CHAPTER

Motion

20

Motion is another aspect of 3D vision that humans are able to interpret with ease. This chapter studies the basic theoretical concepts: it is left to Chapters 22 and 23 to apply them to real problems where motion is crucial, including the monitoring of traffic flow and the tracking of people.

Look out for:

- The basic concepts of optical flow, and its limitations
- The idea of a focus of expansion, and how it leads to the possibility of "structure from motion"
- How motion stereo is achieved
- The important status of the Kalman filter in motion applications
- The ways in which invariant features may be used for wide baseline matching.

Note that this introductory chapter on 3D motion leads to important methods for performing vital surveillance tasks—as will be seen in Chapters 22 and 23.

20.1 INTRODUCTION

This chapter is concerned with the analysis of motion in digital images. In the space available, it will not be possible to cover the whole subject comprehensively: Instead the aim will be to give the flavor of the subject, airing some of the principles that have proved important over the past two or three decades. Over much of this time, optical flow has been topical: It is appropriate to study it in fair detail, because of its importance for surveillance and other applications. Later in the chapter, the use of the Kalman filter for tracking moving objects will be discussed, and the use of invariant features such as SIFT for wide baseline matching, also relevant to motion tracking, will be covered.

Computer Vision. DOI: http://dx.doi.org/10.1016/B978-0-12-809284-2.00020-4

20.2 OPTICAL FLOW

When scenes contain moving objects, analysis is necessarily more complex than for scenes where everything is stationary, since temporal variations in intensity have to be taken into account. However, intuition suggests that it should be possible—even straightforward—to segment moving objects by virtue of their motion: Image differencing over successive pairs of frames should permit this to be achieved. More careful consideration shows that things are not quite so simple, as illustrated in Fig. 20.1. The reason is that regions of constant intensity give no sign of motion, while edges parallel to the direction of motion also give the appearance of not moving: only edges with a component normal to the direction of motion carry information about the motion. In addition, there is some ambiguity in the direction of the velocity vector. This arises partly because there is too little information available within a small aperture to permit the full velocity vector to be computed (Fig. 20.2): This is hence called the *aperture problem*.

These elementary ideas can be taken further, and they lead to the notion of optical flow, wherein a local operator which is applied at all pixels in the image

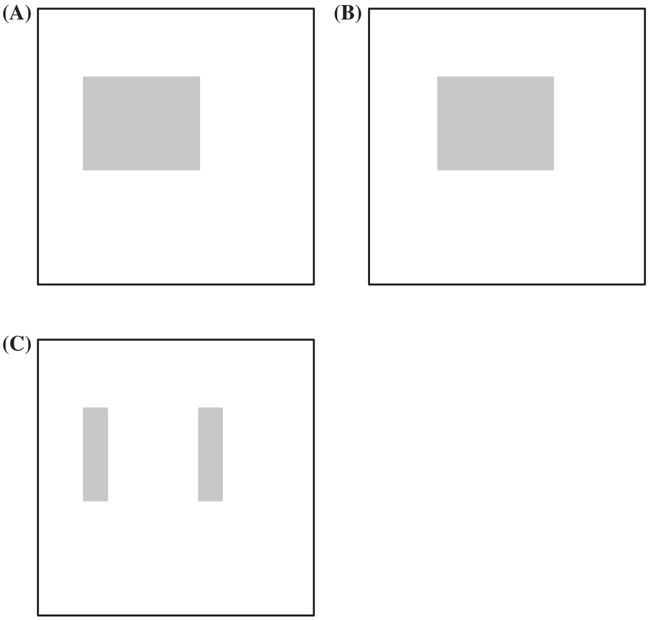

FIGURE 20.1

Effect of image differencing. This figure shows an object which has moved between frames (A) and (B). (C) shows the result of performing an image differencing operation. Note that the edges parallel to the direction of motion do not show up in the difference image. Also, regions of constant intensity give no sign of motion.

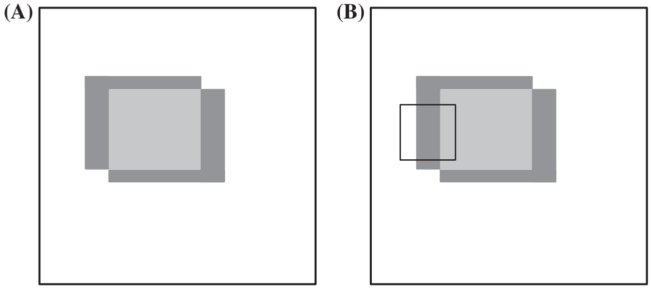

FIGURE 20.2

The aperture problem. This figure illustrates the aperture problem. (A) shows (dark gray) regions of motion of an object whose central uniform region (light gray) gives no sign of motion. (B) shows how little is visible in a small aperture (black border), thereby leading to ambiguity in the deduced direction of motion of the object.

will lead to a motion vector field which varies smoothly over the whole image. The attraction lies in the use of a local operator, with its limited computational burden. Ideally, it would have an overhead comparable to an edge detector in a normal intensity image—though clearly, it will have to be applied locally to pairs of images in an image sequence.

We start by considering the intensity function $I(x, y, t)$ and expanding it in a Taylor series:

$$I(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) = I(x, y, t) + I_x\mathrm{d}x + I_y\mathrm{d}y + I_t\mathrm{d}t + \cdots \tag{20.1}$$

where second and higher order terms have been ignored. In this formula, I_x, I_y, I_t denote respective partial derivatives with respect to x, y, and t.

