September 1987

Models for Motion Perception

David J. Heeger
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-87-91.

Models for Motion Perception

Abstract

As observers move through the environment or shift their direction of gaze, the world moves past them. In addition, there may be objects that are moving differently from the static background, either rigid-body motions or nonrigid (e.g., turbulent) ones. This dissertation discusses several models for motion perception. The models rely on first measuring motion energy, a multi-resolution representation of motion information extracted from image sequences.

The image flow model combines the outputs of a set of spatiotemporal motion-energy filters to estimate image velocity, consonant with current views regarding the neurophysiology and psychophysics of motion perception. A parallel implementation computes a distributed representation of image velocity that encodes both a velocity estimate and the uncertainty in that estimate. In addition, a numerical measure of image-flow uncertainty is derived.

The egomotion model poses the detection of moving objects and the recovery of depth from motion as sensor fusion problems that necessitate combining information from different sensors in the presence of noise and uncertainty. Image sequences are segmented by finding image regions corresponding to entire objects that are moving differently from the stationary background.

The turbulent flow model utilizes a fractal-based model of turbulence, and estimates the fractal scaling parameter of fractal image sequences from the outputs of motion-energy filters. Some preliminary results demonstrate the model's potential for discriminating image regions based on fractal scaling.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-87-91.

MODELS FOR MOTION PERCEPTION

David J. Heeger

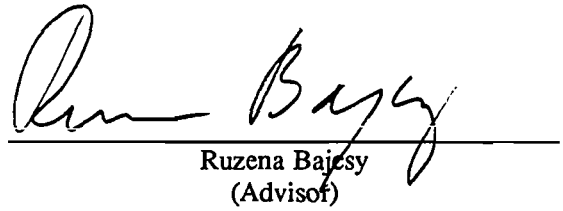
A Dissertation

in

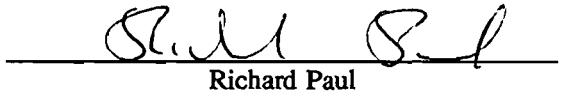
Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy.

September 1987



Ruzena Bajcsy
(Advisor)



Richard Paul
(Graduate Group Chair)

Acknowledgements

I particularly wish to thank Ruzena Bajcsy for her encouragement, and Sandy Pentland for his invaluable help and advice. As a result of their efforts, I have had the unique opportunity of doing this research both at the GRASP Laboratory at the University of Pennsylvania and at the Aritificial Intelligence Center at SRI International. I can only hope to work in such stimulating and amiable environments in the years to come.

Special thanks to Ted Adelson for motivating the image flow research and for providing the psychophysical data on coherence of sine-grating plaids, to Greg Hager for his discussions on probability and estimation theory and for his collaboration on the research presented in Chapter 5, to Jack Nachmias for his dialogs on psychophysics, to Tony Movshon for his dialogs on physiology, to Lynn Quam for image-calc and for generating the Yosemite fly-through image sequence, and to Richard Billington for keeping the Symbolics machine happy.

I also wish to thank all of my friends and fellow graduate students at the University of Pennsylvania (David Smitley, Eric Krotkov, Franc Solina, Amy Felty, Amy Zwarico, Hugh Durrant-Whyte, David Talton, Mark Turner, Stephane Mallat, Alberto Izaguirre), and my friends and colleagues at SRI International (David Marimont, Grahme Smith, Harlyn Baker, Yvan Leclerc, Steve Barnard, Pascal Fua, Andy Hanson).

Mostly I want to thank my parents, Alan and Ruth Heeger. The benefits of an academic lifstyle were always clear to me in our home so I always knew I would go to graduate school for a Ph.D.

This research is supported at the University of Pennsylvania by contracts ARO DAA6-29-84-k-0061, AFOSR 82-NM-299, NSF MCS-8219196-CER, NSF MCS 82-07294, AVRO DAABO7-84-K-FO77, and NIH 1-RO1-HL-29985-01, at SRI International by contracts NSF

DCR-83-12766, DARPA MDA 903-83-C-0027, DARPA DACA76-85-C-0004, and by the Systems Development Foundation.

ABSTRACT

MODELS FOR MOTION PERCEPTION

David J. Heeger
Ruzena Bajcsy (advisor)

As observers move through the environment or shift their direction of gaze, the world moves past them. In addition, there may be objects that are moving differently from the static background, either rigid-body motions or nonrigid (e.g., turbulent) ones. This dissertation discusses several models for motion perception. The models rely on first measuring motion energy, a multiresolution representation of motion information extracted from image sequences.

The image flow model combines the outputs of a set of spatiotemporal motion-energy filters to estimate image velocity, consonant with current views regarding the neurophysiology and psychophysics of motion perception. A parallel implementation computes a distributed representation of image velocity that encodes both a velocity estimate and the uncertainty in that estimate. In addition, a numerical measure of image-flow uncertainty is derived.

The egomotion model poses the detection of moving objects and the recovery of depth from motion as sensor fusion problems that necessitate combining information from different sensors in the presence of noise and uncertainty. Image sequences are segmented by finding image regions corresponding to entire objects that are moving differently from the stationary background.

The turbulent flow model utilizes a fractal-based model of turbulence, and estimates the fractal scaling parameter of fractal image sequences from the outputs of motion-energy filters. Some preliminary results demonstrate the model's potential for discriminating image regions based on fractal scaling.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Models for Motion Perception

The world we live in is constantly in motion — observers (either biological organisms or a computer beings) who depend on visual perception to gain an understanding of the environment must be able to interpret visual motion. Active observers who are moving their head and eyes in order to better perceive the environment rely particularly on motion analysis. Some of the important functions of motion perception are: (1) to act as an early warning system; (2) to allow an observer to track the location of moving objects and recover their three-dimensional structure; (3) to help an observer determine his own movement (egomotion) through the environment; (4) to help an observer divide the visual field into meaningful segments (e.g., moving vs. stationary or rigid vs. nonrigid); (5) to help an observer classify objects (e.g., as inanimate objects or as biological organisms).

The perception of visual motion does not depend on prior interpretation or recognition of shape and form. However, it does depend on there being motion information, i.e., changes in intensity over time throughout the visual field. Without texture, a perfectly smooth moving surface yields an image sequence in which most local regions do not change over time. But in a highly textured world (e.g., natural outdoor scenes with trees and grass), there is motion information throughout the visual field.

The goal of this research is an analysis of visual motion that is at its best performance for

complex, outdoor, natural scenes. This is in sharp contrast to many computer vision efforts to date that are restricted to a world populated by smooth objects, a sort of "Play-Doh" world [20] that is not much more general than the blocks world.

It is generally believed that the analysis of visual motion procedes in two stages. The first stage is the extraction of two-dimensional motion information (direction of motion, speed, displacement) from image sequences. The second stage is the interpretation of image motion. Early computer vision research focused on interpreting image motion as biological motion [10,68,105]. Computer vision research has since concentrated on interpreting image motion as the projection of solid objects undergoing rigid-body motion, including the rigid motion of the stationary environment relative to an observer's own motion, called egomotion (see [18,135] for reviews of the literature).

Not everything in the world, however, is rigid. There is a continuum of motions, of which rigid motion is but one extreme. A list of some of the categories in this continuum (from most coherent to least coherent) is: rigid motion, jointed motion, biological motion, elastic motion, laminar flow, flow with vortices, and fully developed turbulence. A general-purpose vision system must be able to recognize any of these different types of motion, thus allowing the system to make abstract inferences (e.g., solid or fluid), predictions (e.g., future location), and comparisons or contrasts (e.g., viscous or free flowing).

This dissertation presents several models for motion perception. I use the word "model" in several ways in this dissertation: (1) I develop models of processes that operate in the physical world and describe how they project to images; (2) I develop techniques for recognizing such processes, and in some cases I propose that these techniques are models for biological vision; (3) I posit statistical models of the noise, error, and uncertainty in sensor obsevations and estimates.

Background and motivation for the motion models is presented in the remainder of this chapter. Chapter 2 discusses motion energy, a multiresolution representation of motion information extracted from image sequences. Chapter 3 presents a model for the extraction of image flow. Chapter 4 uses the image flow model to simulate psychophysical data on velocity discrimination and on the coherence of sine-grating plaid patterns. Chapter 5 discusses a model for combining uncertain and noisy sensor information about the observer's motion and about image flow in order to detect moving objects and to recover the 3-D spatial layout of the scene.

Chapter 6 proposes two additional sources of information to help solve the difficult problem of recovering the motion parameters of rigidly moving objects. Chapter 7 proposes a model for the recognition of turbulent flow. Chapter 8 summarizes the contributions of this dissertation and proposes directions for future research.

1.2 Perceptual Organization

The emphasis in computer vision research over the past decade has been the recovery of depth information lost to projection (for example, see Marr [96] or Barrow and Tenenbaum [20]). This emphasis can be traced back to Helmholtz who first listed the sources of information (depth cues) about the perceived distance of objects. The research has been primarily based on point-wise models of image formation borrowed from optics, material science, and physics, but the local recovery of depth using such point-wise models is inherently underdetermined. So researchers have constrained the problem by invoking (oftentimes unverifiable) assumptions like smoothness, continuity, or isotropy. In the real world, unfortunately, such assumptions are often in error. Thus, several researchers [91,110,149] have recently critisized the goal of recovering a dense depth map:

> On the whole, the performance and generality of depth recovery techniques has been unimpressive. Those techniques that rely on weak, general assumptions such as isotropy have proved fragile and error-prone; while such assumptions may be frequently valid, they also tend to be violated fairly often. Those that rely on artificial domain restrictions (e.g., smooth Lambertian faces, uniform albedo, point source illumination) clearly do not apply in complex natural scenes.[149]
>
> It may turn out that the problems with current recovery techniques are inherent in the local, quantitative nature of the approaches they use. It is easily demonstrated (e.g., by looking through a reduction tube) that, in general, very little information about surface or boundary characteristics can be gleaned from small image neighborhoods that are viewed out of context.[149]
>
> It is also clear that detailed, analytic models of the image formation process are not essential to human perception; humans function quite well with range finder images (where brightness is proportional to distance rather than a function of surface orientation), electron microscope images (which are approximately the reverse of normal images), and distorted and noisy images of all kinds — not to mention paintings and drawings.[110]
>
> There is, however, a more fundamental issue: Even if a depth map could be reliably obtained, how would it be used? ... a depth map is still fundamentally an

> image, with distance replacing brightness as the dependent variable. Being just an array of numbers, it is difficult to think of tasks that a depth map directly supports. For example, while raw depth values may suffice for elementary obstacle avoidance, grasping or recognizing objects requires that the depth first be organized into larger structures corresponding, e.g., to continuous visible surfaces.[149]

If the recovery of a dense depth map is not the primary basis for visual perception, then what is? Several researchers [22,23,69,87,91,110,149,152] have placed renewed emphasis on understanding perceptual organization:

> People's ability to perceive structure in images exists apart from both the perception of tri-dimensionality and from the recognition of familiar objects. That is, we organize the data even when we have no idea what it is we are organizing ... It is almost as if the visual system has some basis for guessing *what* is important without knowing *why*.[149]

Gestalt psychologists [83,148] were the first to stress the internal organizational processes in visual perception. They tried to enumerate the "laws" of perceptual organization[1], but were not inclined to ask how the visual system benefits from perceptual organization.

What is the function of perceptual organization? Pentland [110] and Bobick [23] have proposed that the function of perceptual organization is to recognize regularities that are abundant in our environment. For example, evolution repeats its solutions when choosing optimal characteristics for living organisms resulting in great regularities across species [128]. Similarly, man-made objects are subject to design constraints which result in regularities, e.g., a chair must have certain geometric properties for people to sit in it. Complex, inanimate, natural processes also exhibit regularity — for example, Mandelbrot [95] has found that clouds, mountain landscapes, turbulent water, lightning, cottage cheese, music, and the aggregation of galaxies all share uniformities that are characterized by a class of mathmatical functions called fractals. Also, Stevens [124] presents evidence that inanimate forms are constrained by physical laws to a limited number of basic patterns, and that natural textures occur in but a few basic forms. Some of the regularities in the world around us project to regularities in images. The function of perceptual organization is to "pick-up" on such regularities.

[1]The Gestalt psychologists assumed a homogeneous mechanism was responsible for perceptual grouping. But, I agree with Zucker [152] who argues that the diversity of grouping processes is the key to understanding early vision — each type of grouping process is mediated by a separate mechanism. Thus, in the proposed paradigm there are many models operating in parallel, each looking for different regularities in the image sequence.

In this context, perception's job is to recover lawful regularities that indicate causal organization in the sensory data. Pentland [110] states, "If we think of the world as an ongoing, moment-to-moment process, then perception's task is to discover the settings of the parameters that govern the process. Knowing what the parameters are and how they are set allows us to anticipate events, to predict the consequences of our actions, and to make abstract comparisons and contrasts."

Gibson [51,50] was the first to emphasize that there are regularities in images. He stressed that the space-time pattern falling on the retina contains all the information needed by an organism to interpret its environment and adjust its behavior — no additional constraints or assumptions are required. But Gibson was not interested in explaining the mechanisms by which these regularities are recognized.

1.3 Generic Process Models

How are regularities in images recognized? As Pentland [110] explains, "Understanding how to recover causal structure from regularities in the sensory data depends on having models of the physical world, and being able to recognize their instantiations. The need for a model cannot be sidestepped, for it is the model that relates the sensory data to the state of the real world. Thus, a theory of visual function that has no model of the world also has no meaning.[2]"

Understanding the early stages of perception as the interpretation of sensory data by use of models has, of course, been a standard vision research paradigm. To date, however, most models have been of two kinds: high-level, specific models, e.g., of people or houses, and low-level models of image formation, e.g., for local recovery of depth. The problems with low-level, local-recovery models are discussed above. The problem with high-level models is that they are too specialized, i.e., they are not flexible or general purpose.

Some researchers have begun to search for a third type of model, one with a grain size intermediate between the point-wise models of image formation and the object-specific models.

[2]Much vision research is not model based, of course: research on the mechanisms of vision (e.g., parallel processors, neurons), or on procedures for accomplishing visual tasks (e.g., regularization and relaxation methods) need not employ models of the world. But to understand visual *function* — that is, how one can infer information about the world — it is necessary to have a model of the salient world structure and of how that structure evidences itself in the image.

Since our environment is abundant in regularities, it may be possible to accurately describe our world as a relatively small set of *generic processes* that occur again and again, with the apparent complexity of our environment being produced from this limited vocabulary by compounding these basic forms in myriad combinations.

Next, I present a simple example of such a process model illustrating how it may be used to make reliable inferences about the world from sensory data. Then, I discuss the familiar model of rigid-body motion within the same paradigm.

