



UniTs - University of Trieste

Faculty of Data Science and Artificial Intelligence
Department of mathematics informatics and geosciences

Computer Vision

Lecturer:
Prof. Felice Andrea Pellegrino

Author:
Christian Faccio

October 21, 2025

This document is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike \(CC BY-NC-SA\)](#) license. You may share and adapt this material, provided you give appropriate credit, do not use it for commercial purposes, and distribute your contributions under the same license.

Preface

As a student of Data Science and Artificial Intelligence, I've created these notes while attending the **Computer Vision** course.

The course provides a comprehensive introduction to the field of computer vision, covering both theoretical concepts and practical applications. The notes encompass a variety of topics, including:

- Fundamental principles of image processing and analysis.
- Techniques for feature detection and extraction.
- Methods for object recognition and classification.
- An overview of deep learning approaches in computer vision.
- Practical implementations using `OpenCV`.

While these notes were primarily created for my personal study, they may serve as a valuable resource for fellow students and professionals interested in computer vision.

Contents

1	Introduction	1
2	3D Vision	5
2.1	Image Formation	5
2.1.1	Lenses	8
2.2	Camera Calibration	13
3	Features and Primitives	14
3.1	Image Processing	14
3.1.1	Linear Filtering	14
3.1.2	Non-linear filters and other neighborhood operators	14
3.1.3	Image warping	14
3.1.4	Multi-resolution representations	14
4	Stereopsis	15
5	Support Vector Machines	16
6	Image Processing	17
7	Feature Detection	18
8	Fitting Geometric Primitives	19
9	Recognition	20
10	Deep Learning for Computer Vision	21

1

Introduction

”What does it mean to see?” The plain man’s answer (and Aristotele’s, too) would be, to know what is where by looking. In other words, vision is the process of discovering from images what is present in the world and where it is.

— David Marr

Definition: *Computer Vision*

Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding.

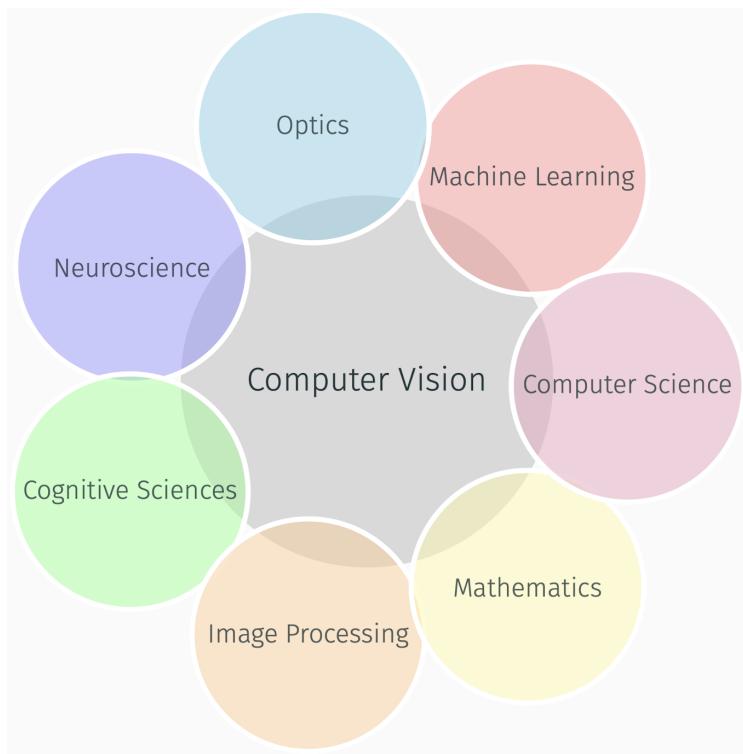


Figure 1.1: What is Computer Vision related to?

What can we extract from an image?

- **Semantic Information** → ”what”
- **Metric 3D Information** → ”where”

What is difficult to define, since images are sensed and therefore represented in computers as arrays of numbers. Such low-level representation is far from **semantics**. One of the goals of CV is to bridge the gap between pixels and "meaning".

Examples include 3D surface reconstruction using either calibrated or uncalibrated images (with or without a model of the camera), dense matching of images, motion analysis, and tracking of objects in image sequences.

What is correct depends on the specific tasks, since a major problem in computer vision is the ambiguity of the visual data. For example, a single 2D image can be generated by an infinite number of 3D scenes. Therefore, additional constraints are needed to solve the problem. There are different levels of interpretation in this field.

Where is difficult as well; we go from 3D (world) to 2D (image) usually, but in understanding what is in the image we infer from 2D to 3D and in this process we loose information.

- The forward problem is well-posed ($3D \rightarrow 2D$);
- The inverse problem is ill-posed ($2D \rightarrow 3D$).

