Computer Vision

Paulo Dias, António Neves

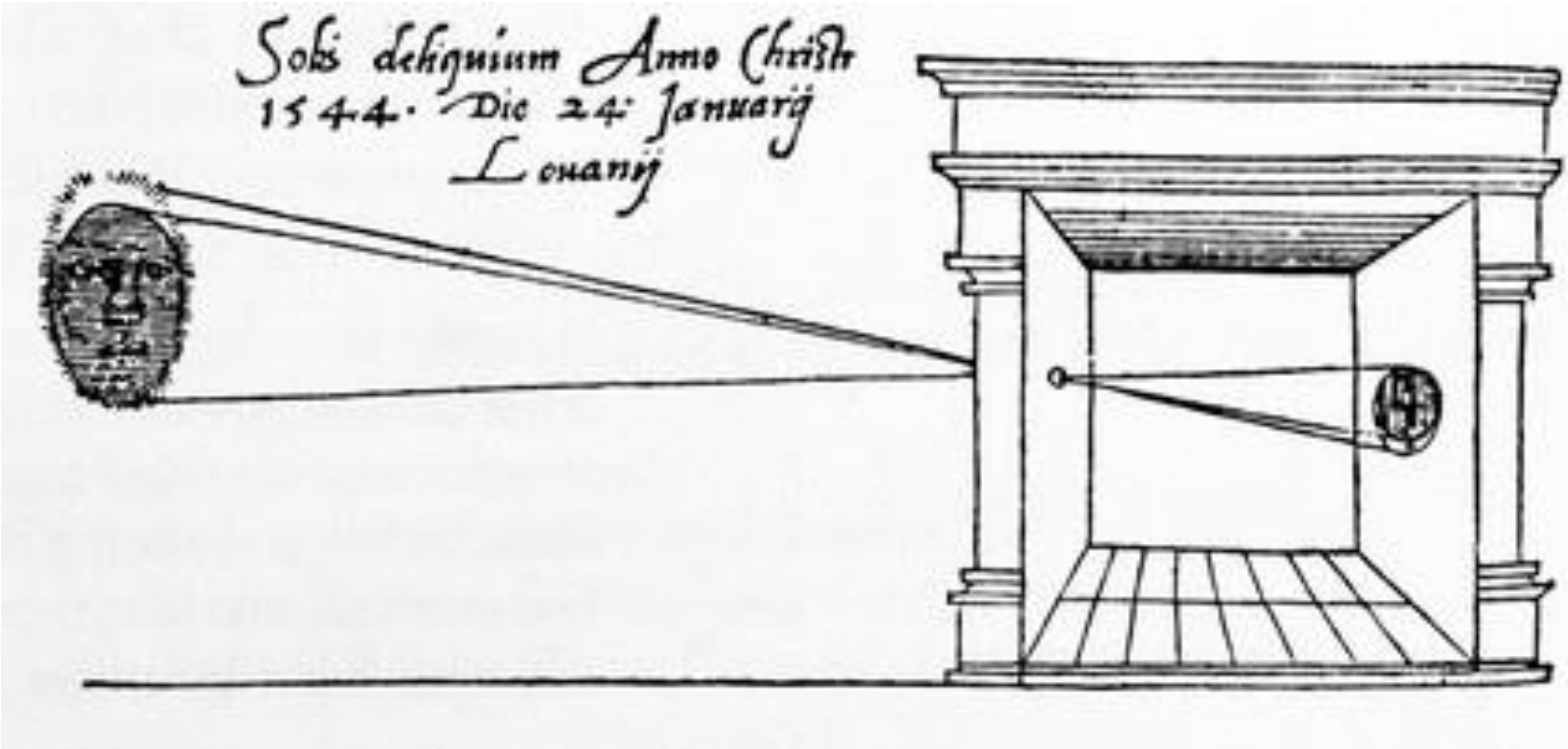
Sumary

- Image Formation
- Cameras
- Digital Images
- Color Spaces
- Image representation in openCV

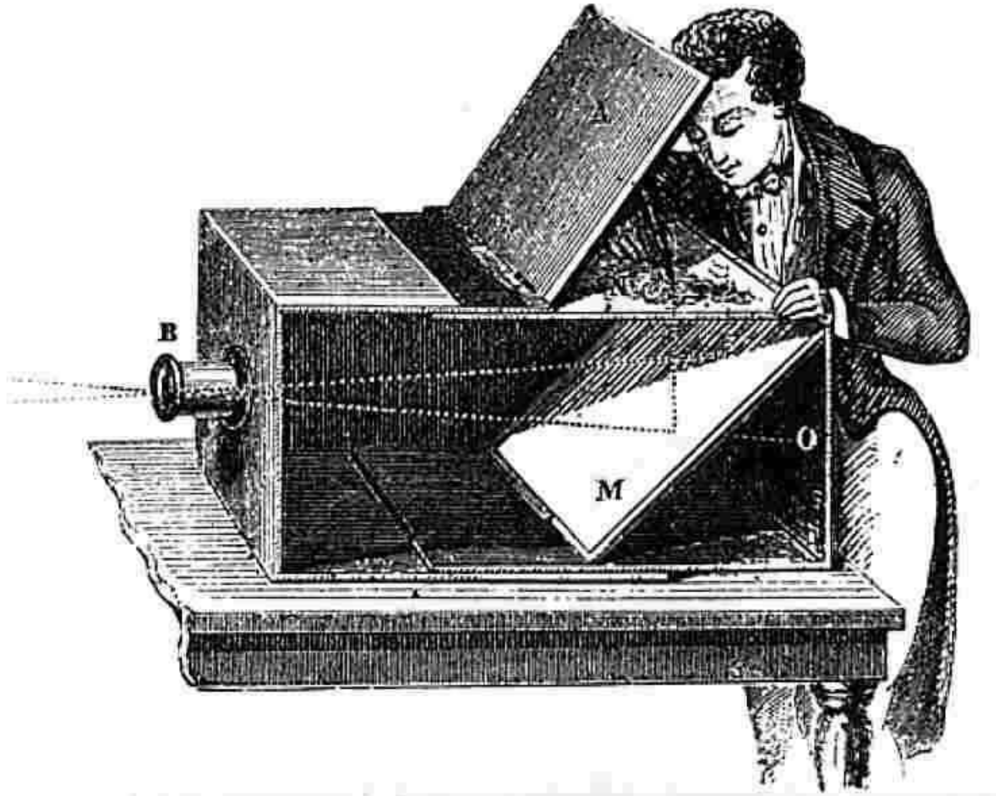
Sumary

- Image Formation
- Cameras
- Digital Images
- Color Spaces
- Image representation in openCV

Hitory of Camera (1544)



