Hochschule München
University of Applied Sciences

FK04 - Fakultät für
Elektrotechnik und
Informationstechnik

Image Processing

1. Introduction

Alfred Schöttl

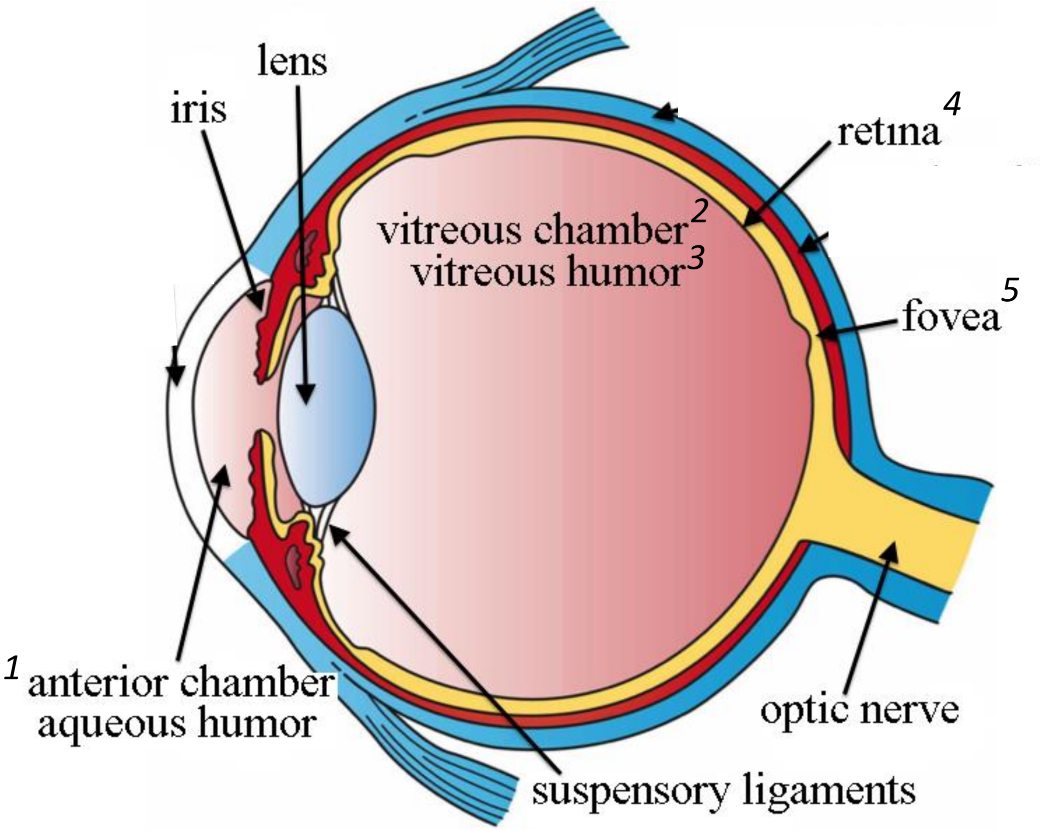
Organizatorial Matters

- ***Class:*** Mon, 10:00, R2.001, practice in E301 (dates in Moodle)
- If you agree: documentation in English, lecture in German. Exam is possible in German or English.
- We have a multi-modal approach to get into touch with image processing: The slides, an live lecture, a (much more comprehensive) script, practical parts with exercises and tutorials.
- This presentation (as a starter) and the accompanying notes can be found in Moodle.
- ***Bonus:*** 10 % Bonus possible (additional group work) -- stay tuned, see Moodle for details --
- ***Exam***: -- stay tuned, see Moodle for details --
- ***Bachelor thesis and further offers***: You're invited to do an internal thesis in the lab, voluntary work etc. there are a lot of interesting topics waiting for you! See Moodle and ask me.
- ***Don't be afraid*** of programming or the maths. Almost no programming knowledge is needed in the final exam. The course is a chance to improve some programming skills and your maths. It is a good idea to take complicated things as pure recipes in the beginning.
- Tools: Python/OpenCV + Jupyter Notebook. Optional: Gimp. See Moodle for details.
- Questions? alfred.schoettl@hm.edu, better: during and after each live session in Zoom.

Bei gesundheitlichen oder sprachlichen Einschränkungen bitte baldmöglichst zu mir kommen! Please get into contact with me as soon as possible in case of linguistic problems, don‘t be shy!

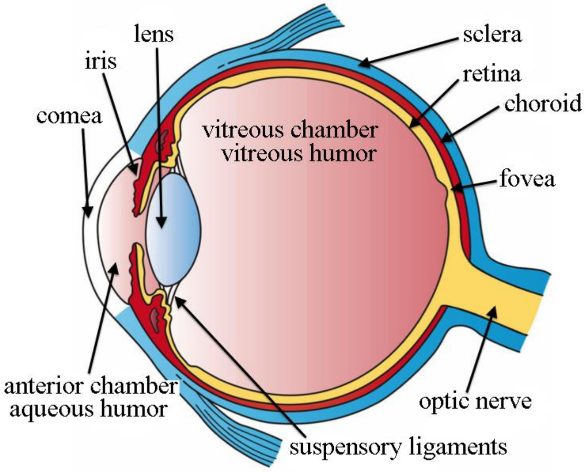
Visual Perception, the human eye

The human eye - anatomy

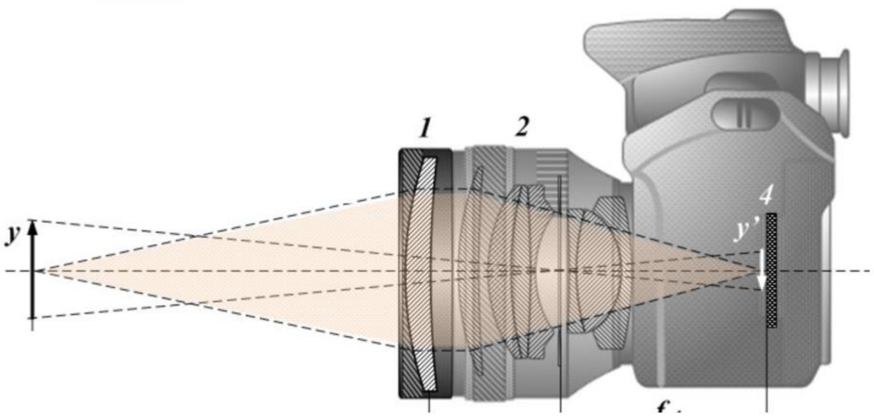


[1]vordere Augenkammer
[2]Hauptkammer
[3]Kammerwasser
[4]Netzhaut
[5]Sehgrube

The human eye



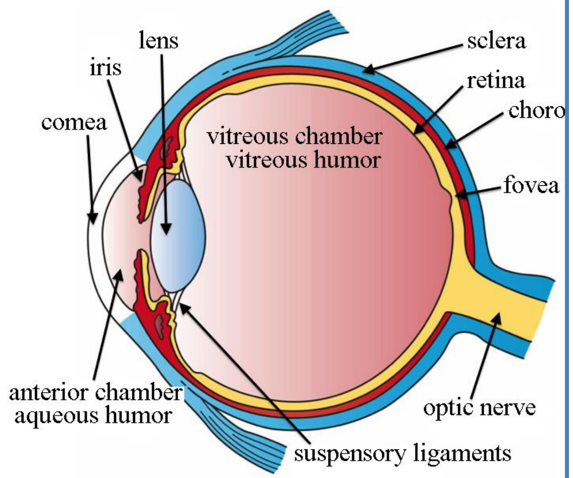
iris:	opening 2-8 mm
lens:	can be stretched by ***ciliary muscles***[1] which pull the suspensory ligaments, contracted muscles: thick lens (appropriate for near vision), relaxed muscles: thin lens (appropriate for far (>3 m) vision)
fovea centralis:	circular area with about 1.5 mm diameter, contains most sensor cells (see next slides), allows for sharp vision (only there!), is part of the ***macula lutea***[2]

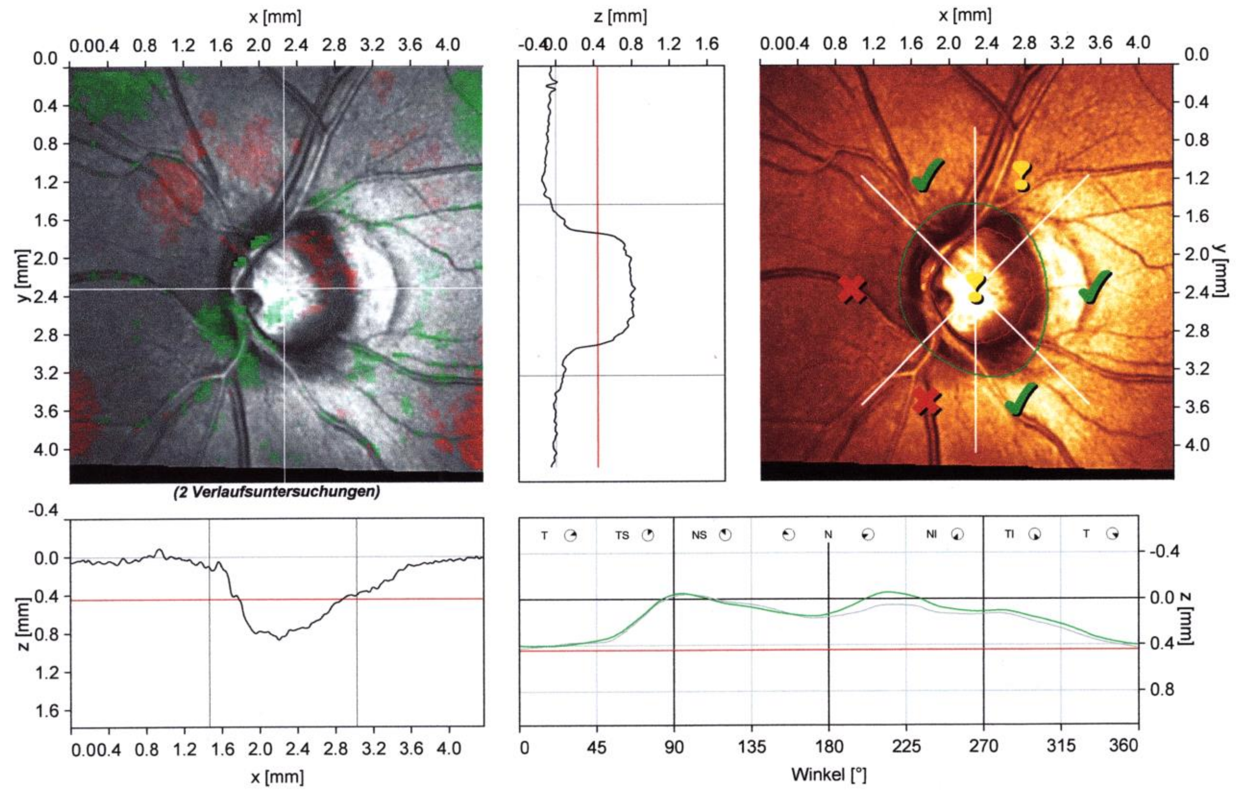


https://commons.wikimedia.org/wiki/File:RevLensMac.png

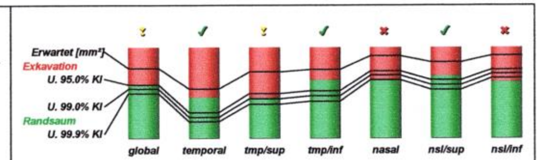
[1]Ziliarmuskeln
[2]gelber Fleck

The human eye – the optic nerve





Stereometrische Analyse		Änderung	Normalbereich
Papillenfläche	2.28	0.00 mm²	1.63 - 2.43
Exkavationsfläche	0.96	0.02 mm²	0.11 - 0.68
Randsaumfläche	**1.31**	**-0.03 mm²**	**1.31 - 1.96**
Exkavationsvolumen	0.23	0.02 mm³	-0.01 - 0.18
Randsaumvolumen	**0.37**	**-0.01 mm³**	**0.30 - 0.61**
Flächenquotient (C/D Ratio)	0.42	0.01	0.07 - 0.30
Lineare C/D Ratio	0.65	0.01	0.27 - 0.55
Mittlere Exkavationstiefe	0.34	-0.02 mm	0.10 - 0.27



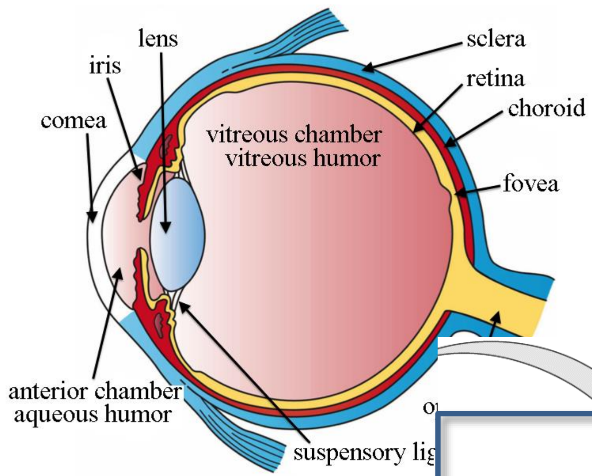
Moorfields Klassifikation: Außerhalb normaler Grenzen (*)

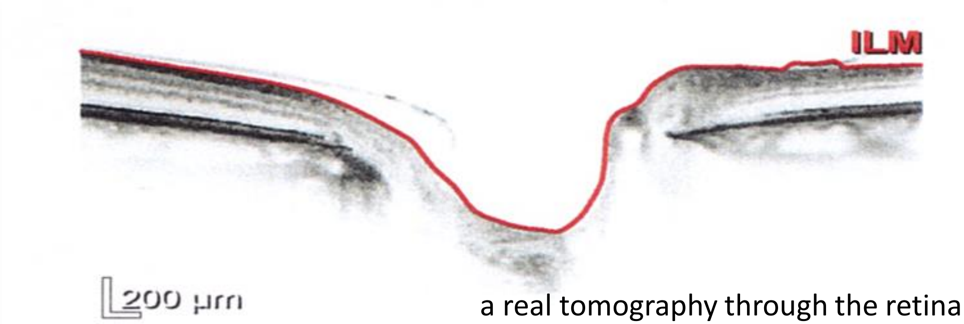
(*) Moorfields Klassifikation (Ophthalmology 1998;105:1557-1563). Die Klassifikation beruht auf Statistik. Die Diagnose liegt in der Verantwortung des Arztes.

the optic nerve

The analysis is a nice example of image processing!

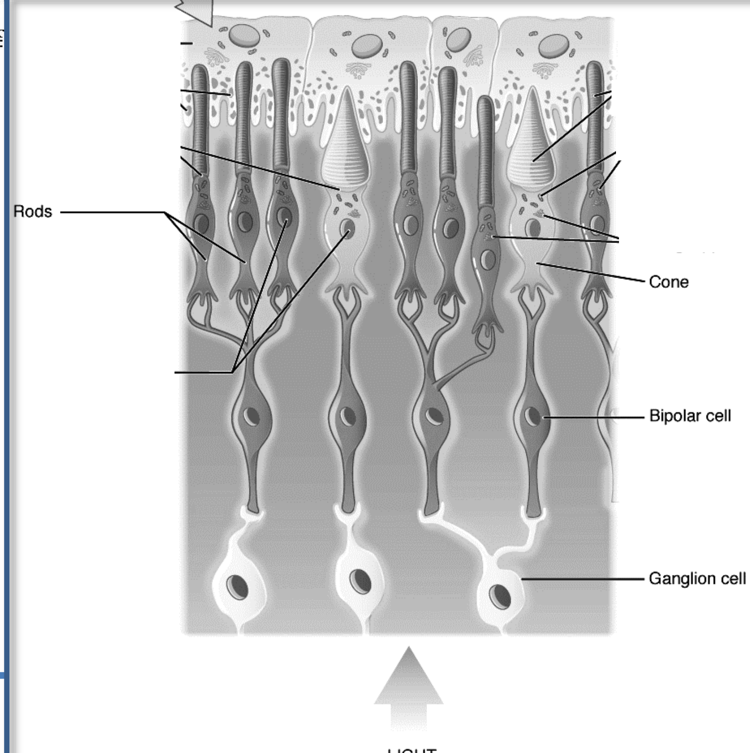
The human eye – the retina

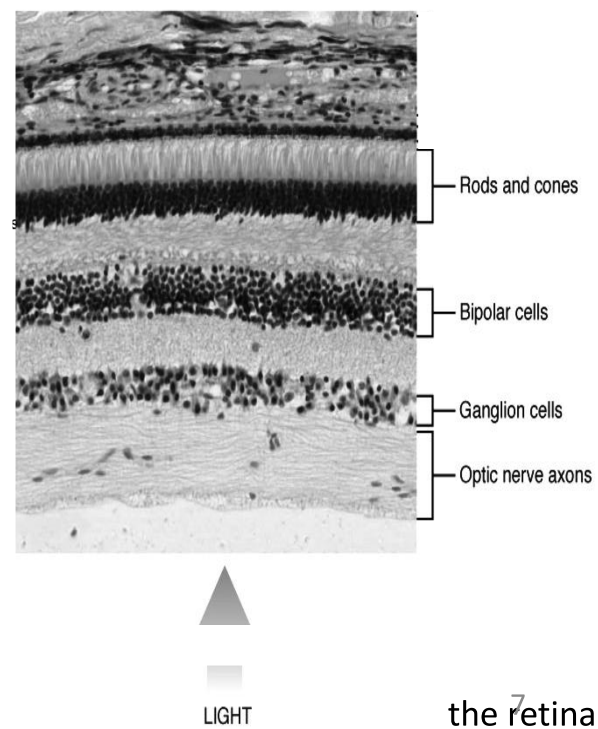




a real tomography through the retina

The retina contains two different kinds of sensor cells: ***rods***[1] and ***cones***[2].