We next set the local condition that the image has shifted by amount $(\mathrm{d}x, \mathrm{d}y)$ in time $\mathrm{d}t$ so that it is functionally identical at $(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t)$ and (x, y, t):

$$I(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) = I(x, y, t) \tag{20.2}$$

Hence, we can deduce:

$$I_t = -\left(I_x\dot{x} + I_y\dot{y}\right) \tag{20.3}$$

Writing the local velocity **v** in the form:

$$\mathbf{v} = \left(v_x, v_y\right) = (\dot{x},\ \dot{y}) \tag{20.4}$$

we find:

$$I_t = -\left(I_xv_x + I_yv_y\right) = -\nabla I.\mathbf{v} \tag{20.5}$$

I_t can be measured by subtracting pairs of images in the input sequence, while ∇I can be estimated by Sobel or other gradient operators. Hence, it should be possible to deduce the velocity field $\mathbf{v}(x,y)$ using the above equation. Unfortunately, this equation is a scalar equation and will not suffice for determining the two local components of the velocity field as we require. There is a further problem with this equation—that the velocity value will depend on the values of both I_t and ∇I, and these quantities are only estimated approximately by the respective differencing operators: In both cases, significant noise will arise, and this will be exacerbated by taking the ratio in order to calculate **v**.

Let us now return to the problem of computing the full velocity field $\mathbf{v}(x, y)$. All we know about **v** is that its components lie on the following line in (v_x, v_y) space (Fig. 20.3):

$$I_x v_x + I_y v_y + I_t = 0 \tag{20.6}$$

This line is normal to the direction (I_x, I_y) and has a distance from the (velocity) origin which is equal to

$$|\mathbf{v}| = -I_t / \left(I_x^2 + I_y^2\right)^{1/2} \tag{20.7}$$

Clearly, we need to deduce the component of **v** along the line given by Eq. (20.6). However, there is no purely local means of achieving this with first derivatives of the intensity function. The accepted solution (Horn and Schunck, 1981) is to use relaxation labeling to arrive iteratively at a self-consistent solution

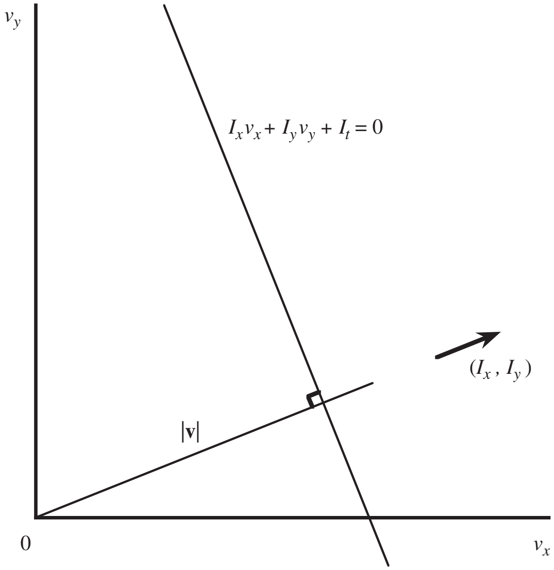

FIGURE 20.3

Computation of the velocity field. This graph shows the line in velocity space on which the velocity vector **v** must lie. The line is normal to the direction (I_x, I_y) and its distance from the origin is known to be $|\mathbf{v}|$ (see text).

which minimizes the global error. In principle, this approach will also minimize the noise problem indicated earlier.

In fact, there are still problems with the method. Essentially, these arise as there are liable to be vast expanses of the image where the intensity gradient is low. In that case, only very inaccurate information is available about the velocity component parallel to ∇I, and the whole problem becomes ill-conditioned. On the other hand, in a highly textured image, this situation should not arise (assuming that the texture has a large enough grain size to give good differential signals).

Finally, we return to the idea mentioned at the beginning of this section—that edges parallel to the direction of motion would not give useful motion information. Such edges will have edge normals normal to the direction of motion, so ∇I will be normal to $\mathbf{v}$. Thus, from Eq. (20.5), I_t will be zero. In addition, regions of constant intensity will have $\nabla I = 0$, so again I_t will be zero. It is interesting and highly useful that such a simple Eq. (20.5) embodies all the cases that were suggested earlier on the basis of intuition.

In what follows we assume that the optical flow (velocity field) image has been computed satisfactorily, i.e., without the disadvantages of inaccuracy or ill-conditioning. It must now be interpreted in terms of moving objects and in some cases a moving camera. In fact, we shall ignore motion of the camera by remaining within its frame of reference.

20.3 INTERPRETATION OF OPTICAL FLOW FIELDS

We start by considering a case where no motion is visible. In that case, the velocity field image contains only vectors of zero length (Fig. 20.4A). Next, we take a case where one object is moving towards the right, with a simple effect on the velocity field image (Fig. 20.4B). Next, we consider the case where the camera is moving forwards; in this case, all the stationary objects in the field of view appear to be diverging from a point which is called the *focus of expansion* (FoE)—see Fig. 20.4C; this image also shows an object which is moving rapidly past the camera, and which has its own separate FoE. Fig. 20.4D shows the case of an object moving directly towards the camera: In this case, its FoE lies within its outline. Similarly, objects which are receding appear to move away from the *focus of contraction*. Next, there are objects which are stationary but which are rotating about the line of sight: For these, the vector field appears as in Fig. 20.4E. There is a final case which is also quite simple—that of an object which is stationary but rotating about an axis normal to the line of sight; if the axis is horizontal, then the features on the object will appear to be moving up or down, while paradoxically the object itself remains stationary (Fig. 20.4F)—though its outline could oscillate as it rotates.

