Euclidean distance problems

Yuliya Aleksieva

Trabajo Fin de Máster
Máster Universitario en Modelización e Investigación Matemática, Estadística y Computación

Universidad Zaragoza

Facultad de Ciencias
Universidad Zaragoza

Tutor: José Ignacio Cogolludo Agustín

Contents

Prologue

Many times I have wondered what the great progress in medicine is due to. In recent decades, Medicine has advanced a lot and already cures diseases that were impossible to cure for a long time. This development is due to new technologies, among them: ultrasound, scan, resonance, etc.

Almost all diseases are tracked thanks to these new technologies that consist of reproducing images of the 3D world in a 2D plane (screen). These analyses are essential for making a correct decision by doctors for the cure of a patient. But what exactly are these technologies based on?

The key is the determination of a general computer vision problem, known as *triangulation*, and it consists of reconstructing a point in space $\mathbf{X}$, knowing its projections on two planes π and π' (image planes). To do this, we visualize the points in 3D space by extending them to the three-dimensional projective space $\mathbb{P}^3$. In addition, this triangulation process is applied in many other spheres of Sciences like Geology, Astronomy, Meteology, etc.

For me this work "Euclidean distance problems" has been a fascinating challenge since it encompasses many branches of Mathematics! It combines techniques from Algebra, Geometry, Numerical Analysis, Probability and even the Computational Part (implementation of algorithms)!

First of all, we give a brief description of Euclidean and Projective geometries. Projective geometry is a non-metric form of elementary geometry, that means that it is not based on a concept of distance. Next, we focus on the study of the transformations that preserve the structure of a projective space, preserving projective subvarieties. This sets a relation between geometric figures and their images (the most common example consists of 3D figures with projections on a plane 2D). Then, we discuss the basic concepts of projective geometry with which the process of capturing an image with cameras is associated. We introduce new notions such as: *line at infinity* and *ideal points.*

Now, having the basic tools of projective geometry, we can give an answer to the following question: *how does a camera create images of the three-dimensional world in two dimensions?* Using two main camera models (for camera model we understand the projection of $\mathbb{P}^3 \to \mathbb{P}^2$), where each model can be described by a (3×4)-matrix (up to scalar multiplication) of maximum rank. On the one hand, the finite camera models are characterized by having the first submatrix of $\mathbf{P}$ regular, and, on the other hand, the camera models at infinity are those whose first submatrix of $\mathbf{P}$ is singular (for example, the cognate chamber). We will also describe the structure that projective cameras have and the properties they possess.

Once we understand how a pinhole camera works, we introduce the concept of epipolar geometry (geometry of a stereo vision: the intersection of two planes π and π' of the image). This geometry can be described algebraically by means of a fundamental matrix $\mathbf{F}$, and this time it fulfills a series of properties related to the points of the planes π and π' (points).

This leads us to the following question: *How to find a point* $\mathbf{X}$ *in 3D that optimally fits the measured image points, that is, their projections* $\mathbf{x}$ *y* $\mathbf{x}'_i$ *on the planes* π *and* π'*?* (see Fig. 1). First, we have to base ourselves on the projective reconstruction theorem which aims to determine $\mathbf{H}$ (a 2D homography). Then, to answer the previous question, we give an algorithm for solving the *reconstruction (triangulation) problem.*

In practice, however, the coordinates of the image points $\mathbf{x}$ and $\mathbf{x}'_i$ are difficult to measure accurately. To do this, we are going to describe some of the best known and most successful algorithms, such as: *Direct Linear Transformation (DLT)* - to reduce errors; *Maximum Probability Estimate (MLE)* - (assuming that the errors, produced when projecting points of 3D space in 2D, follow a pattern) to minimize said error; *RANSAC robust Estimation Algorithm* - which allows detecting outliers (the furthest points), in order to achieve a tighter image (corrected image). In short, all these algorithms are necessary for resolution and correction of pictures.

Resumen

Para mí, este trabajo “Problemas de distancia euclídea” ha sido un desafío fascinante ya que abarca muchas ramas de las matemáticas. Combina técnicas de Álgebra, Geometría, Análisis Numérico, Probabilidad e incluso la Parte Computacional (implementación de algoritmos)

Vamos a dar primero una breve descripción de la geometría euclídea y geometría proyectiva.

El objetivo de este trabajo es explicar de los procedimientos de calibración de imágenes, la mayoría de los cuales se basan en el modelo de cámara estenopeica, y este modelo a su vez se basa en la geometría proyectiva.

A continuación, nos centramos en el estudio de la geometría proyectiva que proporciona un modelo lineal de las imágenes del proceso de captación a medida que estudia la relación entre figuras geométricas y su proyección (el ejemplo más común consiste en figuras 3D con proyecciones en un plano 2D). Después, discutimos los conceptos básicos de la geometría proyectiva con los que está asociado el proceso de captura de una imagen con cámaras. Introducimos nociones como: *recta en el infinito* y *puntos ideales.*

Ahora bien, teniendo las herramientas mencionadas anteriormente, nos centramos en dar una representación de un mundo tridimensional en dos dimensiones (gracias a la proyección), es decir, en describir cómo una cámara (estenopecia) crea imágenes. Destacaremos dos principales modelos de cámaras (por modelo de cámara entendemos la proyección de $\mathbb{P}^3 \to \mathbb{P}^2$), donde cada modelo se puede describir mediante una matriz $\mathbf{P}$ de dimensión (3×4) (salvo escalar) de rango máximo. Por un lado, los modelos finitos de cámara que se caracterizan por tener la primera submatriz de $\mathbf{P}$ regular, y, por otro lado, los modelos de cámaras en

el infinito cuya primera submatriz de $\mathbf{P}$ es singular (por ejemplo, la cámara afín). Además, estudiaremos la estructura que tienen las cámaras proyectivas y así como las propiedades que poseen.

Una vez entendido cómo funciona una cámara estenopecia, introducimos el concepto de geometría epipolar (geometría de una visión estéreo: la intersección de dos planos π y π' de la imagen). Dicha geometría se puede describir algebraicamente mediante una matriz fundamental $\mathbf{F}$, y esta a su vez cumple una serie de propiedades relacionadas con los puntos de los planos π y π' (puntos imagen).

Esto nos lleva a plantearnos el siguiente asunto: *¿Cómo encontrar un punto* $\mathbf{X}$ *en 3D que se ajuste de manera óptima a los puntos de imagen medidos, es decir, sus las proyecciones* $\mathbf{x}$ *y* $\mathbf{x}_i'$ *sobre los planos* π *y* π'*?* (ver Fig. 1).

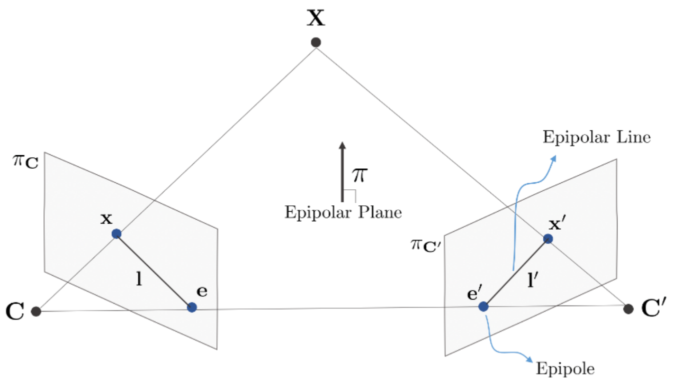


Figure 1. Breve descripción de Geometría Epipolar

Primero, nos basamos en *el teorema de reconstrucción proyectiva* que tiene como objetivo determinar $\mathbf{H}$ (una homografía 2D). Luego, para responder a la pregunta anterior, vamos a dar un algoritmo para resolver el *problema de reconstrucción (triangulación).*

En la práctica, sin embargo, las coordenadas de los puntos imagen $\mathbf{x}$ y $\mathbf{x}_i'$ son difíciles de medir con precisión. Para ello, vamos a describir algunos de los algoritmos más conocidos y exitosos, como por ejemplo: *Transformación Lineal Directa (DLT)*- para reducir los errores; *Probabilidad Máxima Estimacion (MLE)*- (asumiendo que los errores, producidos al proyectar puntos del espacio 3D en 2D, siguen un patrón) para minimizar dicho error; *Algoritmo de Estimación robusto RANSAC*- que permite detectar los valores atípicos (los puntos más alejados), para conseguir de esta manera una imagen más ajustada (imagen corregida). En definitiva, todos estos algoritmos son necesarios para la resolución y corrección de imágenes.

Agradecimientos

¡Quiero agradecer de todo corazón a mi familia por su amor, cariño, apoyo y comprensión que me ofrecen SIEMPRE, y sobretodo durante el tiempo estuve realizando este trabajo, así como la fuerza y el coraje que me inspiraron para terminarlo!

¡Agradezco profundamente a mi tutor, Dr. José Ignacio Cogolludo, por la oportunidad que me dió de trabajar a su lado y haberme guiado hasta llegar a la culminación de este trabajo fin de máster! ¡Gracias Doctor por confiar en mí, por tenerme tanta paciencia, así como ayudarme y apoyarme en los momentos más complicados! ¡SIEMPRE LE ESTARÉ MUY AGRADECIDA!

Y por último, pero no menos importante, ¡GRACIAS a todos: amigos y docentes de la Facultad de Matemáticas de Universidad de Zaragoza, y en especial, a TODOS LOS DOCENTES del Máster Universitario en Modelización e Investigación Matemática, Estadística y Computación!

¡Muchas gracias!

CHAPTER 1

Review of basic concepts

1. Euclidean geometry vs projective geometry

A basic idea between Euclidean geometry and projective geometry is that the first one describes shapes 'as they are', however the second shows objects 'as they appear'.

Euclid, introduces an axiomatic approach to geometry in his book Elements 300 BC. According to it, Euclidean geometry is based on measurements taken on rigid shapes, eg. lengths and angles, hence the notion of shape invariance (under certain transformations called rigid motions).

Projective geometry is less restrictive than Euclidean geometry. It is an inherently non-metric geometry, which means that the facts are independent of any metric structure. Under the projective transformations, the incidence structure and the projective harmonic conjugate relationship are preserved.

In short, a projective geometry is an extension of Euclidean geometry in which the 'direction' of each line is included within the line as an additional 'point', and in which a 'horizon' of directions corresponding to Coplanar lines is considered a 'line'. Therefore, two parallel lines meet on a horizon line by virtue of incorporating the same direction.

Gérard Desargues was the initiator of projective geometry, as he mathematically founded the methods of perspective developed by Renaissance artists. We give a modern introduction to projective spaces in any dimension and focus on 2 and 3-dimensional geometry.

2. Projective geometry

Projective geometry is the study of invariable geometric properties by projection. To create more fundamental geometry than Euclidean geometry, eliminate the distinctions between conics, angles, distance, and parallelism.

This new geometry can be understood in terms of rays of light emanating from a point. In the diagram above, the $\triangle IJK$ drawn on the blue plane would be projected onto the $\triangle LNO$ on the ground. This projection does not preserve either the angles or the lengths of the sides, so the triangle on the ground will have angles and sides of different sizes than the ones on the screen. ("Two triangles are in perspective from a point if and only if they are in perspective from a line"- Desargues theorem).

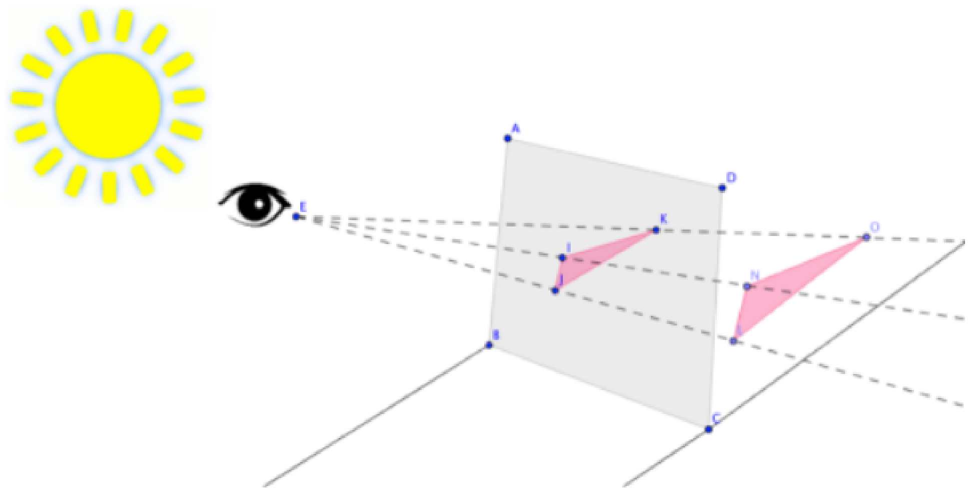


FIGURE 1. Projective geometry and perspective

Why do we need to have a general idea about projective geometry? To understand well the **image calibration procedures**, most of which are based on **the pinhole camera model**, and this model in turn is based on projective geometry.