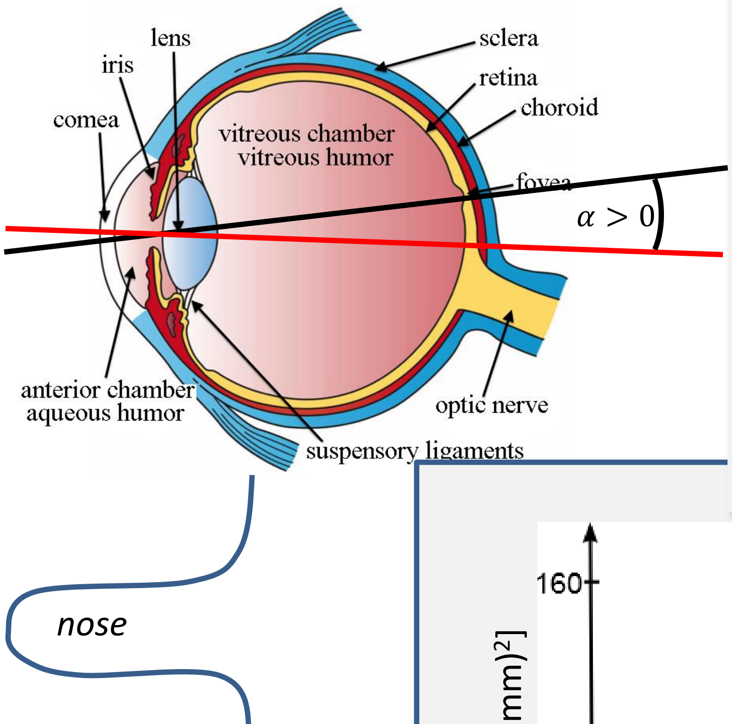




the retina

[1]Stäbchen
[2]Zäpfchen

The human eye – scopic and photopic vision

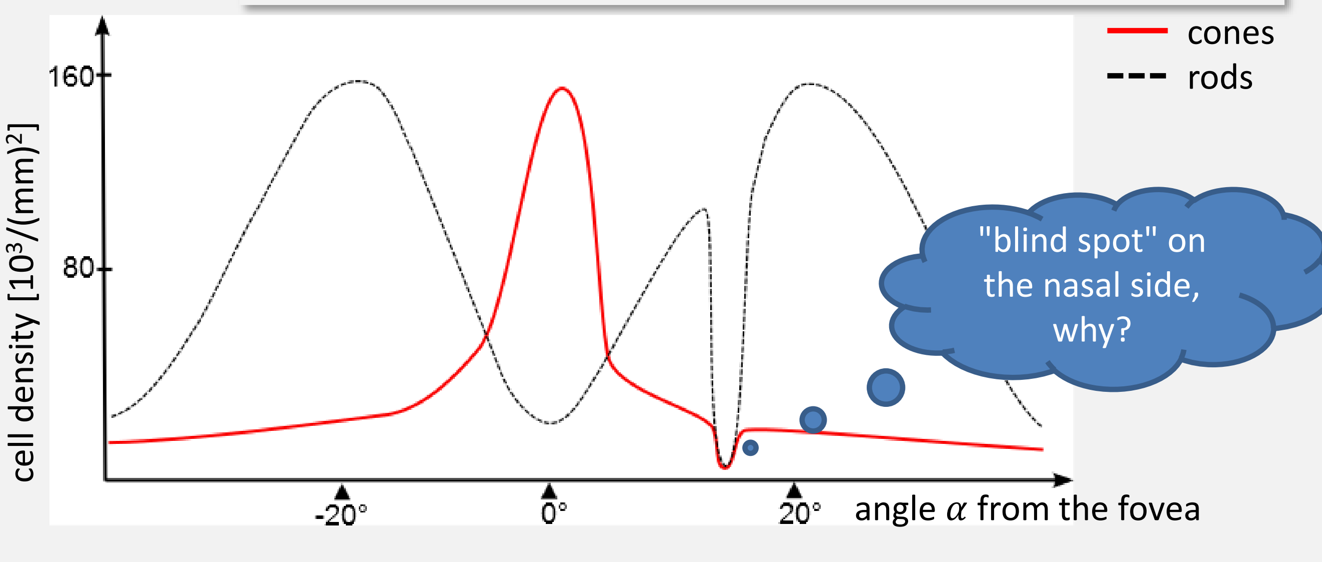


In the fovea area, the concentration of sensory cells is maximum. In this area, there are almost cones only.

Cones are responsible for color vision (but less sensitive): ***photopic vision***, rods are more sensitive to all visible wavelengths, but only allow black-and-white vision: ***scopic vision***.

The sensitivity range of the eye is incredible. *Guess the number of magnitudes between the darkest possible perception and the brightest* (bright sun).

ATTENTION: Looking into the sun DESTROYS THE EYE if exposed longer time!

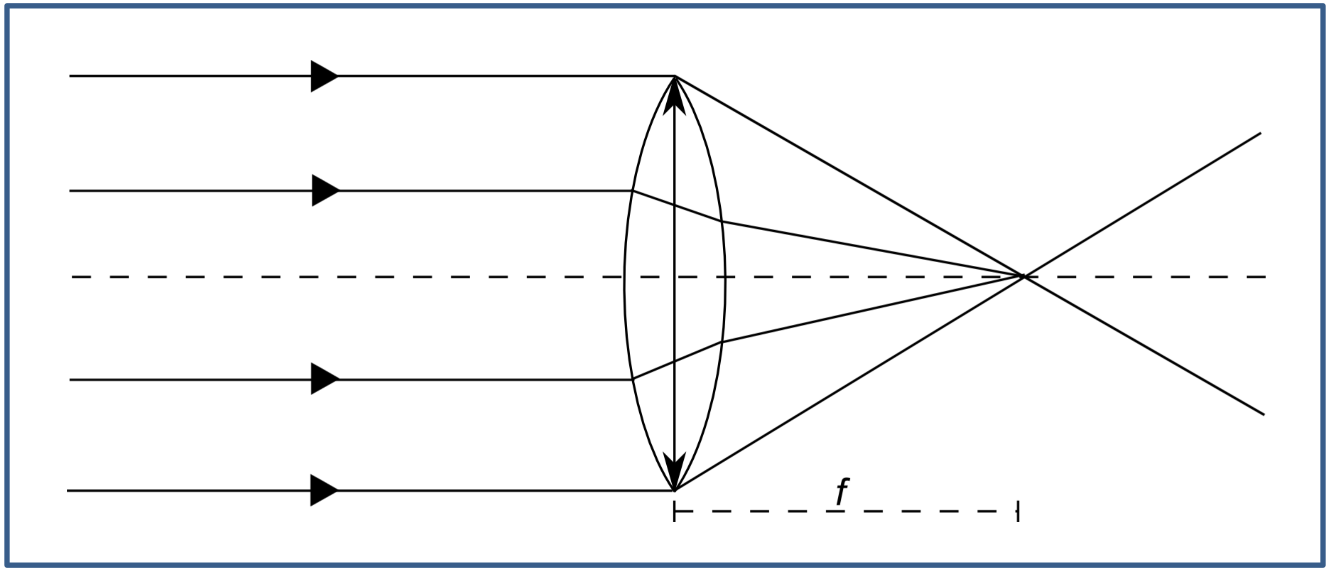


The human eye – perception

Find your blind spots! Cover one eye, look at the red circle and keep it fixated. Change the distance. At the right distance, the cross will disappear.

Introduction to optics, simple camera models

Optics – the thin lens approximation

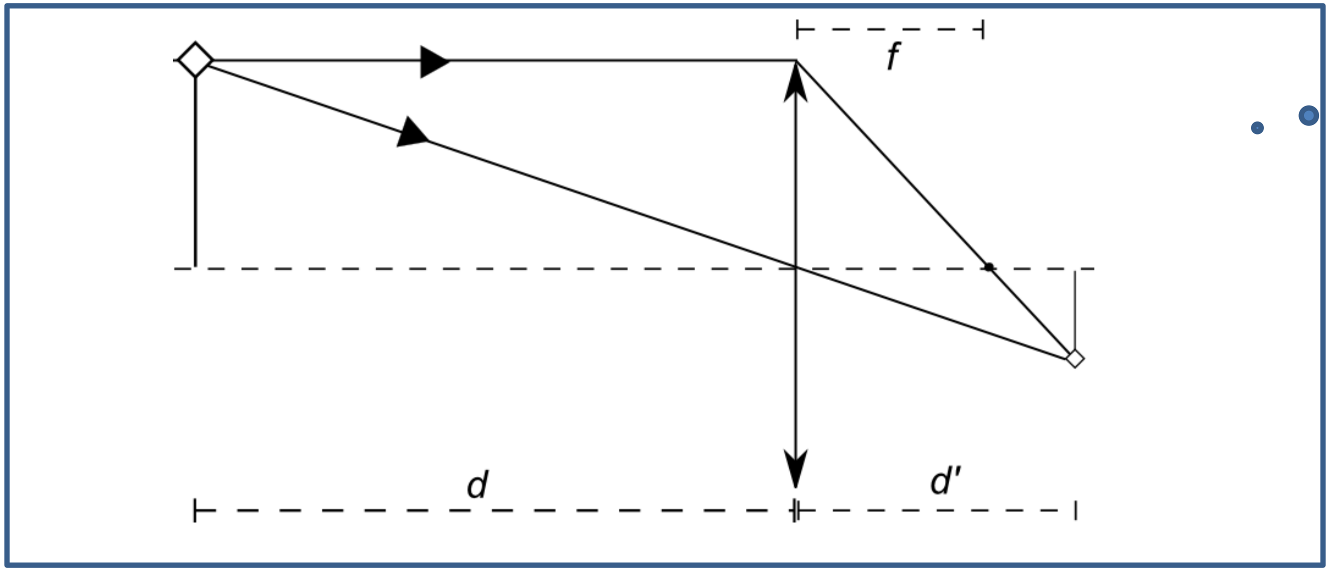


Consider a bunch of light rays, parallel to the optical axis of a convex lens (so-called ***collimated***[1] light). Parallel light rays can be observed to be emitted from objects which are very far away. The rays are ***refracted***[2] at the entry and exit of the lens according to the laws of optics (which is *Snell's law*).

All rays meet at a point, the ***focal point***[3]. It distance from the lens is called ***focal length***[4] f. If the lens is thin enough, we can approximate the geometry by a single refraction at the center plane (depicted here as ↕) of the lens.

[1]kollimiert
[2]gebrochen
[3]Brennpunkt
[4]Brennweite

Optics – image geometry



$$\frac{1}{d}+\frac{1}{d'}=\frac{1}{f}$$

Consider a closer object point ◊ at distance d from the lens. There are a lot of different, non-parallel light rays which travel from the object through the lens. It can be shown (which is difficult) that *all* meet at a certain distance d' behind the lens.

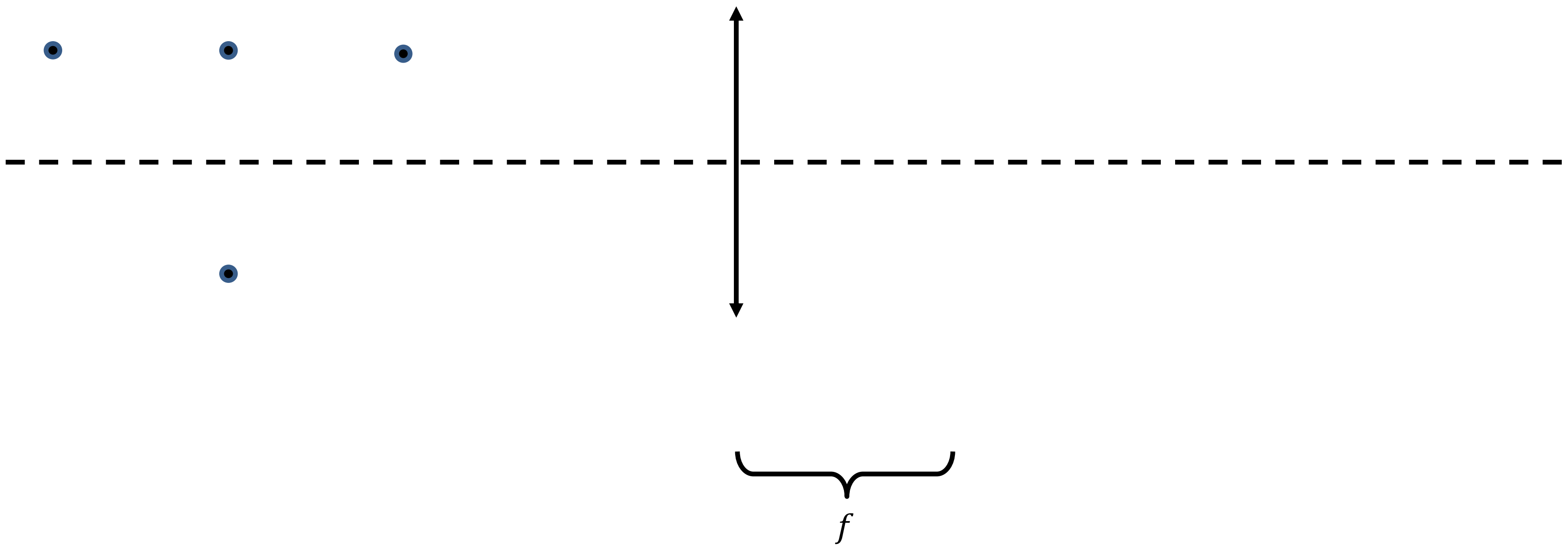
There are *two* light rays which can be followed easily:

- the ray parallel to the optic axis (which goes through the focal point) and
- the ray through the center of the lens (which is not refracted).

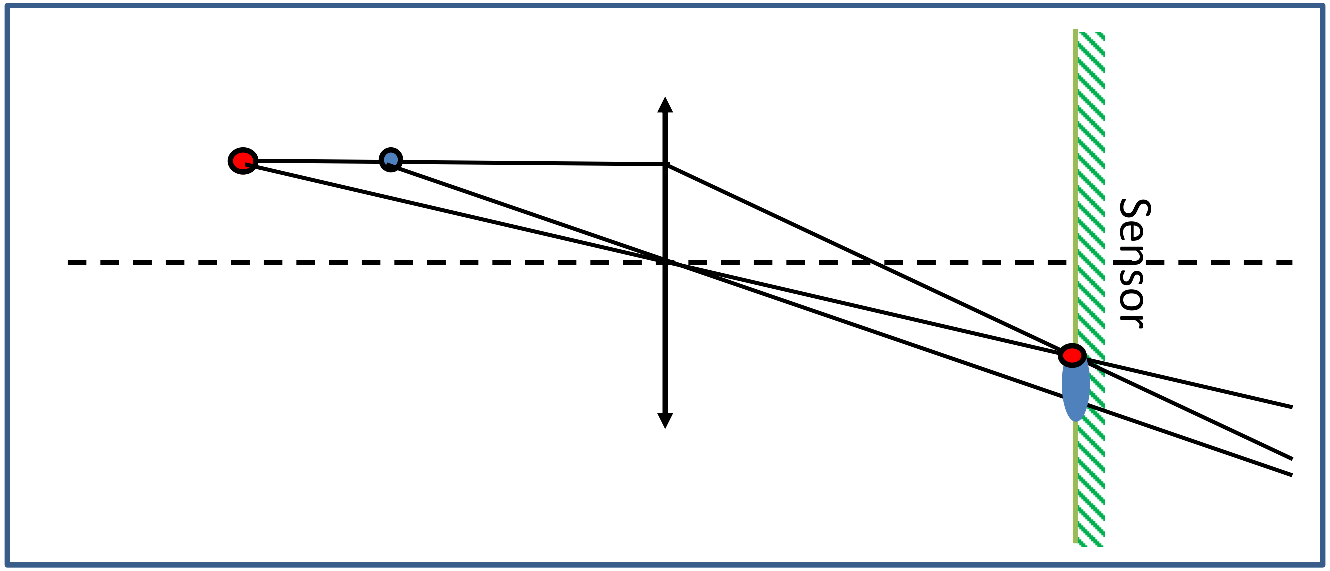
This is enough to construct the intersect point!

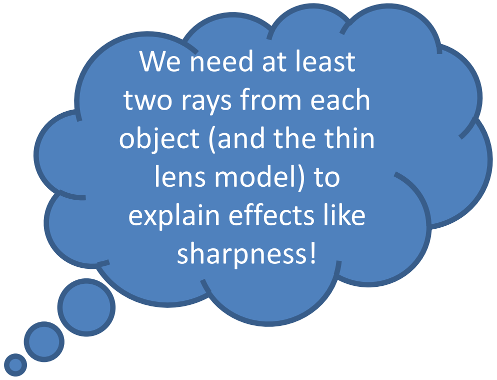
Optics – image geometry

Construct the intersect points of the following 4 object points with the thin lens model!
Can you derive some properties of the intersect points?



Optics – thin lens camera model





We can take a picture by inserting a ***sensor***. If the intersect point lies in the sensor plane, the image will be ***sharp***. Otherwise it is ***blurred***[1]. Only points of one distance d can be mapped sharply.

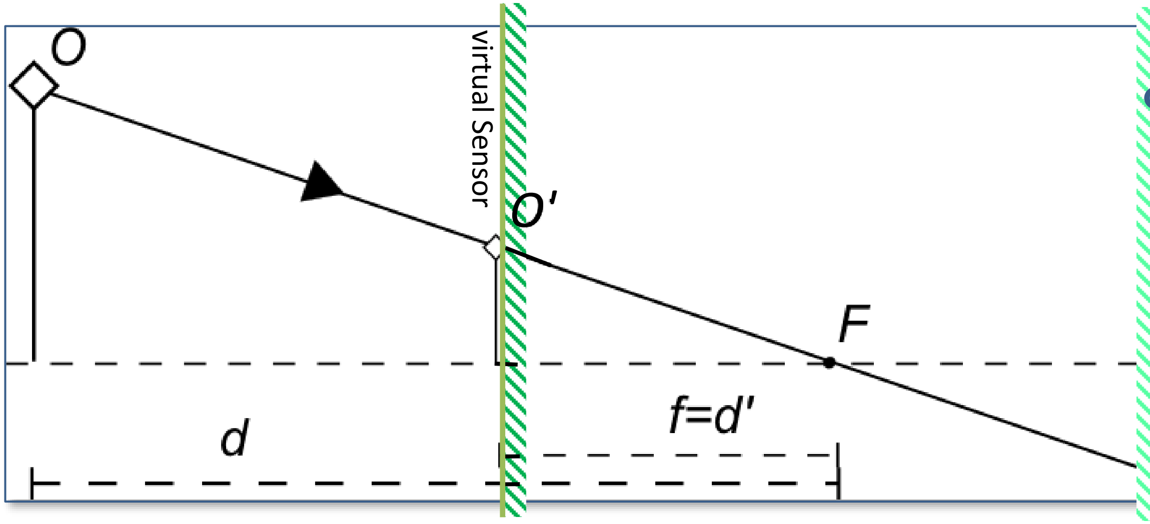
In practice, a distance range (***depth of focus***[2]) around d seems to be sharp. The closer the objects are, the more critical is the range. Most ***objectives*** (or ***lenses***) are adaptable to ***focus***[3] an object. This is done by shifting the lens slightly. ***Zoom objectives*** additionally allow to change the focus.

