MIT Media Laboratory Vision and Modeling Technical Report #202.

Distributed Representation of Image Velocity

Eero P. Simoncelli

The Media Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

October 1992

Abstract

We describe a new form of representation of image velocities, which does not rely on vector fields. For each local spatio-temporal region of the input image, we desire a function over the space of velocities describing the presence of a given velocity in that region. This function may be interpreted as a probability distribution over velocity, although it is not necessary to do so. A primary advantage of this representation is that it is capable of representing more than one velocity at a given image location. A multi-modal distribution indicates the presence of multiple motions. Such situations occur frequently in natural scenes near occlusion boundaries, and in situations of transparency.

We develop an example of this type of representation through a series of modifications of current differential approaches to motion estimation. We define an angular version of the standard gradient constraint equation, and then extend this to represent multiple motions. The derivation is first done for one-dimensional signals and then extended to two dimensions.

We implement an efficient version of this distributed representation, in which the entire distribution may be interpolated from a sparse set of samples. We then demonstrate its use on simple synthetic examples containing occlusion boundaries and transparent surfaces.

Keywords: motion, optical flow, representation, distributions, sampling, interpolation, occlusion, transparency, multiple motions.

1 Introduction

Visual motion provides a rich source of information about the environment for both natural and machine visual systems. In both cases, it is generally assumed that the first stage of motion processing is the computation of the image *velocity field*: that is, the projection of the motion of points in the three-dimensional world onto the image plane. As an approximation to this, computer vision techniques typically compute an estimate of the motion field from the spatial and temporal variations of image brightness. This field of approximate velocities is known as the "optical flow".

Despite the fact that an enormous amount of research effort has been devoted to the problem of optical flow estimation, current techniques are notoriously error-prone. In particular, several commonly occurring conditions seem to cause trouble for most algorithms:

1. Velocities cannot be uniquely determined for regions containing no structure, or one-dimensional structure. The presence of noise causes many algorithms to be unstable in these regions.

2. Many approaches rely on an assumption of intensity conservation. This is frequently violated in real image sequences, due to changes in lighting or non-trivial reflectance functions.

3. Almost all approaches assume that a single velocity vector is sufficient to describe the motion at each point in space and time. This assumption is often violated in natural scenes. For example, in regions that are rotating or expanding (i.e., regions of high divergence or curl), at occlusion boundaries, and in the presence of transparent surfaces or highlights.

In light of the first problem, many authors have suggested that optical flow computations should be augmented by the computation of "confidence" measures [1, 18, 9, 2]. More recently, some authors have developed estimation-theoretic approaches that compute covariance matrices [25, 24, 26], which serve as a two-dimensional confidence measure.

In this paper, we will re-examine the measurement and representation of motion in the image plane. We argue that it is the vector-field representation that is the source of the difficulty. Given that scene motion at a point is often not adequately described by a single motion, the vector representation is a violation of what Marr referred to as the "principle of least commitment" [17]. As an alternative, we advocate *distributed* representations of motion, in which the encoding of image plane velocity is implicit. The reader should have in mind the computation of a probability distribution (over the space of velocities) for each patch of the image, although this interpretation is not necessary.

In previous work, we derived Gaussian probability distributions of optical flow based on the standard gradient constraint and a simple model of measurement and state noise [24]. These distributions are unimodal, and thus have a fundamental assumption of unique velocity description. In this paper, we will develop distributed representations that are no longer restricted to unimodality, thus allowing us to robustly represent multiple motions that occur near occlusion boundaries, in regions of strong divergence or curl, and in transparently moving imagery. This work is an extension of [23], in which we proposed a mechanism for computing multimodal velocity distributions for one-dimensional imagery. This generalized computational algorithm operates by first applying a set of spatio-temporally oriented linear filters, and squaring their outputs. These responses correspond to a sampled representation of local image spatio-temporal energy. These outputs are then linearly combined to produce a sampled distribution over the space of velocities. These two stages are illustrated in figure 1. We will demonstrate the use of this representation on a set of simple synthetic images.

2 Differential Measurements and Regression

We begin with a description of first order differential motion estimation. Since velocity is a differential quantity, it is not surprising that one approach to its computation is through derivative measurements. Many authors have used this approach or variants of this approach [15, 5, 8, 13, 16, 27]. For simplicity we will introduce the problem using one-dimensional signals, extending to two dimensions in the next section.

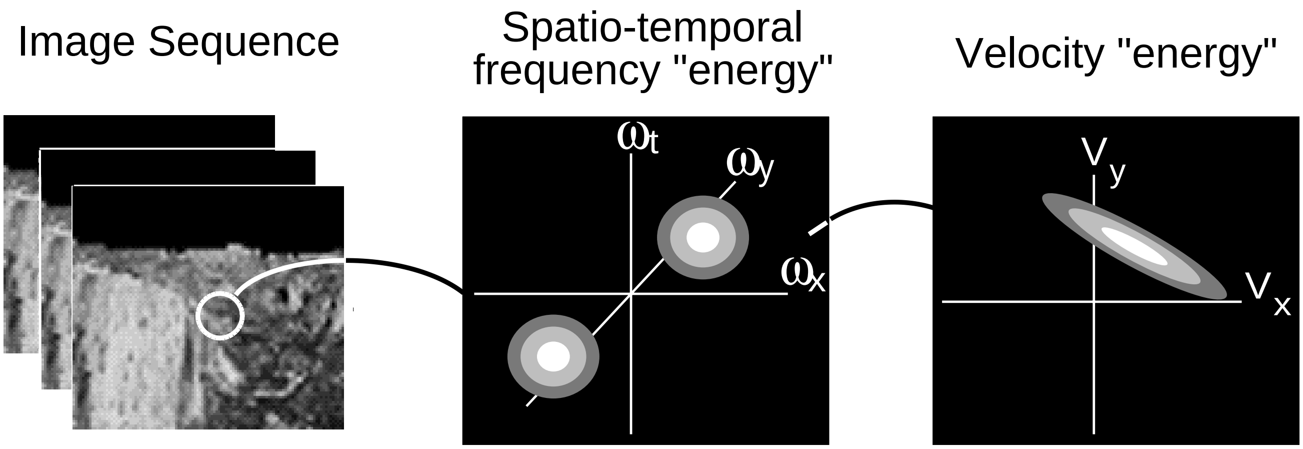

Figure 1: Two-stage algorithm for computing a distributed representation of motion. Depicted on the left is the original temporal sequence of images. In the center is a distribution over spatio-temporal frequency, corresponding to a small spatial patch of the input signal. On the right is a distribution over velocity. Each of these distributions may be interpolated from a small number of samples: we show the full distributions here.

Gradient Formulation

The standard gradient formulation of the optical flow problem is based on an assumption of intensity conservation over time. That is, changes in the image intensity are due only to translation of the local image intensity and not to changes in lighting, reflectance, etc. According to this assumption, the total derivative of the image intensity function should be zero at each position in the image and at every time. We write the image intensity signal as a continuous function of position and time: $f(x,t)$. Setting the total derivative of intensity with respect to time equal to zero gives the gradient constraint:

$$f_x(x,t)v(x,t) + f_t(x,t) = 0, \qquad (1)$$

where f_x and f_t are the spatial and temporal partial derivatives of the image f, and v is the instantaneous optical flow (at the position and time that the derivatives have been computed).

