Elements of Computer Vision: Multiple View Geometry.

Andrea Fusiello
http://www.sci.univr.it/~fusiello

June 20, 2005

1 Introduction

This chapter introduces the basic geometric concepts of multiple-view computer vision. The focus is on geometric models of perspective cameras, and the constraints and properties such models generate when multiple cameras observe the same 3D scene.

Geometric vision is an important and well-studied part of computer vision. A wealth of useful results has been achieved in the last 15 years and has been reported in comprehensive monographies, e.g., [4, 11, 6], a sign of maturity for a research subject.

It is worth reminding the reader that geometry is an important but not the only important aspect of computer vision, and in particular of multi-view vision. The information brought by each image pixel is twofold: its *position* and its *colour* (or brightness, for a monochrome image). Ultimately, each computer vision system must start with brightness values, and, to smaller or greater depth, link such values to the 3D world.

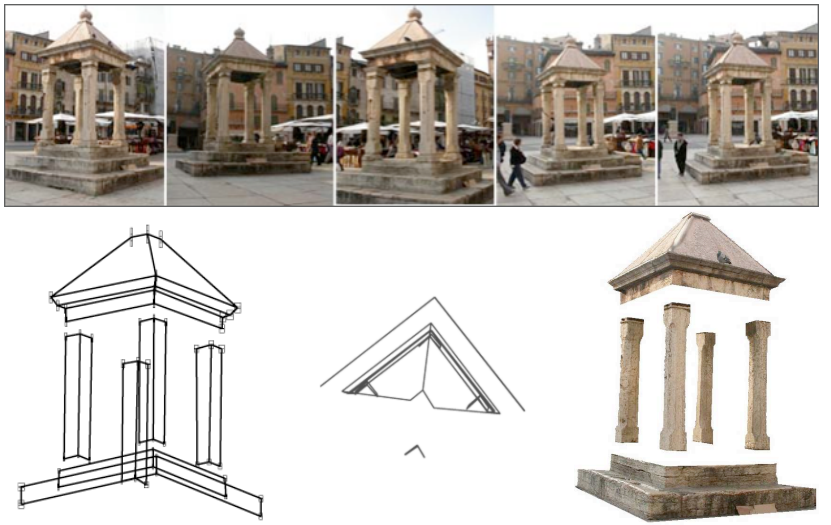


Fig. 1. Example of reconstruction from the five images shown in the top row.

2 Elements of Geometry

The ambient space is modellled as a projective 3-D space $\mathbb{P}_3$, obtained by completing the affine space $\mathbb{X}_3$ with a projective plane, known as plane at infinity Π_∞. In this ideal plane lie the intersections of the planes parallel in X_3.

The projective coordinates of a point in $\mathbb{P}_3$ are 4-tuples defined up to a scale factor. We write

$$\mathbf{M} \simeq (x, y, z, t) \tag{1}$$

where $\simeq$ indicates equality to within a multiplicative factor.

Π_∞ is defined by its equation $t = 0$.

The points of the affine space are those of $\mathbb{P}_3$ which do not belong to Π_∞. Their projective coordinates are thus of the form $(x, y, z, 1)$, where (x, y, z) are the usual affine coordinates.

The linear tranformations of a projective space into itself are called collineations of homographies. Any collineation of $\mathbb{P}_3$ is represented by a generic 4×4 invertible matrix.

Affine transformations of $\mathbb{X}_3$ are the subgroup of collineations of $\mathbb{P}_3$ that preserves the plane at infinity.

Similarity transformations are the subgroup of affine transformations of $\mathbb{X}_3$ that leave invariant a very special curve, the absolute conic, which is in the plane at infinity and whose equation is:

$$x^2 + y^2 + z^2 = 0 \tag{2}$$

The space is therefore stratified into more and more specialized structures:

- projective
- affine (knowing the plane at infinity)
- euclidean (knowing the absolute conic)

The stratification reflects the amount of knowledge that we possess about the scene and the sensor.

3 Pin-hole Camera Geometry

The pin-hole camera is described by its *optical centre* $\mathbf{C}$ (also known as *camera projection centre*) and the *image plane*.

The distance of the image plane from $\mathbf{C}$ is the *focal length* f.

The line from the camera centre perpendicular to the image plane is called the *principal axis* or *optical axis* of the camera.

The plane parallel to the image plane containing the optical centre is called the *principal plane* or *focal plane* of the camera.

The relationship between the 3D coordinates of a scene point and the coordinates of its projection onto the image plane is described by the *central* or *perspective projection*.

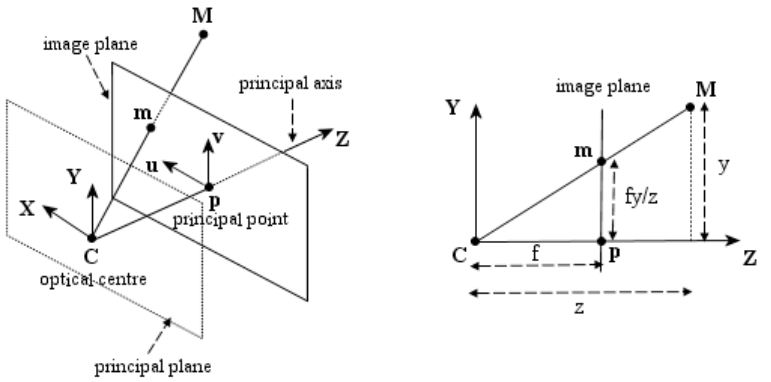


Fig. 2. Pin-hole camera geometry. The left figure illustrates the projection of the point **M** on the image plane by drawing the line through the camera centre **C** and the point to be projected. The right figure illustrates the same situation in the YZ plane, showing the similar triangles used to compute the position of the projected point **m** in the image plane.

A 3D point is projected onto the image plane with the line containing the point and the optical centre (see Figure 2).

Let the centre of projection be the origin of a Euclidean coordinate system wherein the z-axis is the principal axis.

By similar triangles it is readily seen that the 3D point $(x, y, z)^T$ is mapped to the point $(fx/z, fy/z, f)^T$ on the image plane.

3.1 The camera projection matrix

If the world and image points are represented by homogeneous vectors, then perspective projection can be expressed in terms of matrix multiplication as

$$\begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{3}$$

The matrix describing the mapping is called the *camera projection matrix* P.

Equation (3) can be written simply as:

$$z\mathbf{m} = P\mathbf{M} \tag{4}$$

where $\mathbf{M} = (x, y, z, 1)^T$ are the homogeneous coordinates of the 3D point and $\mathbf{m} = (fx/z, fy/z, 1)^T$ are the homogeneous coordinates of the image point.

The projection matrix P in Equation (3) represents the simplest possible case, as it only contains information about the focal distance f.

General camera: bottom up approach

The above formulation assumes a special choice of world coordinate system and image coordinate system. It can be generalized by introducing suitable changes of the coordinates systems.

Changing coordinates in space is equivalent to multiplying the matrix P to the right by a 4×4 matrix:

$$G = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{5}$$

G is composed by a rotation matrix R and a translation vector $\mathbf{t}$. It describes the position and orientation of the camera with respect to an external (world) coordinate system. It depends on six parameters, called *external* parameters.

Changing coordinates in the image plane is equivalent to multiplying the matrix P to the left by a 3×3 matrix:

$$K = \begin{bmatrix} f/s_x & f/s_x \cot\theta & o_x \\ 0 & f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

K is the *camera calibration matrix*; it encodes the transformation from image coordinates to pixel coordinates in the image plane.

It depends on the so-called *intrinsic* parameters:

- focal distance f (in mm),
- image centre coordinates o_x, o_y (in pixel),
- width (s_x) and height (s_y) of the pixel footprint on the camera photosensor (in mm),
- angle θ between the axes (usually $\pi/2$).

The ratio s_y/s_x is the aspect ratio (usually close to 1).

Thus the camera matrix, in general, is the product of three matrices:

$$P = K[I|\mathbf{0}]G = K[R|\mathbf{t}] \tag{7}$$

In general, the projection equation writes:

$$\zeta\mathbf{m} = P\mathbf{M} \tag{8}$$

where ζ is the distance of $\mathbf{M}$ from the focal plane of the camera (this will be shown after).

Note that, except for a very special choice of the world reference frame, *this "depth" does not coincide with the third coordinate of* $\mathbf{M}$.

General camera: top down approach

If P describes a camera, also λP for any $0 \neq \lambda \in \mathbb{R}$ describes the same camera, since these give the same image point for each scene point.

In this case we can also write:

$$\mathbf{m} \simeq P\mathbf{M} \tag{9}$$

where $\simeq$ means "equal up to a scale factor."

In general, the camera projection matrix is a 3×4 full-rank matrix and, being homogeneous, it has 11 degrees of freedom.

Using QR factorization, it can be shown that any 3×4 full rank matrix P can be factorised as:

$$P = \lambda K[R|\mathbf{t}], \tag{10}$$

(λ is defined because $K(3,3) = 1$).

3.2 Camera anatomy

Projection centre

The camera projection centre $\mathbf{C}$ is the only point for which the projection is not defined, i.e.:

$$P\mathbf{C} = P\begin{pmatrix}\tilde{\mathbf{C}}\\ 1\end{pmatrix} = \mathbf{0} \tag{11}$$

where $\tilde{\mathbf{C}}$ is a three-vector containing the affine (non-homogeneous) coordinates of the optical centre.

After solving for $\tilde{\mathbf{C}}$ we obtain:

$$\tilde{\mathbf{C}} = -P_{3\times3}^{-1}\mathbf{p}_4 \tag{12}$$

where the matrix P is represented by the block form: $P = [P_{3\times3}|\mathbf{p}_4]$ ($P_{3\times3}$ is the matrix composed by the first three rows and first three columns of P, and $\mathbf{p}_4$ is the fourth column of P).

