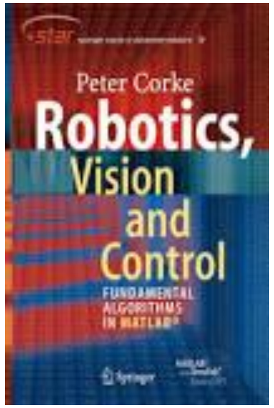
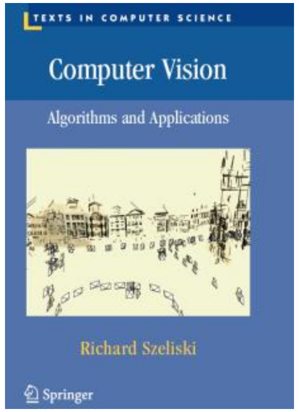
Introduction to Computer Vision

Dr Tomasz Luczynski

Textbooks





- Peter Corke, Robotics, Vision and Control
- Richard Szeliski, Computer Vision: Algorithms and Applications http://szeliski.org/Book/
- *Richard Hartley, Andrew Zisserman: multiple view geometry in computer vision
- *Yi Ma, Stefano Soatto, Jana Kosecka, S Shankar Sasty: An Invitation to 3-D Vision

Other resources

- https://robotacademy.net.au
- https://jordicenzano.name/front-test/2d-3d-paradigm-overview-2011/camera-model/
- https://www.youtube.com/playlist?list=PLgnQpQtFTOGRsi5vzy9PiQpNWHjq-bKN1

Schedule for weeks 5&6

	MONDAY			THURSDAY	FRIDAY
	1015 - 1115	**1315 - 1515**	**1615 - 1715**	**1615 - 1715**	**1315 - 1515**
Week 1	**LECTURE:** - image formation - camera model - distortions - homography	**LAB:** camera calibration: hands on with different tools (matlab, ROS,...), tips and tricks	**LECTURE:** Intro and presentations assignment: - P1: colour spaces - P2: filters - P3: feature extraction	**LECTURE:** Stereo vision: - multiple view reconstruction basics - epipolar geometry - dense/sparse matching - quantization error - 3D video encoding	**LECTURE:** - Student presentations - Discussion / Q&A
Week 2	**LECTURE:** - Mini quiz - Description of the lab assignment + discussion	**LAB:** Work on the assignment (presence facultative, studets can work from home)			**LAB:** Work on the assignment and presentation of the results

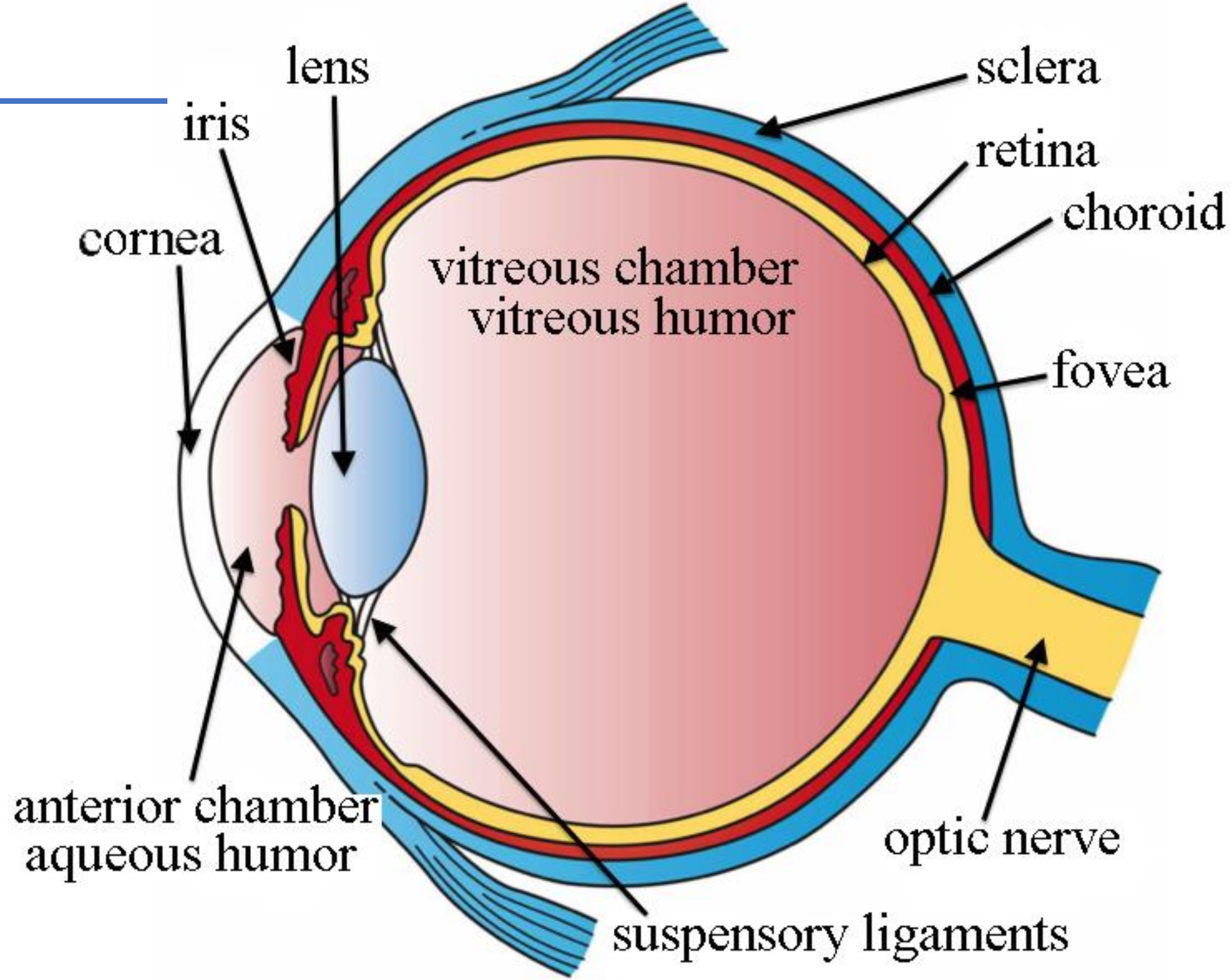
Schedule for weeks 5&6

11:15-13:15

	MONDAY			THURSDAY	FRIDAY
	1015 - 1115	**1315 - 1515**	**1615 - 1715**	**1615 - 1715**	**1315 - 1515**
Week 1	**LECTURE:** - image formation - camera model - distortions - homography	**LAB:** camera calibration: hands on with different tools (matlab, ROS,...), tips and tricks	**LECTURE:** Intro and presentations assignment: - P1: colour spaces - P2: filters - P3: feature extraction	**LECTURE:** Stereo vision: - multiple view reconstruction basics - epipolar geometry - dense/sparse matching - quantization error - 3D video encoding	**LECTURE:** - Student presentations - Discussion / Q&A
Week 2	**LECTURE:** - Mini quiz - Description of the lab assignment + discussion	**LAB:** Work on the assignment (presence facultative, studets can work from home)			**LAB:** Work on the assignment and presentation of the results

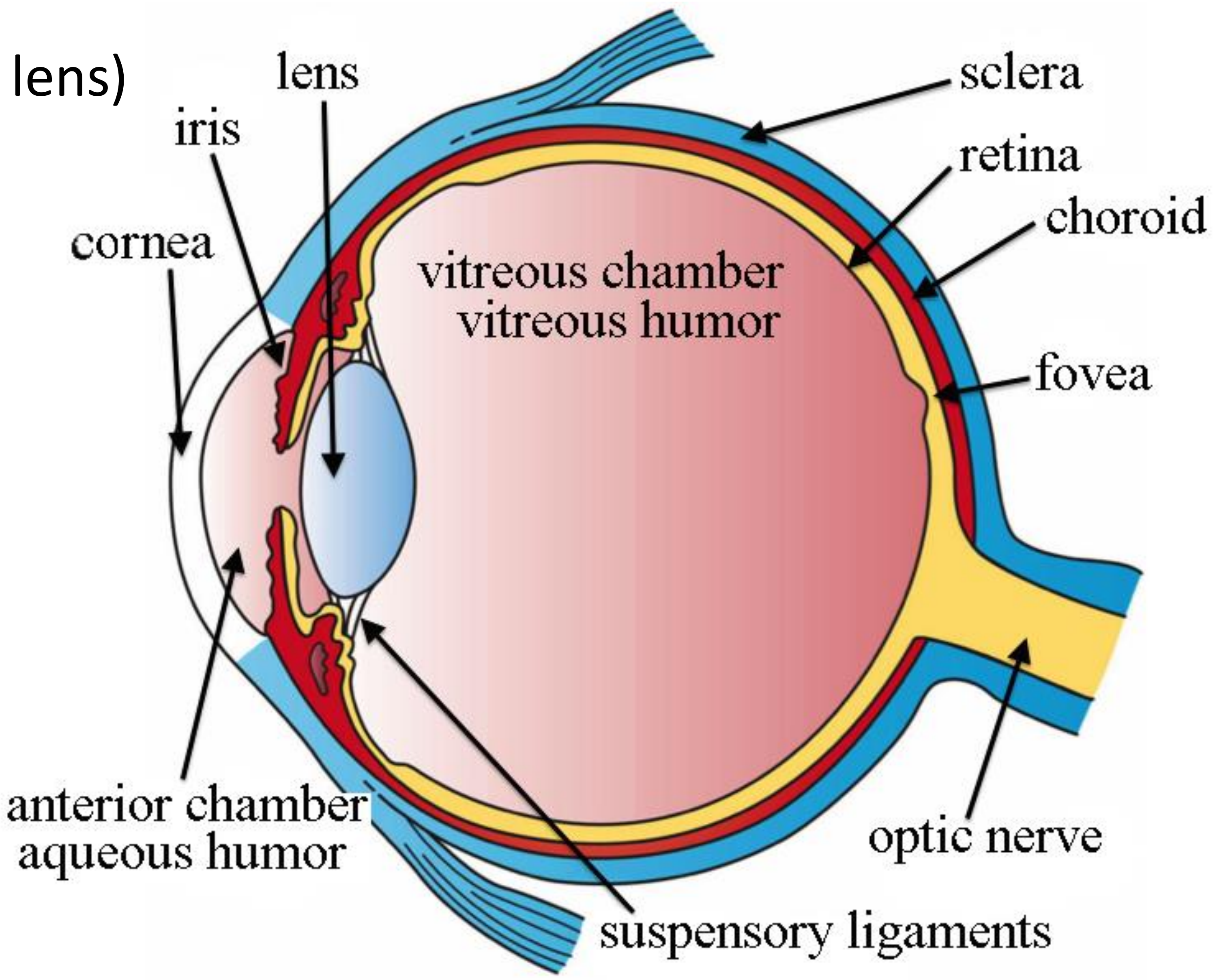
Image formation and camera modelling & calibration

Human eye

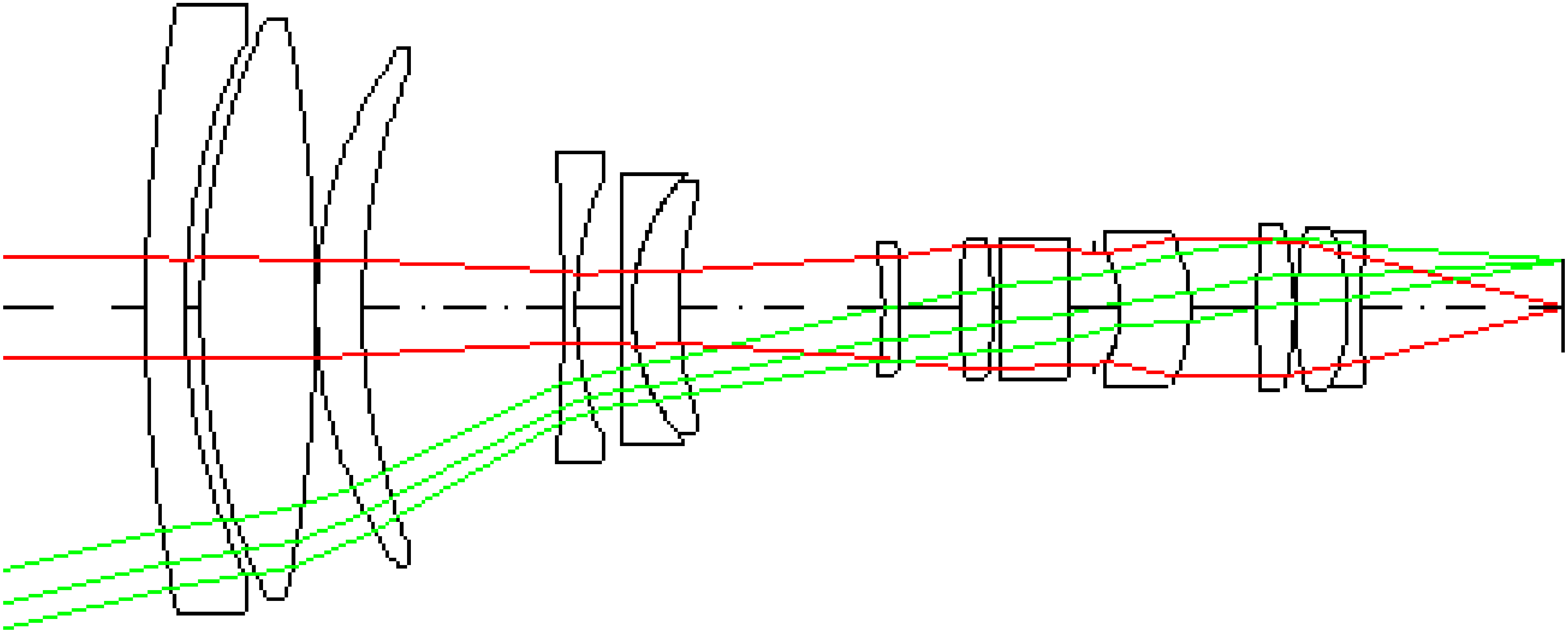

lens
sclera
iris
retina
choroid
cornea
vitreous chamber
vitreous humor
fovea
anterior chamber
aqueous humor
optic nerve
suspensory ligaments

Camera

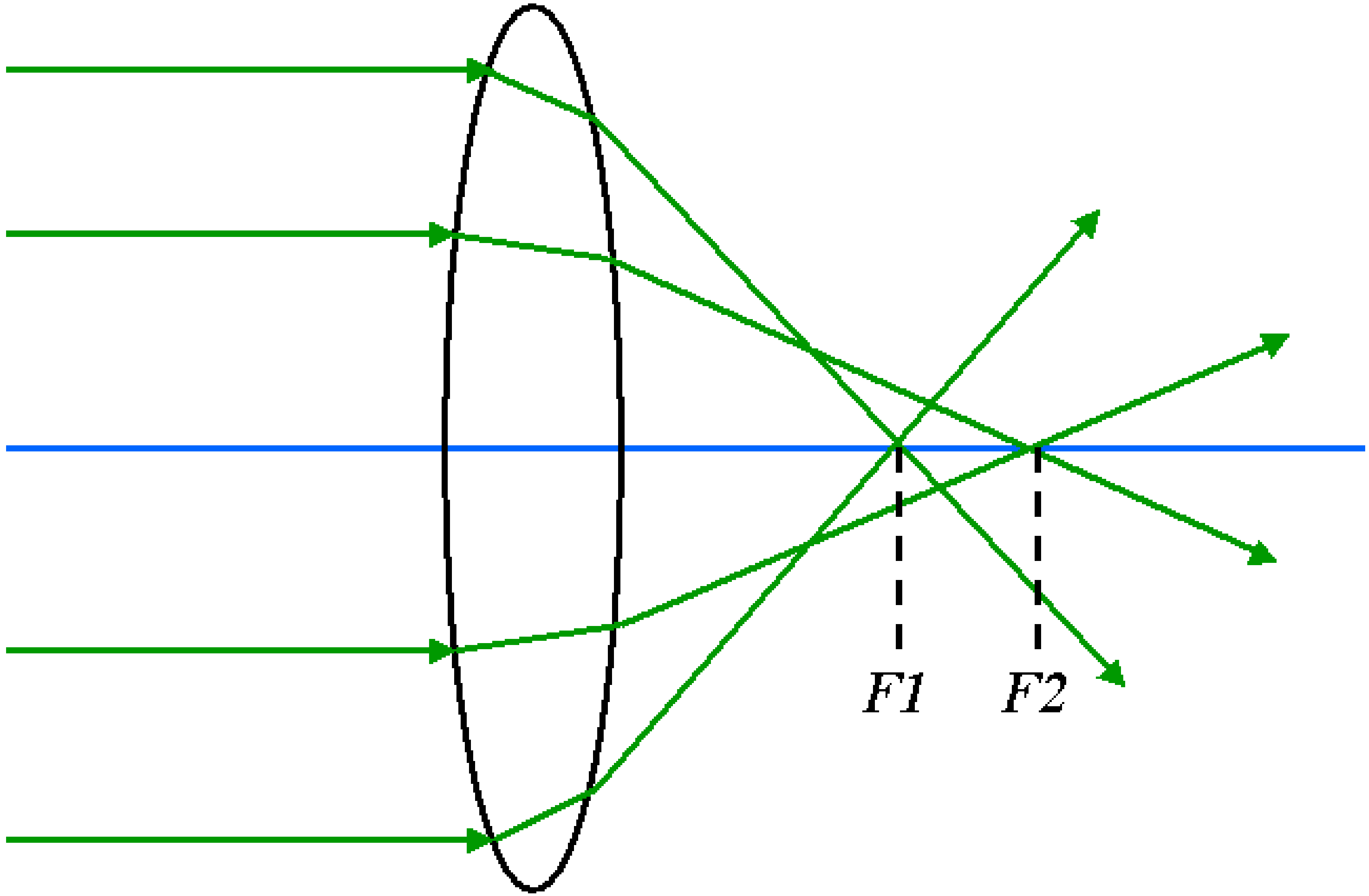
- Camera lens (cornea and the lens)
- Aperture ring (iris)
- Matrix (retina)



