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 22, 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 **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

3 Features and Primitives **20**
3.1 Image Processing 20
3.1.1 Linear Filtering 20
3.1.2 Non-linear filters and other neighborhood operators 20
3.1.3 Image warping 20
3.1.4 Multi-resolution representations 20

4 Stereopsis **21**

5 Support Vector Machines **22**

6 Image Processing **23**

7 Feature Detection **24**

8 Fitting Geometric Primitives **25**

9 Recognition **26**

10 Deep Learning for Computer Vision **27**

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.

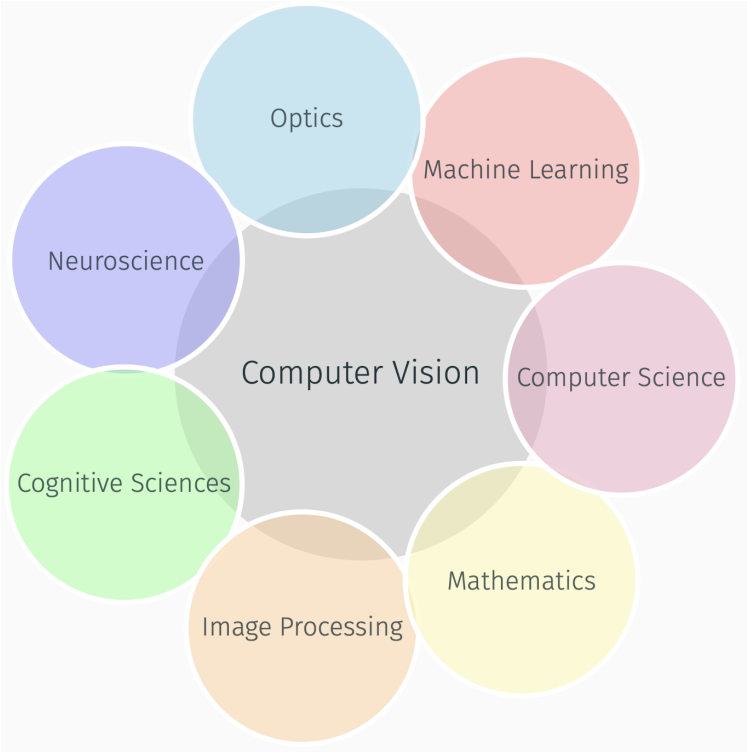


Figure 1.1: What is Computer Vision related to?

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).

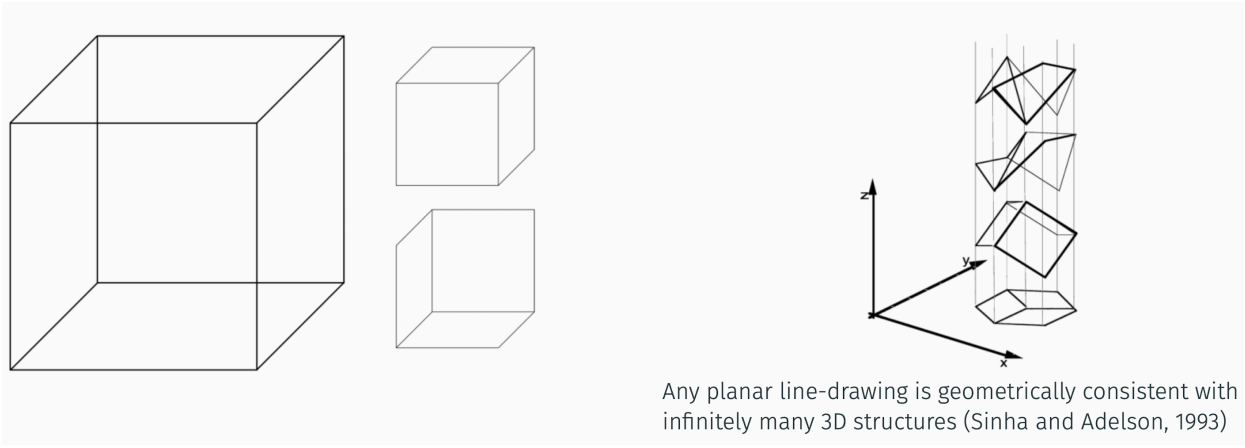


Figure 1.2: The forward and inverse problems.

