

Chapter 2

System Description

Most vision systems are currently task-oriented, which means that different systems are configured for different vision tasks in various environments. Anyway, a good vision system should make full use of all the merits of each component within the system. The main objective of this chapter is to provide a brief introduction of various components involved in the concerned vision system, i.e. structured light system and omni-directional vision system. And their characteristics are analyzed at the same time. Firstly, we introduce the mathematical models for the mirror, the camera and the projector, respectively. Then the coding and decoding strategies for the light pattern of the projector are discussed. Finally, we present some preliminaries that will be used in the succeeding chapters.

2.1 System Introduction

In this book, we consider two different kinds of vision systems. One is the structured light system which can actively illuminate its environment with a predefined light pattern. The distinct advantage of such system is that desired features can be created via the design of the pattern. The other is omni-directional vision system, in which a specially-shaped mirror or lens is generally involved. Hence, such system possesses a much larger field of view than the traditional system.

2.1.1 Structured Light System

In general, the classic stereo vision system is configured with one or more cameras (Fig. 2.1). Before a vision task starts, such as 3D reconstruction, feature points should be tracked among the images or image sequence. Due to occlusions or changing in element characteristics between different images, some of those features may be lost or false matching. Furthermore, it is very difficult to obtain dense matches in some cases. This is named as the correspondence problem.

Fig. 2.1 A traditional stereo vision system

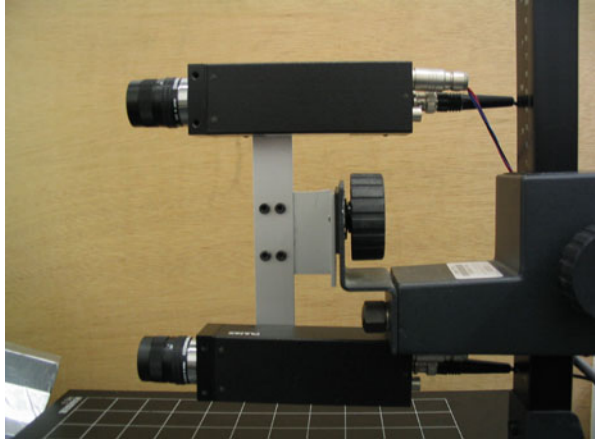
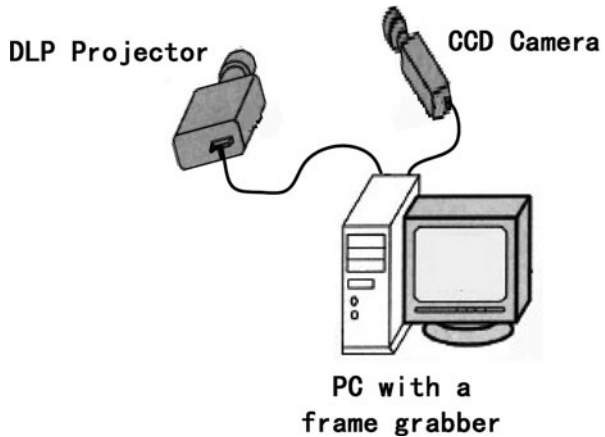


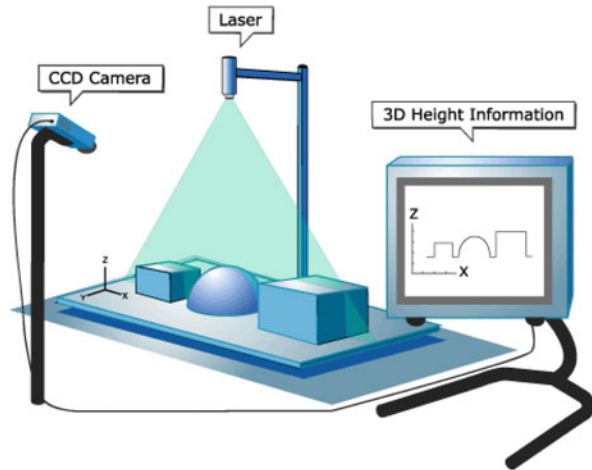
Fig. 2.2 Configuration of a typical structured light system



By replacing one of the cameras in the traditional stereo system with an illumination device, we can obtain a new system usually named the structured light system. The distinct advantage lies in that it provides an elegant solution for the correspondence problem. In practice, there are many alternative illumination devices, such as a laser (Reid [68]), a desk lamp (Bouguet [73]) or a DLP projector (Okatani [132]), which can be used to create desired feature points with a predefined light pattern in the surroundings. Figure 2.2 shows a sketch of a typical structured light system consisting of a CCD camera, a DLP projector and a personal computer with the frame grabber.

When working, the projector which is controlled by a light pattern, projects a bundle of light spots, light planes or light grids into the scene, which in return are reflected by the scene's surface and sensed by the camera as an image (Fig. 2.3). In such scenario, the feature correspondences are easily solved since we already know where they come from according to the design of the light pattern. And dense feature

Fig. 2.3 Profile of a system.
(From http://www.stockeryale.com/i/lasers/structured_light.htm)



points can be extracted on each light stripe in the image. When the vision system is calibrated, 3D modeling of the scene or concerned objects can be carried out with classical triangulation method.