Camera lens



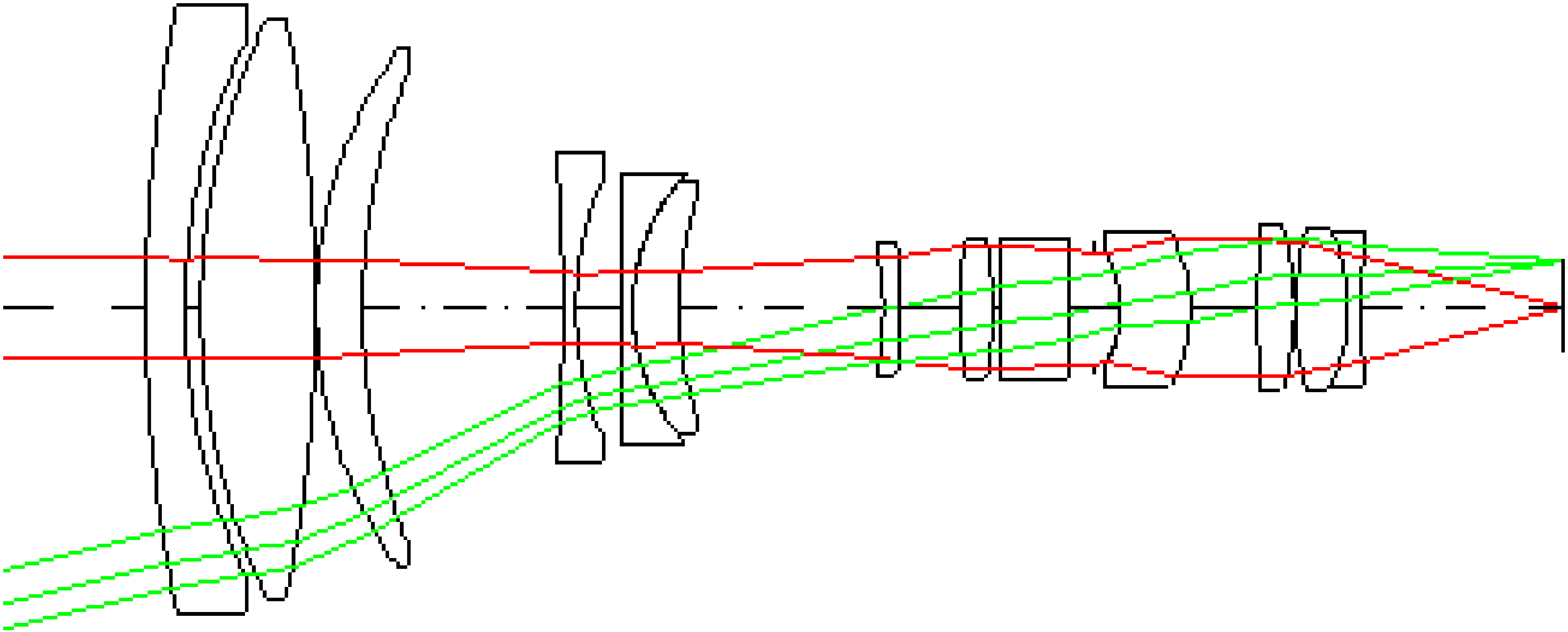
Spherical aberration



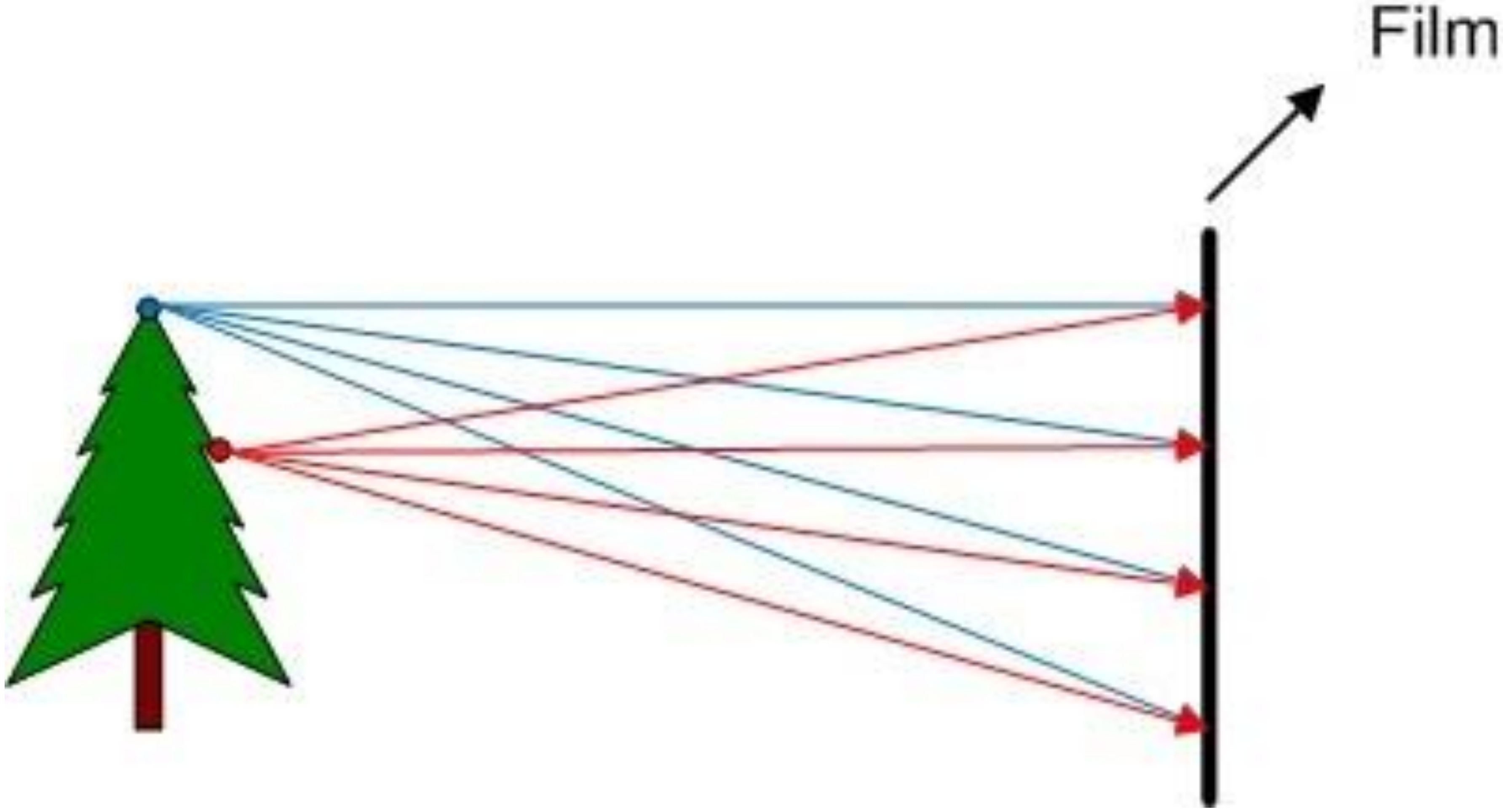
Chromatic aberration and vignetting

Geometrical distortions

Camera model

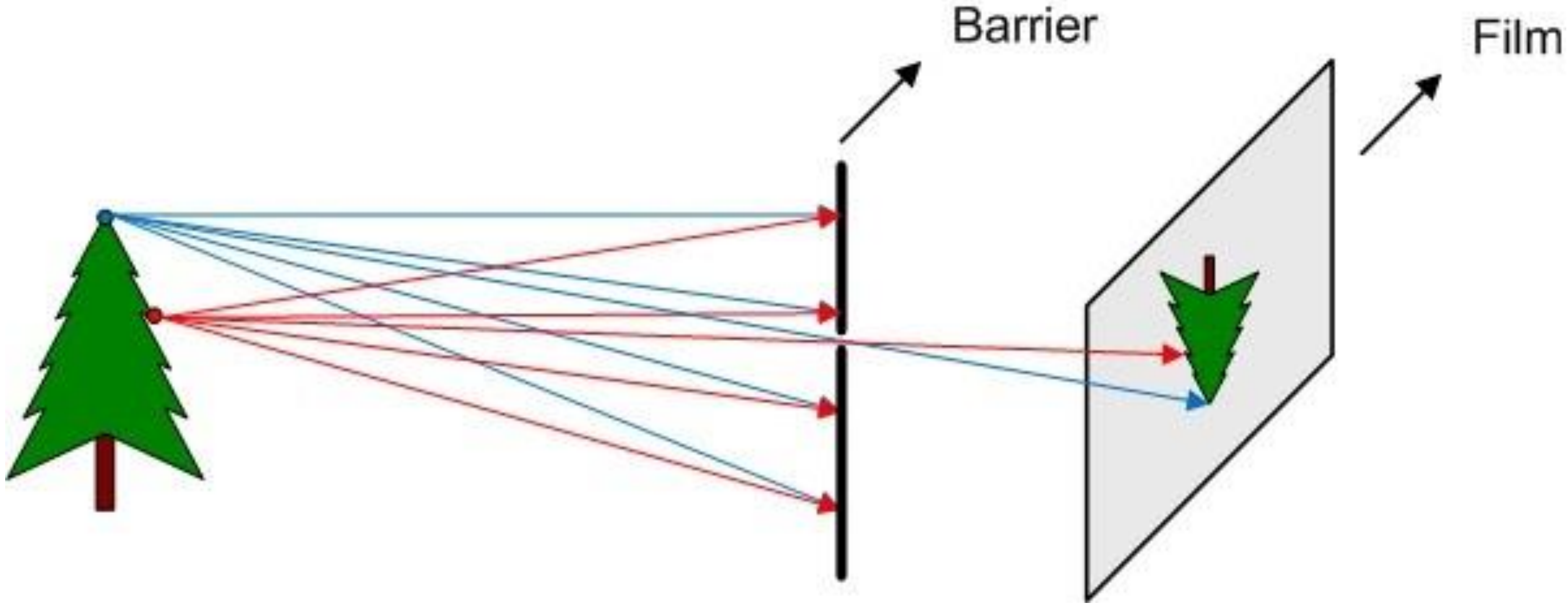


Pinhole camera model

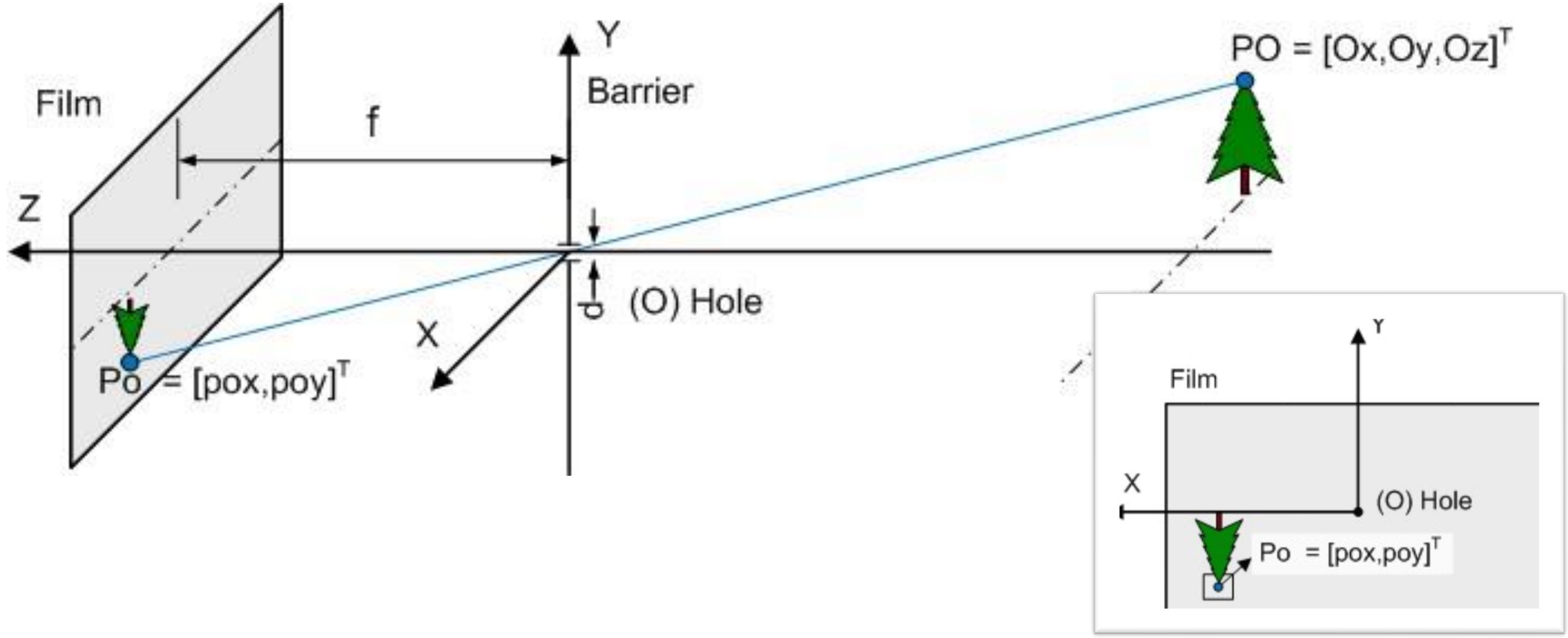


Images from the blog: https://jordicenzano.name/front-test/2d-3d-paradigm-overview-2011/camera-model/

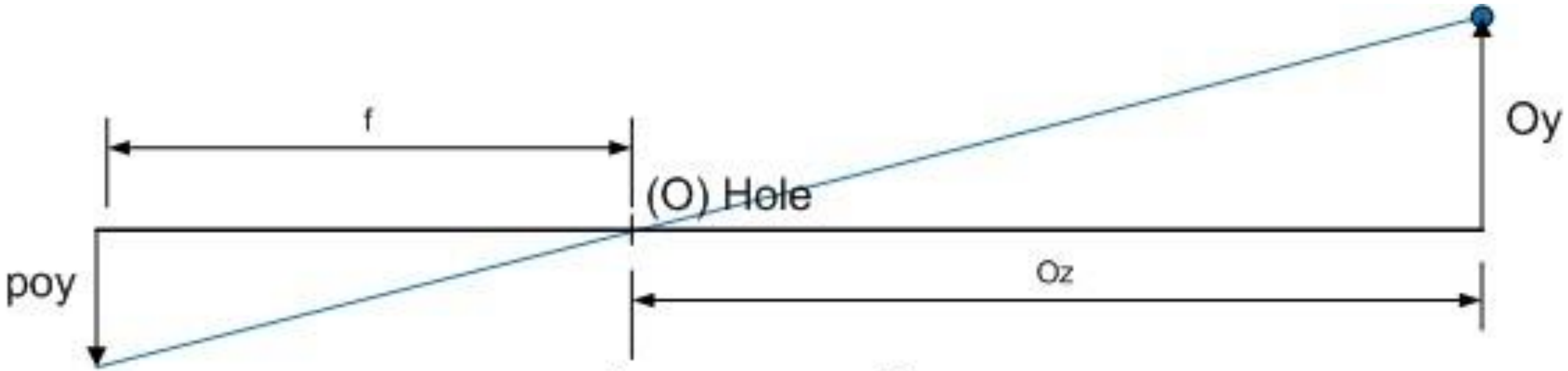
Pinhole camera model



Pinhole camera model - perspective projection equations



Pinhole camera model - perspective projection equations



$$\frac{poy}{f} = \frac{Oy}{Oz} \Rightarrow poy = f\,\frac{Oy}{Oz}$$

$$\frac{pox}{f} = \frac{Ox}{Oz} \Rightarrow pox = f\,\frac{Ox}{Oz}$$

Pinhole camera properties

Pinhole camera properties

- Line preserving: straight lines should remain straight
- Not length preserving
- Not angle preserving

Projective geometry

- https://youtu.be/ZNB6SpEBnBQ
- Camera projects 3D information to a 2D plane
- During this projection there is a loss of information
- Therefore 3D information may be recovered when:

Projective geometry

- https://youtu.be/ZNB6SpEBnBQ
- Camera projects 3D information to a 2D plane
- During this projection there is a loss of information
- Therefore 3D information may be recovered when we know the camera parameters and:
    - There are multiple images, or
    - The distance is known, or
    - The dimensions of the object are known

Vanishing points

- What is point in the infinity?
- In the euclidean space (cartesian coordinates) it is possible to set x/y/z coordinates to infinity, but this way we loose the information about the direction
- Homogenous coordinates (H.C.) and perspective geometry address this and other issues
- H.C. are designed especially to describe 2D<->3D projections

Homogeneous coordinates

- Definition: The representation x of a geometric object is homogeneous if x and sx (where s is a scalar) represent the same object for $s \neq 0$
- Example:
  - Homogeneous: $x=sx$
  - Euclidean: $x \neq sx$
- In practice we use one dimension more

Euclidean <-> H.C.

- Example in 2D
  - $X_{euc}=[x , y]^T$ -> $X_{hc}=[x , y , 1]^T$

  - $X_{hc}=[u , v , w]^T = [u/w , v/w , 1]^T$ -> $X_{euc} =[u/w , v/w]^T$

- Example in 3D
  - $X_{euc}=[x , y , z]^T$ -> $X_{hc}=[x , y , z , 1]^T$

  - $X_{hc}=[t , u , v , w]^T = [t/w , u/w , v/w , 1]^T$ -> $X_{euc} =[t/w , u/w , v/w]^T$

- In H.C. at least one coordinate must be $\neq 0$

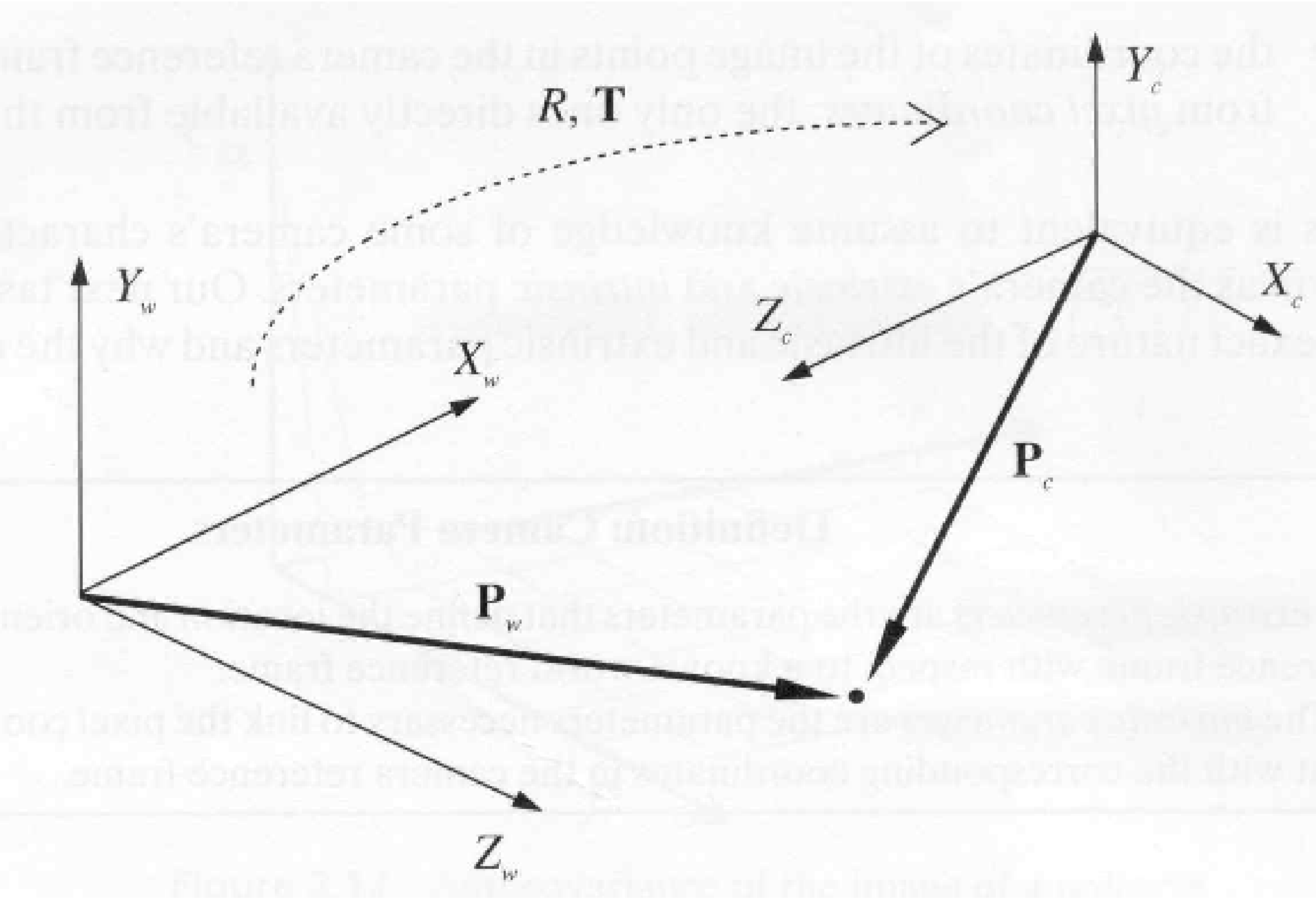
Camera extrinsic matrix

- https://youtu.be/DX2GooBIESs
- https://www.cse.unr.edu/~bebis/CS791E/Notes/CameraParameters.pdf
- Extrinsic matrix describes the position of the camera in the world
- This transformation can be inverted
- **How many parameters are needed?**

Camera extrinsic matrix

- https://youtu.be/DX2GooBIESs
- https://www.cse.unr.edu/~bebis/CS791E/Notes/CameraParameters.pdf
- Extrinsic matrix describes the position of the camera in the world
- This transformation can be inverted
- 6 parameters: 3 for the position and 3 for the heading

Camera extrinsic matrix



Camera extrinsic matrix

- Using the extrinsic camera parameters, we can find the relation between the coordinates of a point P in world (Pw) and camera (Pc) coordinates:

$$P_c = R(P_w - T) \text{ where } R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

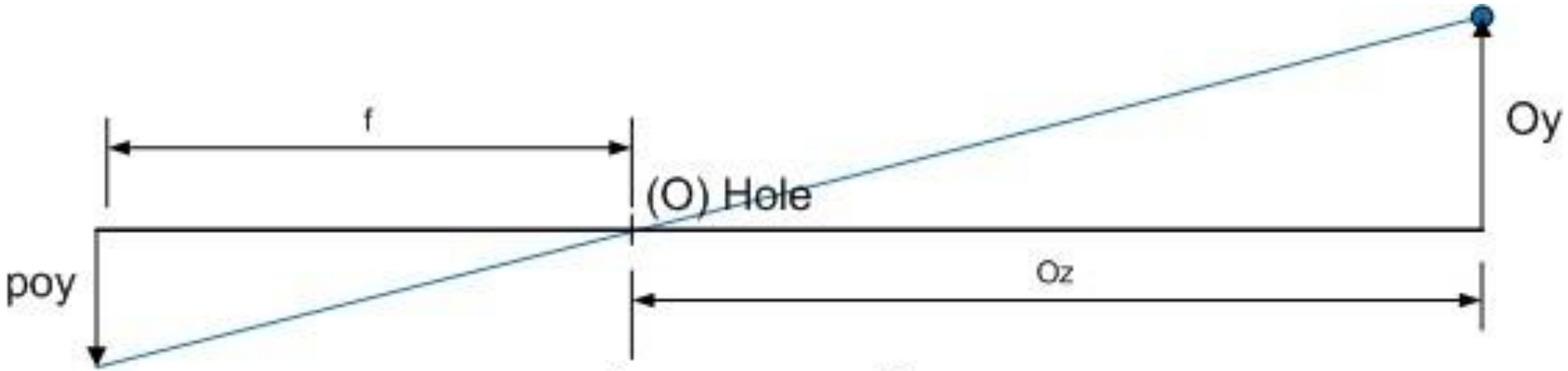
Camera extrinsic matrix

$$M_{ex} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix}$$

Intrinsic parameters

- Intrinsic matrix describes the mapping of the scene in front of the camera to the final image
- Characterize the optical, geometric, and digital characteristics of the camera:
    - the perspective projection (focal length f)
    - the transformation between image plane coordinates and pixel coordinates
    - the geometric distortion introduced by the optics.