Projective geometry provides a linear model of the uptake process images as it studies the relationship between geometric figures and their projection. The common example used consists of 3D figures with projections in a 2D plane (see [**1, 2, 3**]).

Now let's start by discussing *the basic concepts of the projective geometry* that the process of capturing an image with cameras are associated with.

3. Projective coordinates in the plane

3.1. The projective plane as equations of affine lines. Consider in $\mathbb{R}^3$ the following equivalence relation:

$$x \sim y \quad \text{if} \quad \exists \lambda \in \mathbb{R} \setminus \{0\} \quad \text{such that } x = \lambda y. \tag{1}$$

The projective plane $\mathbb{P}^2$ is the set of equivalence classes of vectors in $\mathbb{R}^3 \setminus \{(0,0,0)\}$ under the relation above. Later we will see in detail the general projective space $\mathbb{P}^n$.

Relation (1) appears naturally when considering affine lines in $\mathbb{R}^2$ as follows. In affine geometry, any point p in the real plane is represented as a pair of coordinates $(x, y) \in \mathbb{R}^2$. We can also consider $\mathbb{R}^2$ as a vector space in which (x, y) is identified with a vector. Thus, a point is associated with a vector.

A line $r \subset \mathbb{R}^2$ is represented by its affine equation: $ax + by + c = 0$, where different choices of a, b and c give rise to different lines. In this way, a straight line can be represented by a vector: $v_r = (a, b, c)^\top$. This vector represents the projective coordinates of the affine line r. We will follow column notation for vectors. For simplicity and if no ambiguity seems likely to arise, we will denote this vector as $r = (a, b, c)^\top$. However, the same line r can be described using any vector $(\lambda a, \lambda b, \lambda c)^\top$ for $\lambda \neq 0$. Hence, r is naturally identified with the point $[a, b, c]$ in the projective plane.

In projective geometry coordinates are introduced that allow representations multiple, both of points and lines. Thanks to these new coordinates, concepts can be studied as *improper point* and *projective line* analytically.

Note the following properties.

- **Degrees of freedom (dof)**: The number of parameters, in the case of a line is 2, associated with the dimension of the space of lines in $\mathbb{R}^2$.
- To determine **the line through two points** p and p', just consider the notation $\bar{p} = (x, y, 1)$ for $p = (x, y)$. Note that $r = \bar{p} \times \bar{p}'$ describes such a line.
- The **intersection** p **of two lines** r and r' can be obtained from

$$\bar{p} = r \times r'. \tag{2}$$

Note that the last coordinate of $\bar{p}$ is $ab' - a'b$. Therefore this coordinate is 0 if and only if r and r' are parallel. Otherwise, $r \times r' = (X, Y, Z) = (X/Z, Y/Z, 1) = \bar{p} \in \mathbb{P}^2$ since $Z \neq 0$. The point $p = (X/Z, Y/Z)$ is the affine intersection point of r and r'. Note that $r \times r'$ is well defined as a point of $\mathbb{P}^2$.

3.2. The line at infinity and ideal points. Continuing with the previous observation we see that the affine plane $\mathbb{R}^2$ is identified with the subset $U_z = \{(x, y, 1) \mid x, y \in \mathbb{R}^2\} \subset \mathbb{P}^2$. Now, if the lines r and r' were parallel (and different) $r \times r' = (X, Y, 0) \in \mathbb{P}^2$. The set of all points $(X, Y, 0)$ represents the projective line $Z = 0$ that we will call it the **line at infinite** and denote it by ℓ_∞. Its intuitive interpretation is that these points are the "intersection" of two parallel affine lines (see Figure 2). Finally, if $r = r'$, then $r \times r' = (0, 0, 0)$ which does not represent a projective point. The points on ℓ_∞ are called **ideal points**.

Remark 1.1. *The line at infinity can be represented by the vector* $\ell_\infty = (0, 0, 1)^\top$*, which describes the equation* $Z = 0$.

Note that given $r = (a, b, c)$ and $r' = (a, b, c')$, that is, $r \parallel r'$, each of them "intersects" with ℓ_∞ at the same ideal point, which has equations $(b, -a, 0)$.

In order to visualize it, let's consider this example taking the two green parallel lines r and r' in the figure above, their intersection is the green point (its coordinates indicate the direction of these lines). This point is on the infinity line, the blue line.

Furthermore, ℓ_∞ *could be interpreted as the set of directions of lines in the plane.*

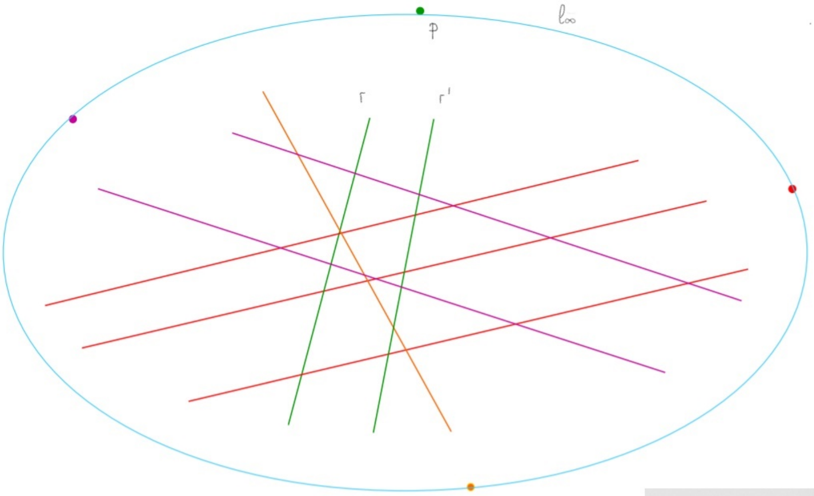


FIGURE 2. Ideal points and line at infinity

That is because $(b, -a)$, in affine notation, is a vector tangent to the line, and orthogonal to (a, b) (which is the normal line). In other words, $(b, -a)$ represents the line's direction. **The ideal point** $(b, -a, 0)$ varies over ℓ_∞, as the line's direction (v_1 and v_2 see Fig. 3 varies).

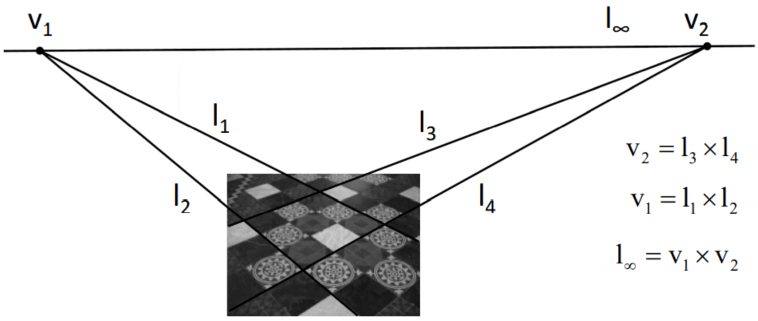


FIGURE 3. Affine Rectification

4. The Projective Space $\mathbb{P}^n$

The projective space is the space of the projective coordinates that have been presented in the previous section.

Definition 1.2. ***The projective space of dimension*** *n, $\mathbb{P}^n$ is a quotient of the set $\mathbb{R}^{n+1} \setminus \{0_{n+1}\}$ by the following relation: $x, y \in \mathbb{R}^{n+1} \setminus \{0_{n+1}\}$, then $x \sim y$ if and only if $\exists \lambda \neq 0$ s. t. $x = \lambda y$.*

4.1. The projective line $\mathbb{P}^1$. The projective space of dimension 1 is known as *projective line.*

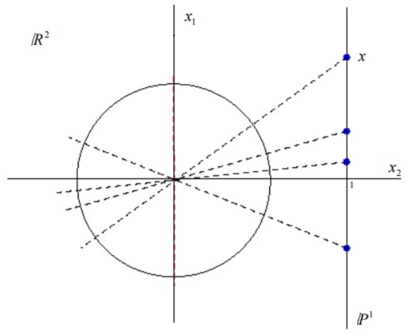


Figure 4. Projective line

Let $x \in \mathbb{P}^1$, whose projective coordinates are $x = [x_1, x_2]$ as shown in Figure 4. If $\ell_\infty : x_2 = 0$, then $\mathbb{P}^1 \setminus \ell_\infty$ can be identified with $\mathbb{R}$ as follows. For any $x \in \mathbb{P}^1 \setminus \ell_\infty$ its Euclidean coordinates can be expressed as $x = x_2[x_1/x_2, 1]$ (see Fig. 4). Hence, each element of the projective line $\mathbb{P}^1$ represents a direction of $\mathbb{R}^2$. The point $x = [x_1, 0] = [1, 0] \in \ell_\infty$, is known as the *improper point* or *point at infinity.*

4.2. A model for the projective plane $\mathbb{P}^2$. As we have already seen, 2D projective geometry is the study of the geometry of $\mathbb{P}^2$. Let's define *the model for the projective plane*:

- Points (resp. lines) in $\mathbb{P}^2$ are identified with lines (resp. planes) through the origin in $\mathbb{R}^3$.
- The set of all vectors $k(x, y, z)$ as k varies forms a line through origin
- Intersecting this set of rays and planes with the plane $z = 1$, one can obtain points and lines.

One of the advantages of projective coordinates is to make points and lines algebraically the same. Thus, the representation in projective coordinates of a

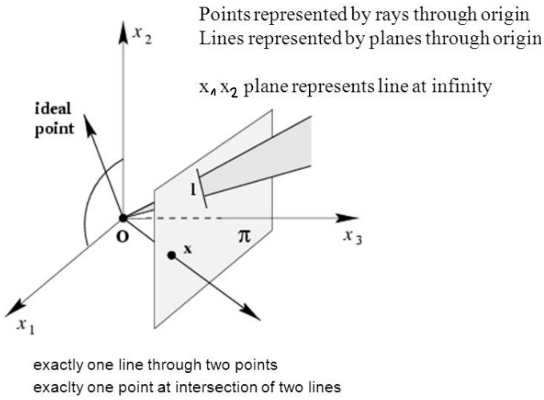


FIGURE 5. A model for the projective plane

point of the plane is formed by a vector of three elements, and the same happens with the representation in projective coordinates of a line of the plane.

PROPOSITION 1.3 (Duality principle). *To any theorem of 2-dimensional projective geometry there corresponds a dual theorem, that can be derived by interchanging the roles of lines and points in the original theorem.*

5. Projective Transformations in a Plane

2D projective geometry is the study of properties of the projective plane $\mathbb{P}^2$ that are invariant under a group of transformations known as projectivities (also known as *projective transformations* or *homographies*).

DEFINITION 1.4. *A projective transformation is an invertible mapping from points in $\mathbb{P}^2$ (that is projective 3-vectors) to points in $\mathbb{P}^2$ that maps lines to lines, in other words,*

$$h : \mathbb{P}^2 \longmapsto \mathbb{P}^2$$

s.t. three points x_1, x_2 and x_3 lie on the same line $\Leftrightarrow$ $h(x_1)$, $h(x_2)$ and $h(x_3)$ do.

Let's see some curiosities of projective transformation. Before that, consider the following **algebraic definition** supported by the following theorem:

THEOREM 1.5. *A mapping $h : \mathbb{P}^2 \to \mathbb{P}^2$ is a projectivity $\Leftrightarrow \exists$ a non-singular (3×3)-matrix $\mathbf{H}$ s.t. for any point in $\mathbb{P}^2$ expressed by $\mathbf{x}$ vector it satisfies that $h(\mathbf{x}) = \mathbf{HX}$, where $\mathbf{H}$ represents a linear transformation*

$$\mathbf{x}' = \mathbf{HX} \qquad \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Moreover, $\mathbf{H}$ is only well defined up to scalar multiplication. The space of such transformations forms a quasi-projective space of dimension 8.