The image on the sensor plane is called a ***projection***.

[1]unscharf
[2]Tiefenschärfe
[3]scharfstellen

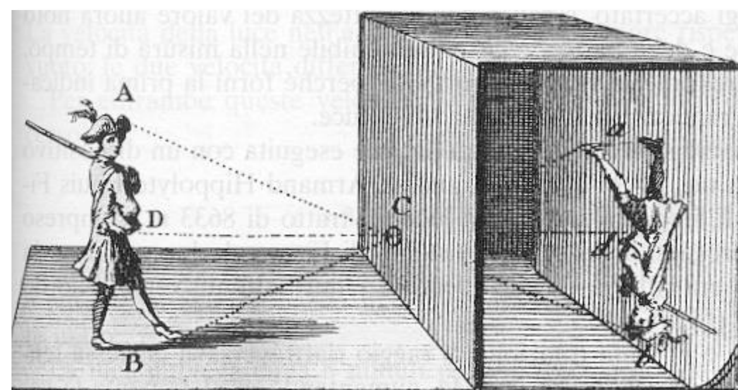
Optics – pinhole camera

pinhole model: the hole is so tight that only one ray from each object makes it through the hole



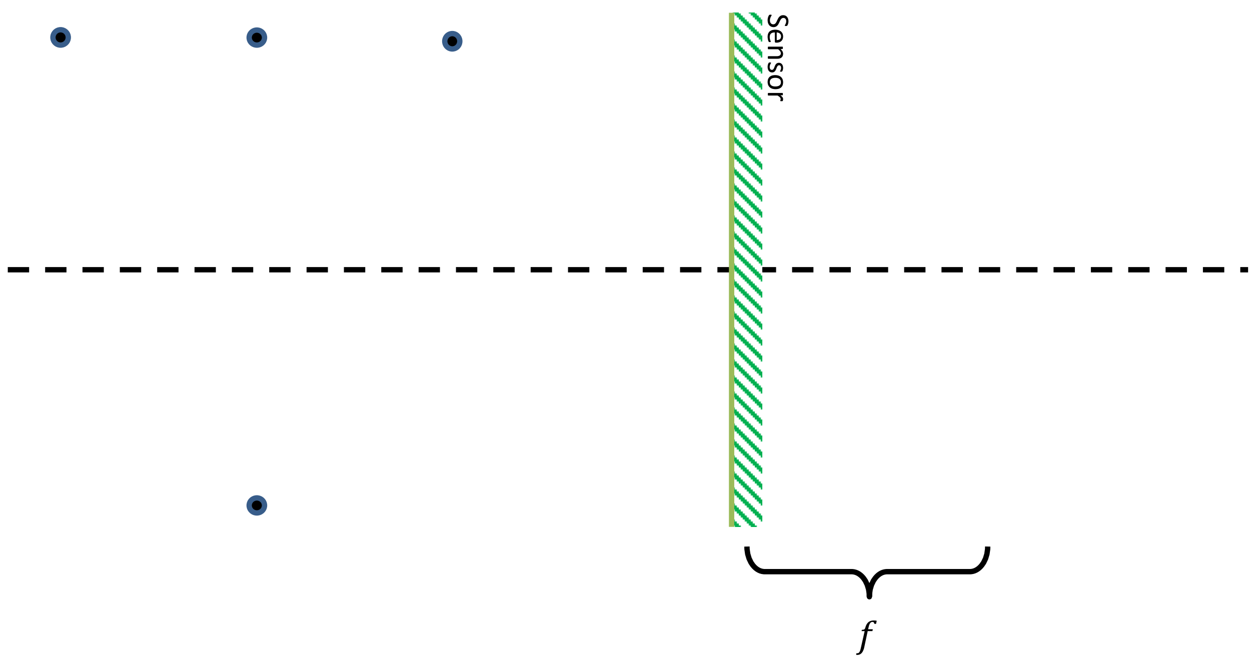
An even simpler model is the ***pinhole camera model***, inspired by the ***camera obscura*** (since about 400 BC!). Approximate the thin camera model by $d' \approx f$. In this model, we can describe sharp projections but cannot explain blur.

It is even simpler to flip the sensor plane to the other side.

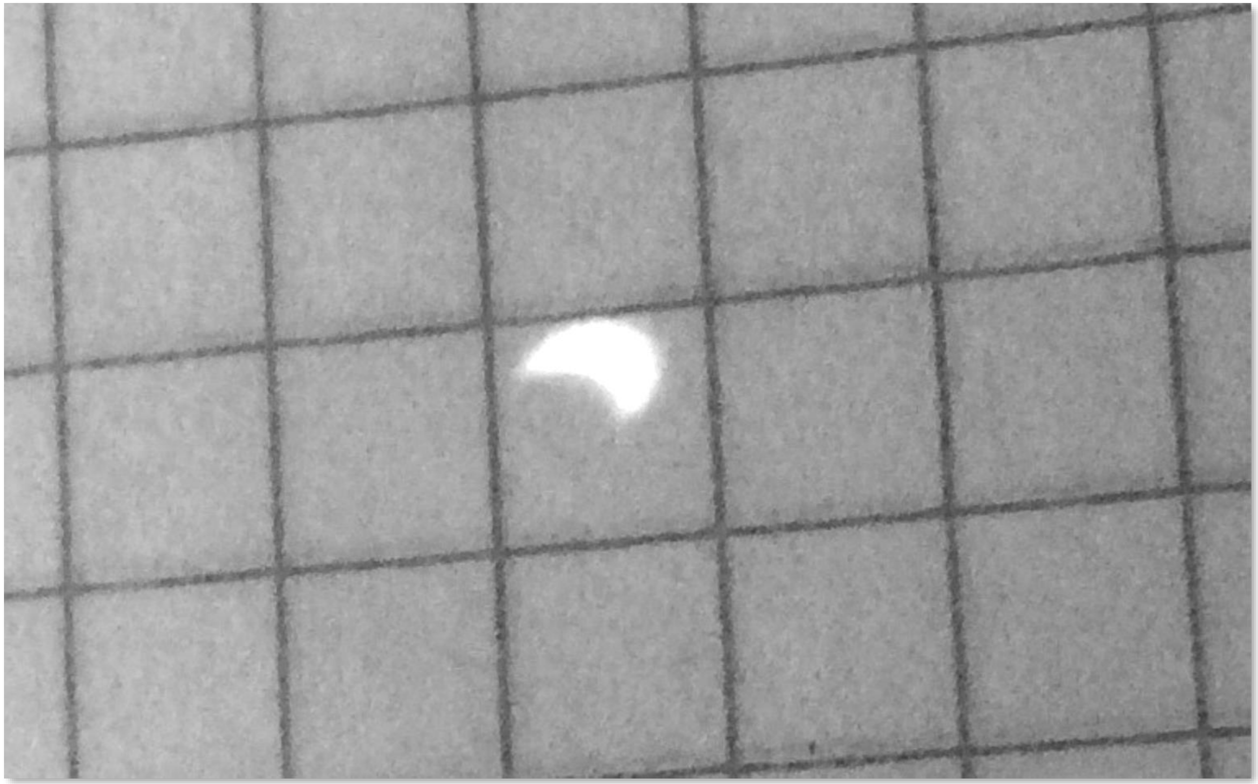


Optics – pinhole camera

Construct the images of the following 4 object points!
Can you derive some properties of the projections?



Optics – pinhole camera



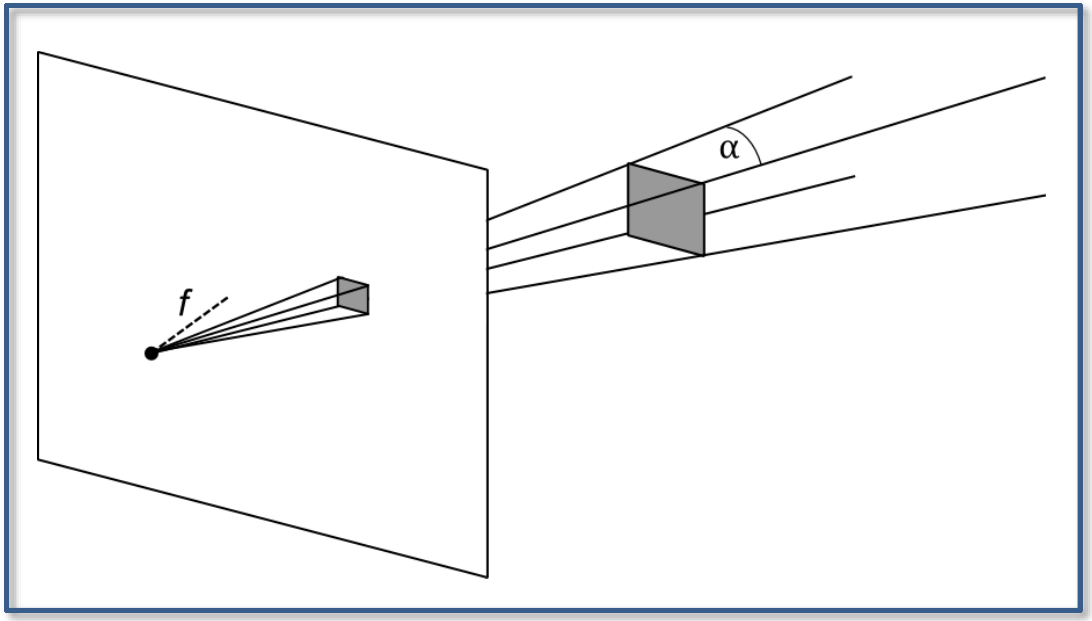
pinhole camera image of the partial eclipse on March 20, 2015 in Munich

Optics – pinhole camera

pinhole camera effect at the wall opposite some balistrarias[1] in the Castelgrande of Bellinzona.

[1]Schießscharten

Optics – aperture angles, pixels



Images are typically quantized. The image elements are called ***pixel***, the pixels are often square. The geometry of pixels is often defined by the ***pixel aperture angle***[1] α. The ***horizontal*** and ***vertical aperture angle*** of the whole image is also an important quantity.

Do not mix up with the ***aperture diaphragm***[2] of some lenses. It allows to adjust the ***brightness*** (or the number of incoming photons) of each pixel which also depends on the ***integration time*** of the sensor.

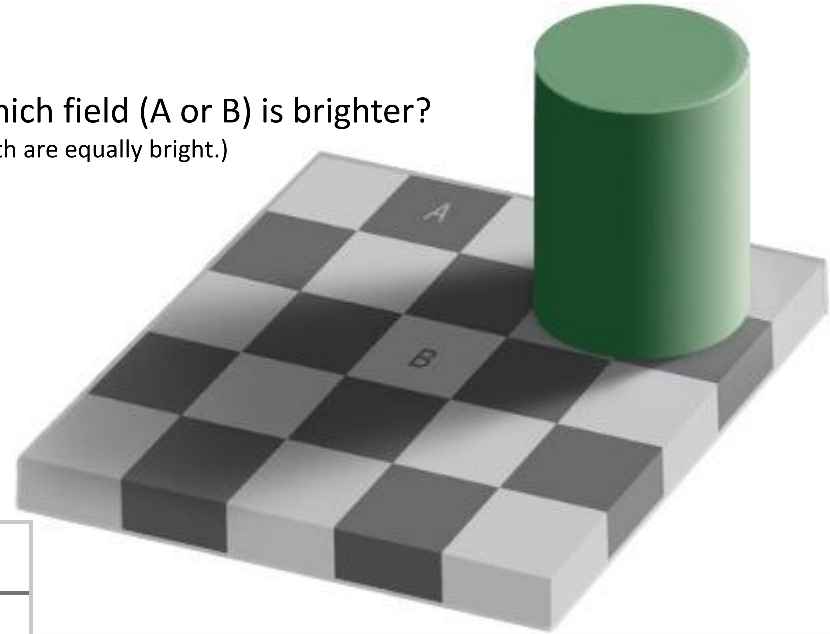
[1]Öffnungswinkel
[2]Blende

Color and brightness

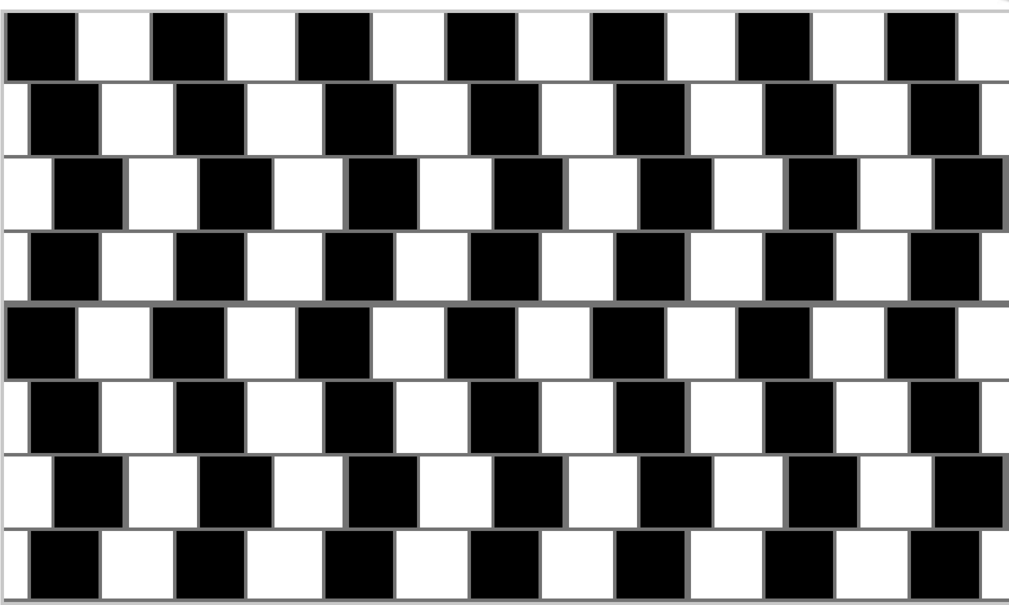
We do not ***perceive*** what we ***see***! There are a lot of processing steps between the sensory input and our perception.

Which field (A or B) is brighter?

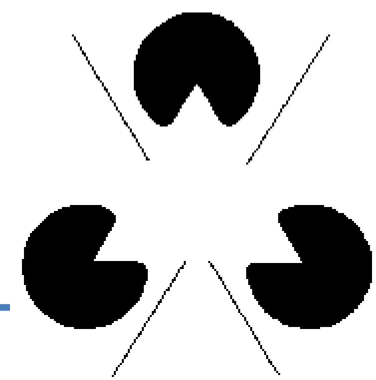
(Both are equally bright.)



Which shape do the lines between the rows have?



3 pieces of cheese and 4 salted sticks.
There is no triangle.

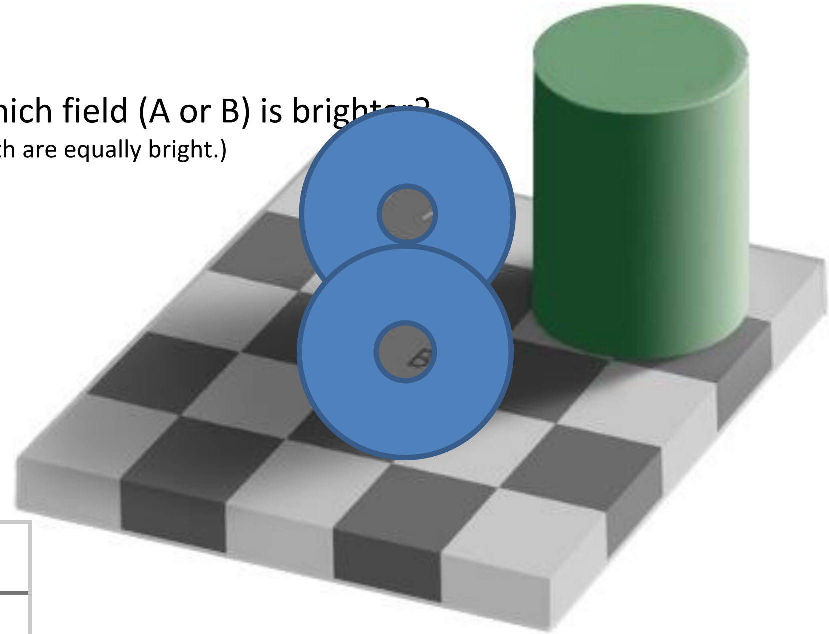


Perception

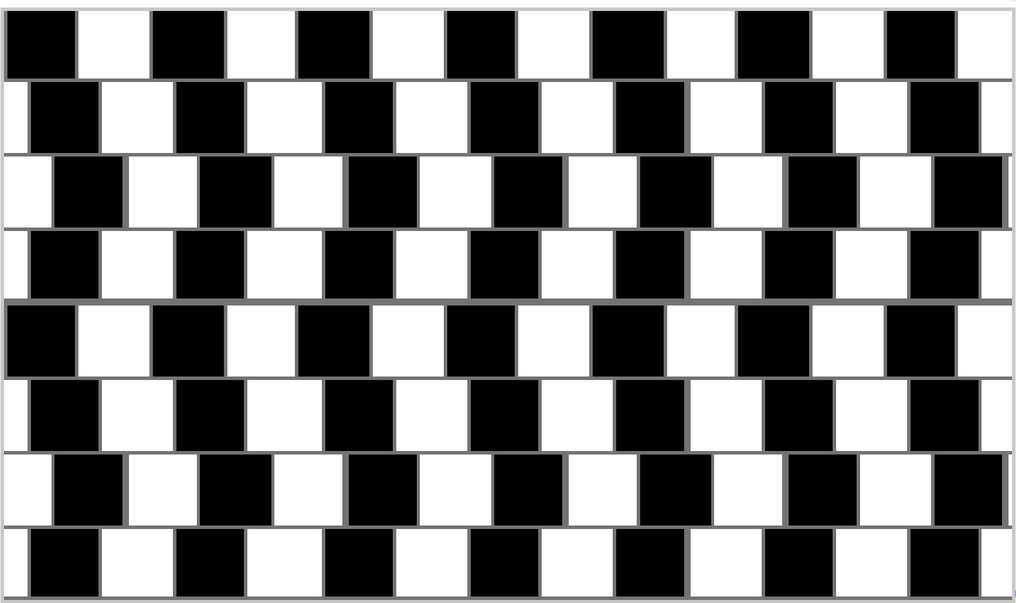
We do not ***perceive*** what we ***see***! There are a lot of processing steps between the sensory input and our perception.

