7

Computation of the Camera Matrix P

This chapter describes numerical methods for estimating the camera projection matrix from corresponding 3-space and image entities. This computation of the camera matrix is known as *resectioning*. The simplest such correspondence is that between a 3D point $\mathbf{X}$ and its image $\mathbf{x}$ under the unknown camera mapping. Given sufficiently many correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ the camera matrix $\mathtt{P}$ may be determined. Similarly, $\mathtt{P}$ may be determined from sufficiently many corresponding world and image lines.

If additional constraints apply to the matrix $\mathtt{P}$, such as that the pixels are square, then a *restricted* camera matrix subject to these constraints may be estimated from world to image correspondences.

Throughout this book it is assumed that the map from 3-space to the image is linear. This assumption is invalid if there is lens distortion. The topic of radial lens distortion correction is dealt with in this chapter.

The internal parameters $\mathtt{K}$ of the camera may be extracted from the matrix $\mathtt{P}$ by the decomposition of section 6.2.4. Alternatively, the internal parameters can be computed directly, without necessitating estimating $\mathtt{P}$, by the methods of chapter 8.

7.1 Basic equations

We assume a number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$ are given. We are required to find a camera matrix $\mathtt{P}$, namely a 3×4 matrix such that $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$ for all i. The similarity of this problem with that of computing a 2D projective transformation $\mathtt{H}$, treated in chapter 4, is evident. The only difference is the dimension of the problem. In the 2D case the matrix $\mathtt{H}$ has dimension 3×3, whereas in the present case, $\mathtt{P}$ is a 3×4 matrix. As one may expect, much of the material from chapter 4 applies almost unchanged to the present case.

As in section 4.1(*p*88) for each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ we derive a relationship

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \\ -y_i\mathbf{X}_i^\mathsf{T} & x_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}. \tag{7.1}$$

where each $\mathbf{P}^{i\mathsf{T}}$ is a 4-vector, the i-th row of $\mathtt{P}$. Alternatively, one may choose to use

only the first two equations:

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \tag{7.2}$$

since the three equations of (7.1) are linearly dependent. From a set of n point correspondences, we obtain a $2n \times 12$ matrix $\mathtt{A}$ by stacking up the equations (7.2) for each correspondence. The projection matrix $\mathtt{P}$ is computed by solving the set of equations $\mathtt{A}\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ is the vector containing the entries of the matrix $\mathtt{P}$.

Minimal solution. Since the matrix $\mathtt{P}$ has 12 entries, and (ignoring scale) 11 degrees of freedom, it is necessary to have 11 equations to solve for $\mathtt{P}$. Since each point correspondence leads to two equations, at a minimum $5\frac{1}{2}$ such correspondences are required to solve for $\mathtt{P}$. The $\frac{1}{2}$ indicates that only one of the equations is used from the sixth point, so one needs only to know the x-coordinate (or alternatively the y-coordinate) of the sixth image point.

Given this minimum number of correspondences, the solution is exact, i.e. the space points are projected exactly onto their measured images. The solution is obtained by solving $\mathtt{A}\mathbf{p} = \mathbf{0}$ where $\mathtt{A}$ is an 11×12 matrix in this case. In general $\mathtt{A}$ will have rank 11, and the solution vector $\mathbf{p}$ is the 1-dimensional right null-space of $\mathtt{A}$.

Over-determined solution. If the data is not exact, because of noise in the point coordinates, and $n \geq 6$ point correspondences are given, then there will not be an exact solution to the equations $\mathtt{A}\mathbf{p} = \mathbf{0}$. As in the estimation of a homography a solution for $\mathtt{P}$ may be obtained by minimizing an algebraic or geometric error.

In the case of algebraic error the approach is to minimize $\|\mathtt{A}\mathbf{p}\|$ subject to some normalization constraint. Possible constraints are

(i) $\|\mathbf{p}\| = 1$;
(ii) $\|\hat{\mathbf{p}}^3\| = 1$, where $\hat{\mathbf{p}}^3$ is the vector $(p_{31}, p_{32}, p_{33})^\mathsf{T}$, namely the first three entries in the last row of $\mathtt{P}$.

The first of these is preferred for routine use and will be used for the moment. We will return to the second normalization constraint in section 7.2.1. In either case, the residual $\mathtt{A}\mathbf{p}$ is known as the *algebraic error*. Using these equations, the complete DLT algorithm for computation of the camera matrix $\mathtt{P}$ proceeds in the same manner as that for $\mathtt{H}$ given in algorithm 4.1(*p*91).

Degenerate configurations. Analysis of the degenerate configurations for estimation of $\mathtt{P}$ is rather more involved than in the case of the 2D homography. There are two types of configurations in which ambiguous solutions exist for $\mathtt{P}$. These configurations will be investigated in detail in chapter 22. The most important critical configurations are as follows:

(i) The camera and points all lie on a twisted cubic.

(ii) The points all lie on the union of a plane and a single straight line containing the camera centre.

For such configurations, the camera cannot be obtained uniquely from the images of the points. Instead, it may move arbitrarily along the twisted cubic, or straight line respectively. If data is close to a degenerate configuration then a poor estimate for P is obtained. For example, if the camera is distant from a scene with low relief, such as a near-nadir aerial view, then this situation is close to the planar degeneracy.

Data normalization. It is important to carry out some sort of data normalization just as in the 2D homography estimation case. The points $\mathbf{x}_i$ in the image are appropriately normalized in the same way as before. Namely the points should be translated so that their centroid is at the origin, and scaled so that their RMS (root-mean-squared) distance from the origin is $\sqrt{2}$. What normalization should be applied to the 3D points $\mathbf{X}_i$ is a little more problematical. In the case where the variation in depth of the points from the camera is relatively slight it makes sense to carry out the same sort of normalization. Thus, the centroid of the points is translated to the origin, and their coordinates are scaled so that the RMS distance from the origin is $\sqrt{3}$ (so that the "average" point has coordinates of magnitude $(1, 1, 1, 1)^\mathsf{T}$). This approach is suitable for a compact distribution of points, such as those on the calibration object of figure 7.1.

