CHAPTER 34

Optic Flow: Basics

Here is a useful trick, widely familiar to flying animals. Compute some measure S of the size of an object in an image. You could use diameter, area, or a similar measure. Move the camera, or eye and compute the time derivative of that measure, S_t. The expression S/S_t is an estimate of time to contact *which is unaffected by calibration* (Figure 34.1; **exercises**). If this number is less than some threshold which has to do with your ability to accelerate, etc. you're in trouble.

This trick is a simple manifestation of the information that *optic flow* (or, on occasion, *optical flow*) – the movement of pixels in an image caused by camera or scene movement – has to offer about the movement of an agent in the world. This chapter describes some basic properties of optic flow; Chapter 35 describes elementary procedures for estimating flow; and Chapter **??** describes learned methods to estimate flow.

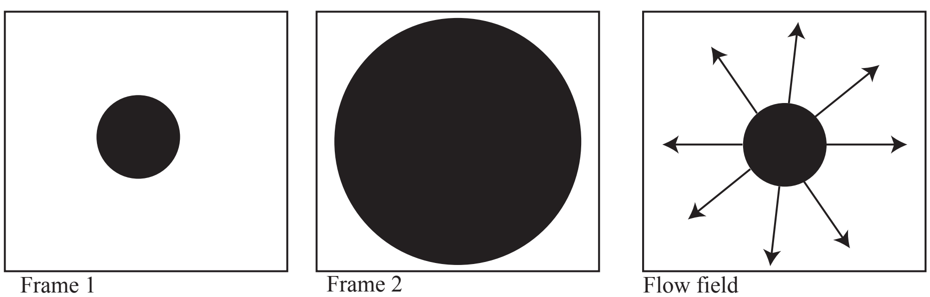

FIGURE 34.1: *Images 1 and 2 here offer an alarming warning of an impending collision, which is captured in the flow field. In my experience, you can startle an audience with carefully constructed slides showing scenes like these two. A darkened room and a bright projector seem to help.*

34.1 OPTIC FLOW AS A CUE TO GEOMETRY AND MOTION

A point in 3D that projects to $\mathbf{x}_1$ in frame 1 moves to $\mathbf{x}_2$ in frame 2. You can visualize this movement by placing the *flow vector* $\mathbf{v}(\mathbf{x}) = \mathbf{x}_2 - \mathbf{x}_1$ at each location in frame 1.

Definition: 34.1 *Optic Flow*

You have two images $\mathcal{I}_1$, $\mathcal{I}_2$, separated in time. A point in 3D that projects to $\mathbf{x}_1$ in the first image projects to $\mathbf{x}_2$ in the second. The optic flow at $\mathbf{x}_1$ in frame 1 is $\mathbf{v}(\mathbf{x}_1) = \mathbf{x}_2 - \mathbf{x}_1$

You can recover information about how you have moved – called *ego-motion* – from flow fields. For example, the three flow fields of Figure 34.2 signal quite different things to a flying agent. Model the agent as a camera viewing a static scene. Assume that the camera is calibrated, starts in the standard configuration, and translates by $\mathbf{p}$. This means camera 2's focal point is at $\mathbf{p}$ in camera 1's frame. The point at $\mathbf{X}$ in camera 1's frame will be at $\mathbf{X} - \mathbf{p}$ in camera 2's frame (check you are sure about the minus sign). Check that the flow will be zero at the point $(p_1/p_3, p_2/p_3)$ in the image plane, whatever objects appear in the 3D world (**exercises**).

This point is often referred to as the *focus of expansion*. Flow close to the focus could be contracting or expanding. If the flow points away from the focus (expanding), the camera will eventually hit the corresponding point in the scene if it keeps moving as it has been moving. Similarly, if the flow points toward this point (contracting), the camera is escaping from that point in the scene.

