# Elements of Geometric Computer Vision

Andrea Fusiello[*]

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

DAGM 2006, Berlin

# Contents

# 1 Introduction

This notes 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 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 15 years and has been reported in comprehensive monographies, e.g., [5, 9, 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 3-D world.

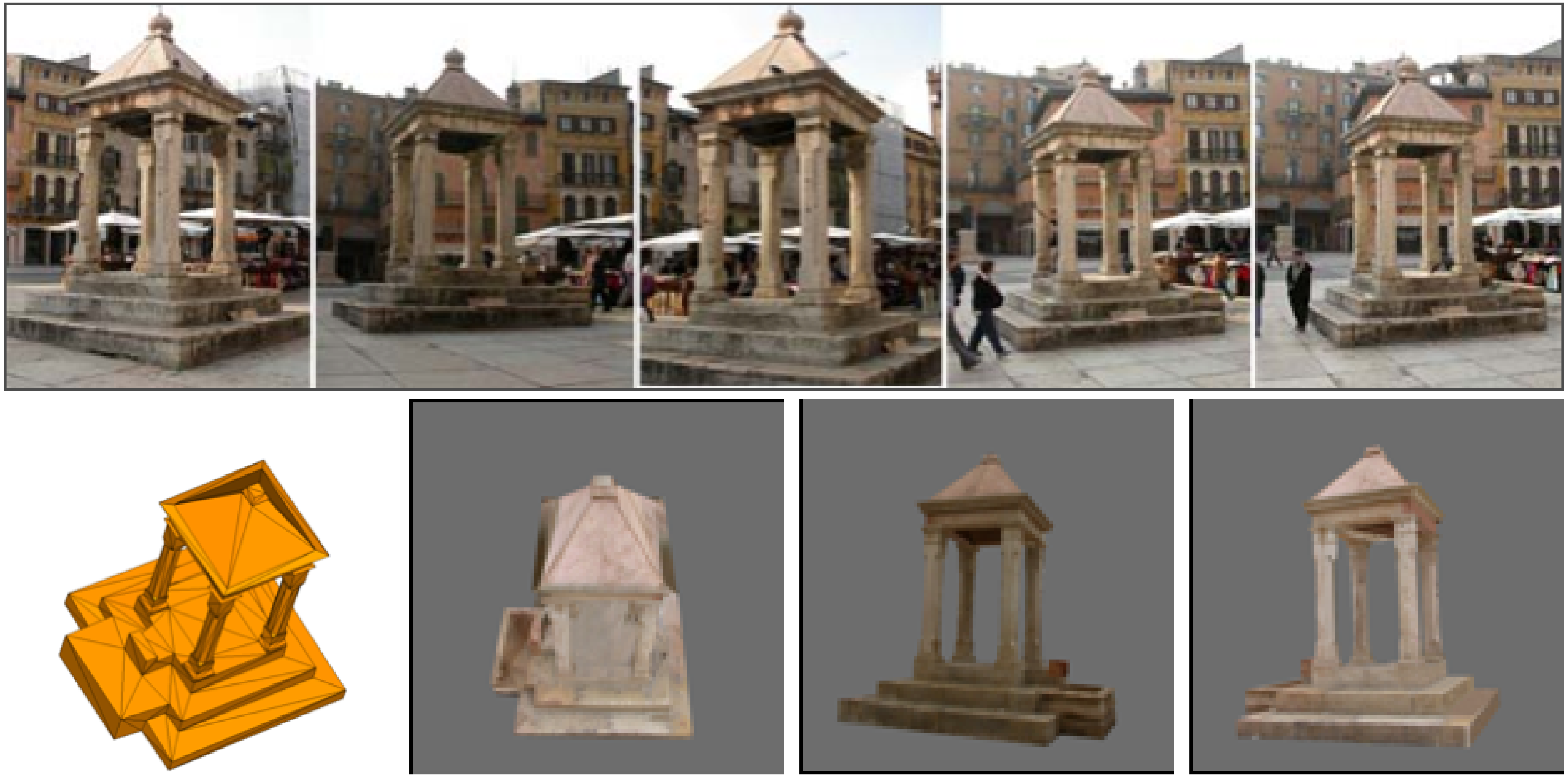


Fig. 1. Example of reconstruction from images. Original images (top row) and reconstructed model (bottom row).

# 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 $\Pi_\infty$. 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 $\Pi_\infty$. Their projective coordinates are of the form $(x, y, z, 1)$, where $(x, y, z)$ are the usual Cartesian coordinates.

$\Pi_\infty$ 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 $4 \times 4$ 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 plane parallel to the image plane containing the optical centre is called the *principal plane* or *focal plane* of the camera.

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

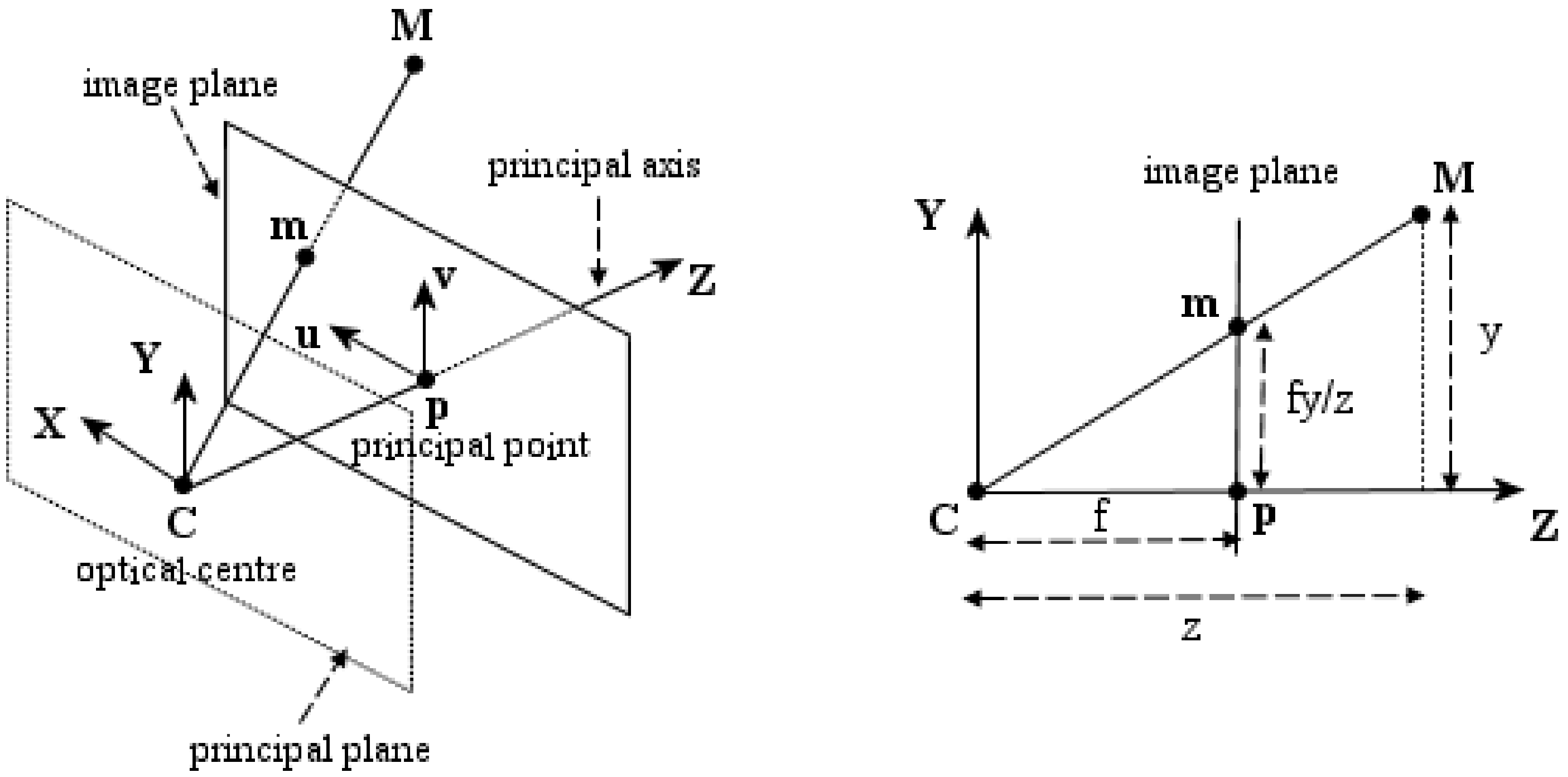


Fig. 2. Pin-hole camera geometry. The left figure illustrates the projection of the point **M** on the image plane by drawing the line through the camera centre **C** and the point to be projected. The right figure illustrates the same situation in the YZ plane, showing the similar triangles used to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

## 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

$$\zeta \mathbf{m} = P\mathbf{M} \tag{3}$$

where

- $\mathbf{M} = (x, y, z, 1)^T$ are the homogeneous coordinates of the 3-D point,
- $\mathbf{m} = (u, v, 1)^T$ are the homogeneous pixel coordinates of the image point,
- $\zeta$ is the distance of $\mathbf{M}$ from the focal plane of the camera and
- $P$ is the matrix describing the mapping, called the *camera projection matrix*.

