20

Duality

It has been known since the work of Carlsson [Carlsson-95] and Weinshall *et al.* [Weinshall-95] that there is a dualization principle that allows one to interchange the role of points being viewed by several cameras and the camera centres themselves. In principle this implies the possibility of dualizing projective reconstruction algorithms to obtain new algorithms. In this chapter, this theme is developed to outline an explicit method for dualizing any projective reconstruction algorithm. At the practical implementation level, however, it is shown that there are difficulties which need to be overcome in order to allow application of this dualization method to produce working algorithms.

20.1 Carlsson–Weinshall duality

Let $\mathbf{E}_1 = (1,0,0,0)^\mathsf{T}$, $\mathbf{E}_2 = (0,1,0,0)^\mathsf{T}$, $\mathbf{E}_3 = (0,0,1,0)^\mathsf{T}$ and $\mathbf{E}_4 = (0,0,0,1)^\mathsf{T}$ form part of a projective basis for $\mathbb{P}^3$. Similarly, let $\mathbf{e}_1 = (1,0,0)^\mathsf{T}$, $\mathbf{e}_2 = (0,1,0)^\mathsf{T}$, $\mathbf{e}_3 = (0,0,1)^\mathsf{T}$ and $\mathbf{e}_4 = (1,1,1)^\mathsf{T}$ be a projective basis for the projective image plane $\mathbb{P}^2$.

Now, consider a camera with matrix $\mathtt{P}$. We assume that the camera centre $\mathbf{C}$ does not sit on any of the axial planes, that is none of the four coordinates of $\mathbf{C}$ is zero. In this case, no three of the points $\mathtt{P}\mathbf{E}_i$ for $i = 1, \ldots, 4$ are collinear in the image. Consequently, one may apply a projective transformation $\mathtt{H}$ to the image so that $\mathbf{e}_i = \mathtt{H}\mathtt{P}\mathbf{E}_i$. We assume that this has been done, and henceforth denote $\mathtt{H}\mathtt{P}$ simply by $\mathtt{P}$. Since $\mathtt{P}\mathbf{E}_i = \mathbf{e}_i$, the form of the matrix $\mathtt{P}$ is

$$\mathtt{P} = \begin{bmatrix} a & & & d \\ & b & & d \\ & & c & d \end{bmatrix}.$$

Definition 20.1. A camera matrix $\mathtt{P}$ is called a *reduced* camera matrix if it maps $\mathbf{E}_i$ to $\mathbf{e}_i$ for each $i = 1, \ldots, 4$. In other words $\mathbf{e}_i = \mathtt{P}\mathbf{E}_i$.

Now, for any point $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{T})^\mathsf{T}$ one verifies that

$$\mathtt{P} = \begin{bmatrix} a & & & d \\ & b & & d \\ & & c & d \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ \mathrm{T} \end{pmatrix} = \begin{pmatrix} a\mathrm{X} + d\mathrm{T} \\ b\mathrm{Y} + d\mathrm{T} \\ c\mathrm{Z} + d\mathrm{T} \end{pmatrix}. \tag{20.1}$$

Notice the symmetry in this equation between the entries of the camera matrix and the coordinates of the point. They may be interchanged as follows

$$\begin{bmatrix} a & & & d \\ & b & & d \\ & & c & d \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ \mathrm{T} \end{pmatrix} = \begin{bmatrix} \mathrm{X} & & & \mathrm{T} \\ & \mathrm{Y} & & \mathrm{T} \\ & & \mathrm{Z} & \mathrm{T} \end{bmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}. \tag{20.2}$$

The interchange of the roles of cameras and points may be interpreted as a form of duality, which will be referred to as Carlsson–Weinshall duality, or more briefly Carlsson duality. The consequences of this duality will be explored in the rest of this chapter.

20.1.1 Dual algorithms

First of all, we will use it for deriving a dual algorithm from a given projective reconstruction algorithm. Specifically, it will be shown that if one has an algorithm for doing projective reconstruction from n views of $m + 4$ points, then there is an algorithm for doing projective reconstruction from m views of $n + 4$ points. This result, observed by Carlsson [Carlsson-95], will be made specific by explicitly describing the steps of the dual algorithm.

We consider a projective reconstruction problem, which will be referred to as $\mathcal{P}(m, n)$. It is the problem of doing reconstruction from m views of n points. We denote image points by $\mathbf{x}^i_j$, which represents the image of the j-th object space point in the i-th view. Thus, the upper index indicates the view number, and the lower index represents the point number. Such a set of points $\{\mathbf{x}^i_j\}$ is called *realizable* if there are a set of camera matrices $\mathtt{P}^i$ and a set of 3D points $\mathbf{X}_j$ such that $\mathbf{x}^i_j = \mathtt{P}^i\mathbf{X}_j$. The projective reconstruction problem $\mathcal{P}(m, n)$ is that of finding such camera matrices $\mathtt{P}^i$ and points $\mathbf{X}_j$ given a realizable set $\{\mathbf{x}^i_j\}$ for m views of n points. The set of camera matrices and 3D points together are called a *realization* (or *projective realization*) of the set of point correspondences.

