In "Exploring Artificial Intelligence in the New Millennium", G. Lakemeyer and B. Nebel (editors), Morgan Kaufmann, San Fancisco, pp. 139-173.

Bayesian Inference of Visual Motion Boundaries

David J. Fleet
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
fleet@parc.com

Michael J. Black
Department of Computer Science
Brown University
Providence, RI
black@cs.brown.edu

Oscar Nestares
Instituto de Óptica (C.S.I.C.),
Serrano 121, 28006-Madrid, Spain
nestares@io.cfmac.csic.es

Abstract

This chapter addresses an open problem in visual motion analysis, the estimation of image motion in the vicinity of occlusion boundaries. With a Bayesian formulation, local image motion is explained in terms of multiple, competing, nonlinear models, including models for smooth (translational) motion and for motion boundaries. The generative model for motion boundaries explicitly encodes the orientation of the boundary, the velocities on either side, the motion of the occluding edge over time, and the appearance/disappearance of pixels at the boundary. We formulate the posterior probability distribution over the models and model parameters, conditioned on the image sequence. Approximate inference is achieved with a combination of tools: A Bayesian filter provides for online computation; factored sampling allows us to represent multimodal non-Gaussian distributions and to propagate beliefs with nonlinear dynamics from one time to the next; and mixture models are used to simplify the computation of joint prediction distributions in the Bayesian filter. To efficiently represent such a high-dimensional space we also initialize samples using the responses of a low-level motion discontinuity detector. The basic formulation and computational model provide a general probabilistic framework for motion estimation with multiple, non-linear, models.

1 Visual Motion Analysis

Motion is an intrinsic property of the world and an integral part of our visual experience. It provides a remarkably rich source of information that supports a wide variety of visual tasks. Examples include 3D model acquisition, event detection, object recognition, temporal prediction, and oculomotor control.

Visual motion has long been recognized as a key source of information for inferring the 3D structure of surfaces and the relative 3D motion between the observer and the scene (Gibson 1950; Ullman 1979; Longuet-Higgins and Prazdny 1980). In particular, the 2D patterns of image velocity that are produced by an observer moving through the world can be used to infer the observer's 3D movement (i.e., egomotion). Many animals are known to use visual motion to help control locomotion and their interaction with objects (Gibson 1950; Warren 1995; Sun and Frost 1998; Srinivasan, Zhang, Altwein, and Tautz 2000).

It is also well-known that visual motion provides information about the 3D structure of the observed scene, including the depth and orientation of surfaces. Given the 3D motion of the observer and the 2D image velocity at each pixel, one can infer a 3D depth map. In particular, it is straightforward to show that the 2D velocities caused by the translational component of an observer's 3D motion are inversely proportional to surface depth (Heeger and Jepson 1992; Longuet-Higgins and Prazdny 1980).

While much of the research on visual motion has focused on the estimation of 2D velocity and the inference of egomotion and 3D depth, it is also widely recognized that visual motion conveys information about object identity and behavior. Many objects exhibit characteristic patterns of motion. Examples include rigid and articulated motions, different types of biological motion, or trees blowing in the wind. From these different classes of motion we effortlessly detect and recognize objects, assess environmental conditions (e.g., wind, rain, and snow), and begin to infer and predict the object behavior. This facilitates a broad range of tasks such as the detection and avoidance of collisions, chasing (or fleeing) other animate objects, and the inference of the activities and intentions of other creatures.

Given the significance of visual motion, it is not surprising that it has become one of the most active areas of computer vision research. In just over two decades the major foci of research on visual motion analysis include:

- *Optical Flow Estimation:* This refers to the estimation of 2D image velocities from image sequences (Horn 1986; Barron, Fleet, and Beauchemin 1994; Otte and Nagel 1994). Although originally viewed as a precursor to the estimation of 3D scene properties, the techniques developed to estimate optical flow have also proven useful for other *registration* problems; examples are found in medical domains, in video compression, in forming image mosaics (panoramas), and in stop-frame animation.
- *Motion-Based Segmentation:* Although optical flow fields are clearly useful, they do not explicitly identify the coherently moving regions in an image. Nor do they separate foreground and background regions. For these tasks, layered motion models and the Expectation-Maximization (EM) algorithm have become popular (Jepson and Black 1993; Sawhney and Ayer 1996; Vasconcelos and Lippman 2001; Weiss and Adelson 1996), as have many other approaches, such as automated clustering based on spatial proximity and motion similarity.

- *Egomotion and Structure-from-Motion:* The estimation of self-motion and 3D depth from optical flow or tracked points over many frames, has been one of the long-standing fundamental problems in visual motion analysis (Broida, Chandrashekhar, and Chellappa 1990; Heeger and Jepson 1992; Tomasi and Kanade 1992; Longuet-Higgins and Prazdny 1980). While typically limited to nearly stationary (rigid) environments, current methods for estimating egomotion can produce accurate results at close to video frame rates (e.g., see (Chiuso, Favarto, Jin, and Saotto 2000)).
- *Visual Tracking:* Improvements in flow estimation have enabled visual tracking of objects reliably over tens and often hundreds of frames. Often these methods are strongly model-based, requiring manual initialization, and prior specification of image appearance and model dynamics (Irani, Rousso, and Peleg 1994; Shi and Tomasi 1994; Sidenbladh, Black, and Fleet 2000). One of the lessons learned from research on visual tracking is the importance of having suitable models of image appearance and temporal dynamics, whether learned prior to tracking (Hager and Belhumeur 1998; Black and Jepson 1998), or adaptively during tracking (Jepson, Fleet, and El-Maraghi 2001).

In this chapter we focus on the problem of estimating 2D image velocity, especially in the neighborhoods of surface boundaries.

1.1 Optical Flow

The estimation of optical flow was first studied in detail over 20 years ago (Fennema and Thompson 1979; Horn and Schunk 1981). Since then, techniques for optical flow estimation have improved significantly. The use of benchmark data sets and publically available code have helped to estabalish the quantitative accuracy of recent methods (Barron, Fleet, and Beauchemin 1994). Accordingly, it is now relatively well accepted that, for smooth textured surfaces, current methods provide accurate and relatively fast estimators for 2D image velocity.

Although many interesting variations exist, perhaps the simplest, most commonly used techniques are known as area-based regression methods. Broadly speaking, these techniques are derived from two main assumptions, namely, *brightness constancy*, and *smoothness*. The brightness constancy assumption states that the light reflected from a surface toward the camera remains invariant through time. If we further assume that visible points at time $t-1$ are also visible at time t, then we can then express the image at time t as a deformation of the image at time $t-1$:

$$I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{u}(\mathbf{x}), t-1) \ . \tag{1}$$

With (1), one can estimate the 2D optical flow, $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))^T$, at different spatial positions, $\mathbf{x} = ((x, y)^T$, by tracking points of constant brightness.

The second common assumption that underpins current methods is that the optical flow field is a smooth function of image position. This is often formulated by constraining the optical flow field $\mathbf{u}(\mathbf{x})$ to lie in a subspace spanned by a basis of smooth flow fields for a local neighborhood of image positions:

$$\mathbf{u}(\mathbf{x};\, \mathbf{a}) \ = \ \sum_{j=1}^{n} a_j \, \mathbf{b}_j(\mathbf{x}) \tag{2}$$

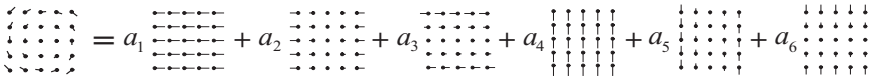

Figure 1 Affine flow fields can be expressed as a linear combination of the elements of a six-dimensional basis set of flow fields, shown here for 5×5 local neighborhoods).

where $\{\mathbf{b}_j(\mathbf{x})\}_{j=1\ldots n}$ is the basis, and $\mathbf{a} = (a_1, ..., a_n)$ denotes the linear coefficients (Bergen, Anandan, Hanna, and Hingorani 1992; Fleet, Black, Yacoob, and Jepson 2000). For example, Fig. 1 shows an affine basis that accounts for translation, scaling, rotation and shear. Estimating the optical flow then amounts to estimating the coefficients $\mathbf{a}$ that produce the flow field that minimizes violations of brightness constancy (1) in the subspace spanned by $\{\mathbf{b}_j(\mathbf{x})\}_{j=1\ldots n}$. Alternatively, the smoothness constraint can be formulated as a regularization term that specifies how the motion at neighboring pixels may vary (Horn and Schunk 1981).

Several properties contribute to the effectiveness of such methods. First, the number of unknowns $\mathbf{a}$ is typically small compared to the number of pixels in the spatial neighborhood each of which provides a brightness constancy constraint. Second, the solution can be found with straightforward numerical methods. If we linearize (1) and discard all but first-order terms, then we obtain a gradient constraint:

$$\nabla I(\mathbf{x}, t-1) \cdot \mathbf{u}(\mathbf{x};\, \mathbf{a}) - \Delta I = 0 \tag{3}$$

where $\nabla I = (I_x, I_y)$ is the spatial image gradient, and $\Delta I = I(\mathbf{x}, t) - I(\mathbf{x}, t-1)$. Because (3) is linear in $\mathbf{u}(\mathbf{x};\mathbf{a})$, and $\mathbf{u}(\mathbf{x};\mathbf{a})$ is linear in $\mathbf{a}$, the collection of these constraints at each pixel in the spatial region yields a linear, least-squares system of equations.

1.2 Motion Boundaries

While current optical flow techniques produce reliable estimates for smooth textured surfaces, there are classes of motion for which they are not effective. There are many situations where brightness constancy does not hold and the motion is not smooth. Examples include motion discontinuities, the motion of bushes or trees in the wind, and the deformations and self-occlusions of clothing as people walk.

Optical flow at surfaces boundaries is often discontinuous because surfaces at different depths usually produce different image velocities. This violates the smoothness assumption. Furthermore, pixels that are visible at one time may not be visible at the next time as the foreground moves and thereby occludes a different portion of the background; this violates the brightness constancy assumption. As a consequence, most optical flow techniques produce poor estimates at occlusion boundaries.

Nevertheless, motion boundaries remain a rich source of scene information. First, they provide information about the position and orientation of surface boundaries. Second, analysis of the occlusion or disocclusion of pixels at motion boundaries can provide information about the relative depth ordering of the adjacent surfaces. In turn, information about surface boundaries and depth ordering may be useful for tasks as diverse as navigation, structure from motion, video compression, perceptual organization, and object recognition.

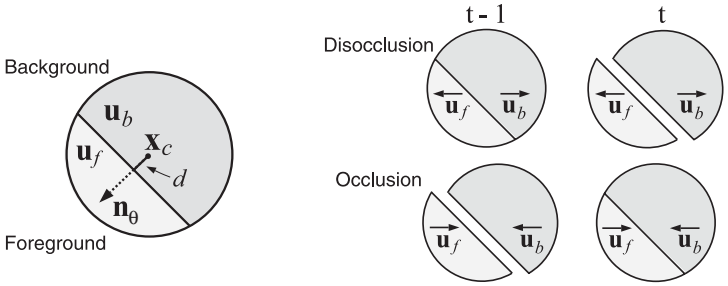

Figure 2 Our motion boundary model is parameterized by foreground and background velocities, $\mathbf{u}_f$ and $\mathbf{u}_b$, an orientation θ with normal $\mathbf{n}_\theta$, and a signed distance d from the neighborhood center $\mathbf{x}_c$. With this model we can predict which pixels are visible between frames at times $t-1$ and t.

In this chapter we formulate a probabilistic, model-based, approach to image motion analysis. The 2D motion in each local neighborhood of an image is estimated and represented using one of several possible models. This approach allows us to use different motion models that are suited to the diverse types of optical flow that occur with natural scenes. In this chapter we consider two models, namely, smooth motion and motion boundaries. Regions of smooth motion may be modeled using conventional translational or affine models while the complex phenomena that occur at motion boundaries are accounted for by an explicit, non-linear, boundary model.

The motion boundary model illustrated in Fig. 2 encodes the boundary orientation, the image velocities of the pixels on each side of the boundary, the foreground/background assignment for the two sides, and the distance from the boundary to the region center. With this model, we can predict the visibility of occluded and disoccluded pixels so that these pixels may be excluded when estimating the probability of a particular motion. Moreover, the explicit offset parameter allows us to predict the location of the edge within the region of interest, and hence track its movement through the region. Tracking the motion of the edge allows foreground/background ambiguities to be resolved.

Generative models like this have not previously been used for detecting motion discontinuities due to the non-linearity of the model and the consequent difficulty of estimating the model parameters. Furthermore, while the use of multiple models clearly complicates the resulting estimation problem, it provides us with a rich framework with which we can address the problem of model selection, to determine what type of motion model is most suited to the nature of the input, and to thereby estimate properties of the motion that provide useful information about the underlying 3D scene structure.

1.3 Previous Work on Motion Boundaries

The detection of motion boundaries has been a long-standing problem in optical flow estimation, primarily because most approaches to computing optical flow fail to be reliable in the vicinity of motion discontinuities (Barron, Fleet, and Beauchemin 1994; Fleet 1992; Otte and Nagel 1994). In addition, it has long been acknowledged that motion boundaries provide useful information about the position and orientation of surface boundaries.

Most previous methods cope with motion boundaries by treating them as a form of *noise*; that is, as the violation of a smoothness assumption. This occurs with regularization schemes where robust statistics, weak continuity, or line processes are used to locally disable smoothing across motion discontinuities (Cornelius and Kanade 1981; Harris, Koch, Staats, and Luo 1990; Heitz and Bouthemy 1993; Konrad and Dubois 1998; Murray and Buxton 1987; Nagel and Enkelmann 1986; Schunck 1989; Shulman and Hervé 1989; Thompson, Mutch, and Berzins 1985). Robust regression (Black and Anandan 1996; Sawhney and Ayer 1996) and mixture models (Sawhney and Ayer 1996; Jepson and Black 1993; Weiss and Adelson 1996) have been used to account for the multiple motions that occur at motion boundaries but these methods fail to explicitly model the boundary and its spatiotemporal structure. They do not determine the boundary orientation, which pixels are occluded or disoccluded, or the depth ordering of the surfaces at the boundary.

