Distributed Representation and Analysis of Visual Motion

by

Eero P. Simoncelli

Vision and Modeling Group Technical Report #209

January 1993[†]

MIT Media Laboratory, E15-385
20 Ames Street
Cambridge, MA 02139

[†]Reprinted with smaller type and spacing, August 1993.

Distributed Representation and Analysis of Visual Motion

by

Eero Peter Simoncelli

B.A., Harvard University (1984)
M.S., Massachusetts Institute of Technology (1988)

Submitted to the
Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

January 1993

Author ______________________________
Department of Electrical Engineering and Computer Science
15 January 1993

Certified by ______________________________
Edward H. Adelson
Associate Professor, MIT Media Laboratory
Thesis Supervisor

Accepted by ______________________________
Campbell L. Searle
Chair, Departmental Committee on Graduate Students

Distributed Representation and Analysis of Visual Motion

by

Eero Peter Simoncelli

Submitted to the Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
on 15 January 1993, in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

Abstract

This thesis describes some new approaches to the representation and analysis of visual motion, as perceived by a biological or machine visual system. We begin by discussing the computation of image motion fields, the projection of motion in the three-dimensional world onto the two-dimensional image plane. This computation is notoriously difficult, and there are a wide variety of approaches that have been developed for use in image processing, machine vision, and biological modeling. We show that a large number of the basic techniques are quite similar in nature, differing primarily in conceptual motivation, and that they each fail to handle a set of situations that occur commonly in natural scenery.

The central theme of the thesis is that the failure of these algorithms is due primarily to the use of vector fields as a *representation* for visual motion. We argue that the translational vector field representation is inherently impoverished and error-prone. Furthermore, there is evidence that a direct optical flow representation scheme is not used by biological systems for motion analysis. Instead, we advocate *distributed* representations of motion, in which the encoding of image plane velocity is implicit.

As a simple example of this idea, and in consideration of the errors in the flow vectors, we re-cast the traditional optical flow problem as a probabilistic one, modeling the measurement and constraint errors as random variables. The resulting framework produces *probability distributions* of optical flow, allowing proper handling of the uncertainties inherent in the optical flow computation, and facilitating the combination with information from other sources. We demonstrate the advantages of this probabilistic approach on a set of examples. In order to overcome the temporal aliasing commonly found in time-sampled imagery (eg, video), we develop a probabilistic "coarse-to-fine" algorithm that functions much like a Kalman filter over scale. We implement an efficient version of this algorithm and show its success in computing Gaussian distributions of optical flow for both synthetic and real image sequences.

We then extend the notion of distributed representation to a generalized framework that is capable of representing multiple motions at a point. We develop an example representation through a series of modifications of the differential approach to optical flow estimation. We show that this example is capable of representing multiple motions at a single image location and we demonstrate its use near occlusion boundaries and on simple synthetic examples containing transparent objects.

Finally, we show that these distributed representation are effective as models for biological motion representation. We show qualitative comparisons of stages of the algorithm with neurons found in mammalian visual systems, suggesting experiments to test the validity of the model. We demonstrate that such a model can account quantitatively for a set of psychophysical data on the perception of moving sinusoidal plaid patterns.

Thesis Supervisor: Edward H. Adelson
Title: Associate Professor, MIT Media Laboratory

Acknowledgments

My greatest thanks go to two colleagues who have consistently supported me (scientifically, emotionally, and financially!) throughout my years as a graduate student. The first is my advisor Ted Adelson, whose uncanny insights are a continual source of surprise and delight. The second is David Heeger, whose clarity of thought and judgement I will always admire. Thanks to Alan Willsky and Berthold Horn, my two other thesis committee members, for their advice and encouragement. Other faculty members in the MIT EECS department have also been inspirational, especially Al Oppenheim, Greg Wornell, and Lou Braida.

The Perceptual Computing (formerly Vision and Modeling) Group at the MIT Media Lab is filled with unusual assortment of characters: I'll miss you all! Thanks to Sandy Pentland, Aaron Bobick and Roz Picard for great discussions, and for creating a cooperative and productive work environment. Special thanks go to my office-mate and friend, Bill Freeman, who was always happy to listen to my latest off-the-wall idea, and who taught me about steerable filters. Thanks also to Trevor Darrell, for our collaboration on motion segmentation (and for fixing the printers!), and to John Wang for helpful discussions about motion, and for the rendering software used to generate some of the figures in this thesis. Our support staff also deserve special mention: Laureen, Bea, and most recently, Judy, have been a tremendous help every step of the way.

I've also been fortunate to have supporters outside of MIT. Thanks to Beau Watson for his encouragement (and cooking lessons), and Tony Movshon for discussions about physiology (and proper choice of computer screen colors). Many mentors from my past have had an indirect but important influence on this work: Tom McMahon, Paul Rozin, John Hopfield, Alan Gelperin, Jim Stellar, Randy Gallistel, and Bob Stauffer.

Last but not least is my non-scientific support team! Many thanks to my friends in the area, especially Lisa & Steve, and Philip. Thanks also to David Heeger's wife, Anne, for her hospitality and support during my many visits. All of my present and past roommates at 17 Dimick Street have been invaluable in balancing my life in Boston, particularly Iris and Josh. I wouldn't have made it through the past year without my Senegalese drum teacher, Ibrahima. Finally I want to thank my father, my brother, my two sisters, and my mother's parents for their constant faith, love, and encouragement.

The majority of this research was performed at the MIT Media Lab, and was supported in part by the National Science Foundation (IRI 871-939-4), Goldstar Company, Inc (#73880) and the Television of Tommorrow Consortium (#73064). A portion of this research was supported by the NASA Ames Research Center (NASA RTOP 506-71-51). The opinions expressed herein are those of the author, and do not necessarily represent those of the sponsors.

Dedicated to the memory of my mother,

Stephanie Maria Tassia Simoncelli,
1937-1984

Contents

List of Figures

Chapter 1

Introduction

Vision is arguably the most important of our senses. Visual images and the dynamic evolution of those images provide enormous amounts of information about our surrounding environment. By observing changes in a scene over time, we can discover the three-dimensional structure of the scene, make predictions about collisions, and infer material properties of objects, such as their stiffness and transparency. Much of this information is revealed by the *motion* of the different parts of the scene. In this thesis, we will explore the problem of extracting, representing and analyzing the motion in visual scenes.

Why should we be interested in the study of visual motion? Aside from basic scientific curiosity, there are two primary motivating forces. Because of the richness of motion as an information source, analysis of visual motion is essential for many *practical applications.* These range from image-processing problems such as efficient coding or enhancement of motion pictures, to passive machine vision problems such as determining the shape of a moving object or recovering the motion of the camera relative to the scene, to active perception applications in which an autonomous agent (i.e., a robot) must explore its environment.

As a second motivation, we would like to understand the motion processing performed by biological visual systems. Most biological organisms inhabit an ever-changing environment. Sensing, processing and acting upon these changes is often essential for survival. There is an abundance of evidence that biological visual systems – even primitive ones – devote considerable resources to the processing of motion. Although the processing constraints in a biological system are somewhat different than those in an artificial vision system, each must extract motion information from the same type of intensity signals.

The past few decades have seen an increase in the conceptual overlap and collaboration between the biological and the machine vision communities. I believe there is a potential for even greater symbiosis between these fields. Computational approaches to vision can be

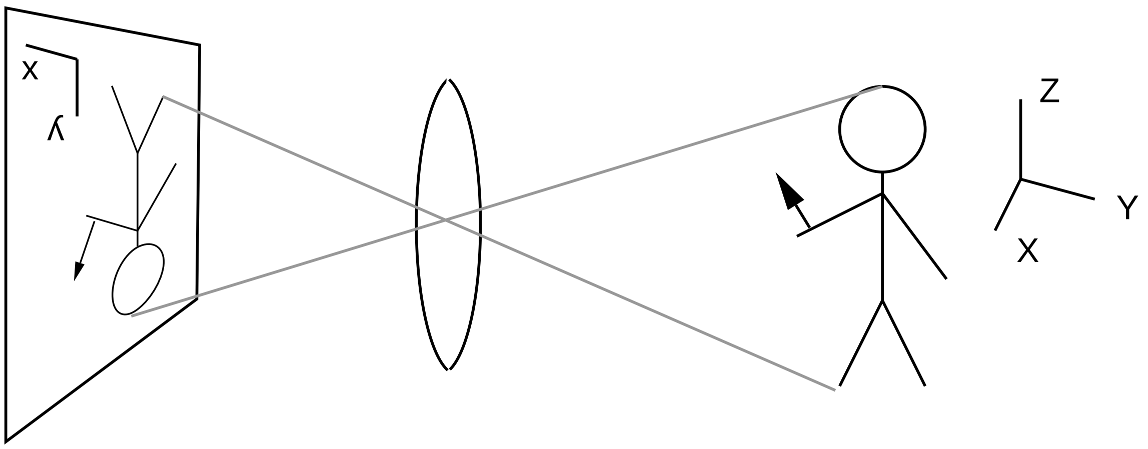


Figure 1-1: Illustration of the projection of motion in the three-dimensional world onto a two-dimensional sensor (or array of sensors).

inspired by our knowledge of biological systems, and in return can provide potential models for those same systems. I will try to show that although machine and biological solutions for motion analysis may be based on very different implementations, they may share the same basic representations and computational algorithms.

1.1 Visual Motion

What do we mean by the phrase "visual motion"? Images are formed as projections of the three-dimensional world onto a two-dimensional light-sensing surface. This surface could be, for example, a piece of film, an array of light sensors in a television camera, or the photoreceptors in the back of a human eye. The brightness of the image at each point indicates how much light fell on the surface at that spatial position at a particular time (or over some interval of time). When an object in the world moves relative to this projection surface, the two-dimensional projection of that object moves within the image. The movement of the projection of each point in the world is referred to as the image velocity or the *motion field.* This is illustrated in figure 1-1

For machine vision tasks, the sensing device is usually a motion picture or video camera. The output of these devices is a *series* of images. Each of these images, also known as "frames", is computed by integrating the light hitting the sensor sheet over a small interval of time. The resulting sequence of images forms what we commonly think of as a "movie", as shown in figure 1-2.

The estimation of the image motion field, is generally assumed to be the first goal of motion processing in machine vision systems. There is also evidence that this sort of computation is performed by biological systems. As an approximation to this, computer vision techniques

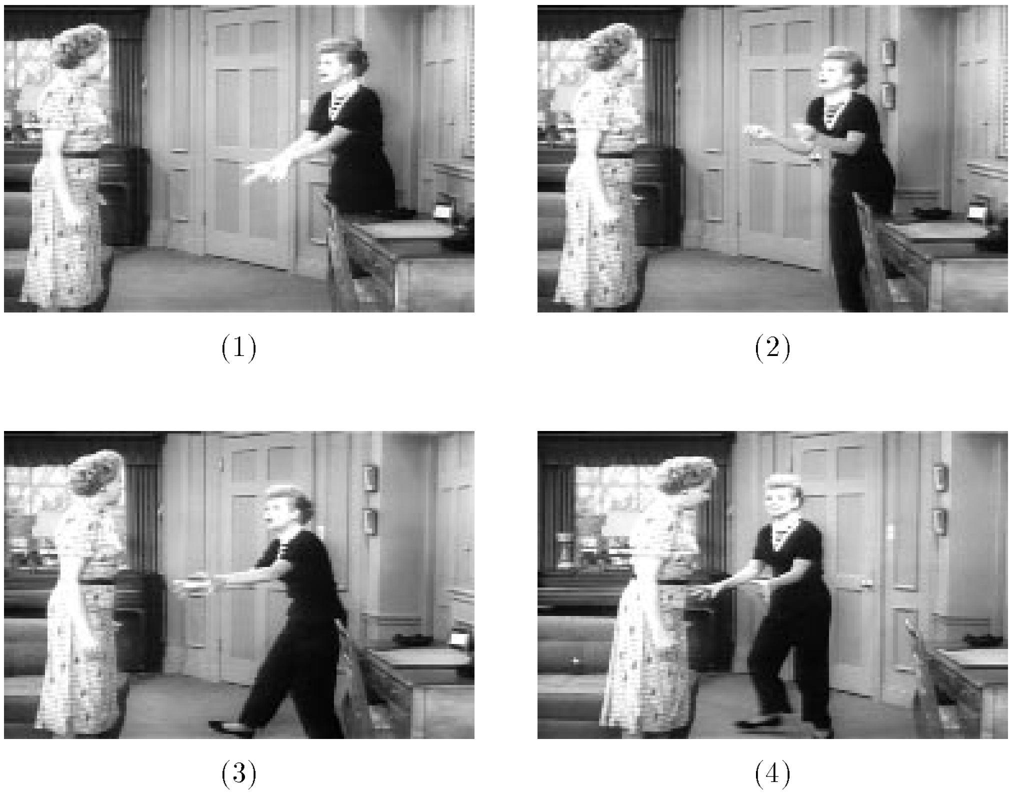


Figure 1-2: Four frames from the "I Love Lucy" show. During this particular sequence, Lucy moves to the left, raising her arms, Ethel moves to the right, and the entire background moves slowly to the right (the camera is panning). Each small patch of the scene has its own motion.

typically compute an estimate of the motion field known as the “optical flow”. The idea is to measure the apparent motion of local regions of the *image brightness pattern* from one frame to the next. In doing this, one is assuming that these intensity patterns are preserved from frame to frame. As many authors have shown, the optical flow is *not* always the same as the motion field (eg, [46, 97]).

What do we mean by the phrase “small pieces of image intensity pattern”? We cannot ask about the motion of an isolated point from one frame to the next without considering the context surrounding it. That is, we can only recognize the motion of *patterns* of intensity. On the other hand, if we ask about the motion of the entire image, we also can not answer the question, because in general the entire image does not move as a single entity. In the example in figure 1-2, the motion of each of the characters (in fact, each of the *limbs* of each of the characters) is different, and the background moves (relative to the camera) in yet another way.

The reader may then wonder why we do not try to extract the motion of each object separately. Many authors have worked on this problem and related problems, but it is a *very difficult* computational task to divide the scene into its component objects. The computational definition of an “object” is the source of the problem. We are interested in solutions that do *not* rely on a symbolic interpretation of the scene. Furthermore, such an algorithm will not be able to handle non-rigid types of motion. For the purposes of this thesis, we will concentrate on the basic problem of describing the motion of small patches. We view this as a building block, leading in a bottom-up fashion to more complex analyses of motion.

1.2 Basic Observations

Despite the fact that an enormous amount of research effort has been spent on the problem, estimates of image velocity fields are notoriously error-prone. In particular, several situations that occur commonly in natural scenes cause trouble for most optical flow algorithms. We will give an intuitive explanation of some of these problems here; in later chapters we will give more precise mathematical descriptions.

Underconstrained Regions

Often there is not enough information in a region to determine the optical flow. For example, if we are looking at an untextured flat surface, we cannot determine how (or even whether) it is moving. Regardless of the motion of the surface, the *image* remains constant over time. Since we are considering techniques that only estimate the motion of small regions, this problem will occur whenever the image brightness pattern is uniform in the region being considered. We will refer to this as the “blank wall” problem.

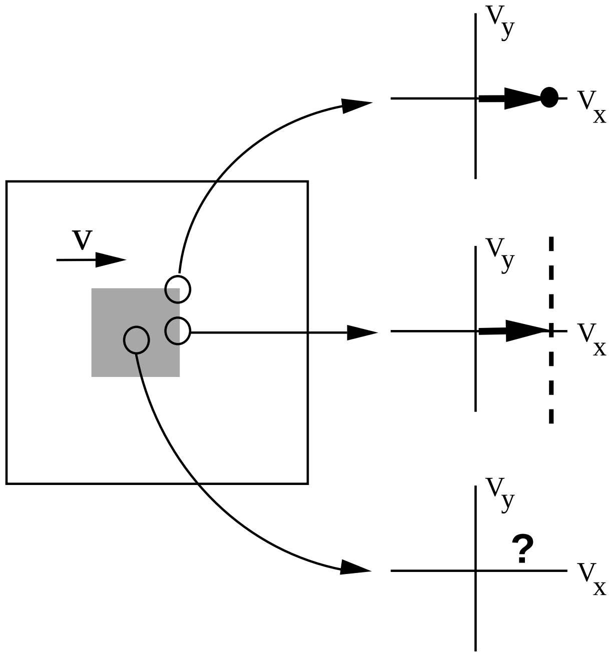


Figure 1-3: Conceptual illustration of motion situations producing the two type of singularities and the non-singular situation. At the corners of a moving square, a unique velocity describes the observed changes in intensity. On the sides, the intensity changes are consistent with a one-dimensional set of velocities: we can determine the motion perpendicular to the edge, but *not* the motion along the edge. This set of consistent velocities corresponds to the dashed line. In the center of the square, the motion is completely unconstrained.

Similarly, if we are looking at a pattern composed of stripes, then we cannot determine how fast (or whether) it is moving along the direction of the stripes. If the pattern slides along the direction of the stripes, the image intensity pattern will not change. Again the pattern need only be striped within the region being considered. This problem is typically known in the literature as the "aperture" problem [57]. The "blank wall" and "aperture" singularities are illustrated in figure 1-3. We will address these problems in chapter 3.

Non-motion Brightness Changes

As we mentioned above, most algorithms for estimating image motion fields actually compute optical flow. The use of optical flow as a substitute for image velocities relies on an assumption that changes in the intensities in the images are due *only* to motion. That is, as a point in the world moves, the image brightness corresponding to that point remains constant. This assumption is frequently violated in real images. For example, if the lighting is decreasing over time (eg, the sun goes behind a cloud), then all of the intensities in the image will decrease. [1] More complicated situations arise when the orientation of the surfaces in the world change

[1] This particular global variation in image brightness has been addressed by some authors (eg. [65]).

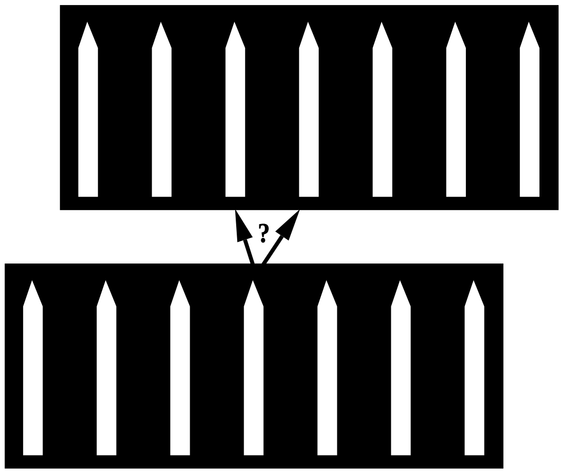


Figure 1-4: Illustration of the "picket fence" (or temporal aliasing) problem. Two frames of a movie are shown, the first below the second. If the motion from one frame to the next is larger than half the distance between pickets, we see the fence moving in the opposite direction.

relative to the camera and thereby alter the amount of light they reflect toward the camera. A motion estimation system that operates under the assumption of image brightness conservation will give incorrect answers in these situations. This problem will also be discussed in chapter 3.

Temporal Aliasing

When the input to a motion estimation system is sampled in time (eg, a sequence of movie frames like the one shown in figure 1-2), large motions may be difficult to estimate. Imagine a film of a picket fence moving to the left. If the fence moves slowly relative to the time between frames, then each picket will move only a small amount from one from to the next, and a local motion algorithm will be able to correctly recognize this. If the fence moves faster, however, we will start to see the pickets moving *backward.* This situation is illustrated in figure 1-4, and is known as "temporal aliasing". In the computer vision literature, the related problem of matching portions of an image or features occuring in two frames is known as the "correspondence problem" [56]. We will discuss this in detail in section 3.2.

Non-translational Motions

One last but very important problem arises from the assumption that motion of each patch of image is translational. Almost all approaches to motion field estimation assume that a single velocity vector is sufficient to describe the motion in each local region of space and time. This assumption is frequently violated in natural scenes: figure 1-5 illustrates three typical examples.

Small violations are due to rotating, expanding, or contracting motions, or the motion

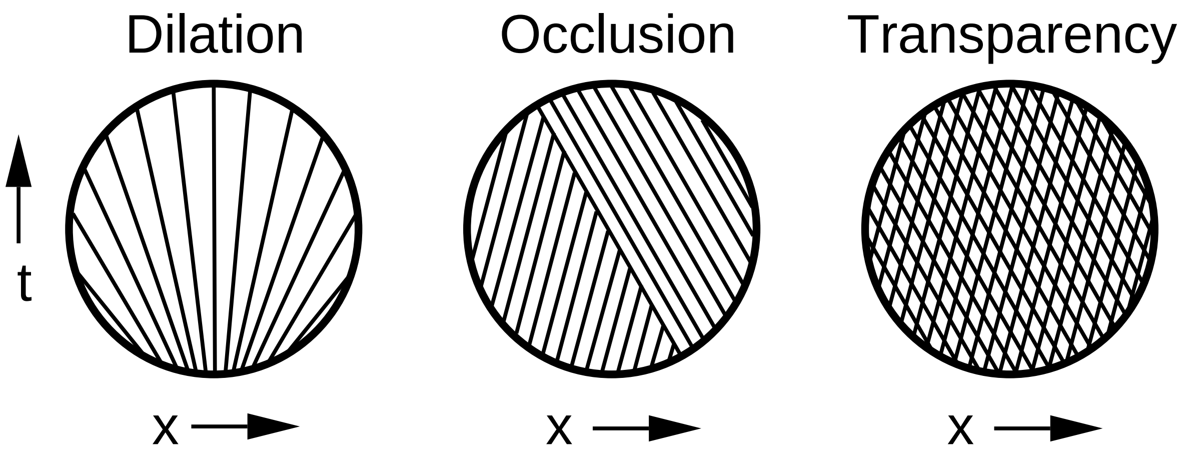


Figure 1-5: Space-time illustrations of three commonly occurring types of non-translational motion. Each circular window shows a local patch of a one-dimensional binary image over time (horizontal axis is space and vertical axis is time). Left: a diverging image. Center: an occlusion boundary, in which two patterns move toward each other, with the right occluding the left. Right: rightward and leftward moving transparently combined patterns.

of non-rigid objects such as fluids or deforming elastic materials (an expanding pattern is illustrated in the left example of figure 1-5). In these instances, the problem is the size of the motion estimation patches. If the regions over which motion is being computed are small compared to the deviations from translational motion, the motion estimation errors will be small.

More severe violations occur at occlusion boundaries, and in the presence of transparent surfaces or highlights (both illustrated in figure 1-5). For example, consider a small patch centered on Lucy's nose in the first frame of figure 1-2. Half of the patch will cover a portion of Lucy's face, and half will cover a portion of the background scenery. Thus the motion of this patch is complicated, and not properly described by a single displacement, even with very small patch sizes. In these situations, there is *more than one motion* occurring within the local region. In chapter 4, we will explore the representation of multiple motions.

1.3 Thesis Overview

In this thesis, we will re-examine both the measurement and representation of visual motion. Our goal is to develop a new framework for the representation of motion information that is mathematically elegant, serves as a basis for biological modeling, and is efficient and robust for use in machine vision applications. We will emphasize the *representational* aspect of the problem and consider the choice of representation as a fundamental aspect of the computational theory.

In chapter 2, we discuss the fundamental issues in motion estimation by describing a set of standard algorithms for computing optical flow. In particular, we formalize the relationship between the standard least-squares gradient-based techniques, least-squares matching techniques, the spatio-temporal filtering models, and the frequency-domain approaches. Each of these approaches provides some unique insights into the problem of motion field estimation.

Remarkably, we show that these algorithms are very similar, and that the primary differences lie in the particular choice of initial linear operators. During the course of our comparisons, we develop a generalized motion algorithm that lies in the intersection of all of these techniques. This generalized algorithm performs better than any of the techniques from which it inherits. Nevertheless, it still suffer from all of the problem sources mentioned in the previous section.

We argue that it is the displacement vector representation of motion that is the source of the difficulty. Given that image motion at a point is often not adequately described by a single translation, the vector representation is a violation of what Marr referred to as the "principle of least commitment" [56]. For example, in a region of the scene suffering from the brick-wall or aperture problem, describing the motion of the region with a single vector is necessarily arbitrary and misleading, since the motion is not uniquely constrained. On the other hand, in a region containing multiple motions, *no single vector* can adequately describe the motion of the region.

As an alternative, we advocate *distributed* representations of motion, in which the encoding of image velocity is implicit. For each small spatial region in the image (and for each time), we define a "velocity energy" function: $E_v(v_x, v_y; x, y, t)$, where x, y, t are the spatial and temporal coordinates of the center of the region, and v_x, v_y correspond to the two components of a candidate velocity. The functional value is large if the candidate velocity is present in that region, and small if not. 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.

The computation of this function is achieved in two stages, as illustrated in figure 1-6. The first stage is based on a set of linear filters whose outputs are squared and normalized to produce an estimate of the local spatiotemporal frequency content of the input imagery. The output of this stage is *also* a distributed representation that we call "spatio-temporal energy", $E_\omega(\omega_x, \omega_y, \omega_t; x, y, t)$. In the second stage, these spectral estimates are combined to form the velocity energy function. In both stages, we show that the entire distribution may be interpolated from a set of values computed at a small number of sample points.

We will describe two related implementations of distributed motion. In chapter 3, we extend the basic algorithm of chapter 2 by introducing a model for the measurement noise and "state" noise in the system. We describe the uncertainty inherent in the computation of

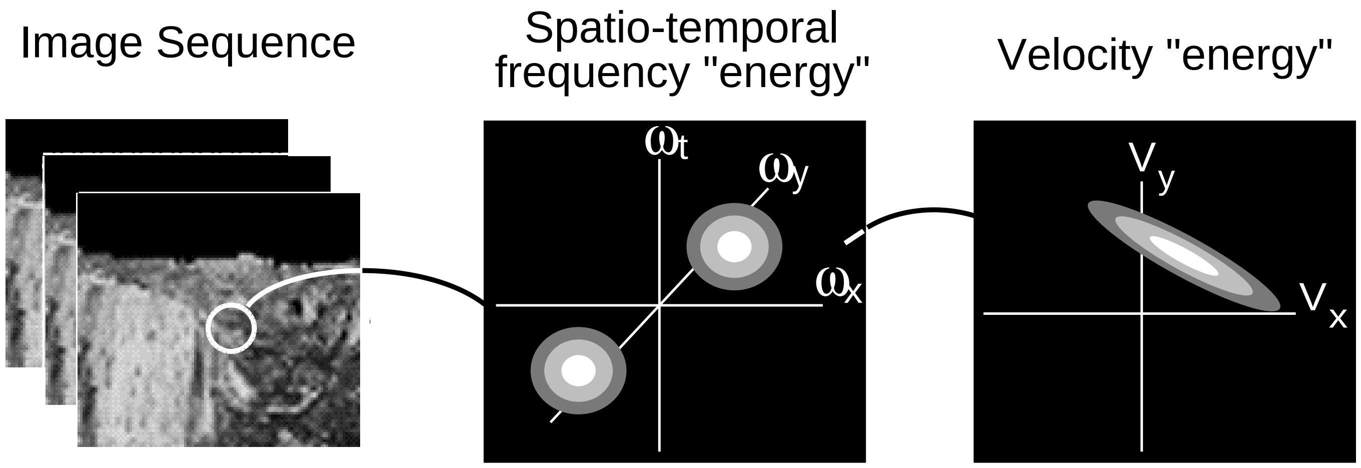


Figure 1-6: Two-stage algorithm for computing a distributed representation of motion. Depicted on the left is the original temporal sequence of images. In the center is a distribution over three-dimensional spatio-temporal frequency, $E_\omega(\omega_x, \omega_y, \omega_t; x, y, t)$, computed in a local patch of the input signal centered at (x, y, t). On the right is a distribution over velocity, $E_v(v_x, v_y x, y, t)$, corresponding to the same local patch. Each of these distributions may be represented by a small number of samples: we illustrate the full interpolated distribution here.

optical flow through use of a simple Gaussian noise model, and we compute a Bayesian least squares estimate solution. The output of the algorithm is a set of probability distributions describing the motion in each patch of the image.

In order to apply this algorithm to temporally sampled imagery, we develop a probabilistic coarse-to-fine algorithm that has the form of a Kalman filter over scale. We show that the resulting algorithm is useful for traditional optical flow estimation, demonstrating its success on both synthetic and real image sequences. We also show that the noise model, although simple, does a reasonable job of accounting for many of the errors in velocity estimation. The main exception to this is the last of the problems mentioned in section 1.2: non-translational motions.

In chapter 4, we extend the basic algorithm of chapter 2 further to develop a model that is capable of representing *multiple motions* at a given position and time in the image. The resulting distributions are no longer restricted to unimodality as in the Gaussian noise case of chapter 3, thus allowing us to represent multiple motions that occur near occlusion boundaries, in regions of strong divergence or curl, and in transparently moving imagery. We demonstrate the use of this model on a set of simple synthetic images.

In the final chapter (chapter 5), we show that the distributed representations of chapters 3 and 4 provide effective models of biological motion perception. We show that the stages of computation can be qualitatively mapped onto the known physiological behavior of cells in the mammalian visual system. We also use them to quantitatively account for a set of psychophysical data on the human perception of motion. Finally we suggest a number of physiological and psychophysical experiments that could be used to test the biological relevance of these types of models.

Chapter 2

Local Mechanisms for Measuring Image Motion

Researchers have studied image velocity for nearly 20 years. Although much of the earliest work is due to television engineers [53, 18] and those working on biological modeling [35, 73], the field today is populated by a wide variety of researchers in Computer Vision and Robotics, Artificial Intelligence, Signal and Image Processing, Communications Engineering, Estimation and Decision Theory, and many other fields. Each of these fields brings a new point of view and a new set of tools to bear on the problem.

Despite this wide variety of approaches, algorithms for computing optical flow are usually divided into three categories:

- **Matching Techniques**. These operate by matching small regions of image intensity or specific "features" from one frame to the next. The matching criterion is usually a least squares (often called Sum-of-Squared-Difference or SSD) or normalized correlation measure.

- **Differential Techniques**. Also known as "gradient" techniques, these estimate optical flow vectors from the derivatives of image intensity over space and time. These are typically derived directly by considering the total temporal derivative of a conserved quantity.

- **Frequency-based or Filter-based Techniques**. These approaches are based on spatio-temporally oriented (i.e., velocity-sensitive) filters, and are typically motivated and are derived by considering the motion problem in the Fourier domain. They fall into two categories – *energy*-based and *phase*-based – both of which will be discussed in section 2.3.

In this chapter, we will describe some of these approaches to motion analysis. In doing this, we will have two goals in mind. The first is to introduce a set of representative solutions to the image velocity problem and to formalize the relationships between them. We will consider approaches that are derived from different motivations in order to understand a set of basic properties that are desirable in a velocity estimation system. In our discussion, we will *not* attempt to give a complete survey of the literature. In particular, we will not devote much discussion to approaches based explicitly on matching (although these constitute a sizable portion of the velocity estimation literature) because they tend to be implausible as biological models, and are usually computationally intensive. Neither will we attempt a thorough experimental comparison of the techniques we consider. It is very difficult to generate comparisons that are both fair and conclusive, although comparative reviews have begun to appear in the literature (eg, [5, 10]).

Our second goal in this section is to sequentially construct a simple yet powerful general algorithm for extracting motion information from spatio-temporal imagery. In the course of considering various approaches to velocity estimation, we will find that many of the solutions are remarkably similar. In particular, we will find that many of the standard approaches in the literature may be written as three simple stages: 1) a linear filtering (convolution) stage, 2) a quadratic combination of the linear filter outputs, and 3) a combination of these quadratic values to compute a velocity energy surface, and find the peak or mean. Perhaps it is not surprising that many of the approaches turn out to be based on quadratic combinations of linear measurements since the underlying error analysis is often a least-squares formulation. Poggio and Reichardt [68] have shown the equivalence of the time-averaged output of all two-input second-order motion detectors. Brockett [16] has shown that several motion extraction algorithms (although not those discussed here) may be described in a common least-squares framework based on Gramians. Some of the work presented in this section has been described previously in [80, 79, 81, 77].

2.1 Differential Techniques

Since velocity is a differential quantity, it is natural to consider its estimation through the use of derivative measurements. In many ways, this is both the simplest and the most elegant approach to motion estimation. Many authors have used differential formulations for motion analysis, with some of the earliest examples being found in [53, 18, 26, 47, 54].

The prototypical gradient formulation of the optical flow problem is based on an assumption of intensity conservation over time [46]. 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 with respect to time 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: $f(x, y, t)$. Setting the total derivative of intensity with respect to time equal to zero gives the *gradient constraint* equation:

$$\begin{aligned} 0 &= \frac{df}{dt} \\ &= \frac{\partial f}{\partial x} v_x + \frac{\partial f}{\partial y} v_y + \frac{\partial f}{\partial t} \\ &= \vec{f_s} \cdot \vec{v} + f_t \qquad (2.1) \end{aligned}$$

where the spatial gradient is written as a vector:

$$\vec{f_s} = \begin{pmatrix} f_x \\ f_y \end{pmatrix},$$

f_x, f_y, and f_t are the spatial and temporal partial derivatives of the image f, $\vec{v}$ is the instantaneous optical flow vector (at the position and time that the derivatives have been computed), and the $(\cdot)$ operator indicates an inner product. Note that the constraint is written for a fixed point in space and time. We have left out the spatial and temporal location parameters in order to simplify the notation.

Measurement Singularities

The gradient constraint given in equation (2.1) defines the relationship between the image intensities and the optical flow, and is fundamental to the development of the rest of this chapter. One immediate property of this equation is that it is insufficient for computing optical flow, since it provides only a single linear constraint for the two unknown components of velocity at each point. To state this more intuitively: by formulating the constraint in terms of first derivatives, we are implicitly modeling the signal as locally linear. A one-dimensional illustration of this is given in figure 2-1.

In two dimensions, the equation implicitly models the image locally as a plane:

$$f(\vec{s}) \approx f(\vec{s}_0) + \vec{f_s} \cdot (\vec{s} - \vec{s}_0), \qquad \text{for } \vec{s} \text{ near } \vec{s}_0,$$

where

$$\vec{s} = \begin{pmatrix} x \\ y \end{pmatrix},$$

and $\vec{s}_0$ is the location at which the derivatives are measured. Since a plane exhibits only one-dimensional intensity variation, the velocity solution will suffer from the aperture problem, as described in the introduction. It is easy to verify that the locus of solutions to equation (2.1) constitutes a line in the two-dimensional space of all velocities, with defining equation

$$\vec{v} = \frac{-f_t \vec{f_s}}{|\vec{f_s}|^2} + \alpha \vec{f_s}^{\perp}, \qquad (2.2)$$

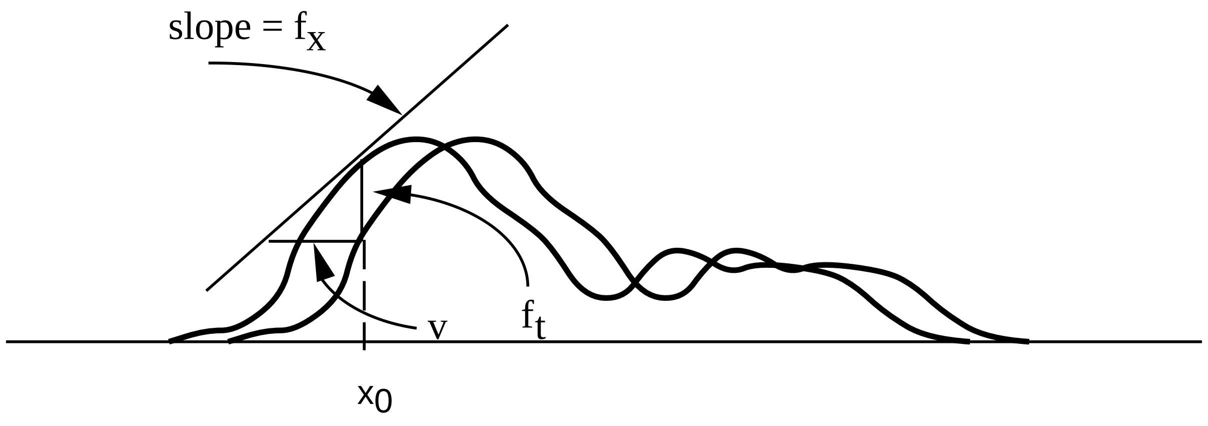


Figure 2-1: Illustration of the gradient constraint on image velocity, in one dimensions. Shown is a one-dimensional intensity signal, and a copy of the signal one time unit later, translated to the right. The spatial derivative defines the slope of a line tangent to the signal at the given point x_0. The temporal derivative measures the change in the signal intensity over time. The velocity corresponds to the distance the signal has moved.

where we define

$$\vec{f}_s^{\perp} = \frac{1}{|\vec{f}_s|}\begin{pmatrix} -f_y \\ f_x \end{pmatrix}, \tag{2.3}$$

and α is the free variable parameterizing the line. This is illustrated in figure 2-2.

The first term of equation (2.2) is known as the *normal velocity*, since it is perpendicular to the local orientation of the signal (as measured by the spatial gradient). The important point here is that the aperture problem is *not* due to the content of the image, f, but is inherent in the gradient measurements. We will therefore refer to this as a "measurement singularity".

We mention one other issue regarding symmetry. The measurement singularity described above is a *one-dimensional* singularity. That is, the velocity is only undetermined in one dimension of the full two-dimensional velocity space. If the spatial gradient of the signal is zero, however, then the singularity will be *two-dimensional.* This situation occurs when the image is locally constant (the "blank-wall" problem, as described in the introduction), or when the image is locally *even-symmetric.*

Local Constraint Combination

Researchers have dealt with the singularity problem by incorporating additional constraints or regularization terms in an energy function to produce a unique solution. Typically, these operate by combining information spatially. Many of these regularization solutions operate by combining the local constraint given by equation (2.1) globally. For example, Horn and Schunck [47] used a constraint of global smoothness of the velocity field. Hildreth [45] used a constraint of smoothness along contours. Nagel [63, 62] used a global *oriented* smoothness

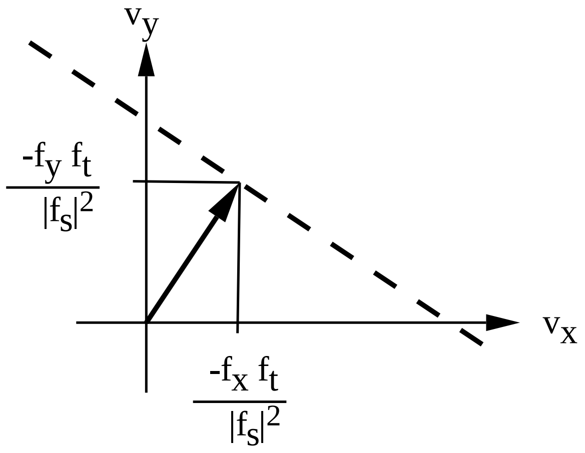


Figure 2-2: The set of velocities that satisfy the gradient constraint in equation (2.1) constitutes a line (illustrated with dashes) in the two-dimensional space of all velocities

constraint, in which less smoothing is done in the direction of the gradient [1]. These solutions perform quite well when their assumptions are met, but problems arise when they are applied to natural images which often do *not* have smooth velocity fields. Another common objection to these approaches is that they are computationally expensive, although several authors have developed more efficient versions (eg, [92, 74, 55]).

For the purposes of this thesis we are interested in *local* combination of constraints. Since we have a constraint on the normal component of velocity at each point, we can choose the velocity that is most consistent with a set of the normal constraints in a small region around the point of interest. Implicitly, this is also a type of smoothness constraint, since we are assuming that the image velocity in the region is constant, but the estimate remains local (i.e., it only depends on the measurements within the region).

