1

Introduction – a Tour of Multiple View Geometry

This chapter is an introduction to the principal ideas covered in this book. It gives an *informal* treatment of these topics. Precise, unambiguous definitions, careful algebra, and the description of well honed estimation algorithms is postponed until chapter 2 and the following chapters in the book. Throughout this introduction we will generally not give specific forward pointers to these later chapters. The material referred to can be located by use of the index or table of contents.

1.1 Introduction – the ubiquitous projective geometry

We are all familiar with projective transformations.When we look at a picture, we see squares that are not squares, or circles that are not circles. The transformation that maps these planar objects onto the picture is an example of a projective transformation.

So what properties of geometry are preserved by projective transformations? Certainly, shape is not, since a circle may appear as an ellipse. Neither are lengths since two perpendicular radii of a circle are stretched by different amounts by the projective transformation. Angles, distance, ratios of distances – none of these are preserved, and it may appear that very little geometry is preserved by a projective transformation. However, a property that is preserved is that of straightness. It turns out that this is the most general requirement on the mapping, and we may define a projective transformation of a plane as any mapping of the points on the plane that preserves straight lines.

To see why we will require projective geometry we start from the familiar Euclidean geometry. This is the geometry that describes angles and shapes of objects. Euclidean geometry is troublesome in one major respect – we need to keep making an exception to reason about some of the basic concepts of the geometry – such as intersection of lines. Two lines (we are thinking here of 2-dimensional geometry) almost always meet in a point, but there are some pairs of lines that do not do so – those that we call parallel. A common linguistic device for getting around this is to say that parallel lines meet "at infinity". However this is not altogether convincing, and conflicts with another dictum, that infinity does not exist, and is only a convenient fiction. We can get around this by enhancing the Euclidean plane by the addition of these points at infinity where parallel lines meet, and resolving the difficulty with infinity by calling them "ideal points."

By adding these points at infinity, the familiar Euclidean space is transformed into a new type of geometric object, projective space. This is a very useful way of thinking, since we are familiar with the properties of Euclidean space, involving concepts such as distances, angles, points, lines and incidence. There is nothing very mysterious about projective space – it is just an extension of Euclidean space in which two lines always meet in a point, though sometimes at mysterious points at infinity.

Coordinates. A point in Euclidean 2-space is represented by an ordered pair of real numbers, (x, y). We may add an extra coordinate to this pair, giving a triple $(x, y, 1)$, that we declare to represent the same point. This seems harmless enough, since we can go back and forward from one representation of the point to the other, simply by adding or removing the last coordinate. We now take the important conceptual step of asking why the last coordinate needs to be 1 – after all, the others two coordinates are not so constrained. What about a coordinate triple $(x, y, 2)$. It is here that we make a definition and say that $(x, y, 1)$ and $(2x, 2y, 2)$ represent the same point, and furthermore, (kx, ky, k) represents the same point as well, for any non-zero value k. Formally, points are represented by *equivalence classes* of coordinate triples, where two triples are equivalent when they differ by a common multiple. These are called the *homogeneous coordinates* of the point. Given a coordinate triple (kx, ky, k), we can get the original coordinates back by dividing by k to get (x, y).

The reader will observe that although $(x, y, 1)$ represents the same point as the coordinate pair (x, y), there is no point that corresponds to the triple $(x, y, 0)$. If we try to divide by the last coordinate, we get the point $(x/0, y/0)$ which is infinite. This is how the points at infinity arise then. They are the points represented by homogeneous coordinates in which the last coordinate is zero.

Once we have seen how to do this for 2-dimensional Euclidean space, extending it to a projective space by representing points as homogeneous vectors, it is clear that we can do the same thing in any dimension. The Euclidean space $\mathbb{R}^n$ can be extended to a projective space $\mathbb{P}^n$ by representing points as homogeneous vectors. It turns out that the points at infinity in the two-dimensional projective space form a line, usually called the *line at infinity*. In three-dimensions they form the *plane at infinity*.

Homogeneity. In classical Euclidean geometry all points are the same. There is no distinguished point. The whole of the space is homogeneous. When coordinates are added, one point is seemingly picked out as the origin. However, it is important to realize that this is just an accident of the particular coordinate frame chosen. We could just as well find a different way of coordinatizing the plane in which a different point is considered to be the origin. In fact, we can consider a change of coordinates for the Euclidean space in which the axes are shifted and rotated to a different position. We may think of this in another way as the space itself translating and rotating to a different position. The resulting operation is known as a Euclidean transform.

A more general type of transformation is that of applying a linear transformation to $\mathbb{R}^n$, followed by a Euclidean transformation moving the origin of the space. We may think of this as the space moving, rotating and finally *stretching* linearly possibly by different ratios in different directions. The resulting transformation is known as an *affine* transformation.

The result of either a Euclidean or an affine transformation is that points at infinity remain at infinity. Such points are in some way preserved, at least as a set, by such transformations. They are in some way distinguished, or special in the context of Euclidean or *affine* geometry.

From the point of view of projective geometry, points at infinity are not any different from other points. Just as Euclidean space is uniform, so is projective space. The property that points at infinity have final coordinate zero in a homogeneous coordinate representation is nothing other than an accident of the choice of coordinate frame. By analogy with Euclidean or affine transformations, we may define a *projective transformation* of projective space. A linear transformation of Euclidean space $\mathbb{R}^n$ is represented by matrix multiplication applied to the coordinates of the point. In just the same way a projective transformation of projective space $\mathbb{P}^n$ is a mapping of the homogeneous coordinates representing a point (an $(n+1)$-vector), in which the coordinate vector is multiplied by a non-singular matrix. Under such a mapping, points at infinity (with final coordinate zero) are mapped to arbitrary other points. The points at infinity are not preserved. Thus, a projective transformation of projective space $\mathbb{P}^n$ is represented by a linear transformation of homogeneous coordinates

$$\mathbf{X}' = \mathtt{H}_{(n+1)\times(n+1)}\mathbf{X}.$$

In computer vision problems, projective space is used as a convenient way of representing the real 3D world, by extending it to the 3-dimensional (3D) projective space. Similarly images, usually formed by projecting the world onto a 2-dimensional representation, are for convenience extended to be thought of as lying in the 2-dimensional projective space. In reality, the real world, and images of it do not contain points at infinity, and we need to keep our finger on which are the fictitious points, namely the line at infinity in the image and the plane at infinity in the world. For this reason, although we usually work with the projective spaces, we are aware that the line and plane at infinity are in some way special. This goes against the spirit of pure projective geometry, but makes it useful for our practical problems. Generally we try to have it both ways by treating all points in projective space as equals when it suits us, and singling out the line at infinity in space or the plane at infinity in the image when that becomes necessary.

1.1.1 Affine and Euclidean Geometry

We have seen that projective space can be obtained from Euclidean space by adding a line (or plane) at infinity. We now consider the reverse process of going backwards. This discussion is mainly concerned with two and three-dimensional projective space.

Affine geometry. We will take the point of view that the projective space is initially homogeneous, with no particular coordinate frame being preferred. In such a space, there is no concept of parallelism of lines, since parallel lines (or planes in the three-dimensional case) are ones that meet at infinity. However, in projective space, there is no concept of which points are at infinity – all points are created equal. We say that parallelism is not a concept of projective geometry. It is simply meaningless to talk about it.

In order for such a concept to make sense, we need to pick out some particular line, and decide that this is the line at infinity. This results in a situation where although all points are created equal, some are more equal than others. Thus, start with a blank sheet of paper, and imagine that it extends to infinity and forms a projective space $\mathbb{P}^2$. What we see is just a small part of the space, that looks a lot like a piece of the ordinary Euclidean plane. Now, let us draw a straight line on the paper, and declare that this is the line at infinity. Next, we draw two other lines that intersect at this distinguished line. Since they meet at the "line at infinity" we define them as being parallel. The situation is similar to what one sees by looking at an infinite plane. Think of a photograph taken in a very flat region of the earth. The points at infinity in the plane show up in the image as the horizon line. Lines, such as railway tracks show up in the image as lines meeting at the horizon. Points in the image lying above the horizon (the image of the sky) apparently do not correspond to points on the world plane. However, if we think of extending the corresponding ray backwards behind the camera, it will meet the plane at a point behind the camera. Thus there is a one-to-one relationship between points in the image and points in the world plane. The points at infinity in the world plane correspond to a real horizon line in the image, and parallel lines in the world correspond to lines meeting at the horizon. From our point of view, the world plane and its image are just alternative ways of viewing the geometry of a projective plane, plus a distinguished line. The geometry of the projective plane and a distinguished line is known as *affine geometry* and any projective transformation that maps the distinguished line in one space to the distinguished line of the other space is known as an *affine transformation*.