Numerous methods have attempted to detect discontinuities in optical flow fields by analyzing local distributions of flow (Spoerri and Ullman 1987) or by performing edge detection on the flow field (Potter 1980; Schunck 1989; Thompson, Mutch, and Berzins 1985). It has often been noted that these methods are sensitive to the accuracy of the optical flow and that accurate optical flow is hard to estimate without prior knowledge of the occlusion boundaries. Other methods have focused on detecting occlusion from the structure of a correlation surface (Black and Anandan 1990), or of the spatiotemporal brightness pattern (Beauchemin and Barron 2000; Chou 1995; Fleet and Langley 1994; Niyogi 1995). Still others have used the presence of unmatched features to detect dynamic occlusions (Mutch and Thompson 1985; Thompson, Mutch, and Berzins 1985).

None of these methods explicitly capture the spatial structure of the image motion present in the immediate neighborhood of the boundary, and they have not proved sufficiently reliable in practice. One recent approach formulated an approximate model for motion boundaries using linear combinations of basis flow fields (Fleet, Black, Yacoob, and Jepson 2000). Estimating the image motion in this case reduces to a regression problem based on brightness constancy and parameterized models as in (2) and (3) above. Moreover, from the estimated linear coefficients, one can compute the orientation of the boundary and the velocities on either side, as shown Fig. 5. While useful, the estimates of the motion and the boundary location produced by this approach are quite noisy. Moreover, they do not identify the foreground side, nor do they identify which image pixels are occluded or disoccluded between frames. Rather, pixels that are not visible in both frames are treated as noise. With the non-linear model developed below, these pixels can be predicted and therefore taken into account in the likelihood computation (cf. (Belhumeur 1996).

Additionally, most of the above methods have no explicit temporal model. With our generative model (described below), we predict the motion of the occlusion boundary over time and hence integrate information over multiple frames. When the motion of the discontinuity is consistent with that of the foreground surface we can explicitly determine the foreground/background relationships (local depth ordering) between the surfaces.

1.4 Bayesian Filtering and Approximate Inference

To cope with image noise, matching ambiguities, and model uncertainty, we adopt a Bayesian probabilistic framework that integrates information over time and represents multiple, competing, model hypotheses. Our goal is to compute the posterior probability distribution over motion models and their parameters, conditioned on image measure-

ments. The posterior is expressed in terms of a likelihood function and a prior (prediction) probability distribution. The likelihood is the probability of observing the current image given the correct model. The prior represents our belief about the motion at the current time based on previous observations. This *temporal* prior embodies our assumptions about the temporal dynamics of how the models and model parameters evolve over time.

Because we use multiple motion models, we need to cope with both discrete and continuous variables (i.e., a hybrid state space). The discrete state variable encodes the type of motion, smooth or discontinuous, and the continuous variables encode the corresponding motion parameters (we use 2 parameters for smooth motion, and 6 for the motion boundary model). As is well known, posterior distributions over hybrid state spaces, where continuous variables depend on discrete variables, are usually multi-modal. To add further complexity, because our likelihood function and temporal dynamics are both nonlinear, we also expect the modes of the distribution to be non-Gaussian.

One form of approximate inference that has recently become very popular for such dynamical vision problems (e.g., motion and tracking), is the particle filter (Doucet, de Freitas, and Gordon 2001; Gordon, Salmond, and Smith 1993; Isard and Blake 1998a; Kitagawa 1987; Liu and Chen 1998; West 1992). Also known as Condensation and Sequential Monte Carlo filtering, the idea is to approximate the posterior with a weighted set of samples; particles (samples) are drawn randomly from a proposal distribution, often called a temporal prior, and then weighted by normalized likelihood values. With such *point-mass approximations* to probability distributions, particle filters are effective for nonlinear systems that produce non-Gaussian, multimodal distributions.

Additionally, we also want to estimate the motion in local image regions throughout the entire image. If we could assume that the motion in each region were independent of its neighbors, then we could estimate the motion in each separately. However, motion in one region is often a good predictor for motion of its neighbors, both at the current time and at successive times. Accordingly, the problem becomes Bayesian inference over a random field in which there is a relatively high dimensional hybrid state space at each location in the field. The result, which is well known, is that computation of the exact posterior (i.e., Bayesian inference) is not tractable, requiring time-consuming approximate solutions. This is typical of many vision problems, and it leads us to search for methods that produce satisfactory approximations to the true posterior.

While the particle filter discussed above is suitable for hybrid state spaces with nonlinear dynamics and likelihood functions, it will not cope with a random field. As is well known, one problem with particle filters is the exponential increase in the required number of particles (i.e., computational cost) as a function of the dimensionality of the state space. Although effective for low-dimensional tracking, they do not scale well to high-dimensional problems (e.g., see (Choo and Fleet 2001; Deutscher, Blake, and Reid 2000; MacCormick and Isard 2000; Sminchisescu and Triggs 2001)). This is particularly significant with random fields where the state dimension grows linearly with the number of random field locations, and conventional iterative solutions (such as MCMC (Gilks, Richardson, and Spiegelhalter 1996)) can be prohibitively slow. The situation worsens when we consider an entire image sequence.

In this chapter we explore two forms of approximate inference, drawing on research described in (Black and Fleet 2000; Nestares and Fleet 2001). In the first case we

approximate the posterior and the temporal dynamics by factoring each so that each local region of the image can be treated separately. This has problems since it prohibits us from encouraging boundary continuity and from allowing one region to predict when edges are going to move from one region to another.

The second model we describe uses combines Bayesian filtering with spatiotemporal predictions to detect and track motion boundaries. We continue to assume that the posterior can be approximated as the product of its marginal distributions for each region, but we introduce a dynamical model that explicitly represents interactions between neighboring regions. To do so we make use of several inference tools: in addition to approximating the joint posterior over multiple regions by its marginals (Murphy and Weiss 2001), we use Monte Carlo (sampled) approximations to these distributions to deal with non-linear dynamics and non-Gaussian likelihoods, and we use mixture models to efficiently approximate the prediction distributions that arise from multiple neighborhoods.

While the method described here can be thought of simply as a motion boundary detector, the framework has wider application. The Bayesian formulation and computational model provide a general probabilistic framework for motion estimation with multiple, non-linear, models. This generalizes previous work on recovering optical flow using linear models (Bergen, Anandan, Hanna, and Hingorani 1992; Fleet, Black, Yacoob, and Jepson 2000). Moreover, the Bayesian formulation provides a principled way of choosing between multiple hypothesized models for explaining the image variation within a region. This work can also be viewed as an exploration of the suitability of different forms of approximate inference in the context of otherwise intractable inference problems in vision.

2 Generative Motion Models

Our Bayesian formulation rests on the specification of *generative* models for smooth motion and motion boundaries. These generative models define our probabilistic assumptions about the spatial structure of the motion within a region, how the parameters are expected to vary through time, and the probability distribution over the image measurements that one would expect to observe given the model.

Accordingly, we first describe our generative model for the motion in a single local image neighborhood. As suggested in Fig. 2, we decompose an image into a grid of circular neighborhoods in which we estimate motion information. We assume that the motion in any region can be modeled by one of several motion models; here we consider only two models, namely smooth (translational) motion and motion boundaries.

For the smooth motion model, we express the optical flow within the circular region as image translation; more complex models can also be used. The translational model has two parameters, namely, the horizontal and vertical components of the velocity, denoted $\mathbf{u}_0 = (u_0, v_0)$. Exploiting the common assumption of brightness constancy, the generative model states that the image intensity, $I(\mathbf{x}', t)$, of a point $\mathbf{x}' = (x', y')$ at time t in a region R is equal to the intensity at some location $\mathbf{x}$ at time $t-1$ with the addition of noise ν_n:

$$I(\mathbf{x}', t) = I(\mathbf{x}, t-1) + \nu_n(\mathbf{x}, t) \ , \tag{4}$$

where $\mathbf{x}' = \mathbf{x} + \mathbf{u}_0$. Here, we are assuming that the noise, $\nu_n(\mathbf{x}, t)$, is white and mean-zero Gaussian with a standard deviation of σ_n; that is, $\nu_n \sim \mathcal{N}(0, \sigma_n^2)$.

The motion boundary model is more complex and contains 6 parameters: the edge orientation, the velocities of the foreground ($\mathbf{u}_f$) and the background ($\mathbf{u}_b$), and the distance from edge to the center of the region $\mathbf{x}_c$. In our parameterization, shown in Fig. 2, the orientation, $\theta \in [-\pi, \pi)$, specifies the direction of a unit vector, $\mathbf{n} = (\cos(\theta), \sin(\theta))$, that is normal to the occluding edge. We represent the location of the edge by its signed perpendicular distance d from the center of the region (positive meaning in the direction of the normal). The edge is normal to $\mathbf{n}$ and passes through the point $\mathbf{x}_c + d\,\mathbf{n}$. Relative to the center of the region, we adopt a convention where the foreground side is that side to which the normal $\mathbf{n}$ points. Thus, a point $\mathbf{x}$ is on the foreground if $(\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{n} > d$. Points on the background satisfy $(\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{n} < d$.

At most motion boundaries some pixels will be occluded or disoccluded. As a consequence, at boundaries one should not expect to find corresponding pixels in adjacent frames. Here, in order to resolve occlusions and disocclusions, we assume that the motion boundary edge moves with the same velocity as the pixels on the foreground side of the edge (i.e., the occluding side).[1] With this assumption, the occurrence of occlusion or disocclusion depends solely on the difference between the background and foreground velocities. Pixels become occluded from one frame to the next when the background moves faster than the foreground in the direction of the edge normal. More precisely, if $u_{fn} = \mathbf{u}_f \cdot \mathbf{n}$ and $u_{bn} = \mathbf{u}_b \cdot \mathbf{n}$ denote the two normal velocities, occlusion occurs when $u_{bn} - u_{fn} > 0$. Disocclusion occurs when $u_{bn} - u_{fn} < 0$. The width of the occluded/disoccluded region, measured normal to the occluding edge, is $|u_{bn} - u_{fn}|$.

With this model, parameterized by (θ, $\mathbf{u}_f$, $\mathbf{u}_b$, d), we can now specify how visible points move from one frame to the next. A pixel $\mathbf{x}$ at time $t-1$, that remains visible at time t, moves to location $\mathbf{x}'$ at time t given by

$$\mathbf{x}' = \begin{cases} \mathbf{x} + \mathbf{u}_f & \text{if } (\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{n} > d \\ \mathbf{x} + \mathbf{u}_b & \text{if } (\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{n} < d + w \end{cases} \tag{5}$$

where $w = \max(u_{bn} - u_{fn}, 0)$ is the width of the occluded region. Finally, with $\mathbf{x}'$ defined by (5), along with the assumptions of brightness constancy and white Gaussian image noise, the image observations associated with a motion edge are also given by (4).

Referring to Fig. 2(right), in the case of disocclusion, a circular neighborhood at time $t-1$ maps to a pair of regions at time t, separated by the width of the disocclusion region $|u_{bn} - u_{fn}|$. Conversely, in the case of occlusion, a pair of neighborhoods at time $t-1$, separated by $|u_{bn} - u_{fn}|$, map to a circular neighborhood at time t. Being able to look forward or backward in time in this way allows us to treat occlusion and disocclusion symmetrically.

So far we have focused on the spatial structure of the generative models. We must also specify the evolution of the model parameters through time since this will be necessary to disambiguate which side of the motion boundary is the foreground. From optical flow alone one cannot determine the motion of the occlusion boundary using only two frames. The boundary must be observed in at least two separate instances (e.g., using three

[1] Physical situations that violate this assumption include rotating objects, such as a baseball where the edge of the ball moves in one direction, but, due to the spin on the ball, the surface texture of the ball moves in another direction. Nevertheless, assuming that the edge moves with the foreground velocity, as we do in this paper, allows one to handle most cases of interest.

consecutive frames) to discern its motion. The image pixels whose motion is consistent with that of the boundary are likely to belong to the occluding surface. Thus, to resolve the foreground/background ambiguity, we propose to accumulate evidence over time using Bayesian tracking.

Towards this end, we assume that the motion parameters of both motion models obey a first-order Markov process; i.e., given the parameter values at the time $t-1$, the values at time t are conditionally independent of the values before time $t-1$. For the smooth motion model we assume that, on average, the image translation remains constant from one time to the next. More precisely, we assume that the image translation at time t, $\mathbf{u}_{0,t}$, is given by

$$\mathbf{u}_{0,t} \;=\; \mathbf{u}_{0,t-1} + \nu_u \;, \qquad \nu_u \sim \mathcal{N}(0, \sigma_u^2\, \mathbf{I}_2) \;, \tag{6}$$

where $\mathbf{I}_2$ is the 2D identity matrix. Here, ν_u represents the modeling uncertainty (process noise) implicit in this simple first-order dynamical model.