A few words about normalization

If $\lambda = 1$ in Eq. (10), the matrix P is said to be *normalized*.

If (and only if) the matrix is normalized, then ζ in the projection equation (8) is the distance of $\mathbf{M}$ from the focal plane of the camera (usually referred to as *depth*).

We observe that:

$$\zeta\mathbf{m} = P\mathbf{M} = P\mathbf{M} - P\mathbf{C} = P(\mathbf{M} - \mathbf{C}) = P_{3\times3}(\tilde{\mathbf{M}} - \tilde{\mathbf{C}}).$$

In particular, the third component is $\zeta = \mathbf{p}_3^T(\tilde{\mathbf{M}} - \tilde{\mathbf{C}})$, where $\mathbf{p}_3^T$ is the third row of $P_{3\times3}$.

If write R in terms of its rows and multiply in Eq. (7) we see that $\mathbf{p}_3^T$ is the third row of R, which correspond to the versor of the principal axis.

Hence, the previous equations says that ζ is the projection of the vector $(\tilde{\mathbf{M}} - \tilde{\mathbf{C}})$ onto the principal axix, i.e., the depth of $\mathbf{M}$.

In general (when the camera is not normalized), ζ contains an arbitrary scale factor. Can we recover this scale factor from a generic camera matrix P without factorizing it like in Eq. (10)?

We only need to observe that if P is given by Eq. (10), $\mathbf{p}_3^T$ is the third row of R multiplied by the scale factor λ. Hence, $\lambda = ||\mathbf{p}_3||$.

Optical ray

The projection can be geometrically modelled by a ray through the optical centre and the point in space that is being projected onto the image plane (see Fig. 2).

The *optical ray* of an image point $\mathbf{m} = (u, v, 1)^T$ is the locus of points in space that projects onto $\mathbf{m}$.

It can be described as a parametric line passing through the camera projection centre $\mathbf{C}$ and a special point (at infinity) that projects onto $\mathbf{m}$:

$$\mathbf{M} = \begin{pmatrix} -P_{3\times3}^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix} + \zeta \begin{pmatrix} P_{3\times3}^{-1}\mathbf{m} \\ 0 \end{pmatrix}, \quad \zeta \in \mathbb{R}. \tag{13}$$

Please note that, provided that P is normalized, the parameter ζ in the equation of the optical ray correspond to the depth of the point $\mathbf{M}$.

3.2.1 Image of the absolute conic

We will prove now that the image of the absolute conic depends on the intrinsic parameters only (it is unaffected by camera position and orientation).

The points in the plane at infinity have the form $\mathbf{M} = [\tilde{\mathbf{M}}, 0]^T$, hence

$$\mathbf{m} = K[R \mid \mathbf{t}][\tilde{\mathbf{M}}0]^T = KR\tilde{\mathbf{M}}^T. \tag{14}$$

The image of points on the plane at infinity does not depend on camera position (it is unaffected by camera translation).

The absolute conic (which is in the plane at infinity) has equation $\tilde{\mathbf{M}}^T\tilde{\mathbf{M}} = 0$, therefore its projection has equation:

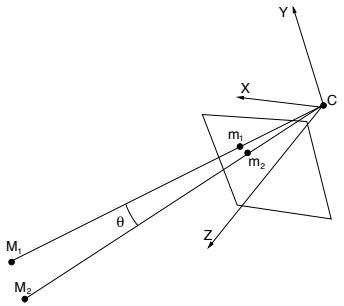
$$\mathbf{m}^T(K^{-T}K^{-1})\mathbf{m} = 0. \tag{15}$$

The conic $\boldsymbol{\omega} = (KK^{-T})^{-1}$ is the image of the absolute conic.

The angle (a metrical property) between two rays is determined by the image of the absolute conic.

Let us consider a camera $P = [K|\mathbf{0}]$, hence $\mathbf{m} = K\tilde{\mathbf{M}}$. Let θ be the angle bewtween the rays trough $\mathbf{M}_1$ and $\mathbf{M}_1$, then

$$\cos\theta = \frac{\tilde{\mathbf{M}}_1^T\tilde{\mathbf{M}}_2}{||\tilde{\mathbf{M}}_1||||\tilde{\mathbf{M}}_2||} = \frac{\mathbf{m}_1^T\boldsymbol{\omega}\mathbf{m}_2}{\sqrt{\mathbf{m}_1^T\boldsymbol{\omega}\mathbf{m}_1}\sqrt{\mathbf{m}_2^T\boldsymbol{\omega}\mathbf{m}_2}}$$



3.3 Camera calibration (or resection)

A number of point correspondences $\mathbf{m}_i \leftrightarrow \mathbf{M}_i$ is given, and we are required to find a camera matrix P such that

$$\mathbf{m}_i \simeq P\mathbf{M}_i \quad \text{for all } i. \tag{16}$$

The equation can be rewritten in terms of the cross product as

$$\mathbf{m}_i \times P\mathbf{M}_i = \mathbf{0}. \tag{17}$$

This form will enable a simple a simple linear solution for P to be derived. Using the properties of the Kronecker product ($\otimes$) and the vec operator, we derive:

$$\begin{aligned} &\mathbf{m}_i \times P\mathbf{M}_i = \mathbf{0} \iff [\mathbf{m}_i]_\times P\mathbf{M}_i = \mathbf{0} \iff \mathrm{vec}([\mathbf{m}_i]_\times P\mathbf{M}_i) = \mathbf{0} \iff \\ &\iff (\mathbf{M}_i^T \otimes [\mathbf{m}_i]_\times)\,\mathrm{vec}\,P = \mathbf{0} \iff ([\mathbf{m}_i]_\times \otimes \mathbf{M}_i^T)\,\mathrm{vec}\,P^T = \mathbf{0} \end{aligned} \tag{18}$$

After expanding the coefficient matrix, we obtain

$$\begin{bmatrix} \mathbf{0}^T & -\mathbf{M}_i^T & v\mathbf{M}_i^T \\ \mathbf{M}_i^T & \mathbf{0}^T & -u\mathbf{M}_i^T \\ -v\mathbf{M}_i^T & u\mathbf{M}_i^T & \mathbf{0}^T \end{bmatrix} \mathrm{vec}\,P^T = \mathbf{0} \tag{19}$$

Although there are three equations, only two of them are linearly independent: we can write the third row (e.g.) as a linear combination of the first two.

From a set of n point correspondences, we obtain a $2n \times 12$ coefficient matrix A by stacking up two equations for each correspondence. The projection matrix P is computed by solving the resulting linear system of equations, for $n \geq 6$.

In general A will have rank 11 (provided that the points are not all coplanar) and the solution is the 1-dimensional right null-space of A.

If the data are not exact (noise is generally present) the rank of A will be 12 and a least-squares solution is sought.

The least-squares solution for $\mathrm{vec}\,P^T$ is the singular vector corresponding to the smallest singular value of A.

This is called the Direct Linear Transform (DLT) algorithm [11].

4 Two-View Geometry

The two-view geometry is the intrinsic geometry of two different perspective views of the same 3D scene (see Figure 3). It is usually referred to as *epipolar geometry*.

The two perspective views may be acquired simultaneously, for example in a stereo rig, or sequentially, for example by a moving camera. From the geometric viewpoint, the two situations are equivalent, but notice that the scene might change between successive snapshots.

Most 3D scene points must be visible in both views simultaneously. This is not true in case of occlusions, i.e., points visible only in one camera. Any unoccluded 3D scene point $\mathbf{M} = (x, y, z, 1)^T$ is projected to the left and right view as $\mathbf{m}_\ell = (u_\ell, v_\ell, 1)^T$ and $\mathbf{m}_r = (u_r, v_r, 1)^T$, respectively (see Figure 3).

Image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ are called *corresponding points* (or conjugate points) as they represent projections of the same 3D scene point $\mathbf{M}$.

The knowledge of image correspondences enables scene reconstruction from images.

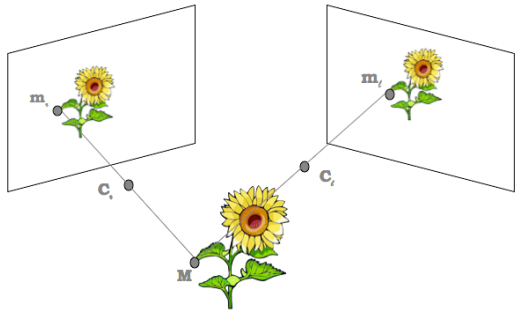


Fig. 3. Two perspective views of the same 3D scene. $\mathbf{m}_\ell$ and $\mathbf{m}_r$ are corresponding points, as they are the projection od the same 3D point, $\mathbf{M}$.

Algebraically, each perspective view has an associated 3×4 camera projection matrix P which represents the mapping between the 3D world and a 2D image. We will refer to the camera projection matrix of the left view as P_ℓ and of the right view as P_r. The 3D point $\mathbf{M}$ is then imaged as (20) in the left view, and (21) in the right view:

$$\zeta_\ell \mathbf{m}_\ell = P_\ell \mathbf{M} \tag{20}$$

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M}. \tag{21}$$

Geometrically, the position of the image point $\mathbf{m}_\ell$ in the left image plane I_ℓ can be found by drawing the optical ray through the left camera projection centre $\mathbf{C}_\ell$ and the scene point $\mathbf{M}$. The ray intersects the left image plane I_ℓ at $\mathbf{m}_\ell$. Similarly, the optical ray connecting $\mathbf{C}_r$ and $\mathbf{M}$ intersects the right image plane I_r at $\mathbf{m}_r$. The relationship between image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ is given by the epipolar geometry, described in Section 4.1.

4.1 Epipolar Geometry

The epipolar geometry describes the geometric relationship between two perspective views of the same 3D scene.

The key finding, discussed below, is that *corresponding image points must lie on particular image lines*, which can be computed without information on the calibration of the cameras.

