Course
« *Computer Vision*»

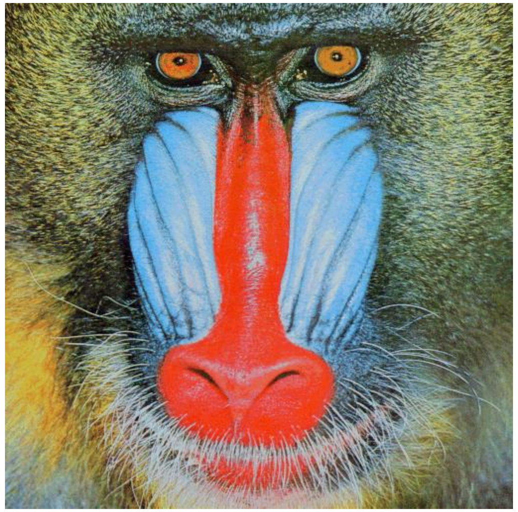
Sid-Ahmed Berrani

2024-2025

Images formation

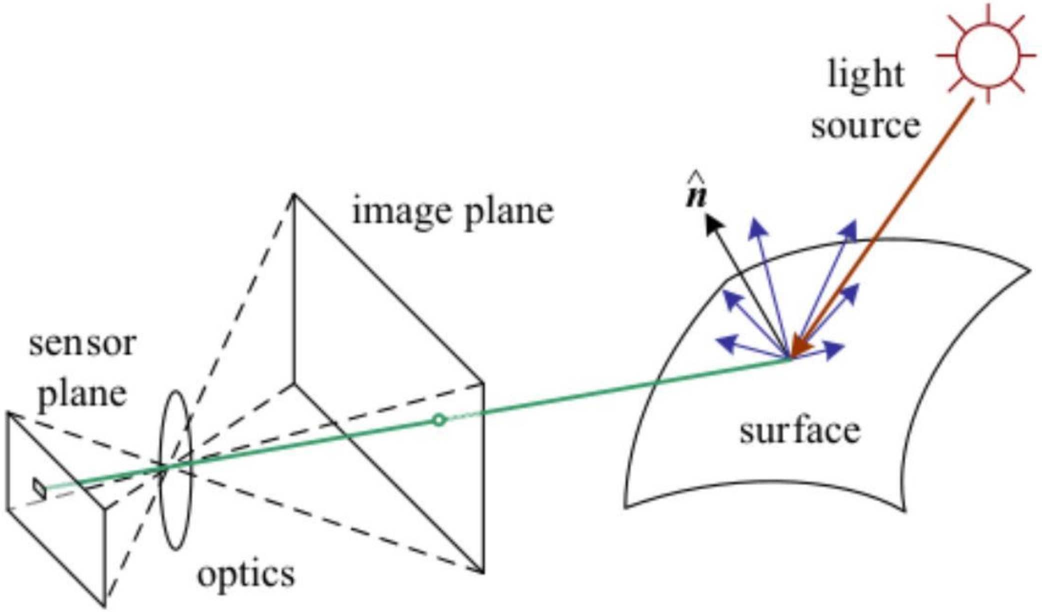
- Almost all the computer vision topics requires the concept of “image”
- In other terms, to apply computer vision techniques, we need images



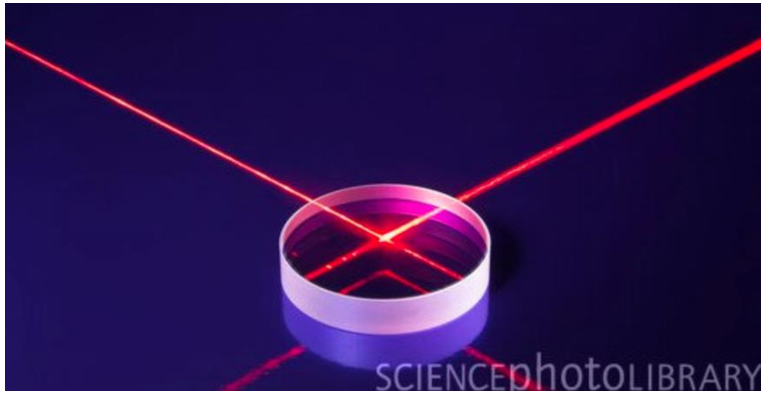



Images formation: All starts with light...

- To being able to see any 3D scene, our eyes or a digital camera needs to capture the light radiation reflected from scene surfaces
- To produce an image, the scene must be illuminated with one or more light sources



Images Formation: The nature of light



Light has a dual nature:

1. Can behave like a particle (photon)
   - Travels in a straight line
   - Light can reflect off a mirror and cast shadows
   - Consists of tiny bits of energy behaving like discrete packets

*A **photon** is an elementary particle that represents a quantum of light or other electromagnetic radiation. It has no mass, always moves at the speed of light in a vacuum, and carries energy and momentum.*

Images Formation: The nature of light

Light has a dual nature:

2. Can behave like a wave

- **Refraction:** This occurs when light changes direction as it passes from one medium to another due to a change in speed.

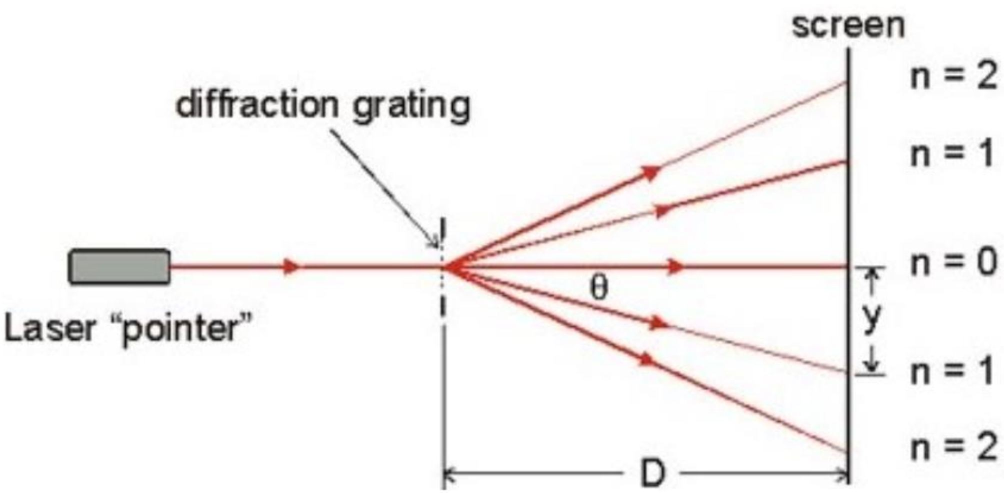
An example is how a straw appears bent when placed in a glass of water.

Images Formation: The nature of light

Light has a dual nature:

2. Can behave like a wave

- **Diffraction**: This happens when light bends around obstacles or spreads out after passing through a narrow slit.



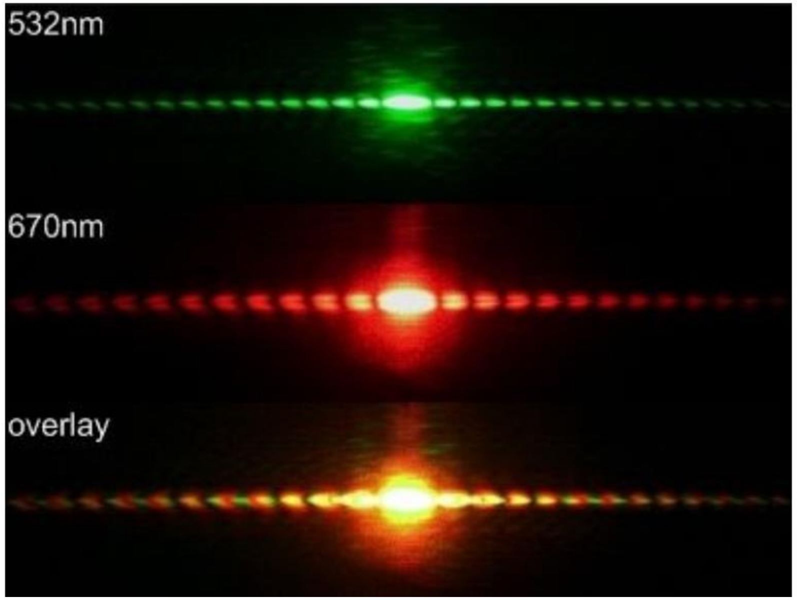
Images Formation: The nature of light

Light has a dual nature:

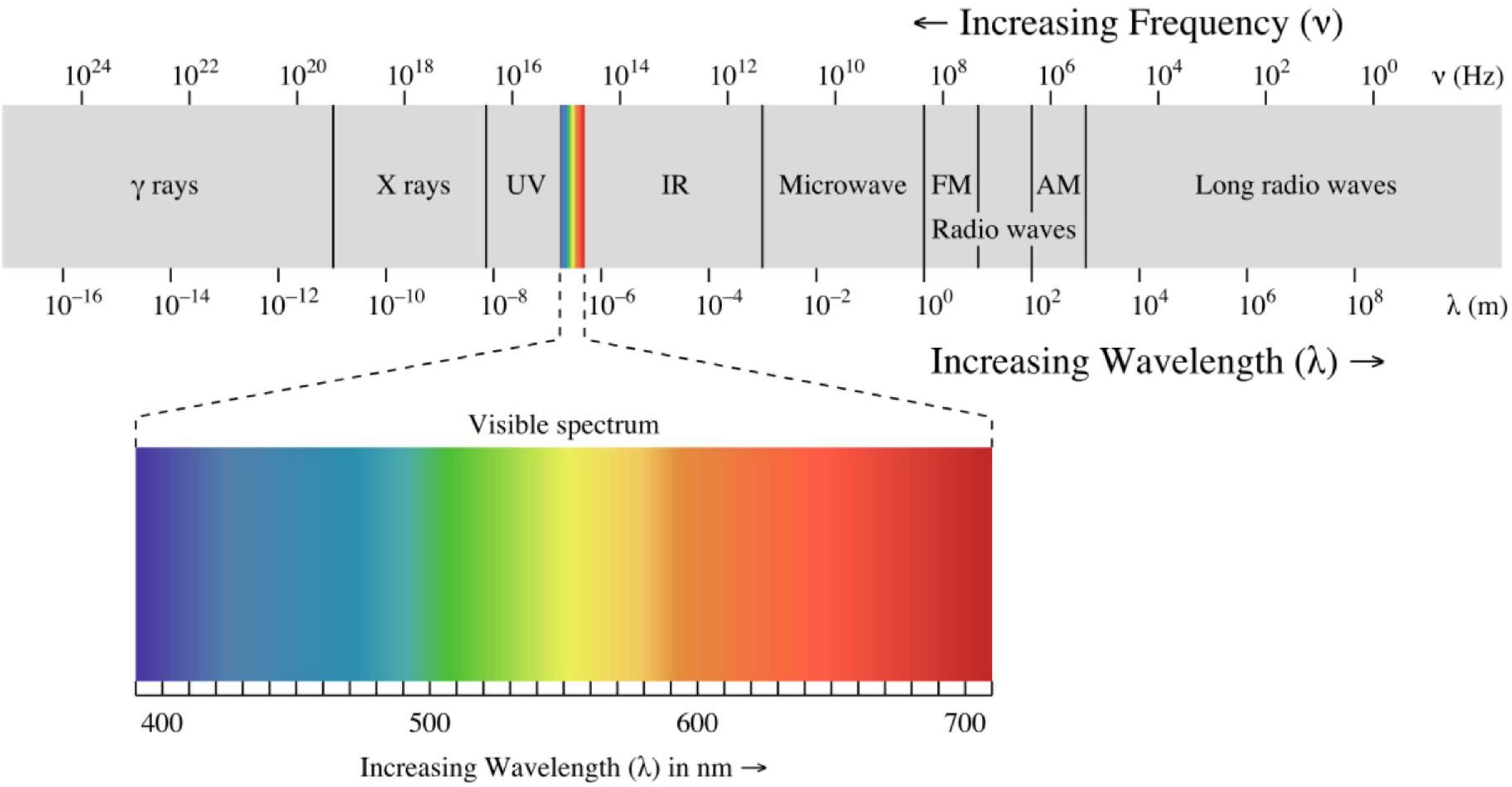
2. Can behave like a wave

- **Wavelength and Amplitude** – Light, as an electromagnetic wave, has a **wavelength** (the distance between two consecutive peaks) and an **amplitude** (which relates to the light's intensity or brightness).

Different wavelengths correspond to different colors in the visible spectrum.



Images Formation: Visible light is part of electromagnetic spectrum



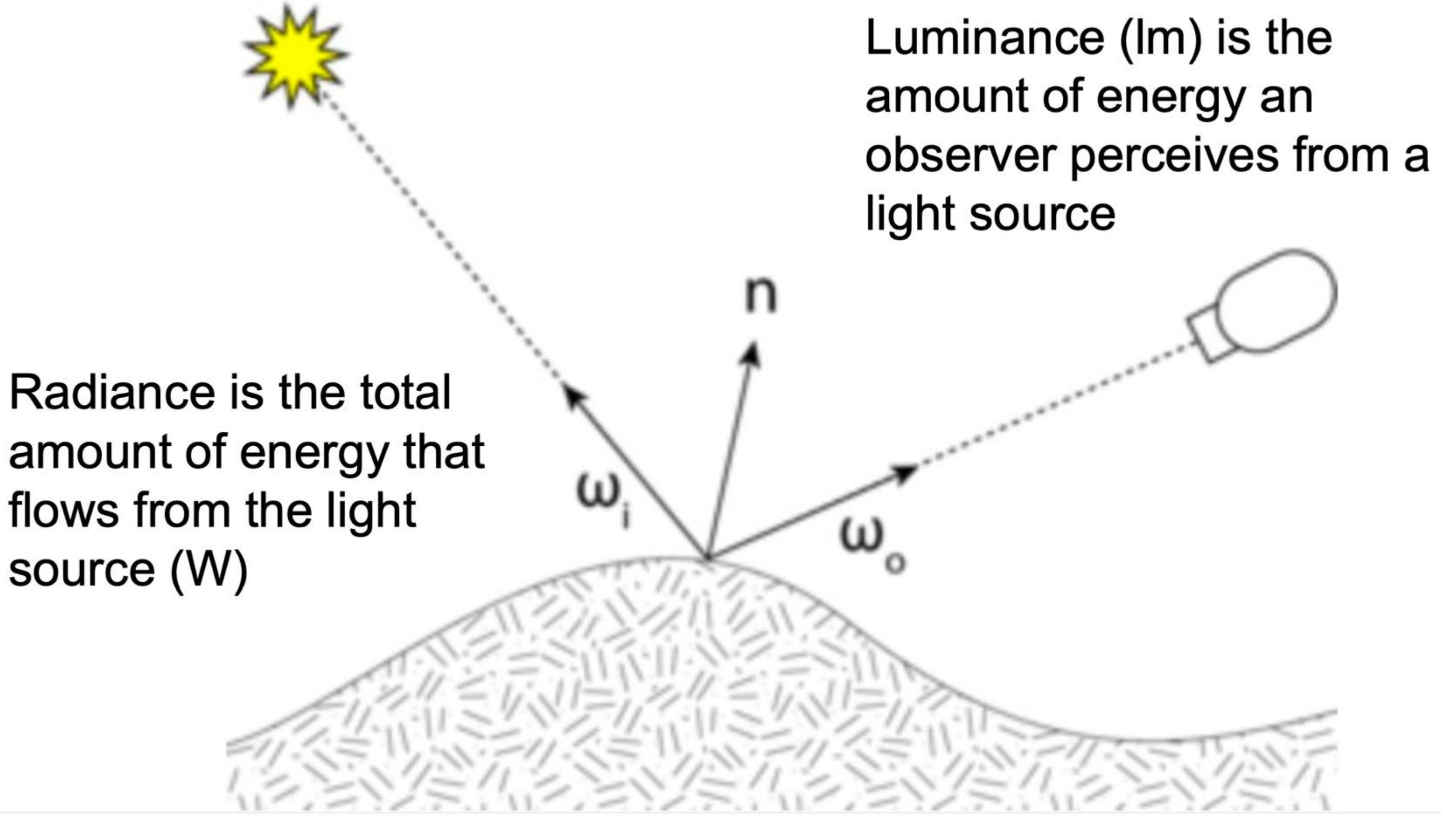
The electromagnetic spectrum can be expressed in terms of wavelength (**λ**) or frequency (**ν**)

Images Formation: Light and Perception

- The colors that humans perceive in an object are determined by the nature of the light reflected from the object:
  - A white object reflects light uniformly among visible wavelengths
  - Green objects reflect light primarily in the 500-570 nm range.
- Quantities that are usually used to describe a light source:
  - Radiance
  - Luminance
  - Brightness

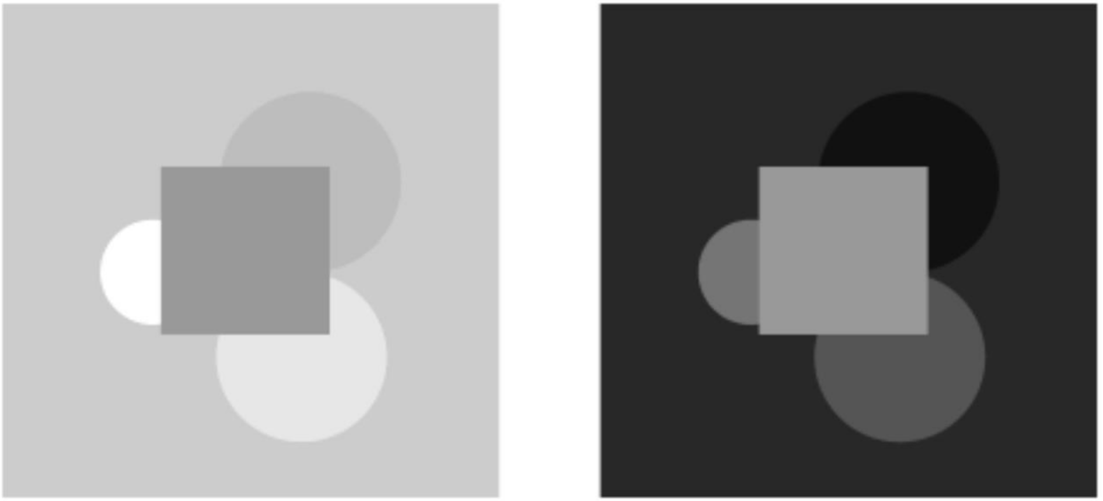
Images Formation: Light and Perception

- Quantities that are usually used to describe a light source:
  - Radiance
  - Luminance
  - Brightness