Intrinsic parameters



$$\frac{poy}{f} = \frac{Oy}{Oz} \Rightarrow poy = f\,\frac{Oy}{Oz}$$

$$\frac{pox}{f} = \frac{Ox}{Oz} \Rightarrow pox = f\,\frac{Ox}{Oz}$$

Intrinsic matrix

$$M_{in} = \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

"-" In front of the focal length comes from the assumed coordinate frame; f/s_x and f/s_y in the intrinsic matrix are positive values

Image distortions due to optics

- Assuming radial distortion:

$$x = x_d(1 + k_1 r^2 + k_2 r^4)$$

$$y = y_d(1 + k_1 r^2 + k_2 r^4)$$

3D -> 2D projection

- Using homogeneous coordinates:

$$\begin{bmatrix} x_h \\ y_h \\ w \end{bmatrix} = M_{in}\ M_{ex} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

3D -> 2D projection

$$\begin{bmatrix} x_h \\ y_h \\ w \end{bmatrix} = M_{in}\ M_{ex} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

M is called the projection matrix (it is a 3 x 4 matrix)

Camera calibration

- If the intrinsics are unknown we call the camera uncalibrated
- If the intrinsics are known, we call the camera calibrated
- The process of obtaining camera intrinsics (and distortion coefficients) is called camera calibration

Camera calibration tools

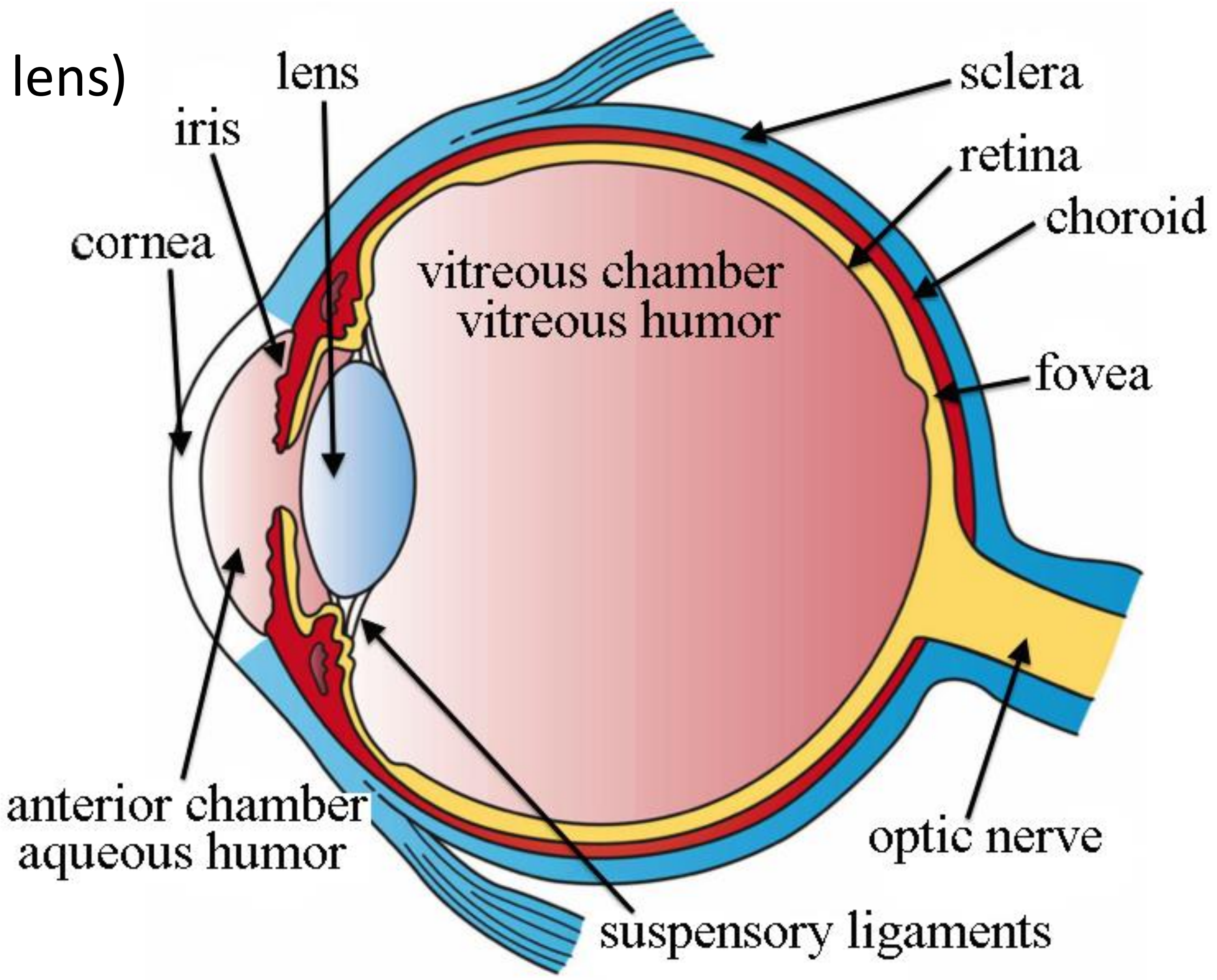
- Matlab
- ROS
- CamOdoCal
- MRPT
- …

Basics of image processing

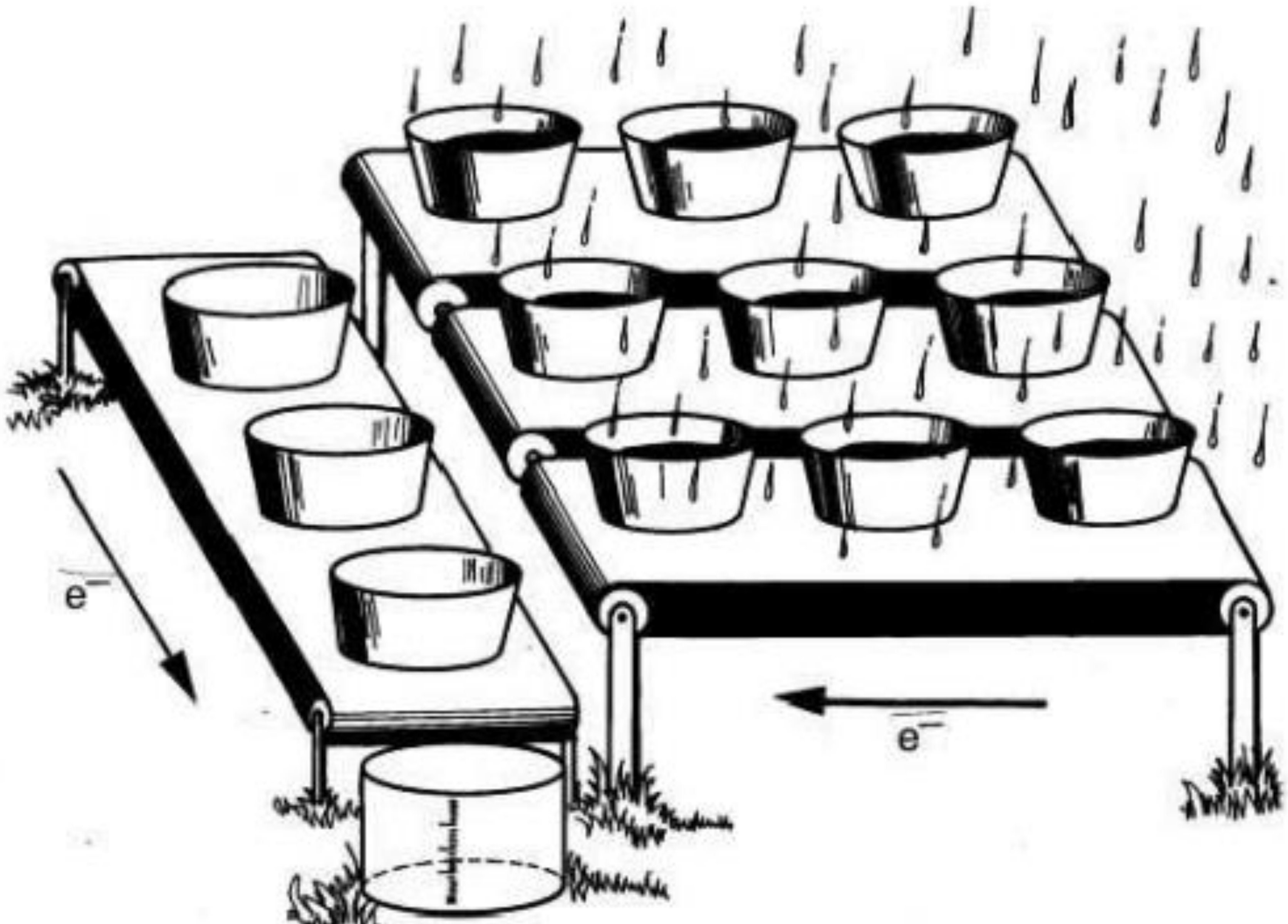
Dr Tomasz Luczynski

Camera

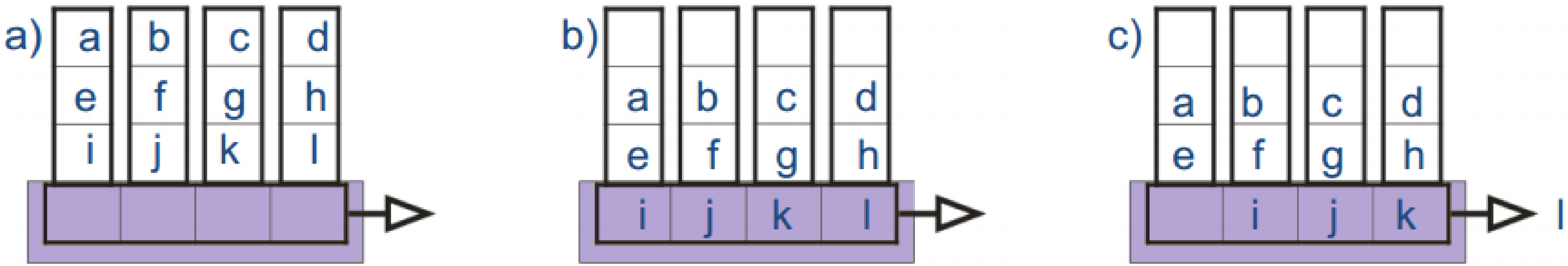
- Camera lens (cornea and the lens)
- Aperture ring (iris)
- Matrix (retina)