The camera matrix is the product of two matrices

$$P = K[I|\mathbf{0}]G = K[R|\mathbf{t}] \tag{4}$$

## Extrinsic parameters

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

$G$ 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.

## Intrinsic parameters

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

$K$ 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 *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 $\theta$ between the axes (usually $\pi/2$).

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

## General camera

If $P$ describes a camera, also $\lambda P$ 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:

$$\mathbf{m} \simeq P\mathbf{M} \tag{7}$$

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

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

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

$$P = \lambda K[R|\mathbf{t}], \tag{8}$$

($\lambda$ 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.:

$$P\mathbf{C} = P \begin{pmatrix} \tilde{\mathbf{C}} \\ 1 \end{pmatrix} = \mathbf{0} \tag{9}$$

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:

$$\tilde{\mathbf{C}} = -P_{1:3}^{-1} P_4 \tag{10}$$

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:

$$\zeta \mathbf{m} = P\mathbf{M} = P\mathbf{M} - P\mathbf{C} = P(\mathbf{M} - \mathbf{C}) = P_{1:3}(\tilde{\mathbf{M}} - \tilde{\mathbf{C}}). \tag{11}$$

In particular, plugging Eq. (8), 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 $\lambda = 1$, $\zeta$ 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 $\mathbf{m}$:

$$\mathbf{M} = \begin{pmatrix} -P_{1:3}^{-1} P_4 \\ 1 \end{pmatrix} + \zeta \begin{pmatrix} P_{1:3}^{-1} \mathbf{m} \\ 0 \end{pmatrix}, \quad \zeta \in \mathbb{R}. \tag{12}$$

The parameter $\zeta$ in Eq. (12) represent the the depth of the point $\mathbf{M}$ only if $P$ has been scaled so that $\lambda = 1$ in Eq. (8). [01]

### 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 $\mathbf{M} = (\tilde{\mathbf{M}}^T, 0)^T$, hence

$$\mathbf{m} \simeq K[R \mid \mathbf{t}](\tilde{\mathbf{M}}^T, 0)^T = KR\tilde{\mathbf{M}}. \tag{13}$$

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:

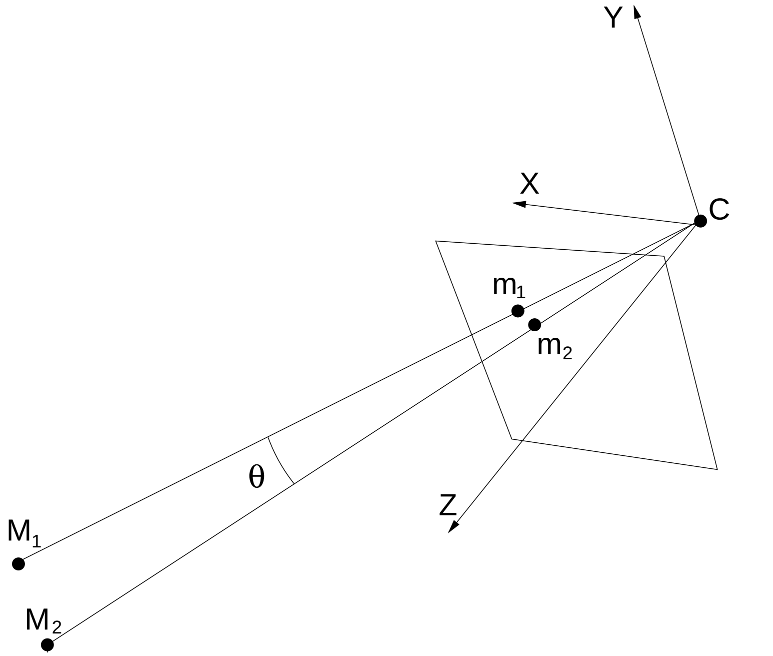
$$\mathbf{m}^T(K^{-T}K^{-1})\mathbf{m} = 0. \tag{14}$$

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. 02

Let us consider a camera $P = [K|\mathbf{0}]$, then $\mathbf{m} = \frac{1}{z}K\tilde{\mathbf{M}}$. Let $\theta$ be the angle between the rays trough $\mathbf{M}_1$ and $\mathbf{M}_1$, then

$$\cos\theta = \frac{\tilde{\mathbf{M}}_1^T\tilde{\mathbf{M}}_2}{||\tilde{\mathbf{M}}_1||||\tilde{\mathbf{M}}_2||} = \frac{\mathbf{m}_1^T\boldsymbol{\omega}\mathbf{m}_2}{\sqrt{\mathbf{m}_1^T\boldsymbol{\omega}\mathbf{m}_1}\sqrt{\mathbf{m}_2^T\boldsymbol{\omega}\mathbf{m}_2}} \tag{15}$$



## 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 [23], we derive:

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

After expanding the coefficient matrix, we obtain

$$\begin{bmatrix} \mathbf{0}^T & -\mathbf{M}_i^T & v_i\mathbf{M}_i^T \\ \mathbf{M}_i^T & \mathbf{0}^T & -u_i\mathbf{M}_i^T \\ -v_i\mathbf{M}_i^T & u_i\mathbf{M}_i^T & \mathbf{0}^T \end{bmatrix} \operatorname{vec} P^T = \mathbf{0} \tag{18}$$

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.

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

This is called the Direct Linear Transform (DLT) algorithm [9].

# 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 [29, 4].

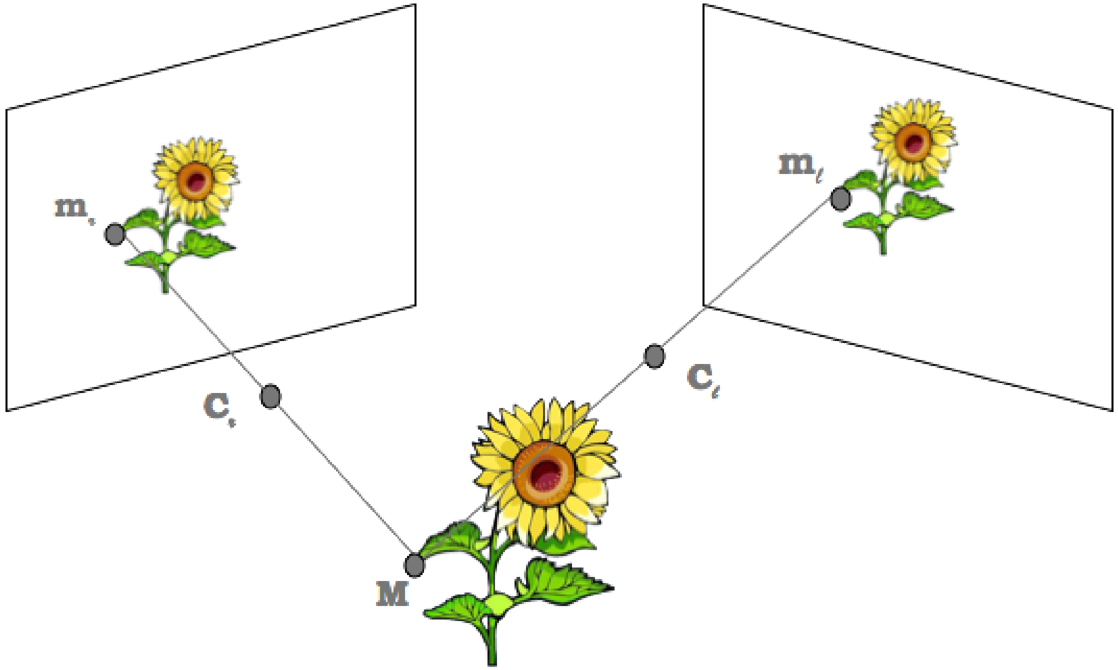


Fig. 3. Two perspective views of the same 3-D scene. $\mathbf{m}_\ell$ and $\mathbf{m}_r$ are corresponding points, as they are the projection of the same 3-D point, $\mathbf{M}$.

