# 9

# 3D Vision (part 1)

**Version ch9p1 (p = prvni, means first in Czech), April 19, 1997, ToDo list**

1. *The image of the camera calibration scene is needed.*
2. *Reference to "Global human visual space is viewer centred and non-Euclidean" (Koenderink). In subsection Other vision theories.*

A number of image analysis techniques aiming at 2D images have been presented in earlier chapters. What has been overlooked hitherto, though, is the observation that the best vision system, our own, and so far unbeatable by machines, is geared to deal with the 3D world. In this chapter about 3D vision we shall fill the gap; we shall concentrate on intermediate-level vision tasks in which 3D scene properties are inferred from 2D image representations. Methods for extracting 3D information and interpreting 3D scenes will be presented.

There are several serious reasons why 3D vision using intensity images as input is regarded as difficult:

- The imaging system of a camera and the human eye performs perspective projection, which leads to considerable loss of information. All points along a line pointing from the optical center towards a scene point are projected to a single image point. We are interested in the inverse task that aims to derive 3D co-ordinates from image measurements – this task is under-constrained, and some additional information must be added to solve it unambiguously.

- The relationship between image intensity and the 3D geometry of the corresponding scene point is very complicated. The pixel intensity depends on surface reflectivity parameters, surface orientation, type and position of illuminants, and the position of the viewer. Attempting to learn about 3D geometry, surfaces orientation and depth represents another ill-conditioned task.

- The mutual occlusion of objects in the scene, and even self-occlusion of one object, further complicates the vision task.

- The presence of noise in images, and the high time complexity of many algorithms, contributes further to the problem, although this is not specific to 3D vision.

The chapter is organized as follows: In section 9.1, we shall consider various 3D vision paradigms, and Marr's theory of 3D vision from the late seventies will be explained in more detail, since even with its known limitations it still the most generally accepted paradigm. Section 9.2 explains the geometrical issues that constitute important mathematical machinery needed to solve 3D vision tasks. We present here recent research material in an uniform fashion; the geometry of one, two and three cameras and related applications are sketched. Section 9.3 tackles the relation between the intensity of a pixel in a 2D image and the 3D shape of the corresponding scene point. Section **??** considers 3D scene representations and measurements that provide 3D information. Section **??** describes various techniques for inferring shape from motion, texture, etc. Section **??** deals with model-based vision, where more a priori knowledge is at hand.

## 9.1 3D vision tasks

The field of 3D vision is young and still developing, and no unified theory is available; different research groups may have different understandings of the task. Several 3D vision tasks and related paradigms illustrate the variety of opinions:

- Marr [Mar82] defines 3D vision as '*From an image (or a series of images) of a scene, derive an accurate three-dimensional geometric description of the scene and quantitatively determine the properties of the object in the scene*'. Here, 3D vision is formulated as a 3D object reconstruction task, i.e. description of the 3D shape in a co-ordinate system independent of the viewer. One rigid object, whose separation from the background is straightforward, is assumed, and the control of the process is strictly bottom-up from an intensity image through intermediate representations. Treating 3D vision as scene recovery seems reasonable. If vision cues give us a precise representation of a 3D scene then almost all visual tasks may be carried out; the navigation of an autonomous vehicle, parts inspection, or object recognition are examples. The recovery paradigm needs to know the relation between an image and the corresponding 3D world, and thus image formation needs to be described.

- Wechsler [Wec90] stresses the control principle of the process: '*The visual system casts most visual tasks as minimization problems and solves them using distributed computation and enforcing nonaccidental, natural constraints*'. Computer vision is seen as a parallel distributed representation, plus parallel distributed processing, plus active perception. The understanding is carried in the 'perception – control – action' cycle.

- Aloimonos [Alo93] asks what principles might enable us to; (i) understand vision of living organisms, (ii) equip machines with visual capabilities. There are several types of related questions:
  - *Empirical questions* (what is?) determine how existing visual systems are designed.
  - *Normative questions* (what should be?) deals with classes of animals or robots that would be desirable.
  - *Theoretical questions* (what could be?) are interested in mechanisms that could exist in intelligent visual systems.
- Aloimonos and Shulman [AS89] see the central problem of computer vision as: '*...from one or the sequence of images of a moving or stationary object or scene taken by a monocular or polynocular moving or stationary observer, to* understand *the object or the scene and its three-dimensional properties*'. In this definition, it is the concept *understand* that makes this approach to vision different. If little a priori knowledge is available, as in human vision, then understanding is complicated. This might be seen as one limiting case; the other extreme in the complexity spectrum is, e.g. a simple object matching problem in which there are only several known possible interpretations.

*System theory* [Kli91] provides a general framework that allows us to treat understanding of complex phenomena using the machinery of mathematics. The inherent complexity of the vision task is solved here by distinguishing the object (or system or phenomenon) from the background, where 'objects' mean anything of interest to solve the task at hand. The objects and their properties need to be characterized, and a formal mathematical model is typically used for this abstraction. The model is specified by a relatively small number of parameters, which are typically estimated from the (image) data.

This methodology allows us to describe the same object using qualitatively different models (e.g. algebraic or differential equations) when *varying resolution* is used during observation. Studying changes of models with respect to several resolutions may give deeper insight into the problem.

An attempt to create a computer based vision system comprises three intertwined problems:

1. *Feature observability in images:* We need to determine whether task-relevant information will be present in the primary image data.

2. *Representation:* This problem is related to the choice of model for the observed world, at various levels of interpretation complexity.

3. *Interpretation:* This problem tackles the semantics of the data. In other words, how are data mapped to (real) world. The task is to make certain

information explicit from a mathematical model storing it in an implicit form.

Two main approaches to artificial vision, according to the flow of information and the amount of a priori knowledge, are typically considered:

1. *Reconstruction, bottom-up:* The aim is to reconstruct the 3D shape of the object from an image or set of images, which might be either intensity or range images. One extreme is given by Marr's theory [Mar82], which is strictly bottom-up with very little a priori knowledge about the objects needed. Some, more practical, approaches aim to create a 3D model from real objects using range images [FJ91, FJ92, SL92, Bow92].

2. *Recognition, top-down, model-based vision:* The a priori knowledge about the objects is expressed by means of the models of the objects, where 3D models are of particular interest [BGB79, Goa86, BJ85, FA93]. Recognition based on CAD models is of practical importance [NFJ93]. Additional constraints embedded in the model make under-determined vision tasks possible in many cases.

Some authors propose object recognition systems in which 3D models are avoided. The *priming-based (geons)* approach is based on the idea that 3D shapes can be inferred directly from 2D drawings [Bie87] – the qualitative features are called *geons*. Constituents of a single object (geons) and their spatial arrangement are pointers to a human memory and are used in the recognition process.

The *alignment of 2D views* is another option – lines or points in 2D views can be used for aligning different 2D views. The correspondence of points, lines or other features must be solved first. A linear combination of views has been used[UB91] for recognition., and various issues related to image based scene representations in which a collection of images with established correspondences is stored instead of a 3D model is considered in [BP96]. How this approach can be used for displaying a 3D scene from any viewpoint is considered in [WHH95, HLW96].

### 9.1.1 Marr's theory

Marr was a pioneer in the study of computer vision whose influence has been, and continues to be, considerable despite his early death. Critical of earlier work that, while successful in limited domains or image classes, was either empirical or unduly restrictive of the images with which it could deal, Marr proposed a more abstract and theoretical approach that permitted work to be put into a larger context. Restricting himself to the 3D interpretation of single, static scenes, Marr proposed that a computer vision system was just an example of an information processing device, and that any such device could be understood at three levels:

1. *Computational theory:* The theory describes what the device is supposed to do; what information it provides from other information provided as input. It should also describe the logic of the strategy that performs this task.

2. *Representation and algorithm*: These address precisely how the computation may be carried out; in particular, information representations and algorithms to manipulate them.

3. *Implementation*: The physical realization of the algorithm; specifically, programs and hardware.

It is stressed that it is important to be clear about which level is being addressed in attempting to solve or understand a particular problem. Marr illustrates this by noting that the effect of an after-image (induced by staring at a light bulb) is a physical effect, while the mental confusion provoked by the well known Necker cube illusion (see Figure 9.1) would appear to be at a different theoretical level entirely.

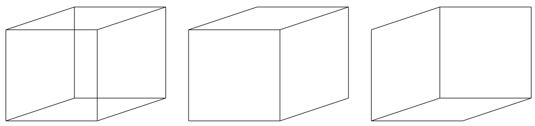

**Figure 9.1** *The Necker cube, and two possible interpretations.*

The point is then made that the lynch-pin of success is addressing the theory rather than algorithms or implementation – any number of edge detectors may be developed, each one specific to particular problems, but we would be no nearer any general understanding of how edge detection should or might be achieved. Marr remarks that the complexity of the vision task dictates a sequence of steps refining descriptions of the geometry of visible surfaces. Having derived some such description, it is then necessary to remove the dependence on the vantage point and to transform the description into an *object centered* one. The requirement, then, is to move from pixels to surface delineation, then to surface characteristic description (orientation), then to a full 3D description. These transformations are effected by moving from the 2D image to a *primal sketch* then to a $2\frac{1}{2}D$ *sketch*, and thence to a full 3D representation.

### The primal sketch

The primal sketch aims to capture, in as general a way as possible, the significant intensity changes in an image. Hitherto, such changes have been referred to as 'edges', but Marr makes the observation that this word implies a physical meaning that cannot at this stage be inferred. The first stage is to locate these changes at a range of scales (see Section **??**) – informally, a range of blurring filters are passed across the image, after which second-order zero crossings (see Section **??**) are located for each scale of blur [MH80]. The blurring recommended is a standard Gaussian filter (see equation (**??**)), while the zero crossings are located with a Laplacian operator (see equation (**??**)). The various blurring filters have the effect of isolating features of particular scales; then zero crossing evidence in the same locality at many scales provides strong evidence of a genuine physical feature in the scene.

To complete the primal sketch, these zero crossings are grouped, according to their location and orientations, to provide information about tokens in the image (edges, bars and blobs) that may help provide later information about (3D) orientation of scene surfaces. The grouping phase, paying attention to the evidence from various scales, extracts tokens that are likely to represent surfaces in the real world.

It is of interest to note that there is strong evidence for the existence of the various components used to build the primal sketch in the human visual system – we too engage in detection of features at various scales, the location of sharp intensity changes and their subsequent grouping into tokens.

### The $2\frac{1}{2}$D sketch

The $2\frac{1}{2}$D sketch reconstructs the relative distances from the viewer of surfaces detected in the scene, and may be called a *depth map*. Observe that the output of this phase uses as input features detected in the preceeding one, but that in itself it does not give us a 3D reconstruction. In this sense it is midway between 2D and 3D representations, and in particular, nothing can be said about the "other side" of any objects in view. Instead, it may be the derivation of a surface normal associated with each likely surface detected in the primal sketch, and there may be an implicit improvement in the quality of this information.

There are various routes to the $2\frac{1}{2}$D sketch, but their common thread is the continuation of the bottom-up approach in that they do not exploit any knowledge about scene contents, but rather employ additional clues such as knowledge about the nature of lighting or motion effects, and are thus generally applicable and not domain-specific. The main approaches are known as *"Shape from X"* techniques, and are described in Section **??**. At the conclusion of this phase, the representation is still in viewer-centered co-ordinates.

### The 3D representation

At this stage the Marr paradigm overlaps with top-down, model-based approaches. It is required to take the evidence derived so far and identify objects within it. This can only be achieved with some knowledge about what "objects" are, and, consequently, some means of describing them. The important point is that this is a transition to an object centered co-ordinate system, allowing object descriptions to be viewer independent.

This is the most difficult phase and successful implementation is remote, especially compared to the success seen with the derivation of the primal and $2\frac{1}{2}$D sketches – specifying what is required, however, has been very successful in guiding computer vision research since the paradigm was formulated. Unlike earlier stages, there is little physiological guidance that can be used to design algorithms since this level of human vision is not well understood. Marr observes that the target co-ordinate system(s) should be modular in the sense that each "object" should be treated differently, rather than employing one global co-ordinate system (usually viewer centered). This prevents having to consider the orientation of model components with respect to the whole. It is further observed that a set of *volumetric* primitives is likely to be of value in representing models (in contrast to surface-based descriptions). Representations based on an object's "natural" axes, derived from symmetries, or the orientation of stick features, are likely to be of greater use.

The Marr paradigm advocates a set of relatively independent modules; the low-level modules aim to recover a meaningful description of the input intensity image, the middle-level modules use different cues such as intensity changes, contours, texture, motion to recover shape or location in space. It was shown later [BPT88, AR94] that most low-level and middle-level tasks are ill-posed, with no unique solution; one popular way developed in the eighties to make the task well-posed is *regularization* [TA77, PTK85]. A constraint requiring continuity and smoothness of the solution is often added.

### 9.1.2 Other vision paradigms: Active and purposive vision

When consistent geometric information has to be explicitly modelled (as for manipulation of the object), an object-centered co-ordinate system seems to be appropriate. It is not certain that Marr's attempt to create object-centered co-ordinates is confirmed in biological vision; for example, Koenderink shows that the the global human visual space is viewer-centered and non-Euclidean [**?**]. For small objects, the existence of an object-centered reference frame has not been confirmed in psychological studies.

There are currently two schools trying to explain the vision mechanism:

- The first and older one tries to use explicit metric information in the early stages of the visual task (lines, curvatures, normals, etc.). Geometry is typically extracted in a bottom-up fashion without any information about the purpose of this representation. The output is a geometric model.

- The second and younger school does not extract metric (geometric) information from visual data until needed for a specific task. Data are collected in a systematic way to ensure all the object's features are present in the data, but may remain uninterpreted until a specific task is involved. A database or collection of intrinsic images (or views) is the model.

Many traditional computer vision systems and theories capture data with cameras with fixed characteristics. The same holds for traditional theories, e.g. Marr's observer is static. Some researchers advocate for *active perception* [Baj88] and purposive vision [Alo93]. In an active vision system, the characteristics of the data acquisition are dynamically controlled by the scene interpretation. Many visual tasks tend to be simpler if the observer is active and controls its visual sensors. The controlled eye (or camera) movement is an example. If there is not enough data to interpret the scene the camera can look at it from other viewpoint. In other words, the active vision is an intelligent data acquisition controlled by the measured, partially interpreted scene parameters and their errors from the scene. The active vision is a vivid area of research.

The active approach can make most ill-posed vision tasks tractable. To give the reader the overview, let us summarize in tabular form [AR94] how an active observer changes ill-posed to well-posed tasks, see Tab. 9.1.

| *Task* | *Passive observer* | *Active observer* |
|---|---|---|
| Shape from shading | Ill-posed. Regularization helps but an unique solution is not guaranteed due to non-linearities. | Well-posed. Stable. Unique solution. Linear equations. |
| Shape from contour | Ill-posed. Regularization solution not formulated yet. Solution exists only for very special cases. | Well-posed. Unique solution for monocular or binocular observer. |
| Shape from texture | Ill-posed. Assumptions about texture needed. | Well-posed without assumptions. |
| Structure from motion | Well-posed but unstable. | Well-posed and stable. Quadratic constraints. simple solution. |

**Table 9.1** *Active vision makes vision tasks well-posed.*

It has been generally accepted in the vision community that accurate shape recovery from intensity images is difficult. The Marr paradigm is a nice theoretic framework, but unfortunately does not lead to successful vision applications performing, e.g., recognition and navigation tasks.

There is no established theory that provides a mathematical (computational) model explaining the 'understanding' aspects of human vision; a recent account of the topic is [Ull96]. Two recent developments towards new vision theory are:

- *Qualitative vision*, which looks for a qualitative description of objects or scenes [**?**]. The motivation is not to represent geometry that is not needed for qualitative (non-geometric) tasks or decisions. Further, qualitative information is more invariant to various unwanted transformations (e.g. slightly differing viewpoints) or noise than quantitative ones. Qualitativeness (or invariance) enables interpretation of observed events at several levels of complexity. Note that the human eye does not give extremely precise measurements either; a vision algorithm should look for qualities in images, e.g. convex and concave surface patches in range data [BJ88].

- The *purposive paradigm*, which may help to come up with simpler solutions [Alo92]. The key question is to identify the goal of the task, the motivation being to ease the task by making explicit just that piece of information that is needed. Collision avoidance for autonomous vehicle navigation is an example where precise shape description is not needed. The approach may be heterogeneous and a qualitative answer may be sufficient in some cases. The paradigm does not yet have a solid theoretical basis, but the study of biological vision is a rich source of inspiration. This shift of research attention resulted in many successful vision applications where no precise geometric description is necessary. Examples are collision avoidance, autonomous vehicle navigation, object tracking, etc. [**?**, **?**, **?**].

There are other vision tasks that need complete geometric 3D models, for example, to create a 3D CAD model from a real object, say a clay model created by a human designer. Other applications are in virtual reality systems where interaction among real and virtual objects is needed. Some object recognition tasks use full 3D models as well.

## 9.2 Geometry for 3D vision

### 9.2.1 Basics of projective geometry

The basic sensor that provides computer vision with information about the surrounding 3D world is a television camera. Here, stressing the geometric aspect, we will explain how to use 2D image information for automated measurement of the 3D world, where measurements of 3D co-ordinates of points or distances from 2D images are of importance. We require to study *perspective projection* (called also central projection), which describes image formation by a pinhole camera or a thin lens. The reader surely recalls from the high school course on geometric optics that explains how an image is formed by a pinhole camera or a thin lens. Parallel lines in the world do not remain parallel in a perspective image

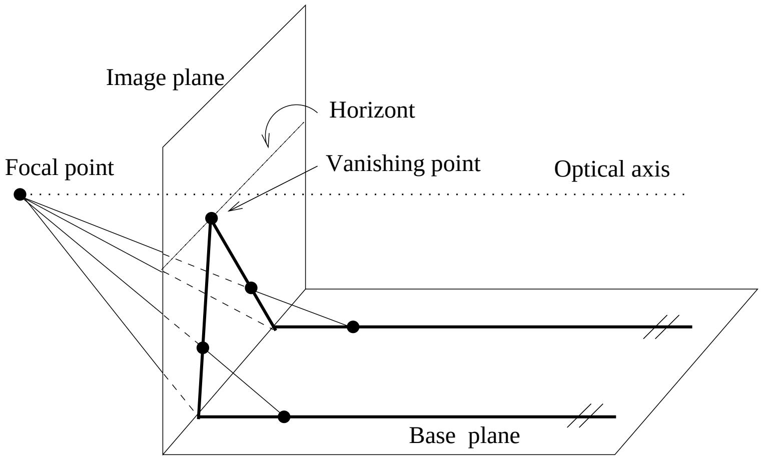


**Figure 9.2** *Perspective projection of parallel lines.*

– consider, for example, a view along a railway or into a long corridor. Figure 9.2 illustrates this, and also some commonly used terms are introduced.

We begin with a concise introduction to basic notation and the definitions of projective space [SK63, Fau93, Moh93]. Consider $(n+1)$ dimensional space without its origin $\mathcal{R}^{n+1} - \{(0,\ldots,0)\}$., and define an equivalence relation

$$[x_1,\ldots,x_{n+1}]^T \equiv [x'_1,\ldots,x'_{n+1}]^T \text{ iff}$$
$$\exists\, \alpha \neq 0 : \; [x_1,\ldots,x_{n+1}]^T = \alpha\,[x'_1,\ldots,x'_{n+1}]^T \tag{9.1}$$

The projective space $\mathcal{P}^n$ is the quotient space of this equivalence relation. Points in the projective space are expressed in *homogeneous* (also projective) co-ordinates, which we will denote in bold with a tilde, e.g. $\tilde{\mathbf{x}}$. Such points are often shown with the number one on the rightmost position, e.g. $[x'_1,\ldots,x'_n,1]^T$. This point is equivalent to any point that differs only by nonzero scaling.

We are more accustomed to $n$-dimensional Euclidean space $\mathcal{R}^n$. The one-to-one mapping from the $\mathcal{R}^n$ into $\mathcal{P}^n$ is given by

$$[x_1,\ldots,x_n]^T \to [x_1,\ldots,x_n,1]^T \tag{9.2}$$

Only the points $[x_1,\ldots,x_n,0]^T$ do not have an Euclidean counterpart. It is easy to demonstrate that they represent points at infinity in a particular direction. Consider $[x_1,\ldots,x_n,0]^T$ as a limiting case of $[x_1,\ldots,x_n,\alpha]^T$ that is projectively equivalent to $[x_1/\alpha,\ldots,x_n/\alpha,1]^T$, and assume that $\alpha \to 0$. This corresponds to a point in $\mathcal{R}^n$ going to infinity in the direction of the radius vector $[x_1/\alpha,\ldots,x_n/\alpha] \in \mathcal{R}^n$.

A *colineation*, or projective transformation, is any mapping $\mathcal{P}^n \to \mathcal{P}^n$ that is defined by a regular $(n+1) \times (n+1)$ matrix $\mathbf{A}$, $\tilde{\mathbf{y}} = A\,\tilde{\mathbf{x}}$. Note that the matrix

$A$ is defined up to a scale factor. Colineations map hyperplanes to hyperplanes; a special case is the mapping of lines to lines that is often used in computer vision.

