Computer Vision

António J. R. Neves / Paulo Dias

Departamento de Electrónica, Telecomunicações e Informática

Universidade de Aveiro

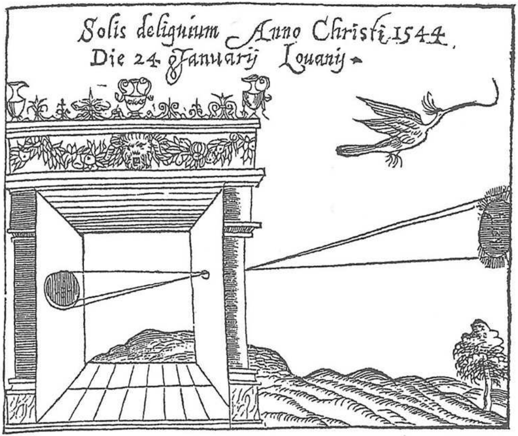
an@ua.pt / paulo.dias@ua.pt

http://elearning.ua.pt/

Contents

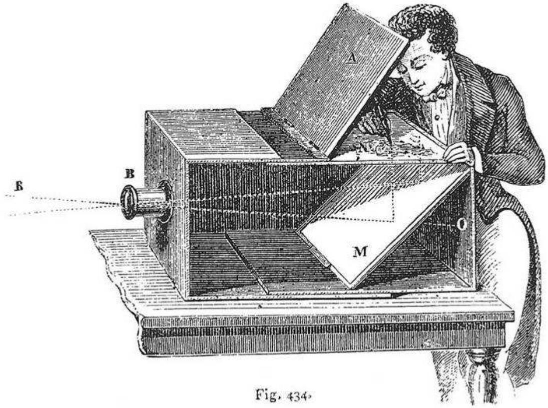
Contents

History of cameras (1544)



Camera Obscura, Gemma Frisius, 1544

History of cameras (1568)

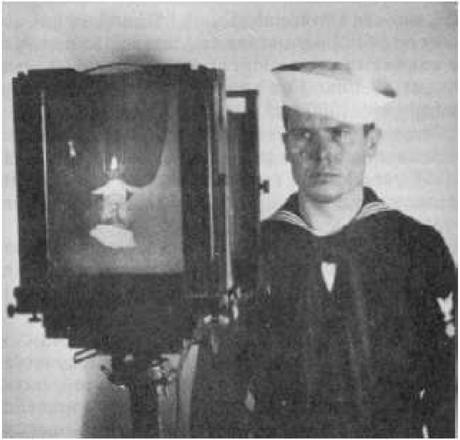


Lens Based Camera Obscura, 1568

History of cameras (1837)

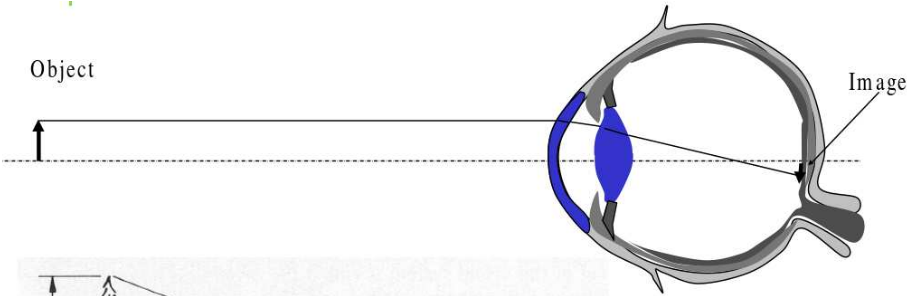
Still Life, Louis Jaques Mande Daguerre, 1837

History of cameras (1930)



Silicon Image Detector, 1970 - digital cameras

Human eye



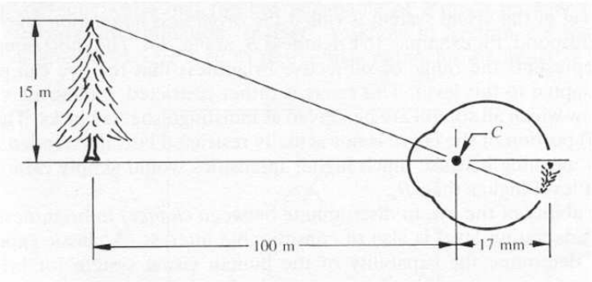
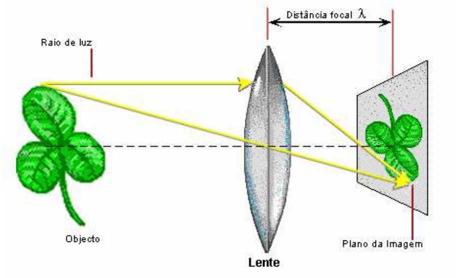


Image through a lens



- 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, ...

