



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

December 10, 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 Model	13
2.2.1	Transformations	14
2.2.2	General Model	16
2.3	Camera Calibration	19
2.3.1	Direct Linear Transform	20
2.3.2	Zhang's method	24
2.3.3	Estimating homographies	27
2.4	Stereopsis	27
2.4.1	Triangulation	27
2.4.2	Epipolar Geometry	29
3	Features and Primitives	30
3.1	Image Processing	30
3.1.1	Linear Filtering	30
3.1.2	Non-linear filters and other neighborhood operators	30
3.1.3	Image warping	30
3.1.4	Multi-resolution representations	30

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.

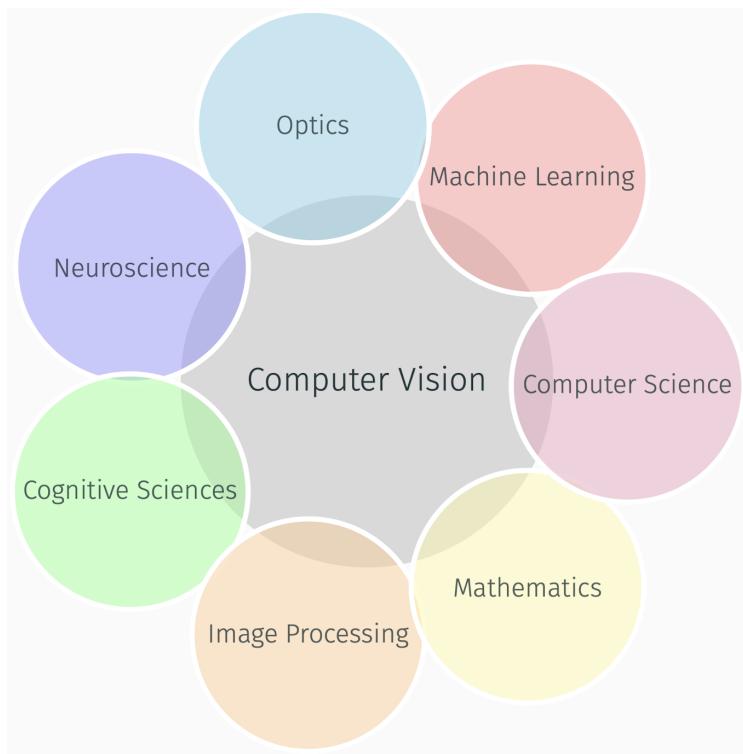


Figure 1.1: What is Computer Vision related to?

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 ($2D \rightarrow 3D$).

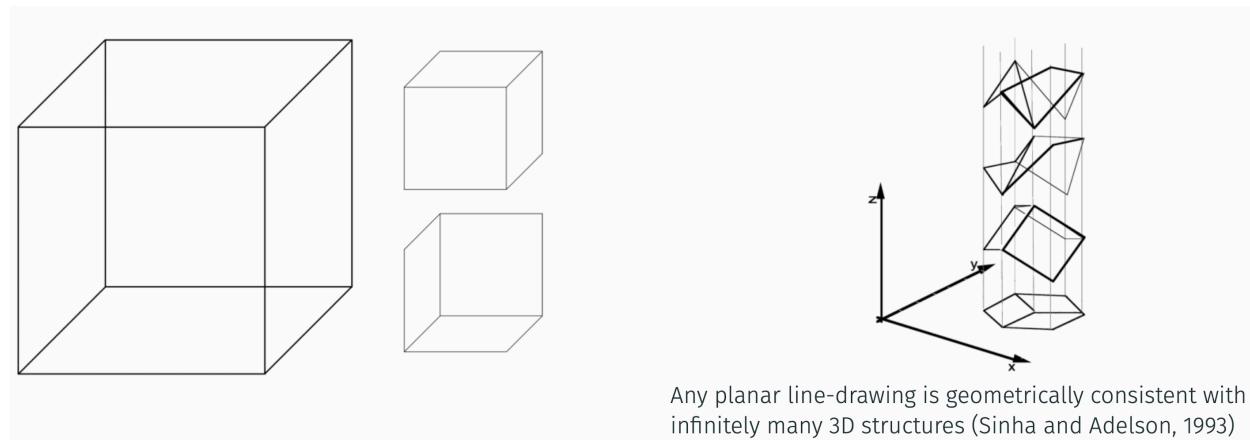


Figure 1.2: The forward and inverse problems.

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**.

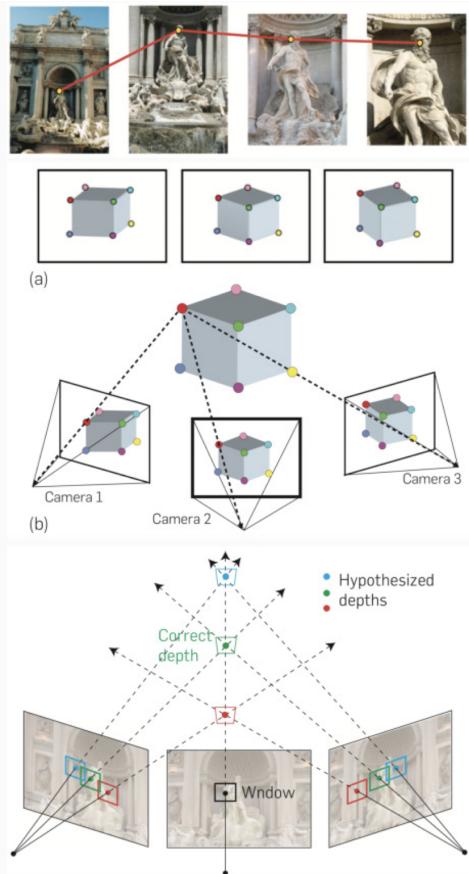
Two classes of problems come from the "where" and "what" tasks:

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

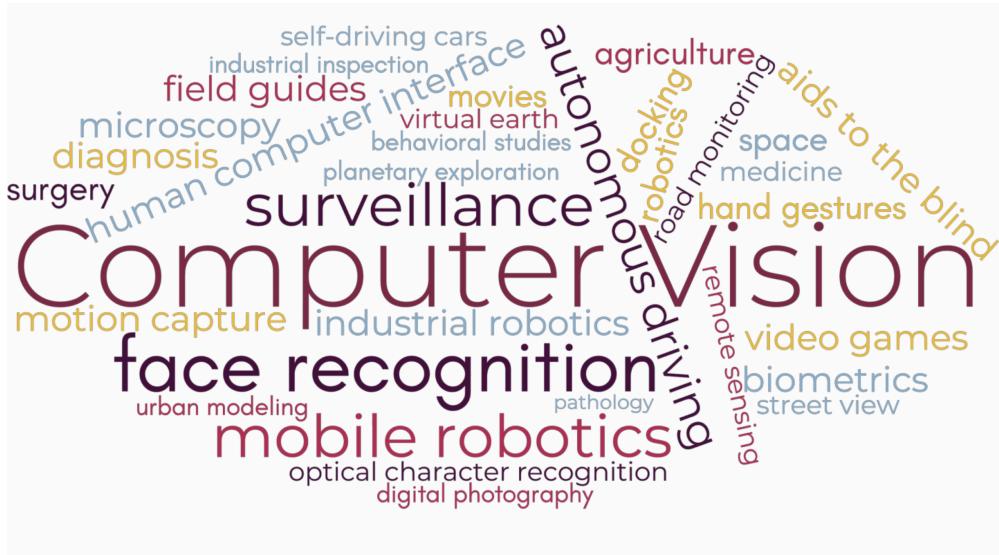


Figure 1.3: Some applications of Computer Vision.

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.