By identifying a special line as the "line at infinity" we are able to define parallelism of straight lines in the plane. However, certain other concepts make sense as well, as soon as we can define parallelism. For instance, we may define equalities of intervals between two points on parallel lines. For instance, if A, B, C and D are points, and the lines AB and CD are parallel, then we define the two intervals AB and CD to have *equal length* if the lines AC and BD are also parallel. Similarly, two intervals on the same line are equal if there exists another interval on a parallel line that is equal to both.

Euclidean geometry. By distinguishing a special line in a projective plane, we gain the concept of parallelism and with it *affine geometry*. Affine geometry is seen as specialization of projective geometry, in which we single out a particular line (or plane – according to the dimension) and call it the line at infinity.

Next, we turn to Euclidean geometry and show that by singling out some special feature of the line or plane at infinity affine geometry becomes Euclidean geometry. In doing so, we introduce one of the most important concepts of this book, the *absolute conic*.

We begin by considering two-dimensional geometry, and start with circles. Note that a circle is not a concept of affine geometry, since arbitrary stretching of the plane, which preserves the line at infinity, turns the circle into an ellipse. Thus, affine geometry does not distinguish between circles and ellipses.

In Euclidean geometry however, they are distinct, and have an important difference. Algebraically, an ellipse is described by a second-degree equation. It is therefore expected, and true that two ellipses will most generally intersect in four points. However, it is geometrically evident that two distinct circles can not intersect in more than two points. Algebraically, we are intersecting two second-degree curves here, or equivalently solving two quadratic equations. We should expect to get four solutions. The question is, what is special about circles that they only intersect in two points.

The answer to this question is of course that there exist two other solutions, the two circles meeting in two other *complex* points. We do not have to look very far to find these two points.

The equation for a circle in homogeneous coordinates (x, y, w) is of the form

$$(x - aw)^2 + (y - bw)^2 = r^2 w^2$$

This represents the circle with centre represented in homogeneous coordinates as $(x_0, y_0, w_0)^\mathsf{T} = (a, b, 1)^\mathsf{T}$. It is quickly verified that the points $(x, y, w)^\mathsf{T} = (1, \pm i, 0)^\mathsf{T}$ lie on every such circle. To repeat this interesting fact, every circle passes through the points $(1, \pm i, 0)^\mathsf{T}$, and therefore they lie in the intersection of any two circles. Since their final coordinate is zero, these two points lie on the line at infinity. For obvious reasons, they are called the *circular points* of the plane. Note that although the two circular points are complex, they satisfy a pair of real equations: $x^2 + y^2 = 0$; $w = 0$.

This observation gives the clue of how we may define Euclidean geometry. Euclidean geometry arises from projective geometry by singling out first a line at infinity and subsequently, two points called *circular points* lying on this line. Of course the circular points are complex points, but for the most part we do not worry too much about this. Now, we may define a circle as being any conic (a curve defined by a second-degree equation) that passes through the two circular points. Note that in the standard Euclidean coordinate system, the circular points have the coordinates $(1, \pm i, 0)^\mathsf{T}$. In assigning a Euclidean structure to a projective plane, however, we may designate any line and any two (complex) points on that line as being the line at infinity and the circular points.

As an example of applying this viewpoint, we note that a general conic may be found passing through five arbitrary points in the plane, as may be seen by counting the number of coefficients of a general quadratic equation $ax^2 + by^2 + \ldots + fw^2 = 0$. A circle on the other hand is defined by only three points. Another way of looking at this is that it is a conic passing through two special points, the circular points, as well as three other points, and hence as any other conic, it requires five points to specify it uniquely.

It should not be a surprise that as a result of singling out two circular points one obtains the whole of the familiar Euclidean geometry. In particular, concepts such as angle and length ratios may be defined in terms of the circular points. However, these concepts are most easily defined in terms of some coordinate system for the Euclidean plane, as will be seen in later chapters.

3D Euclidean geometry. We saw how the Euclidean plane is defined in terms of the projective plane by specifying a line at infinity and a pair of circular points. The same idea may be applied to 3D geometry. As in the two-dimensional case, one may look carefully at spheres, and how they intersect. Two spheres intersect in a circle, and not in a general fourth-degree curve, as the algebra suggests, and as two general ellipsoids (or other *quadric* surfaces) do. This line of thought leads to the discovery that in homogeneous coordinates $(\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{T})^{\mathsf{T}}$ all spheres intersect the plane at infinity in a curve with the equations: $\mathrm{X}^2 + \mathrm{Y}^2 + \mathrm{Z}^2 = 0$; $\mathrm{T} = 0$. This is a second-degree curve (a conic) lying on the plane at infinity, and consisting only of complex points. It is known as the *absolute conic* and is one of the key geometric entities in this book, most particularly because of its connection to camera calibration, as will be seen later.

The absolute conic is defined by the above equations only in the Euclidean coordinate system. In general we may consider 3D Euclidean space to be derived from projective space by singling out a particular plane as the plane at infinity and specifying a particular conic lying in this plane to be the absolute conic. These entities may have quite general descriptions in terms of a coordinate system for the projective space.

We will not here go into details of how the absolute conic determines the complete Euclidean 3D geometry. A single example will serve. Perpendicularity of lines in space is not a valid concept in affine geometry, but belongs to Euclidean geometry. The perpendicularity of lines may be defined in terms of the absolute conic, as follows. By extending the lines until they meet the plane at infinity, we obtain two points called the directions of the two lines. Perpendicularity of the lines is defined in terms of the relationship of the two directions to the absolute conic. The lines are perpendicular if the two directions are conjugate points with respect to the absolute conic (see figure 3.8(*p*83)). The geometry and algebraic representation of conjugate points are defined in section 2.8.1(*p*58). Briefly, if the absolute conic is represented by a 3×3 symmetric matrix Ω_∞, and the directions are the points $\mathbf{d}_1$ and $\mathbf{d}_2$, then they are conjugate with respect to Ω_∞ if $\mathbf{d}_1^{\mathsf{T}} \Omega_\infty \mathbf{d}_2 = 0$. More generally, angles may be defined in terms of the absolute conic in any arbitrary coordinate system, as expressed by (3.23–*p*82).

1.2 Camera projections

One of the principal topics of this book is the process of image formation, namely the formation of a two-dimensional representation of a three-dimensional world, and what we may deduce about the 3D structure of what appears in the images.

The drop from three-dimensional world to a two-dimensional image is a projection process in which we lose one dimension. The usual way of modelling this process is by *central projection* in which a ray from a point in space is drawn from a 3D world point through a fixed point in space, the *centre of projection*. This ray will intersect a specific plane in space chosen as the *image plane*. The intersection of the ray with the image plane represents the image of the point. If the 3D structure lies on a plane then there is no drop in dimension.

This model is in accord with a simple model of a camera, in which a ray of light from a point in the world passes through the lens of a camera and impinges on a film or digital device, producing an image of the point. Ignoring such effects as focus and lens thickness, a reasonable approximation is that all the rays pass through a single point, the centre of the lens.