Images Formation: Light and Perception

- Quantities that are usually used to describe a light source:
  - Radiance
  - Luminance
  - Brightness

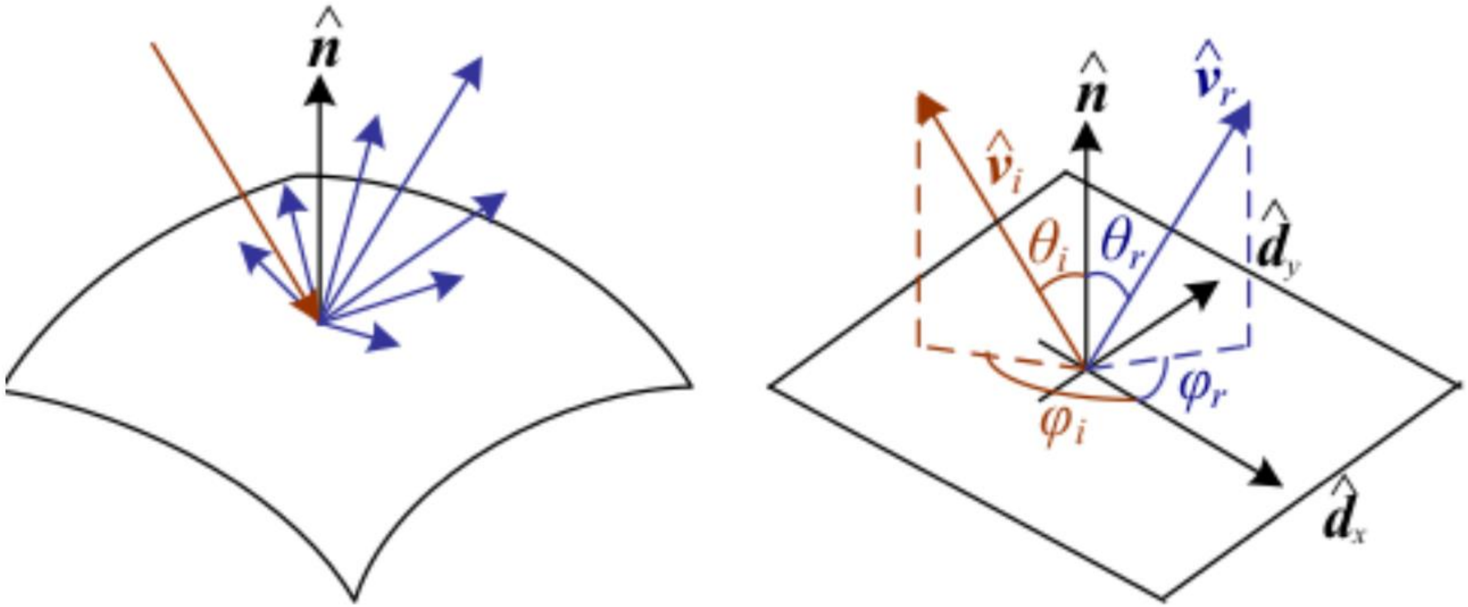


Brightness is a subjective descriptor of light “intensity” and is one of the key factors in describing color sensation.

Luminance of the inner square is exactly the same. Perceived brightness is different.

Images Formation: The BRDF model

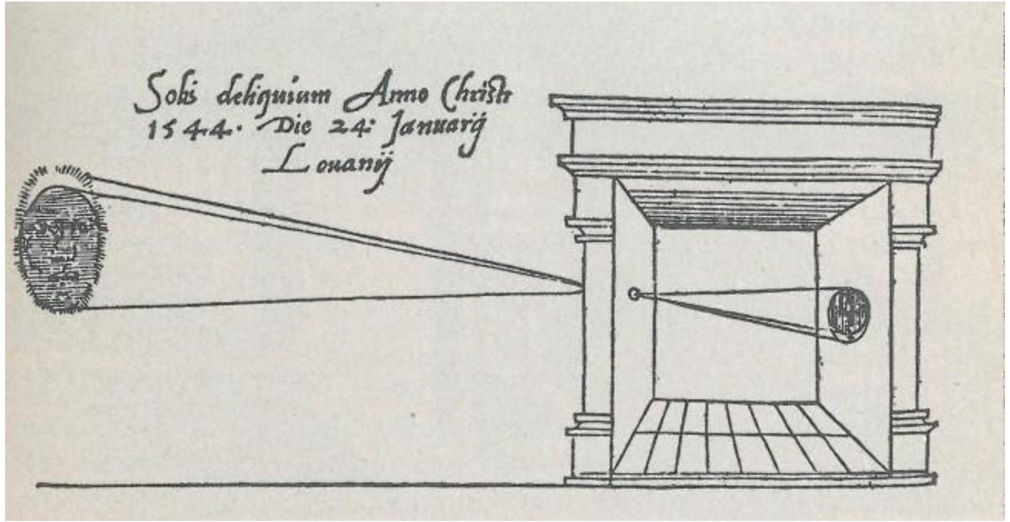
- When light hits a surface it is scattered and reflected. The most general way to model this interaction is through a 5-dimensional function called BRDF (Bidirectional Reflectance Distribution Function).



$$f_r(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)$$

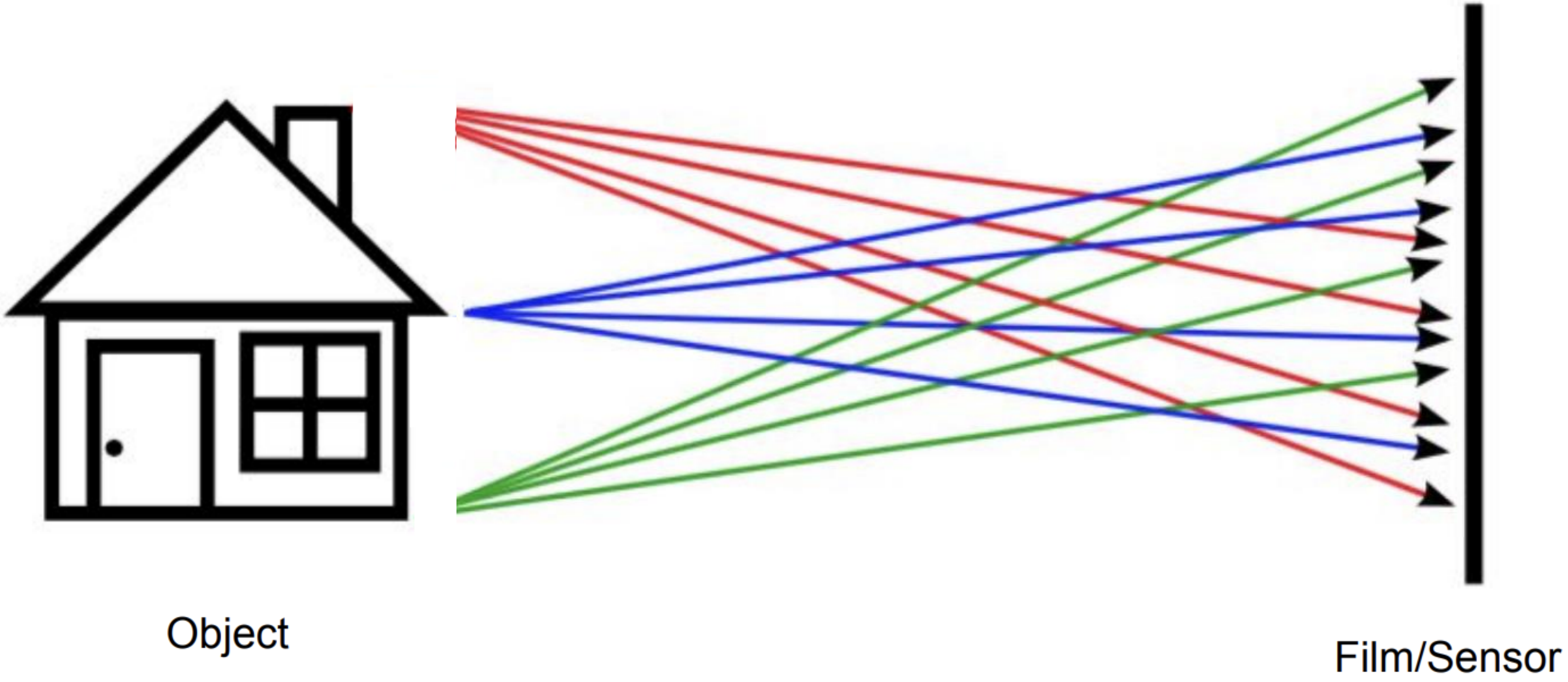
Images formation: Cameras

- Image are captured using cameras
- The 1st models of camera obscura was invented in the 16th century.
- Camera obscura (dark chamber) does not have lenses - It has a *pinhole*.



