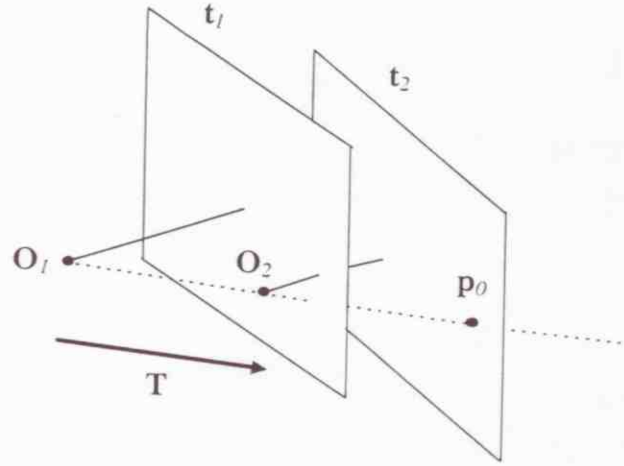

Figure 8.6 The point $\mathbf{p}_0$ as instantaneous epipole.

8.2.5 The Instantaneous Epipole

We close this introductory section with an important remark. The point $\mathbf{p}_0$, being the intersection of the image plane with the direction of translation of the center of projection, can be regarded as the *instantaneous epipole* between pairs of consecutive frames in the sequence (Figure 8.6). The main consequence of this property is that *it is possible to locate* $\mathbf{p}_0$ *without prior knowledge of the camera intrinsic parameters* (section 8.5.2).

☞ Notice that, as in the case of stereo, knowing the epipole's location in image coordinates is *not* equivalent to knowing the direction of translation (the baseline vector for stereo). The relation between epipole location and translation direction is specified by (8.9), which is written in the camera (not image) frame, and contains the focal length f. Therefore, *the epipole's location gives the direction of translation only if the intrinsic parameters of the viewing camera are known.*

8.3 The Notion of Optical Flow

We now move to the problem of *estimating the motion field from image sequences*, that is, from the spatial and temporal variations of the image brightness. To do this, we must model the link between brightness variations and motion field, and arrive at a fundamental equation of motion analysis, the *image brightness constancy equation*. We want also to analyze the power and validity of this equation, that is, understand how much and how well it can help us to estimate the motion field. For simplicity, we will assume that *the image brightness is continuous and differentiable as many times as needed in both the spatial and temporal domain.*

8.3.1 The Image Brightness Constancy Equation

It is common experience that, under most circumstances, the apparent brightness of moving objects remains constant. We have seen in Chapter 2 that the image irradiance is proportional to the scene radiance in the direction of the optical axis of the camera; if we assume that the proportionality factor is the same across the entire image plane, the constancy of the apparent brightness of the observed scene can be written as the stationarity of the image brightness E over time:

$$\frac{dE}{dt} = 0. \tag{8.15}$$

☞ In (8.15), the image brightness, E, should be regarded as a function of both the spatial coordinates of the image plane, x and y, and of time, that is, $E = E(x, y, t)$. Since x and y are in turn functions of t, the *total* derivative in (8.15) should not be confused with the *partial* derivative $\partial E/\partial t$.

Via the chain rule of differentiation, the total temporal derivative reads

$$\frac{dE(x(t), y(t), t)}{dt} = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0. \tag{8.16}$$

The partial spatial derivatives of the image brightness are simply the components of the spatial image gradient, ∇E, and the temporal derivatives, dx/dt and dy/dt, the components of the motion field, $\mathbf{v}$. Using these facts, we can rewrite (8.16) as the image brightness constancy equation.

The Image Brightness Constancy Equation

Given the image brightness, $E = E(x, y, t)$, and the motion field, $\mathbf{v}$,

$$(\nabla E)^\top \mathbf{v} + E_t = 0. \tag{8.17}$$

The subscript t denotes partial differentiation with respect to time.

