

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.

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}^T . In accordance with this convention, a point in the plane will be represented by the column vector $(x, y)^T$, rather than its transpose, the row vector (x, y) . We write $\mathbf{x} = (x, y)^T$, 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 $(a, b, c)^T$. The correspondence between lines and vectors $(a, b, c)^T$ 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 $(a, b, c)^T$ and $k(a, b, c)^T$ 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 $(a, b, c)^T$ is a representative of the equivalence class. The set of equivalence classes of vectors in $\mathbb{R}^3 - (0, 0, 0)^T$ forms the *projective space* \mathbb{P}^2 . The notation $-(0, 0, 0)^T$ indicates that the vector $(0, 0, 0)^T$, which does not correspond to any line, is excluded.

Homogeneous representation of points. A point $\mathbf{x} = (x, y)^T$ lies on the line $\mathbf{l} = (a, b, c)^T$ if and only if $ax + by + c = 0$. This may be written in terms of an inner product of vectors representing the point as $(x, y, 1)(a, b, c)^T = (x, y, 1)\mathbf{l} = 0$; that is the point $(x, y)^T$ 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 $(kx, ky, k)^T$ for varying values of k to be a representation of the point $(x, y)^T$ 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 $\mathbf{x} = (x_1, x_2, x_3)^T$, representing the point $(x_1/x_3, x_2/x_3)^T$ 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 $\mathbf{x}^T\mathbf{l} = 0$.*

Note that the expression $\mathbf{x}^T\mathbf{l}$ is just the inner or scalar product of the two vectors \mathbf{l} and \mathbf{x} . The scalar product $\mathbf{x}^T\mathbf{l} = \mathbf{l}^T\mathbf{x} = \mathbf{x} \cdot \mathbf{l}$. In general, the transpose notation $\mathbf{l}^T\mathbf{x}$ will be preferred, but occasionally, we will use a \cdot to denote the inner product. We distinguish between the *homogeneous coordinates* $\mathbf{x} = (x_1, x_2, x_3)^T$ of a point, which is a 3-vector, and the *inhomogeneous coordinates* $(x, y)^T$, 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 $\mathbf{l} = (a, b, c)^T$ and $\mathbf{l}' = (a', b', c')^T$, 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 $\mathbf{l}^T\mathbf{x} = \mathbf{l}'^T\mathbf{x} = 0$. 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 $\mathbf{l} = (-1, 0, 1)^T$. The line $y = 1$ is equivalent to $-1y + 1 = 0$, and thus has homogeneous representation $\mathbf{l}' = (0, -1, 1)^T$. 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 $(1, 1)^T$ 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 $\mathbf{l} = (a, b, c)^\top$ and $\mathbf{l}' = (a, b, c')^\top$ 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 $\mathbf{l} \times \mathbf{l}' = (c' - c)(b, -a, 0)^\top$, and ignoring the scale factor $(c' - c)$, this is the point $(b, -a, 0)^\top$.

Now if we attempt to find the inhomogeneous representation of this point, we obtain $(b/0, -a/0)^\top$, which makes no sense, except to suggest that the point of intersection has infinitely large coordinates. In general, points with homogeneous coordinates $(x, y, 0)^\top$ 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 $\mathbf{l} = (-1, 0, 1)^\top$, $\mathbf{l}' = (-1, 0, 2)^\top$, 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. \triangle

Ideal points and the line at infinity. Homogeneous vectors $\mathbf{x} = (x_1, x_2, x_3)^\top$ 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 $(x_1, x_2, 0)^\top$, 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 $\mathbf{l}_\infty = (0, 0, 1)^\top$. Indeed, one verifies that $(0, 0, 1)(x_1, x_2, 0)^\top = 0$.

Using result 2.2 one finds that a line $\mathbf{l} = (a, b, c)^\top$ intersects \mathbf{l}_∞ in the ideal point $(b, -a, 0)^\top$ (since $(b, -a, 0)\mathbf{l} = 0$). A line $\mathbf{l}' = (a, b, c')^\top$ *parallel* to \mathbf{l} intersects \mathbf{l}_∞ in the same ideal point $(b, -a, 0)^\top$ irrespective of the value of c' . In inhomogeneous notation $(b, -a)^\top$ 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 $(b, -a, 0)^\top$ varies over \mathbf{l}_∞ . 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

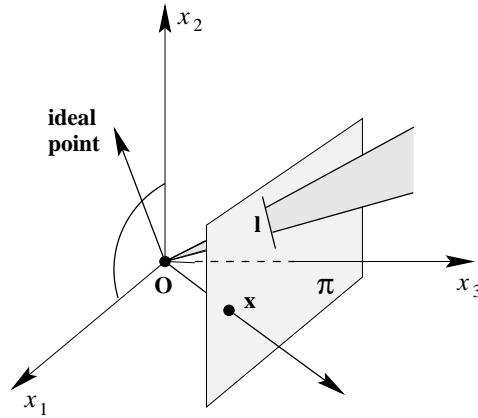


Fig. 2.1. **A model of the projective plane.** Points and lines of \mathbb{P}^2 are represented by rays and planes, respectively, through the origin in \mathbb{R}^3 . Lines lying in the x_1x_2 -plane represent ideal points, and the x_1x_2 -plane represents l_∞ .

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 $k(x_1, x_2, x_3)^T$ 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 l_∞ 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 $l^T x = 0$ for line and point is symmetric, since $l^T x = 0$ implies $x^T l = 0$, 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 it is 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

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0 \quad (2.1)$$

or in matrix form

$$\mathbf{x}^T \mathbf{C} \mathbf{x} = 0 \quad (2.2)$$

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

$$\mathbf{C} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}. \quad (2.3)$$

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 \mathbf{C} by a non-zero scalar does not affect the above equations. Thus \mathbf{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

$$ax_i^2 + bx_iy_i + cy_i^2 + dx_i + ey_i + f = 0.$$

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

Stacking the constraints from five points we obtain

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

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 C at a point \mathbf{x} on C is given by $\mathbf{l} = C\mathbf{x}$.*

Proof. The line $\mathbf{l} = C\mathbf{x}$ passes through \mathbf{x} , since $\mathbf{l}^\top \mathbf{x} = \mathbf{x}^\top C\mathbf{x} = 0$. 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 $\mathbf{y}^\top C\mathbf{y} = 0$ and $\mathbf{x}^\top C\mathbf{y} = \mathbf{l}^\top \mathbf{y} = 0$. From this it follows that $(\mathbf{x} + \alpha\mathbf{y})^\top C(\mathbf{x} + \alpha\mathbf{y}) = 0$ for all α , which means that the whole line $\mathbf{l} = C\mathbf{x}$ joining \mathbf{x} and \mathbf{y} lies on the conic C , which is therefore degenerate (see below). \square

Dual conics. The conic 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 C^* . A line \mathbf{l} tangent to the conic C satisfies $\mathbf{l}^\top C^* \mathbf{l} = 0$. The notation C^* indicates that C^* is the adjoint matrix of C (the adjoint is defined in section A4.2(p580) of appendix 4(p578)). For a non-singular symmetric matrix $C^* = C^{-1}$ (up to scale).

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

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

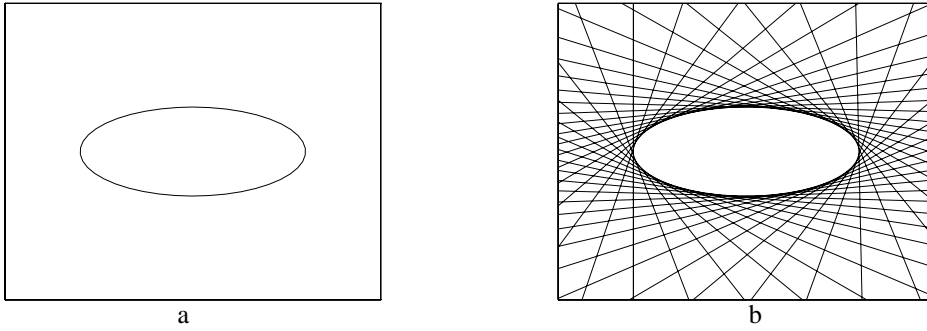


Fig. 2.2. (a) Points \mathbf{x} satisfying $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$ lie on a point conic. (b) Lines \mathbf{l} satisfying $\mathbf{l}^T \mathbf{C}^* \mathbf{l} = 0$ are tangent to the point conic \mathbf{C} . The conic \mathbf{C} is the envelope of the lines \mathbf{l} .

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 \mathbf{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

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

is composed of two lines \mathbf{l} and \mathbf{m} . Points on \mathbf{l} satisfy $\mathbf{l}^T \mathbf{x} = 0$, and are on the conic since $\mathbf{x}^T \mathbf{C} \mathbf{x} = (\mathbf{x}^T \mathbf{l})(\mathbf{m}^T \mathbf{x}) + (\mathbf{x}^T \mathbf{m})(\mathbf{l}^T \mathbf{x}) = 0$. Similarly, points satisfying $\mathbf{m}^T \mathbf{x} = 0$ also satisfy $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$. The matrix \mathbf{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} . \triangle

Degenerate *line* conics include two points (rank 2), and a repeated point (rank 1). For example, the line conic $\mathbf{C}^* = \mathbf{x} \mathbf{y}^T + \mathbf{y} \mathbf{x}^T$ 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 $(\mathbf{C}^*)^* \neq \mathbf{C}$.

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

To interpret this theorem, any point in \mathbb{P}^2 is represented as a homogeneous 3-vector, \mathbf{x} , and $H\mathbf{x}$ 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 l . Thus $l^T \mathbf{x}_i = 0$ for $i = 1, \dots, 3$. Let H be a non-singular 3×3 matrix. One verifies that $l^T H^{-1} H \mathbf{x}_i = 0$. Thus, the points $H\mathbf{x}_i$ all lie on the line $H^{-T} l$, and collinearity is preserved by the transformation.

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

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:

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

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

Note that the matrix 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 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 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 .

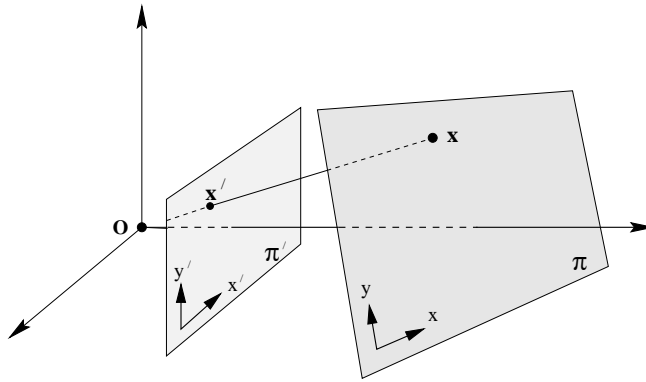


Fig. 2.3. **Central projection maps points on one plane to points on another plane.** The projection also maps lines to lines as may be seen by considering a plane through the projection centre which intersects with the two planes π and π' . Since lines are mapped to lines, central projection is a projectivity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

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 $\mathbf{x}' = H\mathbf{x}$ where 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(p632).

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-

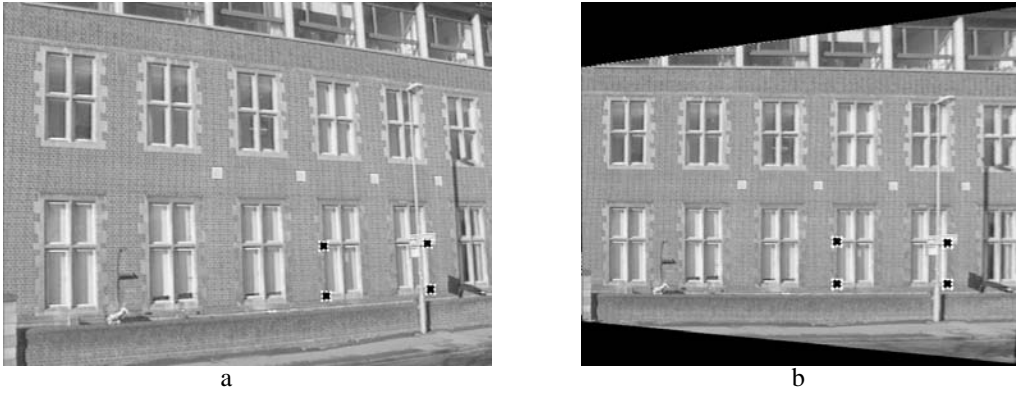


Fig. 2.4. **Removing perspective distortion.** (a) The original image with perspective distortion – the lines of the windows clearly converge at a finite point. (b) Synthesized frontal orthogonal view of the front wall. The image (a) of the wall is related via a projective transformation to the true 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

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

Each point correspondence generates two equations for the elements of 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 H . Four point correspondences lead to eight such linear equations in the entries of H , which are sufficient to solve for 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 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. \triangle

Three remarks concerning this example are appropriate: first, the computation of the rectifying transformation 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

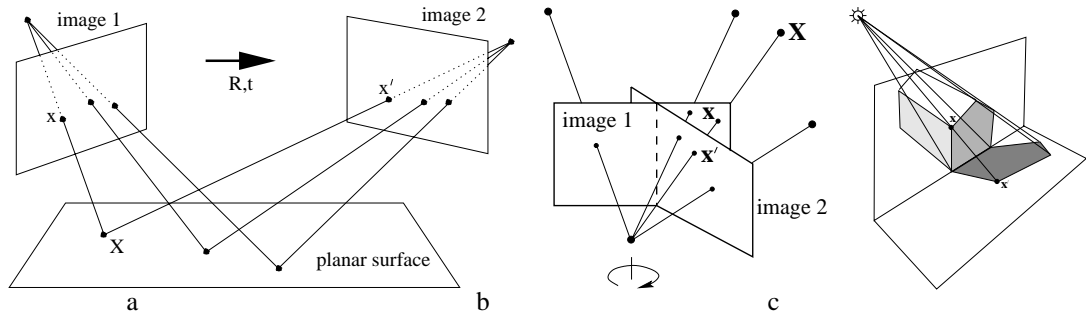


Fig. 2.5. **Examples of a projective transformation, $\mathbf{x}' = \mathbf{H}\mathbf{x}$, arising in perspective images.** (a) The projective transformation between two images induced by a world plane (the concatenation of two projective transformations is a projective transformation); (b) The projective 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 $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$ under a projective transformation lie on the line $\mathbf{l}' = \mathbf{H}^{-\mathbf{T}}\mathbf{l}$. In this way, incidence of points on lines is preserved, since $\mathbf{l}'^{\mathbf{T}}\mathbf{x}'_i = \mathbf{l}^{\mathbf{T}}\mathbf{H}^{-1}\mathbf{H}\mathbf{x}_i = 0$. This gives the transformation rule for lines:

Under the point transformation $\mathbf{x}' = \mathbf{H}\mathbf{x}$, a line transforms as

$$\mathbf{l}' = \mathbf{H}^{-\mathbf{T}}\mathbf{l}. \quad (2.6)$$

One may alternatively write $\mathbf{l}'^{\mathbf{T}} = \mathbf{l}^{\mathbf{T}}\mathbf{H}^{-1}$. Note the fundamentally different way in which lines and points transform. Points transform according to \mathbf{H} , whereas lines (as rows) transform according to \mathbf{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 $\mathbf{x}' = \mathbf{H}\mathbf{x}$, (2.2) becomes

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

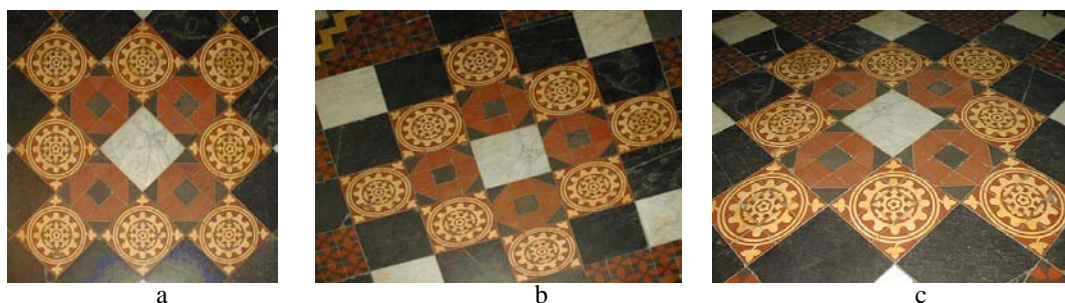


Fig. 2.6. **Distortions arising under central projection.** Images of a tiled floor. (a) **Similarity:** the circular pattern is imaged as a circle. A square tile is imaged as a square. Lines which are parallel or perpendicular have the same relative orientation in the image. (b) **Affine:** The circle 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 $\mathbf{x}'^T \mathbf{C}' \mathbf{x}'$ with $\mathbf{C}' = \mathbf{H}^{-T} \mathbf{C} \mathbf{H}^{-1}$. This gives the transformation rule for a conic:

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

The presence of \mathbf{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 $\mathbf{x}' = \mathbf{H}\mathbf{x}$, a dual conic \mathbf{C}^* transforms to $\mathbf{C}^{*'} = \mathbf{H} \mathbf{C}^* \mathbf{H}^T$.

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(p632).

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 $\text{diag}(-1, 1, 1)$, 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

$$\mathbf{x}' = \mathbf{H}_E \mathbf{x} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x} \quad (2.7)$$

where \mathbf{R} is a 2×2 rotation matrix (an orthogonal matrix such that $\mathbf{R}^\top \mathbf{R} = \mathbf{R} \mathbf{R}^\top = \mathbf{I}$), \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 $\mathbf{R} = \mathbf{I}$). 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

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

This can be written more concisely in block form as

$$\mathbf{x}' = \mathbf{H}_s \mathbf{x} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x} \quad (2.9)$$

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

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

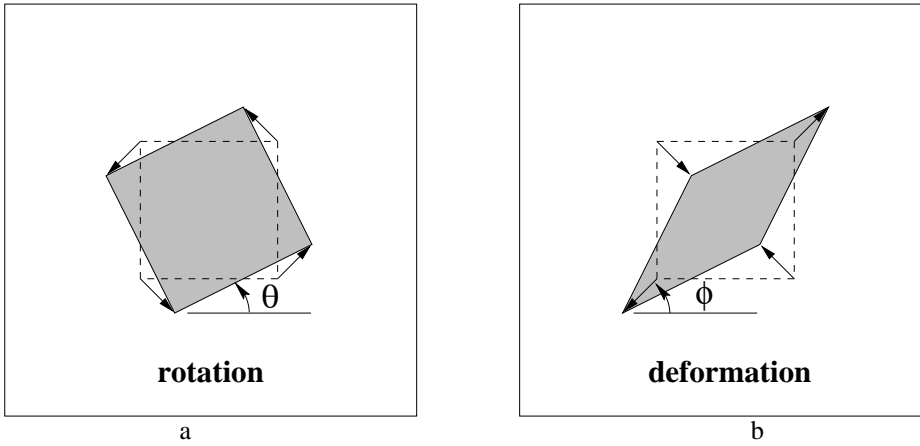


Fig. 2.7. **Distortions arising from a planar affine transformation.** (a) Rotation by $R(\theta)$. (b) A deformation $R(-\phi)D R(\phi)$. Note, the scaling directions in the deformation are orthogonal.

or in block form

$$\mathbf{x}' = H_A \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \quad (2.11)$$

with \mathbf{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 \mathbf{A} of an affine transformation is as the composition of two fundamental transformations, namely rotations and non-isotropic scalings. The affine matrix \mathbf{A} can always be decomposed as

$$\mathbf{A} = R(\theta) R(-\phi) D R(\phi) \quad (2.12)$$

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

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

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

The affine matrix \mathbf{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 $(x_1, x_2, 0)^T$ 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 $\det A$. Thus the area of any shape is scaled by $\det A$, 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 $\det A$ is positive or negative respectively. Since $\det A = \lambda_1 \lambda_2$ 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

$$\mathbf{x}' = H_P \mathbf{x} = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \mathbf{x} \quad (2.13)$$

where the vector $\mathbf{v} = (v_1, v_2)^T$. 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 $\det A$ 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 $(x_1, x_2, 0)^T$ under an affinity and projectivity: First the affine transformation

$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix}. \quad (2.14)$$

Second the projective transformation

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

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.

$$\mathbf{H} = \mathbf{H}_S \mathbf{H}_A \mathbf{H}_P = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^T & v \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \quad (2.16)$$

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

Each of the matrices H_S, H_A, 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: H_P (2 dof) moves the line at infinity; H_A (2 dof) affects the affine properties, but does not move the line at infinity; and finally, H_S is a general similarity transformation (4 dof) which does not affect the affine or projective properties. The transformation H_P is an *elation*, described in section A7.3(p631).

Example 2.15. The projective transformation

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

may be decomposed as

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

△

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 H in (2.16) gives $H^{-1} = H_P^{-1} H_A^{-1} H_S^{-1}$. Since H_P^{-1}, H_A^{-1} and H_S^{-1} are still projective, affine and similarity transformations respectively, a general projective transformation may also be decomposed in the form

$$H = H_P H_A H_S = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{v}^T & 1 \end{bmatrix} \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.17)$$

Note that the actual values of K, 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.*


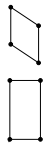


| Group | Matrix | Distortion | Invariant properties |
|---------------------|--|---|--|
| Projective 8 dof | $\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$ |  | Concurrency, collinearity, order of contact : intersection (1 pt contact); 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, l_∞ . |
| 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 $\mathbf{A} = [a_{ij}]$ is an invertible 2×2 matrix, $\mathbf{R} = [r_{ij}]$ 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 $(x_1, x_2)^T$, and a point for which $x_2 = 0$ is an ideal point of the line. We will use the notation \bar{x} to represent the 2-vector $(x_1, x_2)^T$. A projective transformation of a line is represented by a 2×2 homogeneous matrix,

$$\bar{x}' = H_{2 \times 2} \bar{x}$$

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.

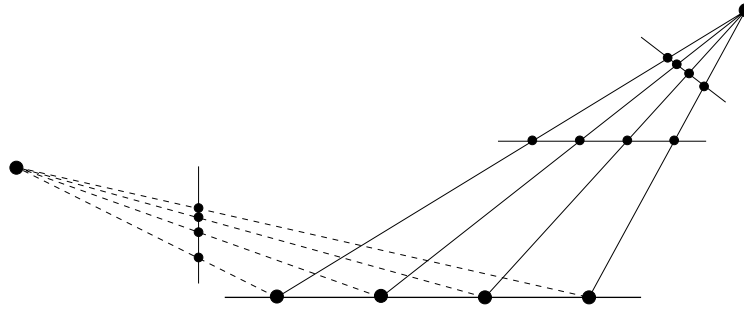


Fig. 2.8. **Projective transformations between lines.** There are four sets of four collinear points in this figure. Each set is related to the others by a line-to-line projectivity. Since the cross ratio is an invariant under a projectivity, the cross ratio has the same value for all the sets shown.

The cross ratio. The cross ratio is the basic projective invariant of \mathbb{P}^1 . Given 4 points \bar{x}_i the *cross ratio* is defined as

$$\text{Cross}(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4) = \frac{|\bar{x}_1 \bar{x}_2| |\bar{x}_3 \bar{x}_4|}{|\bar{x}_1 \bar{x}_3| |\bar{x}_2 \bar{x}_4|}$$

where

$$|\bar{x}_i \bar{x}_j| = \det \begin{bmatrix} x_{i1} & x_{j1} \\ x_{i2} & x_{j2} \end{bmatrix}.$$

A few comments on the cross ratio:

- (i) The value of the cross ratio is not dependent on which particular homogeneous representative of a point \bar{x}_i is used, since the scale cancels between numerator and denominator.
- (ii) If each point \bar{x}_i is a finite point and the homogeneous representative is chosen such that $x_2 = 1$, then $|\bar{x}_i \bar{x}_j|$ represents the signed distance from \bar{x}_i to \bar{x}_j .
- (iii) The definition of the cross ratio is also valid if one of the points \bar{x}_i is an ideal point.
- (iv) The value of the cross ratio is invariant under any projective transformation of the line: if $\bar{x}' = H_{2 \times 2} \bar{x}$ then

$$\text{Cross}(\bar{x}'_1, \bar{x}'_2, \bar{x}'_3, \bar{x}'_4) = \text{Cross}(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4). \quad (2.18)$$

The proof is left as an exercise. Equivalently stated, the cross ratio is invariant to the projective coordinate frame chosen for the line.

Figure 2.8 illustrates a number of projective transformations between lines with equivalent cross ratios.

Under a projective transformation of the plane, a 1D projective transformation is induced on any line in the plane.

Concurrent lines. A configuration of concurrent lines is dual to collinear points on a line. This means that concurrent lines on a plane also have the geometry \mathbb{P}^1 . In particular four concurrent lines have a cross ratio as illustrated in figure 2.9a.

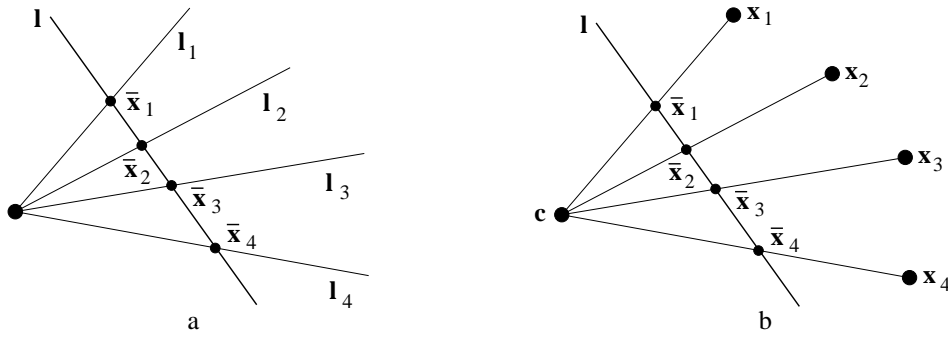


Fig. 2.9. **Concurrent lines.** (a) Four concurrent lines l_i intersect the line l in the four points \bar{x}_i . The cross ratio of these lines is an invariant to projective transformations of the plane. Its value is given by the cross ratio of the points, $\text{Cross}(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4)$. (b) Coplanar points x_i are imaged onto a line l (also in the plane) by a projection with centre c . The cross ratio of the image points \bar{x}_i is invariant to the position of the image line l .

Note how figure 2.9b may be thought of as representing projection of points in \mathbb{P}^2 into a 1-dimensional image. In particular, if c represents a camera centre, and the line l represents an image line (1D analogue of the image plane), then the points \bar{x}_i are the projections of points x_i into the image. The cross ratio of the points \bar{x}_i characterizes the projective configuration of the four image points. Note that the actual position of the image line is irrelevant as far as the projective configuration of the four image points is concerned – different choices of image line give rise to projectively equivalent configurations of image points.

The projective geometry of concurrent lines is important to the understanding of the projective geometry of epipolar lines in chapter 9.

2.6 Topology of the projective plane

We make brief mention of the topology of \mathbb{P}^2 . Understanding of this section is not required for following the rest of the book.

We have seen that the projective plane \mathbb{P}^2 may be thought of as the set of all homogeneous 3-vectors. A vector of this type $\mathbf{x} = (x_1, x_2, x_3)^T$ may be normalized by multiplication by a non-zero factor so that $x_1^2 + x_2^2 + x_3^2 = 1$. Such a point lies on the unit sphere in \mathbb{R}^3 . However, any vector \mathbf{x} and $-\mathbf{x}$ represent the same point in \mathbb{P}^2 , since they differ by a multiplicative factor, -1 . Thus, there is a two-to-one correspondence between the unit sphere S^2 in \mathbb{R}^3 and the projective plane \mathbb{P}^2 . The projective plane may be pictured as the unit sphere with opposite points identified. In this representation, a line in \mathbb{P}^2 is modelled as a great circle on the unit sphere (as ever, with opposite points identified). One may verify that any two distinct (non-antipodal) points on the sphere lie on exactly one great circle, and any two great circles intersect in one point (since antipodal points are identified).

In the language of topology, the sphere S^2 is a 2-sheeted covering space of \mathbb{P}^2 . This implies that \mathbb{P}^2 is not *simply-connected*, which means that there are loops in \mathbb{P}^2 which cannot be contracted to a point inside \mathbb{P}^2 . To be technical, the fundamental group of \mathbb{P}^2 is the cyclic group of order 2.

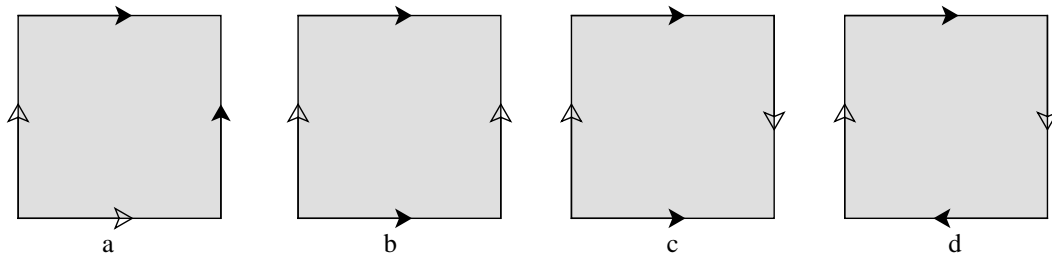


Fig. 2.10. **Topology of surfaces.** Common surfaces may be constructed from a paper square (topologically a disk) with edges glued together. In each case, the matching arrow edges of the square are to be glued together in such a way that the directions of the arrows match. One obtains (a) a sphere, (b) a torus, (c) a Klein bottle and (d) a projective plane. Only the sphere and torus are actually realizable with a real sheet of paper. The sphere and torus are orientable but the projective plane and Klein bottle are not.

In the model for the projective plane as a sphere with opposite points identified one may dispense with the lower hemisphere of S^2 , since points in this hemisphere are the same as the opposite points in the upper hemisphere. In this case, \mathbb{P}^2 may be constructed from the upper hemisphere by identifying opposite points on the equator. Since the upper hemisphere of S^2 is topologically the same as a disk, \mathbb{P}^2 is simply a disk with opposite points on its boundary identified, or glued together. This is not physically possible. Constructing topological spaces by gluing the boundary of a disk is a common method in topology, and in fact any 2-manifold may be constructed in this way. This is illustrated in figure 2.10.

A notable feature of the projective plane \mathbb{P}^2 is that it is non-orientable. This means that it is impossible to define a local orientation (represented for instance by a pair of oriented coordinate axes) that is consistent over the whole surface. This is illustrated in figure 2.11 in which it is shown that the projective plane contains an orientation-reversing path.

The topology of \mathbb{P}^1 . In a similar manner, the 1-dimensional projective line may be identified as a 1-sphere S^1 (that is, a circle) with opposite points identified. If we omit the lower half of the circle, as being duplicated by the top half, then the top half of a circle is topologically equivalent to a line segment. Thus \mathbb{P}^1 is topologically equivalent to a line segment with the two endpoints identified – namely a circle, S^1 .

2.7 Recovery of affine and metric properties from images

We return to the example of projective rectification of example 2.12(p34) where the aim was to remove the projective distortion in the perspective image of a plane to the extent that similarity properties (angles, ratios of lengths) could be measured on the original plane. In that example the projective distortion was completely removed by specifying the position of four reference points on the plane (a total of 8 degrees of freedom), and explicitly computing the transformation mapping the reference points to their images. In fact this overspecifies the geometry – a projective transformation has only 4 degrees of freedom more than a similarity, so it is only necessary to specify 4

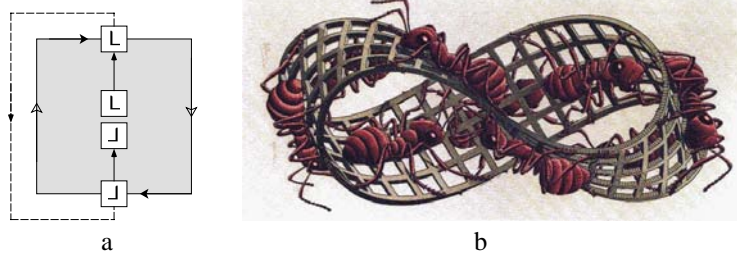


Fig. 2.11. **Orientation of surfaces.** A coordinate frame (represented by an L in the diagram) may be transported along a path in the surface eventually coming back to the point where it started. (a) represents a projective plane. In the path shown, the coordinate frame (represented by a pair of axes) is reversed when it returns to the same point, since the identification at the boundary of the square swaps the direction of one of the axes. Such a path is called an orientation-reversing path, and a surface that contains such a path is called non-orientable. (b) shows the well known example of a Möbius strip obtained by joining two opposite edges of a rectangle (M.C. Escher's "Moebius Strip II [Red Ants]", 1963. ©2000 Cordon Art B.V. – Baarn-Holland. All rights reserved). As can be verified, a path once around the strip is orientation-reversing.

degrees of freedom (not 8) in order to determine metric properties. In projective geometry these 4 degrees of freedom are given “physical substance” by being associated with geometric objects: the line at infinity l_∞ (2 dof), and the two *circular points* (2 dof) on l_∞ . This association is often a more intuitive way of reasoning about the problem than the equivalent description in terms of specifying matrices in the decomposition chain (2.16).

In the following it is shown that the projective distortion may be removed once the image of l_∞ is specified, and the affine distortion removed once the image of the circular points is specified. Then the only remaining distortion is a similarity.

2.7.1 The line at infinity

Under a projective transformation ideal points may be mapped to finite points (2.15), and consequently l_∞ is mapped to a finite line. However, if the transformation is an affinity, then l_∞ is not mapped to a finite line, but remains at infinity. This is evident directly from the line transformation (2.6–p36):

$$l'_\infty = H_A^{-T} l_\infty = \begin{bmatrix} A^{-T} & \mathbf{0} \\ -t^T A^{-T} & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = l_\infty.$$

The converse is also true, i.e. an affine transformation is the most general linear transformation that fixes l_∞ , and may be seen as follows. We require that a point at infinity, say $\mathbf{x} = (1, 0, 0)^T$, be mapped to a point at infinity. This requires that $h_{31} = 0$. Similarly, $h_{32} = 0$, so the transformation is an affinity. To summarize,

Result 2.17. *The line at infinity, l_∞ , is a fixed line under the projective transformation H if and only if H is an affinity.*

However, l_∞ is not fixed pointwise under an affine transformation: (2.14) showed that under an affinity a point on l_∞ (an ideal point) is mapped to a point on l_∞ , but

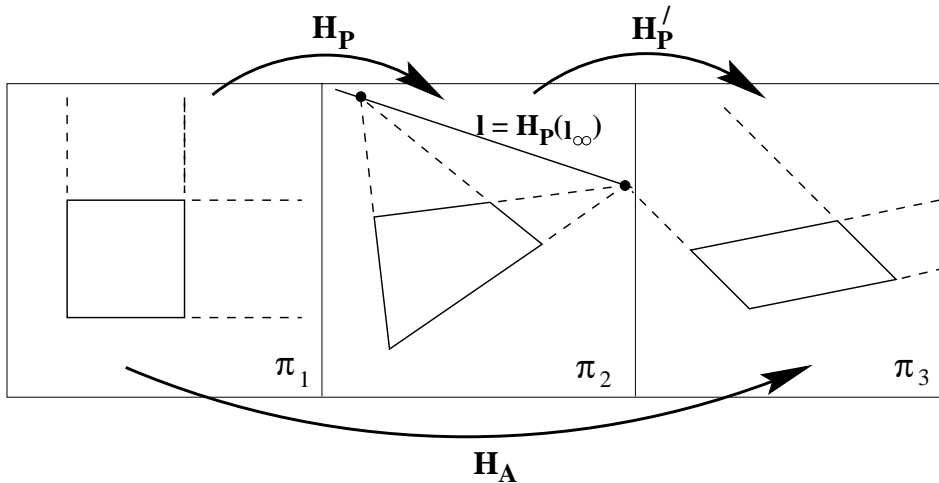


Fig. 2.12. **Affine rectification.** A projective transformation maps l_∞ from $(0, 0, 1)^T$ on a Euclidean plane π_1 to a finite line l on the plane π_2 . If a projective transformation is constructed such that l is mapped back to $(0, 0, 1)^T$ then from result 2.17 the transformation between the first and third planes must be an affine transformation since the canonical position of l_∞ is preserved. This means that affine properties of the first plane can be measured from the third, i.e. the third plane is within an affinity of the first.

it is not the same point unless $A(x_1, x_2)^T = k(x_1, x_2)^T$. It will now be shown that identifying l_∞ allows the recovery of affine properties (parallelism, ratio of areas).

2.7.2 Recovery of affine properties from images

Once the imaged line at infinity is identified in an image of a plane, it is then possible to make affine measurements on the original plane. For example, lines may be identified as parallel on the original plane if the imaged lines intersect on the imaged l_∞ . This follows because parallel lines on the Euclidean plane intersect on l_∞ , and after a projective transformation the lines still intersect on the imaged l_∞ since intersections are preserved by projectivities. Similarly, once l_∞ is identified a length ratio on a line may be computed from the cross ratio of the three points specifying the lengths together with the intersection of the line with l_∞ (which provides the fourth point for the cross ratio), and so forth.

However, a less tortuous path which is better suited to computational algorithms is simply to transform the identified l_∞ to its canonical position of $l_\infty = (0, 0, 1)^T$. The (projective) matrix which achieves this transformation can be applied to every point in the image in order to affinely rectify the image, i.e. after the transformation, affine measurements can be made directly from the rectified image. The key idea here is illustrated in figure 2.12.

If the imaged line at infinity is the line $l = (l_1, l_2, l_3)^T$, then provided $l_3 \neq 0$ a suitable projective point transformation which will map l back to $l_\infty = (0, 0, 1)^T$ is

$$H = H_A \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \quad (2.19)$$

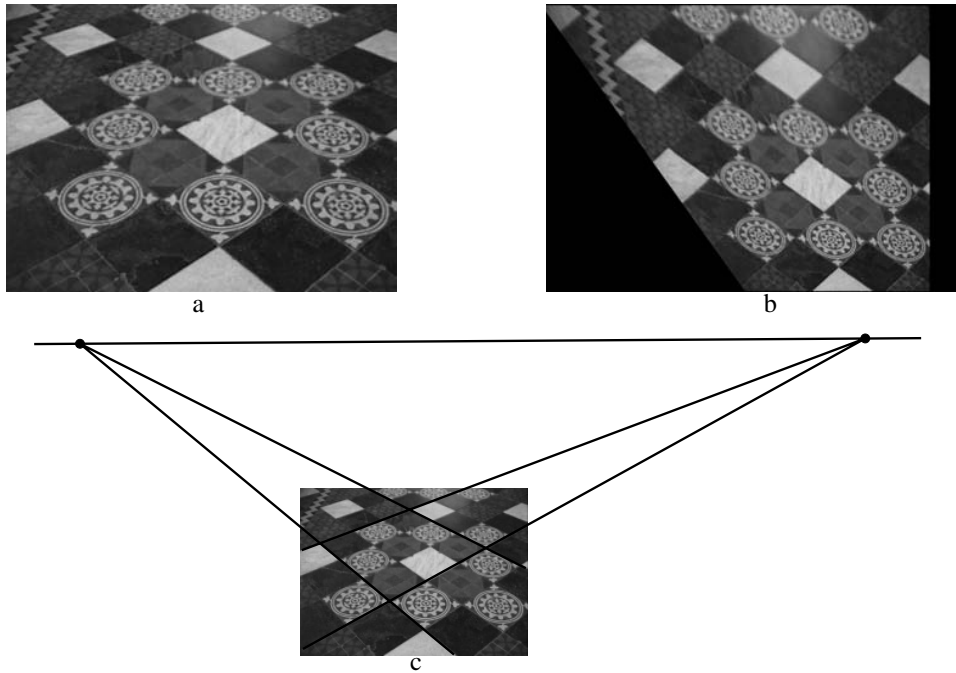


Fig. 2.13. **Affine rectification via the vanishing line.** The vanishing line of the plane imaged in (a) is computed (c) from the intersection of two sets of imaged parallel lines. The image is then projectively warped to produce the affinely rectified image (b). In the affinely rectified image parallel lines are now parallel. However, angles do not have their veridical world value since they are affinely distorted. See also figure 2.17.

where H_A is any affine transformation (the last row of H is \mathbf{l}^T). One can verify that under the line transformation (2.6–p36) $H^{-T}(l_1, l_2, l_3)^T = (0, 0, 1)^T = \mathbf{l}_\infty$.

Example 2.18. Affine rectification

In a perspective image of a plane, the line at infinity on the world plane is imaged as the vanishing line of the plane. This is discussed in more detail in chapter 8. As illustrated in figure 2.13 the vanishing line \mathbf{l} may be computed by intersecting imaged parallel lines. The image is then rectified by applying a projective warping (2.19) such that \mathbf{l} is mapped to its canonical position $\mathbf{l}_\infty = (0, 0, 1)^T$. \triangle

This example shows that affine properties may be recovered by simply specifying a line (2 dof). It is equivalent to specifying only the projective component of the transformation decomposition chain (2.16). Conversely if affine properties are known, these may be used to determine points and the line at infinity. This is illustrated in the following example.