This implies that, given a point in one image, one can search the corresponding point in the other along a line and not in a 2D region, a significant reduction in complexity.

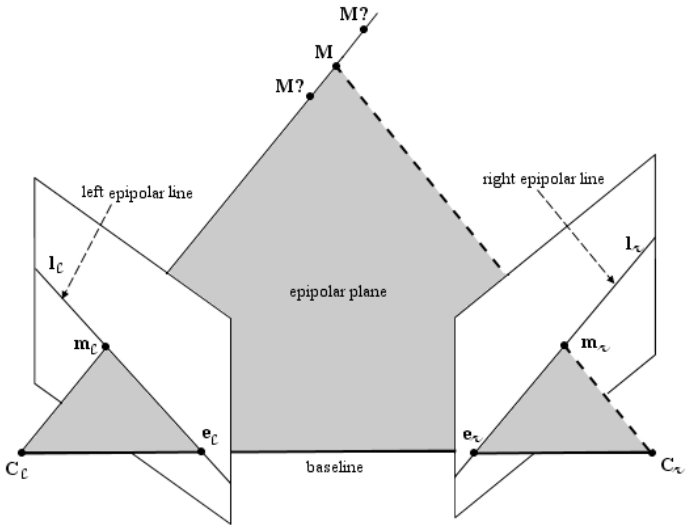


Fig. 4. The epipolar geometry and epipolar constraint.

Any 3D point $\mathbf{M}$ and the camera projection centres $\mathbf{C}_\ell$ and $\mathbf{C}_r$ define a plane that is called *epipolar plane*.

The projections of the point $\mathbf{M}$, image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$, also lie in the epipolar plane since they lie on the rays connecting the corresponding camera projection centre and point $\mathbf{M}$.

The conjugate epipolar lines, $\mathbf{l}_\ell$ and $\mathbf{l}_r$, are the intersections of the epipolar plane with the image planes. The line connecting the camera projection centres $(\mathbf{C}_\ell, \mathbf{C}_r)$ is called the *baseline*.

The baseline intersects each image plane in a point called *epipole*.

By construction, the left epipole $\mathbf{e}_\ell$ is the image of the right camera projection centre $\mathbf{C}_r$ in the left image plane. Similarly, the right epipole $\mathbf{e}_r$ is the image of the left camera projection centre $\mathbf{C}_\ell$ in the right image plane.

All epipolar lines in the left image go through $\mathbf{e}_\ell$ and all epipolar lines in the right image go through $\mathbf{e}_r$.

The epipolar constraint. An epipolar plane is completely defined by the camera projection centres and one image point.

Therefore, given a point $\mathbf{m}_\ell$, one can determine the epipolar line in the right image on which the corresponding point, $\mathbf{m}_r$, must lie.

The equation of the epipolar line can be derived from the equation describing the optical ray. As we mentioned before, the right epipolar line corresponding to $\mathbf{m}_\ell$ geometrically represents the projection (Equation (8)) of the optical ray through $\mathbf{m}_\ell$ (Equation (13)) onto the right image plane:

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M} = \underbrace{P_r \begin{pmatrix} -P_{3\times3,\ell}^{-1}\mathbf{p}_{4,\ell} \\ 1 \end{pmatrix}}_{\mathbf{e}_r} + \zeta_\ell P_r \begin{pmatrix} P_{3\times3,\ell}^{-1}\mathbf{m}_\ell \\ 0 \end{pmatrix} \tag{22}$$

If we now simplify the above equation we obtain the description of the right epipolar line:

$$\zeta_r \mathbf{m}_r = \mathbf{e}_r + \zeta_\ell P_{3\times3,r} P_{3\times3,l}^{-1} \mathbf{m}_\ell \tag{23}$$

This is the equation of a line through the right epipole $\mathbf{e}_r$ and the image point $\mathbf{m}'_\ell = P_{3\times3,r} P_{3\times3,l}^{-1} \mathbf{m}_\ell$ which represents the projection onto the right image plane of the point at infinity of the optical ray of $\mathbf{m}_\ell$.

The equation for the left epipolar line is obtained in a similar way.

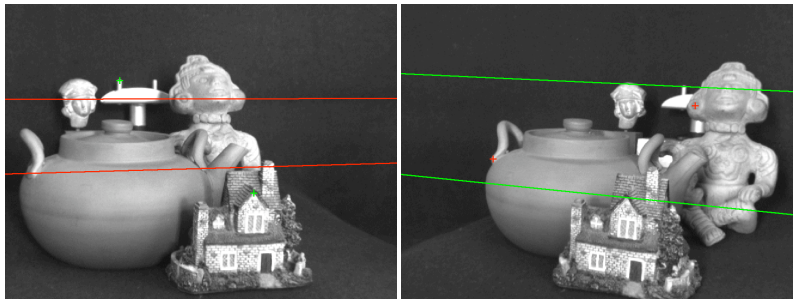


Fig. 5. Left and right images with epipolar lines.

The epipolar geometry can be described analytically in several ways, depending on the amount of the *a priori* knowledge about the stereo system. We can identify three general cases.

If both *intrinsic* and *extrinsic* camera parameters are known, we can describe the epipolar geometry in terms of the projection matrices (Equation (23)).

If only the *intrinsic* parameters are known, we work in normalised coordinates and the epipolar geometry is described by the *essential matrix*.

If neither intrinsic nor extrinsic parameters are known the epipolar geometry is described by the *fundamental matrix*.

4.1.1 The Essential Matrix E

If the intrinsic parameters are known, we can switch to *normalised coordinates*: $\mathbf{m} \leftarrow K^{-1}\mathbf{m}$ (please note that this change of notation will hold throughout this section).

Consider a pair of normalised cameras. Without loss of generality, we can fix the world reference frame onto the first camera, hence:

$$P_\ell = [I|0] \quad \text{and} \quad P_r = [R|\mathbf{t}]. \tag{24}$$

With this choice, the unknown extrinsic parameters have been made explicit.

If we substitute these two particular instances of the camera projection matrices in Equation (23), we get

$$\zeta_r \mathbf{m}_r = \mathbf{t} + \zeta_\ell R \mathbf{m}_\ell; \tag{25}$$

in other words, the point $\mathbf{m}_r$ lies on the line through the points $\mathbf{t}$ and $R\mathbf{m}_\ell$. In homogeneous coordinates, this can be written as follows:

$$\mathbf{m}_r^T \mathbf{t} \times (R\mathbf{m}_\ell) = 0, \tag{26}$$

as the homogeneous line through two points is expressed as their cross product.

Similarly, a dot product of a point and a line is zero if the point lies on the line.

The cross product of two vectors can be written as a product of a skew-symmetric matrix and a vector. Equation (26) can therefore be equivalently written as

$$\mathbf{m}_r^T [\mathbf{t}]_\times R \mathbf{m}_\ell = 0, \tag{27}$$

where $[\mathbf{t}]_\times$ is the skew-symmetric matrix of the vector $\mathbf{t}$. If we multiply the matrices in the above equation, we obtain a single matrix which describes the relationship between the corresponding image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ in normalised coordinates. This matrix is called the *essential matrix* E:

$$E \triangleq [\mathbf{t}]_\times R, \tag{28}$$

and the relationship between two corresponding image points in normalised coordinates is expressed by the defining equation for the essential matrix:

$$\mathbf{m}_r^T E \mathbf{m}_\ell = 0. \tag{29}$$

E encodes only information on the extrinsic camera parameters. Its rank is two, since $\det[\mathbf{t}]_\times = 0$. The essential matrix is a homogeneous quantity. It has only five degrees of freedom: a 3D rotation and a 3D translation direction.

4.1.2 The Fundamental Matrix F

The fundamental matrix can be derived in a similar way to the essential matrix. All camera parameters are assumed unknown; we write therefore a general version of Equation (24):

$$P_\ell = K_\ell [I|0] \quad \text{and} \quad P_r = K_r [R|\mathbf{t}]. \tag{30}$$

Inserting these two projection matrices into Equation (23), we get

$$\zeta_r \mathbf{m}_r = \mathbf{e}_r + \zeta_\ell K_r R K_\ell^{-1} \mathbf{m}_\ell \quad \text{with} \quad \mathbf{e}_r = K_r \mathbf{t}, \tag{31}$$

which states that point $\mathbf{m}_r$ lies on the line through $\mathbf{e}_r$ and $K_r R K_\ell^{-1} \mathbf{m}_\ell$. As in the case of the essential matrix, this can be written in homogeneous coordinates as:

$$\mathbf{m}_r^T [\mathbf{e}_r]_\times K_r R K_\ell^{-1} \mathbf{m}_\ell = 0. \tag{32}$$

The matrix

$$F = [\mathbf{e}_r]_\times K_r R K_\ell^{-1} \tag{33}$$

is the *fundamental matrix* F, giving the relationship between the corresponding image points in pixel coordinates.

The defining equation for the fundamental matrix is therefore

$$\mathbf{m}_r^T F \mathbf{m}_\ell = 0. \tag{34}$$

F is the algebraic representation of the epipolar geometry. It is a 3×3, rank-two homogeneous matrix. It has only seven degrees of freedom since it is defined up to a scale and its determinant is zero. Notice that F is completely defined by pixel correspondences only (the intrinsic parameters are not needed).