Which field (A or B) is brighter?

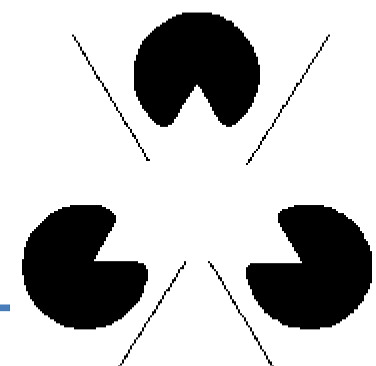
(Both are equally bright.)



Which shape do the lines between the rows have?

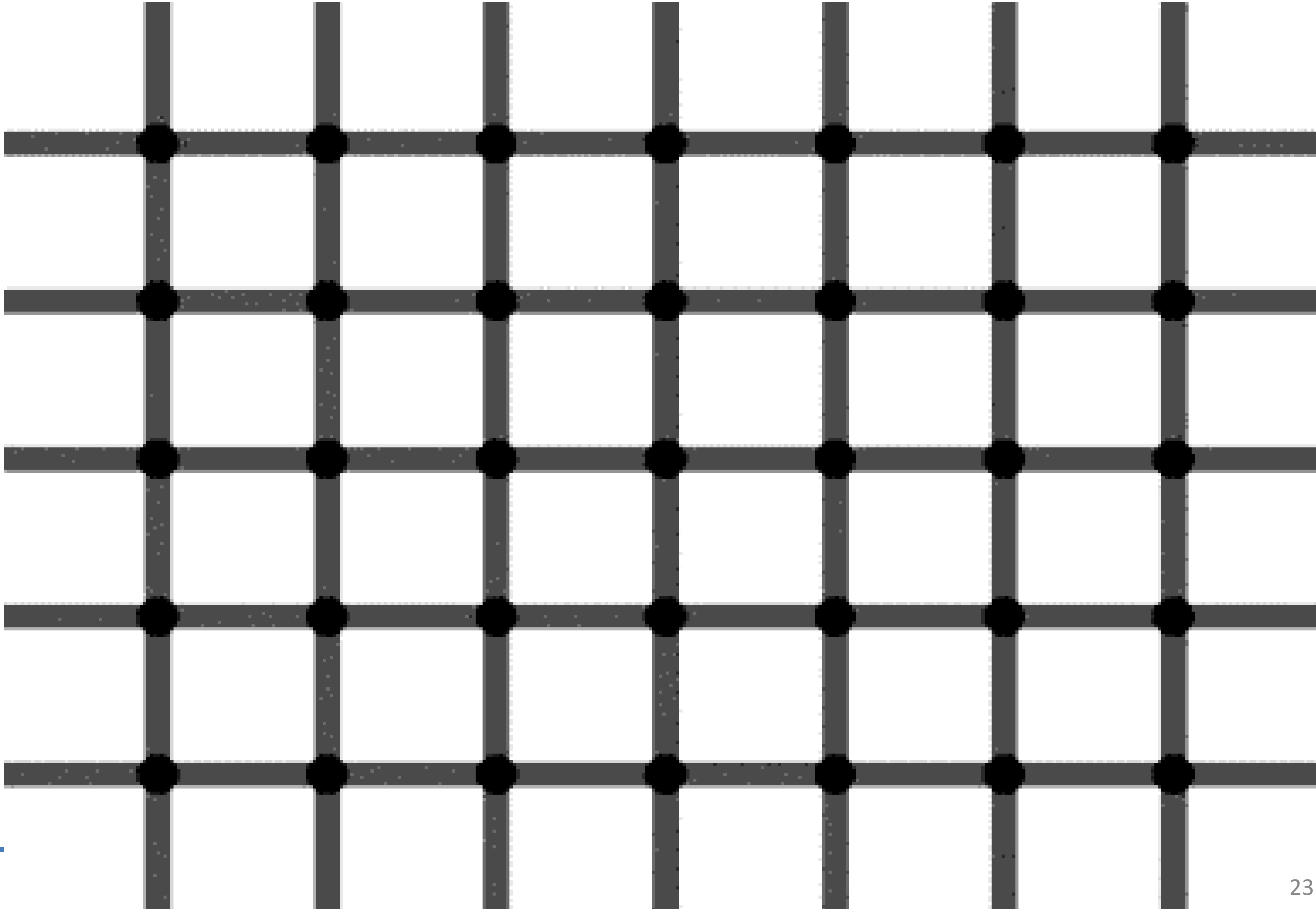


3 pieces of cheese and 4 salted sticks.
There is no triangle.



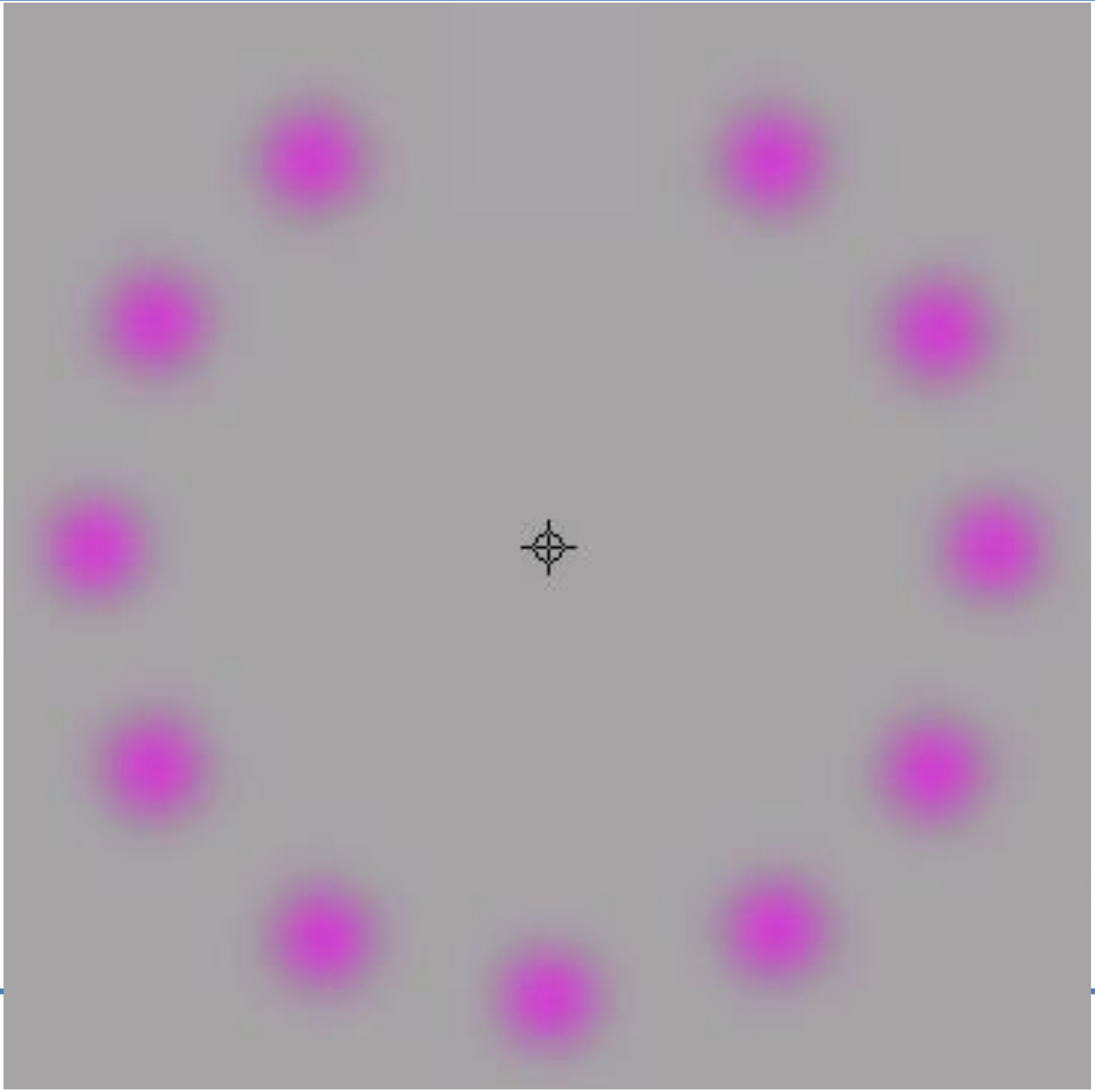
Perception

Come close and count the white circles at the crossings.



Perception

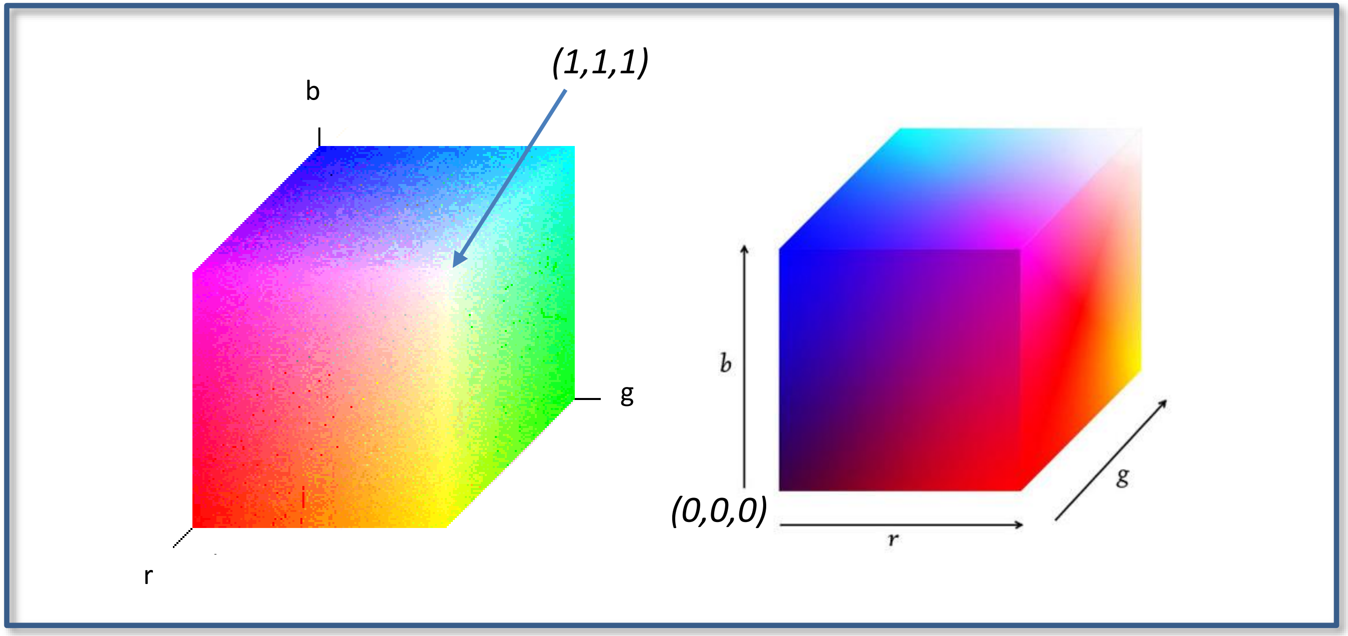
Fixate the cross and watch the moving circle (does not work on printouts;))



Color

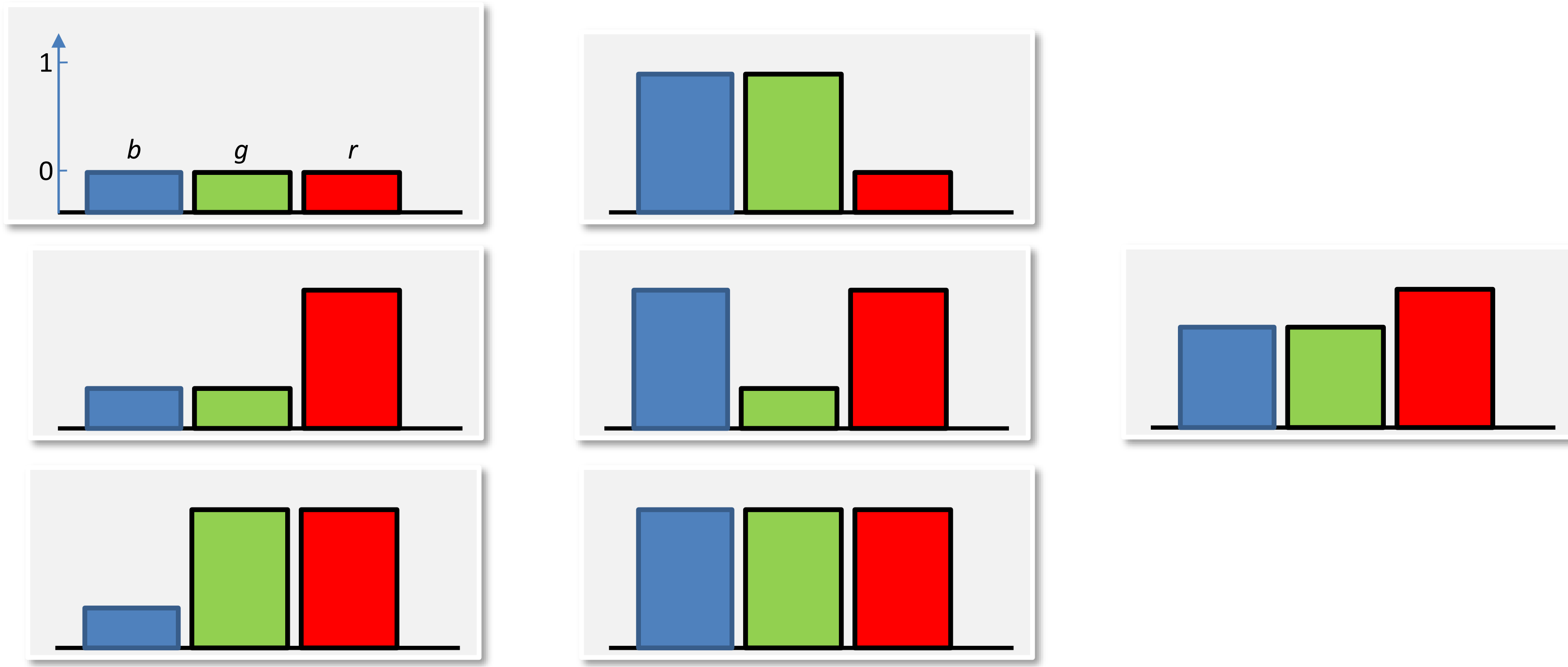
There is also no physical way to define "color" since it is not a physical concept. Only the brain *perceives* color as result of image processing steps! A simple possibility to introduce "color" is to consider the ***RGB model***.

In the RGB model, a color is defined as a triple *(r,g,b)* of 3 numbers in the range $[0,1]$. *r, g, b* are called the ***components*** of the color in the RGB model.

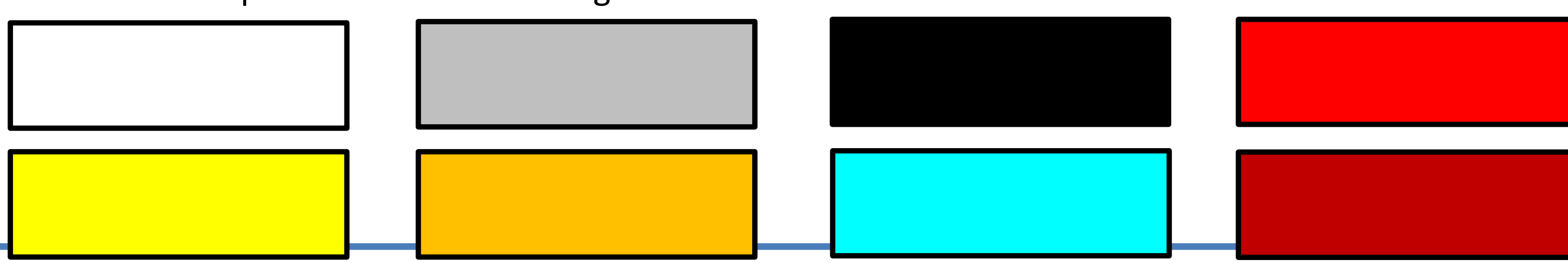


Color – assignment

What is the color expressed by component values in the RGB model?

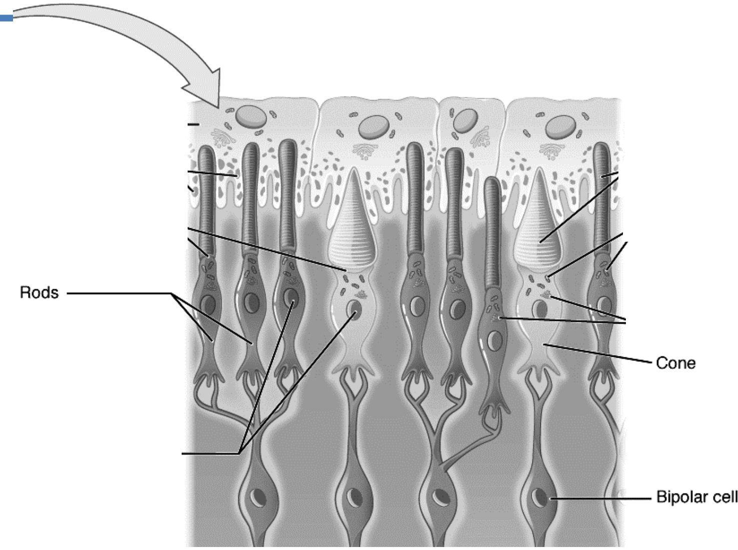


Guess the RGB components of the following colors.



