Part 0

The Background: Projective Geometry, Transformations and Estimation

La reproduction interdite (The Forbidden Reproduction), 1937, René Magritte.
Courtesy of Museum Boijmans van Beuningen, Rotterdam.
© ADAGP, Paris, and DACS, London 2000.

Outline

The four chapters in this part lay the foundation for the representations, terminology, and notation that will be used in the subsequent parts of the book. The ideas and notation of projective geometry are central to an analysis of multiple view geometry. For example, the use of homogeneous coordinates enables non-linear mappings (such as perspective projection) to be represented by linear matrix equations, and points at infinity to be represented quite naturally avoiding the awkward necessity of taking limits.

Chapter 2 introduces projective transformations of 2-space. These are the transformations that arise when a plane is imaged by a perspective camera. This chapter is more introductory and sets the scene for the geometry of 3-space. Most of the concepts can be more easily understood and visualized in 2D than in 3D. Specializations of projective transformations are introduced, including affine and similarity transformations. Particular attention is focussed on the recovery of affine properties (e.g. parallel lines) and metric properties (e.g. angles between lines) from a perspective image.

Chapter 3 covers the projective geometry of 3-space. This geometry develops in much the same manner as that of 2-space, though of course there are extra properties arising from the additional dimension. The main new geometry here is the plane at infinity and the absolute conic.

Chapter 4 introduces *estimation* of geometry from image measurements, which is one of the main topics of this book. The example of estimating a projective transformation from point correspondences is used to illustrate the basis and motivation for the algorithms that will be used throughout the book. The important issue of what should be minimized in a cost function, e.g. algebraic or geometric or statistical measures, is described at length. The chapter also introduces the idea of robust estimation, and the use of such techniques in the automatic estimation of transformations.

Chapter 5 describes how the results of estimation algorithms may be evaluated. In particular how the covariance of an estimation may be computed.

ıe representations, terminology,
ts of the book. The ideas and
ysis of multiple view geometry.
ıbles non-linear mappings (such
matrix equations, and points at
ıe awkward necessity of taking

2-space. These are the transfor-
pective camera. This chapter is
of 3-space. Most of the concepts
ıan in 3D. Specializations of pro-
.e and similarity transformations.
ıne properties (e.g. parallel lines)
ı a perspective image.

ace. This geometry develops in
course there are extra properties
ew geometry here is the plane at

n image measurements, which is
f estimating a projective transfor-
ıte the basis and motivation for the
he important issue of what should
ometric or statistical measures, is
idea of robust estimation, and the
f transformations.

ı algorithms may be evaluated. In
be computed.

2

Projective Geometry and Transformations of 2D

This chapter introduces the main geometric ideas and notation that are required to understand the material covered in this book. Some of these ideas are relatively familiar, such as vanishing point formation or representing conics, whilst others are more esoteric, such as using circular points to remove perspective distortion from an image. These ideas can be understood more easily in the planar (2D) case because they are more easily visualized here. The geometry of 3-space, which is the subject of the later parts of this book, is only a simple generalization of this planar case.

In particular, the chapter covers the geometry of projective transformations of the plane. These transformations model the geometric distortion which arises when a plane is imaged by a perspective camera. Under perspective imaging certain geometric properties are preserved, such as collinearity (a straight line is imaged as a straight line), whilst others are not, for example parallel lines are not imaged as parallel lines in general. Projective geometry models this imaging and also provides a mathematical representation appropriate for computations.

We begin by describing the representation of points, lines and conics in homogeneous notation, and how these entities map under projective transformations. The line at infinity and the circular points are introduced, and it is shown that these capture the affine and metric properties of the plane. Algorithms for rectifying planes are then given which enable affine and metric properties to be computed from images. We end with a description of fixed points under projective transformations.

2.1 Planar geometry

The basic concepts of planar geometry are familiar to anyone who has studied mathematics even at an elementary level. In fact, they are so much a part of our everyday experience that we take them for granted. At an elementary level, geometry is the study of points and lines and their relationships.

To the purist, the study of geometry ought properly to be carried out from a "geometric" or coordinate-free viewpoint. In this approach, theorems are stated and proved in terms of geometric primitives only, without the use of algebra. The classical approach of Euclid is an example of this method. Since Descartes, however, it has been seen that geometry may be algebraicized, and indeed the theory of geometry may be developed

from an algebraic viewpoint. Our approach in this book will be a hybrid approach, sometimes using geometric, and sometimes algebraic methods. In the algebraic approach, geometric entities are described in terms of coordinates and algebraic entities. Thus, for instance a point is identified with a vector in terms of some coordinate basis. A line is also identified with a vector, and a conic section (more briefly, a conic) is represented by a symmetric matrix. In fact, we often carry this identification so far as to consider that the vector actually *is* a point, or the symmetric matrix *is* a conic, at least for convenience of language. A significant advantage of the algebraic approach to geometry is that results derived in this way may more easily be used to derive algorithms and practical computational methods. Computation and algorithms are a major concern in this book, which justifies the use of the algebraic method.

2.2 The 2D projective plane

As we all know, a point in the plane may be represented by the pair of coordinates (x, y) in $\mathbb{R}^2$. Thus, it is common to identify the plane with $\mathbb{R}^2$. Considering $\mathbb{R}^2$ as a vector space, the coordinate pair (x, y) is a vector – a point is identified as a vector. In this section we introduce the *homogeneous* notation for points and lines on a plane.

Row and column vectors. Later on, we will want to consider linear mappings between vector spaces, and represent such mappings as matrices. In the usual manner, the product of a matrix and a vector is another vector, the image under the mapping. This brings up the distinction between "column" and "row" vectors, since a matrix may be multiplied on the right by a column and on the left by a row vector. Geometric entities will by default be represented by column vectors. A bold-face symbol such as $\mathbf{x}$ always represents a column vector, and its transpose is the row vector $\mathbf{x}^\mathsf{T}$. In accordance with this convention, a point in the plane will be represented by the column vector $(x, y)^\mathsf{T}$, rather than its transpose, the row vector (x, y). We write $\mathbf{x} = (x, y)^\mathsf{T}$, both sides of this equation representing column vectors.

2.2.1 Points and lines

Homogeneous representation of lines. A line in the plane is represented by an equation such as $ax+by+c = 0$, different choices of a, b and c giving rise to different lines. Thus, a line may naturally be represented by the vector $(a, b, c)^\mathsf{T}$. The correspondence between lines and vectors $(a, b, c)^\mathsf{T}$ is not one-to-one, since the lines $ax + by + c = 0$ and $(ka)x + (kb)y + (kc) = 0$ are the same, for any non-zero constant k. Thus, the vectors $(a, b, c)^\mathsf{T}$ and $k(a, b, c)^\mathsf{T}$ represent the same line, for any non-zero k. In fact, two such vectors related by an overall scaling are considered as being equivalent. An equivalence class of vectors under this equivalence relationship is known as a *homogeneous* vector. Any particular vector $(a, b, c)^\mathsf{T}$ is a representative of the equivalence class. The set of equivalence classes of vectors in $\mathbb{R}^3 - (0, 0, 0)^\mathsf{T}$ forms the *projective space* $\mathbb{P}^2$. The notation $-(0, 0, 0)^\mathsf{T}$ indicates that the vector $(0, 0, 0)^\mathsf{T}$, which does not correspond to any line, is excluded.

book will be a hybrid approach,
ic methods. In the algebraic ap-
coordinates and algebraic entities.
in terms of some coordinate basis.
section (more briefly, a conic) is
n carry this identification so far as
e symmetric matrix *is* a conic, at
vantage of the algebraic approach
nore easily be used to derive algo-
utation and algorithms are a major
algebraic method.

e plane

resented by the pair of coordinates
lane with $\mathbb{R}^2$. Considering $\mathbb{R}^2$ as a
– a point is identified as a vector. In
n for points and lines on a plane.

ant to consider linear mappings be-
as matrices. In the usual manner, the
, the image under the mapping. This
"row" vectors, since a matrix may be
ft by a row vector. Geometric entities
A bold-face symbol such as $\mathbf{x}$ always
e row vector $\mathbf{x}^\mathsf{T}$. In accordance with
esented by the column vector $(x, y)^\mathsf{T}$,
e write $\mathbf{x} = (x, y)^\mathsf{T}$, both sides of this

in the plane is represented by an equa-
a, b and c giving rise to different lines.
e vector $(a, b, c)^\mathsf{T}$. The correspondence
o-one, since the lines $ax + by + c = 0$
for any non-zero constant k. Thus, the
same line, for any non-zero k. In fact,
are considered as being equivalent. An
lence relationship is known as a *homo-*
$^\mathsf{T}$ is a representative of the equivalence
s in $\mathbb{R}^3 - (0, 0, 0)^\mathsf{T}$ forms the *projective*
hat the vector $(0, 0, 0)^\mathsf{T}$, which does not

Homogeneous representation of points. A point $\mathbf{x} = (x, y)^\mathsf{T}$ lies on the line $\mathbf{l} = (a, b, c)^\mathsf{T}$ if and only if $ax + by + c = 0$. This may be written in terms of an inner product of vectors representing the point as $(x, y, 1)(a, b, c)^\mathsf{T} = (x, y, 1)\mathbf{l} = 0$; that is the point $(x, y)^\mathsf{T}$ in $\mathbb{R}^2$ is represented as a 3-vector by adding a final coordinate of 1. Note that for any non-zero constant k and line $\mathbf{l}$ the equation $(kx, ky, k)\mathbf{l} = 0$ if and only if $(x, y, 1)\mathbf{l} = 0$. It is natural, therefore, to consider the set of vectors $(kx, ky, k)^\mathsf{T}$ for varying values of k to be a representation of the point $(x, y)^\mathsf{T}$ in $\mathbb{R}^2$. Thus, just as with lines, points are represented by homogeneous vectors. An arbitrary homogeneous vector representative of a point is of the form $\mathbf{x} = (x_1, x_2, x_3)^\mathsf{T}$, representing the point $(x_1/x_3, x_2/x_3)^\mathsf{T}$ in $\mathbb{R}^2$. Points, then, as homogeneous vectors are also elements of $\mathbb{P}^2$.

One has a simple equation to determine when a point lies on a line, namely

Result 2.1. *The point* $\mathbf{x}$ *lies on the line* $\mathbf{l}$ *if and only if* $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$.

Note that the expression $\mathbf{x}^\mathsf{T}\mathbf{l}$ is just the inner or scalar product of the two vectors $\mathbf{l}$ and $\mathbf{x}$. The scalar product $\mathbf{x}^\mathsf{T}\mathbf{l} = \mathbf{l}^\mathsf{T}\mathbf{x} = \mathbf{x}.\mathbf{l}$. In general, the transpose notation $\mathbf{l}^\mathsf{T}\mathbf{x}$ will be preferred, but occasionally, we will use a . to denote the inner product. We distinguish between the *homogeneous coordinates* $\mathbf{x} = (x_1, x_2, x_3)^\mathsf{T}$ of a point, which is a 3-vector, and the *inhomogeneous* coordinates $(x, y)^\mathsf{T}$, which is a 2-vector.

Degrees of freedom (dof). It is clear that in order to specify a point two values must be provided, namely its x- and y-coordinates. In a similar manner a line is specified by two parameters (the two independent ratios $\{a : b : c\}$) and so has two degrees of freedom. For example, in an inhomogeneous representation, these two parameters could be chosen as the gradient and y intercept of the line.

Intersection of lines. Given two lines $\mathbf{l} = (a, b, c)^\mathsf{T}$ and $\mathbf{l}' = (a', b', c')^\mathsf{T}$, we wish to find their intersection. Define the vector $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$, where $\times$ represents the vector or cross product. From the triple scalar product identity $\mathbf{l}.(\mathbf{l} \times \mathbf{l}') = \mathbf{l}'.(\mathbf{l} \times \mathbf{l}') = 0$, we see that $\mathbf{l}^\mathsf{T}\mathbf{x} = \mathbf{l}'^\mathsf{T}\mathbf{x} = 0$. Thus, if $\mathbf{x}$ is thought of as representing a point, then $\mathbf{x}$ lies on both lines $\mathbf{l}$ and $\mathbf{l}'$, and hence is the intersection of the two lines. This shows:

Result 2.2. *The intersection of two lines* $\mathbf{l}$ *and* $\mathbf{l}'$ *is the point* $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$.

Note that the simplicity of this expression for the intersection of the two lines is a direct consequence of the use of homogeneous vector representations of lines and points.