In the case where there are some points that lie at a great distance from the camera, the previous normalization technique does not work well. For instance, if there are points close to the camera, as well as points that lie at infinity (which are imaged as vanishing points) or close to infinity, as may occur in oblique views of terrain, then it is not possible or reasonable to translate the points so that their centroid is at the origin. The normalization method described in exercise (iii) on page 128 would be more appropriately used in such a case, though this has not been thoroughly tested.

With appropriate normalization the estimate of P is carried out in the same manner as algorithm 4.2(*p*109) for H.

Line correspondences. It is a simple matter to extend the DLT algorithm to take account of line correspondences as well. A line in 3D may be represented by two points $\mathbf{X}_0$ and $\mathbf{X}_1$ through which the line passes. Now, according to result 8.2(*p*197) the plane formed by back-projecting from the image line $\mathbf{l}$ is equal to $\mathtt{P}^\mathsf{T}\mathbf{l}$. The condition that the point $\mathbf{X}_j$ lies on this plane is then

$$\mathbf{l}^\mathsf{T}\mathtt{P}\mathbf{X}_j = 0 \quad \text{for } j = 0, 1. \tag{7.3}$$

Each choice of j gives a single linear equation in the entries of the matrix P, so two equations are obtained for each 3D to 2D line correspondence. These equations, being linear in the entries of P, may be added to the equations (7.1) obtained from point correspondences and a solution to the composite equation set may be computed.

7.2 Geometric error

As in the case of 2D homographies (chapter 4), one may define geometric error. Suppose for the moment that world points $\mathbf{X}_i$ are known far more accurately than the

Objective

Given $n \geq 6$ world to image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood estimate of the camera projection matrix $\mathtt{P}$, i.e. the $\mathtt{P}$ which minimizes $\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$.

Algorithm

(i) **Linear solution.** Compute an initial estimate of $\mathtt{P}$ using a linear method such as algorithm 4.2(p109):

  (a) **Normalization:** Use a similarity transformation $\mathtt{T}$ to normalize the image points, and a second similarity transformation $\mathtt{U}$ to normalize the space points. Suppose the normalized image points are $\tilde{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$, and the normalized space points are $\tilde{\mathbf{X}}_i = \mathtt{U}\mathbf{X}_i$.

  (b) **DLT:** Form the $2n \times 12$ matrix $\mathtt{A}$ by stacking the equations (7.2) generated by each correspondence $\tilde{\mathbf{X}}_i \leftrightarrow \tilde{\mathbf{x}}_i$. Write $\mathbf{p}$ for the vector containing the entries of the matrix $\tilde{\mathtt{P}}$. A solution of $\mathtt{A}\mathbf{p} = \mathbf{0}$, subject to $\|\mathbf{p}\| = 1$, is obtained from the unit singular vector of $\mathtt{A}$ corresponding to the smallest singular value.

(ii) **Minimize geometric error.** Using the linear estimate as a starting point minimize the geometric error (7.4):

$$\sum_i d(\tilde{\mathbf{x}}_i, \tilde{\mathtt{P}}\tilde{\mathbf{X}}_i)^2$$

over $\tilde{\mathtt{P}}$, using an iterative algorithm such as Levenberg–Marquardt.

(iii) **Denormalization.** The camera matrix for the original (unnormalized) coordinates is obtained from $\tilde{\mathtt{P}}$ as

$$\mathtt{P} = \mathtt{T}^{-1}\tilde{\mathtt{P}}\mathtt{U}.$$

Algorithm 7.1. *The Gold Standard algorithm for estimating* $\mathtt{P}$ *from world to image point correspondences in the case that the world points are very accurately known.*

measured image points. For example the points $\mathbf{X}_i$ might arise from an accurately machined calibration object. Then the geometric error in the image is

$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$$

where $\mathbf{x}_i$ is the measured point and $\hat{\mathbf{x}}_i$ is the point $\mathtt{P}\mathbf{X}_i$, i.e. the point which is the exact image of $\mathbf{X}_i$ under $\mathtt{P}$. If the measurement errors are Gaussian then the solution of

$$\min_{\mathtt{P}} \sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2 \tag{7.4}$$

is the Maximum Likelihood estimate of $\mathtt{P}$.

Just as in the 2D homography case, minimizing geometric error requires the use of iterative techniques, such as Levenberg–Marquardt. A parametrization of $\mathtt{P}$ is required, and the vector of matrix elements $\mathbf{p}$ provides this. The DLT solution, or a minimal solution, may be used as a starting point for the iterative minimization. The complete Gold Standard algorithm is summarized in algorithm 7.1.

Example 7.1. Camera estimation from a calibration object

We will compare the DLT algorithm with the Gold Standard algorithm 7.1 for data

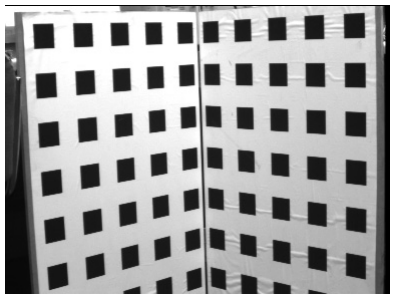

Fig. 7.1. *An image of a typical calibration object. The black and white checkerboard pattern (a "Tsai grid") is designed to enable the positions of the corners of the imaged squares to be obtained to high accuracy. A total of 197 points were identified and used to calibrate the camera in the examples of this chapter.*

	f_y	f_x/f_y	skew	x_0	y_0	residual
linear	1673.3	1.0063	1.39	379.96	305.78	0.365
iterative	1675.5	1.0063	1.43	379.79	305.25	0.364

Table 7.1. *DLT and Gold Standard calibration.*

from the calibration object shown in figure 7.1. The image points $\mathbf{x}_i$ are obtained from the calibration object using the following steps:

(i) Canny edge detection [Canny-86].
(ii) Straight line fitting to the detected linked edges.
(iii) Intersecting the lines to obtain the imaged corners.