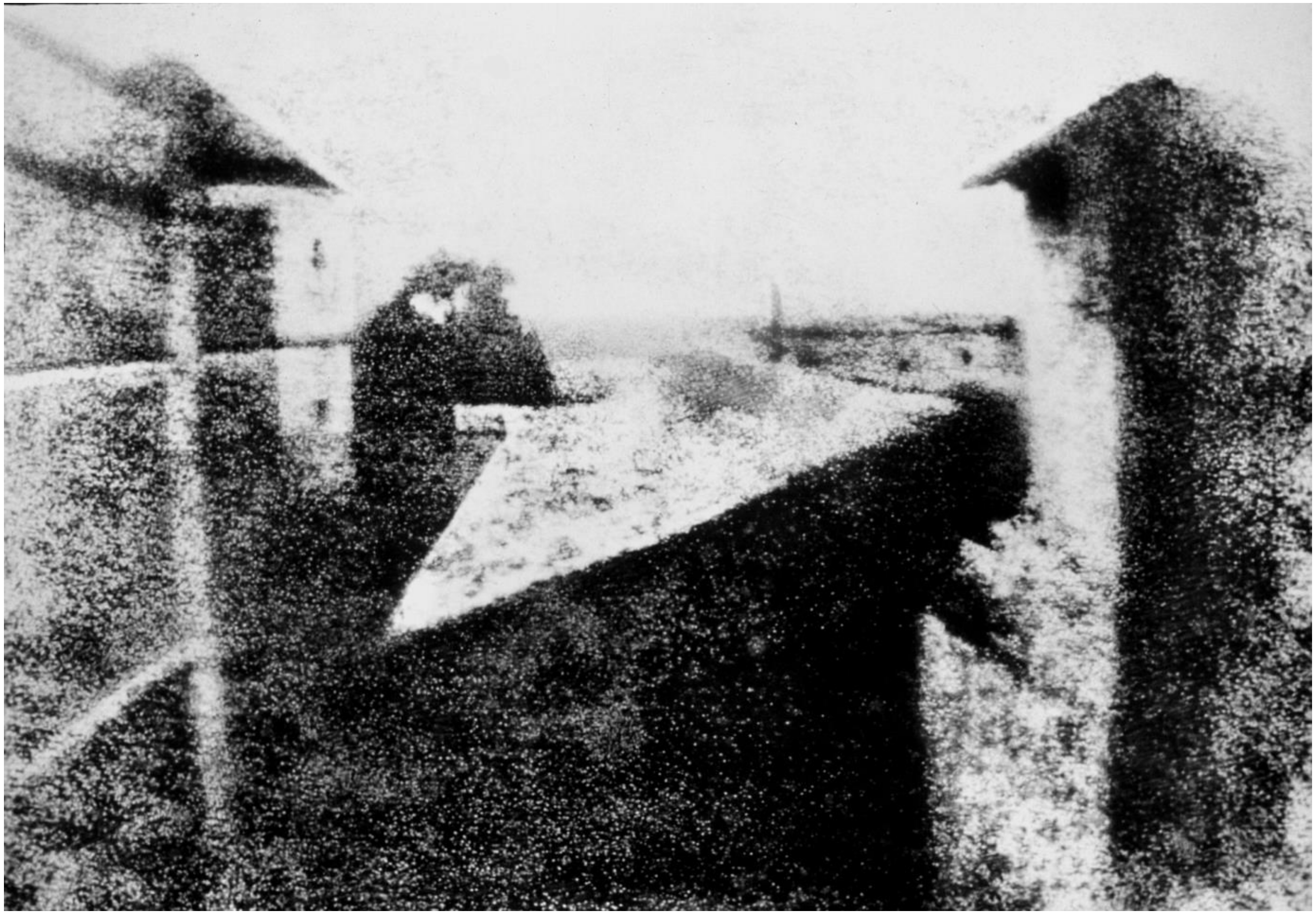
Camera Obscura, Gemma Frisius, 1544



18th-century camera obscura to trace an image

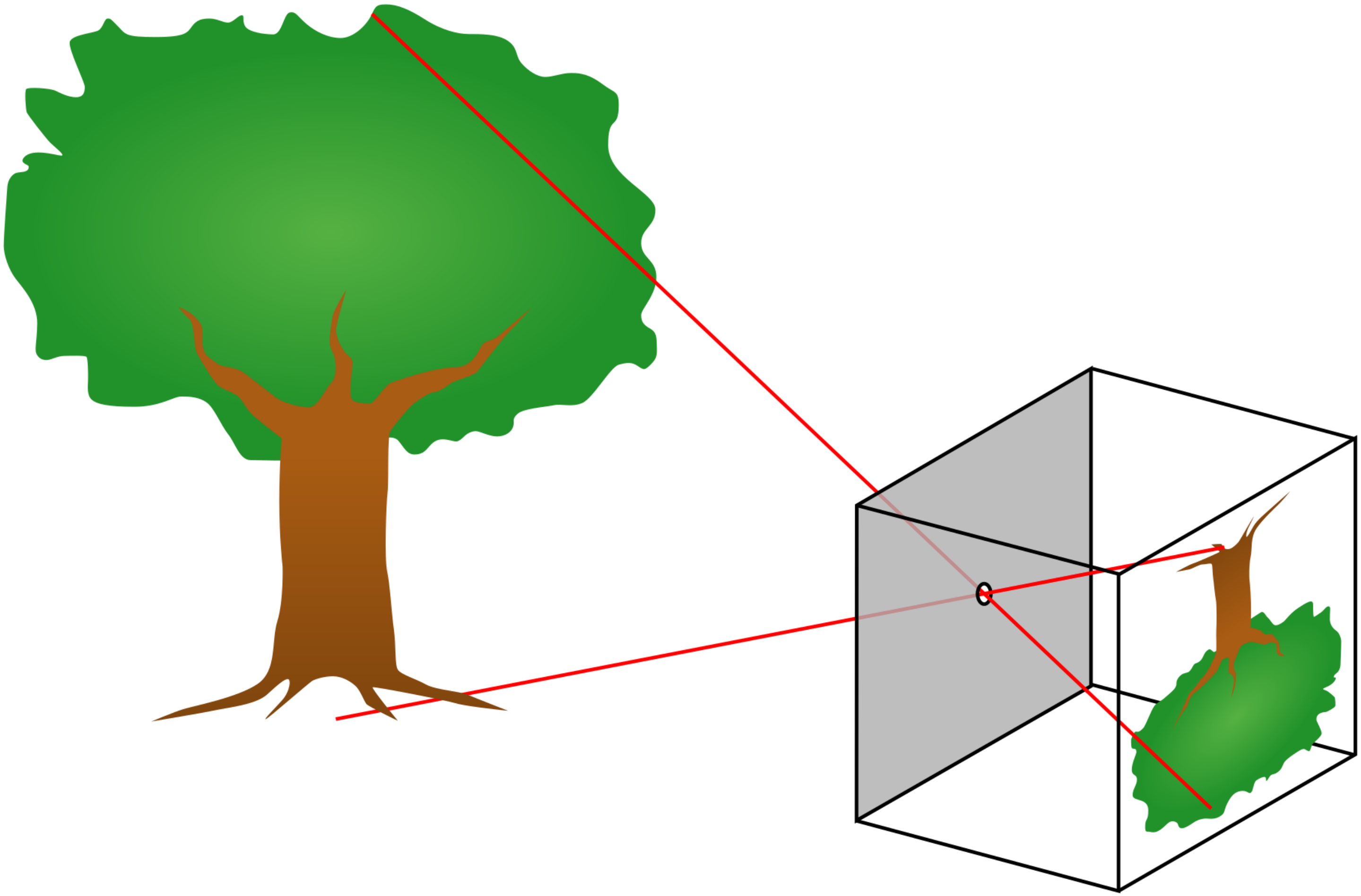
First photograph:

First permanent photograph of a camera image was made in 1825 by Joseph Nicéphore Niépce using a sliding wooden box camera



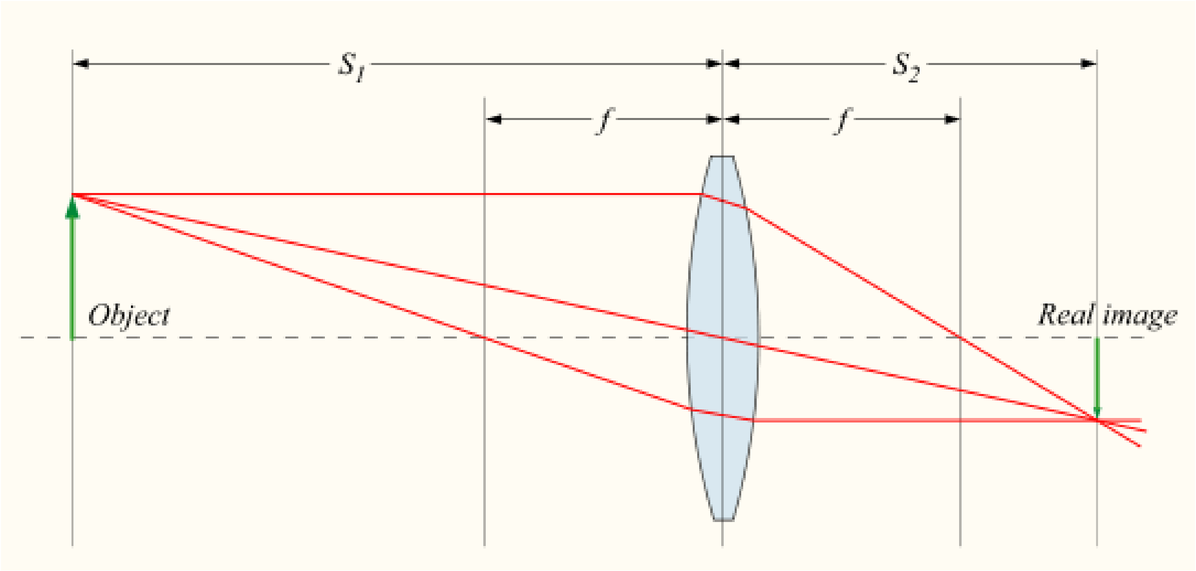
View from the Window at Le Gras

Pinhole Camera principle



Light enters a dark box through a small hole and creates an inverted image on the wall opposite the hole