Some characteristics of projective transformation are:

- A projective transformation in $\mathbb{P}^2$ is simply a linear transformation of $\mathbb{R}^3$ up to scalar multiplication.
- In a projective transformation projective subspaces are imaged to projective subspaces.

As it's known, points $\mathbf{x_i}$ are transformed in $\mathbf{x_i'} = \mathbf{Hx_i}$.

Let l be a line and consider $l^\top$ the row vector of coefficients of its defining equation (we use the same notation when no ambiguity can arise). Note that a point $\mathbf{x_i} \in l$ if and only if $l^\top \mathbf{x_i} = 0$. The transformed points $\mathbf{x_i'} = \mathbf{Hx_i}$ (under a projective transformation) lie on the line l', which is given by $l' = \mathbf{H}^{-1} l$.

5.1. Computing Projective Transformation. If we have the coordinates of points on one image and know where they are mapped in the other image, we can compute the mapping between 2 images (Theorem 1.5).

Note that each point provides 2 independent equations:

$$x' = \frac{x_1'}{x_3'} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} = \frac{h_{11}'x + h_{12}'y + h_{13}'}{h_{31}'x + h_{32}'y + 1},$$

$$y' = \frac{x_2'}{x_3'} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} = \frac{h_{21}'x + h_{22}'y + h_{23}'}{h_{31}'x + h_{32}'y + 1}.$$

Also equations are linear in the 8 unknowns $h_{\mathrm{ij}}' = h_{\mathrm{ij}}/h_{33}$.

5.2. Calibration: from projective to Euclidean geometry. Consider $\ell_\infty = \{z = 0\} \subset \mathbb{P}^2$ a line and $\mathbf{x} = (x, y, z) \in \mathbb{P}^2 \setminus \ell_\infty$. We can define

$$\|\mathbf{x}\| = \left(\frac{x}{z}\right)^2 + \left(\frac{y}{z}\right)^2.$$

In other words, the set of equidistant points is given by

$$\mathbb{S}_k = \left\{(x, y, z) \in \mathbb{P}^2 \setminus \ell_\infty \mid \frac{x^2 + y^2}{z^2} = k\right\} = \left\{(x, y, z) \in \mathbb{P}^2 \setminus \ell_\infty \mid \frac{\mathbf{x}^\top C^*_\infty \mathbf{x}}{\ell^\top_\infty \mathbf{x}} = k\right\},$$

where C^*_∞ is the symmetric matrix defining the quadric $C \equiv x^2 + y^2 + z^2 = 0$ on the line ℓ_∞, and $\ell^\top_\infty = (0, 0, 1)$ is the row vector of coefficients of the line ℓ_∞.

This construction can be generalized for any line ℓ_∞ and any quadric C as long as the intersection $C_\infty = \ell_\infty \cap C$ consists of two imaginary points. This defines a distance in $\mathbf{P}^2 \setminus \ell_\infty$. The real quadric C_∞ is called the *absolute quadric*.

Moreover, this can be generalized to any dimension. For any n, define in $\mathbb{P}^n \setminus \pi_\infty$ an *absolute quadric* C^*_∞ as a real quadratic equation in the hyperplane at infinity π_∞ defining an imaginary quadric.

Using the following formal definition for any given $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{P}^n \setminus \pi_\infty$:

$$\langle \mathbf{x}_1, \mathbf{x}_2 \rangle := \frac{\mathbf{x}_1^\top C^*_\infty \mathbf{x}_2}{(\pi^\top_\infty \mathbf{x}_1)\,(\pi^\top_\infty \mathbf{x}_2)},$$

and $\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$, it is easy to check that

$$\langle \mathbf{x}_1, \mathbf{x}_2 \rangle = \|\mathbf{x}_1\| \cdot \|\mathbf{x}_2\| \cdot \cos\theta.$$

A geometrical interpretation of this is the following. Consider $\mathbf{x} \in \mathbb{P}^n \setminus \pi_\infty$, then $\mathbf{x}$ is in the pencil generated by C^*_∞ and π^2_∞. In other words, $\exists \lambda \in \mathbb{R}$ such that

$$\mathbf{x}^\top \left(C^*_\infty - \lambda \pi_\infty \pi^\top_\infty\right) \cdot \mathbf{x} = 0$$

that is

$$\mathbf{x}^\top C^*_\infty \mathbf{x} = \lambda \mathbf{x}^\top \pi_\infty \pi^\top_\infty \mathbf{x} = (\pi^\top_\infty \mathbf{x})^\top (\pi^\top_\infty \mathbf{x}) = (\pi^\top_\infty \mathbf{x})^2.$$

In other words,

$$\lambda = \frac{\mathbf{x}^\top C^*_\infty \mathbf{x}}{(\pi^\top_\infty \mathbf{x})^2} = \|\mathbf{x}\|^2 = \langle \mathbf{x}, \mathbf{x} \rangle.$$

This implies,

$$
\begin{aligned}
\cos\theta =& \frac{\mathbf{x}_1^\top C_\infty^* \mathbf{x}_2}{(\pi_\infty^\top \mathbf{x}_1)\,(\pi_\infty^\top \mathbf{x}_2)} \sqrt{\frac{\left(\pi_\infty^\top \mathbf{x}_1\right)^2}{\mathbf{x}_1^\top C_\infty^* \mathbf{x}_1}} \sqrt{\frac{\left(\pi_\infty^\top \mathbf{x}_2\right)^2}{\mathbf{x}_2^\top C_\infty^* \mathbf{x}_2}} \\
=& \frac{\mathbf{x}_1^\top C_\infty^* \mathbf{x}_2}{\sqrt{(\mathbf{x}_1^\top C_\infty^* \mathbf{x}_1)(\mathbf{x}_2^\top C_\infty^* \mathbf{x}_2)}}.
\end{aligned}
$$

CHAPTER 2

Camera Geometry

Our main objective is to know how a camera creates images, that is, how to represent a three-dimensional world in two-dimensional and it will be possible thanks to the projection.

1. Camera Finite Models

A mapping between the 3D world (object space) and a 2D image is known as a camera. Our principal objective is the central projection. We are going to see different camera models, where by a **camera model** we understand a projection $\mathbb{R}^3 \to \mathbb{R}^2$, or in more generality a projective onto map $\mathbb{P}^3 \to \mathbb{P}^2$ which will be described by a (3×4)-matrix (up to scalar multiplication) of maximal rank.

1.1. Basic pinhole camera. The simplest (common) camera model is the *pinhole camera*. It consists of a central projection from a point C called the **camera center**.

In other words, in a pinhole camera, the camera center C is the point where the projection rays meet at and its distance from the image plane is the **focal length** f.

Consider affine coordinates such that $C = (0, 0, 0)$, $\{Z = f\}$ is the plane of projection, and $p = (0, 0, f)$ is a point on this plane whose adapted affine coordinates are (p_x, p_y), called the principal point. In a pinhole camera, a point $\mathbf{X} = (X_c, Y_c, Z_c)^\top$ in the affine space is projected to the point $(p_x + fX_c/Z_c, p_y + fY_c/Z_c)^\top$ in the image affine coordinate frame (see Figure 1).

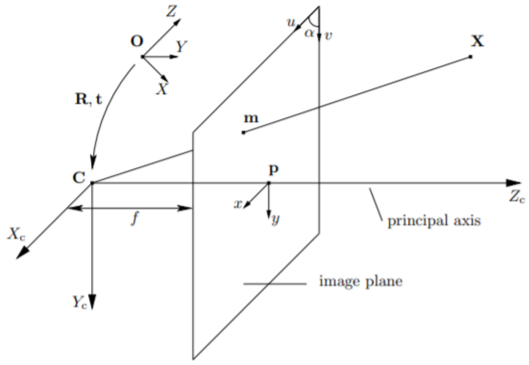


FIGURE 1. **Pinhole camera model**. The camera center: C, is the origin of the camera coordinate frame. The principal point p is the origin of the normalized image coordinate system (x, y), and (u, v) is the pixel image coordinate system.

In order to see this, consider $\{(\lambda X_c, \lambda Y_c, \lambda Z_c) \mid \lambda \in \mathbb{R}\}$ the ray from C to $\mathbf{X}$. Note that if $\lambda_0 := \frac{f}{Z_c}$, then $\lambda_0 \mathbf{X} = \left(\frac{fX_c}{Z_c}, \frac{fY_c}{Z_c}, f\right)$ whose coordinates in the image plane are $(p_x + fX_c/Z_c, p_y + fY_c/Z_c)$.

In terms of homogeneous coordinates the previous projection can be represented by a (3×4)-matrix,

$$\begin{pmatrix} x \\ y \\ t \end{pmatrix} \doteq \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

where $\doteq$ represents equality up to scalar multiplication.

The matrix $\mathbf{K} := \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$ is called the **calibration matrix**.

In general, the camera center does not need to have coordinates $(0, 0, 0)$, for instance when we consider multiple cameras. In this case, the most general projection will have the form:

$$\begin{pmatrix} x \\ y \\ t \end{pmatrix} \doteq \mathbf{KR} \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 1 & -C_z \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \doteq \mathbf{KR}\,[\mathbf{I}_3 \mid -\mathbf{C}] \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}.$$

Note that these camera matrices are very special, because their first minor is non-zero. These cameras are called **finite cameras**. A **general projective camera**, denoted by $\mathbf{P}$ is represented by an arbitrary homogeneous (3×4)-matrix of rank 3.

What happens if the first submatrix is not regular? Such cameras are called **cameras at infinity** and they will be described in the next section.

2. Cameras at infinity

Cameras at infinity appear in case the left-hand (3×3)-submatrix of the camera matrix $\mathbf{P}$ is singular. We will see this means that its center is at infinity.

To prove this, if $\mathbf{P}$ represents a linear map $\mathbb{R}^4 \to \mathbb{R}^3$ of maximal rank, then the center $\mathbf{C}$ is given as the class representing the kernel of this map, that is, $\mathbf{PC} = 0$. If $\mathbf{C}$ is not at infinity, then its last coordinate is non-zero, say $\mathbf{C} = (\mathbf{C_1}, 1)$, where $\mathbf{C_1}$ represents a vector of size 3. Then $\mathbf{PC} = 0$ implies $\mathbf{P}_{1,2,3}\mathbf{C_1} + \mathbf{P}_4 = (0,0,0)$, where $\mathbf{P}_{1,2,3}$ is the left-hand (3×3)-submatrix of the camera matrix $\mathbf{P}$ and $\mathbf{P}_4$ is its last column, that is, $\mathbf{P} = [\mathbf{P}_{1,2,3} \mid \mathbf{P}_4]$. This means that $\mathbf{P}_4$ is a linear combination of the columns of $\mathbf{P}_{1,2,3}$, hence $\operatorname{rank}\mathbf{P}_{1,2,3} = \operatorname{rank}\mathbf{P} = 3$, which means that $\mathbf{P}_{1,2,3}$ is regular.

If $\mathbf{C}$ is on the plane at infinity $\pi_\infty = \{T = 0\}$, then there are coordinates such that $\mathbf{C} = (1,0,0,0) \in \mathbb{P}^3$. Then, the projection of $\mathbf{X} = (X,Y,Z,1)$ on the plane $\{Z = f\}$ can be given as the intersection of the projective line joining $\mathbf{C}$ and $\mathbf{X}$, that is, $\{\lambda(X,Y,Z,1) + \mu(0,0,1,0) \mid (\lambda,\mu) \in \mathbb{P}^1\}$ with the projective plane $Z = fT$. This is $(X,Y,f,1)$ whose adapted coordinates are $(X,Y,1)$. The transformation matrix is:

$$\begin{pmatrix} x \\ y \\ t \end{pmatrix} \doteq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

which implies that the last row of $\mathbf{P}_{1,2,3}$ is zero. In particular, $\mathbf{P}_{1,2,3}$ is singular.

This describes the general idea of an *affine camera*, that is, a camera at infinity where the plane at infinity is projected over the line at infinity of the image.

In general, there could be another type of camera at infinity: the *non-affine* camera, where the plane at infinity is not projected over the line at infinity of the image.

