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

September 26, 2025

github.com/christianfaccio christianfaccio@outlook.it

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

> "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.

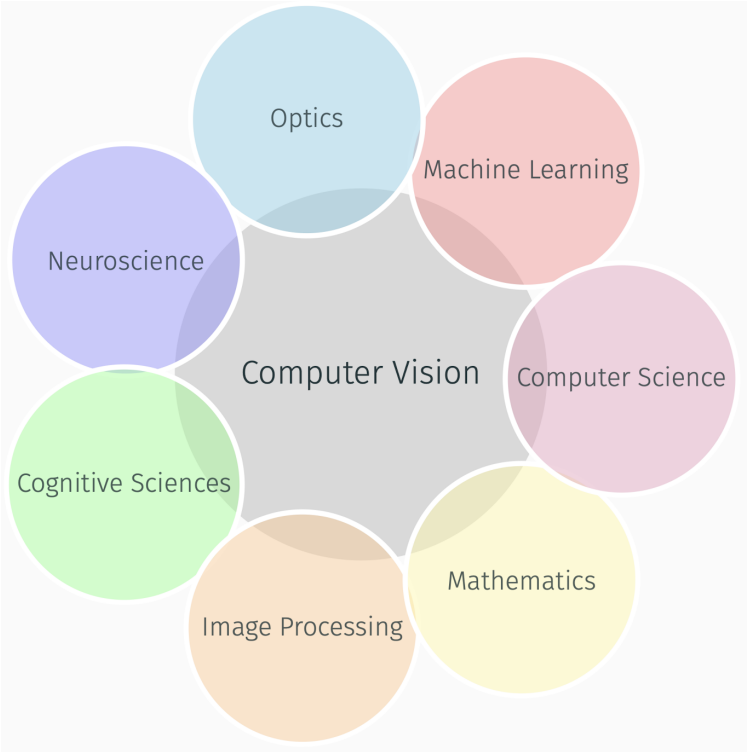


Figure 1.1: What is Computer Vision related to?

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).

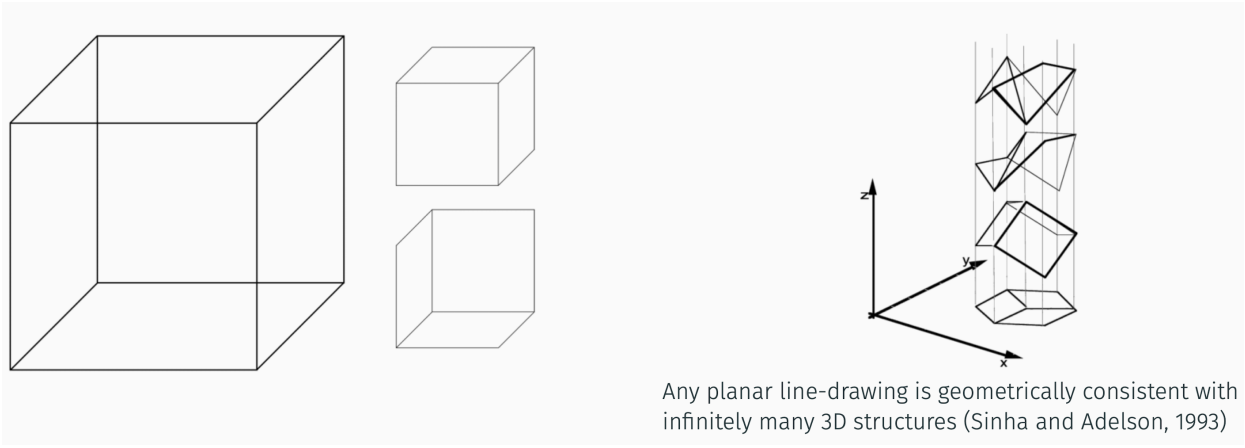


Figure 1.2: The forward and inverse problems.

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.

