Computer Stereo Vision and Camera geometry

Contents

1 Computer stereo vision **1**

1.1 Outline . . . 1

1.2 Active stereo vision . . . 1

1.2.1 Conventional structured-light vision (SLV) . . . 1

1.2.2 Conventional active stereo vision (ASV) . . . 1

1.2.3 Structured-light stereo (SLS)[4] . . . 1

1.3 Applications . . . 1

1.4 Detailed definition . . . 2

1.4.1 Image Rectification . . . 2

1.4.2 Least squares information measure . . . 2

1.4.3 Other measures of information content . . . 3

1.4.4 Information measure for stereoscopic images . . . 3

1.4.5 Smoothness . . . 3

1.5 Methods of implementation . . . 4

1.6 See also . . . 4

1.7 References . . . 4

1.8 External links . . . 4

2 Projective space **5**

2.1 Introduction . . . 5

2.2 Definition of projective space . . . 6

2.3 Projective space as a manifold . . . 7

2.4 Projective spaces in algebraic geometry . . . 7

2.5 Projective spaces in algebraic topology . . . 8

2.6 Projective space and affine space . . . 8

2.7 Axioms for projective space . . . 8

2.7.1 Classification . . . 9

2.7.2 Finite projective spaces and planes . . . 9

2.8 Morphisms . . . 10

2.9 Dual projective space . . . 10

2.10 Generalizations . . . 10

2.11 See also . . . 11

2.11.1 Generalizations . . . 11

2.11.2 Projective geometry . . . 11
2.11.3 Related . . . 11
2.12 Notes . . . 11
2.13 References . . . 11
2.14 External links . . . 11

3 Cardinal point (optics) 12
3.1 Explanation . . . 12
3.1.1 Focal planes . . . 12
3.1.2 Principal planes and points . . . 13
3.1.3 Nodal points . . . 13
3.1.4 Surface vertices . . . 14
3.2 Modeling optical systems as mathematical transformations . . . 14
3.2.1 Rotationally symmetric optical systems; Optical axis, axial points, and meridional planes . . 14
3.2.2 Ideal, rotationally symmetric, optical imaging system . . . 14
3.2.3 Focal and afocal systems, focal points . . . 15
3.2.4 Transformation . . . 15
3.3 See also . . . 15
3.4 Notes and references . . . 15
3.5 External links . . . 15

4 Triangulation (computer vision) 16
4.1 Introduction . . . 16
4.2 Properties of triangulation methods . . . 17
4.2.1 Singularities . . . 17
4.2.2 Invariance . . . 17
4.2.3 Computational complexity . . . 18
4.3 Some triangulation methods found in the literature . . . 18
4.3.1 Mid-point method . . . 18
4.3.2 Direct linear transformation . . . 18
4.3.3 Via the essential matrix . . . 18
4.3.4 Optimal triangulation . . . 18
4.4 References . . . 18

5 Pinhole camera model 19
5.1 The geometry and mathematics of the pinhole camera . . . 19
5.1.1 Rotated image and the virtual image plane . . . 20
5.2 Homogeneous coordinates . . . 20
5.3 See also . . . 21
5.4 References . . . 21
5.5 Bibliography . . . 21

6 Camera matrix 22

6.1 Derivation . . . 22
6.2 Camera position . . . 22
6.3 Normalized camera matrix and normalized image coordinates . . . 23
6.3.1 The camera position . . . 23
6.4 General camera matrix . . . 24
6.5 See also . . . 24
6.6 References . . . 24

7 Epipolar geometry **25**
7.1 Epipolar geometry . . . 25
7.1.1 Epipole or epipolar point . . . 25
7.1.2 Epipolar line . . . 25
7.1.3 Epipolar plane . . . 25
7.1.4 Epipolar constraint and triangulation . . . 26
7.1.5 Simplified cases . . . 26
7.1.6 Epipolar geometry of pushbroom sensor . . . 26
7.2 See also . . . 26
7.3 References . . . 26
7.4 Further reading . . . 26

8 Fundamental matrix (computer vision) **28**
8.1 Introduction . . . 28
8.2 Projective reconstruction theorem . . . 28
8.2.1 Proof . . . 28
8.3 Derivation of fundamental matrix using coplanarity condition . . . 29
8.4 Properties . . . 29
8.5 See also . . . 29
8.6 Notes . . . 29
8.7 References . . . 29
8.8 Toolboxes . . . 29
8.9 External links . . . 30

9 Essential matrix **31**
9.1 Function . . . 31
9.2 Use . . . 31
9.3 Derivation and definition . . . 31
9.4 Properties of the essential matrix . . . 32
9.5 Estimation of the essential matrix . . . 32
9.6 Determining R and t from E . . . 32
9.6.1 Finding one solution . . . 33
9.6.2 Showing that it is valid . . . 33
9.6.3 Finding all solutions . . . 33

9.7 3D points from corresponding image points . . . 34
9.8 References . . . 34
9.9 Text and image sources, contributors, and licenses . . . 35
9.9.1 Text . . . 35
9.9.2 Images . . . 35
9.9.3 Content license . . . 36

Chapter 1

Computer stereo vision

Computer stereo vision is the extraction of 3D information from digital images, such as obtained by a CCD camera. By comparing information about a scene from two vantage points, 3D information can be extracted by examination of the relative positions of objects in the two panels. This is similar to the biological process Stereopsis.

1.1 Outline

In traditional stereo vision, two cameras, displaced horizontally from one another are used to obtain two differing views on a scene, in a manner similar to human binocular vision. By comparing these two images, the relative depth information can be obtained, in the form of disparities, which are inversely proportional to the differences in distance to the objects.

To compare the images, the two views must be superimposed in a stereoscopic device, the image from the right camera being shown to the observer's right eye and from the left one to the left eye.

In real camera systems however, several pre-processing steps are required.[1]

1. The image must first be removed of distortions, such as barrel distortion to ensure that the observed image is purely projectional.

2. The image must be projected back to a common plane to allow comparison of the image pairs, known as image rectification.

3. An information measure which compares the two images is minimized. This gives the best estimate of the position of features in the two images, and creates a disparity map.

4. Optionally, the disparity as observed by the common projection, is converted back to the height map by inversion. Utilising the correct proportionality constant, the height map can be calibrated to provide exact distances.

1.2 Active stereo vision

The active stereo vision is a form of stereo vision which actively employs a light such as a laser or a structured light to simplify the stereo matching problem. The opposed term is passive stereo vision.

1.2.1 Conventional structured-light vision (SLV)

The conventional structured-light vision (SLV) employs a structured light or laser, and finds projector-camera correspondences.[2][3]

1.2.2 Conventional active stereo vision (ASV)

The conventional active stereo vision (ASV) employs a structured light or laser, however, the stereo matching is performed only for camera-camera correspondences, in the same way as the passive stereo vision.

1.2.3 Structured-light stereo (SLS)[4]

There is a hybrid technique, which utilizes both camera-camera and projector-camera correspondences.[4]

1.3 Applications

3D stereo displays finds many applications in entertainment, information transfer and automated systems. Stereo vision is highly important in fields such as robotics, to extract information about the relative position of 3D objects in the vicinity of autonomous systems. Other applications for robotics include object recognition, where depth information allows for the system to separate occluding image components, such as one chair in front of another, which the robot may otherwise not be able to distinguish as a separate object by any other criteria.

Scientific applications for digital stereo vision include the extraction of information from aerial surveys, for calculation of contour maps or even geometry extraction for 3D building mapping, or calculation of 3D heliographical information such as obtained by the NASA STEREO project.

1.4 Detailed definition

Main article: Triangulation (computer vision)

A pixel records color at a position. The position is iden-

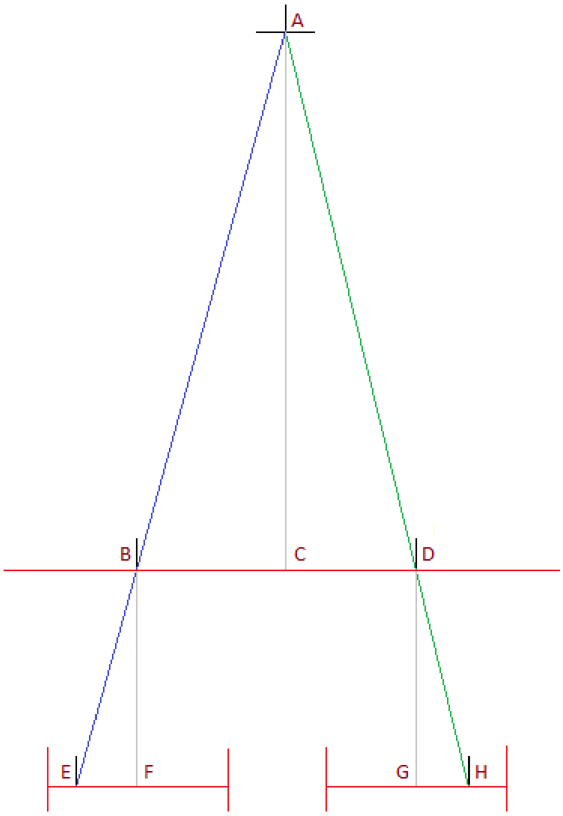


Diagram describing relationship of image displacement to depth with stereoscopic images, assuming flat co-planar images.

tified by position in the grid of pixels (x, y) and depth to the pixel *z*.

Stereoscopic vision gives two images of the same scene, from different positions. In the diagram on the right light from the point *A* is transmitted through the entry points of a pinhole cameras at *B* and *D*, onto image screens at *E* and *H*.

In the attached diagram the distance between the centers of the two camera lens is *BD* = *BC* + *CD*. The triangles are similar,

- *ACB* and *BFE*
- *ACD* and *DGH*

displacement Therefore$d = EF + GH$

$$= BF(\frac{EF}{BF} + \frac{GH}{BF})$$
$$= BF(\frac{EF}{BF} + \frac{GH}{DG})$$
$$= BF(\frac{BC + CD}{AC})$$
$$= BF\frac{BD}{AC}$$
$$= \frac{k}{z}\text{where },$$

- *k* = *BD BF*
- *z* = *AC* is the distance from the camera plane to the object.

So assuming the cameras are level, and image planes are flat on the same plane, the displacement in the y axis between the same pixel in the two images is,

$$d = \frac{k}{z}$$

Where *k* is the distance between the two cameras times the distance from the lens to the image.

The depth component in the two images are z_1 and z_2 , given by,

$$z_2(x, y) = \min\left\{v : v = z_1(x, y - \frac{k}{z_1(x, y)})\right\}$$

$$z_1(x, y) = \min\left\{v : v = z_2(x, y + \frac{k}{z_2(x, y)})\right\}$$

These formulas allow for the occlusion of voxels, seen in one image on the surface of the object, by closer voxels seen in the other image, on the surface of the object.

1.4.1 Image Rectification

Where the image planes are not co-planar image rectification is required to adjust the images as if they were co-planar. This may be achieved by a linear transformation.

The images may also need rectification to make each image equivalent to the image taken from a pinhole camera projecting to a flat plane.

1.4.2 Least squares information measure

The normal distribution is

$$P(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Probability is related to information content described by message length *L*,

$$P(x) = 2^{-L(x)}$$

$$L(x) = -\log_2 P(x)$$

so,

$$L(x, \mu, \sigma) = \log_2(\sigma\sqrt{2\pi}) + \frac{(x-\mu)^2}{2\sigma^2}\log_2 e$$

For the purposes of comparing stereoscopic images, only the relative message length matters. Based on this, the information measure *I*, called the Sum of Squares of Differences (SSD) is,

$$I(x, \mu, \sigma) = \frac{(x-\mu)^2}{\sigma^2}$$

where,

$$L(x, \mu, \sigma) = \log_2(\sigma\sqrt{2\pi}) + I(x, \mu, \sigma)\frac{\log_2 e}{2}$$

1.4.3 Other measures of information content

Because of the cost in processing time of squaring numbers in SSD, many implementations use Sum of Absolute Difference (SAD) as the basis for computing the information measure. Other methods use normalized cross correlation (NCC).

1.4.4 Information measure for stereoscopic images

The least squares measure may be used to measure the information content of the stereoscopic images ,[5] given depths at each point $z(x, y)$. Firstly the information needed to express one image in terms of the other is derived. This is called I_m .

A color difference function should be used to fairly measure the difference between colors. The color difference function is written *cd* in the following. The measure of the information needed to record the color matching between the two images is,

$$I_m(z_1, z_2) = \frac{1}{\sigma_m^2}\sum_{x,y}\mathrm{cd}(\mathrm{color}_1(x, y+\frac{k}{z_1(x,y)}), \mathrm{color}_2(x, y))^2$$

An assumption is made about the smoothness of the image. Assume that two pixels are more likely to be the same color, the closer the voxels they represent are. This measure is intended to favor colors that are similar being grouped at the same depth. For example if an object in front occludes an area of sky behind, the measure of smoothness favors the blue pixels all being grouped together at the same depth.

The total measure of smoothness uses the distance between voxels as an estimate of the expected standard deviation of the color difference,

$$I_s(z_1, z_2) = \frac{1}{2\sigma_h^2}\sum_{i:\{1,2\}}\sum_{x_1,y_1}\sum_{x_2,y_2}\frac{\mathrm{cd}(\mathrm{color}_i(x_1, y_1), \mathrm{color}_i(x_2,}{(x_1-x_2)^2+(y_1-y_2)^2+(z_i(x_1, y_1}$$

The total information content is then the sum,

$$I_t(z_1, z_2) = I_m(z_1, z_2) + I_s(z_1, z_2)$$

The z component of each pixel must be chosen to give the minimum value for the information content. This will give the most likely depths at each pixel. The minimum total information measure is,

$$I_{\text{min}} = \min\{i : i = I_t(z_1, z_2)\}$$

The depth functions for the left and right images are the pair,

$$(z_1, z_2) \in \{(z_1, z_2) : I_t(z_1, z_2) = I_{\text{min}}\}$$

1.4.5 Smoothness

Smoothness is a measure of how similar colors that are close together are. There is an assumption that objects are more likely to be colored with a small number of colors. So if we detect two pixels with the same color they most likely belong to the same object.

The method described above for evaluating smoothness is based on information theory, and an assumption that the influence of the color of a voxel influencing the color of nearby voxels according to the normal distribution on the distance between points. The model is based on approximate assumptions about the world.

Another method based on prior assumptions of smoothness is auto-correlation.

Smoothness is a property of the world. It is not inherently a property of an image. For example an image constructed of random dots would have no smoothness, and inferences about neighboring points would be useless.

Theoretically smoothness, along with other properties of the world should be learnt. This appears to be what the human vision system does.

1.5 Methods of implementation

The minimization problem is NP-complete. This means a general solution to this problem will take a long time to reach a solution. However methods exist for computers based on heuristics that approximate the result in a reasonable amount of time. Also methods exist based on neural networks.[6] Efficient implementation of stereoscopic vision is an area of active research.

1.6 See also

- 3D scanner
- Autostereoscopy
- Epipolar geometry
- Computer vision
- Stereo camera
- Stereoscopic Depth Rendition
- Stereopsis
- Stereophotogrammetry
- 3D reconstruction from multiple images

1.7 References

[1] Bradski, Gary and Kaehler, Adrian. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly.

[2] C. Je, S. W. Lee, and R.-H. Park. High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging. Computer Vision – ECCV 2004, LNCS 3021, pp. 95–107, Springer-Verlag Berlin Heidelberg, May 10, 2004.

[3] C. Je, S. W. Lee, and R.-H. Park. Colour-Stripe Permutation Pattern for Rapid Structured-Light Range Imaging. Optics Communications, Volume 285, Issue 9, pp. 2320-2331, May 1, 2012.

[4] W. Jang, C. Je, Y. Seo, and S. W. Lee. Structured-Light Stereo: Comparative Analysis and Integration of Structured-Light and Active Stereo for Measuring Dynamic Shape. Optics and Lasers in Engineering, Volume 51, Issue 11, pp. 1255-1264, November, 2013.

[5] Lazaros, Nalpantidis; Sirakoulis, Georgios Christou; Gasteratos1, Antonios (2008). "REVIEW OF STEREO VISION ALGORITHMS: FROM SOFTWARE TO HARDWARE". *International Journal of Optomechatronics* **2**: 435–462. doi:10.1080/15599610802438680.

[6] WANG, JUNG-HUA; HSIAO, CHIH-PING (1999). *Proc. Natl. Sci. Counc. ROC(A)* **23** (5): 665–678. Missing or empty |title= (help)

1.8 External links

- Tutorial on uncalibrated stereo vision
- Learn about stereo vision with MATLAB
- Stereo Vision and Rover Navigation Software for Planetary Exploration

Chapter 2

Projective space

In graphical perspective, parallel lines in the plane intersect in a vanishing point on the horizon.