Example 2.3. Consider the simple problem of determining the intersection of the lines $x = 1$ and $y = 1$. The line $x = 1$ is equivalent to $-1x + 1 = 0$, and thus has homogeneous representation $\mathbf{l} = (-1, 0, 1)^\mathsf{T}$. The line $y = 1$ is equivalent to $-1y + 1 = 0$, and thus has homogeneous representation $\mathbf{l}' = (0, -1, 1)^\mathsf{T}$. From result 2.2 the intersection point is

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{vmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

which is the inhomogeneous point $(1, 1)^\mathsf{T}$ as required. △

Line joining points. An expression for the line passing through two points $\mathbf{x}$ and $\mathbf{x}'$ may be derived by an entirely analogous argument. Defining a line $\mathbf{l}$ by $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$, it may be verified that both points $\mathbf{x}$ and $\mathbf{x}'$ lie on $\mathbf{l}$. Thus

Result 2.4. *The line through two points* $\mathbf{x}$ *and* $\mathbf{x}'$ *is* $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$.

2.2.2 Ideal points and the line at infinity

Intersection of parallel lines. Consider two lines $ax+by+c = 0$ and $ax+by+c' = 0$. These are represented by vectors $\mathbf{l} = (a, b, c)^\mathsf{T}$ and $\mathbf{l}' = (a, b, c')^\mathsf{T}$ for which the first two coordinates are the same. Computing the intersection of these lines gives no difficulty, using result 2.2. The intersection is $\mathbf{l} \times \mathbf{l}' = (c' - c)(b, -a, 0)^\mathsf{T}$, and ignoring the scale factor $(c' - c)$, this is the point $(b, -a, 0)^\mathsf{T}$.

Now if we attempt to find the inhomogeneous representation of this point, we obtain $(b/0, -a/0)^\mathsf{T}$, which makes no sense, except to suggest that the point of intersection has infinitely large coordinates. In general, points with homogeneous coordinates $(x, y, 0)^\mathsf{T}$ do not correspond to any finite point in $\mathbb{R}^2$. This observation agrees with the usual idea that parallel lines meet at infinity.

Example 2.5. Consider the two lines $x = 1$ and $x = 2$. Here the two lines are parallel, and consequently intersect "at infinity". In homogeneous notation the lines are $\mathbf{l} = (-1, 0, 1)^\mathsf{T}$, $\mathbf{l}' = (-1, 0, 2)^\mathsf{T}$, and from result 2.2 their intersection point is

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -1 & 0 & 1 \\ -1 & 0 & 2 \end{vmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

which is the point at infinity in the direction of the y-axis. △

Ideal points and the line at infinity. Homogeneous vectors $\mathbf{x} = (x_1, x_2, x_3)^\mathsf{T}$ such that $x_3 \neq 0$ correspond to finite points in $\mathbb{R}^2$. One may augment $\mathbb{R}^2$ by adding points with last coordinate $x_3 = 0$. The resulting space is the set of all homogeneous 3-vectors, namely the projective space $\mathbb{P}^2$. The points with last coordinate $x_3 = 0$ are known as *ideal* points, or points at infinity. The set of all ideal points may be written $(x_1, x_2, 0)^\mathsf{T}$, with a particular point specified by the ratio $x_1 : x_2$. Note that this set lies on a single line, the *line at infinity*, denoted by the vector $\mathbf{l}_\infty = (0, 0, 1)^\mathsf{T}$. Indeed, one verifies that $(0, 0, 1)(x_1, x_2, 0)^\mathsf{T} = 0$.

Using result 2.2 one finds that a line $\mathbf{l} = (a, b, c)^\mathsf{T}$ intersects $\mathbf{l}_\infty$ in the ideal point $(b, -a, 0)^\mathsf{T}$ (since $(b, -a, 0)\mathbf{l} = 0$). A line $\mathbf{l}' = (a, b, c')^\mathsf{T}$ *parallel* to $\mathbf{l}$ intersects $\mathbf{l}_\infty$ in the same ideal point $(b, -a, 0)^\mathsf{T}$ irrespective of the value of c'. In inhomogeneous notation $(b, -a)^\mathsf{T}$ is a vector tangent to the line, and orthogonal to the line normal (a, b), and so represents the line's *direction*. As the line's direction varies the ideal point $(b, -a, 0)^\mathsf{T}$ varies over $\mathbf{l}_\infty$. For these reasons the line at infinity can be thought of as the set of directions of lines in the plane.

Note how the introduction of the concept of points at infinity serves to simplify the intersection properties of points and lines. In the projective plane $\mathbb{P}^2$, one may state without qualification that two distinct lines meet in a single point and two distinct

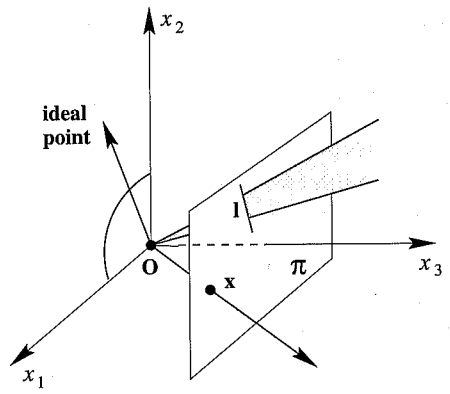


Fig. 2.1. **A model of the projective plane.** *Points and lines of* $\mathbb{P}^2$ *are represented by rays and planes, respectively, through the origin in* $\mathbb{R}^3$. *Lines lying in the* x_1x_2*-plane represent ideal points, and the* x_1x_3*-plane represents* $\mathbf{l}_\infty$.

points lie on a single line. This is not true in the standard Euclidean geometry of $\mathbb{R}^2$, in which parallel lines form a special case.

The study of the geometry of $\mathbb{P}^2$ is known as projective geometry. In a coordinate-free purely geometric study of projective geometry, one does not make any distinction between points at infinity (ideal points) and ordinary points. It will, however, serve our purposes in this book sometimes to distinguish between ideal points and non-ideal points. Thus, the line at infinity will at times be considered as a special line in projective space.

A model for the projective plane. A fruitful way of thinking of $\mathbb{P}^2$ is as a set of rays in $\mathbb{R}^3$. The set of all vectors $k(x_1, x_2, x_3)^\mathsf{T}$ as k varies forms a ray through the origin. Such a ray may be thought of as representing a single point in $\mathbb{P}^2$. In this model, the lines in $\mathbb{P}^2$ are planes passing through the origin. One verifies that two non-identical rays lie on exactly one plane, and any two planes intersect in one ray. This is the analogue of two distinct points uniquely defining a line, and two lines always intersecting in a point.

Points and lines may be obtained by intersecting this set of rays and planes by the plane $x_3 = 1$. As illustrated in figure 2.1 the rays representing ideal points and the plane representing $\mathbf{l}_\infty$ are parallel to the plane $x_3 = 1$.

Duality. The reader has probably noticed how the role of points and lines may be interchanged in statements concerning the properties of lines and points. In particular, the basic incidence equation $\mathbf{l}^\mathsf{T}\mathbf{x} = 0$ for line and point is symmetric, since $\mathbf{l}^\mathsf{T}\mathbf{x} = 0$ implies $\mathbf{x}^\mathsf{T}\mathbf{l} = 0$, in which the positions of line and point are swapped. Similarly, result 2.2 and result 2.4 giving the intersection of two lines and the line through two points are essentially the same, with the roles of points and lines swapped. One may enunciate a general principle, the *duality principle* as follows:

Result 2.6. Duality principle. *To any theorem of 2-dimensional projective geometry there corresponds a dual theorem, which may be derived by interchanging the roles of points and lines in the original theorem.*

In applying this principle, concepts of incidence must be appropriately translated as well. For instance, the line through two points is dual to the point through (that is the point of intersection of) two lines.

Note that is it not necessary to prove the dual of a given theorem once the original theorem has been proved. The proof of the dual theorem will be the dual of the proof of the original theorem.

2.2.3 Conics and dual conics

A conic is a curve described by a second-degree equation in the plane. In Euclidean geometry conics are of three main types: hyperbola, ellipse, and parabola (apart from so-called degenerate conics, to be defined later). Classically these three types of conic arise as conic sections generated by planes of differing orientation (the degenerate conics arise from planes which contain the cone vertex). However, it will be seen that in 2D projective geometry all non-degenerate conics are equivalent under projective transformations.

The equation of a conic in inhomogeneous coordinates is

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

i.e. a polynomial of degree 2. "Homogenizing" this by the replacements: $x \mapsto x_1/x_3,\ y \mapsto x_2/x_3$ gives

$$a{x_1}^2 + bx_1x_2 + c{x_2}^2 + dx_1x_3 + ex_2x_3 + f{x_3}^2 = 0 \tag{2.1}$$

or in matrix form

$$\mathbf{x}^\mathsf{T}\mathtt{C}\mathbf{x} = 0 \tag{2.2}$$

where the conic coefficient matrix $\mathtt{C}$ is given by

$$\mathtt{C} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}. \tag{2.3}$$

Note that the conic coefficient matrix is symmetric. As in the case of the homogeneous representation of points and lines, only the ratios of the matrix elements are important, since multiplying $\mathtt{C}$ by a non-zero scalar does not affect the above equations. Thus $\mathtt{C}$ is a homogeneous representation of a conic. The conic has five degrees of freedom which can be thought of as the ratios $\{a : b : c : d : e : f\}$ or equivalently the six elements of a symmetric matrix less one for scale.

Five points define a conic. Suppose we wish to compute the conic which passes through a set of points, $\mathbf{x}_i$. How many points are we free to specify before the conic is determined uniquely? The question can be answered constructively by providing an

algorithm to determine the conic. From (2.1) each point $\mathbf{x}_i$ places one constraint on the conic coefficients, since if the conic passes through (x_i, y_i) then

$$a{x_i}^2 + bx_iy_i + c{y_i}^2 + dx_i + ey_i + f = 0.$$

This constraint can be written as

$$\begin{pmatrix} x_i^2 & x_iy_i & y_i^2 & x_i & y_i & 1 \end{pmatrix} \mathbf{c} = 0$$

where $\mathbf{c} = (a, b, c, d, e, f)^\mathsf{T}$ is the conic $\mathtt{C}$ represented as a 6-vector.

Stacking the constraints from five points we obtain

$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \end{bmatrix} \mathbf{c} = \mathbf{0} \tag{2.4}$$

and the conic is the null vector of this 5×6 matrix. This shows that a conic is determined uniquely (up to scale) by five points in general position. The method of fitting a geometric entity (or relation) by determining a null space will be used frequently in the computation chapters throughout this book.

Tangent lines to conics. The line $\mathbf{l}$ tangent to a conic at a point $\mathbf{x}$ has a particularly simple form in homogeneous coordinates:

Result 2.7. *The line* $\mathbf{l}$ *tangent to* $\mathtt{C}$ *at a point* $\mathbf{x}$ *on* $\mathtt{C}$ *is given by* $\mathbf{l} = \mathtt{C}\mathbf{x}$.

Proof. The line $\mathbf{l} = \mathtt{C}\mathbf{x}$ passes through $\mathbf{x}$, since $\mathbf{l}^\mathsf{T}\mathbf{x} = \mathbf{x}^\mathsf{T}\mathtt{C}\mathbf{x} = 0$. If $\mathbf{l}$ has one-point contact with the conic, then it is a tangent, and we are done. Otherwise suppose that $\mathbf{l}$ meets the conic in another point $\mathbf{y}$. Then $\mathbf{y}^\mathsf{T}\mathtt{C}\mathbf{y} = 0$ and $\mathbf{x}^\mathsf{T}\mathtt{C}\mathbf{y} = \mathbf{l}^\mathsf{T}\mathbf{y} = 0$. From this it follows that $(\mathbf{x} + \alpha\mathbf{y})^\mathsf{T}\mathtt{C}(\mathbf{x} + \alpha\mathbf{y}) = 0$ for all α, which means that the whole line $\mathbf{l} = \mathtt{C}\mathbf{x}$ joining $\mathbf{x}$ and $\mathbf{y}$ lies on the conic $\mathtt{C}$, which is therefore degenerate (see below). □

Dual conics. The conic $\mathtt{C}$ defined above is more properly termed a *point* conic, as it defines an equation on points. Given the duality result 2.6 of $\mathbb{P}^2$ it is not surprising that there is also a conic which defines an equation on lines. This *dual* (or line) conic is also represented by a 3×3 matrix, which we denote as $\mathtt{C}^*$. A line $\mathbf{l}$ *tangent* to the conic $\mathtt{C}$ satisfies $\mathbf{l}^\mathsf{T}\mathtt{C}^*\mathbf{l} = 0$. The notation $\mathtt{C}^*$ indicates that $\mathtt{C}^*$ is the adjoint matrix of $\mathtt{C}$ (the adjoint is defined in section A4.2(*p*580) of appendix 4(*p*578)). For a non-singular symmetric matrix $\mathtt{C}^* = \mathtt{C}^{-1}$ (up to scale).