### 9.2.2 The single perspective camera

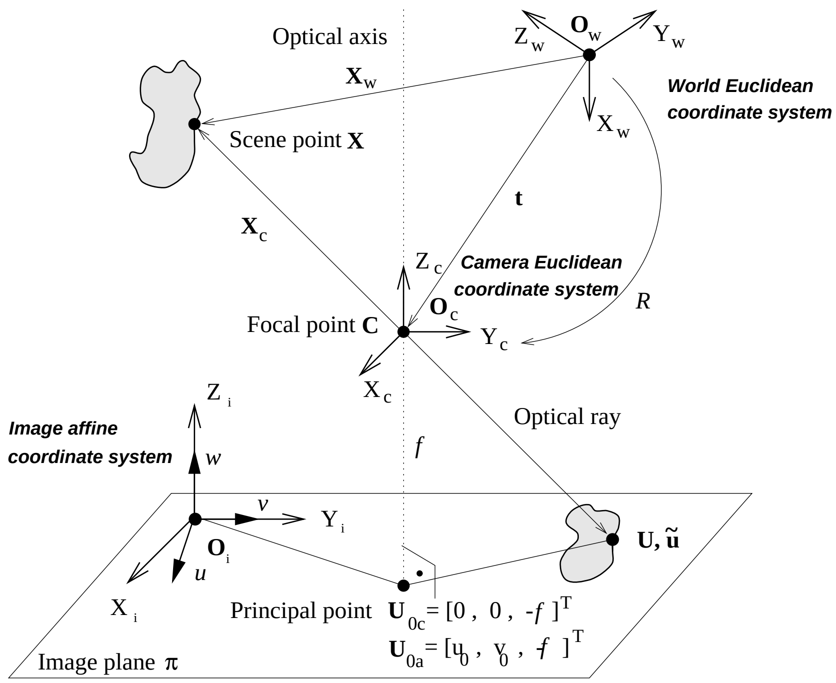


**Figure 9.3** *The geometry of a linear perspective camera.*

Consider the case of one camera with a thin lens. This pinhole model is the simplest approximation that is suitable for many computer vision applications. The pinhole camera performs perspective projection. The geometry of the device is depicted in Figure 9.3; the plane on the bottom is an *image plane* $\pi$ to which the real world projects, and the vertical dotted line is the *optical axis*. The lens is positioned perpendicularly to the optical axis at the *focal point* $\mathbf{C}$ (also called the *optical center*). The focal length $f$ (sometimes called the principal axis distance [Moh93]) is a parameter of the lens.

The projection is performed by an optical ray (also a light beam) reflected from a scene point $\mathbf{X}$ (top left in Figure 9.3). The optical ray passes through the optical center $\mathbf{C}$ and hits the image plane at the point $\mathbf{U}$.

For further explanation, we need to define four co-ordinate systems:

1. The *world Euclidean co-ordinate system* (subscript $_w$) has origin at the point $\mathbf{O}_w$. Points $\mathbf{X}$, $\mathbf{U}$ are expressed in the world co-ordinate system.

2. The *camera Euclidean co-ordinate system* (subscript $_c$) has the focal point $\mathbf{C} \equiv \mathbf{O}_c$ as its origin. The co-ordinate axis $Z_c$ is aligned with the optical axis and points away from the image plane.

   There is a unique relation between world and camera co-ordinate systems. We can align the world to camera co-ordinates by performing an Euclidean transformation consisting of a translation $\mathbf{t}$ and a rotation $R$.

3. The *image Euclidean co-ordinate system* (subscript $_i$) has axes aligned with the camera co-ordinate system, with $X_i$, $Y_i$ lying in the image plane.

4. The *image affine co-ordinate system* (subscript $_a$) has co-ordinate axes $u$, $v$, $w$, and origin $\mathbf{O}_i$ coincident with the origin of the image Euclidean co-ordinate system. The axes $w$, $v$ are aligned with the axes $Z_i$, $X_i$, but the axis $u$ may have a different orientation to the axis $Y_i$.

   The reason for introducing the camera affine co-ordinates is the fact that in general pixels need not perpendicular and axes can be scaled differently. The affine co-ordinate system is induced by the arrangement of the retina.

A camera performs a linear transformation from the 3D projective space $\mathcal{P}^3$ to the 2D projective space $\mathcal{P}^2$.

A scene point $\mathbf{X}$ is expressed in the world Euclidean co-ordinate system as a $3 \times 1$ vector. To express the same point in the camera Euclidean co-ordinate system, i.e. $\mathbf{X}_c$, we have to rotate it as specified by the matrix $R$ and translate it by subtracting vector $\mathbf{t}$

$$\mathbf{X}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R\,(\mathbf{X}_w - \mathbf{t}) \tag{9.3}$$

The point $\mathbf{X}_c$ is projected to the image plane $\pi$ as point $\mathbf{U}_c$. The $x$ and $y$ co-ordinates of the projected point can be derived from the similar triangles illustrated in Figure 9.4

$$\mathbf{U}_c = \left[\ \frac{-f x_c}{z_c},\quad \frac{-f y_c}{z_c},\quad -f\ \right]^T \tag{9.4}$$

It remains to derive where the projected point $\mathbf{U}_c$ is positioned in the image affine co-ordinate system, i.e. to determine the co-ordinates which the real camera actually delivers.

The image affine co-ordinate system, with origin at the top left corner of the image, represents a shear and rescaling (often called the aspect ratio) of the image Euclidean co-ordinate system. The principal point $\mathbf{U}_0$ – sometimes called the center of the image in camera calibration procedures is the intersection of the optical axis with the image plane $\pi$. The principal point $\mathbf{U}_0$ is expressed in the image affine co-ordinate system as $\mathbf{U}_{0a} = [u_0, v_0, 0]^T$.

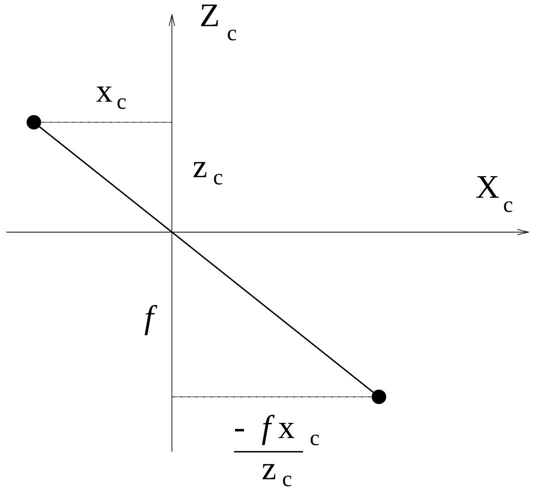


**Figure 9.4** *Calculation of the co-ordinates of the projected point.*

The projected point can be represented in the 2D image plane $\pi$ in homogeneous co-ordinates as $\tilde{\mathbf{u}} = [U, V, W]^T$, and its 2D Euclidean counterpart is $\mathbf{u} = [u, v]^T = [\frac{U}{W}, \frac{V}{W}]^T$. Homogeneous co-ordinates allow us to express the affine transformation as a multiplication by a single $3 \times 3$ matrix where unknowns $a$, $b$, $c$ describe the shear together with scaling along co-ordinate axes, and $u_0$ and $v_0$ give the affine co-ordinates of the principal point in the image.

$$\tilde{\mathbf{u}} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} a & b & -u_0 \\ 0 & c & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{-f x_c}{z_c} \\ \frac{-f y_c}{z_c} \\ 1 \end{bmatrix} \tag{9.5}$$

We aim to collect all constants in this matrix, sometimes called the *camera calibration matrix K*. Since homogeneous co-ordinates are in use, the equation can be multiplied by any nonzero constant; thus we multiply by $z_c$ to remove it. Thus we rewrite

$$\begin{aligned} z_c\, \tilde{\mathbf{u}} &= z_c \begin{bmatrix} -fa & -fb & -u_0 \\ 0 & -fc & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ 1 \end{bmatrix} = \begin{bmatrix} -fa & -fb & -u_0 \\ 0 & -fc & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \\ &= \begin{bmatrix} -fa & -fb & -u_0 \\ 0 & -fc & -v_0 \\ 0 & 0 & 1 \end{bmatrix} R\,(\mathbf{X}_w - \mathbf{t}) = K \;\; R\,(\mathbf{X}_w - \mathbf{t}) \end{aligned} \tag{9.6}$$

The *extrinsic parameters* of the camera depend on the orientation of the camera Euclidean co-ordinates with respect to the world Euclidean co-ordinate system (see Figure 9.3). This relation is given in equation (9.6) by matrices $R$ and $\mathbf{t}$. The rotation matrix $R$ expresses three elementary rotations of the co-ordinate axes – rotations along the axes $x$, $y$, and $z$ are termed pan, tilt, and roll, respectively.

The translation vector $\mathbf{t}$ gives three elements of the translation of the origin of the world co-ordinate system with respect to the camera co-ordinate system. Thus there are six extrinsic camera parameters; three rotations and three translations.

The camera calibration matrix $K$ is upper triangular as can be seen from equation (9.6). The coefficients of this matrix are called *intrinsic parameters* of the camera, and describe the specific camera independent on its position and orientation in space. If the intrinsic parameters are known, a metric measurement can be performed from images. Assume momentarily the simple case in which the world co-ordinates coincide with the camera co-ordinates, meaning that $\mathbf{X}_w = \mathbf{X}_c$. Then equation (9.6) simplifies to

$$z_c\,\tilde{\mathbf{u}} = z_c \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} -fa & -fb & -u_0 \\ 0 & -fc & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{9.7}$$

We can write two separate equations for $u$ and $v$

$$\begin{array}{llllllll} u & = \frac{U}{W} = & -fa\,\frac{x_c}{z_c} & - & fb\,\frac{y_c}{z_c} - u_0 & = & \alpha_u\,\frac{x_c}{z_c} & + \ \alpha_{shear}\ \frac{y_c}{z_c} \ - \ u_0 \\ v & = \frac{U}{W} = & & - & fc\,\frac{y_c}{z_c} - v_0 & = & & \alpha_v\ \frac{y_c}{z_c} \ - \ v_0 \,. \end{array} \tag{9.8}$$

where we make the substitutions $\alpha_u = -fa$, $\alpha_{shear} = -fb$, and $\alpha_v = -fc$. Thus we have five intrinsic parameters, all given in pixels. The formulae also give the interpretation of the intrinsic parameters: $\alpha_u$ represents scaling in the $u$ axis, measuring $F$ in pixels along the $u$ axis, and $\alpha_v$ similarly specifies $f$ in pixels along the $v$-axis. $\alpha_{shear}$ measures in pixels in the $v$-axis direction how much is the focal length $f$ coincident with $u$-axis slanted from the $Y_i$-axis.

This completes the description of the extrinsic and intrinsic camera parameters, and we can return to the general case given by the equation (9.6). If we express the scene point in homogeneous co-ordinates $\tilde{\mathbf{X}}_w = [\mathbf{X}_w, 1]^T$ we can write the perspective projection using a single $3 \times 4$ matrix. The leftmost $3 \times 3$ submatrix describes a rotation and the rightmost column a translation The delimiter $|$ denotes that the matrix is composed of two submatrices.

$$\tilde{\mathbf{u}} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = [KR\,|-K\ R\,\mathbf{t}] \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix} = M \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix} = M\tilde{\mathbf{X}}_w \tag{9.9}$$

where $\tilde{\mathbf{X}}$ is the 3D scene point in homogeneous co-ordinates. The matrix $M$ is called the *projective matrix* (also camera matrix). It can be seen that the camera performs a linear projective transformation from the 3D projective space $\mathcal{P}^3$ to the 2D projective plane $\mathcal{P}^2$. Notice that the introduction of projective space and homogeneous co-ordinates made the expressions simpler. Instead of the nonlinear equation (9.4), we obtained the linear equation (9.9).

The $3 \times 3$ submatrix of the projective matrix $M$ consisting of three leftmost columns is regular, i.e. its determinant is non-zero. The scene point $\tilde{\mathbf{X}}_w$

is expressed up to scale in homogeneous co-ordinates (recall that projection is expressed in the projection space) and thus all $\alpha\, M$ are equivalent for $\alpha \neq 0$.

Sometimes the simplest form of the projection matrix $M$ is used.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{9.10}$$

This special matrix corresponds to the *normalized camera co-ordinate system* [Fau93], in which the specific parameters of the camera can be ignored. This is useful when the properties of stereo and motion are to be explained in a simple way and independently of the specific camera.

### 9.2.3 An overview of single camera calibration

The calibration of one camera is a procedure that allows us to set numeric values in the camera calibration matrix $K$ (equation (9.6)) or the projective matrix $M$ (equation (9.9)). The first case is applicable when we want the intrinsic camera parameters only. If the camera is calibrated, and a point in the image is known, the corresponding line (ray) in camera-centered space is uniquely determined. The second case covers both intrinsic and extrinsic parameters.

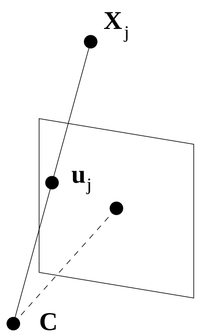


**Figure 9.5** *Camera calibration from a known scene. A minimum of six corresponding pairs of scene points $\mathbf{X}_j$ and image points $\mathbf{u}_j$ are needed to calibrate the camera.*

We consider first basic approaches to the calibration of a single camera to give an overview of the state of the art of this developing branch of computer vision. Then we will consider some basic techniques in more detail. There are two main cases:

1. *Known scene:* Here, a set of $n$ non-degenerate (not co-planar) points lies in the 3D world, and the corresponding 2D image points are known (see

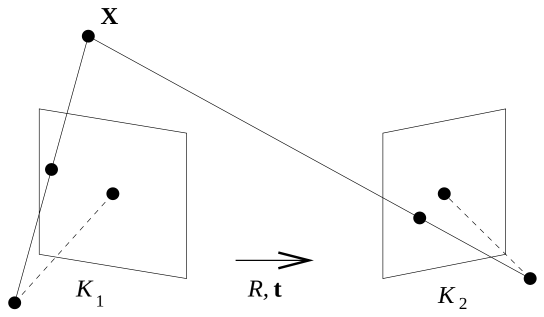


**Figure 9.6** *Camera calibration from an unknown scene. At least two views are needed. It is assumed that the intrinsic parameters of the camera do not change, so* $K_1 = K_2$.

Figure 9.5). This case applies if for a set of $n$ points in a 3D Each correspondence between a 3D scene and 2D image point provides one equation

$$\alpha_j \tilde{\mathbf{u}}_j = M \begin{bmatrix} \mathbf{X}_j \\ 1 \end{bmatrix} \tag{9.11}$$

The solution [Fau93] solves an over-determined system of linear equations. The main disadvantage is that the scene must be known., for which special calibration objects are often used.

2. *Unknown scene:* if the scene is "unknown,, more views of the scene are needed to calibrate the camera (see Figure 9.6). The intrinsic camera parameters will not change for different views, and the correspondence between image points in different views must be established.

   There are two cases:

   (a) *Known camera motion:* Three cases can be distinguished according to the known motion constraint:

      i. *Both rotation and translation:* This general case of arbitrary known motion from one view to another has been solved [HMDB95].

      ii. *Pure rotation:* If camera motion is restricted to pure rotation, the solution is given by [Har94].

      iii. *Pure translation:* The linear solution (pure translation) is due to [PH95].

   (b) *Unknown camera motion:* This is the most general case when there is no a priori knowledge about motion, sometimes called *camera self-calibration*. At least three views are needed and the solution is non-linear [MF92]. Calibration from an unknown scene is still considered numerically hard, and will not be considered here (although see, for example, [?] for a consideration of this problem).

### 9.2.4 Calibration of one camera from the known scene

Considering the case of camera calibration from a known scene in more detail, note this is typically a two stage process. Firstly, the projection matrix $M$ is estimated from the co-ordinates of points with known scene positions. Secondly, the extrinsic and intrinsic parameters are estimated from $M$. The second step is not always needed – the case of stereo vision is an example.

To obtain $M$, observe that each known scene point $\mathbf{X} = [x, y, z]^T$ and its corresponding 2D image point $[u, v]^T$ give one equation (9.11) – we seek the numerical values $m_{ij}$ in the $3\times 4$ projection matrix $M$. Expanding from Equation (9.11),

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{9.12}$$

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = \begin{bmatrix} m_{11}x & + & m_{12}y & + & m_{13}z & + & m_{14} \\ m_{21}x & + & m_{22}y & + & m_{23}z & + & m_{24} \\ m_{31}x & + & m_{32}y & + & m_{33}z & + & m_{34} \end{bmatrix} \tag{9.13}$$

$$\begin{aligned} u(m_{31}x + m_{32}y + m_{33}z + m_{34}) &= m_{11}x + m_{12}y + m_{13}z + m_{14} \\ v(m_{31}x + m_{32}y + m_{33}z + m_{34}) &= m_{21}x + m_{22}y + m_{23}z + m_{24} \end{aligned} \tag{9.14}$$

Thus we obtain two linear equations, each in 12 unknowns $m_{11}, \ldots, m_{34}$, for each known corresponding scene and image point. If $n$ such points are available, we can write the equations 9.14 as a $2n \times 12$ matrix,

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz & -u \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz & -v \\ & & & & & & \vdots & & & & & \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ \vdots \\ m_{34} \end{bmatrix} = 0 \tag{9.15}$$

The matrix $M$ actually has only 11 unknown parameters due to the unknown scaling factor, since homogeneous co-ordinates were used – the proof (named rank analysis) is given in [Fau93]. To generate a solution, at least six known corresponding scene and image points are required. Typically, more points are used and the over-determined equation (9.15) is solved using a robust least squares method to correct for noise in measurements. The result of the calculation is the projective matrix $M$.

To separate the extrinsic parameters (the rotation $R$ and translation $\mathbf{t}$) from the estimated projection matrix $M$, recall that the projection matrix can be written as;

$$M = [KR \,|\, -K\ R\mathbf{t}] = [A\,|\mathbf{b}] \tag{9.16}$$

The $3 \times 3$ submatrix is denoted as $A$, and the rightmost column as $\mathbf{b}$.

Determining the translation vector is easy; we substituted $A = KR$ in equation (9.16), and so can write $\mathbf{t} = -A^{-1}\mathbf{b}$.

To determine $R$, note that the calibration matrix is upper triangular and the rotation matrix is orthogonal. The matrix factorization method called QR decomposition [PT+92, GL89] will decompose $A$ into a product of two such matrices, and hence recover $K$ and $R$.

Alternatively, we can use *Singular Value Decomposition (SVD)* to perform similar job. [1] SVD is a general tool that we shall refer to again in the solution of geometrical problems associated with 3D vision.

So far, we have assumed that the lens performs ideal central projection as the pinhole camera does. This is not the case with the real lenses. Such a typical lens performs distortion of several pixels. A human observer does not notice it if he looks at general scene. In the case an image is used for measurements, the distortion from the idealized pinhole model should be compensated.

When *calibrating a real camera* the more realistic model of the lens includes two distortion components. First, the *radial distortion* bends the ray more or less than in the idealistic case. Second, the *decentering* displaces the principal point from the optical axis.

Recall that the five intrinsic camera parameters were introduced in equation (9.8). Let us note that the focal length $f$ of the lens is replaced by a parameter called the *camera constant* here. Ideally, the focal length and the camera constant should be the same. In reality, this is true when the lens is focused at infinity. Otherwise, the camera constant is slightly less than the focal length. Similarly, the coordinates of the principal point can slightly change from the ideal intersection of the optical axis with the image plane.

The main trick of the intrinsic parameters calibration is to observe a known calibration image with some regular pattern, e.g. blobs or lines covering the whole image. The observed distortions of the pattern allows to estimate the intrinsic camera parameters.

[1] SVD is a powerful linear algebra technique for solving linear equations in the least square sense, and works even for singular matrices or matrices numerically close to singular. The basic information needed to use SVD can be found in [PT+92], and a rigorous mathematical treatment is given in [GL89]. Most software packages such as MATLAB contain SVD.

SVD proceeds by noting that any $m \times n$ matrix $A$, $m \geq n$ can be decomposed into a product of three matrices

$$A = UDV^T \tag{9.17}$$

in which $U$ has orthonormal columns, W is non-negative diagonal, and $V^T$ has orthonormal rows. SVD can be used to find a solution of a set of linear equations corresponding to a singular matrix that has no exact solution – it locates the closest possible solution a least square sense.

Sometimes it is required to find the 'closest' singular matrix to the original matrix $A$ – this decreases the rank from $n$ to $n-1$. This is done by replacing the smallest diagonal element of D by zero – this new matrix is closest to the old one with respect to the Frobenius norm (which is calculated as a sum of the squared values of all matrix elements).

Both the radial distortion and the decentering can be treated in most cases as *rotationally symmetric*. They are often modeled as polynomials. Let $u$, $v$ denote the correct image coordinates; $\tilde{u}$, $\tilde{v}$ denote the measured uncorrected image coordinates that come from the actual pixel coordinates $x$, $y$ and the estimate of the position of the principal point $\hat{u}_0$, $\hat{v}_0$.

$$\begin{aligned} \tilde{u} &= x - \hat{u}_0 \\ \tilde{v} &= y - \hat{v}_0 \end{aligned} \tag{9.18}$$

The correct image coordinates $u$, $v$ are obtained if the compensations for errors $\delta u$, $\delta v$ are added to the measured uncorrected image coordinates $\tilde{u}$, $\tilde{v}$.

$$\begin{aligned} u &= \tilde{u} + \delta u \\ v &= \tilde{v} + \delta v \end{aligned} \tag{9.19}$$

The compensations for errors are often modeled as polynomials in even powers

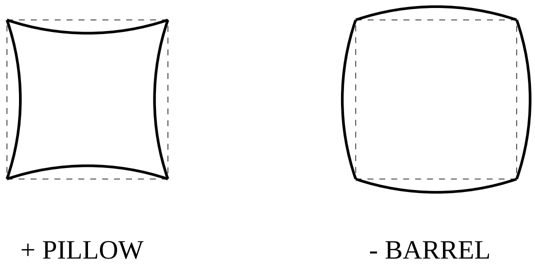


**Figure 9.7** *Radial distortion of a off shelf lens.*

to secure the rotational symmetry property. Typically elements up to maximally degree six are considered.

$$\begin{aligned} \delta u &= (\tilde{u} - u_p)(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) \\ \delta v &= (\tilde{v} - v_p)(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) \end{aligned} \tag{9.20}$$

where $u_p$, $v_p$ is the correction to the position of the principal point. The $r^2$ is the square of the radial distance from the centre of the image.

$$r^2 = (\tilde{u} - u_p)^2 + (\tilde{u} - u_p)^2 \tag{9.21}$$

Recall that $\hat{u}_0$, $\hat{v}_0$ were used in equation (9.18). The $u_p$, $v_p$ are corrections to $\hat{u}_0$, $\hat{v}_0$ that can be applied after calibration to get the proper position of the principal point.

$$\begin{aligned} u_0 &= \hat{u}_0 + u_p \\ v_0 &= \hat{v}_0 + v_p \end{aligned} \tag{9.22}$$

Let visualize typical radial distrortion of the lens for the simple second order model as a special case of equation (9.20), i.e. no decentering is assumed and second order polynomial approximation is considered

$$\begin{aligned} u &= \tilde{u}(1 \pm \kappa_1(\tilde{u}^2 + \tilde{v}^2)) \\ v &= \tilde{v}(1 \pm \kappa_1(\tilde{u}^2 + \tilde{v}^2)) \end{aligned} \tag{9.23}$$

The original image was a square pattern. The distorted images are shown in Fig. 9.7. The left part of the figure shows the pillow like distortion (minus sign in equation (9.23), whereas the right part depicts the barrel like distortion corresponding to the plus sign.