2.1.2 *Omni-Directional Vision System*

The conventional structured light systems generally have limited fields of view, which make them restrictive for certain applications in computational vision. This shortcoming can be compensated through an omni-directional vision system, which combines the refraction and reflection of light rays, usually via lenses (dioptrics) and specially curved mirrors (catoptrics). The angle of view in such system is larger than 180° or even $360 \times 360^\circ$, enabling to capture the entire spherical field of view. In practice, there are various configurations. Figure 2.4 shows two typical systems and their images from [178].

When working, the light rays are firstly reflected by the mirror according to the law of reflection. Then the reflected rays are sensed by the camera into an image. Once the system has been calibrated, 3D reconstruction can be similarly implemented by classical triangulation algorithm.

2.2 Component Modeling

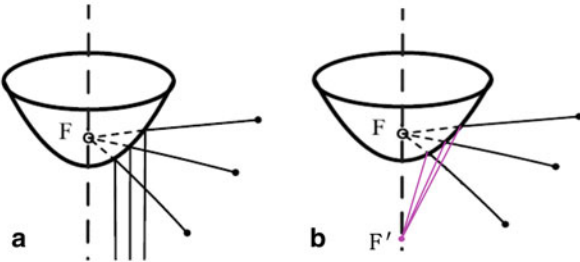
2.2.1 *Convex Mirror*

Many kinds of mirrors, such as planar mirror, Pyramidal multi-faceted mirror, conical and spherical mirror, can be used in an omni-directional vision system to obtain a larger field of view. A plane mirror is simply a mirror with a flat surface while the rest

Fig. 2.4 Two configuration of the catadioptric systems and their images



Fig. 2.5 **a** The parabolic mirror and **b** hyperbolic mirror

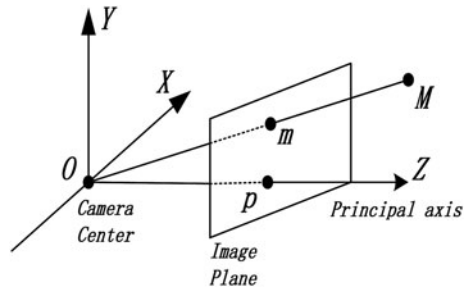


have much complex-shaped surface. They all follow the law of reflection, which says that the direction of the incoming light (incident ray), and the direction of outgoing light (reflected ray) make the same angle with respect to the surface normal. In this book, we mainly employ two types of mirrors, i.e. the parabolic mirror and the hyperbolic mirror.

The parabolic mirror has a particular kind of three-dimensional surface. In the simplest case, it is generated by the revolution of a parabola along its axis of symmetry as illustrated in Fig. 2.5a. Here, the dashed vertical line represents the symmetry axis and F is the focal point. According to its mathematical model, the incident rays will pass through the focal point while their reflected rays will be parallel with the symmetric axis. The parabolic-mirror-camera system, the parabolic camera system for short, is a vision system that incorporates the parabolic mirror with orthogonal projection camera. Certainly, all the light rays round the parabolic mirror can be reflected by its surface and sensed by the camera into an image. Hence, the potential horizontal field of view is 360° .

In mathematics, a hyperboloid is a quadratic surface of revolution which may have one or two sheets. And the symmetry axis passes through its two foci, denoted by F and F' (referring to Fig. 2.5b). The hyperbolic mirror follows the shape of one sheet of a hyperboloid, in which the trajectory of a light ray is: an incoming ray

Fig. 2.6 Illustration of a pinhole camera model



passing through the first focal point, say F , should be reflected such that the outgoing ray will converge at the second focal points, say F' . The hyperbolic-mirror-camera system, the hyperbolic camera system for short, is a vision system that combines a hyperbolic mirror and a pinhole camera that is placed at the point F' . From the ray trajectory, we can see that this type of system has one effective viewpoint and the horizontal field of view is 360° .

2.2.2 Camera Model

A camera is a mapping from the 3D world to 2D image plane. In this book, the perspective camera model is used, which corresponds to an ideal pinhole model. Figure 2.6 illustrates the geometric process for image formation in a pinhole camera. Here, the centre of projection, generally called the camera centre, is placed at the origin of coordinate system. The line from the camera centre perpendicular to the image plane is called the principal axis, while their intersection is called the principal point. Mathematically, the camera model can be represented by the following 3×3 nonsingular camera matrix ([150], see Chap. 5):

$$\mathbf{K}_c = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

where: f_u and f_v represent the focal lengths of the camera in terms of pixel dimensions along U -axis and V -axis respectively, and $(u_0 \ v_0)^T$ is the principal point, s is a skew factor of the camera, representing cosine value of the subtending angle between U -axis and V -axis. Since there are five parameters in this model, it is referred to as a linear five-parameter camera. The ratio of f_u and f_v is often called the aspect ratio of the camera. With a modern camera, the skew can be treated as zero ($s=0$) which means the pixels in the image can be assumed to be rectangular. In this case, the model is referred to as a four-parameter camera.