Example 2.19. Computing a vanishing point from a length ratio. Given two intervals on a line with a known length ratio, the point at infinity on the line may be determined. A typical case is where three points \mathbf{a}' , \mathbf{b}' and \mathbf{c}' are identified on a line in an image. Suppose \mathbf{a} , \mathbf{b} and \mathbf{c} are the corresponding collinear points on the world line, and the length ratio $d(\mathbf{a}, \mathbf{b}) : d(\mathbf{b}, \mathbf{c}) = a : b$ is known (where $d(\mathbf{x}, \mathbf{y})$ is the Euclidean

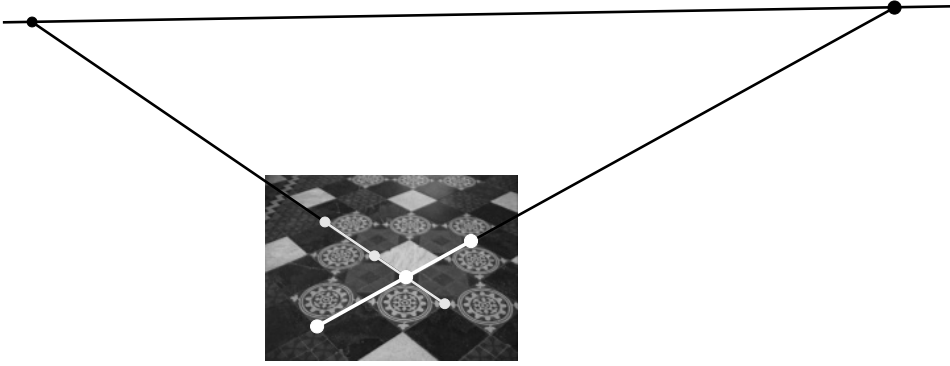


Fig. 2.14. Two examples of using equal length ratios on a line to determine the point at infinity. The line intervals used are shown as the thin and thick white lines delineated by points. This construction determines the vanishing line of the plane. Compare with figure 2.13c.

distance between the points x and y). It is possible to find the vanishing point using the cross ratio. Equivalently, one may proceed as follows:

- (i) Measure the distance ratio in the image, $d(a', b') : d(b', c') = a' : b'$.
- (ii) Points a , b and c may be represented as coordinates 0 , a and $a+b$ in a coordinate frame on the line $\langle a, b, c \rangle$. For computational purposes, these points are represented by homogeneous 2-vectors $(0, 1)^T$, $(a, 1)^T$ and $(a+b, 1)^T$. Similarly, a' , b' and c' have coordinates 0 , a' and $a' + b'$, which may also be expressed as homogeneous vectors.
- (iii) Relative to these coordinate frames, compute the 1D projective transformation $H_{2 \times 2}$ mapping $a \mapsto a'$, $b \mapsto b'$ and $c \mapsto c'$.
- (iv) The image of the point at infinity (with coordinates $(1, 0)^T$) under $H_{2 \times 2}$ is the vanishing point on the line $\langle a', b', c' \rangle$.

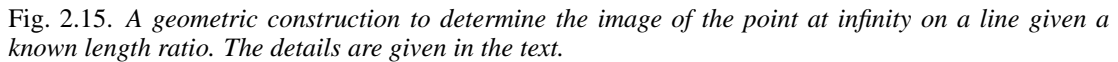
An example of vanishing points computed in this manner is shown in figure 2.14. \triangle

Example 2.20. Geometric construction of vanishing points from a length ratio.

The vanishing points shown in figure 2.14 may also be computed by a purely geometric construction consisting of the following steps:

- (i) Given: three collinear points, a' , b' and c' , in an image corresponding to collinear world points with interval ratio $a : b$.
- (ii) Draw any line l through a' (not coincident with the line $a'c'$), and mark off points $a = a'$, b and c such that the line segments $\langle ab \rangle$, $\langle bc \rangle$ have length ratio $a : b$.
- (iii) Join bb' and cc' and intersect in o .
- (iv) The line through o parallel to l meets the line $a'c'$ in the vanishing point v' .

This construction is illustrated in figure 2.15. \triangle



Under any similarity transformation there are two points on l_∞ which are fixed. These are the *circular points* (also called the *absolute points*) I, J , with canonical coordinates

The circular points are a pair of complex conjugate ideal points. To see that they are fixed under an orientation-preserving similarity:

with an analogous proof for \mathbf{J} . A reflection swaps \mathbf{I} and \mathbf{J} . The converse is also true, i.e. if the circular points are fixed then the linear transformation is a similarity. The proof is left as an exercise. To summarize,

The name “circular points” arises because every circle intersects l_∞ at the circular points. To see this, start from equation (2.1–p30) for a conic. In the case that the conic is a circle: $a = c$ and $b = 0$. Then

Copyright © 2004. Cambridge University Press. All rights reserved.

where a has been set to unity. This conic intersects l_∞ in the (ideal) points for which $x_3 = 0$, namely

$$x_1^2 + x_2^2 = 0$$

with solution $\mathbf{I} = (1, i, 0)^T$, $\mathbf{J} = (1, -i, 0)^T$, i.e. any circle intersects l_∞ in the circular points. In Euclidean geometry it is well known that a circle is specified by three points. The circular points enable an alternative computation. A circle can be computed using the general formula for a conic defined by five points (2.4–p31), where the five points are the three points augmented with the two circular points.

In section 2.7.5 it will be shown that identifying the circular points (or equivalently their dual, see below) allows the recovery of similarity properties (angles, ratios of lengths). Algebraically, the circular points are the orthogonal directions of Euclidean geometry, $(1, 0, 0)^T$ and $(0, 1, 0)^T$, packaged into a single complex conjugate entity, e.g.

$$\mathbf{I} = (1, 0, 0)^T + i(0, 1, 0)^T.$$

Consequently, it is not so surprising that once the circular points are identified, orthogonality, and other metric properties, are then determined.

The conic dual to the circular points. The conic

$$\mathbf{C}_\infty^* = \mathbf{I}\mathbf{I}^T + \mathbf{J}\mathbf{J}^T \quad (2.20)$$

is dual to the circular points. The conic \mathbf{C}_∞^* is a degenerate (rank 2) line conic (see section 2.2.3), which consists of the two circular points. In a Euclidean coordinate system it is given by

$$\mathbf{C}_\infty^* = \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix} \begin{pmatrix} 1 & -i & 0 \end{pmatrix} + \begin{pmatrix} 1 \\ -i \\ 0 \end{pmatrix} \begin{pmatrix} 1 & i & 0 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The conic \mathbf{C}_∞^* is fixed under similarity transformations in an analogous fashion to the fixed properties of circular points. A conic is fixed if the same matrix results (up to scale) under the transformation rule. Since \mathbf{C}_∞^* is a dual conic it transforms according to result 2.14(p37) ($\mathbf{C}^{*'} = \mathbf{H}\mathbf{C}^*\mathbf{H}^T$), and one can verify that under the point transformation $\mathbf{x}' = \mathbf{H}_s\mathbf{x}$,

$$\mathbf{C}_\infty^{*'} = \mathbf{H}_s\mathbf{C}_\infty^*\mathbf{H}_s^T = \mathbf{C}_\infty^*.$$

The converse is also true, and we have

Result 2.22. *The dual conic \mathbf{C}_∞^* is fixed under the projective transformation \mathbf{H} if and only if \mathbf{H} is a similarity.*

Some properties of \mathbf{C}_∞^* in any projective frame:

- (i) \mathbf{C}_∞^* has 4 degrees of freedom: a 3×3 homogeneous symmetric matrix has 5 degrees of freedom, but the constraint $\det \mathbf{C}_\infty^* = 0$ reduces the degrees of freedom by 1.

- (ii) \mathbf{l}_∞ is the null vector of \mathbf{C}_∞^* . This is clear from the definition: the circular points lie on \mathbf{l}_∞ , so that $\mathbf{I}^\top \mathbf{l}_\infty = \mathbf{J}^\top \mathbf{l}_\infty = 0$; then

$$\mathbf{C}_\infty^* \mathbf{l}_\infty = (\mathbf{I}\mathbf{J}^\top + \mathbf{J}\mathbf{I}^\top) \mathbf{l}_\infty = \mathbf{I}(\mathbf{J}^\top \mathbf{l}_\infty) + \mathbf{J}(\mathbf{I}^\top \mathbf{l}_\infty) = \mathbf{0}.$$

2.7.4 Angles on the projective plane

In Euclidean geometry the angle between two lines is computed from the dot product of their normals. For the lines $\mathbf{l} = (l_1, l_2, l_3)^\top$ and $\mathbf{m} = (m_1, m_2, m_3)^\top$ with normals parallel to $(l_1, l_2)^\top, (m_1, m_2)^\top$ respectively, the angle is

$$\cos \theta = \frac{l_1 m_1 + l_2 m_2}{\sqrt{(l_1^2 + l_2^2)(m_1^2 + m_2^2)}}. \quad (2.21)$$

The problem with this expression is that the first two components of \mathbf{l} and \mathbf{m} do not have well defined transformation properties under projective transformations (they are not tensors), and so (2.21) cannot be applied after an affine or projective transformation of the plane. However, an analogous expression to (2.21) which is invariant to projective transformations is

$$\cos \theta = \frac{\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{m}}{\sqrt{(\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{l})(\mathbf{m}^\top \mathbf{C}_\infty^* \mathbf{m})}} \quad (2.22)$$

where \mathbf{C}_∞^* is the conic dual to the circular points. It is clear that in a Euclidean coordinate system (2.22) reduces to (2.21). It may be verified that (2.22) is invariant to projective transformations by using the transformation rules for lines (2.6–p36) ($\mathbf{l}' = \mathbf{H}^{-\top} \mathbf{l}$) and dual conics (result 2.14(p37)) ($\mathbf{C}^{*'} = \mathbf{H} \mathbf{C}^* \mathbf{H}^\top$) under the point transformation $\mathbf{x}' = \mathbf{H} \mathbf{x}$. For example, the numerator transforms as

$$\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{m} \mapsto \mathbf{l}'^\top \mathbf{H}^{-1} \mathbf{H} \mathbf{C}_\infty^* \mathbf{H}^\top \mathbf{H}^{-\top} \mathbf{m} = \mathbf{l}'^\top \mathbf{C}_\infty^{*'} \mathbf{m}.$$

It may also be verified that the scale of the homogeneous objects cancels between the numerator and denominator. Thus (2.22) is indeed invariant to the projective frame. To summarize, we have shown

Result 2.23. *Once the conic \mathbf{C}_∞^* is identified on the projective plane then Euclidean angles may be measured by (2.22).*

Note, as a corollary,

Result 2.24. *Lines \mathbf{l} and \mathbf{m} are orthogonal if $\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{m} = 0$.*

Geometrically, if \mathbf{l} and \mathbf{m} satisfy $\mathbf{l}^\top \mathbf{C}_\infty^* \mathbf{m} = 0$, then the lines are conjugate (see section 2.8.1) with respect to the conic \mathbf{C}_∞^* .

Length ratios may also be measured once \mathbf{C}_∞^* is identified. Consider the triangle shown in figure 2.16 with vertices $\mathbf{a}, \mathbf{b}, \mathbf{c}$. From the standard trigonometric sine rule the ratio of lengths $d(\mathbf{b}, \mathbf{c}) : d(\mathbf{a}, \mathbf{c}) = \sin \alpha : \sin \beta$, where $d(\mathbf{x}, \mathbf{y})$ denotes the Euclidean distance between the points \mathbf{x} and \mathbf{y} . Using (2.22), both $\cos \alpha$ and $\cos \beta$ may be computed from the lines $\mathbf{l}' = \mathbf{a}' \times \mathbf{b}'$, $\mathbf{m}' = \mathbf{c}' \times \mathbf{a}'$ and $\mathbf{n}' = \mathbf{b}' \times \mathbf{c}'$ for any

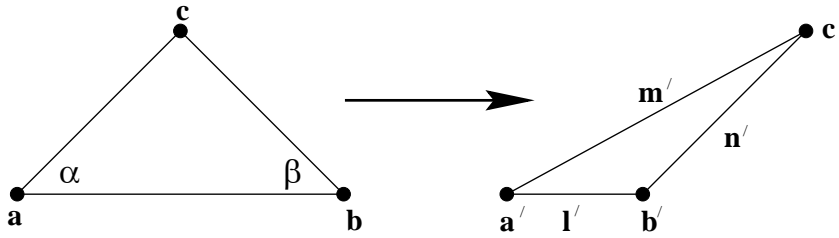


Fig. 2.16. **Length ratios.** Once C_{∞}^* is identified the Euclidean length ratio $d(b, c) : d(a, c)$ may be measured from the projectively distorted figure. See text for details.

projective frame in which C_{∞}^* is specified. Consequently both $\sin \alpha$, $\sin \beta$, and thence the ratio $d(a, b) : d(c, a)$, may be determined from the projectively mapped points.

2.7.5 Recovery of metric properties from images

A completely analogous approach to that of section 2.7.2 and figure 2.12, where affine properties are recovered by specifying l_{∞} , enables metric properties to be recovered from an image of a plane by transforming the circular points to their canonical position. Suppose the circular points are identified in an image, and the image is then rectified by a projective transformation H that maps the imaged circular points to their canonical position (at $(1, \pm i, 0)^T$) on l_{∞} . From result 2.21 the transformation between the world plane and the rectified image is then a similarity since it is projective and the circular points are fixed.

Metric rectification using C_{∞}^* . The dual conic C_{∞}^* neatly packages all the information required for a metric rectification. It enables both the projective and affine components of a projective transformation to be determined, leaving only similarity distortions. This is evident from its transformation under a projectivity. If the point transformation is $x' = Hx$, where the x -coordinate frame is Euclidean and x' projective, C_{∞}^* transforms according to result 2.14(p37) ($C_{\infty}^{*'} = HC^*H^T$). Using the decomposition chain (2.17–p43) for H

$$\begin{aligned} C_{\infty}^{*'} &= (H_P H_A H_S) C_{\infty}^* (H_P H_A H_S)^T = (H_P H_A) (H_S C_{\infty}^* H_S^T) (H_A^T H_P^T) \\ &= (H_P H_A) C_{\infty}^* (H_A^T H_P^T) \\ &= \begin{bmatrix} KK^T & KK^T \mathbf{v} \\ \mathbf{v}^T KK^T & \mathbf{v}^T KK^T \mathbf{v} \end{bmatrix}. \end{aligned} \quad (2.23)$$

It is clear that the projective (\mathbf{v}) and affine (K) components are determined directly from the image of C_{∞}^* , but (since C_{∞}^* is invariant to similarity transformation by result 2.22) the similarity component is undetermined. Consequently,

Result 2.25. *Once the conic C_{∞}^* is identified on the projective plane then projective distortion may be rectified up to a similarity.*

Actually, a suitable rectifying homography may be obtained directly from the identified C_{∞}^{*} in an image using the SVD (section A4.4(p585)): writing the SVD of C_{∞}^{*}

as

$$C_{\infty}^* = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T$$

then by inspection from (2.23) the rectifying projectivity is $H = U$ up to a similarity.

The following two examples show typical situations where C_{∞}^* may be identified in an image, and thence a metric rectification obtained.

Example 2.26. Metric rectification I

Suppose an image has been affinely rectified (as in example 2.18 above), then we require two constraints to specify the 2 degrees of freedom of the circular points in order to determine a metric rectification. These two constraints may be obtained from two imaged right angles on the world plane.

Suppose the lines l', m' in the affinely rectified image correspond to an orthogonal line pair l, m on the world plane. From result 2.24 $l'^T C_{\infty}^* m' = 0$, and using (2.23) with $v = 0$

$$\begin{pmatrix} l'_1 & l'_2 & l'_3 \end{pmatrix} \begin{bmatrix} KK^T & 0 \\ 0^T & 0 \end{bmatrix} \begin{pmatrix} m'_1 \\ m'_2 \\ m'_3 \end{pmatrix} = 0$$

which is a *linear* constraint on the 2×2 matrix $S = KK^T$. The matrix $S = KK^T$ is symmetric with three independent elements, and thus 2 degrees of freedom (as the overall scaling is unimportant). The orthogonality condition reduces to the equation $(l'_1, l'_2)S(m'_1, m'_2)^T = 0$ which may be written as

$$(l'_1 m'_1, l'_1 m'_2 + l'_2 m'_1, l'_2 m'_2) s = 0,$$

where $s = (s_{11}, s_{12}, s_{22})^T$ is S written as a 3-vector. Two such orthogonal line pairs provide two constraints which may be stacked to give a 2×3 matrix with s determined as the null vector. Thus S , and hence K , is obtained up to scale (by Cholesky decomposition, section A4.2.1(p582)). Figure 2.17 shows an example of two orthogonal line pairs being used to metrically rectify the affinely rectified image computed in figure 2.13. \triangle

Alternatively, the two constraints required for metric rectification may be obtained from an imaged circle or two known length ratios. In the case of a circle, the image conic is an ellipse in the affinely rectified image, and the intersection of this ellipse with the (known) l_{∞} directly determines the imaged circular points.

The conic C_{∞}^* can alternatively be identified directly in a perspective image, without first identifying l_{∞} , as is illustrated in the following example.

Example 2.27. Metric rectification II

We start here from the original perspective image of the plane (not the affinely rectified image of example 2.26). Suppose lines l and m are images of orthogonal lines on the world plane; then from result 2.24 $l^T C_{\infty}^* m = 0$, and in a similar manner to constraining

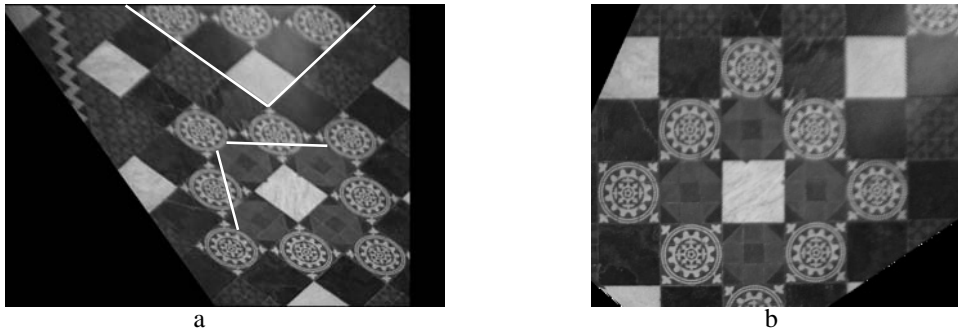


Fig. 2.17. **Metric rectification via orthogonal lines I.** The affine transformation required to metrically rectify an affine image may be computed from imaged orthogonal lines. (a) Two (non-parallel) line pairs identified on the affinely rectified image (figure 2.13) correspond to orthogonal lines on the world plane. (b) The metrically rectified image. Note that in the metrically rectified image all lines orthogonal in the world are orthogonal, world squares have unit aspect ratio, and world circles are circular.

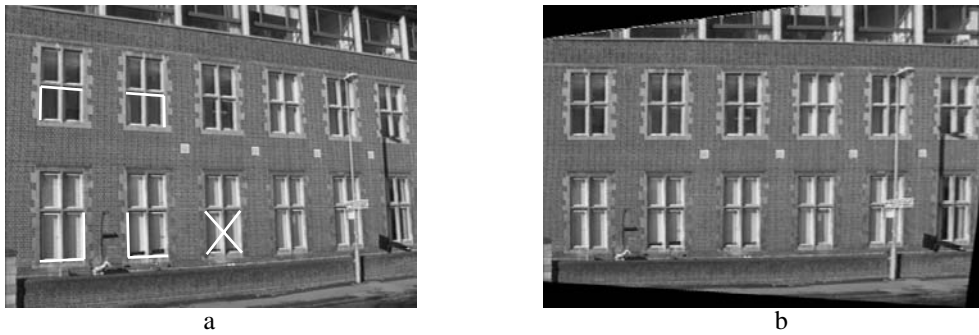


Fig. 2.18. **Metric rectification via orthogonal lines II.** (a) The conic \mathcal{C}_{∞}^* is determined on the perspective imaged plane (the front wall of the building) using the five orthogonal line pairs shown. The conic \mathcal{C}_{∞}^* determines the circular points, and equivalently the projective transformation necessary to metrically rectify the image (b). The image shown in (a) is the same perspective image as that of figure 2.4(p35), where the perspective distortion was removed by specifying the world position of four image points.

a conic to contain a point (2.4–p31), this provides a linear constraint on the elements of \mathcal{C}_{∞}^* , namely

$$(l_1 m_1, (l_1 m_2 + l_2 m_1)/2, l_2 m_2, (l_1 m_3 + l_3 m_1)/2, (l_2 m_3 + l_3 m_2)/2, l_3 m_3) \mathbf{c} = 0$$

where $\mathbf{c} = (a, b, c, d, e, f)^T$ is the conic matrix (2.3–p30) of \mathcal{C}_{∞}^* written as a 6-vector. Five such constraints can be stacked to form a 5×6 matrix, and \mathbf{c} , and hence \mathcal{C}_{∞}^* , is obtained as the null vector. This shows that \mathcal{C}_{∞}^* can be determined linearly from the images of five line pairs which are orthogonal on the world plane. An example of metric rectification using such line pair constraints is shown in figure 2.18. \triangle

Stratification. Note, in example 2.27 the affine and projective distortions are determined in one step by specifying \mathcal{C}_{∞}^* . In the previous example 2.26 first the projective and subsequently the affine distortions were removed. This two-step approach is termed *stratified*. Analogous approaches apply in 3D, and are employed in chapter 10

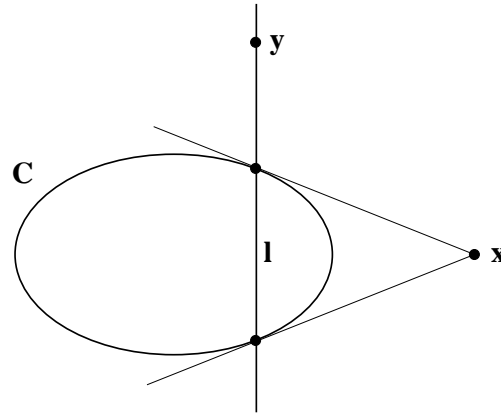


Fig. 2.19. **The pole–polar relationship.** The line $l = Cx$ is the polar of the point x with respect to the conic C , and the point $x = C^{-1}l$ is the pole of l with respect to C . The polar of x intersects the conic at the points of tangency of lines from x . If y is on l then $y^T l = y^T Cx = 0$. Points x and y which satisfy $y^T Cx = 0$ are conjugate.

on 3D reconstruction and chapter 19 on auto-calibration, when obtaining a metric from a 3D projective reconstruction.

2.8 More properties of conics

We now introduce an important geometric relation between a point, line and conic, which is termed *polarity*. Applications of this relation (to the representation of orthogonality) are given in chapter 8.

2.8.1 The pole–polar relationship

A point x and conic C define a line $l = Cx$. The line l is called the *polar* of x with respect to C , and the point x is the *pole* of l with respect to C .

- The polar line $l = Cx$ of the point x with respect to a conic C intersects the conic in two points. The two lines tangent to C at these points intersect at x .

This relationship is illustrated in figure 2.19.

Proof. Consider a point y on C . The tangent line at y is Cy , and this line contains x if $x^T Cy = 0$. Using the symmetry of C , the condition $x^T Cy = (Cx)^T y = 0$ is that the point y lies on the line Cx . Thus the polar line Cx intersects the conic in the point y at which the tangent line contains x . \square

As the point x approaches the conic the tangent lines become closer to collinear, and their contact points on the conic also become closer. In the limit that x lies on C , the polar line has two-point contact at x , and we have:

- If the point x is on C then the polar is the tangent line to the conic at x .

See result 2.7(p31).

Example 2.28. A circle of radius r centred on the x -axis at $x = a$ has the equation $(x - a)^2 + y^2 = r^2$, and is represented by the conic matrix

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & 0 \\ -a & 0 & a^2 - r^2 \end{bmatrix}.$$

The polar line of the origin is given by $\mathbf{l} = \mathbf{C}(0, 0, 1)^T = (-a, 0, a^2 - r^2)^T$. This is a vertical line at $x = (a^2 - r^2)/a$. If $r = a$ the origin lies on the circle. In this case the polar line is the y -axis and is tangent to the circle. \triangle

It is evident that the conic induces a map between points and lines of \mathbb{P}^2 . This map is a projective construction since it involves only intersections and tangency, both properties that are preserved under projective transformations. A projective map between points and lines is termed a *correlation* (an unfortunate name, given its more common usage).

Definition 2.29. A *correlation* is an invertible mapping from points of \mathbb{P}^2 to lines of \mathbb{P}^2 . It is represented by a 3×3 non-singular matrix \mathbf{A} as $\mathbf{l} = \mathbf{A}\mathbf{x}$.

A correlation provides a systematic way to dualize relations involving points and lines. It need not be represented by a symmetric matrix, but we will only consider symmetric correlations here, because of the association with conics.

- **Conjugate points.** If the point \mathbf{y} is on the line $\mathbf{l} = \mathbf{C}\mathbf{x}$ then $\mathbf{y}^T\mathbf{l} = \mathbf{y}^T\mathbf{C}\mathbf{x} = 0$. Any two points \mathbf{x}, \mathbf{y} satisfying $\mathbf{y}^T\mathbf{C}\mathbf{x} = 0$ are conjugate with respect to the conic \mathbf{C} .

The conjugacy relation is symmetric:

- If \mathbf{x} is on the polar of \mathbf{y} then \mathbf{y} is on the polar of \mathbf{x} .

This follows simply because of the symmetry of the conic matrix – the point \mathbf{x} is on the polar of \mathbf{y} if $\mathbf{x}^T\mathbf{C}\mathbf{y} = 0$, and the point \mathbf{y} is on the polar of \mathbf{x} if $\mathbf{y}^T\mathbf{C}\mathbf{x} = 0$. Since $\mathbf{x}^T\mathbf{C}\mathbf{y} = \mathbf{y}^T\mathbf{C}\mathbf{x}$, if one form is zero, then so is the other. There is a dual conjugacy relationship for lines: two lines \mathbf{l} and \mathbf{m} are conjugate if $\mathbf{l}^T\mathbf{C}^*\mathbf{m} = 0$.

2.8.2 Classification of conics

This section describes the projective and affine classification of conics.

Projective normal form for a conic. Since \mathbf{C} is a symmetric matrix it has real eigenvalues, and may be decomposed as a product $\mathbf{C} = \mathbf{U}^T\mathbf{D}\mathbf{U}$ (see section A4.2(p580)), where \mathbf{U} is an orthogonal matrix, and \mathbf{D} is diagonal. Applying the projective transformation represented by \mathbf{U} , conic \mathbf{C} is transformed to another conic $\mathbf{C}' = \mathbf{U}^{-T}\mathbf{C}\mathbf{U}^{-1} = \mathbf{U}^{-T}\mathbf{U}^T\mathbf{D}\mathbf{U}\mathbf{U}^{-1} = \mathbf{D}$. This shows that any conic is equivalent under projective transformation to one with a diagonal matrix. Let $\mathbf{D} = \text{diag}(\epsilon_1 d_1, \epsilon_2 d_2, \epsilon_3 d_3)$ where $\epsilon_i = \pm 1$ or 0 and each $d_i > 0$. Thus, \mathbf{D} may be written in the form

$$\mathbf{D} = \text{diag}(s_1, s_2, s_3)^T \text{diag}(\epsilon_1, \epsilon_2, \epsilon_3) \text{diag}(s_1, s_2, s_3)$$

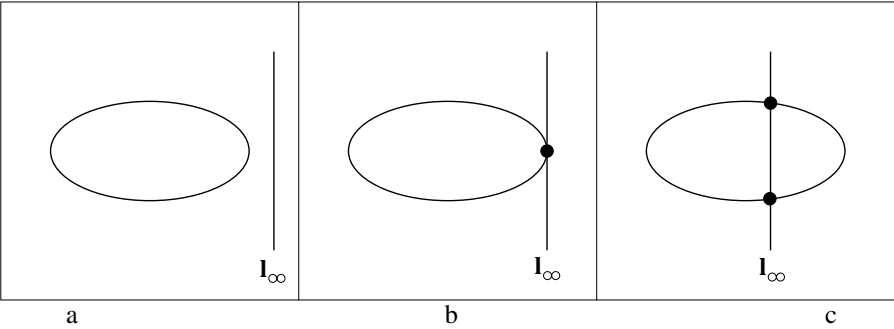


Fig. 2.20. **Affine classification of point conics.** A conic is an (a) ellipse, (b) parabola, or (c) hyperbola; according to whether it (a) has no real intersection, (b) is tangent to (2-point contact), or (c) has 2 real intersections with l_∞ . Under an affine transformation l_∞ is a fixed line, and intersections are preserved. Thus this classification is unaltered by an affinity.

where $s_i^2 = d_i$. Note that $\text{diag}(s_1, s_2, s_3)^T = \text{diag}(s_1, s_2, s_3)$. Now, transforming once more by the transformation $\text{diag}(s_1, s_2, s_3)$, the conic D is transformed to a conic with matrix $\text{diag}(\epsilon_1, \epsilon_2, \epsilon_3)$, with each $\epsilon_i = \pm 1$ or 0. Further transformation by permutation matrices may be carried out to ensure that values $\epsilon_i = 1$ occur before values $\epsilon_i = -1$ which in turn precede values $\epsilon_i = 0$. Finally, by multiplying by -1 if necessary, one may ensure that there are at least as many $+1$ entries as -1 . The various types of conics may now be enumerated, and are shown in table 2.2.

| Diagonal | Equation | Conic type |
|--------------|-----------------------|------------------------------------|
| $(1, 1, 1)$ | $x^2 + y^2 + w^2 = 0$ | Improper conic – no real points. |
| $(1, 1, -1)$ | $x^2 + y^2 - w^2 = 0$ | Circle |
| $(1, 1, 0)$ | $x^2 + y^2 = 0$ | Single real point $(0, 0, 1)^T$ |
| $(1, -1, 0)$ | $x^2 - y^2 = 0$ | Two lines $x = \pm y$ |
| $(1, 0, 0)$ | $x^2 = 0$ | Single line $x = 0$ counted twice. |

Table 2.2. **Projective classification of point conics.** Any plane conic is projectively equivalent to one of the types shown in this table. Those conics for which $\epsilon_i = 0$ for some i are known as degenerate conics, and are represented by a matrix of rank less than 3. The conic type column only describes the real points of the conics – for example as a complex conic $x^2 + y^2 = 0$ consists of the line pair $x = \pm iy$.

Affine classification of conics. The classification of (non-degenerate, proper) conics in Euclidean geometry into hyperbola, ellipse and parabola is well known. As shown above in projective geometry these three types of conic are projectively equivalent to a circle. However, in affine geometry the Euclidean classification is still valid because it depends only on the relation of l_∞ to the conic. The relation for the three types of conic is illustrated in figure 2.20.

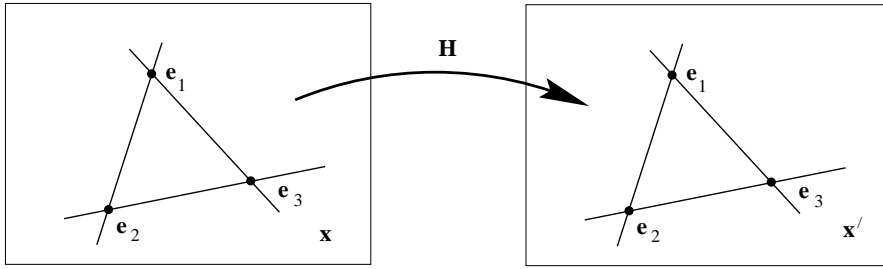


Fig. 2.21. **Fixed points and lines of a plane projective transformation.** There are three fixed points, and three fixed lines through these points. The fixed lines and points may be complex. Algebraically, the fixed points are the eigenvectors, \mathbf{e}_i , of the point transformation ($\mathbf{x}' = \mathbf{H}\mathbf{x}$), and the fixed lines are eigenvectors of the line transformation ($\mathbf{l}' = \mathbf{H}^{-T}\mathbf{l}$). Note, the fixed line is not fixed pointwise: under the transformation, points on the line are mapped to other points on the line; only the fixed points are mapped to themselves.

2.9 Fixed points and lines

We have seen, by the examples of \mathbf{l}_∞ and the circular points, that points and lines may be fixed under a projective transformation. In this section the idea is investigated more thoroughly.

Here, the source and destination planes are identified (the same) so that the transformation maps points \mathbf{x} to points \mathbf{x}' in the same coordinate system. The key idea is that an *eigenvector* corresponds to a *fixed point* of the transformation, since for an eigenvector \mathbf{e} with eigenvalue λ ,

$$\mathbf{H}\mathbf{e} = \lambda\mathbf{e}$$

and \mathbf{e} and $\lambda\mathbf{e}$ represent the same point. Often the eigenvector and eigenvalue have physical or geometric significance in computer vision applications.

A 3×3 matrix has three eigenvalues and consequently a plane projective transformation has up to three fixed points, if the eigenvalues are distinct. Since the characteristic equation is a cubic in this case, one or three of the eigenvalues, and corresponding eigenvectors, is real. A similar development can be given for *fixed lines*, which, since lines transform as (2.6-p36) $\mathbf{l}' = \mathbf{H}^{-T}\mathbf{l}$, correspond to the eigenvectors of \mathbf{H}^T .

The relationship between the fixed points and fixed lines is shown in figure 2.21. Note the lines are fixed as a set, not fixed pointwise, i.e. a point on the line is mapped to another point on the line, but in general the source and destination points will differ. There is nothing mysterious here: The projective transformation of the plane induces a 1D projective transformation on the line. A 1D projective transformation is represented by a 2×2 homogeneous matrix (section 2.5). This 1D projectivity has two fixed points corresponding to the two eigenvectors of the 2×2 matrix. These fixed points are those of the 2D projective transformation.

A further specialization concerns repeated eigenvalues. Suppose two of the eigenvalues (λ_2, λ_3 say) are identical, and that there are two distinct eigenvectors ($\mathbf{e}_2, \mathbf{e}_3$), corresponding to $\lambda_2 = \lambda_3$. Then the line containing the eigenvectors $\mathbf{e}_2, \mathbf{e}_3$ will be fixed pointwise, i.e. it is a line of fixed points. For suppose $\mathbf{x} = \alpha\mathbf{e}_2 + \beta\mathbf{e}_3$; then

$$\mathbf{H}\mathbf{x} = \lambda_2\alpha\mathbf{e}_2 + \lambda_2\beta\mathbf{e}_3 = \lambda_2\mathbf{x}$$

i.e. a point on the line through two degenerate eigenvectors is mapped to itself (only differing by scale). Another possibility is that $\lambda_2 = \lambda_3$, but that there is only one corresponding eigenvector. In this case, the eigenvector has *algebraic dimension* equal to two, but *geometric dimension* equal to one. Then there is one fewer fixed point (2 instead of 3). Various cases of repeated eigenvalues are discussed further in appendix 7(p628).

We now examine the fixed points and lines of the hierarchy of projective transformation subgroups of section 2.4. Affine transformations, and the more specialized forms, have two eigenvectors which are ideal points ($x_3 = 0$), and which correspond to the eigenvectors of the upper left 2×2 matrix. The third eigenvector is finite in general.

A Euclidean matrix. The two ideal fixed points are the complex conjugate pair of circular points I, J, with corresponding eigenvalues $\{e^{i\theta}, e^{-i\theta}\}$, where θ is the rotation angle. The third eigenvector, which has unit eigenvalue, is called the *pole*. The Euclidean transformation is equal to a pure rotation by θ about this point with no translation.

A special case is that of a pure translation (i.e. where $\theta = 0$). Here the eigenvalues are triply degenerate. The line at infinity is fixed pointwise, and there is a pencil of fixed lines through the point $(t_x, t_y, 0)^T$ which corresponds to the translation direction. Consequently lines parallel to \mathbf{t} are fixed. This is an example of an elation (see section A7.3(p631)).

A similarity matrix. The two ideal fixed points are again the circular points. The eigenvalues are $\{1, se^{i\theta}, se^{-i\theta}\}$. The action can be understood as a rotation and isotropic scaling by s about the finite fixed point. Note that the eigenvalues of the circular points again encode the angle of rotation.

An affine matrix. The two ideal fixed points can be real or complex conjugates, but the fixed line $\mathbf{l}_\infty = (0, 0, 1)^T$ through these points is real in either case.

2.10 Closure

2.10.1 The literature

A gentle introduction to plane projective geometry, written for computer vision researchers, is given in the appendix of Mundy and Zisserman [Mundy-92]. A more formal approach is that of Semple and Kneebone [Semple-79], but [Springer-64] is more readable.

On the recovery of affine and metric scene properties for an imaged plane, Collins and Beveridge [Collins-93] use the vanishing line to recover affine properties from satellite images, and Liebowitz and Zisserman [Liebowitz-98] use metric information on the plane, such as right angles, to recover the metric geometry.

2.10.2 Notes and exercises

(i) Affine transformations.

- (a) Show that an affine transformation can map a circle to an ellipse, but cannot map an ellipse to a hyperbola or parabola.
- (b) Prove that under an affine transformation the ratio of lengths on parallel line segments is an invariant, but that the ratio of two lengths that are not parallel is not.
- (ii) **Projective transformations.** Show that there is a three-parameter family of projective transformations which fix (as a set) a unit circle at the origin, i.e. a unit circle at the origin is mapped to a unit circle at the origin (hint, use result 2.13(p37) to compute the transformation). What is the geometric interpretation of this family?
- (iii) **Isotropies.** Show that two lines have an invariant under a similarity transformation; and that two lines and two points have an invariant under a projective transformation. In both cases the equality case of the counting argument (result 2.16(p43)) is violated. Show that for these two cases the respective transformation cannot be fully determined, although it is partially determined.
- (iv) **Invariants.** Using the transformation rules for points, lines and conics show:

- (a) Two lines, l_1, l_2 , and two points, x_1, x_2 , not lying on the lines have the invariant

$$I = \frac{(l_1^T x_1)(l_2^T x_2)}{(l_1^T x_2)(l_2^T x_1)}$$

(see the previous question).

- (b) A conic C and two points, x_1 and x_2 , in general position have the invariant

$$I = \frac{(x_1^T C x_2)^2}{(x_1^T C x_1)(x_2^T C x_2)}.$$

- (c) Show that the projectively invariant expression for measuring angles (2.22) is equivalent to Laguerre's projectively invariant expression involving a cross ratio with the circular points (see [Springer-64]).
- (v) **The cross ratio.** Prove the invariance of the cross ratio of four collinear points under projective transformations of the line (2.18–p45). Hint, start with the transformation of two points on the line written as $\bar{x}'_i = \lambda_i H_{2 \times 2} \bar{x}_i$ and $\bar{x}'_j = \lambda_j H_{2 \times 2} \bar{x}_j$, where equality is *not* up to scale, then from the properties of determinants show that $|\bar{x}'_i \bar{x}'_j| = \lambda_i \lambda_j \det H_{2 \times 2} |\bar{x}_i \bar{x}_j|$ and continue from here. An alternative derivation method is given in [Semple-79].
- (vi) **Polarity.** Figure 2.19 shows the geometric construction of the polar line for a point x *outside* an ellipse. Give a geometric construction for the polar when the point is inside. Hint, start by choosing any line through x . The pole of this line is a point on the polar of x .
- (vii) **Conics.** If the sign of the conic matrix C is chosen such that two eigenvalues are positive and one negative, then internal and external points may be distinguished according to the sign of $x^T C x$: the point x is inside/on/outside the conic

C if $\mathbf{x}^T C \mathbf{x}$ is negative/zero/positive respectively. This can be seen by example from a circle $C = \text{diag}(1, 1, -1)$. Under projective transformations internality is invariant, though its interpretation requires care in the case of an ellipse being transformed to a hyperbola (see figure 2.20).

(viii) **Dual conics.** Show that the matrix $[\mathbf{l}]_{\times} C [\mathbf{l}]_{\times}$ represents a rank 2 dual conic which consists of the two points at which the line \mathbf{l} intersects the (point) conic C (the notation $[\mathbf{l}]_{\times}$ is defined in (A4.5–p581)).

(ix) **Special projective transformations.** Suppose points on a scene plane are related by reflection in a line: for example, a plane object with bilateral symmetry. Show that in a perspective image of the plane the points are related by a projectivity H satisfying $H^2 = I$. Furthermore, show that under H there is a line of fixed points corresponding to the imaged reflection line, and that H has an eigenvector, not lying on this line, which is the vanishing point of the reflection direction (H is a planar harmonic homology, see section A7.2(p629)).