In applying projective geometry to the imaging process, it is customary to model the world as a 3D projective space, equal to $\mathbb{R}^3$ along with points at infinity. Similarly the model for the image is the 2D projective plane $\mathbb{P}^2$. Central projection is simply a map from $\mathbb{P}^3$ to $\mathbb{P}^2$. If we consider points in $\mathbb{P}^3$ written in terms of homogeneous coordinates $(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{T})^\mathsf{T}$ and let the centre of projection be the origin $(0, 0, 0, 1)^\mathsf{T}$, then we see that the set of all points $(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{T})^\mathsf{T}$ for fixed $\mathtt{X}$, $\mathtt{Y}$ and $\mathtt{Z}$, but varying $\mathtt{T}$ form a single ray passing through the point centre of projection, and hence all mapping to the same point. Thus, the final coordinate of $(\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{T})$ is irrelevant to where the point is imaged. In fact, the image point is the point in $\mathbb{P}^2$ with homogeneous coordinates $(\mathtt{X}, \mathtt{Y}, \mathtt{Z})^\mathsf{T}$. Thus, the mapping may be represented by a mapping of 3D homogeneous coordinates, represented by a 3×4 matrix $\mathtt{P}$ with the block structure $\mathtt{P} = [\mathtt{I}_{3\times3}|\mathbf{0}_3]$, where $\mathtt{I}_{3\times3}$ is the 3×3 identity matrix and $\mathbf{0}_3$ a zero 3-vector. Making allowance for a different centre of projection, and a different projective coordinate frame in the image, it turns out that the most general imaging projection is represented by an arbitrary 3×4 matrix of rank 3, acting on the homogeneous coordinates of the point in $\mathbb{P}^3$ mapping it to the imaged point in $\mathbb{P}^2$. This matrix $\mathtt{P}$ is known as the camera matrix.

In summary, the action of a projective camera on a point in space may be expressed in terms of a linear mapping of homogeneous coordinates as

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \mathtt{P}_{3\times4} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ \mathrm{T} \end{pmatrix}$$

Furthermore, if all the points lie on a plane (we may choose this as the plane $\mathtt{Z} = 0$) then the linear mapping reduces to

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \mathtt{H}_{3\times3} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{T} \end{pmatrix}$$

which is a projective transformation.

Cameras as points. In a central projection, points in $\mathbb{P}^3$ are mapped to points in $\mathbb{P}^2$, all points in a ray passing through the centre of projection projecting to the same point in an image. For the purposes of image projection, it is possible to consider all points along such a ray as being equal. We can go one step further, and think of the ray through the projection centre as representing the image point. Thus, the set of all image points is the same as the set of rays through the camera centre. If we represent

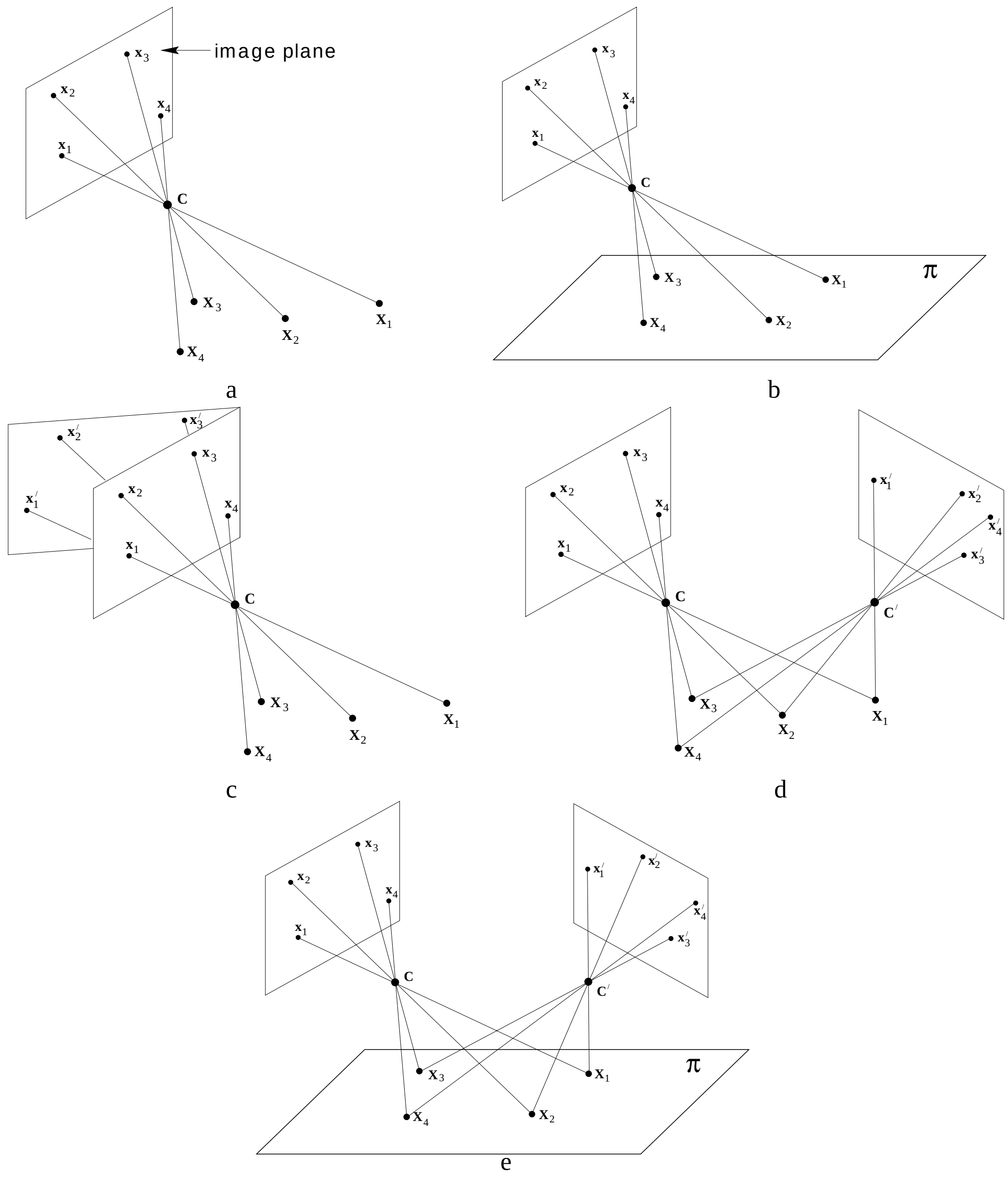


Fig. 1.1. **The camera centre is the essence.** *(a) Image formation: the image points* $\mathbf{x}_i$ *are the intersection of a plane with rays from the space points* $\mathbf{X}_i$ *through the camera centre* $\mathbf{C}$. *(b) If the space points are coplanar then there is a projective transformation between the world and image planes,* $\mathbf{x}_i = \mathtt{H}_{3\times3}\mathbf{X}_i$. *(c) All images with the same camera centre are related by a projective transformation,* $\mathbf{x}'_i = \mathtt{H}'_{3\times3}\mathbf{x}_i$. *Compare (b) and (c) – in both cases planes are mapped to one another by rays through a centre. In (b) the mapping is between a scene and image plane, in (c) between two image planes. (d) If the camera centre moves, then the images are in general not related by a projective transformation, unless (e) all the space points are coplanar.*

the ray from $(0, 0, 0, 1)^\mathsf{T}$ through the point $(\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{T})^\mathsf{T}$ by its first three coordinates $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\mathsf{T}$, it is easily seen that for any constant k, the ray $k(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\mathsf{T}$ represents the same ray. Thus the rays themselves are represented by homogeneous coordinates. In fact they make up a 2-dimensional space of rays. The set of rays themselves may be thought of as a representation of the image space $\mathbb{P}^2$. In this representation of the image, all that is important is the camera centre, for this alone determines the set of rays forming the image. Different camera matrices representing the image formation from the same centre of projection reflect only different coordinate frames for the set of rays forming the image. Thus two images taken from the same point in space are projectively equivalent. It is only when we start to measure points in an image, that a particular coordinate frame for the image needs to be specified. Only then does it become necessary to specify a particular camera matrix. In short, modulo field-of-view which we ignore for now, all images acquired with the same camera centre are equivalent – they can be mapped onto each other by a projective transformation without any information about the 3D points or position of the camera centre. These issues are illustrated in figure 1.1.