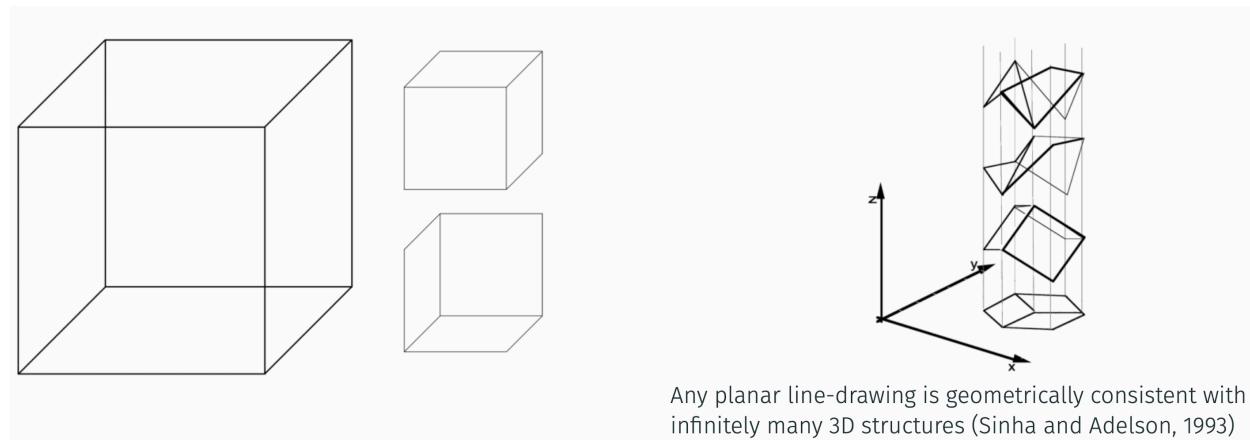


Figure 1.2: The forward and inverse problems.

Why people usually underestimates the difficulties of vision? Well, it is because we are pretty good at this task. There are special mechanisms in human vision that let us process low-level features such as size (length) and intensity in a non-trivial way.

- Human vision may inspire, but most CV deals with finding effective ways for solving specific problems;
- there is no "the" CV algorithm, but instead a **collection of algorithms/approaches** for tackling **specific problems in restricted domains**.

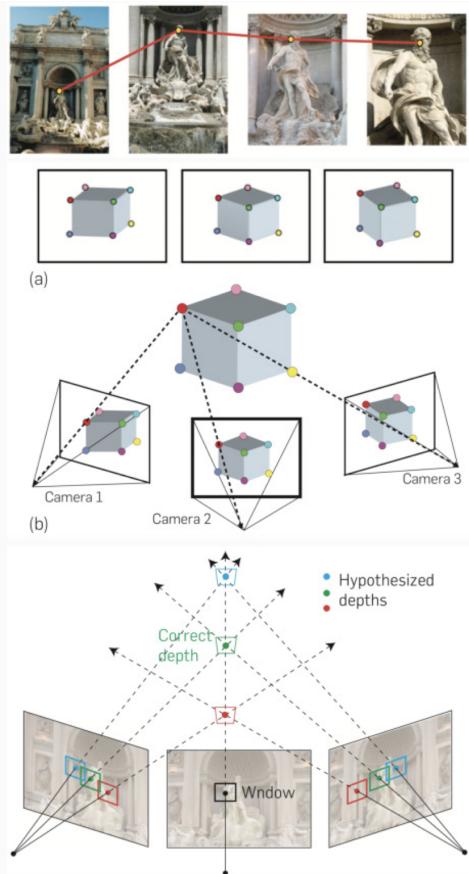
Two classes of problems come from the "where" and "what" tasks:

1. **Reconstruction** → recovering the 3D structure or a scene, given a sufficient amount of images of the object/scene;
2. **Recognition** → here we can find object detection (find all the regions in an image where a specific kind of object is likely to occur), instance recognition (recognize a known specific object potentially being viewed from a novel viewpoint) or category-level recognition (categorize images as belonging to a general class such as "cat" or "bicycle", among many possible classes).

Two noticeable achievements in CV

- Uncalibrated reconstruction (Agarwal et al., 2011)

- 496 processors
- 1984 GB of total memory
- 62 TB of disk space
- 460000 Flickr pictures of Rome, Venice and Dubrovnik
- ≈ 100 hrs of computation
- Output: detailed 3D geometry and colors of monuments in Rome, Venice and Dubrovnik



- Image categorization (Krizhevsky et al., 2012)

⌚ Observation: *Object's position in space*

ù To specify the position of an object in space we need 6 parameters:

- 3 for translation (x, y, z);
- 3 for rotation (roll, pitch, yaw).