So far, we have only dealt with cases in which pure translational or pure rotational motion is occurring. If a rotating meteor is rushing past, or a spinning cricket

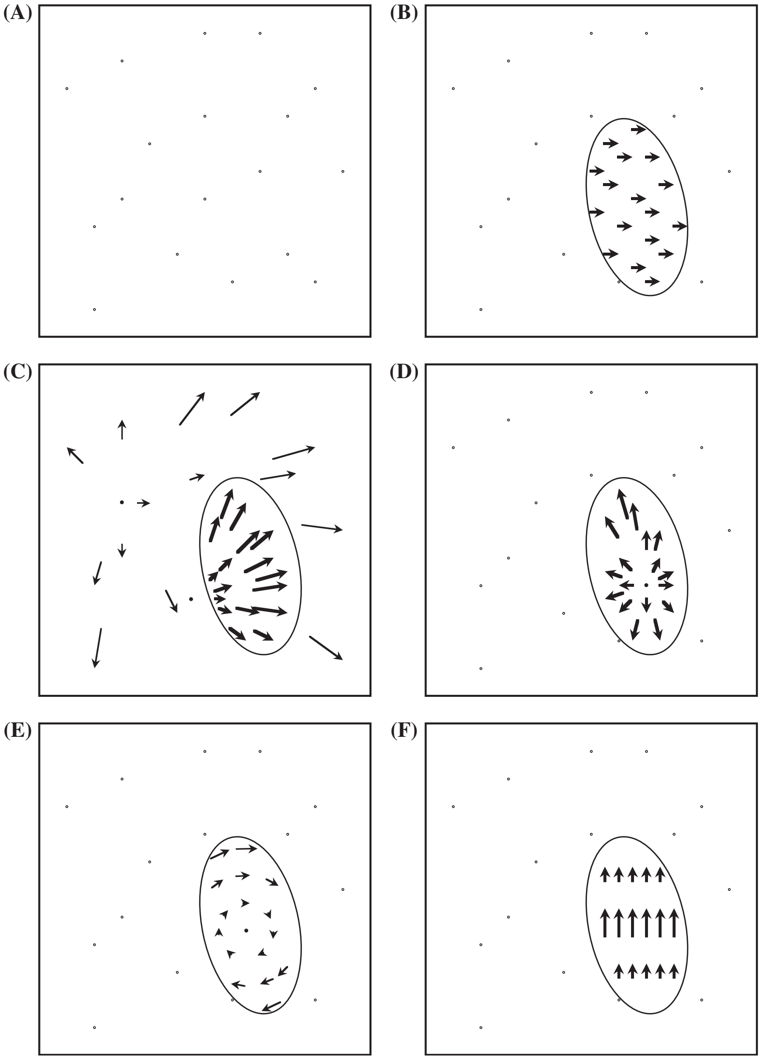

FIGURE 20.4

Interpretation of velocity flow fields. (A) shows a case where the object features all have zero velocity. (B) depicts a case where an object is moving to the right. (C) shows a case where the camera is moving into the scene, and the stationary object features appear to be diverging from a focus of expansion (FOE), while a single large object is moving past the camera and away from a separate FOE. In (D), an object is moving directly towards the camera which is stationary: The object's FOE lies within its outline. In (E), an object is rotating about the line of sight to the camera, and in (F), the object is rotating about an axis perpendicular to the line of sight. In all cases, the length of the arrow indicates the magnitude of the velocity vector.

ball is approaching, then both types of motion will occur together. In that case, unraveling the motion will be far more complex. We shall not solve this problem here but refer the reader to more specialized texts (e.g., Maybank, 1992). However, the complexity is due to the way depth (Z) creeps into the calculations. First, note that pure rotational motion with rotation about the line of sight does not depend on Z: All we have to measure is the angular velocity, and this can be done quite simply.

20.4 USING FOCUS OF EXPANSION TO AVOID COLLISION

We now take a simple case in which a FoE is located in an image and show how it is possible to deduce the distance of closest approach of the camera to a fixed object of known coordinates. This type of information is valuable for guiding robot arms or robot vehicles and helping to avoid collisions.

In the notation of Chapter 16, The Three-Dimensional World, we have the following formulae for the location of an image point (x, y, z) resulting from a world point (X, Y, Z):

$$x = fX/Z \tag{20.8}$$

$$y = fY/Z \tag{20.9}$$

$$z = f \tag{20.10}$$

Assuming that the camera has a motion vector $\left(-\dot{X}, -\dot{Y}, -\dot{Z}\right) = (-u, -v, -w)$, fixed world points will have velocity (u, v, w) relative to the camera. Now, a point (X_0, Y_0, Z_0) will after a time t appear to move to $(X, Y, Z) = (X_0 + ut, Y_0 + vt, Z_0 + wt)$ with image coordinates:

$$(x, y) = \left(\frac{f(X_0 + ut)}{Z_0 + wt}, \frac{f(Y_0 + vt)}{Z_0 + wt}\right) \tag{20.11}$$