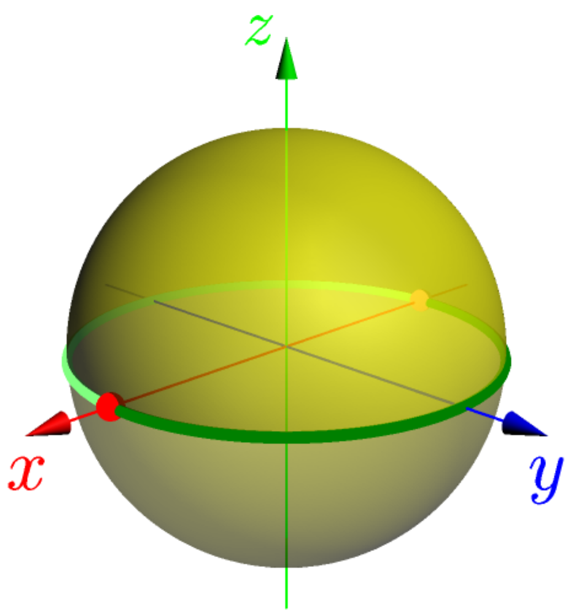
In mathematics, a **projective space** can be thought of as the set of lines through the origin of a vector space V. The cases when $V = \mathbf{R}^2$ and $V = \mathbf{R}^3$ are the real projective line and the real projective plane, respectively, where $\mathbf{R}$ denotes the field of real numbers, $\mathbf{R}^2$ denotes ordered pairs of real numbers, and $\mathbf{R}^3$ denotes ordered triplets of real numbers.

The idea of a projective space relates to perspective, more precisely to the way an eye or a camera projects a 3D scene to a 2D image. All points that lie on a projection line (i.e., a "line-of-sight"), intersecting with the entrance pupil of the camera, are projected onto a common image point. In this case, the vector space is $\mathbf{R}^3$ with the camera entrance pupil at the origin, and the projective space corresponds to the image points.

Projective spaces can be studied as a separate field in mathematics, but are also used in various applied fields, geometry in particular. Geometric objects, such as points, lines, or planes, can be given a representation as elements in projective spaces based on homogeneous coordinates. As a result, various relations between these objects can be described in a simpler way than is possible without homogeneous coordinates. Furthermore, various statements in geometry can be made more consistent and without exceptions. For example, in the standard Euclidian geometry for the plane, two lines always intersect at a point *except* when the lines are parallel. In a projective representation of lines and points, however, such an intersection point exists even for parallel lines, and it can be computed in the same way as other intersection points.

Other mathematical fields where projective spaces play a significant role are topology, the theory of Lie groups and algebraic groups, and their representation theories.

2.1 Introduction



projective space

As outlined above, projective space is a geometric object which formalizes statements like "Parallel lines intersect at infinity". For concreteness, we will give the construction of the real projective plane $\mathbf{P}^2(\mathbf{R})$ in some detail. There are three equivalent definitions:

1. The set of all lines in $\mathbf{R}^3$ passing through the origin (0, 0, 0). Every such line meets the sphere of radius one centered in the origin exactly twice, say in $P = (x, y, z)$ and its antipodal point $(-x, -y, -z)$.

2. $\mathbf{P}^2(\mathbf{R})$ can also be described to be the points on the sphere S^2, where every point P and its antipodal point are not distinguished. For example, the point (1, 0, 0) (red point in the image) is identified with (−1, 0, 0) (light red point), etc.

3. Finally, yet another equivalent definition is the set of equivalence classes of $\mathbf{R}^3 \setminus (0, 0, 0)$, i.e. 3-space without the origin, where two points $P = (x, y, z)$ and $P^* = (x^*, y^*, z^*)$ are equivalent iff there is a nonzero real number λ such that $P = \lambda \cdot P^*$, i.e. $x = \lambda x^*$, $y = \lambda y^*$, $z = \lambda z^*$. The usual way to write an element of the projective plane, i.e. the equivalence class corresponding to an honest point (x, y, z) in $\mathbf{R}^3$, is $[x : y : z]$.

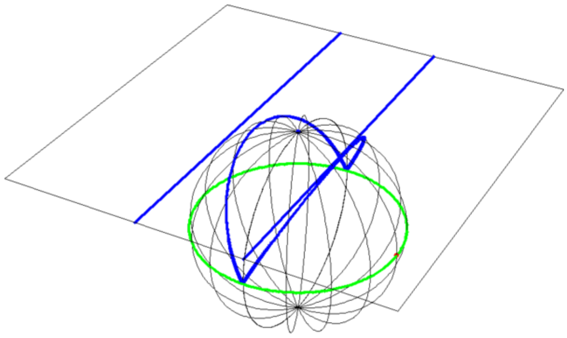
The last formula goes under the name of homogeneous coordinates.

In homogeneous coordinates, any point $[x : y : z]$ with $z \neq 0$ is equivalent to $[x/z : y/z : 1]$. So there are two disjoint subsets of the projective plane: that consisting of the points $[x : y : z] = [x/z : y/z : 1]$ for $z \neq 0$, and that consisting of the remaining points $[x : y : 0]$. The latter set can be subdivided similarly into two disjoint subsets, with points $[x/y : 1 : 0]$ and $[x : 0 : 0]$. In the last case, x is necessarily nonzero, because the origin was not part of $\mathbf{P}^2(\mathbf{R})$. This last point is equivalent to $[1 : 0 : 0]$. Geometrically, the first subset, which is isomorphic (not only as a set, but also as a manifold, as will be seen later) to $\mathbf{R}^2$, is in the image the yellow upper hemisphere (without the equator), or equivalently the lower hemisphere. The second subset, isomorphic to $\mathbf{R}^1$, corresponds to the green line (without the two marked points), or, again, equivalently the light green line. Finally we have the red point or the equivalent light red point. We thus have a disjoint decomposition

$$\mathbf{P}^2(\mathbf{R}) = \mathbf{R}^2 \sqcup \mathbf{R}^1 \sqcup \textit{point}.$$

Intuitively, and made precise below, $\mathbf{R}^1 \sqcup \textit{point}$ is itself the real projective line $\mathbf{P}^1(\mathbf{R})$. Considered as a subset of $\mathbf{P}^2(\mathbf{R})$, it is called *line at infinity*, whereas $\mathbf{R}^2 \subset \mathbf{P}^2(\mathbf{R})$ is called *affine plane*, i.e. just the usual plane.

The next objective is to make the saying "parallel lines meet at infinity" precise. A natural bijection between the plane $z = 1$ (which meets the sphere at the north pole $N = (0, 0, 1)$) and the sphere of the projective plane is accomplished by the gnomonic projection. Each point P



on this plane is mapped to the two intersection points of the sphere with the line through its center and P. These two points are identified in the projective plane. Lines (blue) in the plane are mapped to great circles if one also includes one pair of antipodal points on the equator. Any two great circles intersect precisely in two antipodal points (identified in the projective plane). Great circles corresponding to parallel lines intersect on the equator. So *any* two lines have exactly one intersection point inside $\mathbf{P}^2(\mathbf{R})$. This phenomenon is axiomatized in projective geometry.

2.2 Definition of projective space

The *real projective space* of dimension n or *projective n-space*, $\mathbf{P}^n(\mathbf{R})$, is roughly speaking the set of the lines in $\mathbf{R}^{n+1}$ passing through the origin. For defining it as a topological space and as an algebraic variety it is better to define it as the quotient space of $\mathbf{R}^{n+1}$ by the equivalence relation "to be aligned with the origin". More precisely,

$$\mathbf{P}^n(\mathbf{R}) := (\mathbf{R}^{n+1} \setminus \{\mathbf{0}\}) / \sim,$$

where $\sim$ is the equivalence relation defined by: $(x_0, ..., xn) \sim (y_0, ..., yn)$ if there is a non-zero real number λ such that $(x_0, ..., xn) = (\lambda y_0, ..., \lambda yn)$.

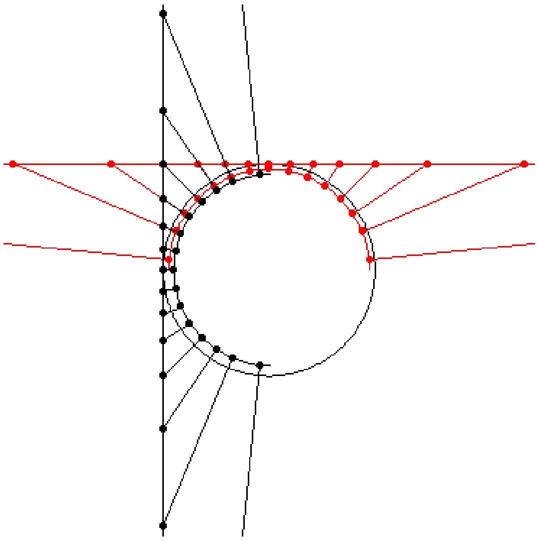
The elements of the projective space are commonly called *points*. The projective coordinates of a point P are x_0, ..., xn, where $(x_0, ..., xn)$ is any element of the corresponding equivalence class. This is denoted $P = [x_0 : ... : xn]$, the colons and the brackets emphasizing that the right-hand side is an equivalence class, which is defined up to the multiplication by a non zero constant.

Instead of $\mathbf{R}$, one may take any field, or even a division ring, K. In these cases it is common[1] to use the notation PG(n, K) for $\mathbf{P}^n(K)$. If K is a finite field of order q, the notation is further simplified to PG(n, q). Taking the complex numbers or the quaternions, one obtains the *complex projective space* $\mathbf{P}^n(\mathbf{C})$ and *quaternionic projective space* $\mathbf{P}^n(\mathbf{H})$.

If n is one or two, it is also called projective line or projective plane, respectively. The complex projective line is also called the Riemann sphere.

Slightly more generally, for a vector space V (over some field k, or even more generally a module V over some division ring), $\mathbf{P}(V)$ is defined to be $(V \setminus \{\mathbf{0}\}) / \sim$, where two non-zero vectors v_1, v_2 in V are equivalent if they differ by a non-zero scalar λ, i.e., $v_1 = \lambda v_2$. The vector space need not be finite-dimensional; thus, for example, there is the theory of projective Hilbert spaces.

2.3 Projective space as a manifold



Manifold structure of the real projective line

The above definition of projective space gives a set. For purposes of differential geometry, which deals with manifolds, it is useful to endow this set with a (real or complex) manifold structure.

Namely, identifying a point of the projective space with its homogeneous coordinates, let us consider the following subsets of the projective space:

$$U_i = \{[x_0 : \cdots : x_n], x_i \neq 0\}, \quad i = 0, \ldots, n.$$

By the definition of projective space, their union is the whole projective space. Furthermore, Ui is in bijection with $\mathbf{R}^n$ (or $\mathbf{C}^n$) via the following maps:

$$[x_0 : \cdots : x_n] \mapsto \left(\frac{x_0}{x_i}, \ldots, \frac{\widehat{x_i}}{x_i}, \ldots, \frac{x_n}{x_i}\right)$$

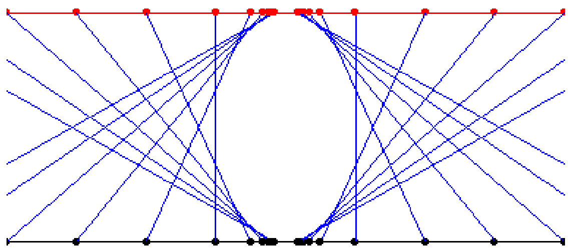
$$[y_0 : \cdots : y_{i-1} : 1 : y_{i+1} : \cdots : y_n] \leftarrow (y_0, \ldots, \widehat{y_i}, \ldots y_n)$$

(the hat means that the i-th entry is missing).

The example image shows $\mathbf{P}^1(\mathbf{R})$. (Antipodal points are identified in $\mathbf{P}^1(\mathbf{R})$, though). It is covered by two copies of the real line $\mathbf{R}$, each of which covers the projective line except one point, which is "the" (or "a") point at infinity.

We first define a topology on projective space by declaring that these maps shall be homeomorphisms, that is, a subset of Ui is open iff its image under the above isomorphism is an open subset (in the usual sense) of $\mathbf{R}^n$. An arbitrary subset A of $\mathbf{P}^n(\mathbf{R})$ is open if all intersections $A \cap Ui$ are open. This defines a topological space.

The manifold structure is given by the above maps, too.



Different visualization of the projective line

Another way to think about the projective line is the following: take two copies of the affine line with coordinates x and y, respectively, and glue them together along the subsets $x \neq 0$ and $y \neq 0$ via the maps

$$x \mapsto \frac{1}{x}, \; y \mapsto \frac{1}{y}.$$

The resulting manifold is the projective line. The charts given by this construction are the same as the ones above. Similar presentations exist for higher-dimensional projective spaces.

The above decomposition in disjoint subsets reads in this generality:

$$\mathbf{P}^n(\mathbf{R}) = \mathbf{R}^n \sqcup \mathbf{R}^{n-1} \sqcup \cdots \sqcup \mathbf{R}^1 \sqcup \mathbf{R}^0,$$

this so-called *cell-decomposition* can be used to calculate the singular cohomology of projective space.

All of the above holds for complex projective space, too. The complex projective line $\mathbf{P}^1(\mathbf{C})$ is an example of a Riemann surface.

2.4 Projective spaces in algebraic geometry

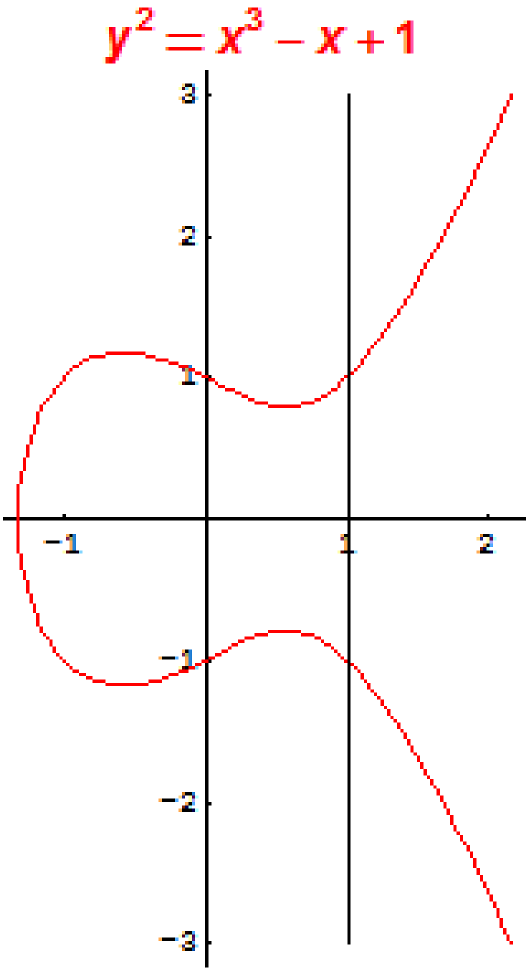
Main article: Algebraic geometry of projective spaces

The covering by the above open subsets also shows that projective space is an algebraic variety (or scheme), it is covered by $n + 1$ affine n-spaces. The construction of projective scheme is an instance of the Proj construction.

2.5 Projective spaces in algebraic topology

Real projective n-space has a quite straightforward CW complex structure. That is, each n-dimensional real projective space has only one n-dimensional cell.

2.6 Projective space and affine space



Example for Bézout's theorem

There are some advantages of the projective space compared with affine space (e.g. $\mathbf{P}^n(\mathbf{R})$ vs. $\mathbf{A}^n(\mathbf{R})$). For these reasons it is important to know when a given manifold or variety is *projective*, i.e. embeds into (is a closed subset of) projective space. (Very) ample line bundles are designed to tackle this question.

Note that a projective space can be formed by the projectivization of a *vector* space, as lines through the origin, but cannot be formed from an *affine* space without a choice of basepoint. That is, affine spaces are open subspaces of projective spaces, which are quotients of vector spaces.

- Projective space is a compact topological space, affine space is not. Therefore, Liouville's theorem applies to show that every holomorphic function on $\mathbf{P}^n(\mathbf{C})$ is constant. Another consequence is, for example, that integration of functions or differential forms on $\mathbf{P}^n$ does not cause convergence issues.

- On a projective complex manifold X, cohomology groups of coherent sheaves are finitely generated. (The above example is $H^0(\mathbf{P}^n(\mathbf{C}), O)$, the zeroth cohomology of the sheaf of holomorphic functions O). In the parlance of algebraic geometry, projective space is proper. The above results hold in this context, too.