- 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...

Contents

- Image acquisition using a digital camera:
  (IEEE SP Magazine, Jan 2005)

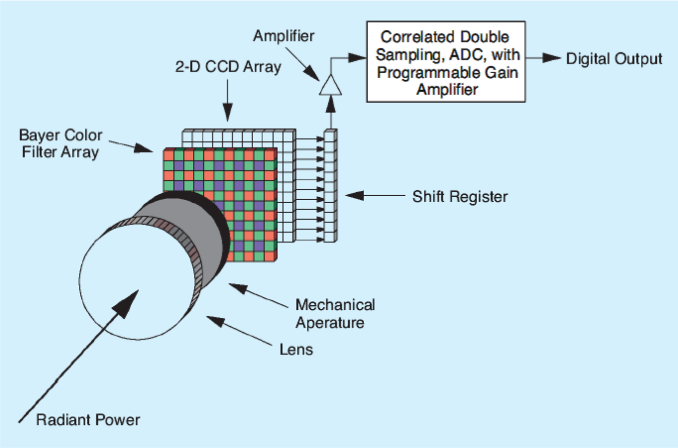
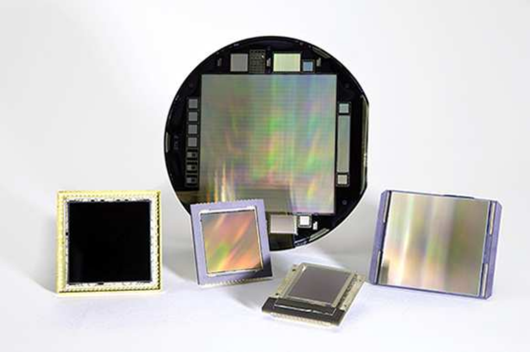
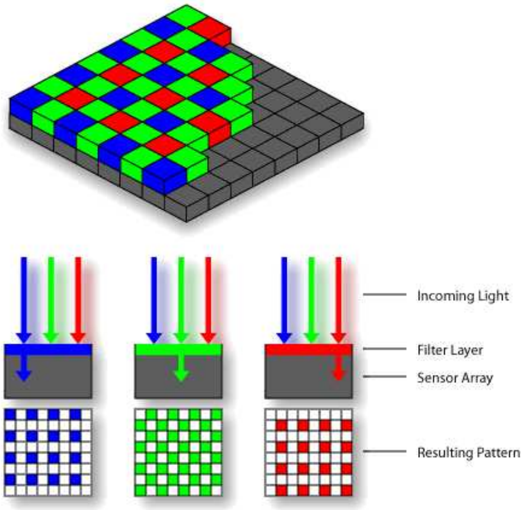


Image sensors



- 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.

The Bayer matrix



Digital cameras - several solutions

Digital cameras - several solutions

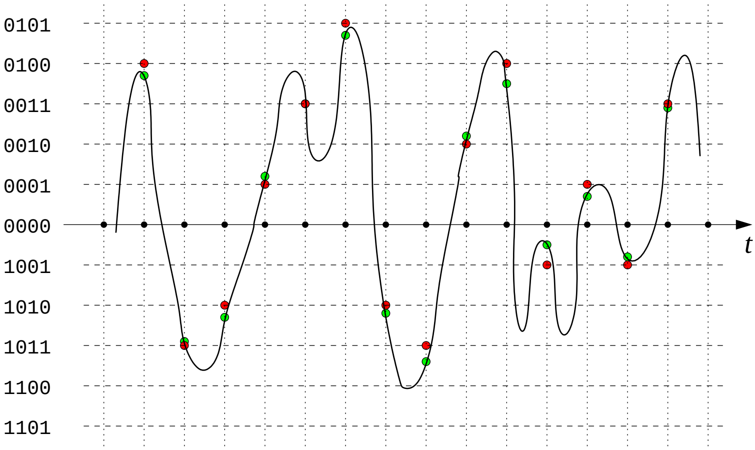
- Several interfaces (Firewire, GigE, CameraLink, USB, ...).
- Scientific usage (high resolution, long exposure time, ...).
- High speed (ex. 1000 fps).
- Linear (ex. 10000 lines per second).
- 3D
- Infrared (ex. 8 to 14 μm).
- High dynamic range (ex. using a prism and two sensors).
- Multispectral

