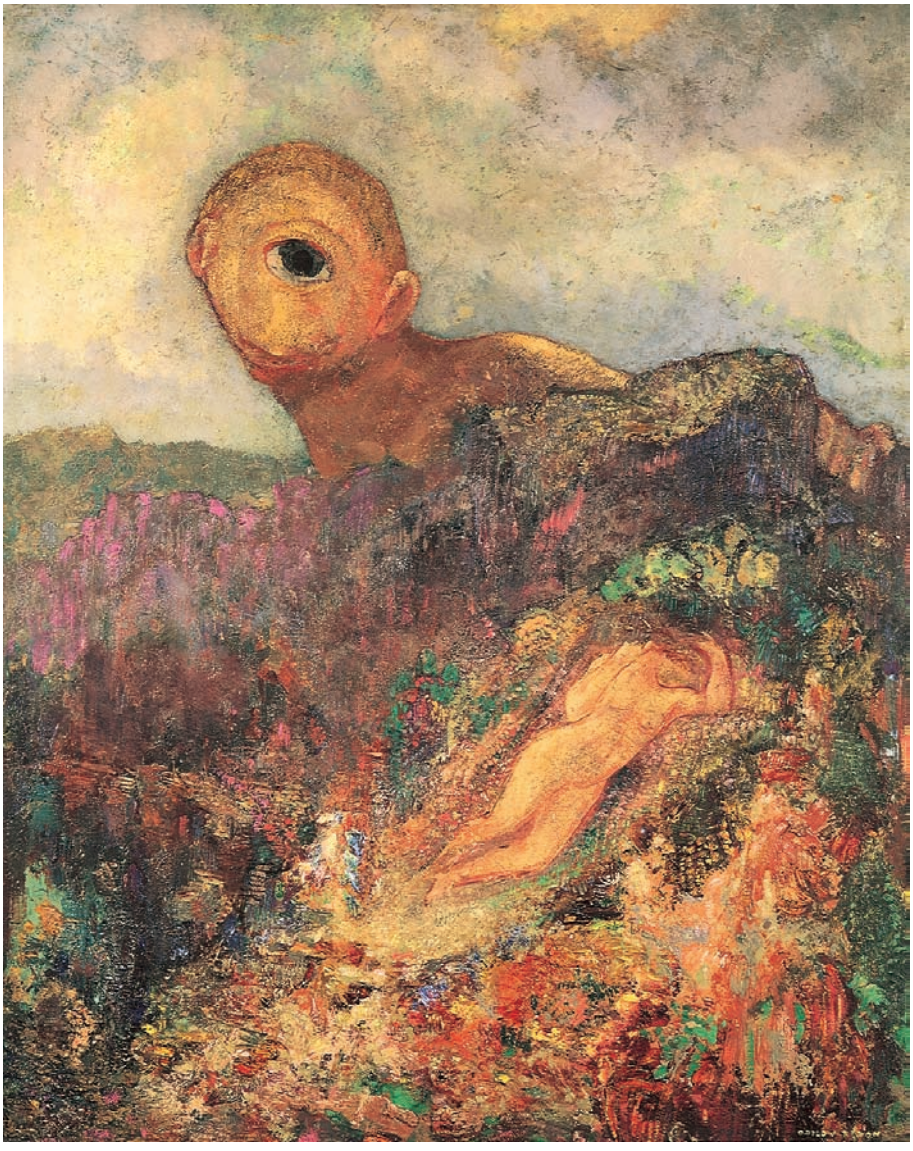
Part I

Camera Geometry and Single View Geometry



The Cyclops, c. 1914 (oil on canvas) by Odilon Redon (1840-1916)
Rijksmuseum Kroller-Muller, Otterlo, Netherlands /Bridgeman Art Library

Outline

This part of the book concentrates on the geometry of a single perspective camera. It contains three chapters.

Chapter 6 describes the projection of 3D scene space onto a 2D image plane. The camera mapping is represented by a matrix, and in the case of mapping points it is a 3×4 matrix $\mathtt{P}$ which maps from homogeneous coordinates of a world point in 3-space to homogeneous coordinates of the imaged point on the image plane. This matrix has in general 11 degrees of freedom, and the properties of the camera, such as its centre and focal length, may be extracted from it. In particular the internal camera parameters, such as the focal length and aspect ratio, are packaged in a 3×3 matrix $\mathtt{K}$ which is obtained from $\mathtt{P}$ by a simple decomposition. There are two particularly important classes of camera matrix: finite cameras, and cameras with their centre at infinity such as the *affine camera* which represents parallel projection.

Chapter 7 describes the estimation of the camera matrix $\mathtt{P}$, given the coordinates of a set of corresponding world and image points. The chapter also describes how constraints on the camera may be efficiently incorporated into the estimation, and a method of correcting for radial lens distortion.

Chapter 8 has three main topics. First, it covers the action of a camera on geometric objects other than finite points. These include lines, conics, quadrics and points at infinity. The image of points/lines at infinity are vanishing points/lines. The second topic is camera calibration, in which the internal parameters $\mathtt{K}$ of the camera matrix are computed, without computing the entire matrix $\mathtt{P}$. In particular the relation of the internal parameters to the image of the absolute conic is described, and the calibration of a camera from vanishing points and vanishing lines. The final topic is the *calibrating conic*. This is a simple geometric device for visualizing camera calibration.

6

Camera Models

A camera is a mapping between the 3D world (object space) and a 2D image. The principal camera of interest in this book is *central projection*. This chapter develops a number of camera *models* which are matrices with particular properties that represent the camera mapping.

It will be seen that all cameras modelling central projection are specializations of the *general projective camera*. The anatomy of this most general camera model is examined using the tools of projective geometry. It will be seen that geometric entities of the camera, such as the projection centre and image plane, can be computed quite simply from its matrix representation. Specializations of the general projective camera inherit its properties, for example their geometry is computed using the same algebraic expressions.

The specialized models fall into two major classes – those that model cameras with a finite centre, and those that model cameras with centre "at infinity". Of the cameras at infinity the *affine camera* is of particular importance because it is the natural generalization of parallel projection.

This chapter is principally concerned with the projection of points. The action of a camera on other geometric entities, such as lines, is deferred until chapter 8.

6.1 Finite cameras

In this section we start with the most specialized and simplest camera model, which is the basic pinhole camera, and then progressively generalize this model through a series of gradations.

The models we develop are principally designed for CCD like sensors, but are also applicable to other cameras, for example X-ray images, scanned photographic negatives, scanned photographs from enlarged negatives, etc.

The basic pinhole model. We consider the central projection of points in space onto a plane. Let the centre of projection be the origin of a Euclidean coordinate system, and consider the plane $\mathrm{Z} = f$, which is called the *image plane* or *focal plane*. Under the pinhole camera model, a point in space with coordinates $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\mathsf{T}$ is mapped to the point on the image plane where a line joining the point $\mathbf{X}$ to the centre of projection meets the image plane. This is shown in figure 6.1. By similar triangles, one quickly

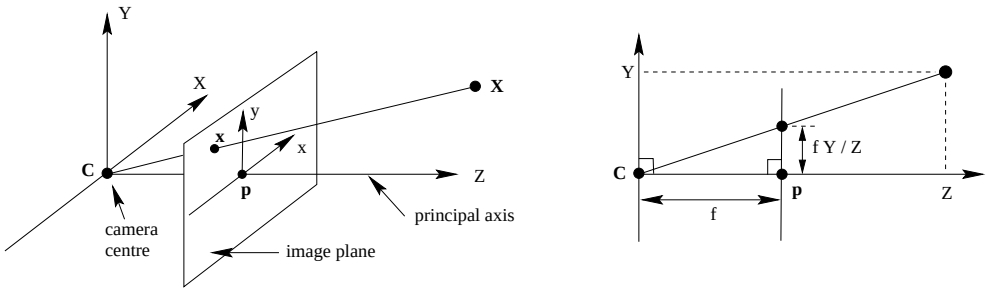


Fig. 6.1. **Pinhole camera geometry.** $\mathbf{C}$ *is the camera centre and* $\mathbf{p}$ *the principal point. The camera centre is here placed at the coordinate origin. Note the image plane is placed in front of the camera centre.*

computes that the point $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\mathsf{T}$ is mapped to the point $(f\mathrm{X}/\mathrm{Z}, f\mathrm{Y}/\mathrm{Z}, f)^\mathsf{T}$ on the image plane. Ignoring the final image coordinate, we see that

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\mathsf{T} \mapsto (f\mathrm{X}/\mathrm{Z}, f\mathrm{Y}/\mathrm{Z})^\mathsf{T} \tag{6.1}$$

describes the central projection mapping from world to image coordinates. This is a mapping from Euclidean 3-space $\mathbb{R}^3$ to Euclidean 2-space $\mathbb{R}^2$.

The centre of projection is called the *camera centre*. It is also known as the *optical centre*. The line from the camera centre perpendicular to the image plane is called the *principal axis* or *principal ray* of the camera, and the point where the principal axis meets the image plane is called the *principal point*. The plane through the camera centre parallel to the image plane is called the *principal plane* of the camera.

Central projection using homogeneous coordinates. If the world and image points are represented by homogeneous vectors, then central projection is very simply expressed as a linear mapping between their homogeneous coordinates. In particular, (6.1) may be written in terms of matrix multiplication as

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} \\ f\mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}. \tag{6.2}$$

The matrix in this expression may be written as $\mathrm{diag}(f, f, 1)[\mathtt{I} \mid \mathbf{0}]$ where $\mathrm{diag}(f, f, 1)$ is a diagonal matrix and $[\mathtt{I} \mid \mathbf{0}]$ represents a matrix divided up into a 3×3 block (the identity matrix) plus a column vector, here the zero vector.

We now introduce the notation $\mathbf{X}$ for the world point represented by the homogeneous 4-vector $(\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)^\mathsf{T}$, $\mathbf{x}$ for the image point represented by a homogeneous 3-vector, and $\mathtt{P}$ for the 3×4 homogeneous *camera projection matrix*. Then (6.2) is written compactly as

$$\mathbf{x} = \mathtt{P}\mathbf{X}$$

which defines the camera matrix for the pinhole model of central projection as

$$\mathtt{P} = \mathrm{diag}(f, f, 1)\,[\mathtt{I} \mid \mathbf{0}].$$

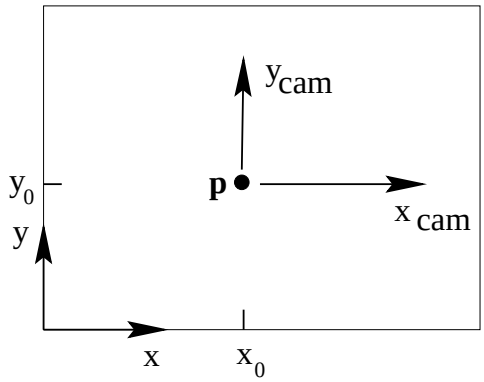


Fig. 6.2. *Image* (x, y) *and camera* $(x_{\text{cam}}, y_{\text{cam}})$ *coordinate systems.*

Principal point offset. The expression (6.1) assumed that the origin of coordinates in the image plane is at the principal point. In practice, it may not be, so that in general there is a mapping

$$(\mathtt{X}, \mathtt{Y}, \mathtt{Z})^\mathsf{T} \mapsto (f\mathtt{X}/\mathtt{Z} + p_x, f\mathtt{Y}/\mathtt{Z} + p_y)^\mathsf{T}$$

where $(p_x, p_y)^\mathsf{T}$ are the coordinates of the principal point. See figure 6.2. This equation may be expressed conveniently in homogeneous coordinates as

$$\begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathtt{X} + \mathtt{Z}p_x \\ f\mathtt{Y} + \mathtt{Z}p_y \\ \mathtt{Z} \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \\ 1 \end{pmatrix}. \tag{6.3}$$

Now, writing

$$\mathtt{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{6.4}$$

then (6.3) has the concise form

$$\mathbf{x} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\text{cam}}. \tag{6.5}$$

The matrix $\mathtt{K}$ is called the *camera calibration matrix*. In (6.5) we have written $(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, 1)^\mathsf{T}$ as $\mathbf{X}_{\text{cam}}$ to emphasize that the camera is assumed to be located at the origin of a Euclidean coordinate system with the principal axis of the camera pointing straight down the $\mathtt{Z}$-axis, and the point $\mathbf{X}_{\text{cam}}$ is expressed in this coordinate system. Such a coordinate system may be called the *camera coordinate frame*.

Camera rotation and translation. In general, points in space will be expressed in terms of a different Euclidean coordinate frame, known as the *world coordinate frame*. The two coordinate frames are related via a rotation and a translation. See figure 6.3. If $\widetilde{\mathbf{X}}$ is an inhomogeneous 3-vector representing the coordinates of a point in the world coordinate frame, and $\widetilde{\mathbf{X}}_{\text{cam}}$ represents the same point in the camera coordinate frame, then we may write $\widetilde{\mathbf{X}}_{\text{cam}} = \mathtt{R}(\widetilde{\mathbf{X}} - \widetilde{\mathbf{C}})$, where $\widetilde{\mathbf{C}}$ represents the coordinates of the camera

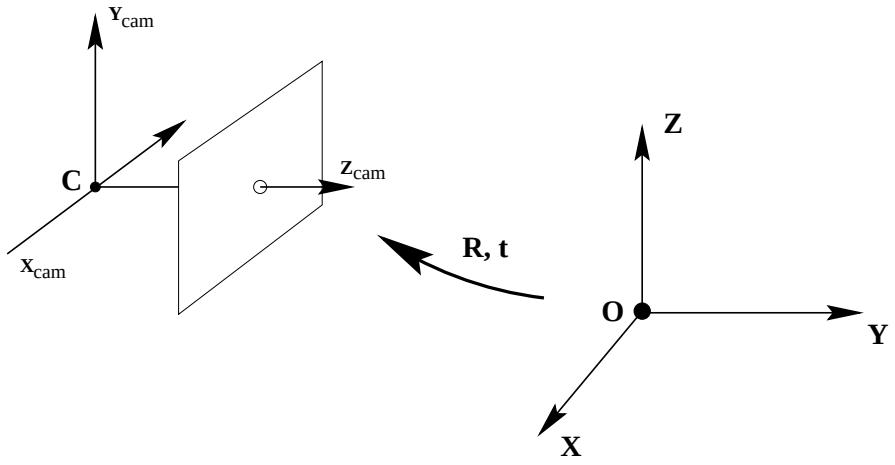


Fig. 6.3. *The Euclidean transformation between the world and camera coordinate frames.*

centre in the world coordinate frame, and $\mathtt{R}$ is a 3×3 rotation matrix representing the orientation of the camera coordinate frame. This equation may be written in homogeneous coordinates as

$$\mathbf{X}_{\text{cam}} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\widetilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \\ 1 \end{pmatrix} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\widetilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}. \tag{6.6}$$

Putting this together with (6.5) leads to the formula

$$\mathbf{x} = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]\mathbf{X} \tag{6.7}$$

where $\mathbf{X}$ is now in a world coordinate frame. This is the general mapping given by a pinhole camera. One sees that a general pinhole camera, $\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]$, has 9 degrees of freedom: 3 for $\mathtt{K}$ (the elements f, p_x, p_y), 3 for $\mathtt{R}$, and 3 for $\widetilde{\mathbf{C}}$. The parameters contained in $\mathtt{K}$ are called the *internal* camera parameters, or the *internal orientation* of the camera. The parameters of $\mathtt{R}$ and $\widetilde{\mathbf{C}}$ which relate the camera orientation and position to a world coordinate system are called the *external* parameters or the *exterior orientation.*

It is often convenient not to make the camera centre explicit, and instead to represent the world to image transformation as $\widetilde{\mathbf{X}}_{\text{cam}} = \mathtt{R}\widetilde{\mathbf{X}} + \mathbf{t}$. In this case the camera matrix is simply

$$\mathtt{P} = \mathtt{K}[\mathtt{R} \mid \mathbf{t}] \tag{6.8}$$

where from (6.7) $\mathbf{t} = -\mathtt{R}\widetilde{\mathbf{C}}$.

CCD cameras. The pinhole camera model just derived assumes that the image coordinates are Euclidean coordinates having equal scales in both axial directions. In the case of CCD cameras, there is the additional possibility of having non-square pixels. If image coordinates are measured in pixels, then this has the extra effect of introducing unequal scale factors in each direction. In particular if the number of pixels per unit

distance in image coordinates are m_x and m_y in the x and y directions, then the transformation from world coordinates to pixel coordinates is obtained by multiplying (6.4) on the left by an extra factor $\mathrm{diag}(m_x, m_y, 1)$. Thus, the general form of the calibration matrix of a CCD camera is

$$\mathtt{K} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{6.9}$$

where $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the focal length of the camera in terms of pixel dimensions in the x and y direction respectively. Similarly, $\tilde{\mathbf{x}}_0 = (x_0, y_0)$ is the principal point in terms of pixel dimensions, with coordinates $x_0 = m_x p_x$ and $y_0 = m_y p_y$. A CCD camera thus has 10 degrees of freedom.

Finite projective camera. For added generality, we can consider a calibration matrix of the form

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}. \tag{6.10}$$

The added parameter s is referred to as the *skew* parameter. The skew parameter will be zero for most normal cameras. However, in certain unusual instances which are described in section 6.2.4, it can take non-zero values.

A camera

$$\mathtt{P} = \mathtt{KR}[\mathtt{I} \mid -\widetilde{\mathbf{C}}] \tag{6.11}$$

for which the calibration matrix K is of the form (6.10) will be called a *finite projective* camera. A finite projective camera has 11 degrees of freedom. This is the same number of degrees of freedom as a 3×4 matrix, defined up to an arbitrary scale.

Note that the left hand 3×3 submatrix of P, equal to KR, is non-singular. Conversely, any 3×4 matrix P for which the left hand 3×3 submatrix is non-singular is the camera matrix of some finite projective camera, because P can be decomposed as $\mathtt{P} = \mathtt{KR}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]$. Indeed, letting M be the left 3×3 submatrix of P, one decomposes M as a product $\mathtt{M} = \mathtt{KR}$ where K is upper-triangular of the form (6.10) and R is a rotation matrix. This decomposition is essentially the RQ matrix decomposition, described in section A4.1.1(*p*579), of which more will be said in section 6.2.4. The matrix P can therefore be written $\mathtt{P} = \mathtt{M}[\mathtt{I} \mid \mathtt{M}^{-1}\mathbf{p}_4] = \mathtt{KR}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]$ where $\mathbf{p}_4$ is the last column of P. In short

- *The set of camera matrices of finite projective cameras is identical with the set of homogeneous* 3×4 *matrices for which the left hand* 3×3 *submatrix is non-singular.*

General projective cameras. The final step in our hierarchy of projective cameras is to remove the non-singularity restriction on the left hand 3×3 submatrix. A *general projective* camera is one represented by an arbitrary homogeneous 3×4 matrix of rank 3. It has 11 degrees of freedom. The rank 3 requirement arises because if the rank is

Camera centre. The camera centre is the 1-dimensional right null-space $\mathbf{C}$ of $\mathtt{P}$, i.e. $\mathtt{P}\mathbf{C} = \mathbf{0}$.

- ◇ **Finite camera** ($\mathtt{M}$ is not singular) $\mathbf{C} = \begin{pmatrix} -\mathtt{M}^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix}$
- ◇ **Camera at infinity** ($\mathtt{M}$ is singular) $\mathbf{C} = \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}$ where $\mathbf{d}$ is the null 3-vector of $\mathtt{M}$, i.e. $\mathtt{M}\mathbf{d} = \mathbf{0}$.

Column points. For $i = 1, \ldots, 3$, the column vectors $\mathbf{p}_i$ are vanishing points in the image corresponding to the X, Y and Z axes respectively. Column $\mathbf{p}_4$ is the image of the coordinate origin.

Principal plane. The principal plane of the camera is $\mathbf{P}^3$, the last row of $\mathtt{P}$.

Axis planes. The planes $\mathbf{P}^1$ and $\mathbf{P}^2$ (the first and second rows of $\mathtt{P}$) represent planes in space through the camera centre, corresponding to points that map to the image lines $x = 0$ and $y = 0$ respectively.

Principal point. The image point $\mathbf{x}_0 = \mathtt{M}\mathbf{m}^3$ is the principal point of the camera, where $\mathbf{m}^{3\mathsf{T}}$ is the third row of $\mathtt{M}$.

Principal ray. The principal ray (axis) of the camera is the ray passing through the camera centre $\mathbf{C}$ with direction vector $\mathbf{m}^{3\mathsf{T}}$. The principal axis vector $\mathbf{v} = \det(\mathtt{M})\mathbf{m}^3$ is directed towards the front of the camera.

Table 6.1. *Summary of the properties of a projective camera* $\mathtt{P}$. *The matrix is represented by the block form* $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$.

less than this then the range of the matrix mapping will be a line or point and not the whole plane; in other words not a 2D image.

6.2 The projective camera

A general projective camera $\mathtt{P}$ maps world points $\mathbf{X}$ to image points $\mathbf{x}$ according to $\mathbf{x} = \mathtt{P}\mathbf{X}$. Building on this mapping we will now dissect the camera model to reveal how geometric entities, such as the camera centre, are encoded. Some of the properties that we consider will apply only to finite projective cameras and their specializations, whilst others will apply to general cameras. The distinction will be evident from the context. The derived properties of the camera are summarized in table 6.1.

6.2.1 Camera anatomy

A general projective camera may be decomposed into blocks according to $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$, where $\mathtt{M}$ is a 3×3 matrix. It will be seen that if $\mathtt{M}$ is non-singular, then this is a finite camera, otherwise it is not.

Camera centre. The matrix $\mathtt{P}$ has a 1-dimensional right null-space because its rank is 3, whereas it has 4 columns. Suppose the null-space is generated by the 4-vector $\mathbf{C}$, that is $\mathtt{P}\mathbf{C} = \mathbf{0}$. It will now be shown that $\mathbf{C}$ is the camera centre, represented as a homogeneous 4-vector.

Consider the line containing $\mathbf{C}$ and any other point $\mathbf{A}$ in 3-space. Points on this line may be represented by the join

$$\mathbf{X}(\lambda) = \lambda\mathbf{A} + (1 - \lambda)\mathbf{C} \ .$$

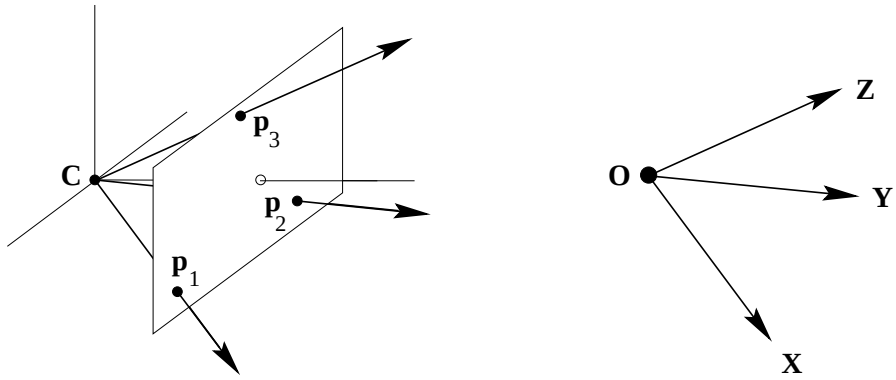


Fig. 6.4. *The three image points defined by the columns* $\mathbf{p}_i, i = 1, \ldots, 3$*, of the projection matrix are the vanishing points of the directions of the world axes.*

Under the mapping $\mathbf{x} = \mathtt{P}\mathbf{X}$ points on this line are projected to

$$\mathbf{x} = \mathtt{P}\mathbf{X}(\lambda) = \lambda \mathtt{P}\mathbf{A} + (1-\lambda)\mathtt{P}\mathbf{C} = \lambda \mathtt{P}\mathbf{A}$$

since $\mathtt{P}\mathbf{C} = \mathbf{0}$. That is all points on the line are mapped to the same image point $\mathtt{P}\mathbf{A}$, which means that the line must be a ray through the camera centre. It follows that $\mathbf{C}$ is the homogeneous representation of the camera centre, since for all choices of $\mathbf{A}$ the line $\mathbf{X}(\lambda)$ is a ray through the camera centre.

This result is not unexpected since the image point $(0,0,0)^\mathsf{T} = \mathtt{P}\mathbf{C}$ is not defined, and the camera centre is the unique point in space for which the image is undefined. In the case of finite cameras the result may be established directly, since $\mathbf{C} = (\widetilde{\mathbf{C}}^\mathsf{T}, 1)^\mathsf{T}$ is clearly the null-vector of $\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]$. The result is true even in the case where the first 3×3 submatrix $\mathtt{M}$ of $\mathtt{P}$ is singular. In this singular case, though, the null-vector has the form $\mathbf{C} = (\mathbf{d}^\mathsf{T}, 0)^\mathsf{T}$ where $\mathtt{M}\mathbf{d} = \mathbf{0}$. The camera centre is then a point at infinity. Camera models of this class are discussed in section 6.3.

Column vectors. The columns of the projective camera are 3-vectors which have a geometric meaning as particular image points. With the notation that the columns of $\mathtt{P}$ are $\mathbf{p}_i, i = 1, \ldots, 4$, then $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are the vanishing points of the world coordinate $\mathtt{X}$, $\mathtt{Y}$ and $\mathtt{Z}$ axes respectively. This follows because these points are the images of the axes' directions. For example the x-axis has direction $\mathbf{D} = (1,0,0,0)^\mathsf{T}$, which is imaged at $\mathbf{p}_1 = \mathtt{P}\mathbf{D}$. See figure 6.4. The column $\mathbf{p}_4$ is the image of the world origin.

Row vectors. The rows of the projective camera (6.12) are 4-vectors which may be interpreted geometrically as particular world planes. These planes are examined next. We introduce the notation that the rows of $\mathtt{P}$ are $\mathbf{P}^{i\mathsf{T}}$ so that

$$\mathtt{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \mathbf{P}^{1\mathsf{T}} \\ \mathbf{P}^{2\mathsf{T}} \\ \mathbf{P}^{3\mathsf{T}} \end{bmatrix}. \tag{6.12}$$

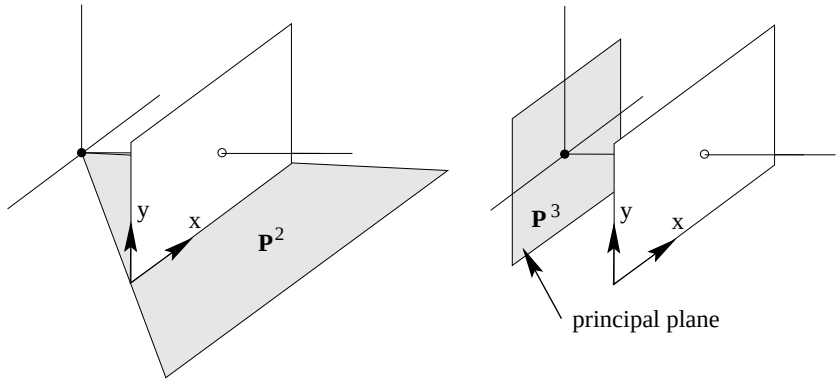


Fig. 6.5. *Two of the three planes defined by the rows of the projection matrix.*

The principal plane. The principal plane is the plane through the camera centre parallel to the image plane. It consists of the set of points $\mathbf{X}$ which are imaged on the line at infinity of the image. Explicitly, $\mathtt{P}\mathbf{X} = (x, y, 0)^\mathsf{T}$. Thus a point lies on the principal plane of the camera if and only if $\mathbf{P}^{3\mathsf{T}}\mathbf{X} = 0$. In other words, $\mathbf{P}^3$ is the vector representing the principal plane of the camera. If $\mathbf{C}$ is the camera centre, then $\mathtt{P}\mathbf{C} = \mathbf{0}$, and so in particular $\mathbf{P}^{3\mathsf{T}}\mathbf{C} = 0$. That is $\mathbf{C}$ lies on the principal plane of the camera.

Axis planes. Consider the set of points $\mathbf{X}$ on the plane $\mathbf{P}^1$. This set satisfies $\mathbf{P}^{1\mathsf{T}}\mathbf{X} = 0$, and so is imaged at $\mathtt{P}\mathbf{X} = (0, y, w)^\mathsf{T}$ which are points on the image y-axis. Again it follows from $\mathtt{P}\mathbf{C} = \mathbf{0}$ that $\mathbf{P}^{1\mathsf{T}}\mathbf{C} = 0$ and so $\mathbf{C}$ lies also on the plane $\mathbf{P}^1$. Consequently the plane $\mathbf{P}^1$ is defined by the camera centre and the line $x = 0$ in the image. Similarly the plane $\mathbf{P}^2$ is defined by the camera centre and the line $y = 0$.

Unlike the principal plane $\mathbf{P}^3$, the axis planes $\mathbf{P}^1$ and $\mathbf{P}^2$ are dependent on the image x- and y-axes, i.e. on the choice of the image coordinate system. Thus they are less tightly coupled to the natural camera geometry than the principal plane. In particular the line of intersection of the planes $\mathbf{P}^1$ and $\mathbf{P}^2$ is a line joining the camera centre and image origin, i.e. the back-projection of the image origin. This line will not coincide in general with the camera principal axis. The planes arising from $\mathbf{P}^i$ are illustrated in figure 6.5.

The camera centre $\mathbf{C}$ lies on all three planes, and since these planes are distinct (as the $\mathtt{P}$ matrix has rank 3) it must lie on their intersection. Algebraically, the condition for the centre to lie on all three planes is $\mathtt{P}\mathbf{C} = \mathbf{0}$ which is the original equation for the camera centre given above.

The principal point. The principal axis is the line passing through the camera centre $\mathbf{C}$, with direction perpendicular to the principal plane $\mathbf{P}^3$. The axis intersects the image plane at the principal point. We may determine this point as follows. In general, the normal to a plane $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^\mathsf{T}$ is the vector $(\pi_1, \pi_2, \pi_3)^\mathsf{T}$. This may alternatively be represented by a point $(\pi_1, \pi_2, \pi_3, 0)^\mathsf{T}$ on the plane at infinity. In the case of the principal plane $\mathbf{P}^3$ of the camera, this point is $(p_{31}, p_{32}, p_{33}, 0)^\mathsf{T}$, which we denote $\widehat{\mathbf{P}}^3$. Projecting that point using the camera matrix $\mathtt{P}$ gives the principal point of the

camera $\mathtt{P}\widehat{\mathbf{P}}^3$. Note that only the left hand 3×3 part of $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$ is involved in this formula. In fact the principal point is computed as $\mathbf{x}_0 = \mathtt{M}\mathbf{m}^3$ where $\mathbf{m}^{3\mathsf{T}}$ is the third row of $\mathtt{M}$.

The principal axis vector. Although any point $\mathbf{X}$ not on the principal plane may be mapped to an image point according to $\mathbf{x} = \mathtt{P}\mathbf{X}$, in reality only half the points in space, those that lie in front of the camera, may be seen in an image. Let $\mathtt{P}$ be written as $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$. It has just been seen that the vector $\mathbf{m}^3$ points in the direction of the principal axis. We would like to define this vector in such a way that it points in the direction towards the front of the camera (the *positive* direction). Note however that $\mathtt{P}$ is only defined up to sign. This leaves an ambiguity as to whether $\mathbf{m}^3$ or $-\mathbf{m}^3$ points in the positive direction. We now proceed to resolve this ambiguity.

We start by considering coordinates with respect to the camera coordinate frame. According to (6.5), the equation for projection of a 3D point to a point in the image is given by $\mathbf{x} = \mathtt{P}_{\text{cam}}\mathbf{X}_{\text{cam}} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\text{cam}}$, where $\mathbf{X}_{\text{cam}}$ is the 3D point expressed in camera coordinates. In this case observe that the vector $\mathbf{v} = \det(\mathtt{M})\mathbf{m}^3 = (0, 0, 1)^{\mathsf{T}}$ points *towards the front of the camera* in the direction of the principal axis, irrespective of the scaling of $\mathtt{P}_{\text{cam}}$. For example, if $\mathtt{P}_{\text{cam}} \rightarrow k\mathtt{P}_{\text{cam}}$ then $\mathbf{v} \rightarrow k^4\mathbf{v}$ which has the same direction.

If the 3D point is expressed in world coordinates then $\mathtt{P} = k\mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathbf{C}}] = [\mathtt{M} \mid \mathbf{p}_4]$, where $\mathtt{M} = k\mathtt{K}\mathtt{R}$. Since $\det(\mathtt{R}) > 0$ the vector $\mathbf{v} = \det(\mathtt{M})\mathbf{m}^3$ is again unaffected by scaling. In summary,

- $\mathbf{v} = \det(\mathtt{M})\mathbf{m}^3$ *is a vector in the direction of the principal axis, directed towards the front of the camera.*

6.2.2 Action of a projective camera on points

Forward projection. As we have already seen, a general projective camera maps a point in space $\mathbf{X}$ to an image point according to the mapping $\mathbf{x} = \mathtt{P}\mathbf{X}$. Points $\mathbf{D} = (\mathbf{d}^{\mathsf{T}}, 0)^{\mathsf{T}}$ on the plane at infinity represent vanishing points. Such points map to

$$\mathbf{x} = \mathtt{P}\mathbf{D} = [\mathtt{M} \mid \mathbf{p}_4]\mathbf{D} = \mathtt{M}\mathbf{d}$$

and thus are only affected by $\mathtt{M}$, the first 3×3 submatrix of $\mathtt{P}$.

Back-projection of points to rays. Given a point $\mathbf{x}$ in an image, we next determine the set of points in space that map to this point. This set will constitute a ray in space passing through the camera centre. The form of the ray may be specified in several ways, depending on how one wishes to represent a line in 3-space. A Plücker representation is postponed until section 8.1.2(*p*196). Here the line is represented as the join of two points.