Let $\mathcal{A}(n, m + 4)$ represent an algorithm for solving the projective reconstruction problem $\mathcal{P}(n, m + 4)$. An algorithm will now be exhibited for solving the projective reconstruction $\mathcal{P}(m, n + 4)$. This algorithm will be denoted $\mathcal{A}^*(m, n + 4)$, the dual of the algorithm $\mathcal{A}(n, m + 4)$.

Initially, the steps of the algorithm will be given without proof. In addition, difficulties will be glossed over so as to give the general idea without getting bogged down in details. In the description of this algorithm it is important to keep track of the range of the indices, and whether they index the cameras or the points. Thus, the following may help to keep track.

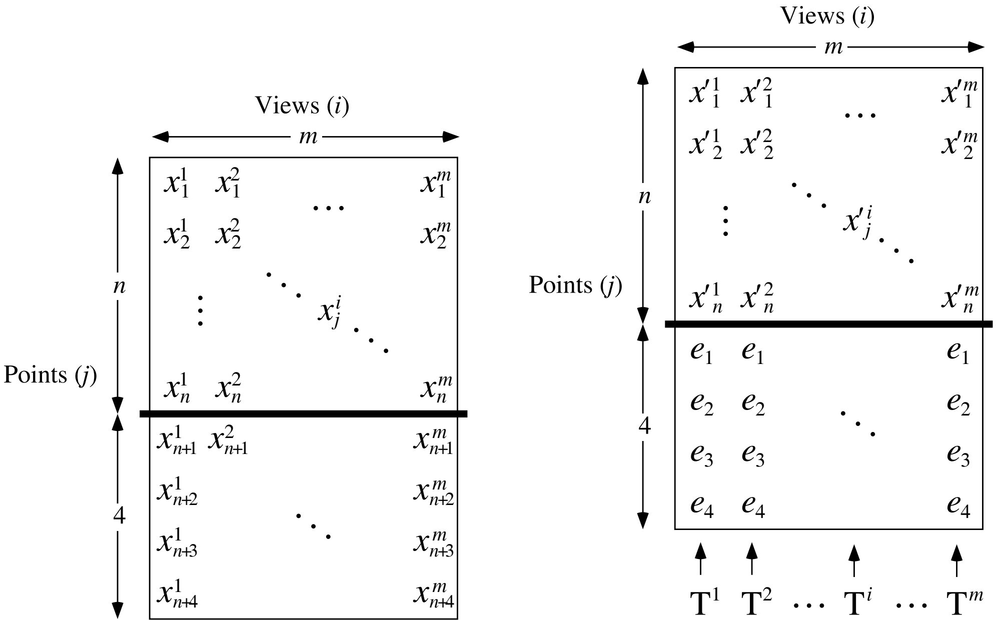

Fig. 20.1. **Left:** *Input to algorithm* $\mathcal{A}^*(m, n+4)$. **Right:** *Input data after transformation.*

- Upper indices represent the view number.
- Lower indices represent the point number.
- i ranges from 1 to m.
- j ranges from 1 to n.
- k ranges from 1 to 4.

The dual algorithm

Given an algorithm $\mathcal{A}(n, m+4)$ the goal is to exhibit a dual algorithm $\mathcal{A}^*(m, n+4)$.

Input:

The input to the algorithm $\mathcal{A}^*(m, n+4)$ consists of a realizable set of $n+4$ points seen in m views. This set of points can be arranged in a table as in figure 20.1(left).

In this table, the points $\mathbf{x}_{n+k}^i$ are separated from the other points $\mathbf{x}_j^i$, since they will receive special treatment.

Step 1: Transform

The first step is to compute, for each i, a transformation $\mathtt{T}^i$ that maps the points $\mathbf{x}_{n+k}^i$, $k = 1, \ldots, 4$ in the i-th view to the points $\mathbf{e}_k$ of a canonical basis for projective 2-space $\mathbb{P}^2$. The transformation $\mathtt{T}^i$ is applied also to each of the points $\mathbf{x}_j^i$ to produce transformed points $\mathbf{x}_j'^i = \mathtt{T}^i \mathbf{x}_j^i$. The result is the transformed point array shown in figure 20.1(right). A different transformation $\mathtt{T}^i$ is computed and applied to each column of the array, as indicated.

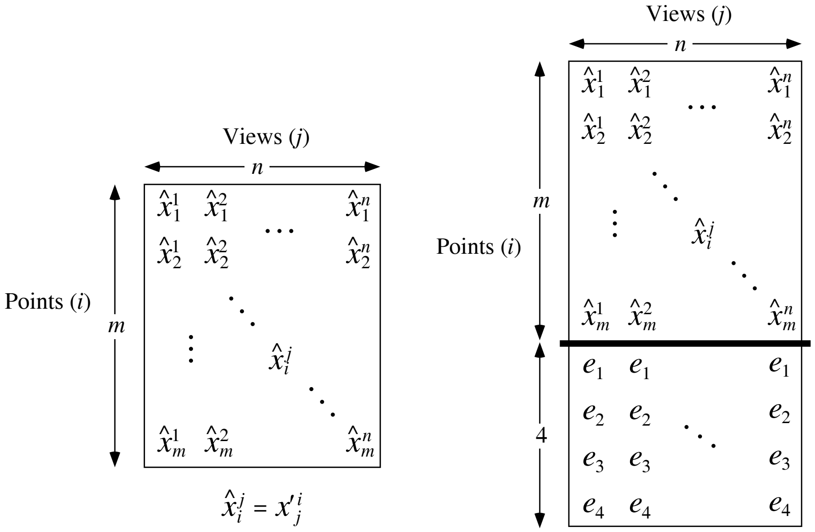

Fig. 20.2. **Left:** *Transposed data.* **Right:** *Transposed data extended by addition of extra points.*