For a camera with fixed optics, these parameters are identical for all the images taken with the camera. For a camera which has zooming and focusing capabilities, the focal lengths can obviously change. However, as the principal point is the intersecting

point of principal axis with the image, it can be assumed to be unchanged sometimes. It is reported that the assumptions are fulfilled to a sufficient extent in practice [151, 152]. Recently, Sturm et al. [153], Cao et al. [154] and Kanatani et al. [155] used these assumptions by permanently setting the principal point to $(0, 0)$, the skew to zero and the aspect ratio to one in their work. Another interesting work is from Borghese et al. [156], where only the focal length is unknown and can be computed with cross ratio, based on zooming in and out a single 3D point.

So in our work, when dynamically calibrating the intrinsic parameters, we always assume that the focal lengths in both pixel dimensions are unknown and variable, i.e. the camera matrix can be simplified as

$$\mathbf{K}_c = \begin{bmatrix} f_u & 0 & 0 \\ 0 & f_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

In general, the camera coordinate system does not coincide with the world coordinate system and their relationship can be described by a rotation matrix and a translation vector, denoted by \mathbf{R}_c and \mathbf{t}_c respectively. Let \mathbf{M} be a 3D point in the world coordinate system and $\tilde{\mathbf{M}}$ its corresponding homogeneous representation. Under the pinhole camera model, its projection point $\tilde{\mathbf{m}}$ in the image is given by

$$\begin{aligned} \tilde{\mathbf{m}} &= \beta \mathbf{K}_c (\mathbf{R}_c \mathbf{M} + \mathbf{t}_c) \\ &= \beta \mathbf{K}_c [\mathbf{R}_c \quad \mathbf{t}_c] \tilde{\mathbf{M}} \end{aligned} \quad (2.3)$$

where β is a nonzero scale factor.

2.2.3 Projector Model

In general, a light projector can be treated as the dual of a camera, i.e. a projection device relating the 3D world and the 2D image. So the projector image formation can be approximated by a pin-hole projection model, similar to the pin-hole camera model. This means that the projector can be described by the following 3×3 matrix

$$\mathbf{K}_p = \begin{bmatrix} f'_u & s' & u'_0 \\ 0 & f'_v & v'_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

with f'_u , f'_v , s' , u'_0 and v'_0 defined similar to those in (2.1).

However, the assumption about a simplified camera model, that the principal point is close to the image center, is not valid for a projector since most projectors use an off-axis projection. For example, when they are set on a table or mounted upside-down on a ceiling, the image is projected through the upper or

Fig. 2.7 The optical characteristic of a projector

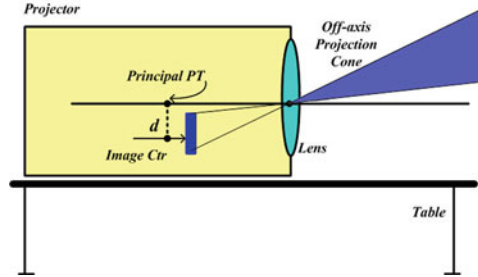
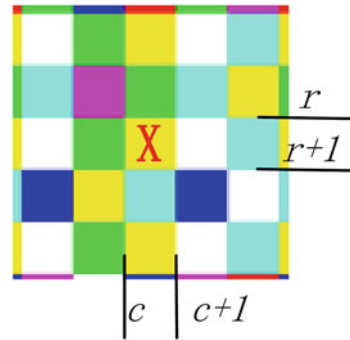


Fig. 2.8 Identification of light plane's index



the lower half of the lens, respectively [157]. As a result, the principal point is vertically shifted or even slightly outside the image. Therefore, we always assume that the projector has fixed optics, which means its intrinsic parameters are kept constant.

In the structured light system, the light pattern (to be discussed later) for the projector is designed as a color-encoded grid, which contains two sets of parallel lines perpendicular to each other. The trace of each line forms a plane in 3D space, named as a stripe light plane. So another straightforward model for the projector is to describe it by two sets of stripe light planes. To identify the index of each light plane, the coordinates of the grid associate with that plane are obtained first. For example, if the coordinates of the grid 'X' are (r, c) , the index of the upper horizontal plane associated with the grid is set to be r and that of the left vertical plane is c (Fig. 2.8 shows a portion of the light pattern). Then the indexes for other consecutive planes can be retrieved by simple plus and minus operations.

In summary, two different models for the DLP projector are established. The first model is used in Chap. 4 by treating the system as an active stereoscopic vision, while the second model is used in Chap. 5 by treating the system as a collection of light planes and a camera.

2.3 Pattern Coding Strategy

2.3.1 Introduction

Studies on the active vision techniques for computation of range data can be traced back to the early 1970s from the work of Shirai and his collaborators [138, 139]. The light pattern used in the vision system ranges from a single point and a single line segment to a more complicated pattern, such as a set of parallel stripes and orthogonal grids, etc. A review work on the recent development in coded structured light techniques can be found in [149]. Broadly speaking, they can be classified into two categories: static coding and dynamic coding techniques.