and as $t \to \infty$, this approaches the FoE F (fu/w, fv/w). This point is in the image, but the true interpretation is that the actual motion of the center of projection of the imaging system is towards the point:

$$\mathbf{p} = \left(fu/w, fv/w, f\right) \tag{20.12}$$

(This is of course consistent with the motion vector (u, v, w) assumed initially.) The distance moved during time t can now be modeled as

$$\mathbf{X}_c = (X_c, Y_c, Z_c) = \alpha t\, \mathbf{p} = f\alpha\, t\left(u/w, v/w, 1\right) \tag{20.13}$$

where α is a normalization constant. To calculate the distance of closest approach of the camera to the world point $\mathbf{X} = (X, Y, Z)$, we merely specify that the vector $\mathbf{X}_c - \mathbf{X}$ be perpendicular to $\mathbf{p}$ (Fig. 20.5) so that

$$(\mathbf{X}_c - \mathbf{X}) \cdot \mathbf{p} = 0 \tag{20.14}$$

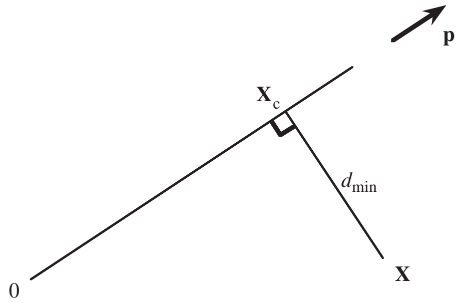

FIGURE 20.5

Calculation of distance of closest approach. Here, the camera is moving from 0 to $\mathbf{X}_c$ in the direction **p**, not in a direct line to the object at **X**. $d_{\min}$ is the distance of closest approach.

That is,

$$(\alpha t\,\mathbf{p} - \mathbf{X})\cdot\mathbf{p} = 0 \tag{20.15}$$

$$\therefore \quad \alpha t\,\mathbf{p}\cdot\mathbf{p} = \mathbf{X}\cdot\mathbf{p} \tag{20.16}$$

$$\therefore \quad t = (\mathbf{X}\cdot\mathbf{p})/\alpha(\mathbf{p}\cdot\mathbf{p}) \tag{20.17}$$

Substituting in the equation for $\mathbf{X}_c$ now gives:

$$\mathbf{X}_c = \mathbf{p}(\mathbf{X}\cdot\mathbf{p})/(\mathbf{p}\cdot\mathbf{p}) \tag{20.18}$$

Hence, the minimum distance of approach is given by

$$\begin{aligned} d_{\min}^2 &= \left[\frac{\mathbf{p}(\mathbf{X}\cdot\mathbf{p})}{(\mathbf{p}\cdot\mathbf{p})} - \mathbf{X}\right]^2 = \frac{(\mathbf{X}\cdot\mathbf{p})^2}{(\mathbf{p}\cdot\mathbf{p})} - \frac{2(\mathbf{X}\cdot\mathbf{p})^2}{(\mathbf{p}\cdot\mathbf{p})} + (\mathbf{X}\cdot\mathbf{X}) \\ &= (\mathbf{X}\cdot\mathbf{X}) - \frac{(\mathbf{X}\cdot\mathbf{p})^2}{(\mathbf{p}\cdot\mathbf{p})} \end{aligned} \tag{20.19}$$

which is naturally zero when **p** is aligned along **X**. Clearly, avoidance of collisions requires an estimate of the size of the machine (e.g., robot or vehicle) attached to the camera and the size to be associated with the world point feature **X**. Finally, note that while **p** is obtained from the image data, **X** can only be deduced from the image data if the depth *Z* can be estimated from other information. In fact, this information should be available from time-to-adjacency analysis (see below) if the speed of the camera through space (and specifically *w*) is known.

20.5 TIME-TO-ADJACENCY ANALYSIS

Next, we consider the extent to which the depths of objects can be deduced from optical flow. First, note that features on the same object share the same FoE, and this can help us to identify them. But how can we get information on the depths of the various features on the object from optical flow? The basic approach is to start with the coordinates of a general image point (x, y), deduce its flow velocity, and then find an equation linking this with the depth Z.

Taking the general image point (x, y) given in Eq. (20.11), we find:

$$\begin{aligned}\dot{x} &= f[(Z_0 + wt)u - (X_0 + ut)w]/(Z_0 + wt)^2 \\ &= f(Zu - Xw)/Z^2\end{aligned} \tag{20.20}$$

and

$$\dot{y} = f(Zv - Yw)/Z^2 \tag{20.21}$$

Hence:

$$\begin{aligned}\dot{x}/\dot{y} &= (Zu - Xw)/(Zv - Yw) = \left(u/w - X/Z\right)/\left(v/w - Y/Z\right) \\ &= (x - x_F)/(y - y_F)\end{aligned} \tag{20.22}$$

This result was to be expected, as the motion of the image point has to be directly away from the FoE (x_F, y_F). With no loss of generality, we now take a set of axes such that the image point considered is moving along the x-axis. Then, we have:

$$\dot{y} = 0 \tag{20.23}$$

$$y_F = y = fY/Z \tag{20.24}$$

Defining the distance from the FoE as Δr (see Fig. 20.6), we find:

$$\Delta r = \Delta x = x - x_F = fX/Z - fu/w = f(Xw - Zu)/Zw \tag{20.25}$$

$$\therefore \quad \Delta r/\dot{r} = \Delta x/\dot{x} = -Z/w \tag{20.26}$$

What this equation means is that the *time to adjacency*, when the origin of the camera coordinate system will arrive at the object point, is the same (Z/w) when seen in real-world coordinates as when seen in image coordinates ($-\Delta r/\dot{r}$). Hence, it is possible to relate the optical flow vectors for object points at different depths in the scene. This is important, as the assumption of identical values of w now allows us to determine the relative depths of object points merely from their apparent motion parameters:

$$\frac{Z_1}{Z_2} = \frac{\Delta r_1/\Delta r_2}{\dot{r}_1/\dot{r}_2} \tag{20.27}$$