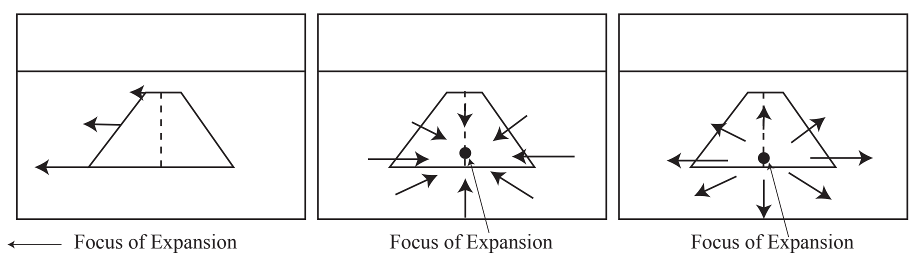

FIGURE 34.2: *Flow offers simple, clear signals about how one is moving with respect to the world. Look out of a bubble canopy on the nose of a small aircraft. If you see the flow on the* **left**, *you are heading from left to right. If you see the flow in the* **center**, *you are leaving the runway behind. If you see the flow on the* **right**, *you are heading towards the near end of the runway. Various reflexes related to flow patterns are found in a number of flying animals. For example, the flow on the right – if large enough compared to the size of the object in the image – will often trigger some collision management reflex.*

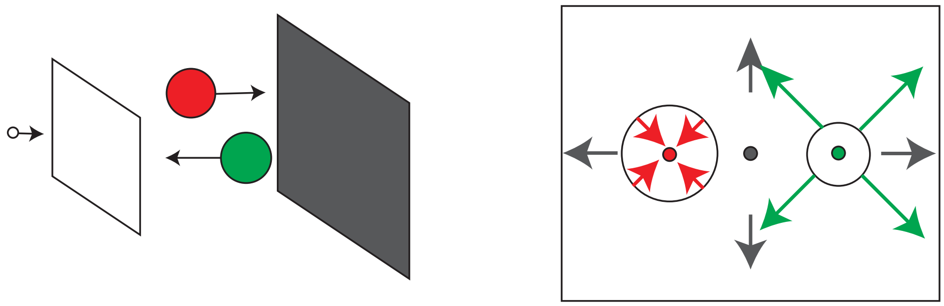

FIGURE 34.3: *Flow fields can have multiple focuses of expansion if there are multiple moving objects. On the* **left**, *a moving camera views two moving spheres and a moving plane (motion in the direction of the arrows). On the* **right**, *the resulting flow field. You can think of this as three flow fields (one for each sphere and one for the plane). Two camera geometry applies to each separate flow field, so each separate flow field has its own focus of expansion (or epipole). It is usually too difficult to segment all independently moving objects, so estimators tend not to make use of epipolar constraints directly.*

Definition: 34.2 *The focus of expansion*

The flow field in a camera that translates in a stationary scene always has at least one point where flow is zero. This is the focus of expansion. Any movement of the camera where the focal point translates will result in a focus of expansion. The focus of expansion, like the epipole to which it is related, may be out of the image plane.

You can recover 3D shape information from flow fields as well. Again, think about a camera viewing a static scene. As before, the camera is calibrated, starts in the standard configuration, and translates. For simplicity, have the camera translate along the X_1 direction, so that now $\mathbf{p} = (p_x, 0, 0)$. The point at $\mathbf{X}$ in camera 1's frame will be at $\mathbf{X} - \mathbf{p}$ in camera 2's frame (check you are sure about the minus sign). The flow at $(X_1/X_3, X_2/X_3)$ in the first image will be $(p_x/X_3, 0)$. If you know the flow, you can recover the depth up to scale. This example should suggest some form of relationship between optic flow and stereopsis.

Remember this: *If you know optic flow, you have strong information about geometry and camera motion.*

34.1.1 Comparing Stereo and Optic Flow

You take an image, then the camera translates by $\mathbf{p} = (p_x, 0, 0)$ and you take another. If nothing in the world has moved, there is no real difference between computing the optic flow and computing the disparity **exercises**. All the two-camera geometry of Chapter 31 applies to optic flow. The focus of expansion is equivalent to an epipole. As you would expect from this example, estimating optic flow has similarities with estimating disparity. The most important similarity is that photometric consistency is the main cue in each case.