Calibrated cameras. To understand fully the Euclidean relationship between the image and the world, it is necessary to express their relative Euclidean geometry. As we have seen, the Euclidean geometry of the 3D world is determined by specifying a particular plane in $\mathbb{P}^3$ as being the plane at infinity, and a specific conic Ω in that plane as being the absolute conic. For a camera not located on the plane at infinity, the plane at infinity in the world maps one-to-one onto the image plane. This is because any point in the image defines a ray in space that meets the plane at infinity in a single point. Thus, the plane at infinity in the world does not tell us anything new about the image. The absolute conic, however being a conic in the plane at infinity must project to a conic in the image. The resulting image curve is called the Image of the Absolute Conic, or IAC. If the location of the IAC is known in an image, then we say that the camera is *calibrated*.

In a calibrated camera, it is possible to determine the angle between the two rays back-projected from two points in the image. We have seen that the angle between two lines in space is determined by where they meet the plane at infinity, relative to the absolute conic. In a calibrated camera, the plane at infinity and the absolute conic Ω_∞ are projected one-to-one onto the image plane and the IAC, denoted $\boldsymbol{\omega}$. The projective relationship between the two image points and $\boldsymbol{\omega}$ is exactly equal to the relationship between the intersections of the back-projected rays with the plane at infinity, and Ω_∞. Consequently, knowing the IAC, one can measure the angle between rays by direct measurements in the image. Thus, for a calibrated camera, one can measure angles between rays, compute the field of view represented by an image patch or determine whether an ellipse in the image back-projects to a circular cone. Later on, we will see that it helps us to determine the *Euclidean structure* of a reconstructed scene.

Example 1.1. 3D reconstructions from paintings

Using techniques of projective geometry, it is possible in many instances to reconstruct scenes from a single image. This cannot be done without some assumptions being made about the imaged scene. Typical techniques involve the analysis of features such as parallel lines and vanishing points to determine the affine structure of the scene, for

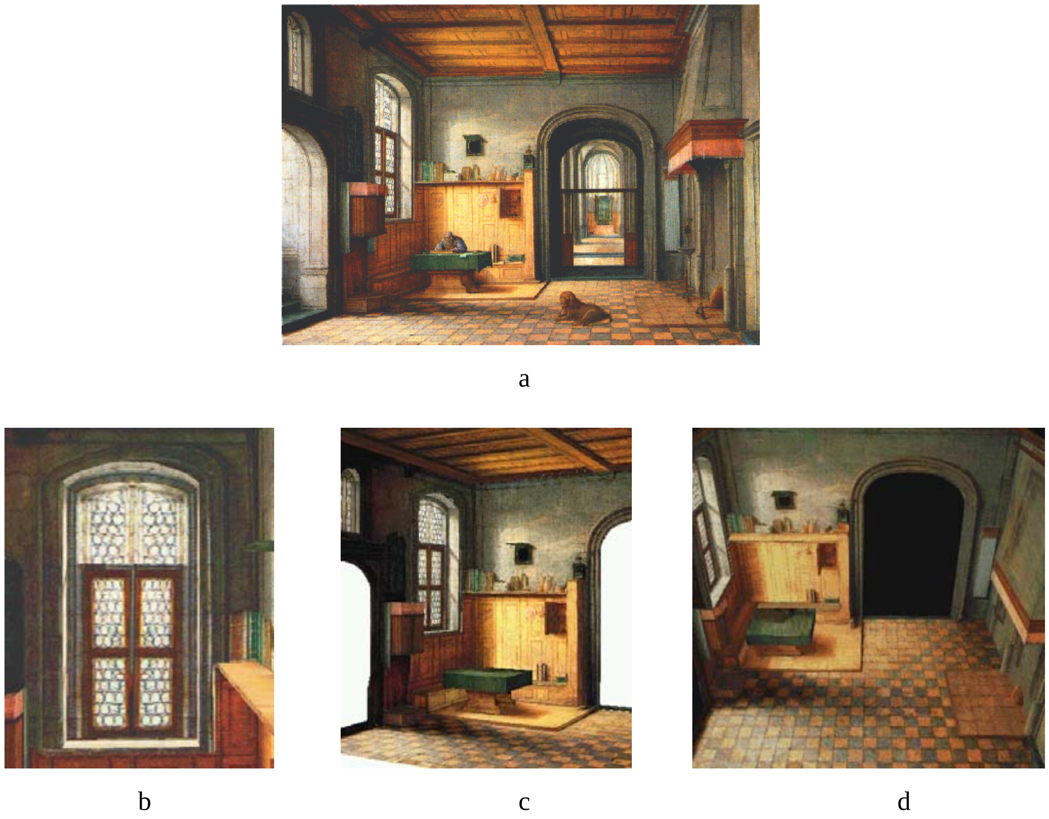


Fig. 1.2. **Single view reconstruction.** *(a) Original painting – St. Jerome in his study, 1630, Hendrick van Steenwijck (1580-1649), Joseph R. Ritman Private Collection, Amsterdam, The Netherlands. (b) (c)(d) Views of the 3D model created from the painting. Figures courtesy of Antonio Criminisi.*

example by determining the line at infinity for observed planes in the image. Knowledge (or assumptions) about angles observed in the scene, most particularly orthogonal lines or planes, can be used to upgrade the affine reconstruction to Euclidean.

It is not yet possible for such techniques to be fully automatic. However, projective geometric knowledge may be built into a system that allows user-guided single-view reconstruction of the scene.

Such techniques have been used to reconstruct 3D texture mapped graphical models derived from old-master paintings. Starting in the Renaissance, paintings with extremely accurate perspective were produced. In figure 1.2 a reconstruction carried out from such a painting is shown. △

1.3 Reconstruction from more than one view

We now turn to one of the major topics in the book – that of reconstructing a scene from several images. The simplest case is that of two images, which we will consider first. As a mathematical abstraction, we restrict the discussion to “scenes” consisting of points only.

The usual input to many of the algorithms given in this book is a set of point correspondences. In the two-view case, therefore, we consider a set of correspondences

$\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ in two images. It is assumed that there exist some camera matrices, $\mathtt{P}$ and $\mathtt{P}'$ and a set of 3D points $\mathbf{X}_i$ that give rise to these image correspondences in the sense that $\mathtt{P}\mathbf{X}_i = \mathbf{x}_i$ and $\mathtt{P}'\mathbf{X}_i = \mathbf{x}'_i$. Thus, the point $\mathbf{X}_i$ projects to the two given data points. However, neither the cameras (represented by projection matrices $\mathtt{P}$ and $\mathtt{P}'$), nor the points $\mathbf{X}_i$ are known. It is our task to determine them.

It is clear from the outset that it is impossible to determine the positions of the points uniquely. This is a general ambiguity that holds however many images we are given, and even if we have more than just point correspondence data. For instance, given several images of a cube, it is impossible to tell its absolute position (is it located in a night-club in Addis Ababa, or the British Museum), its orientation (which face is facing north) or its scale. We express this by saying that the reconstruction is possible at best up to a similarity transformation of the world. However, it turns out that unless something is known about the calibration of the two cameras, the ambiguity in the reconstruction is expressed by a more general class of transformations – projective transformations.

This ambiguity arises because it is possible to apply a projective transformation (represented by a 4×4 matrix $\mathtt{H}$) to each point $\mathbf{X}_i$, and on the right of each camera matrix $\mathtt{P}_j$, without changing the projected image points, thus:

$$\mathtt{P}_j\mathbf{X}_i = (\mathtt{P}_j\mathtt{H}^{-1})(\mathtt{H}\mathbf{X}_i). \tag{1.1}$$

There is no compelling reason to choose one set of points and camera matrices over the other. The choice of $\mathtt{H}$ is essentially arbitrary, and we say that the reconstruction has a projective ambiguity, or is a *projective reconstruction*.

However, the good news is that this is the worst that can happen. It is possible to reconstruct a set of points from two views, up to an unavoidable projective ambiguity. Well, to be able to say this, we need to make a few qualifications; there must be sufficiently many points, at least seven, and they must not lie in one of various well-defined *critical configurations*.

