Foundations and Trends® in
Computer Graphics and Vision
Vol. 4, No. 4 (2008) 287–398

DOI: 10.1561/0600000007

now
the essence of knowledge

3D Reconstruction from Multiple Images Part 1: Principles

By Theo Moons, Luc Van Gool, and Maarten Vergauwen

Contents

Foundations and Trends® in
Computer Graphics and Vision
Vol. 4, No. 4 (2008) 287–398

DOI: 10.1561/0600000007

now
the essence of knowledge

3D Reconstruction from Multiple Images Part 1: Principles

Theo Moons[1], Luc Van Gool[2,3], and Maarten Vergauwen[4]

[1] *Hogeschool — Universiteit Brussel, Stormstraat 2, Brussel, 1000, Belgium, Theo.Moons@hubrussel.be*
[2] *Katholieke Universiteit Leuven, ESAT — PSI, Kasteelpark Arenberg 10, Leuven, B-3001, Belgium, Luc.VanGool@esat.kuleuven.be*
[3] *ETH Zurich, BIWI, Sternwartstrasse 7, Zurich, CH-8092, Switzerland, vangool@vision.ee.ethz.ch*
[4] *GeoAutomation NV, Karel Van Lotharingenstraat 2, Leuven, B-3000, Belgium, maarten.vergauwen@geoautomation.com*

Abstract

This issue discusses methods to extract three-dimensional (3D) models from plain images. In particular, the 3D information is obtained from images for which the camera parameters are unknown. The principles underlying such uncalibrated structure-from-motion methods are outlined. First, a short review of 3D acquisition technologies puts such methods in a wider context and highlights their important advantages. Then, the actual theory behind this line of research is given. The authors have tried to keep the text maximally self-contained, therefore also avoiding to rely on an extensive knowledge of the projective concepts that usually appear in texts about self-calibration 3D methods. Rather, mathematical explanations that are more amenable to intuition are given. The explanation of the theory includes the stratification

of reconstructions obtained from image pairs as well as metric reconstruction on the basis of more than two images combined with some additional knowledge about the cameras used. Readers who want to obtain more practical information about how to implement such uncalibrated structure-from-motion pipelines may be interested in two more Foundations and Trends issues written by the same authors. Together with this issue they can be read as a single tutorial on the subject.

Preface

Welcome to this Foundations and Trends tutorial on three-dimensional (3D) reconstruction from multiple images. The focus is on the creation of 3D models from nothing but a set of images, taken from unknown camera positions and with unknown camera settings. In this issue, the underlying theory for such "self-calibrating" 3D reconstruction methods is discussed. Of course, the text cannot give a complete overview of all aspects that are relevant. That would mean dragging in lengthy discussions on feature extraction, feature matching, tracking, texture blending, dense correspondence search, etc. Nonetheless, we tried to keep at least the geometric aspects of the self-calibration reasonably self-contained and this is where the focus lies.

The issue consists of two main parts, organized in separate sections. Section 1 places the subject of self-calibrating 3D reconstruction from images in the wider context of 3D acquisition techniques. This section thus also gives a short overview of alternative 3D reconstruction techniques, as the uncalibrated structure-from-motion approach is not necessarily the most appropriate one for all applications. This helps to bring out the pros and cons of this particular approach.

Section 2 starts the actual discussion of the topic. With images as our key input for 3D reconstruction, this section first discusses how we can mathematically model the process of image formation by a camera, and which parameters are involved. Equipped with that camera model, it then discusses the process of self-calibration for multiple cameras from a theoretical perspective. It deals with the core issues of this tutorial: given images and incomplete knowledge about the cameras, what can we still retrieve in terms of 3D scene structure and how can we make up for the missing information. This section also describes cases in between fully calibrated and uncalibrated reconstruction. Breaking a bit with tradition, we have tried to describe the whole self-calibration process in intuitive, Euclidean terms. We have avoided the usual explanation via projective concepts, as we believe that entities like the dual of the projection of the absolute quadric are not very amenable to intuition.

Readers who are interested in implementation issues and a practical example of a self-calibrating 3D reconstruction pipeline may be interested in two complementary, upcoming issues by the same authors, which together with this issue can be read as a single tutorial.

1

Introduction to 3D Acquisition

This section discusses different methods for capturing or 'acquiring' the three-dimensional (3D) shape of surfaces and, in some cases, also the distance or 'range' of the object to the 3D acquisition device. The section aims at positioning the methods discussed in this text within this more global context. This will make clear that alternative methods may actually be better suited for some applications that need 3D. This said, the discussion will also show that the kind of approach described here is one of the more flexible and powerful ones.

1.1 A Taxonomy of Methods

A 3D acquisition taxonomy is given in Figure 1.1. A first distinction is between *active* and *passive methods*. With active techniques the light sources are specially controlled, as part of the strategy to arrive at the 3D information. Active lighting incorporates some form of temporal or spatial modulation of the illumination. With passive techniques, on the other hand, light is not controlled or only with respect to image quality. Typically passive techniques work with whichever reasonable, ambient light available. From a computational point of view, active methods

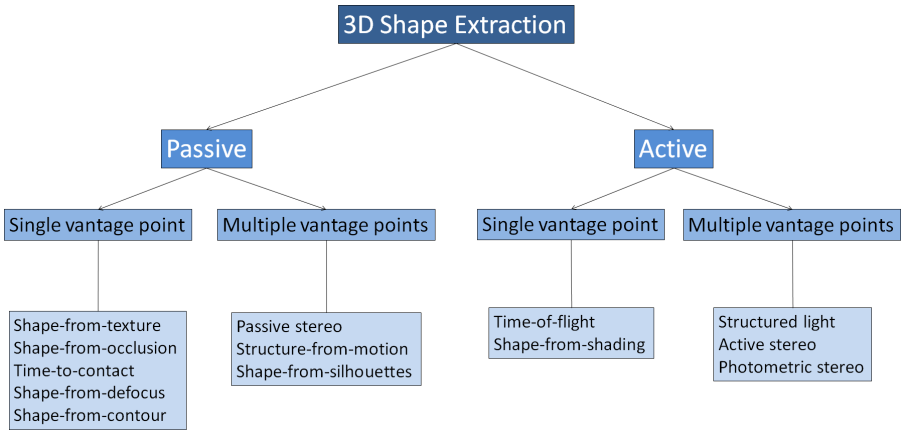

Fig. 1.1 Taxonomy of methods for the extraction of information on 3D shape.

tend to be less demanding, as the special illumination is used to simplify some of the steps in the 3D capturing process. Their applicability is restricted to environments where the special illumination techniques can be applied.

A second distinction is between the number of vantage points from where the scene is observed and/or illuminated. With *single-vantage methods* the system works from a single vantage point. In case there are multiple viewing or illumination components, these are positioned very close to each other, and ideally they would coincide. The latter can sometimes be realized virtually, through optical means like semi-transparent mirrors. With *multi-vantage systems*, several viewpoints and/or controlled illumination source positions are involved. For multi-vantage systems to work well, the different components often have to be positioned far enough from each other. One says that the 'baseline' between the components has to be wide enough. Single-vantage methods have as advantages that they can be made compact and that they do not suffer from the occlusion problems that occur when parts of the scene are not visible from all vantage points in multi-vantage systems.

The methods mentioned in the taxonomy will now be discussed in a bit more detail. In the remaining sections, we then continue with the more elaborate discussion of passive, multi-vantage structure-from-motion (SfM) techniques, the actual subject of this tutorial. As this

overview of 3D acquisition methods is not intended to be in-depth nor exhaustive, just to provide a bit of context for our further image-based 3D reconstruction from uncalibrated images account, we don't include references in this part.

1.2 Passive Triangulation

Several multi-vantage approaches use the principle of *triangulation* for the extraction of depth information. This also is the key concept exploited by the self-calibrating structure-from-motion (SfM) methods described in this tutorial.

1.2.1 (Passive) Stereo

Suppose we have two images, taken at the same time and from different viewpoints. Such setting is referred to as *stereo*. The situation is illustrated in Figure 1.2. The principle behind stereo-based 3D reconstruction is simple: given the two projections of the same point in the world onto the two images, its 3D position is found as the intersection of the two projection rays. Repeating such process for several points

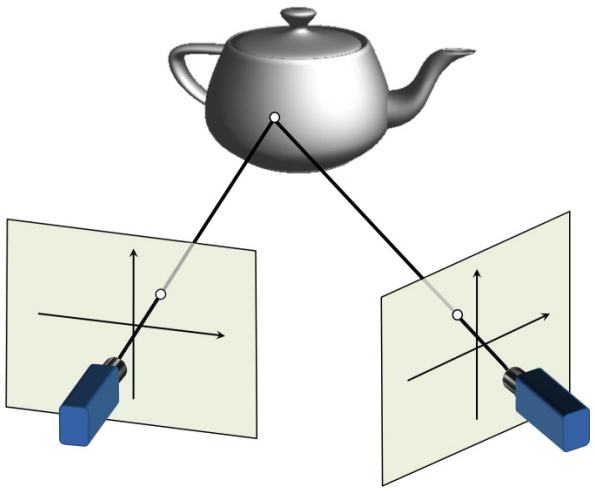

Fig. 1.2 The principle behind stereo-based 3D reconstruction is very simple: given two images of a point, the point's position in space is found as the intersection of the two projection rays. This procedure is referred to a 'triangulation'.

yields the 3D shape and configuration of the objects in the scene. Note that this construction — referred to as *triangulation* — requires the equations of the rays and, hence, complete knowledge of the cameras: their (relative) positions and orientations, but also their settings like the focal length. These camera parameters will be discussed in Section 2. The process to determine these parameters is called *(camera) calibration*.

Moreover, in order to perform this triangulation process, one needs ways of solving the correspondence problem, i.e., finding the point in the second image that corresponds to a specific point in the first image, or vice versa. Correspondence search actually is the hardest part of stereo, and one would typically have to solve it for many points. Often the correspondence problem is solved in two stages. First, correspondences are sought for those points for which this is easiest. Then, correspondences are sought for the remaining points. This will be explained in more detail in subsequent sections.

1.2.2 Structure-from-Motion

Passive stereo uses two cameras, usually synchronized. If the scene is static, the two images could also be taken by placing the same camera at the two positions, and taking the images in sequence. Clearly, once such strategy is considered, one may just as well take more than two images, while moving the camera. Such strategies are referred to as structure-from-motion or SfM for short. If images are taken over short time intervals, it will be easier to find correspondences, e.g., by tracking feature points over time. Moreover, having more camera views will yield object models that are more complete. Last but not least, if multiple views are available, the camera(s) need no longer be calibrated beforehand, and a self-calibration procedure may be employed instead. Self-calibration means that the internal and external camera parameters (see next section) are extracted from images of the unmodified scene itself, and not from images of dedicated calibration patterns. These properties render SfM a very attractive 3D acquisition strategy. A more detailed discussion is given in the following sections.

1.3 Active Triangulation

Finding corresponding points can be facilitated by replacing one of the cameras in a stereo setup by a projection device. Hence, we combine one illumination source with one camera. For instance, one can project a spot onto the object surface with a laser. The spot will be easily detectable in the image taken by the camera. If we know the position and orientation of both the laser ray and the camera projection ray, then the 3D surface point is again found as their intersection. The principle is illustrated in Figure 1.3 and is just another example of the triangulation principle.

The problem is that knowledge about the 3D coordinates of one point is hardly sufficient in most applications. Hence, in the case of the laser, it should be directed at different points on the surface and each time an image has to be taken. In this way, the 3D coordinates of these points are extracted, one point at a time. Such a 'scanning'

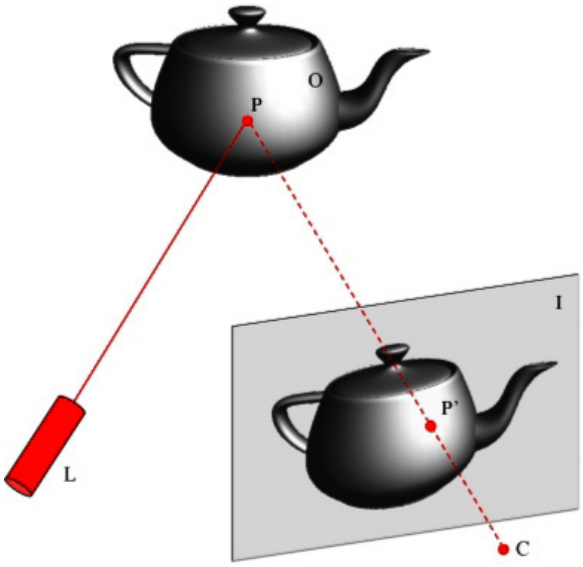

Fig. 1.3 The triangulation principle used already with stereo, can also be used in an active configuration. The laser L projects a ray of light onto the object O. The intersection point P with the object is viewed by a camera and forms the spot P' on its image plane I. This information suffices for the computation of the three-dimensional coordinates of P, assuming that the laser-camera configuration is known.

process requires precise mechanical apparatus (e.g., by steering rotating mirrors that reflect the laser light into controlled directions). If the equations of the laser rays are not known precisely, the resulting 3D coordinates will be imprecise as well. One would also not want the system to take a long time for scanning. Hence, one ends up with the conflicting requirements of guiding the laser spot precisely *and* fast. These challenging requirements have an adverse effect on the price. Moreover, the times needed to take one image per projected laser spot add up to seconds or even minutes of overall acquisition time. A way out is using special, super-fast imagers, but again at an additional cost.

In order to remedy this problem, substantial research has gone into replacing the laser spot by more complicated patterns. For instance, the laser ray can without much difficulty be extended to a plane, e.g., by putting a cylindrical lens in front of the laser. Rather than forming a single laser spot on the surface, the intersection of the plane with the surface will form a curve. The configuration is depicted in Figure 1.4. The 3D coordinates of each of the points along the intersection curve

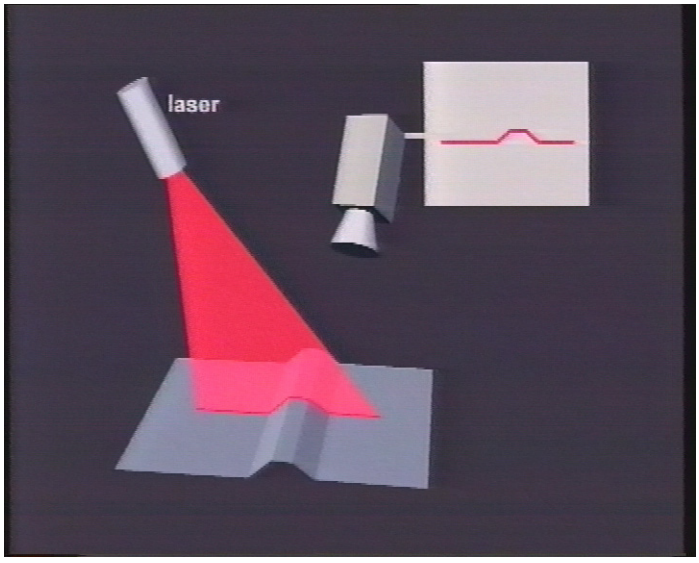

Fig. 1.4 If the active triangulation configuration is altered by turning the laser spot into a line (e.g., by the use of a cylindrical lens), then scanning can be restricted to a one-directional motion, transversal to the line.

can be determined again through triangulation, namely as the intersection of the plane with the viewing ray for that point. This still yields a unique point in space. From a single image, many 3D points can be extracted in this way. Moreover, the two-dimensional scanning motion as required with the laser spot can be replaced by a much simpler one-dimensional sweep over the surface with the laser plane.

It now stands to reason to try and eliminate any scanning altogether. Is it not possible to directly go for a dense distribution of points all over the surface? Unfortunately, extensions to the two-dimensional projection patterns that are required are less straightforward. For instance, when projecting multiple parallel lines of light simultaneously, a camera viewing ray will no longer have a single intersection with such a pencil of illumination planes. We would have to include some kind of code into the pattern to make a distinction between the different lines in the pattern and the corresponding projection planes. Note that counting lines have their limitations in the presence of depth discontinuities and image noise. There are different ways of including a code. An obvious one is to give the lines different colors, but interference by the surface colors may make it difficult to identify a large number of lines in this way. Alternatively, one can project several stripe patterns in sequence, giving up on using a single projection but still only using a few. Figure 1.5 gives a (non-optimal) example of binary patterns. The sequence of being bright or dark forms a unique binary code for each column in the projector. Although one could project different shades of gray, using binary (i.e., all-or-nothing black or white) type of codes

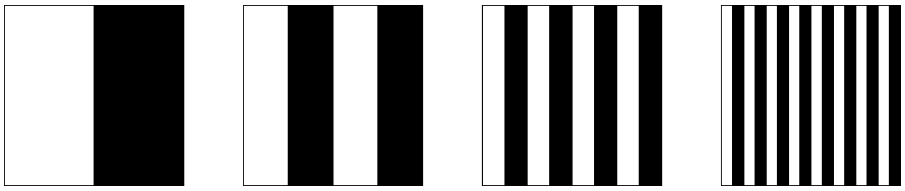

Fig. 1.5 Series of masks that can be projected for active stereo applications. Subsequent masks contain ever finer stripes. Each of the masks is projected and for a point in the scene the sequence of black/white values is recorded. The subsequent bits obtained that way characterize the horizontal position of the points, i.e., the plane of intersection (see text). The resolution that is required (related to the width of the thinnest stripes) imposes the number of such masks that has to be used.

is beneficial for robustness. Nonetheless, so-called phase shift methods successfully use a set of patterns with sinusoidally varying intensities in one direction and constant intensity in the perpendicular direction (i.e., a more gradual stripe pattern than in the previous example). Each of the three sinusoidal patterns has the same amplitude but is 120° phase shifted with respect to each other. Intensity ratios in the images taken under each of the three patterns yield a unique position modulo the periodicity of the patterns. The sine patterns sum up to a constant intensity, so adding the three images yields the scene texture. The three subsequent projections yield dense range values plus texture. An example result is shown in Figure 1.6. These 3D measurements have been obtained with a system that works in real time (30 Hz depth + texture).

One can also design more intricate patterns that contain local spatial codes to identify parts of the projection pattern. An example is shown in Figure 1.7. The figure shows a face on which the single, checkerboard kind of pattern on the left is projected. The pattern is such that each column has its own distinctive signature. It consists of combinations of little white or black squares at the vertices of the checkerboard squares. 3D reconstructions obtained with this technique are shown in Figure 1.8. The use of this pattern only requires the acquisition of a single image. Hence, continuous projection in combi-

Fig. 1.6 3D results obtained with a phase-shift system. *Left*: 3D reconstruction without texture. *Right*: same with texture, obtained by summing the three images acquired with the phase-shifted sine projections.

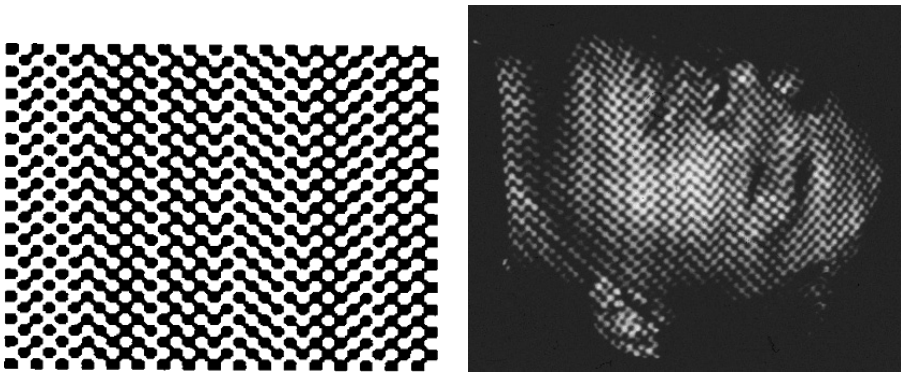

Fig. 1.7 Example of one-shot active range technique. *Left*: The projection pattern allowing disambiguation of its different vertical columns. *Right*: The pattern is projected on a face.

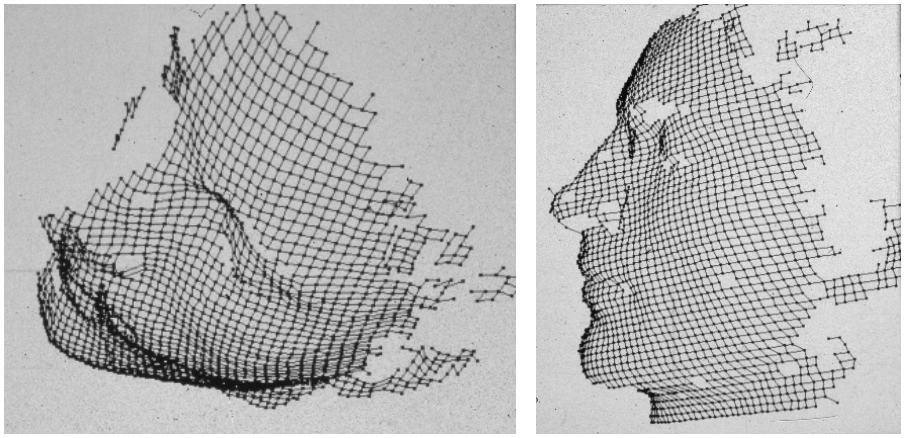

Fig. 1.8 Two views of the 3D description obtained with the active method of Figure 1.7.

nation with video input yields a 4D acquisition device that can capture 3D shape (but not texture) and its changes over time. All these approaches with specially shaped projected patterns are commonly referred to as *structured light* techniques.

1.4 Other Methods

With the exception of time-of-flight techniques, all other methods in the taxonomy of Figure 1.1 are of less practical importance (yet). Hence,

only time-of-flight is discussed to a somewhat greater length. For the other approaches, only their general principles are outlined.

1.4.1 Time-of-Flight

The basic principle of time-of-flight sensors is the measurement of the duration before a sent out time-modulated signal — usually light from a laser — returns to the sensor. This time is proportional to the distance from the object. This is an active, single-vantage approach. Depending on the type of waves used, one calls such devices *radar* (electromagnetic waves of low frequency), *sonar* (acoustic waves), or *optical radar* (optical electromagnetic waves, including near-infrared).

A first category uses pulsed waves and measures the delay between the transmitted and the received pulse. These are the most often used type. A second category is used for smaller distances and measures phase shifts between outgoing and returning sinusoidal waves. The low level of the returning signal and the high bandwidth required for detection put pressure on the signal to noise ratios that can be achieved. Measurement problems and health hazards with lasers can be alleviated by the use of ultrasound. The bundle has a much larger opening angle then, and resolution decreases (a lot).

Mainly optical signal-based systems (typically working in the near-infrared) represent serious competition for the methods mentioned before. Such systems are often referred to as LIDAR (LIght Detection And Ranging) or LADAR (LAser Detection And Ranging, a term more often used by the military, where wavelengths tend to be longer, like 1,550 nm in order to be invisible in night goggles). As these systems capture 3D data point-by-point, they need to scan. Typically a horizontal motion of the scanning head is combined with a faster vertical flip of an internal mirror. Scanning can be a rather slow process, even if at the time of writing there were already LIDAR systems on the market that can measure 50,000 points per second. On the other hand, LIDAR gives excellent precision at larger distances in comparison to passive techniques, which start to suffer from limitations in image resolution. Typically, errors at tens of meters will be within a range of a few centimeters. Triangulation-based techniques require quite some baseline to achieve such small margins. A disadvantage is that surface

texture is not captured and that errors will be substantially larger for dark surfaces, which reflect little of the incoming signal. Missing texture can be resolved by adding a camera, as close as possible to the LIDAR scanning head. But of course, even then the texture is not taken from exactly the same vantage point. The output is typically delivered as a massive, unordered point cloud, which may cause problems for further processing. Moreover, LIDAR systems tend to be expensive.

More recently, 3D cameras have entered the market, that use the same kind of time-of-flight principle, but that acquire an entire 3D image at the same time. These cameras have been designed to yield real-time 3D measurements of smaller scenes, typically up to a couple of meters. So far, resolutions are still limited (in the order of 150×150 range values) and depth resolutions only moderate (couple of millimeters under ideal circumstances but worse otherwise), but this technology is making advances fast. It is expected that this price will drop sharply soon, as some games console manufacturer's plan to offer such cameras as input devices.

1.4.2 Shape-from-Shading and Photometric Stereo

We now discuss the remaining, active techniques in the taxonomy of Figure 1.1.

'Shape-from-shading' techniques typically handle smooth, untextured surfaces. Without the use of structured light or time-of-flight methods these are difficult to handle. Passive methods like stereo may find it difficult to extract the necessary correspondences. Yet, people can estimate the overall shape quite well (qualitatively), even from a single image and under uncontrolled lighting. This would win it a place among the passive methods. No computer algorithm today can achieve such performance, however. Yet, progress has been made under simplifying conditions. One can use directional lighting with known direction and intensity. Hence, we have placed the method in the 'active' family for now. Gray levels of object surface patches then convey information on their 3D orientation. This process not only requires information on the sensor-illumination configuration, but also on the reflection characteristics of the surface. The complex relationship between gray levels

and surface orientation can theoretically be calculated in some cases — e.g., when the surface reflectance is known to be Lambertian — but is usually derived from experiments and then stored in 'reflectance maps' for table-lookup. For a Lambertian surface with known albedo and for a known light source intensity, the angle between the surface normal and the incident light direction can be derived. This yields surface normals that lie on a cone about the light direction. Hence, even in this simple case, the normal of a patch cannot be derived uniquely from its intensity. Therefore, information from different patches is combined through extra assumptions on surface smoothness. Neighboring patches can be expected to have similar normals. Moreover, for a smooth surface the normals at the visible rim of the object can be determined from their tangents in the image if the camera settings are known. Indeed, the 3D normals are perpendicular to the plane formed by the projection ray at these points and the local tangents to the boundary in the image. This yields strong boundary conditions. Estimating the lighting conditions is sometimes made part of the problem. This may be very useful, as in cases where the light source is the sun. The light is also not always assumed to be coming from a single direction. For instance, some lighting models consist of both a directional component and a homogeneous ambient component, where light is coming from all directions in equal amounts. Surface interreflections are a complication which these techniques so far cannot handle.

The need to combine normal information from different patches can be reduced by using different light sources with different positions. The light sources are activated one after the other. The subsequent observed intensities for the surface patches yield only a single possible normal orientation (not withstanding noise in the intensity measurements). For a Lambertian surface, three different lighting directions suffice to eliminate uncertainties about the normal direction. The three cones intersect in a single line, which is the sought patch normal. Of course, it still is a good idea to further improve the results, e.g., via smoothness assumptions. Such 'photometric stereo' approach is more stable than shape-from-shading, but it requires a more controlled acquisition environment. An example is shown in Figure 1.9. It shows a dome with 260 LEDs that is easy to assemble and disassemble (modular design, fitting

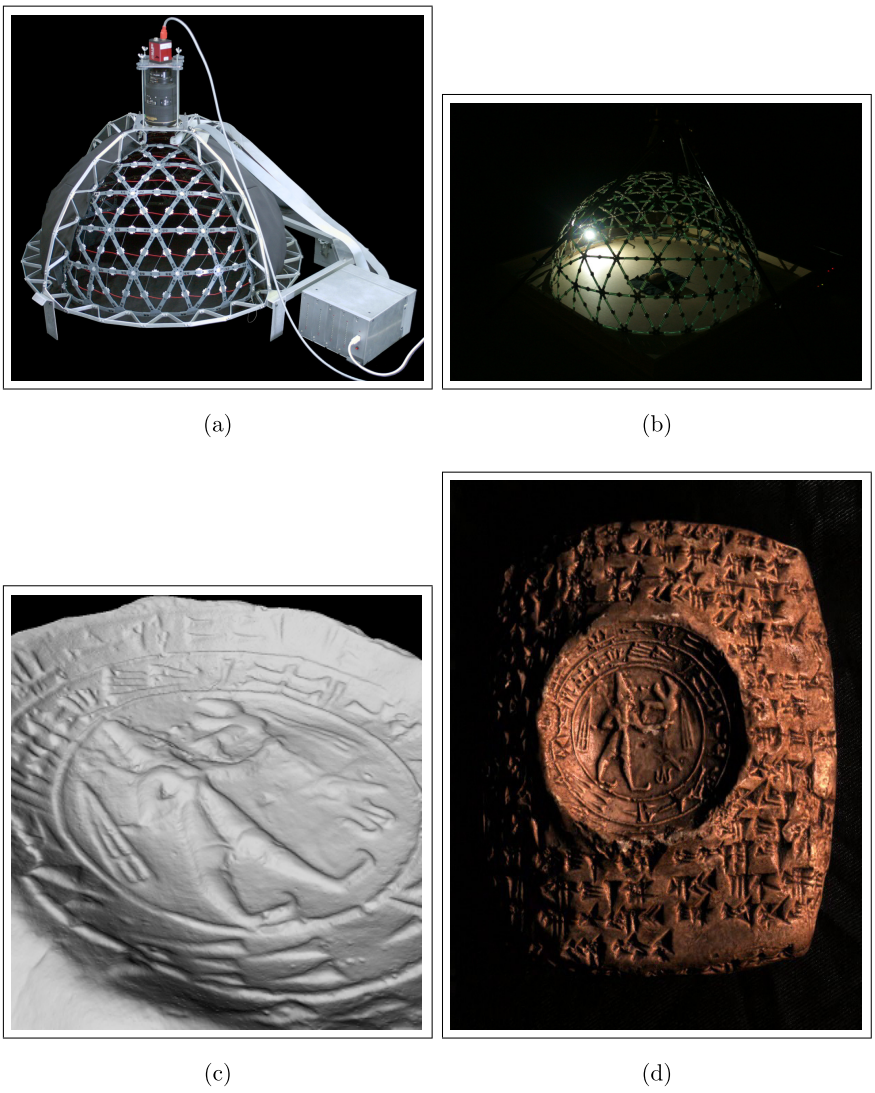

Fig. 1.9 (a) Mini-dome with different LED light sources, (b) scene with one of the LEDs activated, (c) 3D reconstruction of cuneiform tablet, without texture, and (d) same tablet with texture.

in a standard aircraft suitcase; see part (a) of the figure). The LEDs are automatically activated in a predefined sequence. There is one overhead camera. The resulting 3D reconstruction of a cuneiform tablet is shown in Figure 1.9(c) without texture, and in (d) with texture.

As with structured light techniques, one can try to reduce the number of images that have to be taken, by giving the light sources different colors. The resulting mix of colors at a surface patch yields direct information about the surface normal. In case 3 projections suffice, one can exploit the R-G-B channels of a normal color camera. It is like taking three intensity images in parallel, one per spectral band of the camera.

Note that none of the above techniques yield absolute depths, but rather surface normal directions. These can be integrated into full 3D models of shapes.

1.4.3 Shape-from-Texture and Shape-from-Contour

Passive single vantage methods include shape-from-texture and shape-from-contour. These methods do not yield true range data, but, as in the case of shape-from-shading, only surface orientation.

Shape-from-texture assumes that a surface is covered by a homogeneous texture (i.e., a surface pattern with some statistical or geometric regularity). Local inhomogeneities of the imaged texture (e.g., anisotropy in the statistics of edge orientations for an isotropic texture, or deviations from assumed periodicity) are regarded as the result of projection. Surface orientations which allow the original texture to be maximally isotropic or periodic are selected. Figure 1.10 shows an

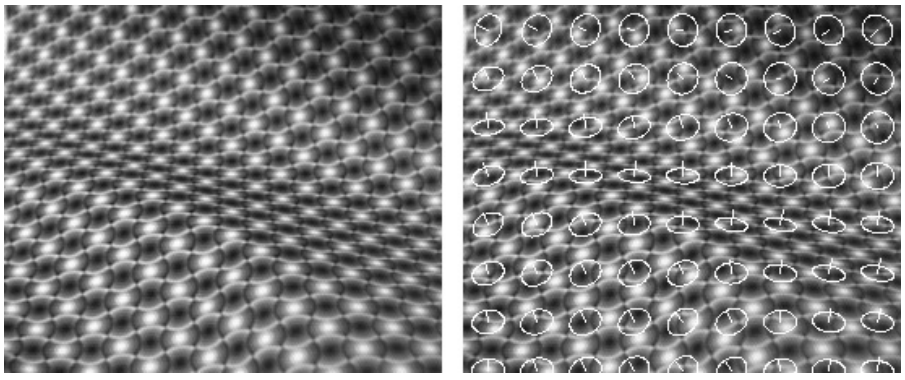

Fig. 1.10 *Left*: The regular texture yields a clear perception of a curved surface. *Right*: the result of a shape-from-texture algorithm.

example of a textured scene. The impression of an undulating surface is immediate. The right-hand side of the figure shows the results for a shape-from-texture algorithm that uses the regularity of the pattern for the estimation of the local surface orientation. Actually, what is assumed here is a square shape of the pattern's period (i.e., a kind of discrete isotropy). This assumption suffices to calculate the local surface orientation. The ellipses represent circles with such calculated orientation of the local surface patch. The small stick at their center shows the computed normal to the surface.

Shape-from-contour makes similar assumptions about the true shape of, usually planar, objects. Observing an ellipse, the assumption can be made that it actually is a circle, and the slant and tilt angles of the plane can be determined. For instance, in the shape-from-texture figure we have visualized the local surface orientation via ellipses. This 3D impression is compelling, because we tend to interpret the elliptical shapes as projections of what in reality are circles. This is an example of shape-from-contour as applied by our brain. The circle–ellipse relation is just a particular example, and more general principles have been elaborated in the literature. An example is the maximization of area over perimeter squared, as a measure of shape compactness, over all possible deprojections, i.e., surface patch orientations. Returning to our example, an ellipse would be deprojected to a circle for this measure, consistent with human vision. Similarly, symmetries in the original shape will get lost under projection. Choosing the slant and tilt angles that maximally restore symmetry is another example of a criterion for determining the normal to the shape. As a matter of fact, the circle–ellipse case also is an illustration for this measure. Regular figures with at least a 3-fold rotational symmetry yield a single orientation that could make up for the deformation in the image, except for the mirror reversal with respect to the image plane (assuming that perspective distortions are too small to be picked up). This is but a special case of the more general result, that a unique orientation (up to mirror reflection) also results when two copies of a shape are observed in the same plane (with the exception where their orientation differs by 0° or 180° in which case nothing can be said on the mere assumption that both shapes are identical). Both cases are more restrictive than

skewed mirror symmetry (without perspective effects), which yields a one-parameter family of solutions only.