Step 2: Transpose

The last four rows of the array are dropped, and the remaining block of the array is transposed. One defines $\hat{\mathbf{x}}_i^j = \mathbf{x}'^i_j$. At the same time, one does a mental switch of points and views. Thus the point $\hat{\mathbf{x}}_i^j$ is now conceived as being the image of the i-th point in the j-th view, whereas the point $\mathbf{x}'^i_j$ was the image of the j-th point in the i-th view. What is happening here effectively is that the roles of points and cameras are being swapped – the basic concept behind Carlsson duality expressed by (20.2). The resulting transposed array is shown in figure 20.2(left).

Step 3: Extend

The array of points is now extended by the addition of four extra rows containing points $\mathbf{e}_k$ in all positions of the $(m + k)$-th row of the array, as shown in figure 20.2(right). The purpose of this extension will be explained in section 20.1.2.

Step 4: Solve

The array of points resulting from the last step has $m+4$ rows and n columns, and may be regarded as the positions of $m + 4$ points seen in n views. As such, it is a candidate for solution by the algorithm $\mathcal{A}(n, m + 4)$, which we have assumed is given. Essential here is that the points in the array form a realizable set of point correspondences. Justification of this is deferred for now. The result of the algorithm $\mathcal{A}(n, m + 4)$ is a set of cameras $\hat{\mathtt{P}}^j$ and points $\widehat{\mathbf{X}}_i$ such that $\hat{\mathbf{x}}_i^j = \hat{\mathtt{P}}^j \widehat{\mathbf{X}}_i$. In addition, corresponding to the last four rows of the array, there are points $\widehat{\mathbf{X}}_{m+k}$ such that $\mathbf{e}_k = \hat{\mathtt{P}}^j \widehat{\mathbf{X}}_{m+k}$ for all j.

Step 5: 3D transformation

Since the reconstruction obtained in the last step is a projective reconstruction, one may transform it (equivalently, choose a projective coordinate frame) such that the points

$\widehat{\mathbf{X}}_{m+k}$ are the four points $\mathbf{E}_k$ of a partial canonical basis for $\mathbb{P}^3$. The only requirement is that the points $\widehat{\mathbf{X}}_{m+k}$ obtained in the projective reconstruction are not coplanar. This assumption is validated later.

At this point, one sees that $\mathbf{e}_k = \hat{\mathtt{P}}^j \widehat{\mathbf{X}}_{m+k} = \hat{\mathtt{P}}^j \mathbf{E}_k$. From this it follows that $\hat{\mathtt{P}}^j$ has the special form

$$\hat{\mathtt{P}}^j = \begin{bmatrix} a^j & & & d^j \\ & b^j & & d^j \\ & & c^j & d^j \end{bmatrix} . \tag{20.3}$$

Step 6: Dualize

Let $\widehat{\mathbf{X}}_i = (\mathrm{X}_i, \mathrm{Y}_i, \mathrm{Z}_i, \mathrm{T}_i)^\mathsf{T}$, and $\hat{\mathtt{P}}^j$ be as given in (20.3). Now define points $\mathbf{X}_j = (a^j, b^j, c^j, d^j)^\mathsf{T}$ and cameras

$$\mathtt{P}'^i = \begin{bmatrix} \mathrm{X}_i & & & \mathrm{T}_i \\ & \mathrm{Y}_i & & \mathrm{T}_i \\ & & \mathrm{Z}_i & \mathrm{T}_i \end{bmatrix} .$$

Then one verifies that

$$\begin{aligned} \mathtt{P}'^i \mathbf{X}_j &= (\mathrm{X}_i a^j + \mathrm{T}_i d^j, \mathrm{Y}_i b^j + \mathrm{T}_i d^j, \mathrm{Z}_i c^j + \mathrm{T}_i d^j)^\mathsf{T} \\ &= \hat{\mathtt{P}}^j \widehat{\mathbf{X}}_i \\ &= \hat{\mathbf{x}}_i^j \\ &= \mathbf{x}_j'^i . \end{aligned}$$

If, in addition, one defines $\mathbf{X}_{n+k} = \mathbf{E}_k$ for $k = 1, \ldots, 4$, then $\mathtt{P}'^i \mathbf{X}_{n+k} = \mathbf{e}_k$. It is then evident that the cameras $\mathtt{P}'^i$ and points $\mathbf{X}_j$ and $\mathbf{X}_{n+k}$ form a projective realization of the transformed data array obtained in step 1 of this algorithm.