Image through a lens



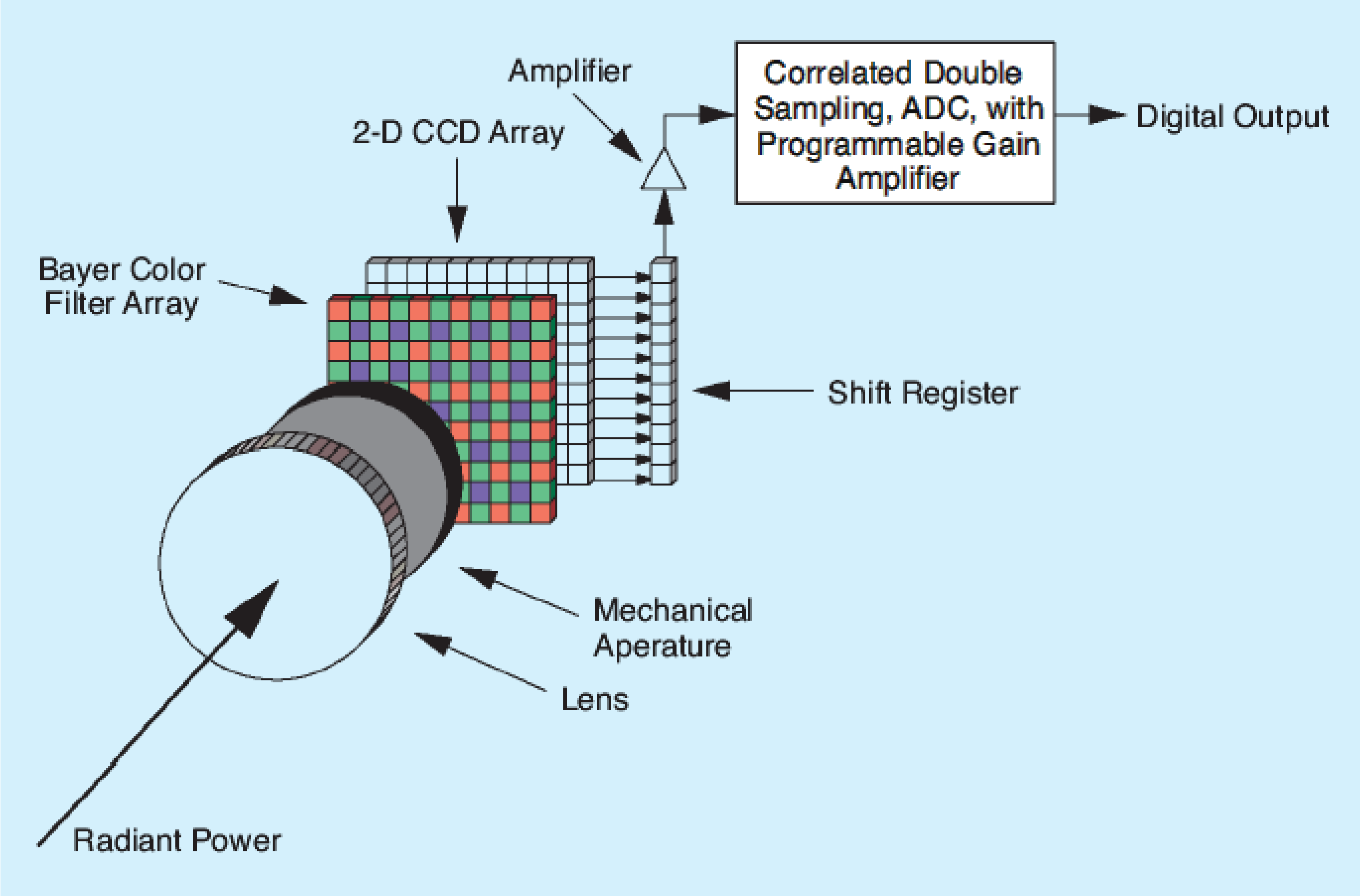
- All the rays of light that came from an object in direction to the lens converge, on the other side, in another point at a certain distance from the lens. This distance is called focal distance.
- All the points that verify this fact are denoted the focal plane.
- There are some other important parameters related to lens: Field of View, Depth of Field, . . .

Basic camera geometrry

- Far objects appear smaller.
- Lines project to lines.
- Lines in 3D project to lines in 2D.
- Distances and angles are not preserved.
- These geometric properties are "common sense". Other properties can be inferred if we formalize the model using . . . Mathematics, of course (see Camera Calibration class)

Sumary

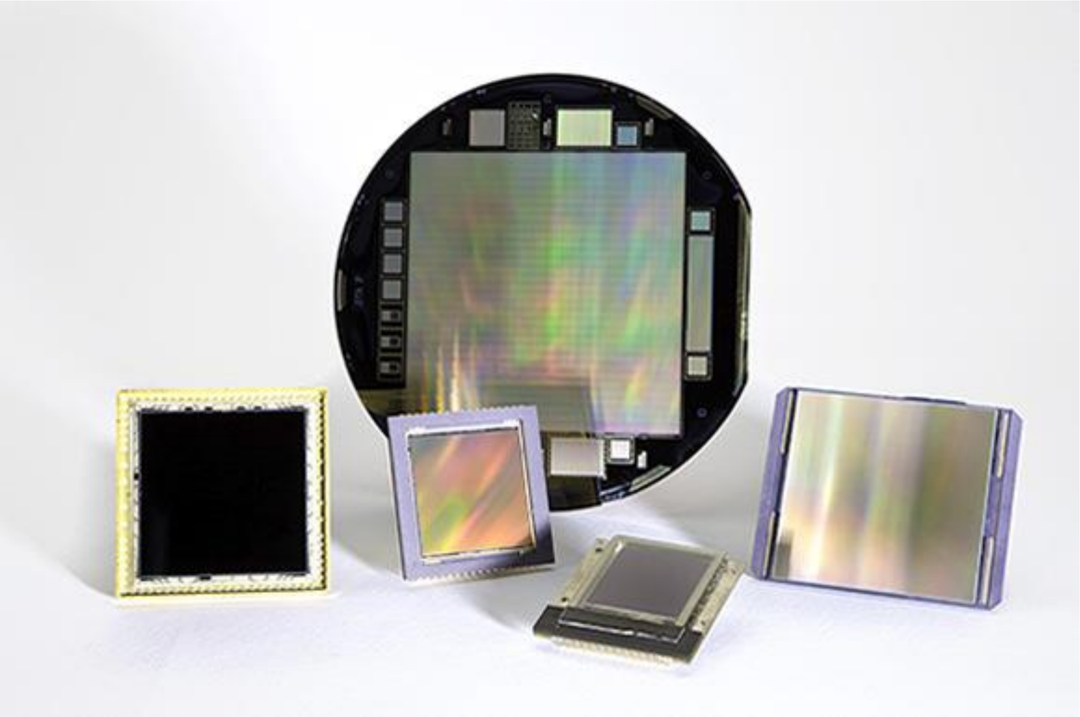
- Image Formation
- Cameras
- Digital Images
- Color Spaces
- Image representation in openCV