1.4.4 Shape-from-Defocus

Cameras have a limited depth-of-field. Only points at a particular distance will be imaged with a sharp projection in the image plane. Although often a nuisance, this effect can also be exploited because it yields information on the distance to the camera. The level of defocus has already been used to create depth maps. As points can be blurred because they are closer or farther from the camera than at the position of focus, shape-from-defocus methods will usually combine more than a single image, taken from the same position but with different focal lengths. This should disambiguate the depth.

1.4.5 Shape-from-Silhouettes

Shape-from-silhouettes is a passive, multi-vantage approach. Suppose that an object stands on a turntable. At regular rotational intervals an image is taken. In each of the images, the silhouette of the object is determined. Initially, one has a virtual lump of clay, larger than the object and fully containing it. From each camera orientation, the silhouette forms a cone of projection rays, for which the intersection with this virtual lump is calculated. The result of all these intersections yields an approximate shape, a so-called visual hull. Figure 1.11 illustrates the process.

One has to be careful that the silhouettes are extracted with good precision. A way to ease this process is by providing a simple background, like a homogeneous blue or green cloth ('blue keying' or 'green keying'). Once a part of the lump has been removed, it can never be retrieved in straightforward implementations of this idea. Therefore, more refined, probabilistic approaches have been proposed to fend off such dangers. Also, cavities that do not show up in any silhouette will not be removed. For instance, the eye sockets in a face will not be detected with such method and will remain filled up in the final model. This can be solved by also extracting stereo depth from neighboring

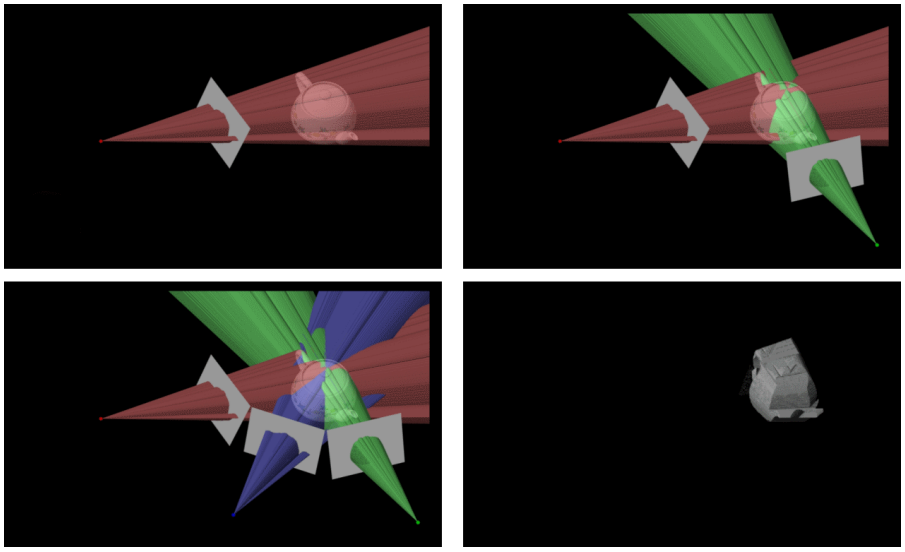

Fig. 1.11 The first three images show different backprojections from the silhouette of a teapot in three views. The intersection of these backprojections form the visual hull of the object, shown in the bottom right image. The more views are taken, the closer the visual hull approaches the true shape, but cavities not visible in the silhouettes are not retrieved.

viewpoints and by combining the 3D information coming from both methods.

The hardware needed is minimal, and very low-cost shape-from-silhouette systems can be produced. If multiple cameras are placed around the object, the images can be taken all at once and the capture time can be reduced. This will increase the price, and also the silhouette extraction may become more complicated. In the case video cameras are used, a dynamic scene like a moving person can be captured in 3D over time (but note that synchronization issues are introduced). An example is shown in Figure 1.12, where 15 video cameras were set up in an outdoor environment.

Of course, in order to extract precise cones for the intersection, the relative camera positions and their internal settings have to be known precisely. This can be achieved with the same self-calibration methods expounded in the following sections. Hence, also shape-from-silhouettes can benefit from the presented ideas and this is all the more interesting

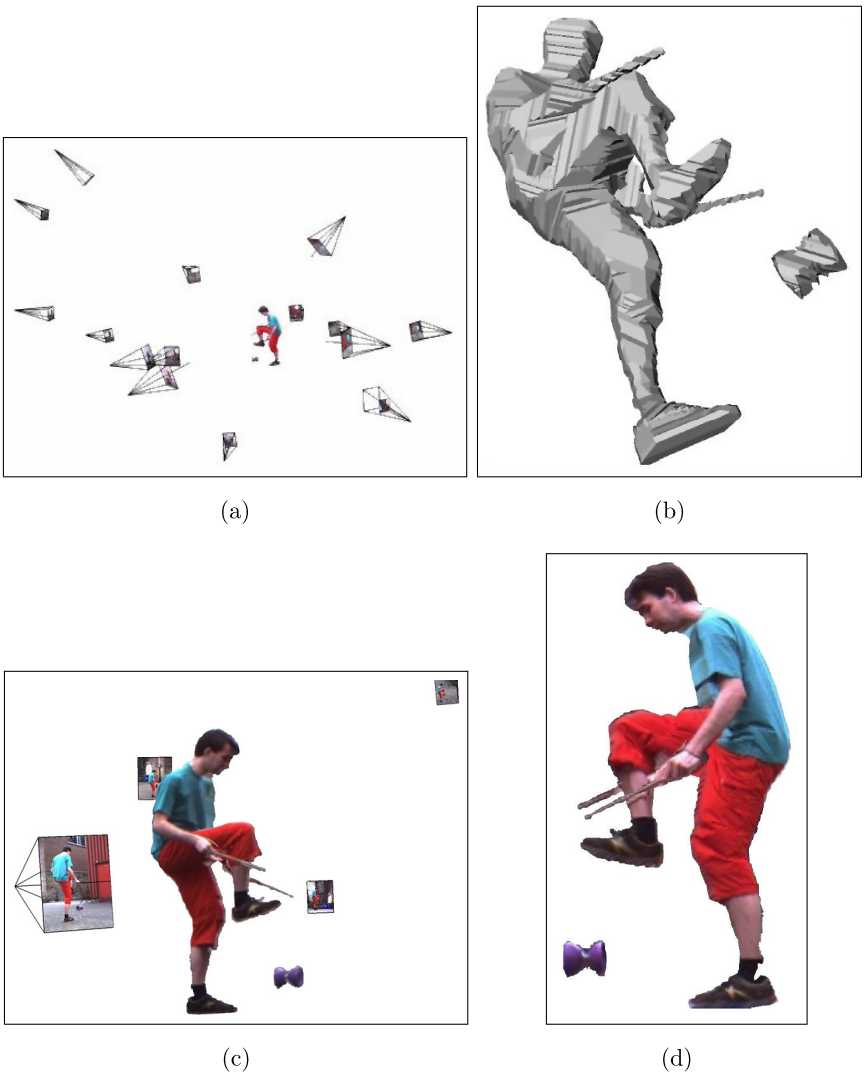

Fig. 1.12 (a) Fifteen cameras setup in an outdoor environment around a person,(b) a more detailed view on the visual hull at a specific moment of the action,(c) detailed view on the visual hull textured by backprojecting the image colors, and (d) another view of the visual hull with backprojected colors. Note how part of the sock area has been erroneously carved away.

as this 3D extraction approach is among the most practically relevant ones for dynamic scenes ('motion capture').

1.4.6 Hybrid Techniques

The aforementioned techniques often have complementary strengths and weaknesses. Therefore, several systems try to exploit multiple techniques in conjunction. A typical example is the combination of shape-from-silhouettes with stereo as already hinted in the previous section. Both techniques are passive and use multiple cameras. The visual hull produced from the silhouettes provides a depth range in which stereo can try to refine the surfaces in between the rims, in particular at the cavities. Similarly, one can combine stereo with structured light. Rather than trying to generate a depth map from the images pure, one can project a random noise pattern, to make sure that there is enough texture. As still two cameras are used, the projected pattern does not have to be analyzed in detail. Local pattern correlations may suffice to solve the correspondence problem. One can project in the near-infrared, to simultaneously take color images and retrieve the surface texture without interference from the projected pattern. So far, the problem with this has often been the weaker contrast obtained in the near-infrared band. Many such integrated approaches can be thought of.

This said, there is no single 3D acquisition system to date that can handle all types of objects or surfaces. Transparent or glossy surfaces (e.g., glass, metals), fine structures (e.g., hair or wires), and too weak, too busy, to too repetitive surface textures (e.g., identical tiles on a wall) may cause problems, depending on the system that is being used. The next section discusses still existing challenges in a bit more detail.

1.5 Challenges

The production of 3D models has been a popular research topic already for a long time now, and important progress has indeed been made since the early days. Nonetheless, the research community is well aware of the fact that still much remains to be done. In this section we list some of these challenges.

As seen in the previous section, there is a wide variety of techniques for creating 3D models, but depending on the geometry and material characteristics of the object or scene, one technique may be much better suited than another. For example, untextured objects are a nightmare for traditional stereo, but too much texture may interfere with the patterns of structured-light techniques. Hence, one would seem to need a battery of systems to deal with the variability of objects — e.g., in a museum — to be modeled. As a matter of fact, having to model the entire collections of diverse museums is a useful application area to think about, as it poses many of the pending challenges, often several at once. Another area is 3D city modeling, which has quickly grown in importance over the last years. It is another extreme in terms of conditions under which data have to be captured, in that cities represent an absolutely uncontrolled and large-scale environment. Also in that application area, many problems remain to be resolved.

Here is a list of remaining challenges, which we don't claim to be exhaustive:

- Many objects have an intricate shape, the scanning of which requires high precision combined with great agility of the scanner to capture narrow cavities and protrusions, deal with self-occlusions, fine carvings, etc.
- The types of objects and materials that potentially have to be handled — think of the museum example — are very diverse, like shiny metal coins, woven textiles, stone or wooden sculptures, ceramics, gems in jewellery and glass. No single technology can deal with all these surface types and for some of these types of artifacts there are no satisfactory techniques yet. Also, apart from the 3D shape the material characteristics may need to be captured as well.
- The objects to be scanned range from tiny ones like a needle to an entire construction or excavation site, landscape, or city. Ideally, one would handle this range of scales with the same techniques and similar protocols.
- For many applications, data collection may have to be undertaken on-site under potentially adverse conditions or

implying transportation of equipment to remote or harsh environments.

- Objects are sometimes too fragile or valuable to be touched and need to be scanned 'hands-off'. The scanner needs to be moved around the object, without it being touched, using portable systems.
- Masses of data often need to be captured, like in our museum collection or city modeling examples. More efficient data capture and model building are essential if this are to be practical.
- Those undertaking the digitization may or may not be technically trained. Not all applications are to be found in industry, and technically trained personnel may very well not be around. This raises the need for intelligent devices that ensure high-quality data through (semi-)automation, self-diagnosis, and effective guidance of the operator.
- In many application areas the money that can be spent is very limited and solutions therefore need to be relatively cheap.
- Also, precision is a moving target in many applications and as higher precisions are achieved, new applications present themselves that push for going even beyond. Analyzing the 3D surface of paintings to study brush strokes is a case in point.

These considerations about the particular conditions under which models may need to be produced, lead to a number of desirable, technological developments for 3D data acquisition:

- Combined extraction of shape and surface reflectance. Increasingly, 3D scanning technology is aimed at also extracting high-quality surface reflectance information. Yet, there still is an appreciable way to go before high-precision geometry can be combined with detailed surface characteristics like full-fledged BRDF (Bidirectional Reflectance Distribution Function) or BTF (Bidirectional Texture Function) information.

- In-hand scanning. The first truly portable scanning systems are already around. But the choice is still restricted, especially when also surface reflectance information is required and when the method ought to work with all types of materials, including metals, glass, etc. Also, transportable here is supposed to mean more than 'can be dragged between places', i.e., rather the possibility to easily move the system around the object, ideally also by hand. But there also is the interesting alternative to take the objects to be scanned in one's hands, and to manipulate them such that all parts get exposed to the fixed scanner. This is not always a desirable option (e.g., in the case of very valuable or heavy pieces), but has the definite advantage of exploiting the human agility in presenting the object and in selecting optimal, additional views.
- On-line scanning. The physical action of scanning and the actual processing of the data often still are two separate steps. This may create problems in that the completeness and quality of the result can only be inspected after the scanning session is over and the data are analyzed and combined at the lab or the office. It may then be too late or too cumbersome to take corrective actions, like taking a few additional scans. It would be very desirable if the system would extract the 3D data on the fly, and would give immediate visual feedback. This should ideally include steps like the integration and remeshing of partial scans. This would also be a great help in planning where to take the next scan during scanning. A refinement can then still be performed off-line.
- Opportunistic scanning. Not a single 3D acquisition technique is currently able to produce 3D models of even a large majority of exhibits in a typical museum. Yet, they often have complementary strengths and weaknesses. Untextured surfaces are a nightmare for passive techniques, but may be ideal for structured light approaches. Ideally, scanners would automatically adapt their strategy to the object at hand, based

on characteristics like spectral reflectance, texture spatial frequency, surface smoothness, glossiness, etc. One strategy would be to build a single scanner that can switch strategy on-the-fly. Such a scanner may consist of multiple cameras and projection devices, and by today's technology could still be small and light-weight.

- Multi-modal scanning. Scanning may not only combine geometry and visual characteristics. Additional features like non-visible wavelengths (UV,(N)IR) could have to be captured, as well as haptic impressions. The latter would then also allow for a full replay to the public, where audiences can hold even the most precious objects virtually in their hands, and explore them with all their senses.
- Semantic 3D. Gradually computer vision is getting at a point where scene understanding becomes feasible. Out of 2D images, objects and scene types can be recognized. This will in turn have a drastic effect on the way in which 'low'-level processes can be carried out. If high-level, semantic interpretations can be fed back into 'low'-level processes like motion and depth extraction, these can benefit greatly. This strategy ties in with the opportunistic scanning idea. Recognizing what it is that is to be reconstructed in 3D (e.g., a car and its parts) can help a system to decide how best to go about, resulting in increased speed, robustness, and accuracy. It can provide strong priors about the expected shape and surface characteristics.
- Off-the-shelf components. In order to keep 3D modeling cheap, one would ideally construct the 3D reconstruction systems on the basis of off-the-shelf, consumer products. At least as much as possible. This does not only reduce the price, but also lets the systems surf on a wave of fast-evolving, mass-market products. For instance, the resolution of still, digital cameras is steadily on the increase, so a system based on such camera(s) can be upgraded to higher quality without much effort or investment. Moreover, as most users will be acquainted with such components, the learning curve to use

the system is probably not as steep as with a totally novel, dedicated technology.

Obviously, once 3D data have been acquired, further processing steps are typically needed. These entail challenges of their own. Improvements in automatic remeshing and decimation are definitely still possible. Also solving large 3D puzzles automatically, preferably exploiting shape in combination with texture information, would be something in high demand from several application areas. Level-of-detail (LoD) processing is another example. All these can also be expected to greatly benefit from a semantic understanding of the data. Surface curvature alone is a weak indicator of the importance of a shape feature in LoD processing. Knowing one is at the edge of a salient, functionally important structure may be a much better reason to keep it in at many scales.

1.6 Conclusions

Given the above considerations, the 3D reconstruction of shapes from multiple, uncalibrated images is one of the most promising 3D acquisition techniques. In terms of our taxonomy of techniques, self-calibrating structure-from-motion ia a passive, multi-vantage point strategy. It offers high degrees of flexibility in that one can freely move a camera around an object or scene. The camera can be hand-held. Most people have a camera and know how to use it. Objects or scenes can be small or large, assuming that the optics and the amount of camera motion are appropriate. These methods also give direct access to both shape and surface reflectance information, where both can be aligned without special alignment techniques. Efficient implementations of several subparts of such SfM pipelines have been proposed lately, so that the online application of such methods is gradually becoming a reality. Also, the required hardware is minimal, and in many cases consumer type cameras will suffice. This keeps prices for data capture relatively low.

2

Principles of Passive 3D Reconstruction

2.1 Introduction

In this section the basic principles underlying self-calibrating, passive 3D reconstruction, are explained. More specifically, the central goal is to arrive at a 3D reconstruction from the uncalibrated image data alone. But, to understand how three-dimensional (3D) objects can be reconstructed from two-dimensional (2D) images, one first needs to know how the reverse process works: i.e., how images of a 3D object arise. Section 2.2 therefore discusses the image formation process in a camera and introduces the camera model which will be used throughout the text. As will become clear this model incorporates internal and external parameters related to the technical specifications of the camera(s) and their location with respect to the objects in the scene. Subsequent sections then set out to extract 3D models of the scene without prior knowledge of these parameters, i.e., without the need to calibrate the cameras internally or externally first. This reconstruction problem is formulated mathematically in Section 2.3 and a solution strategy is initiated. The different parts in this solution of the 3D reconstruction problem are elaborated in the following sections. Along the way, fundamental notions such as the correspondence problem, the epipolar

relation, and the fundamental matrix of an image pair are introduced (Section 2.4), and the possible stratification of the reconstruction process into Euclidean, metric, affine, and projective reconstructions is explained (Section 2.5). Furthermore, self-calibration equations are derived and their solution is discussed in Section 2.6. Apart from the generic case, special camera motions are considered as well (Section 2.7). In particular, camera translation rotations are discussed. These often occur in practice, but their systems of self-calibration equations or reconstruction equations become singular. Attention is paid also to the case of internally calibrated cameras and the important notion and use of the essential matrix is explored for that case.

As a note on widely used terminology in this domain, the word *camera* is often to be interpreted as a certain viewpoint and viewing direction — a field of view or image — and if mention is made of a first, second, ... camera then this can just as well refer to the same camera being moved around to a first, second, ... position.

2.2 Image Formation and Camera Model

2.2.1 The Pinhole Camera

The simplest model of the image formation process in a camera is that of a *pinhole camera* or *camera obscura*. The camera obscura is not more than a black box one side of which is punctured to yield a small hole. The rays of light from the outside world that pass through the hole and fall on the opposite side of the box there form a 2D image of the 3D environment outside the box (called the *scene*), as is depicted in Figure 2.1. Some art historians believe that the painter Vermeer actually used a room-sized version of a camera obscura. Observe that this pinhole image actually is the *photo-negative* image of the scene. The *photo-positive* image one observes when watching a photograph or a computer screen corresponds to the projection of the scene onto a hypothetical plane that is situated *in front* of the camera obscura at the same distance from the hole as the opposite wall on which the image is actually formed. In the sequel, the term *image plane* will always refer to this hypothetical plane in front of the camera. This hypothetical plane is preferred to avoid sign reversals in the computations. The distance

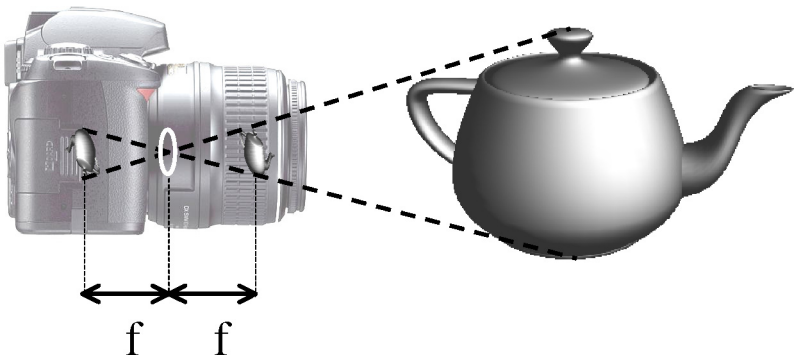

Fig. 2.1 In a *pinhole camera* or *camera obscura* an image of the scene is formed by the rays of light that are reflected by the objects in the scene and fall through the center of projection onto the opposite wall of the box, forming a photo-negative image of the scene. The *photo-positive* image of the scene corresponds to the projection of the scene onto a hypothetical image plane situated in front of the camera. It is this hypothetical plane which is typically used in computer vision, in order to avoid sign reversals.

between the center of projection (the hole) and the image plane is called the *focal length* of the camera.

The amount of light that falls into the box through the small hole is very limited. One can increase this amount of light by making the hole bigger, but then rays coming from different 3D points can fall onto the same point on the image, thereby causing blur. One way of getting around this problem is by making use of lenses, which focus the light. Apart from the introduction of geometric and chromatic aberrations, even the most perfect lens will come with a limited depth-of-field. This means that only scene points within a limited depth range are imaged sharply. Within that depth range the camera with lens basically behaves like the pinhole model. The 'hole' in the box will in the sequel be referred to as the *center of projection* or the *camera center*, and the type of projection realized by this idealized model is referred to as *perspective projection*.

It has to be noted that whereas in principle a single convex lense might be used, real camera lenses are composed of multiple lenses, in order to reduce deviations from the ideal model (i.e., to reduce the aforementioned aberrations). A detailed discussion on this important optical component is out of the scope of this tutorial, however.

2.2.2 Projection Equations for a Camera-Centered Reference Frame

To translate the image formation process into mathematical formulas we first introduce a reference frame for the 3D environment (also called the *world*) containing the scene. The easiest is to fix it to the camera. Figure 2.2 shows such a *camera-centered reference frame*. It is a right-handed and orthonormal reference frame whose origin is at the center of projection. Its Z-axis is the principal axis of the camera, — i.e., the line through the center of projection and orthogonal to the image plane — and the XY-plane is the plane through the center of projection and parallel to the image plane. The image plane is the plane with equation $Z = f$, where f denotes the focal length of the camera. The principal axis intersects the image plane in the *principal point* `p`.

The camera-centered reference frame induces an orthonormal uv reference frame in the image plane, as depicted in Figure 2.2. The image of a scene point `M` is the point `m` where the line through `M` and the origin of the camera-centered reference frame intersects the image plane. If `M` has coordinates $(X, Y, Z) \in \mathbb{R}^3$ with respect to the camera-centered reference frame, then an arbitrary point on the line through

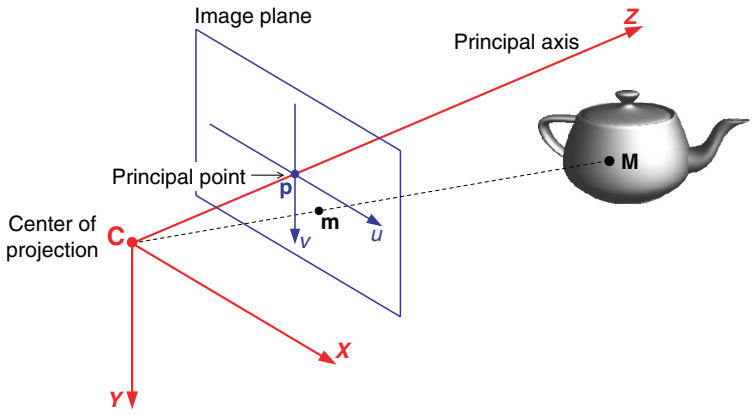

Fig. 2.2 The *camera-centered reference frame* is fixed to the camera and aligned with its intrinsic directions, of which the principal axis is one. The coordinates of the projection `m` of a scene point `M` onto the image plane in a pinhole camera model with a camera-centered reference frame, as expressed by equation (2.1), are given with respect to the *principal point* `p` in the image.

the origin and the scene point M has coordinates $\rho(X,Y,Z)$ for some real number ρ. The point of intersection of this line with the image plane must satisfy the relation $\rho Z = f$, or equivalently, $\rho = \frac{f}{Z}$. Hence, the image m of the scene point M has coordinates (u,v,f), where

$$u = f\frac{X}{Z} \quad \text{and} \quad v = f\frac{Y}{Z}. \tag{2.1}$$

Projections onto the image plane cannot be detected with infinite precision. An image rather consists of physical cells capturing photons, so-called *picture elements*, or *pixels* for short. Apart from some exotic designs (e.g., hexagonal or log-polar cameras), these pixels are arranged in a rectangular grid, i.e., according to rows and columns, as depicted in Figure 2.3 (left). Pixel positions are typically indicated with a row and column number measured with respect to the top left corner of the image. These numbers are called the *pixel coordinates* of an image point. We will denote them by (x,y), where the x-coordinate is measured horizontally and increasing to the right, and the y-coordinate is measured vertically and increasing downwards. This choice has several advantages:

- The way in which x- and y-coordinates are assigned to image points corresponds directly to the way in which an image is read out by several digital cameras with a CCD: starting at the top left and reading line by line.

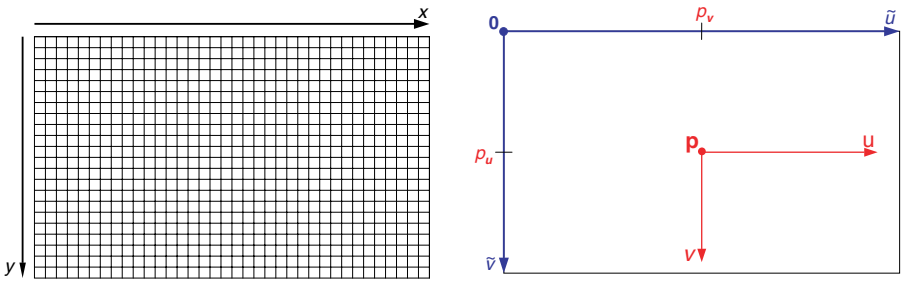

Fig. 2.3 *Left*: In a digital image, the position of a point in the image is indicated by its pixel coordinates. This corresponds to the way in which a digital image is read from a CCD. *Right*: The coordinates (u,v) of the projection of a scene point in the image are defined with respect to the principal point p. Pixel coordinates, on the other hand, are measures with respect to the upper left corner of the image.

- The camera-centered reference frame for the world being right-handed then implies that its Z-axis is pointing away from the image into the scene (as opposed to into the camera). Hence, the Z-coordinate of a scene point corresponds to the "depth" of that point with respect to the camera, which conceptually is nice because it is the big unknown to be solved for in 3D reconstruction problems.

As a consequence, we are not so much interested in the metric coordinates (u, v) indicating the projection $\mathtt{m}$ of a scene point $\mathtt{M}$ in the image and given by formula (2.1), as in the corresponding row and column numbers (x, y) of the underlying pixel. At the end of the day, it will be these pixel coordinates to which we have access when analyzing the image. Therefore we have to make the transition from (u, v)-coordinates to pixel coordinates (x, y) explicit first.

In a camera-centered reference frame the X-axis is typically chosen parallel to the rows and the Y-axis parallel to the columns of the rectangular grid of pixels. In this way, the u- and v-axes induced in the image plane have the same direction and sense as those in which the pixel coordinates x and y of image points are measured. But, whereas pixel coordinates are measured with respect to the top left corner of the image, (u, v)-coordinates are measured with respect to the principal point $\mathtt{p}$. The first step in the transition from (u, v)- to (x, y)-coordinates for an image point $\mathtt{m}$ thus is to apply offsets to each coordinate. To this end, denoted by p_u and p_v the metric distances, measured in the horizontal and vertical directions, respectively, of the principal point $\mathtt{p}$ from the upper left corner of the image (see Figure 2.3 (right)). With the top left corner of the image as origin, the principal point now has coordinates (p_u, p_v) and the perspective projection $\mathtt{m}$ of the scene point $\mathtt{M}$, as described by formula (2.1), will have coordinates

$$\tilde{u} = f\frac{X}{Z} + p_u \quad \text{and} \quad \tilde{v} = f\frac{Y}{Z} + p_v.$$

These $(\tilde{u}, \tilde{v})$-coordinates of the image point $\mathtt{m}$ are still expressed in the metric units of the camera-centered reference frame. To convert them to pixel coordinates, one has to divide $\tilde{u}$ and $\tilde{v}$ by the width and the height of a pixel, respectively. Let m_u and m_v be the inverse of resprespectively

the pixel width and height, then m_u and m_v indicate how many pixels fit into one horizontal resprespectively vertical metric unit. The pixel coordinates (x, y) of the projection $\mathtt{m}$ of the scene point $\mathtt{M}$ in the image are thus given by

$$x = m_u \left(f \frac{X}{Z} + p_u \right) \quad \text{and} \quad y = m_v \left(f \frac{Y}{Z} + p_v \right);$$

or equivalently,

$$x = \alpha_x \frac{X}{Z} + p_x \quad \text{and} \quad y = \alpha_y \frac{Y}{Z} + p_y, \tag{2.2}$$

where $\alpha_x = m_u f$ and $\alpha_y = m_v f$ is the focal length expressed in number of pixels for the x- and y-direction of the image and (p_x, p_y) are the pixel coordinates of the principal point. The ratio $\frac{\alpha_y}{\alpha_x} = \frac{m_v}{m_u}$, giving the ratio of the pixel width with respect to the pixel height, is called the *aspect ratio* of the pixels.

2.2.3 A Matrix Expression for Camera-Centered Projection

More elegant expressions for the projection Equation (2.2) are obtained if one uses *extended pixel coordinates* for the image points. In particular, if a point $\mathtt{m}$ with pixel coordinates (x, y) in the image is represented by the column vector $\mathtt{m} = (x, y, 1)^T$, then formula (2.2) can be rewritten as:

$$Z\,\mathtt{m} = Z \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{2.3}$$

Observe that, if one interpretes the extended pixel coordinates $(x, y, 1)^T$ of the image point $\mathtt{m}$ as a vector indicating a direction in the world, then, since Z describes the "depth" in front of the camera at which the corresponding scene point $\mathtt{M}$ is located, the 3×3-matrix

$$\begin{pmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

represents the transformation that converts world measurements (expressed in meters, centimeters, millimeters, ...) into the pixel metric of the digital image. This matrix is called the *calibration matrix*

of the camera, and it is generally represented as the upper triangular matrix

$$\mathbf{K} = \begin{pmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.4}$$

where α_x and α_y are the focal lengths expressed in number of pixels for the x- and y-directions in the image and with (p_x, p_y) the pixel coordinates of the principal point. The additional scalar s in the calibration matrix $\mathbf{K}$ is called the *skew* factor and models the situation in which the pixels are parallelograms (i.e., not rectangular). It also yields an approximation to the situation in which the physical imaging plane is not perfectly perpendicular to the optical axis of the lens or objective (as was assumed above). In fact, s is inversely proportional to the tangent of the angle between the X- and the Y-axis of the camera-centered reference frame. Consequently, $s = 0$ for digital cameras with rectangular pixels.

Together, the entries α_x, α_y, s, p_x, and p_y of the calibration matrix $\mathbf{K}$ describe the internal behavior of the camera and are therefore called the *internal parameters* of the camera. Furthermore, the projection Equation (2.3) of a pinhole camera with respect to a camera-centered reference frame for the scene are compactly written as:

$$\rho \mathtt{m} = \mathbf{K}\mathtt{M}, \tag{2.5}$$

where $\mathtt{M} = (X, Y, Z)^T$ are the coordinates of a scene point $\mathtt{M}$ with respect to the camera-centered reference frame for the scene, $\mathtt{m} = (x, y, 1)^T$ are the extended pixel coordinates of its projection $\mathtt{m}$ in the image, $\mathbf{K}$ is the calibration matrix of the camera. Furthermore, ρ is a positive real number and it actually represents the "depth" of the scene point $\mathtt{M}$ in front of the camera, because, due to the structure of the calibration matrix $\mathbf{K}$, the third row in the matrix Equation (2.5) reduces to $\rho = Z$. Therefore ρ is called the *projective depth* of the scene point $\mathtt{M}$ corresponding to the image point $\mathtt{m}$.

2.2.4 The General Linear Camera Model

When more than one camera is used, or when the objects in the scene are to be represented with respect to another, non-camera-centered

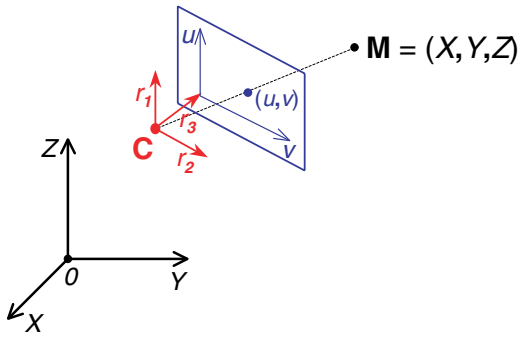

Fig. 2.4 The position and orientation of the camera in the scene are given by a position vector $\mathtt{C}$ and a 3×3-rotation matrix $\mathbf{R}$. The projection $\mathtt{m}$ of a scene point $\mathtt{M}$ is then given by formula (2.6).

reference frame (called the *world frame*), then the position and orientation of the camera in the scene are described by a point $\mathtt{C}$, indicating the center of projection, and a 3×3-rotation matrix $\mathbf{R}$ indicating the orientation of the camera-centered reference frame with respect to the world frame. More precisely, the column vectors $\mathbf{r}_i$ of the rotation matrix $\mathbf{R}$ are the unit direction vectors of the coordinate axes of the camera-centered reference frame, as depicted in Figure 2.4. As $\mathtt{C}$ and $\mathbf{R}$ represent the setup of the camera in the world space, they are called the *external parameters* of the camera.

The coordinates of a scene point $\mathtt{M}$ with respect to the camera-centered reference frame are found by projecting the relative position vector $\mathtt{M} - \mathtt{C}$ orthogonally onto each of the coordinate axes of the camera-centered reference frame. The column vectors $\mathbf{r}_i$ of the rotation matrix $\mathbf{R}$ being the unit direction vectors of the coordinate axes of the camera-centered reference frame, the coordinates of $\mathtt{M}$ with respect to the camera-centered reference frame are given by the dot products of the relative position vector $\mathtt{M} - \mathtt{C}$ with the unit vectors $\mathbf{r}_i$; or equivalently, by premultiplying the column vector $\mathtt{M} - \mathtt{C}$ with the transpose of the orientation matrix $\mathbf{R}$, viz. $\mathbf{R}^T(\mathtt{M} - \mathtt{C})$. Hence, following Equation (2.5), the projection $\mathtt{m}$ of the scene point $\mathtt{M}$ in the image is given by the (general) projection equations:

$$\rho \mathtt{m} = \mathbf{K}\mathbf{R}^T(\mathtt{M} - \mathtt{C}), \tag{2.6}$$

where $\mathtt{M} = (X, Y, Z)^T$ are the coordinates of a scene point $\mathtt{M}$ with respect to an (arbitrary) world frame, $\mathtt{m} = (x, y, 1)^T$ are the extended pixel coordinates of its projection $\mathtt{m}$ in the image, $\mathbf{K}$ is the calibration matrix of the camera, $\mathtt{C}$ is the position and $\mathbf{R}$ is the rotation matrix expressing the orientation of the camera with respect to the world frame, and ρ is a positive real number representing the projective depth of the scene point $\mathtt{M}$ with respect to the camera.