Images formation: The Pinhole Camera Model

- **Imaging device:** Let's try to build a simple image device

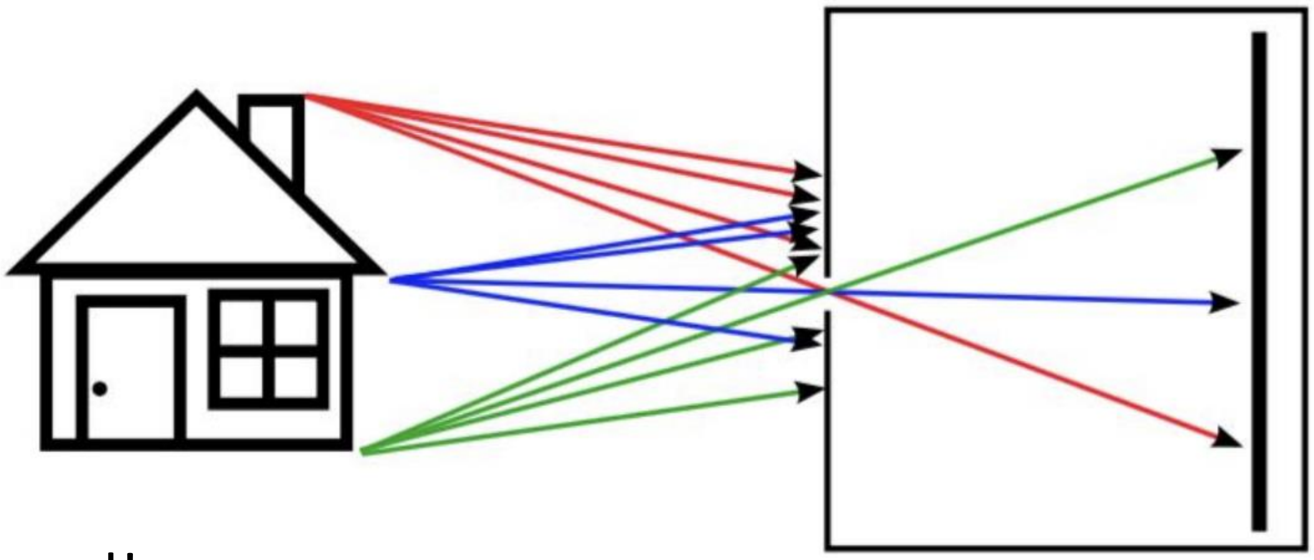


We are unable to get a reasonable image. Why ?

Images formation: The Pinhole Camera Model

- Camera obscura:

Key idea: Put a barrier with a small hole (aperture) between the object and the sensor

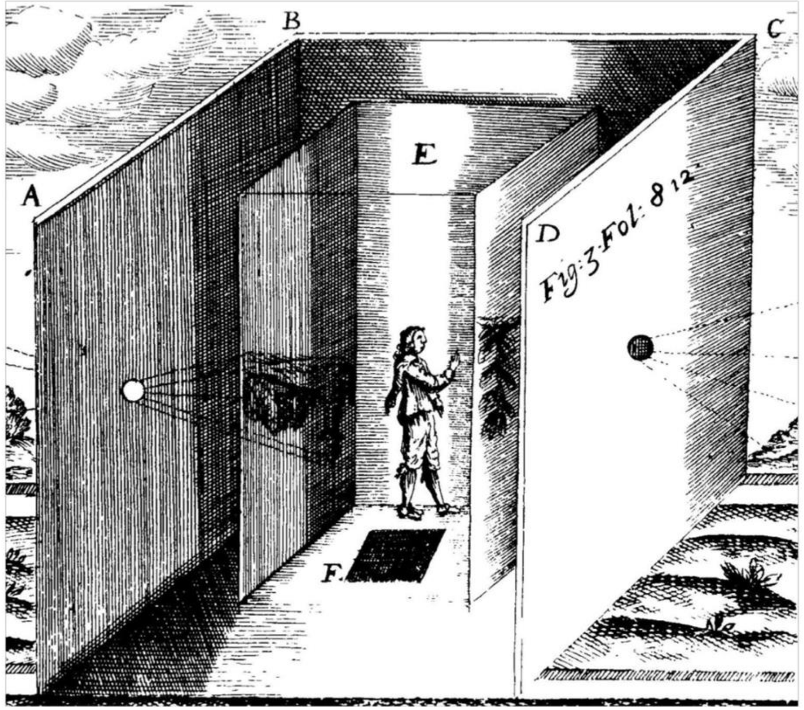


Blurring is reduced!
But the aperture should be as small as possible.
This is also known as pinhole camera.

Images formation: The Pinhole Camera Model

- Camera obscura:

  Leonardo da Vinci (1452–1519), after an extensive study of optics and human vision, wrote the oldest known clear description of the camera obscura in mirror writing in a notebook in 1502.



Images formation: The Pinhole Camera Model

- **Camera Obscura:**

*If the facade of a building, or a place, or a landscape is illuminated by the sun and **a small hole** is drilled in the wall of a room in a building facing this, which is not directly lighted by the sun, then **all objects illuminated by the sun will send their images through this aperture** and will appear, upside down, on the wall facing the hole. You will catch these pictures on a piece of white paper, which placed vertically in the room not far from that opening, and you will see all the above-mentioned objects on this paper in their natural shapes or colors, but t**hey will appear smaller and upside down, on account of crossing of the rays at that aperture**. If these pictures originate from a place which is illuminated by the sun, they will appear colored on the paper exactly as they are. The paper should be very thin and must be viewed from the back.*

Images formation: The Pinhole Camera Model

- Camera obscura: How small must be the pinhole?

  Large pinhole:

  Rays are mixed up -> Blurring!

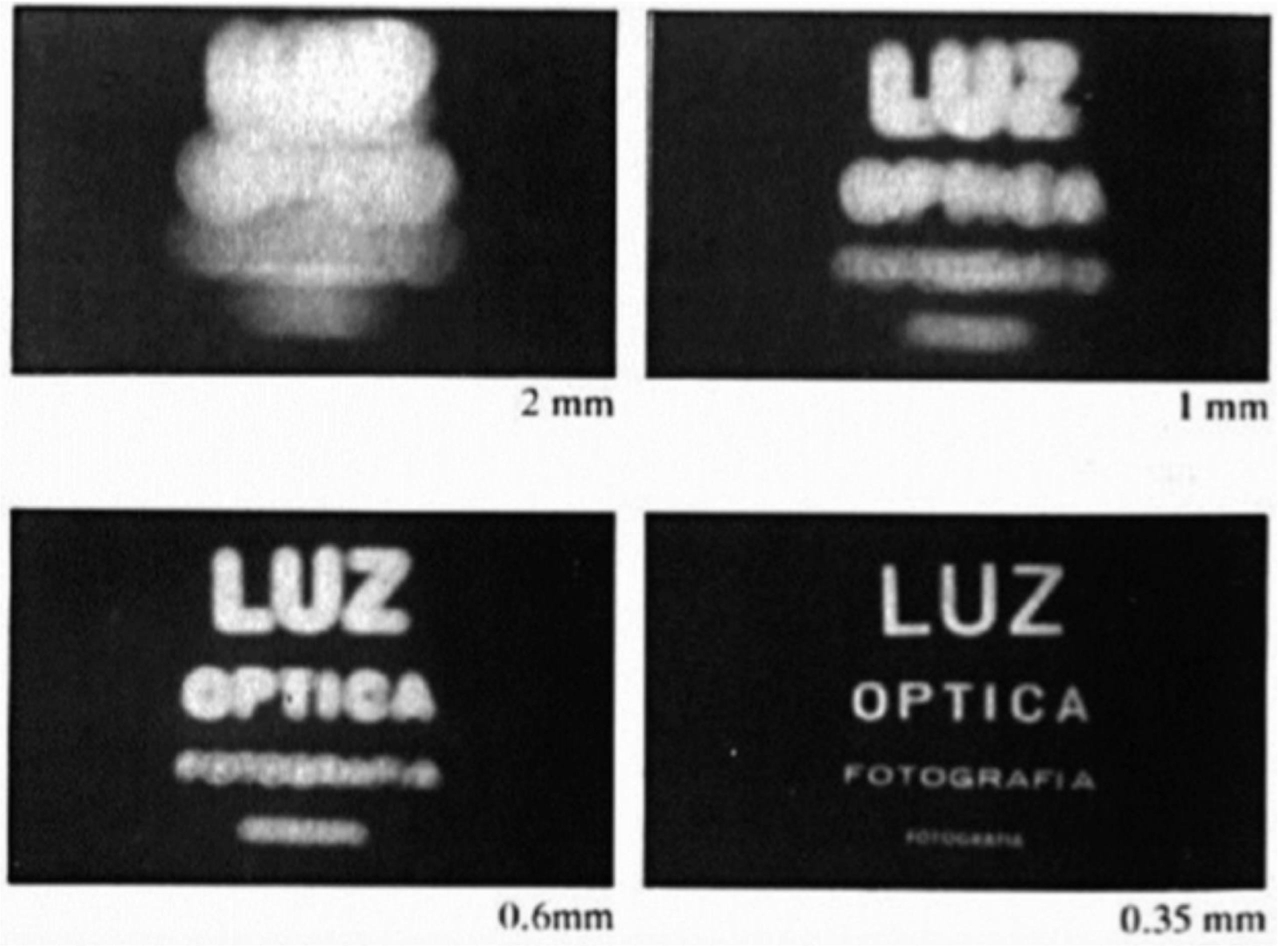
  Small pinhole:

  We gain focus, but

  - Less light passes through (long exposure time)
  - Diffraction effect (we lost focus again!)