Stereo estimation algorithms attempt to find a disparity field $\mathbf{v}$ that has a small value of an expression like

$$\mathcal{L}_{\text{photo}}(\mathcal{I}_{\text{left}}(\mathbf{x}+\mathbf{v}), \mathcal{I}_{\text{right}}(\mathbf{x}))$$

where $\mathcal{L}_{\text{photo}}$ is a photometric loss that compares its two arguments in some way. Here $\mathbf{v}$ is heavily constrained by geometry.

Flow estimation algorithms attempt to find a flow field $\mathbf{v}(\mathbf{x})$ that has a small value of

$$\mathcal{L}_{\text{photo}}(\mathcal{I}(\mathbf{x}+\mathbf{v}, t), \mathcal{I}(\mathbf{x}, t))$$

where $\mathbf{v}$ might be pretty much anything. Even the learned predictors of Chapter 36 do this; they just use exotic strategies to find the minimum.

Computing optical flow is a more general exercise than computing disparities, because objects could move in the world between images. These movements generate flows that are unlike disparities. If the world is static and a camera moves, there will be an epipole in each image and flows will always be along the epipolar lines. In this case, estimating flow is the same as estimating disparity.

If things move in the world, you can get more than one focus of expansion, and flow doesn't need to be along epipolar lines. Figure 34.3 shows a simple example. In these flow fields, there are three focuses of expansion. You could segment the images into three segments. Each of red, green and gray shows camera movement with respect to a static world, but the static world is different for red, green and blue. This view is helpful to understand what flow can be like, but tends not to yield much of use for estimating flow.

At a high level, there are similarities between algorithms for estimating optic flow and those for estimating disparity. In each case, photometric consistency is very powerful, though it manifests in slightly different ways. It turns out this means that learned procedures for each problem have very similar structures (Chapter 36). In detail, algorithms for estimating optic flow tend to differ from those for estimating disparity quite strongly. Epipolar structures give very strong constraints on stereo estimation, but aren't easy to use for flow estimation. You would need to know a segmentation of the world into independently moving scenes, and this is usually quite difficult to get. The constraints of Section 33.1.1 don't apply in any usable way. The family of possible flow fields is very much larger than the family of possible disparities. As a result, flow estimation algorithms use a richer range of procedures to control ambiguities. For example, it is much more usual to use parametric models of flow than to use parametric models of disparity.

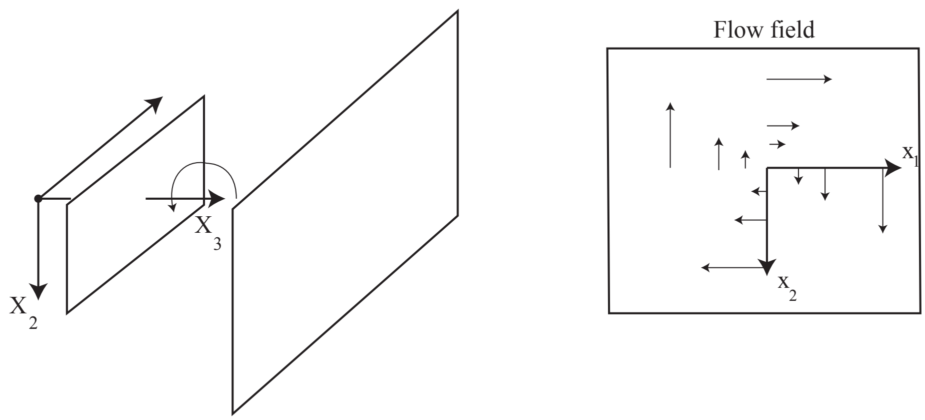

FIGURE 34.4: *A camera viewing a plane parallel to the image plane rotates about the X_3 axis, inducing a rotational flow in the image.*