We know two points on the ray. These are the camera centre $\mathbf{C}$ (where $\mathtt{P}\mathbf{C} = \mathbf{0}$) and the point $\mathtt{P}^+\mathbf{x}$, where $\mathtt{P}^+$ is the pseudo-inverse of $\mathtt{P}$. The pseudo-inverse of $\mathtt{P}$ is the matrix $\mathtt{P}^+ = \mathtt{P}^{\mathsf{T}}(\mathtt{P}\mathtt{P}^{\mathsf{T}})^{-1}$, for which $\mathtt{P}\mathtt{P}^+ = \mathtt{I}$ (see section A5.2(*p*590)). Point $\mathtt{P}^+\mathbf{x}$ lies

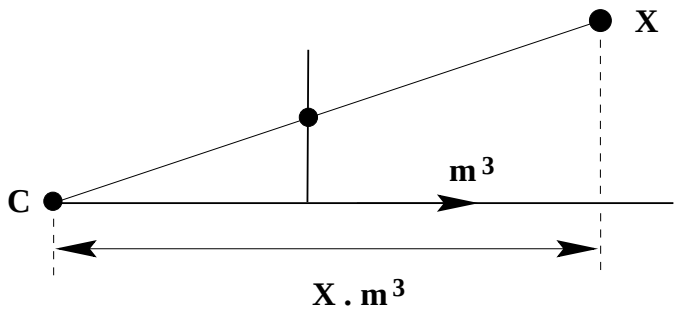


Fig. 6.6. *If the camera matrix* $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$ *is normalized so that* $\|\mathbf{m}^3\| = 1$ *and* $\det \mathtt{M} > 0$*, and* $\mathbf{x} = w(x, y, 1)^\mathsf{T} = \mathtt{P}\mathbf{X}$*, where* $\mathbf{X} = (\mathtt{X}, \mathtt{Y}, \mathtt{Z}, 1)^\mathsf{T}$*, then* w *is the depth of the point* $\mathbf{X}$ *from the camera centre in the direction of the principal ray of the camera.*

on the ray because it projects to $\mathbf{x}$, since $\mathtt{P}(\mathtt{P}^+\mathbf{x}) = \mathtt{I}\mathbf{x} = \mathbf{x}$. Then the ray is the line formed by the join of these two points

$$\mathbf{X}(\lambda) = \mathtt{P}^+\mathbf{x} + \lambda\mathbf{C}. \tag{6.13}$$

In the case of finite cameras an alternative expression can be developed. Writing $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$, the camera centre is given by $\widetilde{\mathbf{C}} = -\mathtt{M}^{-1}\mathbf{p}_4$. An image point $\mathbf{x}$ back-projects to a ray intersecting the plane at infinity at the point $\mathbf{D} = ((\mathtt{M}^{-1}\mathbf{x})^\mathsf{T}, 0)^\mathsf{T}$, and $\mathbf{D}$ provides a second point on the ray. Again writing the line as the join of two points on the ray,

$$\mathbf{X}(\mu) = \mu\begin{pmatrix} \mathtt{M}^{-1}\mathbf{x} \\ 0 \end{pmatrix} + \begin{pmatrix} -\mathtt{M}^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix} = \begin{pmatrix} \mathtt{M}^{-1}(\mu\mathbf{x} - \mathbf{p}_4) \\ 1 \end{pmatrix}. \tag{6.14}$$

6.2.3 Depth of points

Next, we consider the distance a point lies in front of or behind the principal plane of the camera. Consider a camera matrix $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$, projecting a point $\mathbf{X} = (\mathtt{X}, \mathtt{Y}, \mathtt{Z}, 1)^\mathsf{T} = (\widetilde{\mathbf{X}}^\mathsf{T}, 1)^\mathsf{T}$ in 3-space to the image point $\mathbf{x} = w(x, y, 1)^\mathsf{T} = \mathtt{P}\mathbf{X}$. Let $\mathbf{C} = (\widetilde{\mathbf{C}}, 1)^\mathsf{T}$ be the camera centre. Then $w = \mathbf{P}^{3\mathsf{T}}\mathbf{X} = \mathbf{P}^{3\mathsf{T}}(\mathbf{X} - \mathbf{C})$ since $\mathtt{P}\mathbf{C} = \mathbf{0}$ for the camera centre $\mathbf{C}$. However, $\mathbf{P}^{3\mathsf{T}}(\mathbf{X} - \mathbf{C}) = \mathbf{m}^{3\mathsf{T}}(\widetilde{\mathbf{X}} - \widetilde{\mathbf{C}})$ where $\mathbf{m}^3$ is the principal ray direction, so $w = \mathbf{m}^{3\mathsf{T}}(\widetilde{\mathbf{X}} - \widetilde{\mathbf{C}})$ can be interpreted as the dot product of the ray from the camera centre to the point $\mathbf{X}$, with the principal ray direction. If the camera matrix is normalized so that $\det \mathtt{M} > 0$ and $\|\mathbf{m}^3\| = 1$, then $\mathbf{m}^3$ is a unit vector pointing in the *positive* axial direction. Then w may be interpreted as the depth of the point $\mathbf{X}$ from the camera centre $\mathbf{C}$ in the direction of the principal ray. This is illustrated in figure 6.6.

Any camera matrix may be normalized by multiplying it by an appropriate factor. However, to avoid having always to deal with normalized camera matrices, the depth of a point may be computed as follows:

Result 6.1. *Let* $\mathbf{X} = (\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{T})^\mathsf{T}$ *be a 3D point and* $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$ *be a camera matrix for a finite camera. Suppose* $\mathtt{P}(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{T})^\mathsf{T} = w(x, y, 1)^\mathsf{T}$*. Then*

$$\text{depth}(\mathbf{X}; \mathtt{P}) = \frac{\text{sign}(\det \mathtt{M})w}{\mathtt{T}\|\mathbf{m}^3\|} \tag{6.15}$$

is the depth of the point $\mathbf{X}$ *in front of the principal plane of the camera.*

This formula is an effective way to determine if a point $\mathbf{X}$ is in front of the camera. One verifies that the value of $\text{depth}(\mathbf{X}; \mathtt{P})$ is unchanged if either the point $\mathbf{X}$ or the camera matrix $\mathtt{P}$ is multiplied by a constant factor k. Thus, $\text{depth}(\mathbf{X}; \mathtt{P})$ is independent of the particular homogeneous representation of $\mathbf{X}$ and $\mathtt{P}$.

6.2.4 Decomposition of the camera matrix

Let $\mathtt{P}$ be a camera matrix representing a general projective camera. We wish to find the camera centre, the orientation of the camera and the internal parameters of the camera from $\mathtt{P}$.

Finding the camera centre. The camera centre $\mathbf{C}$ is the point for which $\mathtt{P}\mathbf{C} = \mathbf{0}$. Numerically this right null-vector may be obtained from the SVD of $\mathtt{P}$, see section A4.4(*p*585). Algebraically, the centre $\mathbf{C} = (\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{T})^\mathsf{T}$ may be obtained as (see (3.5–*p*67))

$$\begin{aligned} \mathtt{X} &= \det([\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]) \quad \mathtt{Y} &&= -\det([\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4]) \\ \mathtt{Z} &= \det([\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4]) \quad \mathtt{T} &&= -\det([\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]). \end{aligned}$$

Finding the camera orientation and internal parameters. In the case of a finite camera, according to (6.11),

$$\mathtt{P} = [\mathtt{M} \mid -\mathtt{M}\widetilde{\mathbf{C}}] = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathbf{C}}].$$

We may easily find both $\mathtt{K}$ and $\mathtt{R}$ by decomposing $\mathtt{M}$ as $\mathtt{M} = \mathtt{KR}$ using the RQ-decomposition. This decomposition into the product of an upper-triangular and orthogonal matrix is described in section A4.1.1(*p*579). The matrix $\mathtt{R}$ gives the orientation of the camera, whereas $\mathtt{K}$ is the calibration matrix. The ambiguity in the decomposition is removed by requiring that $\mathtt{K}$ have positive diagonal entries.

The matrix $\mathtt{K}$ has the form (6.10)

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where

- α_x is the scale factor in the x-coordinate direction,
- α_y is the scale factor in the y-coordinate direction,
- s is the skew,
- $(x_0, y_0)^\mathsf{T}$ are the coordinates of the principal point.

The *aspect ratio* is α_y/α_x.

Example 6.2. The camera matrix

$$\mathtt{P} = \begin{bmatrix} 3.53553\,e{+}2 & 3.39645\,e{+}2 & 2.77744\,e{+}2 & -1.44946\,e{+}6 \\ -1.03528\,e{+}2 & 2.33212\,e{+}1 & 4.59607\,e{+}2 & -6.32525\,e{+}5 \\ 7.07107\,e{-}1 & -3.53553\,e{-}1 & 6.12372\,e{-}1 & -9.18559\,e{+}2 \end{bmatrix}$$

with $\mathtt{P} = [\mathtt{M} \mid -\mathtt{M}\widetilde{\mathbf{C}}]$, has centre $\widetilde{\mathbf{C}} = (1000.0, 2000.0, 1500.0)^\mathsf{T}$, and the matrix $\mathtt{M}$ decomposes as

$$\mathtt{M} = \mathtt{KR} = \begin{bmatrix} 468.2 & 91.2 & 300.0 \\ & 427.2 & 200.0 \\ & & 1.0 \end{bmatrix} \begin{bmatrix} 0.41380 & 0.90915 & 0.04708 \\ -0.57338 & 0.22011 & 0.78917 \\ 0.70711 & -0.35355 & 0.61237 \end{bmatrix}.$$

△

When is $s \neq 0$? As was shown in section 6.1 a true CCD camera has only four internal camera parameters, since generally $s = 0$. If $s \neq 0$ then this can be interpreted as a skewing of the pixel elements in the CCD array so that the x- and y-axes are not perpendicular. This is admittedly very unlikely to happen.

In realistic circumstances a non-zero skew might arise as a result of taking an image of an image, for example if a photograph is re-photographed, or a negative is enlarged. Consider enlarging an image taken by a pinhole camera (such as an ordinary film camera) where the axis of the magnifying lens is not perpendicular to the film plane or the enlarged image plane.

The most severe distortion that can arise from this "picture of a picture" process is a planar homography. Suppose the original (finite) camera is represented by the matrix $\mathtt{P}$, then the camera representing the picture of a picture is $\mathtt{HP}$, where $\mathtt{H}$ is the homography matrix. Since $\mathtt{H}$ is non-singular, the left 3×3 submatrix of $\mathtt{HP}$ is non-singular and can be decomposed as the product $\mathtt{KR}$ – and $\mathtt{K}$ need not have $s = 0$. Note however that the $\mathtt{K}$ and $\mathtt{R}$ are no longer the calibration matrix and orientation of the original camera.

On the other hand, one verifies that the process of taking a picture of a picture does not change the apparent camera centre. Indeed, since $\mathtt{H}$ is non-singular, $\mathtt{HP}\mathbf{C} = \mathbf{0}$ if and only if $\mathtt{P}\mathbf{C} = \mathbf{0}$.

Where is the decomposition required? If the camera $\mathtt{P}$ is constructed from (6.11) then the parameters are known and a decomposition is clearly unnecessary. So the question arises – where would one obtain a camera for which the decomposition is not known? In fact cameras will be computed in myriad ways throughout this book and decomposing an unknown camera is a frequently used tool in practice. For example cameras can be computed directly by *calibration* – where the camera is computed from a set of world to image correspondences (chapter 7) – and indirectly by computing a multiple view relation (such as the fundamental matrix or trifocal tensor) and subsequently computing projection matrices from this relation.

A note on coordinate orientation. In the derivation of the camera model and its parametrization (6.10) it is assumed that the coordinate systems used in both the image and the 3D world are right handed systems, as shown in figure 6.1(*p*154). However, a common practice in measuring image coordinates is that the y-coordinate increases in the downwards direction, thus defining a left handed coordinate system, contrary to figure 6.1(*p*154). A recommended practice in this case is to negate the y-coordinate of the image point so that the coordinate system again becomes right handed. However, if

the image coordinate system is left handed, then the consequences are not grave. The relationship between world and image coordinates is still expressed by a 3×4 camera matrix. Decomposition of this camera matrix according to (6.11) with $\mathtt{K}$ of the form (6.10) is still possible with α_x and α_y positive. The difference is that $\mathtt{R}$ now represents the orientation of the camera with respect to the negative $\mathtt{Z}$-axis. In addition, the depth of points given by (6.15) will be *negative* instead of positive for points in front of the camera. If this is borne in mind then it is permissible to use left handed coordinates in the image.

6.2.5 Euclidean vs projective spaces

The development of the sections to this point has implicitly assumed that the world and image coordinate systems are Euclidean. Ideas have been borrowed from projective geometry (such as directions corresponding to points on $\boldsymbol{\pi}_\infty$) and the convenient notation of homogeneous coordinates has allowed central projection to be represented linearly.

In subsequent chapters of the book we will go further and use a projective coordinate frame. This is easily achieved, for suppose the world coordinate frame is projective; then the transformation between the camera and world coordinate frame (6.6) is again represented by a 4×4 homogeneous matrix, $\mathbf{X}_{\text{cam}} = \mathtt{H}\mathbf{X}$, and the resulting map from projective 3-space $\mathbb{P}^3$ to the image is still represented by a 3×4 matrix $\mathtt{P}$ with rank 3. In fact, at its most general the projective camera is a map from $\mathbb{P}^3$ to $\mathbb{P}^2$, and covers the composed effects of a projective transformation of 3-space, a projection from 3-space to an image, and a projective transformation of the image. This follows simply by concatenating the matrices representing these mappings:

$$\mathtt{P} = [3 \times 3 \text{ homography}] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} [4 \times 4 \text{ homography}]$$

which results in a 3×4 matrix.

However, it is important to remember that cameras are Euclidean devices and simply because we have a projective model of a camera it does not mean that we should eschew notions of Euclidean geometry.

Euclidean and affine interpretations. Although a (finite) 3×4 matrix can always be decomposed as in section 6.2.4 to obtain a rotation matrix, a calibration matrix $\mathtt{K}$, and so forth, Euclidean interpretations of the parameters so obtained are only meaningful if the image and space coordinates are in an appropriate frame. In the decomposition case a Euclidean frame is required for both image and 3-space. On the other hand, the interpretation of the null-vector of $\mathtt{P}$ as the camera centre is valid even if both frames are projective – the interpretation requires only collinearity, which is a projective notion. The interpretation of $\mathbf{P}^3$ as the principal plane requires at least affine frames for the image and 3-space. Finally, the interpretation of $\mathbf{m}^3$ as the principal ray requires an affine image frame but a Euclidean world frame in order for the concept of orthogonality (to the principal plane) to be meaningful.

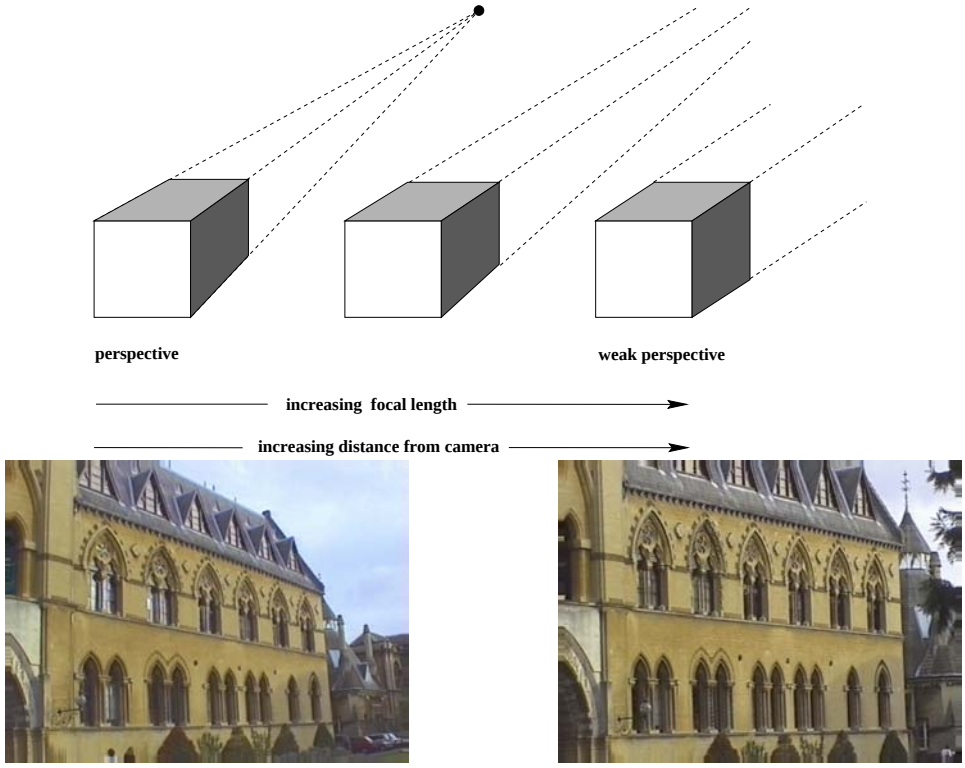


Fig. 6.7. *As the focal length increases and the distance between the camera and object also increases, the image remains the same size but perspective effects diminish.*

6.3 Cameras at infinity

We now turn to consider cameras with centre lying on the plane at infinity. This means that the left hand 3×3 block of the camera matrix $\mathtt{P}$ is singular. The camera centre may be found from $\mathtt{P}\mathbf{C} = \mathbf{0}$ just as with finite cameras.

Cameras at infinity may be broadly classified into two different types, *affine cameras* and *non-affine cameras*. We consider first of all the affine class of cameras which are the most important in practice.

Definition 6.3. An *affine* camera is one that has a camera matrix $\mathtt{P}$ in which the last row $\mathbf{P}^{3\mathsf{T}}$ is of the form $(0, 0, 0, 1)$.

It is called an affine camera because points at infinity are mapped to points at infinity.

6.3.1 Affine cameras

Consider what happens as we apply a cinematographic technique of tracking back while zooming in, in such a way as to keep objects of interest the same size[1]. This is illustrated in figure 6.7. We are going to model this process by taking the limit as both the focal length and principal axis distance of the camera from the object increase.

In analyzing this technique, we start with a finite projective camera (6.11). The

[1] See 'Vertigo' (Dir. Hitchcock, 1958) and 'Mishima' (Dir. Schrader, 1985).

camera matrix may be written as

$$\mathtt{P}_0 = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\widetilde{\mathbf{C}}] = \mathtt{K}\begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & -\mathbf{r}^{1\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{2\mathsf{T}} & -\mathbf{r}^{2\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{3\mathsf{T}} & -\mathbf{r}^{3\mathsf{T}}\widetilde{\mathbf{C}} \end{bmatrix} \tag{6.16}$$

where $\mathbf{r}^{i\mathsf{T}}$ is the i-th row of the rotation matrix. This camera is located at position $\widetilde{\mathbf{C}}$ and has orientation denoted by matrix $\mathtt{R}$ and internal parameters matrix $\mathtt{K}$ of the form given in (6.10–*p*157). From section 6.2.1 the principal ray of the camera is in the direction of the vector $\mathbf{r}^3$, and the value $d_0 = -\mathbf{r}^{3\mathsf{T}}\widetilde{\mathbf{C}}$ is the distance of the world origin from the camera centre in the direction of the principal ray.

Now, we consider what happens if the camera centre is moved backwards along the principal ray at unit speed for a time t, so that the centre of the camera is moved to $\widetilde{\mathbf{C}} - t\mathbf{r}^3$. Replacing $\widetilde{\mathbf{C}}$ in (6.16) by $\widetilde{\mathbf{C}} - t\mathbf{r}^3$ gives the camera matrix at time t:

$$\mathtt{P}_t = \mathtt{K}\begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & -\mathbf{r}^{1\mathsf{T}}(\widetilde{\mathbf{C}} - t\mathbf{r}^3) \\ \mathbf{r}^{2\mathsf{T}} & -\mathbf{r}^{2\mathsf{T}}(\widetilde{\mathbf{C}} - t\mathbf{r}^3) \\ \mathbf{r}^{3\mathsf{T}} & -\mathbf{r}^{3\mathsf{T}}(\widetilde{\mathbf{C}} - t\mathbf{r}^3) \end{bmatrix} = \mathtt{K}\begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & -\mathbf{r}^{1\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{2\mathsf{T}} & -\mathbf{r}^{2\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{3\mathsf{T}} & d_t \end{bmatrix} \tag{6.17}$$

where the terms $\mathbf{r}^{i\mathsf{T}}\mathbf{r}^3$ are zero for $i = 1, 2$ because $\mathtt{R}$ is a rotation matrix. The scalar $d_t = -\mathbf{r}^{3\mathsf{T}}\widetilde{\mathbf{C}} + t$ is the depth of the world origin with respect to the camera centre in the direction of the principal ray $\mathbf{r}^3$ of the camera. Thus

- *The effect of tracking along the principal ray is to replace the (3,4) entry of the matrix by the depth d_t of the camera centre from the world origin.*

Next, we consider zooming such that the camera focal length is increased by a factor k. This magnifies the image by a factor k. It is shown in section 8.4.1(*p*203) that the effect of zooming by a factor k is to multiply the calibration matrix $\mathtt{K}$ on the right by $\mathrm{diag}(k, k, 1)$.

Now, we combine the effects of tracking and zooming. We suppose that the magnification factor is $k = d_t/d_0$ so that the image size remains fixed. The resulting camera matrix at time t, derived from (6.17), is

$$\mathtt{P}_t = \mathtt{K}\begin{bmatrix} d_t/d_0 & & \\ & d_t/d_0 & \\ & & 1 \end{bmatrix}\begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & -\mathbf{r}^{1\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{2\mathsf{T}} & -\mathbf{r}^{2\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{3\mathsf{T}} & d_t \end{bmatrix} = \frac{d_t}{d_0}\mathtt{K}\begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & -\mathbf{r}^{1\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{2\mathsf{T}} & -\mathbf{r}^{2\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{3\mathsf{T}}d_0/d_t & d_0 \end{bmatrix}$$

and one can ignore the factor d_t/d_0. When $t = 0$, the camera matrix $\mathtt{P}_t$ corresponds with (6.16). Now, in the limit as d_t tends to infinity, this matrix becomes

$$\mathtt{P}_\infty = \lim_{t\to\infty} \mathtt{P}_t = \mathtt{K}\begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & -\mathbf{r}^{1\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{r}^{2\mathsf{T}} & -\mathbf{r}^{2\mathsf{T}}\widetilde{\mathbf{C}} \\ \mathbf{0}^{\mathsf{T}} & d_0 \end{bmatrix} \tag{6.18}$$

which is just the original camera matrix (6.16) with the first three entries of the last row set to zero. From definition 6.3 $\mathtt{P}_\infty$ is an instance of an affine camera.

6.3.2 Error in employing an affine camera

It may be noted that the image of any point on the plane through the world origin perpendicular to the principal axis direction $\mathbf{r}^3$ is unchanged by this combined zooming and motion. Indeed, such a point may be written as

$$\mathbf{X} = \begin{pmatrix} \alpha\mathbf{r}^1 + \beta\mathbf{r}^2 \\ 1 \end{pmatrix}.$$

One then verifies that $\mathtt{P}_0\mathbf{X} = \mathtt{P}_t\mathbf{X} = \mathtt{P}_\infty\mathbf{X}$ for all t, since $\mathbf{r}^{3\mathsf{T}}(\alpha\mathbf{r}^1 + \beta\mathbf{r}^2) = 0$.

For points not on this plane the images under $\mathtt{P}_0$ and $\mathtt{P}_\infty$ differ, and we will now investigate the extent of this error. Consider a point $\mathbf{X}$ which is at a perpendicular distance Δ from this plane. The 3D point can be represented as

$$\mathbf{X} = \begin{pmatrix} \alpha\mathbf{r}^1 + \beta\mathbf{r}^2 + \Delta\mathbf{r}^3 \\ 1 \end{pmatrix}$$

and is imaged by the cameras $\mathtt{P}_0$ and $\mathtt{P}_\infty$ at

$$\mathbf{x}_{\text{proj}} = \mathtt{P}_0\mathbf{X} = \mathtt{K}\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ d_0 + \Delta \end{pmatrix} \quad \mathbf{x}_{\text{affine}} = \mathtt{P}_\infty\mathbf{X} = \mathtt{K}\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ d_0 \end{pmatrix}$$

where $\tilde{x} = \alpha - \mathbf{r}^{1\mathsf{T}}\widetilde{\mathbf{C}}$, $\tilde{y} = \beta - \mathbf{r}^{2\mathsf{T}}\widetilde{\mathbf{C}}$. Now, writing the calibration matrix as

$$\mathtt{K} = \begin{bmatrix} \mathtt{K}_{2\times2} & \tilde{\mathbf{x}}_0 \\ \tilde{\mathbf{0}}^{\mathsf{T}} & 1 \end{bmatrix},$$

where $\mathtt{K}_{2\times2}$ is an upper-triangular 2×2 matrix, gives

$$\mathbf{x}_{\text{proj}} = \begin{pmatrix} \mathtt{K}_{2\times2}\tilde{\mathbf{x}} + (d_0 + \Delta)\tilde{\mathbf{x}}_0 \\ d_0 + \Delta \end{pmatrix} \quad \mathbf{x}_{\text{affine}} = \begin{pmatrix} \mathtt{K}_{2\times2}\tilde{\mathbf{x}} + d_0\tilde{\mathbf{x}}_0 \\ d_0 \end{pmatrix}$$

The image point for $\mathtt{P}_0$ is obtained by dehomogenizing, by dividing by the third element, as $\tilde{\mathbf{x}}_{\text{proj}} = \tilde{\mathbf{x}}_0 + \mathtt{K}_{2\times2}\tilde{\mathbf{x}}/(d_0 + \Delta)$, and for $\mathtt{P}_\infty$ the inhomogeneous image point is $\tilde{\mathbf{x}}_{\text{affine}} = \tilde{\mathbf{x}}_0 + \mathtt{K}_{2\times2}\tilde{\mathbf{x}}/d_0$. The relationship between the two points is therefore

$$\tilde{\mathbf{x}}_{\text{affine}} - \tilde{\mathbf{x}}_0 = \frac{d_0 + \Delta}{d_0}(\tilde{\mathbf{x}}_{\text{proj}} - \tilde{\mathbf{x}}_0)$$

which shows that

- *The effect of the affine approximation $\mathtt{P}_\infty$ to the true camera matrix $\mathtt{P}_0$ is to move the image of a point $\mathbf{X}$ radially towards or away from the principal point $\tilde{\mathbf{x}}_0$ by a factor equal to $(d_0 + \Delta)/d_0 = 1 + \Delta/d_0$.*

This is illustrated in figure 6.8.

Affine imaging conditions. From the expressions for $\tilde{\mathbf{x}}_{\text{proj}}$ and $\tilde{\mathbf{x}}_{\text{affine}}$ we can deduce that

$$\tilde{\mathbf{x}}_{\text{affine}} - \tilde{\mathbf{x}}_{\text{proj}} = \frac{\Delta}{d_0}(\tilde{\mathbf{x}}_{\text{proj}} - \tilde{\mathbf{x}}_0) \tag{6.19}$$

which shows that the distance between the true perspective image position and the position obtained using the affine camera approximation $\mathtt{P}_\infty$ will be small provided:

(i) The depth relief (Δ) is small compared to the average depth (d_0), and
(ii) The distance of the point from the principal ray is small.

The latter condition is satisfied by a small field of view. In general, images acquired using a lens with a longer focal length tend to satisfy these conditions as both the field of view and the depth of field are smaller than those obtained by a short focal length lens with the same CCD array.

For scenes at which there are many points at different depths, the affine camera is not a good approximation. For instance where the scene contains close foreground as well as background objects, an affine camera model should not be used. However, a different affine model can be used for each region in these circumstances.

6.3.3 Decomposition of $\mathtt{P}_\infty$

The camera matrix (6.18) may be written as

$$\mathtt{P}_\infty = \begin{bmatrix} \mathtt{K}_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\mathsf{T} & d_0 \end{bmatrix}$$

where $\hat{\mathtt{R}}$ consists of the two first rows of a rotation matrix, $\hat{\mathbf{t}}$ is the vector $(-\mathbf{r}^{1\mathsf{T}}\widetilde{\mathbf{C}}, -\mathbf{r}^{2\mathsf{T}}\widetilde{\mathbf{C}})^\mathsf{T}$, and $\hat{\mathbf{0}}$ the vector $(0,0)^\mathsf{T}$. The 2×2 matrix $\mathtt{K}_{2\times 2}$ is upper-triangular. One quickly verifies that

$$\mathtt{P}_\infty = \begin{bmatrix} \mathtt{K}_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\mathsf{T} & d_0 \end{bmatrix} = \begin{bmatrix} d_0^{-1}\mathtt{K}_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$$

so we may replace $\mathtt{K}_{2\times 2}$ by $d_0^{-1}\mathtt{K}_{2\times 2}$ and assume that $d_0 = 1$. Multiplying out this product gives

$$\begin{aligned} \mathtt{P}_\infty &= \begin{bmatrix} \mathtt{K}_{2\times 2}\hat{\mathtt{R}} & \mathtt{K}_{2\times 2}\hat{\mathbf{t}} + \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \mathtt{K}_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{t}} + \mathtt{K}_{2\times 2}^{-1}\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathtt{K}_{2\times 2} & \mathtt{K}_{2\times 2}\hat{\mathbf{t}} + \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{0}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \end{aligned}$$

Thus, making appropriate substitutions for $\hat{\mathbf{t}}$ or $\tilde{\mathbf{x}}_0$, we can write the affine camera matrix in one of the two forms

$$\mathtt{P}_\infty = \begin{bmatrix} \mathtt{K}_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \mathtt{K}_{2\times 2} & \tilde{\mathbf{x}}_0 \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{0}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.20}$$

Consequently, the camera $\mathtt{P}_\infty$ can be interpreted in terms of these decompositions in

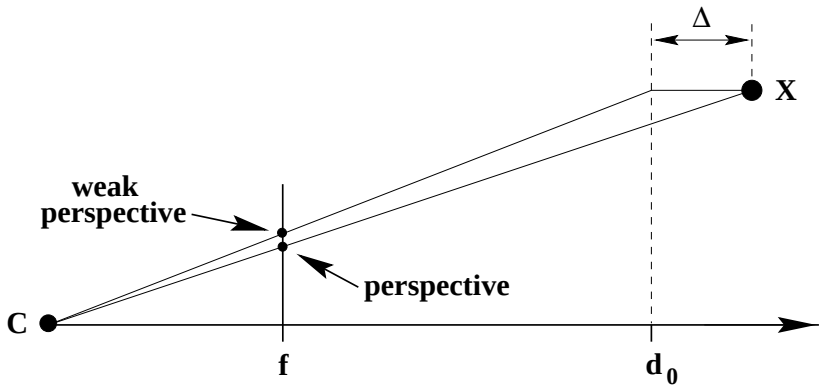


Fig. 6.8. **Perspective vs weak perspective projection.** *The action of the weak perspective camera is equivalent to orthographic projection onto a plane (at $Z = d_0$), followed by perspective projection from the plane. The difference between the perspective and weak perspective image point depends both on the distance Δ of the point $\mathbf{X}$ from the plane, and the distance of the point from the principal ray.*

one of two ways, either with $\tilde{\mathbf{x}}_0 = \mathbf{0}$ or with $\hat{\mathbf{t}} = \hat{\mathbf{0}}$. Using the second decomposition of (6.20), the image of the world origin is $\mathtt{P}_\infty(0,0,0,1)^\mathsf{T} = (\tilde{\mathbf{x}}_0^\mathsf{T}, 1)^\mathsf{T}$. Consequently, the value of $\tilde{\mathbf{x}}_0$ is dependent on the particular choice of world coordinates, and hence is not an intrinsic property of the camera itself. This means that the camera matrix $\mathtt{P}_\infty$ does not have a principal point. Therefore, it is preferable to use the first decomposition of $\mathtt{P}_\infty$ in (6.20), and write

$$\mathtt{P}_\infty = \begin{bmatrix} \mathtt{K}_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathtt{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{6.21}$$

where the two matrices represent the internal camera parameters and external camera parameters of $\mathtt{P}_\infty$.

Parallel projection. In summary the essential differences between $\mathtt{P}_\infty$ and a finite camera are:

- *The parallel projection matrix* $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ *replaces the canonical projection matrix* $[\mathtt{I} \mid \mathbf{0}]$ *of a finite camera (6.5–*p*155).*
- *The calibration matrix* $\begin{bmatrix} \mathtt{K}_{2\times 2} & \hat{\mathbf{0}} \\ \hat{\mathbf{0}}^\mathsf{T} & 1 \end{bmatrix}$ *replaces* $\mathtt{K}$ *of a finite camera (6.10–*p*157) .*
- *The principal point is not defined.*

6.3.4 A hierarchy of affine cameras

In a similar manner to the development of the finite projection camera taxonomy in section 6.1 we can start with the basic operation of parallel projection and build a hierarchy of camera models representing progressively more general cases of parallel projection.

Orthographic projection. Consider projection along the Z-axis. This is represented by a matrix of the form

$$\mathtt{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.22}$$

This mapping takes a point $(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, 1)^\mathsf{T}$ to the image point $(\mathtt{X}, \mathtt{Y}, 1)^\mathsf{T}$, dropping the Z-coordinate.

For a general orthographic projection mapping, we precede this map by a 3D Euclidean coordinate change of the form

$$\mathtt{H} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}.$$

Writing $\mathbf{t} = (t_1, t_2, t_3)^\mathsf{T}$, we see that a general orthographic camera is of the form

$$\mathtt{P} = \begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & t_1 \\ \mathbf{r}^{2\mathsf{T}} & t_2 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.23}$$

An orthographic camera has five degrees of freedom, namely the three parameters describing the rotation matrix R, plus the two offset parameters t_1 and t_2. An orthographic projection matrix $\mathtt{P} = [\mathtt{M} \mid \mathbf{t}]$ is characterized by a matrix M with last row zero, with the first two rows orthogonal and of unit norm, and $t_3 = 1$.

Scaled orthographic projection. A scaled orthographic projection is an orthographic projection followed by isotropic scaling. Thus, in general, its matrix may be written in the form

$$\mathtt{P} = \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & t_1 \\ \mathbf{r}^{2\mathsf{T}} & t_2 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & t_1 \\ \mathbf{r}^{2\mathsf{T}} & t_2 \\ \mathbf{0}^\mathsf{T} & 1/k \end{bmatrix}. \tag{6.24}$$

It has six degrees of freedom. A scaled orthographic projection matrix $\mathtt{P} = [\mathtt{M} \mid \mathbf{t}]$ is characterized by a matrix M with last row zero, and the first two rows orthogonal and of equal norm.

Weak perspective projection. Analogous to a finite CCD camera, we may consider the case of a camera at infinity for which the scale factors in the two axial image directions are not equal. Such a camera has a projection matrix of the form

$$\mathtt{P} = \begin{bmatrix} \alpha_x & & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & t_1 \\ \mathbf{r}^{2\mathsf{T}} & t_2 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}. \tag{6.25}$$

It has seven degrees of freedom. A weak perspective projection matrix $\mathtt{P} = [\mathtt{M} \mid \mathbf{t}]$ is characterized by a matrix M with last row zero, and first two rows orthogonal (but they need not have equal norm as is required in the scaled orthographic case). The geometric action of this camera is illustrated in figure 6.8.

The affine camera, $\mathtt{P}_{\mathrm{A}}$. As has already been seen in the case of $\mathtt{P}_\infty$, a general camera matrix of the affine form, and with no restrictions on its elements, may be decomposed as

$$\mathtt{P}_{\mathrm{A}} = \begin{bmatrix} \alpha_x & s & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^{1\mathsf{T}} & t_1 \\ \mathbf{r}^{2\mathsf{T}} & t_2 \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}.$$

It has eight degrees of freedom, and may be thought of as the parallel projection version of the finite projective camera (6.11–*p*157).