There are more complicated lens models that cover tangential distortions that model such effects as lens decentering [JKS95]. We shall not describe the procedure in detail here due to lack of space. Reader might want to consult the original paper [Tsa87] or the treatment in [JKS95]. Alternative procedure was proposed in [PM97].

### 9.2.5 Two cameras, stereopsis

To the uneducated observer, the most obvious difference between the human visual system and most of the material presented thus far in this book is that we have two eyes and therefore (a priori, at any rate) twice as much input as a single image. From Victorian times, the use of two slightly different views to provide an illusion of 3D has been common, culminating in the '3D movies' of the 1950's. Conversely, we might hope that a 3D scene, if presenting two different views to two eyes, might permit the recapture of depth information when the information therein is combined with some knowledge of the sensor geometry (eye locations).

Stereo vision has enormous importance to us – humans. It has provoked a great deal of research into vision systems with two inputs that exploit the knowledge of their own relative geometry to derive depth information from the two views they receive.

Calibration of one camera and knowledge of the co-ordinates of one image point allows us to determine a ray in space uniquely. If two calibrated cameras observe the same scene point $\mathbf{X}$, its 3D co-ordinates can be computed as the intersection of two such rays. This is the basic principle of *stereo vision* that typically consists of three steps:

- Camera calibration;
- Establishing point correspondences between pairs of points from the left and the right images;
- Reconstruction of 3D co-ordinates of the points in the scene.

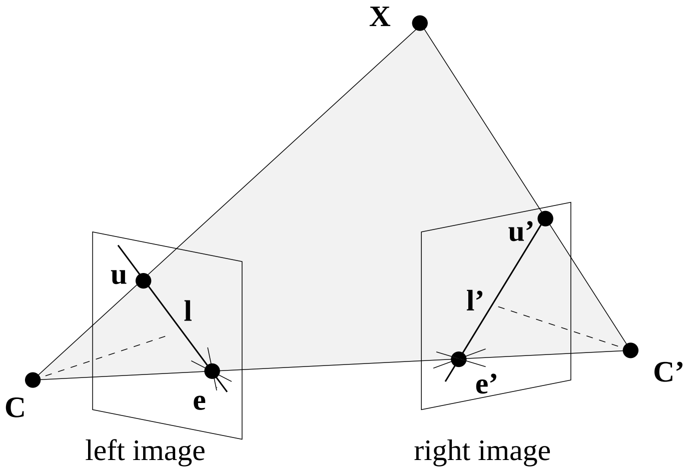


**Figure 9.8** *Epipolar geometry in stereopsis.*

The geometry of the system with two cameras is given in Figure 9.8. The line connecting optical centers $\mathbf{C}$ and $\mathbf{C}'$ is called the *baseline*. Any scene point $\mathbf{X}$ observed by the two cameras and the two corresponding rays from optical centers $\mathbf{C}$, $\mathbf{C}'$ define an *epipolar plane*. This plane intersects the image planes in the *epipolar lines* $\mathbf{l}$, $\mathbf{l}'$. When the scene point $\mathbf{X}$ moves in space, all epipolar lines pass through *epipoles* $\mathbf{e}$, $\mathbf{e}'$ – the epipoles are the intersections of the baseline with the respective image planes.

Let $\mathbf{u}$, $\mathbf{u}'$ be projections of the scene point $\mathbf{X}$ in the left and right images respectively. The ray $\mathbf{CX}$ represents all possible projections of the point $\mathbf{X}$ to the left image, and is also projected into the epipolar line $\mathbf{l}'$ in the right image. The point $\mathbf{u}'$ in the right image that corresponds to the projected point $\mathbf{u}$ in the left image must thus lie on the epipolar line $\mathbf{l}'$ in the right image. This geometry provides a strong *epipolar constraint* that reduces the dimensionality of the search space for a correspondence between $\mathbf{u}$ and $\mathbf{u}'$ in the right image from 2D to 1D.

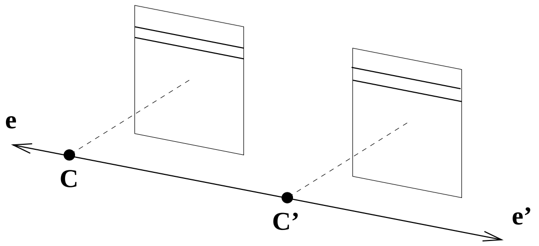


**Figure 9.9** *The canonical stereo configuration where the epipolar lines are parallel in the image, and epipoles move to infinity.*

A special arrangement of the stereo camera rig, called the *canonical configuration* is often used. The baseline is aligned to the horizontal co-ordinate axis, the optical axes of the cameras are parallel, the epipoles move to infinity, and the epipolar lines in the image planes are parallel (see Figure 9.9). For this configuration, the computation is slightly simpler;it is often used when stereo correspondence is to be determined by a human operator who will find matching points linewise to be easier (this non-automatic approach is still used in photogrammetry and remote sensing). A similar conclusion holds for computer programs too; it is easier to move along horizontal lines than along general lines. The geometric transformation that changes a general camera configuration with nonparallel epipolar lines to the canonical one is called *image rectification.* Formulae for image rectification will be given in section 9.2.9.

On the other hand, some authors [Moh93] report practical problems with the canonical stereo configuration, which adds unnecessary technical constraints to the vision hardware. If high precision of reconstruction is an issue, it is better to use general stereo geometry since rectification induces resampling that causes loss of resolution.

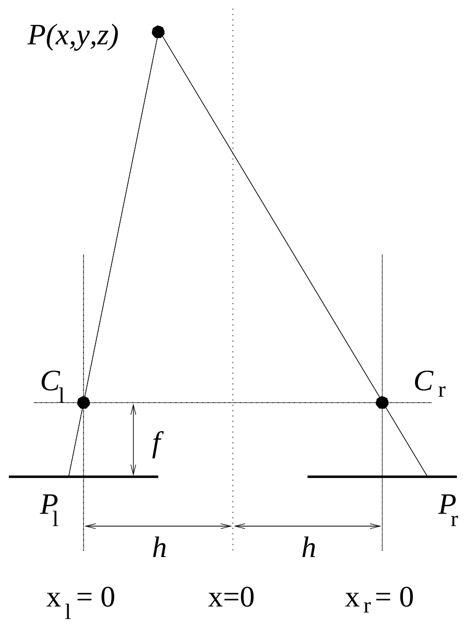


**Figure 9.10** *Elementary stereo geometry in canonical configuration.*

Considering firstly an easy canonical configuration, we shall see how to recover depth. The optical axes are parallel, which leads to the notion of disparity that is often used in stereo literature. A simple diagram demonstrates how we proceed. In Figure 9.10, which is purely schematic, we have a bird's eye view of two cameras with parallel optical axes separated by a distance $2h$. The images they provide, together with one point $P$ with co-ordinates $(x, y, z)$ in the scene, showing this point's projection onto left ($P_l$) and right ($P_r$) images. The co-ordinates in Figure 9.10 have the $z$ axis representing distance from the cameras (at which $z = 0$) and the $x$ axis representing "horizontal" distance (the $y$ co-ordinate, into the page, does not therefore appear). $x = 0$ will be the position midway between the cameras; each image will have a local co-ordinate system ($x_l$ on the left, $x_r$ on the right) which for the sake of convenience we measure from the center of the respective images; that is, a simple translation from the global $x$ co-ordinate. Without fear of confusion $P_l$ will be used simultaneously to represent the position of the projection of $P$ onto the left image, and its $x_l$ co-ordinate – its distance from the center of the left image (and similarly for $P_r$).

It is clear that there is a *disparity* between $x_l$ and $x_r$ as a result of the different camera positions (that is, $|P_l - P_r| > 0$); we can use elementary geometry to deduce the $z$ co-ordinate of $P$.

Note that $P_l$, $C_l$ and $C_l$, $P$ are the hypotenuses of similar right-angled triangles. Noting further that $h$ and $f$ are (positive) numbers, $z$ is a positive co-ordinate and $x$, $P_l$, $P_r$ are co-ordinates that may be positive or negative, we can then write:

$$\frac{P_l}{f} = -\frac{h+x}{z} \tag{9.24}$$

and similarly from the right hand side of Figure 9.10

$$\frac{P_r}{f} = \frac{h-x}{z} \tag{9.25}$$

Eliminating $x$ from these equations gives

$$z\,(P_r - P_l) = 2hf \tag{9.26}$$

and hence

$$z = \frac{2hf}{P_r - P_l} \tag{9.27}$$

Notice in this equation that $P_r - P_l$ is the detected disparity in the observations of $P$. If $P_r - P_l = 0$ then $z = \infty$. Zero disparity indicates the point is (effectively) at an infinite distance from the viewer.

### 9.2.6 The geometry of two cameras. The fundamental matrix

We proceed to derive a mathematical description for the general stereo rig with nonparallel optical axes, see Figure 9.11; the symbol $\simeq$ will be used to denote projection up to unknown scale. The co-ordinate system of the left view can be transformed to the right view by a translation $\mathbf{t}$ from the left camera center $\mathbf{C}$ to the right camera center $\mathbf{C}'$, and the co-ordinate systems can then be transformed by the rotation $R$. We shall use a co-ordinate system with the origin in the left camera center $\mathbf{C}$. If $K$, $K'$ are the calibration matrices of the left and right

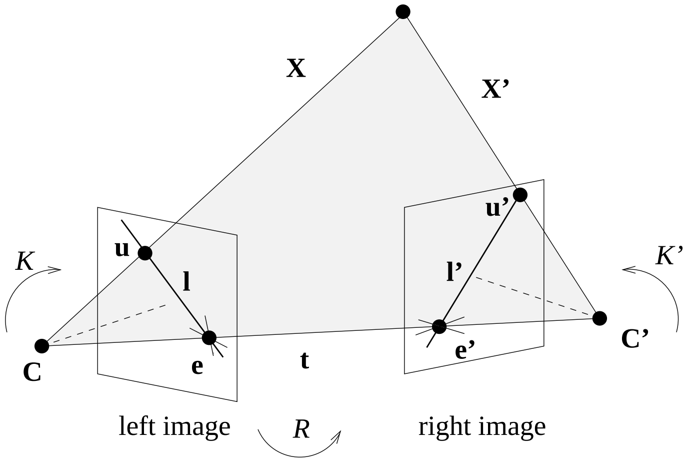


**Figure 9.11** *Stereo with nonparallel axes.*

cameras, we can apply equation (9.9) to get the left projection $\mathbf{u}$ and the right projection $\mathbf{u}'$ of the scene point $\mathbf{X}$

$$\begin{aligned} \mathbf{u} &\simeq [K|\mathbf{0}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = K\,\mathbf{X}, \\ \mathbf{u}' &\simeq [K'R\,|-K'R\,\mathbf{t}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = K'(R\,\mathbf{X} - R\,\mathbf{t}) = K'\mathbf{X}' \end{aligned} \tag{9.28}$$

We know that vectors $\mathbf{X}$, $\mathbf{X}'$ and $\mathbf{t}$ are coplanar. Distinguish co-ordinates of the left and right cameras by the subscript $_L$, $_R$, respectively – the co-ordinate vector $\mathbf{X}'$ is expressed with respect to the right camera co-ordinate system and therefore it is denoted $\mathbf{X}'_R$. We shall express the epipolar constraint using the vector product $\times$, and will do this by expressing the free vector $\mathbf{X}'_R$ with respect to the left camera. The co-ordinate rotation can be written as $\mathbf{X}'_R = R\,\mathbf{X}'_L$, and hence $\mathbf{X}'_L = R^{-1}\mathbf{X}'_R$. The equation expressing coplanarity can be written as

$$\mathbf{X}_L^T \cdot (\mathbf{t} \times \mathbf{X'}_L) = 0 \tag{9.29}$$

Substituting from equations $\mathbf{X}_L = K^{-1}\mathbf{u}$, $\mathbf{X}'_R = (K')^{-1}\mathbf{u}'$, and $\mathbf{X}'_L = R^{-1}(K')^{-1}\mathbf{u}'$ we get

$$(K^{-1}\mathbf{u})^T \cdot (\mathbf{t} \times R^{-1}\, K'^{-1}\mathbf{u}') = 0 \tag{9.30}$$

This equation (9.30) is homogeneous with respect to $\mathbf{t}$, so the scale is not determined. Absolute scale cannot be recovered if the 'yardstick', i.e. the distance between two points known in advance, is not seen in the scene.

It is helpful to replace the vector product by matrix multiplication. The translation vector is $\mathbf{t} = [t_x, t_y, t_z]^T$, and a skew symmetric[2] matrix $S(\mathbf{t})$ can be

[2] S is skew symmetric if $S^T = -S$.

created from it if $\mathbf{t} \neq \mathbf{0}$.

$$S(\mathbf{t}) = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{9.31}$$

Note that $\text{rank}(S) = 2$ if and only if $\mathbf{t} \neq \mathbf{0}$; the vector product can be replaced by the multiplication of two matrices; for any regular matrix $A$, we have

$$\mathbf{t} \times A \ = S(\mathbf{t})\, A \tag{9.32}$$

Thus we can rewrite equation (9.30) as

$$(K^{-1}\mathbf{u})^T\, (S(\mathbf{t})\, R^{-1}\, K'^{-1}\mathbf{u}') = 0$$

which may be re-arranged to

$$\mathbf{u}^T(K^{-1})^T S(\mathbf{t}) R^{-1} K'^{-1}\mathbf{u}' = 0 \tag{9.33}$$

The middle part of this equation can be concentrated into a single matrix $F$ called the *fundamental matrix* of two views.

$$F = (K^{-1})^T S(\mathbf{t}) R^{-1} K'^{-1} \tag{9.34}$$

With the substitution for $F$ in equation (9.33) we finally get the bilinear relation (sometimes called Longuet-Higgins equation after the inventor [LH81] of the similar idea) between any two views

$$\mathbf{u}^T\, F\, \mathbf{u}' = 0 \tag{9.35}$$

It can be seen that the fundamental matrix $F$ captures all information that can be recovered from a pair of images if the correspondence problem is solved. We shall consider the properties of the fundamental matrix further in due course.

### 9.2.7 Relative motion of the camera; the essential matrix.

A case of practical interest is a single camera moving in space, or two cameras with known calibration – this is known as *relative motion of the camera.* Knowledge of the camera calibration matrices $K$, $K'$ allows us to normalize measurement in left and right images; the normalized measurements are denoted $\breve{\mathbf{u}}$, $\breve{\mathbf{u}}'$. The camera calibration matrices give the relations

$$\breve{\mathbf{u}} = K^{-1}\mathbf{u}, \quad \breve{\mathbf{u}}' = K'^{-1}\mathbf{u}' \tag{9.36}$$

If these relations are used in equation (9.33), we get a simplified version

$$\breve{\mathbf{u}}^T\, S(\mathbf{t}) R^{-1}\, \breve{\mathbf{u}}' = 0 \tag{9.37}$$

Substituting $E = S(\mathbf{t})R^{-1}$, where $E$ is called the *essential matrix*, we get

$$\breve{\mathbf{u}}^T \, E \, \breve{\mathbf{u}}' = 0 \tag{9.38}$$

Again, a bilinear relation between two views in correspondence has been obtained. The essential matrix $E$ captures all the information about the relative motion from the first to the second position of the calibrated camera. $E$ can be estimated from image measurements.

Let us summarize important *properties of the essential matrix.*

- The essential matrix $\mathbf{E}$ has rank 2.
- Let $\mathbf{t}$ be the translational vector. Let $\mathbf{t}' = R\,\mathbf{t}$. Then $E\,\mathbf{t}' = 0$ and $\mathbf{t}^T E = 0$.
- The essential matrix $E$ decomposes using SVD into rank two matrix. The SVD decomposes $E$ as $E = UDV^T$ for a diagonal $D$. Then

$$D = \begin{bmatrix} k & & \\ & k & \\ & & 0 \end{bmatrix} \tag{9.39}$$

Assuming that the essential matrix $E$ has already been estimated, we might be interested in the rotation $R$ and translation $\mathbf{t}$ between these two views. We present without proof a procedure to accomplish this [Har92]. Equation (9.37) shows that the essential matrix is a product of the matrices $S(\mathbf{t})\,R^{-1}$. As

$$\breve{\mathbf{u}}^T \, S(\mathbf{t})R^{-1} \, \breve{\mathbf{u}}' = 0 \; , \quad \breve{\mathbf{u}}'^T \, RS(\mathbf{t}) \, \breve{\mathbf{u}} = 0 \tag{9.40}$$

we can see also that $E = RS(\mathbf{t})$. Recall that SVD provides a similar factorization of a matrix, $E = UDV^T$. The matrix $W = \mathrm{diag}[k, k, 0]$. The matrix $W$ has two same singular values $k$ and the third one equals to zero, see equation (9.39). Let

$$G = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} , \; Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{9.41}$$

The rotation matrix $R$ can be calculated as

$$R = UG\,V^T \text{ or } R = UG^T\,V^T \tag{9.42}$$

and the components of the translation vector can be derived from the matrix $S(\mathbf{t})$, remembering equation (9.31). $S(\mathbf{t})$ itself can be estimated as

$$S(\mathbf{t}) = VZ\,V^T \tag{9.43}$$

Let us come to the *properties of the fundamental matrix.*

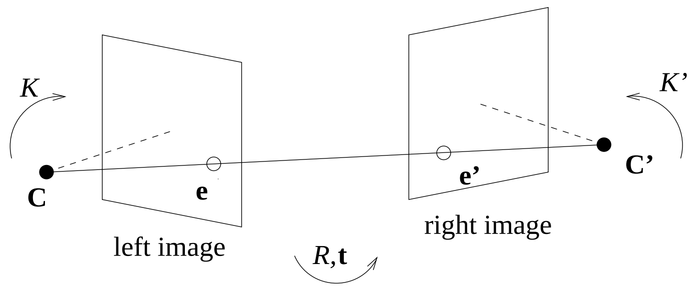


**Figure 9.12** *Epipoles* $\mathbf{e}, \mathbf{e}'$ *and the fundamental matrix* $F$*.*

- We have shown in the previous section that the rank of the essential matrix $E$ is two. As $F = (K^{-1})^T E K'^{-1}$ and the calibration matrices are regular, we see that the fundamental matrix $F$ has rank two as well.
- Consider two epipoles $\mathbf{e}, \mathbf{e}'$, depicted in Figure 9.12. It holds

$$\mathbf{e}^T F = 0 \text{ and } F\,\mathbf{e}' = 0 \tag{9.44}$$

- SVD of the fundamental matrix gives $F = UDV^T$, where

$$D = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ k_1 \neq k_2 \neq 0 \tag{9.45}$$

### 9.2.8 Estimation of a fundamental matrix from image point correspondences

The epipolar geometry has seven degrees of freedom [Moh93]. The epipoles $\mathbf{e}$, $\mathbf{e}'$ in the image have two coordinates each, i.e., the total of four degrees of freedom. Another three degrees of freedom come from the mapping of any three epipolar lines in the first image to the second image. Thus the correspondence of seven points in the left and right images enables to establish the fundamental matrix $F$ using a nonlinear algorithm [FLM92]. Unfortunately the computation is nonlinear and numerically unstable.

If there are eight non-coplanar corresponding points available, the linear method called the *eight points algorithm* can be used. If more points are at hand the estimation of fundamental matrix might be robust to noise and mismatches. The method was originally proposed by Longuet-Higgins [LH81] for the essential matrix estimation.

The eight points algorithm was supposed to be numerically unstable. That is not the case if the normalization (i.e. translation and scaling) of involved values is performed first [Har95]. Algorithm is easy to implement and is fast. The proper normalization is needed in most of the 3D geometry algorithms to obtain numerical stability.

Recall the fundamental matrix $F$,

$$\mathbf{u}_i^T \, F \, \mathbf{u}'_i = 0 \tag{9.46}$$

An image vector in homogeneous coordinates can be written using its elements $\mathbf{u}^T = [u_i, v_i, 1]$. The $3 \times 3$ fundamental matrix $F$ has only eight unknowns as it can only be set up to an unknown scale. We get eight matrix equations from eight point correspondences

$$[u_i,\, v_i,\, 1] \, F \begin{bmatrix} u'_i \\ v'_i \\ 1 \end{bmatrix} = 0 \tag{9.47}$$

Let us rewrite the elements of the fundamental matrix into a column vector with nine elements $\mathbf{f}^T = [f_{11}, f_{12}, \ldots, f_{33}]$. Then, the equation (9.47) can be rewritten as a system of linear equations

$$\begin{bmatrix} u_i u'_i & u_i v'_i & u_i & v_i u'_i & v_i v'_i & v_i & u'_i & v'_i & 1 \\ & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{33} \end{bmatrix} = 0 \tag{9.48}$$

If the left-hand matrix in the equation (9.48) is denoted by $A$, we get

$$A \, \mathbf{f} = 0 \tag{9.49}$$

The matrix $A$ has rank 8 in a perfect case without noise. With the data from real image measurements, the overdetermined system of linear equations is obtained. The least-squares solution to this set is looked for [Har95]. The vector $\mathbf{f}$ is searched that minimizes Frobenius norm $||A\mathbf{f}||$ fulfilling the constraint $||f|| = 1$. The principal component analysis gives the solution, $\mathbf{f}$ is the unit eigenvector of $A^T A$ corresponding to the smallest eigenvalue of $A$. An appropriate algorithm for doing this job is SVD.