We accomplish this by writing a weighted sum-of-squares error function based on the normal constraints from each point within a small region, where the points are indexed by a subscript $i \in \{1, 2, \ldots n\}$:

$$E(\vec{v}) = \sum_i w_i \left[\vec{f}_s(x_i, y_i, t) \cdot \vec{v} + f_t(x_i, y_i, t) \right]^2, \qquad (2.4)$$

where w_i is a set of *positive* weights. Note that since the signal is considered to be continuous, we should write this as an integral over the region, but in anticipation of the application to sampled imagery, we use a sum instead.

To compute a Linear Least-Squares Estimate (LLSE) of $\vec{v}$ as a function of measurements

[1]Note, however, that examination of the Horn and Schunck algorithm reveals that it already does this implicitly.

$\vec{f}_s$ and f_t, we write the gradient (with respect to $\vec{v}$) of this quadratic expression:

$$\begin{aligned} \vec{\nabla}_{\vec{v}}\, E(\vec{v}) &= 2\sum \vec{f}_s \left[\vec{f}_s^T \cdot \vec{v} + f_t\right] \\ &= 2\left[\mathbf{M}\vec{v} + \vec{b}\right] \end{aligned} \tag{2.5}$$

where

$$\mathbf{M} = \sum \vec{f}_s \vec{f}_s^T = \begin{pmatrix} \sum f_x^2 & \sum f_x f_y \\ \sum f_x f_y & \sum f_y^2 \end{pmatrix}, \qquad \vec{b} = \begin{pmatrix} \sum f_x f_t \\ \sum f_y f_t \end{pmatrix}. \tag{2.6}$$

and all of the summations are over the patch, weighted by the weight function w_i as in equation (2.4).

Setting the gradient expression in equation (2.6) equal to the zero vector gives the least-squares velocity estimate:

$$\hat{v} = -\mathbf{M}^{-1}\vec{b}, \tag{2.7}$$

assuming that the matrix M is invertible. Notice that matrix $\mathbf{M}$ and the vector $\vec{b}$ are both composed of blurred quadratic combinations of the spatial and temporal derivatives.

Despite the combination of information over the patch, it is important to recognize that the matrix $\mathbf{M}$ may *still* be singular. As described intuitively in the introduction, we cannot solve for the velocity at locations in the image where the intensity varies only one-dimensionally (corresponding to the "aperture problem") or zero-dimensionally (the "blank wall" problem). More precisely, we can show the following:

Proposition 1 *The matrix* $\mathbf{M}$ *defined in equation (2.6) is singular if and only if there exists a unit vector* $\hat{u}$ *and a set of coefficients* $\{\alpha_i > 0 : i = 1, 2, \ldots n\}$ *such that*

$$\vec{f}_s(x_i, y_i, t) = \alpha_i \hat{u} \qquad \textit{for each } i \leq n.$$

Proof: The proof in the forward direction is simple. Assume the condition holds. Then

$$\mathbf{M} = \sum_i w_i \alpha_i^2 \hat{u}\hat{u}^T, \tag{2.8}$$

and the determinant of $\mathbf{M}$ is:

$$|\mathbf{M}| = \sum_i w_i \alpha_i^2 |\hat{u}\hat{u}^T|.$$

Since the expression in the sum, $|\hat{u}\hat{u}^T|$, is the determinant of an outer product of two two-vectors, it will be zero (i.e., this matrix only spans a one-dimensional subspace).

The converse proof is also fairly straightforward. Since $\mathbf{M}$ is a square matrix, we diagonalize it using its orthonormal eigenvector matrix, $\mathbf{E}$:

$$\mathbf{E}^T\mathbf{M}\mathbf{E} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

If we assume $\mathbf{M}$ is singular, then at least one of the eigenvalues must be zero. Without loss of generality, assume $\lambda_2 = 0$. Then using the definition of $\mathbf{M}$ in equation (2.8):

$$\sum_i w_i \alpha_i^2 \left(\mathbf{E}^T f_{s,i}\right)\left(f_{s,i}^T \mathbf{E}\right) = \begin{pmatrix} \lambda_1 & 0 \\ 0 & 0 \end{pmatrix}. \qquad (2.9)$$

where we are using the shorthand $f_{s,i} = \vec{f}_s(x_i, y_i, t)$.

Now, define a_i and b_i such that

$$\begin{pmatrix} a_i \\ b_i \end{pmatrix} = \mathbf{E}^T f_{s,i}.$$

Substituting into equation (2.9) and extracting the lower right component of the matrix gives

$$\sum w_i \alpha_i^2 b_i^2 = 0$$

which means (recall that w_i is positive) that $b_i = 0$ for all i. Thus,

$$\begin{aligned} f_{s,i} &= \mathbf{E}\begin{pmatrix} a_i \\ b_i \end{pmatrix} \\ &= a_i \hat{e}_1, \end{aligned}$$

where $\hat{e}_1$ is the eigenvector associated with the non-zero eigenvalue, λ_1. ∎

Since the statement that all of the spatial gradient vectors point in the same direction is equivalent to saying that the image varies one-dimensionally, this proposition directly connects the matrix singularity with the aperture problem discussed in the introduction. The blank-wall condition is just the subcase of this situation in which all of the gradient vectors are *zero*.

The solution given in equation (2.7) above may also be derived as a Taylor series approximation to the solution of a matching problem. We define an error function which is the weighted mean squared error of the difference between two image patches at different times and positions:

$$\begin{aligned} E(\vec{v}) &= \sum_i w_i \left[f(x_i + v_x, y_i + v_y, t+1) - f(x_i, y_i, t)\right]^2 \\ &\approx \sum_i w_i \left[f(x_i, y_i, t) + v_x f_x(x_i, y_i, t) + v_y f_y(x_i, y_i, t) + f_t(x_i, y_i, t) - f(x_i, y_i, t)\right]^2 \\ &= \sum_i w_i \left[v_x f_x(x_i, y_i, t) + v_y f_y(x_i, y_i, t) + f_t(x_i, y_i, t)\right]^2 \\ &= \sum_i w_i \left[\vec{f}_s(x_i, y_i, t) \cdot \vec{v} + f_t(x_i, y_i, t)\right]^2, \end{aligned}$$

where we have expanded $f(x_i + v_x, y_i + v_y, t+1)$ in a Taylor series to first order. This is the derivation used by Lucas and Kanade [54], in the context of stereo vision. The resulting error function is identical to that of equation (2.4).

Higher-order Derivatives

The gradient constraint is based on an assumption of image intensity conservation. Some authors have suggested the use of an intensity *derivative* conservation assumption, in which one applies the gradient constraint equation (2.1) to the $x-$, $y-$ and/or $t-$ first derivative sequences.[2] This leads to optical flow calculations that are based on second derivative measurements [38, 63, 94, 95, 69]. If one includes all three partial derivatives (as in [38]), the resulting constraint equations are written as:

$$\begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \\ f_{xt} & f_{yt} \end{pmatrix} \cdot \vec{v} + \begin{pmatrix} f_{xt} \\ f_{yt} \\ f_{tt} \end{pmatrix} = 0. \tag{2.10}$$

As in equation (2.1), this is a pointwise constraint. We have left out the spatial and temporal location parameters in order to simplify notation.

Note that there are now three constraints for the two unknown components of velocity. Thus, unlike equation (2.1), the formulation is no longer inherently underconstrained, but *overconstrained*. Note also that this idea can be taken further to make use of any order of derivative (for example, in [44], we implement a set of third derivative measurements).

It also reasonable to combine constraints arising from different order derivative operators, as suggested. For example, we can combine the second order constraint given above with the first derivative constraint (as was done in [38]):

$$\begin{pmatrix} f_x & f_y \\ f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \\ f_{xt} & f_{yt} \end{pmatrix} \cdot \vec{v} + \begin{pmatrix} f_t \\ f_{xt} \\ f_{yt} \\ f_{tt} \end{pmatrix} = 0. \tag{2.11}$$

As before, we can compute a least-squares solution, averaged over a small patch. Note that the local averaging may no longer be necessary since the system is now overconstrained, but we include it for generality. The solution looks the same as that given in equation (2.7), except that the definitions of $\mathbf{M}$ and $\vec{b}$ in equation (2.6) are replaced by the following:

$$\begin{aligned} \mathbf{M} &= \begin{pmatrix} \sum(f_x^2 + f_{xx}^2 + f_{xy}^2 + f_{xt}^2) & \sum(f_x f_y + f_{xx} f_{xy} + f_{xy} f_{yy} + f_{xt} f_{yt}) \\ \sum(f_x f_y + f_{xx} f_{xy} + f_{xy} f_{yy} + f_{xt} f_{yt}) & \sum(f_y^2 + f_{yy}^2 + f_{xy}^2 f_{yt}^2) \end{pmatrix} \\ \vec{b} &= \begin{pmatrix} \sum(f_x f_t + f_{xx} f_{xt} + f_{xy} f_{yt} + f_{xt} f_{tt}) \\ \sum(f_y f_t + f_{yy} f_{yt} + f_{xy} f_{xt} + f_{yt} f_{tt}) \end{pmatrix}, \end{aligned} \tag{2.12}$$

and the estimator (as before) is:

$$\hat{v} = -\mathbf{M}^{-1}\vec{b}.$$

[2]More generally, applying *any* localized constant-phase filter to an image sequence will not alter its velocity field. Thus, we can apply the gradient constraint equation to a set of images prefiltered with different localized filters (see, for example [88, 87]). We will discuss the issues of prefilter design in section 3.3.

Note that without the summation, the solution is just the standard least-squares pseudo-inverse solution of the linear system in equation (2.11). At this point, this combination of constraints seems a bit arbitrary. In the next two sections, we will give stronger justifications for this "mixed-order" solution.

2.2 Spatio-temporal Filtering Techniques

The previous section discussed differential approaches to estimating image velocity. Another view of motion analysis is based on the use of spatio-temporal filters [68, 24, 30, 96, 99, 1, 40]. Much of this work comes from the computational biology community and is physiologically motivated. These approaches are usually considered to be completely distinct from the differential approaches discussed in the previous section.

In this section, we will discuss a particular set of filter-based models that has received much attention from the computational biology community. These models are are based on the squared outputs of spatio-temporally *oriented* filters. We will refer to these as Spatio-Temporal Energy Models (STEM) [1, 40, 37]. We will show that the mixed-order differential solution discussed in the previous section may be interpreted as a spatio-temporal energy model. The importance of this observation is that it emphasizes the view of derivatives as filtering operations. The design of derivative filters has been somewhat neglected in the optical flow literature, an issue that we address in section 3.3.

Spatio-temporal Energy Models (STEM)

The basic motivation for the spatio-temporal energy models is that motion corresponds to *orientation* in space-time. Consider a translating one-dimensional box signal. If we view this signal in two dimensions of time and space, as illustrated in figure 2-3, then it is clear that translation corresponds to an oblique bar in this space. The inverse slope of the bar corresponds to the speed of the one-dimensional signal. Thus, in a matched-filter fashion, one might imagine applying a set of oriented linear operators, each responding best to signals that are matched to their orientation. The idealized impulse-response lobes of a spatio-temporally oriented filter (matched to the motion of the box) are depicted in figure 2-3.

Note that the linear filter response by itself does not constitute a velocity estimator. In addition to being dependent on the velocity of the underlying signal, the response of the filter is also dependent on the symmetry and the contrast of the signal. Thus, in the example in figure 2-3, the filter would have no response if it were symmetrically centered on the bar. Similarly, a reversal of the contrast would *negate* the filter output. In two-dimensional imagery there is one further unwanted dependency: the response of a spatio-temporally localized filter will vary as we change the *spatial orientation* of the underlying signal. Any mechanism that

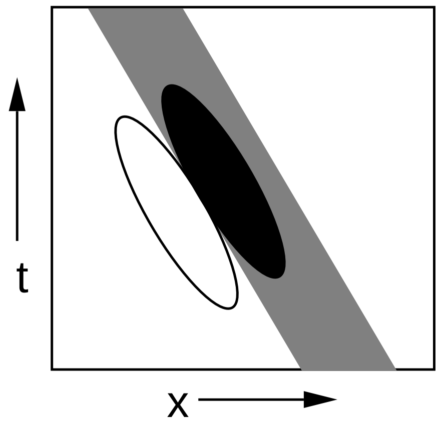


Figure 2-3: Motion corresponds to orientation in space-time. Illustrated is leftward moving one-dimensional signal. It traces out a diagonal path in the two-dimensional space-time diagram. The inverse slope of the lines corresponds to the speed of motion. Also illustrated are positive and negative coefficient lobes of an idealized linear operator (i.e., a filter kernel) with an orientation matched to the orientation of the underlying signal.

computes velocity must eliminate (or at least substantially reduced) these dependencies.

We will describe the basic one-dimensional STEM outlined by Adelson and Bergen here [1]. In order to eliminate the symmetry and contrast dependencies of the filter responses, they proposed the computation of "motion energy" measures from the sum of the square of even- and odd-symmetric filters tuned for the same orientation. They suggested that these energy outputs should be combined in "opponent" fashion, subtracting the output of a mechanism tuned for leftward motion from one tuned for rightward motion. They showed that their opponent STEM approach was closely related to the modified Reichardt correlation model of van Santen and Sperling [73, 96].

Adelson and Bergen also proposed a mechanism for computing a signal monotonically related to speed from these energies [2]:

$$\hat{v} \sim \frac{\sum(R_o^2 + R_e^2) - \sum(L_o^2 + L_e^2)}{\sum(S_o^2 + S_e^2)}, \tag{2.13}$$

where R indicates the output of a linear filter sensitive to rightward motion (space-time orientation), L leftward motion, and S static (zero motion), and the subscripts e and o refer to even- and odd-symmetry of the filters. They showed that this one-dimensional computation was closely related to the basic one-dimensional Lucas and Kanade gradient solution, which is written as:

$$\hat{v} = \frac{\sum(f_x f_t)}{\sum(f_x^2)}. \tag{2.14}$$

We elaborate on this connection in the following sections, extending the argument to two dimensions, and including the use of the even- and odd-symmetric operators of our mixed-order differential solution from the previous section.

Derivatives As Spatio-temporal Filters

In section 2.1, we described differential solutions of the optical flow problem, but we did not discuss the implementation of these solutions. The derivative operator is a linear shift-invariant operator, and we may therefore view its application to a signal as a convolution operation. Derivatives are defined for continuous signals, and in this case, the convolution kernel corresponds to a doublet (or pair of opposite-sign Dirac delta functions). In both machine and biological visual systems, the input imagery is sampled at discrete spatial locations. Since derivatives are only defined on continuous functions, the computation on a discrete function requires (at least implicitly) an intermediate interpolation step with a continuous function $C(x)$. 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:

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

where we assume unit sample spacing in the discrete variables n and m for simplicity. Thus, the derivative operation is defined as convolution with a filter which is the sampled derivative of some continuous interpolation function $C(\vec{r})$. One could use an "ideal" lowpass (sinc) function, or a gentler function such as a gaussian. An optimal choice would depend on a model for the original continuous signal (e.g., bandlimited at the Nyquist limit, with $1/\omega$ power spectrum) and a model of the sensor noise.

Furthermore, one could imagine *prefiltering* the image with a spatially localized filter to extract some spatio-temporal subband and computing the derivatives on this subband (see, for example [59, 88, 87]). Since both operations are linear convolutions, the prefiltering operation can be combined associatively with the derivative operation. The resulting simplified operations is equivalent to convolving with the derivative of the prefiltering function.

Since we have in mind that the output of these filters will be used in the context of the gradient constraint, it is important that each of the derivatives ($x-$, $y-$, and $t-$) be based on the *same* prefilter. Thus, if we were to compute these derivatives with, for example, a purely one-dimensional derivative operator, it is likely that they would produce poor motion estimates, since each applies (perhaps implicitly) a prefilter in the direction of its differentiation, but not in the other directions. Furthermore, in order to ensure that image content is treated homogeneously with respect to orientation, this prefilter should be circularly symmetric in spatial frequency.

We should also mention here that the symmetry of the gradient filter kernels will depend

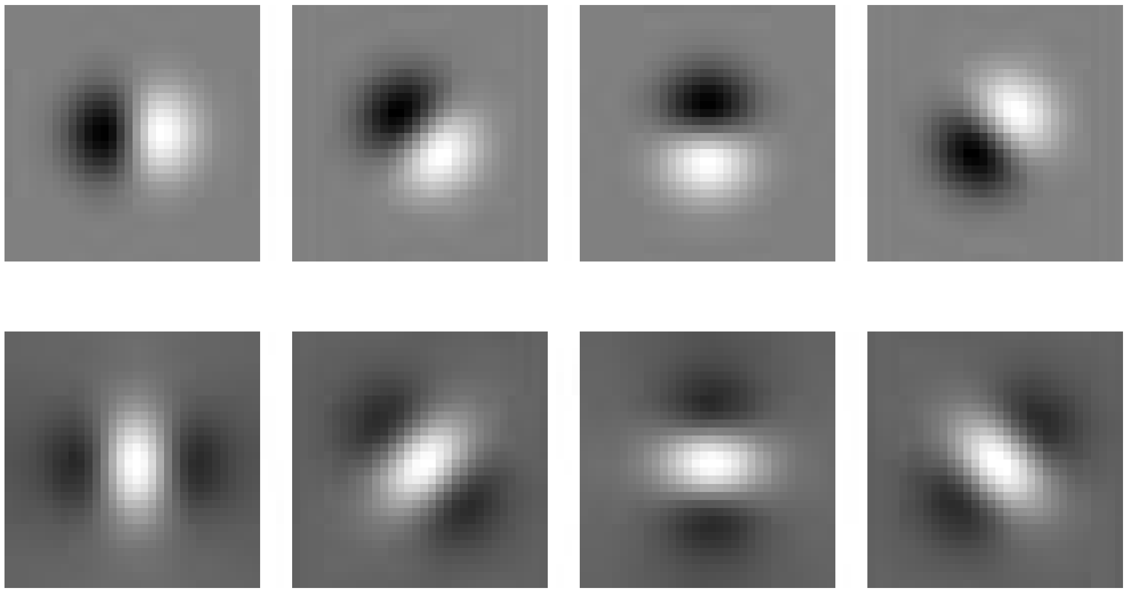


Figure 2-4: Images of two-dimensional derivative kernels at four different orientations, based on a circularly-symmetric prefilter. The top row are first-derivative filters. Note that these are odd-symmetric, since the prefilter is even-symmetric. The bottom row are even-symmetric second-derivatives.

on the symmetry of the underlying prefilter, and on the order of the derivative. Assuming an even-symmetric prefilter, our first derivatives will be odd-symmetric and second derivatives will be even-symmetric. Some example derivative filters, based on a two-dimensional Gaussian prefilter, are illustrated in figure 2-4.

We will discuss the details of filter design in section 3.3. For now, we note only that derivatives of discretely sampled signals are typically computed as differences of neighboring sample values (that is, with the filter kernel $[-1, 1]$), with a two-point average in the non-derivative directions (that is, a filter kernel of $[0.5, 0.5]$). But unless the imagery is bandlimited well below the Nyquist rate, these filters are not very good derivative approximations. For example, Kearney et. al. [50] noted that gradient approaches often perform poorly in highly "textured" regions of an image. We believe that this is due only to the poor choice of prefilter. Examples shown in section 3.4 demonstrate that differential solutions perform very well on textured imagery.

Another point often made in the optical flow literature is that the use of higher-order derivatives (such as the second-order filters illustrated in figure 2-4) is inadvisable, because they are more sensitive to image noise. This argument follows from two facts: 1) image power spectra tend to decrease with increasing frequency and therefore, their high frequency content typically has a low signal-to-noise ratio (SNR), and 2) each additional derivative operation *increases* the influence of the higher frequencies, since the Fourier transform of the derivative operator, for example in the ω_x direction, is $-j\omega_x$. We claim that this is not a fundamental problem in the context of optical flow computation, as it can be overcome through use of a prefilter that counteracts the derivative emphasis on higher frequencies. The real drawback to increasing the order of the derivatives is that the filter kernels become larger (thereby decreasing the localization of the measurements) and that there are more of them (thereby increasing the computation time and storage requirements). On the other hand, we will show in chapter 4 that the narrower orientation bandwidth of the higher-order filters can be advantageous.

The Gradient Solution as a STEM

We have explained that derivative operators are oriented linear filters. But the differential solutions developed in the previous section are based on products of derivatives in different directions, as can be seen from the definitions of $\mathbf{M}$ and $\vec{b}$ in equation (2.12). To view this as a spatio-temporal energy model, we must express it in terms of *squares* of oriented filters.

We first show that the simple differential solution of equation (2.7) may be interpreted as a spatio-temporal energy model, in the sense that it is computed from opponent combinations of squared oriented filter responses. We assume a circularly symmetric prefilter, such as a Gaussian. Then the directional derivatives of this prefilter are oriented filters, as shown in figure 2-4. Furthermore, given the three directional derivatives along the x-, y-, and t- axes (as computed above), we can compute derivatives in other orientations by taking linear combinations of the axis-oriented derivatives.

For example, the spatial derivative of f in a direction oriented at $\pi/4$ (relative to the $x-$axis) is computed as

$$f_p = \frac{1}{\sqrt{2}}(f_x + f_y),$$

where $\vec{p}$ is the unit vector lying in the derivative direction:

$$\vec{p} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

More generally, the spatio-temporal derivative in a direction specified by unit vector $\hat{u}$, may be written as

$$\begin{aligned} f_u &= \left[\hat{u}\cdot \vec{\nabla}\right] f \\ &= \hat{u} \cdot \begin{pmatrix} f_x \\ f_y \\ f_t \end{pmatrix}, \end{aligned} \qquad (2.15)$$

where the operator $\vec{\nabla}$ is the *spatio-temporal* gradient operator:

$$\vec{\nabla} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial t} \end{pmatrix}$$

This ability to interpolate filter responses at arbitrary orientations from a set of filter responses at *fixed* orientations is an extremely important property of derivatives, and we will discuss it in more detail in chapter 4. For now, we note that this property is not limited to the family of derivative functions. A general theory of such filters, known as "steerable" filters, has been developed by Freeman and Adelson [31].

Given the relationship of equation (2.16), we can start to make a connection between the gradient constraint equation and the oriented filtering concept of the STEM. We rewrite equation (2.1) as follows:

$$\begin{aligned} \frac{df}{dt} &= \begin{pmatrix} \vec{v}_x \\ \vec{v}_y \\ 1 \end{pmatrix} \cdot \begin{pmatrix} f_x \\ f_y \\ f_t \end{pmatrix} \\ &= \sqrt{|\vec{v}|^2 + 1}\left[\hat{u}(v) \cdot \vec{\nabla}\right] f, \end{aligned}$$

where $\hat{u}(v) = (\vec{v}_x, \vec{v}_y, 1)^T / \sqrt{|\vec{v}|^2 + 1}$. Apart from the initial factor of $\sqrt{|\vec{v}|^2+1}$, this expression is identical to the directional derivative expression of equation (2.16). When viewed as a function of velocity, it computes a directional derivative for each candidate velocity $\vec{v}$. This fits with the intuition developed by Adelson and Bergen, except that the constraint of equation (2.1) is a *nulling* constraint: it seeks a velocity (orientation) such that the response of the operator is zero. The Adelson and Bergen approach is motivated in terms of the orientation of *maximal* response. We will return to this issue in chapter 4.

Despite the fact that the gradient constraint operates via nulling, we can show that the least-squares solution developed in equation (2.7) may be written in terms of opponent STEM computations. Using the relationship of equation (2.16), we can write the mixed-derivative entries of $\mathbf{M}$ and $\vec{b}$ as defined in equation (2.6) in terms of squared directional derivatives:

$$\mathbf{M} = \begin{pmatrix} \sum f_x^2 & \sum \frac{1}{2}(f_p^2 - f_q^2) \\ \sum \frac{1}{2}(f_p^2 - f_q^2) & \sum f_y^2 \end{pmatrix}, \qquad \vec{b} = \begin{pmatrix} \sum \frac{1}{2}(f_l^2 - f_r^2) \\ \sum \frac{1}{2}(f_d^2 - f_u^2) \end{pmatrix}.$$

where we define

$$\begin{aligned} f_q &= \tfrac{1}{\sqrt{2}}(f_x - f_y), \qquad f_l = \tfrac{1}{\sqrt{2}}(f_x + f_t), \\ f_r &= \tfrac{1}{\sqrt{2}}(f_x - f_t), \qquad f_d = \tfrac{1}{\sqrt{2}}(f_y + f_t), \\ f_u &= \tfrac{1}{\sqrt{2}}(f_y - f_t). \end{aligned} \qquad (2.16)$$

The letters are meant to be mnemonic (l = leftward, r = rightward, d = downward, u = upward). Note that each component of $\mathbf{M}$ and $\vec{b}$ is a local average of squares of a directional filter, or the difference of two such squares. The components of $\mathbf{M}$ and $\vec{b}$ are pure or opponent energy computations, just as those in Adelson and Bergen's opponent motion computation of equation (2.13). Of course, the expression for velocity is more complicated than the one-dimensional version described by Adelson and Bergen: in place of scalar division, we now have multiplication by the inverse of a matrix.

This formulation in terms of directional derivatives provides an elegant interpretation of the standard gradient solution in terms of oriented filtering. In chapter 5 we will show that it is also useful for biological modeling. We note, however, that the implementation of the gradient solution on a standard digital computer is most efficiently accomplished with *separable* derivative filters. Furthermore, it is important to realize that the set of directions chosen in equation (2.16) is not unique. There are many such choices that would allow us to compute the elements of $\mathbf{M}$ and $\vec{b}$. [3]

The Mixed-order Solution as a STEM

Now we consider the mixed-order solution of equation (2.12). As in the first derivative case, we can make use of an interpolation formula for computing *directional second* derivatives from separable second derivatives:

$$\begin{aligned} f_{uu} &= \frac{\partial^2 f}{\partial \hat{u}^2} \\ &= \left[\hat{u}\cdot \vec{\nabla}\right]^2 f \\ &= u_x^2 f_{xx} + 2u_x u_y f_{xy} + u_y^2 f_{yy} + u_x u_t f_{xt} + u_y u_t f_{yt} + u_t^2 f_{tt}. \qquad (2.17) \end{aligned}$$

That is, a directional second derivative may be written as a linear combination of the set of all mixed (i.e., including cross-derivatives) second derivatives.

First consider the upper-left component of $\mathbf{M}$:

$$\begin{aligned} m_{11} &= \sum \left[f_x^2 + (f_{xx}^2 + f_{xy}^2 + f_{xt}^2)\right] \qquad (2.18) \\ &= \sum \left[f_x^2 + \tfrac{1}{2}\left(f_{xx}^2 + \tfrac{1}{2}\{f_{xx}^2 + 4f_{xy}^2 + f_{yy}^2\} + \tfrac{1}{2}\{f_{xx}^2 + 4f_{xt}^2 + f_{tt}^2\} - \tfrac{1}{2}f_{yy}^2 - \tfrac{1}{2}f_{tt}^2\right)\right] \\ &= \sum \left[f_x^2 + \tfrac{1}{2}\left(f_{xx}^2 + f_{pp}^2 + f_{qq}^2 + f_{rr}^2 + f_{ll}^2 - \tfrac{1}{2}f_{yy}^2 - \tfrac{1}{2}f_{tt}^2\right)\right] \qquad (2.19) \end{aligned}$$

The first equation contains squares of cross-derivative terms, so we have used equation (2.17) to rewrite it as a combination of squares of *directional* second derivatives. The direction vectors are defined in equations (2.16). Now compare with the Adelson and Bergen formulation given in equation (2.13). The first term in the brackets, f_x^2, is the square of an odd-symmetric filter response, as in the Adelson and Bergen version. The terms in parentheses are squared even-symmetric filter responses. But unlike the Adelson and Bergen formulation which uses a single even-symmetric filter (the Hilbert transform of the odd-symmetric filter), the solution of equation (2.19) combines *seven* squared directional filter outputs to get the even-symmetric portion of the energy expression. In section 2.3, we will see that there is an important reason for this particular combination.

[3]The space of rotations of the directional derivative filter forms a linear subspace. There are many possible choices of linear basis that span this subspace (cf. [31]).

We can also show that the non-squared components of $\mathbf{M}$ and $\vec{b}$ may be written as opponent combinations of expressions identical to the expression in equation (2.18). In particular, using the interpolation formula of equation (2.16) and a little algebra we can show that:

$$
\begin{aligned}
m_{12} &= \sum \frac{1}{2}\left\{\left[f_p^2 + f_{pp}^2 + f_{pq}^2 + f_{pt}^2\right] - \left[f_q^2 + f_{qq}^2 + f_{pq}^2 + f_{qt}^2\right]\right\}, \\
b_1 &= \sum \frac{1}{2}\left\{\left[f_r^2 + f_{rr}^2 + f_{rl}^2 + f_{ry}^2\right] - \left[f_l^2 + f_{ll}^2 + f_{rl}^2 + f_{ly}^2\right]\right\}, \\
b_2 &= \sum \frac{1}{2}\left\{\left[f_u^2 + f_{uu}^2 + f_{ud}^2 + f_{ux}^2\right] - \left[f_d^2 + f_{dd}^2 + f_{ud}^2 + f_{dx}^2\right]\right\}. \qquad (2.20)
\end{aligned}
$$

Each of these expressions is a difference (i.e., an opponent combination) of two rotated versions of the expression given in equation (2.18). Each could also be written, as in equation (2.19) as a combination of squared directional second derivatives. Altogether, the solution requires the use of 12 directional second-derivatives.

The point of all of this manipulation is that the mixed-order solution of equation (2.12) may be rewritten in terms of opponent spatio-temporal energy mechanisms in which the spatio-temporal filters are first and second-order directional derivatives. Considering the solution in this fashion provides an elegant interpretation of the algorithm in terms of oriented filtering, and leads to useful biological models. In the next section, we will consider the basic and mixed-order solutions from yet another vantage point: the Fourier transform domain.

2.3 Frequency-domain Description

Many important insights about the estimation of motion may be explained most directly by considering the problem in the Fourier domain. This is easily appreciated by considering the motion of a one-dimensional signal. We can represent such a signal as an intensity image, in which the intensity of each pixel corresponds to the value of the signal at a particular location and time. A translating one-dimensional signal has the appearance of a striped pattern, where the stripes are oriented at an angle of $\alpha = \arctan(1/v)$. 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 α. This is illustrated in figure 2-5.

The situation in two dimensions, while more difficult to illustrate, is analogous. A translating two-dimensional pattern has the appearance of oriented "bundles of fibers" in space-time (x, y, t). Watson and Ahumada [98] and others have noted that the Fourier transform spectrum of an image undergoing rigid translation lies in a *plane* in the spatio-temporal frequency domain.

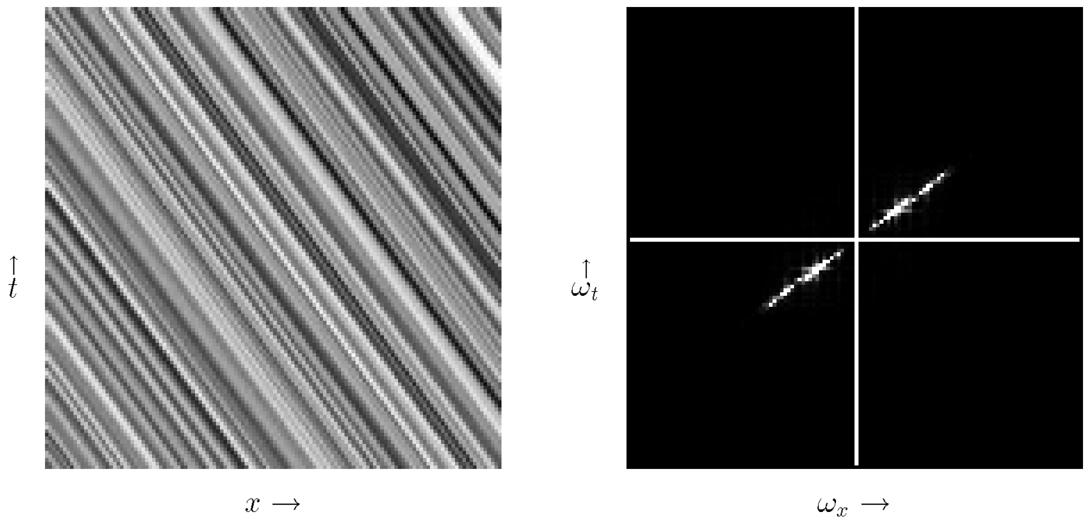


Figure 2-5: The Fourier spectrum of a translating one-dimensional pattern lies on a line in the frequency domain. On the left is a translating fractal noise signal. On the right, its power spectrum, plotted with ω_x and ω_t ranges from $-\pi$ to π. Deviations from the line are due to the use of a finite-size Fourier transform.

Frequency-domain Regression

These examples suggest an alternative approach to the measurement of optical flow: to search for the plane that best fits the power spectrum of the spatio-temporal signal. In practice, we are interested in local estimates of image velocity, and thus we need to use a local estimate of the power spectrum. We note that this type of approach for estimating orientation has been used in the array-processing literature for the problem of "direction-of-arrival" estimation [49, 25].

This concept was used by Heeger [39] to develop an algorithm for the computation of optical flow. He made local measurements of the power spectrum using a set of twelve Gabor filters [4] tuned for different spatio-temporal frequencies. The arrangement of the spectra of these filters is illustrated in figure 2-6. He then used a numerical optimization procedure to find the plane that best accounted for the measurements (the error criterion was least-squares regression on the filter energies). Grzywacz and Yuille [37] also constructed velocity estimators based on energies computed from Gabor filters. They developed both a least-squares solution similar to Heeger's, in addition to a correlational (or template-matching) solution.

A fundamental problem with these approaches is that the resulting velocity estimates depend on the local spatial content of the signal. In particular, each of these techniques gives the wrong velocity for any moving sinusoidal grating whose spatial frequency is not matched to the center response of the filters [5]. The problem is illustrated in figure 2-7, and is due to

[4] A Gabor function is a sinusoid multiplied by a Gaussian window [32].

[5] Grzywacz and Yuille state that their solution avoids this problem, but this is only true in the limit as the spatial bandwidth of the filters approaches zero, or as the number of filters grows toward infinity.

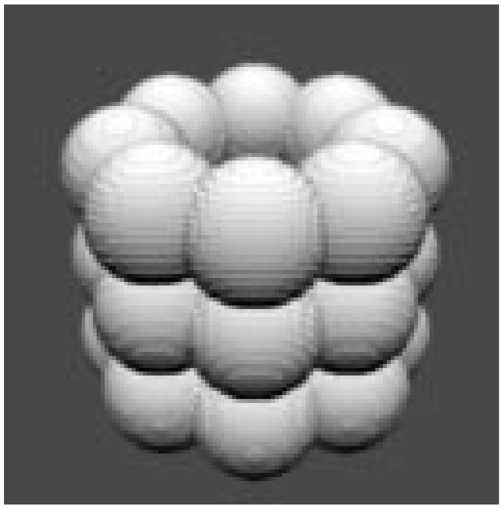


Figure 2-6: Arrangement of the twelve Gabor filters used in the regression solution of [39]. Shown are level surfaces of the power spectra of the filters in the three-dimensional $(\omega_x, \omega_y, \omega_t)$ frequency domain. The ω_t axis is along the cylindrical axis of symmetry of the filter locations. Surfaces are rendered assuming a fixed point light source and a Lambertian reflectance function.

the somewhat arbitrary choice of filters. Gabor filters are chosen because they minimize a joint space-frequency localization criterion [32], and because they have been suggested for use in biological modeling [22]. But this joint-localization constraint is not of primary importance in the velocity estimation problem, and many other choices of oriented bandpass filter serve equally well for purposes of physiological modeling.

Consider the basic gradient algorithm in the Fourier domain. Recalling equation (2.16), the solution may be rewritten as a computation based on squares of spatio-temporally oriented filters, just as the solutions discussed above. There are eight of these directional derivative filters. An example set, based on a Gaussian prefilter, are illustrated in figure 2-8. Note that the shape and position of these filters is different than the Gabor filters of figure 2-6. They are spherically *polar-separable*, and the basic geometry of their arrangement is spherical rather than cylindrical [6].

We can now show that the velocity estimate computed from these directional derivative filters will be invariant to simple scaling of the spatial frequency of a sinusoidal input signal. The response of each of the filters has the form $-j[\hat{u}_i \cdot \vec{\omega}_0]G(\vec{\omega}_0)$ where $\vec{\omega}_0$ is the spatio-temporal frequency of the sinusoidal input, $\hat{u}_i$ is a unit vector in the direction of the ith filter derivative, and $G(\vec{\omega})$ is the prefilter. Now if the input signal undergoes the transformation $F(\vec{\omega}) \longrightarrow F(\alpha\vec{\omega})$, then the filter responses will become $\alpha[\hat{u}_i \cdot \vec{\omega}_0]G(\alpha\vec{\omega}_0)$. That is, the result of the transformation is that each filter response will be multiplied by a factor of $\alpha G(\alpha\vec{\omega}_0)/G(\vec{\omega}_0)$. Since the velocity solution of equation (2.7) only depends on the ratios of linear combinations

[6]This is partially due to the choice of a spherically symmetric prefilter – a spatially bandpass and temporally lowpass prefilter would lead to a cylindrical geometry.

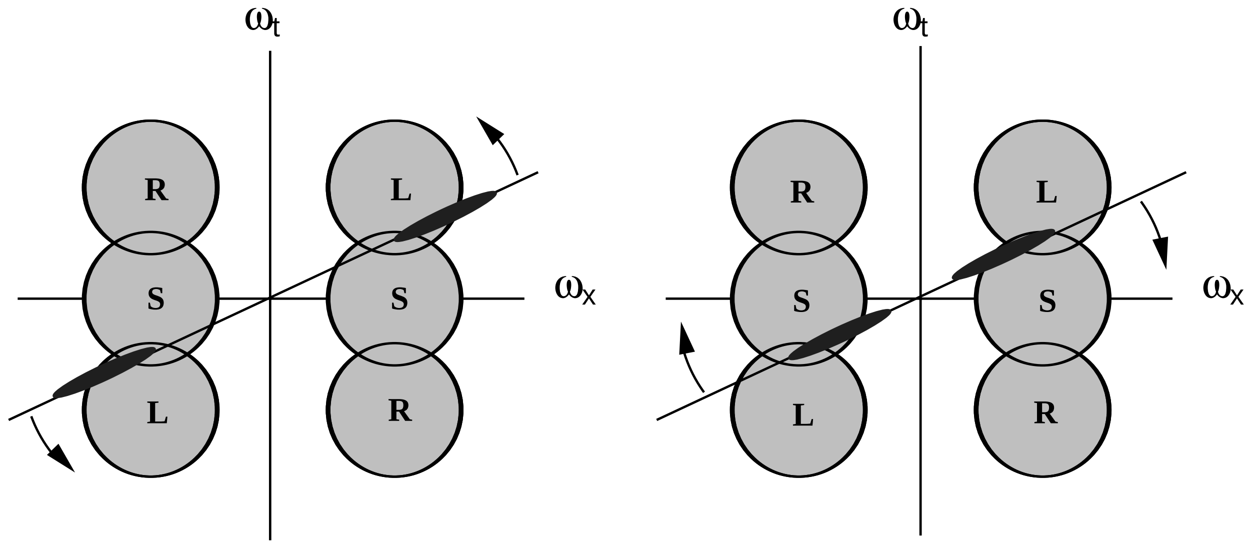


Figure 2-7: One-dimensional illustration of the systematic errors in velocity estimates computed from Gabor filters. The level surfaces of the filter power spectra are elliptical (circular here), and are represented by the overlapping circles. Assume the signal power spectrum lies on the line shown (that is, the signal is moving at a speed corresponding to the slope of this line). If the spectral distribution is concentrated at higher spatial frequencies (illustrated by the bulge in the line of the plot on the left), the "leftward" filter response will increase relative to that of the other two filters. Since the velocity estimator relies on the relative responses of the filters, it will over-estimate the speed of the signal. Similarly, if the spectral content is at lower frequencies (shown on the right), a regression estimator will underestimate the speed.