Equation (1) can only be solved for v at positions where the spatial derivative, f_x, is non-zero. Following the solution of Lucas and Kanade [16], we can avoid these singularities by writing a least-squares error function based on the combination of constraints from a small spatial patch:

$$E(v) \;=\; \sum_i w_i \left[\vec{f}_s(x_i,t)v + f_t(x_i,t)\right]^2 . \qquad (2)$$

In practice, the signal $f(x,t)$ is typically discretized, and thus the computation of derivatives involves (at least implicitly) an intermediate interpolation step with a continuous function $c(\vec{r})$. The derivative of the interpolated function must then be re-sampled at the points of the original sampling lattice. The sequence of operations may be written for a one-dimensional signal as follows:

$$\begin{aligned} \frac{df}{dx}(n) &\equiv \left[\frac{d}{dx}\left(\sum_m f(m)c(x-m)\right)\right](n) \\ &= \left[\sum_m f(m)\frac{dC}{dx}(n-m)\right] \\ &= f(n) * g_x(n), \end{aligned}$$

where we assume unit sample spacing in the discrete variables n and m for simplicity. Thus, the derivative operation may be computed in a single step as convolution with a filter, $g_x(n)$, which is the sampled spatial derivative of the continuous interpolation function $c(\vec{r})$. Furthermore, the interpolation function need not be a lowpass filter, but could act as a subband "prefilter", emphasizing some spatio-temporal frequencies at the expense of others.

Frequency Domain Regression

Additional insights about the estimation of motion may be gained by considering the problem in the Fourier domain. We represent a one-dimensional signal over time as a "space-time" intensity image, in which the intensity of each pixel corresponds to the value of the signal at a particular location and time. A uniformly translating one dimensional signal has the appearance of a striped pattern, where the stripes are oriented at an angle of $\alpha = \arctan v$. Clearly, the Fourier decomposition of this signal is a set of sinusoids of this same orientation, and varying wavenumber (spatial frequency magnitude). Thus, the Fourier transform will have power only on a line through the origin at angle $\alpha + \frac{\pi}{2}$. This is illustrated in figure 2.

This example suggests an alternative approach to the measurement of optical flow: to search for the line that best fits the power spectrum of the spatio-temporal signal. In

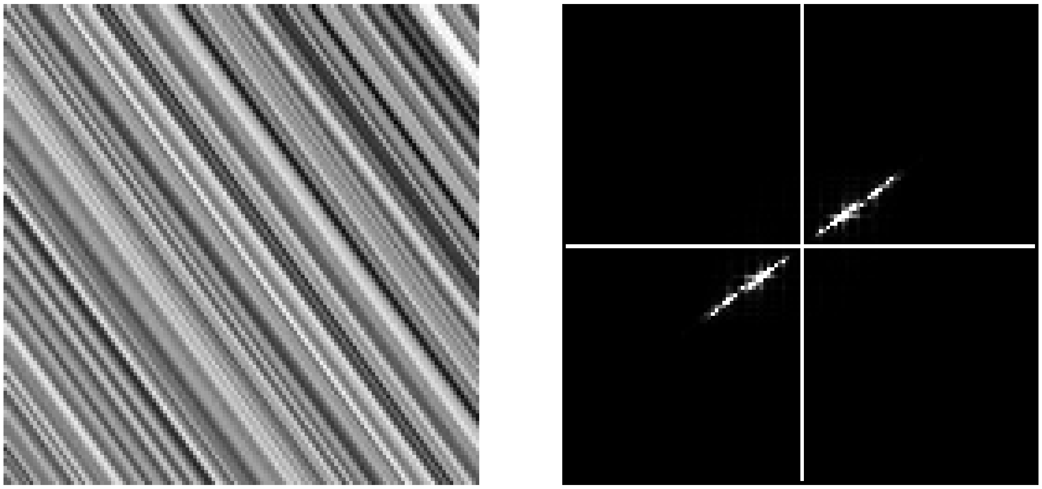

Figure 2: The Fourier spectrum of a translating pattern lies on a line in the frequency domain. On the left is a space-time intensity image of a translating fractal noise signal. On the right, its power spectrum, plotted over the ranges $\omega_x, \omega_t \in [-\pi, \pi]$.

practice, the entire image is seldom translating. One is therefore interested in a measure of the *local* power spectrum. This concept was used by Heeger [12] to develop a regression algorithm for the computation optical flow. He made local measurements of the power spectrum using a set of Gabor functions tuned for different spatio-temporal frequencies, and then computed a least-squares regression estimate to find the best-fitting plane to account for the measurements. Watson and Ahumada [28], Fleet and Jepson [9], and Gryzwacz and Yuille [11] have also used spatiotemporal filters to compute optical flow velocities.

A fundamental problem with each of these previous filtering approaches is that the velocity estimates depend on the local spatial content of the signal. In particular, each of these techniques gives the *wrong velocity* for moving sinusoidal gratings that are not matched to the optimal spatial frequency of the filters. This problem is a result of the choice of filters: a proper choice of filters allows an estimate of velocity that is unbiased by the spatial frequency of the signal.

Specifically, a set of filters that produce an unbiased estimate are the derivative filters discussed previously. In previous work, we have shown that the gradient approach may be viewed as a spatio-temporal regression analysis much like that described above [23, 22]. Consider the energy function given in equation (2) with the summation occurring over the entire image:

$$\begin{aligned} E(\vec{v}) &= \sum_{\vec{x}} \left| v g_x * f + g_t * f \right|^2 \\ &= \sum_{\vec{\omega}} \left| v G(\vec{\omega}) F(\vec{\omega}) \omega_x + G(\vec{\omega}) F(\vec{\omega}) \omega_t \right|^2 \\ &= \sum_{\vec{\omega}} \left[v\omega_x + \omega_t \right]^2 \cdot \left| G(\vec{\omega}) F(\vec{\omega}) \right|^2 \qquad (3) \end{aligned}$$

where the sum on the first line is over all image pixels and the sums on the latter two lines are over all spatio-temporal frequencies, $\vec{\omega}$ (we have used Parseval's rule to switch to the frequency domain). We have also used the fact that the Fourier transform of a directional derivative operator is a unit slope ramp function in the direction of the derivative.

The term in square brackets is the squared ω_t-distance between the point $\vec{\omega}$ and the line defined by $v\omega_x = -\omega_t$. This equation is a least-squares linear regression error function in v, weighted by the prefiltered image power spectrum, $|G(\vec{\omega})F(\vec{\omega})|^2$. It is easy to see that the error function will be zero at $v = \omega_t/\omega_x$ for a sinusoidal input with spatial and temporal frequencies (ω_x, ω_t). In addition to giving the correct solution for sinusoidal input patterns, the gradient filters have the added advantage that the solution is analytic. Previous spatio-temporal filtering solutions required numerical optimization.