The basic tool in the reconstruction of point sets from two views is the *fundamental matrix*, which represents the constraint obeyed by image points $\mathbf{x}$ and $\mathbf{x}'$ if they are to be images of the same 3D point. This constraint arises from the coplanarity of the camera centres of the two views, the images points and the space point. Given the fundamental matrix $\mathtt{F}$, a pair of matching points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ must satisfy

$$\mathbf{x}'^{\mathsf{T}}_i\mathtt{F}\mathbf{x}_i = 0$$

where $\mathtt{F}$ is a 3×3 matrix of rank 2. These equations are linear in the entries of the matrix $\mathtt{F}$, which means that if $\mathtt{F}$ is unknown, then it can be computed from a set of point correspondences.

A pair of camera matrices $\mathtt{P}$ and $\mathtt{P}'$ uniquely determine a fundamental matrix $\mathtt{F}$, and conversely, the fundamental matrix determines the pair of camera matrices, up to a 3D projective ambiguity. Thus, the fundamental matrix encapsulates the complete projective geometry of the pair of cameras, and is unchanged by projective transformation of 3D.

The fundamental-matrix method for reconstructing the scene is very simple, consisting of the following steps:

(i) Given several point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ across two views, form linear equations in the entries of $\mathtt{F}$ based on the coplanarity equations $\mathbf{x}'^{\mathsf{T}}_i \mathtt{F} \mathbf{x}_i = 0$.
(ii) Find $\mathtt{F}$ as the solution to a set of linear equations.
(iii) Compute a pair of camera matrices from $\mathtt{F}$ according to the simple formula given in section 9.5(*p*253).
(iv) Given the two cameras $(\mathtt{P}, \mathtt{P}')$ and the corresponding image point pairs $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, find the 3D point $\mathbf{X}_i$ that projects to the given image points. Solving for $\mathbf{X}$ in this way is known as *triangulation*.

The algorithm given here is an outline only, and each part of it is examined in detail in this book. The algorithm should not be implemented directly from this brief description.

1.4 Three-view geometry

In the last section it was discussed how reconstruction of a set of points, and the relative placement of the cameras, is possible from two views of a set of points. The reconstruction is possible only up to a projective transformation of space, and the corresponding adjustment to the camera matrices.

In this section, we consider the case of three views. Whereas for two views, the basic algebraic entity is the fundamental matrix, for three views this role is played by the trifocal tensor. The trifocal tensor is a $3 \times 3 \times 3$ array of numbers that relate the coordinates of corresponding points or lines in three views. Just as the fundamental matrix is determined by the two camera matrices, and determines them up to projective transformation, so in three views, the trifocal tensor is determined by the three camera matrices, and in turn determines them, again up to projective transformation. Thus, the trifocal tensor encapsulates the relative projective geometry of the three cameras.

For reasons that will be explained in chapter 15 it is usual to write some of the indices of a tensor as lower and some as upper indices. These are referred to as the covariant and contravariant indices. The trifocal tensor is of the form $\mathcal{T}_i^{jk}$, having two upper and one lower index.

The most basic relationship between image entities in three views concerns a correspondence between two lines and a point. We consider a correspondence $\mathbf{x} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ between a point $\mathbf{x}$ in one image and two lines $\mathbf{l}'$ and $\mathbf{l}''$ in the other two images. This relationship means that there is a point $\mathbf{X}$ in space that maps to $\mathbf{x}$ in the first image, and to points $\mathbf{x}'$ and $\mathbf{x}''$ lying on the lines $\mathbf{l}'$ and $\mathbf{l}''$ in the other two images. The coordinates of these three images are then related via the trifocal tensor relationship:

$$\sum_{ijk} x^i l'_j l''_k \mathcal{T}_i^{jk} = 0. \tag{1.2}$$

This relationship gives a single linear relationship between the elements of the tensor. With sufficiently many such correspondences, it is possible to solve linearly for the elements of the tensor. Fortunately, one can obtain more equations from a point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$. In fact, in this situation, one can choose any lines $\mathbf{l}'$ and $\mathbf{l}''$ passing through the points $\mathbf{x}'$ and $\mathbf{x}''$ and generate a relation of the sort (1.2). Since it is possible to choose two independent lines passing through $\mathbf{x}'$, and two others passing through $\mathbf{x}''$, one can obtain four independent equations in this way. A total of seven point correspondences are sufficient to compute the trifocal tensor linearly in this way. It can be computed from a minimum of six point correspondences using a non-linear method.

The 27 elements of the tensor are not independent, however, but are related by a set of so called *internal constraints*. These constraints are quite complicated, but tensors satisfying the constraints can be computed in various ways, for instance by using the 6 point non-linear method. The fundamental matrix (which is a 2-view tensor) also satisfies an internal constraint but a relatively simple one: the elements obey $\det \mathtt{F} = 0$.

As with the fundamental matrix, once the trifocal tensor is known, it is possible to extract the three camera matrices from it, and thereby obtain a reconstruction of the scene points and lines. As ever, this reconstruction is unique only up to a 3D projective transformation; it is a projective reconstruction.

Thus, we are able to generalize the method for two views to three views. There are several advantages to using such a three-view method for reconstruction.

(i) It is possible to use a mixture of line and point correspondences to compute the projective reconstruction. With two views, only point correspondences can be used.

(ii) Using three views gives greater stability to the reconstruction, and avoids unstable configurations that may occur using only two views for the reconstruction.

1.5 Four view geometry and n-view reconstruction

It is possible to go one more step with tensor-based methods and define a quadrifocal tensor relating entities visible in four views. This method is seldom used, however, because of the relative difficulty of computing a quadrifocal tensor that obey its internal constraints. Nevertheless, it does provide a non-iterative method for computing a projective reconstruction based on four views. The tensor method does not extend to more than four views, however, and so reconstruction from more than four views becomes more difficult.

Many methods have been considered for reconstruction from several views, and we consider a few of these in the book. One way to proceed is to reconstruct the scene bit by bit, using three-view or two-view techniques. Such a method may be applied to any image sequence, and with care in selecting the right triples to use, it will generally succeed.

There are methods that can be used in specific circumstances. The task of reconstruction becomes easier if we are able to apply a simpler camera model, known as the *affine camera*. This camera model is a fair approximation to perspective projection whenever the distance to the scene is large compared with the difference in depth between the back and front of the scene. If a set of points are visible in all of a set of n views involving an affine camera, then a well-known algorithm, the *factorization algorithm*, can be used to compute both the structure of the scene, and the specific camera models in one step using the Singular Value Decomposition. This algorithm is very reliable and simple to implement. Its main difficulties are the use of the affine camera model, rather than a full projective model, and the requirement that all the points be visible in all views.

This method has been extended to projective cameras in a method known as projective factorization. Although this method is generally satisfactory, it can not be proven to converge to the correct solution in all cases. Besides, it also requires all points to be visible in all images.

Other methods for n-view reconstruction involve various assumptions, such as knowledge of four coplanar points in the world visible in all views, or six or seven points that are visible in all images in the sequence. Methods that apply to specific motion sequences, such as linear motion, planar motion or single axis (turntable) motion have also been developed.

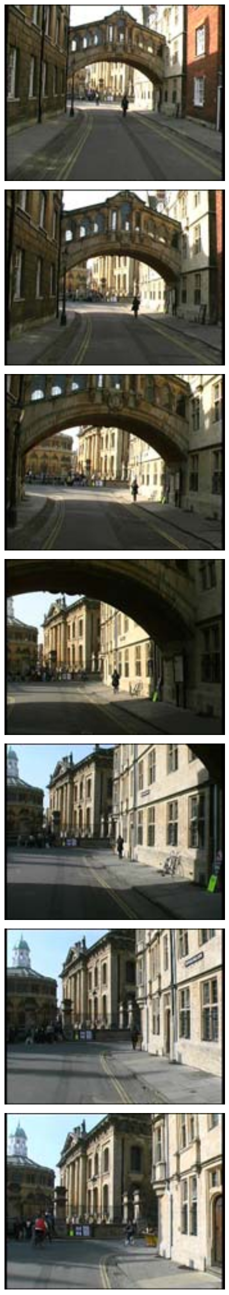
The dominant methodology for the general reconstruction problem is *bundle adjustment*. This is an iterative method, in which one attempts to fit a non-linear model to the measured data (the point correspondences). The advantage of bundle-adjustment is that it is a very general method that may be applied to a wide range of reconstruction and optimization problems. It may be implemented in such a way that the discovered solution is the Maximum Likelihood solution to the problem, that is a solution that is in some sense optimal in terms of a model for the inaccuracies of image measurements.