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.

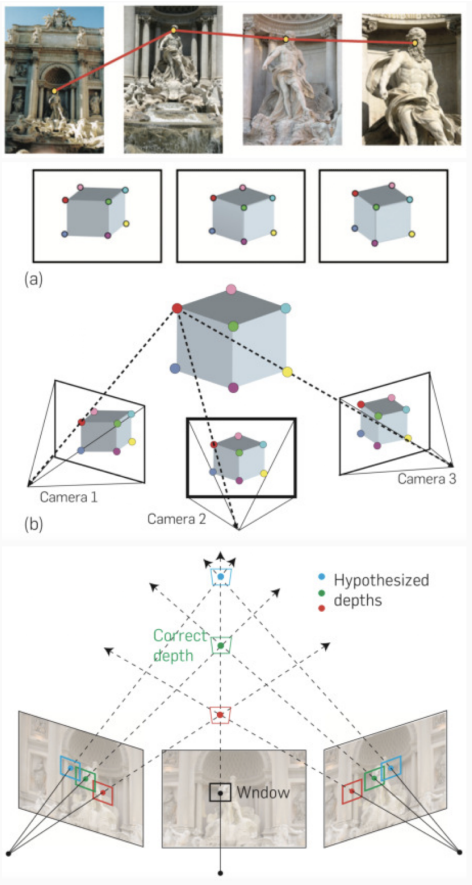
- Human vision may inspire, but most CV deals with finding effective ways for solving specific problems;
- there is no ”the” CV algorithm, but instead a **collection of algorithms/approaches** for tackling **specific problems** in **restricted domains**.

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

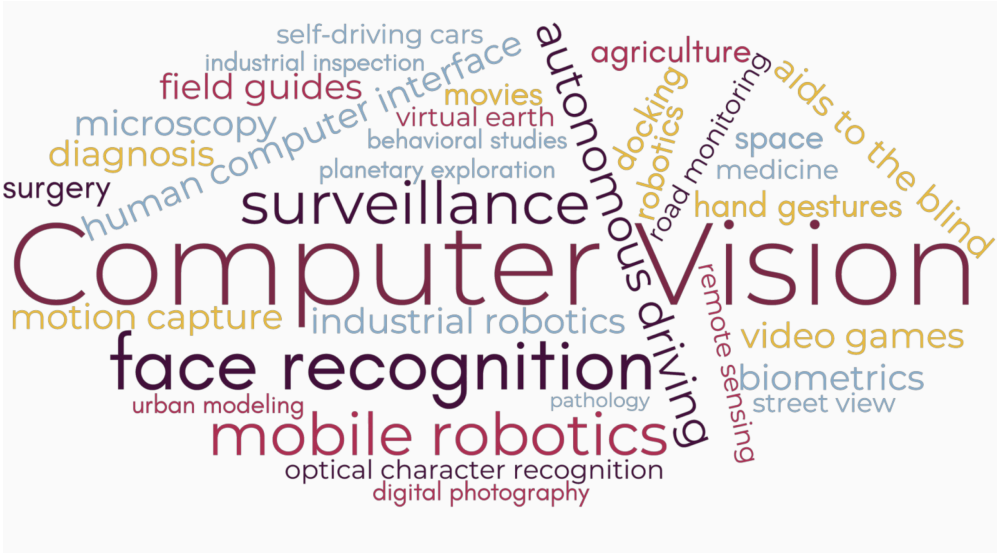


Figure 1.3: Some applications of Computer Vision.

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.

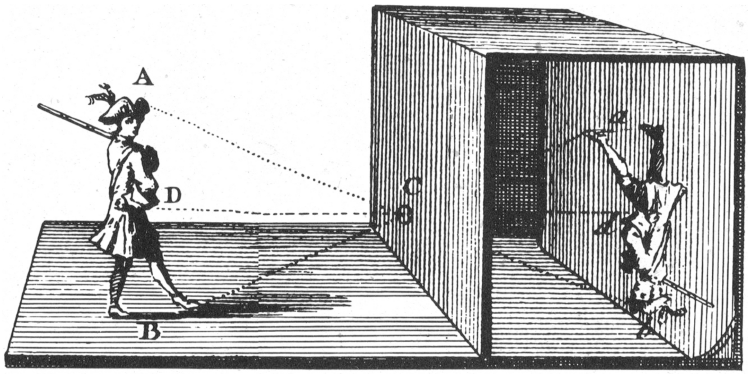


Figure 2.1: Camera Obscura principle.