- For complex projective space, every complex submanifold $X \subset \mathbf{P}^n(\mathbf{C})$ (i.e., a manifold cut out by holomorphic equations) is necessarily an algebraic variety (i.e., given by *polynomial* equations). This is Chow's theorem, it allows the direct use of algebraic–geometric methods for these ad hoc analytically defined objects.

- As outlined above, lines in $\mathbf{P}^2$ or more generally hyperplanes in $\mathbf{P}^n$ always do intersect. This extends to non-linear objects, as well: appropriately defining the degree of an algebraic curve, which is roughly the degree of the polynomials needed to define the curve (see Hilbert polynomial), it is true (over an algebraically closed field k) that any two projective curves $C_1, C_2 \subset \mathbf{P}^n(k)$ of degree e and f intersect in exactly ef points, counting them with multiplicities (see Bézout's theorem). This is applied, for example, in defining a group structure on the points of an elliptic curve, like $y^2 = x^3 - x + 1$. The degree of an elliptic curve is 3. Consider the line $x = 1$, which intersects the curve (inside affine space) exactly twice, namely in (1, 1) and (1, −1). However, inside $\mathbf{P}^2$, the projective closure of the curve is given by the homogeneous equation

  $$y^2 \cdot z = x^3 - x \cdot z^2 + z^3,$$

  which intersects the line (given inside $\mathbf{P}^2$ by $x = z$) in three points: [1: 1: 1], [1: −1: 1] (corresponding to the two points mentioned above), and [0: 1: 0].

- Any projective group variety, i.e. a projective variety, whose points form an abstract group, is necessarily an abelian variety. Elliptic curves are examples for abelian varieties. The commutativity fails for non-projective group varieties, as the example GL$n(k)$ (the general linear group) shows.

2.7 Axioms for projective space

A **projective space** S can be defined axiomatically as a set P (the set of points), together with a set L of subsets of *P* (the set of lines), satisfying these axioms:[2]

- Each two distinct points *p* and *q* are in exactly one line.
- Veblen's axiom:[3] If *a*, *b*, *c*, *d* are distinct points and the lines through *ab* and *cd* meet, then so do the lines through *ac* and *bd*.
- Any line has at least 3 points on it.

The last axiom eliminates reducible cases that can be written as a disjoint union of projective spaces together with 2-point lines joining any two points in distinct projective spaces. More abstractly, it can be defined as an incidence structure (*P*, *L*, *I*) consisting of a set *P* of points, a set *L* of lines, and an incidence relation *I* stating which points lie on which lines.

The structures defined by these axioms are more general than those obtained from the vector space construction given above. If the (projective) dimension is at least three then, by the Veblen–Young theorem, there is no difference. However, for dimension two there are examples which satisfy these axioms that can not be constructed from vector spaces (or even modules over division rings). These examples do not satisfy the Theorem of Desargues and are known as Non-Desarguesian planes. In dimension one, any set with at least three elements satisfies the axioms, so it is usual to assume additional structure for projective lines defined axiomatically.[4]

It is possible to avoid the troublesome cases in low dimensions by adding or modifying axioms that define a projective space. Coxeter (1969, p. 231) gives such an extension due to Bachmann.[5] To ensure that the dimension is at least two, replace the three point per line axiom above by;

- There exist four points, no three of which are collinear.

To avoid the non-Desarguesian planes, include Pappus's theorem as an axiom;[6]

- If the six vertices of a hexagon lie alternately on two lines, the three points of intersection of pairs of opposite sides are collinear.

And, to ensure that the vector space is defined over a field that does not have even characteristic include *Fano's axiom*;[7]

- The three diagonal points of a complete quadrangle are never collinear.

A **subspace** of the projective space is a subset *X*, such that any line containing two points of *X* is a subset of *X* (that is, completely contained in *X*). The full space and the empty space are always subspaces.

The geometric dimension of the space is said to be *n* if that is the largest number for which there is a strictly ascending chain of subspaces of this form:

$$\varnothing = X_{-1} \subset X_0 \subset \cdots X_n = P.$$

A subspace X_i in such a chain is said to have (geometric) dimension i . Subspaces of dimension 0 are called *points*, those of dimension 1 are called *lines* and so on. If the full space has dimension n then any subspace of dimension $n-1$ is called a hyperplane.

2.7.1 Classification

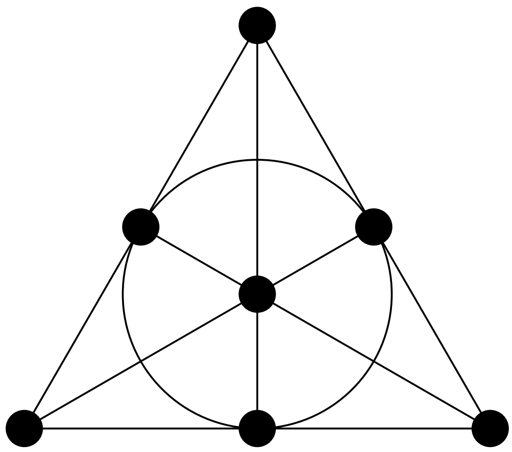
- Dimension 0 (no lines): The space is a single point.
- Dimension 1 (exactly one line): All points lie on the unique line.
- Dimension 2: There are at least 2 lines, and any two lines meet. A projective space for *n* = 2 is equivalent to a projective plane. These are much harder to classify, as not all of them are isomorphic with a PG(*d*, *K*). The Desarguesian planes (those which are isomorphic with a PG(2, *K*)) satisfy Desargues's theorem and are projective planes over division rings, but there are many non-Desarguesian planes.
- Dimension at least 3: Two non-intersecting lines exist. Veblen & Young (1965) proved the Veblen–Young theorem that every projective space of dimension *n* ≥ 3 is isomorphic with a PG(*n*, *K*), the *n*-dimensional projective space over some division ring *K*.

2.7.2 Finite projective spaces and planes

For more details on finite projective planes, see Projective plane § Finite projective planes.

A *finite projective space* is a projective space where *P* is a finite set of points. In any finite projective space, each line contains the same number of points and the *order* of the space is defined as one less than this common number. For finite projective spaces of dimension at least three, Wedderburn's theorem implies that the division ring over which the projective space is defined must be a finite field, GF(*q*), whose order (that is, number of elements) is *q* (a prime power). A finite projective space defined over such a finite field will have *q* + 1 points on a line, so the two concepts of order will coincide. Notationally, PG(*n*, GF(*q*)) is usually written as PG(*n*, *q*).

All finite fields of the same order are isomorphic, so, up to isomorphism, there is only one finite projective space for each dimension greater than or equal to three, over a



The Fano plane

given finite field. However, in dimension two there are non-Desarguesian planes. Up to isomorphism there are

1, 1, 1, 1, 0, 1, 1, 4, 0, … (sequence A001231 in OEIS)

finite projective planes of orders 2, 3, 4, ..., 10, respectively. The numbers beyond this are very difficult to calculate and are not determined except for some zero values due to the Bruck–Ryser theorem.

The smallest projective plane is the Fano plane, PG(2, 2) with 7 points and 7 lines.

2.8 Morphisms

Injective linear maps $T \in L(V, W)$ between two vector spaces V and W over the same field k induce mappings of the corresponding projective spaces $\mathbf{P}(V) \to \mathbf{P}(W)$ via:

$$[v] \to [T(v)],$$

where v is a non-zero element of V and [...] denotes the equivalence classes of a vector under the defining identification of the respective projective spaces. Since members of the equivalence class differ by a scalar factor, and linear maps preserve scalar factors, this induced map is well-defined. (If T is not injective, it will have a null space larger than $\{0\}$; in this case the meaning of the class of $T(v)$ is problematic if v is non-zero and in the null space. In this case one obtains a so-called rational map, see also birational geometry).

Two linear maps S and T in $L(V, W)$ induce the same map between $\mathbf{P}(V)$ and $\mathbf{P}(W)$ if and only if they differ by a scalar multiple, that is if $T = \lambda S$ for some $\lambda \neq 0$. Thus if one identifies the scalar multiples of the identity map with the underlying field K, the set of K-linear morphisms from $\mathbf{P}(V)$ to $\mathbf{P}(W)$ is simply $\mathbf{P}(L(V, W))$.

The automorphisms $\mathbf{P}(V) \to \mathbf{P}(V)$ can be described more concretely. (We deal only with automorphisms preserving the base field K). Using the notion of sheaves generated by global sections, it can be shown that any algebraic (not necessarily linear) automorphism has to be linear, i.e. coming from a (linear) automorphism of the vector space V. The latter form the group $\mathrm{GL}(V)$. By identifying maps that differ by a scalar, one concludes that

$$\mathrm{Aut}(\mathbf{P}(V)) = \mathrm{Aut}(V)/K^\times = \mathrm{GL}(V)/K^\times =: \mathrm{PGL}(V),$$

the quotient group of $\mathrm{GL}(V)$ modulo the matrices which are scalar multiples of the identity. (These matrices form the center of $\mathrm{Aut}(V)$.) The groups PGL are called projective linear groups. The automorphisms of the complex projective line $\mathbf{P}^1(\mathbf{C})$ are called Möbius transformations.

2.9 Dual projective space

When the construction above is applied to the dual space V^* rather than V, one obtains the dual projective space, which can be canonically identified with the space of hyperplanes through the origin of V. That is, if V is n dimensional, then $\mathbf{P}(V^*)$ is the Grassmannian of $n-1$ planes in V.

In algebraic geometry, this construction allows for greater flexibility in the construction of projective bundles. One would like to be able associate a projective space to *every* quasi-coherent sheaf E over a scheme Y, not just the locally free ones. See EGAII, Chap. II, par. 4 for more details.

2.10 Generalizations

dimension The projective space, being the "space" of all one-dimensional linear subspaces of a given vector space V is generalized to Grassmannian manifold, which is parametrizing higher-dimensional subspaces (of some fixed dimension) of V.

sequence of subspaces More generally flag manifold is the space of flags, i.e. chains of linear subspaces of V.

other subvarieties Even more generally, moduli spaces parametrize objects such as elliptic curves of a given kind.

other rings Generalizing to associative rings (rather than fields) yields the projective line over a ring

patching Patching projective spaces together yields projective space bundles.

Severi–Brauer varieties are algebraic varieties over a field *k* which become isomorphic to projective spaces after an extension of the base field *k*.

Another generalization of projective spaces are weighted projective spaces; these are themselves special cases of toric varieties.[8]

2.11 See also

2.11.1 Generalizations

- Grassmannian manifold
- Projective line over a ring
- Space (mathematics)

2.11.2 Projective geometry

- projective transformation
- projective representation

2.11.3 Related

- Geometric algebra

2.12 Notes

[1] Mauro Biliotti, Vikram Jha, Norman L. Johnson (2001) *Foundations of Translation Planes*, p 506, Marcel Dekker ISBN 0-8247-0609-9

[2] Beutelspacher & Rosenbaum 1998, pgs. 6–7

[3] also referred to as the *Veblen–Young axiom* and mistakenly as the axiom of Pasch (Beutelspacher & Rosenbaum 1998, pgs. 6–7). Pasch was concerned with real projective space and was attempting to introduce order, which is not a concern of the Veblen–Young axiom.

[4] Baer 2005, p. 71

[5] Bachmann, F. (1959), *Aufbau der Geometrie aus dem Spiegelsbegriff*, Grundlehren der mathematischen Wissenschaftern, 96, Berlin: Springer, pp. 76–77

[6] As Pappus's theorem implies Desargues's theorem this eliminates the non-Desarguesian planes and also implies that the space is defined over a field (and not a division ring).

[7] This restriction allows the real and complex fields to be used (zero characteristic) but removes the Fano plane and other planes that exhibit atypical behavior.

[8] Mukai 2003, example 3.72

2.13 References

- Afanas'ev, V.V. (2001), "projective space", in Hazewinkel, Michiel, *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4
- Baer, Reinhold (2005) [first published 1952], *Linear Algebra and Projective Geometry*, Dover, ISBN 978-0-486-44565-6
- Beutelspacher, Albrecht; Rosenbaum, Ute (1998), *Projective geometry: from foundations to applications*, Cambridge University Press, ISBN 978-0-521-48277-6, MR 1629468
- Coxeter, Harold Scott MacDonald (1974), *Introduction to Geometry*, New York: John Wiley & Sons, ISBN 0-471-18283-4
- Coxeter, Harold Scott MacDonald (1969), *Projective geometry*, Toronto, Ont.: University of Toronto Press, ISBN 0-8020-2104-2, MR 0346652, OCLC 977732
- Dembowski, P. (1968), *Finite geometries*, Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 44, Berlin, New York: Springer-Verlag, ISBN 3-540-61786-8, MR 0233275
- Greenberg, M.J.; *Euclidean and non-Euclidean geometries*, 2nd ed. Freeman (1980).
- Hartshorne, Robin (1977), *Algebraic Geometry*, Berlin, New York: Springer-Verlag, ISBN 978-0-387-90244-9, MR 0463157, esp. chapters I.2, I.7, II.5, and II.7
- Hilbert, D. and Cohn-Vossen, S.; *Geometry and the imagination*, 2nd ed. Chelsea (1999).
- Mukai, Shigeru (2003), *An Introduction to Invariants and Moduli*, Cambridge Studies in Advanced Mathematics, Cambridge University Press, ISBN 978-0-521-80906-1
- Veblen, Oswald; Young, John Wesley (1965), *Projective geometry. Vols. 1, 2*, Blaisdell Publishing Co. Ginn and Co. New York-Toronto-London, MR 0179666 (Reprint of 1910 edition)

2.14 External links

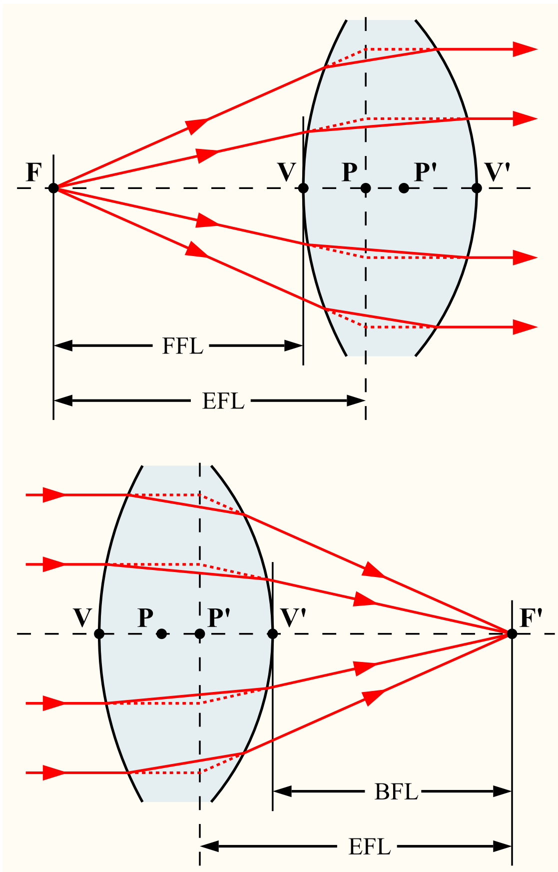
- Weisstein, Eric W., "Projective Space", *MathWorld*.
- http://planetmath.org/encyclopedia/ProjectiveSpace.html
- Projective Planes of Small Order

Chapter 3

Cardinal point (optics)

For other uses, see Cardinal point (disambiguation).

In Gaussian optics, the **cardinal points** consist of three pairs of points located on the optical axis of a rotationally symmetric, focal, optical system. These are the **focal points**, the **principal points**, and the **nodal points**.[1] For *ideal* systems, the basic imaging properties such as image size, location, and orientation are completely determined by the locations of the cardinal points; in fact only four points are necessary: the focal points and either the principal or nodal points. The only ideal system that has been achieved in practice is the plane mirror,[2] however the cardinal points are widely used to *approximate* the behavior of real optical systems. Cardinal points provide a way to analytically simplify a system with many components, allowing the imaging characteristics of the system to be approximately determined with simple calculations.



The cardinal points of a thick lens in air.
F, F' *front and rear focal points,*
P, P' *front and rear principal points,*
V, V' *front and rear surface vertices.*

3.1 Explanation

The cardinal points lie on the optical axis of the optical system. Each point is defined by the effect the optical system has on rays that pass through that point, in the paraxial approximation. The paraxial approximation assumes that rays travel at shallow angles with respect to the optical axis, so that $\sin\theta \approx \theta$ and $\cos\theta \approx 1$.[3] Aperture effects are ignored: rays that do not pass through the aperture stop of the system are not considered in the discussion below.