The static coding methods include binary codes [140] and Gray codes [141, 142], etc, where a sequence of patterns are used to generate the codeword for each pixel. This kind of techniques is often called time-multiplexing methods because the bits of the codeword are multiplexed in time. The advantage is that the pattern is simple and the resolution and accuracy of the reconstructed data can be very high. However, it is limited to static scenes with motionless objects, and hence termed as static coding.

On the other hand, the dynamic coding methods are developed based on De Bruijn sequences [143, 144] and M-arrays [145, 146], etc, in which the light pattern is encoded into a single shot. In general, spatial neighborhood strategy is employed so that the light pattern is divided into a certain number of regions, in which some information generates a different codeword. The major advantage is that such strategy permits a static as well as dynamic scene with moving or deformable objects.

In our work, we require that a single image should be sufficient for the calibration and 3D reconstruction, which implies that the light pattern for the DLP projector should be encoded into a single shot. Here, a dynamic coding method from Griffin [147, 148] is adopted. The algorithms can be summarized as follows.

2.3.2 Color-Encoded Light Pattern

Let $\omega = \{1, 2, \dots, \gamma\}$ be a set of color primitives (for example, 1 = red, 2 = green, 3 = blue, etc). Given these color primitives, a one-dimensional string V_{hp} is constructed such that each triplet of adjacent colors is distinct from every other triplet. This string is considered as the first row in the light pattern and its size is $\gamma^3 + 2$. Similarly, another string V_{vp} is constructed such that each pair of adjacent colors is distinct from every other pair and the size is $\gamma^2 + 1$. For the other rows in the pattern, modulo operation is iteratively performed with the preceding row and each element of the second string. Then we have a matrix for the light pattern whose size is $(\gamma^2 + 2) \times (\gamma^3 + 2)$. For example, if $\omega = \{1, 2, 3\}$ is taken, the following two vectors are obtained:

$$\begin{aligned} V_{hp} &= (331 \ 321 \ 311 \ 231 \ 221 \ 211 \ 133 \ 232 \ 223 \ 33) \\ V_{vp} &= (31 \ 21 \ 13 \ 22 \ 33) \end{aligned}$$

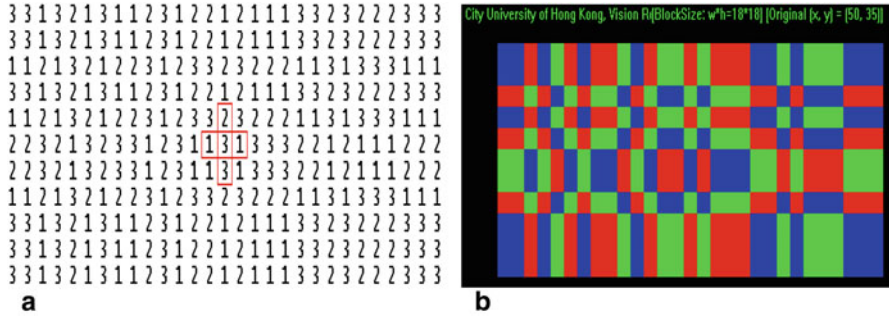


Fig. 2.9 Example of an encoded matrix (Three color primitives). **a** The pattern matrix. **b** A screen shot of the color-encoded light pattern

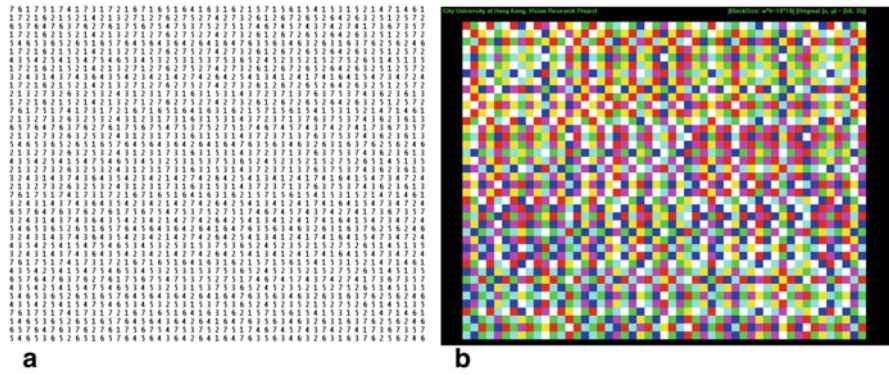


Fig. 2.10 Example of an encoded matrix (Seven color primitives). **a** The pattern matrix. **b** A screen shot of the color-encoded light pattern

Then, the first row of the pattern is $p_{0i} = V_{hpi}$. The rest of the matrix elements are calculated using

$$p_{ij} = (p_{(i-1)j} + V_{vpj}) \bmod 3 + 1 \quad (2.5)$$

Finally, a matrix with size 11×29 is obtained as shown in Fig. 2.9a, b gives a screen shot of the light pattern. Here, a codeword at location (i, j) is defined by the color primitive and its four neighbors (east, south, west and north). Griffin [147] had proved that all the codewords are unique in this pattern.

