Contents

List of Algorithms

Chapter 9

3D vision, geometry and radiometry

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 underconstrained, 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 – surface 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 is 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 a 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.

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 [Marr 82] 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.

- Aloimonos and Shulman [Aloimonos and Shulman 89] 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.

- Wechsler [Wechsler 90] 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 [Aloimonos 93] 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?) deal 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.

System theory [Klir 91] 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 the (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 (see Chapter 8):

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 [Marr 82], 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 [Flynn and Jain 91, Flynn and Jain 92, Soucy and Laurendeau 92, Bowyer 92].

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 [Brooks et al. 79, Goad 86, Besl and Jain 85, Farshid and Aggarwal 93]. Recognition based on CAD models is of practical importance [Newman et al. 93]. 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 [Biederman 87] – the qualitative features are called *geons*. This mimics the human recognition process in which constituents of a single object (geons) and their spatial arrangement are pointers to a human memory.

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

made first. A linear combination of views has been used [Ullman and Basri 91] 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 [Beymer and Poggio 96]. How this approach can be used for displaying a 3D scene from any viewpoint is considered in [Werner et al. 95, Hlaváč et al. 96].

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.

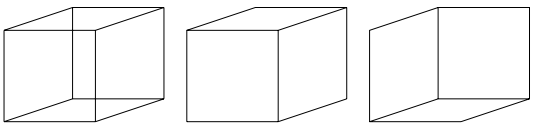


Figure 9.1: *The Necker cube, and two possible interpretations.*

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 **2.5D 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 4.3.4) – informally, a range of blurring filters are passed across the image, after which second-order zero crossings (see Section 4.3.2) are located for each scale of blur [Marr and Hildreth 80]. The blurring recommended is a standard Gaussian filter (see equation (4.51)), while the zero crossings are located with a Laplacian operator (see equation (4.38)). 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 2.5D sketch

The 2.5D 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 preceding 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 2.5D 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 10.1. 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 2.5D 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 [Bertero et al. 88, Aloimonos and Rosenfeld 94] 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** [Tichonov and Arsenin 77, Poggio et al. 85]. 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 modeled (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 global human visual space is viewer-centered and non-Euclidean [Koenderink 90]. 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 **active perception** [Bajcsy 88, Landy et al. 96] and purposive vision [Aloimonos 93]: 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. Controlled eye (or camera) movement is an example, where if there are not enough data to interpret the scene the camera can look at it from another viewpoint. In other words, active vision is intelligent data acquisition controlled by the measured, partially interpreted scene parameters and their errors from the scene. Active vision is an area of much current research.

The active approach can make most ill-posed vision tasks tractable. To provide an overview, we summarize in tabular form [Aloimonos and Rosenfeld 94] how an active observer can change ill-posed tasks to well-posed – see Table 9.1.

Task	*Passive observer*	*Active observer*
Shape from shading	Ill-posed. Regularization helps but a 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 [Ullman 96]. Two recent developments towards new vision theory are:

- *Qualitative vision*, which looks for a qualitative description of objects or scenes [Aloimonos 94]. 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 [Besl and Jain 88].

- The *purposive vision* paradigm, which may help to come up with simpler solutions [Aloimonos 92]. 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. [Howarth 94, Buxton and Howarth 95, Fernyhough 97].

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. Parallel lines in the world do not remain parallel in a perspective image – consider, for example, a view along a railway or into a long corridor. Figure 9.2 illustrates this, where also some commonly used terms are introduced.

We begin with a concise introduction to basic notation and the definitions of projective space [Semple and Kneebone 63, Faugeras 93, Mohr 93]. Consider $(n+1)$ dimensional space without its origin $\mathcal{R}^{n+1} - \{(0,\ldots,0)\}$, and define an equivalence relation

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

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 in 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.

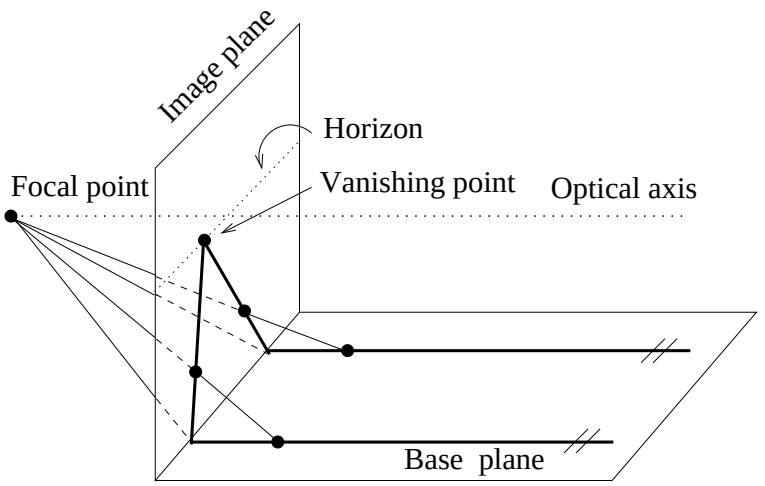


Figure 9.2: *Perspective projection of parallel lines.*

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

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

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 $\alpha \rightarrow 0$. 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 $\mathcal{P}^n \rightarrow \mathcal{P}^n$ 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. Co-lineations 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

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** π 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 C** (also called the **optical center**). The focal length f (sometimes called the principal axis distance [Mohr 93]) is a parameter of the lens.

The projection is performed by an optical ray (also a light beam) reflected from a scene point **X** (top left in Figure 9.3) or originated from a light source. The optical ray passes through the optical center **C** and hits the image plane at the point **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 **X**, **U** are expressed in the world co-ordinate system.

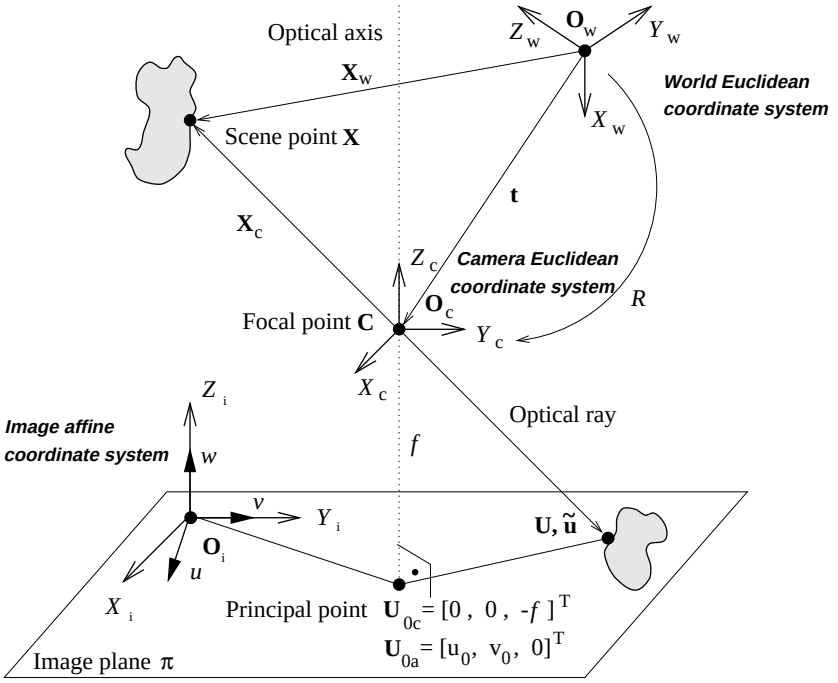


Figure 9.3: *The geometry of a linear perspective camera.*

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 a 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, Y_i, but the axis u may have a different orientation to the axis X_i.

   The reason for introducing these co-ordinates is the fact that in general pixels need not be 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 3×1 vector. To express the same point in the camera Euclidean co-ordinate system, i.e. $\mathbf{X}_c$, we have to

translate it by subtracting vector $\mathbf{t}$ and rotate it as specified by the matrix R.

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

The point $\mathbf{X}_c$ is projected to the image plane π as point $\mathbf{U}_c$. The x and y co-ordinates of

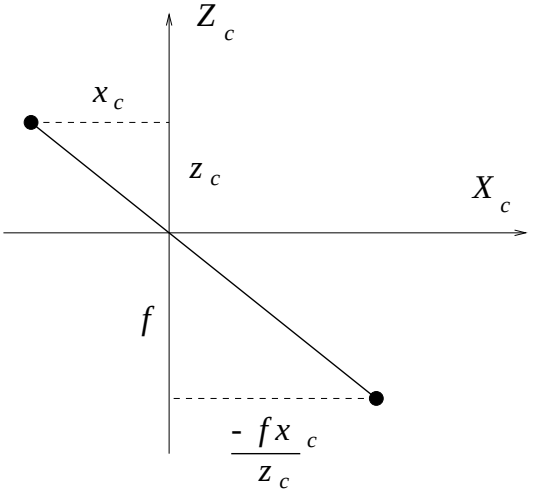


Figure 9.4: *Calculation of the co-ordinates of the projected point.*

the projected point can be derived from the similar triangles illustrated in Figure 9.4

$$\mathbf{U}_c = \begin{bmatrix} \frac{-fx_c}{z_c}, & \frac{-fy_c}{z_c}, & -f \end{bmatrix}^T \tag{9.4}$$

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 π. The principal point $\mathbf{U}_0$ is expressed in the image affine co-ordinate system as $\mathbf{U}_{0a} = [u_0, v_0, 0]^T$.

The projected point can be represented in the 2D image plane π 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 3×3 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.

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

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, and can rewrite

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

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

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

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: α_u represents scaling in the u axis, measuring f in pixels along the u axis, and α_v similarly specifies f in pixels along the v-axis. α_{shear} measures in pixels the degree of slant of the co-ordinate axes in the camera image plane, giving in the v-axis direction how far the focal length f coincident with u-axis is 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 3×4 matrix. The leftmost 3×3 submatrix describes a rotation and the rightmost

column a translation The delimiter | denotes that the matrix is composed of two submatrices.

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

where $\tilde{\mathbf{X}}$ is the 3D scene point in homogeneous co-ordinates. The matrix M is called the **projective matrix** (or 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 3×3 submatrix of the projective matrix M consisting of the 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 and thus all αM 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* [Faugeras 93], 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.

We first consider 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 9.5[1]). Each correspondence between a 3D scene and 2D image point provides one equation

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

[1]Here and in some further figures the image plane is positioned in front of the focal point – this differs from earlier figures where the image plane was behind the focal point. Such a presentation makes figures easier to comprehend and should not cause any confusion.

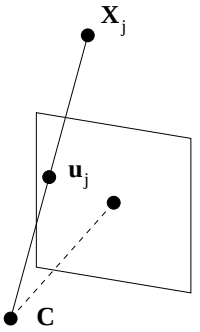


Figure 9.5: *Camera calibration from a known scene. A minimum of six corresponding pairs of scene points* $\mathbf{X}_j$ *and image points* $\mathbf{u}_j$ *are needed to calibrate the camera.*

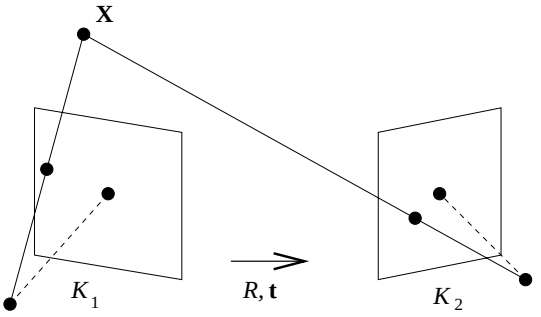


Figure 9.6: *Camera calibration from an unknown scene. At least two views are needed. It is assumed that the intrinsic parameters of the camera do not change, so* $K_1 = K_2$.

The solution [Faugeras 93] 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 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 [Horaud et al. 95].

      ii. *Pure rotation:* If camera motion is restricted to pure rotation, the solution is given by [Hartley 94].

      iii. *Pure translation:* The linear solution (pure translation) is due to [Pajdla and Hlaváč 95].

   (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 nonlinear [Maybank and Faugeras 92]. Calibration from an unknown scene is still considered numerically hard, and will not be considered here (although see, for example, [Butterfield 97] for a consideration of this problem).

9.2.4 Calibration of one camera from a 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 3×4 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 [Faugeras 93]. 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;

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

The 3×3 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 [Press et al. 92, Golub and Loan 89] will decompose A into a product of two such matrices, and hence recover K and R.

Alternatively, we can use **Singular Value Decomposition (SVD)**[2]. 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 a pinhole camera does, but this is not the case with real lenses. A typical lens performs distortion of several pixels which a human observer does not notice looking at a general scene. However, when an image is used for measurements, compensation for the distortion is necessary.

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

Recall the five intrinsic camera parameters introduced in equation (9.8). Here, we shall replace the focal length f of the lens by a parameter called the **camera constant**. Ideally, this is equal to the focal length, but in reality this is true only when the lens is focused at infinity; otherwise, the camera constant is slightly less than the focal length. Similarly, the co-ordinates of the principal point can change slightly from the ideal intersection of the optical axis with the image plane.