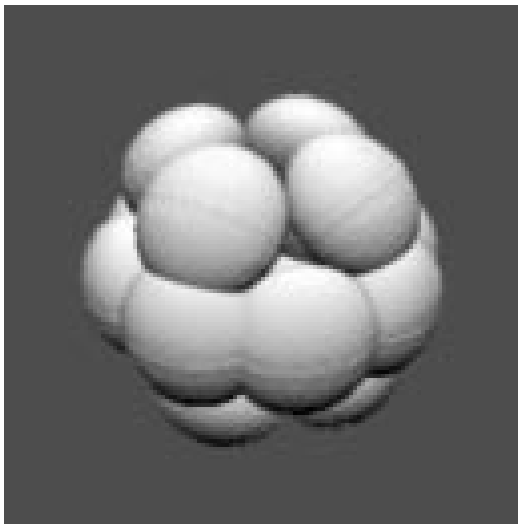


Figure 2-8: Illustration of the spatio-temporal frequency spectra of the directional derivative filters used to compute the basic gradient solution. Shown are level surfaces of the spectra for a set of eight filters. Each filter appears as two flattened balls, symmetrically arranged about the origin. These should be compared to those in figure 2-6.

of the squared filter responses, it will be unchanged.

Furthermore, we can show that the basic gradient approach may be viewed *directly* as a planar regression analysis in spatio-temporal frequency. Consider the energy function given in equation (2.4) with the summation occurring over the entire image:

$$\begin{aligned} E(\vec{v}) &= \sum_{\vec{x}} |v_x f_x + v_y f_y + f_t|^2 \\ &= \sum_{\vec{\omega}} |v_x F(\vec{\omega})\omega_x + v_y F(\vec{\omega})\omega_y + F(\vec{\omega})\omega_t|^2 \\ &= \sum_{\vec{\omega}} [v_x\omega_x + v_y\omega_y + \omega_t]^2 \cdot |F(\vec{\omega})|^2 \end{aligned} \tag{2.21}$$

where the sum on the first line is over all image pixels and the sums on the latter two lines are over all frequencies, $\vec{\omega}$. We have used Parseval's rule to switch to the frequency domain, and the fact that the Fourier transform of the derivative operator in, for example, the $x-$ direction is $j\omega_x$. The term in square brackets is the squared ω_t-distance between the point $\vec{\omega}$ and the plane defined by $v_x\omega_x + v_y\omega_y = -\omega_t$. This equation is exactly in the form of a least-squares planar regression error function, weighted by the image power spectrum, $|F(\vec{\omega})|^2$! This means that if the input signal spectrum lies on a plane (and does *not* lie on a line), the solution will always be correct, regardless of the spectral distribution on the plane.

More generally, we can combine the differential constraints derived from a set of different prefilters, $g_n(x)$ and the result may still be viewed as a regression solution. We rewrite equation (2.21) as:

$$\begin{aligned} E(\vec{v}) &= \sum_{n}\sum_{\vec{x}} |v_x f_{n,x} + v_y f_{n,y} + f_{n,t}|^2 \\ &= \sum_{n}\sum_{\vec{\omega}} |v_x F(\vec{\omega})G_n(\vec{\omega})\omega_x + v_y F(\vec{\omega})G_n(\vec{\omega})\omega_y + F(\vec{\omega})G_n(\vec{\omega})\omega_t|^2 \\ &= \sum_{\vec{\omega}} [v_x\omega_x + v_y\omega_y + \omega_t]^2 \cdot \sum_n |G_n(\vec{\omega})|^2 \cdot |F(\vec{\omega})|^2 \end{aligned} \tag{2.22}$$

This is still a regression solution, but now the power spectrum of the image is weighted by the sum of the spectra of the prefilters, $\sum_n |G_n(\vec{\omega})|^2$.

We have shown that the global form of the basic gradient algorithm operates as a regression algorithm, seeking the plane that best accounts for the derivative measurements. In doing this, we assumed that the summations in equation (2.6) were performed over all space and time. In practice, we compute a local approximation to this by averaging over a small patch. An alternative method for estimating local power spectra is to use quadrature pairs of filters that are Hilbert transforms of each other. We will discuss this concept next, and show that the mixed derivative solution performs an operation very similar to this.

Local Estimates of Phase and Spectral Energy

Phase and spectral energy are usually defined in the context of the Fourier basis set. The phase of a sinusoid is its position (in units of the sinusoid period, modulo 2π) relative to an arbitrary but fixed origin. The space of all translations of a sinusoid at a given frequency is spanned by two sinusoids, differing in phase by $\pi/2$: $\sin(\omega x)$ and $\sin(\omega x + \pi/2) = \cos(\omega x)$. The phase response of the Fourier spectrum is computed from the projection of the signal onto these two sinusoids:

$$\phi(\omega) = \arctan\left(\frac{\sum_x f(x)\sin(x)}{\sum_x f(x)\cos(x)}\right)$$

The spectral energy is just the sum of squares of these two projections. An important property is that the spectral energy of a signal is invariant to translations of the signal.

There are also definitions of phase and energy for transforms based on localized bandpass basis functions. Typically, the basis set is constructed in one of two ways:

1. For each basis function, $G(\omega)$, one can compute a Hilbert transform, $-j\mathrm{sgn}(\omega)G(\omega)$. Each pair of basis functions comprised of a basis function and its Hilbert transform is known as a *quadrature pair.* In multiple dimensions, the Hilbert transform is directional, and is computed as $-j\mathrm{sgn}(\hat{u}\cdot\vec{\omega})G(\vec{\omega})$, where $\hat{u}$ is the direction. Typically, the quadrature pair is designed so that one kernel is purely odd-symmetric and the other even-symmetric.

2. One can also define a windowed Fourier transform, based on some lowpass windowing function. That is, the basis functions are formed as a product of $\cos(\omega x)$ and $\sin(\omega x)$ with the window function. An example that has been used often in the image processing literature is the Gabor basis set [32] which is composed of pairs of Gaussian-windowed sinusoids and cosinusoids.

In both of these cases, local phase is defined as the arctangent of the ratio of responses of the pairs of filters, and local spectral energy is the squared sum of the these.

This definition of local phase and energy will give the correct phase and energy measures for sinusoidal input signals (i.e., they will correspond to the phase and squared amplitude of the input sinusoid). We wish to point out, however, that these definitions are lacking an important property that is present in the Fourier definitions. A shift in local phase does *not* correspond to a simple translation of the basis function. This is because the quadrature pair of functions does not span the space of their own translations. The price we pay for using a local estimate of these quantities is blindness to certain signals.

For example, consider an image composed of two sinusoids of nearby frequencies $\vec{\omega}_1 = (\frac{\pi}{2}, \Delta)$ and $\vec{\omega}_2 = (\frac{\pi}{2}, -\Delta)$:

$$f(\vec{x}) = \cos(\vec{\omega}_1 \cdot \vec{x}) - \cos(\vec{\omega}_2 \cdot \vec{x}).$$

Now consider the response of a quadrature pair of Gabor kernels with center frequency $\vec{\omega}_0 = (\frac{\pi}{2}, 0)$, centered at the origin. Clearly, the signal spectrum lies within the region of response for the filters. The inner product of the odd-phase Gabor kernel with the signal will be zero. Let $g_e(\vec{x})$ be an the even-phase Gabor kernel:

$$g_e(\vec{x}) = e^{-|\vec{x}|^2/2} \cos(\vec{\omega} \cdot \vec{x}).$$

Then its inner product with the signal will be:

$$\begin{aligned} \sum_{\vec{x}} f(\vec{x}) g_e(\vec{x}) &= \sum_{\vec{\omega}} F(\vec{\omega}) G_e(\vec{\omega}) \\ &= G_e(\vec{\omega}_1) - G_e(\vec{\omega}_2) \\ &= 0, \end{aligned}$$

where we have used Parseval's rule to write the inner product in the frequency domain, and the symmetry of $G_e(\vec{\omega})$ leads to the cancellation of the last line.

Thus, the local spectral energy measurement will be zero, even though these the input signal clearly falls within the passband of the filters. We refer to this as the "phase-cancellation" problem. The number of signals (actually, the dimensionality of the space of signals) to which the local energy measure will be blind is proportional to the size of the passband. Of course, the energy will be non-zero for the Gabor pair at slightly different locations. Therefore, if we average the local energy over a small patch (as we did in the basic gradient solution), we can overcome this singularity. In other words, we could use a sum of squared responses of *more* than two basis functions as a local estimate of spectral energy.

In the motion literature, several authors previously mentioned [1, 40, 29] have used quadrature pairs in the computation of motion. As we discussed in the previous section, this can be justified from a filtering point of view in terms of symmetry. If the local signal is even-symmetric about a particular location, an inner product with an odd-symmetric operator (such as a first derivative) will return zero. An even-symmetric filter, however, will respond. The situation is reversed when the signal is odd-symmetric. Therefore, combining the two measurements will eliminate these singularities.

The Mixed-order Solution as a Quadrature Energy

Now, we would like to interpret the mixed first and second derivative computation of equation (2.12) in terms of quadrature measurements. We have already noted that the first derivatives are odd-symmetric, and the second derivatives are even-symmetric. But the second derivative filter cannot be *equal* to the Hilbert transform of a first derivative filter. Consider the Fourier transform of, for example, the first and second derivative filters in the x direction:

$$\mathcal{F}\left(\frac{\partial g_1(x,y,t)}{\partial x}\right) = -j\omega_x G_1(r)$$

$$\begin{aligned} &= -jr\frac{\omega_x}{r}G_1(r) \\ &= -jr[\hat{u}_x \cdot \hat{\omega}]G_1(r) \qquad (2.23) \end{aligned}$$

$$\begin{aligned} \mathcal{F}\left(\frac{\partial^2 g_2(x,y,t)}{\partial x^2}\right) &= -{\omega_x}^2 G_2(r) \\ &= -r^2\frac{\omega_x^2}{r^2}G_2(r) \\ &= -r^2[\hat{u}_x \cdot \hat{\omega}]^2 G_2(r), \qquad (2.24) \end{aligned}$$

where $g_1(x,y,t)$ and $g_2(x,y,t)$ are the circularly symmetric prefilters, as described in a previous section, $r = |\vec{\omega}|$ is the radial frequency coordinate, $\hat{u}_x$ is the ω_x-axis unit vector, and $\hat{\omega} = \vec{\omega}/|\vec{\omega}|$.

Note that the inner product (in square brackets) corresponds to the cosine of the angle between the vector $\vec{\omega}$ and the ω_x-axis. The frequency-domain forms of the two filters differ both in their angular and radial component. We may eliminate the radial difference by defining:

$$r^2 G_2(r) = rG_1(r) = G(r). \qquad (2.25)$$

Combining equations (2.23), (2.24), and (2.25) produces a two filters that are products of a common prefilter, $G(\vec{\omega})$, and a *directional cosine* function or its square. Thus, the angular portion of the two filters cannot be made the same.

Nevertheless, we can show that the sums of quadratic terms in the solution of equation (2.12) may be viewed as quadrature energies. Consider the upper left component of the matrix **M**. Using Parseval's rule to rewrite this component in the frequency domain gives:

$$\begin{aligned} m_{11} &= \sum \left[f_x^2 + (f_{xx}^2 + f_{xy}^2 + f_{xt}^2)\right] \\ &= \sum_{\vec{\omega}} |F(\vec{\omega})|^2 |G(\vec{\omega})|^2 \left(\tfrac{\omega_x}{r}\right)^2 + \sum_{\vec{\omega}} |F(\vec{\omega})|^2 |G(\vec{\omega})|^2 \left[\left(\tfrac{\omega_x^2}{r^2}\right)^2 + \left(\tfrac{\omega_x\omega_y}{r^2}\right)^2 + \left(\tfrac{\omega_x\omega_t}{r^2}\right)^2\right] \\ &= \sum_{\vec{\omega}} |F(\vec{\omega})|^2 |G(\vec{\omega})|^2 \left(\tfrac{\omega_x^2}{r^2}\right) + \sum_{\vec{\omega}} |F(\vec{\omega})|^2 |G(\vec{\omega})|^2 \left(\tfrac{\omega_x^2}{r^2}\right)\left[\tfrac{\omega_x^2+\omega_y^2+\omega_t^2}{r^2}\right]. \qquad (2.26) \end{aligned}$$

But since $r^2 = \omega_x^2 + \omega_y^2 + \omega_t^2$, we see that the second summation is equivalent to the first summation! That is, this combination of second derivative terms has the same power spectral response as the first derivative term. This is illustrated in figure 2-9. We have remarked, however, that the first and second derivative operators will have opposite symmetry. Thus the computation is very similar to a quadrature energy computation.

The same argument may be made to show that the other components of **M** and the components of $\vec{b}$ may be viewed as quadrature energy expressions, or differences of quadrature energies, using equation (2.20). Notice the role of $G(\vec{\omega})$ in the equation: it serves to weight the signal spectrum before the regression is computed.

In summary, typical quadrature energy computations operate by summing the squares of responses to an even-symmetric and an odd-symmetric filter, where the two filters are

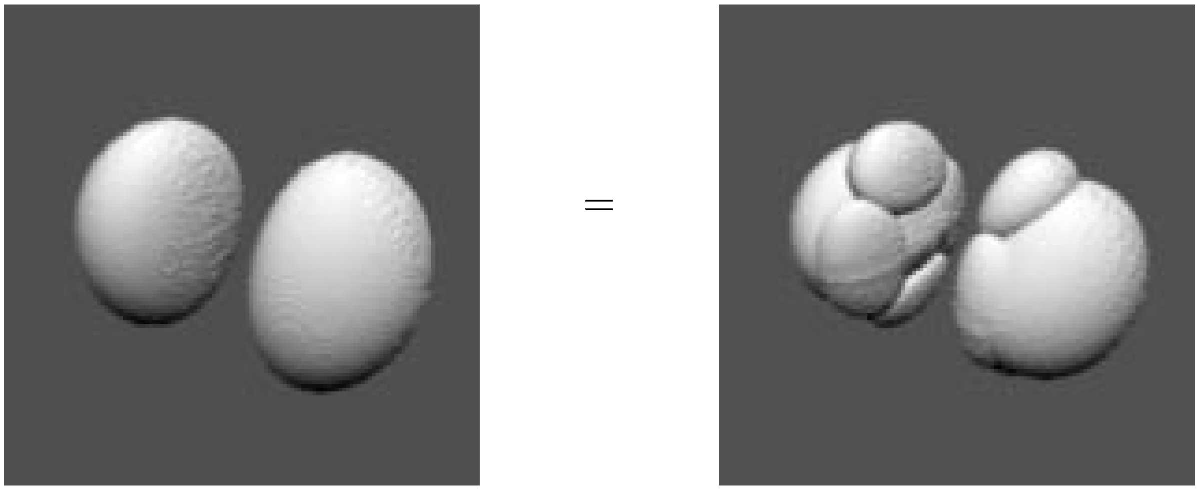


Figure 2-9: Illustration of the similarity of derivative filters and quadrature pairs. On the left is a lambertian-shaded rendering of a level surface of the power spectrum of a first derivative filter. On the right is a rendering of the level surfaces of three second derivative power spectra: f_{xx}, f_{xy}, and f_{xt}. The sum of these power spectra is exactly equal to the first derivative power spectrum. Note that the prefilters in the two cases differ by a factor of r, as described in the text.

Hilbert transforms of each other. The quadratic expressions in our mixed-order solution (equation (2.12)) are computed from squares of even- and odd-symmetric filters, but the even-symmetric computation involves *three* different filters. Spectrally, however, the even-symmetric operator is identical to the odd-symmetric one, and thus may be viewed as a quadrature energy computation!

Given these similarities between the mixed-order derivative algorithm and the spatio-temporal energy approaches described by Heeger and Grzwywacz & Yuille, we may ask: Is there any advantage to using the mixed-order solution? Both algorithms are based on a set of linear measurements, from which a velocity estimate is computed via least-squares regression. The fundamental difference lies in the choice of spatio-temporal filters. Perhaps the most important advantage of the mixed-order solution is that the choice of directional cosine filters allows the velocity estimate to be computed *analytically*, whereas the other approaches require a numerical optimization procedure to solve for velocity. Another important advantage of the mixed-order approach is that the choice of gradient filters allows us to perform regression that is invariant to changes in the spatial frequency content of the signal.

The Phase-derivative Approach

We discuss one more solution developed in the literature. Fleet and Jepson used quadrature pairs of Gabor filters to compute measures of local phase [29]. Rather than derive optical flow from the conservation of intensities (or prefiltered intensities), they choose to derive it from the conservation of this local phase measurement. They demonstrate that this approach is less sensitive than the standard gradient algorithm to non-motion changes in intensity (eg, a change of illumination), and non-translational motions, such as dilations. This is because, for example, the local phase estimates for a pattern of intensities does not vary if the contrast of the pattern is modified. Some related work has been done by other authors [100, 87].

We give a shortened derivation of the Fleet and Jepson estimator here. The basic algorithm is one-dimensional. Even- and odd-symmetric Gabor filters tuned for a particular orientation are applied to the image, producing responses f_e and f_o. The assumption is that the local phase, defined as $\arctan(f_o/f_e)$, is conserved. The authors consider this to be a constraint only on the velocity component *normal* to the orientation of the underlying Gabor filters. Without loss of generality, we assume that the filters are oriented in the x direction. Then we write:

$$\begin{aligned} \frac{d\arctan(f_o/f_e)}{dt} &= \frac{f_e\frac{df_o}{dt} - f_o\frac{df_e}{dt}}{f_e^2 + f_o^2} \\ &= \frac{(f_e f_{o,x} - f_{e,x} f_o)}{(f_e^2 + f_o^2)} \cdot v + \frac{(f_e f_{o,t} - f_{e,t} f_o)}{(f_e^2 + f_o^2)} \\ &= 0, \end{aligned}$$

where the secondary subscripts t and x indicate partial derivatives. Note that as in the basic gradient case, these constraints could be combined in a least-squares fashion over some local neighborhood.

Solving for v gives an estimator that is quadratic in the linear filter measurements:

$$\hat{v}_x = -\frac{(f_e f_{o,t} - f_{e,t} f_o)}{(f_e f_{o,x} - f_{e,x} f_o)}. \tag{2.27}$$

Fleet and Jepson extended this to two dimensions by fitting an affine model of the velocity field using a least squares combination of the one-dimensional normal constraints from different spatial orientations in a small region of space (a technique that was developed by Waxman and Wohn [100]).

We can show that the solution of equation (2.27) corresponds spectrally to a quadrature energy solution like the mixed-order solution we have developed. First, we can see that the numerator (denominator) contains two terms. One is a product of the output of odd-symmetric operators, and the other a product of outputs of even-symmetric operators. Now, we show that given a particular choice of filters, each of these terms is spectrally equivalent to the numerator (denominator) of the one-dimensional gradient solution given in equation (2.14).

We define the following even and odd filters in the frequency domain:

$$\begin{aligned} G_e(\vec{\omega}) &= G(\vec{\omega})\sqrt{|\omega_x|} \\ G_o(\vec{\omega}) &= -j\,\mathrm{sgn}(\omega_x) G(\vec{\omega})\sqrt{|\omega_x|}, \end{aligned}$$

where $G(\vec{\omega})$ is an arbitrary prefilter and sgn$(()\cdot)$ is the "signum" operator that returns the sign of its argument. Clearly, the two operators are related by the Hilbert transform operator in the ω_x direction. Then, we can make the following spectral correspondences (using, once again, Parseval's rule):

$$\begin{aligned}
\sum f_e f_{o,t} &= \sum |F(\vec{\omega})|^2 G_e(\vec{\omega})[-j\omega_t G_o(\vec{\omega})]^* \\
&= \sum |F(\vec{\omega})|^2 |G(\vec{\omega})|^2 \omega_x \omega_t \\
&= \sum f_x f_t \\
\sum -f_o f_{e,t} &= \sum |F(\vec{\omega})|^2 G_o(\vec{\omega})[-j\omega_t G_e(\vec{\omega})]^* \\
&= \sum |F(\vec{\omega})|^2 |G(\vec{\omega})|^2 \omega_x \omega_t \\
&= \sum f_x f_t
\end{aligned}$$

where f_x and f_t correspond to derivatives computed with the prefilter $G(\vec{\omega})$. Thus the two numerator terms are spectrally equivalent to the numerator of the basic gradient solution. Since the two terms are computed from operators of opposite symmetry, they act as quadrature pairs. The denominator terms are of the same form, and will also correspond spectrally to the denominator of equation (2.14), f_x^2. Thus, the Fleet and Jepson solution is spectrally equivalent to a quadrature energy solution, and with this particular choice of filters is spectrally equivalent to the mixed-order derivative solution. As with the equivalences shown in previous section, this only holds in the case of global summation. With local summations, the equivalence is only approximate.

2.4 Summary

We have described a set of local approaches to motion estimation. In the process of doing this we have accomplished two things: 1) we have shown that these approaches are very closely related and may be made spectrally equivalent through a particular choice of linear operators, and 2) we have developed an efficient general algorithm based on a combination of first and second-order *directional cosine* filters. This particular choice of filters is important, as it produces a velocity estimate that is free of systematic errors due to spatial frequency content.

Unfortunately, the algorithm developed in this chapter still suffers from the problems mentioned in section 1.2. The next two chapters attempt to address these failures. In particular, chapter 3 introduces a noise model and a prior probability distribution on velocities, to handle the singularities and lighting changes mentioned in section 1.2. We will also develop a probabilistic coarse-to-fine algorithm to handle the temporal aliasing problem. The output of the algorithm will be a Gaussian distribution over velocities for each position and time in the input imagery.

In chapter 4, we develop a computation that is capable of representing multiple motions in a patch. We return to the concept of directional cosines, and construct an algorithm based on higher-order filters that is capable of representing multiple motions.

Chapter 3

Estimation of Optical Flow

In the previous chapter, we discussed several standard local approaches to the problem of computing optical flow. Although they are generally considered to be unrelated, we showed that they are all quite similar, with the main difference being the choice of linear measurements that are made. Unfortunately, all of the techniques suffer from the basic problems mentioned in section 1.2: intensity singularities, non-motion brightness changes, temporal aliasing, and multiple motions. In this chapter, we must set about the difficult business of attacking these problems.

Consider the first problem: the presence of ambiguous regions. When the images are constant in the region being considered, how can we compute an optical flow vector? Since there is no information in the intensities themselves, we must rely on some sort of *prior* information about image velocities (remember, we are only considering local estimates here). But computing a vector in such a region is misleading: later stages of computation will not be able to discern which vectors are computed from the data and which are guessed from prior knowledge.

Choosing a vector to describe the motion in a blank region constitutes an overcommitment to a solution. Some researchers have advocated the use of "confidence" or "reliability" measures to indicate which optical flow vectors are to be trusted [63, 8, 29]. These are typically compared to a threshold value, and all flow vectors with confidences below the threshold are discarded. This approach does not take advantage of the full information present in the intensity signal. For example, in a region suffering from the aperture problem, the optical flow is only constrained in the direction normal to the local orientation. Thus, we want to represent the normal component of flow, *and* indicate uncertainty about the parallel component of flow.

Recently, some authors have developed approaches that compute two-dimensional covariance matrices which serve as a two-dimensional confidence measure. In particular, Heeger [40] computed a Fischer information matrix by linearizing his regression model about the minimum. Anandan and others [8, 91, 86] have fitted quadratic functions to sum-of-squared-difference error surfaces to estimate confidence, although Barron et. al. [10] note that these confidence measures are often not very reliable. Some authors have also considered autocorrelation or displacement histograms and used these as a sort of distribution over velocity [8, 36]. Szeliski [91] has discussed the use of Bayesian techniques for a variety of problems in low-level vision, including optical flow estimation. Estimation-theoretic approaches to optical flow are beginning to appear [74, 84, 82, 86].

The concepts of integrating prior information and providing directional uncertainty information suggests a Bayesian analysis of the problem. In this chapter, we will extend the basic and mixed-order algorithms of in the previous section to include uncertainty. Previous versions of this work appear in [82].

In our view, this constitutes a fundamental shift in the representation of visual motion. Whereas we previously computed vector-field estimates of the image velocity, we will now compute *probability distributions* over the set of all possible image velocity vectors at a given spatio-temporal location. That is, we will be computing a function $E_v(v_x, v_y; x, y, t)$ that represents the motion of the imagery.

Viewing the problem probabilistically has many advantages:

1. It produces useful extensions of the standard quadratic gradient techniques for computing optical flow, including an automatic gain control mechanism, and the incorporation of a prior distribution on the velocity field.

2. It provides (two-dimensional) confidence information, allowing later stages of processing to tailor their use of the velocity estimates according to the shape of the distribution. For example, it enables the development of algorithms to combine information recursively over time, or to combine information over space to estimate higher-order motion parameters such as rigid body translations.

3. It provides a framework for "sensor fusion", in which image velocity estimates must be combined with information derived from other uncertain sources.

4. As we will discuss in chapter 5, a distributed representation is appropriate for modeling biological motion processing.

Once we have a probabilistic formulation of the problem, we introduce a coarse-to-fine mechanism to handle the problem of temporal aliasing. The remainder of the chapter discusses the implementation of the algorithms and provides examples of velocity field estimation and error analysis on synthetic and real image sequences. These results demonstrate an improvement in performance when compared to those of a recent comparison study of optical flow techniques [10].

3.1 Probabilistic Modeling

Our goal is to compute an expression for the probability of the image velocity conditional on measurements made from the image sequence. We develop this representation by introducing an uncertainty model to characterize deviations from the standard gradient constraint equation. Recall the original differential constraint on optical flow as given in equation (2.1):

$$\vec{f}_s \cdot \vec{v} + f_t = 0$$

Each of the quantities in this equation is an idealization. First, we do not have the actual spatial or temporal derivatives, but *measurements* of these derivatives that are corrupted by errors due to image noise, filter inaccuracies, quantization, etc. Second, the equation is a constraint on the optical flow, but we are interested in estimating the *motion field*. As explained in the introduction, these two quantities often differ because variations in image intensities can be caused by overall changes in brightness, highlights, or changes in the three-dimensional orientation of surfaces [46].

We can make these idealizations explicit by introducing a set of additive random variables. We define $\tilde{v}$ as the optical flow, and $\vec{v}$ as the actual velocity field. Then we describe the difference between these using a random variable, $\vec{n}_1$,

$$\tilde{v} = \vec{v} + \vec{n}_1$$

Similarly, let $\tilde{f}_t$ be the actual temporal derivative, and f_t the measured derivative. Then, we write

$$f_t = \tilde{f}_t + n_2$$

with n_2 a random variable characterizing the uncertainty in this measurement relative to the true derivative. And similarly,

$$\vec{f}_s = \tilde{f}_s + \vec{n}_3$$

is the measured spatial derivative.

Now the gradient constraint applies to the actual measurements, and the optical flow vector, and so we may write:

$$\begin{aligned} 0 &= \tilde{f}_s \cdot \tilde{v} + \tilde{f}_t \\ &= (\vec{f}_s - \vec{n}_3) \cdot (\vec{v} - \vec{n}_1) + f_t - n_2 \\ \Rightarrow \vec{f}_s \cdot \vec{v} + f_t &= \vec{f}_s \cdot \vec{n}_1 + \vec{v} \cdot \vec{n}_3 - \vec{n}_3 \vec{n}_1 + n_2. \end{aligned} \qquad (3.1)$$

This equation gives us a probabilistic relationship between the image *motion field* and the *measurements* of spatio-temporal gradient. It accounts for errors in our derivative measurements, and for deviations of the velocity field from the optical flow. But it assumes that the underlying optical flow constraint is valid. In particular, we can expect that the multiple motions problem mentioned in section 1.2 will still cause trouble.

In order to make use of this formulation, we must characterize the random variables n_i in these definitions. It is desirable to choose these probability distributions such that $\vec{v}$ may be estimated analytically (as opposed to numerically). Thus, we will assume that reliable estimates of the spatial derivatives are available: thus we may neglect the random variable $\vec{n}_3$. We will further assume that we may characterize the remaining random variables with independent zero-mean Gaussian distributions. It would be most unlikely that the actual uncertainties would have this form, but as with many models in the estimation literature, we may be able to account for a large part of the uncertainty in this manner.

Given these assumptions of independent zero-mean Gaussian noise sources, the right side of equation (3.1) is a zero-mean Gaussian random variable with variance equal to $\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2$, where Λ_1 and Λ_2 are a covariance matrix and a variance corresponding to $\vec{n}_1$ and n_2, respectively. We interpret the equation as providing a conditional probability expression:

$$\mathcal{P}(f_t \mid \vec{v}, \vec{f}_s) = \exp\left\{-\tfrac{1}{2}(\vec{f}_s \cdot \vec{v} + f_t)(\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2)^{-1}(\vec{f}_s \cdot \vec{v} + f_t)\right\}$$

In order to write down the desired conditional probability, we can use Bayes' rule to switch the order of the arguments:

$$\mathcal{P}(\vec{v} \mid \vec{f}_s, f_t) = \frac{\mathcal{P}(f_t \mid \vec{v}, \vec{f}_s) \cdot \mathcal{P}(\vec{v})}{\mathcal{P}(f_t)}.$$

For the prior distribution $\mathcal{P}(\vec{v})$, we choose a zero-mean Gaussian with covariance Λ_p. The denominator, $\mathcal{P}(f_t)$, is only present for normalization purposes and doesn't affect the relative probabilities. The resulting distribution is Gaussian:

$$\begin{aligned}
\mathcal{P}(\vec{v} \mid \vec{f}_s, f_t) &\propto \exp\left\{-\tfrac{1}{2}(\vec{f}_s \cdot \vec{v} + f_t)^T(\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2)^{-1}(\vec{f}_s \cdot \vec{v} + f_t)\right\} \exp\left\{-\tfrac{1}{2}\vec{v}^T \Lambda_p^{-1} \vec{v}\right\} \\
&= \exp\left\{-\tfrac{1}{2}\vec{v}^T \left[\vec{f}_s(\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2)^{-1}\vec{f}_s^T + \Lambda_p^{-1}\right]\vec{v}\right. && (3.2) \\
&\qquad - f_t(\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2)^{-1}\vec{f}_s^T \vec{v} && (3.3) \\
&\qquad \left. -\tfrac{1}{2} f_t(\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2) f_t\right\} && (3.4) \\
&= \exp\left\{-\tfrac{1}{2}(\mu_{\vec{v}} - \vec{v})^T \Lambda_{\vec{v}}^{-1}(\mu_{\vec{v}} - \vec{v})\right\}. && (3.5)
\end{aligned}$$

The covariance matrix, $\Lambda_{\vec{v}}$, and mean vector, $\mu_{\vec{v}}$ may be derived using standard techniques (i.e., completing the square in the exponent):

$$\begin{aligned}
\Lambda_{\vec{v}} &= \left[\vec{f}_s(\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2)^{-1}\vec{f}_s^T + \Lambda_p^{-1}\right]^{-1} \\
\mu_{\vec{v}} &= -\Lambda_{\vec{v}}\vec{f}_s(\vec{f}_s^T \Lambda_1 \vec{f}_s + \Lambda_2)^{-1} f_t.
\end{aligned}$$

The advantage of the Gaussian form is that it is parameterized by these two quantities that are computed in analytic form from the derivative measurements. [1]

If we choose Λ_1 to be a diagonal matrix, with diagonal entry λ_1, and write the scalar variance of n_2 as $\lambda_2 \equiv \Lambda_2$, then we can write this as:

$$\begin{aligned} \Lambda_{\vec{v}} &= \left[\frac{\mathbf{M}}{\left(\lambda_1 \|\vec{f}_s\|^2 + \lambda_2\right)} + \Lambda_p^{-1} \right]^{-1} \\ \mu_{\vec{v}} &= -\Lambda_{\vec{v}} \cdot \frac{\vec{b}}{\left(\lambda_1 \|\vec{f}_s\|^2 + \lambda_2\right)}. \end{aligned} \tag{3.6}$$

where matrix $\mathbf{M}$ and vector $\vec{b}$ are defined as in equation (2.6), but without the summations. Note that multiplying Λ_p, λ_1 and λ_2 by a scale factor will not affect the mean, $\mu_{\vec{v}}$ of the distribution (although it *will* scale the variance).

The maximum likelihood estimate (MLE) is simply the mean, $\mu_{\vec{v}}$, since the distribution is Gaussian. This solution is very similar to that specified by equation (2.6). This is not really surprising, since computing the MLE of a Gaussian distribution is equivalent to computing the linear least-squares estimate (LLSE). The differences are that 1) the addition of the prior variance Λ_p ensures the invertibility of the matrix $\mathbf{M}$, and 2) the quadratic derivative terms in $\mathbf{M}$ and $\vec{b}$ are modified by a compressive nonlinearity. That is, for regions with low contrast (i.e., small $\|\vec{f}_s\|^2$), the λ_2 term dominates the divisor of $\mathbf{M}$. For high-contrast regions, the $\lambda_1 \|\vec{f}_s\|^2$ term will normalize the magnitude of the quadratic terms in $\mathbf{M}$.

This seems intuitively reasonable: When the contrast (SNR) of the signal is low, an increase in contrast should increase our certainty of the velocity estimate. But as the contrast increases above the noise level of the signal, the certainty should asymptotically reach some maximum value rather than continuing to rise quadratically. The noise term n_2 accounts for errors in the derivative measurements. At low signal amplitudes, these will be the dominant source of error. The term $\vec{n}_1$ accounts for failures of the constraint equation. At high contrasts, these will be the dominant source of error. The nonlinearity is illustrated in figure 3-1, where we have plotted the trace of the inverse covariance $\Lambda_{\vec{v}}^{-1}$ (i.e., the certainty of the estimate) as a function of contrast, $\|\vec{f}_s\|^2$.

The solution described thus far computes velocity for one point in isolation. As described in section 2.1, there will be a measurement singularity and we may therefore only compute the component of flow normal to the local orientation. In the solution above, the mean will be (approximately) the *normal* flow vector, and the width of these distributions in the direction perpendicular to the normal direction will be determined by Λ_p. The variance in the normal

[1]If we incorporate the additional noise term $\vec{n}_3$ in equation (3.1), the noise model is more general, but the solution is no longer Gaussian.

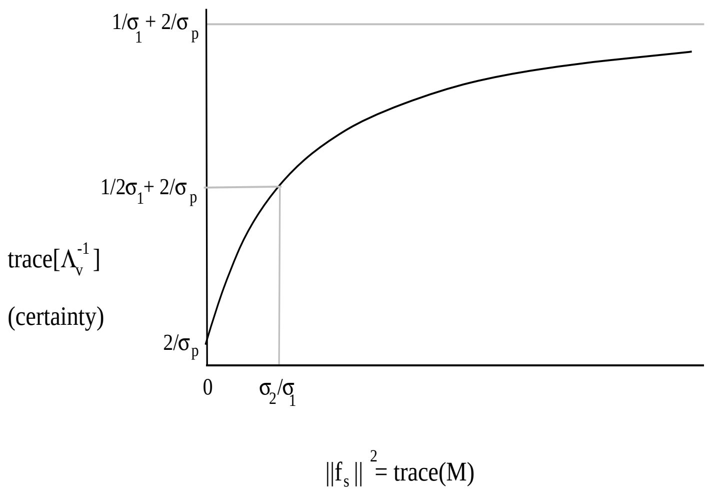


Figure 3-1: Plot of the nonlinearity that operates on the quadratic energy measurements in the solution given in equation (3.6).

direction will be determined by both Λ_p and the trace of $\mathbf{M}$ (i.e., the sum of the squared magnitudes of the spatial derivatives).

If normal flow (along with variance information) does not provide a satisfactory input for the next stage of processing, then we can combine information in small neighborhoods (as in equation (2.4)). We now need an uncertainty model for the entire neighborhood of points. The simplest assumption is that the noise at each point in the neighborhood is independent. In practice this will not be correct, since, for example, a portion of the uncertainty in the derivative measurements is due to noise in the image intensities. In addition, the velocity field uncertainty (characterized by $\vec{n}_3$) will also contain significant correlation between neighboring points.

Nevertheless, as a first approximation, if we treat the uncertainties as pointwise independent, then the resulting mean and variance are:

$$\Lambda_{\vec{v}} = \left[\sum_i \frac{w_i \mathbf{M}_i}{\left(\lambda_1 \|\vec{f}_s(x_i, y_i, t)\|^2 + \lambda_2\right)} + \Lambda_p^{-1} \right]^{-1} \tag{3.7}$$

$$\mu_{\vec{v}} = -\Lambda_{\vec{v}} \cdot \sum_i \frac{w_i \vec{b}_i}{\left(\lambda_1 \|\vec{f}_s(x_i, y_i, t)\|^2 + \lambda_2\right)}, \tag{3.8}$$

where, as before, w_i is a weighting function over the patch, with the points in the patch indexed by i. Here, the effect of the nonlinearity on the combination of information over the patch is to provide a type of gain control mechanism. If we ignore λ_2, the solution above normalizes the information, equalizing the contribution from each point in the neighborhood by the magnitude of the spatial gradient. We will refer to this in later sections as the *basic* solution.

In the solution above, we are combining information over fixed size patches, using a fixed

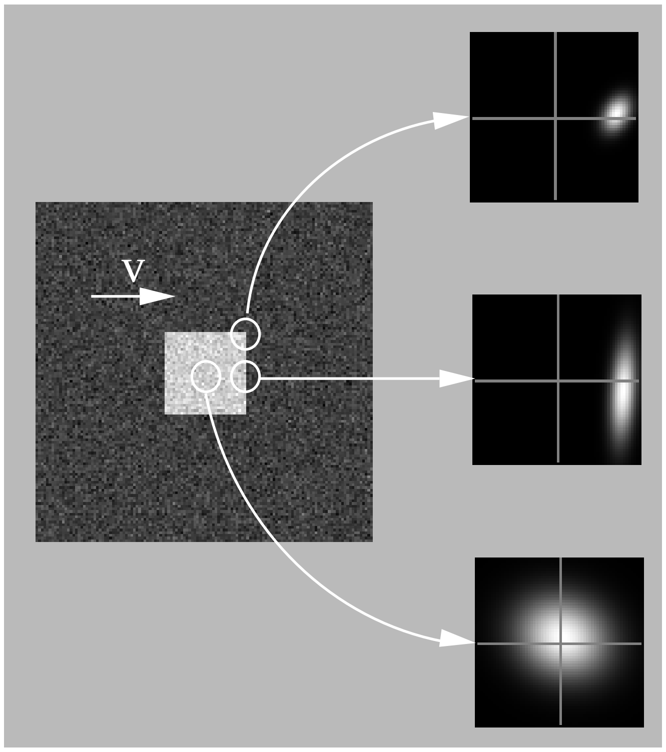


Figure 3-2: Probability distributions computed by the model in different regions of a moving square image. A small amount of noise was added to the image sequence.

weighting function. An adaptive version of this algorithm could proceed by blurring over larger and larger regions until the magnitude of the variance (determinant of the variance matrix) was below some threshold. Since the variance matrix $\Lambda_{\vec{v}}$ describes a two-dimensional shape, this could even be done directionally, averaging pixels which lie in the direction of maximal variance until the variance in this direction was below a threshold.