In full generality an affine camera has the form

$$\mathtt{P}_{\mathrm{A}} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It has eight degrees of freedom corresponding to the eight non-zero and non-unit matrix elements. We denote the top left 2×3 submatrix by $\mathtt{M}_{2\times 3}$. The sole restriction on the affine camera is that $\mathtt{M}_{2\times 3}$ has rank 2. This arises from the requirement that the rank of $\mathtt{P}$ is 3.

The affine camera covers the composed effects of an affine transformation of 3-space, an orthographic projection from 3-space to an image, and an affine transformation of the image. This follows simply by concatenating the matrices representing these mappings:

$$\mathtt{P}_{\mathrm{A}} = [3 \times 3 \text{ affine}] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [4 \times 4 \text{ affine}]$$

which results in a 3×4 matrix of the affine form.

Projection under an affine camera is a linear mapping on *inhomogeneous* coordinates composed with a translation:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

which is written more concisely as

$$\tilde{\mathbf{x}} = \mathtt{M}_{2\times 3}\tilde{\mathbf{X}} + \tilde{\mathbf{t}} \ . \tag{6.26}$$

The point $\tilde{\mathbf{t}} = (t_1, t_2)^{\mathsf{T}}$ is the image of the world origin.

The camera models of this section are seen to be affine cameras satisfying additional constraints, thus the affine camera is an abstraction of this hierarchy. For example, in the case of the weak perspective camera the rows of $\mathtt{M}_{2\times 3}$ are scalings of rows of a rotation matrix, and thus are orthogonal.

6.3.5 More properties of the affine camera

The plane at infinity in space is mapped to points at infinity in the image. This is easily seen by computing $\mathtt{P}_{\mathrm{A}}(\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 0)^{\mathsf{T}} = (\mathrm{X}, \mathrm{Y}, 0)^{\mathsf{T}}$. Extending the terminology of finite

projective cameras, we interpret this by saying that the principal plane of the camera is the plane at infinity. The optical centre, since it lies on the principal plane, must also lie on the plane at infinity. From this we have

(i) Conversely, any projective camera matrix for which the principal plane is the plane at infinity is an affine camera matrix.

(ii) Parallel world lines are projected to parallel image lines. This follows because parallel world lines intersect at the plane at infinity, and this intersection point is mapped to a point at infinity in the image. Hence the image lines are parallel.

(iii) The vector $\mathbf{d}$ satisfying $\mathtt{M}_{2\times 3}\mathbf{d} = \mathbf{0}$ is the *direction* of parallel projection, and $(\mathbf{d}^\mathsf{T}, 0)^\mathsf{T}$ the camera centre since $\mathtt{P}_\mathrm{A}\begin{pmatrix}\mathbf{d}\\0\end{pmatrix} = \mathbf{0}$.

Any camera which consists of the composed effects of affine transformations (either of space, or of the image) with parallel projection will have the affine form. For example, *para-perspective* projection consists of two such mappings: the first is parallel projection onto a plane $\boldsymbol{\pi}$ through the centroid and parallel to the image plane. The direction of parallel projection is the ray joining the centroid to the camera centre. This parallel projection is followed by an affine transformation (actually a similarity) between $\boldsymbol{\pi}$ and the image. Thus a para-perspective camera is an affine camera.

6.3.6 General cameras at infinity

An affine camera is one for which the principal plane is the plane at infinity. As such, its camera centre lies on the plane at infinity. However, it is possible for the camera centre to lie on the plane at infinity without the whole principal plane being the plane at infinity.

A camera centre lies at infinity if $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$ with $\mathtt{M}$ a singular matrix. This is clearly a weaker condition than insisting that the last row of $\mathtt{M}$ is zero, as is the case for affine cameras. If $\mathtt{M}$ is singular, but the last row of $\mathtt{M}$ is not zero, then the camera is not affine, but not a finite projective camera either. Such a camera is rather a strange object, however, and will not be treated in detail in this book. We may compare the properties of affine and non-affine infinite cameras:

	Affine camera	Non-affine camera
Camera centre on $\boldsymbol{\pi}_\infty$	yes	yes
Principal plane is $\boldsymbol{\pi}_\infty$	yes	no
Image of points on $\boldsymbol{\pi}_\infty$ on $\mathbf{l}_\infty$	yes	no in general

In both cases the camera centre is the direction of projection. Furthermore, in the case of an affine camera all non-infinite points are in front of the camera. For a non-affine camera space is partitioned into two sets of points by the principal plane.

A general camera at infinity could arise from a perspective image of an image produced by an affine camera. This imaging process corresponds to left-multiplying the

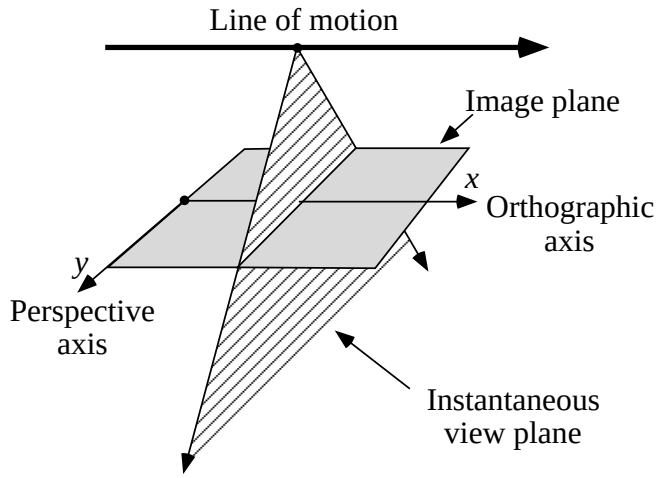


Fig. 6.9. *Acquisition geometry of a pushbroom camera.*

affine camera by a general 3×3 matrix representing the planar homography. The resulting 3×4 matrix is still a camera at infinity, but it does not have the affine form, since parallel lines in the world will in general appear as converging lines in the image.

6.4 Other camera models

6.4.1 Pushbroom cameras

The Linear Pushbroom (LP) camera is an abstraction of a type of sensor common in satellites, for instance the SPOT sensor. In such a camera, a linear sensor array is used to capture a single line of imagery at a time. As the sensor moves the sensor plane sweeps out a region of space (hence the name pushbroom), capturing the image a single line at a time. The second dimension of the image is provided by the motion of the sensor. In the linear pushbroom model, the sensor is assumed to move in a straight line at constant velocity with respect to the ground. In addition, one assumes that the orientation of the sensor array with respect to the direction of travel is constant. In the direction of the sensor, the image is effectively a perspective image, whereas in the direction of the sensor motion it is an orthographic projection. The geometry of the LP camera is illustrated in figure 6.9. It turns out that the mapping from object space into the image may be described by a 3×4 camera matrix, just as with a general projective camera. However, the way in which this matrix is used is somewhat different.

- *Let* $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)^\mathsf{T}$ *be an object point, and let* $\mathtt{P}$ *be the camera matrix of the LP camera. Suppose that* $\mathtt{P}\mathbf{X} = (x, y, w)^\mathsf{T}$. *Then the corresponding image point (represented as an inhomogeneous 2-vector) is* $(x, y/w)^\mathsf{T}$.

One must compare this with the projective camera mapping. In that case the point represented by $(x, y, w)^\mathsf{T}$ is $(x/w, y/w)^\mathsf{T}$. Note the difference that in the LP case, the coordinate x is not divided by the factor w to get the image coordinate. In this formula, the x-axis in the image is the direction of the sensor motion, whereas the y-axis is in the direction of the linear sensor array. The camera has 11 degrees of freedom.

Another way of writing the formula for LP projection is

$$\tilde{x} = x = \mathtt{P}^{1\mathsf{T}}\mathbf{X} \quad \tilde{y} = y/z = \frac{\mathtt{P}^{2\mathsf{T}}\mathbf{X}}{\mathtt{P}^{3\mathsf{T}}\mathbf{X}} \tag{6.27}$$

where $(\tilde{x}, \tilde{y})$ is the image point.

Note that the $\tilde{y}$-coordinate behaves projectively, whereas the $\tilde{x}$ is obtained by orthogonal projection of the point $\mathbf{X}$ on the direction perpendicular to the plane $\mathtt{P}^1$. The vector $\mathtt{P}^1$ represents the sweep plane of the camera at time $t = 0$ – that is the moment when the line with coordinates $\tilde{x} = 0$ is captured.

Mapping of lines. One of the novel features of the LP camera is that straight lines in space are not mapped to straight lines in the image (they are mapped to straight lines in the case of a projective camera – see section 8.1.2). The set of points $\mathbf{X}$ lying on a 3D line may be written as $\mathbf{X}_0 + \alpha\mathbf{D}$, where $\mathbf{X}_0 = (\mathtt{X}, \mathtt{Y}, \mathtt{Z}, 1)^\mathsf{T}$ is a point on the line and $\mathbf{D} = (D_\mathtt{X}, D_\mathtt{Y}, D_\mathtt{Z}, 0)^\mathsf{T}$ is the intersection of this line with the plane at infinity. In this case, we compute from (6.27)

$$\begin{aligned} \tilde{x} &= \mathbf{P}^{1\mathsf{T}}(\mathbf{X}_0 + t\mathbf{D}) \\ \tilde{y} &= \frac{\mathbf{P}^{2\mathsf{T}}(\mathbf{X}_0 + t\mathbf{D})}{\mathbf{P}^{3\mathsf{T}}(\mathbf{X}_0 + t\mathbf{D})}. \end{aligned}$$

This may be written as a pair of equations $\tilde{x} = a+bt$ and $(c+dt)\tilde{y} = e+ft$. Eliminating t from these equations leads to an equation of the form $\alpha\tilde{x}\tilde{y} + \beta\tilde{x} + \gamma\tilde{y} + \delta = 0$, which is the equation of a hyperbola in the image plane, asymptotic in one direction to the line $\alpha\tilde{x} + \gamma = 0$, and in the other direction to the line $\alpha\tilde{y} + \beta = 0$. A hyperbola is made up of two curves. However, only one of the curves making up the image of a line actually appears in the image – the other part of the hyperbola corresponds to points lying behind the camera.

6.4.2 Line cameras

This chapter has dealt with the central projection of 3-space onto a 2D image. An analogous development can be given for the central projection of a plane onto a 1D line contained in the plane. See figure 22.1(*p*535). The camera model for this geometry is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{bmatrix} \begin{bmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \end{bmatrix} = \mathtt{P}_{2\times 3}\mathbf{X}$$

which is a linear mapping from homogeneous representation of the plane to a homogeneous representation of the line. The camera has 5 degrees of freedom. Again the null-space, $\mathbf{c}$, of the $\mathtt{P}_{2\times 3}$ projection matrix is the camera centre, and the matrix can be decomposed in a similar manner to the finite projective camera (6.11–*p*157) as

$$\mathtt{P}_{2\times 3} = \mathtt{K}_{2\times 2}\mathtt{R}_{2\times 2}[\mathtt{I}_{2\times 2} \mid -\tilde{\mathbf{c}}]$$

where $\tilde{\mathbf{c}}$ is the inhomogeneous 2-vector representing the centre (2 dof), $\mathtt{R}_{2\times2}$ is a rotation matrix (1 dof), and

$$\mathtt{K}_{2\times2} = \begin{bmatrix} \alpha_x & x_0 \\ & 1 \end{bmatrix}$$

the internal calibration matrix (2 dof).

6.5 Closure

This chapter has covered camera models, their taxonomy and anatomy. The subsequent chapters cover the estimation of cameras from a set of world to image correspondences, and the action of a camera on various geometric objects such as lines and quadrics. Vanishing points and vanishing lines are also described in more detail in chapter 8.

6.5.1 The literature

[Aloimonos-90] defined a hierarchy of camera models including para-perspective. Mundy and Zisserman [Mundy-92] generalized this with the affine camera. Faugeras developed properties of the projective camera in his textbook [Faugeras-93]. Further details on the linear pushbroom camera are given in [Gupta-97], and on the 2D camera in [Quan-97b].

6.5.2 Notes and exercises

(i) Let I_0 be a projective image, and I_1 be an image of I_0 (an image of an image). Let the composite image be denoted by I'. Show that the apparent camera centre of I' is the same as that of I_0. Speculate on how this explains why a portrait's eyes "follow you round the room." Verify on the other hand that all other parameters of I' and I_0 may be different.

(ii) Show that the ray back-projected from an image point $\mathbf{x}$ under a projective camera $\mathtt{P}$ (as in (6.14–*p*162)) may be written as

$$\mathtt{L}^* = \mathtt{P}^\mathsf{T}[\mathbf{x}]_\times \mathtt{P} \tag{6.28}$$

where $\mathtt{L}^*$ is the dual Plücker representation of a line (3.9–*p*71).

(iii) **The affine camera.**

(a) Show that the affine camera is the most general linear mapping on homogeneous coordinates that maps parallel world lines to parallel image lines. To do this consider the projection of points on $\boldsymbol{\pi}_\infty$, and show that only if $\mathtt{P}$ has the affine form will they map to points at infinity in the image.

(b) Show that for parallel lines mapped by an affine camera the ratio of lengths on line segments is an invariant. What other invariants are there under an affine camera?

(iv) **The rational polynomial camera** is a general camera model, used extensively

in the satellite surveillance community. Image coordinates are defined by the ratios

$$x = N_x(\mathbf{X})/D_x(\mathbf{X}) \quad y = N_y(\mathbf{X})/D_y(\mathbf{X})$$

where the functions N_x, D_x, N_y, D_y are homogeneous *cubic* polynomials in the 3-space point $\mathbf{X}$. Each cubic has 20 coefficients, so that overall the camera has 78 degrees of freedom. All of the cameras surveyed in this chapter (projective, affine, pushbroom) are special cases of the rational polynomial camera. Its disadvantage is that it is severely over-parametrized for these cases. More details are given in Hartley and Saxena [Hartley-97e].

(v) A finite projective camera (6.11–*p*157) $\mathtt{P}$ may be transformed to an orthographic camera (6.22) by applying a 4×4 homography $\mathtt{H}$ on the right such that

$$\mathtt{PH} = \mathtt{KR}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]\mathtt{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathtt{P}_{\mathrm{orthog}} \ .$$

(the last row of $\mathtt{H}$ is chosen so that $\mathtt{H}$ has rank 4). Then since

$$\mathbf{x} = \mathtt{P}(\mathtt{H}\mathtt{H}^{-1})\mathbf{X} = (\mathtt{P}\mathtt{H})(\mathtt{H}^{-1}\mathbf{X}) = \mathtt{P}_{\mathrm{orthog}}\mathbf{X}'$$

imaging under $\mathtt{P}$ is equivalent to first transforming the 3-space points $\mathbf{X}$ to $\mathbf{X}' = \mathtt{H}^{-1}\mathbf{X}$ and then applying an orthographic projection. Thus the action of any camera may be considered as a projective transformation of 3-space followed by orthographic projection.

7

Computation of the Camera Matrix P

This chapter describes numerical methods for estimating the camera projection matrix from corresponding 3-space and image entities. This computation of the camera matrix is known as *resectioning*. The simplest such correspondence is that between a 3D point $\mathbf{X}$ and its image $\mathbf{x}$ under the unknown camera mapping. Given sufficiently many correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ the camera matrix $\mathtt{P}$ may be determined. Similarly, $\mathtt{P}$ may be determined from sufficiently many corresponding world and image lines.

If additional constraints apply to the matrix $\mathtt{P}$, such as that the pixels are square, then a *restricted* camera matrix subject to these constraints may be estimated from world to image correspondences.

Throughout this book it is assumed that the map from 3-space to the image is linear. This assumption is invalid if there is lens distortion. The topic of radial lens distortion correction is dealt with in this chapter.

The internal parameters $\mathtt{K}$ of the camera may be extracted from the matrix $\mathtt{P}$ by the decomposition of section 6.2.4. Alternatively, the internal parameters can be computed directly, without necessitating estimating $\mathtt{P}$, by the methods of chapter 8.

7.1 Basic equations

We assume a number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$ are given. We are required to find a camera matrix $\mathtt{P}$, namely a 3×4 matrix such that $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$ for all i. The similarity of this problem with that of computing a 2D projective transformation $\mathtt{H}$, treated in chapter 4, is evident. The only difference is the dimension of the problem. In the 2D case the matrix $\mathtt{H}$ has dimension 3×3, whereas in the present case, $\mathtt{P}$ is a 3×4 matrix. As one may expect, much of the material from chapter 4 applies almost unchanged to the present case.

As in section 4.1(*p*88) for each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ we derive a relationship

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \\ -y_i\mathbf{X}_i^\mathsf{T} & x_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}. \tag{7.1}$$

where each $\mathbf{P}^{i\mathsf{T}}$ is a 4-vector, the i-th row of $\mathtt{P}$. Alternatively, one may choose to use

only the first two equations:

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \tag{7.2}$$

since the three equations of (7.1) are linearly dependent. From a set of n point correspondences, we obtain a $2n \times 12$ matrix $\mathtt{A}$ by stacking up the equations (7.2) for each correspondence. The projection matrix $\mathtt{P}$ is computed by solving the set of equations $\mathtt{A}\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ is the vector containing the entries of the matrix $\mathtt{P}$.

Minimal solution. Since the matrix $\mathtt{P}$ has 12 entries, and (ignoring scale) 11 degrees of freedom, it is necessary to have 11 equations to solve for $\mathtt{P}$. Since each point correspondence leads to two equations, at a minimum $5\frac{1}{2}$ such correspondences are required to solve for $\mathtt{P}$. The $\frac{1}{2}$ indicates that only one of the equations is used from the sixth point, so one needs only to know the x-coordinate (or alternatively the y-coordinate) of the sixth image point.

Given this minimum number of correspondences, the solution is exact, i.e. the space points are projected exactly onto their measured images. The solution is obtained by solving $\mathtt{A}\mathbf{p} = \mathbf{0}$ where $\mathtt{A}$ is an 11×12 matrix in this case. In general $\mathtt{A}$ will have rank 11, and the solution vector $\mathbf{p}$ is the 1-dimensional right null-space of $\mathtt{A}$.

Over-determined solution. If the data is not exact, because of noise in the point coordinates, and $n \geq 6$ point correspondences are given, then there will not be an exact solution to the equations $\mathtt{A}\mathbf{p} = \mathbf{0}$. As in the estimation of a homography a solution for $\mathtt{P}$ may be obtained by minimizing an algebraic or geometric error.

In the case of algebraic error the approach is to minimize $\|\mathtt{A}\mathbf{p}\|$ subject to some normalization constraint. Possible constraints are

(i) $\|\mathbf{p}\| = 1$;
(ii) $\|\hat{\mathbf{p}}^3\| = 1$, where $\hat{\mathbf{p}}^3$ is the vector $(p_{31}, p_{32}, p_{33})^\mathsf{T}$, namely the first three entries in the last row of $\mathtt{P}$.

The first of these is preferred for routine use and will be used for the moment. We will return to the second normalization constraint in section 7.2.1. In either case, the residual $\mathtt{A}\mathbf{p}$ is known as the *algebraic error*. Using these equations, the complete DLT algorithm for computation of the camera matrix $\mathtt{P}$ proceeds in the same manner as that for $\mathtt{H}$ given in algorithm 4.1(*p*91).

Degenerate configurations. Analysis of the degenerate configurations for estimation of $\mathtt{P}$ is rather more involved than in the case of the 2D homography. There are two types of configurations in which ambiguous solutions exist for $\mathtt{P}$. These configurations will be investigated in detail in chapter 22. The most important critical configurations are as follows:

(i) The camera and points all lie on a twisted cubic.

(ii) The points all lie on the union of a plane and a single straight line containing the camera centre.

For such configurations, the camera cannot be obtained uniquely from the images of the points. Instead, it may move arbitrarily along the twisted cubic, or straight line respectively. If data is close to a degenerate configuration then a poor estimate for P is obtained. For example, if the camera is distant from a scene with low relief, such as a near-nadir aerial view, then this situation is close to the planar degeneracy.

Data normalization. It is important to carry out some sort of data normalization just as in the 2D homography estimation case. The points $\mathbf{x}_i$ in the image are appropriately normalized in the same way as before. Namely the points should be translated so that their centroid is at the origin, and scaled so that their RMS (root-mean-squared) distance from the origin is $\sqrt{2}$. What normalization should be applied to the 3D points $\mathbf{X}_i$ is a little more problematical. In the case where the variation in depth of the points from the camera is relatively slight it makes sense to carry out the same sort of normalization. Thus, the centroid of the points is translated to the origin, and their coordinates are scaled so that the RMS distance from the origin is $\sqrt{3}$ (so that the "average" point has coordinates of magnitude $(1, 1, 1, 1)^\mathsf{T}$). This approach is suitable for a compact distribution of points, such as those on the calibration object of figure 7.1.

In the case where there are some points that lie at a great distance from the camera, the previous normalization technique does not work well. For instance, if there are points close to the camera, as well as points that lie at infinity (which are imaged as vanishing points) or close to infinity, as may occur in oblique views of terrain, then it is not possible or reasonable to translate the points so that their centroid is at the origin. The normalization method described in exercise (iii) on page 128 would be more appropriately used in such a case, though this has not been thoroughly tested.

With appropriate normalization the estimate of P is carried out in the same manner as algorithm 4.2(*p*109) for H.

Line correspondences. It is a simple matter to extend the DLT algorithm to take account of line correspondences as well. A line in 3D may be represented by two points $\mathbf{X}_0$ and $\mathbf{X}_1$ through which the line passes. Now, according to result 8.2(*p*197) the plane formed by back-projecting from the image line $\mathbf{l}$ is equal to $\mathtt{P}^\mathsf{T}\mathbf{l}$. The condition that the point $\mathbf{X}_j$ lies on this plane is then

$$\mathbf{l}^\mathsf{T}\mathtt{P}\mathbf{X}_j = 0 \quad \text{for } j = 0, 1. \tag{7.3}$$

Each choice of j gives a single linear equation in the entries of the matrix P, so two equations are obtained for each 3D to 2D line correspondence. These equations, being linear in the entries of P, may be added to the equations (7.1) obtained from point correspondences and a solution to the composite equation set may be computed.

7.2 Geometric error

As in the case of 2D homographies (chapter 4), one may define geometric error. Suppose for the moment that world points $\mathbf{X}_i$ are known far more accurately than the

Objective

Given $n \geq 6$ world to image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood estimate of the camera projection matrix $\mathtt{P}$, i.e. the $\mathtt{P}$ which minimizes $\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$.

Algorithm

(i) **Linear solution.** Compute an initial estimate of $\mathtt{P}$ using a linear method such as algorithm 4.2(*p*109):

  (a) **Normalization:** Use a similarity transformation $\mathtt{T}$ to normalize the image points, and a second similarity transformation $\mathtt{U}$ to normalize the space points. Suppose the normalized image points are $\tilde{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$, and the normalized space points are $\tilde{\mathbf{X}}_i = \mathtt{U}\mathbf{X}_i$.

  (b) **DLT:** Form the $2n \times 12$ matrix $\mathtt{A}$ by stacking the equations (7.2) generated by each correspondence $\tilde{\mathbf{X}}_i \leftrightarrow \tilde{\mathbf{x}}_i$. Write $\mathbf{p}$ for the vector containing the entries of the matrix $\tilde{\mathtt{P}}$. A solution of $\mathtt{A}\mathbf{p} = \mathbf{0}$, subject to $\|\mathbf{p}\| = 1$, is obtained from the unit singular vector of $\mathtt{A}$ corresponding to the smallest singular value.

(ii) **Minimize geometric error.** Using the linear estimate as a starting point minimize the geometric error (7.4):

$$\sum_i d(\tilde{\mathbf{x}}_i, \tilde{\mathtt{P}}\tilde{\mathbf{X}}_i)^2$$

over $\tilde{\mathtt{P}}$, using an iterative algorithm such as Levenberg–Marquardt.

(iii) **Denormalization.** The camera matrix for the original (unnormalized) coordinates is obtained from $\tilde{\mathtt{P}}$ as

$$\mathtt{P} = \mathtt{T}^{-1}\tilde{\mathtt{P}}\mathtt{U}.$$

Algorithm 7.1. *The Gold Standard algorithm for estimating* $\mathtt{P}$ *from world to image point correspondences in the case that the world points are very accurately known.*

measured image points. For example the points $\mathbf{X}_i$ might arise from an accurately machined calibration object. Then the geometric error in the image is

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$$

where $\mathbf{x}_i$ is the measured point and $\hat{\mathbf{x}}_i$ is the point $\mathtt{P}\mathbf{X}_i$, i.e. the point which is the exact image of $\mathbf{X}_i$ under $\mathtt{P}$. If the measurement errors are Gaussian then the solution of

$$\min_{\mathtt{P}} \sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2 \tag{7.4}$$

is the Maximum Likelihood estimate of $\mathtt{P}$.

Just as in the 2D homography case, minimizing geometric error requires the use of iterative techniques, such as Levenberg–Marquardt. A parametrization of $\mathtt{P}$ is required, and the vector of matrix elements $\mathbf{p}$ provides this. The DLT solution, or a minimal solution, may be used as a starting point for the iterative minimization. The complete Gold Standard algorithm is summarized in algorithm 7.1.

Example 7.1. Camera estimation from a calibration object

We will compare the DLT algorithm with the Gold Standard algorithm 7.1 for data

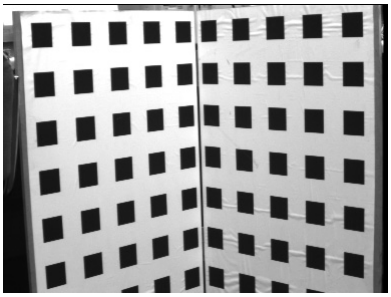


Fig. 7.1. *An image of a typical calibration object. The black and white checkerboard pattern (a "Tsai grid") is designed to enable the positions of the corners of the imaged squares to be obtained to high accuracy. A total of 197 points were identified and used to calibrate the camera in the examples of this chapter.*

	f_y	f_x/f_y	skew	x_0	y_0	residual
linear	1673.3	1.0063	1.39	379.96	305.78	0.365
iterative	1675.5	1.0063	1.43	379.79	305.25	0.364

Table 7.1. *DLT and Gold Standard calibration.*

from the calibration object shown in figure 7.1. The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:

(i) Canny edge detection [Canny-86].
(ii) Straight line fitting to the detected linked edges.
(iii) Intersecting the lines to obtain the imaged corners.

If sufficient care is taken the points $\mathbf{x}_i$ are obtained to a localization accuracy of far better than $1/10$ of a pixel. A rule of thumb is that for a good estimation the number of constraints (point measurements) should exceed the number of unknowns (the 11 camera parameters) by a factor of five. This means that at least 28 points should be used.

Table 7.1 shows the calibration results obtained by using the linear DLT method and the Gold Standard method. Note that the improvement achieved using the Gold Standard algorithm is very slight. The difference of residual of one thousandth of a pixel is insignificant. △

Errors in the world points

It may be the case that world points are not measured with "infinite" accuracy. In this case one may choose to estimate P by minimizing a 3D geometric error, or an image geometric error, or both.

If only errors in the world points are considered then the 3D geometric error is defined as

$$\sum_i d(\mathbf{X}_i, \widehat{\mathbf{X}}_i)^2$$

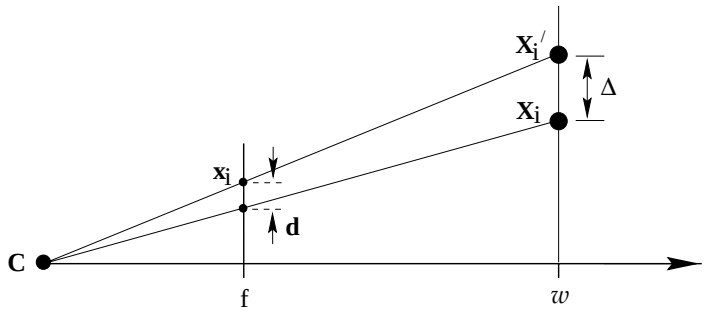


Fig. 7.2. *The DLT algorithm minimizes the sum of squares of geometric distance Δ between the point $\mathbf{X}_i$ and the point $\mathbf{X}_i'$ mapping exactly onto $\mathbf{x}_i$ and lying in the plane through $\mathbf{X}_i$ parallel to the principal plane of the camera. A short calculation shows that $wd = f\Delta$.*

where $\widehat{\mathbf{X}}_i$ is the closest point in space to $\mathbf{X}_i$ that maps exactly onto $\mathbf{x}_i$ via $\mathbf{x}_i = \mathtt{P}\widehat{\mathbf{X}}_i$.

More generally, if errors in both the world and image points are considered, then a weighted sum of world and image errors is minimized. As in the 2D homography case, this requires that one augment the set of parameters by including parameters $\widehat{\mathbf{X}}_i$, the estimated 3D points. One minimizes

$$\sum_{i=1}^{n} d_{\text{Mah}}(\mathbf{x}_i, \mathtt{P}\widehat{\mathbf{X}}_i)^2 + d_{\text{Mah}}(\mathbf{X}_i, \widehat{\mathbf{X}}_i)^2$$

where d_{Mah} represents Mahalanobis distance with respect to the known error covariance matrices for each of the measurements $\mathbf{x}_i$ and $\mathbf{X}_i$. In the simplest case, the Mahalanobis distance is simply a weighted geometric distance, where the weights are chosen to reflect the relative accuracy of measurements of the image and 3D points, and also the fact that image and world points are typically measured in different units.

7.2.1 Geometric interpretation of algebraic error

Suppose all the points $\mathbf{X}_i$ in the DLT algorithm are normalized such that $\mathbf{X}_i = (\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i, 1)^\mathsf{T}$, and $\mathbf{x}_i = (x_i, y_i, 1)^\mathsf{T}$. In this case, it was seen in section 4.2.4-(*p*95) that the quantity being minimized by the DLT algorithm is $\sum_i (\hat{w}_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i))^2$, where $\hat{w}_i(\hat{x}_i, \hat{y}_i, 1)^\mathsf{T} = \mathtt{P}\mathbf{X}_i$. However, according to (6.15–*p*162),

$$\hat{w}_i = \pm \|\hat{\mathbf{p}}^3\| \text{ depth}(\mathbf{X}; \mathtt{P}) \ .$$

Thus, the value $\hat{w}_i$ may be interpreted as the depth of the point $\mathbf{X}_i$ from the camera in the direction along the principal ray, provided the camera is normalized so that $\|\hat{\mathbf{p}}^3\|^2 = p_{31}^2 + p_{32}^2 + p_{33}^2 = 1$. Referring to figure 7.2 one sees that $\hat{w}_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)$ is proportional to $f d(\mathbf{X}', \mathbf{X})$, where f is the focal length and $\mathbf{X}_i'$ is a point mapping to $\mathbf{x}_i$ and lying in a plane through $\mathbf{X}_i$ parallel to the principal plane of the camera. Thus, the algebraic error being minimized is equal to $f \sum_i d(\mathbf{X}_i, \mathbf{X}_i')^2$.

The distance $d(\mathbf{X}_i, \mathbf{X}_i')$ is the correction that needs to be made to the measured 3D points in order to correspond precisely with the measured image points $\mathbf{x}_i$. The restriction is that the correction must be made in the direction perpendicular to the principal axis of the camera. Because of this restriction, the point $\mathbf{X}_i'$ is not the same as the closest point $\widehat{\mathbf{X}}_i$ to $\mathbf{X}_i$ that maps to $\mathbf{x}_i$. However, for points $\mathbf{X}_i$ not too far from the principal

ray of the camera, the distance $d(\mathbf{X}_i, \mathbf{X}_i')$ is a reasonable approximation to the distance $d(\mathbf{X}_i, \widehat{\mathbf{X}}_i)$. The DLT slightly weights the points farther away from the principal ray by minimizing the squared sum of $d(\mathbf{X}_i, \mathbf{X}_i')$, which is slightly larger than $d(\mathbf{X}_i, \widehat{\mathbf{X}}_i)$. In addition, the presence of the focal length f in the expression for algebraic error suggests that the DLT algorithm will be biased towards minimizing focal length at a cost of a slight increase in 3D geometric error.

Transformation invariance. We have just seen that by minimizing $\|\mathtt{A}\mathbf{p}\|$ subject to the constraint $\|\hat{\mathbf{p}}^3\| = 1$ one may interpret the solution in terms of minimizing 3D geometric distances. Such an interpretation is not affected by similarity transformations in either 3D space or the image space. Thus, one is led to expect that carrying out translation and scaling of the data, either in the image or in 3D point coordinates, will not have any effect on the solutions. This is indeed the case as may be shown using the arguments of section 4.4.2(*p*105).

7.2.2 Estimation of an affine camera

The methods developed above for the projective cameras can be applied directly to affine cameras. An affine camera is one for which the projection matrix has last row $(0, 0, 0, 1)$. In the DLT estimation of the camera in this case one minimizes $\|\mathtt{A}\mathbf{p}\|$ subject to this condition on the last row of P. As in the case of computing 2D affine transformations, for affine cameras, algebraic error and geometric image error are equal. This means that geometric image distances may be minimized by a linear algorithm.

Suppose as above that all the points $\mathbf{X}_i$ are normalized such that $\mathbf{X}_i = (\mathrm{X}_i, \mathrm{Y}_i, \mathrm{Z}_i, 1)^\mathsf{T}$, and $\mathbf{x}_i = (x_i, y_i, 1)^\mathsf{T}$, and also that the last row of P has the affine form. Then (7.2) for a single correspondence reduces to

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -\mathbf{X}_i^\mathsf{T} \\ \mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \end{pmatrix} + \begin{pmatrix} y_i \\ -x_i \end{pmatrix} = \mathbf{0} \tag{7.5}$$

which shows that the squared algebraic error in this case equals the squared geometric error

$$\|\mathtt{A}\mathbf{p}\|^2 = \sum_i \left(x_i - \mathbf{P}^{1\mathsf{T}}\mathbf{X}_i\right)^2 + \left(y_i - \mathbf{P}^{2\mathsf{T}}\mathbf{X}_i\right)^2 = \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2.$$

This result may also be seen geometrically by comparison of figure 6.8(*p*170) and figure 7.2.

A linear estimation algorithm for an affine camera which minimizes geometric error is given in algorithm 7.2. Under the assumption of Gaussian measurement errors this is the Maximum Likelihood estimate of $\mathtt{P}_\mathrm{A}$.

7.3 Restricted camera estimation

The DLT algorithm, as it has so far been described, computes a general projective camera matrix P from a set of 3D to 2D point correspondences. The matrix P with centre at a finite point may be decomposed as $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathbf{C}}]$ where R is a 3×3

Objective

Given $n \geq 4$ world to image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood Estimate of the affine camera projection matrix $\mathtt{P}_\mathrm{A}$, i.e. the camera $\mathtt{P}$ which minimizes $\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$ subject to the affine constraint $\mathbf{P}^{3\mathsf{T}} = (0, 0, 0, 1)$.