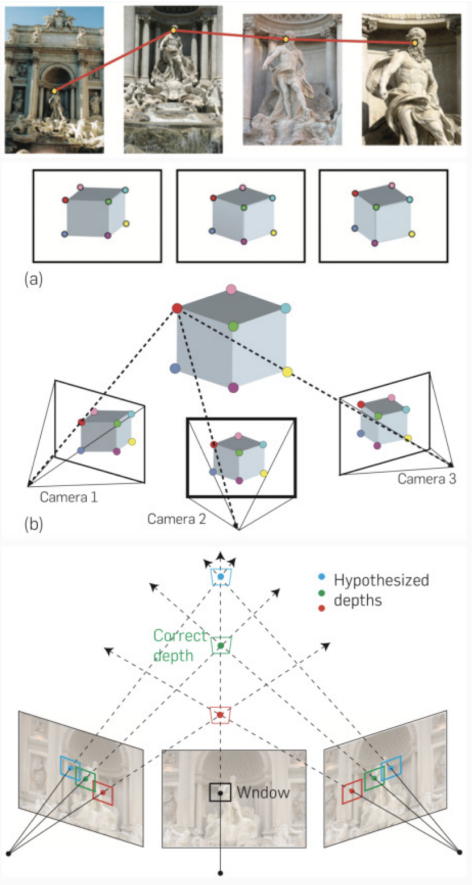
- Human vision may inspire, but most CV deals with finding effective ways for solving specific problems;
- there is no ”the” CV algorithm, but instead a **collection of algorithms/approaches** for tackling **specific problems** in **restricted domains**.

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

1. **Reconstruction** $\rightarrow$ recovering the 3D structure or a scene, given a sufficient amount of images of the object/scene;
2. **Recognition** $\rightarrow$ 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

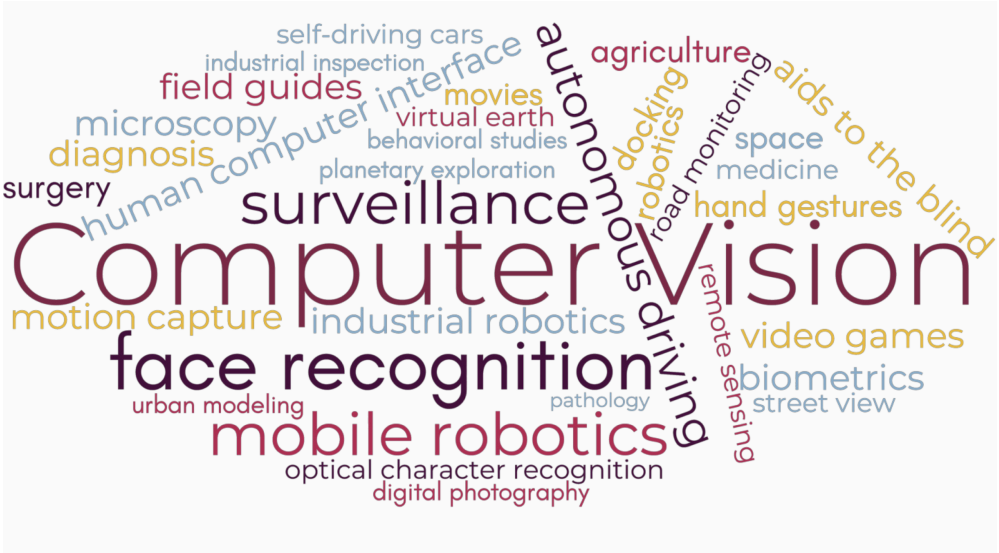


Figure 1.3: Some applications of Computer Vision.

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.

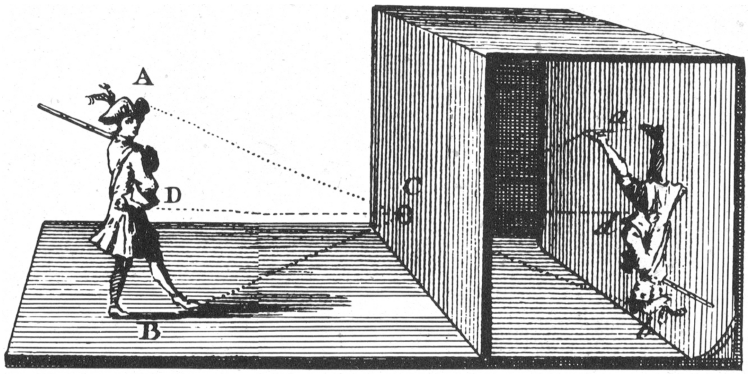


Figure 2.1: Camera Obscura principle.