Contents

Sampling and quantization

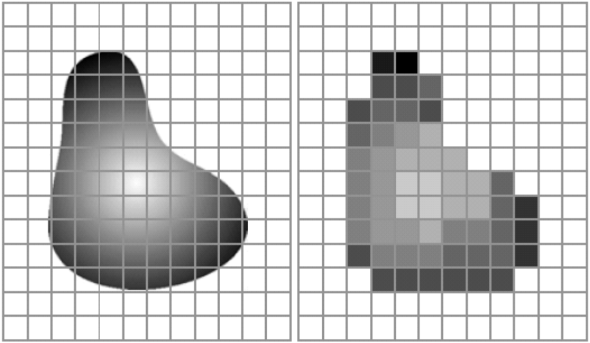
- Generally, an image can be represented by a two-dimensional function, $f(x, y)$, where x and y are spatial coordinates.
- The meaning of f in a given point in space, (x, y), depends on the source that generated the image (visible light, x-rays, ultrasound, radar, ...).
- Nevertheless, we generally assume that $f(x, y) \geq 0$.
- Moreover, both the spatial coordinates and the function values are continuous quantities.
- Therefore, to convert $f(x, y)$ into a digital image, it is necessary to perform spatial sampling and amplitude quantization.

Digitalization: sampling + quantization

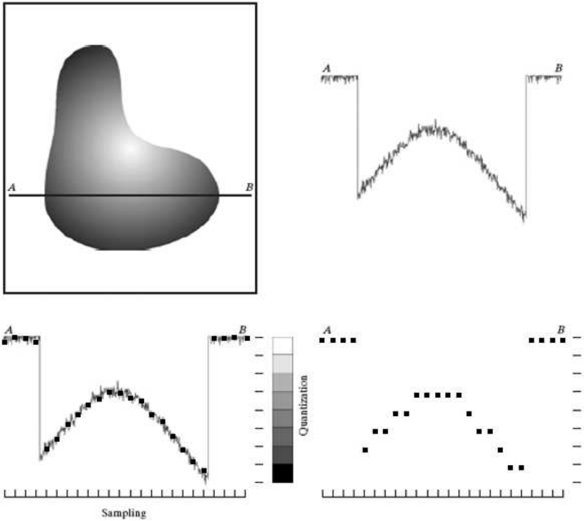


0100 1011 1010 0001 0011 0101 1010...

- Sampling and quantization — example:
  (Gonzalez & Woods)

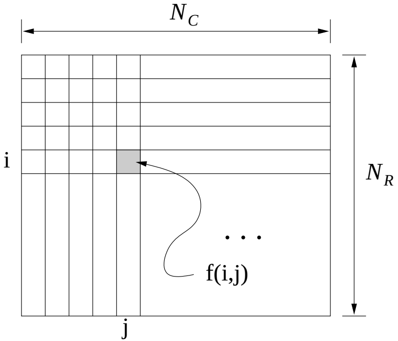


- Sampling and quantization — example:
  (Gonzalez & Woods)



Digital images

- Typically, a digital image is represented by a rectangular matrix of scalars or vectors.



- The $f(i,j)$ are named *pixels* and, usually, $f(i,j) \in \mathcal{I} \subset \mathbb{N}_0^n$.

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.
    $f(i,j) \in \{0,1,\ldots,2^b-1\} \xrightarrow{\alpha} \mathcal{I} \subset \{0,1,\ldots,2^{b'}-1\}^3$
  - Color images (for example, RGB images).
    $f(i,j) \in \{0,1,\ldots,2^b-1\}^3$

Examples

Color

Color-indexed (256)

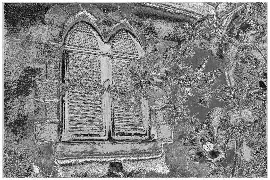
Grayscale (256)

Black and white

Color-indexed images

- Usually, images having a reduced set of colors are represented using a matrix of indexes (the index image) and a color table.

=



Index image

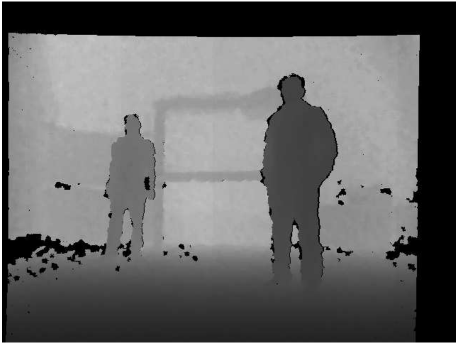
+

	R	G	B
0	100	035	203
1	030	025	200
2	167	205	010
3	132	219	045
...			