The idea behind calibration of intrinsic parameters is to observe a known calibration image with some regular pattern, for example blobs or lines covering the whole image. Distortions observed in the pattern allow estimation of the parameters.

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

$$\tilde{u} \quad = \quad x - \hat{u}_0$$

[2]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 [Press et al. 92], and a rigorous mathematical treatment is given in [Golub and Loan 89]. Most software packages for numerical calculations such as MATLAB (trade mark of MathWorks Inc.) 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, D 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 in 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).

$$\tilde{v} = y - \hat{v}_0 \tag{9.18}$$

The correct image co-ordinates u, v are obtained if compensations for errors δu, δv are added to the measured uncorrected image co-ordinates $\tilde{u}$, $\tilde{v}$.

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

Compensations for errors are often modeled as even power polynomials to secure rotational

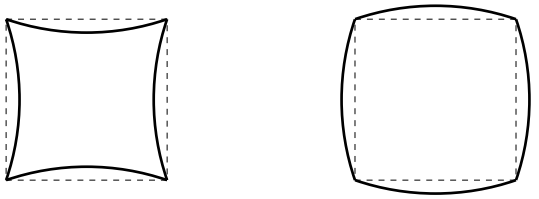


Figure 9.7: *Radial distortion of an off-the-shelf lens.*

symmetry. Typically, polynomial degrees up to 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. r^2 is the square of the radial distance from the center 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). 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;

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

We can visualize typical lens radial distortion for the simple second order model as a special case of equation (9.20), i.e. no decentering is assumed and a second order 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, and the distorted images are shown in Figure 9.7. On the left is pincushion like distortion (a minus sign in equation (9.23)), and the the right part depicts barrel like distortion corresponding to a plus sign.

There are more complicated lens models that cover tangential distortions that model such effects as lens decentering [Jain et al. 95] which we shall not describe in detail here. The reader can consult the original paper [Tsai 87] or the treatment in [Jain et al. 95]. An alternative procedure was proposed in [Prescott and McLean 97].

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.

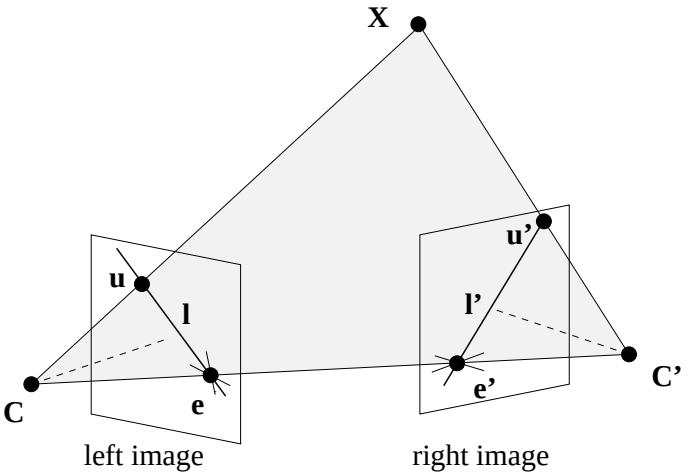


Figure 9.8: *Epipolar geometry in stereopsis.*

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.

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 positions of the point $\mathbf{X}$ for 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.

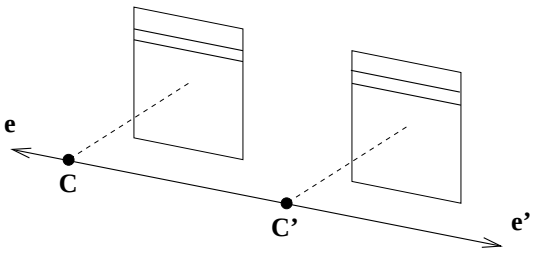


Figure 9.9: *The canonical stereo configuration where the epipolar lines are parallel in the image, and epipoles move to infinity.*

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 (rasters) 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 [Mohr 93] 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.

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

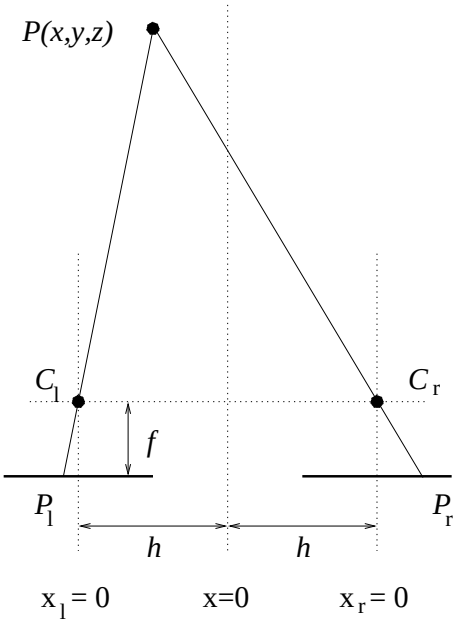


Figure 9.10: *Elementary stereo geometry in canonical configuration.*

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

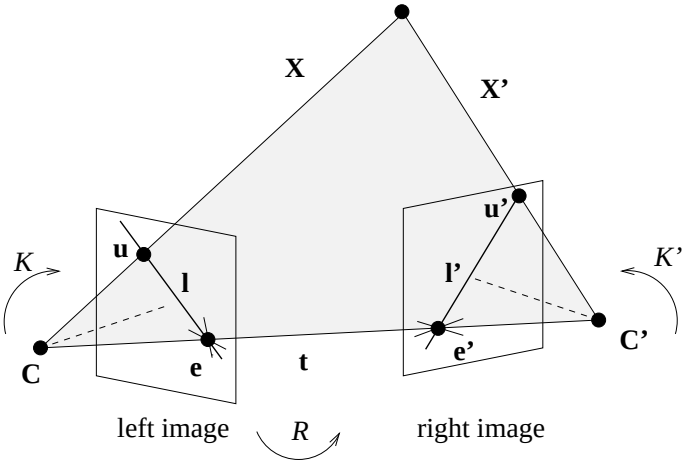


Figure 9.11: *Stereo with nonparallel axes.*

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 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

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

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

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

This equation (9.30) is homogeneous with respect to $\mathbf{t}$, so the scale is not determined. Absolute scale cannot be recovered if a '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[3] matrix $S(\mathbf{t})$ can be created from it if $\mathbf{t} \neq \mathbf{0}$.

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

Recall that rank(S) is a number of the linearly independent lines in matrix S. Note that 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 [Longuet-Higgins 81] of a 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; we denote the normalized measurements $\breve{\mathbf{u}}$, $\breve{\mathbf{u}}'$. The camera calibration matrices give the relations

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

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

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

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

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.

We summarize important *properties of the essential matrix.*

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

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

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 [Hartley 92]. Equation (9.37) shows that the essential matrix is a product of the matrices $S(\mathbf{t})$ and 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 $D = diag[k, k, 0]$ (where $diag[x, y, \ldots]$ describes a diagonal matrix, with diagonal $x, y, \ldots$). 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

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

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

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

We consider now the *properties of the fundamental matrix.*

- We have seen 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.

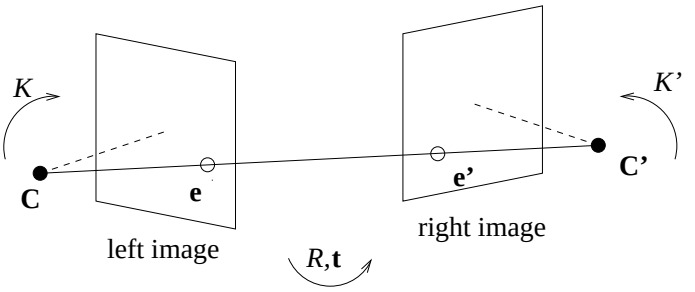


Figure 9.12: *Epipoles* $\mathbf{e}, \mathbf{e}'$ *and the fundamental matrix* F.

- Consider two epipoles $\mathbf{e}, \mathbf{e}'$, depicted in Figure 9.12. Then

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

- 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

Epipolar geometry has seven degrees of freedom [Mohr 93]: The epipoles $\mathbf{e}$, $\mathbf{e}'$ in the image have two co-ordinates each (giving 4 dof), while another three come from the mapping of any three epipolar lines in the first image to the second. Thus the correspondence of seven points in left and right images enables the establishment of the fundamental matrix F using a nonlinear algorithm [Faugeras et al. 92]. Unfortunately this computation is numerically unstable.

If there are eight non-coplanar corresponding points available, a linear method called the **eight point algorithm** can be used and if more points are at hand the estimation might be robust to noise and mismatches. The method was originally proposed by Longuet-Higgins [Longuet-Higgins 81] for essential matrix estimation.

The eight point algorithm was supposed to be numerically unstable, but this is not the case if normalization (i.e. translation and scaling) of values is performed first [Hartley 95, Butterfield 97]. The algorithm is easy to implement and is fast; proper normalization is needed in most 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 co-ordinates can be written $\mathbf{u}^T = [u_i, v_i, 1]$. The 3×3 fundamental matrix F has only eight unknowns as it is only known up to scale; eight correspondences will generate eight matrix equations;

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

Rewriting the elements of the fundamental matrix as a column vector with nine elements $\mathbf{f}^T = [f_{11}, f_{12}, \ldots, f_{33}]$, 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 data from real image measurements, an overdetermined system of linear equations is obtained, and a least-squares solution to this set is sought. [Hartley 95]. The vector $\mathbf{f}$ is determined that minimizes the Frobenius norm $\|A\,\mathbf{f}\|$ fulfilling the constraint $||f|| = 1$. Principal component analysis gives the solution, and $\mathbf{f}$ is the unit eigenvector of $A^T A$ corresponding to the smallest eigenvalue of A. An appropriate algorithm for achieving this is SVD. Note that another numerically plausible solution to the overdetermined systems of linear equations (9.49) is given in [Faugeras 93].

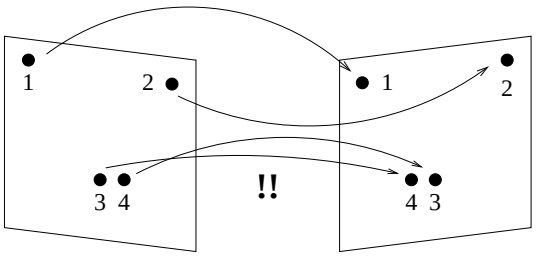


Figure 9.13: *Problem with mismatches in stereo correspondence.*

Estimation of the fundamental matrix can be corrupted by gross errors caused by **mismatches** in stereo correspondence, illustrated in Figure 9.13. The obvious solution to the problem is to attempt to drop out erroneous matches. One approach uses the least median of squares method for robust estimation instead of standard least squares;

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

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

We have already seen that the fundamental matrix F should have rank 2, but a solution of equation (9.49) will not in general give such a matrix. F should be replaced by the matrix $\hat{F}$ that minimizes the Frobenius norm of $\|F - \hat{F}\|$ fulfilling the condition $\text{rank}(A) = 2$. SVD decomposes as $F{=}UDV^T$, $D{=}\text{diag}[r, s, t]$, $r \geq s \geq t$, and the solution we seek is $\hat{F}{=}U\,diag[r, s, 0]\,V^T$.

9.2.9 Applications of epipolar geometry in vision

Image rectification to ease the search for correspondences

We have seen that stereo geometry implies that corresponding points can be sought in 1D space along epipolar lines. In general, epipolar lines in the left image are not parallel to epipo-

lar lines in the right image (non-parallel optical axes). Parallel epipolar lines are preferred, as they ease the search for correspondence, whether by computer or human eye. It is always possible to apply image rectification to images captured by a stereo rig with non-parallel optical axes; this results in a new set of images with parallel epipolar lines that are typically horizontal.

Image rectification recalculates pixel co-ordinates using a linear transformation in projective space. This is illustrated in Figure 9.14, where C, C' are optical centers. Image planes with dashed borders show input before rectification, and 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,

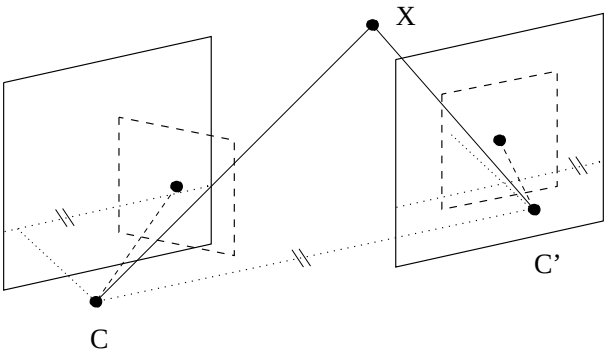


Figure 9.14: *Image rectification to get parallel epipolar lines.*

$\mathbf{u}^T F \mathbf{u}' = 0$. We seek the two 3×3 transformation matrices A, B that rectify co-ordinates (denoted with $\breve{}$) of points in left and right images respectively, $\breve{\mathbf{u}} = A\mathbf{u}$, $\breve{\mathbf{u}}' = B\mathbf{u}'$. The fundamental matrix of the rectified images $\breve{F} = (A^{-1})^T F B^{-1}$ should correspond to epipoles that moved along horizontal axes to $-\infty$ or ∞ for the left and right images, respectively.