If sufficient care is taken the points $\mathbf{x}_i$ are obtained to a localization accuracy of far better than $1/10$ of a pixel. A rule of thumb is that for a good estimation the number of constraints (point measurements) should exceed the number of unknowns (the 11 camera parameters) by a factor of five. This means that at least 28 points should be used.

Table 7.1 shows the calibration results obtained by using the linear DLT method and the Gold Standard method. Note that the improvement achieved using the Gold Standard algorithm is very slight. The difference of residual of one thousandth of a pixel is insignificant. △

Errors in the world points

It may be the case that world points are not measured with "infinite" accuracy. In this case one may choose to estimate P by minimizing a 3D geometric error, or an image geometric error, or both.

If only errors in the world points are considered then the 3D geometric error is defined as

$$\sum_i d(\mathbf{X}_i, \widehat{\mathbf{X}}_i)^2$$

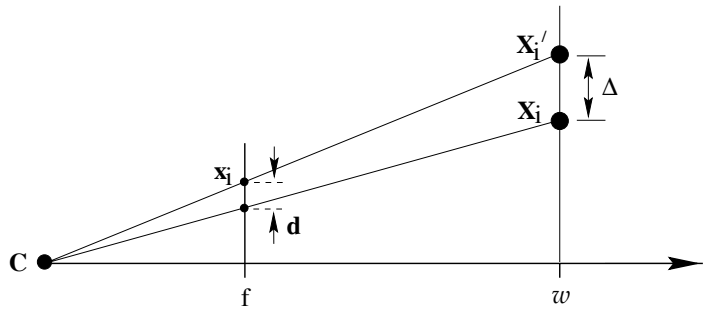

Fig. 7.2. *The DLT algorithm minimizes the sum of squares of geometric distance Δ between the point $\mathbf{X}_i$ and the point $\mathbf{X}_i'$ mapping exactly onto $\mathbf{x}_i$ and lying in the plane through $\mathbf{X}_i$ parallel to the principal plane of the camera. A short calculation shows that $wd = f\Delta$.*

where $\widehat{\mathbf{X}}_i$ is the closest point in space to $\mathbf{X}_i$ that maps exactly onto $\mathbf{x}_i$ via $\mathbf{x}_i = \mathtt{P}\widehat{\mathbf{X}}_i$.

More generally, if errors in both the world and image points are considered, then a weighted sum of world and image errors is minimized. As in the 2D homography case, this requires that one augment the set of parameters by including parameters $\widehat{\mathbf{X}}_i$, the estimated 3D points. One minimizes

$$\sum_{i=1}^{n} d_{\text{Mah}}(\mathbf{x}_i, \mathtt{P}\widehat{\mathbf{X}}_i)^2 + d_{\text{Mah}}(\mathbf{X}_i, \widehat{\mathbf{X}}_i)^2$$

where d_{Mah} represents Mahalanobis distance with respect to the known error covariance matrices for each of the measurements $\mathbf{x}_i$ and $\mathbf{X}_i$. In the simplest case, the Mahalanobis distance is simply a weighted geometric distance, where the weights are chosen to reflect the relative accuracy of measurements of the image and 3D points, and also the fact that image and world points are typically measured in different units.

7.2.1 Geometric interpretation of algebraic error

Suppose all the points $\mathbf{X}_i$ in the DLT algorithm are normalized such that $\mathbf{X}_i = (\mathrm{X}_i, \mathrm{Y}_i, \mathrm{Z}_i, 1)^\mathsf{T}$, and $\mathbf{x}_i = (x_i, y_i, 1)^\mathsf{T}$. In this case, it was seen in section 4.2.4-(*p*95) that the quantity being minimized by the DLT algorithm is $\sum_i (\hat{w}_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i))^2$, where $\hat{w}_i(\hat{x}_i, \hat{y}_i, 1)^\mathsf{T} = \mathtt{P}\mathbf{X}_i$. However, according to (6.15–*p*162),

$$\hat{w}_i = \pm \|\hat{\mathbf{p}}^3\| \ \text{depth}(\mathbf{X}; \mathtt{P}) \ .$$

Thus, the value $\hat{w}_i$ may be interpreted as the depth of the point $\mathbf{X}_i$ from the camera in the direction along the principal ray, provided the camera is normalized so that $\|\hat{\mathbf{p}}^3\|^2 = p_{31}^2 + p_{32}^2 + p_{33}^2 = 1$. Referring to figure 7.2 one sees that $\hat{w}_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)$ is proportional to $f d(\mathbf{X}', \mathbf{X})$, where f is the focal length and $\mathbf{X}_i'$ is a point mapping to $\mathbf{x}_i$ and lying in a plane through $\mathbf{X}_i$ parallel to the principal plane of the camera. Thus, the algebraic error being minimized is equal to $f \sum_i d(\mathbf{X}_i, \mathbf{X}_i')^2$.

The distance $d(\mathbf{X}_i, \mathbf{X}_i')$ is the correction that needs to be made to the measured 3D points in order to correspond precisely with the measured image points $\mathbf{x}_i$. The restriction is that the correction must be made in the direction perpendicular to the principal axis of the camera. Because of this restriction, the point $\mathbf{X}_i'$ is not the same as the closest point $\widehat{\mathbf{X}}_i$ to $\mathbf{X}_i$ that maps to $\mathbf{x}_i$. However, for points $\mathbf{X}_i$ not too far from the principal

ray of the camera, the distance $d(\mathbf{X}_i, \mathbf{X}'_i)$ is a reasonable approximation to the distance $d(\mathbf{X}_i, \widehat{\mathbf{X}}_i)$. The DLT slightly weights the points farther away from the principal ray by minimizing the squared sum of $d(\mathbf{X}_i, \mathbf{X}'_i)$, which is slightly larger than $d(\mathbf{X}_i, \widehat{\mathbf{X}}_i)$. In addition, the presence of the focal length f in the expression for algebraic error suggests that the DLT algorithm will be biased towards minimizing focal length at a cost of a slight increase in 3D geometric error.