- Image acquisition using a digital camera:

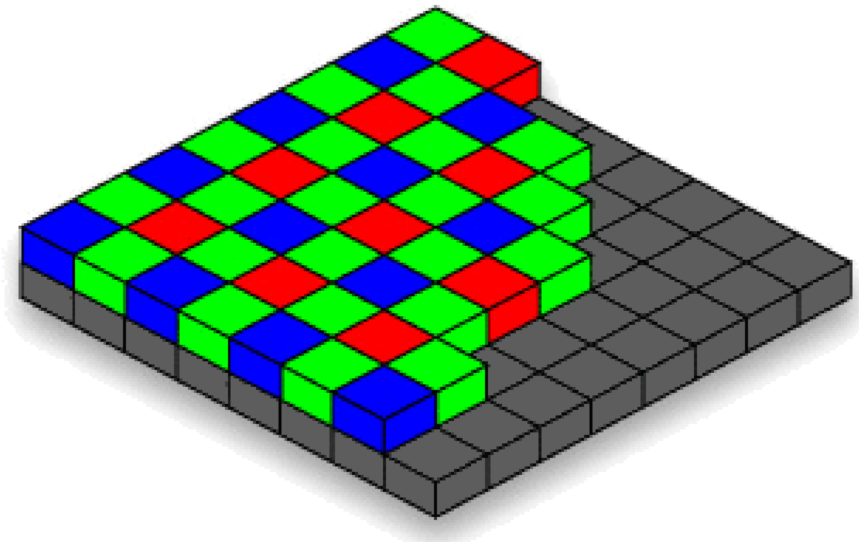
(IEEE SP Magazine, Jan 2005)

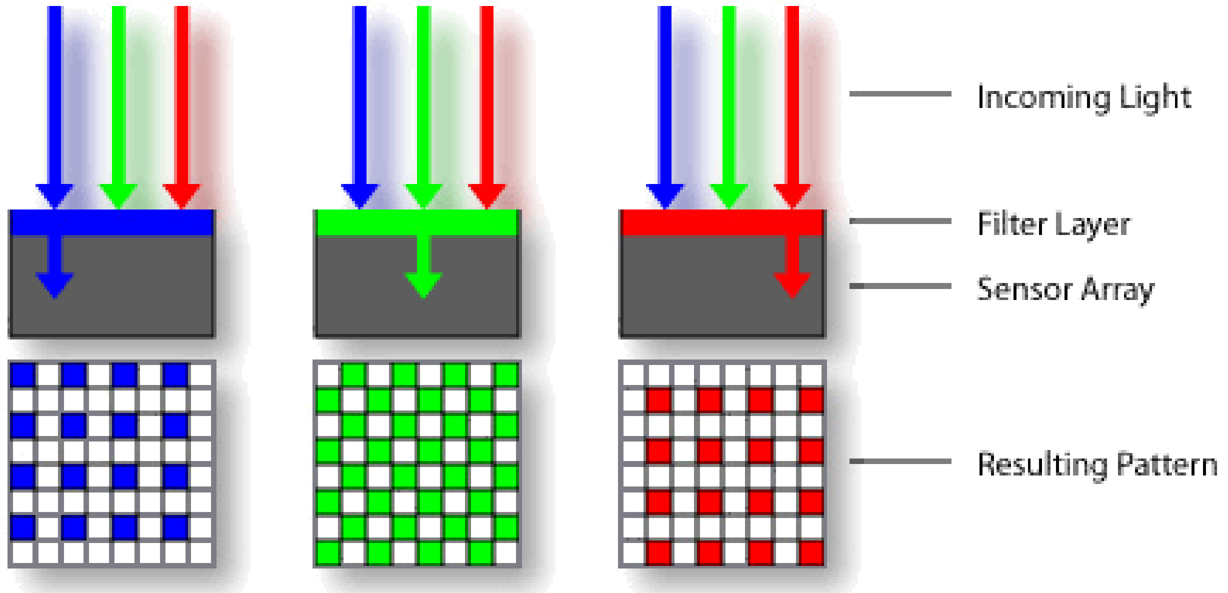
Image Sensors



- Some considerations: speed, resolution, cost, signal/noise ratio, . . .
- CCD - charge coupled device - Higher dynamic range, High uniformity, Lower noise.
- CMOS - Complementary Metal Oxide Semiconductor - Lower voltage, Higher speed, Lower system complexity

Color: Bayer Matrix





Digital Cameras: several alternatives

Digital Cameras: several alternatives

- Several interfaces (Firewire, GigE, CameraLink, USB, . . .).
- Scientific usage (high resolution, long exposure time, . . .).
- High speed (ex. 1000 fps).
- Linear (ex. 10000 lines per second).
- 3D
- Infrared (ex. 8 to 14 µm).
- High dynamic range (ex. using a prism and two sensors).
- Multispectral