34.2 PARAMETRIC OPTIC FLOW FIELDS

For many situations, you can compute a simple parametric expression for the flow field. These parametric expressions turn out to be a useful way to control the complexity of flow estimation (Chapter 35). Assume you can model the flow field $\mathbf{v}(\mathbf{x}) = \mathbf{v}(\mathbf{x}, \theta)$ where θ is a manageable number of parameters. Each of the examples below has this form. Then you could try to minimize

$$\mathcal{L}_{\text{photo}}(\mathcal{I}(\mathbf{x} + \mathbf{v}(\mathbf{x}, \theta), t), \mathcal{I}(\mathbf{x}, t))$$

as a function of θ. This approach has a long and useful history. In the rest of this section, I develop some examples of these parametric models.

34.2.1 Example: Rotating a Camera

A camera looks directly at a plane, then rotates about the X_3 axis. The plane is given by $X_3 = c$, so the point (x_1, x_2) in the image plane is at (cx_1, cx_2, c) in 3D. Under rotation about X_3, this point moves to $(cx_1 - \epsilon cx_2, cx_2 + \epsilon cx_1, c)$ (use the Taylor series for sine and cosine). In turn, the flow at (x_1, x_2) is

$$\begin{bmatrix} -\epsilon x_2 \\ \epsilon x_1 \end{bmatrix}.$$

This flow is shown in Figure 34.4. Notice the flow is linear in position, so linear in (x_1, x_2).

34.2.2 Example: Flow Looking out of a Train Window

A calibrated camera looks at a ground plane from a fixed height above it (Figure 34.5). The camera translates parallel to the ground plane and parallel to the image plane. The ground plane does not pass through the focal point, so its equation takes the form $bX_2 + cX_3 - 1 = 0$ in the camera coordinate system. The

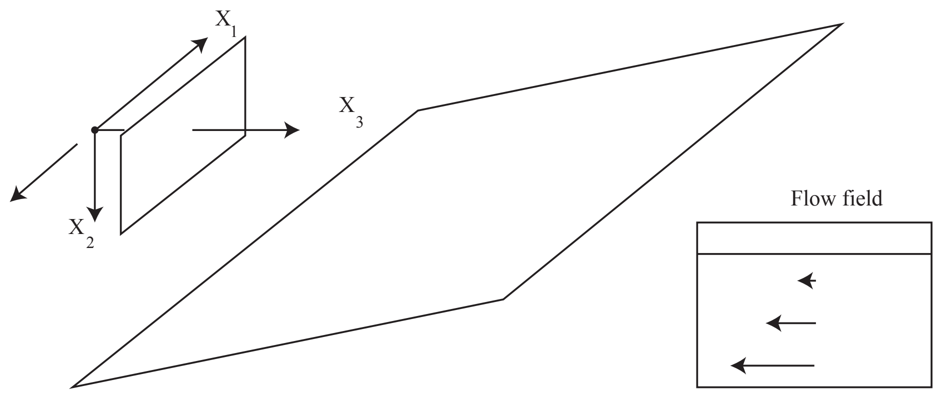

FIGURE 34.5: *A camera looks out of a train window at right angles to the direction of travel (the small arrow) and views a ground plane. As the text shows, the flow field looks like the inset – the flow at the horizon is zero, and flow grows linearly as you move down from the horizon.*

camera translates in the X_1 direction by t_x. The ray from the focal point through $\mathbf{x} = [x_1, x_2]$ in the image plane pierces the ground plane at $X_3 = 1/(bx_2 + c)$. This means the flow is

$$\begin{bmatrix} -t_x(bx_2 + c) \\ 0 \end{bmatrix}.$$

This flow is linear in image position. Notice the flow is zero when $(bx_2 + c) = 0$, which is also the horizon of the plane (check this **exercises**). If you know the flow, you can estimate two parameters, $t_x b$ and $t_x c$. You cannot recover both geometry and camera movement exactly from these two parameters. For example, if $c = 0$, you can interpret this ambiguity as implying a large movement by a camera high above the ground plane results in the same flow as a small movement by a camera closer to the plane.