Unfortunately, bundle adjustment is an iterative process, which can not be guaranteed to converge to the optimal solution from an arbitrary starting point. Much research in reconstruction methods seeks easily computable non-optimal solutions that can be used as a starting point for bundle adjustment. An initialization step followed by bundle adjustment is the generally preferred technique for reconstruction. A common impression is that bundle-adjustment is necessarily a slow technique. The truth is that it is quite efficient when implemented carefully. A lengthy appendix in this book deals with efficient methods of bundle adjustment.

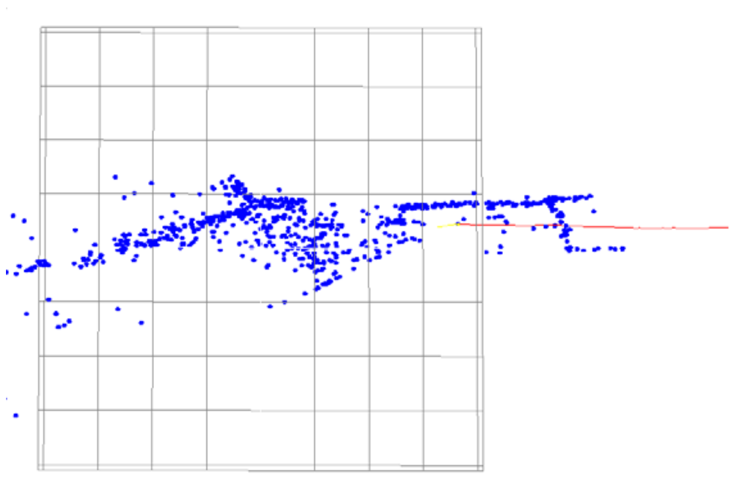
Using n-view reconstruction techniques, it is possible to carry out reconstructions automatically from quite long sequences of images. An example is given in figure 1.3, showing a reconstruction from 700 frames.

1.6 Transfer

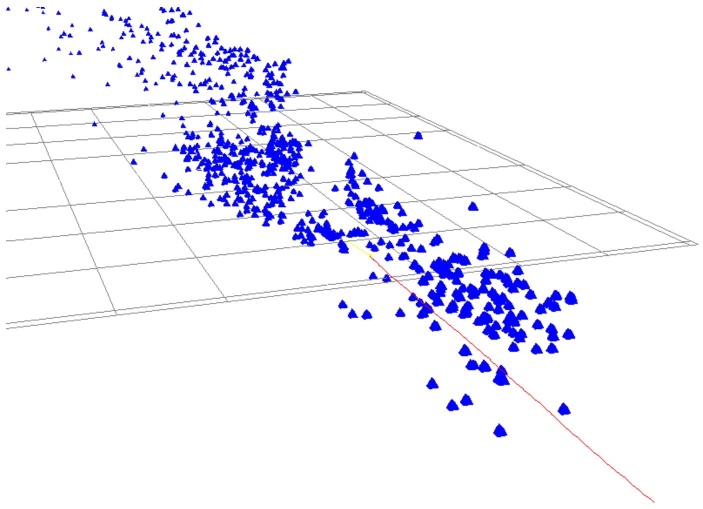
We have discussed 3D reconstruction from a set of images. Another useful application of projective geometry is that of *transfer*: given the position of a point in one (or more) image(s), determine where it will appear in all other images of the set. To do this, we must first establish the relationship between the cameras using (for instance) a set of auxiliary point correspondences. Conceptually transfer is straightforward given that a reconstruction is possible. For instance, suppose the point is identified in two views (at $\mathbf{x}$ and $\mathbf{x}'$) and we wish to know its position $\mathbf{x}''$ in a third, then this may be computed by the following steps:



(a)



(b)



(c)

Fig. 1.3. **Reconstruction.** *(a) Seven frames of a 700 frame sequence acquired by a hand held camera whilst walking down a street in Oxford. (b)(c) Two views of the reconstructed point cloud and camera path (the red curve). Figures courtesy of David Capel and 2d3 (*`www.2d3.com`*).*

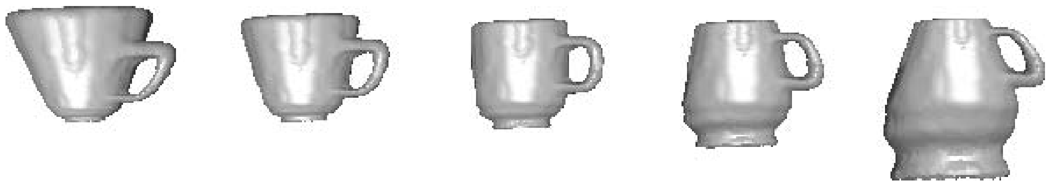


Fig. 1.4. **Projective ambiguity:** *Reconstructions of a mug (shown with the true shape in the centre) under 3D projective transformations in the* Z *direction. Five examples of the cup with different degrees of projective distortion are shown. The shapes are quite different from the original.*

(i) Compute the camera matrices of the three views $\mathtt{P}, \mathtt{P}', \mathtt{P}''$ from other point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$.
(ii) Triangulate the 3D point $\mathbf{X}$ from $\mathbf{x}$ and $\mathbf{x}'$ using $\mathtt{P}$ and $\mathtt{P}'$.
(iii) Project the 3D point into the third view as $\mathbf{x}'' = \mathtt{P}''\mathbf{X}$.

This procedure only requires projective information. An alternative procedure is to use the multi-view tensors (the fundamental matrix and trifocal tensor) to transfer the point directly without an explicit 3D reconstruction. Both methods have their advantages.

Suppose the camera rotates about its centre or that all the scene points of interest lie on a plane. Then the appropriate multiple view relations are the planar projective transformations between the images. In this case, a point seen in just one image can be transferred to any other image.

1.7 Euclidean reconstruction

So far we have considered the reconstruction of a scene, or transfer, for images taken with a set of uncalibrated cameras. For such cameras, important parameters such as the focal length, the geometric centre of the image (the principal point) and possibly the aspect ratio of the pixels in the image are unknown. If a complete calibration of each of the cameras is known then it is possible to remove some of the ambiguity of the reconstructed scene.

So far, we have discussed projective reconstruction, which is all that is possible without knowing something about the calibration of the cameras or the scene. Projective reconstruction is insufficient for many purposes, such as application to computer graphics, since it involves distortions of the model that appear strange to a human used to viewing a Euclidean world. For instance, the distortions that projective transformations induce in a simple object are shown in figure 1.4. Using the technique of projective reconstruction, there is no way to choose between any of the possible shapes of the mug in figure 1.4, and a projective reconstruction algorithm is as likely to come up with any one of the reconstructions shown there as any other. Even more severely distorted models may arise from projective reconstruction.

In order to obtain a reconstruction of the model in which objects have their correct (Euclidean) shape, it is necessary to determine the calibration of the cameras. It is easy to see that this is sufficient to determine the Euclidean structure of the scene. As we have seen, determining the Euclidean structure of the world is equivalent to specifying the plane at infinity and the absolute conic. In fact, since the absolute conic lies in a plane, the plane at infinity, it is enough to find the absolute conic in space. Now, suppose that we have computed a projective reconstruction of the world, using calibrated cameras. By definition, this means that the IAC is known in each of the images; let it be denoted by $\boldsymbol{\omega}_i$ in the i-th image. The back-projection of each $\boldsymbol{\omega}_i$ is a cone in space, and the absolute conic must lie in the intersection of all the cones. Two cones in general intersect in a fourth-degree curve, but given that they must intersect in a conic, this curve must split into two conics. Thus, reconstruction of the absolute conic from two images is not unique – rather, there are two possible solutions in general. However, from three or more images, the intersection of the cones is unique in general. Thus the absolute conic is determined and with it the Euclidean structure of the scene.

