Elements of Geometric Computer Vision

Andrea Fusiello*

www.diegm.uniud.it/fusiello/

October 16, 2012

©E. Trucco

Contents

1 Introduction

This notes introduces the basic geometric concepts of 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 3-D scene.

Geometric vision is an important and well-studied part of computer vision. A wealth of useful results has been achieved in the last 20 years and has been reported in comprehensive monographies [3, 10, 5] to which the reader is referred for a deepr study.

Fig. 1. Example of reconstruction from images. Original images (top row) and reconstructed model (bottom row).

2 Projective Geometry

The physical space is the Euclidean 3-D space $\mathbb{E}^3$, a real 3-dimensional affine space endowed with the inner product.

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

The projective (or homogeneous) 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.

The affine points are those of $\mathbb{P}^3$ which do not belong to Π_∞. Their projective coordinates are of the form $(x, y, z, 1)$, where (x, y, z) are the usual Cartesian coordinates.

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

The linear transformations of a projective space into itself are called collineations or homographies. Any collineation of $\mathbb{P}^3$ is represented by a generic 4×4 invertible matrix.

Affine transformations are the subgroup of collineations of $\mathbb{P}^3$ that preserves the plane at infinity (i.e., parallelism).

Similarity transformations are the subgroup of affine transformations that leave invariant a very special curve, the *absolute conic*, which is in the plane at infinity and whose equation is:

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

Similarity transformations preserves the angles.

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

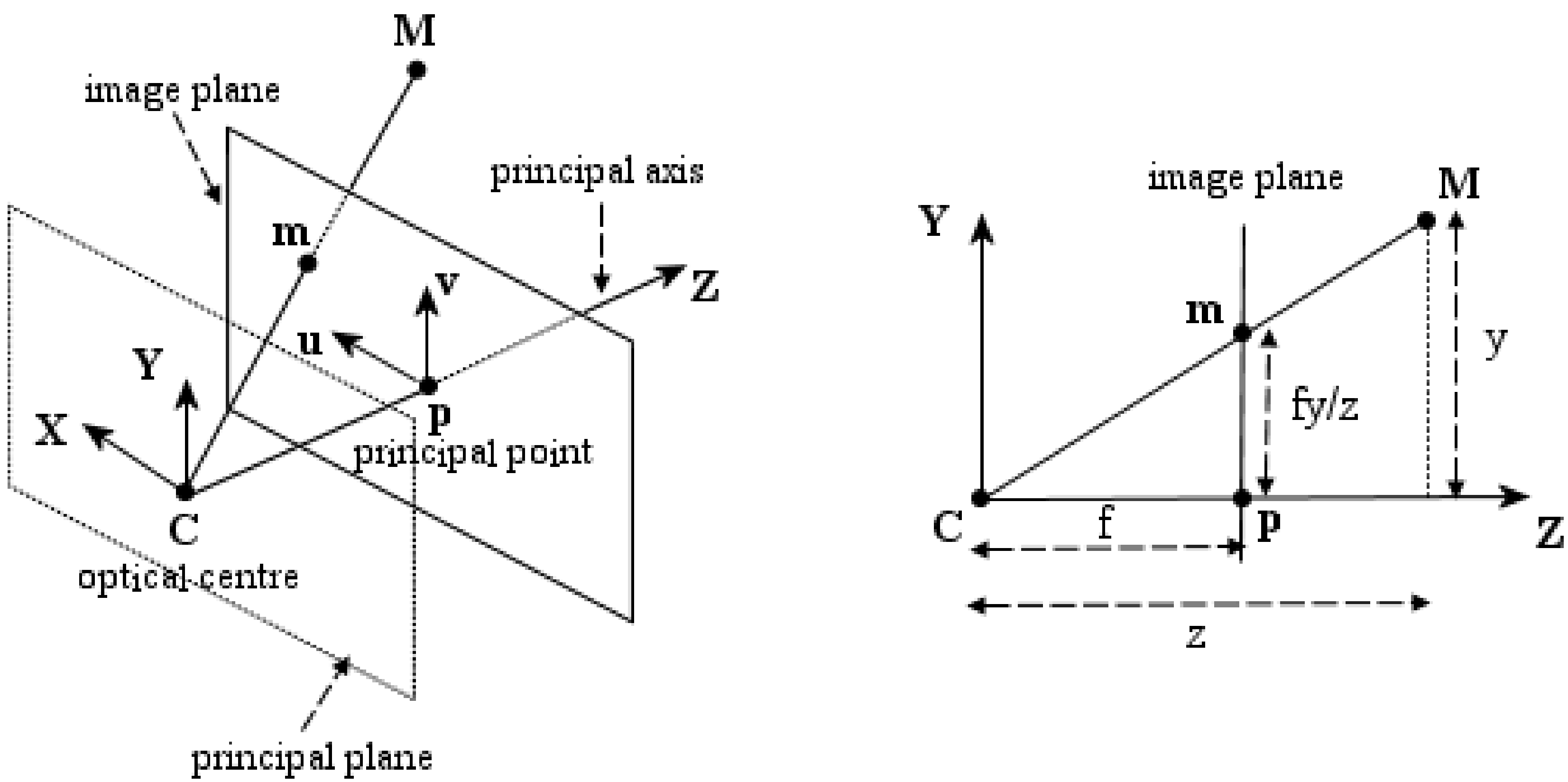


Fig. 2. Pin-hole camera geometry. The left figure illustrates the projection of the point **M** on the image plane by drawing the line through the camera centre **C** and the point to be projected. The right figure illustrates the same situation in the YZ plane, showing the similar triangles used to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

A 3-D 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 Cartesian coordinate system wherein the z-axis is the principal axis.

By similar triangles it is readily seen that the 3-D point $(x, y, z)^T$ is mapped to the point $(fx/z, fy/z)^T$ 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 3-D point and $\mathbf{m} = (fx/z, fy/z, 1)^T$ are the homogeneous coordinates of the image point.

The projection matrix P in Eq. (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 *extrinsic* parameters.

The rows of R are unit vectors that, together with the optical centre, define the *camera reference frame*, expressed in world coordinates.

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 in the image plane from the so-called *normalized camera coordinates* to *pixel coordinates*.

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

- focal distance f (in mm),
- principal point (or 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), and $\mathbf{m} = (u, v, 1)^T$.

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 recovered from $K(3,3) = 1$).

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 3-D vector containing the Cartesian (non-homogeneous) coordinates of the optical centre.

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

$$\tilde{\mathbf{C}} = -P_{1:3}^{-1}P_4 \tag{12}$$

where the matrix P is represented by the block form: $P = [P_{1:3}|P_4]$ (the subscript denotes a range of columns).

Depth of a point

We observe that:

$$\zeta \mathbf{m} = P\mathbf{M} = P\mathbf{M} - P\mathbf{C} = P(\mathbf{M} - \mathbf{C}) = P_{1:3}(\tilde{\mathbf{M}} - \tilde{\mathbf{C}}). \tag{13}$$

In particular, plugging Eq. (10), the third component of this equation is

$$\zeta = \lambda \mathbf{r}_3^T(\tilde{\mathbf{M}} - \tilde{\mathbf{C}})$$

where $\mathbf{r}_3^T$ is the third row of the rotation matrix R, which correspond to the versor of the principal axis.

If $\lambda = 1$, ζ is the projection of the vector $(\tilde{\mathbf{M}} - \tilde{\mathbf{C}})$ onto the principal axis, i.e., the *depth* of $\mathbf{M}$.

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 $\mathbf{m}$ 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 $\mathbf{m}$:

$$\mathbf{M} = \begin{pmatrix} -P_{1:3}^{-1} P_4 \\ 1 \end{pmatrix} + \zeta \begin{pmatrix} P_{1:3}^{-1} \mathbf{m} \\ 0 \end{pmatrix}, \quad \zeta \in \mathbb{R}. \tag{14}$$

If $\lambda = 1$ the parameter ζ in Eq. (14) represent the the depth of the point $\mathbf{M}$. 01

Knowing the intrinsic parameters is equivalent to being able to trace the optical ray of any image point (with $P = [K|\mathbf{0}]$).

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

$$\mathbf{m}_i \simeq P\mathbf{M}_i \quad \text{for all } i. \tag{15}$$

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

$$\mathbf{m}_i \times P\mathbf{M}_i = \mathbf{0}. \tag{16}$$

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 [18], we derive:

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

These are three equations in 12 unknown.

Although there are three equations, only two of them are linearly independent: Indeed, the rank of $(\mathbf{M}_i^T \otimes [\mathbf{m}_i]_\times)$ is two because it is the Kronecker product of a rank-1 matrix by a a rank-2 matrix.

From a set of n point correspondences, we obtain a $2n \times 12$ coefficient matrix A by stacking up two equations for each correspondence.

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.

The projection matrix P is computed by solving the resulting linear system of equations, for $n \geq 6$.

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 $\mathrm{vec}(P)$ is the singular vector corresponding to the smallest singular value of A.

This is called the Direct Linear Transform (DLT) algorithm [10].

4 Two-View Geometry

The two-view geometry is the intrinsic geometry of two different perspective views of the same 3-D 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, provided that that the scene do not change between successive snapshots.

Most 3-D 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 3-D 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 3-D scene point $\mathbf{M}$.

The knowledge of image correspondences enables scene reconstruction from images.

The concept of correspondence is a cornerstone of multiple-view vision. In this notes we assume *known correspondences*, and explore their use in geometric algorithms. Techniques for computing dense correspondences are surveyed in [23, 2].