34.2.3 Example: Tilting a Camera Toward the Ground

A calibrated camera sees a ground plane from a fixed height above it. In frame 1, the ground plane is at $X_2 = c$. This means the image point (x_1, x_2) corresponds to the 3D point $(\frac{x_1 c}{x_2}, c, \frac{c}{x_2})$. The camera then tilts slightly about the X_1-axis, so the 3D point at (X_1, X_2, X_3) moves to $(X_1, X_2 - \epsilon X_3, X_3 + \epsilon X_2)$ (use the Taylor series for sine and cosine). This means the image point at (x_1, x_2) will move to $(\frac{x_1 c}{x_2}, c - \epsilon \frac{c}{x_2}, \frac{c}{x_2} + \epsilon c)$ **exercises**. To first order, the flow is then

$$\begin{bmatrix} -\epsilon x_1 x_2 \\ -\epsilon(1 + x_2^2) \end{bmatrix}$$

exercises. The flow is shown in Figure 34.6. This flow is not linear in image position. Check that the horizon of the plane is at $x_2 = 0$, so the rather odd form

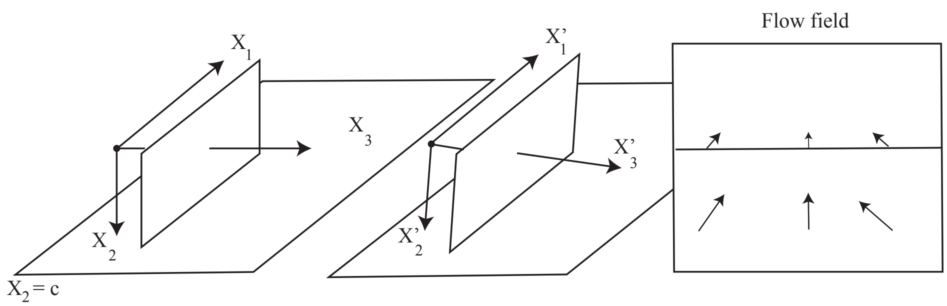

FIGURE 34.6: *A camera tilts slightly toward the ground plane, turning around the X_1 axis (**left**, **center**). The flow (**right**) reveals the tilt. Notice there is no flow above the horizon.*

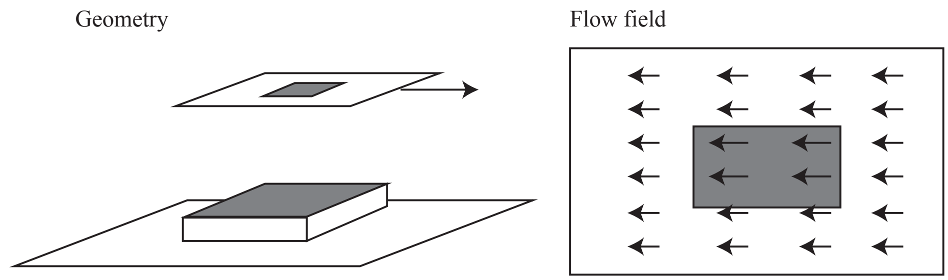

FIGURE 34.7: *A camera looks down at a ground plane with some small relief on it (**left**). The camera translates parallel to the image plane. The flow (**right**) reveals the relief. Points closer to the camera have slightly larger flow.*

of the flow when x_2 changes sign is easily explained: you see only the portion of the plane that is in front of the camera **exercises**.

34.2.4 Example: Flow Looking Down on Bumpy Ground

A calibrated camera looks directly down at a ground plane from a fixed height above it (Figure 34.7). The camera translates parallel to the ground plane, and the image plane is parallel to the ground plane. But the ground isn't really a plane. Houses, trees, people and such stick up out of the ground, and generate variations in the flow field. Write $X_3 = c + \delta X_3$ for some constant c, where δX_3 is small compared to c and represents this relief. Now set up a coordinate system so that, at time 1, the camera is in the standard coordinate system for a perspective camera. The camera translates parallel to the ground plane, so by $(t_x, t_y, 0)$. The time interval is small.