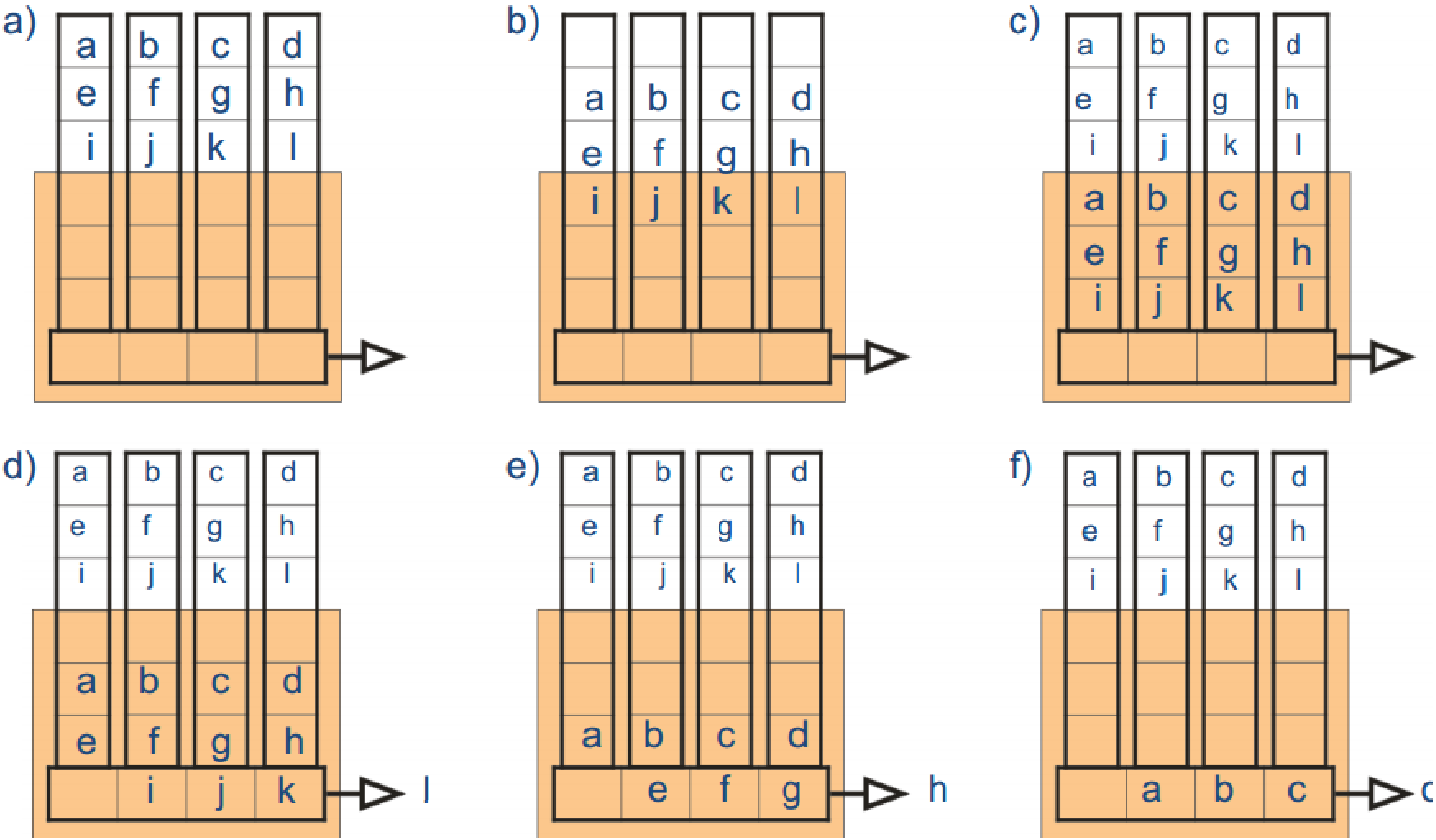
CCD matrix



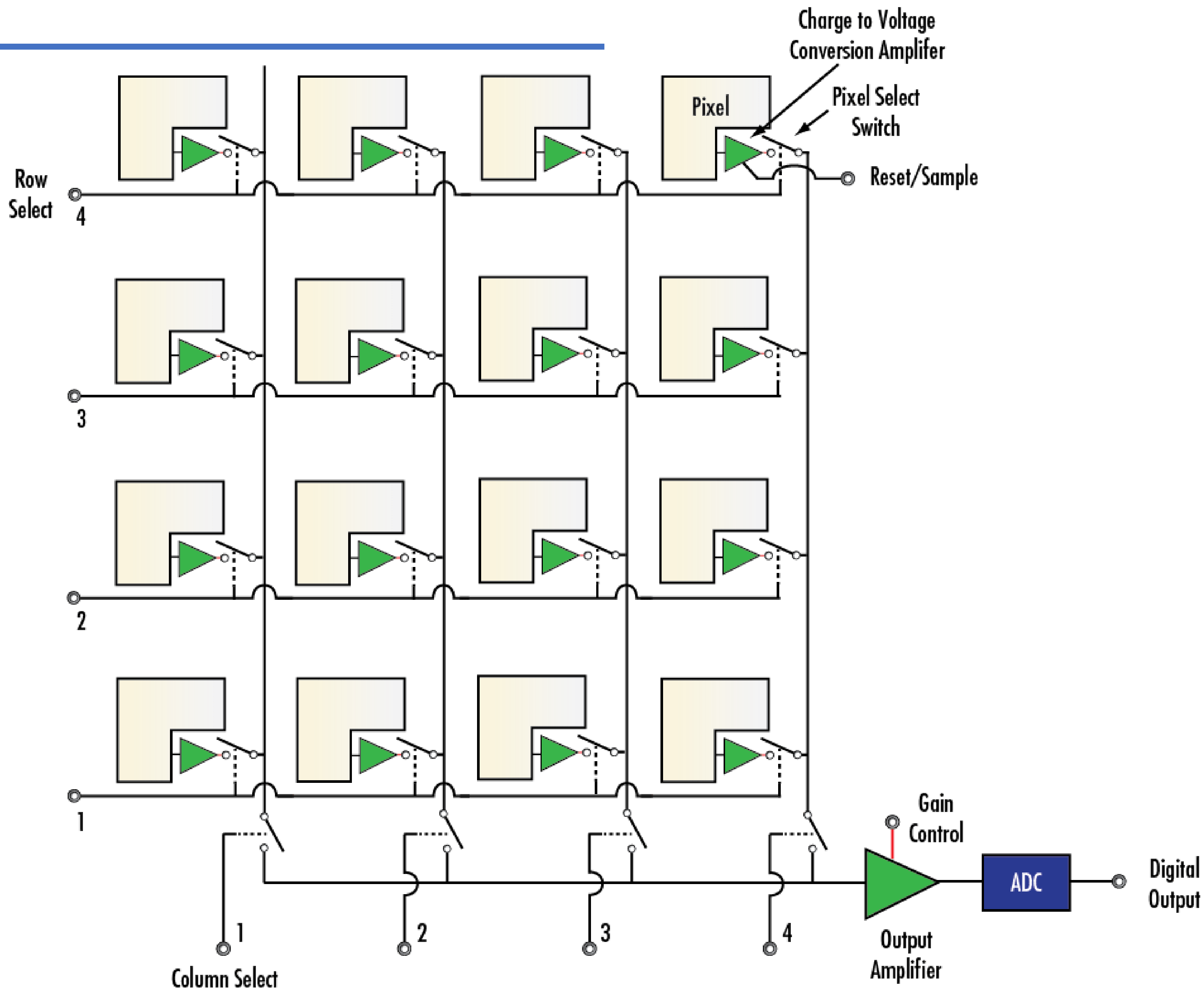
CCD matrix: progressive scan readout



CCD matrix: frame transfer



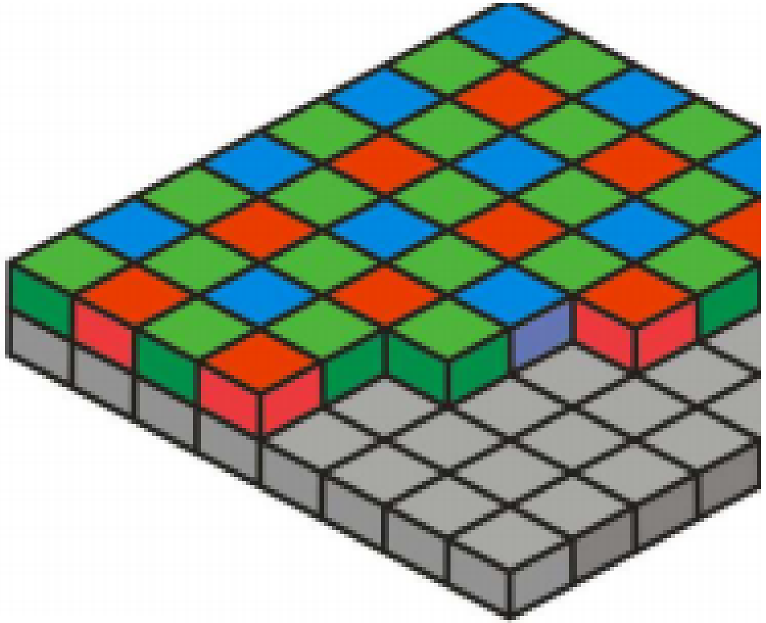
CMOS matrix

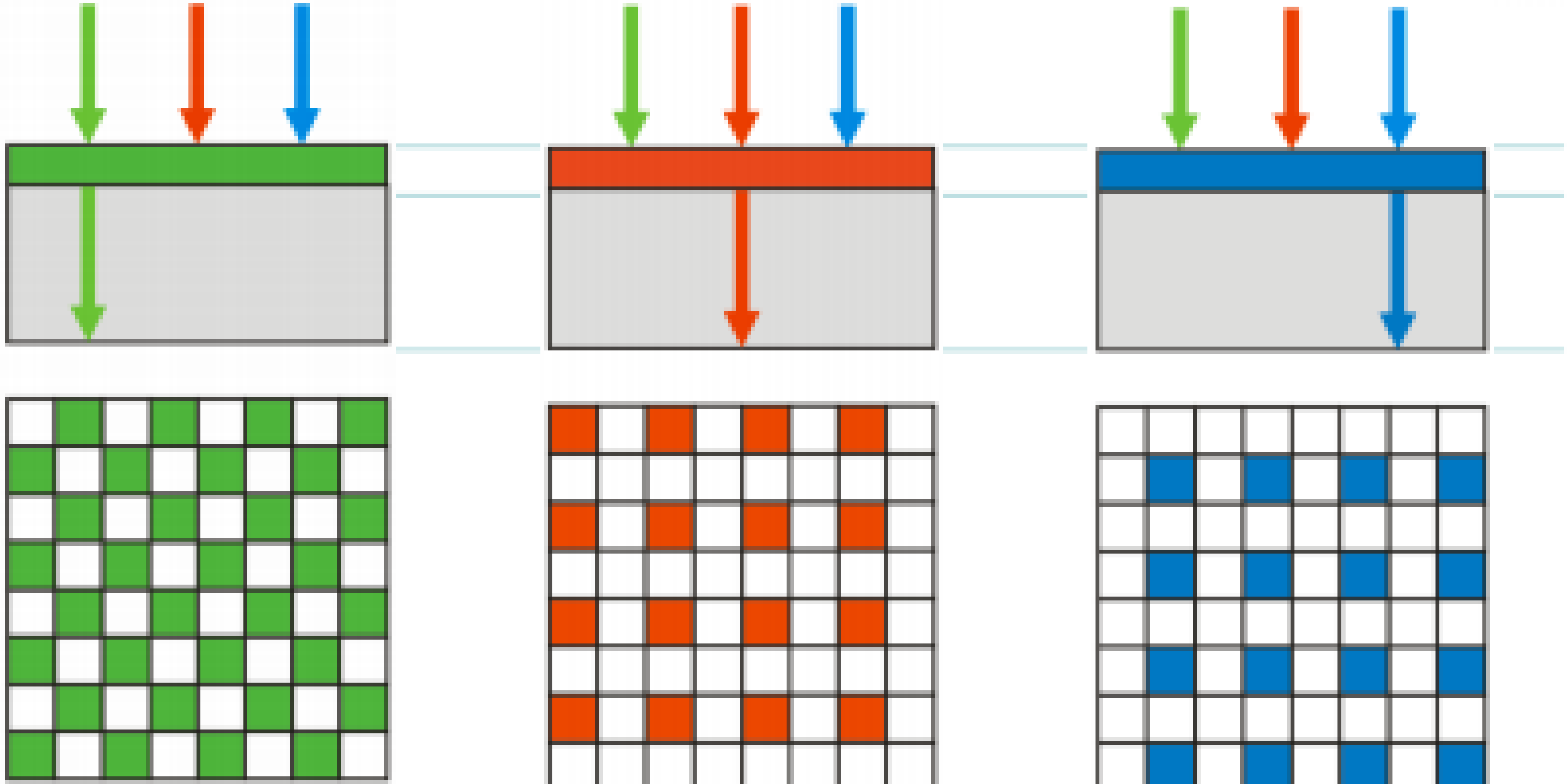


Charge to Voltage
Conversion Amplifer
Pixel
Pixel Select
Switch
Reset/Sample
Row
Select
4
3
2
1
Gain
Control
ADC
Digital
Output
1
2
3
4
Column Select
Output
Amplifier

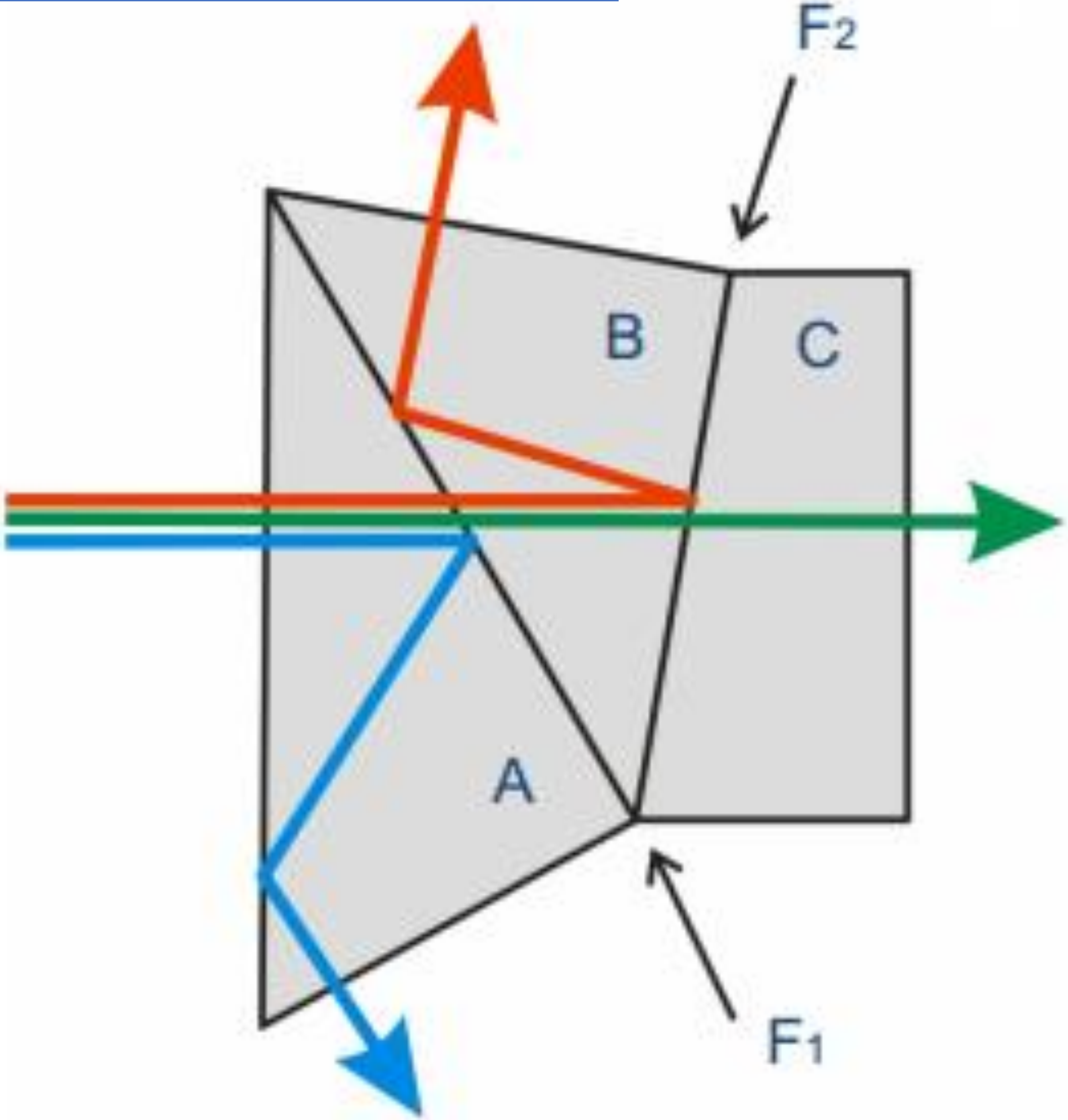
Rolling shutter effect

Colour registration

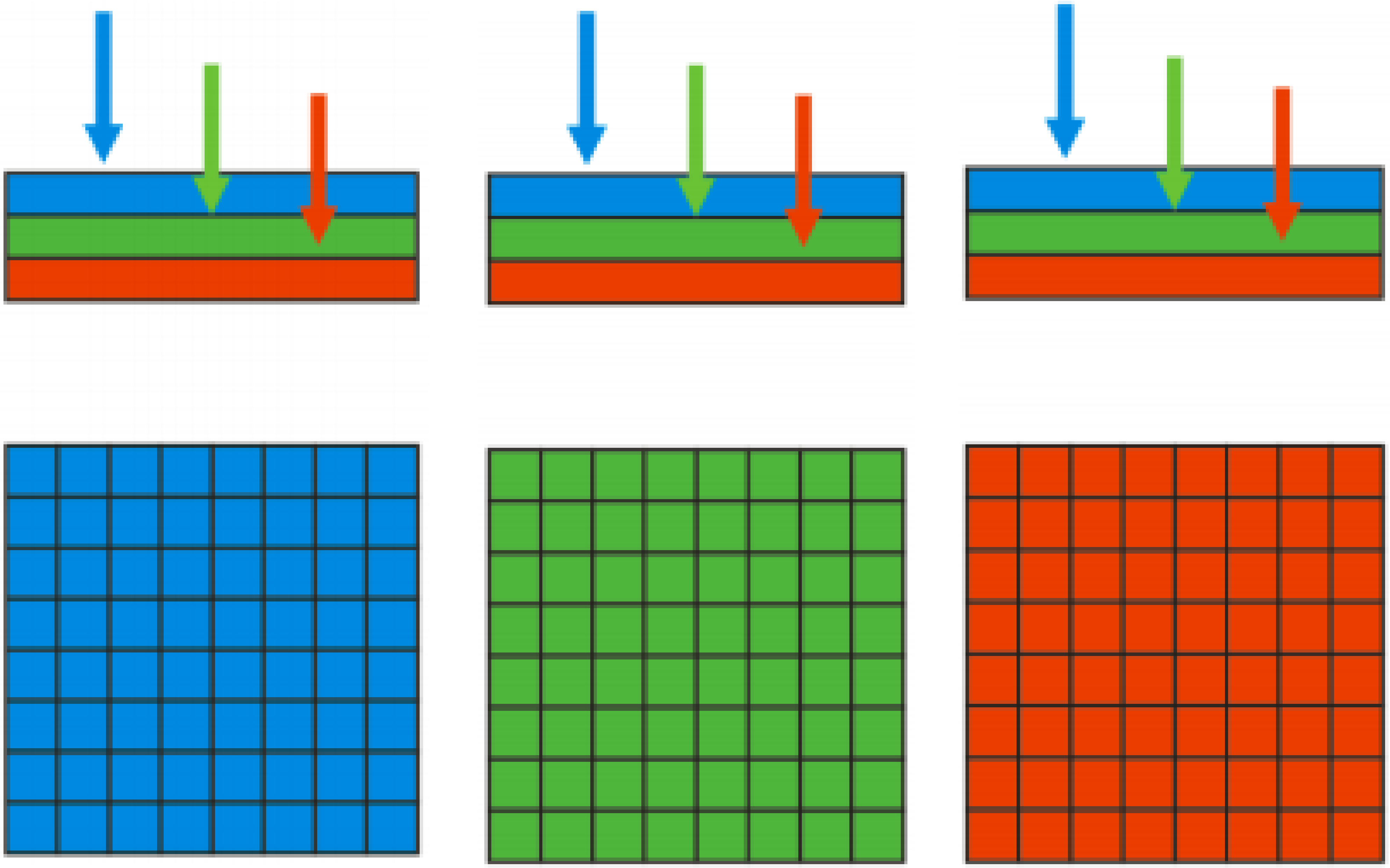




Colour registration

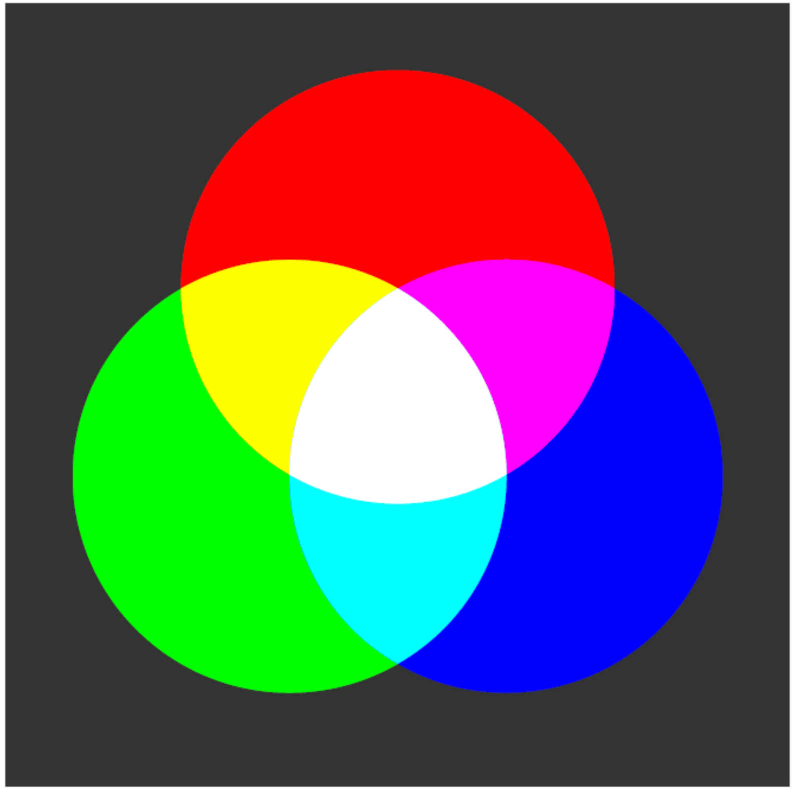


Colour registration



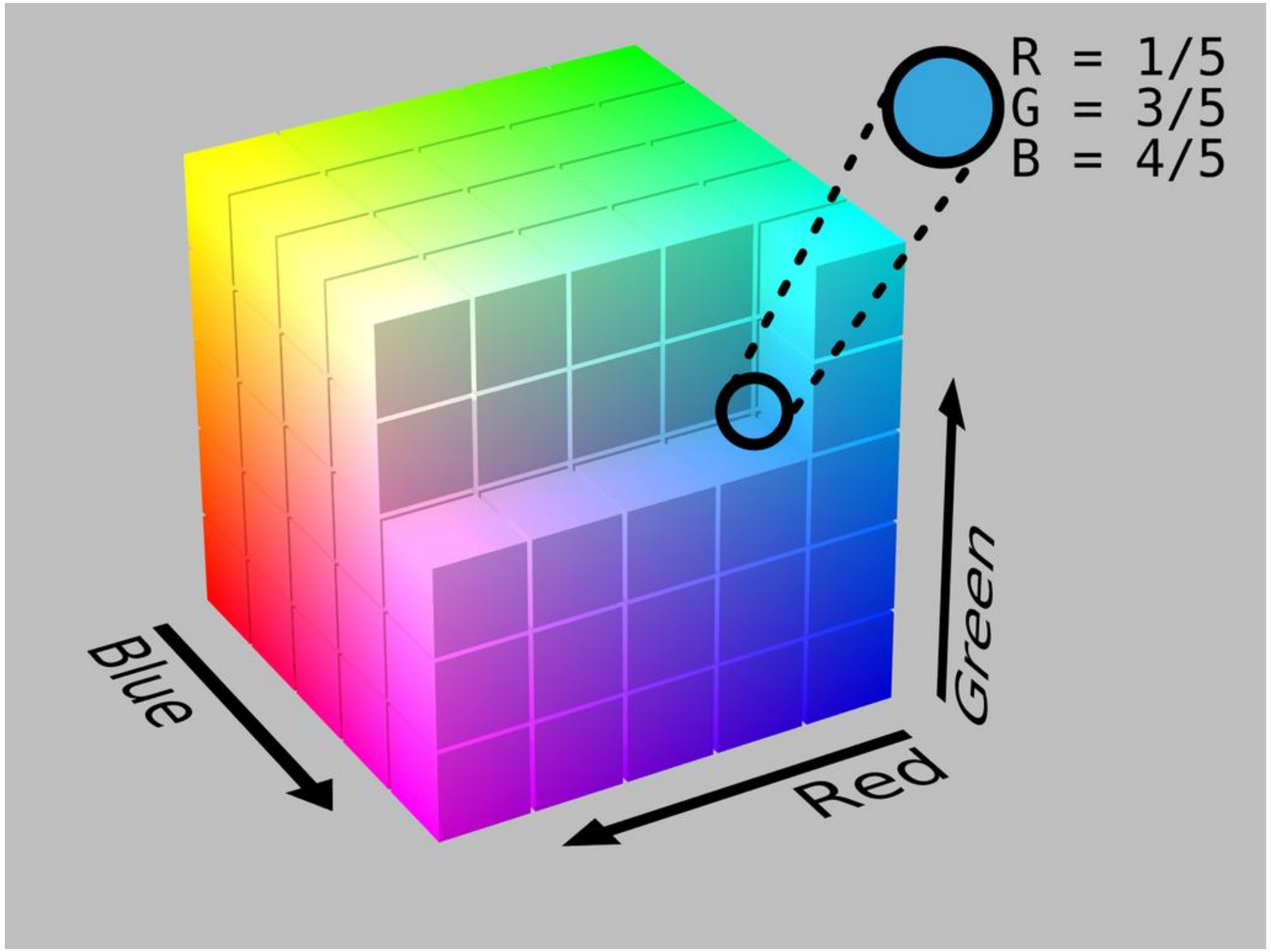
Colour spaces

- Different ways of encoding colour
- Based on different creation method, e.g. additive or substractive



Colour space: RGB

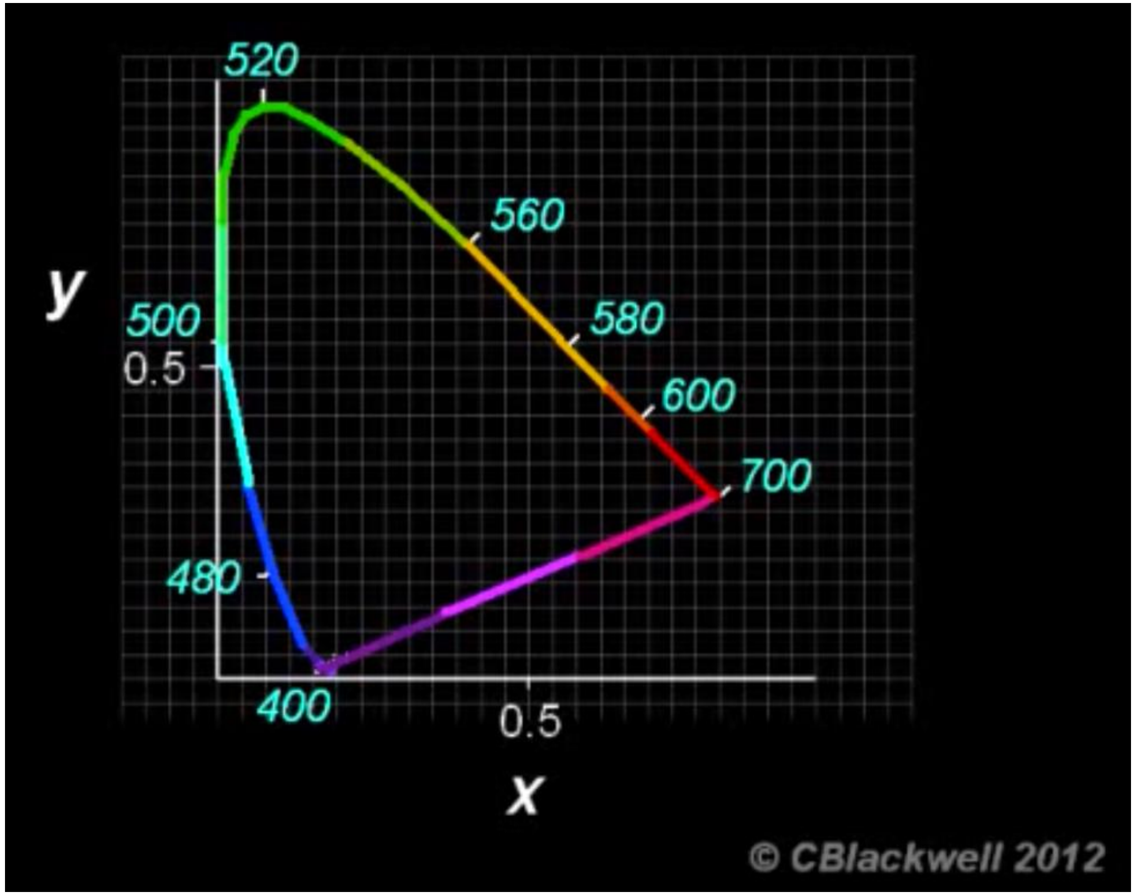
- Additive colour space
- Inspired by human vision