We will refer to the camera projection matrix of the left view as $P_\ell$ and of the right view as $P_r$. The 3-D point $\mathbf{M}$ is then imaged as (19) in the left view, and (20) in the right view:

$$\zeta_\ell \mathbf{m}_\ell = P_\ell \mathbf{M} \tag{19}$$

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M}. \tag{20}$$

Geometrically, the position of the image point $\mathbf{m}_\ell$ in the left image plane $I_\ell$ 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 $I_\ell$ 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.

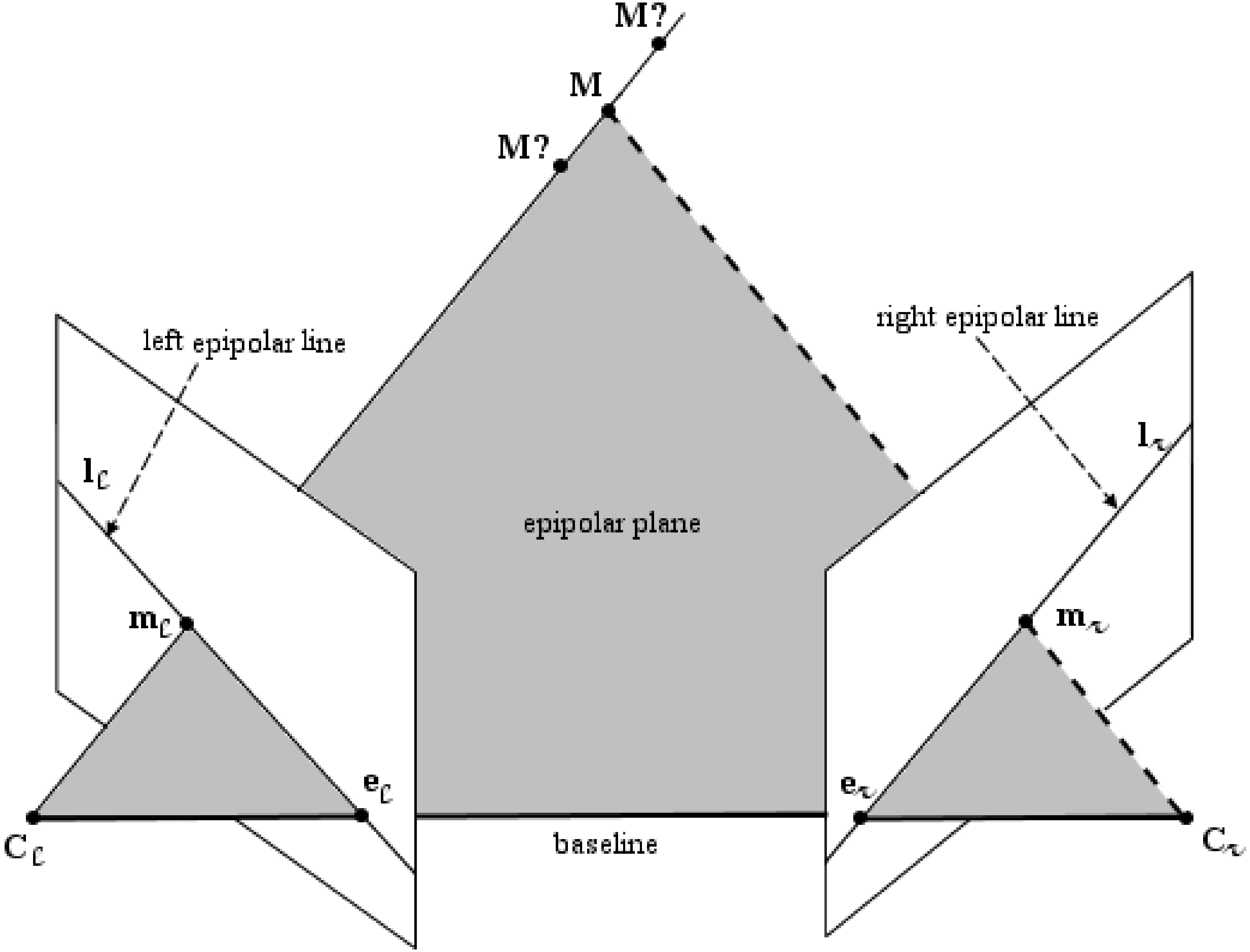


Fig. 4. The epipolar geometry and epipolar constraint.

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 (Equation (3)) of the optical ray through $\mathbf{m}_\ell$ (Equation (12)) onto the right image plane:

$$\zeta_r \mathbf{m}_r = P_r \mathbf{M} = \underbrace{P_r \begin{pmatrix} -P_{\ell_{1:3}}^{-1} P_{\ell_4} \\ 1 \end{pmatrix}}_{\mathbf{e}_r} + \zeta_\ell P_r \begin{pmatrix} P_{\ell_{1:3}}^{-1} \mathbf{m}_\ell \\ 0 \end{pmatrix} \tag{21}$$

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

$$\zeta_r \mathbf{m}_r = \mathbf{e}_r + \zeta_\ell \underbrace{P_{r_{1:3}} P_{\ell_{1:3}}^{-1} \mathbf{m}_\ell}_{\mathbf{m}'_\ell} \tag{22}$$

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.

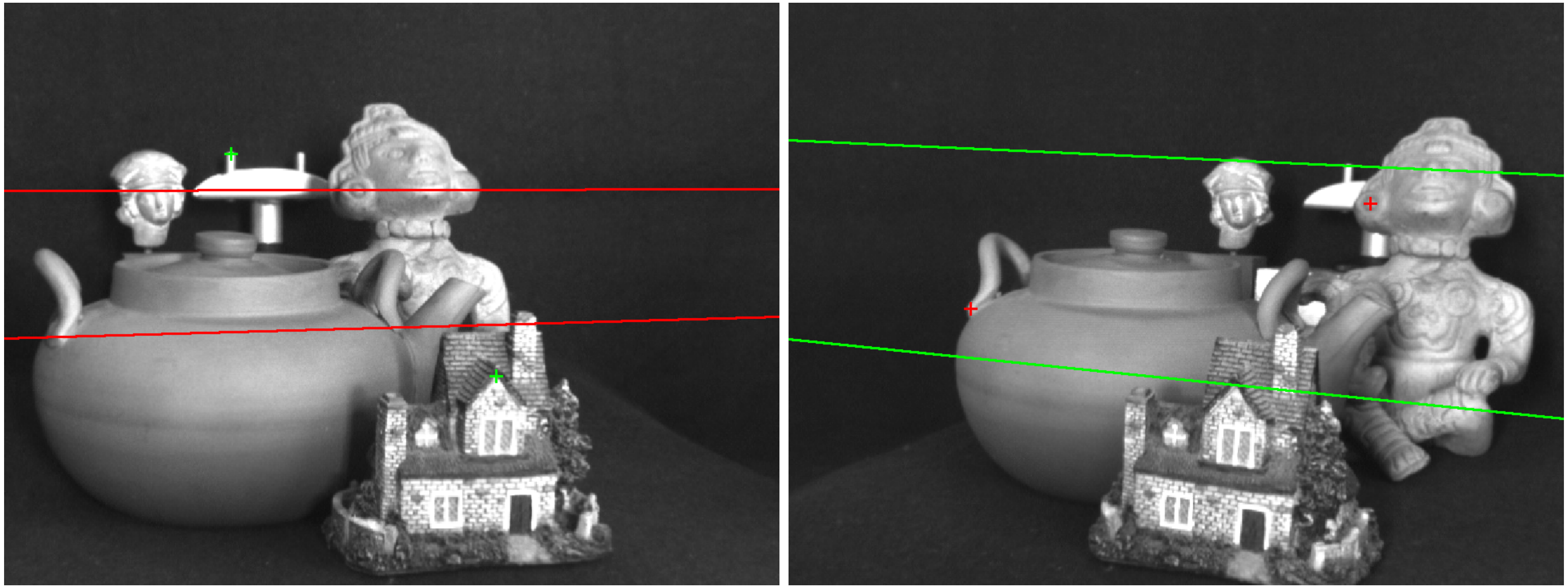


Fig. 5. Left and right images with epipolar lines.

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 (22)).