Transformation invariance. We have just seen that by minimizing $\|\mathtt{A}\mathbf{p}\|$ subject to the constraint $\|\hat{\mathbf{p}}^3\| = 1$ one may interpret the solution in terms of minimizing 3D geometric distances. Such an interpretation is not affected by similarity transformations in either 3D space or the image space. Thus, one is led to expect that carrying out translation and scaling of the data, either in the image or in 3D point coordinates, will not have any effect on the solutions. This is indeed the case as may be shown using the arguments of section 4.4.2(*p*105).

7.2.2 Estimation of an affine camera

The methods developed above for the projective cameras can be applied directly to affine cameras. An affine camera is one for which the projection matrix has last row $(0, 0, 0, 1)$. In the DLT estimation of the camera in this case one minimizes $\|\mathtt{A}\mathbf{p}\|$ subject to this condition on the last row of $\mathtt{P}$. As in the case of computing 2D affine transformations, for affine cameras, algebraic error and geometric image error are equal. This means that geometric image distances may be minimized by a linear algorithm.

Suppose as above that all the points $\mathbf{X}_i$ are normalized such that $\mathbf{X}_i = (\mathrm{X}_i, \mathrm{Y}_i, \mathrm{Z}_i, 1)^\mathsf{T}$, and $\mathbf{x}_i = (x_i, y_i, 1)^\mathsf{T}$, and also that the last row of $\mathtt{P}$ has the affine form. Then (7.2) for a single correspondence reduces to

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -\mathbf{X}_i^\mathsf{T} \\ \mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \end{pmatrix} + \begin{pmatrix} y_i \\ -x_i \end{pmatrix} = \mathbf{0} \tag{7.5}$$

which shows that the squared algebraic error in this case equals the squared geometric error

$$\|\mathtt{A}\mathbf{p}\|^2 = \sum_i \left(x_i - \mathbf{P}^{1\mathsf{T}}\mathbf{X}_i\right)^2 + \left(y_i - \mathbf{P}^{2\mathsf{T}}\mathbf{X}_i\right)^2 = \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2.$$

This result may also be seen geometrically by comparison of figure 6.8(*p*170) and figure 7.2.

A linear estimation algorithm for an affine camera which minimizes geometric error is given in algorithm 7.2. Under the assumption of Gaussian measurement errors this is the Maximum Likelihood estimate of $\mathtt{P}_\mathrm{A}$.

7.3 Restricted camera estimation

The DLT algorithm, as it has so far been described, computes a general projective camera matrix $\mathtt{P}$ from a set of 3D to 2D point correspondences. The matrix $\mathtt{P}$ with centre at a finite point may be decomposed as $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathtt{C}}]$ where $\mathtt{R}$ is a 3×3

Objective

Given $n \geq 4$ world to image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood Estimate of the affine camera projection matrix $\mathtt{P}_\mathrm{A}$, i.e. the camera $\mathtt{P}$ which minimizes $\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$ subject to the affine constraint $\mathbf{P}^{3\mathsf{T}} = (0, 0, 0, 1)$.

Algorithm

(i) **Normalization:** Use a similarity transformation $\mathtt{T}$ to normalize the image points, and a second similarity transformation $\mathtt{U}$ to normalize the space points. Suppose the normalized image points are $\tilde{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$, and the normalized space points are $\tilde{\mathbf{X}}_i = \mathtt{U}\mathbf{X}_i$, with unit last component.

(ii) Each correspondence $\tilde{\mathbf{X}}_i \leftrightarrow \tilde{\mathbf{x}}_i$ contributes (from (7.5)) equations

$$\begin{bmatrix} \tilde{\mathbf{X}}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ \mathbf{0}^\mathsf{T} & \tilde{\mathbf{X}}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{P}}^1 \\ \tilde{\mathbf{P}}^2 \end{pmatrix} = \begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \end{pmatrix}$$

which are stacked into a $2n \times 8$ matrix equation $\mathtt{A}_8\mathbf{p}_8 = \mathbf{b}$, where $\mathbf{p}_8$ is the 8-vector containing the first two rows of $\tilde{\mathtt{P}}_\mathrm{A}$.

(iii) The solution is obtained by the pseudo-inverse of $\mathtt{A}_8$ (see section A5.2(*p*590))

$$\mathbf{p}_8 = \mathtt{A}_8^+\mathbf{b}$$

and $\tilde{\mathbf{P}}^{3\mathsf{T}} = (0, 0, 0, 1)$.

(iv) **Denormalization:** The camera matrix for the original (unnormalized) coordinates is obtained from $\tilde{\mathtt{P}}_\mathrm{A}$ as

$$\mathtt{P}_\mathrm{A} = \mathtt{T}^{-1}\tilde{\mathtt{P}}_\mathrm{A}\mathtt{U}$$

Algorithm 7.2. *The Gold Standard Algorithm for estimating an affine camera matrix* $\mathtt{P}_\mathrm{A}$ *from world to image correspondences.*

rotation matrix and $\mathtt{K}$ has the form (6.10–*p*157):

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} . \tag{7.6}$$

The non-zero entries of $\mathtt{K}$ are geometrically meaningful quantities, the internal calibration parameters of $\mathtt{P}$. One may wish to find the best-fit camera matrix $\mathtt{P}$ subject to restrictive conditions on the camera parameters. Common assumptions are

(i) The skew s is zero.
(ii) The pixels are square: $\alpha_x = \alpha_y$.
(iii) The principal point (x_0, y_0) is known.
(iv) The complete camera calibration matrix $\mathtt{K}$ is known.

In some cases it is possible to estimate a restricted camera matrix with a linear algorithm (see the exercises at the end of the chapter).

As an example of restricted estimation, suppose that we wish to find the best pinhole camera model (that is projective camera with $s = 0$ and $\alpha_x = \alpha_y$) that fits a set of point measurements. This problem may be solved by minimizing either geometric or algebraic error, as will be discussed next.