Algorithm

(i) **Normalization:** Use a similarity transformation $\mathtt{T}$ to normalize the image points, and a second similarity transformation $\mathtt{U}$ to normalize the space points. Suppose the normalized image points are $\tilde{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$, and the normalized space points are $\tilde{\mathbf{X}}_i = \mathtt{U}\mathbf{X}_i$, with unit last component.

(ii) Each correspondence $\tilde{\mathbf{X}}_i \leftrightarrow \tilde{\mathbf{x}}_i$ contributes (from (7.5)) equations

$$\begin{bmatrix} \tilde{\mathbf{X}}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} & \tilde{\mathbf{X}}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{P}}^1 \\ \tilde{\mathbf{P}}^2 \end{pmatrix} = \begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \end{pmatrix}$$

which are stacked into a $2n \times 8$ matrix equation $\mathtt{A}_8\mathbf{p}_8 = \mathbf{b}$, where $\mathbf{p}_8$ is the 8-vector containing the first two rows of $\tilde{\mathtt{P}}_\mathrm{A}$.

(iii) The solution is obtained by the pseudo-inverse of $\mathtt{A}_8$ (see section A5.2(*p*590))

$$\mathbf{p}_8 = \mathtt{A}_8^+\mathbf{b}$$

and $\tilde{\mathbf{P}}^{3\mathsf{T}} = (0, 0, 0, 1)$.

(iv) **Denormalization:** The camera matrix for the original (unnormalized) coordinates is obtained from $\tilde{\mathtt{P}}_\mathrm{A}$ as

$$\mathtt{P}_\mathrm{A} = \mathtt{T}^{-1}\tilde{\mathtt{P}}_\mathrm{A}\mathtt{U}$$

Algorithm 7.2. *The Gold Standard Algorithm for estimating an affine camera matrix* $\mathtt{P}_\mathrm{A}$ *from world to image correspondences.*

rotation matrix and $\mathtt{K}$ has the form (6.10–*p*157):

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} . \tag{7.6}$$

The non-zero entries of $\mathtt{K}$ are geometrically meaningful quantities, the internal calibration parameters of $\mathtt{P}$. One may wish to find the best-fit camera matrix $\mathtt{P}$ subject to restrictive conditions on the camera parameters. Common assumptions are

(i) The skew s is zero.
(ii) The pixels are square: $\alpha_x = \alpha_y$.
(iii) The principal point (x_0, y_0) is known.
(iv) The complete camera calibration matrix $\mathtt{K}$ is known.

In some cases it is possible to estimate a restricted camera matrix with a linear algorithm (see the exercises at the end of the chapter).

As an example of restricted estimation, suppose that we wish to find the best pinhole camera model (that is projective camera with $s = 0$ and $\alpha_x = \alpha_y$) that fits a set of point measurements. This problem may be solved by minimizing either geometric or algebraic error, as will be discussed next.

Minimizing geometric error. To minimize geometric error, one selects a set of parameters that characterize the camera matrix to be computed. For instance, suppose we wish to enforce the constraints $s = 0$ and $\alpha_x = \alpha_y$. One can parametrize the camera matrix using the remaining 9 parameters. These are x_0, y_0, α, plus 6 parameters representing the orientation $\mathtt{R}$ and location $\widetilde{\mathbf{C}}$ of the camera. Let this set of parameters be denoted collectively by $\mathbf{q}$. The camera matrix $\mathtt{P}$ may then be explicitly computed in terms of the parameters.

The geometric error may then be minimized with respect to the set of parameters using iterative minimization (such as Levenberg–Marquardt). Note that in the case of minimization of image error only, the size of the minimization problem is $9 \times 2n$ (supposing 9 unknown camera parameters). In other words the LM minimization is minimizing a function $f : \mathbb{R}^9 \to \mathbb{R}^{2n}$. In the case of minimization of 3D and 2D error, the function f is from $\mathbb{R}^{3n+9} \to \mathbb{R}^{5n}$, since the 3D points must be included among the measurements and minimization also includes estimation of the true positions of the 3D points.

Minimizing algebraic error. It is possible to minimize algebraic error instead, in which case the iterative minimization problem becomes much smaller, as will be explained next. Consider the parametrization map taking a set of parameters $\mathbf{q}$ to the corresponding camera matrix $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathbf{C}}]$. Let this map be denoted by g. Effectively, one has a map $\mathbf{p} = g(\mathbf{q})$, where $\mathbf{p}$ is the vector of entries of the matrix $\mathtt{P}$. Minimizing algebraic error over all point matches is equivalent to minimizing $\|\mathtt{A}g(\mathbf{q})\|$.

The reduced measurement matrix. In general, the $2n \times 12$ matrix $\mathtt{A}$ may have a very large number of rows. It is possible to replace $\mathtt{A}$ by a square 12×12 matrix $\hat{\mathtt{A}}$ such that $\|\mathtt{A}\mathbf{p}\| = \mathbf{p}^\mathsf{T}\mathtt{A}^\mathsf{T}\mathtt{A}\mathbf{p} = \|\hat{\mathtt{A}}\mathbf{p}\|$ for any vector $\mathbf{p}$. Such a matrix $\hat{\mathtt{A}}$ is called a *reduced measurement matrix*. One way to do this is using the Singular Value Decomposition (SVD). Let $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}$ be the SVD of $\mathtt{A}$, and define $\hat{\mathtt{A}} = \mathtt{D}\mathtt{V}^\mathsf{T}$. Then

$$\mathtt{A}^\mathsf{T}\mathtt{A} = (\mathtt{V}\mathtt{D}\mathtt{U}^\mathsf{T})(\mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}) = (\mathtt{V}\mathtt{D})(\mathtt{D}\mathtt{V}^\mathsf{T}) = \hat{\mathtt{A}}^\mathsf{T}\hat{\mathtt{A}}$$

as required. Another way of obtaining $\hat{\mathtt{A}}$ is to use the QR decomposition $\mathtt{A} = \mathtt{Q}\hat{\mathtt{A}}$, where $\mathtt{Q}$ has orthogonal columns and $\hat{\mathtt{A}}$ is upper triangular and square.

Note that the mapping $\mathbf{q} \mapsto \hat{\mathtt{A}}g(\mathbf{q})$ is a mapping from $\mathbb{R}^9$ to $\mathbb{R}^{12}$. This is a simple parameter-minimization problem that may be solved using the Levenberg–Marquardt method. The important point to note is the following:

- *Given a set of n world to image correspondences, $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, the problem of finding a constrained camera matrix $\mathtt{P}$ that minimizes the sum of algebraic distances $\sum_i d_{alg}(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$ reduces to the minimization of a function $\mathbb{R}^9 \to \mathbb{R}^{12}$, independent of the number n of correspondences.*

Minimization of $\|\hat{\mathtt{A}}g(\mathbf{q})\|$ takes place over all values of the parameters $\mathbf{q}$. Note that if $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathbf{C}}]$ with $\mathtt{K}$ as in (7.6) then $\mathtt{P}$ satisfies the condition $p_{31}^2 + p_{32}^2 + p_{33}^2 = 1$, since these entries are the same as the last row of the rotation matrix $\mathtt{R}$. Thus, minimizing $\mathtt{A}g(\mathbf{q})$ will lead to a matrix $\mathtt{P}$ satisfying the constraints $s = 0$ and $\alpha_x = \alpha_y$ and scaled

such that $p_{31}^2 + p_{32}^2 + p_{33}^2 = 1$, and which in addition minimizes the algebraic error for all point correspondences.

Initialization. One way of finding camera parameters to initialize the iteration is as follows.

(i) Use a linear algorithm such as DLT to find an initial camera matrix.
(ii) Clamp fixed parameters to their desired values (for instance set $s = 0$ and set $\alpha_x = \alpha_y$ to the average of their values obtained using DLT).
(iii) Set variable parameters to their values obtained by decomposition of the initial camera matrix (see section 6.2.4).

Ideally, the assumed values of the fixed parameters will be close to the values obtained by the DLT. However, in practice this is not always the case. Then altering these parameters to their desired values results in an incorrect initial camera matrix that may lead to large residuals, and difficulty in converging. A method which works better in practice is to use *soft* constraints by adding extra terms to the cost function. Thus, for the case where $s = 0$ and $\alpha_x = \alpha_y$, one adds extra terms $ws^2 + w(\alpha_x - \alpha_y)^2$ to the cost function. In the case of geometric image error, the cost function becomes

$$\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2 + ws^2 + w(\alpha_x - \alpha_y)^2 \quad .$$

One begins with the values of the parameters estimated using the DLT. The weights begin with low values and are increased at each iteration of the estimation procedure. Thus, the values of s and the aspect ratio are drawn gently to their desired values. Finally they may be clamped to their desired values for a final estimation.

Exterior orientation. Suppose that all the internal parameters of the camera are known, then all that remains to be determined are the position and orientation (or *pose*) of the camera. This is the "exterior orientation" problem, which is important in the analysis of calibrated systems.

To compute the exterior orientation a configuration with accurately known position in a world coordinate frame is imaged. The pose of the camera is then sought. Such a situation arises in hand–eye calibration for robotic systems, where the position of the camera is required, and also in model-based recognition using alignment where the position of an object relative to the camera is required.

There are six parameters that must be determined, three for the orientation and three for the position. As each world to image point correspondence generates two constraints it would be expected that three points are sufficient. This is indeed the case, and the resulting non-linear equations have four solutions in general.

Experimental evaluation

Results of constrained estimation for the calibration grid of example 7.1 are given in table 7.2.

Both the algebraic and geometric minimization involve an iterative minimization

	f_y	f_x/f_y	skew	x_0	y_0	residual
algebraic	1633.4	1.0	0.0	371.21	293.63	0.601
geometric	1637.2	1.0	0.0	371.32	293.69	0.601

Table 7.2. *Calibration for a restricted camera matrix.*

over 9 parameters. However, the algebraic method is far quicker, since it minimizes only 12 errors, instead of $2n = 396$ in the geometric minimization. Note that fixing skew and aspect ratio has altered the values of the other parameters (compare table 7.1) and increased the residual.

Covariance estimation. The techniques of covariance estimation and propagation of the errors into an image may be handled in just the same way as in the 2D homography case (chapter 5). Similarly, the minimum expected residual error may be computed as in result 5.2(*p*136). Assuming that all errors are in the image measurements, the expected ML residual error is equal to

$$\epsilon_{\text{res}} = \sigma(1 - d/2n)^{1/2} \ .$$

where d is the number of camera parameters being fitted (11 for a full pinhole camera model). This formula may also be used to estimate the accuracy of the point measurements, given a residual error. In the case of example 7.1 where $n = 197$ and $\epsilon_{\text{res}} = 0.365$ this results in a value of $\sigma = 0.37$. This value is greater than expected. The reason, as we will see later, lies in the camera model – we are ignoring radial distortion.

Example 7.2. Covariance ellipsoid for an estimated camera
Suppose that the camera is estimated using the Maximum Likelihood (Gold Standard) method, optimizing over a set of camera parameters. The estimated covariance of the point measurements can then be used to compute the covariance of the camera model by back-propagation, according to result 5.10(*p*142). This gives $\Sigma_{\text{camera}} = (\mathtt{J}^{\mathsf{T}}\Sigma_{\text{points}}^{-1}\mathtt{J})^{-1}$ where J is the Jacobian matrix of the measured points in terms of the camera parameters. Uncertainty in 3D world points may also be taken into account in this way. If the camera is parametrized in terms of meaningful parameters (such as camera position), then the variance of each parameter can be measured directly from the diagonal entries of the covariance matrix.

Knowing the covariance of the camera parameters, error bounds or ellipsoids can be computed. For instance, from the covariance matrix for all the parameters we may extract the subblock representing the 3×3 covariance matrix of the camera position, $\Sigma_{\mathbf{C}}$. A confidence ellipsoid for the camera centre is then defined by

$$(\mathbf{C} - \bar{\mathbf{C}})^{\mathsf{T}}\Sigma_{\mathbf{C}}^{-1}(\mathbf{C} - \bar{\mathbf{C}}) = k^2$$

where k^2 is computed from the inverse cumulative χ^2_n distribution in terms of the desired confidence level α: namely $k^2 = F_n^{-1}(\alpha)$ (see figure A2.1(*p*567)). Here n is the

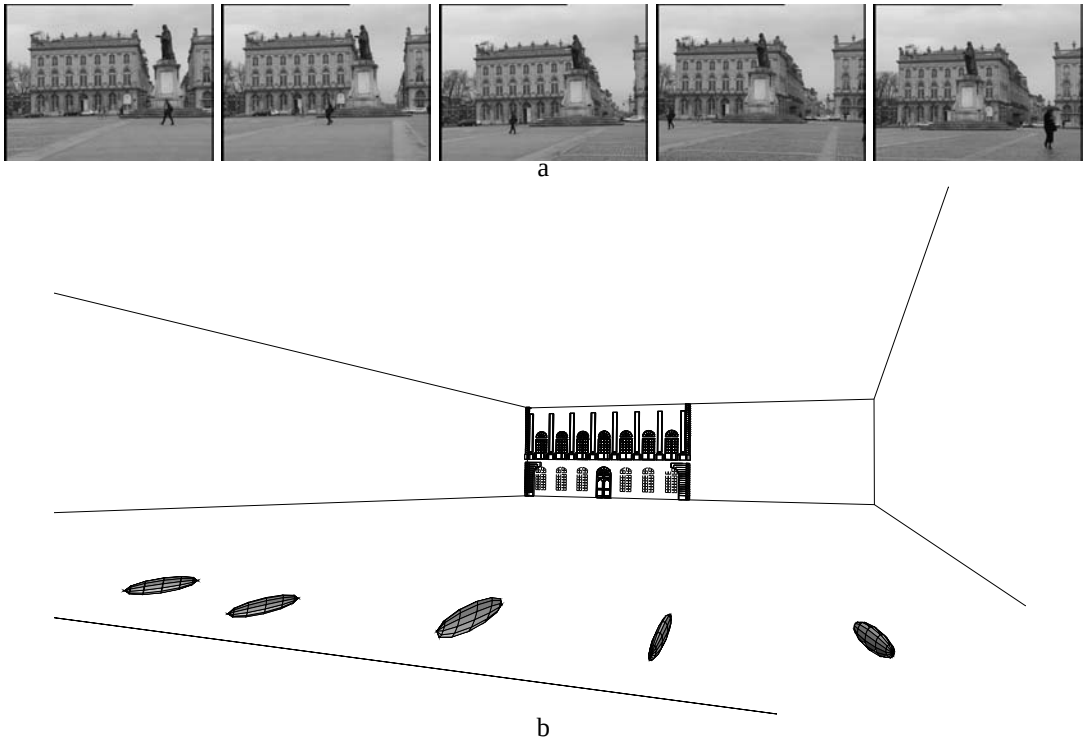


Fig. 7.3. **Camera centre covariance ellipsoids**. *(a) Five images of Stanislas square (Nancy, France), for which 3D calibration points are known. (b) Camera centre covariance ellipsoids corresponding to each image, computed for cameras estimated from the imaged calibration points. Note, the typical cigar shape of the ellipsoid aligned towards the scene data. Figure courtesy of Vincent Lepetit, Marie-Odile Berger and Gilles Simon.*

number of variables – that is 3 in the case of the camera centre. With the chosen level of certainty α, the camera centre lies inside the ellipsoid.

Figure 7.3 shows an example of ellipsoidal uncertainty regions for computed camera centres. Given the estimated covariance matrix for the computed camera, the techniques of section 5.2.6(*p*148) may be used to compute the uncertainty in the image positions of further 3D world points. △

7.4 Radial distortion

The assumption throughout these chapters has been that a linear model is an accurate model of the imaging process. Thus the world point, image point and optical centre are collinear, and world lines are imaged as lines and so on. For real (non-pinhole) lenses this assumption will not hold. The most important deviation is generally a radial distortion. In practice this error becomes more significant as the focal length (and price) of the lens decreases. See figure 7.4.

The cure for this distortion is to correct the image measurements to those that would have been obtained under a perfect linear camera action. The camera is then effectively again a linear device. This process is illustrated in figure 7.5. This correction must

a b

Fig. 7.4. *(a) Short vs (b) long focal lengths. Note the curved imaged lines at the periphery in (a) which are images of straight scene lines.*

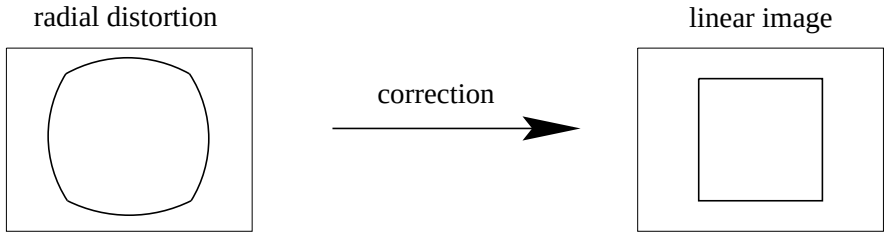


Fig. 7.5. *The image of a square with significant radial distortion is corrected to one that would have been obtained under a perfect linear lens.*

be carried out in the right place in the projection process. Lens distortion takes place during the initial projection of the world onto the image plane, according to (6.2–*p*154). Subsequently, the calibration matrix (7.6) reflects a choice of affine coordinates in the image, translating physical locations in the image plane to pixel coordinates.

We will denote the image coordinates of a point under ideal (non-distorted) pinhole projection by $(\tilde{x}, \tilde{y})$, measured in units of focal-length. Thus, for a point $\mathbf{X}$ we have (see (6.5–*p*155))

$$(\tilde{x}, \tilde{y}, 1)^\mathsf{T} = [\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\text{cam}}$$

where $\mathbf{X}_{\text{cam}}$ is the 3D point in camera coordinates, related to world coordinates by (6.6–*p*156). The actual projected point is related to the ideal point by a radial displacement. Thus, radial (lens) distortion is modelled as

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \tag{7.7}$$

where

- $(\tilde{x}, \tilde{y})$ is the ideal image position (which obeys linear projection).
- (x_d, y_d) is the actual image position, after radial distortion.
- $\tilde{r}$ is the radial distance $\sqrt{\tilde{x}^2 + \tilde{y}^2}$ from the centre for radial distortion.
- $L(\tilde{r})$ is a distortion factor, which is a function of the radius $\tilde{r}$ only.

Correction of distortion. In pixel coordinates the correction is written

$$\hat{x} = x_c + L(r)(x - x_c) \qquad \hat{y} = y_c + L(r)(y - y_c).$$

where (x, y) are the measured coordinates, $(\hat{x}, \hat{y})$ are the corrected coordinates, and (x_c, y_c) is the centre of radial distortion, with $r^2 = (x - x_c)^2 + (y - y_c)^2$. Note, if the aspect ratio is not unity then it is necessary to correct for this when computing r. With this correction the coordinates $(\hat{x}, \hat{y})$ are related to the coordinates of the 3D world point by a linear projective camera.

Choice of the distortion function and centre. The function $L(r)$ is only defined for positive values of r and $L(0) = 1$. An approximation to an arbitrary function $L(r)$ may be given by a Taylor expansion $L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \ldots$. The coefficients for radial correction $\{\kappa_1, \kappa_2, \kappa_3, \ldots, x_c, y_c\}$ are considered part of the interior calibration of the camera. The principal point is often used as the centre for radial distortion, though these need not coincide exactly. This correction, together with the camera calibration matrix, specifies the mapping from an image point to a ray in the camera coordinate system.

Computing the distortion function. The function $L(r)$ may be computed by minimizing a cost based on the deviation from a linear mapping. For example, algorithm 7.1(*p*181) estimates P by minimizing geometric image error for calibration objects such as the Tsai grids of figure 7.1. The distortion function may be included as part of the imaging process, and the parameters κ_i computed together with P during the iterative minimization of the geometric error. Similarly, the distortion function may be computed when estimating the homography between a single Tsai grid and its image.

A simple and more general approach is to determine $L(r)$ by the requirement that images of straight scene lines should be straight. A cost function is defined on the imaged lines (such as the distance between the line joining the imaged line's ends and its mid-point) after the corrective mapping by $L(r)$. This cost is iteratively minimized over the parameters κ_i of the distortion function and the centre of radial distortion. This is a very practical method for images of urban scenes since there are usually plenty of lines available. It has the advantage that no special calibration pattern is required as the scene provides the calibration entities.

Example 7.3. Radial correction. The function $L(r)$ is computed for the image of figure 7.6a by minimizing a cost based on the straightness of imaged scene lines. The image is 640×480 pixels and the correction and centre are computed as $\kappa_1 = 0.103689$, $\kappa_2 = 0.00487908$, $\kappa_3 = 0.00116894$, $\kappa_4 = 0.000841614$, $x_c = 321.87$, $y_c = 241.18$ pixels, where pixels are normalized by the average half-size of the image. This is a correction by 30 pixels at the image periphery. The result of warping the image is shown in figure 7.6b. △

Example 7.4. We continue with the example of the calibration grid shown in figure 7.1 and discussed in example 7.1(*p*181). Radial distortion was removed by the straight line

a b

Fig. 7.6. **Radial distortion correction.** *(a) The original image with lines which are straight in the world, but curved in the image. Several of these lines are annotated by dashed curves. (b) The image warped to remove the radial distortion. Note that the lines in the periphery of the image are now straight, but that the boundary of the image is curved.*

method, and then the camera calibrated using the methods described in this chapter. The results are given in table 7.3.

Note that the residuals after radial correction are substantially smaller. Estimation of the error of point measurements from the residual leads to a value of $\sigma = 0.18$ pixels. Since radial distortion involves selective stretching of the image, it is quite plausible that the effective focal length of the image is changed, as seen here. △

	f_y	f_x/f_y	skew	x_0	y_0	residual
linear	1580.5	1.0044	0.75	377.53	299.12	0.179
iterative	1580.7	1.0044	0.70	377.42	299.02	0.179
algebraic	1556.0	1.0000	0.00	372.42	291.86	0.381
iterative	1556.6	1.0000	0.00	372.41	291.86	0.380
linear	1673.3	1.0063	1.39	379.96	305.78	0.365
iterative	1675.5	1.0063	1.43	379.79	305.25	0.364
algebraic	1633.4	1.0000	0.00	371.21	293.63	0.601
iterative	1637.2	1.0000	0.00	371.32	293.69	0.601

Table 7.3. **Calibration with and without radial distortion correction.** *The results above the line are after radial correction – the results below for comparison are without radial distortion (from the previous tables). The upper two methods in each case solve for the general camera model, the lower two are for a constrained model with square pixels.*

In correcting for radial distortion, it is often not actually necessary to warp the image. Measurements can be made in the original image, for example the position of a corner feature, and the measurement simply mapped according to (7.7). The question of where features *should* be measured does not have an unambiguous answer. Warping the image will distort noise models (because of averaging) and may well introduce aliasing effects. For this reason feature detection on the unwarped image will often be preferable. However, feature grouping, such as linking edgels into straight line primitives,

is best performed after warping since thresholds on linearity may well be erroneously exceeded in the original image.

7.5 Closure

7.5.1 The literature

The original application of the DLT in [Sutherland-63] was for camera computation. Estimation by iterative minimization of geometric errors is a standard procedure of photogrammetrists, e.g. see [Slama-80].

A minimal solution for a calibrated camera (pose from the image of 3 points) was the original problem studied by Fischler and Bolles [Fischler-81] in their RANSAC paper. Solutions to this problem reoccur *often* in the literature; a good treatment is given in [Wolfe-91] and also [Haralick-91]. Quasi-linear solutions for one more than the minimum number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ are in [Quan-98] and [Triggs-99a].

Another class of methods, which are not covered here, is the iterative estimation of a projective camera starting from an affine one. The algorithm of "Model based pose in 25 lines of code" by Dementhon and Davis [Dementhon-95] is based on this idea. A similar method is used in [Christy-96].

Devernay and Faugeras [Devernay-95] introduced a straight line method for computing radial distortion into the computer vision literature. In the photogrammetry literature the method is known as "plumb line correction", see [Brown-71].

7.5.2 Notes and exercises

(i) Given 5 world-to-image point correspondences, $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, show that there are in general four solutions for a camera matrix $\mathtt{P}$ *with zero skew* that exactly maps the world to image points.

(ii) Given 3 world-to-image point correspondences, $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, show that there are in general four solutions for a camera matrix $\mathtt{P}$ *with known calibration* $\mathtt{K}$ that exactly maps the world to image points.

(iii) Find a linear algorithm for computing the camera matrix $\mathtt{P}$ under each of the following conditions:

- (a) The camera location (but not orientation) is known.
- (b) The direction of the principal ray of the camera is known.
- (c) The camera location and the principal ray of the camera are known.
- (d) The camera location and complete orientation of the camera are known.
- (e) The camera location and orientation are known, as well as some subset of the internal camera parameters (α_x, α_y, s, x_0 and y_0).

(iv) **Conflation of focal length and position on principal axis.** Compare the imaged position of a point of depth d before and after an increase in camera focal length Δf, or a displacement Δt_3 of the camera backwards along the principal axis. Let $(x, y)^\mathsf{T}$ and $(x', y')^\mathsf{T}$ be the image coordinates of the point before and

after the change. Following a similar derivation to that of (6.19–*p*169), show that

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + k \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

where $k^f = \Delta f / f$ for a focal length change, or $k^{t_3} = -\Delta t_3 / d$ for a displacement (here skew $s = 0$, and $\alpha_x = \alpha_y = f$).

For a set of calibration points $\mathbf{X}_i$ with depth relief (Δ_i) small compared to the average depth (d_0),

$$k_i^{t_3} = -\Delta t_3 / d_i = -\Delta t_3 / (d_0 + \Delta_i) \approx -\Delta t_3 / d_0$$

i.e. $k_i^{t_3}$ is approximately constant across the set. It follows that in calibrating from such a set, similar image residuals are obtained by changing the focal length k^f or displacing the camera k^{t_3}. Consequently, the estimated parameters of focal length and position on the principal axis are correlated.

(v) **Pushbroom camera computation.** The pushbroom camera, described in section 6.4.1, may also be computed using a DLT method. The x (orthographic) part of the projection matrix has 4 degrees of freedom which may be determined by four or more point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$; the y (perspective) part of the projection matrix has 7 degrees of freedom and may be determined from 7 correspondences. Hence, a minimal solution requires 7 points. Details are given in [Gupta-97].

8

More Single View Geometry

Chapter 6 introduced the projection matrix as the model for the action of a camera on points. This chapter describes the link between other 3D entities and their images under perspective projection. These entities include planes, lines, conics and quadrics; and we develop their forward and back-projection properties.

The camera is dissected further, and reduced to its centre point and image plane. Two properties are established: images acquired by cameras with the same centre are related by a plane projective transformation; and images of entities on the plane at infinity, $\boldsymbol{\pi}_\infty$, do not depend on camera position, only on camera rotation and internal parameters, $\mathtt{K}$.

The images of entities (points, lines, conics) on $\boldsymbol{\pi}_\infty$ are of particular importance. It will be seen that the image of a point on $\boldsymbol{\pi}_\infty$ is a vanishing point, and the image of a line on $\boldsymbol{\pi}_\infty$ a vanishing line; their images depend on both $\mathtt{K}$ and camera rotation. However, the image of the absolute conic, $\boldsymbol{\omega}$, depends only on $\mathtt{K}$; it is unaffected by the camera's rotation. The conic $\boldsymbol{\omega}$ is intimately connected with camera calibration, $\mathtt{K}$, and the relation $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$ is established. It follows that $\boldsymbol{\omega}$ defines the angle between rays back-projected from image points.

These properties enable camera relative rotation to be computed from vanishing points independently of camera position. Further, since $\mathtt{K}$ enables the angle between rays to be computed from image points, in turn $\mathtt{K}$ may be computed from the known angle between rays. In particular $\mathtt{K}$ may be determined from vanishing points corresponding to orthogonal scene directions. This means that a camera can be calibrated from scene features, without requiring known world coordinates.

A final geometric entity introduced in this chapter is the calibrating conic, which enables a geometric visualization of $\mathtt{K}$.

8.1 Action of a projective camera on planes, lines, and conics

In this section (and indeed in most of this book) it is only the 3×4 *form* and rank of the camera projection matrix $\mathtt{P}$ that is important in determining its action. The particular properties and relations of its elements are often not relevant.

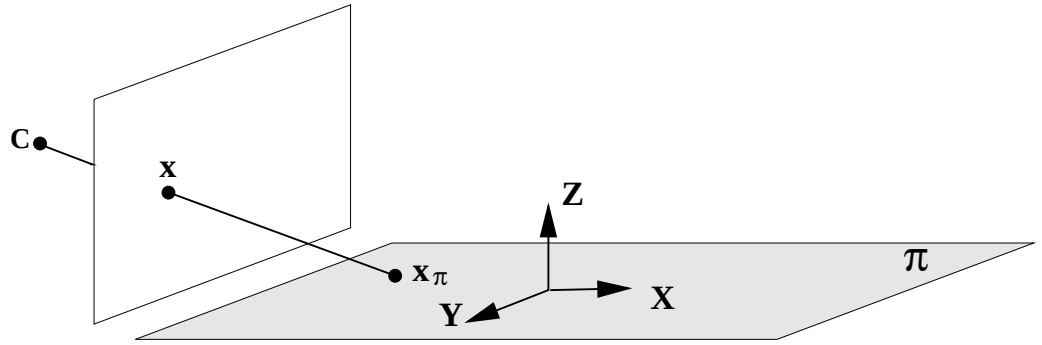


Fig. 8.1. **Perspective image of points on a plane.** *The* XY*-plane of the world coordinate frame is aligned with the plane* π*. Points on the image and scene planes are related by a plane projective transformation.*

8.1.1 On planes

The point imaging equation $\mathbf{x} = \mathtt{P}\mathbf{X}$ is a map from a point in a world coordinate frame, to a point in image coordinates. We have the freedom to choose the world coordinate frame. Suppose it is chosen such that the XY-plane corresponds to a plane $\boldsymbol{\pi}$ in the scene, so that points on the scene plane have zero Z-coordinate as shown in figure 8.1 (it is assumed that the camera centre does not lie on the scene plane). Then, if the columns of P are denoted as $\mathbf{p}_i$, the image of a point on $\boldsymbol{\pi}$ is given by

$$\mathbf{x} = \mathtt{P}\mathbf{X} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_4 \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ 1 \end{pmatrix}.$$

So that the map between points $\mathbf{x}_\pi = (\mathtt{X}, \mathtt{Y}, 1)^\mathsf{T}$ on $\boldsymbol{\pi}$ and their image $\mathbf{x}$ is a general planar homography (a plane to plane projective transformation): $\mathbf{x} = \mathtt{H}\mathbf{x}_\pi$, with H a 3×3 matrix of rank 3. This shows that:

- *The most general transformation that can occur between a scene plane and an image plane under perspective imaging is a plane projective transformation.*

If the camera is affine, then a similar derivation shows that the scene and image planes are related by an affine transformation.

Example 8.1. For a calibrated camera (6.8–*p*156) $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid \mathbf{t}]$, the homography between a world plane at $\mathtt{Z} = 0$ and the image is

$$\mathtt{H} = \mathtt{K}\left[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}\right] \tag{8.1}$$

where $\mathbf{r}_i$ are the columns of R. △

8.1.2 On lines

Forward projection. A line in 3-space projects to a line in the image. This is easily seen geometrically – the line and camera centre define a plane, and the image is the intersection of this plane with the image plane (figure 8.2) – and is proved algebraically

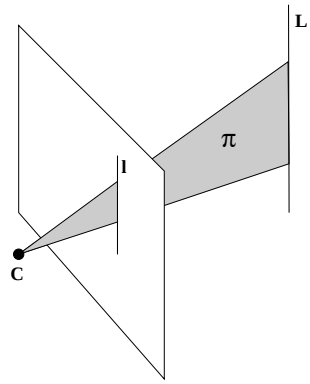


Fig. 8.2. **Line projection.** *A line* $\mathbf{L}$ *in 3-space is imaged as a line* $\mathbf{l}$ *by a perspective camera. The image line* $\mathbf{l}$ *is the intersection of the plane* $\boldsymbol{\pi}$*, defined by* $\mathbf{L}$ *and the camera centre* $\mathbf{C}$*, with the image plane. Conversely an image line* $\mathbf{l}$ *back-projects to a plane* $\boldsymbol{\pi}$ *in 3-space. The plane is the "pull-back" of the line.*

by noting that if $\mathbf{A}, \mathbf{B}$ are points in 3-space, and $\mathbf{a}, \mathbf{b}$ their images under $\mathtt{P}$, then a point $\mathbf{X}(\mu) = \mathbf{A} + \mu\mathbf{B}$ on a line which is the join of $\mathbf{A}, \mathbf{B}$ in 3-space projects to a point

$$\begin{aligned} \mathbf{x}(\mu) &= \mathtt{P}(\mathbf{A} + \mu\mathbf{B}) = \mathtt{P}\mathbf{A} + \mu\mathtt{P}\mathbf{B} \\ &= \mathbf{a} + \mu\mathbf{b} \end{aligned}$$

which is on the line joining $\mathbf{a}$ and $\mathbf{b}$.

Back-projection of lines. The set of points in space which map to a line in the image is a plane in space defined by the camera centre and image line, as shown in figure 8.2. Algebraically,

Result 8.2. *The set of points in space mapping to a line* $\mathbf{l}$ *via the camera matrix* $\mathtt{P}$ *is the plane* $\mathtt{P}^\mathsf{T}\mathbf{l}$*.*

Proof. A point $\mathbf{x}$ lies on $\mathbf{l}$ if and only if $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$. A space point $\mathbf{X}$ maps to a point $\mathtt{P}\mathbf{X}$, which lies on the line if and only if $\mathbf{X}^\mathsf{T}\mathtt{P}^\mathsf{T}\mathbf{l} = 0$. Thus, if $\mathtt{P}^\mathsf{T}\mathbf{l}$ is taken to represent a plane, then $\mathbf{X}$ lies on this plane if and only if $\mathbf{X}$ maps to a point on the line $\mathbf{l}$. In other words, $\mathtt{P}^\mathsf{T}\mathbf{l}$ is the back-projection of the line $\mathbf{l}$. □

Geometrically there is a *star* (two-parameter family) of planes through the camera centre, and the three rows of the projection matrix $\mathbf{P}^{i\mathsf{T}}$ (6.12–*p*159) are a basis for this star. The plane $\mathtt{P}^\mathsf{T}\mathbf{l}$ is a linear combination of this basis corresponding to the element of the star containing the camera centre and the line $\mathbf{l}$. For example, if $\mathbf{l} = (0, 1, 0)^\mathsf{T}$ then the plane is $\mathbf{P}^2$, and is the back projection of the image x-axis.

Plücker line representation. *Understanding this material on Plücker line mapping is not required for following the rest of the book.*

We now turn to forward projection of lines. If a line in 3-space is represented by Plücker coordinates then its image can be expressed as a linear map on these coordinates. We will develop this map for both the 4×4 matrix and 6-vector line representations.