Angular Regression

The regression equation (3) given above measures errors in the ω_t direction. Thus, all orientations in space-time are *not* treated equally. In order to extract the essential aspect of the constraint equation, we will be more interested in measuring *angular* errors.[1] To this end, we will rewrite the gradient constraint in the frequency domain with the angular and radial portions separated. First, we extract the angular portion of the velocity dependence:

$$\begin{aligned} E(\vec{v}) &= \left(\sqrt{v^2+1}\right)^2 \sum_{\vec{\omega}} \left[\hat{v} \cdot \vec{\omega} \right]^2 \left| G(\vec{\omega}) F(\vec{\omega}) \right|^2 \qquad (4) \\ &= \left(v^2 + 1 \right) \sum_{\vec{\omega}} \left[\mathcal{D}_{\hat{v}}(\vec{\omega}) \right]^2 |G(\vec{\omega}) F(\vec{\omega})|^2, \qquad (5) \end{aligned}$$

[1] Other regression error measures are also possible. For example, Shizawa and Mase [20] use the perpendicular distance to the line.

where $\hat{v}$ is the normalized angular velocity vector defined by $\hat{v} = (v,1)^T/\sqrt{v^2+1}$, and $\mathcal{D}_{\hat{v}}(\vec{\omega}) = \hat{v}\cdot\vec{\omega}$ is (the Fourier transform of) the directional derivative in the $\hat{v}$ direction. $\hat{v}$ is sometimes called a "steering" vector. The first parenthesized expression containing v does not depend on $\vec{\omega}$, and so has been removed from the sum. This expression provides a weighting on the space of v, giving preference to those v with small magnitude.

Now, we also separate the radial portion of the dependence on $\vec{\omega}$:

$$\begin{aligned} E(\vec{v}) &= \left(v^2+1\right)\sum_{\vec{\omega}} \left[\hat{v}\cdot\hat{\omega}\right]^2 |G'(\vec{\omega})F(\vec{\omega})|^2 && (6) \\ &= \left(v^2+1\right)\sum_{\vec{\omega}} \left[\mathcal{C}_{\hat{v}}(\vec{\omega})\right]^2 |G'(\vec{\omega})F(\vec{\omega})|^2, && (7) \end{aligned}$$

where $\hat{\omega} = \vec{\omega}/|\vec{\omega}|$, $G'(\vec{\omega}) = |\vec{\omega}|G(\vec{\omega})$, and $\mathcal{C}_{\hat{v}}(\vec{\omega}) = \mathcal{D}_{\hat{v}}(\vec{\omega})/|\vec{\omega}|$ is a *directional cosine* function: the cosine of the angle between the normalized vectors $\hat{v}$ and $\hat{\omega}$. The point of writing this is to separate the computation into three components:

1. $|G'(\vec{\omega})|^2$, a spatio-temporal prefilter (i.e., a weighting factor on the signal spectrum),

2. $\sum_{\vec{\omega}} \left[\mathcal{C}_{\hat{v}}(\vec{\omega})\right]^2$, a purely angular component, and

3. (v^2+1), a weighting factor in the *velocity* domain.

In the expression of equation (7), the second (angular) component provides the constraint that links the velocity to the derivative measurements. This constraint is at the heart of most differential algorithms. We will use this constraint as the basis for our distributed representation. To emphasize this, we define an *angular regression* function $A(\hat{v})$ by removing the velocity and spectral weighting factors:

$$A(\hat{v}) = \sum_{\vec{\omega}} \left[\mathcal{C}_{\hat{v}}(\vec{\omega})\right]^2 |F(\vec{\omega})|^2. \qquad (8)$$

3 Distributed Representation of Motion

In this section, we develop the concept of distributed velocity representation. The angular regression function in equation (8) is in fact a distributed representation of motion:

for each candidate v, it computes an error measure which tells us whether the velocity is consistent with a given set of directional derivative measurements.

The value of the function will be low for velocities that are well-supported. This is because the gradient algorithm operates by finding a directional derivative with *minimal* response. In the frequency domain, recall that the power spectrum of a translating pattern lies on a line. The directional derivative *perpendicular* to this line will have a zero response. Thus, the standard gradient algorithm may be termed a "null-steering" algorithm.

We wish to write the distribution such that it has a *maximum* at the correct velocity. In order to accomplish this, we note that the sum of the squared directional cosine in direction $\hat{v}$ and the squared directional cosine in the perpendicular direction is unity:

$$[\mathcal{C}_{\hat{v}}(\vec{\omega})]^2 + [\mathcal{C}_{\check{v}}(\vec{\omega})]^2 = \cos^2(\theta) + \sin^2(\theta) = 1,$$

where $\check{v}$ is the unit vector perpendicular to $\hat{v}$, and θ is the angle between $\hat{v}$ and $\hat{\omega}$. Thus, minimization of $[\mathcal{C}_{\hat{v}}(\vec{\omega})]^2$ is equivalent to maximization of $[\mathcal{C}_{\check{v}}(\vec{\omega})]^2$. Using this fact, we can write a "max-steering" angular error function as:

$$A^{+}(\hat{v}) = \sum_{\vec{\omega}} [\mathcal{C}_{\check{v}}(\vec{\omega})]^2 \, |F(\vec{\omega})|^2.$$

This function will respond to the *presence* of the line rather than the absence. [2]

Finally, we would like to convert back to a distribution on the space of velocities, v. If we are interested in using the distributions as probability density functions (i.e., if we will be integrating functions against them), then we must include a Jacobian weighting factor in the conversion:

$$\begin{aligned} \mathcal{P}(v) &= \left|\frac{\partial \check{v}}{\partial v}\right| A(\check{v}) \\ &= \frac{1}{v^2+1} \sum_{\vec{\omega}} [\mathcal{C}_{\check{v}}(\vec{\omega})]^2 \, |F(\vec{\omega})|^2. \end{aligned}$$

Note that in order to interpret these as probability distributions, we must also include a normalization factor of $\sum_{\vec{\omega}} |F(\vec{\omega})|^2$.

[2] There are other ways to accomplish this. For example, one could negate and exponentiate the expression to produce a Gaussian distribution as in [24].

Distribution Sampling and Interpolation

The previous equation gives a functional form for the distributed representation of velocity. In practice, one does not wish to compute and store the value of this function for at a large number of v values and for each point in space-time. In this section, we show that in fact, the distribution may be interpolated from the values taken at a few sample points.

Consider the directional cosine measurements:

$$\begin{aligned} \mathcal{C}_{\check{v}}(\vec{\omega})F(\vec{\omega}) &= \check{v}\cdot\hat{\omega}F(\vec{\omega}) \\ &= \check{v}_x\frac{\omega_x}{|\vec{\omega}|}F(\vec{\omega}) + \check{v}_t\frac{\omega_t}{|\vec{\omega}|}F(\vec{\omega}). \end{aligned}$$

This expression is a sum of two directional cosines:

$$\mathcal{C}_{\check{v}}(\vec{\omega})F(\vec{\omega}) = \check{v}_x\mathcal{C}_{\hat{e}_x}F(\vec{\omega}) + \check{v}_t\mathcal{C}_{\hat{e}_t}F(\vec{\omega}) \tag{9}$$

where $\hat{e}_x$ and $\hat{e}_t$ are the unit vectors along the ω_x and ω_t axes. In other words, the directional cosine in *any* direction may be computed as a linear combination of the directional cosines along the axes. Also, for a fixed set of directional cosine measurements, the distribution $\mathcal{P}(v)$ has the form of a squared cosine function. It is therefore always *unimodal.* This is illustrated in figure 3.