Let us note that another numerically plausible solution to overdetermined system of linear equations (9.49) is given in [Fau93].

The estimation of the fundamental matrix can be corrupted by the gross errors caused by the *mismatches* in stereo correspondence, cf. Fig. 9.13. The obvious solution to the problem is the robust estimation that allows to drop out erroneous mismatches. The first approach uses the least median of squares method for robust estimation instead of standard least squares

$$\min_{\mathbf{f}} (\mathbf{f}^T \, A^T \, A \, \mathbf{f}) \longrightarrow \min_{\mathbf{f}} \left( \mathrm{median}(||A \;\, \mathbf{f}||^2) \right) . \tag{9.50}$$

The eight points algorithm based on SVD presented above does very similar job.

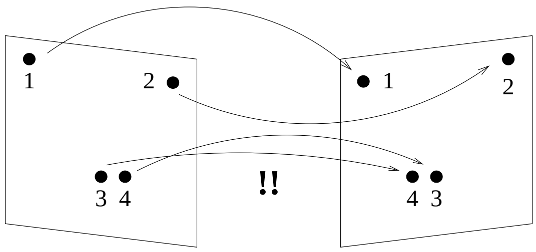


**Figure 9.13** *Problem with mismatches in stereo correspondence.*

We have already seen that the fundamental matrix $F$ should have rank 2. The solution of the equation (9.49) will not give the matrix $F$ with rank 2 in general. The matrix $F$ should be replaced by the matrix $\hat{F}$ that minimizes Frobenius norm of the $\|F - \hat{F}\|$ fulfilling condition $\mathrm{rank}(A) = 2$. The SVD decomposes $F{=}UDV^T$, $D{=}\mathrm{diag}[r,s,t]$, $r \geq s \;\geq t$, where *diag* denotes the diagonal matrix. The solution we look for is $\hat{F}{=}U\,\mathrm{diag}[r,s,0]\,V^T$.

## 9.2.9 Applications of the epipolar geometry in vision

### 9.2.9.1 Image rectification to ease the search for correspondences

We learned that the stereo geometry implies that the corresponding points can be searched for in 1D space along the epipolar lines. The epipolar line in the left image is not parallel to the epipolar line in the right image in general (non-parallel optical axes). Parallel epipolar lines are preferred as the search for correspondence performed by a human or by a computer program is simpler. It is always possible to apply the image rectification to images captured by a stereo rig with non-parallel optical axes. The result is a new set of images with parallel epipolar lines that are typically horizontal.

The image rectification recalculates pixel coordinates using a linear transformation in projective space. The rectification is illustrated in Fig. 9.14. $C$, $C'$ are optical centres. The image planes with dashed borders show input images before rectification. The image planes with solid borders and parallel horizontal epipolar lines (dotted lines in rectified images) are the desired result. Points in the left and right images are bilinearly related through the fundamental matrix $F$, $\mathbf{u}^T\,F\,\mathbf{u}' = 0$. We look for the two $3\times 3$ transformation matrices $A$, $B$ that rectify coordinates (rectified ones are denoted by $\breve{}$ ) of points in left and right images respectively, $\breve{\mathbf{u}} = A\mathbf{u}$, $\breve{\mathbf{u}}' = A\mathbf{u}'$. The fundamental matrix of the rectified images $\breve{F} = (A^{-1})^T F B^{-1}$ should correspond to epipoles that moved along horizontal axis to $-\infty$ or $\infty$ for the left and right images, respectively.

To fulfil the rectification task we need to set values in transformation matrices $A$ and $B$. Let us present briefly main ideas and results from the paper [AH88]. The transformation of the coordinates is constrained and thus number of unknown is reduced. The image coordinate transformation using matrices $A$, $B$ should

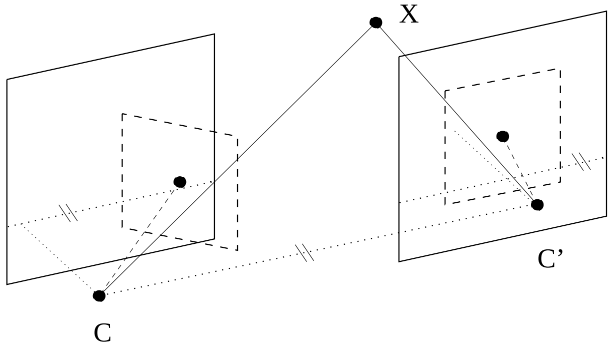


**Figure 9.14** *Image rectification to get parallel epipolar lines.*

not change the position of optical centres but should align two distinct image planes into one common image plane that is parallel to line joining $C$ and $C'$ and perpendicular to both newly recalculated optical axes. Moreover, it is desired that the corresponding epipolar lines have the same vertical coordinate as this simplifies calculations.

The image coordinates after rectification are given

$$\check{\mathbf{u}} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = A \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} , \quad \check{\mathbf{u}}' = \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = B \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \tag{9.51}$$

Recall the projection matrix from equation (9.12). We have two such matrices $M$, $M'$ for left and right images before rectification. The $3 \times 3$ submatrix on the left side of $M$ is composed of three rows ($1 \times 3$ vectors) that we denote as $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$. Similarly for the right image the matrix $M'$ gives three vectors $\mathbf{m}'_1$, $\mathbf{m}'_2$, and $\mathbf{m}'_3$. $C$ and $C'$ are coordinates of the optical centres. We are ready to write the transformation matrices that perform rectification.

$$A = \begin{bmatrix} ((C \times C') \times C)^T \\ (C \times C')^T \\ ((C - C') \times (C \times C'))^T \end{bmatrix} [\mathbf{m}_2 \times \mathbf{m}_3,\ \mathbf{m}_3 \times \mathbf{m}_1,\ \mathbf{m}_1 \times \mathbf{m}_2] \tag{9.52}$$

$$B = \begin{bmatrix} ((C \times C') \times C')^T \\ (C \times C')^T \\ ((C - C') \times (C \times C'))^T \end{bmatrix} [\mathbf{m}'_2 \times \mathbf{m}'_3,\ \mathbf{m}'_3 \times \mathbf{m}'_1,\ \mathbf{m}'_1 \times \mathbf{m}'_2] \tag{9.53}$$

The rectification is computationally inexpensive. Only two $3 \times 3$ transformation matrices must be stored. Only 6 multiplications, 6 additions and 2 divisions are needed per each rectified pixel. Notice that the rectificiation is a linear transformation in projective space that preserves straight lines. If an image consists of linear segments then it is sufficient to rectify end points of these segments. The rectification procedure can be easily generalized to three and more images [AH88].

#### 9.2.9.2 Ego-motion estimation from the calibrated camera measurements

The camera *ego-motion* estimation of the calibrated camera treats the case of the unknown movement of the camera. The rotation $R$ and the translation $\mathbf{t}$ should be learned from point correspondences between two images. Let a point $\mathbf{u}_i$ from the first image corresponds to the point $\mathbf{u}'_i$. The following algorithm [Har92] allows to compute an unknown rotation $R$ and a translation $\mathbf{t}$ of the camera.

| Algorithm 9.1: Ego-motion estimation |
|---|

1. Find correspondences between points $\mathbf{u}_i$ and $\mathbf{u}'_i$. These correspondences will be used to estimate a fundamental matrix.

2. The measured data should be normalized. This normalization helps to minimize numerical errors.

$$\breve{\mathbf{u}} = H_1\,\mathbf{u}, \qquad \breve{\mathbf{u}}' = H_2\,\mathbf{u}' \tag{9.54}$$

$$H_1 = \begin{bmatrix} a_1 & 0 & c_1 \\ 0 & b_1 & d_1 \\ 0 & 0 & 1 \end{bmatrix}, \qquad H_2 = \begin{bmatrix} a_2 & 0 & c_2 \\ 0 & b_2 & d_2 \\ 0 & 0 & 1 \end{bmatrix} \tag{9.55}$$

   The data should have similar order after normalization, i.e. $\mathrm{mean}(\breve{\mathbf{u}}) = 0$ and $\mathrm{var}(\breve{\mathbf{u}}) = [1,1]^T$.

3. Compute the fundamental matrix $\hat{F}$ (the symbol $\hat{}$ denotes an estimated value) using the linear algorithm given in section 9.2.8. Due to numerical inaccuracies the estimated fundamental matrix need not to have the property that after SVD the $D = diag(k,k,0)$.

4. Compute the estimated essential matrix $\hat{E}$. This is easy as calibration matrices $K$, $K'$ are known,

$$\hat{E} = K^T\,\hat{F}\,K' \tag{9.56}$$

5. Find an unknown rotation $R$ and translation $\mathbf{t}$ from the estimated essential matrix $\hat{E}$ using singular value decomposition (SVD). The translation $\mathbf{t}$ is given up to scale only.

$$\hat{E} = UDV^T\,, \quad D = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & t \end{bmatrix} \tag{9.57}$$

   Notice that we expect three different singular values due to numerical inaccuracies. We know that the essential matrix $E$ has to have two same singular values and the third must equal zero. We can adjust singular values by zeroing $t$ and taking average of $r$ and $s$

$$E = u \begin{bmatrix} \frac{r+s}{2} & 0 & 0 \\ 0 & \frac{r+s}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \tag{9.58}$$

The essential matrix $E$ can be decomposed into the rotation $R$ and translation $\mathbf{t}$ the same way as was used in section 9.2.7. Recall that matrices $G$ and $Z$ were defined by equations (9.41). Then we can calculate

$$R = UGV^T \text{ or } UG^T V^T\,, \qquad S(\mathbf{t}) = VZ\,V^T \tag{9.59}$$

---

Notice that the translation $\mathbf{t}$ is obtained up to unknown scale only. This is a natural result. As nothing was known in advance about the scene, the same images could be seen when a half-size objects are observed from a half distance.

#### 9.2.9.3 3D similarity reconstruction from two cameras with known intrinsic calibration

The task of the 3D similarity reconstruction aims to measure 3D coordinates of the scene point $\mathbf{X}$ from two image measurements $\mathbf{u}$ and $\mathbf{u}'$, see Figure 9.15. We assume that cameras are calibrated, i.e. they intrinsic calibration parameters are known. Thus calibration matrices $K$ and $K'$ are known. The extrinsic parameters are unknown. This case differs from the standard stereo (full 3D Euclidean reconstruction) where the relative position of the cameras is known. Common sense hints us that less will be measured in an unknown scene compared to the standard stereo case. The reconstruction of the unknown scene is given up to a similarity. Here we have two cameras with known intrinsic calibration

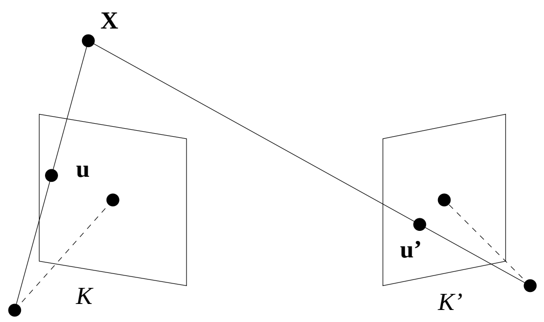


**Figure 9.15** *3D similarity reconstruction from two cameras.*

parameters. Extrinsic parameters, i.e. relative rotation and translation between cameras are unknown.

The image measurements are

$$\mathbf{u} \simeq [K\,|\mathbf{0}]\,\mathbf{X}\,,\ \mathbf{u}' \simeq [K'R\,|-K'R\mathbf{t}]\,\mathbf{X} \tag{9.60}$$

Algorithm 9.2: 3D similarity reconstruction from two cameras

1. Find correspondences between two images.
2. Compute essential matrix $E$.
3. Obtain the rotation $R$ and the translation $\mathbf{t}$ from the essential matrix $E$.
4. Solve equations (9.60) to get $\mathbf{X}$.

---

Notice that the computed $\mathbf{X}$ is found up to scale only. That means that we do not get the Euclidean reconstruction but the similarity reconstruction. We cannot get the full Euclidean reconstruction as in stereo vision. The reason is that the distance between two cameras is unknown in this case.

#### 9.2.9.4 3D projective reconstruction from two uncalibrated cameras

Let us study the most general 3D reconstruction case when the point correspondence in two uncalibrated cameras can be established. That means that both intrinsic and extrinsic camera calibration parameters are unknown. We shall see that what can be obtained is the 3D projective reconstruction. This is a practically appealing case as we can learn something about the geometry of the scene even from a video sequence where nothing is known about the conditions under which it was captured. The position of the camera is unknown. The camera zoom lense may be used and we do not know what was the actual focal length.

The perspective projection performed by the first camera is expressed using the projective matrix $M$, recall equation (9.9). The matrix $M$ is divided into the three row vectors $\mathbf{m}_1^T$, $\mathbf{m}_2^T$, $\mathbf{m}_3^T$. Similarly for the second camera where primed symbols are used:

$$\text{1st image} \qquad \mathbf{u} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \simeq M\,\mathbf{X} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} \mathbf{X} \tag{9.61}$$

$$\text{2nd image} \qquad \mathbf{u}' = \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} \simeq M'\,\mathbf{X} = \begin{bmatrix} \mathbf{m}'^T_1 \\ \mathbf{m}'^T_2 \\ \mathbf{m}'^T_3 \end{bmatrix} \mathbf{X} \tag{9.62}$$

Let us express the ratio among three rows in the projection matrix $M$ to eliminate unknown scale factor [FM95].

$$\begin{aligned} u : v : w &= \mathbf{m}_1^T\mathbf{X} : \mathbf{m}_2^T\mathbf{X} : \mathbf{m}_3^T\mathbf{X} \\ u' : v' : w' &= \mathbf{m'}_1^T\mathbf{X} : \mathbf{m'}_2^T\mathbf{X} : \mathbf{m'}_3^T\mathbf{X} \end{aligned} \tag{9.63}$$

The same mutual ratio can be written as three equations holds for both the first and the second camera.

$$\begin{aligned} u\mathbf{m}_2^T\mathbf{X} &= v\mathbf{m}_1^T\mathbf{X} & \qquad u'\mathbf{m'}_2^T\mathbf{X} &= v'\mathbf{m'}_1^T\mathbf{X} \\ u\mathbf{m}_3^T\mathbf{X} &= w\mathbf{m}_1^T\mathbf{X} & u'\mathbf{m'}_3^T\mathbf{X} &= w'\mathbf{m'}_1^T\mathbf{X} \\ v\mathbf{m}_3^T\mathbf{X} &= w\mathbf{m}_2^T\mathbf{X} & v'\mathbf{m'}_3^T\mathbf{X} &= w'\mathbf{m'}_2^T\mathbf{X} \end{aligned} \tag{9.64}$$

The equations (9.64) can be written in a matrix form. We show it for the first camera only. The similar expression holds for the second camera.

$$\begin{bmatrix} u\mathbf{m}_2^T & - & v\mathbf{m}_1^T \\ u\mathbf{m}_3^T & - & w\mathbf{m}_1^T \\ v\mathbf{m}_3^T & - & w\mathbf{m}_2^T \end{bmatrix} \mathbf{X} = 0 \tag{9.65}$$

If the first row in the matrix obtained in the previous equation (9.65) is multiplied by $w$ and second row by $-v$ and added we get

$$\left(uw\mathbf{m}_2^T - vw\mathbf{m}_1^T - uv\mathbf{m}_3^T + vw\mathbf{m}_1^T\right)\mathbf{X} = \left(uw\mathbf{m}_2^T - uv\mathbf{m}_3^T\right)\mathbf{X} = 0 \tag{9.66}$$

If we extract the equation corresponding to the third row of the matrix in equation (9.65) we get

$$(-w\mathbf{m}_2^T + v\mathbf{m}_3^T)\mathbf{X} = 0 \tag{9.67}$$

We can see that the equations (9.66) and (9.67) are linearly dependent. The same reasoning holds for measurements from the second image. As only two equations are linearly independent let us use the second and the third equation.

$$\begin{aligned} (u\mathbf{m}_3^T - w\mathbf{m}_1^T)\mathbf{X} &= 0 & \qquad (u'\mathbf{m}_3'^T - w'\mathbf{m'}_1^T)\mathbf{X} &= 0 \\ (v\mathbf{m}_3^T - w\mathbf{m}_2^T)\mathbf{X} &= 0 & (v'\mathbf{m'}_3^T - w'\mathbf{m'}_2^T)\mathbf{X} &= 0 \end{aligned} \tag{9.68}$$

The same can be rewritten into a matrix form.

$$\begin{bmatrix} u\mathbf{m}_3^T - w\mathbf{m}_1^T \\ v\mathbf{m}_3^T - w\mathbf{m}_2^T \\ u'\mathbf{m}_3'^T - w'\mathbf{m'}_1^T \\ v'\mathbf{m'}_3^T - w'\mathbf{m'}_2^T \end{bmatrix} \mathbf{X} = A\ \mathbf{X} = 0 \tag{9.69}$$

The matrix $A$ has size $4 \times 4$ and $\mathbf{X}$ is a $4 \times 1$ vector. Recall that $[u, v, w]^T$ and $[u', v', w']^T$ are corresponding points in the first and in the second image in homogeneous coordinates. $\mathbf{m}_i^T$ is i-th row in the projection matrix $M$ of the first image, $i = 1, 2, 3$. The same with the prime symbol $'$ holds for the second image.

We are interested in a nontrivial solution of the equation (9.69). Therefore we look for the case when $det(A) = 0$. That implies that the matrix $A$ should have rank 3 if $\mathbf{u}$ and $\mathbf{u}'$ are really corresponding points in the first and in the second image. There are two important cases how to reconstruct a 3D point from two corresponding 2D points in two images:

1. *Scene reconstruction with calibrated cameras.*
   This is a special case (called stereopsis, 3D Euclidean reconstruction) that has been already treated in section 9.2.5. The currently used formalism concentrates all knowns into the matrix $A$. The projective matrices $M$, $M'$ and image measurements $\mathbf{u}$, $\mathbf{u}'$ are known. The equation (9.69) can be easily replaced by inverse mapping.

2. *Scene reconstruction with uncalibrated cameras.*
   If the calibration of a stereo rig is unknown it can be shown that the reconstructed coordinates $\tilde{\mathbf{X}}$ differ from the correct Euclidean reconstruction by some (unknown) projective transformation $H$.

$$\tilde{\mathbf{X}} = H\,\mathbf{X} \tag{9.70}$$

   The matrix $H$ is a regular $4 \times 4$ matrix. The transformation $H$ ranges from Euclidean over affine to the general projective case according to how much knowledge about the calibration is at hand. The matrix $H$ is the same for all scene points for one position and calibration of the camera. Of course, the same algorithm with different scene gives different $H$.

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \simeq M\,\mathbf{X} = M\,H^{-1}\,H\,\mathbf{X} = \tilde{M}\,\tilde{\mathbf{X}} \tag{9.71}$$

$$\mathbf{u}' = \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} \simeq M'\,\mathbf{X} = M'\,H^{-1}\,H\,\mathbf{X} = \tilde{M}'\,\tilde{\mathbf{X}} \tag{9.72}$$

   Notice that $M\,\mathbf{X}$ and $\tilde{M}\,\tilde{\mathbf{X}}$ give the same measurement. The measurements $\tilde{\mathbf{X}}$ differs from the correct Euclidean measurement $\mathbf{X}$ by a projective transformation $H$.

   The projective transformation is determined by at least 5 corresponding points. The projective matrix $\tilde{M}$ should be created in such a way that it differs from the matrix $M$ only projectively. See [Fau93, FM95] for more details.

### 9.2.10 Three and more cameras

In this section, we will study the case when three and more cameras look at the same scene. It is further assumed that the mutually corresponding points can be found in all views. We have already seen that views of two cameras are described using bilinear relation expressed by the fundamental matrix. The question to be answered here is what more can be learned if three or more views are available.

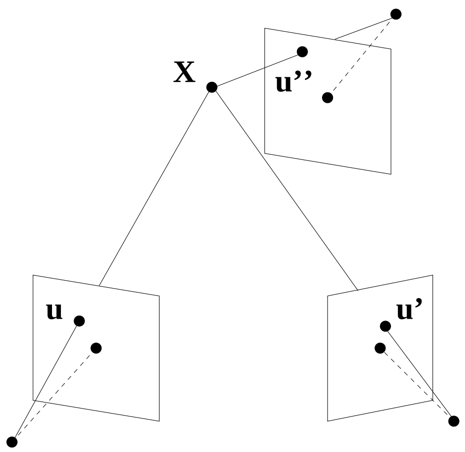


**Figure 9.16** *Geometry of three cameras,* $\mathbf{u} \simeq M\,\mathbf{x}$, $\mathbf{u}' \simeq M'\,\mathbf{x}$, $\mathbf{u}'' \simeq M''\,\mathbf{x}$.