For any point $\mathbf{m}_\ell$ in the left image, the corresponding epipolar line $\mathbf{l}_r$ in the right image can be expressed as

$$\mathbf{l}_r = F \mathbf{m}_\ell. \tag{35}$$

Similarly, the epipolar line $\mathbf{l}_\ell$ in the left image for the point $\mathbf{m}_r$ in the right image can be expressed as

$$\mathbf{l}_\ell = F^T \mathbf{m}_r. \tag{36}$$

The left epipole $\mathbf{e}_\ell$ is the right null-vector of the fundamental matrix and the right epipole is the left null-vector of the fundamental matrix:

$$F\mathbf{e}_\ell = 0 \tag{37}$$
$$\mathbf{e}_r^T F = 0 \tag{38}$$

One can see from the derivation that the essential and fundamental matrices are related through the camera calibration matrices K_ℓ and K_r:

$$F = K_r^{-T} E K_\ell^{-1}. \tag{39}$$

Consider a camera pair. Using the fact that if F maps points in the left image to epipolar lines in the right image, then F^T maps points in the right image to epipolar lines in the left image, Equation (31) gives:

$$\zeta_r F^T \mathbf{m}_r = \zeta_\ell (\mathbf{e}_\ell \times \mathbf{m}_\ell). \tag{40}$$

This is another way of writing the epipolar constraint: the epipolar line of $\mathbf{m}_r$ ($F^T\mathbf{m}_r$) is the line containing its corresponding point ($\mathbf{m}_\ell$) and the epipole in the left image ($\mathbf{e}_\ell$).

4.1.3 Estimating F: the eight-point algorithm

If a number of point correspondences $\mathbf{m}_\ell^i \leftrightarrow \mathbf{m}_r^i$ is given, we can use Equation (34) to compute the unknown matrix F.

We would like to convert Equation (34) from its bilinear form to a form that matches the null space problem, as in the DLT algorithm. To this end we introduce the vec operator:

$$\mathbf{m}_r^T F \mathbf{m}_\ell = 0 \iff \mathrm{vec}(\mathbf{m}_r^T F \mathbf{m}_\ell) = 0 \iff (\mathbf{m}_r^T \otimes \mathbf{m}_\ell^T)\,\mathrm{vec}\,F = 0. \tag{41}$$

Each point correspondence gives rise to one linear equation in the unknown entries of F. From a set of n point correspondences, we obtain a $n \times 9$ coefficient matrix A by stacking up one equation for each correspondence. The fundamental matrix F is computed by solving the resulting linear system of equations, for $n \geq 8$.

In general A will have rank 8 and the solution is the 1-dimensional right null-space of A.

If the data are not exact (noise is generally present) the rank of A will be 9 and a least-squares solution is sought.

The least-squares solution for $\mathrm{vec}\,F$ is the singular vector corresponding to the smallest singular value of A.

This method does not explicitly enforce F to be singular, so it must be done *a posteriori*.

Replace F by F' such that $\det F' = 0$, by forcing to zero the least singular value.

It can be shown that F' is the closest singular matrix to F in Frobenius norm.

Geometrically, the singularity constraint ensures that the epipolar lines meet in a common epipole.

This simple algorithm provides good results in many situations and can be used to initialise a variety of more accurate, iterative algorithms. Details of these can be found in [37, 41, 11].

4.2 Triangulation

Given the camera matrices P_ℓ and P_r, let $\mathbf{m}_\ell$ and $\mathbf{m}_r$ be two corresponding points satisfying the epipolar constraint $\mathbf{m}_r^T F \mathbf{m}_\ell = 0$. It follows that $\mathbf{m}_r$ lies on the epipolar line $F\mathbf{m}_\ell$ and so the two rays back-projected from image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ lie in a common epipolar plane. Since they lie in the same plane, they will intersect at some point. This point is the reconstructed 3D scene point $\mathbf{M}$.

Analytically, the reconstructed 3D point $\mathbf{M}$ can be found by solving for parameter ζ_ℓ and ζ_r inEquation (23). Let us rewrite it as:

$$\mathbf{m}'_\ell = -\frac{1}{\zeta_\ell}\mathbf{e}_r + \frac{\zeta_r}{\zeta_\ell}\mathbf{m}_r \tag{42}$$

The unknowns are ζ_r and ζ_ℓ. Both encode the position of $\mathbf{M}$ in space, as ζ_r is the depth of $\mathbf{M}$ wrt the right camera and ζ_ℓ is the depth of $\mathbf{M}$ wrt the left camera.

The three points $\mathbf{m}_r, \mathbf{e}_r$ and $\mathbf{m}'_\ell$ are known and are collinear, so we can solve for ζ_ℓ using the following closed form expressions [29]:

$$\frac{1}{\zeta_\ell} = \frac{(\mathbf{m}'_\ell \times \mathbf{m}_r) \cdot (\mathbf{m}_r \times \mathbf{e}_r)}{||\mathbf{m}_r \times \mathbf{e}_r||^2}, \tag{43}$$

The reconstructed point $\mathbf{M}$ can then be calculated by inserting the value ζ into Equation (13).

In reality, camera parameters and image locations are known only approximately. The back-projected rays therefore do not actually intersect in space. It can be shown, however, that the above formula, solve Eq. (42) in a least squares sense [23].

Triangulation is addressed in more details in [2, 13, 11, 39].

4.3 Rectification

Given a pair of stereo images, *epipolar rectification* (or simply *rectification*) determines a transformation of each image plane such that *pairs of conjugate epipolar lines become collinear and parallel to one of the image axes* (usually the horizontal one).

The rectified images can be thought of as acquired by a virtual stereo pair, obtained by rotating the original cameras and possibly modifying the intrinsic parameters.

The important advantage of rectification is that computing stereo correspondences is made simpler, because search is done along the horizontal lines of the rectified images.

We assume here that *the stereo pair is calibrated*, i.e., the cameras' internal parameters, mutual position and orientation are known. This assumption is not strictly necessary [15, 25, 21], but leads to a simple technique.

Specifying virtual cameras.

Given the actual camera matrices P_{or} and $P_{o\ell}$, the idea behind rectification is to define two new *virtual* cameras P_{nr} and $P_{n\ell}$ obtained by rotating the actual ones around their optical centers until focal planes becomes coplanar, thereby containing the baseline (Figure 6). This ensures that epipoles are at infinity, hence epipolar lines are *parallel*.

To have *horizontal* epipolar lines, the baseline must be parallel to the x-axis of both virtual cameras. In addition, to have a proper rectification, conjugate points must have the *same vertical coordinate*.

In summary: positions (i.e, optical centers) of the virtual cameras are the same as the actual cameras, whereas the orientation of both virtual cameras differs from the actual ones by suitable rotations; intrinsic parameters are the same for both cameras.

Therefore, the two resulting virtual cameras will differ only in their optical centers, and they can be thought as a single camera translated along the x-axis of its reference system.

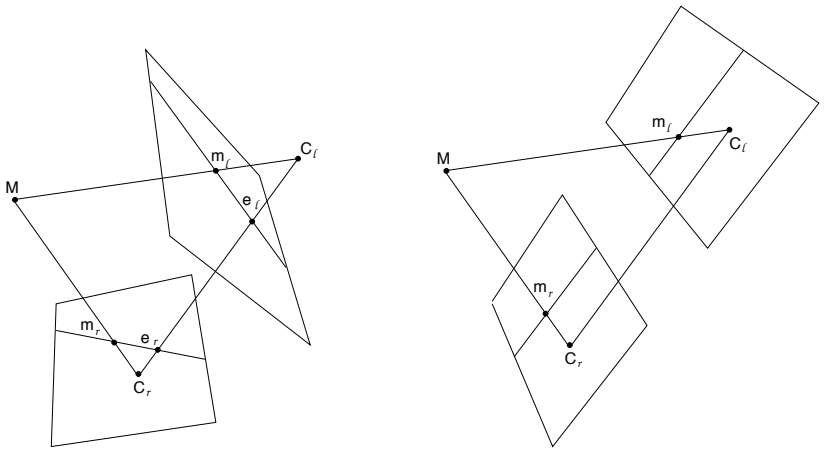


Fig. 6. Epipolar geometry before and after rectification.

Using Eq. (10) and Eq. (12), we can write the virtual cameras matrices as:

$$P_{n\ell} = K[R \mid -R\,\tilde{\mathbf{C}}_\ell], \quad P_{nr} = K[R \mid -R\,\tilde{\mathbf{C}}_r]. \tag{44}$$

In order to define them we need to assign $K, R, \tilde{\mathbf{C}}_\ell, \tilde{\mathbf{C}}_r$

The optical centers $\mathbf{C}_\ell$ and $\mathbf{C}_r$ are the same as the actual cameras. The intrinsic parameters matrix K can be chosen arbitrarily. The matrix R, which gives the orientation of both cameras will be specified by means of its row vectors:

$$R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \tag{45}$$

that are the x, y, and z-axes, respectively, of the virtual camera reference frame, expressed in world coordinates.

According to the previous comments, we take:

(i) The x-axis parallel to the baseline: $\mathbf{r}_1 = (\tilde{\mathbf{C}}_r - \tilde{\mathbf{C}}_\ell)/||\tilde{\mathbf{C}}_r - \tilde{\mathbf{C}}_\ell||$

(ii) The y-axis orthogonal to x (mandatory) and to an arbitrary unit vector $\mathbf{k}$: $\mathbf{r}_2 = \mathbf{k} \times \mathbf{r}_1$

(iii) The z-axis orthogonal to xy (mandatory) : $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$

In point 2, $\mathbf{k}$ fixes the position of the y-axis in the plane orthogonal to x. In order to ensure that the virtual cameras look in the same direction as the actual ones, $\mathbf{k}$ is set equal to the direction of the optical axis of one of the two actual cameras.

We assumed that both virtual cameras have the same intrinsic parameters. Actually, the horizontal component of the image center can be different, and this degree of freedom might be exploited to "center" the rectified images in the viewport by applying a suitable horizontal translation.

The rectifying transformation. In order to rectify, say, the left image, we need to compute the transformation mapping the image plane of $P_{o\ell}$ onto the image plane of $P_{n\ell}$.