Applications

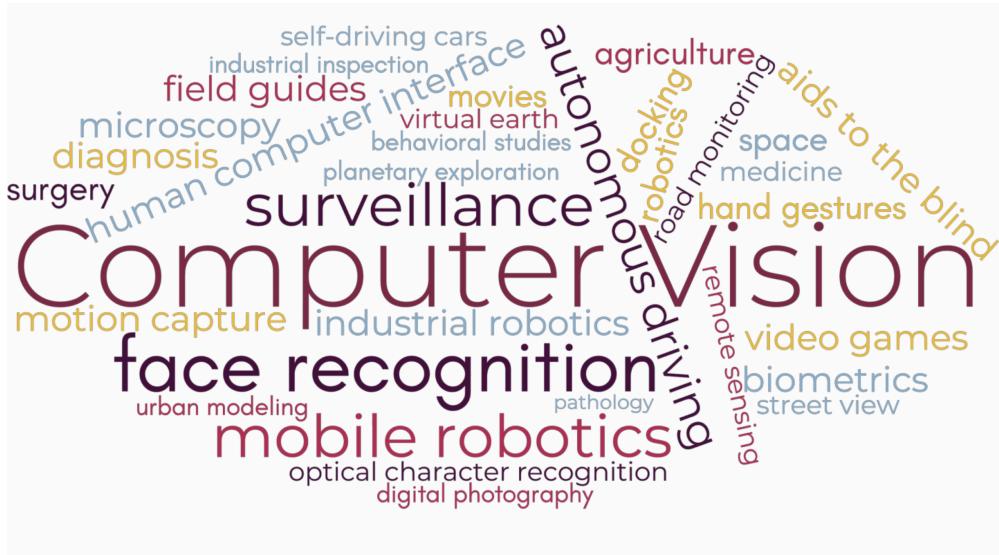


Figure 1.3: Some applications of Computer Vision.

About notation

In most of the cases:

- vectors are denoted by lowercase italic (e.g., v);
- scalars are denoted by mixed case italic (e.g., γ, A);
- matrices are denoted by uppercase italic (e.g., M);
- vectors operate as column vectors, i.e., they post-multiply matrices (e.g., Mv);
- transposition is denoted by T ;
- v_1 denotes a vector v_1 or the first component of vector v , or the first column of matrix V ;
- v_1^T denotes the first row of matrix V or the transpose of vector v_1 ;
- $P \succ 0$ means that the matrix P is positive definite;
- $P \succeq 0$ means that the matrix P is positive semi-definite;
- ∇f denotes the gradient of function f and is, by convention, a column vector;

2

3D Vision

2.1 Image Formation

The basic model is based on the principle of **camera obscura**, a room with a hole by which the light can enter the room.

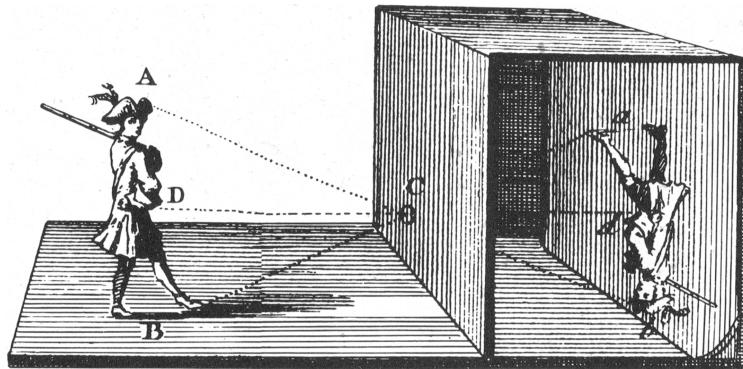


Figure 2.1: Camera Obscura principle.

A simple setting for creating images on a white piece of paper is by projecting a shadow on it and, in the middle of the shadow, appears a picture of the scene in front of it.

- Leonardo da Vinci (1452-1519) was the first to describe the camera obscura in his notebooks.
- Johann Zahn (1685-1771) designed the first portable camera obscura in 1685.
- Joseph Nicéphore Niépce (1765-1833) created the first permanent photograph in 1826 using a camera obscura.

From a geometrical point of view, the light rays of the object hit the film plane on different points, so we don't directly have the picture of the object. BUT, if we set a barrier in the middle, we have a one-to-one correspondence between the points of the object and the points of the film plane. That is why the pinhole camera works.

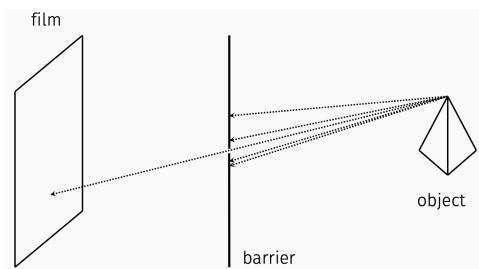
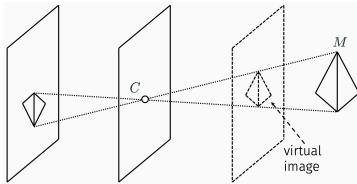


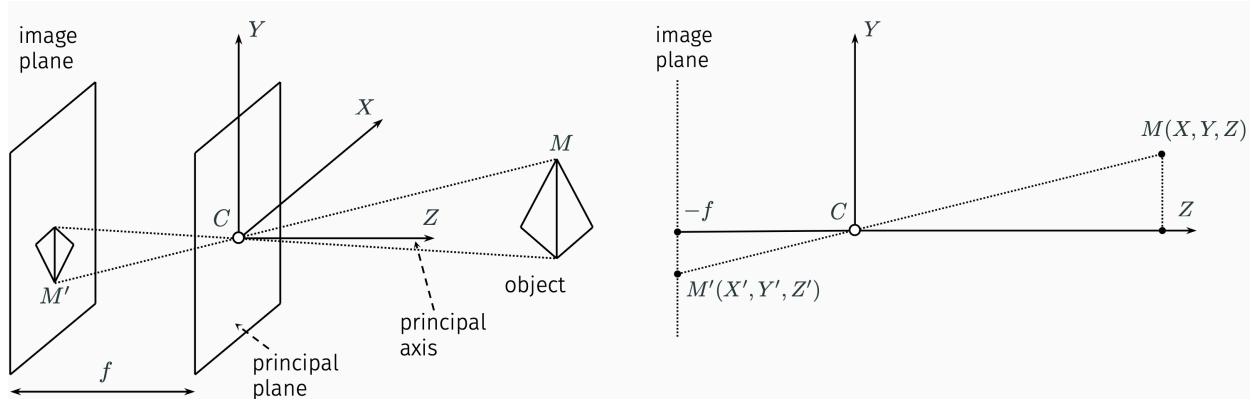
Figure 2.2: Pinhole camera principle.