Three cameras looking at the same point are sketched in Fig. 9.16. The relation between projected image points $\mathbf{u}$, $\mathbf{u}'$, $\mathbf{u}''$ and their respective 3D counterparts $\mathbf{X}$, $\mathbf{X}'$, $\mathbf{X}''$ are given by the projection matrices $M$, $M'$, $M''$. Let us use similar sequence of thoughts as we did in section 9.2.9.4. Our aim is to obtain set of linear equations that relate image measurements with their 3D counterparts.

$$\mathbf{u} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \simeq M\,\mathbf{X} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} \mathbf{X}\,, \quad \mathbf{u}' = \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} \simeq M'\,\mathbf{X} = \begin{bmatrix} \mathbf{m}'^T_1 \\ \mathbf{m}'^T_2 \\ \mathbf{m}'^T_3 \end{bmatrix} \mathbf{X} \quad (9.73)$$

$$\mathbf{u}'' = \begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} \simeq M''\,\mathbf{X} = \begin{bmatrix} \mathbf{m}''^T_1 \\ \mathbf{m}''^T_2 \\ \mathbf{m}''^T_3 \end{bmatrix} \mathbf{X} \quad (9.74)$$

We have to do similar manipulations as in equations (9.62) – (9.69). This eliminates unknown scale factors and provides the desired relation in a matrix form. In the next equation, the labels from `1:` to `6:` only number the equations as they are going to be referenced shortly.

$$\begin{array}{l} \mathtt{1:} \\ \mathtt{2:} \\ \mathtt{3:} \\ \mathtt{4:} \\ \mathtt{5:} \\ \mathtt{6:} \end{array} \begin{bmatrix} u\mathbf{m}_3^T - w\mathbf{m}_1^T \\ v\mathbf{m}_3^T - w\mathbf{m}_2^T \\ u'\mathbf{m}_3'^T - w'\mathbf{m}_1'^T \\ v'\mathbf{m}_3'^T - w'\mathbf{m}_2'^T \\ u''\mathbf{m}''^T_3 - w''\mathbf{m}''^T_1 \\ v''\mathbf{m}''^T_3 - w''\mathbf{m}''^T_2 \end{bmatrix} \mathbf{X} = A\,\mathbf{X} = 0 \tag{9.75}$$

We are interested in the nontrivial solution to the matrix equation (9.75). That implies that the matrix $A$ should have rank 3. This means that determinant of all $4 \times 4$ submatrices must equal to zero. The number of $4 \times 4$ subdeterminants is $\mathrm{C}_4^6 = \frac{6!}{4!\,2!} = 15$.

Let write write down all 15 quadruples of equations and classify them according to the number of cameras involved into two groups, i.e. two or three cameras. The equations will be point at by labels used in equation (9.75). Let us simplify a bit an explanation given in [FM95].

The three sets of equations express bilinear relation between two cameras that are given by the fundamental matrix $F$ as we already know. These are equations [1234], [1256] and [3456]. Notice that even the squares of the same variable do not appear in this bilinear equations. Let us list sets of equations

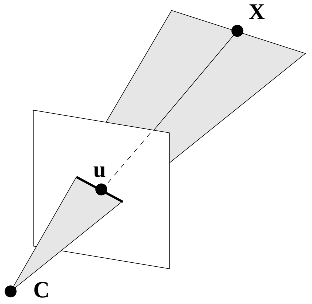


**Figure 9.17** *Each of six rows defines a plane passing through the optical center* **C** *and the point* **X**.

that express trilinear relation among images of the same point as seen by three cameras. From all 12 trilinearities only four are linearly independent. Three possibilities of quadruples of linearly independent equations are the following. Notice that there are always two rows from equation (9.75) corresponding to one of the cameras and one row per each of the remaining two cameras.

$$\begin{array}{lllll} a) & [1235] & [1245] & [1236] & [1246] \\ b) & [1345] & [2345] & [1346] & [2346] \\ c) & [1356] & [1456] & [2356] & [2456] \end{array} \tag{9.76}$$

The geometric interpretation of rows of the equation (9.75) might help to better understanding. Each row defines a plane passing through optical center **C** and the point **X** for which correspondence in all three views was established, see Fig. 9.17.

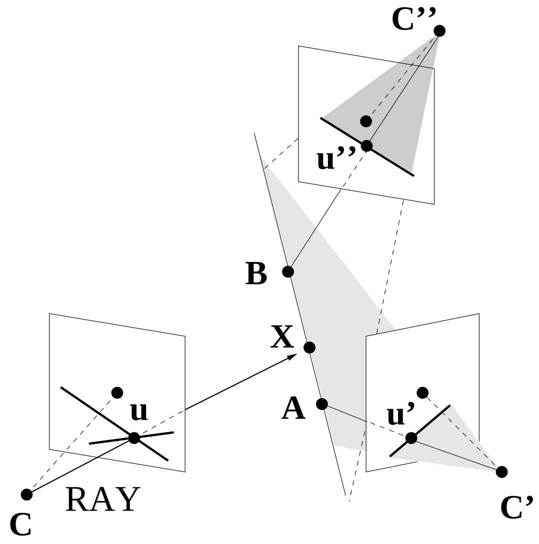


**Figure 9.18** *Illustration that one trilinear relation does not assure that three measured points are coincident as expected.*

Notice that one trilinearity relation does not assure that observed points $\mathbf{u}$, $\mathbf{u}'$, and $\mathbf{u}''$ correspond to only one scene point $\mathbf{X}$. Only one of the views plays role of the ray. This is illustrated in Fig. 9.18. Two rows in equation (9.75) corresponded to the measurement $\mathbf{u}$ taken by the first camera. The agreeable ray points to $\mathbf{X}$. The other two views constraint the point position in the space to a plane only. One trilinearity relation assures that the ray and two planes have a common point in the projective space $\mathcal{P}^3$. In other words, the $\mathbf{X}$, $\mathbf{A}$, and $\mathbf{B}$ are colinear but must not be coincident.

If $\mathbf{X}$, $\mathbf{A}$, and $\mathbf{B}$ should be colinear at least three trilinearities should be chosen that all $\mathbf{u}$, $\mathbf{u}'$, and $\mathbf{u}''$ play a role of a ray.

Let us consider what would happen if we had four cameras. In the equation (9.75) we would have two more equations. Now we can consider $4 \times 4$ subdeterminanants which contain one row arising from one camera. This is called a *quadrilinear constraint* that is a polynomial of total degree four in the coordinates of the points $\mathbf{m}_i$ and linear in the coordinates of each of them. Assuming that all the bilinear and trilinear constraints are satisfied, it is possible to show that the quadrilinear constraint can be obtained as linear combination of bilinear and trilinear constraints. This means that the *fourth view does not bring any additional information* if exact measurements in the image are assumed. To sum up, the relations among corresponding projections of a single point in two, three and four images are completely understood under orthographic, similarity, and perspective projection. There is no relation involving five and more cameras that cannot be factor into relations of fewer cameras [WWS95].

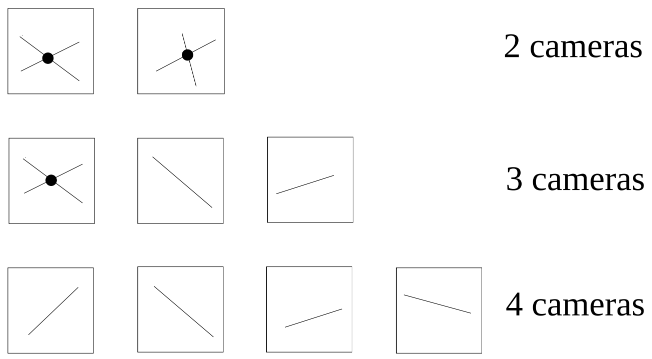


**Figure 9.19** *Geometric interpretation of bilinear, trilinear, and quadrilinear constraint.*

The case of two, three, and four cameras is illustrated in Fig. 9.19. There three rows with images are shown. The upper row shows the case of bilinear constraint (given by the fundamental matrix $F$) that relates corresponding points in two images. The middle row illustrates trilinearity constraint where correspondence among one point and two lines is established. The bottom row shows the quadrilinear constraint where correspondence of four lines is taken into account.

The reader is likely to have an intuitive geometric understanding of the trilinear relation among views now. Let us sketch the algebraic derivation as well. Let us assume that the first camera is in a canonical configuration, i.e. its projective matrix is in the simplest form.

$$\begin{aligned}
\mathbf{u} &\simeq M\,\tilde{\mathbf{X}} = M\,H^{-1}\,H\,\mathbf{X} = [I|0]\,\mathbf{X} \\
\mathbf{u}' &\simeq M'\,\tilde{\mathbf{X}} = M'\,H^{-1}\,H\,\mathbf{X} = [a_{ij}]\,\mathbf{X} \qquad i = 1\dots 3 \\
\mathbf{u}'' &\simeq M''\,\tilde{\mathbf{X}} = M''\,H^{-1}\,H\,\mathbf{X} = [b_{ij}]\,\mathbf{X} \qquad j = 1\dots 4
\end{aligned} \tag{9.77}$$

The scale in the image measurement is unknown, as $\mathbf{u} \simeq [I|0]\,\mathbf{X}$.

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{u} \\ \rho \end{bmatrix} \tag{9.78}$$

The scale factor $\rho$ is to be determined. Let us project the scene point $\mathbf{X}$ into the second camera.

$$\mathbf{u}' \simeq [a_{ij}]\,\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{u} \\ \rho \end{bmatrix} \tag{9.79}$$

$$u_i' \simeq a_i^k u_k + a_{i4}\rho\,, \qquad k = 1\dots 3 \tag{9.80}$$

The Einstein's summation convention was used in the previous equation (9.80) that yields more compact notatation. The summation symbol is not written. The $a_i^k u_k$ would originally be written as $\sum_{k=1}^{k=3} a_{ik} u_k$.

The scale factor $\rho$ should be eliminated. We get three equations and only two of them are independent:

$$u'_i \, (a_j^k \, u_k + a_{j4}\rho) = u'_j \, (a_i^k \, u_k + a_{i4}\rho) \tag{9.81}$$

This yields three estimates of $\rho$

$$\rho = \frac{u_k(u'_i \, a_j^k - u'_j \, a_i^k)}{u'_j \, a_{i4} - u'_i \, a_{j4}} \tag{9.82}$$

The scale factor $\rho$ is substituted back to the $\tilde{\mathbf{X}}$

$$\tilde{\mathbf{X}} = \left[ \begin{array}{c} \mathbf{u} \\ \frac{u_k(u'_i \, a_j^k - u'_j \, a_i^k)}{u'_j \, a_{i4} - u'_i \, a_{j4}} \end{array} \right] \simeq \left[ \begin{array}{c} (u'_j \, a_{i4} - u'_i \, a_{j4}) \, \mathbf{u} \\ u_k(u'_i \, a_j^k - u'_j \, a_i^k) \end{array} \right] \tag{9.83}$$

Let us project $\tilde{\mathbf{X}}$ by the third camera

$$\mathbf{u}'' \simeq [b_{lk}]\mathbf{X} = b_l^k x_k \tag{9.84}$$

Notice that the

$$\begin{aligned} u''_l &\simeq b_l^k u_k(u'_j a_{i4} - u'_i a_{j4}) + b_{l4} u_k (u'_i a_j^k - u'_j a_i^k) \simeq \\ &\simeq u_k u_i (a_j^k b_{l4} - a_{j4} b_l^k) - u_k u'_j (a_i^k b_{l4} - a_{i4} b_l^k) \simeq \\ &\simeq u_k (u'_i T_{kjl} - u'_j T_{kil}) \end{aligned} \tag{9.85}$$

The $T_{ijk}$, $i, j, k = 1, 2, 3$ is an algebraic entity called tensor that depends on three indices. If the reader is not familiar with tensors the entity $T_{ijk}$ can be imagined as "three dimensional matrix", i.e. a $3 \times 3 \times 3$ cube consisting of 27 numbers.

The unknown scale can be eliminated if all three views are combined together

$$u_k (u'_i u''_m T_{kjl} - u'_j u''_m T_{kil}) = u_k (u'_i u''_l T_{kjm} - u'_j u''_l T_{kim}) \tag{9.86}$$

The above equation is symmetric with respect to $i$, $j$ and $l$, $m$. Thus $i < j$ and $l < m$. There are 9 equations but only four of them are linearly independent. Let assume $j = m = 3$ and for simplicity $u_3 = u'_3 = u''_3 = 1$. After some manipulations we get the *trilinear constraint among three views.*

$$u_k (u'_i u''_l T_{k33} - u''_l T_{ki3} - u'_i T_{k3l} + T_{kil}) = 0 \tag{9.87}$$

As indices $i, l$ have domain $1, 2$ we have four linearly independent equations.

The tensor $T_{ijk}$ has 27 unknowns that can be estimated at least from 7 corresponding points in three images.

The use of trilinear constraint yields three practical advantages [SW95].

1. The trilinear tensor can be recovered linearly from 7 corresponding points in three views. The fundamental matrix calculated from pair of views needs at least 8 points for linear solution. Practically, the overdetermined system of equation is solved using some robust estimation method.

2. The tensor can be used instead of three fundamental matrices. This is possible even in the case in which some of the fundamental matrices are singular.

3. The estimate of the constraint among three views should be numerically more stable than the estimate through three fundamental matrices.

One of the important applications of trilinear tensor is the *epipolar transfer*. Assume that the trilinear tensor was already estimated. If two images are known any third image can be computed using equation (9.87).

The other application of trilinear tensor is in reconstruction and recognition. So far we have studied how one point is seen in one, two, three or four images. The dual problem, i.e. the geometry of $N$ 3D points in one image allows to approach shape under perspective projection with uncalibrated cameras [WWS95].

### 9.2.11 Stereo correspondence algorithms

The solution of the correspondence problem is the key step into any photogrammetric, stereo vision or motion analysis task. We have seen in section 9.2.6 that a lot can be learned about the geometry of the 3D screen if it is known which point from one image corresponds to a point in the second image. This subsection describes how the same point can be found in two images if the same scene is observed from two different viewpoints. Of course, it is supposed that two images overlap and thus the corresponding points are looked for in this overlapped area.

In image analysis, some methods are based on the assumption that images constitute a linear (vector) space (e.g. eigenimages or linear interpolation in images [WHH95, UB91]. The linearity assumption[3] is not valid for images in general [BP96]. Some authors did not paid attention to this fact. A structure of a vector space assumes that the $i$-th component of one vector must refer to the $i$-th component of the other vector. This assumes that the correspondence problem was solved.

An attempt to solve automatically the correspondence problem seems to be an evergreen computer vision topic. The pessimistic conclusion is that it is not solvable in general case at all. The trouble is that the correspondence problem is inherently ambiguous. Imagine an extreme case, e.g. a scene containing a white nontextured flat object. Its image constitutes a large region with uniform brightness. When corresponding points are looked for in left and right images of the flat

[3]Unformally, sum of any two points from a linear space must belong to the linear space; similarly for any point multiplied by any real number.

object there are not any features that could distinguish them. Another unavoid-

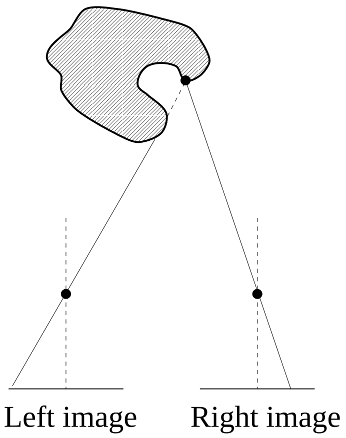


**Figure 9.20** *Selfocclussion makes search for some corresponding points impossible.*

able difficulty in searching for corresponding points is the *selfocclussion* problem. It occurs in images of nonconvex objects. Some points that are visible by the left camera are not visible by the right camera and vice versa, see Fig. 9.20.

Fortunately, not all images originating from scenes of practical interest are uniform in intensity nor majority of scene points is selfoccluded. The task to establish correspondence between projections of a same point in different views is based on finding image characteristics that are similar in both views. The way how this local similarity is calculated distinguishes various correspondence methods.

The inherent ambiguity of the *correspondence problem* can be in practical cases fortunately reduced using several *constraints*. Some of them follow from the geometry of the image capturing process, some from photometric properties of a scene, and some from prevailing object properties in our natural world. There has been a vast number of different stereo correspondence algorithms proposed so far. We will give here only a concise taxonomy of approaches to finding correspondence. Not all the constraints are used in all of them. We list constraints here briefly [KKS96] to ease the reader insight into the correspondence problem.

Let us start with the first group of constraints that depends mainly on the geometry and the photometry of the image capturing process. These are:

**Epipolar constraint** says that the corresponding point can lie on the epipolar line in the second image only. This reduces the potential 2D search space into the 1D one. The epipolar constraint was explained in detail in section 9.2.5.

**Uniqueness constraint** states that a pixel from the first image can correspond in most cases to only one pixel in the second image. The exception is the case when two or more points lie on one ray coming from the first camera and can be seen as a separate points in the second camera. This case is illustrated in Fig. 9.21. Actually, this exception the same case as selfocclussion.

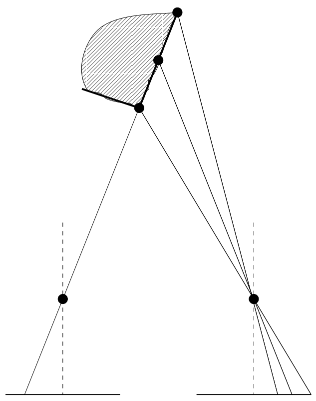


**Figure 9.21** *Exception from the uniqueness constraint.*

**Photometric compatibility constraint** states that intensities of the point in the first and the second likely image differ only a little. Intensities are not supposed to be exactly the same as the mutual angle between the light source, surface normal and the viewer position is not the same. As the views do not differ much the induced intensity change is typically small. Practically, this constraint is very sensible to image capturing conditions. The advantage is that intensities in the left image can be transformed into intensities in the right image using very simple transformation.

**Geometric similarity constraints** builds on the observation that the geometric characteristics of the features found in the first and in the second image do not differ much (e.g. length or orientation of the line segment, region or contour).

The second group of constraints exploits some common properties of objects in typical scenes. The word common means that the scene properties are likely the same.

**Disparity smoothness constraint** claims that the disparity changes slowly almost everywhere in the image. Assume two scene points $\mathbf{p}$ and $\mathbf{q}$ close to each other. Let us denote the projection of the point $\mathbf{p}$ into the left image as $\mathbf{p}_L$ and into the right image as $\mathbf{p}_R$. The projection of the point $\mathbf{q}$ is denoted accordingly. Let us assume that correspondence between $\mathbf{p}_L$ and $\mathbf{p}_R$ was successfully established. Then the absolute difference between disparities for points $\mathbf{p}$ and $\mathbf{q}$ must be small.

**Figural disparity constraint** says that the corresponding points should lie on the edge element in both right and left images and fulfil the disparity smoothness constraint as well.

**Feature compatibility constraint** bases restriction of possible matches on the physical origin of matched points. Points can match only if they have the same physical origin (i.e. object surface discontinuity, border of a shadow cast by some objects, occluding boundary, specularity boundary, etc.). Notice that the edges in the image caused by the specularity or selfocclussion cannot be used to solve correspondence problem as they move with changing viewpoint. The good news is that if the selfocclussion case caused by the abrupt discontinuity of the surface can be identified, cf. Fig. 9.22.

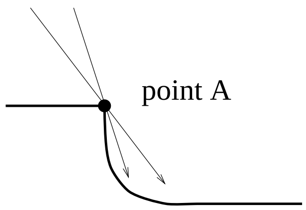


**Figure 9.22** *Selfocclusion due to abrupt surface discontinuity can be detected.*

**Disparity limit constraint** originates from psychophysical experiments. A human vision system can fuse stereo images only if the disparity is smaller than some limit. This fact constraints the lengths of the search in artificial methods that look for correspondence.

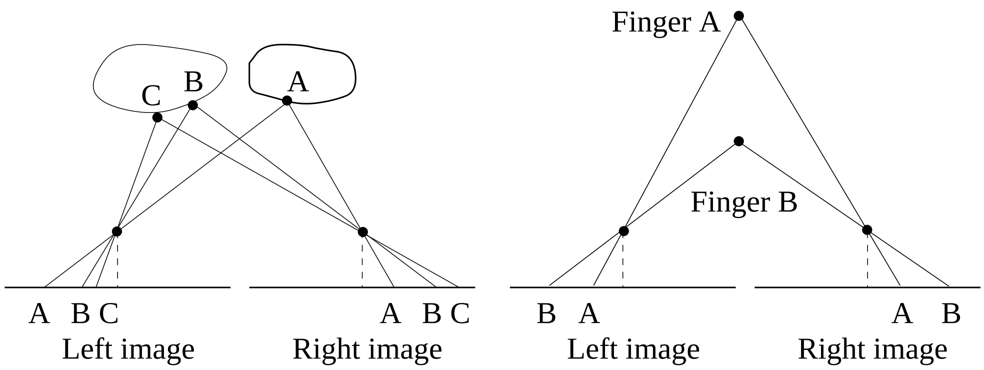


**Figure 9.23** *Corresponding point lie in the same order on epipolar lines (a). This rule does not hold if there is a big discontinuity in depths as in the image (b).*

**Ordering constraint** says that for surfaces in similar depth corresponding feature points typically lie in the same order on the epipolar line, see Figure 9.23(a). If there is a narrow object much closer to the camera compared to the rest of the scene the order can be changed, cf. Figure 9.23(b). It is easy to demonstrate the violation of the ordering constraint with your fingers. Put both pointing fingers vertically, almost aligned but at different depth in front of your eyes. If you close left, respectively right eye the order of fingers will change.

**Mutual correspondence constraint** helps to rule out those points that do not have corresponding counterpart due to occlusion, highlight or noise. Assume the search started from the left image point $\mathbf{p}_L$ and the corresponding right image point $\mathbf{p}_R$ was found. If the task is reversed, the search starts from the point $\mathbf{p}_R$ and the point $\mathbf{p}_L$ has not been found the match is not reliable and should be ruled out.

Constraints listed above are supposed to be used in actual stereo correspondence algorithms. Many such algorithms were already proposed and new improved ones still appear. Our aim is to make a taxonomy of stereo correspondence algorithms. From the historical point of view, the correspondence algorithms for stereopsis were and still are driven by two main paradigms.

1. Low-level, correlation-based, bottom-up methods.
2. High-level, feature-based, top-down methods.

At the beginning, it was believed that higher-level features as corners, straight line segments should be identified (automatically) first and then matched. This was a natural development from the photogrammetry that has been using feature points identified by human operator since the beginning of the 20th century.

Psychological experiments with *random dot stereograms* performed by Julesz [Jul90] brought entirely new view. Experiments have shown that human does not need to create monocular features before binocular depth perception can take place. The random dot stereogram is created in the following way. The left image is entirely random. The right image is created from the left image in a consistent way that some part of it is shifted according to disparity of the desired stereo effect. The viewer must glare at the random dot stereogram from a given distance of about 20 centimeters. Reader is likely familiar with random dot stereograms as they have been published under the name “3D images” in magazines.