From this figure, we can see that some adjacent grids may have the same color and will not be distinguished from each other. So we use a codeword-based filtering procedure to discard those grids having the same color with their neighbors. In our system, we take $\omega = \{1, 2, \dots, 7\}$ where seven different colors, i.e. red, green, blue; white, cyan, magenta; yellow, are used. According to the formula, we can have a matrix of 51×345 . Among which, those grids who are distinct with their neighbors are selected. The final results are shown in Fig. 2.10. Here, 40×51 matrix is used.

Table 2.1 Look-up table for horizontal triplets

Triplet	Column index	Triplet	Column index	Triplet	Column index
(3,3,1)	2	(2,3,1)	11	(1,3,3)	20
(3,1,3)	3	(3,1,2)	12	(3,3,2)	21
(1,3,2)	4	(1,2,2)	13	(3,2,3)	22
(3,2,1)	5	(2,2,1)	14	(2,3,2)	23
(2,1,3)	6	(2,1,2)	15	(3,2,2)	24
(1,3,1)	7	(1,2,1)	16	(2,2,2)	25
(3,1,1)	8	(2,1,1)	17	(2,2,3)	26
(1,1,2)	9	(1,1,1)	18	(2,3,3)	27
(1,2,3)	10	(1,1,3)	19	(3,3,3)	28

Remark

1. It should be noted that rather than color dots used in Griffin [147], we use the color-encoded grid blocks. In our opinion, the grid blocks can be segmented more easily by edge detection. The encoded points are the intersection of these edges, so they can be found very accurately. When projecting dots, their mass centres must be located. When a dot appears only partially in the image or the surface of the object is somewhat bumpy, the mass centre will be incorrect. Obviously, our light pattern overcomes these drawbacks. Moreover, the grid techniques allow adjacent cross-points to be located by tracking the edges, but this is not applicable for the dot representation. These features not only decrease the complexity of image processing but also simplify the pattern decoding process.
2. There is a trade off between the resolution of the pattern, i.e. the number of color primitives used and the complexity of image processing. In general, the larger the number of color primitives used, the higher the resolution we will have and more details of the scene can be captured, but the more the noise sensitivity and hence the higher the complexity of image processing. Through our experiments, we find that a good balance can be obtained when seven color primitives are used. Anyway, an interpolation operation can provide an approximation of dense information when required.

2.3.3 Decoding the Light Pattern

Once the light pattern is projected on the concerning scene, an image is grabbed with the camera. The decoding problem involves finding the coordinates of the codewords extracted from the image. For simplicity, we take the three-color-primitive case in Fig. 2.9 as an example.

Before decoding, two tables are constructed. Table 2.1 consists of a list of each triplet in V_{hp} as well as the column position of the middle primitive of each triplet. Table 2.2 consists of a list of each pair in V_{vp} along with the row position of the first

Table 2.2 Look-up table for vertical pairs

Vertical pairs	(3,1)	(1,2)	(2,1)	(1,1)	(1,3)	(3,2)	(2,2)	(2,3)	(3,3)
Cumul. jumps	3	4	6	7	8	11	13	15	18
Row index	2	3	4	5	6	7	8	9	10

primitive of each pair and the cumulative jump. The cumulative jump value is defined as $c\delta_i = c\delta_{i-1} + \delta_i$, where δ_i is the first element value of the pair at position i .

After the preprocessing stage, the algorithm for decoding a codeword w_{ij} is presented as follows:

1. Determine the horizontal triplet of $w_{ij} : (h_1, h_2, h_3) = (w_{i,j-1}, w_{i,j}, w_{i,j+1})$;
2. Determine the vertical triplet of $w_{ij} : (v_1, v_2, v_3) = (w_{i-1,j}, w_{i,j}, w_{i+1,j})$;
3. Calculate the number of jumps (a, b) between the primitives:
 $a = (v_2 - v_1) \bmod \gamma$ and $b = (v_3 - v_2) \bmod \gamma$
then identify the row index i from Table 2.2 which corresponds to (a, b) and get the cumulative jump $c\delta$ for this value;
4. Determine the alias (a_1, a_2, a_3) by:

$$b\delta = (c\delta) \bmod \gamma$$

$$a_1 = (h_1 - b\delta) \bmod \gamma$$

$$a_2 = (h_2 - b\delta) \bmod \gamma$$

$$a_3 = (h_3 - b\delta) \bmod \gamma$$

Then identify the column index j by locating the alias in Table 2.1.

We can obtain the coordinate (i, j) for the codeword w_{ij} .

Note that in the above algorithm, if $(k - l) \leq 0$ then $(k - l) \bmod \gamma = k - l + \gamma$.

For example, we consider the codeword (3, 1, 2, 1, 2) labeled by red line in Fig. 2.9. For this codeword, the vertical triplet is (2, 3, 3), so the number of jumps (a, b) is (1, 3). From Table 2.2, the pair (1,3) matches the row index $i=6$ and the cumulative jumps $c\delta=8$. Therefore, $b\delta=2$. From the horizontal triplet (1, 3, 1), we get the vertical alias $(a_1, a_2, a_3)=(2, 1, 2)$. From Table 2.1, the column index is $j=15$. Consequently, the coordinates for the codeword is (6, 15). From the light pattern matrix, we can verify that it is correct.