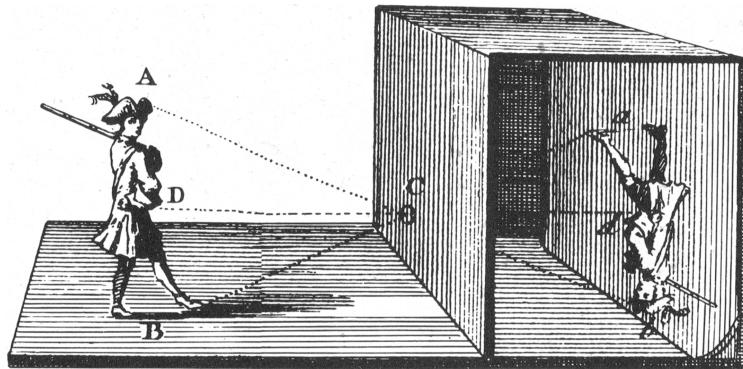


Figure 2.1: Camera Obscura principle.

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.

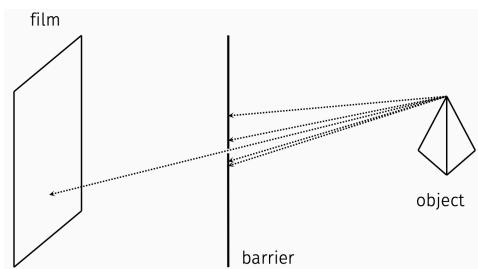
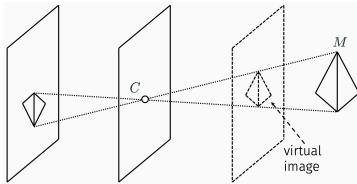


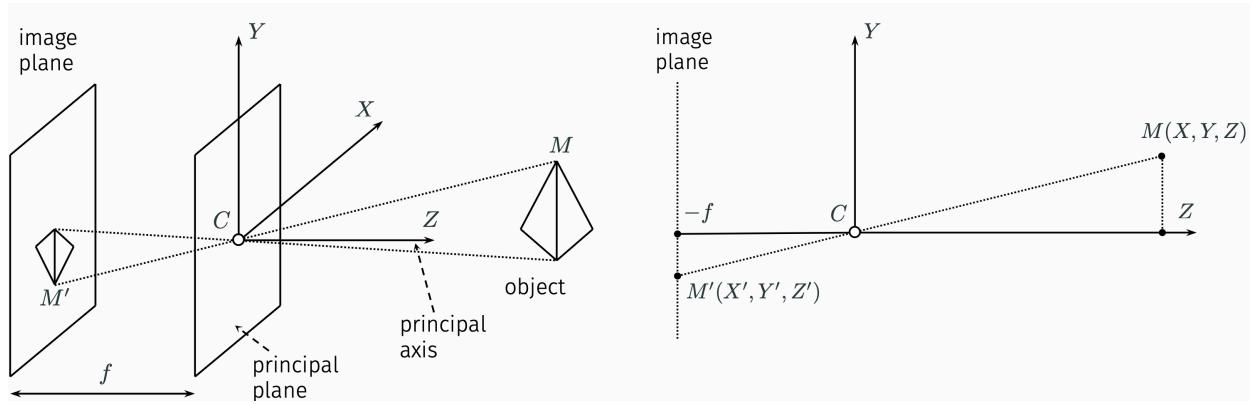
Figure 2.2: Pinhole camera principle.



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

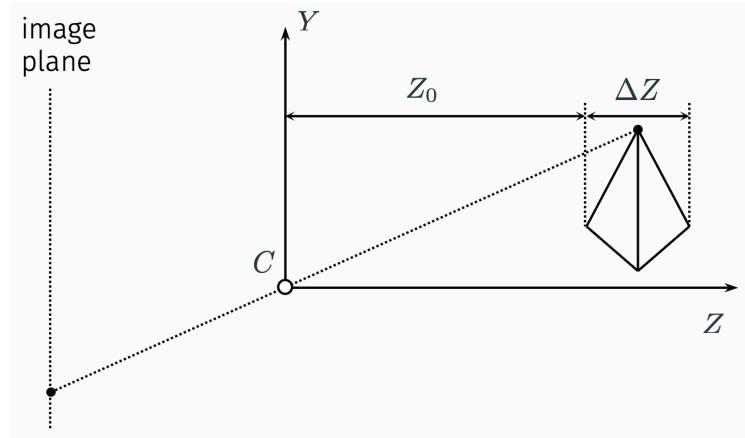


Figure 2.4: Weak perspective projection.

The **Aperture** is related to the amount of light that enters the camera. A larger aperture 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; bright but blurred image, because many directions are averaged;
- **Pinhole too small** → diffraction effects blur the image; sharp but dark image, because little light reaches the sensor.

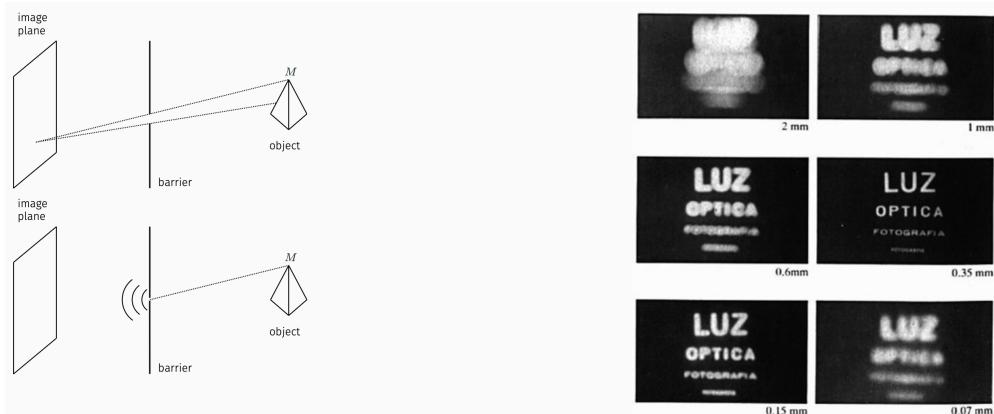


Figure 2.5: Aperture effects.

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.

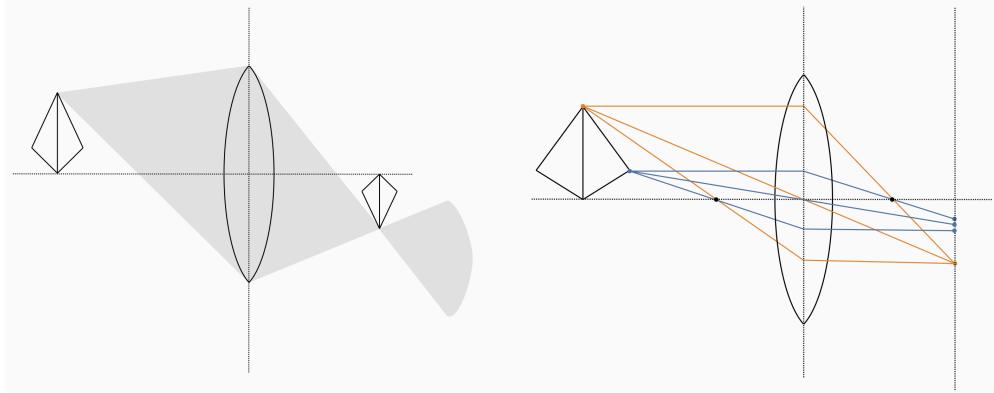
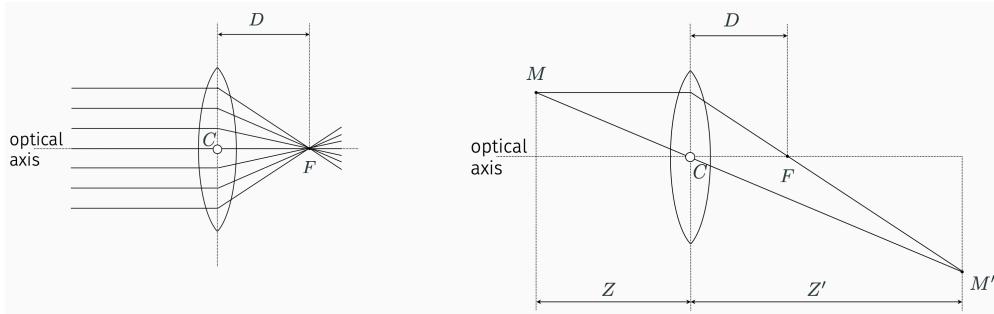


Figure 2.6: Lens principle. The gray areas represent the set of rays originated from the object.