Now suppose that the points are related by a finite rotational symmetry: for example, points on a hexagonal bolt head. Show in this case that $H^n = I$, where n is the order of rotational symmetry (6 for a hexagonal symmetry), that the eigenvalues of H determine the rotation angle, and that the eigenvector corresponding to the real eigenvalue is the image of the centre of the rotational symmetry.

Projective Geometry and Transformations of 3D

This chapter describes the properties and entities of projective 3-space, or \mathbb{P}^3 . Many of these are straightforward generalizations of those of the projective plane, described in chapter 2. For example, in \mathbb{P}^3 Euclidean 3-space is augmented with a set of ideal points which are on a *plane* at infinity, π_∞ . This is the analogue of l_∞ in \mathbb{P}^2 . Parallel lines, and now parallel *planes*, intersect on π_∞ . Not surprisingly, homogeneous coordinates again play an important role, here with all dimensions increased by one. However, additional properties appear by virtue of the extra dimension. For example, two lines always intersect on the projective plane, but they need not intersect in 3-space.

The reader should be familiar with the ideas and notation of chapter 2 before reading this chapter. We will concentrate here on the differences and additional geometry introduced by adding the extra dimension, and will not repeat the bulk of the material of the previous chapter.

3.1 Points and projective transformations

A point \mathbf{X} in 3-space is represented in homogeneous coordinates as a 4-vector. Specifically, the homogeneous vector $\mathbf{X} = (x_1, x_2, x_3, x_4)^T$ with $x_4 \neq 0$ represents the point $(x, y, z)^T$ of \mathbb{R}^3 with inhomogeneous coordinates

$$x = x_1/x_4, \quad y = x_2/x_4, \quad z = x_3/x_4.$$

For example, a homogeneous representation of $(x, y, z)^T$ is $\mathbf{X} = (x, y, z, 1)^T$. Homogeneous points with $x_4 = 0$ represent points at infinity.

A projective transformation acting on \mathbb{P}^3 is a linear transformation on homogeneous 4-vectors represented by a non-singular 4×4 matrix: $\mathbf{X}' = \mathbf{H}\mathbf{X}$. The matrix \mathbf{H} representing the transformation is homogeneous and has 15 degrees of freedom. The degrees of freedom follow from the 16 elements of the matrix less one for overall scaling.

As in the case of planar projective transformations, the map is a collineation (lines are mapped to lines), which preserves incidence relations such as the intersection point of a line with a plane, and order of contact.

3.2 Representing and transforming planes, lines and quadrics

In \mathbb{P}^3 points and *planes* are dual, and their representation and development is analogous to the point–line duality in \mathbb{P}^2 . Lines are self-dual in \mathbb{P}^3 .

3.2.1 Planes

A plane in 3-space may be written as

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0. \quad (3.1)$$

Clearly this equation is unaffected by multiplication by a non-zero scalar, so only the three independent ratios $\{\pi_1 : \pi_2 : \pi_3 : \pi_4\}$ of the plane coefficients are significant. It follows that a plane has 3 degrees of freedom in 3-space. The homogeneous representation of the plane is the 4-vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^\top$.

Homogenizing (3.1) by the replacements $X \mapsto X_1/X_4$, $Y \mapsto X_2/X_4$, $Z \mapsto X_3/X_4$ gives

$$\pi_1 X_1 + \pi_2 X_2 + \pi_3 X_3 + \pi_4 X_4 = 0$$

or more concisely

$$\boldsymbol{\pi}^\top \mathbf{X} = 0 \quad (3.2)$$

which expresses that the point \mathbf{X} is on the plane $\boldsymbol{\pi}$.

The first 3 components of $\boldsymbol{\pi}$ correspond to the plane normal of Euclidean geometry – using inhomogeneous notation (3.2) becomes the familiar plane equation written in 3-vector notation as $\mathbf{n} \cdot \tilde{\mathbf{X}} + d = 0$, where $\mathbf{n} = (\pi_1, \pi_2, \pi_3)^\top$, $\tilde{\mathbf{X}} = (X, Y, Z)^\top$, $X_4 = 1$ and $d = \pi_4$. In this form $d/\|\mathbf{n}\|$ is the distance of the plane from the origin.

Join and incidence relations. In \mathbb{P}^3 there are numerous geometric relations between planes and points and lines. For example,

- (i) A plane is defined uniquely by the join of three points, or the join of a line and point, in general position (i.e. the points are not collinear or incident with the line in the latter case).
- (ii) Two distinct planes intersect in a unique line.
- (iii) Three distinct planes intersect in a unique point.

These relations have algebraic representations which will now be developed in the case of points and planes. The representations of the relations involving lines are not as simple as those arising from 3D vector algebra of \mathbb{P}^2 (e.g. $1 = \mathbf{x} \times \mathbf{y}$), and are postponed until line representations are introduced in section 3.2.2.

Three points define a plane. Suppose three points \mathbf{X}_i are incident with the plane $\boldsymbol{\pi}$. Then each point satisfies (3.2) and thus $\boldsymbol{\pi}^\top \mathbf{X}_i = 0$, $i = 1, \dots, 3$. Stacking these equations into a matrix gives

$$\begin{bmatrix} \mathbf{X}_1^\top \\ \mathbf{X}_2^\top \\ \mathbf{X}_3^\top \end{bmatrix} \boldsymbol{\pi} = \mathbf{0}. \quad (3.3)$$

Since three points $\mathbf{X}_1, \mathbf{X}_2$ and \mathbf{X}_3 in general position are linearly independent, it follows that the 3×4 matrix composed of the points as rows has rank 3. The plane π defined by the points is thus obtained uniquely (up to scale) as the 1-dimensional (right) null-space. If the matrix has only a rank of 2, and consequently the null-space is 2-dimensional, then the points are collinear, and define a pencil of planes with the line of collinear points as axis.

In \mathbb{IP}^2 , where points are dual to lines, a line l through two points \mathbf{x}, \mathbf{y} can similarly be obtained as the null-space of the 2×3 matrix with \mathbf{x}^T and \mathbf{y}^T as rows. However, a more convenient direct formula $l = \mathbf{x} \times \mathbf{y}$ is also available from vector algebra. In \mathbb{IP}^3 the analogous expression is obtained from properties of determinants and minors.

We start from the matrix $M = [\mathbf{X}, \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3]$ which is composed of a general point \mathbf{X} and the three points \mathbf{X}_i which define the plane π . The determinant $\det M = 0$ when \mathbf{X} lies on π since the point \mathbf{X} is then expressible as a linear combination of the points $\mathbf{X}_i, i = 1, \dots, 3$. Expanding the determinant about the column \mathbf{X} we obtain

$$\det M = X_1 D_{234} - X_2 D_{134} + X_3 D_{124} - X_4 D_{123}$$

where D_{jkl} is the determinant formed from the jkl rows of the 4×3 matrix $[\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3]$. Since $\det M = 0$ for points on π we can then read off the plane coefficients as

$$\pi = (D_{234}, -D_{134}, D_{124}, -D_{123})^T. \quad (3.4)$$

This is the solution vector (the null-space) of (3.3) above.

Example 3.1. Suppose the three points defining the plane are

$$\mathbf{X}_1 = \begin{pmatrix} \tilde{\mathbf{X}}_1 \\ 1 \end{pmatrix} \quad \mathbf{X}_2 = \begin{pmatrix} \tilde{\mathbf{X}}_2 \\ 1 \end{pmatrix} \quad \mathbf{X}_3 = \begin{pmatrix} \tilde{\mathbf{X}}_3 \\ 1 \end{pmatrix}$$

where $\tilde{\mathbf{X}} = (X, Y, Z)^T$. Then

$$D_{234} = \begin{vmatrix} Y_1 & Y_2 & Y_3 \\ Z_1 & Z_2 & Z_3 \\ 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} Y_1 - Y_3 & Y_2 - Y_3 & Y_3 \\ Z_1 - Z_3 & Z_2 - Z_3 & Z_3 \\ 0 & 0 & 1 \end{vmatrix} = ((\tilde{\mathbf{X}}_1 - \tilde{\mathbf{X}}_3) \times (\tilde{\mathbf{X}}_2 - \tilde{\mathbf{X}}_3))_1$$

and similarly for the other components, giving

$$\pi = \begin{pmatrix} (\tilde{\mathbf{X}}_1 - \tilde{\mathbf{X}}_3) \times (\tilde{\mathbf{X}}_2 - \tilde{\mathbf{X}}_3) \\ -\tilde{\mathbf{X}}_3^T (\tilde{\mathbf{X}}_1 \times \tilde{\mathbf{X}}_2) \end{pmatrix}.$$

This is the familiar result from Euclidean vector geometry where, for example, the plane normal is computed as $(\tilde{\mathbf{X}}_1 - \tilde{\mathbf{X}}_3) \times (\tilde{\mathbf{X}}_2 - \tilde{\mathbf{X}}_3)$. \triangle

Three planes define a point. The development here is dual to the case of three points defining a plane. The intersection point \mathbf{X} of three planes π_i can be computed straightforwardly as the (right) null-space of the 3×4 matrix composed of the planes as rows:

$$\begin{bmatrix} \pi_1^T \\ \pi_2^T \\ \pi_3^T \end{bmatrix} \mathbf{X} = \mathbf{0}. \quad (3.5)$$

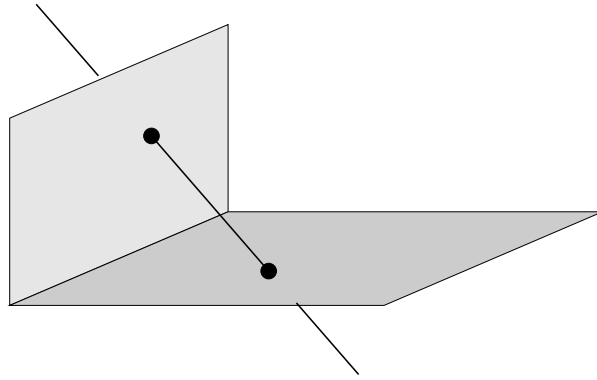


Fig. 3.1. A line may be specified by its points of intersection with two orthogonal planes. Each intersection point has 2 degrees of freedom, which demonstrates that a line in \mathbb{P}^3 has a total of 4 degrees of freedom.

A direct solution for \mathbf{X} , in terms of determinants of 3×3 submatrices, is obtained as an analogue of (3.4), though computationally a numerical solution would be obtained by algorithm A5.1(p589).

The two following results are direct analogues of their 2D counterparts.

Projective transformation. Under the point transformation $\mathbf{X}' = \mathbf{H}\mathbf{X}$, a plane transforms as

$$\pi' = \mathbf{H}^{-T} \pi. \quad (3.6)$$

Parametrized points on a plane. The points \mathbf{X} on the plane π may be written as

$$\mathbf{X} = \mathbf{M}\mathbf{x} \quad (3.7)$$

where the columns of the 4×3 matrix \mathbf{M} generate the rank 3 null-space of π^T , i.e. $\pi^T \mathbf{M} = \mathbf{0}$, and the 3-vector \mathbf{x} (which is a point on the projective plane \mathbb{P}^2) parametrizes points on the plane π . \mathbf{M} is not unique, of course. Suppose the plane is $\pi = (a, b, c, d)^T$ and a is non-zero, then \mathbf{M}^T can be written as $\mathbf{M}^T = [\mathbf{p} \mid \mathbf{I}_{3 \times 3}]$, where $\mathbf{p} = (-b/a, -c/a, -d/a)^T$.

This parametrized representation is simply the analogue in 3D of a line l in \mathbb{P}^2 defined as a linear combination of its 2D null-space as $\mathbf{x} = \mu \mathbf{a} + \lambda \mathbf{b}$, where $l^T \mathbf{a} = l^T \mathbf{b} = 0$.

3.2.2 Lines

A line is defined by the *join* of two points or the intersection of two planes. Lines have 4 degrees of freedom in 3-space. A convincing way to count these degrees of freedom is to think of a line as defined by its intersection with two orthogonal planes, as in figure 3.1. The point of intersection on each plane is specified by two parameters, producing a total of 4 degrees of freedom for the line.

Lines are very awkward to represent in 3-space since a natural representation for an object with 4 degrees of freedom would be a homogeneous 5-vector. The problem is that a homogeneous 5 vector cannot easily be used in mathematical expressions together with the 4-vectors representing points and planes. To overcome this problem

a number of line representations have been proposed, and these differ in their mathematical complexity. We survey three of these representations. In each case the representation provides mechanisms for a line to be defined by: the join of two points, a dual version where the line is defined by the intersection of two planes, and also a map between the two definitions. The representations also enable join and incidence relations to be computed, for example the point at which a line intersects a plane.

I. Null-space and span representation. This representation builds on the intuitive geometric notion that a line is a pencil (one-parameter family) of collinear points, and is defined by any two of these points. Similarly, a line is the axis of a pencil of planes, and is defined by the intersection of any two planes from the pencil. In both cases the actual points or planes are not important (in fact two points have 6 degrees of freedom and are represented by two 4-vectors – far too many parameters). This notion is captured mathematically by representing a line as the *span* of two vectors. Suppose A, B are two (non-coincident) space points. Then the line joining these points is represented by the span of the row space of the 2×4 matrix W composed of A^T and B^T as rows:

$$W = \begin{bmatrix} A^T \\ B^T \end{bmatrix}.$$

Then:

- (i) The span of W^T is the pencil of points $\lambda A + \mu B$ on the line.
- (ii) The span of the 2-dimensional right null-space of W is the pencil of planes with the line as axis.

It is evident that two other points, A'^T and B'^T , on the line will generate a matrix W' with the same span as W , so that the span, and hence the representation, is independent of the particular points used to define it.

To prove the null-space property, suppose that P and Q are a basis for the null-space. Then $WP = 0$ and consequently $A^T P = B^T P = 0$, so that P is a plane containing the points A and B . Similarly, Q is a distinct plane also containing the points A and B . Thus A and B lie on both the (linearly independent) planes P and Q , so the line defined by W is the plane intersection. Any plane of the pencil, with the line as axis, is given by the span $\lambda'P + \mu'Q$.

The dual representation of a line as the intersection of two planes, P, Q , follows in a similar manner. The line is represented as the span (of the row space) of the 2×4 matrix W^* composed of P^T and Q^T as rows:

$$W^* = \begin{bmatrix} P^T \\ Q^T \end{bmatrix}$$

with the properties

- (i) The span of W^{*T} is the pencil of planes $\lambda'P + \mu'Q$ with the line as axis.
- (ii) The span of the 2-dimensional null-space of W^* is the pencil of points on the line.

The two representations are related by $W^* W^T = W W^{*T} = 0_{2 \times 2}$, where $0_{2 \times 2}$ is a 2×2 null matrix.

Example 3.2. The x-axis is represented as

$$W = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad W^* = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

where the points A and B are here the origin and ideal point in the x-direction, and the planes P and Q are the XY- and XZ-planes respectively. \triangle

Join and incidence relations are also computed from null-spaces.

- (i) The plane π defined by the join of the point X and line W is obtained from the null-space of

$$M = \begin{bmatrix} W \\ X^T \end{bmatrix}.$$

If the null-space of M is 2-dimensional then X is on W, otherwise $M\pi = 0$.

- (ii) The point X defined by the intersection of the line W with the plane π is obtained from the null-space of

$$M = \begin{bmatrix} W^* \\ \pi^T \end{bmatrix}.$$

If the null-space of M is 2-dimensional then the line W is on π , otherwise $MX = 0$.

These properties can be derived almost by inspection. For example, the first is equivalent to three points defining a plane (3.3).

The span representation is very useful in practical numerical implementations where null-spaces can be computed simply by using the SVD algorithm (see section A4.4-(p585)) available with most matrix packages. The representation is also useful in estimation problems, where it is often not a problem that the entity being estimated is over-parametrized (see the discussion of section 4.5(p110)).

II. Plücker matrices. Here a line is represented by a 4×4 skew-symmetric homogeneous matrix. In particular, the line joining the two points A, B is represented by the matrix L with elements

$$l_{ij} = A_i B_j - B_i A_j$$

or equivalently in vector notation as

$$L = AB^T - BA^T \quad (3.8)$$

First a few properties of L:

- (i) L has rank 2. Its 2-dimensional null-space is spanned by the pencil of planes with the line as axis (in fact $LW^{*T} = 0$, with 0 a 4×2 null-matrix).

- (ii) The representation has the required 4 degrees of freedom for a line. This is accounted as follows: the skew-symmetric matrix has 6 independent non-zero elements, but only their 5 ratios are significant, and furthermore because $\det L = 0$ the elements satisfy a (quadratic) constraint (see below). The net number of degrees of freedom is then 4.
- (iii) The relation $L = AB^T - BA^T$ is the generalization to 4-space of the vector product formula $\mathbf{l} = \mathbf{x} \times \mathbf{y}$ of \mathbb{P}^2 for a line \mathbf{l} defined by two points \mathbf{x}, \mathbf{y} all represented by 3-vectors.
- (iv) The matrix L is independent of the points A, B used to define it, since if a different point C on the line is used, with $C = A + \mu B$, then the resulting matrix is

$$\begin{aligned}\hat{L} &= AC^T - CA^T = A(A^T + \mu B^T) - (A + \mu B)A^T \\ &= AB^T - BA^T = L.\end{aligned}$$

- (v) Under the point transformation $\mathbf{X}' = H\mathbf{X}$, the matrix transforms as $L' = HLH^T$, i.e. it is a valency-2 tensor (see appendix 1(p562)).

Example 3.3. From (3.8) the X-axis is represented as

$$L = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

where the points A and B are (as in the previous example) the origin and ideal point in the X-direction respectively. \triangle

A dual Plücker representation L^* is obtained for a line formed by the intersection of two planes P, Q ,

$$L^* = PQ^T - QP^T \quad (3.9)$$

and has similar properties to L . Under the point transformation $\mathbf{X}' = H\mathbf{X}$, the matrix L^* transforms as $L^{*'} = H^{-T}LH^{-1}$. The matrix L^* can be obtained directly from L by a simple rewrite rule:

$$l_{12} : l_{13} : l_{14} : l_{23} : l_{42} : l_{34} = l_{34}^* : l_{42}^* : l_{23}^* : l_{14}^* : l_{13}^* : l_{12}^*. \quad (3.10)$$

The correspondence rule is very simple: the indices of the dual and original component always include all the numbers $\{1, 2, 3, 4\}$, so if the original is ij then the dual is those numbers of $\{1, 2, 3, 4\}$ which are not ij . For example $12 \mapsto 34$.

Join and incidence properties are very nicely represented in this notation:

- (i) The plane defined by the join of the point \mathbf{X} and line L is

$$\pi = L^* \mathbf{X}$$

and $L^* \mathbf{X} = 0$ if, and only if, \mathbf{X} is on L .

(ii) The point defined by the intersection of the line L with the plane π is

$$\mathbf{X} = L\pi$$

and $L\pi = \mathbf{0}$ if, and only if, L is on π .

The properties of two (or more) lines L_1, L_2, \dots can be obtained from the null-space of the matrix $M = [L_1, L_2, \dots]$. For example if the lines are coplanar then M^T has a 1-dimensional null-space corresponding to the plane π of the lines.

Example 3.4. The intersection of the X -axis with the plane $x = 1$ is given by $\mathbf{X} = L\pi$ as

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

which is the inhomogeneous point $(x, y, z)^T = (1, 0, 0)^T$. \triangle

III. Plücker line coordinates. The Plücker line coordinates are the six non-zero elements of the 4×4 skew-symmetric Plücker matrix (3.8) L , namely¹

$$\mathcal{L} = \{l_{12}, l_{13}, l_{14}, l_{23}, l_{24}, l_{34}\}. \quad (3.11)$$

This is a homogeneous 6-vector, and thus is an element of \mathbb{P}^5 . It follows from evaluating $\det L = 0$ that the coordinates satisfy the equation

$$l_{12}l_{34} + l_{13}l_{42} + l_{14}l_{23} = 0. \quad (3.12)$$

A 6-vector \mathcal{L} only corresponds to a line in 3-space if it satisfies (3.12). The geometric interpretation of this constraint is that the lines of \mathbb{P}^3 define a (co-dimension 1) surface in \mathbb{P}^5 which is known as the *Klein quadric*, a quadric because the terms of (3.12) are quadratic in the Plücker line coordinates.

Suppose two lines $\mathcal{L}, \hat{\mathcal{L}}$ are the joins of the points A, B and \hat{A}, \hat{B} respectively. The lines intersect if and only if the four points are coplanar. A necessary and sufficient condition for this is that $\det[A, B, \hat{A}, \hat{B}] = 0$. It can be shown that the determinant expands as

$$\begin{aligned} \det[A, B, \hat{A}, \hat{B}] &= l_{12}\hat{l}_{34} + \hat{l}_{12}l_{34} + l_{13}\hat{l}_{42} + \hat{l}_{13}l_{42} + l_{14}\hat{l}_{23} + \hat{l}_{14}l_{23} \\ &= (\mathcal{L}|\hat{\mathcal{L}}). \end{aligned} \quad (3.13)$$

Since the Plücker coordinates are independent of the particular points used to define them, the bilinear product $(\mathcal{L}|\hat{\mathcal{L}})$ is independent of the points used in the derivation and only depends on the lines \mathcal{L} and $\hat{\mathcal{L}}$. Then we have

Result 3.5. Two lines \mathcal{L} and $\hat{\mathcal{L}}$ are coplanar (and thus intersect) if and only if $(\mathcal{L}|\hat{\mathcal{L}}) = 0$.

This product appears in a number of useful formulae:

¹ The element l_{42} is conventionally used instead of l_{24} as it eliminates negatives in many of the subsequent formulae.

- (i) A 6-vector \mathcal{L} only represents a line in \mathbb{P}^3 if $(\mathcal{L}|\mathcal{L}) = 0$. This is simply repeating the Klein quadric constraint (3.12) above.
- (ii) Suppose two lines $\mathcal{L}, \hat{\mathcal{L}}$ are the intersections of the planes \mathbf{P}, \mathbf{Q} and $\hat{\mathbf{P}}, \hat{\mathbf{Q}}$ respectively. Then

$$(\mathcal{L}|\hat{\mathcal{L}}) = \det[\mathbf{P}, \mathbf{Q}, \hat{\mathbf{P}}, \hat{\mathbf{Q}}]$$

and again the lines intersect if and only if $(\mathcal{L}|\hat{\mathcal{L}}) = 0$.

- (iii) If \mathcal{L} is the intersection of two planes \mathbf{P} and \mathbf{Q} and $\hat{\mathcal{L}}$ is the join of two points \mathbf{A} and \mathbf{B} , then

$$(\mathcal{L}|\hat{\mathcal{L}}) = (\mathbf{P}^\top \mathbf{A})(\mathbf{Q}^\top \mathbf{B}) - (\mathbf{Q}^\top \mathbf{A})(\mathbf{P}^\top \mathbf{B}). \quad (3.14)$$

Plücker coordinates are useful in algebraic derivations. They will be used in defining the map from a line in 3-space to its image in chapter 8.

3.2.3 Quadrics and dual quadrics

A quadric is a surface in \mathbb{P}^3 defined by the equation

$$\mathbf{X}^\top \mathbf{Q} \mathbf{X} = 0 \quad (3.15)$$

where \mathbf{Q} is a symmetric 4×4 matrix. Often the matrix \mathbf{Q} and the quadric surface it defines are not distinguished, and we will simply refer to the quadric \mathbf{Q} .

Many of the properties of quadrics follow directly from those of conics in section 2.2.3(p30). To highlight a few:

- (i) A quadric has 9 degrees of freedom. These correspond to the ten independent elements of a 4×4 symmetric matrix less one for scale.
- (ii) Nine points in general position define a quadric.
- (iii) If the matrix \mathbf{Q} is singular, then the quadric is *degenerate*, and may be defined by fewer points.
- (iv) A quadric defines a polarity between a point and a plane, in a similar manner to the polarity defined by a conic between a point and a line (section 2.8.1). The plane $\pi = \mathbf{Q}\mathbf{X}$ is the polar plane of \mathbf{X} with respect to \mathbf{Q} . In the case that \mathbf{Q} is non-singular and \mathbf{X} is outside the quadric, the polar plane is defined by the points of contact with \mathbf{Q} of the cone of rays through \mathbf{X} tangent to \mathbf{Q} . If \mathbf{X} lies on \mathbf{Q} , then $\mathbf{Q}\mathbf{X}$ is the tangent plane to \mathbf{Q} at \mathbf{X} .
- (v) The intersection of a plane π with a quadric \mathbf{Q} is a conic \mathbf{C} . Computing the conic can be tricky because it requires a coordinate system for the plane. Recall from (3.7) that a coordinate system for the plane can be defined by the complement space to π as $\mathbf{X} = \mathbf{M}\mathbf{x}$. Points on π are on \mathbf{Q} if $\mathbf{X}^\top \mathbf{Q} \mathbf{X} = \mathbf{x}^\top \mathbf{M}^\top \mathbf{Q} \mathbf{M} \mathbf{x} = 0$. These points lie on a conic \mathbf{C} , since $\mathbf{x}^\top \mathbf{C} \mathbf{x} = 0$, with $\mathbf{C} = \mathbf{M}^\top \mathbf{Q} \mathbf{M}$.
- (vi) Under the point transformation $\mathbf{X}' = \mathbf{H}\mathbf{X}$, a (point) quadric transforms as

$$\mathbf{Q}' = \mathbf{H}^{-\top} \mathbf{Q} \mathbf{H}^{-1}. \quad (3.16)$$

The dual of a quadric is also a quadric. Dual quadrics are equations on planes: the tangent planes π to the point quadric \mathbf{Q} satisfy $\pi^\top \mathbf{Q}^* \pi = 0$, where $\mathbf{Q}^* = \text{adjoint } \mathbf{Q}$,

or Q^{-1} if Q is invertible. Under the point transformation $X' = HX$, a dual quadric transforms as

$$Q^{*'} = HQ^*H^T. \quad (3.17)$$

The algebra of imaging a quadric is far simpler for a dual quadric than a point quadric. This is detailed in chapter 8.

3.2.4 Classification of quadrics

Since the matrix, Q , representing a quadric is symmetric, it may be decomposed as $Q = U^T D U$ where U is a real orthogonal matrix and D is a real diagonal matrix. Further, by appropriate scaling of the rows of U , one may write $Q = H^T D H$ where D is diagonal with entries equal to 0, 1, or -1 . We may further ensure that the zero entries of D appear last along the diagonal, and that the $+1$ entries appear first. Now, replacement of $Q = H^T D H$ by D is equivalent to a projective transformation effected by the matrix H (see (3.16)). Thus, up to projective equivalence, we may assume that the quadric is represented by a matrix D of the given simple form.

The *signature* of a diagonal matrix D , denoted $\sigma(D)$, is defined to be the number of $+1$ entries minus the number of -1 entries. This definition is extended to arbitrary real symmetric matrices Q by defining $\sigma(Q) = \sigma(D)$ such that $Q = H^T D H$, where H is a real matrix. It may be proved that the signature is well defined, being independent of the particular choice of H . Since the matrix representing a quadric is defined only up to sign, we may assume that its signature is non-negative. Then, the projective type of a quadric is uniquely determined by its rank and signature. This will allow us to enumerate the different projective equivalence classes of quadrics.

A quadric represented by a diagonal matrix $\text{diag}(d_1, d_2, d_3, d_4)$ corresponds to a set of points satisfying an equation $d_1 X^2 + d_2 Y^2 + d_3 Z^2 + d_4 T^2 = 0$. One may set $T = 1$ to get an equation for the non-infinite points on the quadric. See table 3.1. Examples of quadric surfaces are shown in figure 3.2 – figure 3.4.

| Rank | σ | Diagonal | Equation | Realization |
|------|----------|----------------------|---------------------------|----------------------------|
| 4 | 4 | (1, 1, 1, 1) | $X^2 + Y^2 + Z^2 + 1 = 0$ | No real points |
| | 2 | (1, 1, 1, -1) | $X^2 + Y^2 + Z^2 = 1$ | Sphere |
| | 0 | (1, 1, -1 , -1) | $X^2 + Y^2 = Z^2 + 1$ | Hyperboloid of one sheet |
| 3 | 3 | (1, 1, 1, 0) | $X^2 + Y^2 + Z^2 = 0$ | One point $(0, 0, 0, 1)^T$ |
| | 1 | (1, 1, -1 , 0) | $X^2 + Y^2 = Z^2$ | Cone at the origin |
| 2 | 2 | (1, 1, 0, 0) | $X^2 + Y^2 = 0$ | Single line (Z-axis) |
| | 0 | (1, -1 , 0, 0) | $X^2 = Y^2$ | Two planes $X = \pm Y$ |
| 1 | 1 | (1, 0, 0, 0) | $X^2 = 0$ | The plane $X = 0$ |

Table 3.1. *Categorization of point quadrics.*

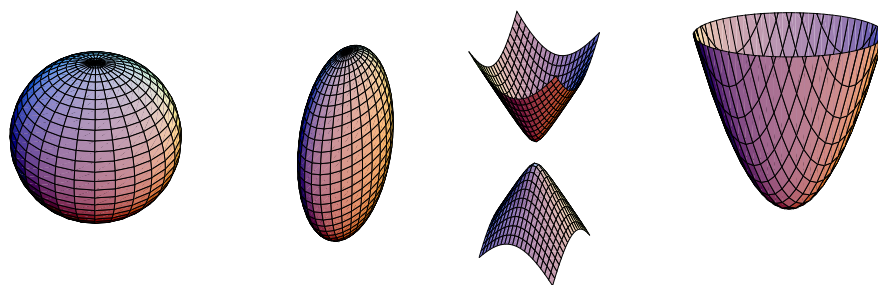


Fig. 3.2. **Non-ruled quadrics.** This shows plots of a sphere, ellipsoid, hyperboloid of two sheets and paraboloid. They are all projectively equivalent.

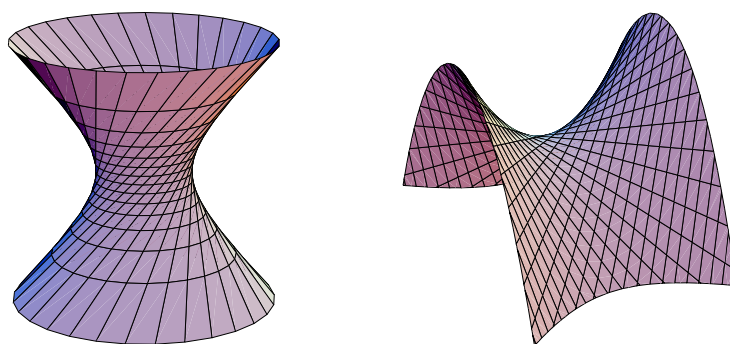


Fig. 3.3. **Ruled quadrics.** Two examples of a hyperboloid of one sheet are given. These surfaces are given by equations $x^2 + y^2 = z^2 + 1$ and $xy = z$ respectively, and are projectively equivalent. Note that these two surfaces are made up of two sets of disjoint straight lines, and that each line from one set meets each line from the other set. The two quadrics shown here are projectively (though not affinely) equivalent.

Ruled quadrics. Quadrics fall into two classes – ruled and unruled quadrics. A ruled quadric is one that contains a straight line. More particularly, as shown in figure 3.3, the non-degenerate ruled quadric (hyperboloid of one sheet) contains two families of straight lines called *generators*. For more properties of ruled quadrics, refer to [Semple-79].

The most interesting of the quadrics are the two quadrics of rank 4. Note that these two quadrics differ even in their topological type. The quadric of signature 2 (the sphere) is (obviously enough) topologically a sphere. On the other hand, the hyperboloid of 1 sheet is *not* topologically equivalent (homeomorphic) to a sphere. In fact, it is topologically a torus (topologically equivalent to $S^1 \times S^1$). This gives the clearest indication that they are not projectively equivalent.

3.3 Twisted cubics

The twisted cubic may be considered to be a 3-dimensional analogue of a 2D conic (although in other ways it is a quadric surface which is the 3-dimensional analogue of a 2D conic.)

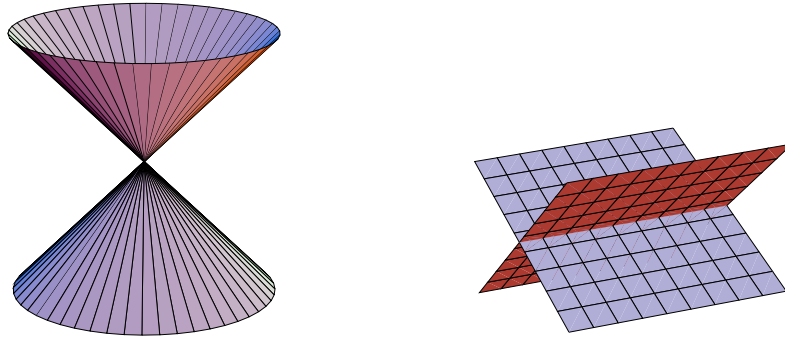


Fig. 3.4. **Degenerate quadrics.** The two most important degenerate quadrics are shown, the cone and two planes. Both these quadrics are ruled. The matrix representing the cone has rank 3, and the null-vector represents the nodal point of the cone. The matrix representing the two (non-coincident) planes has rank 2, and the two generators of the rank 2 null-space are two points on the intersection line of the planes.

A conic in the 2-dimensional projective plane may be described as a parametrized curve given by the equation

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = A \begin{pmatrix} 1 \\ \theta \\ \theta^2 \end{pmatrix} = \begin{pmatrix} a_{11} + a_{12}\theta + a_{13}\theta^2 \\ a_{21} + a_{22}\theta + a_{23}\theta^2 \\ a_{31} + a_{32}\theta + a_{33}\theta^2 \end{pmatrix} \quad (3.18)$$

where A is a non-singular 3×3 matrix.

In an analogous manner, a twisted cubic is defined to be a curve in \mathbb{P}^3 given in parametric form as

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = A \begin{pmatrix} 1 \\ \theta \\ \theta^2 \\ \theta^3 \end{pmatrix} = \begin{pmatrix} a_{11} + a_{12}\theta + a_{13}\theta^2 + a_{14}\theta^3 \\ a_{21} + a_{22}\theta + a_{23}\theta^2 + a_{24}\theta^3 \\ a_{31} + a_{32}\theta + a_{33}\theta^2 + a_{34}\theta^3 \\ a_{41} + a_{42}\theta + a_{43}\theta^2 + a_{44}\theta^3 \end{pmatrix} \quad (3.19)$$

where A is a non-singular 4×4 matrix.

Since a twisted cubic is perhaps an unfamiliar object, various views of the curve are shown in figure 3.5. In fact, a twisted cubic is a quite benign space curve.

Properties of a twisted cubic. Let c be a non-singular twisted cubic. Then c is not contained within any plane of \mathbb{P}^3 ; it intersects a general plane at three distinct points. A twisted cubic has 12 degrees of freedom (counted as 15 for the matrix A , less 3 for a 1D projectivity on the parametrization θ , which leaves the curve unaltered). Requiring the curve to pass through a point X places two constraints on c , since $X = A(1, \theta, \theta^2, \theta^3)^T$ is three independent ratios, but only two constraints once θ is eliminated. Thus, there is a unique c through six points in general position. Finally, all non-degenerate twisted cubics are projectively equivalent. This is clear from the definition (3.19): a projective transformation A^{-1} maps c to the standard form $c(\theta') = (1, \theta', \theta'^2, \theta'^3)^T$, and since

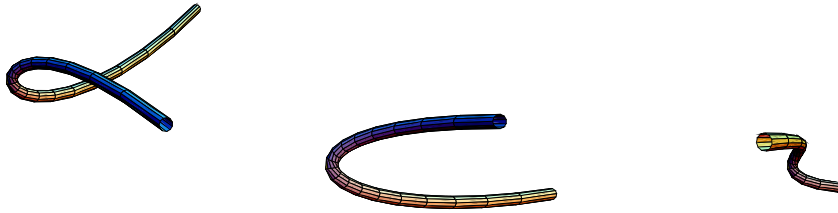


Fig. 3.5. Various views of the twisted cubic $(t^3, t^2, t)^T$. The curve is thickened to a tube to aid in visualization.

all twisted cubics can be mapped to this curve, it follows that all twisted cubics are projectively equivalent.

A classification of the various special cases of a twisted cubic, such as a conic and coincident line, are given in [Semple-79]. The twisted cubic makes an appearance as the horopter for two-view geometry (chapter 9), and plays the central role in defining the degenerate set for camera resectioning (chapter 22).

3.4 The hierarchy of transformations

There are a number of specializations of a projective transformation of 3-space which will appear frequently throughout this book. The specializations are analogous to the strata of section 2.4(p37) for planar transformations. Each specialization is a subgroup, and is identified by its matrix form, or equivalently by its invariants. These are summarized in table 3.2. This table lists only the *additional* properties of the 3-space transformations over their 2-space counterparts – the transformations of 3-space also have the invariants listed in table 2.1(p44) for the corresponding 2-space transformations.

The 15 degrees of freedom of a projective transformation are accounted for as seven for a similarity (three for rotation, three for translation, one for isotropic scaling), five for affine scalings, and three for the projective part of the transformation.

Two of the most important characterizations of these transformations are parallelism and angles. For example, after an affine transformation lines which were originally parallel remain parallel, but angles are skewed; and after a projective transformation parallelism is lost.

In the following we briefly describe a decomposition of a Euclidean transformation that will be useful when discussing special motions later in this book.

3.4.1 The screw decomposition

A Euclidean transformation on the plane may be considered as a specialization of a Euclidean transformation of 3-space with the restrictions that the translation vector t lies in the plane, and the rotation axis is perpendicular to the plane. However, Euclidean actions on 3-space are more general than this because the rotation axis and translation are not perpendicular in general. The screw decomposition enables any Euclidean

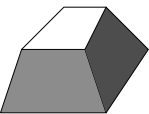
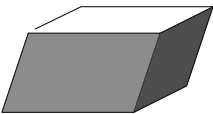

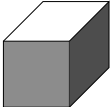
| Group | Matrix | Distortion | Invariant properties |
|----------------------|--|---|---|
| Projective 15 dof | $\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}$ |  | Intersection and tangency of surfaces in contact. Sign of Gaussian curvature. |
| Affine 12 dof | $\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$ |  | Parallelism of planes, volume ratios, centroids. The plane at infinity, π_∞ , (see section 3.5). |
| Similarity 7 dof | $\begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$ |  | The absolute conic, Ω_∞ , (see section 3.6). |
| Euclidean 6 dof | $\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$ |  | Volume. |

Table 3.2. **Geometric properties invariant to commonly occurring transformations of 3-space.** The matrix \mathbf{A} is an invertible 3×3 matrix, \mathbf{R} is a 3D rotation matrix, $\mathbf{t} = (t_X, t_Y, t_Z)^T$ a 3D translation, \mathbf{v} a general 3-vector, v a scalar, and $\mathbf{0} = (0, 0, 0)^T$ a null 3-vector. The distortion column shows typical effects of the transformations on a cube. 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 five points can be transformed to any other five points (provided no three points are collinear, or four coplanar).

action (a rotation composed with a translation) to be reduced to a situation almost as simple as the 2D case. The screw decomposition is that

Result 3.6. Any particular translation and rotation is equivalent to a rotation about a screw axis together with a translation along the screw axis. The screw axis is parallel to the rotation axis.

In the case of a translation and an *orthogonal* rotation axis (termed *planar motion*), the motion is equivalent to a rotation *alone* about the screw axis.

Proof. We will sketch a constructive geometric proof that can easily be visualized. Consider first the 2D case – a Euclidean transformation on the plane. It is evident from figure 3.6 that a screw axis exists for such 2D transformations. For the 3D case, decompose the translation \mathbf{t} into two components $\mathbf{t} = \mathbf{t}_\parallel + \mathbf{t}_\perp$, parallel and orthogonal respectively to the rotation axis direction ($\mathbf{t}_\parallel = (\mathbf{t} \cdot \mathbf{a})\mathbf{a}$, $\mathbf{t}_\perp = \mathbf{t} - (\mathbf{t} \cdot \mathbf{a})\mathbf{a}$).