Recently, it seems that the vision community tends to trust to combination of both low-level and the high-level stereo correspondence methods [TK90].

### Correlation-based stereo correspondence

The correlation-based correspondence algorithms use the assumption that pixels in correspondence have very similar intensities (recall the photometric compatibility constraint). The intensity of an individual pixel does not give sufficient information as there are typically more potential candidates with the similar intensity. Thus intensities of several neighbouring pixels are taken into account. The square (say 5×5 or 7×7) or rectangular window (say 3×9) is used for this purpose. These methods are sometimes called *area-based stereo*.

Let us explain a simple algorithm called *block-matching* [KKS96] to demonstrate correlation based approach to stereo. Let us assume the canonical stereo setup with parallel optical axes of both cameras. The basic idea of the block-matching algorithm is that all pixels in the window (called block) have the same disparity. That means that only and only one disparity is computed for each block. One of the images, e.g. left one, is tiled into blocks. The search for correspondence in the right image is done only for each of these blocks in the left image. The measure of the similarity between blocks can be, e.g., the mean square error of the intensity. The disparity is set for the position where the mean square error is minimal. The maximal change of the position is limited by the disparity limit constraint. The mean square error can have more than one minimum. In this case, an additional constraint should be used to cope with ambiguity.

The result of the block matching algorithm is a sparse matrix of disparities. The disparity was calculated only for the representative point of the block. There are methods that allow to refine the result that the dense disparity matrix is obtained. The block-matching algorithms are typically slow. The implementation in the regular pyramid is often used to speed up the process.

Another approach worth to mention is that of Nishihara [Nis84], who observes that the algorithm attempting to correlate individual pixels (by, e.g. matching zero crossings) [MP79] is inclined toward poor performance when noise causes the detected location of such features to be unreliable. A secondary observation is that such pointwise correlators are very heavy on processing time in arriving at a correspondence. Nishihara notes that the *sign* (and magnitude) of an edge detector response is likely to be a much more stable property to match than the edge or feature locations, and devises an algorithm that simultaneously exploits a scale-space matching attack.

The approach is to match large patches at a large scale, and then refine the quality of the match by reducing the scale, using the coarser information to initialize the finer grained match. An edge response is generated at each pixel of both images at a large scale (see Section **??**), and then a large area of the left (represented by, say, its central pixel) is correlated with a large area of the right. This can be done quickly and efficiently by using the fact that the correlation function peaks very sharply at the correct position of a match, and so a small number of tests permits an ascent to a maximum of a correlation measure. This coarse area match may then be refined to any desired resolution in an iterative manner, using the knowledge from the coarser scale as a clue to the correct disparity at a given position. At any stage of the algorithm, therefore, the surfaces in view are modelled as square prisms of varying height; the area of the squares may be reduced by performing the algorithm at a finer scale – for tasks such as obstacle avoidance it is possible that only coarse scale information is necessary, and there will be a consequent gain in efficiency.

This algorithm is enhanced by casting random dot light patterns on the scene to provide patterns to match even in areas of the scene that are texturally uniform. The resulting system has been demonstrated in use in robot guidance and bin-picking applications, and has been implemented robustly in real time.

### Feature-based stereo correspondence

Feature-based correspondence methods use points or set of points that are striking and easy to find. These are pixels on edges, lines, corners, etc. The correspondence is looked for according to properties of such features as, e.g., orientation along edges, or lengths of line segments. The advantages of feature-based methods compared to intensity-based correlation are:

- Feature-based methods are less ambiguous. The number of potential candidates for correspondence is smaller.

- The resulting correspondence is less dependent to photometric variations in images.

- Disparities can be computed with higher precision. The features can be looked for in the image with subpixel precision.

We have chosen one example of the feature-based correspondence method – the *PMF algorithm*, named after its inventors [PMF85]. It proceeds by assuming that a set of feature points (for example, detected edges) has been extracted from each image by some interest operator. The output is a correspondence between pairs of such points. In order to do this, three constraints are applied: the epipolar constraint, the uniqueness constraint and the disparity gradient limit constraint.

The preceding two constraints are not peculiar to this algorithm (indeed they are also used by Marr [MP79]) – the third, however, of stipulating a disparity gradient limit, is its novelty. The *disparity gradient* measures the relative disparity of two pairs of matching points. Suppose (Figure **??**) that a point $A$ ($B$) in 3D appears as $A_l = (a_{xl}, a_y)$ ($B_l = (b_{xl}, b_y)$) in the left image and $A_r = (a_{xr}, a_y)$ ($B_r = (b_{xr}, b_y)$) in the right (the epipolar constraint requires the $y$ co-ordinates to be equal); the *cyclopean* image is defined as that given by their average co-ordinates;

$$A_c = \left(\frac{a_{xl} + a_{xr}}{2}, a_y\right) \qquad (9.88)$$

$$B_c = \left(\frac{b_{xl} + b_{xr}}{2}, b_y\right) \qquad (9.89)$$

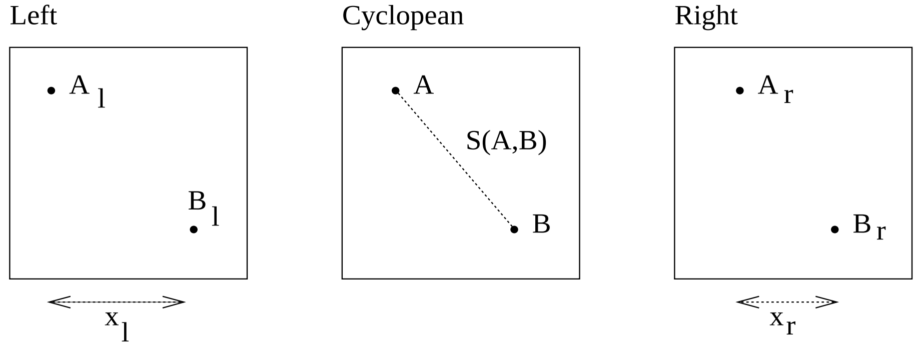


**Figure 9.24** *Definition of the disparity gradient.*

and their **cyclopean separation** $S$ is given by their distance apart in this image;

$$\begin{aligned} S(A,B) &= \sqrt{\left(\left(\frac{a_{xl}+a_{xr}}{2}\right)-\left(\frac{b_{xl}+b_{xr}}{2}\right)\right)^2+(a_y-b_y)^2} \\ &= \sqrt{\frac{1}{4}((a_{xl}-b_{xl})+(a_{xr}-b_{xr}))^2+(a_y-b_y)^2} \\ &= \sqrt{\frac{1}{4}(x_l+x_r)^2+(a_y-b_y)^2} \end{aligned} \tag{9.90}$$

The difference in disparity between the matches of $A$ and $B$ is

$$\begin{aligned} D(A,B) &= (a_{xl}-a_{xr})-(b_{xl}-b_{xr}) \\ &= (a_{xl}-b_{xl})-(a_{xr}-b_{xr}) \\ &= x_l-x_r \end{aligned} \tag{9.91}$$

The disparity gradient of the pair of matches is then given by the ratio of the disparity difference to the cyclopean separation;

$$\begin{aligned} \Gamma(A,B) &= \frac{D(A,B)}{S(A,B)} \\ &= \frac{x_l-x_r}{\sqrt{\frac{1}{4}(x_l+x_r)^2+(a_y-b_y)^2}} \end{aligned} \tag{9.92}$$

Given these definitions, the constraint exploited is that, in practice, the disparity gradient $\Gamma$ can be expected to be limited; in fact, it is unlikely to exceed 1. This means that very small differences in disparity are not acceptable if the corresponding points are extremely close to each other in 3D – this seems an intuitively reasonable observation, and it is supported by a good deal of physical evidence [PMF85]. A solution to the correspondence problem is then extracted by a relaxation process in which all possible matches are scored according to whether they are supported by other (possible) matches that do not violate the stipulated disparity gradient limit. High scoring matches are regarded as correct, permitting firmer evidence to be extracted about subsequent matches.

Algorithm 9.3: PMF stereo correspondence

1. Extract features to match in left and right images. These may be, for example, edge pixels.

2. For each feature in the left (say) image, consider its possible matches in the right; these are defined by the appropriate epipolar line.

3. For each such match, increment its likelihood score according to the number of other possible matches found that do not violate the chosen disparity gradient limit.

4. Any match which is highest scoring for *both* the pixels composing it is now regarded as correct. Using the uniqueness constraint, these pixels are removed from all other considerations.

5. Return to (2) and re-compute the scores taking account of the definite match derived.

6. Terminate when all possible matches have been extracted

---

Note here that the epipolar constraint is used at point (2) to limit to one dimension the possible matches of a pixel, and the uniqueness constraint is used at (4) to ensure that a particular pixel is never used more than once in the calculation of a gradient.

The scoring mechanism has to take account of the fact that the more remote two (possible) matches are, the more likely they are to satisfy the disparity gradient limit. This is catered for by:

- Considering only matches that are "close" to the one being scored. In practice it is typically adequate to consider only those inside a circle of radius equal to seven pixels, centred at the matching pixels (although this number depends on the precise geometry and scene in hand).

- Weighting the score by the reciprocal of its distance from the match being scored. Thus more remote pairs, which are more likely to satisfy the limit by chance, count for less.

The PMF algorithm has been demonstrated to work relatively successfully. It is attractive also because it lends itself to parallel implementation and could be extremely fast on suitably chosen hardware. It has a drawback (along with a number of similar algorithms) in that horizontal line segments are hard to match; they often move across adjacent rasters and, with parallel camera geometry, any point on one such line can match any point on the corresponding line in the other image.

### 9.2.12 Active acquisition of range images

It is extremely difficult to extract 3D shape information from intensity images of real scenes directly. The reader likely noticed this fact. What can be found out about the shape from shading of an intensity image will be explained in a coming section 9.3.

The way around the difficult shape from shading task is to measure distances from the viewer to points on surfaces in the 3D scene. Such measurements are called *geometric signals*, i.e. a collection of 3D points in a known coordinate system. If the surface relief is measured from one viewpoint only it is called a *range image* or a *depth map*. Such an explicit 3D information makes geometry recovery easier.[4] It is closer to the geometric model that is looked for.

Two steps are needed to get geometric information from a range image:

1. The range image should be captured. The related principles are sketched in this very section.

2. The geometric information should be extracted from the range image. Some features are looked for in the range image and they are compared to a selected 3D model. The questions related to the selection of the features and geometric models lead to the one of the most fundamental problems in computer vision, i.e. how to represent a solid shape [Koe90].

Let us explain the ideas of several active sensors that provide range images. Term active here means that the sensor uses and controls own electromagnetic energy or more specifically illumination for measuring a distance between scene surfaces and "observer". An active sensor should not be confused with active perception strategy where the sensing subject plans how to look at objects from different views.

The *RADAR* and the *LIDAR*[5] in one measurement yields the distance between the sensor and the particular point on a measured scene. The sensor is mounted on the assembly that allows movement in two angles, azimuth $\Theta$ and tilt angle $\Phi$ corresponding to spherical coordinates. The distance is proportional to the time interval between the emission of energy and the echo reflected from the measured

[4] There are techniques that measure directly a full 3D information like mechanical coordinate measuring machines or computer tomography. These are treated in section ??.

[5] Acronyms come from: RADAR – RAdio Detecting And Ranging and LIDAR – LIght Detecting And Ranging.

scene object. As the elapsed time intervals are very short the phase difference between the emitted and received signals are often used.

The *RADAR* emits electromagnetic waves in meters, centimeters or milimeters wavelength bands. Besides its military use it is quite often explored for navigation of autonomous guided vehicles.

The *LIDAR* often uses laser as a source of a focused light beam. The higher the power of the laser is the stronger is the reflected signal and accordingly preciser is the measured range. If the LIDAR should work in the environment together with humans then the energy has an upper limit due to potential harm to unprotected eye. Another factor that influences safety of the LIDAR is the diameter of the laser beam. If the beam is to be safe it should not be focused to much. LIDARs have troubles when the object surface is almost tangent to the beam. Very little energy reflects back to the sensor in this case. Measurements of specular surfaces are not very accurate as they scatter the reflected light. Transparent objects cannot be measured with optical lasers too. The advantage of the LIDAR is wide range of measured distances from a tenth of millimeter to several kilometers. The accuracy of the measured range is typically around one hundreds of a millimeter. The LIDAR provides one range in one instant. If the whole range image is to be captured the measurement takes several tenth of seconds as the scene should be scanned.

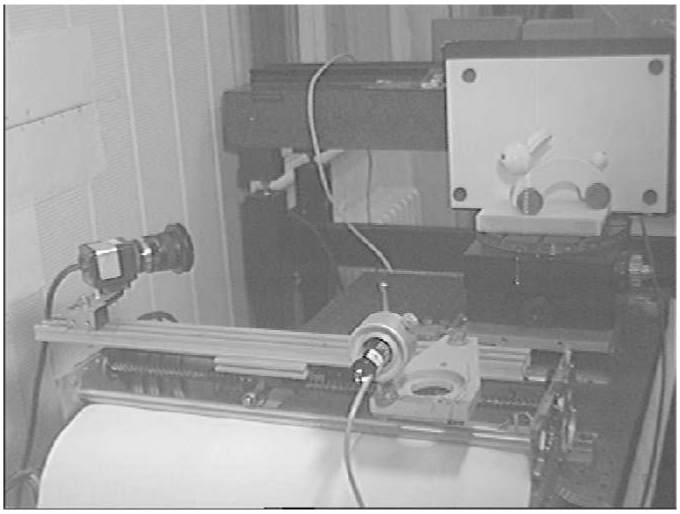

**Figure 9.25** *The bird eye view of the laser plane rangefinder. The camera is on the top left side, Laser diode on the bottom left.*

Another principle of active range imaging is *structured light* triangulation-based one. The geometric arrangement is similar as in the stereo vision with two cameras having parallel optical axes. One camera is replaced by the illuminant that yields a light plane perpendicular to epipolars. The camera is in the fixed distance from the illuminant. The trick is that the correspondence problem, that makes passive stereo so problematic is avoided. There is only one significantly bright point on each image line. Notice that there are problems with selfocclussion in the scene.The distance from the observer can be easily calculated as in Figure 9.10. If the whole range image is to be captured the rod with camera and illuminant should mechanically move relatively to the scene. The trace of the laser should gradually illuminate measured points. The mechanical movement as well as the processing of several hundreds images, i.e. one image for a distinct position of the laser stripe, takes some time. This is typically from couple of seconds to about a minute. The quicker laser stripe range finders find a bright point corresponding to the intersection of a current image line using a special purpose electronics.

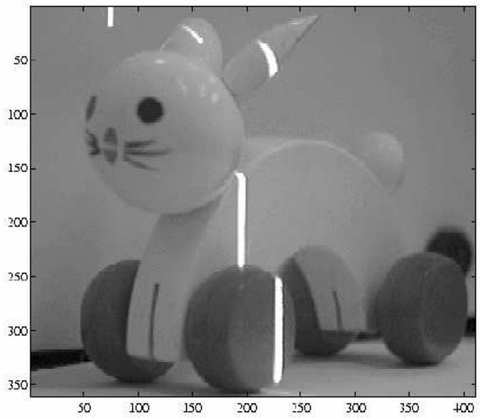


**Figure 9.26** *The image seen by the range finder camera with the bright laser stripe.*

Let us show an example of such the laser stripe range scanner build in the Center for Machine Perception of the Czech Technical University in Prague. The view of the scanner together with the measured object (a wooden toy - a rabbit) and one of the laser planes is shown in Fig. **??**. The image seen by the camera with the

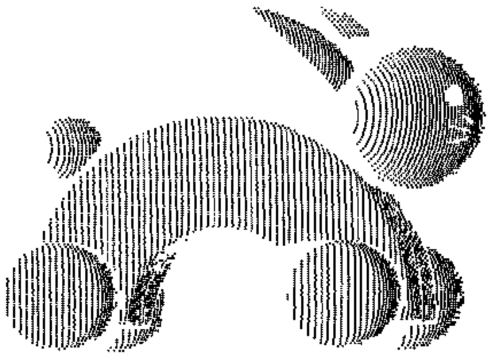

**Figure 9.27** *Reconstructed range image displayed as a point cloud.*

distinct bright laser stripe is in Fig. 9.26. The resulting measured range image is show in Fig. 9.27.

In some applications, a range image should be acquired in an instant, that typically means in one TV frame. This case includes capturing range images of moving objects, e.g. moving humans. There is a possibility to illuminate the scene by several stripes at once and code them. The Figure 9.28 shows a human hand lit by the binary pattern that codes light stripes using cyclic code. The local configuration of squares in the image allows to decode which stripe it is. In this very case, the pattern with coded stripes is projected from a $36{\times}24$ mm slide using almost standard slide projector. The resulting range image does not yield as many samples as in the previous case of moving laser stripe, in our case only $64{\times}80$, see Fig. 9.29.

There is a possibility to get as dense range samples as in the laser stripe case in one TV frame. Individual stripes can be encoded using spectral colours and the image captured by colour TV camera [Smu93].

Before we finish active range sensors section, two more measuring principles

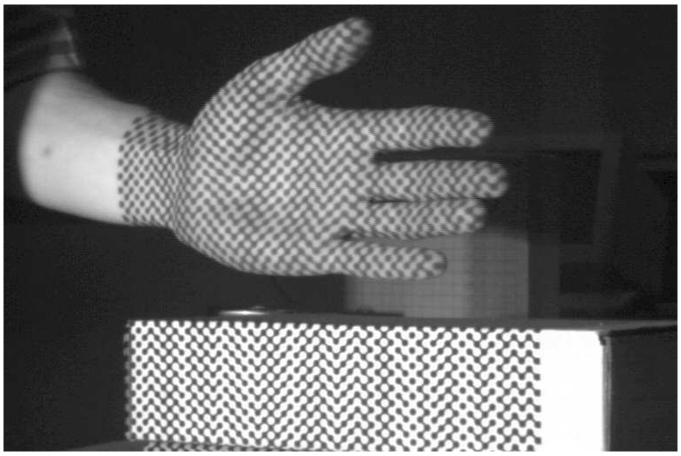

**Figure 9.28** *Illumination of the binary coded range finder.*

should be mentioned. The first one is sonar that uses ultrasonic waves as source of energy. Sonars are used in robot navigation for the close range measurements. The disadvantage is that measurements are typically very noisy.

The second principle is the Moiré interferometry [KKS96]. Two periodic pattern, typically stripes, are projected on the scene. Due to interference is the object covered by the system of closed, non-intersecting curves. Each of the curves lies in the plane of constant distance from the viewer. The obtained distance measurement is only relative. Moiré itself does not provide absolute distance. The properties of Moiré curves are very similar to level curves on maps.

## 9.3 Radiometry and 3D vision

### 9.3.1 Grey level of a captured pixel from radiometric point of view

The TV camera and most other artificial vision sensors measure the amount of received light energy in individual pixels as the outcome of interactions among various materials and light source(s). The measured value is informally called grey level (or brightness). If the mechanism of image creation is to be understood then radiometry is the proper tool. *Radiometry* is a branch of physics that deals measurement of the flow and transfer of radiant energy. The grey level corresponding to a point on the 3D surface depends, informally speaking, on the shape of the object, its reflectance properties, position of the viewer and properties and position of the illuminants [NRH$^+$77]. The aim of this paragraph is to give the reader an initial insight into radiometry concepts related to computer vision. This knowledge will be used later when 3D shape from shading will be

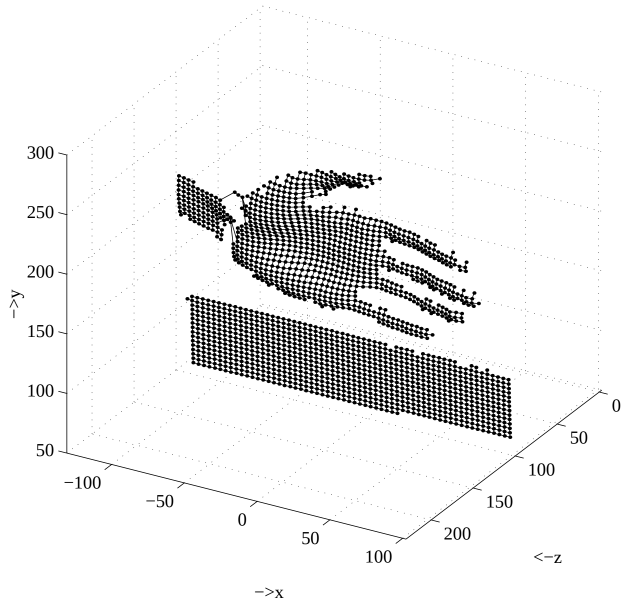


**Figure 9.29** *Reconstructed surfaces from the binary coded range finder.*

explained.

The above mentioned radiometric approach to understanding grey levels is very often avoided in practical applications because of its complexity and numerical unstability. The measured grey level typically does not serve as a precise quantitative measurement[6]. The way around is to use a task-suited illumination that allows to find objects of interest on a qualitative level and separate them from the background.

*Photometry* is a closely related discipline to radiometry that studies sensation of radiant light energy in human eye. Both disciplines describe similar phenomena using quantities that are alike. Physical units are given in square brackets. When there is a danger of confusion we shall denote photometric quantities by subscript $_{ph}$ and leave radiometric ones without subscript.