- 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 .

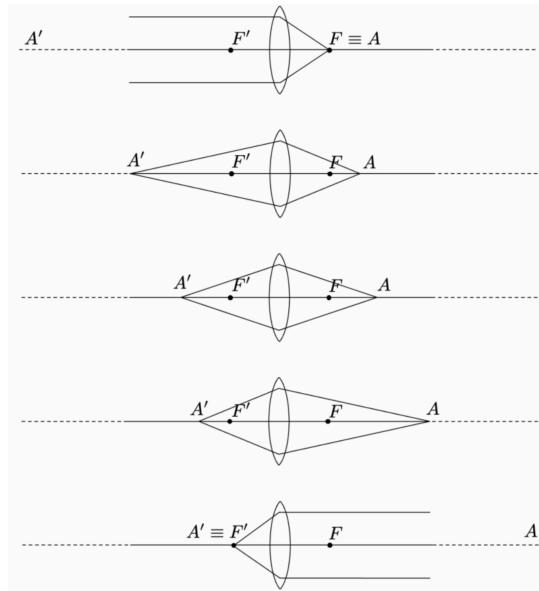


Figure 2.7: Conjugate points.

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:

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

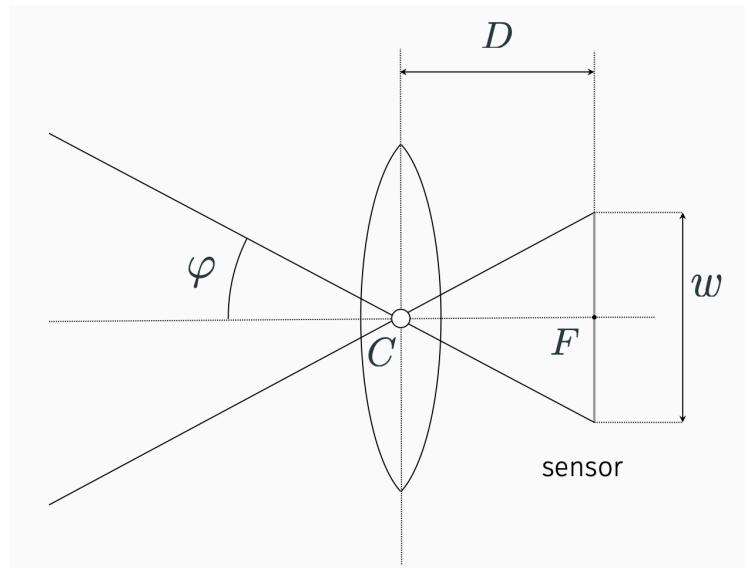


Figure 2.8: Field of view.

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 :

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

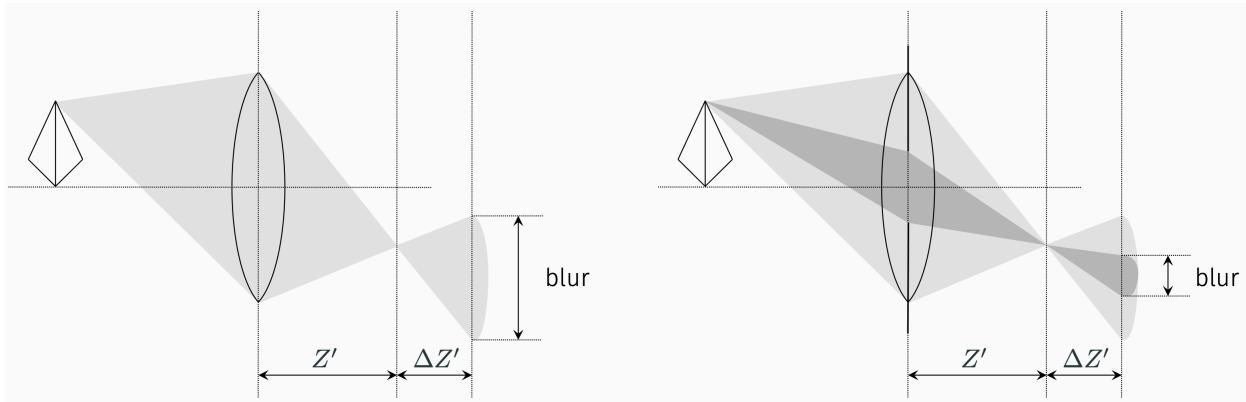


Figure 2.9: Blur circle formation.

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.

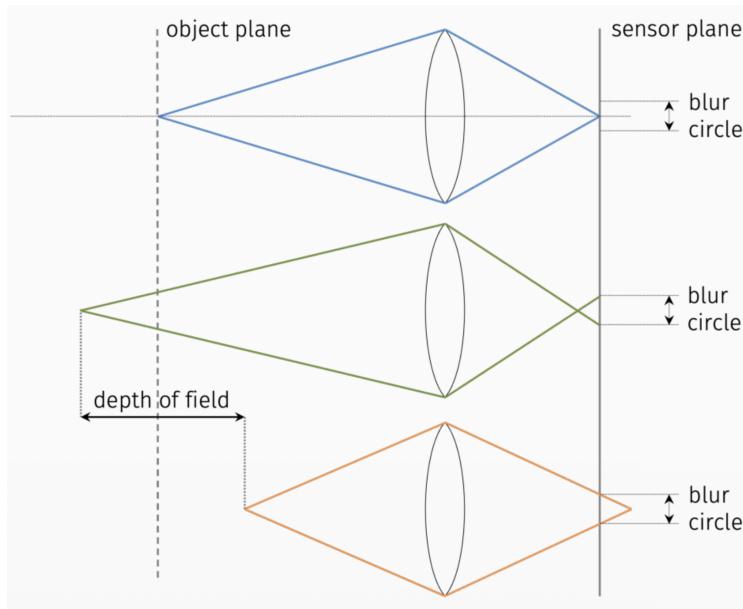


Figure 2.10: Depth of field.

⌚ 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.

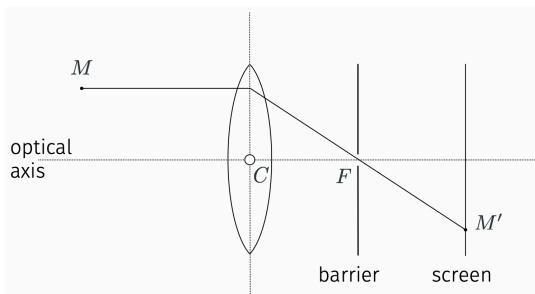


Figure 2.11: Lens aberrations.

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.

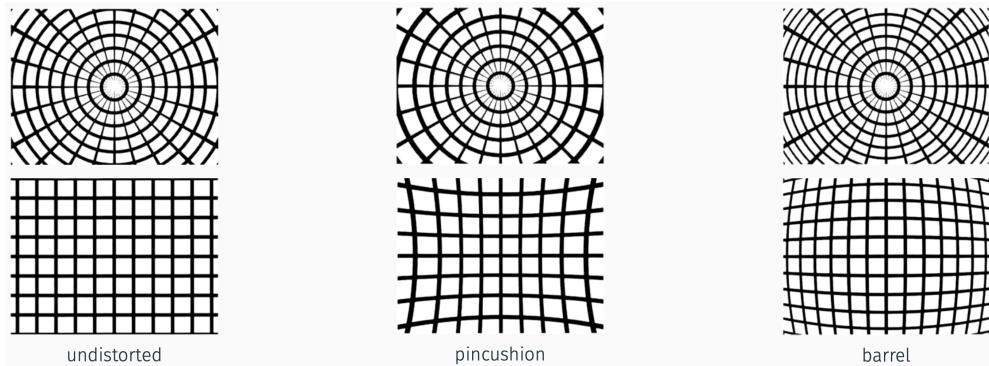


Figure 2.12: Examples of radial distortion

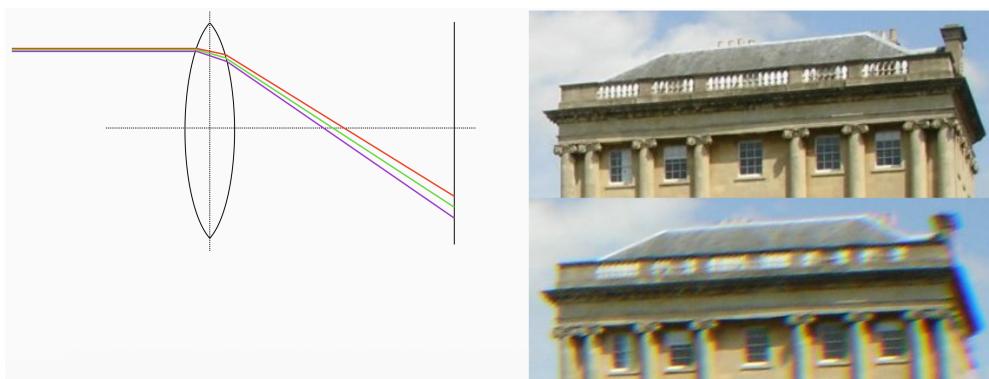


Figure 2.13: Example of chromatic aberration.