Step 7: Reverse transformation

Finally, defining $\mathtt{P}^i = (\mathtt{T}^i)^{-1}\mathtt{P}'^i$, and with the points $\mathbf{X}_j$ and $\mathbf{X}_{n+k}$ obtained in the previous step, one has a projective realization of the original data. Indeed, one verifies

$$\mathtt{P}^i \mathbf{X}_j = (\mathtt{T}^i)^{-1} \mathtt{P}'^i \mathbf{X}_j = (\mathtt{T}^i)^{-1} \mathbf{x}_j'^i = \mathbf{x}_j^i .$$

This completes the description of the algorithm. One can see that it takes place in various stages.

(i) In step 1, the data is transformed into canonical image reference frames based on the selection of four distinguished points.
(ii) In steps 2 and 3 the problem is mapped into the dual domain, resulting in a dual problem $\mathcal{P}(n, m+4)$.
(iii) The dual problem is solved in steps 4 and 5.
(iv) Step 6 maps the solution back into the original domain.
(v) Step 7 undoes the effects of the initial transformation.

20.1.2 Justification of the algorithm

To justify this algorithm, one needs to be sure that at step 4 there indeed exists a solution to the transformed problem. Before considering this, it is necessary to explain the purpose of step 3, which extends the data by the addition of rows of image points $\mathbf{e}_k$, and step 5, which transforms the arbitrary projective solution to one in which four points are equal to the 3D basis points $\mathbf{E}_k$.

The purpose of these steps is to ensure that one obtains a solution to the dual reconstruction problem in which $\hat{\mathtt{P}}^j$ has the special form given by (20.3) in which the camera matrix is parametrized by only 4 values. The dual algorithm is described in this manner so that it will work with any algorithm $\mathcal{A}(n, m+4)$ whatever. However, both steps 3 and 5 may be eliminated if the known algorithm $\mathcal{A}(n, m+4)$ has the capability of enforcing this constraint on the camera matrices directly. Algorithms based on the fundamental matrix, trifocal or quadrifocal tensors may easily be modified in this way, as will be seen.

In the mean time, since $\hat{\mathtt{P}}^j$ of the form (20.3) is called a *reduced camera matrix*, we call any reconstruction of a set of image correspondences in which each camera matrix is of this form a *reduced reconstruction*. Not all sets of realizable point correspondences allow a reduced realization, however. The following result characterizes sets of point correspondences that do have this property.

Result 20.2. *A set of image points* $\{\mathbf{x}_j^i \ : \ i = 1, \ldots, m \ ; \ j = 1, \ldots, n\}$ *admits a reduced realization if and only if it may be augmented with supplementary correspondences* $\mathbf{x}_{n+k}^i = \mathbf{e}_k$ *for* $k = 1, \ldots, 4$ *such that*

(i) *The total set of image correspondences is realizable, and*
(ii) *The reconstructed points* $\mathbf{X}_{n+k}$ *corresponding to the supplementary image correspondences are non-coplanar.*

Proof. The proof is straightforward enough. Suppose the set permits a reduced realization, and let $\mathtt{P}^i$ be the set of reduced camera matrices. Let points $\mathbf{X}_{n+k} = \mathbf{E}_k$ for $k = 1, \ldots, 4$ be projected into the m images. The projections are $\mathbf{x}_{n+k}^i = \mathtt{P}^i\mathbf{X}_{n+k} = \mathtt{P}^i\mathbf{E}_k = \mathbf{e}_k$ for all i.
Conversely, suppose the augmented set of points is realizable and the points $\mathbf{X}_{n+k}$ are non-coplanar. In this case, a projective basis may be chosen such that $\mathbf{X}_{n+k} = \mathbf{E}_k$. Then for each view, one has $\mathbf{e}_k = \mathtt{P}^i\mathbf{E}_k$ for all k. From this it follows that each $\mathtt{P}^i$ has the desired form (20.3). □

One other remark must be made before proving the correctness of the algorithm.

Result 20.3. *If a set of image points* $\{\mathbf{x}_j^i \ : \ i = 1, \ldots, m \ ; \ j = 1, \ldots, n\}$ *permits a reduced realization then so does the transposed set* $\{\hat{\mathbf{x}}_i^j \ : \ j = 1, \ldots, n \ ; \ i = 1, \ldots, m\}$ *where* $\hat{\mathbf{x}}_i^j = \mathbf{x}_j^i$ *for all* i *and* j.

This is the basic duality property, effectively proved by the construction given in step 6 of the algorithm above. Now it is possible to prove the correctness of the algorithm.

Result 20.4. *Let $\mathbf{x}^i_j$ and $\mathbf{x}^i_{n+k}$ as in figure 20.1(left) be a set of realizable image point correspondences, and suppose*

(i) *for each i, the four points $\mathbf{x}^i_{n+k}$ do not include three collinear points.*
(ii) *the four points $\mathbf{X}_{n+k}$ in a projective reconstruction are non-coplanar.*

Then the algorithm of section 20.1.1 will succeed.