2.4 Some Preliminaries

2.4.1 Notations and Definitions

I. Notations Π, Π_k, \dots , represent the projective planes: The image plane of the CCD camera is denoted by Π , and the k -th light stripe plane by Π_k .

$F_w, F_\Pi, F_{2\Pi_k}, F_{3\Pi_k}, \dots$, represent the coordinate frames. The subscript “2” specifies a 2D coordinate frame, and “3” defines a 3D coordinate frame. We denote the world coordinate frame by F_w , the image coordinate system by F_Π , and the coordinate frame of the k -th stripe light plane by $F_{2\Pi_k}$ and $F_{3\Pi_k}$.

The scale factor is denoted by normal face symbols, e.g. a . A vector represents a column of real numbers denoted by lowercase boldface letter, such as \mathbf{a} . If not specified, we mean it a 3×1 vector. A matrix is an array of real numbers denoted by uppercase boldface letter, such as \mathbf{A} . If not specified, we mean it a 3×3 matrix. Superscript T in Greek style represents the transpose of a vector or matrix.

The boldface letters $\mathbf{0}$, \mathbf{I} , \mathbf{R}_c and \mathbf{t}_c always denotes the zero matrix, identity matrix, rotation matrix and translation vector, respectively, while the notations \mathbf{M} and \mathbf{m} denote the 3D and 2D points.

The symbol $[*]_{\times}$ represents skew symmetric matrix of a vector $*$. For example, if $\mathbf{t}_c = (t_1 \ t_2 \ t_3)^T$ then $[\mathbf{t}_c]_{\times} = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$.

When necessary, further notation choices are described in each chapter.

II. Definitions In 2D Euclidean space, a 2-vector $\mathbf{m} = (x \ y)^T$ can be used to represent the inhomogeneous coordinates of a point \mathbf{m} . By adding a final coordinate with value 1, the 3-vector $\tilde{\mathbf{m}} = (x \ y \ 1)^T$ becomes a homogeneous representation of that point. On the other hand, if the 3-vector $(x_1 \ x_2 \ x_3)^T$ represents a 2D point, its inhomogeneous coordinates is given by $(\frac{x_1}{x_3} \ \frac{x_2}{x_3})^T$. In the homogeneous representation, if the final coordinate is zero or close to zero, e.g. $x_3 = 0$, the inhomogeneous coordinates will be infinity or close to infinity. Such point is called point at infinity. The line at infinity is such a line that only consists of the points at infinity. The 2D Euclidean space and the line at infinity make up a 2D projective space.

In the 2D projective space, the joint or cross product of two points gives a line in the same space. This line is called the line at infinity if the two points are both point at infinity. Dually, the intersection of any two lines provides a point. When the two lines are parallel to each other, this point is the point at infinity. It depends only on the direction of those lines, but not their positions. In vision community, the image of a line at infinity is named a vanishing line while the image of a point at infinity a vanishing point. Similar results can be obtained for 3D projective space which consists of 3D Euclidean space and the plane at infinity.

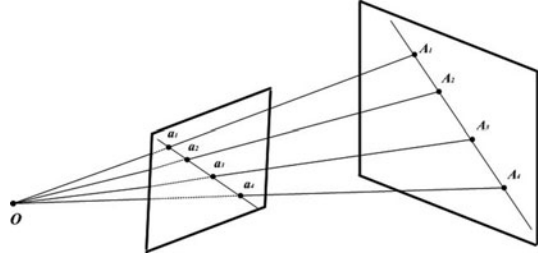
2.4.2 Cross Ratio

In definition, the cross ratio is a ratio of ratios of distances. Given four collinear points in 3D space, \mathbf{A}_1 , \mathbf{A}_2 , \mathbf{A}_3 and \mathbf{A}_4 , the cross ratio is expressed as

$$(\mathbf{A}_1, \mathbf{A}_2; \mathbf{A}_3, \mathbf{A}_4) = \frac{\overline{\mathbf{A}_1 \mathbf{A}_3}}{\overline{\mathbf{A}_2 \mathbf{A}_3}} \cdot \frac{\overline{\mathbf{A}_2 \mathbf{A}_4}}{\overline{\mathbf{A}_1 \mathbf{A}_4}} \quad (2.6)$$

where $\overline{\mathbf{A}_1 \mathbf{A}_3}$ denotes the length between \mathbf{A}_1 and \mathbf{A}_3 , etc.