For the motion boundary model, we assume that, on average, the velocities on either side of the boundary and the boundary orientation remain constant from one time to the next. Moreover, as discussed above, we assume that the expected location of the boundary translates with the foreground velocity. More formally, these assumed dynamics are then given by

$$\begin{aligned}
\mathbf{u}_{f,t} &= \mathbf{u}_{f,t-1} + \nu_{u,f} \;, \qquad \nu_{u,f} \sim \mathcal{N}(0, \sigma_u^2\, \mathbf{I}_2) && (7)\\
\mathbf{u}_{b,t} &= \mathbf{u}_{b,t-1} + \nu_{u,b} \;, \qquad \nu_{u,b} \sim \mathcal{N}(0, \sigma_u^2\, \mathbf{I}_2) && (8)\\
\theta_t &= [\theta_{t-1} + \nu_\theta] \bmod 2\pi \;, \qquad \nu_\theta \sim \mathcal{N}(0, \sigma_\theta^2) && (9)\\
d_t &= d_{t-1} + \mathbf{n}_{t-1} \cdot \mathbf{u}_{f,t-1} + \nu_d \;, \qquad \nu_d \sim \mathcal{N}(0, \sigma_d^2). && (10)
\end{aligned}$$

Here we use a wrapped-normal distribution over angles; therefore, orientation, θ_{t-1}, is propagated in time by adding Gaussian noise and then removing an integer multiple of 2π so that $\theta_t \in [-\pi, \pi)$. The location of the boundary moves with the velocity of the foreground, and therefore its expected location at time t is equal to that at time $t-1$ plus the component of the foreground velocity projected onto the direction of the boundary normal. As above, Gaussian noise is added to represent the modeling errors implicit in this simple dynamical model. Note that more sophisticated models of temporal dynamics (e.g., constant acceleration) could also be used.

3 Probabilistic Framework

Given the generative models described above, we are now ready to formulate our state description and the posterior probability distribution over the models and model parameters. Initially we will make the assumption that there are no probabilistic dependencies between each local image region and its neighboring regions. This allows us to consider each neighborhood independently of all other neighborhoods.

For each single neighborhood, let the *states* be denoted by $\mathbf{s} = (\mu, \mathbf{c})$, where μ is the model type (translation or motion boundary), and $\mathbf{c}$ is a parameter vector appropriate for the model type. For the translational model the parameter vector is 2-dimensional, $\mathbf{c} = (\mathbf{u}_0)$. For the motion boundary model it is 6-dimensional, $\mathbf{c} = (\theta, \mathbf{u}_f, \mathbf{u}_b, d)$. Our goal is to find

the posterior probability distribution over states at time t given the image measurement history up to time t; i.e., $p(\mathbf{s}_t \mid \vec{\mathbf{Z}}_t)$. Here, $\vec{\mathbf{Z}}_t = (\mathbf{z}_t, ..., \mathbf{z}_0)$ denotes the measurement history, where $\mathbf{z}_t$ simply denotes the image at time t, $\mathbf{z}_t \equiv I(\mathbf{x}, t)$.

From the Markov assumption above in the generative models, the conditional independence of the current state at time t and states before time $t-1$ is written as

$$p(\mathbf{s}_t \mid \vec{\mathbf{S}}_{t-1}) \;=\; p(\mathbf{s}_t \mid \mathbf{s}_{t-1}) \;,$$

where $\vec{\mathbf{S}}_t = (\mathbf{s}_t, ..., \mathbf{s}_0)$ denotes the state history. Similarly, the generative model assumes conditional independence of the observations and the dynamics; that is, given $\mathbf{s}_t$ and $\mathbf{z}_{t-1}$, the most recent image observation, $\mathbf{z}_t$, is independent of previous observations $\vec{\mathbf{Z}}_{t-2}$. With these assumptions one can show that the posterior distribution $p(\mathbf{s}_t \mid \vec{\mathbf{Z}}_t)$ can be factored and reduced using Bayes' rule to obtain

$$p(\mathbf{s}_t \mid \vec{\mathbf{Z}}_t) \;=\; k\, p(\mathbf{z}_t \mid \mathbf{s}_t, \mathbf{z}_{t-1})\, p(\mathbf{s}_t \mid \vec{\mathbf{Z}}_{t-1}) \tag{11}$$

where k is a constant used to ensure that the distribution integrates to one. Here, $p(\mathbf{z}_t \mid \mathbf{s}_t, \mathbf{z}_{t-1})$ represents the likelihood of observing the current measurement given the current state, while $p(\mathbf{s}_t \mid \vec{\mathbf{Z}}_{t-1})$ is the prediction of the current state given all previous observations; it is referred to as a temporal prior or prediction density.

The specific form of the likelihood function $p(\mathbf{z}_t \mid \mathbf{s}_t, \mathbf{z}_{t-1})$ follows from the generative models. In particular, the state specifies the motion model and the mapping from visible pixels at time $t-1$ to those at time t. The observation equation, derived from the brightness constancy assumption (4), specifies that the intensity differences between corresponding pixels at times t and $t-1$ should be white and Gaussian, with zero mean and standard deviation σ_n.

Using Bayes' rule and the conditional independence assumed above, it is straightforward to show that the temporal prior, also called the prediction distribution, can be written in terms of the posterior distribution at time $t-1$ and the temporal dynamics that propagate states from time $t-1$ to time t. In particular,

$$p(\mathbf{s}_t \mid \vec{\mathbf{Z}}_{t-1}) \;=\; \int p(\mathbf{s}_t \mid \mathbf{s}_{t-1})\, p(\mathbf{s}_{t-1} \mid \vec{\mathbf{Z}}_{t-1})\, d\,\mathbf{s}_{t-1} \;, \tag{12}$$

where the conditional probability distribution $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ embodies the temporal dynamics, and $p(\mathbf{s}_{t-1} \mid \vec{\mathbf{Z}}_{t-1})$ is the posterior distribution over the state space at time $t-1$.

This completes our description of the state space, and the mathematical form of the posterior probability distribution over the possible interpretations of the motion within an image region.

4 Computational Model: Individual Neighborhood

We now describe the details of our computational embodiment of the probabilistic framework outlined above for a single neighborhood. First, we consider the representation of the posterior distribution and its propagation through time using a particle filter. We then address the computation of the likelihood function and discuss the nature of the prediction distribution that facilitates the state space search for the most probable models and model parameters.

4.1 Particle Filter

The first issue concerns the representation of the posterior distribution, $p(\mathbf{s}_t \,|\, \vec{\mathbf{Z}}_t)$. Because of the non-linear nature of the motion boundary model, the existence of multiple models, and because we expect foreground/background and matching ambiguities, we should assume that $p(\mathbf{s}_t \,|\, \vec{\mathbf{Z}}_t)$ is non-Gaussian, and often multi-modal. For this reason we approximate the posterior distribution non-parametrically, using factored sampling. We then use a particle filter to propagate the posterior through time (Gordon, Salmond, and Smith 1993; Isard and Blake 1998a; Liu and Chen 1998).

The posterior is approximated with a discrete, weighted set of N samples $\{(\mathbf{s}_t^{(i)},\, w_t^{(i)})\}_{i=1\ldots N}$. At each time step, fair samples are drawn from the prediction distribution. The likelihood function is then evaluated at each sample state. Finally, by normalizing the likelihood values so that they sum to one, we obtain the weights $w_t^{(i)}$:

$$w_t^{(i)} = \frac{p(\mathbf{z}_t \,|\, \mathbf{s}_t^{(i)}, \mathbf{z}_{t-1})}{\sum_{n=1}^{N} p(\mathbf{z}_t \,|\, \mathbf{s}_t^{(n)}, \mathbf{z}_{t-1})} \,.$$

These weights ensure that our sample set $\{(\mathbf{s}_t^{(i)},\, w_t^{(i)})\}_{i=1,\ldots,N}$ contains properly weighted samples with respect to the desired posterior distribution $p(\mathbf{s}_t \,|\, \vec{\mathbf{Z}}_t)$ (Liu and Chen 1998). A sufficiently large number of independent samples then provides a reasonable approximation to the posterior.

4.2 Likelihood Function

We assume that the likelihood of observing the current image observations can be written as a product of two factors, namely, a motion likelihood that depends on the difference between frames at time t and $t-1$, and an edge likelihood that depends solely on the band-pass properties of the image at time t:

$$p(\mathbf{z}_t \,|\, \mathbf{s}_t^{(i)}, \mathbf{z}_{t-1}) \;=\; p_m(\mathbf{z}_t \,|\, \mathbf{s}_t^{(i)}, \mathbf{z}_{t-1})\, p_e(\{\psi_k, a_k\}_t \,|\, \mathbf{s}_t^{(i)}) \,. \tag{13}$$

These two likelihood factors are discussed in detail below.

4.2.1 Motion Likelihood

According to the generative model, the motion likelihood $p_m(\mathbf{z}_t \,|\, \mathbf{s}_t^{(i)}, \mathbf{z}_{t-1})$ is straightforward to compute. Given the state, $\mathbf{s}_t^{(i)}$, we can warp one image according to the motion and subtract it from the other. According to the generative model, these image differences at each visible pixel should be normally distributed and independent. The likelihood function, to within a constant κ, is therefore given by

$$p_m(\mathbf{z}_t \,|\, \mathbf{s}_t^{(i)}, \mathbf{z}_{t-1}) \;=\; \kappa \left(\exp\left[\frac{-1}{2\sigma_n^2} \sum_{\mathbf{x}\in\mathcal{R}} D(\mathbf{x}, t; \mathbf{s}_t^{(i)})^2 \right] \right)^{1/T} \tag{14}$$

where $D(\mathbf{x}, t; \mathbf{s}_t^{(i)}) = I(\mathbf{x}', t) - I(\mathbf{x}, t-1)$, $T = |\mathcal{R}|$ is the number of pixels in the circular neighborhood, and $\mathbf{x}'$ denotes the warped image coordinates which depend on the motion encoded in $\mathbf{s}_t^{(i)}$. (The warping here is done simply with bilinear interpolation.)

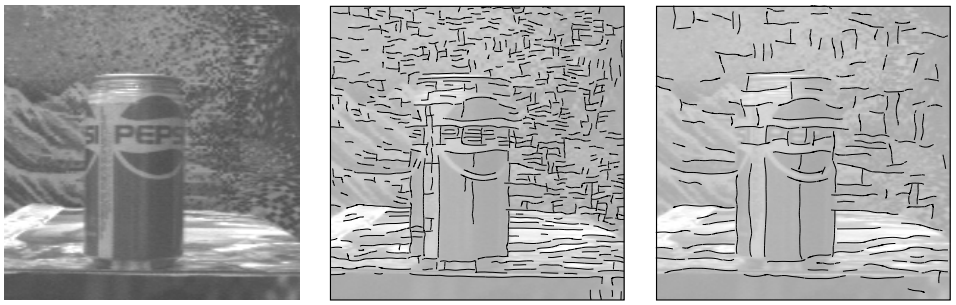

Figure 3 An image is shown, with its dominant level phase contours at $\pm\pi/2$, from the output of filters tuned to vertical and horizontal orientations, and at two different scales.

We note that this likelihood function differs from the generative model in one important respect, that is, the introduction of the exponent $1/T$. This is computationally, rather than probabilistically, motivated. A large value of T has the effect of smoothing the posterior distribution making the peaks broader. Within a sampling framework, this allows a more effective search of the parameter space, reducing the chances of missing a significant peak.

4.2.2 Edge Likelihood

Intensity edges in static images have many physical causes, including surface reflectance variations, lighting effects, and of course surface boundaries. As shown in Fig. 3, not all image edges are motion boundaries. But because motion boundaries are generally caused by depth discontinuities, most motion boundaries do coincide with intensity edges. Edge information can therefore provide useful information about the position and orientation of motion boundaries. To take advantage of this we combine the motion likelihood with the likelihood of observing an image edge, conditioned on the location and orientation of a motion boundary.

We chose the edge likelihood to be the observation density over the responses of an oriented band-pass filter tuned to the edge orientation. Filtering the image in this way removes all oriented image structure except that near the orientation of the edge in question. To do this efficiently for many edges we first apply a steerable transform to the image (Simoncelli, Freeman, Adelson, and Heeger 1992). From the steerable basis set we can quickly compute responses of filters tuned to any orientation. Here, we use the (G_2, H_2) quadrature-pair filters defined in (Freeman and Adelson 1991). These are complex-valued filters so we express their response at each (subsampled) spatial location in terms of amplitude and phase (Fleet and Jepson 1993). The edge likelihood is simply the observation density over phase and amplitude of the subsampled filters responses at points along the edge.

Modeling this observation density can, however, be difficult. The appearance of image edges at surface boundaries depends greatly on surface reflectance properties and on local surface illumination. Given the variability of natural surface textures, and the variability of local lighting, the local structure of images at surface boundaries differs greatly from image to image. Therefore, rather than attempting to design an edge likelihood that captures the variability of edge appearance from first principles, we develop an empirical model for the observation density based on the statistics of natural images.

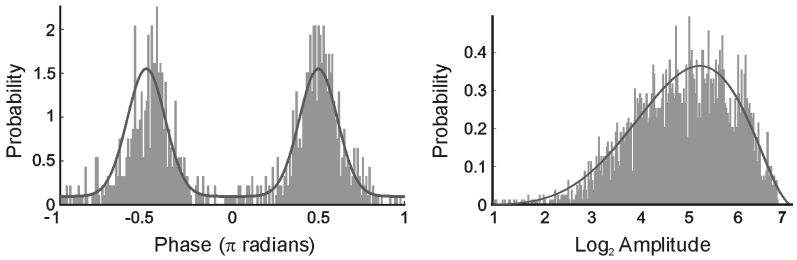

Figure 4 Histograms are shown of (left) phase conditioned on amplitude and the edge, $p_\phi(\phi \,|\, a,\, E)$, and of (right) log amplitude conditioned on the edge, $p_a(a \,|\, E)$.

In short, we manually labelled 800 surface boundaries in 25 images. To each image we applied the steerable filters and extracted phase ϕ and log amplitude a of the responses along each edge at each scale. We sampled these responses with a sampling distance of one wavelength of the filters' tuning frequency. This sparse sampling reduces measurement correlations, and allows us to make the simplifying assumption that the measurements at the different locations along the edge are conditionally independent.