Proof. Because of the first condition, transformations $\mathtt{T}^i$ exist for each i, transforming the input data to the form shown in figure 20.1(right). This transformed data is also realizable, since the transformed data differs only by a projective transformation of the image.
Now, according to result 20.2 applied to figure 20.1(right), the correspondences $\mathbf{x}'^i_j$ admit a reduced realization. By result 20.3 the transposed data figure 20.2(left) also admits a reduced realization. Applying result 20.2 once more shows that the extended data figure 20.2(right) is realizable. Furthermore, the points $\widehat{\mathbf{X}}_{m+k}$ are non-coplanar, and so step 5 is valid. The subsequent steps 6 and 7 go forward without problems. □

The first condition may be checked from the image correspondences $\mathbf{x}^i_j$. It may be thought that to check the second condition requires reconstruction to be carried out. It is, however, possible to check whether the reconstructed points will be coplanar without carrying out the reconstruction. This is left as an exercise for the reader (page 342).

20.2 Reduced reconstruction

In this section, we concentrate on and reevaluate steps 3 – 5 of the algorithm described in the preceding section. To recapitulate, the purpose of these steps is to obtain a reduced reconstruction from a set of image correspondences. Thus, the input is a set of image correspondences $\hat{\mathbf{x}}^j_i$ admitting a reduced realization (see figure 20.2(left)). The output is a set of *reduced* camera matrices $\hat{\mathtt{P}}^j$ and points $\widehat{\mathbf{X}}_i$ such that $\hat{\mathtt{P}}^j\widehat{\mathbf{X}}_i = \hat{\mathbf{x}}^j_i$ for all i, j.

As we have seen, one way to do this (as in steps 3 – 5) of the given algorithm is to augment the points by the addition of four extra synthetic point correspondences $\hat{\mathbf{x}}^j_{m+k}$, carrying out projective reconstruction, and then applying a 3D homography so that the 3D points $\widehat{\mathbf{X}}_{m+k}$ are mapped to the points $\mathbf{E}_k$ of a projective basis for $\mathbb{P}^3$. The problem with this is that in the presence of noise, the projective reconstruction is not exact. Thus, the camera matrices obtained by this method will map points $\mathbf{E}_k$ to points close to, but not identical with $\mathbf{e}_k$. This means that the camera matrices are not exactly in reduced form. Therefore, we now consider methods of computing a realization of the point correspondences in which the cameras are exactly reduced.

20.2.1 The reduced fundamental matrix

The most evident applications of these dual methods are to dualize the reconstruction algorithms involving the fundamental matrix and trifocal tensor. These will lead to reconstruction algorithms for 6 or 7 points (respectively) across N views. In this section, we consider reconstruction from 6 points. The dual of a reconstruction problem

$\mathcal{P}(N, 6)$ is a problem $\mathcal{P}(2, N+4)$, namely reconstruction from $N+4$ points in 2 views. The method of chapter 10 involving the fundamental matrix is a standard way of solving such a problem.

To this end we define a reduced fundamental matrix:

Definition 20.5. A fundamental matrix $\hat{\mathtt{F}}$ is called a *reduced* fundamental matrix if it satisfies the condition $\mathbf{e}_i^\mathsf{T}\hat{\mathtt{F}}\mathbf{e}_i = 0$ for $i = 1, \ldots, 4$.

It is evident that since a reduced fundamental matrix already satisfies constraints derived from four point correspondences, it may be computed from a small number of additional points, in fact linearly from four points, or non-linearly from three points.

20.2.2 Computation of the reduced fundamental matrix

For a reduced fundamental matrix, the condition $\mathbf{e}_i^\mathsf{T}\hat{\mathtt{F}}\mathbf{e}_i = 0$ for $i = 1, \ldots, 3$ implies that the diagonal entries of $\hat{\mathtt{F}}$ are zero. The requirement that $(1,1,1)\hat{\mathtt{F}}(1,1,1)^\mathsf{T} = 0$ gives the additional condition that the sum of entries of $\hat{\mathtt{F}}$ is zero. Thus one may write $\hat{\mathtt{F}}$ in the form

$$\hat{\mathtt{F}} = \begin{bmatrix} 0 & p & q \\ r & 0 & s \\ t & -(p+q+r+s+t) & 0 \end{bmatrix} \tag{20.4}$$

thereby parametrizing a fundamental matrix satisfying all the linear constraints (though not the condition $\det \hat{\mathtt{F}} = 0$). Now, a further point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ satisfying $\mathbf{x}'^\mathsf{T}\hat{\mathtt{F}}\mathbf{x} = 0$ is easily seen to provide a linear equation in the parameters $p, \ldots, t$ of $\hat{\mathtt{F}}$. Given at least four such correspondences, one may solve for these parameters, up to an inconsequential scale. Given only three such correspondences, the extra constraint $\det \hat{\mathtt{F}} = 0$ may be used to provide the extra constraint necessary to determine $\hat{\mathtt{F}}$. There may be one or three solutions. This computation is analogous to the method used in section 11.1.2(*p*281) to compute the fundamental matrix from seven point correspondences. With four correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ or more, one finds a least-squares solution.