1.3.1 Fly Detectors

Hoffman and Bennett [67] present an example of a simple perceptual mechanism similar to the following: we want to detect and localize in 3-space a certain species of flies. The fly detector has access only to the x- and y-coordinates of objects in its visual field, so it will have to infer the z-coordinate. It so happens that this species of flies exhibit a very specific behavior; they always move so that $z = m_1 x = m_2 y$ (a line through the origin). If we know we are looking at a fly it is simple to locate it in 3-space. For example, if we are told that a certain dot in the image is a fly, we need only record the dot's position and our model of the fly's behavior lets us infer z.

But, how do we know if we are looking at a fly? The answer is that there is no way to be certain we are looking at one, but we can be certain when we are *not* looking at one. For example, a dot in the image that moves from location $(2,3)$ to location $(2,4)$ is certainly not a fly because the line through those two points does not pass through the origin. By random chance, it is unlikely (probability zero) that we will observe a dot moving between two points that lie along a line through the origin.

We have a model of a class of objects in the world (flies that live along the lines that pass through the origin in 3-space), and we know how these objects appear in images (moving from point to point along lines that pass through the origin in the image). The model is *overconstrained*; we have three points (the origin and the two observed positions) to define a line. If these three points are not colinear then we know we are *not* looking at a fly. If they are colinear then we may infer that we are looking at one. Hoffman and Bennett argue that without overconstraint there will be no basis for making such perceptual inferences.

But, even with overconstraint the inference may be wrong. For example, what if there is a dot moving along the line $z = m_1 x = m_2 y$ that is not a fly? What if there is a dot moving along the curve $z^2 = m_1 x = m_2 y$? In order to make relible inferences, we need one additional bit of information — that there is an abundance of flies in the world, i.e., that real flies are much more prevalent than dots that merely appear like flies.

Dots that are not flies may move along any random path in three-space with equal likelihood. Thus, it is extremely unlikely (in fact, probablity zero) that such dots will move along paths that project to straight lines, $m_1 x = m_2 y$, through the origin in the image. On the other hand, flies always move along such paths. If flies are abundant, then the probability of seeing a fly is high and the probability of seeing a dot that merely appears like a fly is zero. Thus, when we observe motion consistent with a fly-interpretation, we may reliably infer that it is a fly.

In summary, Hoffman and Bennett [67] prove the fly detector model allows us to make reliable inferences because is has the following properties: (1) the projection of a fly's movement exhibits a lawful regularity that is overconstrained in the image plane — we can measure two positions of a dot in the image and check to see if the line between them also passes through the origin; (2) the fly detector model is overconstrained in the physical world, i.e., given a line in the image plane there is a unique solution for z[3]; (3) this species of flies is abundant in the world — real flies are much more prevalent than dots that merely appear like flies.

In the real world, there is always noise and uncertainty. Our fly detector will miss some real flies that are observed at positions $(x+\delta_x, y+\delta_y)$. In fact, it is unlikely that the fly detector will ever detect an ideal fly. One solution is to follow a dot's motion, recording its position, over a period of time. The best-fit line for a real fly will approach $m_1 x = m_2 y$ and the residuals of the fit will be small. As we take more observations, it will be less likely that a dot is following a fly's path by chance, so the residuals of the fit for a non-fly will be large.

In order to make reliable inferences and estimates, more overconstraint is better. First, overconstraint allows a detector to test whether or not observations in the presence of noise and uncertainty are consistent with a particular model (in the above example, are the residuals large or small?). Second, overconstraint provides reliability by using more data for estimating the

[3]A planar subspace of the 3-D world projects to the line $m_1 x = m_2 y$. Within that planar subspace, there are many possible 3-D motions (e.g., $z = m_1 x = m_2 y$ or $z^2 = m_1 x = m_2 y$), but our model specifies that only one of them is correct.

parameters of the model.

1.3.2 Detecting Rigid Motion

Ullman [136] was the first to develop a computational model for the perception of rigid-body motion. A collection of isolated features in an image, which are the projections of points in space, are tracked over a discrete series of views. Ullman shows that given enough views of enough points the solution for rigid motion is *overdetermined.*

A rigidly moving object is a 3-D process that results in sensory data with a distinctive structure: points on the surface of the object move in a way that is unlikely to occur by chance. Thus, if the motion of the projections of the points are consistent with rigidity, we can infer that we are observing a rigid 3-D motion. A solid object moving rigidly in 3-space projects into an image sequence with a specific type of regularity. We use the rigid-motion model to detect that regularity, and then make two inferences: (1) the regularity is due to rigid motion in 3-space; (2) the rigid motion is a result of a single, solid object moving through space. Having recognized an instance of our rigid-motion model, we may then proceed to estimate the parameters of the model, i.e., its 3-D structure and 3-D motion.

For rigid motion in an idealized noiseless world, we know that our inferences will generally be reliable, because we can normally preclude both the ways in which our inferences can go wrong. The first type of potential error is that we think we have a rigid motion when in fact we do not. We can preclude this type of error because the equations are overconstrained; i.e., we can estimate the motion parameters using part of our data, and then check our answers using the remaining data. The second type of error is that we think we do not have an instance of rigid motion when in fact we do. We will never make this error since we will always infer rigidity when the the data is self-consistent.

I will apply this rigid-motion paradigm to several models of visual perception: the goal is to be able to recognize image regularities that allow us to infer that we have an instance of a given model, and then to recover the parameters of that model. These models are *generic process models*; each is a model for a class of processes in the world (e.g., rigid motion, turbulent flow, surface roughness). The models are overconstrained to make it possible to make reliable inferences and estimates in the presence of noise and uncertainty.

An crucial aspect of such a model is its *robustness*, its sensitivity to assumptions (for example, assuming that the measurement errors are normally distributed) that are open to question. In general, robustness must be tested empirically with test cases that violate the assumptions in a variety of ways.

The most important characteristic of generic process models is that they recognize specific regularities in the image that correspond to regularities in the three-dimensional world. The models do **not** *make unverifiable assumptions about the world (e.g., the assumption of rigid motion), but rather they test the validity of such hypotheses.*

1.4 Active Vision and Sensor Fusion

Most of the past and present research in machine perception involves analysis of passively sampled data. Some researchers [8,11,49,50,51,84] have argued that perception is not passive, but *active*. By active sensing, these authors do not mean to say that the sensor transmits energy (e.g., radar or sonar). Rather, active sensing refers to employing a passive sensor in an active fashion:

> Perceptual activity is exploratory, probing, searching; percepts do not simply fall onto sensors as rain falls onto the ground. We do not just see; we look. And in the course of looking, our pupils adjust to the level of illumination, our eyes bring the world into sharp optical focus, our eyes converge or diverge, we move our heads or change our position to get a better view of something, and sometimes we even put on spectacles.[49]

What are the advantages of active sensing over passive sensing? First, Aloimonos et al [8] demonstrate that an active observer has a theoretic and algorithmic advantage over a passive one for solving a number of vision problems. Some problems that are ill-posed for a passive observer are well-posed for an active one, and some problems that are unstable for a passive observer are stable for an active one.

Second, active vision gives an observer the opportunity to take more data. In the context of the generic process models discussed above, taking more data leads to more constraint. A general rule of thumb of the active vision/sensor fusion paradigm is, "when in doubt, take more data".

Finally, information from perceptual sources (e.g., estimating image motion and camera motion) is inherently noisy and uncertain. A sensing system can make substantial gains by explicitly representing the uncertainty in sensor data and taking actions to reduce it (e.g., by moving to a different viewing position):

> In the real world and using real sensors we must contend with the three basic sources of uncertainty in sensor data: (1) statistical uncertainty due to random noise processes in the sensing device; (2) non-statistical uncertainties modeling quantization or mechanical backlash; (3) incompleteness or underdeterminedness due to limited sensor scope. Interpreting sensory data in these circumstances requires methods for determining the consistency of data, and methods for combining observations across sensors, space and time into a single statement about the world – the *sensor fusion* problem [58].

Sensor fusion becomes particularly important for active vision techniques that, for example, require the integration of information extracted from image data with information about camera position.

In order to combine noisy information from different sensors, each sensor must provide us both with obserations and with some measure of the uncertainty in its observations. A sensor model [36] is a description of a sensor's ability to observe the environment. This is in general a function of the state of the environment, the state of the sensor itself, and the state of other sensors or cues in a multi-sensor system. A static sensor model or *observation* model describes the dependence of an observation on the state of the environment. In the chapters that follow, I make use of probabilistic observation models to characterize the uncertainty in motion observations.

1.5 Summary

This chapter argues that the basic function of preattentive/peripheral/immediate visual perception is perceptual organization, the detection of regularities in images that correspond to regularities in the environment. I propose using generic process models, exemplified by the fly-detector model and the rigid-body motion detector model, to detect such regularities. A process model allows us to make reliable inferences if and only if it satisfies three conditions: (1) the projection of the process exhibits a lawful regularity that is overconstrained in the image plane; (2) the model is

overconstrained in the physical world, i.e., once the image regularity has been detected there is a unique solution for the model's parameters; (3) the process is abundant in the world. The most important characteristic of process models is that they do not make unverifiable assumptions about the world, but rather they test the validity of such hypothesis. Overconstraint allows one to test whether or not observations (image data) are consistent with a particular model of the world.

The ultimate goal of this research is to determine which model is most appropriate for a given region. In the proposed paradigm there are many process models operating in parallel. For example, this dissertation discusses models for the recognition of turbulent flows, for the detection of moving objects, and for the recognition of rigid-body motion. How do we decide which model is "more" appropriate for a region of an image? Actually, two models may both be appropriate — for example, waves at the beach are turbulent at a small scale, but at a large scale their motion is approximately rigid translation toward the shoreline. Overconstraint allows one to test whether or not observations (image data) are consistent with a particular model of the world.

Combining data from different sensors and using active vision (e.g., using head and eye movements) provides extra constraints on a number of vision problems. For example, this dissertation poses the detection of moving objects and the recovery of depth from motion as sensor fusion problems that necessitate combining information from different sensors in the presence of noise and uncertainty.

Motion analysis plays a key role for an active observer who is moving his head and eyes in order to better perceive his environment. Conversely, active vision and sensor fusion are key ingredients for motion analysis, particularly since sensor information is subject to noise and uncertainty. Sensor error can be characterized by sensor models, statistical models of a sensor's ability to observe the environment. For example, this dissertation discusses a sensor model for the extraction of image flow.

Chapter 2

Motion Energy

As Fleet and Jepson [42] suggest, I view the first functional level of visual processing as "consisting of several concurrent, image-independent processes applied blindly throughout the image to extract any available information that appears salient and functionally useful, extracting as much information as possible while requiring no previous or concurrent interpretation." By contrast, token-matching techniques require a significant amount of scene interpretation — tokens must be identified while noise and other irrelevant features are removed. This chapter discusses motion energy, a multiresolution representation of motion information extracted from image sequences.

In the next section, I review the Gaussian pyramid, a multiscale decomposition of images. Section 2.1 reviews the mathematics of image motion in the spatiotemporal-frequency domain. A family of motion-sensitive Gabor-energy filters are then described in Sections 2.3 and 2.4.

2.1 The Gaussian Pyramid

Low-level image processing often involves computing some property of an image within local windows. It is usually not known a priori what window size to use, so it is necessary to do the computations for a variety of window sizes. The gaussian pyramid [28] is an efficient representation for computing properties of images for a number of different window sizes.

The pyramid is built by repeatedly convolving the image with a small weighting function. Samples that contribute to each weighted sum are not contiguous pixels, but rather are separated

by a distance that doubles with each iteration. This generates a sequence of low-pass filtered images in which the bandlimit of each image is one octave lower than that of its predecessor.

The pyramid is implemented as a series of separable 5 X 5 convolutions. A 5 X 5 Gaussian is approximated by separable convolution with a kernel whose elements are in the ratio 1:4:6:4:1. The output from each convolution is subsampled (reduced) by a factor of two. This yields an image that is one level above (half the size) its predecessor in the pyramid.

Similarly, a Laplacian pyramid is a sequence of bandpass filters computed by expanding each level of the Gaussian pyramid to twice its size and subtracting from the level below. The expansion is computed similarly to the reduction, as a convolution with a low-pass filter.

Performing the same operation at each level of a pyramid is equivalent to performing operations within different-sized Gaussian windows of the original image. For example, the models presented in the following chapters use families of motion-energy filters tuned to different spatiotemporal-frequency bands. This is accomplished by computing a Gaussian pyramid for each image in the sequence and using the same family of filters at each level of the pyramid.

2.2 Motion in the Frequency Domain

Several authors [37,41,45,46,141,140] have pointed out that some properties of image motion are most evident in the Fourier domain. This section describes one-dimensional motion in terms of spatial and temporal frequencies and observes that the power spectrum of a moving one-dimensional signal occupies a line in the spatiotemporal-frequency domain. Analogously, the power spectrum of a translating two-dimensional texture occupies a tilted plane in the frequency domain.

One-Dimensional Motion. The spatial frequency of a moving sine wave is expressed in cycles per unit of distance (e.g., cycles per pixel), and its temporal frequency is expressed in cycles per unit of time (e.g., cycles per frame). Velocity which is distance over time or pixels per frame, equals the temporal frequency divided by the spatial frequency:

$$v = \omega_t / \omega_x \tag{1}$$

When a signal is sampled evenly in time frequency components greater than the Nyquist

frequency (1/2 cycles per frame) become undersampled, or aliased. As a consequence, if a sine wave pattern is shifted more than half its period from frame to frame it will appear to move in the opposite direction. For example, a sine wave with a spatial frequency of 1/2 cycles per pixel can have a maximum velocity of one pixel per frame and a sine wave with spatial frequency 1/4 cycles per pixel can have a maximum velocity of two pixels per frame. In other words, the range of possible velocities of a moving sine wave is limited by its spatial frequency.

Now consider a one-dimensional signal moving with a given velocity v that has many spatial-frequency components. Each such component ω_x has a temporal frequency of $\omega_{t_1} = \omega_x v$, while each spatial-frequency component $2\omega_x$ has twice the temporal frequency $\omega_{t_2} = 2\omega_x v$. In fact, the temporal frequency of this moving signal as a function of its spatial frequency is a straight line passing through the origin where the slope of the line is v.

Two-Dimensional Motion. Analogously, two-dimensional patterns (textures) translating in the image plane occupy a plane in the spatiotemporal-frequency domain:

$$\omega_t = u\omega_x + v\omega_y \tag{2}$$

where $\vec{\theta} = (u, v)$ is the velocity of the pattern [141]. For example, the expected value of the sample power spectrum of a translating random-dot field is a constant within this plane and zero outside of it.