According to the equation of the optical ray, for any 3D point $\mathbf{M}$ that projects to $\mathbf{m}_{o\ell}$ in the actual image and to $\mathbf{m}_{n\ell}$ in the rectified image, there exist two parameters ζ_o and ζ_n such that:

$$\begin{cases} \tilde{\mathbf{M}} = \tilde{\mathbf{C}}_\ell + \zeta_o P_{3\times3,o\ell}^{-1} \mathbf{m}_{o\ell} \\ \tilde{\mathbf{M}} = \tilde{\mathbf{C}}_\ell + \zeta_n P_{3\times3,n\ell}^{-1} \mathbf{m}_{n\ell} \end{cases} \tag{46}$$

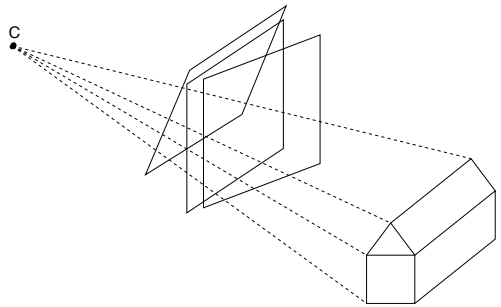
hence

$$\mathbf{m}_{n\ell} = \frac{\zeta_o}{\zeta_n} P_{3\times3,n\ell} P_{3\times3,o\ell}^{-1} \mathbf{m}_{o\ell} \tag{47}$$

The transformation sought is a linear transformation of the projective plane (called *collineaton*) given by the 3×3 matrix $H_\ell = P_{3\times3,n\ell} P_{3\times3,o\ell}^{-1}$.

Note that the scale factor $\frac{\zeta_o}{\zeta_n}$ can be neglected, as the transformation H_ℓ is defined up to a scale factor (being homogeneous). The same result applies to the right image.

It is useful to think of an image as the intersection of the image plane with the cone of rays between points in 3D space and the optical centre. We are moving the image plane while leaving fixed the cone of rays.



Reconstruction of 3D points by triangulation can be performed from the rectified images directly, using P_{nr} and $P_{n\ell}$.

More details on the rectification algorithm can be found in [10], from which this section has been adapted.

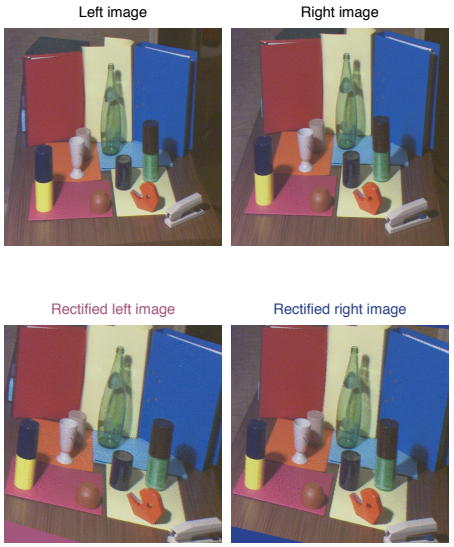


Fig. 7. Original and rectified stereo pair.

4.4 Planes and collineations

When observing a plane, we obtain an interesting specialization of the epipolar geometry of two views.

First, let us establish that the map between a world plane and its perspective image is a collineation of $\mathbb{P}_2$. The easiest way to see it, is to choose the world coordinate system such that the plane of the points have zero z coordinate:

Expanding gives:

$$\zeta \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,4} \\ P_{2,1} & P_{2,2} & P_{2,4} \\ P_{3,1} & P_{3,2} & P_{3,4} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (48)$$

Points are mapped from the world plane to the image plane with a 3×3 non-singular matrix, which represents a collineation of $\mathbb{P}_2$.

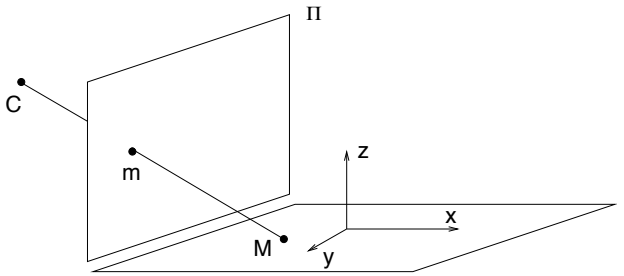


Fig. 8. The map between a world plane Π and a perspective image is a collineation.

Next, we prove that: images of points on a plane are related to corresponding image points in a second view by a *collineation* (or homography) of $\mathbb{P}_2$.

Let us consider two cameras and a plane Π: we have two collineations. Composing the inverse of the left collineation with the right one defines a collineations from the image plane of the left camera to the image plane of the right camera.

The plane Π induces a collineation H_Π between the views, which transfer points from one view to the other:

$$\mathbf{m}_r \simeq H_\Pi \mathbf{m}_\ell \quad \text{if } \mathbf{M} \in \Pi. \quad (49)$$

where H_Π is a 3×3 non-singular matrix.

Even though a collination of $\mathbb{P}_2$ depends upon eight parameters, there is no contraddiction with the fact that a plane depends upon three parameters. Indeed, the collineation induced by a plane must be compatible with the epipolar geometry, i.e.:

$$(H_\Pi \mathbf{m}_\ell)^T F \mathbf{m}_\ell = 0 \tag{50}$$

for all points $\mathbf{m}$. This implies that the matrix $H_\Pi^T F$ is antisymmetric:

$$H_\Pi^T F + F^T H_\Pi = \mathbf{0} \tag{51}$$

and this imposes six homogeneous costraints on H_Π.

A collineation H that satisfies Eq. (51) is said to be *compatible* with F.

A collineation H is compatible with F if and only if

$$F = [\mathbf{e}_r]_\times H \tag{52}$$

From this follows that – provided that Π does not contain $\mathbf{C}_r$ –

$$H_\Pi \mathbf{e}_\ell = \mathbf{e}_r \tag{53}$$

4.4.1 Homography induced by a plane

If the 3D point $\mathbf{M}$ lies on a plane Π with equation $\mathbf{n}^T \tilde{\mathbf{M}} = d$, Eq. (31) can be specialized, obtaining (after elaborating):

$$\frac{\zeta_r}{\zeta_\ell}\mathbf{m}_r = K_r \left(R + \frac{\mathbf{t}\,\mathbf{n}^T}{d} \right) K_\ell^{-1} \mathbf{m}_\ell. \tag{54}$$

Therefore, the collineation induced by Π is given by:

$$H_\Pi = K_r \left(R + \frac{\mathbf{t}\,\mathbf{n}^T}{d} \right) K_\ell^{-1} \tag{55}$$

This is a three-parameter family of collineations, parametrized by $\mathbf{n}/d$.

4.4.2 Infinite homography

The infinite homography H_∞ is the collineation induced by the plane at infinity; it maps vanishing points to vanishing points.

It can be derived by letting $d \to \infty$ in (54), thereby obtaining:

$$H_\infty = K_r R K_\ell^{-1} \tag{56}$$

The infinity homograhy does not depend on the translation between views.

Note that H_∞ can be obtained if $\mathbf{t} = \mathbf{0}$ in Eq. (31), which corresponds to a rotation about the camera centre. Thus H_∞ not only relates points at infinity when the camera describes a general motion, but it also relates image points of any depth if the camera rotates about its centre.

4.4.3 Plane induced parallax

In general, when points are not on the plane, the homography induced by a plane generates a virtual parallax. This gives rise to an alternative representation of the epipolar geometry and scene structure [32].

First let us note that Eq. (31), which we write:

$$\frac{\zeta_r}{\zeta_\ell}\mathbf{m}_r = H_\infty \mathbf{m}_\ell + \frac{1}{\zeta_\ell}\mathbf{e}_r, \tag{57}$$

can be seen as composed by a transfer of point according to the infinity homography $(H_\infty \mathbf{m}_\ell)$ plus a parallax correction term $(\frac{1}{\zeta_\ell}\mathbf{e}_r)$.

We want to generalize this equation to any plane. To this end we substitute

$$H_\infty = H_\Pi - K_r \left(\frac{\mathbf{t}\ \mathbf{n}^T}{d} \right) K_\ell^{-1} \tag{58}$$

into Eq. (57), obtaining

$$\frac{\zeta_r}{\zeta_\ell} \mathbf{m}_r = H_\Pi \mathbf{m}_\ell + \gamma \mathbf{e}_r \tag{59}$$

with $\gamma = \left(\frac{a}{d\ \zeta_\ell} \right)$, where a is the distance of $\mathbf{M}$ to the plane Π.

When $\mathbf{M}$ is on the 3D plane Π, then $\mathbf{m}_r \simeq H_\Pi \mathbf{m}_\ell$. Otherwise there is a residual displacement, called *parallax*, which is proportional to γ and oriented along the epipolar line.

The magnitude parallax depends only on the left view and the plane. It does not depend on the parameters of the right view.

From from Eq. (59) we derive $\mathbf{m}_r^T (\mathbf{e}_r \times H_\Pi \mathbf{m}_\ell) = 0$, hence

$$F = [\mathbf{e}_r]_\times H_\Pi \tag{60}$$

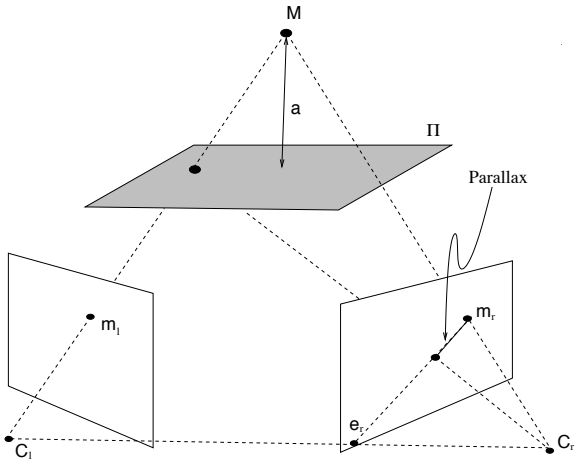


Fig. 9. Plane induced parallax.