- The image is reversed upside-down and left-right;
- A *virtual image* is formed in front of the camera.

Figure 2.3: Pinhole camera model.

Perspective Projection is composed by a *principal plane*, parallel to the image plane and that contains the *optical center* (C). We then create the three axes, with the Z one named *principal axis*.



Point M is projected onto the image plane at point M' . By similarity of triangles, it follows that the 3D point $[X, Y, Z]^T$ is mapped to the point

$$[X', Y', Z']^T = \left[-\frac{f}{Z}X, -\frac{f}{Z}Y, -f \right]^T$$

where $\frac{f}{Z}$ is the *perspective scale factor*. Farther away objects (larger Z) appear smaller).

Weak Perspective If the object is thin w.r.t. its distance from the camera, then the perspective scale factor is roughly constant:

$$\frac{f}{Z_0 + \Delta Z} \approx \frac{f}{Z_0}$$

Then, perspective projection can be approximated by a *scaled orthographic projection*

$$X' = -\frac{f}{Z_0}X, \quad Y' = -\frac{f}{Z_0}Y$$

Tip:

Rule of thumb due to Leonardo da Vinci: $\frac{\Delta Z}{Z_0} < \frac{1}{10}$

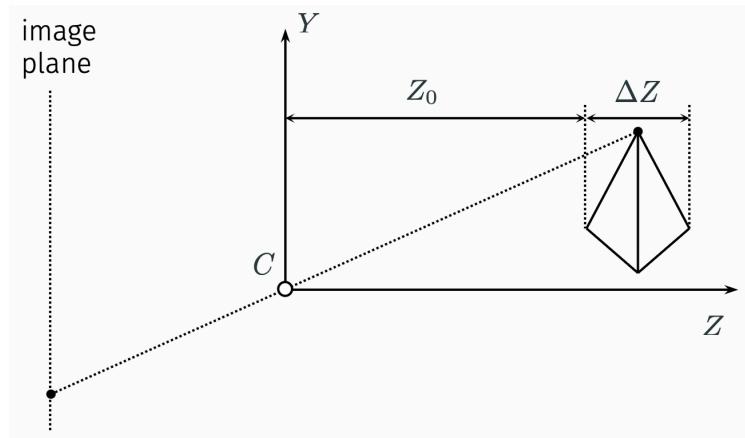


Figure 2.4: Weak perspective projection.

The **Aperture** is related to the amount of light that enters the camera. A larger aperture (smaller *f*-number) allows more light to enter, resulting in a brighter image. However, a larger aperture also reduces the depth of field, which is the range of distances within which objects appear sharp in the image.

- **Pinhole too big** → many directions are averaged, blurring the image; sharp but dark image, because little light reaches the sensor;
- **Pinhole too small** → diffraction effects blur the image; bright but blurred image, because many directions are averaged.

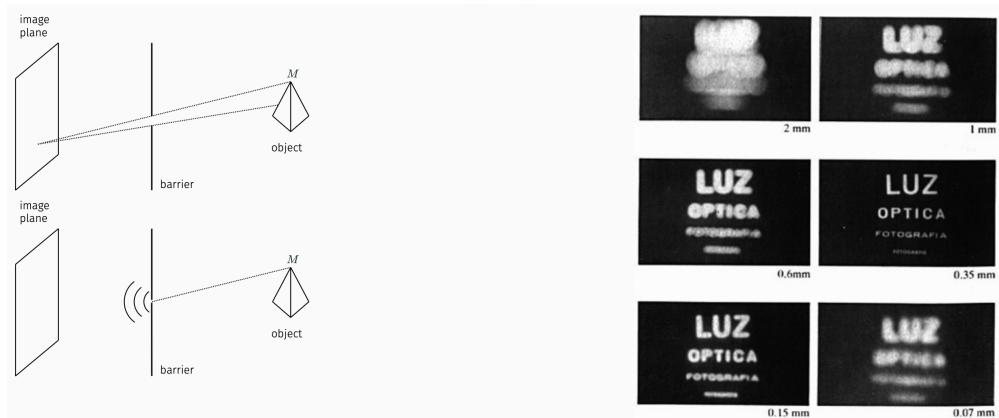


Figure 2.5: Aperture effects.

Solution? Lenses!