If only the *intrinsic* parameters are known, we work in normalized camera 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

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 $P_\ell$ and $P_r$. Without loss of generality, we can fix the world reference frame onto the first camera, hence:

$$P_\ell = [I|0] \quad \text{and} \quad P_r = [R|\mathbf{t}]. \tag{23}$$

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 (22), we get

$$\zeta_r \mathbf{m}_r = \mathbf{t} + \zeta_\ell R \mathbf{m}_\ell; \tag{24}$$

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]

$$\mathbf{m}_r^T (\mathbf{t} \times R\mathbf{m}_\ell) = 0, \tag{25}$$

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 (25) can therefore be equivalently written as

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

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

$$E \triangleq [\mathbf{t}]_\times R. \tag{27}$$

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

$$\mathbf{m}_r^T E \mathbf{m}_\ell = 0. \tag{28}$$

$E$ encodes only information on the extrinsic camera parameters. It is singular, since $\det[\mathbf{t}]_\times = 0$. The essential matrix is a homogeneous quantity. It has only five degrees of freedom: a 3-D rotation and a 3-D 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 (23):

$$P_\ell = K_\ell[I|0] \quad \text{and} \quad P_r = K_r[R|\mathbf{t}]. \tag{29}$$

Inserting these two projection matrices into Equation (22), we get

$$\zeta_r \mathbf{m}_r = \mathbf{e}_r + \zeta_\ell K_r R K_\ell^{-1} \mathbf{m}_\ell \quad \text{with} \quad \mathbf{e}_r = K_r \mathbf{t}, \tag{30}$$

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:

$$\mathbf{m}_r^T [\mathbf{e}_r]_\times K_r R K_\ell^{-1} \mathbf{m}_\ell = 0. \tag{31}$$

The matrix

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

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:

$$\mathbf{m}_r^T F \mathbf{m}_\ell = 0. \tag{33}$$

$F$ is the algebraic representation of the epipolar geometry in the least information case. It is a $3 \times 3$, 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

$$\mathbf{l}_r = F \mathbf{m}_\ell. \tag{34}$$

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

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

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:

$$F\mathbf{e}_\ell = 0 \tag{36}$$

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

One can see from the derivation that the essential and fundamental matrices are related through the camera calibration matrices $K_\ell$ and $K_r$:

$$F = K_r^{-T} E K_\ell^{-1}. \tag{38}$$

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 (30) gives: ⑩

$$\zeta_r F^T \mathbf{m}_r = \zeta_\ell(\mathbf{e}_\ell \times \mathbf{m}_\ell). \tag{39}$$

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 (33) to compute the unknown matrix $F$.

We need to convert Equation (33) from its bilinear form to a form that matches the null-space problem. To this end we use again the $\mathrm{vec}$ 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.

## 4.2 Triangulation

Given the camera matrices $P_\ell$ 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 3-D scene point $\mathbf{M}$.

Analytically, the reconstructed 3-D point $\mathbf{M}$ can be found by solving for parameter $\zeta_\ell$ or $\zeta_r$ in Eq. (22). Let us rewrite it as:

$$\mathbf{e}_r = \zeta_r \mathbf{m}_r - \zeta_\ell \mathbf{m}'_\ell \tag{40}$$

The depth $\zeta_r$ and $\zeta_\ell$ are unknown. Both encode the position of $\mathbf{M}$ in space, as $\zeta_r$ is the depth of $\mathbf{M}$ wrt the right camera and $\zeta_\ell$ 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 $\zeta_r$ using the following closed form expressions [28]:

$$\zeta_r = \frac{(\mathbf{e}_r \times \mathbf{m}'_\ell) \cdot (\mathbf{m}_r \times \mathbf{m}'_\ell)}{||\mathbf{m}_r \times \mathbf{m}'_\ell||^2} \tag{41}$$

The reconstructed point $\mathbf{M}$ can then be calculated by inserting the value $\zeta$ into Equation (12).

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 Formula (41) solve Eq. (40) in a least squares sense [18].

Triangulation can be also cast as a null-space problem, starting from the general projection equation (3).[12]

Triangulation is addressed in more details in [2, 11, 9, 36].

## 4.3 3-D 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.

### 4.3.1 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 [15].

**Theorem 4.1** *A real $3 \times 3$ 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 [15]) that describes how $E$ can be factorised into rotation and translation using its Singular Value Decomposition (SVD).
$\textcircled{07}$

The rotation $R$ and translation $\mathbf{t}$ are then used to instantiate a camera pair as in Equation (23), 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 (27) 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*. ⑮

### 4.3.2 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 3-D 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 $P_\ell$ and $P_r$, as well as the 3-D points $\mathbf{M}_i$ such that

$$\mathbf{m}_\ell^i = P_\ell \mathbf{M}^i \quad \text{and} \quad \mathbf{m}_r^i = P_r \mathbf{M}^i, \quad \forall i \tag{42}$$

If $T$ is any $4 \times 4$ invertible matrix, representing a collineation of $\mathbb{P}_3$, then replacing points $\mathbf{M}^i$ by $T\mathbf{M}^i$ and matrices $P_\ell$ 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 4.1.2), and from a (non-unique) factorization of $F$ of the form

$$F = [\mathbf{e}_r]_\times A \tag{43}$$

two camera matrices $P_\ell$ and $P_r$:

$$P_\ell = [I|\mathbf{0}] \quad \text{and} \quad P_r = [A|\mathbf{e}_r], \tag{44}$$

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* [21]). $\textcircled{08}$

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

Indeed, for any $A$ satisfying Equation (43), also $A + \mathbf{e}_r\mathbf{x}^T$ for any vector $\mathbf{x}$, satisfies Equation (43).

# 5 Multiple View Geometry

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

This geometry could be described in terms of fundamental matrices linking pairs of cameras, but a more compact and elegant description is provided by a suitable *trilinear form*, in the same way as the epipolar (bifocal) geometry is described by a bilinear form.

We also discover that three views are all we need, in the sense that additional views do not allow us to compute anything we could not already compute (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 6. The three fundamental matrices describe completely the trifocal geometry [7].

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

$$F_{3,1}\mathbf{e}_{3,2} \simeq \mathbf{e}_{1,3} \times \mathbf{e}_{1,2} \quad F_{1,2}\mathbf{e}_{1,3} \simeq \mathbf{e}_{2,1} \times \mathbf{e}_{2,3} \quad F_{2,3}\mathbf{e}_{2,1} \simeq \mathbf{e}_{3,2} \times \mathbf{e}_{3,1} \tag{45}$$

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.

