rotation. One possible choice is to assume that at time $t = 0$ the world and camera reference frame coincide.

Here is a concise description of the entire method. A method for determining $Q$ from (8.39) is discussed in Exercise 8.8.

**Algorithm MOTSTRUCT_FROM_FLATS**

The input is the registered measurement matrix $\hat{W}$, computed from $n$ features tracked over $N$ consecutive frames.

1. Compute the SVD of $\hat{W}$,

   $$ \hat{W} = UD\Sigma V^T, $$

   where $U$ is a $2N \times 2n$ matrix, $V \times n$, and $D$ a $2n \times n$; $U^TU = I$, $V^TV = I$; and $D$ is the diagonal matrix of the singular values.

2. Set to zero all but the three largest singular values in $D$.

3. Define $\hat{R}$ and $\hat{S}$ as in (8.5.1).

4. Solve (8.39) for $Q$, for example by means of Newton's method (Exercise 8.8).

The output are the rotation and shape matrices, given by

$$ R = \hat{R}Q \; \text{ and } \; S = Q^{-1}\hat{S}. $$

The algorithm determines the rotation of a set of 3-D points with respect to the camera, but how about their translation? The component of the translation parallel to the image plane is simply proportional to the frame-by-frame motion of the centroid of the data points on the image plane. However, because of the orthographic assumption, the component of the translation along the optical axis cannot be determined.

8.5.2 3-D Motion and Structure from a Dense Motion Field

We now discuss the reconstruction of 3-D motion and structure from optical flow. The two major differences with the previous section are that

- optical flow provides dense but often inaccurate estimates of the motion field;
- the analysis is instantaneous, not integrated over many frames.

**Problem Statement**

Given an optical flow and the intrinsic parameters of the viewing camera, recover the 3-D motion and structure of the observed scene with respect to the camera reference frame.

We have chosen a method that represents a good compromise between ease of implementation and quality of results. The method consists of two stages:
1. determine the direction of translation through approximate motion parallax;
2. determine a least-squares approximation of the rotational component of the optical flow, and use it in the motion field equations to compute depth.

**Stage 1: Translation Direction.** The first stage is rather complex. We start by explaining the solution in the ideal case of exact motion parallax, then move to the case of approximate parallax. We learned in section 8.2.1 that the relative motion field of two *instantaneously coincident* image points, \([\Delta x_i, \Delta y_i, \Delta z_i]\), is directed toward (or away from) the vanishing point of the translation direction, \(p_0\) (the instantaneous epipole), according to

\[
\Delta x_i = (T_x z - T_z f) \left( \frac{1}{Z} \right)
\]

\[
\Delta y_i = (T_y z - T_z f) \left( \frac{1}{Z} \right)
\]

(8.40)

where \(Z\) and \(\tilde{Z}\) are the depths of the 3-D points \(P = (X, Y, Z)\) and \(\tilde{P} = (\tilde{X}, \tilde{Y}, \tilde{Z})\), which project on the same image point, \(p = [x, y]^T\), in the frame considered. If (8.40) can be written for two different image points, we can locate the epipole, \(p_0\), as the intersection of the estimated relative motion fields. Once the epipole is known, it is straightforward to get the direction of translation from (8.9).

This solution can be extended to the more realistic case of approximate motion parallax, in which the estimates of the relative motion field are available only for *almost coincident* image points. The key observation is that the differences between the optical flow vectors at an image point \(p\) and at any point close to \(p\) can be regarded as noisy estimates of the motion parallax \(\Delta p\) (section 8.2.4).

We must now rewrite the (8.40) for the case of approximate parallax. We begin by writing the translational and rotational components of the relative motion field, \([\Delta x_i, \Delta y_i, \Delta z_i]\) and \([\Delta x_r, \Delta y_r, \Delta z_r]\) respectively, for two almost coincident image points, \(p\) and \(\tilde{p}\). These are

\[
\Delta x_i = \frac{T_x x - T_z f}{Z} - \frac{T_x \tilde{x} - T_z f}{\tilde{Z}}
\]

\[
\Delta y_i = \frac{T_y y - T_z f}{Z} - \frac{T_y \tilde{y} - T_z f}{\tilde{Z}}
\]

(8.41)

and

\[
\Delta x_r = a_0 (x - \tilde{x}) + \frac{a_0}{f} (y - \tilde{y}) - \frac{a_0}{f} (x^2 - \tilde{x}^2)
\]

\[
\Delta y_r = -a_0 (x - \tilde{x}) - \frac{a_0}{f} (y - \tilde{y}) + \frac{a_0}{f} (y^2 - \tilde{y}^2)
\]