The equation for a dual conic is straightforward to derive in the case that $\mathtt{C}$ has full rank. From result 2.7, at a point $\mathbf{x}$ on $\mathtt{C}$ the tangent is $\mathbf{l} = \mathtt{C}\mathbf{x}$. Inverting, we find the point $\mathbf{x}$ at which the line $\mathbf{l}$ is tangent to $\mathtt{C}$ is $\mathbf{x} = \mathtt{C}^{-1}\mathbf{l}$. Since $\mathbf{x}$ satisfies $\mathbf{x}^\mathsf{T}\mathtt{C}\mathbf{x} = 0$ we obtain $(\mathtt{C}^{-1}\mathbf{l})^\mathsf{T}\mathtt{C}(\mathtt{C}^{-1}\mathbf{l}) = \mathbf{l}^\mathsf{T}\mathtt{C}^{-1}\mathbf{l} = 0$, the last step following from $\mathtt{C}^{-\mathsf{T}} = \mathtt{C}^{-1}$ because $\mathtt{C}$ is symmetric.

Dual conics are also known as conic envelopes, and the reason for this is illustrated

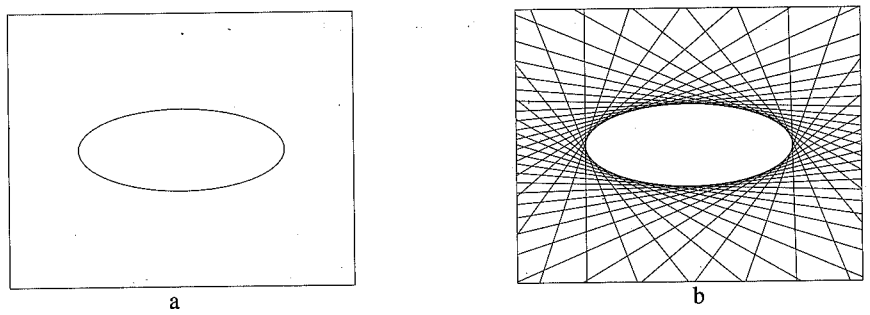


Fig. 2.2. *(a) Points* $\mathbf{x}$ *satisfying* $\mathbf{x}^\mathsf{T}\mathtt{C}\mathbf{x} = 0$ *lie on a point conic. (b) Lines* $\mathbf{l}$ *satisfying* $\mathbf{l}^\mathsf{T}\mathtt{C}^*\mathbf{l} = 0$ *are tangent to the point conic* $\mathtt{C}$*. The conic* $\mathtt{C}$ *is the envelope of the lines* $\mathbf{l}$*.*

in figure 2.2. A dual conic has five degrees of freedom. In a similar manner to points defining a point conic, it follows that five lines in general position define a dual conic.

Degenerate conics. If the matrix $\mathtt{C}$ is not of full rank, then the conic is termed degenerate. Degenerate point conics include two lines (rank 2), and a repeated line (rank 1).

Example 2.8. The conic

$$\mathtt{C} = \mathbf{l}\,\mathbf{m}^\mathsf{T} + \mathbf{m}\,\mathbf{l}^\mathsf{T}$$

is composed of two lines $\mathbf{l}$ and $\mathbf{m}$. Points on $\mathbf{l}$ satisfy $\mathbf{l}^\mathsf{T}\mathbf{x} = 0$, and are on the conic since $\mathbf{x}^\mathsf{T}\mathtt{C}\mathbf{x} = (\mathbf{x}^\mathsf{T}\mathbf{l})(\mathbf{m}^\mathsf{T}\mathbf{x}) + (\mathbf{x}^\mathsf{T}\mathbf{m})(\mathbf{l}^\mathsf{T}\mathbf{x}) = 0$. Similarly, points satisfying $\mathbf{m}^\mathsf{T}\mathbf{x} = 0$ also satisfy $\mathbf{x}^\mathsf{T}\mathtt{C}\mathbf{x} = 0$. The matrix $\mathtt{C}$ is symmetric and has rank 2. The null vector is $\mathbf{x} = \mathbf{l} \times \mathbf{m}$ which is the intersection point of $\mathbf{l}$ and $\mathbf{m}$. △

Degenerate *line* conics include two points (rank 2), and a repeated point (rank 1). For example, the line conic $\mathtt{C}^* = \mathbf{x}\mathbf{y}^\mathsf{T} + \mathbf{y}\mathbf{x}^\mathsf{T}$ has rank 2 and consists of lines passing through either of the two points $\mathbf{x}$ and $\mathbf{y}$. Note that for matrices that are not invertible $(\mathtt{C}^*)^* \neq \mathtt{C}$.

2.3 Projective transformations

In the view of geometry set forth by Felix Klein in his famous "Erlangen Program", [Klein-39], geometry is the study of properties invariant under groups of transformations. From this point of view, 2D projective geometry is the study of properties of the projective plane $\mathbb{P}^2$ that are invariant under a group of transformations known as *projectivities*.

A projectivity is an invertible mapping from points in $\mathbb{P}^2$ (that is homogeneous 3-vectors) to points in $\mathbb{P}^2$ that maps lines to lines. More precisely,

Definition 2.9. A *projectivity* is an invertible mapping h from $\mathbb{P}^2$ to itself such that three points $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$ lie on the same line if and only if $h(\mathbf{x}_1)$, $h(\mathbf{x}_2)$ and $h(\mathbf{x}_3)$ do.

Projectivities form a group since the inverse of a projectivity is also a projectivity, and so is the composition of two projectivities. A projectivity is also called a *collineation*

(a helpful name), a *projective transformation* or a *homography*: the terms are synonymous.

In definition 2.9, a projectivity is defined in terms of a coordinate-free geometric concept of point line incidence. An equivalent algebraic definition of a projectivity is possible, based on the following result.

Theorem 2.10. *A mapping $h : \mathbb{P}^2 \to \mathbb{P}^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix $\mathtt{H}$ such that for any point in $\mathbb{P}^2$ represented by a vector $\mathbf{x}$ it is true that $h(\mathbf{x}) = \mathtt{H}\mathbf{x}$.*

To interpret this theorem, any point in $\mathbb{P}^2$ is represented as a homogeneous 3-vector, $\mathbf{x}$, and $\mathtt{H}\mathbf{x}$ is a linear mapping of homogeneous coordinates. The theorem asserts that any projectivity arises as such a linear transformation in homogeneous coordinates, and that conversely any such mapping is a projectivity. The theorem will not be proved in full here. It will only be shown that any invertible linear transformation of homogeneous coordinates is a projectivity.

Proof. Let $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ lie on a line $\mathbf{l}$. Thus $\mathbf{l}^\mathsf{T}\mathbf{x}_i = 0$ for $i = 1, \ldots, 3$. Let $\mathtt{H}$ be a non-singular 3×3 matrix. One verifies that $\mathbf{l}^\mathsf{T}\mathtt{H}^{-1}\mathtt{H}\mathbf{x}_i = 0$. Thus, the points $\mathtt{H}\mathbf{x}_i$ all lie on the line $\mathtt{H}^{-\mathsf{T}}\mathbf{l}$, and collinearity is preserved by the transformation.

The converse is considerably harder to prove, namely that each projectivity arises in this way. □

As a result of this theorem, one may give an alternative definition of a projective transformation (or collineation) as follows.

Definition 2.11. Projective transformation. A planar projective transformation is a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{2.5}$$

or more briefly, $\mathbf{x}' = \mathtt{H}\mathbf{x}$.

Note that the matrix $\mathtt{H}$ occurring in this equation may be changed by multiplication by an arbitrary non-zero scale factor without altering the projective transformation. Consequently we say that $\mathtt{H}$ is a *homogeneous* matrix, since as in the homogeneous representation of a point, only the ratio of the matrix elements is significant. There are eight independent ratios amongst the nine elements of $\mathtt{H}$, and it follows that a projective transformation has eight degrees of freedom.

A projective transformation projects every figure into a projectively equivalent figure, leaving all its projective properties invariant. In the ray model of figure 2.1 a projective transformation is simply a linear transformation of $\mathbb{R}^3$.

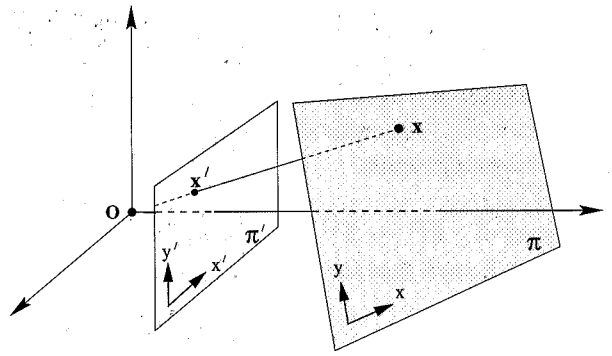


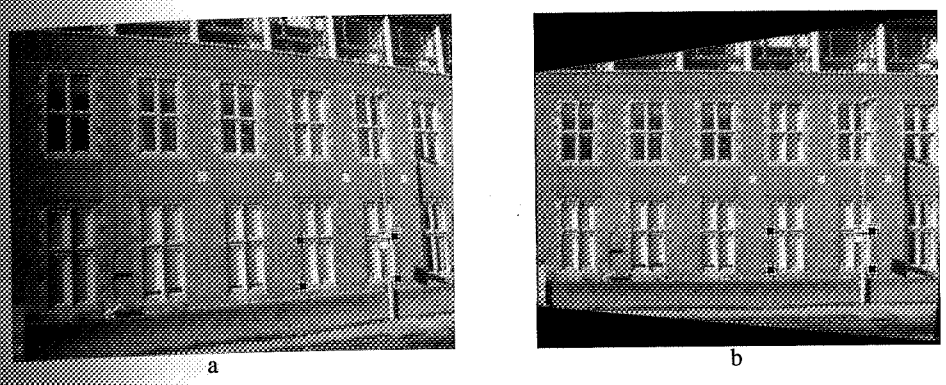
Fig. 2.3. **Central projection maps points on one plane to points on another plane.** *The projection also maps lines to lines as may be seen by considering a plane through the projection centre which intersects with the two planes π and π'. Since lines are mapped to lines, central projection is a projectivity and may be represented by a linear mapping of homogeneous coordinates* $\mathbf{x}' = \mathtt{H}\mathbf{x}$.

Mappings between planes. As an example of how theorem 2.10 may be applied, consider figure 2.3. Projection along rays through a common point (the centre of projection) defines a mapping from one plane to another. It is evident that this point-to-point mapping preserves lines in that a line in one plane is mapped to a line in the other. If a coordinate system is defined in each plane and points are represented in homogeneous coordinates, then the *central projection* mapping may be expressed by $\mathbf{x}' = \mathtt{H}\mathbf{x}$ where $\mathtt{H}$ is a non-singular 3×3 matrix. Actually, if the two coordinate systems defined in the two planes are both Euclidean (rectilinear) coordinate systems then the mapping defined by central projection is more restricted than an arbitrary projective transformation. It is called a *perspectivity* rather than a full projectivity, and may be represented by a transformation with six degrees of freedom. We return to perspectivities in section A7.4(*p*632).

Example 2.12. Removing the projective distortion from a perspective image of a plane.

Shape is distorted under perspective imaging. For instance, in figure 2.4a the windows are not rectangular in the image, although the originals are. In general parallel lines on a scene plane are not parallel in the image but instead converge to a finite point. We have seen that a central projection image of a plane (or section of a plane) is related to the original plane via a projective transformation, and so the image is a projective distortion of the original. It is possible to "undo" this projective transformation by computing the inverse transformation and applying it to the image. The result will be a new synthesized image in which the objects in the plane are shown with their correct geometric shape. This will be illustrated here for the front of the building of figure 2.4a. Note that since the ground and the front are not in the same plane, the projective transformation that must be applied to rectify the front is not the same as the one used for the ground.