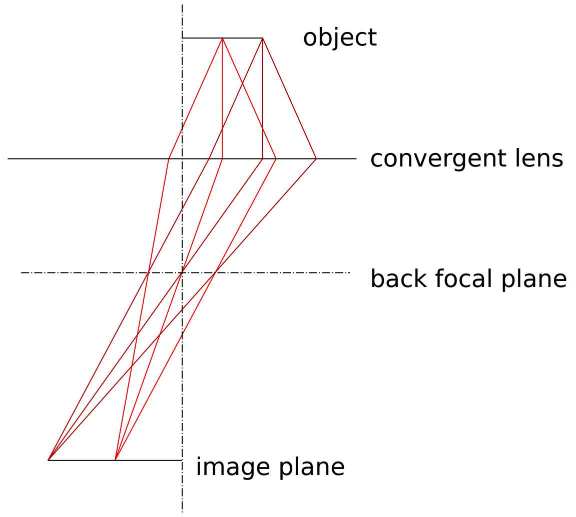
3.1.1 Focal planes

See also: Focus (optics) and Focal length

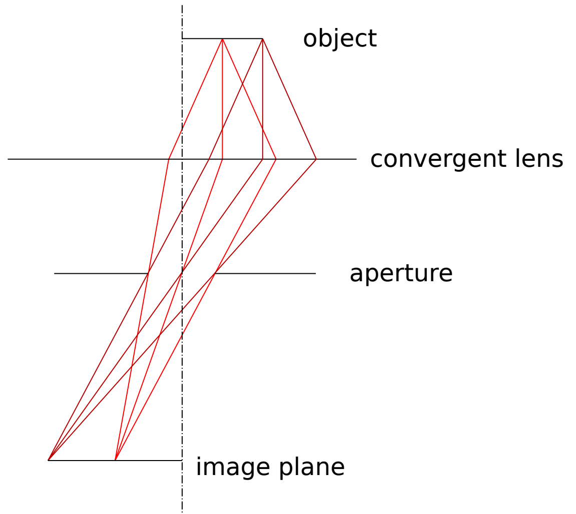
The front focal point of an optical system, by definition, has the property that any ray that passes through it will emerge from the system parallel to the optical axis. The rear (or back) focal point of the system has the reverse property: rays that enter the system parallel to the optical axis are focused such that they pass through the rear focal point.

The front and rear (or back) focal *planes* are defined as the planes, perpendicular to the optic axis, which pass through the front and rear focal points. An object infinitely far from the optical system forms an image at the rear focal plane. For objects a finite distance away, the image is formed at a different location, but rays that leave the object parallel to one another cross at the rear focal plane.

A diaphragm or “stop” at the rear focal plane can be used to filter rays by angle, since:



Rays that leave the object with the same angle cross at the back focal plane.



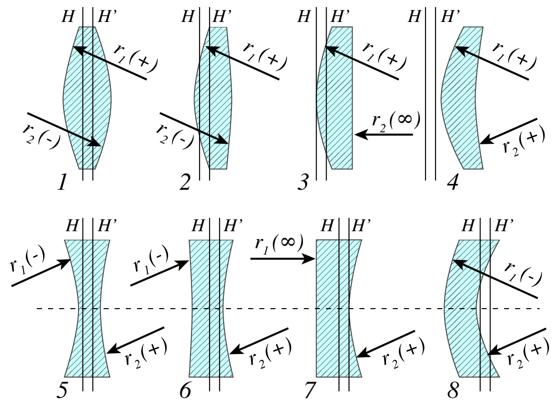
Angle filtering with an aperture at the rear focal plane.

1. It only allows rays to pass that are emitted at an angle (relative to the optical axis) that is sufficiently small. (An infinitely small aperture would only allow rays that are emitted along the optical axis to pass.)
2. No matter where on the object the ray comes from, the ray will pass through the aperture as long as the angle at which it is emitted from the object is small enough.

Note that the aperture must be centered on the optical axis for this to work as indicated. Using a sufficiently small aperture in the focal plane will make the lens telecentric.

Similarly, the allowed range of angles on the output side of the lens can be filtered by putting an aperture at the front focal plane of the lens (or a lens group within the overall lens). This is important for DSLR cameras having CCD sensors. The pixels in these sensors are more sensitive to rays that hit them straight on than to those that strike at an angle. A lens that does not control the angle of incidence at the detector will produce pixel vignetting in the images.

3.1.2 Principal planes and points



Various lens shapes, and the location of the principal planes.

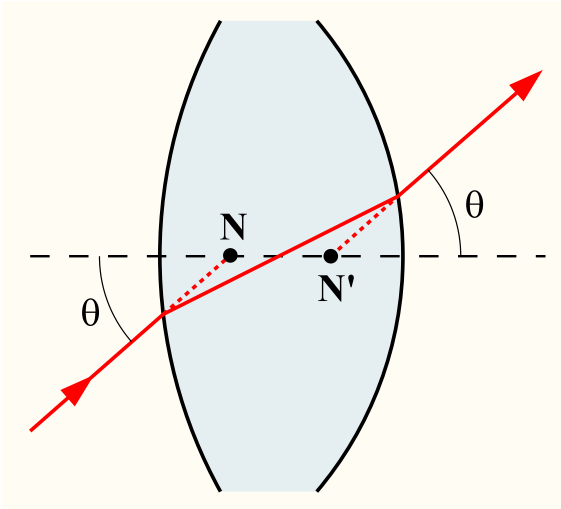
The two principal planes have the property that a ray emerging from the lens *appears* to have crossed the rear principal plane at the same distance from the axis that that ray *appeared* to cross the front principal plane, as viewed from the front of the lens. This means that the lens can be treated as if all of the refraction happened at the principal planes. The principal planes are crucial in defining the optical properties of the system, since it is the distance of the object and image from the front and rear principal planes that determines the magnification of the system. The *principal points* are the points where the principal planes cross the optical axis.

If the medium surrounding the optical system has a refractive index of 1 (e.g., air or vacuum), then the distance from the principal planes to their corresponding focal points is just the focal length of the system. In the more general case, the distance to the foci is the focal length multiplied by the index of refraction of the medium.

For a thin lens in air, the principal planes both lie at the location of the lens. The point where they cross the optical axis is sometimes misleadingly called the **optical centre** of the lens. Note, however, that for a real lens the principal planes do not necessarily pass through the centre of the lens, and in general may not lie inside the lens at all.

3.1.3 Nodal points

The front and rear nodal points have the property that a ray aimed at one of them will be refracted by the lens such that it appears to have come from the other, and with the same angle with respect to the optical axis. The nodal points therefore do for angles what the principal planes do for transverse distance. If the medium on both sides of



N, N' The front and rear nodal points of a thick lens.

the optical system is the same (e.g., air), then the front and rear nodal points coincide with the front and rear principal points, respectively.

The nodal points are widely misunderstood in photography, where it is commonly asserted that the light rays "intersect" at "the nodal point", that the iris diaphragm of the lens is located there, and that this is the correct pivot point for panoramic photography, so as to avoid parallax error.[4][5][6] These claims generally arise from confusion about the optics of camera lenses, as well as confusion between the nodal points and the other cardinal points of the system. (A better choice of the point about which to pivot a camera for panoramic photography can be shown to be the centre of the system's entrance pupil.[4][5][6] On the other hand, swing-lens cameras with fixed film position rotate the lens about the rear nodal point to stabilize the image on the film.[6][7])

3.1.4 Surface vertices

The surface vertices are the points where each optical surface crosses the optical axis. They are important primarily because they are the physically measurable parameters for the position of the optical elements, and so the positions of the cardinal points must be known with respect to the vertices to describe the physical system.

In anatomy, the surface vertices of the eye's lens are called the anterior and posterior *poles* of the lens.[8]

3.2 Modeling optical systems as mathematical transformations

In geometrical optics for each ray entering an optical system a single, unique, ray exits. In mathematical terms, the optical system performs a transformation that maps every object ray to an image ray.[1] The object ray and its associated image ray are said to be *conjugate to* each other. This term also applies to corresponding pairs of object and image points and planes. The object and image rays and points are considered to be in two distinct optical spaces, *object space* and *image space*; additional intermediate optical spaces may be used as well.

3.2.1 Rotationally symmetric optical systems; Optical axis, axial points, and meridional planes

An optical system is rotationally symmetric if its imaging properties are unchanged by *any* rotation about some axis. This (unique) axis of rotational symmetry is the optical axis of the system. Optical systems can be folded using plane mirrors; the system is still considered to be rotationally symmetric if it possesses rotational symmetry when unfolded. Any point on the optical axis (in any space) is an *axial point*.

Rotational symmetry greatly simplifies the analysis of optical systems, which otherwise must be analyzed in three dimensions. Rotational symmetry allows the system to be analyzed by considering only rays confined to a single transverse plane containing the optical axis. Such a plane is called a *meridional plane*; it is a cross-section through the system.

3.2.2 Ideal, rotationally symmetric, optical imaging system

An *ideal*, rotationally symmetric, optical imaging system must meet three criteria:

1. All rays "originating" from *any* object point converge to a single image point (Imaging is *stigmatic*).
2. Object planes perpendicular to the optical axis are conjugate to image planes perpendicular to the axis.
3. The image of an object confined to a plane normal to the axis is geometrically similar to the object.

In some optical systems imaging is stigmatic for one or perhaps a few object points, but to be an ideal system imaging must be stigmatic for *every* object point.

Unlike rays in mathematics, optical rays extend to infinity in both directions. Rays are *real* when they are in the part of the optical system to which they apply, and are *virtual* elsewhere. For example, object rays are real on the object side of the optical system. In stigmatic imaging an object ray intersecting any specific point in object space must be conjugate to an image ray intersecting the conjugate point in image space. A consequence is that every point on an object ray is conjugate to some point on the conjugate image ray.

Geometrical similarity implies the image is a scale model of the object. There is no restriction on the image's orientation. The image may be inverted or otherwise rotated with respect to the object.

3.2.3 Focal and afocal systems, focal points

In afocal systems an object ray parallel to the optical axis is conjugate to an image ray parallel to the optical axis. Such systems have no focal points (hence *afocal*) and also lack principal and nodal points. The system is focal if an object ray parallel to the axis is conjugate to an image ray that intersects the optical axis. The intersection of the image ray with the optical axis is the focal point F' in image space. Focal systems also have an axial object point F such that any ray through F is conjugate to an image ray parallel to the optical axis. F is the object space focal point of the system.

3.2.4 Transformation

The transformation between object space and image space is completely defined by the cardinal points of the system, and these points can be used to map any point on the object to its conjugate image point.

3.3 See also

- Film plane
- Pinhole camera model
- Radius of curvature (optics)
- Vergence (optics)

3.4 Notes and references

[1] Greivenkamp, John E. (2004). *Field Guide to Geometrical Optics*. SPIE Field Guides vol. **FG01**. SPIE. pp. 5–20. ISBN 0-8194-5294-7.

[2] Welford, W.T. (1986). *Aberrations of Optical Systems*. CRC. ISBN 0-85274-564-8.

[3] Hecht, Eugene (2002). *Optics* (4th ed.). Addison Wesley. p. 155. ISBN 0-321-18878-0.

[4] Kerr, Douglas A. (2005). "The Proper Pivot Point for Panoramic Photography" (PDF). *The Pumpkin*. Archived from the original (PDF) on 13 May 2006. Retrieved 5 March 2006.

[5] van Walree, Paul. "Misconceptions in photographic optics". Retrieved 1 January 2007. Item #6.

[6] Littlefield, Rik (6 February 2006). "Theory of the "No-Parallax" Point in Panorama Photography" (pdf). ver. 1.0. Retrieved 14 January 2007.

[7] Searle, G.F.C. 1912 *Revolving Table Method of Measuring Focal Lengths of Optical Systems* in "Proceedings of the Optical Convention 1912" p.168-171.

[8] Gray, Henry (1918). "Anatomy of the Human Body". p. 1019. Retrieved 12 February 2009.

- Hecht, Eugene (1987). *Optics* (2nd ed.). Addison Wesley. ISBN 0-201-11609-X.
- Lambda Research Corporation (2001). *OSLO Optics Reference* (PDF) (Version 6.1 ed.). Retrieved 5 March 2006. Pages 74–76 define the cardinal points.

3.5 External links

- Learn to use TEM

Chapter 4

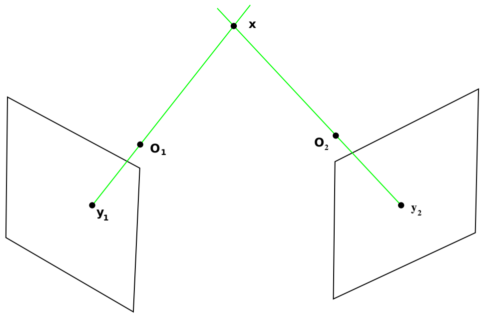
Triangulation (computer vision)

For a broader coverage related to this topic, see Computer stereo vision.
For other uses, see Triangulation (disambiguation).

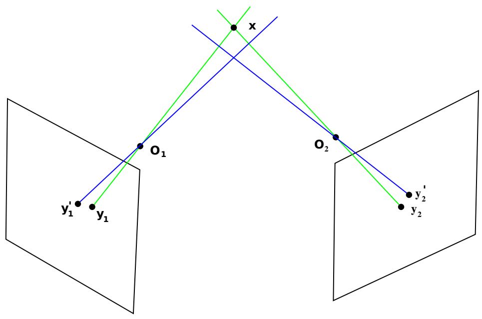
In computer vision **triangulation** refers to the process of determining a point in 3D space given its projections onto two, or more, images. In order to solve this problem it is necessary to know the parameters of the camera projection function from 3D to 2D for the cameras involved, in the simplest case represented by the camera matrices. Triangulation is sometimes also referred to as **reconstruction**.

The triangulation problem is in theory trivial. Since each point in an image corresponds to a line in 3D space, all points on the line are projected to the point in the image. If a pair of corresponding points in two, or more images, can be found it must be the case that they are the projection of a common 3D point $\mathbf{x}$. The set of lines generated by the image points must intersect at $\mathbf{x}$ and the algebraic formulation of the coordinates of $\mathbf{x}$ can be computed in a variety of ways, as is presented below.

In practice, however, the coordinates of image points cannot be measured with arbitrary accuracy. Instead, various types of noise, such as geometric noise from lens distortion or interest point detection error, lead to inaccuracies in the measured image coordinates. As a consequence, the lines generated by the corresponding image points do not always intersect in 3D space. The problem, then, is to find a 3D point which optimally fits the measured image points. In the literature there are multiple proposals for how to define optimality and how to find the optimal 3D point. Since they are based on different optimality criteria, the various methods produce different estimates of the 3D point $\mathbf{x}$ when noise is involved.



The ideal case of epipolar geometry. A 3D point $\mathbf{x}$ is projected onto two camera images through lines (green) which intersect with each camera's focal point, $\mathbf{O}_1$ and $\mathbf{O}_2$. The resulting image points are $\mathbf{y}_1$ and $\mathbf{y}_2$. The green lines intersect at $\mathbf{x}$.



In practice, the image points $\mathbf{y}_1$ and $\mathbf{y}_2$ cannot be measured with arbitrary accuracy. Instead points $\mathbf{y'}_1$ and $\mathbf{y'}_2$ are detected and used for the triangulation. The corresponding projection lines (blue) do not, in general, intersect in 3D space and may also not intersect with point $\mathbf{x}$.

4.1 Introduction

In the following, it is assumed that triangulation is made on corresponding image points from two views generated by pinhole cameras. Generalization from these assumptions are discussed here.

The image to the left illustrates the epipolar geometry of a pair of stereo cameras of pinhole model. A point $\mathbf{x}$ in 3D space is projected onto the respective image plane along a line (green) which goes through the camera's focal point, $\mathbf{O}_1$ and $\mathbf{O}_2$, resulting in the two corresponding image points $\mathbf{y}_1$ and $\mathbf{y}_2$. If $\mathbf{y}_1$ and $\mathbf{y}_2$ are given and the geometry of the two cameras are known, the two projection lines can be determined and it must be the case that they intersect at point $\mathbf{x}$. Using basic linear algebra that intersection point can be determined in a straightforward way.

The image to the right shows the real case. The position of the image points $\mathbf{y}_1$ and $\mathbf{y}_2$ cannot be measured exactly. The reason is a combination of factors such as

- Geometric distortion, for example lens distortion, which means that the 3D to 2D mapping of the camera deviates from the pinhole camera model. To some extent these errors can be compensated for, leaving a residual geometric error.
- A single ray of light from $\mathbf{x}$ is dispersed in the lens system of the cameras according to a point spread function. The recovery of the corresponding image point from measurements of the dispersed intensity function in the images gives errors.
- In digital camera the image intensity function is only measured in discrete sensor elements. Inexact interpolation of the discrete intensity function have to be used to recover the true one.
- The image points used for triangulation are often found using various types of feature extractors, for example of corners or interest points in general. There is an inherent localization error for any type of feature extraction based on neighborhood operations.