We shall now discuss the relevance and applicability of this equation for the estimation of the motion field.

8.3.2 The Aperture Problem

How much of the motion field can be determined through (8.17)? *Only its component in the direction of the spatial image gradient,*[9] v_n. We can see this analytically by isolating the measurable quantities in (8.17):

$$-\frac{E_t}{\|\nabla E\|} = \frac{(\nabla E)^\top \mathbf{v}}{\|\nabla E\|} = v_n \tag{8.18}$$

[9] This component is sometimes called the *normal component*, because the spatial image gradient is normal to the spatial direction along which image intensity remains constant.

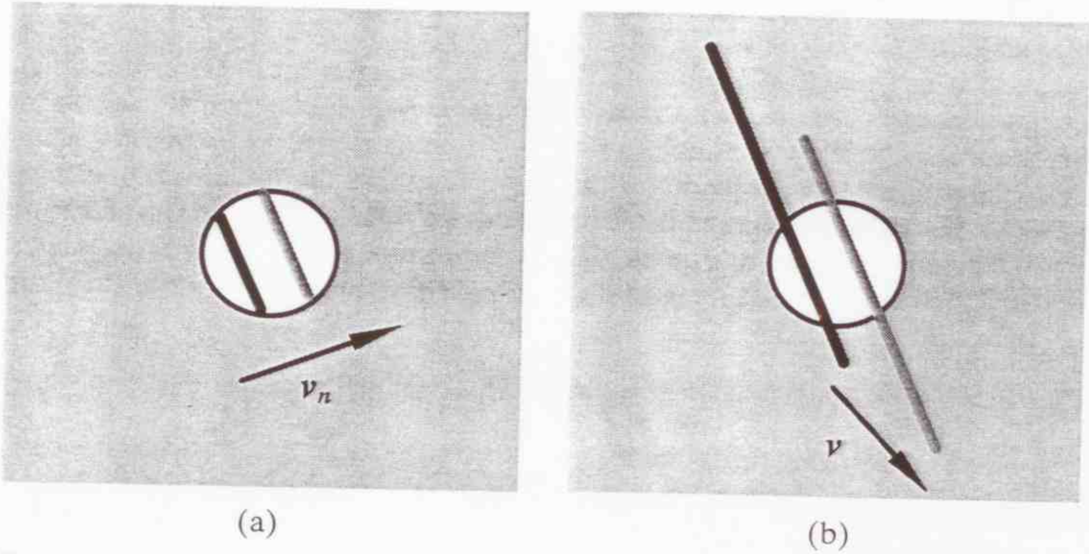

Figure 8.7 The aperture problem: the black and grey lines show two positions of the same image line in two consecutive frames. The image velocity perceived in (a) through the small aperture, $\mathbf{v}_n$, is only the component parallel to the image gradient of the true image velocity, $\mathbf{v}$, revealed in (b).

The Aperture Problem

The component of the motion field in the direction *orthogonal* to the spatial image gradient is not constrained by the image brightness constancy equation.

The aperture problem can be visualized as follows. Imagine to observe a thin, black rectangle moving against a white background through a small aperture. "Small" means that the corners of the rectangle are not visible through the aperture (Figure 8.7(a)); the small aperture simulates the narrow support of a differential method. Clearly, there are many, actually infinite, motions of the rectangle compatible with what you see through the aperture (Figure 8.7(b)); the visual information available is only sufficient to determine the velocity in the direction *orthogonal* to the visible side of the rectangle; the velocity in the *parallel* direction cannot be estimated.

☞ Notice that the parallel between (8.17) and Figure 8.7 is not perfect. Equation (8.17) relates the image gradient and the motion field at the *same* image point, thereby establishing a constraint on an *infinitely small* spatial support; instead, Figure 8.7 describes a state of affairs over a *small but finite* spatial region. This immediately suggests that a possible strategy for solving the aperture problem is to look at the spatial and temporal variations of the image brightness over a neighborhood of each point.[10]

---

[10] Incidentally, this strategy appears to be adopted by the visual system of primates.