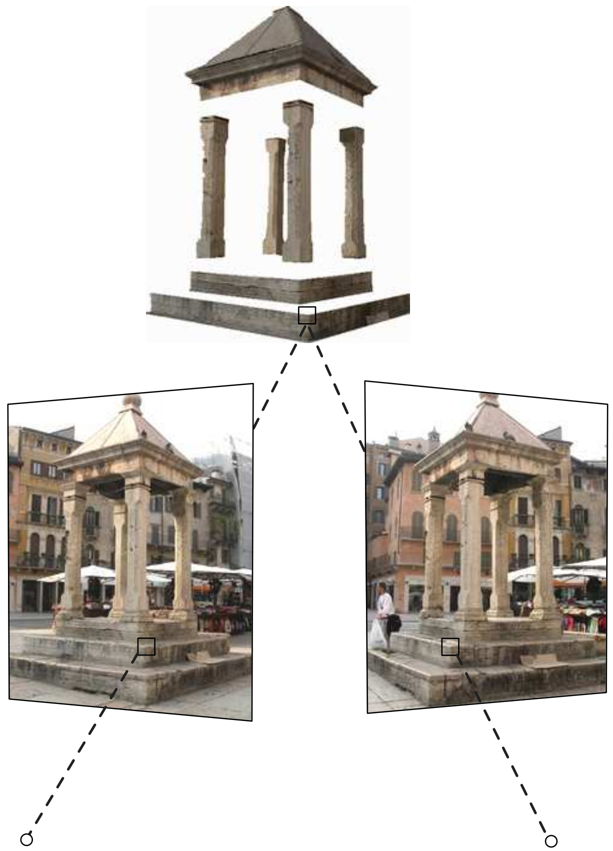


Fig. 3. Two perspective views of the same 3-D scene with conjugate points highlighted

We will refer to the camera projection matrix of the left view as P_ℓ and of the right view as P_r. The 3-D point $\mathbf{M}$ is then imaged as (17) in the left view, and (18) in the right view:

$$\zeta_\ell \mathbf{m}_\ell = P_\ell \mathbf{M} \tag{17}$$

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M}. \tag{18}$$

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 3-D 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 2-D region, a significant reduction in complexity.

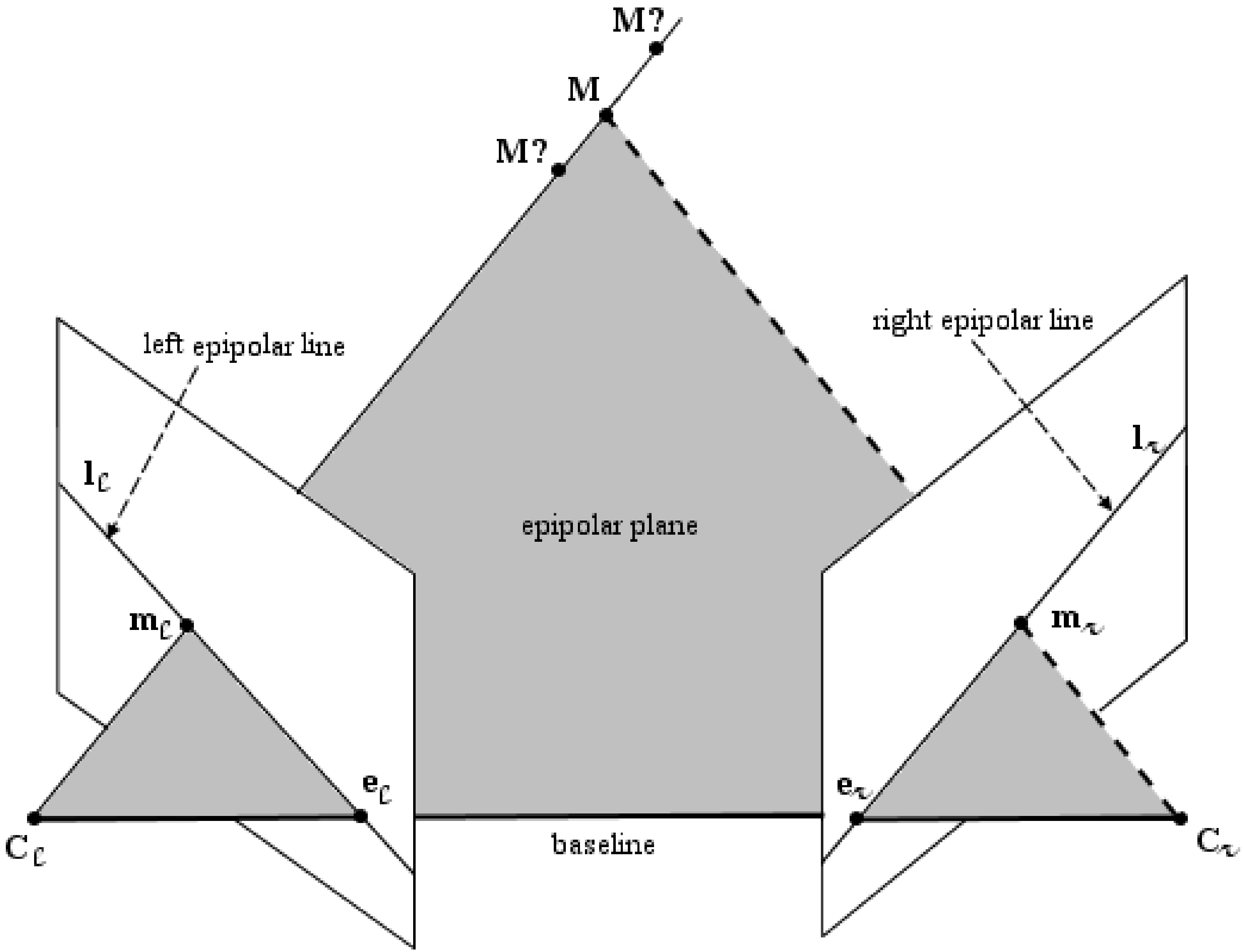


Fig. 4. The epipolar geometry and epipolar constraint.

Any 3-D 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 (Eq. (8)) of the optical ray through $\mathbf{m}_\ell$ (Eq. (14)) onto the right image plane:

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M} = \underbrace{P_r \begin{pmatrix} -P_{\ell_{1:3}}^{-1} P_{\ell_4} \\ 1 \end{pmatrix}}_{\mathbf{e}_r} + \zeta_\ell P_r \begin{pmatrix} P_{\ell_{1:3}}^{-1} \mathbf{m}_\ell \\ 0 \end{pmatrix} \tag{19}$$

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

$$\zeta_r \mathbf{m}_r = \mathbf{e}_r + \zeta_\ell \underbrace{P_{r_{1:3}} P_{\ell_{1:3}}^{-1} \mathbf{m}_\ell}_{\mathbf{m}'_\ell} \tag{20}$$

This is the equation of a line through the right epipole $\mathbf{e}_r$ and the image point $\mathbf{m}'_\ell$ 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.

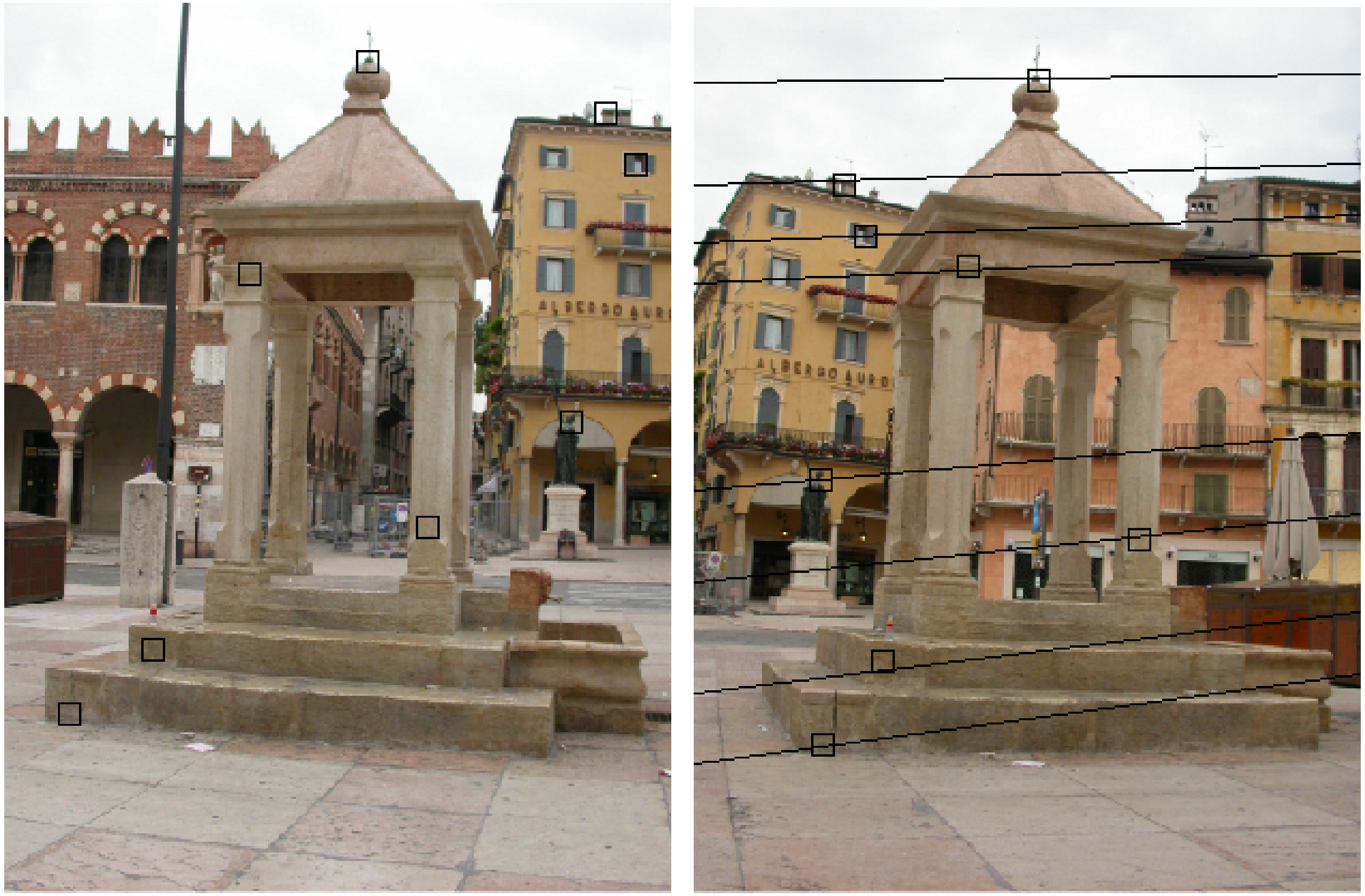


Fig. 5. Left and right images with epipolar lines.