Color

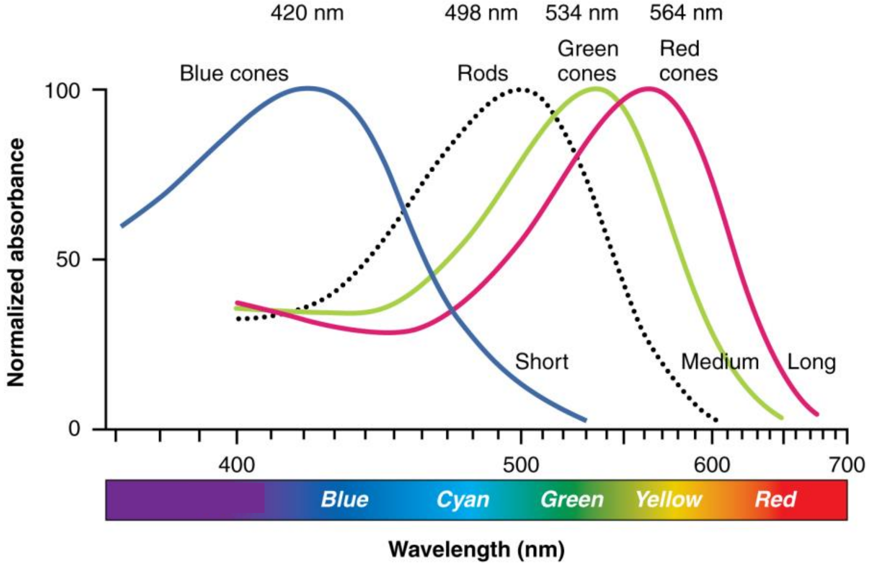
Image processing in the brain is expensive. This is a reason why the sharp field is so small (about 1°). We still need 50% of our cortex for visual perception!



There are typically 3 types of cone cells which respond to light of different wavelength:

- ***S-cones*** (also called "blue cones") have their maximum at a wavelength of 420-440 nm,
- ***M-cones*** (also called "green cones") at 534-545 nm
- ***L-cones*** (also called "red cones") at 564-580 nm which is NOT red but green-yellowish.

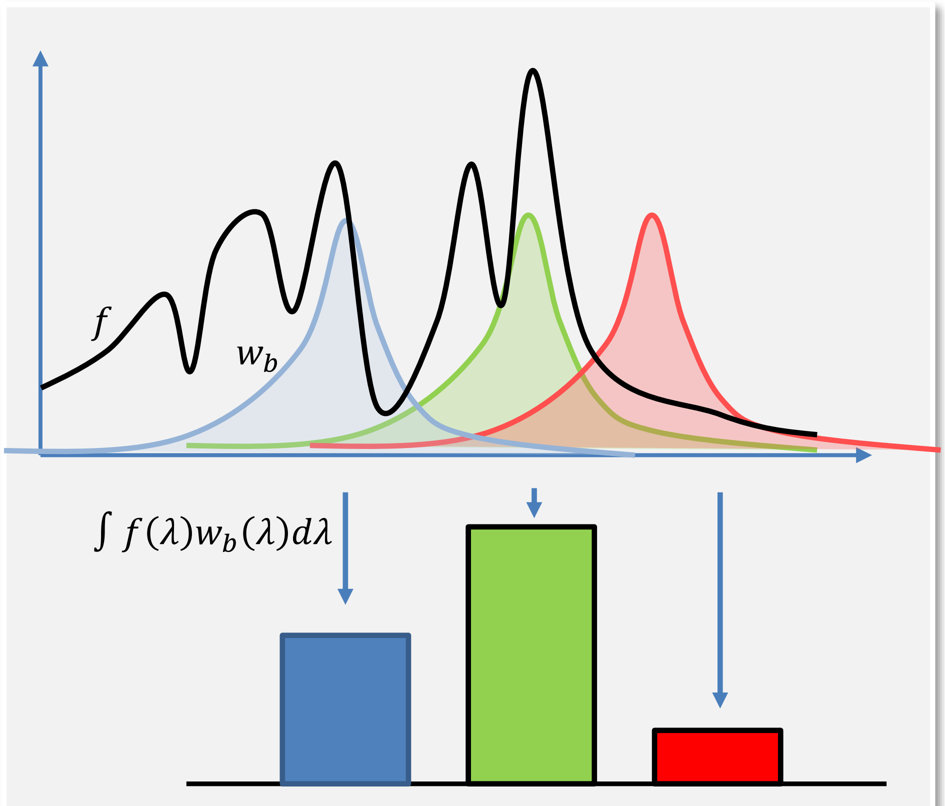
S-cones are rare (only 2%) and not concentrated on the fovea.

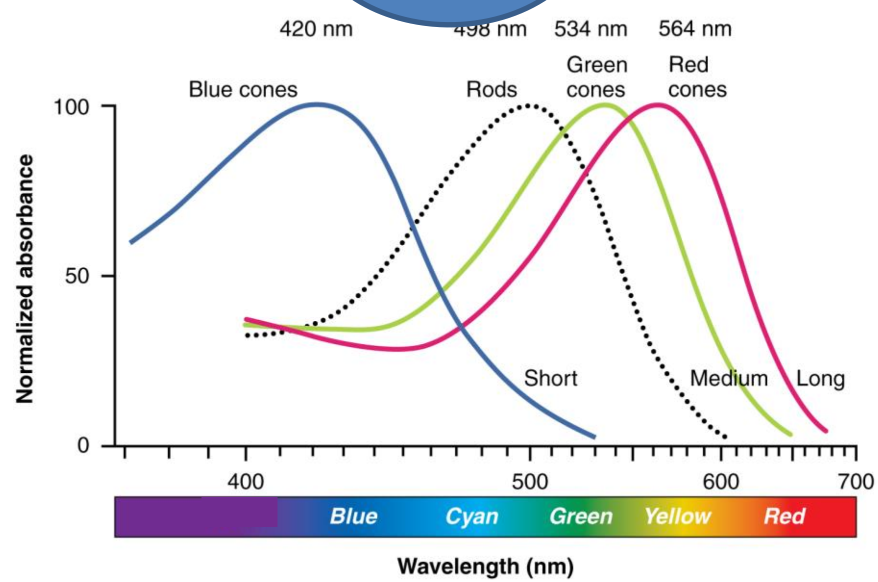


Color

There is no color in nature, color is only fiction created by our brain. In "reality" there is an electro-magnetic spectrum from which certain portions are sensed.

About 8% of all men and 0.5% of all women only have 2 working cone cell types (are ***dichromates,*** the L-cones are typically missing), also most mammals except apes[1]. Prehistoric animals, insects, birds have 4 (are ***tetrachomates***, the additional one allows for UV vision).





Which color do we see with this spectrum?

[1]dogs and cats miss the L-cones, horses the M-cones

Color – assignment

Review the left image on the last slide.
Draw two spectra (to be more precise, power density functions) which give us both the impression of "white".

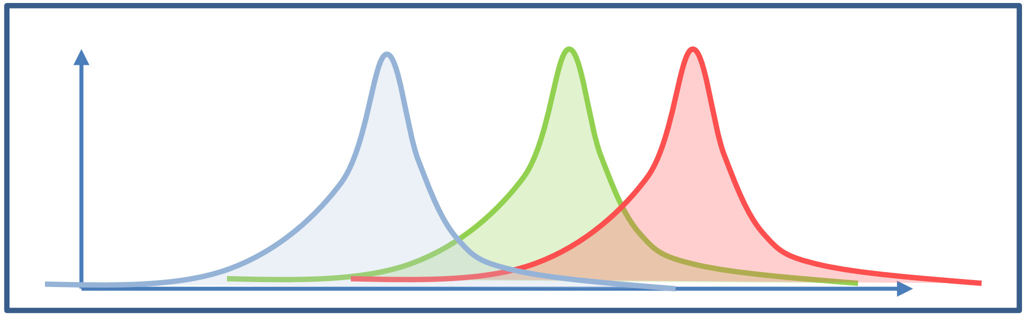


Image processing in the brain is expensive. This is a reason why the sharp field is so small (about 1°). We still need 50% of our cortex for visual perception!

The data is already preprocessed in the retina by bipolar and ganglion cells. They do a linear combination of the inputs. What comes out is information about intensity[1] I, a blueish C_b and a greenish C_g component.

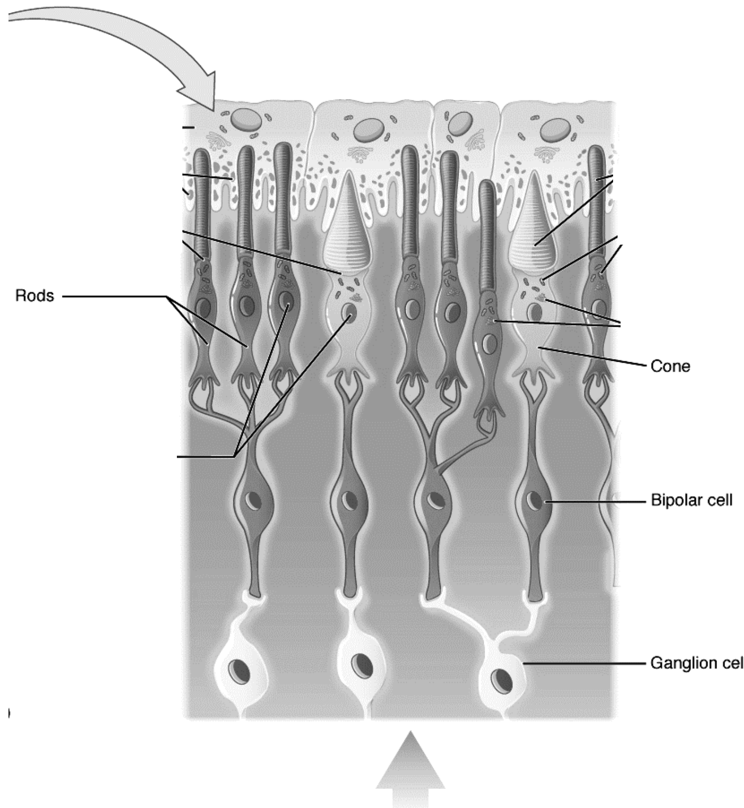
A technical standard which is close to this biological process is the ***YUV color model*** (Y: ***intensity*** in [0,1], U/V stands for two components to describe the color in [-1,1]). So, one component could be made "better interpretable" and only two are left for the color impression.

It is defined by

Y = 0.299 R + 0.587 G + 0.114 B

U = 0.493 (B-Y)

V = 0.877 (R-Y)

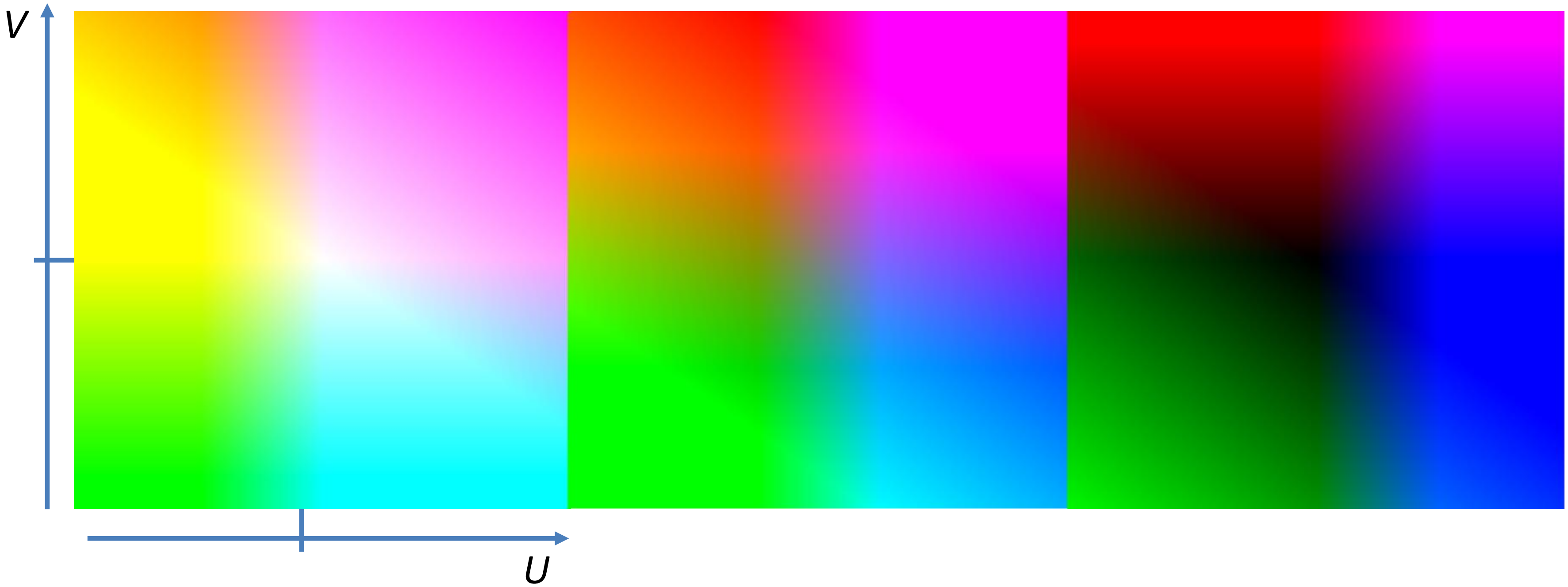


In the YUV-model:
U: ***blueish***
V: ***reddish***

output:
(I, C_b, C_g) or
(technical) (Y, U, V)

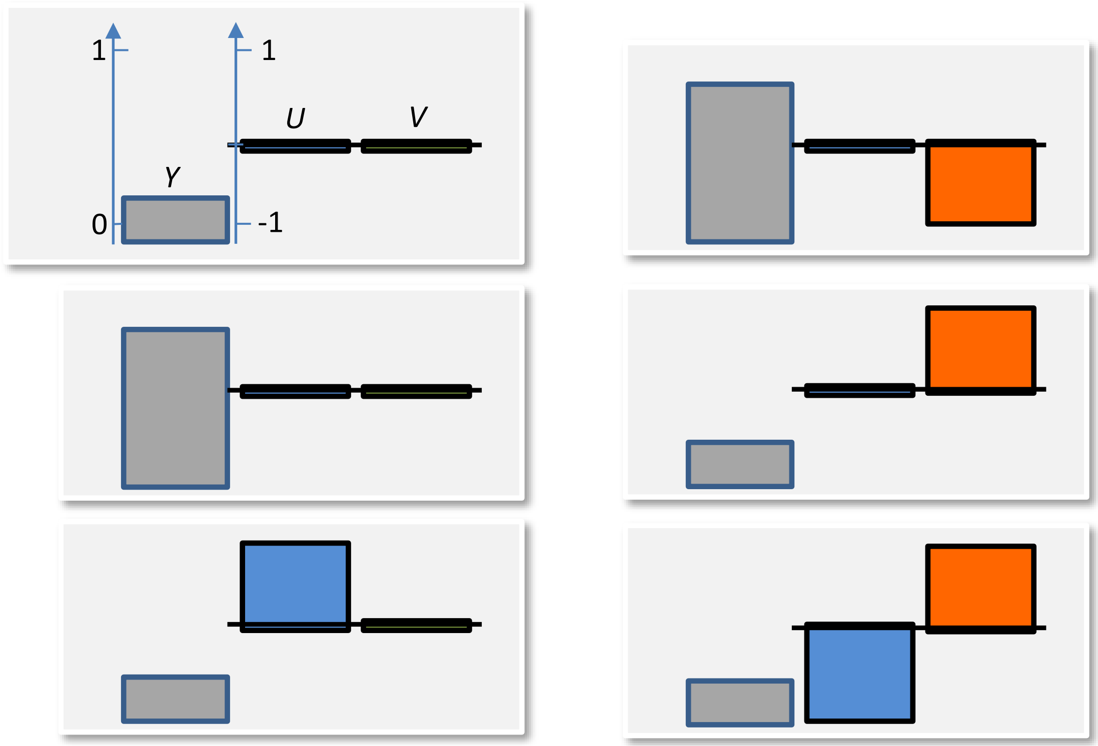
[1]*intensity* is just another word for *brightness*

$Y = 1$ $\qquad Y = \frac{1}{2}$ $\qquad Y = 0$

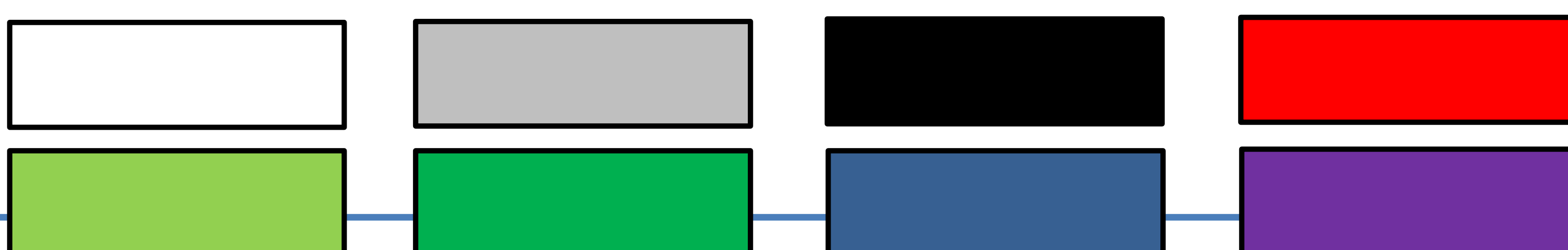


Color – assignment

What is the color expressed by the component values in the YUV model?