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.

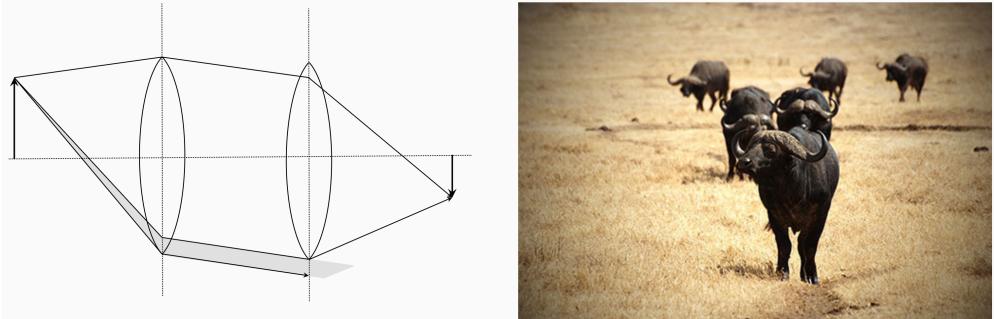


Figure 2.14: Vignetting effect.

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.

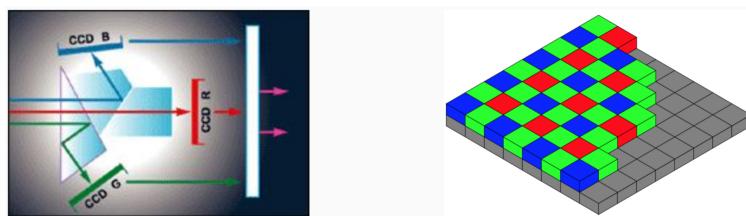


Figure 2.15: 3-sensor beam splitter on the left, Bayer mosaic on the right.

2.2 Camera Model

The **simplified model** of a camera is based on the pinhole camera, where the image plane is placed in front of the pinhole to avoid the inversion of the image.

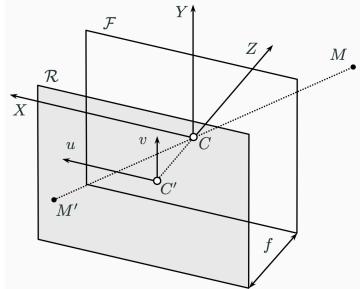


Figure 2.16: Simplified pinhole camera model.

- (X, Y, Z) is the world reference frame, centered in $\mathcal{C} \in \mathcal{F}$;
- The Z axis corresponds to the optical axis;
- (u, v) is the image frame, centered in the intersection between the optical axis and the image plane;
- u and v are oriented as X and Y ;
- The perspective projection is a map $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ such that

$$(x, y, z) \rightarrow \left(-\frac{fx}{z}, -\frac{fy}{z} \right)$$

- The map is nonlinear (due to the division by z);
- A linear map can be recovered by using homogeneous coordinates.

Homogeneous coordinates allow to add **improper points** (points at infinite) to the usual Euclidean plane (or space). The basic idea is to represent a point (x, y) of the plane as a triplet (u, v, w) of real numbers, in such a way that:

$$x = \frac{u}{w} \quad \text{and} \quad y = \frac{v}{w}$$

where $w \neq 0$. The triplet (u, v, w) is called the **homogeneous representation** of the point (x, y) . The null triplet $(0, 0, 0)$ does not represent any point. Moreover, any two triplets (u, v, w) and (ku, kv, kw) , with $k \neq 0$, represent the same point (x, y) .

Definition: Projective plane

The **projective plane** \mathbb{P}^2 is the set of equivalence classes of non-zero triplets (u, v, w) , where two triplets are equivalent if they differ by a non-zero scale factor k .

$$(u, v, w) \sim \lambda(u, v, w) \quad \forall \lambda \neq 0$$

The projective plane includes all points in the Euclidean plane (where $w \neq 0$) as well as points at infinity (where $w = 0$).

A similar definition holds for the **projective space** \mathbb{P}^3 , where points are represented by quadruplets (X, Y, Z, W) and two quadruplets are equivalent if they differ by a non-zero scale factor k .

Example:

Consider the vector $(x', y', 0)$. We have points that specify a direction but are at an infinite distance from focal point. So,

$$\begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} \frac{x'}{0} \\ \frac{y'}{0} \\ \infty \end{pmatrix} \rightarrow \begin{pmatrix} \infty \\ \infty \end{pmatrix}$$

Points whose last coordinate is zero are said to be **improper**.

Improper points represent a point at infinite along a direction specified by the non-zero coordinates. Points whose last coordinate is not zero, are said to be **proper**. To represent them, we can simply "add" a coordinate. With no loss of generality it may be taken as equal to 1, obtaining the so called **augmented vector**.

$$(x, y) \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (x, y, z) \rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Converting from homogeneous coordinates to Cartesian coordinates is called **dehomogenization** and amounts to dividing by the last coordinate:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \rightarrow \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \\ \frac{1}{w} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \\ 1 \end{bmatrix}$$

2.2.1 Transformations

Let the augmented vector corresponding to x be denoted by \bar{x} .

- **Translation:** a translation of t may be written as

$$x' = x + t$$

or

$$x' = [I|t]\bar{x}$$

or

$$\bar{x}' = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

- **Rotation and translation** (rigid body motion):

$$x' = Rx + t$$

or

$$x' = [R|t]\bar{x}$$

or

$$\bar{x}' = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

where R is an orthonormal matrix, thus $RR^T = R^T R = I$ and $\det(R) = 1$.

⌚ Observation:

There is also $\det(R) = -1$, which encodes a **mirroring** effect, not allowed in rigid transformations.

In the 2D case, given the angle θ of rotation, R takes the form

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

where R is a rotation matrix.

- **Similarity transformation** (where s is the scale factor):

$$x' = sRx + t$$

or

$$x' = [sR|t]\bar{x}$$

or

$$\bar{x}' = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I & t \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R & t \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} sR & t \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$	6	parallelism	

Figure 2.17: 2D Transformations

Let m and m' be the vectors of homogeneous coordinates of M (w.r.t. the camera frame) and M' (w.r.t. the image frame):

$$m = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad m' = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Then

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = z \begin{bmatrix} -\frac{fx}{z} \\ -\frac{fy}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} -fx \\ -fy \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Thus, in matrix notation

$$m' = \frac{1}{z} P m$$

or equivalently

$$m' \sim P m$$

where \sim means equality up to a scale factor.

2.2.2 General Model

For a more general camera model we need to consider:

- the rigid transformation from the world reference frame to the camera frame;
- the **pixelization** (from meters to pixel, in the image reference frame);
- The pixelization amounts to a **rescaling** followed by a translation:

$$V = \begin{bmatrix} -k_u & 0 & u_0 \\ 0 & -k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where (u_0, v_0) are the coordinates (in pixels) of \mathcal{C}' ;

- k_u and k_v the reciprocal of the pixel size along u and v :

$$k_u = \frac{1}{s_u} \quad k_v = \frac{1}{s_v} \quad [\text{pixel} \times m^{-1}]$$

Thus the new P taking into account the sensor is:

$$P = \begin{bmatrix} -k_u & 0 & u_0 \\ 0 & -k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & -u_0 & 0 \\ 0 & fk_v & -v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = K [I|0]$$

where, by posing $fk_u = \alpha_u$ and $fk_v = \alpha_v$, we have

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad [I|0] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- K is the **calibration matrix**;
- K describes the camera intrinsics (those parameters that depend on the camera itself, as opposite to the camera's pose in space, which are called extrinsics);
- Notice that K is upper triangular.