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. 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 3-D scene point $\mathbf{M}$.

Analytically, the reconstructed 3-D point $\mathbf{M}$ can be found by solving for parameter ζ_ℓ or ζ_r in Eq. (20). Let us rewrite it as:

$$\mathbf{e}_r = \zeta_r \mathbf{m}_r - \zeta_\ell \mathbf{m}'_\ell \tag{21}$$

The depth ζ_r and ζ_ℓ are unknown. 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.

However, triangulation can also be cast as a null-space problem.

Let us consider $\mathbf{m} = [u, v, 1]^T$, the projection of the 3D point M according to the perspective projection matrix P. From (8) one obtains:

$$\begin{cases} (\mathbf{p}_1 - u\mathbf{p}_3)^T\mathbf{M} = 0 \\ (\mathbf{p}_2 - v\mathbf{p}_3)^T\mathbf{M} = 0 \end{cases} \tag{22}$$

and then, in matrix form:

$$\begin{bmatrix} (\mathbf{p}_1 - u\mathbf{p}_3)^T \\ (\mathbf{p}_2 - v\mathbf{p}_3)^T \end{bmatrix} \mathbf{M} = \mathbf{0}_{2\times 1} \tag{23}$$

Hence, one point gives two homogeneous equations.

Let us consider now $\mathbf{m}' = [u', v', 1]^T$, the corresponding point of $\mathbf{m}$ in the second image, and let P' be the second perspective projection matrix.

Being both projection of the same 3D point $\mathbf{M}$, the equations provided by $\mathbf{m}$ and $\mathbf{m}'$ can be stacked:

$$\begin{bmatrix} (\mathbf{p}_1 - u\mathbf{p}_3)^T \\ (\mathbf{p}_2 - v\mathbf{p}_3)^T \\ (\mathbf{p}'_1 - u'\mathbf{p}'_3)^T \\ (\mathbf{p}'_2 - v'\mathbf{p}'_3)^T \end{bmatrix} \mathbf{M} = \mathbf{0}_{4\times 1} \tag{24}$$

The solution is the null-space of the 4×4 coefficient matrix, which must then have rank three, otherwise only the trivial solution $\mathbf{M} = \mathbf{0}$ would be possible. In the presence of noise this rank condition cannot be fulfilled exactly, so a lest squares solution is sought, typically via SVD, as in calibration with DLT. In [12] this method is called "*linear-eigen*".

This method generalizes to the case of $N > 2$ cameras: each one gives two equations and one ends up with $2N$ equations in four unknowns.

Triangulation is addressed in more details in [1, 12, 10].

Levels of description...

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.

(i) If both *intrinsic* and *extrinsic* camera parameters are known, we can describe the epipolar geometry in terms of the projection matrices (Equation (20)).

(ii) If only the *intrinsic* parameters are known, we work in normalized camera coordinates and the epipolar geometry is described by the *essential matrix*.

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

...and ambiguity in reconstruction.

Likewise, what can be reconstructed (by triangulation) 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.

(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 (+ a rigid transformation that correspond to the arbitrariness in fixing the world reference frame). 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. This ambiguity w may be reduced if additional information is supplied on the cameras or the scene (see Sec .5).

4.3 The calibrated case

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 3-D points $\mathbf{M}_i$, which are unknown.

The intrinsic parameters are known, i.e. the cameras are *calibrated*, but the position and attitude of the cameras are unknown.

The situation – discussed previously – when the intrinsic *and* extrinsic parameters are known will be referred to as *full calibrated* for the sake of clarity.

We will see that the epipolar geometry is described by the essential matrix and 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".

4.3.1 The Essential Matrix E

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

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

$$P_\ell = [I|0] \quad \text{and} \quad P_r = [R|\mathbf{t}]. \tag{25}$$

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 (20), we get

$$\zeta_r \mathbf{m}_r = \mathbf{t} + \zeta_\ell R \mathbf{m}_\ell; \tag{26}$$

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: 14

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

as the homogeneous line through two points is expressed as their cross product, and 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 (27) can therefore be equivalently written as

$$\mathbf{m}_r^T [\mathbf{t}]_\times R \mathbf{m}_\ell = 0, \tag{28}$$

where $[\mathbf{t}]_\times$ is the skew-symmetric matrix of the vector $\mathbf{t}$. Let us define the *essential matrix* E:

$$E \triangleq [\mathbf{t}]_\times R. \tag{29}$$

In summary, the relationship between the corresponding image points $\mathbf{m}_\ell$ and $\mathbf{m}_r$ in normalized camera coordinates is the bilinear form:

$$\mathbf{m}_r^T E \mathbf{m}_\ell = 0. \tag{30}$$

E encodes only information on the rigid displacement between cameras. It has five degrees of freedom: a 3-D rotation and a 3-D translation direction.

E is singular, since $\det[\mathbf{t}]_\times = 0$, and it is a homogeneous quantity.

4.3.2 Reconstruction up to a Similarity

If a sufficient number of point correspondences $\mathbf{m}_\ell^i \leftrightarrow \mathbf{m}_r^i$ is given, we can use Equation (30) to compute the unknown matrix E (see Sec. 4.4.2).

The reconstruction is achieved starting from the essential matrix, which contains – entangled – the unknown extrinsic parameters.

Unlike the fundamental matrix, the only property of which is to have rank two, the essential matrix is characterised by the following theorem [15].

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.*

Proof. Let $E = SR$ where R is a rotation matrix and S is skew-symmetric. Let $S = [\mathbf{t}]_\times$ where $||\mathbf{t}|| = 1$. Then

$$EE^T = SRR^TS^T = SS^T = I - \mathbf{t}\mathbf{t}^T$$

Let U the orthogonal matrix such that $U\mathbf{t} = [0, 0, 1]^T$. Then

$$UEE^TU^T = U(I - \mathbf{t}\mathbf{t}^T)U^T = I - U\ \mathbf{t}\ \mathbf{t}^TU^T = I - [0, 0, 1]^T\ [0, 0, 1] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The elements of the diagonal matrix are the eigenvalues of EE^T i.e., the singular values of E. This demonstrate one implication.

Let us now give a constructive proof of the converse. Let $E = UDV^T$ be the SVD of E, with $D = \mathrm{diag}(1, 1, 0)$ (with no loss of generality, since E is defined up to a scale factor) and U and V orthogonal. The key observation is that

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \stackrel{\Delta}{=} S'R'$$

where S' is skew symmetric and R' a rotation. Hence

$$E=UDV^T=US'R'V^T=(US'U^T)(UR'V^T) = \underbrace{\det(UV^T)(US'U^T)}_{S}\underbrace{\det(UV^T)(UR'V^T)}_{R}.$$

Q.E.D.

This factorization is not unique. Because of homogeneity of E, we can change its sign, either by changing the sign of S' or by taking the transpose of R' (because $S'R'^T = -D$). In total, we have four possible factorizations given by:

$$S = U(\pm S')U^T \tag{31}$$

$$R = \det(UV^T)UR'V^T \text{ or } R = \det(UV^T)UR'^TV^T, \tag{32}$$

The choice between the four displacements is determined by the requirement that the 3-D points must lie in front of both cameras, i.e., their depth must be positive.

The rotation R and translation $\mathbf{t}$ are then used to instantiate a camera pair as in Equation (25), 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 (29) 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. 15

This is also known as *depth-speed ambiguity* (in a context where points are moving and camera is stationary): a large motion of a distant point and a small motion of a nearby point produces the same motion in the image.

4.4 The weakly calibrated case

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 3-D points $\mathbf{M}_i$, which are unknown.

Similarly, the position, attitude and calibration of the cameras are not known.

This situation is usually referred to as *weak calibration*, and we will see that the epipolar geometry is described by the *fundamental matrix* and the scene may be reconstructed up to a projective ambiguity.