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 Nicephore Niepce (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.

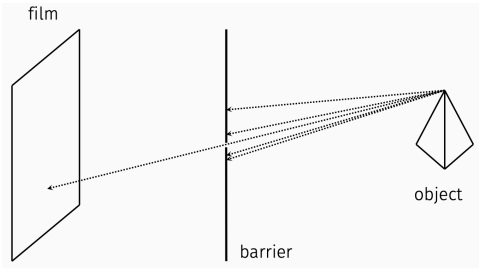


Figure 2.2: Pinhole camera principle.

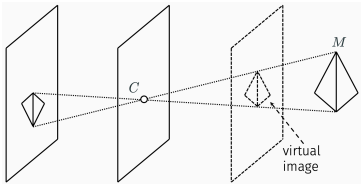
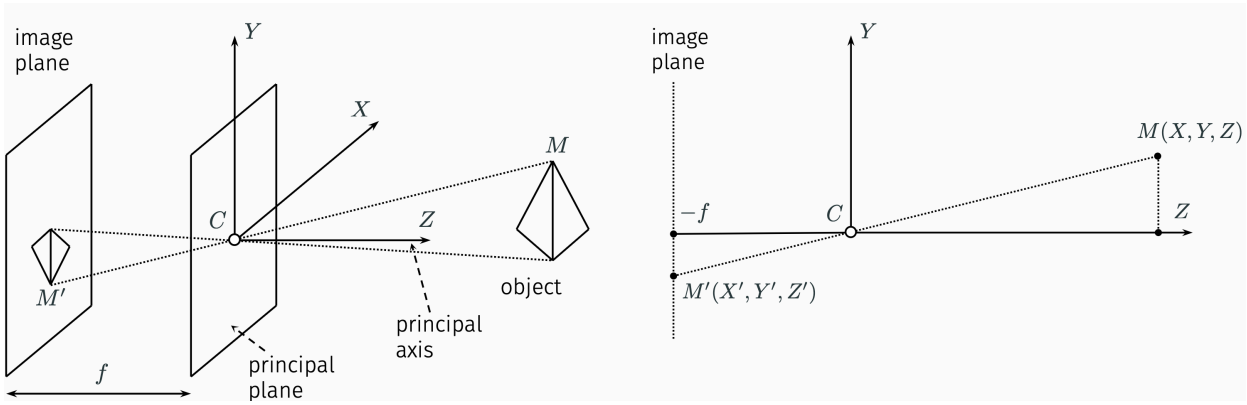


Figure 2.3: Pinhole camera model.

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

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 constand:

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

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: $\frac{\Delta Z}{Z_0} < \frac{1}{10}$

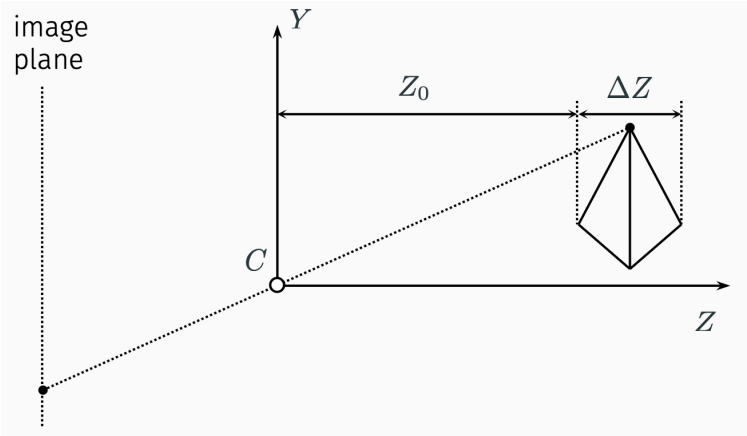


Figure 2.4: Weak perspective projection.