2.1.1 Lenses

A lens collects rays departing from the same points and focuses them onto the screen. There is a specific distance at which objects are in focus. Perspective projection is still valid within the thin lens assumption.

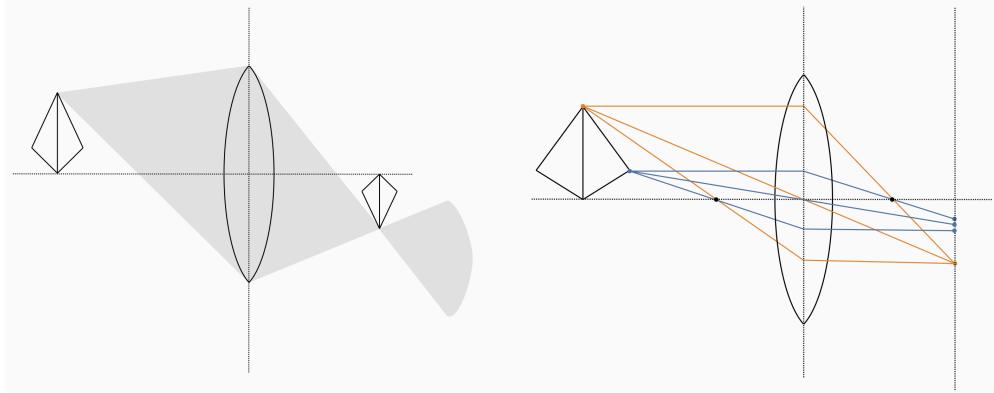
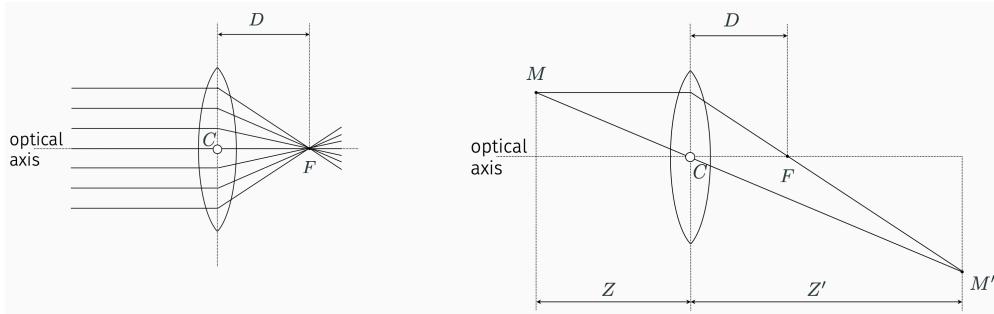


Figure 2.6: Lens principle. The gray areas represent the set of rays originated from the object.

- A **thin lens** is composed of a single piece of glass with very low, equal curvature on both sides;
- Any ray that enters parallel to the axis on one side of the lens proceeds towards the **focal point** F ;
- Any ray that passes through the center of the lens C does not change its direction;
- The distance D from the center to the focal point is called **focal length** f ;
- The image M' of M can be found by intersecting two rays.



Let's now try to use mathematical notation. Based on triangle similarity:

$$\frac{Y'}{Y} = \frac{Z'}{Z}, \quad \frac{Y'}{Y} = \frac{Z' - D}{D}$$

Thus, we get the **thin lens equation**, also known as the lensmaker's formula:

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{D}$$

which basically tells that points that are far away from the lens ($Z \rightarrow \infty$) are focused at the focal length ($Z' = D = f$). Point M is projected, when in focus, into the same position of a pinhole model having the optical center located in the lens center C .

- Two points lying on opposite sides of the lens at distances that satisfy the thin lens equation are **conjugate points**;
- In Figure 2.7, A and A' are conjugate;
- Parallel rays from infinity focus at distance D from the lens;
- As a source of light rays moves closer to the lens, they focus further away on the other side;
- Rays originating from a point at a distance D from the lens become parallel after passing through the lens, i.e., they focus at infinity;
- To focus on objects at different distances, move sensor relative to the lens;
- at $Z = Z' = 2D$ we have 1 : 1 imaging, since $\frac{1}{2D} + \frac{1}{2D} = \frac{1}{D}$;
- Can't focus on objects closer to the lens than its focal length D .

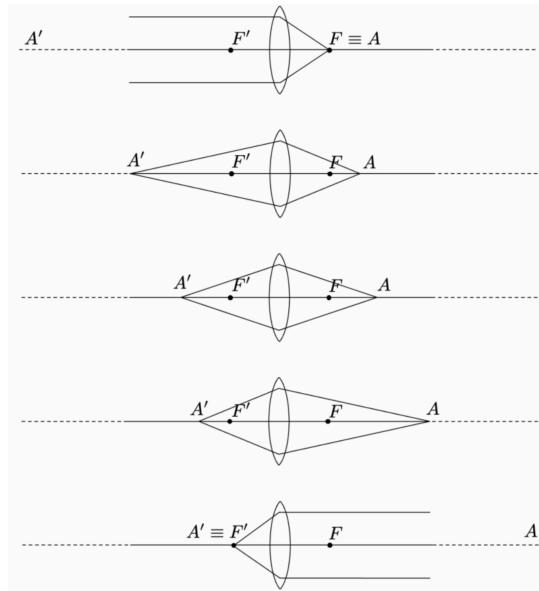


Figure 2.7: Conjugate points.