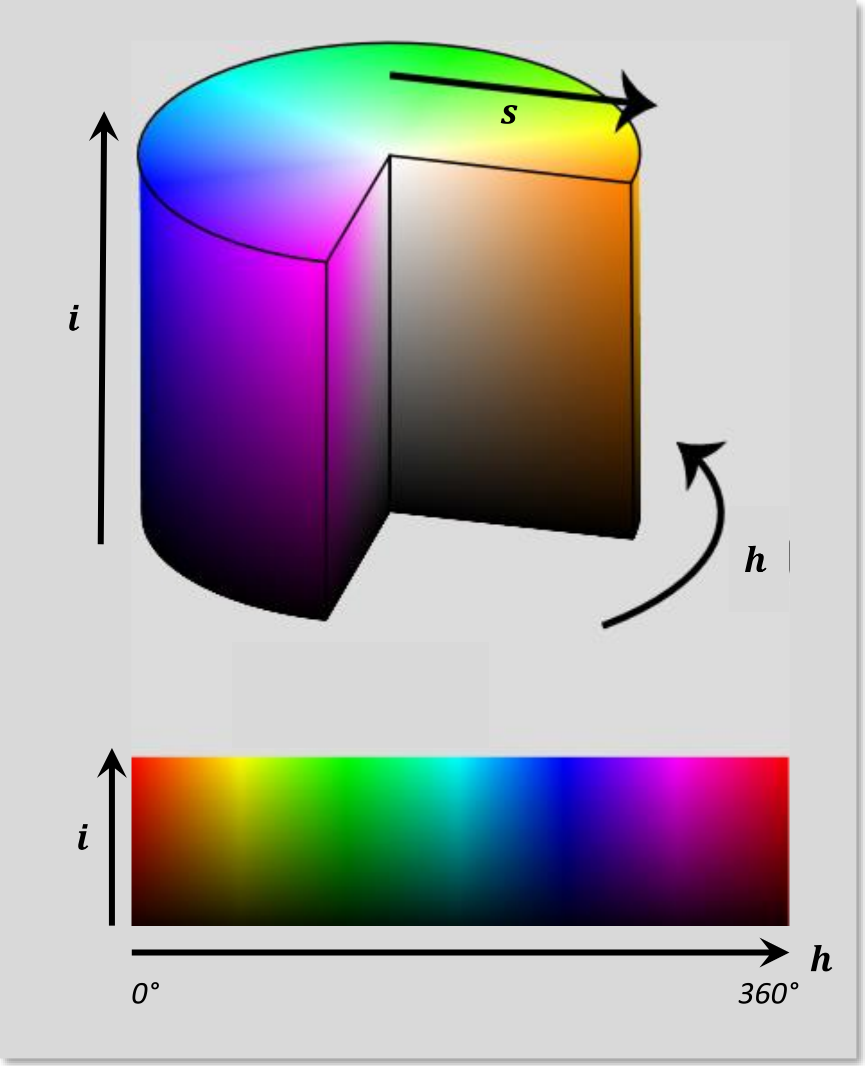


Guess the RGB components of the following colors.



Another important color space is the ***HSI space*** (hue[1], saturation, intensity). In this space, there is only one "color" component h and two "better interpretable" components s and i.

h is an angle, s and i are values in [0,1]. See the script for details.



[1]*hue* is just another word for *color*

Cameras and camera sensors

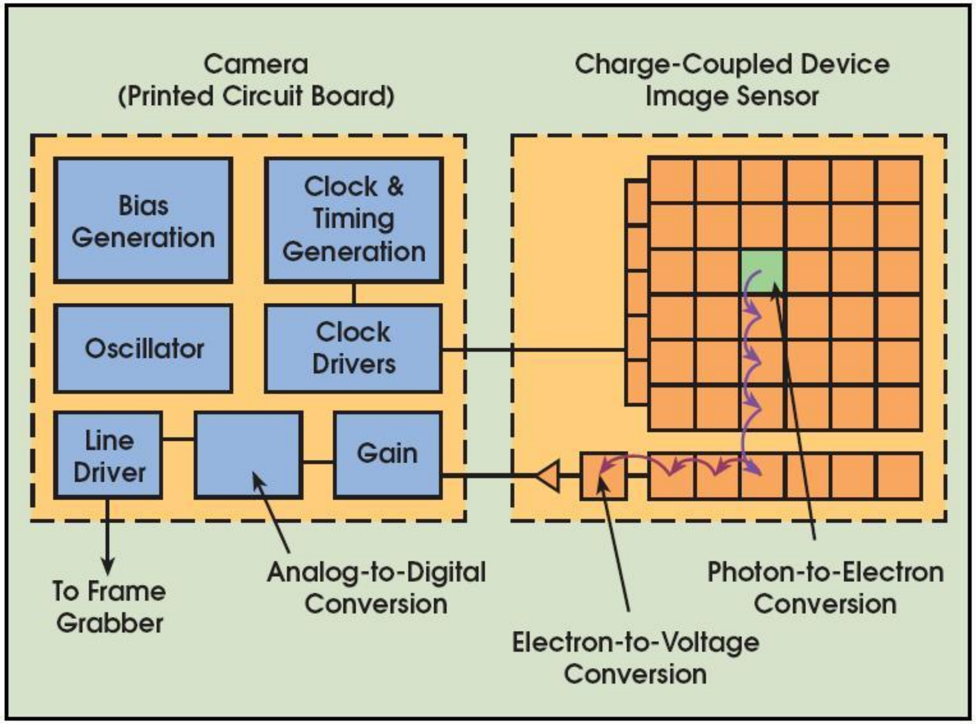
Introduction

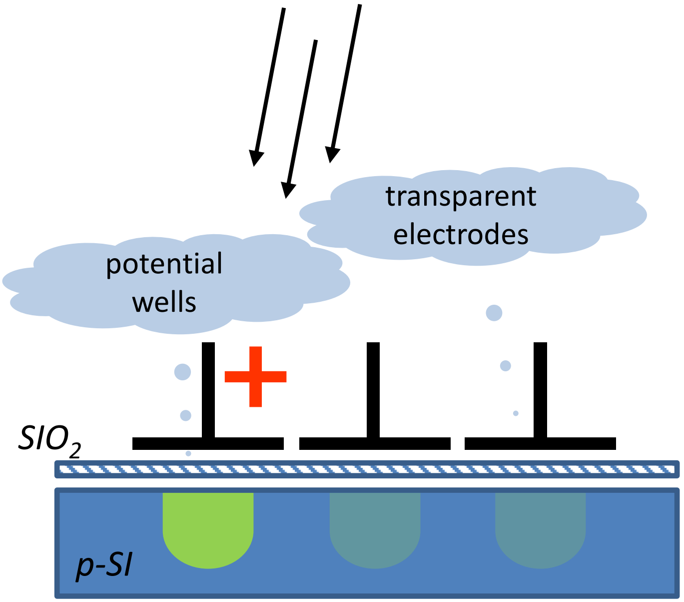
Cameras consist of

- sensor chip(s)
- lens & mount
- electronics (2 interfaces, possibly processing)

Two sensor chip technologies: CMOS & CCD

CCD



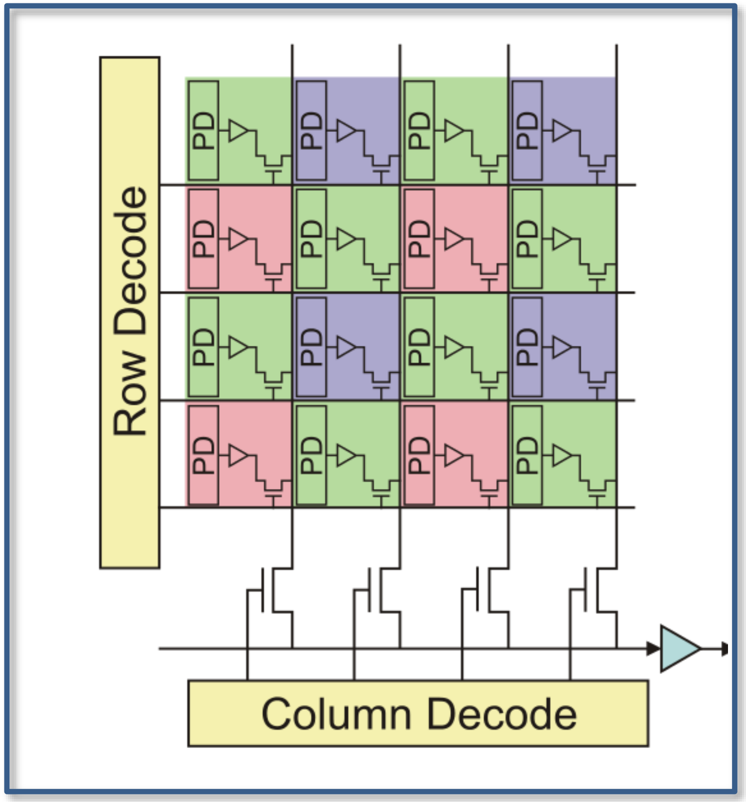


- positive/negative charge pairs are decoupled by the photoelectric effect
- the electrons gather in the well
- after exposure, the well is moved by changing the potential on the electrodes (+-- → ++- → -+-)

- different architecture possible to avoid light exposure during transport
- often covered (opaque) regions between transparent areas
- serial amplification and read out

CMOS

- uses photo diodes
- amplification for each pixel separately
- addressing like with a memory chip



Chip characteristics

Chip characteristics (see script):

- ***Spatial resolution***
- ***Pixel depth***
- ***Frame rate***
- ***Shutter***
- ***Mode (interlaced/progressive)***
- ***Pixel aspect ratio***
- ***Chip size***
- ***Chip aspect ratio***
- ***Dark noise***
- ***Dynamic range***

Chip characteristics (see script) for color chips:

- ***Color solution (1-Chip/3-Chip)***
- ***1-Chip: Pattern, typical: Bayer-Pattern:***

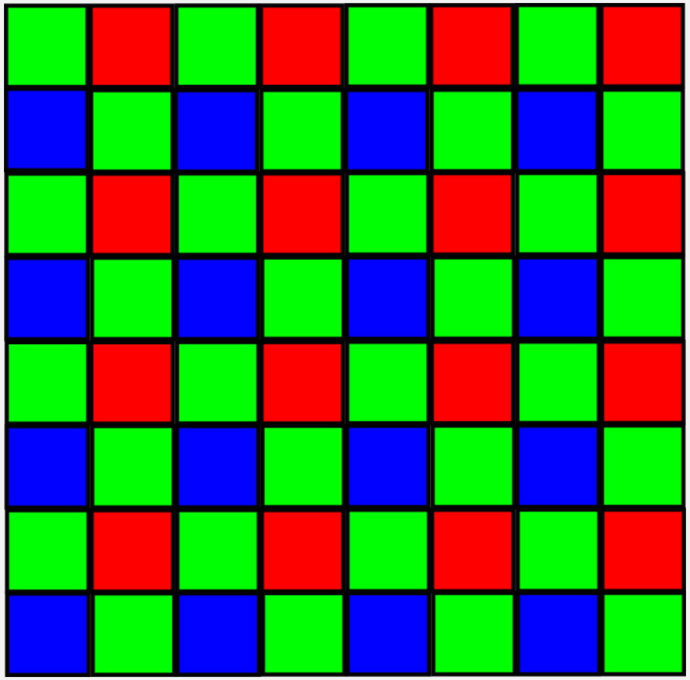


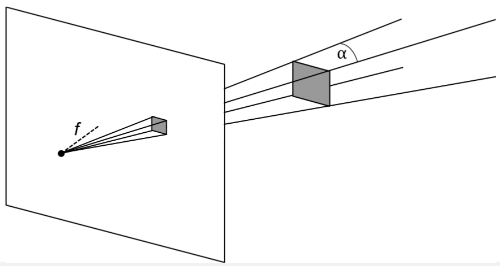
Image Processing

2. Homogeneous coordinates and applications

Alfred Schöttl

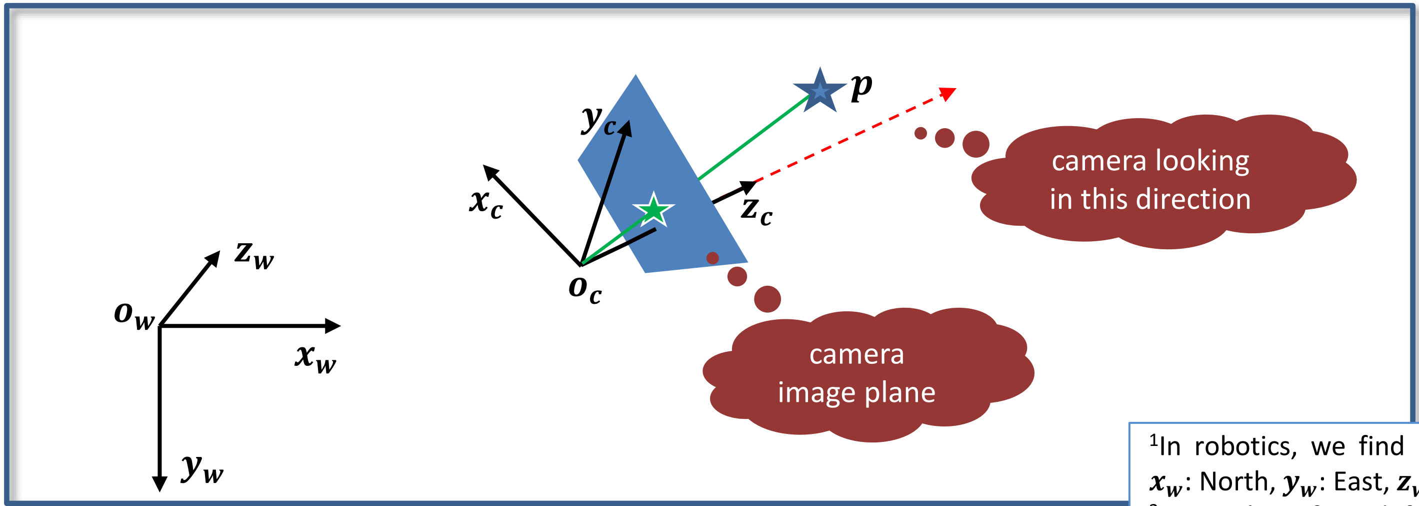
Homogeneous coordinates

Is there a way of *calculating* such a projection?



We consider (at least) two ***coordinate systems*** or ***frames***:

- The ***world frame*** w is fixed and can be ***defined freely***, often origin $\boldsymbol{o}_w$: some Earth-fixed point, $\boldsymbol{x}_w$: East, $\boldsymbol{y}_w$: down, $\boldsymbol{z}_w$: North[1]. It is our reference (often at a fixed point in a room or a map).
- The ***camera frame*** c is located in the camera, often origin $\boldsymbol{o}_c$: focal point, $\boldsymbol{z}_c$: optical axis, pointing forward[2].



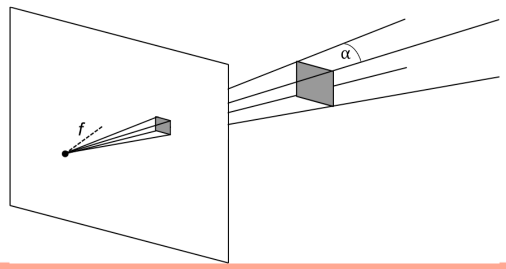
problem: find projection ★ of a point $\boldsymbol{p}$ ★

[1]In robotics, we find more often the convention $\boldsymbol{x}_w$: North, $\boldsymbol{y}_w$: East, $\boldsymbol{z}_w$: down

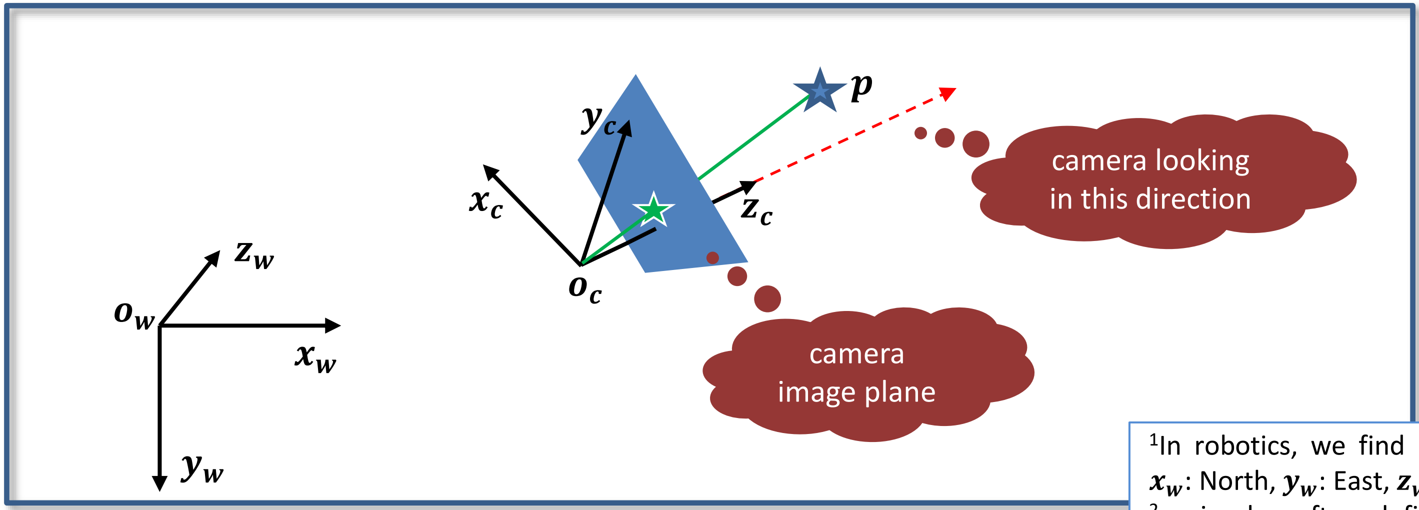
[2]$\boldsymbol{z}_c$ is also often defined to be the optical axis pointing backward

Homogeneous coordinates

Is there a way of *calculating* such a projection?



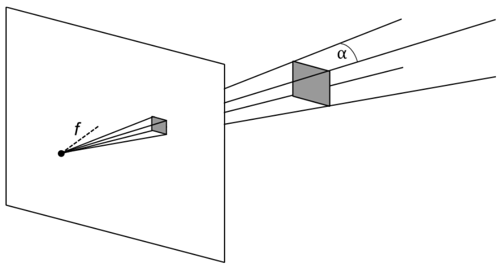
Our ***geometric*** points (such as $\boldsymbol{p}$) and frames (such as $(\boldsymbol{o}_c, \boldsymbol{x}_c, \boldsymbol{y}_c, \boldsymbol{z}_c)$) will be represented by ***coordinates***. These coordinates are a vector$\in \mathbb{R}^{2/3}$ of numbers and refer to a specific frame. We denote coordinates by an exponent: $\boldsymbol{p}^c$ are the coordinates of $\boldsymbol{p}$ in the c-frame, $\boldsymbol{o}_c^w$ are the coordinates of $\boldsymbol{o}_c$ in the w-frame.



problem: find projection ★ of a point $\boldsymbol{p}$ ★

[1]In robotics, we find more often the convention $\boldsymbol{x}_w$: North, $\boldsymbol{y}_w$: East, $\boldsymbol{z}_w$: down
[2]$\boldsymbol{z}_c$ is also often defined to be the optical axis pointing backward

As we have seen, images can be interpreted as projections of object points onto the image plane. Is there a way of *calculating* such a projection **easily**?