Fig. 10. Left and right images. The leftmost image is a superposition of the warped left image and the right image. The reference plane exactly coincide. However, points off the plane (such as the bottle) do not coincide.

4.4.4 Estimating H

A number of point correspondences $\mathbf{m}_r^i \leftrightarrow \mathbf{m}_\ell^i$ is given, and we are required to find an homography matrix H such that

$$\mathbf{m}_r^i \simeq H \mathbf{m}_\ell^i \quad \text{for all } i \tag{61}$$

The equation (we drop the index i for simplicity) can be rewritten in terms of the cross product as

$$\mathbf{m}_r \times H \mathbf{m}_\ell = \mathbf{0} \tag{62}$$

As we did before, we exploit the properties of the Kronecker product and the vec operator to transform this into a null-space problem and then derive a linear solution.

$$\begin{aligned} \mathbf{m}_r \times H\mathbf{m}_\ell = \mathbf{0} &\iff [\mathbf{m}_r]_\times H\mathbf{m}_\ell = \mathbf{0} \iff \mathrm{vec}([\mathbf{m}_r]_\times H\mathbf{m}_\ell) = \mathbf{0} \\ &\iff (\mathbf{m}_\ell^T \otimes [\mathbf{m}_r]_\times)\,\mathrm{vec}\,H = \mathbf{0} \iff ([\mathbf{m}_r]_\times \otimes \mathbf{m}_\ell^T)\,\mathrm{vec}\,H^T = \mathbf{0} \end{aligned} \tag{63}$$

After expanding the coefficient matrix, we obtain

$$\begin{bmatrix} \mathbf{0}^T & -\mathbf{m}_\ell^T & v\mathbf{m}_\ell^T \\ \mathbf{m}_\ell^T & \mathbf{0}^T & -u\mathbf{m}_\ell^T \\ -v\mathbf{m}_\ell^T & u\mathbf{m}_\ell^T & \mathbf{0}^T \end{bmatrix} \mathrm{vec}\,H^T = \mathbf{0} \tag{64}$$

Although there are three equations, only two of them are linearly independent: we can write the third row (e.g.) as a linear combination of the first two.

From a set of n point correspondences, we obtain a $2n \times 9$ coefficient matrix A by stacking up two equations for each correspondence. The projection matrix H is computed by solving the resulting linear system of equations, for $n \geq 4$.

In general A will have rank 8 and the solution is the 1-dimensional right null-space of A.

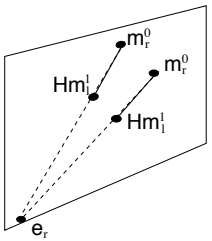
If the data are not exact (noise is generally present) the rank of A will be 9 and a least-squares solution is sought.

The least-squares solution for $\mathrm{vec}\, H^T$ is the singular vector corresponding to the smallest singular value of A.

This is another incarnation of the DLT algorithm.

4.4.5 Estimating the epipole

The epipole can be located [36] given the homography H_Π between two views and two off-plane conjugate pairs $\mathbf{m}_\ell^0 \leftrightarrow \mathbf{m}_r^0$ and $\mathbf{m}_\ell^1 \leftrightarrow \mathbf{m}_r^1$.



Following simple geometric consideration, the epipole is computed as the intersection between the line containing $H_\Pi \mathbf{m}_\ell^0, \mathbf{m}_r^0$ and the line containing $H_\Pi \mathbf{m}_\ell^1, \mathbf{m}_r^1$:

$$\mathbf{e}_r \simeq (H_\Pi \mathbf{m}_\ell^0 \times \mathbf{m}_r^0) \times (H_\Pi \mathbf{m}_\ell^1 \times \mathbf{m}_r^1) \tag{65}$$

In the projective plane, the line determined by two points is given by their cross product, as well as the point determined by two lines.

4.4.6 Estimating the parallax

We are required to compute the magnitude of the parallax γ for a point $\mathbf{m}_\ell$ given its corresponding point $\mathbf{m}_r$, the homography H_Π between the two views and the epipole. To this end we rewrite (59) as:

$$H_\Pi \mathbf{m}_\ell = -\gamma \mathbf{e}_r + \frac{\zeta_r}{\zeta_\ell} \mathbf{m}_r \tag{66}$$

and, given that points $\mathbf{e}_r$, $\mathbf{m}_r$ and $H_\Pi \mathbf{m}_\ell$ are collinear, we solve for γ using:

$$\gamma = \frac{(\mathbf{H}_\Pi \mathbf{m}_\ell \times \mathbf{m}_r)^T (\mathbf{m}_r \times \mathbf{e}_r)}{||\mathbf{m}_r \times \mathbf{e}_r||^2} \tag{67}$$

Please note that the epipole and the homography can be computed from images only up to an unknown scale factor. It follows that the magnitude of the parallax as well is known only up to a scale factor.

4.4.7 Applications

Mosaics. Image mosaicing is the automatic alignment (or registration) of multiple images into larger aggregates [34]. There are two types of mosaics. In both cases, it turns out that images are related by homographies, as we discussed previously.

Planar mosaic: result from the registration of different views of a planar scene.

Panoramic mosaic result from the registration of views taken by a camera rotating around its optical centre (typ. panning). In some cases, in order to cope with large rotations (> 180 deg), the images are converted to cylindrical coordinates.

Fig. 11. Planar mosaic with components location shown as white outlines.

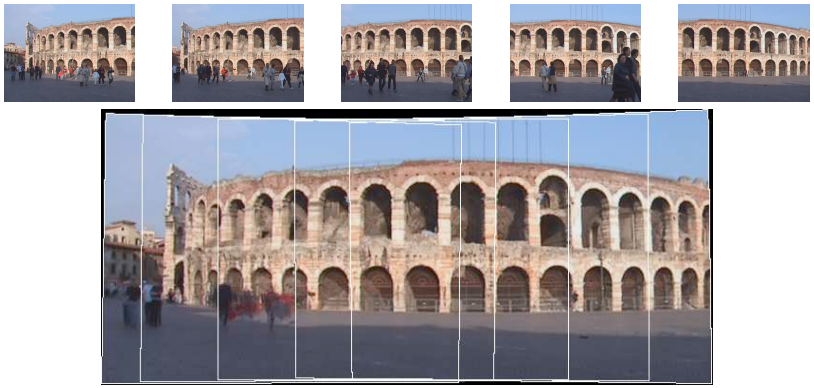


Fig. 12. Selected frames from "Arena" sequence (top) and panoramic mosaic (bottom). Components location shown as white outlines.

Orthogonal rectification. The map between a world plane and its perspective image is an homography. The world-plane to image-plane homography is fully defined by four points of which we know the relative position in the world plane. Once this homography is determined, the image can be back projected (warped) onto the world plane. This is equivalent to synthesize an image as taken from a fronto-parallel view of the plane. This is known as *orthogonal rectification* [24] of a perspective image.

Fig. 13. A perspective image and a ortho-rectified image of the floor plane

4.5 3D Reconstruction

What can be reconstructed depends on what is known about the scene and the stereo system. We can identify three cases.

(i) *If both the intrinsic and extrinsic camera parameters are known*, we can solve the reconstruction problem unambiguously by triangulation.

(ii) *If only the intrinsic parameters are known*, we can estimate the extrinsic parameters and solve the reconstruction problem up to an unknown scale factor. In other words, R can be estimated completely, and $\mathbf{t}$ up to a scale factor.

(iii) *If neither intrinsic nor extrinsic parameters are known*, i.e., the only information available are pixel correspondences, we can still solve the reconstruction problem but only up to an unknown, global projective transformation of the world.

4.5.1 Reconstruction up to a Similarity

If only intrinsics are known (plus point correspondences between images), the epipolar geometry is described by the essential matrix (Section 4.1.1). We will see that, starting from the essential matrix, only a reconstruction up to a similarity transformation (rigid+ uniform scale) can be achieved. Such a reconstruction is referred to as "Euclidean".

Unlike the fundamental matrix, the only property of which is to have rank two, the essential matrix is characterised by the following theorem [20].

Theorem 4.1 *A real 3×3 matrix E can be factorised as product of a nonzero skew-symmetric matrix and a rotation matrix if and only if E has two identical singular values and a zero singular value.*

The theorem has a constructive proof (see [12]) that describes how E can be factorised into rotation and translation using its Singular Value Decomposition (SVD).

The rotation R and translation $\mathbf{t}$ are then used to instantiate a camera pair as in Equation (24), and this camera pair is subsequently used to reconstruct the structure of the scene by triangulation.

The rigid displacement ambiguity arises from the arbitrary choice of the world reference frame, whereas the scale ambiguity derives from the fact that $\mathbf{t}$ can be scaled arbitrarily in Equation (28) and one would get the same essential matrix (E is defined up to a scale factor).

Therefore translation can be recovered from E only up to an unknown scale factor which is inherited by the reconstruction. This is also known as *depth-speed ambiguity*.

4.5.2 Reconstruction up to a Projective Transformation

Suppose that a set of image correspondences $\mathbf{m}_\ell^i \leftrightarrow \mathbf{m}_r^i$ are given. It is assumed that these correspondences come from a set of 3D points $\mathbf{M}_i$, which are unknown. Similarly, the position, orientation and calibration of the cameras are not known. This situation is usually referred to as *weak calibration*, and we will see that the scene may be reconstructed up to a projective ambiguity, which may be reduced if additional information is supplied on the cameras or the scene.