The basic radiometric quantity is *radiant flux* $\Phi$ $[W]$. Its photometric counterpart is *luminous flux* $\Phi_{ph}$ $[lm$ (=lumen)$]$. Both quantities can be converted as $1\ [W] = 680\ [lm]$, but only for the wavelength of the light $\lambda$=555 $[nm]$ and daylight seeing [HD91]. Photometric quantities depend on the spectral characteristic of the radiation source and on the sensitivity of photoreceptive cells of a human retina. Different people have different ability to perceive light. That was the reason why an international standardization body CIE defined the standard observer that corresponds to average abilities. Let $K(\lambda)$ be the luminous

[6]First reason is that CCD cameras are much more precise geometrically than radiometrically. The second and more serious reason is that the relation between grey level and shape is too complicated.

efficacy $[lm\, W^{-1}]$, $S(\lambda)$ $[W]$ the spectral power of the light source, and $\lambda$ $[W]$, the wavelength. The luminous flux $\Phi_{ph}$ is proportional to the intensity of perception and is given as

$$\Phi_{ph} = \int_\lambda K(\lambda)S(\lambda)d\lambda \tag{9.93}$$

Let us concentrate more on the radiometric quantities as photometric ones are too dependent on the observer. As looked at from a viewer point of view, the surface

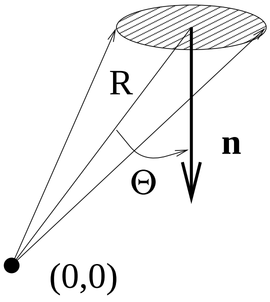


**Figure 9.30** *Spatial angle for an elementary surface area.*

of an object can reflect energy into a halfsphere and differently into different directions. The *spatial angle* is given by the area on the surface of the unit sphere that is bounded by a cone with an apex in the centre of the sphere. The whole halfsphere corresponds to the spatial angle $2\pi$ $[sr$ (=steradians)$]$. The small area $A$ at the distance $R$ from the origin (i.e. $R^2 \gg A$) and with angle $\Theta$ between the normal vector to the area and radiusvector between the origin and the area corresponds to the spatial angle $\Omega$ $[sr]$, see Fig. 9.30.

$$\Omega = \frac{A\,\cos\Theta}{R^2} \tag{9.94}$$

*Irradiance* $E$ $[W\, m^{-2}]$ gives the power of the light energy that falls onto the unit area of the object surface, $E = \delta\Phi/\delta A$, where $\delta A$ is the infinitesimal element of the surface area. The corresponding photometric quantity is *illumination* $[lm\, m^{-2}]$. The *radiance* $L$ $[W\, m^{-2}\, sr^{-1}]$ is the power of light that is emitted from a unit surface area into some spatial angle. The photometric quantity corresponding to radiance is called *brightness* $L_{ph}$ $[lm\, m^{-2}\, sr^{-1}]$. Brightness is used informally in image analysis to describe the quantity that the camera measures.

The irradiance is given by the amount of energy that the image capturing device equipped with the lens gets per unit of an efficient sensitive area of the camera [Hor86]. Grey levels of image pixels are quantized estimates of image irradiance. The efficient area copes with foreshortening that is caused by the mutual rotation between the elementary patch on the emitting surface and the elementary surface patch of the sensor. We shall express the relation between the irradiance $E$ measured in the image and the radiance $L$ produced by a small

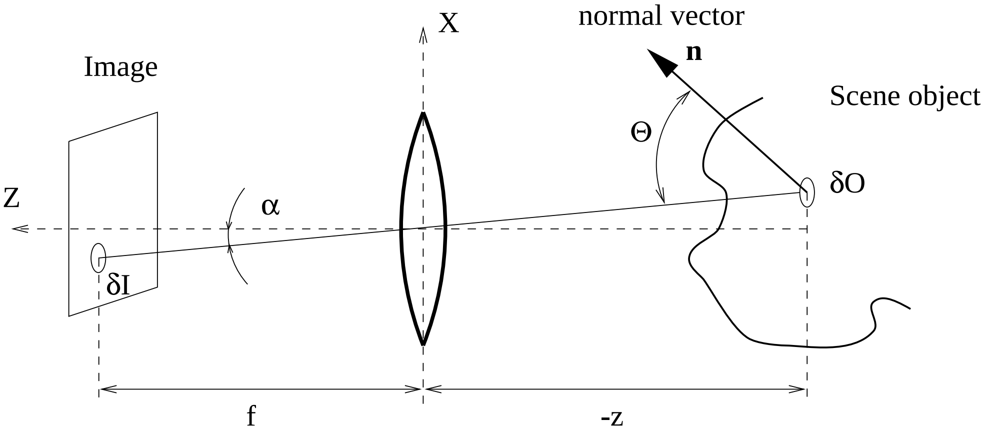


**Figure 9.31** *Sketch of the geometry used to derive relation between irradiance $E$ and the radiance $L$.*

patch on the object surface. Only part of this radiance is captured by the lens of the camera.

The geometry of the capturing setup is sketched in Fig. 9.31. The optical axis is aligned with the horizontal axis $Z$. The lens with the focal length $f$ is placed into the origin of coordinates (optical centre). The elementary object surface patch $\delta O$ is in distance $z$. We are interested how much light energy reaches elementary patch of the sensor surface $\delta I$. The off-axis angle $\alpha$ spans between the axis $Z$ and the line connecting $\delta O$ with $\delta I$. As we deal with a perspective projection, this line must pass through the origin. The elementary object surface patch $\delta O$ is tilted by the angle $\Theta$ measured between the object surface normal $\mathbf{n}$ at the patch and line between $\delta O$ and $\delta I$.

The light rays passing through the lens origin are not refracted. Thus the spatial angle attached to the elementary surface patch in the scene is equal to the spatial angle corresponding to the elementary patch in the image. The foreshortened elementary image patch as seen from the optical centre is $\delta I \ \ \cos\alpha$ and its distance from the optical centre is $\frac{f}{\cos\alpha}$. The corresponding spatial angle is $\frac{\delta I \ \ \cos\alpha}{(\frac{f}{\cos\alpha})^2}$. Analogically the spatial angle corresponding to the elementary patch $\delta O$ on the object surface is $\frac{\delta O \ \ \cos\Theta}{(\frac{z}{\cos\alpha})^2}$. As the spatial angles are equal

$$\frac{\delta O}{\delta I} = \frac{\cos\alpha}{\cos\Theta}\frac{z^2}{f^2} \tag{9.95}$$

Let us calculate how much light energy passes through the lens if its aperture has diameter $d$. The spatial angle $\Omega_L$ that sees the lens from the elementary patch on the object is

$$\Omega_L = \frac{\pi}{4}\frac{d^2 \ \cos\alpha}{(\frac{z}{\cos\alpha})^2} = \frac{\pi}{4}\left(\frac{d}{z}\right)^2 \cos^3\alpha \tag{9.96}$$

Let $L$ be the radiance of the object surface patch that is oriented towards the lens. Then the elementary contribution to the radiant flux $\Phi$ falling at the lens is

$$\delta\Phi = L\delta O \;\; \Omega_L \,\cos\Theta = \pi L\delta O \;\left(\frac{d}{z}\right)^2 \frac{\cos^3\alpha\,\cos\Theta}{4} \tag{9.97}$$

The lens concentrates the light energy into the image. If the energy loses in the lens are neglected and no other light falls on the image element we can express the irradiation $E$ of the elementary image patch

$$E = \frac{\delta\Phi}{\delta I} = L\frac{\delta O}{\delta I}\frac{\pi}{4}\left(\frac{d}{z}\right)^2 \cos^3\alpha\,\cos\Theta \tag{9.98}$$

If we substitute for $\frac{\delta O}{\delta I}$ from the equation (9.95) we get an important equation that tells how the scene radiance influences the irradiation in the image

$$E = L\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha \tag{9.99}$$

The term $\cos^4\alpha$ describes an systematic optical defect of the lens called *vignetting*[7] that tells that optical rays with bigger span-off angle $\alpha$ are attenuated more. This means that pixels closer to the image borders are darker. This effect is more severe with wide angle lenses than with tele-lenses. Since vignetting is a systematic error it can be compensated for a radiometrically calibrated lens.

The term $\frac{d}{f}$ is called the $f$-number of the lens and tells how much the lens differs from a pinhole model.

### 9.3.2 Surface reflectance

The grey level in the image pixel as an estimate of image irradiance is a result of light reflection from scene object in many vision applications. Thus it is needed to understand different mechanisms involved in the reflection. In this paragraph, we will give only a sketchy overview that later allows to explain the main idea of shape from shading. Consult [Ike94, FDFH90, KKS96] for more detailed explanation.

The radiance of an opaque object that does not emit own energy depends on irradiance caused by other sources of energy. The illumination that the viewer perceives depends on the strength, position, orientation, type (point or diffuse) of the light sources, ability of the object surface to reflect energy and the local surface orientation (given by its normal vector).

Another needed concept is the *gradient space* that is a way of describing surface orientations, and is a concept of general value that has also been used to

[7] One of the meanings of *vignette* is: a photograph or drawing with edges that are shaded off.

help in the analysis of line labelling problems [Mac73]. Let $z(x, y)$ be the surface height. We proceed by noting that at nearly every point a surface has a unique normal $\mathbf{n}$. The components of surface gradient

$$p = \frac{\partial z}{\partial x} \qquad \text{and} \qquad q = \frac{\partial z}{\partial y} \tag{9.100}$$

can be used to specify the surface orientation. Let us express the unit surface normal using surface gradient components. If we move a small distance $\partial x$ in $x$-direction the change of height is $\partial z = p\partial x$. Thus the vector $[1, 0, p]^T$ is the tangent to the surface. Analogically $[0, 1, q]^T$ is also the tangent to the surface. The surface normal is perpendicular to all its tangents that can be computed using the vector product as

$$\begin{bmatrix} 1 \\ 0 \\ p \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ q \end{bmatrix} = \begin{bmatrix} -p \\ -q \\ 1 \end{bmatrix} \tag{9.101}$$

The unit surface normal $\mathbf{n}$ can be written as

$$\mathbf{n} = \frac{1}{\sqrt{1 + p^2 + q^2}} \begin{bmatrix} -p \\ -q \\ 1 \end{bmatrix} \tag{9.102}$$

Here we suppose that the $z$-component of the surface normal is positive as only the surface part oriented towards the viewer is visible.

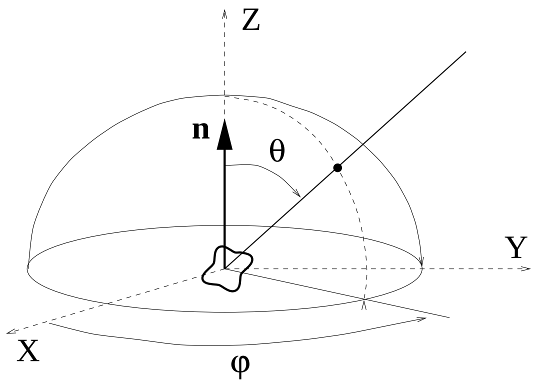


**Figure 9.32** *Polar and spherical angles used to describe orientation of the surface patch.*

The pair $[p, q]$ is the two dimensional gradient space representation of the surface orientation. The gradient space has a number of attractive properties that allow elegant description of the surface. Interpreting the image plane as $z = 0$, we see that the origin of gradient space corresponds to the vector $[p, q]=[0, 0]$, that is normal to the image plane. Thus $[p, q]=[0, 0]$ implies that the surface is parallel to the image plane. The more remote a vector is from the origin of gradient space, the steeper its corresponding surface patch is inclined to the image plane.

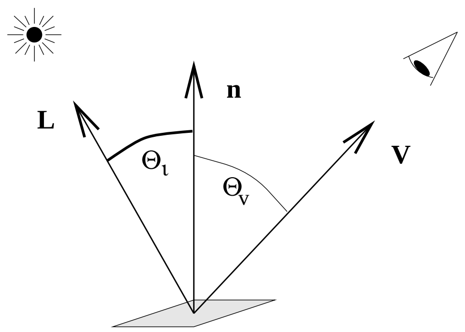


**Figure 9.33** *Directions towards the viewer and the light source.*

Let us describe spherical coordinates that are used to express the geometry of the infinitesimal surface patch, cf. Fig. 9.32. The *polar angle*[8] is $\Theta$ and the *azimuth* is $\varphi$. We shall attempt to describe ability of different materials to reflect light. The direction towards point light source is denoted by subscript $_i$, i.e. $\Theta_i$ and $\varphi_i$. Similarly, subscript $_v$ identifies the direction toward the viewer, $\Theta_v$ and $\varphi_v$, see Fig. 9.33. The irradiance of the elementary surface patch from the light source is $\delta E(\Theta_i, \varphi_i)$. The elementary contribution of the radiance in direction towards the viewer is $\delta L(\Theta_v, \varphi_v)$. In general, the ability of the body to reflect light is described using *bidirectional reflectance distribution function* $f_r$ $[sr^{-1}]$, abbreviated BDRF [NRH+77].

$$f_r(\Theta_i, \varphi_i; \Theta_v, \varphi_v) = \frac{\delta L(\Theta_v, \varphi_v)}{\delta E(\Theta_i, \varphi_i)} \tag{9.103}$$

The BDRF $f_r$ tells how bright si the elementary surface patch for a specific material, light source, and viewer directions. The modeling of the BDRF is important also for realistic rendering in computer graphics [FDFH90]. The BDRF in its full complexity (equation 9.103) is used for modeling reflection properties of materials with oriented microstructure (e.g. tiger's eye - a semiprecious golden-brown stone, peacock's feather, rough cut of alluminium).

Fortunately, for most practically applicable surfaces, the BDRF remain constant if the elementary surface patch rotates along the normal vector to the surface. In this case the BDRF is simplified and depends on the $\varphi_i - \varphi_v$, i.e. $f_r(\Theta_i, \Theta_v, \varphi_i - \varphi_v))$. This simplification holds for both ideal diffuse (Lambertian) surfaces and for ideal mirror.

Let us explain the concept of reflectance function. Let $E_i(\lambda)$ denote the irradiance caused by the illumination of the surface element and $E_r(\lambda)$ the energy

[8] $\Theta$ is also called zenith angle.

flux per unit area scattered by the surface element back to the whole half-space. The ratio

$$\rho(\lambda) = \frac{E_r(\lambda)}{E_i(\lambda)} \tag{9.104}$$

is called *reflectance coefficient* or, briefly, *albedo*. Albedo tells what is the portion of incident energy that is reflected back to the half-space. Let us make an simplifying assumption, neglect the colour properties of the surface and suppose that albedo does not depend on the wavelength $\lambda$.

This portion is an integral of the surface radiance $L$ over the solid angle $\Omega$ representing the half-space.

$$E_r = \int_\Omega L \, d\,\Omega \tag{9.105}$$

Let us introduce a reflectance function $R(\Omega)$ that models the influence of the local surface geometry into the spatial spread of the reflected energy. The $\Omega$ is an infinitesimal solid angle around the viewing direction.

$$\int_\Omega R \, d\,\Omega = 1 \tag{9.106}$$

In general, the surface reflectance properties depend on three angles among the direction to the light source $\mathbf{L}$, direction towards the viewer $\mathbf{V}$ and local surface orientation given by the surface normal $\mathbf{n}$, recall Fig. 9.33. The cosines of those angles can be expressed as dot products of vectors. Thus the reflectance function is a scalar function of the following three dot products

$$R = R(\mathbf{n} \cdot \mathbf{L}, \mathbf{n} \cdot \mathbf{V}, \mathbf{V} \cdot \mathbf{L}) \tag{9.107}$$

The *Lambertian surface* (also ideally opaque, with ideal diffusion) reflects light energy to all direction. Thus is the radiance to all direction constant. The BDRF $f_{\mathrm{Lambert}}$ is constant

$$f_{\mathrm{Lambert}}(\Theta_i, \Theta_v, \varphi_i - \varphi_v) = \frac{\rho(\lambda)}{\pi} \tag{9.108}$$

If the constant albedo $\rho(\lambda)$ is assumed then the Lambertian surface reflectance can be expressed as

$$R = \frac{1}{\pi}\mathbf{n} \cdot \mathbf{L} = \frac{1}{\pi}\cos\Theta_i \tag{9.109}$$

Because of its simplicity, the Lambertian reflectance function has been widely accepted as a reasonable relflectance model for shape from shading. Notice that the reflectance function for the Lambertian surface is independent of the viewing direction $\mathbf{V}$.

The dependence of the surface radiance on the local surface orientation can be expressed in gradient space. The *reflectance map* $R(p, q)$ is used for this purpose. The $R(p,q)$ can be visualized in the gradient space as nested isocontours corresponding to the same observed irradiance.

Values of the reflectance map can be: (1) measured experimentally on the device called the goniometer stage that is able to set mechanically angles $\Theta$ and $\varphi$. The sample of the surface is attached to the goniometer and its reflectance measured for different orientation of a viewer and light sources. (2) set experimentally if the calibration object is used. Typically the halfsphere is utilized for this purpose. (3) derived from a mathematical phenomenonical model describing surface reflecting properties[9].

The irradiance $E(x,y)$ of an infinitely small light sensor located at position $x,y$ in the image plane is equal to the surface radiance at corresponding surface patch given by its surface parameters $u,v$ if the light is not attenuated in the optical medium between the surface and the sensor. This important relation between surface orientation and perceived image intensity is called the *image irradiance equation.*

$$E(x,y) = \rho(u,v) R(\mathbf{N}(u,v)\cdot\mathbf{L}, \mathbf{N}(u,v)\cdot\mathbf{V}, \mathbf{V}\cdot\mathbf{L}) \tag{9.110}$$

In the attempt to reduce complexity, *several simplifying assumptions* [Hor90] are usually made to ease the shape from shading task. It is assumed that:

- The object has uniform reflecting properties, i.e. $\rho(u,v) = \text{const}$.
- The light sources are very far. Then the irradiation in different places in the scene is approximately the same and the incident direction towards the light sources is the same.
- The viewer is very distant. Then the radiance emitted by the scene surfaces does not depend on the position but only on its orientation. The perspective projection is simplified to orthographic one.

Let us write the simplified version of the image irradiance equation for the Lambertian surface, constant albedo, single distant illuminant, distant viewer in the same direction as illuminant, and the reflectance function $R$ expressed in gradient space $(p,q)$.

$$E(x,y) = \beta R(p(x,y), q(x,y)) \tag{9.111}$$

The $R(p,q)$ gives the radiance of the corresponding point in the scene. The proportionality constant $\beta$ comes from equation (9.99) and depends on the $f$-number of the used lens. The vignetting degradation of the lens is negligible

[9] The most known surface reflectance models are Lambertian model for ideally opaque surface, Phong model that copes with reflection from dielectric materials, Torrance-Sparrow model the surface as collection of planar mirror-like microfacets with normally distributed normals, and wave theory based Beckmann-Spizzichino model. The survey of the surface reflection models from the computer vision point of view and their recent modifications can be found in [Ike94].

as the viewer is aligned to the illuminant. The measured irradiance $E$ can be normalized and the factor $\beta$ can be omitted. This allows to write *image irradiance equation in the simplest form* as

$$E(x, y) = R(p(x, y), q(x, y)) = R(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}) \tag{9.112}$$

The image irradiance equation in its simplest form is a first-order differential equation. It is typically nonlinear as the reflectance function $R$ in most cases depends nonlinearly on the surface gradient. This is the basic equation that is used to recover surface orientation from the intensity image.

### 9.3.3 Shape from shading

The human brain is able to make very good use of clues from shadows and shading in general. Not only do detected shadows give a clear indication of where occluding edges are, and the possible orientation of their neighbouring surfaces, but general shading properties are of great value in deducing depth. A fine example of this is a photograph of a face; from a straight-on, 2D representation, our brains make good guesses about the probable lighting model, and then deductions about the 3D nature of the face – for example, deep eye sockets and protuberant noses or lips are often recognizable without difficulty.

Recall that the intensity of a particular pixel depends on the light source(s), surface reflectance properties, and local surface orientation expressed by surface normal $\mathbf{n}$. The aim of the shape from shading is to extract information about normal of surfaces in view on the basis of solely intensity image. If simplifying assumptions are made about illumination, surface reflectance properties, and surface smoothness the shape from shading task has proven to be solvable. The first computer vision related formulation comes from Horn [Hor70, Hor75].

Techniques similar to shape from shading were proposed independently in photoclinometry a bit earlier [Rin66]. Astrogeologists wanted to measure steepness of slopes on planets in the solar system from intensity images observed by telescopes on the Earth. There are two main items that make photoclinometry different from shape from shading: (a) The surface normals are calculated by the integration along a space curve (called the profile; it is an 1D entity if the curve is arclength parametrized). In shape from shading, the integration si performed on the surface area which is a 2D entity if the surface is parametrized. (b) Shape from shading is more concerned with ambiguity of the obtained solution. The use of singular points and occluding boundaries helps to combat this ambiguity. The surface normal can be then uniquely computed.

Let us classify shape from shading methods into three categories and describe each of them:

**Methods propagating solution incrementally from surface points of known heights**

Let us start our description with the oldest solution propagation method (called characteristic strip method) where the solution is developed along a space curve. This method is also the easiest to explain.