Let's now take into account the rigid transformation from the world reference frame to the camera frame and the pixelization. Let m_c be the vector of homogeneous coordinates of M with respect to the camera reference frame and let

$$G = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$$

be the rigid body motion from the world frame to the camera frame (a rotation R followed by a translation t). It follows that

$$m_c = Gm$$

The matrix G represents the camera extrinsics. The perspective projection is thus:

$$m' \sim K [I|0] Gm$$

The perspective projection matrix in the general case is

$$P = K [I|0] G = K [R|t]$$

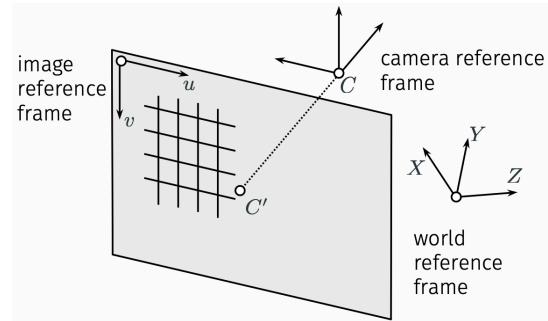


Figure 2.18: General camera model.

By partitioning R and t row-wise:

$$R = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix}, \quad t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

we obtain

$$P = \begin{bmatrix} \alpha_u r_1^T + u_0 r_3^T & \alpha_u t_1 + u_0 t_3 \\ \alpha_v r_2^T + v_0 r_3^T & \alpha_v t_2 + v_0 t_3 \\ r_3^T & t_3 \end{bmatrix}$$

Observation:

1. A more general form of K takes into account a further intrinsic parameter (the skew angle between the u and v axes, usually very close to $\frac{\pi}{2}$), thus

$$K = \begin{bmatrix} \alpha_u & \alpha_u \cot \theta & u_0 \\ 0 & \frac{\alpha_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. P is defined up to a scale factor (for $\lambda \neq 0$, λP results in the same projection as P). A prospective projection matrix is said to be *normalized* if has the form

$$\left[\begin{array}{ccc|c} * & * & * & * \\ * & * & * & * \\ \hline a^T & & & * \end{array} \right]$$

3. P has 12 entries, but (due to the scale factor) 11 degrees of freedom. Indeed, it depends on $6 + 5 = 11$ independent parameters (6 extrinsic and 5 intrinsic).
4. Partitioning P row-wise

$$P = \begin{bmatrix} p_1^T \\ p_2^T \\ p_3^T \end{bmatrix}$$

we get

$$m' = Pm = \begin{bmatrix} p_1^T m \\ p_2^T m \\ p_3^T m \end{bmatrix} \xrightarrow{\text{back to inhomogeneous}} \begin{cases} u = \frac{p_1^T m}{p_3^T m} \\ v = \frac{p_2^T m}{p_3^T m} \end{cases}$$

where u and v are the Cartesian coordinates of point M' in the image plane. Thus the last pair of equations is the generalization of the basic perspective projection equations.

Theorem 1. Let $P = [Q|q]$ be 3×4 matrix and let $q_i^t, i = 1, 2, 3$ denote the rows of Q .

- A necessary and sufficient condition for P to be a perspective projection matrix is that $\det(Q) \neq 0$
- A necessary and sufficient condition for P to be a zero-skew perspective projection matrix is that $\det(Q) \neq 0$ and $(q_1 \times q_3)^T (q_2 \times q_3) = 0$

Center of Projection

The focal plane \mathcal{F} contains the points that are projected to the infinite (because the straight line through C does not intersect the image plane). Thus its equation is

$$p_3^T x = 0$$

This is the equation of the focal plane and all points that satisfy it are in the focal plane, meaning that their projection goes to infinity. The planes $p_1^T x = 0$ and $p_2^T x = 0$ project to the axes $u = 0$ and $v = 0$, respectively.

The center of projection C belongs to all the three planes, thus it is defined by

$$\begin{cases} p_1^T x = 0 \\ p_2^T x = 0 \\ p_3^T x = 0 \end{cases}$$

In other words, C is represented (in homogeneous coordinates) by the kernel of P .

P has rank 3 by construction (both K and R are invertible) hence, by the Rank-Nullity Theorem

$$\dim(\text{Ker}(P)) = 4 - \text{rank}(P) = 4 - 3 = 1$$

Let

$$\tilde{c} = [x_C \ y_C \ z_C]^t$$

be the vector of the Cartesian coordinates of C w.r.t. the world frame. We can write

$$P_c = [Q|q] \begin{bmatrix} \tilde{c} \\ 1 \end{bmatrix} = 0$$

then, solving for \tilde{c} we get

$$\tilde{c} = -Q^{-1}q$$

Optical Ray

The optical ray of an image point M' is the locus of points in space that project onto M' . May be expressed as the line through C and a point α located at infinity that projects onto M'

$$\alpha = \begin{bmatrix} Q^{-1}m' \\ 0 \end{bmatrix}$$

thus a parametric representation (in homogeneous coordinates) of the optical ray is

$$m = c + \lambda \begin{bmatrix} Q^{-1}m' \\ 0 \end{bmatrix}, \quad \lambda \in \mathbb{R}$$

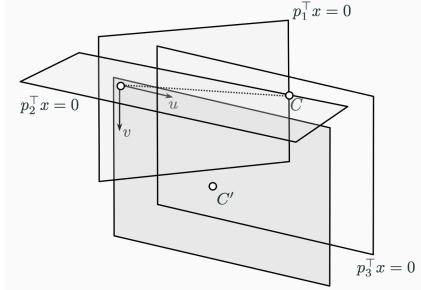


Figure 2.19: Center of projection.

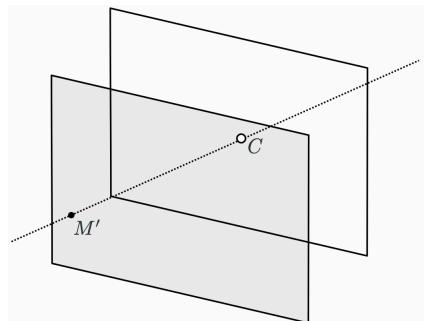


Figure 2.20: Optical ray.

② Advanced Concept: Depth of a point

Consider a point M whose Cartesian coordinates w.r.t. the world reference frame are $\tilde{m} = [x_M \ y_M \ z_M]^T$. The projection equation with explicit scale factor ζ is

$$\zeta m' = Pm$$

Representing M in homogeneous coordinates as $m = [\tilde{m}^T \ 1]^T$ we get

$$\begin{bmatrix} \zeta u \\ \zeta v \\ \zeta z \end{bmatrix} = P \begin{bmatrix} \tilde{m} \\ 1 \end{bmatrix}$$

If P is normalized, we obtain, from the third scalar equation, that

$$\zeta = r_3^T \tilde{m} + t_3$$

Recalling that $m_c = Gm$, it follows that ζ is the third coordinate of M w.r.t. the camera frame or the distance from M to the focal plane (depth of the point M). In other words, if

1. the last homogeneous coordinate is chosen as 1;

2. P is normalized;

then the last homogeneous projected coordinate ζ is the depth of the point.