The reconstruction task is to find the camera matrices P_ℓ and P_r, as well as the 3D points $\mathbf{M}_i$ such that

$$\mathbf{m}_\ell^i = P_\ell \mathbf{M}^i \quad \text{and} \quad \mathbf{m}_r^i = P_r \mathbf{M}^i, \quad \forall i \tag{68}$$

In particular, if T is any 4×4 invertible matrix, representing a projective transformation of the 3D space, then replacing points $\mathbf{M}^i$ by $T\mathbf{M}^i$ and matrices P_ℓ and P_r by $P_\ell T^{-1}$ and $P_r T^{-1}$ does not change the image points. This shows that, if nothing is known but the image points, the points $\mathbf{M}^i$ and the cameras can be determined, at best, only up to a projective transformation.

The procedure for reconstruction follows the previous one. Given the weak calibration assumption, the fundamental matrix can be computed (using the algorithm described in Section 4.1.2), and from a (non-unique) factorization of F of the form

$$F = [\mathbf{e}_r]_\times A \tag{69}$$

two camera matrices P_ℓ and P_r:

$$P_\ell = [I|\mathbf{0}] \quad \text{and} \quad P_r = [A|\mathbf{e}_r], \tag{70}$$

can be created in such a way that they yield the fundamental matrix F, as can be easily verified. The position in space of the points $\mathbf{M}^i$ is then obtained by triangulation.

The only difference with the previous case is that F does not admit a unique factorization, whence the projective ambiguity follows.

Indeed, for any A satisfying Equation (69), also $A + \mathbf{e}_r \mathbf{x}^T$ for any vector $\mathbf{x}$, satisfies Equation (69).

One matrix A satisfying Equation (69) can be obtained as $A = -[\mathbf{e}_r]_\times F$ (this is called the epipolar projection matrix [26]).

More in general, any homography induced by a plane can be taken as the A matrix (cfr. Eq. (52)).

5 Multiple View Geometry

In this section we study the relationship that links three or more views of the same 3D scene, known in the three-view case as *trifocal geometry*.

This geometry can be described in terms of fundamental matrices linking pairs of cameras, but in the three-view case a more compact and elegant description is provided by a special algebraic operator, the *trifocal tensor*.

We also discover that four views are all we need, in the sense that additional views do not allow us to compute anything we could not already comput (Section 5.4).

5.1 Trifocal geometry

Denoting the cameras by $1, 2, 3$, there are now three fundamental matrices, $F_{1,2}$, $F_{1,3}$, $F_{2,3}$, and six epipoles, $\mathbf{e}_{i,j}$, as in Figure 14. The three fundamental matrices describe completely the trifocal geometry [8].

The plane containing the three optical centres is called the *trifocal plane*. It intersects each image plane along a line which contains the two epipoles.

Writing Eq. (40) for each camera pair (taking the centre of the third camera as the point $\mathbf{M}$) results in three epipolar constraints:

$$F_{3,1}\mathbf{e}_{3,2} \simeq \mathbf{e}_{1,3} \times \mathbf{e}_{1,2} \quad F_{1,2}\mathbf{e}_{1,3} \simeq \mathbf{e}_{2,1} \times \mathbf{e}_{2,3} \quad F_{2,3}\mathbf{e}_{2,1} \simeq \mathbf{e}_{3,2} \times \mathbf{e}_{3,1} \qquad (71)$$

Three fundamental matrices include 21 free parameters, less the 3 constraints above; the trifocal geometry is therefore determined by 18 parameters.

This description of the trifocal geometry fails when the three cameras are collinear, and the trifocal plane reduces to a line.

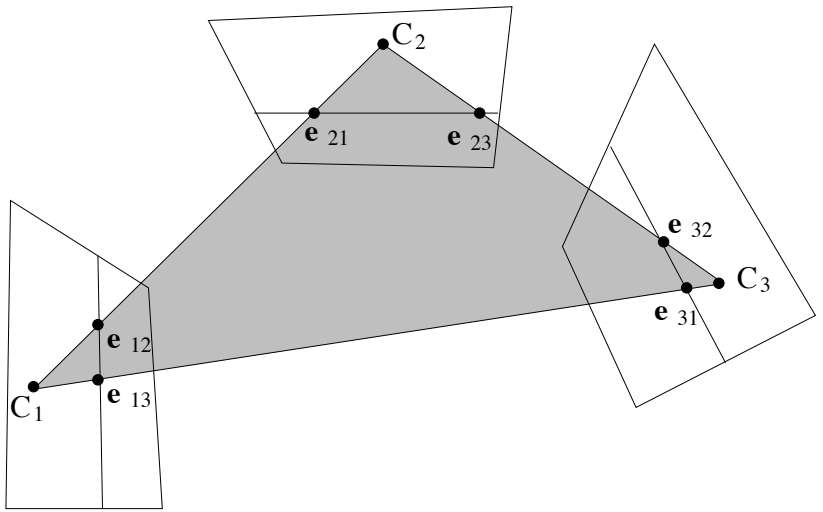


Fig. 14. Trifocal geometry.

If the trifocal geometry is known, given two conjugate points $\mathbf{m}_1$ and $\mathbf{m}_2$ in view 1 and 2 respectively, the position of the conjugate point $\mathbf{m}_3$ in view 3 is completely determined (Figure 15).

This allows for *point transfer* or prediction. Indeed, $\mathbf{m}_3$ belongs simultaneously to the epipolar line of $\mathbf{m}_1$ and to the epipolar line of $\mathbf{m}_2$, hence:

$$\mathbf{m}_3 \simeq F_{1,3}\mathbf{m}_1 \times F_{2,3}\mathbf{m}_2 \tag{72}$$

Epipolar transfer fails for 3D points on the trifocal plane, as the epipolar lines are coincident. Even worse, if the three cameras are collinear, the transfer is not possible for any point.

These deficiencies motivate the introduction of the trifocal tensor. We follow here the formulation iven in [30].

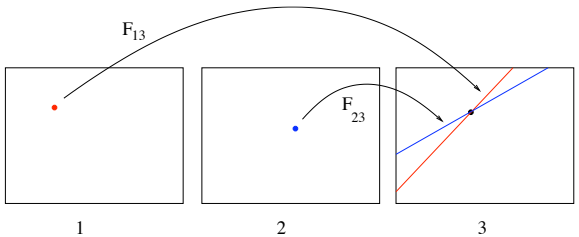


Fig. 15. Point transfer using epipolar constraints between three views.

5.2 The trifocal tensor

Recalling Equation (70), consider the following three cameras:

$$P_1 = [I|0], \quad P_2 = [A|\mathbf{e}_{2,1}], \quad \text{and} \quad P_3 = [B|\mathbf{e}_{3,1}]. \tag{73}$$

Consider a point $\mathbf{M}$ in space projecting to $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$ in the three cameras. Let us write the epipolar line of $\mathbf{m}_1$ in the other two views (using Equation (23)):

$$\zeta_2\mathbf{m}_2 = \mathbf{e}_{2,1} + \zeta_1 A\mathbf{m}_1 \tag{74}$$
$$\zeta_3\mathbf{m}_3 = \mathbf{e}_{3,1} + \zeta_1 B\mathbf{m}_1. \tag{75}$$

Consider a line through $\mathbf{m}_2$, represented by $\mathbf{s}$; we have $\mathbf{s}^T\mathbf{m}_2 = 0$, that substituted in (74) gives:

$$0 = \mathbf{s}^T\mathbf{e}_{2,1} + \zeta_1\mathbf{s}^T A\mathbf{m}_1 \tag{76}$$

Since a point is determined by two lines, we can write a similar independent constraint for a second line $\mathbf{l}$ through $\mathbf{m}_2$:

$$0 = \mathbf{l}^T\mathbf{e}_{2,1} + \zeta_1\mathbf{l}^T A\mathbf{m}_1 \tag{77}$$

To write the last two equations in a more compact way we introduce a 2×3 matrix

$$S = \begin{bmatrix}\mathbf{s}^T \\ \mathbf{l}^T\end{bmatrix}, \tag{78}$$

and switch to tensor notation, where S becomes s^μ_j, and Equations (76) and (77) become:

$$0 = s^\mu_j e^j_{2,1} + \zeta_1 p^i_1 s^\mu_j a^j_i \tag{79}$$

By the same token, we can represent two lines through $\mathbf{m}_3$ in tensor notation by means of 2×3 matrix r^ρ_k, obtaining:

$$0 = r^\rho_k e^k_{3,1} + \zeta_1 p^i_1 r^\rho_k b^k_i \tag{80}$$

After eliminating ζ_1 from Equation (79) and (80) we obtain

$$(s^\mu_j e^j_{2,1})(p^i_1 r^\rho_k b^k_i) = (r^\rho_k e^k_{3,1})(p^i_1 s^\mu_j a^j_i) \tag{81}$$

and after some re-writing:

$$p^i_1 s^\mu_j r^\rho_k \mathcal{T}_i^{jk} = 0, \tag{82}$$

where

$$\mathcal{T}_i^{jk} \triangleq e^j_{2,1} b^k_i - e^k_{3,1} a^j_i \tag{83}$$

is a $3 \times 3 \times 3$ homogeneous tensor, called the *trifocal tensor*. The tensorial equation (82) represents *four trilinear equations*, since $\mu = 1, 2$ and $\rho = 1, 2$ are free indices.