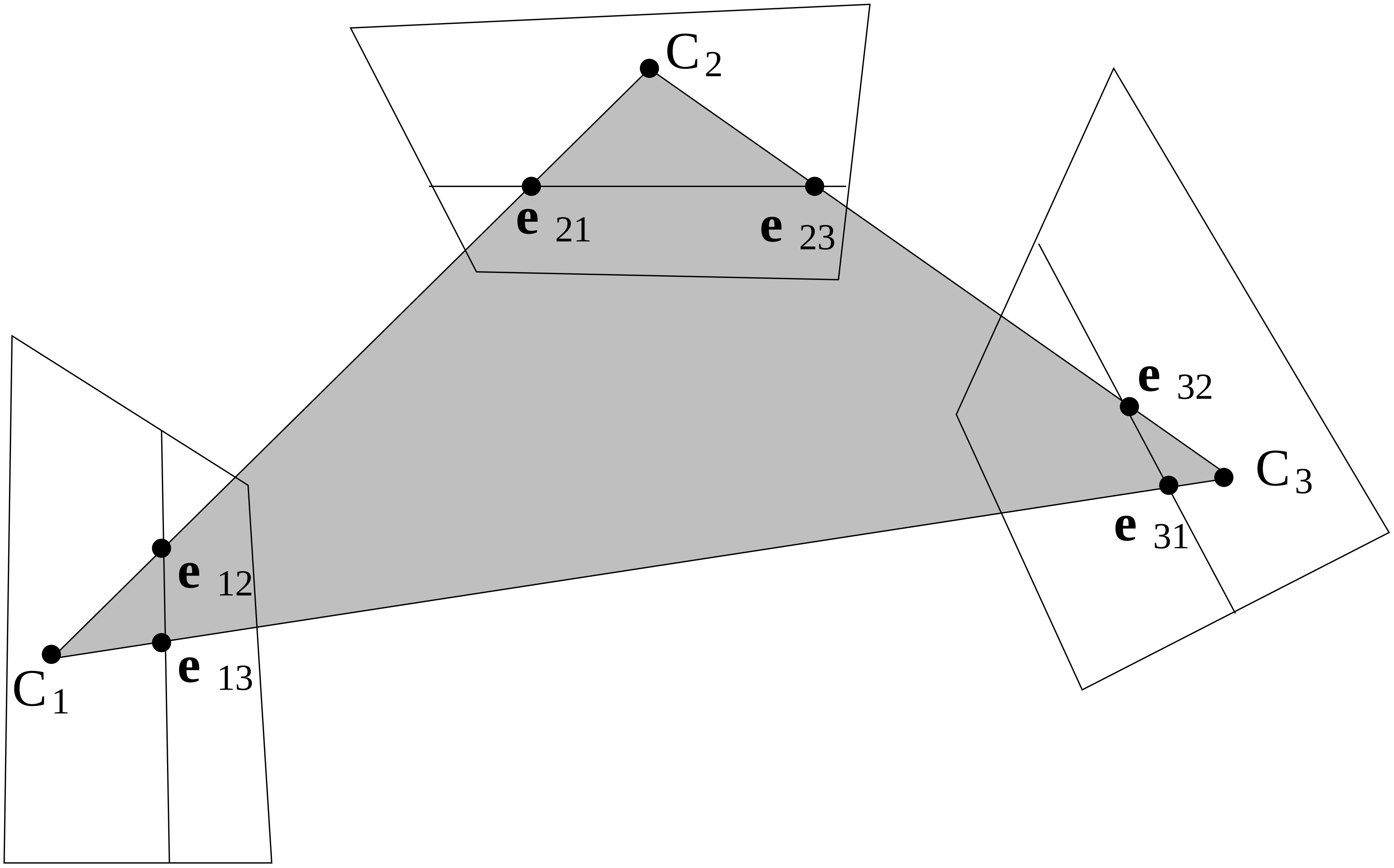


Fig. 6. Trifocal geometry.

## Point transfer

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 7).

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:

$$\mathbf{m}_3 \simeq F_{1,3}\mathbf{m}_1 \times F_{2,3}\mathbf{m}_2 \tag{46}$$

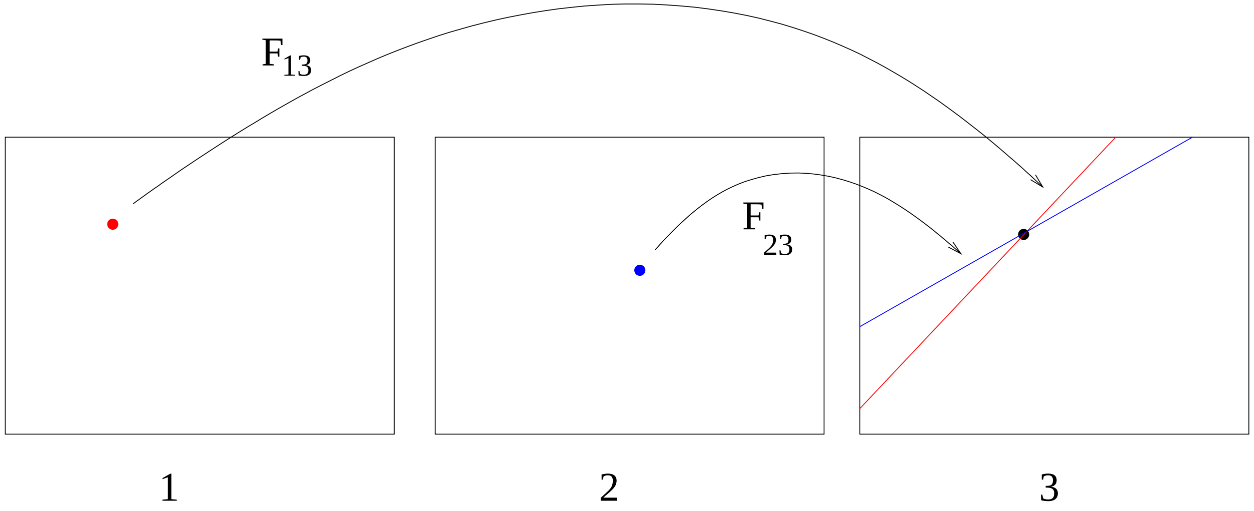


Fig. 7. Point transfer using epipolar constraints between three views.

*View synthesis* [19, 1, 3], exploit the trifocal geometry to generate novel (synthetic) images starting from two reference views. A related topic is *image-based rendering* [20, 37, 16].

Epipolar transfer fails when the three optical rays are coplanar, because the epipolar lines are coincident. This happens:

- if the 3-D point is on the trifocal plane;
- if the three cameras centres are collinear (independently of the position of 3-D point).

These deficiencies motivate the introduction of an independent trifocal constraint.

In addition, by generalizing the case of two views, one might conjecture that the trifocal geometry should be represented by a trilinear form in the coordinates of three conjugate points.

## 5.2 The trifocal constraint

Consider a point $\mathbf{M}$ in space projecting to $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$ in three cameras

$$P_1 = [I|0], \quad P_2 = [A_2|\mathbf{e}_{2,1}], \quad \text{and} \quad P_3 = [A_3|\mathbf{e}_{3,1}]. \tag{47}$$

Let us write the epipolar line of $\mathbf{m}_1$ in the other two views (using Equation (22)):

$$\zeta_2\mathbf{m}_2 = \mathbf{e}_{2,1} + \zeta_1 A_2\mathbf{m}_1 \tag{48}$$

$$\zeta_3\mathbf{m}_3 = \mathbf{e}_{3,1} + \zeta_1 A_3\mathbf{m}_1. \tag{49}$$

Consider a line through $\mathbf{m}_2$, represented by $\mathbf{s}_2$; we have $\mathbf{s}_2^T\mathbf{m}_2 = 0$, that substituted in (48) gives:

$$0 = \mathbf{s}_2^T\mathbf{e}_{2,1} + \zeta_1\mathbf{s}_2^T A_2\mathbf{m}_1 \tag{50}$$

Likewise, for a line through $\mathbf{m}_3$ represented by $\mathbf{s}_3$ we can write:

$$0 = \mathbf{s}_3^T\mathbf{e}_{3,1} + \zeta_1\mathbf{s}_3^T A_3\mathbf{m}_1 \tag{51}$$

After eliminating $\zeta_1$ from Equation (50) and (51) we obtain:

$$0 = (\mathbf{s}_2^T \mathbf{e}_{2,1})(\mathbf{s}_3^T A_3 \mathbf{m}_1) - (\mathbf{s}_3^T \mathbf{e}_{3,1})(\mathbf{s}_2^T A_2 \mathbf{m}_1) \tag{52}$$

and after some re-writing:

$$0 = \mathbf{s}_2^T \left( \mathbf{e}_{2,1} \mathbf{m}_1^T A_3^T - A_2 \mathbf{m}_1 \mathbf{e}_{3,1}^T \right) \mathbf{s}_3 \tag{53}$$

This is the *fundamental trifocal constraint*, that links (via a trilinear form) $\mathbf{m}_1$, $\mathbf{s}_2$ (any line through $\mathbf{m}_2$) and $\mathbf{s}_3$ (any line through $\mathbf{m}_3$).

Geometrically, the trifocal constraint imposes that the optical rays of $\mathbf{m}_1$ intersect the 3-D line $L$ that projects onto $\mathbf{s}_2$ in the second image and $\mathbf{s}_3$ in the third image.

Please note that given two (arbitrary) lines in two images, they can be always seen as the image of a 3-D line $L$, because two planes always define a line, in projective space (this is why there is no such thing as the epipolar constraint between lines.)

The trifocal constraint represents the trifocal geometry (nearly) without singularities. It only fails is when the cameras are collinear *and* the 3-D point is on the same line.