The **field of view** of a camera is the portion of the scene space that actually projects onto the sensor. It depends on the focusing distance and on the sensor width w . Conventionally it is defined when focusing at infinity:

$$f.o.v. = 2\phi = 2 \arctan \frac{w}{2D}$$

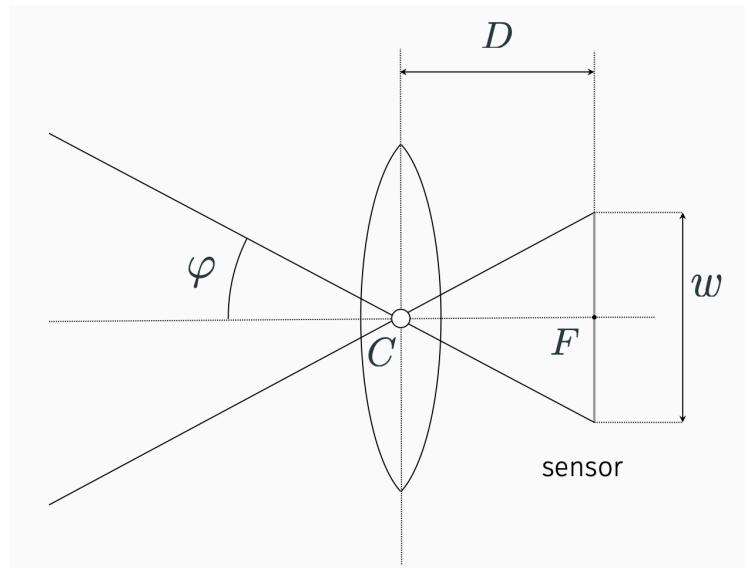


Figure 2.8: Field of view.

The **blur circle** is the circle of confusion created when a point is not in focus. It is formed by rays and its size depends on $\Delta Z'$ and on the lens diameter d :

$$\text{blur circle diameter} = d \frac{|\Delta Z'|}{Z'}$$

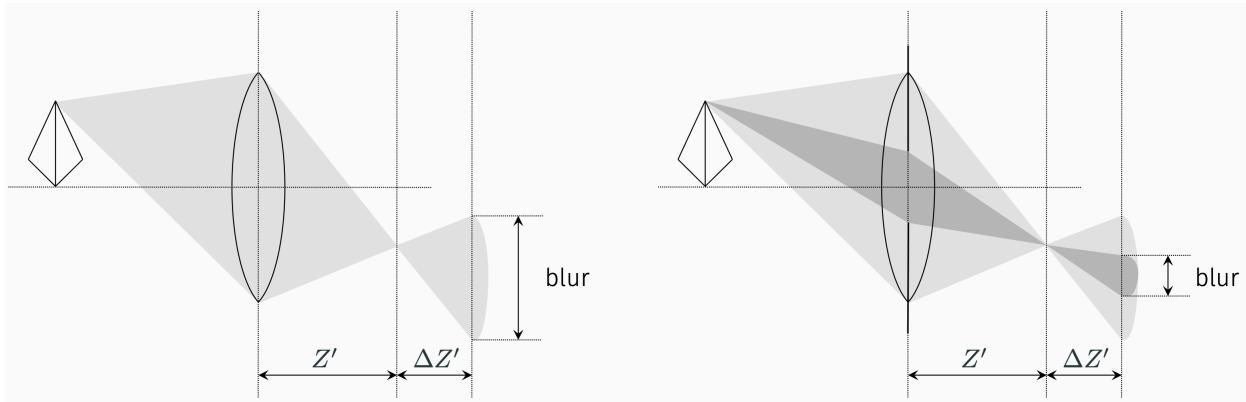


Figure 2.9: Blur circle formation.

The smaller the diameter, the smaller the blur circle, the sharper the image.

Once fixed the diameter of the blur circle, locate the farthest and the closest planes whose points are projected inside the blur circle: the distance between these planes is known as **depth of field**. Moreover, for a fixed diameter of the blur circle, the depth of field changes as the diameter of the lens (aperture) changes; in particular, the depth of field increases as the aperture decreases.

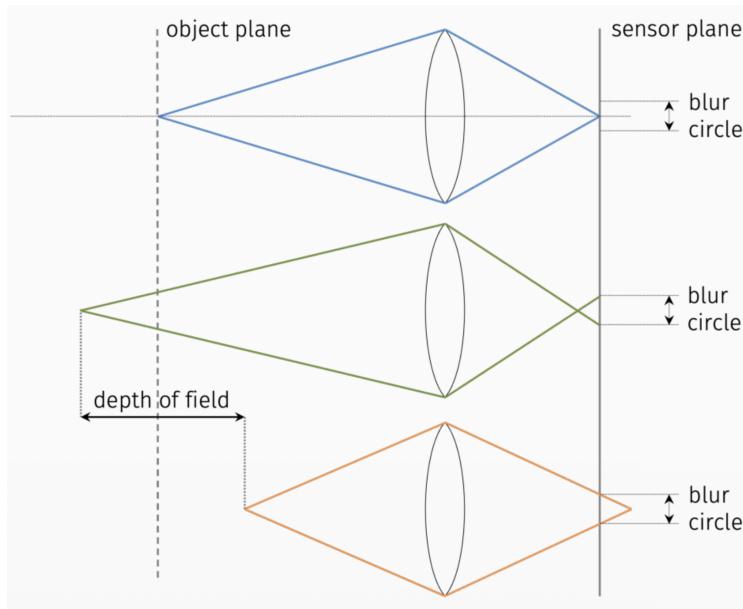


Figure 2.10: Depth of field.