This is thus the first step in the determination of structure from motion. In this context, note how the implicit assumption that the objects under observation are rigid is included—namely that all points on the same object are characterized by identical values of w. The assumption of rigidity underlies much of the work on interpretation of motion in images.

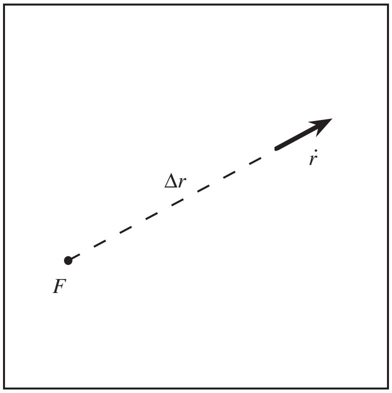

FIGURE 20.6

Calculation of time to adjacency. Here, an object feature is moving directly away from the focus of expansion F with speed $\dot{r}$. At the time of observation, the distance of the feature from F is Δr. These measurements permit the time to adjacency and hence also the relative depth of the feature to be calculated.

20.6 BASIC DIFFICULTIES WITH THE OPTICAL FLOW MODEL

When the optical flow ideas presented above are tried on real images, certain problems arise which are not apparent from the above model. First, not all edge points which should appear in the motion image are actually present. This is due to the contrast between the moving object and the background vanishing locally and limiting visibility. The situation is exactly as for edges which are located by edge-detection operators in nonmoving images: The contrast simply drops to a low value in certain localities and the edge peters out. This signals that the edge model, and now the velocity flow model, is limited and such local procedures are ad hoc and too impoverished to permit proper segmentation unaided.

Here, we take the view that simple models can be useful, but they become inadequate on certain occasions and robust methods are required to overcome the problems that then arise. Some of the problems were noticed by Horn as early as 1986. First, a smooth sphere may be rotating but the motion will not show up in an optical flow (difference) image. We can if we wish regard this is a simple optical illusion, as the rotation of the sphere may well be invisible to the eye too. Second, a motionless sphere may *appear* to rotate as the light rotates around it: The object is simply subject to the laws of Lambertian optics, and again we may if we wish regard this effect is an optical illusion. (The illusion is relative to the baseline provided by the *normally correct* optical flow model.)

We next return to the optical flow model and see where it could be wrong or misleading. The answer is at once apparent: We stated in writing Eq. (20.2) that we were assuming that the *image* is being shifted. Yet, it is not images that shift but the objects imaged within them. Thus, we ought to be considering the images of objects moving against a fixed background (or a variable background if the camera is moving). This will then permit us to see how sections of the motion edge can go from high to low contrast and back again in a rather fickle way, which we must nevertheless allow for in our algorithms. With this in mind, it should be permissible to go on using optical flow and difference imaging, even though these concepts have distinctly limited theoretical validity. (For a more thoroughgoing analysis of the underlying theory, see Faugeras, 1993.)

20.7 STEREO FROM MOTION

An interesting aspect of camera motion is that over time, the camera sees a succession of images that span a baseline in a similar way to binocular (stereo) images. Thus, it should be possible to obtain depth information by taking two such images and tracking object features between them. The technique is in principle more straightforward than normal stereo imaging in that feature tracking is possible, so the correspondence problem should be nonexistent. However, there is a difficulty in that the object field is viewed from almost the same direction in the succession of images so that the full benefit of the available baseline is not obtained (Fig. 20.7). We can analyze the effect as follows:

First, in the case of camera motion, the equations for lateral displacement in the image depend not only on X but also on Y, though we can make a simplification in the theory by working with R, the radial distance of an object point from the optical axis of the camera, where