Many authors prefer to use extended coordinates for scene points as well. So, if $\binom{\mathtt{M}}{1} = (X, Y, Z, 1)^T$ are the extended coordinates of the scene point $\mathtt{M} = (X, Y, Z)^T$, then the projection Equation (2.6) becomes

$$\rho\mathtt{m} = \left(\mathbf{K}\mathbf{R}^T \mid -\mathbf{K}\mathbf{R}^T\mathtt{C}\right)\begin{pmatrix}\mathtt{M}\\ 1\end{pmatrix}. \tag{2.7}$$

The 3×4-matrix $\mathbf{P} = (\mathbf{K}\mathbf{R}^T \mid -\mathbf{K}\mathbf{R}^T\mathtt{C})$ is called the *projection matrix* of the camera.

Notice that, if only the 3×4-projection matrix $\mathbf{P}$ is known, it is possible to retrieve the internal and external camera parameters from it. Indeed, as is seen from formula (2.7), the upper left 3×3-submatrix of $\mathbf{P}$ is formed by multiplying $\mathbf{K}$ and $\mathbf{R}^T$. Its inverse is $\mathbf{R}\mathbf{K}^{-1}$, since $\mathbf{R}$ is a rotation matrix and thus $\mathbf{R}^T = \mathbf{R}^{-1}$. Furthermore, $\mathbf{K}$ is a non-singular upper triangular matrix and so is $\mathbf{K}^{-1}$. In particular, $\mathbf{R}\mathbf{K}^{-1}$ is the product of an orthogonal matrix and an upper triangular one. Recall from linear algebra that every 3×3-matrix of maximal rank can uniquely be decomposed as a product of an orthogonal and a non-singular, upper triangular matrix with positive diagonal entries by means of the QR-decomposition [3] (with Q the orthogonal and R the upper-diagonal matrix). Hence, given the 3×4-projection matrix $\mathbf{P}$ of a pinhole camera, the calibration matrix $\mathbf{K}$ and the orientation matrix $\mathbf{R}$ of the camera can easily be recovered from the inverse of the upper left 3×3-submatrix of $\mathbf{P}$ by means of QR-decomposition. If $\mathbf{K}$ and $\mathbf{R}$ are known, then the center of projection $\mathtt{C}$ is found by premultiplying the fourth column of $\mathbf{P}$ with the matrix $-\mathbf{R}\mathbf{K}^{-1}$.

The camera model of Equation (2.7) is usually referred to as the *general linear camera model*. Taking a close look at this formula shows how general the camera projection matrix $\mathbf{P} = (\mathbf{K}\mathbf{R}^T \mid -\mathbf{K}\mathbf{R}^T\mathtt{C})$ is as a matrix, in fact. Apart from the fact that the 3×3-submatrix on

the left has to be full rank, one cannot demand more than that it has to be QR-decomposable, which holds for any such matrix. The attentive reader may now object that, according to formula (2.4), the calibration matrix $\mathbf{K}$ must have entry 1 at the third position in the last row, whereas there is no such constraint for the upper triangular matrix in a QR-decomposition. This would seem a further restriction on the left 3×3-submatrix of $\mathbf{P}$, but it can easily be lifted by observing that the camera projection matrix $\mathbf{P}$ is actually only determined up to a scalar factor. Indeed, due to the non-zero scalar factor ρ in the left-hand side of formula (2.7), one can always ensure this property to hold. Put differently, *any* 3×4*-matrix whose upper left* 3×3*-submatrix is non-singular can be interpreted as the projection matrix of a* (*linear*) *pinhole camera*.

2.2.5 Non-linear Distortions

The perspective projection model described in the previous sections is linear in the sense that the scene point, the corresponding image point and the center of projection are collinear, and that straight lines in the scene do generate straight lines in the image. Perspective projection therefore only models the linear effects in the image formation process. Images taken by real cameras, on the other hand, also experience non-linear deformations or distortions which make the simple linear pinhole model inaccurate. The most important and best known non-linear distortion is *radial distortion*. Figure 2.5(left) shows an example of a radially distorted image. Radial distortion is caused by a systematic variation of the optical magnification when radially moving away from a certain point, called the center of distortion. The larger the distance between an image point and the center of distortion, the larger the effect of the distortion. Thus, the effect of the distortion is mostly visible near the edges of the image. This can clearly be seen in Figure 2.5 (left). Straight lines near the edges of the image are no longer straight but are bent. For practical use, the center of radial distortion can often be assumed to coincide with the principal point, which usually also coincides with the center of the image. But it should be noted that these are only approximations and dependent on the accuracy requirements, a more precise determination may be necessary [7].

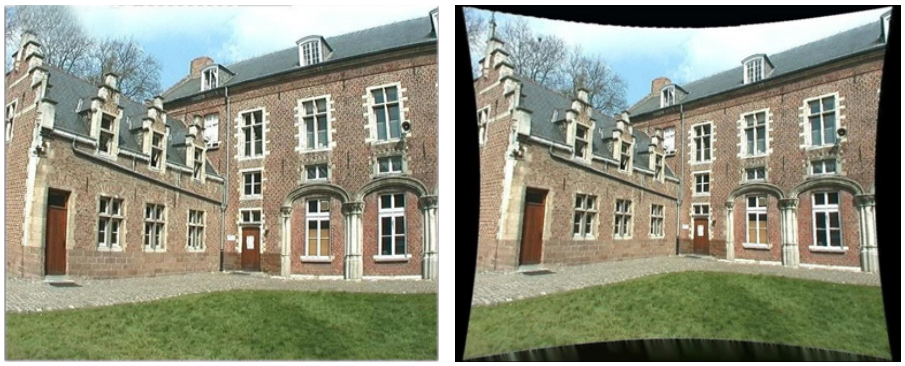

Fig. 2.5 *Left*: An image exhibiting radial distortion. The vertical wall at the left of the building appears bent in the image and the gutter on frontal wall on the right appears curved too. *Right*: The same image after removal of the radial distortion. Straight lines in the scene now appear as straight lines in the image as well.

Radial distortion is a non-linear effect and is typically modeled using a Taylor expansion. Typically, only the even order terms play a role in this expansion, i.e., the effect is symmetric around the center. The effect takes place in the lens, hence mathematically the radial distortion should be between the external and internal parameters of the pinhole model. The model we will propose here follows this strategy. Let us define

$$\rho\, \mathtt{m}_\mathtt{u} = \begin{pmatrix} m_{ux} \\ m_{uy} \\ 1 \end{pmatrix} = \mathbf{R}^T \left(\mathtt{M} - \mathbf{C} \right),$$

where $\mathtt{M} = (X, Y, Z)^T$ are the coordinates of a scene point with respect to the world frame. The distance r of the point $\mathtt{m}_\mathtt{u}$ from the optical axis is then

$$r^2 = m_{ux}^2 + m_{uy}^2.$$

We now define $\mathtt{m}_\mathtt{d}$ as

$$\mathtt{m}_\mathtt{d} = \begin{pmatrix} m_{dx} \\ m_{dy} \\ 1 \end{pmatrix} = \begin{pmatrix} \left(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \ldots\right) m_{ux} \\ \left(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \ldots\right) m_{uy} \\ 1 \end{pmatrix}. \tag{2.8}$$

The lower order terms of this expansion are the most important ones and typically one does not compute more than three parameters (κ_1, κ_2, κ_3). Finally the projection $\mathtt{m}$ of the 3D point $\mathtt{M}$ is:

$$\mathtt{m} = \mathbf{K}\mathtt{m}_\mathtt{d}. \tag{2.9}$$

When the distortion parameters are known, the image can be undistorted in order to make all lines straight again and thus make the linear pinhole model valid. The undistorted version of Figure 2.5 (left) is shown in the same Figure 2.5 (right).

The model described above puts the radial distortion parameters between the external and linear internal parameters of the camera. In the literature one often finds that the distortion is put on the left of the internal parameters, i.e., a 3D point is first projected into the image via the linear model and then shifted. Conceptually the latter model is less suited than the one used here because putting the distortion at the end makes it dependent on the internal parameters, especially on the focal length. This means one has to re-estimate the radial distortion parameters every time the focal length changes. This is not necessary in the model as suggested here. This, however, does not mean that this model is perfect. In reality the center of radial distortion (now assumed to be the principal point) sometimes changes when the focal length is altered. This effect cannot be modeled with this approach.

In the remainder of the text we will assume that radial distortion has been removed from all images if present, unless stated otherwise.

2.2.6 Explicit Camera Calibration

As explained before a perspective camera can be described by its internal and external parameters. The process of determining these parameters is known as *camera calibration*. Accordingly, we make a distinction between internal or internal calibration and external or external calibration, also known as pose estimation. For the determination of all parameters one often uses the term complete calibration. Traditional 3D passive reconstruction techniques had a separate, explicit camera calibration step. As highlighted before, the difference with self-calibration techniques as explained in this tutorial is that in the latter

Fig. 2.6 Calibration objects: 3D (left) and 2D (right).

the same images used for 3D scene reconstruction are also used for camera calibration.

Traditional internal calibration procedures [1, 17, 18] extract the camera parameters from a set of known 3D–2D correspondences, i.e., a set of 3D coordinates with corresponding 2D coordinates in the image for their projections. In order to easily obtain such 3D–2D correspondences, they employ special calibration objects, like the ones displayed in Figure 2.6. These objects contain easily recognizable markers. Internal calibration sometimes starts by fitting a linearized calibration model to the 3D–2D correspondences, which is then improved with a subsequent non-linear optimization step.

Some applications do not suffer from unknown scales or effects of projection, but their results would deteriorate under the influence of non-linear effects like radial distortion. It is possible to only undo such distortion without having to go through the entire process of internal calibration. One way is by detecting structures that are known to be straight, but appear curved under the influence of the radial distortion. Then, for each line (e.g., the curved roof top in the example of Figure 2.5 (left)), points are sampled along it and a straight line is fitted through these data. If we want the distortion to vanish, the error consisting of the sum of the distances of each point to their line should be zero. Hence a non-linear minimization algorithm like Levenberg–Marquardt is applied to these data. The algorithm is initialized with the distortion parameters set to zero. At every iteration, new values are computed for these parameters, the points are warped accordingly and new lines are

fitted. The algorithm stops when it converges to a solution where all selected lines are straight (i.e., the resulting error is close to zero). The resulting unwarped image can be seen on the right in Figure 2.5.

2.3 The 3D Reconstruction Problem

The aim of passive 3D reconstruction is to recover the geometric structure of a (static) scene from one or more of its images: Given a point $\mathtt{m}$ in an image, determine the point $\mathtt{M}$ in the scene of which $\mathtt{m}$ is the projection. Or, in mathematical parlance, given the pixel coordinates (x, y) of a point $\mathtt{m}$ in a digital image, determine the world coordinates (X, Y, Z) of the scene point $\mathtt{M}$ of which $\mathtt{m}$ is the projection in the image. As can be observed from the schematic representation of Figure 2.7, a point $\mathtt{m}$ in the image can be the projection of *any* point $\mathtt{M}$ in the world space that lies on the line through the center of projection and the image point $\mathtt{m}$. Such line is called the *projecting ray* or the *line of sight* of the image point $\mathtt{m}$ in the given camera. Thus, 3D reconstruction from one image is an underdetermined problem.

On the other hand, if two images of the scene are available, the position of a scene point $\mathtt{M}$ can be recovered from its projections $\mathtt{m}_1$ and $\mathtt{m}_2$ in the images by triangulation: $\mathtt{M}$ is the point of intersection of the projecting rays of $\mathtt{m}_1$ and $\mathtt{m}_2$, as depicted in Figure 2.8. This stereo setup and the corresponding principle of triangulation have already been intro-

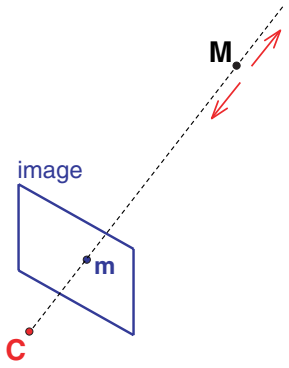

Fig. 2.7 3D reconstruction from one image is an underdetermined problem: a point $\mathtt{m}$ in the image can be the projection of *any* world point $\mathtt{M}$ along the projecting ray of $\mathtt{m}$.

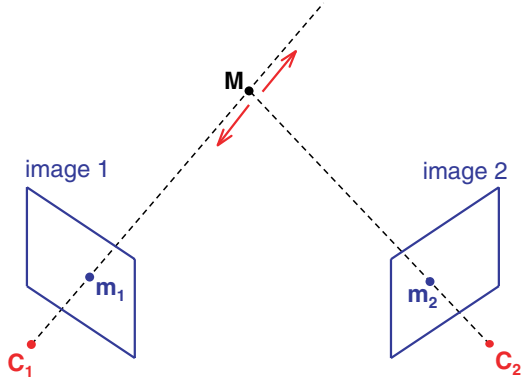

Fig. 2.8 Given two images of a static scene, the location of the scene point $\mathtt{M}$ can be recovered from its projections $\mathtt{m}_1$ and $\mathtt{m}_2$ in the respective images by means of triangulation.

duced in Section 1. As already noted there, the 3D reconstruction problem has not yet been solved, unless the internal and external parameters of the cameras are known. Indeed, if we assume that the images are corrected for radial distortion and other non-linear effects and that the general linear pinhole camera model is applicable, then, according to Equation (2.6) in Section 2.2.4, the projection equations of the first camera are modeled as:

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1), \tag{2.10}$$

where $\mathtt{M} = (X, Y, Z)^T$ are the coordinates of the scene point $\mathtt{M}$ with respect to the world frame, $\mathtt{m}_1 = (x_1, y_1, 1)^T$ are the extended pixel coordinates of its projection $\mathtt{m}_1$ in the first image, $\mathbf{K}_1$ is the calibration matrix of the first camera, $\mathtt{C}_1$ is the position and $\mathbf{R}_1$ is the orientation of the first camera with respect to the world frame, and ρ_1 is a positive real number representing the projective depth of $\mathtt{M}$ with respect to the first camera. To find the projecting ray of an image point $\mathtt{m}_1$ in the first camera and therefore all points projecting onto $\mathtt{m}_1$ there, recall from Section 2.2.3 that the calibration matrix $\mathbf{K}_1$ converts world measurements (expressed in meters, centimeters, millimeters, etc) into the pixel metric of the digital image. Since $\mathtt{m}_1 = (x_1, y_1, 1)$ are the extended pixel coordinates of the point $\mathtt{m}_1$ in the first image, the direction of the projecting ray of $\mathtt{m}_1$ in the camera-centered reference frame of the first camera is given by the three-vector $\mathbf{K}_1^{-1}\mathtt{m}_1$. With respect to the

world frame, the direction vector of the projecting ray is $\mathbf{R}_1\mathbf{K}_1^{-1}\mathtt{m}_1$, by definition of $\mathbf{R}_1$. As the position of the first camera in the world frame is given by the point $\mathtt{C}_1$, the parameter equations of the projecting ray of $\mathtt{m}_1$ in the world frame are:

$$\mathtt{M} = \mathtt{C}_1 + \rho_1\mathbf{R}_1\mathbf{K}_1^{-1}\mathtt{m}_1 \quad \text{for some } \rho_1 \in \mathbb{R}. \tag{2.11}$$

So, every scene point $\mathtt{M}$ satisfying Equation (2.11) for some real number ρ_1 projects onto the point $\mathtt{m}_1$ in the first image. Notice that Equation (2.11) can be found directly by solving the projection Equation (2.10) for $\mathtt{M}$. Clearly, the parameter Equation (2.11) of the projecting ray of a point $\mathtt{m}_1$ in the first image are only fully known, provided the calibration matrix $\mathbf{K}_1$ and the position $\mathtt{C}_1$ and orientation $\mathbf{R}_1$ of the camera with respect to the world frame are known (i.e., when the first camera is fully calibrated).

Similarly, the projection equations for the second camera are:

$$\rho_2\mathtt{m}_2 = \mathbf{K}_2\mathbf{R}_2^T(\mathtt{M} - \mathtt{C}_2), \tag{2.12}$$

where $\mathtt{m}_2 = (x_2, y_2, 1)^T$ are the extended pixel coordinates of $\mathtt{M}$'s projection $\mathtt{m}_2$ in the second image, $\mathbf{K}_2$ is the calibration matrix of the second camera, $\mathtt{C}_2$ is the position and $\mathbf{R}_2$ the orientation of the second camera with respect to the world frame, and ρ_2 is a positive real number representing the projective depth of $\mathtt{M}$ with respect to the second camera. Solving Equation (2.12) for $\mathtt{M}$ yields:

$$\mathtt{M} = \mathtt{C}_2 + \rho_2\mathbf{R}_2\mathbf{K}_2^{-1}\mathtt{m}_2; \tag{2.13}$$

and, if in this equation ρ_2 is seen as a parameter, then formula (2.13) is just the parameter equation for the projecting ray of the image point $\mathtt{m}_2$ in the second camera. Again, these parameter equations are fully known only if $\mathbf{K}_2$, $\mathtt{C}_2$, and $\mathbf{R}_2$ are known (i.e., when the second camera is fully calibrated). The system of six Equations (2.10) and (2.12) can be solved for the five unknowns X, Y, Z, ρ_1, and ρ_2. Observe that this requires the system to be rank-deficient, which is guaranteed if the points $\mathtt{m}_1$ and $\mathtt{m}_2$ are in correspondence (i.e., their projecting rays intersect) and therefore special relations (in particular, the so-called *epipolar relations*, which will be derived in the next section) hold.

When the cameras are not internally and externally calibrated, then it is not immediately clear how to perform triangulation from the image data alone. On the other hand, one intuitively feels that every image of a static scene constrains in one way or another the shape and the relative positioning of the objects in the world, even if no information about the camera parameters is known. The key to the solution of the 3D reconstruction problem is found in understanding how the locations $\mathtt{m}_1$ and $\mathtt{m}_2$ of the projections of a scene point $\mathtt{M}$ in different views are related to each other. This relationship is explored in the next section.

2.4 The Epipolar Relation Between 2 Images of a Static Scene

2.4.1 The Fundamental Matrix

A point $\mathtt{m}_1$ in a first image of the scene is the projection of a scene point $\mathtt{M}$ that can be at any position along the projecting ray of $\mathtt{m}_1$ in that first camera. Therefore, the corresponding point $\mathtt{m}_2$ (i.e., the projection of $\mathtt{M}$) in a second image of the scene must lie on the projection ℓ_2 of this projecting ray in the second image, as depicted in Figure 2.9. To derive the equation of this projection ℓ_2, suppose for a moment that the internal and external parameters of both cameras are known. Then, the projecting ray of the point $\mathtt{m}_1$ in the first camera is given

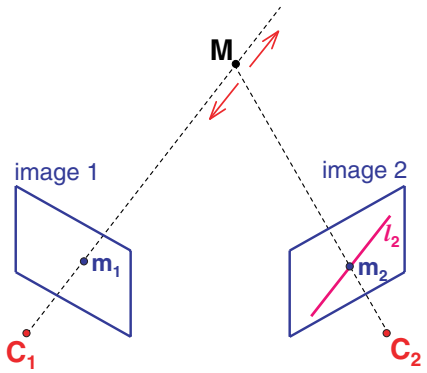

Fig. 2.9 The point $\mathtt{m}_2$ in the second image corresponding to a point $\mathtt{m}_1$ in the first image lies on the epipolar line ℓ_2 which is the projection in the second image of the projecting ray of $\mathtt{m}_1$ in the first camera.

by Equation (2.11), viz. $\mathtt{M} = \mathtt{C}_1 + \rho_1 \mathbf{R}_1 \mathbf{K}_1^{-1} \mathtt{m}_1$. Substituting the right-hand side in the projection Equation (2.12) of the second camera, yields

$$\rho_2 \mathtt{m}_2 = \rho_1 \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1} \mathtt{m}_1 + \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{C}_1 - \mathtt{C}_2). \tag{2.14}$$

The last term in this equation corresponds to the projection $\mathtt{e}_2$ of the position $\mathtt{C}_1$ of the first camera in the second image:

$$\rho_{e2} \mathtt{e}_2 = \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{C}_1 - \mathtt{C}_2). \tag{2.15}$$

$\mathtt{e}_2$ is called the *epipole* of the first camera in the second image. The first term in the right-hand side of Equation (2.14), on the other hand, indicates the direction of the projecting ray (2.11) in the second image. Indeed, recall from Section 2.3 that $\mathbf{R}_1 \mathbf{K}_1^{-1} \mathtt{m}_1$ is the direction vector of the projecting ray of $\mathtt{m}_1$ with respect to the world frame. In the camera-centered reference frame of the second camera, the coordinates of this vector are $\mathbf{R}_2^T \mathbf{R}_1 \mathbf{K_1}^{-1} \mathtt{m}_1$. The point in the second image that corresponds to this viewing direction is then given by $\mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K_1}^{-1} \mathtt{m}_1$. Put differently, $\mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K_1}^{-1} \mathtt{m}_1$ are homogeneous coordinates for the vanishing point of the projecting ray (2.11) in the second image, as can be seen from Figure 2.10.

To simplify the notation, put $\mathbf{A} = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1}$. Then $\mathbf{A}$ is an invertible 3×3-matrix which, for every point $\mathtt{m}_1$ in the first image, gives

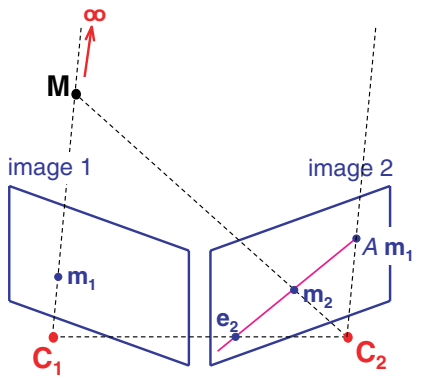

Fig. 2.10 The epipole $\mathtt{e}_2$ of the first camera in the second image indicates the position in the second image where the center of projection $\mathtt{C}_1$ of the first camera is observed. The point $\mathbf{A}\mathtt{m}_1$ in the second image is the vanishing point of the projecting ray of $\mathtt{m}_1$ in the second image.

homogeneous coordinates $\mathbf{A}\,\mathtt{m}_1$ for the vanishing point in the second view of the projecting ray of $\mathtt{m}_1$ in the first camera. In the literature this matrix is referred to as the *infinite homography*, because it corresponds to the 2D projective transformation induced between the images by the plane at infinity of the scene. More about this interpretation of the matrix $\mathbf{A}$ can be found in Section 2.4.3. Formula (2.14) can now be rewritten as:

$$\rho_2\,\mathtt{m}_2 = \rho_1\,\mathbf{A}\,\mathtt{m}_1 + \rho_{e2}\,\mathtt{e}_2. \tag{2.16}$$

Formula (2.16) algebraically expresses the geometrical observation that, *for a given point* $\mathtt{m}_1$ *in one image, the corresponding point* $\mathtt{m}_2$ *in another image of the scene lies on the line* ℓ_2 *through the epipole* $\mathtt{e}_2$ *and the vanishing point* $\mathbf{A}\,\mathtt{m}_1$ *of the projecting ray of* $\mathtt{m}_1$ in the first camera (cf. Figure 2.10). The line ℓ_2 is called the *epipolar line* in the second image corresponding to $\mathtt{m}_1$, and Equation (2.16) is referred to as the *epipolar relation* between corresponding image points. ℓ_2 is the sought projection in the second image of the entire projecting ray of $\mathtt{m}_1$ in the first camera. It is important to realize that the epipolar line ℓ_2 in the second image relies on the point $\mathtt{m}_1$ in the first image. Put differently, selecting another point $\mathtt{m}_1$ in the first image generically will result in another epipolar line ℓ_2 in the second view. However, as is seen from formula (2.16), these epipolar lines all run through the epipole $\mathtt{e}_2$. This was to be expected, of course, as all the projecting rays of the first camera originate from the center of projection $\mathtt{C}_1$ of the first camera. Hence, their projections in the second image — which, by definition, are epipolar lines in the second view — must also all run through the projection of $\mathtt{C}_1$ in the second image, which is just the epipole $\mathtt{e}_2$. Figure 2.11 illustrates this observation graphically.

In the literature the epipolar relation (2.16) is usually expressed in closed form. To this end, we fix some notations: for a three-vector $\mathtt{a} = (a_1, a_2, a_3)^T \in \mathbb{R}^3$, let $[\mathtt{a}]_\times$ denote the skew-symmetric 3×3-matrix

$$[\mathtt{a}]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}, \tag{2.17}$$

which represents the cross product with $\mathtt{a}$; i.e., $[\mathtt{a}]_\times \mathtt{v} = \mathtt{a} \times \mathtt{v}$ for all three-vectors $\mathtt{v} \in \mathbb{R}^3$. Observe that $[\mathtt{a}]_\times$ has rank 2 if $\mathtt{a}$ is non-zero.

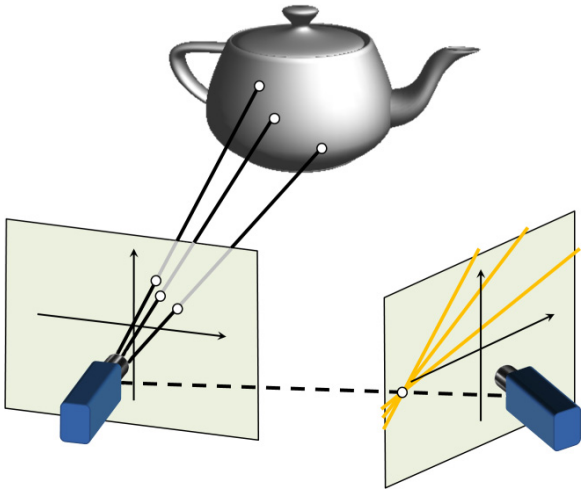

Fig. 2.11 All projecting rays of the first camera originate from its center of projection. Their projections into the image plane of the second camera are therefore all seen to intersect in the projection of this center of projection, i.e., at the epipole.

The epipolar relation states that, for a point $\mathtt{m}_1$ in the first image, its corresponding point $\mathtt{m}_2$ in the second image must lie on the line through the epipole $\mathtt{e}_2$ and the vanishing point $\mathbf{A}\mathtt{m}_1$. Algebraically, this is expressed by demanding that the three-vectors $\mathtt{m}_2$, $\mathtt{e}_2$, and $\mathbf{A}\mathtt{m}_1$ representing homogeneous coordinates of the corresponding image points are linearly dependent (cf. Equation (2.16)). Recall from linear algebra that this is equivalent to $|\mathtt{m}_2\mathtt{e}_2\mathbf{A}\mathtt{m}_1| = 0$, where the vertical bars denote the determinant of the 3×3-matrix whose columns are the specified column vectors. Moreover, by definition of the cross product, this determinant equals

$$|\mathtt{m}_2 \ \mathtt{e}_2 \ \mathbf{A}\mathtt{m}_1| = \mathtt{m}_2^T(\mathtt{e}_2 \times \mathbf{A}\mathtt{m}_1).$$

Expressing the cross product as a matrix multiplication then yields

$$|\mathtt{m}_2 \ \mathtt{e}_2 \ \mathbf{A}\mathtt{m}_1| = \mathtt{m}_2^T[\mathtt{e}_2]_\times \mathbf{A}\mathtt{m}_1.$$

Hence, the epipolar relation (2.16) is equivalently expressed by the equation

$$\mathtt{m}_2^T \mathbf{F} \mathtt{m}_1 = 0, \tag{2.18}$$

where $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}$ is a 3×3-matrix, called the *fundamental matrix* of the image pair, and with $\mathtt{e}_2$ the epipole in the second image and $\mathbf{A}$ the invertible 3×3-matrix defined above [2, 4]. Note that, since $[\mathtt{a}]_\times$ is a rank 2 matrix, the fundamental matrix $\mathbf{F}$ also has rank 2.

2.4.2 Gymnastics with F

The closed form (2.18) of the epipolar relation has the following advantages:

(1) *The fundamental matrix* $\mathbf{F}$ *can, up to a non-zero scalar factor, be computed from the image data alone.*
Indeed, for each pair of corresponding points $\mathtt{m}_1$ and $\mathtt{m}_2$ in the images, Equation (2.18) yields one homogeneous linear equation in the entries of the fundamental matrix $\mathbf{F}$. Knowing (at least) eight pairs of corresponding point between the two images, the fundamental matrix $\mathbf{F}$ can, up to a non-zero scalar factor, be computed from these point correspondences in a linear manner. Moreover, by also exploiting the fact that $\mathbf{F}$ has rank 2, the fundamental matrix $\mathbf{F}$ can even be computed, up to a non-zero scalar factor, from 7 point correspondences between the images, albeit by a non-linear algorithm as the rank 2 condition involves a relation between products of three entries of $\mathbf{F}$. Different methods for efficient and robust computation of $\mathbf{F}$ will be explained in Section 4.2.2 of Section 4.

(2) Given $\mathbf{F}$, *the epipole* $\mathtt{e}_2$ *in the second image is the unique three-vector with third coordinate equal to* 1, *satisfying* $\mathbf{F}^T \mathtt{e}_2 = 0$.
This observation follows immediately from the fact that $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}$ and that $[\mathtt{e}_2]_\times^T \mathtt{e}_2 = -[\mathtt{e}_2]_\times \mathtt{e}_2 = -\mathtt{e}_2 \times \mathtt{e}_2 = \mathtt{0}$.

(3) Similarly, *the epipole* $\mathtt{e}_1$ *of the second camera in the first image — i.e., the projection* $\mathtt{e}_1$ *of the position* $\mathtt{C}_2$ *of the second camera in the first image — is the unique three-vector with third coordinate equal to* 1, *satisfying* $\mathbf{F}\mathtt{e}_1 = 0$.
According to Equation (2.6) in Section 2.2.4, the projection $\mathtt{e}_1$ of the position $\mathtt{C}_2$ of the second camera in

the first image is given by $\rho_{e1}\,\mathtt{e}_1 = \mathbf{K}_1\,\mathbf{R}_1^T(\mathtt{C}_2 - \mathtt{C}_1)$, with ρ_{e1} a non-zero scalar factor. Since $\mathbf{A} = \mathbf{K}_2\mathbf{R}_2^T\mathbf{R}_1\mathbf{K}_1^{-1}$, $\rho_{e1}\,\mathbf{A}\mathtt{e}_1 = \mathbf{K}_2\mathbf{R}_2^T(\mathtt{C}_2 - \mathtt{C}_1) = -\rho_{e2}\mathtt{e}_2$, and thus $\rho_{e1}\,\mathbf{F}\mathtt{e}_1 = [\mathtt{e}_2]_\times\,(\rho_{e1}\mathbf{A}\mathtt{e}_1) = [\mathtt{e}_2]_\times(-\rho_{e2}\,\mathtt{e}_2) = \mathtt{0}$. Notice that this also shows that the infinite homography $\mathbf{A}$ maps the epipole $\mathtt{e}_1$ in the first image onto the epipole $\mathtt{e}_2$ in the second image.

(4) *Given a point* $\mathtt{m}_1$ *in the first image, the three-vector* $\mathbf{F}\mathtt{m}_1$ *yields homogeneous coordinates for the epipolar line* ℓ_2 *in the second image corresponding to* $\mathtt{m}_1$; *i.e.,* $\ell_2 \simeq \mathbf{F}\mathtt{m}_1$.
Recall that the epipolar relation $\mathtt{m}_2^T\mathbf{F}\mathtt{m}_1 = 0$ expresses the geometrical observation that the point $\mathtt{m}_2$ in the second image, which corresponds to $\mathtt{m}_1$, lies on the line ℓ_2 through the epipole $\mathtt{e}_2$ and the point $\mathbf{A}\mathtt{m}_1$, which by definition is the epipolar line in the second image corresponding to $\mathtt{m}_1$. This proves the claim.

(5) Similarly, *given a point* $\mathtt{m}_2$ *in the second image, the three-vector* $\mathbf{F}^T\mathtt{m}_2$ *yields homogeneous coordinates for the epipolar line* ℓ_1 *in the first image corresponding to* $\mathtt{m}_2$; *i.e.,* $\ell_1 \simeq \mathbf{F}^T\mathtt{m}_2$.
By interchanging the role of the two images in the reasoning leading up to the epipolar relation derived above, one easily sees that the epipolar line ℓ_1 in the first image corresponding to a point $\mathtt{m}_2$ in the second image is the line through the epipole $\mathtt{e}_1$ in the first image and the vanishing point $\mathbf{A}^{-1}\mathtt{m}_2$ in the first image of the projecting ray of $\mathtt{m}_2$ in the second camera. The corresponding epipolar relation

$$|\mathtt{m}_1\ \mathtt{e}_1\ \mathbf{A}^{-1}\mathtt{m}_2| = 0 \tag{2.19}$$

expresses that $\mathtt{m}_1$ lies on that line. As $\mathbf{A}$ is an invertible matrix, its determinant $|\mathbf{A}|$ is a non-zero scalar. Multiplying the left-hand side of Equation (2.19) with $|\mathbf{A}|$ yields

$$\begin{aligned} |\mathbf{A}||\mathtt{m}_1\ \mathtt{e}_1\ \mathbf{A}^{-1}\mathtt{m}_2| &= |\mathbf{A}\mathtt{m}_1\ \mathbf{A}\mathtt{e}_1\ \mathtt{m}_2| \\ &= |\mathbf{A}\mathtt{m}_1\ -(\rho_{e2}/\rho_{e1})\mathtt{e}_2\ \mathtt{m}_2| \\ &= \frac{\rho_{e2}}{\rho_{e1}}\,|\mathtt{m}_2\ \mathtt{e}_2\ \mathbf{A}\mathtt{m}_1| = \frac{\rho_{e2}}{\rho_{e1}}\,\mathtt{m}_2^T\mathbf{F}\mathtt{m}_1, \end{aligned}$$

because $\rho_{e1}\,\mathbf{A}\mathtt{e}_1 = -\rho_{e2}\,\mathtt{e}_2$, as seen in number 3 above, and $|\mathtt{m}_2\ \ \mathtt{e}_2\ \ \mathbf{A}\mathtt{m}_1| = \mathtt{m}_2^T(\mathtt{e}_2 \times \mathbf{A}\mathtt{m}_1) = \mathtt{m}_2^T\mathbf{F}\mathtt{m}_1$, by definition of the fundamental matrix $\mathbf{F}$ (cf. Equation (2.18)). Consequently, the epipolar relation (2.19) is equivalent to $\mathtt{m}_2^T\mathbf{F}\mathtt{m}_1 = 0$, and the epipolar line ℓ_1 in the first image corresponding to a given point $\mathtt{m}_2$ in the second image has homogeneous coordinates $\mathbf{F}^T\mathtt{m}_2$. We could have concluded this directly from Equation 2.18 based on symmetry considerations.