Computation of a projective transformation from point-to-point correspondences will be considered in great detail in chapter 4. For now, a method for computing the trans-



a b

Fig. 2.4. **Removing perspective distortion.** *(a) The original image with perspective distortion – the lines of the windows clearly converge at a finite point. (b) Synthesized frontal orthogonal view of the front wall. The image (a) of the wall is related via a projective transformation to the true geometry of the wall. The inverse transformation is computed by mapping the four imaged window corners to corners of an appropriately sized rectangle. The four point correspondences determine the transformation. The transformation is then applied to the whole image. Note that sections of the image of the ground are subject to a further projective distortion. This can also be removed by a projective transformation.*

formation is briefly indicated. One begins by selecting a section of the image corresponding to a planar section of the world. Local 2D image and world coordinates are selected as shown in figure 2.3. Let the inhomogeneous coordinates of a pair of matching points $\mathbf{x}$ and $\mathbf{x}'$ in the world and image plane be (x, y) and (x', y') respectively. We use inhomogeneous coordinates here instead of the homogeneous coordinates of the points, because it is these inhomogeneous coordinates that are measured directly from the image and from the world plane. The projective transformation of (2.5) can be written in inhomogeneous form as

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}.$$

Each point correspondence generates two equations for the elements of $\mathtt{H}$, which after multiplying out are

$$\begin{aligned} x'\,(h_{31}x + h_{32}y + h_{33}) &= h_{11}x + h_{12}y + h_{13} \\ y'\,(h_{31}x + h_{32}y + h_{33}) &= h_{21}x + h_{22}y + h_{23}. \end{aligned}$$

These equations are *linear* in the elements of $\mathtt{H}$. Four point correspondences lead to eight such linear equations in the entries of $\mathtt{H}$, which are sufficient to solve for $\mathtt{H}$ up to an insignificant multiplicative factor. The only restriction is that the four points must be in "general position", which means that no three points are collinear. The inverse of the transformation $\mathtt{H}$ computed in this way is then applied to the whole image to undo the effect of perspective distortion on the selected plane. The results are shown in figure 2.4b. △

Three remarks concerning this example are appropriate: first, the computation of the rectifying transformation $\mathtt{H}$ in this way does not require knowledge of *any* of the camera's parameters or the pose of the plane; second, it is not always necessary to

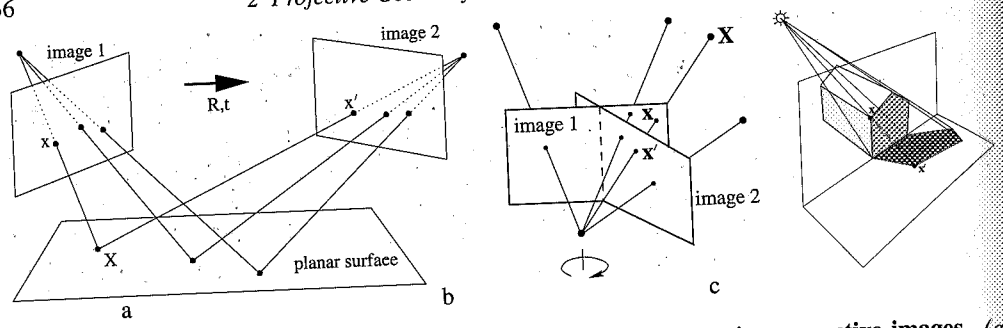


Fig. 2.5. **Examples of a projective transformation, $\mathbf{x}' = \mathtt{H}\mathbf{x}$, arising in perspective images.** *(a) The projective transformation between two images induced by a world plane (the concatenation of two projective transformations is a projective transformation); (b) The projective transformation between two images with the same camera centre (e.g. a camera rotating about its centre or a camera varying its focal length); (c) The projective transformation between the image of a plane (the end of the building) and the image of its shadow onto another plane (the ground plane). Figure (c) courtesy of Luc Van Gool.*

know coordinates for four points in order to remove projective distortion: alternative approaches, which are described in section 2.7, require less, and different types of, information; third, superior (and preferred) methods for computing projective transformations are described in chapter 4.

Projective transformations are important mappings representing many more situations than the perspective imaging of a world plane. A number of other examples are illustrated in figure 2.5. Each of these situations is covered in more detail later in the book.

2.3.1 Transformations of lines and conics

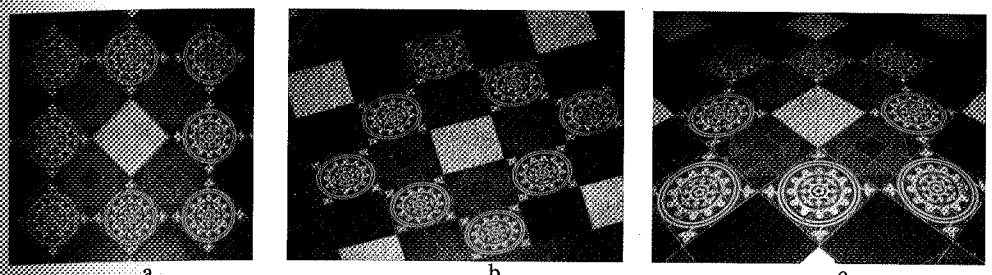
Transformation of lines. It was shown in the proof of theorem 2.10 that if points $\mathbf{x}_i$ lie on a line $\mathbf{l}$, then the transformed points $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$ under a projective transformation lie on the line $\mathbf{l}' = \mathtt{H}^{-\mathsf{T}}\mathbf{l}$. In this way, incidence of points on lines is preserved, since $\mathbf{l}'^{\mathsf{T}}\mathbf{x}'_i = \mathbf{l}^{\mathsf{T}}\mathtt{H}^{-1}\mathtt{H}\mathbf{x}_i = 0$. This gives the transformation rule for lines:
Under the point transformation $\mathbf{x}' = \mathtt{H}\mathbf{x}$, a line transforms as

$$\mathbf{l}' = \mathtt{H}^{-\mathsf{T}}\mathbf{l}. \tag{2.6}$$

One may alternatively write $\mathbf{l}'^{\mathsf{T}} = \mathbf{l}^{\mathsf{T}}\mathtt{H}^{-1}$. Note the fundamentally different way in which lines and points transform. Points transform according to $\mathtt{H}$, whereas lines (as rows) transform according to $\mathtt{H}^{-1}$. This may be explained in terms of "covariant" or "contravariant" behaviour. One says that points transform *contravariantly* and lines transform *covariantly*. This distinction will be taken up again, when we discuss tensors in chapter 15 and is fully explained in appendix 1(p562).

Transformation of conics. Under a point transformation $\mathbf{x}' = \mathtt{H}\mathbf{x}$, (2.2) becomes

$$\begin{aligned} \mathbf{x}^{\mathsf{T}}\mathtt{C}\mathbf{x} &= \mathbf{x}'^{\mathsf{T}}[\mathtt{H}^{-1}]^{\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}\mathbf{x}' \\ &= \mathbf{x}'^{\mathsf{T}}\mathtt{H}^{-\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}\mathbf{x}' \end{aligned}$$



a b c

Fig. 2.6. **Distortions arising under central projection.** *Images of a tiled floor. (a)* **Similarity***: the circular pattern is imaged as a circle. A square tile is imaged as a square. Lines which are parallel or perpendicular have the same relative orientation in the image. (b)* **Affine***: The circle is imaged as an ellipse. Orthogonal world lines are not imaged as orthogonal lines. However, the sides of the square tiles, which are parallel in the world are parallel in the image. (c)* **Projective***: Parallel world lines are imaged as converging lines. Tiles closer to the camera have a larger image than those further away.*

which is a quadratic form $\mathbf{x}'^{\mathsf{T}}\mathtt{C}'\mathbf{x}'$ with $\mathtt{C}' = \mathtt{H}^{-\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}$. This gives the transformation rule for a conic:

Result 2.13. *Under a point transformation* $\mathbf{x}' = \mathtt{H}\mathbf{x}$, *a conic* $\mathtt{C}$ *transforms to* $\mathtt{C}' = \mathtt{H}^{-\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}$.

The presence of $\mathtt{H}^{-1}$ in this equation may be expressed by saying that a conic transforms *covariantly*. The transformation rule for a dual conic is derived in a similar manner. This gives:

Result 2.14. *Under a point transformation* $\mathbf{x}' = \mathtt{H}\mathbf{x}$, *a dual conic* $\mathtt{C}^*$ *transforms to* $\mathtt{C}^{*\prime} = \mathtt{H}\mathtt{C}^*\mathtt{H}^{\mathsf{T}}$.

2.4 A hierarchy of transformations

In this section we describe the important specializations of a projective transformation and their geometric properties. It was shown in section 2.3 that projective transformations form a group. This group is called the *projective linear group*, and it will be seen that these specializations are *subgroups* of this group.

The group of invertible $n \times n$ matrices with real elements is the (real) general linear group on n dimensions, or $GL(n)$. To obtain the projective linear group the matrices related by a scalar multiplier are identified, giving $PL(n)$ (this is a quotient group of $GL(n)$). In the case of projective transformations of the plane $n = 3$.

The important subgroups of $PL(3)$ include the *affine group*, which is the subgroup of $PL(3)$ consisting of matrices for which the last row is $(0, 0, 1)$, and the *Euclidean group*, which is a subgroup of the affine group for which in addition the upper left hand 2×2 matrix is orthogonal. One may also identify the *oriented Euclidean group* in which the upper left hand 2×2 matrix has determinant 1.

We will introduce these transformations starting from the most specialized, the isometries, and progressively generalizing until projective transformations are reached.

This defines a *hierarchy* of transformations. The distortion effects of various transformations in this hierarchy are shown in figure 2.6.

Some transformations of interest are not groups, for example, perspectivities (because the composition of two perspectivities is a projectivity, not a perspectivity). This point is covered in section A7.4(*p*632).

Invariants. An alternative to describing the transformation *algebraically*, i.e. as a matrix acting on coordinates of a point or curve, is to describe the transformation in terms of those elements or quantities that are preserved or *invariant*. A (scalar) invariant of a geometric configuration is a function of the configuration whose value is unchanged by a particular transformation. For example, the separation of two points is unchanged by a Euclidean transformation (translation and rotation), but not by a similarity (e.g. translation, rotation and isotropic scaling). Distance is thus a Euclidean, but not similarity invariant. The angle between two lines is both a Euclidean and a similarity invariant.

2.4.1 Class I: Isometries

Isometries are transformations of the plane $\mathbb{R}^2$ that preserve Euclidean distance (from *iso* = same, *metric* = measure). An isometry is represented as

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon\cos\theta & -\sin\theta & t_x \\ \epsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

where $\epsilon = \pm 1$. If $\epsilon = 1$ then the isometry is *orientation-preserving* and is a *Euclidean* transformation (a composition of a translation and rotation). If $\epsilon = -1$ then the isometry reverses orientation. An example is the composition of a reflection, represented by the matrix $\mathrm{diag}(-1, 1, 1)$, with a Euclidean transformation.

Euclidean transformations model the motion of a rigid object. They are by far the most important isometries in practice, and we will concentrate on these. However, the orientation reversing isometries often arise as ambiguities in structure recovery.

A planar Euclidean transformation can be written more concisely in block form as

$$\mathbf{x}' = \mathtt{H}_{\mathrm{E}}\mathbf{x} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \mathbf{x} \tag{2.7}$$

where $\mathtt{R}$ is a 2×2 rotation matrix (an orthogonal matrix such that $\mathtt{R}^{\mathsf{T}}\mathtt{R} = \mathtt{R}\mathtt{R}^{\mathsf{T}} = \mathtt{I}$), $\mathbf{t}$ a translation 2-vector, and $\mathbf{0}$ a null 2-vector. Special cases are a pure rotation (when $\mathbf{t} = \mathbf{0}$) and a pure translation (when $\mathtt{R} = \mathtt{I}$). A Euclidean transformation is also known as a *displacement*.

A planar Euclidean transformation has three degrees of freedom, one for the rotation and two for the translation. Thus three parameters must be specified in order to define the transformation. The transformation can be computed from two point correspondences.

Invariants. The invariants are very familiar, for instance: length (the distance between two points), angle (the angle between two lines), and area.

Groups and orientation. An isometry is orientation-preserving if the upper left hand 2×2 matrix has determinant 1. Orientation-*preserving* isometries form a group, orientation-*reversing* ones do not. This distinction applies also in the case of similarity and affine transformations which now follow.