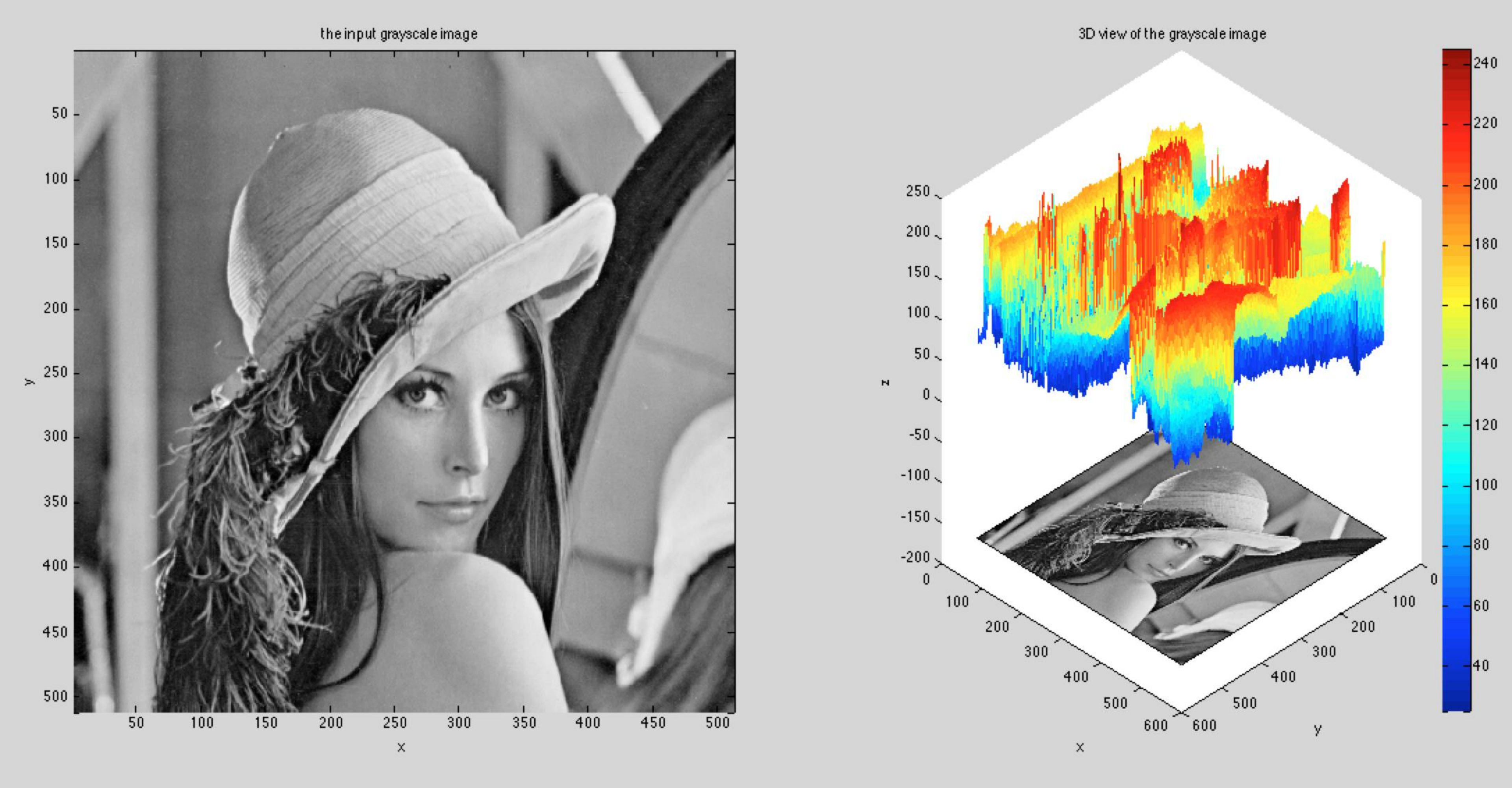
Sumary

- Image Formation
- Cameras
- Digital Images
- Color Spaces
- Image representation in openCV

Images as a function

Grey Scale Image - Lena

Images as a function



Lena 3D plot

Image as a function: Real Lena

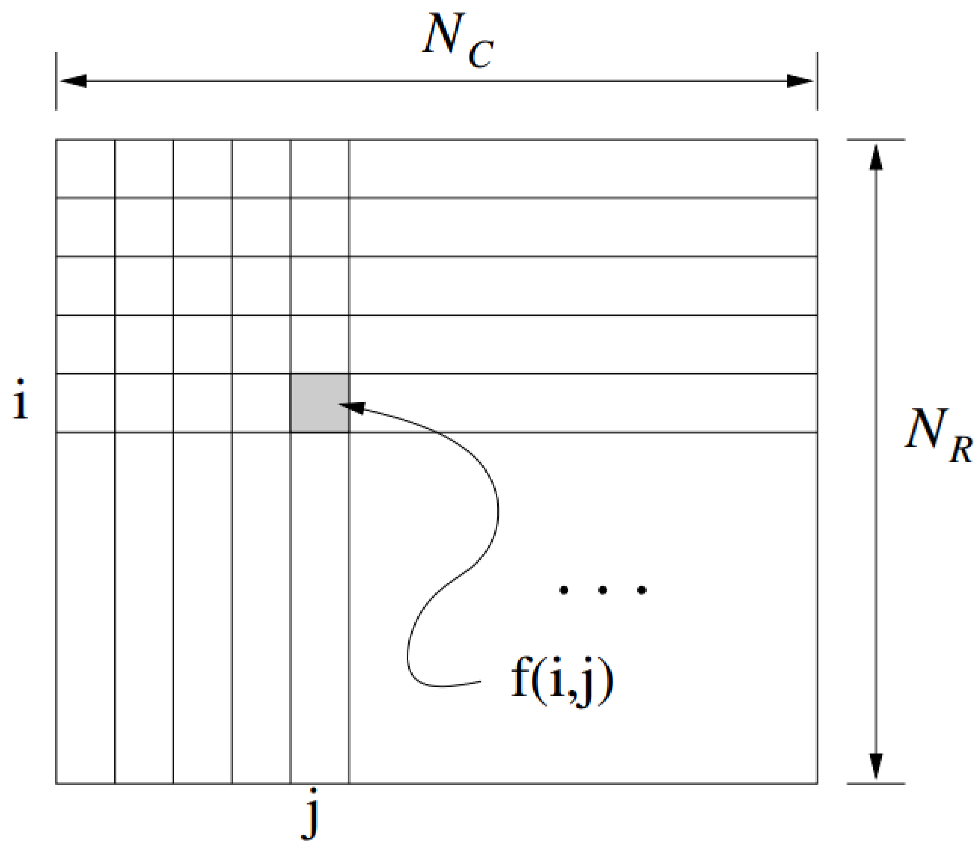
$$
\begin{array}{cccccccccccc}
148 & 123 & 52 & 107 & 123 & 162 & 172 & 123 & 64 & 89 & \cdots \\
147 & 130 & 92 & 95 & 98 & 130 & 171 & 155 & 169 & 163 & \cdots \\
141 & 118 & 121 & 148 & 117 & 107 & 144 & 137 & 136 & 134 & \cdots \\
82 & 106 & 93 & 172 & 149 & 131 & 138 & 114 & 113 & 129 & \cdots \\
57 & 101 & 72 & 54 & 109 & 111 & 104 & 135 & 106 & 125 & \cdots \\
138 & 135 & 114 & 82 & 121 & 110 & 34 & 76 & 101 & 111 & \cdots \\
138 & 102 & 128 & 159 & 168 & 147 & 116 & 129 & 124 & 117 & \cdots \\
113 & 89 & 89 & 109 & 106 & 126 & 114 & 150 & 164 & 145 & \cdots \\
120 & 121 & 123 & 87 & 85 & 70 & 119 & 64 & 79 & 127 & \cdots \\
145 & 141 & 143 & 134 & 111 & 124 & 117 & 113 & 64 & 112 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}
$$

F(x,y)

I(u,v)

Image as a function

- A digital image is represented by a rectangular matrix of scalars or vectors



- The f(i; j) are named pixels

Image as a Function

- An image can be represented by a two-dimensional function, f(x; y), where x and y are spatial coordinates.
- Meaning of f in a given point in space, (x; y), depends on the source (visible light, x-rays, ultrasound, radar, . . .).
- Spatial coordinates and the function values are continuous quantities.
- To convert f(x; y) into a digital image, it is necessary to perform spatial sampling and amplitude quantization.

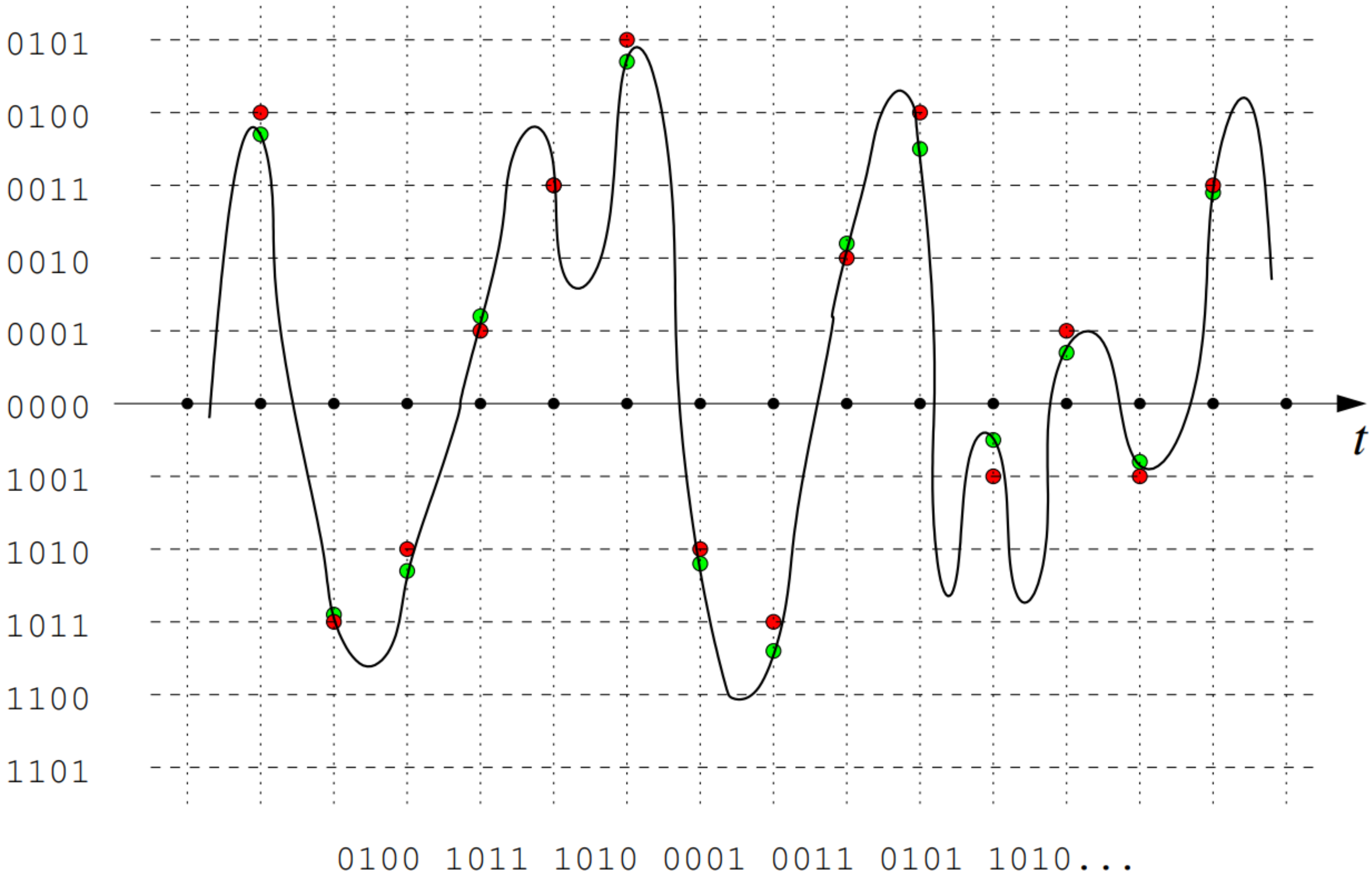
Digitalization: Sampling and quantization