DEFINITION 2.1. *An affine camera is of the form*

$$\mathrm{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

i. e., a camera matrix $\tilde{P}$, whose last row is $p^{3T} = (0,0,0,1)^\top$. In this case, points at infinity are mapped to points at infinity.

In the figure (2) we can see what happens when we apply a cinematographic technique (creating sequences of images that simulate movement) of follow-up while zooming in, thus and so to keep objects of interest remain the same size. As the focal length increases and the distance between the object and the camera also extents, the image remains the same size but the perspective effects reduce.

FIGURE 2. Perspective effects reduce

Let's begin with a finite projective camera given by

$$\mathbf{P}_0 = \mathbf{KR}[\mathrm{I}_3 \mid -\mathbf{C}] = \mathbf{K}\begin{bmatrix} \rho^{1\top} & -\rho^{1\top}\mathbf{C} \\ \rho^{2\top} & -\rho^{2\top}\mathbf{C} \\ \rho^{3\top} & -\rho^{3\top}\mathbf{C} \end{bmatrix}$$

where $\rho^{i\top}$ is the i-th row of $\mathbf{R}$. The principal axis has direction ρ^3, and $d_0 = -\rho^{3\top}\mathbf{C}$ measures the distance of the world origin from the camera center in the direction of the principal axis.

If the camera center is moved to $\mathbf{C} - t\rho^3$, i.e, backwards along the principal ray at a unit speed for a time t, and the camera matrix is (replacing $\mathbf{C}$ by $\mathbf{C} - t\rho^3$ in the previous formula)

$$\mathbf{P}_t = \mathbf{K} \begin{bmatrix} \rho^{1\top} & -\rho^{1\top}\mathbf{C} \\ \rho^{2\top} & -\rho^{2\top}\mathbf{C} \\ \rho^{3\top} & d_t \end{bmatrix}$$

where $d_t = -\rho^{3\top}\mathbf{C} + t$, (the depth of the world origin with respect to the camera center in the direction of $\rho^{3\top}$).

Remark 2.2. *The effect of tracking along the principal ray $\rho^{3\top}$ is to replace the $(3,4)$ entry of the matrix by the depth d_t of the camera center from the world origin.*

Now let's look at zooming where the focal camera's length is inflated by a factor k, that means to multiply the calibration matrix $\mathbf{K}$ on the right by

$$\mathbf{P}_t = \mathbf{K} \begin{bmatrix} d_t/d_0 & 0 & 0 \\ 0 & d_t/d_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \rho^{1\top} & -\rho^{1\top}\mathbf{C} \\ \rho^{2\top} & -\rho^{2\top}\mathbf{C} \\ \rho^{3\top} & d_t \end{bmatrix} \doteq \frac{d_t}{d_0}\mathbf{K} \begin{bmatrix} \rho^{1\top} & -\rho^{1\top}\mathbf{C} \\ \rho^{2\top} & -\rho^{2\top}\mathbf{C} \\ \rho^{3\top} d_0/d_t & d_0 \end{bmatrix}$$

So,

$$\mathbf{P}_\infty = \lim_{t\to\infty} \mathbf{P}_t = \mathbf{K} \begin{bmatrix} \rho^{1\top} & -\rho^{1\top}\mathbf{C} \\ \rho^{2\top} & -\rho^{2\top}\mathbf{C} \\ \mathbf{0}^\top & d_0 \end{bmatrix}$$

Hence, $\mathbf{P}_\infty$ is an instance of an affine camera at infinity (by Definition 2.1). Note that $\mathbf{P}_\infty$ can be descomposed as a product of the two matrices representing the internal camera parameters and external camera parameters.

$$\mathbf{P}_\infty = \begin{bmatrix} \mathbf{K}_{2\times 2} & \hat{0} \\ \hat{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

In the following section we will study the reverse point of view, that is, the intrisic properties of a camera by means of its matrix.

3. Structure and properties of the projective cameras

As we already know, $\mathbf{P}$ (a general projective camera) maps world points $\mathbf{X}$ to image points $\mathbf{x}$ under $\mathbf{x} = \mathbf{P}\mathbf{X}$.

3.1. Structure and properties of the projective cameras.

- **Camera center:** Since $\mathbf{P}$ has 4 columns and maximal rank rank$\mathbf{P} = 3$, the projective camera has a one-dimensional right null-space. In other words, there exists a map $\phi : \mathbb{R}^4 \to \mathbb{R}^3$ and a vector $v \in \mathbb{R}^4$ s.t. $\mathbf{P}v = 0$, that is, $v \in \ker(\phi)$. Let's show that v, a represented as projective 4-vector, is the camera center, and by agreement denote it by $\mathbf{C}$. Take the line l that contains $\mathbf{C}$ and consider a point $\mathbf{A}$ in 3-space. Points on l have the following form: $\mathbf{X}(\lambda) = \lambda\mathbf{A} + (1-\lambda)\mathbf{C}$.

  Points on this line l are projected to $\mathbf{x} = \mathbf{P}\mathbf{X}(\lambda) = \lambda\mathbf{P}\mathbf{A} + (1-\lambda)\mathbf{P}\mathbf{C} = \lambda\mathbf{P}\mathbf{A} + 0 = \lambda\mathbf{P}\mathbf{A}$ (because $\mathbf{P}\mathbf{C} = 0$).

  In other words, all points on l are mapped to the same image point $\mathbf{P}\mathbf{A}$, which means that l must be a ray through $\mathbf{C}$, the camera center.

  Note that the image of camera center is $(0,0,0)^\top = \mathbf{P}\mathbf{C}$, i.e. undefined (the camera center is the unique point in space for which the image is undefined in the projective plane).

  How to calculate the center of the camera? According to the type of camera, we distinguish two cases:

  For finite cameras: $\mathbf{C} = \begin{pmatrix} -\mathbf{P}_{1,2,3}^{-1}\mathbf{P}_4 \\ 1 \end{pmatrix}$

  For infinite cameras: $\mathbf{C} = \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$, $\mathbf{P}_{1,2,3}\mathbf{d} = 0$ (i.e. $\mathbf{d}$ is a non-zero null 3-vector of $\mathbf{P}_{1,2,3}$).
- **Column points**: The columns of the projective camera (denoting the columns of $\mathbf{P}$ by $\mathbf{P}_i$ for $i = 1,2,3,4$) are 3-vectors, and $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ are known as **vanishing points** in the image corresponding to the X, Y and Z axes respectively. Column $\mathbf{P}_4$ is the image of the coordinate origin.

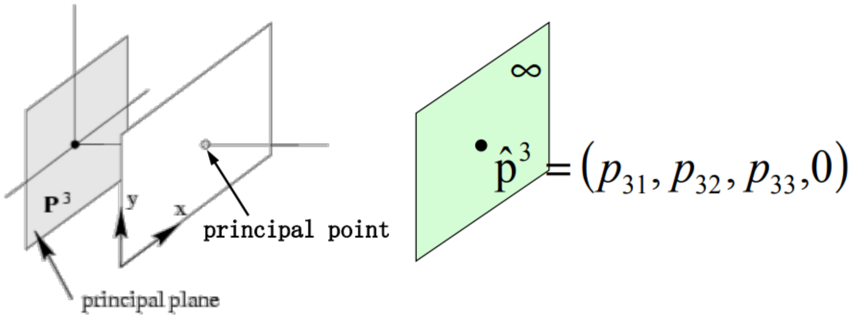


FIGURE 3. The principal point $\hat{p}^3 = (p_{31}, p_{32}, p_{33}, 0)^\top$ of the principal plane $\mathbf{P}^3$ of the camera.

- **The principal plane**: of the camera is $\mathbf{P}^3$, the last row of $\mathbf{P}$ (geometrically, it is the plane through the camera center parallel to the image plane.)
- **Axis planes**: the first and second rows of $\mathbf{P}$ represent planes $\mathbf{P}^1$ and $\mathbf{P}^2$ in space through the camera center C, corresponding to points that map to the image lines $x = 0$ and $y = 0$ respectively.
- **Principal point**: the line passing through C, whose direction is perpendicular to the principal plane $\mathbf{P}^3$, is the principal axis. This axis intersects the image plane at *the principal point.* This point is given by $x_0 = \mathbf{P}_{1,2,3}\tilde{\mathbf{P}}^3$, where $\tilde{\mathbf{P}}^{3\top}$ is the 3^{rd} row of $\mathbf{P}_{1,2,3}$.

- **Principal ray (axis)**: the ray passing across the camera center with direction vector $\tilde{\mathbf{P}}^{3\top}$. The principal axis vector $v = \det(\mathbf{P}_{1,2,3})\tilde{\mathbf{P}}^3$ is oriented in front of the camera if $\det(\mathbf{P}_{1,2,3}) > 0$.

An interesting question in this context is if and how we can determine if a point $\mathbf{X}$ is in front of the camera.

REMARK 2.3. *Let* $\mathbf{X} = (X, Y, Z, T)^\top = (\tilde{\mathbf{X}}^\top, 1)$ *be a* $3D$ *point and* $\mathbf{P} = [\mathbf{P}_{1,2,3} \mid \mathbf{P}_4]$ *be a finite camera matrix. Suppose* $\mathbf{PX} = w(x, y, 1)^\top$*, where* $w = \tilde{\mathbf{P}}^{3\top}(\tilde{\mathbf{X}} - \mathbf{C})$*, (remember that* $\tilde{\mathbf{P}}^3$ *is the* 3^{rd} *column of* $\mathbf{P}_{1,2,3}$*.)*

Then $\text{depth}(\mathbf{X}; \mathbf{P}) = \frac{\text{sign}(\det \mathbf{P}_{1,2,3})w}{T\|\tilde{\mathbf{P}}^3\|}$

Note that depth$(\mathbf{X};\mathbf{P})$ **does not** depend on the particular projective representation of $\mathbf{X}$ and $\mathbf{P}$. To prove it, let's take $\mathbf{X}_\lambda = (\lambda X, \lambda Y, \lambda Z, \lambda T)^\top$ and $\mathbf{P}_\mu = [\mu \mathbf{P}_{1,2,3} | \mu \mathbf{P}_4]$, with $\lambda, \mu \neq 0$. So $\mathbf{P}_\mu \mathbf{X}_\lambda = \mu \lambda \omega (x, y, 1)^\top$.

Then, $\text{depth}(\mathbf{X}_\lambda; \mathbf{P}_\mu) = \frac{\text{sign}(\det(\mu \mathbf{P}_{1,2,3}) \mu \lambda w}{\lambda T \left\| \mu \tilde{\mathbf{P}}^3 \right\|} = \frac{\text{sign}(\det(\mu \mathbf{P}_{1,2,3}) \mu \cancel{\lambda} w}{\cancel{\lambda} T |\mu| \left\| \tilde{\mathbf{P}}^3 \right\|} =$

$$= \begin{cases} \frac{\text{sign}(\det(\mathbf{P}_{1,2,3}) \cancel{\mu} w}{T \cancel{\mu} \left\| \tilde{\mathbf{P}}^3 \right\|}, & \textbf{if } \mu > 0 \\ \frac{\cancel{-}\text{sign}(\det(\mathbf{P}_{1,2,3})) \cancel{-\mu} w}{T \cancel{\mu} \left\| \tilde{\mathbf{P}}^3 \right\|}, & \textbf{if } \mu < 0 \end{cases} = \text{depth}(\mathbf{X}; \mathbf{P})$$

Therefore, depth$(\mathbf{X};\mathbf{P})$ does not depend on the coordinates of $\mathbf{X}$ and $\mathbf{P}$.

CHAPTER 3

Two-View Geometry

Two cameras take a picture of the same scene from different points of view (two perspective views). The relationship between the two resulting views is described using *epipolar geometry*.