CIE xy chromaticity diagram and the CIE xyY colour space

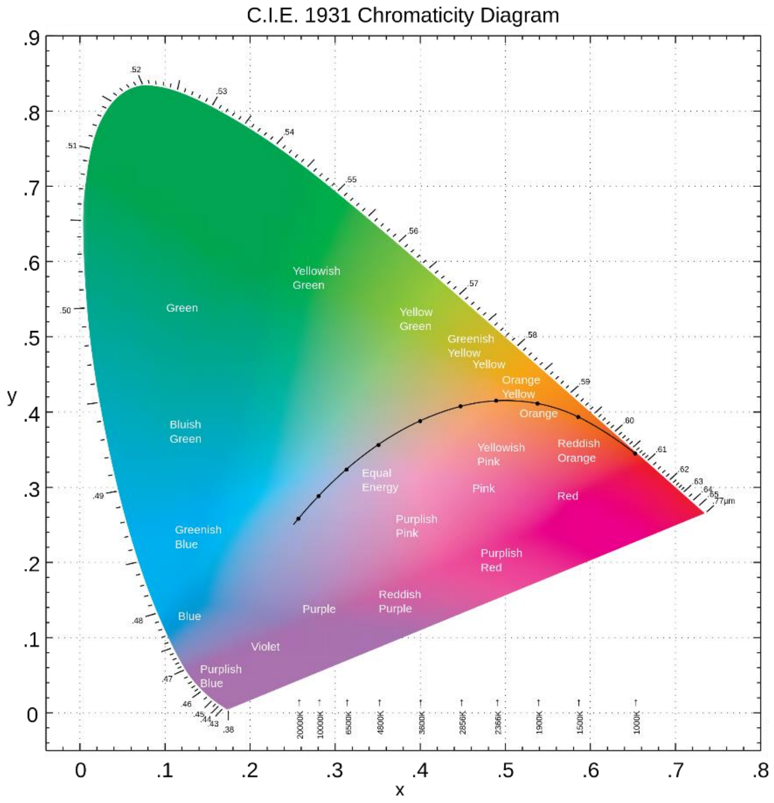
Experts: ABASSEBAY Djouzar and AUDRY Hanako

CIE xy chromaticity Diagram



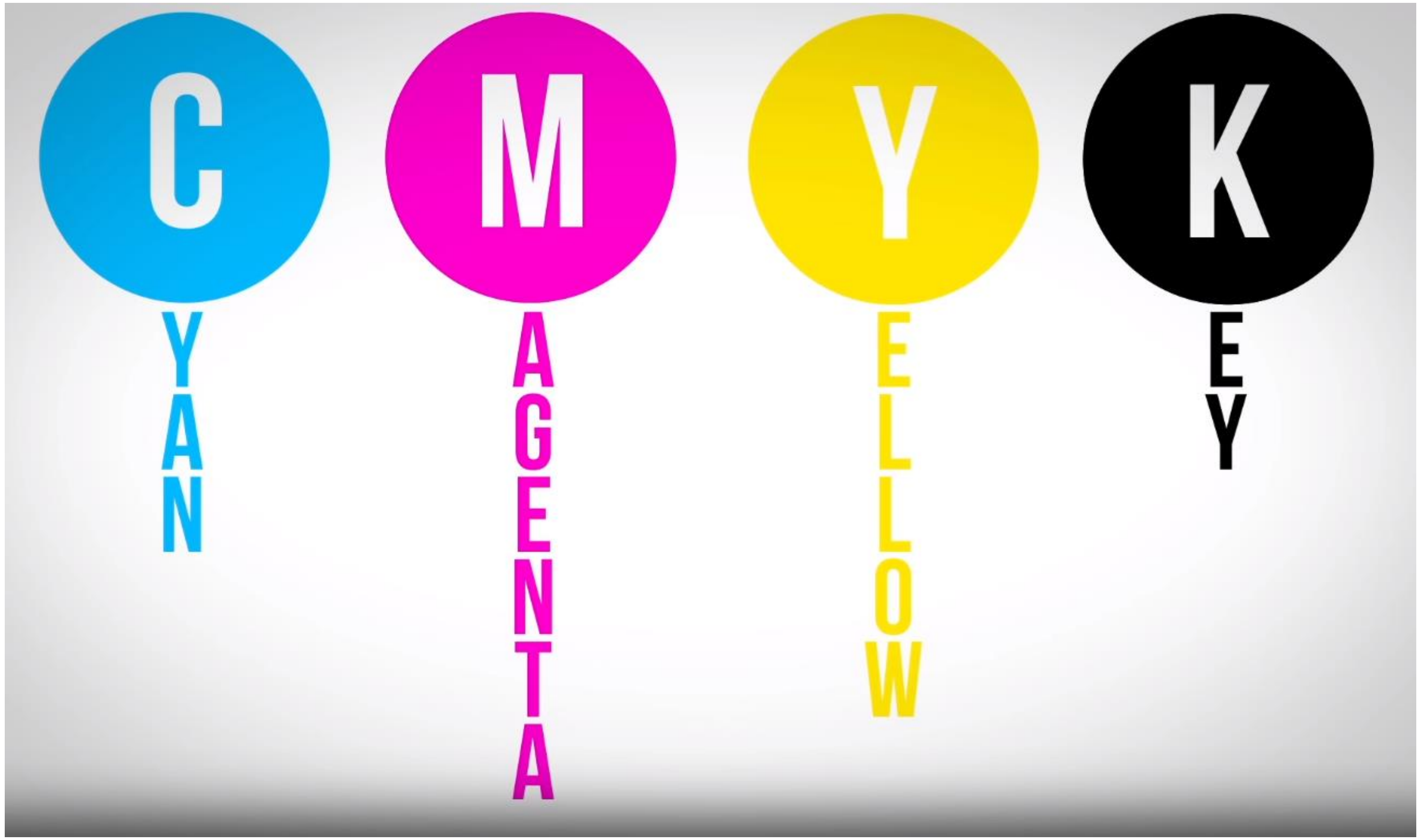
xyY colour space

- xy : chromaticity
- Y : luminance



CMYK: Subtractive Colouring

Expert: Christopher Baak

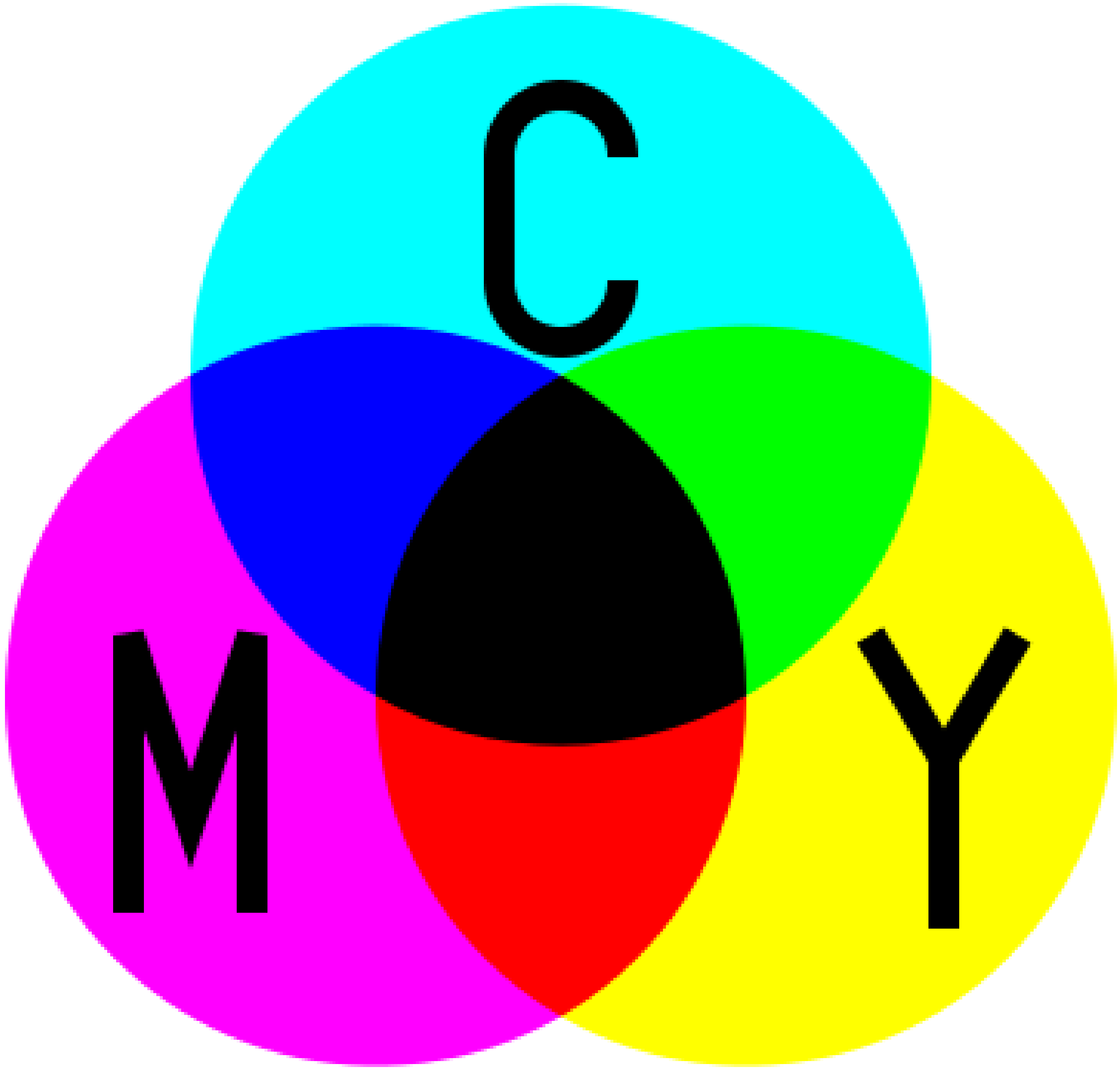


Cyan:

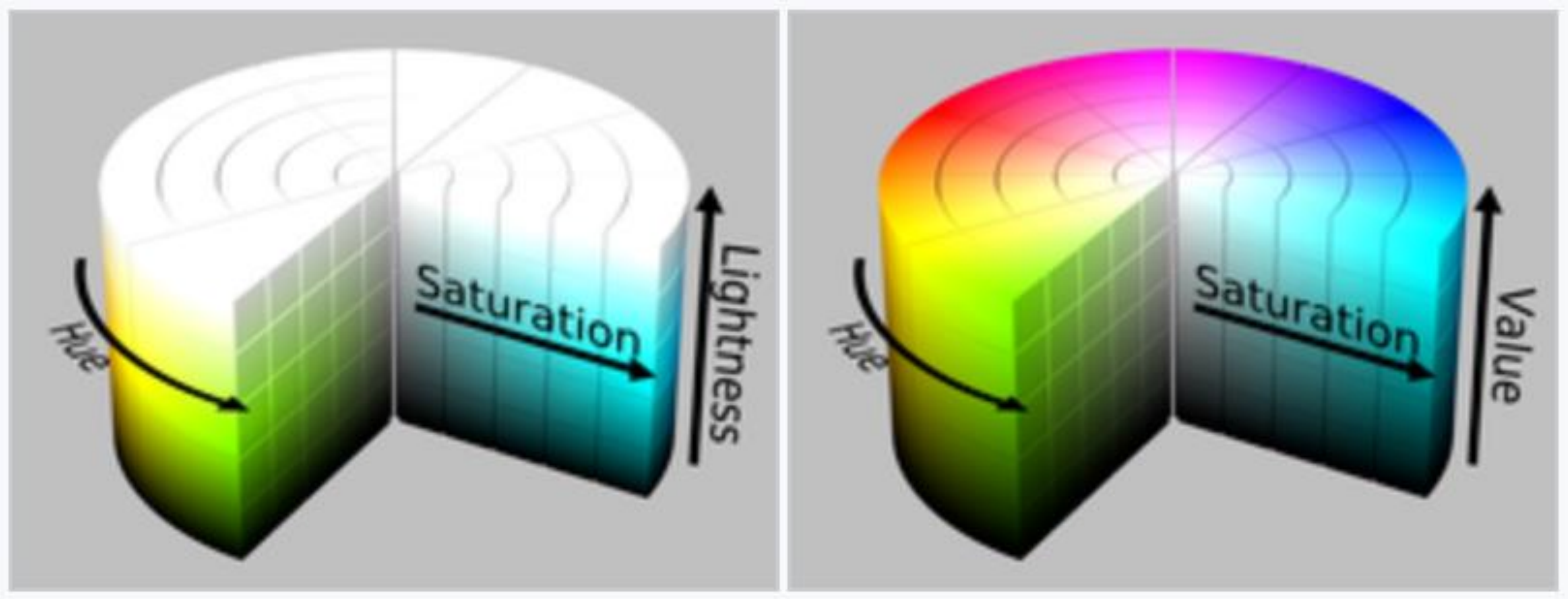
Magenta:

Yellow:

CMYK: Subtractive Colouring



Colour spaces: HSV/HSL



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

L*a*b

Expert: Christopher Brown

- RGB is replaced with L*a*b
- L*a*b is designed to approximate human vision.
  - Perceptually uniform with respect to human vision, unlike RGB.
- Used when:
  - Converting from RGB to CMKY, L*a*b is happening behind this.
  - Sharpening images, enhancing colour microvariations.

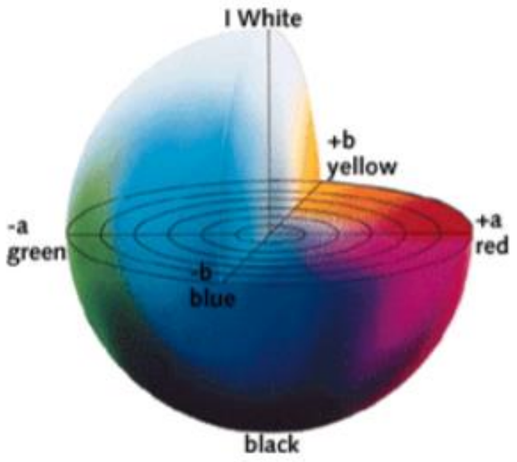


Fig: The CIELAB color space (from www.linocolor.com)

Fig: Before L*a*b (from https://digital-photography-school.com)

Fig: After L*a*b (from https://digital-photography-school.com)

Colour Space – Other
Expert: Lewis Winters

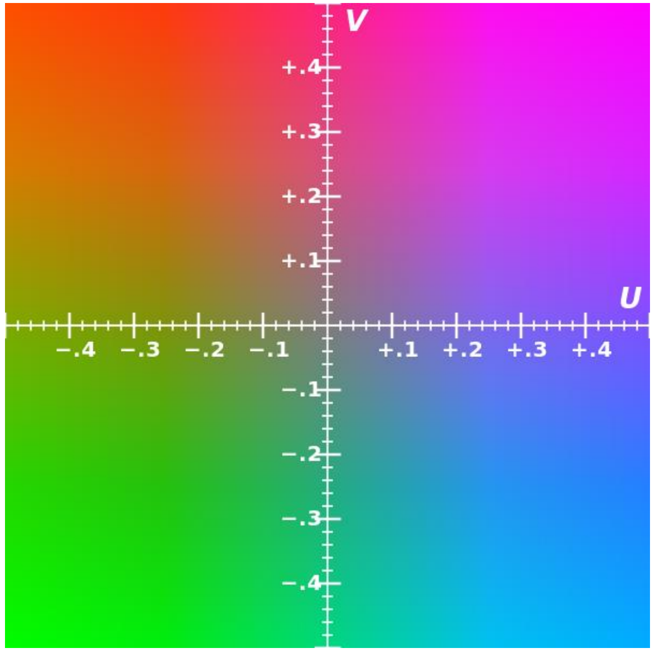
YIQ -

YIQ was a colour space formerly used on national television in North America and other countries. It works by storing a luma value with two chrominance values used to approximately describe the amounts of blue and red in the colour

YUV -

YUV is similar to YIQ but is rotated by 33°. It is typically created from RGB source. The values of R, G and B are weighted to produce Y', a measure of the luminance. The YUV model defines a colour space in terms of one luma(Y') and two chrominance(U and V). The primary advantage of luma/chroma systems such as YUV and YIQ is that they remain
compatible with black and white analogue television. The Y' channel saves all the data recorded by the black and white cameras, so it produces a signal suitable for reception on old monochrome displays. For black and white the U and V values can be discarded.

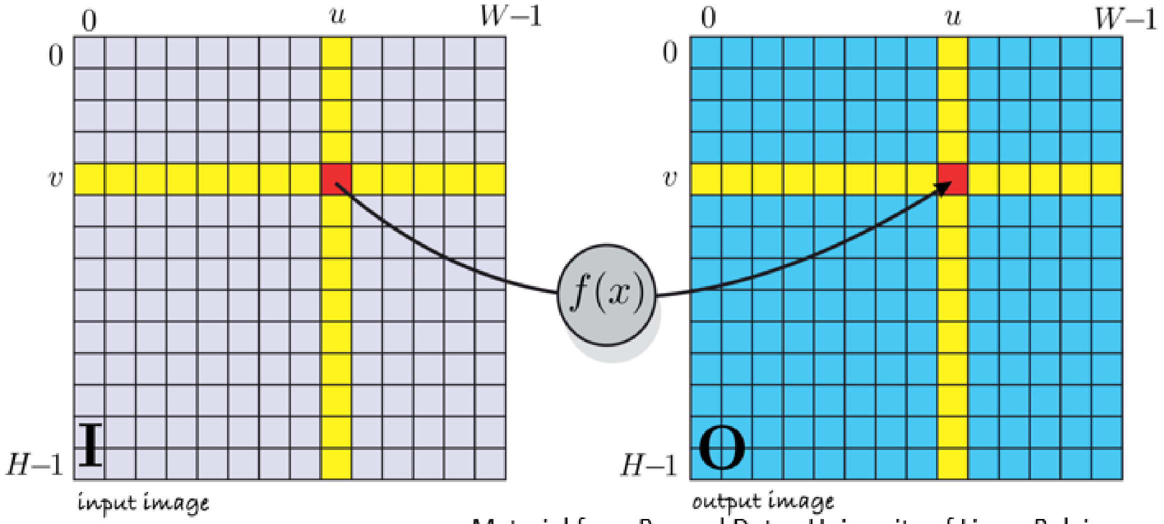
UV colour plane
With Y = 0.5,
Shown within
RGB color gamut



MONADIC (ONE PICTURE) OPERATIONS

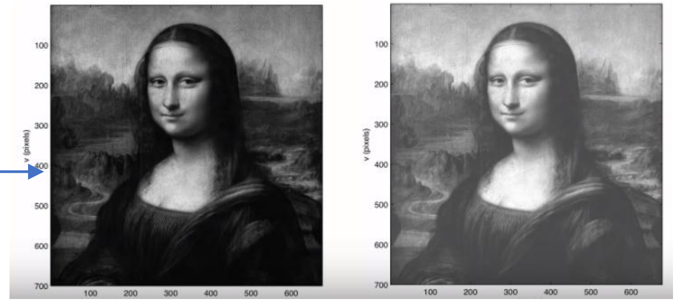
Experts: Sara Cooper and Flavien David

- Each output pixel is a function of corresponding input pixel



Materiel from Renaud Detry, University of Liege, Belgium

f(x) = x + 0.25



f(x) = 2x

valPix1= x1 & valPix2=x2

Different functions

- Brightness
- Contrast
- Thresholding

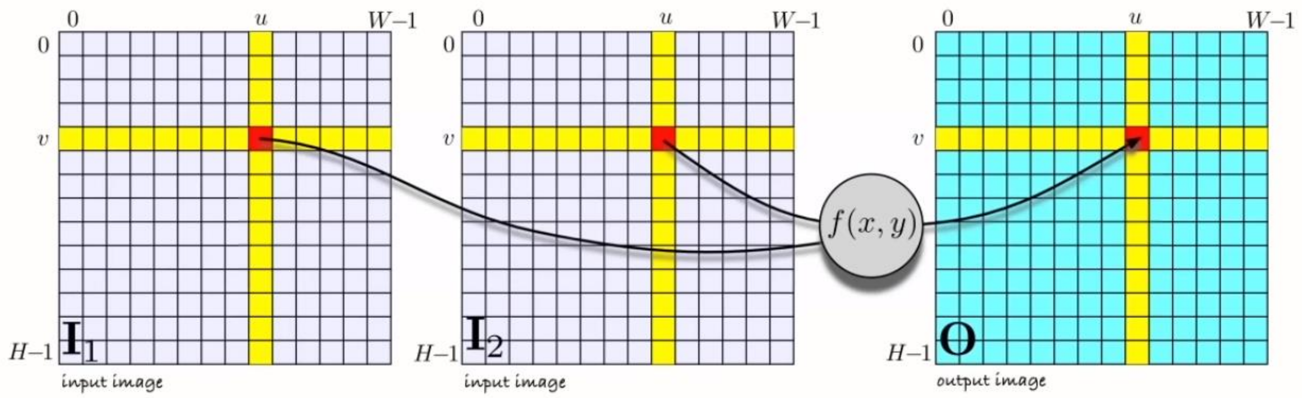
- Data conversion type:
- uint8 to double
- Binary to float

Etc …

Image processing: Dyadic operations

Experts: Clement DESBAN-ESTEVES and Teva DEMANGEOT

→ Mathematical operations on pixels involving two pictures.