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.

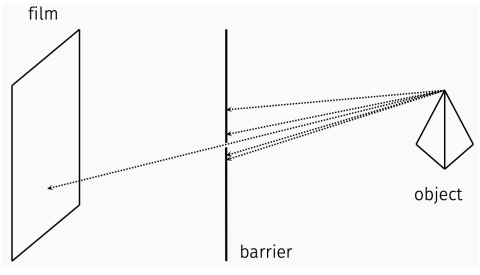


Figure 2.2: Pinhole camera principle.

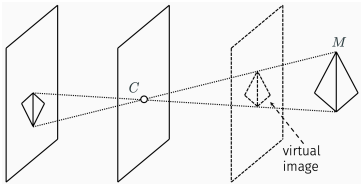
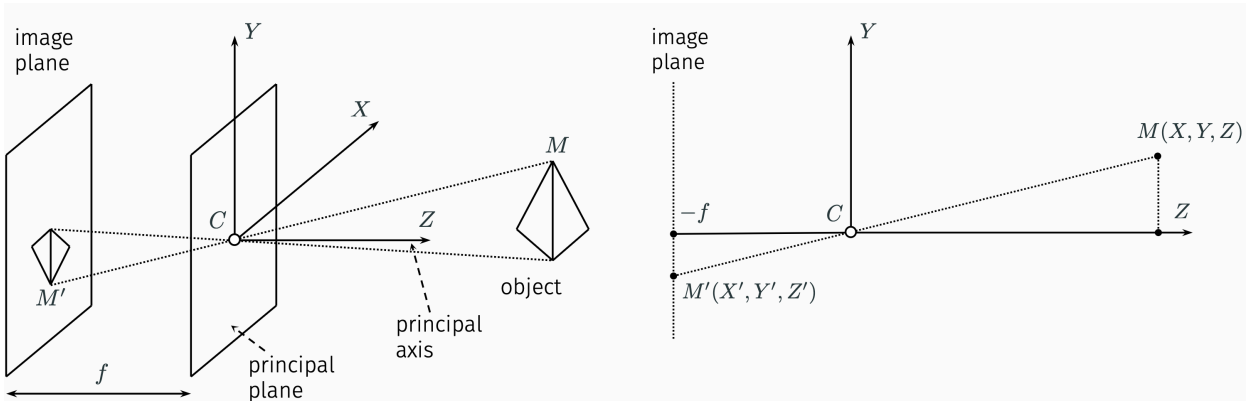


Figure 2.3: Pinhole camera model.

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

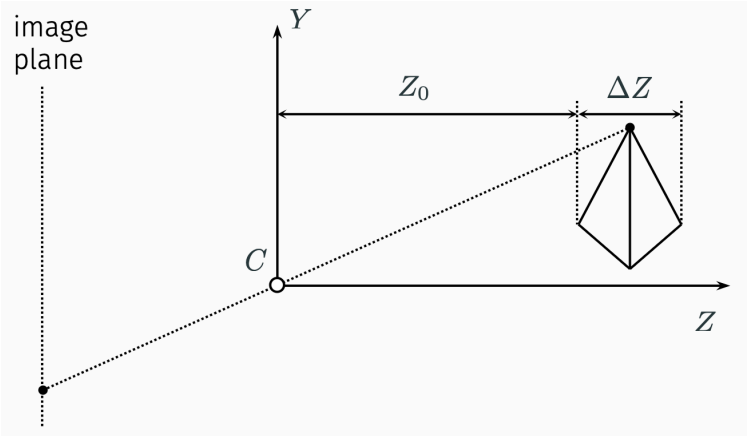


Figure 2.4: Weak perspective projection.

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.