Minimizing geometric error. To minimize geometric error, one selects a set of parameters that characterize the camera matrix to be computed. For instance, suppose we wish to enforce the constraints $s = 0$ and $\alpha_x = \alpha_y$. One can parametrize the camera matrix using the remaining 9 parameters. These are x_0, y_0, α, plus 6 parameters representing the orientation $\mathtt{R}$ and location $\widetilde{\mathbf{C}}$ of the camera. Let this set of parameters be denoted collectively by $\mathbf{q}$. The camera matrix $\mathtt{P}$ may then be explicitly computed in terms of the parameters.

The geometric error may then be minimized with respect to the set of parameters using iterative minimization (such as Levenberg–Marquardt). Note that in the case of minimization of image error only, the size of the minimization problem is $9 \times 2n$ (supposing 9 unknown camera parameters). In other words the LM minimization is minimizing a function $f : \mathbb{R}^9 \to \mathbb{R}^{2n}$. In the case of minimization of 3D and 2D error, the function f is from $\mathbb{R}^{3n+9} \to \mathbb{R}^{5n}$, since the 3D points must be included among the measurements and minimization also includes estimation of the true positions of the 3D points.

Minimizing algebraic error. It is possible to minimize algebraic error instead, in which case the iterative minimization problem becomes much smaller, as will be explained next. Consider the parametrization map taking a set of parameters $\mathbf{q}$ to the corresponding camera matrix $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathbf{C}}]$. Let this map be denoted by g. Effectively, one has a map $\mathbf{p} = g(\mathbf{q})$, where $\mathbf{p}$ is the vector of entries of the matrix $\mathtt{P}$. Minimizing algebraic error over all point matches is equivalent to minimizing $\|\mathtt{A}g(\mathbf{q})\|$.

The reduced measurement matrix. In general, the $2n \times 12$ matrix $\mathtt{A}$ may have a very large number of rows. It is possible to replace $\mathtt{A}$ by a square 12×12 matrix $\hat{\mathtt{A}}$ such that $\|\mathtt{A}\mathbf{p}\| = \mathbf{p}^\mathsf{T}\mathtt{A}^\mathsf{T}\mathtt{A}\mathbf{p} = \|\hat{\mathtt{A}}\mathbf{p}\|$ for any vector $\mathbf{p}$. Such a matrix $\hat{\mathtt{A}}$ is called a *reduced measurement matrix*. One way to do this is using the Singular Value Decomposition (SVD). Let $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}$ be the SVD of $\mathtt{A}$, and define $\hat{\mathtt{A}} = \mathtt{D}\mathtt{V}^\mathsf{T}$. Then

$$\mathtt{A}^\mathsf{T}\mathtt{A} = (\mathtt{V}\mathtt{D}\mathtt{U}^\mathsf{T})(\mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}) = (\mathtt{V}\mathtt{D})(\mathtt{D}\mathtt{V}^\mathsf{T}) = \hat{\mathtt{A}}^\mathsf{T}\hat{\mathtt{A}}$$

as required. Another way of obtaining $\hat{\mathtt{A}}$ is to use the QR decomposition $\mathtt{A} = \mathtt{Q}\hat{\mathtt{A}}$, where $\mathtt{Q}$ has orthogonal columns and $\hat{\mathtt{A}}$ is upper triangular and square.

Note that the mapping $\mathbf{q} \mapsto \hat{\mathtt{A}}g(\mathbf{q})$ is a mapping from $\mathbb{R}^9$ to $\mathbb{R}^{12}$. This is a simple parameter-minimization problem that may be solved using the Levenberg–Marquardt method. The important point to note is the following:

- *Given a set of n world to image correspondences, $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, the problem of finding a constrained camera matrix $\mathtt{P}$ that minimizes the sum of algebraic distances $\sum_i d_{alg}(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$ reduces to the minimization of a function $\mathbb{R}^9 \to \mathbb{R}^{12}$, independent of the number n of correspondences.*

Minimization of $\|\hat{\mathtt{A}}g(\mathbf{q})\|$ takes place over all values of the parameters $\mathbf{q}$. Note that if $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\widetilde{\mathbf{C}}]$ with $\mathtt{K}$ as in (7.6) then $\mathtt{P}$ satisfies the condition $p_{31}^2 + p_{32}^2 + p_{33}^2 = 1$, since these entries are the same as the last row of the rotation matrix $\mathtt{R}$. Thus, minimizing $\mathtt{A}g(\mathbf{q})$ will lead to a matrix $\mathtt{P}$ satisfying the constraints $s = 0$ and $\alpha_x = \alpha_y$ and scaled

such that $p_{31}^2 + p_{32}^2 + p_{33}^2 = 1$, and which in addition minimizes the algebraic error for all point correspondences.

Initialization. One way of finding camera parameters to initialize the iteration is as follows.

(i) Use a linear algorithm such as DLT to find an initial camera matrix.
(ii) Clamp fixed parameters to their desired values (for instance set $s = 0$ and set $\alpha_x = \alpha_y$ to the average of their values obtained using DLT).
(iii) Set variable parameters to their values obtained by decomposition of the initial camera matrix (see section 6.2.4).

Ideally, the assumed values of the fixed parameters will be close to the values obtained by the DLT. However, in practice this is not always the case. Then altering these parameters to their desired values results in an incorrect initial camera matrix that may lead to large residuals, and difficulty in converging. A method which works better in practice is to use *soft* constraints by adding extra terms to the cost function. Thus, for the case where $s = 0$ and $\alpha_x = \alpha_y$, one adds extra terms $ws^2 + w(\alpha_x - \alpha_y)^2$ to the cost function. In the case of geometric image error, the cost function becomes

$$\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2 + ws^2 + w(\alpha_x - \alpha_y)^2 \quad .$$

One begins with the values of the parameters estimated using the DLT. The weights begin with low values and are increased at each iteration of the estimation procedure. Thus, the values of s and the aspect ratio are drawn gently to their desired values. Finally they may be clamped to their desired values for a final estimation.