To illustrate the solution given in equation (3.8), we consider the response to a moving square as we did in figure 1-3. We have added a small amount of Gaussian-distributed white noise. Figure 3-2 shows one frame of the input image, along with the resulting distributions near the corner, on a side, and in the center. In the corner, the output is a fairly narrow distribution centered near the correct velocity. The error in the mean is due to the noise in the input. On the side, the ambiguity of the motion along the edge (i.e., the aperture problem) is indicated by the elongated shape of the distribution. In the center, the motion is completely ambiguous and the resulting distribution corresponds almost entirely to the prior. We also show the response for a low-contrast moving square, with the same amount of Gaussian noise, in figure 3-3. Note that the velocity distribution corresponding to the corner is now substantially broader, as is that of the edge.

We can also write a probabilistic version of the mixed-order solution of equation (2.12).

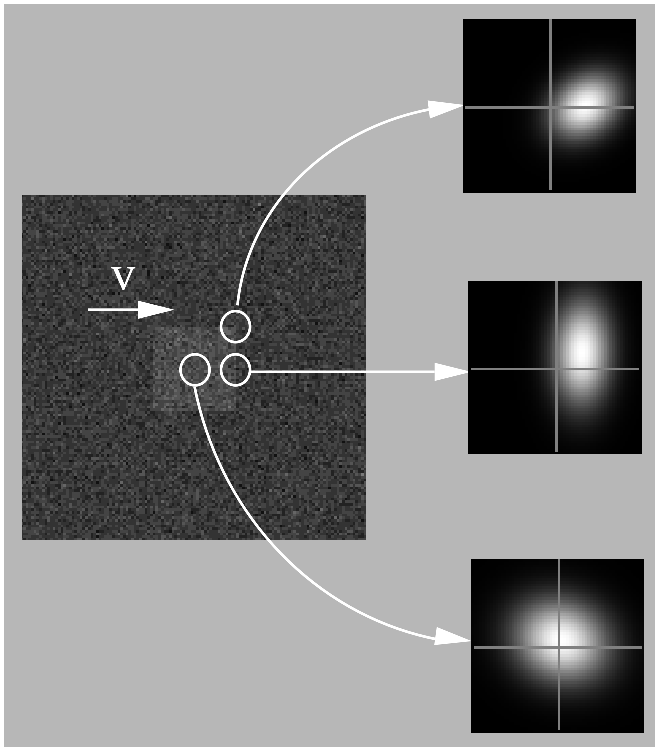


Figure 3-3: Probability distributions computed by the model in different regions of a low-contrast moving square image. The noise added to the sequence was of the same amplitude as that in figure 3-2.

We rewrite that equation with uncertainty terms as:

$$\vec{d}(f) + \mathbf{C}(f) \cdot \vec{v} = \mathbf{C}(f) \cdot \vec{n}_1 + \vec{n}_2,$$

where

$$\mathbf{C}(f) = \begin{pmatrix} f_x & f_y \\ f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \\ f_{xt} & f_{yt} \end{pmatrix}, \qquad \vec{d}(f) = \begin{pmatrix} f_t \\ f_{xt} \\ f_{yt} \\ f_{tt} \end{pmatrix}.$$

As with the single derivative solution of equation (3.6), the solution is Gaussian, with covariance and mean specified by

$$\Lambda_{\vec{v}} = \left[\mathbf{C}^T(f)\left(\mathbf{C}(f)\Lambda_1\mathbf{C}^T(f) + \Lambda_2\right)^{-1}\mathbf{C}(f) + \Lambda_p^{-1}\right]^{-1} \tag{3.9}$$

$$\mu_{\vec{v}} = -\Lambda_{\vec{v}}\mathbf{C}^T(f)\left[\mathbf{C}(f)\Lambda_1\mathbf{C}^T(f) + \Lambda_2\right]^{-1}\vec{d}(f). \tag{3.10}$$

We assume that Λ_1 and Λ_2 are multiples of the identity matrix (2x2 and 4x4, respectively). The solution is nevertheless much less efficient to compute than that of equation (3.6), since it requires the inversion of the 4x4 matrix:

$$\left[\lambda_1\mathbf{C}(f)\mathbf{C}^T(f) + \lambda_2\mathbf{I}\right]$$

The only way around this is to assume that the diagonal terms dominate the matrix $\mathbf{C}(f)\mathbf{C}^T(f)$, in which case the solution becomes

$$\Lambda_{\vec{v}} = \left[\frac{\mathbf{C}^T(f)\mathbf{C}(f)}{\lambda_1\mathrm{trace}\left(\mathbf{C}(f)\mathbf{C}^T(f)\right) + \lambda_2} + \Lambda_p^{-1}\right]^{-1}$$

$$\mu_{\vec{v}} = -\Lambda_{\vec{v}}\frac{\mathbf{C}^T(f)\vec{d}(f)}{\lambda_1\mathrm{trace}\left(\mathbf{C}(f)\mathbf{C}^T(f)\right) + \lambda_2}. \tag{3.11}$$

We will refer to this as the *mixed-order* solution.

Note that

$$\mathrm{trace}\left(\mathbf{C}(f)\mathbf{C}^T(f)\right) = f_x^2 + f_{xx}^2 + f_{xy}^2 + f_{xt}^2 + f_y^2 + f_{xy}^2 + f_{yy}^2 + f_{yt}^2$$

and

$$\mathbf{C}^T(f)\mathbf{C}(f) = \begin{pmatrix} \sum(f_x^2 + f_{xx}^2 + f_{xy}^2 + f_{xt}^2) & \sum(f_xf_y + f_{xx}f_{xy} + f_{xy}f_{yy} + f_{xt}f_{yt}) \\ \sum(f_xf_y + f_{xx}f_{xy} + f_{xy}f_{yy} + f_{xt}f_{yt}) & \sum(f_y^2 + f_{yy}^2 + f_{xy}^2 f_{yt}^2) \end{pmatrix}$$

$$\mathbf{C}^T(f)\vec{d}(f) = \begin{pmatrix} \sum(f_xf_t + f_{xx}f_{xt} + f_{xy}f_{yt} + f_{xt}f_{tt}) \\ \sum(f_yf_t + f_{yy}f_{yt} + f_{xy}f_{xt} + f_{yt}f_{tt}) \end{pmatrix},$$

which are identical to the definitions of $\mathbf{M}$ and $\vec{b}$ in equation (2.12). Thus, as with the basic gradient solution, the uncertainty model modifies the mixed-order solution by incorporating a non-linear gain-control mechanism.

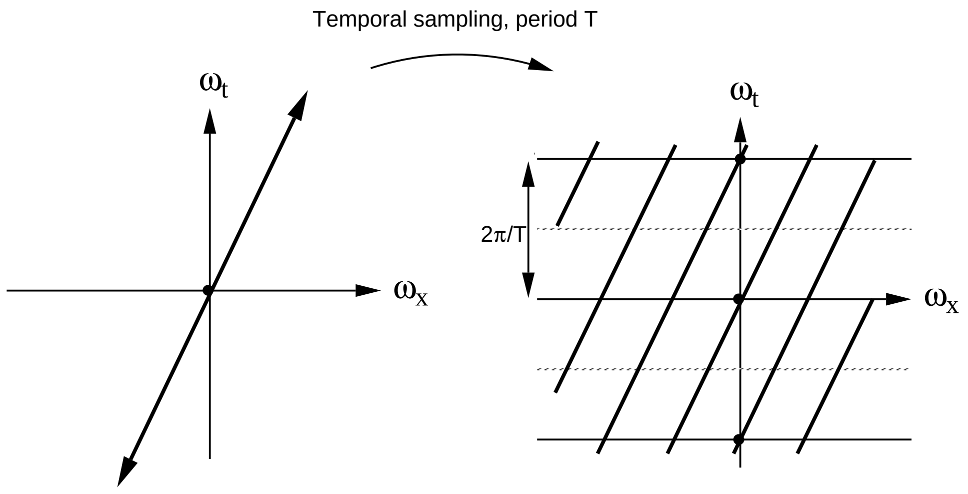


Figure 3-4: Illustration of the temporal aliasing problem. On the left is an idealized depiction of the power spectrum of a one-dimensional pattern translating at speed v. The power spectrum is distributed along the heavy line, which has a slope of v. Temporally sampling the signal causes a replication of its power spectrum in the frequency domain at temporal frequency intervals of $2\pi/T$. When the velocity of the signal is high, the replicas of the spectrum will interfere with the filter (gradient) measurements.

3.2 Temporal Aliasing

In section 1.2 of the introduction, we mentioned temporal aliasing as a typical problem in computing the optical flow of real image sequences. Humans can easily see temporal aliasing. The classic example is the wheel of a covered wagon in a Western movie: when the wheel rotates at the right speed, its direction of rotation appears to reverse. This phenomenon is not a property of the eye, but of the sparseness of the temporal sampling introduced by the camera.

Temporal aliasing can cause problems for any type of motion algorithm. In the case of algorithms based on matching, it is manifested as the problem of "false matches", as was illustrated in figure 1-4. For filtering algorithms, the effect is easiest to see in the frequency domain. Consider a one-dimensional signal that is moving at a constant velocity. As discussed in section 2, the power spectrum of this signal lies on a line through the origin. We assume that the spatial sampling is dense enough to avoid aliasing (i.e., the imagery is spatially band-limited, and the sampling frequency is above the Nyquist limit). The temporal sampling of the imagery causes a replication of the signal spectrum at temporal frequency intervals of $2\pi/T$ radians, where T is the time between frames. This is illustrated in figure 3-4. It is easy to see that these replicated spectra could confuse a motion estimation algorithm.

Eliminating Temporal Aliasing: Tracking and Warping

An important observation concerning this type of temporal aliasing is that it affects the higher spatial frequencies of an image. In particular, for a fixed global velocity, those spatial frequencies moving more than half of their period per frame will be aliased, but lower spatial frequencies will not.

This suggests a simple, but effective approach for avoiding the problem of temporal aliasing. We can estimate the velocity using a low-frequency (or coarse-scale) prefilter that ignores the (potentially aliased) higher frequency content. Note that this prefilter will tend to be quite large in spatial extent, in inverse proportion to its small spatial-frequency extent. Thus these velocity estimates will correspond to the average velocity over a large region of the image.

Given imagery that contains only a single global motion, or a motion field that varies slowly, we could stop our computation at this point. But in typical scenes, slowly varying motion fields are rare. Although a single velocity may account for much of the motion in a large region, it is likely that subportions are moving at least slightly differently. The low-frequency subband estimates will be unable to capture these local variations.

In order to get better estimates of *local* velocity, higher-frequency bands must be used, with spatially smaller filters. What we would like to do is to use the coarse motion estimate to "undo" the motion, roughly stabilizing the position of the image over time. Then higher frequency filters can be used to extract local perturbations to this large-scale motion. That is, we can use higher frequency filters to estimate optical flow on the warped sequence, and this "optical flow correction" may then be composed with the previously computed optical flow to give a new optical flow estimate. This correction process may be repeated for finer and finer scales.

There are two mechanisms that one could imagine using to stabilize the image. In an interactive setting (i.e., a biological or robot visual system), the sensors can be moved so as to track a given point or object in the scene. This action reduces the image velocity of the object to zero. It is well-known that human beings do this when observing moving imagery.

Alternatively, in image-processing situations, where the image-gathering has already occurred, we can "warp" a spatially and temporally localized region of the image content in a direction opposite to the computed motion. For our purposes, we compute the warped image sequence:

$$\mathcal{W}\{f, v\}(x, y, t + \Delta t) = f(x - v_x \Delta t, y - v_y \Delta t, t + \Delta t),$$

where (v_x, v_y) is the warp vector field corresponding to the velocity estimated from the coarser scale measurements. Note that the warping only need be done over a range of Δt that covers the temporal extent of the derivative filters that will be applied.

We will concentrate on the warping approach here, although many of the observations

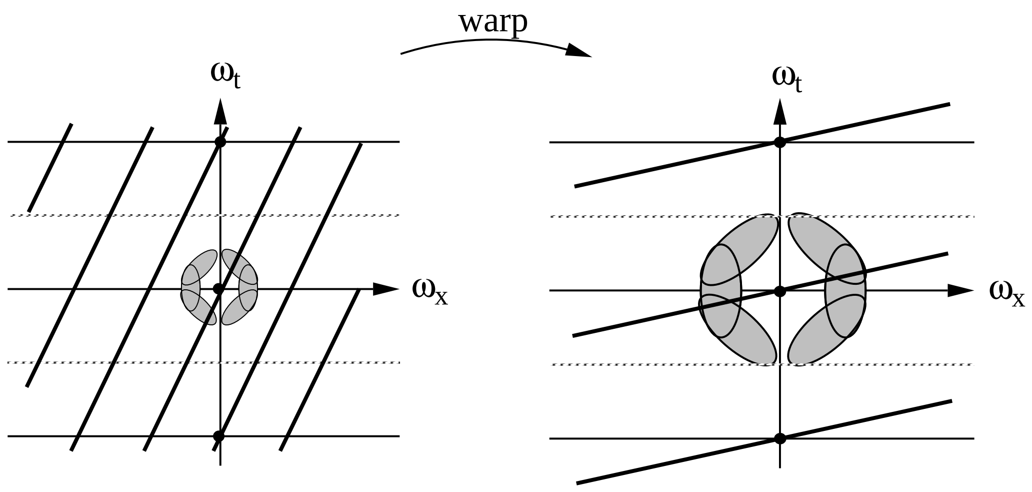


Figure 3-5: Illustration of the coarse-to-fine approach for eliminating temporal aliasing. On the left is an idealized illustration of the (aliased) power spectrum of the signal. A low frequency subband is used to estimate the motion of this signal. These estimates are then used to "undo" the motion, leaving a smaller, unaliased residual motion (shown on the right – note that the spectrum lies on a line of smaller slope). This motion may then be estimated using higher frequency filters.

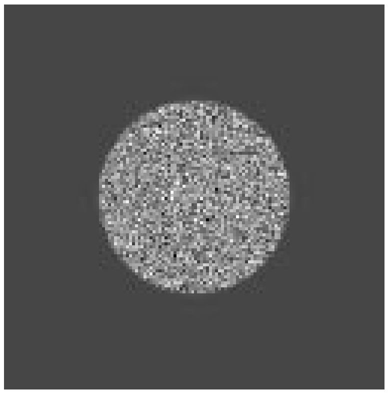
apply to the tracking case as well. The warping procedure may be applied recursively to higher frequency subbands. This "coarse-to-fine" estimation process is illustrated in figure 3.2. This type of multi-scale "warping" approach has been suggested and used by a number of authors [54, 71, 23, 8, 12].

Coarse-to-Fine Algorithms

As described above, in order to generate estimates at different scales, we can apply the differential algorithm to lowpass prefilters of different bandwidth. To illustrate the effectiveness of this technique, consider a simple test pattern containing a disk containing high-frequency texture moving at a fairly high velocity. This is illustrated in figure 3.2. A local operator attempting to compute motion in the center of the disk would fail. But the multi-scale algorithm succeeds since the coarse scale motion of the disc allows the algorithm to "lock onto" the correct motion.

As we have described, a "coarse-to-fine" algorithm can be used to handle problems of temporal aliasing. We noted that it also may be viewed as a technique for combining information from different spatial scales. It is also a technique for imposing a prior smoothness constraint (see, for example, [91, 20]). This basic technique does, however, have a serious drawback. If the coarse-scale estimates are incorrect, then the fine-scale estimates will have no chance of correcting the errors.

To fix this, we must have knowledge of the error in the coarse-scale estimates. Since we



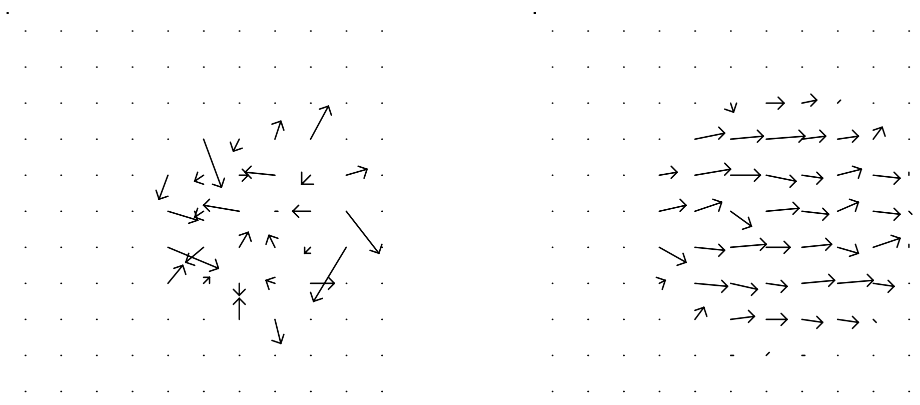


Figure 3-6: Top image is the stimulus, a rapidly moving disc with fine-scale texture on it. Bottom/left: Optical flow computed, direct gradient algorithm. Bottom/right: Optical flow computed using coarse-to-fine gradient algorithm. The single dots correspond to optical flow vectors of length zero.

are working in a probabilistic framework, and we have information describing the uncertainty of our measurements, we may use this information to properly combine the information from scale to scale. We define a *state evolution* equation with respect to scale:

$$\vec{v}(l+1) = E(l)\vec{v}(l) + \vec{n}_0(l), \qquad \vec{n}_0(l) \sim N(0, \Lambda_0)$$

where l is an index for scale (larger values of l correspond to finer scale), $E(l)$ is the linear interpolation operator used to extend a coarse scale flow field to finer resolution, and $\vec{n_0}$ is a random variable corresponding to the certainty of the prediction of the fine-scale motion from the coarse-scale motion. We assume that the $\vec{n_0}(l)$ are independent, zero-mean, and normally distributed. Implicitly, we are imposing a sort of fractal model on the velocity field. This type of scale-to-scale Markov relationship has been explored in an estimation context in [20, 11].

We also define the *measurement* equation:

$$-f_t(l) = \vec{f}_s(l) \cdot \vec{v}(l) + (n_2 + \vec{f}_s(l) \cdot \vec{n}_1)$$

as in section 3.1. We will assume, as before, that the random variables are zero-mean, independent and normally distributed. Remember that this equation is initially derived from the total derivative constraint for optical flow. This equation is a bit different than the measurement equation used in most estimation contexts. Here, the linear operator relating the quantity to be estimated to the measurement f_t is *also* a measurement.

Given these two equations, we may write down the optimal estimator for $\vec{v}(l+1)$, the velocity at the fine scale, given an estimate for the velocity at the previous coarse scale, $\mu_{\vec{v}}(l)$, and a set of fine scale (gradient) measurements. The solution is in the form of a standard Kalman filter [33], but with the time variable replaced by the *scale*, l:

$$\begin{aligned}
\mu_{\vec{v}}(l+1) &= E(l)\mu_{\vec{v}}(l) + K(l+1)\nu(l+1) \\
\Lambda(l+1) &= \Lambda'(l+1) - K(l+1)\vec{f}_s^T(l+1)\Lambda'(l+1) \\
K(l+1) &= \Lambda'(l+1)\vec{f}_s(l+1) \cdot \left[\vec{f}_s^T(l+1)\left(\Lambda'(l+1) + \Lambda_1\right)\vec{f}_s(l+1) + \Lambda_2\right]^{-1} \\
\nu(l+1) &= -f_t(l+1) - \vec{f}_s^T(l+1)E(l)\mu_{\vec{v}}(l) \\
\Lambda'(l+1) &= E(l)\Lambda(l)E(l)^T + \Lambda_0
\end{aligned}$$

Here, $\nu(l)$ corresponds to an *innovations* process. The innovations process represents the new information contributed by the measurements at level l.

The problem with the equations given above is that we cannot compute the derivative measurements at scale l without making use of the velocity estimate at scale $l-1$, due to the temporal aliasing problem. In order to avoid this problem, we must write $\nu(l)$ in terms of derivatives of the warped sequence. That is, expanding around a time t_0, we write:

$$\nu(l+1) \;=\; -f_t(l+1) - \vec{f}_s(l+1) \cdot E(l)\mu_{\vec{v}}(l)$$

$$\begin{aligned} &= -\frac{\partial}{\partial t} f\left(\vec{x} + (t_0 + t)E(l)\mu_{\vec{v}}(l), t_0 + t\right) \\ &\approx -\frac{\partial}{\partial t} \mathcal{W}\{f(l+1), E(l)\mu_{\vec{v}}(l)\}(x, y, t_0) \end{aligned}$$

Thus, the innovations process is computed as the temporal derivative of the the image at scale $l+1$, after it has been warped with the interpolated flow field from scale l. In order to make the solution computationally feasible, we ignore the off-diagonal elements in $\Lambda'(l+1)$ (i.e., the correlations between adjacent interpolated flow vectors).

Now the Kalman solution may be put into the alternative "update" form by use of the following matrix identity [33]:

$$\left[B - BC^T(CBC^T + A)^{-1}CB\right]^{-1} = B^{-1} + C^TA^{-1}C.$$

The left side corresponds to the inverse of the updated covariance matrix given in the Kalman equations above:

$$\begin{aligned} \Lambda(l+1) &= \left[\Lambda'(l+1)^{-1} + \vec{f}_s(\vec{f}_s^T\Lambda_1\vec{f}_s + \Lambda_2)^{-1}\vec{f}_s^T\right]^{-1} \\ &= \left[\Lambda'(l+1)^{-1} + \frac{\mathbf{M}}{\left(\lambda_1\|\vec{f}_s\|^2 + \lambda_2\right)}\right]^{-1}, \end{aligned} \tag{3.12}$$

Similarly, we may rewrite the updated mean vector as:

$$\begin{aligned} \mu_{\vec{v}}(l+1) &= E(l)\mu_{\vec{v}}(l) + \Lambda(l+1)\vec{f}_s(\vec{f}_s^T\Lambda_1 + \Lambda_2)^{-1}\nu(l+1) \\ &= E(l)\mu_{\vec{v}}(l) + \Lambda(l+1)\frac{\vec{b}'}{\left(\lambda_1\|\vec{f}_s\|^2 + \lambda_2\right)}, \end{aligned} \tag{3.13}$$

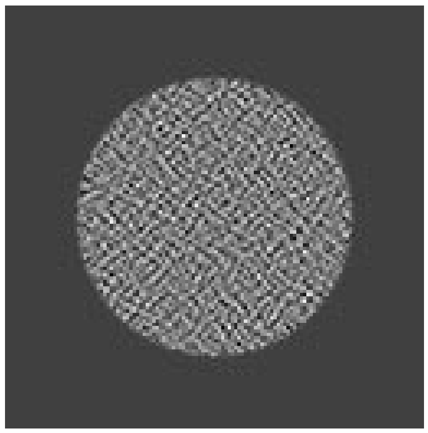
where the vector $\vec{b}'$ is defined by

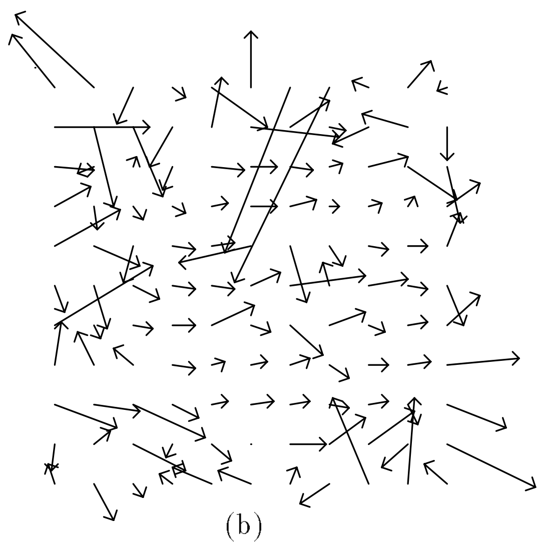
$$\vec{b}' = \vec{f}_s\nu(l+1).$$

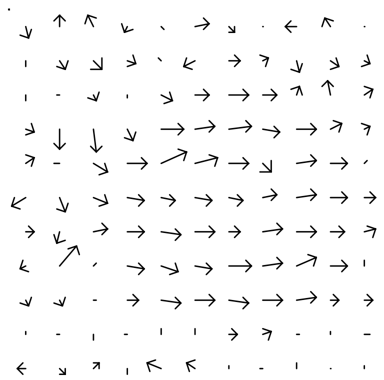
These mean and covariance expressions are the same as those of equation (3.6) except that: 1) the prior covariance Λ_p has been replaced by $\Lambda'(l+1)$, 2) the vector $\vec{b}$ has been replaced by $\vec{b}'$, which is computed in the same manner but using the *warped* temporal derivative measurements, and 3) the mean $\mu_{\vec{v}}(l+1)$ is augmented by the interpolated estimate from the previous scale.

Figure 3.2 illustrates the effectiveness of this "Kalman filter over scale". The stimulus is a slowly moving textured disk, *with noise added.* The ordinary coarse-to-fine gradient algorithm gives terrible results, because the noise leads to large errors in the coarse-scale velocity estimates that cannot be corrected at finer scales. The covariance-propagating version defined by equations (3.12) and (3.13) produces better estimates (i.e., the mean vectors are closer to the actual flow), and the covariance information accurately indicates the more uncertain vectors.

Given that the derivative measurements will fail when the image velocity is too high, a more sophisticated version of this algorithm could "prune" the tree during the coarse-to-fine operation. That is, we can terminate the recursion at a given location (x, y, t) and level l if the interpolated covariance estimate from the previous scale, $\Lambda'(l)$ is too large.


(a)


(b)


(c)

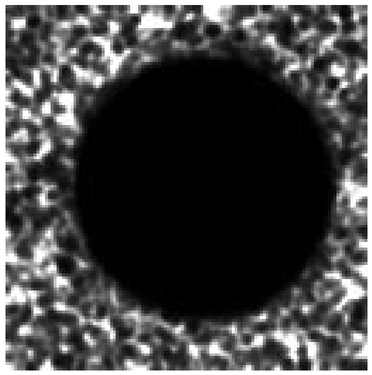

(d)

Figure 3-7: Example using the covariance-propagating coarse-to-fine algorithm. (a) the stimulus, a slowly moving disc with fine-scale texture on it and with added Gaussian white noise. (b) Optical flow computed using standard coarse-to-fine gradient algorithm. (c) Optical flow computed using Kalman-like coarse-to-fine gradient algorithm with covariance propagation. (d) the determinant of the terminal covariance matrix, indicating uncertainty of the estimate.

3.3 Implementation Issues

In this section we will discuss some important issues that arise when implementing the algorithms discussed thus far.

Derivative Filter Design

The design of filters for performing the derivative operation is a difficult problem, and yet many authors do not even describe the filters that they use. The most common choice in the literature is a first-order difference. First we show that simple first order differences are likely to produce poor results for the optical flow problem when applied directly to the input imagery, especially in highly textured regions (i.e., regions with much high-frequency content). Figure 3-8 shows the graph of the Fourier magnitude of the two-point derivative operator $g'(n) = [-1, 1]$. On the same plot is shown the Fourier transform of the derivative of the two-point averaging operator $g(n) = [0.5, 0.5]$, which is computed by multiplying its Fourier magnitude by the function $|\omega|$.

Although the two curves coincide reasonably well for the lowest frequencies, they are clearly far apart at mid or high frequencies. For signals with significant content in this range of frequencies, the derivative will be overestimated. In the context of motion analysis, this could result in an overestimation or an underestimation of velocity. For example, a high spatial-frequency signal moving slowly would result in an overestimated spatial derivative, thus producing an underestimate of velocity. Because of this, these two filters make a poor derivative/prefilter pair, unless the input signal is severely band-limited. We will demonstrate this explicitly in section 3.4.

Some authors have used higher-order difference filters or binomial expansions. These are often combined with lowpass prefilters. We note that it is easier to design the derivative of the prefilter than to design a prefilter and a separate derivative filter. If we assume ideal (sinc-function) interpolation, the true derivative operator is global in extent, so any small FIR filter will be a fairly crude approximation. Rather than choosing a low cutoff frequency for the prefilter in order to render the derivative accurate, we design the prefilter and its derivative simultaneously.

We choose to work in the frequency domain. Let $G(\vec{\omega})$ be the Fourier transform of the prefilter, and $G'(\vec{\omega})$ the Fourier transform of the derivative filter. Then our design method must attempt to meet the following requirements:

1. The derivative filters must be very good approximations to the derivative of the prefilter. That is we would like $|\omega_x G(\vec{\omega})| \approx |G'(\vec{\omega})|$.

2. The prefilter should be a constant-phase filter, preferably even- or odd-symmetric (i.e.,

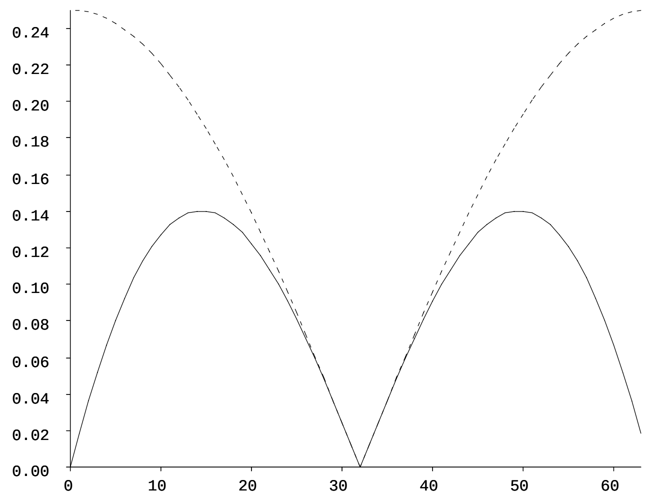


Figure 3-8: Illustration of the two point derivative operator in the frequency domain. Shown are the magnitude of the Fourier transforms over the range $-\pi < \omega\pi$ of a) the derivative operator $[-1, 1]$ (dashed line), and b) the frequency-domain derivative of the two-point averaging operator $[0.5, 0.5]$ (that is, its Fourier magnitude multiplied by $|\omega|$). If these were a perfect derivative/prefilter pair, the curves would coincide.

phase of zero or $\pi/2$).

3. Temporally, the prefilter should be a lowpass filter, to allow sensitivity to low or zero speeds.

4. Spatially, the prefilter can be either bandpass or lowpass, but it should not vary with spatial orientation. That is, it should be rotationally symmetric.

5. For computational efficiency and ease of design, it is preferable that the prefilter be separable. In this case, the derivatives will also be separable, and the design problem will be reduced to one-dimension.

The requirements on the temporal and spatial aspects of the prefilter restrict us to two possible types of symmetry: spherical (or elliptical) and cylindrical.

If we insist on the separability of the filters (item 5), then our search is narrowed considerably. In particular, there is only one choice: a Gaussian. The main drawback of this choice is that we cannot create a set of mixed-order filters that will act exactly as quadrature pairs, as described in section 2.3. According to equation (2.25), the quadrature pair constraint requires that

$$rG_2(r) = G_1(r).$$

It is easy to see that there is no solution to this equation if $G_1(r)$ and $G_2(r)$ are Gaussians. Nevertheless, by choosing these Gaussians with different variances for our first and second derivatives, we can design reasonable approximations.

We have designed two first derivative filters (with their prefilters) and a second derivative filter (with associated first derivative and prefilter). The design criterion was weighted least squares in the frequency domain. The weighting function was intended to mimic the expected frequency content of natural images: $W(\omega) = 1/|\omega|$. With this simple error measure, we can compute solutions analytically and avoid complex optimizations with local minima.

We first fixed the size of the filter, N. Let $G_N(\omega)$ be the Fourier transform of the desired N-tap prefilter, and $G'_N(\omega)$ the Fourier transform of the desired N-tap derivative filter. Then we minimize the following function over σ:

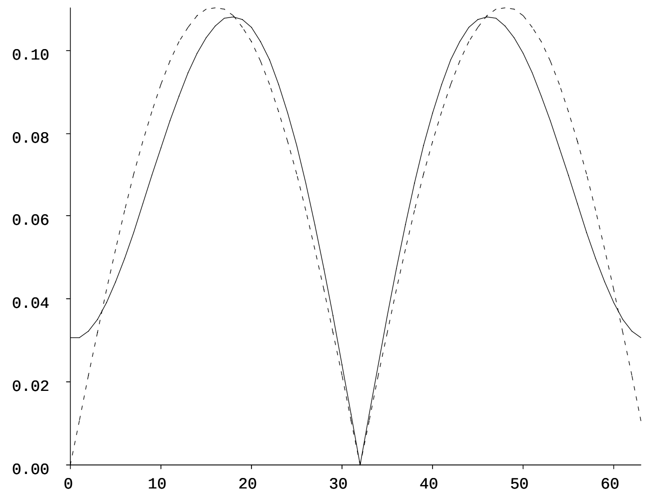
$$E(\sigma; N) = \min_{G'_N(\omega)} \left\{ W(\omega)|G'_N(\omega)| - W(\omega)|\omega| \arg \min_{G_N(\omega)} \left[W(\omega)|G_N(\omega)| - W(\omega)e^{-\omega^2/2\sigma^2} \right]^2 \right\}^2$$

In words, the solution is found as follows. For a Gaussian of standard deviation σ, we analytically compute samples of the Fourier transform at 64 points, from 0 to 2π. Given this ideal filter shape, we found the best (in a weighted least squares sense) prefilter of N taps. We computed the 64 point FFT of this prefilter, multiplied by $-j\omega$, and then found the filter of N taps best approximating this Fourier transform (this is the derivative filter). The parameter σ was chosen so as to minimize the weighted least square error between the prefilter derivative and the derivative filter. The resulting filter kernels and plots of their Fourier transforms are given in figures 3-9 and 3-10. Notice that the three-tap filter is substantially more accurate than the two-tap filter of figure 3-8, and the five-tap filter is substantially better than the three-tap.

Blurring Filters

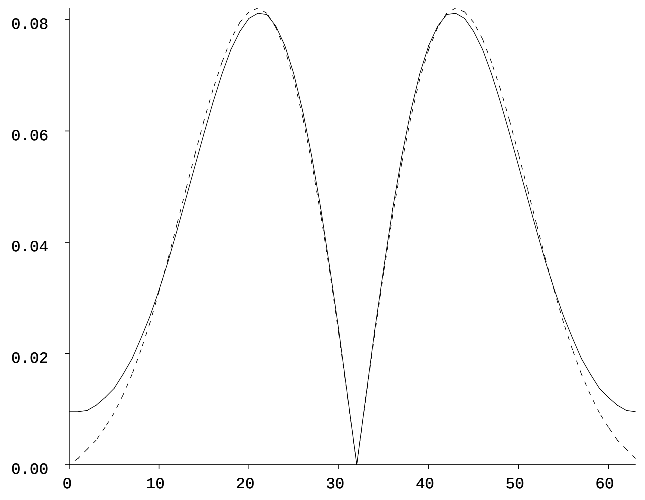
The algorithm also requires averaging over a patch, which is equivalent to applying a lowpass filter. The purpose of the blurring is to eliminate phase cancellations, or to improve the estimates of local spectral power (recall that all of our frequency-domain arguments of chapter 2 are based on integration over the entire image).

We desire this lowpass filter to have all positive weights, since it will be used combine a set of squared constraints and should produce a positive value. There is a tradeoff in choosing the spatial extent of the filter. A large filter will produce better power spectral estimates by combining information over a larger region. But it is also more likely to combine inconsistent motions. The question can only be properly settled given a knowledge of the statistics of the motion of the imagery to be analyzed. We experimented with binomial blurring filters and found that separable application of the kernel $[0.25, 0.5, 0.25]$ with the mixed-order algorithm produced reliable results without over-blurring. The basic algorithm generally required a larger blurring filter: $[0.0625, 0.25, 0.375, 0.25, 0.0625]$.



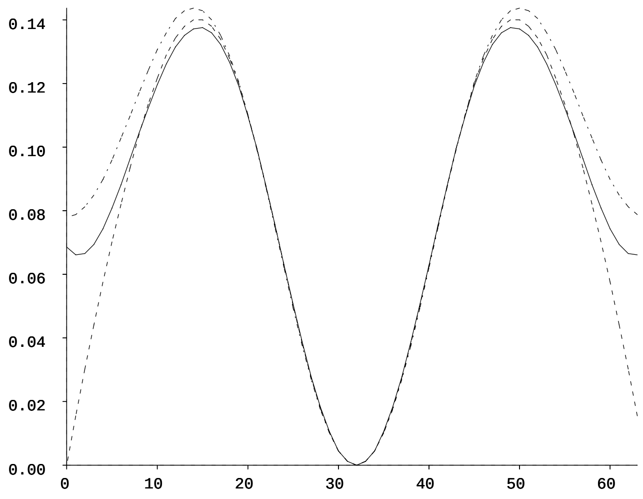
filter	$n=-1$	$n=0$	$n=1$
prefilt	0.230366	0.539269	0.230366
deriv	-0.441419	0.00000	0.441419

Figure 3-9: Illustration of the 3-tap derivative/prefilter pair. See caption of figure 3-8 for explanation. The table contains tap values for the two filters.



filter	$n=-2$	$n=-1$	$n=0$	$n=1$	$n=2$
prefilt	4.504187e-2	0.243908	0.422100	0.243908	4.504187e-2
deriv	-0.108144	-0.269869	0.00000	0.269869	0.108144

Figure 3-10: Illustration of the 5-tap derivative/prefilter pair. See caption of figure 3-9



filter	$n=-2$	$n=-1$	$n=0$	$n=1$	$n=2$
prefilt	3.342604e-2	0.241125	0.450898	0.241125	3.342604e-2
deriv1	-9.186104e-2	-0.307610	0.00000	0.307610	9.186104e-2
deriv2	0.202183	9.181186e-2	-0.587989	9.181186e-2	0.202183

Figure 3-11: Illustration of Fourier magnitudes of a 5-tap prefilter with both its first and second derivatives. Shown are the second derivative filter (dash-dot pattern), the first derivative multiplied by $|\omega|$ (dash pattern), and the prefilter multiplied by $|\omega|^2$ (solid line). This set of filters was used as a quadrature pair to the 3-tap filters of figure 3-9

Multi-scale Warping

In section 3.2, we discussed the implementation of coarse-to-fine algorithms for defeating temporal aliasing. Conceptually, this approach operates by using prefilters of varying bandwidths.

A more efficient technique for generating multi-scale representations is to construct an image pyramid [17], by recursively applying lowpass filtering and subsampling operations. In this case, the images at different scales are also represented at different sampling rates. Assuming the lowpass filter prevents aliasing, the effect of the subsampling in the frequency domain is to stretch the spectrum out. This allows us to use the *same* derivative filters at each scale, rather than designing a whole family of derivative filters at different scales.

The algorithm begins by building a multi-scale pyramid on each frame of the input sequence, and computing the optical flow on the sequence of top level (lowest frequency) images using the computation specified by equation (3.8). An upsampled and interpolated version of this coarse, low-resolution flow field must then be used to warp the sequence of images in the next pyramid level. We used a simple bilinear interpolator in this case, since the optical flow is somewhat smooth due to the blurring operation. Optical flow is then computed on this warped sequence, and this "optical flow correction" was composed with the previously computed optical flow to give a new optical flow estimate. This correction process is repeated for each level of the pyramid until the flow fields are at the resolution of the original image sequence.

The warping equation is fairly unambiguous in the continuous case. But there are many ways in which one can implement a warping algorithm on discretely sampled image data. Consider the task of warping a frame at time t_1 back to time t_0. The primary issues are:

Indexing Should we use the velocity estimate at t_0 or t_1 (or a velocity estimate *between* the frames) as the warp field? Assuming the velocity vectors are in units of pixels/frame, does the velocity estimate at position (x, y, t) correspond to the displacement of intensity at $(x - v_x, y - v_y, t - 1)$ to (x, y, t), or from (x, y, t) to $(x + v_x, y + v_y, t - 1)$?

Order If our filters are several frames long and thus require warping of several frames, should we use the velocity estimates at each of these frames, or just the velocity estimate of the central frame?