Result 8.3. *Under the camera mapping* $\mathtt{P}$*, a line in 3-space represented as a Plücker matrix* $\mathtt{L}$*, as defined in (3.8–*p*70), is imaged as the line* $\mathbf{l}$ *where*

$$[\mathbf{l}]_\times = \mathtt{P}\mathtt{L}\mathtt{P}^\mathsf{T}. \tag{8.2}$$

where the notation $[\mathbf{l}]_\times$ *is defined in (A4.5–*p*581).*

Proof. Suppose as above that $\mathbf{a} = \mathtt{P}\mathbf{A},\ \mathbf{b} = \mathtt{P}\mathbf{B}$. The Plücker matrix $\mathtt{L}$ for the line through $\mathbf{A}, \mathbf{B}$ in 3-space is $\mathtt{L} = \mathbf{A}\mathbf{B}^\mathsf{T} - \mathbf{B}\mathbf{A}^\mathsf{T}$. Then the matrix $\mathtt{M} = \mathtt{P}\mathtt{L}\mathtt{P}^\mathsf{T} = \mathbf{a}\mathbf{b}^\mathsf{T} - \mathbf{b}\mathbf{a}^\mathsf{T}$ is 3×3 and antisymmetric, with null-space $\mathbf{a} \times \mathbf{b}$. Consequently, $\mathtt{M} = [\mathbf{a} \times \mathbf{b}]_\times$, and since the line through the image points is given by $\mathbf{l} = \mathbf{a} \times \mathbf{b}$, this completes the proof. □

It is clear from the form of (8.2) that there is a linear relation between the image line coordinates l_i and the world line coordinates L_{jk}, but that this relation is quadratic in the elements of the point projection matrix $\mathtt{P}$. Thus, (8.2) may be rearranged such that the map between the Plücker line coordinates, $\mathcal{L}$ (a 6-vector), and the image line coordinates $\mathbf{l}$ (a 3-vector) is represented by a single 3×6 matrix. It can be shown that

Definition 8.4. The *line projection matrix* $\mathcal{P}$ is the 3×6 matrix of rank 3 given by

$$\mathcal{P} = \begin{bmatrix} \mathbf{P}^2 \wedge \mathbf{P}^3 \\ \mathbf{P}^3 \wedge \mathbf{P}^1 \\ \mathbf{P}^1 \wedge \mathbf{P}^2 \end{bmatrix} \tag{8.3}$$

where $\mathbf{P}^{i\mathsf{T}}$ are the rows of the point camera matrix $\mathtt{P}$, and $\mathbf{P}^i \wedge \mathbf{P}^j$ are the Plücker line coordinates of the intersection of the planes $\mathbf{P}^i$ and $\mathbf{P}^j$.

Then the forward line projection is given by

Result 8.5. *Under the line projection matrix* $\mathcal{P}$*, a line in* $\mathbb{P}^3$ *represented by Plücker line coordinates* $\mathcal{L}$*, as defined in (3.11–*p*72), is mapped to the image line*

$$\mathbf{l} = \mathcal{P}\mathcal{L} = \begin{bmatrix} (\mathbf{P}^2 \wedge \mathbf{P}^3 | \mathcal{L}) \\ (\mathbf{P}^3 \wedge \mathbf{P}^1 | \mathcal{L}) \\ (\mathbf{P}^1 \wedge \mathbf{P}^2 | \mathcal{L}) \end{bmatrix} \tag{8.4}$$

where the product $(\mathcal{L}|\hat{\mathcal{L}})$ *is defined in (3.13–*p*72).*

Proof. Suppose the line in 3-space is the join of the points $\mathbf{A}$ and $\mathbf{B}$, and these project to $\mathbf{a} = \mathtt{P}\mathbf{A}, \mathbf{b} = \mathtt{P}\mathbf{B}$ respectively. Then the image line $\mathbf{l} = \mathbf{a} \times \mathbf{b} = (\mathtt{P}\mathbf{A}) \times (\mathtt{P}\mathbf{B})$. Consider the first element

$$\begin{aligned} l_1 &= (\mathbf{P}^{2\mathsf{T}}\mathbf{A})(\mathbf{P}^{3\mathsf{T}}\mathbf{B}) - (\mathbf{P}^{2\mathsf{T}}\mathbf{B})(\mathbf{P}^{3\mathsf{T}}\mathbf{A}) \\ &= (\mathbf{P}^2 \wedge \mathbf{P}^3 | \mathcal{L}) \end{aligned}$$

where the second equality follows from (3.14–*p*73). The other components follow in a similar manner. □

The line projection matrix $\mathcal{P}$ plays the same role for lines as $\mathtt{P}$ does for points. The rows of $\mathcal{P}$ may be interpreted geometrically as *lines*, in a similar manner to the interpretation of the rows of the point camera matrix $\mathtt{P}$ as *planes* given in section 6.2.1(*p*158). The rows $\mathbf{P}^{i\mathsf{T}}$ of $\mathtt{P}$ are the principal plane and axis planes of the camera. The rows of $\mathcal{P}$ are the lines of intersection of pairs of these camera planes. For example, the first row of $\mathcal{P}$ is $\mathbf{P}^2 \wedge \mathbf{P}^3$, and this is the 6-vector Plücker line representation of the line of intersection of the $y = 0$ axis plane, $\mathbf{P}^2$, and the principal plane, $\mathbf{P}^3$. The three lines corresponding to the three rows of $\mathcal{P}$ intersect at the camera centre. Consider lines $\mathcal{L}$ in 3-space for which $\mathcal{P}\mathcal{L} = \mathbf{0}$. These lines are in the null-space of $\mathcal{P}$. Since each row of $\mathcal{P}$ is a line, and from result 3.5(*p*72) the product $(\mathcal{L}_1|\mathcal{L}_2) = 0$ if two lines intersect, if follows that $\mathcal{L}$ intersects each of the lines represented by the rows of $\mathcal{P}$. These lines are the intersection of the camera planes, and the only point on all 3 camera planes is the camera centre. Thus we have

- *The lines $\mathcal{L}$ in $\mathbb{P}^3$ for which $\mathcal{P}\mathcal{L} = \mathbf{0}$ pass through the camera centre.*

The 3×6 matrix $\mathcal{P}$ has a 3-dimensional null-space. Allowing for the homogeneous scale factor, this null-space is a two-parameter family of lines containing the camera centre. This is to be expected since there is a star (two parameter family) of lines in $\mathbb{P}^3$ concurrent with a point.

8.1.3 On conics

Back-projection of conics. A conic $\mathtt{C}$ back-projects to a cone. A cone is a degenerate quadric, i.e. the 4×4 matrix representing the quadric does not have full rank. The cone vertex, in this case the camera centre, is the null-vector of the quadric matrix.

Result 8.6. *Under the camera $\mathtt{P}$ the conic $\mathtt{C}$ back-projects to the cone*

$$\mathtt{Q}_{\mathrm{co}} = \mathtt{P}^{\mathsf{T}}\mathtt{C}\mathtt{P}.$$

Proof. A point $\mathbf{x}$ lies on $\mathtt{C}$ if and only if $\mathbf{x}^{\mathsf{T}}\mathtt{C}\mathbf{x} = 0$. A space point $\mathbf{X}$ maps to a point $\mathtt{P}\mathbf{X}$, which lies on the conic if and only if $\mathbf{X}^{\mathsf{T}}\mathtt{P}^{\mathsf{T}}\mathtt{C}\mathtt{P}\mathbf{X} = 0$. Thus, if $\mathtt{Q}_{\mathrm{co}} = \mathtt{P}^{\mathsf{T}}\mathtt{C}\mathtt{P}$ is taken to represent a quadric, then $\mathbf{X}$ lies on this quadric if and only if $\mathbf{X}$ maps to a point on the conic $\mathtt{C}$. In other words, $\mathtt{Q}_{\mathrm{co}}$ is the back-projection of the conic $\mathtt{C}$. □

Note the camera centre $\mathbf{C}$ is the vertex of the degenerate quadric since $\mathtt{Q}_{\mathrm{co}}\mathbf{C} = \mathtt{P}^{\mathsf{T}}\mathtt{C}(\mathtt{P}\mathbf{C}) = \mathbf{0}$.

Example 8.7. Suppose that $\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$; then the conic $\mathtt{C}$ back-projects to the cone

$$\mathtt{Q}_{\mathrm{co}} = \begin{bmatrix} \mathtt{K}^{\mathsf{T}} \\ \mathbf{0}^{\mathsf{T}} \end{bmatrix} \mathtt{C} \begin{bmatrix} \mathtt{K} \mid \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathtt{K}^{\mathsf{T}}\mathtt{C}\mathtt{K} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 0 \end{bmatrix}.$$

The matrix $\mathtt{Q}_{\mathrm{co}}$ has rank 3. Its null-vector is the camera centre $\mathbf{C} = (0, 0, 0, 1)^{\mathsf{T}}$. △

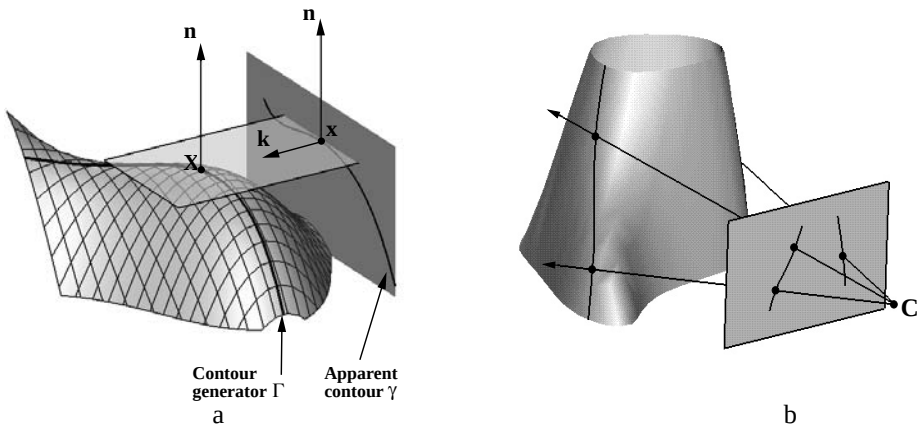


Fig. 8.3. **Contour generator and apparent contour.** *(a) for parallel projection; (b) for central projection. The ray from the camera centre through* $\mathbf{x}$ *is tangent to the surface at* $\mathbf{X}$*. The set of such tangent points* $\mathbf{X}$ *defines the contour generator, and their image defines the apparent contour. In general the contour generator is a space curve. Figure courtesy of Roberto Cipolla and Peter Giblin.*

8.2 Images of smooth surfaces

The image outline of a smooth surface S results from surface points at which the imaging rays are *tangent* to the surface, as shown in figure 8.3. Similarly, *lines* tangent to the outline back-project to planes which are *tangent planes* to the surface.

Definition 8.8. The *contour generator* $\boldsymbol{\Gamma}$ is the set of points $\mathbf{X}$ on S at which rays are tangent to the surface. The corresponding image *apparent contour* $\boldsymbol{\gamma}$ is the set of points $\mathbf{x}$ which are the image of $\mathbf{X}$, i.e. $\boldsymbol{\gamma}$ is the image of $\boldsymbol{\Gamma}$.

The apparent contour is also called the "outline" and "profile". If the surface is viewed in the direction of $\mathbf{X}$ from the camera centre, then the surface appears to fold, or to have a boundary or occluding contour.

It is evident that the contour generator $\boldsymbol{\Gamma}$ depends only on the relative position of the camera centre and surface, not on the image plane. However, the apparent contour $\boldsymbol{\gamma}$ is defined by the intersection of the image plane with the rays to the contour generator, and so does depend on the position of the image plane.

In the case of parallel projection with direction $\mathbf{k}$, consider all the rays parallel to $\mathbf{k}$ which are tangent to S, see figure 8.3a. These rays form a "cylinder" of tangent rays, and the curve along which this cylinder is tangent to S is the contour generator $\boldsymbol{\Gamma}$. The curve in which the cylinder meets the image plane is the apparent contour $\boldsymbol{\gamma}$. Note that both $\boldsymbol{\Gamma}$ and $\boldsymbol{\gamma}$ depend in an essential way on $\mathbf{k}$. The set $\boldsymbol{\Gamma}$ slips over the surface as the direction of $\mathbf{k}$ changes. For example, with S a sphere, $\boldsymbol{\Gamma}$ is the great circle orthogonal to $\mathbf{k}$. In this case, the contour generator $\boldsymbol{\Gamma}$ is a plane curve, but in general $\boldsymbol{\Gamma}$ is a space curve.

We next describe the projection properties of quadrics. For this class of surface algebraic expressions can be developed for the contour generator and apparent contour.

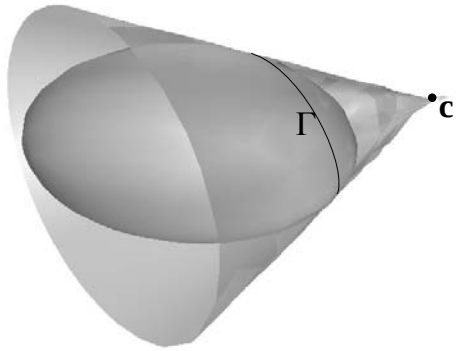


Fig. 8.4. **The cone of rays for a quadric.** *The vertex of the cone is the camera centre. (a) The contour generator* $\mathbf{\Gamma}$ *of a quadric is a plane curve (a conic) which is the intersection of the quadric with the polar plane of the camera centre,* $\mathbf{C}$.

8.3 Action of a projective camera on quadrics

A quadric is a smooth surface and so its outline curve is given by points where the back-projected rays are tangent to the quadric surface as shown in figure 8.4.

Suppose the quadric is a sphere, then the cone of rays between the camera centre and quadric is right-circular, i.e. the contour generator is a circle, with the plane of the circle orthogonal to the line joining the camera and sphere centres. This can be seen from the rotational symmetry of the geometry about this line. The image of the sphere is obtained by intersecting the cone with the image plane. It is clear that this is a classical conic section, so that the apparent contour of a sphere is a conic. In particular if the sphere centre lies on the principal ($\mathtt{Z}$) camera axis, then the conic is a circle.

Now consider a 3-space projective transformation of this geometry. Under this map the sphere is transformed to a quadric and the apparent contour to a conic. However, since intersection and tangency are preserved, the contour generator is a (plane) conic. Consequently, the apparent contour of a general quadric is a conic, and the contour generator is also a conic. We will now give algebraic representations for these geometric results.

Forward projection of quadrics. Since the outline arises from tangency, it is not surprising that the dual of the quadric, $\mathtt{Q}^*$, is important here since it defines the tangent planes to the quadric $\mathtt{Q}$.

Result 8.9. *Under the camera matrix* $\mathtt{P}$ *the outline of the quadric* $\mathtt{Q}$ *is the conic* $\mathtt{C}$ *given by*

$$\mathtt{C}^* = \mathtt{P}\mathtt{Q}^*\mathtt{P}^\mathsf{T}. \tag{8.5}$$

Proof. This expression is simply derived from the observation that lines $\mathbf{l}$ tangent to the conic outline satisfy $\mathbf{l}^\mathsf{T}\mathtt{C}^*\mathbf{l} = 0$. These lines back-project to planes $\boldsymbol{\pi} = \mathtt{P}^\mathsf{T}\mathbf{l}$ that are tangent to the quadric and thus satisfy $\boldsymbol{\pi}^\mathsf{T}\mathtt{Q}^*\boldsymbol{\pi} = 0$. Then it follows that for each line

$$\begin{aligned}\boldsymbol{\pi}^\mathsf{T}\mathtt{Q}^*\boldsymbol{\pi} &= \mathbf{l}^\mathsf{T}\mathtt{P}\mathtt{Q}^*\mathtt{P}^\mathsf{T}\mathbf{l} \\ &= \mathbf{l}^\mathsf{T}\mathtt{C}^*\mathbf{l} = 0\end{aligned}$$

and since this is true for all lines tangent to $\mathtt{C}$ the result follows. □

Note the similarity of (8.5) with the projection of a line represented by a Plücker matrix (8.2). An expression for the projection of the point quadric $\mathtt{Q}$ can be derived from (8.5) but it is quite complicated. However, the plane of the contour generator is easily expressed in terms of $\mathtt{Q}$:

- *The plane of* Γ *for a quadric* $\mathtt{Q}$ *and camera with centre* $\mathbf{C}$ *is given by* $\boldsymbol{\pi}_\Gamma = \mathtt{Q}\mathbf{C}$.

This result follows directly from the pole–polar relation for a point and quadric of section 3.2.3(*p*73). Its proof is left as an exercise. Note, the intersection of a quadric and plane is a conic. So Γ is a conic and its image $\boldsymbol{\gamma}$, which is the apparent contour, is also a conic as has been seen above.

We may also derive an expression for the cone of rays formed by the camera centre and quadric. This cone is a degenerate quadric of rank 3.

Result 8.10. *The cone with vertex* $\mathbf{V}$ *and tangent to the quadric* $\mathtt{Q}$ *is the degenerate quadric*

$$\mathtt{Q}_{\mathrm{co}} = (\mathbf{V}^\mathsf{T}\mathtt{Q}\mathbf{V})\mathtt{Q} - (\mathtt{Q}\mathbf{V})(\mathtt{Q}\mathbf{V})^\mathsf{T}.$$

Note that $\mathtt{Q}_{\mathrm{co}}\mathbf{V} = \mathbf{0}$, so that $\mathbf{V}$ is the vertex of the cone as required. The proof is omitted.

Example 8.11. We write the quadric in block form:

$$\mathtt{Q} = \begin{bmatrix} \mathtt{Q}_{3\times3} & \mathbf{q} \\ \mathbf{q}^\mathsf{T} & q_{44} \end{bmatrix}.$$

Then if $\mathbf{V} = (0,0,0,1)^\mathsf{T}$, which corresponds to the cone vertex being at the centre of the world coordinate frame,

$$\mathtt{Q}_{\mathrm{co}} = \begin{bmatrix} q_{44}\mathtt{Q}_{3\times3} - \mathbf{q}\mathbf{q}^\mathsf{T} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix}$$

which is clearly a degenerate quadric. △

8.4 The importance of the camera centre

An object in 3-space and camera centre define a set of rays, and an image is obtained by intersecting these rays with a plane. Often this set is referred to as a *cone* of rays, even though it is not a classical cone. Suppose the cone of rays is intersected by two planes, as shown in figure 8.5, then the two images, I and I', are clearly related by a perspective map. This means that images obtained with the same camera centre may be mapped to one another by a plane projective transformation, in other words they are projectively equivalent and so have the same projective properties. A camera can thus be thought of as a projective imaging device – measuring projective properties of the cone of rays with vertex the camera centre.

The result that the two images I and I' are related by a homography will now be derived algebraically to obtain a formula for this homography. Consider two cameras

$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\widetilde{\mathbf{C}}], \quad \mathtt{P}' = \mathtt{K}'\mathtt{R}'[\mathtt{I} \mid -\widetilde{\mathbf{C}}]$$

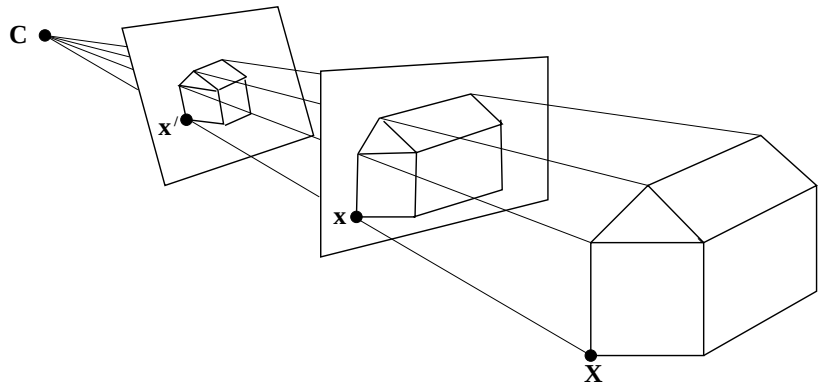


Fig. 8.5. **The cone of rays with vertex the camera centre.** *An image is obtained by intersecting this cone with a plane. A ray between a 3-space point* $\mathbf{X}$ *and the camera centre* $\mathbf{C}$ *pierces the planes in the image points* $\mathbf{x}$ *and* $\mathbf{x}'$*. All such image points are related by a planar homography,* $\mathbf{x}' = \mathtt{H}\mathbf{x}$.

with the same centre. Note that since the cameras have a common centre there is a simple relation between them, namely $\mathtt{P}' = (\mathtt{K}'\mathtt{R}')(\mathtt{K}\mathtt{R})^{-1}\mathtt{P}$. It then follows that the images of a 3-space point $\mathbf{X}$ by the two cameras are related as

$$\mathbf{x}' = \mathtt{P}'\mathbf{X} = (\mathtt{K}'\mathtt{R}')(\mathtt{K}\mathtt{R})^{-1}\mathtt{P}\mathbf{X} = (\mathtt{K}'\mathtt{R}')(\mathtt{K}\mathtt{R})^{-1}\mathbf{x}.$$

That is, the corresponding image points are related by a planar homography (a 3×3 matrix) as $\mathbf{x}' = \mathtt{H}\mathbf{x}$, where $\mathtt{H} = (\mathtt{K}'\mathtt{R}')(\mathtt{K}\mathtt{R})^{-1}$.

We will now investigate several cases of moving the image plane whilst fixing the camera centre. For simplicity the world coordinate frame will be chosen to coincide with the camera's, so that $\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$ (and it will be assumed that the image plane never contains the centre, as the image would then be degenerate).

8.4.1 Moving the image plane

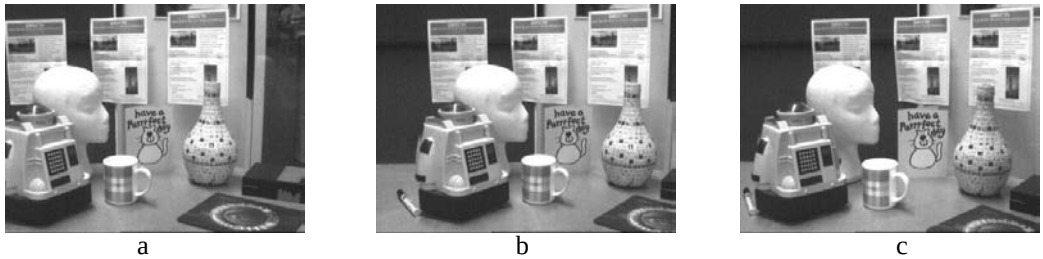
Consider first an increase in focal length. To a first approximation this corresponds to a displacement of the image plane along the principal axis. The image effect is a simple magnification. This is only a first approximation because with a compound lens zooming will perturb both the principal point and the effective camera centre. Algebraically, if $\mathbf{x}$, $\mathbf{x}'$ are the images of a point $\mathbf{X}$ before and after zooming, then

$$\begin{aligned} \mathbf{x} &= \mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X} \\ \mathbf{x}' &= \mathtt{K}'[\mathtt{I} \mid \mathbf{0}]\mathbf{X} = \mathtt{K}'\mathtt{K}^{-1}\left(\mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X}\right) = \mathtt{K}'\mathtt{K}^{-1}\mathbf{x} \end{aligned}$$

so that $\mathbf{x}' = \mathtt{H}\mathbf{x}$ with $\mathtt{H} = \mathtt{K}'\mathtt{K}^{-1}$. If only the focal lengths differ between $\mathtt{K}$ and $\mathtt{K}'$ then a short calculation shows that

$$\mathtt{K}'\mathtt{K}^{-1} = \begin{bmatrix} k\mathtt{I} & (1-k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}.$$

where $\tilde{\mathbf{x}}_0$ is the inhomogeneous principal point, and $k = f'/f$ is the magnification factor. This result follows directly from similar triangles: the effect of zooming by a



a b c

Fig. 8.6. *Between images (a) and (b) the camera rotates about the camera centre. Corresponding points (that is images of the same 3D point) are related by a plane projective transformation. Note that 3D points at different depths which are coincident in image (a), such as the mug lip and cat body, are also coincident in (b), so there is no motion parallax in this case. However, between images (a) and (c) the camera rotates about the camera centre* and *translates. Under this general motion coincident points of differing depth in (a) are imaged at different points in (c), so there* is *motion parallax in this case due to the camera translation.*

factor k is to move the image point $\tilde{\mathbf{x}}$ on a line radiating from the principal point $\tilde{\mathbf{x}}_0$ to the point $\tilde{\mathbf{x}}' = k\tilde{\mathbf{x}} + (1-k)\tilde{\mathbf{x}}_0$. Algebraically, using the most general form (6.10–*p*157) of the calibration matrix $\mathtt{K}$, we may write

$$\begin{aligned}\mathtt{K}' &= \begin{bmatrix} k\mathtt{I} & (1-k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \mathtt{K} = \begin{bmatrix} k\mathtt{I} & (1-k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathtt{A} & \tilde{\mathbf{x}}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \\ &= \begin{bmatrix} k\mathtt{A} & \tilde{\mathbf{x}}_0 \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} = \mathtt{K} \begin{bmatrix} k\mathtt{I} & \\ & 1 \end{bmatrix}.\end{aligned}$$

This shows that

- *The effect of zooming by a factor k is to multiply the calibration matrix $\mathtt{K}$ on the right by* $\mathrm{diag}(k, k, 1)$.

8.4.2 Camera rotation

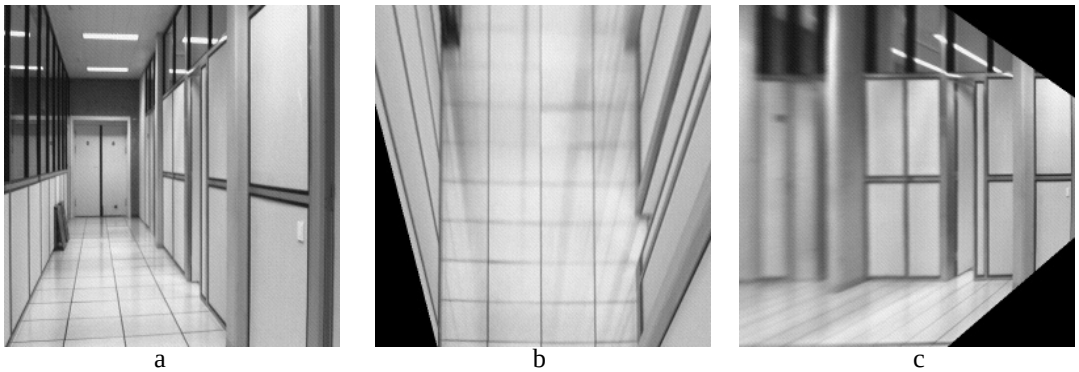
A second common example is where the camera is rotated about its centre with no change in the internal parameters. Examples of this "pure" rotation are given in figure 8.6 and figure 8.9. Algebraically, if $\mathbf{x}$, $\mathbf{x}'$ are the images of a point $\mathbf{X}$ before and after the pure rotation

$$\begin{aligned}\mathbf{x} &= \mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X} \\ \mathbf{x}' &= \mathtt{K}\,[\mathtt{R} \mid \mathbf{0}]\,\mathbf{X} = \mathtt{K}\mathtt{R}\mathtt{K}^{-1}\mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X} = \mathtt{K}\mathtt{R}\mathtt{K}^{-1}\mathbf{x}\end{aligned}$$

so that $\mathbf{x}' = \mathtt{H}\mathbf{x}$ with $\mathtt{H} = \mathtt{K}\mathtt{R}\mathtt{K}^{-1}$. This homography is a *conjugate rotation* and is discussed further in section A7.1(*p*628). For now, we mention a few of its properties by way of an example.

Example 8.12. Properties of a conjugate rotation

The homography $\mathtt{H} = \mathtt{K}\mathtt{R}\mathtt{K}^{-1}$ has the same eigenvalues (up to scale) as the rotation matrix, namely $\{\mu, \mu e^{i\theta}, \mu e^{-i\theta}\}$, where μ is an unknown scale factor (if $\mathtt{H}$ is scaled such that $\det \mathtt{H} = 1$, then $\mu = 1$). Consequently the angle of rotation between views may be computed directly from the phase of the complex eigenvalues of $\mathtt{H}$. Similarly, it can be



a b c

Fig. 8.7. **Synthetic views.** *(a) Source image. (b) Fronto-parallel view of the corridor floor generated from (a) using the four corners of a floor tile to compute the homography. (c) Fronto-parallel view of the corridor wall generated from (a) using the four corners of the door frame to compute the homography.*

shown (see exercises) that the eigenvector of $\mathtt{H}$ corresponding to the real eigenvalue is the vanishing point of the rotation axis.

For example, between images (a) and (b) of figure 8.6 there is a pure rotation of the camera. The homography $\mathtt{H}$ is computed by algorithm 4.6(*p*123), and from this the angle of rotation is estimated as $4.66°$, and the axis vanishing point as $(-0.0088, 1, 0.0001)^\mathsf{T}$, i.e. virtually at infinity in the y direction, so the rotation axis is almost parallel to the y-axis. △

The transformation $\mathtt{H} = \mathtt{K}\mathtt{R}\mathtt{K}^{-1}$ is an example of the *infinite homography* mapping $\mathtt{H}_\infty$, that will appear many times through this book. It is defined in section 13.4(*p*338). The conjugation property is used for auto-calibration in chapter 19.

8.4.3 Applications and examples

The homographic relation between images with the same camera centre can be exploited in several ways. One is the creation of synthetic images by projective warping. Another is mosaicing, where panoramic images can be created by using planar homographies to "sew" together views obtained by a rotating camera.

Example 8.13. Synthetic views

New images corresponding to different camera orientations (with the same camera centre) can be generated from an existing image by warping with planar homographies.

In a fronto-parallel view a rectangle is imaged as a rectangle, and the world and image rectangle have the same aspect ratio. Conversely, a fronto-parallel view can be synthesized by warping an image with the homography that maps a rectangle imaged as a quadrilateral to a rectangle with the correct aspect ratio. The algorithm is:

(i) Compute the homography $\mathtt{H}$ which maps the image quadrilateral to a rectangle with the correct aspect ratio.
(ii) Projectively warp the source image with this homography.

Examples are shown in figure 8.7. △

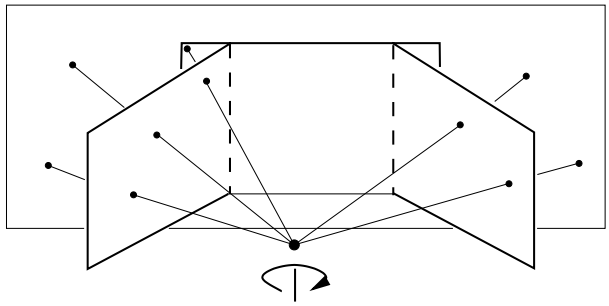


Fig. 8.8. *Three images acquired by a rotating camera may be registered to the frame of the middle one, as shown, by projectively warping the outer images to align with the middle one.*

Example 8.14. Planar panoramic mosaicing

Images acquired by a camera rotating about its centre are related to each other by a planar homography. A set of such images may be registered with the plane of one of the images by projectively warping the other images, as illustrated in figure 8.8.

Fig. 8.9. **Planar panoramic mosaicing.** *Eight images (out of thirty) acquired by rotating a camcorder about its centre. The thirty images are registered (automatically) using planar homographies and composed into the single panoramic mosaic shown. Note the characteristic "bow tie" shape resulting from registering to an image at the middle of the sequence.*

In outline the algorithm is:

(i) Choose one image of the set as a reference.
(ii) Compute the homography $\mathtt{H}$ which maps one of the other images of the set to this reference image.

(iii) Projectively warp the image with this homography, and augment the reference image with the non-overlapping part of the warped image.
(iv) Repeat the last two steps for the remaining images of the set.

The homographies may be computed by identifying (at least) four corresponding points, or by using the automatic method of algorithm 4.6(*p*123). An example mosaic is shown in figure 8.9. △

8.4.4 Projective (reduced) notation

It will be seen in chapter 20 that if canonical projective coordinates are chosen for world and image points, i.e.

$$\mathbf{X}_1 = (1,0,0,0)^\mathsf{T},\ \mathbf{X}_2 = (0,1,0,0)^\mathsf{T},\ \mathbf{X}_3 = (0,0,1,0)^\mathsf{T},\ \mathbf{X}_4 = (0,0,0,1)^\mathsf{T},$$

and

$$\mathbf{x}_1 = (1,0,0)^\mathsf{T},\ \mathbf{x}_2 = (0,1,0)^\mathsf{T},\ \mathbf{x}_3 = (0,0,1)^\mathsf{T},\ \mathbf{x}_4 = (1,1,1)^\mathsf{T},$$

then the camera matrix

$$\mathtt{P} = \begin{bmatrix} a & 0 & 0 & -d \\ 0 & b & 0 & -d \\ 0 & 0 & c & -d \end{bmatrix} \tag{8.6}$$

satisfies $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i,\ i = 1,\ldots,4$, and also that $\mathtt{P}(a^{-1}, b^{-1}, c^{-1}, d^{-1})^\mathsf{T} = \mathbf{0}$, which means that the camera centre is $\mathbf{C} = (a^{-1}, b^{-1}, c^{-1}, d^{-1})^\mathsf{T}$. This is known as the *reduced* camera matrix, and it is clearly completely specified by the 3 degrees of freedom of the camera centre $\mathbf{C}$. This is a further illustration of the fact that all images acquired by cameras with the same camera centre are projectively equivalent – the camera has been reduced to its essence: a projective device whose action is to map $\mathbb{P}^3$ to $\mathbb{P}^2$ with only the position of the camera centre affecting the result. This camera representation is used in establishing duality relations in chapter 20.

8.4.5 Moving the camera centre

The cases of zooming and camera rotation illustrate that moving the image plane, whilst fixing the camera centre, induces a transformation between images that depends only on the image plane motion, but *not* on the 3-space structure. Conversely, no information on 3-space structure can be obtained by this action. However, if the camera centre is moved then the map between corresponding image points *does* depend on the 3-space structure, and indeed may often be used to (partially) determine the structure. This is the subject of much of the remainder of this book.

