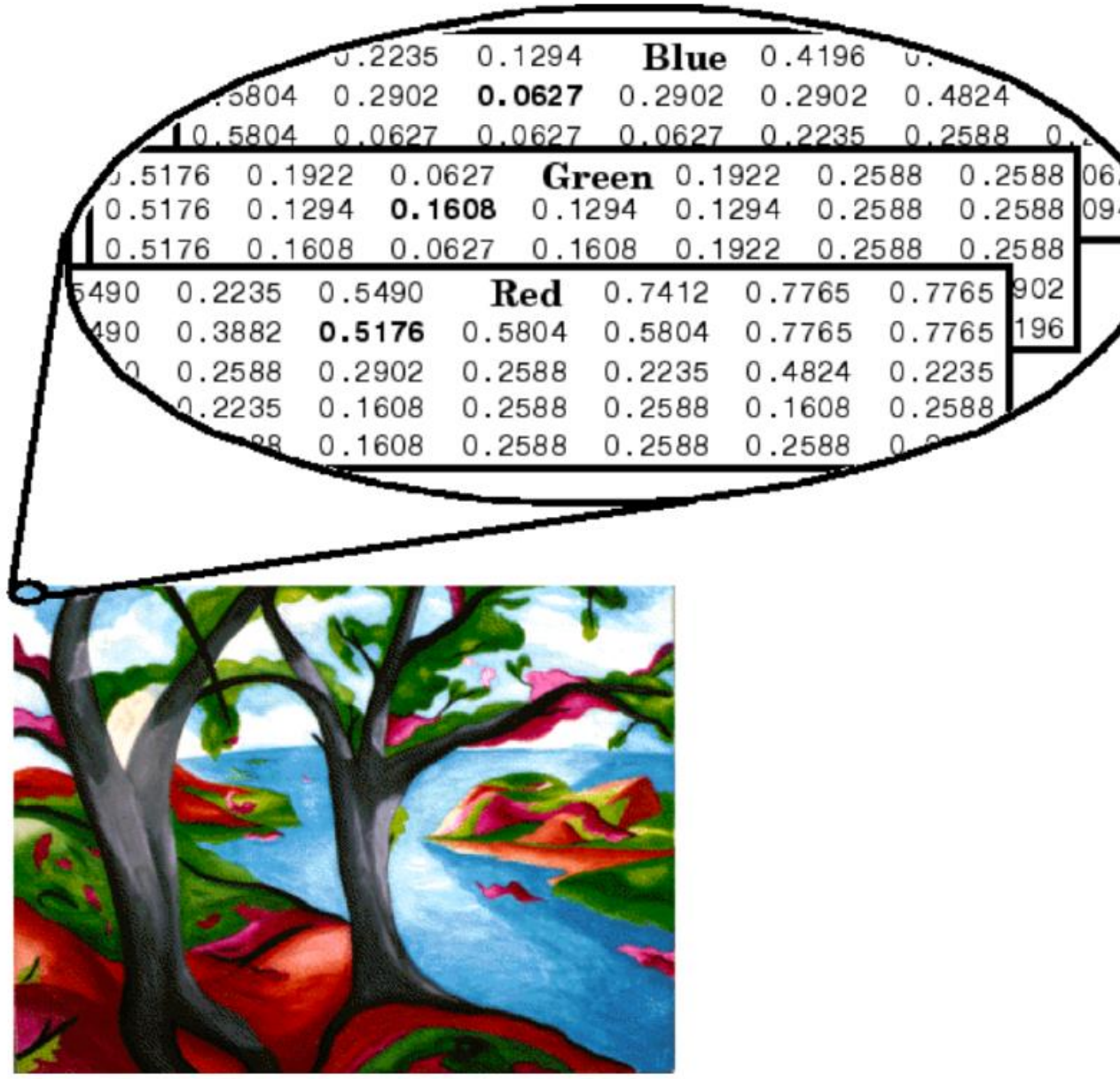


Image types – RGB

- Example of RGB image of double format.



Exercises

- All exercises for this course will be uploaded on moodle