The resulting ensemble of phase and amplitude measurements exhibits a striking regularity that suggests a factorization of the joint observation density:

$$p_e(\phi, a \,|\, \mathbf{s}) \;=\; p_\phi(\phi \,|\, a,\, \mathbf{s})\; p_a(a \,|\, \mathbf{s}) \;. \tag{15}$$

As shown in Fig. 4(left), phase responses, ϕ, are typically close to $\pm\pi/2$, depending on the sign of the intensity gradient at the edge. These conditional phase distributions are very well described by a mixture of two Gaussian modes at $\pi/2$ and $-\pi/2$, and a uniform outlier density. A maximum likelihood fit of this model to the data with the EM algorithm is shown as the solid curve in Fig. 4(left). Alternatively, collapsing the two modes by wrapping the phase about π yields the equivalent density for $\psi \equiv (\phi \bmod \pi)$:

$$p_\psi(\psi \,|\, a,\, \mathbf{s}) \;=\; m(a)\, G(\psi;\, \frac{\pi}{2}, \sigma^2) + (1 - m(a))p_0 \;. \tag{16}$$

where $p_0 = 1/\pi$ is the phase outlier probability, and m is the Gaussian mixing probability.

With this mixture model (16), we find that the mixing probability m depends significantly on log amplitude. When amplitude is very small, phase is unreliable (Fleet and Jepson 1993) and when it is approximately 20% or more of its typical range in 8-bit images, then it is usually quite stable. The standard deviation of the Gaussian is also found to decrease slowly as a function of log amplitude. Using a simple Bayesian model selection criteria (MacKay 1991), we find that a good model for the phase observation density is the mixture in (16) where the standard deviation of the Gaussian mode is held fixed at approximately $\pi/8$, and the mixing probability $m(a)$ is a linear function of log amplitude, given approximately by $m(a) = 0.1(1.5 + \log a)$ where $8 > \log a > 0$ on 8 bit images.

Amplitudes vary widely over the different edges that are encountered in natural scenes. In practice, we find that a simple Beta distribution fits the conditional distribution of amplitudes. An example of this is shown in Fig. 4(right). The Beta distribution is natural in that it is defined on a finite interval which is appropriate for images with a limited range of intensities, and it provides a reasonable approximation to the empirical distribution.

Our edge-based likelihood is given by the factorization in (15), along with the parametric models for the phase and amplitude densities. Given a set of K phase and amplitude measurements, conditioned on a motion boundary state, $\mathbf{s}_t^{(j)}$, the joint likelihood is

$$p_e(\{\psi_k, a_k\} \mid \mathbf{s}_t^{(j)}) = \left(\prod_k p_\psi(\psi_k \mid a_k,\, \mathbf{s}_t^{(j)})\, p_a(a_k \mid \mathbf{s}_t^{(j)}) \right)^{1/2K} \tag{17}$$

Finally, when the state is a smooth motion model, the observation density for the phase is taken to be a uniform density.

4.3 Prediction Distribution

The prediction distribution serves to shepherd our samples to relevant portions of the parameter space. Because we are seeking solutions from a high-dimensional state space with continuous state variables, naive approaches for representing or searching it will be inefficient. Therefore, unlike a conventional particle filter for which the prediction is derived solely by propagating the posterior from the previous time instant, we also exploit an *initialization prior* that provides a form of bottom-up information to initialize new states. This is useful at time 0 when no posterior is available from the previous time instant. It is also useful to help avoid getting trapped at local maxima thereby missing the occurrence of novel events that might not have been predicted from the posterior at the previous time. For example, it helps to detect sudden appearances of motion edges in regions where only translational state samples existed at the previous time instant. This is particularly important since information is not passed between adjacent regions in this formulation.

The actual prediction used here is a linear mixture of a temporal prior and an initialization prior. In the experiments that follow in Section 5 we use constant mixture proportions of 0.8 and 0.2 respectively; that is, 80% of the samples are drawn from the temporal prior. Importance sampling (Gordon, Salmond, and Smith 1993; Isard and Blake 1998b; Liu and Chen 1998) provides an alternative way of achieving similar results.

4.3.1 Temporal Prior

According to the temporal dynamics in generative model for the two motions, (6) through (10), our predictions about the current state, $\mathbf{s}_t$, given the previous state $\mathbf{s}_{t-1}$, are just Gaussian densities. For smooth motion, the temporal dynamics (6) yield

$$p(\mathbf{s}_t \mid \mathbf{s}_{t-1}) = \mathcal{N}(\Delta\mathbf{u}_0, \sigma_u^2) \tag{18}$$

where $\mathcal{N}(\Delta\mathbf{u}_0, \sigma_u^2)$ denotes a mean-zero Gaussian with covariance matrix $\sigma_u^2\, \mathbf{I}_2$, evaluated at the temporal velocity difference $\Delta\mathbf{u}_0 = \mathbf{u}_{0,t} - \mathbf{u}_{0,t-1}$. Similarly, the generative model for the motion boundary ((7) – (10)) specifies that

$$p(\mathbf{s}_t \mid \mathbf{s}_{t-1}) \;=\; \mathcal{N}(\Delta\mathbf{u}_f,\, \sigma_u^2\mathbf{I})\, \mathcal{N}(\Delta\mathbf{u}_b,\, \sigma_u^2\mathbf{I})\, \mathcal{N}^w(\Delta\theta,\, \sigma_d^2)\, \mathcal{N}(\Delta d - \mathbf{n}\cdot\mathbf{u}_{f,t-1},\, \sigma_\theta^2) \tag{19}$$

where $\mathcal{N}^w$ denotes a wrapped-normal (for circular distributions) and, as above, $\Delta\theta = \theta_t - \theta_{t-1}$ and $\Delta d = d_t - d_{t-1}$.

Because the posterior, $p(\mathbf{s}_{t-1}|\vec{\mathbf{Z}}_{t-1})$, at time $t-1$ is represented as a weighted sample set, and the dynamics in (18) and (19) are Gaussian, the temporal prior given by (12) can be viewed as a Gaussian mixture model (West 1992):

$$\sum_{j=1\ldots N} w_{t-1}^{(j)}\, p(\mathbf{s}_t \mid \mathbf{s}_{t-1}^{(j)}) \ . \tag{20}$$

To see this, note that the posterior is approximated by a weighted sum of delta functions (at the sample states). So (12) becomes a convolution of the Gaussian dynamics with the sum of delta functions. The result is a weighted sum of Gaussians, with one Gaussian for each sample $\mathbf{s}_{t-1}^{(j)}$ at time $t-1$. To draw a fair sample from a Gaussian mixture, one first draws a Gaussian component with probabilities equal to the weights. Then, one can draw a random sample from that Gaussian component. This amounts to first selecting a single state, $\mathbf{s}_{t-1}^{(j)}$, for propagation to time t, and then drawing a sample from the Gaussian dynamics, $p(\mathbf{s}_t|\mathbf{s}_{t-1}^{(j)})$. This is repeated for every sample drawn from the temporal prior. In practice, residual sampling and quasi Monte Carlo sampling can be used to reduce random sampling variability (Liu and Chen 1998; Ormoneit, Lemieux, and Fleet 2001).

Thus far we have assumed that the motion class (i.e., smooth or boundary) remains constant as we propagate states from one time to the next. However, when a boundary passes through a region and out the other side, the motion type should switch from a motion boundary model to smooth motion. Accordingly, given a motion boundary state at time $t-1$, we let the probability of switching to a translational model at time t be given by the probability that the temporal dynamics will place the boundary outside the region of interest at time t. This can be computed as the integral of $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ over boundary locations d that fall outside of the region. In practice, we accomplish this by sampling from the temporal prior as described above. Then, whenever we sample a motion boundary state $\mathbf{s}_t^{(j)}$ for which the edge is outside the circular neighborhood, we simply change model types, sampling instead from a translational model whose velocity is consistent with whatever side of the motion boundary would have remained in the region of interest.

4.3.2 Initialization Prior

Low-Level Motion Boundary Detection. To initialize new states and provide a distribution over their parameters from which to sample, we use the motion boundary detector in (Fleet, Black, Yacoob, and Jepson 2000). This approach uses a robust, gradient-based technique for estimating optical flow with a linear parameterized motion model. Motion edges are approximated with a linear basis (2), the coefficients of which are estimated using area-based regression. Fleet *et al.* then solve for the parameters of the motion edge that are most consistent (in a least squares sense) with the linear coefficients.

Figure 5 shows the result of applying this method to two frames of an image sequence in which a camera moves to the right while viewing a Pepsi can sitting on a table. The resulting motions are all leftward as the camera moves to the right, with the can moving somewhat faster than the background. The method provides a mean velocity estimate at each pixel (i.e., the average of the velocities on each side of the motion edge). This is simply the translational velocity when no motion edge is present. A confidence measure, $c(\mathbf{x}) \in [0, 1]$ is used in (Fleet, Black, Yacoob, and Jepson 2000) to detect the most likely edges (Fig. 5, "Confidence"). Fig. 5(bottom) show estimates for the edge orientation and

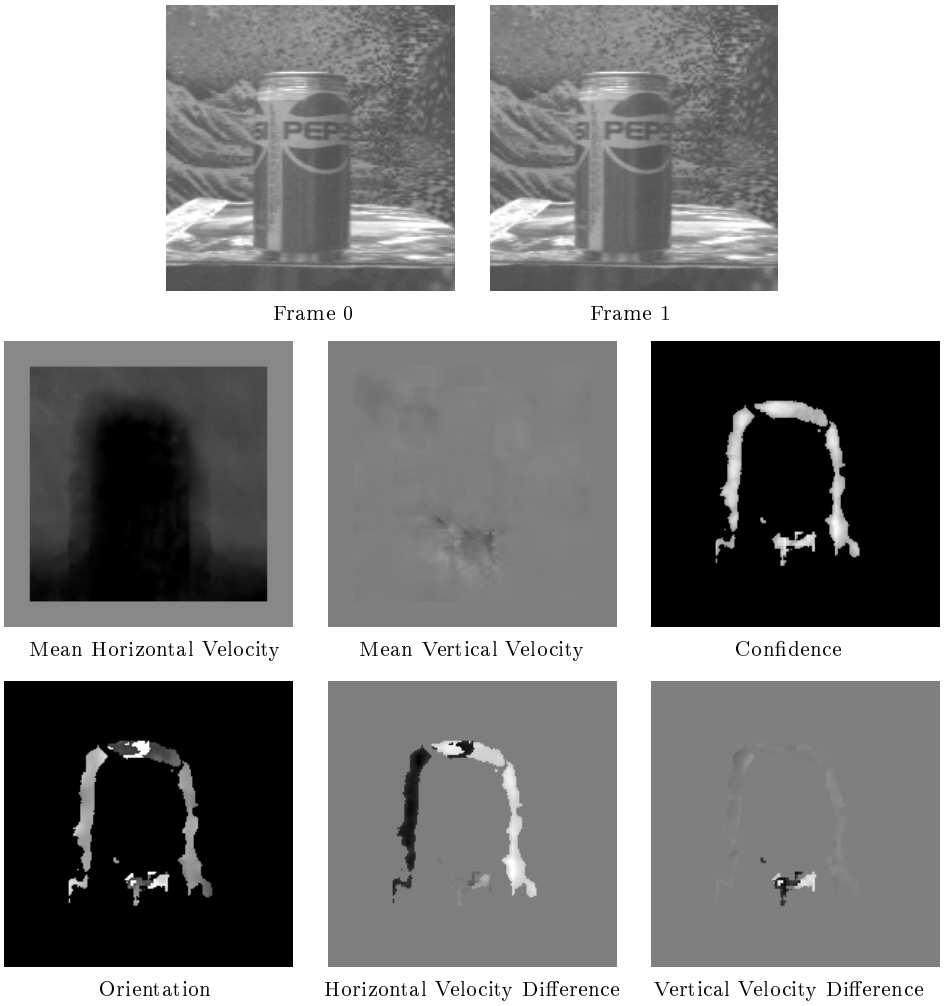

Figure 5 The top row shows two frames of the Pepsi Sequence. The remaining two rows show responses from the low-level motion edge detector. The image velocities and velocity differences are nearly horizontal. In the orientation image, grey denotes vertical orientations, while white and dark grey denote near horizontal orientations.

for the horizontal and vertical velocity differences across the edge, at locations where $c(\mathbf{x}) > 0.5$.

While the method provides reasonable estimates of motion boundaries, it produces false positives and the parameter estimates are corrupted by noise. Localization of the boundary is particularly crude, and since the detector does not determine the foreground side, it does not predict the velocity of the occluding edge. Despite these weaknesses, it is a relatively straightforward, but sometimes error prone, source of information about the presence of motion discontinuities. This information can be used to constrain the regions of the state space that we need to sample in the particle filter.

Formulating the Initialization Prior. When initializing a new state we use the distribution of confidence values $c(\mathbf{x})$ within a region to first decide on the motion type (translation or motion boundary). If a motion boundary is present, then we expect some fraction of confidence values, $c(\mathbf{x})$, within our region of interest, to be high. We therefore rank order the confidence values within the region and let the probability of a motion boundary state be the 95^{th} percentile confidence value, denoted C_{95}. Accordingly, the probability of initializing a translation model is $1 - C_{95}$.