20.2.3 Retrieving reduced camera matrices

Computing a pair of reduced camera matrices that correspond to a reduced fundamental matrix is surprisingly tricky. One can not assume that the first camera is $[\mathtt{I} \mid \mathbf{0}]$ as in the usual projective camera case, since this is non-generic, the camera centre corresponding with the basis point $\mathbf{E}_4 = (0,0,0,1)^\mathsf{T}$. However, we may instead assume that the pair of cameras are of the form

$$\mathtt{P} = \begin{bmatrix} 1 & & & 1 \\ & 1 & & 1 \\ & & 1 & 1 \end{bmatrix} \text{ and } \mathtt{P}' = \begin{bmatrix} a & & & d \\ & b & & d \\ & & c & d \end{bmatrix} \tag{20.5}$$

since the centre of the first camera is $(1,1,1,1)^\mathsf{T}$ which is a generic point with respect to the basis points $\mathbf{E}_1, \ldots, \mathbf{E}_4$. Further, if $d \neq 0$, then we may assume that $d = 1$, but we prefer not to do this.

The reduced fundamental matrix corresponding to this pair of cameras is

$$\hat{\mathtt{F}} = \begin{bmatrix} 0 & b(d-c) & -c(d-b) \\ -a(d-c) & 0 & c(d-a) \\ a(d-b) & -b(d-a) & 0 \end{bmatrix} \tag{20.6}$$

which the reader may verify satisfies the four linear constraints as well as the zero-determinant condition. The task at hand is to retrieve the values of (a, b, c, d) given the fundamental matrix. This seemingly requires a solution of simultaneous quadratic equations, but there is a linear method as follows.

(i) The ratio $a : b : c$ may be found by solving the set of homogeneous linear equations

$$\begin{bmatrix} f_{12} & f_{21} & 0 \\ f_{13} & 0 & f_{31} \\ 0 & f_{23} & f_{32} \end{bmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \mathbf{0} \tag{20.7}$$

where f_{ij} is the ij-th entry of $\hat{\mathtt{F}}$. The matrix appearing here clearly has the same rank as $\hat{\mathtt{F}}$ (namely 2), so there is a unique solution (up to scale) to this equation set. The solution $a : b : c = A : B : C$ provides a set of homogeneous equations in a, b and c, namely $Ba = Ab$, $Ca = Ac$ and $Cb = Bc$, of which two are linearly independent.

(ii) Similarly, one may find the ratio $d - a : d - b : d - c$ by solving $(d - a, d - b, d - c)\hat{\mathtt{F}} = \mathbf{0}$. Once again, the solution is unique. This provides two more linear equations in the values of a, b, c, d.

(iii) From the set of equations one may solve for (a, b, c, d) up to scale, and hence reconstruct the second camera matrix according to (20.5).

20.2.4 Solving for six points in three views

The minimal case involving the computation of the reduced fundamental matrix involves three point, in which case there may be three solutions. By dualization, this leads to a solution to the reconstruction problem for six points in three views. Its use in outlier detection using the trifocal tensor has been described in algorithm 16.4(*p*401). Because of its intrinsic interest as a minimum-case solution, and because of its practical use, the algorithm is given explicitly as algorithm 20.1. The algorithm consists essentially of putting together what has been described previously. However there are a few minor variations.

In algorithm 20.1, the final estimation of the camera matrices takes place in the domain of the original point measurements. As an alternative, one could apply a DLT algorithm in the dual domain, as in the basic algorithm of section 20.1.1. However, the present method seems simpler. It has the advantage of avoiding the need for applying the inverse transformations $\mathtt{T}, \mathtt{T}', \mathtt{T}''$. More significantly the final computation of the camera matrices is carried out using the original data. This is important, because the transformations can severely skew the noise characteristics of the data.

The basis for this algorithm and the n-view case to follow is the dual fundamental

<u>Objective</u> Given a set of six point correspondences $\mathbf{x}_j \leftrightarrow \mathbf{x}'_j \leftrightarrow \mathbf{x}''_j$ across three views, compute a reconstruction from these points, consisting of the three camera matrices $\mathtt{P}$, $\mathtt{P}'$ and $\mathtt{P}''$ and the six 3D points $\mathbf{X}_1, \ldots, \mathbf{X}_6$.

<u>Algorithm</u>

(i) Select a set of four points no three of which are collinear in any of the three views. Let these be the correspondences $\mathbf{x}_{2+k} \leftrightarrow \mathbf{x}'_{2+k} \leftrightarrow \mathbf{x}''_{2+k}$ for $k = 1, \ldots, 4$.

(ii) For the first view, find a projective transformation $\mathtt{T}$ that maps each $\mathbf{x}_{2+k}$ to $\mathbf{e}_k$. Apply $\mathtt{T}$ to the other two points $\mathbf{x}_1$ and $\mathbf{x}_2$ resulting in points $\hat{\mathbf{x}}_1 = \mathtt{T}\mathbf{x}_1$ and $\hat{\mathbf{x}}_2 = \mathtt{T}\mathbf{x}_2$.

(iii) From the *dual correspondence* $\hat{\mathbf{x}}_1 \leftrightarrow \hat{\mathbf{x}}_2$ derive an equation in the entries p, q, r, s, t of the reduced fundamental matrix $\hat{\mathtt{F}}$ as in (20.4). The equation is derived from the relationship $\hat{\mathbf{x}}_2^{\mathsf{T}} \hat{\mathtt{F}} \hat{\mathbf{x}}_1 = 0$.