4.4.1 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 more general version of Equation (25):

$$P_\ell = K_\ell[I|0] \quad \text{and} \quad P_r = K_r[R|\mathbf{t}]. \tag{33}$$

Inserting these two projection matrices into Equation (20), we get

$$\zeta_r \mathbf{m}_r = \mathbf{e}_r + \zeta_\ell K_r R K_\ell^{-1} \mathbf{m}_\ell \quad \text{with} \quad \mathbf{e}_r = K_r \mathbf{t}, \tag{34}$$

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:

$$\mathbf{m}_r^T [\mathbf{e}_r]_\times K_r R K_\ell^{-1} \mathbf{m}_\ell = 0. \tag{35}$$

The matrix

$$F = [\mathbf{e}_r]_\times K_r R K_\ell^{-1} \tag{36}$$

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

Therefore, the bilinear form that links corresponding points writes:

$$\mathbf{m}_r^T F \mathbf{m}_\ell = 0. \tag{37}$$

F is the algebraic representation of the epipolar geometry in the least information case. 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

$$\mathbf{l}_r = F\mathbf{m}_\ell. \tag{38}$$

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

$$\mathbf{l}_\ell = F^T\mathbf{m}_r. \tag{39}$$

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:

$$F\mathbf{e}_\ell = 0 \tag{40}$$

$$\mathbf{e}_r^T F = 0 \tag{41}$$

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

$$F = K_r^{-T} E K_\ell^{-1}. \tag{42}$$

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 (34) gives: 10

$$\zeta_r F^T \mathbf{m}_r = \zeta_\ell (\mathbf{e}_\ell \times \mathbf{m}_\ell). \tag{43}$$

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 $(\mathbf{e}_\ell)$.

4.4.2 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 (37) to compute the unknown matrix F.

We need to convert Equation (37) from its bilinear form to a form that matches the null-space problem. To this end we use again the vec operator, as in the DLT algorithm:

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

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.

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

The fundamental matrix F is computed by solving the resulting linear system of equations, for $n \geq 8$.

If the data are not exact and more than 8 points are used, 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.

4.4.3 Reconstruction up to a Projective Transformation

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

$$\mathbf{m}_\ell^i = P_\ell \mathbf{M}^i \quad \text{and} \quad \mathbf{m}_r^i = P_r \mathbf{M}^i, \quad \forall i \tag{44}$$

If T is any 4×4 invertible matrix, representing a collineation of $\mathbb{P}_3$, 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 $\mathbf{m}_\ell^i$. This shows that, if nothing is known but the image points, the structure $\mathbf{M}^i$ and the cameras can be determined 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 4.4.1), and from a (non-unique) factorization of F of the form

$$F = [\mathbf{e}_r]_\times A \tag{45}$$

two camera matrices P_ℓ and P_r:

$$P_\ell = [I|\mathbf{0}] \quad \text{and} \quad P_r = [A|\mathbf{e}_r], \tag{46}$$

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 matrix A in the factorization of F can be set to $A = -[\mathbf{e}_r]_\times F$ (this is called the *epipolar projection matrix* [17]). 08

Unlike the essential matrix, F does not admit a unique factorization, whence the projective ambiguity follows.

Indeed, for any A satisfying Equation (45), also $A + \mathbf{e}_r \mathbf{x}^T$ for any vector $\mathbf{x}$, satisfies Equation (45).

More in general, any homography induced by a plane can be taken as the A matrix.

4.5 More on calibration

Calibration problems can be cast as determining the transformation between two reference frames. Depending on the nature of the available measures and where the reference frames are attached we have four calibration or *orientation* problems[1]:

Relative orientation is the problem of determining the position and attitude of one perspective camera with respect to another camera from correspondences between points in 2-D images (See. Sec. 4.3.2).

Absolute orientation is the problem of aligning two sets of points, whose 3-D locations have been measured (or reconstructed) in two different reference frames.

Exterior orientation is the problem of determining the position and attitude of a perspective camera from correspondences between 3-D points and their 2-D images.

Interior orientation is the problem of determining the (affine) transformation from normalized camera coordinates to pixel coordinates, i.e., the intrinsic parameters of the camera.

[1]this terminology comes from Photogrammetry.

4.5.1 Absolute orientation (with scaling)

Given two sets of 3-D points $\mathbf{X}^i$ and $\mathbf{Y}^i$, related by[2]

$$\mathbf{X}^i = s(R\mathbf{Y}^i + t) \quad \text{for all } i = 1 \ldots N \tag{47}$$

we are required to find the rotation matrix R, the vector $\mathbf{t}$ and the scalar s.

Summing these equations for all i and dividing by N shows that the translation is found with:

$$t = \frac{1}{s}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{X}^i\right) - R\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{Y}^i\right)$$

Combining this with Eq. (47) gives

$$\bar{\mathbf{X}}^i = sR\bar{\mathbf{Y}}^i$$

where $\bar{\mathbf{X}}^i = \mathbf{X}^i - \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}^i$ and $\bar{\mathbf{Y}}^i = \mathbf{Y}^i - \frac{1}{N}\sum_{i=1}^{N}\mathbf{Y}^i$.

Because the rotation matrix does not change the length of the vectors, we can immediately solve for the scale from $||\bar{\mathbf{X}}^i|| = s||\bar{\mathbf{Y}}^i||$.

[2]Please note the change of notation: points are represented by Cartesian (non-homogeneous) coordinates.

We are left with the problem of estimating the unknown rotation between two sets of points.

Let $\bar{X}$ be the $3 \times N$ matrix formed by stacking the points $\bar{\mathbf{X}}^i$ side by side and $\bar{Y}$ be the matrix formed likewise by stacking the scaled points $s\bar{\mathbf{Y}}^i$.In presence of noise, we would like to minimize the sum of the square of the errors, or

$$\sum_{i=1}^{N} ||\bar{\mathbf{X}}^i - sR\bar{\mathbf{Y}}^i||^2 = ||\bar{X} - R\bar{Y}||_F^2$$

where $||\cdot||_F$ is the Frobenius norm.

This problem is known as the Orthogonal Procrustes Problem and the solution is given by [16]

$$R = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(VU^T) \end{bmatrix} U^T$$

where $UDV^T = \bar{Y}\bar{X}^T$ is the SVD of the 3×3 matrix $\bar{Y}\bar{X}^T$.

4.5.2 Exterior orientation

Exterior orientation (also called *Perspective n-Points* camera pose) is a problem that appears repeatedly in computer vision, but in context different from 3-D reconstruction, such as visual servoing and augmented reality.

Given a number of point correspondences $\mathbf{m}^i \leftrightarrow \mathbf{M}^i$ and the intrinsic camera parameters K, we are required to find a rotation matrix R and a translation vector $\mathbf{t}$ (which specify attitude and position of the camera) such that:

$$\zeta^i K^{-1}\mathbf{m}^i = [R|\mathbf{t}]\mathbf{M}^i = (R\tilde{\mathbf{M}}^i + \mathbf{t}) \quad \text{for all } i. \tag{48}$$

One could immediately solve this problem by doing camera resection with DLT in normalized camera coordinates. The algorithm is linear, but it does not enforces the orthonormality constraints on the rotation matrix.

Instead, we present here the linear method proposed by Fiore [6]. He first recovers the unknown depths ζ^i, and then observes that what is left is an absolute orientation problem, whose solution yields a rotation matrix which is inherently orthonormal.

In order to recover the depths, let's write Eq. (48) in matrix form:

$$K^{-1}\underbrace{\left[\zeta^1\mathbf{m}^1,\ \zeta^2\mathbf{m}^2,\ \ldots\ \zeta^n\mathbf{m}^n\right]}_{W} = [R|\mathbf{t}]\underbrace{\left[\mathbf{M}^1,\ \mathbf{M}^2,\ \ldots\ \mathbf{M}^n\right]}_{M}. \tag{49}$$

Let $r = \operatorname{rank} M$. Take its SVD: $M = UDV^T$ and let V_2 be a matrix composed by the last $n - r$ columns of V, which spans the null-space of M. Then, $MV_2 = 0_{3\times(n-r)}$, and also

$$K^{-1}WV_2 = 0_{3\times(n-r)} \tag{50}$$

By taking vec on both sides we get:

$$(V_2^T \otimes K^{-1})\operatorname{vec}(W) = \mathbf{0}. \tag{51}$$

Let us observe that:

$$\operatorname{vec}(W) = \begin{bmatrix} \zeta^1\mathbf{m}^1 \\ \zeta^1\mathbf{m}^2 \\ \vdots \\ \zeta^n\mathbf{m}^n \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{m}^1 & 0 & \ldots & 0 \\ & & \ddots & \\ 0 & 0 & \ldots & \mathbf{m}^n \end{bmatrix}}_{D}\underbrace{\begin{bmatrix} \zeta^1 \\ \vdots \\ \zeta^n \end{bmatrix}}_{\boldsymbol{\zeta}} \tag{52}$$

Hence

$$\left((V_2^T \otimes K^{-1})D\right) \boldsymbol{\zeta} = \mathbf{0}. \tag{53}$$

From the last equation the depths $\boldsymbol{\zeta}$ can be recovered (up to a scale factor) by solving a null-space problem.

The size of the coefficients matrix is $3(n-r) \times n$, and in order to determine a one-parameter family of solutions, it must have rank $n-1$, hence $3(n-r) \geq n-1$.

Therefore, at least $n \geq (3r-1)/2$ points are needed. If points are in general position, 6 are sufficient, but if they are on a plane, only 4 suffices.

Now that the left side of Eq. (48) is known, up to a scale factor, we are left with an absolute orientation (with scale) problem:

$$\zeta^i K^{-1}\mathbf{m}^i = s(R\tilde{\mathbf{M}}^i + \mathbf{t}) \quad \text{for all } i. \tag{54}$$

which we solve using the algorithm of Sec. 4.5.1. As a result, the rotation matrix estimate is orthonormal by construction.

5 Autocalibration

The aim of *autocalibration* is to compute the intrinsic parameters, starting from weakly calibrated cameras.

More in general, the task is to recover metric properties of camera and/or scene, i.e., to compute a Euclidean reconstruction.

There are two classes of methods:

1. Direct: solve directly for the intrinsic parameters.
2. Stratified: first obtain a projective reconstruction and then transform it to a Euclidean reconstruction (in some cases an affine reconstruction is obtained in between).