To complete this task we need to set values in the transformation matrices A and B – a solution is given in [Ayache and Hansen 88] which we summarize here. Since the transformation is constrained the number of unknowns is reduced; image co-ordinate transformations using matrices A, B should not change the position of optical centers, but should align two distinct image planes into one common image plane that is parallel to the 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 co-ordinate, as this simplifies calculations.

Image co-ordinates after rectification are

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

Recall the projection matrix from equation (9.12). We have two such matrices M, M' for left and right images before rectification. The 3×3 submatrix on the left side of M is composed of three rows (1×3 vectors) that we denote as $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$, and 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 co-ordinates of the

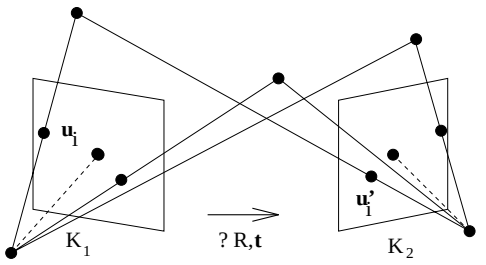


Figure 9.15: *Ego-motion estimation.*

optical centers. The transformation matrices that perform rectification are then;

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

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

This procedure is computationally inexpensive. Only two 3×3 transformation matrices need be stored, and only 6 multiplications, 6 additions and 2 divisions are needed per rectified pixel. Notice that the rectification 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 procedure can be easily generalized to three and more images [Ayache and Hansen 88].

Ego-motion estimation from calibrated camera measurements

Camera **ego-motion** estimation of a calibrated camera considers the case of unknown movement of the camera, where rotation R and translation $\mathbf{t}$ need to be learned from point correspondences between two images.

Suppose a point $\mathbf{u}_i$ from the first image corresponds to the point $\mathbf{u}'_i$. The following algorithm [Hartley 92] allows the computation of an unknown rotation R and 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 will be used to estimate a fundamental matrix.

2. The data should be normalized – this helps to minimize numerical errors.

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

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

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

3. Compute an estimate of the fundamental matrix $\hat{F}$ using the linear algorithm given in Section 9.2.8. Numerical inaccuracies may cause the estimate not to have the property that after SVD, $D = diag(k,k,0)$.

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. Determine a rotation R and translation $\mathbf{t}$ from the estimated essential matrix $\hat{E}$ using 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 should have two equal singular values, and the third must be zero. We can adjust singular values by zeroing t and averaging r and s
$$E = u \begin{bmatrix} \frac{r+s}{2} & 0 & 0 \\ 0 & \frac{r+s}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \tag{9.58}$$
This matrix E can be decomposed into rotation R and translation $\mathbf{t}$ in 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
$$R = UGV^T \text{ or } UG^TV^T, \qquad S(\mathbf{t}) = VZ\,V^T \tag{9.59}$$

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

3D similarity reconstruction from two cameras with known intrinsic calibration

3D similarity reconstruction aims to measure 3D co-ordinates of a scene point $\mathbf{X}$ from two image measurements $\mathbf{u}$ and $\mathbf{u}'$ (see Figure 9.16). We assume that the cameras are calibrated; that is, their intrinsic calibration parameters are known and are available as calibration matrices K and K'. The extrinsic parameters are unknown. This case differs from standard stereo (full 3D Euclidean reconstruction) where the relative position of the cameras is known. Common sense suggests that less will be measured in an unknown scene compared to the standard stereo case; the reconstruction of the unknown scene is achieved up to a similarity.

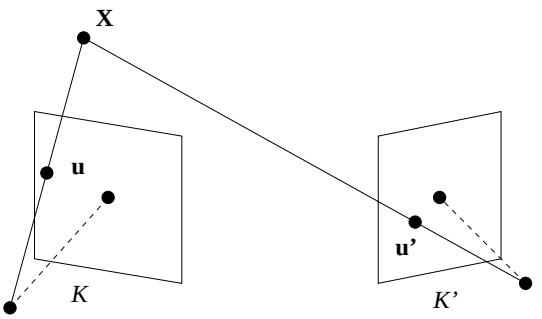


Figure 9.16: *3D similarity reconstruction from two cameras.*

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 the essential matrix E.
3. Obtain the rotation R and translation $\mathbf{t}$ from the essential matrix E.
4. Solve equations (9.60) to get $\mathbf{X}$.

Notice that $\mathbf{X}$ is found up to scale only, meaning that we do not get a Euclidean reconstruction but a similarity reconstruction. A full Euclidean reconstruction (as in stereo vision) is unavailable because the distance between the cameras is unknown in this case.

3D projective reconstruction from two uncalibrated cameras

We now consider the most general 3D reconstruction case when the point correspondence in two uncalibrated cameras can be established, meaning that both intrinsic and extrinsic camera calibration parameters are unknown. We shall see that a 3D projective reconstruction can be obtained, which is practically appealing 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 camera position is unknown, and a zoom lens may be used, and we do not know the actual focal length.

The perspective projection performed by the first camera is expressed using the projective matrix M (recall equation (9.9)), which 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:

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

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

To eliminate the unknown scale factor, consider the ratio between the three rows in the projection matrix M [Faugeras and Mourrain 95].

$$\begin{aligned} u : v : w &= \mathbf{m}_1^T\mathbf{X} : \mathbf{m}_2^T\mathbf{X} : \mathbf{m}_3^T\mathbf{X} \\ u' : v' : w' &= \mathbf{m}'^T_1\mathbf{X} : \mathbf{m}'^T_2\mathbf{X} : \mathbf{m}'^T_3\mathbf{X} \end{aligned} \tag{9.63}$$

Thus three equations hold for both the first and the second camera;

$$\begin{aligned} u\mathbf{m}_2^T\mathbf{X} &= v\mathbf{m}_1^T\mathbf{X} & u'\mathbf{m}'^T_2\mathbf{X} &= v'\mathbf{m}'^T_1\mathbf{X} \\ u\mathbf{m}_3^T\mathbf{X} &= w\mathbf{m}_1^T\mathbf{X} & u'\mathbf{m}'^T_3\mathbf{X} &= w'\mathbf{m}'^T_1\mathbf{X} \\ v\mathbf{m}_3^T\mathbf{X} &= w\mathbf{m}_2^T\mathbf{X} & v'\mathbf{m}'^T_3\mathbf{X} &= w'\mathbf{m}'^T_2\mathbf{X} \end{aligned} \tag{9.64}$$

Equations (9.64) can be written in a matrix form. We present this for the first camera only; a 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 is multiplied by w and second row by $-v$ and added we get

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

Extracting 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 see that equations (9.66) and (9.67) are linearly dependent, and the same reasoning holds for measurements from the second image. Since only two equations are linearly independent, we use the second and the third equations.

$$\begin{aligned} (u\mathbf{m}_3^T - w\mathbf{m}_1^T)\mathbf{X} &= 0 & (u'\mathbf{m}'^T_3 - w'\mathbf{m}'^T_1)\mathbf{X} &= 0 \\ (v\mathbf{m}_3^T - w\mathbf{m}_2^T)\mathbf{X} &= 0 & (v'\mathbf{m}'^T_3 - w'\mathbf{m}'^T_2)\mathbf{X} &= 0 \end{aligned} \tag{9.68}$$

This can be rewritten in matrix form;

$$\begin{bmatrix} u\mathbf{m}_3^T - w\mathbf{m}_1^T \\ v\mathbf{m}_3^T - w\mathbf{m}_2^T \\ u'\mathbf{m}'^T_3 - w'\mathbf{m}'^T_1 \\ v'\mathbf{m}'^T_3 - w'\mathbf{m}'^T_2 \end{bmatrix} \mathbf{X} = A\,\mathbf{X} = 0 \tag{9.69}$$

The matrix A has dimension 4×4 and $\mathbf{X}$ is a 4×1 vector.

We are interested in a nontrivial solution of equation (9.69), and therefore consider the case $det(A) = 0$. This 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 to consider in reconstructing 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 considered in Section 9.2.5. The current 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 easily be replaced by an inverse mapping.

2. *Scene reconstruction with uncalibrated cameras.*
   If the calibration of a stereo rig is unknown it can be shown that the reconstructed co-ordinates $\tilde{\mathbf{X}}$ differ from the correct Euclidean reconstruction by some (unknown) projective transformation H.
   $$\tilde{\mathbf{X}} = H\,\mathbf{X} \tag{9.70}$$
   H is a regular 4×4 matrix. The transformation H ranges from Euclidean through affine to the general projective case according to how much calibration knowledge is at hand. H is the same for all scene points for one position and calibration of the camera. Of course, the same algorithm with a different scene gives a 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 [Faugeras 93, Faugeras and Mourrain 95] for more details.

9.2.10 Three and more cameras

In this section we will consider the case of three or more cameras observing the same scene, assuming mutually corresponding points can be found in all views. We have already seen that views of two cameras are described using a bilinear relation expressed by the fundamental matrix, and it is natural to ask what more can be learned if three or more views are available.

Three cameras looking at the same point are sketched in Figure 9.17. The relations 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''. using a similar approach to that given when computing 3D projective reconstruction from two uncalibrated cameras, we aim to obtain a set of linear equations that relate image measurements to their 3D counterparts.

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

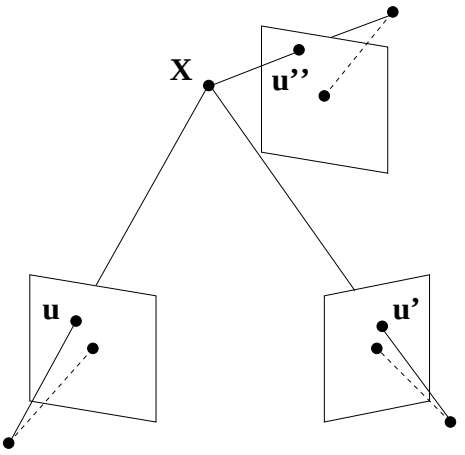


Figure 9.17: *Geometry of three cameras,* $\mathbf{u} \simeq M\,\mathbf{x}$, $\mathbf{u}' \simeq M'\,\mathbf{x}$, $\mathbf{u}'' \simeq M''\,\mathbf{x}$.

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

Using similar manipulations to equations (9.62) – (9.69) eliminates unknown scale factors and provides the desired relation in matrix form. We will assign labels `1:` to `6:` in the following for convenience of future reference;

$$\begin{matrix} \texttt{1:} \\ \texttt{2:} \\ \texttt{3:} \\ \texttt{4:} \\ \texttt{5:} \\ \texttt{6:} \end{matrix} \begin{bmatrix} u\mathbf{m}^T_3 - w\mathbf{m}^T_1 \\ v\mathbf{m}^T_3 - w\mathbf{m}^T_2 \\ u'\mathbf{m}'^T_3 - w'\mathbf{m}'^T_1 \\ v'\mathbf{m}'^T_3 - w'\mathbf{m}'^T_2 \\ u''\mathbf{m}''^T_3 - w''\mathbf{m}''^T_1 \\ v''\mathbf{m}''^T_3 - w''\mathbf{m}''^T_2 \end{bmatrix} \mathbf{X} = A\,\mathbf{X} = 0 \tag{9.75}$$

We shall follow (but simplify) an explanation given in [Faugeras and Mourrain 95], and shall use the reference numbers of equation (9.75). We are interested in the nontrivial solution to this equation, meaning that the matrix A should have rank 3. This means that the determinant of all its 4×4 submatrices must be zero; there are $\mathrm{C}^6_4 = \frac{6!}{4!\,2!} = 15$ such submatrices. Consider these 15 quadruples of equations and classify then according to whether they involve two or three cameras.

Three sets of equations express a 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 squares of the same variable do not appear in these bilinear equations. Consider now sets of equations that express a trilinear relation among images of the same point as seen by three cameras. From the 12 trilinearities only four are linearly independent; three possibilities for linearly independent quadruples of equations are the following – (notice that there are always two rows of equation (9.75) corresponding to one camera, with one row

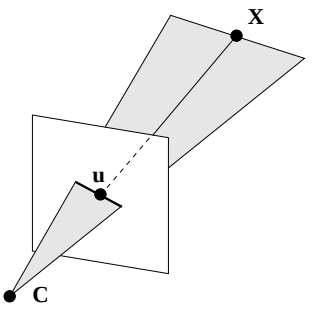


Figure 9.18: *Each of six rows defines a plane passing through the optical center* **C** *and the point* **X**.

for each of the remaining two cameras):

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

A geometric interpretation of the rows of equation (9.75) assists 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 Figure 9.18.