Color table

Infrared and depth images . . .

An Infrared image (Gobi Camera)



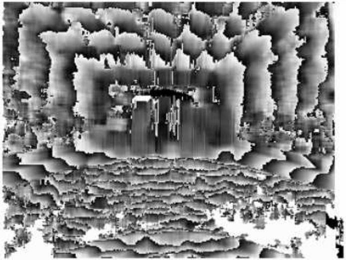
A Depth image (Kinect sensor)

Stereo vision

Left

Right

Rectified image



Depth image

Digital video

- 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 both spatial and temporal sampling, besides amplitude quantization.
- Therefore, a digital video is a temporal sequence of digital images which we represent by $v(i, j, k)$, with $k = t/T, k \in \mathbb{N}_0$.
- $T \in \mathbb{R}$ indicates the period of time between two consecutive images (we call them frames). Therefore, $1/T$ (Hz) is the frame rate.
- Sometimes we will refer to video fields. They occur in interlaced video and are made of the even (odd) lines of a frame.

Contents

Spectral colors (pure colors)

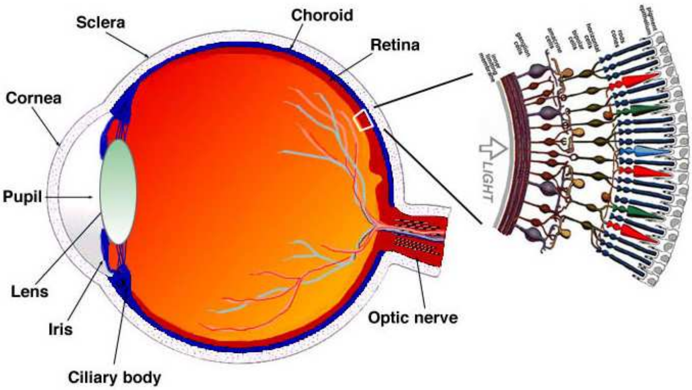
Cor	Wavelength
Violet	$\approx$ 380–440 nm
Blue	$\approx$ 440–485 nm
Cyan	$\approx$ 485–500 nm
Green	$\approx$ 500–565 nm
Yellow	$\approx$ 565–590 nm
Orange	$\approx$ 590–625 nm
Red	$\approx$ 625–740 nm

The human perception of color

- Normally, the characteristics that allow colors to be distinguished are:
  - The brightness (how bright is the color).
  - The hue (the dominant color).
  - The saturation (how pure is the color).
- Together, the hue and the saturation define the chromaticity.
- Therefore, a color can be characterized by the brightness and the chromaticity.

The human perception of color

- The human eye has photoreceptors that are sensitive to short wavelengths (S), medium wavelengths (M) and long wavelengths (L), also known as the blue, green and red photoreceptors.



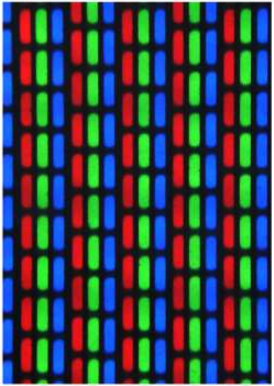
Additive primaries

- The red, green and blue are the three additive primary colors.

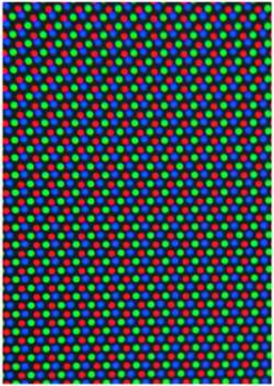
- Adding these three colors produces white.

The *RGB* color space

- Besides the use in acquisition on digital cameras, for example, the displays have pigments of these three colors…

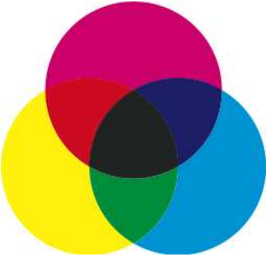


21" TV CRT Display



17" PC CRT Display

- The *CMY* color space is based on the subtractive properties of inks.
- The cyan, magenta and yellow are the subtractive primaries. They are the complements, respectively, of the red, green and blue. For example, the cyan subtracts the red from the white.



- Conversion from *RGB* to *CMY*: $C = 1 - R$, $M = 1 - G$, $Y = 1 - B$.

The *CMY* color space

C component

M component

Y component

- Due to technological difficulties regarding the reproduction of black, the *CMYK* color space is generally used for printing.