⌚ Observation: Aperture size

The aperture size is usually expressed by the *f*-number, defined as

$$f\text{-number} = \frac{D}{d}$$

where *D* is the focal length and *d* is the diameter of the aperture. Smaller *f*-numbers correspond to larger apertures, allowing more light to enter the camera, while larger *f*-numbers correspond to smaller apertures, reducing the amount of light.

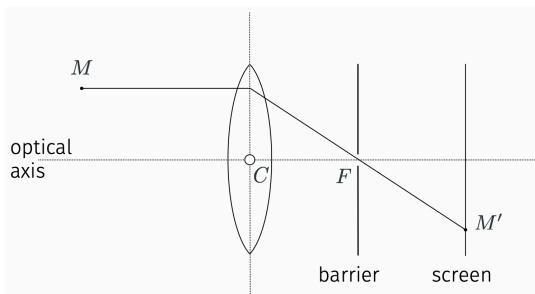


Figure 2.11: Lens aberrations.

Real (thick) lenses introduce several types of **aberrations** that cause deviations from the ideal pinhole camera model:

- **Radial distortion** causes straight lines to appear curved in the image, especially towards the edges. It can be corrected using calibration techniques.
 - **Barrel distortion** makes lines bulge outward from the center of the image.
 - **Pincushion distortion** makes lines pinch inward towards the center of the image.
- **Chromatic aberration** occurs because different wavelengths of light are focused at different points, leading to color fringing around high-contrast edges.
- **Spherical aberration** arises when light rays passing through the edges of a spherical lens focus at different points than those passing through the center, resulting in a blurred image.

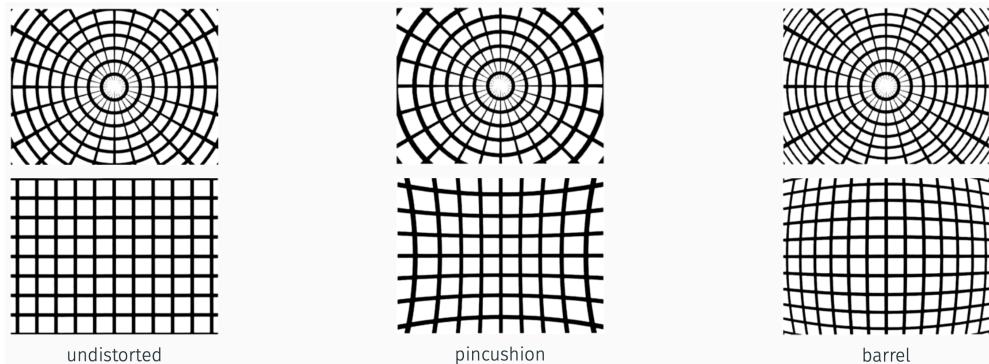


Figure 2.12: Examples of radial distortion

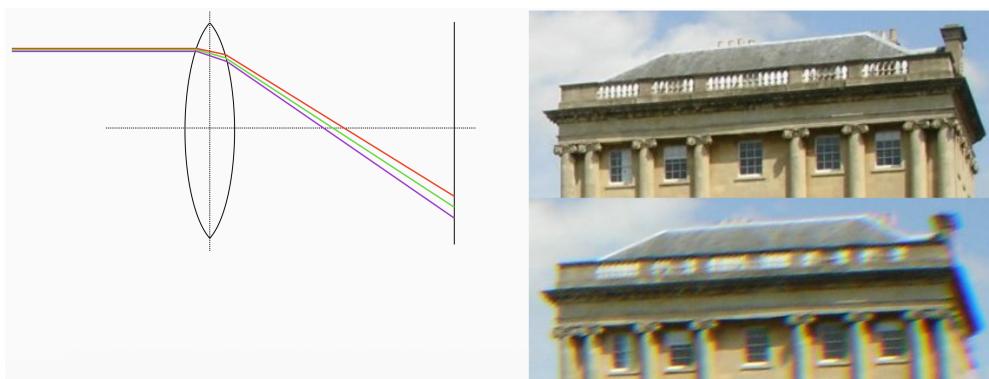


Figure 2.13: Example of chromatic aberration.

Now, if we put a barrier with a hole in correspondence of the focus F , the light beams from M are all blocked, except those parallel to the optical axis. The image of M does not depend on Z (depth), but only on X and Y and so the device performs an orthographic projection. This is a **Telecentric camera**.