The reader is referred to [7] for a review of autocalibration, and to [19, 26, 14, 21, 20, 9] for classical and recent work on the subject.

5.1 Counting argument

Consider m cameras. The difference between the d.o.f. of the multifocal geometry (e.g. 7 for two views) and the d.o.f. of the rigid displacements (e.g. 5 for two views) is the number of independent constraints available for the computation of the intrinsic parameters (e.g. 2 for two views).

The multifocal geometry of m cameras (represented by the m-focal tensor) has $11m - 15$ d.o.f. Proof: a set of m cameras have $11m$ d.o.f., but they determine the m-focal geometry up to a collineation of $\mathbb{P}_3$, which has 15 d.o.f. The net sum is $11m - 15$ d.o.f.

On the other hand, the rigid displacements in m views are described by $6m - 7$ parameters: $3(m-1)$ for rotations, $2(m-1)$ for translations, and $m-2$ ratios of translation norms.

Thus, m *weakly calibrated views give* $5m - 8$ *constraints available for computing the intrinsic parameters*.

Let us suppose that m_k parameters are known and m_c parameters are constant.

The first view introduces $5 - m_k$ unknowns. Every view but the first introduces $5 - m_k - m_c$ unknowns.

Therefore, the unknown intrinsic parameters can be computed provided that

$$5m - 8 \geq (m-1)(5 - m_k - m_c) + 5 - m_k. \tag{55}$$

For example, if the intrinsic parameters are constant, three views are sufficient to recover them.

If one parameter (usually the skew) is known and the other parameters are varying, at least eight views are needed.

5.2 A simple direct method

If we consider two views, two independent constraints are available for the computation of the intrinsic parameters from the fundamental matrix.

Indeed, F has 7 d.o.f, whereas E, which encode the rigid displacement, has only 5 d.o.f. There must be two additional constraint that E must satisfy, with respect to F.

In particular, these constraints stem from the equality of two singular values of the essential matrix (Theorem 4.1) which can be decomposed in two independent polynomial equations.

Let F_{ij} be the (known) fundamental matrix relating views i and j, and let K_i and K_j be the respective (unknown) intrinsic parameter matrices.

The idea of [20] is that the matrix

$$E_{ij} = K_i^T F_{ij} K_j, \tag{56}$$

satisfies the constraints of Theorem 4.1 only if the intrinsic parameters are correct.

Hence, the cost function to be minimized is

$$C(K_i,\ i = 1 \dots n) = \sum_{i=1}^{n} \sum_{j>n}^{n} w_{ij} \frac{{}^1\sigma_{ij} - {}^2\sigma_{ij}}{{}^1\sigma_{ij} + {}^2\sigma_{ij}}, \tag{57}$$

where ${}^1\sigma_{ij} > {}^2\sigma_{ij}$ are the non zero singular values of E_{ij} and w_{ij} are normalized weight factors (linked to the reliability of the fundamental matrix estimate).

The previous counting argument shows that, in the general case of n views, the $n(n-1)/2$ two-view constraints that can be derived are not independent, nevertheless they can be used as they over-determine the solution.

A non-linear least squares solution is obtained with an iterative algorithm (e.g. Gauss-Newton) that uses analytical derivatives of the cost function.

A starting guess is needed, but this cost function is less affected than others by local minima problems. A globally convergent algorithm based on this cost function is described in [8].

5.3 Stratification

We have seen that a projective reconstruction can be computed starting from points correspondences only (weak calibration), without any knowledge of the camera matrices.

Projective reconstruction differs from Euclidean by an unknown projective transformation in the 3-D projective space, which can be seen as a suitable change of basis.

Starting from a projective reconstruction the problem is computing the transformation that "straighten" it, i.e., that upgrades it to an Euclidean reconstruction.

To this purpose the problem is *stratified* [17, 4] into different representations: depending on the amount of information and the constraints available, it can be analyzed at a projective, affine, or Euclidean level.

Let us assume that a projective reconstruction is available, that is a sequence P_i of $m+1$ camera matrices and a set $\mathbf{M}^j$ of $n+1$ 3-D points such that:

$$\mathbf{m}_i^j \simeq P_i \mathbf{M}^j \quad i = 0 \dots m, \quad j = 0 \dots n. \tag{58}$$

Without loss of generality, we can assume that camera matrices writes:

$$P_0 = [I \mid \mathbf{0}]; \qquad P_i = [A_i \mid \mathbf{e}_i] \quad \text{for } i = 1 \dots m \tag{59}$$

We are looking for the a 4×4 non-singular matrix T that upgrades the projective reconstruction to Euclidean:

$$\mathbf{m}_i^j \simeq \underbrace{P_i T}_{P_i^{\mathrm{E}}} \underbrace{T^{-1} \mathbf{M}^j}_{\text{structure}}, \tag{60}$$

$P_i^{\mathrm{E}} = P_i T$ is the Euclidean camera,

We can choose the first Euclidean-calibrated camera to be $P_0^{\mathrm{E}} = K_0[I \mid \mathbf{0}]$, thereby fixing arbitrarily the world reference frame:

$$P_0^{\mathrm{E}} = K_0[I \mid \mathbf{0}] \qquad P_i^{\mathrm{E}} = K_i[R_i \mid \mathbf{t}_i] \quad \text{for } i = 1 \dots m. \tag{61}$$

With this choice, it is easy to see that $P_0^{\mathrm{E}} = P_0 T$ implies

$$T = \begin{bmatrix} K_0 & \mathbf{0} \\ \mathbf{r}^T & s \end{bmatrix} \quad (62)$$

where $\mathbf{r}^T$ is a 3-D vector and s is a scale factor, which we will arbitrarily set to 1 (the Euclidean reconstruction is up to a scale factor).

Under this parametrization T is clearly non singular, and it depends on eight parameters.

Substituting (62) in $P_i^{\mathrm{E}} \simeq P_i T$ gives

$$P_i^{\mathrm{E}} = [K_i R_i \mid K_i \mathbf{t}_i] \simeq P_i T = [A_i K_0 + \mathbf{e}_i \mathbf{r}^T \mid \mathbf{e}_i] \quad \text{for } i > 0 \quad (63)$$

and, considering only the leftmost 3×3 submatrix, gives

$$K_i R_i \simeq A_i K_0 + \mathbf{e}_i \mathbf{r}^T = P_i \begin{bmatrix} K_0 \\ \mathbf{r}^T \end{bmatrix} \quad (64)$$

Rotation can be eliminated using $RR^T = I$, leaving:

$$K_i K_i^T \simeq P_i \begin{bmatrix} K_0 K_0^T & K_0 \mathbf{r} \\ \mathbf{r}^T K_0^T & \mathbf{r}^T \mathbf{r} \end{bmatrix} P_i^T \tag{65}$$

This is the basic equation for autocalibration (called *absolute quadric constraint*), relating the unknowns K_i $(i = 0 \dots m)$ and $\mathbf{r}$ to the available data P_i (obtained from weakly calibrated images).

Note that (65) contains five equations, because the matrices of both members are symmetric, and the homogeneity reduces the number of equations with one.

5.3.1 Geometric interpretation

Under camera matrix P the outline of the quadric Q is the conic C given by:

$$C^* \simeq PQ^*P^T \tag{66}$$

where C^* is the dual conic and Q^* is the dual quadric. An expression with Q and C may be derived, but it is quite complicated. C^* (resp. Q^*) is the adjoint matrix of C (resp. Q). If C is non singular, then $C^* = C^{-1}$.

In the beginning we introduced the absolute conic $\mathbf{\Omega}$, which is invariant under similarity transformation, hence deeply linked with the Euclidean stratum.

In a Euclidean frame, its equation is $x_1^2 + x_2^2 + x_3^2 = 0 = x_4$.

The absolute conic may be regarded as a special quadric (a disk quadric), therefore its dual is a quadric, the *dual absolute quadric*, denoted by $\mathbf{\Omega}^*$. Its representation is:

$$\mathbf{\Omega}^* = \mathrm{diag}(1, 1, 1, 0). \tag{67}$$

As we already know, the *image of the absolute conic* under camera matrix P^{E} is given by $\boldsymbol{\omega} = (KK^T)^{-1}$, that is :

$$\boldsymbol{\omega}^* = (KK^T) \simeq P^{\mathrm{E}} \boldsymbol{\Omega}^* P^{\mathrm{E}T} \tag{68}$$

This property is independent on the choice of the projective basis. What changes is the representation of the dual absolute quadric, which is mapped to

$$\boldsymbol{\Omega}^* = T \mathrm{diag}(1,1,1,0) T^T. \tag{69}$$

under the collineation T.

Substituting T from Eq. (62) into the latter gives:

$$\boldsymbol{\Omega}^* = \begin{bmatrix} K_0 K_0^T & K_0 \mathbf{r} \\ \mathbf{r}^T K_0^T & \mathbf{r}^T \mathbf{r} \end{bmatrix} \tag{70}$$

Recalling that $\boldsymbol{\omega}_i^* = K_i K_i^T$, then Eq. (65) is equivalent to

$$\boldsymbol{\omega}_i^* \simeq P_i \boldsymbol{\Omega}^* P_i^T \tag{71}$$

5.3.2 Solution strategies

Autocalibration requires to solve Eq. (71), with respect to $\mathbf{\Omega}^*$ (and $\boldsymbol{\omega}_i^*$).

If $\mathbf{\Omega}^*$ is known, the collineation T that upgrades cameras from projective to Euclidean is obtained by decomposing $\mathbf{\Omega}^*$ as in Eq. (69).