Then the Euclidean motion is partitioned into two parts: first a rotation about the screw

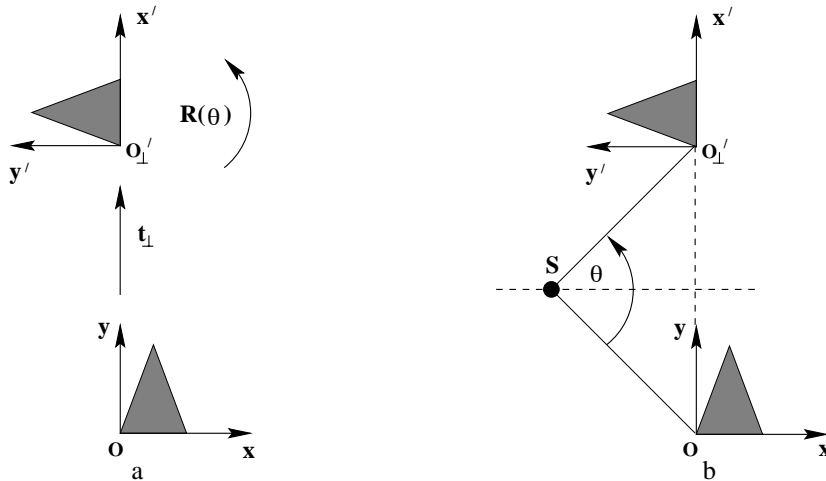


Fig. 3.6. **2D Euclidean motion and a “screw” axis.** (a) The frame $\{x, y\}$ undergoes a translation \mathbf{t}_\perp and a rotation by θ to reach the frame $\{x', y'\}$. The motion is in the plane orthogonal to the rotation axis. (b) This motion is equivalent to a single rotation about the screw axis S . The screw axis lies on the perpendicular bisector of the line joining corresponding points, such that the angle between the lines joining S to the corresponding points is θ . In the figure the corresponding points are the two frame origins and θ has the value 90° .

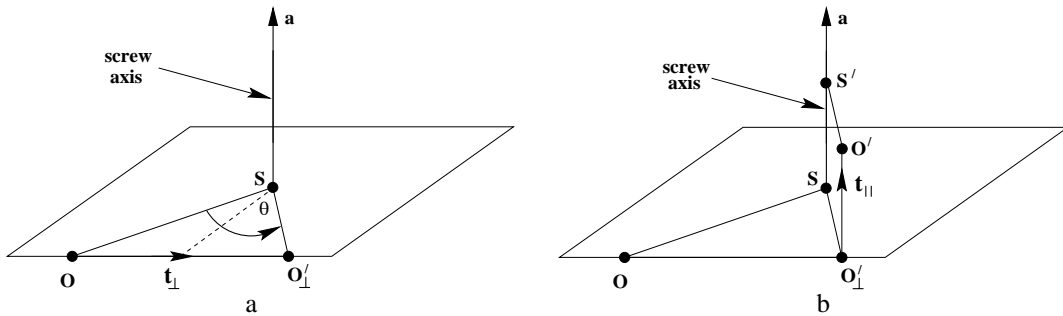


Fig. 3.7. **3D Euclidean motion and the screw decomposition.** Any Euclidean rotation R and translation \mathbf{t} may be achieved by (a) a rotation about the screw axis, followed by (b) a translation along the screw axis by \mathbf{t}_\parallel . Here \mathbf{a} is the (unit) direction of the rotation axis (so that $R\mathbf{a} = \mathbf{a}$), and \mathbf{t} is decomposed as $\mathbf{t} = \mathbf{t}_\parallel + \mathbf{t}_\perp$, which are vector components parallel and orthogonal respectively to the rotation axis direction. The point S is closest to O on the screw axis (so that the line from S to O is perpendicular to the direction of \mathbf{a}). Similarly S' is the point on the screw axis closest to O' .

axis, which covers the rotation and \mathbf{t}_\perp ; second a translation by \mathbf{t}_\parallel along the screw axis. The complete motion is illustrated in figure 3.7. \square

The screw decomposition can be determined from the fixed points of the 4×4 matrix representing the Euclidean transformation. This idea is examined in the exercises at the end of the chapter.

3.5 The plane at infinity

In planar projective geometry identifying the line at infinity, l_∞ , allowed affine properties of the plane to be measured. Identifying the circular points on l_∞ then allowed

the measurement of metric properties. In the projective geometry of 3-space the corresponding geometric entities are the plane at infinity, π_∞ , and the absolute conic, Ω_∞ .

The plane at infinity has the canonical position $\pi_\infty = (0, 0, 0, 1)^T$ in affine 3-space. It contains the directions $\mathbf{D} = (x_1, x_2, x_3, 0)^T$, and enables the identification of affine properties such as parallelism. In particular:

- Two planes are parallel if, and only if, their line of intersection is on π_∞ .
- A line is parallel to another line, or to a plane, if the point of intersection is on π_∞ .

We then have in \mathbb{P}^3 that any pair of planes intersect in a line, with parallel planes intersecting in a line on the plane at infinity.

The plane π_∞ is a geometric representation of the 3 degrees of freedom required to specify affine properties in a projective coordinate frame. In loose terms, the plane at infinity is a fixed plane under any affine transformation, but “sees” (is moved by) a projective transformation. The 3 degrees of freedom of π_∞ thus measure the projective component of a general homography – they account for the 15 degrees of freedom of this general transformation compared to an affinity (12 dof). More formally:

Result 3.7. *The plane at infinity, π_∞ , is a fixed plane under the projective transformation H if, and only if, H is an affinity.*

The proof is the analogue of the derivation of result 2.17(p48). It is worth clarifying two points:

- (i) The plane π_∞ is, in general, only fixed as a set under an affinity; it is not fixed pointwise.
- (ii) Under a particular affinity (for example a Euclidean motion) there may be planes in addition to π_∞ which are fixed. However, only π_∞ is fixed under any affinity.

These points are illustrated in more detail by the following example.

Example 3.8. Consider the Euclidean transformation represented by the matrix

$$H_E = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.20)$$

This is a rotation by θ about the Z-axis with a zero translation (it is a planar screw motion, see section 3.4.1). Geometrically it is evident that the family of XY-planes orthogonal to the rotation axis are simply rotated about the Z-axis by this transformation. This means that there is a pencil of fixed planes orthogonal to the Z-axis. The planes are fixed as sets, but not pointwise as any (finite) point (not on the axis) is rotated in horizontal circles by this Euclidean action. Algebraically, the fixed planes of H are the eigenvectors of H^T (refer to section 2.9). In this case the eigenvalues are $\{e^{i\theta}, e^{-i\theta}, 1, 1\}$

and the corresponding eigenvectors of H_E^T are

$$\mathbf{E}_1 = \begin{pmatrix} 1 \\ i \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{E}_2 = \begin{pmatrix} 1 \\ -i \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{E}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{E}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

The eigenvectors \mathbf{E}_1 and \mathbf{E}_2 do not correspond to real planes, and will not be discussed further here. The eigenvectors \mathbf{E}_3 and \mathbf{E}_4 are degenerate. Thus there is a pencil of fixed planes which is spanned by these eigenvectors. The axis of this pencil is the line of intersection of the the planes (perpendicular to the Z-axis) with π_∞ , and the pencil includes π_∞ . \triangle

The example also illustrates the connection between the geometry of the projective plane, \mathbb{P}^2 , and projective 3-space, \mathbb{P}^3 . A plane π intersects π_∞ in a line which is the line at infinity, l_∞ , of the plane π . A projective transformation of \mathbb{P}^3 induces a *subordinate* plane projective transformation on π .

Affine properties of a reconstruction. In later chapters on reconstruction, for example chapter 10, it will be seen that the projective coordinates of the (Euclidean) scene can be reconstructed from multiple views. Once π_∞ is identified in projective 3-space, i.e. its projective coordinates are known, it is then possible to determine affine properties of the reconstruction such as whether geometric entities are parallel – they are parallel if they intersect on π_∞ .

A more algorithmic approach is to transform \mathbb{P}^3 so that the identified π_∞ is moved to its canonical position at $\pi_\infty = (0, 0, 0, 1)^T$. After this mapping we then have the situation that the Euclidean scene, where π_∞ has the coordinates $(0, 0, 0, 1)^T$, and our reconstruction are related by a projective transformation that fixes π_∞ at $(0, 0, 0, 1)^T$. It follows from result 3.7 that the scene and reconstruction are related by an affine transformation. Thus affine properties can now be measured directly from the coordinates of the entities.

3.6 The absolute conic

The absolute conic, Ω_∞ , is a (point) conic on π_∞ . In a metric frame $\pi_\infty = (0, 0, 0, 1)^T$, and points on Ω_∞ satisfy

$$\left. \begin{array}{l} x_1^2 + x_2^2 + x_3^2 \\ x_4 \end{array} \right\} = 0. \quad (3.21)$$

Note that two equations are required to define Ω_∞ .

For directions on π_∞ (i.e. points with $x_4 = 0$) the defining equation can be written

$$(x_1, x_2, x_3) I (x_1, x_2, x_3)^T = 0$$

so that Ω_∞ corresponds to a conic C with matrix $C = I$. It is thus a conic of purely imaginary points on π_∞ .

The conic Ω_∞ is a geometric representation of the 5 additional degrees of freedom that are required to specify metric properties in an affine coordinate frame. A key property of Ω_∞ is that it is a fixed conic under any similarity transformation. More formally:

Result 3.9. *The absolute conic, Ω_∞ , is a fixed conic under the projective transformation H if, and only if, H is a similarity transformation.*

Proof. Since the absolute conic lies in the plane at infinity, a transformation fixing it must fix the plane at infinity, and hence must be affine. Such a transformation is of the form

$$H_A = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}.$$

Restricting to the plane at infinity, the absolute conic is represented by the matrix $I_{3 \times 3}$, and since it is fixed by H_A , one has $A^{-T} I A^{-1} = I$ (up to scale), and taking inverses gives $A A^T = I$. This means that A is orthogonal, hence a scaled rotation, or scaled rotation with reflection. This completes the proof. \square

Even though Ω_∞ does not have any real points, it shares the properties of any conic – such as that a line intersects a conic in two points; the pole–polar relationship etc. Here are a few particular properties of Ω_∞ :

- (i) Ω_∞ is only fixed as a set by a general similarity; it is not fixed pointwise. This means that under a similarity a point on Ω_∞ may travel to another point on Ω_∞ , but it is not mapped to a point off the conic.
- (ii) All circles intersect Ω_∞ in two points. Suppose the support plane of the circle is π . Then π intersects π_∞ in a line, and this line intersects Ω_∞ in two points. These two points are the circular points of π .
- (iii) All spheres intersect π_∞ in Ω_∞ .

Metric properties. Once Ω_∞ (and its support plane π_∞) have been identified in projective 3-space then metric properties, such as angles and relative lengths, can be measured.

Consider two lines with directions (3-vectors) d_1 and d_2 . The angle between these directions in a Euclidean world frame is given by

$$\cos \theta = \frac{(d_1^T d_2)}{\sqrt{(d_1^T d_1)(d_2^T d_2)}}. \quad (3.22)$$

This may be written as

$$\cos \theta = \frac{(d_1^T \Omega_\infty d_2)}{\sqrt{(d_1^T \Omega_\infty d_1)(d_2^T \Omega_\infty d_2)}} \quad (3.23)$$

where d_1 and d_2 are the points of intersection of the lines with the plane π_∞ containing the conic Ω_∞ , and Ω_∞ is the matrix representation of the absolute conic in that plane.

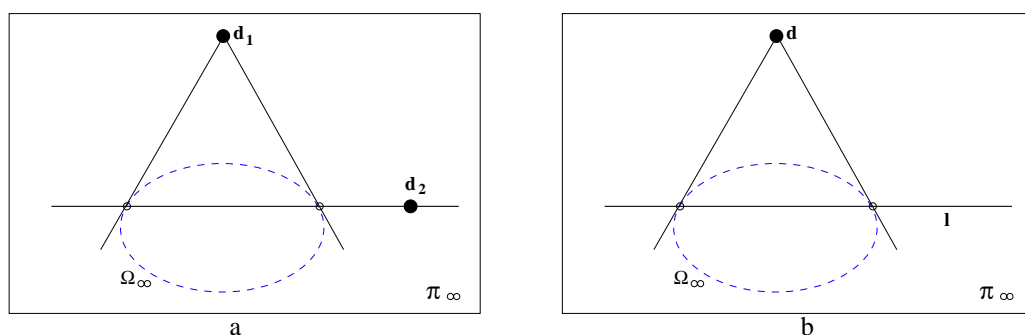


Fig. 3.8. **Orthogonality and Ω_∞ .** (a) On π_∞ orthogonal directions d_1 , d_2 are conjugate with respect to Ω_∞ . (b) A plane normal direction d and the intersection line l of the plane with π_∞ are in pole–polar relation with respect to Ω_∞ .

The expression (3.23) reduces to (3.22) in a Euclidean world frame where $\Omega_\infty = I$. However, the expression is valid in any projective coordinate frame as may be verified from the transformation properties of points and conics (see (iv)(b) on page 63).

There is no simple formula for the angle between two planes computed from the directions of their surface normals.

Orthogonality and polarity. We now give a geometric representation of orthogonality in a projective space based on the absolute conic. The main device will be the pole–polar relationship between a point and line induced by a conic.

An immediate consequence of (3.23) is that two directions d_1 and d_2 are orthogonal if $d_1^T \Omega_\infty d_2 = 0$. Thus orthogonality is encoded by *conjugacy* with respect to Ω_∞ . The great advantage of this is that conjugacy is a projective relation, so that in a projective frame (obtained by a projective transformation of Euclidean 3-space) directions can be identified as orthogonal if they are conjugate with respect to Ω_∞ in that frame (in general the matrix of Ω_∞ is not I in a projective frame). The geometric representation of orthogonality is shown in figure 3.8.

This representation is helpful when considering orthogonality between rays in a camera, for example in determining the normal to a plane through the camera centre (see section 8.6(p213)). If image points are conjugate with respect to the *image* of Ω_∞ then the corresponding rays are orthogonal.

Again, a more algorithmic approach is to projectively transform the coordinates so that Ω_∞ is mapped to its canonical position (3.21), and then metric properties can be determined directly from coordinates.

3.7 The absolute dual quadric

Recall that Ω_∞ is defined by *two* equations – it is a conic on the plane at infinity. The dual of the absolute conic Ω_∞ is a degenerate dual *quadric* in 3-space called the *absolute dual quadric*, and denoted Q_∞^* . Geometrically Q_∞^* consists of the planes tangent to Ω_∞ , so that Ω_∞ is the “rim” of Q_∞^* . This is called a *rim quadric*. Think of the set of planes tangent to an ellipsoid, and then squash the ellipsoid to a pancake.

Algebraically Q_∞^* is represented by a 4×4 homogeneous matrix of rank 3, which in

metric 3-space has the canonical form

$$Q_{\infty}^* = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}. \quad (3.24)$$

We will show that any plane in the dual absolute quadric envelope is indeed tangent to Ω_{∞} , so the Q_{∞}^* is truly a dual of Ω_{∞} . Consider a plane represented by $\pi = (\mathbf{v}^T, k)^T$. This plane is in the envelope defined by Q_{∞}^* if and only if $\pi^T Q_{\infty}^* \pi = 0$, which given the form (3.24) is equivalent to $\mathbf{v}^T \mathbf{v} = 0$. Now, (see section 8.6(p213)) \mathbf{v} represents the line in which the plane $(\mathbf{v}^T, k)^T$ meets the plane at infinity. This line is tangent to the absolute conic if and only if $\mathbf{v}^T I \mathbf{v} = 0$. Thus, the envelope of Q_{∞}^* is made up of just those planes tangent to the absolute conic.

Since this is an important fact, we consider it from another angle. Consider the absolute conic as the limit of a series of squashed ellipsoids, namely quadrics represented by the matrix $Q = \text{diag}(1, 1, 1, k)$. As $k \rightarrow \infty$, these quadrics become increasingly close to the plane at infinity, and in the limit the only points they contain are the points $(x_1, x_2, x_3, 0)^T$ with $x_1^2 + x_2^2 + x_3^2 = 0$, that is points on the absolute conic. However, the dual of Q is the quadric $Q^* = Q^{-1} = \text{diag}(1, 1, 1, k^{-1})$, which in the limit becomes the absolute dual quadric $Q_{\infty}^* = \text{diag}(1, 1, 1, 0)$.

The dual quadric Q_{∞}^* is a degenerate quadric and has 8 degrees of freedom (a symmetric matrix has 10 independent elements, but the irrelevant scale and zero determinant condition each reduce the degrees of freedom by 1). It is a geometric representation of the 8 degrees of freedom that are required to specify metric properties in a projective coordinate frame. Q_{∞}^* has a significant advantage over Ω_{∞} in algebraic manipulations because both π_{∞} and Ω_{∞} are contained in a single geometric object (unlike Ω_{∞} which requires two equations (3.21) in order to specify it). In the following we give its three most important properties.

Result 3.10. *The absolute dual quadric, Q_{∞}^* , is fixed under the projective transformation H if, and only if, H is a similarity.*

Proof. This follows directly from the invariance of the absolute conic under a similarity transform, since the planar tangency relationship between Q_{∞}^* and Ω_{∞} is transformation invariant. Nevertheless, we give an independent direct proof.

Since Q_{∞}^* is a dual quadric, it transforms according to (3.17–p74), so it is fixed under H if and only if $Q_{\infty}^* = H Q_{\infty}^* H^T$. Applying this with an arbitrary transform

$$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & k \end{bmatrix}$$

we find

$$\begin{aligned} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} &= \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & k \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} A^T & \mathbf{v} \\ \mathbf{t}^T & k \end{bmatrix} \\ &= \begin{bmatrix} AA^T & A\mathbf{v} \\ \mathbf{v}^T A^T & \mathbf{v}^T \mathbf{v} \end{bmatrix} \end{aligned}$$

which must be true up to scale. By inspection, this equation holds if and only if $\mathbf{v} = \mathbf{0}$ and \mathbf{A} is a scaled orthogonal matrix (scaling, rotation and possible reflection). In other words, \mathbf{H} is a similarity transform. \square

Result 3.11. *The plane at infinity π_∞ is the null-vector of \mathbf{Q}_∞^* .*

This is easily verified when \mathbf{Q}_∞^* has its canonical form (3.24) in a metric frame since then, with $\pi_\infty = (0, 0, 0, 1)^T$, $\mathbf{Q}_\infty^* \pi_\infty = \mathbf{0}$. This property holds in any frame as may be readily seen algebraically from the transformation properties of planes and dual quadrics: if $\mathbf{X}' = \mathbf{H}\mathbf{X}$, then $\mathbf{Q}_\infty^{*'} = \mathbf{H} \mathbf{Q}_\infty^* \mathbf{H}^T$, $\pi_\infty' = \mathbf{H}^{-T} \pi_\infty$, and

$$\mathbf{Q}_\infty^{*'} \pi_\infty' = (\mathbf{H} \mathbf{Q}_\infty^* \mathbf{H}^T) \mathbf{H}^{-T} \pi_\infty = \mathbf{H} \mathbf{Q}_\infty^* \pi_\infty = \mathbf{0}.$$

Result 3.12. *The angle between two planes π_1 and π_2 is given by*

$$\cos \theta = \frac{\pi_1^T \mathbf{Q}_\infty^* \pi_2}{\sqrt{(\pi_1^T \mathbf{Q}_\infty^* \pi_1) (\pi_2^T \mathbf{Q}_\infty^* \pi_2)}}. \quad (3.25)$$

Proof. Consider two planes with Euclidean coordinates $\pi_1 = (\mathbf{n}_1^T, d_1)^T$, $\pi_2 = (\mathbf{n}_2^T, d_2)^T$. In a Euclidean frame, \mathbf{Q}_∞^* has the form (3.24), and (3.25) reduces to

$$\cos \theta = \frac{\mathbf{n}_1^T \mathbf{n}_2}{\sqrt{(\mathbf{n}_1^T \mathbf{n}_1) (\mathbf{n}_2^T \mathbf{n}_2)}}$$

which is the angle between the planes expressed in terms of a scalar product of their normals.

If the planes and \mathbf{Q}_∞^* are projectively transformed, (3.25) will still determine the angle between planes due to the (covariant) transformation properties of planes and dual quadrics. \square

The details of the last part of the proof are left as an exercise, but are a direct 3D analogue of the derivation of result 2.23(p54) on the angle between two lines in \mathbb{P}^2 computed using the dual of the circular points. Planes in \mathbb{P}^3 are the analogue of lines in \mathbb{P}^2 , and the absolute dual quadric is the analogue of the dual of the circular points.

3.8 Closure

3.8.1 The literature

The textbooks cited in chapter 2 are also relevant here. See also [Boehm-94] for a general background from the perspective of descriptive geometry, and Hilbert and Cohn-Vossen [Hilbert-56] for many clearly explained properties of curves and surfaces.

An important representation for points, lines and planes in \mathbb{P}^3 , which is omitted in this chapter, is the Grassmann–Cayley algebra. In this representation geometric operations such as incidence and joins are represented by a “bracket algebra” based on matrix determinants. A good introduction to this area is given by [Carlsson-94], and its application to multiple view tensors is illustrated in [Triggs-95].

Faugeras and Maybank [Faugeras-90] introduced Ω_∞ into the computer vision literature (in order to determine the multiplicity of solutions for relative orientation), and Triggs introduced Q_∞^* in [Triggs-97] for use in auto-calibration.

3.8.2 Notes and exercises

(i) Plücker coordinates.

- (a) Using Plücker line coordinates, \mathcal{L} , write an expression for the point of intersection of a line with a plane, and the plane defined by a point and a line.
- (b) Now derive the condition for a point to be on a line, and a line to be on a plane.
- (c) Show that parallel planes intersect in a line on π_∞ . Hint, start from (3.9–p71) to determine the line of intersection of two parallel planes L^* .
- (d) Show that parallel lines intersect on π_∞ .

(ii) Projective transformations. Show that a (real) projective transformation of 3-space can map an ellipsoid to a paraboloid or hyperboloid of two sheets, but cannot map an ellipsoid to a hyperboloid of one sheet (i.e. a surface with real rulings).

(iii) Screw decomposition. Show that the 4×4 matrix representing the Euclidean transformation $\{R, t\}$ (with a the direction of the rotation axis, i.e. $Ra = a$) has two complex conjugate eigenvalues, and two equal real eigenvalues, and the following eigenvector structure:

- (a) if a is perpendicular to t , then the eigenvectors corresponding to the real eigenvalues are distinct;
- (b) otherwise, the eigenvectors corresponding to the real eigenvalues are coincident, and on π_∞ .

(E.g. choose simple cases such as (3.20), another case is given on page 495). In the first case the two real points corresponding to the real eigenvalues define a line of fixed points. This is the screw axis for planar motion. In the second case, the direction of the screw axis is defined, but it is not a line of fixed points. What do the eigenvectors corresponding to the complex eigenvalues represent?

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 \mathbf{F} consistent with these correspondences. The fundamental matrix, discussed in chapter 9, is a singular 3×3 matrix \mathbf{F} satisfying $\mathbf{x}'_i{}^T \mathbf{F} \mathbf{x}_i = 0$ 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^{jkl} 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 \mathbf{H} such that $\mathbf{H}\mathbf{x}_i = \mathbf{x}'_i$ 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 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 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 $H\mathbf{x}_i$. 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 H fully.

Approximate solutions. It will be seen that if exactly four correspondences are given, then an exact solution for the matrix 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 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 function: 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 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 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 $\mathbf{x}'_i = H\mathbf{x}_i$. Note that this is an equation involving homogeneous vectors; thus the 3-vectors \mathbf{x}'_i and $H\mathbf{x}_i$ 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 $\mathbf{x}'_i \times H\mathbf{x}_i = \mathbf{0}$. This form will enable a simple linear solution for H to be derived.

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

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

Writing $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top$, the cross product may then be given explicitly as

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}^{3\top}\mathbf{x}_i - w'_i \mathbf{h}^{2\top}\mathbf{x}_i \\ w'_i \mathbf{h}^{1\top}\mathbf{x}_i - x'_i \mathbf{h}^{3\top}\mathbf{x}_i \\ x'_i \mathbf{h}^{2\top}\mathbf{x}_i - y'_i \mathbf{h}^{1\top}\mathbf{x}_i \end{pmatrix}.$$

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

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{x}_i^\top \\ -y'_i \mathbf{x}_i^\top & x'_i \mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \quad (4.1)$$

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

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

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

- (i) The equation $A_i \mathbf{h} = \mathbf{0}$ is an equation *linear* in the unknown \mathbf{h} . The matrix elements of 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 H . It is usual to omit the third equation in solving for H ([Sutherland-63]). Then (for future reference) the set of equations becomes

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{x}_i^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \quad (4.3)$$

This will be written

$$A_i \mathbf{h} = \mathbf{0}$$

where A_i is now the 2×9 matrix of (4.3).

- (iii) The equations hold for any homogeneous coordinate representation $(x'_i, y'_i, w'_i)^\top$ 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 $A\mathbf{h} = \mathbf{0}$, where A is the matrix of equation coefficients built from the matrix rows 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 $A\mathbf{h} = \mathbf{0}$ 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 $A\mathbf{h} = \mathbf{0}$ 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 $A\mathbf{h} = \mathbf{0}$, it seems natural to attempt to minimize the norm $\|A\mathbf{h}\|$ instead, subject to the usual constraint, $\|\mathbf{h}\| = 1$. This is identical to the problem of finding the minimum of the quotient $\|A\mathbf{h}\|/\|\mathbf{h}\|$. As shown in section A5.3(p592) the solution is the (unit) eigenvector of $A^T A$ 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 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

Objective

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

Algorithm

- (i) For each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ compute the matrix \mathbf{A}_i from (4.1). Only the first two rows need be used in general.
- (ii) Assemble the $n \times 9$ matrices \mathbf{A}_i into a single $2n \times 9$ matrix \mathbf{A} .
- (iii) Obtain the SVD of \mathbf{A} (section A4.4(p585)). The unit singular vector corresponding to the smallest singular value is the solution \mathbf{h} . Specifically, if $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ with \mathbf{D} diagonal with positive diagonal entries, arranged in descending order down the diagonal, then \mathbf{h} is the last column of \mathbf{V} .
- (iv) The matrix \mathbf{H} is determined from \mathbf{h} as in (4.2).

Algorithm 4.1. *The basic DLT for \mathbf{H} (but see algorithm 4.2(p109) which includes normalization).*

$\tilde{\mathbf{M}}\tilde{\mathbf{h}} = \mathbf{b}$, where \mathbf{M} has 8 columns and \mathbf{b} is an 8-vector. Such an equation may be solved for $\tilde{\mathbf{h}}$ using standard techniques for solving linear equations (such as Gaussian elimination) in the case where \mathbf{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 $h_9 = H_{33}$ is zero if the coordinate origin is mapped to a point at infinity by \mathbf{H} . Since $(0, 0, 1)^T$ represents the coordinate origin \mathbf{x}_0 , and also $(0, 0, 1)^T$ represents the line at infinity \mathbf{l} , this condition may be written as $\mathbf{l}^T \mathbf{H} \mathbf{x}_0 = (0, 0, 1) \mathbf{H} (0, 0, 1)^T = 0$, thus $H_{33} = 0$. 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 $H_{33} = h_9 = 0$. Consequently, an $h_9 = 1$ normalization can be a serious failing in practical situations. \triangle

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 \mathbf{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 H . How is this apparent contradiction to be resolved?

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

We showed that if $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are collinear then $H^* = \mathbf{x}'_4 l^\top$ is a solution to (4.1). There are two cases: either H^* is the unique solution (up to scale) or there is a further solution H . In the first case, since H^* is a singular matrix, there exists no transformation taking each \mathbf{x}_i to \mathbf{x}'_i . This occurs when $\mathbf{x}_1, \dots, \mathbf{x}_3$ are collinear but $\mathbf{x}'_1, \dots, \mathbf{x}'_3$ are not. In the second case, where a further solution H exists, then any matrix of the form $\alpha H^* + \beta H$ 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 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 $l_i = H^\top l'_i$, a matrix equation of the form $A\mathbf{h} = \mathbf{0}$ 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 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 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.

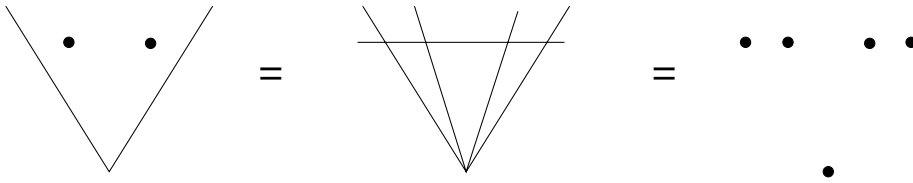


Fig. 4.1. **Geometric equivalence of point–line configurations.** A configuration of two points and two lines is equivalent to five lines with four concurrent, or five points with four collinear.

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 $\|Ah\|$. The vector $\epsilon = Ah$ 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 A . Each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ contributes a partial error vector ϵ_i from (4.1) or (4.3) towards the full error vector ϵ . This vector ϵ_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*:

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

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

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

The relation of this distance to a geometric distance is described in section 4.2.4.

Given a set of correspondences, the quantity $\epsilon = \mathbf{A}\mathbf{h}$ is the algebraic error vector for the complete set, and one sees that

$$\sum_i d_{\text{alg}}(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2 = \sum_i \|\epsilon_i\|^2 = \|\mathbf{A}\mathbf{h}\|^2 = \|\epsilon\|^2. \quad (4.5)$$

The concept of algebraic distance originated in the conic-fitting work of Bookstein [Bookstein-79]. Its disadvantage is that the quantity that is minimized is not geometrically or statistically meaningful. As Bookstein demonstrated, the solutions that minimize algebraic distance may not be those expected intuitively. Nevertheless, with a good choice of normalization (as will be discussed in section 4.4) methods which minimize algebraic distance do give very good results. Their particular advantages are a linear (and thus a unique) solution, and computational cheapness. Often solutions based on algebraic distance are used as a starting point for a non-linear minimization of a geometric or statistical cost function. The non-linear minimization gives the solution a final “polish”.

4.2.2 Geometric distance

Next we discuss alternative error functions based on the measurement of geometric distance in the image, and minimization of the difference between the measured and estimated image coordinates.

Notation. Vectors \mathbf{x} represent the *measured* image coordinates; $\hat{\mathbf{x}}$ represent estimated values of the points and $\bar{\mathbf{x}}$ represent true values of the points.

Error in one image. We start by considering error only in the second image, with points in the first measured perfectly. Clearly, this will not be true in most practical situations with images. An example where the assumption is more reasonable is in estimating the projective transformation between a calibration pattern or a world plane, where points are measured to a very high accuracy, and its image. The appropriate quantity to be minimized is the *transfer* error. This is the Euclidean image distance in the second image between the measured point \mathbf{x}' and the point $\mathbf{H}\bar{\mathbf{x}}$ at which the corresponding point $\bar{\mathbf{x}}$ is mapped from the first image. We use the notation $d(\mathbf{x}, \mathbf{y})$ to represent the Euclidean distance between the inhomogeneous points represented by \mathbf{x} and \mathbf{y} . Then the transfer error for the set of correspondences is

$$\sum_i d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2. \quad (4.6)$$

The estimated homography $\hat{\mathbf{H}}$ is the one for which the error (4.6) is minimized.

Symmetric transfer error. In the more realistic case where image measurement errors occur in both the images, it is preferable that errors be minimized in both images, and not solely in the one. One way of constructing a more satisfactory error function is to

consider the forward (H) and backward (H^{-1}) transformation, and sum the geometric errors corresponding to each of these two transformations. Thus, the error is

$$\sum_i d(\mathbf{x}_i, H^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, H\mathbf{x}_i)^2. \quad (4.7)$$

The first term in this sum is the transfer error in the first image, and the second term is the transfer error in the second image. Again the estimated homography \hat{H} is the one for which (4.7) is minimized.

4.2.3 Reprojection error – both images

An alternative method of quantifying error in each of the two images involves estimating a “correction” for each correspondence. One asks how much it is necessary to correct the measurements in each of the two images in order to obtain a perfectly matched set of image points. One should compare this with the geometric one-image transfer error (4.6) which measures the correction that it is necessary to make to the measurements in one image (the second image) in order to get a set of perfectly matching points.

In the present case, we are seeking a homography \hat{H} and pairs of *perfectly* matched points $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ that minimize the total error function

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \quad \text{subject to } \hat{\mathbf{x}}'_i = \hat{H}\hat{\mathbf{x}}_i \quad \forall i. \quad (4.8)$$

Minimizing this cost function involves determining both \hat{H} and a set of subsidiary correspondences $\{\hat{\mathbf{x}}_i\}$ and $\{\hat{\mathbf{x}}'_i\}$. This estimation models, for example, the situation that measured correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ arise from images of points on a world plane. We wish to estimate a point on the world plane $\hat{\mathbf{X}}_i$ from $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ which is then *reprojected* to the estimated perfectly matched correspondence $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$.

This reprojection error function is compared with the symmetric error function in figure 4.2. It will be seen in section 4.3 that (4.8) is related to the Maximum Likelihood estimation of the homography and correspondences.

4.2.4 Comparison of geometric and algebraic distance

We return to the case of errors only in the second image. Let $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top$ and define a vector $(\hat{x}'_i, \hat{y}'_i, \hat{w}'_i)^\top = \hat{\mathbf{x}}'_i = H\hat{\mathbf{x}}_i$. Using this notation, the left hand side of (4.3) becomes

$$\mathbf{A}_i \mathbf{h} = \boldsymbol{\epsilon}_i = \begin{pmatrix} y'_i \hat{w}'_i - w'_i \hat{y}'_i \\ w'_i \hat{x}'_i - x'_i \hat{w}'_i \end{pmatrix}.$$

This vector is the *algebraic error vector* associated with the point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and the camera mapping H . Thus,

$$d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 = (y'_i \hat{w}'_i - w'_i \hat{y}'_i)^2 + (w'_i \hat{x}'_i - x'_i \hat{w}'_i)^2.$$

For points \mathbf{x}'_i and $\hat{\mathbf{x}}'_i$ the geometric distance is

$$d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i) = \left((x'_i/w'_i - \hat{x}'_i/\hat{w}'_i)^2 + (y'_i/w'_i - \hat{y}'_i/\hat{w}'_i)^2 \right)^{1/2}$$

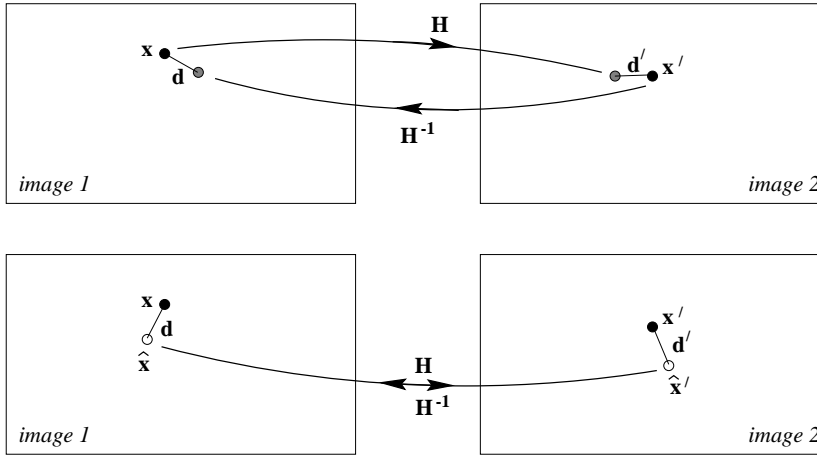


Fig. 4.2. A comparison between symmetric transfer error (upper) and reprojection error (lower) when estimating a homography. The points \mathbf{x} and \mathbf{x}' are the measured (noisy) points. Under the estimated homography the points \mathbf{x}' and $H\mathbf{x}$ do not correspond perfectly (and neither do the points \mathbf{x} and $H^{-1}\mathbf{x}'$). However, the estimated points, $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$, do correspond perfectly by the homography $\hat{\mathbf{x}}' = H\hat{\mathbf{x}}$. Using the notation $d(\mathbf{x}, \mathbf{y})$ for the Euclidean image distance between \mathbf{x} and \mathbf{y} , the symmetric transfer error is $d(\mathbf{x}, H^{-1}\mathbf{x}')^2 + d(\mathbf{x}', H\mathbf{x})^2$; the reprojection error is $d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$.

$$= d_{\text{alg}}(\mathbf{x}'_i, \hat{\mathbf{x}}'_i) / \hat{w}'_i w'_i.$$

Thus, geometric distance is related to, but not quite the same as, algebraic distance. Note, though, that if $\hat{w}'_i = w'_i = 1$, then the two distances are identical.

One can always assume that $w_i = 1$, thus expressing the points \mathbf{x}_i in the usual form $\mathbf{x}_i = (x_i, y_i, 1)^T$. For one important class of 2D homographies, the values of \hat{w}'_i will always be 1 as well. A 2D *affine* transformation is represented by a matrix of the form (2.10–p39)

$$H_A = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.9)$$

One verifies immediately from $\hat{\mathbf{x}}'_i = H_A \bar{\mathbf{x}}_i$ that $\hat{w}'_i = 1$ if $w_i = 1$. This demonstrates that in the case of an affine transformation geometric distance and algebraic distance are identical. The DLT algorithm is easily adapted to enforce the condition that the last row of H has the form $(0, 0, 1)$ by setting $h_7 = h_8 = 0$. Hence, for affine transformations, geometric distance can be minimized by the linear DLT algorithm based on algebraic distance.

4.2.5 Geometric interpretation of reprojection error

The estimation of a homography between two planes can be thought of as fitting a “surface” to points in a 4D space, \mathbb{R}^4 . Each pair of image points \mathbf{x}, \mathbf{x}' defines a single point denoted \mathbf{X} in a measurement space \mathbb{R}^4 , formed by concatenating the inhomogeneous coordinates of \mathbf{x} and \mathbf{x}' . For a given specific homography H , the image correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$ that satisfy $\mathbf{x}' \times (H\mathbf{x}) = \mathbf{0}$ define an algebraic variety¹ \mathcal{V}_H in \mathbb{R}^4 which is the

¹ A *variety* is the simultaneous zero-set of one or more multivariate polynomials defined in \mathbb{R}^N .

intersection of two quadric hypersurfaces. The surface is a quadric in \mathbb{R}^4 because each row of (4.1) is a degree 2 polynomial in x, y, x', y' . The elements of H determine the coefficient of each term of the polynomial, and so H specifies the particular quadric. The two independent equations of (4.1) define two such quadrics.

Given points $\mathbf{X}_i = (x_i, y_i, x'_i, y'_i)^T$ in \mathbb{R}^4 , the task of estimating a homography becomes the task of finding a variety \mathcal{V}_H that passes (or most nearly passes) through the points \mathbf{X}_i . In general, of course, it will not be possible to fit a variety precisely. In this case, let \mathcal{V}_H be some variety corresponding to a transformation H , and for each point \mathbf{X}_i , let $\hat{\mathbf{X}}_i = (\hat{x}_i, \hat{y}_i, \hat{x}'_i, \hat{y}'_i)^T$ be the closest point to \mathbf{X}_i lying on the variety \mathcal{V}_H . One sees immediately that

$$\begin{aligned} \|\mathbf{X}_i - \hat{\mathbf{X}}_i\|^2 &= (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (x'_i - \hat{x}'_i)^2 + (y'_i - \hat{y}'_i)^2 \\ &= d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2. \end{aligned}$$

Thus geometric distance in \mathbb{R}^4 is equivalent to the reprojection error measured in both the images, and finding the variety \mathcal{V}_H and points $\hat{\mathbf{X}}_i$ on \mathcal{V}_H that minimize the squared sum of distances to the measured points \mathbf{X}_i is equivalent to finding the homography \hat{H} and the estimated points $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ that minimize the reprojection error function (4.8).

The point $\hat{\mathbf{X}}$ on \mathcal{V}_H that lies closest to a measured point \mathbf{X} is a point where the line between \mathbf{X} and $\hat{\mathbf{X}}$ is perpendicular to the tangent plane to \mathcal{V}_H at $\hat{\mathbf{X}}$. Thus

$$d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 = d_\perp(\mathbf{X}_i, \mathcal{V}_H)^2$$

where $d_\perp(\mathbf{X}, \mathcal{V}_H)$ is the perpendicular distance of the point \mathbf{X} to the variety \mathcal{V}_H . As may be seen from the conic-fitting analogue discussed below, there may be more than one such perpendicular from \mathbf{X} to \mathcal{V}_H .

The distance $d_\perp(\mathbf{X}, \mathcal{V}_H)$ is invariant to rigid transformations of \mathbb{R}^4 , and this includes as a special case rigid transformations of the coordinates $(x, y), (x', y')$ of each image individually. This point is returned to in section 4.4.3.

Conic analogue. Before proceeding further we will first sketch an analogous estimation problem that can be visualized more easily. The problem is fitting a conic to 2D points, which occupies a useful intermediate position between fitting a straight line (no curvature, too simple) and fitting a homography (four dimensions, with non-zero curvature).

Consider the problem of fitting a conic to a set of $n > 5$ points $(x_i, y_i)^T$ on the plane such that an error based on geometric distance is minimized. The points may be thought of as “correspondences” $x_i \leftrightarrow y_i$. The transfer distance and reprojection (perpendicular) distance are illustrated in figure 4.3. It is clear from this figure that d_\perp is less than or equal to the transfer error.