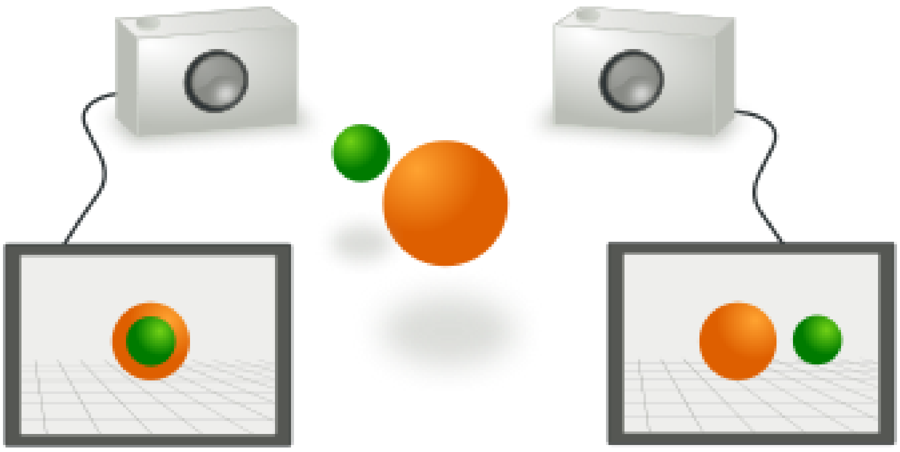


FIGURE 1. Example of epipolar geometry

The epipolar geometry does *not* dependend on scene structure (only depends on the internal parameters of the cameras and relative pose). Let $\mathbf{X}$ be a point in a 3-space; denote by $\mathbf{x}$ its image in the first view and $\mathbf{x}'$ in the second, then the relation between the image points is given by: $(\mathbf{x}')^\top \mathbf{F}\mathbf{x} = 0$, where $\mathbf{F}$, a 3×3 matrix of rank 2, is known as the *fundamental matrix*.

In the following subsection, these concepts are described in more detail.

1. Epipolar geometry and the fundamental matrix F

Definition 3.1. *An epipolar geometry between two views is the geometry of the intersection of the image planes (π, π') with the pencil of planes having the line joining the camera centers* $\mathbf{C}$ *and* $\mathbf{C}'$ *as axis. Epipolar geometry is the geometry of stereo vision.*

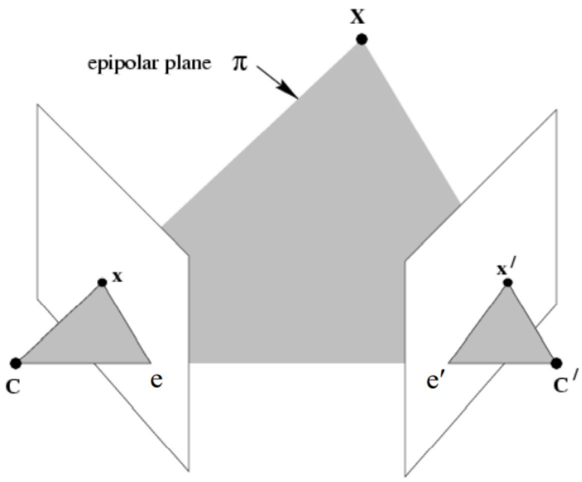


Figure 2. Description 2-view Geometry/ epipolar geometry

The *fundamental elements* of epipolar geometry are the following:

- The *epipolar plane*: π_e is defined by a world point $\mathbf{X}$ and two camera centers $\mathbf{C}$ and $\mathbf{C}'$.
- The *baseline*: l_b is the line joining the camera centers $\mathbf{C}$ and $\mathbf{C}'$.
- The *epipolar line*: l_e (respectively l'_e) is obtained by intersecting an epipolar plane with the image plane π (resp. π').
- The *epipole*: $\mathbf{e}$ (resp. $\mathbf{e}'$) is defined as the intersection of the baseline with the image plane π (resp. π').

Note that all epipolar lines intersect at the epipole. An epipolar plane intersects the left and right image planes in epipolar lines.

Epipolar geometry can be described algebraically by means of the *the fundamental matrix* $\mathbf{F}$.

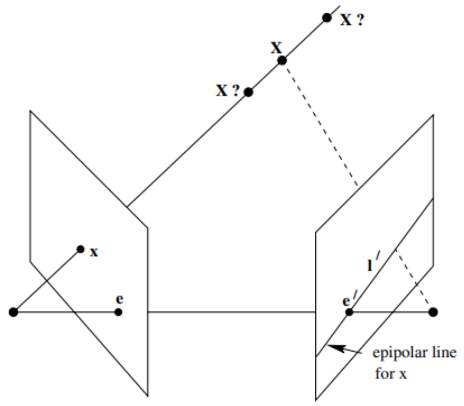


FIGURE 3. point-to-line projective mapping

As we have seen in section 3, an image point $\mathbf{x}$ back-projects to the ray in the 3-space defined by $\mathbf{C}$ (the first camera center), and $\mathbf{x}$. The image of this ray is a line l' in the second view (π'). The world point $\mathbf{X}$, which projects to $\mathbf{x}$, must lie on this ray, hence the image of $\mathbf{X}$ in the second view (π') must lie on the epipolar line $l'_e = l'$.

Hence, the epipolar line is the projection in the second image of the ray from the point $\mathbf{x}$ through the $\mathbf{C}$ of the first camera, therefore, there is a map $\mathbf{x} \mapsto l'_e$ from a point in π (1^{st} image) to its corresponding epipolar line l'_e in π' (2^{nd} image). Observe that this map only depends on the cameras: $\mathbf{P}$ and $\mathbf{P}'$ (not on the structure). This is a *point-to-line projective mapping*, which is represented by the fundamental matrix $\mathbf{F}$. Moreover rank$\mathbf{F} = 2$ since l'_e always contains the epipolar point $\mathbf{e}'$ and thus $\mathbf{F}$ represents a map $\mathbb{P}^2 \to \mathbb{P}^1$, i.e., from 2-dimensional projective space of the first image to 1-dimensional projective space, which is *the pencil of epipolar lines through the epipole* $\mathbf{e}'$).

We are going to describe this mapping in next section.

2. 2D Homography and Geometric derivation of F

We will see how a plane in space $\tilde{\pi}$ can induce a 2D homography, denoted by $H_{\tilde{\pi}}$, between the image planes π and π' such that $\mathbf{x}' = \mathbf{H}_{\tilde{\pi}}\mathbf{x}$, where $\mathbf{x} \in \pi$, $\mathbf{x}' \in \pi'$ via a world point $\mathbf{X} \in \tilde{\pi}$. This means that, the set of points $\mathbf{x}$ in the first image and the corresponding points $\mathbf{x}'$ in the second image are projectively equivalent since they are each projectively equivalent to the planar point set $\mathbf{X}_{\tilde{\pi}}$ (see Figure 4).

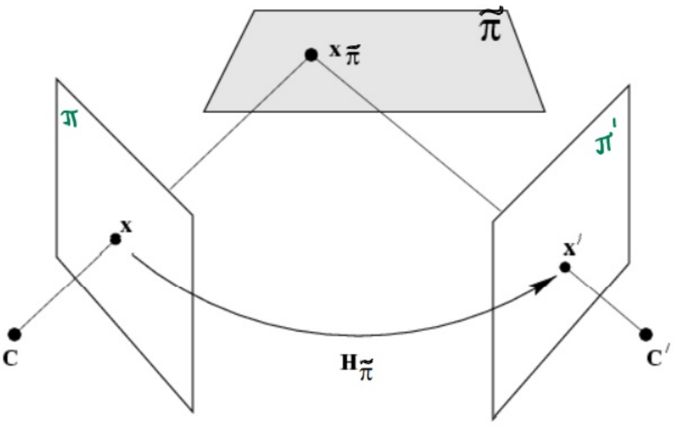


FIGURE 4. Homography between two image planes

This can be constructed as follows. First, take a plane $\tilde{\pi}$ in space s.t. does *not* pass through either of the two camera centers $\mathbf{C}$, $\mathbf{C}'$. The ray through the first camera center $\mathbf{C}$ corresponding to the point $\mathbf{x}$ meets the plane $\tilde{\pi}$ at a point $\mathbf{X}_{\tilde{\pi}}$, and then, this $\mathbf{X}_{\tilde{\pi}}$ is projected to a point $\mathbf{x}'$ in the second image (on the right-hand side in figure 4).

To obtain what is called the *geometric derivation* of $\mathbf{F}$, we need to consider the epipolar line l'_e. This line is obtained by joing the points $\mathbf{x}'$ and $\mathbf{e}'$, i.e. $l'_e = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$. But $\mathbf{x}' = \mathbf{H}_{\tilde{\pi}}\mathbf{x}$ (by the 2D homography), then $l'_e = [\mathbf{e}']_\times \mathbf{H}_{\tilde{\pi}}\mathbf{x}$. This defines the fundamental matrix $\mathbf{F}$ as $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}_{\tilde{\pi}}$ proving the following result.

THEOREM 3.2. *The fundamental matrix* $\mathbf{F}$ *may be written as* $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}_{\tilde{\pi}}$, *where* $\mathbf{H}_{\tilde{\pi}}$ *is the transfer mapping from one image to another via any plane* $\tilde{\pi}$. *Furthermore, since* $[\mathbf{e}']_\times$ *has rank 2 and* $\mathbf{H}_{\tilde{\pi}}$ *rank* 3, $\mathbf{F}$ *is a matrix of rank* 2.

3. Algebraic derivation of F

The purpose of this section is to show a procedure to calculate the fundamental matrix $\mathbf{F}$ knowing the two camera projection matrices $\mathbf{P}$ and $\mathbf{P}'$.

Here is a program made in Sage with the help of my tutor. It is about obtaining (without the need to do any calculation) the fundamental matrix $\mathbf{F}$, knowing $\mathbf{P}_1$ and $\mathbf{P}_2$, two camera matrices.

First, we need to find the center $\mathbf{C}_1$ of the first camera $\mathbf{P}_1$ starting from the minors of the matrix $\mathbf{P}_1$.

```
In [70]: def Centro(P1):
             xP=[]
             for i in range(4):
                 xP=xP+[(-1)^i*det(P1.delete_columns([i]))]
             C1=[]
             if xP[3]!=0:
                 for i in xP:
                     C1=C1+[i/xP[3]]
             else:
                 return "el centro de la cámara está en el infinito"
             return vector(C1)
```

And then, depending on whether or not each of the two cameras are affine or not, we get to obtain the fundamental matrix $\mathbf{F}$.

```
def MatrizFundamental(P1,P2):
    if rank(P1)<3:
       return "primera cámara no es afín"
    if rank(P2)<3:
       return "la segunda cámara no es afín"
    R1=P1.pseudoinverse()
    C1v=Centro(P1)
    e2=P2*C1v
    E2=matrix([[0,-e2[2],e2[1]],[e2[2],0,-e2[0]],[-e2[1],e2[0],0]])
    F=(E2)*P2*R1
    return F
```

4. Properties of F and the epipolar line homography

We recall the definition of the fundamental matrix introduced in section 3.

DEFINITION 3.3. *Given two cameras, the fundamental matrix* $\mathbf{F}$ *is the unique* 3×3 *projective matrix, with rank*$\mathbf{F} = 2$ *satisfying*

$$\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0$$

for all corresponding image points $\mathbf{x} = \mathbf{P}\mathbf{X} \leftrightarrow \mathbf{x}' = \mathbf{P}'\mathbf{X}$.

Let's describe the most important properties of $\mathbf{F}$ the fundamental matrix of a pair of cameras $(\mathbf{P}, \mathbf{P}')$:

- $\mathbf{F}^\top$ is the fundamental matrix of the pair $(\mathbf{P}', \mathbf{P})$.
- for any point $\mathbf{x} \in \pi$ (the first image), the corresponding epipolar line is $l'_e = \mathbf{F}\mathbf{x}$. Similarly, $l_e = \mathbf{F}^\top\mathbf{x}'$ is the epipolar line corresponding to $\mathbf{x}'$.
- for any point $\mathbf{x} \neq \mathbf{e}$ the epipolar line $l'_e = \mathbf{F}\mathbf{x}$ contains the epipole $\mathbf{e}'$. So, $\mathbf{e}'^\top(\mathbf{F}\mathbf{x}) = (\mathbf{e}'^\top\mathbf{F})\mathbf{x} = 0$, $\forall\mathbf{x}$. Hence, $\mathbf{e}'$ is the left null-vector of $\mathbf{F}$, i.e., $\mathbf{e}'^\top\mathbf{F} = 0$. Analogously, $\mathbf{e}$ is the right null-vector of $\mathbf{F}$.
- The set of fundamental matrices forms a projective space of dimension 7. This is because the (3×3)-matrix is defined only up to scalar multiplication and $\det(\mathbf{F}) = 0$, which describes a hypersurface in $\mathbb{P}^8$.
- Any point $\mathbf{x} \in l_e$ is mapped to the same line l'_e, i.e., $\nexists$ inverse mapping and $\mathbf{F}$ is not of full rank.