If the motion of a small region of an image may be approximated by translation in the image plane, the velocity of the region may be computed in the Fourier domain by finding the plane in which all the power resides. To extract optical flow we could take small spatiotemporal windows out of the image sequence and fit a plane to each of their power spectra. In Chapter 3 I present a technique for estimating velocity by using motion-sensitive spatiotemporal Gabor-energy filters to efficiently sample these power spectra.

The Aperture Problem in the Frequency Domain. An oriented pattern, such as a two-dimensional sine grating or an extended step edge, suffers from what has been called the aperture problem (for example, see Hildreth [64]). For such a pattern there is not enough information in the image sequence to disambiguate the true direction of motion. At best, we may extract only one of the two velocity components as there is one extra degree of freedom. In the

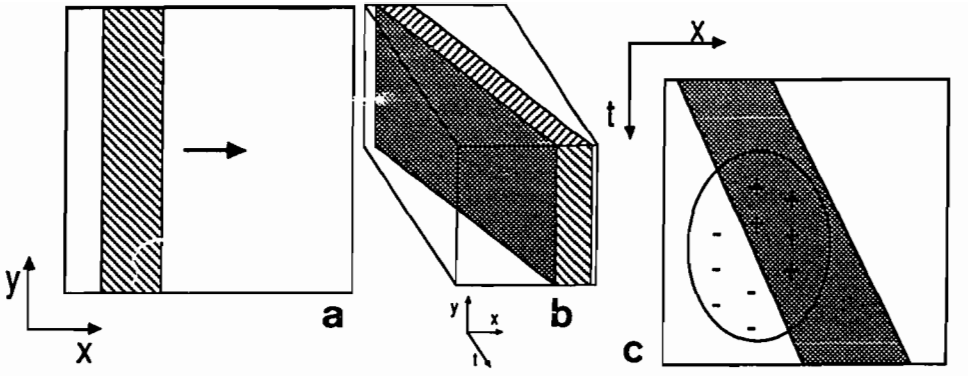


Figure 1: Spatiotemporal Orientation (redrawn from Adelson and Bergen [2]). (a) A vertical bar translating to the right. (b) The space-time cube for a vertical bar moving to the right. (c) An $x - t$ slice through the space-time cube. The orientation of the edges in the $x - t$ slice is the horizontal component of the velocity. Motion is like orientation in space-time and spatiotemporally oriented filters can be used to detect it.

spatiotemporal-frequency domain the power spectrum of such an image sequence is restricted to a line and the many planes that contain the line correspond to the possible velocities. Normal flow, defined as the component of motion in the direction of the image gradient, is the slope of that line.

2.3 Motion-Sensitive Filters

Adelson and Bergen [2] have pointed out that image motion is characterized by orientation in space-time. For example, Figure 1(a) depicts a vertical bar moving to the right over time. Imagine that we film a movie of this stimulus and stack the consecutive frames one after the next. We end up with a three-dimensional volume (space-time cube) of luminance data like that shown in Figure 1(b). Figure 1(c) shows an $x - t$ slice through this space-time cube; the slope of the edges in the $x - t$ slice equals the horizontal component of the bar's velocity (change in position over time). The figure also depicts a linear filter that is tuned for the motion of this moving bar. Thus, motion is like orientation in space-time and spatiotemporally oriented filters can be used to detect it. Three-dimensional Gabor-energy filters, presented below, are such oriented spatiotemporal filters.

A one-dimensional sine- (or odd-) phase Gabor filter is simply a sine wave multiplied by a

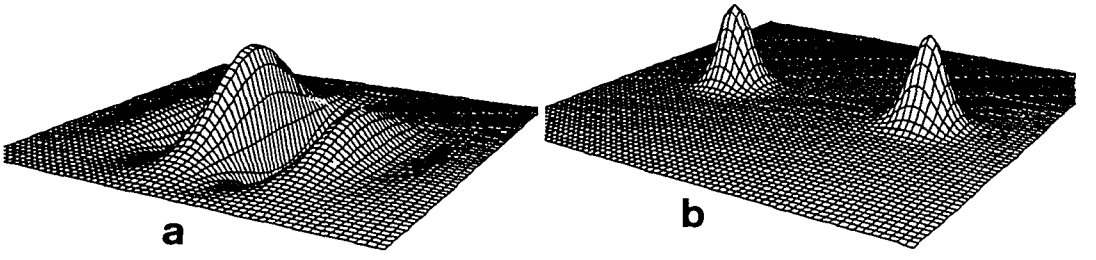


Figure 2: Perspective views of (a) a two-dimensional sine-phase Gabor function and (b) its power spectrum.

Gaussian window:

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{\frac{-t^2}{2\sigma^2}\right\} \sin(2\pi\omega t) \tag{3}$$

These filters were originally introduced by Gabor [44]. The power spectrum of a sine wave is a pair of impulses located at ω and $-\omega$ in the frequency domain. The power spectrum of a Gaussian is itself a Gaussian (i.e., it is a lowpass filter). Since multiplication in the space (or time) domain is equivalent to convolution in the frequency domain, the power spectrum of a Gabor filter is the sum of a pair of Gaussians centered at ω and $-\omega$ in the frequency domain, i.e., it is a bandpass filter. Thus, a Gabor function is localized in a Gaussian window in the space (or time) domain and it is localized in a pair of Gaussian windows in the frequency domain.

Daugman [32,33] has extended Gabor filters to a family of two-dimensional functions, an example of which is shown along with its power spectrum in Figure 2.

An example of a 3-D (space-time) Gabor filter is

$$g(x,y,t) = \frac{1}{\sqrt{2}\pi^{3/2}\sigma_x\sigma_y\sigma_t} \exp\left\{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{t^2}{2\sigma_t^2}\right)\right\} \sin(2\pi\omega_{x_0}x + 2\pi\omega_{y_0}y + 2\pi\omega_{t_0}t) \tag{4}$$

where $(\omega_{x_0}, \omega_{y_0}, \omega_{t_0})$ is the center frequency (the spatial and temporal frequency for which this filter gives its greatest output) and $(\sigma_x, \sigma_y, \sigma_t)$ is the spread of the spatiotemporal Gaussian window. Three-dimensional Gabor functions look something like a stack of plates with small plates on the top and bottom of the stack and the largest plates in the middle of the stack. The stack can be tilted in any orientation in space-time.

It is a simple matter to tune the filter to different frequencies and orientations while trading bandwidth for localization. To change the frequency tuning we independently vary ω_{x_0}, ω_{y_0},

and ω_{t_0}. Narrowing the Gaussian window in the space-time domain broadens the bandpass window in the spatiotemporal-frequency domain and vice versa.

Gabor filters have the additional property that they can be built from separable components, thereby greatly increasing the efficiency of the computations. A new technique for computing Gabor filter outputs from separable convolutions is presented in appendix A. Let k be the size of the convolution kernel, let m be the number of images in a sequence, and let each image be n pixels in size. By simplifying the complexity[1] of three-dimensional convolution from $O(k^3n^2m)$ to $O(kn^2m)$, separability speeds it up by two orders of magnitude, given a kernel size of 10 pixels.

The model presented in the following sections employs quadrature pairs of filters, odd-phase and even-phase filters of identical orientation and bandwidth. The sum of the squared output of a sine-phase filter, Equation (4), plus the squared output of a cosine-phase filter gives a measure of Gabor energy that is invariant to the phase of the signal. The frequency response of such a Gabor-energy filter is the sum of a pair of 3-D Gaussians (a one-dimensional version of this equation is derived in Appendix B):

$$\begin{aligned} G(\omega_x,\omega_y,\omega_t) &= (1/4)\exp\{-4\pi^2[\sigma_x^2(\omega_x-\omega_{x_0})^2+\sigma_y^2(\omega_y-\omega_{y_0})^2+\sigma_t^2(\omega_t-\omega_{t_0})^2]\} \\ &+ (1/4)\exp\{-4\pi^2[\sigma_x^2(\omega_x+\omega_{x_0})^2+\sigma_y^2(\omega_y+\omega_{y_0})^2+\sigma_t^2(\omega_t+\omega_{t_0})^2]\} \end{aligned} \quad (5)$$

Equation (5) means that a motion-energy filter with center frequency $(\omega_{x_0},\omega_{y_0},\omega_{t_0})$ will give an output of $G(\omega_x,\omega_y,\omega_t)$ for a moving sine grating with spatial and temporal frequencies $(\omega_x,\omega_y,\omega_t)$. The filter will give a large output for a stimulus that has a lot of power near the filter's center frequency and it will give a smaller output for a stimulus that has little power near the filter's center frequency.

In principle, the models presented in the following chapters could utilize oriented spatiotemporal bandpass filters other than Gabor filters. For example, Mallat [93] and Adelson and Simoncelli [4] have proposed orthogonal and complete multiscale representations for two-dimensional images that could be extended to space-time. Also, it may be important for some applications to eliminate delay and use filters with a causal temporal response (Gabor filters are not causal) like those suggested by Adelson and Bergen [2] or Watson and Ahumada [141].

[1]Complexity is defined as the order of magnitude, $O()$, of the number of operations required for a computation.

2.4 A Family of Motion-Energy Filters

The models presented in the following chapters use a family of Gabor-energy filters, all of which are tuned to the same spatial frequency band but to different spatial orientations and temporal frequencies, i.e., $\omega_0 = \sqrt{\omega_{x0}^2 + \omega_{y0}^2}$ is constant for all of the filters in one such family.

Eight of the twelve energy filters used in the present implementation have their peak response for patterns moving in a given direction — for example, one of them is most sensitive to rightward motion of vertically oriented patterns, while another is most sensitive to leftward motion. The other four filters have their peak response for stationary patterns, each with a different spatial orientation. The power spectra of the 12 filters are pairs of 3-D Gaussians (each pair of Gaussians corresponds to one filter) that are positioned on the surface of a cylinder in the spatiotemporal-frequency domain (Figure 3): eight of them around the top of the cylinder, eight of them around the middle, and eight around the bottom.

We can build several such families of filters tuned to different spatiotemporal-frequency bands. For the current implementation I have opted to compute a Gaussian pyramid [28] for each image in the sequence and I convolve with a single family of filters at each level of the pyramid. This is essentially the same as using families of filters with equal bandwidths that are spaced one octave apart in spatial frequency, but are tuned to the same temporal frequencies. Filters higher up in the pyramid achieve their peak response for patterns with lower spatial frequency, but with the same temporal frequency. Thus, the lower-frequency filters have their greatest outputs for patterns moving at greater velocities.

Psychophysical evidence [27,37,80] suggests that human motion channels exhibit such a relationship between spatial frequency and velocity. This makes sense from a computational viewpoint since patterns containing only high spatial frequencies may move at only low velocities, whereas patterns containing only lower spatial frequencies may move at greater velocities (see the discussion in Section 2.2 on sampling and temporal aliasing).

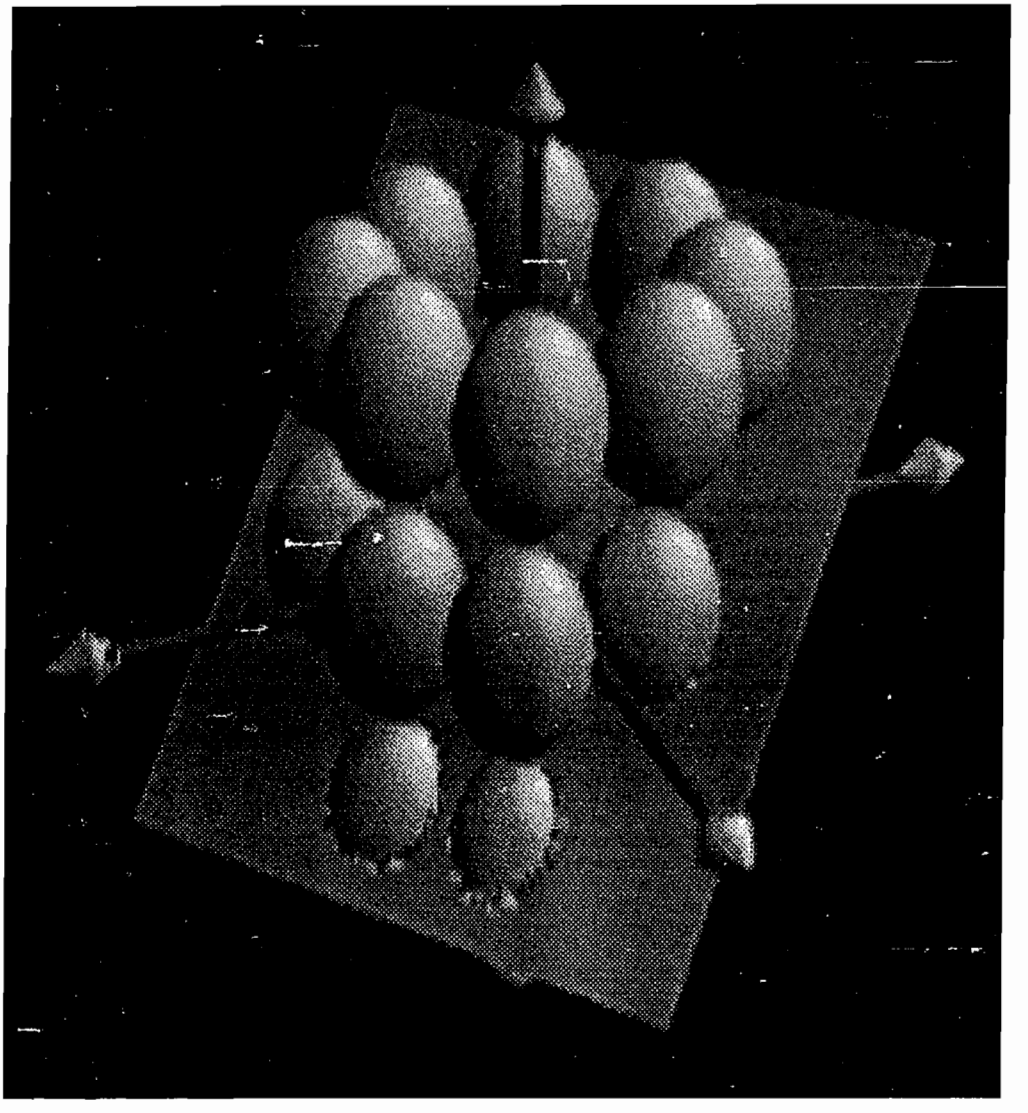


Figure 3: The power spectra of the 12 motion-sensitive Gabor-energy filters are positioned in pairs on a cylinder in the spatiotemporal-frequency domain. Each symmetrically-positioned pair of ellipsoids represents the power spectrum of one filter. The plane represents the power spectrum of a translating texture. A filter will give a large output only for a stimulus that has a lot of power near the centers of its corresponding ellipsoids and it will give a relatively small output only for a stimulus that has no power near the centers of its ellipsoids. Each velocity corresponds to a different tilt of the plane, and thus to a different distribution of outputs for the collection of motion-energy mechanisms.