2.4.3 Grasping the Infinite Homography

Before continuing our investigation on how to recover 3D information about the scene from images alone, it is worth to have a closer look at the invertible matrix $\mathbf{A}$ introduced in Section 2.4.1 first. The matrix $\mathbf{A}$ is defined algebraically as $\mathbf{A} = \mathbf{K}_2\mathbf{R}_2^T\mathbf{R}_1\mathbf{K}_1^{-1}$, but it also has a clear geometrical interpretation: *the matrix* $\mathbf{A}$ *transfers vanishing points of directions in the scene from the first image to the second one*. Indeed, consider a line $\mathtt{L}$ in the scene with direction vector $\mathtt{V} \in \mathbb{R}^3$. The vanishing point $\mathtt{v}_1$ of its projection ℓ_1 in the first image is the point of intersection of the line through the center of projection $\mathtt{C}_1$ and parallel to $\mathtt{L}$ with the image plane of the first camera, as depicted in Figure 2.12. Parameter equations of this line are $\mathtt{M} = \mathtt{C}_1 + \tau\mathtt{V}$ with τ a scalar parameter, and the projection of every point $\mathtt{M}$ on this line in the first image is given by $\mathbf{K}_1\mathbf{R}_1^T(\mathtt{X} - \mathtt{C}_1) = \tau\mathbf{K}_1\mathbf{R}_1^T\mathtt{V}$. The vanishing

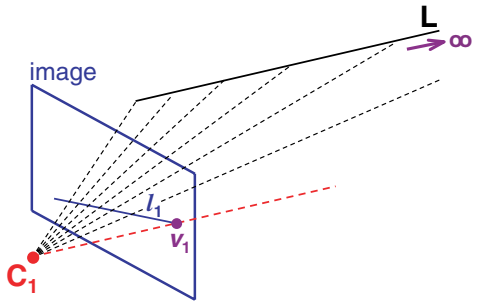

Fig. 2.12 The vanishing point $\mathtt{v}_1$ in the first image of the projection ℓ_1 of a line $\mathtt{L}$ in the scene is the point of intersection of the line through the center of projection $\mathtt{C}_1$ and parallel to $\mathtt{L}$ with the image plane.

point $\mathtt{v}_1$ of the line ℓ_1 in the first image thus satisfies the equation $\rho_{v1}\,\mathtt{v}_1 = \mathbf{K}_1\,\mathbf{R}_1^T\,\mathtt{V}$ for some non-zero scalar ρ_{v1}. Similarly, the vanishing point $\mathtt{v}_2$ of the projection ℓ_2 of the line $\mathtt{L}$ in the second image is given by $\rho_{v2}\,\mathtt{v}_2 = \mathbf{K}_2\,\mathbf{R}_2^T\,\mathtt{V}$ for some non-zero scalar ρ_{v2}. Conversely, given a vanishing point $\mathtt{v}_1$ in the first image, the corresponding direction vector $\mathtt{V}$ in the scene is $\mathtt{V} = \rho_{v1}\,\mathbf{R}_1\,\mathbf{K}_1^{-1}\,\mathtt{v}_1$ for some scalar ρ_{v1}, and its vanishing point in the second image is given by $\rho_{v2}\,\mathtt{v}_2 = \rho_{v1}\,\mathbf{K}_2\,\mathbf{R}_2^T\,\mathbf{R}_1\,\mathbf{K}_1^{-1}\,\mathtt{v}_1$. As $\mathbf{A} = \mathbf{K}_2\,\mathbf{R}_2^T\,\mathbf{R}_1\,\mathbf{K}_1^{-1}$, this relation between the vanishing points $\mathtt{v}_1$ and $\mathtt{v}_2$ can be simplified to $\rho\,\mathtt{v}_2 = \mathbf{A}\,\mathtt{v}_1$, where $\rho = \frac{\rho_{v2}}{\rho_{v1}}$ is a non-zero scalar factor. Hence, *if* $\mathtt{v}_1$ *is the vanishing point of a line in the first image, then* $\mathbf{A}\mathtt{v}_1$ *are homogeneous coordinates of the vanishing point of the corresponding line in the second image*, as was claimed. In particular, the observation made in Section 2.4.1 that for any point $\mathtt{m}_1$ in the first image $\mathbf{A}\,\mathtt{m}_1$ are homogeneous coordinates for the vanishing point in the second view of the projecting ray of $\mathtt{m}_1$ in the first camera is in fact another instance of the same general property.

As is explained in more detail in Appendix A, in projective geometry, direction vectors $\mathtt{V}$ in the scene are represented as points on the plane at infinity of the scene. A vanishing point in an image then is just the perspective projection onto the image plane of a point on the plane at infinity in the scene. In this respect, the matrix $\mathbf{A}$ is a homography matrix of the projective transformation that maps points from the first image via the plane at infinity of the scene into the second image. This explains why $\mathbf{A}$ is called the *infinite homography* in the computer vision literature.

2.5 Two Image-Based 3D Reconstruction Up-Close

From an algebraic point of view, triangulation can be interpreted as solving the two camera projection equations for the scene point $\mathtt{M}$. Formulated in this way, passive 3D reconstruction is seen as solving the following problem: Given two images $\mathcal{I}_1$ and $\mathcal{I}_2$ of a static scene and a set of corresponding image points $\mathtt{m}_1 \in \mathcal{I}_1$ and $\mathtt{m}_2 \in \mathcal{I}_2$ between these images, determine a calibration matrix $\mathbf{K}_1$, a position $\mathtt{C}_1$ and an orientation $\mathbf{R}_1$ for the first camera and a calibration matrix $\mathbf{K}_2$, a position $\mathtt{C}_2$ and an orientation $\mathbf{R}_2$ for the second camera, and for every pair of

corresponding image points $\mathtt{m}_1 \in \mathcal{I}_1$ and $\mathtt{m}_2 \in \mathcal{I}_2$ compute world coordinates (X, Y, Z) of a scene point $\mathtt{M}$ such that:

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1) \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{M} - \mathtt{C}_2). \tag{2.20}$$

These equations are the point of departure for our further analysis. In traditional stereo one would know $\mathbf{K}_1$, $\mathbf{R}_1$, $\mathtt{C}_1$, $\mathbf{K}_2$, $\mathbf{R}_2$, and $\mathtt{C}_2$. Then formula (2.20) yields a system of six linear equations in five unknowns from which the coordinates of $\mathtt{M}$ as well as the scalar factors ρ_1 and ρ_2 can be computed, as was explained in Section 2.3.

Here, however, we are interested in the question which information can be salvaged in cases where our knowledge about the camera configuration is incomplete. In the following sections, we will gradually assume less and less information about the camera parameters to be known. For each case, we will examine the damage to the precision with which we can still reconstruct the 3D structure of the scene. As will be seen, depending on what is still known about the camera setup, the geometric uncertainty about the 3D reconstruction can range from a Euclidean motion up to a 3D projectivity.

2.5.1 Euclidean 3D Reconstruction

Let us first assume that we do not know about the camera positions and orientations relative to the world coordinate frame, but that we only know the position and orientation of the second camera relative to (the camera-centered reference frame of) the first one. We will also assume that both cameras are internally calibrated so that the matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are known as well. This case is relevant when, e.g., using a hand-held stereo rig and taking a single stereo image pair.

A moment's reflection shows that it is not possible to determine the world coordinates of $\mathtt{M}$ from the images in this case. Indeed, changing the position and orientation of the world frame does not alter the setup of the cameras in the scene, and consequently, does not alter the images. Thus it is impossible to recover absolute information about the cameras' external parameters in the real world from the projection Equation (2.20) alone, beyond what we already know (relative camera pose). Put differently, one cannot hope for more than to recover the

3D structure of the scene up to a 3D Euclidean transformation of the scene from the projection Equation (2.20) alone.

On the other hand, the factor $\mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$ in the right-hand side of the first projection equation in formula (2.20) is just a Euclidean transformation of the scene. Thus, without loss of generality, we may replace this factor by $\mathtt{M}'$, because we just lost all hope of retrieving the 3D coordinates of $\mathtt{M}$ more precisely than up to some unknown 3D Euclidean transformation anyway. The first projection equation then simplifies to $\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathtt{M}'$. Solving $\mathtt{M}' = \mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$ for $\mathtt{M}$ gives $\mathtt{M} = \mathbf{R}_1 \mathtt{M}' + \mathtt{C}_1$, and substituting this into the second projection equation in (2.20) yields $\rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathtt{M}' + \mathbf{K}_2 \mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$. Together,

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathtt{M}' \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathtt{M}' + \mathbf{K}_2 \mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2) \tag{2.21}$$

constitute a system of equations which allow to recover the 3D structure of the scene *up to a 3D Euclidean transformation* $\mathtt{M}' = \mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$. Indeed, as the cameras are assumed to be internally calibrated, $\mathbf{K}_1$ and $\mathbf{K}_2$ are known and the first equation in (2.21) can be solved for $\mathtt{M}'$, viz. $\mathtt{M}' = \rho_1 \mathbf{K}_1^{-1} \mathtt{m}_1$. Plugging this new expression for $\mathtt{M}'$ into the right-hand side of the second equation in formula (2.21), one gets

$$\rho_2 \mathtt{m}_2 = \rho_1 \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1} \mathtt{m}_1 + \mathbf{K}_2 \mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2). \tag{2.22}$$

Notice that this actually brings us back to the epipolar relation (2.16) which was derived in Section 2.4.1. In this equation $\mathbf{R}_2^T \mathbf{R}_1$ and $\mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$ represent, respectively, the relative orientation and the relative position of the first camera with respect to the second one. As these are assumed to be known too, Equation (2.22) yields a system of three linear equations from which the two unknown scalar factors ρ_1 and ρ_2 can be computed. And, when ρ_1 is found, the Euclidean transformation $\mathtt{M}'$ of the scene point $\mathtt{M}$ is found as well.

In summary, *if the cameras are internally calibrated and the relative position and orientation of the cameras is known, then for each pair of corresponding points* $\mathtt{m}_1$ *and* $\mathtt{m}_2$ *in the images the Euclidean transformation* $\mathtt{M}'$ *of the underlying scene point* $\mathtt{M}$ *can be recovered from the relations* (2.21). Formula (2.21) is therefore referred to as *a system of Euclidean reconstruction equations* for the scene and the 3D points $\mathtt{M}'$ satisfying these equations constitute *a Euclidean reconstruction* of

the scene. As a matter of fact, better than Euclidean reconstruction is often not needed, as the 3D shape of the objects in the scene is perfectly retrieved, only not their position relative to the world coordinate frame, which is completely irrelevant in many applications.

Notice how we have absorbed the unknown parameters into the new coordinates $\mathtt{M}'$. This is a strategy that will be used repeatedly in the next sections. It is interesting to observe that $\mathtt{M}' = \mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$ are, in fact, the coordinates of the scene point $\mathtt{M}$ with respect to the camera-centered reference frame of the first camera, as was calculated in Section 2.2.4. The Euclidean reconstruction Equation (2.21) thus coincides with the system of projection Equation (2.20) if the world frame is the camera-centered reference frame of the first camera (i.e., $\mathtt{C}_1 = \mathtt{0}$ and $\mathbf{R}_1 = \mathbf{I_3}$) and the rotation matrix $\mathbf{R}_2$ then expresses the relative orientation of the second camera with respect to the first one.

2.5.2 Metric 3D Reconstruction

Next consider a stereo setup as that of the previous section, but suppose that we do not know the distance between the centers of projection $\mathtt{C}_1$ and $\mathtt{C}_2$ any more. We do, however, still know the relative orientation of the cameras and the direction along which the second camera is shifted with respect to the first one. This means that $\mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$ is only known up to a non-zero scalar factor. As the cameras still are internally calibrated, the calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are known and it follows that the last term in the second of the Euclidean reconstruction Equation (2.21), viz. $\mathbf{K}_2\mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$, can only be determined up to an unknown scalar factor. According to formula (2.15), $\mathbf{K}_2\mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2) = \rho_{e2}\,\mathtt{e}_2$ yields the epipole $\mathtt{e}_2$ of the first camera in the second image. Knowledge of the direction in which the second camera is shifted with respect to the first one, combined with knowledge of $\mathbf{K}_2$, clearly also allows to determine $\mathtt{e}_2$. It is interesting to notice, however, that, if a sufficient of corresponding points can be found between the images, then the fundamental matrix of the image pair can be computed, up to a non-zero scalar factor, and the epipole $\mathtt{e}_2$ can also be recovered that way (as explained in item 2 of Section 2.4.2 and, in more detail, in Section 4). Having a sufficient number of point

correspondences would thus eliminate the need to know about the direction of camera shift beforehand. On the other hand, the assumption that the inter-camera distance is not known implies that the scalar factor ρ_{e2} in the last term of the Euclidean reconstruction equations

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathtt{M}' \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathtt{M}' + \rho_{e2} \mathtt{e}_2 \tag{2.23}$$

is unknown. This should not come as a surprise, since ρ_{e2} is the projective depth of $\mathtt{C}_1$ in the second camera, and thus it is directly related to the inter-camera distance. Algebraically, the six homogeneous equations do not suffice to solve for the six unknowns $\mathtt{M}'$, ρ_1, ρ_2, and ρ_{e2}. One only gets a solution up to an unknown scale. But, using the absorption trick again, we introduce the new coordinates $\bar{\mathtt{M}} = \frac{1}{\rho_{e2}} \mathtt{M}' = \frac{1}{\rho_{e2}} \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1)$, which is a 3D similarity transformation of the original scene. Formulas (2.23) then reduces to

$$\bar{\rho}_1 \mathtt{m}_1 = \mathbf{K}_1 \bar{\mathtt{M}} \quad \text{and} \quad \bar{\rho}_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \bar{\mathtt{M}} + \mathtt{e}_2, \tag{2.24}$$

where $\bar{\rho}_1 = \frac{\rho_1}{\rho_{e2}}$ and $\bar{\rho}_2 = \frac{\rho_2}{\rho_{e2}}$ are scalar factors expressing the projective depth of the scene point underlying $\mathtt{m}_1$ and $\mathtt{m}_2$ in each camera relative to the scale ρ_{e2} of the metric reconstruction of the scene. The coordinates $\bar{\mathtt{M}}$ provide a 3D reconstruction of the scene point $\mathtt{M}$ up to an unknown 3D similarity, as expected.

We could have seen this additional scaling issue coming also intuitively. If one were to scale a scene together with the cameras in it, then this would have no impact on the images. In terms of the relative camera positions, this would only change the distance between them, not their relative orientations or the relative direction in which one camera is displaced with respect to the other. The calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ would remain the same, since both the focal lengths and the pixel sizes are supposed to be scaled by the same factor and the number of pixels in the image is kept the same as well, so that the offsets in the calibration matrices do not change. Again, as such changes are not discernible in the images, having internally calibrated cameras and external calibration only up to the exact distance between the cameras leaves us with one unknown, but fixed, scale factor. Together with the unknown Euclidean motion already present in the 3D reconstruction derived in the previous section, this unknown scaling brings the

geometric uncertainty about the 3D scene up to an unknown *3D similarity transformation*. Such a reconstruction of the scene is commonly referred to in the computer vision literature as a *metric reconstruction* of the scene, and formula (2.24) is referred to as *a system of metric reconstruction equations*. Although annoying, it should be noted that fixing the overall unknown scale usually is the least of our worries in practice, as indeed knowledge about a single distance or length in the scene suffices to lift the uncertainty about scale.

2.5.3 Affine 3D Reconstruction

A further step toward our goal of 3D reconstruction from the image data alone is to give up on knowledge of the internal camera parameters as well. For the metric reconstruction Equation (2.24) this implies that the calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are also unknown. As before, one can perform a change of coordinates $\tilde{\mathtt{M}} = \mathbf{K}_1 \bar{\mathtt{M}}$ and replace $\bar{\mathtt{M}}$ in the reconstruction Equation (2.24) by $\bar{\mathtt{M}} = \mathbf{K}_1^{-1} \tilde{\mathtt{M}}$. This gives:

$$\tilde{\rho}_1 \mathtt{m}_1 = \tilde{\mathtt{M}} \quad \text{and} \quad \tilde{\rho}_2 \mathtt{m}_2 = \mathbf{A} \tilde{\mathtt{M}} + \mathtt{e}_2, \tag{2.25}$$

where $\tilde{\rho}_1 = \bar{\rho}_1$, $\tilde{\rho}_2 = \bar{\rho}_2$, and $\mathbf{A} = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1}$ is the infinite homography introduced in Section 2.4.1. If the invertible matrix $\mathbf{A}$ is known, then this system (2.25) can be solved for the scalars $\tilde{\rho}_1$, $\tilde{\rho}_2$, and, more importantly, for $\tilde{\mathtt{M}}$, as in the metric case (cf. Section 2.5.2). More on how to extract $\mathbf{A}$ from image information only is to follow shortly. As $\tilde{\mathtt{M}} = \mathbf{K}_1 \bar{\mathtt{M}} = \frac{1}{\rho_{e2}} \mathbf{K}_1 \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1)$ represents a 3D affine transformation of the world space, formula (2.25) is referred to as *a system of affine reconstruction equations* for the scene and the 3D points $\tilde{\mathtt{M}}$ satisfying these equations constitute *an affine reconstruction* of the scene, i.e., a reconstruction which is correct up to an unknown 3D affine transformation.

It suffices to know $\mathbf{A}$ and $\mathtt{e}_2$ in order to compute an affine reconstruction of the scene. As explained in Section 2.4.2 and in more detail in Section 4, $\mathtt{e}_2$ can be extracted from $\mathbf{F}$, and $\mathbf{F}$ can be derived — up to a non-zero scalar factor — from corresponding points between the images. Since $\mathtt{e}_2$ is in the left nullspace of $\mathbf{F}$, an unknown scalar factor on $\mathbf{F}$ does not prevent the extraction of $\mathtt{e}_2$, however. Unfortunately, determining $\mathbf{A}$ is not that easy in practice. $\mathbf{A}$ was defined as

$\mathbf{A} = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1}$, where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the calibration matrices and $\mathbf{R}_2^T \mathbf{R}_1$ represents the relative orientation of the cameras. If this information about the cameras is not available, then this formula cannot be used to compute $\mathbf{A}$. On the other hand, the fundamental matrix $\mathbf{F}$ of the image pair has been defined in Section 2.4.1 as $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}$. But, unfortunately, the relation $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}$ does not define the matrix $\mathbf{A}$ uniquely. Indeed, suppose $\mathbf{A}_1$ and $\mathbf{A}_2$ are 3×3-matrices such that $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}_1$ and $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}_2$. Then $[\mathtt{e}_2]_\times (\mathbf{A}_1 - \mathbf{A}_2) = 0$. As $[\mathtt{e}_2]_\times$ is the skew-symmetric 3×3-matrix which represents the cross product with the three-vector $\mathtt{e}_2$; i.e., $[\mathtt{e}_2]_\times \mathtt{v} = \mathtt{e}_2 \times \mathtt{v}$ for all $\mathtt{v} \in \mathbb{R}^3$, the columns of $\mathbf{A}_1$ and $\mathbf{A}_2$ can differ by a scalar multiple of $\mathtt{e}_2$. In particular, $\mathbf{A}_1 = \mathbf{A}_2 + \mathtt{e}_2 \mathtt{a}^T$ for some three-vector $\mathtt{a} \in \mathbb{R}^3$. Hence, the infinite homography $\mathbf{A}$ cannot be recovered from point correspondences between the images alone.

So, what other image information can then be used to determine $\mathbf{A}$? Recall from Section 2.4.3 that the matrix $\mathbf{A}$ transfers vanishing points of directions in the scene from the first image to the second one. Each pair of corresponding vanishing points in the images therefore yields constraints on the infinite homography $\mathbf{A}$. More precisely, if $\mathtt{v}_1$ and $\mathtt{v}_2$ are the vanishing points in, respectively, the first and the second image of a particular direction in the scene, then $\rho \mathtt{v}_2 = \mathbf{A} \mathtt{v}_1$ for some non-zero scalar factor ρ. Since $\mathbf{A}$ is a 3×3-matrix and each such constraint brings three equations, but also one additional unknown ρ, at least four such constraints are needed to determine the matrix $\mathbf{A}$ up to a scalar factor ρ_A. This unknown factor does not form an obstacle for the obtained matrix to be used in the affine reconstruction Equation (2.25), because by multiplying the first reconstruction equation with the same factor and then absorbing it into $\tilde{\mathtt{M}}$ in the right-hand side of both equations, one still obtains an affine 3D reconstruction of the scene.

Identifying the vanishing points of four independent directions in an image pair is rarely possible. More often, one has three dominant directions, typically orthogonal to each other. This is the case for many built-up environments. Fortunately, there is one direction which is always available, namely the line passing through the positions $\mathtt{C}_1$ and $\mathtt{C}_2$ in the scene. The vanishing points of this line in the images are the

intersection of the line with each image plane. But these are just the epipoles. So, the epipoles $\mathtt{e}_1$ and $\mathtt{e}_2$ in a pair of images are corresponding vanishing points of the direction of the line through the camera positions $\mathtt{C}_1$ and $\mathtt{C}_2$ in the scene and therefore satisfy the relation

$$\rho_e \, \mathtt{e}_2 = \mathbf{A} \, \mathtt{e}_1 \quad \text{for some } \rho_e \in \mathbb{R},$$

as we have already noted in Section 2.4.2, third item. Consequently, *if the vanishing points of three independent directions in the scene can be identified in the images, then the infinite homography* $\mathbf{A}$ *can be computed up to a non-zero scalar factor*; at least if none of the three directions is that of the line connecting the centers of projection $\mathtt{C}_1$ and $\mathtt{C}_2$.

Notice that we have only absorbed $\mathbf{K}_1$ into the affine coordinates $\tilde{\mathtt{M}}$ of the 3D reconstruction, but that $\mathbf{K}_2$ also can remain unknown, as it is absorbed by $\mathbf{A}$.

2.5.4 Projective 3D Reconstruction

Finally, we have arrived at the situation where we assume no knowledge about the camera configuration or about the scene whatsoever. Instead, we only assume that one can find point correspondences between the images and extract the fundamental matrix of the image pair.

The main conclusion of the previous section is that, if no information about the internal and external parameters of the camera is available and if insufficient vanishing points can be found and matched, then the only factor that separates us from an affine 3D reconstruction of the scene is the infinite homography $\mathbf{A}$. Let us therefore investigate whether some partial knowledge about $\mathbf{A}$ can still be retrieved from general point correspondences between the images.

Recall from Section 2.4.1 that $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}$ and that the epipole $\mathtt{e}_2$ can uniquely be determined from the fundamental matrix $\mathbf{F}$. It is not difficult to verify that $([\mathtt{e}_2]_\times)^3 = -\|\mathtt{e}_2\|^2 [\mathtt{e}_2]_\times$, where $\|\mathtt{e}_2\|$ denotes the norm of the three-vector $\mathtt{e}_2$. As $\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}$, it follows that

$$[\mathtt{e}_2]_\times ([\mathtt{e}_2]_\times \mathbf{F}) = ([\mathtt{e}_2]_\times)^3 \mathbf{A} = -\|\mathtt{e}_2\|^2 [\mathtt{e}_2]_\times \mathbf{A} = -\|\mathtt{e}_2\|^2 \mathbf{F}.$$

So, $[\mathtt{e}_2]_\times \mathbf{F}$ is a 3×3-matrix which when premultiplied with $[\mathtt{e}_2]_\times$ yields a non-zero scalar multiple of the fundamental matrix $\mathbf{F}$. In other words, up to a non-zero scalar factor, the 3×3-matrix $[\mathtt{e}_2]_\times \mathbf{F}$

could be a candidate for the unknown matrix $\mathbf{A}$. Unfortunately, as both the fundamental matrix $\mathbf{F}$ and $[\mathtt{e}_2]_\times$ have rank 2, the matrix $[\mathtt{e}_2]_\times\mathbf{F}$ is not invertible as $\mathbf{A}$ ought to be. But, recall from the previous section that two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ satisfying $\mathbf{F} = [\mathtt{e}_2]_\times\mathbf{A}_1$ and $\mathbf{F} = [\mathtt{e}_2]_\times\mathbf{A}_2$ are related by $\mathbf{A}_1 = \mathbf{A}_2 + \mathtt{e}_2\,\mathtt{a}^T$ for some three-vector $\mathtt{a} \in \mathbb{R}^3$. This implies that the unknown matrix $\mathbf{A}$ must be of the form $\mathbf{A} = -(1/\|\mathtt{e}_2\|^2)\,[\mathtt{e}_2]_\times\mathbf{F} + \mathtt{e}_2\,\mathtt{a}^T$ for some three-vector $\mathtt{a} \in \mathbb{R}^3$. It follows that the invertible matrix $\mathbf{A}$, needed for an affine reconstruction of the scene, can only be recovered up to three unknown components of $\mathtt{a}$. As we do not know them, the simplest thing to do is to put them to zero or to make a random guess. This section analyzes what happens to the reconstruction if we do just that.

The expression for $\mathbf{A}$ only takes on the particular form given above in case $\mathbf{F}$ is obtained from camera calibration (i.e., from the rotation and calibration matrices). In case $\mathbf{F}$ is to be computed from point correspondences — as is the case here — it can only be determined up to a non-zero scalar factor. Let $\hat{\mathbf{F}}$ be an estimate of the fundamental matrix as obtained from point correspondences, then $\mathbf{F} = \kappa\hat{\mathbf{F}}$ for some non-zero scalar factor κ. Now define $\hat{\mathbf{A}} = -(1/\|\mathtt{e}_2\|^2)\,[\mathtt{e}_2]_\times\hat{\mathbf{F}}$. Then $\mathbf{A} = \kappa\,\hat{\mathbf{A}} + \mathtt{e}_2\,\mathtt{a}^T$ for some unknown three-vector $\mathtt{a} \in \mathbb{R}^3$. Notice that, as observed before, the scalar factor κ between $\mathbf{F}$ and $\hat{\mathbf{F}}$ has no influence on the pixel coordinates of $\mathtt{e}_2$, as derived from $\hat{\mathbf{F}}$ instead of $\mathbf{F}$. Using $\hat{\mathbf{A}}$ for $\mathbf{A}$ in the affine reconstruction equations (2.25), for corresponding image points $\mathtt{m}_1$ and $\mathtt{m}_2$ we now solve the following system of linear equations:

$$\hat{\rho}_1\,\mathtt{m}_1 = \hat{\mathtt{M}} \quad \text{and} \quad \hat{\rho}_2\,\mathtt{m}_2 = \hat{\mathbf{A}}\,\hat{\mathtt{M}} + \mathtt{e}_2, \tag{2.26}$$

where $\hat{\rho}_1$ and $\hat{\rho}_2$ are non-zero scalar factors, and where the 3D points $\hat{\mathtt{M}}$ constitute a 3D reconstruction of the scene, which — as will be demonstrated next — differs from the original scene by a (unique, but unknown) 3D projective transformation. The set of 3D points $\hat{\mathtt{M}}$ is called a *projective 3D reconstruction* of the scene and formula (2.26) is referred to as *a system of projective reconstruction equations*. Figure 2.13 summarizes the steps that have lead up to these equations.

In order to prove that the points $\hat{\mathtt{M}}$ obtained from Equation (2.26) do indeed constitute a projective 3D reconstruction of the scene,

Projective 3D reconstruction from two uncalibrated images

***Given*:** A set of point correspondences $\mathtt{m}_1 \in \mathcal{I}_1$ and $\mathtt{m}_2 \in \mathcal{I}_2$ between two uncalibrated images $\mathcal{I}_1$ and $\mathcal{I}_2$ of a static scene

***Objective*:** A projective 3D reconstruction $\hat{\mathtt{M}}$ of the scene

***Algorithm*:**

(1) Compute an estimate $\hat{\mathbf{F}}$ for the fundamental matrix[a]
(2) Compute[b] the epipole $\mathtt{e}_2$ from $\hat{\mathbf{F}}$
(3) Compute the 3×3-matrix $\hat{\mathbf{A}} = -(1 / \|\mathtt{e}_2\|^2)\,[\mathtt{e}_2]_\times \hat{\mathbf{F}}$
(4) For each pair of corresponding image points $\mathtt{m}_1$ and $\mathtt{m}_2$, solve the following system of linear equations for $\hat{\mathtt{M}}$:

$$\hat{\rho}_1\,\mathtt{m}_1 = \hat{\mathtt{M}} \qquad \text{and} \qquad \hat{\rho}_2\,\mathtt{m}_2 = \hat{\mathbf{A}}\,\hat{\mathtt{M}} + \mathtt{e}_2$$

($\hat{\rho}_1$ and $\hat{\rho}_2$ are non-zero scalars)

[a] Cf. Section 2.4.2. See also Sections 4.2.2 and 4.2.3 in Section 4.
[b] Cf. Section 2.4.2. See also Section 4.3in Section 4.

Fig. 2.13 A basic algorithm for projective 3D reconstruction from two uncalibrated images.

we first express the Equation (2.26) in terms of projection matrices and extended coordinates $\binom{\hat{\mathtt{M}}}{1} = (\hat{X}, \hat{Y}, \hat{Z}, 1)^T$ for the 3D point $\hat{\mathtt{M}} = (\hat{X}, \hat{Y}, \hat{Z})^T$ (cf. formula (2.7) in Section 2.2.4):

$$\hat{\rho}_1\,\mathtt{m}_1 = (\mathbf{I}_3 \mid \mathtt{0}) \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} \quad \text{and} \quad \hat{\rho}_2\,\mathtt{m}_2 = (\hat{\mathbf{A}} \mid \mathtt{e}_2) \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix}. \tag{2.27}$$

Similarly, the affine reconstruction Equation (2.25) are written as:

$$\tilde{\rho}_1\,\mathtt{m}_1 = (\mathbf{I}_3 \mid \mathtt{0}) \begin{pmatrix} \tilde{\mathtt{M}} \\ 1 \end{pmatrix} \quad \text{and} \quad \tilde{\rho}_2\,\mathtt{m}_2 = (\mathbf{A} \mid \mathtt{e}_2) \begin{pmatrix} \tilde{\mathtt{M}} \\ 1 \end{pmatrix}, \tag{2.28}$$

where $\binom{\tilde{\mathtt{M}}}{1} = (\tilde{X}, \tilde{Y}, \tilde{Z}, 1)^T$ are the extended coordinates of the 3D point $\tilde{\mathtt{M}} = (\tilde{X}, \tilde{Y}, \tilde{Z})^T$. Recall that the invertible matrix $\mathbf{A}$ is of the form

$\mathbf{A} = \kappa\,\hat{\mathbf{A}} + \mathtt{e}_2\,\mathtt{a}^T$ for some non-zero scalar κ and three-vector $\mathtt{a} \in \mathbb{R}^3$. The last equality in Equation (2.28) therefore is:

$$\tilde{\rho}_2\,\mathtt{m}_2 = (\kappa\,\hat{\mathbf{A}} + \mathtt{e}_2\,\mathtt{a}^T \mid \mathtt{e}_2)\begin{pmatrix}\tilde{\mathtt{M}}\\ 1\end{pmatrix} = (\hat{\mathbf{A}} \mid \mathtt{e}_2)\begin{pmatrix}\kappa\mathbf{I}_3 & \mathtt{0}\\ \mathtt{a}^T & 1\end{pmatrix}\begin{pmatrix}\tilde{\mathtt{M}}\\ 1\end{pmatrix}; \qquad (2.29)$$

and the first equality in Equation (2.28) can be rewritten as:

$$\kappa\,\tilde{\rho}_1\,\mathtt{m}_1 = (\mathbf{I}_3 \mid \mathtt{0})\begin{pmatrix}\kappa\mathbf{I}_3 & \mathtt{0}\\ \mathtt{a}^T & 1\end{pmatrix}\begin{pmatrix}\tilde{\mathtt{M}}\\ 1\end{pmatrix}.$$

Comparing these expressions to formula (2.27), it follows that

$$\lambda\begin{pmatrix}\hat{\mathtt{M}}\\ 1\end{pmatrix} = \begin{pmatrix}\kappa\mathbf{I}_3 & \mathtt{0}\\ \mathtt{a}^T & 1\end{pmatrix}\begin{pmatrix}\tilde{\mathtt{M}}\\ 1\end{pmatrix}. \qquad (2.30)$$

for some non-zero scalar $\lambda \in \mathbb{R}$ and that $\hat{\rho}_1 = (\kappa/\lambda)\,\tilde{\rho}_1$ and $\hat{\rho}_2 = (1/\lambda)\,\tilde{\rho}_2$. Notice that one cannot simply go for a solution with $\lambda = 1$, as this implies that $\mathtt{a}^T = \mathtt{0}^T$ and therefore $\mathbf{A} = \kappa\hat{\mathbf{A}}$. This would mean that we have been lucky enough to guess the correct infinite homography matrix $\mathbf{A}$ (up to a scalar factor) right away. But this cannot be the case according to our proposed choice: $\hat{\mathbf{A}}$ has rank 2 and thus is singular whereas the correct matrix $\mathbf{A}$ is invertible. Rather, eliminating λ from Equation (2.30) gives:

$$\hat{\mathtt{M}} = \frac{\kappa\tilde{\mathtt{M}}}{\mathtt{a}^T\tilde{\mathtt{M}} + 1}.$$

Recall from Section 2.5.3 that $\tilde{\mathtt{M}} = \frac{1}{\rho_{e2}}\mathbf{K}_1\,\mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$. Substituting this into the previous equation, one sees that

$$\hat{\mathtt{M}} = \frac{\kappa\mathbf{K}_1\,\mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)}{\mathtt{a}^T\mathbf{K}_1\,\mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1) + \rho_{e2}}$$

is a projective transformation of the scene. Moreover, since $\tilde{\mathtt{M}} = \mathbf{K}_1\bar{\mathtt{M}}$ with $\bar{\mathtt{M}} = \frac{1}{\rho_{e2}}\mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$ being the metric reconstruction of the scene defined in Section 2.5.2, formula (2.30) can also be written as:

$$\lambda\begin{pmatrix}\hat{\mathtt{M}}\\ 1\end{pmatrix} = \begin{pmatrix}\kappa\mathbf{K}_1 & \mathtt{0}\\ \mathtt{a}^T\mathbf{K}_1 & 1\end{pmatrix}\begin{pmatrix}\bar{\mathtt{M}}\\ 1\end{pmatrix}; \qquad (2.31)$$

or, after elimination of λ,

$$\hat{\mathtt{M}} = \frac{\kappa \mathbf{K}_1 \bar{\mathtt{M}}}{\mathtt{a}^T \mathbf{K}_1 \bar{\mathtt{M}} + 1},$$

which shows that $\hat{\mathtt{M}}$ also is a projective transformation of $\bar{\mathtt{M}}$.