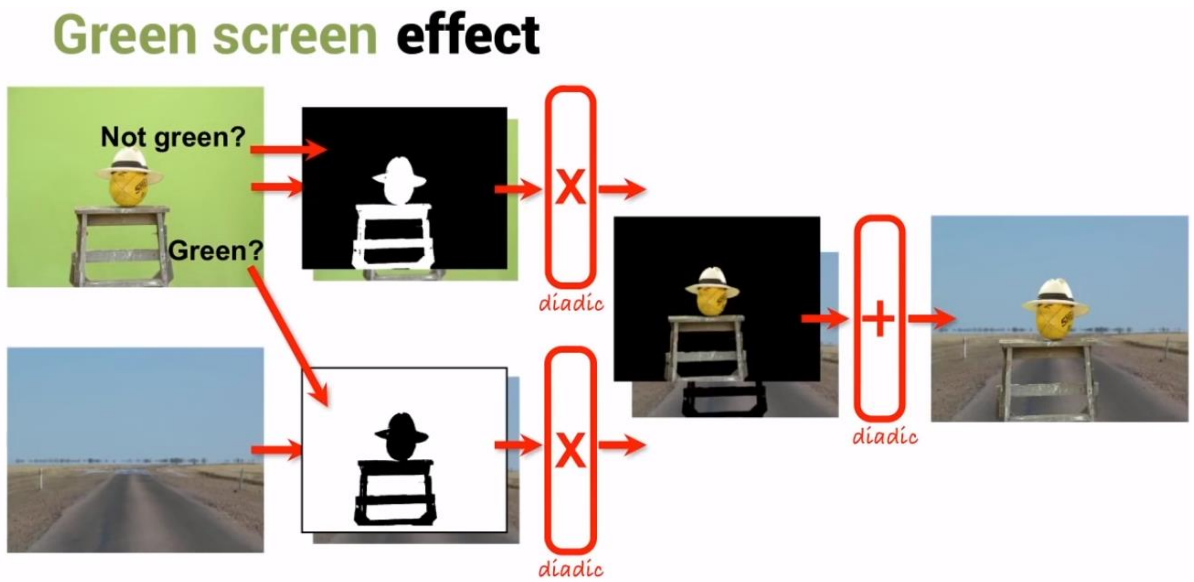
Robotics, vision and control: Fundamental algorithms in MATLAB 2011, p. 296
Corke, P. | Reproduced with permission from Springer Science, & Business Media

Kind of functions:

- Sum,
- Difference,
- Multiplication,
- Division,
- Logical operators (=, <=, >…)

Applications:

- Detection of variations between frames,
- Masking,
- Green screen effect…



Convolution

$$(f * g)(t) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) \, d\tau$$

Image processing: spatial operations

- Convolution

$$g(i,j) = \sum_{k,l} f(i-k, j-l)h(k,l;i,j),$$

- Convolution vs cross correlation?

Image processing: properties of convolution

- Commutativity

$$f * g = g * f$$

- Associativity

$$f * (g * h) = (f * g) * h$$

Image processing: properties of convolution

- Distributivity

$$f * (g + h) = (f * g) + (f * h)$$

- Associativity with scalar multiplication

$$a(f * g) = (af) * g$$

Convolution theorem

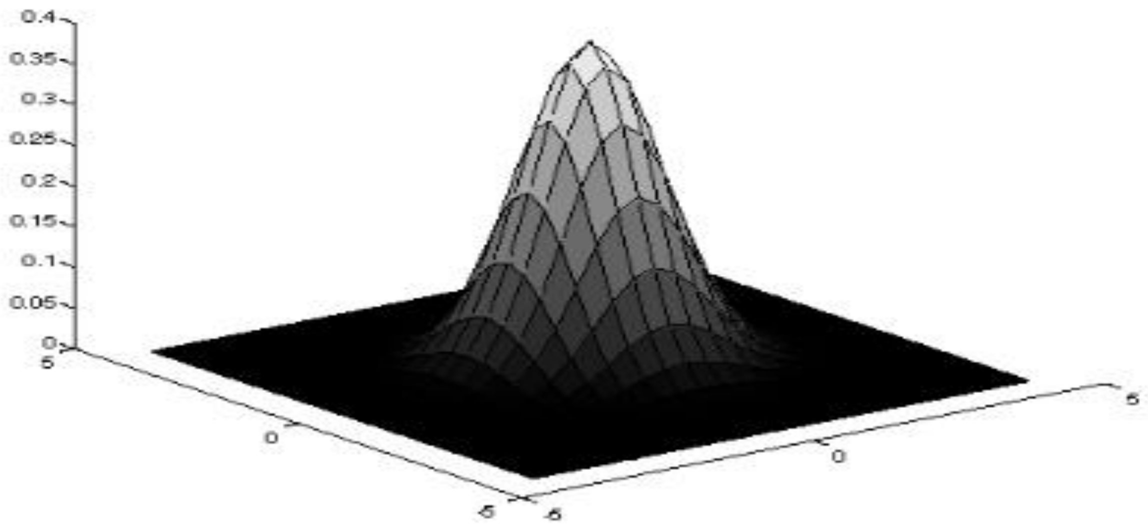
- In the frequency domain convolution becomes multiplication:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$

Image processing : Gaussian Filter

Expert: Sebastien Dilhuit

- Reduce the noise and to smooth the images
- $G\ (x, y) = \frac{1}{2 * \pi * \sigma^2} * \ e^{-\frac{x^2+y^2}{2*\sigma^2}}$
- σ : standard deviation of the distribution

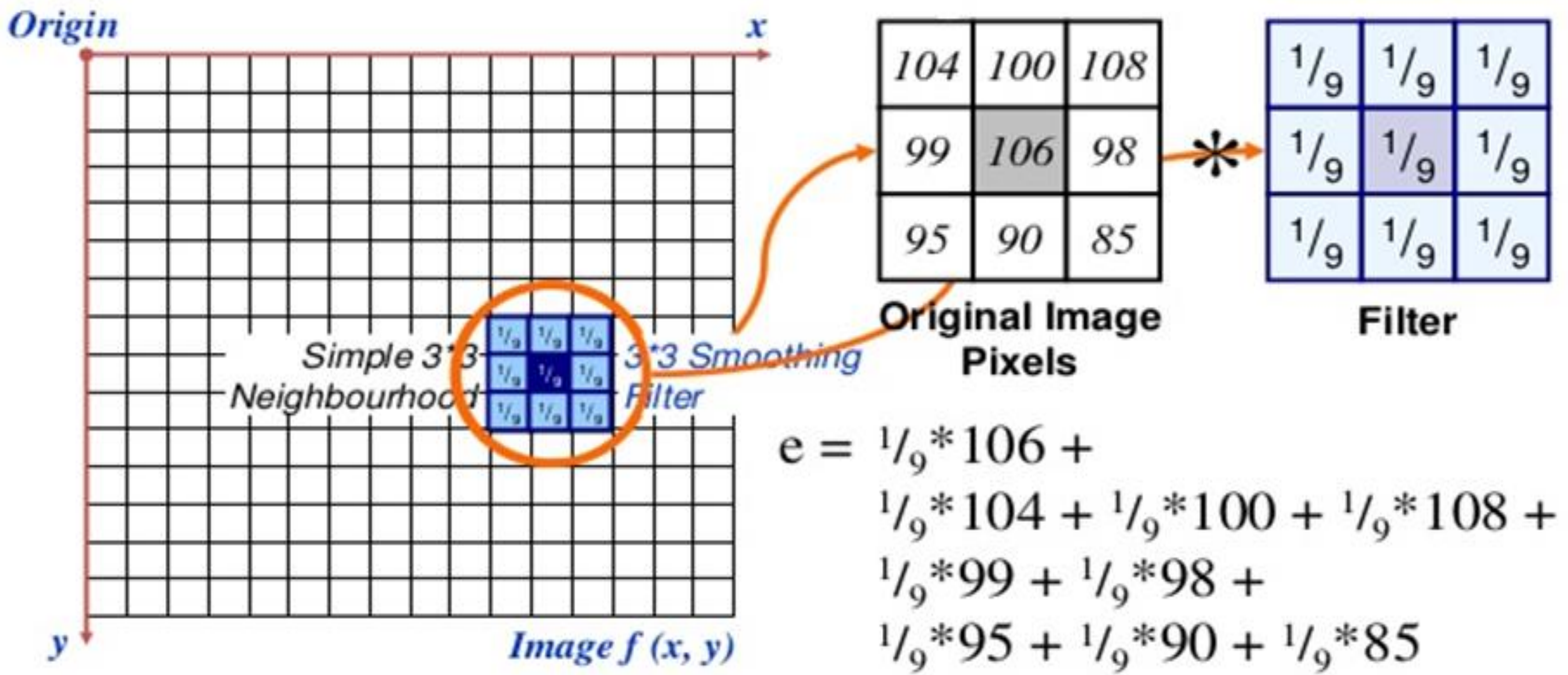


Filter (5,5) σ = 1

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Image processing : Gaussian Filter



$$
\begin{aligned}
e = {} & {}^{1}/_{9}*106 + \\
& {}^{1}/_{9}*104 + {}^{1}/_{9}*100 + {}^{1}/_{9}*108 + \\
& {}^{1}/_{9}*99 + {}^{1}/_{9}*98 + \\
& {}^{1}/_{9}*95 + {}^{1}/_{9}*90 + {}^{1}/_{9}*85 \\
= {} & 98.3333
\end{aligned}
$$

Image processing : Gaussian Filter

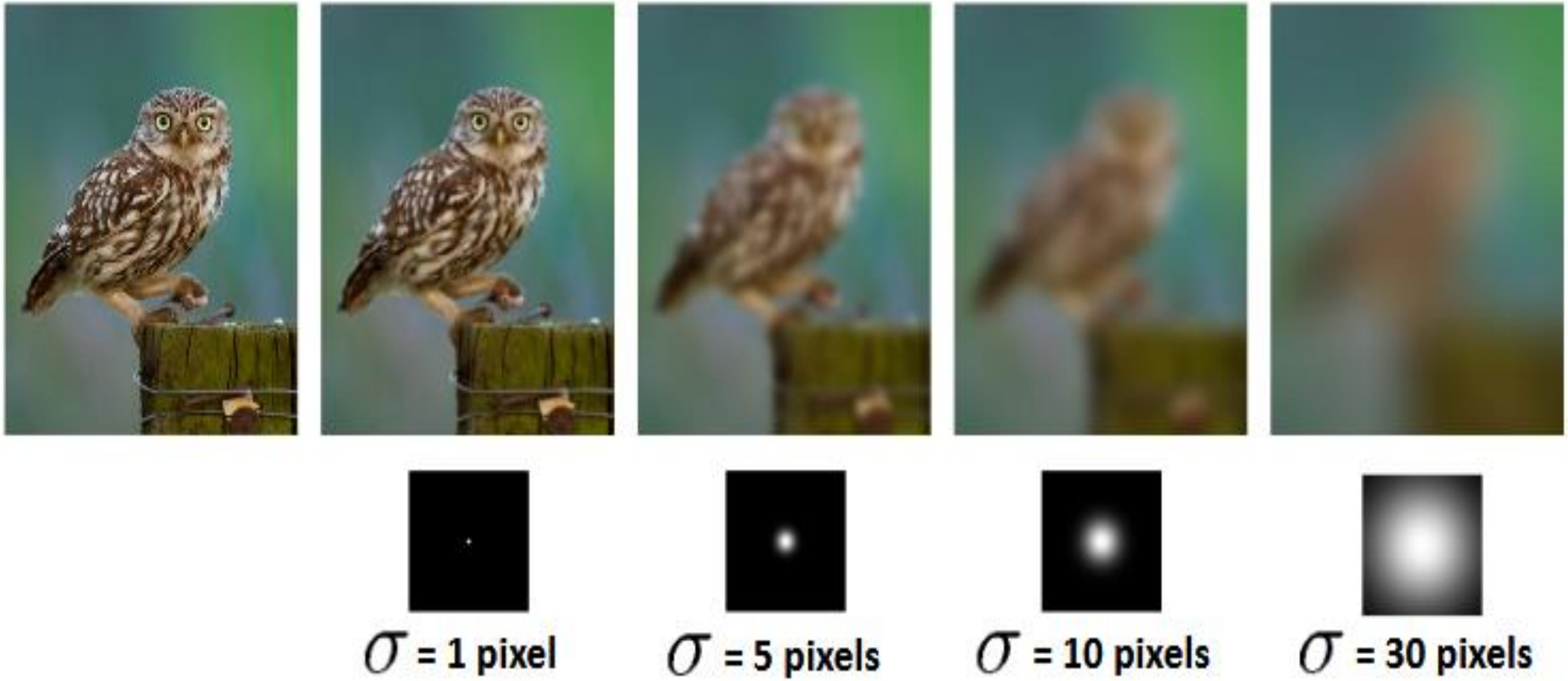


Image Processing: Laplacian of Gaussian(LoG) Filter

Experts: Lucas HERAUD and Sean KATAGIRI

Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by: $L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ which can be calculated using a **convolution filter**.

Input image is a **set of discrete pixels**, we need a discrete convolution kernel to approximate the second derivatives in L(x,y), such as these commonly used kernels:

0	−1	0
−1	4	−1
0	−1	0

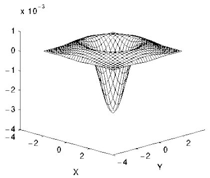
−1	−1	−1
−1	8	−1
−1	−1	−1

Because these kernels are approximations, they are **very sensitive to noise**. Thus the image is **smoothed with a Gaussian filter** before applying the Laplacian filter.

Convolution is associative, thus we can **convolve the Gaussian smoothing filter with the Laplacian filter** first, and then **convolve the hybrid filter** with the image.

- Requires **far fewer arithmetic operations**
- The LoG kernel can be **precalculated in advance** so only one convolution performed at run-time.

The 2-D LoG function centred on zero and with Gaussian standard deviation σ has the form:

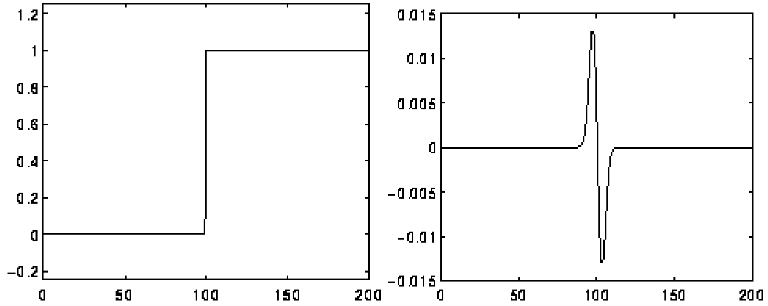


$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

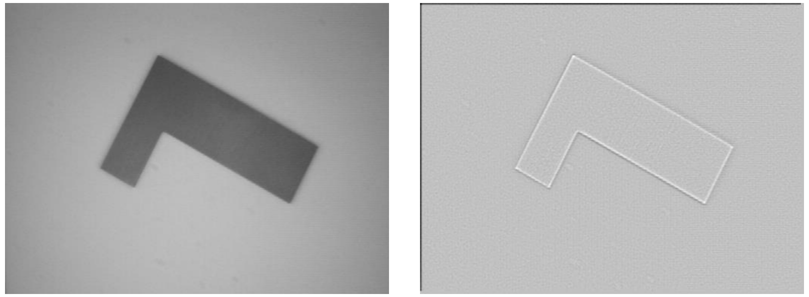
0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Usage of LoG:

- Areas with **constant intensity**, the **LoG response** will be **zero**.
- In the vicinity of a change in intensity, the LoG response will be positive on the darker side and negative on the lighter side.



- Example of the effect of LoG filter on an image.



Proofs if required for presentation

$$\triangle[G_\sigma(x,y) * f(x,y)] = \boxed{[\triangle G_\sigma(x,y)]} * f(x,y) = \boxed{LoG} * f(x,y)$$

Laplacian filter Gaussian filter

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\frac{\partial}{\partial x} G_\sigma(x,y) = \frac{\partial}{\partial x} e^{-(x^2+y^2)/2\sigma^2} = -\frac{x}{\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

($\frac{1}{\sqrt{2\pi\sigma^2}}$ is normalised for simplicity)

$$\frac{\partial^2}{\partial^2 x} G_\sigma(x,y) = \frac{x^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2} - \frac{1}{\sigma^2} e^{-(x^2+y^2)/2\sigma^2} = \frac{x^2-\sigma^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2}$$

Similarly, $\frac{\partial^2}{\partial^2 y} G_\sigma(x,y) = \frac{y^2-\sigma^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2}$

$$LoG \triangleq \triangle G_\sigma(x,y) = \frac{\partial^2}{\partial x^2} G_\sigma(x,y) + \frac{\partial^2}{\partial y^2} G_\sigma(x,y) = \underline{\frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2}}$$

Lucas HERAUD
Sean KATAGIRI

Image Processing: Sobel Operator

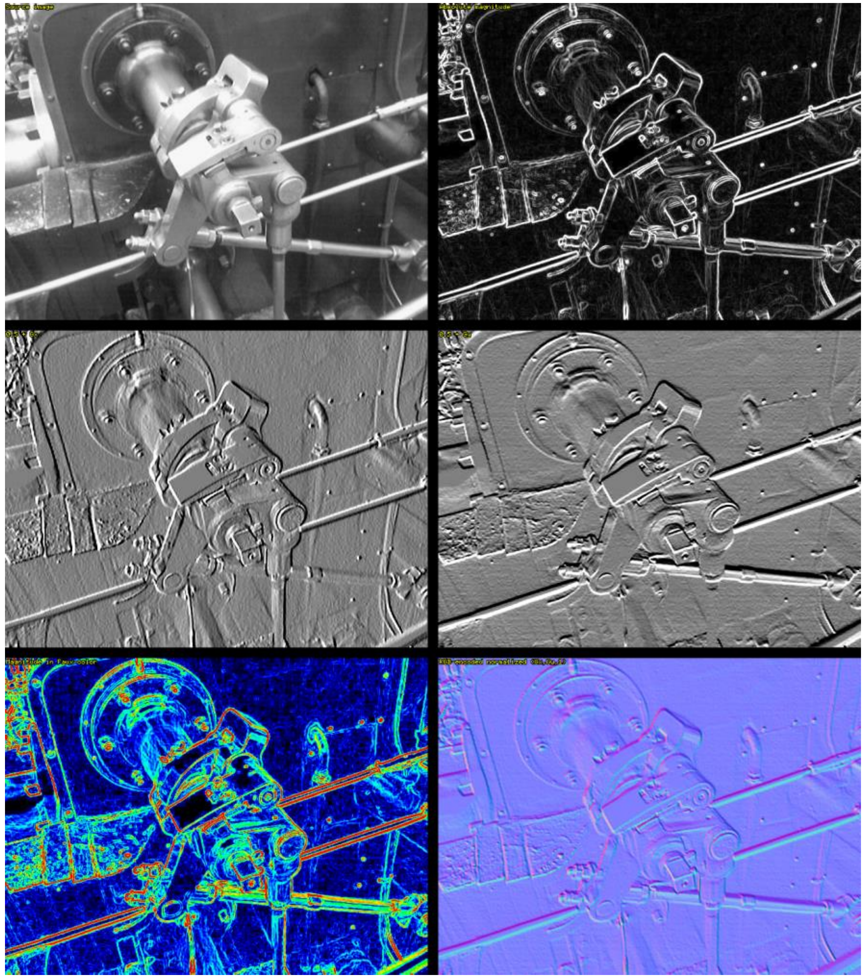
Expert: Marek Kujawa

- Used for edge detection in an image,
- It is a simple way to approximate the gradient of the intensity in an image,
- Uses Kernel Convolution for Its calculations:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A$$

where A is the source image matrix and * is the convolution operator,

$$G = \sqrt{G_x^2 + G_y^2}$$



Mathematical Morphology: “Dilation”

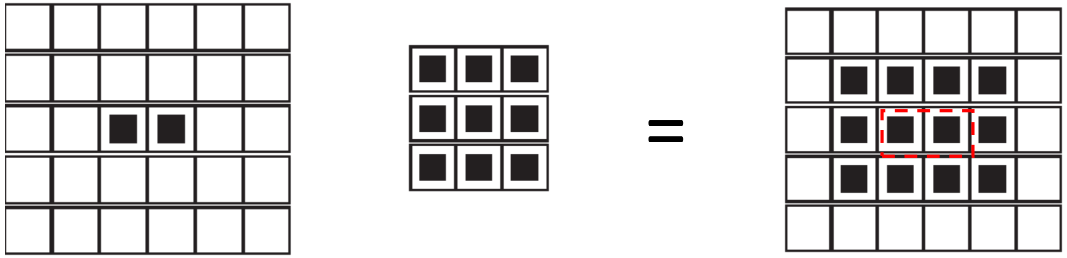
Experts: Pierre Le Hen, Matthew Osborne

Principle : The Dilation function δ makes an object *X* larger by multiplying a template matrix S by all squares belonging to X

Mathematical Notation:

$\delta(X) = \{x+s | x \in X \wedge s \in S\}$: the set of all possible additions of an element of X and an element of S

Example : The dilation of X by S is given by this 6 x 6 matrix.