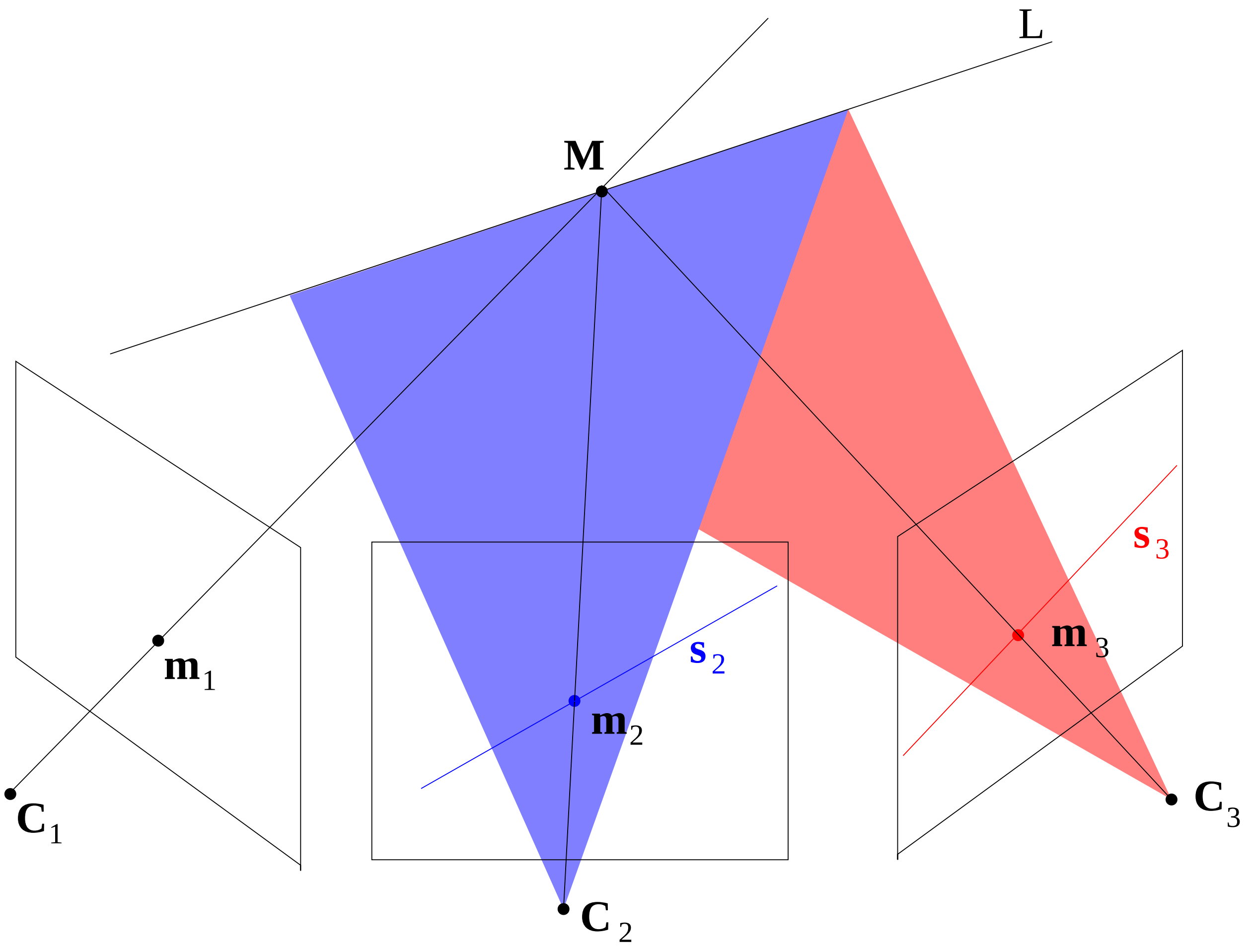


Fig. 8. Two arbitrary lines $\mathbf{s}_2$ and $\mathbf{s}_3$ through corresponding points $\mathbf{m}_2$ and $\mathbf{m}_3$ in the second and third image respectively, define a 3-D line $L$ that must intersect the optical ray of $\mathbf{m}_1$.

Using the properties of the Kronecker product (Cfr. pg. 38), the fundamental trifocal constraint (Eq. (53)) can be written as:

$$0 = (\mathbf{s}_3^T \otimes \mathbf{s}_2^T) T \mathbf{m}_1 = (\mathbf{m}_1^T \otimes \mathbf{s}_3^T \otimes \mathbf{s}_2^T) \operatorname{vec}(T) = \mathbf{s}_2^T (T\mathbf{m}_1)^{(3)} \mathbf{s}_3 \tag{54}$$

where $T$ is the $9 \times 3$ matrix defined by

$$T = (A_3 \otimes \mathbf{e}_{2,1}) - (\mathbf{e}_{3,1} \otimes A_2)$$

and $(T\mathbf{m}_1)^{(3)}$ is a $3 \times 3$ matrix such that $\operatorname{vec}(T\mathbf{m}_1)^{(3)} = T\mathbf{m}_1$. The *vector transposition* operator $A^{(p)}$generalizes the transpose of a matrix $A$ by operating on vectors of $p$ entries at a time (see [25]).

The matrix $T$ represents the trilinear form, in the sense that contains its 27 coefficient. It also encodes the trifocal geometry, hence we[1] call it the *trifocal matrix*.

Each of the three equations (54) is an equivalent formulation of the fundamental trifocal constraint.

[1] This is an alternative approach to trifocal geometry, so please be aware that there is no such thing as the trifocal matrix in the literature.

### 5.2.1 Trifocal constraints for points.

Since a point is determined by two lines, we can write a similar independent constraint for a second line. Such two lines determining a point $\mathbf{m}$ are represented by any two rows of $[\mathbf{m}]_\times$. To keep notation compact let us consider the whole matrix $[\mathbf{m}]_\times$. The trifocal constraints for three points writes:

$$[\mathbf{m}_2]_\times (T\mathbf{m}_1)^{(3)} [\mathbf{m}_3]_\times = 0_{3\times 3}. \tag{55}$$

This is a matrix equation which gives 9 scalar equations, only four of which are independent. Equivalently

$$(\mathbf{m}_1^T \otimes [\mathbf{m}_3]_\times \otimes [\mathbf{m}_2]_\times)\, \mathrm{vec}(T) = \mathbf{0} \tag{56}$$

This equation can be used to recover $T$ (likewise we did for $F$). The coefficient matrix is a $9 \times 27$ matrix of rank 4 (the rank of the Kronecker product is the product of the ranks), therefore every triplet $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ of corresponding points gives four linear independent equations. Seven triplets determine the 27 entries of $T$.

## Point transfer.

Let $\mathbf{s}_2^T$ be a row of $[\mathbf{m}_2]_\times$, then

$$\left(\mathbf{s}_2^T (T\mathbf{m}_1)^{(3)}\right) [\mathbf{m}_3]_\times = 0 \tag{57}$$

This implies that the transpose of the leftmost term in parentheses (which is a 3-D vector) belongs to the kernel of $[\mathbf{m}_3]_\times$, which is equal to $\mathbf{m}_3$ (up to a scale factor) by construction. Hence

$$\mathbf{m}_3 \simeq (T\mathbf{m}_1)^{(3)^T} \mathbf{s}_2 \tag{58}$$

This is the point transfer equation: if $\mathbf{m}_1$ and $\mathbf{m}_2$ are conjugate points in the first and second view respectively, the position of the conjugate point $\mathbf{m}_3$ in the third view is computed by means of the trifocal matrix.

### 5.2.2 Trifocal constraint for lines.

Consider a line $\mathbf{L}$ in space projecting to $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$ in the three cameras. The fundamental trifocal constraint must hold for any point $\mathbf{m}_1$ contained in the line $\mathbf{s}_1$:

$$(\mathbf{s}_3^T \otimes \mathbf{s}_2^T) T \mathbf{m}_1 = 0 \quad \forall \mathbf{m}_1 : \mathbf{s}_1^T \mathbf{m}_1 = 0$$

hence

$$\mathbf{s}_1^T = (\mathbf{s}_3^T \otimes \mathbf{s}_2^T) T \tag{59}$$

This is the trifocal constraint for lines, which also allows directly line transfer.

## 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.

If the intrinsic 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.

In both cases, the solution is not a straightforward generalization of the two view case, as the problem of *global* consistency comes into play (i.e., how to relate each other the *local* reconstructions that can be obtained from view pairs).

### 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 4.3.1 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, (the symbol $\hat{}$ indicates a unit vector).

The “true” displacements must satisfy the following compositional rule

$$\mathbf{t}_{13} = R_{23}\mathbf{t}_{12} + \mathbf{t}_{23} \tag{60}$$

which can be rewritten as

$$\hat{\mathbf{t}}_{13} = \mu_1 R_{23}\hat{\mathbf{t}}_{12} + \mu_2\hat{\mathbf{t}}_{23} \tag{61}$$

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

However, Eq. (60) constraints $\hat{\mathbf{t}}_{13}, R_{23}\hat{\mathbf{t}}_{12}$ and $\hat{\mathbf{t}}_{23}$ to be coplanar, hence the ratios $\mu_1, \mu_2$ can be recovered:

$$\frac{||\mathbf{t}_{12}||}{||\mathbf{t}_{13}||} = \mu_1 = \frac{(\hat{\mathbf{t}}_{13} \times \hat{\mathbf{t}}_{23}) \cdot (R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23})}{||R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23}||^2} \tag{62}$$

And similarly for $\mu_2$.

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

If one applies the method of Section 4.3.2 to consecutive pairs of views, she would obtain, in general, a set of reconstructions linked to each other by an unknown projective transformation (because each camera pair defines its own projective frame).

The trifocal geometry could be used to link together consistently triplets of views. In Section 4.3.2 we saw how a camera pair can be extracted from the fundamental matrix. Likewise, a triplet of consistent cameras can extracted from the trifocal matrix (or tensor). The procedure is more tricky, though.