2.4.2 Class II: Similarity transformations

A similarity transformation (or more simply a *similarity*) is an isometry composed with an isotropic scaling. In the case of a Euclidean transformation composed with a scaling (i.e. no reflection) the similarity has matrix representation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{2.8}$$

This can be written more concisely in block form as

$$\mathbf{x}' = \mathtt{H}_s\mathbf{x} = \begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \mathbf{x} \tag{2.9}$$

where the scalar s represents the isotropic scaling. A similarity transformation is also known as an *equi-form* transformation, because it preserves "shape" (form). A planar similarity transformation has four degrees of freedom, the scaling accounting for one more degree of freedom than a Euclidean transformation. A similarity can be computed from two point correspondences.

Invariants. The invariants can be constructed from Euclidean invariants with suitable provision being made for the additional scaling degree of freedom. Angles between lines are not affected by rotation, translation or isotropic scaling, and so are similarity invariants. In particular parallel lines are mapped to parallel lines. The length between two points is not a similarity invariant, but the *ratio* of two lengths is an invariant, because the scaling of the lengths cancels out. Similarly a ratio of areas is an invariant because the scaling (squared) cancels out.

Metric structure. A term that will be used frequently in the discussion on reconstruction (chapter 10) is *metric*. The description *metric structure* implies that the structure is defined up to a similarity.

2.4.3 Class III: Affine transformations

An affine transformation (or more simply an *affinity*) is a non-singular linear transformation followed by a translation. It has the matrix representation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.10}$$

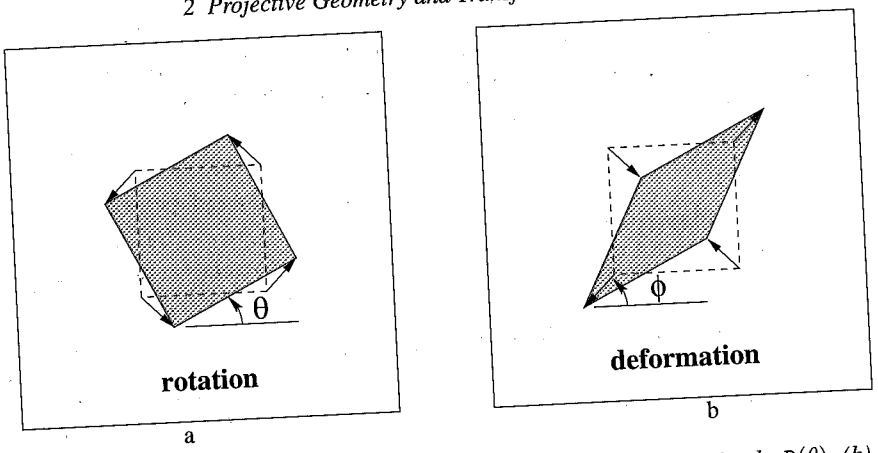


Fig. 2.7. **Distortions arising from a planar affine transformation.** *(a) Rotation by* $\mathtt{R}(\theta)$. *(b) A deformation* $\mathtt{R}(-\phi)\,\mathtt{D}\,\mathtt{R}(\phi)$. *Note, the scaling directions in the deformation are orthogonal.*

or in block form

$$\mathbf{x}' = \mathtt{H}_{\mathtt{A}}\mathbf{x} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \mathbf{x} \tag{2.11}$$

with $\mathtt{A}$ a 2×2 non-singular matrix. A planar affine transformation has six degrees of freedom corresponding to the six matrix elements. The transformation can be computed from three point correspondences.

A helpful way to understand the geometric effects of the linear component $\mathtt{A}$ of an affine transformation is as the composition of two fundamental transformations, namely rotations and non-isotropic scalings. The affine matrix $\mathtt{A}$ can always be decomposed as

$$\mathtt{A} \;=\; \mathtt{R}(\theta)\,\mathtt{R}(-\phi)\,\mathtt{D}\,\mathtt{R}(\phi) \tag{2.12}$$

where $\mathtt{R}(\theta)$ and $\mathtt{R}(\phi)$ are rotations by θ and ϕ respectively, and $\mathtt{D}$ is a diagonal matrix:

$$\mathtt{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

This decomposition follows directly from the SVD (section A4.4(p585)): writing $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^{\mathsf{T}} = (\mathtt{U}\mathtt{V}^{\mathsf{T}})(\mathtt{V}\mathtt{D}\mathtt{V}^{\mathsf{T}}) = \mathtt{R}(\theta)\,(\mathtt{R}(-\phi)\,\mathtt{D}\,\mathtt{R}(\phi))$, since $\mathtt{U}$ and $\mathtt{V}$ are orthogonal matrices.

The affine matrix $\mathtt{A}$ is hence seen to be the concatenation of a rotation (by ϕ); a scaling by λ_1 and λ_2 respectively in the (rotated) x and y directions; a rotation back (by $-\phi$); and finally another rotation (by θ). The only "new" geometry, compared to a similarity, is the non-isotropic scaling. This accounts for the two extra degrees of freedom possessed by an affinity over a similarity. They are the angle ϕ specifying the scaling direction, and the ratio of the scaling parameters $\lambda_1 : \lambda_2$. The essence of an affinity is this scaling in orthogonal directions, oriented at a particular angle. Schematic examples are given in figure 2.7.

Invariants. Because an affine transformation includes non-isotropic scaling, the similarity invariants of length ratios and angles between lines are not preserved under an affinity. Three important invariants are:

(i) **Parallel lines.** Consider two parallel lines. These intersect at a point $(x_1, x_2, 0)^\mathsf{T}$ at infinity. Under an affine transformation this point is mapped to another point at infinity. Consequently, the parallel lines are mapped to lines which still intersect at infinity, and so are parallel after the transformation.

(ii) **Ratio of lengths of parallel line segments.** The length scaling of a line segment depends only on the angle between the line direction and scaling directions. Suppose the line is at angle α to the x-axis of the orthogonal scaling direction, then the scaling magnitude is $\sqrt{\lambda_1^2 \cos^2\alpha + \lambda_2^2 \sin^2\alpha}$. This scaling is common to all lines with the same direction, and so cancels out in a ratio of parallel segment lengths.

(iii) **Ratio of areas.** This invariance can be deduced directly from the decomposition (2.12). Rotations and translations do not affect area, so only the scalings by λ_1 and λ_2 matter here. The effect is that area is scaled by $\lambda_1\lambda_2$ which is equal to $\det \mathtt{A}$. Thus the area of any shape is scaled by $\det \mathtt{A}$, and so the scaling cancels out for a ratio of areas. It will be seen that this does not hold for a projective transformation.

An affinity is orientation-preserving or -reversing according to whether $\det \mathtt{A}$ is positive or negative respectively. Since $\det \mathtt{A} = \lambda_1\lambda_2$ the property depends only on the sign of the scalings.

2.4.4 Class IV: Projective transformations

A projective transformation was defined in (2.5). It is a general non-singular linear transformation of *homogeneous* coordinates. This generalizes an affine transformation, which is the composition of a general non-singular linear transformation of *inhomogeneous* coordinates and a translation. We have earlier seen the action of a projective transformation (in section 2.3). Here we examine its block form

$$\mathbf{x}' = \mathtt{H}_\mathrm{P}\mathbf{x} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} \mathbf{x} \tag{2.13}$$

where the vector $\mathbf{v} = (v_1, v_2)^\mathsf{T}$. The matrix has nine elements with only their ratio significant, so the transformation is specified by eight parameters. Note, it is not always possible to scale the matrix such that v is unity since v might be zero. A projective transformation between two planes can be computed from four point correspondences, with no three collinear on either plane. See figure 2.4.

Unlike the case of affinities, it is not possible to distinguish between orientation preserving and orientation reversing projectivities in $\mathbb{P}^2$. We will return to this point in section 2.6.

Invariants. The most fundamental projective invariant is the cross ratio of four collinear points: a ratio of lengths on a line is invariant under affinities, but not under projectivities. However, a ratio of ratios or *cross ratio* of lengths on a line is a projective invariant. We return to properties of this invariant in section 2.5.

2.4.5 Summary and comparison

Affinities (6 dof) occupy the middle ground between similarities (4 dof) and projectivities (8 dof). They generalize similarities in that angles are not preserved, so that shapes are skewed under the transformation. On the other hand their action is homogeneous over the plane: for a given affinity the $\det \mathtt{A}$ scaling in area of an object (e.g. a square) is the same anywhere on the plane; and the orientation of a transformed line depends only on its initial orientation, not on its position on the plane. In contrast, for a given projective transformation, area scaling varies with position (e.g. under perspective a more distant square on the plane has a smaller image than one that is nearer, as in figure 2.6); and the orientation of a transformed line depends on both the orientation and position of the source line (however, it will be seen later in section 8.6(p213) that a line's vanishing point depends only on line orientation, not position).

The key difference between a projective and affine transformation is that the vector $\mathbf{v}$ is not null for a projectivity. This is responsible for the non-linear effects of the projectivity. Compare the mapping of an ideal point $(x_1, x_2, 0)^\mathsf{T}$ under an affinity and projectivity: First the affine transformation

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix}. \tag{2.14}$$

Second the projective transformation

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1 x_1 + v_2 x_2 \end{pmatrix}. \tag{2.15}$$

In the first case the ideal point remains ideal (i.e. at infinity). In the second it is mapped to a finite point. It is this ability which allows a projective transformation to model vanishing points.

2.4.6 Decomposition of a projective transformation

A projective transformation can be decomposed into a chain of transformations, where each matrix in the chain represents a transformation higher in the hierarchy than the previous one.

$$\mathtt{H} = \mathtt{H}_\mathtt{S}\,\mathtt{H}_\mathtt{A}\,\mathtt{H}_\mathtt{P} = \begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathtt{K} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathtt{I} & \mathbf{0} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} \tag{2.16}$$

with $\mathtt{A}$ a non-singular matrix given by $\mathtt{A} = s\mathtt{R}\mathtt{K} + \mathbf{t}\mathbf{v}^\mathsf{T}$, and $\mathtt{K}$ an upper-triangular matrix normalized as $\det \mathtt{K} = 1$. This decomposition is valid provided $v \neq 0$, and is unique if s is chosen positive.

Each of the matrices $\mathtt{H}_S, \mathtt{H}_A, \mathtt{H}_P$ is the "essence" of a transformation of that type (as indicated by the subscripts S, A, P). Consider the process of rectifying the perspective image of a plane as in example 2.12: $\mathtt{H}_P$ (2 dof) moves the line at infinity; $\mathtt{H}_A$ (2 dof) affects the affine properties, but does not move the line at infinity; and finally, $\mathtt{H}_S$ is a general similarity transformation (4 dof) which does not affect the affine or projective properties. The transformation $\mathtt{H}_P$ is an *elation*, described in section A7.3(*p*631).

Example 2.15. The projective transformation

$$\mathtt{H} = \begin{bmatrix} 1.707 & 0.586 & 1.0 \\ 2.707 & 8.242 & 2.0 \\ 1.0 & 2.0 & 1.0 \end{bmatrix}$$

may be decomposed as

$$\mathtt{H} = \begin{bmatrix} 2\cos 45^\circ & -2\sin 45^\circ & 1 \\ 2\sin 45^\circ & 2\cos 45^\circ & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

△

This decomposition can be employed when the objective is to only partially determine the transformation. For example, if one wants to measure length ratios from the perspective image of a plane, then it is only necessary to determine (rectify) the transformation up to a similarity. We return to this approach in section 2.7.

Taking the inverse of $\mathtt{H}$ in (2.16) gives $\mathtt{H}^{-1} = \mathtt{H}_P^{-1}\mathtt{H}_A^{-1}\mathtt{H}_S^{-1}$. Since $\mathtt{H}_P^{-1}, \mathtt{H}_A^{-1}$ and $\mathtt{H}_S^{-1}$ are still projective, affine and similarity transformations respectively, a general projective transformation may also be decomposed in the form

$$\mathtt{H} = \mathtt{H}_P\,\mathtt{H}_A\,\mathtt{H}_S = \begin{bmatrix} \mathtt{I} & \mathbf{0} \\ \mathbf{v}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathtt{K} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{2.17}$$

Note that the actual values of $\mathtt{K}$, $\mathtt{R}$, $\mathbf{t}$ and $\mathbf{v}$ will be different from those of (2.16).