The algebraic distance of a point \mathbf{x} from a conic C is defined as $d_{\text{alg}}(\mathbf{x}, C)^2 = \mathbf{x}^T C \mathbf{x}$. A linear solution for C can be obtained by minimizing $\sum_i d_{\text{alg}}(\mathbf{x}_i, C)^2$ with a suitable normalization on C . There is no linear expression for the perpendicular distance of a point (x, y) to a conic C , since through each point in \mathbb{R}^2 there are up to 4 lines perpendicular to C . The solution can be obtained from the roots of a quartic. However, a function $d_\perp(\mathbf{x}, C)$ may be defined which returns the shortest distance between a conic

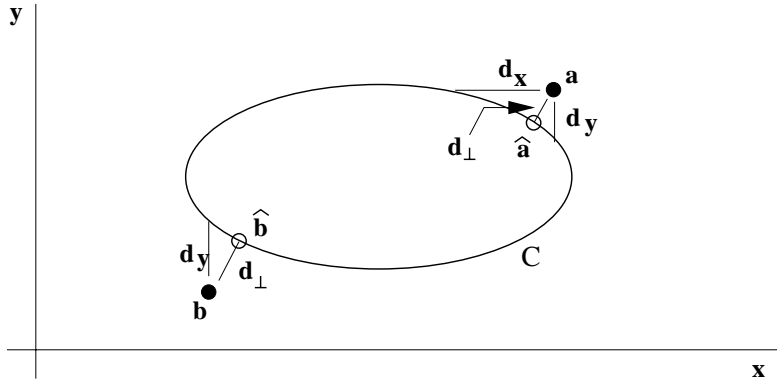


Fig. 4.3. A conic may be estimated from a set of 2D points by minimizing “symmetric transfer error” $d_x^2 + d_y^2$ or the sum of squared perpendicular distances d_{\perp}^2 . The analogue of transfer error is to consider x as perfect and measure the distance d_y to the conic in the y direction, and similarly for d_x . For point \mathbf{a} it is clear that $d_{\perp} \leq d_x$ and $d_{\perp} \leq d_y$. Also d_{\perp} is more stable than d_x or d_y as illustrated by point \mathbf{b} where d_x cannot be defined.

and a point. A conic can then be estimated by minimizing $\sum_i d_{\perp}(\mathbf{x}_i, C)^2$ over the five parameters of C , though this cannot be achieved by a linear solution. Given a conic C and a measured point \mathbf{x} , a corrected point $\hat{\mathbf{x}}$ is obtained simply by choosing the closest point on C .

We return now to estimating a homography. In the case of an affine transformation the variety is the intersection of two hyperplanes, i.e. it is a linear subspace of dimension 2. This follows from the form (4.9) of the affine matrix which for $\mathbf{x}' = H_A \mathbf{x}$ yields one linear constraint between x, x', y and another between x, y, y' , each of which defines a hyperplane in \mathbb{R}^4 . An analogue of this situation is line fitting to points on the plane. In both cases the relation (affine transformation or line) may be estimated by minimizing the perpendicular distance of points to the variety. In both cases there is a closed form solution as discussed in the following section.

4.2.6 Sampson error

The geometric error (4.8) is quite complex in nature, and minimizing it requires the simultaneous estimation of both the homography matrix and the points $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i$. This non-linear estimation problem will be discussed further in section 4.5. Its complexity contrasts with the simplicity of minimizing the algebraic error (4.4). The geometric interpretation of geometric error given in section 4.2.5 leads to a further cost function that lies between the algebraic and geometric cost functions in terms of complexity, but gives a close approximation to geometric error. We will refer to this cost function as *Sampson error* since Sampson [Sampson-82] used this approximation for conic fitting.

As described in section 4.2.5, the vector $\hat{\mathbf{X}}$ that minimizes the geometric error $\|\mathbf{X} - \hat{\mathbf{X}}\|^2$ is the closest point on the variety \mathcal{V}_H to the measurement \mathbf{X} . This point can not be estimated directly except via iteration, because of the non-linear nature of the variety \mathcal{V}_H . The idea of the Sampson error function is to estimate a first-order approximation to the point $\hat{\mathbf{X}}$, assuming that the cost function is well approximated linearly in the neighbourhood of the estimated point. The discussion to follow is related directly to

the 2D homography estimation problem, but applies substantially unchanged to the other estimation problems discussed in this book.

For a given homography H , any point $\mathbf{X} = (x, y, x', y')^T$ that lies on \mathcal{V}_H will satisfy the equation (4.3–p89), or $A\mathbf{h} = \mathbf{0}$. To emphasize the dependency on \mathbf{X} we will write this instead as $\mathcal{C}_H(\mathbf{X}) = \mathbf{0}$, where $\mathcal{C}_H(\mathbf{X})$ is in this case a 2-vector. To first order, this cost function may be approximated by a Taylor expansion

$$\mathcal{C}_H(\mathbf{X} + \delta_{\mathbf{X}}) = \mathcal{C}_H(\mathbf{X}) + \frac{\partial \mathcal{C}_H}{\partial \mathbf{X}} \delta_{\mathbf{X}}. \quad (4.10)$$

If we write $\delta_{\mathbf{X}} = \hat{\mathbf{X}} - \mathbf{X}$ and desire $\hat{\mathbf{X}}$ to lie on the variety \mathcal{V}_H so that $\mathcal{C}_H(\hat{\mathbf{X}}) = \mathbf{0}$, then the result is $\mathcal{C}_H(\mathbf{X}) + (\partial \mathcal{C}_H / \partial \mathbf{X}) \delta_{\mathbf{X}} = \mathbf{0}$, which we will henceforth write as $J\delta_{\mathbf{X}} = -\epsilon$ where J is the partial-derivative matrix, and ϵ is the cost $\mathcal{C}_H(\mathbf{X})$ associated with \mathbf{X} . The minimization problem that we now face is to find the smallest $\delta_{\mathbf{X}}$ that satisfies this equation, namely:

- Find the vector $\delta_{\mathbf{X}}$ that minimizes $\|\delta_{\mathbf{X}}\|$ subject to $J\delta_{\mathbf{X}} = -\epsilon$.

The standard way to solve problems of this type is to use Lagrange multipliers. A vector λ of Lagrange multipliers is introduced, and the problem reduces to that of finding the extrema of $\delta_{\mathbf{X}}^T \delta_{\mathbf{X}} - 2\lambda^T (J\delta_{\mathbf{X}} + \epsilon)$, where the factor 2 is simply introduced for convenience. Taking derivatives with respect to $\delta_{\mathbf{X}}$ and equating to zero gives

$$2\delta_{\mathbf{X}}^T - 2\lambda^T J = \mathbf{0}^T$$

from which we obtain $\delta_{\mathbf{X}} = J^T \lambda$. The derivative with respect to λ gives $J\delta_{\mathbf{X}} + \epsilon = \mathbf{0}$, the original constraint. Substituting for $\delta_{\mathbf{X}}$ leads to

$$JJ^T \lambda = -\epsilon$$

which may be solved for λ giving $\lambda = -(JJ^T)^{-1}\epsilon$, and so finally

$$\delta_{\mathbf{X}} = -J^T (JJ^T)^{-1} \epsilon, \quad (4.11)$$

and $\hat{\mathbf{X}} = \mathbf{X} + \delta_{\mathbf{X}}$. The norm $\|\delta_{\mathbf{X}}\|^2$ is the Sampson error:

$$\|\delta_{\mathbf{X}}\|^2 = \delta_{\mathbf{X}}^T \delta_{\mathbf{X}} = \epsilon^T (JJ^T)^{-1} \epsilon. \quad (4.12)$$

Example 4.2. Sampson approximation for a conic

We will compute the Sampson approximation to the geometric distance $d_{\perp}(\mathbf{x}, C)$ between a point \mathbf{x} and conic C shown in figure 4.3. In this case the conic variety \mathcal{V}_C is defined by the equation $\mathbf{x}^T C \mathbf{x} = 0$, so that $\mathbf{X} = (x, y)^T$ is a 2-vector, $\epsilon = \mathbf{x}^T C \mathbf{x}$ is a scalar, and J is the 1×2 matrix given by

$$J = \left[\frac{\partial(\mathbf{x}^T C \mathbf{x})}{\partial x}, \frac{\partial(\mathbf{x}^T C \mathbf{x})}{\partial y} \right].$$

This means that JJ^T is a scalar. The elements of J may be computed by the chain rule as

$$\frac{\partial(\mathbf{x}^T C \mathbf{x})}{\partial x} = \frac{\partial(\mathbf{x}^T C \mathbf{x})}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial x} = 2\mathbf{x}^T C (1, 0, 0)^T = 2(C\mathbf{x})_1$$

where $(\mathbf{Cx})_i$ denotes the i -th component of the 3-vector \mathbf{Cx} . Then from (4.12)

$$d_{\perp}^2 = \|\delta_{\mathbf{x}}\|^2 = \boldsymbol{\epsilon}^T (\mathbf{J}\mathbf{J}^T)^{-1} \boldsymbol{\epsilon} = \frac{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}{\mathbf{J}\mathbf{J}^T} = \frac{(\mathbf{x}^T \mathbf{Cx})^2}{4((\mathbf{Cx})_1^2 + (\mathbf{Cx})_2^2)}$$

△

A few points to note:

- (i) For the 2D homography estimation problem, $\mathbf{X} = (x, y, x', y')^T$ where the 2D measurements are $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$.
- (ii) $\boldsymbol{\epsilon} = \mathcal{C}_{\mathbf{H}}(\mathbf{X})$ is the algebraic error vector $\mathbf{A}_i \mathbf{h}$ – a 2-vector – and \mathbf{A}_i is defined in (4.3–p89).
- (iii) $\mathbf{J} = \partial \mathcal{C}_{\mathbf{H}}(\mathbf{X}) / \partial \mathbf{X}$ is a 2×4 matrix. For example

$$J_{11} = \partial(-w'_i \mathbf{x}_i^T \mathbf{h}^2 + y'_i \mathbf{x}_i^T \mathbf{h}^3) / \partial x = -w'_i h_{21} + y'_i h_{31}.$$

- (iv) Note the similarity of (4.12) to the algebraic error $\|\boldsymbol{\epsilon}\| = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$. The Sampson error may be interpreted as being the Mahalanobis norm (see section A2.1–(p565)), $\|\boldsymbol{\epsilon}\|_{\mathbf{J}\mathbf{J}^T}$.
- (v) One could alternatively use \mathbf{A} defined by (4.1–p89), in which case \mathbf{J} has dimension 3×4 and $\boldsymbol{\epsilon}$ is a 3-vector. However, in general the Sampson error, and consequently the solution $\delta_{\mathbf{x}}$, will be independent of whether (4.1–p89) or (4.3–p89) is used.

The Sampson error (4.12) is derived here for a single point pair. In applying this to the estimation of a 2D homography \mathbf{H} from several point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, the errors corresponding to all the point correspondences must be summed, giving

$$\mathcal{D}_{\perp} = \sum_i \boldsymbol{\epsilon}_i^T (\mathbf{J}_i \mathbf{J}_i^T)^{-1} \boldsymbol{\epsilon}_i \quad (4.13)$$

where $\boldsymbol{\epsilon}$ and \mathbf{J} both depend on \mathbf{H} . To estimate \mathbf{H} , this expression must be minimized over all values of \mathbf{H} . This is a simple minimization problem in which the set of variable parameters consists only of the entries (or some other parametrization) of \mathbf{H} .

This derivation of the Sampson error assumed that each point had isotropic (circular) error distribution, the same in each image. The appropriate formulae for more general Gaussian error distributions are given in the exercises at the end of this chapter.

Linear cost function

The algebraic error vector $\mathcal{C}_{\mathbf{H}}(\mathbf{X}) = \mathbf{A}(\mathbf{X})\mathbf{h}$ is typically multilinear in the entries of \mathbf{X} . The case where $\mathbf{A}(\mathbf{X})\mathbf{h}$ is *linear* is, however, important in its own right. The first point to note is that in this case, the first-order approximation to geometric error given by the Taylor expansion in (4.10) is exact (the higher order terms are zero), which means that *the Sampson error is identical to geometric error*.

In addition, the variety $\mathcal{V}_{\mathbf{H}}$ defined by the equation $\mathcal{C}_{\mathbf{H}}(\mathbf{X}) = \mathbf{0}$, a set of linear equations, is a hyperplane depending on \mathbf{H} . The problem of finding \mathbf{H} now becomes a hyperplane fitting problem – find the best fit to the data \mathbf{X}_i among the hyperplanes parametrized by \mathbf{H} .

As an example of this idea a *linear* algorithm which minimizes geometric error (4.8) for an affine transformation is developed in the exercises at the end of this chapter.

4.2.7 Another geometric interpretation

It was shown in section 4.2.5 that finding a homography that takes a set of points \mathbf{x}_i to another set \mathbf{x}'_i is equivalent to the problem of fitting a variety of a given type to a set of points in \mathbb{R}^4 . We now consider a different interpretation in which the set of all measurements is represented by a single point in a measurement space \mathbb{R}^N .

The estimation problems we consider may all be fitted into a common framework. In abstract terms the estimation problem has two components,

- a *measurement space* \mathbb{R}^N consisting of *measurement vectors* \mathbf{X} , and
- a *model*, which in abstract terms may be thought of simply as a subset S of points in \mathbb{R}^N . A measurement vector \mathbf{X} that lies inside this subset is said to *satisfy the model*. Typically the subspace that satisfies the model is a submanifold, or variety in \mathbb{R}^N .

Now, given a measurement vector \mathbf{X} in \mathbb{R}^N , the estimation problem is to find the vector $\hat{\mathbf{X}}$, closest to \mathbf{X} , that satisfies the model.

It will now be pointed out how the 2D homography estimation problem fits into this framework.

Error in both images. Let $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ be a set of measured matched points for $i = 1, \dots, n$. In all, there are $4n$ measurements, namely two coordinates in each of two images for n points. Thus, the set of matched points represents a point in \mathbb{R}^N , where $N = 4n$. The vector made up of the coordinates of all the matched points in both images will be denoted \mathbf{X} .

Of course, not all sets of point pairs $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ are related via a homography H . A set of point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ for which there exists a projective transformation H satisfying $\mathbf{x}'_i = H\mathbf{x}_i$ for all i constitutes the subset of \mathbb{R}^N satisfying the model. In general, this set of points will form a submanifold S in \mathbb{R}^N (in fact a variety) of some dimension. The dimension of this submanifold is equal to the minimal number of parameters that may be used to parametrize the submanifold.

One may arbitrarily choose n points $\hat{\mathbf{x}}_i$ in the first image. In addition, a homography H may be chosen arbitrarily. Once these choices have been made, the points $\hat{\mathbf{x}}'_i$ in the second image are determined by $\hat{\mathbf{x}}'_i = H\hat{\mathbf{x}}_i$. Thus, a feasible choice of points is determined by a set of $2n + 8$ parameters: the $2n$ coordinates of the points $\hat{\mathbf{x}}_i$, plus the 8 independent parameters (degrees of freedom) of the transformation H . Thus, the submanifold $S \subset \mathbb{R}^N$ has dimension $2n + 8$, and hence codimension $2n - 8$.

Given a set of measured point pairs $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, corresponding to a point \mathbf{X} in \mathbb{R}^N , and an estimated point $\hat{\mathbf{X}} \in \mathbb{R}^N$ lying on S , one easily verifies that

$$\|\mathbf{X} - \hat{\mathbf{X}}\|^2 = \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2.$$

Thus, finding the point $\hat{\mathbf{X}}$ on S lying closest to \mathbf{X} in \mathbb{R}^N is equivalent to minimizing the cost function given by (4.8). The estimated correct correspondences $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ are

those corresponding to the closest surface point $\hat{\mathbf{X}}$ in \mathbb{R}^N . Once $\hat{\mathbf{X}}$ is known \mathbf{H} may be computed.

Error in one image only. In the case of error in one image, one has a set of correspondences $\{\bar{\mathbf{x}}_i \leftrightarrow \mathbf{x}'_i\}$. The points $\bar{\mathbf{x}}_i$ are assumed perfect. The inhomogeneous coordinates of the \mathbf{x}'_i constitute the measurement vector \mathbf{X} . Hence, in this case the measurement space has dimension $N = 2n$. The vector $\hat{\mathbf{X}}$ consists of the inhomogeneous coordinates of the mapped perfect points $\{\mathbf{H}\bar{\mathbf{x}}_1, \mathbf{H}\bar{\mathbf{x}}_2, \dots, \mathbf{H}\bar{\mathbf{x}}_n\}$. The set of measurement vectors satisfying the model is the set $\hat{\mathbf{X}}$ as \mathbf{H} varies over the set of all homography matrices. Once again this subspace is a variety. Its dimension is 8, since this is the total number of degrees of freedom of the homography matrix \mathbf{H} . As with the previous case, the codimension is $2n - 8$. One verifies that

$$\|\mathbf{X} - \hat{\mathbf{X}}\|^2 = \sum_i d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2.$$

Thus, finding the closest point on S to the measurement vector \mathbf{X} is equivalent to minimizing the cost function (4.6).

4.3 Statistical cost functions and Maximum Likelihood estimation

In section 4.2, various cost functions were considered that were related to geometric distance between estimated and measured points in an image. The use of such cost functions is now justified and then generalized by a consideration of error statistics of the point measurements in an image.

In order to obtain a best (optimal) estimate of \mathbf{H} it is necessary to have a model for the measurement error (the “noise”). We are assuming here that in the absence of measurement error the true points exactly satisfy a homography, i.e. $\bar{\mathbf{x}}'_i = \mathbf{H}\bar{\mathbf{x}}_i$. A common assumption is that image coordinate measurement errors obey a Gaussian (or normal) probability distribution. This assumption is surely not justified in general, and takes no account of the presence of outliers (grossly erroneous measurements) in the measured data. Methods for detecting and removing outliers will be discussed later in section 4.7. Once outliers have been removed, the assumption of a Gaussian error model, if still not strictly justified, becomes more tenable. Therefore, for the present, we assume that image measurement errors obey a zero-mean isotropic Gaussian distribution. This distribution is described in section A2.1(p565).

Specifically we assume that the noise is Gaussian on each image coordinate with zero mean and uniform standard deviation σ . This means that $x = \bar{x} + \Delta x$, with Δx obeying a Gaussian distribution with variance σ^2 . If it is further assumed that the noise on each measurement is independent, then, if the true point is $\bar{\mathbf{x}}$, the probability density function (PDF) of each measured point \mathbf{x} is

$$\Pr(\mathbf{x}) = \left(\frac{1}{2\pi\sigma^2} \right) e^{-d(\mathbf{x}, \bar{\mathbf{x}})^2 / (2\sigma^2)}. \quad (4.14)$$

Error in one image. First we consider the case where the errors are only in the second image. The probability of obtaining the set of correspondences $\{\bar{\mathbf{x}}_i \leftrightarrow \mathbf{x}'_i\}$ is

simply the product of their individual PDFs, since the errors on each point are assumed independent. Then the PDF of the noise-perturbed data is

$$\Pr(\{\mathbf{x}'_i\}|\mathbf{H}) = \prod_i \left(\frac{1}{2\pi\sigma^2} \right) e^{-d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2/(2\sigma^2)}. \quad (4.15)$$

The symbol $\Pr(\{\mathbf{x}'_i\}|\mathbf{H})$ is to be interpreted as meaning the probability of obtaining the measurements $\{\mathbf{x}'_i\}$ given that the true homography is \mathbf{H} . The *log-likelihood* of the set of correspondences is

$$\log \Pr(\{\mathbf{x}'_i\}|\mathbf{H}) = -\frac{1}{2\sigma^2} \sum_i d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2 + \text{constant}.$$

The *Maximum Likelihood estimate* (MLE) of the homography, $\hat{\mathbf{H}}$, maximizes this log-likelihood, i.e. minimizes

$$\sum_i d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2.$$

Thus, we note that ML estimation is equivalent to minimizing the geometric error function (4.6).

Error in both images. Following a similar development to the above, if the true correspondences are $\{\bar{\mathbf{x}}_i \leftrightarrow \mathbf{H}\bar{\mathbf{x}}_i = \bar{\mathbf{x}}'_i\}$, then the PDF of the noise-perturbed data is

$$\Pr(\{\mathbf{x}_i, \mathbf{x}'_i\}|\mathbf{H}, \{\bar{\mathbf{x}}_i\}) = \prod_i \left(\frac{1}{2\pi\sigma^2} \right) e^{-(d(\mathbf{x}_i, \bar{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2)/(2\sigma^2)}.$$

The additional complication here is that we have to seek “corrected” image measurements that play the role of the true measurements ($\mathbf{H}\bar{\mathbf{x}}$ above). Thus the ML estimate of the projective transformation \mathbf{H} and the correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, is the homography $\hat{\mathbf{H}}$ and corrected correspondences $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$ that minimize

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

with $\hat{\mathbf{x}}'_i = \hat{\mathbf{H}}\hat{\mathbf{x}}_i$. Note that in this case, the ML estimate is identical with minimizing the reprojection error function (4.8).

Mahalanobis distance. In the general Gaussian case, one may assume a vector of measurements \mathbf{X} satisfying a Gaussian distribution function with covariance matrix Σ . The cases above are equivalent to a covariance matrix which is a multiple of the identity.

Maximizing the log-likelihood is then equivalent to minimizing the Mahalanobis distance (see section A2.1(p565))

$$\|\mathbf{X} - \bar{\mathbf{X}}\|_{\Sigma}^2 = (\mathbf{X} - \bar{\mathbf{X}})^{\top} \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}).$$

In the case where there is error in each image, but assuming that errors in one image are independent of the error in the other image, the appropriate cost function is

$$\|\mathbf{X} - \bar{\mathbf{X}}\|_{\Sigma}^2 + \|\mathbf{X}' - \bar{\mathbf{X}}'\|_{\Sigma'}^2$$

where Σ and Σ' are the covariance matrices of the measurements in the two images.

Finally, if we assume that the errors for all the points \mathbf{x}_i and \mathbf{x}'_i are independent, with individual covariance matrices Σ_i and Σ'_i respectively, then the above expression expands to

$$\sum \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|_{\Sigma_i}^2 + \sum \|\mathbf{x}'_i - \bar{\mathbf{x}}'_i\|_{\Sigma'_i}^2 \quad (4.16)$$

This equation allows the incorporation of the type of anisotropic covariance matrices that arise for point locations computed as the intersection of two non-perpendicular lines. In the case where the points are known exactly in one of the two images, errors being confined to the other image, one of the two summation terms in (4.16) disappears.

4.4 Transformation invariance and normalization

We now start to discuss the properties and performance of the DLT algorithm of section 4.1 and how it compares with algorithms minimizing geometric error. The first topic is the invariance of the algorithm to different choices of coordinates in the image. It is clear that it would generally be undesirable for the result of an algorithm to be dependent on such arbitrary choices as the origin and scale, or even orientation, of the coordinate system in an image.

4.4.1 Invariance to image coordinate transformations

Image coordinates are sometimes given with the origin at the top-left of the image, and sometimes with the origin at the centre. The question immediately occurs whether this makes a difference to the results of computing the transformation. Similarly, if the units used to express image coordinates are changed by multiplication by some factor, then is it possible that the result of the algorithm changes also? More generally, to what extent is the result of an algorithm that minimizes a cost function to estimate a homography dependent on the choice of coordinates in the image? Suppose, for instance, that the image coordinates are changed by some similarity, affine or even projective transformation before running the algorithm. Will this materially change the result?

Formally, suppose that coordinates \mathbf{x} in one image are replaced by $\tilde{\mathbf{x}} = \mathbf{T}\mathbf{x}$, and coordinates \mathbf{x}' in the other image are replaced by $\tilde{\mathbf{x}}' = \mathbf{T}'\mathbf{x}'$, where \mathbf{T} and \mathbf{T}' are 3×3 homographies. Substituting in the equation $\mathbf{x}' = \mathbf{H}\mathbf{x}$, we derive the equation $\tilde{\mathbf{x}}' = \mathbf{T}'\mathbf{H}\mathbf{T}^{-1}\tilde{\mathbf{x}}$. This relation implies that $\tilde{\mathbf{H}} = \mathbf{T}'\mathbf{H}\mathbf{T}^{-1}$ is the transformation matrix for the point correspondences $\tilde{\mathbf{x}} \leftrightarrow \tilde{\mathbf{x}}'$. An alternative method of finding the transformation taking \mathbf{x}_i to \mathbf{x}'_i is therefore suggested, as follows.

- (i) Transform the image coordinates according to transformations $\tilde{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$ and $\tilde{\mathbf{x}}'_i = \mathbf{T}'\mathbf{x}'_i$.
- (ii) Find the transformation $\tilde{\mathbf{H}}$ from the correspondences $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i$.
- (iii) Set $\mathbf{H} = \mathbf{T}'^{-1}\tilde{\mathbf{H}}\mathbf{T}$.

The transformation matrix \mathbf{H} found in this way applies to the original untransformed point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. What choice should be made for the transformations \mathbf{T} and \mathbf{T}' will be left unspecified for now. The question to be decided now is whether the

outcome of this algorithm is independent of the transformations T and T' being applied. Ideally it ought to be, at least when T and T' are similarity transformations, since the choice of a different scale, orientation or coordinate origin in the images should not materially affect the outcome of the algorithm.

In the subsequent sections it will be shown that an algorithm that minimizes geometric error is invariant to similarity transformations. On the other hand, for the DLT algorithm as described in section 4.1, the result unfortunately is not invariant to similarity transformations. The solution is to apply a normalizing transformation to the data before applying the DLT algorithm. This normalizing transformation will nullify the effect of the arbitrary selection of origin and scale in the coordinate frame of the image, and will mean that the combined algorithm is invariant to a similarity transformation of the image. Appropriate normalizing transformations will be discussed later.

4.4.2 Non-invariance of the DLT algorithm

Consider a set of correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and a matrix H that is the result of the DLT algorithm applied to this set of corresponding points. Consider further a related set of correspondences $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i$ where $\tilde{\mathbf{x}}_i = T\mathbf{x}_i$ and $\tilde{\mathbf{x}}'_i = T'\mathbf{x}'_i$, and let \tilde{H} be defined by $\tilde{H} = T'HT^{-1}$. Following section 4.4.1, the question to be decided here is the following:

- Does the DLT algorithm applied to the correspondence set $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i$ yield the transformation \tilde{H} ?

We will use the following notation: Matrix A_i is the DLT equation matrix (4.3–p89) derived from a point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, and A is the $2n \times 9$ matrix formed by stacking the A_i . Matrix \tilde{A}_i is similarly defined in terms of the correspondences $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i$, where $\tilde{\mathbf{x}}_i = T\mathbf{x}_i$ and $\tilde{\mathbf{x}}'_i = T'\mathbf{x}'_i$ for some projective transformations T and T' .

Result 4.3. Let T' be a similarity transformation with scale factor s , and let T be an arbitrary projective transformation. Further, suppose H is any 2D homography and let \tilde{H} be defined by $\tilde{H} = T'HT^{-1}$. Then $\|\tilde{A}\tilde{h}\| = s\|Ah\|$ where h and \tilde{h} are the vectors of entries of H and \tilde{H} .

Proof. Define the vector $\epsilon_i = \mathbf{x}'_i \times H\mathbf{x}_i$. Note that $A_i h$ is the vector consisting of the first two entries of ϵ_i . Let $\tilde{\epsilon}_i$ be similarly defined in terms of the transformed quantities as $\tilde{\epsilon}_i = \tilde{\mathbf{x}}'_i \times \tilde{H}\tilde{\mathbf{x}}_i$. One computes:

$$\begin{aligned}\tilde{\epsilon}_i &= \tilde{\mathbf{x}}'_i \times \tilde{H}\tilde{\mathbf{x}}_i = T'\mathbf{x}'_i \times (T'HT^{-1})T\mathbf{x}_i \\ &= T'\mathbf{x}'_i \times T'H\mathbf{x}_i = T'^*(\mathbf{x}'_i \times H\mathbf{x}_i) \\ &= T'^*\epsilon_i\end{aligned}$$

where T'^* represents the cofactor matrix of T' and the second-last equality follows from lemma A4.2(p581). For a general transformation T , the error vectors $A_i h$ and $\tilde{A}_i \tilde{h}$ (namely the first two components of ϵ_i and $\tilde{\epsilon}_i$) are not simply related. However, in the special case where T' is a similarity transformation, one may write $T' = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$ where R is a rotation matrix, t is a translation and s is a scaling factor. In this case, we

see that $T'^* = s \begin{bmatrix} R & 0 \\ -t^T R & s \end{bmatrix}$. Applying T'^* just to the first two components of ϵ_i , one sees that

$$\tilde{A}_i \tilde{h} = (\tilde{\epsilon}_{i1}, \tilde{\epsilon}_{i2})^T = sR(\epsilon_{i1}, \epsilon_{i2})^T = sRA_i h.$$

Since rotation does not affect vector norms, one sees that $\|\tilde{A}h\| = s\|Ah\|$, as required. This result may be expressed in terms of algebraic error as

$$d_{\text{alg}}(\tilde{x}'_i, \tilde{H}\tilde{x}_i) = sd_{\text{alg}}(x'_i, Hx_i).$$

□

Thus, there is a one-to-one correspondence between H and \tilde{H} giving rise to the same error, except for constant scale. It may appear therefore that the matrices H and \tilde{H} minimizing the algebraic error will be related by the formula $\tilde{H} = T'HT^{-1}$, and hence one may retrieve H as the product $T'^{-1}\tilde{H}T$. This conclusion is **false** however. For, although H and \tilde{H} so defined give rise to the same error ϵ , the condition $\|H\| = 1$, imposed as a constraint on the solution, is not equivalent to the condition $\|\tilde{H}\| = 1$. Specifically, $\|H\|$ and $\|\tilde{H}\|$ are not related in any simple manner. Thus, there is no one-to-one correspondence between H and \tilde{H} giving rise to the same error ϵ , subject to the constraint $\|H\| = \|\tilde{H}\| = 1$. Specifically,

$$\begin{aligned} & \text{minimize } \sum_i d_{\text{alg}}(x'_i, Hx_i)^2 \text{ subject to } \|H\| = 1 \\ \Leftrightarrow & \text{minimize } \sum_i d_{\text{alg}}(\tilde{x}'_i, \tilde{H}\tilde{x}_i)^2 \text{ subject to } \|H\| = 1 \\ \nRightarrow & \text{minimize } \sum_i d_{\text{alg}}(\tilde{x}'_i, \tilde{H}\tilde{x}_i)^2 \text{ subject to } \|\tilde{H}\| = 1. \end{aligned}$$

Thus, the method of transformation leads to a different solution for the computed transformation matrix. This is a rather undesirable feature of the DLT algorithm as it stands, that the result is changed by a change of coordinates, or even simply a change of the origin of coordinates. If the constraint under which the norm $\|Ah\|$ is minimized is invariant under the transformation, however, then one sees that the computed matrices H and \tilde{H} are related in the right way. Examples of minimization conditions for which H is transformation-invariant are discussed in the exercises at the end of this chapter.

4.4.3 Invariance of geometric error

It will be shown now that minimizing geometric error to find H is invariant under similarity (scaled Euclidean) transformations. As before, consider a point correspondence $x \leftrightarrow x'$ and a transformation matrix H . Also, define a related set of correspondences $\tilde{x} \leftrightarrow \tilde{x}'$ where $\tilde{x} = Tx$ and $\tilde{x}' = T'x'$, and let \tilde{H} be defined by $\tilde{H} = T'HT^{-1}$. Suppose that T and T' represent Euclidean transformations of \mathbb{P}^2 . One verifies that

$$d(\tilde{x}', \tilde{H}\tilde{x}) = d(T'x', T'HT^{-1}Tx) = d(T'x', T'Hx) = d(x', Hx)$$

where the last equality holds because Euclidean distance is unchanged under a Euclidean transformation such as T' . This shows that if H minimizes the geometric error

for a set of correspondences, then \tilde{H} minimizes the geometric error for the transformed set of correspondences, and so minimizing geometric error is invariant under Euclidean transformations.

For similarity transformations, geometric error is multiplied by the scale factor of the transformation, hence the minimizing transformations correspond in the same way as in the Euclidean transformation case. Minimizing geometric error is invariant to similarity transformations.

4.4.4 Normalizing transformations

As was shown in section 4.4.2, the result of the DLT algorithm for computing 2D homographies depends on the coordinate frame in which points are expressed. In fact the result is not invariant to similarity transformations of the image. This suggests the question whether some coordinate systems are in some way better than others for computing a 2D homography. The answer to this is an emphatic *yes*. In this section a method of normalization of the data is described, consisting of translation and scaling of image coordinates. This normalization should be carried out before applying the DLT algorithm. Subsequently an appropriate correction to the result expresses the computed H with respect to the original coordinate system.

Apart from improved accuracy of results, data normalization provides a second desirable benefit, namely that an algorithm that incorporates an initial data normalization step will be invariant with respect to arbitrary choices of the scale and coordinate origin. This is because the normalization step undoes the effect of coordinate changes, by effectively choosing a canonical coordinate frame for the measurement data. Thus, algebraic minimization is carried out in a fixed canonical frame, and the DLT algorithm is in practice invariant to similarity transformations.

Isotropic scaling. As a first step of normalization, the coordinates in each image are translated (by a different translation for each image) so as to bring the centroid of the set of all points to the origin. The coordinates are also scaled so that on the average a point \mathbf{x} is of the form $\mathbf{x} = (x, y, w)^T$, with each of x , y and w having the same average magnitude. Rather than choose different scale factors for each coordinate direction, an isotropic scaling factor is chosen so that the x and y -coordinates of a point are scaled equally. To this end, we choose to scale the coordinates so that the average distance of a point \mathbf{x} from the origin is equal to $\sqrt{2}$. This means that the “average” point is equal to $(1, 1, 1)^T$. In summary the transformation is as follows:

- (i) The points are translated so that their centroid is at the origin.
- (ii) The points are then scaled so that the average distance from the origin is equal to $\sqrt{2}$.
- (iii) This transformation is applied to each of the two images independently.

Why is normalization essential? The recommended version of the DLT algorithm with data normalization is given in algorithm 4.2. We will now motivate why this

version of the algorithm, incorporating data normalization, should be used in preference to the basic DLT of algorithm 4.1(p91). Note that normalization is also called *pre-conditioning* in the numerical literature.

The DLT method of algorithm 4.1 uses the SVD of $A = UDV^T$ to obtain a solution to the overdetermined set of equations $Ah = 0$. These equations do not have an exact solution (since the $2n \times 9$ matrix A will not have rank 8 for noisy data), but the vector h , given by the last column of V , provides a solution which minimizes $\|Ah\|$ (subject to $\|h\| = 1$). This is equivalent to finding the rank 8 matrix \hat{A} which is closest to A in Frobenius norm and obtaining h as the exact solution of $\hat{A}h = 0$. The matrix \hat{A} is given by $\hat{A} = U\hat{D}V^T$ where \hat{D} is D with the smallest singular value set to zero. The matrix \hat{A} has rank 8 and minimizes the difference to A in Frobenius norm because

$$\|A - \hat{A}\|_F = \|UDV^T - U\hat{D}V^T\|_F = \|D - \hat{D}\|_F.$$

where $\|\cdot\|_F$ is the Frobenius norm, i.e. the square root of the sum of squares of all entries.

Without normalization typical image points x_i, x'_i are of the order $(x, y, w)^T = (100, 100, 1)^T$, i.e., x, y are much larger than w . In A the entries xx', xy', yx', yy' will be of order 10^4 , entries xw', yw' etc. of order 10^2 , and entries ww' will be unity. Replacing A by \hat{A} means that some entries are increased and others decreased such that the square sum of differences of these changes is minimal (and the resulting matrix has rank 8). However, and this is the key point, increasing the term ww' by 100 means a huge change in the image points, whereas increasing the term xx' by 100 means only a slight change. This is the reason why all entries in A must have similar magnitude and why normalization is essential.

The effect of normalization is related to the condition number of the set of DLT equations, or more precisely the ratio d_1/d_{n-1} of the first to the second-last singular value of the equation matrix A . This point is investigated in more detail in [Hartley-97c]. For the present it is sufficient to say that for exact data and infinite precision arithmetic the results will be independent of the normalizing transformation. However, in the presence of noise the solution will diverge from the correct result. The effect of a large condition number is to amplify this divergence. This is true even for infinite-precision arithmetic – this is not a round-off error effect.

The effect that this data normalization has on the results of the DLT algorithm is shown graphically in figure 4.4. The conclusion to be drawn here is that data normalization gives dramatically better results. The examples shown in the figure are chosen to make the effect easily visible. However, a marked advantage remains even in cases of computation from larger numbers of point correspondences, with points more widely distributed. To emphasize this point we remark:

- *Data normalization is an essential step in the DLT algorithm. It must not be considered optional.*

Data normalization becomes even more important for less well conditioned problems, such as the DLT computation of the fundamental matrix or the trifocal tensor, which will be considered in later chapters.

Objective

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

Algorithm

- (i) **Normalization of \mathbf{x} :** Compute a similarity transformation \mathbf{T} , consisting of a translation and scaling, that takes points \mathbf{x}_i to a new set of points $\tilde{\mathbf{x}}_i$ such that the centroid of the points $\tilde{\mathbf{x}}_i$ is the coordinate origin $(0, 0)^T$, and their average distance from the origin is $\sqrt{2}$.
- (ii) **Normalization of \mathbf{x}' :** Compute a similar transformation \mathbf{T}' for the points in the second image, transforming points \mathbf{x}'_i to $\tilde{\mathbf{x}}'_i$.
- (iii) **DLT:** Apply algorithm 4.1(p91) to the correspondences $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i$ to obtain a homography $\tilde{\mathbf{H}}$.
- (iv) **Denormalization:** Set $\mathbf{H} = \mathbf{T}'^{-1}\tilde{\mathbf{H}}\mathbf{T}$.

Algorithm 4.2. *The normalized DLT for 2D homographies.*

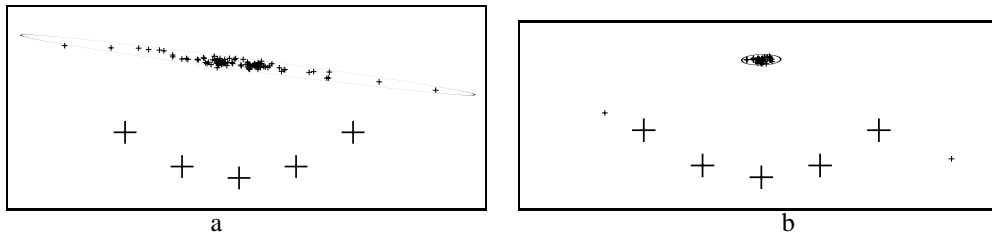


Fig. 4.4. *Results of Monte Carlo simulation (see section 5.3(p149) of computation of 2D homographies). A set of 5 points (denoted by large crosses) was used to compute a 2D homography. Each of the 5 points is mapped (in the noise-free case) to the point with the same coordinates, so that homography \mathbf{H} is the identity mapping. Now, 100 trials were made with each point being subject to 0.1 pixel Gaussian noise in one image. (For reference, the large crosses are 4 pixels across.) The mapping \mathbf{H} computed using the DLT algorithm was then applied to transfer a further point into the second image. The 100 projections of this point are shown with small crosses and the 95% ellipse computed from their scatter matrix is also shown. (a) are the results without data normalization, and (b) the results with normalization. The left- and rightmost reference points have (unnormalized) coordinates (130, 108) and (170, 108).*

Non-isotropic scaling. Other methods of scaling are also possible. In non-isotropic scaling, the centroid of the points is translated to the origin as before. After this translation the points form a cloud about the origin. Scaling is then carried out so that the two principal moments of the set of points are both equal to unity. Thus, the set of points will form an approximately symmetric circular cloud of points of radius 1 about the origin. Experimental results given in [Hartley-97c] suggest that the extra effort required for non-isotropic scaling does not lead to significantly better results than isotropic scaling.

A further variant on scaling was discussed in [Muehlich-98], based on a statistical analysis of the estimator, its bias and variance. In that paper it was observed that some columns of \mathbf{A} are not affected by noise. This applies to the third and sixth columns in (4.3–p89), corresponding to the entry $w_i w'_i = 1$. Such error-free entries in \mathbf{A} should not be varied in finding $\hat{\mathbf{A}}$, the closest rank-deficient approximation to \mathbf{A} . A method known

as Total Least Squares - Fixed Columns is used to find the best solution. For estimation of the fundamental matrix (see chapter 11), [Muehlich-98] reports slightly improved results compared with non-isotropic scaling.

Scaling with points near infinity. Consider the case of estimation of a homography between an infinite plane and an image. If the viewing direction is sufficiently oblique, then very distant points in the plane may be visible in the image – even points at infinity (vanishing points) if the horizon is visible. In this case it makes no sense to normalize the coordinates of points in the infinite plane by setting the centroid at the origin, since the centroid may have very large coordinates, or be undefined. An approach to normalization in this case is considered in exercise (iii) on page 128.