To first order, the flow is

$$\begin{bmatrix} \frac{-t_x}{c}(1 - \frac{\delta Z}{c}) \\ \frac{-t_y}{c}(1 - \frac{\delta Z}{c}) \end{bmatrix}.$$

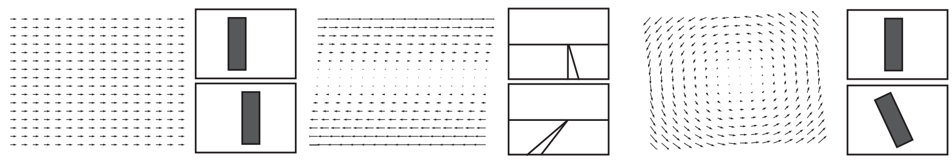

FIGURE 34.8: *A straightforward affine flow model in Section 34.2.5 encodes many useful cases. On the* **left**, *constant flow (the first column of $\mathcal{M}$ gives the flow direction, all others are zeros). In the* **center**, *the flow viewed from a train window as in Figure 34.5 (here the first three columns of $\mathcal{M}$ are non-zero, others are zeros). On the* **right**, *an object rotates (again, the first three columns of $\mathcal{M}$ are non-zero, others are zeros).* **Top** *image frame is frame 1,* **bottom** *is frame 2. Details in* **exercises**.

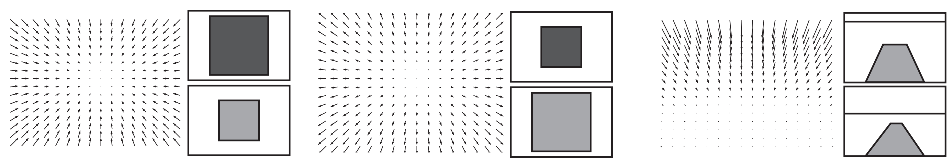

FIGURE 34.9: *A straightforward affine flow model in Section 34.2.5 encodes many useful cases. On the* **left**, *the viewer leaves an object; in the* **center**, *a viewer is heading towards an object; and on the* **right**, *the plane on which the rectangle sits is tilted backwards (or, equivalently, the viewer tilts backwards).* **Top** *image frame is frame 1,* **bottom** *is frame 2. Details of the relevant $\mathcal{M}$ for each case in* **exercises**.

(**exercises**). I have grouped the variables quite deliberately to show an ambiguity. You can't recover absolute relief without knowing either the height of the camera or the translation. If you just know the flow field, you can recover two parameters, $\frac{t_x}{c}$ and $\frac{t_y}{c}$ and a relative height field $\frac{\delta Z}{c}$ (**exercises**). The flow is not linear in image position, but it reveals relief. This situation is exactly analogous to stereo when you don't know the baseline **exercises**.

34.2.5 Affine Flows

Some of the parametric flow examples in the previous sections are linear in position in the image. For example, the rotational flow of Section 34.2.1 is $\epsilon(-x_2, x_1)^T$. Others are linear in functions of position in the image. For example, the flow of Section 34.2.3 is $-\epsilon(x_1 x_2, (1 + x_2^2))^T$. This suggests (correctly) that you can build a parametric model that is (a) linear in parameters and (b) covers many important flows.