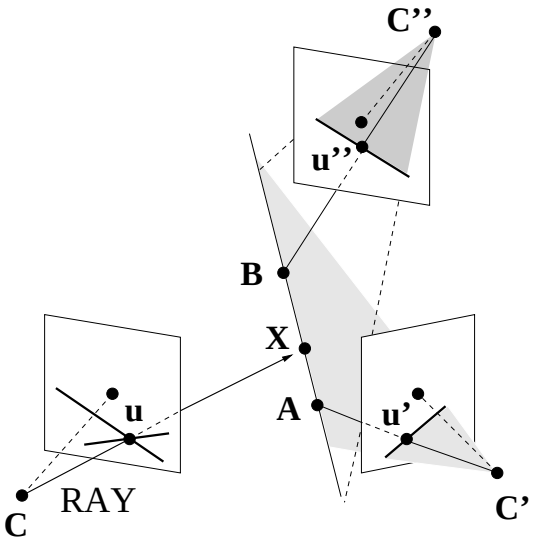


Figure 9.19: *Illustration that one trilinear relation does not assure that three measured points are coincident as expected.*

Notice that one trilinearity relation does not ensure that observed points $\mathbf{u}$, $\mathbf{u}'$, and $\mathbf{u}''$ correspond to only one scene point $\mathbf{X}$. Only one of the views plays a role of the ray; this is illustrated in Figure 9.19. Two rows of equation (9.75) corresponded to the measurement $\mathbf{u}$ taken by the first camera, and the corresponding ray points to $\mathbf{X}$. The other two views constrain the point position in the space to a plane only. One trilinearity relation ensures

that the ray and two planes have a common point in the projective space $\mathcal{P}^3$. In other words, $\mathbf{X}$, $\mathbf{A}$, and $\mathbf{B}$ are colinear but need not be coincident.

Consider now what happens if we had four cameras. In equation (9.75) we would have two more equations. Now we can consider 4×4 subdeterminants which contain one row arising from one camera. This is called a **quadrilinear constraint** which is a polynomial of degree four in the co-ordinates of the points $\mathbf{m}_i$ and linear in the co-ordinates 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 a linear combination of bilinear and trilinear constraints. This means that the *fourth view does not contribute 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 factored into relations of fewer cameras [Weinshall et al. 95].

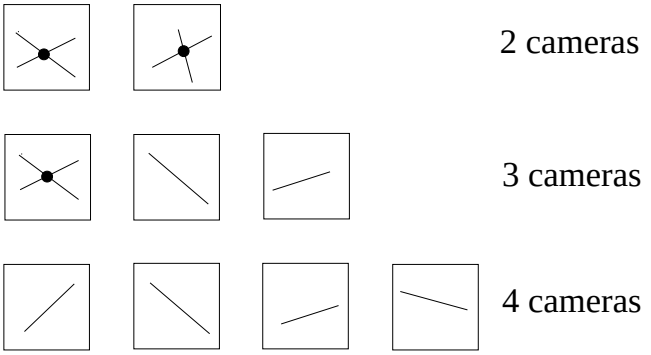


Figure 9.20: *Geometric interpretation of bilinear, trilinear, and quadrilinear constraint.*

The case of two, three, and four cameras is illustrated in Figure 9.20. The upper row shows the case of the bilinear constraint (given by the fundamental matrix F) that relates corresponding points in two images. The middle row illustrates the trilinearity constraint where correspondence between one point and two lines is established. The bottom row shows the quadrilinear constraint where correspondence of four lines is taken into account.

Given this intuitive geometric understanding of the trilinear relation among views, we shall proceed to an algebraic derivation as well. Assume that the first camera is in a canonical configuration, i.e. its projective matrix is in the simplest form;

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

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 ρ is to be determined. Project the scene point $\mathbf{X}$ into the second camera.

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

$$u'_i \simeq a^k_i u_k + a_{i4}\rho\,, \quad k = 1\ldots 3 \tag{9.80}$$

Here we have adopted Einstein's convention of omitting the summation symbol for compactness of representation. Thus $a^k_i u_k$ would originally be written as $\sum_{k=1}^{k=3} a_{ik} u_k$.

The scale factor ρ need to be eliminated. We get three equations of which two are independent;

$$u'_i\,(a^k_j\,u_k + a_{j4}\rho) = u'_j\,(a^k_i\,u_k + a_{i4}\rho) \tag{9.81}$$

This yields three estimates of ρ

$$\rho = \frac{u_k(u'_i\,a^k_j - u'_j\,a^k_i)}{u'_j\,a_{i4} - u'_i\,a_{j4}} \tag{9.82}$$

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

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{u} \\ \frac{u_k(u'_i\,a^k_j - u'_j\,a^k_i)}{u'_j\,a_{i4} - u'_i\,a_{j4}} \end{bmatrix} \simeq \begin{bmatrix} (u'_j\,a_{i4} - u'_i\,a_{j4})\,\mathbf{u} \\ u_k(u'_i a^k_j - u'_j\,a^k_i) \end{bmatrix} \tag{9.83}$$

Now $\tilde{\mathbf{X}}$ is projected by the third camera;

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

Notice that

$$\begin{aligned} u''_l &\simeq b^k_l u_k(u'_j a_{i4} - u'_i a_{j4}) + b_{l4}u_k(u'_i a^k_j - u'_j a^k_i) \simeq \\ &\simeq u_k u_i(a^k_j b_{l4} - a_{j4}b^k_l) - u_k u'_j(a^k_i b_{l4} - a_{i4}b^k_l) \simeq \\ &\simeq u_k(u'_i T_{kjl} - u'_j T_{kil}) \end{aligned} \tag{9.85}$$

T_{ijk}, $i,j,k = 1,2,3$ is an algebraic entity called a tensor that depends on three indices. This can be imagined as a 'three-dimensional matrix', i.e. a $3\times3\times3$ cube consisting of 27 numbers.

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

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

This 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. 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 can have values 1 or 2 we have four linearly independent equations.

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

The use of the trilinear constraint yields three practical advantages [Shashua and Werman 95].

1. The trilinear tensor can be recovered linearly from 7 corresponding points in three views, while the fundamental matrix calculated from a pair of views needs at least 8 points for linear solution. Practically, an overdetermined system of equations 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 the trilinear tensor is **epipolar transfer**. Assuming that the trilinear tensor has been estimated, if two images are known any third image can be computed using equation (9.87).

The other application of the 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 an approach to shape under perspective projection with uncalibrated cameras [Weinshall et al. 95].

9.2.11 Stereo correspondence algorithms

We have seen in Section 9.2.6 that much can be learned about the geometry of a 3D scene if it is known which point from one image corresponds to a point in a second image. The solution of this **correspondence problem** is a key step in any photogrammetric, stereo vision or motion analysis task. Here we describe how the same point can be found in two images if the same scene is observed from two different viewpoints. Of course, it is assumed that two images overlap and thus the corresponding points are sought in this overlapping area.

In image analysis, some methods are based on the assumption that images constitute a linear (vector) space (e.g. eigen-images or linear interpolation in images [Werner et al. 95, Ullman and Basri 91]); this linearity assumption[4] is not valid for images in the general [Beymer and Poggio 96], but some authors have overlooked this fact. The structure of a vector space assumes that the i^{th} component of one vector must refer to the i^{th} component of another; this assumes that the correspondence problem has been solved.

Automatic solution of the correspondence problem is an evergreen computer vision topic, and the pessimistic conclusion is that it is not soluble in the 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 sought in left and right images of the flat object there are not any features that could distinguish them. Another unavoidable difficulty in searching for corresponding points is the **self-occlusion** problem which 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 Figure 9.21).

Fortunately, uniform intensity and self-occlusion are rare, or at least uncommon, in scenes of practical interest. Establishing correspondence between projections of the same point in different views is based on finding image characteristics that are similar in both views, and the local similarity is calculated.

The inherent ambiguity of the *correspondence problem* can in practical cases be reduced using several **constraints**. Some of these follow from the geometry of the image capturing

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

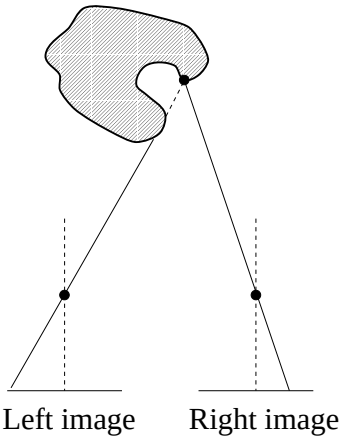


Figure 9.21: *Self-occlusion makes search for some corresponding points impossible.*

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. There follows a list of constraints commonly used [Klette et al. 96] to provide insight into the correspondence problem.

The first group of constraints depends mainly on the geometry and the photometry of the image capturing process. These are:

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

Uniqueness constraint: This states that, in most cases, a pixel from the first image can correspond to only one pixel in the second image. The exception arises when two or more points lie on one ray coming from the first camera and can be seen as separate points from the second. This case, which arises in the same way as self-occlusion, is illustrated in Figure 9.22.

Photometric compatibility constraint: This states that intensities of a point in the first and the second image are likely to differ only a little. They are unlikely to be exactly the same due to the mutual angle between the light source, surface normal and the viewer differing, but the difference will typically be small as the views will not differ much. Practically, this constraint is very natural 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 transformations.

Geometric similarity constraints: These build on the observation that geometric characteristics of the features found in the first and second images 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.

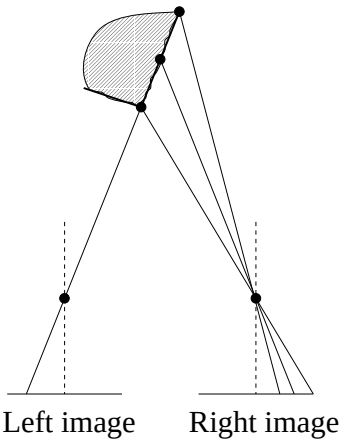


Figure 9.22: *Exception from the uniqueness constraint.*

Disparity smoothness constraint: This claims that disparity changes slowly almost everywhere in the image. Assume two scene points $\mathbf{p}$ and $\mathbf{q}$ are close to each other, and denote the projection of $\mathbf{p}$ into the left image as $\mathbf{p}_L$ and into the right image as $\mathbf{p}_R$, and $\mathbf{q}$ similarly. If we assume that the correspondence between $\mathbf{p}_L$ and $\mathbf{p}_R$ has been established, then the quantity

$$|\,(\,|\mathbf{p}_L - \mathbf{p}_R| - |\mathbf{q}_L - \mathbf{q}_R|\,)\,|$$

(the absolute disparity difference) should be small.

Figural disparity constraint: This says that corresponding points should lie on an edge element in both right and left images, as well as fulfilling the disparity smoothness constraint.

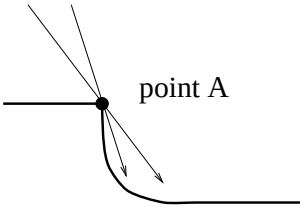


Figure 9.23: *Self-occlusion due to abrupt surface discontinuity can be detected.*

Feature compatibility constraint: This place a restriction on possible matches on the physical origin of matched points. Points can match only if they have the same physical origin – for example, object surface discontinuity, border of a shadow cast by some objects, occluding boundary, specularity boundary, etc. Notice that edges in an image caused by specularity or self-occlusion cannot be used to solve the correspondence problem as they move with changing viewpoint. On the other hand, self-occlusion caused by abrupt discontinuity of the surface can be identified – see Figure 9.23.

Disparity limit constraint: This originates from psycho-physical experiments in which it is demonstrated that the human vision system can only fuse stereo images if the disparity is smaller than some limit. This constrains the lengths of the search in artificial methods that seek correspondence.

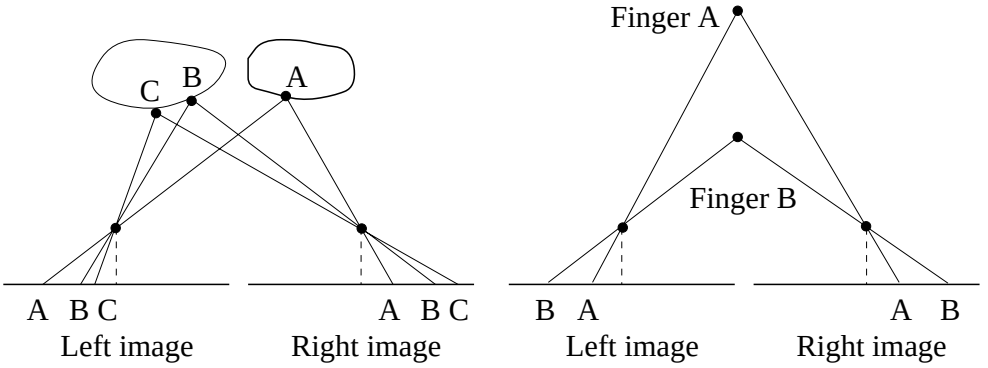


Figure 9.24: *(a) Corresponding points lie in the same order on epipolar lines; (b)This rule does not hold if there is a big discontinuity in depths.*

Ordering constraint: This says that for surfaces of similar depth, corresponding feature points typically lie in the same order on the epipolar line (see Figure 9.24(a)). If there is a narrow object much closer to the camera than its background, the order can be changed (see Figure 9.24(b)). It is easy to demonstrate violation of this ordering constraint; hold two forefingers vertically, almost aligned but at different depth in front of your eyes. Closing the left eye and then the right eyes interchanges the left/right order of the fingers.