4.5 Iterative minimization methods

This section describes methods for minimizing the various geometric cost functions developed in section 4.2 and section 4.3. Minimizing such cost functions requires the use of iterative techniques. This is unfortunate, because iterative techniques tend to have certain disadvantages compared to linear algorithms such as the normalized DLT algorithm 4.2:

- (i) They are slower.
- (ii) They generally need an initial estimate at which to start the iteration.
- (iii) They risk not converging, or converging to a local minimum instead of the global minimum.
- (iv) Selection of a stopping criterion for iteration may be tricky.

Consequently, iterative techniques generally require more careful implementation.

The technique of iterative minimization generally consists of five steps:

- (i) **Cost function.** A cost function is chosen as the basis for minimization. Different possible cost functions were discussed in section 4.2.
- (ii) **Parametrization.** The transformation (or other entity) to be computed is expressed in terms of a finite number of parameters. It is not in general necessary that this be a minimum set of parameters, and there are in fact often advantages to over-parametrization. (See the discussion below.)
- (iii) **Function specification.** A function must be specified that expresses the cost in terms of the set of parameters.
- (iv) **Initialization.** A suitable initial parameter estimate is computed. This will generally be done using a linear algorithm such as the DLT algorithm.
- (v) **Iteration.** Starting from the initial solution, the parameters are iteratively refined with the goal of minimizing the cost function.

A word about parametrization

For a given cost function, there are often several choices of parametrization. The general strategy that guides parametrization is to select a set of parameters that cover the complete space over which one is minimizing, while at the same time allowing one to

compute the cost function in a convenient manner. For example, H may be parametrized by 9 parameters – that is, it is over-parametrized, since there are really only 8 degrees of freedom, overall scale not being significant. A *minimal* parametrization (i.e. the same number of parameters as degrees of freedom) would involve only 8 parameters.

In general no bad effects are likely to occur if a minimization problem of this type is over-parametrized, as long as for all choices of parameters the corresponding object is of the desired type. In particular for homogeneous objects, such as the 3×3 projection matrix encountered here, it is usually not necessary or advisable to attempt to use a minimal parametrization by removing the scale-factor ambiguity.

The reasoning is the following: it is not *necessary* to use minimal parametrization because a well-performing non-linear minimization algorithm will “notice” that it is not necessary to move in redundant directions, such as the matrix scaling direction. The algorithm described in Gill and Murray [Gill-78], which is a modification of the Gauss–Newton method, has an effective strategy for discarding redundant combinations of the parameters. Similarly, the Levenberg-Marquardt algorithm (see section A6.2(p600)) handles redundant parametrizations easily. It is not *advisable* because it is found empirically that the cost function surface is more complicated when minimal parametrizations are used. There is then a greater possibility of becoming stuck in a local minimum.

One other issue that arises in choosing a parametrization is that of restricting the transformation to a particular class. For example, suppose H is known to be a homology, then as described in section A7.2(p629) it may be parametrized as

$$H = I + (\mu - 1) \frac{\mathbf{v}\mathbf{a}^T}{\mathbf{v}^T\mathbf{a}}$$

where μ is a scalar, and \mathbf{v} and \mathbf{a} 3-vectors. A homology has 5 degrees of freedom which correspond here to the scalar μ and the directions of \mathbf{v} and \mathbf{a} . If H is parametrized by its 9 matrix entries, then the estimated H is unlikely to exactly be a homology. However, if H is parametrized by μ , \mathbf{v} and \mathbf{a} (a total of 7 parameters) then the estimated H is guaranteed to be a homology. This parametrization is *consistent* with a homology (it is also an over-parametrization). We will return to the issues of consistent, local, minimal and over-parametrization in later chapters. The issues are also discussed further in appendix A6.9(p623).

Function specification

It has been seen in section 4.2.7 that a general class of estimation problems is concerned with a measurement space \mathbb{R}^N containing a model surface S . Given a measurement $\mathbf{X} \in \mathbb{R}^N$ the estimation task is to find the point $\hat{\mathbf{X}}$ lying on S closest to \mathbf{X} . In the case where a non-isotropic Gaussian error distribution is imposed on \mathbb{R}^N , the word *closest* is to be interpreted in terms of Mahalanobis distance. Iterative minimization methods will now be described in terms of this estimation model. In iterative estimation through parameter fitting, the model surface S is locally parametrized, and the parameters are allowed to vary to minimize the distance to the measured point. More specifically,

- (i) One has a *measurement vector* $\mathbf{X} \in \mathbb{R}^N$ with covariance matrix Σ .

- (ii) A set of parameters are represented as a vector $\mathbf{P} \in \mathbb{R}^M$.
- (iii) A mapping $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ is defined. The range of this mapping is (at least locally) the model surface S in \mathbb{R}^N representing the set of allowable measurements.
- (iv) The cost function to be minimized is the squared Mahalanobis distance

$$\|\mathbf{X} - f(\mathbf{P})\|_{\Sigma}^2 = (\mathbf{X} - f(\mathbf{P}))^T \Sigma^{-1} (\mathbf{X} - f(\mathbf{P})).$$

In effect, we are attempting to find a set of parameters \mathbf{P} such that $f(\mathbf{P}) = \mathbf{X}$, or failing that, to bring $f(\mathbf{P})$ as close to \mathbf{X} as possible, with respect to Mahalanobis distance. The Levenberg–Marquardt algorithm is a general tool for iterative minimization, when the cost function to be minimized is of this type. We will now show how the various different types of cost functions described in this chapter fit into this format.

Error in one image. Here one fixes the coordinates of points \mathbf{x}_i in the first image, and varies \mathbf{H} so as to minimize cost function (4.6–p94), namely

$$\sum_i d(\mathbf{x}'_i, \mathbf{H}\bar{\mathbf{x}}_i)^2.$$

The measurement vector \mathbf{X} is made up of the $2n$ inhomogeneous coordinates of the points \mathbf{x}'_i . One may choose as parameters the vector \mathbf{h} of entries of the homography matrix \mathbf{H} . The function f is defined by

$$f : \mathbf{h} \mapsto (\mathbf{H}\mathbf{x}_1, \mathbf{H}\mathbf{x}_2, \dots, \mathbf{H}\mathbf{x}_n)$$

where it is understood that here, and in the functions below, $\mathbf{H}\mathbf{x}_i$ indicates the inhomogeneous coordinates. One verifies that $\|\mathbf{X} - f(\mathbf{h})\|^2$ is equal to (4.6–p94).

Symmetric transfer error. In the case of the symmetric cost function (4.7–p95)

$$\sum_i d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}'_i)^2 + d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2$$

one chooses as measurement vector \mathbf{X} the $4n$ -vector made up of the inhomogeneous coordinates of the points \mathbf{x}_i followed by the inhomogeneous coordinates of the points \mathbf{x}'_i . The parameter vector as before is the vector \mathbf{h} of entries of \mathbf{H} , and the function f is defined by

$$f : \mathbf{h} \mapsto (\mathbf{H}^{-1}\mathbf{x}'_1, \dots, \mathbf{H}^{-1}\mathbf{x}'_n, \mathbf{H}\mathbf{x}_1, \dots, \mathbf{H}\mathbf{x}_n).$$

As before, we find that $\|\mathbf{X} - f(\mathbf{h})\|^2$ is equal to (4.7–p95).

Reprojection error. Minimizing the cost function (4.8–p95) is more complex. The difficulty is that it requires a simultaneous minimization over all choices of points $\hat{\mathbf{x}}_i$ as well as the entries of the transformation matrix \mathbf{H} . If there are many point correspondences, then this becomes a very large minimization problem. Thus, the problem may be parametrized by the coordinates of the points $\hat{\mathbf{x}}_i$ and the entries of the matrix $\hat{\mathbf{H}}$ – a total of $2n + 9$ parameters. The coordinates of $\hat{\mathbf{x}}'_i$ are not required, since they are related to the other parameters by $\hat{\mathbf{x}}'_i = \hat{\mathbf{H}}\hat{\mathbf{x}}_i$. The parameter vector is therefore

$\mathbf{P} = (\mathbf{h}, \hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n)$. The measurement vector contains the inhomogeneous coordinates of all the points \mathbf{x}_i and \mathbf{x}'_i . The function f is defined by

$$f : (\mathbf{h}, \hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n) \mapsto (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}'_1, \dots, \hat{\mathbf{x}}_n, \hat{\mathbf{x}}'_n)$$

where $\hat{\mathbf{x}}'_i = \mathbf{H}\hat{\mathbf{x}}_i$. One verifies that $\|\mathbf{X} - f(\mathbf{P})\|^2$, with \mathbf{X} a $4n$ -vector, is equal to the cost function (4.8–p95). This cost function must be minimized over all $2n + 9$ parameters.

Sampson approximation. In contrast with $2n + 9$ parameters of reprojection error, minimizing the error in one image (4.6–p94) or symmetric transfer error (4.7–p95) requires a minimization over the 9 entries of the matrix \mathbf{H} only – in general a more tractable problem. The Sampson approximation to reprojection error enables reprojection error also to be minimized with only 9 parameters.

This is an important consideration, since the iterative solution of an m -parameter non-linear minimization problem using a method such as Levenberg–Marquardt involves the solution of an $m \times m$ set of linear equations at each iteration step. This is a problem with complexity $O(m^3)$. Hence, it is appropriate to keep the size of m low.

The Sampson error avoids minimizing over the $2n + 9$ parameters of reprojection error because effectively it determines the $2n$ variables $\{\hat{\mathbf{x}}_i\}$ for each particular choice of \mathbf{h} . Consequently the minimization then only requires the 9 parameters of \mathbf{h} . In practice this approximation gives excellent results provided the errors are small compared to the measurements.

Initialization

An initial estimate for the parametrization may be found by employing a linear technique. For example, the normalized DLT algorithm 4.2 directly provides \mathbf{H} and thence the 9-vector \mathbf{h} used to parametrize the iterative minimization. In general if there are $n \geq 4$ correspondences, then all will be used in the linear solution. However, as will be seen in section 4.7 on robust estimation, when the correspondences contain outliers it may be advisable to use a carefully selected minimal set of correspondences (i.e. four correspondences). Linear techniques or minimal solutions are the two initialization techniques recommended in this book.

An alternative method that is sometimes used (for instance see [Horn-90, Horn-91]) is to carry out a sufficiently dense sampling of parameter space, iterating from each sampled starting point and retaining the best result. This is only possible if the dimension of the parameter space is sufficiently small. Sampling of parameter space may be done either randomly, or else according to some pattern. Another initialization method is simply to do without any effective initialization at all, starting the iteration at a given fixed point in parameter space. This method is not often viable. Iteration is very likely to fall into a false minimum or not converge. Even in the best case, the number of iteration steps required will increase the further one starts from the final solution. For this reason using a good initialization method is the best plan.

Objective

Given $n > 4$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the Maximum Likelihood estimate $\hat{\mathbf{H}}$ of the homography mapping between the images.

The MLE involves also solving for a set of subsidiary points $\{\hat{\mathbf{x}}_i\}$, which minimize

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

where $\hat{\mathbf{x}}'_i = \hat{\mathbf{H}}\hat{\mathbf{x}}_i$.

Algorithm

- (i) **Initialization:** Compute an initial estimate of $\hat{\mathbf{H}}$ to provide a starting point for the geometric minimization. For example, use the linear normalized DLT algorithm 4.2, or use RANSAC (section 4.7.1) to compute $\hat{\mathbf{H}}$ from four point correspondences.
- (ii) **Geometric minimization of – either Sampson error:**

- Minimize the Sampson approximation to the geometric error (4.12–p99).
- The cost is minimized using the Newton algorithm of section A6.1(p597) or Levenberg–Marquardt algorithm of section A6.2(p600) over a suitable parametrization of $\hat{\mathbf{H}}$. For example the matrix may be parametrized by its 9 entries.

or Gold Standard error:

- Compute an initial estimate of the subsidiary variables $\{\hat{\mathbf{x}}_i\}$ using the measured points $\{\mathbf{x}_i\}$ or (better) the Sampson correction to these points given by (4.11–p99).
- Minimize the cost

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

over $\hat{\mathbf{H}}$ and $\hat{\mathbf{x}}_i, i = 1, \dots, n$. The cost is minimized using the Levenberg–Marquardt algorithm over $2n+9$ variables: $2n$ for the n 2D points $\hat{\mathbf{x}}_i$, and 9 for the homography matrix $\hat{\mathbf{H}}$.

- If the number of points is large then the sparse method of minimizing this cost function given in section A6.4(p607) is the recommended approach.

Algorithm 4.3. The Gold Standard algorithm and variations for estimating \mathbf{H} from image correspondences. The Gold Standard algorithm is preferred to the Sampson method for 2D homography computation.

Iteration methods

There are various iterative methods for minimizing the chosen cost function, of which the most popular are Newton iteration and the Levenberg–Marquardt method. These methods are described in appendix 6(p597). Other general methods for minimizing a cost function are available, such as Powell’s method and the simplex method both described in [Press-88].

Summary. The ideas in this section are collected together in algorithm 4.3, which describes the Gold Standard and Sampson methods for estimating the homography mapping between point correspondences in two images.

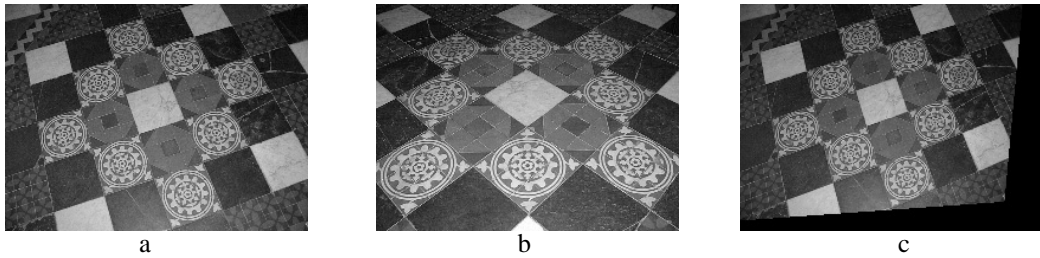


Fig. 4.5. Three images of a plane which are used to compare methods of computing projective transformations from corresponding points.

| Method | Pair 1 figure 4.5 a & b | Pair 2 figure 4.5 a & c |
|---------------------|----------------------------|----------------------------|
| Linear normalized | 0.4078 | 0.6602 |
| Gold Standard | 0.4078 | 0.6602 |
| Linear unnormalized | 0.4080 | 26.2056 |
| Homogeneous scaling | 0.5708 | 0.7421 |
| Sampson | 0.4077 | 0.6602 |
| Error in 1 view | 0.4077 | 0.6602 |
| Affine | 6.0095 | 2.8481 |
| Theoretical optimal | 0.5477 | 0.6582 |

Table 4.1. Residual errors in pixels for the various algorithms.

4.6 Experimental comparison of the algorithms

The algorithms are compared for the images shown in figure 4.5. Table 4.1 shows the results of testing several of the algorithms described in this chapter. Residual error is shown for two pairs of images. The methods used are fairly self-explanatory, with a few exceptions. The method “affine” was an attempt to fit the projective transformation with an optimal affine mapping. The “optimal” is the ML estimate assuming a noise level of one pixel.

The first pair of images are (a) and (b) of figure 4.5, with 55 point correspondences. It appears that all methods work almost equally well (except the affine method). The optimal residual is greater than the achieved results, because the noise level (unknown) is less than one pixel.

Image (c) of figure 4.5 was produced synthetically by resampling (a), and the second pair consists of (a) and (c) with 20 point correspondences. In this case, almost all methods perform almost optimally, as shown in the table 4.1. The exception is the affine method (expected to perform badly, since it is not an affine transformation) and the unnormalized linear method. The unnormalized method is expected to perform badly (though maybe not this badly). Just why it performs well in the first pair and very badly for the second pair is not understood. In any case, it is best to avoid this method and use a normalized linear or Gold Standard method.

A further evaluation is presented in figure 4.6. The transformation to be estimated is the one that maps the chessboard image shown here to a square grid aligned with the

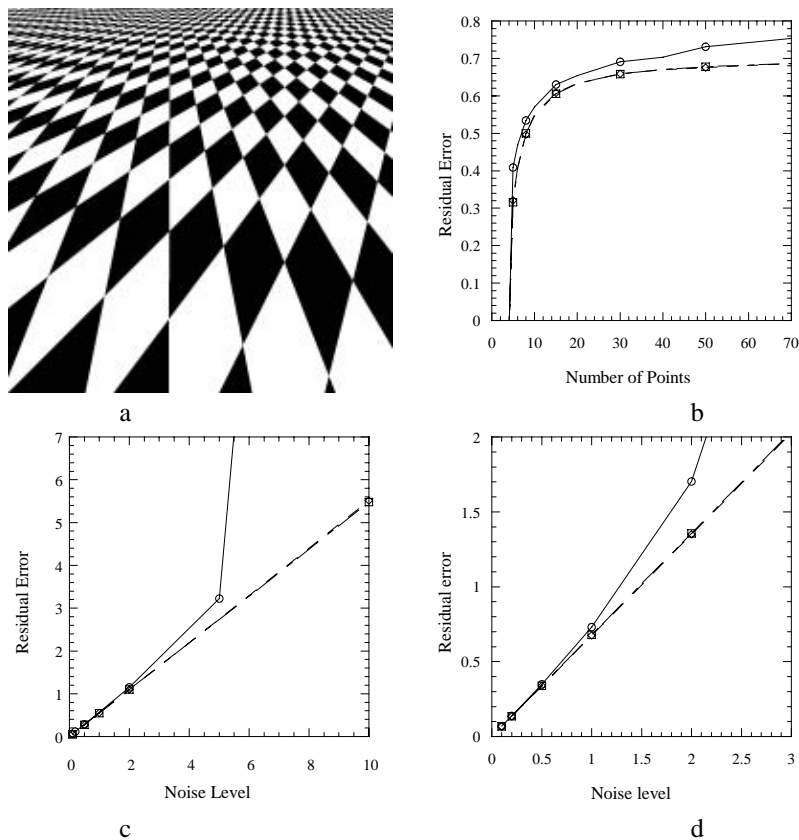


Fig. 4.6. **Comparison of the DLT and Gold Standard algorithms to the theoretically optimal residual error.** (a) The homography is computed between a chessboard and this image. In all three graphs, the result for the Gold Standard algorithm overlap and are indistinguishable from the theoretical minimum. (b) Residual error as a function of the number of points. (c) The effect of varying noise level for 10 points, and (d) 50 points.

axes. As may be seen, the image is substantially distorted, with respect to a square grid. For the experiments, randomly selected points in the image were chosen and matched with the corresponding point on the square grid. The (normalized) DLT algorithm and the Gold Standard algorithm are compared to the theoretical minimum or residual error (see chapter 5). Note that for noise up to 5 pixels, the DLT algorithm performs adequately. However, for a noise level of 10 pixels it fails. Note however that in a 200-pixel image, an error of 10 pixels is extremely high. For less severe homographies, closer to the identity map, the DLT performs almost as well as the Gold Standard algorithm.

4.7 Robust estimation

Up to this point it has been assumed that we have been presented with a set of correspondences, $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, where the only source of error is in the measurement of the point's position, which follows a Gaussian distribution. In many practical situations this assumption is not valid because points are mismatched. The mismatched points are *outliers* to the Gaussian error distribution. These outliers can severely disturb the

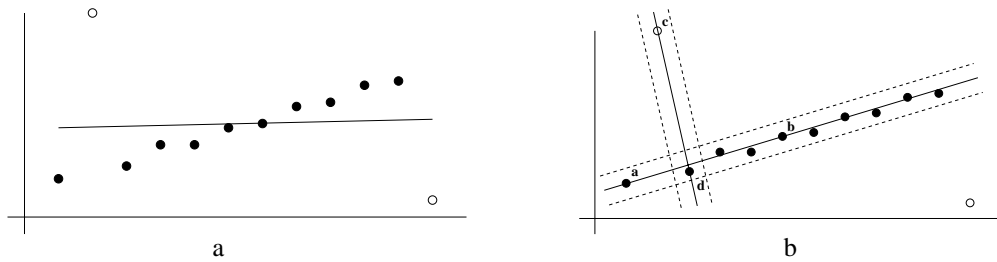


Fig. 4.7. **Robust line estimation.** The solid points are inliers, the open points outliers. (a) A least-squares (orthogonal regression) fit to the point data is severely affected by the outliers. (b) In the RANSAC algorithm the support for lines through randomly selected point pairs is measured by the number of points within a threshold distance of the lines. The dotted lines indicate the threshold distance. For the lines shown the support is 10 for line $\langle \mathbf{a}, \mathbf{b} \rangle$ (where both of the points \mathbf{a} and \mathbf{b} are inliers); and 2 for line $\langle \mathbf{c}, \mathbf{d} \rangle$ where the point \mathbf{c} is an outlier.

estimated homography, and consequently should be identified. The goal then is to determine a set of *inliers* from the presented “correspondences” so that the homography can then be estimated in an optimal manner from these inliers using the algorithms described in the previous sections. This is *robust estimation* since the estimation is robust (tolerant) to outliers (measurements following a different, and possibly unmodelled, error distribution).

4.7.1 RANSAC

We start with a simple example that can easily be visualized – estimating a straight line fit to a set of 2-dimensional points. This can be thought of as estimating a 1-dimensional affine transformation, $x' = ax + b$, between corresponding points lying on two lines.

The problem, which is illustrated in figure 4.7a, is the following: given a set of 2D data points, find the line which minimizes the sum of squared perpendicular distances (orthogonal regression), subject to the condition that none of the valid points deviates from this line by more than t units. This is actually two problems: a line fit to the data; and a classification of the data into inliers (valid points) and outliers. The threshold t is set according to the measurement noise (for example $t = 3\sigma$), and is discussed below. There are many types of robust algorithms and which one to use depends to some extent on the proportion of outliers. For example, if it is known that there is only one outlier, then each point can be deleted in turn and the line estimated from the remainder. Here we describe in detail a general and very successful robust estimator – the RANdom SAMple Consensus (RANSAC) algorithm of Fischler and Bolles [Fischler-81]. The RANSAC algorithm is able to cope with a large proportion of outliers.

The idea is very simple: two of the points are selected randomly; these points define a line. The *support* for this line is measured by the number of points that lie within a distance threshold. This random selection is repeated a number of times and the line with most support is deemed the robust fit. The points within the threshold distance are the inliers (and constitute the eponymous *consensus* set). The intuition is that if one of the points is an outlier then the line will not gain much support, see figure 4.7b.

Furthermore, scoring a line by its support has the additional advantage of favouring better fits. For example, the line $\langle a, b \rangle$ in figure 4.7b has a support of 10, whereas the line $\langle a, d \rangle$, where the sample points are neighbours, has a support of only 4. Consequently, even though both samples contain no outliers, the line $\langle a, b \rangle$ will be selected.

More generally, we wish to fit a *model*, in this case a line, to data, and the random *sample* consists of a minimal subset of the data, in this case two points, sufficient to determine the model. If the model is a planar homography, and the data a set of 2D point correspondences, then the minimal subset consists of four correspondences. The application of RANSAC to the estimation of a homography is described below.

As stated by Fischler and Bolles [Fischler-81] “The RANSAC procedure is opposite to that of conventional smoothing techniques: Rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate the invalid data points, RANSAC uses as small an initial data set as feasible and enlarges this set with consistent data when possible”.

The RANSAC algorithm is summarized in algorithm 4.4. Three questions immediately arise:

Objective

Robust fit of a model to a data set S which contains outliers.

Algorithm

- (i) Randomly select a sample of s data points from S and instantiate the model from this subset.
- (ii) Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S .
- (iii) If the size of S_i (the number of inliers) is greater than some threshold T , re-estimate the model using all the points in S_i and terminate.
- (iv) If the size of S_i is less than T , select a new subset and repeat the above.
- (v) After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i .

Algorithm 4.4. *The RANSAC robust estimation algorithm, adapted from [Fischler-81]. A minimum of s data points are required to instantiate the free parameters of the model. The three algorithm thresholds t , T , and N are discussed in the text.*

1. What is the distance threshold? We would like to choose the distance threshold, t , such that with a probability α the point is an inlier. This calculation requires the probability distribution for the distance of an inlier from the model. In practice the distance threshold is usually chosen empirically. However, if it is assumed that the measurement error is Gaussian with zero mean and standard deviation σ , then a value for t may be computed. In this case the square of the point distance, d_{\perp}^2 , is a sum of squared Gaussian variables and follows a χ_m^2 distribution with m degrees of freedom, where m equals the codimension of the model. For a line the codimension is 1 – only the perpendicular distance to the line is measured. If the model is a point the codimension is 2, and the square of the distance is the sum of squared x and y measurement errors.

The probability that the value of a χ_m^2 random variable is less than k^2 is given by the cumulative chi-squared distribution, $F_m(k^2) = \int_0^{k^2} \chi_m^2(\xi) d\xi$. Both of these distributions are described in section A2.2(p566). From the cumulative distribution

$$\begin{cases} \text{inlier} & d_{\perp}^2 < t^2 \\ \text{outlier} & d_{\perp}^2 \geq t^2 \end{cases} \text{ with } t^2 = F_m^{-1}(\alpha)\sigma^2. \quad (4.17)$$

Usually α is chosen as 0.95, so that there is a 95% probability that the point is an inlier. This means that an inlier will only be incorrectly rejected 5% of the time. Values of t for $\alpha = 0.95$ and for the models of interest in this book are tabulated in table 4.2.

| Codimension m | Model | t^2 |
|-----------------|---------------------------|-----------------|
| 1 | line, fundamental matrix | $3.84 \sigma^2$ |
| 2 | homography, camera matrix | $5.99 \sigma^2$ |
| 3 | trifocal tensor | $7.81 \sigma^2$ |

Table 4.2. The distance threshold $t^2 = F_m^{-1}(\alpha)\sigma^2$ for a probability of $\alpha = 0.95$ that the point (correspondence) is an inlier.

2. How many samples? It is often computationally infeasible and unnecessary to try every possible sample. Instead the number of samples N is chosen sufficiently high to ensure with a probability, p , that at least one of the random samples of s points is free from outliers. Usually p is chosen at 0.99. Suppose w is the probability that any selected data point is an inlier, and thus $\epsilon = 1 - w$ is the probability that it is an outlier. Then at least N selections (each of s points) are required, where $(1 - w^s)^N = 1 - p$, so that

$$N = \log(1 - p) / \log(1 - (1 - \epsilon)^s). \quad (4.18)$$

Table 4.3 gives examples of N for $p = 0.99$ for a given s and ϵ .

| Sample size | | Proportion of outliers ϵ | | | | | | |
|-------------|----|-----------------------------------|-----|-----|-----|-----|------|--|
| s | 5% | 10% | 20% | 25% | 30% | 40% | 50% | |
| 2 | 2 | 3 | 5 | 6 | 7 | 11 | 17 | |
| 3 | 3 | 4 | 7 | 9 | 11 | 19 | 35 | |
| 4 | 3 | 5 | 9 | 13 | 17 | 34 | 72 | |
| 5 | 4 | 6 | 12 | 17 | 26 | 57 | 146 | |
| 6 | 4 | 7 | 16 | 24 | 37 | 97 | 293 | |
| 7 | 4 | 8 | 20 | 33 | 54 | 163 | 588 | |
| 8 | 5 | 9 | 26 | 44 | 78 | 272 | 1177 | |

Table 4.3. The number N of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, s , and proportion of outliers, ϵ .

Example 4.4. For the line-fitting problem of figure 4.7 there are $n = 12$ data points, of

which two are outliers so that $\epsilon = 2/12 = 1/6$. From table 4.3 for a minimal subset of size $s = 2$, at least $N = 5$ samples are required. This should be compared with the cost of exhaustively trying every point pair, in which case $\binom{12}{2} = 66$ samples are required (the notation $\binom{n}{2}$ means the number of choices of 2 among n , specifically, $\binom{n}{2} = n(n-1)/2$). \triangle

Note

- (i) The number of samples is linked to the proportion rather than number of outliers. This means that the number of samples required may be smaller than the number of outliers. Consequently the computational cost of the sampling can be acceptable even when the number of outliers is large.
- (ii) The number of samples increases with the size of the minimal subset (for a given ϵ and p). It might be thought that it would be advantageous to use more than the minimal subset, three or more points in the case of a line, because then a better estimate of the line would be obtained, and the measured support would more accurately reflect the true support. However, this possible advantage in measuring support is generally outweighed by the severe increase in computational cost incurred by the increase in the number of samples.

3. How large is an acceptable consensus set? A rule of thumb is to terminate if the size of the consensus set is similar to the number of inliers believed to be in the data set, given the assumed proportion of outliers, i.e. for n data points $T = (1 - \epsilon)n$. For the line-fitting example of figure 4.7 a conservative estimate of ϵ is $\epsilon = 0.2$, so that $T = (1.0 - 0.2)12 = 10$.

Determining the number of samples adaptively. It is often the case that ϵ , the fraction of data consisting of outliers, is unknown. In such cases the algorithm is initialized using a worst case estimate of ϵ , and this estimate can then be updated as larger consistent sets are found. For example, if the worst case guess is $\epsilon = 0.5$ and a consensus set with 80% of the data is found as inliers, then the updated estimate is $\epsilon = 0.2$.

This idea of “probing” the data via the consensus sets can be applied repeatedly in order to adaptively determine the number of samples, N . To continue the example above, the worst case estimate of $\epsilon = 0.5$ determines an initial N according to (4.18). When a consensus set containing more than 50% of the data is found, we then know that there is at least that proportion of inliers. This updated estimate of ϵ determines a reduced N from (4.18). This update is repeated at each sample, and whenever a consensus set with ϵ lower than the current estimate is found, then N is again reduced. The algorithm terminates as soon as N samples have been performed. It may occur that a sample is found for which ϵ determines an N less than the number of samples that have already been performed. In such a case sufficient samples have been performed and the algorithm terminates. In pseudo-code the adaptive computation of N is summarized in algorithm 4.5.

This adaptive approach works very well and in practice covers the questions of both

- $N = \infty$, sample_count = 0.
- While $N > \text{sample_count}$ Repeat
 - Choose a sample and count the number of inliers.
 - Set $\epsilon = 1 - (\text{number of inliers})/(\text{total number of points})$
 - Set N from ϵ and (4.18) with $p = 0.99$.
 - Increment the sample_count by 1.
- Terminate.

Algorithm 4.5. Adaptive algorithm for determining the number of RANSAC samples.

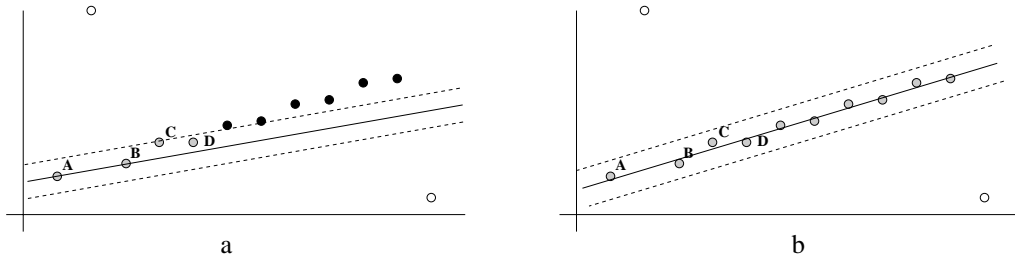


Fig. 4.8. **Robust ML estimation.** The grey points are classified as inliers to the line. (a) A line defined by points $\langle A, B \rangle$ has a support of four (from points $\{A, B, C, D\}$). (b) The ML line fit (orthogonal least-squares) to the four points. This is a much improved fit over that defined by $\langle A, B \rangle$. 10 points are classified as inliers.

the number of samples and terminating the algorithm. The initial ϵ can be chosen as 1.0, in which case the initial N will be infinite. It is wise to use a conservative probability p such as 0.99 in (4.18). Table 4.4 on page 127 gives example ϵ 's and N 's when computing a homography.

4.7.2 Robust Maximum Likelihood estimation

The RANSAC algorithm partitions the data set into inliers (the largest consensus set) and outliers (the rest of the data set), and also delivers an estimate of the model, M_0 , computed from the minimal set with greatest support. The final step of the RANSAC algorithm is to re-estimate the model using all the inliers. This re-estimation should be optimal and will involve minimizing a ML cost function, as described in section 4.3. In the case of a line, ML estimation is equivalent to orthogonal regression, and a closed form solution is available. In general, though, the ML estimation involves iterative minimization, and the minimal set estimate, M_0 , provides the starting point.

The only drawback with this procedure, which is often the one adopted, is that the inlier–outlier classification is irrevocable. After the model has been optimally fitted to the consensus set, there may well be additional points which would now be classified as inliers if the distance threshold was applied to the new model. For example, suppose the line $\langle A, B \rangle$ in figure 4.8 was selected by RANSAC. This line has a support of four points, all inliers. After the optimal fit to these four points, there are now 10 points which would correctly be classified as inliers. These two steps: optimal fit to inliers; re-classify inliers using (4.17); can then be iterated until the number of inliers converges.

A least-squares fit with inliers weighted by their distance to the model is often used at this stage.

Robust cost function. An alternative to minimizing $\mathcal{C} = \sum_i d_{\perp i}^2$ over the inliers is to minimize a robust version including all data. A suitable robust cost function is

$$\mathcal{D} = \sum_i \gamma(d_{\perp i}) \quad \text{with } \gamma(e) = \begin{cases} e^2 & e^2 < t^2 \text{ inlier} \\ t^2 & e^2 \geq t^2 \text{ outlier} \end{cases} \quad (4.19)$$

Here $d_{\perp i}$ are point errors and $\gamma(e)$ is a robust cost function [Huber-81] where outliers are given a fixed cost. The χ^2 motivation for the threshold is the same as that of (4.17), where t^2 is defined. The quadratic cost for inliers arises from the Gaussian error model, as described in section 4.3. The constant cost for outliers in the robust cost function arises from the assumption that outliers follow a diffuse or uniform distribution, the log-likelihood of which is a constant. It might be thought that outliers could be excluded from the cost function by simply thresholding on $d_{\perp i}$. The problem with thresholding alone is that it would result in only outliers being included because they would incur no cost.

The cost function \mathcal{D} allows the minimization to be conducted on all points whether they are outliers or inliers. At the start of the iterative minimization \mathcal{D} differs from \mathcal{C} only by a constant (given by 4 times the number of outliers). However, as the minimization progresses outliers can be redesignated inliers, and this typically occurs in practice. A discussion and comparison of cost functions is given in appendix A6.8-(p616).

4.7.3 Other robust algorithms

In RANSAC a model instantiated from a minimal set is scored by the number of data points within a threshold distance. An alternative is to score the model by the median of the distances to all points in the data. The model with least median is then selected. This is Least Median of Squares (LMS) estimation, where, as in RANSAC, minimum size subset samples are selected randomly with the number of samples obtained from (4.18). The advantage of LMS is that it requires *no* setting of thresholds or *a priori* knowledge of the variance of the error. The disadvantage of LMS is that it fails if more than half the data is outlying, for then the median distance will be to an outlier. The solution is to use the proportion of outliers to determine the selection distance. For example if there are 50% outliers then a distance below the median value (the quartile say) should be used.

Both the RANSAC and LMS algorithms are able to cope with a large proportion of outliers. If the number of outliers is small, then other robust methods may well be more efficient. These include case deletion, where each point in turn is deleted and the model fitted to the remaining data; and iterative weighted least-squares, where a data point's influence on the fit is weighted inversely by its residual. Generally these methods are **not** recommended. Both Torr [Torr-95b] and Xu and Zhang [Xu-96] describe and compare various robust estimators for estimating the fundamental matrix.

Objective

Compute the 2D homography between two images.

Algorithm

- (i) **Interest points:** Compute interest points in each image.
- (ii) **Putative correspondences:** Compute a set of interest point matches based on proximity and similarity of their intensity neighbourhood.
- (iii) **RANSAC robust estimation:** Repeat for N samples, where N is determined adaptively as in algorithm 4.5:
 - (a) Select a random sample of 4 correspondences and compute the homography H .
 - (b) Calculate the distance d_{\perp} for each putative correspondence.
 - (c) Compute the number of inliers consistent with H by the number of correspondences for which $d_{\perp} < t = \sqrt{5.99} \sigma$ pixels.

Choose the H with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.

- (iv) **Optimal estimation:** re-estimate H from all correspondences classified as inliers, by minimizing the ML cost function (4.8–p95) using the Levenberg–Marquardt algorithm of section A6.2(p600).
- (v) **Guided matching:** Further interest point correspondences are now determined using the estimated H to define a search region about the transferred point position.

The last two steps can be iterated until the number of correspondences is stable.

Algorithm 4.6. *Automatic estimation of a homography between two images using RANSAC.*

4.8 Automatic computation of a homography

This section describes an algorithm to automatically compute a homography between two images. The input to the algorithm is simply the images, with no other *a priori* information required; and the output is the estimated homography together with a set of interest points in correspondence. The algorithm might be applied, for example, to two images of a planar surface or two images acquired by rotating a camera about its centre.

The first step of the algorithm is to compute interest points in each image. We are then faced with a “chicken and egg” problem: once the correspondence between the interest points is established the homography can be computed; conversely, given the homography the correspondence between the interest points can easily be established. This problem is resolved by using robust estimation, here RANSAC, as a “search engine”. The idea is first to obtain by some means a set of putative point correspondences. It is expected that a proportion of these correspondences will in fact be mismatches. RANSAC is designed to deal with exactly this situation – estimate the homography and also a set of inliers consistent with this estimate (the true correspondences), and outliers (the mismatches).

The algorithm is summarized in algorithm 4.6, with an example of its use shown in figure 4.9, and the steps described in more detail below. Algorithms with essentially the same methodology enable the automatic computation of the fundamental matrix and trifocal tensor directly from image pairs and triplets respectively. This computation is described in chapter 11 and chapter 16.

Determining putative correspondences. The aim, in the absence of any knowledge of the homography, is to provide an initial point correspondence set. A good proportion of these correspondences should be correct, but the aim is not perfect matching, since RANSAC will later be used to eliminate the mismatches. Think of these as “seed” correspondences. These putative correspondences are obtained by detecting interest points independently in each image, and then matching these interest points using a combination of proximity and similarity of intensity neighbourhoods as follows. For brevity, the interest points will be referred to as ‘corners’. However, these corners need not be images of physical corners in the scene. The corners are defined by a minimum of the image auto-correlation function.

For each corner at (x, y) in image 1 the match with highest neighbourhood cross-correlation in image 2 is selected within a square search region centred on (x, y) . Symmetrically, for each corner in image 2 the match is sought in image 1. Occasionally there will be a conflict where a corner in one image is “claimed” by more than one corner in the other. In such cases a “winner takes all” scheme is applied and only the match with highest cross-correlation is retained.

A variation on the similarity measure is to use Squared Sum of intensity Differences (SSD) instead of (normalized) Cross-Correlation (CC). CC is invariant to the affine mapping of the intensity values (i.e. $I \mapsto \alpha I + \beta$, scaling plus offset) which often occurs in practice between images. SSD is not invariant to this mapping. However, SSD is often preferred when there is small variation in intensity between images, because it is a more sensitive measure than CC and is computationally cheaper.

RANSAC for a homography. The RANSAC algorithm is applied to the putative correspondence set to estimate the homography and the (inlier) correspondences which are consistent with this estimate. The sample size is four, since four correspondences determine a homography. The number of samples is set adaptively as the proportion of outliers is determined from each consensus set, as described in algorithm 4.5.

There are two issues: what is the “distance” in this case? and how should the samples be selected?

- (i) **Distance measure:** The simplest method of assessing the error of a correspondence from a homography H is to use the symmetric transfer error, i.e. $d_{\text{transfer}}^2 = d(\mathbf{x}, H^{-1}\mathbf{x}')^2 + d(\mathbf{x}', H\mathbf{x})^2$, where $\mathbf{x} \leftrightarrow \mathbf{x}'$ is the point correspondence. A better, though more expensive, distance measure is the reprojection error, $d_{\perp}^2 = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$, where $\hat{\mathbf{x}}' = H\hat{\mathbf{x}}$ is the perfect correspondence. This measure is more expensive because $\hat{\mathbf{x}}$ must also be computed. A further alternative is Sampson error.
- (ii) **Sample selection:** There are two issues here. First, degenerate samples should be disregarded. For example, if three of the four points are collinear then a homography cannot be computed; second, the sample should consist of points with a good spatial distribution over the image. This is because of the extrapolation problem – an estimated homography will accurately map the region straddled by the computation points, but the accuracy generally deteriorates

with distance from this region (think of four points in the very top corner of the image). Distributed spatial sampling can be implemented by tiling the image and ensuring, by a suitable weighting of the random sampler, that samples with points lying in different tiles are the more likely.