Freeman and Adelson have developed a theory of such functions, which they call "steerable" [10]. They describe a sampling theorem in orientation and derive the interpolation functions that are used to synthesize the response of a filter at a desired orientation from the responses at some fixed set of orientations. Others have also worked on the analysis of orientation using rotationally-invariant operators (eg., [14, 19]).

We can take the interpolation in equation (9) one step further, and compute the value of the distribution $\mathcal{P}(v)$ at any v from three precomputed measurements. Consider the square of the directional cosine:

$$\begin{aligned} \sum_{\vec{\omega}}\left[\mathcal{C}_{\check{v}}(\vec{\omega})F(\vec{\omega})\right]^2 &= \sum_{\vec{\omega}}\left|\left(\check{v}_x\omega_x + \check{v}_t\omega_t\right)F(\vec{\omega})\right|^2 \\ &= \check{v}_x^2\sum_{\vec{\omega}}\left[\omega_x^2|F(\vec{\omega})|^2\right] + 2\check{v}_x\check{v}_t\sum_{\vec{\omega}}\left[\omega_x\omega_t|F(\vec{\omega})|^2\right] + \check{v}_t^2\sum_{\vec{\omega}}\left[\omega_t^2|F(\vec{\omega})|^2\right] \end{aligned} \tag{10}$$

Thus $\mathcal{P}(v)$ may be computed as a linear combination of the three quadratic measurements corresponding to the three summations in the equation.

We can write this as a linear combination of samples of $\mathcal{P}(v)$ by solving three simultaneous linear equations:

$$\begin{pmatrix} \mathcal{P}(v_1) \\ \mathcal{P}(v_2) \\ \mathcal{P}(v_3) \end{pmatrix} = M \cdot \vec{Q}$$

where the $\vec{v}_i$ are three arbitrary but fixed choices of v, and

$$M = \begin{pmatrix} \check{v}_{1x}^2 & 2\check{v}_{1x}\check{v}_{1t} & \check{v}_{1t}^2 \\ \check{v}_{2x}^2 & 2\check{v}_{2x}\check{v}_{1t} & \check{v}_{2t}^2 \\ \check{v}_{3x}^2 & 2\check{v}_{3x}\check{v}_{1t} & \check{v}_{3t}^2 \end{pmatrix}$$

and $\vec{Q}$ is a vector of the three quadratic measurements from equation (10). Assuming the $\vec{v}_i$s are suitably chosen, we can invert the matrix M, and use it to solve for $\mathcal{P}(v)$:

$$\mathcal{P}(v) = \begin{pmatrix} \check{v}_x^2 \\ 2\check{v}_x\check{v}_t \\ \check{v}_t^2 \end{pmatrix}^T \cdot M^{-1} \cdot \begin{pmatrix} \mathcal{P}(v_1) \\ \mathcal{P}(v_2) \\ \mathcal{P}(v_3) \end{pmatrix}$$

That is, $\mathcal{P}(v)$ may be written as a *linear* combination of the three values of $\mathcal{P}(v_i)$. This sparse sampling of the v provides a complete representation of the function $\mathcal{P}(v)$.

Not only does this allow more efficient storage and computation of the distributions, it has a natural interpretation in terms of biological visual systems. One can postulate three tuned "units" that compute the values of the the three $\mathcal{P}(v_i)$. Later stages of the computation may access the value of $\mathcal{P}(v)$ for any v by simply computing a weighted sum of these three values.

Multi-modality: Higher Order Distributions

Nearly all previous techniques for analyzing motion attempt to compute a single motion at each point in space and time. But in naturally occurring scenes, there are often regions that are not adequately described in this way. The most common example is that of occlusion boundaries. In the neighborhood of such a boundary, there are two motions. Another common example is that of transparent surfaces. Since the operators used to compute velocity are of some finite size, the problem will arise whenever there are changes in the motion field that are abrupt compared to the size of the operators. Biological systems also provide inspiration for addressing this problem: humans can

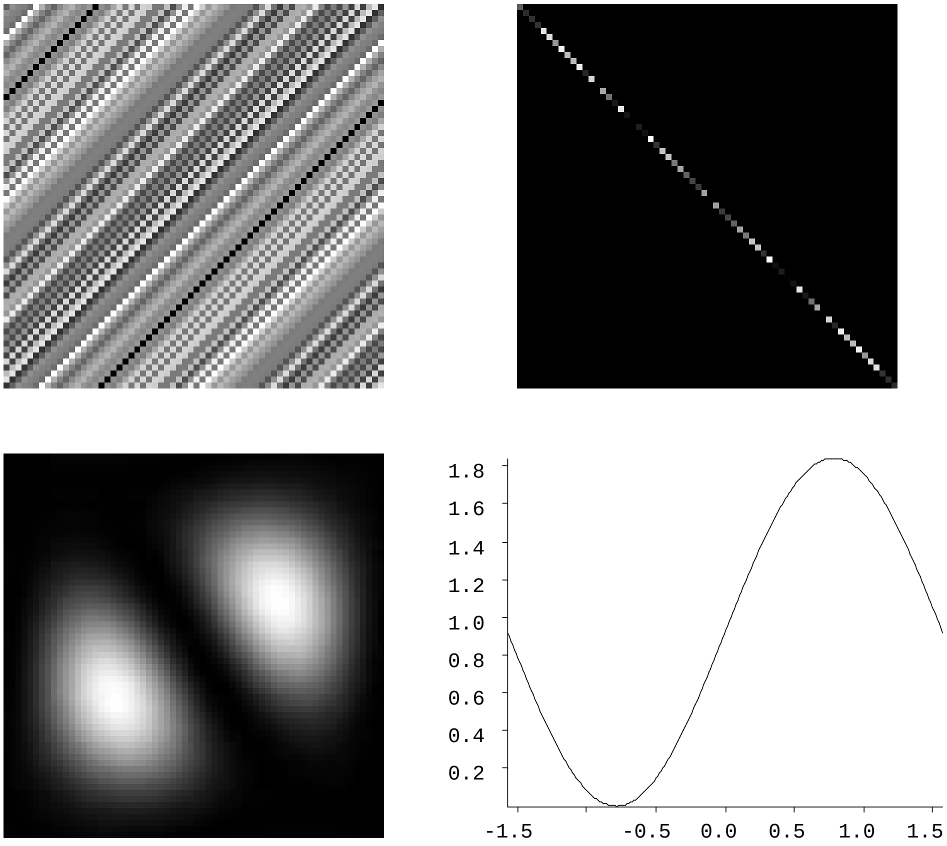

Figure 3: Illustration of the computation of the distribution over velocity space. On the top is a space-time translating one-dimensional noise signal, and its power spectrum, plotted over the range $[-\pi, \pi]$. Below is the power spectrum of a directional derivative filter at an arbitrary angle. On the right is the resulting distribution over $\alpha = \arctan(v)$. Conceptually, the distribution is computed by rotating the derivative filter through all angles α, and computing the inner product of the its power spectrum with that of the signal.

clearly “see” multiple motions at a point, and have no trouble distinguishing the motion of transparently moving sheets.