To do so, choose a set of basis functions that depend on position in the image, $\phi_1(x, y), \ldots, \phi_n(x, y)$. An *affine flow model* is a flow model where flow at the pixel

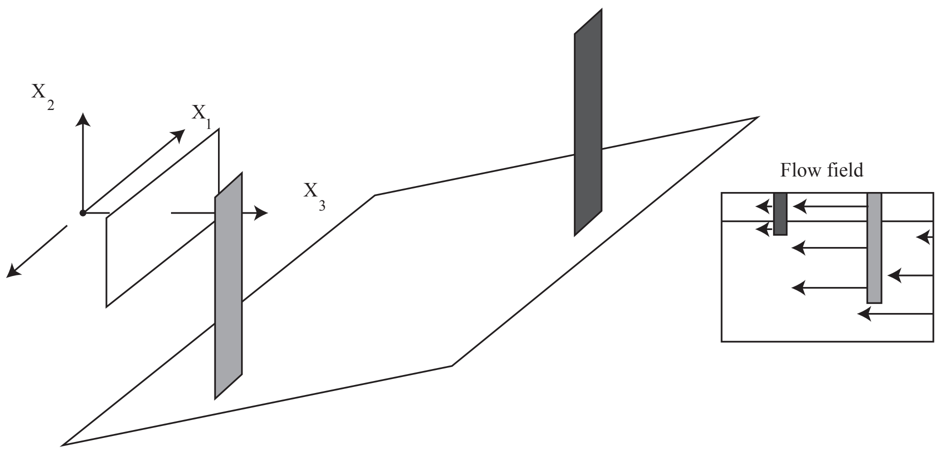

FIGURE 34.10: *A camera looks out of a train window at right angles to the direction of travel (the small arrow) and views a ground plane with two trees on it. The flow field on the ground plane looks like the inset of Figure 34.5, but the flow at the trees is different. The nearby tree has a large flow, and the distant tree has a small flow.*

at $(x, y)^T$ is given by

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \mathcal{M} \begin{bmatrix} \phi_1(x,y) \\ \dots \\ \phi_n(x,y) \end{bmatrix}.$$

By far the most usual choice is given in the box, below.

Remember this: *The flow components at x, y are $u(x, y)$ and $v(x, y)$. Model these components by*

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \mathcal{M} \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

where $\mathcal{M}$ is a 2×6 matrix of parameters that determine the particular flow field

The model in the box covers a number of useful cases (Figures 34.8 and 34.9; **exercises**).

34.2.6 Layered Motion Models

A calibrated camera translates in the X_1 direction at a fixed height above a ground plane. The ground plane is at $X_2 = c$ in camera coordinates, so the image plane is perpendicular to the ground plane. A collection of N flat objects stick up out of the ground plane at fixed depths (Figure 34.10; the i'th object is at $X_3 = d_i$). For this setup, the flow in the x_2 direction in the image plane is always zero. Check that, if the pixel at $(x_1, x_2)^T$ sees the ground plane, the x_3 coordinate is c/x_2. The x_1 component of flow at a pixel takes one of $N+1$ values

$$\begin{cases} -\frac{t_x c}{x_2} & \text{if the pixel sees the ground plane} \\ -\frac{t_x}{c_1} & \text{if the pixel sees object 1} \\ \dots & \dots \\ -\frac{t_x}{c_N} & \text{if the pixel sees object N} \end{cases}$$

You can think of the image as consisting of *layers*: one is the ground plane, and the others are the trees. This class of model is often referred to as a *layered motion* model (after []). Notice that in this example each layer is actually an affine flow model. This is a common choice, expressed in the box.

Remember this: *A layered affine motion model is built by: choosing a number K of layers; choosing K affine flow fields (given by $\mathcal{M}_1, \dots, \mathcal{M}_K$); then allocating each pixel to a layer. This determines the flow at each pixel.*

34.3 YOU SHOULD

34.3.1 remember these definitions:

34.3.2 remember these facts:

34.3.3 be able to:

- Explain what optic flow is.
- Explain what information optic flow conveys and why.
- Compare estimating flow and estimating disparity.
- Derive parametric flow models in simple cases.

EXERCISES

QUICK CHECKS

34.1. Assume a camera translates while viewing a static scene. Assume that the camera is calibrated, starts in the standard configuration, and translates by $\mathbf{p}$. Work in camera coordinates. Show that flow will be zero at the point $(p_1/p_3, p_2/p_3)$ in the image plane, whatever objects appear in the 3D world.