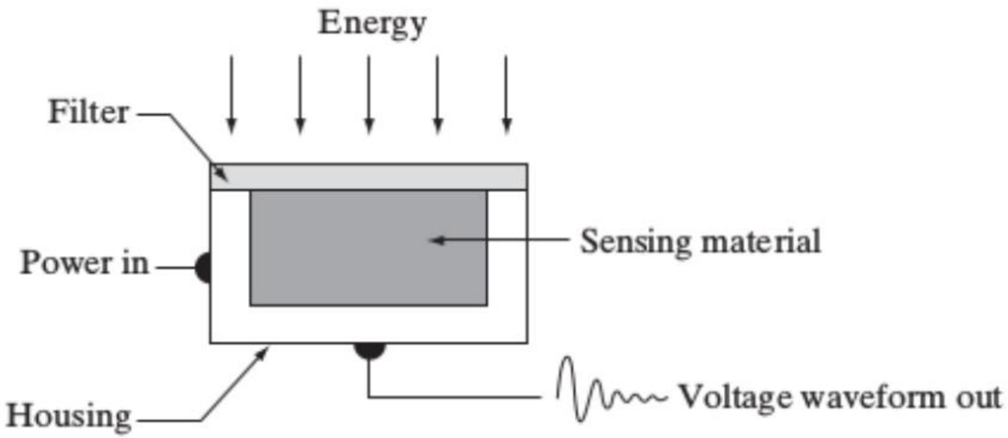
Images formation: The Pinhole Camera Model

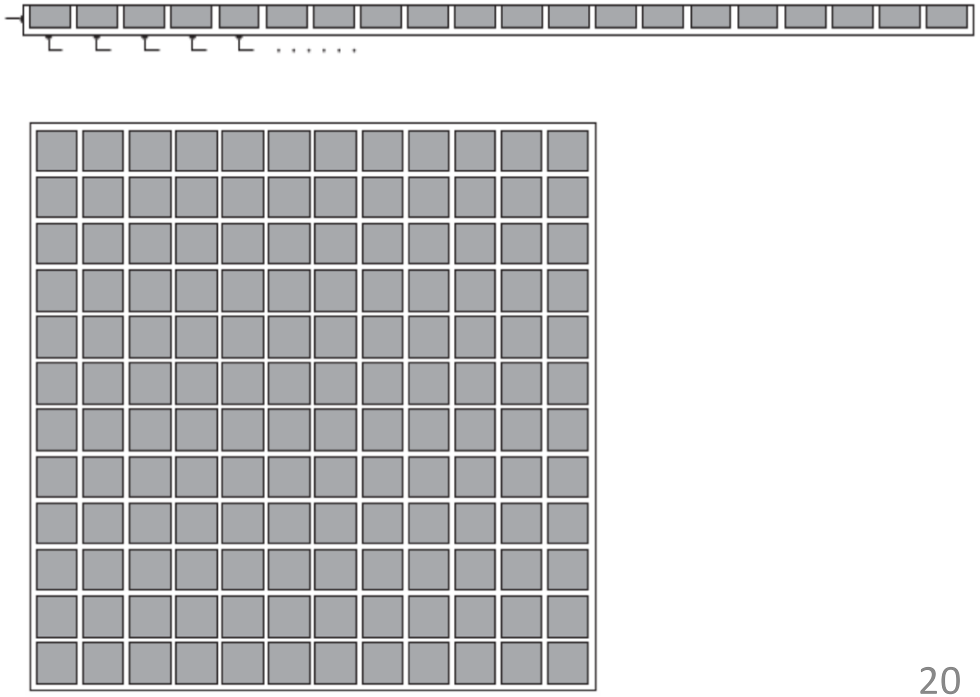
- Camera obscura: How small must be the pinhole?



Images Formation: Digital image sensing

- Incoming light radiation reflected from a 3D scene is transformed to a voltage by an imaging sensor that is sensible to a specific type of energy (wavelength).
- Usually, sensors are arranged in linear or 2D arrays

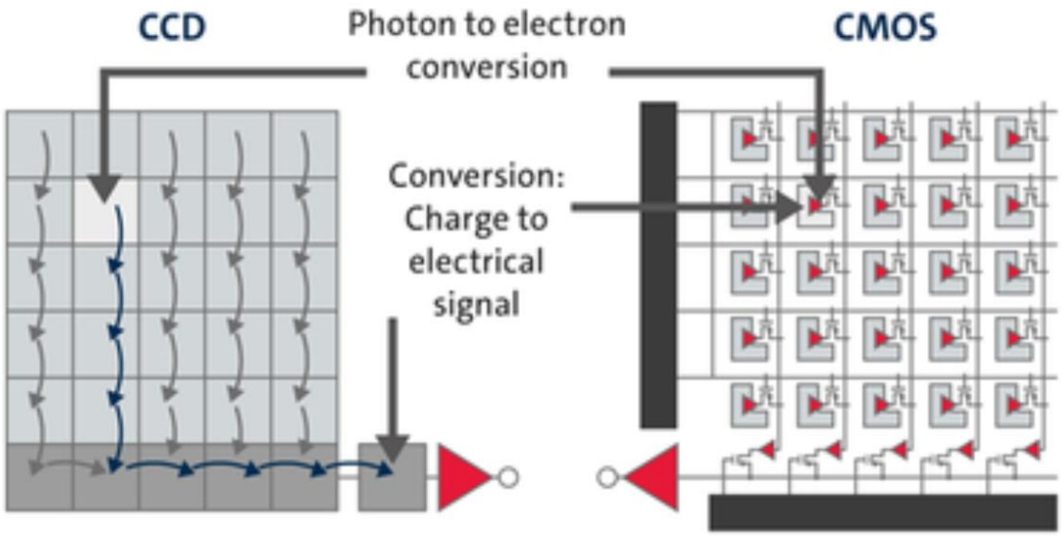




Images Formation: Digital image sensing

- Light falling on an imaging sensor is usually picked up by an active sensing area, integrating for the duration of the exposure, and then passed to a set of sense amplifiers.

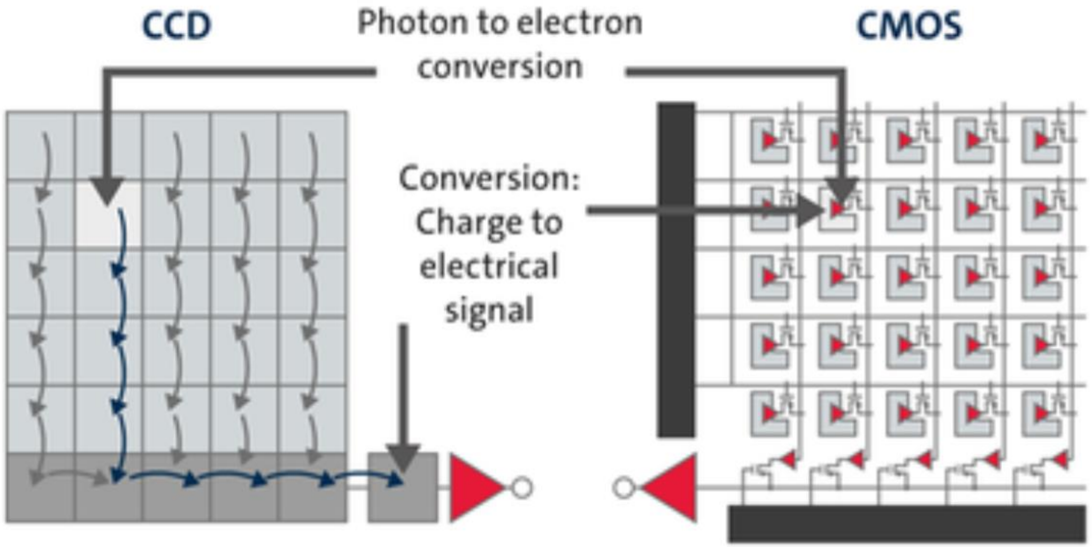
- The two main kinds of sensors used in digital steel and video cameras today are charged-couple of the device CCD and complementary metal oxide on silicon CMOS.