2.3 Camera Calibration

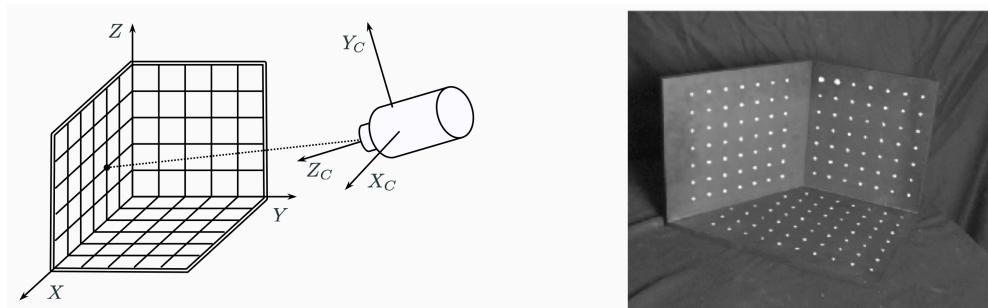


Figure 2.21: Camera calibration.

Camera calibration is the process of estimating the intrinsic and extrinsic parameters of a camera. This is typically done by capturing images of a known calibration pattern (e.g., a checkerboard) from different viewpoints and using these images to solve for the camera parameters.

- **Camera calibration problem:** estimate intrinsic and extrinsic camera parameters, based on some measurements;
- **Calibration pattern:** a 3D object of known geometry and generating image features which can be located accurately;
- Parameters may be estimated based on a set of known 3D-2D correspondences;
- **Direct methods:** formulate a problem in which the camera parameters are the unknowns;
- **Indirect methods:** first estimate the projection matrix P and then decompose it to get intrinsic and extrinsic parameters.

2.3.1 Direct Linear Transform

This is an indirect method. Suppose having n correspondences

$$m_{(i)} = \begin{bmatrix} x_{(i)} \\ y_{(i)} \\ z_{(i)} \\ 1 \end{bmatrix} \quad \text{and} \quad m'_{(i)} = \begin{bmatrix} u_{(i)} \\ v_{(i)} \\ 1 \end{bmatrix} \quad i = 1, \dots, n$$

For each correspondence, two scalar equations hold

$$\begin{cases} u_{(i)} = \frac{p_1^T m_{(i)}}{p_3^T m_{(i)}} \\ v_{(i)} = \frac{p_2^T m_{(i)}}{p_3^T m_{(i)}} \end{cases} \rightarrow \begin{cases} m_{(i)}^T \underbrace{p_1}_{\text{transpose of 1st row}} - u_{(i)} m_{(i)}^T p_3 = 0 \\ m_{(i)}^T \underbrace{p_2}_{\text{transpose of 2nd row}} - v_{(i)} m_{(i)}^T p_3 = 0 \end{cases}$$

Tip:

Remember that when applying a transposition or an inversion, the following properties hold:

- $(AB)^T = B^T A^T$;
- $(A^T)^{-1} = (A^{-1})^T$.

Thus, in a more compact form:

$$\underbrace{\begin{bmatrix} m_{(i)}^T & 0^T & -u_{(i)} m_{(i)}^T \\ 0^T & m_{(i)}^T & -v_{(i)} m_{(i)}^T \end{bmatrix}}_{A \text{ (knowns)}} \underbrace{\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}}_{P \text{ (unknowns)}} = 0$$

By stacking the n pairs of equations we get an **homogeneous linear system** of $2n$ equations in 12 unknowns (entries of P). We then have a $2n \times 12$ coefficient matrix A where at least $n = 6$ points are needed to compute P . In general, and in absense of measuring errors, A has rank 11, and the solution is the null-space of A . In practice, A will be full-rank and b can be estimated by solving the following optimization problem:

$$\arg \min_{\|p\|=1} \|Ap\|^2$$

where the constraint forces to discard the trivial solution $p = 0$ (which is not acceptable since $P \neq 0$). Notice that since P is defined up to a scale factor, any constraint of the form $\|p\| = \lambda \neq 0$ will result in the same actual estimated projection.

The solution of the minimization problem is the right singular vector of A corresponding to the smallest singular value. If

$$A = U\Sigma V^T$$

then $p = v_{12}$, i.e. the rightmost column of V .

Observation:

We expect $\text{rank}(A) \leq 11$, but in practice (due to the noise) we usually get 12. There is thus no exact solution except for 0, but it is not acceptable by default.

The **cross product** $a \times b$ is a vector defined as

$$a \times b = \begin{bmatrix} \det \begin{bmatrix} a_2 & a_3 \\ b_2 & b_3 \end{bmatrix} \\ -\det \begin{bmatrix} a_1 & a_3 \\ b_1 & b_3 \end{bmatrix} \\ \det \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \end{bmatrix}$$

It is null if and only if $b = \lambda a$ for any scalar λ . It may be obtained as a matrix multiplication with a skew-symmetric matrix. Let

$$[a]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

then $[a]_{\times}b = 0$ if and only if $b = \lambda a$.

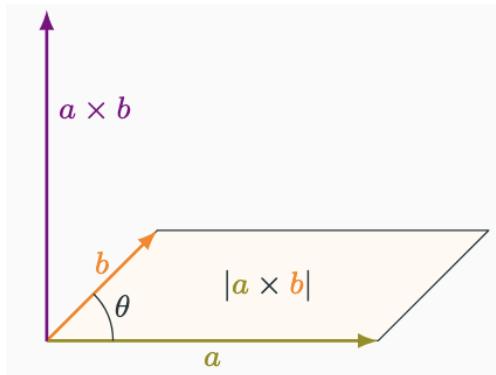


Figure 2.22: Cross product geometric interpretation.

Definition: Kronecker product

Given the matrices $A^{m \times n}$ and $B^{p \times q}$, their **Kronecker product** $A \otimes B$ is the matrix of size $(mp) \times (nq)$ defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}$$

This product is defined for any pair of matrices, is non commutative and enjoys the following properties:

- $(A \otimes B) \otimes C = A \otimes (B \otimes C)$;
- $(A \otimes B)^T = A^T \otimes B^T$;
- $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$;
- For square A and B : if λ_A is an eigenvalue of A and λ_B is an eigenvalue of B , then $\lambda_A \lambda_B$ is an eigenvalue of $A \otimes B$;
- $\text{rank}(A \otimes B) = \text{rank}(A) \times \text{rank}(B)$.

Definition: Vector Operator

Given a matrix $X = [x_1 \ x_2 \ \dots \ x_n]$, the **vector operator** vec is the operator that stacks the columns one underneath the other to a single vector:

$$\text{vec}(X) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The following identity holds:

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X)$$

and is used to "pull out" the unknown X from a matrix equation.

For square matrices, the vech (vector half) operator turns out to be useful:

Definition: Vector Half Operator

Given a square matrix $X = [x_1 \ x_2 \ \dots \ x_n]$, the **vector half operator** (vech) is the operator that stacks in a single column the elements of X , which are on or below the diagonal, taken in column-wise order. Thus, $\text{vech}(X)$ has $n(n+1)/2$ elements.

The projection equation states that:

$$m'_{(i)} \sim Pm_{(i)}$$

and taking the product of both sides by the vector $Pm_{(i)}$ we get:

$$m'_{(i)} \times (Pm_{(i)}) = 0$$

Thus we can get rid of the arbitrary scale factor and write:

$$m'_{(i)} \times Pm_{(i)} = 0$$

that is equivalent to

$$\left[m'_{(i)} \right]_{\times} Pm_{(i)} = 0$$

and by transposing, to

$$m_{(i)}^T P^T \left[m'_{(i)} \right]_{\times}^T = 0^T$$

Then, by applying the vector operator to both sides, and recalling that

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X)$$

we obtain a triplet of linear equations in the unknown $\text{vec}(P^T)$:

$$\left(\left[m'_{(i)} \right]_{\times} \otimes m_{(i)}^T \right) \text{vec}(P^T) = 0$$

or, by expanding the coefficients:

$$\begin{bmatrix} 0^T & -m_{(i)}^T & v_{(i)}m_{(i)}^T \\ m_{(i)}^T & 0^T & -u_{(i)}m_{(i)}^T \\ -v_{(i)}m_{(i)}^T & u_{(i)}m_{(i)}^T & 0^T \end{bmatrix} \text{vec}(P^T) = 0$$

Notice that only two equations are independent since the Kronecker product ($\left[m'_{(i)}\right]_\times \otimes m^T_{(i)}$) is rank two.