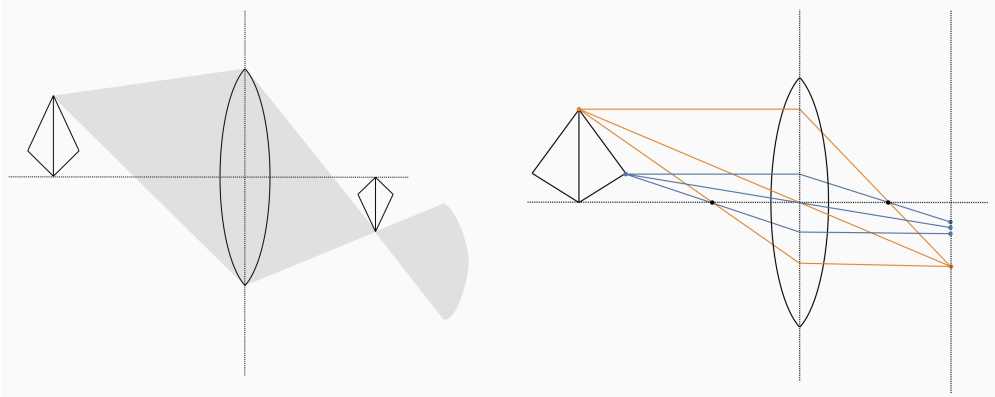
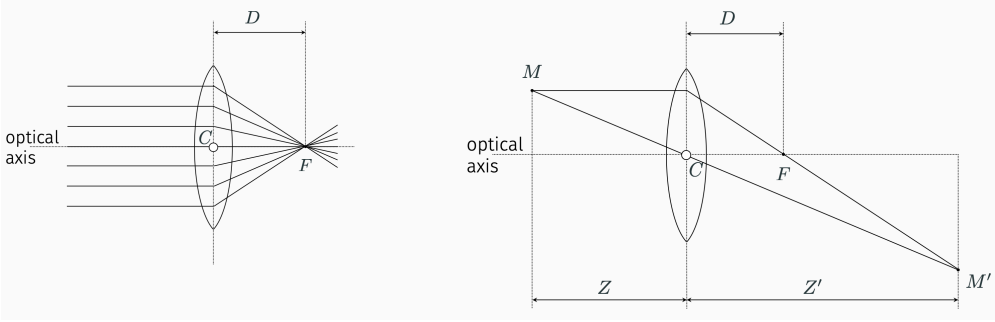


Figure 2.6: Lens principle. The gray areas represent the set of rays originated from the object.

- 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;
- 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.

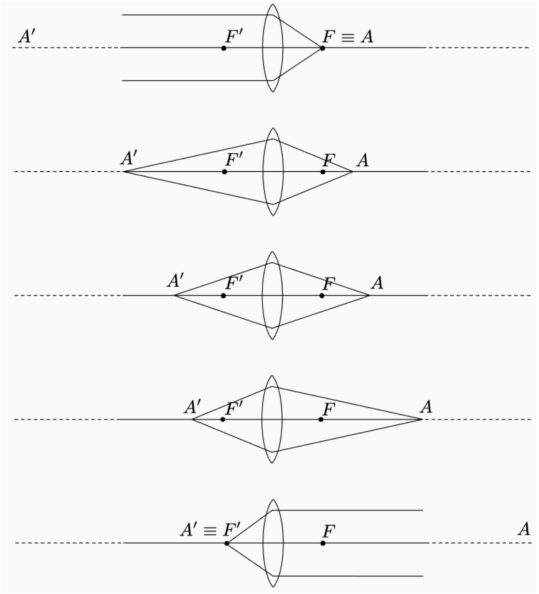


Figure 2.7: Conjugate points.

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:

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

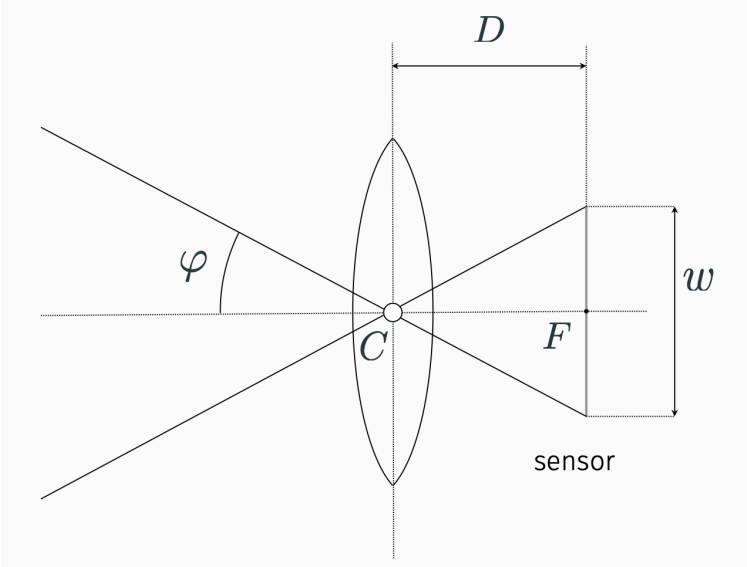


Figure 2.8: Field of view.

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:

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

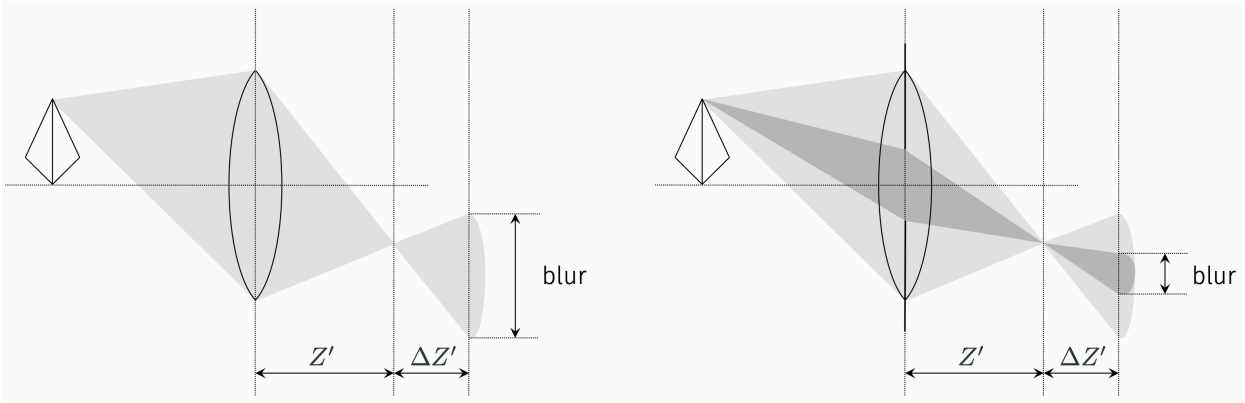


Figure 2.9: Blur circle formation.

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.

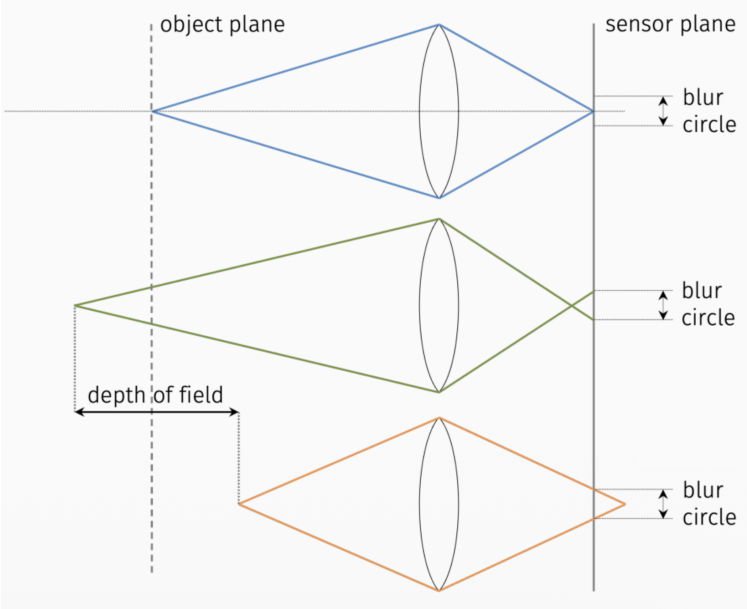


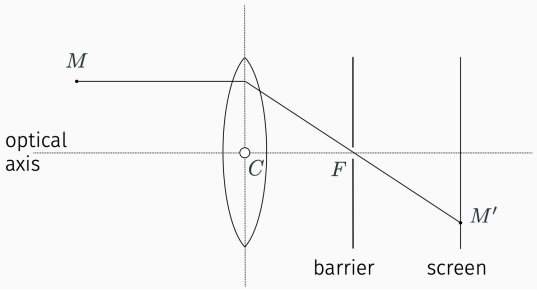
Figure 2.10: Depth of field.