Some authors have tried to handle this by using higher-order expansions of the motion field (e.g., affine models [4, 7]). Shizawa and Mase [20, 21], and Bergen et. al. [3] have described algorithms for explicitly computing two motion vectors at each point in the scene.

We take a different approach here. In the previous section, we described the computation of a unimodal distribution function over the space of all velocities, v, for each point in space and time. We showed that the unimodality of the solution was a property of the use of first derivative (or directional cosine) filters: the space spanned by these particular linear operators is *only capable of representing a single velocity.*

In order to represent multiple motions, we need a set of filters that are more narrowly tuned in orientation. To this end, we can make use of higher order directional derivatives (or powers of directional cosines). In the frequency domain, we simply raise the directional cosine function to the Nth power:

$$\begin{aligned}
\mathcal{P}_N(v) &= \frac{1}{v^2+1}\sum_{\vec{\omega}}\left[\mathcal{C}_{\check{v}}^{N}(\vec{\omega})\right]^2 |F(\vec{\omega})|^2 \\
&= \frac{1}{v^2+1}\sum_{\vec{\omega}}\left[\check{v}\cdot\hat{\omega}\right]^2 N|F(\vec{\omega})|^2 \\
&= \frac{1}{v^2+1}\sum_{\vec{\omega}}\left|F(\vec{\omega})\sum_{n=0}^{N}\left(\tfrac{N!}{n!(N-n)!}\right)(\hat{v}_x)^n(\hat{v}_t)^{N-n}(\hat{\omega}_x)^n(\hat{\omega}_t)^{N-n}\right|^2 .
\end{aligned}$$

We have written the last expanded equation to emphasize two points. First, the distribution is computed from a set of linear directional cosine measurements. That is, the Nth directional cosine is computed through a linear combination of measurements. Each term in the sum corresponds to a measurement of a mixed x and t Nth-order derivative. Second, although the distribution is no longer quadratic in the components of $\hat{v}$, it is *still* quadratic in these linear measurements.

Many authors have argued that the use of higher-order derivatives for estimating motion leads to increased noise sensitivity. An Nth order x-derivative filter, for example, has a Fourier transform of ω_x^N and thus will strongly emphasize the high-frequency content of the signal, which is likely to be dominated by noise. But we have eliminated this effect by replacing directional derivative filters with directional *cosine* filters. The

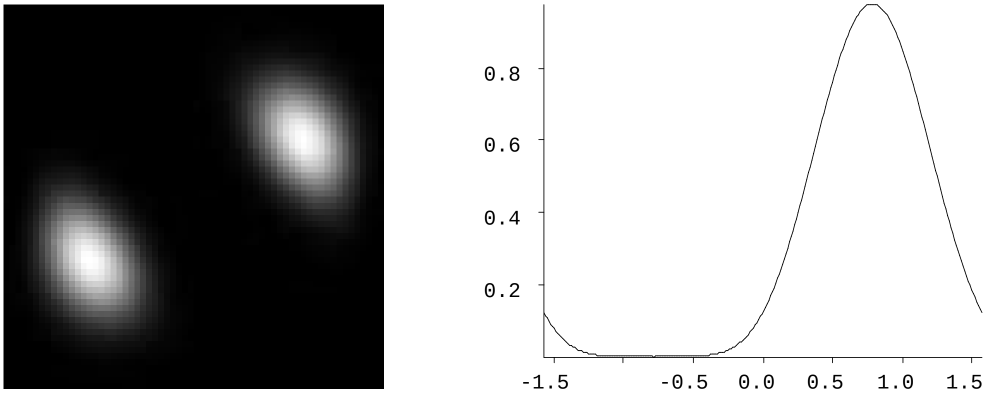

Figure 4: Illustration of the computation of the distribution over velocity space. The signal being analyzed is the same as the one in figure 3. On the left is the power spectrum of an example third directional derivative filter. Four such filters are used to analyze the image. On the right is the resulting distribution over $\alpha = \arctan(v)$.

spectrum of these filters is flat with respect to frequency magnitude: the importance of using higher-order filters is the *narrower orientation tuning* of the operators.

We must decide how high a derivative order to use. As is often the case in such questions, there is a tradeoff here. Lower order filters are more broadly tuned in orientation, but can generally be made smaller in space. For the examples given in the next section, we will use third derivatives. Figure 4 illustrates the application of a set of third order filters to the example shown previously in figure 3. Note that the resulting distribution is narrower than in the first derivative example. As an aside, we also mention here the importance of the conversion to a max-steering algorithm taken earlier in this section. A null-steering algorithm based on higher order derivatives would perform poorly, because the minimum is very broad.

Extensions to Two Dimensions: “Donut” Mechanisms

Extending the distributed representation to two dimensions is relatively simple. The Fourier spectrum of a translating two-dimensional pattern lies in a *plane* through the origin in the spatio-temporal frequency domain. Analogous to the one-dimensional case, the gradient constraint equation may be viewed as a planar regression function:

$$E(\vec{v}) = \sum_{\vec{v}} \left[v_x\omega_x + v_y\omega_y + \omega_t\right]^2 \cdot \left|G(\vec{\omega})F(\vec{\omega})\right|^2 .$$

As in one dimension, we can "angularize" this equation:

$$A(\hat{v}) = \sum_{\vec{\omega}} \left[\mathcal{C}_{\hat{v}}(\vec{\omega})\right]^2 |F(\vec{\omega})|^2,$$

where the steering vector is now defined as $\hat{v} = (v_x, v_y, 1)^T / \sqrt{|v|^2 + 1}$.

This is again a null-steering function: it has a minimum when the directional cosine function nulls the plane containing the spectrum of the moving image. To convert this expression into a max-steering expression, we must search for the *presence* of the spectral plane. As in the one-dimensional case, the sum of squares of the directional cosines along a set of three orthogonal axes is unity:

$$\left[\mathcal{C}_{\hat{v}}(\vec{\omega})\right]^2 + \left[\mathcal{C}_{\check{v}}(\vec{\omega})\right]^2 + \left[\mathcal{C}_{\check{\check{v}}}(\vec{\omega})\right]^2 = 1,$$

where we define (without loss of generality) $\check{v} = \hat{v} \times \hat{e}_x$, and $\check{\check{v}} = \hat{v} \times \check{v}$. Thus the squared directional cosine in the direction of $\hat{v}$ is just one minus the sum of the squared directional cosines in the $\check{v}$ and $\check{\check{v}}$ directions. Analogous to the one-dimensional case, we define a max-steering function as follows:

$$A^{+}(\hat{v}) = \sum_{\vec{\omega}} \left|\mathcal{C}_{\check{v}}(\vec{\omega}) F(\vec{\omega})\right|^2 + \sum_{\vec{\omega}} \left|\mathcal{C}_{\check{\check{v}}}(\vec{\omega}) F(\vec{\omega})\right|^2$$

That is, the value is computed as a sum of squared responses of two directional cosines lying in the plane perpendicular to the normalized candidate velocity vector. We emphasize that *this does not imply that we must measure all possible directional cosines.* These may be interpolated from a small set of fixed measurements. This sum of the two filters in the plane will form a smooth ring or "donut", bisected by the plane; thus we call it a "donut mechanism". This construction is illustrated in figure 5.