Two different principles: CCD vs CMOS

Images Formation: Digital image sensing

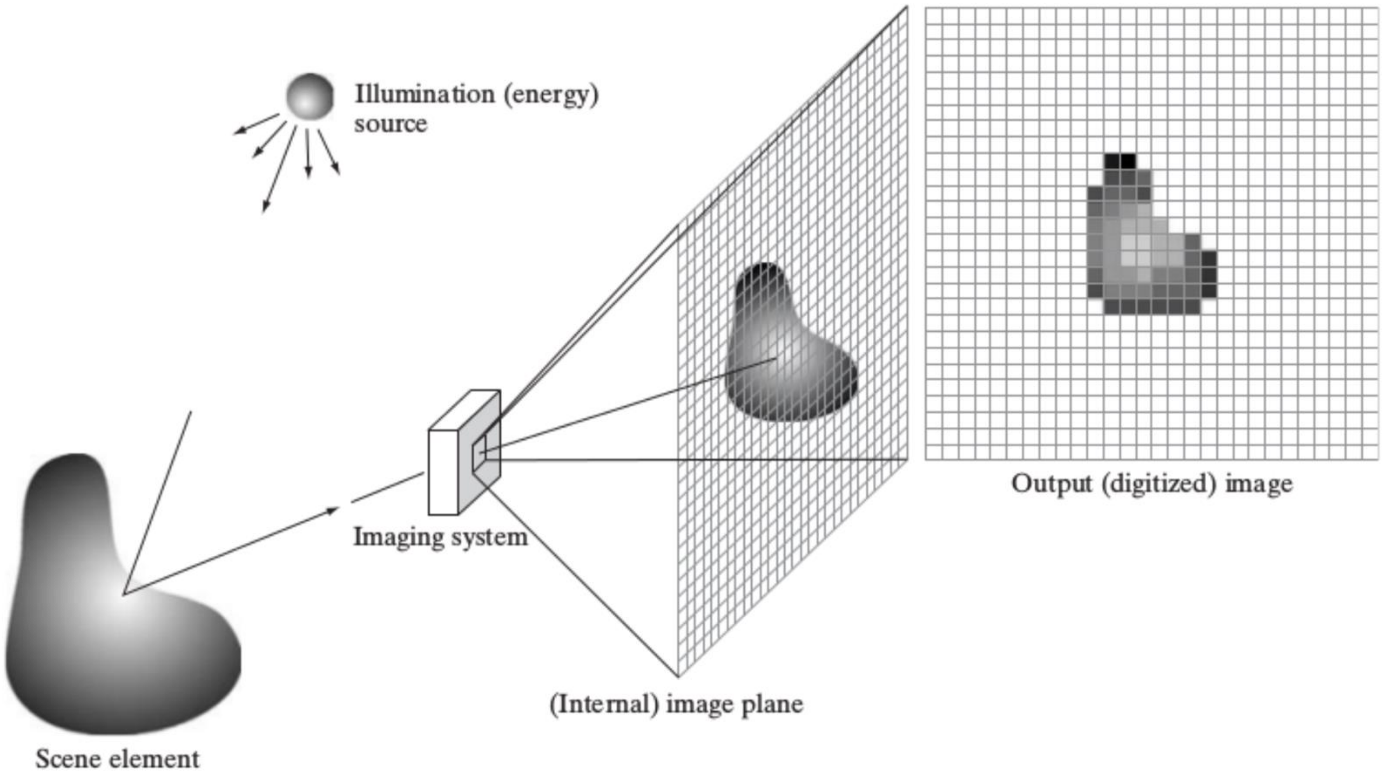
- A/D conversion from voltage to a digital signal can happen in 2 ways: At the end of each row/column (CCD) or directly at each sensing cell (CMOS).
- CMOS is in general better but CCD is faster and works better in low light.
- Today, CMOS is used in most digital cameras



Two different principles: CCD vs CMOS

Images Formation Process

- The sensor array, coincident with the focal plane of the lens, produces outputs proportional to the integral of the light received at each sensor for a certain amount of time.



Images Formation: Image Function

- An image can be modelled as a function:

$$I : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$$

The domain is a (usually rectangular) subset of the real image plane.

$I\,(x, y)$ is proportional to the amount of light energy that is collected at the image plane at coordinates x, y. The amount of energy is called **intensity**.

Images Formation: Sampling and Quantization

- A continuous real image is converted into a digital one through a process of ***sampling*** and ***quantization***.
- **Sampling**: Reduces the image domain to a finite set of spatial coordinates.

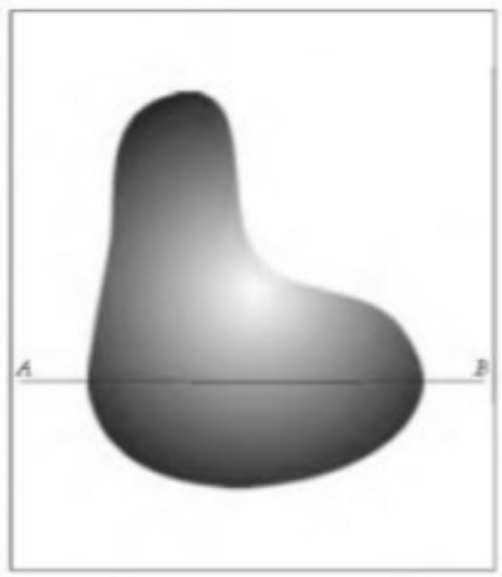
$$\mathbb{R}^2 \longrightarrow M \times N$$

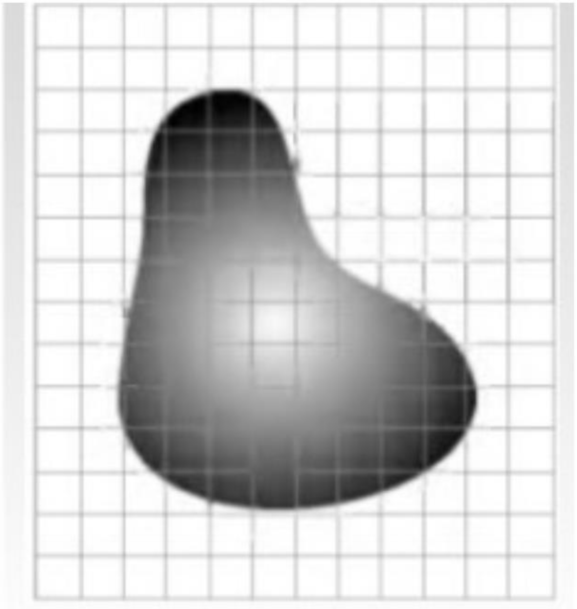
- **Quantization**: Reduces the sensor response (function codomain) to a finite set of values.

$$\mathbb{R} \longrightarrow [0, 1 \ldots 2^b - 1]$$

Images Formation: Sampling

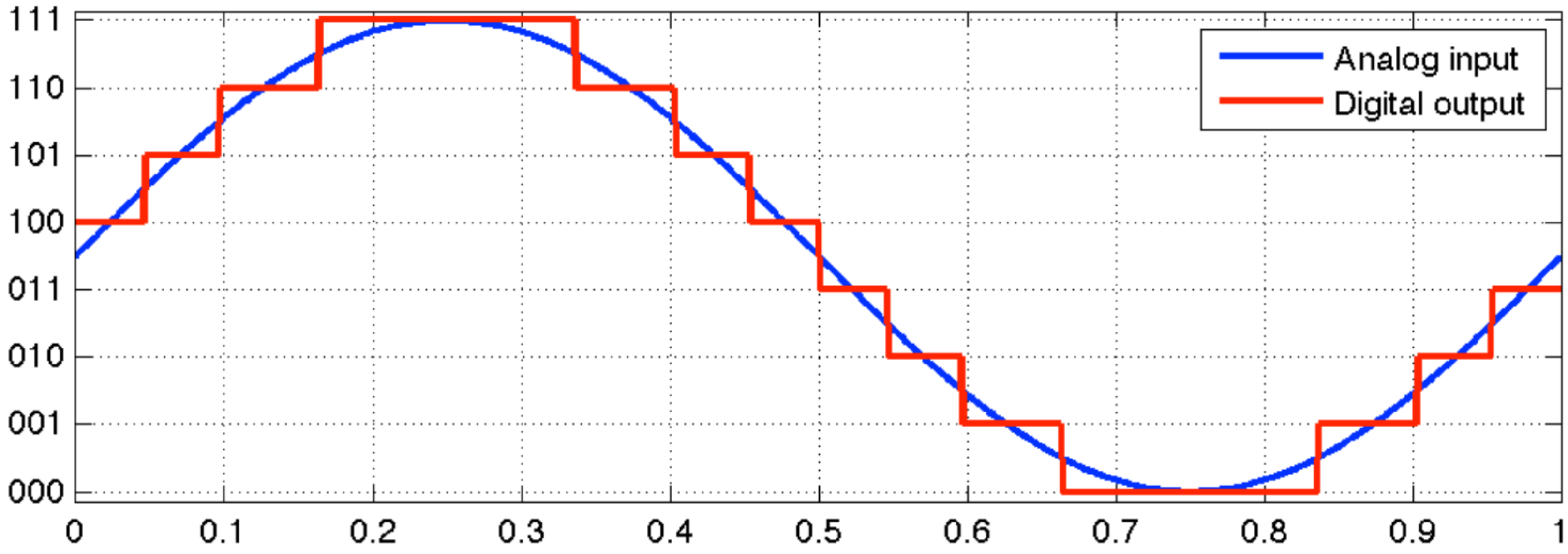
- To sampling the continuous image domain into a finite set of values we usually consider an equi-spaced grid of values in a given area (matrix).
- This reflect the regular arrangement of cells in a CMOS or CCD sensor.
- Each sample is called a **pixel**.