- Continuous light distribution is spatially sampled

- Still image created by time sampling that discrete distribution

- Resulting values are quantized to a finite set of numerical values

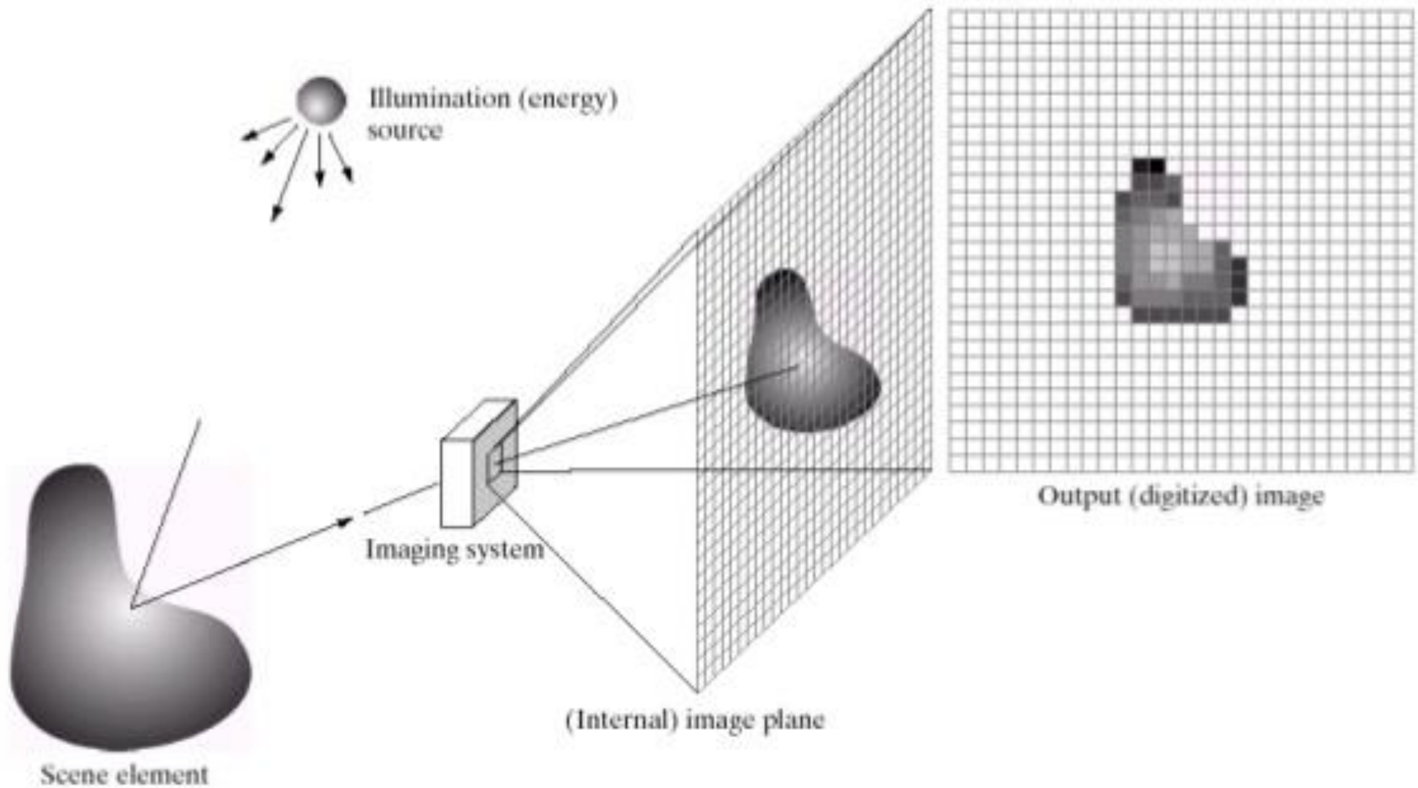
Digitalization: Sampling and quantization



Sampling and quantization of signal

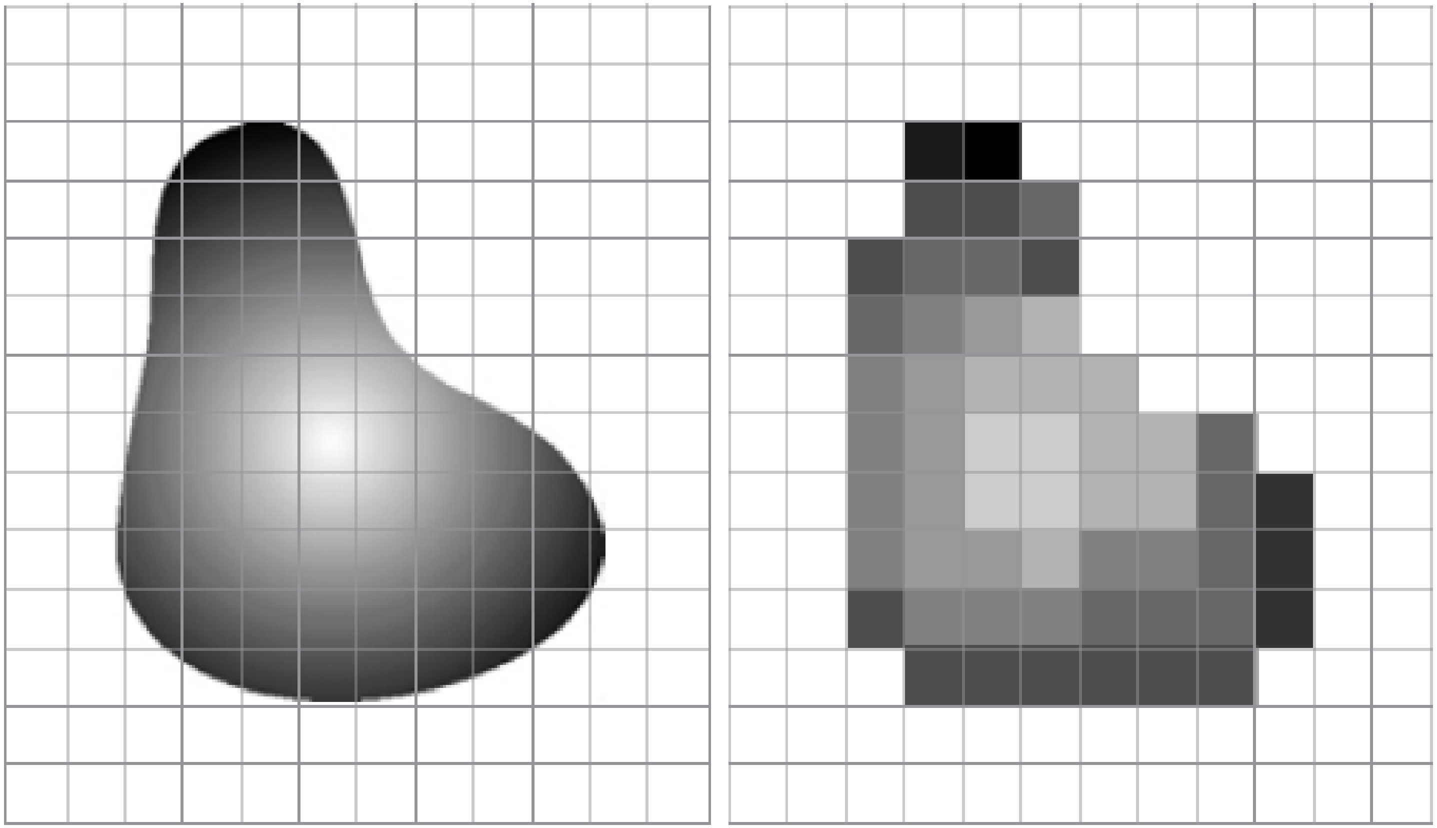
Digitalization: Sampling and quantization