As a consequence, the measured image points are $\mathbf{y}'_1$ and $\mathbf{y}'_2$ instead of $\mathbf{y}_1$ and $\mathbf{y}_2$. However, their projection lines (blue) do not have to intersect in 3D space or come close to $\mathbf{x}$. In fact, these lines intersect if and only if $\mathbf{y}'_1$ and $\mathbf{y}'_2$ satisfy the epipolar constraint defined by the fundamental matrix. Given the measurement noise in $\mathbf{y}'_1$ and $\mathbf{y}'_2$ it is rather likely that the epipolar constraint is not satisfied and the projection lines do not intersect.

This observation leads to the problem which is solved in triangulation. Which 3D point $\mathbf{x}_{\mathbf{est}}$ is the best estimate of $\mathbf{x}$ given $\mathbf{y}'_1$ and $\mathbf{y}'_2$ and the geometry of the cameras? The answer is often found by defining an error measure which depends on $\mathbf{x}_{\mathbf{est}}$ and then minimize this error. In the following some of the various methods for computing $\mathbf{x}_{\mathbf{est}}$ presented in the literature are briefly described.

All triangulation methods produce $\mathbf{x}_{\mathbf{est}} = \mathbf{x}$ in the case that $\mathbf{y}_1 = \mathbf{y}'_1$ and $\mathbf{y}_2 = \mathbf{y}'_2$, that is, when the epipolar constraint is satisfied (except for singular points, see below). It is what happens when the constraint is not satisfied which differs between the methods.

4.2 Properties of triangulation methods

A triangulation method can be described in terms of a function τ such that

$$\mathbf{x} \sim \tau(\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1, \mathbf{C}_2)$$

where $\mathbf{y}'_1, \mathbf{y}'_2$ are the homogeneous coordinates of the detected image points and $\mathbf{C}_1, \mathbf{C}_2$ are the camera matrices. $\mathbf{x}$ is the homogeneous representation of the resulting 3D point. The $\sim$ sign implies that τ is only required to produce a vector which is equal to $\mathbf{x}$ up to a multiplication by a non-zero scalar since homogeneous vectors are involved.

Before looking at the specific methods, that is, specific functions τ , there are some general concepts related to the methods that need to be explained. Which triangulation method is chosen for a particular problem depends to some extent on these characteristics.

4.2.1 Singularities

Some of the methods fail to correctly compute an estimate of $\mathbf{x}$ if it lies in a certain subset of the 3D space, corresponding to some combination of $\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1, \mathbf{C}_2$. A point in this subset is then a *singularity* of the triangulation method. The reason for the failure can be that some equation system to be solved is under-determined or that the projective representation of $\mathbf{x}_{\mathbf{est}}$ becomes the zero vector for the singular points.

4.2.2 Invariance

In some applications, it is desirable that the triangulation is independent of the coordinate system used to represent 3D points; if the triangulation problem is formulated in one coordinate system and then transformed into another the resulting estimate $\mathbf{x}_{\mathbf{est}}$ should transform in the same way. This property is commonly referred to as *invariance*. Not every triangulation method assures invariance, at least not for general types of coordinate transformations.

For a homogeneous representation of 3D coordinates, the most general transformation is a projective transformation, represented by a 4×4 matrix $\mathbf{T}$. If the homogeneous coordinates are transformed according to

$$\hat{\mathbf{x}} \sim \mathbf{T}\,\mathbf{x}$$

then the camera matrices must transform as

$$\hat{\mathbf{C}}_k \sim \mathbf{C}_k\,\mathbf{T}^{-1}$$

to produce the same homogeneous image coordinates

$$\mathbf{y}_k \sim \hat{\mathbf{C}}_k\,\hat{\mathbf{x}} = \mathbf{C}_k\,\mathbf{x}$$

If the triangulation function τ is invariant to $\mathbf{T}$ then the following relation must be valid

$$\tilde{\mathbf{x}}_{\text{est}} \sim \mathbf{T}\,\mathbf{x}_{\text{est}}$$

from which follows that

$$\tau(\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1, \mathbf{C}_2) \sim \mathbf{T}^{-1}\,\tau(\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1\,\mathbf{T}^{-1}, \mathbf{C}_2\,\mathbf{T}^{-1}), \quad \text{for all } \mathbf{y}'_1, \mathbf{y}'_2$$

For each triangulation method, it can be determined if this last relation is valid. If it is, it may be satisfied only for a subset of the projective transformations, for example, rigid or affine transformations.

4.2.3 Computational complexity

The function τ is only an abstract representation of a computation which, in practice, may be relatively complex. Some methods result in a τ which is a closed-form continuous function while others need to be decomposed into a series of computational steps involving, for example, SVD or finding the roots of a polynomial. Yet another class of methods results in τ which must rely on iterative estimation of some parameters. This means that both the computation time and the complexity of the operations involved may vary between the different methods.

4.3 Some triangulation methods found in the literature

4.3.1 Mid-point method

Each of the two image points $\mathbf{y}'_1$ and $\mathbf{y}'_2$ has a corresponding projection line (blue in the right image above), here denoted as $\mathbf{L}'_1$ and $\mathbf{L}'_2$, which can be determined given the camera matrices $\mathbf{C}_1, \mathbf{C}_2$. Let d be a distance function between a 3D line and a 3D point such that

$d(\mathbf{L}, \mathbf{x})$ = the Euclidean distance between $\mathbf{L}$ and $\mathbf{x}$.

The midpoint method finds the point $\mathbf{x}_{\text{est}}$ which minimizes

$$d(\mathbf{L}'_1, \mathbf{x})^2 + d(\mathbf{L}'_2, \mathbf{x})^2$$

It turns out that $\mathbf{x}_{\text{est}}$ lies exactly at the middle of the shortest line segment which joins the two projection lines.

4.3.2 Direct linear transformation

Main article: Direct linear transformation

4.3.3 Via the essential matrix

4.3.4 Optimal triangulation

4.4 References

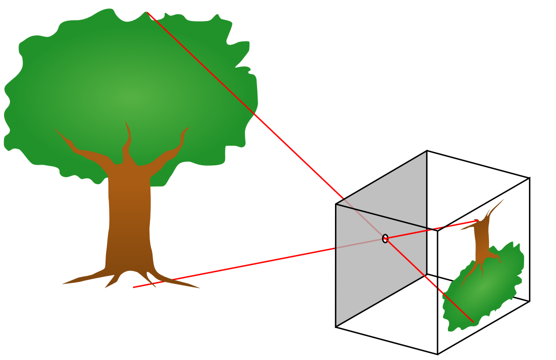
- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*. Cambridge University Press. ISBN 978-0-521-54051-3.

Chapter 5

Pinhole camera model

For a broader coverage related to this topic, see Epipolar geometry.

The **pinhole camera model** describes the mathematical

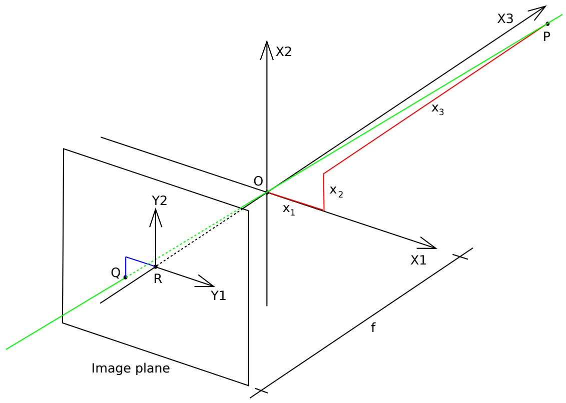


A diagram of a pinhole camera.

relationship between the coordinates of a 3D point and its projection onto the image plane of an *ideal* pinhole camera, where the camera aperture is described as a point and no lenses are used to focus light. The model does not include, for example, geometric distortions or blurring of unfocused objects caused by lenses and finite sized apertures. It also does not take into account that most practical cameras have only discrete image coordinates. This means that the pinhole camera model can only be used as a first order approximation of the mapping from a 3D scene to a 2D image. Its validity depends on the quality of the camera and, in general, decreases from the center of the image to the edges as lens distortion effects increase.

Some of the effects that the pinhole camera model does not take into account can be compensated, for example by applying suitable coordinate transformations on the image coordinates, and other effects are sufficiently small to be neglected if a high quality camera is used. This means that the pinhole camera model often can be used as a reasonable description of how a camera depicts a 3D scene, for example in computer vision and computer graphics.

5.1 The geometry and mathematics of the pinhole camera



The geometry of a pinhole camera

NOTE: The $x_1x_2x_3$ coordinate system in the figure is left-handed, that is the direction of the OZ axis is in reverse to the system the reader may be used to.

The geometry related to the mapping of a pinhole camera is illustrated in the figure. The figure contains the following basic objects:

- A 3D orthogonal coordinate system with its origin at **O**. This is also where the *camera aperture* is located. The three axes of the coordinate system are referred to as X1, X2, X3. Axis X3 is pointing in the viewing direction of the camera and is referred to as the *optical axis*, *principal axis*, or *principal ray*. The 3D plane which intersects with axes X1 and X2 is the front side of the camera, or *principal plane*.

- An image plane where the 3D world is projected through the aperture of the camera. The image plane is parallel to axes X1 and X2 and is located at distance f from the origin **O** in the negative direction of the X3 axis. A practical implementation of a pinhole camera implies that the image plane is located such that it intersects the X3 axis at coordinate *-f* where $f > 0$. f is also referred to as the *focal length* of the pinhole camera.

- A point **R** at the intersection of the optical axis and the image plane. This point is referred to as the *principal point* or *image center*.

- A point **P** somewhere in the world at coordinate (x_1, x_2, x_3) relative to the axes X1,X2,X3.

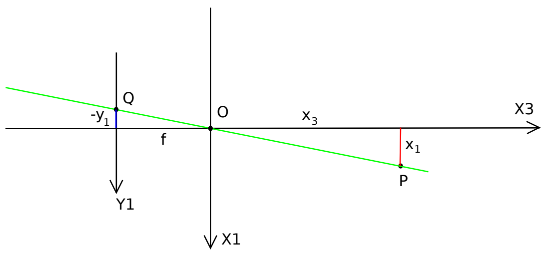
- The *projection line* of point **P** into the camera. This is the green line which passes through point **P** and the point **O**.

- The projection of point **P** onto the image plane, denoted **Q**. This point is given by the intersection of the projection line (green) and the image plane. In any practical situation we can assume that $x_3 > 0$ which means that the intersection point is well defined.

- There is also a 2D coordinate system in the image plane, with origin at **R** and with axes Y1 and Y2 which are parallel to X1 and X2, respectively. The coordinates of point **Q** relative to this coordinate system is (y_1, y_2) .

The *pinhole* aperture of the camera, through which all projection lines must pass, is assumed to be infinitely small, a point. In the literature this point in 3D space is referred to as the *optical (or lens or camera) center*.[1]

Next we want to understand how the coordinates (y_1, y_2) of point **Q** depend on the coordinates (x_1, x_2, x_3) of point **P**. This can be done with the help of the following figure which shows the same scene as the previous figure but now from above, looking down in the negative direction of the X2 axis.



The geometry of a pinhole camera as seen from the X2 axis

In this figure we see two similar triangles, both having parts of the projection line (green) as their hypotenuses. The catheti of the left triangle are $-y_1$ and f and the catheti of the right triangle are x_1 and x_3 . Since the two triangles are similar it follows that

$$\frac{-y_1}{f} = \frac{x_1}{x_3} \text{ or } y_1 = -\frac{f\,x_1}{x_3}$$

A similar investigation, looking in the negative direction of the X1 axis gives

$$\frac{-y_2}{f} = \frac{x_2}{x_3} \text{ or } y_2 = -\frac{f\,x_2}{x_3}$$

This can be summarized as

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = -\frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

which is an expression that describes the relation between the 3D coordinates (x_1, x_2, x_3) of point **P** and its image coordinates (y_1, y_2) given by point **Q** in the image plane.

5.1.1 Rotated image and the virtual image plane

The mapping from 3D to 2D coordinates described by a pinhole camera is a perspective projection followed by a 180° rotation in the image plane. This corresponds to how a real pinhole camera operates; the resulting image is rotated 180° and the relative size of projected objects depends on their distance to the focal point and the overall size of the image depends on the distance *f* between the image plane and the focal point. In order to produce an unrotated image, which is what we expect from a camera, there are two possibilities:

- Rotate the coordinate system in the image plane 180° (in either direction). This is the way any practical implementation of a pinhole camera would solve the problem; for a photographic camera we rotate the image before looking at it, and for a digital camera we read out the pixels in such an order that it becomes rotated.

- Place the image plane so that it intersects the X3 axis at *f* instead of at -*f* and rework the previous calculations. This would generate a *virtual (or front) image plane* which cannot be implemented in practice, but provides a theoretical camera which may be simpler to analyse than the real one.

In both cases the resulting mapping from 3D coordinates to 2D image coordinates is given by

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

(same as before except no minus sign)

5.2 Homogeneous coordinates

Main article: Camera matrix

The mapping from 3D coordinates of points in space to 2D image coordinates can also be represented in homogeneous coordinates. Let **x** be a representation of a 3D point in homogeneous coordinates (a 4-dimensional vector), and let **y** be a representation of the image of this point in the pinhole camera (a 3-dimensional vector). Then the following relation holds

$$\mathbf{y} \sim \mathbf{C}\,\mathbf{x}$$

where **C** is the 3×4 camera matrix and the $\sim$ means equality between elements of projective spaces. This implies that the left and right hand sides are equal up to a non-zero scalar multiplication. A consequence of this relation is that also **C** can be seen as an element of a projective space; two camera matrices are equivalent if they are equal up to a scalar multiplication. This description of the pinhole camera mapping, as a linear transformation **C** instead of as a fraction of two linear expressions, makes it possible to simplify many derivations of relations between 3D and 2D coordinates.

5.3 See also

- Entrance pupil, the equivalent location of the pinhole in relation to object space in a real camera.
- Exit pupil, the equivalent location of the pinhole in relation to the image plane in a real camera.
- Pinhole camera, the practical implementation of the mathematical model described in this article.

5.4 References

[1] Andrea Fusiello (2005-12-27). "Elements of Geometric Computer Vision". Homepages.inf.ed.ac.uk. Retrieved 2013-12-18.

5.5 Bibliography

- David A. Forsyth and Jean Ponce (2003). *Computer Vision, A Modern Approach*. Prentice Hall. ISBN 0-12-379777-2.
- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*. Cambridge University Press. ISBN 0-521-54051-8.
- Bernd Jähne (1997). *Practical Handbook on Image Processing for Scientific Applications*. CRC Press. ISBN 0-8493-8906-2.
- Linda G. Shapiro and George C. Stockman (2001). *Computer Vision*. Prentice Hall. ISBN 0-13-030796-3.
- Gang Xu and Zhengyou Zhang (1996). *Epipolar geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers. ISBN 0-7923-4199-6.

Chapter 6

Camera matrix

In computer vision a **camera matrix** or **(camera) projection matrix** is a 3×4 matrix which describes the mapping of a pinhole camera from 3D points in the world to 2D points in an image.

Let **x** be a representation of a 3D point in homogeneous coordinates (a 4-dimensional vector), and let **y** be a representation of the image of this point in the pinhole camera (a 3-dimensional vector). Then the following relation holds

$$\mathbf{y} \sim \mathbf{C}\,\mathbf{x}$$

where **C** is the camera matrix and the $\sim$ sign implies that the left and right hand sides are equal up to a non-zero scalar multiplication.

Since the camera matrix **C** is involved in the mapping between elements of two projective spaces, it too can be regarded as a projective element. This means that it has only 11 degrees of freedom since any multiplication by a non-zero scalar results in an equivalent camera matrix.

6.1 Derivation

The mapping from the coordinates of a 3D point **P** to the 2D image coordinates of the point's projection onto the image plane, according to the pinhole camera model is given by

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

where (x_1, x_2, x_3) are the 3D coordinates of **P** relative to a camera centered coordinate system, (y_1, y_2) are the resulting image coordinates, and f is the camera's focal length for which we assume $f > 0$. Furthermore, we also assume that $x_3 > 0$.