We can begin to analyse the problem of global extracting shape from shading information when the reflectance function and the lighting model are both known perfectly [Hor90]. Even given these constraints, it should be clear that the mapping "surface orientation to brightness" is many-to-one. There are many orientations that can produce the same point intensity. Acknowledging this, a particular brightness can be produced by an infinite number of orientations that can be plotted as a (continuous) line in gradient space. An example for the simple case of a light source directly adjacent to the viewer, incident on a matte surface, is shown in Figure 9.34 – two points lying on the same curve (circles in this case) indicate two different orientations that will reflect light of the same intensity, thereby producing the same pixel grey level. The original formulation [Hor70] to

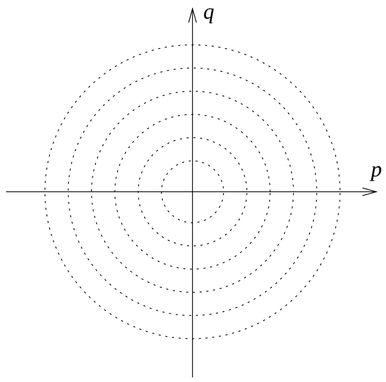


**Figure 9.34** *Reflectance map for a matte surface – the light source is adjacent to the viewer.*

the general shape from shading task assumes a Lambertian surface, one distant point light source, a distant observer, and no interreflections in the scene. The proposed method is based on the notion of a *characteristic strip.* Suppose that we already calculated coordinates of a surface point $[x, y, z]^T$ and we want to propagate the solution following an infinitesimal step on the studied surface, e.g. taking small steps $\delta x$ and $\delta y$. We need to calculate the change in the height $\delta z$. This can be done if the components of the surface gradient $p$, $q$ are known, cf. equation( 9.100). Let us use the compact notation to express partial derivatives using indices, e.g. $p{=}\delta z/\delta x$ is shortened to $z_x$. Similarly for second derivatives, e.g. $z_{xx}{=}\delta^2 x/\delta x^2$. The infinitesimal change of height is

$$\delta z = p\,\delta x + q\,\delta y \tag{9.113}$$

As the surface is followed stepwise, the values of $p$, $q$ should be recorded in addition to $x$, $y$, $z$. The changes in $p$, $q$ should be calculated using second derivatives of height $r{=}z_{xx}$, $s{=}z_{xy}{=}z_{yx}$, $t{=}z_{yy}$

$$\delta p = r\,\delta x + s\,\delta y \qquad \text{and} \qquad \delta q = s\,\delta x + t\,\delta y \tag{9.114}$$

As we like to avoid expressing infinitely many changes of changes we explore image irradiance equation $E(x,y) = R(p,q)$, cf. equation (9.112). Let differentiate the image irradiance equation with respect to $x$, $y$ to obtain brightness gradient

$$E_x = rR_p + sR_q \qquad \text{and} \qquad E_y = sR_p + tR_q \tag{9.115}$$

The direction of the step $\delta x$, $\delta y$ can be arbitrarily chosen.

$$\delta x = R_p \xi \qquad \text{and} \qquad \delta y = R_q \xi \tag{9.116}$$

The parameter $\xi$ changes along particular solution curve. Moreover, the orientation of the surface along this curve is known. Thus it is called the the characteristic strip.

Now we can express changes of gradient $\delta p$, $\delta q$ as dependent on the gradient image intensities. This is the crucial "trick". We explore equations (9.114) and (9.115). The derived results can be summarized as the set of the ordinary differential equations; dot denotes differentiation with respect to $\xi$

$$\dot{x} = R_p\,, \quad \dot{y} = R_q\,, \quad \dot{z} = p\,R_p + q\,R_q\,, \quad \dot{p} = E_x\,, \quad \dot{q} = E_y \tag{9.117}$$

There are points on the surface for which the surface orientation is known in advance and yield as boundary conditions during normal vector calculations. These are (a) points of the surface occluding boundary and (b) singular points in the image.

The *occluding boundary* which is a curve on the surface due to surface rolling away from the viewer, i.e the set of points for which the local tangent plane contains the direction towards the viewer. The surface normal at the occlusion boundary can be uniquely determined as it is parallel to the image plane and perpendicular to the direction towards the viewer. The surface normal information can be propagated into the recovered surface patch from the occluding boundary. Although the occlusion boundary constraints surface orientation uniquely, it does not sufficiently constrain the solution in the explicit depth form [Oli91].

We have seen that at the most surface points its gradient is not fully constraint by the image intensities. Let us suppose that the reflectance function $R(p,q)$ has a global maximum, i.e. $R(p,q) < R(p_0,q_0)$ for all $[p,q] \neq p_0, q_0$. This maximum corresponds to the singular points in the image.

$$E(x_0,y_0) = R(p_0,q_0) \tag{9.118}$$

The surface normal is parallel to the direction towards the light source at singular points. Singular points are in general sources and sinks of characteristic stripes.

It is reported [Hor90] that direct implementation of the easy to explain characteristic strip method does not yield particularly good results due to numerical unstability.

### Global optimization methods

The methods are formulated as the variational task where the whole image plays role in the chosen functional. The obtained results are in general better than in the methods propagating solution incrementally that have been described recently.

Let us outline the basic idea of global shape from shading methods now. We already know that under the simplifying conditions for recovery surface normals from intensities (stated in section 9.3.2) the image irradiance equation (9.112) relates image irradiance $E$ and surface reflection $R$ as simply as

$$E(x, y) = R(p(x, y), q(x, y)) \tag{9.119}$$

The task is to find the surface hight $z(x, y)$ given the image $E(x, y)$ and reflectance map $R(p, q)$.

Now presented with an intensity image, a locus of possible orientations for each pixel can be located in gradient space, immediately reducing the number of possible interpretations of the scene. Of course, at this stage, a pixel may be part of any surface lying on the gradient space contour; to determine which, another constraint needs to be deployed. The key to deciding which point on the contour is correct is to note that “almost everywhere” 3D surfaces are smooth, in the sense that neighbouring pixels are very likely to represent orientations whose gradient space positions are also very close. This additional constraint allows a relaxation process to determine a best fit (minimum-cost) solution to the problem. The details of the procedure are very similar to those used to extract optical flow, and are discussed more fully in Section **??**, but may be summarized as:

| Algorithm 9.4: Extracting shape from shading |
|---|

1. For each pixel $(x, y)$, select an initial guess to orientation $p^0(x, y)$, $q^0(x, y)$.
2. Apply two constraints:
   (a) The observed intensity $f(x, y)$ should be close to that predicted by the reflectance map $R(p, q)$ derived from foreknowledge of the lighting and surface properties.
   (b) $p$ and $q$ vary smoothly – therefore their Laplacians $\nabla^2 p$ and $\nabla^2 q$ should be small.

3. Apply the method of Lagrange multipliers to minimize the quantity

$$\Sigma_{(x,y)}\text{Energy}(x,y) \tag{9.120}$$

where

$$\text{Energy}(x,y) = (f(x,y) - R(p,q))^2 + \lambda((\nabla^2 p)^2 + (\nabla^2 q)^2) \tag{9.121}$$

---

The first term of equation (9.121) is the intensity "knowledge", while the second, in which $\lambda$ is a Lagrange multiplier, represents the smoothness constraint. Deriving a suitable solution to this problem is a standard technique that iterates $p$ and $q$ until $E$ falls below some reasonable bound; a closely related problem is inspected in more detail in Section **??**, and the method of optimization using Lagrange multipliers is described in many books [Hor86].

The significant work in this area is due to Horn [Hor75] and Ikeuchi [IH81] and predated the publication of Marr's theory, being complete in the form presented here by 1980. Shape from shading, as implemented by machine vision systems, is often not as reliable as other "Shape from" techniques since it is so easy to confuse with reflections, or for it to fail through poorly modelled reflectance functions. This observation serves to reinforce the point that the human recognition system is very powerful since in deploying elaborate knowledge, it does not suffer these drawbacks. A review of significant developments in the area since may be found in [HB89].

**Local shading analysis**

Local shading analysis methods use only a small neighbourhood of the current point on the surface. They look for direct relation between the differential surface structure and the local structure of the corresponding intensity image. The surface is treated as "surface in small", i.e. defined in some small local neighbourhood of one of its points. Only an estimate of the local surface orientation is available not the information about the height of the particular surface point.

The main advantage of the local shading analysis is that it provides surface related information to higher-level vision algorithms from single monocular intensity image without need to reconstruct the surface in explicit depth form [Š95]. This is possible since the intensity image is strongly related to the local surface orientation. The change of an intensity value in a particular direction to the surface curvature in that direction. The surface normal and the shape operator ("curvature matrix") form a natural shape model that can be recovered from an intensity image by local computations. This is, of course, much faster than the solution propagation or global variational methods.

The fundamental contribution to the local shading analysis formulation comes from Pentland [Pen84]. The overview of the local shading analysis can be found in [PB94]. We would like to point at the recent results of Šára [Š94]: (i) It was known that the local surface orientation and the Gaussian curvature sign can be determined uniquely at the occlusion boundary. The new result is that the orientation on the self-shadow boundary can be determined uniquely as well. The self-shadow contour is thus a rich source of unambiguous information about the surface. (ii) The differential properties of isophotes (curves of constant image intensity) are well related to the properties of underlying surface. Isophotes are projections of curves of constant slant from the light direction if the surface reflectance is space-invariant or the illuminant is located at the vantage point.

### 9.3.4 Photometric stereo

Woodham proposed photometric stereo as a method that recovers surface orientation unambiguously, assuming a known reflectance function [Woo80]. Let us suppose that one particular Lambertian surface with varying albedo $\rho$ is studied. The key idea of photometric stereo is to look at the surface from one fixed viewing direction while changing direction of incident illumination. Assume three or more such images of the Lambertian surface. Then, the surface normals can be uniquely determined based on the shading variations in observed images.

The lines of constant reflectance on the surface map to lines of constant irradiation $E$ in the image (called also isophotes). These curves observed in images correspond to second order polynomials. The local surface orientation $\mathbf{n} = [p, q]$ is constrained along a second order curve in the reflectance map. For different illumination direction, the surface reflectance remains the same on the surface but the observed reflectance map $R(p, q)$ changes. This provides additional constraint on possible surface orientation that is again a second order polynomial. Two views corresponding to two distinct illumination directions are not enough to determine uniquely the surface orientation $[p, q]$. The third view is needed to get unique solution. If more than three distinct illuminations are at hand the overdetermined set of equation can be solved.

A practical setup for image capturing consists of one camera and K point illumination sources, $K \geq 3$ with known intensities and illumination directions $L_1, \ldots, L_K$. Only one of the light sources is switched on at the same moment. The setup should be photometrically calibrated to take into account light source intensities, particular camera gain, and offset. The calibration is described in [HS93]. After photometric calibration, the images give $K$ estimates of image irradiances $E_i(x, y)$; $i = 1, \ldots, K$.

If not all light is reflected from a surface then albedo $\rho$, $0 \leq \rho \leq 1$, occurs in the image irradiance as was shown equation (9.110). For the Lambertian surface the image irradiance equation simplifies to

$$E(x, y) = \rho \, R(p, q) \tag{9.122}$$

Recall equation (9.109), called cosine law, showing that the reflectance map of the Lambertian surface is given by the dot product of the surface normal $\mathbf{n}$ and the direction of the incident light $L_i$. If the surface reflectance map is substituted into equation (9.122) we get $K$ image irradiance equations

$$E_i(x, y) = \rho \, L_i \, \mathbf{n} \, , \qquad i = 1, \ldots, K \tag{9.123}$$

For each point $x, y$ in the image we get vector of image irradiances $\mathbf{E} = [E_1, \ldots, E_K]^T$. The light directions can be written in a form of $K \times 3$ matrix

$$L = \begin{bmatrix} L_1 \\ \vdots \\ L_K \end{bmatrix} \tag{9.124}$$

At each image point, the system of image irradiance equations can be written

$$\mathbf{E} = \rho \, L \, \mathbf{n} \tag{9.125}$$

The matrix L does not depend on the pixel position in the image. We can express a vector representing simultinously surface albedo and a local surface orientation.

If we have three light sources, $K = 3$, we get solution using inverse of a regular matrix $L$

$$\rho \, \mathbf{n} = L^{-1} \, \mathbf{E} \tag{9.126}$$

The unit normal is obtained if the albedo is divided out.

$$\mathbf{n} = \frac{L^{-1} \, \mathbf{E}}{||L^{-1} \, \mathbf{E}||} \tag{9.127}$$

For more than three light sources the pseudoinversion of the rectangular matrix is used to get solution in the least square sense

$$\mathbf{n} = \frac{(L^T \, L)^{-1} \, L^T \, \mathbf{E}}{||(L^T \, L)^{-1} \, L^T \, \mathbf{E}||} \tag{9.128}$$

Let us note that the pseudoinversion (or inversion in equation (9.127) can be done only once as it is independent of the pixel position $x, y$ in the image. This calculation is repeated for each image pixel $x, y$ and the estimate of the corresponding normals is obtained.

# References


[AH88] N. Ayache and C. Hansen. Rectification of images for binocular and trinocular stereovision. In *Proceedings of the 9th International Conference on Pattern Recognition, Rome, Italy*, pages 11–16, Los Alamitos, Ca., USA, 1988. IEEE Computer Society Press.

[Alo92] Y. Aloimonos, editor. *Special Issue on Purposive and Qualitative Active Vision*, volume 56, 1992.

[Alo93] Y. Aloimonos, editor. *Active Perception.* Lawrence Erlbaum Associates, Inc., Publishers, Hillsdale, New Jersey, 1993.

[AR94] Y. Aloimonos and A. Rosenfeld. Principles of computer vision. In Young [You94], pages 1–15.

[AS89] Y. Aloimonos and D. Shulman. *Integration of Visual Modules - An Extension of the Marr Paradigm.* Academic Press, New York, 1989.

[Baj88] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, 1988.

[BGB79] R. A. Brooks, R. Greiner, and T. O. Binford. The ACRONYM model-based vision system. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-6, Tokyo*, pages 105–113, 1979.

[Bie87] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[BJ85] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, March 1985.

[BJ88] P. J. Besl and R. Jain. *Surfaces in range image understanding.* Springer-Verlag, New York, 1988.

[Bow92] K. W. Bowyer, editor. *Special issue on directions in CAD-based vision*, volume 55, 1992.

[BP96] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272:1905–1909, June 28 1996.

[BPT88] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. *IEEE Proceedings*, 76:869–889, 1988.

[FA93] A. Farshid and J. K. Aggarwal. Model-based object recognition in dense range images – a review. *ACM Computing Surveys*, 25(1):5–43, March 1993.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, 1993.

[FDFH90] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics - Principles and Practice.* Addison-Wesley Publishing Company, second edition, 1990.

[FJ91] P. J. Flynn and A. K. Jain. CAD-based computer vision: From CAD models to relational graphs. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13(2):114–132, February 1991.

[FJ92] P. J. Flynn and A. K. Jain. 3D object recognition using invariant feature indexing of interpretation tables. *Computer Vision, Graphics, and Image Processing*, 55(2):119–129, 1992.

[FLM92] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision*, pages 321–333, 1992.

[FM95] O. Faugeras and B. Mourrain. On the geometry and algera of the point and line correspondences between n images. In *Proceedings of the 5th International Conference on Computer Vision*, pages 951–956, Boston, USA, June 1995. IEEE.

[GL89] G. H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[Goa86] C. Goad. Fast 3D model-based vision. In A.P. Pentland, editor, *From Pixel to Predicates*, pages 371–374. Ablex Publishing Corporation, Norwood, New Jersey, USA, 1986.

[Har92] R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proceedings of the 2nd European Conference on Computer Vision*, volume LNCS 588, pages 579–587, Heidelberg, Germany, May 1992. Springer - Verlag,.

[Har94] R. I. Hartley. Self-calibration from multiple views with a rotating camera. In *Proceeding of the European Conference on Computer Vision*, pages A:471–478. Springer-Verlag, Heidelberg, 1994.

[Har95] R. I. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, Boston, USA, June 1995. IEEE.

[HB89] B. K. P. Horn and M. J. Brooks, editors. *Shape from Shading.* MIT Press, Cambridge, Ma, 1989.

[HD91] O. S. Heavens and R. W. Ditchburn. *Insight into optics.* John Wiley & Sons, Chichester, U.K., 1991.

[HLW96] V. Hlaváč, A. Leonardis, and T. Werner. Automatic selection of reference views for image-based scene representations. In *Proceedings of the European Conference in Computer Vision*, volume 1 of *Lecture Notes in Computer Science, No. 1064*, pages 526–535, Cambridge, U.K., April 1996. Springer-Verlag, Heidelberg, Germany.

[HMDB95] R. Horaud, R. Mohr, F. Dornaika, and B. Boufama. The advantage of mounting a camera onto a robot arm. In *Proc. of the Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision, Xian, China*, pages 206–213, 1995.

[Hor70] B. K. P. Horn. *Shape from shading: A method for obtaining the shape of a smooth opaque sobject from one view.* PhD thesis, M.I.T., Department of Electrical Engineering, Cambridge, MA., USA, 1970.

[Hor75] B. K. P. Horn. Shape from shading. In P. H. Winston, editor, *The Psychology of Computer Vision.* McGraw Hill, New York, 1975.

[Hor86] B. K. P. Horn. *Robot Vision (Russian Translation 1989).* The MIT Press, Cambridge, Massachusetts, USA, 1986.

[Hor90] B. K. P. Horn. Height and gradient from shading. *International Journal of Computer Vision*, 5(1):37–75, 1990.

[HS93] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision, Volume II.* Addison Wesley, Reading, Ma., 1993.

[IH81] K. Ikeuchi and B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.

[Ike94] K. Ikeuchi. Surface reflection mechanism. In Young [You94], pages 131–160.

[JKS95] R. Jain, R. Kasturi, and B. G. Schunk. *Machine Vision.* McGraw-Hill, New York, USA, 1995.

[Jul90] B. Julesz. Binocular depth perception of computer-generated patterns. *Bell Syst. Tech. Journal*, 39, 1990.

[KKS96] R. Klette, A. Koschan, and K. Schlüns. *Computer Vision - Räumliche Information aus digitalen Bildern.* Friedr. Vieweg & Sohn, Braunschweig, 1996.

[Kli91] G. J. Klir. *Facets of System Science.* Plenum Press, New York, USA, 1991.

[Koe90] J. J. Koenderink. *Solid Shape.* The M.I.T. Press, 1990.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstruction a scene from two projections. *Nature*, 293(10):133–135, September 1981.

[Mac73] A. K. Mackworth. Interpreting pictures of polyhedral scenes. *Artificial Intelligence*, 4(2):121–137, 1973.

[Mar82] D. Marr. *Vision – A Computational Investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman and Co., San Francisco, 1982.

[MF92] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[MH80] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings Royal Society London*, B 207:187–217, 1980.

[Moh93] R. Mohr. Projective geometry and computer vision. In C.H. Chen, L.F. Pau, and P.S.P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, chapter "2.4", pages 369–393. World Scientific, Singapore, 1993.

[MP79] D. Marr and T. A. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society*, B 207:301–328, 1979.

[NFJ93] T. S. Newman, P. J. Flynn, and A. K. Jain. Model-based classification of quadric surfaces. *CVGIP: Image Understanding*, 58(2):235–249, September 1993.

[Nis84] H. K. Nishihara. Practical real-time imaging stereo matcher. *Optical Engineering*, 23(5):536–545, 1984.

[NRH+77] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, I.W. Ginsberg, and T. Limperis. Geometrical considerations and nomeclature for reflectance. U.S. Department of Commerce, National Bureau of Standards, Washington D.C., USA, 1977.

[Oli91] J. Oliensis. Shape from shading as a partially well-constrained problem. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54(2):163–183, September 1991.

[PB94] A. P. Pentland and M. Bichsel. Extracting shape from shading. In Young [You94], pages 161–183.

[Pen84] A. P. Pentland. Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):170–187, March 1984.

[PH95] Tomáš Pajdla and Václav Hlaváč. Camera calibration and Euclidean reconstruction from known translations. Presented at the workshop *Computer Vision and Applied Geometry*, Nordfjordeid, Norway, August 1–7 1995.

[PM97] B. Prescott and G. F. McLean. Line-based correction of radial lens distortion. *GMIP*, 59(1):39–77, January 1997.

[PMF85] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[PT+92] W. H. Press, , S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, Cambridge, U.K., 2nd edition, 1992.

[PTK85] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[Rin66] T. Rindfleisch. Photometric method form lunar topography. *Photogrammetric Engineering*, 32(2):262–267, 1966.

[SK63] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry.* Oxford University Press, Ely House, London W., 1963.

[SL92] M. Soucy and D. Laurendeau. Surface modeling from dynamic integration of multiple range views. In *Proceedings of the 11th International Conference on Pattern Recognition*, volume I, pages 449–452, The Hague, The Netherlands, September 1992. IEEE Computer Society Press, Volume I.

[Smu93] V. Smutný. Analysis of rainbow range finder errors. In V. Hlaváč and T. Pajdla, editors, *1st Czech Pattern Recognition Workshop*, pages 59–66. Czech Pattern Recognition Society, CTU, Prague, November 1993.

[SW95] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proceedings of the 5th International Conference on Computer Vision*, pages 920–925. IEEE, May 1995.

[TA77] A. N. Tichonov and V. Y. Arsenin. *Solution of ill-posed problems.* Winston and Wiley, Washington, D.C., USA, 1977.

[TK90] S. Tanaka and A. C. Kak. Chapter 2. a rule-based approach to binocular stereopsis. In R. C. Jain and A. K. Jain, editors, *Anaysis and interpretation of range images*, page ?? Springer-Verlag, Berlin, 1990.

[Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accurancy 3D machine vision metrology using off-the-shelf cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323 – 344, August 1987.

[UB91] S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 13(10):992–1005, October 1991.

[Ull96] S. Ullman. *High-level Vision: Object Recognition and Visual Cognition.* A Bradford Book. MIT Press, Cambridge, Ma., USA, 1996.

[Š94] R. Šára. *Local Shading Analysis via Isophotes Properties.* PhD thesis, Johannes Kepler University Linz, Dept. of System Sciences, March 1994.

[Š95] R. Šára. Isophotes: the key to tractable local shading analysis. In *Proceedings CAIP '95*, pages 416–423. Springer Verlag, 1995.

[Wec90] H. Wechsler. *Computational Vision.* Academic Press, London – San Diego, 1990.

[WHH95] T. Werner, R.D. Hersch, and V. Hlaváč. Rendering real-world objects using view interpolation. In *Proceedings of the 5th International Conference on Computer Vision*, pages 957–962, Boston, USA, June 1995. IEEE Press.

[Woo80] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19:139–144, 1980.

[WWS95] D. Weinshall, M. Werman, and A. Shashua. Shape tensors for efficient and learnable indexing. In *Proceedings of the IEEE Workshop Representation of Visual Scenes, June 24, 1995, Cambridge, Ma., USA*, pages 58–65, Los Alamitos, Ca. USA, 1995. IEEE Computer Society Press.

[You94] T.Y. Young, editor. *Hanbook of Pattern Recognition and Image Processing: Computer Vision*, volume 2, San Diego, USA, 1994. Academic Press.