8.3.3 The Validity of the Constancy Equation: Optical Flow

How well does (8.17) estimate the normal component of the motion field? To answer this question, we can look at the difference, Δv, between the true value and the one estimated by the equation. To do this, we must introduce a model of image formation, accounting for the reflectance of the surfaces and the illumination of the scene.

For the purposes of this discussion, we restrict ourselves to a Lambertian surface, S, illuminated by a pointwise light source infinitely far away from the camera (Chapter 2). Therefore, ignoring photometric distorsion, we can write the image brightness, E, as

$$E = \rho \mathbf{I}^\top \mathbf{n}, \tag{8.19}$$

where ρ is the surface albedo, $\mathbf{I}$ identifies the direction and intensity of illumination, and $\mathbf{n}$ is the unit normal to S at $\mathbf{P}$.

Let us now compute the total temporal derivative of both sides of (8.19). The only quantity that depends on time on the right hand side is the normal to the surface. If the surface is moving relative to the camera with translational velocity $\mathbf{T}$ and angular velocity ω, the orientation of the normal vector $\mathbf{n}$ will change according to

$$\frac{d\mathbf{n}}{dt} = \omega \times \mathbf{n}, \tag{8.20}$$

where $\times$ indicates vector product. Therefore, taking the total temporal derivative of both sides of (8.19), and using (8.17) and (8.20), we have

$$\nabla E^\top \mathbf{v} + E_t = \rho \mathbf{I}^\top (\omega \times \mathbf{n}). \tag{8.21}$$

We can obtain the desired expression for Δv from (8.18) and (8.21):

$$|\Delta v| = \rho \frac{|\mathbf{I}^\top \omega \times \mathbf{n}|}{\|\nabla E\|}.$$

We conclude that, even under the simplifying assumption of Lambertian reflectance, the image brightness constancy equation yields the true normal component of the motion field (that is, $|\Delta v|$ is identically 0 for every possible surface) only for (a) purely translational motion, or (b) for any rigid motion such that the illumination direction is parallel to the angular velocity.

Other factors being equal, the difference Δv decreases as the magnitude of the spatial gradient increases; this suggests that *points with high spatial image gradient are the locations at which the motion field can be best estimated by the image brightness constancy equation.*

In general, $|\Delta v|$ is unlikely to be identically zero, and *the apparent motion of the image brightness is almost always different from the motion field*. For this reason, to avoid confusion, we call the apparent motion an *optical flow*, and refer to techniques estimating the motion field from the image brightness constancy equation as *optical flow techniques*. Here is a summary of similarities and differences between motion field and optical flow.

Definition: Optical Flow

The *optical flow* is a vector field subject to the constraint (8.17), and loosely defined as the *apparent motion* of the image brightness pattern.

Optical Flow and Motion Field

The optical flow is the *approximation of the motion field* which can be computed from time-varying image sequences. Under the simplifying assumptions of

- Lambertian surfaces
- pointwise light source at infinity
- no photometric distortion

the *error* of this approximation is

- *small* at points with high spatial gradient
- *exactly zero* only for translational motion or for any rigid motion such that the illumination direction is parallel to the angular velocity

We are now ready to learn algorithms estimating the motion field.

8.4 Estimating the Motion Field

The estimation of the motion field is a useful starting point for the solution of many motion problems. The many techniques devised by the computer vision community can be roughly divided into two major classes: *differential techniques* and *matching techniques*. Differential techniques are based on the spatial and temporal variations of the image brightness at all pixels, and can be regarded as methods for computing optical flow. Matching techniques, instead, estimate the disparity of special image points (features) between frames. We examine differential techniques in section 8.4.1; matching is the theme of section 8.4.2.