Interpolation Given that velocity vector components are typically not multiples of the pixel spacing, how should we interpolate the intensities of the warped images?

We compared several different variants and chose a simple and efficient warping scheme. We assume an odd-length temporal derivative filter of length N_t, and we use a velocity field estimate associated with the center frame. Let $t_c = (N_t - 1)/2$ be the number of this center frame. Since our derivative filters are separable, we apply the spatial portion to N_t frames centered at frame t_c. Let $f'(x,y,t), 0 \leq t < N_t$ be the set of spatially filtered frames. We then combine the temporal portion of the derivative with the warping operation as follows:

$$f'_t(x,y,0) = \sum_{t=0}^{N_t-1} g_t(t) I\left\{ f'\left(x + (t-t_c)\hat{v}_x(x,y,t_c),\ y + (t-t_c)\hat{v}_y(x,y,t_c),\ t\right)\right\}$$

where $\hat{v}$ is the previous estimate of optical flow, and $I\{\cdot\}$ is a bi-cubic spline interpolator used to evaluate f' at fractional-pixel locations.

Boundary Handling

Convolutions are used to compute the derivative filter responses, and to blur the energies. They are also used in coarse-to-fine schemes to construct the multi-resolution image pyramid. Traditionally, convolution boundaries are handled by computing *circular* convolution. That is, the image is treated as one period of a periodic signal. This often produces poor results because it associates the the image content near one boundary with that near the opposite boundary.

There are many alternatives for handling edges. Let $f(n)$ be a one-dimensional signal, indexed by the discrete variable n, with $n = 0$ corresponding to the leftmost sample. Then we define an edge-handling mechanism (for the left edge) by assigning a value for the function f at negative values of n. Several example methods that we have experimented with are:

1. Reflect the image about its edge pixel (or just beyond the edge pixel): $f(-n) = f(n)$ (or $f(-n) = f(n-1)$).
2. Imbed the image in a "sea of zeros": $f(-n) = 0$, for each $n > 0$.
3. Repeat the edge pixel: $f(-n) = f(0)$
4. Reflect and invert (so as to preserve zeroth and first-order continuity): $f(-n) = 2f(0) - f(n)$.
5. Return zero for the convolution inner product whenever the filter kernel overhangs an edge of the image.

For the blurring and pyramid filtering operations we have found that reflection (item 1) is preferable. For the derivative operations, we choose to repeat the edge pixel (item 3).

3.4 Examples

We computed velocity field estimates for a set of synthetic and real image sequences in order to examine the behavior of the basic (first derivative) solution of equation (3.8) and the mixed-order solution given in equation (3.11).

Performance Measures

In cases where we the velocity field is known, we can analyze the errors in our estimates. There are a number of ways to do this. The simplest measure is the squared magnitude of the difference between the correct and estimated flow:

$$E_{\text{mag2}} = |\hat{v} - \vec{v}|^2.$$

where $\vec{v}$ is the actual velocity, and $\hat{v}$ is the estimate. Viewing an image containing these values at each spatial location often provides useful information about the spatial structure of the errors. Errors in optical flow are sometimes reported as a ratio of the error magnitude to magnitude of the actual flow, but this is problematic when the actual flow vectors are small.

Fleet and Jepson [29] used an error criterion based on the unit vector normal to the velocity plane in spatio-temporal frequency:

$$E_{\text{angular}} = \arccos\left[\hat{u}(\hat{v}) \cdot \hat{u}(\vec{v})\right],$$

where $\hat{u}(\cdot)$ is a function producing a three-dimensional unit vector:

$$\hat{u}(\vec{v}) = \frac{1}{\sqrt{|\vec{v}|^2 + 1}} \begin{pmatrix} \vec{v}_x \\ \vec{v}_y \\ 1 \end{pmatrix},$$

and the resulting angular error is reported in units of degrees.

We also define a measure of bias in order to quantify characteristic over- or under- estimation of velocity magnitudes:

$$E_{\text{bias}} = \frac{\vec{v} \cdot (\vec{v} - \hat{v})}{|\vec{v}|}.$$

Positive values of this measure, for example, indicate that the algorithm is overestimating the velocity magnitude.

Mean values of all three of the error measures defined above, collected over single or multiple frames, produce useful statistics for characterizing performance. In addition, one can compute the variance of these quantities.

In situations where we have estimated velocity field covariances, $\Lambda_{\vec{v}}$, as well as means, $\mu_{\vec{v}}$, we can check that the covariance information adequately describes the errors in the flow estimates. The appropriate technique here is to normalize each of the errors according to the covariance information:

$$E_{\text{normalized}} = \sqrt{\left(\vec{v}_{\text{actual}} - \vec{v}_{\text{est}}\right)^T \Lambda_{\vec{v}}^{-1} \left(\vec{v}_{\text{actual}} - \vec{v}_{\text{est}}\right)}.$$

If the flow field errors are exactly modeled by the additive Gaussian noise model, then a histogram of the values of the $E_{\text{normalized}}$ values should be distributed as a two-dimensional

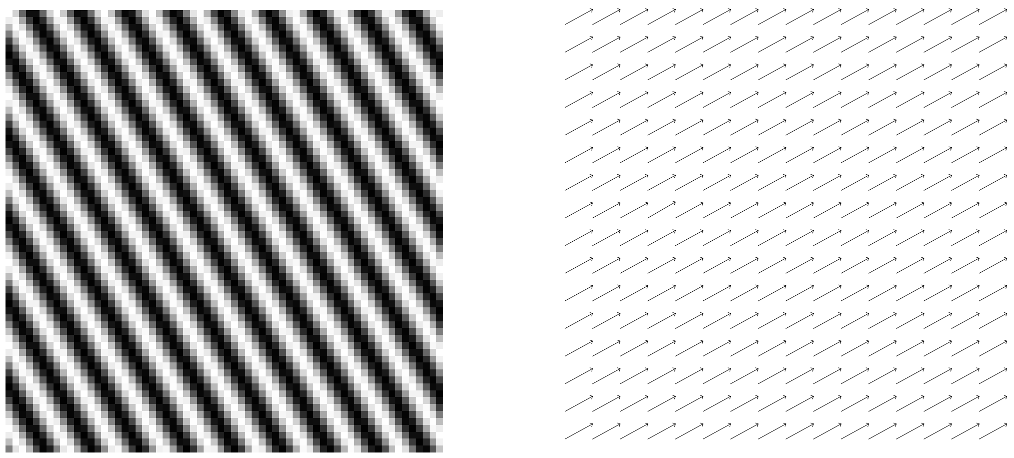


Figure 3-12: One frame of a sinusoidal image sequence, and its computed velocity field.

univariate Gaussian integrated over its angular coordinate:

$$h(x) \propto xe^{-x^2/2}, \qquad x > 0.$$

Translating Synthetic Sequences

We generated a series of very simple synthetic test sequences to study the error behavior of the algorithm. These stimuli involve only translation of the image patterns, and therefore fully obey (modulo intensity quantization noise) the total derivative equation (2.1) for optical flow. Furthermore, since the entire image translates with a single velocity, the combination of information in a neighborhood is fully justified. Thus, these examples are primarily a test of the filters used to measure the derivatives, and the prior probability constraint used to determine a solution when there is an aperture or blank-wall problem. For this section, we used only the single-scale basic gradient algorithm, and we set the noise parameters as $\lambda_1 = 0, \lambda_2 = 1, \lambda_p = 1e - 5$.

The first example is a translating sinusoidal grating:

$$f(x, y, t) = \sin(k\cos(\theta)x + k\sin(\theta)y + vkt),$$

where k is the spatial frequency (or *wavenumber*), v the speed in the direction normal to the grating, and θ determines the orientation. An example, with $k = 2\pi/6$, $v = 0.6$ pixels/frame, and $\theta = \pi/6$ is shown in figure 3-12, along with the estimated motion field. Note that the algorithm computes the normal flow because of the prior distribution, which gives a preference for zero velocities in the case of intensity singularities.

We examined the effects of varying the parameters v, k, and θ on the magnitude of the velocity estimate, for the two-tap, three-tap and five-tap derivative filters. The default pa-

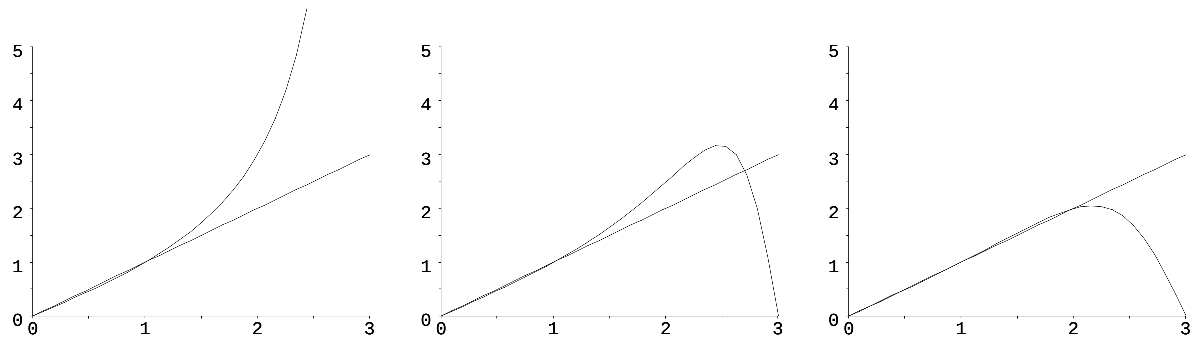


Figure 3-13: Magnitude of the velocity estimate at one point as a function of the speed v of the sinusoid. v varies from 0 to 3 pixels/frame. The three plots are for three different derivative/prefilter pairs: 2-tap on the left, 3-tap in the center, and 5-tap on the right. The straight line indicates the correct speed.

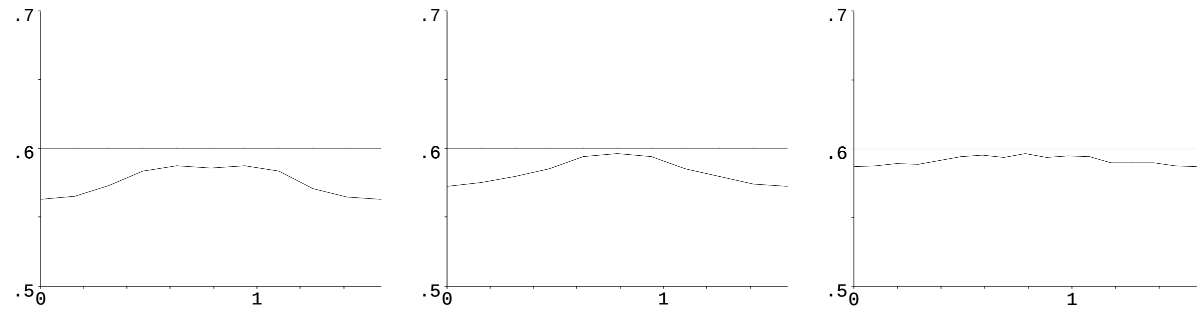


Figure 3-14: Magnitude of the velocity estimate at one point as a function of the normal direction θ of the sinusoid. The three plots are for three different derivative/prefilter pairs: 2-tap on the left, 3-tap in the center, and 5-tap on the right. θ varies from 0 to $\pi/2$. The straight line indicates the correct speed of 0.6 pixels/frame.

rameter values (i.e., the values used for the parameters that are not varied) were as follows: $v = 0.6$, $\theta = 0$, $k = \pi/3$. These are plotted in figures 3-13, 3-14, and 3-15.

Figure 3-13 shows plots of the estimated speed as a function of the actual speed, v. As the speed of the sinusoid increases above one pixel/frame, the two-tap filter is seen to wildly overestimate the velocity magnitude. The three-tap filter also overestimates these speeds, but not as drastically. The five-tap filter produces reliable estimates out to two pixels/frame, and then falls off gracefully.

Figure 3-14 shows plots of the estimated speed as a function of the orientation, θ, of the sinusoid. This is a test of the circular symmetry of the filters. As the orientation varies, the magnitude of the velocity estimate made with the two-tap filter is seen to vary markedly. Furthermore, it is always underestimated. The three-tap filter also varies substantially, but is not as far from the correct speed. The five-tap filter is noticeably better than the other two.

Figure 3-15 shows plots of the estimated speed as a function of spatial frequency, k. All three filters produce deviations from the correct value (0.6 pixels/frame) at high frequencies. The five-tap filter produces relatively accurate estimates out to $k = 3\pi/4$.

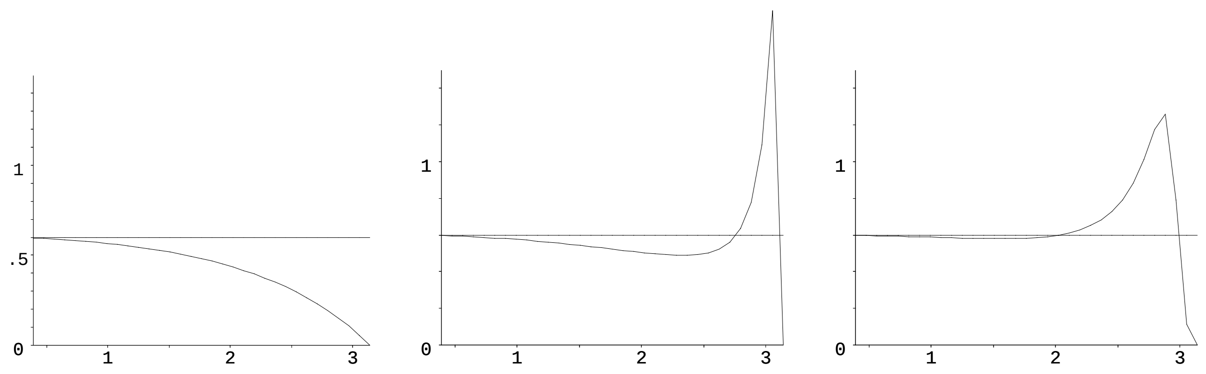


Figure 3-15: Magnitude of the velocity estimate at one point as a function of the spatial frequency k of the sinusoid. The three plots are for three different derivative/prefilter pairs: 2-tap on the left, 3-tap in the center, and five-tap on the right. k varies from $\pi/8$ to π. The straight line indicates the correct speed of 0.6 pixels/frame.

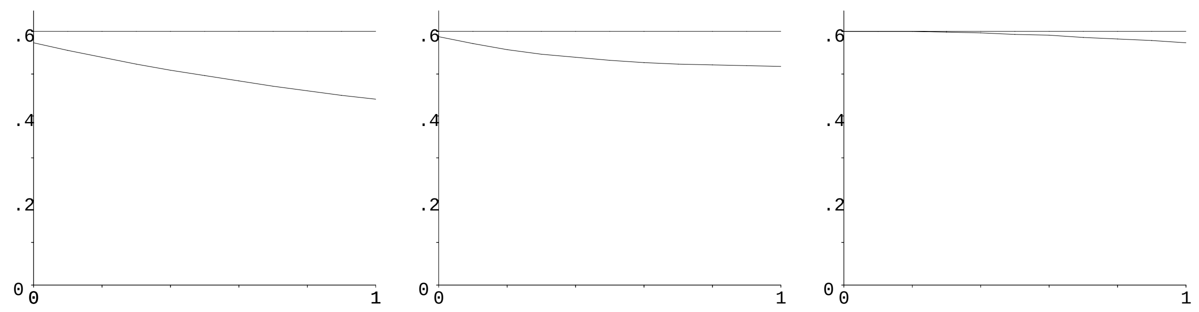


Figure 3-16: Magnitude of the velocity estimate at one point as a function of the percentage increase of amplitude applied at each time step. Contrast increase per frame varies from 0% (contrast is constant) to 100% (contrast doubles at each time step). The three plots are for three different derivative/prefilter pairs: 3-tap on the left, and 5-tap in the center, and the mixed first and second order derivative solution on the right.

In order to test the quality of the mixed-order derivatives as quadrature pairs, we examined a moving sinusoid in which the intensity amplitude increases geometrically with time. We looked at the estimated speed as a function of the percentage increase of amplitude per frame. These results are plotted in figure 3-16 for the three-tap, five-tap, and mixed-derivative estimators. Behavior in these examples depends very much on the size of the blurring filter used to combine the local information. The examples were computed with a three-tap blurring kernel: $[0.25, 0.5, 0.25]$. The mixed derivative solution appears to be substantially more robust to non-motion changes in intensity, as suggested by Fleet and Jepson [29]. We note, however, that the behavior of the basic derivative solution improves when a larger blurring filter is used.

To illustrate the spatial behavior of the algorithm, we estimated the velocity field of an impulse image moving at one pixel per frame. The flow field is shown in figure 3-17. The estimated velocity is correct in the center of the image (at the location of the impulse). The finite size of the derivative filters (five-tap kernels were used for this example) and the blurring of the energies leads to the situation shown, in which the impulse "drags" part of the background along. The velocity surrounding the impulse is consistent with the image intensities: since the

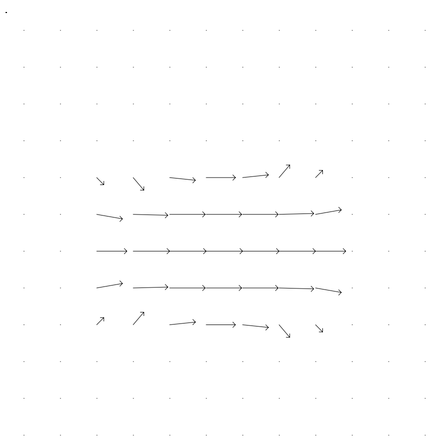


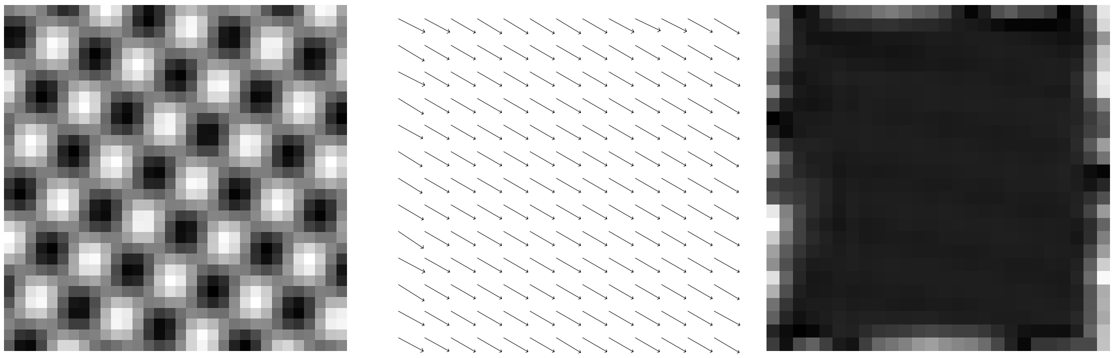
Figure 3-17: Velocity field estimated for a spatial impulse moving rightward at one pixel/frame. Dots correspond to zero velocity vectors.

image is zero everywhere except at the impulse, the motion is completely indeterminate.

Next, we examined a sinusoidal plaid pattern, taken from Barron et. al. [10]. Two sinusoidal gratings with spatial frequency 6 pixels/cycle are additively combined. Their normal orientations are at 54° and $-27°$ with speeds of 1.63 pixels/frame and 1.02 pixels/frame, respectively. The flow is computed using the multi-scale mixed-order algorithm. We built a one-level Gaussian pyramid on each frame, using the following five-tap kernel: $[0.0625, 0.25, 0.375, 0.25, 0.0625]$. Derivative filters used are the five-tap first and second derivative kernels given in figures 3-10 and 3-11. The covariance parameters were set as follows: $\lambda_1 = 0, \lambda_2 = 1, \lambda_p = 1e-5, \lambda_0 = 0.15$. One frame of the sequence, the estimated flow, and the error magnitude image are shown in figure 3-18.

Also shown is a table of error statistics. The errors compare quite favorably with the mean angular errors reported by Barron et. al. [10]. Our mean angular error is an order of magnitude less than all the methods examined, except for that of Fleet and Jepson for which the value was 0.03°. Barron et. al. state that the Fleet and Jepson results are computed with filters that are tuned for the sinusoids in the stimulus and that for a stimulus composed of different sinusoids, the algorithm would exhibit the biases.

We also note also that our mixed-order algorithm is *significantly* more efficient than most of the algorithms in [10]. For example, the Fleet and Jepson algorithm is implemented with a set of 46 kernels. These are implemented separably as 75 convolutions with one-dimensional 21-tap kernels. The mixed-order algorithm requires 8 convolutions (with one-dimensional kernels) for the first derivative measurements, and 15 convolutions (also with one-dimensional kernels) for the second derivatives. In addition, the one-dimensional kernels are only three or five taps in length.



mean E_{mag2}	$1.597e-4$
mean E_{angular}	$0.2746°$
st. dev. E_{angular}	$0.300°$
mean E_{bias}	0.004378

Figure 3-18: Velocity field estimates for sinusoidal plaid sequence. On the left is a frame from the sequence. In the center is the estimated velocity field. On the right is the error magnitude image. Note that the error is concentrated at the boundaries, where derivative measurement is difficult. Error statistics for this computation are given in the table (see text for definitions).

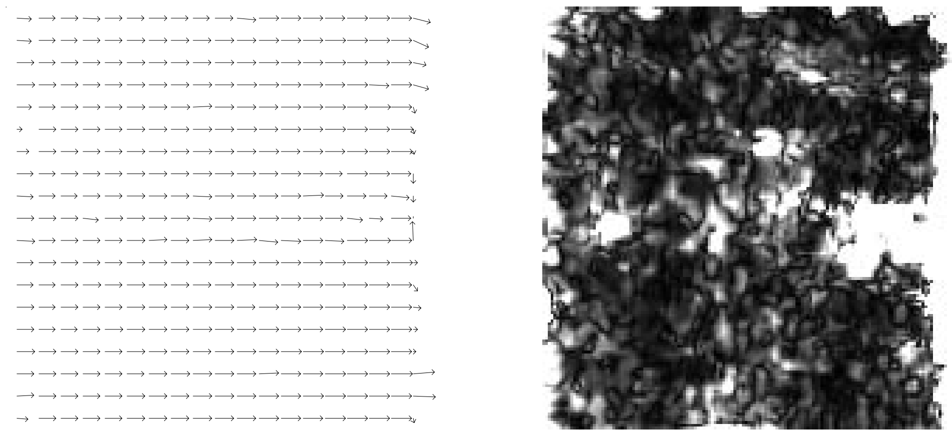
Realistic Synthetic Sequences

We also estimated image velocity fields for the "Translating Tree" and "Diverging Tree" sequences used in [10]. These sequences were generated by warping an image of a tree to simulate translational camera motion with respect to the image plane. One frame from the sequence is shown in figure 3-19. All parameters of the algorithm were the same as those used in the plaid sequence.

Figure 3-20 shows a frame of the estimated image velocity field and the error image for the "Translating Tree" sequence. Also given is a table of the error statistics. Figure 3-21 shows the same data for the "Diverging Tree" sequence. These results compare favorably to those reported in [10]. In particular, the best reported result in [10] for the translating sequence is the Fleet and Jepson algorithm which produced a mean angular error of 0.23 degrees. Note, however, that this result was for a flow field of 50% density (i.e., half of the velocity vectors were discarded as unreliable). The best result at full density is one using the Uras et al. algorithm, which produces a mean angular error of 0.71 degrees, which is comparable to our result of 0.817. The best result reported for the diverging sequence at 100% density is 5.11 degrees (again, using the algorithm of Uras et al.), as compared with our mean angular error of 1.952 degrees.

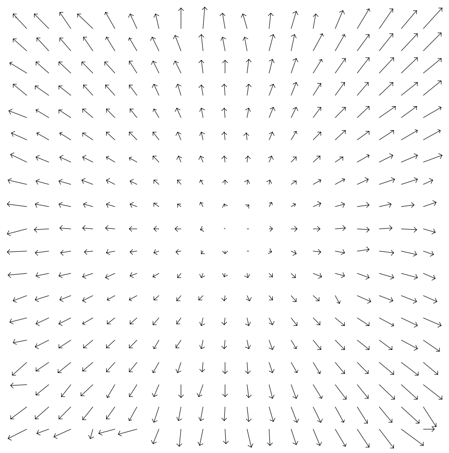
Moving one step closer to real imagery, we estimated velocity fields for a "texture-mapped"

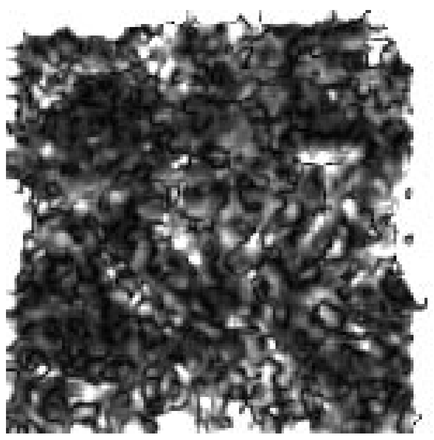
Figure 3-19: An example frame from the "Translating Tree" and "Diverging Tree" sequences.



mean E_{mag2}	0.0625
mean E_{angular}	0.817°
st. dev. E_{angular}	2.93°
mean E_{bias}	−0.0244

Figure 3-20: Estimated velocity for the "Translating Tree" sequence. On the left is the velocity field. On the right is the error magnitude image. The table contains error statistics.





mean E_{mag2}	0.00337
mean E_{angular}	1.95°
st. dev. E_{angular}	1.66°
mean E_{bias}	−0.00364

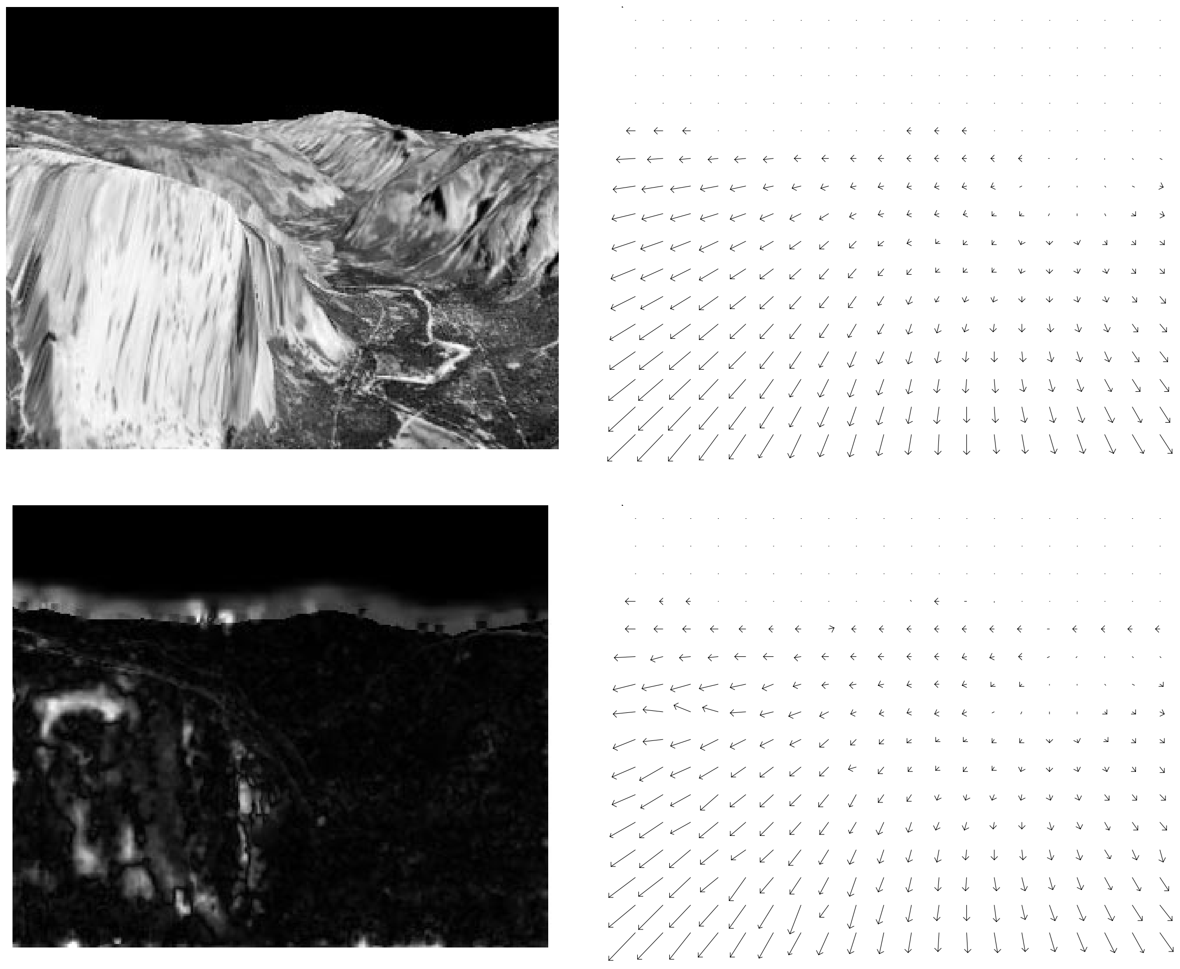
Figure 3-21: Estimated velocity for the "Diverging Tree" sequence. On the left is the velocity field. On the right is the error magnitude image. Also given is a table of statistics.

fly-through sequence of the Yosemite valley [2]. Starting with an aerial photograph and a range (height) map of the Yosemite valley, a sequence of images was rendered for a series of camera positions. Photometric effects are *not* included in this rendering process: the image pixel values are interpolated directly from the intensities of the original photograph. Thus, the sequence contains all of the problem sources except for lighting effects (i.e., singular regions, temporal aliasing, and multiple motions at occlusions). Note that we have the camera motion and the depth of each point in the image, we can compute the *actual* image motion fields.

Again, we computed velocity fields using the multi-scale mixed-order solution. This time, we build a three-level Gaussian pyramid. Parameter settings were as follows: $\lambda_1 = 2e-5, \lambda_2 = 0.004, \lambda_p = 0.5, \lambda_0 = 0.15$. The results are illustrated in figure 3-22. We show a frame from the original sequence, the correct velocity field, the estimated velocity field, and the error magnitude image. Also given is a table of statistics. The statistics do not include the points closer than 10 pixels to the border.

The results are quite accurate, with most errors occurring (as expected) at occlusion boundaries, and at the borders of the image (which may be viewed as a type of occlusion boundary). But qualitative comparisons with the results of the Heeger or Fleet and Jepson algorithm indicate that the errors near these boundaries are contained within smaller regions near the

[2]This sequence was generated by Lyn Quam at SRI International.



mean E_{mag2}	0.031747
mean E_{angular}	3.8105°
st. dev. E_{angular}	7.09378°
mean E_{bias}	−0.01083

Figure 3-22: Results of applying the algorithm to the synthetic "Yosemite" sequence. Upper left: a frame from the original sequence. Upper right: the correct flow field. Lower right: estimated velocity field. Lower left: error magnitude image. Note that errors are concentrated near occlusion boundaries.

boundaries, since the support of the filters is much smaller.

Furthermore, the error statistics compare quite favorably to those reported in [10]. In particular, the best result reported is that of Lucas and Kanade, with a mean angular error of 3.55° and standard deviation of 7.09°. This is almost identical to our result, but the flow vector density is only 8.8%. The best result reported at 100% is that of Uras, which had a mean angular error of 10.44°, and standard deviation of 15.00°. The values given in figure 3-22 are significantly lower.

To analyze the appropriateness of the noise model, we computed a $E_{\text{normalized}}$ at each point. We show this image in figure 3-23, along with the histogram of values. If the flow field errors were exactly modeled by the simple additive gaussian noise terms, then this histogram would be in the form of the distribution plotted in figure 3-24. The error histogram is seen to qualitatively match, suggesting that the noise model is not unreasonable.

Video Imagery

Finally, we examined the qualitative behavior of the algorithm on two real video sequences: the "Rubik Cube" sequence (created by R. Szeliski at DEC Cambridge Research Labs), and the "Garden" sequence.

We used the same parameter settings as for the Yosemite sequence. In order to test the quality of the velocity fields, we used them to interpolate an estimate of frame i by warping frame $i-1$ forward and warping frame $i+1$ backward and averaging the two. Note that this is not a true test of velocity field quality. A vector field very different from the velocity field might do an excellent job of matching pixel intensities from one frame to the next.

Figure 3-25 shows one frame from the "Garden" sequence, one frame of the computed velocity field, and the squared error of the interpolated frame. The SNR of an interpolated frame for this sequence (not shown) was 22.92 dB. As in the synthetic sequences, note that errors are concentrated at occlusion boundaries.

Figure 3-26 shows one frame from the "Rubik" cube sequence, and one frame of the computed velocity field. The SNR of an interpolated frame for this sequence (not shown) was 35.09 dB.

3.5 Summary

In this chapter, we used a Gaussian noise model and a Bayesian estimator to introduce the concept of distributed representation of motion. The uncertainty model chosen produces a Gaussian distribution over velocity space as output. The mean of the distribution is a gain-controlled modification of the basic differential energy solution developed in chapter 2. The

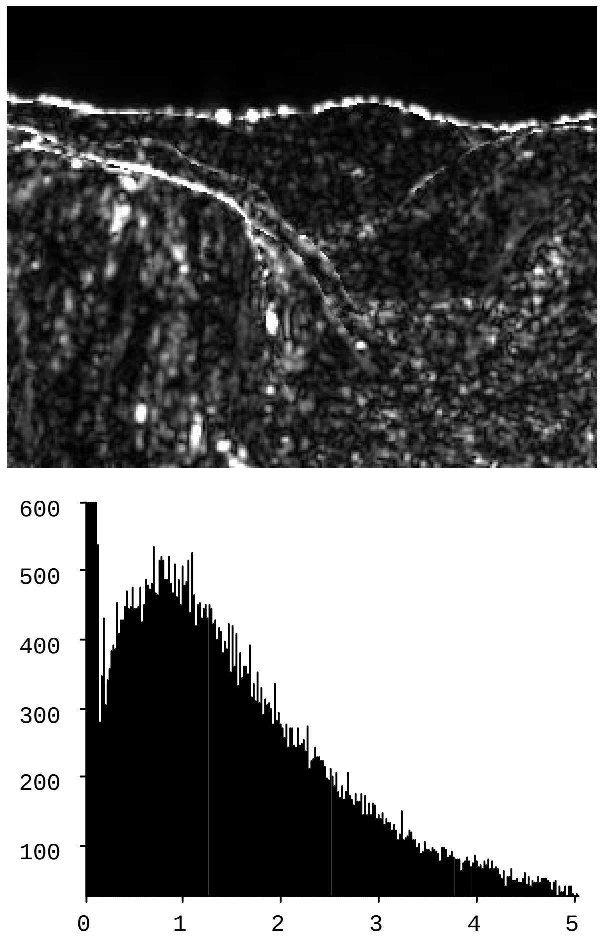


Figure 3-23: Image of $E_{\text{normalized}}$ for the Yosemite sequence velocity estimates. Also shown is the histogram of values.

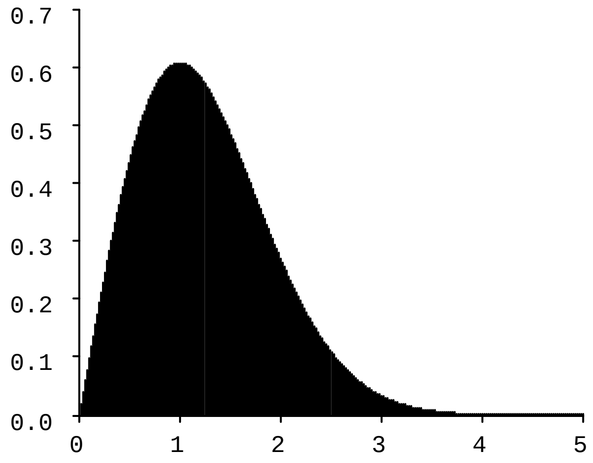


Figure 3-24: Ideal distribution $h(x)$ for the histogram shown above (see text).

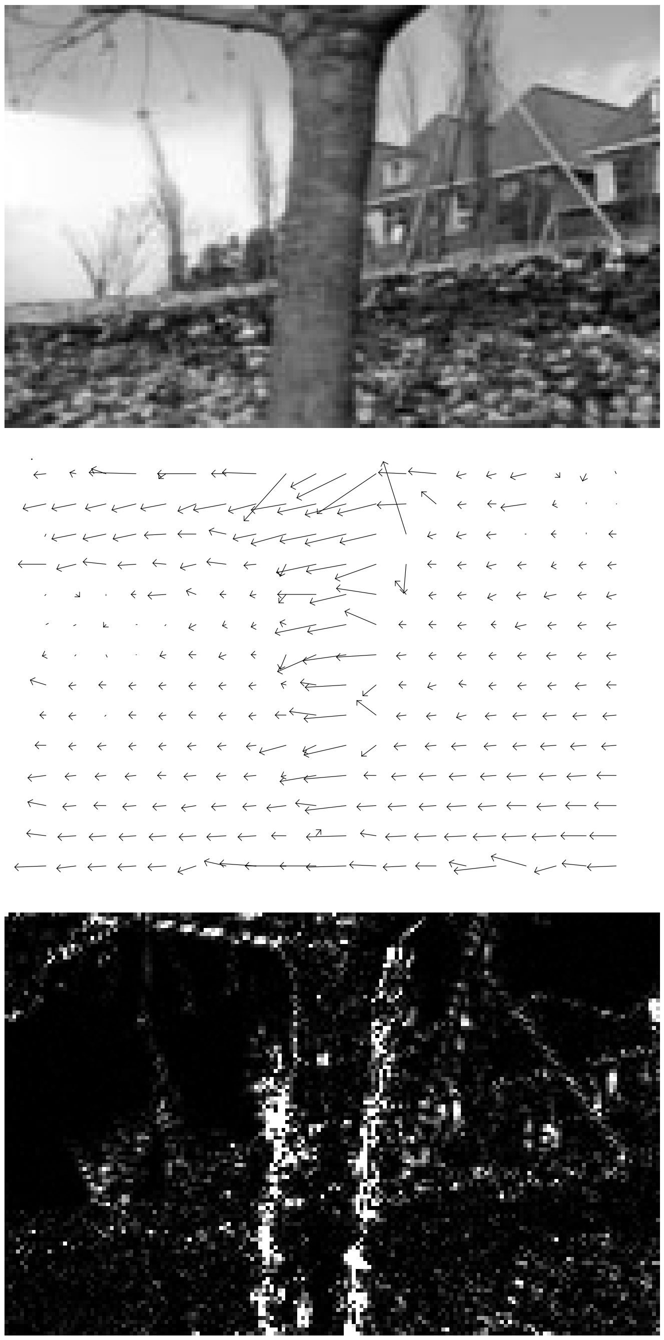


Figure 3-25: Velocity results for the “Garden” sequence. At the top is a frame of the original sequence. In the center is a frame of the estimated motion field. On the bottom is the squared error image of the interpolated frame (see text). The SNR of the interpolated frame is 22.92 dB.