Chapter 3

Image Flow

Optical flow, a two-dimensional velocity vector for each small region of the visual field, is one representation of image motion. The perception of visual motion does not depend on prior interpretation or recognition of shape and form. However, it does depend on there being motion information, i.e., changes in intensity over time throughout the visual field. Without texture, a perfectly smooth moving surface yields an image sequence in which most local regions do not change over time. But in a highly textured world (e.g., natural outdoor scenes with trees and grass), there is motion information throughout the visual field. This chapter addresses the issue of extracting a velocity vector for each region of the visual field by taking advantage of the abundance of motion information in a highly textured image sequence.

Most machine vision efforts that try to extract image flow employ just two frames from an image sequence; either matching features from one frame to the next [17] or computing the change in intensity between successive frames along the image gradient direction [70,79]. In a highly textured world neither of these approaches seems appropriate, since there may be too many features for matching to be successful and the image gradient direction may vary randomly from point to point. In fact, an error analysis of gradient-based methods [79] confirms that a major problem with the approach is that large errors are made where the image is highly textured, precisely where there is the greatest amount of motion information!.

There have recently been several approaches to motion measurement based on spatiotemporal filters [2,37,40,46,45,137,141,140] that utilize a large number of frames sampled closely together

in time. These papers describe families of motion-sensitive mechanisms each of which is selective for motion in different directions. To be able to use such mechanisms in computing optical flow, one must overcome two obstacles: (1) the aperture problem; (2) the fact that the filter outputs do not depend solely on the velocity of a stimulus, but rather on its spatial and temporal frequencies.

In the previous chapter, I reviewed the mathematics of motion in the spatiotemporal-frequency domain and described how 3-D Gabor filters act as motion-sensitive mechanisms. In Section 3.1 I formulate a model for extracting image velocity from the outputs of these filters. Section 3.1.3 reformulates the model as a parallel mechanism that computes a distributed representation of image velocity. In Section 3.2 I formulate a measure of uncertainty in the velocity estimates. Section 3.3 discusses how the model deals with the aperture problem, comparing its performance to that of the human visual system. The next chapter uses the model to simulate psychophysical data.

3.1 Motion Energy to Extract Image Flow

Spatiotemporal bandpass filters like Gabor-energy filters and those filters discussed in previous papers [2,40,141] are *not* velocity-selective mechanisms, but rather are tuned to particular spatiotemporal frequencies. A single such mechanism cannot distinguish between variations in the spatial-frequency content of the stimulus, variations in its temporal-frequency content, or variations in its contrast. But, an unambiguous velocity estimate may be computed from the ouputs of a collection of such mechanisms.

In what follows I describe a new way of combining the outputs of a collection of motion-energy mechanisms in order to extract velocity. The role of the filters is to sample the power spectrum of the moving texture. The problem is to estimate the slope of the plane in the frequency domain that corresponds to the actual velocity. First, I derive equations for Gabor energy resulting from motion of random textures or random-dot fields. Based on these equations I formulate a least-squares estimate of velocity.

Consider an analogous two-dimensional problem — estimating the slope of a line that passes through the origin by viewing it with a finite number of circular windows. Figure 4 shows a

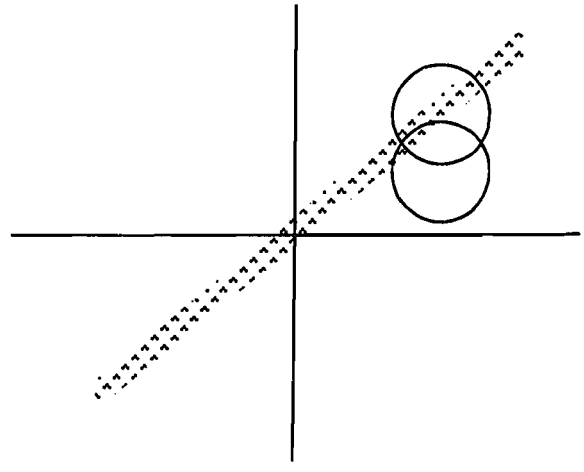


Figure 4: A problem analogous to that of extracting velocity — estimating the slope of a line that passes through the origin by viewing it with a finite number of circular windows. The upper window encloses many points while the lower one encloses significantly fewer. In other words, the line must pass close to the center of the upper window while staying far from the center of the lower one.

dotted line and two circular windows. We are given a family of such windows, a finite number of them centered at known positions. The only information we have is the number of points from the dotted line that lie within each window (in particular, we do not know the spacing between the dots). The upper window in the figure encloses many points while the lower one encloses significantly fewer. Therefore, the line must pass close to the center of the upper window while staying far from the center of the lower one. Notice that it is impossible to estimate the slope given only one circular window since the number of dots within a particular window depends both on the slope of the line and on the dot density.

3.1.1 Extracting Pattern Flow

In order to extract image velocity from the outputs of motion-energy filters we replace, in Figure 4, both the dotted line with a plane and the circular windows with 3-D Gaussian windows. A circular window simply counts the number of points it encloses. A Gaussian window counts the points and weights each according to its distance from the center of the window. This is formalized by Parseval's theorem that states that the integral of the squared values over the space-time domain is proportional to the integral of the squared Fourier components over the

frequency domain:

$$\begin{aligned}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|f(x,y,t)|^2dxdydt &= \frac{1}{8\pi^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|F(\omega_x,\omega_y,\omega_t)|^2d\omega_xd\omega_yd\omega_t \quad (6)\\ &= \frac{1}{8\pi^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}P(\omega_x,\omega_y,\omega_t)d\omega_xd\omega_yd\omega_t\end{aligned}$$

where $F(\omega_x,\omega_y,\omega_t)$ is the Fourier transform of $f(x,y,t)$ and $P(\omega_x,\omega_y,\omega_t)$ is the power spectrum. Convolution with a bandpass filter results in a signal that is restricted to a limited range of frequencies. Therefore, the integral of the square of the convolved signal is proportional to the integral of the power of the original signal over this range of frequencies.

Parseval's thereom may be used to derive an equation that predicts the output of a Gabor-energy filter in response to a moving random texture. The expected value of the sample power spectrum of a translating random-dot field is zero, except within a plane (Equation 2) where it is a constant k. The frequency response of a Gabor-energy filter is the sum of a pair of 3-D Gaussians. By Parseval's theorem, Gabor energy in response to a moving-random texture is twice the integral of the product of a 3-D Gaussian and a plane — by substituting Equation (2) for ω_t in Equation (5), multiplying by two, and integrating over the frequency domain we get:

$$\begin{aligned}\mathcal{R}(u,v,k;\omega_{x_0},\omega_{y_0},\omega_{t_0}) &= (k^2/2)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\exp\{-4\pi^2[\sigma_x^2(\omega_x-\omega_{x_0})^2 \quad (7)\\ &+ \sigma_y^2(\omega_y-\omega_{y_0})^2+\sigma_t^2(u\omega_x+v\omega_y-\omega_{t_0})^2\}d\omega_xd\omega_y\end{aligned}$$

where $(\omega_{x_0},\omega_{y_0},\omega_{t_0})$ is the center frequency of the motion energy filter, $(\sigma_x,\sigma_y,\sigma_t)$ is the spread of the filter's spatiotemporal Gaussian window, (u,v) is the velocity of the stimulus, and k is proportional to image contrast. This integral evaluates to

$$\begin{aligned}\mathcal{R}(u,v,k;\omega_{x_0},\omega_{y_0},\omega_{t_0}) &= H_4(u,v,k)\exp[-4\pi^2\sigma_x^2\sigma_y^2\sigma_t^2H_1(u,v;\omega_{x_0},\omega_{y_0},\omega_{t_0})] \quad (8)\\ H_1(u,v;\omega_{x_0},\omega_{y_0},\omega_{t_0}) &= \frac{H_2(u,v)}{H_3(u,v)}\\ H_2(u,v;\omega_{x_0},\omega_{y_0},\omega_{t_0}) &= (u\omega_{x_0}+v\omega_{y_0}+\omega_{t_0})^2\\ H_3(u,v) &= (v\sigma_x\sigma_t)^2+(u\sigma_y\sigma_t)^2+(\sigma_x\sigma_y)^2\\ H_4(u,v,k) &= \frac{k^2}{8\pi\sqrt{H_3(u,v)}}\end{aligned}$$

Equation (8) means that a motion-energy filter with center frequency $(\omega_{x_0},\omega_{y_0},\omega_{t_0})$, will give an output of $\mathcal{R}(u,v,k)$ for a random-dot texture moving with speed (u,v). If we multiply the

grey levels at each pixel of the image sequence by a constant c, then the filter's output will increase by a factor of c^2.

For a family of Gabor-energy filters, we get a system of equations (one for each filter) in the three unknowns (u, v, k). The factor $H_4(u, v, k)$ which appears in each of these equations does not depend on the center frequency of the filters — it can be eliminated by dividing each equation by the sum or average of them all. This results in a system of equations depending only on u and v that predict the outputs of the family of Gabor-energy filters. These predicted energies are exact for a pattern with a flat power spectrum.

But, what if the power spectrum of the pattern is not flat? In particular, what if the image contrast is different for different spatial orientations? Rather than dividing each filter output by the sum of *all* of the filter outputs, we can group the filters according to their spatial orientation and normalize each spatial orientation separately. Filters that differ only in their temporal-frequency tunings line up in vertical columns in the spatiotemporal-frequency domain (see figure 3). One such column is sensitive only to a small range of spatial frequencies and orientations. By using filters with narrow spatial bandwidths I believe that many natural textures will have a power spectrum that is flat within this range. The results presented below on images of real textures indicate that this is the case.

In order to specify a procedure for estimating velocity, I must now introduce some additional notation. Let m_i $(i = 1 - 12)$ be the twelve measured motion energies where each i corresponds to the output of a filter with a different center frequency. For each m_i, let $\mathcal{R}_i(u, v)$ be the corresponding predicted motion energy,

$$\mathcal{R}_i(u, v) = \exp\left\{-4\pi^2\sigma_x^2\sigma_y^2\sigma_t^2 H_1(u, v; \omega_{x_i}, \omega_{y_i}, \omega_{t_i})\right\} \tag{9}$$

where $H_1(u, v; \omega_{x_i}, \omega_{y_i}, \omega_{t_i})$ is defined in Equation (8). In addition, let $\overline{m_i}$ be the sum of the outputs of those filters that have the same preferred spatial orientation as the ith filter, and let $\overline{\mathcal{R}_i}(u, v)$ be the corresponding sum of the predicted motion energies,

$$\begin{aligned} \overline{m_i} &= \sum_{j \in M_i} m_j \\ \overline{\mathcal{R}_i}(u, v) &= \sum_{j \in M_i} \mathcal{R}_j(u, v) \end{aligned} \tag{10}$$

where M_i is the set of motion-energy filters that share the same spatial orientation as the ith filter.

A least-squares estimate for (u, v) minimizes the difference between the predicted and measured motion energies, i.e., it minimizes

$$l(u,v) = \sum_{i=1}^{12} \left[m_i - \overline{m}_i \frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)} \right]^2 \tag{11}$$

There are standard numerical methods for estimating $\vec{\theta} = (u, v)$ to minimize Equation (11), e.g., the Gauss-Newton gradient-descent method [53].

Alternatively, the least-squares estimate of $\vec{\theta} = (u, v)$ maximizes

$$f(u,v) = \exp\left\{ -\sum_{i=1}^{12} \left[m_i - \overline{m}_i \frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)} \right]^2 \right\} \tag{12}$$

Equation (12) is a response surface; the location of the peak in this surface corresponds to the velocity extracted by the model. Section 3.1.3 describes how Equation (12) can be used to compute a distributed representation of image velocity.

3.1.2 The Algorithm

The main steps in the computations performed by the model are: (1) to convolve the image sequence with 3-D Gabor filters; (2) to compute motion energy as the squared sum of the sine- and cosine-phase Gabor filter outputs; (3) to estimate velocity by either minimizing Equation (11) or maximizing Equation (12). In this section I explain the additional steps that need to be computed and I summarize the entire algorithm.

Firstly Parseval's theorem, Equation (6), relates an integral over the space-time domain to an integral over the frequency domain — since the filters are localized in both domains convolving with a Gaussian is one way to approximate this integral. We can think of the model as computing the average image velocity within this Gaussian window.

Of course, Gaussian convolution will tend to smooth over motion boundaries and other regions where the velocity changes rapidly from point to point. Some possible solutions to this problem are: (1) to use images of higher resolution; (2) to use a different method for combining information other than Gaussian convolution, e.g., relaxation labeling methods (for references,

see Hummel and Zucker [73]) or finite-element regularization methods (for references, see Terzopoulos [127] or Poggio et al [112]).

There are two situations for which this smoothing problem is particularly bad. First, in regions moving with high speed we must use filters that are higher in the pyramid, i.e., of lower spatial resolution. Second, where there is a region of low image contrast adjacent to one of high contrast the filter outputs for the high contrast region (since they are greater on average) will bias the velocity estimates for the low contrast region. The former situation may be controlled by incorporating eye/camera movements — an initial low-resolution estimate may be used to drive tracking eye movements thereby decreasing the image velocity and allowing for estimates of higher spatial resolution. The latter situation may be solved by "adaptation" (automatic gain control) — for example, we may "equalize" image contrast by computing the zero-crossings [97] of each image and then applying the model to the resulting zero-crossing image sequence.

Finally, a problem with Gabor filters is that all but the sine-phase filters have some dc response. If an image is very bright (large mean luminance) and of low contrast the output of the filter may be dominated by response to the dc rather than to the image contrast signal. Clearly this is undesirable. This difficulty can be alleviated by first subtracting the local mean luminance, e.g., by convolving with a center-surround filter that has a very sharp positive center and a broad negative surround. The dc-problem may also be alleviated by using only sine-phase filters — if the stimulus has uncorrelated random phase, then a phase-independent motion energy can be computed from sine-phase filters alone by averaging their squared outputs within appropriately-sized windows.

In summary, an algorithm for extracting image flow proceeds as follows:

1. Compute a Gaussian pyramid for each image in the image sequence.

2. Convolve each of the resulting images with a center-surround filter to remove the dc and lowest spatial frequencies.

3. Compute the sine- and cosine-phase Gabor-filter outputs using the separable convolutions described in Appendix A.

4. Compute motion energy as the squared sum of the sine- and cosine-phase Gabor filter

outputs.

5. Convolve the resulting motion energies with a Gaussian to approximate the integral in Parseval's theorem.

6. Find the "best" choice of u and v given by Equations (11) or (12), e.g., by employing the Gauss-Newton gradient-descent method or the parallel technique presented in Section 3.1.3.