$\mathbf{\Omega}^*$ might be parametrized as in Eq. (70) with 8 d.o.f. or parametrized as a generic 4×4 symmetric matrix (10 d.o.f.). The latter is an over-parametrization, as $\mathbf{\Omega}^*$ is also singular and defined up to a scale factor (which gives again 8 d.o.f.).

There are several strategies for dealing with the scale factor.

- Introduce the scale factor explicitly as an additional unknown [13]:

$$\boldsymbol{\omega}_i^* - \lambda_i P_i \mathbf{\Omega}^* P_i^T = \mathbf{0} \tag{72}$$

  This gives 6 equations but introduces one additional unknown (the net sum is 5).

- Eliminate it by using the same idea of the cross product for 3-D vectors [26].

$$\mathrm{vech}(\boldsymbol{\omega}_i^*) \simeq \mathrm{vech}(P_i \boldsymbol{\Omega}^* P_i^T) \iff \mathrm{rank}\,\underbrace{[\mathrm{vech}(\boldsymbol{\omega}_i^*) |\, \mathrm{vech}(P_i \boldsymbol{\Omega}^* P_i^T)]}_{B} = 1$$

  where vech is the column-wise vectorization with the upper portion excluded, as matrices in Eq. (71) are symmetric.

  This is tantamount to say that every 2×2 minor of B is zero. There are 15 different order-2 minors of a 6×2 matrix, but only 5 equations are independent. 13

- Use a matrix norm (namely, Frobenius norm) [21]:

$$\frac{\boldsymbol{\omega}_i^*}{||\boldsymbol{\omega}_i^*||_F} - \frac{P_i \boldsymbol{\Omega}^* P_i^T}{||P_i \boldsymbol{\Omega}^* P_i^T||_F} = \mathbf{0} \tag{73}$$

In any case, a non-linear least-squares problem has to be solved. Available numerical techniques (based on the Gauss-Newton method) are iterative, and requires an estimate of the solution to start.

This can be obtained by doing an educated guess about skew, principal point and aspect ratio, and solve the linear problem that results [22].

Linear solution

If some of the intrinsic parameters are known, one can perform a partial normalization of the coordinates, such that the corresponding elements of $\boldsymbol{\omega}_i^*$ vanish. Linear equations on $\boldsymbol{\Omega}^*$ are generated from zero-entries of $\boldsymbol{\omega}_i^*$ (because this eliminates the scale factor):

$$\boldsymbol{\omega}_i^*(k, \ell) = 0 \;\Rightarrow\; \mathbf{p}_{i,k}^T \boldsymbol{\Omega}^* \mathbf{p}_{i,\ell} = 0$$

where $\mathbf{p}_{i,k}^T$ is the k-th row of P_i.

Likewise, linear constraints on $\boldsymbol{\Omega}^*$ can be obtained from the equality of elements in the the upper (or lower) triangular part of $\boldsymbol{\omega}_i^*$ (because $\boldsymbol{\omega}_i^*$ is symmetric).

In order to be able to solve linearly for $\boldsymbol{\Omega}^*$, at least 10 linear equations must be stacked up, to form a homogeneous linear system, which can be solved as usual (via SVD). Singularity of $\boldsymbol{\Omega}^*$ can be enforced a-posteriori by forcing the smallest singular value to zero.

If the principal point is known, $\boldsymbol{\omega}_i^*(1, 3) = 0 = \boldsymbol{\omega}_i^*(2, 3)$ and this gives two linear constraints. If, in addition, skew is zero we have $\boldsymbol{\omega}_i^*(1, 2) = 0$. Known aspect ratio r provides a further constraint: $r\boldsymbol{\omega}_i^*(1, 1) = \boldsymbol{\omega}_i^*(2, 2)$.

Constant intrinsic parameters

If all the cameras has the same intrinsic parameters, so $K_i = K$, then Eq. (65) becomes

$$KK^T \simeq P_i \begin{bmatrix} KK^T & K\mathbf{r} \\ \mathbf{r}^T K^T & \mathbf{r}^T\mathbf{r} \end{bmatrix} P_i^T \tag{74}$$

The constraints expressed by Eq. (74) are called the Kruppa constraints in [13].

Since each camera matrix, apart from the first one, gives five equations in the eight unknowns, a unique solution is obtained when at least three views are available.

The resulting system of equations is solved with a non-linear least-squares technique (e.g. Gauss-Newton).

6 Dealing with errors

In this section we will approach estimation problems from a more “practical” point of view.

First, we will discuss how the presence of errors in the data affects our estimates and describe the countermeasures that must be taken to obtain a good estimate.

Errors can be small (the match is correct but the position of the point has a limited accuracy), and these are usually modeled as Gaussian noise, or large (the match is wrong) and these are called *outliers*.

Since they have a different nature, they will be treated differently.

6.1 Pre-conditioning

In presence of noise (or errors) on input data, the accuracy of the solution of a linear system depends crucially on the *condition number* of the system. The lower the condition number, the less the input error gets amplified (the system is more stable).

As [11] pointed out, it is crucial for linear algorithms (as the DLT algorithm) that input data is properly pre-conditioned, by a suitable coordinate change (origin and scale): points are translated so that their centroid is at the origin and are scaled so that their average distance from the origin is $\sqrt{2}$.

This improves the condition number of the linear system that is being solved.

Apart from improved accuracy, this procedure also provides invariance under similarity transformations in the image plane.

6.2 Algebraic vs geometric error

Measured data (i.e., image or world point positions) is noisy.

Usually, to counteract the effect of noise, we use more equations than necessary and solve with least-squares.

What is actually being minimized by least squares?

In a typical null-space problem formulation $Ax = 0$ (like the DLT algorithm) the quantity that is being minimized is the square of the residual $||Ax||$.

In general, if $||Ax||$ can be regarded as a distance between the geometrical entities involved (points, lines, planes, etc..), than what is being minimized is a geometric error, otherwise (when the error lacks a good geometrical interpretation) it is called an algebraic error.

All the linear algorithm (DLT and others) we have seen so far minimize an algebraic error. Actually, there is no justification in minimizing an algebraic error apart from the ease of implementation, as it results in a linear problem.

Usually, the minimization of a geometric error is a non-linear problem, that admit only iterative solutions and requires a starting point.

So, why should we prefer to minimize a geometric error? Because:

- The quantity being minimized has a meaning
- The solution is more stable
- The solution is invariant under Euclidean transforms

Often linear solution based on algebraic residuals are used as a starting point for a non-linear minimization of a geometric cost function, which "gives the solution a final polish" [10].

Geometric error for resection

The goal is to estimate the camera matrix, given a number of correspondences $(\mathbf{m}^j, \mathbf{M}^j) \quad j = 1 \dots n$

The geometric error associated to a camera estimate $\hat{P}$ is the distance between the measured image point $\mathbf{m}^j$ and the re-projected point $\hat{P}_i \mathbf{M}^j$:

$$\min_{\hat{P}} \sum_j d(\hat{P}\mathbf{M}^j, \mathbf{m}^j)^2 \tag{75}$$

where $d()$ is the Euclidean distance between the homogeneous points.

The DLT solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton)

Geometric error for triangulation

The goal is to estimate the 3-D coordinates of a point $\mathbf{M}$, given its projection $\mathbf{m}_i$ and the camera matrix $\mathbf{P}_i$ for every view $i = 1 \ldots m$.

The geometric error associated to a point estimate $\hat{\mathbf{M}}$ in the i-th view is the distance between the measured image point $\mathbf{m}_i$ and the re-projected point $P_i \hat{\mathbf{M}}$:

$$\min_{\hat{\mathbf{M}}} \sum_i d(P_i \hat{\mathbf{M}}, \mathbf{m}_i)^2 \tag{76}$$

where $d()$ is the Euclidean distance between the homogeneous points.

The linear solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton).

Geometric error for F

The goal is to estimate F given a a number of point correspondences $\mathbf{m}_\ell^i \leftrightarrow \mathbf{m}_r^i$.

The geometric error associated to an estimate $\hat{F}$ is given by the distance of conjugate points from conjugate lines (note the symmetry):

$$\min_{\hat{F}} \sum_j d(\hat{F}\mathbf{m}_\ell^j, \mathbf{m}_r^j)^2 + d(\hat{F}^T\mathbf{m}_r^j, \mathbf{m}_\ell^j)^2 \tag{77}$$

where $d()$ here is the Euclidean distance between a line and a point (in homogeneous coordinates).

The eight-point solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton).

Note that F must be suitably parameterized, as it has only seven d.o.f. 11

Geometric error for H

The goal is to estimate H given a a number of point correspondences $\mathbf{m}_\ell^i \leftrightarrow \mathbf{m}_r^i$.

The geometric error associated to an estimate $\hat{H}$ is given by the symmetric distance between a point and its transformed conjugate:

$$\min_{\hat{H}} \sum_j d(\hat{H}\mathbf{m}_\ell^j, \mathbf{m}_r^j)^2 + d(\hat{H}^{-1}\mathbf{m}_r^j, \mathbf{m}_\ell^j)^2 \tag{78}$$

where $d()$ is the Euclidean distance between the homogeneous points. This also called the *symmetric transfer error*.

The linear solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton).

Bundle adjustment (reconstruction)

If measurements are noisy, the projection equation will not be satisfied exactly by the reconstructed camera matrices and structure.

We wish to minimize the image distance between the re-projected point $\hat{P}_i \hat{\mathbf{M}}^j$ and measured image points $\mathbf{m}_i^j$ for every view in which the 3-D point appears:

$$\min_{\hat{P}_i, \hat{\mathbf{M}}^j} \sum_{i,j} d(\hat{P}_i \hat{\mathbf{M}}^j, \mathbf{m}_i^j)^2 \tag{79}$$

where $d()$ is the Euclidean distance between the homogeneous points.