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.



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**.

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.

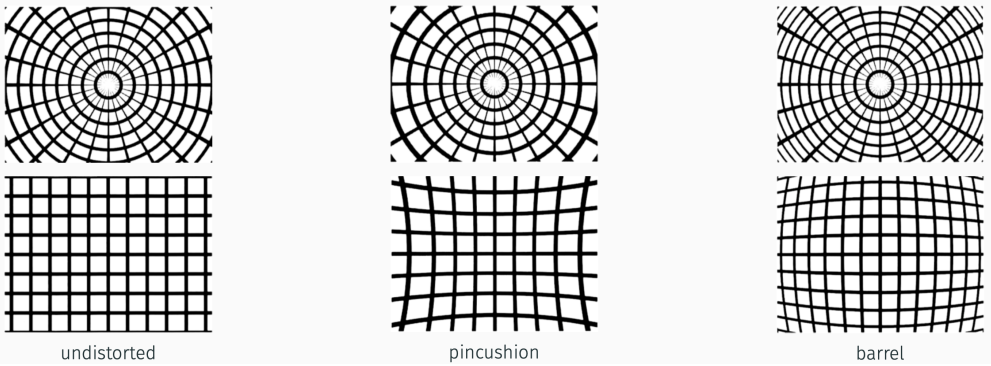


Figure 2.12: Examples of radial distortion

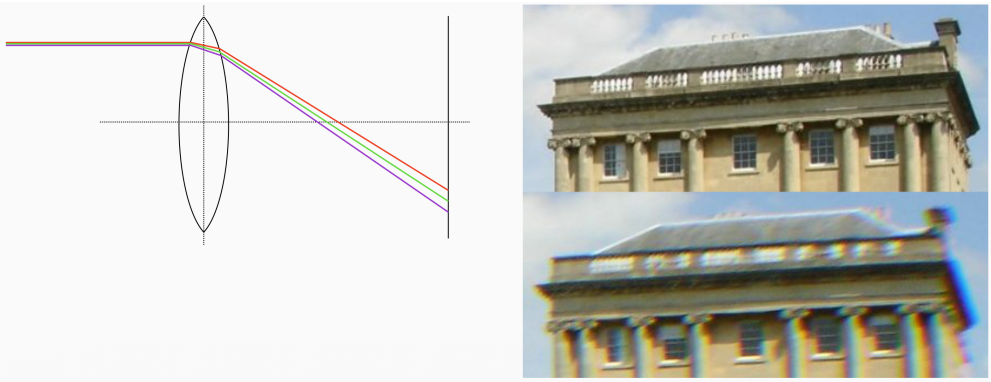


Figure 2.13: Example of chromatic aberration.

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.

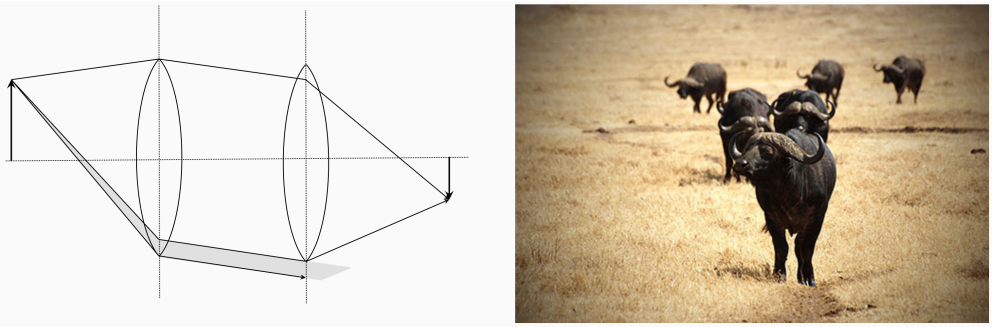


Figure 2.14: Vignetting effect.

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.