Given that we wish to initialize (sample) a motion boundary state, we assume that actual boundary locations are distributed according to the confidence values in the region; i.e., the boundary is more likely to pass through pixel locations with large $c(\mathbf{x})$. Sampling from the confidence values gives potential boundary locations. Given a boundary position, the low-level detector parameters at that position provide estimates of the edge orientation and the image velocity on each side, but they do not specify which side is the foreground. Thus, the probability distribution over the state space, conditioned on the detector parameters and boundary location, will have two distinct modes, one for each of the two possible foreground assignments. We take this distribution to be a mixture of two Gaussians which are separable with covariance matrices $2.25\sigma_u^2\mathbf{I}_2$ for the velocity axes, and variances $16\sigma_\theta^2$ for the orientation axis and $4\sigma_d^2$ for the position axis. The variances are larger than those used in the temporal dynamics described in Section 2 because we expect greater noise from these low-level estimates.

To produce a sample for the smooth (translation) model, we sample a spatial position according to the distribution of $1 - c(\mathbf{x})$. The distribution over translational velocities is then taken to be a Gaussian centered at the mean velocity estimate of the low-level detector at the sample position, with a covariance matrix of $2.25\sigma_u^2\mathbf{I}_2$

4.4 Algorithm Summary and Model Comparison

Initially, at time 0, a set of N samples is drawn from the initialization prior. Their likelihoods are then computed and normalized to give the weights $w_0^{(i)}$. At each subsequent time, as shown in Fig. 6, the algorithm repeats the process of sampling from the combined prior, computing the likelihoods, and normalizing.

From the non-parametric, sampled approximation to the posterior distribution, $p(\mathbf{s}_t|\vec{\mathbf{Z}}_t)$, we can compute moments and marginalize over various parameters of interest. In particular we can compute the expected value for some state parameter, $f(\mathbf{s}_t)$, as

$$E[f(\mathbf{s}_t) \mid \vec{\mathbf{Z}}_t] = \sum_{n=1\ldots N} f(\mathbf{s}_t^{(n)}) w_t^{(n)}.$$

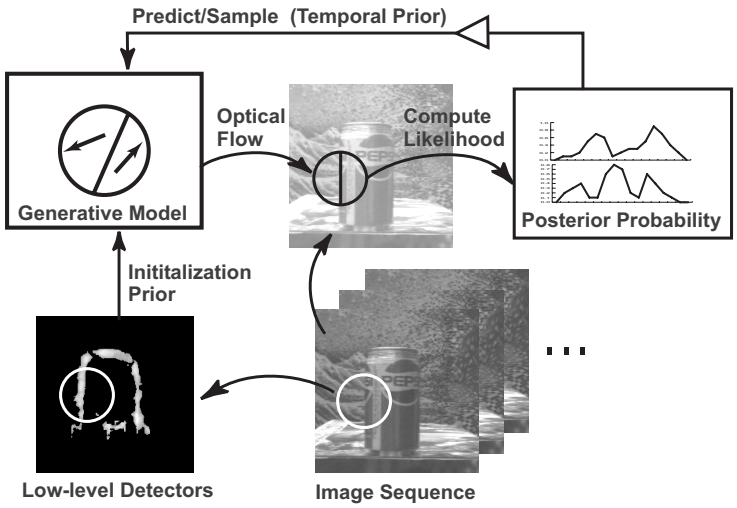

Figure 6 Particle filtering algorithm: State samples are drawn from a mixture of the temporal prior and the initialization prior. The temporal prior combines information from the posterior probability distribution at the previous time instant with the temporal dynamics of the motion models. The initialization prior is derived from the responses of low-level motion boundary detectors within an image region. The parameters of a state determine the image motion within a neighborhood as specified by the generative models for each type of motion. These generative models assume brightness constancy and hence specify how to compute the likelihood of a particular state in terms of the pixel intensity differences between an the image region at one time instant and a warped version of the image at the next time instant. Normalizing the likelihood values for N states gives an approximate, discretely sampled, representation of the posterior probability distribution at the next time instant. In this way the posterior distribution is predicted and updated over time integrating new information within the Bayesian framework.

However, in doing so, care needs to be taken because the posterior will often be multimodal, in which case the mean may not be a highly probable state. Thus, for model comparison and display purposes, we first isolate three distinct modes in the posterior. One mode is often associated with the best fitting smooth motion model. The other two modes are associated with the motion boundary model. These two boundary models typically differ in orientation by π, reflecting two opposite foreground assignments. For display purposes, a simple Bayesian model selection criteria is used to select the mode with the largest cumulative probability mass, and we display only the mean of that most likely mode.

5 Experimental Results: Individual Neighborhoods

We illustrate the method with experiments on 8-bit natural image sequences. For these experiments, the standard deviation of the image noise was $\sigma_n = 7.0$. The standard deviations for the temporal dynamics were empirically determined and remained the same in all experiments. We used circular image regions with a 16 pixel radius and used 3500

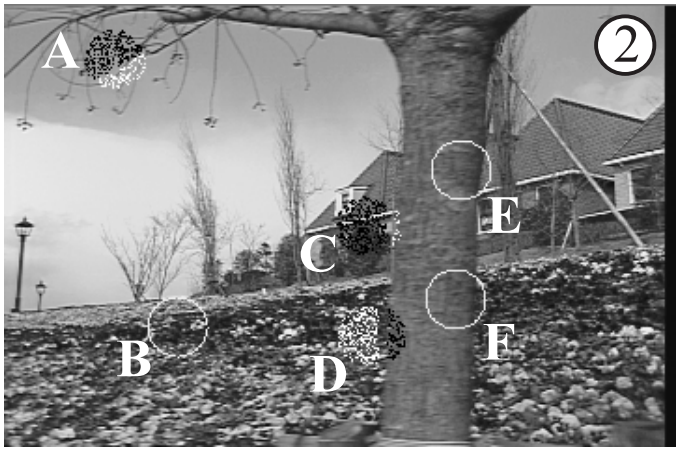

Figure 7 Flower Garden results at frame 2 are shown, with the most likely models overlaid on the image. Translational models are shown as empty circles (as in region **B**). Motion boundaries are shown as filled disks (as in region **D**). The white and black dots, respectively, lie on the foreground and background sides of the model. The position and orientation of the boundaries are depicted by the edges between the white and black sides.

state samples to represent the posterior probability distribution in each region. A few regions were chosen to illustrate the performance of the method and its failure modes.

As shown in Fig. 7, in each of the selected regions we display the mean state of the most likely motion model. The smooth motion (translation) models are shown as empty circles (e.g., Fig. 7, region **B**). For motion boundary models we sample pixel locations from the generative model of the mean state; pixels that lie on the foreground are white and background pixels are black. The position and orientation of the edge are depicted by the boundary between the white and black sides of the region. The occluded pixels are not color-coded (e.g., Fig. 7, region **D**).

Flower Garden Sequence. The flower garden image sequence (Fig. 7) depicts a static scene while a camera translates to the right. Therefore the image velocities are leftward, with the tree moving quickly in front of a slowly moving background. The low-level detector responses for the initialization prior are shown in Fig. 8. The detectors find the occluding and disoccluding sides of the tree and provide reasonable estimates of the edge orientation and the velocities on either side of the boundary. One can see from the confidence map in Fig. 8, however, that the boundary localization is not precise.

Results of the particle filter from frames 2 through 7 are shown in Fig. 9. Regions **C**, **D**, **E**, and **F** correctly model the tree boundary (both occlusion and disocclusion) and, after the first three frames, correctly assign the tree trunk as the foreground side. Initially, in frame 2, regions **C** and **D** detect a motion boundary, but region **D** has incorrectly assigned the foreground to the flower garden rather than the tree. As discussed above, this is not surprising because we expect the correct foreground assignment to require more than two frames. By the third frame, the most likely mode of the posterior corresponds to

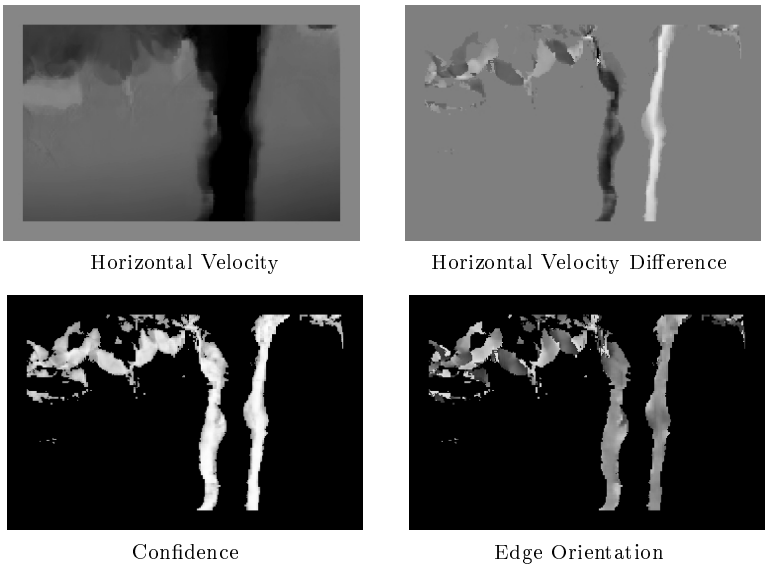

Figure 8 Typical low level detector responses for the flower garden sequence are shown.

the correct assignment of the foreground. Regions **E** and **F** are initially labeled with the smooth motion model since the tree boundary is just touching the right-most edge of the regions. These regions switch to boundary models in the next frame as the tree edge enters the regions. Motion boundary models then remain in all four regions along the tree trunck boundary until the last frame when the edge of the tree leaves the regions.

Beneath each of the images in Fig. 9 are plots that show the marginal posterior distributions for the horizontal component of the foreground velocity for region **D**. Initially, at frame 2, there are two clear modes in the distribution. One mode corresponds to a fast speed, approximately equal to the image speed of the tree trunk, while the other mode corresponds to the slower speed of the flower garden. These two modes reflect the foreground ambiguity, where there is evidence for assigning the foreground to both sides. In frame 2 it is the case that the probability of assigning the foreground to the flower garden in higher. However, with the accumulation of evidence through time, and because this foreground assignment is not consistent with the motion of the boundary, the probability of assigning the foreground to the flower garden decreases, while the probability of assigning the foreground to the tree trunk increases. In frame 3 the probability of assigning the foreground to the tree trunk is slightly larger, and hence the foreground assignment in region **D** switches between frame 2 to frame 3. As time continues the probability associated with this correct foreground assignment increases to become the dominant interpretation.

Region **B** corresponds to translation and is correctly modeled as such. While translation can be equally well accounted for by the motion boundary model, the low-level detectors do not respond in this region and hence the distribution is initialized with more samples corresponding to the translational model. Region **A** is more interesting; if the sky were completely uniform, this region would also be modeled as translation. Note, however, that there are significant low-level detector responses in this area (Fig. 8) due to the fact that

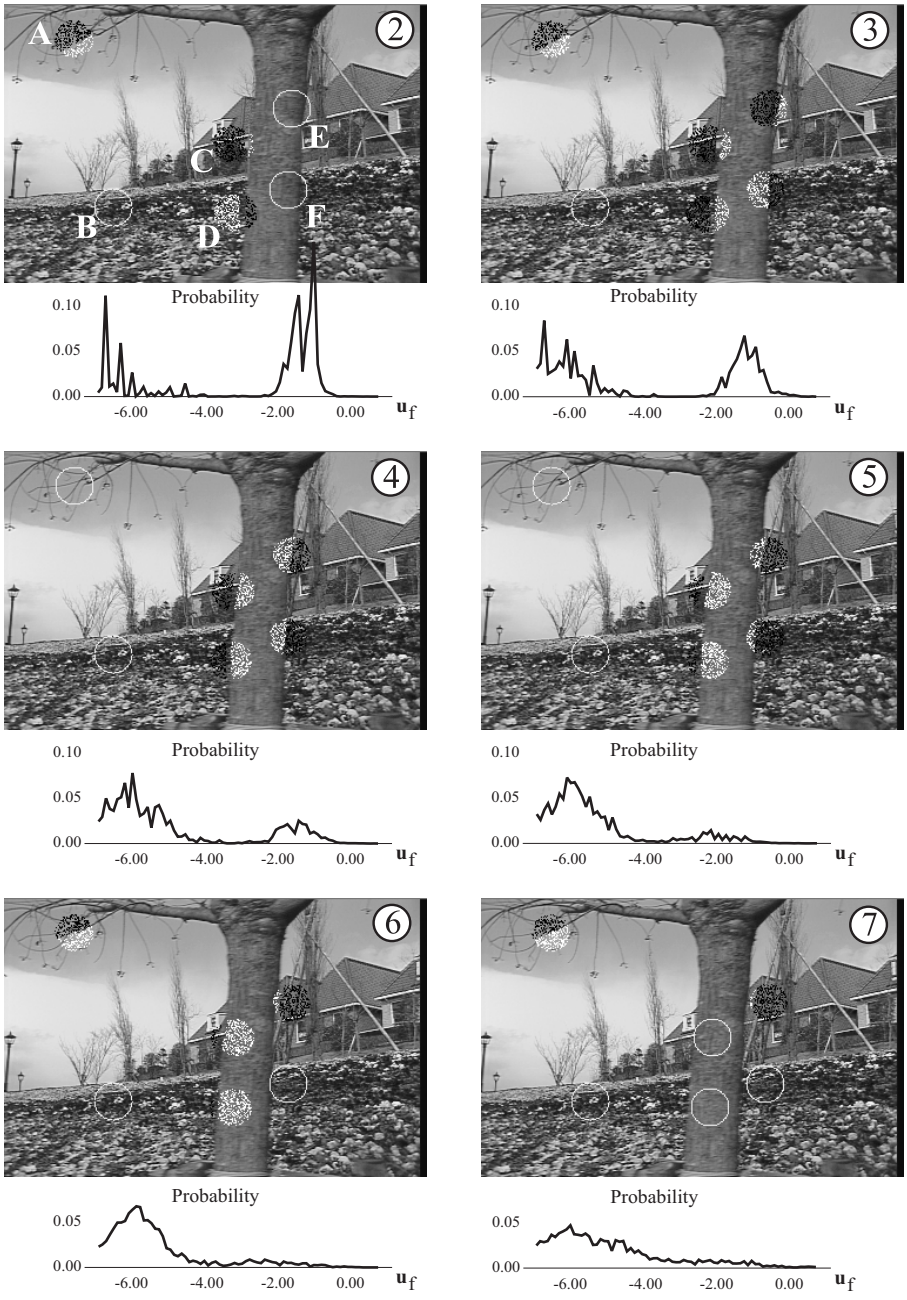

Figure 9 Flower Garden frames 2–7 are shown with the most probable motion models overlaid. Marginal distributions for the foreground velocity in region **D** are also shown.