Robust ML estimation and guided matching. The aim of this final stage is two-fold: first, to obtain an improved estimate of the homography by using all the inliers in the estimation (rather than only the four points of the sample); second, to obtain more inlying matches from the putative correspondence set because a more accurate homography is available. An improved estimate of the homography is then computed from the inliers by minimizing an ML cost function. This final stage can be implemented in two ways. One way is to carry out an ML estimation on the inliers, then recompute the inliers using the new estimated H , and repeat this cycle until the number of inliers converges. The ML cost function minimization is carried out using the Levenberg–Marquardt algorithm described in section A6.2(p600). The alternative is to estimate the homography and inliers simultaneously by minimizing a robust ML cost function of (4.19) as described in section 4.7.2. The disadvantage of the simultaneous approach is the computational effort incurred in the minimization of the cost function. For this reason the cycle approach is usually the more attractive.

4.8.1 Application domain

The algorithm requires that interest points can be recovered fairly uniformly across the image, and this in turn requires scenes and resolutions which support this requirement. Scenes should be lightly textured – images of blank walls are not ideal.

The search window proximity constraint places an upper limit on the image motion of corners (the *disparity*) between views. However, the algorithm is not defeated if this constraint is not applied, and in practice the main role of the proximity constraint is to reduce computational complexity, as a smaller search window means that fewer corner matches must be evaluated.

Ultimately the scope of the algorithm is limited by the success of the corner neighbourhood similarity measure (SSD or CC) in providing disambiguation between correspondences. Failure generally results from lack of spatial invariance: the measures are only invariant to image translation, and are severely degraded by transformations outside this class such as image rotation or significant differences in foreshortening between images. One solution is to use measures with a greater invariance to the homography mapping between images, for example measures which are rotationally invariant. An alternative solution is to use an initial estimate of the homography to map between intensity neighbourhoods. Details are beyond the scope of this discussion, but are provided in [Pritchett-98, Schmid-98]. The use of robust estimation confers moderate immunity to independent motion, changes in shadows, partial occlusions etc.

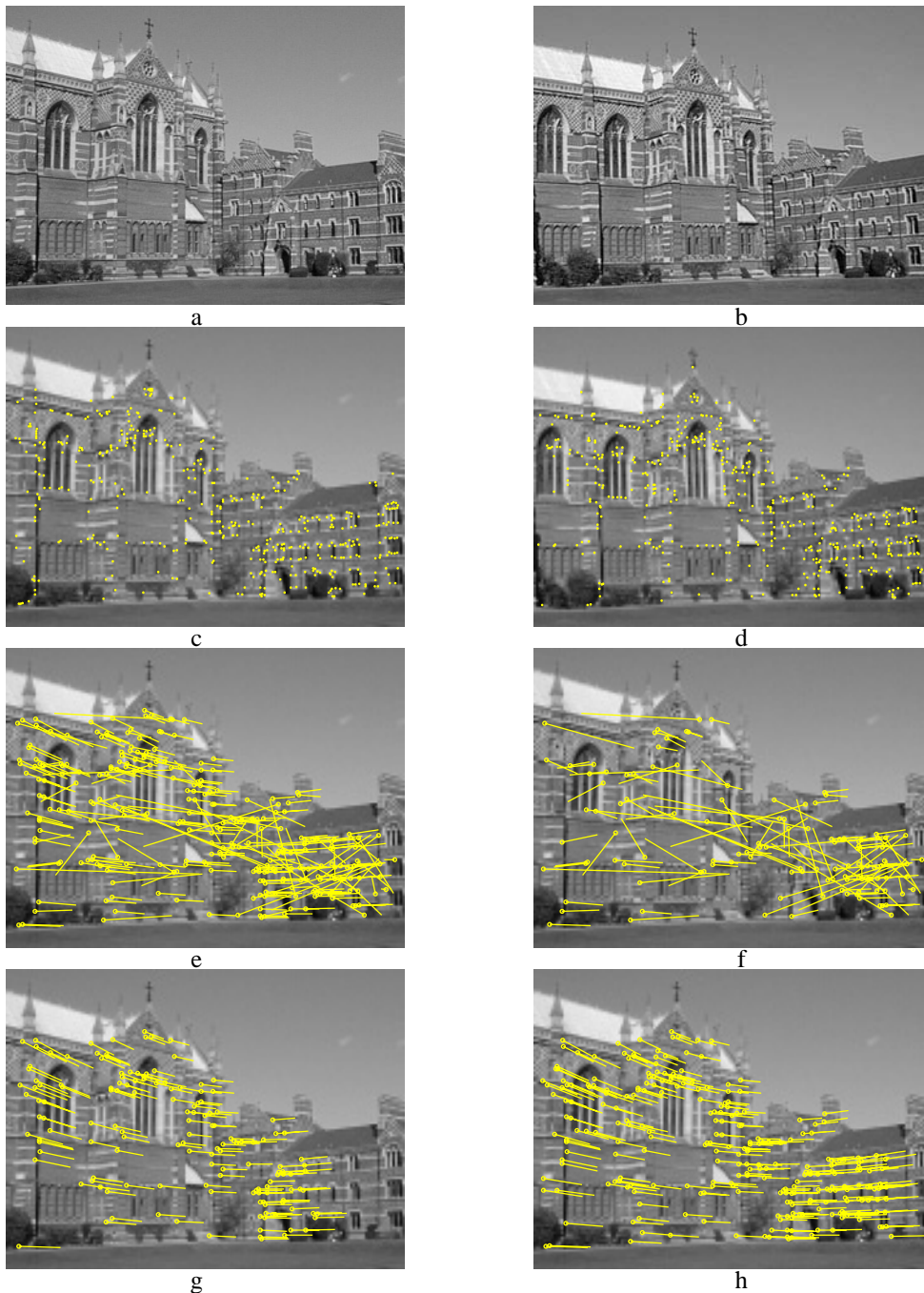


Fig. 4.9. Automatic computation of a homography between two images using RANSAC. *The motion between views is a rotation about the camera centre so the images are exactly related by a homography. (a) (b) left and right images of Keble College, Oxford. The images are 640×480 pixels. (c) (d) detected corners superimposed on the images. There are approximately 500 corners on each image. The following results are superimposed on the left image: (e) 268 putative matches shown by the line linking corners, note the clear mismatches; (f) outliers – 117 of the putative matches; (g) inliers – 151 correspondences consistent with the estimated H ; (h) final set of 262 correspondences after guided matching and MLE.*

4.8.2 Implementation and run details

Interest points are obtained using the Harris [Harris-88] corner detector. This detector localizes corners to sub-pixel accuracy, and it has been found empirically that the correspondence error is usually less than a pixel [Schmid-98].

When obtaining seed correspondences, in the putative correspondence stage of the algorithm, the threshold on the neighbourhood similarity measure for match acceptance is deliberately conservative to minimize incorrect matches (the SSD threshold is 20). For the guided matching stage this threshold is relaxed (it is doubled) so that additional putative correspondences are available.

| Number of inliers | $1 - \epsilon$ | Adaptive N |
|----------------------|----------------|-----------------|
| 6 | 2% | 20,028,244 |
| 10 | 3% | 2,595,658 |
| 44 | 16% | 6,922 |
| 58 | 21% | 2,291 |
| 73 | 26% | 911 |
| 151 | 56% | 43 |

Table 4.4. *The results of the adaptive algorithm 4.5 used during RANSAC to compute the homography for figure 4.9. N is the total number of samples required as the algorithm runs for $p = 0.99$ probability of no outliers in the sample. The algorithm terminated after 43 samples.*

For the example of figure 4.9 the images are 640×480 pixels, and the search window ± 320 pixels, i.e. the entire image. Of course a much smaller search window could have been used given the actual point disparities in this case. Often in video sequences a search window of ± 40 pixels suffices (i.e. a square of side 80 centred on the current position). The inlier threshold was $t = 1.25$ pixels.

A total of 43 samples were required, with the sampling run as shown in table 4.4. The guided matching required two iterations of the MLE–inlier classification cycle. The RMS values for d_{\perp} pixel error were 0.23 before the MLE and 0.19 after. The Levenberg–Marquardt algorithm required 10 iterations.

4.9 Closure

This chapter has illustrated the issues and techniques that apply to estimating the tensors representing multiple view relations. These ideas will reoccur in each of the computation chapters throughout the book. In each case there are a minimal number of correspondences required; degenerate configurations that should be avoided; algebraic and geometric errors that can be minimized when more than the minimal number of correspondences are available; parametrizations that enforce internal constraints on the tensor etc.

4.9.1 The literature

The DLT algorithm dates back at least to Sutherland [Sutherland-63]. Sampson’s classic paper on conic fitting (an improvement on the equally classic Bookstein algorithm)

appeared in [Sampson-82]. Normalization was made public in the Computer Vision literature by Hartley [Hartley-97c].

Related reading on numerical methods may be found in the excellent *Numerical Recipes in C* [Press-88], and also Gill and Murray [Gill-78] for iterative minimization.

Fischler and Bolles' [Fischler-81] RANSAC was one of the earliest robust algorithms, and in fact was developed to solve a Computer Vision problem (pose from 3 points). The original paper is very clearly argued and well worth reading. Other background material on robustness may be found in Rousseeuw [Rousseeuw-87]. The primary application of robust estimation in computer vision was to estimating the fundamental matrix (chapter 11), by Torr and Murray [Torr-93] using RANSAC, and, Zhang *et al.* [Zhang-95] using LMS. The automatic ML estimation of a homography was described by Torr and Zisserman [Torr-98].

4.9.2 Notes and exercises

- (i) **Computing homographies of \mathbb{P}^n .** The derivation of (4.1–p89) and (4.3–p89) assumed that the dimension of \mathbf{x}'_i is three, so that the cross-product is defined. However, (4.3) may be derived in a way that generalizes to all dimensions. Assuming that $w'_i = 1$, we may solve for the unknown scale factor explicitly by writing $\mathbf{H}\mathbf{x}_i = k(x_i, y_i, 1)^\top$. From the third coordinate we obtain $k = \mathbf{h}^{3\top}\mathbf{x}_i$, and substituting this into the original equation gives

$$\begin{pmatrix} \mathbf{h}^{1\top}\mathbf{x}_i \\ \mathbf{h}^{2\top}\mathbf{x}_i \end{pmatrix} = \begin{pmatrix} x'_i \mathbf{h}^{3\top}\mathbf{x}_i \\ y'_i \mathbf{h}^{3\top}\mathbf{x}_i \end{pmatrix}$$

which leads directly to (4.3).

- (ii) **Computing homographies for ideal points.** If one of the points \mathbf{x}'_i is an ideal point, so that $w'_i = 0$, then the pair of equations (4.3) collapses to a single equation although (4.1) does contain two independent equations. To avoid such degeneracy, while including only the minimum number of equations, a good way to proceed is as follows. We may rewrite the equation $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$ as

$$[\mathbf{x}'_i]^\perp \mathbf{H}\mathbf{x}_i = \mathbf{0}$$

where $[\mathbf{x}'_i]^\perp$ is a matrix with rows orthogonal to \mathbf{x}'_i so that $[\mathbf{x}'_i]^\perp \mathbf{x}'_i = \mathbf{0}$. Each row of $[\mathbf{x}'_i]^\perp$ leads to a separate linear equation in the entries of \mathbf{H} . The matrix $[\mathbf{x}'_i]^\perp$ may be obtained by deleting the first row of an orthogonal matrix \mathbf{M} satisfying $\mathbf{M}\mathbf{x}'_i = (1, 0, \dots, 0)^\top$. A Householder matrix (see section A4.1.2(p580)) is an easily constructed matrix with the desired property.

- (iii) **Scaling unbounded point sets.** In the case of points at or near infinity in a plane, it is neither reasonable nor feasible to normalize coordinates using the isotropic (or non-isotropic) scaling schemes presented in this chapter, since the centroid and scale are infinite or near infinite. A method that seems to give good results is to normalize the set of points $\mathbf{x}_i = (x_i, y_i, w_i)^\top$ such that

$$\sum_i x_i = \sum_i y_i = 0 ; \quad \sum_i x_i^2 + y_i^2 = 2 \sum_i w_i^2 ; \quad x_i^2 + y_i^2 + w_i^2 = 1 \forall i$$

Note that the coordinates x_i and y_i appearing here are the homogeneous coordinates, and the conditions no longer imply that the centroid is at the origin. Investigate methods of achieving this normalization, and evaluate its properties.

- (iv) **Transformation invariance of DLT.** We consider computation of a 2D homography by minimizing algebraic error $\|A\mathbf{h}\|$ (see (4.5–p94)) subject to various constraints. Prove the following cases:

- (a) If $\|A\mathbf{h}\|$ is minimized subject to the constraint $h_9 = H_{33} = 1$, then the result is invariant under change of scale but *not* translation of coordinates.
- (b) If instead the constraint is $H_{31}^2 + H_{32}^2 = 1$ then the result is similarity invariant.
- (c) **Affine case:** The same is true for the constraint $H_{31} = H_{32} = 0$; $H_{33} = 1$.

- (v) **Expressions for image coordinate derivatives.** For the map $\mathbf{x}' = (x', y', w')^T = H\mathbf{x}$, derive the following expressions (where $\tilde{\mathbf{x}}' = (\tilde{x}', \tilde{y}')^T = (x'/w', y'/w')^T$ are the inhomogeneous coordinates of the image point):

- (a) Derivative wrt \mathbf{x}

$$\partial \tilde{\mathbf{x}}' / \partial \mathbf{x} = \frac{1}{w'} \begin{bmatrix} \mathbf{h}^{1T} - \tilde{x}' \mathbf{h}^{3T} \\ \mathbf{h}^{2T} - \tilde{y}' \mathbf{h}^{3T} \end{bmatrix} \quad (4.20)$$

where \mathbf{h}^{jT} is the j -th row of H .

- (b) Derivative wrt H

$$\partial \tilde{\mathbf{x}}' / \partial \mathbf{h} = \frac{1}{w'} \begin{bmatrix} \mathbf{x}^T & 0 & -\tilde{x}' \mathbf{x}^T \\ 0 & \mathbf{x}^T & -\tilde{y}' \mathbf{x}^T \end{bmatrix} \quad (4.21)$$

with \mathbf{h} as defined in (4.2–p89).

- (vi) **Sampson error with non-isotropic error distributions.** The derivation of Sampson error in section 4.2.6(p98) assumed that points were measured with circular error distributions. In the case where the point $\mathbf{X} = (x, y, x', y')$ is measured with covariance matrix $\Sigma_{\mathbf{X}}$ it is appropriate instead to minimize the Mahalanobis norm $\|\delta_{\mathbf{X}}\|_{\Sigma_{\mathbf{X}}}^2 = \delta_{\mathbf{X}}^T \Sigma_{\mathbf{X}}^{-1} \delta_{\mathbf{X}}$. Show that in this case the formulae corresponding to (4.11–p99) and (4.12–p99) are

$$\delta_{\mathbf{X}} = -\Sigma_{\mathbf{X}} \mathbf{J}^T (\mathbf{J} \Sigma_{\mathbf{X}} \mathbf{J}^T)^{-1} \epsilon \quad (4.22)$$

and

$$\|\delta_{\mathbf{X}}\|_{\Sigma_{\mathbf{X}}}^2 = \epsilon^T (\mathbf{J} \Sigma_{\mathbf{X}} \mathbf{J}^T)^{-1} \epsilon. \quad (4.23)$$

Note that if the measurements in the two images are independent, then the covariance matrix $\Sigma_{\mathbf{X}}$ will be block-diagonal with two 2×2 diagonal blocks corresponding to the two images.

- (vii) **Sampson error programming hint.** In the case of 2D homography estimation, and in fact every other similar problem considered in this book, the cost function $\mathcal{C}_{\mathbf{H}}(\mathbf{X}) = A(\mathbf{X})\mathbf{h}$ of section 4.2.6(p98) is multilinear in the coordinates

Objective

Given $n \geq 4$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the affine homography H_A which minimizes reprojection error in both images (4.8–p95).

Algorithm

- (a) Express points as inhomogeneous 2-vectors. Translate the points \mathbf{x}_i by a translation \mathbf{t} so that their centroid is at the origin. Do the same to the points \mathbf{x}'_i by a translation \mathbf{t}' . Henceforth work with the translated coordinates.
- (b) Form the $n \times 4$ matrix A whose rows are the vectors

$$\mathbf{X}_i^T = (\mathbf{x}_i^T, \mathbf{x}'_i^T) = (x_i, y_i, x'_i, y'_i).$$

- (c) Let \mathbf{V}_1 and \mathbf{V}_2 be the right singular-vectors of A corresponding to the two largest (sic) singular values.
- (d) Let $H_{2 \times 2} = CB^{-1}$, where B and C are the 2×2 blocks such that

$$[\mathbf{V}_1 \mathbf{V}_2] = \begin{bmatrix} B \\ C \end{bmatrix}.$$

- (e) The required homography is

$$H_A = \begin{bmatrix} H_{2 \times 2} & H_{2 \times 2}\mathbf{t} - \mathbf{t}' \\ \mathbf{0}^T & 1 \end{bmatrix},$$

and the corresponding estimate of the image points is given by

$$\hat{\mathbf{X}}_i = (\mathbf{V}_1 \mathbf{V}_1^T + \mathbf{V}_2 \mathbf{V}_2^T) \mathbf{X}_i$$

Algorithm 4.7. *The Gold Standard Algorithm for estimating an affine homography H_A from image correspondences.*

of \mathbf{X} . This means that the partial derivative $\partial \mathcal{C}_H(\mathbf{X}) / \partial \mathbf{X}$ may be very simply computed. For instance, the derivative

$$\partial \mathcal{C}_H(x, y, x', y') / \partial x = \mathcal{C}_H(x + 1, y, x', y') - \mathcal{C}_H(x, y, x', y')$$

is *exact*, not a finite difference approximation. This means that for programming purposes, one does not need to code a special routine for taking derivatives – the routine for computing $\mathcal{C}_H(\mathbf{X})$ will suffice. Denoting by \mathbf{E}_i the vector containing 1 in the i -th position, and otherwise 0, one sees that $\partial \mathcal{C}_H(\mathbf{X}) / \partial X_i = \mathcal{C}_H(\mathbf{X} + \mathbf{E}_i) - \mathcal{C}_H(\mathbf{X})$, and further

$$\mathbf{J}\mathbf{J}^T = \sum_i (\mathcal{C}_H(\mathbf{X} + \mathbf{E}_i) - \mathcal{C}_H(\mathbf{X})) (\mathcal{C}_H(\mathbf{X} + \mathbf{E}_i) - \mathcal{C}_H(\mathbf{X}))^T.$$

Also note that computationally it is more efficient to solve $\mathbf{J}\mathbf{J}^T \boldsymbol{\lambda} = -\boldsymbol{\epsilon}$ directly for $\boldsymbol{\lambda}$, rather than take the inverse as $\boldsymbol{\lambda} = -(\mathbf{J}\mathbf{J}^T)^{-1} \boldsymbol{\epsilon}$.

- (viii) **Minimizing geometric error for affine transformations.** Given a set of correspondences $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$, find an affine transformation H_A that minimizes geometric error (4.8–p95). We will step through the derivation of a linear algorithm based on Sampson's approximation which is exact in this case. The complete method is summarized in algorithm 4.7.

- (a) Show that the optimum affine transformation takes the centroid of the \mathbf{x}_i to the centroid of \mathbf{x}'_i , so by translating the points to have their centroid at the origin, the translation part of the transformation is determined. It is only necessary then to determine the upper-left 2×2 submatrix $H_{2 \times 2}$ of H_A , which represents the linear part of the transformation.
 - (b) The point $\mathbf{X}_i = (\mathbf{x}_i^T, \mathbf{x}'_i{}^T)^T$ lies on \mathcal{V}_H if and only if $[H_{2 \times 2} | -I_{2 \times 2}]\mathbf{X} = \mathbf{0}$. So \mathcal{V}_H is a codimension-2 subspace of \mathbb{R}^4 .
 - (c) Any codimension-2 subspace may be expressed as $[H_{2 \times 2} | -I]\mathbf{X} = \mathbf{0}$ for suitable $H_{2 \times 2}$. Thus given measurements \mathbf{X}_i , the estimation task is equivalent to finding the best-fitting codimension-2 subspace.
 - (d) Given a matrix M with rows \mathbf{X}_i^T , the best-fitting subspace to the \mathbf{X}_i is spanned by the singular vectors \mathbf{V}_1 and \mathbf{V}_2 corresponding to the two largest singular values of M .
 - (e) The $H_{2 \times 2}$ corresponding to the subspace spanned by \mathbf{V}_1 and \mathbf{V}_2 is found by solving the equations $[H_{2 \times 2} | -I][\mathbf{V}_1 \mathbf{V}_2] = \mathbf{0}$.
- (ix) **Computing homographies of \mathbb{P}^3 from line correspondences.** Consider computing a 4×4 homography H from lines correspondences alone, assuming the lines are in general position in \mathbb{P}^3 . There are two questions: how many correspondences are required?, and how to formulate the algebraic constraints to obtain a solution for H ? It might be thought that four line correspondences would be sufficient because each line in \mathbb{P}^3 has four degrees of freedom, and thus four lines should provide $4 \times 4 = 16$ constraints on the 15 degrees of freedom of H . However, a configuration of four lines is degenerate (see section 4.1.3(p91)) for computing the transformation, as there is a 2D isotropy subgroup. This is discussed further in [Hartley-94c]. Equations linear in H can be obtained in the following way:

$$\pi_i^T H \mathbf{X}_j = 0, \quad i = 1, 2, \quad j = 1, 2,$$

where H transfers a line defined by the two points $(\mathbf{X}_1, \mathbf{X}_2)$ to a line defined by the intersection of the two planes (π_1, π_2) . This method was derived in [Oskarsson-02], where more details are to be found.

Algorithm Evaluation and Error Analysis

This chapter describes methods for assessing and quantifying the results of estimation algorithms. Often it is not sufficient to simply have an estimate of a variable or transformation. Instead some measure of confidence or uncertainty is also required.

Two methods for computing this uncertainty (covariance) are outlined here. The first is based on linear approximations and involves concatenating various Jacobian expressions. The second is the easier to implement Monte Carlo method.

5.1 Bounds on performance

Once an algorithm has been developed for the estimation of a certain type of transformation it is time to test its performance. This may be done by testing it on real or on synthetic data. In this section, testing on synthetic data will be considered, and a methodology for testing will be sketched.

We recall the notational convention:

- A quantity such as \mathbf{x} represents a measured image point.
- Estimated quantities are represented by a hat, such as $\hat{\mathbf{x}}$ or $\hat{\mathbf{H}}$.
- True values are represented by a bar, such as $\bar{\mathbf{x}}$ or $\bar{\mathbf{H}}$.

Typically, testing will begin with the synthetic generation of a set of image correspondences $\bar{\mathbf{x}}_i \leftrightarrow \bar{\mathbf{x}}'_i$ between two images. The number of such correspondences will vary. Corresponding points will be chosen in such a way that they correspond via a given fixed projective transformation $\bar{\mathbf{H}}$, and the correspondence is exact, in the sense that $\bar{\mathbf{x}}'_i = \bar{\mathbf{H}}\bar{\mathbf{x}}_i$ precisely, up to machine accuracy.

Next, artificial Gaussian noise will be added to the image measurements by perturbing both the x - and y -coordinates of the point by a zero-mean Gaussian random variable with known variance. The resulting noisy points are denoted \mathbf{x}_i and \mathbf{x}'_i . A suitable Gaussian random number generator is given in [Press-88]. The estimation algorithm is then run to compute the estimated quantity. For the 2D projective transformation problem considered in chapter 4, this means the projective transformation itself, and also perhaps estimates of the correct original noise-free image points. The algorithm is then evaluated according to how closely the computed model matches the (noisy) input data, or alternatively, how closely the estimated model agrees with the original

noise-free data. This procedure should be carried out many times with different noise (i.e. a different seed for the random number generator, though each time with the same noise variance) in order to obtain a statistically meaningful performance evaluation.

5.1.1 Error in one image

To illustrate this, we continue our investigation of the problem of 2D homography estimation. For simplicity we consider the case where noise is added to the coordinates of the second image only. Thus, $\mathbf{x}_i = \bar{\mathbf{x}}_i$ for all i . Let $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ be a set of noisy matched points between two images, generated from a perfectly matched set of data by injection of Gaussian noise with variance σ^2 in each of the two coordinates of the second (primed) image. Let there be n such matched points. From this data, a projective transformation $\hat{\mathbf{H}}$ is estimated using any one of the algorithms described in chapter 4. Obviously, the estimated transformation $\hat{\mathbf{H}}$ will not generally map \mathbf{x}_i to \mathbf{x}'_i , nor $\bar{\mathbf{x}}_i$ to $\bar{\mathbf{x}}'_i$ precisely, because of the injected noise in the coordinates of \mathbf{x}'_i . The RMS (root-mean-squared) residual error

$$\epsilon_{\text{res}} = \left(\frac{1}{2n} \sum_{i=1}^n d(\mathbf{x}'_i, \hat{\mathbf{H}}\mathbf{x}_i)^2 \right)^{1/2} \quad (5.1)$$

measures the average difference between the noisy input data (\mathbf{x}'_i) and the estimated points $\hat{\mathbf{x}}'_i = \hat{\mathbf{H}}\bar{\mathbf{x}}_i$. It is therefore appropriately called *residual error*. It measures how well the computed transformation matches the input data, and as such is a suitable quality measure for the estimation procedure.

The value of the residual error is *not* in itself an absolute measure of the quality of the solution obtained. For instance, consider the 2D projectivity problem in the case where the input data consists of just 4 matched points. Since a projective transformation is defined uniquely and exactly by 4 point correspondences, any reasonable algorithm will compute an $\hat{\mathbf{H}}$ that matches the points exactly, in the sense that $\mathbf{x}'_i = \hat{\mathbf{H}}\mathbf{x}_i$. This means that the residual error is zero. One cannot expect any better performance from an algorithm than this.

Note that $\hat{\mathbf{H}}$ matches the projected points to the input data \mathbf{x}'_i , and not to the original noise-free data, $\bar{\mathbf{x}}'_i$. In fact, since the difference between the noise-free and the noisy coordinates has variance σ^2 , in the minimal four-point case the residual difference between projected points $\hat{\mathbf{H}}\mathbf{x}_i$ and the noise-free data $\bar{\mathbf{x}}'_i$ also has variance σ^2 . Thus, in the case of 4 points, the model fits the noisy input data perfectly (i.e. the residual is zero), but does not give a very close approximation to the true noise-free values.

With more than 4 point matches, the value of the residual error will increase. In fact, intuitively, one expects that as the number of measurements (matched points) increases, the estimated model should agree more and more closely with the noise-free true values. Asymptotically, the variance should decrease in inverse proportion to the number of point matches. At the same time, the residual error will increase.

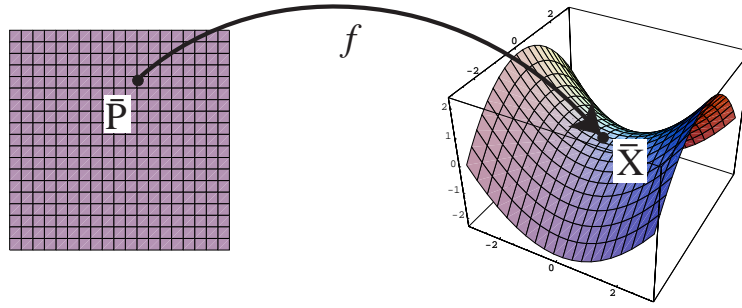


Fig. 5.1. As the values of the parameters \mathbf{P} vary, the function image traces out a surface S_M through the true value $\bar{\mathbf{X}}$.

5.1.2 Error in both images

In the case of error in both images, the residual error is

$$\epsilon_{\text{res}} = \frac{1}{\sqrt{4n}} \left(\sum_{i=1}^n d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + \sum_{i=1}^n d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \right)^{1/2} \quad (5.2)$$

where $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ are estimated points such that $\hat{\mathbf{x}}'_i = \hat{\mathbf{H}}\hat{\mathbf{x}}_i$.

5.1.3 Optimal estimators (MLE)

Bounds for estimation performance will be considered in a general framework, and then specialized to the two cases of error in one or both images. The goal is to derive formulae for the expected residual error of the Maximum Likelihood Estimate (MLE). As described previously, minimization of geometric error is equivalent to MLE, and so the goal of any algorithm implementing minimization of geometric error should be to achieve the theoretical bound given for the MLE. Another algorithm minimizing a different cost function (such as algebraic error) can be judged according to how close it gets to the bound given by the MLE.

A general estimation problem is concerned with a function f from \mathbb{R}^M to \mathbb{R}^N as described in section 4.2.7(p101), where \mathbb{R}^M is a parameter space, and \mathbb{R}^N is a space of measurements. Consider now a point $\bar{\mathbf{X}} \in \mathbb{R}^N$ for which there exists a vector of parameters $\bar{\mathbf{P}} \in \mathbb{R}^M$ such that $f(\bar{\mathbf{P}}) = \bar{\mathbf{X}}$ (i.e. a point $\bar{\mathbf{X}}$ in the range of f with preimage $\bar{\mathbf{P}}$). In the context of 2D projectivities with measurements in the second image only, this corresponds to a noise-free set of points $\bar{\mathbf{x}}'_i = \bar{\mathbf{H}}\bar{\mathbf{x}}_i$. The x - and y -components of the n points $\bar{\mathbf{x}}'_i, i = 1, \dots, n$ constitute the N -vector $\bar{\mathbf{X}}$ with $N = 2n$, and the parameters of the homography constitute the vector $\bar{\mathbf{P}}$ which may be an 8- or 9-vector depending on the parametrization of $\bar{\mathbf{H}}$.

Let \mathbf{X} be a measurement vector chosen according to an isotropic Gaussian distribution with mean the true measurement $\bar{\mathbf{X}}$ and variance $N\sigma^2$ (this notation means that each of the N components has variance σ^2). As the value of the parameter vector \mathbf{P} varies in a neighbourhood of the point $\bar{\mathbf{P}}$, the value of the function $f(\mathbf{P})$ traces out a surface S_M in \mathbb{R}^N through the point $\bar{\mathbf{X}}$. This is illustrated in figure 5.1. The surface S_M

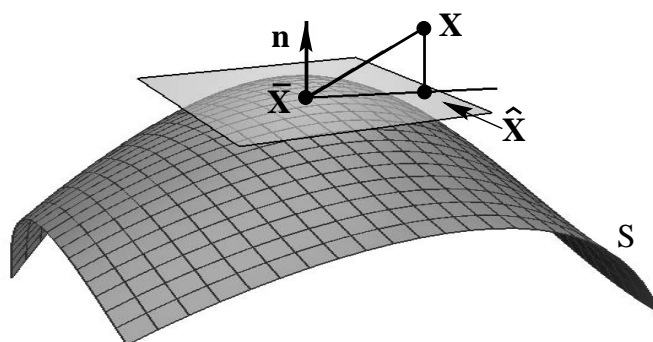


Fig. 5.2. Geometry of the errors in measurement space using the tangent plane approximation to S_M . The estimated point $\hat{\mathbf{X}}$ is the closest point on S_M to the measured point \mathbf{X} . The residual error is the distance between the measured point \mathbf{X} and $\hat{\mathbf{X}}$. The estimation error is the distance from $\hat{\mathbf{X}}$ to the true point $\bar{\mathbf{X}}$.

is given by the range of f . The dimension of the surface as a submanifold of \mathbb{R}^N is equal to d , where d is the number of essential parameters (that is the number of degrees of freedom, or minimum number of parameters). In the single-image error case, this equals 8, since the mapping determined by the matrix H is independent of scale.

Now, given a measurement vector \mathbf{X} , the maximum likelihood (ML) estimate $\hat{\mathbf{X}}$ is the point on S_M closest to \mathbf{X} . The ML estimator is the one that returns this closest point to \mathbf{X} that lies on this surface. Denote this ML estimate by $\hat{\mathbf{X}}$.

We now assume that in the neighbourhood of $\bar{\mathbf{X}}$, the surface is essentially planar and is well approximated by the tangent surface – at least for neighbourhoods around $\bar{\mathbf{X}}$ of the order of magnitude of noise variance. In this linear approximation, the ML estimate $\hat{\mathbf{X}}$ is the foot of the perpendicular from \mathbf{X} onto the tangent plane. The residual error is the distance from the point \mathbf{X} to the estimated value $\hat{\mathbf{X}}$. Furthermore, the distance from $\hat{\mathbf{X}}$ to (the unknown) $\bar{\mathbf{X}}$ is the distance from the optimally estimated value to the true value as seen in figure 5.2. Our task is to compute the expected value of these errors.

Computing the expected ML residual error has now been abstracted to a geometric problem as follows. The *total variance* of an N -dimensional Gaussian distribution is the trace of the covariance matrix, that is the sum of variances in each of the axial directions. This is, of course, unchanged by a change of orthogonal coordinate frame. For an N -dimensional isotropic Gaussian distribution with independent variances σ^2 in each variable, the total variance is $N\sigma^2$. Now, given an isotropic Gaussian random variable defined on \mathbb{R}^N with total variance $N\sigma^2$ and mean the true point $\bar{\mathbf{X}}$, we wish to compute the expected distance of the random variable from a dimension d hyperplane passing through $\bar{\mathbf{X}}$. The projection of a Gaussian random variable in \mathbb{R}^N onto the d -dimensional tangent plane gives the distribution of the *estimation error* (the difference between the estimated value and the true result). Projection onto the

$(N - d)$ -dimensional normal to the tangent surface gives the distribution of the residual error.

By a rotation of axes if necessary, one may assume, without loss of generality, that the tangent surface coincides with the first d coordinate axes. Integration over the remaining axial directions provides the following result.

Result 5.1. *The projection of an isotropic Gaussian distribution defined on \mathbb{R}^N with total variance $N\sigma^2$ onto a subspace of dimension s is an isotropic Gaussian distribution with total variance $s\sigma^2$.*

The proof of this is straightforward, and is omitted. We apply this in the two cases where $s = d$ and $s = N - d$ to obtain the following results.

Result 5.2. *Consider an estimation problem where N measurements are to be modelled by a function depending on a set of d essential parameters. Suppose the measurements are subject to independent Gaussian noise with standard deviation σ in each measurement variable.*

- (i) *The RMS residual error (distance of the measured from the estimated value) for the ML estimator is*

$$\epsilon_{\text{res}} = E[\|\hat{\mathbf{X}} - \mathbf{X}\|^2/N]^{1/2} = \sigma(1 - d/N)^{1/2} \quad (5.3)$$

- (ii) *The RMS estimation error (distance of the estimated from the true value) for the ML estimator is*

$$\epsilon_{\text{est}} = E[\|\hat{\mathbf{X}} - \bar{\mathbf{X}}\|^2/N]^{1/2} = \sigma(d/N)^{1/2} \quad (5.4)$$

where \mathbf{X} , $\hat{\mathbf{X}}$ and $\bar{\mathbf{X}}$ are respectively the measured, estimated and true values of the measurement vector.

Result 5.2 follows directly from result 5.1 by dividing by N to get the variance per measurement, then taking a square root to get standard deviation, instead of variance.

These values give lower bounds for residual error against which a particular estimation algorithm may be measured.

2D homography – error in one image. For the 2D projectivity estimation problem considered in this chapter, assuming error in the second image only, we have $d = 8$ and $N = 2n$, where n is the number of point matches. Thus, we have for this problem

$$\begin{aligned} \epsilon_{\text{res}} &= \sigma(1 - 4/n)^{1/2} \\ \epsilon_{\text{est}} &= \sigma(4/n)^{1/2}. \end{aligned} \quad (5.5)$$

Graphs of these errors as n varies are shown in figure 5.3.

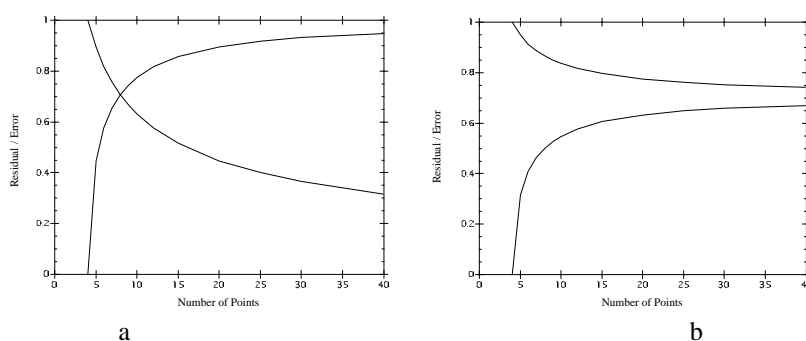


Fig. 5.3. Optimal error when noise is present in (a) one image, and in (b) both images as the number of points varies. An error level of one pixel is assumed. The descending curve shows the estimation error ϵ_{est} and the ascending curve shows the residual error ϵ_{res} .

Error in both images. In this case, $N = 4n$ and $d = 2n + 8$. As before, assuming linearity of the tangent surface in the neighbourhood of the true measurement vector $\hat{\mathbf{X}}$, result 5.2 gives the following expected errors.

$$\begin{aligned}\epsilon_{\text{res}} &= \sigma \left(\frac{n-4}{2n} \right)^{1/2} \\ \epsilon_{\text{est}} &= \sigma \left(\frac{n+4}{2n} \right)^{1/2}.\end{aligned}\quad (5.6)$$

Graphs of these errors as n varies are also shown in Figure 5.3.

An interesting observation to be made from this graph is that the asymptotic error with respect to the true values is $\sigma/\sqrt{2}$, and not 0 as in the case of error in one image. This result is expected, since in effect, one has two measurements of the position of each point, one in each image, related by the projective transformation. With two measurements of a point the variance in the estimate of the point position decreases by a factor of $\sqrt{2}$. By contrast, in the previous case where errors occur in one image only, one has one exact measurement for each point (i.e. in the first image). Thus, as the transformation H is estimated with greater and greater accuracy, the exact position of the point in the second image becomes known with uncertainty asymptotically approaching 0.

Mahalanobis distance. The formulae quoted above were derived under the assumption that the error distribution in measurement space was an isotropic Gaussian distribution, meaning that errors in each coordinate were independent. This assumption is not essential. We may assume any Gaussian distribution of error, with covariance matrix Σ . The formulae of result 5.2 remain true with ϵ being replaced with the expected Mahalanobis distance $E[\|\hat{\mathbf{X}} - \mathbf{X}\|_{\Sigma}^2/N]^{1/2}$. The standard deviation σ also disappears, since it is taken care of by the Mahalanobis distance.

This follows from a simple change of coordinates in the measurement space \mathbb{R}^N to make the covariance matrix equal to the identity. In this new coordinate frame, Mahalanobis distance becomes the same as Euclidean distance.

5.1.4 Determining the correct convergence of an algorithm

The relations given in (5.3) and (5.4) give a simple way of determining correct convergence of an estimation algorithm, without the need to determine the number of degrees of freedom of the problem. As seen in figure 5.2, the measurement space corresponding to the model specified by the parameter vector \mathbf{P} forms a surface S_M . If near the noise-free data $\bar{\mathbf{X}}$ the surface is nearly planar, then it may be approximated by its tangent plane, and the three points $\hat{\mathbf{X}}$, \mathbf{X} and $\bar{\mathbf{X}}$ form a right-angled triangle. In most estimation problems this assumption of planarity will be very close to correct at the scale set by typical noise magnitude. In this case, the Pythagorean equality may be written as

$$\|\mathbf{X} - \bar{\mathbf{X}}\|^2 = \|\mathbf{X} - \hat{\mathbf{X}}\|^2 + \|\bar{\mathbf{X}} - \hat{\mathbf{X}}\|^2 \quad (5.7)$$

In evaluating an algorithm with synthetic data, this equality allows a simple test to see whether the algorithm has converged to the optimal value. If the estimated value $\hat{\mathbf{X}}$ satisfies this equality, then it is a strong indication that the algorithm has found the true global minimum. Note that it is unnecessary in applying this test to determine the number of degrees of freedom of the problem. A few more properties are listed:

- This test can be used to determine on a run-by-run basis whether the algorithm has succeeded. Thus, with repeated runs, it allows an estimate of the percentage success rate for the algorithm.
- This test can only be used for synthetic data, or at least data for which the true measurements $\bar{\mathbf{X}}$ are known.
- The equality (5.7) depends on the assumption that the surface S_M consisting of valid measurements is locally planar. If the equality is not satisfied for a particular run of the estimation algorithm, then this is because the surface is not planar, or (far more likely) because the algorithm is failing to find the best solution.
- The test (5.7) is a test for the algorithm finding the global, not a local solution. If $\hat{\mathbf{X}}$ settles to a local cost minimum, then the right-hand-side of (5.7) is likely to be much larger than the left-hand-side. The condition is unlikely to be satisfied entirely by chance if the algorithm finds the incorrect point $\hat{\mathbf{X}}$.