(iv) In the same way as in the last two steps, form two more equations from the points in the other two views. This results in a set of three homogeneous equations in five unknowns. There will be a two-parameter family of solutions for the reduced fundamental matrix, generated by two independent solutions $\hat{\mathtt{F}}_1$ and $\hat{\mathtt{F}}_2$.

(v) The general solution is $\hat{\mathtt{F}} = \lambda \hat{\mathtt{F}}_1 + \mu \hat{\mathtt{F}}_2$. From the requirement $\det \hat{\mathtt{F}} = 0$ one derives a homogeneous cubic equation in (λ, μ), which one may solve to find (λ, μ), and hence $\hat{\mathtt{F}}$. There will be 1 or 3 real solutions. Further steps are applied to each solution in turn.

(vi) Use the method of section 20.2.3 to extract the parameters $(a, b, c, d)^{\mathsf{T}}$ of the second reduced camera matrix $\hat{\mathtt{P}}'$ defined in (20.5).

(vii) We complete the reconstruction in the original measurement domain. The dual of camera $\hat{\mathtt{P}}'$ defined by (a, b, c, d) is the point $\mathbf{X}_2 = (a, b, c, d)^{\mathsf{T}}$. Thus the six 3D points are $\mathbf{X}_1 = (1, 1, 1, 1)^{\mathsf{T}}$, $\mathbf{X}_2 = (a, b, c, d)^{\mathsf{T}}$ and $\mathbf{X}_{2+k} = \mathbf{E}_k$ for $k = 1, \ldots, 4$. This gives the structure of the reconstructed scene. One may then compute the three camera matrices using the DLT algorithm for camera calibration described in section 7.1(*p*178). Since we require the camera matrices defined with respect to the original camera coordinates here, we use the original coordinates to solve for $\mathtt{P}$, $\mathtt{P}'$ and $\mathtt{P}''$ such that $\mathtt{P}\mathbf{X}_j = \mathbf{x}_j$, *etc.* Since the solution will be exact, the DLT solution will be sufficient.

Algorithm 20.1. *Algorithm for computing a projective reconstruction from six points in three views.*

matrix, $\hat{\mathtt{F}}$. Note how the dual fundamental matrix expresses a relationship between points in the same image. Indeed the equations used to solve for $\hat{\mathtt{F}}$ are constructed from points in the same image. This contrasts with the standard fundamental matrix where the relationships being encoded are between points seen in separate images.

20.2.5 Six points in n views

The method for six points in three views can be applied with little modification to the case of six points in many views. The main difference is that the reduced fundamental matrix $\hat{\mathtt{F}}$ will be uniquely determined by the data. Specifically at step 4 of the algorithm of section 20.2.4, each view will contribute one equation. With four views or more, this will be sufficient to determine $\hat{\mathtt{F}}$.

In this redundant-data case, one must be careful with the effects of noise. For this reason, it appears preferable to carry out the last step of the algorithm, as shown, with original untransformed points. This mitigates the effect of noise distortion that would result from working with transformed points.

20.2.6 Seven points in n views

The problem of seven points in n views is dual to the case of three views of $n+4$ points and is solved by computing the reduced trifocal tensor from n point correspondences.

Definition 20.6. A trifocal tensor $\mathcal{T}$ is called a *reduced trifocal tensor* if it satisfies the linear constraints imposed by synthetic point correspondences $\mathbf{e}_k \leftrightarrow \mathbf{e}'_k \leftrightarrow \mathbf{e}''_k$ for $i = 1, \ldots, 4$.

The general method for reconstruction from 7 points is similar to the method for six points in n views, except that the reduced trifocal tensor is used instead of the reduced fundamental matrix. There are, however some minor differences.

In computing the reduced trifocal tensor, the constraints corresponding to the synthetic correspondences $\mathbf{e}_j \leftrightarrow \mathbf{e}_j \leftrightarrow \mathbf{e}_j$ should be satisfied exactly, whereas the other correspondences used to compute the tensor are subject to noise, and will only be satisfied approximately. Otherwise the computed tensor will not be exactly in reduced form. In the case of the reduced fundamental matrix this was handled by giving a specific parametrization of the reduced fundamental matrix. That is, it was parametrized in such a way that the constraints generated by the synthetic correspondences were automatically satisfied (see (20.4)). In the case of the trifocal tensor, it is not obvious that such a convenient parametrization is possible. The synthetic constraints are of the form

$$e^i e^p e^q \epsilon_{jpr} \epsilon_{kqs} \mathcal{T}_i^{jk} = 0_{rs}$$

which is rather more complicated than the linear constraints on the reduced fundamental matrix. Instead, one may proceed in the following manner.