2.5.5 Taking Stock — Stratification

The goal put forward at the beginning of Section 2.5 was to recover the 3D geometrical structure of a scene from two images of it, without necessarily having complete information about the internal and external parameters of the cameras. It was immediately seen that one can only recover it up to a 3D Euclidean transformation if only information about the relative camera poses is available for the external camera parameters. If the precise inter-camera distance is also unknown, 3D reconstruction is only possible up to a 3D similarity and the 3D reconstruction is said to be metric. Moreover, if the calibration matrix of the first camera is unknown (and also of the second camera for that matter), then at most an affine 3D reconstruction of the scene is feasible. And, if the infinite homography $\mathbf{A}$ introduced in Section 2.4.1 is unknown, then the scene can only be reconstructed up to a 3D projective transformation. The latter case is very relevant, as it applies to situations where the internal and external parameters of the camera pair are unknown, but where point correspondences can be found.

Figure 2.14 illustrates these different situations. In the figure the scene consists of a cube. In a metric reconstruction a cube is found, but the actual size of the cube is undetermined. In an affine reconstruction the original cube is reconstructed as a parallelepiped. Affine transformations preserve parallelism, but they do not preserve metric relations such as angles and relative lengths. In a projective reconstruction the scene appears as an (irregular) hexahedron, because projective transformations only preserve incidence relations such as collinearity and coplanarity, but parallelism or any metric information is not preserved. Table 2.1 shows the mathematical expressions that constitute these geometrical transformations. The mutual relations between the different types of 3D reconstruction are often referred to as *stratification* of the

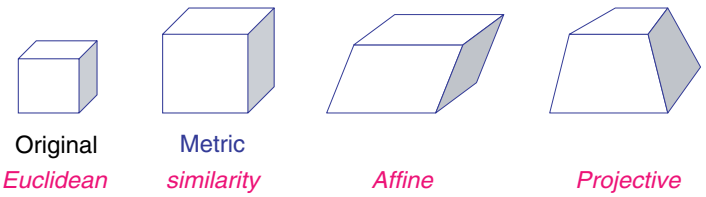

Fig. 2.14 The aim of passive 3D reconstruction is to recover the 3D geometrical structure of the scene from images. With a fully calibrated setup of two cameras a Euclidean reconstruction can be achieved. The reconstruction degenerates to metric if the inter-camera distance (sometimes referred to as baseline) is unknown. If the calibration matrices of the cameras are unknown, then the scene structure can only be recovered up to a 3D affine transformation; and, if no information about the camera setup is available, then only a projective reconstruction is feasible.

Table 2.1. The stratification of geometries.

Geometrical transf.	Mathematical expression
Euclidean transf.	$\mathtt{M}' = \mathbf{R}\mathtt{M} + \mathtt{T}$ with $\mathbf{R}$ a rotation matrix, $\mathtt{T} \in \mathbb{R}^3$
similarity transform.	$\mathtt{M}' = \kappa\mathbf{R}\mathtt{M} + \mathtt{T}$ with $\mathbf{R}$ a rotation matrix, $\mathtt{T} \in \mathbb{R}^3$, $\kappa \in \mathbb{R}$
affine transf.	$\mathtt{M}' = \mathbf{Q}\mathtt{M} + \mathtt{T}$ with $\mathbf{Q}$ an invertible matrix, $\mathtt{T} \in \mathbb{R}^3$
projective transf.	$\begin{cases} X' = \frac{p_{11}X+p_{12}Y+p_{13}Z+p_{14}}{p_{41}X+p_{42}Y+p_{43}Z+p_{44}} \\ Y' = \frac{p_{21}X+p_{22}Y+p_{23}Z+p_{24}}{p_{41}X+p_{42}Y+p_{43}Z+p_{44}} \\ Z' = \frac{p_{31}X+p_{32}Y+p_{33}Z+p_{34}}{p_{41}X+p_{42}Y+p_{43}Z+p_{44}} \end{cases}$ with $\mathbf{P} = (p_{ij})$ an invertible 4×4-matrix

geometries. This term reflects that the transformations higher up in the list are special types (subgroups) of the transformations lower down. Obviously, the uncertainty about the reconstruction increases when going down the list, as also corroborated by the number of degrees of freedom in these transformations.

In the first instance, the conclusion of this section — namely that without additional information about the (internal and external) camera parameters the three-dimensional structure of the scene only can be recovered from two images up to an unknown projective transformation — might come as a disappointment. From a mathematical point of view, however, this should not come as a surprise, because algebraically passive 3D reconstruction boils down to solving the reconstruction Equations (2.20) for the camera parameters and the scene

points $\mathtt{M}$. In terms of projection matrices and extended coordinates (cf. formula (2.7) in Section 2.2.4), the reconstruction Equations (2.20) are formulated as:

$$\rho_1 \mathtt{m}_1 = \mathbf{P}_1 \begin{pmatrix} \mathtt{M} \\ 1 \end{pmatrix} \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{P}_2 \begin{pmatrix} \mathtt{M} \\ 1 \end{pmatrix}, \tag{2.32}$$

where $\mathbf{P}_j = (\mathbf{K}_j \mathbf{R}_j^T \mid -\mathbf{K}_j \mathbf{R}_j^T \mathtt{C}_j)$ is the 3×4-projection matrix of the j-th camera, with $j = 1$ or $j = 2$, and $\binom{\mathtt{M}}{1} = (X, Y, Z, 1)^T$ are the extended coordinates of the scene point $\mathtt{M} = (X, Y, Z)^T$. Moreover, it was observed in Section 2.2.4 that in the general linear camera model *any* 3×4-matrix of maximal rank can be interpreted as the projection matrix of a linear pinhole camera. Consequently, inserting an arbitrary invertible 4×4-matrix and its inverse in the right-hand sides of the projection Equations (2.32) does not alter the image points $\mathtt{m}_1$ and $\mathtt{m}_2$ in the left-hand sides of the equations and yields another — but equally valid — decomposition of the reconstruction equations:

$$\rho_1 \mathtt{m}_1 = \mathbf{P}_1 \mathbf{H}^{-1} \mathbf{H} \begin{pmatrix} \mathtt{M} \\ 1 \end{pmatrix} \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{P}_2 \mathbf{H}^{-1} \mathbf{H} \begin{pmatrix} \mathtt{M} \\ 1 \end{pmatrix};$$

or equivalently,

$$\hat{\rho}_1 \mathtt{m}_1 = \hat{\mathbf{P}}_1 \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} \quad \text{and} \quad \hat{\rho}_2 \mathtt{m}_2 = \hat{\mathbf{P}}_2 \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix}, \tag{2.33}$$

with $\hat{\mathbf{P}}_1 = \mathbf{P}_1 \mathbf{H}^{-1}$ and $\hat{\mathbf{P}}_2 = \mathbf{P}_2 \mathbf{H}^{-1}$ two 3×4-matrices of maximal rank, $\lambda \binom{\hat{\mathtt{M}}}{1} = \mathbf{H} \binom{\mathtt{M}}{1}$ with λ a non-zero scalar a 3D projective transformation of the scene, and $\hat{\rho}_1 = \frac{\rho_1}{\lambda}$ and $\hat{\rho}_2 = \frac{\rho_2}{\lambda}$ non-zero scalar factors. Clearly, formulas (2.33) can be interpreted as the projection equations of scene points $\hat{\mathtt{M}}$ that, when observed by cameras with respective projection matrices $\hat{\mathbf{P}}_1$and $\hat{\mathbf{P}}_2$, yield the same set of image points $\mathtt{m}_1$ and $\mathtt{m}_2$. As $\mathbf{H}$ can be any invertible 4×4-matrix, it is clear that one cannot hope to do better than recovering the 3D geometric structure of the scene up to an arbitrary 3D projective transformation if no information about the cameras is available. But, the longer analysis presented in the previous sections and leading up to the same conclusion has provided an explicit algorithm for projective 3D reconstruction, which will be refined in the next sections.

Awareness of the stratification is also useful when additional information on the scene (rather than the cameras) is available. One may know some (relative) lengths or angles (including orthogonalities and parallelisms). Exploiting such information can make it possible to upgrade the geometric structure of the reconstruction to one with less uncertainty, i.e., to one higher up in the stratification table. Based on known lengths and angles, it may, for instance, become possible to construct a 3D projective transformation matrix (homography) $\mathbf{H}$ that converts the projective reconstruction $\hat{\mathtt{M}}$, obtained from the image data (cf. Figure 2.13 and Equation (2.31)), directly into a Euclidean one. And, even if no metric information about the scene is available, other geometrical relations that are known to exist in the scene may be useful. In particular, we have already discussed the case of parallel lines in three independent directions, that suffice to upgrade a projective reconstruction into an affine one through the three vanishing points they deliver. Indeed, they allow to determine the three unknown parameters $\mathtt{a} \in \mathbb{R}^3$ of the invertible matrix $\mathbf{A}$ (cf. Section 2.5.3). Knowing $\mathtt{a}$ allows to upgrade the projective 3D reconstruction $\hat{\mathtt{M}}$ to the affine 3D reconstruction $\kappa\tilde{\mathtt{M}}$ by using Equation (2.30) in Section 2.5.4. However, one does not always need the projections of (at least) two parallel lines in the image to compute a vanishing point. Alternatively, if in an image three points can be identified that are the projections of collinear scene points $\mathtt{M}_1$, $\mathtt{M}_2$, and $\mathtt{M}_3$ of which the ratio $\frac{d(\mathtt{M}_1,\mathtt{M}_2)}{d(\mathtt{M}_1,\mathtt{M}_3)}$ of their Euclidean distances in the real world is known (e.g., three equidistant points in the scene), then one can determine the vanishing point of this direction in the image using the cross ratio (cf. Appendix A). In Section 4.6.1 of Section 4 such possibilities for improving the 3D reconstruction will be explored further. In the next section, however, we will assume that no information about the scene is available and we will investigate how more than two images can contribute to better than a projective 3D reconstruction.

2.6 From Projective to Metric Using More Than Two Images

In the geometric stratification of the previous section, we ended up with a 3D projective reconstruction in case no prior camera calibration

information is available whatsoever and we have to work purely from image correspondences. In this section we will investigate how and when we can work our way up to a metric 3D reconstruction, if we were to have more than just two uncalibrated images.

2.6.1 Projective Reconstruction and Projective Camera Matrices from Multiple Images

Suppose we are given m images $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_m$ of a static scene and a set of corresponding points $\mathtt{m}_j \in \mathcal{I}_j$ between the images ($j \in \{1, 2, \ldots, m\}$). As in formula (2.20) the projection equations of the j-th camera are:

$$\rho_j \, \mathtt{m}_j = \mathbf{K}_j \, \mathbf{R}_j^T (\mathtt{M} - \mathtt{C}_j) \quad \text{for } j \in \{1, 2, \ldots, m\}; \tag{2.34}$$

or, in terms of projection matrices and extended coordinates as in formula (2.32):

$$\rho_j \, \mathtt{m}_j = \left(\mathbf{K}_j \mathbf{R}_j^T \mid -\mathbf{K}_j \mathbf{R}_j^T \mathtt{C}_j \right) \begin{pmatrix} \mathtt{M} \\ 1 \end{pmatrix} \quad \text{for } j \in \{1, 2, \ldots, m\}. \tag{2.35}$$

Of course, when one has more than just two views, one can still extract at least a projective reconstruction of the scene, as when one only had two. Nonetheless, a note of caution is in place here. If we were to try and build a projective reconstruction by pairing the first with each of the other images separately and then combine the resulting projective reconstructions, this would in general not work. Indeed, one has to ensure that the same projective distortion is obtained for each of the reconstructions. That this will not automatically amount from a pairwise reconstruct-and-then-combine procedure becomes clear if one writes down the formulas explicitly. When the internal and external parameters of the cameras are unknown, a projective 3D reconstruction of the scene can be computed from the first two images by the procedure of Figure 2.13 in section 2.5.4. In particular, for each point correspondence $\mathtt{m}_1 \in \mathcal{I}_1$ and $\mathtt{m}_2 \in \mathcal{I}_2$ in the first two images, the reconstructed 3D point $\hat{\mathtt{M}}$ is the solution of the system of linear equations:

$$\hat{\rho}_1 \, \mathtt{m}_1 = \hat{\mathtt{M}} \quad \text{and} \quad \hat{\rho}_2 \, \mathtt{m}_2 = \hat{\mathbf{A}}_2 \hat{\mathtt{M}} + \mathtt{e}_2, \tag{2.36}$$

where $\hat{\rho}_1$ and $\hat{\rho}_2$ are non-zero scalars (which depend on $\hat{\mathtt{M}}$ and thus are also unknown) and $\hat{\mathbf{A}}_2 = (-1/\|\mathtt{e}_2\|^2)\,[\mathtt{e}_2]_\times \hat{\mathbf{F}}_{12}$ with $\hat{\mathbf{F}}_{12}$ an estimate of

the fundamental matrix between the first two images as computed from point correspondences. The resulting points $\hat{\mathtt{M}}$ constitute a 3D reconstruction of the scene which relates to the metric reconstruction $\bar{\mathtt{M}}$ by the projective transformation:

$$\lambda \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \kappa \mathbf{K}_1 & \mathtt{0} \\ \mathtt{a}^T \mathbf{K}_1 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix}, \tag{2.37}$$

as was demonstrated in Section 2.5.4 (formula (2.31)). Let $\mathbf{H}$ be the homography matrix in the right-hand side of formula (2.37), viz.:

$$\mathbf{H} = \begin{pmatrix} \kappa \mathbf{K}_1 & \mathtt{0} \\ \mathtt{a}^T \mathbf{K}_1 & 1 \end{pmatrix}. \tag{2.38}$$

For the other images a similar set of equations can be derived by pairing each additional image with the first one. But, as mentioned before, one has to be careful in order to end up with the same projective distortion already resulting from the reconstruction based on the first two views. Indeed, for an additional image $\mathcal{I}_j$ with $j \geq 3$, blindly applying the procedure of Figure 2.13 in Section 2.5.4 to the first and the j-th images without considering the second image or the already obtained projectively reconstructed 3D points $\hat{\mathtt{M}}$ would indeed result in a 3D projective reconstruction $\hat{\mathtt{M}}_{(j)}$, but one which relates to the metric reconstruction $\bar{\mathtt{M}}$ by the projective transformation:

$$\lambda \begin{pmatrix} \hat{\mathtt{M}}_{(j)} \\ 1 \end{pmatrix} = \begin{pmatrix} \kappa_{(j)} \mathbf{K}_1 & \mathtt{0} \\ \mathtt{a}_{(j)}^T \mathbf{K}_1 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix},$$

in which the parameters $\kappa_{(j)} \in \mathbb{R}$ and $\mathtt{a}_{(j)} \in \mathbb{R}^3$ are not related to the parameters κ and $\mathtt{a}$ in the projective transformation (2.37) and the corresponding homography matrix $\mathbf{H}$ in formula (2.38). A consistent projective reconstruction implies that $\mathtt{a}_{(j)}^T/\kappa_{(j)} = \mathtt{a}^T/\kappa$. So, when introducing additional images $\mathcal{I}_j$ with $j \geq 3$ into the reconstruction process, one cannot simply choose the matrix $\hat{\mathbf{A}}_j$ for each new view independently, but one has to make sure that the linear system of reconstruction equations:

$$\begin{aligned} & \hat{\rho}_1 \mathtt{m}_1 = \hat{\mathtt{M}} \\ \text{and} \quad & \hat{\rho}_j \mathtt{m}_j = \hat{\mathbf{A}}_j \hat{\mathtt{M}} + \mathtt{e}_j \quad \text{for } j \in \{2, \ldots, m\} \end{aligned} \tag{2.39}$$

is consistent, i.e., that it is such that the solutions $\hat{\mathtt{M}}$ satisfy all reconstruction equations at once, in as far as image projections $\mathtt{m}_j$ are available. The correct way to proceed for the j-th image with $j \geq 3$ therefore is to express $\hat{\mathbf{A}}_j$ more generally as $\hat{\mathbf{A}}_j = \kappa_j \, (-1/\|\mathtt{e}_j\|^2) \, [\mathtt{e}_j]_\times \hat{\mathbf{F}}_{1j} + \mathtt{e}_j \, \mathtt{a}_j^T$ where $\kappa_j \in \mathbb{R}$ and $\mathtt{a}_j \in \mathbb{R}^3$ are parameters such that the reconstruction equations $\hat{\rho}_j \, \mathtt{m}_j = \hat{\mathbf{A}}_j \hat{\mathtt{M}} + \mathtt{e}_j$ hold for all reconstructed 3D points $\hat{\mathtt{M}}$ obtained from Equations (2.36). Each image point $\mathtt{m}_j$ brings three linear equations for the unknown parameters κ_j and $\mathtt{a}_j$, but also introduces 1 unknown scalar factor $\hat{\rho}_j$. Hence, theoretically 2 point correspondences between the first, the second and the j-th image would suffice to determine κ_j and $\mathtt{a}_j$ in a linear manner. However, the parameterization of the matrix $\hat{\mathbf{A}}_j$ relies on the availability of the fundamental matrix $\mathbf{F}_{1j}$ between the first and the j-th image. But, as explained in Section 2.4.2, if $\mathbf{F}_{1j}$ is to be estimated from point correspondences too, then at least 7 point correspondences are needed. Therefore, from a computational point of view it is better to estimate both $\hat{\mathbf{A}}$ and $\mathtt{e}_3$ from the relation $\hat{\rho}_j \, \mathtt{m}_j = \hat{\mathbf{A}}_j \hat{\mathtt{M}} + \mathtt{e}_j$ directly. Indeed, for each image point $\mathtt{m}_j$ this formula yields three linear equations in the nine unknown entries of the matrix $\hat{\mathbf{A}}_j$ and the two unknown pixel coordinates of the epipole $\mathtt{e}_j$, but it also introduces one unknown scalar factor $\hat{\rho}_j$. Consequently, at least 6 point correspondences are needed to uniquely determine $\hat{\mathbf{A}}_j$ and $\mathtt{e}_j$ in a linear manner. Moreover, since the fundamental matrix between the first and the j-th image is defined in Section 2.4.1 as $\mathbf{F}_{1j} = [\mathtt{e}_j]_\times \mathbf{A}_j$, an estimate for $\mathbf{A}_j$ and $\mathtt{e}_j$ immediately implies an estimate for $\mathbf{F}_{1j}$ as well. At first sight this may seem a better approach to estimate the fundamental matrix $\mathbf{F}_{1j}$, because it only needs 6 point correspondences instead of 7, but one should realize that three images are involved here (instead of only two images in Section 2.4.2). The relations between three or more views of a static scene will be further explored in Section 3. More information on how to efficiently compute $\hat{\mathbf{A}}_j$ in practice can be found in Section 4.4 of Section 4.

2.6.2 From Projective to Affine 3D Reconstruction

Now that the discussion of projective reconstruction is extended to the case of more than two cameras, we next consider options to go beyond

this and upgrade to an affine or even a metric reconstruction. But before doing so, we first make explicit the link between the plane at infinity of the scene and the projective transformation matrix:

$$\mathbf{H} = \begin{pmatrix} \kappa \mathbf{K}_1 & \mathtt{0} \\ \mathtt{a}^T \mathbf{K}_1 & 1 \end{pmatrix}, \tag{2.40}$$

which describes the transition from a metric to a projective reconstruction. Readers who are not very familiar with projective geometry can find in Appendix A all necessary background material that is used in this section, or they may skip this section and continue immediately with Section 2.6.3.

Recall that in projective geometry direction vectors are represented as points on the plane at infinity of the world space. If we would be able to identify the plane at infinity of the scene in the projective reconstruction, then by moving it to infinity, we can already upgrade the projective reconstruction to an affine one. In the projective reconstruction $\hat{\mathtt{M}}$ of Section 2.6.1, the plane at infinity of the scene is found as the plane with equation $\mathtt{a}^T \hat{\mathtt{M}} - \kappa = 0$. Indeed, as explained in Appendix A, the plane at infinity of the scene has homogeneous coordinates $(0,0,0,1)^T$ in the world space. Moreover, if a projective transformation whose action on homogeneous coordinates of 3D points is represented by an invertible 4×4-homography matrix $\mathbf{H}$ is applied to the world space, then homogeneous coordinates of planes are transformed by the inverse transpose $\mathbf{H}^{-T} = (\mathbf{H}^{-1})^T = (\mathbf{H}^T)^{-1}$ of the homography matrix $\mathbf{P}$. For example, the 3D similarity transformation $\bar{\mathtt{M}} = \frac{1}{\rho_{e2}} \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1)$ is represented in matrix notation by:

$$\begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\rho_{e2}} \mathbf{R}_1^T & -\frac{1}{\rho_{e2}} \mathbf{R}_1^T \mathtt{C}_1 \\ \mathtt{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathtt{M} \\ 1 \end{pmatrix}.$$

The corresponding homography matrix is:

$$\begin{pmatrix} \frac{1}{\rho_{e2}} \mathbf{R}_1^T & -\frac{1}{\rho_{e2}} \mathbf{R}_1^T \mathtt{C}_1 \\ \mathtt{0}^T & 1 \end{pmatrix}$$

and its inverse transpose is:

$$\begin{pmatrix} \rho_{e2} \mathbf{R}_1^T & \mathtt{0} \\ \mathtt{C}_1^T & 1 \end{pmatrix}.$$

The plane at infinity of the scene has homogeneous coordinates $(0,0,0,1)^T$ and is mapped by this similarity transformation onto itself, because

$$\begin{pmatrix} \rho_{e2}\mathbf{R}_1^T & \mathtt{0} \\ \mathtt{C}_1^T & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

This illustrates again the general fact that Euclidean, similarly and affine transformations of the world space do not affect the plane at infinity of the scene. A projective transformation on the contrary generally does affect the plane at infinity. In particular, the projective transformation $\lambda \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix}$ defined by the homography matrix (2.40) maps the plane at infinity of the scene to the plane with homogeneous coordinates:

$$\mathbf{H}^{-T} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\kappa}\mathbf{K}_1^{-T} & -\frac{1}{\kappa}\mathtt{a} \\ \mathtt{0}^T & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{\kappa}\mathtt{a} \\ 1 \end{pmatrix}.$$

In other words, the plane at infinity of the scene is found in the projective reconstruction $\hat{\mathtt{M}}$ as the plane with equation $-\frac{1}{\kappa}\mathtt{a}^T\hat{\mathtt{M}} + 1 = 0$, or equivalently, $\mathtt{a}^T\hat{\mathtt{M}} - \kappa = 0$. Given this geometric interpretation of $\frac{1}{\kappa}\mathtt{a}$, it becomes even clearer why we need to keep these values the same for the different choices of $\hat{\mathbf{A}}_j$ in the foregoing discussion about building a consistent, multi-view projective reconstruction of the scene. All two-view projective reconstructions need to share the same plane at infinity for them to be consistent. This said, even if one keeps the 3D reconstruction consistent following the aforementioned method, one still does not know $\mathbf{H}$ or $\frac{1}{\kappa}\mathtt{a}^T$.

If the position of the plane at infinity of the scene were known in the projective reconstruction, one could derive a projective transformation which will turn the projective reconstruction into an affine one, i.e., to put the plane at infinity really at infinity. Indeed, Equation (2.37) can

be rewritten as:

$$\lambda \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \kappa \mathbf{K}_1 & \mathtt{0} \\ \mathtt{a}^T \mathbf{K}_1 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \kappa \mathbf{I}_3 & \mathtt{0} \\ \mathtt{a}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{K}_1 \bar{\mathtt{M}} \\ 1 \end{pmatrix}$$
$$= \begin{pmatrix} \kappa \mathbf{I}_3 & \mathtt{0} \\ \mathtt{a}^T & 1 \end{pmatrix} \begin{pmatrix} \tilde{\mathtt{M}} \\ 1 \end{pmatrix},$$

where $\tilde{\mathtt{M}} = \mathbf{K}_1 \bar{\mathtt{M}}$ is the affine 3D reconstruction that was introduced in section 2.5.3 (cf. formula (2.30) in section 2.5.4). Knowing that $-\frac{1}{\kappa}\mathtt{a}^T \hat{\mathtt{M}} + 1 = 0$ is the equation of the plane at infinity of the scene in the projective reconstruction, the previous equation can also be written as:

$$\lambda \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \kappa \mathbf{I}_3 & \mathtt{0} \\ \mathtt{a}^T & 1 \end{pmatrix} \begin{pmatrix} \tilde{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_3 & \mathtt{0} \\ \frac{1}{\kappa}\mathtt{a}^T & 1 \end{pmatrix} \begin{pmatrix} \kappa \tilde{\mathtt{M}} \\ 1 \end{pmatrix}. \tag{2.41}$$

Observe that since $\tilde{\mathtt{M}}$ is an affine 3D reconstruction of the scene, $\kappa\tilde{\mathtt{M}}$ is an affine 3D reconstruction as well. Denoting $\pi_\infty = -\frac{1}{\kappa}\mathtt{a}$, the plane at infinity has equation $\pi_\infty^T \hat{\mathtt{M}} + 1 = 0$ and Equation (2.41) reads as:

$$\lambda \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_3 & \mathtt{0} \\ -\pi_\infty^T & 1 \end{pmatrix} \begin{pmatrix} \kappa \tilde{\mathtt{M}} \\ 1 \end{pmatrix}.$$

This is the 3D projective transformation which maps the plane at infinity in the affine reconstruction $\kappa\tilde{\mathtt{M}}$ to the plane with equation $\pi_\infty^T \hat{\mathtt{M}} + 1 = 0$ in the projective 3D reconstruction $\hat{\mathtt{M}}$. Put differently, if the plane at infinity of the scene can be identified as the plane $\pi_\infty^T \hat{\mathtt{M}} + 1 = 0$ in the projective reconstruction $\hat{\mathtt{M}}$ (e.g., from directions which are known to be parallel in the scene or from vanishing points in the images), then the inverse projective transformation:

$$\tilde{\lambda} \begin{pmatrix} \kappa\tilde{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_3 & \mathtt{0} \\ \pi_\infty^T & 1 \end{pmatrix} \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} \quad \text{with } \tilde{\lambda} \text{ a non-zero scalar,} \tag{2.42}$$

or equivalently,

$$\kappa\tilde{\mathtt{M}} = \frac{\hat{\mathtt{M}}}{{\pi_\infty}^T \hat{\mathtt{M}} + 1}$$

turns the projective 3D reconstruction $\hat{\mathtt{M}}$ into the affine reconstruction $\kappa\tilde{\mathtt{M}}$. In mathematical parlance, *the projective transformation* (2.42) *maps the plane* $\pi_\infty^T \hat{\mathtt{M}} + 1 = 0$ *to infinity.*

2.6.3 Recovering Metric Structure: Self-Calibration Equations

If no information is available about where the plane at infinity is located in the projective reconstruction $\hat{\mathtt{M}}$, one can still upgrade the reconstruction, even to metric. Recall from formulas (2.39) that $\hat{\rho}_j \mathtt{m}_j = \hat{\mathbf{A}}_j \hat{\mathtt{M}} + \mathtt{e}_j$ for all $j \geq 2$ and from Equation (2.37) that:

$$\lambda \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \kappa \mathbf{K}_1 & \mathtt{0} \\ \mathtt{a}^T \mathbf{K}_1 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix}.$$

Together, these equations yield

$$\begin{aligned} \hat{\rho}_j \mathtt{m}_j &= \hat{\mathbf{A}}_j \hat{\mathtt{M}} + \mathtt{e}_j = (\hat{\mathbf{A}}_j \mid \mathtt{e}_j) \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} \\ &= \frac{1}{\lambda} (\hat{\mathbf{A}}_j \mid \mathtt{e}_j) \begin{pmatrix} \kappa \mathbf{K}_1 & \mathtt{0} \\ \mathtt{a}^T \mathbf{K}_1 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix} \\ &= \frac{1}{\lambda} \begin{pmatrix} \kappa \hat{\mathbf{A}}_j \mathbf{K}_1 + \mathtt{e}_j \mathtt{a}^T \mathbf{K}_1 & \mathtt{e}_j \end{pmatrix} \begin{pmatrix} \bar{\mathtt{M}} \\ 1 \end{pmatrix} \end{aligned}$$

or equivalently,

$$\lambda \hat{\rho}_j \ \mathtt{m}_j = (\kappa \hat{\mathbf{A}}_j \mathbf{K}_1 + \mathtt{e}_j \mathtt{a}^T \mathbf{K}_1) \bar{\mathtt{M}} + \mathtt{e}_j \qquad \text{for all } j \geq 2.$$

On the other hand, a similar computation as in the derivation of the metric reconstruction Equations (2.24) in Section 2.5.2 shows that

$$\bar{\rho}_j \mathtt{m}_j = \mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \bar{\mathtt{M}} + \mathtt{e}_j, \text{ with } \bar{\rho}_j = \frac{\rho_j}{\rho_{ej}}.$$

Comparing both equations, one sees that

$$\kappa \hat{\mathbf{A}}_j \mathbf{K}_1 + \mathtt{e}_j \mathtt{a}^T \mathbf{K}_1 = \ \lambda_j \ \mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \quad \text{for all } j \geq 2$$

and for some scalar λ_j. In Section 2.6.2 it is observed that the parameters $\kappa \in \mathbb{R}$ and $\mathbf{a} \in \mathbb{R}^3$ determine the plane at infinity of the scene in the projective reconstruction: $\pi_\infty = -\frac{1}{\kappa} \mathbf{a}$. Making this reference to the plane at infinity of the scene explicit in the previous equation, one gets

$$\kappa_j (\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T) \mathbf{K}_1 = \mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \quad \text{for all } j \geq 2, \tag{2.43}$$

where $\kappa_j = \kappa/\lambda_j$ is a non-zero scalar. This equation has two interesting consequences. First of all, multiplying both sides of the equality on the right with the inverse of the calibration matrix $\mathbf{K}_1$ gives

$$\kappa_j(\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T) = \mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \mathbf{K}_1^{-1} \quad \text{for all } j \geq 2,$$

which, since the right-hand side of this equation is just the invertible matrix $\mathbf{A}_j$ defined in Section 2.4.1, yields an explicit relation between the infinite homography $\mathbf{A}_j$ and the 3×3-matrices $\hat{\mathbf{A}}_j$ computed from the images as described in section 2.6.1, viz

$$\mathbf{A}_j = \kappa_j (\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T) \quad \text{for all } j \geq 2 \tag{2.44}$$

with non-zero scalars $\kappa_j \in \mathbb{R}$. And secondly, multiplying both sides in equality (2.43) on the right with the inverse $\mathbf{R}_1^{-1} = \mathbf{R}_1^T$ of the rotation matrix $\mathbf{R}_1$ gives

$$\kappa_j(\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T)\mathbf{K}_1 \mathbf{R}_1^T = \mathbf{K}_j \mathbf{R}_j^T \quad \text{for all } j \geq 2.$$

If one now multiplies both sides of this last equation with its transpose, then

$$(\mathbf{K}_j \mathbf{R}_j^T)(\mathbf{K}_j \mathbf{R}_j^T)^T = \kappa_j^2(\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T)\mathbf{K}_1 \mathbf{R}_1^T(\mathbf{K}_1 \mathbf{R}_1^T)^T(\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T)^T$$

for all $j \geq 2$, which by $\mathbf{R}_j^T = \mathbf{R}_j^{-1}$ reduces to

$$\mathbf{K}_j \mathbf{K}_j^T = \kappa_j^2(\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T)\mathbf{K}_1 \mathbf{K}_1^T(\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T)^T \tag{2.45}$$

for all $j \in \{2, \ldots, m\}$. Equations (2.45) are the so-called *self-calibration* or *autocalibration equations* [8] and all self-calibration methods essentially are variations on solving these equations for the calibration matrices $\mathbf{K}_j$ and the three-vector π_∞ locating the plane at infinity of the scene in the projective reconstruction. The various methods may differ in the constraints or the assumptions on the calibration matrices they employ, however.

2.6.4 Scrutinizing the Self-Calibration Equations

The self-calibration Equations (2.45) have a simple geometric interpretation, which we will explore first before looking into ways for solving them. Readers who are merely interested in the practice of 3D reconstruction may skip this section and continue with Section 2.6.5.

2.6.4.1 Metric Structure and the Preservation of Angles

In Section 2.5.2 it was observed that if one wants to reconstruct a scene from images only and if no absolute distance is given for any parts of the scene, then one can never hope to do better than a metric reconstruction, i.e., a 3D reconstruction which differs from the original scene by an unknown 3D similarity transformation. Typical of a 3D similarity transformation is that all distances in the scene are scaled by a fixed scalar factor and that all angles are preserved. Moreover, *for a projective 3D reconstruction of the scene to be a metric one, it is necessary and sufficient that the angles of any triangle formed by three points in the reconstruction are equal to the corresponding angles in the triangle formed by the original three scene points.* The self-calibration Equations (2.45) enforce this condition in the projective reconstruction at hand, as will be explained now.