For random configurations of $n \geq 6$ points, A will have rank 11 or 12. In either case, a unique solution can be found (up to a scale factor). The **degenerate configurations** are those for which P cannot be uniquely determined.

$$\dim(\ker(A)) \geq 2 \rightarrow \text{rank}(A) \leq 10$$

meaning that the space of solutions has dimension greater than one. The most important degenerate configurations are:

- the points all lie on the union of a plane and a single straight line containing the camera center;
- the camera and the points all lie on a twisted cubic passing through the origin.

 **Tip:**

In practice, don't take more than six non-coplanar points.

So, DLT minimizes the algebraic residual $\|Ap\|^2$, which has no geometrical meaning and is not invariant with respect to changes of the reference frame. Instead, the **geometric residual**:

$$\varepsilon(P) = \sum_{i=1}^n \left(\frac{p_1^T m_{(i)}}{p_3^T m_{(i)}} - u_{(i)} \right)^2 + \left(\frac{p_2^T m_{(i)}}{p_3^T m_{(i)}} - v_{(i)} \right)^2$$

best expresses the actual need, that is to minimize the reprojection error (the distance between the estimated and the measured projections). The resulting non-linear optimization problem can be solved iteratively, starting from an initial guess obtained via the DLT.

Given the estimated perspective projection matrix P , the intrinsic and extrinsic camera parameters can be computed as follows. The matrix P is given in the form

$$P = [Q|q]$$

where Q is **invertible**. We want to express P as

$$P = K[R|t] = [KR|Kt]$$

where K is upper triangular and R is orthogonal. Thus,

$$Q = KR \quad \text{and} \quad q = Kt$$

Recalling that any real square matrix can be written as the product of an orthogonal matrix and an upper triangular matrix, and letting $Q^{-1} = US$ be the QR factorization of Q^{-1} , we can write:

$$Q = S^{-1}U^{-1}$$

where S^{-1} is upper triangular being the inverse of an upper triangular matrix, and U^{-1} is orthogonal as U is orthogonal. Hence a solution is:

$$R = U^{-1}, \quad K = S^{-1}, \quad t = K^{-1}q$$

 **Observation:**

Note that P may contain an arbitrary scale factor that will affect K . The proper scale for K is obtained by imposing $K_{33} = 1$.

A further observation is that it is not possible to recover the "true" focal length f from the 3D-2D correspondences only (without knowing the actual sensor size or the pixel size). Indeed, recalling that the calibration matrix K takes the form

$$K = \begin{bmatrix} \alpha_u & \alpha_u \cot \theta & u_0 \\ 0 & \frac{\alpha_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fk_u & fk_u \cot \theta & u_0 \\ 0 & \frac{fk_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where k_u and k_v are the reciprocal of the pixel sizes, along u and v respectively, it is clear that f is "entangled" with k_u and k_v . Thus, without knowing the pixel sizes, it is not possible to separate f from them. If we, for instance, double the focal length as well as the pixel sizes we get the same calibration matrix. Thus, we can only recover α_u and α_v .

?

Advanced Concept: Dealing with radial distortion

2.3.2 Zhang's method

Since DLT calibration requires an image of many (≥ 6) non co-planar points, and so building an accurate 3D calibration object may be difficult, we can use an easier method that requires ≥ 3 images of a plane and is based on *homography*: the **Zhang's calibration method**.

≡ Definition: Homography

A **homography** is an isomorphism of P^n , i.e., a map

$$x \rightarrow Hx, \quad H^{(n+1) \times (n+1)}, \text{ non-singular}$$

where H and $\lambda H, \lambda \neq 0$ represent the same homography.

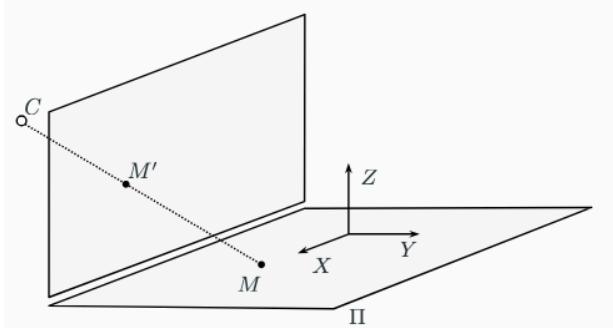


Figure 2.23: Homography mapping between two planes.

The map from a 3D plane π to its perspective image is easily seen to be a homography. Assume for simplicity that the plane has equation $z = 0$. Then

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = [p_1 \ p_2 \ p_3 \ p_4] \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \underbrace{[p_1 \ p_2 \ p_4]}_{=H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where the matrix H is non-singular unless for degenerate cases. Given at least 4 pairs of original and projected points, a matrix H corresponding to a homography can be computed via DLT.

Suppose a homography H of a 3D plane π to the image plane is known (e.g. estimated from measurements):

$$H = \lambda [p_1 \ p_2 \ p_4]$$

Since $P = K[R|t] = K[r_1 \ r_2 \ r_3 \ t]$, we get $p_1 = Kr_1$, $p_2 = Kr_2$ and $p_4 = Kt$. Then, by substitution:

$$H = \lambda K [r_1 \ r_2 \ t]$$

Thus:

$$\begin{cases} h_1 = \lambda Kr_1 \\ h_2 = \lambda Kr_2 \end{cases} \rightarrow \begin{cases} r_1 = \frac{1}{\lambda} K^{-1} h_1 \\ r_2 = \frac{1}{\lambda} K^{-1} h_2 \end{cases}$$

Now, being R a rotation matrix, we have

$$\begin{cases} r_1^T r_2 = 0 \\ r_1^T r_1 = r_2^T r_2 = 1 \end{cases} \rightarrow \begin{cases} h_1^T (KK^T)^{-1} h_2 = 0 \\ h_1^T (KK^T)^{-1} h_1 = h_2^T (KK^T)^{-1} h_2 \end{cases}$$

Thus, from a plane π and the corresponding estimated homography H , we get two scalar constraints:

$$\begin{cases} h_1^T Bh_2 = 0 \\ h_1^T Bh_1 = h_2^T Bh_2 \end{cases}$$

where $B = (KK^T)^{-1}$. The constraints are linear in the unknown B . By taking n planes and the corresponding estimated homographies

$$\Pi^{(k)}, H^{(k)} \quad k = 1, \dots, n$$

we get $2n$ constraints on the same B .

How to solve for B ? First, note that $B = B^T$ by construction, so B is symmetric; hence there are 6 unknowns. Let

$$b = [B_{11} \ B_{12} \ B_{22} \ B_{13} \ B_{23} \ B_{33}]^T$$

by the vector of unknowns. For $i, j \in \{1, 2\}$, it is verified that

$$h_i^T Bh_j = v_{ij}^T b$$

where

$$v_{ij} = \begin{bmatrix} H_{1i}H_{1j} \\ H_{1i}H_{2j} + H_{2i}H_{1j} \\ H_{2i}H_{2j} \\ H_{3i}H_{1j} + H_{1i}H_{3j} \\ H_{3i}H_{2j} + H_{2i}H_{3j} \\ H_{3i}H_{3j} \end{bmatrix}$$

Thus, the two equations become:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0$$

For n planes, by stacking equations of the above form, we get

$$Vb = 0$$

where V is $2n \times 6$.

For b being uniquely determined (up to a scale factor), V must have rank 5:

- At least $n = 3$ planes in general configuration are needed;

- In practice, the more the planes, the better the accuracy;
- In practice, due to noise, V will be full rank and the system can be solved by linear least squares:

$$\underset{\|b\|=1}{\arg \min} \|Vb\|^2$$

where the constraint forces to discard the trivial solution $b = 0$; The solution of the above minimization problem is well known and is the right singular vector of V corresponding to the smallest singular value. If the singular value decomposition of V is

$$V = U\Sigma V^T$$

then $b = v_6$, i.e. the rightmost column of S .