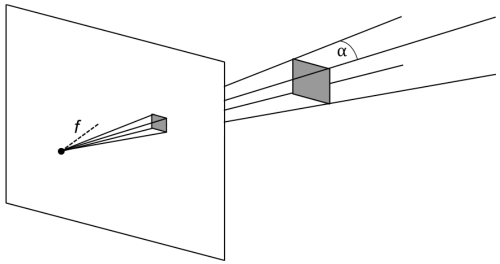


This can be done by ***projective geometry*** which uses ***homogeneous coordinates***.

- A vector $\boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \in \mathbb{R}^3$ can be written in homogeneous coordinates as $\tilde{\boldsymbol{v}} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ 1 \end{pmatrix} \in \wp^3$.
- ***All*** vectors $\tilde{\boldsymbol{w}} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} \in \wp^3$ with $w_4 \neq 0$ correspond to the Euclidean vector $\boldsymbol{w} = \begin{pmatrix} w_1/w_4 \\ w_2/w_4 \\ w_3/w_4 \end{pmatrix} \in \mathbb{R}^3$.

The same trick can be applied to $\mathbb{R}^2/\wp^2$. In contrast, the classical Euclidean coordinates $\in \mathbb{R}^3$ are called ***inhomogeneous coordinates***, the process of converting between inhomogeneous and homogeneous coordinates is called ***homogenization*** and ***dehomogenization.***

Why homogeneous coordinates?



As we will see, many basic operations (coordinate transformations, projections etc.) can be expressed as a multiplication of a matrix and the homogeneous coordinates vector!

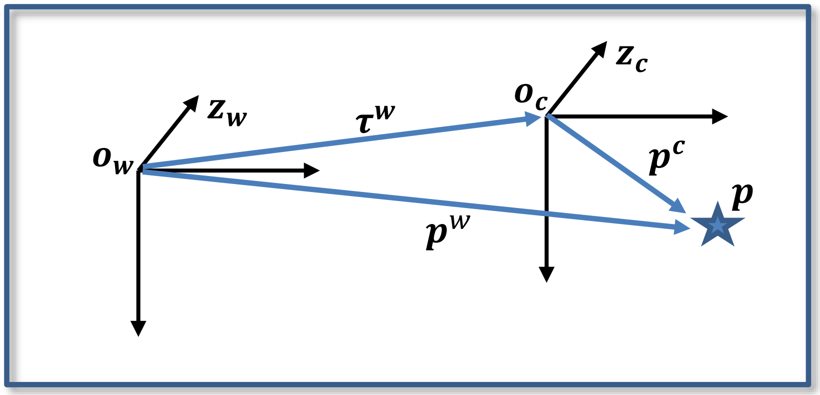
A coordinate transformation from coordinates in a to coordinates in b can be calculated by

$$\tilde{\boldsymbol{p}}^b = T_a^b \tilde{\boldsymbol{p}}^a$$

with a ***transformation matrix*** T_a^b. You just need to know how to build T_a^b.

Translation

Let the camera system just be translated by a translation vector $\boldsymbol{\tau}^w$ expressed in the world frame and let be $\boldsymbol{p}^w$ the coordinates of $\boldsymbol{p}$ in the world frame. What is $\boldsymbol{p}^c$?



It is $\boldsymbol{p}^w = \boldsymbol{p}^c + \boldsymbol{\tau}^w$ or $\boldsymbol{p}^c = \boldsymbol{p}^w - \boldsymbol{\tau}^w$.

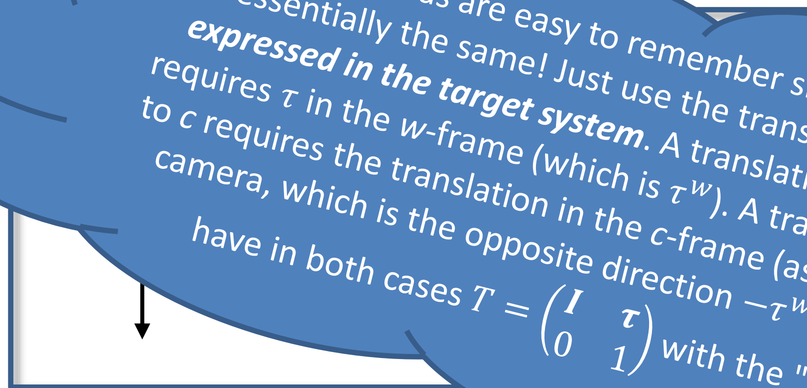
check it!

Let $T_c^w = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{\tau}^w \\ \boldsymbol{0} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \tau_1^w \\ 0 & 1 & 0 & \tau_2^w \\ 0 & 0 & 1 & \tau_3^w \\ 0 & 0 & 0 & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^w = \boldsymbol{p}^c + \boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^w = T_c^w \tilde{\boldsymbol{p}}^c.}$

Let $T_w^c = \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{\tau}^w \\ \boldsymbol{0} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -\tau_1^w \\ 0 & 1 & 0 & -\tau_2^w \\ 0 & 0 & 1 & -\tau_3^w \\ 0 & 0 & 0 & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^c = \boldsymbol{p}^w - \boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w.}$

Translation

Let the camera system just be translated by a translation vector $\boldsymbol{\tau}^w$ expressed in the world frame and let be $\boldsymbol{p}^w$ the coordinates of $\boldsymbol{p}$ in the world frame. What is $\boldsymbol{p}^c$?



The formulas are easy to remember since both are essentially the same! Just use the translation vector ***expressed in the target system***. A translation from c to w requires τ in the w-frame (which is τ^w). A translation from w to c requires the translation in the c-frame (as seen from the camera, which is the opposite direction $-\tau^w$). Hence, we have in both cases $T = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{\tau} \\ 0 & 1 \end{pmatrix}$ with the "right" $\boldsymbol{\tau}$.

It is $\boldsymbol{p}^w = \boldsymbol{p}^c + \boldsymbol{\tau}^w$ or $\boldsymbol{p}^c = \boldsymbol{p}$

Let $T_c^w = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{\tau}^w \\ \boldsymbol{0} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \tau_1^w \\ 0 & 1 & 0 & \tau_2^w \\ 0 & 0 & 1 & \tau_3^w \\ 0 & 0 & 0 & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^w = \boldsymbol{p}^c + \boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^w = T_c^w \tilde{\boldsymbol{p}}^c.}$

Let $T_w^c = \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{\tau}^w \\ \boldsymbol{0} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -\tau_1^w \\ 0 & 1 & 0 & -\tau_2^w \\ 0 & 0 & 1 & -\tau_3^w \\ 0 & 0 & 0 & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^c = \boldsymbol{p}^w - \boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w.}$

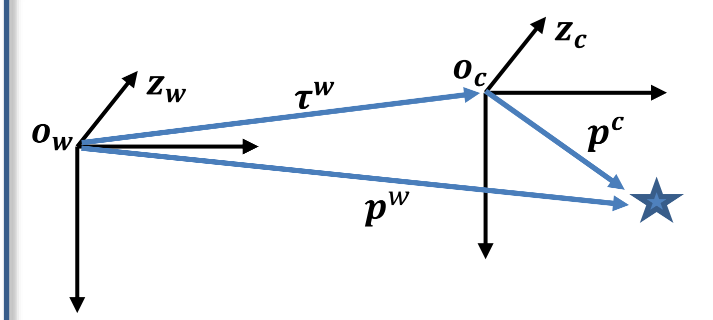
Homogeneous coordinates - Assignment

A camera at world coordinates (20, 0, 10) takes a picture of a tree. The camera is not rotated with respect to the world system. The tree is located at camera coordinates (100, -2, 5). What are the inhomogeneous coordinates of the tree in the world frame? Use homogeneous coordinates.

$\boldsymbol{p}^c =$ $\qquad$ $\tilde{\boldsymbol{p}}^c =$

$T_c^w =$

$\tilde{\boldsymbol{p}}^w = T_c^w \, \tilde{\boldsymbol{p}}^c =$



Let $T_c^w = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{\tau}^w \\ \boldsymbol{0} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \tau_1^w \\ 0 & 1 & 0 & \tau_2^w \\ 0 & 0 & 1 & \tau_3^w \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

Then, $\tilde{\boldsymbol{p}}^w = T_c^w \tilde{\boldsymbol{p}}^c$

Homogeneous coordinates - Assignment

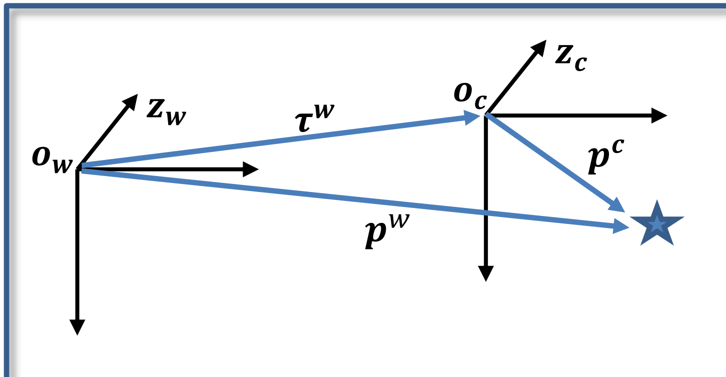
A camera at world coordinates (20, 0, 10) takes a picture of a tree. The camera is not rotated with respect to the world system. The tree is located at camera coordinates (100, -2, 5). What are the inhomogeneous coordinates of the tree in the world frame? Use homogeneous coordinates.

$$\boldsymbol{p}^c = \begin{pmatrix} 100 \\ -2 \\ 5 \end{pmatrix} \qquad \tilde{\boldsymbol{p}}^c = \begin{pmatrix} 100 \\ -2 \\ 5 \\ 1 \end{pmatrix}$$

$$T_c^w = \begin{pmatrix} 1 & 0 & 0 & 20 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\tilde{\boldsymbol{p}}^w = T_c^w \, \tilde{\boldsymbol{p}}^c = \begin{pmatrix} 120 \\ -2 \\ 15 \\ 1 \end{pmatrix}.$$

The inhomogenous coordinates of the tree in the world frame are (120, -2, 15).

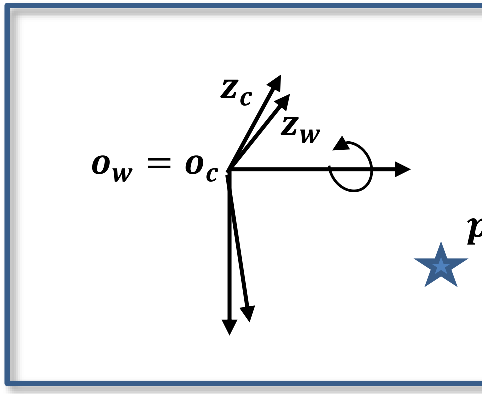


$$\text{Let } T_c^w = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{\tau}^w \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \tau_1^w \\ 0 & 1 & 0 & \tau_2^w \\ 0 & 0 & 1 & \tau_3^w \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$\text{Then, } \tilde{\boldsymbol{p}}^w = T_c^w \tilde{\boldsymbol{p}}^c$$

Rotation

Let the camera system just be rotated by a ***rotation matrix*** C_w^c and let be $\boldsymbol{p}^w$ the coordinates of $\boldsymbol{p}$ in the world system. What is $\boldsymbol{p}^c$?



You can find formulas on how to determine C_w^c in the script. One way is to use ***Euler angles***. This is not important for the moment, just assume that you are given C_w^c.

It is $\boldsymbol{p}^c = C_w^c\, \boldsymbol{p}^w$ (or $\boldsymbol{p}^w = C_w^{c\,\prime} \boldsymbol{p}^c = C_c^w \boldsymbol{p}^c$).

Rotation matrices are orthogonal: The ***inverse*** C_c^w=${C_w^c}^{-1}$ is simply the ***transpose*** C_c^w=${C_w^c}'$.

Let $T_w^c = \begin{pmatrix} C_w^c & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^c = C_w^c \boldsymbol{p}^w$ as $\boxed{\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w\,.}$

Let $T_c^w = \begin{pmatrix} C_c^w & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^w = C_c^w \boldsymbol{p}^c$ as $\boxed{\tilde{\boldsymbol{p}}^w = T_c^w \tilde{\boldsymbol{p}}^c\,.}$

check it!

Homogeneous coordinates - Assignment

A tree is located at world coordinates (10, -5, 100). A camera takes a picture at the origin of the world system, the camera is rotated around the x-axis by 45° (it is looking half-upwards). The according rotation matrix from *w* to *c* is

$$C_w^c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos 45° & \sin 45° \\ 0 & -\sin 45° & \cos 45° \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.707 & 0.707 \\ 0 & -0.707 & 0.707 \end{pmatrix}$$

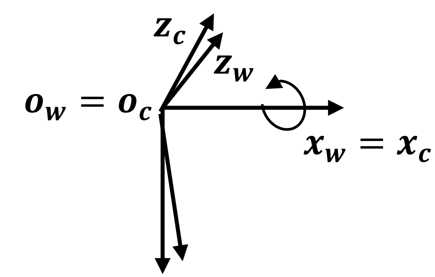
What are the inhomogeneous coordinates of the tree in the camera frame? Use homogeneous coordinates.

$$\boldsymbol{p}^w = \begin{pmatrix} 10 \\ -5 \\ 100 \end{pmatrix} \qquad \tilde{\boldsymbol{p}}^w = \begin{pmatrix} 10 \\ -5 \\ 100 \\ 1 \end{pmatrix}$$

$$T_w^c = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.707 & 0.707 & 0 \\ 0 & -0.707 & 0.707 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w = \begin{pmatrix} 10 \\ 67.2 \\ 74.2 \\ 1 \end{pmatrix}$$

The tree is located at (10, 67.2, 74.2) in the camera frame.

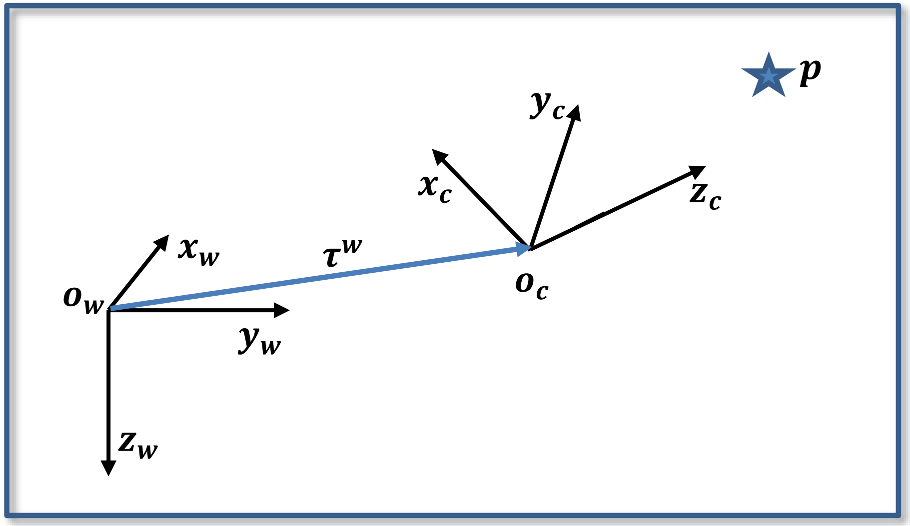


Let $T_w^c = \begin{pmatrix} C_w^c & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{pmatrix}$. Then, $\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w$

Homogeneous coordinate transformation
The general case

Affine transformation

A translation and a rotation can be combined.



It is $\boldsymbol{p}^w = C_c^w \boldsymbol{p}^c + \boldsymbol{\tau}^w$ or $\boldsymbol{p}^c = C_w^c \boldsymbol{p}^w - C_w^c \boldsymbol{\tau}^w$.

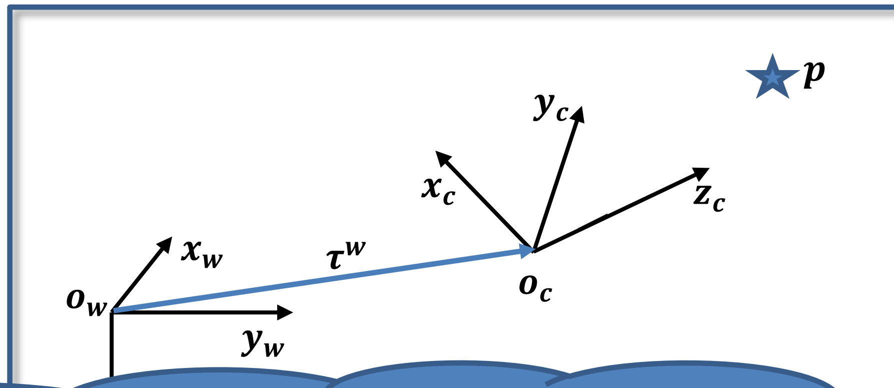
Let $T_w^c = \begin{pmatrix} C_w^c & -C_w^c \boldsymbol{\tau}^w \\ \mathbf{0} & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^c = C_w^c \boldsymbol{p}^w - C_w^c \boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w.}$

Let $T_c^w = \begin{pmatrix} C_c^w & \boldsymbol{\tau}^w \\ \mathbf{0} & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^w = C_c^w \boldsymbol{p}^c + \boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^w = T_c^w \tilde{\boldsymbol{p}}^c.}$

Homogeneous coordinates

Affine transformation

A translation and a rotation can be combined.