2.4.7 The number of invariants

The question naturally arises as to how many invariants there are for a given geometric configuration under a particular transformation. First the term "number" needs to be made more precise, for if a quantity is invariant, such as length under Euclidean transformations, then any function of that quantity is invariant. Consequently, we seek a counting argument for the number of functionally independent invariants. By considering the number of transformation parameters that must be eliminated in order to form an invariant, it can be seen that:

Result 2.16. *The number of functionally independent invariants is equal to, or greater than, the number of degrees of freedom of the configuration less the number of degrees of freedom of the transformation.*

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, **I**, **J** (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

Table 2.1. **Geometric properties invariant to commonly occurring *planar* transformations.** *The matrix* $\mathtt{A} = [a_{ij}]$ *is an invertible* 2×2 *matrix,* $\mathtt{R} = [r_{ij}]$ *is a 2D rotation matrix, and* (t_x, t_y) *a 2D translation. The distortion column shows typical effects of the transformations on a square. Transformations higher in the table can produce all the actions of the ones below. These range from Euclidean, where only translations and rotations occur, to projective where the square can be transformed to any arbitrary quadrilateral (provided no three points are collinear).*

For example, a configuration of four points in general position has 8 degrees of freedom (2 for each point), and so 4 similarity, 2 affinity and zero projective invariants since these transformations have respectively 4, 6 and 8 degrees of freedom.

Table 2.1 summarizes the 2D transformation groups and their invariant properties. Transformations lower in the table are specializations of those above. A transformation lower in the table inherits the invariants of those above.

2.5 The projective geometry of 1D

The development of the projective geometry of a line, $\mathbb{P}^1$, proceeds in much the same way as that of the plane. A point x on the line is represented by homogeneous coordinates $(x_1, x_2)^\mathsf{T}$, and a point for which $x_2 = 0$ is an ideal point of the line. We will use the notation $\bar{\mathbf{x}}$ to represent the 2-vector $(x_1, x_2)^\mathsf{T}$. A projective transformation of a line is represented by a 2×2 homogeneous matrix,

$$\bar{\mathbf{x}}' = \mathtt{H}_{2\times 2}\bar{\mathbf{x}}$$

and has 3 degrees of freedom corresponding to the four elements of the matrix less one for overall scaling. A projective transformation of a line may be determined from three corresponding points.

4

Estimation – 2D Projective Transformations

In this chapter, we consider the problem of estimation. In the present context this will be taken to mean the computation of some transformation or other mathematical quantity, based on measurements of some nature. This definition is somewhat vague, so to make it more concrete, here are a number of estimation problems of the type that we would like to consider.

(i) **2D homography.** Given a set of points $\mathbf{x}_i$ in $\mathbb{P}^2$ and a corresponding set of points $\mathbf{x}'_i$ likewise in $\mathbb{P}^2$, compute the projective transformation that takes each $\mathbf{x}_i$ to $\mathbf{x}'_i$. In a practical situation, the points $\mathbf{x}_i$ and $\mathbf{x}'_i$ are points in two images (or the same image), each image being considered as a projective plane $\mathbb{P}^2$.

(ii) **3D to 2D camera projection.** Given a set of points $\mathbf{X}_i$ in 3D space, and a set of corresponding points $\mathbf{x}_i$ in an image, find the 3D to 2D projective mapping that maps $\mathbf{X}_i$ to $\mathbf{x}_i$. Such a 3D to 2D projection is the mapping carried out by a projective camera, as discussed in chapter 6.

(iii) **Fundamental matrix computation.** Given a set of points $\mathbf{x}_i$ in one image, and corresponding points $\mathbf{x}'_i$ in another image, compute the fundamental matrix $\mathtt{F}$ consistent with these correspondences. The fundamental matrix, discussed in chapter 9, is a singular 3×3 matrix $\mathtt{F}$ satisfying $\mathbf{x}_i'^{\mathsf{T}}\mathtt{F}\mathbf{x}_i = 0$ for all i.

(iv) **Trifocal tensor computation.** Given a set of point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$ across three images, compute the trifocal tensor. The trifocal tensor, discussed in chapter 15, is a tensor $\mathcal{T}_i^{jk}$ relating points or lines in three views.

These problems have many features in common, and the considerations that relate to one of the problems are also relevant to each of the others. Therefore, in this chapter, the first of these problems will be considered in detail. What we learn about ways of solving this problem will teach us how to proceed in solving each of the other problems as well.

Apart from being important for illustrative purposes, the problem of estimating 2D projective transformations is of importance in its own right. We consider a set of point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ between two images. Our problem is to compute a 3×3 matrix $\mathtt{H}$ such that $\mathtt{H}\mathbf{x}_i = \mathbf{x}'_i$ for each i.

Number of measurements required. The first question to consider is how many corresponding points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are required to compute the projective transformation $\mathtt{H}$. A lower bound is available by a consideration of the number of degrees of freedom and number of constraints. On the one hand, the matrix $\mathtt{H}$ contains 9 entries, but is defined only up to scale. Thus, the total number of degrees of freedom in a 2D projective transformation is 8. On the other hand, each point-to-point correspondence accounts for two constraints, since for each point $\mathbf{x}_i$ in the first image the two degrees of freedom of the point in the second image must correspond to the mapped point $\mathtt{H}\mathbf{x}_i$. A 2D point has two degrees of freedom corresponding to the x and y components, each of which may be specified separately. Alternatively, the point is specified as a homogeneous 3-vector, which also has two degrees of freedom since scale is arbitrary. As a consequence, it is necessary to specify four point correspondences in order to constrain $\mathtt{H}$ fully.

Approximate solutions. It will be seen that if exactly four correspondences are given, then an exact solution for the matrix $\mathtt{H}$ is possible. This is the *minimal* solution. Such solutions are important as they define the size of the subsets required in robust estimation algorithms, such as RANSAC, described in section 4.7. However, since points are measured inexactly ("noise"), if more than four such correspondences are given, then these correspondences may not be fully compatible with any projective transformation, and one will be faced with the task of determining the "best" transformation given the data. This will generally be done by finding the transformation $\mathtt{H}$ that minimizes some cost function. Different cost functions will be discussed during this chapter, together with methods for minimizing them. There are two main categories of cost functions: those based on minimizing an algebraic error; and those based on minimizing a geometric or statistical image distance. These two categories are described in section 4.2.

The Gold Standard algorithm. There will usually be one cost function which is optimal in the sense that the $\mathtt{H}$ that minimizes it gives the best possible estimate of the transformation under certain assumptions. The computational algorithm that enables this cost function to be minimized is called the "Gold Standard" algorithm. The results of other algorithms are assessed by how well they compare to this Gold Standard. In the case of estimating a homography between two views the cost function is (4.8), the assumptions for optimality are given in section 4.3, and the Gold Standard is algorithm 4.3(p114).

4.1 The Direct Linear Transformation (DLT) algorithm

We begin with a simple linear algorithm for determining $\mathtt{H}$ given a set of four 2D to 2D point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. The transformation is given by the equation $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$. Note that this is an equation involving homogeneous vectors; thus the 3-vectors $\mathbf{x}'_i$ and $\mathtt{H}\mathbf{x}_i$ are not equal, they have the same direction but may differ in magnitude by a non-zero scale factor. The equation may be expressed in terms of the vector cross product as $\mathbf{x}'_i \times \mathtt{H}\mathbf{x}_i = \mathbf{0}$. This form will enable a simple linear solution for $\mathtt{H}$ to be derived.

If the j-th row of the matrix $\mathtt{H}$ is denoted by $\mathbf{h}^{j\mathsf{T}}$, then we may write

$$\mathtt{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\mathsf{T}}\mathbf{x}_i \\ \mathbf{h}^{2\mathsf{T}}\mathbf{x}_i \\ \mathbf{h}^{3\mathsf{T}}\mathbf{x}_i \end{pmatrix}.$$

Writing $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\mathsf{T}$, the cross product may then be given explicitly as

$$\mathbf{x}'_i \times \mathtt{H}\mathbf{x}_i = \begin{pmatrix} y'_i\mathbf{h}^{3\mathsf{T}}\mathbf{x}_i - w'_i\mathbf{h}^{2\mathsf{T}}\mathbf{x}_i \\ w'_i\mathbf{h}^{1\mathsf{T}}\mathbf{x}_i - x'_i\mathbf{h}^{3\mathsf{T}}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\mathsf{T}}\mathbf{x}_i - y'_i\mathbf{h}^{1\mathsf{T}}\mathbf{x}_i \end{pmatrix}.$$

Since $\mathbf{h}^{j\mathsf{T}}\mathbf{x}_i = \mathbf{x}_i^\mathsf{T}\mathbf{h}^j$ for $j = 1, \ldots, 3$, this gives a set of three equations in the entries of $\mathtt{H}$, which may be written in the form

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w'_i\mathbf{x}_i^\mathsf{T} & y'_i\mathbf{x}_i^\mathsf{T} \\ w'_i\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x'_i\mathbf{x}_i^\mathsf{T} \\ -y'_i\mathbf{x}_i^\mathsf{T} & x'_i\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \tag{4.1}$$

These equations have the form $\mathtt{A}_i\mathbf{h} = \mathbf{0}$, where $\mathtt{A}_i$ is a 3×9 matrix, and $\mathbf{h}$ is a 9-vector made up of the entries of the matrix $\mathtt{H}$,

$$\mathbf{h} = \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix}, \qquad \mathtt{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{4.2}$$

with h_i the i-th element of $\mathbf{h}$. Three remarks regarding these equations are in order here.

(i) The equation $\mathtt{A}_i\mathbf{h} = \mathbf{0}$ is an equation *linear* in the unknown $\mathbf{h}$. The matrix elements of $\mathtt{A}_i$ are quadratic in the known coordinates of the points.

(ii) Although there are three equations in (4.1), only two of them are linearly independent (since the third row is obtained, up to scale, from the sum of x'_i times the first row and y'_i times the second). Thus each point correspondence gives two equations in the entries of $\mathtt{H}$. It is usual to omit the third equation in solving for $\mathtt{H}$ ([Sutherland-63]). Then (for future reference) the set of equations becomes

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w'_i\mathbf{x}_i^\mathsf{T} & y'_i\mathbf{x}_i^\mathsf{T} \\ w'_i\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x'_i\mathbf{x}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \tag{4.3}$$

This will be written

$$\mathtt{A}_i\mathbf{h} = \mathbf{0}$$

where $\mathtt{A}_i$ is now the 2×9 matrix of (4.3).

(iii) The equations hold for any homogeneous coordinate representation $(x'_i, y'_i, w'_i)^\mathsf{T}$ of the point $\mathbf{x}'_i$. One may choose $w'_i = 1$, which means that (x'_i, y'_i) are the coordinates measured in the image. Other choices are possible, however, as will be seen later.

Solving for H

Each point correspondence gives rise to two independent equations in the entries of H. Given a set of four such point correspondences, we obtain a set of equations $\mathtt{A}\mathbf{h} = \mathbf{0}$, where A is the matrix of equation coefficients built from the matrix rows $\mathtt{A}_i$ contributed from each correspondence, and $\mathbf{h}$ is the vector of unknown entries of H. We seek a non-zero solution $\mathbf{h}$, since the obvious solution $\mathbf{h} = \mathbf{0}$ is of no interest to us. If (4.1) is used then A has dimension 12×9, and if (4.3) the dimension is 8×9. In either case A has rank 8, and thus has a 1-dimensional null-space which provides a solution for $\mathbf{h}$. Such a solution $\mathbf{h}$ can only be determined up to a non-zero scale factor. However, H is in general only determined up to scale, so the solution $\mathbf{h}$ gives the required H. A scale may be arbitrarily chosen for $\mathbf{h}$ by a requirement on its norm such as $\|\mathbf{h}\| = 1$.

4.1.1 Over-determined solution

If more than four point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are given, then the set of equations $\mathtt{A}\mathbf{h} = \mathbf{0}$ derived from (4.3) is over-determined. If the position of the points is exact then the matrix A will still have rank 8, a one dimensional null-space, and there is an exact solution for $\mathbf{h}$. This will not be the case if the measurement of image coordinates is inexact (generally termed *noise*) – there will not be an exact solution to the over-determined system $\mathtt{A}\mathbf{h} = \mathbf{0}$ apart from the zero solution. Instead of demanding an exact solution, one attempts to find an approximate solution, namely a vector $\mathbf{h}$ that minimizes a suitable cost function. The question that naturally arises then is: what should be minimized? Clearly, to avoid the solution $\mathbf{h} = \mathbf{0}$ an additional constraint is required. Generally, a condition on the norm is used, such as $\|\mathbf{h}\| = 1$. The value of the norm is unimportant since H is only defined up to scale. Given that there is no exact solution to $\mathtt{A}\mathbf{h} = \mathbf{0}$, it seems natural to attempt to minimize the norm $\|\mathtt{A}\mathbf{h}\|$ instead, subject to the usual constraint, $\|\mathbf{h}\| = 1$. This is identical to the problem of finding the minimum of the quotient $\|\mathtt{A}\mathbf{h}\|/\|\mathbf{h}\|$. As shown in section A5.3(*p*592) the solution is the (unit) eigenvector of $\mathtt{A}^\mathsf{T}\mathtt{A}$ with least eigenvalue. Equivalently, the solution is the unit singular vector corresponding to the smallest singular value of A. The resulting algorithm, known as the basic DLT algorithm, is summarized in algorithm 4.1.