Let $\mathtt{M}$, $\mathtt{P}$, and $\mathtt{Q}$ be three arbitrary points in the scene. The angle between the line segments $[\mathtt{M},\mathtt{P}]$ and $[\mathtt{M},\mathtt{Q}]$ in Euclidean three-space is found as the angle between the three-vectors $\mathtt{P}-\mathtt{M}$ and $\mathtt{Q}-\mathtt{M}$ in $\mathbb{R}^3$. This angle is uniquely defined by its cosine, which is given by the formula:

$$\cos(\mathtt{P}-\mathtt{M},\mathtt{Q}-\mathtt{M}) = \frac{\langle \mathtt{P}-\mathtt{M},\mathtt{Q}-\mathtt{M}\rangle}{\|\mathtt{P}-\mathtt{M}\|\,\|\mathtt{Q}-\mathtt{M}\|},$$

where $\langle \mathtt{P}-\mathtt{M},\mathtt{Q}-\mathtt{M}\rangle$ denotes the (standard) inner product; and, $\|\mathtt{P}-\mathtt{M}\| = \sqrt{\langle \mathtt{P}-\mathtt{M},\mathtt{P}-\mathtt{M}\rangle}$ and $\|\mathtt{Q}-\mathtt{M}\| = \sqrt{\langle \mathtt{Q}-\mathtt{M},\mathtt{Q}-\mathtt{M}\rangle}$ are the norms of the three-vectors $\mathtt{P}-\mathtt{M}$ and $\mathtt{Q}-\mathtt{M}$ in $\mathbb{R}^3$. Now, $\mathtt{P}-\mathtt{M}$ is a direction vector for the line defined by the points $\mathtt{M}$ and $\mathtt{P}$ in the scene. As explained in Appendix A, the vanishing point $\mathtt{v}_j$ of the line $\overline{\mathtt{MP}}$ in the j-th image ($j \in \{1,2,\ldots,m\}$) is given by $\rho_{vj}\,\mathtt{v}_j = \mathbf{K}_j\,\mathbf{R}_j^T(\mathtt{P}-\mathtt{M})$, where $\rho_j\,\mathtt{m}_j = \mathbf{K}_j\,\mathbf{R}_j^T(\mathtt{M}-\mathtt{C}_j)$ are the projection equations of the j-th camera, as defined by formula (2.34). Since $\mathbf{R}_j$ is a rotation matrix, $\mathbf{R}_j^T = \mathbf{R}_j^{-1}$ and the three-vector $\mathtt{P}-\mathtt{M}$ can (theoretically) be recovered up to scale from its vanishing point $\mathtt{v}_j$ in the j-th image by the formula $\mathtt{P}-\mathtt{M} = \rho_{vj}\,\mathbf{R}_j\,\mathbf{K}_j^{-1}\mathtt{v}_j$. Similarly, $\mathtt{Q}-\mathtt{M}$ is a direction vector of the line through the points $\mathtt{M}$ and $\mathtt{Q}$ in the scene and the vanishing point $\mathtt{w}_j$ of this line in the j-th image is given by $\rho_{wj}\,\mathtt{w}_j = \mathbf{K}_j\,\mathbf{R}_j^T(\mathtt{Q}-\mathtt{M})$, thus yielding $\mathtt{Q}-\mathtt{M} = \rho_{wj}\,\mathbf{R}_j\,\mathbf{K}_j^{-1}\mathtt{w}_j$. By definition of the inner

product in $\mathbb{R}^3$,

$$\begin{aligned}\langle \mathtt{P}-\mathtt{M}, \mathtt{Q}-\mathtt{M}\rangle &= (\mathtt{P}-\mathtt{M})^T(\mathtt{Q}-\mathtt{M}) \\ &= (\rho_{vj}\,\mathbf{R}_j\,\mathbf{K}_j^{-1}\mathtt{v}_j)^T(\rho_{wj}\,\mathbf{R}_j\,\mathbf{K}_j^{-1}\mathtt{w}_j) \\ &= \rho_{vj}\,\rho_{wj}\ \mathtt{v}_j^T\,\mathbf{K}_j^{-T}\,\mathbf{K}_j^{-1}\,\mathtt{w}_j \\ &= \rho_{vj}\,\rho_{wj}\ \mathtt{v}_j^T(\mathbf{K}_j\,\mathbf{K}_j^T)^{-1}\mathtt{w}_j.\end{aligned}$$

A similar calculation yields

$$\|\mathtt{P}-\mathtt{M}\|^2 = (\mathtt{P}-\mathtt{M})^T(\mathtt{P}-\mathtt{M}) = \rho_{vj}^2\ \mathtt{v}_j^T(\mathbf{K}_j\,\mathbf{K}_j^T)^{-1}\mathtt{v}_j$$

$$\text{and}\quad \|\mathtt{Q}-\mathtt{M}\|^2 = (\mathtt{Q}-\mathtt{M})^T(\mathtt{Q}-\mathtt{M}) = \rho_{wj}^2\ \mathtt{w}_j^T(\mathbf{K}_j\,\mathbf{K}_j^T)^{-1}\mathtt{w}_j.$$

Combining the previous expressions, one gets the following formula for the cosine of the angle between the line segments $[\mathtt{M},\mathtt{P}]$ and $[\mathtt{M},\mathtt{Q}]$ in the scene:

$$\cos(\mathtt{P}-\mathtt{M},\mathtt{Q}-\mathtt{M}) = \frac{\mathtt{v}_j^T(\mathbf{K}_j\,\mathbf{K}_j^T)^{-1}\mathtt{w}_j}{\sqrt{\mathtt{v}_j^T(\mathbf{K}_j\,\mathbf{K}_j^T)^{-1}\mathtt{v}_j}\ \sqrt{\mathtt{w}_j^T(\mathbf{K}_j\,\mathbf{K}_j^T)^{-1}\mathtt{w}_j}}. \qquad (2.46)$$

This equation states that *the angle between two lines in the scene can be measured from a perspective image of the scene if the vanishing points of these lines can be identified in the image and if the calibration matrix* $\mathbf{K}_j$ *of the camera — or rather* $\mathbf{K}_j\,\mathbf{K}_j^T$ *— is known.*

This should not come as a surprise, since vanishing points encode 3D directions in a perspective image. What is more interesting to notice, however, is that the inner product of the scene is encoded in the image by the symmetric matrix $(\mathbf{K}_j\,\mathbf{K}_j^T)^{-1}$. In the computer vision literature this matrix is commonly denoted by ω_j and referred to as the *image of the absolute conic* in the j-th view [16]. We will not expand on this interpretation right now, but the interested reader can find more details on this matter in Appendix B. The fact that the calibration matrix $\mathbf{K}_j$ appears in a formula relating measurements in the image to measurements in the scene was to be expected. The factor $\mathbf{R}_j^T(\mathtt{M}-\mathtt{C}_j)$ in the right-hand side of the projection equations $\rho_j\,\mathtt{m}_j = \mathbf{K}_j\,\mathbf{R}_j^T(\mathtt{M}-\mathtt{C}_j)$ corresponds to a rigid motion of the scene, and hence does not have an influence on angles and distances in the scene. The calibration matrix $\mathbf{K}_j$, on the other hand, is an upper triangular matrix, and thus introduces scaling and skewing. When measuring scene angles and distances

from the image, one therefore has to undo this skewing and scaling first by premultiplying the image coordinates with the inverse matrix $\mathbf{K}_j^{-1}$. And, last but not least, from the point of view of camera self-calibration, Equation (2.46) introduces additional constraints between different images of the same scene, viz.:

$$\frac{\mathtt{v}_i^T(\mathbf{K}_i\mathbf{K}_i^T)^{-1}\mathtt{w}_i}{\sqrt{\mathtt{v}_i^T(\mathbf{K}_i\mathbf{K}_i^T)^{-1}\mathtt{v}_i}\,\sqrt{\mathtt{w}_i^T(\mathbf{K}_i\mathbf{K}_i^T)^{-1}\mathtt{w}_i}} = \frac{\mathtt{v}_j^T(\mathbf{K}_j\mathbf{K}_j^T)^{-1}\mathtt{w}_j}{\sqrt{\mathtt{v}_j^T(\mathbf{K}_j\mathbf{K}_j^T)^{-1}\mathtt{v}_j}\,\sqrt{\mathtt{w}_j^T(\mathbf{K}_j\mathbf{K}_j^T)^{-1}\mathtt{w}_j}},$$

which must hold for every pair of images $i, j \in \{1, 2, \ldots, m\}$ and for every two pairs of corresponding vanishing points $\mathtt{v}_i$, $\mathtt{v}_j$ and $\mathtt{w}_i$, $\mathtt{w}_j$ in these images. Obviously, only $m - 1$ of these relations are independent:

$$\frac{\mathtt{v}_1^T\left(\mathbf{K}_1\mathbf{K}_1^T\right)^{-1}\mathtt{w}_1}{\sqrt{\mathtt{v}_1^T\left(\mathbf{K}_1\mathbf{K}_1^T\right)^{-1}\mathtt{v}_1}\,\sqrt{\mathtt{w}_1^T\left(\mathbf{K}_1\mathbf{K}_1^T\right)^{-1}\mathtt{w}_1}} = \frac{\mathtt{v}_j^T\left(\mathbf{K}_j\mathbf{K}_j^T\right)^{-1}\mathtt{w}_j}{\sqrt{\mathtt{v}_j^T\left(\mathbf{K}_j\mathbf{K}_j^T\right)^{-1}\mathtt{v}_j}\,\sqrt{\mathtt{w}_j^T\left(\mathbf{K}_j\mathbf{K}_j^T\right)^{-1}\mathtt{w}_j}} \tag{2.47}$$

for every $j \geq 2$ and for every two pairs of corresponding vanishing points $\mathtt{v}_1$, $\mathtt{v}_j$ and $\mathtt{w}_1$, $\mathtt{w}_j$ between the first and the j-th image. The self-calibration Equations (2.45) enforce these constraints in the projective reconstruction of the scene, as will be demonstrated next.

2.6.4.2 Infinity Homographies and the Preservation of Angles

To see why the final claim in the last section holds, we have to reinterpret the underlying relation (2.46) in terms of projective geometry. Consider again the (arbitrary) scene points $\mathtt{M}$, $\mathtt{P}$, and $\mathtt{Q}$. Their extended coordinates, respectively, are $\binom{\mathtt{M}}{1}$, $\binom{\mathtt{P}}{1}$, and $\binom{\mathtt{Q}}{1}$. As explained in Appendix A, the direction vector $\mathtt{P} - \mathtt{M}$ of the line $\mathtt{L}$ through the points $\mathtt{M}$ and $\mathtt{P}$ in the scene corresponds in projective three-space to the point of intersection of the line through the projective points $\binom{\mathtt{M}}{1}$ and $\binom{\mathtt{P}}{1}$ with the plane at infinity of the scene; and the vanishing point $\mathtt{v}_j$ of $\mathtt{L}$ in the j-th image is the perspective projection of this point of intersection onto the image plane of the j-th camera ($j \in \{1, 2, \ldots, m\}$). In particular, the vanishing points $\mathtt{v}_1, \mathtt{v}_2, \ldots, \mathtt{v}_m$ are corresponding points in the images $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_m$. Moreover, it was explained in Section 2.4.3 that the matrix $\mathbf{A}_j = \mathbf{K}_j\mathbf{R}_j^T\mathbf{R}_1\mathbf{K}_1^{-1}$ introduced in Section 2.4.1 actually is

a homography matrix representing the projective transformation that maps (vanishing) points from the first image via the plane at infinity of the scene onto the corresponding (vanishing) point in the j-th image ($j \geq 2$), and therefore is referred to in the computer vision literature as the *infinite homography* between the first and the j-th image. Explicitly, $\mathbf{A}_j \mathtt{v}_1 = \rho_j \mathtt{v}_j$ for some non-zero scalar factor ρ_j. Similarly, the vanishing points $\mathtt{w}_1$ and $\mathtt{w}_j$ of the line through $\mathtt{M}$ and $\mathtt{Q}$ in, respectively, the first and j-th image satisfy $\mathbf{A}_j \mathtt{w}_1 = \sigma_j \mathtt{w}_j$ for some non-zero scalar factor σ_j. Using the infinite homography $\mathbf{A}_j = \mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \mathbf{K}_1^{-1}$, the inner product $\mathtt{v}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{w}_j$ in the j-th image can also be expressed in terms of the corresponding vanishing points $\mathtt{v}_1$ and $\mathtt{w}_1$ in the first image:

$$\begin{aligned} \mathtt{v}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{w}_j &= \left(\frac{1}{\rho_j} \mathbf{A}_j \mathtt{v}_1 \right)^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \left(\frac{1}{\sigma_j} \mathbf{A}_j \mathtt{w}_1 \right) \\ &= \frac{1}{\rho_j} \frac{1}{\sigma_j} \mathtt{v}_1^T \mathbf{A}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathbf{A}_j \mathtt{w}_1. \end{aligned} \tag{2.48}$$

Using again $\mathbf{A}_j = \mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \mathbf{K}_1^{-1}$ and the fact that $\mathbf{R}_j^T = \mathbf{R}_j^{-1}$, since $\mathbf{R}_j$ is a rotation matrix, the 3×3-matrix $\mathbf{A}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathbf{A}_j$ in the right-hand side of the previous equality simplifies to:

$$\begin{aligned} \mathbf{A}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathbf{A}_j &= (\mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \mathbf{K}_1^{-1})^T (\mathbf{K}_j^{-T} \mathbf{K}_j^{-1}) (\mathbf{K}_j \mathbf{R}_j^T \mathbf{R}_1 \mathbf{K}_1^{-1}) \\ &= (\mathbf{K}_1 \mathbf{K}_1^T)^{-1}. \end{aligned}$$

Equation (2.48) then reads

$$\mathtt{v}_1^T (\mathbf{K}_1 \mathbf{K}_1^T)^{-1} \mathtt{w}_1 = \rho_j \sigma_j \mathtt{v}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{w}_j.$$

By similar calculations, one also finds that $\mathtt{v}_1^T (\mathbf{K}_1 \mathbf{K}_1^T)^{-1} \mathtt{v}_1 = \rho_j^2 \mathtt{v}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{v}_j$ and $\mathtt{w}_1^T (\mathbf{K}_1 \mathbf{K}_1^T)^{-1} \mathtt{w}_1 = \sigma_j^2 \mathtt{w}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{w}_j$. Together, these three equalities establish the relations (2.47) in an almost trivial manner. It is important to realize, however, that it actually is the equality:

$$\mathbf{A}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathbf{A}_j = (\mathbf{K}_1 \mathbf{K}_1^T)^{-1}, \tag{2.49}$$

which makes that the constraints (2.47) are satisfied. Our claim is that the self-calibration Equations (2.45) are nothing else but this fundamental relation (2.49) expressed in terms of the projective

reconstruction of the scene which was computed from the given images as described in Section 2.6.1.

2.6.4.3 Equivalence of Self-Calibration and the Preservation of Angles

Now, suppose that no information whatsoever on the scene is given, but that we have computed matrices $\hat{\mathbf{A}}_j$ and epipoles $\mathtt{e}_j$ for each image ($j \geq 2$) as well as a collection of 3D points $\hat{\mathtt{M}}$ satisfying the projective reconstruction Equations (2.39), viz.:

$$\hat{\rho}_1 \mathtt{m}_1 = \hat{\mathtt{M}} \quad \text{and} \quad \hat{\rho}_j \mathtt{m}_j = \hat{\mathbf{A}}_j \hat{\mathtt{M}} + \mathtt{e}_j \quad \text{for } j \in \{2, \ldots, m\},$$

for given point correspondences $\mathtt{m}_1, \mathtt{m}_2, \ldots, \mathtt{m}_m$ between the images. The 3D points $\hat{\mathtt{M}}$ have been shown in Section 2.5.4 to form a projective 3D reconstruction of the scene. To prove our claim about the self-calibration equations, we will follow the same line of reasoning as the one that led to relation (2.49) above. Let $\hat{\mathtt{M}}$ and $\hat{\mathtt{P}}$ be two arbitrary points in the projective reconstruction. As explained in Appendix A, the vanishing point $\mathtt{v}_j$ in the j-th image of the line $\hat{\mathtt{L}}$ through the points $\hat{\mathtt{M}}$ and $\hat{\mathtt{P}}$ in the 3D reconstruction is the projection of the point of intersection of the line $\hat{\mathtt{L}}$ with the plane at infinity of the scene. Since no information whatsoever on the scene is available, we do not know where the plane at infinity of the scene is located in the reconstruction. However, we do know that for every 3D line in the projective reconstruction, its vanishing points in the respective images should be mapped onto each other by the projective transformation which maps one image to another via the plane at infinity of the scene. The idea now is to identify a plane in the projective reconstruction which does exactly that. Suppose that the (unknown) equation of this plane is $\pi_\infty^T \hat{\mathtt{M}} + 1 = 0$. The projective transformation which maps the first image to the j-th one via this plane is computed as follows: From the first projective reconstruction equation $\hat{\rho}_1 \mathtt{m}_1 = \hat{\mathtt{M}}$ it follows that parameter equations for the projecting ray of an arbitrary image point $\mathtt{m}_1$ in the first camera are given by $\hat{\mathtt{M}} = \hat{\rho} \mathtt{m}_1$ where $\hat{\rho} \in \mathbb{R}$ is a scalar parameter. This projecting ray intersects the plane at infinity of the scene in the point $\hat{\mathtt{M}}_\infty = \hat{\rho}_\infty \mathtt{m}_1$ that satifies the

equation $\pi_\infty^T \hat{\mathtt{M}}_\infty + 1 = 0$. Hence,

$$\hat{\rho}_\infty = -\frac{1}{\pi_\infty^T \mathtt{m}_1} \quad \textit{and} \quad \hat{\mathtt{M}}_\infty = -\frac{1}{\pi_\infty^T \mathtt{m}_1}\, \mathtt{m}_1.$$

By the projective reconstruction Equations (2.39), the projection $\mathtt{m}_j$ of this point $\hat{\mathtt{M}}_\infty$ in the j-th image for $j \geq 2$ satisfies $\hat{\rho}_j \mathtt{m}_j = \hat{\mathbf{A}}_j \hat{\mathtt{M}}_\infty + \mathtt{e}_j$. Substituting the expression for $\hat{\mathtt{M}}_\infty$ in this equation yields

$$\hat{\rho}_j \mathtt{m}_j = -\frac{1}{\pi_\infty^T \mathtt{m}_1}\ \hat{\mathbf{A}}_j \mathtt{m}_1 + \mathtt{e}_j,$$

or equivalently,

$$-(\pi_\infty^T \mathtt{m}_1)\hat{\rho}_j\ \mathtt{m}_j = \hat{\mathbf{A}}_j \mathtt{m}_1 - \mathtt{e}_j(\pi_\infty^T \mathtt{m}_1) = (\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)\mathtt{m}_1.$$

Consequently, the 3×3-matrix $\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T$ is a homography matrix of the projective transformation which maps the first image to the j-th one via the plane at infinity of the scene. In Section 2.4.3, on the other hand, it was demonstrated that the invertible matrix $\mathbf{A}_j$ introduced in Section 2.4.1 also is a matrix for this infinite homography. Therefore, $\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T$ and $\mathbf{A}_j$ must be equal up to a non-zero scalar factor, i.e., $\mathbf{A}_j = \kappa_j (\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)$ for some non-zero scalar $\kappa_j \in \mathbb{R}$. Notice that this is exactly the same expression for $\mathbf{A}_j$ as was found in Equation (2.44) of Section 2.6.3, but now it is obtained by a geometrical argument instead of an algebraic one.

Clearly, for any plane $\pi^T \hat{\mathtt{M}} + 1 = 0$ the homography matrix $\hat{\mathbf{A}}_j - \mathtt{e}_j \pi^T$ will map the first image to the j-th one via that plane. But only the plane at infinity of the scene will guarantee that the cosines

$$\frac{\mathtt{v}_1^T (\mathbf{K}_1 \mathbf{K}_1^T)^{-1} \mathtt{w}_1}{\sqrt{\mathtt{v}_1^T (\mathbf{K}_1 \mathbf{K}_1^T)^{-1} \mathtt{v}_1} \sqrt{\mathtt{w}_1^T (\mathbf{K}_1 \mathbf{K}_1^T)^{-1} \mathtt{w}_1}}$$

$$\text{and} \quad \frac{\mathtt{v}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{w}_j}{\sqrt{\mathtt{v}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{v}_j} \sqrt{\mathtt{w}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathtt{w}_j}}$$

(cf. Equation (2.47)) computed in each image j ($j \geq 2$) admit the same value whenever $\mathtt{v}_1$ is mapped onto $\mathtt{v}_j$ and $\mathtt{w}_1$ is mapped onto $\mathtt{w}_j$ by the homography $\hat{\mathbf{A}}_j - \mathtt{e}_j \pi^T$. And, as was observed earlier in this section, this will only be guaranteed if $\mathbf{A}_j^T (\mathbf{K}_j \mathbf{K}_j^T)^{-1} \mathbf{A}_j = (\mathbf{K}_1 \mathbf{K}_1^T)^{-1}$ where

$\mathbf{A}_j$ now has to be interpreted as the infinite homography mapping the first image to the j-th image by the plane at infinity of the scene (cf. Equation (2.49)). Since $\mathbf{A}_j = \kappa_j(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)$, the plane at infinity of the scene is that plane $\pi_\infty^T \hat{\mathtt{M}} + 1 = 0$ in the projective reconstruction for which

$$\kappa_j^2(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)^T(\mathbf{K}_j \mathbf{K}_j^T)^{-1}(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T) = (\mathbf{K}_1 \mathbf{K}_1^T)^{-1}$$

for all $j \in \{2,3,\ldots,m\}$. This equality expresses that the intrinsic way of measuring in the j-th image — represented by the symmetric matrix $(\mathbf{K}_j \mathbf{K}_j^T)^{-1}$ — must be compatible with the intrinsic way of measuring in the first image — represented by the symmetric matrix $(\mathbf{K}_1 \mathbf{K}_1^T)^{-1}$ — and, more importantly, that it is the infinite homography between the first and the j-th image — represented by the matrix $(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)$ — which actually transforms the metric $(\mathbf{K}_j \mathbf{K}_j^T)^{-1}$ into the metric $(\mathbf{K}_1 \mathbf{K}_1^T)^{-1}$. Finally, if both sides of this matrix equality are inverted, one gets the relation:

$$\frac{1}{\kappa_j^2}\,(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)^{-1}\,\mathbf{K}_j \mathbf{K}_j^T\,(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)^{-T} = \mathbf{K}_1 \mathbf{K}_1^T,$$

or, after solving for $\mathbf{K}_j \mathbf{K}_j^T$,

$$\mathbf{K}_j \mathbf{K}_j^T = \kappa_j^2(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)\,\mathbf{K}_1 \mathbf{K}_1^T\,(\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)^T, \tag{2.50}$$

which must hold for all $j \in \{2,3,\ldots,m\}$. Observe that these are exactly the self-calibration Equations (2.45), as we claimed earlier.

Intuitively, these equations state that if a plane $\pi_\infty^T \hat{\mathtt{M}} + 1 = 0$ in the reconstruction is known or found to be the plane at infinity of the scene and if a metric — represented by a symmetric, positive-definite matrix $\mathbf{K}_1 \mathbf{K}_1^T$ — is induced in the first image, then by really mapping the plane $\pi_\infty \hat{\mathtt{M}} + 1 = 0$ to infinity and by measuring in the j-th image $(j \geq 2)$ according to a metric induced from $\mathbf{K}_1 \mathbf{K}_1^T$ through formula (2.50), the projective reconstruction will be upgraded to a metric 3D reconstruction of the scene. This is in accordance with our findings in Section 2.6.2 that the projective transformation:

$$\kappa \bar{\mathtt{M}} = \frac{\mathbf{K}_1^{-1} \hat{\mathtt{M}}}{\pi_\infty^T \hat{\mathtt{M}} + 1}$$

transforms the projective reconstruction $\hat{\mathtt{M}}$ into the metric reconstruction $\kappa\bar{\mathtt{M}}$.

2.6.5 A Glimpse on Absolute Conics and Quadrics

The right-hand side of the self-calibration Equations (2.50) can also be written as:

$$\begin{aligned}\mathbf{K}_j\mathbf{K}_j^T &= \kappa_j^2(\hat{\mathbf{A}}_j - \mathsf{e}_j\pi_\infty^T)\,\mathbf{K}_1\mathbf{K}_1^T\,(\hat{\mathbf{A}}_j - \mathsf{e}_j\pi_\infty^T)^T \\ &= \kappa_j^2\begin{pmatrix}\hat{\mathbf{A}}_j & \mathsf{e}_j\end{pmatrix}\begin{pmatrix}\mathbf{I}_3 \\ -\pi_\infty^T\end{pmatrix}\mathbf{K}_1\mathbf{K}_1^T\begin{pmatrix}\mathbf{I}_3 & -\pi_\infty\end{pmatrix}\begin{pmatrix}\hat{\mathbf{A}}_j^T \\ \mathsf{e}_j^T\end{pmatrix} \\ &= \kappa_j^2\begin{pmatrix}\hat{\mathbf{A}}_j & \mathsf{e}_j\end{pmatrix}\begin{pmatrix}\mathbf{K}_1\mathbf{K}_1^T & -\mathbf{K}_1\mathbf{K}_1^T\pi_\infty \\ -\pi_\infty^T\mathbf{K}_1\mathbf{K}_1^T & \pi_\infty^T\mathbf{K}_1\mathbf{K}_1^T\pi_\infty\end{pmatrix}\begin{pmatrix}\hat{\mathbf{A}}_j^T \\ \mathsf{e}_j^T\end{pmatrix}.\end{aligned}$$

In the computer vision literature, the 3×3-matrix $\mathbf{K}_j\mathbf{K}_j^T$ in the left-hand side of the equality is commonly denoted by ω_j^* and referred to as the *dual image of the absolute conic* in the j-th view [16]. It is the mathematical dual of the *image of the absolute conic* ω_j in the j-th view. The 4×4-matrix

$$\Omega^* = \begin{pmatrix}\mathbf{K}_1\mathbf{K}_1^T & -\mathbf{K}_1\mathbf{K}_1^T\pi_\infty \\ -\pi_\infty^T\mathbf{K}_1\mathbf{K}_1^T & \pi_\infty^T\mathbf{K}_1\mathbf{K}_1^T\pi_\infty\end{pmatrix} \tag{2.51}$$

in the right-hand side of the previous equality, on the other hand, is in the computer vision literature usually referred to as the *absolute quadric* [16]. The self-calibration Equations (2.50) are compactly written in this manner as:

$$\omega_j^* = \kappa_j^2\,(\hat{\mathbf{A}}_j \mid \mathsf{e}_j)\;\Omega^*\,(\hat{\mathbf{A}}_j \mid \mathsf{e}_j)^T, \tag{2.52}$$

where $\omega_j^* = \mathbf{K}_j\mathbf{K}_j^T$ and with Ω^* as defined above. We will not expand on this interpretation in terms of projective geometry right now, but the interested reader can find more details on this matter in Appendix B.

The main advantage of writing the self-calibration equations in the form (2.52) is that it yields linear equations in the entries of ω_j^* and the entries of Ω^*, which are easier to solve in practice than the non-linear formulation (2.50). Moreover, the absolute quadric Ω^* encodes

both the plane at infinity of the scene and the internal calibration of the first camera in a very concise fashion. Indeed,

$$\Omega^* = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \\ -\pi_\infty^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{K}_1\mathbf{K}_1^T & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \\ -\pi_\infty^T & 1 \end{pmatrix}^T,$$

from which it immediately follows that Ω^* has rank 3 and that its nullspace is spanned by the plane at infinity of the scene $(\pi_\infty^T\, 1)^T$. Moreover, Ω^* can also be decomposed as:

$$\Omega^* = \begin{pmatrix} \mathbf{K}_1 & \mathbf{0} \\ -\pi_\infty^T\mathbf{K}_1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{K}_1 & \mathbf{0} \\ -\pi_\infty^T\mathbf{K}_1 & 1 \end{pmatrix}^T,$$

where the 4×4-matrix

$$\begin{pmatrix} \mathbf{K}_1 & \mathbf{0} \\ -\pi_\infty^T\mathbf{K}_1 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} \\ -\pi_\infty^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{K}_1 & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}, \tag{2.53}$$

is the homography matrix of the projective transformation

$$\lambda \begin{pmatrix} \hat{\mathtt{M}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{K}_1 & \mathbf{0} \\ -\pi_\infty^T\mathbf{K}_1 & 1 \end{pmatrix} \begin{pmatrix} \kappa\bar{\mathtt{M}} \\ 1 \end{pmatrix},$$

which relates the projective reconstruction $\hat{\mathtt{M}}$ to the metric reconstruction $\kappa\bar{\mathtt{M}}$, as discussed in Section 2.6.2. Hence, the rectifying homography to update the projective reconstruction of the scene to a metric one is directly available once the absolute quadric Ω^* has been recovered. It is interesting to observe that the decomposition (2.53) of the rectifying homography exhibits the stratification of geometries as discussed in Section 2.5.5. The rightmost matrix in the right-hand side of Equation (2.53) represents the affine deformation induced on the metric reconstruction $\kappa\bar{\mathtt{M}}$ by including the uncertainty about the internal camera parameters $\mathbf{K}_1$ in the 3D reconstruction, yielding the affine reconstruction $\kappa\tilde{\mathtt{M}}$; and the leftmost matrix in the decomposition (2.53) changes the plane at infinity to the plane $\pi_\infty^T\hat{\mathtt{M}} + 1 = 0$ in the projective reconstruction $\hat{\mathtt{M}}$ of the scene.

How Ω^* is computed in practice, given a projective reconstruction of the scene is discussed in more detail in Section 4.6.2 of Section 4. Therefore we will not continue with this topic any further here, but instead we will address the question of how many images are needed in order for the self-calibration equations to yield a unique solution.

2.6.6 When do the Self-Calibration Equations Yield a Unique Solution ?

The self-calibration equations yield additional constraints on the calibration matrices of the cameras and about the location of the plane at infinity of the scene in a projective 3D reconstruction. But as was already observed in Sections 2.5.4 and 2.5.5, if no additional information about the internal and/or external calibration of the cameras or about the Euclidean structure of the scene is available, then with only two images one cannot hope for better than a projective reconstruction of the scene. The question that remains unsolved up till now is: Is it possible to recover a metric reconstruction of the scene with only the images at our disposal; and, more importantly, how many images are needed to obtain a unique solution?

Consider again the self-calibration Equations (2.50) in their compact formulation:

$$\mathbf{K}_j \mathbf{K}_j^T = \kappa_j^2 (\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T) \mathbf{K}_1 \mathbf{K}_1^T (\hat{\mathbf{A}}_j - \mathbf{e}_j \pi_\infty^T)^T \tag{2.54}$$

for each $j \in \{2, 3, \ldots, m\}$. In these equations the calibration matrices $\mathbf{K}_j$ all appear as $\mathbf{K}_j \mathbf{K}_j^T$. It is thus advantageous to take the entries of $\mathbf{K}_j \mathbf{K}_j^T$ as the unknowns in the self-calibration equations instead of expressing them in terms of the internal camera parameters constituting $\mathbf{K}_j$ (cf. formula (2.4)). As $\mathbf{K}_j$ is an invertible upper-triangular matrix, $\mathbf{K}_j \mathbf{K}_j^T$ is a positive-definite symmetric matrix. So, if $\mathbf{K}_j \mathbf{K}_j^T$ is known, the calibration matrix $\mathbf{K}_j$ itself can uniquely be obtained from $\mathbf{K}_j \mathbf{K}_j^T$ by Cholesky factorization [3]. Furthermore, each $\mathbf{K}_j \mathbf{K}_j^T$ is a symmetric 3×3-matrix whose $(3,3)$-th entry equals 1 (cf. formula (2.4)). Consequently, each $\mathbf{K}_j \mathbf{K}_j^T$ is completely characterized by five scalar parameters, viz. the diagonal elements other than the $(3,3)$-th one and the upper-triangular entries. Similarly, the scalar factors κ_j^2 can be considered as being single variables in the self-calibration equations. Together with the three unknown components of the three-vector π_∞, the number of unknowns in the self-calibration Equations (2.54) for m images add up to $5m + (m-1) + 3 = 6m + 2$. On the other hand, for m images, formula (2.54) yields $m - 1$ matrix equations. Since both sides of these equations are formed by symmetric

3×3-matrices, each matrix equation induces only six different non-linear equations in the components of $\mathbf{K}_j\mathbf{K}_j^T$, the components of π_∞ and the scalars κ_j^2 for $j \in \{1, 2, \ldots, m\}$. Hence, for m images, the self-calibration Equations (2.54) yield a system of $6(m-1) = 6m - 6$ non-linear equations in $6m + 2$ unknowns. Clearly, without additional constraints on the unknowns, this system does not have a unique solution.

In practical situations, however, quantitative or qualitative information about the cameras can be used to constrain the number of solutions. Let us consider some examples.

- **Images obtained by the same or identical cameras.**
  If the images are obtained with one or more cameras whose calibration matrices are the same, then $\mathbf{K}_1 = \mathbf{K}_2 = \ldots = \mathbf{K}_m = \mathbf{K}$ and the self-calibration Equations (2.54) reduce to
  $$\mathbf{K}\mathbf{K}^T = \kappa_j^2(\hat{\mathbf{A}}_j - \mathsf{e}_j\pi_\infty^T)\mathbf{K}\mathbf{K}^T(\hat{\mathbf{A}}_j - \mathsf{e}_j\pi_\infty^T)^T$$
  for all $j \in \{2, \ldots, m\}$. In this case, only five internal camera parameters — in practice, the five independent scalars characterizing $\mathbf{K}\mathbf{K}^T$ — are to be determined, reducing the number of unknowns to $5 + (m-1) + 3 = m + 7$. On the other hand, the self-calibration equations yield six equations for each image other than the first one. If these equations are independent for each view, a solution is determined provided $6(m-1) \geq m + 7$. Consequently, *if $m \geq 3$ images obtained by cameras with identical calibration matrices, then* the calibration matrix $\mathbf{K}$ and the plane at infinity of the scene can — *in principle* — be determined from the self-calibrations equations and *a metric reconstruction of the scene can be obtained.*
- **Images obtained by the same or identical cameras with different focal lengths.**
  If only the focal length of the camera is varying between the images, then four of the five internal parameters are the same for all cameras, which brings the total number of unknowns to $4 + m + (m-1) + 3 = 2m + 6$. Since the self-calibration

Equations (2.54) bring six equations for each image other than the first one, a solution is in principle determined provided $6(m-1) \geq 2m+6$. In other words, *when the focal length of the camera is allowed to vary between the images, then — in principle — a metric reconstruction of the scene can be obtained from $m \geq 3$ images.*

- **Known aspect ratio and skew, but unknown and different focal length and principal point.**

  When the aspect ratio and the skew of the cameras are known, but the focal length and the principal point of the cameras are unknown and possibly different for each image, only three internal parameters have to be determined for each camera. This brings the number of unknowns for all m images to $3m + (m-1) + 3 = 4m + 2$. As formula (2.54) brings six equations for each image other than the first one, a solution is in principle determined provided $6(m-1) \geq 4m+2$. In other words, *when the aspect ratio and the skew of the cameras are known, but the focal length and the principal point of the cameras are unknown and allowed to vary between the images, then — in principle — a metric reconstruction of the scene can be obtained from $m \geq 4$ images.* Note that the case of square pixels, usual with digital cameras, is a special case of this.