This relationship involves the trifocal tensor and the three conjugate points, as s_j^μ and r_k^ρ define $\mathbf{m}_2$ and $\mathbf{m}_3$ respectively. To make the coordinates of $\mathbf{m}_2$ and $\mathbf{m}_3$ appear explicitly, let us choose the two lines parallel to the coordinate axes, thereby obtaining:

$$s_j^\mu = \begin{bmatrix} -1 & 0 & u_2 \\ 0 & -1 & v_2 \end{bmatrix} \quad \text{and} \quad r_k^\rho = \begin{bmatrix} -1 & 0 & u_3 \\ 0 & -1 & v_3 \end{bmatrix}. \tag{84}$$

After substituting the above expressions for s_j^μ and r_k^ρ, Equation (82) becomes:

$$\begin{aligned} u_3 \mathcal{T}_i^{13} p_1^i - u_3 u_2 \mathcal{T}_i^{33} p_1^i + u_2 \mathcal{T}_i^{31} p_1^i - \mathcal{T}_i^{11} p_1^i = 0 \\ v_3 \mathcal{T}_i^{13} p_1^i - v_3 u_2 \mathcal{T}_i^{33} p_1^i + u_2 \mathcal{T}_i^{32} p_1^i - \mathcal{T}_i^{12} p_1^i = 0 \\ u_3 \mathcal{T}_i^{23} p_1^i - u_3 v_2 \mathcal{T}_i^{33} p_1^i + v_2 \mathcal{T}_i^{31} p_1^i - \mathcal{T}_i^{21} p_1^i = 0 \\ v_3 \mathcal{T}_i^{23} p_1^i - v_3 v_2 \mathcal{T}_i^{33} p_1^i + v_2 \mathcal{T}_i^{32} p_1^i - \mathcal{T}_i^{22} p_1^i = 0 \end{aligned} \tag{85}$$

Every triplet ($\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$) of corresponding points gives four linear independent equations, hence seven triplets determine the trifocal tensor.

Notice that the trifocal tensor represents the trifocal geometry without singularities: it can be safely used for point transfer in any situation. A transferred point, $\mathbf{m}_3$, can be computed from Equation (85) or, in closed form, as:

$$p_3^k = p_1^i s_j \mathcal{T}_i^{jk} \tag{86}$$

where s_j represents a line through $\mathbf{m}_2$. In a similar way, the trifocal tensor can be used to transfer lines:

$$q_i = s_j r_k \mathcal{T}_i^{jk} \tag{87}$$

where s_j and r_k represent two matching lines in the first two views, and q_i is the transferred line in the third view.

5.3 Reconstruction

As in the case of two views, what can be reconstructed depends on what is known about the scene and the cameras. In general, if the internal parameters of the cameras are known, we can obtain a *Euclidean reconstruction*, that differs from the true reconstruction by a similarity transformation. This is composed by a rigid displacement (due to the arbitrary choice of the world reference frame) plus a a uniform change of scale (due to the well-known depth-speed ambiguity). In the weakly calibrated case, i.e., when point correspondences are the only information available, a projective reconstruction can be obtained.

The reconstruction in the multiple-views case, however, poses some additional problems.

5.3.1 Euclidean Reconstruction

Let us consider for simplicity the case of three views, which generalizes straightforward to N views.

If one applies the method of Section 4.5.1 to view pairs 1-2, 1-3 and 2-3 one obtains three displacements $(R_{12}, \hat{\mathbf{t}}_{12}), (R_{13}, \hat{\mathbf{t}}_{13})$ and $(R_{23}, \hat{\mathbf{t}}_{23})$ known up a scale factor, as the norm of translation cannot be recovered, owing to the depth-sped ambiguity (the symbol $\hat{}$ indicates a unit-norm vector).

The "true" displacements must satisfy the following compositional rule

$$\mathbf{t}_{13} = R_{23}\mathbf{t}_{12} + \mathbf{t}_{23} \tag{88}$$

which can be rewritten as

$$\hat{\mathbf{t}}_{13} = \mu_1 R_{23}\hat{\mathbf{t}}_{12} + \mu_2 \hat{\mathbf{t}}_{23} \tag{89}$$

where $\mu_1 = ||\mathbf{t}_{12}||/||\mathbf{t}_{13}||$ and $\mu_2 = ||\mathbf{t}_{23}||/||\mathbf{t}_{13}||$ are unknown.

However Eq. (88) constraints $\hat{\mathbf{t}}_{13}, R_{23}\hat{\mathbf{t}}_{12}$ and $\hat{\mathbf{t}}_{23}$ to be coplanar, hence the ratios μ_1, μ_2 can be recovered :

$$\frac{||\mathbf{t}_{12}||}{||\mathbf{t}_{13}||} = \mu_1 = \frac{(\hat{\mathbf{t}}_{13} \times \hat{\mathbf{t}}_{23}) \cdot (R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23})}{||R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23}||^2} \tag{90}$$

And similarly for μ_2.

In this way three consistent camera matrices can be instantiated.

Note that only ratios of translation norm can be computed, hence the global scale factor remains undetermined.

5.3.2 Projective Reconstruction

As in the case of two cameras, given only point correspondences, it is possible to reconstruct scene structure and camera matrices up to a global unknown projective transform.

The reconstruction from N views, however, cannot be obtained by simply applying the method of Section 4.5.2 to pairs of views. One would obtain, in general, a set of projective reconstructions linked to each other by an unknown projective transformation (i.e., each camera pair defines its own projective frame).

An elegant method for multi-image reconstruction was described in [33], based on the idea of factorization method [35] idea.

Consider m cameras $P_1 \dots P_m$ looking at n 3D points $\mathbf{M}^1 \dots \mathbf{M}^n$. The usual projection equation

$$\zeta_i^j \mathbf{m}_i^j = P_i \mathbf{M}^j \quad i = 1 \dots m, \quad j = 1 \dots n. \tag{91}$$

can be written in matrix form:

$$\underbrace{\begin{bmatrix} \zeta_1^1\mathbf{m}_1^1, & \zeta_1^2\mathbf{m}_1^2, & \dots & \zeta_1^n\mathbf{m}_1^n \\ \zeta_2^1\mathbf{m}_2^1, & \zeta_2^2\mathbf{m}_2^2, & \dots & \zeta_2^n\mathbf{m}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_m^1\mathbf{m}_m^1, & \zeta_m^2\mathbf{m}_m^2, & \dots & \zeta_m^n\mathbf{m}_m^n \end{bmatrix}}_{\text{measurements } W} = \underbrace{\begin{bmatrix} P_1, \\ P_2, \\ \vdots \\ P_m \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} \mathbf{M}^1, \mathbf{M}^2, \dots \mathbf{M}^n \end{bmatrix}}_{\text{structure } S}. \tag{92}$$

In this formula the $\mathbf{m}_i^j$ are known, but all the other quantities are unknown, including the projective depths ζ_i^j. Equation (92) tells us that W can be factored into the product of a $3m \times 4$ matrix P and a $4 \times n$ matrix S. This also means that W has rank four.

If we assume for a moment that the projective depths ζ_i^j are known, then matrix M is known too and we can compute its singular value decomposition:

$$W = UDV. \tag{93}$$

In the noise-free case, $D = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, 0, \dots 0)$, thus, only the first 4 columns (rows) of U (V) contribute to this matrix product. Let U' (V') the matrix of the first 4 columns (rows) of U (V). Then:

$$W = U' \,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)\, V'. \tag{94}$$

The sought reconstruction is obtained by setting:

$$P = U' \,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4) \quad \text{and} \quad S = V'$$

For any non singular projective transformation T, TP and $T^{-1}S$ is an equally valid factorization of the data into projective motion and structure.

As expected, the reconstruction is up to an unknown projective transformation.

A consequence of this is that the choice of attaching the diagonal matrix to U' is arbitrary. It could be attached to V' or even factorized in two matrices.

As the $\zeta_{i,j}$ are unknown, we are left with the problem of estimating them. The original algorithm [33] uses the epipolar constraint (Eq.40) to fix the ratio of the projective depths of one point in successive images.

In [17] the projective depths are estimated in an iterative fashion. First let us note that for a fixed camera i the projection equation writes:

$$\left[\mathbf{m}_i^1,\ \mathbf{m}_i^2,\ \ldots\ \mathbf{m}_i^n\right] Z_i = P_i S \tag{95}$$

where $Z_i = \mathrm{diag}(\zeta_{i,1}, \zeta_{i,2}, \ldots \zeta_{i,n})$. The following iterative procedure is used:

1. Set $\zeta_{i,j} = 1$;
2. Factorize W and obtain an estimate of P and S;
3. If σ_5 is sufficiently small then stop;
4. Use W, P and S to estimate Z_i from Equation (95);
5. Goto 2.

It can be proven that the quantity that is being minimized is σ_5.

This technique is fast, requires no initialization, and gives good results in practice, although there is no guarantee that the iterative process will converge.

A provably convergent iterative method have been presented in [27].

5.4 Multifocal constraints

We outline here an alternative and elegant way to derive all the meaningfull multilinear constraints, based on determinants, described in [16]. Consider one image point viewed by m cameras:

$$\zeta_i \mathbf{m}_i = P_i \mathbf{M} \quad i = 1 \ldots m \tag{96}$$

By stacking all these equations we obtain:

$$\begin{bmatrix} P_1 & \mathbf{m}_1 & 0 & \ldots & 0 \\ P_2 & 0 & \mathbf{m}_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_m & 0 & 0 & \ldots & \mathbf{m}_m \end{bmatrix} \begin{bmatrix} \mathbf{M} \\ -\zeta_1 \\ -\zeta_2 \\ \vdots \\ -\zeta_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{97}$$

This implies that the $3m \times (m+4)$ matrix (let us call it L) is rank-deficient, i.e., $\mathrm{rank}\ L < m+4$. In other words, all the $(m+4) \times (m+4)$ minors of L are equal to 0.

It has been proven that there are three different types of such minors that translates into meaningful multi-view constraints, depending on the number of rows taken from each view. Since one row has to be taken from each view and the remaining four can be distributed freely, one can choose:

1. Two rows from one view and two rows from another view. This gives a bilinear two-view constraint, expressed by the bifocal tensor i.e., the fundamental matrix.
2. Two rows from one view, one row from another view and one row from a third view. This gives a trilinear three-view constraint, expressed by the trifocal tensor.
3. One row from each of four different views. This gives a quadrilinear four-view constraint, expressed by the quadrifocal tensor.

All the other type of minors can be factorised as product of the two-, three-, or four-views constraints and point coordinates in the other images. This indicates that no interesting constraints can be written for more than four views.