Exterior orientation. Suppose that all the internal parameters of the camera are known, then all that remains to be determined are the position and orientation (or *pose*) of the camera. This is the "exterior orientation" problem, which is important in the analysis of calibrated systems.

To compute the exterior orientation a configuration with accurately known position in a world coordinate frame is imaged. The pose of the camera is then sought. Such a situation arises in hand–eye calibration for robotic systems, where the position of the camera is required, and also in model-based recognition using alignment where the position of an object relative to the camera is required.

There are six parameters that must be determined, three for the orientation and three for the position. As each world to image point correspondence generates two constraints it would be expected that three points are sufficient. This is indeed the case, and the resulting non-linear equations have four solutions in general.

Experimental evaluation

Results of constrained estimation for the calibration grid of example 7.1 are given in table 7.2.

Both the algebraic and geometric minimization involve an iterative minimization

	f_y	f_x/f_y	skew	x_0	y_0	residual
algebraic	1633.4	1.0	0.0	371.21	293.63	0.601
geometric	1637.2	1.0	0.0	371.32	293.69	0.601

Table 7.2. *Calibration for a restricted camera matrix.*

over 9 parameters. However, the algebraic method is far quicker, since it minimizes only 12 errors, instead of $2n = 396$ in the geometric minimization. Note that fixing skew and aspect ratio has altered the values of the other parameters (compare table 7.1) and increased the residual.

Covariance estimation. The techniques of covariance estimation and propagation of the errors into an image may be handled in just the same way as in the 2D homography case (chapter 5). Similarly, the minimum expected residual error may be computed as in result 5.2(*p*136). Assuming that all errors are in the image measurements, the expected ML residual error is equal to

$$\epsilon_{\text{res}} = \sigma(1 - d/2n)^{1/2} \quad .$$

where d is the number of camera parameters being fitted (11 for a full pinhole camera model). This formula may also be used to estimate the accuracy of the point measurements, given a residual error. In the case of example 7.1 where $n = 197$ and $\epsilon_{\text{res}} = 0.365$ this results in a value of $\sigma = 0.37$. This value is greater than expected. The reason, as we will see later, lies in the camera model – we are ignoring radial distortion.

Example 7.2. Covariance ellipsoid for an estimated camera
Suppose that the camera is estimated using the Maximum Likelihood (Gold Standard) method, optimizing over a set of camera parameters. The estimated covariance of the point measurements can then be used to compute the covariance of the camera model by back-propagation, according to result 5.10(*p*142). This gives $\Sigma_{\text{camera}} = (\mathtt{J}^{\mathsf{T}}\Sigma^{-1}_{\text{points}}\mathtt{J})^{-1}$ where J is the Jacobian matrix of the measured points in terms of the camera parameters. Uncertainty in 3D world points may also be taken into account in this way. If the camera is parametrized in terms of meaningful parameters (such as camera position), then the variance of each parameter can be measured directly from the diagonal entries of the covariance matrix.

Knowing the covariance of the camera parameters, error bounds or ellipsoids can be computed. For instance, from the covariance matrix for all the parameters we may extract the subblock representing the 3×3 covariance matrix of the camera position, $\Sigma_{\mathbf{C}}$. A confidence ellipsoid for the camera centre is then defined by

$$(\mathbf{C} - \bar{\mathbf{C}})^{\mathsf{T}}\Sigma^{-1}_{\mathbf{C}}(\mathbf{C} - \bar{\mathbf{C}}) = k^2$$

where k^2 is computed from the inverse cumulative χ^2_n distribution in terms of the desired confidence level α: namely $k^2 = F_n^{-1}(\alpha)$ (see figure A2.1(*p*567)). Here n is the

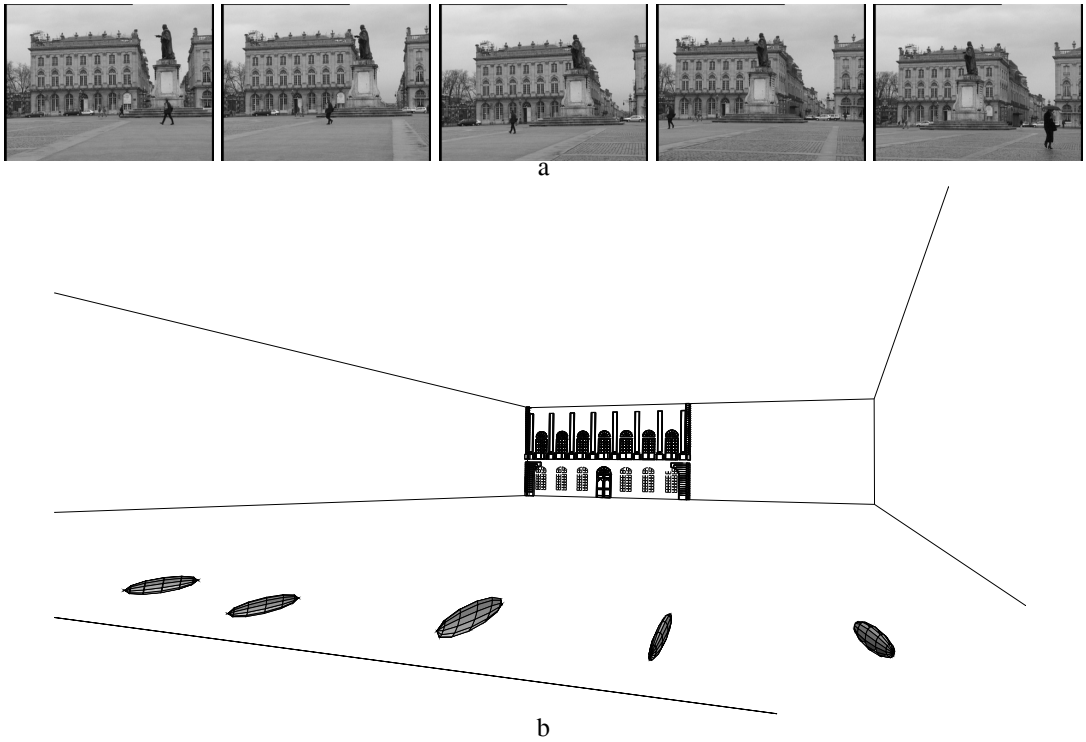

Fig. 7.3. **Camera centre covariance ellipsoids**. *(a) Five images of Stanislas square (Nancy, France), for which 3D calibration points are known. (b) Camera centre covariance ellipsoids corresponding to each image, computed for cameras estimated from the imaged calibration points. Note, the typical cigar shape of the ellipsoid aligned towards the scene data. Figure courtesy of Vincent Lepetit, Marie-Odile Berger and Gilles Simon.*

number of variables – that is 3 in the case of the camera centre. With the chosen level of certainty α, the camera centre lies inside the ellipsoid.