How can one determine from the images alone whether the camera centre has moved? Consider two 3-space points which have coincident images in the first view, i.e. the points are on the same ray. If the camera centre is moved (not along that ray) the image coincidence is lost. This relative displacement of previously coincident image points is termed *parallax*, and is illustrated in figure 8.6 and shown schematically in figure 8.10. If the scene is static and motion parallax is evident between two views then the camera centre has moved. Indeed, a convenient method for obtaining a camera

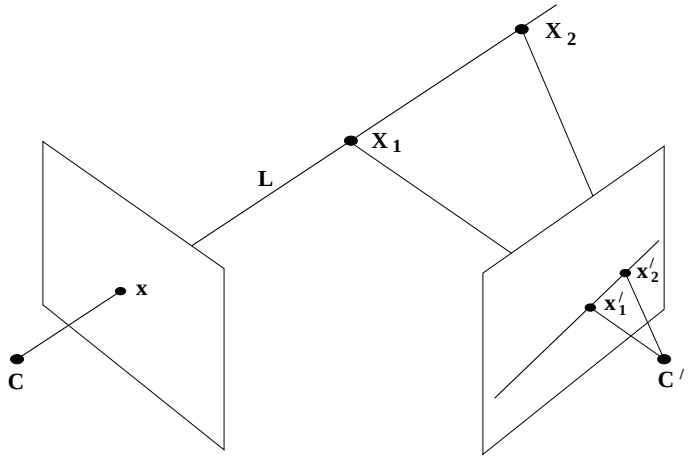


Fig. 8.10. **Motion parallax.** *The images of the space points $\mathbf{X}_1$ and $\mathbf{X}_2$ are coincident when viewed by the camera with centre $\mathbf{C}$. However, when viewed by a camera with centre $\mathbf{C}'$, which does not lie on the line $\mathbf{L}$ through $\mathbf{X}_1$ and $\mathbf{X}_2$, the images of the space points are not coincident. In fact the line through the image points $\mathbf{x}'_1$ and $\mathbf{x}'_2$ is the image of the ray $\mathbf{L}$, and will be seen in chapter 9 to be an* **epipolar line**. *The vector between the points $\mathbf{x}'_1$ and $\mathbf{x}'_2$ is the parallax.*

motion that is only a rotation about its centre (for example for a camera mounted on a robot head) is to adjust the motion until there is no parallax.

An important special case of 3-space structure is when all scene points are coplanar. In this case the images of corresponding points are related by a planar homography even if the camera centre is moved. The map between images in this case is discussed in detail in chapter 13 on planes. In particular vanishing points, which are images of points on the plane $\boldsymbol{\pi}_\infty$, are related by a planar homography for any camera motion. We return to this in section 8.6.

8.5 Camera calibration and the image of the absolute conic

Up to this point we have discussed projective properties of the forward and back-projection of various entities (point, lines, conics ...). These properties depend only on the 3×4 form of the projective camera matrix $\mathtt{P}$. Now we describe what is gained if the camera internal calibration, $\mathtt{K}$, is known. It will be seen that Euclidean properties, such as the angle between two rays, can then be measured.

What does calibration give? An image point $\mathbf{x}$ back-projects to a ray defined by $\mathbf{x}$ and the camera centre. Calibration relates the image point to the ray's *direction*. Suppose points on the ray are written as $\widetilde{\mathbf{X}} = \lambda\mathbf{d}$ in the camera Euclidean coordinate frame, then these points map to the point $\mathbf{x} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}](\lambda\mathbf{d}^\mathsf{T}, 1)^\mathsf{T} = \mathtt{K}\mathbf{d}$ up to scale. Conversely the direction $\mathbf{d}$ is obtained from the image point $\mathbf{x}$ as $\mathbf{d} = \mathtt{K}^{-1}\mathbf{x}$. Thus we have established:

Result 8.15. *The camera calibration matrix $\mathtt{K}$ is the (affine) transformation between $\mathbf{x}$ and the ray's direction $\mathbf{d} = \mathtt{K}^{-1}\mathbf{x}$ measured in the camera's Euclidean coordinate frame.*

Note, $\mathbf{d} = \mathtt{K}^{-1}\mathbf{x}$ is in general *not* a unit vector.

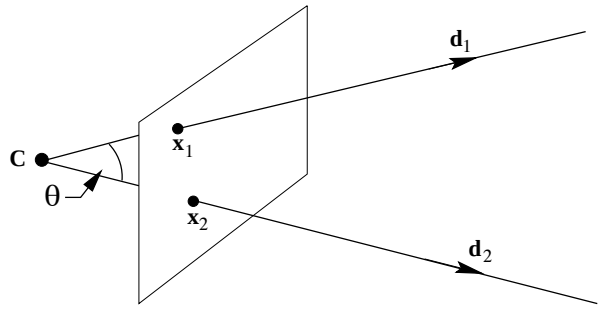


Fig. 8.11. *The angle θ between two rays.*

The angle between two rays, with directions $\mathbf{d}_1$, $\mathbf{d}_2$ corresponding to image points $\mathbf{x}_1, \mathbf{x}_2$ respectively, may be obtained from the familiar cosine formula for the angle between two vectors:

$$\begin{aligned}\cos\theta &= \frac{\mathbf{d}_1^\mathsf{T}\mathbf{d}_2}{\sqrt{\mathbf{d}_1^\mathsf{T}\mathbf{d}_1}\sqrt{\mathbf{d}_2^\mathsf{T}\mathbf{d}_2}} = \frac{(\mathtt{K}^{-1}\mathbf{x}_1)^\mathsf{T}(\mathtt{K}^{-1}\mathbf{x}_2)}{\sqrt{(\mathtt{K}^{-1}\mathbf{x}_1)^\mathsf{T}(\mathtt{K}^{-1}\mathbf{x}_1)}\sqrt{(\mathtt{K}^{-1}\mathbf{x}_2)^\mathsf{T}(\mathtt{K}^{-1}\mathbf{x}_2)}} \\ &= \frac{\mathbf{x}_1^\mathsf{T}(\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1})\mathbf{x}_2}{\sqrt{\mathbf{x}_1^\mathsf{T}(\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1})\mathbf{x}_1}\sqrt{\mathbf{x}_2^\mathsf{T}(\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1})\mathbf{x}_2}} \ . \end{aligned} \tag{8.7}$$

The formula (8.7) shows that if $\mathtt{K}$, and consequently the matrix $\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1}$, is known, then the angle between rays can be measured from their corresponding image points. A camera for which $\mathtt{K}$ is known is termed *calibrated*. A calibrated camera is a *direction sensor*, able to measure the direction of rays – like a 2D protractor.

The calibration matrix $\mathtt{K}$ also provides a relation between an image line and a scene plane:

Result 8.16. *An image line* $\mathbf{l}$ *defines a plane through the camera centre with normal direction* $\mathbf{n} = \mathtt{K}^\mathsf{T}\mathbf{l}$ *measured in the camera's Euclidean coordinate frame.*

Note, the normal $\mathbf{n}$ will not in general be a unit vector.

Proof. Points $\mathbf{x}$ on the line $\mathbf{l}$ back-project to directions $\mathbf{d} = \mathtt{K}^{-1}\mathbf{x}$ which are orthogonal to the plane normal $\mathbf{n}$, and thus satisfy $\mathbf{d}^\mathsf{T}\mathbf{n} = \mathbf{x}^\mathsf{T}\mathtt{K}^{-\mathsf{T}}\mathbf{n} = 0$. Since points on $\mathbf{l}$ satisfy $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$, it follows that $\mathbf{l} = \mathtt{K}^{-\mathsf{T}}\mathbf{n}$, and hence $\mathbf{n} = \mathtt{K}^\mathsf{T}\mathbf{l}$. □

8.5.1 The image of the absolute conic

We now derive a very important result which relates the calibration matrix $\mathtt{K}$ to the image of the absolute conic, $\boldsymbol{\omega}$. First we must determine the map between the plane at infinity, $\boldsymbol{\pi}_\infty$, and the camera image plane. Points on $\boldsymbol{\pi}_\infty$ may be written as $\mathbf{X}_\infty = (\mathbf{d}^\mathsf{T}, 0)^\mathsf{T}$, and are imaged by a general camera $\mathtt{P} = \mathtt{KR}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]$ as

$$\mathbf{x} = \mathtt{P}\mathbf{X}_\infty = \mathtt{KR}[\mathtt{I} \mid -\widetilde{\mathbf{C}}]\begin{pmatrix}\mathbf{d}\\0\end{pmatrix} = \mathtt{KR}\mathbf{d}.$$

This shows that

- *the mapping between $\boldsymbol{\pi}_\infty$ and an image is given by the planar homography $\mathbf{x} = \mathtt{H}\mathbf{d}$ with*

$$\mathtt{H} = \mathtt{K}\mathtt{R}. \tag{8.8}$$

Note that this map is independent of the position of the camera, $\mathbf{C}$, and depends only on the camera internal calibration and orientation with respect to the world coordinate frame.

Now, since the absolute conic Ω_∞ (section 3.6(*p*81)) is on $\boldsymbol{\pi}_\infty$ we can compute its image under $\mathtt{H}$, and find

Result 8.17. *The image of the absolute conic (the IAC) is the conic $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1} = \mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1}$.*

Proof. From result 2.13(*p*37) under a point homography $\mathbf{x} \mapsto \mathtt{H}\mathbf{x}$ a conic $\mathtt{C}$ maps as $\mathtt{C} \mapsto \mathtt{H}^{-\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}$. It follows that Ω_∞, which is the conic $\mathtt{C} = \Omega_\infty = \mathtt{I}$ on $\boldsymbol{\pi}_\infty$, maps to $\boldsymbol{\omega} = (\mathtt{K}\mathtt{R})^{-\mathsf{T}}\mathtt{I}(\mathtt{K}\mathtt{R})^{-1} = \mathtt{K}^{-\mathsf{T}}\mathtt{R}\mathtt{R}^{-1}\mathtt{K}^{-1} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$. So the IAC $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$. □

Like Ω_∞ the conic $\boldsymbol{\omega}$ is an imaginary point conic with no real points. For the moment it may be thought of as a convenient algebraic device, but it will be used in computations later in this chapter, and also in chapter 19 on camera auto-calibration.
A few remarks here:

(i) The image of the absolute conic, $\boldsymbol{\omega}$, depends only on the internal parameters $\mathtt{K}$ of the matrix $\mathtt{P}$; it does not depend on the camera orientation or position.

(ii) It follows from (8.7) that the angle between two rays is given by the simple expression

$$\cos\theta \;=\; \frac{\mathbf{x}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_2}{\sqrt{\mathbf{x}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_1}\sqrt{\mathbf{x}_2^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_2}}\;. \tag{8.9}$$

This expression is independent of the projective coordinate frame in the image, that is, it is unchanged under projective transformation of the image. To see this consider any 2D projective transformation, $\mathtt{H}$. The points $\mathbf{x}_i$ are transformed to $\mathtt{H}\mathbf{x}_i$, and $\boldsymbol{\omega}$ transforms (as any image conic) to $\mathtt{H}^{-\mathsf{T}}\boldsymbol{\omega}\mathtt{H}^{-1}$. Thus, (8.9) is unchanged, and hence holds in any projective coordinate frame in the image.

(iii) A particularly important specialization of (8.9) is that if two image points $\mathbf{x}_1$ and $\mathbf{x}_2$ correspond to orthogonal directions then

$$\mathbf{x}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_2 = 0. \tag{8.10}$$

This equation will be used at several points later in the book as it provides a linear constraint on $\boldsymbol{\omega}$.

(iv) We may also define the dual image of the absolute conic (the DIAC) as

$$\boldsymbol{\omega}^* = \boldsymbol{\omega}^{-1} = \mathtt{K}\mathtt{K}^\mathsf{T}. \tag{8.11}$$

This is a dual (line) conic, whereas $\boldsymbol{\omega}$ is a point conic (though it contains no real points). The conic $\boldsymbol{\omega}^*$ is the image of $\mathtt{Q}_\infty^*$ and is given by (8.5) $\boldsymbol{\omega}^* = \mathtt{P}\mathtt{Q}_\infty^*\mathtt{P}^\mathsf{T}$.

(v) Result 8.17 shows that once $\boldsymbol{\omega}$ (or equivalently $\boldsymbol{\omega}^*$) is identified in an image then $\mathtt{K}$ is also determined. This follows because a symmetric matrix $\boldsymbol{\omega}$ may be uniquely decomposed into a product $\boldsymbol{\omega}^* = \mathtt{K}\mathtt{K}^\mathsf{T}$ of an upper-triangular matrix with positive diagonal entries and its transpose by the Cholesky factorization (see result A4.5(*p*582)).

(vi) It was seen in chapter 3 that a plane $\boldsymbol{\pi}$ intersects $\boldsymbol{\pi}_\infty$ in a line, and this line intersects Ω_∞ in two points which are the circular points of $\boldsymbol{\pi}$. The imaged circular points lie on $\boldsymbol{\omega}$ at the points at which the vanishing line of the plane $\boldsymbol{\pi}$ intersects $\boldsymbol{\omega}$.

These final two properties of $\boldsymbol{\omega}$ are the basis for a calibration algorithm, as shown in the following example.

Example 8.18. A simple calibration device

The image of three squares (on planes which are not parallel, but which need not be orthogonal) provides sufficiently many constraints to compute $\mathtt{K}$. Consider one of the squares. The correspondences between its four corner points and their images define the homography $\mathtt{H}$ between the plane $\boldsymbol{\pi}$ of the square and the image. Applying this homography to circular points on $\boldsymbol{\pi}$ determines their images as $\mathtt{H}(1, \pm i, 0)^\mathsf{T}$. Thus we have two points on the (as yet unknown) $\boldsymbol{\omega}$. A similar procedure applied to the other squares generates a total of six points on $\boldsymbol{\omega}$, from which it may be computed (since five points are required to determine a conic). In outline the algorithm has the following steps:

(i) For each square compute the homography $\mathtt{H}$ that maps its corner points, $(0,0)^\mathsf{T}, (1,0)^\mathsf{T}, (0,1)^\mathsf{T}, (1,1)^\mathsf{T}$, to their imaged points. (The alignment of the plane coordinate system with the square is a similarity transformation and does not affect the position of the circular points on the plane).

(ii) Compute the imaged circular points for the plane of that square as $\mathtt{H}(1, \pm i, 0)^\mathsf{T}$. Writing $\mathtt{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$, the imaged circular points are $\mathbf{h}_1 \pm i\mathbf{h}_2$.

(iii) Fit a conic $\boldsymbol{\omega}$ to the six imaged circular points. The constraint that the imaged circular points lie on $\boldsymbol{\omega}$ may be rewritten as two real constraints. If $\mathbf{h}_1 \pm i\mathbf{h}_2$ lies on $\boldsymbol{\omega}$ then $(\mathbf{h}_1 \pm i\mathbf{h}_2)^\mathsf{T} \boldsymbol{\omega} (\mathbf{h}_1 \pm i\mathbf{h}_2) = 0$, and the imaginary and real parts give respectively:

$$\mathbf{h}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{h}_2 = 0 \quad \text{and} \quad \mathbf{h}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{h}_1 = \mathbf{h}_2^\mathsf{T}\boldsymbol{\omega}\mathbf{h}_2 \tag{8.12}$$

which are equations linear in $\boldsymbol{\omega}$. The conic $\boldsymbol{\omega}$ is determined up to scale from five or more such equations.

(iv) Compute the calibration $\mathtt{K}$ from $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$ using the Cholesky factorization.

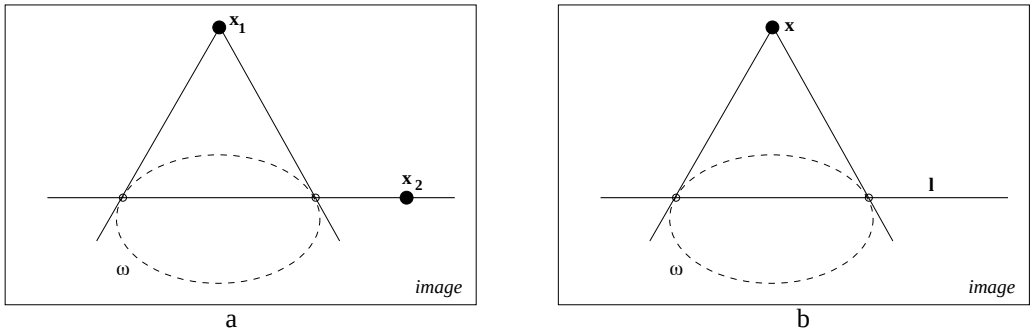
Figure 8.12 shows a calibration object consisting of three planes imprinted with squares, and the computed matrix $\mathtt{K}$. For the purpose of internal calibration, the squares have the advantage over a standard calibration object (e.g. figure 7.1(*p*182)) that no measured 3D co-ordinates are required. △

a

$$\mathtt{K} = \begin{bmatrix} 1108.3 & -9.8 & 525.8 \\ 0 & 1097.8 & 395.9 \\ 0 & 0 & 1 \end{bmatrix}$$

b

Fig. 8.12. **Calibration from metric planes.** *(a) Three squares provide a simple calibration object. The planes need not be orthogonal. (b) The computed calibration matrix using the algorithm of example 8.18. The image size is* 1024×768 *pixels.*



a b

Fig. 8.13. **Orthogonality represented by conjugacy and pole–polar relationships.** *(a) Image points* $\mathbf{x}_1, \mathbf{x}_2$ *back-project to orthogonal rays if the points are conjugate with respect to* $\boldsymbol{\omega}$*, i.e.* $\mathbf{x}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_2 = 0$*. (b) The point* $\mathbf{x}$ *and line* $\mathbf{l}$ *back-project to a ray and plane that are orthogonal if* $\mathbf{x}$ *and* $\mathbf{l}$ *are pole–polar with respect to* $\boldsymbol{\omega}$*, i.e.* $\mathbf{l} = \boldsymbol{\omega}\mathbf{x}$*. For example (see section 8.6.3), the vanishing point of the normal direction to a plane and the vanishing line of the plane are pole–polar with respect to* $\boldsymbol{\omega}$*.*

We will return to camera calibration in section 8.8, where vanishing points and lines provide constraints on $\mathtt{K}$. The geometric constraints that are used in example 8.18 are discussed further in section 8.8.1.

8.5.2 Orthogonality and $\boldsymbol{\omega}$

The conic $\boldsymbol{\omega}$ is a device for representing orthogonality in an image. It has already been seen (8.10) that if two image points $\mathbf{x}_1$ and $\mathbf{x}_2$ back-project to orthogonal rays, then the points satisfy $\mathbf{x}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_2 = 0$. Similarly, it may be shown that

Result 8.19. *A point* $\mathbf{x}$ *and line* $\mathbf{l}$ *back-projecting to a ray and plane respectively that are orthogonal are related by* $\mathbf{l} = \boldsymbol{\omega}\mathbf{x}$*.*

Geometrically these relations express that image points back-projecting to orthogonal rays are conjugate with respect to $\boldsymbol{\omega}$ ($\mathbf{x}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{x}_2 = 0$), and that a point and line back-projecting to an orthogonal ray and plane are in a pole–polar relationship ($\mathbf{l} = \boldsymbol{\omega}\mathbf{x}$). See section 2.8.1(*p*58). A schematic representation of these two relations is given in figure 8.13.

These geometric representations of orthogonality, and indeed the projective representation (8.9) of the angle between two rays measured from image points, are simply specializations and a recapitulation of relations derived earlier in the book. For example, we have already developed a projective representation (3.23–*p*82) of the angle between two lines in 3-space, namely

$$\cos\theta = \frac{\mathbf{d}_1^\mathsf{T}\Omega_\infty\mathbf{d}_2}{\sqrt{\mathbf{d}_1^\mathsf{T}\Omega_\infty\mathbf{d}_1}\sqrt{\mathbf{d}_2^\mathsf{T}\Omega_\infty\mathbf{d}_2}}$$

where $\mathbf{d}_1$ and $\mathbf{d}_2$ are the directions of the lines (which are the points at which the lines intersect $\boldsymbol{\pi}_\infty$). Rays are lines in 3-space which are coincident at the camera centre, and so (3.23–*p*82) may be applied directly to rays. This is precisely what (8.9) does – it is simply (3.23–*p*82) computed in the image.

Under the map (8.8) $\mathtt{H} = \mathtt{KR}$, which is the homography between the plane $\boldsymbol{\pi}_\infty$ in the world coordinate frame and the image plane, $\Omega_\infty \mapsto \mathtt{H}^\mathsf{T}\boldsymbol{\omega}\mathtt{H} = (\mathtt{KR})^\mathsf{T}\boldsymbol{\omega}(\mathtt{KR})$ and $\mathbf{d}_i = \mathtt{H}^{-1}\mathbf{x}_i = (\mathtt{KR})^{-1}\mathbf{x}_i$. Substituting these relations into (3.23–*p*82) gives (8.9). Similarly the conjugacy and pole–polar relations for orthogonality in the image are a direct image of those on $\boldsymbol{\pi}_\infty$, as can be seen by comparing figure 3.8(*p*83) with figure 8.13.

In practice these orthogonality results find greatest application in the case of vanishing points and vanishing lines.

8.6 Vanishing points and vanishing lines

One of the distinguishing features of perspective projection is that the image of an object that stretches off to infinity can have finite extent. For example, an infinite scene line is imaged as a line terminating in a *vanishing point*. Similarly, parallel world lines, such as railway lines, are imaged as converging lines, and their image intersection is the vanishing point for the direction of the railway.

8.6.1 Vanishing points

The perspective geometry that gives rise to vanishing points is illustrated in figure 8.14. It is evident that geometrically the vanishing point of a line is obtained by intersecting the image plane with a ray parallel to the world line and passing through the camera centre. Thus a vanishing point depends only on the *direction* of a line, not on its position. Consequently a set of parallel world lines have a common vanishing point, as illustrated in figure 8.16.

Algebraically the vanishing point may be obtained as a limiting point as follows: Points on a line in 3-space through the point $\mathbf{A}$ and with direction $\mathbf{D} = (\mathbf{d}^\mathsf{T}, 0)^\mathsf{T}$ are written as $\mathbf{X}(\lambda) = \mathbf{A} + \lambda\mathbf{D}$, see figure 8.14b. As the parameter λ varies from 0 to ∞ the point $\mathbf{X}(\lambda)$ varies from the finite point $\mathbf{A}$ to the point $\mathbf{D}$ at infinity. Under a projective camera $\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$, a point $\mathbf{X}(\lambda)$ is imaged at

$$\mathbf{x}(\lambda) = \mathtt{P}\mathbf{X}(\lambda) = \mathtt{P}\mathbf{A} + \lambda\mathtt{P}\mathbf{D} = \mathbf{a} + \lambda\mathtt{K}\mathbf{d}$$

where $\mathbf{a}$ is the image of $\mathbf{A}$. Then the vanishing point $\mathbf{v}$ of the line is obtained as the

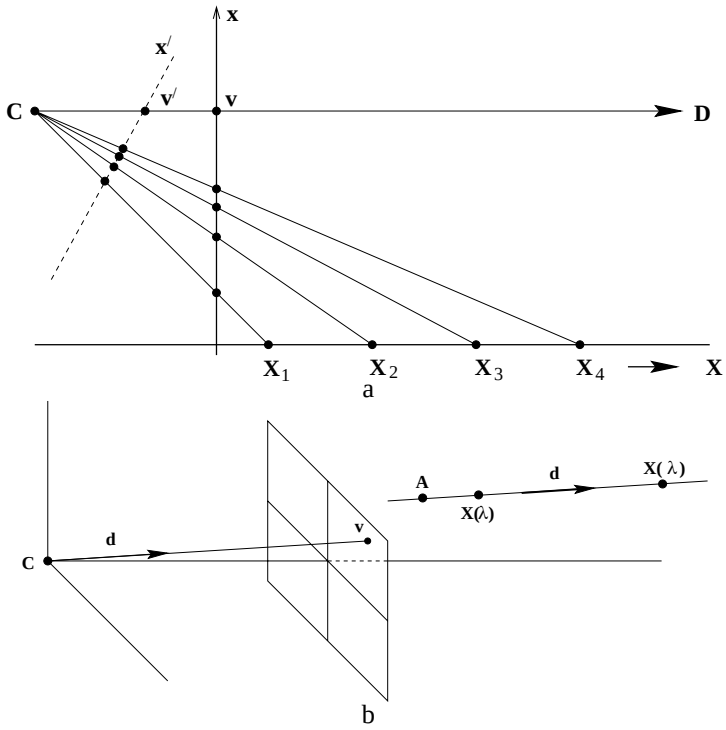


Fig. 8.14. **Vanishing point formation.** *(a) Plane to line camera. The points* $\mathbf{X}_i, i = 1, \ldots, 4$ *are equally spaced on the world line, but their spacing on the image line monotonically decreases. In the limit* $\mathbf{X} \to \infty$ *the world point is imaged at* $\mathbf{x} = \mathbf{v}$ *on the vertical image line, and at* $\mathbf{x}' = \mathbf{v}'$ *on the inclined image line. Thus the vanishing point of the world line is obtained by intersecting the image plane with a ray parallel to the world line through the camera centre* $\mathbf{C}$*. (b) 3-space to plane camera. The vanishing point,* $\mathbf{v}$*, of a line with direction* $\mathbf{d}$ *is the intersection of the image plane with a ray parallel to* $\mathbf{d}$ *through* $\mathbf{C}$*. The world line may be parametrized as* $\mathbf{X}(\lambda) = \mathbf{A} + \lambda\mathbf{D}$*, where* $\mathbf{A}$ *is a point on the line, and* $\mathbf{D} = (\mathbf{d}^\mathsf{T}, 0)^\mathsf{T}$*.*

limit

$$\mathbf{v} = \lim_{\lambda\to\infty} \mathbf{x}(\lambda) = \lim_{\lambda\to\infty} (\mathbf{a} + \lambda \mathtt{K}\mathbf{d}) = \mathtt{K}\mathbf{d}.$$

From result 8.15, $\mathbf{v} = \mathtt{K}\mathbf{d}$ means that the vanishing point $\mathbf{v}$ back-projects to a ray with direction $\mathbf{d}$. Note that $\mathbf{v}$ depends only on the direction $\mathbf{d}$ of the line, not on its position specified by $\mathbf{A}$.

In the language of projective geometry this result is obtained directly: In projective 3-space the plane at infinity $\boldsymbol{\pi}_\infty$ is the plane of directions, and all lines with the same direction intersect $\boldsymbol{\pi}_\infty$ in the same point (see chapter 3). The vanishing point is simply the image of this intersection. Thus if a line has direction $\mathbf{d}$, then it intersects $\boldsymbol{\pi}_\infty$ in the point $\mathbf{X}_\infty = (\mathbf{d}^\mathsf{T}, 0)^\mathsf{T}$. Then $\mathbf{v}$ is the image of $\mathbf{X}_\infty$

$$\mathbf{v} = \mathtt{P}\mathbf{X}_\infty = \mathtt{K}[\mathtt{I} \mid \mathbf{0}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = \mathtt{K}\mathbf{d}.$$

To summarize:

Result 8.20. *The vanishing point of lines with direction* $\mathbf{d}$ *in 3-space is the intersection*

$\mathbf{v}$ *of the image plane with a ray through the camera centre with direction* $\mathbf{d}$*, namely* $\mathbf{v} = \mathtt{K}\mathbf{d}$.

Note, lines parallel to the image plane are imaged as parallel lines, since $\mathbf{v}$ is at infinity in the image. However, the converse – that parallel image lines are the image of parallel scene lines – does not hold since lines which intersect on the principal plane are imaged as parallel lines.

Example 8.21. Camera rotation from vanishing points

Vanishing points are images of points at infinity, and provide orientation (attitude) information in a similar manner to that provided by the fixed stars. Consider two images of a scene obtained by calibrated cameras, where the two cameras differ in orientation *and position*. The points at infinity are part of the scene and so are independent of the camera. Their images, the vanishing points, are not affected by the change in camera position, but are affected by the camera rotation. Suppose both cameras have the same calibration matrix $\mathtt{K}$, and the camera rotates by $\mathtt{R}$ between views.

Let a scene line have vanishing point $\mathbf{v}_i$ in the first view, and $\mathbf{v}'_i$ in the second. The vanishing point $\mathbf{v}_i$ has direction $\mathbf{d}_i$ measured in the first camera's Euclidean coordinate frame, and the corresponding vanishing point $\mathbf{v}'_i$ has direction $\mathbf{d}'_i$ measured in the second camera's Euclidean coordinate frame. These directions can be computed from the vanishing points, for example $\mathbf{d}_i = \mathtt{K}^{-1}\mathbf{v}_i / \|\mathtt{K}^{-1}\mathbf{v}_i\|$, where the normalizing factor $\|\mathtt{K}^{-1}\mathbf{v}_i\|$ is included to ensure that $\mathbf{d}_i$ is a unit vector. The directions $\mathbf{d}_i$ and $\mathbf{d}'_i$ are related by the camera rotation as $\mathbf{d}'_i = \mathtt{R}\mathbf{d}_i$, which represents two independent constraints on $\mathtt{R}$. Thus the rotation matrix $\mathtt{R}$ can be computed from two such corresponding directions. △

The angle between two scene lines. We have seen that the vanishing point of a scene line back-projects to a ray parallel to the scene line. Consequently (8.9), which determines the angle between rays back-projected from image points, enables the angle between the directions of two scene lines to be measured from their vanishing points:

Result 8.22. *Let* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *be the vanishing points of two lines in an image, and let* $\boldsymbol{\omega}$ *be the image of the absolute conic in the image. If* θ *is the angle between the two line directions, then*

$$\cos\theta = \frac{\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_2}{\sqrt{\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_1}\sqrt{\mathbf{v}_2^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_2}} \quad . \tag{8.13}$$

A note on computing vanishing points

Often vanishing points are computed from the image of a set of parallel line segments, though they may be determined in other ways for example by using equal length intervals on a line as described in example 2.18(*p*50) and example 2.20(*p*51). In the case of imaged parallel line segments the objective is to estimate their common image intersection – which is the image of the direction of the parallel scene lines. Due to measurement noise the imaged line segments will generally *not* intersect in a unique

Fig. 8.15. **ML estimate of a vanishing point from imaged parallel scene lines.** *(a) Estimating the vanishing point* $\mathbf{v}$ *involves fitting a line (shown thin here) through* $\mathbf{v}$ *to each measured line (shown thick here). The ML estimate of* $\mathbf{v}$ *is the point which minimizes the sum of squared orthogonal distances between the fitted lines and the measured lines' end points. (b) Measured line segments are shown in white, and fitted lines in black. (c) A close-up of the dashed square in (b). Note the very slight angle between the measured and fitted lines.*

point. Commonly the vanishing point is then computed by intersecting the lines pairwise and using the centroid of these intersections, or finding the closest point to all the measured lines. However, these are *not* optimal procedures.

Under the assumption of Gaussian measurement noise, the maximum likelihood estimate (MLE) of the vanishing point *and* line segments is computed by determining a set of lines that do intersect in a single point, and which minimize the sum of squared orthogonal distances from the endpoints of the measured line segments as shown in figure 8.15(a). This minimization may be computed numerically using the Levenberg–Marquardt algorithm (section A6.2(*p*600)). Note that if the lines are defined by fitting to many points, rather than just their end points, one can use the method described in section 16.7.2(*p*404) to reduce each line to an equivalent pair of weighted end points which can then be used in this algorithm. Figure 8.15(b)(c) shows an example of a vanishing point computed in this manner. It is evident that the residuals between the measured and fitted lines are very small.

8.6.2 Vanishing lines

Parallel planes in 3-space intersect $\boldsymbol{\pi}_\infty$ in a common line, and the image of this line is the vanishing line of the plane. Geometrically the vanishing line is constructed, as shown in figure 8.16, by intersecting the image with a plane parallel to the scene plane through the camera centre. It is clear that a vanishing line depends only on the *orientation* of the scene plane; it does not depend on its position. Since lines parallel to a plane intersect the plane at $\boldsymbol{\pi}_\infty$, it is easily seen that the vanishing point of a line parallel to a plane lies on the vanishing line of the plane. An example is shown in figure 8.17.

If the camera calibration $\mathtt{K}$ is known then a scene plane's vanishing line may be used to determine information about the plane, and we mention three examples here:

(i) The plane's orientation relative to the camera may be determined from its vanishing line. From result 8.16 a plane through the camera centre with normal

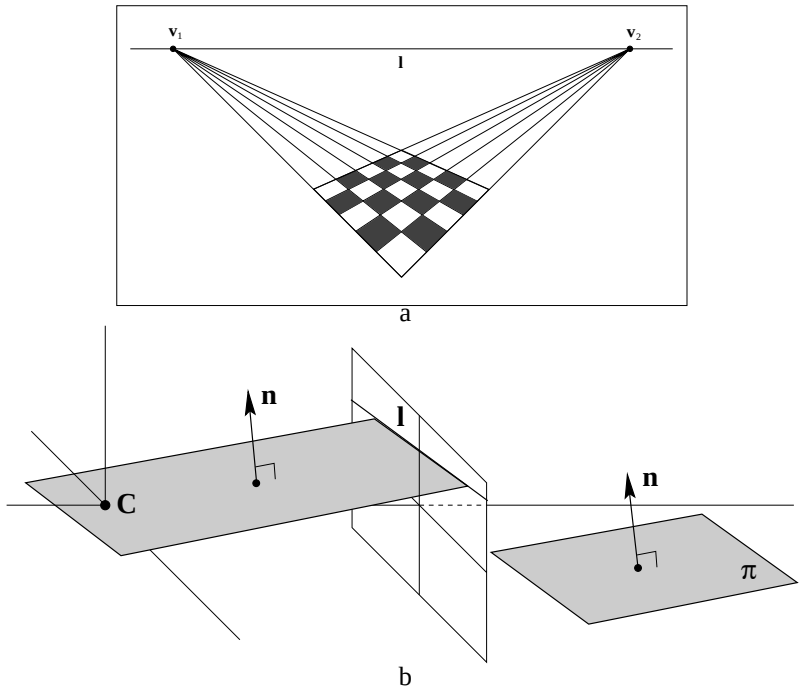


Fig. 8.16. **Vanishing line formation.** *(a) The two sets of parallel lines on the scene plane converge to the vanishing points* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *in the image. The line* $\mathbf{l}$ *through* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *is the vanishing line of the plane. (b) The vanishing line* $\mathbf{l}$ *of a plane* $\boldsymbol{\pi}$ *is obtained by intersecting the image plane with a plane through the camera centre* $\mathbf{C}$ *and parallel to* $\boldsymbol{\pi}$.