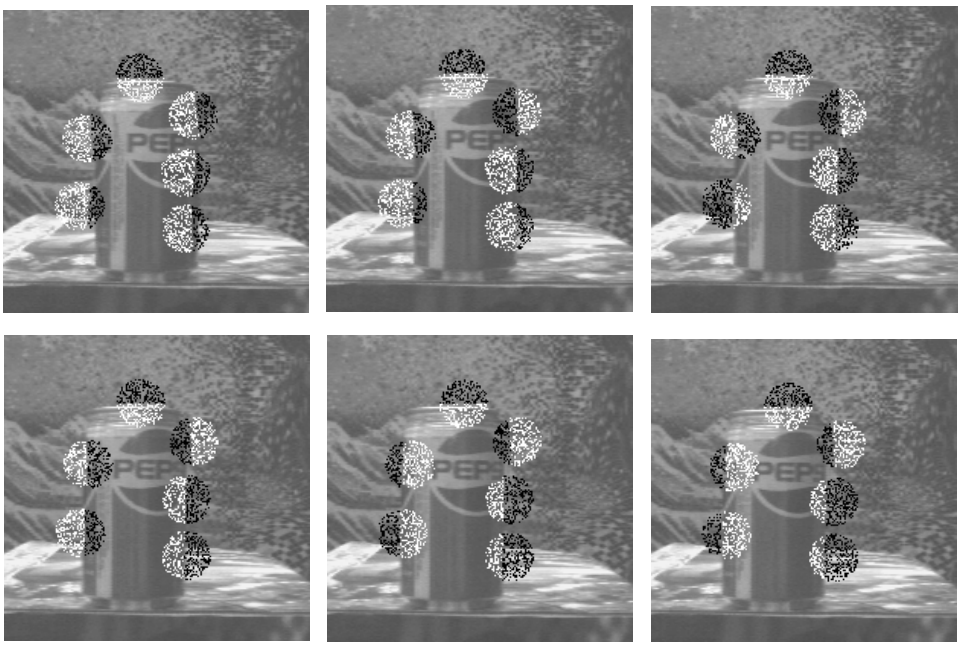

Figure 10 The most probable motion models in selected regions at frames 1, 3, 5, 7, 9, and 10 of the Pepsi Sequence are shown.

the sky is not uniform. The probabilities of the translation and motion boundary models are roughly equal here and the displayed model flips back and forth between them. For the motion boundary model, the orientation corresponds to the orientation of the tree branches in the region.

Pepsi Sequence. The Pepsi Sequence, two frames of which are shown in Fig. 3, depicts a translating camera that views a Pepsi can sitting on a table in front of a textured background. This is a relatively difficult sequence from which to detect motion boundaries because the intensities of the foreground and the background are very similar, and because the can and the background are moving with similar speeds. The difference in the 2D image speeds of the can and the background is less than one pixel per frame. Figure 10 shows the tracking behavior of the method. Fig. 11 shows an enlarged, more detailed, image of the bottom region on the left side of the can. Note that in most regions the edge is tracked correctly and, in the detailed images, we see that the accuracy of the edge boundary location improves over time.

Note that, because the foreground and background velocities are very similar, the foreground/background ambiguity often remains for several frames. For example, consider the region that is enlarged in Fig. 11. Here, in frame 5 there is a switch from the incorrect foreground to the correct assignment and then back again in frame 7. In this case, the posterior distribution has two modes of almost equal probability mass for these two interpretations. Finally, in frame 9 the foreground assignment again switches to the correct

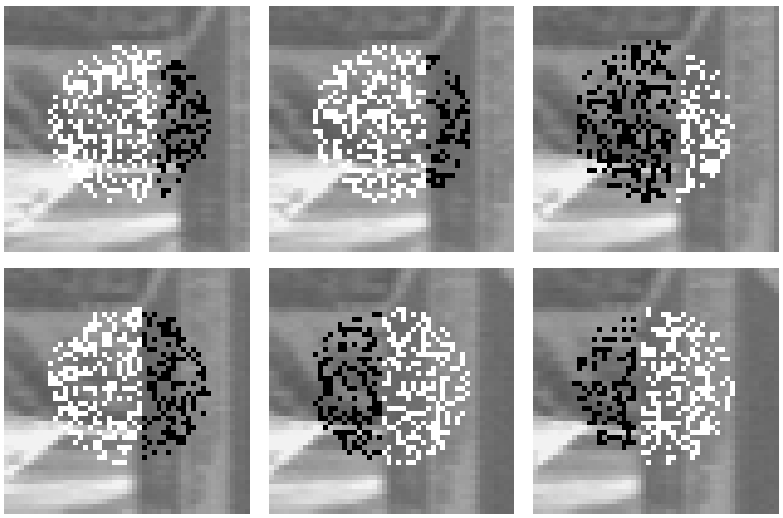

Figure 11 Enlargements of the neighborhood at the bottom-left edge of the Pepsi can from the images in Fig. 10.

interpretation. In general, propagation of information from neighboring regions would be needed to resolve such ambiguities.

Finally, it is important to note that the particle filter does not detect and track motion boundaries in all cases as desired. In the Pepsi Sequence, the region at the top of those on the right side of the can is not tracking the boundary well. A motion boundary is detected in this region but, in the first frame, the most likely mode does not place the edge in the correct location at the correct orientation. Over time the edge appears to move roughly with the can, but it never locks onto the can, nor is the foreground/background assignment correct. This behavior may be the result of low image contrast in this region and the similarity of the image velocities of the two surfaces.

6 Bayesian Filtering with Spatiotemporal Dynamics

Thus far we have only considered a Bayesian formulation for motion anaylsis in a single image region. By assuming that local regions are independent, we greatly simplified the mathematical development and the computational cost of the approach. However, by doing so we also failed to exploit the information that one region could obtain from its neighbors; it is difficult to encourage boundary continuity and to make accurate predictions about boundary locations from one time to the next. For example, when we apply the algorithm described above to a dense array of small local regions with a separate Bayesian filter for each region, we typically find results very much like that shown in Fig. 12(middle). By comparison, Fig. 12(right) shows motion estimates produced when information is propagated between regions using the approach described below.

The main problem with introducing probabilistic dependencies between regions is that the estimation task then involves inference over the entire random field, the posterior and

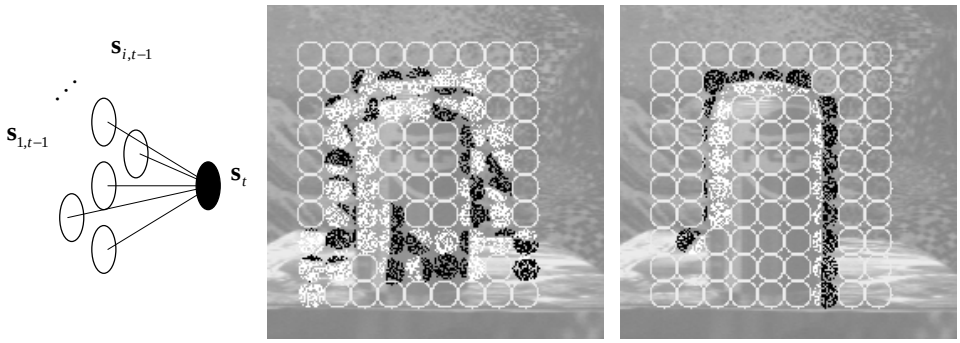

Figure 12 The assumed spatiotemporal dependency is shown on the left. The other two images show motion estimates at frame 6 of the Pepsi sequence without (middle) and with (right) spatiotemporal dependencies.

dynamics of which are no longer easily factored into products of the marginal distributions of individual regions. The research described in the remainder of this chapter is an attempt to consider a form of approximate inference that allows us include a some degree of spatiotemporal dependence. First, as illustrated in Fig. 12(left), we assume a simple form of dependence where each region at time t, conditioned on nearby regions at time $t-1$, is independent of other regions at current and past times. In doing so, our goal is to pass information between regions from one time to the next, but not between adjacent regions at the same time. Second, we continue to approximate the joint posterior distribution over the motion in all image regions by the marginal distributions for each region. As shown in Fig. 12, even with these crude approximations, the resulting inference often provides a vast improvement over the case in which regions are treated completely separately.

To accommodate spatiotemporal predictions, the main change required of the formulation in Section 3 concerns the prediction distribution in (12). Following the graphical model in Fig. 12(left), let $\{\mathbf{s}_{i,t-1}\}_{i=1}^{M}$ denote the M neighbors at time $t-1$ that influence a specific region $\mathbf{s}_t$ at time t. We begin by writing the prediction distribution as a marginalization of the joint distribution for $\mathbf{s}_t$ and its neighbors $\{\mathbf{s}_{i,t-1}\}_{i=1}^{M}$:

$$p(\mathbf{s}_t \,|\, \vec{\mathbf{Z}}_{t-1}) \;=\; \int_{\{\mathbf{s}_{i,t-1}\}} p(\mathbf{s}_t, \{\mathbf{s}_{i,t-1}\}_{i=1}^{M} \,|\, \vec{\mathbf{Z}}_{t-1}) \;. \tag{21}$$

Factoring the integrand, and exploiting assumed conditional independence, yields

$$p(\mathbf{s}_t \,|\, \vec{\mathbf{Z}}_{t-1}) \;=\; \int_{\{\mathbf{s}_{i,t-1}\}} p(\mathbf{s}_t \,|\, \{\mathbf{s}_{i,t-1}\})\; p(\{\mathbf{s}_{i,t-1}\} \,|\, \vec{\mathbf{Z}}_{t-1}) \,. \tag{22}$$

The dynamics, $p(\mathbf{s}_t \,|\, \{\mathbf{s}_{i,t-1}\})$, can be factored if we assume that the neighbors at time $t-1$ have uniform priors and are independent when conditioned on $\mathbf{s}_t$. The joint posterior over all neighbors cannot be factored in general. However, for computational efficiency, as above, we approximate the joint posterior as a product of its marginals (cf. (Murphy and Weiss 2001)), to yield

$$p(\mathbf{s}_t \,|\, \vec{\mathbf{Z}}_{t-1}) \;\approx\; \kappa \prod_{i=1\ldots M} \int_{\mathbf{s}_{i,t-1}} p(\mathbf{s}_t \,|\, \mathbf{s}_{i,t-1})\; p(\mathbf{s}_{i,t-1} \,|\, \vec{\mathbf{Z}}_{t-1}) \tag{23}$$

where κ is a constant to ensure that the distribution has unit probability mass. Note that this approximate prediction distribution is now the product of the predictions from each neighboring location, each of which has the form of (12).

6.1 Particles and Gaussian Mixtures

The simplified prediction distribution in (23) allows us to combine predictions from each of the neighbors at time $t-1$ in a straightforward manner. As with the particle filter above, we approximate each distribution $p(\mathbf{s}_{i,t-1} \mid \vec{\mathbf{Z}}_{t-1})$ with a weighted set of N samples $\{\mathbf{s}_{i,t-1}^{(j)}, w_{i,t-1}^{(j)}\}_{j=1}^{N}$. Accordingly, when this distribution is propagated through the dynamics, the approximate prediction distribution for $\mathbf{s}_t$, conditioned on the state of a single neighbor, $\mathbf{s}_{i,t-1}$, is just a mixture model:

$$\sum_{j=1\ldots N} w_{i,t-1}^{(j)}\, p(\mathbf{s}_t \mid \mathbf{s}_{i,t-1}^{(j)}) \ . \tag{24}$$

From this perspective, the multi-neighbor prediction distribution in (23) is just a product of mixture models. However, because we typically use thousands of particles (e.g., $N = 10^3$), and about $M = 5$ neighbors, the number of components in the product, i.e., N^M, quickly becomes unmanageable. We overcome this problem by fitting a mixture model to the individual prediction distributions prior to their multiplication in (23). We use mixture models with a small number of Gaussian components (often 3 to 5) plus a uniform outlier process. As a result, the product in (23) reduces to fewer than 10^3 components. The mixture models are fit with a straightforward version of the EM algorithm (Dempster, Laird, and Rubin 1977).

Note that we first propagate individual samples from the neighboring posteriors at the previous time, and then we fit the mixture model. As with *assumed density filtering* and *unscented filtering*, this is done because it is relatively easy to propagate individual samples through nonlinear dynamics. The final prediction distribution in (23) is obtained by multiplying the individual mixture model predictions.

7 Computational Model: Spatiotemporal Predictions

Given weighted sample sets that approximate the posterior distribution in each local region at time $t-1$, the steps toward the computation of the posterior distribution in a specific region at time t can be summarized as follows:

1. For each neighbor i at the previous time $t-1$:
   - Draw N samples with replacement from the posterior at $t-1$, $\{\mathbf{s}_{i,t-1}^{(j)}, w_{i,t-1}^{(j)}\}_{j=1}^{N}$.
   - Propagate the samples using the model dynamics (Sec. 7.1), and then sample from the prediction density (24) to get a new sample set at time t.
   - Use EM to fit the robust mixture model to the new sample set.
2. Multiply the individual mixture models to form the joint prediction distribution (23).
3. Draw N samples with replacement from this prediction and compute their likelihoods.
4. Normalize the likelihoods to obtain the sample weights.