Mutual correspondence constraint: This helps to rule out points that do not have a corresponding counterpart due to occlusion, highlight or noise. Assume the search started from the left image point $\mathbf{p}_L$ and a corresponding $\mathbf{p}_R$ was found. If the task is reversed, and a search starting from the point $\mathbf{p}_R$ fails to find the point $\mathbf{p}_L$, then the match is not reliable and should be ruled out.

All these constraints have been of use in one or more existing stereo correspondence algorithms. We present here a taxonomy of such algorithms; from the historical point of view, 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.

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

Psychological experiments with **random dot stereograms** performed by Julesz [Julesz 90] generated a new view; these experiments show that humans do not need to create monocular features before binocular depth perception can take place. A random dot stereogram is

created in the following way: A left image is entirely random, and the right image is created from it in a consistent way such 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. Such 'random dot stereograms' have been widely published under the name '3D images' in many popular magazines.

Recent developments in this area use a combination of both low-level and the high-level stereo correspondence methods [Tanaka and Kak 90].

Correlation-based stereo correspondence

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 many potential candidates with similar intensity, and thus intensities of several neighboring pixels are considered. Typically, a 5×5 or 7×7 or 3×9 window may be used. These methods are sometimes called **area-based stereo**.

We shall illustrate the approach with a simple algorithm called **block-matching** [Klette et al. 96]. Assuming the canonical stereo setup with parallel optical axes of both cameras, the basic idea of the algorithm is that all pixels in the window (called a block) have the same disparity, meaning that one and only one disparity is computed for each block. One of the images, say the left, is tiled into blocks, and a search for correspondence in the right image is conducted for each of these blocks in the right image. The measure of similarity between blocks can be, e.g. the mean square error of the intensity, and the disparity is accepted for the position where the mean square error is minimal. Maximal change of position is limited by the disparity limit constraint. The mean square error can have more than one minimum and in this case an additional constraint is used to cope with ambiguity.

The result of the block matching algorithm is a sparse matrix of disparities, where disparity is calculated only for a representative point of the block; various methods allow us to refine the result to a dense disparity matrix. Block-matching algorithms are typically slow, and regular pyramid implementations are often used to speed up the process.

Another relevant approach is that of Nishihara [Nishihara 84], who observes that an algorithm attempting to correlate individual pixels (by, e.g. matching zero crossings [Marr and Poggio 79]) 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 4.3.4), 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 modeled 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. Characteristically, these are pixels on edges, lines, corners, etc., and correspondence is sought according to properties of such features as, e.g. orientation along edges, or lengths of line segments. The advantages of feature-based methods over intensity-based correlation are:

- Feature-based methods are less ambiguous since the number of potential candidates for correspondence is smaller.
- The resulting correspondence is less dependent on photometric variations in images.
- Disparities can be computed with higher precision; features can be sought in the image to subpixel precision.

We shall present one example of a feature-based correspondence method – the **PMF algorithm**, named after its inventors [Pollard et al. 85]. 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 first two constraints are not peculiar to this algorithm (for example, they are also used by Marr [Marr and Poggio 79]) – 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 9.25) 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;

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

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

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

$$S(A,B) = \sqrt{\left(\left(\frac{a_{xl} + a_{xr}}{2}\right) - \left(\frac{b_{xl} + b_{xr}}{2}\right)\right)^2 + (a_y - b_y)^2}$$

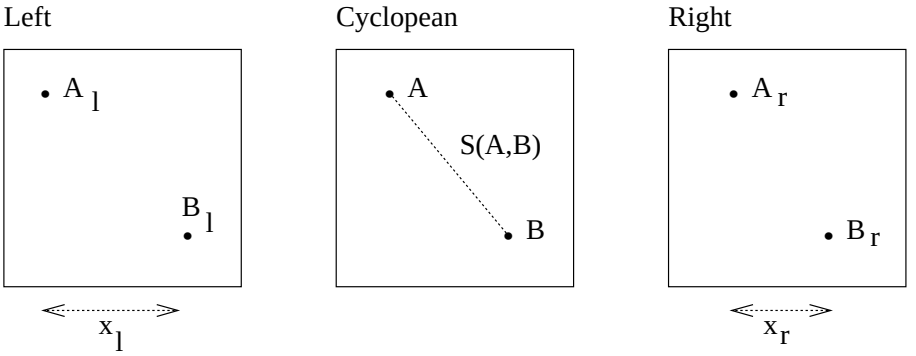


Figure 9.25: *Definition of the disparity gradient.*

$$= \sqrt{\frac{1}{4}((a_{xl} - b_{xl}) + (a_{xr} - b_{xr}))^2 + (a_y - b_y)^2}$$
$$= \sqrt{\frac{1}{4}(x_l + x_r)^2 + (a_y - b_y)^2} \quad (9.90)$$

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

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

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

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

Given these definitions, the constraint exploited is that, in practice, the disparity gradient , 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 [Pollard et al. 85]. 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, centered 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. Another approach – 'shape from shading' – will be explained in Section 9.3.

One way to circumvent these problems is to measure distances from the viewer to points on surfaces in the 3D scene explicitly; such measurements are called **geometric signals**, i.e. a collection of 3D points in a known co-ordinate system. If the surface relief is measured from a single viewpoint it is called a **range image** or a **depth map**. Such explicit 3D information, being closer to the geometric model that is sought, makes geometry recovery easier.[5]

[5]There are techniques that directly measure full 3D information, like mechanical co-ordinate measuring machines (considered in Section 10) or computer tomography.

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

1. The range image must be captured; this procedure is discussed in this Section.

2. Geometric information must be extracted from the range image. Features are sought and compared to a selected 3D model. The selection of features and geometric models leads to one of the most fundamental problems in computer vision; how to represent a solid shape [Koenderink 90].

The term **active sensor** refers to a sensor that uses and controls its own images – the term 'active' means that the sensor uses and controls electromagnetic energy, or more specifically illumination, for measuring a distance between scene surfaces and the 'observer'. An active sensor should not be confused with the active perception strategy, where the sensing subject plans how to look at objects from different views.

RADAR (RAdio Detecting And Ranging) and **LIDAR** (LIght Detecting And Ranging) in one measurement yield the distance between the sensor and a particular point in a measured scene. The sensor is mounted on an assembly that allows movement around two angles, azimuth Θ and tilt Φ, corresponding to spherical co-ordinates. The distance is proportional to the time interval between the emission of energy and the echo reflected from the measured scene object. The elapsed time intervals are very short, so very high precision is required. For this reason, the phase difference between emitted and received signals is often used.

RADAR emits electromagnetic waves in meter, centimeter or millimeter wavelength bands. Aside from military use, it is frequently used for navigation of autonomous guided vehicles.

LIDAR often uses laser as a source of a focused light beam. The higher the power of the laser, the stronger is the reflected signal, and the more precise the measured range. If LIDAR is required to work in an environment together with humans then the energy has an upper limit, due to potential harm to the unprotected eye. Another factor that influences LIDAR safety is the diameter of the laser beam: If it is to be safe it should not be focused too much. LIDARs have trouble when the object surface is almost tangential to the beam, as 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, while transparent objects (obviously) cannot be measured with optical lasers. The advantage of LIDAR is a wide range of measured distances, from a tenth of a millimeter to several kilometers; the accuracy of the measured range is typically around one hundreds of a millimeter. LIDAR provides one range in an instant. If the whole range image is to be captured, the measurement takes several tenths of a seconds as the whole scene is scanned.

Another principle of active range imaging is **structured light triangulation**, where we employ a geometric arrangement similar to that used for stereo vision, with optical axes. One camera is replaced by an illuminant that yields a light plane perpendicular to the epipolars; the image capturing camera is at a fixed distance from the illuminant. Since there is only one significantly bright point on each image line, the correspondence problem that makes passive stereo so problematic is avoided, although there will still be problems with self-occlusion in the scene. Distance from the observer can easily be calculated as in Figure 9.10. To capture a whole range image, the rod with camera and illuminant should be made to move mechanically relative to the scene, and the trace of the laser should gradually illuminate all points to be measured. The conduct of the movement, together with the processing of several hundreds

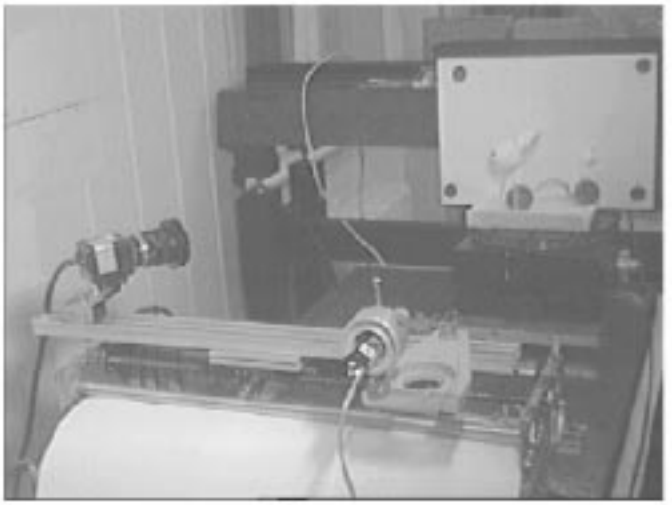


Figure 9.26: *Laser plane range finder. The camera is on the left side, the laser diode on the bottom left. Courtesy T. Pajdla, Czech Technical University, Prague.*

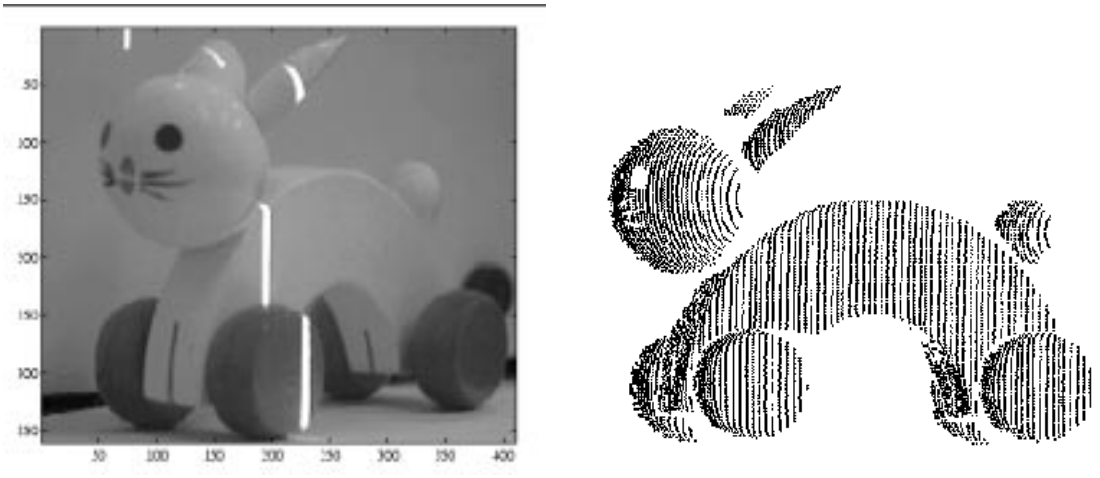
of images, (i.e. one image for each distinct position of the laser stripe) takes some time, typically from a couple of seconds to about a minute. Faster laser stripe range finders find a bright point corresponding to the intersection of a current image line using special purpose electronics.

We shall illustrate an example of such a scanner built in the Center for Machine Perception of the Czech Technical University in Prague. Figure 9.26 shows a view of the scanner together with a target object (a wooden toy - a rabbit). The image seen by the camera with the distinct bright laser stripe is in Figure 9.27(a), and the resulting range image is shown in Figure 9.27(b).

In some applications, a range image is required in an instant, typically meaning one TV frame; this is especially useful for capturing range images of moving objects, e.g. moving humans. One possibility is to illuminate the scene by several stripes at once and code them; Figure 9.28(a) shows a human hand lit by a binary pattern that codes light stripes using a cyclic code such that the local configuration of squares in the image allows to us to decode which stripe it is. In this case, the pattern with coded stripes is projected from a 36×24 mm slide using a standard slide projector. The resulting range image does not provide as many samples as in the case of a moving laser stripe, in our case only 64×80, see Figure 9.28(b).