8.4.1 Differential Techniques

In recent (and not so recent) years a large number of differential techniques for computing optical flow have been proposed. Some of them require the solution of a system of partial differential equations, others the computation of second and higher-order derivatives of the image brightness, others again least-squares estimates of the parameters characterizing the optical flow. Methods in the latter class have at least two advantages over those in the first two:

- They are not iterative; therefore, they are genuinely local, and less biased than iterative methods by possible discontinuities of the motion field.
- They do not involve derivatives of order higher than the first; therefore, they are less sensitive to noise than methods requiring higher-order derivatives.

We describe a differential technique that gives good results. The basic assumption is that the motion field is well approximated by a *constant* vector field, $\mathbf{v}$, within any small region of the image plane.[11]

Assumptions

1. The image brightness constancy equation yields a good approximation of the normal component of the motion field.
2. The motion field is well approximated by a *constant* vector field within any small patch of the image plane.

An Optical Flow Algorithm. Given Assumption 1, for each point $\mathbf{p}_i$ within a small, $N \times N$ patch, Q, we can write

$$(\nabla E)^\top \mathbf{v} + E_t = 0$$

where the spatial and temporal derivatives of the image brightness are computed at $\mathbf{p}_1, \mathbf{p}_2 \ldots \mathbf{p}_{N^2}$.

☞ A typical size of the "small patch" is 5×5.

Therefore, the optical flow can be estimated within Q as the constant vector, $\bar{\mathbf{v}}$, that minimizes the functional

$$\Psi[\mathbf{v}] = \sum_{\mathbf{p}_i \in Q} \left[(\nabla E)^\top \mathbf{v} + E_t \right]^2 .$$

The solution to this least squares problem can be found by solving the linear system

$$A^\top A \mathbf{v} = A^\top \mathbf{b}. \tag{8.22}$$

The i-th row of the $N^2 \times 2$ matrix A is the spatial image gradient evaluated at point $\mathbf{p}_i$:

$$A = \begin{bmatrix} \nabla E(\mathbf{p}_1) \\ \nabla E(\mathbf{p}_2) \\ \cdot \\ \cdot \\ \cdot \\ \nabla E(\mathbf{p}_{N \times N}) \end{bmatrix}, \tag{8.23}$$

and $\mathbf{b}$ is the N^2-dimensional vector of the partial temporal derivatives of the image brightness, evaluated at $\mathbf{p}_1, \ldots \mathbf{p}_{N^2}$, after a sign change:

$$\mathbf{b} = -\left[E_t(\mathbf{p}_1), \ldots, E_t(\mathbf{p}_{N \times N}) \right]^\top . \tag{8.24}$$

---

[11] Notice that this is in agreement with the first conclusion of section 8.2.3 (motion field of moving planes) regarding the approximation of smooth motion fields.

The least squares solution of the overconstrained system (8.22) can be obtained as[12]

$$\bar{\mathbf{v}} = (A^\top A)^{-1} A^\top \mathbf{b}. \tag{8.25}$$

$\bar{\mathbf{v}}$ is the optical flow (the estimate of the motion field) at the center of patch Q; repeating this procedure for all image points, we obtain a dense optical flow. We summarize the algorithm as follows:

Algorithm CONSTANT_FLOW

The input is a time-varying sequence of n images, $E_1, E_2, \ldots E_n$. Let Q be a square region of $N \times N$ pixels (typically, $N = 5$).

1. Filter each image of the sequence with a Gaussian filter of standard deviation equal to σ_s (typically $\sigma_s = 1.5$ pixels) along each spatial dimension.
2. Filter each image of the sequence along the temporal dimension with a Gaussian filter of standard deviation σ_t (typically $\sigma_t = 1.5$ frames). If $2k + 1$ is the size of the temporal filter, leave out the first and last k images.
3. For each pixel of each image of the sequence:
   (a) compute the matrix A and the vector $\mathbf{b}$ using (8.23) and (8.24)
   (b) compute the optical flow using (8.25)

The output is the optical flow computed in the last step.