Fig. 2.11 Invariance of cross ratio of four collinear points under perspective projection



If their correspondent points on the projective plane (such as CCD camera image plane) are a_1, a_2, a_3 and a_4 , according to the property of projective geometry, they are also collinear (Fig. 2.11). As cross ratio is invariant under projective transformation, we have the following equation:

$$(A_1, A_2; A_3, A_4) = (a_1, a_2; a_3, a_4) \quad (2.7)$$

The cross ratio is independent of the coordinate system established. Especially, if the points are parameterized by $\theta_{A_1}, \theta_{A_2}, \theta_{A_3}$ and θ_{A_4} , then

$$(A_1, A_2; A_3, A_4) = \frac{\theta_{A_1} - \theta_{A_3}}{\theta_{A_2} - \theta_{A_3}} \cdot \frac{\theta_{A_2} - \theta_{A_4}}{\theta_{A_3} - \theta_{A_4}} \quad (2.8)$$

Here, the cross ratio is defined in terms of collinear points. This definition can be extended to the pencil of lines and pencil of planes. With the tool of cross ratio, we can calculate a 3D point through its image point coordinate. This is very useful when calibrating the light planes as in Chap. 5.

2.4.3 Plane-Based Homography

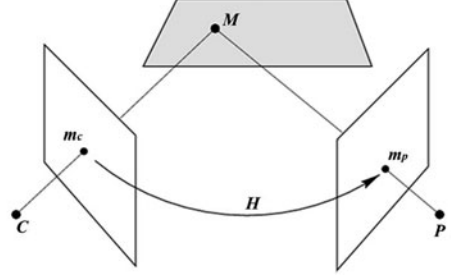
According to the projective geometry, the term plane-based Homography refers to the plane-to-plane transformation in the projective space, which maps a point on one plane to a point on the other. The Homography arises when a planar surface is imaged or two views of the planar surface are obtained. Figure 2.12 shows one of the cases, in which M represents a space point on the plane π , m_c , and m_p denote its corresponding projections.

Algebraically, the Homography can be described by a 3×3 non-singular matrix, e.g. $H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$. Under the perspective projection, the corresponding points \tilde{m}_p and \tilde{m}_c are related by

$$\tilde{m}_p = \lambda H \tilde{m}_c \quad (2.9)$$

where λ is a scale factor.

Fig. 2.12 An illustration for plane-based Homography



From (2.9), each pair of corresponding points provides two constraints on the Homography. Hence, four pairs are sufficient for the estimation. If more than four pairs are available, the Homography can be determined in the least squared sense. Let the i -th pair of points be $\tilde{\mathbf{m}}_{c,i} = [u_i \ v_i \ 1]^T$ and $\tilde{\mathbf{m}}_{p,i} = [u'_i \ v'_i \ 1]^T$. In the follows, we talk about two standard linear methods for solving it.

I. Non-Homogeneous Solution In non-homogeneous method, one of nine elements in the Homographic matrix is assumed to be a fixed value. Usually, we let $h_9 = 1$ and stack the remaining eight elements into a vector $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8)^T$.

For the i -th pair of points, we have

$$\begin{aligned} (u_i, v_i, 1, 0, 0, 0, -u'_i u_i, -u'_i v_i) \mathbf{h} &= u'_i \\ (0, 0, 0, u_i, v_i, 1, -v'_i u_i, -v'_i v_i) \mathbf{h} &= v'_i \end{aligned} \quad (2.10)$$

From (2.10), given $n(n \geq 4)$ pairs of correspondences, we can have the following matrix equation:

$$\mathbf{A} \mathbf{h} = \mathbf{b} \quad (2.11)$$

$$\text{where } \mathbf{A} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u'_1 u_1 & -u'_1 v_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -v'_1 u_1 & -v'_1 v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 & -u'_n u_n & -u'_n v_n \\ 0 & 0 & 0 & u_n & v_n & 1 & -v'_n u_n & -v'_n v_n \end{bmatrix}$$

and $\mathbf{b} = (u'_1 \ v'_1 \ \cdots \ u'_n \ v'_n)^T$.

Equation (2.11) is a standard linear equation system. There are many ways for solving it, such as Gaussian Elimination, LU decomposition and Jacobi iteration method. Once the vector \mathbf{h} is obtained, the Homographic matrix can be got by unstacking this vector.

II. Homogeneous Solution In this case, we can stack the Homography matrix into a 9-vector as $\mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)^T$.

For the i -th pair of points, we have

$$\begin{aligned} (u_i, v_i, 1, 0, 0, 0, -u'_i u_i, -u'_i v_i, u'_i) \mathbf{h} &= 0 \\ (0, 0, 0, u_i, v_i, 1, -v'_i u_i, -v'_i v_i, v'_i) \mathbf{h} &= 0 \end{aligned} \quad (2.12)$$

From (2.12), given $n(n \geq 4)$ pairs of correspondences, we can obtain the following matrix equation:

$$\mathbf{A} \mathbf{h} = \mathbf{0} \quad (2.13)$$

Where $\mathbf{A} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u'_1 u_1 & -u'_1 v_1 & u'_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -v'_1 u_1 & -v'_1 v_1 & v'_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 & -u'_n u_n & -u'_n v_n & u'_n \\ 0 & 0 & 0 & u_n & v_n & 1 & -v'_n u_n & -v'_n v_n & v'_n \end{bmatrix}$.