(8.42)
From the rotation equations we notice that $\Delta v_z^2 \to 0$ and $\Delta v_y^2 \to 0$ for $\bar{p} \to p$. As to the translation equations, we can rewrite them as

$$
\Delta v_x^2 = (T_x - T_i f) \left( \frac{1}{Z} \right) + \frac{T_y}{Z} (x - \bar{x})
$$

$$
\Delta v_y^2 = (T_y - T_i f) \left( \frac{1}{Z} \right) + \frac{T_z}{Z} (y - \bar{y})
$$

(8.43)

The second terms of the right-hand side of (8.43) tend to zero for $\bar{p} \to p$, while the first terms tend to the expression obtained for the exact motion parallax (8.40). We can therefore write the relative motion field of two almost coincident points concisely as

$$
\Delta v_x = (T_x - T_i f) \left( \frac{1}{Z} \right) + e_x(p - \bar{p})
$$

$$
\Delta v_y = (T_y - T_i f) \left( \frac{1}{Z} \right) + e_y(p - \bar{p})
$$

(8.44)

with $e_x$ and $e_y$ smooth functions of the difference between $p$ and $\bar{p}$, and $e_x(0) = e_y(0) = 0$.

Equations (8.44) show that, if $p$ and $\bar{p}$ are close enough, a large relative motion field can only be due to a large difference in depth between the 3-D points $p$ and $\bar{p}$. This observation suggests a relatively simple algorithm for locating the instantaneous epipole (and therefore the direction of translation) from a number of approximate motion parallax estimates. We compute the flow differences $(\Delta v_x, \Delta v_y)$ between a point $p_i$ and all its neighbors within a small patch $Q_i$, then determine the eigenvalues and eigenvectors of the matrix

$$
\lambda_i = \left[ \sum \Delta^2 v_x \sum \Delta v_x \Delta v_y \sum \Delta v_y \right].
$$

(8.45)

where the sums are taken over all the $Q_i$; the eigenvector corresponding to $\lambda_i$, the greater eigenvalue, identifies the direction of the line $k_i$ through $p_i$ which minimizes the sum of the squared distances to the set of difference vectors (Appendix, section A.6). This direction is taken to be the optimal estimate of motion parallax within the patch $Q_i$.

Moreover, $\lambda_i$ itself can be regarded as a measure of the estimate’s reliability. If $\lambda_i$ is large, the underlying distribution of the flow differences has a peak in the direction of $k_i$. This is likely to be due to the presence of considerable differences in depth within $Q_i$. Instead, if $\lambda_i$ is small, the underlying distribution of the flow differences is flatter, and almost certainly created by the flow fields of a surface that does not vary much in depth within $Q_i$.

---

14 One might argue that what really counts should be the ratio between the smaller and greater eigenvalues. However, since the range of $\lambda_1$ and $\lambda_2$ is finite, the greater eigenvalue is large in absolute terms.
We can now formulate a weighted least squares scheme to compute the intersection of the several lines $l_i$, that is, the epipole $p_0$. Since $l_i$, $p_0$, and $p_i$ are coplanar, for each patch $O_i$, we can write

$$\hat{d}_i \cdot p_i^* = 0.$$  

(8.46)

If there are $N$ patches, we can write $N$ simultaneous instances of (8.46), that is, in matrix notation,

$$Bp_0 = 0$$  

(8.47)

with

$$B = \begin{bmatrix} l_1 \times p_1^* \\ l_2 \times p_2^* \\ \vdots \\ l_N \times p_N^* \end{bmatrix}.$$  

(8.46)

The problem of determining a least-squares estimate of $p_0$ is thus reduced to the problem of solving the overconstrained homogeneous system (8.47). As customary, the solution can be found from the SVD of $B$, $B = UDV^T$ (Appendix, section A.6), as the columns of $V$ corresponding to the null (in practice, the smallest) singular value of $B$.

**Stage Two: Rotational Flow and Depth.** The rest of the algorithm is straightforward. We simply form the pointwise dot product $x_i \cdot y_i$ between the optical flow at point $p_i = [x_i, y_i]^T$ and the vector $[y_i - y_0, -(x_i - x_0)]^T$. As we know from section 8.2, $u_1$ depends only on the rotational component of motion; therefore, at each point $p_i$ of the image plane we have

$$u_1 = v_1^x(y_1 - y_0) = v_1^x(x_1 - x_0)$$  

(8.49)

with $v_1^x$ and $v_1^y$ as in (8.42). If the intrinsic parameters of the camera are known, we can write a linear system of $N$ simultaneous instances of (8.49) in the image reference frame by using (8.14), and solve for the three components of the angular velocity using least squares. Finally, we recover the translational direction from the epipole coordinates by means of (8.9), and solve (8.7) for the depth $Z$ of each image point.

It is now time to summarize the method.

**Algorithm MOTSTRUCT_FROM_FLOW**

The input quantities are the intrinsic parameters of the viewing camera, and a dense optical flow field, $v$, produced by a single rigid motion.