The maximal-steering version of the gradient algorithm may now be extended to higher order derivatives by raising the directional cosines to the Nth power. One complication arises in the two-dimensional case: if we simply raise our two directional cosines to the Nth power, they will no longer cover the plane evenly. In fact, we require a set of $(N+1)$ Nth derivatives to cover the plane uniformly. We define a set of equally-spaced directions as:

$$\hat{v}_i = \cos\left(\frac{2\pi i}{N+1}\right) \check{v} + \sin\left(\frac{2\pi i}{N+1}\right) \check{\check{v}}.$$

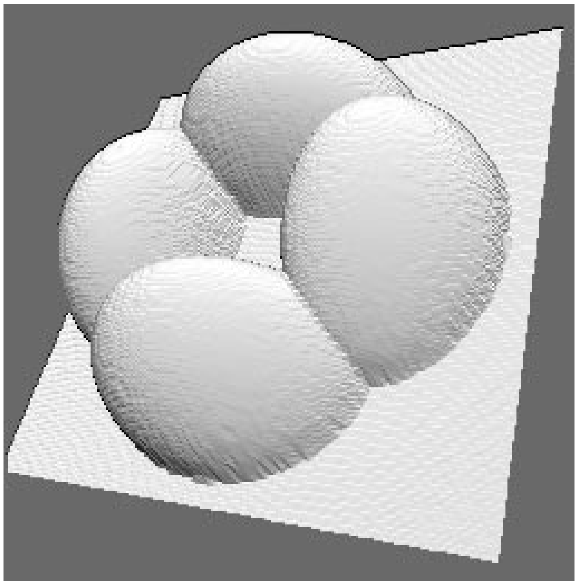

Figure 5: Illustration of the gradient algorithm as a “max-steering” solution. The “velocity energy” surface is computed by adding up the power response of two directional derivatives lying in the spatio-temporal frequency plane corresponding to a given velocity. Illustrated is an instance of such a plane, and idealized level surfaces of the power spectra of two such derivative filters. Note that the level surfaces of the sum of the two will form a smooth ring or “donut”, bisected by the plane.

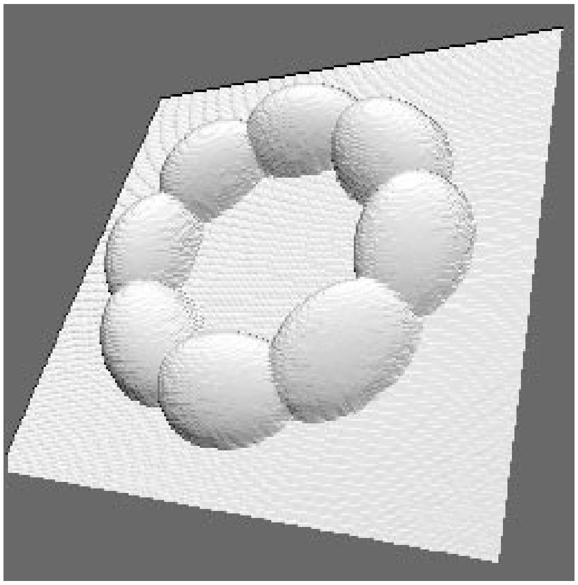

Figure 6: Illustration of the "donut" mechanism, based on third derivatives. A ring of third derivatives lying on the plane corresponding to a particular velocity are used to measure evidence for the presence of that velocity. These directional derivatives are computed efficiently via interpolation from a fixed set of derivatives. Note also that a level surface of the sum of power spectra of these filters produces a smooth torus: they are depicted here as spherical to indicate their locations.

The generalized error function now looks like:

$$\begin{aligned}\mathcal{P}_N(\vec{v}) &= \left|\frac{\partial\check{v}}{\partial\vec{v}}\right| \sum_{\vec{\omega}} \sum_{i=0}^{N} \left[\mathcal{C}_{\check{v}_i}\right]^2 \left|F(\vec{\omega})\right|^2 \\ &= \frac{1}{\sqrt{|\vec{v}|^2+1}^{3}} \sum_{\vec{\omega}} \left|F(\vec{\omega})\right|^2 \sum_{i=0}^{N} \left[\mathcal{C}_{\check{v}_i}\right]^2, \qquad (11)\end{aligned}$$

where we have included the Jacobian weighting factor. This is a sum of squares of Nth directional cosines lying in the plane corresponding to the vector $\vec{v}$. This construction is illustrated in figure 6, for a set of third derivative filters.

4 Examples

We implemented a set of third derivative filters, based on a Gaussian prefilter. The advantage of Gaussians is that they are both circularly symmetric and separable. Thus, they don't introduce angular biases, and the convolution operations may be performed efficiently. The filters were applied to a set of synthetic imagery and the outputs used to construct distributed representations of motion. Third directional derivative measurements (or directional cosines raised to the third power) may be interpolated from the set of ten separable third derivative measurements. The squares of these may thus be interpolated from a set of the 55 possible quadratic combination terms.

Figure 7 illustrates the behavior of the distributed mechanism in three prototypical singularity situations. The input signal is a moving square (white on a black background). Near the corners, there is sufficient local information to completely constrain the two-dimensional velocity. The response of the mechanism is a fairly localized peak of activity in $\vec{v}$ space. The mean of the distribution is at the location of the correct velocity, but this is not generally true of the peak of the distribution. Note that if we were to leave out the Jacobian factor in equation (11), then the peak *would* be at the correct velocity.

On the sides of the square, there is a one-dimensional singularity: the motion along the boundary cannot be determined using purely local measurements (this is known as "the aperture problem"). Thus, the resulting velocity distribution is elongated in the direction of the edge. In the center of the square, there is *no* intensity variation and so there is a full two-dimensional singularity. Here, the distribution is flat.

Figure 8 illustrates the behavior of the mechanism near an occlusion boundary. The input signal consists of two sheets of white noise, drifting in opposite directions, with the left one occluding the right one. The occlusion boundary is in the center of the image. The velocity distribution near the occlusion boundary is bimodal.

Figure 9 shows the behavior of the mechanism in the presence of additively transparent surfaces. Two fractal noise patterns moving in different directions are additively superposed. Again the velocity distribution is bimodal.

Finally, we show the response of the mechanism to transparently moving random dots in figure 10. Again, the response is bimodal.