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** $\rightarrow$ many directions are averaged, blurring the image; sharp but dark image, because little light reaches the sensor;
- **Pinhole too small** $\rightarrow$ diffraction effects blur the image; brught 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.

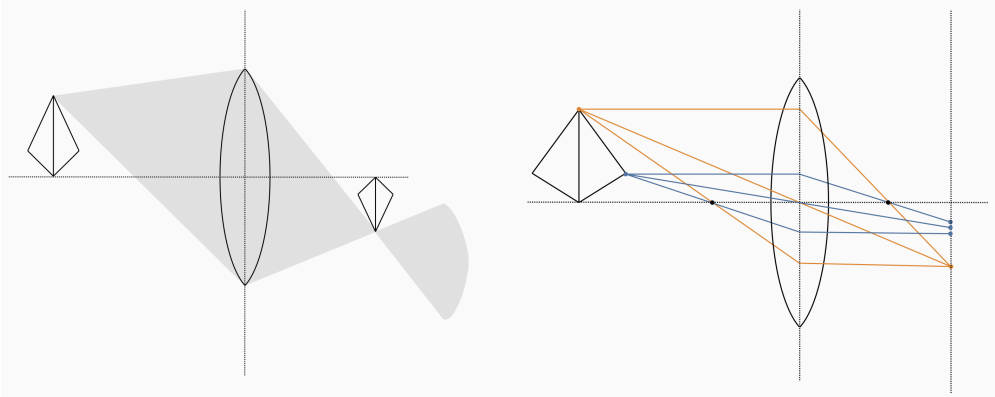
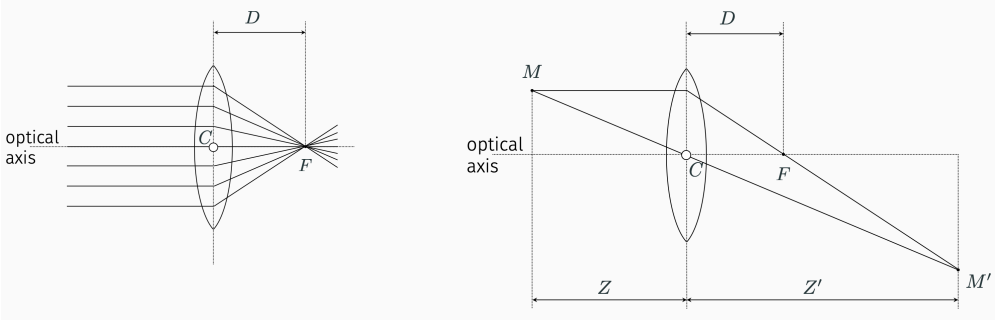


Figure 2.6: Lens principle. The gray areas represent the set of rays originated from the object.

- 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 ($Z \to \infty$) 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 throught the lens, i.e., they focus at infinity.

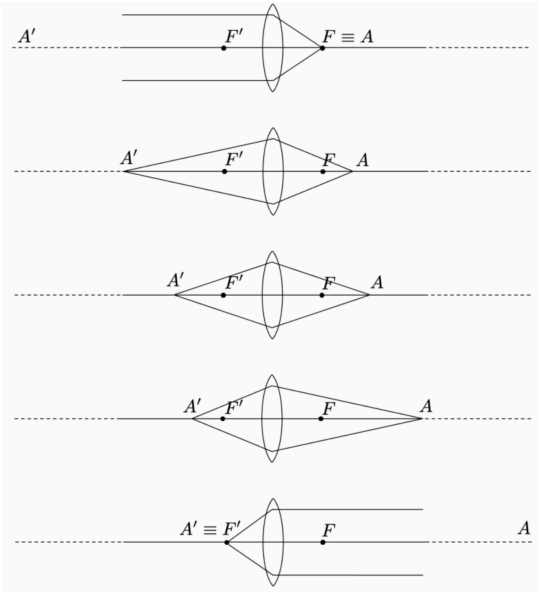


Figure 2.7: Conjugate points.

3
Camera Calibration

4
Stereopsis

5 Support Vector Machines

6
Image Processing

7
Feature Detection

8
Fitting Geometric Primitives

9
Recognition

10
Deep Learning for Computer Vision

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.