It is possible to acquire a dense range sample as in the laser stripe case in one TV frame; individual stripes can be encoded using spectral colors and the image captured by a color TV camera [Smutný 93].

Two further measuring principles will conclude this discussion of active range sensors: One is sonar, that uses ultrasonic waves as an energy source. Sonars are used in robot navigation for close range measurements. Their disadvantage is that measurements are typically very noisy. The second principle is Moiré interferometry [Klette et al. 96], in which two periodic patterns, typically stripes, are projected on the scene. Due to interference, the object is covered by a system of closed, non-intersecting curves, each of which lies in a plane of constant



(a) (b)

Figure 9.27: *Measurement using a laser stripe range finger: (a) The image seen by a camera with a bright laser stripe, (b) reconstructed range image displayed as a point cloud. Courtesy T. Pajdla, Czech Technical University, Prague.*

distance from the viewer. Distance measurements obtained are only relative, and absolute distances are unavailable. The properties of Moiré curves are very similar to level curves on maps.

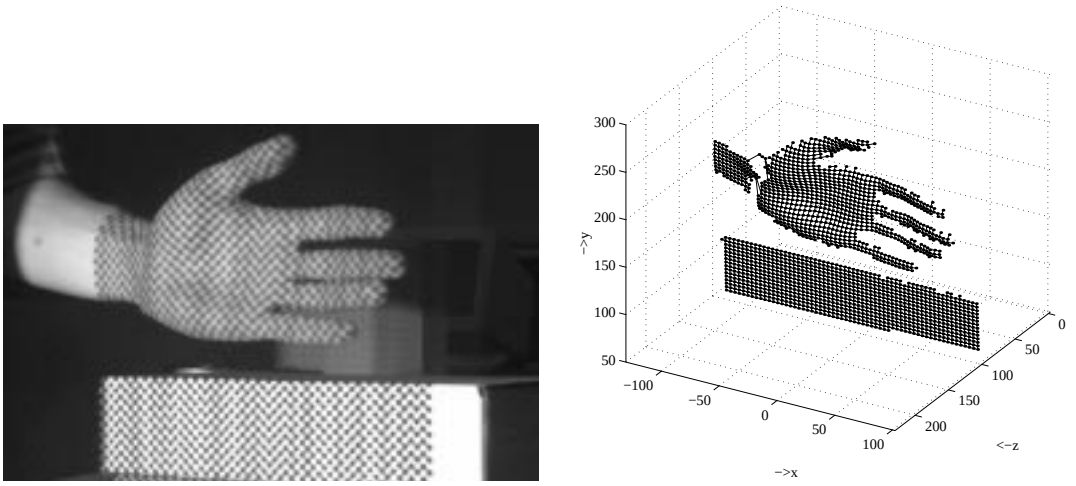
9.3 Radiometry and 3D vision

9.3.1 Radiometric considerations in determining gray level

A TV camera and most other artificial vision sensors measure the amount of received light energy in individual pixels as the result of interaction among various materials and light source(s); the value measured is informally called gray level (or brightness). **Radiometry** is a branch of physics that deals with the measurement of the flow and transfer of radiant energy, and is the appropriate tool to consider the mechanism of image creation. The gray level corresponding to a point on a 3D surface depends, informally speaking, on the shape of the object, its reflectance properties, the position of the viewer and properties and position of the illuminants [Nicodemus et al. 77]. We will later use these concepts to consider derivation of 3D shape from shading.

The radiometric approach to understanding gray levels is very often avoided in practical applications because of its complexity and numerical instability. The gray level measured typically does not provide a precise quantitative measurement (one reason is that CCD cameras are much more precise geometrically than radiometrically; another more serious reason is that the relation between gray level and shape is too complicated). One way to circumvent this is to use task specific illumination that allows the location objects of interest on a qualitative level, and their separation from the background.

Photometry is a discipline closely related to radiometry that studies the sensation of



(a) (b)

Figure 9.28: *Binary coded range finder: (a) The captured image of a hand, (b) reconstructed surface. Courtesy T. Pajdla, Czech Technical University, Prague.*

radiant light energy in the human eye; both disciplines describe similar phenomena using similar quantities. Herein, we shall describe physical units using square brackets; when there is a danger of confusion we shall denote photometric quantities using the subscript $_{ph}$, and leave radiometric ones with no subscript.

The basic radiometric quantity is **radiant flux** Φ $[W]$, and its photometric counterpart is **luminous flux** Φ_{ph} $[lm$ (=lumen)]. For light of wavelength $\lambda = 555$ and daylight vision, we can convert between these quantities with the relation $1\ [W] = 680\ [lm]$. Different people have different abilities to perceive light, and photometric quantities depend on the spectral characteristic of the radiation source and on the sensitivity of photoreceptive cells of a human retina. For this reason, the international standardization body CIE defined a 'standard observer' corresponding to average abilities. Let $K(\lambda)$ be the **luminous efficacy** $[lm\,W^{-1}]$, $S(\lambda)$ $[W]$ the spectral power of the light source, and λ $[W]$, the wavelength. Then luminous flux Φ_{ph} is proportional to the intensity of perception and is given by

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

Since photometric quantities are too observer dependent, we shall consider radiometric ones. From a viewer's point of view, the surface of an object can reflect energy into a half-sphere, 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 center of the sphere. The whole half-sphere corresponds to the spatial angle 2π $[sr$ (=steradians)]. A small area A at distance R from the origin (i.e. $R^2 \gg A$) and with angle Θ between the normal vector to the area and radius vector between the origin and the area corresponds to the spatial angle Ω $[sr]$ (see Figure 9.29).

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

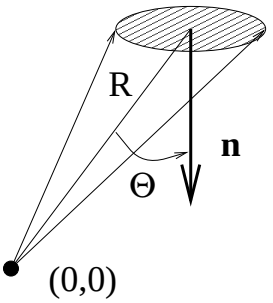


Figure 9.29: *Spatial angle for an elementary surface area.*

Irradiance E $[W\,m^{-2}]$ describes the power of the light energy that falls onto a unit area of the object surface, $E = \delta\Phi\delta A$, where δA is an infinitesimal element of the surface area; the corresponding photometric quantity is **illumination** $[lm\,m^{-2}]$. **Radiance** L $[W\,m^{-2}\,sr^{-1}]$ is the power of light that is emitted from a unit surface area into some spatial angle, and the corresponding photometric quantity 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.

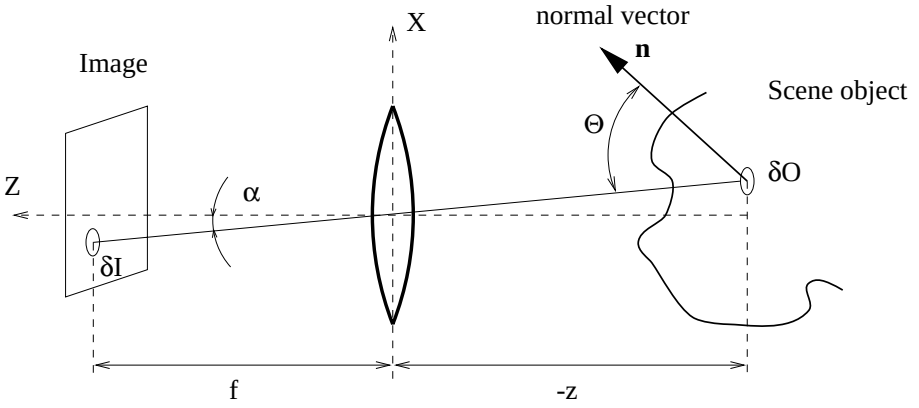


Figure 9.30: *The relation between irradiance E and radiance L.*

Irradiance is given by the amount of energy that an image capturing device gets per unit of an efficient sensitive area of the camera [Horn 86] - then gray 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 consider the relationship between the irradiance E measured in the image and the radiance L produced by a small 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 given in Figure 9.30 The optical axis is aligned with the horizontal axis Z, and a lens with focal length f is placed at the co-ordinate origin (the optical center). The elementary object surface patch δO is at distance z. We are interested in how much light energy reaches an elementary patch of the sensor surface δI. The off-axis angle α spans between the axis Z and the line connecting δO with δI; as we are considering a perspective projection, this line must pass through the origin. The elementary

object surface patch δO is tilted by the angle Θ measured between the object surface normal $\mathbf{n}$ at the patch and a line between δO and δI.

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 center is $\delta I \cos\alpha$, and its distance from the optical center is $f/\cos\alpha$. The corresponding spatial angle is

$$\frac{\delta I \cos\alpha}{(\frac{f}{\cos\alpha})^2}$$

Analogously, the spatial angle corresponding to the elementary patch δO on the object surface is

$$\frac{\delta O \cos Theta}{(\frac{z}{\cos\alpha})^2}$$

As the spatial angles are equal

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

Consider how much light energy passes through the lens if its aperture has diameter d; the spatial angle Ω_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 Φ 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 energy losses 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 as

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

If we substitute for $\frac{\delta O}{\delta I}$ from equation (9.95) we obtain an important equation that reveals how scene radiance influences 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 a systematic lens optical defect called **vignetting**[6] that notes that optical rays with larger span-off angle α are attenuated more; this means that pixels closer to 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 with a radiometrically calibrated lens. The term $\frac{d}{f}$ is called the f-number of the lens and describes how much the lens differs from a pinhole model.

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

9.3.2 Surface reflectance

In many applications, pixel gray level is constructed as an estimate of image irradiance as a result of light reflection from scene objects. Consequently, it is necessary to understand different mechanisms involved in reflection. Here we give just a brief overview that later permits us to explain the main ideas behind shape from shading. Consult [Ikeuchi 94, Foley et al. 90, Klette et al. 96] for more detailed explanations.

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

An important concept now is **gradient space** which is a way of describing surface orientations (and has also been used in the analysis of line labeling problems [Mackworth 73]). 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. We shall express the unit surface normal using surface gradient components; if we move a small distance ∂x in the 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, and analogously $[0, 1, q]^T$ is also tangent to the surface. The surface normal is perpendicular to all its tangents, and may 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.

The pair $[p, q]$ is the two-dimensional gradient space representation of the surface orientation. 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.

Consider now spherical co-ordinates used to express the geometry of an infinitesimal surface patch – see Figure 9.31. The **polar angle** (also called zenith angle) is Θ and the **azimuth** is φ. We shall attempt to describe the ability of different materials to reflect light. The direction towards a point light source is denoted by subscript $_i$ (i.e. Θ_i and φ_i),

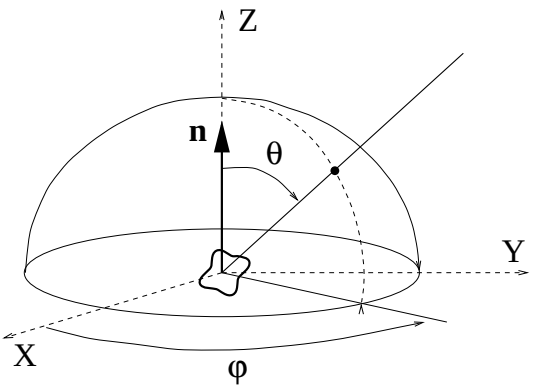


Figure 9.31: *Polar and spherical angles used to describe orientation of a surface patch.*

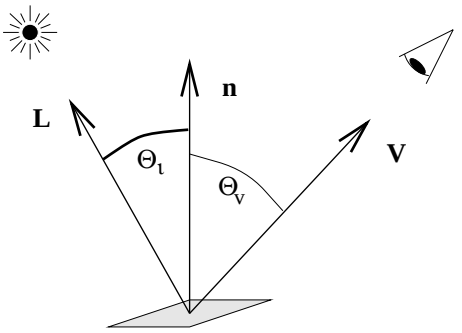


Figure 9.32: *Directions towards the viewer and the light source.*

while subscript $_v$ identifies the direction toward the viewer (Θ_v and φ_v) – see Figure 9.32. The irradiance of the elementary surface patch from the light source is $\delta E(\Theta_i, \varphi_i)$, and the elementary contribution of the radiance in the direction towards the viewer is $\delta L(\Theta_v, \varphi_v)$. In general, the ability of the body to reflect light is described using a **bidirectional reflectance distribution function** f_r $[sr^{-1}]$, abbreviated BDRF [Nicodemus et al. 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 describes the brightness of an elementary surface patch for a specific material, light source, and viewer directions. Modeling of the BDRF is also important for realistic rendering in computer graphics [Foley et al. 90]. 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 aluminum).