Of course, if the Euclidean structure of the scene is known, then so is the position of the absolute conic. In this case we may project it back into each of the images, producing the IAC in each image, and hence calibrating the cameras. Thus knowledge of the camera calibration is equivalent to being able to determine the Euclidean structure of the scene.

1.8 Auto-calibration

Without any knowledge of the calibration of the cameras, it is impossible to do better than projective reconstruction. There is no information in a set of feature correspondences across any number of views that can help us find the image of the absolute conic, or equivalently the calibration of the cameras. However, if we know just a little about the calibration of the cameras then we may be able to determine the position of the absolute conic.

Suppose, for instance that it is known that the calibration is the same for each of the cameras used in reconstructing a scene from an image sequence. By this we mean the following. In each image a coordinate system is defined, in which we have measured the image coordinates of corresponding features used to do projective reconstruction. Suppose that in all these image coordinate systems, the IAC is the same, but just where it is located is unknown. From this knowledge, we wish to compute the position of the absolute conic.

One way to find the absolute conic is to hypothesize the position of the IAC in one image; by hypothesis, its position in the other images will be the same. The back-projection of each of the conics will be a cone in space. If the three cones all meet in a single conic, then this must be a possible solution for the position of the absolute conic, consistent with the reconstruction.

Note that this is a conceptual description only. The IAC is of course a conic containing only complex points, and its back-projection will be a complex cone. However, algebraically, the problem is more tractable. Although it is complex, the IAC may be described by a real quadratic form (represented by a real symmetric matrix). The back-projected cone is also represented by a real quadratic form. For some value of the IAC, the three back-projected cones will meet in a conic curve in space.

Generally given three cameras known to have the same calibration, it is possible to determine the absolute conic, and hence the calibration of the cameras. However, although various methods have been proposed for this, it remains quite a difficult problem.

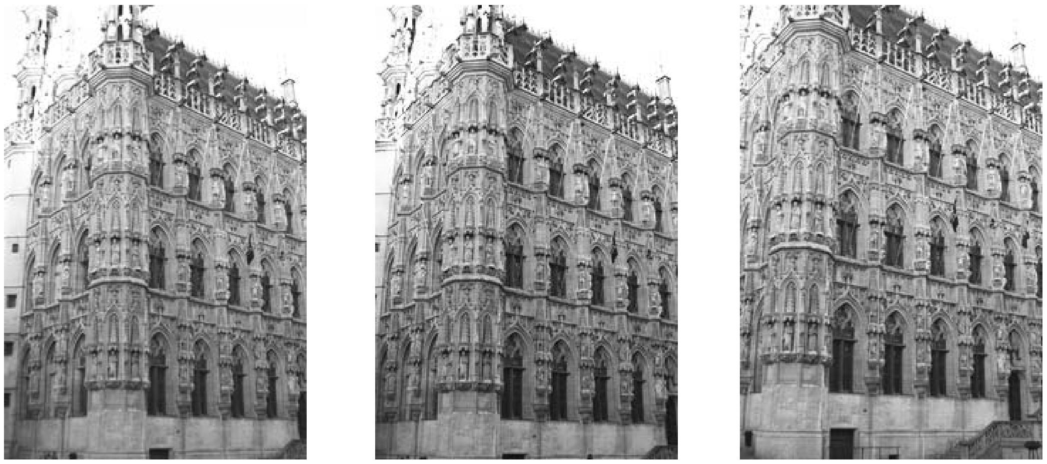
Knowing the plane at infinity. One method of auto-calibration is to proceed in steps by first determining the plane on which it lies. This is equivalent to identifying the plane at infinity in the world, and hence to determining the affine geometry of the world. In a second step, one locates the position of the absolute conic on the plane to determine the Euclidean geometry of space. Assuming one knows the plane at infinity, one can back-project a hypothesised IAC from each of a sequence of images and intersect the resulting cones with the plane at infinity. If the IAC is chosen correctly, the intersection curve is the absolute conic. Thus, from each pair of images one has a condition that the back-projected cones meet in the same conic curve on the plane at infinity. It turns out that this gives a linear constraint on the entries of the matrix representing the IAC. From a set of linear equations, one can determine the IAC, and hence the absolute conic. Thus, auto-calibration is relatively simple, once the plane at infinity has been identified. The identification of the plane at infinity itself is substantially more difficult.

Auto-calibration given square pixels in the image. If the cameras are partially calibrated, then it is possible to complete the calibration starting from a projective reconstruction. One can make do with quite minimal conditions on the calibration of the cameras, represented by the IAC. One interesting example is the square-pixel constraint on the cameras. What this means is that a Euclidean coordinate system is known in each image. In this case, the absolute conic, lying in the plane at infinity in the world must meet the image plane in its two circular points. The circular points in a plane are the two points where the absolute conic meets that plane. The back-projected rays through the circular points of the image plane must intersect the absolute conic. Thus, each image with square pixels determines two rays that must meet the absolute conic. Given n images, the autocalibration task then becomes that of determining a space conic (the absolute conic) that meets a set of $2n$ rays in space. An equivalent geometric picture is to intersect the set of rays with a plane and require that the set of intersection points lie on a conic. By a simple counting argument one may see that there are only a finite number of conics that meet eight prescribed rays in space. Therefore, from four images one may determine the calibration, albeit up to a finite number of possibilities.

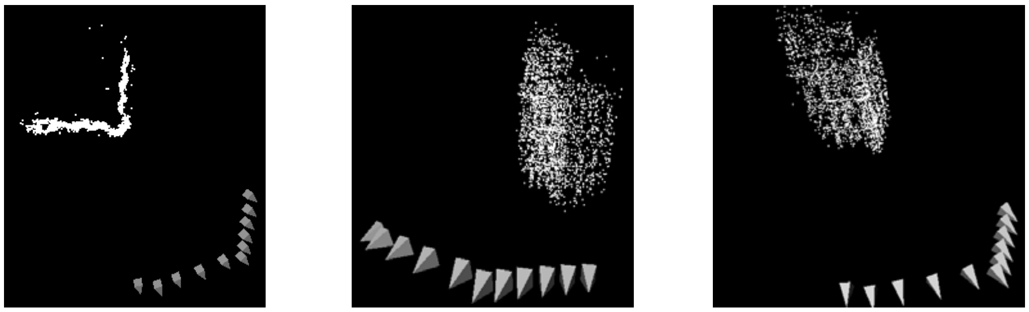
1.9 The reward I : 3D graphical models

We have now described all the ingredients necessary to compute realistic graphics models from image sequences. From point matches between images, it is possible to carry out first a projective reconstruction of the point set, and determine the motion of the camera in the chosen projective coordinate frame.

Using auto-calibration techniques, assuming some restrictions on the calibration of the camera that captured the image sequence, the camera may be calibrated, and the scene subsequently transformed to its true Euclidean structure.



a



b

Fig. 1.5. *(a) Three high resolution images (*3000×2000 *pixels) from a set of eleven of the cityhall in Leuven, Belgium. (b) Three views of a Euclidean reconstruction computed from the image set showing the 11 camera positions and point cloud.*

Knowing the projective structure of the scene, it is possible to find the epipolar geometry relating pairs of images and this restricts the correspondence search for further matches to a line – a point in one image defines a line in the other image on which the (as yet unknown) corresponding point must lie. In fact for suitable scenes, it is possible to carry out a dense point match between images and create a dense 3D model of the imaged scene. This takes the form of a triangulated shape model that is subsequently shaded or texture-mapped from the supplied images and used to generate novel views. The steps of this process are illustrated in figure 1.5 and figure 1.6.

1.10 The reward II: video augmentation

We finish this introduction with a further application of reconstruction methods to computer graphics. Automatic reconstruction techniques have recently become widely used in the film industry as a means for adding artificial graphics objects in real video sequences. Computer analysis of the motion of the camera is replacing the previously used manual methods for correctly aligning the artificial inserted object.