7. Compute the uncertainty in the velocity estimate as discussed in Section 3.2.

3.1.3 Parallel Distributed Processing

Electrophysiological studies of the middle temporal (MT) area in macaque and owl monkeys reveal cells that are velocity tuned. Thus, it is generally believed that one of the functions of MT cells is to encode local image velocity. This section describes how Equation (12) can be used to compute a distributed representation of image velocity.

The distributed representation of image velocity is made up of velocity-tuned units analogous to the velocity-tuned cells of area MT. The outputs of each of the velocity-tuned units are computed in parallel by combining the motion-energy measurements (recall that the motion-energy filters are not themselves velocity-tuned since they confound spatial-frequency, temporal-frequency, and image contrast).

The last step in the algorithm in Section 3.1.2 is to find the maximum of a two-parameter function, $f(u, v)$ in Equation (12). One way to locate this maximum is to evaluate the function in parallel at a number of points (say, on a fixed square grid), and pick the largest result. The maximum can be located to any precision by using a finer or coarser grid. The grid need only be of limited extent since bandpass filtering limits the range of possible velocities (as discussed in Section 2.2). In the context of the model each point on the grid corresponds to a velocity. Thus, evaluating the function for a particular point on the grid gives an output that is velocity-tuned.

For a fixed velocity the predicted motion energies $\mathcal{R}_i(u, v)$ defined by Equation (9) are fixed constants, denote them by w_{in} where each i corresponds to a different motion-energy filter and

each n corresponds to a different velocity. We may rewrite Equation (12) for a fixed $\vec{\theta}$ as

$$f_n = \exp\left\{-\sum_{i=1}^{12} \frac{1}{c^2}\left[m_i - \overline{m}_i \frac{w_{in}}{\overline{w}_{in}}\right]^2\right\} \tag{13}$$

where c is proportional to the average of the m_i's, f_n is the response of a single velocity-tuned unit, and w_{in} and $\overline{w}_{in}$ are constant weights corresponding to the ith filter and the nth velocity. A mechanism that computes a velocity-tuned output from the motion-energy measurements performs the following operations:

1. A linear stage, a weighted summation given by $\left(m_i - \overline{m}_i \frac{w_{in}}{\overline{w}_{in}}\right)$.
2. A nonlinear stage, squaring.
3. A second linear stage, the summation over i.
4. A second nonlinear stage, multiplication by $\frac{1}{c^2}$ and exponentiation.

The model's computations are simply a series of linear steps (convolutions, weighted sums) alternating with point nonlinearities (squaring, exponentiation). The model is therefore encompassed by the general framework for parallel distributed processing put forth by Rummelhart and McClelland [119].

An example of the resulting distributed representation is shown in Figure 5 that displays a map of velocity space with each point corresponding to a particular velocity. The brightness at each point is the velocity-tuned output for that particular velocity. The maximum in the distribution of outputs corresponds to the velocity estimate.

3.1.4 Some Results

All of the results presented in this chapter were produced with a single choice for each of the model's parameters — the spatial frequency tuning of each Gabor filter is $\sqrt{\omega_{x_0}^2 + \omega_{y_0}^2} = 1/4$ cycles per pixel; the temporal frequency tunings are either $\omega_{t_0} = 0$ cycles per frame (stationary filters), or $\omega_{t_0} = \pm 1/4$ cycles per frame (right/left, up/down, etc.); the standard deviation of all of the spatial Gaussians is $\sigma_x = \sigma_y = 4$ (the spatial kernel size of the filters is 23 pixels) and that of the temporal Gaussians is $\sigma_t = 1$ (the temporal kernel size is 7 frames). Except for the

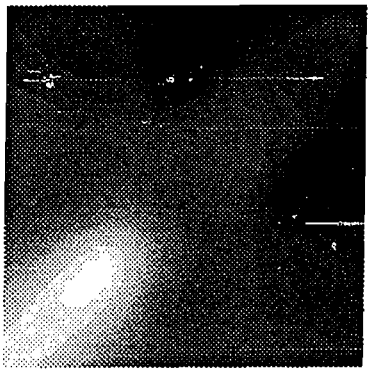


Figure 5: Distributed representation of image velocity for a random-dot field moving leftward and downward one pixel per frame. Each point in the image corresponds to a different velocity — for example, $\vec{\theta} = (0,0)$ is at the center of the image, $\vec{\theta} = (2,2)$ is at the top-right corner. The maximum in the distribution of outputs corresponds to the velocity estimate.

Yosemite fly-through sequence discussed below, all of the results are computed using only the lowest level of the pyramid.

Each vector in the flow fields depicted below represents a motion in a direction given by the vector's angle at a speed given by the vector's length. Errors in the velocity estimates are expressed in terms of the percentage error in each component of the actual velocity vectors.

Translating Image Sequences. Translating image sequences were generated from a textured image by: (1) blowing the image up to four-times its original size; (2) shifting the resulting image by an integral number of pixels i horizontally and j vertically for each consecutive frame; (3) reducing each image in the resulting sequence back to the original resolution. The final result is an image sequence with velocity $(i/4, j/4)$ pixels per frame.

The model gives accurate velocity estimates (within 10% of the actual velocities) for translating image sequences of a wide variety of textured patterns including random-dot patterns (with dot densities ranging from 5 to 50%), images of fractal textures[1], some sine-grating plaid patterns (discussed in Section 3.3), and natural textures (discussed below).

[1]Brownian fractal functions (see Chapter 7 for definitions and references) are characterized by similarity across scales, and have an expected power spectrum that falls off as $P(\omega) \sim \omega^{-\beta}$ for some constant β. Fractals may be used to generate natural-looking textures.

Noise Sensitivity. Translating image sequences of random-dot textures and Gaussian white-noise random textures were used to study the error in the velocity estimates. For image sequences with speeds ranging from 0.0 to 1.75 pixels per frame, the absolute value of the error in the velocity estimates is proportional to the actual speed (see figure 11). The mean percentage error is -2.9% and the standard deviation of 3.6%.

Noise sensitivity was studied by adding spatiotemporal Gaussian white-noise to translating random-dot sequences. Define the signal-to-noise ratio (S/N) to be the brightness of the image dots divided by the standard deviation of the noise. If $S/N = 10$, then the mean percentage error in the estimates is -4.3% and the standard deviation is 4.1%. This demonstrates that when the standard deviation of the sensor noise is as much as 10% of the sensor's dynamic range most velocity estimates are still within 10% of the actual values.

Images of Natural Textures. Image sequences were generated from each of the 14 natural textures shown in Figure 6(a). A sample flow field, shown in 6(b), was extracted from an image sequence of the straw texture in the upper-left corner of 6(a). The model correctly estimates the velocity (to within 10%) for every one of these textures. This is particularly impressive for the straw texture in the upper-left corner, the brick texture in lower-right corner, and the texture second from the lower-right corner of 6(a) because they have such strong spatial orientations. The model is capable of recovering accurate velocity estimates for these textures since it normalizes each spatial orientation separately in Equations (11) and (12). Conversely if we were to normalize the filter outputs isotropically (i.e., by dividing each motion energy by the sum of them all), then the estimates for these three textures would be erroneous.

A Rotating Spiral. Figure 7(a) shows one frame of a rotating spiral image sequence. The spiral, defined in polar coordinates by $r = \theta$, was rotated counter-clockwise one full revolution over seven frames. Figure 7(b) shows the extracted flow field. The flow vectors point inward corresponding to what human observers see.

A Rotating Sphere. Figure 8(a) shows one frame of a random-dot image sequence of a sphere rotating in front of a stationary background. Figure 8(b) shows the actual flow field

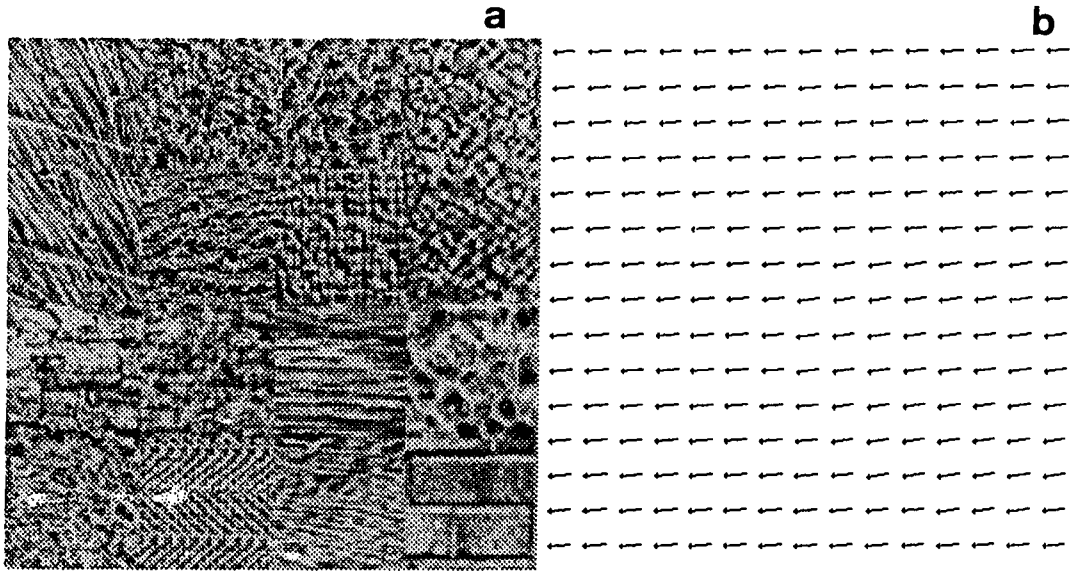


Figure 6: (a) Fourteen natural textures (the two texture squares in the upper-left are the same, and so are the two in the upper-right). Each texture square was used to generate motion sequences translating $1/2$ pixels per frame in each of eight directions. The velocities extracted by the model are accurate to within 10%. (b) Example flow field extracted from a motion sequence generated from the straw texture in the upper-left corner of (a). The actual motion was $(-0.5, 0.0)$. The mean of the extracted velocities is $(-0.473, -0.04)$ and the standard deviation for both the horizontal and vertical components is 0.01.

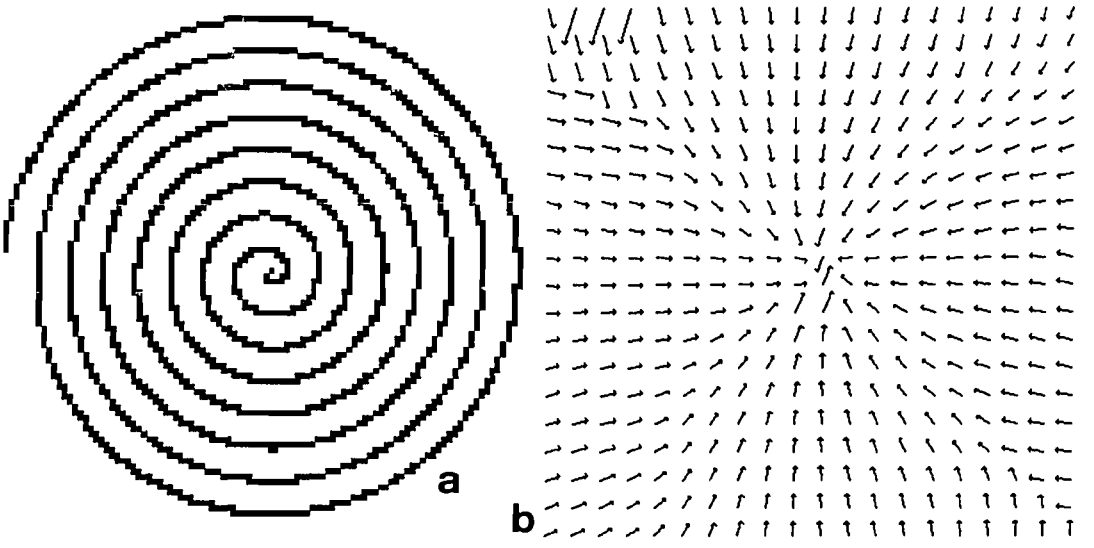


Figure 7: (a) A frame from a motion sequence of a counter-clockwise rotating spiral. The perceived direction of motion is toward the center of the image and the actual displacement in that direction is $2\pi/7$ pixels per frame. (b) The extracted flow field. For 72% of the flow vectors the estimated speed is within 10% of the actual displacement. For 94% of the flow vectors the estimated speed is within 20% of the actual displacement.

for this image sequence, 8(c) shows the flow field extracted by the model, and 8(d) shows the difference between them. The impact of the Gaussian smoothing is clearly evident as there are errors along the motion boundary.

A Realistic Example. Figure 9(a) shows one frame of a computer-generated image sequence flying through Yosemite valley. Each frame was generated by mapping an aerial photograph onto a digital-terrain map (altitude map). The observer is moving toward the horizon. The clouds in the background were generated with fractals (see Chapter 7) and move to the right while changing their shape over time.