THEOREM 3.4. *Let* l_e *and* l'_e *be corresponding epipolar lines, and* k *any line not passing through the epipole* $\mathbf{e}$*, then the relation between* l_e *and* l'_e *is:*

$$l'_e = \mathbf{F}[k]_\times l_e \quad \text{and} \quad l_e = \mathbf{F}^\top[k']_\times l'_e.$$

PROOF. Consider k any line that does not passing through the epipole. Note that $[k]_\times l_e$ is a point, moreover, the point of the intersection of the lines l_e and k. Let's denote it by $\mathbf{x}$, since $[k]_\times l_e \in l_e$, the epipolar line (in the first image). Thus, $\mathbf{F}[k]_\times l_e = \mathbf{F}\mathbf{x} = l'_e$ (by 2^{nd} property of the fundamental matrix $\mathbf{F}$.) Analogously, $[k']_\times l'_e$ is the point which lies on the intersection of the lines k' and l'_e. Denotyng that point by $\mathbf{x}'$ (because it lies on the epipolar line l'_e), we obtain $\mathbf{F}^\top[k']_\times l'_e = \mathbf{F}^\top\mathbf{x}' = l_e$ (this last equality comes from the 2^{nd} property of $\mathbf{F}$.)

5. Image reconstruction from two views

In this section we address the question about what can be determined if we only know the image points $\mathbf{x}_i$, $\mathbf{x}'_i$ and their correspondences. Without knowing anything about the calibration (or position of the two cameras $\mathbf{P}$ and $\mathbf{P}'$), we can compute a projective reconstruction of a scene from two views using the following result:

Depending on the context and the data, this is what we mean by reconstruction.

- A reconstruction is the computation of (the coordinates of) a point in a 3D space, starting with the coordinates of its image in a number of cameras whose position is known.
- Projective reconstruction is the computation of the structure of a scene from images (that are taken with uncalibrated cameras), inducing in a scene structure, and camera motion that may differ from the true geometry by an unknown 3D projective transformation.

THEOREM 3.5 (Projective reconstruction). *Suppose that* $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ *is a set of correspondences between points in two images and that the fundamental matrix* $\mathbf{F}$ *is uniquely determined by the condition* $\mathbf{x}'^{\top}_i \mathbf{F} \mathbf{x}_i = 0\ \forall i$. *Let* $(\mathbf{P}_1, \mathbf{P}'_1, \{\mathbf{X}_{1i}\})$ *and* $(\mathbf{P}_2, \mathbf{P}'_2, \{\mathbf{X}_{2i}\})$ *be two reconstructions of the correspondences* $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. *Then* $\exists$ *a non-singular matrix* $\mathbf{H}$ *s.t.* $\mathbf{P}_1 = \mathbf{P}_2\mathbf{H}$, $\mathbf{P}'_1 = \mathbf{P}'_2\mathbf{H}$ *and* $\mathbf{X}_{2i} = \mathbf{H}\mathbf{X}_{1i}$ *for all* i, *except for those* i *s.t.* $\mathbf{F}\mathbf{x}_i = \mathbf{x}'^{\top}_i \mathbf{F} = \mathbf{0}$

See the proof [**7**, p 266]

6. Description of the algorithm for solving the reconstruction problem

Let's describe the process (the projective reconstruction) in other words, using these three steps:

(1) Compute the epipolar geometry (represented by $\mathbf{F}$, the fundamental matrix) from point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$
(2) Now, compute the motion, the cameras $\mathbf{P}$ and $\mathbf{P}'$ from $\mathbf{F}$. We get $\mathbf{P} = [I \mid \mathbf{0}]$, $\mathbf{P}' = [[\mathbf{e}']_{\times}\mathbf{F} \mid \mathbf{e}']$, where $\mathbf{F}^{\top}\mathbf{e}' = 0$ (from the properties of $\mathbf{F}$)
(3) Finally, compute the 3D structure $\mathbf{X}_i$ from $\mathbf{P}$, $\mathbf{P}'$ and the point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ (*triangulation*, see (5)).

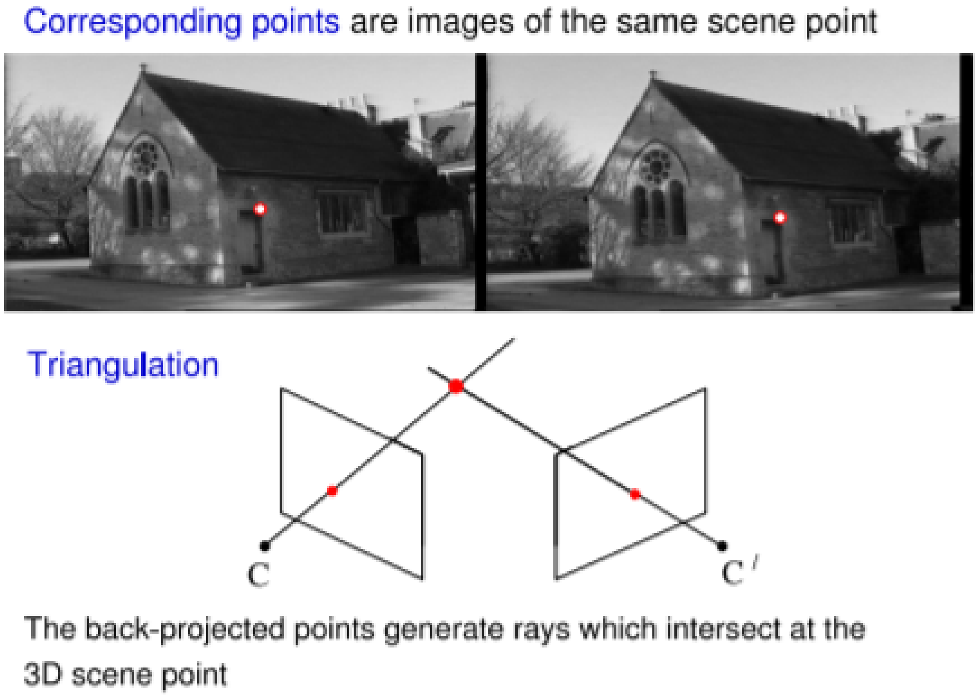


FIGURE 5. Projective reconstruction via triangulation

6.1. Statement of the problem. The term reconstruction refers to the following problem. Given corresponding measured (i.e. approximated or noisy) image points $\mathbf{x}_i$ and $\mathbf{x}'_i$, and (exact) cameras $\mathbf{P}$ and $\mathbf{P}'$, compute the world (3D) point $\mathbf{X}$. In this case, there is a problem: because of the presence of noise, back projected rays *might not* intersect (see (6)).

Now, we are going to describe three solutions from three different points of view.

6.2. The vector solution. The *vector solution* is geometrically discribed in Figure 7. The solution is given by the middle point of the segment minimizing the distance between the back projected rays.

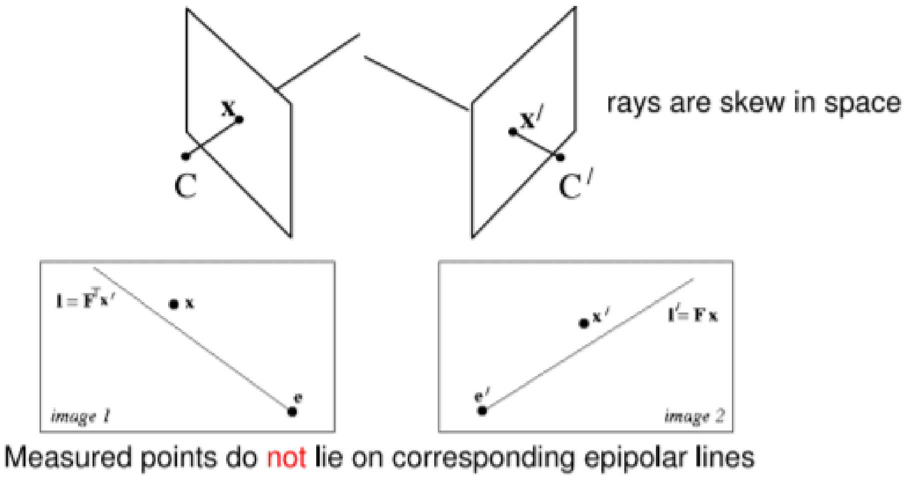


FIGURE 6. A noise problem

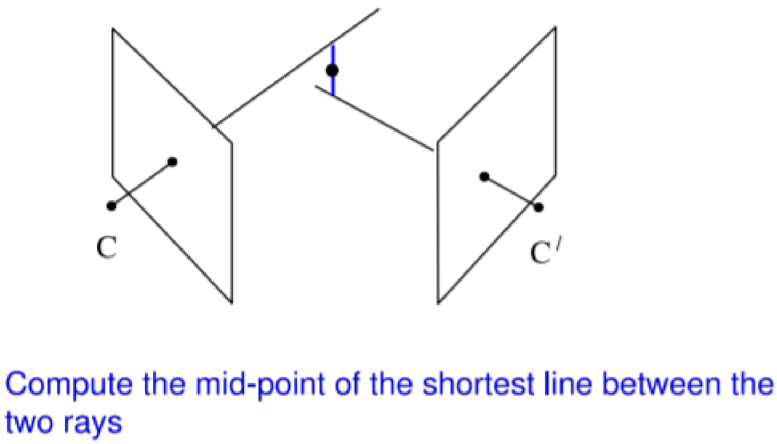


FIGURE 7. A vector solution

6.3. The algebraic solution. Let's describe the linear solution, which is referred to as the *algebraic solution*. To solve for $\mathbf{X}$, use these equations $\mathbf{x} = \mathbf{P}\mathbf{X}$ and $\mathbf{x}' = \mathbf{P}'\mathbf{X}$.

For the 1st camera:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \mathbf{P}^{1\top} \\ \mathbf{P}^{2\top} \\ \mathbf{P}^{3\top} \end{bmatrix}$$

where $\mathbf{P}^{i\top}$ are the rows of $\mathbf{P}$.

- Forming a cross product $\mathbf{x} \times (\mathbf{PX}) = \mathbf{0}$ to eliminate unknown scale in $\lambda\mathbf{x} = \mathbf{PX}$

$$x\left(\mathbf{P}^{3\top}\mathbf{X}\right) - \left(\mathbf{P}^{1\top}\mathbf{X}\right) = 0$$
$$y\left(\mathbf{P}^{3\top}\mathbf{X}\right) - \left(\mathbf{P}^{2\top}\mathbf{X}\right) = 0$$
$$x\left(\mathbf{P}^{2\top}\mathbf{X}\right) - y\left(\mathbf{P}^{1\top}\mathbf{X}\right) = 0$$

- Realign as (first two equations only)

$$\begin{bmatrix} x\mathbf{P}^{3\top} - \mathbf{P}^{1\top} \\ y\mathbf{P}^{3\top} - \mathbf{P}^{2\top} \end{bmatrix} \mathbf{X} = \mathbf{0}$$

Analogously for the 2nd camera:

$$\begin{bmatrix} x'\mathbf{P}'^{3\top} - \mathbf{P}'^{1\top} \\ y'\mathbf{P}'^{3\top} - \mathbf{P}'^{2\top} \end{bmatrix} \mathbf{X} = \mathbf{0}$$

Joining together gives

$$\mathbf{BX} = \mathbf{0}$$

where $\mathbf{B}$ is the (4×4)-matrix

$$\mathbf{B} = \begin{bmatrix} x\mathbf{P}^{3\top} - \mathbf{P}^{1\top} \\ y\mathbf{P}^{3\top} - \mathbf{P}^{2\top} \\ x'\mathbf{P}'^{3\top} - \mathbf{P}'^{1}\top \\ y'\mathbf{P}'^{3\top} - \mathbf{P}'^{2\top} \end{bmatrix}$$

from which $\mathbf{X}$ can be solved up to scale.
Note that, $\mathbf{BX} = \mathbf{0}$ means that $\mathbf{X} \in ker(\phi)$, where ϕ is the map $\phi : \mathbb{R}^4 \to \mathbb{R}^3$. (See subsection 3.1, Chapter 2)

The problem with this approach is that it does not minimize anything with a meaningful geometrically, but it has the clear advantage that it extends to more than two views.