An elegant method for multi-image reconstruction was described in [32], based on the same idea of the factorization method [33].

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

$$\zeta_i^j \mathbf{m}_i^j = P_i \mathbf{M}^j \quad i = 1 \dots m, \quad j = 1 \dots n. \tag{63}$$

can be written in matrix form:

$$\underbrace{\begin{bmatrix} \zeta_1^1 \mathbf{m}_1^1 & \zeta_1^2 \mathbf{m}_1^2 & \dots & \zeta_1^n \mathbf{m}_1^n \\ \zeta_2^1 \mathbf{m}_2^1 & \zeta_2^2 \mathbf{m}_2^2 & \dots & \zeta_2^n \mathbf{m}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_m^1 \mathbf{m}_m^1 & \zeta_m^2 \mathbf{m}_m^2 & \dots & \zeta_m^n \mathbf{m}_m^n \end{bmatrix}}_{\text{scaled measurements } W} = \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}}_{P} \underbrace{\left[\mathbf{M}^1, \mathbf{M}^2, \dots \mathbf{M}^n\right]}_{\text{structure } M}. \tag{64}$$

In this formula the $\mathbf{m}_i^j$ are known, but all the other quantities are unknown, including the projective depths $\zeta_i^j$. Equation (64) tells us that $W$ can be factored into the product of a $3m \times 4$ matrix $P$ and a $4 \times n$ matrix $M$. This also means that $W$ has rank four.

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

$$W = UDV^T. \tag{65}$$

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 of $U$ ($V$) contribute to this matrix product. Let $U_{3m\times 4}$ ($V_{n\times 4}$) the matrix of the first 4 columns of $U$ ($V$). Then:

$$W = U_{3m\times 4}\,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)\,V_{n\times 4}^T. \tag{66}$$

The sought reconstruction is obtained by setting:

$$P = U_{3m\times 4}\,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4) \quad \text{and} \quad M = V_{n\times 4}^T \tag{67}$$

This reconstruction is unique up to a (unknown) projective transformation. Indeed, for any non singular projective transformation $T$, $TP$ and $T^{-1}M$ is an equally valid factorization of the data into projective motion and structure.

Consistently, the choice to subsume $\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ in $P$ is arbitrary.

In presence of noise, $\sigma_5$ will not be zero. By forcing $D = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, 0, \dots 0)$ one computes the solution that minimizes the following error:

$$||W - PM||_F^2 = \sum_{i,j} ||\zeta_i^j \mathbf{m}_i^j - P_i \mathbf{M}^j||^2$$

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

As the depth $\zeta_i^j$ are unknown, we are left with the problem of estimating them.

An iterative solution is to alternate estimating $\zeta_i^j$ (given $P$ and $M$) with estimating $P$ and $M$ (given$\zeta_i^j$).

If $P$ and $M$ are known, estimating $\zeta_i^j$ is a linear problem. Indeed, for a given point $j$ the projection equation writes:

$$\begin{bmatrix} \zeta_1^j \mathbf{m}_1^j \\ \zeta_2^j \mathbf{m}_2^j \\ \vdots \\ \zeta_m^j \mathbf{m}_m^j \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{m}_1^j & 0 & \dots & 0 \\ 0 & \mathbf{m}_2^j & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{m}_m^j \end{bmatrix}}_{Q^j} \underbrace{\begin{bmatrix} \zeta_1^j \\ \zeta_2^j \\ \vdots \\ \zeta_m^j \end{bmatrix}}_{\boldsymbol{\zeta}^j} = PM^j \tag{68}$$

Starting from an initial guess for $\zeta_i^j$ (typically $\zeta_i^j = 1$), the following iterative procedure[2] is used:

1. Normalize $W$ such that $||W|||_F = 1$;
2. Factorize $W$ and obtain an estimate of $P$ and $M$;
3. If $||W - PM||_F^2$ is sufficiently small then stop;
4. Solve for $\boldsymbol{\zeta}^j$ in $Q^j \boldsymbol{\zeta}^j = PM^j$, for all $j = 1 \dots n$;
5. Update $W$.
6. Goto 1.

Step 1 is necessary to avoid trivial solutions (e.g. $\zeta_i^j = 0$).

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 has been presented in [24].

[2] Whilst this procedure captures the main idea of Sturm and Triggs, it is not exactly the algorithm proposed in [32]. To start with, the original algorithm [32] was not iterative and used the epipolar constraint (Eq.39) to fix the ratio of the projective depths of one point in successive images. It was [34] who made the scheme iterative. Moreover in [32] the normalization of $W$ is performed by normalizing rows and columns of $W$. The Frobenius norm was used by [27]. A similar scheme was also proposed by [12].

## 5.4 Multifocal constraints

We outline here an alternative and elegant way to derive all the meaningful multilinear constraints on $N$ views, based on determinants, described in [13]. Consider one image point viewed by $m$ cameras:

$$\zeta_i \mathbf{m}_i = P_i \mathbf{M} \quad i = 1 \dots m \tag{69}$$

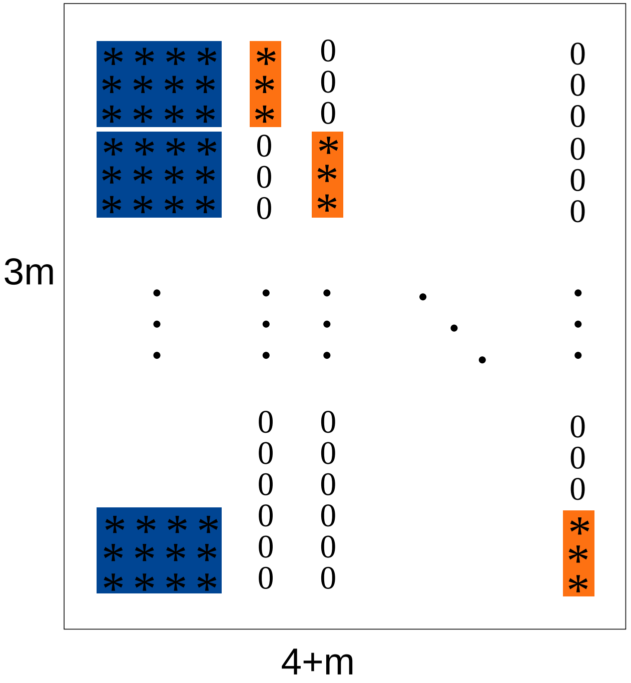
By stacking all these equations we obtain:

$$\underbrace{\begin{bmatrix} P_1 & \mathbf{m}_1 & 0 & \dots & 0 \\ P_2 & 0 & \mathbf{m}_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_m & 0 & 0 & \dots & \mathbf{m}_m \end{bmatrix}}_{L} \begin{bmatrix} \mathbf{M} \\ -\zeta_1 \\ -\zeta_2 \\ \vdots \\ -\zeta_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{70}$$

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

There are three different types of such minors that translates into meaningful multiview constraints, depending on the number of rows taken from each view.

The minors that does not contain at least one row from each camera are identically zero, since they contain a zero column.
Hence, one row has to be taken from each view. Since there are $m$ views, the remaining four rows can be chosen as follows:



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.

If $m > 4$, the minors will contain only one row from some views, and the image coordinate corresponding to this row can be factored out (using Laplace expansion).

In general, constraints involving more than 4 cameras can be factorised as product of the two-, three-, or four-views constraints and image point coordinates.

This indicates that no interesting constraints can be written for more than four views[3].

[3] Actually, it can be proven that also the quadrifocal constraints are not independent [22].

# 6 Dealing with noise and mismatches

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

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

Second, we will discuss the issue of algebraic vs geometric error in estimation.

Finally, we will deal with data corrupted by mismatches, or outliers.

## 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 [10] 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” [9].

### 6.2.1 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 $\hat{P}_i \mathbf{M}^j$:

$$\min_{\hat{P}} \sum_j d(\hat{P}\mathbf{M}^j, \mathbf{m}^j)^2 \tag{71}$$

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)

### 6.2.2 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 $i = 1 \dots m$.

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 $P_i\hat{\mathbf{M}}$:

$$\min_{\hat{\mathbf{M}}} \sum_i d(P_i\hat{\mathbf{M}}, \mathbf{m}_i)^2 \tag{72}$$

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).

### 6.2.3 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):

$$\min_{\hat{F}} \sum_j d(\hat{F}\mathbf{m}_\ell^j, \mathbf{m}_r^j)^2 + d(\hat{F}^T\mathbf{m}_r^j, \mathbf{m}_\ell^j)^2 \tag{73}$$

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 parametrized, as it has only seven d.o.f. ⑪

### 6.2.4 Bundle adjustment (reconstruction)

If measurements are noisy, the projection equation will not be satisfied exactly by the camera matrices and structure computed in Sec. 5.3.2.