$$R = \left(X^2 + Y^2\right)^{1/2} \tag{20.28}$$

We now obtain the radial distances in the two images as

$$r_1 = Rf/Z_1 \tag{20.29}$$

$$r_2 = Rf/Z_2 \tag{20.30}$$

So, the disparity is

$$D = r_2 - r_1 = Rf\left(1/Z_2 - 1/Z_1\right) \tag{20.31}$$

Writing the baseline as

$$b = Z_1 - Z_2 \tag{20.32}$$

and assuming that $b << Z_1, Z_2$, and then dropping the suffices, gives

$$D = Rbf/Z^2 \tag{20.33}$$

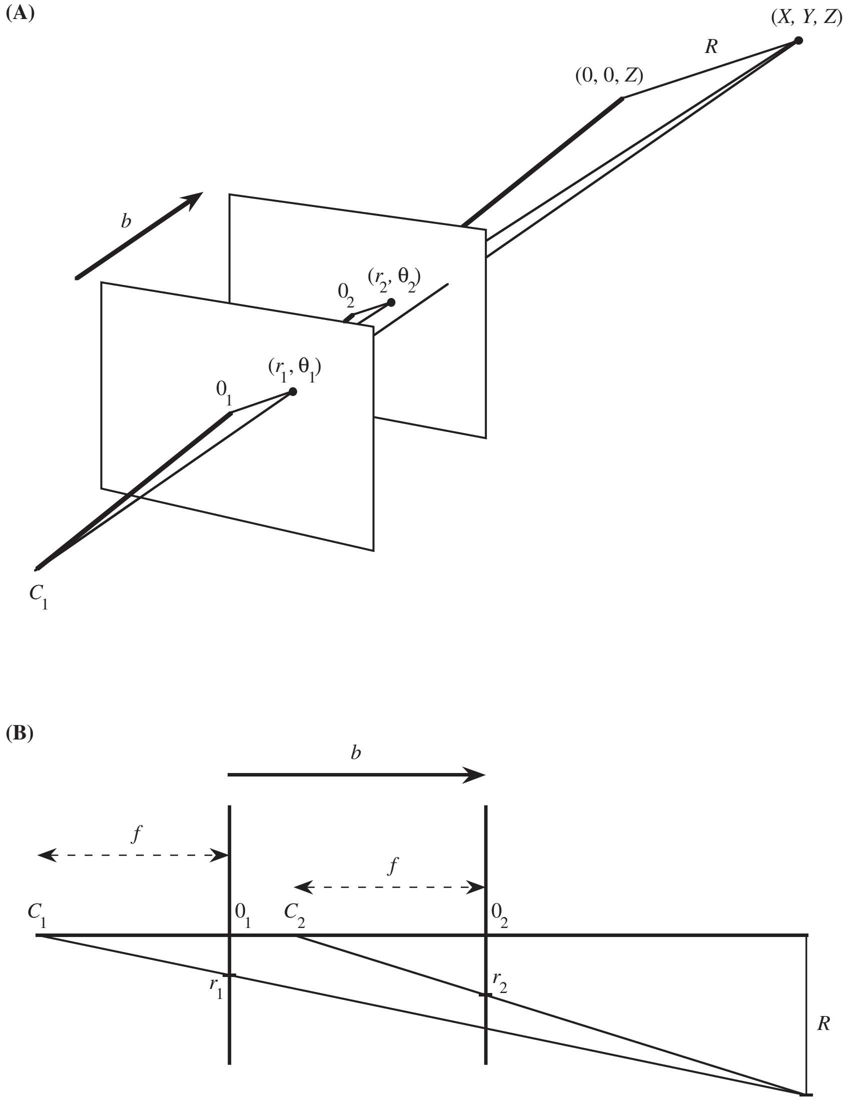

FIGURE 20.7

Calculation of stereo from camera motion. (A) shows how stereo imaging can result from camera motion, the vector **b** representing the baseline. (B) shows the simplified planar geometry required to calculate the disparity. It is assumed that the motion is directly along the optical axis of the camera.

While this would appear to mitigate against finding Z without knowing R, we can overcome this problem by observing that

$$R/Z = r/f \tag{20.34}$$

where r is approximately the mean value $\frac{1}{2}(r_1 + r_2)$. Substituting R now gives

$$D = br/Z \tag{20.35}$$

Hence, we can deduce the depth of the object point as

$$Z = br/D = br/(r_2 - r_1) \tag{20.36}$$

This equation should be compared with Eq. (15.5) representing the normal stereo situation. The important point to note is that for motion stereo, the disparity depends on the radial distance r of the image point from the optical axis of the camera, whereas for normal stereo, the disparity is independent of r; as a result, motion stereo gives no depth information for points on the optical axis, and the accuracy of depth information depends on the magnitude of r.

20.8 THE KALMAN FILTER

When tracking moving objects, it is desirable to be able to predict where they will be in future frames, as this will make maximum use of preexisting information and permit the least amount of search in the subsequent frames. It will also serve to offset the problems of temporary occlusion, such as when one vehicle passes behind another, or one person passes behind another, or even when one limb of a person passes behind another. (There are also many military needs for tracking prediction, and others on the sports field.) The obvious equations to employ for this purpose involve sequentially updating the position and the velocity of points on the object being tracked:

$$x_i = x_{i-1} + v_{i-1} \tag{20.37}$$

$$v_i = x_i - x_{i-1} \tag{20.38}$$

assuming for convenience a unit time interval between each pair of samples.

In fact, this approach is too crude to yield the best results. First, it is necessary to make three quantities explicit: (1) the raw measurements (e.g., x), (2) the best estimates of the values of the corresponding variables *before* observation (denoted by "$-$"), and (3) the best estimates of these same model parameters *following* observation (denoted by "$+$"). In addition, it is necessary to include explicit noise terms so that rigorous optimization procedures can be derived for making the best estimates.