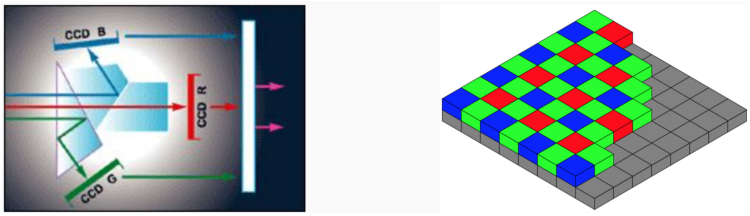


Figure 2.15: 3-sensor beam splitter on the left, Bayer mosaic on the right.

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.

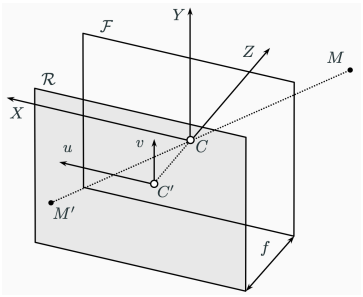


Figure 2.16: Simplified pinhole camera model.

- (X,Y,Z) is the world reference frame, centered in $\mathscr{C} \in \mathscr{F}$;
- 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 $\mathbb{R}^3 \to \mathbb{R}^2$ such that

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

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

Homogeneus coordinates allow to add **improper points** (points ad 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,

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

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 homogeneus coordinates to Cartesian coordinates is called **dehomogenization** and amounts to dividing by th elast coordinate:

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

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	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	3	lengths	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	4	angles	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{2\times 3}$	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

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

Thus, in matrix notation

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

or equivalently

$$m' \sim Pm$$

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 $\mathscr{C}'$;
- k_u and k_v the reciprocal of the pixel size along u and v:

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

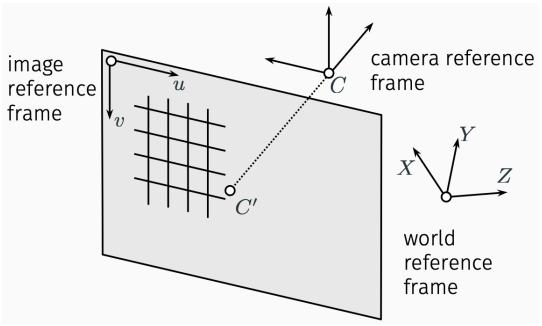


Figure 2.18: General camera model.

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

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

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
$$m' = Pm = \begin{bmatrix} p_1^T m \\ p_2^T m \\ p_3^T m \end{bmatrix} \rightarrow^{\text{back to inhomogeneous}} \begin{cases} u = \frac{p_1^T m}{p_3^T m} \\ v = \frac{p_2^T m}{p_3^T m} \end{cases}$$
where u and v are the Cartesian coordinates of point M' in the image plane. Thus the last pair or 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*
$$\det(Q) \neq 0 \quad \textit{and} \quad (q_1 \times q_3)^T (q_2 \times q_3) = 0$$

Center of Projection

The focal plane $\mathscr{F}$ 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

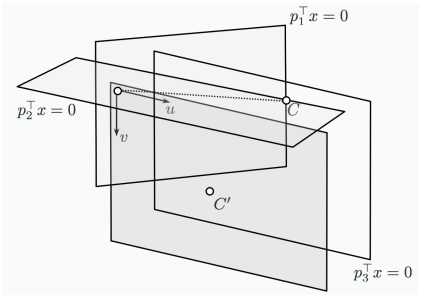


Figure 2.19: Center of projection.

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

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

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

Let

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

be the vector of the Cartesian coordiantes 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 parametrics representation (in homogeneous coordinates) of the optical ray is

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

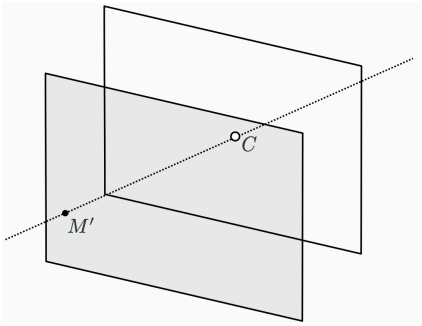


Figure 2.20: Optical ray.

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 $m = \left[\tilde{m}^T 1\right]^T$ 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

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 show to be equivalent to the following optimization problem:

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

  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

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.