Given $B = (KK^T)^{-1}$, the matrix K can be found by the Cholesky factorization. Recall that any real symmetric and positive definite matrix can be decomposed **uniquely** as the product of lower triangular matrix having strictly positive eigenvalues and its transpose. Let

$$B = LL^T$$

be the Cholesky factorization of B , where L is lower triangular. On the other hand we have

$$B = (KK^T)^{-1} = (K^T)^{-1}K^{-1} = (K^{-1})^T K^{-1}$$

where $(K^{-1})^T$ is lower triangular, being the transpose of an upper triangular matrix. By comparison, we get

$$K^{-1} = L^T \quad \rightarrow \quad K = (L^{-T})^{-1}$$

As usual, the proper scale for K is obtained by imposing $K_{33} = 1$.

Finally, the extrinsic parameters for each image are readily computed as:

$$r_1 = \lambda K^{-1} h_1, \quad r_2 = \lambda K^{-1} h_2, \quad r_3 = r_1 \times r_2, \quad t = \lambda K^{-1} h_3$$

where $\lambda = \frac{1}{\|K^{-1}h_1\|} = \frac{1}{\|K^{-1}h_2\|}$ and \times denotes the cross product.

Due to noise, the resulting matrix R may not be orthogonal. The closest (in the Frobenius norm) orthogonal matrix R' to a given matrix R is known to be

$$R' = UV^T$$

where $R = U\Sigma V^T$ is the singular value decomposition of R .

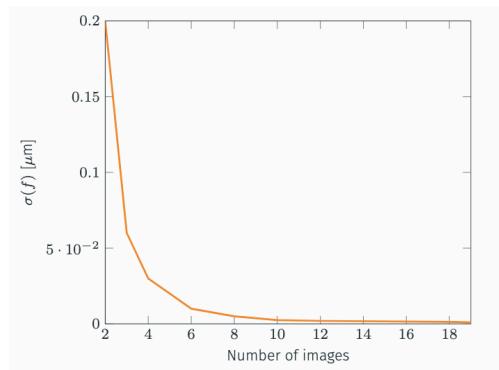


Figure 2.24: Std of the focal length distance f as function of the number of images that are used for calibration. The std is computed on the outcomes of every possible subset of l images from a total of 20 images.

② Advanced Concept: *Alternative derivation*

2.3.3 Estimating homographies

2.4 Stereopsis

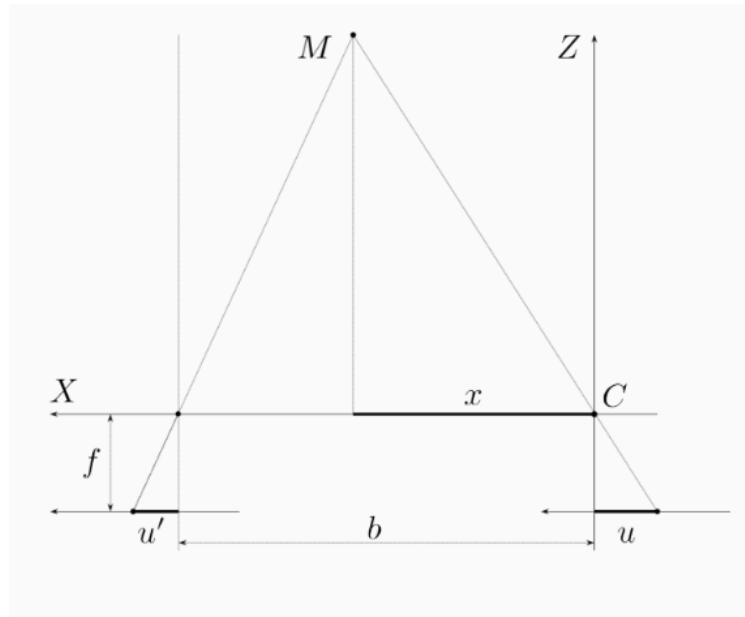
Computational stereopsis has the goal of getting information regarding the 3D structure of a scene, given a pair of images captured by two cameras in different positions. There are two steps:

- Computing the correspondences between the two images;
- Performing triangulation to get the 3D points.

The first is based on feature detection and matching. Two points in the two images that correspond to the same point of the scene are called **conjugate points**. Given the location of them corresponding to the 3D point M of the scene, and the two perspective projection matrices, the location in the 3D space of M can be found by triangulation.

2.4.1 Triangulation

Given two cameras, aligned and having parallel optical axes, the disparity is defined as the horizontal difference between the coordinates of the conjugate points in the two images. Consider the following figure:



Placing the world reference frame in correspondence of the camera on the right, the perspective projection equations are:

$$\begin{cases} \frac{f}{z} = \frac{-u}{x} \\ \frac{f}{z} = \frac{-u'}{x-b} \end{cases}$$

Thus, solving for z we get:

$$z = \frac{fb}{u' - u}$$

so it is possible to compute the depth of point M from the disparity $u' - u$ and the geometry of the stereo pair (baseline b and f). Note that b acts as a scale factor and that the depth z is inversely proportional to the disparity.

Observation: Sensitivity to disparity error

Suppose that the disparity is affected by an error δd . What is the error on the computed depth z ?

By taking the derivative of $z = \frac{fb}{d}$ w.r.t. d we get:

$$\frac{\partial z}{\partial d} = -\frac{fb}{d^2}$$

thus, for small errors δd we have:

$$\delta z \approx -\frac{fb}{d^2} \delta d$$

and by substitution:

$$\delta z \approx -\frac{z^2}{fb} \delta d$$

This means that for a given δd , the error is proportional to the square of the working distance z and inversely proportional to the baseline b and to the focal length f .

In the general case, let P' and P'' be the two perspective projection matrices and let $m' = [u' \ v' \ 1]^T$ be the homogeneous coordinates of M' and $m'' = [u'' \ v'' \ 1]^T$ be the homogeneous coordinates of M'' . Let $m = [x \ y \ z \ 1]^T$. As for M' , we have:

$$\begin{cases} (p_1'^T - u' p_3'^T)m = 0 \\ (p_2'^T - v' p_3'^T)m = 0 \end{cases}$$

By considering the conjugate point M'' , we get another pair of equations, thus we can write a homogeneous linear system of 4 equations and 4 unknowns:

$$\begin{bmatrix} p_1'^T - u' p_3'^T \\ p_2'^T - v' p_3'^T \\ p_1''^T - u'' p_3''^T \\ p_2''^T - v'' p_3''^T \end{bmatrix} m = 0$$

The solution is the null space of the 4×4 coefficient matrix. The matrix must be rank 3, otherwise the only solution is the trivial one ($m = 0$). In practice, due to noise, the rank is 4 and a least squares solution is sought for, using SVD.

2.4.2 Epipolar Geometry

Epipolar geometry describes the relationship between conjugate points. Consider Fig. 2.25: given M'' , any possible conjugate point M' must lie on the intersection between \mathcal{R}' and the plane containing M, C', C'' . Such intersection is a straight line called **epipolar line** (relative to M''). All the epipolar lines in \mathcal{R}' pass through the point E' (the epipole) and, similarly, all the epipolar lines in \mathcal{R}'' pass through the epipole E'' . The plane containing M, C', C'' is the epipolar plane of M .

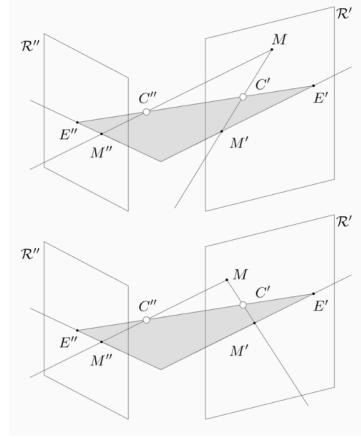


Figure 2.25: Epipolar geometry.



Figure 2.26: A stereo pair. Epipolar lines corresponding to the points on the left image, are drawn on the right image.

Let $P' = [Q'|q']$ and $P'' = [Q''|q'']$. Then:

$$\begin{cases} m' = P'm \\ m'' = P''m \end{cases}$$

The epipolar line corresponding to M' is the projection through P'' of the optical ray of M' , whose equation is:

$$m = c' + \lambda \begin{bmatrix} (Q')^{-1}m' \\ 0 \end{bmatrix}$$

where c' is the vector of homogeneous coordinates of C' w.r.t. the world reference frame.

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

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.