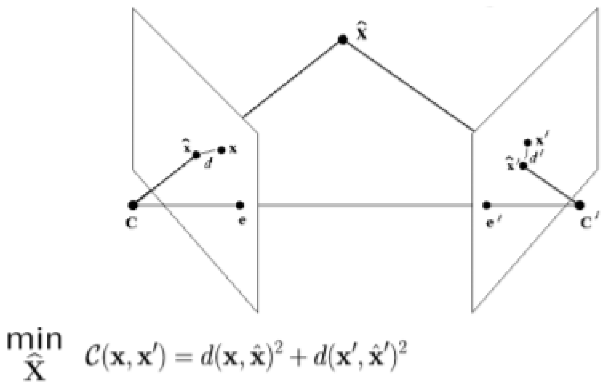
Conclusion: If we have the fundamental matrix $\mathbf{F}$ we can obtain, using (3.5), the matrices $\mathbf{P}$ and $\mathbf{P}'$ in order to construct the rays (see Figure 6) and by the equations $\mathbf{x} = \mathbf{PX}$ and $\mathbf{x}' = \mathbf{P}'\mathbf{X}$, we find the minimum distance between two lines (see Figure 7) whose midpoint is the solution sought $\mathbf{X}$.

6.4. Geometric and Statistical error. Here we only give a general idea of the *geometric-statistical error.* This perspective will be explained in more detail in section 2.

The objective is to estimate a 3D point $\hat{\mathbf{X}}$ that (exactly) satisfies the supplied camera geometry, then it projects as

$$\hat{\mathbf{x}} = \mathbf{P}\hat{\mathbf{X}} \quad \hat{\mathbf{x}}' = \mathbf{P}'\hat{\mathbf{X}}$$

and the purpose is to estimate $\hat{\mathbf{X}}$ from the image measurements $\mathbf{x}$ and $\mathbf{x}'$, minimizing $d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$, where $d(\bullet, \bullet)$ denotes the Euclidean distance (see Figure 6.4).



The statistical machinery comes after assuming that the error follows a certain distribution.

Remark 3.6. *If the measurement noise follows a Gaussian (or Normal) distribution of mean zero,* $\sim N(0, \sigma^2)$*, then the minimizing geometric error is the Maximum Likelihood Estimate of* $\mathbf{X}$.

See the [**7**, p 284-285]

CHAPTER 4

Different algorithms for image resolution and correction

1. Basic DLT algorithm

The purpose of this section is to describe an algorithm to approximate a 2D homography given a set of measured point images using the Direct Linear Transformation (DLT) algorithm. We will also explain what we mean by *approximate*.

1.1. General setting. As we know, 2D homography consists in: given a set of points $\mathbf{x}_i \in \mathbb{P}^2$ and a corresponding set of points $\mathbf{x}'_i$, compute the projective transformation $\mathbf{H}$ such that $\mathbf{H}\mathbf{x}_i = \mathbf{x}'_i$. To determine the 2D homography matrix $\mathbf{H}$, we need solve $\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0}$ for each i. A simple linear solution to $\mathbf{H}$ can be calculated by expressing the transform in terms of a vector cross-product.

Now we denote the j-th row of the matrix $\mathbf{H}$ by $\mathbf{h}^{j\top}$, and $\mathbf{x}_i$ by (x_i, y_i, w_i). Then,

$$\mathbf{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top} \cdot \mathbf{x}_i \\ \mathbf{h}^{2\top} \cdot \mathbf{x}_i \\ \mathbf{h}^{3\top} \cdot \mathbf{x}_i \end{pmatrix} = \begin{pmatrix} h_1^1 & h_2^1 & h_3^1 \\ h_1^2 & h_2^2 & h_3^2 \\ h_1^3 & h_2^3 & h_3^3 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ w_i \end{pmatrix}$$

Denoting $\mathbf{x}'_i$ as $(x'_i, y'_i, w'_i)^\top$ the cross-product is given explicitly by:

$$\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \begin{pmatrix} y'_i\mathbf{h}^{3\top}\mathbf{x}_i - w'_i\mathbf{h}^{2\top}\mathbf{x}_i \\ w'_i\mathbf{h}^{1\top}\mathbf{x}_i - x'_i\mathbf{h}^{3\top}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\top}\mathbf{x}_i - y'_i\mathbf{h}^{1\top}\mathbf{x}_i \end{pmatrix}$$

Since $\mathbf{h}^{j\top}\mathbf{x}_i = \mathbf{x}_i^\top\mathbf{h}^j$ for $j = 1, 2, 3$, we obtain a set of 3 equations for $\mathbf{H}$ that can be written as in the following form:

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \\ -y'_i\mathbf{x}_i^\top & x'_i\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = 0$$

Note that each column has 9 elements and the previous ecuation has the form $\mathbf{A}_i\mathbf{h} = 0$.

When each of the four coordinates being considered is presented in this form we have a set of equations: $\mathbf{A}_i\mathbf{h} = 0$. $\mathbf{A}$ is a (3×9)-matrix; $\mathbf{H}$ is a 9-vector whose entries are those from the matrix $\mathbf{H}$. This equation is linear in the unknown $\mathbf{h}$.

It should be noted that whilst each set of coordinate matches leads us to a set of three equations only two of them are linearly independent. Thus, it is standard practice whilst using the DLT algorithm to ignore the third equation whilst solving for $\mathbf{H}$. The set of equations then becomes:

$$\begin{bmatrix} \mathbf{0}^\top & -w_i'\mathbf{x}_i^\top & -y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & \mathbf{0}^\top & -x_i'\mathbf{x}_i^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = 0.$$

This set up the equation $\mathbf{A}_i\mathbf{h} = 0$, where A_i is a (2×9)-matrix.

Each point correspondence gives 2 equations in the entries of $\mathbf{H}$. With 4 points correspondences, we have $\mathbf{Ah} = 0$ where $\mathbf{A}$ is a (8×9)-matrix. Note that we can determine $\mathbf{H}$ up to scale, or uniquely by setting $\|\mathbf{h}\| = 1$.

1.2. Over-determined systems. If more than four point correspondences are given, $\mathbf{Ah} = \mathbf{0}$ is *over-determined*, and in general there will not be an exact solution (since the measurements are inexact).

Given there is no exact solution to $\mathbf{Ah} = \mathbf{0}$, it seems natural to minimize $\|\mathbf{Ah}\|$ subject to the constraint $\|\mathbf{h}\| = 1$. This is what we referred to as *approximating the solution* at the beginning of the section.

Let $\mathbf{A} = \mathbf{UDV}^\top$ be the SVD descomposition of $\mathbf{A}$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{D}$ is diagonal with the *singular values* of $\mathbf{A}$. We want to minimize $\left\|\mathbf{UDV}^\top\mathbf{h}\right\| = \left\|\mathbf{DV}^\top\mathbf{h}\right\|$ subject to $\|\mathbf{h}\| = \left\|\mathbf{V}^\top\mathbf{h}\right\| = 1$. Since $\mathbf{D}$ is a diagonal matrix whose diagonal contains all the singular values of $\mathbf{A}$, the solution is given by the smallest of these, say $\mathbf{V}^\top\mathbf{h} = (0, \ldots, 0, 1)^\top$. Thus $\mathbf{h}$ is the last column of $\mathbf{V}$.

The resulting algorithm, known as the *basic DLT algorithm*, is summarized in the following steps.

1.3. The Basic DLT algorithm. Given n, 2D to 2D point correspondences $\mathbf{x}_i \mapsto \mathbf{x}'_i$, $i = 1, ..., n$ with $n \geq 4$, determine the 2D-homography matrix $\mathbf{H}$ such that $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$.

The algorithm is briefly described as follows:

(1) For each correspondence, compute the 2×9 matrix $\mathbf{A}_i$.
(2) Merge the $\mathbf{A}_i$'s into a single matrix $\mathbf{A}$ which is a $(2n \times 9)$-matrix.
(3) Obtain the SVD of $\mathbf{A}$. The unit singular vector corresponding to the smallest singular value is the solution $\mathbf{h}$.
(4) Recover $\mathbf{H}$ from $\mathbf{h}$.

2. Statistical cost functions and MLE (Maximum Likelihood Estimation)

To apply this approach we will assume that the image coordinate measurement errors obey a Gaussian probability distribution (possibly after removing the outliers).

We denote by $\Delta\mathbf{x}$ the estimated error, subordinated to a Gaussian distribution with variance σ^2, take $\mathbf{x} = \bar{\mathbf{x}} + \Delta\mathbf{x}$, where $\mathbf{x}$ represents the measured image points and $\bar{\mathbf{x}}$ represents the true values of the points.

Assuming that the noise (on each measurement) is independent, the probability density function of each measured point $\mathbf{x}$ is

$$\Pr(\mathbf{x}) = \left(\frac{1}{2\pi\sigma^2}\right) e^{-d(\mathbf{x},\bar{\mathbf{x}})^2/\left(2\sigma^2\right)}.$$

2.1. Error in only one image. For simplicity, let's consider first the case when the errors are only in the second image. The PDF (the probability density function) of the noise-perturbed data is given by:

$$\Pr\left(\{\mathbf{x}'_i\} \mid \mathbf{H}\right) = \prod_i \left(\frac{1}{2\pi\sigma^2}\right) e^{-d\left(\mathbf{x}'_i,\mathbf{H}\bar{\mathbf{x}}_i\right)^2/\left(2\sigma^2\right)}.$$

The log-likelihood of the set of correspondences is

$$\log \Pr\left(\{\mathbf{x}'_i\} \mid \mathbf{H}\right) = -\frac{1}{2\sigma^2}\sum_i d\left(\mathbf{x}'_i, \mathbf{H}\overline{\mathbf{x}}_i\right)^2 + k,$$

where k is a constant.

The Maximum Likelihood estimate (MLE) of $\mathbf{H}$ (the homography). In other words, it minimizes

$$\sum_i d\left(\mathbf{x}'_i, \mathbf{H}\overline{\mathbf{x}}_i\right)^2.$$

So, MLE is equivalent to minimizing the geometric error function for several points (for two points we have described it in section 6.4).

2.2. Error in both images. Similarly, the PDF of the perturbed data is

$$\Pr\left(\{\mathbf{x}_i, \mathbf{x}'_i\} \mid \mathbf{H}, \{\overline{\mathbf{x}}_i\}\right) = \prod_i \left(\frac{1}{2\pi\sigma^2}\right)^2 e^{-\left(d(\mathbf{x}_i,\overline{\mathbf{x}}_i)^2 + d\left(\mathbf{x}'_i,\mathbf{H}\overline{\mathbf{x}}_i\right)^2\right)/\left(2\sigma^2\right)}$$

Therefore, the MLE of the projective transformation $\mathbf{H}$ and the correspondences, is one that minimizes

$$\sum_i d\left(\mathbf{x}_i, \overline{\mathbf{x}}_i\right)^2 + \sum_i d\left(\mathbf{x}'_i, \mathbf{H}\overline{\mathbf{x}}_i\right)^2.$$

2.3. RANSAC algorithm. A general and successful robust estimator is a RANdom SAmple Consensus (RANSAC). The RANSAC technique counts the number of inliers that are within ϵ of their predicted location (see [**8**, p 9-10]).

Let S be a set that contains outliers. The purpose of this algorithm is to reduce the sample data inliers in order to obtain the minimal data set that determines a robust fit of a model to the data set S.

The RANSAC robust estimation algorithm:

- Calculate the model from a subset, made by selecting randomly a sample of s data points from S.
- Determine the set of data points S_i (the consensus set) which are within a distance threshold t of the model.

- If the number of inliers (the size $\#S_i$ of S_i) is greater than some threshold T, re-estimate the model using all the points in S_i and terminate.
- If $\#S_i < T$, select a new subset and repeat the above.
- After N trials the largest consensus set S_i is selected, and the model is reestimated using all the points in the subset S_i.

In this algorithm there are three constraints that need to be explained: the thresholds t and T, and the number of trial samples N.