5.2 Covariance of the estimated transformation

In the previous section the ML estimate was considered, and how its expected average error may be computed. Comparing the achieved residual error or estimation error of an algorithm against the ML error is a good way of evaluating the performance of a particular estimation algorithm, since it compares the results of the algorithm against the best that may be achieved (the optimum estimate) in the absence of any other prior information.

Nevertheless, the chief concern is how accurately the transformation itself has been computed. The uncertainty of the estimated transformation depends on many factors, including the number of points used to compute it, the accuracy of the given point matches, as well as the configuration of the points in question. To illustrate the importance of the configuration suppose the points used to compute the transformation are

close to a degenerate configuration; then the transformation may not be computed with great accuracy. For instance, if the transformation is computed from a set of points that lie close to a straight line, then the behaviour of the transformation in the dimension perpendicular to that line is not accurately determined. Thus, whereas the achievable residual error and estimation error were seen to be dependent only on the number of point correspondences and their accuracy, by contrast, the accuracy of the computed transformation is dependent on the particular points. The uncertainty of the computed transformation is conveniently captured in the *covariance matrix* of the transformation. Since \mathbf{H} is a matrix with 9 entries, its covariance matrix will be a 9×9 matrix. In this section it will be seen how this covariance matrix may be computed.

5.2.1 Forward propagation of covariance

The covariance matrix behaves in a pleasantly simple manner under affine transformations, as described in the following theorem.

Result 5.3. Let \mathbf{v} be a random vector in \mathbb{R}^M with mean $\bar{\mathbf{v}}$ and covariance matrix Σ , and suppose that $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ is an affine mapping defined by $f(\mathbf{v}) = f(\bar{\mathbf{v}}) + \mathbf{A}(\mathbf{v} - \bar{\mathbf{v}})$. Then $f(\mathbf{v})$ is a random variable with mean $f(\bar{\mathbf{v}})$ and covariance matrix $\mathbf{A}\Sigma\mathbf{A}^\top$.

Note that it is not assumed that \mathbf{A} is a square matrix. Instead of giving a proof of this theorem, we give an example.

Example 5.4. Let x and y be independent random variables with mean 0 and standard deviations of 1 and 2 respectively. What are the mean and standard deviation of $x' = f(x, y) = 3x + 2y - 7$?

The mean is $\bar{x}' = f(0, 0) = -7$. Next we compute the variance of x' . In this case, Σ is the matrix $\begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$ and \mathbf{A} is the matrix $[3 \ 2]$. Thus, the variance of x' is $\mathbf{A}\Sigma\mathbf{A}^\top = 25$. Thus $3x + 2y - 7$ has standard deviation 5. \triangle

Example 5.5. Let $x' = 3x + 2y$ and $y' = 3x - 2y$. Find the covariance matrix of (x', y') , given that x and y have the same distribution as before.

In this case, the matrix $\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 3 & -2 \end{bmatrix}$. One computes $\mathbf{A}\Sigma\mathbf{A}^\top = \begin{bmatrix} 25 & -7 \\ -7 & 25 \end{bmatrix}$. Thus, one sees that both x' and y' have variance 25 (standard deviation 5), whereas x' and y' are negatively correlated, with covariance $E[x'y'] = -7$. \triangle

Non-linear propagation. If \mathbf{v} is a random vector in \mathbb{R}^M and $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ is a non-linear function acting on \mathbf{v} , then we may compute an approximation to the mean and covariance of $f(\mathbf{v})$ by assuming that f is approximately affine in the vicinity of the mean of the distribution. The affine approximation to f is $f(\mathbf{v}) \approx f(\bar{\mathbf{v}}) + \mathbf{J}(\mathbf{v} - \bar{\mathbf{v}})$, where \mathbf{J} is the partial derivative (Jacobian) matrix $\partial f / \partial \mathbf{v}$ evaluated at $\bar{\mathbf{v}}$. Note that \mathbf{J} has dimension $N \times M$. Then we have the following result.

Result 5.6. Let \mathbf{v} be a random vector in \mathbb{R}^M with mean $\bar{\mathbf{v}}$ and covariance matrix Σ ,

and let $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ be differentiable in a neighbourhood of $\bar{\mathbf{v}}$. Then up to a first-order approximation, $f(\mathbf{v})$ is a random variable with mean $f(\bar{\mathbf{v}})$ and covariance $\mathbf{J}\Sigma\mathbf{J}^\top$, where \mathbf{J} is the Jacobian matrix of f , evaluated at $\bar{\mathbf{v}}$.

The extent to which this result gives a good approximation to the actual mean and variance of $f(\bar{\mathbf{v}})$ depends on how closely the function f is approximated by a linear function in a region about $\bar{\mathbf{v}}$ commensurate in size with the support of the probability distribution of \mathbf{v} .

Example 5.7. Let $\mathbf{x} = (x, y)^\top$ be a Gaussian random vector with mean $(0, 0)^\top$ and covariance matrix $\sigma^2 \text{diag}(1, 4)$. Let $x' = f(x, y) = x^2 + 3x - 2y + 5$. Then one may compute the true values of the mean and standard deviation of $f(x, y)$ according to the formulae

$$\begin{aligned}\bar{x}' &= \int \int_{-\infty}^{\infty} P(x, y) f(x, y) dx dy \\ \sigma_{x'}^2 &= \int \int_{-\infty}^{\infty} P(x, y) (f(x, y) - \bar{x}')^2 dx dy\end{aligned}$$

where

$$P(x, y) = \frac{1}{4\pi\sigma^2} e^{-(x^2+y^2/4)/2\sigma^2}$$

is the Gaussian probability distribution (A2.1–p565). One obtains

$$\begin{aligned}\bar{x}' &= 5 + \sigma^2 \\ \sigma_{x'}^2 &= 25\sigma^2 + 2\sigma^4.\end{aligned}$$

Applying the approximation given by result 5.6, and noting that $\mathbf{J} = [3 \ -2]$, we find that the estimated values are

$$\begin{aligned}\bar{x}' &= 5 \\ \sigma_{x'}^2 &= \sigma^2 [3 \ -2] \begin{bmatrix} 1 & \\ & 4 \end{bmatrix} [3 \ -2]^\top = 25\sigma^2.\end{aligned}$$

Thus, as long as σ is small, this is a close approximation to the correct values of the mean and variance of x' . The following table shows the true and approximated values for the mean and standard deviation of $f(x, y)$ for two different values of σ .

| | $\sigma = 0.25$ | | $\sigma = 0.5$ | |
|----------|-----------------|---------------|----------------|---------------|
| | \bar{x}' | $\sigma_{x'}$ | \bar{x}' | $\sigma_{x'}$ |
| estimate | 5.0000 | 1.25000 | 5.00 | 2.5000 |
| true | 5.0625 | 1.25312 | 5.25 | 2.5249 |

For reference, in the case $\sigma = 0.25$, one sees that as long as $|x| < 2\sigma$ (about 95% of the total distribution) the value $f(x, y) = x^2 + 3x - 2y + 5$ differs from its linear approximation $3x - 2y + 5$ by no more than $x^2 < 0.25$. \triangle

Example 5.8. More generally, assuming that x and y are independent zero-mean Gaussian random variables, one may compute that for the function $f(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$,

$$\begin{aligned}\text{mean} &= a\sigma_x^2 + c\sigma_y^2 + f \\ \text{variance} &= 2a^2\sigma_x^4 + b^2\sigma_x^2\sigma_y^2 + 2c^2\sigma_y^4 + d^2\sigma_x^2 + e^2\sigma_y^2\end{aligned}$$

which are close to the estimated values $\text{mean} = f$, $\text{variance} = d^2\sigma_x^2 + e^2\sigma_y^2$ as long as σ_x and σ_y are small. \triangle

5.2.2 Backward propagation of covariance

The material in this and the following section 5.2.3 is more advanced. The examples in section 5.2.4 show the straightforward application of the results of these sections, and can be read first.

Consider a differentiable mapping f from a “parameter space”, \mathbb{R}^M to a “measurement space” \mathbb{R}^N , and let a Gaussian probability distribution be defined on \mathbb{R}^N with covariance matrix Σ . Let S_M be the image of the mapping f . We assume that $M < N$ and that S_M has the same dimension M as the parameter space \mathbb{R}^M . Thus we are not considering the over-parametrized case at present. A vector \mathbf{P} in \mathbb{R}^M represents a parametrization of the point $f(\mathbf{P})$ on S_M . Finding the point on S_M closest in Mahalanobis distance to a given point \mathbf{X} in \mathbb{R}^N defines a map from \mathbb{R}^N to the surface S_M . We call this mapping $\eta : \mathbb{R}^N \rightarrow S_M$. Now, f is by assumption invertible on the surface S_M , and we define $f^{-1} : S_M \rightarrow \mathbb{R}^M$ to be the inverse function.

By composing the map $\eta : \mathbb{R}^N \rightarrow S_M$ and $f^{-1} : S_M \rightarrow \mathbb{R}^M$ we have a mapping $f^{-1} \circ \eta : \mathbb{R}^N \rightarrow \mathbb{R}^M$. This mapping assigns to a measurement vector \mathbf{X} , the set of parameters \mathbf{P} corresponding to the ML estimate $\hat{\mathbf{X}}$. In principle we may propagate the covariance of the probability distribution in the measurement space \mathbb{R}^N to compute a covariance matrix for the set of parameters \mathbf{P} corresponding to ML estimation. Our goal is to apply result 5.3 or result 5.6.

We consider first the case where the mapping f is an affine mapping from \mathbb{R}^M into \mathbb{R}^N . We will show next that the mapping $f^{-1} \circ \eta$ is also an affine mapping, and a specific form will be given for $f^{-1} \circ \eta$, thereby allowing us to apply result 5.3 to compute the covariance of the estimated parameters $\hat{\mathbf{P}} = f^{-1} \circ \eta(\mathbf{X})$.

Since f is affine, we may write $f(\mathbf{P}) = f(\bar{\mathbf{P}}) + \mathbf{J}(\mathbf{P} - \bar{\mathbf{P}})$, where $f(\bar{\mathbf{P}}) = \bar{\mathbf{X}}$ is the mean of the probability distribution on \mathbb{R}^N . Since we are assuming that the surface $S_M = f(\mathbb{R}^M)$ has dimension M , the rank of \mathbf{J} is equal to its column dimension. Given a measurement vector \mathbf{X} , the ML estimate $\hat{\mathbf{X}}$ minimizes $\|\mathbf{X} - \hat{\mathbf{X}}\|_{\Sigma} = \|\mathbf{X} - f(\hat{\mathbf{P}})\|_{\Sigma}$. Thus, we seek $\hat{\mathbf{P}}$ to minimize this latter quantity. However,

$$\|\mathbf{X} - f(\hat{\mathbf{P}})\|_{\Sigma} = \|(\mathbf{X} - \bar{\mathbf{X}}) - \mathbf{J}(\hat{\mathbf{P}} - \bar{\mathbf{P}})\|_{\Sigma}$$

and this is minimized (see (A5.2–p591) in section A5.2.1(p591)) when

$$(\hat{\mathbf{P}} - \bar{\mathbf{P}}) = (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) .$$

Writing $\bar{\mathbf{P}} = f^{-1}\bar{\mathbf{X}}$ and $\hat{\mathbf{P}} = f^{-1}\hat{\mathbf{X}}$, we see that

$$\begin{aligned} f^{-1} \circ \eta(\mathbf{X}) &= \hat{\mathbf{P}} \\ &= (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) + f^{-1}(\bar{\mathbf{X}}) \\ &= (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) + f^{-1} \circ \eta(\bar{\mathbf{X}}) . \end{aligned}$$

This shows that $f^{-1} \circ \eta$ is affine and $(\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1}$ is its linear part. Applying result 5.3, we see that the covariance matrix for $\hat{\mathbf{P}}$ is

$$\begin{aligned} [(\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1}] \Sigma [(\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1}]^T &= (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \mathbf{J}^T \Sigma^{-1} \Sigma \Sigma^{-1} \mathbf{J} (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \\ &= (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1}, \end{aligned}$$

recalling that Σ is symmetric. We have proved the following theorem.

Result 5.9 Backward transport of covariance – affine case. *Let $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ be an affine mapping of the form $f(\mathbf{P}) = f(\bar{\mathbf{P}}) + \mathbf{J}(\mathbf{P} - \bar{\mathbf{P}})$, where \mathbf{J} has rank M . Let \mathbf{X} be a random variable in \mathbb{R}^N with mean $\bar{\mathbf{X}} = f(\bar{\mathbf{P}})$ and covariance matrix Σ . Let $f^{-1} \circ \eta : \mathbb{R}^N \rightarrow \mathbb{R}^M$ be the mapping that maps a measurement \mathbf{X} to the set of parameters corresponding to the ML estimate $\hat{\mathbf{X}}$. Then $\hat{\mathbf{P}} = f^{-1} \circ \eta(\mathbf{X})$ is a random variable with mean $\bar{\mathbf{P}}$ and covariance matrix*

$$\Sigma_{\mathbf{P}} = (\mathbf{J}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{J})^{-1}. \quad (5.8)$$

In the case where f is not affine, an approximation to the mean and covariance may be obtained by approximating f by an affine function in the usual way, as follows.

Result 5.10 Backward transport of covariance – non-linear case. *Let $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ be a differentiable mapping and let \mathbf{J} be its Jacobian matrix evaluated at a point $\bar{\mathbf{P}}$. Suppose that \mathbf{J} has rank M . Then f is one-to-one in a neighbourhood of $\bar{\mathbf{P}}$. Let \mathbf{X} be a random variable in \mathbb{R}^N with mean $\bar{\mathbf{X}} = f(\bar{\mathbf{P}})$ and covariance matrix $\Sigma_{\mathbf{X}}$. Let $f^{-1} \circ \eta : \mathbb{R}^N \rightarrow \mathbb{R}^M$ be the mapping that maps a measurement \mathbf{X} to the set of parameters corresponding to the ML estimate $\hat{\mathbf{X}}$. Then to first-order, $\hat{\mathbf{P}} = f^{-1} \circ \eta(\mathbf{X})$ is a random variable with mean $\bar{\mathbf{P}}$ and covariance matrix $(\mathbf{J}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{J})^{-1}$.*

5.2.3 Over-parametrization

One may generalize result 5.9 and result 5.10 to the case of redundant sets of parameters – the over-parametrized case. In this case, the mapping f from the parameter space \mathbb{R}^M to measurement space \mathbb{R}^N is not locally one-to-one. For instance, in the case of estimating a 2D homography as discussed in section 4.5(p110) there is a mapping $f(\mathbf{P})$ where \mathbf{P} is a 9-vector representing the entries of the homography matrix \mathbf{H} . Since the homography has only 8 degrees of freedom, the mapping f is not one-to-one. In particular, for any constant k , the matrix $k\mathbf{H}$ represents the same map, and so the image coordinate vectors $f(\mathbf{P})$ and $f(k\mathbf{P})$ are equal.

In the general case of a mapping $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ the Jacobian matrix \mathbf{J} does not have full rank M , but rather a smaller rank $d < M$. This rank d is called the number of *essential parameters*. The matrix $\mathbf{J}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{J}$ in this case has dimension M but rank

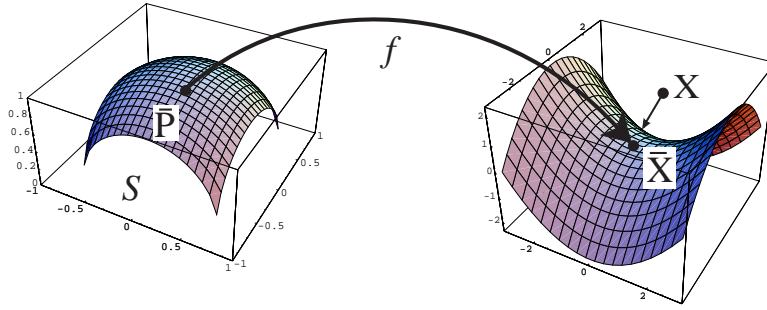


Fig. 5.4. **Back propagation (over-parametrized).** Mapping f maps constrained parameter surface to measurement space. A measurement \mathbf{X} is mapped (by a mapping η) to the closest point on the surface $f(S_P)$ and then back via f^{-1} to the parameter space, providing the ML estimate of the parameters. The covariance of \mathbf{X} is transferred via $f^{-1} \circ \eta$ to a covariance of the parameters.

$d < M$. The formula (5.8), $\Sigma_P = (J^T \Sigma_X^{-1} J)^{-1}$, clearly does not hold, since the matrix on the right side is not invertible.

In fact, it is clear that without any further restriction, the elements of the estimated vector \hat{P} may vary without bound, namely through multiplication by an arbitrary constant k . Hence the elements have infinite variance. It is usual to restrict the estimated homography matrix H or more generally the parameter vector P by imposing some constraint. The usual constraint is that $\|P\| = 1$ though other constraints are possible, such as demanding that the last parameter should equal 1 (see section 4.4.2(p105)). Thus, the parameter vector P is constrained to lie on a surface in the parameter space \mathbb{R}^9 , or generally \mathbb{R}^M . In the first case the surface $\|P\| = 1$ is the unit sphere in \mathbb{R}^M . The constraint $P_m = 1$ represents a plane in \mathbb{R}^M . In the general case we may assume that the estimated vector P lies on some submanifold of \mathbb{R}^M as in the following theorem.

Result 5.11. Backward transport of covariance – over-parametrized case. Let $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ be a differentiable mapping taking a parameter vector \bar{P} to a measurement vector \bar{X} . Let S_P be a smooth manifold of dimension d embedded in \mathbb{R}^M passing through point \bar{P} , and such that the map f is one-to-one on the manifold S_P in a neighbourhood of \bar{P} , mapping S_P locally to a manifold $f(S_P)$ in \mathbb{R}^N . The function f has a local inverse, denoted f^{-1} , restricted to the surface $f(S_P)$ in a neighbourhood of \bar{X} . Let a Gaussian distribution on \mathbb{R}^N be defined with mean \bar{X} and covariance matrix Σ_X and let $\eta : \mathbb{R}^N \rightarrow f(S_P)$ be the mapping that takes a point in \mathbb{R}^N to the closest point on $f(S_P)$ with respect to Mahalanobis norm $\|\cdot\|_{\Sigma_X}$. Via $f^{-1} \circ \eta$ the probability distribution on \mathbb{R}^N with covariance matrix Σ_X induces a probability distribution on \mathbb{R}^M with covariance matrix, to first-order equal to

$$\Sigma_P = (J^T \Sigma_X^{-1} J)^{+A} = A(A^T J^T \Sigma_X^{-1} J A)^{-1} A^T \quad (5.9)$$

where A is any $m \times d$ matrix whose column vectors span the tangent space to S_P at \bar{P} .

This is illustrated in figure 5.4. The notation $(J^T \Sigma_X^{-1} J)^{+A}$, defined by (5.9), is discussed further in section A5.2(p590).

Proof. The proof of result 5.11 is straightforward. Let d be the number of essential parameters. One defines a map $g : \mathbb{R}^d \rightarrow \mathbb{R}^M$ mapping an open neighbourhood U in \mathbb{R}^d to an open set of S_P containing the point \bar{P} . Then the combined mapping $f \circ g : \mathbb{R}^d \rightarrow \mathbb{R}^N$ is one-to-one on the neighbourhood U . Let us denote the partial derivative matrices of f by J and of g by A . The matrix of partial derivatives of $f \circ g$ is then JA . Result 5.10 now applies, and one sees that the probability distribution function with covariance matrix Σ on \mathbb{R}^N may be transported backwards to a covariance matrix $(A^T J^T \Sigma^{-1} J A)^{-1}$ on \mathbb{R}^d . Transporting this forwards again to \mathbb{R}^M , applying result 5.6, we arrive at the covariance matrix $A(A^T J^T \Sigma^{-1} J A)^{-1} A^T$ on S_P . This matrix, which will be denoted here by $(J^T \Sigma^{-1} J)^+ A$, is related to the pseudo-inverse of $(J^T \Sigma^{-1} J)$ as defined in section A5.2(p590). The expression (5.9) is not dependent on the particular choice of the matrix A as long as the column span of A is unchanged. In particular, if A is replaced by AB for any invertible $d \times d$ matrix B , then the value of (5.9) does not change. Thus, any matrix A whose columns span the tangent space of S_P at \bar{P} will do. \square

Note that the proof gives a specific way of computing a matrix A spanning the tangent space – namely the Jacobian matrix of g . In many instances, as we will see, there are easier ways of finding A . Note that the covariance matrix (5.9) is singular. In particular, it has dimension M and rank $d < M$. This is because the variance of the estimated parameter set in directions orthogonal to the constraint surface S_P is zero – there can be no variation in that direction. Note that whereas $J^T \Sigma^{-1} J$ is non-invertible, the $d \times d$ matrix $A^T J^T \Sigma^{-1} J A$ has rank d and is invertible.

An important case occurs when the constraint surface is locally orthogonal to the null-space of the Jacobian matrix. Denote by $N_L(X)$ the left null-space of matrix X , namely the space of all vectors x such that $x^T X = 0$. Then (as shown in section A5.2-(p590)), the *pseudo-inverse* X^+ is given by

$$X^+ = X^{+A} = A(A^T X A)^{-1} A^T$$

if and only if $N_L(A) = N_L(X)$. The following result then derives directly from result 5.11.

Result 5.12. Let $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ be a differentiable mapping taking \bar{P} to \bar{X} , and let J be the Jacobian matrix of f . Let a Gaussian distribution on \mathbb{R}^N be defined at \bar{X} with covariance matrix Σ_X and let $f^{-1} \circ \eta : \mathbb{R}^N \rightarrow \mathbb{R}^M$ as in result 5.11 be the mapping taking a measurement X to the MLE parameter vector P constrained to lie on a surface S_P locally orthogonal to the null-space of J . Then $f^{-1} \circ \eta$ induces a distribution on \mathbb{R}^M with covariance matrix, to first-order equal to

$$\Sigma_P = (J^T \Sigma_X^{-1} J)^+. \quad (5.10)$$

Note that the restriction that P be constrained to lie on a surface locally orthogonal to the null-space of J is in many cases the natural constraint. For instance, if P is a *homogeneous* parameter vector (such as the entries of a homogeneous matrix), the restriction is satisfied for the usual constraint $\|P\| = 1$. In such a case, the constraint surface is the unit sphere, and the tangent plane at any point is perpendicular to the parameter vector. On the other hand, since P is a homogeneous vector, the function

$f(\mathbf{P})$ is invariant to changes of scale, and so \mathbf{J} has a null-vector in the radial direction, thus perpendicular to the constraint surface.

In other cases, it is often not critical what restriction we place on the parameter set for the purpose of computing the covariance matrix of the parameters. In addition, since the pseudo-inversion operation is its own inverse, we can retrieve the original matrix from its pseudo-inverse, according to $\mathbf{J}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{J} = \Sigma_{\mathbf{P}}^+$. One can then compute the covariance matrix corresponding to any other subspace, according to

$$(\mathbf{J}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{J})^{+A} = (\Sigma_{\mathbf{P}}^+)^{+A}$$

where the columns of \mathbf{A} span the constrained subspace of parameter space.

5.2.4 Application and examples

Error in one image. Let us consider the application of this theory to the problem of finding the covariance of an estimated 2D homography \mathbf{H} . First, we look at the case where the error is limited to the second image. The 3×3 matrix \mathbf{H} is represented by a 9-dimensional parameter vector which will be denoted by \mathbf{h} instead of \mathbf{P} so as to remind us that it is made up of the entries of \mathbf{H} . The covariance of the estimated $\hat{\mathbf{h}}$ is a 9×9 symmetric matrix. We are given a set of matched points $\bar{\mathbf{x}}_i \leftrightarrow \mathbf{x}'_i$. The points $\bar{\mathbf{x}}_i$ are fixed true values, and the points \mathbf{x}'_i are considered as random variables subject to Gaussian noise with variance σ^2 in each component, or if desired, with a more general covariance. The function $f: \mathbb{R}^9 \rightarrow \mathbb{R}^{2n}$ is defined as mapping a 9-vector \mathbf{h} representing a matrix \mathbf{H} to the $2n$ -vector made up of the coordinates of the points $\mathbf{x}'_i = \mathbf{H}\bar{\mathbf{x}}_i$. The coordinates of \mathbf{x}'_i make up a composite vector in \mathbb{R}^N , which we denote by \mathbf{X}' . As we have seen, as \mathbf{h} varies, the point $f(\mathbf{h})$ traces out an 8-dimensional surface $S_{\mathbf{P}}$ in \mathbb{R}^{2n} . Each point \mathbf{X}' on the surface represents a set of points \mathbf{x}'_i consistent with the first-image points $\bar{\mathbf{x}}_i$. Given a vector of measurements \mathbf{X}' , one selects the closest point $\hat{\mathbf{X}}'$ on the surface $S_{\mathbf{P}}$ with respect to Mahalanobis distance. The pre-image $\hat{\mathbf{h}} = f^{-1}(\hat{\mathbf{X}}')$, subject to constraint $\|\mathbf{h}\| = 1$, represents the estimated homography matrix $\hat{\mathbf{H}}$, estimated using the ML estimator. From the probability distribution of values of \mathbf{X}' one wishes to derive the distribution of the estimated $\hat{\mathbf{h}}$. The covariance matrix $\Sigma_{\mathbf{h}}$ is given by result 5.12. This covariance matrix corresponds to the constraint $\|\mathbf{h}\| = 1$.

Thus, a procedure for computing the covariance matrix of the estimated transformation is as follows.

- (i) Estimate the transformation $\hat{\mathbf{H}}$ from the given data.
- (ii) Compute the Jacobian matrix $\mathbf{J}_f = \partial \mathbf{X}' / \partial \mathbf{h}$, evaluated at $\hat{\mathbf{h}}$.
- (iii) The covariance matrix of the estimated \mathbf{h} is given by (5.10): $\Sigma_{\mathbf{h}} = (\mathbf{J}_f^T \Sigma_{\mathbf{X}'}^{-1} \mathbf{J}_f)^{+}$.

We investigate the two last steps of this method in slightly more detail.

Computation of the derivative matrix. Consider first the Jacobian matrix $\mathbf{J} = \partial \mathbf{X}' / \partial \mathbf{h}$. This matrix has a natural decomposition into blocks so that $\mathbf{J} = (\mathbf{J}_1^T, \mathbf{J}_2^T, \dots, \mathbf{J}_i^T, \dots, \mathbf{J}_n^T)^T$ where $\mathbf{J}_i = \partial \mathbf{x}'_i / \partial \mathbf{h}$. A formula for $\partial \mathbf{x}'_i / \partial \mathbf{h}$ is given in

(4.21–p129):

$$\mathbf{J}_i = \partial \mathbf{x}'_i / \partial \mathbf{h} = \frac{1}{w'_i} \begin{bmatrix} \tilde{\mathbf{x}}_i^\top & \mathbf{0}^\top & -x'_i \tilde{\mathbf{x}}_i^\top \\ \mathbf{0}^\top & \tilde{\mathbf{x}}_i^\top & -y'_i \tilde{\mathbf{x}}_i^\top \end{bmatrix} \quad (5.11)$$

where $\tilde{\mathbf{x}}_i^\top$ represents the vector $(x_i, y_i, 1)$.

Stacking these matrices on top of each other for all points \mathbf{x}_i gives the derivative matrix $\partial \mathbf{X}' / \partial \mathbf{h}$. An important case is when the image measurements \mathbf{x}'_i are independent random vectors. In this case $\Sigma = \text{diag}(\Sigma_1, \dots, \Sigma_n)$ where each Σ_i is the 2×2 covariance matrix of the i -th measured point \mathbf{x}'_i . Then one computes

$$\Sigma_{\mathbf{h}} = (\mathbf{J}^\top \Sigma_{\mathbf{X}'}^{-1} \mathbf{J})^+ = \left(\sum_i \mathbf{J}_i^\top \Sigma_i^{-1} \mathbf{J}_i \right)^+. \quad (5.12)$$

Example 5.13. We consider the simple numerical example of a point correspondence containing just 4 points as follows:

$$\begin{aligned} \mathbf{x}_1 &= (1, 0)^\top \leftrightarrow (1, 0)^\top = \mathbf{x}'_1 \\ \mathbf{x}_2 &= (0, 1)^\top \leftrightarrow (0, 1)^\top = \mathbf{x}'_2 \\ \mathbf{x}_3 &= (-1, 0)^\top \leftrightarrow (-1, 0)^\top = \mathbf{x}'_3 \\ \mathbf{x}_4 &= (0, -1)^\top \leftrightarrow (0, -1)^\top = \mathbf{x}'_4 \end{aligned}$$

namely, the identity map on the points of a projective basis. We assume that points \mathbf{x}_i are known exactly, and points \mathbf{x}'_i have one pixel standard deviation in each coordinate direction. This means that the covariance matrix $\Sigma_{\mathbf{x}'_i}$ is the identity.

Obviously, the computed homography will be the identity map. For simplicity we normalize (scale it) so that it is indeed the identity matrix, and hence $\|\mathbf{H}\|^2 = 3$ instead of the usual normalization $\|\mathbf{H}\| = 1$. In this case, all the w'_i in (5.11) are equal to 1. The matrix \mathbf{J} is easily computed from (5.11) to equal

$$\mathbf{J} = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 \\ \hline -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 1 \end{array} \right].$$

Then

$$\mathbf{J}^\top \mathbf{J} = \left[\begin{array}{ccc|ccc|ccc} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & -2 & 0 & 0 \\ \hline 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & -2 & 0 \\ \hline 0 & 0 & -2 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 0 & 2 & 0 \\ -2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 4 \end{array} \right]. \quad (5.13)$$

To take the pseudo-inverse of this matrix, we may use (5.9) where A is a matrix with columns spanning the tangent plane to the constraint surface. Since H is computed subject to the condition $\|H\|^2 = 3$, which represents a hypersphere, the constraint surface is perpendicular to the vector h corresponding to the computed homography H . A Householder matrix A (see section A4.1.2(p580)) corresponding to the vector h has the property that $Ah = (0, \dots, 0, 1)^T$, so the first 8 columns of A (denoted A_1) are perpendicular to h as required. This allows the pseudo-inverse to be computed exactly without using SVD. Applying (5.9) the pseudo-inverse is computed to be

$$\Sigma_h = (J^T J)^{+A_1} = A_1 (A_1^T (J^T J) A_1)^{-1} A_1^T = \frac{1}{18} \left[\begin{array}{ccc|ccc|ccc} 5 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & -1 \\ 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 & 9 & 0 & 0 \\ \hline 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 & 9 & 0 \\ \hline 0 & 0 & 9 & 0 & 0 & 0 & 18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 & 18 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 \end{array} \right]. \quad (5.14)$$

The diagonals give the individual variances of the entries of H . \triangle

This computed covariance is used to assess the accuracy of point transfer in example 5.14.

5.2.5 Error in both images

In the case of error in both images, computation of the covariance of the transformation is a bit more complicated. As seen in section 4.2.7(p101), one may define a set of $2n+8$ parameters, where 8 parameters describe the transformation matrix and $2n$ parameters \hat{x}_i represent estimates of the points in the first image. More conveniently, one may over-parametrize by using 9 parameters for the transformation H . The Jacobian matrix naturally splits up into two parts as $J = [A \mid B]$ where A and B are the derivatives with respect to the camera parameters and the points x_i respectively. Applying (5.10) one computes

$$J^T \Sigma_x^{-1} J = \begin{bmatrix} A^T \Sigma_x^{-1} A & A^T \Sigma_x^{-1} B \\ B^T \Sigma_x^{-1} A & B^T \Sigma_x^{-1} B \end{bmatrix}.$$

The pseudo-inverse of this matrix is the covariance of the parameter set and the top-left block of this pseudo-inverse is the covariance of the entries of H . A detailed discussion of this is given in section A6.4.1(p608), where it is shown how one can make use of the block structure of the Jacobian to simplify the computation.

In example 5.13 on estimating the covariance of H from four points in the previous section, the covariance turns out to be $\Sigma_h = 2(J^T \Sigma_x^{-1} J)^+$, namely twice the covariance computed for noise in one image only. This assumes that points are measured with the same covariance in both images. This simple relationship between the covariances in the one and two-image cases does not generally hold.

5.2.6 Using the covariance matrix in point transfer

Once one has the covariance, one may compute the uncertainty associated with a given point transfer. Consider a new point \mathbf{x} in the first image, not used in the computation of the transformation, H . The corresponding point in the second image is $\mathbf{x}' = H\mathbf{x}$. However, because of the uncertainty in the estimation of H , the correct location of the point \mathbf{x}' will also have associated uncertainty. One may compute this uncertainty from the covariance matrix of H .

The covariance matrix for the point \mathbf{x}' is given by the formula

$$\Sigma_{\mathbf{x}'} = J_h \Sigma_h J_h^T \quad (5.15)$$

where $J_h = \partial \mathbf{x}' / \partial \mathbf{h}$. A formula for $\partial \mathbf{x}' / \partial \mathbf{h}$ is given in (4.21–p129).

If in addition, the point \mathbf{x} itself is measured with some uncertainty, then one has instead

$$\Sigma_{\mathbf{x}'} = J_h \Sigma_h J_h^T + J_x \Sigma_x J_x^T \quad (5.16)$$

assuming that there is no cross-correlation between \mathbf{x} and \mathbf{h} , which is reasonable, since point \mathbf{x} is assumed to be a new point not used in the computation of the transformation H . A formula for the Jacobian matrix $J_x = \partial \mathbf{x}' / \partial \mathbf{x}$ is given in (4.20–p129).

The covariance matrix $\Sigma_{\mathbf{x}'}$ given by (5.15) is expressed in terms of the covariance matrix Σ_h of the transformation H . We have seen that this covariance matrix Σ_h depends on the particular constraint used in estimating H , according to (5.9). It may therefore appear that $\Sigma_{\mathbf{x}'}$ also depends on the particular method used to constrain H . It may however be verified that these formulae are independent of the particular constraint A used to compute the covariance matrix $\Sigma_p = (J^T \Sigma_x^{-1} J)^{+A}$.

Example 5.14. To continue example 5.13, let the computed 2D homography H be given by the identity matrix, with covariance matrix Σ_h as in (5.14). Consider an arbitrary point (x, y) mapped to the point $\mathbf{x}' = H\mathbf{x}$. In this case the covariance matrix $\Sigma_{\mathbf{x}'} = J_h \Sigma_h J_h^T$ may be computed symbolically to equal

$$\Sigma_{\mathbf{x}'} = \begin{bmatrix} \sigma_{x'x'} & \sigma_{x'y'} \\ \sigma_{x'y'} & \sigma_{y'y'} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 2 - x^2 + x^4 + y^2 + x^2 y^2 & xy(x^2 + y^2 - 2) \\ xy(x^2 + y^2 - 2) & 2 - y^2 + y^4 + x^2 + x^2 y^2 \end{bmatrix}.$$

Note that $\sigma_{x'x'}$ and $\sigma_{y'y'}$ are even functions of x and y , whereas $\sigma_{x'y'}$ is an odd function. This is a consequence of the symmetry about the x and y axes of the point set used to compute H . Also note that $\sigma_{x'x'}$ and $\sigma_{y'y'}$ differ by swapping x and y , which is a further consequence of the symmetry of the defining point set.

As may be seen, the variance $\sigma_{x'x'}$ varies as the fourth power of x , and hence the standard deviation varies as the square. This shows that extrapolating the values of transformed points $\mathbf{x}' = H\mathbf{x}$ far beyond the set of points used to compute H is not reliable. More specifically, the RMS uncertainty of the position of \mathbf{x}' is equal to $(\sigma_{x'x'} + \sigma_{y'y'})^{1/2} = \sqrt{\text{trace}(\Sigma_{\mathbf{x}'})}$ which one finds is equal to $(1 + (x^2 + y^2)^2)^{1/2} = (1 + r^4)^{1/2}$, where r is the radial distance from the origin. Note the interesting fact that the RMS error is only dependent on the radial distance. In fact, one may verify that the probability distribution for point \mathbf{x}' depends only on the radial distance of \mathbf{x}' , its

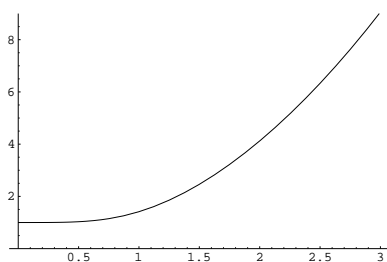


Fig. 5.5. *RMS error in the position of a projected point \mathbf{x}' as a function of radial distance of \mathbf{x}' from the origin. The homography \mathbf{H} is computed from 4 evenly spaced points on a unit circle around the origin with errors in the second image only. The RMS error is proportional to the assumed error in the points used to compute \mathbf{H} , and the vertical axis is calibrated in terms of this assumed error.*

two principal axes pointing radially and tangentially. Figure 5.5 shows the graph of the RMS error in \mathbf{x}' as a function of r . \triangle

This example has computed the covariance of a transferred point in the minimal case of four point correspondences. For more than four correspondences, the situation is not substantially different. Extrapolation beyond the set of points used to compute the homography is unreliable. In fact, one may show that if \mathbf{H} is computed from n points evenly spaced around a unit circle (instead of 4 as in the computation above) then the RMS error is equal to $\sigma_{x'x'} + \sigma_{y'y'} = 4(1 + r^4)/n$, so the error exhibits the same quadratic growth.

5.3 Monte Carlo estimation of covariance

The method of estimating covariance discussed in the previous sections has relied on an assumption of linearity. In other words, it has been assumed that the surface $f(\mathbf{h})$ is locally flat in the vicinity of the estimated point, at least over a region corresponding to the approximate extent of the noise distribution. It has also been assumed that the method of estimation of the transformation \mathbf{H} was the Maximum Likelihood Estimate. If the surface is not entirely flat then the estimate of covariance may be incorrect. In addition, a particular estimation method may be inferior to the ML estimate, thereby introducing additional uncertainty in the values of the estimated transformation \mathbf{H} .

A general (though expensive) method of getting an estimate of the covariance is by exhaustive simulation. Assuming that the noise is drawn from a given noise distribution, one starts with a set of point matches corresponding perfectly to a given transformation. One then adds noise to the points and computes the corresponding transformation using the chosen estimation procedure. The covariance of the transformation \mathbf{H} or a further transferred point is then computed statistically from multiple trials with noise drawn from the assumed noise distribution. This is illustrated for the case of the identity mapping in figure 5.6.

Both the analytical and the Monte Carlo methods of estimating covariance of the transformation \mathbf{H} may be applied to the estimation of covariance from real data for which one does not know the true value of \mathbf{H} . From the given data, an estimate of \mathbf{H} and the corresponding true values of the points \mathbf{x}'_i and \mathbf{x}_i are computed. Then the

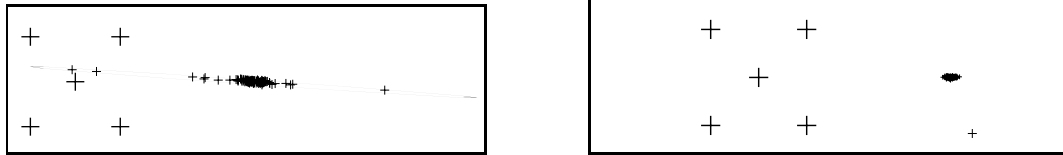


Fig. 5.6. Transfer of a point under the identity mapping for the normalized and unnormalized DLT algorithm. See also figure 4.4(p109) for further explanation.

covariance is computed as if the estimated values were the true values of the matched data points and the transformation. The resulting covariance matrix is assumed to be the covariance of the true transformation. This identification is based on the assumption that the true values of the data points are sufficiently close to the estimated values that the covariance matrix is essentially unaffected.

5.4 Closure

An extended discussion of bias and variance of estimated parameters is given in appendix 3(p568).

5.4.1 The literature

The derivations throughout this chapter have been considerably simplified by only using first-order Taylor expansions, and assuming Gaussian error distributions. Similar ideas (ML, covariance . . .) can be developed for other distributions by using the Fisher Information matrix. Related reading may be found in Kanatani [Kanatani-96], Press *et al.* [Press-88], and other statistical textbooks.

Criminisi *et al.* [Criminisi-99b] give many examples of the computed covariances in point transfer as the correspondences used to determine the homography vary in number and position.

5.4.2 Notes and exercises

- (i) Consider the problem of computing a best line fit to a set of 2D points in the plane using orthogonal regression. Suppose that N points are measured with independent standard deviations of σ in each coordinate. What is the expected RMS distance of each point from a fitted line? **Answer :** $\sigma ((n - 2)/n)^{1/2}$.
- (ii) (Harder) : In section 18.5.2(p450) a method is given for computing a projective reconstruction from a set of $n + 4$ point correspondences across m views, where 4 of the point correspondences are presumed to be known to be from a plane. Suppose the 4 planar correspondences are known exactly, and the other n image points are measured with 1 pixel error (each coordinate in each image). What is the expected residual error of $\|\mathbf{x}_j^i - \hat{\mathbf{P}}^i \mathbf{X}_j\|$?