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:

$$\min_{\hat{P}_i,\hat{\mathbf{M}}^j} \sum_{i,j} d(\hat{P}_i\hat{\mathbf{M}}^j, \mathbf{m}_i^j)^2 \tag{74}$$

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

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

A solution is to alternate minimizing the re-projection error by varying $\hat{P}_i$ with minimizing the re-projection error by varying $\hat{\mathbf{M}}^j$.

See [35] 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., rogues 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. 9.)

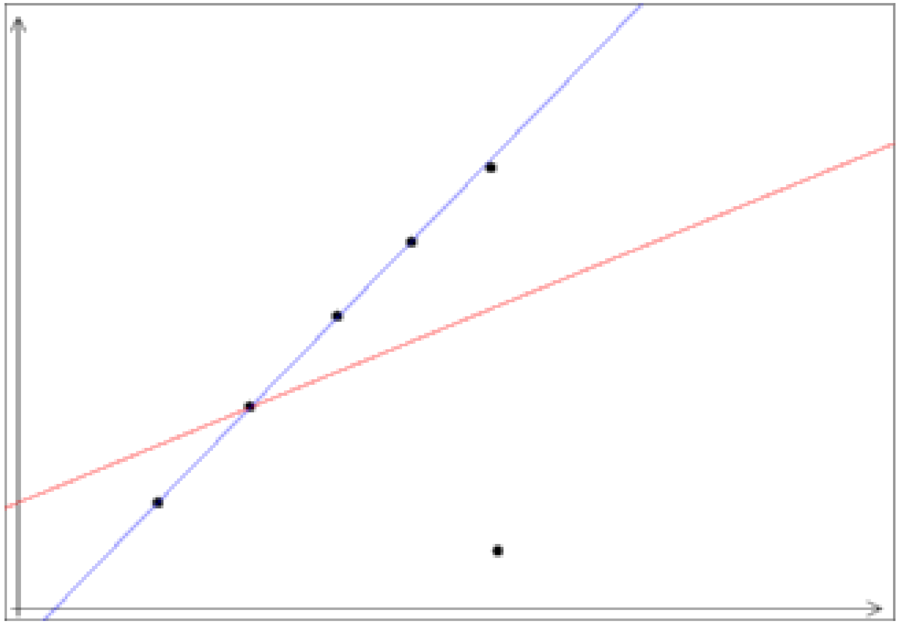


Fig. 9. A single outlier can severely offset the least-squares estimate (red line), whereas the robust estimate (blue line) is unaffected.

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

### 6.3.1 M-estimators

Least squares:

$$\min_\theta \sum_i (r_i/\sigma_i)^2 \tag{75}$$

where $\theta$ 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

$$\min_\theta \sum_i \rho(r_i/\sigma_i) \tag{76}$$

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

Differentiating with respect to $\theta$ yields:

$$\sum_i \frac{1}{\sigma_i}\rho'(r_i/\sigma_i)\frac{dr_i}{d\theta} = 0 \tag{77}$$

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

### 6.3.2 RANSAC

Given a model that requires a minimum of $p$ data points to instantiate its free parameters $\theta$, 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 $\epsilon$.

$N$ 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:

$$P = 1 - (1 - (1 - \epsilon)^p)^N \tag{78}$$

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

$T$ 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 [9].

As pointed out in [31], 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 $t$.

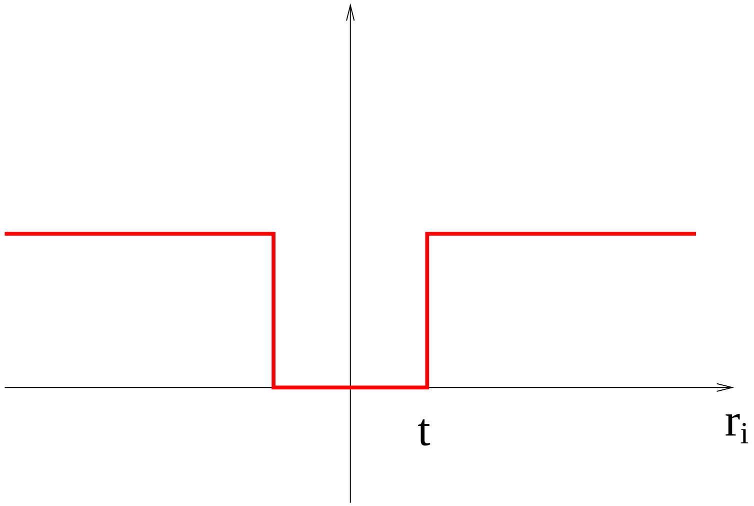


Fig. 10. RANSAC loss function

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

### 6.3.3 LMedS

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

$$\min_{\theta} \mathrm{med}_i r_i \tag{79}$$

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 Further readings

General books on (Geometric) Computer Vision are: [5, 36, 6, 9].

## Acknowledgements

The backbone of this tutorial has been adapted from [17]. Many ideas are taken from [9]. Sec. 5.4 is based on [14].

Michela Farenzena, Spela Ivekovic, Alessandro Negrente, Sara Ceglie, Alberto Valinetti and Roberto Marzotto produced some of the pictures.

# References

[1] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1034–1040, 1997.

[2] 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.

[3] B. S. Boufama. The use of homographies for view synthesis. In *Proceedings of the International Conference on Pattern Recognition*, pages 563–566, 2000.

[4] 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.

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

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

[7] O. D. Faugeras and L. Robert. What can two images tell us about a third one? In *Proceedings of the European Conference on Computer Vision*, pages 485–492, Stockholm, 1994.

[8] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition, 2003.

[10] R. I. Hartley. In defence of the 8-point algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, 1995.

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

[12] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Scandinavian Conference on Image Analysis*, 1997.

[13] A. Heyden. A common framework for multiple-view tensors. In *Proceedings of the European Conference on Computer Vision*, Freiburg, Germany,, 1998.

[14] A. Heyden. Tutorial on multiple view geometry. In conjunction with ICPR00, September 2000.

[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] F. Isgrò, E. Trucco, P. Kauff, and O. Schreer. 3-D image processing in the future of immersive media. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3):288–303, 2004.

[17] S. Ivekovic, A. Fusiello, and E. Trucco. Fundamentals of multiple view geometry. In O. Schreer, P. Kauff, and T. Sikora, editors, *3D Videocommunication. Algorithms, concepts and real-time systems in human centered communication*, chapter 6. John Wiley & Sons, 2005. ISBN: 0-470-02271-X.

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

[19] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images and foundamental matrices. Technical Report 2205, INRIA, Institut National de Recherche en Informatique et an Automatique, February 1994.

[20] Jed Lengyel. The convergence of graphics and vision. *IEEE Computer*, 31(7):46–53, July 1998.

[21] 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.

[22] Yi Ma, Stefano Soatto, Jana Kosecka, and Shankar S. Sastry. *An Invitation to 3-D Vision*. Springer, November 2003.

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

[24] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:1018–1025, 2001.

[25] Thomas Minka. Old and new matrix algebra useful for statistics. MIT Media Lab note.

[26] Theo Moons. A guided tour through multiview relations. In *SMILE*, pages 304–346, 1998.

[27] J. Oliensis. Fast and accurate self-calibration. In *Proceedings of the IEEE International Conference on Computer Vision*, 1999.

[28] L. Robert, C. Zeller, O. Faugeras, and M. Hébert. Applications of non-metric vision to some visually-guided robotics tasks. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter 5, pages 89–134. Lawrence Erlbaum Associates, 1997.

[29] 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.

[30] A. Shashua. Trilinear tensor: The fundamental construct of multiple-view geometry and its applications. In *International Workshop on Algebraic Frames For The Perception Action Cycle (AFPAC)*, Kiel Germany, Sep. 8-9 1997.

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

[32] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the European Conference on Computer Vision*, pages 709–720, Cambridge, UK, 1996.

[33] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography – a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[34] B. Triggs. Factorization methods for projective structure from motion. In *CVPR*, pages 845–851, 1996.

[35] 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.

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

[37] Cha Zhang and Tsuhan Chen. A survey on image-based rendering - representation, sampling and compression. Technical Report AMP 03-03, Electrical and Computer Engineering - Carnegie Mellon University, Pittsburgh, PA 15213, June 2003.

[38] A. Zisserman. Single view and two-view geometry. Handout, EPSRC Summer School on Computer Vision, 1998. available from http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/EPSRC_SSAZ/epsrc_ssaz.html.