Digital Image Acquisition Process



(Images from Rafael C. Gonzalez and Richard Wood, Digital Image Processing, 2nd Edition.

Digitalization: Sampling and quantization



Gonzalez and Woods

- Sampling process means digitizing the coordinate values
- Quantization means digitizing the amplitude values

Digital Images

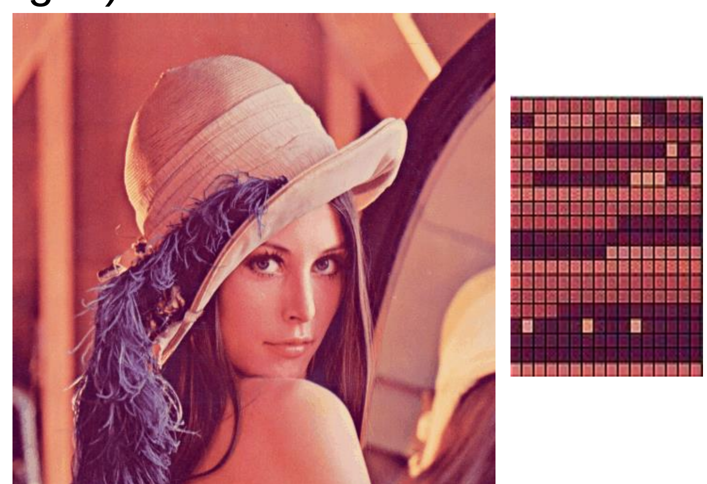
- We will consider digital images of the following types:
  - Black and white (binary images).
    $$f(i,j) \in \{0,1\}$$
  - Grayscale images.
    $$f(i,j) \in \{0,1,\ldots,2^b - 1\}$$
  - Color-indexed images.
    $$f(i,j) \in \{0,1,\ldots,2^b - 1\} \xrightarrow{\alpha} I \in \left\{0,1,\ldots,2^{b'} - 1\right\}^3$$
  - Color images (for example, RGB images)
    $$f(i,j) \in \{0,1,\ldots,2^{b'} - 1\}^3$$

Digital Images

Black and white
(binary images)

Grayscale image

Colour image / RGB



Colour-indexed image

Digital images

- Colour indexed images – why?

- saves memory, storage space, and transmission time
  - RGB

    24 bits/pixels, vga - 640×480×3 = 921,600 bytes (900 KiB).
  - 256 indexed colours

    8 bits/pixel, vga - 640×480×1 = 307,200 bytes (300 KiB)

    + 256×3 = 768 bytes for the RGB palette map

    ≈1/3 of original size

Digital Images

- Black and white (binary images).

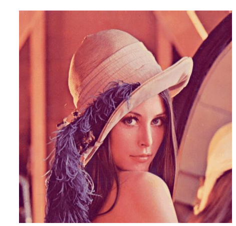
$$f(i,j) \in \{0,1\}$$

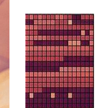
- Grayscale images.

$$f(i,j) \in \{0,1,\dots,2^b - 1\}$$

- Color-indexed images.

$$f(i,j) \in \{0,1,\dots,2^b - 1\} \xrightarrow{\alpha} I \in \left\{0,1,\dots,2^{b'} - 1\right\}^3$$





- Color images (for example, RGB images)

$$f(i,j) \in \{0,1,\dots,2^{b\prime} - 1\}^3$$

Digital Images

Grayscale (Intensity Images):

Chan.	*Bits/Pix.*	*Range*	*Use*
1	1	$0 \ldots 1$	Binary image: document, illustration, fax
1	8	$0 \ldots 255$	Universal: photo, scan, print
1	12	$0 \ldots 4095$	High quality: photo, scan, print
1	14	$0 \ldots 16383$	Professional: photo, scan, print
1	16	$0 \ldots 65535$	Highest quality: medicine, astronomy

Color Images:

Chan.	*Bits/Pix.*	*Range*	*Use*
3	24	$[0 \ldots 255]^3$	RGB, universal: photo, scan, print
3	36	$[0 \ldots 4095]^3$	RGB, high quality: photo, scan, print
3	42	$[0 \ldots 16383]^3$	RGB, professional: photo, scan, print
4	32	$[0 \ldots 255]^4$	CMYK, digital prepress

Special Images:

Chan.	*Bits/Pix.*	*Range*	*Use*
1	16	$-32768 \ldots 32767$	Whole numbers pos./neg., increased range
1	32	$\pm 3.4 \cdot 10^{38}$	Floating point: medicine, astronomy
1	64	$\pm 1.8 \cdot 10^{308}$	Floating point: internal processing

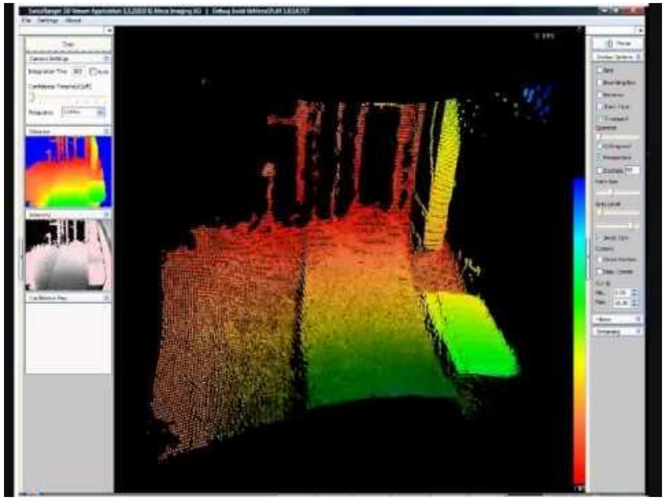
Gonzalez and Woods

Digital Images

- Other images

Infra red (Gobi camera)

Depth image (Kinect)



Point Cloud (swissranger)

Digital Videos

- A video signal can be represented by a 3D-function

  $v(x; y; t)$,

where x and y are spatial coordinates and t denotes time.

- The process of converting analog video into digital video requires spatial and temporal sampling, besides amplitude quantization.

- A digital video is a temporal sequence of digital images which we represent by $v(i; j; k)$, with

$$k = \frac{t}{T}.$$

- T indicates the time period between two consecutive images (frames). Therefore, the frame rate 1/T (Hz) is inversely proportional to T.