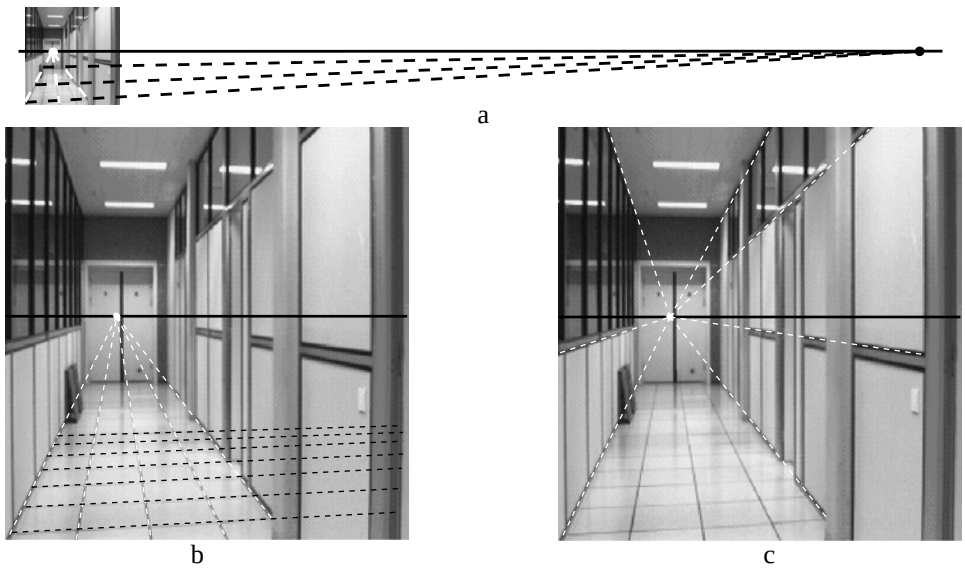


Fig. 8.17. **Vanishing points and lines.** *The vanishing line of the ground plane (the horizon) of the corridor may be obtained from two sets of parallel lines on the plane. (a) The vanishing points of lines which are nearly parallel to the image plane are distant from the finite (actual) image. (b) Note the monotonic decrease in the spacing of the imaged equally spaced parallel lines corresponding to the sides of the floor tiles. (c) The vanishing point of lines parallel to a plane (here the ground plane) lies on the vanishing line of the plane.*

direction $\mathbf{n}$ intersects the image plane in the line $\mathbf{l} = \mathtt{K}^{-\mathsf{T}}\mathbf{n}$. Consequently, $\mathbf{l}$ is the vanishing line of planes perpendicular to $\mathbf{n}$. Thus a plane with vanishing line $\mathbf{l}$ has orientation $\mathbf{n} = \mathtt{K}^{\mathsf{T}}\mathbf{l}$ in the camera's Euclidean coordinate frame.

(ii) The plane may be metrically rectified given only its vanishing line. This can be seen by considering a synthetic rotation of the camera in the manner of example 8.13(*p*205). Since the plane normal is known from the vanishing line, the camera can be synthetically rotated by a homography so that the plane is fronto-parallel (i.e. parallel to the image plane). The computation of this homography is discussed in exercise (ix).

(iii) The angle between two scene planes can be determined from their vanishing lines. Suppose the vanishing lines are $\mathbf{l}_1$ and $\mathbf{l}_2$, then the angle θ between the planes is given by

$$\cos\theta = \frac{\mathbf{l}_1^{\mathsf{T}}\boldsymbol{\omega}^*\mathbf{l}_2}{\sqrt{\mathbf{l}_1^{\mathsf{T}}\boldsymbol{\omega}^*\mathbf{l}_1}\sqrt{\mathbf{l}_2^{\mathsf{T}}\boldsymbol{\omega}^*\mathbf{l}_2}}. \tag{8.14}$$

The proof is left as an exercise.

Computing vanishing lines

A common way to determine a vanishing line of a scene plane is first to determine vanishing points for two sets of lines parallel to the plane, and then to construct the line through the two vanishing points. This construction is illustrated in figure 8.17. Alternative methods of determining vanishing points are shown in example 2.19(*p*51) and example 2.20(*p*51).

However, the vanishing line may be determined directly, without using vanishing points as an intermediate step. For example, the vanishing line may be computed given an imaged set of equally spaced coplanar parallel lines. This is a useful method in practice because such sets commonly occur in man-made structures, such as: stairs, windows on the wall of a building, fences, radiators and zebra crossings. The following example illustrates the projective geometry involved.

Example 8.23. The vanishing line given the image of three coplanar equally spaced parallel lines

A set of equally spaced lines on the scene plane may be represented as $ax' + by' + \lambda = 0$, where λ takes integer values. This set (a pencil) of lines may be written as $\mathbf{l}'_n = (a, b, n)^{\mathsf{T}} = (a, b, 0)^{\mathsf{T}} + n(0, 0, 1)^{\mathsf{T}}$, where $(0, 0, 1)^{\mathsf{T}}$ is the line at infinity on the scene plane. Under perspective imaging the point transformation is $\mathbf{x} = \mathtt{H}\mathbf{x}'$, and the corresponding line map is $\mathbf{l}_n = \mathtt{H}^{-\mathsf{T}}\mathbf{l}'_n = \mathbf{l}_0 + n\mathbf{l}$, where $\mathbf{l}$, the image of $(0, 0, 1)^{\mathsf{T}}$, is the vanishing line of the plane. The imaged geometry is shown in figure 8.18(c). Note all lines $\mathbf{l}_n$ intersect in a common vanishing point (which is given by $\mathbf{l}_i \times \mathbf{l}_j$, for $i \neq j$) and the spacing decreases monotonically with n. The vanishing line $\mathbf{l}$ may be determined from three lines of the set provided their index (n) is identified. For example, from the image of three equally spaced lines, $\mathbf{l}_0, \mathbf{l}_1$ and $\mathbf{l}_2$, the closed form solution for the vanishing line is:

$$\mathbf{l} = \Big((\mathbf{l}_0 \times \mathbf{l}_2)^{\mathsf{T}}(\mathbf{l}_1 \times \mathbf{l}_2)\Big)\,\mathbf{l}_1 + 2\,\Big((\mathbf{l}_0 \times \mathbf{l}_1)^{\mathsf{T}}(\mathbf{l}_2 \times \mathbf{l}_1)\Big)\,\mathbf{l}_2. \tag{8.15}$$

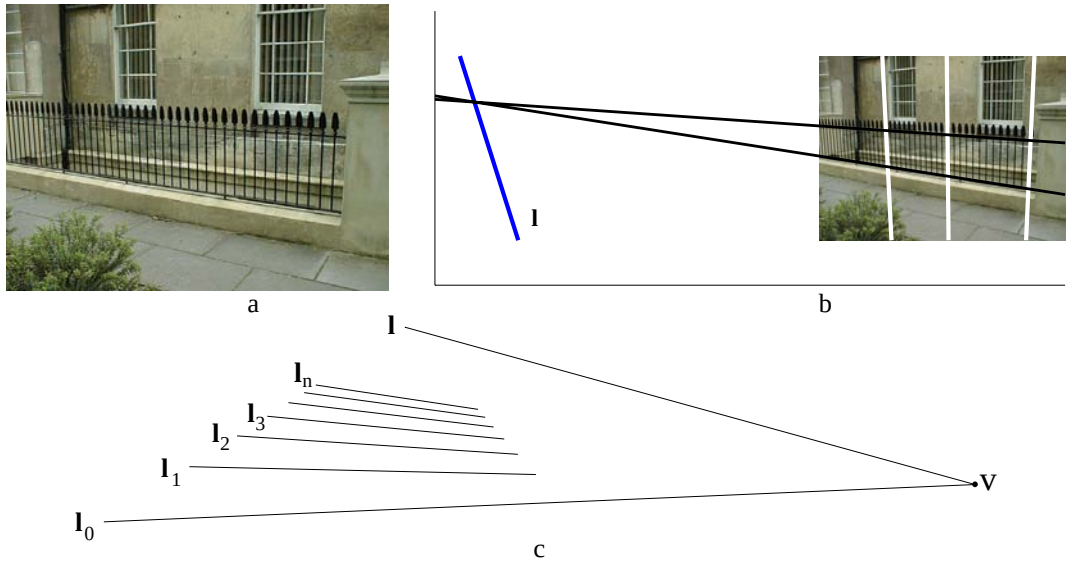


Fig. 8.18. **Determining a planes vanishing line from imaged equally spaced parallel lines.** *(a) Image of a vertical fence with equally spaced bars. (b) The computed vanishing line* $\mathbf{l}$ *from three equally spaced bars (12 apart). Note the vanishing point of the horizontal lines lies on this vanishing line. (c) The spacing between the imaged lines* $\mathbf{l}_n$ *monotonically decreases with* n.

The proof is left as an exercise. Figure 8.18(b) shows a vanishing line computed in this way. △

8.6.3 Orthogonality relationships amongst vanishing points and lines

It is often the case in practice that the lines and planes giving rise to vanishing points are orthogonal. In this case there are particularly simple relationships amongst their vanishing points and lines involving $\boldsymbol{\omega}$, and furthermore these relations can be used to (partially) determine $\boldsymbol{\omega}$, and consequently the camera calibration $\mathtt{K}$ as will be seen in section 8.8.

It follows from (8.13) that the vanishing points, $\mathbf{v}_1, \mathbf{v}_2$, of two perpendicular world lines satisfy $\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_2 = 0$. This means that the vanishing points are conjugate with respect to $\boldsymbol{\omega}$, as illustrated in figure 8.13. Similarly it follows from result 8.19 that the vanishing point $\mathbf{v}$ of a direction perpendicular to a plane with vanishing line $\mathbf{l}$ satisfies $\mathbf{l} = \boldsymbol{\omega}\mathbf{v}$. This means that the vanishing point and line are in a pole–polar relation with respect to $\boldsymbol{\omega}$, as is also illustrated in figure 8.13. Summarizing these image relations:

(i) The vanishing points of lines with perpendicular directions satisfy

$$\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_2 = 0. \tag{8.16}$$

(ii) If a line is perpendicular to a plane then their respective vanishing point $\mathbf{v}$ and vanishing line $\mathbf{l}$ are related by

$$\mathbf{l} = \boldsymbol{\omega}\mathbf{v} \tag{8.17}$$

and inversely $\mathbf{v} = \boldsymbol{\omega}^*\mathbf{l}$.

(iii) The vanishing lines of two perpendicular planes satisfy $\mathbf{l}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{l}_2 = 0$.

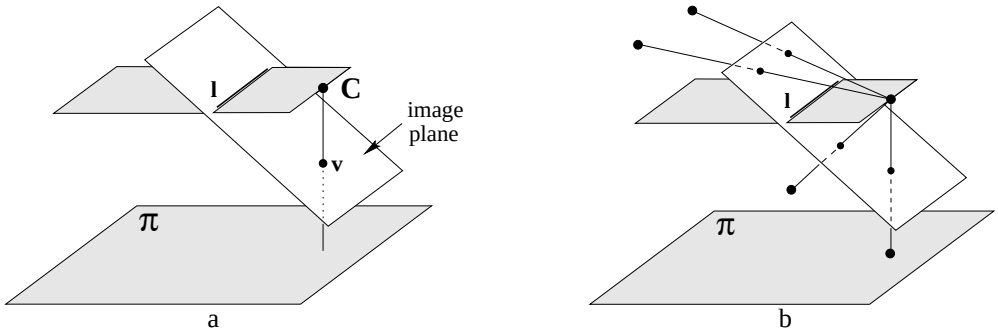


Fig. 8.19. **Geometry of a vertical vanishing point and ground plane vanishing line.** *(a) The vertical vanishing point* $\mathbf{v}$ *is the image of the vertical "footprint" of the camera centre on the ground plane* π*. (b) The vanishing line* $\mathbf{l}$ *partitions all points in scene space. Any scene point projecting onto the vanishing line is at the same distance from the plane* π *as the camera centre; if it lies "above" the line it is farther from the plane, and if "below" then it is closer to the plane than the camera centre.*

For example, suppose the vanishing line $\mathbf{l}$ of the ground plane (the horizon) is identified in an image, and the internal calibration matrix $\mathtt{K}$ is known, then the vertical vanishing point $\mathbf{v}$ (which is the vanishing point of the normal direction to the plane) may be obtained from $\mathbf{v} = \boldsymbol{\omega}^*\mathbf{l}$.

8.7 Affine 3D measurements and reconstruction

It has been seen in section 2.7.2(*p*49) that identifying a scene plane's vanishing line allows affine properties of the scene plane to be measured. If in addition a vanishing point for a direction not parallel to the plane is identified, then affine properties can be computed for the 3-space of the perspectively imaged scene. We will illustrate this idea for the case where the vanishing point corresponds to a direction orthogonal to the plane, although orthogonality is not necessary for the construction. The method described in this section does not require that the internal calibration of the camera $\mathtt{K}$ be known.

It will be convenient to think of the scene plane as the horizontal ground plane, in which case the vanishing line is the *horizon*. Similarly, it will be convenient to think of the direction orthogonal to the scene plane as vertical, so that $\mathbf{v}$ is the vertical vanishing point. This situation is illustrated in figure 8.19.

Suppose we wish to measure the relative lengths of two line segments in the vertical direction as shown in figure 8.20(a). We will show the following result:

Result 8.24. *Given the vanishing line of the ground plane* $\mathbf{l}$ *and the vertical vanishing point* $\mathbf{v}$*, then the relative length of vertical line segments can be measured provided their end point lies on the ground plane.*

Clearly the relative lengths cannot be measured directly from their imaged lengths because as a vertical line recedes deeper into the scene (i.e. further from the camera) then its imaged length decreases. The construction to determine the relative lengths proceeds in two steps:

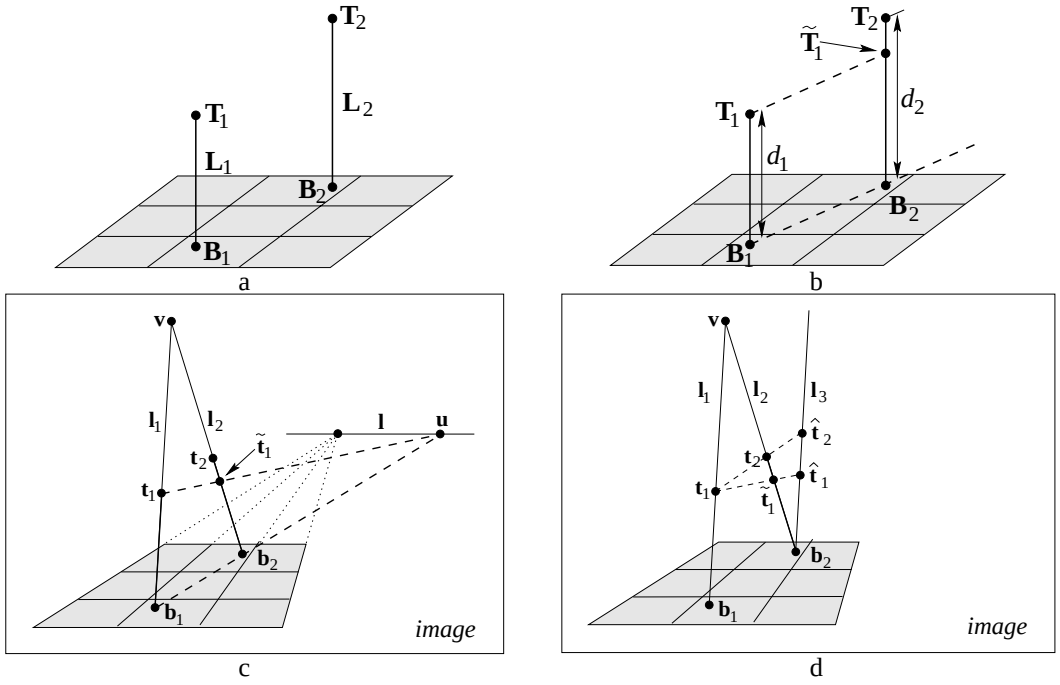


Fig. 8.20. **Computing length ratios of parallel scene lines.** *(a)* **3D geometry:** *The vertical line segments* $\mathbf{L}_1 = \langle \mathbf{B}_1, \mathbf{T}_1 \rangle$ *and* $\mathbf{L}_2 = \langle \mathbf{B}_2, \mathbf{T}_2 \rangle$ *have length* d_1 *and* d_2 *respectively. The base points* $\mathbf{B}_1, \mathbf{B}_2$ *are on the ground plane. We wish to compute the scene length ratio* $d_1 : d_2$ *from the imaged configuration. (b) In the scene the length of the line segment* $\mathbf{L}_1$ *may be transferred to* $\mathbf{L}_2$ *by constructing a line parallel to the ground plane to generate the point* $\widetilde{\mathbf{T}}_1$*. (c)* **Image geometry:** $\mathbf{l}$ *is the ground plane vanishing line, and* $\mathbf{v}$ *the vertical vanishing point. A corresponding parallel line construction in the image requires first determining the vanishing point* $\mathbf{u}$ *from the images* $\mathbf{b}_i$ *of* $\mathbf{B}_i$*, and then determining* $\tilde{\mathbf{t}}_1$ *(the image of* $\widetilde{\mathbf{T}}_1$*) by the intersection of* $\mathbf{l}_2$ *and the line* $\langle \mathbf{t}_1, \mathbf{u} \rangle$*. (d) The line* $\mathbf{l}_3$ *is parallel to* $\mathbf{l}_1$ *in the image. The points* $\hat{\mathbf{t}}_1$ *and* $\hat{\mathbf{t}}_2$ *are constructed by intersecting* $\mathbf{l}_3$ *with the lines* $\langle \mathbf{t}_1, \tilde{\mathbf{t}}_1 \rangle$ *and* $\langle \mathbf{t}_1, \mathbf{t}_2 \rangle$ *respectively. The distance ratio* $d(\mathbf{b}_2, \hat{\mathbf{t}}_1) : d(\mathbf{b}_2, \hat{\mathbf{t}}_2)$ *is the computed estimate of* $d_1 : d_2$*.*

Step 1: Map the length of one line segment onto the other. In 3D the length of $\mathbf{L}_1$ may be compared to $\mathbf{L}_2$ by constructing a line parallel to the ground plane in the direction $\langle \mathbf{B}_1, \mathbf{B}_2 \rangle$ that transfers $\mathbf{T}_1$ onto $\mathbf{L}_2$. This transferred point will be denoted $\widetilde{\mathbf{T}}_1$ (see figure 8.20(b)). In the image a corresponding construction is carried out by first determining the vanishing point $\mathbf{u}$ which is the intersection of $\langle \mathbf{b}_1, \mathbf{b}_2 \rangle$ with $\mathbf{l}$. Now any scene line parallel to $\langle \mathbf{B}_1, \mathbf{B}_2 \rangle$ is imaged as a line through $\mathbf{u}$, so in particular the image of the line through $\mathbf{T}_1$ parallel to $\langle \mathbf{B}_1, \mathbf{B}_2 \rangle$ is the line through $\mathbf{t}_1$ and $\mathbf{u}$. The intersection of the line $\langle \mathbf{t}_1, \mathbf{u} \rangle$ with $\mathbf{l}_2$ defines the image $\tilde{\mathbf{t}}_1$ of the transferred point $\widetilde{\mathbf{T}}_1$ (see figure 8.20(c)).

Step 2: Determine the ratio of lengths on the scene line. We now have four collinear points on an imaged scene line and wish to determine the actual length ratio in the scene. The four collinear image points are $\mathbf{b}_2, \tilde{\mathbf{t}}_1, \mathbf{t}_2$ and $\mathbf{v}$. These may be treated as images of scene points at distances $0, d_1, d_2$ and ∞, respectively, along the scene line. The affine ratio $d_1 : d_2$ may be obtained by applying a projective transfor-

Objective

Given the vanishing line of the ground plane $\mathbf{l}$ and the vertical vanishing point $\mathbf{v}$ and the top $(\mathbf{t}_1, \mathbf{t}_2)$ and base $(\mathbf{b}_1, \mathbf{b}_2)$ points of two line segments as in figure 8.20, compute the ratio of lengths of the line segments in the scene.

Algorithm

(i) Compute the vanishing point $\mathbf{u} = (\mathbf{b}_1 \times \mathbf{b}_2) \times \mathbf{l}$.
(ii) Compute the transferred point $\tilde{\mathbf{t}}_1 = (\mathbf{t}_1 \times \mathbf{u}) \times \mathbf{l}_2$ (where $\mathbf{l}_2 = \mathbf{v} \times \mathbf{b}_2$).
(iii) Represent the four points $\mathbf{b}_2, \tilde{\mathbf{t}}_1, \mathbf{t}_2$ and $\mathbf{v}$ on the image line $\mathbf{l}_1$ by their distance from $\mathbf{b}_2$, as $0, \tilde{t}_1, t_2$ and v respectively.
(iv) Compute a 1D projective transformation $\mathtt{H}_{2\times2}$ mapping homogeneous coordinates $(0,1) \mapsto (0,1)$ and $(v,1) \mapsto (1,0)$ (which maps the vanishing point $\mathbf{v}$ to infinity). A suitable matrix is given by

$$\mathtt{H}_{2\times2} = \begin{bmatrix} 1 & 0 \\ 1 & -v \end{bmatrix}.$$

(v) The (scaled) distance of the scene points $\widetilde{\mathbf{T}}_1$ and $\mathbf{T}_2$ from $\mathbf{B}_2$ on $\mathbf{L}_2$ may then be obtained from the position of the points $\mathtt{H}_{2\times2}(\tilde{t}_1, 1)^\mathsf{T}$ and $\mathtt{H}_{2\times2}(t_2, 1)^\mathsf{T}$. Their distance ratio is then given by

$$\frac{d_1}{d_2} = \frac{\tilde{t}_1(v - t_2)}{t_2(v - \tilde{t}_1)}$$

Algorithm 8.1. *Computing scene length ratios from a single image.*

mation to the image line which maps $\mathbf{v}$ to infinity. A geometric construction of this projectivity is shown in figure 8.20(d) (see example 2.20(*p*51)).

Details of the algorithm to carry out these two steps are given in algorithm 8.1.

Note, no knowledge of the camera calibration $\mathtt{K}$ or pose is necessary to apply the algorithm. In fact, the position of the camera centre relative to the ground plane can also be computed. The algorithm is well conditioned even when the vanishing point and/or line are at infinity in the image. For example, under affine image conditionings, or if the image plane is parallel to the vertical scene direction (so that $\mathbf{v}$ is at infinity). In these cases the distance ratio simplifies to $\frac{d_1}{d_2} = \frac{\tilde{t}_1}{t_2}$.

Example 8.25. Measuring a person's height in a single image

Suppose we have an image which contains sufficient information to compute the ground plane vanishing line and the vertical vanishing point, and also one object of known height for which the top and base are imaged. Then the height of a person standing on the ground plane can be measured anywhere in the scene provided that their head and feet are both visible. Figure 8.21(a) shows an example. The scene contains plenty of horizontal lines from which to compute a horizontal vanishing point. Two such vanishing points determine the vanishing line of the floor (which is the horizon for this image). The scene also contains plenty of vertical lines from which to compute a vertical vanishing point (figure 8.21(c)). Assuming that the two people are standing vertically, then their relative height may be computed directly from their length ratio using algorithm 8.1. Their absolute height may be determined by comput-

Fig. 8.21. **Height measurements using affine properties.** *(a) The original image. We wish to measure the height of the two people. (b) The image after radial distortion correction (see section 7.4*(p189)*). (c) The vanishing line (shown) is computed from two vanishing points corresponding to horizontal directions. The lines used to compute the vertical vanishing points are also shown. The vertical vanishing point is not shown since it lies well below the image. (d) Using the known height of the filing cabinet on the left of the image, the absolute height of the two people are measured as described in algorithm 8.1. The measured heights are within 2cm of ground truth. The computation of the uncertainty is described in [Criminisi-00].*

ing their height relative to an object on the ground plane with known height. Here the known height is provided by the filing cabinet. The result is shown in figure 8.21(d).

△

8.8 Determining camera calibration K from a single view

We have seen that once $\boldsymbol{\omega}$ is known the angle between rays can be measured. Conversely if the angle between rays is known then a constraint is placed on $\boldsymbol{\omega}$. Each known angle between two rays gives a constraint of the form (8.13) on $\boldsymbol{\omega}$. Unfortunately, for arbitrary angles, and known $\mathbf{v}_1$ and $\mathbf{v}_2$, this gives a quadratic constraint on the entries of $\boldsymbol{\omega}$. If the lines are perpendicular, however, (8.13) reduces to (8.16) $\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_2 = 0$, and the constraint on $\boldsymbol{\omega}$ is linear.

A linear constraint on $\boldsymbol{\omega}$ also results from a vanishing point and vanishing line arising from a line and its orthogonal plane. A common example is a vertical direction and horizontal plane as in figure 8.19. From (8.17) $\mathbf{l} = \boldsymbol{\omega}\mathbf{v}$. Writing this as $\mathbf{l} \times (\boldsymbol{\omega}\mathbf{v}) = \mathbf{0}$

Condition	constraint	type	# constraints
vanishing points $\mathbf{v}_1$, $\mathbf{v}_2$ corresponding to orthogonal lines	$\mathbf{v}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_2 = 0$	linear	1
vanishing point $\mathbf{v}$ and vanishing line $\mathbf{l}$ corresponding to orthogonal line and plane	$[\mathbf{l}]_\times\boldsymbol{\omega}\mathbf{v} = \mathbf{0}$	linear	2
metric plane imaged with known homography $\mathtt{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$	$\mathbf{h}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{h}_2 = 0$ $\mathbf{h}_1^\mathsf{T}\boldsymbol{\omega}\mathbf{h}_1 = \mathbf{h}_2^\mathsf{T}\boldsymbol{\omega}\mathbf{h}_2$	linear	2
zero skew	$\omega_{12} = \omega_{21} = 0$	linear	1
square pixels	$\omega_{12} = \omega_{21} = 0$ $\omega_{11} = \omega_{22}$	linear	2

Table 8.1. **Scene and internal constraints on $\boldsymbol{\omega}$.**

removes the homogeneous scaling factor and results in three homogeneous equations linear in the entries of $\boldsymbol{\omega}$. These are equivalent to two independent constraints on $\boldsymbol{\omega}$.

All these conditions provide linear constraints on $\boldsymbol{\omega}$. Given a sufficient number of such constraints $\boldsymbol{\omega}$ may be computed and hence the camera calibration $\mathtt{K}$ also follows since $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$.

The number of entries of $\boldsymbol{\omega}$ that need be determined from scene constraints of this sort can be reduced if the calibration matrix $\mathtt{K}$ has a more specialized form than (6.10–*p*157). In the case where $\mathtt{K}$ is known to have zero skew ($s = 0$), or square pixels ($\alpha_x = \alpha_y$ and $s = 0$), we can take advantage of this condition to help find $\boldsymbol{\omega}$. In particular, it is quickly verified by direct computation that:

Result 8.26. *If $s = \mathtt{K}_{12} = 0$ then $\omega_{12} = \omega_{21} = 0$. If in addition $\alpha_x = \mathtt{K}_{11} = \mathtt{K}_{22} = \alpha_y$, then $\omega_{11} = \omega_{22}$.*

Thus, in solving for the image of the absolute conic, one may easily take into account the zero-skew or square-aspect ratio constraint on the camera, if such a constraint is known to exist. One may also verify that no such simple connection as result 8.26 exists between the entries of $\mathtt{K}$ and those of $\boldsymbol{\omega}^* = \mathtt{K}\mathtt{K}^\mathsf{T}$.

We have now seen three sources of constraints on $\boldsymbol{\omega}$:

(i) metric information on a plane imaged with a known homography, see (8.12–*p*211)
(ii) vanishing points and lines corresponding to perpendicular directions and planes, (8.16)
(iii) "internal constraints" such as zero skew or square pixels, as in result 8.26

These constraints are summarized in table 8.1. We now describe how these constraints may be combined to estimate $\boldsymbol{\omega}$ and thence $\mathtt{K}$.

Since all the above constraints (including the internal constraints) are described algebraically as linear equations on $\boldsymbol{\omega}$, it is a simple matter to combine them as rows of

Objective

Compute K via $\boldsymbol{\omega}$ by combining scene and internal constraints.

Algorithm

(i) Represent $\boldsymbol{\omega}$ as a homogeneous 6-vector $\mathbf{w} = (w_1, w_2, w_3, w_4, w_5, w_6)^\mathsf{T}$ where:

$$\boldsymbol{\omega} = \begin{bmatrix} w_1 & w_2 & w_4 \\ w_2 & w_3 & w_5 \\ w_4 & w_5 & w_6 \end{bmatrix}$$

(ii) Each available constraint from table 8.1 may be written as $\mathbf{a}^\mathsf{T}\mathbf{w} = 0$. For example, for the orthogonality constraint $\mathbf{u}^\mathsf{T}\boldsymbol{\omega}\mathbf{v} = 0$, where $\mathbf{u} = (u_1,\ u_2,\ u_3)^\mathsf{T}$ and $\mathbf{v} = (v_1,\ v_2,\ v_3)^\mathsf{T}$, the 6-vector $\mathbf{a}$ is given by

$$\mathbf{a} = (v_1u_1, v_1u_2 + v_2u_1, v_2u_2, v_1u_3 + v_3u_1, v_2u_3 + v_3u_2, v_3u_3)^\mathsf{T}.$$

Similar constraints vectors are obtained from the other sources of scene and internal constraints. For example a metric plane generates two such constraints.

(iii) Stack the equations $\mathbf{a}^\mathsf{T}\mathbf{w} = 0$ from each constraint in the form $\mathtt{A}\mathbf{w} = \mathbf{0}$, where A is a $n \times 6$ matrix for n constraints.

(iv) Solve for $\mathbf{w}$ using the SVD as in algorithm 4.2(*p*109). This determines $\boldsymbol{\omega}$.

(v) Decompose $\boldsymbol{\omega}$ into K using matrix inversion and Cholesky factorization (see section A4.2.1(*p*582)).

Algorithm 8.2. *Computing* K *from scene and internal constraints.*

a constraint matrix. All constraints may be collected together so that for n constraints the system of equations may be written as $\mathtt{A}\mathbf{w} = \mathbf{0}$, where A is a $n \times 6$ matrix and $\mathbf{w}$ is a 6-vector containing the six distinct homogeneous entries of $\boldsymbol{\omega}$. With a minimum of 5 constraint equations an exact solution is found. With more than five equations, a least-squares solution is found by algorithm A5.4(*p*593). The method is summarized in algorithm 8.2.

With more than the minimum required five constraints, we have the option to apply some of the constraints as hard constraints – that is, constraints that will be satisfied exactly. This can be done by parametrizing $\boldsymbol{\omega}$ so that the constraints are satisfied explicitly (for instance setting $\omega_{21} = \omega_{12} = 0$ for the zero skew constraint, and also $\omega_{11} = \omega_{22}$ for the square-pixel constraint). The minimization method of algorithm A5.5(*p*594) may also be used to enforce hard constraints. Otherwise, treating all constraints as soft constraints and using algorithm A5.4(*p*593) will produce a solution in which the constraints are not satisfied exactly in the presence of noise – for instance, pixels may not be quite square.

Finally, an important issue in practice is that of degeneracy. This occurs when the combined constraints are not independent and results in the matrix A dropping rank. If the rank is less than the number of unknowns, then a parametrized family of solutions for $\boldsymbol{\omega}$ (and hence K) is obtained. Also, if conditions are near degenerate then the solution is ill-conditioned and the particular member of the family is determined by "noise". These degeneracies can often be understood geometrically – for example in example 8.18 if the three metric planes are parallel then the three pairs of imaged

Fig. 8.22. *For the case that image skew is zero and the aspect ratio unity* **the principal point is the orthocentre of an orthogonal triad of vanishing points**. *(a) Original image. (b) Three sets of parallel lines in the scene, with each set having direction orthogonal to the others. (c) The principal point is the orthocentre of the triangle with the vanishing points as vertices.*

circular points are coincident and only provide a total of two constraints instead of six. A pragmatic solution to the problem of degeneracy, popularized by Zhang [Zhang-00], is to image a metric plane many times in varying positions. This reduces the chances of degeneracy occurring, and also provides a very over-determined solution.

Example 8.27. Calibration from three orthogonal vanishing points

Suppose that it is known that the camera has zero skew, and that the pixels are square (or equivalently their aspect ratio is known). A triad of orthogonal vanishing point directions supplies three more constraints. This gives a total of 5 constraints – sufficient to compute $\boldsymbol{\omega}$, and hence $\mathtt{K}$.

In outline the algorithm has the following steps:

(i) In the case of square pixels $\boldsymbol{\omega}$ has the form

$$\boldsymbol{\omega} = \begin{bmatrix} w_1 & 0 & w_2 \\ 0 & w_1 & w_3 \\ w_2 & w_3 & w_4 \end{bmatrix}.$$

(ii) Each pair of vanishing points $\mathbf{v}_i, \mathbf{v}_j$ generates an equation $\mathbf{v}_i^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_j = 0$, which is linear in the elements of $\boldsymbol{\omega}$. The constraints from the three pairs of vanishing points are stacked together to form an equation $\mathtt{A}\mathbf{w} = \mathbf{0}$, where $\mathtt{A}$ is a 3×4 matrix.

(iii) The vector $\mathbf{w}$ is obtained as the null vector of $\mathtt{A}$, and this determines $\boldsymbol{\omega}$. The matrix $\mathtt{K}$ is obtained from $\boldsymbol{\omega} = (\mathtt{K}\mathtt{K}^\mathsf{T})^{-1}$ by Cholesky factorization of $\boldsymbol{\omega}$, followed by inversion.