- **Rectangular pixels (and, hence, known skew) and unknown, but fixed aspect ratio.**

  In case the skew of the pixels is known and if the aspect ratio is identical for all cameras, but unknown, then the total number of unknowns in the self-calibration equations is $1 + 3m + (m-1) + 3 = 4m + 3$. Because there are six self-calibration equations for each image other than the first one, a solution is in principle determined provided $6(m-1) \geq 4m+3$. In other words, *when the skew of the cameras is known and if the aspect ratio is identical for all cameras, but unknown, and if the focal length and the principal point of the cameras are unknown and allowed to vary between the*

images, then — in principle — a metric reconstruction of the scene can be obtained from $m \geq 5$ images.

- **Aspect ratio and skew identical, but unknown.**

  In the situation where the aspect ratio and the skew of the cameras are identical, but unknown, two of the five internal parameters are the same for all cameras, which brings the total number of unknowns to $2 + 3m + (m - 1) + 3 = 4m + 4$. As formula (2.54) brings six equations for each image other than the first one, a solution is in principle determined provided $6(m - 1) \geq 4m + 4$. In other words, *if the aspect ratio and the skew are the same for each camera, but unknown, and when the focal length and the principal point of the camera are allowed to vary between the images, then — in principle — a metric reconstruction of the scene can be obtained from $m \geq 5$ images.*

- **Rectangular pixels or known skew.**

  If only the skew is known for each image, then four internal parameters have to be determined for each camera, which brings the total number of unknowns to $4m + (m - 1) + 3 = 5m + 2$. Since there are six self-calibration equations for each image other than the first one, a solution is in principle determined provided $6(m - 1) \geq 5m + 2$. In other words, *when only the skew is known for each image, but all the other internal parameters of the camera are unknown and allowed to vary between the images, then — in principle — a metric reconstruction of the scene can be obtained from $m \geq 8$ images.*

- **Aspect ratio unknown, but fixed.**

  When the aspect ratio of the pixels is the same for all cameras, but its value is unknown, then of the five unknown internal parameters of the cameras, one is the same for all cameras, thus bringing the total number of unknowns to $1 + 4m + (m - 1) + 3 = 5m + 3$. With six self-calibration equations for each image other than the first one, a solution is in principle determined provided $6(m - 1) \geq 5m + 3$.

> In other words, *if the aspect ratio of the pixels is the same for each camera, but its value is unknown, and when all the other internal parameters of the cameras are unknown and allowed to vary between the images, then — in principle — a metric reconstruction of the scene can be obtained from* $m \geq 9$ *images.*

In conclusion, although the self-calibration Equations (2.54) bring too few equations to allow a unique solution for the calibration matrice $\mathbf{K}_j$ of each camera and to uniquely identify the plane at infinity of the scene in the projective 3D reconstruction, in most practical situations a unique solution can be obtained by exploiting additional constraints on the internal parameters of the cameras, provided a sufficient number of images are available. The required minimum number of images as given above for different situations only is indicative in that it is correct, provided the resulting self-calibration equations are independent. In most applications this will be the case. However, one should always keep in mind that there do exist camera configurations and camera motions for which the self-calibration equations become dependent and the system is degenerate. Such special situations are referred to in the literature as *critical motion sequences*. A detailed analysis of all these cases is beyond the scope of this text, but the interested reader is referred to [9, 14, 15] for further information. Moreover, it must also be emphasized that the self-calibration equations in (2.54) are not very well suited for numerical computations. For practical use, their linear formulation (2.52) in terms of the absolute quadric is recommended. Section 4.6.2 of Section 4 discusses this issue in more detail.

2.7 Some Important Special Cases

In the preceding sections the starting point was that no information about the internal and external parameters of the cameras was available and that the positions and orientations of the cameras could be completely arbitrary. In the last section, it was observed that one has to assume some constraints on the internal parameters in order to solve the self-calibration equations, thereby still leaving the external parameters

completely unknown at the outset. In practical applications, the latter might not always be the case either and quite a bit of knowledge about the camera motion may be available. Situations in which the object or the camera has purely translated in between the acquisition of the images are quite common (e.g., camera on rails), and so are cases of pure camera rotation (e.g., camera on tripod). These simple camera motions both offer opportunities and limitations, which will be explored in this section.

Moreover, sometimes it makes more sense to first calibrate cameras internally, and then to only recover 3D structure and external parameters from the images. An example is digital surveying (3D measuring in large-scale environments like cities) with one or more fixed cameras mounted on a van. These cameras can be internally calibrated before driving off for a new surveying campaign, during which the internal parameters can be expected to remain fixed.

And, last but not least, in practical applications it may often be useful and important — apart from reconstructing the 3D structure of the scene — to obtain also information about the camera positions and orientations or about the camera parameters underlying the image data. In the computer vision literature this is referred to as *structure-and-motion*. The 3D reconstruction process outlined in the previous sections can provide such information too. The fundamental concepts needed to retrieve camera pose information will be introduced in this section as well.

2.7.1 Camera Translation and Stereo Rigs

Suppose that in between the acquisition of the first and the second image the camera only has translated and that the internal camera parameters did not change. In that case, the orientation of the camera has not changed — i.e., $\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{R}$ — and the calibration matrices are the same — i.e., $\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{K}$. Consequently, the invertible matrix $\mathbf{A}$ introduced in Section 2.4.1 reduces to:

$$\mathbf{A} = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1} = \mathbf{K} \mathbf{R}^T \mathbf{R} \mathbf{K}^{-1} = \mathbf{I}_3,$$

the 3×3-identity matrix $\mathbf{I}_3$, because $\mathbf{R}$ is an orthogonal matrix and thus $\mathbf{R}^t = \mathbf{R}^{-1}$. In other words, if one knows that the camera has

only translated between the acquisition of the images, then the infinite homography $\mathbf{A}$ is theoretically known to be the 3×3-identity matrix. Recall from Section 2.5.3 that, if $\mathbf{A}$ is known, an affine reconstruction of the scene can be computed by solving the system of affine reconstruction Equations (2.25), viz.:

$$\tilde{\rho}_1 \mathtt{m}_1 = \tilde{\mathtt{M}} \quad \text{and} \quad \tilde{\rho}_2 \mathtt{m}_2 = \mathbf{A}\tilde{\mathtt{M}} + \mathtt{e}_2 = \tilde{\mathtt{M}} + \mathtt{e}_2,$$

yielding Six equations in five unknowns. Put differently, *in case of a pure camera translation an affine reconstruction of the scene can be computed from two uncalibrated images.*

An example of an image pair, taken with a camera that has translated parallel to the object, is shown in Figure 2.15. Two views of the resulting affine 3D reconstruction are shown in Figure 2.16. The results are quite convincing for the torsos, but the homogeneous wall in the background could not be reconstructed. The difference lies in the presence or absence of texture, respectively, and the related ease or difficulty in finding corresponding points. In the untextured background all points look the same and the search for correspondences fails.

It is important to observe that without additional information about the camera(s) or the scene, one can still do no better than an affine reconstruction even if one gets additional, translated views. Indeed, the self-calibration Equations (2.45) derived in Section 2.6.3 are:

$$\mathbf{K}_j \mathbf{K}_j^T = \kappa_j^2 (\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T) \mathbf{K}_1 \mathbf{K}_1^T (\hat{\mathbf{A}}_j - \mathtt{e}_j \pi_\infty^T)^T$$

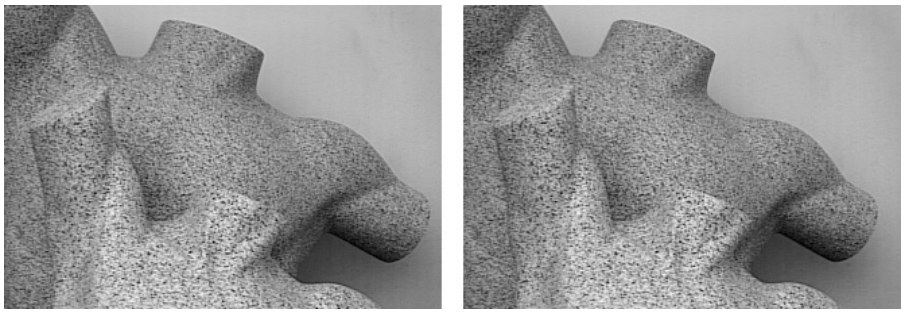

Fig. 2.15 A pair of stereo images for a scene with two torsos, where the cameras have identical settings and are purely translated with respect to each other.

Fig. 2.16 Two views of a three-dimensional affine reconstruction obtained from the stereo image pair of Figure 2.15.

for all $j \in \{2,\ldots,m\}$. In case of a translating camera with constant internal parameters, $\mathbf{K}_1 = \mathbf{K}_2 = \ldots = \mathbf{K}_m = \mathbf{K}$, $\mathbf{A}_2 = \mathbf{A}_3 = \ldots = \mathbf{A}_m = \mathbf{I}_3$ and $\pi_\infty = (0,0,0)^t$, as the scene structure has been recovered up to a 3D affine transformation (instead of only a general projective one). Hence, the self-calibration equations reduce to:

$$\mathbf{K}\mathbf{K}^T = \kappa_j^2(\mathbf{I}_3 - \mathbf{e}_j\,0^T)\mathbf{K}\mathbf{K}^T(\mathbf{I}_3 - \mathbf{e}_j\,0^T)^T$$

for all $j \in \{2,\ldots,m\}$; or equivalently, $\mathbf{K}\mathbf{K}^T = \kappa_j^2\mathbf{K}\mathbf{K}^T$, which only implies that all κ_j^2 must be equal to 1, but do not yield any information about the calibration matrix $\mathbf{K}$. In summary, *with a translating camera an affine reconstruction of the scene can be obtained already from two images, but self-calibration and metric reconstruction are not possible.*

A special case of camera translation, which is regularly used in practical applications, is a *stereo rig* with two identical cameras (i.e., cameras having the same internal parameters) in the following configuration: The optical axes of the cameras are parallel and their image planes are coplanar, with coincident x-axes. This special configuration is depicted in Figure 2.17. The distance between the two centers of projection is called the *baseline* of the stereo rig and is denoted by b. To simplify the mathematical analysis, we may, without loss of generality, let the world frame coincide with the camera-centered reference frame

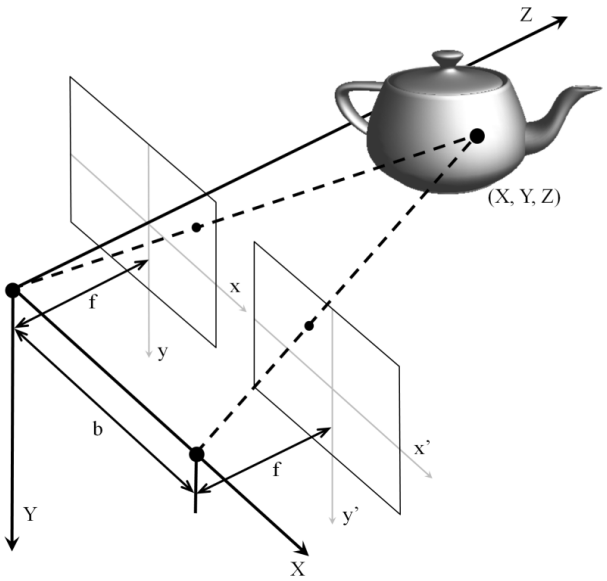

Fig. 2.17 Schematic representation of a simple stereo rig: two identical cameras having coplanar image planes and parallel axes. In particular, their x-axes are aligned.

of the left or 'first' camera. The projection equations of the stereo rig then reduce to:

$$\rho_1 \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad \text{and} \quad \rho_2 \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} X - b \\ Y \\ Z \end{pmatrix},$$

$$\text{where } \mathbf{K} = \begin{pmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \text{ is the calibration matrix of the}$$

two cameras. The pixel coordinates (x_1, y_1) and (x_2, y_2) of the projections $\mathtt{m}_1$ and $\mathtt{m}_2$ of a scene point $\mathtt{M}$ whose coordinates with respect to the world frame are (X, Y, Z), are given by:

$$\begin{cases} x_1 = \alpha_x \frac{X}{Z} + s \frac{Y}{Z} + p_x \\ y_1 = \alpha_y \frac{Y}{Z} + p_y \end{cases} \quad \text{and} \quad \begin{cases} x_2 = \alpha_x \frac{X-b}{Z} + s \frac{Y}{Z} + p_x \\ y_2 = \alpha_y \frac{Y}{Z} + p_y \end{cases}.$$

In particular, $y_1 = y_2$ and $x_1 = x_2 + \alpha_x \frac{b}{Z}$. In other words, corresponding points in the images are found on the same horizontal line (the

epipolar line for this particular setup) and the horizontal distance between them, viz. $x_1 - x_2 = \frac{\alpha_x b}{Z}$, is inversely proportional to the Z-coordinate (i.e., the projective depth) of the underlying scene point M. In the computer vision literature the difference $x_1 - x_2$ is called the *disparity* between the image points and the projective depth Z is sometimes also referred to as the *range* of the scene point. In photography the *resolution* of an image is defined as the minimum distance two points in the image have to be apart in order to be visually distinguishable. As the range Z is inversely proportional to the disparity, it follows that beyond a certain distance, depth measurement will become very coarse. Human stereo depth perception, for example, which is based on a two-eye configuration similar to this stereo rig, is limited to distances of about 10 m. Beyond, depth impressions arise from other cues. In a stereo rig the disparity between corresponding image points, apart from being inversely proportional to the projective depth Z, also is directly proportional to the baseline b and to α_x. Since α_x expresses the focal length of the cameras in number of pixels for the x-direction of the image (cf. formula (2.2) in Section 2.2.2), the depth resolution of a stereo rig can be increased by increasing one or both of these variables. Upon such a change, the same distance will correspond to a larger disparity and therefore distance sampling gets finer. It should be noted, however, that one should strike a balance between increasing resolution and keeping visible to both cameras as much of the scene as possible. Indeed, when disparities get larger, chances of finding both projections within the images diminish. Figure 2.18 shows planes of equal disparity for two focal lengths and two baseline distances. Notice how for this particular stereo setup points with identical disparities form planes at equal depth or distance from the stereo system. The same distance is seen to be sampled finer after the focal length or the baseline have been increased. A smaller part of the scene is visible to both cameras upon such a change, certainly for those parts which are close.

Similar considerations are also relevant for more general relative camera displacements than pure translations: Precisions go up as camera views differ more and projecting rays are intersecting under larger angles (i.e., if images are taken under *wide baseline* conditions). On the other hand, without intermediate views at one's disposal, there tend

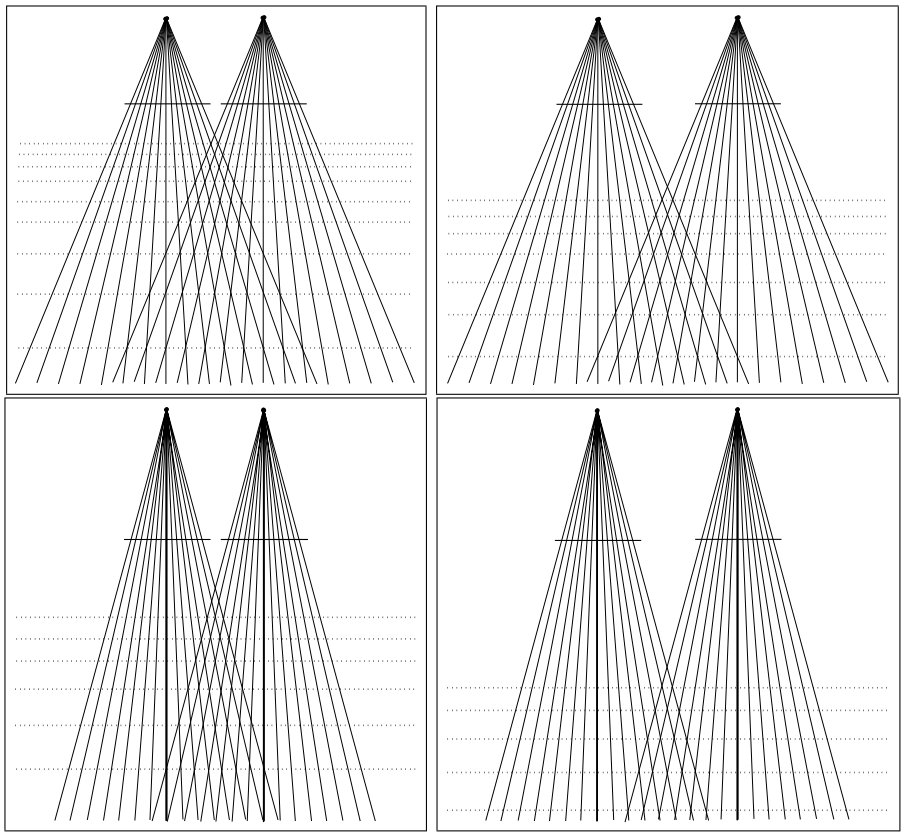

Fig. 2.18 Equal disparities correspond to points at equal distance to the stereo system (iso-depth planes): Rays that project onto points with a fixed disparity intersect in planes of constant range. At a given distance these planes get closer (i.e., sample distance more densely) if the baseline is increased (top-left to top-right), the focal length is increased (top-left to bottom-left), or both (top-left to bottom-right).

to be holes in the reconstruction, for points visible in only one or no views.

2.7.2 Pure Rotation Around the Center of Projection

Another special case of camera motion is a camera that rotates around the center of projection. This situation is depicted in Figure 2.19. The point $\mathtt{C}$ denotes the center of projection. A first image $\mathcal{I}_1$ is recorded and then the camera is rotated around the center of projection to record

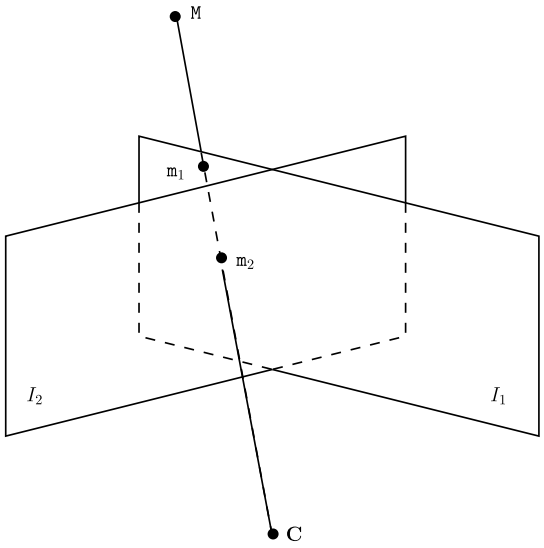

Fig. 2.19 Two images $\mathcal{I}_1$ and $\mathcal{I}_2$ are recorded by a camera which rotates around the center of projection $\mathtt{C}$. A scene point $\mathtt{M}$ is projected in the images onto the image points $\mathtt{m}_1$ and $\mathtt{m}_2$, respectively.

the second image $\mathcal{I}_2$. A scene point $\mathtt{M}$ is projected in the images $\mathcal{I}_1$ and $\mathcal{I}_2$ onto the image points $\mathtt{m}_1$ and $\mathtt{m}_2$, respectively. As the figure shows, 3D reconstruction from point correspondences is not possible in this situation. The underlying scene point $\mathtt{M}$ is to be found at the intersection of the projecting rays of $\mathtt{m}_1$ and $\mathtt{m}_2$, but these coincide.

This conclusion can also be obtained algebraically by investigating the reconstruction equations for this case. Recall from Section 2.3 that the projection equations for two cameras in general position are:

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1) \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{M} - \mathtt{C}_2).$$

If the camera performs a pure rotation around the center of projection, then $\mathtt{C}_1 = \mathtt{C}_2 = \mathtt{C}$, and the projection equations become:

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}) \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{M} - \mathtt{C}).$$

Solving the first equation for the scene point $\mathtt{M}$ and substituting this into the second equation, as in Section 2.4.1, gives

$$\rho_2 \mathtt{m}_2 = \rho_1 \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1} \mathtt{m}_1,$$

or, using $\mathbf{A} = \mathbf{K}_2\mathbf{R}_2^T\mathbf{R}_1\mathbf{K}_1^{-1}$ as before, one gets

$$\rho_2 \mathtt{m}_2 = \rho_1 \mathbf{A}\, \mathtt{m}_1. \tag{2.55}$$

This equation establishes a direct relationship between the pixel coordinates of corresponding image points. In fact, the equation states that *the second image $\mathcal{I}_2$ is a projective transformation of the first image $\mathcal{I}_1$ with the invertible matrix A introduced in Section 2.4.1 as homography matrix*. This should not come as a surprise, because looking back at Figure 2.19 and forgetting for a moment the scene point $\mathtt{M}$, one sees that image $\mathcal{I}_2$ is a perspective projection of image $\mathcal{I}_1$ with $\mathtt{C}$ as the center of projection. Hence, searching for corresponding points between two images obtained by a camera that has only rotated around the center of projection, is quite simple: One only needs to determine the invertible matrix $\mathbf{A}$. If the internal parameters of the cameras are known and if the relative orientation $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2^T$ between the two cameras is known as well, then the infinite homography $\mathbf{A}$ can be computed directly from its definition $\mathbf{A} = \mathbf{K}_2\mathbf{R}_2^T\mathbf{R}_1\mathbf{K}_1^{-1}$. However, when the internal and external camera parameters are not known, then $\mathbf{A}$ can be computed from four pairs of corresponding points between the images. Indeed, each corresponding point pair $(\mathtt{m}_1, \mathtt{m}_2) \in \mathcal{I}_1 \times \mathcal{I}_2$ must satisfy the relation $\mathbf{A}\mathtt{m}_1 = \rho\mathtt{m}_2$ for some non-zero scalar factor ρ, and thus brings a system of three linear equations in the nine unknown components of the matrix $\mathbf{A}$ and the unknown scalar ρ. As these equations are homogeneous, at least four point correspondences are needed to determine the matrix $\mathtt{A}$ up to a non-zero scalar factor. We will not pursue these computational issues any further here, since they are discussed in detail in Section 4.2.4 of Section 4. Once the matrix $\mathtt{A}$ is known, Equation (2.55) says that for every point $\mathtt{m}_1$ in the first image $\mathbf{A}\mathtt{m}_1$ are homogeneous coordinates of the corresponding point $\mathtt{m}_2$ in the second image.

Observe that Equation (2.55) actually expresses the epipolar relation between the images $\mathcal{I}_1$ and $\mathcal{I}_2$. Indeed, in its homogeneous form (2.16) the epipolar relation between corresponding image points is:

$$\rho_2 \mathtt{m}_2 = \rho_1 \mathbf{A}\, \mathtt{m}_1 + \rho_{e2}\, \mathtt{e}_2,$$

where $\rho_{e2}\, \mathtt{e}_2 = \mathbf{K}_2\mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$ is the epipole of the first camera in the second image. For a rotating camera, $\mathtt{C}_1 = \mathtt{C}_2$, which implies that

$\rho_{e2}\,\mathtt{e}_2 = \mathtt{0}$, and Equation (2.55) results. On the other hand, as the infinite homography $\mathbf{A}$ can be determined from point correspondences between the images, one could hope for an affine 3D reconstruction of the scene, as explained in Section 2.5.3. Unfortunately, this is not possible — as we already know — because in this case the affine reconstruction Equations (2.25), viz. $\tilde{\rho}_1\,\mathtt{m}_1 = \tilde{\mathtt{M}}$ and $\tilde{\rho}_2\,\mathtt{m}_2 = \mathbf{A}\,\tilde{\mathtt{M}} + \mathtt{e}_2$, reduce to $\tilde{\rho}_1\,\mathtt{m}_1 = \tilde{\mathtt{M}}$ and $\tilde{\rho}_2\,\mathtt{m}_2 = \mathbf{A}\,\tilde{\mathtt{M}}$. Taking $\tilde{\mathtt{M}} = \tilde{\rho}_1\,\mathtt{m}_1$ from the first equation and substituting it into the second one in order to compute the unknown scalar $\tilde{\rho}$ now brings back Equation (2.55), which does not allow to uniquely determine $\tilde{\rho}_1$. This proves algebraically that *3D reconstruction from point correspondences is not possible in case the images are obtained by a camera that rotates around the center of projection.*

Although 3D reconstruction is not possible from images acquired with a camera that rotates around the center of projection it is possible to recover the internal camera parameters from the images alone. Indeed, as explained in Section 2.6.4, the self-calibration Equation (2.45) result from the fundamental relation (2.49), viz:

$$\mathbf{A}^T(\mathbf{K}_2\,\mathbf{K}_2^T)^{-1}\mathbf{A} = (\mathbf{K}_1\mathbf{K}_1^T)^{-1}, \tag{2.56}$$

where $\mathbf{A}$ is the infinite homography introduced in Section 2.4.1. In case of a rotating camera the matrix $\mathbf{A}$ can be determined up to a non-zero scalar factor if at least four pairs of corresponding points are identified in the images. Let $\hat{\mathbf{A}}$ be such an estimate for $\mathbf{A}$. Then $\mathbf{A} = \kappa\,\hat{\mathbf{A}}$ for some unknown scalar κ. Substitution in Equation (2.56) and solving for $\mathbf{K}_2\,\mathbf{K}_2^T$ yields $\mathbf{K}_2\,\mathbf{K}_2^T = \kappa^2\,\hat{\mathbf{A}}(\mathbf{K}_1\,\mathbf{K}_1^T)\hat{\mathbf{A}}^T$. If the internal camera parameters have remained constant during camera rotation, then $\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{K}$ and the self-calibration equations reduce to $\mathbf{K}\,\mathbf{K}^T = \kappa^2\,\hat{\mathbf{A}}\,(\mathbf{K}\,\mathbf{K}^T)\,\hat{\mathbf{A}}^T$. Since $\hat{\mathbf{A}}$ is only determined up to a non-zero-scalar factor, one may assume without loss of generality that its determinant equals 1. Taking determinants of both sides of the self-calibration equations, it follows that $\kappa^2 = 1$, because $\mathbf{K}$ is an invertible matrix and generally has non-unit determinant. Consequently, the self-calibration equations become $\mathbf{K}\,\mathbf{K}^T = \hat{\mathbf{A}}(\mathbf{K}\,\mathbf{K}^T)\hat{\mathbf{A}}^T$ and they yield a system of six linear equations in the five unknown entries of the symmetric matrix $\mathbf{K}\,\mathbf{K}^T$. The calibration matrix $\mathbf{K}$ itself can be recovered from $\mathbf{K}\,\mathbf{K}^T$ by Cholesky factorization [3], as explained in Section 2.6.6.

In summary, *with a camera that rotates around the center of projection 3D reconstruction of the scene is not possible, but self-calibration is* [5].

2.7.3 Internally Calibrated Cameras and the Essential Matrix

In some applications, the internal parameters of the cameras may be known, through a (self-)calibration procedure applied prior to the current processing of new input images. It will be demonstrated in this section that when the internal parameters of the cameras are known, but no information about the (absolute or relative) position and orientation of the cameras is available, a metric 3D reconstruction from two images is feasible.

2.7.3.1 Known Camera Matrices and 3D Reconstruction

Consider again the projection Equation (2.20) for two cameras observing a static scene, as used in Section 2.5:

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1) \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{M} - \mathtt{C}_2). \tag{2.57}$$

If the calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are known, then the (unbiased) perspective projections of the scene point $\mathtt{M}$ in each image plane can be retrieved as $\mathtt{q}_1 = \mathbf{K}_1^{-1}\mathtt{m}_1$ and $\mathtt{q}_2 = \mathbf{K}_2^{-1}\mathtt{m}_2$, respectively. By multiplying the metric 3D reconstruction Equations (2.24) derived in Section 2.5.2, viz.:

$$\bar{\rho}_1 \mathtt{m}_1 = \mathbf{K}_1 \bar{\mathtt{M}} \quad \text{and} \quad \bar{\rho}_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \bar{\mathtt{M}} + \mathtt{e}_2,$$

on the left with $\mathbf{K}_1^{-1}$ and $\mathbf{K}_2^{-1}$, respectively, they simplify to

$$\bar{\rho}_1 \mathtt{q}_1 = \bar{\mathtt{M}} \quad \text{and} \quad \bar{\rho}_2 \mathtt{q}_2 = \mathbf{R}_2^T \mathbf{R}_1 \bar{\mathtt{M}} + \mathtt{q}_e, \tag{2.58}$$

where $\mathtt{q}_e = \mathbf{K}_2^{-1}\mathtt{e}_2$ is the (unbiased) perspective projection of the position $\mathtt{C}_1$ of the first camera onto the image plane of the second camera. It is interesting to observe that, since the epipole $\mathtt{e}_2$ is defined by formula (2.15) in Section 2.4.1 as $\rho_{e2}\mathtt{e}_2 = \mathbf{K}_2\mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$, it follows that $\rho_{e2}\mathtt{q}_e = \mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$. In other words, $\rho_{e2}\mathtt{q}_e$ gives the position $\mathtt{C}_1$ of the first camera with respect to the camera-centered reference frame of the

second camera, i.e., the relative position. And, as ρ_{e2} is the unknown scale of the metric reconstruction $\bar{\mathtt{M}}$, $\mathtt{q}_e$ in fact represents the translation direction between the two cameras in the metric 3D reconstruction of the scene.

Thus, the rotation matrix $\mathbf{R} = \mathbf{R}_2^T \mathbf{R}_1$ is the only unknown factor in the 3D reconstruction Equations (2.58) that separates us from a metric 3D reconstruction of the scene. In fact, $\mathbf{R}$ represents the orientation of the first camera in the camera-centered reference frame of the second one (cf. Section 2.2.4), i.e., the relative orientation of the cameras. It will be demonstrated below that the rotation matrix $\mathbf{R}$ can be recovered from the so-called *essential matrix* of this pair of calibrated images. But first the notion of essential matrix has to be defined.

2.7.3.2 The Essential Matrix

Recall from Section 2.4.1 that the epipolar relation (2.18) between corresponding image points is found by solving the first projection equation in formula (2.57) for $\mathtt{M}$ and substituting the resulting expression in the second projection equation, thus yielding

$$\rho_2 \mathtt{m}_2 = \rho_1 \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1} \mathtt{m}_1 + \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{C}_1 - \mathtt{C}_2)$$

(cf. formula (2.14) in Section 2.4.1). Multiplying both sides of this equation on the left by $\mathbf{K}_2^{-1}$ yields

$$\rho_2 \mathbf{K}_2^{-1} \mathtt{m}_2 = \rho_1 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1} \mathtt{m}_1 + \mathbf{R}_2^T (\mathtt{C}_1 - \mathtt{C}_2).$$

Introducing $\mathtt{q}_1 = \mathbf{K}_1^{-1}\mathtt{m}_1$, $\mathtt{q}_2 = \mathbf{K}_2^{-1}\mathtt{m}_2$, and $\mathbf{R} = \mathbf{R}_2^T \mathbf{R}_1$ as above, one gets

$$\rho_2 \mathtt{q}_2 = \rho_1 \mathbf{R} \mathtt{q}_1 + \mathbf{R}_2^T (\mathtt{C}_1 - \mathtt{C}_2).$$

Let us denote the last term in this equation by $\mathtt{t}$, then $\mathtt{t} = \mathbf{R}_2^T (\mathtt{C}_1 - \mathtt{C}_2) = \rho_{e2} \mathtt{q}_e$ represents the relative position of the first camera with respect to the second one, as explained above. The epipolar relation then is

$$\rho_2 \mathtt{q}_2 = \rho_1 \mathbf{R} \mathtt{q}_1 + \mathtt{t}. \tag{2.59}$$

From an algebraic point of view, this equation expresses that the three-vectors $\mathtt{q}_2$, $\mathbf{R}\,\mathtt{q}_1$, and $\mathtt{t}$ are linearly dependent, and hence the determinant $|\,\mathtt{q}_2\ \mathtt{t}\ \mathbf{R}\,\mathtt{q}_1\,| = 0$. Following the same reasoning as in Section 2.4.1,

$$|\,\mathtt{q}_2\ \mathtt{t}\ \mathbf{R}\mathtt{q}_1\,| = \mathtt{q}_2^T(\mathtt{t} \times \mathbf{R}\mathtt{q}_1) = \mathtt{q}_2^T\,[\mathtt{t}]_\times\,\mathbf{R}\,\mathtt{q}_1,$$

where $[\mathtt{t}]_\times$ is the skew-symmetric 3×3-matrix that represents the cross product with the three-vector $\mathtt{t}$. The 3×3-matrix $\mathbf{E} = [\mathtt{t}]_\times\,\mathbf{R}$ is known in the literature as the *essential matrix* of the (calibrated) image pair [11] and the epipolar relation between the calibrated images is expressed by the equation:

$$\mathtt{q}_2^T\,\mathbf{E}\,\mathtt{q}_1 = 0. \tag{2.60}$$

Given enough corresponding projections $\mathtt{q}_1$ and $\mathtt{q}_2$, the essential matrix $\mathbf{E}$ can be recovered up to a non-zero scalar factor from this relation in a linear manner. In fact, since $\mathbf{E} = [\mathtt{t}]_\times\,\mathbf{R}$ with $\mathtt{t}$ a three-vector and $\mathbf{R}$ a 3×3-rotation matrix, the essential matrix $\mathbf{E}$ has six degrees of freedom. Consequently, five corresponding projections $\mathtt{q}_1$ and $\mathtt{q}_2$ suffice to compute $\mathbf{E}$ up to a non-zero scalar factor [13, 12]. How this can be achieved in practice will be discussed in Section 4.6.3 of Section 4.

2.7.3.3 The Mathematical Relationship Between E and F

For the sake of completeness, it might be useful to highlight the mathematical relationship between the essential matrix $\mathbf{E}$ and the fundamental matrix $\mathbf{F}$. Since the content of this subsection is not essential for understanding the remainder of the text, readers who are primarily interested in the practice of 3D reconstruction can skip this subsection.