2.4. The thresholds t and T. The value t is chosen so that a point whose distance to the model is less than t has a certain probability p_t of being an inlier. Usually t is chosen empirically. However, one can also assume, as mentioned before, that the measurement error follows a normal distribution with zero mean $N(0, \sigma)$.

The other threshold T determines the consensus set. An acceptable rule is one for which the consensus set has a number of points similar to the inliers assumed to be in the original data set. For example, if ϵ is the probability that a point is an outlier and $n = \#S$, then $T = n(1 - \epsilon)$.

2.5. Adaptive algorithm for determining the number of RANSAC samples N. Oftentimes it is not reasonable to calculate the model for all possible choices of s data in the set S. The number of selections N (each of s points) required can be determined in order to ensure that one of the samples contains no outliers with a probability p. For instance, for $p = 0.99$ and given ϵ and s, the number of samples can be obtained as:

$$N = \log(1 - p)/\log(1 - (1 - \epsilon)s). \tag{3}$$

Remark 4.1. *To get an idea of the automatic estimation of a homography between two images using RANSAC, consult* [**7**, p 123].

Here is a sketch of the algorithm:

- If $N = \infty$, then `sample count` $== 0$.
- While $N >$ `sample count` Repeat
  - Choose a sample and count the number of inliers.

- Set $\epsilon = 1-$(number of inliers)/(total number of points)
- Set N from ϵ and (3) with $p = 0.99$.
- Increment the sample count by 1.

- Terminate.

3. An complete algorithm to estimate H from image correspondences

In order to summarize the previous sections, we will describe a complete algorithm for the Maximum Likelihood estimate $\hat{\mathbf{H}}$ of the homography mapping between the images, knowing $n > 4$ image point correspondences.

The MLE also involves solving for a set of subsidiary points $\{\hat{\mathbf{x}}_i\}$, which minimize

$$\sum_i d\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right)^2 + d\left(\mathbf{x}'_i, \hat{\mathbf{x}}'_i\right)^2 .$$

Algorithm:

- Use the linear normalized DLT algorithm, or use RANSAC to compute $\hat{\mathbf{H}}$ from four point correspondences.
- Geometric minimization (of Sampson error). Minimize the Sampson approximation to the geometric error. Either use the Newton algorithm, or the Levenberg-Marquardt algorithm.
- Geometric minimization (of Gold Standard error). Compute an initial estimate of the subsidiary variable $\{\hat{\mathbf{x}}_i\}$ using the measured points $\{\mathbf{x}_i\}$ or the Sampson correction to these points. Then minimize the cost

$$\sum_i d\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right)^2 + d\left(\mathbf{x}'_i, \hat{\mathbf{x}}'_i\right)^2 .$$

Appendix

Geometric visualization of points in space and their projections under a given camera matrix

Steps to take:

(1) I define the elements of the (3×4) camera matrix $\mathbf{P}$, denoting them by $\mathbf{p}_{ij}$ with $i = 1, 2, 3; j = 1, 2, 3, 4$.

(2) Then create the matrix $\mathbf{M} = [\mathbf{P}_{1,2,3}]$ (square, 3×3) from the first three columns of $\mathbf{P}$.

(3) I check that $\mathbf{M}$ is regular, calculating its determinant and denote it by $m = \det(\mathbf{M})$

(4) To get the center $\mathbf{C}$ from the camera, we need the minors of $\mathbf{P}$ (See [**7**, p 163])

(5) For this I consider T the 4^{th} coordinate of $\mathbf{C}$.

(6) The first coordinate of the center of the camera is: $c_1 = e/T$ with $e = \det(E)$. E is the minor of order 3×3, formed by the $2^{nd}, 3^{rd}$ and 4^{th} column of $\mathbf{P}$.

(7) Analogously, we obtain the other coordinates of point $\mathbf{C}$:

$c_2 = f/T$ with $f = \det(F)$, where F is the minor of order 3×3, formed by the $1^{st}, 3^{rd}$ and 4^{th} column of $\mathbf{P}$ and $c_3 = h/T$ with $h = \det(H)$, where H is the minor of order 3×3, formed by the columns: $1^{st}, 2^{nd}$ and 4^{th}.

(8) The principal plane (denoted by PP in SAGE) is given by the 3^{rd} row of $\mathbf{P}$.

(9) We take $\mathbf{C}$ the point (in projective coordinates) of the center of the camera.

(10) $m_2 = \mathbf{P} \cdot \mathbf{C}$ (to check that $\mathbf{C}$, the center of the camera, is the one-dimensional right null space).

(11) I construct g, a line that passes through $\mathbf{C}$ and is perpendicular to the principal plane π

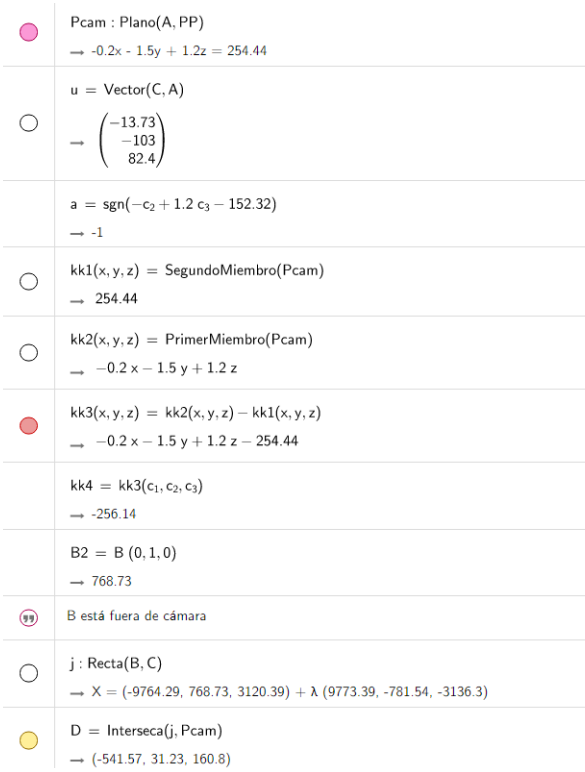
(12) Choose a point $A \in g$.

(13) I define a new $\mathbf{P}_{cam}$ plane, the camera plane, which passes through A and it is parallel to π.

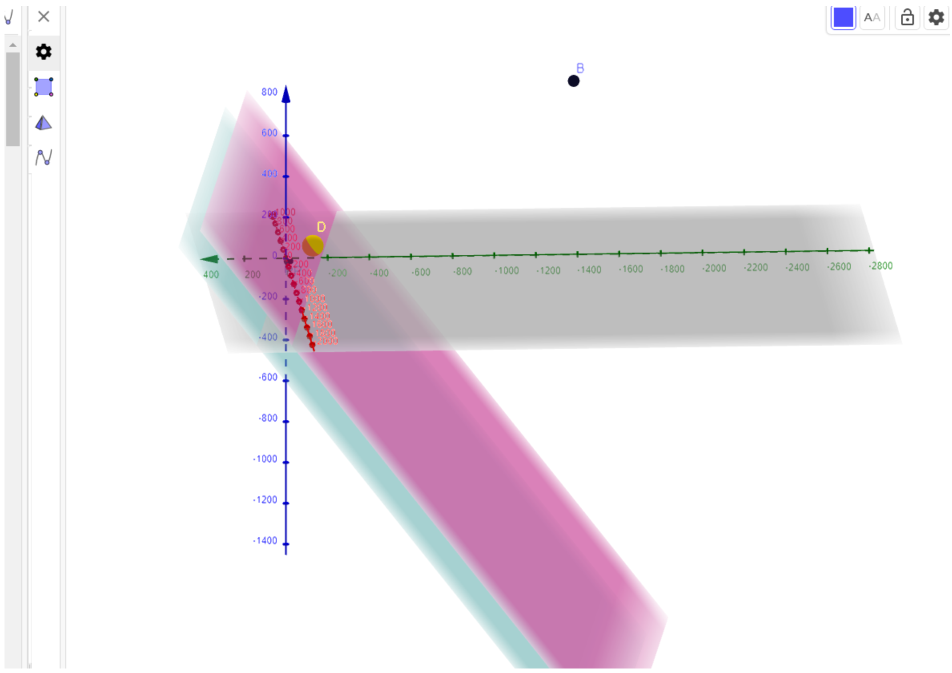
(14) take y, the vector perpendicular to P_c, and passing through points $\mathbf{C}, A$.

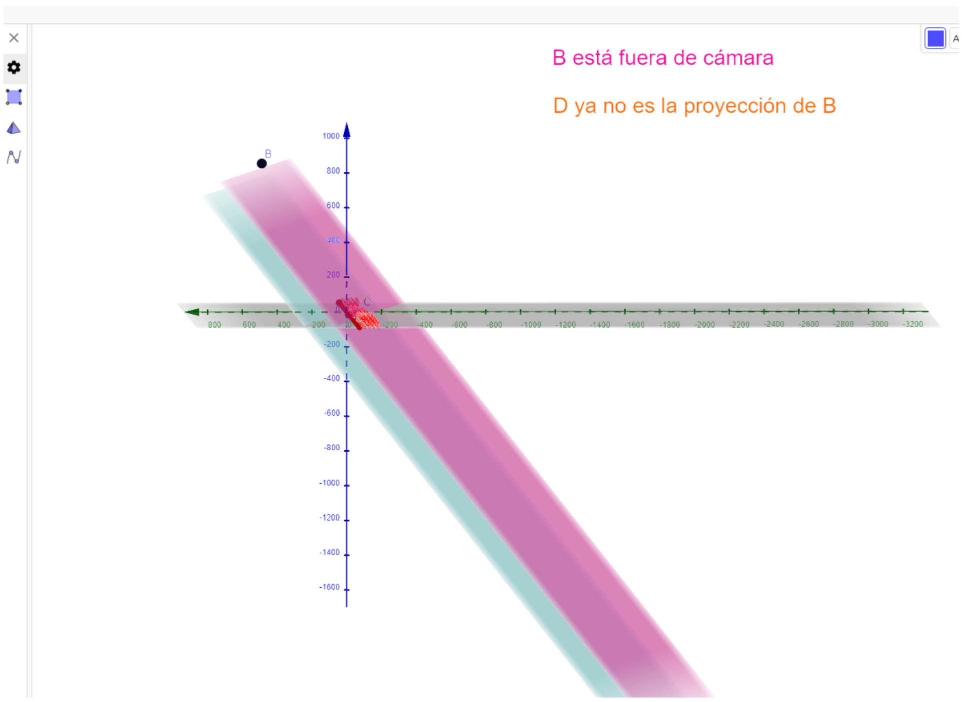
(15) I evaluate the $\mathbf{C}$ in the $\mathbf{P}_{cam}$ plane to see what sign it has $(a = -1)$.

The image below shows point D, the image point of B (when B, a world point, is in front of the projection camera).



The image below shows the case when B, a world point, is NOT front of the projection camera:





All of these algebraic and geometric calculations help us to visualize the process of projection camera.

Following this process in geogebra we get to build and see the famous triangulation problem (it will be seen in detail in the exposition of this final master's work).

Bibliography

[1] H. S. M. Coxeter *Projective Geometry*, University of Toronto, second edition, 1974.

[2] Joseph L. Mundy and Andrew Zisserman *Appendix - Projective Geometry for Machine Vision*
$http://www.cs.cmu.edu/~ph/869/papers/zisser-mundy.pdf$

[3] M. H. Eggar *Pinhole cameras, perspective, and Projective Geometry*, The American Mathematical Monthly
$https://www.jstor.org/stable/2589244$

[4] *Caja de herramientas de calibración de cámara para Matlab*:
$http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html$

[5] David A. Forsyth *Computer Vision. A modern approach*, University of Illinois at Urbana-Champaign

[6] Olivier Faugeras *Stratification of three-dimensional vision: projective, affine, and metric representations*
$https: www.researchgate.net/publication/247654335_Stratification_of_3D_vision_Projective_affine_and_metric_represen$

[7] Andrew Zisserman, Richard Hartley *Multiple View Geometry in Computer Vision*

Cambridge University Press

[8] Richard Szeliski *Image Alignment and Stitching*

[9] Arne Henrichsen *3D Reconstruction and Camera Calibration from 2D Images*
$https://dip.ee.uct.ac.za/publications/theses/MScArne.pdf$