These two formulas are again easy to remember since they are essentially the same: The general form of T is $\begin{pmatrix} C & \tau \\ 0 & 1 \end{pmatrix}$, τ is the translation vector ***as seen in the target system*** (which is $-C\tau^w$)

Let $T_w^c = \begin{pmatrix} C_w^c & -C_w^c\boldsymbol{\tau}^w \\ \mathbf{0} & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^c = C_w^c\boldsymbol{p}^w - C_w^c\boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^c = T_w^c\tilde{\boldsymbol{p}}^w.}$

Let $T_c^w = \begin{pmatrix} C_c^w & \boldsymbol{\tau}^w \\ \mathbf{0} & 1 \end{pmatrix}$. Then, we can "simply" write $\boldsymbol{p}^w = C_c^w\boldsymbol{p}^c + \boldsymbol{\tau}^w$ as $\boxed{\tilde{\boldsymbol{p}}^w = T_c^w\tilde{\boldsymbol{p}}^c.}$

Homogeneous coordinates - Assignment

A tree is located at world coordinates (10, -5, 100). A camera takes a picture at coordinates (20, 0, 10). The camera is rotated around the x-axis by 45° (it is looking half-upwards). The according rotation matrix from *w* to *c* is

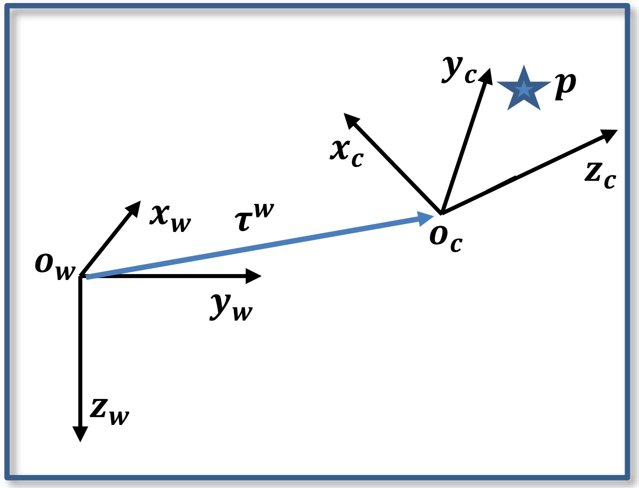
$$C_w^c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos 45° & \sin 45° \\ 0 & -\sin 45° & \cos 45° \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.707 & 0.707 \\ 0 & -0.707 & 0.707 \end{pmatrix}$$

What are the inhomogeneous coordinates of the tree in the camera frame? Use homogeneous coordinates.

$\boldsymbol{p}^w =$ $\qquad$ $\tilde{\boldsymbol{p}}^w =$ $\qquad$ $-C_w^c \boldsymbol{\tau}^w =$

$T_w^c =$

$\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w =$



Let $T_w^c = \begin{pmatrix} C_w^c & -C_w^c \boldsymbol{\tau}^w \\ \boldsymbol{0} & 1 \end{pmatrix}$.

Then, $\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w$

Homogeneous coordinates - Assignment

A tree is located at world coordinates (10, -5, 100). A camera takes a picture at coordinates (20, 0, 10). The camera is rotated around the x-axis by 45° (it is looking half-upwards). The according rotation matrix from *w* to *c* is

$$C_w^c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos 45° & \sin 45° \\ 0 & -\sin 45° & \cos 45° \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.707 & 0.707 \\ 0 & -0.707 & 0.707 \end{pmatrix}$$

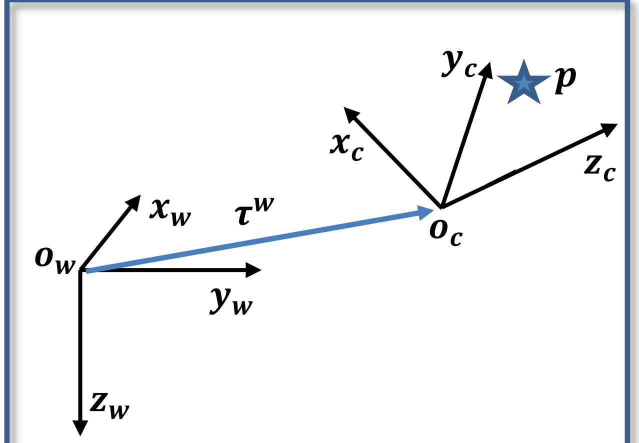
What are the inhomogeneous coordinates of the tree in the camera frame? Use homogeneous coordinates.

$$\boldsymbol{p}^w = \begin{pmatrix} 10 \\ -5 \\ 100 \end{pmatrix} \qquad \tilde{\boldsymbol{p}}^w = \begin{pmatrix} 10 \\ -5 \\ 100 \\ 1 \end{pmatrix} \qquad -C_w^c \boldsymbol{\tau}^w = \begin{pmatrix} -20 \\ -7.1 \\ -7.1 \end{pmatrix}$$

$$T_w^c = \begin{pmatrix} 1 & 0 & 0 & -20 \\ 0 & 0.707 & 0.707 & -7.1 \\ 0 & -0.707 & 0.707 & -7.1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w = \begin{pmatrix} -10 \\ 60.1 \\ 67.2 \\ 1 \end{pmatrix}$$

The inhomogeneous coordinates of the tree in the camera frame are (-10, 60.1, 67.2).

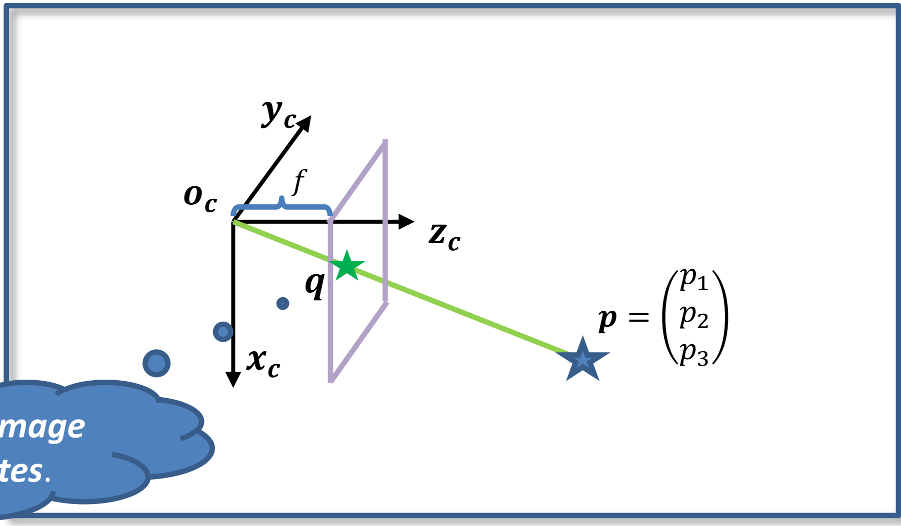


Let $T_w^c = \begin{pmatrix} C_w^c & -C_w^c \boldsymbol{\tau}^w \\ 0 & 1 \end{pmatrix}$.

Then, $\tilde{\boldsymbol{p}}^c = T_w^c \tilde{\boldsymbol{p}}^w$

We can now transform every object point into camera coordinates. According to the pinhole camera model, the image is just a ***central projection***.

Central projection



$\boldsymbol{q}$ is called ***image coordinates***.

A central projection Z on a plane at distance f in direction of the z axis.

It is $\boldsymbol{q} = \begin{pmatrix} f\frac{p_1}{p_3} \\ f\frac{p_2}{p_3} \end{pmatrix}$.

A central projection maps from $\wp^3$ to $\wp^2$.

Let $Z = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$. Then, we can "simply" write the central projection as $\boxed{\tilde{\mathbf{q}} = Z\tilde{\boldsymbol{p}}^c}$.

Homogeneous coordinates - Assignment

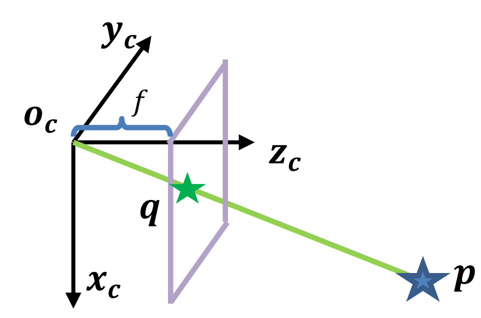
A tree is located at camera coordinates (-10, 20, 100), the focal length is $f = 0.5$. What are the inhomogeneous image coordinates of the central projection of the tree? Use homogeneous coordinates.

$\boldsymbol{p}^c =$ $\qquad$ $\tilde{\boldsymbol{p}}^c =$

$Z =$

$\tilde{\boldsymbol{q}} = Z\tilde{\boldsymbol{p}}^c =$

$\boldsymbol{q} =$



Let $Z = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.

Then, $\tilde{\mathbf{q}} = Z\widetilde{\boldsymbol{p}^c}$

Homogeneous coordinates - Assignment

A tree is located at camera coordinates (-10, 20, 100), the focal length is $f = 0.5$. What are the inhomogeneous image coordinates of the central projection of the tree? Use homogeneous coordinates.

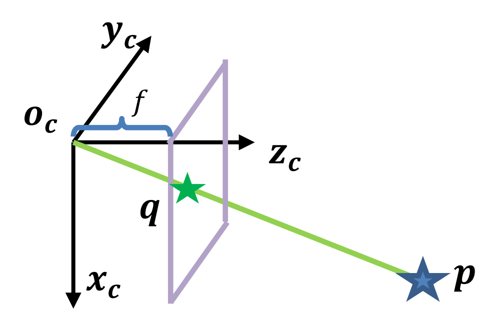
$$\boldsymbol{p}^c = \begin{pmatrix} -10 \\ 20 \\ 100 \end{pmatrix} \qquad \tilde{\boldsymbol{p}}^c = \begin{pmatrix} -10 \\ 20 \\ 100 \\ 1 \end{pmatrix}$$

$$Z = \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$\tilde{\boldsymbol{q}} = Z\tilde{\boldsymbol{p}}^c = \begin{pmatrix} -5 \\ 10 \\ 100 \end{pmatrix}$. For **dehomogenization**, divide by 100: $\tilde{\boldsymbol{q}} = \begin{pmatrix} -0.05 \\ 0.1 \\ 1 \end{pmatrix}$

$$\boldsymbol{q} = \begin{pmatrix} -0.05 \\ 0.1 \end{pmatrix}$$

The (inhomogeneous) image coordinates are (-0.05,0.1).

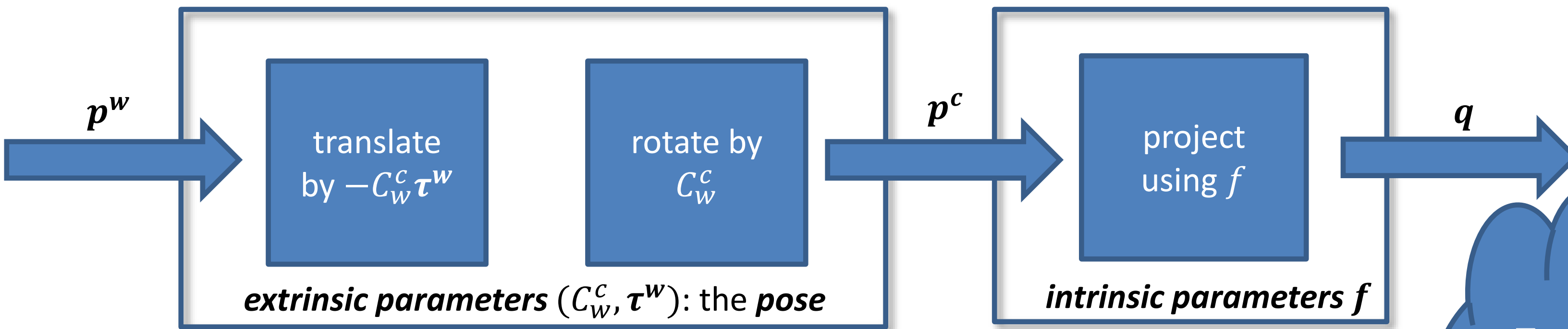


Let $Z = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.

Then, $\tilde{\mathbf{q}} = Z\widetilde{\boldsymbol{p}^c}$

Homogeneous coordinates – application to cameras

Let $\boldsymbol{p}^w \in \mathbb{R}^3$ be an object point (in world coordinates). We can separate the process of obtaining the image coordinates $\boldsymbol{q} \in \mathbb{R}^2$ into three steps:



$$\tilde{\boldsymbol{q}} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} C_w^c & -C_w^c\boldsymbol{\tau}^w \\ 0 & 1 \end{pmatrix} \widetilde{\boldsymbol{p}^w} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} (C_w^c|-C_w^c\boldsymbol{\tau}^w)\,\tilde{\boldsymbol{p}} = K C_w^c\,(\boldsymbol{I}|-\boldsymbol{\tau}^w)\,\widetilde{\boldsymbol{p}^w}$$

with the ***intrinsic parameter matrix*** $K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}$

and the ***extrinsic parameter matrix*** $C_w^c(\boldsymbol{I}|-\boldsymbol{\tau}^w)$.

We end up with $\boxed{\tilde{\boldsymbol{q}} = P\widetilde{\boldsymbol{p}^w}}$ *and the* ***camera matrix*** $\boxed{P = K C_w^c(\boldsymbol{I}|-\boldsymbol{\tau}^w)}$.

The vertical bar | is just used for better readability. The matrix is $(\boldsymbol{I}|-\boldsymbol{\tau}^w) = \begin{pmatrix} 1 & 0 & 0 & -\tau_1^w \\ 0 & 1 & 0 & -\tau_2^w \\ 0 & 0 & 1 & -\tau_3^w \end{pmatrix}$

For easier notation, we allow to factor out a matrix, $C(\boldsymbol{I}|-\boldsymbol{\tau}^w) := (C|-C\boldsymbol{\tau}^w)$

Summary: The camera matrix

Here comes all about homogeneous coordinates and transforms together:

Let $\boldsymbol{p} \in \mathbb{R}^3$ be an object point (in world coordinates) and $\boldsymbol{q} \in \mathbb{R}^2$ its image coordinates. The whole imaging process (for a pinhole camera) is defined by

$$\tilde{\boldsymbol{q}} = P\tilde{\boldsymbol{p}}$$

with the camera matrix P,

$$P = KC_w^c(\boldsymbol{I}|-\boldsymbol{\tau}^w) \quad .$$

- The ***intrinsic parameter matrix*** $K \in \mathbb{R}^{3\times 3}$ describes the optical characteristics of the camera (in our case, the focal length, there might be other parameters as well).
- The ***extrinsic parameter matrix*** $C_w^c(\boldsymbol{I}|-\boldsymbol{\tau}^w) \in \mathbb{R}^{3\times 4}$ describes the camera pose (the position and orientation of the camera) and does the projection onto the image plane.

Do you see an efficient way to obtain the image coordinates of thousands of 3D-points (as, e. g. needed for computer games)?

Homogeneous coordinates – application to cameras

Some additional remarks:

- We will sometimes need the ***reverse transformation*** $T_c^w = T_w^{c\,-1}$ from the camera to the world frame with $\widetilde{\boldsymbol{p}}^w = T_c^w \widetilde{\boldsymbol{p}}^c$. It holds

$$T_w^c = \begin{pmatrix} C_w^c & -C_w^c \boldsymbol{\tau}^w \\ 0 & 1 \end{pmatrix} \Rightarrow T_c^w = \begin{pmatrix} C_w^{c\,\prime} & \boldsymbol{\tau}^w \\ 0 & 1 \end{pmatrix}$$

- We can rearrange $\widetilde{\boldsymbol{q}} = K C_w^c (\boldsymbol{I}|-\boldsymbol{\tau}^w)\widetilde{\boldsymbol{p}}$ as $K^{-1}\widetilde{\boldsymbol{q}} = C_w^c(\boldsymbol{I}|-\boldsymbol{\tau}^w)\widetilde{\boldsymbol{p}}$. It is therefore possible to separate intrinsic and extrinsic parameters. $\widetilde{\boldsymbol{q}}_{norm} := K^{-1}\widetilde{\boldsymbol{q}}$ can be interpreted as ***normalized image coordinates***, since all intrinsic effects are removed.

- There are more intrinsic parameters, the parameters also may be given in different units since meters are a strange unit for pixel coordinates on a sensor chip;) See the script for details.

- The exact intrinsic parameters must be measured, data sheet information is not precise enough. The process of estimating the intrinsic parameters is called ***calibration.*** The calibration is often done by taking pictures of objects with known dimensions. See the script for details.

- Several transformations can be executed one after the other (the transformation matrices are multiplied from right to left, the first transformation is the right-most one). In this way, we can let a car move and the camera rotate while watching the car or even mount a camera on the moving car. This is how computer games work (more precisely, the renderer of OpenGL or DirectX).

Homogeneous coordinates – Assignment
Homework

What is the camera matrix of a camera with focal length 0.1 which is located 10 m North (z-axis) from the world frame origin and not rotated? How does the matrix change if the camera is turned 90° around the y-axis?

$K =$

$C_1 =$

$P_1 =$

$C_2 =$

$P_2 =$

$$P = KR(\boldsymbol{I}|-\boldsymbol{\tau}^w)$$

What is the camera matrix of a camera with focal length 0.1 which is located 10 m North (z-axis) from the world frame origin and not rotated? There is an object at 120 m East (x-axis), 110 m Down (y-axis), 210 m North (z-axis). At which point on the image plane is the object visible?

$P_1 =$

(as before)

$\tilde{\boldsymbol{p}}^w =$

$\tilde{\boldsymbol{q}} =$

$\boldsymbol{q} =$

$$P = KR(\boldsymbol{I}|-\boldsymbol{\tau}^w)$$

Homogeneous coordinates - Assignment

What is the camera matrix of a camera with focal length 0.1 which is located 10 units North (z-axis) from the world frame origin and not rotated? How does the matrix change if the camera is turned +90° around the y-axis?

$$K = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad C_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$P_1 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -10 \end{pmatrix} = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & -10 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -10 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -0.1 & 1 \\ 0 & 0.1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$P = K C_w^c (\boldsymbol{I} | -\boldsymbol{\tau}^w)$$

Homogeneous coordinates - Assignment

What is the camera matrix of a camera with focal length 0.1 which is located 10 m North (z-axis) from the world frame origin and not rotated? There is an object at 120 m East (x-axis), 110 m Down (y-axis), 210 m North (z-axis). At which point on the image plane is the object visible?

$$K = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad C_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$P_1 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -10 \end{pmatrix} = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & -10 \end{pmatrix}$$

$$\tilde{\boldsymbol{p}}^w = \begin{pmatrix} 120 \\ 110 \\ 210 \\ 1 \end{pmatrix}$$

$$\tilde{\boldsymbol{q}} = P_1 \tilde{\boldsymbol{p}}^w = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & -10 \end{pmatrix} \begin{pmatrix} 120 \\ 110 \\ 210 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 100 \end{pmatrix}$$

$$\boldsymbol{q} = \begin{pmatrix} 0.02 \\ 0.01 \end{pmatrix}$$

$$P = K C_w^c (\boldsymbol{I} | -\boldsymbol{\tau}^w)$$