Since the image velocities in the Yosemite fly-through image sequence are as high as 5 pixels per frame, we must use three levels from the pyramid. In future research, I hope to develop a rule for automatically combining estimates from the different levels. For now, I simply pick the level that is most appropriate for a given image region — the level zero estimate is chosen if the actual velocity is between 0 and 1.25 pixels per frame, the level one estimate is chosen if it is between 1.25 and 2.5 pixels per frame, and the level two estimate is chosen if it is between

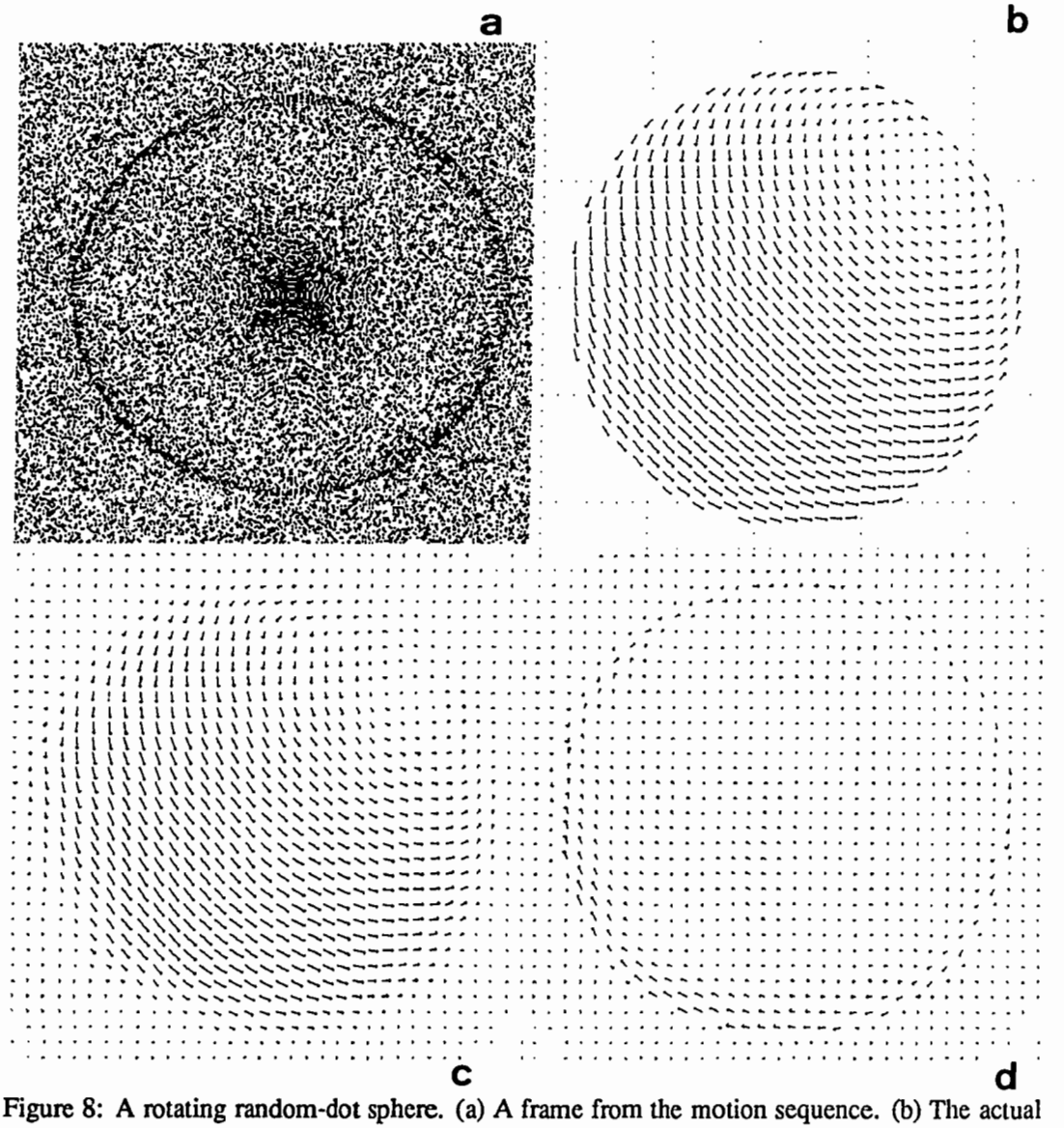


Figure 8: A rotating random-dot sphere. (a) A frame from the motion sequence. (b) The actual flow field. (c) Flow field extracted by the model. (d) Difference between (b) and (c).

2.5 and 5.0 pixels per frame.

In the yosemite fly-through image sequence, there are regions of low contrast adjacent to high contrast regions (e.g., the face of El Capitan and the cloud region are of low contrast). This exacerbates the smoothing problem as discussed in Section 3.1.2. For this image sequence, contrast was first "equalized" by computing the zero-crossings [97] of each image. The model was then applied to the resulting zero-crossing image sequence. Using the zero-crossing image sequence improves the accuracy of the velocity estimates only within the low contrast regions. If we window the low contrast regions to remove them from the context of the surrounding high contrast regions, then there is little difference between the accuracy of the velocity estimates using either the zero-crossing image sequence or the original grey-level image sequence. Zero-crossings were used simply for convenience. I expect that other mechanisms for automatic gain control (contrast adaptation) will prove more fruitful.

Figure 9(b) shows the actual flow field for this image sequence, 9(c) shows the flow field extracted by the model, and 9(d) shows the difference between them. The impact of Gaussian smoothing is evident along the boundary at the horizon. Small errors are also evident on the face of El Capitan (in the lower-left) since it is moving with high speed (see the discussion in Section 3.1.2), and in the cloud region since the clouds change shape over time while moving rightward.

3.2 Image-Flow Uncertainty

Information from perceptual sources is inherently noisy and uncertain. A sensing system can make substantial gains by explicitly representing the uncertainty in sensor data and taking actions to reduce it. In particular estimates of image motion are exhibit variability due to the stochastic nature of image textures. Decisions and computations that rely on motion estimates will be more robust if we keep track of uncertainty.

This section uses tools from probability and statistical estimation theory to formulate a measure of uncertainty for image flow by characterizing the variability in the model's velocity estimates for translating image sequences of Gaussian white-noise random textures. Since image textures are stochastic the predicted motion energies given by Equation (8) are correct only on

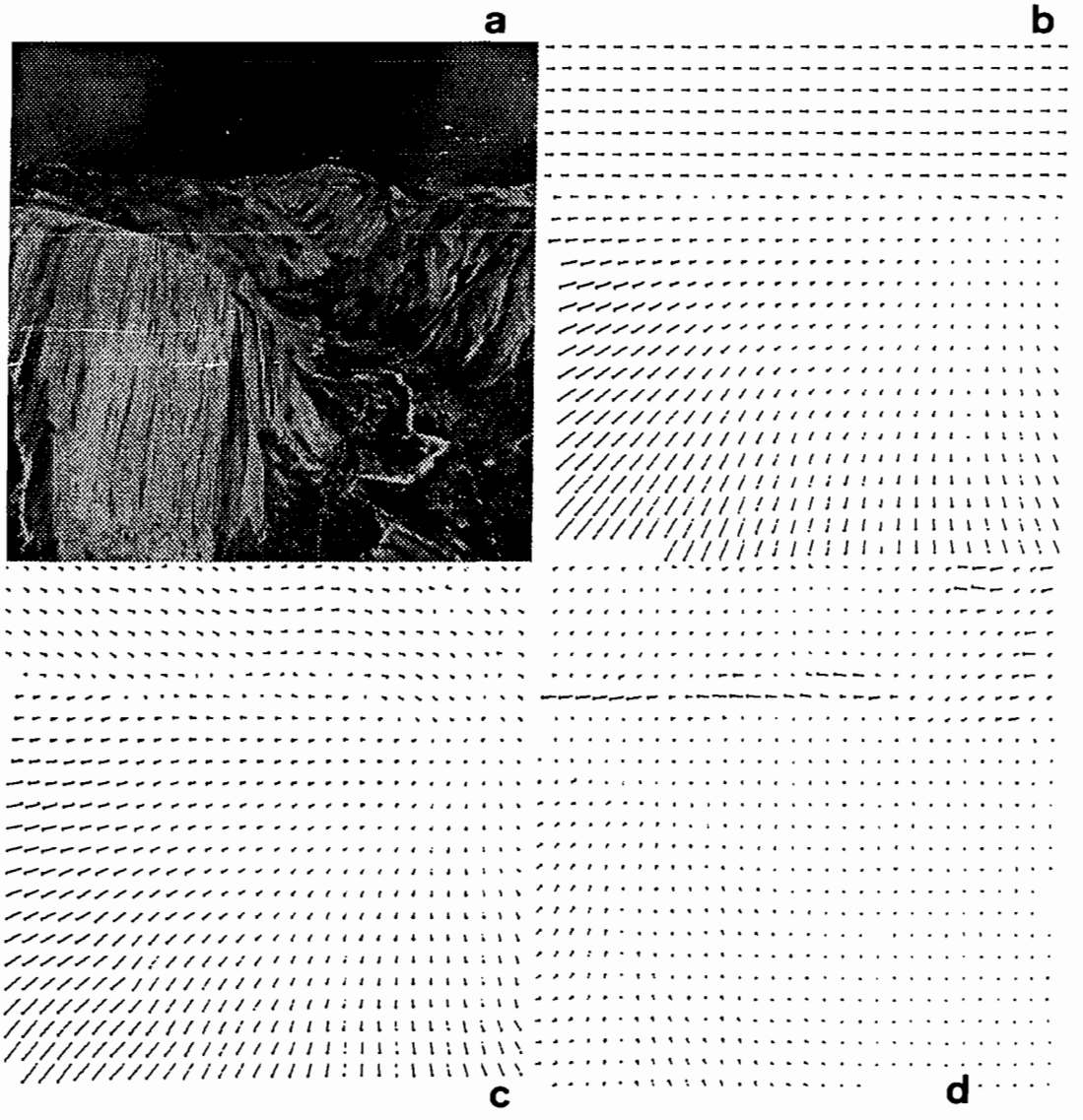


Figure 9: (a) One frame of an image sequence flying through Yosemite valley. (b) The actual flow field. (c) Flow field extracted by the model. (d) Difference between (b) and (c).

average. For a particular region of a translating image sequence the measured motion energies deviate from the expected value.

Below I posit an additive Gausian model for the variability in the motion energy measurements. If a normal distribution is a valid approximation for this variability, then the least squares estimate of image velocity is optimal in the sense that it is equal to the maximum-likelihood estimate (MLE). Normality can be tested empirically by translating a camera a fixed distance in front of a variety of planar textured surfaces. If the camera motion is known, then the actual image translation is easily computed, and we can compare the predicted motion energies given by Equation (8) to those measured from the image sequence.

However, if normal distributions were *not* valid approximations for the measurement variability, then least-squares estimation would not be optimal. In addition, the uncertainty measure formulated below would not be accurate. In spite of this, some insight can be gained by proceeding under the normality assumption. In future research, I hope to extend the analysis presented below to allow for more general assumptions about the form of the distributions.

First, I review some aspects of statistical parameter estimation in the presence of additive noise. I use the notation $\hat{\theta}$ to denote estimates of the parameter θ.

Consider the case in which we take independent measurements, $\vec{m} = (m_1, \ldots, m_{12})$, that are nonlinearly related to an unknown parameter, $\vec{\theta} = (u, v)$, in the presence of zero-mean additive Gaussian noise, $\vec{n} = (n_1, \ldots, n_{12})$,

$$\begin{aligned} \vec{m} &= \vec{R}(\vec{\theta}) + \vec{n} \\ n_i &\sim N(0, \sigma_i^2) \end{aligned} \tag{14}$$

for some nonlinear vector-function, $\vec{R}(\vec{\theta}) = [\mathcal{R}_1(\vec{\theta}), \ldots, \mathcal{R}_{12}(\vec{\theta})]$. Equation (14) may be rewritten as

$$[m_i - \mathcal{R}_i(\vec{\theta})] \sim N(0, \sigma_i^2) \tag{15}$$

The varaibility in the measurements may be represented by the Fisher information matrix (see DeGroot [34] for the definition in the scalar case). For a jointly normal density the information matrix, denoted by Λ_m^{-1} is the inverse of the variance-covariance matrix, Λ_m, and the conditional

probability density is given by

$$f(\vec{m}|\vec{\theta}) = (2\pi)^{-6}|\Lambda_m|^{-1/2} \exp\left\{\sum_{i=1}^{12} \frac{1}{2\sigma_i^2}[m_i - \mathcal{R}_i(\vec{\theta})]^2\right\} \tag{16}$$

The posterior density, $f(\vec{\theta}|\vec{m})$, is the probability that a certain value of $\vec{\theta}$ is equal to the true value, given the measurements $\vec{m}$. In the absence of prior information on the parameter $\vec{\theta}$, the posterior density and the conditional density are one and the same. The maximum-likelihood estimate (MLE), $\hat{\theta} = (\hat{u}, \hat{v})$, is that which maximizes the conditional density, thus maximizing the probability that the estimate is equal to the true value. For additive Gaussian noise the MLE is the same as the least-squares estimate.

The uncertainty in the estimate may be represented as an information matrix, $\Lambda_\theta^{-1}(\hat{\theta})$, computed from Λ_m^{-1} and from $\hat{\theta}$ (see Melsa and Cohn [100] for derivation):

$$\Lambda_\theta^{-1}(\hat{\theta}) = \mathbf{J}^T(\hat{\theta})\Lambda_m^{-1}\mathbf{J}(\hat{\theta}) \tag{17}$$

where $\mathbf{J}(\vec{\theta})$ is the Jacobian matrix of $\vec{R}(\vec{\theta})$ and $\mathbf{J}^T(\vec{\theta})$ is the transpose of $\mathbf{J}(\vec{\theta})$. The information matrix, $\Lambda_\theta^{-1}(\hat{\theta})$, is a random variable that depends on the estimate $\hat{\theta}$.

The eigenvectors and eigenvalues of the information matrix, $\Lambda_\theta^{-1}(\hat{\theta})$, are the directions and values in parameter space (e.g., in image-velocity space) of minimum and maximum information. The mean-squared-error, given by the trace of the variance-covariance matrix, is an estimate of the actual squared-error of the estimate, that is, $\mathrm{Tr}[\Lambda_\theta(\hat{\theta})]$ is an estimate of $\|(u - \hat{u}, v - \hat{v})\|^2$.

If there is only partial information about $\vec{\theta}$ then the minimum-information eigenvalue is zero. For example, in the presence of the aperture problem (as discussed in Section 3.3) there is only partial information about image motion. We represent uncertainty with the information matrix, $\Lambda_\theta^{-1}(\hat{\theta})$, instead of using the variance-covariance matrix, $\Lambda_\theta(\hat{\theta})$, because the latter may be undefined when there is only partial information.

Equation (17) may be used to compute the uncertainty of an image flow estimate. But we must have a statistical model, called a *sensor model*, of the measurement variability (denoted above by Λ_m^{-1}). The predicted motion energies given by Equation (8) are correct only on average; the difference between the measured and predicted motion energies is given by

$$m_i = K_i\mathcal{R}_i(u, v) + n_i \tag{18}$$

where $\mathcal{R}_i(u,v)$ is defined in Equation (9), K_i is an unknown constant that depends on image contrast, and n_i is additive process variability.

The procedure presented in Section 3.1 for estimating image velocity picks the estimate, $(\hat{u},\hat{v})$, to minimize

$$\begin{aligned} l(u,v) &= \sum_{i=1}^{12}\left[m_i - \hat{K}_i\mathcal{R}_i(u,v)\right]^2 \\ \hat{K}_i &= \frac{\overline{m}_i}{\overline{\mathcal{R}}_i(u,v)} \end{aligned} \tag{19}$$

where $\mathcal{R}_i(u,v)$ is defined in Equation (9), and $\hat{K}_i$ is used as an estimate of K_i with $\overline{m}_i$ and $\overline{\mathcal{R}}_i(u,v)$ as defined in Equation (10).

As discussed at the beginning of this section, we posit a Gaussian model for the variability in the motion energy measurements

$$\left[m_i - \hat{K}_i\mathcal{R}_i(u,v)\right] \sim N(0,\sigma_i^2) \tag{20}$$

where σ_i^2 is the variance of the additive Gaussian variability.

Figure 10 is an empirical test of the Gausianity assumption. The plot shows a histogram of $[m_i - \hat{K}_i\mathcal{K}_i(u,v)]$ for one motion-energy filter over four hundred trials. The data in this histogram pass both the Chi-squared and the Kolmogorov-Smimov tests for Gaussianity. However, other examples fail these tests for Gaussianity. Further experimentation with real image sequences is called for.