This yields a weighted sample set $\{\mathbf{s}_t^{(j)}, w_t^{(j)}\}$ that approximates the posterior for a region at time t, $p(\mathbf{s}_t \mid \vec{\mathbf{Z}}_t)$.

7.1 Temporal Dynamics

The final issue we must now consider is the form of temporal dynamics that is suitable for the spatiotemporal dependencies. Since the joint prediction in (23) is the product of the individual predictions, we need only specify the form of the dynamics between a state $\mathbf{s}_t$ and a single neighbor $\mathbf{s}_{i,t-1}$ at time $t-1$. As this is somewhat more complicated than the case developed in Section 4, here we describe dynamics in more detail.

First, it is useful to expand the state $\mathbf{s}$ into its discrete and continuous components, μ and $\mathbf{c}$. This allows us to express the pair-wise prediction distribution as

$$p(\mathbf{s}_t|\vec{\mathbf{Z}}_{t-1}) \;=\; \sum_{\mu_{t-1}} \int_{\mathbf{c}_{t-1}} [\, p(\mathbf{c}_t \,|\, \mu_t, \mu_{t-1}, \mathbf{c}_{t-1})\, p(\mu_t \,|\, \mu_{t-1}, \mathbf{c}_{t-1})\; p(\mu_{t-1}, \mathbf{c}_{t-1} \,|\, \vec{\mathbf{Z}}_{t-1})\,] \,, \quad (25)$$

where $p(\mu_t \,|\, \mu_{t-1}, \mathbf{c}_{t-1})$ and $p(\mathbf{c}_t \,|\, \mu_t, \mu_{t-1}, \mathbf{c}_{t-1})$ denote the discrete and continuous transition distributions. To avoid singularities (where probabilities go to 0) and to allow for modeling errors in the dynamics, we let both distributions be robust; that is, we define

$$\begin{aligned} p(\mu_t \,|\, \mu_{t-1}, \mathbf{c}_{t-1}) &= \alpha\, p_\mu(\mu_t \,|\, \mu_{t-1}, \mathbf{c}_{t-1}) \;+\; (1-\alpha)\, p_{\mu,0} \\ p(\mathbf{c}_t \,|\, \mu_t, \mu_{t-1}, \mathbf{c}_{t-1}) &= \beta\, p_{\mathbf{c}}(\mathbf{c}_t \,|\, \mu_t, \mu_{t-1}, \mathbf{c}_{t-1}) \;+\; (1-\beta)\, p_{\mathbf{c},0} \end{aligned}$$

where $p_{\mu,0}$ and $p_{\mathbf{c},0}$ are uniform outlier distributions for discrete and continuous state variables, with mixing probabilities α and β. The inlier dynamics, $p_\mu(\mu_t \,|\, \mu_{t-1}, \mathbf{c}_{t-1})$ and $p_{\mathbf{c}}(\mathbf{c}_t \,|\, \mu_t, \mu_{t-1}, \mathbf{c}_{t-1})$, are summarized in Tables 1 and 2.

Referring to Table 1, where $\mu = 0$ denotes the smooth motion model and $\mu = 1$ denotes the motion boundary model, we assume that smooth motion states will encounter an edge and switch to a boundary model with probability $p_{0\to1}$. In the case of motion boundaries, as in Section 4, we assume dynamics such that edges move with the foreground velocity on average, and that there exists mean-zero Gaussian process noise in the velocities and the boundary orientation otherwise. Then, given a motion boundary state $\mathbf{s}_{i,t-1}$ in the region centered at $\mathbf{x}_{t-1}$, the distribution of motion boundary parameters at time t for the state $\mathbf{s}_t$ in a region centered at $\mathbf{x}_t$ is given by

$$f(\mathbf{c}_{e,t}|\mathbf{c}_{e,t-1}) \quad = \quad \mathcal{N}((\mathbf{u}_{f_{t-1}}, \mathbf{u}_{b_{t-1}}), \sigma_u^2 \mathbf{I}_4)\; \mathcal{N}^w(\theta_{t-1}, \sigma_\theta^2)\; \mathcal{N}(loc(\mathbf{c}_{e,t-1}), \sigma_d^2) \quad (26)$$

where $loc(\mathbf{c}_{e,t-1}) \equiv d_{t-1} + (\mathbf{u}_{f_{t-1}} + \mathbf{x}_{t-1} - \mathbf{x}_t) \cdot \hat{\mathbf{n}}_{t-1}$ is the mean edge location at time t relative to the region center $\mathbf{x}_t$, and $\hat{\mathbf{n}}_{t-1} = (\sin(\theta_{t-1}), \cos(\theta_{t-1}))$. Given this distribution, we define the probability of changing from an edge state at $\mathbf{x}_{t-1}$ to a smooth motion state at $\mathbf{x}_t$ as the probability of the edge not intersecting the region at $\mathbf{x}_t$ at time t: that is,

$$p_{1\to0} = \int_{|d_t|>R} \mathcal{N}_{d_t}(loc(\mathbf{c}_{e,t-1}), \sigma_d^2) \quad (27)$$

where $\mathcal{N}_{d_t}$ is the probability density for d_t, and R is the region radius.

Table 2 defines the continuous prediction distributions conditioned on the discrete motion classes. For example, if the neighbor state at time $t-1$ and the current state are both smooth motions, then the current velocity is normally distributed about the velocity of the previous state. If the previous state was a motion boundary and the current state

Neighbor \ Current	$\mu_t = 0$	$\mu_t = 1$
$\mu_{t-1} = 0$	$1-p_{0\to 1}$	$p_{0\to 1}$
$\mu_{t-1} = 1$	$p_{1\to 0}$	$1-p_{1\to 0}$

Table 1 Discrete transition probabilities $p_\mu(\mu_t|\mu_{t-1}, \mathbf{c}_{t-1})$ from a motion class at the neighbor (i.e., $\mathbf{s}_{i,t-1}$) to the motion class for $\mathbf{s}_t$, where $\mu = 0$ denotes the smooth motion model and $\mu = 1$ denotes the motion boundary model.

Neighbor \ Current	$\mu_t = 0$
$\mu_{t-1} = 0$	$p_{\mathbf{c}}(\mathbf{u}_t|\mathbf{u}_{t-1}) = \mathcal{N}(\mathbf{u}_{t-1}, \sigma_u^2\mathbf{I}_2)$
$\mu_{t-1} = 1$	if $\mathbf{x}_t$ is in neighbor's foreground $p_{\mathbf{c}}(\mathbf{u}_t|\mathbf{c}_{e,t-1}) = \mathcal{N}(\mathbf{u}_{f_{t-1}}, \sigma_u^2\mathbf{I}_2)$ else $p_{\mathbf{c}}(\mathbf{u}_t|\mathbf{c}_{e,t-1}) = \mathcal{N}(\mathbf{u}_{b_{t-1}}, \sigma_u^2\mathbf{I}_2)$

Neighbor \ Current	$\mu_t = 1$
$\mu_{t-1} = 0$	$p_{\mathbf{c}}(\mathbf{c}_{e,t}|\mathbf{u}_{t-1}) = p(\theta_t, d_t)\, p(\mathbf{u}_{f_t}|\mathbf{u}_{t-1})\, p(\mathbf{u}_{b_t}|\mathbf{u}_{t-1})$ where $p(\theta_t, d_t) = \text{Uniform}(\theta_t, d_t|\text{edge outside neighbor})$ if $\mathbf{x}_{t-1}$ is in current foreground $p(\mathbf{u}_{f_t}|\mathbf{u}_{i,t-1}) = \mathcal{N}(\mathbf{u}_{t-1}, \sigma_u^2\mathbf{I}_2)$; $p(\mathbf{u}_{b_t}|\mathbf{u}_{i,t-1}) = \mathcal{N}(\mathbf{0}, 50\mathbf{I}_2)$ (broad prior); else $p(\mathbf{u}_{f_t}|\mathbf{u}_{i,t-1}) = \mathcal{N}(\mathbf{0}, 50\mathbf{I}_2)$ (broad prior); $p(\mathbf{u}_{b_t}|\mathbf{u}_{i,t-1}) = \mathcal{N}(\mathbf{u}_{t-1}, \sigma_u^2\mathbf{I}_2)$;
$\mu_{t-1} = 1$	$p_{\mathbf{c}}(\mathbf{c}_{e,t}|\mathbf{c}_{e,t-1}) = f(\mathbf{c}_{e,t}|\mathbf{c}_{e,t-1})\mathbb{1}(|d_t| < R) \,/\, (1-p_{1\to 0})$ where $\mathbb{1}(|d_t| < R) = 1$ when $|d_t| < R$, and 0 otherwise

Table 2 Model dynamics, $p_{\mathbf{c}}(\mathbf{c}_t\,|\mu_t, \mu_{t-1}, \mathbf{c}_{t-1})$, for the continuous parameters, conditioned on the discrete motion classes ($\mu = 0$ for smooth motion and $\mu = 1$ for motion boundary). Here, $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are the centers of the current and neighbor regions at times t and $t-1$. The variances, $\sigma_u^2, \sigma_\theta^2$, and σ_d^2, control the process noise in the dynamics; we let each of them increase as a function of the spatial distance between the region centers $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$. We omitted the dependence on the neighbor (i) for notational simplicity.

is smooth, then the current velocity is normally distributed about the foreground or background velocity, depending on whether the current region is on the foreground or background side of the previous region. If the previous state was smooth and the current state is a boundary, then θ_t and d_t are uniformly distributed over values for which the edge does not intersect the previous region, and the velocity distributions depend on the previous velocity state. Finally, if previous and current states are motion boundaries, then the distribution over the current state is Gaussian, but only for parameters such that the edge intersects the current region.

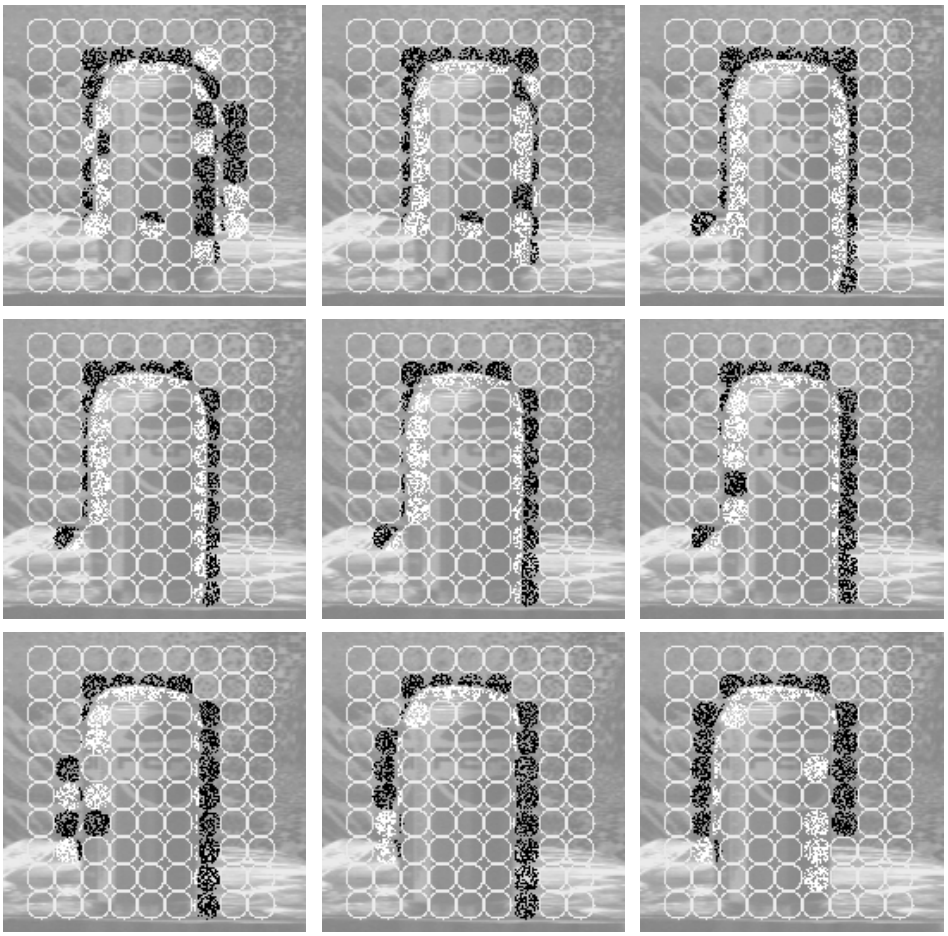

Figure 13 Pepsi results for frames 2–10 (in lexicographic order and cropped slightly to improve the resolution of the display).

This dynamical model is applied to individual particles. The nonlinear components of the dynamics include the model switching and the computation of the propagated edge distance, which depends on the normal to the edge direction $\hat{\mathbf{n}}_{t-1} = (\sin(\theta_{t-1}), \cos(\theta_{t-1}))$. Nonlinearities make it difficult to propagate distributions analytically, even if the neighbor posterior at the time $t-1$ had been Gaussian.