In the particular case outlined above, the velocity—and possible variations on it which we shall ignore here for simplicity—constitutes a best estimate model parameter. We include position measurement noise by the parameter u and velocity (model) estimation noise by the parameter w. The above equations now become

$$x_i^- = x_{i-1}{}^+ + v_{i-1} + u_{i-1} \tag{20.39}$$

$$v_i^- = v_{i-1}{}^+ + w_{i-1} \tag{20.40}$$

In the case that the velocity is constant and the noise is Gaussian, we can spot the optimum solutions to this problem:

$$x_i^- = x_{i-1}{}^+ \tag{20.41}$$

$$\sigma_i^- = \sigma_{i-1}{}^+ \tag{20.42}$$

these being called the *prediction* equations, and

$$x_i^+ = \frac{x_i/\sigma_i^2 + \left(x_i^-\right)/\left(\sigma_i^-\right)^2}{1/\sigma_i^2 + 1/\left(\sigma_i^-\right)^2} \tag{20.43}$$

$$\sigma_i^+ = \left[\frac{1}{1/\sigma_i^2 + 1/\left(\sigma_i^-\right)^2}\right]^{1/2} \tag{20.44}$$

these being called the *correction* equations: these are nothing more than the well-known equations for weighted averages (Cowan, 1998). In these equations, $\sigma^{\pm}$ are the standard deviations for the respective model estimates $x^{\pm}$, and σ is the standard deviation for the raw measurement x.

What these equations show is how repeated measurements improve the estimate of the position parameter and the error upon it at each iteration. Notice the particularly important feature—that the noise is being modeled as well as the position itself. This permits all positions earlier than $i-1$ to be forgotten. The fact that there were many such positions, whose values can all be averaged to improve the accuracy of the latest estimate, is of course rolled up into the values of x_i^- and σ_i^-, and eventually into the values for x_i^+ and σ_i^+.

The next problem is how to generalize this result, both to multiple variables and to possibly varying velocity and acceleration. This is the function of the widely used Kalman filter. It achieves this by continuing with a linear approximation and by employing a state vector comprising position, velocity, and acceleration (or other relevant parameters), all in one state vector **s**. This constitutes the dynamic model. The raw measurements x have to be considered separately.

In the general case, the state vector is not updated simply by writing:

$$\mathbf{s}_i^- = \mathbf{s}_{i-1}{}^+ \tag{20.45}$$

but requires a fuller exposition because of the interdependence of position, velocity, and acceleration; hence, we have:

$$\mathbf{s}_i^- = K_i \mathbf{s}_{i-1}{}^+ \tag{20.46}$$

Some authors write K_{i-1} in this equation, but it is only a matter of definition whether the label matches the previous or the new state. Similarly, the standard deviations σ_i, $\sigma_i^{\pm}$ in Eqs. (20.42)–(20.44) (or rather, the corresponding variances) have to be replaced by the covariance matrices Σ_i, $\Sigma_i^{\pm}$, and the equations become significantly more complicated. We will not go into the calculations fully here as they are nontrivial and need several pages to iterate. Suffice it to say that the aim is to produce an optimum linear filter by a least-squares calculation (see, e.g., Maybeck, 1979).

Overall, the Kalman filter is the optimal estimator for a linear system for which the noise is zero mean, white and Gaussian, though it will often provide good estimates even if the noise is not Gaussian.

Finally, it will be noticed that the Kalman filter itself works by averaging processes which will give erroneous results if any outliers are present. This will certainly occur in most motion applications. Thus, there is a need to test each prediction to determine if it is too far away from reality. If this is the case, it is not unlikely that the object in question has become partially or fully occluded: A simple option is to assume that the object continues in the same motion (albeit with a larger uncertainty as time goes on), and to wait for it to emerge from behind another object. At the very least, it is prudent to keep a number of such possibilities alive for some time, but the extent of this will naturally vary from situation to situation and from application to application.

20.9 WIDE BASELINE MATCHING

The need for wide baseline matching was noted in Chapter 6, Corner, Interest Point and Invariant Feature Detection, where considerable discussion was included on detection of suitable invariant features (see Section 6.7 and its various subsections). The topic has been left until the present chapter, because it is relevant for both 3D vision and motion analysis, and the latter topic has only been covered in this chapter. The wide baseline scenario arises from situations where the same object is viewed from widely different directions, with the result that its appearance may change dramatically so that it may become extremely difficult to recognize. While narrow baseline stereo is the norm for depth estimation using two cameras, wide baselines are common in surveillance applications—e.g., where a pedestrian precinct is viewed by several independent cameras that are widely separated, as described in Chapter 22, Surveillance. They are also the norm when objects are being sought in image databases. However, one of the most likely situations when they occur is with objects that are in motion. While this may appear to be immaterial in surveillance or with driver assistance systems, because every pair of frames will give instances of narrow baseline stereo, it can easily happen that objects will be *temporarily occluded* and come back into view with different orientations or backgrounds; in addition, the *attention* of the software (like that of a human operator) may only be on part of the scene for part of the time. Hence, wide baseline viewing is bound to be a common consequence of motion. Overall, then, wide baseline matching techniques will be needed in a variety of instances of 3D viewing and motion tracking.

Chapter 6, Corner, Interest Point, and Invariant Feature Detection showed how features could be designed to cover a variety of wide baseline views as far apart as 50°. In these circumstances, an important factor in designing suitable feature detectors is to make them invariant to scale and affine distortions. However, that alone is not enough: The feature detectors must also provide descriptors of each feature that are sufficiently rich in information that matching between views is made as unambiguous as possible. In that way, wide baseline matching has a chance of being highly reliable. Lowe (2004) has found that reliable recognition of objects is possible with as few as three features. Indeed, it is highly important to aim to achieve this when images typically contain several thousand features which come from many different objects as well as background clutter. In this way, the number of false positives is reduced to minimal levels, and there is a high chance of detecting all objects of the chosen type in the input image. That this can be possible it underlined by the richness of Lowe's SIFT features whose descriptors contain 128 parameters. (As discussed in Chapter 6: Corner, Interest Point, and Invariant Feature Detection, features devised by other workers may contain fewer parameters, but in the end risk not working in all possible scenarios.)