☞ The purpose of spatial filtering is to attenuate noise in the estimation of the spatial image gradient; temporal filtering prevents aliasing in the time domain. For the implementation of the temporal filtering, imagine to stack the images one on top of the other, and filter sequences of pixels having the same coordinates. Note that the size of the temporal filter is linked to the maximum speed that can be "measured" by the algorithm.

An Improved Optical Flow Algorithm. We can improve CONSTANT_FLOW by observing that the error made by approximating the motion field at $\mathbf{p}$ with its estimate at the center of a patch increases with the distance of $\mathbf{p}$ from the center itself. This suggests a *weighted* least-square algorithm, in which the points close to the center of the patch are given more weight than those at the periphery. If W is the weight matrix, the solution, $\bar{\mathbf{v}}_w$, is given by

$$\bar{\mathbf{v}}_w = (A^\top W^2 A)^{-1} A^\top W^2 \mathbf{b}.$$

Concluding Remarks on Optical Flow Methods. It is instructive to examine the image locations at which CONSTANT_FLOW fails. As we have seen in Chapter 4, the 2×2 matrix

$$A^\top A = \begin{pmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{pmatrix}, \tag{8.26}$$

[12] See Appendix, section A.6 for alternative ways of solving overconstrained linear systems.

computed over an image region Q, is singular if and only if all the spatial gradients in Q are null or parallel. In this case the aperture problem cannot be solved, and the only possibility is to pick the solution of minimum norm, that is, the normal flow. The fact that we have already met the matrix $A^\top A$ in Chapter 4 is not a coincidence; the next section tells you why.

Notice that CONSTANT_FLOW gives good results because the spatial structure of the motion field of a rigid motion is well described by a low-degree polynomial in the image coordinates (as shown in section 8.2.3). For this reason, the assumption of local constancy of the motion field over small image patches is quite effective.

8.4.2 Feature-based Techniques

The second class of methods for estimating the motion field is formed by so-called *matching techniques*, which estimate the motion field at feature points only. The result is a sparse motion field. We start with a two-frame analysis (finding feature disparities between consecutive frames), then illustrate how *tracking* the motion of a feature across a long image sequence can improve the robustness of frame-to-frame matching.

Two-Frame Methods: Feature Matching. If motion analysis is restricted to two consecutive frames, the same matching methods can be used for stereo and motion.[13] This is true for both correlation-based and feature-based methods (Chapter 7). Here we concentrate on *matching feature points*. You can easily adapt this method for the stereo case too.

The point-matching method we describe is reminiscent of the CONSTANT_FLOW algorithm, and based on the features we met in Chapter 4. There, we looked at the matrix $A^\top A$ of (8.26), computed over small, square image regions: the features were the centers of those regions for which the smallest eigenvalue of $A^\top A$ was larger than a threshold. The idea of our matching method is simple: compute the displacement of such feature points by iterating algorithm CONSTANT_FLOW.

The procedure consists of three steps. First, the uniform displacement of the square region Q is estimated through CONSTANT_FLOW, and added to the current displacement estimate (initially set to 0). Second, the patch Q is *warped* according to the estimated flow. This means that Q is displaced according to the estimated flow, and the resulting patch, Q', is resampled in the pixel grid of frame I_2. If the estimated flow equals (v_x, v_y), the gray value at pixel (i, j) of Q' can be obtained from the gray values of the pixels of Q close to $(i - v_y, j - v_x)$. For our purpose, bilinear interpolation[14] is sufficient. Third, the first and second steps are iterated until a stopping criterion is met. Here is the usual algorithm box, containing an example of stopping criterion.

[13] But keep in mind the discussion of section 8.2.1 on the differences between stereo and motion disparities.

[14] Bilinear interpolation means that the interpolation is linear in each of the four pixels closest to $(i - v_y, j - v_x)$.