The most important requirement for realistic insertion of an artificial object in a video

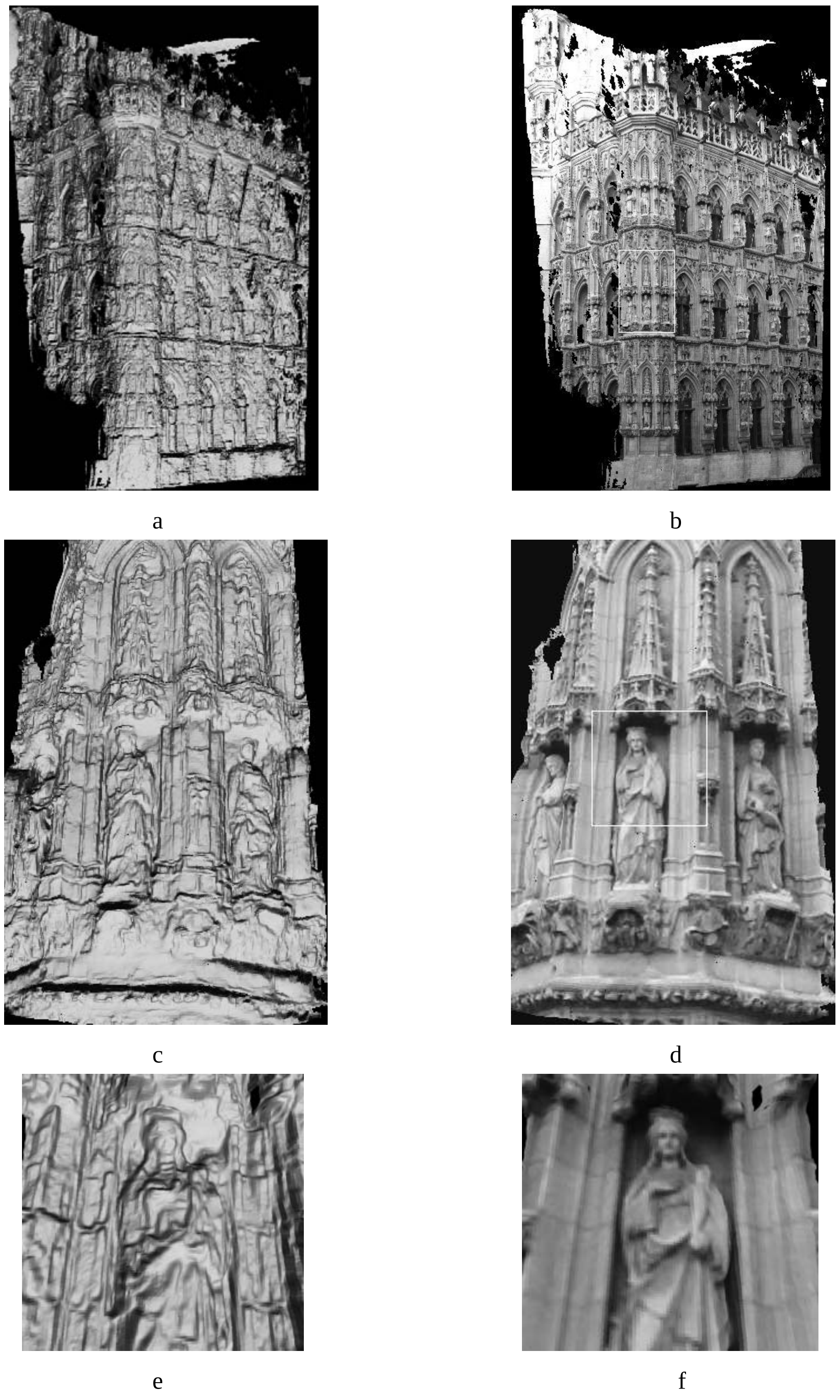


Fig. 1.6. **Dense reconstructions.** *These are computed from the cameras and image of figure 1.5. (a) Untextured and (b) textured reconstruction of the full scene. (c) Untextured and (d) textured close up of the area shown in the white rectangle of (b). (e) Untextured and (f) textured close up of the area shown in the white rectangle of (d). The dense surface is computed using the three-view stereo algorithm described in [Strecha-02]. Figures courtesy of Christoph Strecha, Frank Verbiest, and Luc Van Gool.*

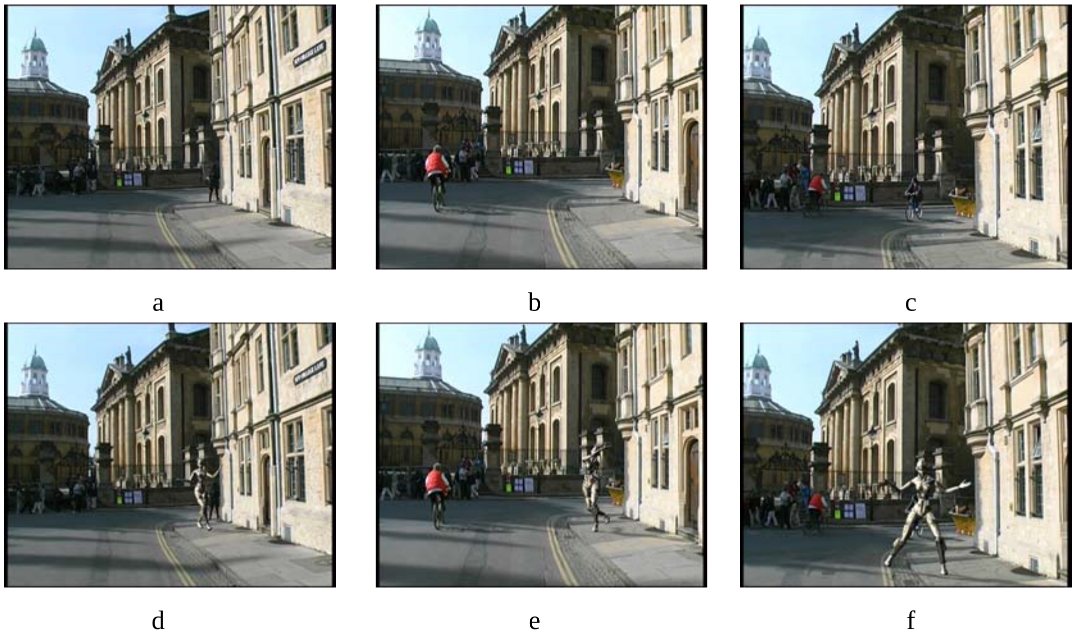


Fig. 1.7. **Augmented video.** *The animated robot is inserted into the scene and rendered using the computed cameras of figure 1.3. (a)-(c) Original frames from the sequence. (d)-(f) The augmented frames. Figures courtesy of 2d3 (*`www.2d3.com`*).*

sequence is to compute the correct motion of the camera. Unless the camera motion is correctly determined, it is impossible to generate the correct sequences of views of the graphics model in a way that will appear consistent with the background video. Generally, it is only the motion of the camera that is important here; we do not need to reconstruct the scene, since it is already present in the existing video, and novel views of the scene visible in the video are not required. The only requirement is to be able to generate correct perspective views of the graphics model.

It is essential to compute the motion of the camera in a Euclidean frame. It is not enough merely to know the projective motion of the camera. This is because a Euclidean object is to be placed in the scene. Unless this graphics object and the cameras are known in the same coordinate frame, then generated views of the inserted object will be seen to distort with respect to the perceived structure of the scene seen in the existing video.

Once the correct motion of the camera, and its calibration are known the inserted object may be rendered into the scene in a realistic manner. If the change of the camera calibration from frame to frame is correctly determined, then the camera may change focal length (zoom) during the sequence. It is even possible for the principal point to vary during the sequence through cropping.

In inserting the rendered model into the video, the task is relatively straight-forward if it lies in front of all the existing scene. Otherwise the possibility of occlusions arises, in which the scene may obscure parts of the model. An example of video augmentation is shown in figure 1.7.