4.1.2 Inhomogeneous solution

An alternative to solving for $\mathbf{h}$ directly as a homogeneous vector is to turn the set of equations (4.3) into a inhomogeneous set of linear equations by imposing a condition $h_j = 1$ for some entry of the vector $\mathbf{h}$. Imposing the condition $h_j = 1$ is justified by the observation that the solution is determined only up to scale, and this scale can be chosen such that $h_j = 1$. For example, if the last element of $\mathbf{h}$, which corresponds to $\mathtt{H}_{33}$, is chosen as unity then the resulting equations derived from (4.3) are

$$\begin{bmatrix} 0 & 0 & 0 & -x_i w'_i & -y_i w'_i & -w_i w'_i & x_i y'_i & y_i y'_i \\ x_i w'_i & y_i w'_i & w_i w'_i & 0 & 0 & 0 & -x_i x'_i & -y_i x'_i \end{bmatrix} \tilde{\mathbf{h}} = \begin{pmatrix} -w_i y'_i \\ w_i x'_i \end{pmatrix}$$

where $\tilde{\mathbf{h}}$ is an 8-vector consisting of the first 8 components of $\mathbf{h}$. Concatenating the equations from four correspondences then generates a matrix equation of the form

in the entries of $\mathbf{h}$.
equations $\mathtt{A}\mathbf{h} = \mathbf{0}$,
rows $\mathtt{A}_i$ contributed
es of $\mathtt{H}$. We seek a
rest to us. If (4.1) is
$\times 9$. In either case
des a solution for $\mathbf{h}$.
actor. However, $\mathtt{H}$ is
required $\mathtt{H}$. A scale
as $\|\mathbf{h}\| = 1$.

the set of equations
of the points is exact
pace, and there is an
of image coordinates
solution to the over-
ead of demanding an
imely a vector $\mathbf{h}$ that
arises then is: what
dditional constraint is
$\mathbf{h}\| = 1$. The value of
h that there is no exact
e norm $\|\mathtt{A}\mathbf{h}\|$ instead.
he problem of finding
5.3(p592) the solution
ntly, the solution is the
ue of $\mathtt{A}$. The resulting
algorithm 4.1.

tor is to turn the set of
y imposing a condition
n $h_j = 1$ is justified by
e, and this scale can be
h, which corresponds to
n (4.3) are

$$\begin{bmatrix} y_i' \\ x_i' \end{bmatrix} \tilde{\mathbf{h}} = \begin{pmatrix} -w_i y_i' \\ w_i x_i' \end{pmatrix}$$

of $\mathbf{h}$. Concatenating the
ix equation of the form

Objective

Given $n \geq 4$ 2D to 2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i'\}$, determine the 2D homography matrix $\mathtt{H}$ such that $\mathbf{x}_i' = \mathtt{H}\mathbf{x}_i$.

Algorithm

(i) For each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$ compute the matrix $\mathtt{A}_i$ from (4.1). Only the first two rows need be used in general.

(ii) Assemble the n 2×9 matrices $\mathtt{A}_i$ into a single $2n \times 9$ matrix $\mathtt{A}$.

(iii) Obtain the SVD of $\mathtt{A}$ (section A4.4(p585)). The unit singular vector corresponding to the smallest singular value is the solution $\mathbf{h}$. Specifically, if $\mathtt{A} = \mathtt{UDV}^\mathsf{T}$ with $\mathtt{D}$ diagonal with positive diagonal entries, arranged in descending order down the diagonal, then $\mathbf{h}$ is the last column of $\mathtt{V}$.

(iv) The matrix $\mathtt{H}$ is determined from $\mathbf{h}$ as in (4.2).

Algorithm 4.1. *The basic DLT for* $\mathtt{H}$ *(but see algorithm 4.2(*p*109) which includes normalization).*

$\mathtt{M}\mathbf{h} = \mathbf{b}$, where $\mathtt{M}$ has 8 columns and $\mathbf{b}$ is an 8-vector. Such an equation may be solved for $\mathbf{h}$ using standard techniques for solving linear equations (such as Gaussian elimination) in the case where $\mathtt{M}$ contains just 8 rows (the minimum case), or by least-squares techniques (section A5.1(p588)) in the case of an over-determined set of equations.

However, if in fact $h_j = 0$ is the true solution, then no multiplicative scale k can exist such that $kh_j = 1$. This means that the true solution cannot be reached. For this reason, this method can be expected to lead to unstable results in the case where the chosen h_j is close to zero. Consequently, this method is *not* recommended in general.

Example 4.1. It will be shown that $h_9 = \mathtt{H}_{33}$ is zero if the coordinate origin is mapped to a point at infinity by $\mathtt{H}$. Since $(0,0,1)^\mathsf{T}$ represents the coordinate origin $\mathbf{x}_0$, and also $(0,0,1)^\mathsf{T}$ represents the line at infinity $\mathbf{l}$, this condition may be written as $\mathbf{l}^\mathsf{T}\mathtt{H}\mathbf{x}_0 = (0,0,1)\mathtt{H}(0,0,1)^\mathsf{T} = 0$, thus $\mathtt{H}_{33} = 0$. In a perspective image of a scene plane the line at infinity is imaged as the vanishing line of the plane (see chapter 8), for example the horizon is the vanishing line of the ground plane. It is not uncommon for the horizon to pass through the image centre, and for the coordinate origin to coincide with the image centre. In this case the mapping that takes the image to the world plane maps the origin to the line at infinity, so that the true solution has $\mathtt{H}_{33} = h_9 = 0$. Consequently, an $h_9 = 1$ normalization can be a serious failing in practical situations. △

4.1.3 Degenerate configurations

Consider a minimal solution in which a homography is computed using four point correspondences, and suppose that three of the points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are collinear. The question is whether this is significant. If the corresponding points $\mathbf{x}_1', \mathbf{x}_2', \mathbf{x}_3'$ are also collinear then one might suspect that the homography is not sufficiently constrained, and there will exist a family of homographies mapping $\mathbf{x}_i$ to $\mathbf{x}_i'$. On the other hand, if the corresponding points $\mathbf{x}_1', \mathbf{x}_2', \mathbf{x}_3'$ are not collinear then clearly there can be no transformation $\mathtt{H}$ taking $\mathbf{x}_i$ to $\mathbf{x}_i'$, since a projective transformation must preserve collinearity. Never-

theless the set of eight homogeneous equations derived from (4.3) must have a non-zero solution, giving rise to a matrix $\mathtt{H}$. How is this apparent contradiction to be resolved?

The equations (4.3) express the condition that $\mathbf{x}'_i \times \mathtt{H}\mathbf{x}_i = \mathbf{0}$ for $i = 1, \ldots, 4$, and so the matrix $\mathtt{H}$ found by solving the system of 8 equations will satisfy this condition. Suppose that $\mathbf{x}_1, \ldots, \mathbf{x}_3$ are collinear and let $\mathbf{l}$ be the line that they lie on, so that $\mathbf{l}^\mathsf{T}\mathbf{x}_i = 0$ for $i = 1, \ldots, 3$. Now define $\mathtt{H}^* = \mathbf{x}'_4\mathbf{l}^\mathsf{T}$, which is a 3×3 matrix of rank 1. In this case, one verifies that $\mathtt{H}^*\mathbf{x}_i = \mathbf{x}'_4(\mathbf{l}^\mathsf{T}\mathbf{x}_i) = \mathbf{0}$ for $i = 1, \ldots, 3$, since $\mathbf{l}^\mathsf{T}\mathbf{x}_i = 0$. On the other hand, $\mathtt{H}^*\mathbf{x}_4 = \mathbf{x}'_4(\mathbf{l}^\mathsf{T}\mathbf{x}_4) = k\mathbf{x}'_4$. Therefore the condition $\mathbf{x}'_i \times \mathtt{H}^*\mathbf{x}_i = \mathbf{0}$ is satisfied for all i. Note that the vector $\mathbf{h}^*$ corresponding to $\mathtt{H}^*$ is given by $\mathbf{h}^{*\mathsf{T}} = (x_4\mathbf{l}^\mathsf{T}, y_4\mathbf{l}^\mathsf{T}, w_4\mathbf{l}^\mathsf{T})$, and one easily verifies that this vector satisfies (4.3) for all i. The problem with this solution for $\mathtt{H}^*$ is that $\mathtt{H}^*$ is a rank 1 matrix and hence does not represent a projective transformation. As a consequence the points $\mathtt{H}^*\mathbf{x}_i = \mathbf{0}$ for $i = 1, \ldots, 3$ are not well defined.

We showed that if $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are collinear then $\mathtt{H}^* = \mathbf{x}'_4\mathbf{l}^\mathsf{T}$ is a solution to (4.1). There are two cases: either $\mathtt{H}^*$ is the unique solution (up to scale) or there is a further solution $\mathtt{H}$. In the first case, since $\mathtt{H}^*$ is a singular matrix, there exists no transformation taking each $\mathbf{x}_i$ to $\mathbf{x}'_i$. This occurs when $\mathbf{x}_1, \ldots, \mathbf{x}_3$ are collinear but $\mathbf{x}'_1, \ldots, \mathbf{x}'_3$ are not. In the second case, where a further solution $\mathtt{H}$ exists, then any matrix of the form $\alpha\mathtt{H}^* + \beta\mathtt{H}$ is a solution. Thus a 2-dimensional family of transformations exist, and it follows that the 8 equations derived from (4.3) are not independent.

A situation where a configuration does not determine a unique solution for a particular class of transformation is termed *degenerate*. Note that the definition of degeneracy involves both the configuration and the type of transformation. The degeneracy problem is not limited to a minimal solution, however. If additional (perfect, i.e. error-free) correspondences are supplied which are also collinear (lie on $\mathbf{l}$), then the degeneracy is not resolved.

4.1.4 Solutions from lines and other entities

The development to this point, and for the rest of the chapter, is exclusively in terms of computing homographies from point correspondences. However, an identical development can be given for computing homographies from line correspondences. Starting from the line transformation $\mathbf{l}_i = \mathtt{H}^\mathsf{T}\mathbf{l}'_i$, a matrix equation of the form $\mathtt{A}\mathbf{h} = \mathbf{0}$ can be derived, with a minimal solution requiring four lines in general position. Similarly, a homography may be computed from conic correspondences and so forth.

There is the question then of how many correspondences are required to compute the homography (or any other relation). The general rule is that the number of constraints must equal or exceed the number of degrees of freedom of the transformation. For example, in 2D each corresponding point or line generates two constraints on $\mathtt{H}$, in 3D each corresponding point or plane generates three constraints. Thus in 2D the correspondence of four points or four lines is sufficient to compute $\mathtt{H}$, since $4 \times 2 = 8$, with 8 the number of degrees of freedom of the homography. In 3D a homography has 15 degrees of freedom, and five points or five planes are required. For a planar affine transformation (6 dof) only three corresponding points or lines are required, and so on. A conic provides five constraints on a 2D homography.

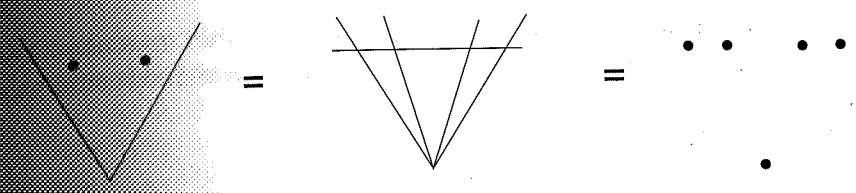


Fig. 4.1. Geometric equivalence of point–line configurations. *A configuration of two points and two lines is equivalent to five lines with four concurrent, or five points with four collinear.*

Care has to be taken when computing H from correspondences of mixed type. For example, a 2D homography cannot be determined uniquely from the correspondences of two points and two lines, but can from three points and one line or one point and three lines, even though in each case the configuration has 8 degrees of freedom. The case of three lines and one point is geometrically equivalent to four points, since the three lines define a triangle and the vertices of the triangle uniquely define three points. We have seen that the correspondence of four points in general position uniquely determines a homography, which means that the correspondence of three lines and one point also uniquely determines a homography. Similarly the case of three points and a line is equivalent to four lines, and again the correspondence of four lines in general position (i.e. no three concurrent) uniquely determines a homography. However, as a quick sketch shows (figure 4.1), the case of two points and two lines is equivalent to five lines with four concurrent, or five points with four collinear. As shown in the previous section, this configuration is degenerate and a one-parameter family of homographies map the two-point and two-line configuration to the corresponding configuration.