C component

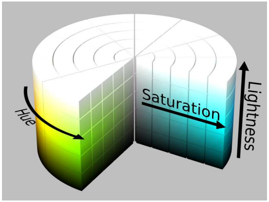
M component

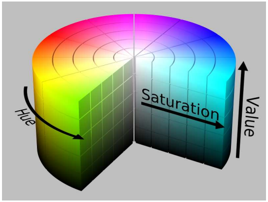
Y component

K component

The *HLS* and *HSV* color spaces

- The HSL and HSV are the two most common cylindrical coordinate representations of colors.
- They rearrange the geometry of RGB colors in an attempt to be more intuitive and perceptually relevant than the cartesian (cube) representation.
- They were developed in the 1970s for computer graphics applications, and are used for color pickers, in color-modification tools in image editing software, and commonly for image analysis and computer vision.





The *HLS* and *HSV* color spaces

- RGB to HSV:

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

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

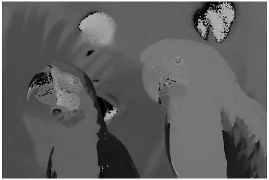
$$H = \begin{cases} 60(G-B)/S & if\ V = R \\ 120 + 60(B-R)/S & if\ V = G \\ 240 + 60(R-G)/S & if\ V = B \end{cases}$$

- RGB to HSL:

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

$$
\begin{aligned}
L &= \frac{V_{max} + V_{min}}{2} \\
S &= \begin{cases} \frac{V_{max} - V_{min}}{V_{max} + V_{min}} & \text{if } L < 0.5 \\ \frac{V_{max} - V_{min}}{2 - (V_{max} + V_{min})} & L \geq 0.5 \end{cases} \\
H &= \begin{cases} 60(G - B)/S & \text{if } V_{max} = R \\ 120 + 60(B - R)/S & \text{if } V_{max} = G \\ 240 + 60(R - G)/S & \text{if } V_{max} = B \end{cases}
\end{aligned}
$$

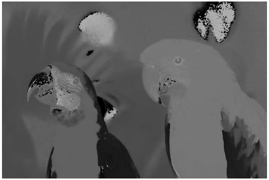
The *HLS* and *HSV* color spaces



H component

S component

V component



H component

S component

L component

The *YUV* color space

- The *YUV* color space is used in some television standards.
- Y is the luminance component:

$$Y = 0.299R + 0.587G + 0.114B$$

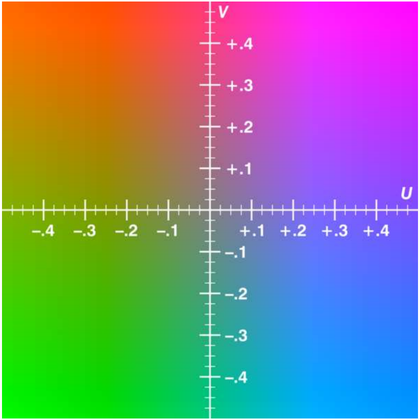
- Components U and V represent the chrominance:

$$\begin{aligned} U &= -0.147R - 0.289G + 0.436B = 0.492(B - Y) \\ V &= 0.615R - 0.515G - 0.100B = 0.877(R - Y) \end{aligned}$$

- For $R, G, B \in [0, 1]$, we have $Y \in [0, 1]$, $U \in [-0.436, 0.436]$ and $V \in [-0.615, 0.615]$.

The *YUV* color space

- $U - V$ plane, for a constant value of Y, equal to 0.5:



Advantages of the *YUV* color space

- The *YUV* color space allowed to maintain the compatibility with the old “black and white” television receivers.
- The human eye is more sensitive to the green color, which is represented mainly by the *Y* component.
- The *U* and *V* components are related to the blue and red.
- Since the human eye is less sensitive to the blue and red, it is possible to reduce the bandwidth used to represent the *U* and *V* components, without introducing significant perceptual degradation.

The YC_bC_r color space

- This is usually designated the digital version of YUV.
- The JPEG standard, as well as some other MPEG video standards, allows all 256 values in an 8 bits per component representation.
- In this case, considering $R, G, B \in \{0, \ldots, 255\}$, we have:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ C_b &= 128 - 0.168736R - 0.331264G + 0.5B \\ C_r &= 128 + 0.5R - 0.418688G - 0.081312B \end{aligned}$$

- After the conversion, $Y, C_b, C_r \in \{0, \ldots, 255\}$.
- Besides its use in image and video coding, this color space is also used in some computer vision applications.

The YC_bC_r color space

Y component

C_b component

C_r component

Contents

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 (2)

- 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 - attention of the matrix is stored in a continues manner `isContinuous()` 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_`, ...