X =(2,2),(3,2)

S =(-1,1), (0,1), (1,1)
(-1,0), (0,0), (1,0)
(-1,-1),(0,-1),(1,-1)

δ(*X*)

(2,2)+ S = (1,3),(2,3),(3,3),(1,2)(2,2)(3,2),(1,1)(2,1)(3,1)
(3,2)+ S = (2,3),(3,3)(4,3),(2,0)(3,2)(4,2),(2,1)(3,1)(4,1)

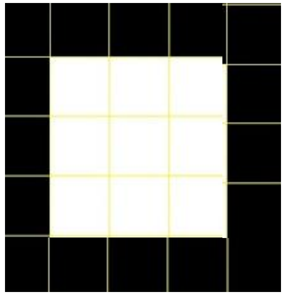
Note: S has Origin (0,0) and adds 1 pixel to all adjacent pixels for all *x* in *X*.

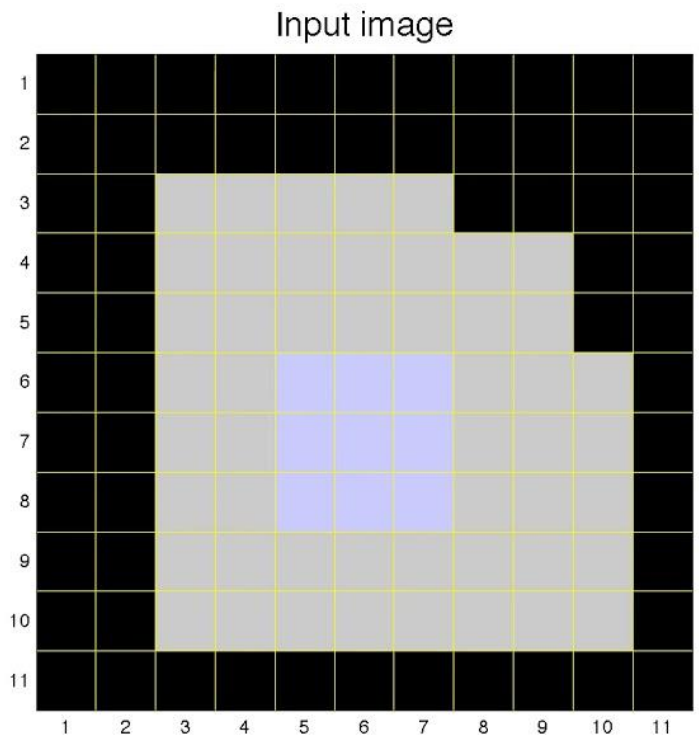
*N.B. If S is not symmetrical S^T must be used.

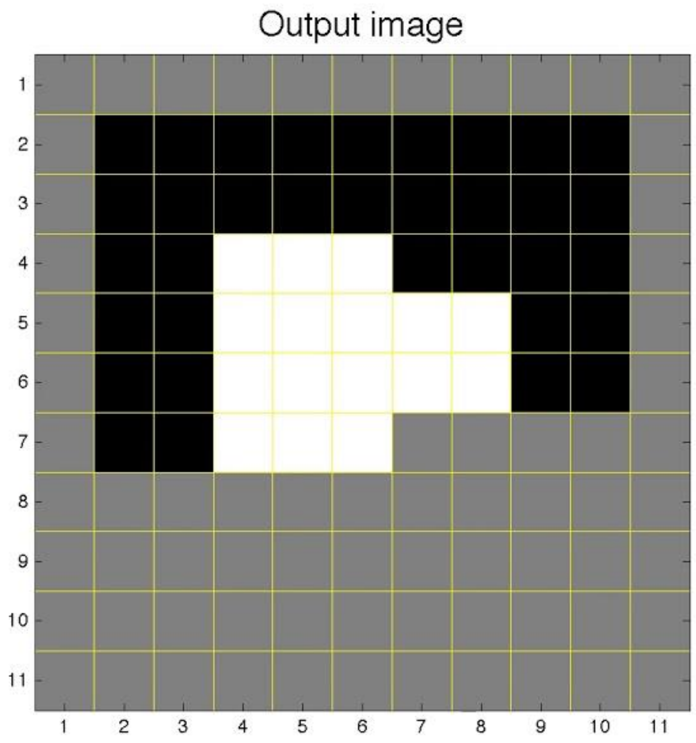
Mathematical Morphology Erosion

Expert: Scott Mathers

Structuring Element







https://i.ytimg.com/vi/b5lgnNEzGeU/maxresdefault.jpg

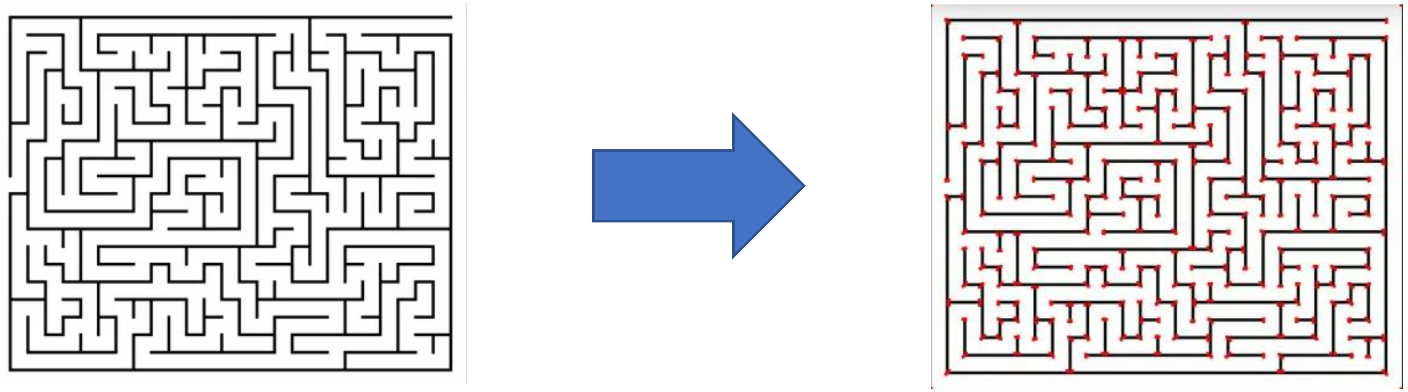
Point feature methods

- Used to describe a point using its surrounding
- Features differ in computational cost and robustness
- Usually the faster it may be computed the less robust it is: can only be used to distinguish points locally
- Feature detector vs descriptors

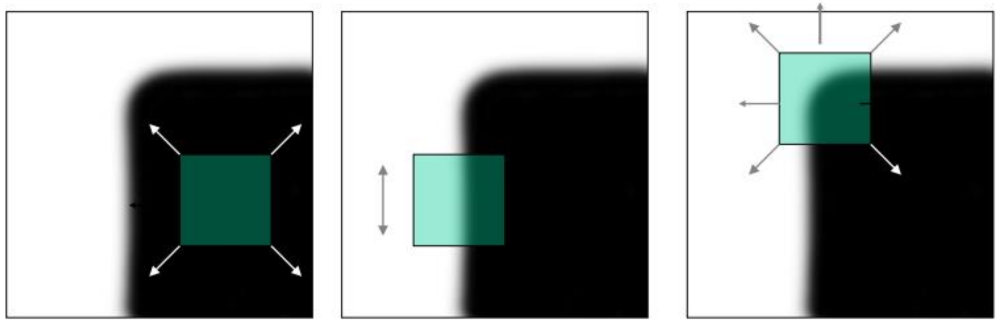
Harris Corner Detector

Expert: Linda MBONGUE NGUISSI

Harris Corner Detector : Mathematical way to determine regions where there are corners in an image.



How does it work ?



"flat" region:
no change in all directions

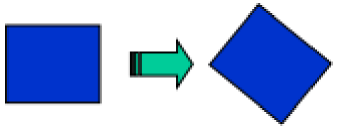
"edge": no change along the edge direction

"corner": significant change in all directions

Shifting a window in any direction should give a *large change* in *intensity*.

Properties

- Invariant to rotation
- Robust
- Not very fast



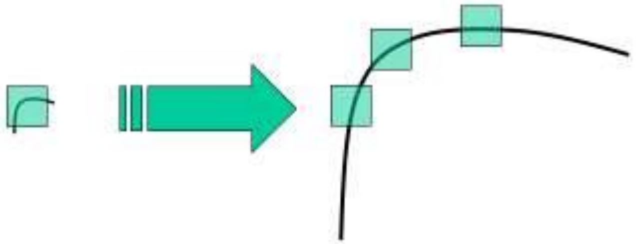
Applications

Object recognition, Motion detection, Video tracking, etc.

Point feature methods: SIFT

Experts: Srap Melkonyan, Joshua Roe

SIFT is Scale-invariant feature transform. In comparison with Harris corner Detector SIFT detects corners even if the image is scaled. A Harris example is seen below:

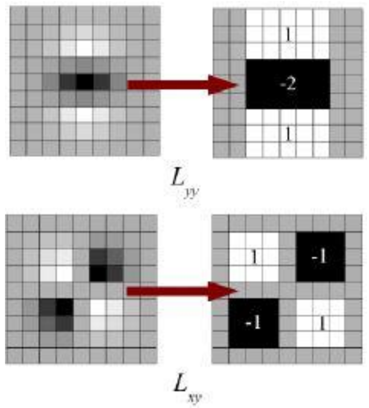


What Happens here is that when the corner is too small, it cannot be detected by Harris. Only when the scale is changed can it be detected by the Harris.
The SIFT algorithm solves this problem in a few steps.

1. **Scale-space keypoint detection**
   It uses x,y,σ (σ is scale) to detect key points that potentially have corners, σ is used to differentiate small/large corners.
2. **Keypoint Localization**
   When we have all the keypoints from the previous step, Taylor series expansion is used to reject the lesser corners.
3. **Orientation Assignment**
   In order to be rotationally invariant, a neighborhood is taken around the keypoint with 1.5 times the σ (scale) to calculate the gradient magnitude and direction. After a process of elimination it creates keypoints identical in location and scale but with different directions.
4. **Keypoint Descriptor**
   A histogram is created from the neighborhood and sub-blocks around the keypoint. A vector is used to form the keypoint descriptor, plus some refining to achieve robustness to changes/rotation etc.
5. **Keypoint Matching**
   The algorithm tries to match keypoints between 2 images by comparing neighborhoods, but there may be two really close matches (due to noise etc.). 90% of false matches are successfully found and rejected, and only 5% of correct matches are wrongly discarded.

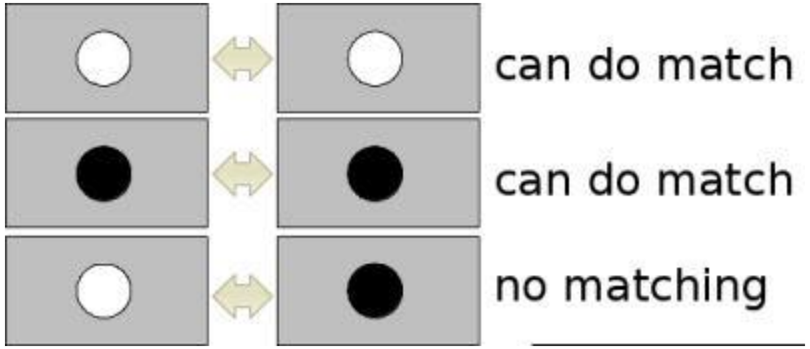
Speeded up Robust Features (SURF)

Expert: Jack Rome and Loris Montbarbon



- Acts as an approximation of SIFT
- Local feature detector and descriptor
- Used for object recognition, image registration, classification or 3d reconstruction
- 3 Steps: Detection, Descriptor, Matching
- Hessian Matrix:

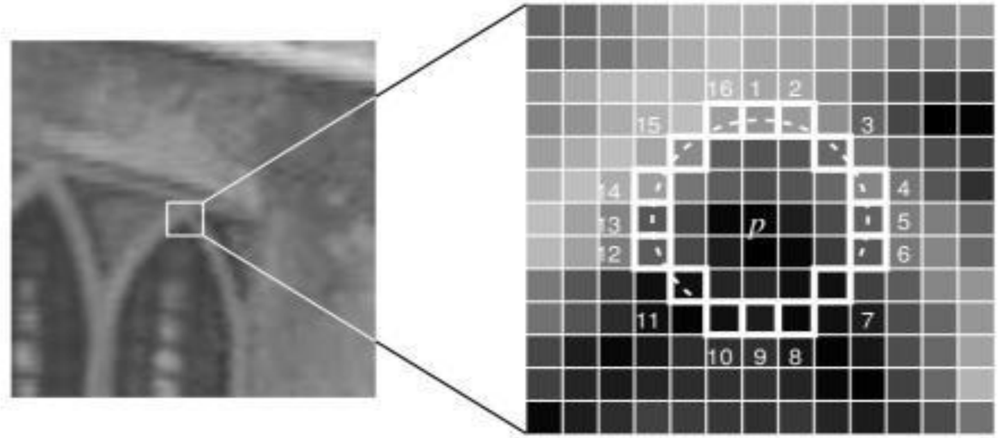
$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix}$$



FAST Point Detector

Experts: Shayne Shaw, Eduardo Ochoa Melendez

Introduced as an option for computing features in real-time applications.



- A pixel **p** is selected with intensity **Ip**.
- A threshold value **T** is set (assumed to be around 20% of the pixel under test)
- A circle of 16 pixels surrounding **p** is considered.
- If **N** adjacent pixels on this circle are either above or beneath **Ip+/-T**, then p is an interest point.
- First test pixels 1,5,9 and 13 of the circle.
- If at least three of them are above **Ip+T** or beneath **Ip-T**, then for each of the 16 pixels it is checked that **N** contiguous pixels fall under the criterion.
- Repeat the process for all pixels in the image.

Weaknesses

- Multiple adjacent features may be detected
- It does not reject as many candidates for n < 12
- Pixel choice is not optimal.
- Does not function well with high noise levels

Advantages

- It is several times faster than other corner detectors

Experts: Mohit Vaishnav and Gideon Oladipupo

ORIENTED FAST AND ROTATED BRIEF (ORB)

- ORB essentially fused the features of both FAST and BRIEF for optimum image recognition .
- One of the setbacks in FAST method of features detection is that it does not compute the orientation of detected points and it is unstable rotationally. Also, BRIEF poorly performs if there is an in-plane rotation.
- ORB facilitates the orientation of keypoints in the image thereby dealing with problems associated with rotational invariance.
- In ORB, a rotation matrix is computed using the orientation of patch and then the BRIEF descriptors are steered according to the orientation

References:

1. Ebrahim Karami, Siva Prasad, and Mohamed Shehata Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images accessed at *https://arxiv.org/abs/1710.02726*.
2. Ethan RubleeVincent RabaudKurt KonoligeGary BradskiORB, an efficient alternative to SIFT or SURF accessed at http://www.willowgarage.com/sites/default/files/orb_final.pdf.
3. Reinhard Klette, Keypoints and Descriptors accessed at https://www.cs.auckland.ac.nz/~rklette/CCV-Dalian/pdfs/E02_Features.pdf

Multiview and stereo 3D reconstruction

Dr Tomasz Luczynski

Epipolar geometry

- e_L and e_R are called **epipoles**
- Line O_L–X projected to the right image is called **epipolar line**
- Plane X, O_L & O_R is called **epipolar plane**
- **Epipolar constraint:** if the relative orientation between the cameras is known, the search for the correspondences is limited to the epipolar line

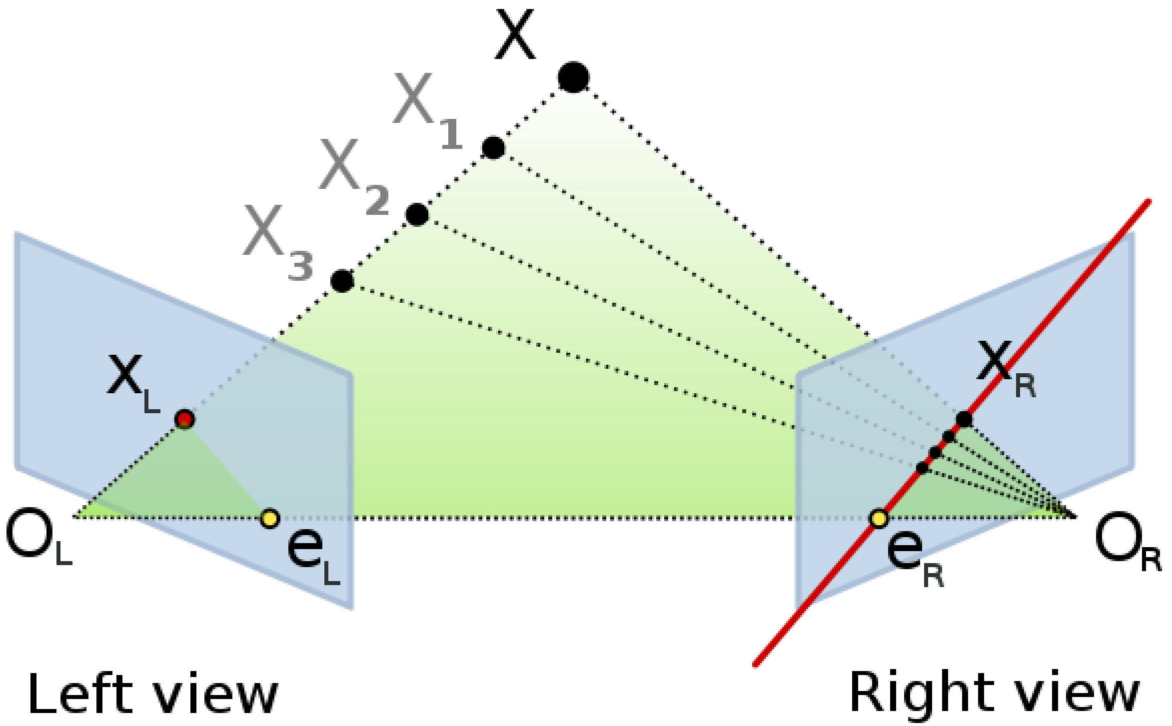


image from https://en.wikipedia.org/wiki/Epipolar_geometry

Essential and fundamental matrices

- **Szeliski R. Computer Vision: Algorithms and Applications, Chapter 7.2**
- http://www.cse.psu.edu/~rtc12/CSE486/lecture19.pdf
- http://robotics.stanford.edu/~birch/projective/node20.html
- The essential and fundamental matrices are 3x3 matrices that “encode” the epipolar geometry of two views.
- Given a point in one image, multiplying by the essential/fundamental matrix will tell us which epipolar line to search along in the second view.
- **The main difference is that the essential matrix reveals the relation in global coordinates, while the fundamental matrix uses each camera's intrinsics to relate them in pixel coordinates.**

Essential matrix

- Essential matrix deals with calibrated cameras and fundamental matrix with uncalibrated cameras
- The Essential matrix contains five parameters (three for rotation and two for the direction of translation - the magnitude of translation cannot be recovered due to the depth/speed ambiguity) and has two constraints: (1) its determinant is zero, and (2) its two non-zero singular values are equal.

$$\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times$$

Fundamental matrix

- Unlike essential, fundamental matrix deals with uncalibrated systems:

$$\mathbf{E} = \mathbf{K}'^{\top}\ \mathbf{F}\ \mathbf{K}$$

- The fundamental matrix can be determined by a set of point correspondences. For all pairs of corresponding points holds:

$$\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0$$

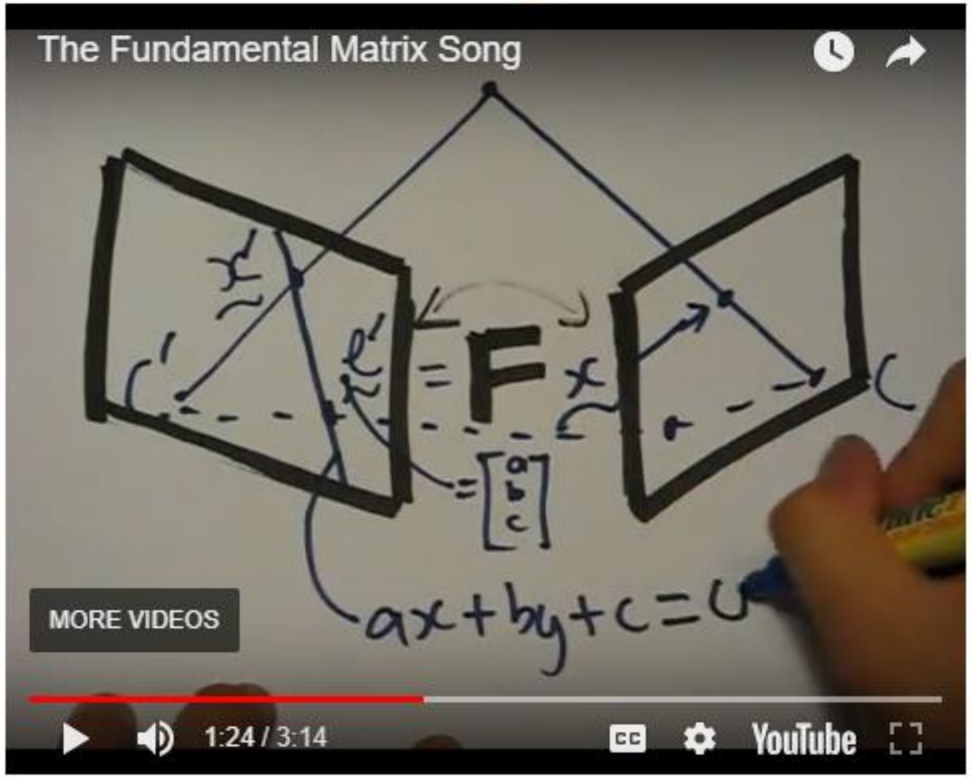
- The Fundamental matrix contains seven parameters and its rank is always two.