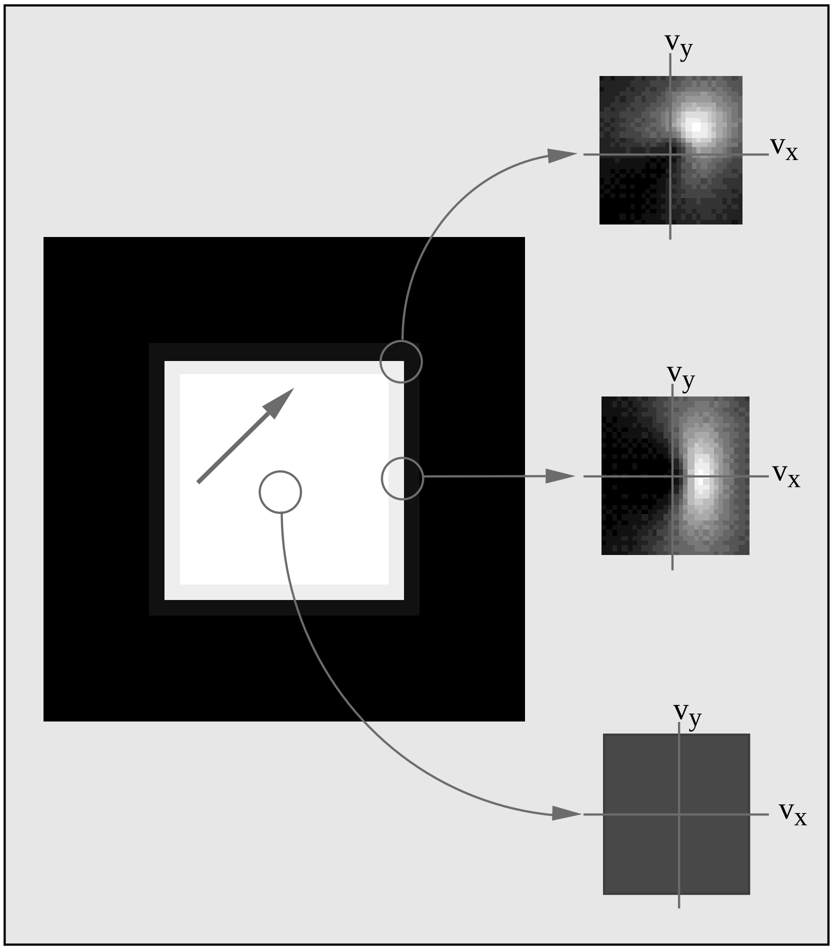

Figure 7: Response of the mechanism in several regions of a moving square sequence. In the corner, the velocity is well defined, and the distributed response is a well-localized "lump". On the side, the velocity is only constrained in the direction normal to the edge, and the distributed response is a ridge. In the center, the velocity is completely unconstrained, and the distributed response is flat.

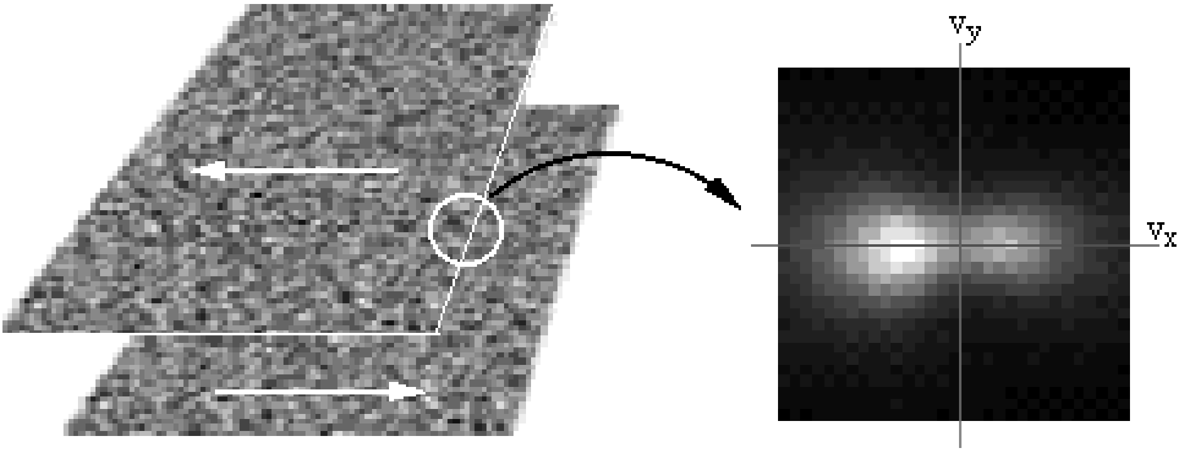

Figure 8: Response of the mechanism near an occlusion boundary. On the left is an illustration of the synthetic image sequence. The sequence consists of two white noise patterns, displayed on the right and left sides of the image, such that the left one occludes the right one. The patterns move in opposite directions at a speed of one pixel/frame. The white line separating the two patterns is for figure clarity and is not part of the image sequence. On the right is the bimodal response of the distributed third derivative mechanism located at the occlusion boundary.

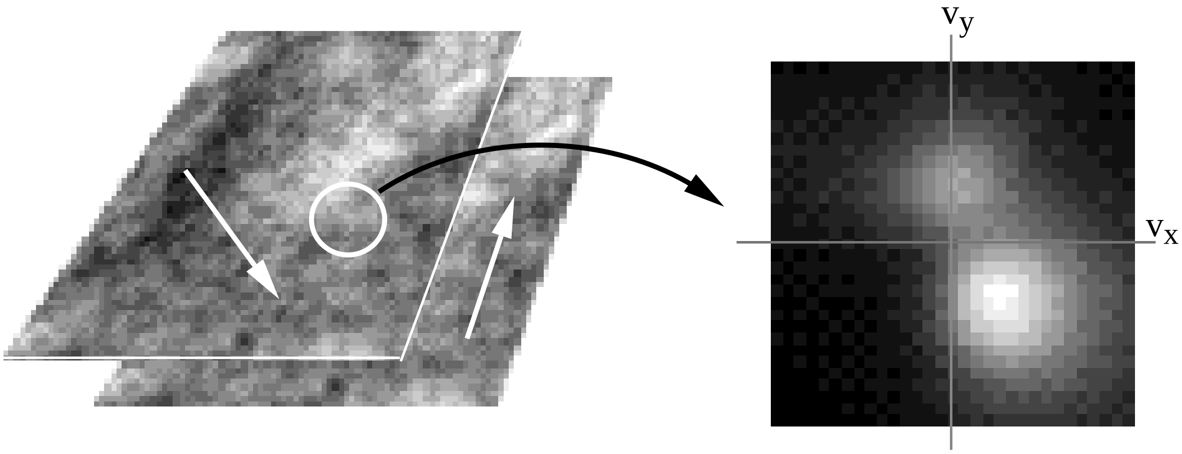

Figure 9: Response of the mechanism in the presence of additive transparency. On the left is an illustration of the synthetic image sequence, which consists of two additively combined fractal noise patterns moving in different directions (one upward, the other down and to the right). The white line separating the two patterns is for figure clarity and is not part of the image sequence. On the right is the bimodal response of the distributed third derivative mechanism located in the center of the image.