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

Let $E_i(\lambda)$ denote the irradiance caused by the illumination of the surface element, and $E_r(\lambda)$ the energy 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 the **reflectance coefficient** or **albedo**. Albedo describes what proportion of incident energy is reflected back to the half-space. For simplicity, assume that we may neglect color properties of the surface, and suppose that albedo does not depend on the wavelength λ. This proportion is then an integral of the surface radiance L over the solid angle Ω representing the half-space;

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

Now define a **reflectance function** $R(\Omega)$ that models the influence of the local surface geometry into the spatial spread of the reflected energy. Ω is an infinitesimal solid angle around the viewing direction.

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

In general, surface reflectance properties depend on three angles between the direction to the light source $\mathbf{L}$, the direction towards the viewer $\mathbf{V}$ and local surface orientation given by the surface normal $\mathbf{n}$ (recall Figure 9.32). The cosines of these angles can be expressed as scalar (dot) products of vectors; thus the reflectance function is a scalar function of the following three dot products

$$R = R(\mathbf{nL}, \mathbf{nV}, \mathbf{VL}) \tag{9.107}$$

A **Lambertian surface** (also ideally opaque, with ideal diffusion) reflects light energy in all directions, and thus the radiance is constant in all all directions. The BDRF f_{Lambert} is constant

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

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

$$R = \frac{1}{\pi}\mathbf{nL} = \frac{1}{\pi}\cos\Theta_i \tag{9.109}$$

Because of its simplicity, the Lambertian reflectance function has been widely accepted as a reasonable reflectance 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, and the **reflectance map** $R(p, q)$ is used for this purpose. The $R(p, q)$ can be visualized in gradient space as nested isocontours corresponding to the same observed irradiance.

Values of the reflectance map may be:

1. Measured experimentally on a device called a goniometer stage that is able to set angles Θ and φ mechanically. A sample of the surface is attached to the goniometer and its reflectance measured for different orientations of viewer and light sources.

2. Set experimentally if a calibration object is used. Typically a half-sphere is used for this purpose.

3. Derived from a mathematical phenomenonical model describing surface reflecting properties

The best known surface reflectance models are the Lambertian model for ideal opaque surfaces, the **Phong** model that models reflection from dielectric materials, the **Torrance-Sparrow** model which describes surfaces as a collection of planar mirror-like microfacets with normally distributed normals, and the wave theory based **Beckmann-Spizzichino** model. A survey of surface reflection models from the point of view of computer vision, and their recent modifications, can be found in [Ikeuchi 94].

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 a 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**;

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

In an attempt to reduce complexity, several simplifying assumptions [Horn 90] are usually made to ease the shape from shading task. It is assumed that:

- The object has uniform reflecting properties, i.e. $\rho(u,v)$ is constant.
- The light sources are distant; then 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 scene surfaces does not depend on position but only on orientation. The perspective projection is simplified to an orthographic one.

We present 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);

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

$R(p,q)$ gives the radiance of the corresponding point in the scene; the proportionality constant β comes from equation (9.99) and depends on the f-number of the lens. The vignetting degradation of the lens is negligible as the viewer is aligned to the illuminant. The measured irradiance E can be normalized and the factor β omitted; this permits us to write the **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 intensity images.

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 neighboring 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 a surface normal **n**. The aim of shape from shading is to extract information about normals of surfaces in view solely on the basis of an 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 [Horn 70, Horn 75].

Techniques similar to shape from shading were earlier proposed independently in photoclinometry [Rindfleisch 66] when astrogeologists wanted to measure steepness of slopes on planets in the solar system from intensity images observed by terrestrial telescopes. There are two significant differences here from shape from shading:

- Surface normals are calculated by the integration along a space curve (called the profile; it is a 1D entity if the curve is arclength parameterized). In shape from shading, the integration is performed on the surface area, which is a 2D entity if the surface is parametrized.

- Shape from shading is more concerned with ambiguity of solutions. The use of singular points and occluding boundaries helps to combat this ambiguity. The surface normal can be then uniquely computed.

We shall classify shape from shading methods into three categories, and proceed to describe them:

Incremental propagation from surface points of known height

The oldest, and easiest to explain, method develops a solution along a space curve. This is also called the characteristic strip method.

We can begin to analyze the problem of global shape extraction from shading information when the reflectance function and the lighting model are both known perfectly [Horn 90]. Even given these constraints, it should be clear that the mapping 'surface orientation to brightness' is many-to-one, since 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 9.33 – 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 gray level.

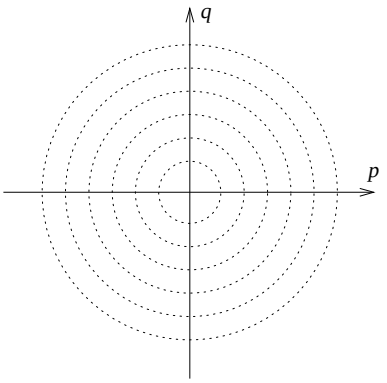


Figure 9.33: *Reflectance map for a matte surface – the light source is adjacent to the viewer.*

The original formulation [Horn 70] to 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 co-ordinates of a surface point $[x, y, z]^T$ and we want to propagate the solution along an infinitesimal step on the surface, e.g. taking small steps δx and δy, then calculating the change in height δz. This can be done if the components of the surface gradient p, q are known. For compactness we use an index notation, and express $p{=}\delta z/\delta x$ as z_x, and $\delta^2 x/\delta x^2$ as z_{xx}. The infinitesimal change of height is

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

The surface is followed stepwise, with values of p, q being traced along with x, y, z. Changes in p, q are calculated using second derivatives of height $r{=}z_{xx}$, $s{=}z_{xy}{=}z_{yx}$, $t{=}z_{yy}$

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

Consider now the image irradiance equation $E(x, y) = R(p, q)$ – equation (9.112) - and differentiate with respect to x, y to obtain the 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 δx, δy can be arbitrarily chosen;

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

The parameter ξ changes along particular solution curves. Moreover, the orientation of the surface along this curve is known; thus it is called a characteristic strip.

We can now express changes of gradient δp, δq as dependent on gradient image intensities, which is the crucial 'trick'. A set of ordinary differential equations can be generated by considering equations (9.114) and (9.115); dot denotes differentiation with respect to ξ

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

There are points on the surface for which the surface orientation is known in advance and these provide boundary conditions during normal vector calculations. These are

- Points of a surface **occluding boundary**; an occluding boundary is a curve on the surface due to the surface rolling away from the viewer, i.e. the set of points for which the local tangent plane coincides with the direction towards the viewer. The surface normal at such a boundary can be uniquely determined as it is parallel to the image plane and perpendicular to the direction towards the viewer. This normal information can be propagated into the recovered surface patch from the occluding boundary. Although the occlusion boundary uniquely constrains surface orientation, it does not constrain the solution sufficiently to recover depth uniquely [Oliensis 91].

- Singular points in the image; we have seen that at most surface points gradient is not fully constrained by image intensities. Suppose that the reflectance function $R(p,q)$ has a global maximum, so $R(p,q) < R(p_0,q_0)$ for all $[p,q] \neq p_0,q_0$. This maximum corresponds to singular points in the image;

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

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

It is reported [Horn 90] that direct implementation of this characteristic strip method does not yield particularly good results due to numerical instability.

Global optimization methods

These methods are formulated as a variational task in which the whole image plays a role in the chosen functional. Results obtained are in general better than those generated by incremental methods.

We already know that under the simplifying conditions for recovery of 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

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

The task is to find the surface height $z(x,y)$ 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 neighboring 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 15.2.1, 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 λ 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 15.2.1, and the method of optimization using Lagrange multipliers is described in many books [Horn 86].

The significant work in this area is due to Horn [Horn 75] and Ikeuchi [Ikeuchi and Horn 81] 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 modeled 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 [Horn and Brooks 89].

Local shading analysis

Local shading analysis methods use just a small neighborhood of the current point on the surface, and seek a direct relation between the differential surface structure and the local structure of the corresponding intensity image. The surface is considered as a set of small neighborhoods, each defined in some local neighborhood of one of its points. Only an estimate of local surface orientation is available, not information about the height of a particular surface point.

The main advantage of local shading analysis is that it provides surface related information to higher-level vision algorithms from a single monocular intensity image without any need to reconstruct the surface in explicit depth form [Šára 95]. This is possible since the intensity image is closely related to local surface orientation. 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 approach is, of course, much faster than the solution propagation or global variational methods.

The fundamental contribution to local shading analysis comes from Pentland [Pentland 84]; an overview can be found in [Pentland and Bichsel 94]. In addition, Šára [Šára 94] demonstrates:

1. It was known that local surface orientation and Gaussian curvature sign can be determined uniquely at occlusion boundaries. Further orientation on a self-occluding boundary can also be determined uniquely; self-occluding contours are thus a rich source of unambiguous information about the surface.

2. The differential properties of isophotes (curves of constant image intensity) are closely related to the properties of the 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 [Woodham 80]. Consider a particular Lambertian surface with varying albedo ρ. The key idea of photometric stereo is to look at the surface from one fixed viewing direction while changing the direction of incident illumination. Assume we have three or more such images of the Lambertian surface; then the surface normals can be uniquely determined based on the shading variations in the observed images.

The lines of constant reflectance on the surface correspond to lines of constant irradiation E in the image (called also isophotes); these curves observed in images are 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 directions, the surface reflectance remains the same on the surface but the observed reflectance map $R(p, q)$ changes. This provides an additional constraint on possible surface orientation that is another second order polynomial. Two views corresponding to two distinct illumination directions are not enough to determine the surface orientation $[p, q]$ uniquely, and a third view is needed to derive a unique solution. If more than three distinct illuminations are at hand, an overdetermined set of equations can be solved.

A practical setup for image capture 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 light source is active at any one time. The setup should be photometrically calibrated to take into account light source intensities, particular camera gain, and offset; such a calibration is described in [Haralick and Shapiro 93]. 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 ρ, $0 \leq \rho \leq 1$, occurs in the image irradiance (as shown in equation (9.110)). For a Lambertian surface the image irradiance equation simplifies to

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

Recall equation (9.109) (called the cosine law) showing that the reflectance map of a 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

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

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

If we have three light sources, $K = 3$, we can derive a solution by inverting the regular matrix L

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

The unit normal is then

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

For more than three light sources, the pseudo-inverse of a rectangular matrix is determined to get a solution in the least square sense

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

Note that the pseudoinversion (or inversion in equation (9.127)) must be repeated for each image pixel x, y to derive an estimate of the corresponding normal.

9.4 Summary

- 3D vision aims at inferring 3D information from 2D scenes, a task with embedded geometric and radiometric difficulties. The geometric problem is that a single image does not provide enough information about 3D structures, and the radiometric problem is the complexity of the physical process of intensity image creation. This process is complex, and typically not all input parameters are known precisely.
- **3D vision tasks**
  - There are several approaches to 3D vision which may be categorized as *bottom-up (or reconstruction)* or *top-down (model-based vision)*.
  - *Marr's theory*, formulated in the late Seventies, is an example of the bottom-up approach. The aim is to reconstruct qualitative and quantitative 3D geometric descriptions from one or more intensity images under very weak assumptions about objects in the scene.

- There are four representations ordered in bottom-up fashion: (1) input intensity image(s); (2) primal sketch, representing in viewer-centered co-ordinates significant edges in the image; (3) 2.5D sketch, representing depth from the observer and local orientation of the surface; and (4) 3D representation, representing object geometry in co-ordinates related to the objects themselves.
- The 2.5D sketch is derived from the primal sketch by a variety of techniques called shape from X.
- 3D representations are very hard to obtain; this step has not been solved in the general case.
- More recent perception paradigms such as active, purposive and qualitative vision try to provide a computational model explaining the 'understanding' aspects of vision.
- None have yet led to direct practical applications, but many partial techniques (such as shape from X) are widely used in practice.

- **Radiometry and 3D vision**
  - 3D perspective geometry is the basic mathematical tool for 3D vision as it explains a pinhole camera.
  - Lines parallel in the 3D world do not project as parallel lines in 2D image.
  - The case of the single perspective camera permits carefully study of calibration of intrinsic and extrinsic camera parameters.
  - Two perspective cameras constitute stereopsis and allow depth measurements in 3D scenes.
  - Epipolar geometry teaches us that the search for corresponding points is inherently one-dimensional. This can be expressed algebraically using the fundamental matrix.
  - This tool has several applications such as image rectification, ego-motion estimation from calibrated cameras measurements, 3D Euclidean reconstruction from two fully calibrated cameras, 3D similarity reconstruction from two cameras with only intrinsic calibration parameters known, and 3D projective reconstruction from two uncalibrated cameras.
  - There a is trilinear relation among views of from three cameras that is algebraically expressed using a trifocal tensor.
  - The application of the trilinear relation is in epipolar transfer; if two images are known together with the trifocal tensor, the third perspective image can be computed.
  - The correspondence problem is core to 3D vision; various passive and active techniques to solve it exist.