Sumary

- Image Formation
- Cameras
- Digital Images
- Color Spaces
- Image representation in openCV

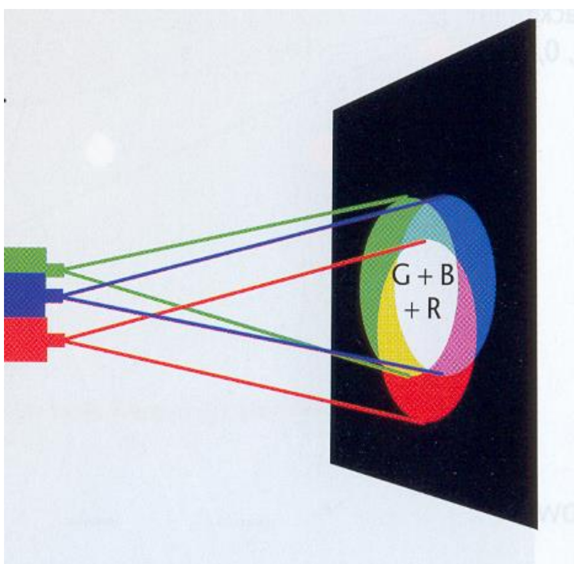
Colour models

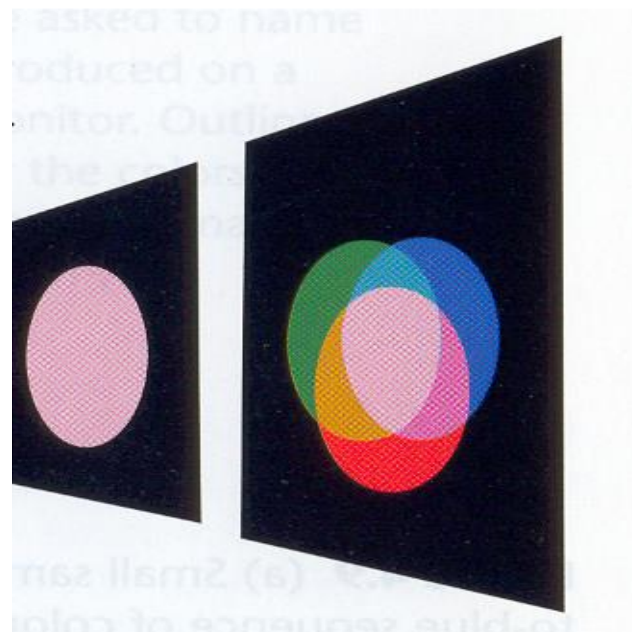
- Objects are perceived as having a color depending on the spectrum of the reflected light (or emitted)
- But different spectra may induce similar color sensations
- It is important to be able to describe color objectively
- There are to types of color production systems:
  - Additive (eg.: monitors, TV sets, projectors)
  - Subtractive (e.g.: printers)

- Any color may be represented by the superposition of 3 basic colors, adjusting their intensity to match the intended color (RGB in additive systems)

$$C= a_1R + a_2G + a_3B$$

Additive system





https://en.wikipedia.org/wiki/Additive_color

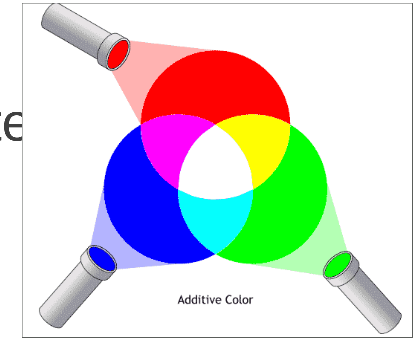
Colour models

- There are several models that include:

  - a 3D coordinate system

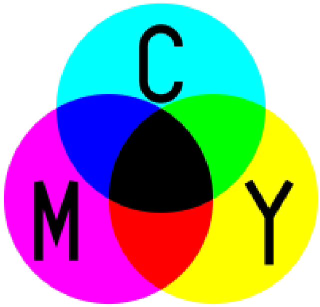
  - a geometric solid

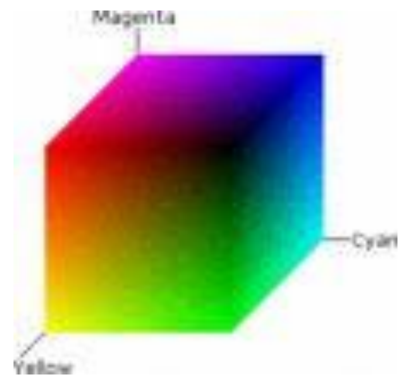
Additive



- **RGB** (Red Green Blue) is H/W oriente and standard for computer monitors

- **CMY** (Cian, Magenta, Yellow) is H/W oriented and standard for printers





Subtractive

Colour models

- Models designed to more closely aligned with the way humans perceive color-making attributes

https://en.wikipedia.org/wiki/Hue

- **HSV** (Hue, Saturation and Value)
  - Hue is wavelength of color
  - Saturation is amount of pure color
    - 0% = gray, 100% = pure
  - Value is brightness
    - 0% = dark, 100% = bright

- **HLS** (Hue, Lightness and Saturation)
  - White has lightness 1.0
  - Pure colors have lightness 0.5

color Picker:

RGB to HSV:

$$V = \max(R, G, B)$$

$$S = \begin{cases} \frac{V - \min R,G,B}{V} & if\_V \neq 0 \\ 0 & otherwise \end{cases}$$

$$H = \begin{cases} 60(G - B)/S & if\ V = R \\ 120 + 60(B - R)/S & if\ V = G \\ 240 + 60(R - G)/S & if\ V = B \end{cases}$$

Colour models - conversions

RGB to HSL:

$$
\begin{aligned}
V_{max} &= \max R, G, B \\
V_{min} &= \min R, G, B
\end{aligned}
$$

$$
L = \frac{V_{max} + V_{min}}{2}
$$

$$
S = \begin{cases} \frac{V_{max} - V_{min}}{V_{max} + V_{min}} & \text{if } L < 0.5 \\ \frac{V_{max} - V_{min}}{2 - (V_{max} + V_{min})} & L \geq 0.5 \end{cases}
$$

$$
H = \begin{cases} 60(G - B)/S & \text{if } V_{max} = R \\ 120 + 60(B - R)/S & \text{if } V_{max} = G \\ 240 + 60(R - G)/S & \text{if } V_{max} = B \end{cases}
$$

Sumary

- Image Formation
- Cameras
- Digital Images
- Color Spaces
- Image representation in openCV

Class mat

- OpenCV is an image processing library. It contains a large collection of image processing functions.
- To solve a computational challenge, most of the time you will end up using multiple functions of the library.
- Mat is basically a class with two data parts: the matrix header and a pointer to the matrix containing the pixel values.
- Mat does not need manual memory allocation - most of the OpenCV functions will allocate its output data automatically.
- If we pass an already existing Mat object to a function, this will be reused.
- The copy operators will only copy the headers and the pointer to the large matrix, not the data itself.
- OpenCV also provides the `cv::Mat::clone()` and `cv::Mat::copyTo()` functions.

Class mat

- Constructor: `Mat M(rows , cols, CV_8UC3,`
- `Scalar(0,0,255))`
- `create` method: `M.create(rows,cols, CV_8UC3)`
- How the image matrix is stored in the memory?
- Access to the pixels:
  - `ptr()` method
  - Iterator method - `MatIterator_<Vec3b> it`,
  - `img.begin(),img.end().`
  - On-the-fly address calculation with reference returning - `at()` method

Other classes

- Point_ (x and y)
  - typedef Point_<int> Point2i
  - typedef Point2i Point
  - typedef Point_<float> Point2f
  - typedef Point_<double> Point2d
- Point3_ (x, y and z)
- Vec: typedef Vec<uchar, 2> Vec2b, typedef
- Vec<double, 3> Vec3d, . . .
- Size_, Rect_, . . .