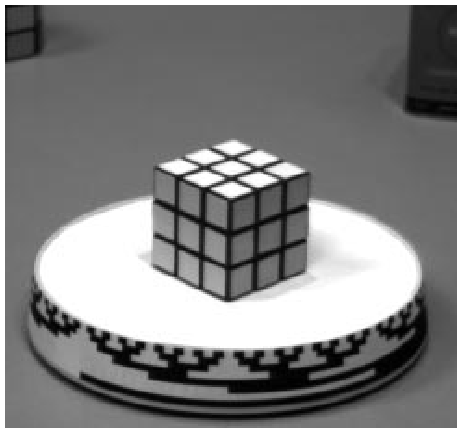
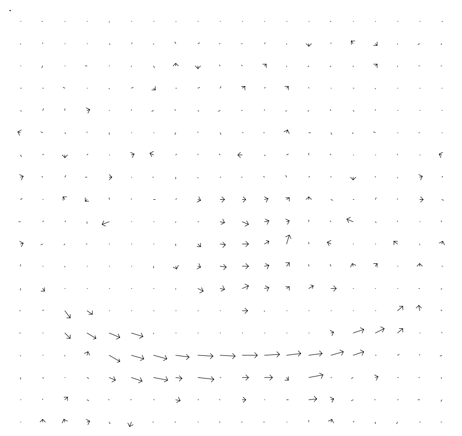
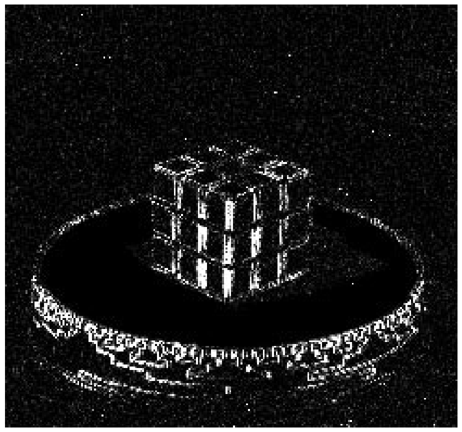




Figure 3-26: Velocity results for the "Rubik Cube" sequence. On the top is a frame of the original sequence. In the center is a frame of the estimated motion field. On the bottom is the squared error image of the interpolated frame (see text). The SNR of the interpolated frame is 35.09 dB.

covariance matrix captures directional uncertainties, allowing proper combination of the output with other sources of information.

We developed a coarse-to-fine estimation algorithm for handling the problem of temporal aliasing. Here we are able to take advantage of the uncertainty information provided by the covariance estimates, propagating this information using an algorithm resembling a Kalman filter over scale.

We discussed the details of algorithm implementation, especially the issue of derivative filter design. Most algorithms developed in the literature rely on very poor two-point derivative estimates. We design a set of prefilters and derivatives that are separable, and optimal according to a simple weighted least-squares criterion in the frequency domain. These filters are extremely compact (3 or 5 taps), especially when compared with other filter-based flow algorithms in the literature.

We showed a large set of diagnostic examples, designed to demonstrate the various strengths and weaknesses of the algorithm we have developed. We demonstrated the reliability of the three-tap and five-tap derivative filters, and showed that the mixed-order solution performs well in the presence of non-motion changes in intensity.

We generated velocity field estimates for a number of synthetic sequences and compared the results to those reported by Barron et al. The probabilistic mixed-order solution was found to outperform all reported algorithms on the the examples tested. We also estimated velocity fields for real sequences, evaluating the quality of the results by interpolating a frame by warping and averaging its two neighboring frames.

Given the estimates of means and covariances, how can we make use of the probabilistic information? The probabilistic coarse-to-fine algorithm is an example of this. Future work could include the prediction of the motion field over time, the estimation of depth from motion, and the estimation of rigid-body motion parameters (i.e., three-dimensional rotations and translations) from image velocities.

The fundamental problem with the algorithm is its failure in regions of multiple motions. Examples were shown at an occlusion boundaries. Not only are the estimates blurred over the boundaries, but the covariance estimates typically indicate high confidence at these locations. This is particularly frustrating, since intuitively the motion at occlusion boundaries seems to be quite rich in information (i.e., motion discontinuities at occlusion boundaries inform us of discontinuities in depth).

The problem is that we are combining information over a patch using a sum-of-squares rule. This assumes that the information under the patch corresponds to a single velocity. One possible route to a solution is through use of robust estimation techniques: we should only combine measurements that correspond to the same motion, discarding the others as “outliers”.

For example, given the set of mixed-order derivative measurements within a patch, we could compute a consistency measure to determine whether it is reasonable to combine them. An example of such a robust estimator was implemented by Black [15].

Some authors [61, 14] have proposed Markov Random Field models that represent discontinuities with "line processes" [34]. As with the "edge detection" problem, this approach tends to be very sensitive to threshold parameters that determine whether a change in the estimate corresponds to a discontinuity or just a rapid variation. They also run into difficulties in producing extended contours by linking together "chains" of discontinuity segments.

Furthermore, these robust mechanisms will only be helpful for multiple motions at occlusion boundaries. In the case of transparency, all of the local motion *measurements* within a patch will contain information about both motions. Thus it is probable that none of the pointwise estimates will produce a motion estimate corresponding to either of the motions. This thought leads us to the topic of the next chapter: the local representation of multiple motions.

Chapter 4

Representing Multiple Motions

Nearly all previous techniques for analyzing motion attempt to compute a single motion at each point in space and time. But as we mentioned in section 1.2, 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 can be two distinct motions: one on each side of the boundary. Since the operators used to compute velocity are of some finite size, they will incorporate information from both velocities into the estimate. This problem will arise whenever there are changes in the motion field that are abrupt compared to the size of the operators.

Another common example is that of transparent surfaces. A scene viewed through a piece of dirty glass will exhibit two motions at each location: that of the glass, and that of the scene behind it. These multiple-motion situations are prevalent, and cause problems for motion algorithms. Biological systems provide inspiration here: humans have no trouble distinguishing the motion of transparently moving sheets.

Several authors have discussed the problem of multiple motions [26, 4, 29, 13]. Some authors have tried to handle the problem by using higher-order expansions of the motion field (eg, affine). Here we will instead consider representing more than one motion at each point. Shizawa and Mase [75, 76] have described algorithms for explicitly computing two motion vectors at each point in the scene. Bergen et. al. [13] have developed an algorithm for separating two transparently combined images moving according to an affine velocity field.

We take a different approach here. In the previous section, we generated a quadratic error function over the space of all velocities $\vec{v}$ for each point in space and time. We wish to 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.

We will build up a representation gradually, through a series of modifications of the differential approaches developed in chapter 2. We define an angular regression version of the standard gradient constraint equation, and then extend this to represent multiple motions. 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, as was illustrated in figure 1-6.

We implement an efficient version of this distributed representation, in which the entire distribution may be interpolated from a sparse set of samples. Note that this parameterization of the distribution is quite different from that of the previous chapter in which we represent the mean and covariance. A sampled representation is also more likely to be appropriate for biological modeling. We demonstrate the use of this sampled representation on simple synthetic examples containing occlusion boundaries and transparent surfaces. Preliminary versions of this work have appeared in [80, 78].

4.1 One-Dimensional Case

In this section, we develop the concept of distributed velocity representation. We will start by analyzing the one-dimensional gradient approach in the frequency domain.

Angular Regression

In chapter 2, we showed that the basic gradient algorithm could be viewed as a regression operation in the spatio-temporal frequency domain. But there are different forms of regression, in addition to the standard ω_t-weighted regression we described previously. In particular, since the spectrum of a moving one-dimensional pattern lies on a line through the origin, with the line orientation corresponding to the arctangent of the speed, it would seem natural to consider optical flow computation as an *angular* estimation problem. Recall the least-squares regression formulation of the basic gradient constraint, as given in equation (2.21):

$$\begin{aligned} E(\vec{v}) &= \sum_{\vec{\omega}} \left[v\omega_x G_1(\vec{\omega}) + \omega_t G_1(\vec{\omega})\right]^2 \cdot |F(\vec{\omega})|^2 \\ &= \sum_{\vec{\omega}} \left[(v,1)^T \cdot \frac{\vec{\omega}}{|\vec{\omega}|}\right]^2 \cdot |G(\vec{\omega})F(\vec{\omega})|^2 \\ &= (v^2+1)\sum_{\vec{\omega}} \left[\hat{u}(v)\cdot\hat{\omega}\right]^2 \cdot |G(\vec{\omega})F(\vec{\omega})|^2 \end{aligned} \tag{4.1}$$

where we are writing $G_1(\vec{\omega})$ as our first derivative prefilter and defining $G(\vec{\omega}) = |\vec{\omega}|G_1(\vec{\omega})$, $\hat{\omega} = \vec{\omega}/|\vec{\omega}|$. The angular velocity vector, $\hat{u}(v)$, is

$$\hat{u}(v) = \frac{1}{(v^2+1)}\begin{pmatrix} v \\ 1 \end{pmatrix},$$

which we will refer to as a "steering" vector.

We have written the expression in this fashion to emphasize that the computation is composed of three operations:

1. The image spectrum $|F(\vec{\omega})|^2$ is weighted by the spectrum of a spatio-temporal prefilter, $|G(\vec{\omega})|^2$.
2. This spectral measure is then connected to the velocity through multiplication by a squared inner product of two unit vectors: $(\hat{u}(v) \cdot \hat{\omega})^2$. Note that the inner product is equal to the cosine of the angle between the two vectors.
3. After summing over frequencies, the resulting function of v is multiplied by a weighting factor in the velocity domain, $(v^2 + 1)$. This weighting imposes a preference for lower speeds.

In the expression of equation (4.1), the inner product component provides the constraint that links the velocity to the derivative measurements. This constraint is at the heart of most differential algorithms. Note that it does not depend on the magnitude of the frequency, $\vec{\omega}$, but only on the *angular* component. 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 weighting factor:

$$A(v) = \sum_{\vec{\omega}} \left[\hat{u}(v) \cdot \hat{\omega}\right]^2 |G(\vec{\omega})F(\vec{\omega})|^2. \tag{4.2}$$

That is, this is a least squares regression fit to a line of slope v, where the errors are measured as the cosine of the angle between a point $\vec{\omega}$ and the line in the direction $\hat{u}(\vec{v})$. Similar "beamforming" functions are used in the direction-of-arrival estimation problem, in which one attempts to estimate the location of electromagnetic waves impinging upon an array of sensors.

Note that there are other forms of regression. For example, we could use the distance perpendicular to the line, as has been suggested by Shizawa and Mase [75, 76]. But this places undue emphasis on high frequency spectral content, and our preliminary tests indicate that it may perform poorly for optical flow estimation.

Nulling versus Maximizing

The value of the regression functions of the previous section will be minimized at the true velocity. This is because the gradient algorithm operates by finding a directional derivative with

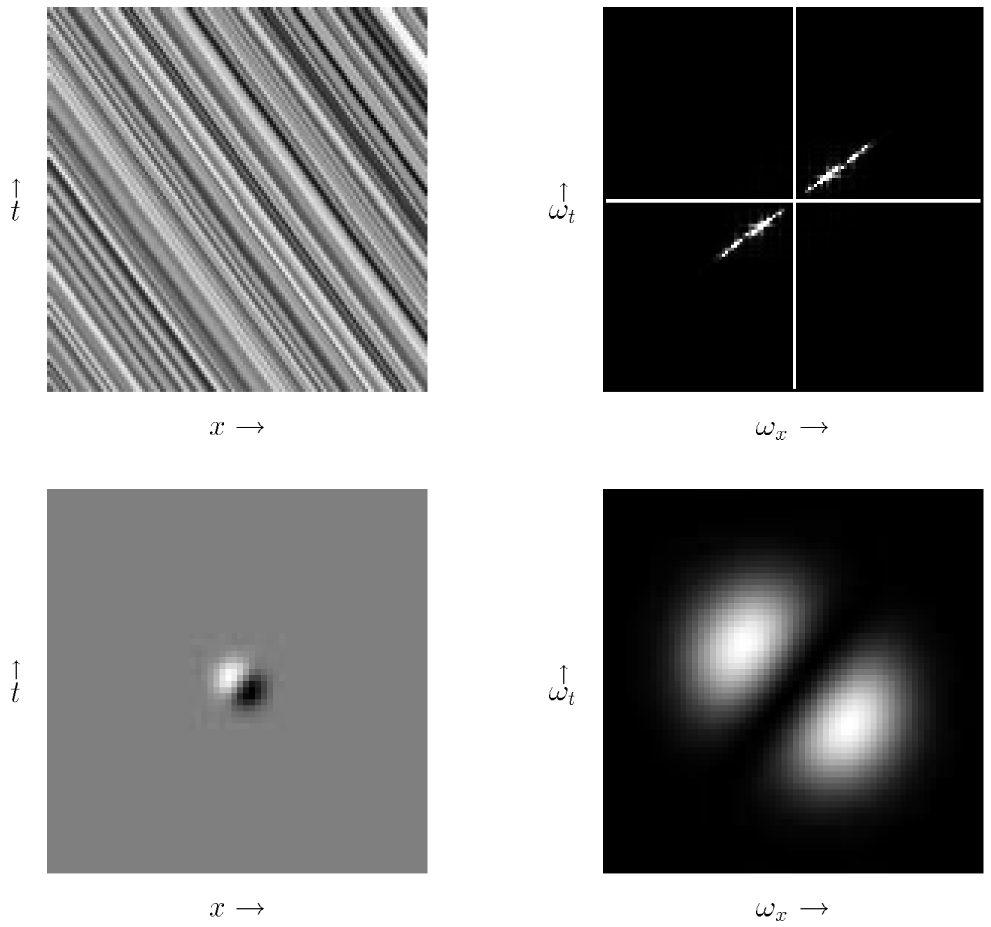


Figure 4-1: Illustration of the gradient as a "nulling" operator. On the upper left is a moving one-dimensional pattern. Below this is the kernel of a directional derivative filter with a null response to this pattern. The corresponding frequency domain picture appears on the right. In the upper right is the power spectrum of the signal, which lies on a line. Below this is the Fourier spectrum of the filter, which is *zero* along this line.

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. This is illustrated in figure 4-1. Thus, the standard gradient algorithm may be termed a "null-steering" algorithm.

This is counterintuitive in light of the intuition expressed by Adelson and Bergen [1], and by other authors of spatio-temporal energy models. In these approaches, one searches for the oriented filter of *maximal* response. In the case of first derivatives, we can easily rewrite the solution as a maximization. In order to accomplish this, we note that the sum of the squared directional cosine in direction $\hat{u}(v)$ and the squared directional cosine in the perpendicular direction, $\hat{u_\perp}(v)$ is unity:

$$[\hat{u}(v) \cdot \hat{\omega}]^2 + [\hat{u_\perp}(v) \cdot \hat{\omega}]^2 = |\hat{\omega}|^2 = 1,$$

since the squared terms just correspond to the projection of $\hat{\omega}$ onto two orthogonal vectors.

Using this fact, we can write the angular function as

$$A(v) = \sum_{\vec{\omega}} \left[1 - (\hat{u_\perp}(v) \cdot \hat{\omega})^2\right] |G(\vec{\omega})F(\vec{\omega})|^2. \tag{4.3}$$

Since the first term in the brackets does not depend on v, minimizing this function is equivalent to maximizing the second term. Thus we write an angular "velocity-energy" function as:

$$\mathcal{P}_1(v) = \sum_{\vec{\omega}} [\hat{u_\perp}(v) \cdot \hat{\omega}]^2 \, |G(\vec{\omega})F(\vec{\omega})|^2. \tag{4.4}$$

This function will respond maximally at the correct velocity. We illustrate this function in figure 4-2.

Note that if we wish to consider this distribution probabilistically, it must be normalized by its integral over v. Note also that despite our manipulations, this expression is still based on *first derivative* measurements. We can transform it back into the spatial domain (once again, using Parseval's rule):

$$\begin{aligned}
\mathcal{P}_1(v) &= \sum_{\vec{\omega}} \left| \hat{u_\perp}(v) \cdot \vec{\omega} \frac{G(\vec{\omega})}{|\vec{\omega}|} F(\vec{\omega}) \right|^2 \\
&= \sum_{\vec{x}} \left[\hat{u_\perp}(v) \cdot \left(\vec{\nabla}\, g_1(\vec{x}) * f(\vec{x}) \right) \right]^2,
\end{aligned}$$

where $g_1(\vec{x})$ is a prefilter with Fourier transform $G_1(\vec{\omega}) = G(\vec{\omega})/|\vec{\omega}|$.

Representing Multiple Motions

The function illustrated in figure 4-2 is *always* unimodal. 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: We simply replace the directional first derivative with a directional Nth derivative. In the frequency domain, we can write the distributed representation in terms of Nth order derivatives by raising the directional cosine function to the Nth power:

$$\begin{aligned}
\mathcal{P}_N(v) &= \sum_{\vec{\omega}} \left[(\hat{u_\perp}(v) \cdot \vec{\omega})^N G_N(\vec{\omega}) F(\vec{\omega}) \right]^2 \\
&= \sum_{\vec{\omega}} \left[(\hat{u_\perp}(v) \cdot \hat{\omega})^N G(\vec{\omega}) \right]^2 |F(\vec{\omega})|^2 \qquad (4.5) \\
&= \sum_{\vec{\omega}} \left[G(\vec{\omega}) \sum_{n=0}^{N} \left(\tfrac{N!}{n!(N-n)!} \right) (\hat{u_{\perp x}}(v))^n (\hat{u_{\perp t}}(v))^{N-n} (\hat{\omega}_x)^n (\hat{\omega}_t)^{N-n} \right]^2 |F(\vec{\omega})|^2,
\end{aligned}$$

where, analogous to the previous case, we define $G(\vec{\omega}) = |\vec{\omega}|^N G_N(\vec{\omega})$. We have written the last expanded equation to emphasize two points. First, the Nth directional cosine is computed as

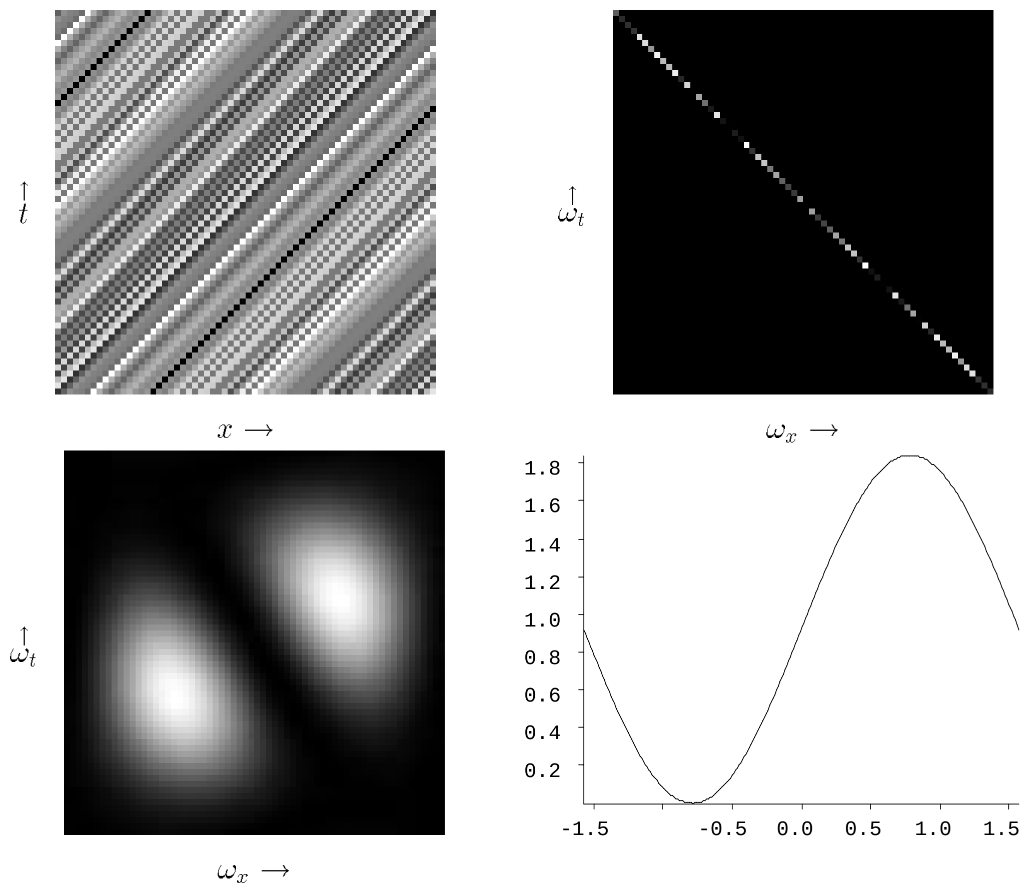


Figure 4-2: Illustration of the computation of the distribution $\mathcal{P}_1(v)$. On the top is a space-time translating one-dimensional noise signal moving at velocity $\hat{v}$. On the right is its power spectrum, plotted over the range $[-\pi, \pi]$. Below/left is the power spectrum of a directional derivative filter, $(\hat{u_\perp}(\tan(\alpha)) \cdot \hat{\omega})^2 |G(\vec{\omega})|^2$, for an arbitrary angle α. 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. On the right is the resulting distribution, $\mathcal{P}_1(v)$, plotted as a function of $\alpha = \arctan(v)$, which is in the form of a squared cosine (this is, of course, not true if we plot it as a function of v). The peak of the distribution is at $\hat{\alpha} = \arctan \hat{v}$.

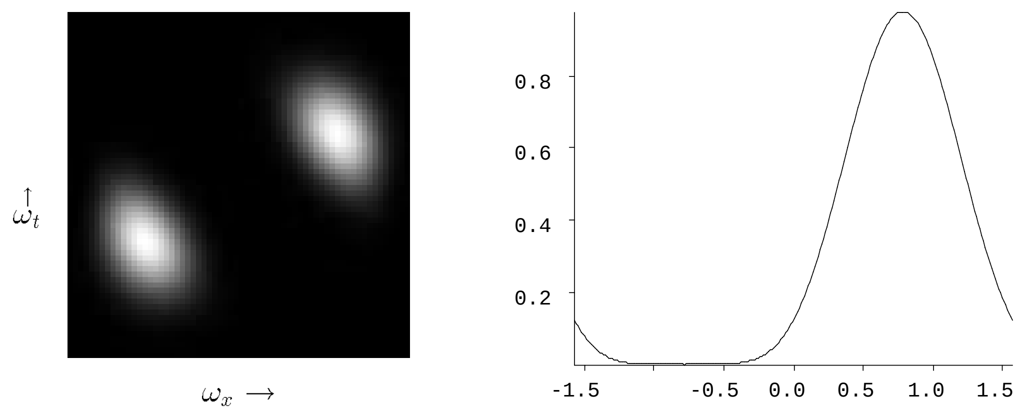


Figure 4-3: Illustration of the computation of the distribution over velocity space. The signal being analyzed is the same as the one in figure 4-2. On the left is the power spectrum of a directional third derivative filter. Four such filters are used to analyze the image. On the right is the resulting distribution over $\alpha = \arctan(v)$.

a *linear* combination of linear measurements. Each term in the sum over n corresponds (in the spatial domain) to a measurement of an Nth-order derivative. Note that some of them are cross-derivatives. Second, although the distribution is no longer quadratic in the components of v, it is *still* quadratic in these linear measurements.

Figure 4-3 illustrates the application of a set of third-order filters to the example shown previously in figure 4-2. Note that the resulting distribution is narrower than in the first derivative example. In a signal containing two motions, the distribution will typically exhibit two modes. [1] 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 these higher-order derivative measurements would perform poorly, because the minimum is very broad.

Many authors have argued that the use of higher-order derivatives for estimating motion leads to increased noise sensitivity. An Nth-order x-derivative operator, for example, has a Fourier magnitude of $|\omega_x|^N$ 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 factors of $|\omega_x|$ are absorbed into the definition of the prefilter $G(\vec{\omega})$. The 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

[1]We note, however, that the peaks of these two modes will *not* necessarily align with the arctangents of the two velocities.

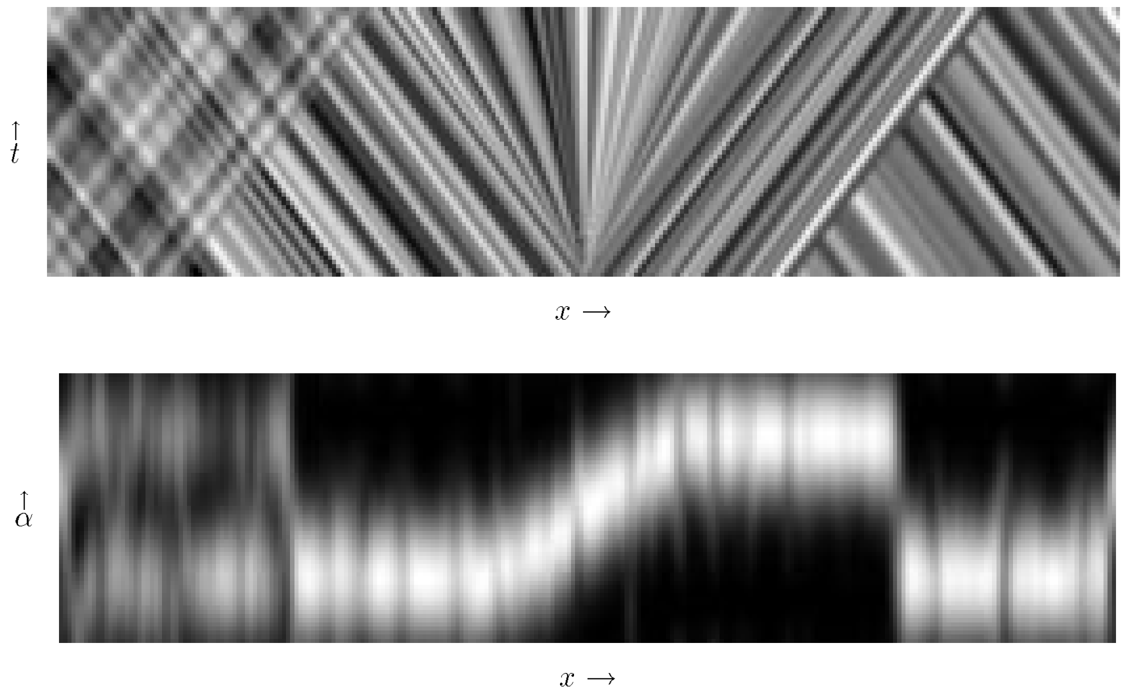


Figure 4-4: Computation of $\mathcal{P}_3(v)$ for a moving one-dimensional pattern. At the top is the pattern, displayed as a space-time image. Consider moving from left to right across a horizontal scanline centered in the image. On the left side, the motion is of two transparently combined bandpassed noise signals. Past the transparent region, the motion is leftward. After this the motion gradually varies (there is a dilation) until it becomes rightward. Then we reach an occlusion boundary, at which point the motion becomes abruptly leftward again. On the bottom is the response of the operator, plotted as a function of space and $\alpha = \arctan(v)$. The operator response is seen to match the motion of the pattern. On the left, the operator indicates two motions via a multimodal distribution over α. The dilating central region appears as a diagonal band. The operator response changes fairly abruptly at the occlusion boundary, although it is bimodal at the boundary.

generally be made smaller in their spatial extent. For the examples given in the next section, we will use third derivatives.

To demonstrate that this algorithm is capable of representing multiple motions, we applied it to a one-dimensional moving image sequence. The input image is illustrated (as an $x - t$ image) in figure 4-4. The image contains several translating regions, a region of additive transparency, a dilating region, and an occlusion boundary. Also shown in the figure is the response of the operator, $\mathcal{P}_3(v)$, plotted as a function of x and $\alpha = \arctan(v)$. The response is seen to match the signal motion.

Distribution Sampling and Interpolation

Equation (4.6) 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 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 sparse set of sample points.

First, we note that the directional *linear* measurements may be interpolated from a fixed set of directional measurements. Expanding the Nth-order directional derivative gives

$$\begin{aligned}(\hat{u_\perp}(v)\cdot\hat{\omega})^N G(\vec{\omega})F(\vec{\omega}) &= \sum_{n=0}^{N}\left(\tfrac{N!}{n!(N-n)!}\right)(\hat{u_{\perp x}}(v))^n(\hat{u_{\perp t}}(v))^{N-n}[\hat{\omega}_x^n\hat{\omega}_t^{N-n}G(\vec{\omega})F(\vec{\omega})] \\ &= \vec{a}(v)\cdot\vec{m}(\vec{\omega}),\end{aligned}$$

where we have rewritten the expression from equation (4.6) as an inner product of vectors $\vec{a}(v)$ and $\vec{m}(\vec{\omega})$ defined by:

$$a_n(v) = \left(\tfrac{N!}{n!(N-n)!}\right)(\hat{u_{\perp x}}(v))^n(\hat{u_{\perp t}}(v))^{N-n}$$

and

$$m_n(\vec{\omega}) = [\hat{\omega}_x^n\hat{\omega}_t^{N-n}G(\vec{\omega})F(\vec{\omega})]$$

as in equation (4.6). Note that the subscripts x and t indicate the spatial and temporal components of $\hat{u}_i$, respectively. This is just the frequency domain version of the interpolation formula of equation (2.17), for arbitrary N.

Now, we can write down a set of these expansions for $N+1$ distinct velocities, v_i:

$$(\hat{u}_i\cdot\hat{\omega})^N G(\vec{\omega})F(\vec{\omega}) = \vec{a}(v_i)\cdot\vec{m}(\vec{\omega})$$

This is a set of linear equations for the $N+1$ directional measurements, as a function of the $N+1$ separable measurements given by the bracketed expression. We rewrite this in matrix notation as:

$$\vec{d}(\vec{\omega}) = \mathbf{A}\vec{m}(\vec{\omega})$$

where

$$\mathbf{A} = \begin{pmatrix}\vec{a}(v_0)\\ \vec{a}(v_1)\\ \vdots\\ \vec{a}(v_N)\end{pmatrix}$$

Now, assuming $\mathbf{A}$ is invertible, we may write the general directional measurement as a linear combination of the vector of fixed directional measurements, $\vec{d}(\vec{\omega})$:

$$(\hat{u_\perp}(v)\cdot\hat{\omega})^N G(\vec{\omega})F(\vec{\omega}) = \vec{a}(v)\mathbf{A}^{-1}\vec{d}(\vec{\omega}). \qquad (4.6)$$

The important point here is that our linear operators span the space of their own rotations. Freeman and Adelson have elucidated an elegant theory of such functions, which they call "steerable" [31]. 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. Similar concepts have been studied by other authors [51, 67].

We can take the interpolation in equation (4.6) one step further, and compute the value of the distribution $\mathcal{P}(v)$ at any v from its value at a set of fixed velocities. For simplicity, we demonstrate this result for the first derivative case. The higher derivative case is analogous. We rewrite the expression in equation (4.4) as follows:

$$\begin{aligned}
\mathcal{P}_1(v) &= [(\hat{u_\perp}(v)\cdot\hat{\omega})G(\vec{\omega})F(\vec{\omega})]^2 && (4.7)\\
&= |(\hat{u_\perp}_x(v)\omega_x + \hat{u_\perp}_t(v)\omega_t)\,G(\vec{\omega})F(\vec{\omega})|^2 \\
&= \hat{u_\perp}_x^2(v)\left[\omega_x^2|G(\vec{\omega})F(\vec{\omega})|^2\right] && (4.8)\\
&\quad + 2\hat{u_\perp}_x(v)\hat{u_\perp}_t(v)\left[\omega_x\omega_t|G(\vec{\omega})F(\vec{\omega})|^2\right] && (4.9)\\
&\quad + \hat{u_\perp}_t^2(v)\left[\omega_t^2|G(\vec{\omega})F(\vec{\omega})|^2\right]. && (4.10)
\end{aligned}$$

Thus $\mathcal{P}(v)$ may be computed as a linear combination of the three quadratic measurements corresponding to the three bracketed terms on the right hand side.

Now we can use the same trick as in the linear case to write this as a linear combination of samples of $\mathcal{P}(v)$ by solving three simultaneous linear equations:

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

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

$$\mathbf{B} = \begin{pmatrix}\hat{u_\perp}_x^2(v_1) & 2\hat{u_\perp}_x(v_1)\hat{u_\perp}_t(v_1) & \hat{u_\perp}_t^2(v_1)\\ \hat{u_\perp}_x^2(v_2) & 2\hat{u_\perp}_x(v_1)\hat{u_\perp}_t(v_2) & \hat{u_\perp}_t^2(v_2)\\ \hat{u_\perp}_x^2(v_3) & 2\hat{u_\perp}_x(v_1)\hat{u_\perp}_t(v_3) & \hat{u_\perp}_t^2(v_3)\end{pmatrix}$$

and $\vec{Q}$ is a vector of the three quadratic measurements from equation (4.10):

$$\vec{Q} = \begin{pmatrix}\sum_{\vec{\omega}}\left[\omega_x^2|G(\vec{\omega})F(\vec{\omega})|^2\right]\\ \sum_{\vec{\omega}}\left[\omega_x\omega_t|G(\vec{\omega})F(\vec{\omega})|^2\right]\\ \sum_{\vec{\omega}}\left[\omega_t^2|G(\vec{\omega})F(\vec{\omega})|^2\right]\end{pmatrix}.$$

Assuming the $\vec{v}_i$ are suitably chosen (in practice, this is not a problem), we can invert the matrix $\mathbf{B}$, and use it to solve for $\mathcal{P}(v)$:

$$\mathcal{P}(v) = \begin{pmatrix}\hat{u_\perp}_x^2(v)\\ 2\hat{u_\perp}_x\hat{u_\perp}_t(v)\\ \hat{u_\perp}_t^2(v)\end{pmatrix}^T\cdot\mathbf{B}^{-1}\cdot\begin{pmatrix}\mathcal{P}(v_1)\\ \mathcal{P}(v_2)\\ \mathcal{P}(v_3)\end{pmatrix}$$

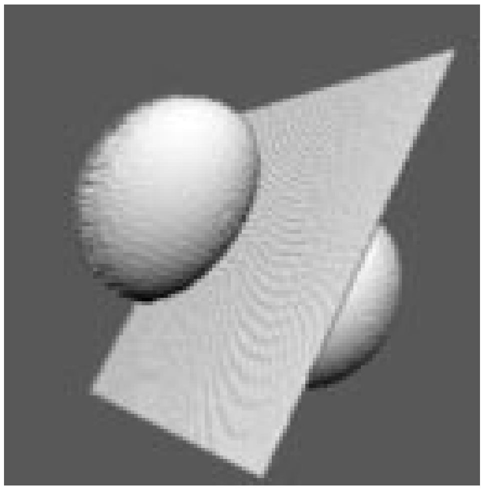


Figure 4-5: Frequency-domain illustration of the "nulling" aspect of the gradient constraint. The gradient solution operates by finding the direction in which a derivative operator has no response. Shown is a plane corresponding to a particular velocity, and a level surface of the power spectrum of a directional derivative filter that does not respond to energy on this plane.

That is, $\mathcal{P}(v)$ may be written as a *linear* combination of the three values $\mathcal{P}(v_i)$. 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.

4.2 Two-Dimensional Case

Now we will extend the analysis of the previous section to two dimensions. The situation becomes a bit more complicated. As in one dimension, we can "angularize" the gradient constraint equation:

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

where the steering vector is now defined as $\hat{u}(\vec{v}) = (v_x, v_y, 1)^T / \sqrt{|v|^2 + 1}$. The dot product in the brackets now corresponds to the cosine of the angle between two three-dimensional vectors.

This expression is again a null-steering function. Recall that the power spectrum of a two-dimensional translating image lies on a plane in the Fourier domain. Then the expression above is minimized when the steering vector $\hat{u}(\vec{v})$ is perpendicular to the spectral plane. This is illustrated in figure 4-5. 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, we note that the sum of squares of the directional cosines

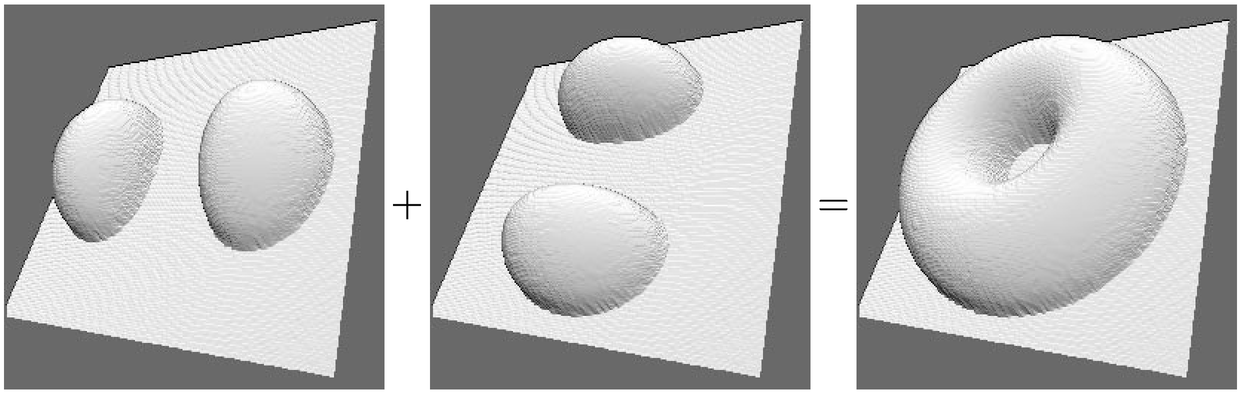


Figure 4-6: Illustration of the gradient algorithm as a "max-steering" solution. The "velocity energy" surface is computed by summing the power responses of two directional derivatives lying in the spatio-temporal frequency plane corresponding to a given velocity. Illustrated is an instance of such a 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 "donut", bisected by the plane.

along a set of *three* orthogonal axes is unity. Thus we can write a sum of the squared directional cosine in direction $\hat{u}(\vec{v})$ and cosines in two perpendicular directions:

$$[\hat{u}(\vec{v}) \cdot \hat{\omega}]^2 + [\hat{u_a}(\vec{v}) \cdot \hat{\omega}]^2 + [\hat{u_b}(\vec{v}) \cdot \hat{\omega}]^2 = 1,$$

where we define (assuming that $\hat{u}(v) \neq \hat{e}_x$):

$$\begin{aligned} \hat{u_a}(\vec{v}) &= \hat{u}(v) \times \hat{e}_x \\ \hat{u_b}(\vec{v}) &= \hat{u}(v) \times \hat{u_a}(\vec{v}) \end{aligned}$$

where $\hat{e}_x$ is a unit vector in the direction of the x-axis. Thus the squared directional cosine in the direction of $\hat{u}(\vec{v})$ is just one minus the sum of the squared directional cosines in the $\hat{u_a}(\vec{v})$ and $\hat{u_b}(\vec{v})$ directions.

Analogous to the one-dimensional case, we define a max-steering function as follows:

$$\mathcal{P}(\vec{v}) = \sum_{\vec{\omega}} |(\hat{u_a}(\vec{v}) \cdot \hat{\omega}) G(\vec{\omega}) F(\vec{\omega})|^2 + \sum_{\vec{\omega}} |(\hat{u_b}(\vec{v}) \cdot \hat{\omega}) G(\vec{\omega}) F(\vec{\omega})|^2$$

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. This sum of the two filters in the plane will form a smooth ring-shaped weighting function or "donut", bisected by the plane; thus we call this computation a "donut mechanism". This construction is illustrated in figure 4-6. The reader may wonder why we do not extract the ring-shaped frequency band directly, with a single filter. The answer is that a large ring filter would be very susceptible to phase cancellations, as discussed in section 2.3.