34.2. Check that the focus of expansion for flow observed in a translating camera in a static scene is the epipole.

34.3. Check that any movement of the camera where the focal point translates will result in a focus of expansion.

34.4. Section 34.2.2 has: "This flow is linear in image position. " Explain.

34.5. Section 34.2.2 has: "Notice the flow is zero when $(bx_2 + c) = 0$, which is also the horizon of the plane." Check this.

34.6. What is $\mathcal{M}$ for a constant flow? (notation in Section 34.2.5)

34.7. What is $\mathcal{M}$ for the flow of Section 34.2.2? (notation in Section 34.2.5)

34.8. Can you model the flow looking down on bumpy ground (Section 34.2.4) with an affine flow model?

34.9. What is $\mathcal{M}$ for the flow of Section **??**?(notation in Section 34.2.5)

34.10. Can you model the flow of Section 34.2.6) with a single affine flow model?

LONGER PROBLEMS

34.11. At time 0, a standard camera views the line segment from $(0, 0, d)$ to $(0, -h, d)$. The camera now moves with velocity $(0, 0, v_z)$.

(a) Show the length of the line segment in the frame at time t is $l(t) = (h/(d - v_z t))$.

(b) Now show $\frac{l}{\frac{dl}{dt}}$ is an estimate of the time to contact between the camera and the line segment.

(c) Assume the camera has focal length f, which is unknown. Show $\frac{l}{\frac{dl}{dt}}$ does not depend on f.

(d) What happens if the other camera parameters are unknown?

34.12. At time 0, a standard camera views the disk $X_1^2 + X_2^2 \leq R^2$ with center at $(0, 0, d)$ on the plane $X_3 = d$. The camera now moves with velocity $(0, 0, v_z)$.

(a) Show the area of the disk in the frame at time t is $A(t) = 2\pi(R/(d - v_z t)^2$.

(b) Now show $\frac{A}{\frac{dA}{dt}}$ is an estimate of the time to contact between the camera and the disk.

(c) Assume the camera has focal length f, which is unknown. Show $\frac{A}{\frac{dA}{dt}}$ does not depend on f.

(d) What happens if the other camera parameters are unknown?

34.13. Section 34.2.4 describes a calibrated camera looking directly down at a ground plane from a fixed height above it. The ground plane is at $X_3 = c$, and has small relief δX_3 The camera translates parallel to the ground plane, so by $(t_x, t_y, 0)$. The time interval is small.

(a) Show that, to first order, the flow is

$$\begin{bmatrix} \frac{-t_x}{c}(1 - \frac{\delta X_3}{c}) \\ \frac{-t_y}{c}(1 - \frac{\delta X_3}{c}) \end{bmatrix}.$$

(It may help to remember that

$$\frac{1}{c+\delta X_3} \approx \frac{1}{c}(1 - \frac{\delta X_3}{c}).$$

(b) Show that, if you know the flow, you can estimate two parameters, $\frac{t_x}{c}$ and $\frac{t_y}{c}$ and a relative height field $\frac{\delta Z}{c}$. Explain the ambiguity implied by the fact you can recover three values for four parameters.

34.14. Section **??** describes a calibrated camera looking at a ground plane from a fixed height above it, then tilting slightly about the X_1-axis. In frame 1, the ground plane is at $X_2 = c$.

(a) Show that an image point at (x_1, x_2) is the projection of $(\frac{x_1 c}{x_2}, c, \frac{c}{x_2})$

(b) Now assume the camera is fixed and the ground tilts. Show that in frame 2 this point is now at $(\frac{x_1 c}{x_2}, c - \epsilon\frac{c}{x_2}, \frac{c}{x_2} + \epsilon c)$ (It may help to remember that $\sin\epsilon \approx \epsilon + O(\epsilon^3)$ and $\cos\epsilon \approx 1 + O(\epsilon^2)$).

(c) Show that, to first order, the flow is

$$\begin{bmatrix} -\epsilon x_1 x_2 \\ -\epsilon(1 - x_2^2) \end{bmatrix}$$