4.2 Different cost functions

We will now describe a number of cost functions which may be minimized in order to determine H for over-determined solutions. Methods of minimizing these functions are described later in the chapter.

4.2.1 Algebraic distance

The DLT algorithm minimizes the norm $\|\mathbf{A}\mathbf{h}\|$. The vector $\boldsymbol{\epsilon} = \mathbf{A}\mathbf{h}$ is called the residual vector and it is the norm of this error vector that is minimized. The components of this vector arise from the individual correspondences that generate each row of the matrix $\mathbf{A}$. Each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ contributes a partial error vector $\boldsymbol{\epsilon}_i$ from (4.1) or (4.3) towards the full error vector $\boldsymbol{\epsilon}$. This vector $\boldsymbol{\epsilon}_i$ is the *algebraic error vector* associated with the point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and the homography H. The norm of this vector is a scalar which is called the *algebraic distance*:

$$d_{\text{alg}}(\mathbf{x}'_i, \mathtt{H}\mathbf{x}_i)^2 = \|\boldsymbol{\epsilon}_i\|^2 = \left\| \begin{bmatrix} \mathbf{0}^\mathsf{T} & -w'_i\mathbf{x}_i^\mathsf{T} & y'_i\mathbf{x}_i^\mathsf{T} \\ w'_i\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x'_i\mathbf{x}_i^\mathsf{T} \end{bmatrix} \mathbf{h} \right\|^2 . \tag{4.4}$$

More generally, and briefly, for any two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ we may write

$$d_{\text{alg}}(\mathbf{x}_1, \mathbf{x}_2)^2 = a_1^2 + a_2^2 \text{ where } \mathbf{a} = (a_1, a_2, a_3)^\mathsf{T} = \mathbf{x}_1 \times \mathbf{x}_2 .$$

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 $Z = f$, which is called the *image plane* or *focal plane*. Under the pinhole camera model, a point in space with coordinates $\mathbf{X} = (X, Y, Z)^\mathsf{T}$ 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

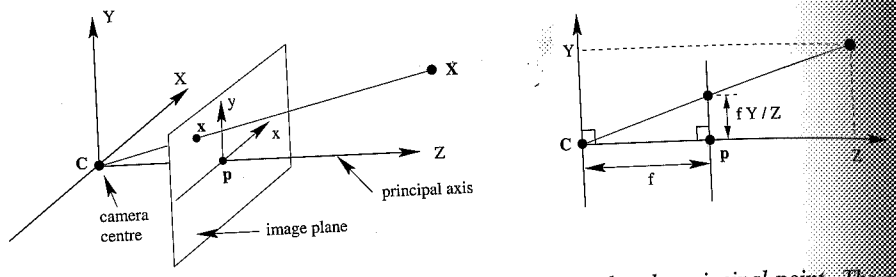


Fig. 6.1. **Pinhole camera geometry.** $\mathbf{C}$ *is the camera centre and* $\mathbf{p}$ *the principal point. The camera centre is here placed at the coordinate origin. Note the image plane is placed in front of the camera centre.*

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 $\mathrm{diag}(f, f, 1)[\mathtt{I} \mid \mathtt{0}]$ where $\mathrm{diag}(f, f, 1)$ is a diagonal matrix and $[\mathtt{I} \mid \mathtt{0}]$ 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

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

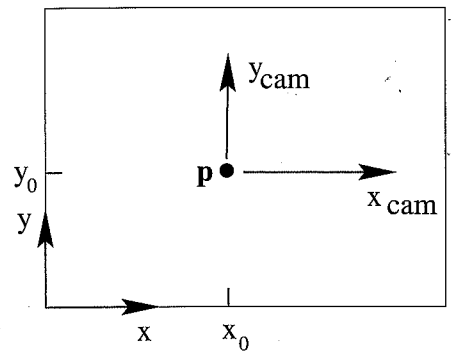


Fig. 6.2. *Image* (x, y) *and camera* $(x_{\text{cam}}, y_{\text{cam}})$ *coordinate systems.*

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

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

where $(p_x, p_y)^\top$ 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

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

The matrix $\mathtt{K}$ is called the *camera calibration matrix*. In (6.5) we have written $(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, 1)^\top$ 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 $\tilde{\mathbf{X}}$ is an inhomogeneous 3-vector representing the coordinates of a point in the world coordinate frame, and $\mathbf{X}_{\text{cam}}$ represents the same point in the camera coordinate frame, then we may write $\mathbf{X}_{\text{cam}} = \mathtt{R}(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$, where $\tilde{\mathbf{C}}$ represents the coordinates of the camera

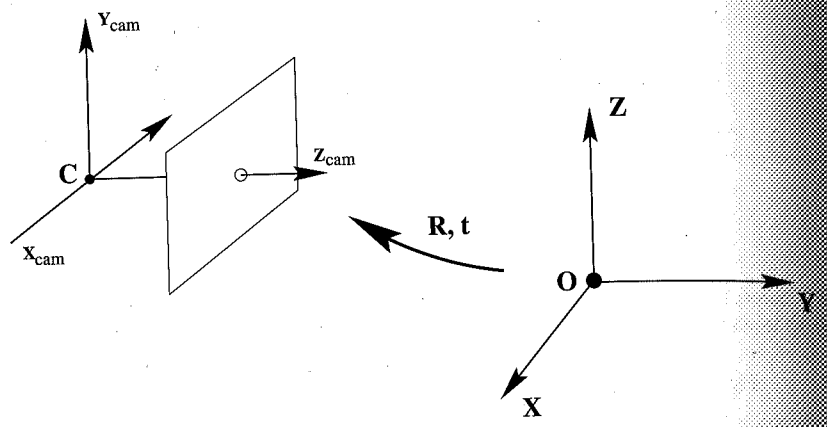


Fig. 6.3. *The Euclidean transformation between the world and camera coordinate frames.*

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

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

Putting this together with (6.5) leads to the formula

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

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, $\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\widetilde{\mathtt{C}}]$, has 9 degrees of freedom: 3 for $\mathtt{K}$ (the elements f, p_x, p_y), 3 for $\mathtt{R}$, and 3 for $\widetilde{\mathtt{C}}$. 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 $\widetilde{\mathtt{C}}$ 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) $\mathbf{t} = -\mathtt{R}\widetilde{\mathtt{C}}$.

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

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

for which the calibration matrix $\mathtt{K}$ 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 $\mathtt{P}$, equal to $\mathtt{KR}$, is non-singular. Conversely, any 3×4 matrix $\mathtt{P}$ for which the left hand 3×3 submatrix is non-singular is the camera matrix of some finite projective camera, because $\mathtt{P}$ can be decomposed as $\mathtt{P} = \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}]$. Indeed, letting $\mathtt{M}$ be the left 3×3 submatrix of $\mathtt{P}$, one decomposes $\mathtt{M}$ as a product $\mathtt{M} = \mathtt{KR}$ where $\mathtt{K}$ is upper-triangular of the form (6.10) and $\mathtt{R}$ is a rotation matrix. This decomposition is essentially the RQ matrix decomposition, described in section A4.1.1(p579), of which more will be said in section 6.2.4. The matrix $\mathtt{P}$ can therefore be written $\mathtt{P} = \mathtt{M}[\mathtt{I} \mid \mathtt{M}^{-1}\mathbf{p}_4] = \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}]$ where $\mathbf{p}_4$ is the last column of $\mathtt{P}$. 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 $\mathtt{X}$, $\mathtt{Y}$ and $\mathtt{Z}$ 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

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

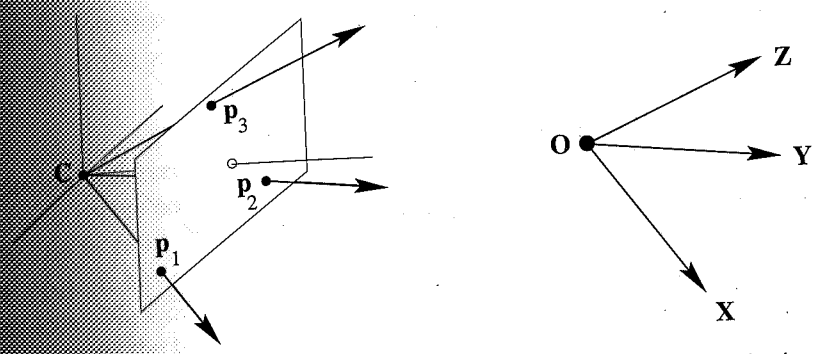


Fig. 6.4. *The three image points defined by the columns* $\mathbf{p}_i, i = 1, \ldots, 3$, *of the projection matrix are the vanishing points of the directions of the world axes.*

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 $\mathbf{C} = (\tilde{\mathbf{C}}^\mathsf{T}, 1)^\mathsf{T}$ is clearly the null-vector of $\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\tilde{\mathbf{C}}]$. 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

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

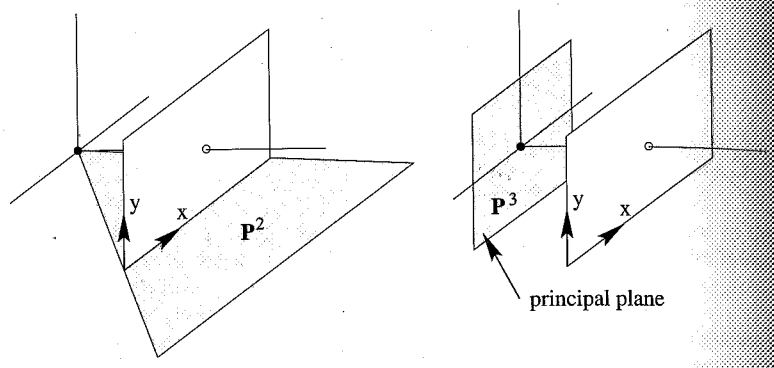


Fig. 6.5. *Two of the three planes defined by the rows of the projection matrix.*

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 $\mathtt{P}\mathbf{P}^3$. 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 $\mathtt{P} = k\mathtt{K}[\mathtt{R} \mid -\mathtt{R}\tilde{\mathbf{C}}] = [\mathtt{M} \mid \mathbf{p}_4]$, 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

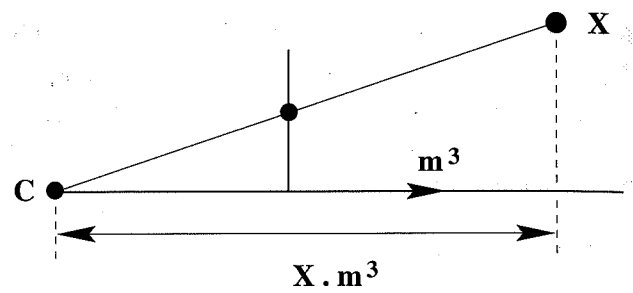


Fig. 6.6. *If the camera matrix* $\mathtt{P} = [\mathtt{M} \mid \mathbf{p}_4]$ *is normalized so that* $\|\mathbf{m}^3\| = 1$ *and* $\det \mathtt{M} > 0$, *and* $\mathbf{x} = w(x, y, 1)^\mathsf{T} = \mathtt{P}\mathbf{X}$, *where* $\mathbf{X} = (\mathtt{X}, \mathtt{Y}, \mathtt{Z}, 1)^\mathsf{T}$, *then*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

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

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))

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

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 $\mathtt{P} = [\mathtt{M} \mid -\mathtt{M}\widetilde{\mathtt{C}}]$, has centre $\widetilde{\mathtt{C}} = (1000.0, 2000.0, 1500.0)^{\mathsf{T}}$, and the matrix 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 P, then the camera representing the picture of a picture is HP, where H is the homography matrix. Since H is non-singular, the left 3×3 submatrix of HP is non-singular and can be decomposed as the product KR – and K need not have $s = 0$. Note however that the K and 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 H is non-singular, $\mathtt{HPC} = \mathbf{0}$ if and only if $\mathtt{PC} = \mathbf{0}$.

Where is the decomposition required? If the camera 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(p154). 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(p154). 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 π_∞) 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.

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(p88) 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:

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

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 $(p_{31}, p_{32}, p_{33})^\top$, 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(p91).

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(p109) 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(p197) 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(p109):

(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