To derive the camera matrix this expression is rewritten in terms of homogeneous coordinates. Instead of the 2D vector (y_1, y_2) we consider the projective element (a 3D vector) $\mathbf{y} = (y_1, y_2, 1)$ and instead of equality we consider equality up to scaling by a non-zero number, denoted $\sim$. First, we write the homogeneous image coordinates as expressions in the usual 3D coordinates.

$$\begin{pmatrix} y_1 \\ y_2 \\ 1 \end{pmatrix} = \frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \\ \frac{x_3}{f} \end{pmatrix} \sim \begin{pmatrix} x_1 \\ x_2 \\ \frac{x_3}{f} \end{pmatrix}$$

Finally, also the 3D coordinates are expressed in a homogeneous representation **x** and this is how the camera matrix appears:

$$\begin{pmatrix} y_1 \\ y_2 \\ 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} \text{ or } \mathbf{y} \sim \mathbf{C}\,\mathbf{x}$$

where **C** is the camera matrix, which here is given by

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{pmatrix}$$

and the corresponding camera matrix now becomes

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The last step is a consequence of **C** itself being a projective element.

The camera matrix derived here may appear trivial in the sense that it contains very few non-zero elements. This depends to a large extent on the particular coordinate systems which have been chosen for the 3D and 2D points. In practice, however, other forms of camera matrices are common, as will be shown below.

6.2 Camera position

The camera matrix **C** derived in the previous section has a null space which is spanned by the vector

$$\mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

This is also the homogeneous representation of the 3D point which has coordinates (0,0,0), that is, the "camera center" (aka the entrance pupil; the position of the pinhole of a pinhole camera) is at **O**. This means that the camera center (and only this point) cannot be mapped to a point in the image plane by the camera (or equivalently, it maps to all points on the image as every ray on the image goes through this point).

For any other 3D point with $x_3 = 0$, the result $\mathbf{y} \sim \mathbf{C}\,\mathbf{x}$ is well-defined and has the form $\mathbf{y} = (y_1\, y_2\, 0)^\top$. This corresponds to a point at infinity in the projective image plane (even though, if the image plane is taken to be a Euclidean plane, no corresponding intersection point exists).

6.3 Normalized camera matrix and normalized image coordinates

The camera matrix derived above can be simplified even further if we assume that *f = 1*:

$$\mathbf{C}_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \left(\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array} \right)$$

where **I** here denotes a 3×3 identity matrix. Note that 3×4 matrix **C** here is divided into a concatenation of a 3×3 matrix and a 3-dimensional vector. The camera matrix $\mathbf{C}_0$ is sometimes referred to as a *canonical form*.

So far all points in the 3D world have been represented in a *camera centered* coordinate system, that is, a coordinate system which has its origin at the camera center (the location of the pinhole of a pinhole camera). In practice however, the 3D points may be represented in terms of coordinates relative to an arbitrary coordinate system (X1',X2',X3'). Assuming that the camera coordinate axes (X1,X2,X3) and the axes (X1',X2',X3') are of Euclidean type (orthogonal and isotropic), there is a unique Euclidean 3D transformation (rotation and translation) between the two coordinate systems. In other words, the camera is not necessarily at the origin looking along the *z* axis.

The two operations of rotation and translation of 3D coordinates can be represented as the two 4×4 matrices

$$\left(\begin{array}{c|c} \mathbf{R} & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array} \right) \text{ and } \left(\begin{array}{c|c} \mathbf{I} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array} \right)$$

where **R** is a 3×3 rotation matrix and **t** is a 3-dimensional translation vector. When the first matrix is multiplied onto the homogeneous representation of a 3D point, the result is the homogeneous representation of the rotated point, and the second matrix performs instead a translation. Performing the two operations in sequence gives a combined rotation and translation matrix

$$\left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array} \right)$$

Assuming that **R** and **t** are precisely the rotation and translations which relate the two coordinate system (X1,X2,X3) and (X1',X2',X3') above, this implies that

$$\mathbf{x} = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array} \right) \mathbf{x}'$$

where $\mathbf{x}'$ is the homogeneous representation of the point **P** in the coordinate system (X1',X2',X3').

Assuming also that the camera matrix is given by $\mathbf{C}_0$, the mapping from the coordinates in the (X1',X2',X3') system to homogeneous image coordinates becomes

$$\mathbf{y} \sim \mathbf{C}_0\,\mathbf{x} = \left(\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array} \right) \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array} \right) \mathbf{x}' = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array} \right) \mathbf{x}'$$

Consequently, the camera matrix which relates points in the coordinate system (X1',X2',X3') to image coordinates is

$$\mathbf{C}_N = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array} \right)$$

a concatenation of a 3D rotation matrix and a 3-dimensional translation vector.

This type of camera matrix is referred to as a *normalized camera matrix*, it assumes focal length = 1 and that image coordinates are measured in a coordinate system where the origin is located at the intersection between axis X3 and the image plane and has the same units as the 3D coordinate system. The resulting image coordinates are referred to as *normalized image coordinates*.

6.3.1 The camera position

Again, the null space of the normalized camera matrix, $\mathbf{C}_N$ described above, is spanned by the 4-dimensional vector

$$\mathbf{n} = \begin{pmatrix} -\mathbf{R}^{-1}\,\mathbf{t} \\ 1 \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{n}} \\ 1 \end{pmatrix}$$

This is also, again, the coordinates of the camera center, now relative to the (X1',X2',X3') system. This can be seen by applying first the rotation and then the translation to the 3-dimensional vector $\tilde{\mathbf{n}}$ and the result is the homogeneous representation of 3D coordinates (0,0,0).

This implies that the camera center (in its homogeneous representation) lies in the null space of the camera matrix, provided that it is represented in terms of 3D coordinates relative to the same coordinate system as the camera matrix refers to.

The normalized camera matrix $\mathbf{C}_N$ can now be written as

$$\mathbf{C}_N = \mathbf{R} \left(\begin{array}{c|c} \mathbf{I} & \mathbf{R}^{-1}\,\mathbf{t} \end{array} \right) = \mathbf{R} \left(\begin{array}{c|c} \mathbf{I} & -\tilde{\mathbf{n}} \end{array} \right)$$

where $\tilde{\mathbf{n}}$ is the 3D coordinates of the camera relative to the (X1',X2',X3') system.

6.4 General camera matrix

Given the mapping produced by a normalized camera matrix, the resulting normalized image coordinates can be transformed by means of an arbitrary 2D homography. This includes 2D translations and rotations as well as scaling (isotropic and anisotropic) but also general 2D perspective transformations. Such a transformation can be represented as a 3×3 matrix $\mathbf{H}$ which maps the homogeneous normalized image coordinates $\mathbf{y}$ to the homogeneous transformed image coordinates $\mathbf{y}'$:

$$\mathbf{y}' = \mathbf{H}\,\mathbf{y}$$

Inserting the above expression for the normalized image coordinates in terms of the 3D coordinates gives

$$\mathbf{y}' = \mathbf{H}\,\mathbf{C}_N\,\mathbf{x}'$$

This produces the most general form of camera matrix

$$\mathbf{C} = \mathbf{H}\,\mathbf{C}_N = \mathbf{H} \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array} \right)$$

6.5 See also

- 3D projection

6.6 References

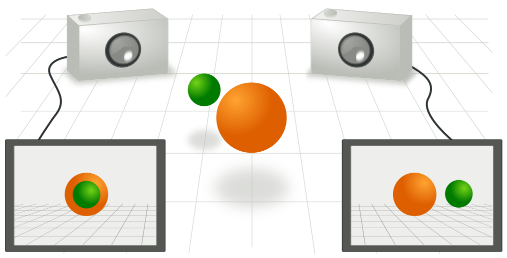
- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*. Cambridge University Press. ISBN 0-521-54051-8.

Chapter 7

Epipolar geometry

For a broader coverage related to this topic, see Computer stereo vision.

Epipolar geometry is the geometry of stereo vision.



Typical use case for epipolar geometry
Two cameras take a picture of the same scene from different points of view. The epipolar geometry then describes the relation between the two resulting views.

When two cameras view a 3D scene from two distinct positions, there are a number of geometric relations between the 3D points and their projections onto the 2D images that lead to constraints between the image points. These relations are derived based on the assumption that the cameras can be approximated by the pinhole camera model.

7.1 Epipolar geometry

The figure below depicts two pinhole cameras looking at point **X**. In real cameras, the image plane is actually behind the center of projection, and produces an image that is rotated 180 degrees. Here, however, the projection problem is simplified by placing a *virtual image plane* in front of the center of projection of each camera to produce an unrotated image. **O**L and **O**R represent the centers of projection of the two cameras. **X** represents the point of interest in both cameras. Points **x**L and **x**R are the projections of point **X** onto the image planes.

Each camera captures a 2D image of the 3D world. This conversion from 3D to 2D is referred to as a perspective projection and is described by the pinhole camera model. It is common to model this projection operation by rays that emanate from the camera, passing through its center of projection. Note that each emanating ray corresponds to a single point in the image.

7.1.1 Epipole or epipolar point

Since the centers of projection of the cameras are distinct, each center of projection projects onto a distinct point into the other camera's image plane. These two image points are denoted by **e**L and **e**R and are called *epipoles* or *epipolar points*. Both epipoles **e**L and **e**R in their respective image planes and both centers of projection **O**L and **O**R lie on a single 3D line.

7.1.2 Epipolar line

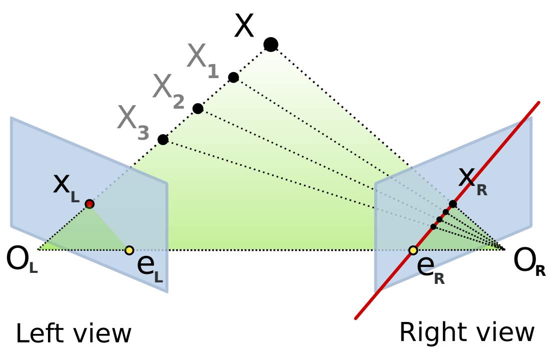
The line **O**L–**X** is seen by the left camera as a point because it is directly in line with that camera's center of projection. However, the right camera sees this line as a line in its image plane. That line (**e**R–**x**R) in the right camera is called an *epipolar line*. Symmetrically, the line **O**R–**X** seen by the right camera as a point is seen as epipolar line **e**L–**x**Lby the left camera.

An epipolar line is a function of the 3D point **X**, i.e. there is a set of epipolar lines in both images if we allow **X** to vary over all 3D points. Since the 3D line **O**L–**X** passes through the center of projection **O**L, the corresponding epipolar line in the right image must pass through the epipole **e**R (and correspondingly for epipolar lines in the left image). This means that all epipolar lines in one image must intersect the epipolar point of that image. In fact, any line which intersects with the epipolar point is an epipolar line since it can be derived from some 3D point **X**.

7.1.3 Epipolar plane

As an alternative visualization, consider the points **X**, **O**L & **O**R that form a plane called the *epipolar plane*. The epipolar plane intersects each camera's image plane where it forms lines—the epipolar lines. All epipolar planes and epipolar lines intersect the epipole regardless of where **X** is located.

7.1.4 Epipolar constraint and triangulation



Epipolar geometry

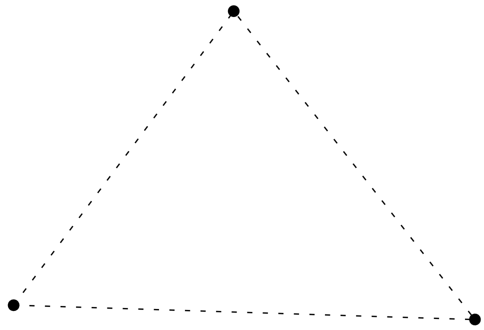
If the relative translation and rotation of the two cameras is known, the corresponding epipolar geometry leads to two important observations

- If the projection point **x**L is known, then the epipolar line **e**R–**x**R is known and the point **X** projects into the right image, on a point **x**R which must lie on this particular epipolar line. This means that for each point observed in one image the same point must be observed in the other image on a known epipolar line. This provides an *epipolar constraint* which corresponding image points must satisfy and it means that it is possible to test if two points really correspond to the same 3D point. Epipolar constraints can also be described by the essential matrix or the fundamental matrix between the two cameras.

- If the points **x**L and **x**R are known, their projection lines are also known. If the two image points correspond to the same 3D point **X** the projection lines must intersect precisely at **X**. This means that **X** can be calculated from the coordinates of the two image points, a process called *triangulation*.

7.1.5 Simplified cases

The epipolar geometry is simplified if the two camera image planes coincide. In this case, the epipolar lines also coincide (**EL**–**PL** = **ER**–**PR**). Furthermore, the epipolar lines are parallel to the line **OL**–**OR** between the centers of projection, and can in practice be aligned with the horizontal axes of the two images. This means that for each point in one image, its corresponding point in the other image can be found by looking only along a horizontal line. If the cameras cannot be positioned in this way, the image coordinates from the cameras may be transformed to emulate having a common image plane. This process is called image rectification.



Example of epipolar geometry. *Two cameras, with their respective centers of projection points **OL** and **OR**, observe a point **P**. The projection of **P** onto each of the image planes is denoted **pL** and **pR**. Points **EL** and **ER** are the epipoles.*

7.1.6 Epipolar geometry of pushbroom sensor

In contrast to the conventional frame camera which uses a two-dimensional CCD, pushbroom camera adopts an array of one-dimensional CCDs to produce long continuous image strip which is called "image carpet". Epipolar geometry of this sensor is quite different from that of frame cameras. First, the epipolar line of pushbroom sensor is not straight, but hyperbola-like curve. Second, epipolar 'curve' pair does not exist.[1]

7.2 See also

- 3D reconstruction
- 3D reconstruction from multiple images
- 3D scanner
- Binocular disparity
- Photogrammetry

7.3 References

[1] Jaehong Oh. "Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-based Georeferencing of Aerial Images", 2011, accessed 2011-08-05.

7.4 Further reading

- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*. Cambridge University Press. ISBN 0-521-54051-8.

- Quang-Tuan Luong. "Learning Epipolar Geometry". *Artificial Intelligence Center*. SRI International. Retrieved 2007-03-04.

- Robyn Owens. "Epipolar geometry". Retrieved 2007-03-04.

- Linda G. Shapiro and George C. Stockman (2001). *Computer Vision*. Prentice Hall. pp. 395–403. ISBN 0-13-030796-3.

- Vishvjit S. Nalwa (1993). *A Guided Tour of Computer Vision*. Addison Wesley. pp. 216–240. ISBN 0-201-54853-4.

- Roberto Cipolla and Peter Giblin (2000). *Visual motion of curves and surfaces*. Cambridge University Press, Cambridge. ISBN 0-521-63251-X.

Chapter 8

Fundamental matrix (computer vision)

In computer vision, the **fundamental matrix F** is a 3×3 matrix which relates corresponding points in stereo images. In epipolar geometry, with homogeneous image coordinates, $\mathbf{x}$ and $\mathbf{x}'$, of corresponding points in a stereo image pair, $\mathbf{Fx}$ describes a line (an epipolar line) on which the corresponding point $\mathbf{x}'$ on the other image must lie. That means, for all pairs of corresponding points holds

$$\mathbf{x}'^{\top}\mathbf{Fx} = 0.$$

Being of rank two and determined only up to scale, the fundamental matrix can be estimated given at least seven point correspondences. Its seven parameters represent the only geometric information about cameras that can be obtained through point correspondences alone.

The term “fundamental matrix” was coined by QT Luong in his influential PhD thesis. It is sometimes also referred to as the "**bifocal tensor**". As a tensor it is a two-point tensor in that it is a bilinear form relating points in distinct coordinate systems.

The above relation which defines the fundamental matrix was published in 1992 by both Faugeras and Hartley. Although Longuet-Higgins’ essential matrix satisfies a similar relationship, the essential matrix is a metric object pertaining to calibrated cameras, while the fundamental matrix describes the correspondence in more general and fundamental terms of projective geometry. This is captured mathematically by the relationship between a fundamental matrix $\mathbf{F}$ and its corresponding essential matrix $\mathbf{E}$, which is

$$\mathbf{E} = \mathbf{K}'^{\top}\ \mathbf{F}\ \mathbf{K}$$

$\mathbf{K}$ and $\mathbf{K}'$ being the intrinsic calibration matrices of the two images involved.

8.1 Introduction

The fundamental matrix is a relationship between any two images of the same scene that constrains where the projection of points from the scene can occur in both images. Given the projection of a scene point into one of the images the corresponding point in the other image is constrained to a line, helping the search, and allowing for the detection of wrong correspondences. The relation between corresponding image points which the fundamental matrix represents is referred to as *epipolar constraint*, *matching constraint*, *discrete matching constraint*, or *incidence relation*.