Feature matching

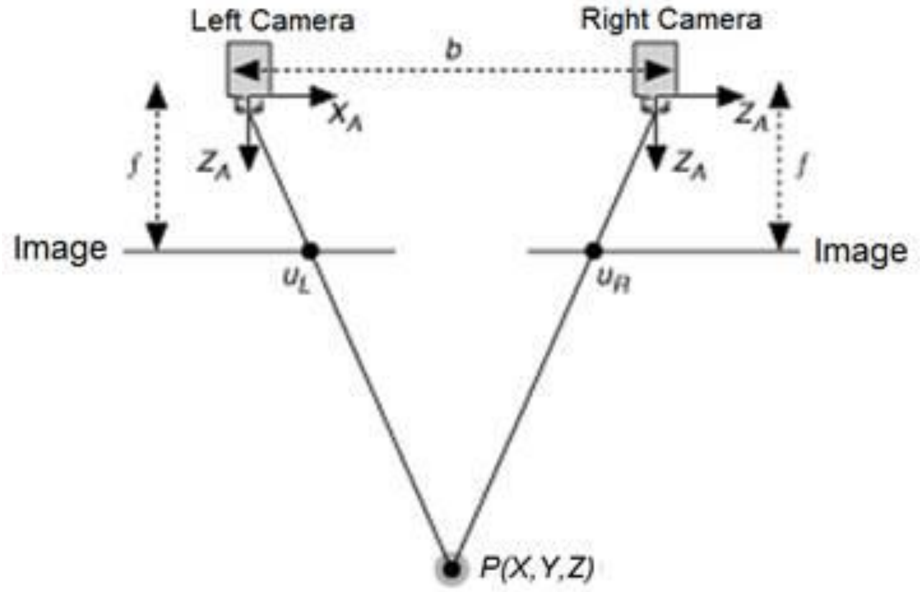
http://www.morethantechnical.com/2012/02/07/structure-from-motion-and-3d-reconstruction-on-the-easy-in-opencv-2-3-w-code/

Fundamental matrix song ;)

- For the brave ones: https://youtu.be/DgGV3l82NTk



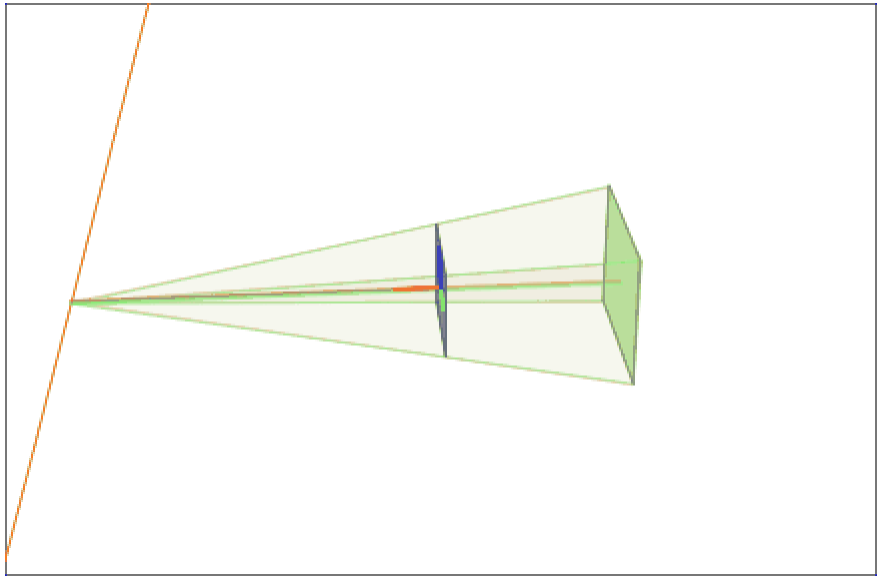
Stereo setup



http://www.ni.com/white-paper/14103/en/

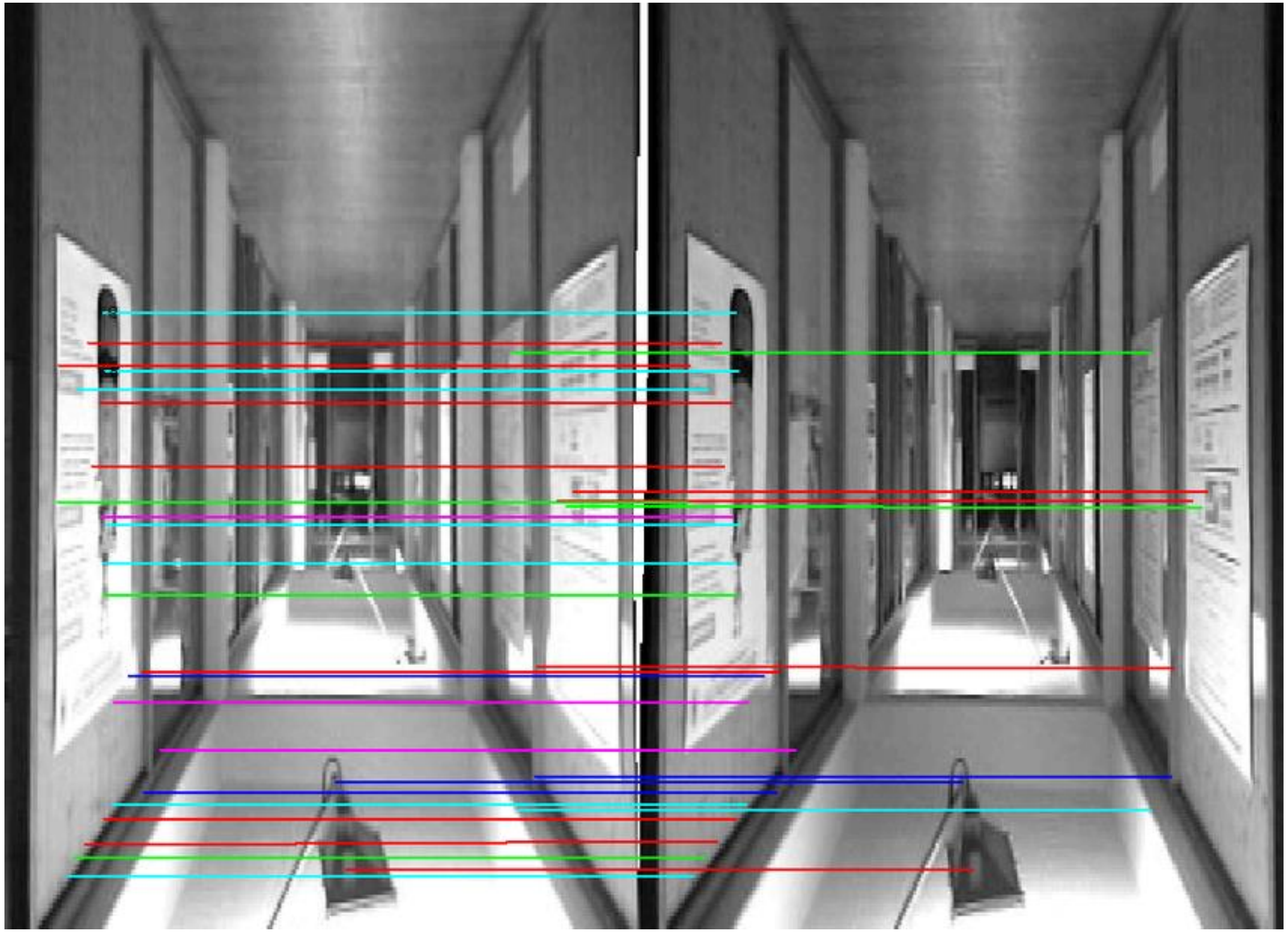
Image rectification

- In the rectification process left and right images are projected onto a common plane
- Rectification simplifies search for correspondences between the images



Gif from https://en.wikipedia.org/wiki/Image_rectification

Image rectification



Arturo Gil et al. Improving Data Association in Vision-based SLAM

Dense reconstruction

- Sum of absolute differences window algorithm

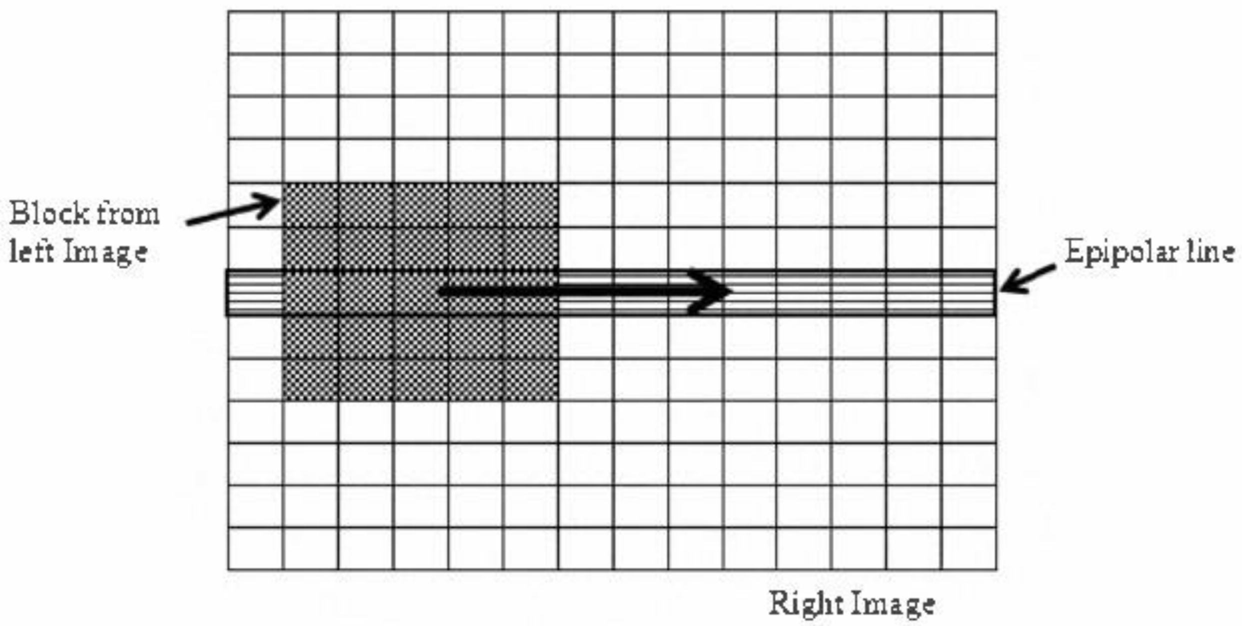
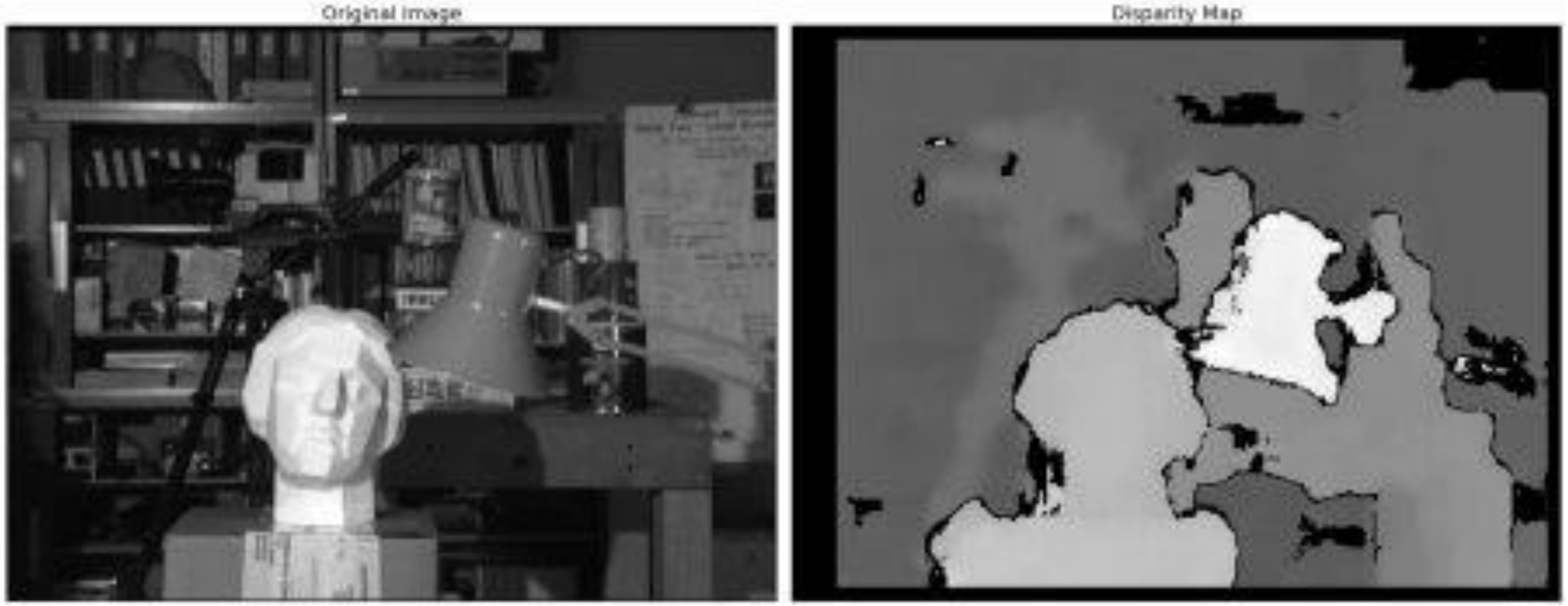


Figure 6. SAD block matching process

https://www.semanticscholar.org/paper/Sum-of-Absolute-Differences-algorithm-in-stereo-for-Hamzah-Rahim/fdf9c44158f99cae3e708784aa5dd9c17032ed6c

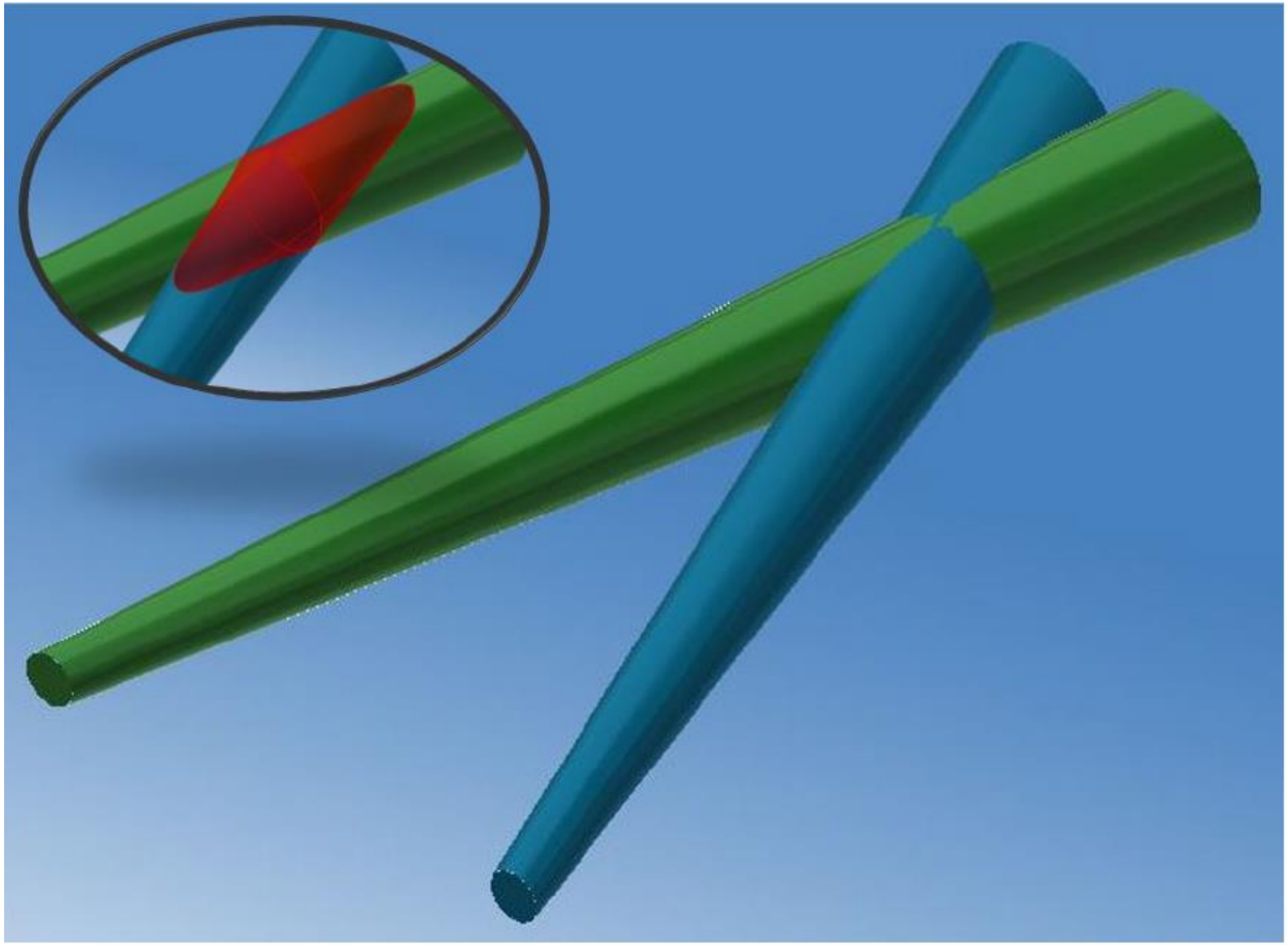
Disparity image

- Disparity - the distance between two corresponding points in the left and right image of a stereo pair.
- Disparity image (map) can be directly translated to 3D points



https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_calib3d/py_depthmap/py_depthmap.html

Quantization error



3D video encoding

- Anaglyph 3D
- Pixel subsampling (side-by-side, checkerboard)
- Enhanced video stream coding (2D+Delta, 2D+Metadata, 2D plus depth)

3D video encoding

https://trac.ffmpeg.org/wiki/Stereoscopic