Let $\mathbf{Q} = \mathbf{A}^T \mathbf{A}$. Using eigenvalue decomposition, the solution for the vector \mathbf{h} can be determined by the eigenvector corresponding to the smallest eigenvalue of \mathbf{Q} .

Remark

1. The non-homogeneous solution has the disadvantage that poor estimation is obtained if the chosen element should actually have the value zero or close to zero. The homogeneous one overcomes this disadvantage.
2. Appropriate choice of the methods will provide convenience in different cases. For example, we will use the non-homogeneous solution when evaluating the computational complexity in Chap. 5, and the homogeneous solution will be used when doing error analysis in Chap. 4.

2.4.4 Fundamental Matrix

The homography matrix describes the mutual relationship among an arbitrary 3D planar patch, its images and the characteristic of the vision system, while the fundamental matrix encapsulates the intrinsic epipolar geometry. It is independent of the scene structure, and only depends on the camera's internal parameters and relative pose.

Mathematically, the fundamental matrix is a 3×3 matrix and the rank is 2. Since it is a singular matrix, there are many different parameterizations. For example, we can express one row (or column) of the fundamental matrix as the linear combination of the other two rows (or columns). As a result, there are many different approaches for estimating this matrix. We will next show a simple homogeneous solution.

Assuming an arbitrary point \mathbf{M} in the scene, the corresponding image pixels are denoted by \mathbf{m}_c and \mathbf{m}_p in Fig. 2.13, then we have

$$\mathbf{m}_c^T \mathbf{F} \mathbf{m}_p = 0 \quad (2.14)$$

where \mathbf{F} is called the fundamental matrix.

Fig. 2.13 An illustration for fundamental matrix

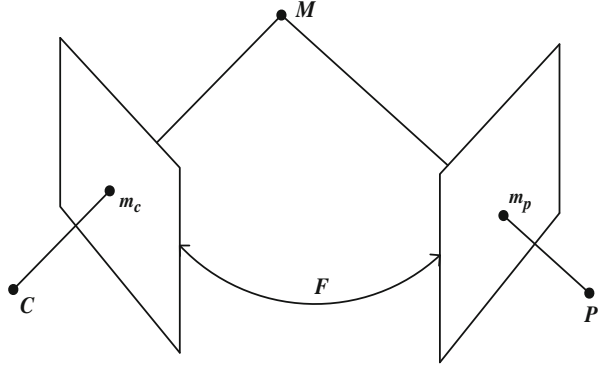
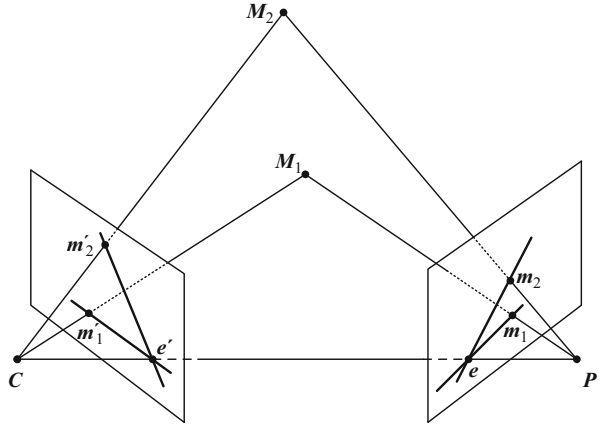


Fig. 2.14 Epipolar Geometry in a vision system



From (2.14), each point pair provides one constraint on the fundamental matrix.

Let $\mathbf{F} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}$. Its row-first vector is $\mathbf{f} = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9)^T$.

Let the coordinates of \mathbf{m}_c and \mathbf{m}_p be set as in the previous section. Rearranging (2.14), we have

$$[uu', uv', u, vu', vv', v, u', v', 1]\mathbf{f} = 0 \quad (2.15)$$

Similarly, given $n(n \geq 8)$ pairs of correspondences, we can obtain the following matrix equation:

$$\mathbf{A}\mathbf{f} = \mathbf{0} \quad (2.16)$$

where \mathbf{A} represents the coefficient matrix.

The solution for the vector \mathbf{f} can be determined up to a scale factor using eigenvalue decomposition. And so is the fundamental matrix \mathbf{F} . The solution can be

improved by nonlinear optimization with the following energy function:

$$E = \sum_i (d^2(\mathbf{m}_{ci}, \mathbf{F}\mathbf{m}_{pi}) + d^2(\mathbf{m}_{pi}, \mathbf{F}^T\mathbf{m}_{ci})) \quad (2.17)$$

Remark The fundamental matrix describes the mutual relationship between any two images of the same scene. It has found various applications in computer vision. For example, given the projection of a scene point into one of the images the corresponding point in the other image is constrained to a line, helping the search of feature correspondences. This is generally termed as epipolar geometry (Fig. 2.14). With the fundamental matrix a projective reconstruction can be immediately obtained. We will discuss its use in a catadioptric camera system in Chap. 6.

Automatic Calibration and Reconstruction for Active
Vision Systems

Zhang, B.; Li, Y.F.

2012, X, 166 p., Hardcover

ISBN: 978-94-007-2653-6