8.2 Projective reconstruction theorem

The fundamental matrix can be determined by a set of point correspondences. Additionally, these corresponding image points may be *triangulated* to world points with the help of camera matrices derived directly from this fundamental matrix. The scene composed of these world points is within a projective transformation of the true scene.[1]

8.2.1 Proof

Say that the image point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ derives from the world point $\mathbf{X}$ under the camera matrices $\left(\mathbf{P}, \mathbf{P}'\right)$ as

$$\begin{aligned} \mathbf{x} &= \mathbf{PX} \\ \mathbf{x}' &= \mathbf{P}'\mathbf{X} \end{aligned}$$

Say we transform space by a general homography matrix $\mathbf{H}_{4\times4}$ such that $\mathbf{X}_0 = \mathbf{HX}$.

The cameras then transform as

$$\begin{aligned} \mathbf{P}_0 &= \mathbf{PH}^{-1} \\ \mathbf{P}'_0 &= \mathbf{P}'\mathbf{H}^{-1} \end{aligned}$$

$\mathbf{P}_0\mathbf{X}_0 = \mathbf{PH}^{-1}\mathbf{HX} = \mathbf{PX} = \mathbf{x}$ and likewise with $\mathbf{P}'_0$ still get us the same image points.

8.3 Derivation of fundamental matrix using coplanarity condition

Fundamental matrix can be derived using the coplanarity condition. [2]

8.4 Properties

The fundamental matrix is of rank 2. Its kernel defines the epipole.

8.5 See also

- Epipolar geometry
- Essential matrix

8.6 Notes

[1] Hartley 2003, pp. 266–267

[2] Jaehong Oh. "Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-based Georeferencing of Aerial Images", 2011, pp. 22–29 accessed 2011-08-05.

8.7 References

- Olivier D. Faugeras (1992). "What can be seen in three dimensions with an uncalibrated stereo rig?". *Proceedings of European Conference on Computer Vision*.
- Olivier D. Faugeras; Quang-Tuan Luong; Steven Maybank (1992). "Camera self-calibration: Theory and experiments". *Proceedings of European Conference on Computer Vision*.
- Q. T. Luong and Olivier D. Faugeras (1996). "The Fundamental Matrix: Theory, Algorithms, and Stability Analysis". *International Journal of Computer Vision* **17** (1): 43–75. doi:10.1007/BF00127818.
- Olivier Faugeras and Q. T. Luong (2001). *The Geometry of Multiple Images*. MIT Press. ISBN 0-262-06220-8.
- Richard I. Hartley (1992). "Estimation of relative camera positions for uncalibrated cameras". *Proceedings of European Conference on Computer Vision*.
- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*. Cambridge University Press. ISBN 0-521-54051-8.
- Richard I. Hartley (1997). "In Defense of the Eight-Point Algorithm". *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19** (6): 580–593. doi:10.1109/34.601246.
- Q. T. Luong (1992). *Matrice fondamentale et autocalibration en vision par ordinateur*. PhD Thesis, University of Paris, Orsay.
- Yi Ma; Stefano Soatto; Jana Košecká; S. Shankar Sastry (2004). *An Invitation to 3-D Vision*. Springer. ISBN 0-387-00893-4.
- Marc Pollefeys, Reinhard Koch and Luc van Gool (1999). "Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Intrinsic Camera Parameters". *International Journal of Computer Vision* **32** (1): 7–25. doi:10.1023/A:1008109111715.
- Philip H. S. Torr (1997). "The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix". *International Journal of Computer Vision* **24** (3): 271–300. doi:10.1023/A:1007927408552.
- Philip H. S. Torr and A. Zisserman (2000). "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry". *Journal of Computer Vision and Image Understanding* **78** (1): 138–156.
- Gang Xu and Zhengyou Zhang (1996). *Epipolar geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers. ISBN 0-7923-4199-6.
- Zhengyou Zhang (1998). "Determining the epipolar geometry and its uncertainty: A review". *International Journal of Computer Vision* **27** (2): 161–195. doi:10.1023/A:1007941100561.

8.8 Toolboxes

- fundest is a GPL C/C++ library for robust, non-linear (based on the Levenberg–Marquardt algorithm) fundamental matrix estimation from matched point pairs and various objective functions (Manolis Lourakis).
- Structure and Motion Toolkit in MATLAB (Philip H. S. Torr)
- Fundamental Matrix Estimation Toolbox (Joaquim Salvi)
- The Epipolar Geometry Toolbox (EGT)

8.9 External links

- Epipolar Geometry and the Fundamental Matrix (chapter from Hartley & Zisserman)
- Determining the epipolar geometry and its uncertainty: A review (Zhengyou Zhang)
- Visualization of epipolar geometry (originally by Sylvain Bougnoux of INRIA Robotvis, requires Java)
- The Fundamental Matrix Song Video demonstrating laws of epipolar geometry.

Chapter 9

Essential matrix

In computer vision, the **essential matrix** is a 3×3 matrix, $\mathbf{E}$, with some additional properties described below, which relates corresponding points in stereo images assuming that the cameras satisfy the pinhole camera model.

9.1 Function

More specifically, if $\mathbf{y}$ and $\mathbf{y}'$ are homogeneous *normalized* image coordinates in image 1 and 2, respectively, then

$$(\mathbf{y}')^\top \mathbf{E}\,\mathbf{y} = 0$$

if $\mathbf{y}$ and $\mathbf{y}'$ correspond to the same 3D point in the scene.

The above relation which defines the essential matrix was published in 1981 by Longuet-Higgins, introducing the concept to the computer vision community. Hartley & Zisserman's book reports that an analogous matrix appeared in photogrammetry long before that. Longuet-Higgins' paper includes an algorithm for estimating $\mathbf{E}$ from a set of corresponding normalized image coordinates as well as an algorithm for determining the relative position and orientation of the two cameras given that $\mathbf{E}$ is known. Finally, it shows how the 3D coordinates of the image points can be determined with the aid of the essential matrix.

9.2 Use

The essential matrix can be seen as a precursor to the fundamental matrix. Both matrices can be used for establishing constraints between matching image points, but the essential matrix can only be used in relation to calibrated cameras since the inner camera parameters must be known in order to achieve the normalization. If, however, the cameras are calibrated the essential matrix can be useful for determining both the relative position and orientation between the cameras and the 3D position of corresponding image points.

9.3 Derivation and definition

This derivation follows the paper by Longuet-Higgins.

Two normalized cameras project the 3D world onto their respective image planes. Let the 3D coordinates of a point $\mathbf{P}$ be (x_1, x_2, x_3) and (x'_1, x'_2, x'_3) relative to each camera's coordinate system. Since the cameras are normalized, the corresponding image coordinates are

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{1}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \text{ and } \begin{pmatrix} y'_1 \\ y'_2 \end{pmatrix} = \frac{1}{x'_3} \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}$$

A homogeneous representation of the two image coordinates is then given by

$$\begin{pmatrix} y_1 \\ y_2 \\ 1 \end{pmatrix} = \frac{1}{x_3} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ and } \begin{pmatrix} y'_1 \\ y'_2 \\ 1 \end{pmatrix} = \frac{1}{x'_3} \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix}$$

which also can be written more compactly as

$$\mathbf{y} = \frac{1}{x_3}\,\tilde{\mathbf{x}} \text{ and } \mathbf{y}' = \frac{1}{x'_3}\,\tilde{\mathbf{x}}'$$

where $\mathbf{y}$ and $\mathbf{y}'$ are homogeneous representations of the 2D image coordinates and $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ are proper 3D coordinates but in two different coordinate systems.

Another consequence of the normalized cameras is that their respective coordinate systems are related by means of a translation and rotation. This implies that the two sets of 3D coordinates are related as

$$\tilde{\mathbf{x}}' = \mathbf{R}\,(\tilde{\mathbf{x}} - \mathbf{t})$$

where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3-dimensional translation vector.

Define the essential matrix as

where $[\mathbf{t}]_\times$ is the matrix representation of the cross product with $\mathbf{t}$.

To see that this definition of the essential matrix describes a constraint on corresponding image coordinates multiply $\mathbf{E}$ from left and right with the 3D coordinates of point $\mathbf{P}$ in the two different coordinate systems:

$$(\tilde{\mathbf{x}}')^T \mathbf{E}\,\tilde{\mathbf{x}} \overset{(1)}{=} (\tilde{\mathbf{x}}-\mathbf{t})^T \mathbf{R}^T \mathbf{R}\,[\mathbf{t}]_\times \tilde{\mathbf{x}} \overset{(2)}{=} (\tilde{\mathbf{x}}-\mathbf{t})^T\,[\mathbf{t}]_\times \tilde{\mathbf{x}} \overset{(3)}{=} 0$$

1. Insert the above relations between $\tilde{\mathbf{x}}'$ and $\tilde{\mathbf{x}}$ and the definition of $\mathbf{E}$ in terms of $\mathbf{R}$ and $\mathbf{t}$.
2. $\mathbf{R}^T\,\mathbf{R} = \mathbf{I}$ since $\mathbf{R}$ is a rotation matrix.
3. Properties of the matrix representation of the cross product.

Finally, it can be assumed that both x_3 and x'_3 are > 0, otherwise they are not visible in both cameras. This gives

$$0 = (\tilde{\mathbf{x}}')^T\,\mathbf{E}\,\tilde{\mathbf{x}} = \frac{1}{x'_3}(\tilde{\mathbf{x}}')^T\,\mathbf{E}\,\frac{1}{x_3}\tilde{\mathbf{x}} = (\mathbf{y}')^T\,\mathbf{E}\,\mathbf{y}$$

which is the constraint that the essential matrix defines between corresponding image points.

9.4 Properties of the essential matrix

Not every arbitrary 3×3 matrix can be an essential matrix for some stereo cameras. To see this notice that it is defined as the matrix product of one rotation matrix and one skew-symmetric matrix, both 3×3 . The skew-symmetric matrix must have two singular values which are equal and another which is zero. The multiplication of the rotation matrix does not change the singular values which means that also the essential matrix has two singular values which are equal and one which is zero. The properties described here are sometimes referred to as *internal constraints* of the essential matrix.

If the essential matrix $\mathbf{E}$ is multiplied by a non-zero scalar, the result is again an essential matrix which defines exactly the same constraint as $\mathbf{E}$ does. This means that $\mathbf{E}$ can be seen as an element of a projective space, that is, two such matrices are considered equivalent if one is a non-zero scalar multiplication of the other. This is a relevant position, for example, if $\mathbf{E}$ is estimated from image data. However, it is also possible to take the position that $\mathbf{E}$ is defined as

$$\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times$$

and then $\mathbf{E}$ has a well-defined "scaling". It depends on the application which position is the more relevant.

The constraints can also be expressed as

$$\det \mathbf{E} = 0$$

and

$$2\mathbf{E}\mathbf{E}^T\mathbf{E} - \mathrm{tr}(\mathbf{E}\mathbf{E}^T)\mathbf{E} = 0.$$

Here the last equation is matrix constraint, which can be seen as 9 constraints, one for each matrix element. These constraints are often used for determining the essential matrix from five corresponding point pairs.

The essential matrix has five or six degrees of freedom, depending on whether or not it is seen as a projective element. The rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ have three degrees of freedom each, in total six. If the essential matrix is considered as a projective element, however, one degree of freedom related to scalar multiplication must be subtracted leaving five degrees of freedom in total.

9.5 Estimation of the essential matrix

Given a set of corresponding image points it is possible to estimate an essential matrix which satisfies the defining epipolar constraint for all the points in the set. However, if the image points are subject to noise, which is the common case in any practical situation, it is not possible to find an essential matrix which satisfies all constraints exactly.

Depending on how the error related to each constraint is measured, it is possible to determine or estimate an essential matrix which optimally satisfies the constraints for a given set of corresponding image points. The most straightforward approach is to set up a total least squares problem, commonly known as the eight-point algorithm.

9.6 Determining R and t from E

Given that the essential matrix has been determined for a stereo camera pair, for example, using the estimation method above this information can be used for determining also the rotation and translation (up to a scaling) between the two camera's coordinate systems. In these derivations $\mathbf{E}$ is seen as a projective element rather than having a well-determined scaling.

The following method for determining $\mathbf{R}$ and $\mathbf{t}$ is based on performing a SVD of $\mathbf{E}$, see Hartley & Zisserman's book. It is also possible to determine $\mathbf{R}$ and $\mathbf{t}$ without an SVD, for example, following Longuet-Higgins' paper.

9.6.1 Finding one solution

An SVD of $\mathbf{E}$ gives

$$\mathbf{E} = \mathbf{U}\,\Sigma\,\mathbf{V}^T$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal 3×3 matrices and Σ is a 3×3 diagonal matrix with

$$\Sigma = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The diagonal entries of Σ are the singular values of $\mathbf{E}$ which, according to the internal constraints of the essential matrix, must consist of two identical and one zero value. Define

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ with } \mathbf{W}^{-1} = \mathbf{W}^T = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and make the following ansatz

$$[\mathbf{t}]_\times = \mathbf{U}\,\mathbf{W}\,\Sigma\,\mathbf{U}^T$$

$$\mathbf{R} = \mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{V}^T$$

Since Σ may not completely fulfill the constraints when dealing with real world data (f.e. camera images), the alternative

$$[\mathbf{t}]_\times = \mathbf{U}\,\mathbf{Z}\,\mathbf{U}^T \text{ with } \mathbf{Z} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

may help.

9.6.2 Showing that it is valid

First, these expressions for $\mathbf{R}$ and $[\mathbf{t}]_\times$ do satisfy the defining equation for the essential matrix

$$\mathbf{R}\,[\mathbf{t}]_\times = \mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{V}^T\,\mathbf{V}\,\mathbf{W}\,\Sigma\,\mathbf{V}^T = \mathbf{U}\,\Sigma\,\mathbf{V}^T = \mathbf{E}$$

Second, it must be shown that this $[\mathbf{t}]_\times$ is a matrix representation of the cross product for some $\mathbf{t}$. Since

$$\mathbf{W}\,\Sigma = \begin{pmatrix} 0 & -s & 0 \\ s & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

it is the case that $\mathbf{W}\,\Sigma$ is skew-symmetric, i.e., $(\mathbf{W}\,\Sigma)^T = -\mathbf{W}\,\Sigma$. This is also the case for our $[\mathbf{t}]_\times$, since

$$([\mathbf{t}]_\times)^T = \mathbf{U}\,(\mathbf{W}\,\Sigma)^T\,\mathbf{U}^T = -\mathbf{U}\,\mathbf{W}\,\Sigma\,\mathbf{U}^T = -[\mathbf{t}]_\times$$

According to the general properties of the matrix representation of the cross product it then follows that $[\mathbf{t}]_\times$ must be the cross product operator of exactly one vector $\mathbf{t}$.

Third, it must also need to be shown that the above expression for $\mathbf{R}$ is a rotation matrix. It is the product of three matrices which all are orthogonal which means that $\mathbf{R}$, too, is orthogonal or $\det(\mathbf{R}) = \pm 1$. To be a proper rotation matrix it must also satisfy $\det(\mathbf{R}) = 1$. Since, in this case, $\mathbf{E}$ is seen as a projective element this can be accomplished by reversing the sign of $\mathbf{E}$ if necessary.

9.6.3 Finding all solutions

So far one possible solution for $\mathbf{R}$ and $\mathbf{t}$ has been established given $\mathbf{E}$. It is, however, not the only possible solution and it may not even be a valid solution from a practical point of view. To begin with, since the scaling of $\mathbf{E}$ is undefined, the scaling of $\mathbf{t}$ is also undefined. It must lie in the null space of $\mathbf{E}$ since

$$\mathbf{E}\,\mathbf{t} = \mathbf{R}\,[\mathbf{t}]_\times\,\mathbf{t} = \mathbf{0}$$

For the subsequent analysis of the solutions, however, the exact scaling of $\mathbf{t}$ is not so important as its "sign", i.e., in which direction it points. Let $\hat{\mathbf{t}}$ be normalized vector in the null space of $\mathbf{E}$. It is then the case that both $\hat{\mathbf{t}}$ and $-\hat{\mathbf{t}}$ are valid translation vectors relative $\mathbf{E}$. It is also possible to change $\mathbf{W}$ into $\mathbf{W}^{-1}$ in the derivations of $\mathbf{R}$ and $\mathbf{t}$ above. For the translation vector this only causes a change of sign, which has already been described as a possibility. For the rotation, on the other hand, this will produce a different transformation, at least in the general case.

To summarize, given $\mathbf{E}$ there are two opposite directions which are possible for $\mathbf{t}$ and two different rotations which are compatible with this essential matrix. In total this gives four classes of solutions for the rotation and translation between the two camera coordinate systems. On top of that, there is also an unknown scaling $s > 0$ for the chosen translation direction.