An example is shown in figure 8.22(a). Vanishing points are computed corresponding to the three perpendicular directions shown in figure 8.22(b). The image is 1024×768 pixels, and the calibration matrix is computed to be

$$\mathtt{K} = \begin{bmatrix} 1163 & 0 & 548 \\ 0 & 1163 & 404 \\ 0 & 0 & 1 \end{bmatrix}.$$

△

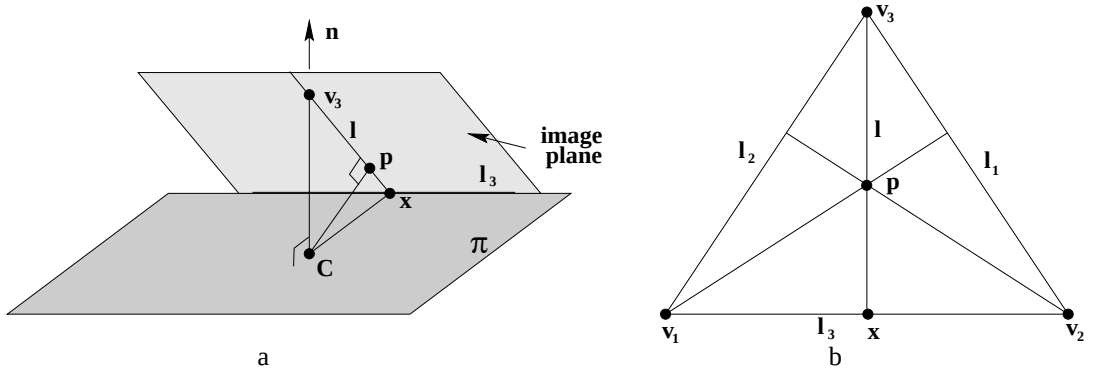


Fig. 8.23. **Geometric construction of the principal point.** *The vanishing line* $\mathbf{l}_3$ *back-projects to a plane* $\boldsymbol{\pi}$ *with normal* $\mathbf{n}$*. The vanishing point* $\mathbf{v}_3$ *back-projects to a line orthogonal to the plane* $\boldsymbol{\pi}$*. (a) The normal* $\mathbf{n}$ *of the plane* $\boldsymbol{\pi}$ *through the camera centre* $\mathbf{C}$ *and the principal axis define a plane, which intersects the image in the line* $\mathbf{l} = \langle \mathbf{v}_3, \mathbf{x} \rangle$*. The line* $\mathbf{l}_3$ *is the intersection of* $\boldsymbol{\pi}$ *with the image plane, and is also its vanishing line. The point* $\mathbf{v}_3$ *is the intersection of the normal with the image plane, and is also its vanishing point. Clearly the principal point lies on* $\mathbf{l}$*, and* $\mathbf{l}$ *and* $\mathbf{l}_3$ *are perpendicular on the image plane. (b) The principal point may be determined from three such constraints as the orthocentre of the triangle.*

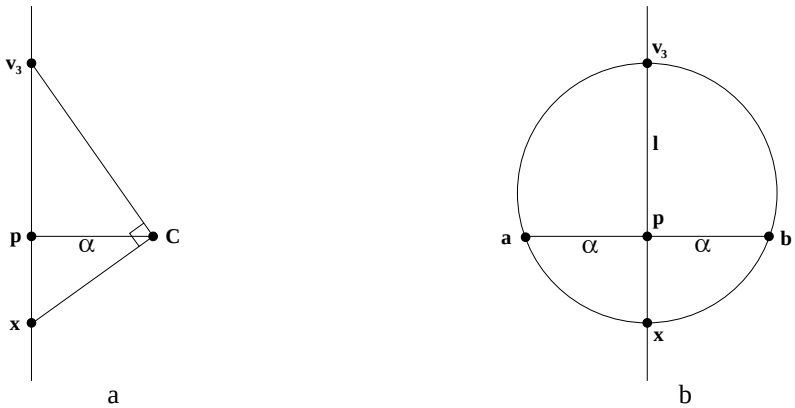


Fig. 8.24. **Geometric construction of the focal length.** *(a) Consider the plane defined by the camera centre* $\mathbf{C}$*, principal point and one of the vanishing points, e.g.* $\mathbf{v}_3$ *as shown in figure 8.23(a). The rays from* $\mathbf{C}$ *to* $\mathbf{v}_3$ *and* $\mathbf{x}$ *are perpendicular to each other. The focal length,* α*, is the distance from the camera centre to the image plane. By similar triangles,* $\alpha^2 = d(\mathbf{p}, \mathbf{v}_3)d(\mathbf{p}, \mathbf{x})$*, where* $d(\mathbf{u}, \mathbf{v})$ *is the distance between the points* $\mathbf{u}$ *and* $\mathbf{v}$*. (b) In the image a circle is drawn with diameter the line between* $\mathbf{v}_3$ *and* $\mathbf{x}$*. A line through* $\mathbf{p}$ *perpendicular to* $\langle \mathbf{v}_3, \mathbf{x} \rangle$ *meets the circle in two points* $\mathbf{a}$ *and* $\mathbf{b}$*. The focal length equals the distance* $d(\mathbf{p}, \mathbf{a})$*.*

The principal point and focal length may also be computed geometrically in this case. The principal point is the orthocentre of the triangle with vertices the vanishing points. Figure 8.23 shows that the principal point lies on the perpendicular line from one triangle side to the opposite vertex. A similar construction for the other two sides shows that the principal point is the orthocentre. An algebraic derivation of this result is left to the exercises. The focal length can also be computed geometrically as shown in figure 8.24.

As a cautionary note, this estimation method is degenerate if one of the vanishing

a b

Fig. 8.25. **Plane rectification via partial internal parameters** *(a) Original image. (b) Rectification assuming the camera has square pixels and principal point at the centre of the image. The focal length is computed from the single orthogonal vanishing point pair. The aspect ratio of a window in the rectified image differs from the ground truth value by 3.7%. Note that the two parallel planes, the upper building facade and the lower shopfront, are both mapped to fronto-parallel planes.*

points, say the vertical, is at infinity. In this case $\mathtt{A}$ drops rank to two, and there is a one-parameter family of solutions for $\boldsymbol{\omega}$ and correspondingly for $\mathtt{K}$. This degeneracy can be seen geometrically from the orthocentre construction of figure 8.23. If $\mathbf{v}_3$ is at infinity then the principal point $\mathbf{p}$ lies on the line $\mathbf{l}_3 = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$, but its x position is not defined.

Example 8.28. Determining the focal length when the other internal parameters are known

We consider a further example of calibration from a single view. Suppose that it is known that the camera has zero skew, that the pixels are square (or equivalently their aspect ratio is known), and also that the principal point is at the image centre. Then only the focal length is unknown. In this case, the form of $\boldsymbol{\omega}$ is very simple: it is a diagonal matrix $\mathrm{diag}(1/f^2, 1/f^2, 1)$ with only one degree of freedom. Using algorithm 8.2, the focal length f may be determined from one further constraint, such as the one arising from two vanishing points corresponding to orthogonal directions.

An example is shown in figure 8.25(a). Here the vanishing points used in the constraint are computed from the horizontal edges of the windows and pavement, and the vertical edges of the windows. These vanishing points also determine the vanishing line $\mathbf{l}$ of the building facade. Given $\mathtt{K}$ and the vanishing line $\mathbf{l}$, the camera can be synthetically rotated such that the facade is fronto-parallel by mapping the image with a homography as in example 8.13(*p*205). The result is shown in figure 8.25(b). Note, in example 8.13 it was necessary to know the aspect ratio of a rectangle on the scene plane in order to rectify the plane. Here it is only necessary to know the vanishing line of the plane because the camera calibration $\mathtt{K}$ provides the additional information required for the homography. △

8.8.1 The geometry of the constraints

Although the algebraic constraints given in table 8.1 appear to arise from distinct sources, they are in fact all equivalent to one of two simple geometric relations: two points lying on the conic $\boldsymbol{\omega}$, or conjugacy of two points with respect to $\boldsymbol{\omega}$.

For example, the zero skew constraint is an orthogonality constraint: it specifies that

the image x and y axes are orthogonal. These axes correspond to rays with directions in the camera's Euclidean coordinate frame, $(1,0,0)^\mathsf{T}$ and $(0,1,0)^\mathsf{T}$, respectively, that are imaged at $\mathbf{v}_x = (1,0,0)^\mathsf{T}$ and $\mathbf{v}_y = (0,1,0)^\mathsf{T}$ (since the rays are parallel to the image plane). The zero skew constraint $\boldsymbol{\omega}_{12} = \boldsymbol{\omega}_{21} = 0$ is just another way of writing the orthogonality constraint (8.16) $\mathbf{v}_y^\mathsf{T}\boldsymbol{\omega}\mathbf{v}_x = 0$. Geometrically skew zero is equivalent to conjugacy of the points $(1,0,0)^\mathsf{T}$ and $(0,1,0)^\mathsf{T}$ with respect to $\boldsymbol{\omega}$.

The square pixel constraint may be interpreted in two ways. A square has the property of defining two sets of orthogonal lines: adjacent edges are orthogonal, and so are the two diagonals. Thus, the square pixel constraint may be interpreted as a pair of orthogonal line constraints. The diagonal vanishing points of a square pixel are $(1,1,0)^\mathsf{T}$ and $(-1,1,0)^\mathsf{T}$. The resulting orthogonality constraints lead to the square pixel constraints given in table 8.1.

Alternatively, the square pixel constraint can be interpreted in terms of two known points lying on the IAC. If the image plane has square pixels, then it has a Euclidean coordinate system and the circular points have known coordinates $(1,\pm i,0)^\mathsf{T}$. It may be verified that the two square pixel equations are equivalent to $(1,\pm i,0)\boldsymbol{\omega}(1,\pm i,0)^\mathsf{T} = 0$.

This is the most important geometric equivalence. In essence an image plane with square pixels acts as a metric plane in the scene. A square pixel image plane is equivalent to a metric plane imaged with a homography given by the identity. Indeed if the homography H in the "metric plane imaged with known homography" constraint of table 8.1 is replaced by the identity then the square pixel constraints are immediately obtained.

Thus, we see that all the constraints given in table 8.1 are derived either from known points lying on $\boldsymbol{\omega}$, or from pairs of points that are conjugate with respect to $\boldsymbol{\omega}$. Determining $\boldsymbol{\omega}$ may therefore be viewed as a conic fitting problem, given points on the conic and conjugate point pairs.

It is well to bear in mind that conic fitting is a delicate problem, often unstable ([Bookstein-79]) if the points are not well distributed on the conic. The same observation is true of the present problem, which we have seen is equivalent to conic fitting. The method given in algorithm 8.2 for finding the calibration from vanishing points amounts to minimization of algebraic error, and therefore does not give an optimal solution. For greater accuracy, the methods of chapter 4, for instance the Sampson error method of section 4.2.6(*p*98) should be used.

8.9 Single view reconstruction

As an application of the methods developed in this chapter we demonstrate now the 3D reconstruction of a texture mapped piecewise planar graphical model from a single image. The camera calibration methods of section 8.8 and the rectification method of example 8.28 may be combined to back project image regions to texture the planes of the model.

The method will be illustrated for the image of figure 8.26(a), where the scene contains three dominant and mutually orthogonal planes: the building facades on the left and right and the ground plane. The parallel line sets in three orthogonal directions de-

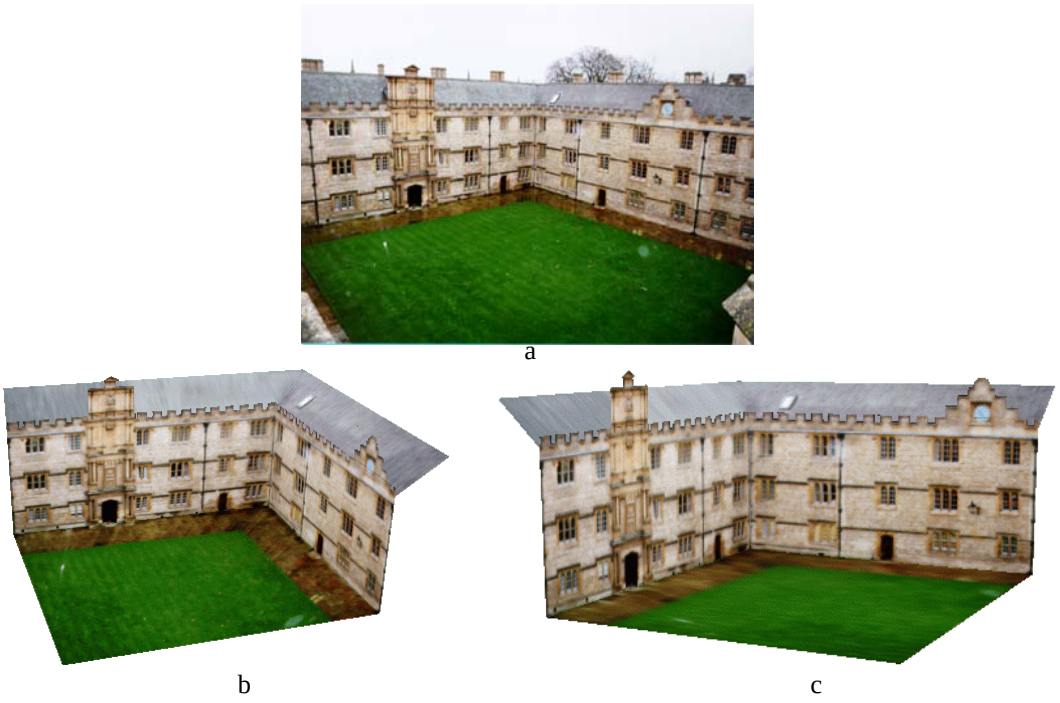


Fig. 8.26. **Single view reconstruction.** *(a) Original image of the Fellows quad, Merton College, Oxford. (b) (c) Views of the 3D model created from the single image. The vanishing line of the roof planes is computed from the repetition of the texture pattern.*

fine three vanishing points and together with the constraint of square pixels the camera calibration may be computed using the method described in section 8.8. From the vanishing lines of the three planes, likewise determined by the vanishing points, together with the computed ω, homographies may be computed to texture map the appropriate image regions onto the orthogonal planes of the model.

In more detail taking the left facade as a reference plane in figure 8.26(a), its correctly proportioned width and height are determined by the rectification. The right facade and ground planes define 3D planes orthogonal to the reference (we have assumed the orthogonality of the planes in computing the camera, so relative orientations are defined). Scaling of the right and ground planes is computed from the points common to the planes and this completes a three orthogonal plane model.

Having computed the calibration, the relative orientation of planes in the scene that are not orthogonal (such as the roof) can be computed if their vanishing lines can be found using (8.14–*p*218). Their relative positions and dimensions can be determined if the intersection of a pair of planes is visible in the image, so that there are points common to both planes. Relative size can be computed from the rectification of a distance between common points using the homographies of both planes. Views of the model, with texture mapped correctly to the planes, appear in figure 8.26(b) and (c).

8.10 The calibrating conic

The image of the absolute conic (IAC) is an imaginary conic in an image, and hence is not visible. Sometimes it is useful for visualization purposes to consider a different conic that is closely related to the calibration of the camera. Such a conic is the *calibrating conic*, which is the image of a cone with apex angle 45° and axis coinciding with the principal axis of the camera.

We wish to compute a formula for this cone in terms of the calibration matrix of the camera. Since the 45° cone moves with the camera, its image is clearly independent of the orientation and position of the camera. Thus, we may assume that the camera is located at the origin and oriented directly along the Z-axis. Thus, let the camera matrix be $\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$. Now, any point on the 45° cone satisfies $\mathtt{X}^2 + \mathtt{Y}^2 = \mathtt{Z}^2$. Points on this cone map to points on the conic

$$\mathtt{C} = \mathtt{K}^{-\mathsf{T}} \begin{bmatrix} 1 & & \\ & 1 & \\ & & -1 \end{bmatrix} \mathtt{K}^{-1} \tag{8.18}$$

as one easily verifies from result 8.6(*p*199). This conic will be referred to as the calibrating conic of the camera. For a calibrated camera with identity calibration matrix $\mathtt{K} = \mathtt{I}$, the calibrating conic is a unit circle centred at the origin (which is the principal point of the image). The conic of (8.18) is simply this unit circle transformed by an affine transformation according to the conic transformation rule of result 2.13(*p*37): $(\mathtt{C} \mapsto \mathtt{H}^{-\mathsf{T}}\mathtt{C}\mathtt{H}^{-1})$. Thus the calibrating conic of a camera with calibration matrix K is the affine transformation of a unit circle centred on the origin by the matrix K.

The calibration parameters are easily read from the calibrating conic. The principal point is the centre of the conic, and the scale factors and skew are easily identified, as in figure 8.27. In the case of zero skew, the calibrating conic has its principal axes aligned with the image coordinate axes. An example on a real image is shown in figure 8.29.

Example 8.29. Suppose $\mathtt{K} = \mathrm{diag}(f, f, 1)$, which is the calibration matrix for a camera of focal length f pixels, with no skew, square pixels, and image origin coincident with the principal point. Then from (8.18) the calibrating conic is $\mathtt{C} = \mathrm{diag}(1, 1, -f^2)$, which is a circle of radius f centred on the principal point. △

Orthogonality and the calibrating conic

A formula was given in (8.9–*p*210) for the angle between the rays corresponding to two image points. In particular the rays corresponding to two points $\mathbf{x}$ and $\mathbf{x}'$ are perpendicular when $\mathbf{x}'^{\mathsf{T}}\boldsymbol{\omega}\mathbf{x} = 0$. As shown in figure 8.13(*p*212) this may be interpreted as the point $\mathbf{x}'$ lying on the line $\boldsymbol{\omega}\mathbf{x}$, which is the polar of $\mathbf{x}$ with respect to the IAC.

We wish to carry out a similar analysis in terms of the calibrating conic. Writing $\mathtt{C} = \mathtt{K}^{-\mathsf{T}}\mathtt{D}\mathtt{K}^{-1}$, where $\mathtt{D} = \mathrm{diag}(1, 1, -1)$, we find

$$\mathtt{C} = (\mathtt{K}^{-\mathsf{T}}\mathtt{K}^{-1})(\mathtt{K}\mathtt{D}\mathtt{K}^{-1}) = \boldsymbol{\omega}\mathtt{S}$$

where $\mathtt{S} = \mathtt{K}\mathtt{D}\mathtt{K}^{-1}$. However, for any point $\mathbf{x}$, the product $\mathtt{S}\mathbf{x}$ represents the reflection of

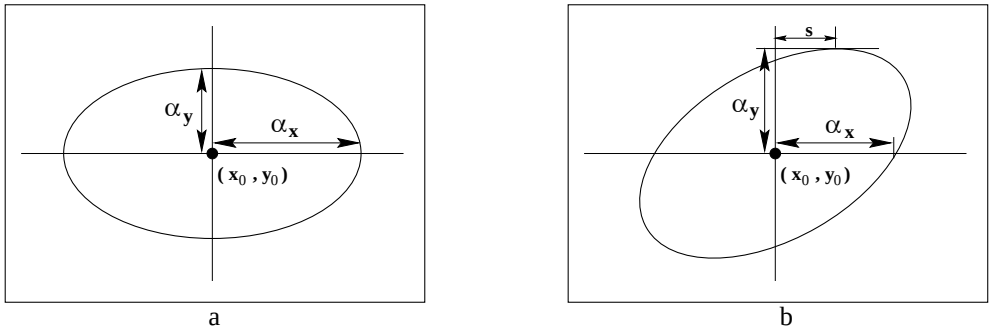


Fig. 8.27. **Reading the internal camera parameters** $\mathtt{K}$ **from the calibrating conic.** *(a) Skew s is zero. (b) Skew s is non-zero. The skew parameter of* $\mathtt{K}$ *(see (6.10–*p*157), is given by the x-coordinate of the highest point of the conic.*

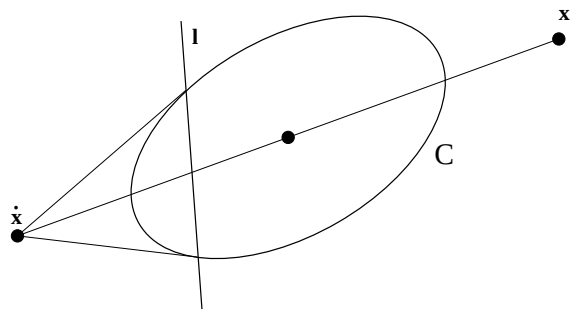


Fig. 8.28. *To construct the line perpendicular to the ray through image point* $\mathbf{x}$ *proceed as follows: (i) Reflect* $\mathbf{x}$ *through the centre of* $\mathtt{C}$ *to get point* $\dot{\mathbf{x}}$ *(i.e. at the same distance from the centre as* $\mathbf{x}$*). (ii) The desired line is the polar of* $\dot{\mathbf{x}}$*.*

the point $\mathbf{x}$ through the centre of the conic $\mathtt{C}$, that is, the principal point of the camera. Representing this reflected point by $\dot{\mathbf{x}}$, one finds that

$$\mathbf{x}'^{\mathsf{T}}\boldsymbol{\omega}\mathbf{x} = \mathbf{x}'^{\mathsf{T}}\mathtt{C}\dot{\mathbf{x}} \ . \tag{8.19}$$

This leads to the following geometric result:

Result 8.30. *The line in an image corresponding to the plane perpendicular to a ray through image point* $\mathbf{x}$ *is the polar* $\mathtt{C}\dot{\mathbf{x}}$ *of the reflected point* $\dot{\mathbf{x}}$ *with respect to the calibrating conic.*

This construction is illustrated in figure 8.28.

Example 8.31. The calibrating conic given three orthogonal vanishing points
The calibrating conic can be drawn directly for the example of figure 8.22. Again assume there is no skew and square pixels, then the calibrating conic is a circle. Now given three mutually perpendicular vanishing points, one can find the calibrating conic by direct geometric construction as shown in figure 8.29.

(i) First, construct the triangle with vertices the three vanishing points $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$.
(ii) The centre of $\mathtt{C}$ is the orthocentre of the triangle.

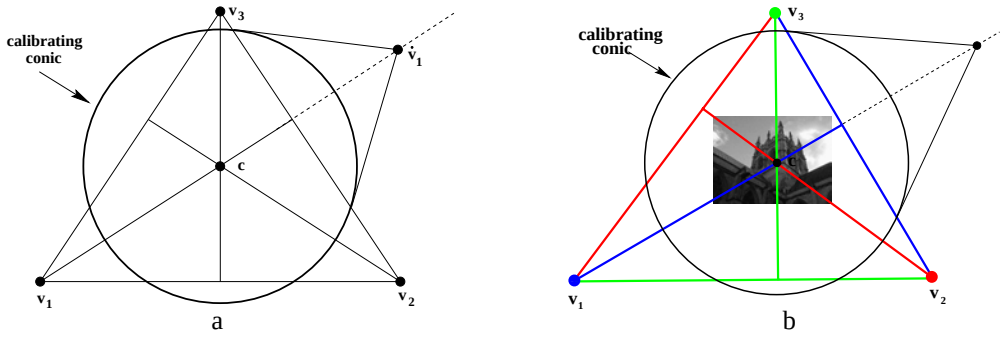


Fig. 8.29. *The calibrating conic computed from three orthogonal vanishing points. (a) The geometric construction. (b) The calibrating conic for the image of figure 8.22.*

(iii) Reflect one of the vanishing points (say $\mathbf{v}_1$) in the centre to get $\dot{\mathbf{v}}_1$.

(iv) The radius of $\mathtt{C}$ is determined by the condition that the polar of $\dot{\mathbf{v}}_1$ is the line passing though $\mathbf{v}_2$ and $\mathbf{v}_3$.

△

8.11 Closure

8.11.1 The literature

Faugeras and Mourrain [Faugeras-95a], and Faugeras and Papadopoulo [Faugeras-97] develop the projection of lines using Plücker coordinates. Koenderink [Koenderink-84, Koenderink-90], and Giblin and Weiss [Giblin-87] give many properties of the contour generator and apparent contour, and their relation to the differential geometry of surfaces.

[Kanatani-92] gives an alternative, calibrated, treatment of vanishing points and lines, and of the result that images acquired by cameras with the same centre are related by a planar homography. Mundy and Zisserman [Mundy-92] showed this result geometrically, and [Hartley-94a] gave a simple algebraic derivation based on camera projection matrices. [Faugeras-92b] introduced the projective (reduced) camera matrix. The link between the image of the absolute conic and camera calibration was first given in [Faugeras-92a].

The computation of panoramic mosaics is described in [Capel-98, Sawhney-98, Szeliski-97]. The ML method of computing vanishing points is given in Liebowitz & Zisserman [Liebowitz-98]. Applications of automatic vanishing line estimation from coplanar equally spaced lines are given in [Schaffalitzky-00b] and also [Se-00]. Affine 3D measurements from a single view is described in [Criminisi-00, Proesmans-98].

The result that $\mathtt{K}$ may be computed from multiple scene planes on which metric structure (such as a square) is known was given in [Liebowitz-98]. Algorithms for this computation are given in [Liebowitz-99a, Sturm-99c, Zhang-00]. The advantage in using $\boldsymbol{\omega}$, rather than $\boldsymbol{\omega}^*$, when imposing the skew zero constraint was first noted in [Armstrong-96b]. The method of internal calibration using three vanishing points for orthogonal directions was given by Caprile and Torre [Caprile-90], though there

is an earlier reference to this result in the photogrammetry literature [Gracie-68]. A simple formula for the focal length in this case is given in [Cipolla-99, Hartley-02b]. A discussion of the degeneracies that arise when combining multiple constraints is given in [Liebowitz-99b, Liebowitz-01]. Single view reconstruction is investigated in [Criminisi-99a, Criminisi-01, Horry-97, Liebowitz-99a, Sturm-99a].

8.11.2 Notes and exercises

(i) **Homography from a world plane**. Suppose $\mathtt{H}$ is computed (e.g. from the correspondence between four or more known world points and their images) and $\mathtt{K}$ known, then the pose of the camera $\{\mathtt{R}, \mathbf{t}\}$ may be computed from the camera matrix $[\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_1 \times \mathbf{r}_2, \mathbf{t}]$, where

$$[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}] = \pm \mathtt{K}^{-1}\mathtt{H}/\|\mathtt{K}^{-1}\mathtt{H}\|.$$

Note that there is a two-fold ambiguity. This result follows from (8.1–*p*196) which gives the homography between a world plane and calibrated camera $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid \mathbf{t}]$.

Show that the homography $\mathbf{x} = \mathtt{H}\widetilde{\mathbf{X}}$ between points on a world plane $(\mathbf{n}^\mathsf{T}, d)^\mathsf{T}$ and the image may be expressed as $\mathtt{H} = \mathtt{K}(\mathtt{R} - \mathbf{t}\mathbf{n}^\mathsf{T}/d)$. The points on the plane have coordinates $\widetilde{\mathbf{X}} = (\mathtt{X}, \mathtt{Y}, \mathtt{Z})^\mathsf{T}$.

(ii) **Line projection.**

(a) Show that any line containing the camera centre lies in the null-space of the map (8.2–*p*198), i.e. it is projected to the line $\mathbf{l} = \mathbf{0}$.

(b) Show that the line $\mathcal{L} = \mathcal{P}^\mathsf{T}\mathbf{x}$ in $\mathbb{P}^3$ is the ray through the image point $\mathbf{x}$ and the camera centre. Hint: start from result 3.5(*p*72), and show that the camera centre $\mathbf{C}$ lies on $\mathcal{L}$.

(c) What is the geometric interpretation of the columns of $\mathcal{P}$?

(iii) **Contour generator of a quadric.** The contour generator $\mathbf{\Gamma}$ of a quadric consists of the set of points $\mathbf{X}$ on $\mathtt{Q}$ for which the tangent planes contain the camera centre, $\mathbf{C}$. The tangent plane at a point $\mathbf{X}$ on $\mathtt{Q}$ is given by $\boldsymbol{\pi} = \mathtt{Q}\mathbf{X}$, and the condition that $\mathbf{C}$ is on $\boldsymbol{\pi}$ is $\mathbf{C}^\mathsf{T}\boldsymbol{\pi} = \mathbf{C}^\mathsf{T}\mathtt{Q}\mathbf{X} = 0$. Thus points $\mathbf{X}$ on $\mathbf{\Gamma}$ satisfy $\mathbf{C}^\mathsf{T}\mathtt{Q}\mathbf{X} = 0$, and thus lie on the *plane* $\boldsymbol{\pi}_\mathbf{\Gamma} = \mathtt{Q}\mathbf{C}$ since $\boldsymbol{\pi}_\mathbf{\Gamma}^\mathsf{T}\mathbf{X} = \mathbf{C}^\mathsf{T}\mathtt{Q}\mathbf{X} = 0$. This shows that the contour generator of a quadric is a plane curve and furthermore, since $\boldsymbol{\pi}_\mathbf{\Gamma} = \mathtt{Q}\mathbf{C}$, that the plane of $\mathbf{\Gamma}$ is the polar plane of the camera centre with respect to the quadric.

(iv) **Apparent contour of an algebraic surface.** Show that the apparent contour of a homogeneous algebraic surface of degree n is a curve of degree $n(n-1)$. For example, if $n = 2$ then the surface is a quadric and the apparent contour a conic. Hint: write the surface as $F(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{W}) = 0$, then the tangent plane contains the camera centre $\mathbf{C}$ if

$$C_\mathtt{X}\frac{\partial F}{\partial \mathtt{X}} + C_\mathtt{Y}\frac{\partial F}{\partial \mathtt{Y}} + C_\mathtt{Z}\frac{\partial F}{\partial \mathtt{Z}} + C_\mathtt{W}\frac{\partial F}{\partial \mathtt{W}} = 0$$

which is a surface of a degree $n-1$.

(v) **Rotation axis vanishing point for $\mathtt{H} = \mathtt{KRK}^{-1}$.** The homography of a conjugate rotation $\mathtt{H} = \mathtt{KRK}^{-1}$ has an eigenvector $\mathtt{Ka}$, where $\mathbf{a}$ is the direction of the rotation axis, since $\mathtt{HKa} = \mathtt{KRa} = \mathtt{Ka}$. The last equality follows because $\mathtt{Ra} = 1\mathbf{a}$, i.e. $\mathbf{a}$ is the unit eigenvector of $\mathtt{R}$. It follows that (a) $\mathtt{Ka}$ is a fixed point under the homography $\mathtt{H}$; and (b) from result 8.20(*p*215) $\mathbf{v} = \mathtt{Ka}$ is the vanishing point of the rotation axis.

(vi) **Synthetic rotations.** Suppose, as in example 8.12(*p*205), that a homography is estimated between two images related by a pure rotation about the camera centre. Then the estimated homography will be a conjugate rotation, so that $\mathtt{H} = \mathtt{KR}(\theta)\mathtt{K}^{-1}$ (though $\mathtt{K}$ and $\mathtt{R}$ are unknown). However, $\mathtt{H}^2$ applied to the first image generates the image that would have been obtained by rotating the camera about the same axis through twice the angle, since $\mathtt{H}^2 = \mathtt{KR}^2\mathtt{K}^{-1} = \mathtt{KR}(2\theta)\mathtt{K}^{-1}$.

More generally we may write $\mathtt{H}^\lambda$ to represent a rotation through any fractional angle $\lambda\theta$. To make sense of $\mathtt{H}^\lambda$, observe that the eigenvalue decomposition of $\mathtt{H}$ is $\mathtt{H}(\theta) = \mathtt{U}\,\mathrm{diag}(1, e^{i\theta}, e^{-i\theta})\,\mathtt{U}^{-1}$, and both θ and $\mathtt{U}$ may be computed from the estimated $\mathtt{H}$. Then

$$\mathtt{H}^\lambda = \mathtt{U}\,\mathrm{diag}(1, e^{i\lambda\theta}, e^{-i\lambda\theta})\,\mathtt{U}^{-1} = \mathtt{KR}(\lambda\theta)\mathtt{K}^{-1}.$$

which is the conjugate of a rotation through angle $\lambda\theta$. Writing ϕ instead of $\lambda\theta$, we may use this homography to generate synthetic images rotated through any angle ϕ. The images are interpolated between the original images (if $0 < \phi < \theta$), or extrapolated (if $\phi > \theta$).

(vii) Show that the imaged circular points of a perspectively imaged plane may be computed if any of the following are on the plane: (i) a square grid; (ii) two rectangles arranged such that the sides of one rectangle are not parallel to the sides of the other; (iii) two circles of equal radius; (iv) two circles of unequal radius.

(viii) Show that in the case of zero skew, $\boldsymbol{\omega}$ is the conic

$$\left(\frac{x - x_0}{\alpha_x}\right)^2 + \left(\frac{y - y_0}{\alpha_y}\right)^2 + 1 = 0$$

which may be interpreted as an ellipse aligned with the axes, centred on the principal point, and with axes of length $i\alpha_x$ and $i\alpha_y$ in the x and y directions respectively.

(ix) If the camera calibration $\mathtt{K}$ and the vanishing line $\mathbf{l}$ of a scene plane are known then the scene plane can be metric rectified by a homography corresponding to a synthetic rotation $\mathtt{H} = \mathtt{KRK}^{-1}$ that maps $\mathbf{l}$ to $\mathbf{l}_\infty$, i.e. it is required that $\mathtt{H}^{-\mathsf{T}}\mathbf{l} = (0, 0, 1)^\mathsf{T}$. This condition arises because if the plane is rotated such that its vanishing line is $\mathbf{l}_\infty$ then it is fronto-parallel. Show that $\mathtt{H}^{-\mathsf{T}}\mathbf{l} = (0, 0, 1)^\mathsf{T}$ is equivalent to $\mathtt{R}\mathbf{n} = (0, 0, 1)^\mathsf{T}$, where $\mathbf{n} = \mathtt{K}^\mathsf{T}\mathbf{l}$ is the normal to the scene plane. This is the condition that the scene normal is rotated to lie along the camera $\mathtt{Z}$ axis. Note the rotation is not uniquely defined since a rotation about the plane's

normal does not affect its metric rectification. However, the last row of $\mathtt{R}$ equals $\mathbf{n}$, so that $\mathtt{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{n}]^\mathsf{T}$ where $\mathbf{n}, \mathbf{r}_1$ and $\mathbf{r}_2$ are a triad of orthonormal vectors.

(x) Show that the angle between two planes with vanishing lines $\mathbf{l}_1$ and $\mathbf{l}_2$ is

$$\cos\theta = \frac{\mathbf{l}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{l}_2}{\sqrt{\mathbf{l}_1^\mathsf{T}\boldsymbol{\omega}^*\mathbf{l}_1}\sqrt{\mathbf{l}_2^\mathsf{T}\boldsymbol{\omega}^*\mathbf{l}_2}}.$$

(xi) Derive (8.15–*p*218). Hint, the line $\mathbf{l}$ lies in the pencil defined by $\mathbf{l}_1$ and $\mathbf{l}_2$, so it can be expressed as $\mathbf{l} = \alpha\mathbf{l}_1 + \beta\mathbf{l}_2$. Then use the relations $\mathbf{l}_n = \mathbf{l}_0 + n\mathbf{l}$ for $n = 1, 2$ to solve for α and β.

(xii) For the case of vanishing points arising from three orthogonal directions, and for an image with square pixels, show algebraically that the principal point is the orthocentre of the triangle with vertices the vanishing points. Hint: suppose the vanishing point at one vertex of the triangle is $\mathbf{v}$ and the line of the opposite side (through the other two vanishing points) is $\mathbf{l}$. Then from (8.17–*p*219) $\mathbf{v} = \boldsymbol{\omega}^*\mathbf{l}$ since $\mathbf{v}$ and $\mathbf{l}$ arise from an orthogonal line and plane respectively. Show that the principal point lies on the line from $\mathbf{v}$ to $\mathbf{l}$ which is perpendicular *in the image* to $\mathbf{l}$. Since this result is true for any vertex the principal point is the orthocentre of the triangle.

(xiii) Show that the vanishing points of an orthogonal triad of directions are the vertices of a self-polar triangle [Springer-64] with respect to $\boldsymbol{\omega}$.

(xiv) If a camera has square pixels, then the apparent contour of a sphere centred on the principal axis is a circle. If the sphere is translated parallel to the image plane, then the apparent contour deforms from a circle to an ellipse with the principal point on its major axis.

(a) How can this observation be used as a method of internal parameter calibration?

(b) Show by a geometric argument that the aspect ratio of the ellipse does not depend on the distance of the sphere from the camera.

If the sphere is now translated parallel to the principal axis the apparent contour can deform to a hyperbola, but only one branch of the hyperbola is imaged. Why is this?

(xv) Show that for a general camera the apparent contour of a sphere is related to the IAC as:

$$\boldsymbol{\omega} = \mathtt{C} + \mathbf{v}\mathbf{v}^\mathsf{T}$$

where $\mathtt{C}$ is the conic outline of the imaged sphere, and $\mathbf{v}$ is a 3-vector that depends on the position of the sphere. A proof is given in [Agrawal-03]. Note this relation places two constraints on $\boldsymbol{\omega}$, so that in principle $\boldsymbol{\omega}$, and hence the calibration $\mathtt{K}$, may be computed from three images of a sphere. However, in practice this is not a well conditioned method for computing $\mathtt{K}$ because the deviation of the sphere's outline from a circle is small.