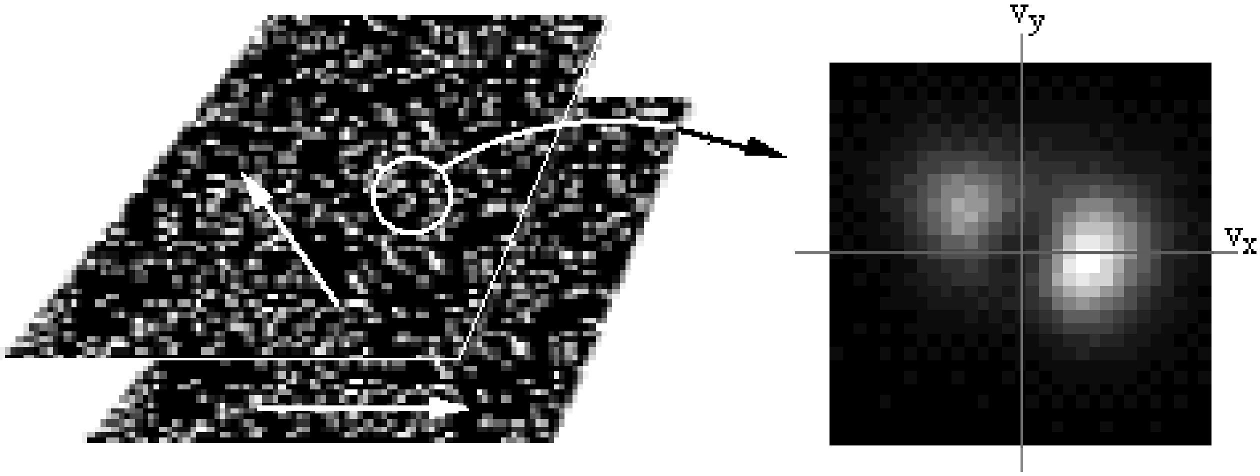

Figure 10: Response of the mechanism to interspersed moving dot patterns. transparency. On the left is an illustration of the synthetic image sequence, which consists of two random dot patterns that have been combined using a logical "or" operation. The white line separating the two patterns is for figure clarity and is not part of the image sequence. On the right is the bimodal response of the distributed third derivative mechanism located in the center of the image.

5 Discussion

We have developed a distributed representation of image motion that is capable of locally analyzing regions of a scene containing multiple velocities. We derived a general mechanism for computing these distributions based on the angular portion of the gradient constraint equation. The computation is based on a set of higher-order directional cosine measurements. The mechanism may be implemented efficiently by interpolation from a sparse set of samples: these interpolation functions are computed analytically.

The derivations in this paper are based on an angular version of the differential motion constraint equation. This angular formulation lends itself more easily to the multi-modal extensions proposed, and simplifies the discussion of interpolation. We do *not* claim that the angular error function is actually preferable to the standard temporal-frequency error function, although it may prove to be so. Regardless, the methods presented in this paper may be applied (although not as easily) to the standard velocity constraint equation. This work will be described in future publications.

The concepts presented in this paper suggest many interesting directions for research.

There are numerous variants of the general distributed representation framework presented here. Different choices of the prefilter and of the velocity weighting terms in equation (7) will produce distributions with different characteristics. The choice of prefilter should be based partly on the noise properties of the measurements and knowledge of the input signal spectrum (eg., the prefilter could be chosen as a Wiener filter). The velocity weighting may be interpreted as a sort of "prior" probability, giving preference to some velocities that are considered to be more likely (eg., smaller speeds).

More importantly, these distributed representations may be incorporated into practical problems in image processing and computer vision. One simple example is that of frame interpolation (or prediction): if we interpret the distributions as probability distributions, we can use these to generate an estimate of the expected content of an intermediate frame, given the surrounding frames. Simple flow-based algorithms would perform poorly at this task in the presence of multiple motions. We expect that the distributed representation should also prove useful for segmenting or grouping scenes according to coherency of motion [6].

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. Technical Report RPL-TR-9107, Robotics and Perception Laboratory Technical Report, Queen's University, Kingston, Ontario, July 1992.

[3] J. R. Bergen, P. J. Burt, K. Hanna, R. Hingorani, P. Jeanne, and S. Peleg. Dynamic multiple-motion computation. In Y. A. Feldman and A. Bruckstein, editors, *Artificial Intelligence and Computer Vision*, pages 147–156. Elsevier Science Publishers B.V., 1991.

[4] R. W. Brockett. Gramians, generalized inverses, and the least-squares approximation of optical flow. *J. Vis. Comm. and Image Rep.*, 1(1):3–11, Septemeber 1990.

[5] C. Cafforio and F. Rocca. Methods for measuring small displacements of television images. *IEEE Trans. Info. Theory*, IT-22:573–579, September 1976.

[6] T. Darrell and A. P. Pentland. Robust estimation of a multi-layer motion representation. In *Proceedings IEEE Workshop on Visual Motion*, Princeton, October 1991.

[7] R. Eagleson. Measurement of the 2D affine lie group parameters for visual motion analysis. *J. Spatial Vision*, 1992.

[8] C. L. Fennema and W. Thompson. Velocity determination in scenes containing several moving objects. In *CGIP-9*, pages 301–315, 1979.

[9] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *Intl. J. Comp. Vis.*, 5(1):77–104, 1990.

[10] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Pat. Anal. Mach. Intell.*, 13(9):891–906, 1991.

[11] N. M. Grzywacz and A. L. Yuille. A model for the estimate of local image velocity by cells in the visual cortex. *Proc. R. Soc. Lond. A*, 239:129–161, 1990.

[12] D. J. Heeger. Model for the extraction of image flow. *J. Opt. Soc. Am. A*, 4(8):1455–1471, August 1987.

[13] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[14] H. Knutsson and G. H. Granlund. Texture analysis using two-dimensional quadrature filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pages 206–213, 1983.

[15] J. O. Limb and J. A. Murphy. Estimating the velocity of moving images in television signals. *Computer Graphics and Image Processing*, 4:311–327, December 1975.

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, 1981.

[17] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman and Company, San Fransisco, 1982.

[18] H. H. Nagel. Displacement vectors derived from second order intensity variations in image sequences. *Computer Vision, Pattern Recognition, and Image Processing*, 21:85–117, 1983.

[19] P. Perona. Deformable kernels for early vision. In *IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, pages 222–227, Maui, 1991.

[20] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Atlantic City, June 1990.

[21] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, June 1991.

[22] E. P. Simoncelli and E. H. Adelson. Computation of optical flow: Relationship between several standard techniques. Vision and Modeling Technical Report 165, MIT Media Laboratory, November 1990.

[23] E. P. Simoncelli and E. H. Adelson. Optical flow distributions: gradient, energy and regression methods. In *Optical Society of America, Annual Meeting*, Boston, MA, November 1990.

[24] E. P. Simoncelli and E. H. Adelson. Probability distributions of optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, June 1991.

[25] E. P. Simoncelli, D. J. Heeger, and E. H. Adelson. Perception of 3D motion in the presence of uncertainty. In *Investigative Opthalmology and Visual Science Supplement (ARVO)*, volume 31, 1990.

[26] A. Singh. Incremental estimation of image-flow using a kalman filter. In *Proceedings IEEE Workshop on Visual Motion*, Princeton, October 1991.

[27] A. Verri, F. Girosi, and V. Torre. Differential techniques for optical flow. *J. Opt. Soc. Am. A*, 7:912–922, 1990.

[28] A. B. Watson and A. J. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Am. A*, 2:322–342, 1985.