It turns out, however, that only one of the four classes of solutions can be realized in practice. Given a pair of corresponding image coordinates, three of the solutions will always produce a 3D point which lies *behind* at least one of the two cameras and therefore cannot be seen. Only one of the four classes will consistently produce 3D points which are in front of both cameras. This must then be the correct solution. Still, however, it has an undetermined positive scaling related to the translation component.

It should be noted that the above determination of $\mathbf{R}$ and $\mathbf{t}$ assumes that $\mathbf{E}$ satisfy the internal constraints of the essential matrix. If this is not the case which, for example, typically is the case if $\mathbf{E}$ has been estimated from real (and noisy) image data, it has to be assumed that it approximately satisfy the internal constraints. The vector $\hat{\mathbf{t}}$ is then chosen as right singular vector of $\mathbf{E}$ corresponding to the smallest singular value.

9.7 3D points from corresponding image points

The problem to be solved there is how to compute (x_1, x_2, x_3) given corresponding normalized image coordinates (y_1, y_2) and (y'_1, y'_2) . If the essential matrix is known and the corresponding rotation and translation transformations have been determined, this algorithm (described in Longuet-Higgins' paper) provides a solution.

Let $\mathbf{r}_k$ denote row *k* of the rotation matrix $\mathbf{R}$:

$$\mathbf{R} = \begin{pmatrix} -\mathbf{r}_1- \\ -\mathbf{r}_2- \\ -\mathbf{r}_3- \end{pmatrix}$$

Combining the above relations between 3D coordinates in the two coordinate systems and the mapping between 3D and 2D points described earlier gives

$$y'_1 = \frac{x'_1}{x'_3} = \frac{\mathbf{r}_1 \cdot (\tilde{\mathbf{x}} - \mathbf{t})}{\mathbf{r}_3 \cdot (\tilde{\mathbf{x}} - \mathbf{t})} = \frac{\mathbf{r}_1 \cdot (\mathbf{y} - \mathbf{t}/x_3)}{\mathbf{r}_3 \cdot (\mathbf{y} - \mathbf{t}/x_3)}$$

or

$$x_3 = \frac{(\mathbf{r}_1 - y'_1\,\mathbf{r}_3) \cdot \mathbf{t}}{(\mathbf{r}_1 - y'_1\,\mathbf{r}_3) \cdot \mathbf{y}}$$

Once x_3 is determined, the other two coordinates can be computed as

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_3 \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

The above derivation is not unique. It is also possible to start with an expression for y'_2 and derive an expression for x_3 according to

$$x_3 = \frac{(\mathbf{r}_2 - y'_2\,\mathbf{r}_3) \cdot \mathbf{t}}{(\mathbf{r}_2 - y'_2\,\mathbf{r}_3) \cdot \mathbf{y}}$$

In the ideal case, when the camera maps the 3D points according to a perfect pinhole camera and the resulting 2D points can be detected without any noise, the two expressions for x_3 are equal. In practice, however, they are not and it may be advantageous to combine the two estimates of x_3 , for example, in terms of some sort of average.

There are also other types of extensions of the above computations which are possible. They started with an expression of the primed image coordinates and derived 3D coordinates in the unprimed system. It is also possible to start with unprimed image coordinates and obtain primed 3D coordinates, which finally can be transformed into unprimed 3D coordinates. Again, in the ideal case the result should be equal to the above expressions, but in practice they may deviate.

A final remark relates to the fact that if the essential matrix is determined from corresponding image coordinate, which often is the case when 3D points are determined in this way, the translation vector $\mathbf{t}$ is known only up to an unknown positive scaling. As a consequence, the reconstructed 3D points, too, are undetermined with respect to a positive scaling.

9.8 References

- David Nistér (June 2004). "An efficient solution to the five-point relative pose problem". *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26** (6): 756–777. doi:10.1109/TPAMI.2004.17. PMID 18579936.
- H. Christopher Longuet-Higgins (September 1981). "A computer algorithm for reconstructing a scene from two projections". *Nature* **293** (5828): 133–135. doi:10.1038/293133a0.
- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*. Cambridge University Press. ISBN 978-0-521-54051-3.
- Yi Ma; Stefano Soatto; Jana Košecká; S. Shankar Sastry (2004). *An Invitation to 3-D Vision*. Springer. ISBN 0-387-00893-4.
- Gang Xu and Zhengyou Zhang (1996). *Epipolar geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers. ISBN 978-0-7923-4199-4.

9.9 Text and image sources, contributors, and licenses

9.9.1 Text

- **Computer stereo vision** *Source:* https://en.wikipedia.org/wiki/Computer_stereo_vision?oldid=677195163 *Contributors:* Richie, Sgauria, Toussaint, Kri, Banus, KYN, Gilliam, Mion, Magioladitis, User A1, Cubic333, 1ForTheMoney, WikHead, Fgnievinski, Fotaun, Gvitaly, Onoes, Gwestheimer, Thepigdog, Fluffystar, Avi.nehemiah, JuliusHuelsmann and Anonymous: 13
- **Projective space** *Source:* https://en.wikipedia.org/wiki/Projective_space?oldid=673292836 *Contributors:* AxelBoldt, Zundark, XJaM, JeLuF, Edward, Michael Hardy, Charles Matthews, Dosei, Rvollmert, Lowellian, MathMartin, Tobias Bergemann, Giftlite, BenFrantzDale, Lethe, Herbee, Gro-Tsen, Pmanderson, Almit39, Rich Farmbrough, Qutezuce, Zaslav, Pt, Rgdboer, Shenme, Tsirel, Oleg Alexandrov, BD2412, MarSch, R.e.b., John Z, Nimur, David H Braun (1964), YurikBot, RussBot, KSmrq, Archelon, Crasshopper, FRR~enwiki, Nekura, DVD R W, SmackBot, RDBury, KYN, Molinagaray, Nbarth, Tamfang, Daqu, Eliyak, Joerg Winkelmann~enwiki, Thijs!bot, Aquishix, Escarbot, Steelpillow, Almwi, Magioladitis, Jakob.scholbach, J0equ1nn, Vinograd19, TechnoFaye, Maurice Carbonaro, Yonidebot, Jonathanzung, TomyDuby, Shiznick, Caiodnh, YoungFrog, Davidsevilla, LokiClock, TXiKiBoT, Voorlandt, Kmhkmh, Arcfrk, Moonriddengirl, Lucasbfrbot, JerroldPease-Atlanta, Sfan00 IMG, Yasmar, He7d3r, Hans Adler, Addbot, Ollydbg, Luckas-bot, Yobot, Ht686rg90, Amirobot, AnomieBOT, Ciphers, Citation bot, Andersæøå, Howard McCay, Sławomir Biały, Rdragn, Trappist the monk, Mc laudt, Slawekb, Charles Staats, Quondum, D.Lazard, Wikfr, Maximilianklein, Crown Prince, Movyn, Wcherowi, Helpful Pixie Bot, Gilgoldm, Nuermann, Quantrillo, Nicojonesgodel, JPaestpreornJeolhlna, Cosmia Nebula and Anonymous: 35
- **Cardinal point (optics)** *Source:* https://en.wikipedia.org/wiki/Cardinal_point_(optics)?oldid=645003827 *Contributors:* DrBob, Leandrod, BenFrantzDale, Cynical, Hooperbloob, Jeltz, Srleffler, YurikBot, ZabMilenko, SmackBot, Melchoir, Elagatis, Kostmo, Fmlburnay, Roguegeek, Gonioul, Severoon, Pflatau, Misteror, Dicklyon, Garvin58, ShelfSkewed, Thijs!bot, AntiVandalBot, Richard Giuly, DinoBot, R'n'B, Panoramikz~enwiki, Cometstyles, YonaBot, Odo Benus, Anchor Link Bot, Arvind Guru - Zetta, Addbot, Debresser, Yobot, AnomieBOT, 1exec1, Redbobblehat, Piano non troppo, Citation bot, HHahn, Justinlee22, Citation bot 1, Tamasflex, Lotje, Vajr, GianniG46, TedEyeMD, ClueBot NG, Helpful Pixie Bot, Bertoche, Nigellwh and Anonymous: 36
- **Triangulation (computer vision)** *Source:* https://en.wikipedia.org/wiki/Triangulation_(computer_vision)?oldid=669665640 *Contributors:* Scatter~enwiki, ALoopingIcon, KYN, Chris the speller, Kostmo, Iridescent, Klapi, Afaz, ChrisHodgesUK, Addbot, Fgnievinski, Licyoon, Yobot, Mentibot, Khazar2 and Anonymous: 7
- **Pinhole camera model** *Source:* https://en.wikipedia.org/wiki/Pinhole_camera_model?oldid=669667598 *Contributors:* Leandrod, Stefan-S, LMB, BenFrantzDale, Elmarco, Srleffler, Grafen, SmackBot, KYN, Dicklyon, Dr.K., PhiLiP, Bakasuprman, Pbroks13, Philip Trueman, SCriBu, Kolyma, Fgnievinski, Balabiot, Ettrig, Redbobblehat, Theoprakt, Citation bot, FrescoBot, Checkingfax, Normvcr, Vikram360, ClueBot NG, BattyBot, MadGuy7023 and Anonymous: 20
- **Camera matrix** *Source:* https://en.wikipedia.org/wiki/Camera_matrix?oldid=669010121 *Contributors:* BenFrantzDale, Rgdboer, Ruziklan, Glopk, KYN, Nbarth, Slysplace, Lourakis, Jilltoronto, Pibben, Baudway, Addbot, Louperibot, Josve05a, Zfeinst and Anonymous: 7
- **Epipolar geometry** *Source:* https://en.wikipedia.org/wiki/Epipolar_geometry?oldid=669665883 *Contributors:* Michael Hardy, BenFrantzDale, Norro, Waldir, Page~enwiki, SmackBot, KYN, Fuzzypeg, Disavian, Kfrance, ShelfSkewed, Alaibot, LibLord, Sider82083, User A1, Sensenmann, Singularitarian, Des Curly, Unbound, TXiKiBoT, Lgrove, Mcmcvick, Ericmccreath, Mleconte, Fnymnky, ZooFari, Addbot, Fgnievinski, Materialscientist, Citation bot, Prunesqualer, WildBot, ClueBot NG, Snotbot, Helpful Pixie Bot, Gottogo95, Jtlehtin, Dexbot, LokeshRavindranathan, KasparBot and Anonymous: 31
- **Fundamental matrix (computer vision)** *Source:* https://en.wikipedia.org/wiki/Fundamental_matrix_(computer_vision)?oldid=683535179 *Contributors:* The Anome, Michael Hardy, Greenrd, MathMartin, Giftlite, BenFrantzDale, Rgdboer, Cyc~enwiki, Tbsmith, Jemiller226, Rjwilmsi, KYN, Kostmo, Kleuske, Arsalank2, Sbressler, Hendrixjoseph, Joaoluiscarreira, Caltas, Lourakis, PipepBot, BOTarate, Zhengyou zhang, DOI bot, Citation bot, Mrcricket48, Citation bot 1, Gnitpicker, Trappist the monk, RjwilmsiBot, Helpful Pixie Bot, KLBot2, Gottogo95, Dexbot, Kauevestena and Anonymous: 20
- **Essential matrix** *Source:* https://en.wikipedia.org/wiki/Essential_matrix?oldid=673349219 *Contributors:* Edward, MathMartin, BenFrantzDale, Joriki, Tabletop, Jemiller226, Rjwilmsi, AndyKali, Nihiltres, Intgr, Gadget850, SmackBot, KYN, Chris the speller, Vendettax, Slysplace, Lourakis, Iknowyourider, Chaosdruid, Smf68, DOI bot, Download, Citation bot, Citation bot 1, Trappist the monk, Jfhullo, Zfeinst, Helpful Pixie Bot, Kalleastrom, Betheev, Monkbot, Cansen Jiang and Anonymous: 21

9.9.2 Images

- **File:Aufnahme_mit_zwei_Kameras.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/7c/Aufnahme_mit_zwei_Kameras.svg *License:* CC BY-SA 3.0 *Contributors:* Own illustration *Original artist:* Arne Nordmann (norro)
- **File:BackFocalPlane.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/47/BackFocalPlane.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:BackFocalPlane_aperture.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/6/6f/BackFocalPlane_aperture.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Cardinal-points-1.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/e3/Cardinal-points-1.svg *License:* CC-BY-SA-3.0 *Contributors:* en:Image:Cardinal-points-1.svg *Original artist:* en:User:DrBob
- **File:Cardinal-points-2.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/7/78/Cardinal-points-2.svg *License:* Cc-by-sa-3.0 *Contributors:*

  Bob Mellish (talk) (Uploads) *Original artist:*

  Bob Mellish (talk) (Uploads)
- **File:Crystal_Clear_app_kedit.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/e8/Crystal_Clear_app_kedit.svg *License:* LGPL *Contributors:* Sabine MINICONI *Original artist:* Sabine MINICONI

- **File:Edit-clear.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/f/f2/Edit-clear.svg *License:* Public domain *Contributors:* The *Tango! Desktop Project*. *Original artist:*
  The people from the Tango! project. And according to the meta-data in the file, specifically: “Andreas Nilsson, and Jakub Steiner (although minimally).”
- **File:Epipolar_Geometry1.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/f2/Epipolar_Geometry1.svg *License:* Public domain *Contributors:* Own work *Original artist:* ZooFari
- **File:Epipolar_geometry.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/14/Epipolar_geometry.svg *License:* CC-BY-SA-3.0 *Contributors:* Own work (Own drawing) *Original artist:* Arne Nordmann (norro)
- **File:Fano_plane.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/af/Fano_plane.svg *License:* Public domain *Contributors:* created using vi *Original artist:* de:User:Gunther
- **File:Lens_shapes.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/01/Lens_shapes.svg *License:* GFDL *Contributors:* *Original artist:* Fred the Oyster
- **File:P1_ako_varieta.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/6/62/P1_ako_varieta.png *License:* CC-BY-SA-3.0 *Contributors:* Transferred from en.wikipedia to Commons. *Original artist:* Jakob.scholbach at English Wikipedia
- **File:P1_dalsia_predstava.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/b5/P1_dalsia_predstava.png *License:* CC BY-SA 3.0 *Contributors:* Anglická wikipédia *Original artist:* Jakob.scholbach (http://en.wikipedia.org/wiki/User:Jakob.scholbach)
- **File:Pinhole-camera.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3b/Pinhole-camera.svg *License:* Public domain *Contributors:* http://commons.wikimedia.org/wiki/Image:Pinhole-camera.png *Original artist:* en:User:DrBob (original); en:User:Pbroks13 (redraw)
- **File:Pinhole.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/7/7c/Pinhole.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Pinhole2.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/f/f8/Pinhole2.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Priklad_pre_Bezoutov_teorem.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/4c/Priklad_pre_Bezoutov_teorem.png *License:* CC BY-SA 3.0 *Contributors:* Anglická wikipédia *Original artist:* Jakob.scholbach (http://en.wikipedia.org/wiki/User:Jakob.scholbach)
- **File:Projective_plane2.png** *Source:* https://upload.wikimedia.org/wikipedia/en/d/d8/Projective_plane2.png *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Projektivna_rovina.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/40/Projektivna_rovina.png *License:* CC-BY-SA-3.0 *Contributors:* Transferred from en.wikipedia to Commons. *Original artist:* Jakob.scholbach at English Wikipedia
- **File:Question_book-new.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/9/99/Question_book-new.svg *License:* Cc-by-sa-3.0 *Contributors:*
  Created from scratch in Adobe Illustrator. Based on Image:Question book.png created by User:Equazcion *Original artist:*
  Tkgd2007
- **File:Railroad-Tracks-Perspective.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/2a/Railroad-Tracks-Perspective.jpg *License:* Public domain *Contributors:* Own work *Original artist:* User:MikKBDFJKGeMalak
- **File:Stereoscopic_images,_depth_to_displacement_relationship_assuming_flat_co-planar_images..png** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/b9/Stereoscopic_images%2C_depth_to_displacement_relationship_assuming_flat_co-planar_images..png *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Thepigdog
- **File:Tesseract.gif** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/55/Tesseract.gif *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Text_document_with_red_question_mark.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a4/Text_document_with_red_question_mark.svg *License:* Public domain *Contributors:* Created by bdesham with Inkscape; based upon Text-x-generic.svg from the Tango project. *Original artist:* Benjamin D. Esham (bdesham)
- **File:TriangulationIdeal.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/2/2c/TriangulationIdeal.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:TriangulationReal.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/a/ae/TriangulationReal.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Wiki_letter_w_cropped.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1c/Wiki_letter_w_cropped.svg *License:* CC-BY-SA-3.0 *Contributors:*
- Wiki_letter_w.svg *Original artist:* Wiki_letter_w.svg: Jarkko Piiroinen

9.9.3 Content license