8 Experimental Results: Spatiotemporal Predictions

We demonstrate some experimental results of this approach applied to the Pepsi Sequence. We use small circular regions with radii of 8 pixels which overlap by 2 pixels. We use 5000 samples for particle approximations in each region. We draw 10% of the particles from the initialization prior, and the remaining 90% from the prediction density in (23). The

parameters for the dynamics between a location at time t and a neighbor at time $t-1$ depend on the spatial separation between the two locations. For the same spatial location at t and $t-1$ we use $\sigma_u = 0.75$ pixel/frame, $\sigma_\theta = 0.1$ radians, and $\sigma_d = 1$ pixel. For an adjacent region at $t-1$ we use $\sigma_u = 1.5$ pixels/frame, $\sigma_\theta = 0.2$ radians, and $\sigma_d = 1.5$. In both cases $\alpha = 0.975$ and $\beta = 0.95$. Finally, the probability of a motion boundary, conditioned on the motion of a neighbor being smooth, is $p_{0\to 1} = 0.4$; this value reflects the fact that edges occur in roughly 10% of the image regions, and that such motion boundary predictions are relatively unconstrained, requiring a large number of samples to search the state space effectively.

Figure 13 shows results from frames 2–10 of the Pepsi Sequence. At frame 1, the results look very much like those in of the method above in which individual regions are treated separately (see Fig. 12(middle)). By frame 2 the neighborhood interactions appear to introduce some coherence. By frame 3, compared to the results of obtained with individual regions, it is clear that the current method produces more coherent boundary estimates. Noteworthy in Fig. 13 are the correct assignment of the foreground and the accurate localization of the motion boundaries. Also evident in Fig. 13, is the importance of the neighborhood propagation that allows regions to anticipate the arrival of a boundary from a neighboring region. This is evident in frames 7–9 on the left boundary and later in frames 9–10 on the right side. This propagation allows the correct assignment of the foreground to be infered quickly.

9 Conclusions

Research on image motion estimation has typically exploited relatively weak models of the spatiotemporal structure of image motion. Our goal is to move towards a richer description of image motion using a vocabulary of motion primitives. Here we describe a step in that direction with the introduction of an explicit non-linear model of motion boundaries and a Bayesian framework for representing a posterior probability distribution over models and model parameters. Unlike previous work that attempts to find a maximum-likelihood estimate of image motion, we represent the probability distribution over the parameter space using discrete samples. This facilitates the correct Bayesian propagation of information over time when ambiguities make the distribution non-Gaussian.

However, exact Bayesian inference for this problem, like many problems in vision, is not tractable. As a result we explore different forms of approximate inference. Particle filters are effective for visual tracking, allowing for a Bayesian framework even with non-Gaussian distributions and non-linear dynamics. Here we extend their use, in conjunction with other methods for approximate inference, to the detection and estimation of multiple motion models defined over a random field. In particular, we consider the detection and tracking of motion boundaries for which predictions of motion and of boundary locations/orientations are obtained from nearby image regions at the previous time. This helps to encourage boundary continuity, and to direct samples to the appropriate regions of the state space as an edges leaves one region and enters another. It also improves the inference of surface depth ordering.

This work represents an early effort in what we hope will be a rich area of inquiry. In particular, we can now begin to think about the spatial interaction of these and other local motion models. For example, we might formulate a more eleborate probabilistic

spatial "grammar" of motion features and how they relate to their neighbors in space and time. This raises the question of what is the right vocabulary for describing image motion and what role learning may play in formulating local models and in determining spatial interactions between them (see (Freeman and Pasztor 1999)). In summary, the techniques described here (generative models, Bayesian propagation, and approximate inference with Monte Carlo methods) permit us to explore problems within motion estimation that were previously inaccessible.

Acknowledgments

We thank Allan Jepson for many discussions about motion discontinuities, generative models, sampling methods, probability theory and, of course, GRITS. Also thanks to Ray Luo for helping to validate the edge likelihood model.

References

Barron, J. L., D. J. Fleet, and S. S. Beauchemin (1994). Performance of optical flow techniques. *International Journal of Computer Vision 12*(1), 43–77.

Beauchemin, S. S. and J. L. Barron (2000). The local frequency structure of 1d occluding image signals. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22*(2), 200–206.

Belhumeur, P. (1996). A Bayesian-approach to binocular stereopsis. *International Journal of Computer Vision 19*(3), 237–260.

Bergen, J. R., P. Anandan, K. Hanna, and R. Hingorani (1992). Hierarchical model-based motion estimation. In *Proc. European Conf. on Computer Vision*, pp. 237–252. Springer-Verlag.

Black, M. J. and P. Anandan (1990). Constraints for the early detection of discontinuity from motion. In *Proc. National Conf. on Artificial Intelligence, AAAI-90*, Boston, MA, pp. 1060–1066.

Black, M. J. and P. Anandan (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding 63*(1), 75–104.

Black, M. J. and D. J. Fleet (2000). Probabilistic detection and tracking of motion discontinuities. *International Journal of Computer Vision 38*(3), 229–243.

Black, M. J. and A. D. Jepson (1998). EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision 26*(1), 63–84.

Broida, T. J., S. Chandrashekhar, and R. Chellappa (1990). Recursive 3-d motion estimation from a monocular image sequence. *IEEE Trans. Aerosp. Electron. Syst. 26*(4), 639–656.

Chiuso, A., P. Favarto, H. Jin, and S. Saotto (2000). 3D motion and structure from 2D motion causally integrated over time: Implementation. In D. Vernon (Ed.), *European Conference on Computer Vision*, Dublin, Part 2, pp. 734–750. Springer-Verlag.

Choo, K. and D. J. Fleet (2001). People tracking with hybrid Monte Carlo. In *Proc. IEEE International Conference on Computer Vision*, Volume II, Vancouver, pp. 321–328.

Chou, G. T. (1995). A model of figure-ground segregation from kinetic occlusion. In *IEEE International Conference on Computer Vision*, Boston, pp. 1050–1057.

Cornelius, N. and T. Kanade (1981). Adapting optical flow to measure object motion in reflectance and X-ray image sequences. In *Proc. ACM Workshop on Motion: Representation and Perception*, Toronto, pp. 50–58.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society B 39*, 1–38.

Deutscher, J., A. Blake, and I. Reid (2000). Articulated body motion capture by annealled particle filtering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Volume II, Hilton Head, pp. 126–133.

Doucet, A., N. de Freitas, and N. Gordon (2001). *Sequential Monte Carlo Methods in Practice*. Berlin: Springer-Verlag.

Fennema, C. L. and W. B. Thompson (1979). Velocity determination in scenes containing several moving objects. *Computer Vision, Graphics, and Image Processing 9*, 301–315.

Fleet, D. J. (1992). *Measurement of Image Velocity*. Boston: Kluwer.

Fleet, D. J., M. J. Black, Y. Yacoob, and A. D. Jepson (2000). Design and use of linear models for image motion analysis. *International Journal of Computer Vision 36*(3), 169–191.

Fleet, D. J. and A. D. Jepson (1993). Stability of phase information. *IEEE Trans. on Pattern Analysis and Machine Intelligence 15*, 1253–1268.

Fleet, D. J. and K. Langley (1994). Computational analysis of non-fourier motion. *Vision Research 22*, 3057–3079.

Freeman, W. and E. H. Adelson (1991). The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence 13*, 891–906.

Freeman, W. and E. Pasztor (1999). Learning to estimate scenes from images. In *Adv. Neural Information Processing Systems*, Volume 11.

Gibson, J. (1950). *The Perception of the Visual World*. Boston: Houghton Mifflin.

Gilks, W. R., S. Richardson, and D. J. Spiegelhalter (1996). *Markov Chain Monte Carlo Methods in Practice*. London: Chapman and Hall.

Gordon, N. J., D. J. Salmond, and A. F. M. Smith (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. on Radar, Sonar and Navigation 140*(2), 107–113.

Hager, G. D. and P. N. Belhumeur (1998). Efficient region tracking with parametric models of geometry and llumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence 27*(10), 1025–1039.

Harris, J. G., C. Koch, E. Staats, and J. Luo (1990). Analog hardware for detecting discontinuities in early vision. *International Journal of Computer Vision 4*(3), 211–223.

Heeger, D. J. and A. D. Jepson (1992). Subspace methods for recovering rigid motion I: Algorithms and implementation. *International Journal of Computer Vision 7*(2), 95–117.

Heitz, F. and P. Bouthemy (1993). Multimodal motion estimation of discontinuous optical flow using Markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence 15*(12), 1217–1232.

Horn, B. K. P. (1986). *Robot Vision*. Cambridge, Massachusetts: MIT Press.

Horn, B. K. P. and B. G. Schunk (1981). Determining optical flow. *Artificial Intelligence 17*, 185–203.

Irani, M., B. Rousso, and S. Peleg (1994). Computing occluding and transparent motions. *International Journal of Computer Vision 12*(1), 5–16.

Isard, M. and A. Blake (1998a). Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision 29*(1), 2–28.

Isard, M. and A. Blake (1998b). Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In H. Burkhardt and B. Neumann (Eds.), *European Conference on Computer Vision, ECCV-98*, Freiburg, pp. 893–908. Springer-Verlag.

Jepson, A. and M. J. Black (1993). Mixture models for optical flow computation. In *Proc. IEEE Computer Vision and Pattern Recognition*, New York, pp. 760–761.

Jepson, A. D., D. J. Fleet, and T. F. El-Maraghi (2001). Robust online appearance models for visual tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Volume I, Kauai, pp. 415–422.

Kitagawa, G. (1987). Non-gaussian state-space modelling of non-stationary time series. *Journal of the American Statistical Association 82*, 1032–1063.

Konrad, J. and E. Dubois (1998). Multigrid Bayesian estimation of image motion fields using stochastic relaxation. In *Proc. IEEE International Conference on Computer Vision*, Tampa, Florida, pp. 354–362.

Liu, J. S. and R. Chen (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association 93*(443), 1032–1044.

Longuet-Higgins, H. C. and K. Prazdny (1980). The interpretation of a moving retinal image. *Proc. Royal Society London B-208*, 385–397.

MacCormick, J. and M. Isard (2000). Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. European Conference on Computer Vision*, Volume II, Dublin, pp. 134–149.

MacKay, D. J. C. (1991). Bayesian interpolation. *Neural Computation 4*, 415–447.

Murphy, K. and Y. Weiss (2001). The factored frontier algorithm for approximate inference in DBNs. In *Proc. Uncertainty in Artificial Intelligence*, Seattle, pp. 378–385.

Murray, D. W. and B. F. Buxton (1987). Scene segmentation from visual motion using global optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence 9*(2), 220–228.

Mutch, K. and W. Thompson (1985). Analysis of accretion and deletion at boundaries in dynamic scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence 7*(2), 133–138.

Nagel, H. H. and W. Enkelmann (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence 8*(5), 565–593.

Nestares, O. and D. J. Fleet (2001). Probabilistic tracking of motion boundaries with spatiotemporal predictions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Volume II, Kauai, pp. 358–365.

Niyogi, S. A. (1995). Detecting kinetic occlusion. In *IEEE International Conference on Computer Vision*, Boston, pp. 1044–1049.

Ormoneit, D., C. Lemieux, and D. J. Fleet (2001). Lattice particle filters. In *Proc. Uncertainty in Artificial Intelligence*, Seattle, pp. 395–402. Morgan Kaufmann.

Otte, M. and H. H. Nagel (1994). Optical flow estimation: Advances and comparisons. In J. Eklundh (Ed.), *European Conference on Computer Vision*, Stockholm, pp. 51–60. Springer-Verlag.

Potter, J. L. (1980). Scene segmentation using motion information. *IEEE Trans. on Systems, Man and Cybernetics 5*, 390–394.

Sawhney, H. S. and S. Ayer (1996). Compact representations of videos through dominant and multiple motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 18*(8), 814–831.

Schunck, B. G. (1989). Image flow segmentation and estimation by constraint line clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence 11*(10), 1010–1027.

Shi, J. and C. Tomasi (1994). Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.

Shulman, D. and J. Hervé (1989). Regularization of discontinuous flow fields. In *Proc. IEEE Workshop on Visual Motion*, Irvine, CA, pp. 81–85.

Sidenbladh, H., M. J. Black, and D. J. Fleet (2000). Stochastic tracking of 3D human figures using 2d image motion. In *Proc. European Conference on Computer Vision*, Volume II, pp. 702–718. Dublin. Springer-Verlag.

Simoncelli, E. P., W. T. Freeman, E. H. Adelson, and D. Heeger (1992). Shiftable multiscale transforms. *IEEE Trans. on Information Theory 38*(2), 587–607.

Sminchisescu, C. and B. Triggs (2001). Covariance scaled sampling for monocular 3d body tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Volume I, Kauai, pp. 447–454.

Spoerri, A. and S. Ullman (1987). The early detection of motion boundaries. In *IEEE International Conference on Computer Vision*, London, pp. 209–218.

Srinivasan, M., S. Zhang, M. Altwein, and J. Tautz (2000). Honeybee navigation: Nature and calibration of the odometer. *Science 287*(5454), 851–853.

Sun, H. J. and B. J. Frost (1998). Computation of different optical variables of looming objects in pigeon nucleus rotundus neurons. *Nature Neuroscience 1*, 296–303.

Thompson, W. B., K. M. Mutch, and V. A. Berzins (1985). Dynamic occlusion analysis in optical flow fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence 7*, 374–383.

Tomasi, C. and T. Kanade (1992). Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision 9*(2), 137–154.

Ullman, S. (1979). The interpretation of structure from motion. *Proc. Royal Society London B-203*, 405–426.

Vasconcelos, N. and A. Lippman (2001). Empirical Bayesian motion segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 23*(2), 217–221.

Warren, W. H. (1995). Self-motion: Visual perception and visual control. In *Handbook of Perception and Cognition*, Volume 5: Perception of Space and Motion. New York: Academic Press.

Weiss, Y. and E. H. Adelson (1996). A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, pp. 321–326.

West, M. (1992). Mixture Models, Monte Carlo, Bayesian Updating and Dynamic Models. *Computer Science and Statistics 24*, 325–333.