As m and n increase, this becomes a very large minimization problem.

However the Jacobian of the residual is very sparse (not all points are seen in all views) and this can be exploited to gain efficiency.

If the reconstruction is projective $\hat{P}_i$ is parameterized with its 11 d.o.f. whereas if the reconstruction is Euclidean, one should use $\hat{P}_i = \hat{K}_i[\hat{R}_i|\hat{\mathbf{t}}_i]$ where the rotation has to be suitably parameterized with 3 d.o.f.

See [27] for a review and a more detailed discussion on bundle adjustment.

6.3 Robust estimation

Up to this point, we have assumed that the only source of error affecting correspondences is in the measurements of point's position. This is a small-scale noise that gets averaged out with least-squares.

In practice, we can be presented with *mismatched* points, which are *outliers* to the noise distribution (i.e., rogue measurements following a different, unmodelled, distribution).

These outliers can severely disturb least-squares estimation (even a single outlier can totally offset the least-squares estimation, as illustrated in Fig. 6.)

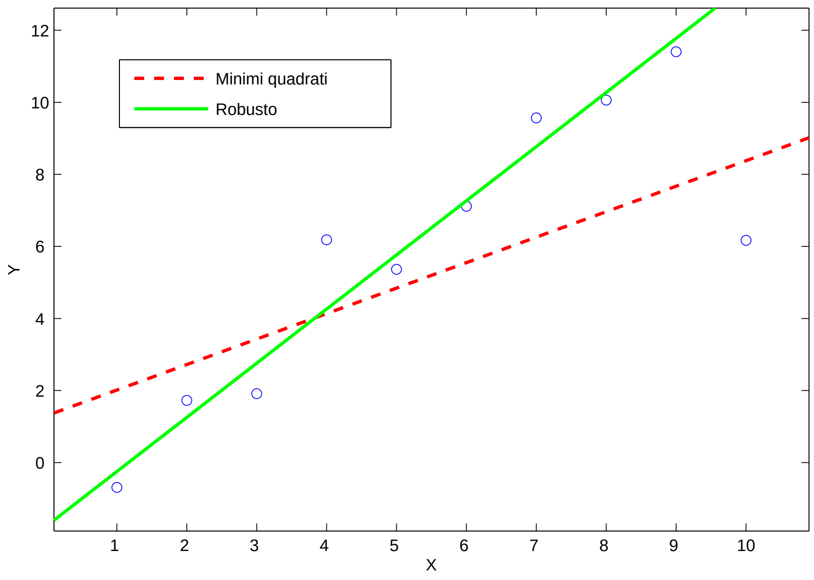


Fig. 6. A single outlier can severely offset the least-squares estimate (red line), whereas the robust estimate (green line) is unaffected.

The goal of robust estimation is to be insensitive to outliers (or at least to reduce sensitivity).

M-estimators

Least squares:

$$\min_\theta \sum_i (r_i/\sigma_i)^2 \tag{80}$$

where θ are the regression coefficient (what is being estimated) and r_i is the residual. M-estimators are based on the idea of replacing the squared residuals by another function of the residual, yielding

$$\min_\theta \sum_i \rho(r_i/\sigma_i) \tag{81}$$

ρ is a symmetric function with a unique minimum at zero that grows sub-quadratically, called *loss function*.

Differentiating with respect to θ yields:

$$\sum_i \frac{1}{\sigma_i}\rho'(r_i/\sigma_i)\frac{dr_i}{d\theta} = 0 \tag{82}$$

The M-estimate is obtained by solving this system of non-linear equations.

RANSAC

Given a model that requires a minimum of p data points to instantiate its free parameters θ, and a set of data points S containing outliers:

1. Randomly select a subset of p points of S and instantiate the model from this subset
2. Determine the set S_i of data points that are within an error tolerance t of the model. S_i is the consensus set of the sample.
3. If the size of S_i is greater than a threshold T, re-estimate the model (possibly using least-squares) using S_i (the set of inliers) and terminate.
4. If the size of S_i is less than T, repeat from step 1.
5. Terminate after N trials and choose the largest consensus set found so far.

Three parameters need to be specified: t, T and N.

Both T and N are linked to the (unknown) fraction of outliers ϵ.

N should be large enough to have a high probability of selecting at least one sample containing all inliers. The probability to randomly select p inliers in N trials is:

$$P = 1 - (1 - (1 - \epsilon)^p)^N \tag{83}$$

By requiring that P must be near 1, N can be solved for given values of p and ϵ.

T should be equal to the expected number of inliers, which is given (in fraction) by $(1 - \epsilon)$.

At each iteration, the largest consensus set found so fare gives a lower bound on the fraction of inliers, or, equivalently, an upper bound on the number of outliers. This can be used to adaptively adjust the number of trials N.

t is determined empirically, but in some cases it can be related to the probability that a point under the threshold is actually an inlier [10].

As pointed out in [25], RANSAC can be viewed as a particular M-estimator.

The objective function that RANSAC maximizes is the number of data points having absolute residuals smaller that a predefined value t. This may be seen a minimising a binary loss function that is zero for small (absolute) residuals, and 1 for large absolute residuals, with a discontinuity at t.

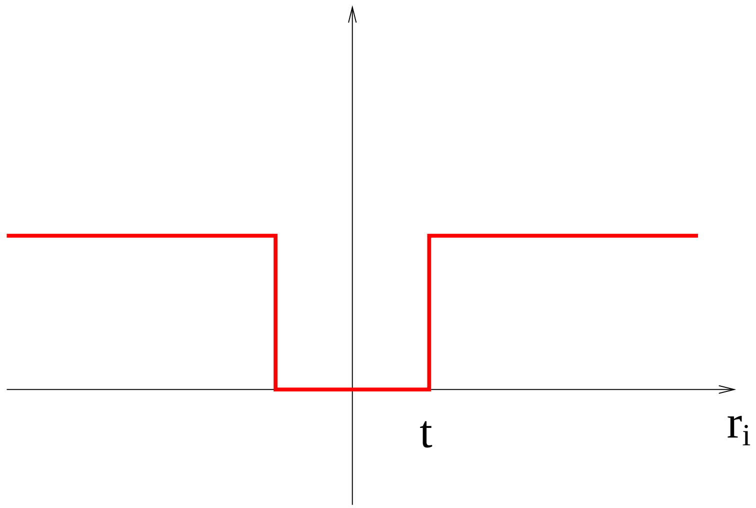


Fig. 7. RANSAC loss function

By virtue of the prespecified inlier band, RANSAC can fit a model to data corrupted by substantially more than half outliers.

LMedS

Another popular robust estimator is the Least Median of Squares. It is defined by:

$$\min_{\theta} \mathrm{med}_i r_i \tag{84}$$

It can tolerate up to 50% of outliers, as up to half of the data point can be arbitrarily far from the "true" estimate without changing the objective function value.

Since the median is not differentiable, a random sampling strategy similar to RANSAC is adopted. Instead of using the consensus, each sample of size p is scored by the median of the residuals of all the data points. The model with the least median (lowest score) is chosen.

A final weighted least-squares fitting is used.

With respect to RANSAC, LMedS can tolerate "only" 50% of outliers, but requires no setting of thresholds.

7 Sequential structure from motion

In this section, we discuss the problem of structure and motion recovery from **multiple views** as it has been successfully approached in practical cases (namely, Bundler [24]).

We present an approach to reconstruct the projective (or Euclidean) structure and motion from a **sequence** of images[3].

We assume that for each pair of consecutive views we are given a set of potentially corresponding feature points.

These feature points are typically obtained by using a keypoint detector and correspondences are identified by comparing the respective descriptors.

[3] We assume that even if images are unordered they can be suitably sequenced

Overview of the approach

A sequential approach is used in practice, which offers the advantage that corresponding features are not required to be visible in all views.

The first step consists of computing the geometric relation between two consecutive views.

Once the epipolar geometry has been computed, the next step consists of reconstructing the structure and motion for the whole sequence.

The structure and motion is initialized for two views and then gradually extended toward the whole sequence.

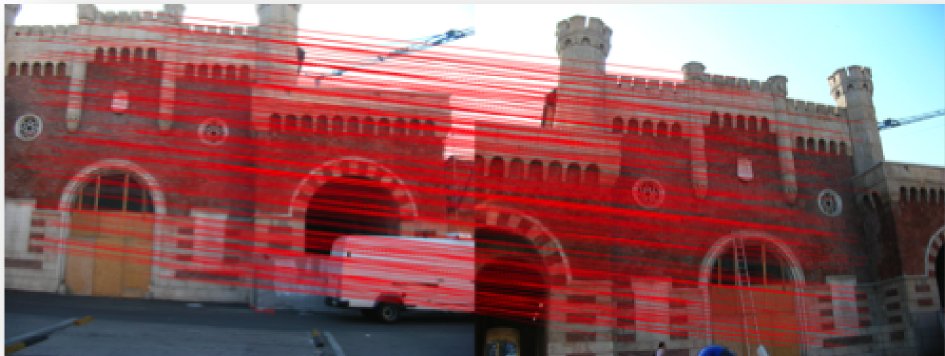
Finally, the solution is refined through a global minimization over all the unknown parameters.

Computing epipolar geometry

This consist in recovering the fundamental matrix (projective) or the essential matrix (Euclidean) from a set of corresponding points perturbed with noise and containing a significant proportion of outliers.