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

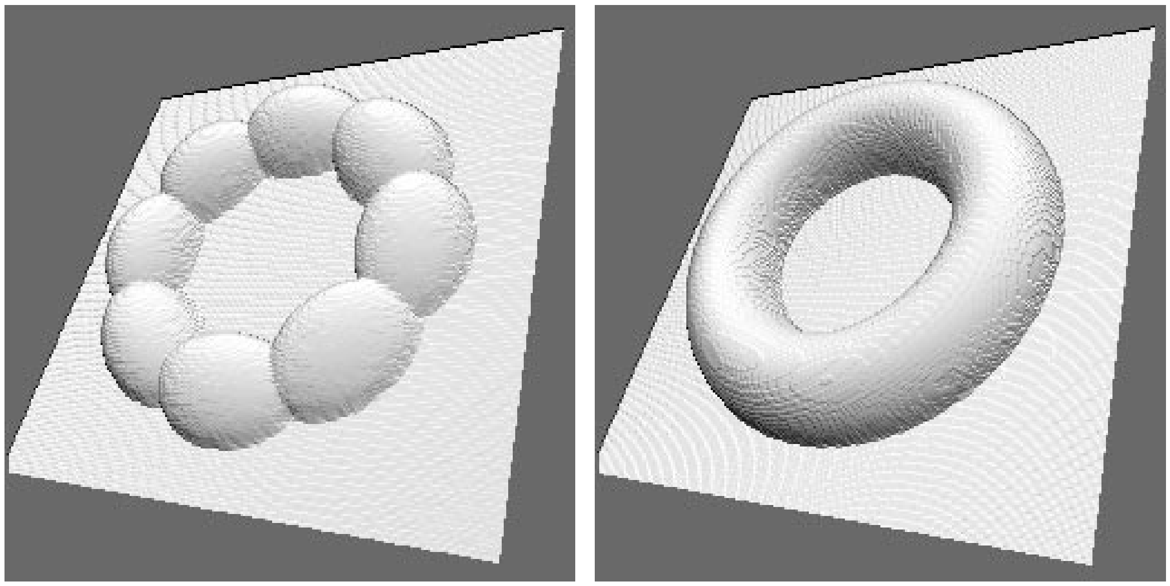


Figure 4-7: Illustration of the "donut" mechanism, based on third derivatives of a Gaussian prefilter. A ring of four third derivatives lying on the plane corresponding to a particular velocity are used to measure evidence for the presence of that velocity. These directional derivatives are computed 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 "donut", illustrated on the right.

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 *at least* $(N+1)$ Nth derivatives to cover the plane uniformly. We define a set of equally-spaced directions as:

$$\hat{u}_i(\vec{v}) = \cos\left(\frac{2\pi i}{N+1}\right)\hat{u}_a(\vec{v}) + \sin\left(\frac{2\pi i}{N+1}\right)\hat{u}_b(\vec{v}) \qquad 0 \le i \le N.$$

The generalized error function now looks like:

$$\begin{aligned} \mathcal{P}_N(\vec{v}) &= \sum_{\vec{\omega}} \left| \sum_{i=0}^{N} [\hat{u}_i(\vec{v}) \cdot \hat{\omega}]^N \, G(\vec{\omega}) F(\vec{\omega}) \right|^2 \\ &= \sum_{\vec{\omega}} \sum_{i=0}^{N} [\hat{u}_i(\vec{v}) \cdot \hat{\omega}]^{2N} \, |F(\vec{\omega})|^2 \,. \end{aligned} \tag{4.11}$$

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 4-7, for a set of third derivative filters.

4.3 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 4-8 illustrates the behavior of the distributed mechanism in three prototypical singularity situations. As before, 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.

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, as with the probabilistic velocity distribution of the previous chapter, 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 4-9 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 has two modes, indicating the presence of two velocities.

Figure 4-10 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 4-11. Again, the response is bimodal.

We note that we have *not* solved for the velocities (as stated earlier, these typically do not correspond to the peaks of the distribution). We also have not solved the "assignment" problem of determining which content of the image is moving with which velocity.

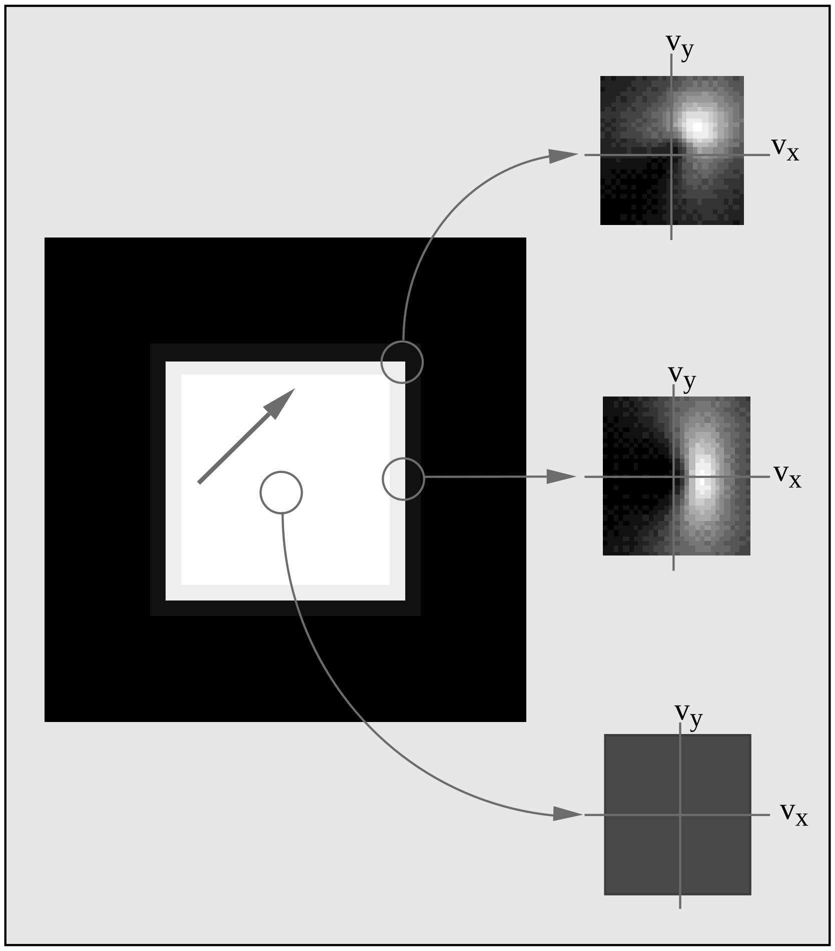


Figure 4-8: Response of the mechanism in several regions of a moving square sequence. In the corner, the velocity is well defined, and the distributed response is a well-localized "lump". On the side, the velocity is only constrained in the direction normal to the edge, and the distributed response is a ridge. In the center, the velocity is completely unconstrained, and the distributed response is flat. These distributions are similar to those for the probabilistic algorithm, illustrated in figure 3-2.

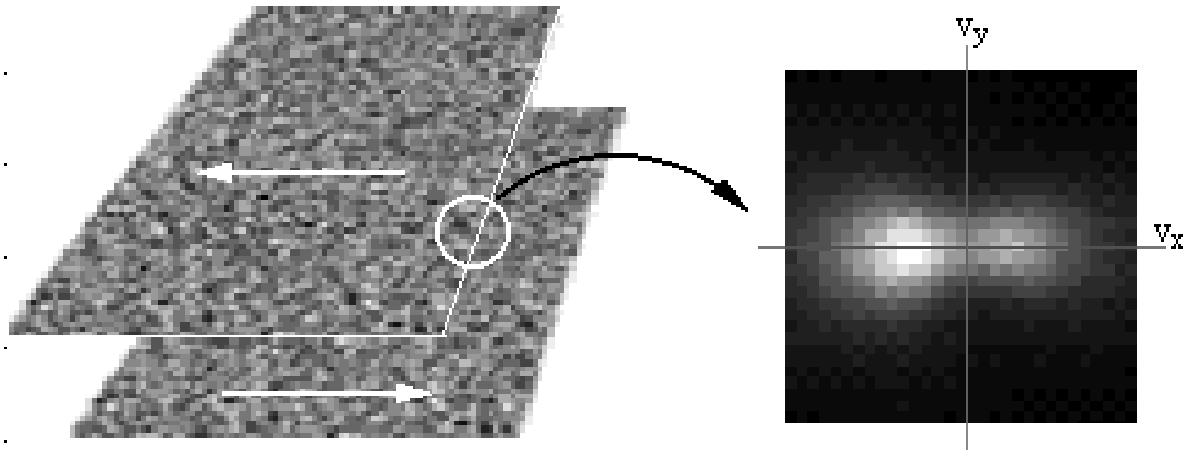


Figure 4-9: Response of the mechanism near an occlusion boundary. On the left is an illustration of the synthetic image sequence. The sequence consists of two white noise patterns, displayed on the right and left sides of the image, such that the left one occludes the right one. The patterns move in 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.

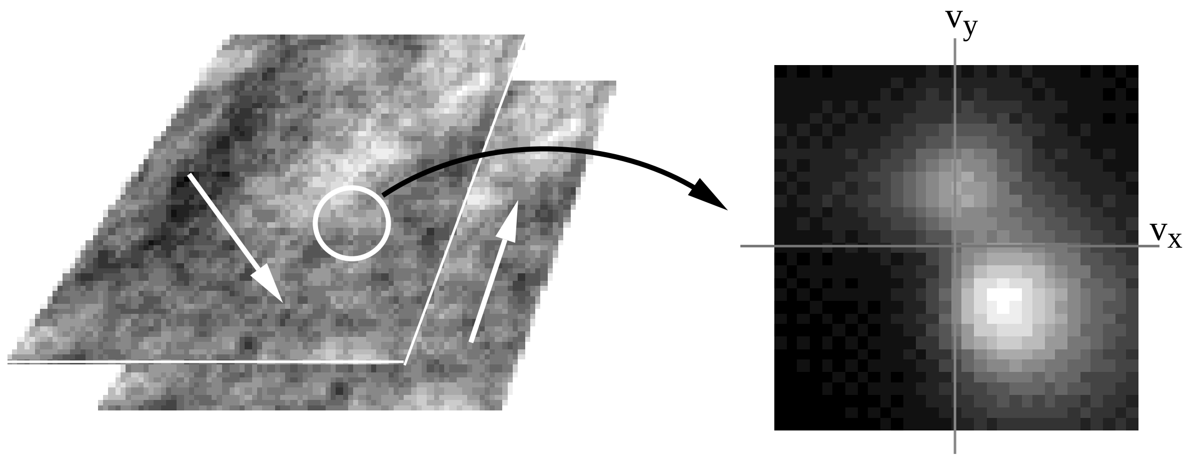


Figure 4-10: Response of the mechanism in the presence of additive transparency. On the left is an illustration of the synthetic image sequence, which consists of two additively combined fractal noise patterns moving in different directions (one upward, the other down and to the right). The white line 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.

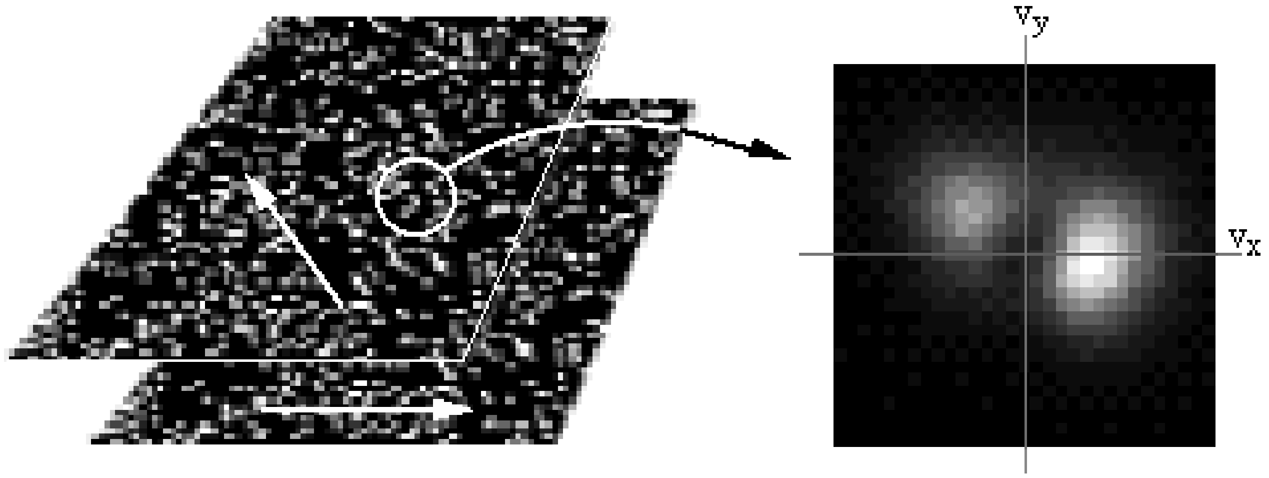


Figure 4-11: Response of the mechanism to interspersed moving dot patterns. transparency. On the left is an illustration of the synthetic image sequence, which consists of two random dot patterns that have been combined using a logical "or" operation. The white line separating the two patterns is for 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.

4.4 Summary

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 an angular regression version 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 chapter 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 weighted regression error function, although it may prove to be so. Regardless, the methods presented in this chapter may be applied (although not as easily) to the standard velocity constraint equation.

The concepts presented in this chapter 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 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 [21].

Chapter 5

Biological Modeling

In this section, we discuss the relevance of the models developed in previous sections to biological vision. As discussed in the introduction, mammalian visual systems devote significant resources to the processing of visual motion. We will focus on experimental knowledge of these visual systems gathered from the fields of physiology and psychophysics.

We first discuss psychophysics of human motion perception. A large literature exists on the topic of motion psychophysics, starting at least as far back as the beginning of this century. A review of this topic may be found in [64]. We will demonstrate the use of a distributed model for quantitative prediction of psychophysical data on the perception of sinusoidal plaids.

Then we will discuss mammalian physiology. We will give a brief synopsis of the physiology of the visual system, provide a qualitative correspondence of this physiology with the model of the previous section, and suggest some experiments that could be used to test the validity of such a model. Some of this work has been published previously in [85, 43, 44, 83].

5.1 Psychophysics

Much of the recent work in motion psychophysics has been based on simple sinusoidal stimuli. As discussed earlier, the motion of a one-dimensional signal is ambiguous: there is no information present in the signal to indicate the speed of movement parallel to the stimulus orientation. Thus the velocity of the stimulus is constrained to lie on a line in the two-dimensional space of velocities, as defined by equation (2.2). Nevertheless, when a human observer is presented with a moving one-dimensional pattern (and there are no other directional queues such as boundary shape), the viewer can assign a direction and speed of motion to the stimulus. The direction reported is that normal to the stimulus orientation. [1] For high-contrast gratings, the perceived speed is the actual translation speed of the grating, although experiments by Thompson and Stone indicate that the perceived speed *does* vary with contrast [93, 89].

When two such gratings are superimposed, the perception becomes more complicated. We first note that more than one physical explanation can account for the stimulus. For example, the visual pattern could be formed from 1) two additively transparent gratings sliding over one another (each moving at some velocity consistent with its constraint line), or 2) a rigid "plaid" pattern translating at a single velocity.

A series of experiments have revealed that the parameters of such a two-grating stimulus stimulus may be adjusted to give either of these perceptions [3, 60, 52]. The general rule seems to be that when the gratings are "similar" (i.e., similar contrast, color, spatial-frequency, orientation, speed), they are perceived as a coherent plaid. Patterns in which the gratings are very different in one or more of these parameters tend to generate perceptions of transparency.

Preliminary studies indicate that the full "donut-mechanism" model described in chapter 4 can account for these perceptions of transparency. If we consider the distribution produced by various stimuli, and we assume that a bimodal distribution corresponds to a perception of transparency, we can make predictions about these perceptions. Currently, all statements are qualitative, as the model has too many free parameters to make reasonable quantitative predictions. But it seems clear that the model has an increased tendency to produce bimodal distributions when the superposed gratings are: 1) of very different spatial frequencies, 2) of high temporal frequency, 3) at very different orientations.

In the case of the coherent plaid interpretation, the unique physically consistent velocity for the pattern is the one satisfying the constraints of both gratings. This is easily depicted via a geometrical construction known as the "intersection-of-constraints" (IOC), as illustrated in figure 5-1. Note that in the case of gratings moving in opposite directions, the IOC solution does not exist, as the constraint lines from the two gratings do not intersect. In this case, the perception is of a flickering stationary sinusoidal pattern (this stimulus is known as a "counterphase" grating).

The original descriptions of plaid perception by Adelson and Movshon [3, 60] suggested that IOC might explain the perceived motion of coherent plaids, but later work [27, 90] clearly indicates that a strict interpretation of IOC as a model for human perception fails in many situations. Some authors have proposed models that compute a sort of average of the IOC solution and a vector sum (i.e., the sum of the grating normal velocity vectors) [27]. It is, however, unnecessary to invoke a second calculation such as vector sum in order to explain the deviations of perceived velocity from the IOC velocity. In the next section, we use the basic

[1] This perceived direction can, however, be altered by changing the shape of the aperture through which the stimulus is viewed.

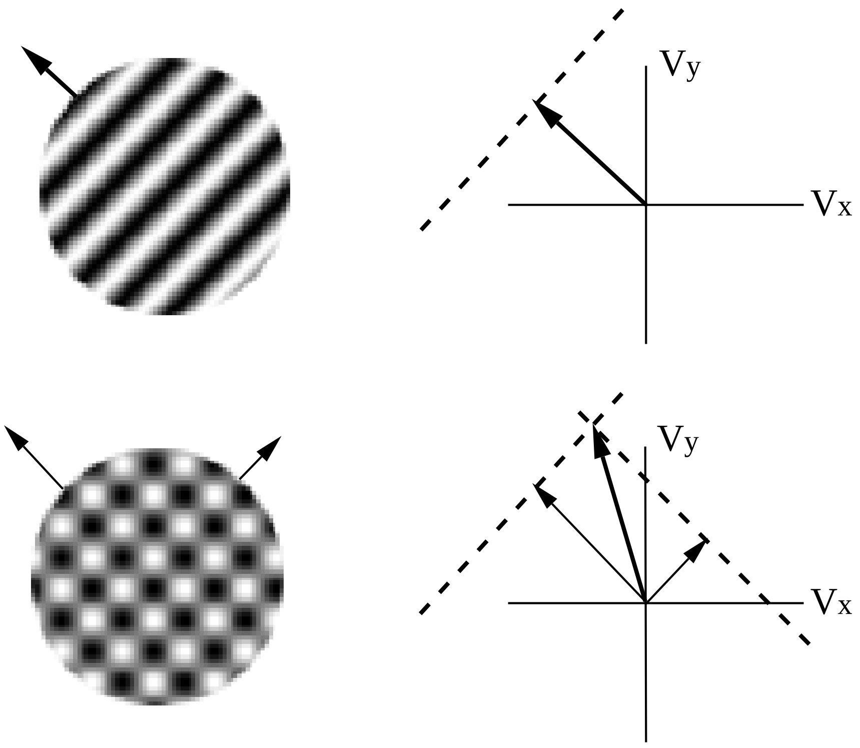


Figure 5-1: Illustration of the "intersection-of-constraints" determination of velocity. (a) A snapshot of a moving sinusoidal grating pattern. (b) Velocity-space description of the motion of the grating. The stimulus is consistent with a set of velocities lying on the dashed constraint line, which is parallel to the grating orientation with distance to the origin equal to the grating normal velocity (indicated by the arrow). (c) Snapshot of a moving plaid (sum of two moving sinusoids). (d) Velocity-space description of the motion of the plaid. The dashed constraint lines for the two individual gratings are shown. These two lines intersect at a single point which corresponds to the motion of the coherent plaid pattern.

distributed model of chapter 3 to explain these deviations.

Modeling the Perception of Coherently Moving Plaids

We propose that the human visual system represents velocities in distributed fashion, as in the model presented in chapter 4. In order to model the perception of coherent plaid velocities, we assume that the perceived velocity of these patterns corresponds to the mean of the representation distribution.

In particular, we use the basic probabilistic model defined by equation (3.6). Since we are only interested in the mean of the distribution, we re-parameterize the solution in terms of two values, σ_e and σ_p: [2]

$$\mu_{\vec{v}} = -\left[\frac{\mathbf{M}}{\left(\|\vec{f}_s\|^2 + \sigma_e\right)} + \sigma_p^{-1}\right]^{-1} \cdot \frac{\vec{b}}{\left(\|\vec{f}_s\|^2 + \sigma_e\right)}.$$

where $\mathbf{M}$ and $\vec{b}$ are defined as in equation (2.6), but without the summations. The parameter σ_e is a semi-saturation constant for the energy normalization, and σ_p is the prior probability variance. These were fitted to the data as described below. The prefilter used with the derivatives was a Gaussian.

We consider an experiment by Stone et. al. [90]. The authors examined the effects of contrast ratio on the perception of coherent plaid motion direction. In general, they found that the perceived direction of motion of a plaid was biased away from the IOC solution toward the normal velocity of the higher contrast grating.

The stimulus in the experiment was a superposition of moving sinusoidal gratings whose orientations were symmetrically in opposite directions from vertical. Under normal grating conditions (equal contrast, equal temporal frequency), the motion of this pattern is perceived as upward, as predicted by the IOC rule. If the relative contrast or the relative temporal frequency of the two gratings is varied, however, the perceived direction of motion changes. For a given contrast ratio, Stone et. al. used a staircase procedure to vary the relative temporal frequencies of the two gratings until the subject perceived the pattern motion to be vertical. The authors repeated this experiment for a variety of total contrasts (defined as the sum of the two grating contrasts), plaid angles (the magnitude of the grating orientation relative to vertical), spatial frequencies, and mean temporal frequencies.

The model was made to perform the same task as the human subjects: for a given contrast ratio, we adjusted the relative temporal frequency of the two gratings until the estimated velocity was vertical. In figure 5-2, we present a comparison of the Stone et. al. data with

[2]We have eliminated one degree of freedom that served only to modify the magnitude of the covariance matrix.

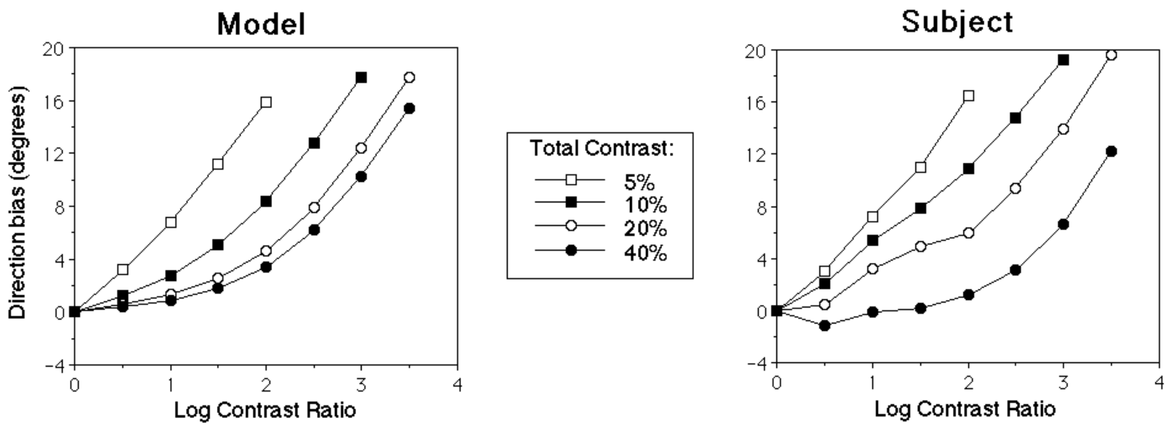


Figure 5-2: Comparison of human subject data (replotted from [90]) and the basic model described in section 3.1. The four curves correspond to different values of total contrast.

data computed from the model. Four curves are plotted, for different levels of total stimulus contrast. Plotted is the angular deviation of the IOC velocity (as determined by the relative temporal frequency of the two gratings) from vertical. The parameters σ_e and σ_p in the model were determined using a least-squares criteria to a subset of the data points from the four curves. Resulting values were $\sigma_e = 0.0209$ and $\sigma_p = 0.0021$.

The plots are seen to be quite similar in form. The model does show a slightly more pronounced bias at the highest total contrast of 40%. For small contrast ratios at this contrast, the subjects actually saw a motion bias toward the *lower* contrast grating. This somewhat counterintuitive result is not predicted by the model.

We also simulated other experiments from the same paper in which the authors varied the plaid angle and the spatial and mean temporal frequency of the gratings. These are shown in figures 5-3,5-4 and 5-5. Again, the behavior of the model is quite similar to the human observers. Figure 5-5 does, however, contain another curve with substantial bias toward the *lower* contrast grating. We were unable to duplicate this behavior in any of our simulations.

We have also modeled data by Ferrara and Wilson [27] on the perceived direction of coherent plaids. Perceived directions were measured by comparison with a one-dimensional grating. Measurements were only made for three types of plaid. Type I-S is a symmetric plaid: each grating has the same spatial and temporal frequency. Type I-A is an asymmetric plaid: the two gratings have different temporal frequencies, but the angle of the IOC velocity vector is between those of the two grating normal velocities. Type II is an asymmetric plaid in which the IOC velocity vector lies outside the angular region between the two component normal velocities. In figure 5-6, we show a bar plot of the direction bias for subjects and for our model.

Finally, we modeled data from another Ferrara and Wilson paper [28], in which they mea-

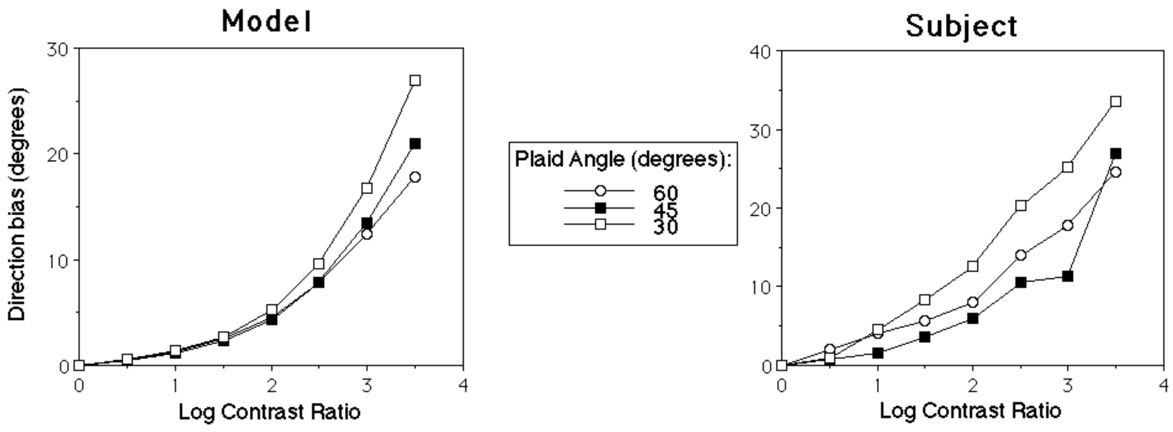


Figure 5-3: Comparison of human subject data (replotted from [90]) and the model. The three curves correspond to different values of plaid angle.

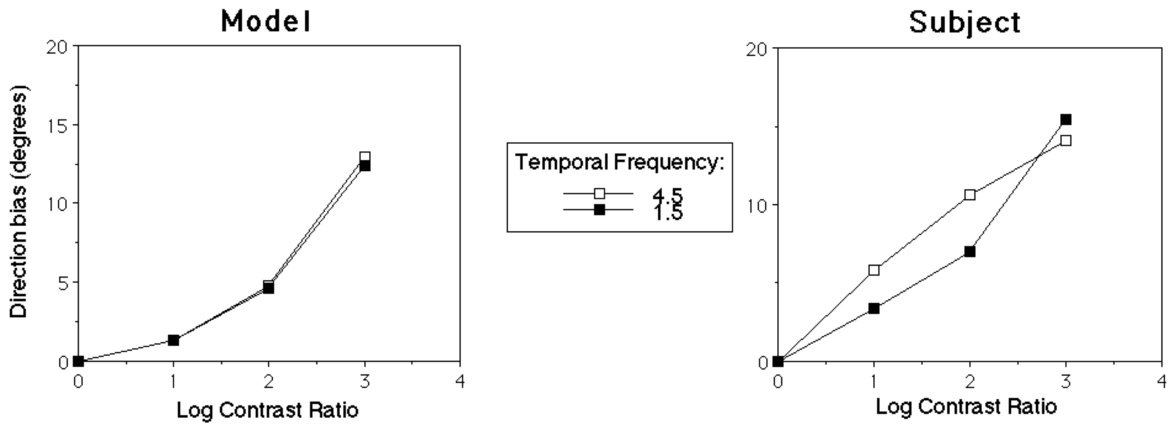


Figure 5-4: Comparison of human subject data (replotted from [90]) and the model. The two curves correspond to different values of average temporal frequency.

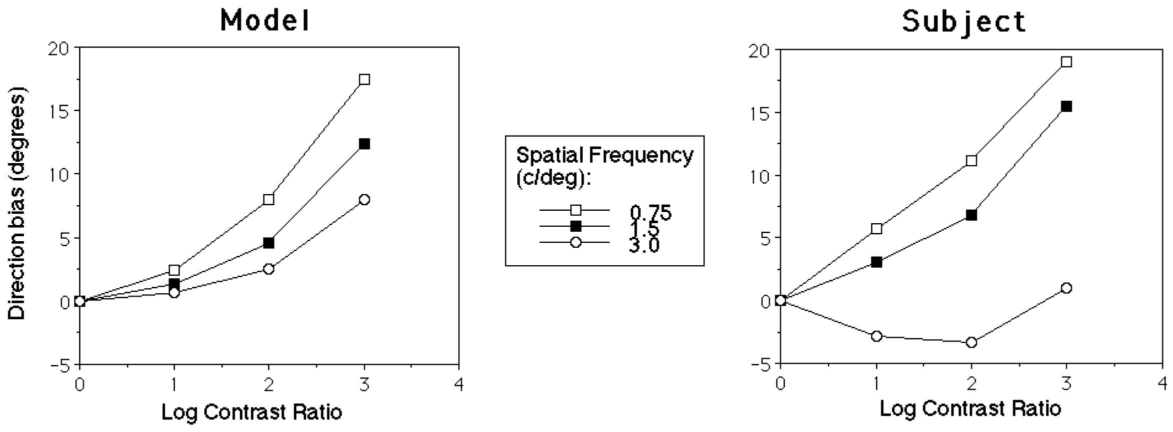


Figure 5-5: Comparison of human subject data (replotted from [90]) and the model. The three curves correspond to different values of grating spatial frequency.

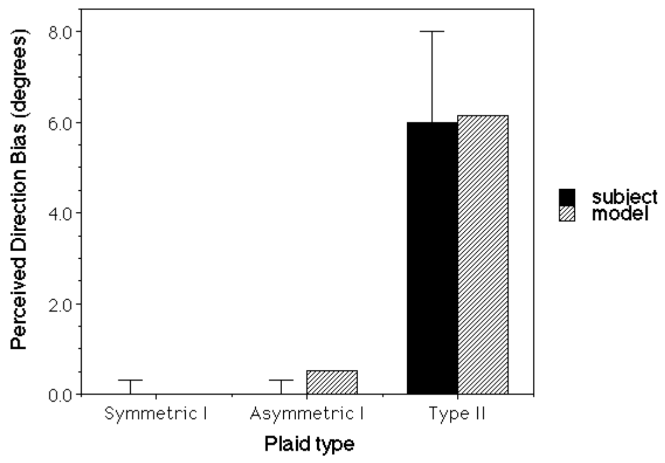


Figure 5-6: Comparison of human subject data (replotted from [27]) and the basic model described in section 3.1. Shown is perceived direction bias for three different types of sinusoidal plaid pattern (see text).

sure the perceived speed of coherent plaids as a function of plaid angle. The subject's task was to compare the speed of a plaid to a fixed reference grating (both were moving upward). At each plaid angle, the temporal frequency of the plaid components was adjusted to find the point at which the plaid speed matched the reference. As a measure of the plaid speed relative to the IOC prediction, They plot the perceived plaid speed (i.e., the reference speed) relative to the IOC speed of the plaid. In figure 5-7, we replot their data and compare it with our model. To match the data of this experiment, we had to modify the parameter values of the model. In particular, we chose values of $\sigma_e = 0.334$ and $\sigma_p = 0.008$. The plots in figure 5-7 are seen to be in good agreement with the subject data.

5.2 Physiology

Researchers have studied the processing of visual information in single neurons since the late 30s. One of the most striking features of the mammalian visual system is its organization into functionally and anatomically defined stages. A large volume of literature describes the successive set of stages of processing that take place when light enters the eye. Photoreceptors in the retina act as transducers, converting visible light into graded electrical signals. These are processed by a sequence of cell layers within the retina whose output is sent through the optic nerve, eventually reaching the cell layers of the cerebral cortex.

The classic physiological paradigm for investigating the functionality of the system is to measure, through electrode recordings, the response properties of individual cells in different parts of the system as the animal is exposed to specific stimuli. This sort of data poses difficulties for computational modeling. The models for motion analysis presented earlier are

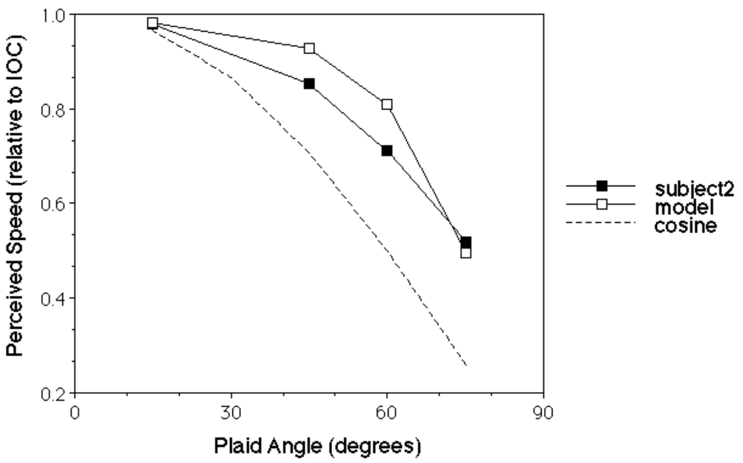


Figure 5-7: Comparison of human subject data (replotted from [27]) and the model. Plotted is perceived speed bias as a function of plaid angle. The dashed line corresponds to the speed perceived by a mechanism that simply matches temporal frequencies of the constituent gratings to that of the reference grating.

derived to produce a particular *distribution* of responses to stimuli, and the detailed "tuning" of the individual mechanisms is not uniquely determined. For example, we showed that the general filtering model for computing optical flow can be equally well implemented in terms of separable or oriented filters, and that the choice of prefilter is somewhat arbitrary. Even worse, we do not require the use of the *same* prefilter at every spatial location. What is required is a particular *relationship* between filters at the same location.

A Two-Stage Model

For the purposes of this thesis, we will propose a correspondence of the behavior of certain classes of cell with computational portions of the models presented in previous chapters, and we will discuss some of the fundamental implications of such a correspondence. In particular, we will only be concerned with the relationship between the parameters of an input stimulus to the visual system, and the output (measured as average firing rate) of a given cell. We will not model any details of the single-cell computation. Furthermore, we will model only the *steady-state* behavior of the system.

A number of researchers have begun to converge on a two-stage model for the computation of motion information in visual cortex [3, 1, 39, 101, 37, 102, 44, 85]. As in the computer vision literature, these models all seem to have a similar flavor. Linear filters are used to extract particular subbands of spatio-temporal information. These outputs are quadratically combined to produce local Fourier energy estimates. These are then combined to represent velocity. Physiologically, these two stages are typically assigned to areas V1 and V5 (also known as "MT") in the visual cortex, although the existence of populations of velocity-tuned cells is somewhat hypothetical.

There is a very large literature on the behavior of so-called "simple cells" and "complex cells" in layer V1 of the visual cortex. The standard view of simple cells is that they behave like oriented linear filters [48, 19]. There are two noticeable departures from linearity. The first is that the cell responses are rectified: cell firing rates are by definition positive, and simple cells have a fairly low background firing rate. The second is that the outputs of the cells do not continue to grow linearly with input contrast. Rather, the response saturates at high contrasts. Several researchers have suggested that this saturation may be achieved by normalizing the responses with respect to stimulus contrast.

Heeger [42, 41] has modeled an extensive amount of physiological data with a model based on squaring the cell outputs and dividing by the sum of the squares of a population of cells and a semi-saturation constant. This computation is identical to that of equation (3.8), in which directional linear measurements are squared and divided by a sum of squared directional measurements plus the constant λ_2. A similar model has been proposed by Albrecht and Geisler [6].

The "complex cells" in area V1 have similar behavior to simple cells, except that they are not sensitive to the phase (or symmetry) of the stimulus. These cells have been modeled as a quadrature combination of simple cell responses, corresponding to equation (3.11) [70, 1, 72].

More recently, authors have studied the behavior of cells in the cortical area known as V5 or "MT". There is growing evidence that some of the cells are tuned for speed and direction of motion [60, 66, 58, 7].

We propose a two-stage model based on the "donut mechanisms" developed in the previous chapter. The model computes a distributed representation of motion in two stages, as illustrated in figure 1-6. The first stage is computed via a set of spatio-temporally oriented linear operators, whose outputs are squared and normalized. The second stage is computed as a linear combination of the first stage outputs, which are again squared and normalized.

The parameters of the model are as follows:

1. The linear filters. We assume these are directional cosines raised to a power in the frequency domain, and multiplied by the prefilter transfer function. We also assume that the prefilter has "cylindrical" symmetry in the spatio-temporal frequency domain (i.e., it is bandpass spatially, and lowpass temporally).

2. The gain control parameters for the first stage: σ_e, and σ_p.

3. We assume that the linear weighting of the second stage is as described in section 4.2.

4. Gain control parameter, σ_v for the second stage.

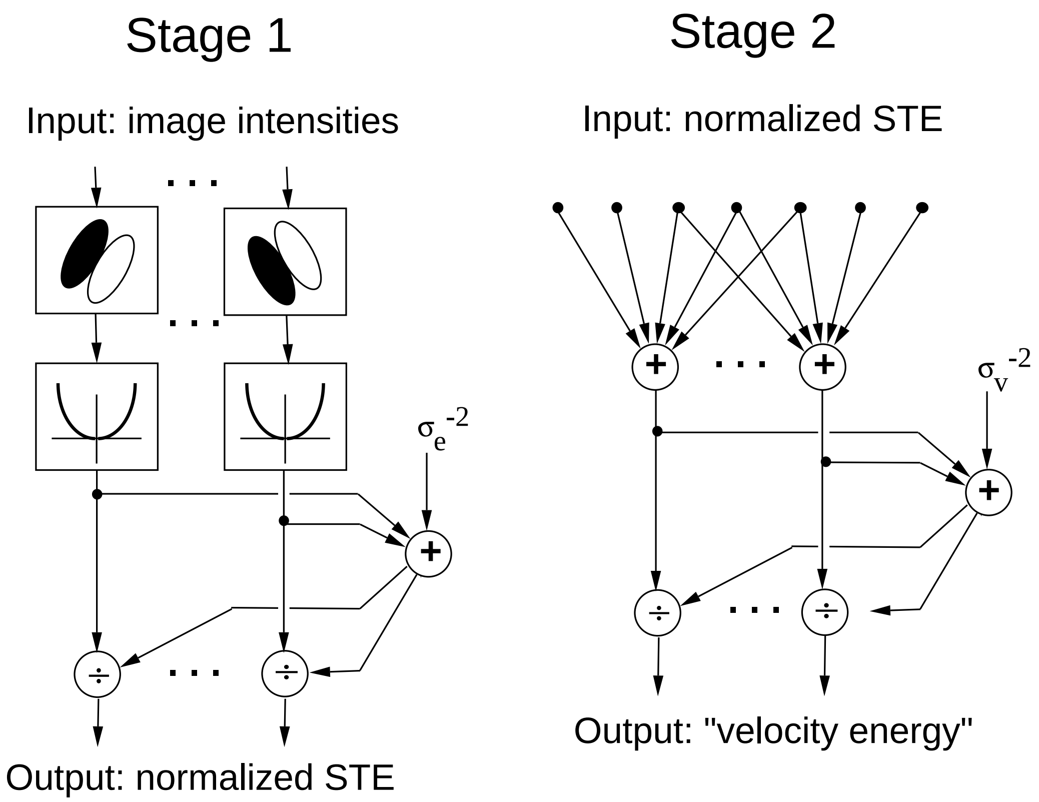


Figure 5-8: A two-stage model for motion representation. The first stage computes spatio-temporal energy via linear spatio-temporally oriented operators. The linear outputs are squared and normalized by their own sum, augmented by a semi-saturation constant σ_e. The second stage computes "velocity" energy. Again a set of linear weights are used to combine the inputs (i.e., the spatio-temporal energies of the previous stage), and again these are normalized. An additional squaring non-linearity may also be included.