- **Radiometry and 3D vision**
  - Radiometry informs us about the physics of image formation.

- If it is understood together with position of illuminants, type, surface reflectance and viewer position, something can be learned about depth and scene surface orientation from one intensity image.
- This task is called *shape from shading*.
- The task is ambiguous and numerically unstable. Shape from shading can be understood in the simple case of Lambertian surfaces.
- There is a practical method that uses one camera and three known illuminants, selective illumination provides three intensity images.
- Photometric stereo allows measure of orientation of surfaces.

9.5 Exercises

Short-answer questions

1. Explain the difference between a bottom-up approach (object reconstruction) to 3D vision as opposed to top-down (model-based).
2. Explain the basic idea of active vision, and give some examples of how this approach eases vision tasks.
3. Give examples of perspective images from everyday life. Where do parallel lines in the world not correspond to parallel ones in images?
4. What are the intrinsic and extrinsic calibration parameters of a single perspective camera? How are they estimated from known scenes?
5. Do zoom lenses typically have worse geometric distortion compared to fixed focal length lenses? Is the difference significant?
6. What is the main contribution of epipolar geometry in stereopsis?
7. Where are epipoles in the case of two cameras with parallel optical axes (the *canonical* configuration for stereopsis)?
8. What is the difference between the fundamental and essential matrices in stereopsis?
9. How are mismatches in correspondences treated in stereopsis?
10. What are the applications of epipolar geometry in computer vision?
11. Explain the principle, advantages and applications of a trilinear relation among three cameras. What is epipolar transfer?
12. Stereo correspondence algorithms are typically lost if the left and right images have large regions of uniform brightness. How can depth acquisition still be made possible?
13. Active range finders (e.g. with a laser plane) suffer from occlusions; some points are not visible by the camera and some are not lit. What are the ways of dealing with this problem?
14. What is Moiré interferometry? Does it give absolute depth?
15. Why is the relation between pixel intensity on one side and surface orientation, surface reflectance, illuminant types and position, and viewer position on the other side difficult?
16. What is the vignetting error of a lens?
17. Under which circumstances can the surface orientation be derived from intensity changes in an image?

Problems

1. This problem relates to Marr's theory, in particular the representation scheme called primal sketch (see Section 9.1.1).

   Capture an intensity image (e.g. of an office scene), and run an advanced edge detector (such as Canny's or similar) on it. Threshold the magnitude of the image gradient.

   Answer the following questions in an essay: Are the lines which you get the primal sketch? Would it be possible to derive a 2.5D sketch directly from it? How? Discuss what more would you need in the primal sketch. What about multiple scales?

2. Explain the notion of homogeneous co-ordinates. Is projective transformation linear if expressed in homogeneous co-ordinates? Why are homogeneous co-ordinates often used in robotics to express the kinematics of a manipulating arm? (Hint: express rotation and translation of an object in 3D space using homogeneous co-ordinates)

3. Take a camera with an off-the-shelf lens. Design and perform an experiment to find the intrinsic calibration parameters of it. Design and use an appropriate calibration object (for example, a grid like structure printed by laser printer on paper; you might capture it at different heights by placing it on a box of known height). Discuss the precision of your results. Is the pinhole model of your camera appropriate? (Hint: look at distortions of a grid as in Figure 9.7).

4. Consider the case of two cameras (stereopsis) with baseline $2h$. A scene point lies on the optical axis at depth d from the baseline. Assume that the precision of pixel position measurement x in the image plane is given by dispersion σ^2. Derive a formula showing the dependence of the precision of depth measurement against dispersion. (Hint: differentiate d according to x.)

5. Conduct an experiment with stereo correspondences. For simplicity, capture a pair of stereo images using cameras in canonical configuration (epipolar lines correspond to lines in images), and cut corresponding lines from both images. Visualize the brightness profiles in those lines (for example, using MATLAB or another package). First, try to find correspondences in brightness manually. Second, decide if correlation-based or feature-based stereo techniques are more suitable for your case. Program it and test on your profiles.

6. Conduct a laboratory experiment with photometric stereo. You will need one camera and three light sources. Take some opaque object and measure its surface orientation using photometric stereo (see Section 9.3.4).

9.6 References

[Aloimonos 92] Y Aloimonos, editor. *Special Issue on Purposive and Qualitative Active Vision*, volume 56, 1992.

[Aloimonos 93] Y Aloimonos, editor. *Active Perception*. Lawrence Erlbaum Associates, Inc., Publishers, Hillsdale, New Jersey, 1993.

[Aloimonos 94] Y Aloimonos. What i have learned. *CVGIP: Image Understanding*, 60(1):74–85, 1994.

[Aloimonos and Rosenfeld 94] Y Aloimonos and A Rosenfeld. Principles of computer vision. In Young [Young 94], pages 1–15.

[Aloimonos and Shulman 89] Y Aloimonos and D Shulman. *Integration of Visual Modules - An Extension of the Marr Paradigm*. Academic Press, New York, 1989.

[Ayache and Hansen 88] 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, IEEE Computer Society Press, Los Alamitos, Ca., USA, 1988.

[Bajcsy 88] R Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, 1988.

[Bertero et al. 88] M Bertero, T Poggio, and V Torre. Ill-posed problems in early vision. *IEEE Proceedings*, 76:869–889, 1988.

[Besl and Jain 85] P J Besl and R C Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, March 1985.

[Besl and Jain 88] P J Besl and R Jain. *Surfaces in range image understanding.* Springer-Verlag, New York, 1988.

[Beymer and Poggio 96] D Beymer and T Poggio. Image representations for visual learning. *Science*, 272:1905–1909, 1996.

[Biederman 87] I Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[Bowyer 92] K W Bowyer, editor. *Special issue on directions in CAD-based vision*, volume 55, 1992.

[Brooks et al. 79] 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.

[Butterfield 97] S Butterfield. *Reconstruction of Extended Environments from Image Sequences.* PhD thesis, School of Computer Studies, University of Leeds, Leeds, UK, 1997.

[Buxton and Howarth 95] H Buxton and R J Howarth. Spatial and temporal reasoning in the generation of dynamic scene representations. In R V Rodriguez, editor, *Proceedings of Spatial and Temporal Reasoning*, pages 107–115, IJCAI-95, Montreal, Canada, 1995.

[Farshid and Aggarwal 93] 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.

[Faugeras 93] O Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, 1993.

[Faugeras and Mourrain 95] 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, IEEE Computer Society Press, Boston, USA, June 1995.

[Faugeras et al. 92] 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.

[Fernyhough 97] J F Fernyhough. *Generation of qualitative spatio-temporal representations from visual input.* PhD thesis, School of Computer Studies, University of Leeds, Leeds, UK, 1997.

[Flynn and Jain 91] 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.

[Flynn and Jain 92] P J Flynn and A K Jain. 3D object recognition using invariant feature indexing of interpretation tables. *CVGIP – Image Understanding*, 55(2):119–129, 1992.

[Foley et al. 90] 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.

[Goad 86] C Goad. Fast 3D model-based vision. In A P Pentland, editor, *From Pixels to Predicates*, pages 371–374. Ablex Publishing Corporation, Norwood, NJ, 1986.

[Golub and Loan 89] G H Golub and C F Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[Haralick and Shapiro 93] R M Haralick and L G Shapiro. *Computer and Robot Vision, Volume II.* Addison Wesley, Reading, Ma., 1993.

[Hartley 92] 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, Springer - Verlag,, Heidelberg, Germany, May 1992.

[Hartley 94] 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.

[Hartley 95] R I Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, IEEE, Boston, USA, June 1995.

[Hlaváč et al. 96] 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, Springer-Verlag, Heidelberg, Germany, Cambridge, U K, April 1996.

[Horaud et al. 95] 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.

[Horn 70] B K P Horn. *Shape from shading: A method for obtaining the shape of a smooth opaque sobject from one view.* PhD thesis, MIT, Department of Electrical Engineering, Cambridge, MA., USA, 1970.

[Horn 75] B K P Horn. Shape from shading. In P H Winston, editor, *The Psychology of Computer Vision.* McGraw Hill, New York, 1975.

[Horn 86] B K P Horn. *Robot Vision.* MIT Press, Cambridge, Ma, 1986.

[Horn 90] B K P Horn. Height and gradient from shading. *International Journal of Computer Vision*, 5(1):37–75, 1990.

[Horn and Brooks 89] B K P Horn and M J Brooks, editors. *Shape from Shading.* MIT Press, Cambridge, Ma, 1989.

[Howarth 94] R J Howarth. *Spatial representation and control for a surveillance system.* PhD thesis, Department of Computer Science, Queen Mary and Westfield College, Univerity of London, UK, 1994.

[Ikeuchi 94] K Ikeuchi. Surface reflection mechanism. In Young [Young 94], pages 131–160.

[Ikeuchi and Horn 81] K Ikeuchi and B K P Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.

[Jain et al. 95] R Jain, R Kasturi, and B G Schunk. *Machine Vision.* McGraw-Hill, New York, USA, 1995.

[Julesz 90] B Julesz. Binocular depth perception of computer-generated patterns. *Bell Syst. Tech. Journal*, 39, 1990.

[Klette et al. 96] R Klette, A Koschan, and K Schlüns. *Computer Vision - Räumliche Information aus digitalen Bildern.* Friedr. Vieweg & Sohn, Braunschweig, 1996.

[Klir 91] G J Klir. *Facets of System Science.* Plenum Press, New York, USA, 1991.

[Koenderink 90] J J Koenderink. *Solid Shape.* The MIT Press, 1990.

[Landy et al. 96] M S Landy, L T Maloney, and M Pavel, editors. *Exploratory Vision: The active eye*. Springer Series in Perception Engineering. Springer-Verlag, New York, USA, 1996.

[Longuet-Higgins 81] H C Longuet-Higgins. A computer algorithm for reconstruction a scene from two projections. *Nature*, 293(10):133–135, September 1981.

[Mackworth 73] A K Mackworth. Interpreting pictures of polyhedral scenes. *Artificial Intelligence*, 4(2):121–137, 1973.

[Marr 82] D Marr. *Vision – A Computational Investigation into the Human Representation and Processing of Visual Information*. W H Freeman and Co., San Francisco, 1982.

[Marr and Hildreth 80] D Marr and E Hildreth. Theory of edge detection. *Proceedings of the Royal Society*, B 207:187–217, 1980.

[Marr and Poggio 79] D Marr and T A Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society*, B 207:301–328, 1979.

[Maybank and Faugeras 92] 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.

[Mohr 93] 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.

[Newman et al. 93] 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.

[Nicodemus et al. 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.

[Nishihara 84] H K Nishihara. Practical real-time imaging stereo matcher. *Optical Engineering*, 23(5):536–545, 1984.

[Oliensis 91] 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.

[Pajdla and Hlaváč 95] 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.

[Pentland 84] A P Pentland. Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):170–187, March 1984.

[Pentland and Bichsel 94] A P Pentland and M Bichsel. Extracting shape from shading. In Young [Young 94], pages 161–183.

[Poggio et al. 85] T Poggio, V Torre, and C Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[Pollard et al. 85] 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.

[Prescott and McLean 97] B Prescott and G F McLean. Line-based correction of radial lens distortion. *GMIP*, 59(1):39–77, January 1997.

[Press et al. 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.

[Rindfleisch 66] T Rindfleisch. Photometric method form lunar topography. *Photogrammetric Engineering*, 32(2):262–267, 1966.

[Semple and Kneebone 63] J G Semple and G T Kneebone. *Algebraic Projective Geometry*. Oxford University Press, Ely House, London W, 1963.

[Shashua and Werman 95] 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.

[Smutný 93] 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.

[Soucy and Laurendeau 92] 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, IEEE Computer Society Press, Volume I, The Hague, The Netherlands, September 1992.

[Tanaka and Kak 90] 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.

[Tichonov and Arsenin 77] A N Tichonov and V Y Arsenin. *Solution of ill-posed problems*. Winston and Wiley, Washington, D C , USA, 1977.

[Tsai 87] 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.

[Ullman 96] S Ullman. *High-level Vision: Object Recognition and Visual Cognition*. A Bradford Book. MIT Press, Cambridge, Ma., USA, 1996.

[Ullman and Basri 91] 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.

[Šára 94] R Šára. *Local Shading Analysis via Isophotes Properties*. PhD thesis, Johannes Kepler University Linz, Dept. of System Sciences, March 1994.

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

[Wechsler 90] H Wechsler. *Computational Vision*. Academic Press, London – San Diego, 1990.

[Weinshall et al. 95] 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, IEEE Computer Society Press, Los Alamitos, Ca. USA, 1995.

[Werner et al. 95] 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, IEEE Press, Boston, USA, June 1995.

[Woodham 80] R J Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19:139–144, 1980.

[Young 94] T Y Young, editor. *Handbook of Pattern Recognition and Image Processing: Computer Vision*, volume 2, San Diego, USA, 1994. Academic Press.