In the usual linear method of computing the trifocal tensor, one must solve a set of linear equations of the form $\mathtt{A}\mathbf{t} = \mathbf{0}$, or more precisely, find the vector $\mathbf{t}$ that minimizes $\|\mathtt{A}\mathbf{t}\|$ subject to $\|\mathbf{t}\| = 1$. In solving for the reduced trifocal tensor, the matrix $\mathtt{A}$ may be divided into two parts, corresponding to those constraints from the synthetic correspondences, which should be satisfied exactly, and the constraints from real correspondences, which must be satisfied in a least-squares sense. The first set of constraints are of the form $\mathtt{C}\mathbf{t} = \mathbf{0}$, and the second set may be written as $\hat{\mathtt{A}}\mathbf{t} = \mathbf{0}$. The problem becomes: find $\mathbf{t}$ that minimizes $\|\hat{\mathtt{A}}\mathbf{t}\|$ subject to $\|\mathbf{t}\| = 1$ and $\mathtt{C}\mathbf{t} = \mathbf{0}$. An algorithm for solving this problem is given as algorithm A5.5(*p*594).

For the problem of extracting the three camera matrices from the reduced tensor no simple method seems to be available, similar to that described in section 20.2.3 for the fundamental matrix. Instead, one may use the method described in steps 5 and 6 of the general dual algorithm (see page 505).

The minimal configuration of this type is seven points in two views. In this case, the problem is best solved directly by the method of section 11.1.2(*p*281), rather than by dualization to the case of three points in six views.

20.2.7 Performance issues

Dual reconstruction algorithms based on the reduced fundamental matrix and trifocal tensor have been implemented and tested. The results of these tests were contained in a student report presented in August 1996 by Gilles Debunne. Since this report is effectively unavailable, the results are summarized here.

The most serious difficulty is the distortion of the noise distribution by the application of projective transformations T^i to the images. Application of projective transformations to the image data has the effect of distorting any noise distribution that may apply to the data. The problem is related to the need to choose four points that are non-collinear in any of the images. If the points are close to collinear in any of the images, then the projective transformation applied to the image in step 1 of the algorithm may entail extreme distortion of the image. This sort of distortion can degrade performance of the algorithm severely.

Without special attention being paid to the noise-distortion, performance of the algorithms was generally unsatisfactory, Despite great care being taken to minimize errors due to noise in steps 4–6 of the algorithm (page $505ff$), when the inverse projective transformations are applied in step 7, the average error became very large. Some points retained quite small error, whereas in those images where distortion was significant, quite large errors resulted.

Normalization in the sense of section 4.4.4(p107) is also a problem. It has been shown to be essential for performance of the linear reconstruction algorithms to apply data normalization. However, what sort of normalization should be applied to the transformed data of figure 20.1(right), which is geometrically unrelated to actual image measurements, is a mystery.

To get good results, it would seem that one would need to propagate assumed error distributions forward in step 1 of the algorithm to get assumed error distributions for the transformed data figure 20.1(right), and then during reconstruction to minimize residual error relative to this propagated error distribution. Equivalently, cost functions being minimized during reconstruction must be related back to measurement error in the original image points. Recent work reported in [Hartley-00a, Schaffalitzky-00c] has shown that this indeed gives significantly improved results.

20.3 Closure

20.3.1 The literature

The idea behind Carlsson–Weinshall duality was first described in a pair of papers appearing simultaneously: [Carlsson-95] and Weinshall [Weinshall-95] and subsequently in a joint paper [Carlsson-98]. The treatment given here for the general method of dualizing an algorithm was given in [Hartley-98b], derived from these earlier papers. Details of methods for handling with noise propagation were given in [Hartley-00a], building on earlier implementations contained in an unavailable report by Gilles Debunne (August 1996).

The problem of reconstruction from six points in three views was perhaps first treated in a technical report [Hartley-92b] (later published as [Hartley-94c]) where the existence of up to eight solutions was shown. The problem was given a complete solution in [Quan-94] where it was shown that only three solutions are possible. This was also pointed out in [Ponce-94]. The paper [Carlsson-95] established that this problem is dual to the two-view, seven point problem for which a solution was known. This enabled the formulation of the method given in this chapter. The minimum six point configuration was used in [Torr-97] for robust estimation of the trifocal tensor. An alternative method for computing a reconstruction from six points in $n \geq 3$ views is given in [Schaffalitzky-00c]. This method does not require that images are first projectively transformed to a canonical basis.

20.3.2 Notes and Exercises

(i) In the dual algorithm of section 20.1.1 it was noted that the method will only work if the four points used to define the image transformations are non-coplanar. However, note that in this case, the algorithm of section 18.5.2(*p*450) will compute a projective reconstruction linearly.

(ii) If the four chosen points are coplanar, then the homographies $\mathtt{T}^i$ will map the plane to a common coordinate system. The transformed points $\mathbf{x}'^i_j$ will then satisfy the condition of section 17.5.2(*p*425), namely the lines joining any pair of points $\mathbf{x}'^i_j$ and $\mathbf{x}'^i_k$ (j and k fixed and a different line for each i) will meet in a common point. The duality in the case is described in [Criminisi-98, Irani-98].

(iii) Still in the case of the four chosen points being coplanar: after applying the $\mathtt{T}^i$, any point on the plane of the four points will map to the same point in all images. Thus, the fundamental matrix consistent with the point correspondences will be skew-symmetric.