Vignetting is the darkening of image corners due to lens limitations or improper lens hoods. It can be corrected in post-processing. It is caused by *compound lenses*, which may reduce aberrations but can introduce vignetting. In them, light beams emanating from object points located off-axis are partially blocked by individual lens components, resulting in a reduction of brightness toward the image periphery.

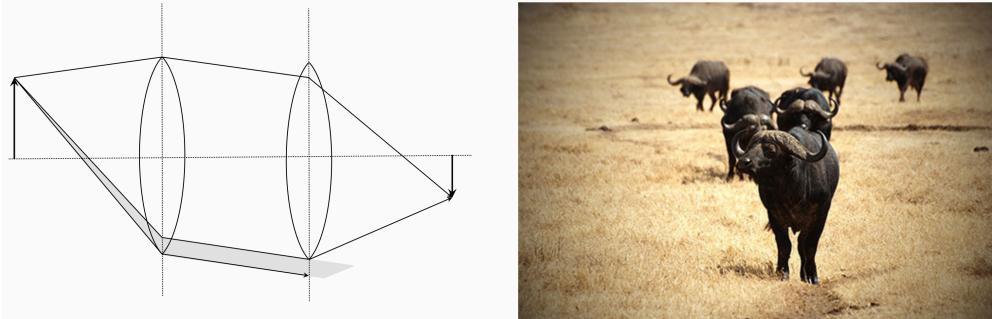


Figure 2.14: Vignetting effect.

Sensing, instead, is the process of converting the continuous image formed on the sensor into a discrete digital image. This involves sampling the continuous image at regular intervals (pixels) and quantizing the intensity values into discrete levels (bit depth). Since the sensor is a rectangular array of sensing elements (pixels), the image formed on the sensor is a rectangular portion of the image formed by the lens. The light charges a capacitor in each pixel, which is then read out and converted into a digital value. The longer the exposure time, the more light is collected, resulting in a brighter image. However, longer exposure times can also lead to motion blur if the camera or subject moves during the exposure. Finally, the capacitor is discharged periodically (the period is called *shutter time*).

A simple sensor model is composed by:

- an integrator that collects light over the exposure time T ;
- a sampler that samples the integrated light at each pixel location;
- a quantizer that converts the sampled light intensity into discrete digital values.

Moreover, there are two approaches for sensing color:

- the **3-sensor method**: a set of glass prisms uses a combination of internal reflection and refraction to split the incoming image into three, each reaching a different sensor;
- the **Bayer Mosaic**: half the pixels are sensitive to the green band of the light spectrum, while one quarter each to blue and red. This is consistent with the distribution of color-sensitive cones in the human retina. Color components that are not sensed (two per pixel), are inferred by interpolation from neighboring pixels.

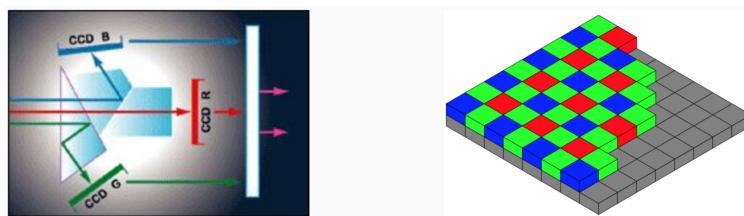


Figure 2.15: 3-sensor beam splitter on the left, Bayer mosaic on the right.

2.2 Camera Calibration

3

Features and Primitives

3.1 Image Processing

3.1.1 Linear Filtering

3.1.2 Non-linear filters and other neighborhood operators

An example of this is the **Median Filter**, which substitutes each pixel value with the *median* of the distribution of the neighborhood. Since the median is robust to outliers, median filtering helps in filtering out the **shot noise**, where linear filters, like Gaussian smoothing perform poorly. It is robust to outliers.



Figure 3.1

We can go on using different versions of the median filter, considering also mean filters:

- **α -trimmed mean filter**: it is a non-linear filter that averages together all the pixels except for the α fraction that are the smallest and the largest;
- **Weighted median filter**: corresponds to the median of a weighted distribution, where each pixel appears a number of times depending on its distance from the center of the neighborhood. It can be shown to be equivalent to the following optimization problem:

$$\min_{g(i,j)} \sum_{k,k} w(k,l) |f(i+k, j+l) - g(i,j)|^p \quad (3.1)$$

where $-W \leq k, l \leq W$ and $p = 1$. For $p = 2$ we have the weighted mean filter.

3.1.3 Image warping

3.1.4 Multi-resolution representations

4

Stereopsis

4.1

5

Support Vector Machines

Support
Vector
Machines

6

Image Processing

Image

7

Feature Detection

7.1

8

Fitting Geometric Primitives

Chapter 8

9

Recognition

Chapter 9

10

Deep Learning for Computer Vision

10.1

Bibliography

- [1] Andrea Fusiello. *Computer Vision: Three-Dimensional Reconstruction Techniques*. Springer, 2024.
- [2] Reinhard Klette. *Concise computer vision*. Vol. 233. Springer, 2014.
- [3] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.
- [4] Antonio Torralba et al. *Foundations of computer vision*. MIT Press, 2024.