Granted that wide baseline matching is desirable, and that SIFT and other features have rich descriptor sets, how should the matching actually be achieved? Ideally, all that is necessary is to compare the feature descriptors from each pair of images and find which ones match well, and which therefore lead to corecognition of objects in the two views. Clearly, the first requirement is a similarity test for pairs of features. Lowe (2004) achieved this using a nearest neighbor (Euclidean) distance measure in his 128-dimensional descriptor space. He then used a Hough transform to identify clusters of features giving the same interpretations of poses for objects appearing in the two images. Because of the relatively small number of inliers that may occur in this type of situation, he found that the Hough transform approach performed significantly better than RANSAC. Mikolajczyk and Schmid (2004) used a Mahalanobis distance measure for selecting the most similar descriptors to obtain a set of initial matches; they then used cross correlation to reject low-score matches; finally, they performed a robust estimation of the transformation between the two images using RANSAC. Tuytelaars and Van Gool (2004) developed this further, using semi-local constraints involving geometric consistency and photometric constraints to refine the selection of matches before (again) relying on RANSAC to perform the final robust estimation of poses. In contrast to the approaches outlined above, Bay et al. (2008) fed the descriptor information to a naïve Bayes classifier working on a "bag-of-words" representation (Dance et al., 2004) in order to perform object recognition. Bay et al. (2008) make no mention of determination of object pose in this application, which was targeted more at recognizing objects in an image database—though it could equally well have been targeted at repeated recognition of cars on the road, for which pose would not be especially relevant.

Overall, it is clear that the new regime of utilizing invariant feature detectors with rich descriptors of local image content forms a powerful approach to wide baseline object matching and takes much of the heat out of the subsequent algorithms.

20.10 CONCLUDING REMARKS

Early in this chapter, we described the formation of optical flow fields and showed how a moving object or a moving camera leads to a FoE. In the case of moving objects, the FoE can be used to decide whether a collision will occur. In addition, analysis of the motion taking account of the position of the FoE led to the possibility of determining structure from motion. Specifically, this can be achieved via time-to-adjacency analysis, which yields the relative depth in terms of the motion parameters measurable directly from the image. We then went on to demonstrate some basic difficulties with the optical flow model, which arise since the motion edge can have a wide range of contrast values, making it difficult to measure motion accurately. In practice, this means that larger time intervals may have to be employed to increase the motion signal. Otherwise, feature-based processing related to that of Chapter 11, The Generalized Hough Transform can be used. Corners are the features which are the most widely used for this purpose, because of their ubiquity and because they are highly localized in 3D. Space prevents details of this approach from being described here: Details may be found in Barnard and Thompson (1980), Scott (1988), Shah and Jain (1984), and Ullman (1979). However, the value of the Kalman filter for alleviating the difficulties of temporary occlusion has been considered, and the use of invariant features for wide baseline matching (which includes motion tracking applications) has been covered.

Further work on motion as it arises in real applications will be dealt with in Chapters 22 and 23, which address the problems of surveillance and in-vehicle vision systems.

> *The obvious way to understand motion is by image differencing and determination of optical flow. This chapter has shown that the "aperture problem" is a difficulty that is avoidable by using corner tracking. Further difficulties are caused by temporary occlusions, thus necessitating techniques such as occlusion reasoning and Kalman filtering.*

20.11 BIBLIOGRAPHICAL AND HISTORICAL NOTES

Optical flow has been investigated by many workers over a good many years: See, e.g., Horn and Schunck (1981) and Heikkonen (1995). A definitive account of the mathematics relating to FoE appeared in 1980 (Longuet-Higgins and Prazdny, 1980). In fact, foci of expansion can be obtained either from the optical flow field or directly (Jain, 1983). The results of Section 20.5 on time-to-adjacency analysis stem originally from the work of Longuet-Higgins and Prazdny (1980) which provides some deep insights into the whole problem of optical flow and the possibilities of using its shear components. Note that numerical solution of the velocity field problem is not trivial; typically, least-squares analysis is required to overcome the effects of measurement inaccuracies and noise, and to finally obtain the required position measurements and motion parameters (Maybank, 1986). Overall, resolving ambiguities of interpretation is one of the main problems and challenges of image sequence analysis [see Longuet-Higgins (1984) for an interesting analysis of ambiguity in the case of a moving plane].

Unfortunately, the substantial and important literature on motion, image sequence analysis, and optical flow, which impinges heavily on 3D vision, could not be discussed in detail here for reasons of space. For seminal work on these topics, see, e.g., Huang (1983), Jain (1983), Nagel (1983, 1986), and Hildreth (1984).

For early work on the use of Kalman filters for tracking, see Marslin et al. (1991). For the huge amount of more recent work on tracking and surveillance of moving objects, including the tracking of people and vehicles, see Chapters 22 and 23 (in fact, Chapter 23: In-Vehicle Vision Systems is especially concerned with monitoring moving objects from within vehicles). For recent references on tracking, particle filters, and detection of moving objects, see the bibliographies in Chapters 22 and 23.

For further references on invariant features for wide baseline matching, see Chapter 6, Corner, Interest Point, and Invariant Feature Detection.

20.12 PROBLEM

1. Explain why, in Eq. (20.44), the variances are combined in this particular way (in most applications of statistics, variances are combined by addition).