5. A prior function on velocity, $\rho(v)$, which we set to unity for the purposes of this paper.

We view this as the basic model: one can easily extend it by including additional nonlinearities on the output stages of each of the layers. An example is the exponential used in the probabilistic model of section 3.1. Furthermore, the velocity-tuned second stage can be modified to be sensitive to other types of motion such as divergence or rotation.

Stimuli

How can we tell, by examining the response of a cell to parameterized stimuli, what the cell is doing? Our working hypothesis is that a typical cell is tuned simultaneously for *many* different parameters. For some of these parameters, a given cell may be narrowly tuned, and for others it may be responsive over the entire range of the parameter. Our principle example would be a cell tuned narrowly for velocity. One might expect such a cell to be fairly insensitive to changes in stimulus color, local spatial frequency content, orientation, and even contrast.

Researchers have used a variety of stimuli to probe the behavior of cortical cells. Typically, experiments are conducted with one-dimensional parametric variations. That is, one parameter is varied while all others are held fixed, usually at values that are optimal for the cell. Early experiments by Hubel and Wiesel used oriented bars as stimuli, mapping the response of cells as a function of the orientation of the bars. More recent experiments, influenced by linear systems theory, map the response of simple cells with drifting sinusoids. Sinusoid plaids and fields of random dots have also been used. In a particularly influential set of experiments, Movshon et. al. [60] used gratings and plaids as stimuli for testing complex and MT cells. They found two populations of cells: one that responded to the motion of the individual gratings of the plaid, and the other that responded to the motion of the overall pattern.

We conduct a small thought experiment to characterize the difference in response between an idealized spatio-temporal energy operator and an idealized velocity operator, and to illustrate the two "spaces" of interest: the spatio-temporal frequency domain, and the plane of two-dimensional velocities. Imagine an operator that is well-tuned to patterns moving at a particular velocity. In the frequency domain, the operator responds to energy lying an a plane. Imagine that we show this operator drifting sinusoids of a fixed spatial frequency, but varying orientation and temporal frequency. In the Fourier domain, these sinusoids live on a cylinder. The intersection of the cylinder with the plane, an oblique ellipse, corresponds to the set of sinusoids that might cause the operator to respond. This is illustrated in figure 5-9.

The set of consistent sinusoids may also be plotted in two-dimensional velocity space, as shown on the right side of the figure. Here, the dark circle indicates the *normal* velocities of the consistent sinusoids. In this ideal setting, the velocity operator responds to a one-dimensional set of sinusoids. On the other hand, its tuning for a moving pattern that is spectrally broad, such as a field of random dots, would be a point.

Similarly, we can plot the set of velocities that are consistent with a given spatio-temporal energy operator. This is shown in figure 5-10. The spatio-temporal energy operator responds to energy at two points in the spatio-temporal frequency domain, on opposite sides of the origin. An infinite set of velocity planes go through these points, as illustrated in the figure. Thus, a narrowly-tuned spatio-temporal energy operator would respond only to a particular sinusoidal stimulus, but to an entire one-dimensional set of broad-spectrum velocity patterns. This set of consistent velocities can also be plotted in two-dimensional velocity space, where it corresponds to a line, as illustrated in figure 5-10.

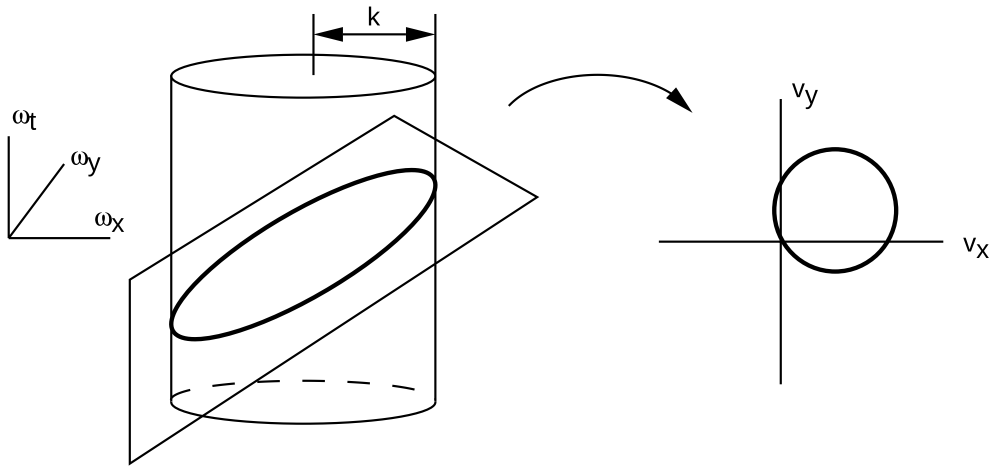


Figure 5-9: Illustration of the set of sinusoids of a fixed spatial frequency that are consistent with a given velocity-tuned unit. On the left is a plot of spatio-temporal frequency space. The cylindrical surface corresponds to all sinusoids of a given spatial frequency magnitude (wavenumber). The oblique ellipse (dark line) indicates the set of sinusoids consistent with the velocity specified by the plane. On the right is a plot of two-dimensional velocity space. The dark circle corresponds to the normal velocities of a set of sinusoids consistent with the velocity specified by the point on the circle furthest from the origin.

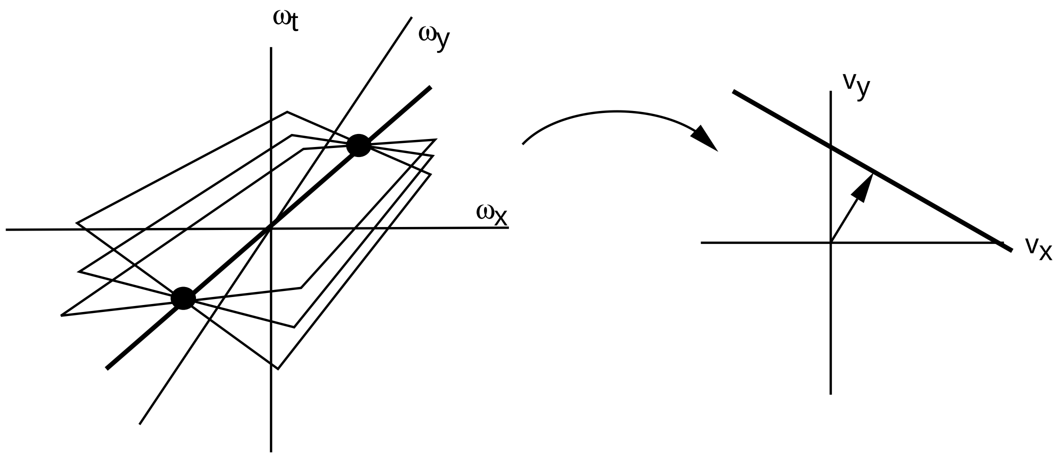


Figure 5-10: Illustration of the set of velocities planes that are consistent with a given spatio-temporal energy operator. On the left is a plot of spatio-temporal frequency space. The dark balls correspond to the tuning of the operator. There is an infinite set of planes passing through these two points, each corresponding to a velocity. On the right is a plot of two-dimensional velocity space. The dark line corresponds to the set of velocities that are consistent with the spatio-temporal operator.

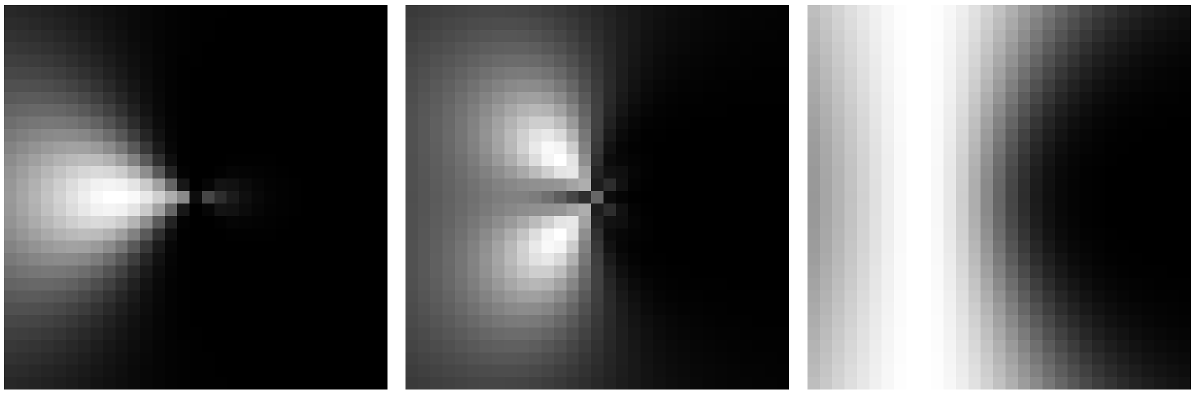


Figure 5-11: Tuning of a single spatio-temporal energy unit to a drifting sinusoid, a sinusoidal plaid, and a field of random dots. Image intensity corresponds to the strength of response.

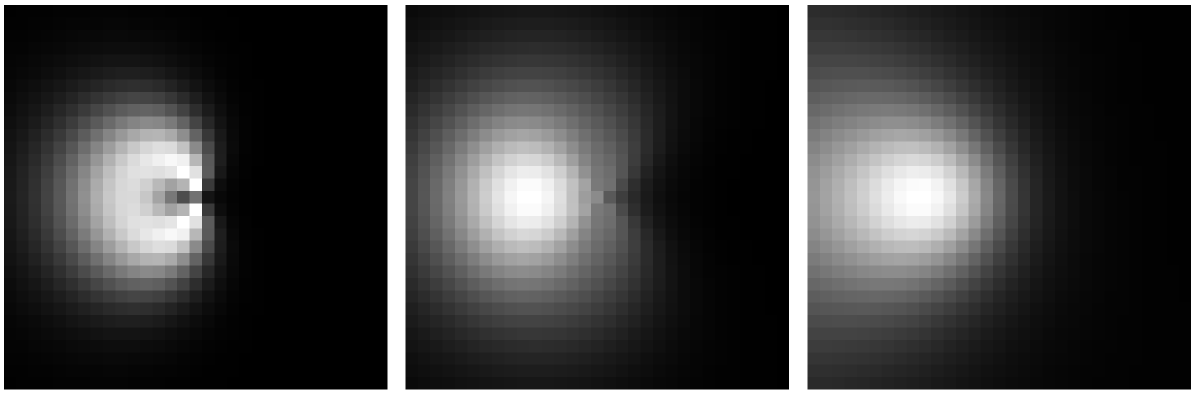


Figure 5-12: Tuning of a single "velocity energy" unit to a drifting sinusoid, a sinusoidal plaid, and a field of random dots.

Model Behavior

We now plot the response of the two stages of our "donut" model of chapter 4 to three types of stimuli: sinusoids, 90-degree plaids, and random dot patterns. For each model stage and stimulus, we show both the response of a single unit, *and* the response of a population of these units. Each of the stimuli are parameterized according to their speed and direction of motion (DOM). Thus the data are represented in a set of 12 two-dimensional images.

Figure 5-11 shows the response of a prototypical spatio-temporal energy mechanism (i.e., a model complex cell) to three different types of stimuli: a sine grating, a 90-degree plaid, and a field of random dots. The coordinates correspond to the speed and DOM of the pattern (for the grating, the DOM is the normal direction). Figure 5-12 shows the response of a prototypical velocity energy mechanism to the three different types of stimuli.

Figure 5-13 shows the *population response* of a set of spatio-temporal energy mechanisms that span the space of orientation and temporal frequency tunings.

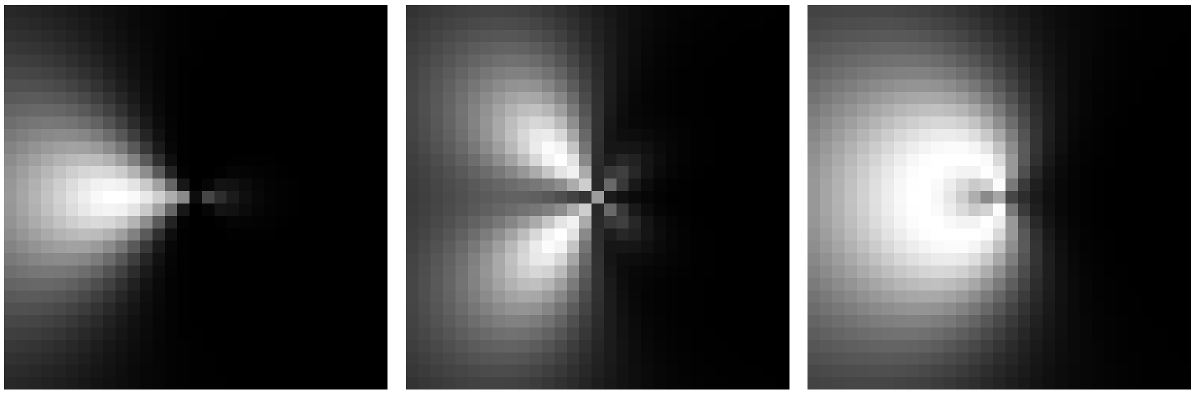


Figure 5-13: Population response of a set of spatio-temporal energy units to a drifting sinusoid, a sinusoidal plaid, and a field of random dots.

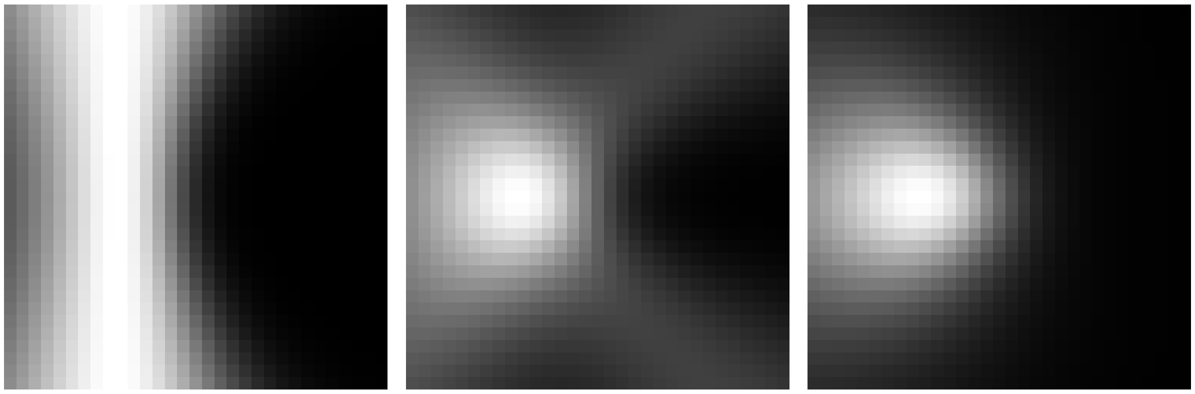


Figure 5-14: Population response of a set of "velocity energy" units to a drifting sinusoid, a sinusoidal plaid, and a field of random dots.

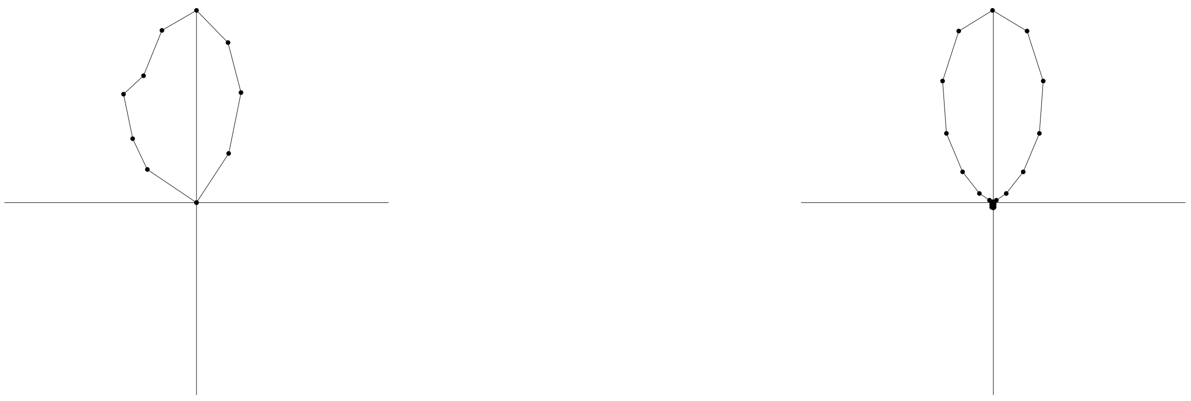


Figure 5-15: Comparison of the spatio-temporal energy stage of the model to simple cell orientation tuning data (replotted from Movshon et. al. [60]. The stimulus is a sinusoidal grating. Response of the cell is shown on the left in a polar plot as a function of stimulus orientation. Response of the model is shown on the right (note that this is simply a circular slice through the leftmost plot in 5-11).

We can see some similarity with the idealized responses of figures 5-9 and 5-10, but it is apparent that the actual operators are not nearly as distinctive as the idealization.

Nevertheless, differences in tuning between the STE and velocity cells are apparent. For example, the STE unit has a distinctive bimodal response to a plaid stimulus, whereas the velocity unit has a more unimodal response. The velocity unit has a more narrowly tuned response to dot stimuli, and the STE unit has a more narrowly tuned response to sinusoidal stimuli. Furthermore, the plots predict that as the speed of a random dot stimulus increases, a plot of STE response vs. DOM should go from unimodal to bimodal. Similarly, as the speed of a grating increases, a plot of velocity unit response vs. DOM should start out bimodal and become unimodal. These predictions can easily be tested in single cell recording experiments.

In figures 5-15 through 5-18, we show comparisons of the two stages of the model with data from Movshon et. al. [60]. Depicted are polar plots of the response as a function of the stimulus DOM. Figure 5-19 shows a comparison of the final stage of the model with data recorded in MT by Maunsell and van Essen [58]. Depicted is a plot of response versus the log of the stimulus speed, relative to the speed that produces maximal response. The agreement between the cells and the model is excellent for all of these plots.

5.3 Summary

We have discussed the distributed models of chapters 3 and 4 in the context of biological visual systems. The simple probabilistic model of chapter 3 has been successfully used to quantitatively mimic psychophysical experiments on moving plaid perception. We have also shown qualitative consistency of the "donut mechanisms" with physiological data.

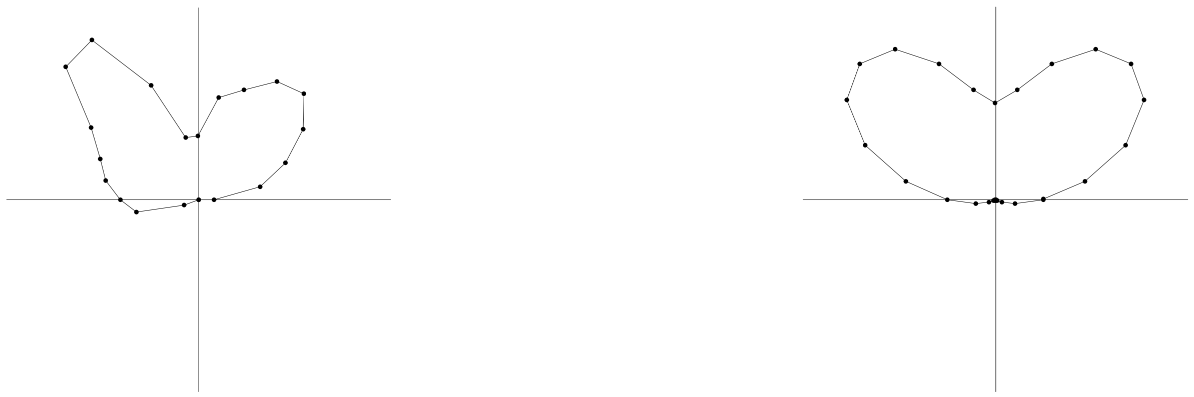


Figure 5-16: Comparison of the spatio-temporal energy stage of the model to complex cell data (replotted from Movshon et. al. [60]. The stimulus is a ninety degree sinusoidal plaid. Response of the cell is on the left, response of the model is shown on the right.

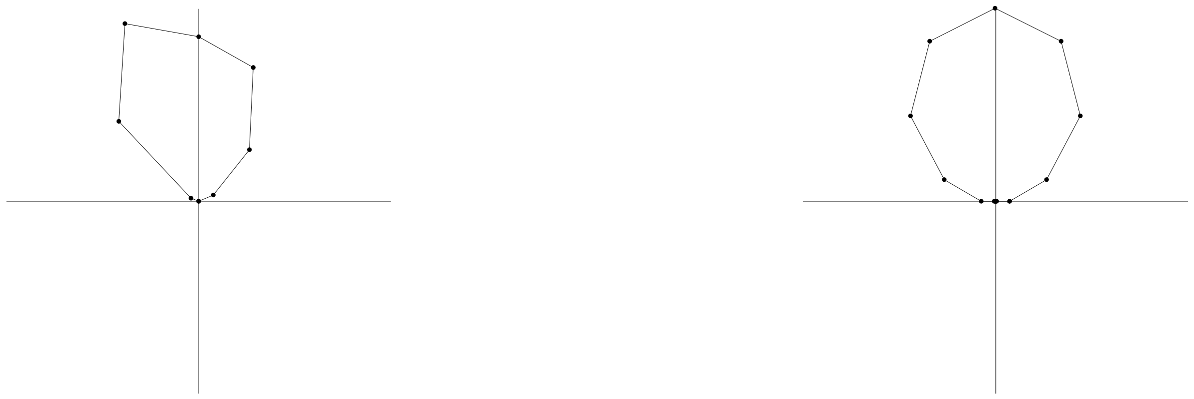


Figure 5-17: Comparison of the velocity energy stage of the model to MT pattern cell orientation tuning data (replotted from Movshon et. al. [60]. The stimulus is a sinusoidal grating. Response of the cell is on the left, response of the model is shown on the right.

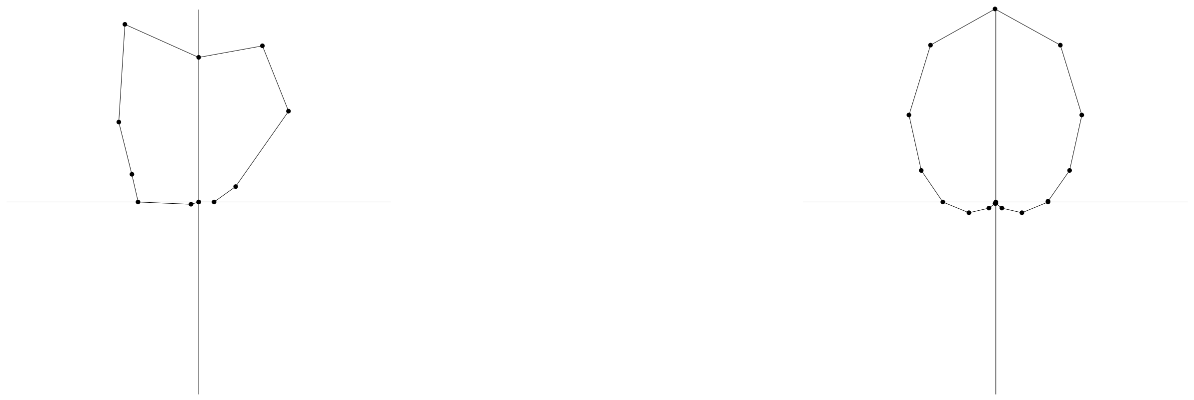


Figure 5-18: Comparison of the velocity energy stage of the model to MT pattern cell data (replotted from Movshon et. al. [60]. The stimulus is a ninety degree sinusoidal plaid. Response of the cell is on the left, response of the model is shown on the right.

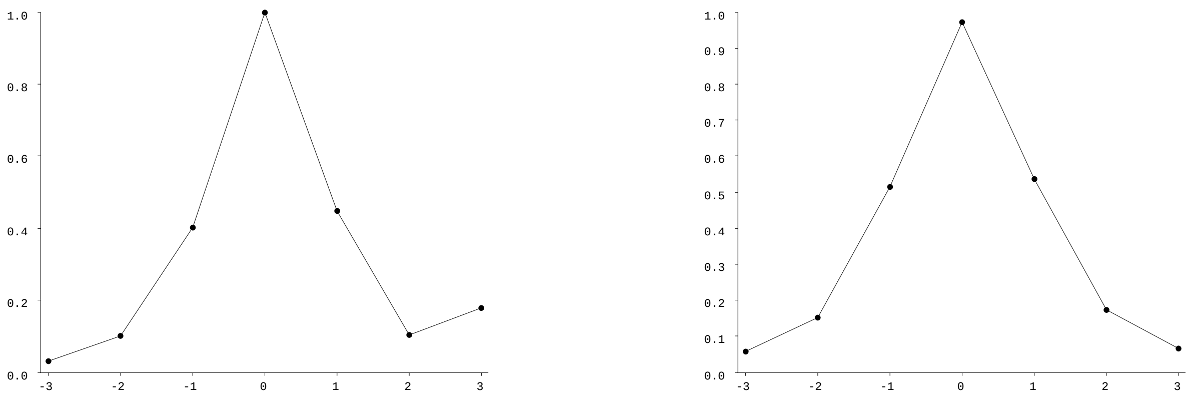


Figure 5-19: Comparison of the response of the velocity energy stage of the model to MT pattern cell data (replotted from Maunsell and Van Essen [58]). The stimulus is an oriented slit of light, moving in the cell's preferred direction. Plotted is the response as a function of log speed, relative to the optimal speed. The cell response is shown on the left, the model on the right.

Conclusions

This thesis has explored the topic of visual motion analysis and representation. We have unified a large number of standard algorithms for motion computation, and developed a set of measurement techniques based on directional cosine filters. These filters have a number of advantageous properties when applied to motion analysis:

- They are derived from the gradient approach, and therefore offer an analytic solution when applied to the velocity field estimation problem.
- They may be viewed as performing a planar regression in spatio-temporal frequency. They do not introduce systematic biases depending on the spatial frequency content of the signal.
- They are efficient to implement. With the proper choice of prefilter, separable implementations are possible.
- They may be used in place of quadrature energy computations.
- They are steerable.

We developed a weighted least-squares design strategy in the frequency domain and demonstrated that the resulting filters produced stable and accurate velocity estimates.

In chapter 3, we developed a family of probabilistic algorithms for computing optical flow. The algorithms are based on three independent Gaussian random variables. The resulting probability distribution over velocity is also Gaussian, and may thus be parameterized by a mean and covariance that are computed analytically. The computation is efficient relative to typical optical flow algorithms (due to assumptions of independence).

We develop a probabilistic coarse-to-fine strategy – a Kalman filter over scale – for defeating temporal aliasing, and demonstrate the behavior of the algorithm on a number of synthetic and real image sequences. The results are shown to be superior to those published in a recent comparison of optical flow techniques [10]. We also demonstrate that the estimated covariance matrices provide useful information about the quality of the velocity estimates. The most noticeable failure of the algorithm occurs at occlusion boundaries, as is to be expected. Future implementations could include consistency measures, as described at the end of chapter 3, although this will probably be computationally expensive.

We also develop a distributed representation of motion based on *angular* frequency regression. This algorithm operates by combining a set of directional derivative energies with directions lying on a plane. Spectrally, this corresponds to weighting the input signal with a smooth ring: thus we have called this computation a "donut" mechanism. We demonstrate that these mechanisms are capable of representing multiple motions in a local region. In future work, we would like to make use of this representation to extrapolate frames and to segment moving imagery, even in the presence of transparency.

In the last chapter, we discussed the use of these models for biological modeling. The distributed models of chapters 3 and 4 are readily mapped onto physiology and are able to account for psychophysical perception of plaid patterns. Future work will focus on a more detailed physiological model that accounts for a larger set of data.

Bibliography

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284–299, February 1985.

[2] E. H. Adelson and J. R. Bergen. Perceptual dimensions of spatio-temporal vision. *Investigative Opthalmology and Visual Science Supplement (ARVO)*, 27:141, 1986.

[3] E. H. Adelson and J. A. Movshon. Phenomenal coherence of moving visual patterns. *Nature*, 300(5892):523–525, 1982.

[4] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Pat. Anal. Mach. Intell.*, 4:384–401, 1985.

[5] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images — a review. *Proc. IEEE*, 76:917–935, 1988.

[6] D. G. Albrecht and W. S. Geisler. Motion sensitivity and the contrast-response function of simple cells in the visual cortex. *Visual Neuroscience*, 7:531–546, 1991.

[7] T. D. Albright. Direction and orientation selectivity of neurons in visual area mt of the macaque. *J Neurophysiol*, 52(6):1106–1130, December 1984.

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

[9] H. B. Barlow. Single units and sensation: a neural doctrine for perceptual psychology? *Perception I*, pages 371–394, 1972.

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

[11] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Willsky. Modeling and estimation of multiresolution stochastic processes. Technical Report CICS-P-283, MIT Center for Intelligent Control Systems, February 1991.

[12] R. Battiti, E. Amaldi, and C. Koch. Computing optical flow across multiple scales: an adaptive coarse-to-fine strategy. *Intl. J. Comp. Vis.*, 6(2):133–145, 1991.

[13] 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.

[14] M. Black and P. Anandan. Robust dynamic motion estimation over time. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–302, Maui, June 1991.

[15] M. J. Black. A robust gradient method for determiningn optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, Champagne-Urbana, June 1992.

[16] 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.

[17] P. J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16:20–51, 1981.

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

[19] F. W. Campbell, G. F. Cooper, and C. Enroth-Cugell. The angular selectivity of visual cortical cells to moving gratings. *J. Physiology (London)*, 198:237–250, 1968.

[20] K. C. Chou, A. S. Willsky, A. Benveniste, and M. Basseville. Recursive and iterative estimation algorithms for multi-resolution stochastic processes. Technical Report LIDS-P-1857, MIT Laboratory for Information and Decision Sciences, March 1989.

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

[22] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169, July 1985.

[23] W. Enkelmann and H. Nagel. Investigation of multigrid algorithms for estimation of optical flow fields in image sequences. *Comp. Vis. Graphics Image Proc.*, 43:150–177, 1988.

[24] M. Fahle and T. Poggio. Visual hyperacuity: spatiotemporal interpolation in human vision. *Proceedings of the Royal Society of London, B*, 213:451–477, 1981.

[25] M. Feder and E. Weinstein. Parameter estimation of superimposed signals using the EM algorithm. *IEEE Trans. Acoust. Speech Signal Proc.*, 36(4):477–489, April 1988.

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

[27] V. P. Ferrara and H. R. Wilson. Perceived direction of moving two-dimensional patterns. *Vis. Res.*, 30(2):273–287, 1990.

[28] V. P. Ferrara and H. R. Wilson. Perceived speed of moving two-dimensional patterns. *Vis. Res.*, 31(5):273–287, 1990.

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

[30] D. J. Fleet and A. D. Jepson. A cascaded filter approach to the construction of velocity selective mechanisms. Technical Report RBCV-TR-84-6, Department of Computer Science, University of Toronto, 1984.

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

[32] D. Gabor. Theory of communication. *J. IEE*, 93:492–457, 1946.

[33] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, 1989.

[34] S. Geman and D. Geman. Stochastic relaxation, gibbs distibutions, and bayesian restoration of images. *IEEE Pattern Analysis and Machine Intelligence*, 6:721–741, November 1984.

[35] J. J. Gibson. *The Perception of the Visual World.* Houghton Mifflin, Boston, MA, 1950.

[36] B. Girod and D. Kuo. Direct estimation of displacement histograms. In *OSA meeting on Image Understanding and Machine Vision*, Cape Code, Mass, 1989.

[37] 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.

[38] R. M. Haralick and J. S. Lee. The facet approach to optic flow. In L. Baumann, editor, *Proceedings, Image Understanding Workshop*, pages 84–93. Science Applications, Arlington, VA, 1983.

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

[40] D. J. Heeger. Optical flow using spatiotemporal filters. *Intl. J. Comp. Vis.*, pages 279–302, 1988.

[41] D. J. Heeger. Half-squaring in responses of cat simple cells. *Visual Neuroscience*, 9, 1992. In press.

[42] D. J. Heeger. Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, 9, 1992. In press.

[43] D. J. Heeger, A. D. Jepson, and E. P. Simoncelli. Model of cell responses in visual area MT. In *Proceedings IEEE Workshop on Visual Motion*, Princeton, October 1991.

[44] D. J. Heeger and E. P. Simoncelli. Model of visual motion sensing. In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots.* Cambridge University Press, 1992.

[45] E. C. Hildreth. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23(3):309–355, 1984.

[46] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

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

[48] D. Hubel and T. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *J. Physiology (London)*, 160:106–154, 1962.

[49] D. H. Johnson. The application of spectral estimation methods to bearing estimation problems. *Proc. IEEE*, 70:1018–1028, September 1982.

[50] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Pat. Anal. Mach. Intell.*, 9(2):229–244, 1987.

[51] 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.

[52] F. L. Kooi, R. L. DeValois, and T. K. Wyman. Perceived direction of moving plaids. In *Invest. Opthalmol. and Vis. Sci. Suppl. (ARVO)*, volume 29, 1989.

[53] 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.

[54] 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.

[55] M. Luettgen, W. Karl, and A. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. Technical Report LIDS-P-2115, MIT Laboratory for Information and Decision Systems, 1992. Submitted, to IEEE Trans. Image Proc.

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

[57] D. Marr and S. Ullman. Directional selectivity and its use in early visual processing. *Proc. Royal Society of London B*, 211:151–180, 1981.

[58] J. H. R. Maunsell and D. C. V. Essen. Functional properties of neurons in middle temporal visual area of the macaque monkey i. selectivity for stimulus direction, speed, and orientation. *Journal of Neurophysiology*, 49(5):1127–1147, 1983.

[59] A. Mitche, Y. F. Yang, and J. K. Aggarwal. Experiments in computing optical flow with the gradient-based, multiconstraint method. *Pattern Recognition*, 20(2):173–179, 1987.

[60] J. A. Movshon. Processing of motion information by neurons in the striate and extrastriate visual cortex of the macaque. *Investigative Opthalmology and Visual Science Supplement (ARVO)*, 26:133, 1985.

[61] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Pat. Anal. Mach. Intell.*, 9:220–228, March 1987.

[62] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Pat. Anal. Mach. Intell.*, 8:565–593, Septemeber 1986.

[63] 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.

[64] K. Nakayama. Biological image motion processing: A review. *Vis. Res.*, 25:625–660, 1985.

[65] S. Negahdaripour, A. Shokrollahi, and M. A. Gennert. Relaxing the brightness constancy assumption in computing optical flow. In *Proc. IEEE Intern. Conf. Image Processing*, pages 806–810, September 1989.

[66] W. T. Newsome, M. S. Gizzi, and J. A. Movshon. Spatial and temporal properties of neurons in macaque mt. *Investigative Opthalmology and Visual Science Supplement (ARVO)*, 24:106, 1983.

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

[68] T. Poggio and W. Reichardt. Considerations on models of movement detection. *Kybernet*, 13:223–227, 1973.

[69] T. Poggio, W. Yang, and V. Torre. Optical flow: Computational properties and networks, biological and analog. In R. Durbin, C. Miall, and G. Mitcheson, editors, *The Computing Neuron*, chapter 19, pages 355–370. Addison-Wesley, Wokingham, England, 1988.

[70] D. Pollen and S. Ronner. Visual cortical neurons as localized spatial-frequency filters. *IEEE Trans. Sys. Man Cyber.*, 13:907–916, 1983.

[71] L. Quam. Hierarchical warp stereo. In *Proceedings of the DARPA Image Understanding Workshop*, September 1984.

[72] E. H. A. R C Emerson, J R Bergen. Directionally selective complex cells and the computation of motion energy in cat visual cortex. *Vision Research*, 32(2):203–218, 1992.

[73] W. Reichardt. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In W. A. Rosenblith, editor, *Sensory Communication.* Wiley, New York, 1961.

[74] A. Rougee, B. C. Levy, and A. S. Willsky. An estimation-based approach to the reconstruction of optical flow. Technical Report LIDS-P-1663, MIT Laboratory for Information and Decision Systems, April 1987.

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

[76] 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.

[77] E. P. Simoncelli. Computing optical flow from first and second order directional cosines. Vision and modeling technical report, MIT Media Laboratory, November 1992.

[78] E. P. Simoncelli. Distributed representations of image velocity. Vision and Modeling Technical Report 202, MIT Media Laboratory, October 1992.

[79] 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. Revised, March 1991.

[80] 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.

[81] E. P. Simoncelli and E. H. Adelson. Relationship between gradient, spatio-temporal energy, and regression models for motion perception. In *Investigative Opthalmology and Visual Science Supplement (ARVO)*, volume 32, 1991.

[82] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, Mauii, Hawaii, June 1991.

[83] E. P. Simoncelli and D. J. Heeger. A computational model for representation of image velocities. In *Investigative Opthalmology and Visual Science Supplement (ARVO)*, volume 34, 1993.

[84] 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.

[85] E. P. Simoncelli, D. J. Heeger, and E. H. Adelson. A computational model for perception of two-dimensional pattern velocities. In *Investigative Opthalmology and Visual Science Supplement (ARVO)*, volume 33, 1992.

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

[87] P. Sobey and M. V. Srinivasan. Measurement of optical flow by a generalized gradient scheme. *J. Opt. Soc. Am. A*, 8(9):1488–1498, September 1991.

[88] M. V. Srinivasan. Generalized gradient schemes for the measurement of two-dimensional image motion. *Biol. Cybern.*, 63:421–431, 1990.

[89] L. S. Stone and P. Thompson. Human speed perception is contrast dependent. *Vis. Res.*, 32(8):1535–1549, 1992.

[90] L. S. Stone, A. B. Watson, and J. B. Mulligan. Effect of contrast on the perceived direction of a moving plaid. *Vis. Res.*, 30(7):1049–1067, 1990.

[91] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301, December 1990.

[92] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Pat. Anal. Mach. Intell.*, 8:129–139, 1986.

[93] P. Thompson. Perceived rate of movement depends on contrast. *Vis. Res.*, 22:377–380, 1982.

[94] O. Tretiak and L. Pastor. Velocity estimation from image sequences with second order differential operators. In *IEEE*, pages 16–19, 1984.

[95] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biol. Cybern.*, 60:79–87, 1988.

[96] J. P. H. van Santen and G. Sperling. Temporal covariance model of human motion perception. *J. Opt. Soc. Am. A*, 1:451–473, 1984.

[97] A. Verri and T. Poggio. Motion field and optical flow: Qualitative properties. *IEEE Pat. Anal. Mach. Intell.*, pages 490–498, 1989.

[98] A. B. Watson and A. J. Ahumada. A look at motion in the frequency domain. In J. K. Tsotsos, editor, *Motion: Perception and representation*, pages 1–10. 1983.

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

[100] A. M. Waxman and K. Wohn. Contour evolution, neighbourhood deformation, and global image flow: Planar surfaces in motion. *Inter. J. Robotics Res.*, 4:95–108, 1985.

[101] L. Welch. The perception of moving plaids reveals two motion-processing stages. *Nature*, 337:734–736, 1989.

[102] H. R. Wilson, V. P. Ferrera, and C. Yo. A psychophysically motivated model for two-dimensional motion perception. *Visual Neuroscience*, 9:79–97, 1992.