Recall from formula (2.60) that $\mathtt{q}_2^T\,\mathbf{E}\,\mathtt{q}_1 = 0$ describes the epipolar relation between corresponding projections $\mathtt{q}_1$ and $\mathtt{q}_2$ in the image planes of the two cameras. Substituting $\mathtt{q}_1 = \mathbf{K}_1^{-1}\mathtt{m}_1$ and $\mathtt{q}_2 = \mathbf{K}_2^{-1}\mathtt{m}_2$ in Equation (2.60) thus yields the epipolar relation between the two images in terms of the image (i.e., pixel) coordinates $\mathtt{m}_1$ and $\mathtt{m}_2$, viz.:

$$\mathtt{m}_2^T\,\mathbf{K}_2^{-T}\,\mathbf{E}\,\mathbf{K}_1^{-1}\,\mathtt{m}_1 = 0. \tag{2.61}$$

Comparison with the common form of the epipolar relation $\mathtt{m}_2^T\,\mathbf{F}\,\mathtt{m}_1 = 0$ shows that the fundamental matrix $\mathbf{F}$ is a scalar multiple of the 3×3-matrix $\mathbf{K}_2^{-T}\,\mathbf{E}\,\mathbf{K}_1^{-1}$. More precisely, recall from Section 2.4.1 that

$\mathbf{F} = [\mathtt{e}_2]_\times \mathbf{A}$ and that

$$\mathtt{m}_2^T \mathbf{F} \mathtt{m}_1 = \mathtt{m}_2^T [\mathtt{e}_2]_\times \mathbf{A} \mathtt{m}_1 = \mathtt{m}_2^T (\mathtt{e}_2 \times \mathbf{A}\mathtt{m}_1) = |\mathtt{m}_2\, \mathtt{e}_2\, \mathbf{A}\mathtt{m}_1|.$$

Substituting $\mathtt{m}_1 = \mathbf{K}_1 \mathtt{q}_1$, $\mathtt{m}_2 = \mathbf{K}_2 \mathtt{q}_2$, and $\mathtt{e}_2 = \mathbf{K}_2 \mathtt{q}_e$ in the previous expression gives

$$\mathtt{m}_2^T \mathbf{F} \mathtt{m}_1 = |\mathbf{K}_2 \mathtt{q}_2 \mathbf{K}_2 \mathtt{q}_e \mathbf{A} \mathbf{K}_1 \mathtt{q}_1| = |\mathbf{K}_2|\,|\mathtt{q}_2 \mathtt{q}_e \mathbf{K}_2^{-1} \mathbf{A} \mathbf{K}_1 \mathtt{q}_1|.$$

Using $\mathbf{A} = \mathbf{K}_2 \mathbf{R}_2^T \mathbf{R}_1 \mathbf{K}_1^{-1}$, $\mathbf{R} = \mathbf{R}_2^T \mathbf{R}_1$, and $\mathtt{t} = \rho_{e2} \mathtt{q}_e$, the right-hand side simplifies to

$$\begin{aligned} \mathtt{m}_2^T \mathbf{F} \mathtt{m}_1 &= \tfrac{|\mathbf{K}_2|}{\rho_{e2}} |\mathtt{q}_2 \mathtt{t} \mathbf{R} \mathtt{q}_1| = \tfrac{|\mathbf{K}_2|}{\rho_{e2}} \mathtt{q}_2^T (\mathtt{t} \times \mathbf{R}\mathtt{q}_1) \\ &= \tfrac{|\mathbf{K}_2|}{\rho_{e2}} \mathtt{q}_2^T [\mathtt{t}]_\times \mathbf{R} \mathtt{q}_1 = \tfrac{|\mathbf{K}_2|}{\rho_{e2}} \mathtt{q}_2^T \mathbf{E} \mathtt{q}_1. \end{aligned}$$

Substituting $\mathtt{q}_1$ and $\mathtt{q}_2$ by $\mathtt{q}_1 = \mathbf{K}_1^{-1} \mathtt{m}_1$ and $\mathtt{q}_2 = \mathbf{K}_2^{-1} \mathtt{m}_2$ again, one finally gets

$$\mathtt{m}_2^T \mathbf{F} \mathtt{m}_1 = \frac{|\mathbf{K}_2|}{\rho_{e2}} \mathtt{m}_2^T \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \mathtt{m}_1 = \mathtt{m}_2^T \left(\frac{|\mathbf{K}_2|}{\rho_{e2}} \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \right) \mathtt{m}_1.$$

Because this equality must hold for all three-vectors $\mathtt{m}_1$ and $\mathtt{m}_2$, it follows that

$$\mathbf{F} = \frac{|\mathbf{K}_2|}{\rho_{e2}} \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1}; \quad \text{or equivalently, } \mathbf{E} = \frac{\rho_{e2}}{|\mathbf{K}_2|} \mathbf{K}_2^T \mathbf{F} \mathbf{K}_1.$$

This precise relationship exists between the theoretical definitions of the fundamental matrix $\mathbf{F}$ and the essential matrix $\mathbf{E}$ only. In practice, however, the fundamental matrix $\mathbf{F}$ and the essential matrix $\mathbf{E}$ can only be recovered up to a non-zero scalar factor from point correspondences between the images. Therefore, it suffices to compute such an estimate for one of them and to use the relevant formula:

$$\hat{\mathbf{F}} = \mathbf{K}_2^{-T} \hat{\mathbf{E}} \mathbf{K}_1^{-1} \qquad \text{or} \qquad \hat{\mathbf{E}} = \mathbf{K}_2^T \hat{\mathbf{F}} \mathbf{K}_1 \tag{2.62}$$

as an estimate for the other one, as could be inferred directly from Equation (2.61).

2.7.3.4 Recovering the Relative Camera Setup from the Essential Matrix

Suppose an estimate $\hat{\mathbf{E}}$ for the essential matrix has been computed from point correspondences between the images. We will now demonstrate

how the relative setup of the cameras (i.e., the rotation matrix $\mathbf{R}$ and (the direction of) the translation vector $\mathtt{t}$ defining the essential matrix $\mathbf{E}$) can be recovered from $\hat{\mathbf{E}}$. Due to the homogeneous nature of the epipolar relation (2.60), $\hat{\mathbf{E}}$ is a non-zero scalar multiple of the essential matrix $\mathbf{E} = [\mathtt{t}]_\times \mathbf{R}$ defined above. Therefore, $\hat{\mathbf{E}} = \lambda [\mathtt{t}]_\times \mathbf{R}$ for some non-zero scalar factor λ. Observe that $\hat{\mathbf{E}}$ is a 3×3-matrix of rank 2. Indeed, $[\mathtt{t}]_\times$ is skew-symmetric and thus has rank 2 if $\mathtt{t}$ is a non-zero three-vector, whereas $\mathbf{R}$ is a rotation matrix and hence it is invertible. Moreover,

$$\begin{aligned}\hat{\mathbf{E}}^T\mathtt{t} &= \lambda([\mathtt{t}]_\times \mathbf{R})^T\mathtt{t} = \lambda\mathbf{R}^T([\mathtt{t}]_\times)^T\mathtt{t} = \lambda\mathbf{R}^T(-[\mathtt{t}]_\times)\mathtt{t} \\ &= -\lambda\mathbf{R}^T(\mathtt{t}\times\mathtt{t}) = \mathtt{0},\end{aligned}$$

which implies that the three-vector $\mathtt{t}$ belongs to the left nullspace of $\hat{\mathbf{E}}$.

Let $\hat{\mathbf{E}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the singular value decomposition of the matrix $\hat{\mathbf{E}}$ [3]. Then $\mathbf{\Sigma}$ is a diagonal matrix of rank 2 and the left nullspace of $\hat{\mathbf{E}}$ is spanned by the three-vector $\mathtt{u}_3$ constituting the third column of the orthogonal matrix $\mathbf{U}$. As $\mathtt{t}$ belongs to the left nullspace of $\hat{\mathbf{E}}$, $\mathtt{t}$ must be a scalar multiple of $\mathtt{u}_3$. This already yields $\mathtt{t}$, up to a scale.

Write $\mathtt{t} = \mu\mathtt{u}_3$ for some non-zero scalar μ. Then $\hat{\mathbf{E}} = \lambda[\mathtt{t}]_\times\mathbf{R} = \kappa[\mathtt{u}_3]_\times\mathbf{R}$ with $\kappa = \lambda\mu$ a non-zero scalar factor. Furthermore, recall from linear algebra that the singular values of $\hat{\mathbf{E}}$ are the square root of the eigenvalues of the symmetric matrix $\hat{\mathbf{E}}\hat{\mathbf{E}}^T$. Now

$$\begin{aligned}\hat{\mathbf{E}}\hat{\mathbf{E}}^T &= (\kappa[\mathtt{u}_3]_\times\mathbf{R})(\kappa[\mathtt{u}_3]_\times\mathbf{R})^T = \kappa^2[\mathtt{u}_3]_\times\mathbf{R}\mathbf{R}^T([\mathtt{u}_3]_\times)^T \\ &= -\kappa^2([\mathtt{u}_3]_\times)^2,\end{aligned}$$

where the last equality follows from the fact that $\mathbf{R}$ is a rotation matrix — and thus $\mathbf{R}\mathbf{R}^T = \mathbf{I}_3$ is the 3×3-identity matrix — and that $[\mathtt{u}_3]_\times$ is a skew-symmetric matrix — i.e., $([\mathtt{u}_3]_\times)^T = -[\mathtt{u}_3]_\times$. Because $\mathbf{U} = [\mathtt{u}_1\,\mathtt{u}_2\,\mathtt{u}_3]$ is an orthogonal matrix, the first two columns $\mathtt{u}_1$ and $\mathtt{u}_2$ of $\mathbf{U}$ are orthogonal to the third column $\mathtt{u}_3$ and, as they are all unit vectors, it follows that

$$\begin{aligned}&([\mathtt{u}_3]_\times)^2\mathtt{u}_1 = \mathtt{u}_3\times(\mathtt{u}_3\times\mathtt{u}_1) = -\mathtt{u}_1 \\ \text{and}\quad &([\mathtt{u}_3]_\times)^2\mathtt{u}_2 = \mathtt{u}_3\times(\mathtt{u}_3\times\mathtt{u}_2) = -\mathtt{u}_2.\end{aligned}$$

Consequently,

$$\begin{aligned} \hat{\mathbf{E}}\hat{\mathbf{E}}^T\mathtt{u}_1 &= -\kappa^2\,([\mathtt{u}_3]_\times)^2\,\mathtt{u}_1 = \kappa^2\,\mathtt{u}_1 \\ \text{and}\quad \hat{\mathbf{E}}\hat{\mathbf{E}}^T\mathtt{u}_2 &= -\kappa^2\,([\mathtt{u}_3]_\times)^2\,\mathtt{u}_2 = \kappa^2\,\mathtt{u}_2. \end{aligned}$$

In particular, $\mathtt{u}_1$ and $\mathtt{u}_2$ are eigenvectors of $\hat{\mathbf{E}}\hat{\mathbf{E}}^T$ with eigenvalue κ^2. Furthermore, $\mathtt{u}_3$ is an eigenvector of $\hat{\mathbf{E}}\hat{\mathbf{E}}^T$ with eigenvalue 0, because

$$\hat{\mathbf{E}}\hat{\mathbf{E}}^T\mathtt{u}_3 = -\kappa^2\,([\mathtt{u}_3]_\times)^2\,\mathtt{u}_3 = -\kappa^2\,\mathtt{u}_3 \times (\mathtt{u}_3 \times \mathtt{u}_3) = \mathtt{0}.$$

Together this proves that the diagonal matrix $\mathbf{\Sigma}$ in the singular value decomposition of the matrix $\hat{\mathbf{E}}$ equals

$$\mathbf{\Sigma} = \begin{pmatrix} |\kappa| & 0 & 0 \\ 0 & |\kappa| & 0 \\ 0 & 0 & 0 \end{pmatrix} = |\kappa| \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.63}$$

where $|\kappa|$ denotes the absolute value of κ. If we denote the columns of the orthogonal matrix $\mathbf{V}$ by $\mathtt{v}_1$, $\mathtt{v}_2$, and $\mathtt{v}_3$, respectively, then $\mathbf{V} = [\mathtt{v}_1\,\mathtt{v}_2\,\mathtt{v}_3]$ and the singular value decomposition of $\hat{\mathbf{E}}$ is given by:

$$\hat{\mathbf{E}} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T = |\kappa|\,\mathtt{u}_1\,\mathtt{v}_1^T + |\kappa|\,\mathtt{u}_2\,\mathtt{v}_2^T.$$

As $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and because $\mathtt{u}_3$ and $\mathtt{v}_3$ do not actively participate in the singular value decomposition of the rank 2 matrix $\hat{\mathbf{E}}$, we can infer them to be $\mathtt{u}_3 = \mathtt{u}_1 \times \mathtt{u}_2$ and $\mathtt{v}_3 = \mathtt{v}_1 \times \mathtt{v}_2$.

On the other hand, $\hat{\mathbf{E}} = \kappa\,[\mathtt{u}_3]_\times\,\mathbf{R}$ and our aim is to compute the unknown rotation matrix $\mathbf{R}$. To this end, we will re-express the skew-symmetric matrix $[\mathtt{u}_3]_\times$ in terms of the orthogonal matrix $\mathbf{U} = [\mathtt{u}_1\,\mathtt{u}_2\,\mathtt{u}_3]$. Recall that:

$$\begin{gathered} [\mathtt{u}_3]_\times\,\mathtt{u}_1 = \mathtt{u}_3 \times \mathtt{u}_1 = \mathtt{u}_2, \quad [\mathtt{u}_3]_\times\,\mathtt{u}_2 = \mathtt{u}_3 \times \mathtt{u}_2 = -\mathtt{u}_1 \\ \text{and}\quad [\mathtt{u}_3]_\times\,\mathtt{u}_3 = \mathtt{u}_3 \times \mathtt{u}_3 = \mathtt{0}; \end{gathered}$$

or, in matrix form,

$$\begin{aligned} [\mathtt{u}_3]_\times\,\mathbf{U} &= [\mathtt{u}_3]_\times\,[\mathtt{u}_1\,\mathtt{u}_2\,\mathtt{u}_3] = [\mathtt{u}_2\ -\mathtt{u}_1\,\mathtt{0}] \\ &= [\mathtt{u}_1\,\mathtt{u}_2\,\mathtt{u}_3] \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \mathbf{U} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

Consequently, $[\mathtt{u}_3]_\times \mathbf{U} = \mathbf{U}\mathbf{Z}$, or, equivalently, $[\mathtt{u}_3]_\times = \mathbf{U}\mathbf{Z}\mathbf{U}^T$ with

$$Z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

since $\mathbf{U}$ is an orthogonal matrix. The matrix $\hat{\mathbf{E}} = \kappa [\mathtt{u}_3]_\times \mathbf{R}$ can now be rewritten as $\hat{\mathbf{E}} = \kappa \mathbf{U}\mathbf{Z}\mathbf{U}^T\mathbf{R}$. Combining this expression with the singular value decomposition $\hat{\mathbf{E}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ yields $\kappa \mathbf{U}\mathbf{Z}\mathbf{U}^T\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. By some algebraic manipulations this equality can be simplified to

$$\begin{aligned}
&\kappa \mathbf{U}\mathbf{Z}\mathbf{U}^T\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\
&\quad \Longleftrightarrow \kappa \mathbf{Z}\mathbf{U}^T\mathbf{R} = \mathbf{\Sigma}\mathbf{V}^T \quad \text{(multiplying on the left with } \mathbf{U}^T\text{)} \\
&\quad \Longleftrightarrow \kappa \mathbf{Z}\mathbf{U}^T = \mathbf{\Sigma}\mathbf{V}^T\mathbf{R}^T \quad \text{(multiplying on the right with } \mathbf{R}^T\text{)} \\
&\quad \Longleftrightarrow \kappa \mathbf{U}\mathbf{Z}^T = \mathbf{R}\mathbf{V}\mathbf{\Sigma}^T \quad \text{(taking transposes of both sides)} \\
&\quad \Longleftrightarrow \kappa [\mathtt{u}_1\,\mathtt{u}_2\,\mathtt{u}_3] \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = |\kappa| \mathbf{R} [\mathtt{v}_1\,\mathtt{v}_2\,\mathtt{v}_3] \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(expanding } \mathbf{U}\text{, } \mathbf{Z}\text{, } \mathbf{V} \text{ and } \mathbf{\Sigma}\text{)} \\
&\quad \Longleftrightarrow \kappa [-\mathtt{u}_2\,\mathtt{u}_1\,\mathtt{0}] = |\kappa| \mathbf{R} [\mathtt{v}_1\,\mathtt{v}_2\,\mathtt{0}] \quad \text{(matrix multiplication)} \\
&\quad \Longleftrightarrow \mathbf{R}\mathtt{v}_1 = -\epsilon\,\mathtt{u}_2 \quad \text{and} \quad \mathbf{R}\mathtt{v}_2 = \epsilon\,\mathtt{u}_1 \quad \text{(equality of matrices)}
\end{aligned}$$

where $\epsilon = \frac{\kappa}{|\kappa|}$ equals 1 if κ is positive and -1 if κ is negative. Because $\mathbf{R}$ is a rotation matrix and since $\mathtt{v}_3 = \mathtt{v}_1 \times \mathtt{v}_2$ and $\mathtt{u}_3 = \mathtt{u}_1 \times \mathtt{u}_2$,

$$\begin{aligned}
\mathbf{R}\mathtt{v}_3 &= \mathbf{R}(\mathtt{v}_1 \times \mathtt{v}_2) = (\mathbf{R}\mathtt{v}_1) \times (\mathbf{R}\mathtt{v}_2) \\
&= (-\epsilon\,\mathtt{u}_2) \times (\epsilon\,\mathtt{u}_1) = \epsilon^2\,\mathtt{u}_1 \times \mathtt{u}_2 = \mathtt{u}_3.
\end{aligned}$$

But then

$$\mathbf{R}[\mathtt{v}_1\,\mathtt{v}_2\,\mathtt{v}_3] = [-\epsilon\,\mathtt{u}_2\,\epsilon\,\mathtt{u}_1\,\mathtt{u}_3] = [\mathtt{u}_1\,\mathtt{u}_2\,\mathtt{u}_3] \begin{pmatrix} 0 & \epsilon & 0 \\ -\epsilon & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

or equivalently,

$$\mathbf{R}\mathbf{V} = \mathbf{U} \begin{pmatrix} 0 & \epsilon & 0 \\ -\epsilon & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

This yields the following formula for the rotation matrix $\mathbf{R}$:

$$\mathbf{R} = \mathbf{U} \begin{pmatrix} 0 & \epsilon & 0 \\ -\epsilon & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{V}^T.$$

Observe that $\mathbf{U}$ and $\mathbf{V}$ are the orthogonal matrices in the singular value decomposition $\hat{\mathbf{E}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}$ of the matrix $\hat{\mathbf{E}}$ which is computed from point correspondences in the images, and thus are known. The scalar ϵ on the other hand, equals $\epsilon = \frac{\kappa}{|\kappa|}$ where κ is the unknown scalar factor in $\hat{\mathbf{E}} = \kappa[\mathtt{u}_3]_\times \mathbf{R}$ and hence is not known. But, as ϵ can take only values 1 and -1, the previous formula yields two possible solutions for the rotation matrix $\mathbf{R}$, viz.:

$$\hat{\mathbf{R}} = \mathbf{U} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{V}^T \quad \text{or} \quad \hat{\mathbf{R}}' = \mathbf{U} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{V}^T. \tag{2.64}$$

2.7.3.5 Euclidean 3D Reconstruction for a known Inter-Camera Distance

With the conclusion that the unknown relative rotation matrix $\mathbf{R}$ must be one of the two matrices in formula (2.64), it is now proven that a metric 3D reconstruction of the scene can be computed from two images by the metric reconstruction Equations (2.58) (still assuming that the calibration matrices of both cameras are known). If the distance between the two camera positions $\mathtt{C}_1$ and $\mathtt{C}_2$ is known too, then a Euclidean 3D reconstruction of the scene can be computed. This is quite evident from the fact that this distance allows us to fix the scale of the metric reconstruction. In the remainder of this section, we will give a more formal proof, as it also allows us to dwell further on the ambiguities that persist.

Consider again the projection Equations (2.57) for two cameras observing a static scene:

$$\rho_1 \mathtt{m}_1 = \mathbf{K}_1 \mathbf{R}_1^T (\mathtt{M} - \mathtt{C}_1), \quad \text{and} \quad \rho_2 \mathtt{m}_2 = \mathbf{K}_2 \mathbf{R}_2^T (\mathtt{M} - \mathtt{C}_2).$$

If the calibration matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are known, then the (unbiased) perspective projections $\mathtt{q}_1 = \mathbf{K}_1^{-1}\mathtt{m}_1$ and $\mathtt{q}_2 = \mathbf{K}_2^{-1}\mathtt{m}_2$ of the scene point

$\mathtt{M}$ in the respective image planes can be retrieved. Multiplying both equations on the left with $\mathbf{K}_1^{-1}$ and $\mathbf{K}_2^{-1}$, respectively, yields

$$\rho_1 \mathtt{q}_1 = \mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1) \quad \text{and} \quad \rho_2 \mathtt{q}_2 = \mathbf{R}_2^T(\mathtt{M} - \mathtt{C}_2). \tag{2.65}$$

The right-hand side of the first equation, viz. $\mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$, gives the 3D coordinates of the scene point $\mathtt{M}$ with respect to the camera-centered reference frame of the first camera (cf. Section 2.2.4). And similarly, the right-hand side of the second equation, viz. $\mathbf{R}_2^T(\mathtt{M} - \mathtt{C}_2)$, gives the 3D coordinates of the scene point $\mathtt{M}$ with respect to the camera-centered reference frame of the second camera. As argued in Section 2.5.1, it is not possible to recover absolute information about the cameras' external parameters in the real world from the previous equations alone. Therefore, the strategy proposed in Section 2.5.1 is to reconstruct the scene with respect to the camera-centered reference frame of the first camera, thus yielding the Euclidean 3D reconstruction $\mathtt{M}' = \mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$. Solving this expression for $\mathtt{M}$, gives $\mathtt{M} = \mathtt{C}_1 + \mathbf{R}_1\mathtt{M}'$ and substituting these expressions in formulas (2.65) yields

$$\rho_1 \mathtt{q}_1 = \mathtt{M}' \quad \text{and} \quad \rho_2 \mathtt{q}_2 = \mathbf{R}_2^T\mathbf{R}_1\mathtt{M}' + \mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2).$$

Using $\mathbf{R} = \mathbf{R}_2^T\mathbf{R}_1$ and $\mathtt{t} = \mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$ as in the previous subsections, one gets the following Euclidean 3D reconstruction equations for calibrated images:

$$\rho_1 \mathtt{q}_1 = \mathtt{M}' \quad \text{and} \quad \rho_2 \mathtt{q}_2 = \mathbf{R}\mathtt{M}' + \mathtt{t}. \tag{2.66}$$

It was demonstrated in the previous subsection that, if an estimate $\hat{\mathbf{E}}$ of the essential matrix $\mathbf{E}$ of the calibrated image pair is available, then an estimate for the rotation matrix $\mathbf{R}$ and for the direction of the translation vector $\mathtt{t}$ can be derived from a singular value decomposition of $\hat{\mathbf{E}}$. More precisely, if $\hat{\mathbf{E}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is a singular value decomposition of $\hat{\mathbf{E}}$, then $\mathtt{t}$ is a scalar multiple of the unit three-vector $\mathtt{u}_3$ constituting the third column of the orthogonal matrix $\mathbf{U}$ and $\mathbf{R}$ is one of the two matrices $\hat{\mathbf{R}}$ or $\hat{\mathbf{R}}'$ defined in formula (2.64). Now, since $\mathtt{u}_3$ is a unit vector, there are two possibilities for $\mathtt{t}$, namely:

$$\mathtt{t} = \|\mathtt{t}\|\mathtt{u}_3 \quad \text{or} \quad \mathtt{t}' = -\|\mathtt{t}\|\mathtt{u}_3. \tag{2.67}$$

Together, formulas (2.64) and (2.67) yield four possible, but different, candidates for the relative setup of the two cameras, viz.:

$$(\hat{\mathtt{t}}, \hat{\mathbf{R}}), (\hat{\mathtt{t}}', \hat{\mathbf{R}}), (\hat{\mathtt{t}}, \hat{\mathbf{R}}'), \quad \text{and} \quad (\hat{\mathtt{t}}', \hat{\mathbf{R}}'). \tag{2.68}$$

Observe that $\|\mathtt{t}\|$ actually is the Euclidean distance between the two camera positions $\mathtt{C}_1$ and $\mathtt{C}_2$. Indeed, $\mathtt{t} = \mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)$ and thus $||\mathtt{t}||^2 = \mathtt{t}^T\mathtt{t} = ||\,\mathtt{C}_1 - \mathtt{C}_2\,||^2$. Hence, *if the distance between the camera positions* $\mathtt{C}_1$ *and* $\mathtt{C}_2$ *is known, then each of the four possibilities in formula* (2.68) *together with the reconstruction Equation* (2.66) *yields a Euclidean 3D reconstruction of the scene* and one of these 3D reconstructions corresponds to a description of the scene in coordinates with respect to the camera-centered reference frame of the first camera. In particular, in the Euclidean reconstruction $\mathtt{M}'$ of the scene computed from Equation (2.66) the first camera is positioned at the origin and its orientation is given by the 3×3-identity matrix $\mathbf{I}_3$. The position of the second camera in the 3D reconstruction $\mathtt{M}'$, on the other hand, is given by

$$\begin{aligned}\mathbf{R}_1^T(\mathtt{C}_2 - \mathtt{C}_1) &= \mathbf{R}_1^T\mathbf{R}_2\mathbf{R}_2^T(\mathtt{C}_2 - \mathtt{C}_1)\\ &= \left(\mathbf{R}_2^T\mathbf{R}_1\right)^T\left[-\mathbf{R}_2^T(\mathtt{C}_1 - \mathtt{C}_2)\right] = -\mathbf{R}^T\mathtt{t}\end{aligned}$$

and the orientation of the second camera in the 3D reconstruction $\mathtt{M}'$ is given by $\mathbf{R}_1^T\mathbf{R}_2 = \mathbf{R}^T$. The four possibilities for a setup of the cameras which are compatible with an estimated essential matrix $\hat{\mathbf{E}}$ and which are listed in formula (2.68) correspond to four mirror symmetric configurations, as depicted in Figure 2.20. Since formulas (2.67) imply that $\hat{\mathtt{t}}' = -\hat{\mathtt{t}}$, changing $\hat{\mathtt{t}}$ into $\hat{\mathtt{t}}'$ in the relative setup of the cameras results in a reversal of the baseline of the camera pair. And, it follows from formulas (2.64) that

$$\hat{\mathbf{R}}' = \hat{\mathbf{R}}\,\mathbf{V}\begin{pmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{V}^T,$$

where the matrix product

$$\mathbf{V}\begin{pmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{V}^T$$

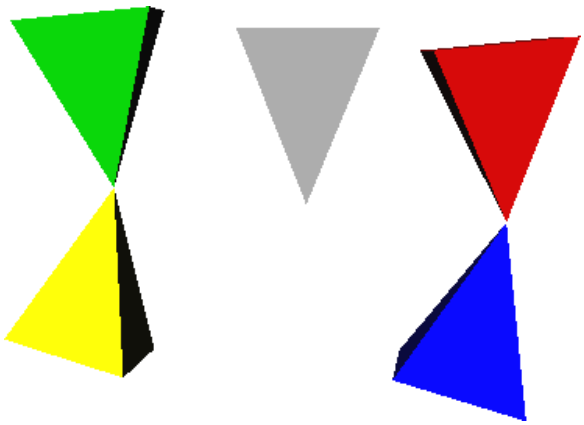

Fig. 2.20 From the singular value decomposition of (an estimate $\hat{\mathbf{E}}$ of) the essential matrix $\mathbf{E}$, four possibilities for the relative translation and rotation between the two cameras can be computed. In the figure, the first camera is depicted in grey and the other four cameras correspond to these four different possibilities. In particular, with the notations used in the text (cf. formula (2.68)), the four possible solutions for the camera setup, viz $(\hat{\mathtt{t}}, \hat{\mathbf{R}})$, $(\hat{\mathtt{t}}', \hat{\mathbf{R}})$, $(\hat{\mathtt{t}}, \hat{\mathbf{R}}')$ and $(\hat{\mathtt{t}}', \hat{\mathbf{R}}')$, correspond to the blue, yellow, red and green camera, respectively.

in the right-hand side of the equation represents a rotation through 180° about the line joining the centers of projection. Hence, changing $\hat{\mathbf{R}}$ into $\hat{\mathbf{R}}'$ in the relative setup of the cameras results in rotating the second camera 180° about the baseline of the camera pair. Moreover, given a pair $\mathtt{m}_1$ and $\mathtt{m}_2$ of corresponding points between the images, the reconstructed 3D point $\mathtt{M}'$ will be in front of both cameras in only one of the four possibilities for the camera setup computed from $\hat{\mathbf{E}}$. Hence, *to identify the correct camera setup among the candidates* (2.68) *it suffices to test for a single reconstructed point* $\mathtt{M}'$ *in which of the four possibilities it is in front of both the cameras* (i.e., for which of the four candidates the projective depths ρ_1 and ρ_2 of $\mathtt{M}'$ in the Euclidean reconstruction Equation (2.66) are both positive) [6, 10].

2.7.3.6 Metric 3D Reconstruction from Two Calibrated Images

Finally, if the distance between the camera positions $\mathtt{C}_1$ and $\mathtt{C}_2$ is not known, then the transformation $\bar{\mathtt{M}} = \frac{1}{\|\mathtt{t}\|}\mathtt{M}' = \frac{1}{\|\mathtt{t}\|}\mathbf{R}_1^T(\mathtt{M} - \mathtt{C}_1)$ still

yields a metric 3D reconstruction of the scene at scale $\|\mathtt{t}\| = \|\,\mathtt{C}_1 - \mathtt{C}_2\,\|$. Reconstruction equations for this metric reconstruction follow immediately from equations (2.66) by dividing both equations by $\|\mathtt{t}\|$, viz.:

$$\bar{\rho}_1\,\mathtt{q}_1 = \bar{\mathtt{M}} \quad \text{and} \quad \bar{\rho}_2\,\mathtt{q}_2 = \mathbf{R}\;\bar{\mathtt{M}} + \mathtt{u}, \tag{2.69}$$

where $\mathtt{u} = \frac{\mathtt{t}}{\|\mathtt{t}\|}$. It follows from formula (2.67) that the three-vector $\mathtt{u}_3$ constituting the third column of the matrix $\mathbf{U}$ in a singular value decomposition of an estimate $\hat{\mathbf{E}}$ of the essential matrix $\mathbf{E}$ yields two candidates for the unit vector $\mathtt{u}$ in the metric 3D reconstruction Equations (2.69), viz. $\mathtt{u}_3$ and $-\mathtt{u}_3$. Together with the two possibilities for the rotation matrix $\mathbf{R}$ given in formula (2.64), one gets the following four candidates for the relative setup of the two cameras in the metric reconstruction:

$$(\mathtt{u}_3, \hat{\mathbf{R}}),\;\; (-\mathtt{u}_3, \hat{\mathbf{R}}),\;\; (\mathtt{u}_3, \hat{\mathbf{R}}'),\quad \textit{and}\quad (-\mathtt{u}_3, \hat{\mathbf{R}}'). \tag{2.70}$$

As before, *the correct camera setup can easily be identified among the candidates* (2.70) *by testing for a single reconstructed point* $\bar{\mathtt{M}}$ *in which of the four possibilities it is in front of both the cameras* (i.e., for which of the four candidates the projective depths $\bar{\rho}_1$ and $\bar{\rho}_2$ of $\bar{\mathtt{M}}$ in the metric reconstruction Equations (2.69) are both positive).

It is important to note the difference between these metric 3D reconstructions and the metric 3D reconstruction described in Section 2.5.2 : Apart from different setups of the cameras, all four possible metric reconstructions described here differ from the scene by a fixed scale, which is the (unknown) distance between the two camera positions; whereas for the metric 3D reconstruction in Section 2.5.2, nothing is known or guaranteed about the actual scale of the reconstruction with respect to the original scene.

References

[1] J. Y. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/.

[2] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig," in *Computer Vision — (ECCV'92)*, pp. 563–578, vol. LNCS 588, Berlin/Heidelberg/New York/Tokyo: Springer-Verlag, 1992.

[3] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, ML, USA: The John Hopkins University Press, 1996.

[4] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *Computer Vision — (ECCV'92)*, pp. 579–587, vol. LNCS 588, Berlin/Heidelberg/New York/Tokyo: Springer-Verlag, 1992.

[5] R. Hartley, "Self-calibration from multiple views with a rotating camera," in *Computer Vision — (ECCV'94)*, pp. 471–478, vol. LNCS 800/801, Berlin/Heidelberg/New York/Tokyo: Springer-Verlag, 1994.

[6] R. Hartley, "Cheirality," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 41–61, 1998.

[7] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1309–1321, doi:10.1109/TPAMI.2007.1147, June 2007.

[8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004.

[9] F. Kahl, B. Triggs, and K. Åström, "Critical motions for auto-calibration when some intrinsic parameters can vary," *Journal of Mathematical Imaging and Vision*, vol. 13, no. 2, pp. 131–146, October 2000.

[10] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 10, pp. 133–135, 1981.
[11] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.
[12] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–777, June 2004.
[13] J. Philip, "A non-iterative algorithm for determining all essential matrices corresponding to five point Pairs," *The Photogrammetric Record*, vol. 15, no. 88, pp. 589–599, 1996.
[14] P. Sturm, "Critical motion sequences and conjugacy of ambiguous euclidean reconstructions," in *Proceedings of the 10th Scandinavian Conference on Image Analysis, Lappeenranta, Finland*, vol. I, (M. Frydrych, J. Parkkinen, and A. Visa, eds.), pp. 439–446, June 1997.
[15] P. Sturm, "Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length," in *British Machine Vision Conference, Nottingham, England*, pp. 63–72, September 1999.
[16] B. Triggs, "Autocalibration and the absolute quadric," in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 609–614, Washington, DC, USA: IEEE Computer Society, 1997.
[17] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *Radiometry*, pp. 221–244, 1992.
[18] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *ICCV*, pp. 666–673, 1999.