The variance of $[m_i - \hat{K}_i\mathcal{R}_i(u,v)]$ is given by

$$\begin{aligned} \sigma_i^2(u,v) &= \mathrm{var}\left(m_i - \overline{m}_i\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)}\right) \\ &= \left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)} - 1\right)^2 \mathrm{var}(m_i) \\ &\quad + \left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)}\right)^2 [\mathrm{var}(m_2) + \mathrm{var}(m_3)] \\ &\quad + 2\left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)} - 1\right)\left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)}\right)[\mathrm{cov}(m_i,m_2) + \mathrm{cov}(m_i,m_3)] \\ &\quad + 2\left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)}\right)^2 [\mathrm{cov}(m_2,m_3)] \end{aligned} \tag{21}$$

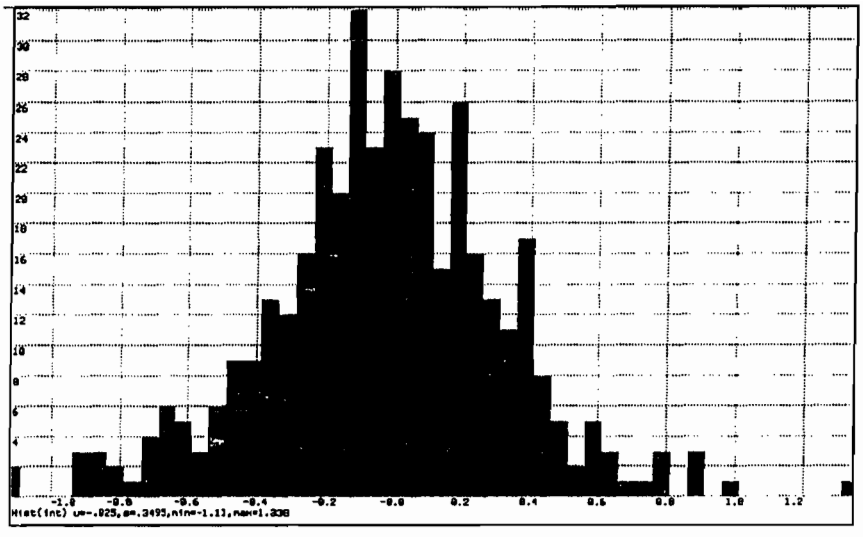


Figure 10: Four hundred Gaussian white-noise random textures were generated, and each was used to generate a translating image sequence with the same velocity (one pixel per frame upward and rightward). The plot shows a histogram of $[m_i - \hat{K}_i \mathcal{R}_i(u,v)]$ for the motion-energy filter that is most sensitive for rightward motion. The data in this histogram pass both the Chi-squared and the Kolmogorov-Smirnov tests for Gaussianity. The distribution is zero-mean and its variance is 0.12.

where m_i is the output of the ith filter, m_1 and m_2 are the outputs of the two filters that share the same orientation the ith filter, $\mathcal{R}_i(u,v)$, $\mathcal{R}_1(u,v)$ and $\mathcal{R}_2(u,v)$ are the corresponding predicted motion energies given by Equation (9), and $\text{var}(m_i) = \text{cov}(m_i, m_i)$. In Appendices B and C I derive an equation for the covariances of the motion energy measurements, $\text{cov}(m_i, m_j)$, for image sequences of translating Gaussian white-noise random textures.

Table 1 in Appendix C shows empirical tests of the accuracy of the sensor model given by Equation (21). The average percent error in the variance estimates is 17.7%. So there is reasonably good agreement in the table between the actual and simulated measurement variability for translating Gaussian white-noise textures. However since $\sigma_i(u,v)$ depends on the actual value of (u,v) we must make one further approximation — we approximate the measurement variability using the image velocity estimate, $\sigma_i(\hat{u},\hat{v}) \approx \sigma_i(u,v)$.

The conditional probability density of the motion-energy measurements given the actual

image velocity is therefore approximated by

$$f(\vec{m}|u,v) \approx (2\pi)^{-6}|\Lambda_m|^{-1/2} \exp\left\{\frac{1}{2\sigma_i^2(\hat{u},\hat{v})}\sum_{i=1}^{12}\left[m_i - \hat{K}_i\mathcal{R}_i(u,v)\right]^2\right\} \tag{22}$$

where, assuming independence of the motion energy measurements, Λ_m is approximated by

$$\Lambda_m \approx \begin{pmatrix} \sigma_1^2(\hat{u},\hat{v}) & 0 & \cdots & 0 \\ 0 & \sigma_2^2(\hat{u},\hat{v}) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \sigma_{12}^2(\hat{u},\hat{v}) \end{pmatrix} \tag{23}$$

In this case the information matrix, Λ_m^{-1}, is a diagonal matrix with entries equal to $1/\sigma_i^2(\hat{u},\hat{v})$, and the Jacobian matrix[2] is given by

$$\mathbf{J}(\hat{u},\hat{v}) = \begin{pmatrix} \hat{K}_1\frac{\partial\mathcal{R}_1(\hat{u},\hat{v})}{\partial u} & \hat{K}_1\frac{\partial\mathcal{R}_1(\hat{u},\hat{v})}{\partial v} \\ \vdots & \vdots \\ \hat{K}_{12}\frac{\partial\mathcal{R}_{12}(\hat{u},\hat{v})}{\partial u} & \hat{K}_{12}\frac{\partial\mathcal{R}_{12}(\hat{u},\hat{v})}{\partial v} \end{pmatrix} \tag{24}$$

where $\frac{\partial\mathcal{R}_i(u,v)}{\partial u}$ and $\frac{\partial\mathcal{R}_i(u,v)}{\partial v}$ are obtained by differentiating Equation (9).

Equations (17), (21), and (24) may be used to estimate image-flow uncertainty as follows:

1. Computing the image velocity estimate, $(\hat{u},\hat{v})$, as discussed in Section 3.1.
2. Computing the measurement variability estimates, $\sigma_i(\hat{u},\hat{v})$, using Equation (21).
3. Computing the information matrix, $\Lambda_\theta^{-1}(\hat{u},\hat{v})$, using Equations (17) and (24).

A test of the accuracy of the uncertainty measure is to compare the mean-squared-error, $\mathrm{Tr}[\Lambda_\theta(\hat{u},\hat{v})]$, with the actual squared-error of velocity estimates, $\|(u-\hat{u},v-\hat{v})\|^2$. Figure 11 shows that the uncertainty measure reflects the actual error for translating Gaussian white-noise random textures.

However, the uncertainty measure significantly underestimates the actual errors for the Yosemite fly-through image sequence (Figure 9) because these errors are mainly due to the blurring problem discussed in Section 3.1.2, not due to the motion-energy measurement variability. It may be possible to extend the sensor model to account for other sources of error like camera noise and local variations in image velocity.

[2]To be thorough, we could treat the K_i's as variables and include derivatives $\frac{\partial[K_i\mathcal{R}_i(\hat{u},\hat{v})]}{\partial K_i} = \mathcal{R}_i(\hat{u},\hat{v})$ in the Jacobian matrix. But we are not interested in estimating K_i, so there is no reason to estimate the uncertainty in $\hat{K}_i$.

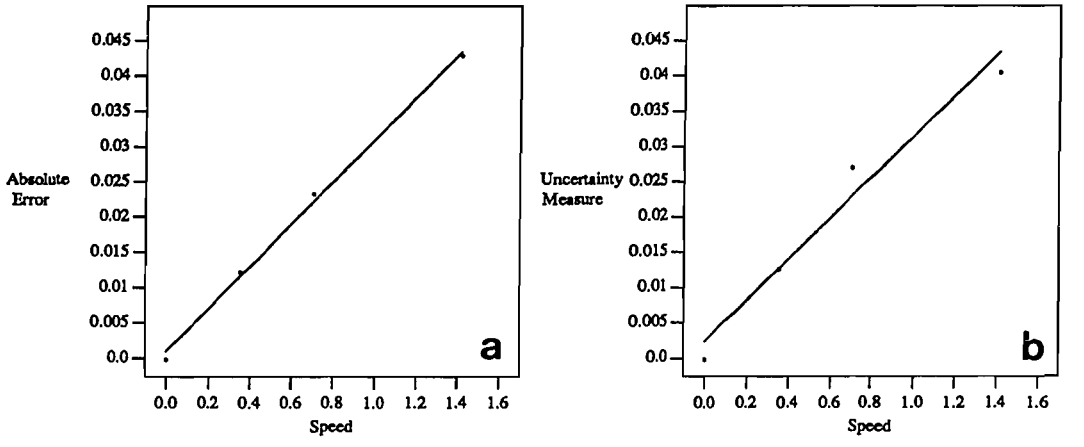


Figure 11: Two hundred translating Gaussian white-noise random textures were generated with each of four different velocities ranging from 0.0 pixels per frame to $\sqrt{2}$ pixels per frame. (a) The average absolute error in the velocity estimates as a function of speed. The least-squares best-fit line is drawn through the data points with slope 0.029 and y-intercept 0.0024. (b) The average square-root of the trace of the estimated variance-covariance matrix as a function of speed plotted on the same scale as in (a). The least-squares best-fit line is drawn through the data points with slope 0.030 and y-intercept 0.0010.

3.3 Dealing with the Aperture Problem

In this section, I use a class of moving stimuli known as sine-grating plaids in order to test the model's capability for solving the aperture problem and I compare the model's performance to that of the human visual system. I also propose using the uncertainty measure presented in the previous section to recognize when there is an ambiguous velocity estimate resulting from the motion of a strongly oriented pattern.

3.3.1 Sine-Grating Plaids

A sine-grating plaid is the sum of two moving gratings and may be seen as a single coherent plaid motion. The gratings are not combined as the vector sum or vector average of the two component normal-flow velocities, but rather as the intersection of the perpendiculars to the two velocity vectors. Figure 12(a) depicts a single grating moving behind an aperture — the arrows represent flow vectors and the diagonal line represents the locus of velocities compatible with the grating's motion. There are an infinite number of such compatible motions any of

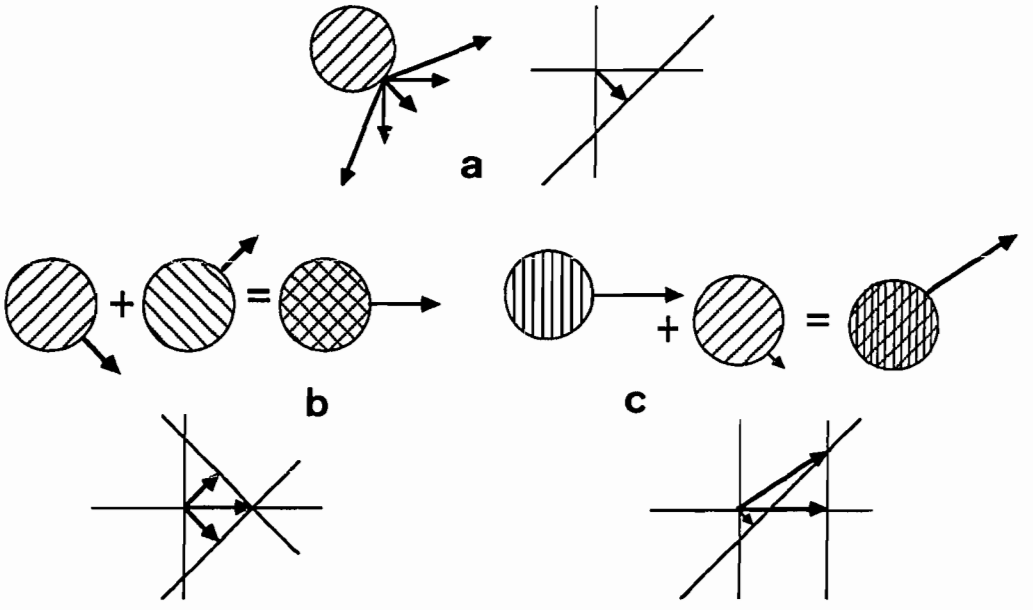


Figure 12: (redrawn from Adelson and Movshon [3]) The perceived motion of two moving gratings is the intersection of the perpendiculars to the two velocity vectors. (a) A single moving grating — the diagonal line indicates the locus of velocities compatible with the motion of the grating. (b) and (c) Plaids composed of two moving gratings. The lines give the possible motions of each grating alone. Their intersection is the only shared motion, and corresponds to what is seen.

which will result in exactly the same stimulus. Figure 12(b) shows a plaid composed of two orthogonal gratings moving at the same speed — the intersection of the perpendiculars to the two normal-flow velocities (the intersection of the two constraint lines) is the only shared motion, and corresponds to what is seen. Figure 12(c) shows a plaid composed of two oblique gratings, one moving slowly and the other more rapidly — one grating moves rightward and the other moves downward and rightward, but the pattern moves *upward* and rightward.

The model recovers the correct pattern-flow velocity for a number of such plaids. Examples of flow fields extracted by the model for plaids made up of gratings with equal contrasts and spatial frequencies are shown in Figure 13. The combined motion extracted by the model in both 13(a) and 13(b) is accurate to within 5%.

The model does not always recover the correct pattern-flow velocity for sine-grating plaids. For example, the model's estimates are in error (correct direction of motion but wrong speed) when the spatial frequencies of the gratings are not equal to the spatial-frequency tuning of the

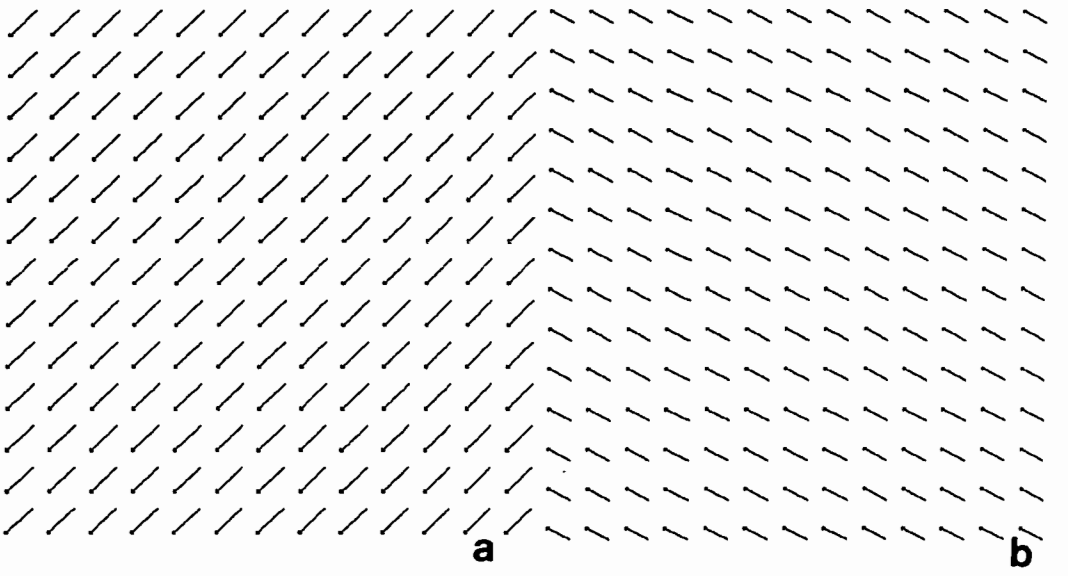


Figure 13: (a) Flow field extracted by the model for a plaid pattern made up of a sine grating moving leftward one pixel per frame plus a sine grating moving downward one pixel per frame. The combined motion extracted by the model is one pixel leftward and one pixel downward each frame. (b) Flow field for a plaid pattern made up of a sine grating moving leftward one pixel per frame plus a sine grating moving downward and leftward a quarter pixel each frame. The counter-intuitive combined motion is leftward one pixel per frame and *upward* a half pixel per frame as shown in the flow field extracted by the model. The spatial frequency of the gratings for both (a) and (b) was 0.25 cycles pixel^{-1}.

filters.