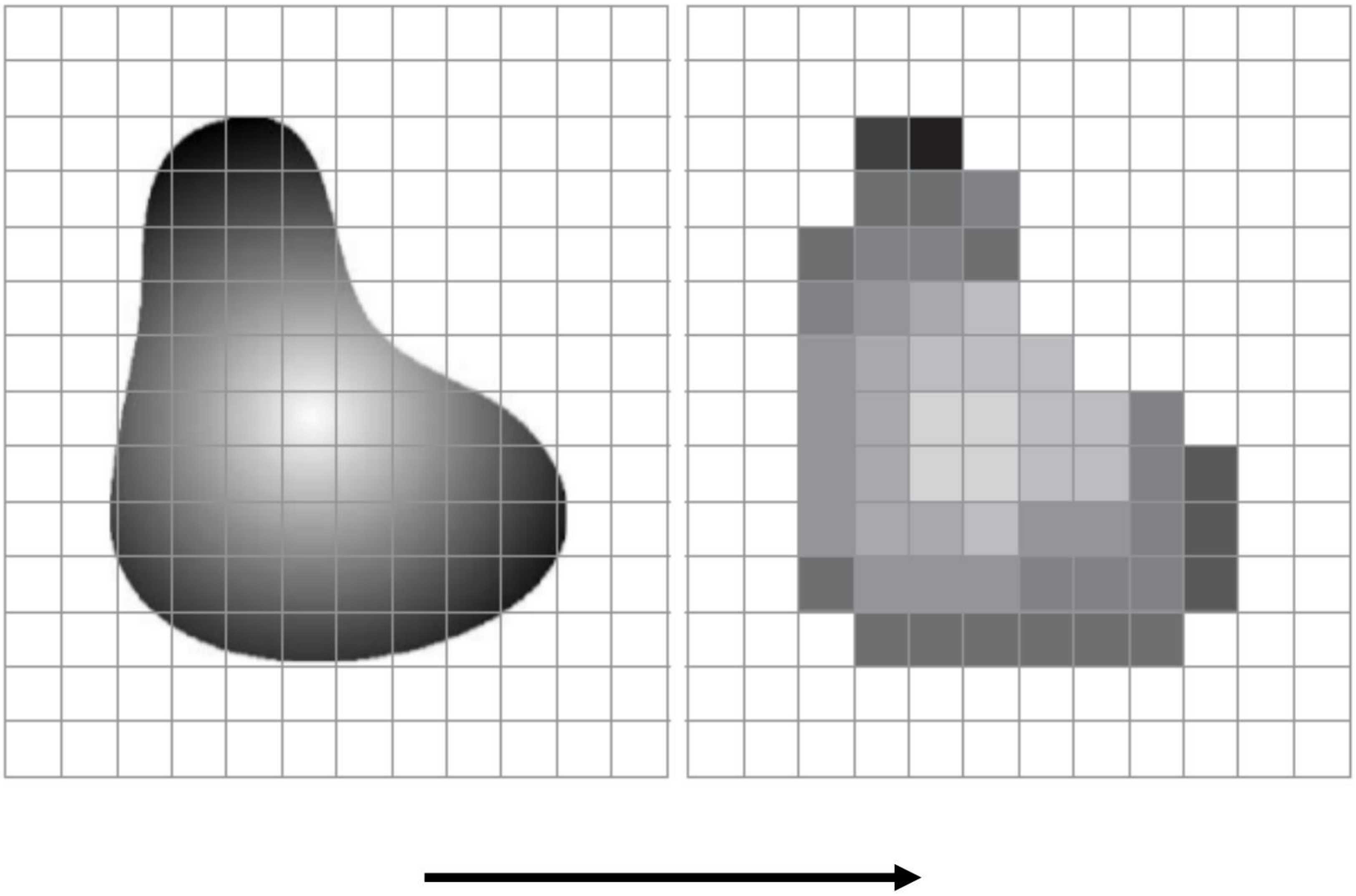


Images Formation: Quantization

- The intensity (output) of the function must also be discretized (quantized) in a finite set of values to be digitized and used in a computer.
- Image codomain is divided into a set of values and each ***f(x, y)*** is rounded to the closest one.

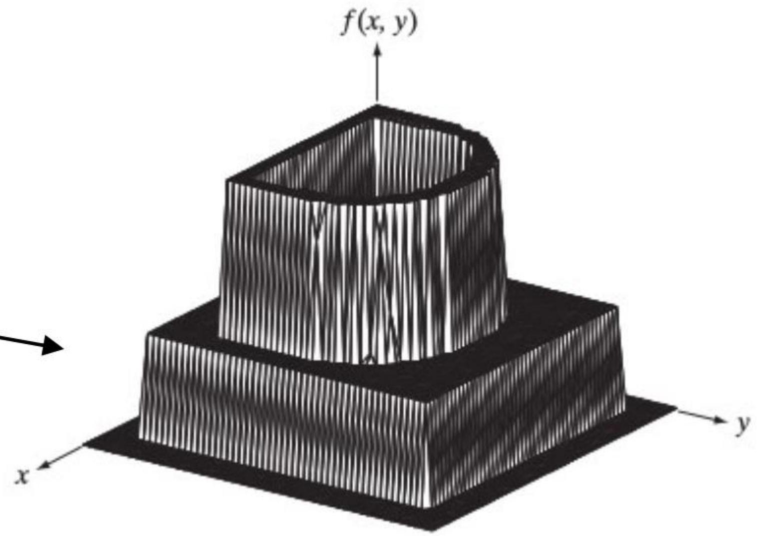


Images Formation: Sampling and Quantization



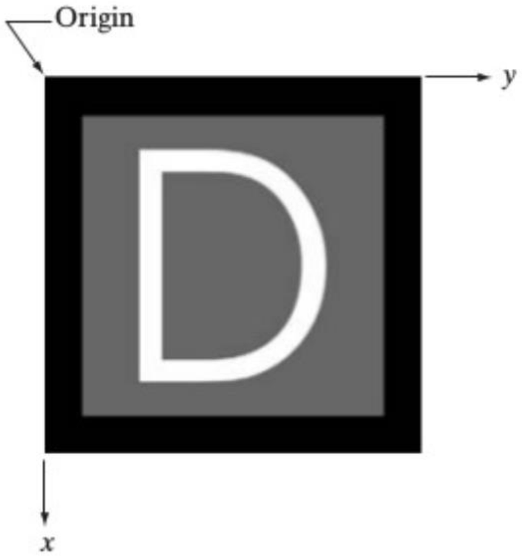
Digital Images

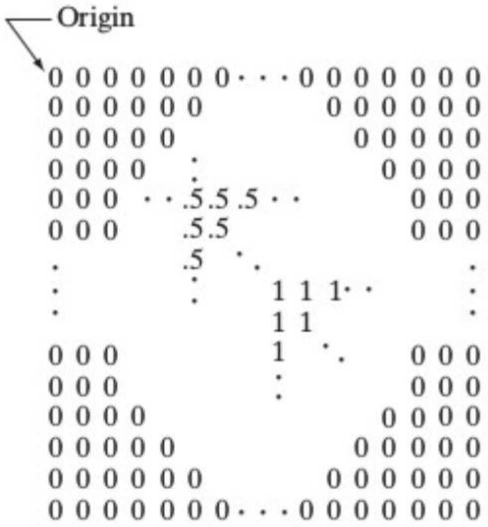
As a surface



As a Matrix

As a visual intensity array





Digital Images

- Intensity and Matrix representation are the most common to be used in practice.
- The digitization process requires that decisions be made regarding the matrix size M, N and the number L (usually a power of 2) of quantized intensity levels.
- **M, N** are related to the **spatial resolution** of an image.
- **L** is related to the **intensity resolution**.

Digital Images: Spatial Resolution

- Spatial resolution is a measure of the smallest discernible detail in an image.
- Usually defined as dots (pixels) per unit distance.
- To be meaningful, measures of spatial resolution must be stated with respect to spatial units.
- A 1024x1024 pixel image is not meaningful without stating the spatial dimensions encompassed by the image (the size of each pixel in the 3D world).

Digital Images: Spatial Resolution

- The number of pixels of an image directly affect spatial resolution only with comparable lenses and if the subjects are taken at the same distance.

a. Landsat ETM+

b. ATLAS

c. QuickBird

Digital Images: Spatial Resolution

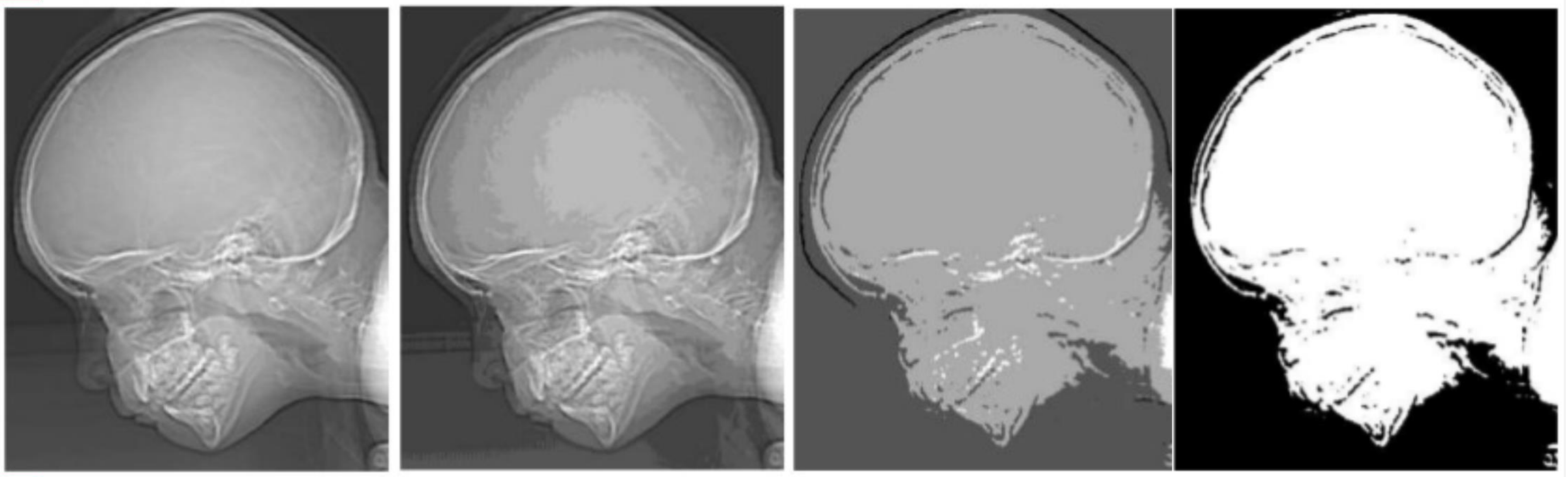
1250 dpi

300 dpi

72 dpi

Digital Images: Intensity resolution

- Intensity resolution refers to the smallest discernible change in intensity level.
- Usually, it is a power of 2 (8 bits most common).