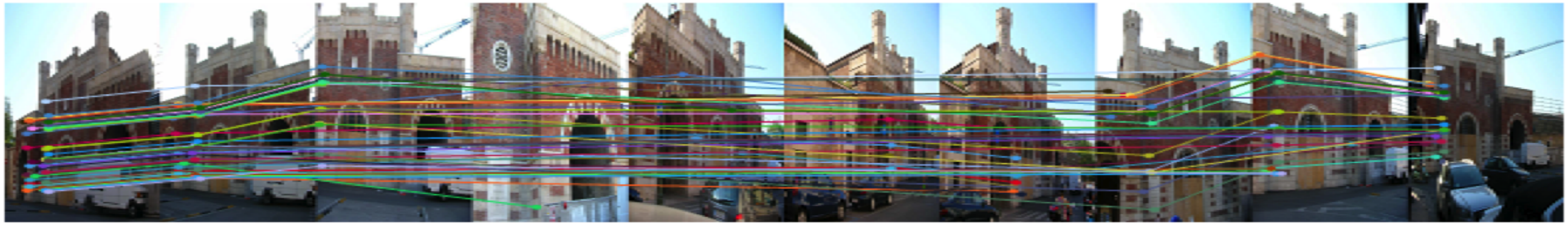
Hence, a robust procedure must be used to cope with outliers, followed by a non-linear refinement to obtain an optimal estimate in presence of noise.

The epipolar geometry is essentially used only to **validate matches**.



Keyoints are then connected into multiple-view correspondences, called *tracks*.

Tracks consistency and track length can be used to reject spurious matches.

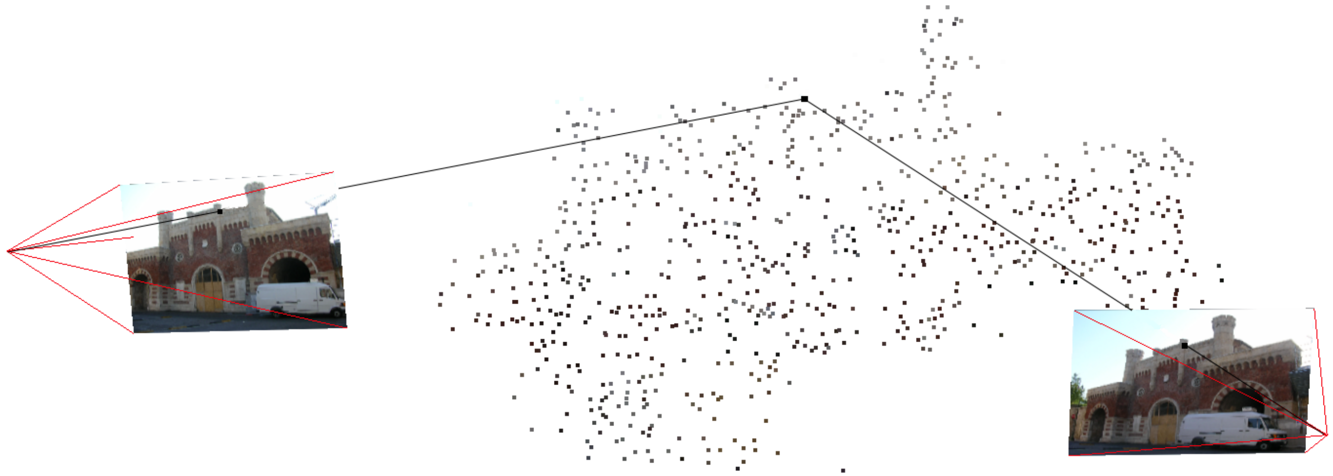


Initializing the structure and motion

Two images of the sequence are used to determine a reference frame. The world frame is aligned with the first camera P_1

The second camera P_2 is chosen so that the epipolar geometry corresponds to the computed **fundamental matrix** F (projective) or **essential matrix** E (Euclidean).

Once two projection matrices have been fully determined, the matches can be reconstructed through **triangulation**.



Updating the structure and motion

For every additional view $P_i \quad i > 2$, its orientation toward the preexisting reconstruction is determined.

This can be **exterior orientation** in the Euclidean case or **resection** (exterior + interior) in the projective case.

The matches that correspond to already reconstructed points are used to infer correspondences between 2D and 3D. Based on these, the projection matrix P_i is computed using a robust procedure.

The structure is updated by refining the position of 3D points that have been observed in the current view using **triangulation**.

When point matches becomes available that were not related to an existing point in the structure, then a new point can be initialized.

Once the structure and motion has been obtained, it is recommended to refine it through **bundle adjustment** (projective or Euclidean).

This should be done once every N views have been added, not only at the end, in order to contain error accumulation.

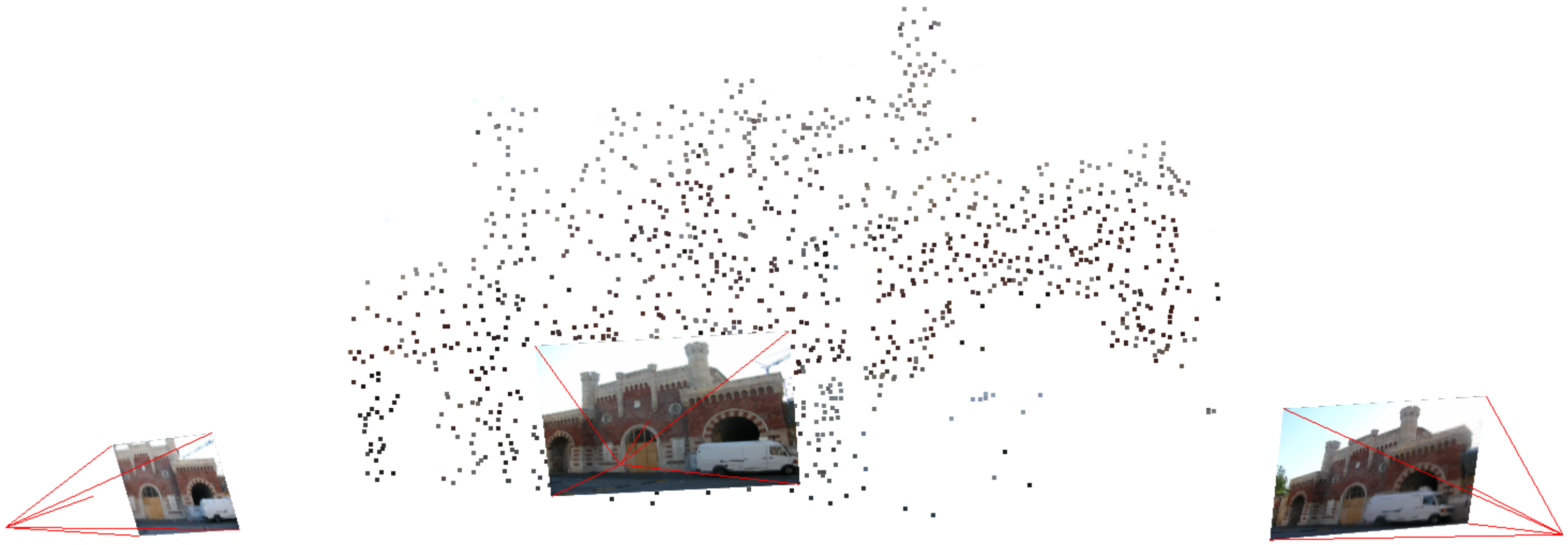


Fig. 8. Final reconstruction from the 10 images of *Porta Vescovo*

8 Further readings

General books on (Geometric) Computer Vision are: [3, 28, 5, 10].

References

[1] P. Beardsley, A. Zisserman, and D. Murray. Sequential update of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997.

[2] Myron Z. Brown, Darius Burschka, and Gregory D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):933–1008, August 2003.

[3] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

[4] O. Faugeras. Stratification of 3-D vision: projective, affine, and metric representations. *Journal of the Optical Society of America A*, 12(3):465–484, 1994.

[5] O. Faugeras and Q-T Luong. *The geometry of multiple images*. MIT Press, 2001.

[6] Paul D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001.

[7] A. Fusiello. Uncalibrated Euclidean reconstruction: A review. *Image and Vision Computing*, 18(6-7):555–563, May 2000.

[8] A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using interval analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1633–1638, December 2004.

[9] R. Hartley, E. Hayman, L. de Agapito, and I. Reid. Camera calibration and the search for infinity. In *Proceedings of the International Conference on Computer Vision*, 1999.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.

[11] R. I. Hartley. In defence of the 8-point algorithm. In *Proceedings of the International Conference on Computer Vision*, pages 1064–1071, Washington, DC, USA, 1995. IEEE Computer Society.

[12] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.

[13] A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Proceedings of the International Conference on Pattern Recognition*, pages 339–343, Vienna, 1996.

[14] A. Heyden and K. Åström. Minimal conditions on intrinsic parameters for Euclidean reconstruction. In *Proceedings of the Asian Conference on Computer Vision*, page XXX, Hong Kong, 1998.

[15] T.S. Huang and O.D. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, December 1989.

[16] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993.

[17] Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229, 1996.

[18] J. R. Magnus and H. Neudecker. "*Matrix Differential Calculus with Applications in Statistics and Econometrics*". John Wiley & Sons, revised edition, 1999.

[19] S. J. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[20] P.R.S. Mendonça and R. Cipolla. A simple technique for self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:500–505, 1999.

[21] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the International Conference on Computer Vision*, pages 90–95, Bombay, 1998.

[22] M. Pollefeys, F. Verbiest, and L. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proceedings of the European Conference on Computer Vision*, pages 837–851, 2002.

[23] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.

[24] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[25] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[26] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 609–614, Puerto Rico, 1997.

[27] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms*, pages 298–372. Springer-Verlag, 2000.

[28] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, 1998.