Figure 7.3 shows an example of ellipsoidal uncertainty regions for computed camera centres. Given the estimated covariance matrix for the computed camera, the techniques of section 5.2.6(p148) may be used to compute the uncertainty in the image positions of further 3D world points. △

7.4 Radial distortion

The assumption throughout these chapters has been that a linear model is an accurate model of the imaging process. Thus the world point, image point and optical centre are collinear, and world lines are imaged as lines and so on. For real (non-pinhole) lenses this assumption will not hold. The most important deviation is generally a radial distortion. In practice this error becomes more significant as the focal length (and price) of the lens decreases. See figure 7.4.

The cure for this distortion is to correct the image measurements to those that would have been obtained under a perfect linear camera action. The camera is then effectively again a linear device. This process is illustrated in figure 7.5. This correction must

a b

Fig. 7.4. *(a) Short vs (b) long focal lengths. Note the curved imaged lines at the periphery in (a) which are images of straight scene lines.*

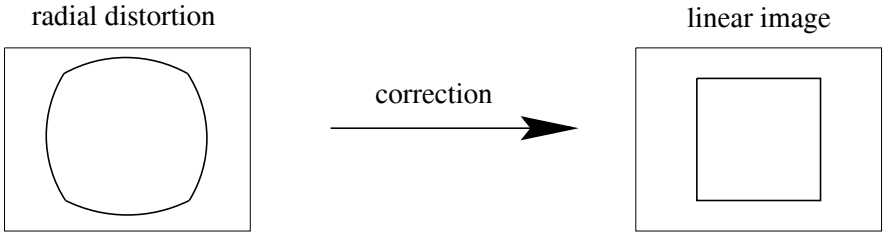

Fig. 7.5. *The image of a square with significant radial distortion is corrected to one that would have been obtained under a perfect linear lens.*

be carried out in the right place in the projection process. Lens distortion takes place during the initial projection of the world onto the image plane, according to (6.2–*p*154). Subsequently, the calibration matrix (7.6) reflects a choice of affine coordinates in the image, translating physical locations in the image plane to pixel coordinates.

We will denote the image coordinates of a point under ideal (non-distorted) pinhole projection by $(\tilde{x}, \tilde{y})$, measured in units of focal-length. Thus, for a point $\mathbf{X}$ we have (see (6.5–*p*155))

$$(\tilde{x}, \tilde{y}, 1)^{\mathsf{T}} = [\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\text{cam}}$$

where $\mathbf{X}_{\text{cam}}$ is the 3D point in camera coordinates, related to world coordinates by (6.6–*p*156). The actual projected point is related to the ideal point by a radial displacement. Thus, radial (lens) distortion is modelled as

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \tag{7.7}$$

where

- $(\tilde{x}, \tilde{y})$ is the ideal image position (which obeys linear projection).
- (x_d, y_d) is the actual image position, after radial distortion.
- $\tilde{r}$ is the radial distance $\sqrt{\tilde{x}^2 + \tilde{y}^2}$ from the centre for radial distortion.
- $L(\tilde{r})$ is a distortion factor, which is a function of the radius $\tilde{r}$ only.

Correction of distortion. In pixel coordinates the correction is written

$$\hat{x} = x_c + L(r)(x - x_c) \qquad \hat{y} = y_c + L(r)(y - y_c).$$

where (x, y) are the measured coordinates, $(\hat{x}, \hat{y})$ are the corrected coordinates, and (x_c, y_c) is the centre of radial distortion, with $r^2 = (x - x_c)^2 + (y - y_c)^2$. Note, if the aspect ratio is not unity then it is necessary to correct for this when computing r. With this correction the coordinates $(\hat{x}, \hat{y})$ are related to the coordinates of the 3D world point by a linear projective camera.

Choice of the distortion function and centre. The function $L(r)$ is only defined for positive values of r and $L(0) = 1$. An approximation to an arbitrary function $L(r)$ may be given by a Taylor expansion $L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \ldots$. The coefficients for radial correction $\{\kappa_1, \kappa_2, \kappa_3, \ldots, x_c, y_c\}$ are considered part of the interior calibration of the camera. The principal point is often used as the centre for radial distortion, though these need not coincide exactly. This correction, together with the camera calibration matrix, specifies the mapping from an image point to a ray in the camera coordinate system.

Computing the distortion function. The function $L(r)$ may be computed by minimizing a cost based on the deviation from a linear mapping. For example, algorithm 7.1(p181) estimates P by minimizing geometric image error for calibration objects such as the Tsai grids of figure 7.1. The distortion function may be included as part of the imaging process, and the parameters κ_i computed together with P during the iterative minimization of the geometric error. Similarly, the distortion function may be computed when estimating the homography between a single Tsai grid and its image.

A simple and more general approach is to determine $L(r)$ by the requirement that images of straight scene lines should be straight. A cost function is defined on the imaged lines (such as the distance between the line joining the imaged line's ends and its mid-point) after the corrective mapping by $L(r)$. This cost is iteratively minimized over the parameters κ_i of the distortion function and the centre of radial distortion. This is a very practical method for images of urban scenes since there are usually plenty of lines available. It has the advantage that no special calibration pattern is required as the scene provides the calibration entities.