256 levels **16 levels** **4 levels** **2 levels**

Digital Images: Few Definitions

- **Dynamic range** of an imaging system:

The *ratio* of the maximum measurable intensity to the minimum detectable intensity level.

Depends by the number of bits we use and establish the lowest and highest intensity levels that a system can represent.

=> Human eye has a dynamic range of about 10^9!

- A high dynamic range (HDR) image preserves details in both the shadows and highlights.
- A low dynamic range (LDR) image loses details in either the dark or bright areas.

Digital Images: Few Definitions

- **Contrast:**

The *difference* between the highest and the lowest intensity levels of an image.

- Even if we use techniques to acquire images at high dynamic range, we will probably will not be able to display it on a computer screen...
- Compromises have to be done in the acquisition process to either discard dark details or saturate heavily illuminated areas.

Dynamic range vs. Intensity resolution

Dynamic range and **intensity resolution** in a digital image both relate to how brightness levels are represented, but they refer to different aspects of image quality:

- **Dynamic Range** refers to the ratio between the brightest and darkest intensities that a system (such as a camera or a display) can capture or reproduce.

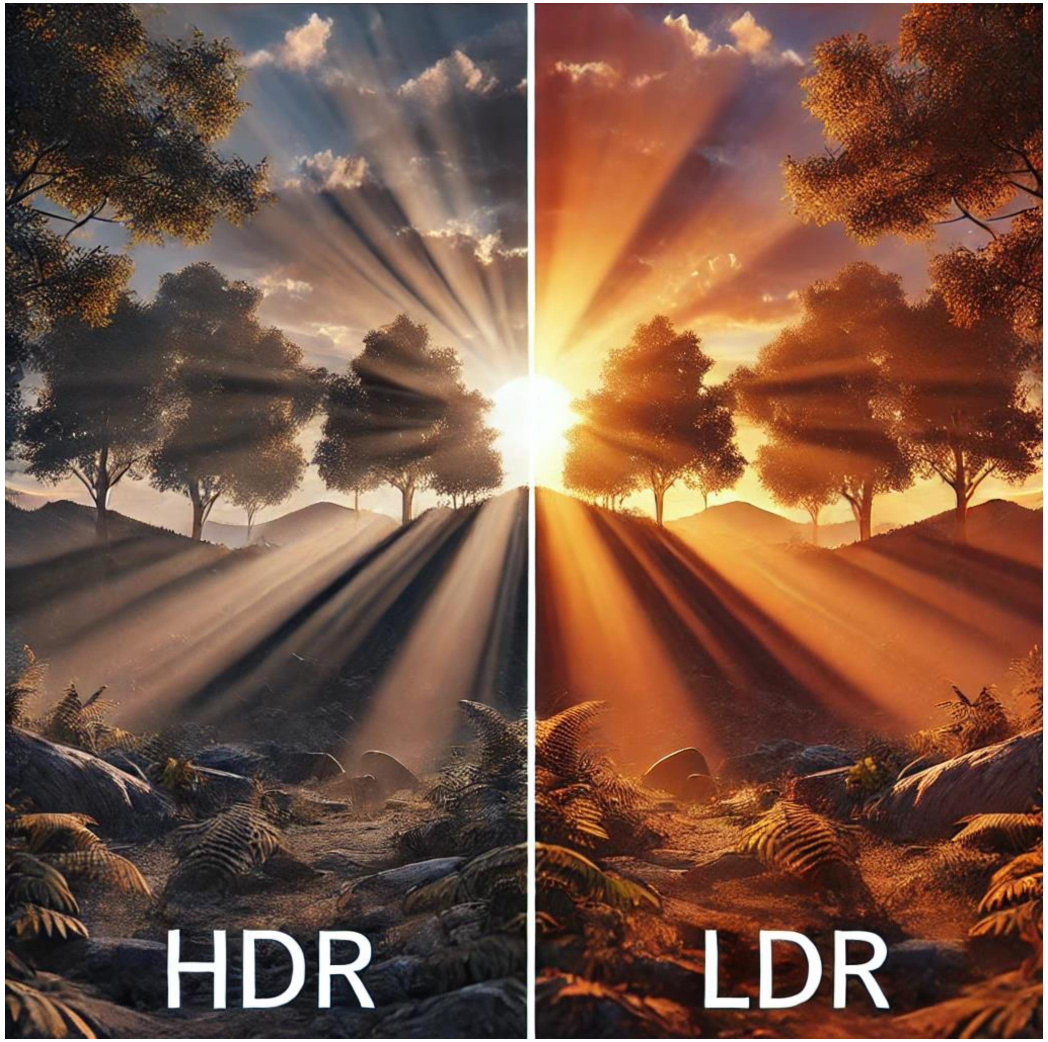
  A **higher dynamic range** means better detail in both very bright and very dark areas of the image.

  Example: HDR (High Dynamic Range) imaging captures a wider range of light intensities compared to standard images.

- **Intensity Resolution** (Bit Depth) refers to the **number of discrete levels** used to represent pixel intensities in a digital image.

  A **higher intensity resolution** allows for smoother transitions between brightness levels and reduces banding artifacts.

Dynamic range



- **Left (High Dynamic Range - HDR)**: Preserves details in both the shadows and highlights, making the scene appear more natural and well-balanced.
- **Right (Low Dynamic Range - LDR)**: Shadows are too dark, and highlights are overexposed, leading to a loss of details in the brightest and darkest areas.

Dynamic range

```python
# Load a real-world image
image = cv2.imread("image1.jpg", cv2.IMREAD_GRAYSCALE)

# Simulate low dynamic range by clipping intensities
low_dynamic_range = np.clip(image, 50, 200)  # Compress to a narrow range
cv2.imwrite("low_dynamic_range_example1.png", low_dynamic_range)

# Simulate low dynamic range by clipping intensities
low_dynamic_range = np.clip(image, 75, 175)  # Compress to a narrow range
cv2.imwrite("low_dynamic_range_example2.png", low_dynamic_range)

# Simulate low dynamic range by clipping intensities
low_dynamic_range = np.clip(image, 100, 150)  # Compress to a narrow range
cv2.imwrite("low_dynamic_range_example3.png", low_dynamic_range)

# Simulate low dynamic range by clipping intensities
low_dynamic_range = np.clip(image, 120, 130)  # Compress to a narrow range
cv2.imwrite("low_dynamic_range_example4.png", low_dynamic_range)

# Simulate high dynamic range by preserving the full range
high_dynamic_range = cv2.normalize(image, None, 0, 255, cv2.NORM_MINMAX)
cv2.imwrite("high_dynamic_range_example.png", high_dynamic_range)
```

Dynamic range

high_dynamic_range [0, 255]

low dynamic range by clipping intensities [75, 175]

low dynamic range by clipping intensities [100, 150]

low dynamic range by clipping intensities [120, 130]

Intensity resolution

```python
# Create an image with 2-bit intensity resolution (4 levels)
image = np.linspace(0, 255, 256, dtype=np.uint8)
image = np.tile(image, (256, 1))  # Create a gradient
image = (image // 64) * 64  # Quantize to 4 levels
cv2.imwrite("2_bit_intensity_resolution.png", image)

# Create an image with 4-bit intensity resolution (16 levels)
image = np.linspace(0, 255, 256, dtype=np.uint8)
image = np.tile(image, (256, 1))  # Create a gradient
image = (image // 16) * 16  # Quantize to 16 levels
cv2.imwrite("4_bit_intensity_resolution.png", image)

# Create an image with 6-bit intensity resolution (64 levels)
image = np.linspace(0, 255, 256, dtype=np.uint8)
image = np.tile(image, (256, 1))  # Create a gradient
image = (image // 4) * 4  # Quantize to 64 levels
cv2.imwrite("6_bit_intensity_resolution.png", image)

# Create an image with 8-bit intensity resolution (256 levels)
image = np.linspace(0, 255, 256, dtype=np.uint8)
image = np.tile(image, (256, 1))  # Smooth gradient
cv2.imwrite("8_bit_intensity_resolution.png", image)

# Create an image with 16-bit intensity resolution (65,536 levels)
image = np.linspace(0, 65535, 256, dtype=np.uint16)
image = np.tile(image, (256, 1))  # Extremely smooth gradient
cv2.imwrite("16_bit_intensity_resolution.png", image)
```