Example 7.3. Radial correction. The function $L(r)$ is computed for the image of figure 7.6a by minimizing a cost based on the straightness of imaged scene lines. The image is 640×480 pixels and the correction and centre are computed as $\kappa_1 = 0.103689$, $\kappa_2 = 0.00487908$, $\kappa_3 = 0.00116894$, $\kappa_4 = 0.000841614$, $x_c = 321.87$, $y_c = 241.18$ pixels, where pixels are normalized by the average half-size of the image. This is a correction by 30 pixels at the image periphery. The result of warping the image is shown in figure 7.6b. △

Example 7.4. We continue with the example of the calibration grid shown in figure 7.1 and discussed in example 7.1(p181). Radial distortion was removed by the straight line

a b

Fig. 7.6. **Radial distortion correction.** *(a) The original image with lines which are straight in the world, but curved in the image. Several of these lines are annotated by dashed curves. (b) The image warped to remove the radial distortion. Note that the lines in the periphery of the image are now straight, but that the boundary of the image is curved.*

method, and then the camera calibrated using the methods described in this chapter. The results are given in table 7.3.

Note that the residuals after radial correction are substantially smaller. Estimation of the error of point measurements from the residual leads to a value of $\sigma = 0.18$ pixels. Since radial distortion involves selective stretching of the image, it is quite plausible that the effective focal length of the image is changed, as seen here. △

	f_y	f_x/f_y	skew	x_0	y_0	residual
linear	1580.5	1.0044	0.75	377.53	299.12	0.179
iterative	1580.7	1.0044	0.70	377.42	299.02	0.179
algebraic	1556.0	1.0000	0.00	372.42	291.86	0.381
iterative	1556.6	1.0000	0.00	372.41	291.86	0.380
linear	1673.3	1.0063	1.39	379.96	305.78	0.365
iterative	1675.5	1.0063	1.43	379.79	305.25	0.364
algebraic	1633.4	1.0000	0.00	371.21	293.63	0.601
iterative	1637.2	1.0000	0.00	371.32	293.69	0.601

Table 7.3. **Calibration with and without radial distortion correction.** *The results above the line are after radial correction – the results below for comparison are without radial distortion (from the previous tables). The upper two methods in each case solve for the general camera model, the lower two are for a constrained model with square pixels.*

In correcting for radial distortion, it is often not actually necessary to warp the image. Measurements can be made in the original image, for example the position of a corner feature, and the measurement simply mapped according to (7.7). The question of where features *should* be measured does not have an unambiguous answer. Warping the image will distort noise models (because of averaging) and may well introduce aliasing effects. For this reason feature detection on the unwarped image will often be preferable. However, feature grouping, such as linking edgels into straight line primitives,

is best performed after warping since thresholds on linearity may well be erroneously exceeded in the original image.

7.5 Closure

7.5.1 The literature

The original application of the DLT in [Sutherland-63] was for camera computation. Estimation by iterative minimization of geometric errors is a standard procedure of photogrammetrists, e.g. see [Slama-80].

A minimal solution for a calibrated camera (pose from the image of 3 points) was the original problem studied by Fischler and Bolles [Fischler-81] in their RANSAC paper. Solutions to this problem reoccur *often* in the literature; a good treatment is given in [Wolfe-91] and also [Haralick-91]. Quasi-linear solutions for one more than the minimum number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ are in [Quan-98] and [Triggs-99a].

Another class of methods, which are not covered here, is the iterative estimation of a projective camera starting from an affine one. The algorithm of "Model based pose in 25 lines of code" by Dementhon and Davis [Dementhon-95] is based on this idea. A similar method is used in [Christy-96].

Devernay and Faugeras [Devernay-95] introduced a straight line method for computing radial distortion into the computer vision literature. In the photogrammetry literature the method is known as "plumb line correction", see [Brown-71].

7.5.2 Notes and exercises

(i) Given 5 world-to-image point correspondences, $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, show that there are in general four solutions for a camera matrix `P` *with zero skew* that exactly maps the world to image points.

(ii) Given 3 world-to-image point correspondences, $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, show that there are in general four solutions for a camera matrix `P` *with known calibration* `K` that exactly maps the world to image points.

(iii) Find a linear algorithm for computing the camera matrix `P` under each of the following conditions:

- (a) The camera location (but not orientation) is known.
- (b) The direction of the principal ray of the camera is known.
- (c) The camera location and the principal ray of the camera are known.
- (d) The camera location and complete orientation of the camera are known.
- (e) The camera location and orientation are known, as well as some subset of the internal camera parameters (α_x, α_y, s, x_0 and y_0).

(iv) **Conflation of focal length and position on principal axis.** Compare the imaged position of a point of depth d before and after an increase in camera focal length Δf, or a displacement Δt_3 of the camera backwards along the principal axis. Let $(x, y)^\mathsf{T}$ and $(x', y')^\mathsf{T}$ be the image coordinates of the point before and

after the change. Following a similar derivation to that of (6.19–*p*169), show that

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + k \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

where $k^f = \Delta f/f$ for a focal length change, or $k^{t_3} = -\Delta t_3/d$ for a displacement (here skew $s = 0$, and $\alpha_x = \alpha_y = f$).

For a set of calibration points $\mathbf{X}_i$ with depth relief (Δ_i) small compared to the average depth (d_0),

$$k_i^{t_3} = -\Delta t_3/d_i = -\Delta t_3/(d_0 + \Delta_i) \approx -\Delta t_3/d_0$$

i.e. $k_i^{t_3}$ is approximately constant across the set. It follows that in calibrating from such a set, similar image residuals are obtained by changing the focal length k^f or displacing the camera k^{t_3}. Consequently, the estimated parameters of focal length and position on the principal axis are correlated.

(v) **Pushbroom camera computation.** The pushbroom camera, described in section 6.4.1, may also be computed using a DLT method. The x (orthographic) part of the projection matrix has 4 degrees of freedom which may be determined by four or more point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$; the y (perspective) part of the projection matrix has 7 degrees of freedom and may be determined from 7 correspondences. Hence, a minimal solution requires 7 points. Details are given in [Gupta-97].