3.3.2 Sine-Grating Plaids and the Aperture Problem

Adelson and Movshon [3] studied the phenomenon of coherence by varying the angle between the two gratings, their relative contrasts, and their relative spatial frequencies. They found that for a range of relative angles, contrasts, and spatial frequencies the two gratings are seen as a single coherent plaid motion, and that beyond this range the two gratings look like separate motions moving past one another. The phenomenon of coherence tests the human visual system's ability to solve the aperture problem; given the ambiguous motion of a single moving grating, how much additional information is needed from the second grating to give an unambigous coherent percept?

The model is capable of extracting the correct pattern-flow velocity for plaids that have large

differences in contrast, e.g., for plaids made up of orthogonal gratings the velocity estimates are accurate to within 10% for contrast ratios of greater than 32 : 1. This is comparable with human performance [1]. As the contrast difference between the two component gratings gets larger than this the model begins to tilt the extracted velocity vector toward the higher contrast grating. Although the perceived velocity of plaids has not yet been measured precisely Adelson [1] notes that observers also see the direction of motion tilt toward the higher contrast grating when the relative contrast difference is large.

To withstand large contrast ratios it is crucial that the spatial bandwidths of the model's filters be less than their temporal bandwidths — in the frequency domain, this means that the filters are oblong hotdog-shaped (longer in t than in x and y) instead of spherical in shape. As an illustrative example, consider a plaid made up of rightward- and upward-moving gratings. The idea of normalizing the filter outputs separately for each spatial orientation is that the upward- and downward-sensitive filters should give the same responses relative to one another regardless of the contrast ratio between the two gratings. If the filters were spherical in shape, then the response of the downward-sensitive filter would be dominated by the rightward-moving grating (the impulse from the rightward-moving grating is closest to the center-frequency of the downward-sensitive filter). This would be bad because we want the relative responses of the upward- and downward-sensitive filters to be unaffected by varying the contrast of the rightward-moving grating. But, since the filters are oblong in shape the response of the downward-sensitive filter is dominated by the grating moving upward for a wide range of relative contrasts.

3.3.3 Recognizing Ambiguity

An isotropic texture (e.g., a random-dot field) does not suffer from the aperture problem since there is enough information within a local window to disambiguate the true direction of motion. A strongly oriented pattern (e.g., a sine grating) offers only partial information about image velocity. Between these two extremes there is a continuum of stimuli offering information about image velocity that is more and more ambiguous. The level of ambiguity should be reflected by the level of uncertainty in the velocity estimate.

The distributed representation of image velocity introduced in Section 3.1.3 forms a surface in velocity space; the height of the surface at a particular velocity is the likelihood that it is

the true velocity. Some examples will illustrate that ambiguity due to the aperture problem is reflected by the shape of this response surface.

Figure 14 shows the distributed representation of image velocity for some sine-grating plaids. As the relative contrast of one the component gratings is varied the peak in the surface gets broader in one direction. This is evident by comparing Figures 14(a), (b), (c), and (d). In (a), the two component gratings are of equal contrast and the peak is symmetrical. When the contrast ratio is increased as in (b) and (c), the location of the peak does not change, but its shape elongates in one direction. Eventually as shown in (d), the peak turns into a ridge.

Figure 15 shows the distributed represenations for image sequences generated from the straw-texture image. There is enough information in these image sequences for the model to disambiguate the true direction of motion as there are clearly defined peaks in the distributions. The shape of each peak matches the orientation of the texture thereby reflecting the image-flow uncertainty.

When there is an unambiguous peak we can extract the correct pattern-flow velocity, but how do we know if there is a ridge or a peak? Intuitively, it is a peak if it falls off sharply in all directions and it is a ridge if it stays constant in one direction. We know from differential geometry (for example, see doCarmo [35]) that a surface can be characterized locally by its maximum and minimum curvatures. If the minimum curvature of a surface is small or zero at a point while the maximum curvature is large then the surface looks likc a ridge. If both curvatures are large then it looks like a peak.

In [62] I suggest using the minimum curvature of the surface at the peak divided by the height of the peak as a measure of ambiguity due to the aperture problem. We may pick a value to act as a threshold; if the curvature measure is above this value we pick the pattern flow given by the location of the peak, and if it falls below this value we may pick the normal flow vector or we may choose any other velocity along the ridge (a familiar example of when people see motion other than in the normal-flow direction is the barberpole illusion).

Instead, I propose that we use the information matrix introduced in Section 3.2 to recognize ambiguity. Define the *ambiguity mesure* as the quotient of the minimum-eigenvalue of the information matrix divided by its maximum eigenvalue. Figure 17(b) in Chapter 4 shows a plot of the ambiguity measure as the relative contrast of a plaid's component gratings is varied.

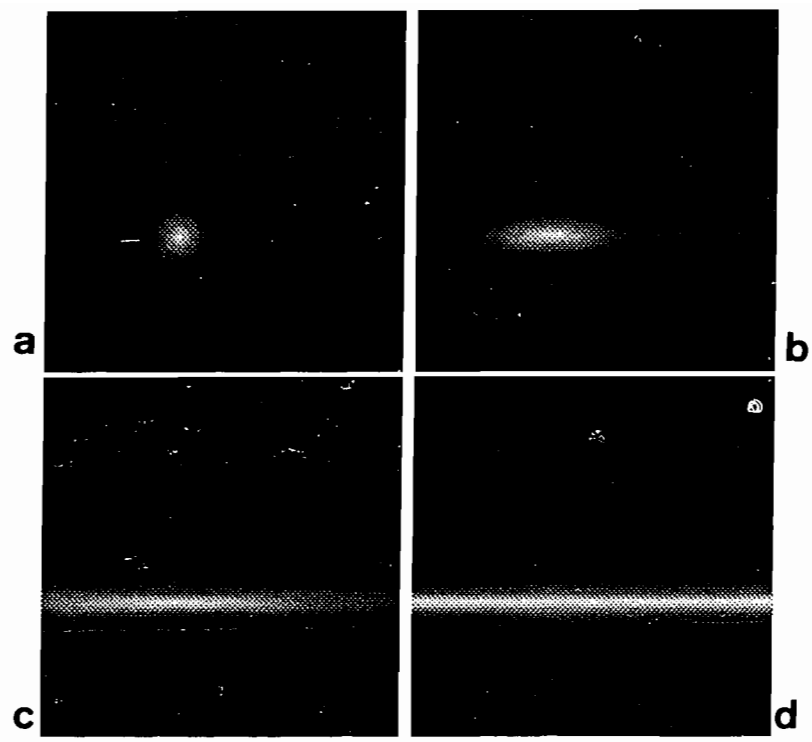


Figure 14: Distributed representation of image velocity for sine-grating plaids made up of orthogonal gratings. The gratings moved 1 pixel frame^{-1} leftward and downward and their spatial frequency was 0.25 cycles pixel^{-1}. (a) The two component gratings had the same contrast. The location of the maximum in the distribution corresponds to the velocity extracted by the model. (b) One grating had twice the contrast of the other grating. (c) One grating had four-times the contrast of the other grating. (d) One grating had zero contrast; the aperture problem is evident as there is a ridge of maxima. Each velocity-tuned unit along this ridge has the same output (to within 1 part in 100,000).

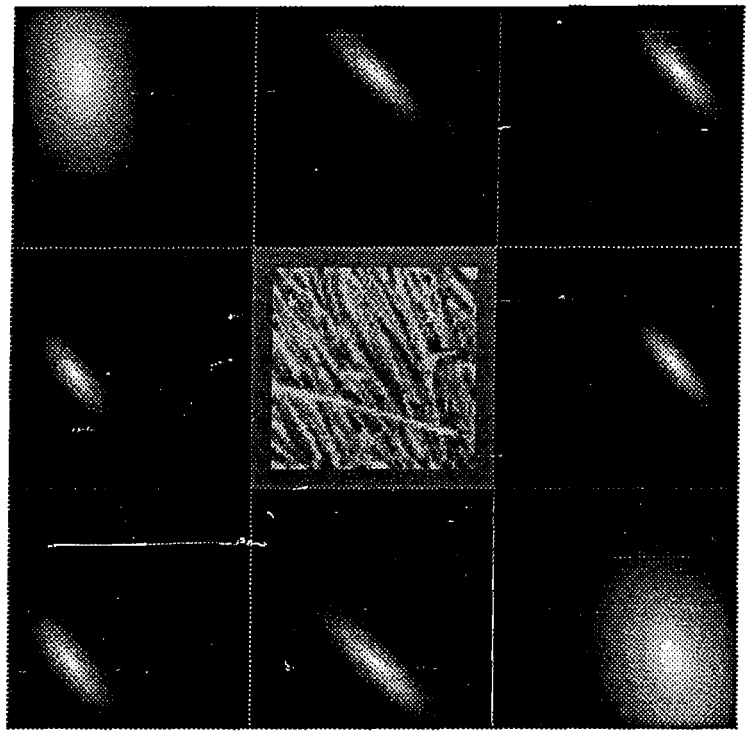


Figure 15: Translating image sequences were generated from the straw texture shown in the middle. Each pane shows the distributed representation of velocity computed from sequences moving $1/2$ pixels frame^{-1} in each of eight directions. The locations of the peaks in these distributions correspond to the velocities extracted by the model. The shape of each peak matches the orientation of the texture thereby reflecting the image-flow uncertainty.

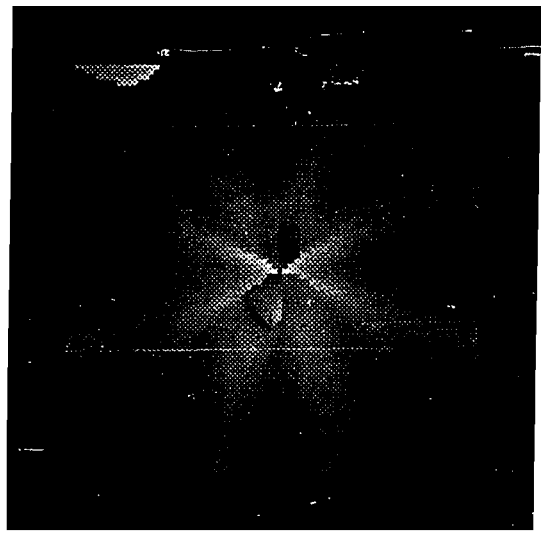


Figure 16: The brightness at each pixel is proportional to the ambiguity measure for the rotating spiral image sequence (Figure 7). The ambiguity measure reflects the ambiguity in the image sequence.

The ambiguity measure decreases monotonically with the contrast of the test grating for a wide range of relative contrasts.

Figure 16 shows the values of the ambiguity measure for each pixel of the rotating spiral image sequence (Figure 7). As we move away from the center of the image there is less and less curvature in the contour of the spiral. The ambiguity measure reflects this variation in the level of velocity ambiguity.

The results in Figures 17(b) and 16 indicate that the ambiguity measure may lead to a reliable test for ambiguity due to the aperture problem.

3.4 Summary

This chapter presents a model for computing local image velocity consonant with current views regarding the neurophysiology and psychophysics of motion perception. The power spectrum of a moving texture occupies a tilted plane in the spatiotemporal-frequency domain. The model uses 3-D (space-time) Gabor filters to sample this power spectrum and by combining the outputs of several such filters the model estimates the slope of the plane (i.e., the velocity of the moving

texture). The model gives accurate estimates of two-dimensional velocity for a wide variety of test cases including realistic images, sequences generated from images of natural textures, and some sine-grating plaid patterns.

The error in the velocity estimates for translating image sequences is from two sources. First, since image textures are stochastic, Equation (8) is correct only on average. Second, the maximum-likelihood estimate is equal to the least-squares estimate only for the case of additive Gaussian process variability. Thus least-squares estimation is optimal only if the normal approximation in Equation (22) is valid.

The primary source of error for realistic image sequences is that the model assumes image translation, ignoring motion boundaries, accelerations, deformations (rotation, divergence, shear), and motion transparency. Rather, the model computes the average image velocity within a Gaussian-shaped window.

A parallel implementation of the model results in a distributed representation of image velocity. The computations leading to this distributed representation are simply a series of linear steps (convolutions, weighted sums) alternating with point nonlinearities (squaring, exponentiation). The model is therefore encompassed by the general framework for parallel distributed processing put forth by Rummelhart and McClelland [119].

An image-flow sensor model is developed and it is demonstrated that the sensor model reflects the actual error in the velocity estimates for translating image sequences of Gaussian white-noise random textures. However, the sensor model significantly underestimates the actual errors for realistic image sequences because these errors are not simply due to the error in the motion-energy measurements. Rather, they are mainly due to blurring across deformations, accelerations, and motion boundaries.

Ambiguity due to the aperture problem is a special case of image flow uncertainty. It is proposed that the sensor model be used to test for ambiguity due to the aperture problem.

The model appears to solve the aperture problem as well as the human visual system since it extracts the correct velocity for patterns having large differences in contrast at different spatial orientations ($> 32:1$ contrast ratio for some patterns).

This chapter demonstrates the promise of computing optical flow using spatiotemporal filters. There are any number of related techniques using different filters, or using different rules

for combining the filter outputs. I suggest using psychophysical and electrophysiological experimentation to distinguish between them.

Chapter 4

Simulating Psychophysics

In this chapter, I use the image-flow model presented in the previous chapter to simulate psychophysical data. In [62] I compare the the computations performed by the model to the stages of the visual motion pathway of the primate brain, and I suggest how the model might be used to simulate electrophysiological data.

For the most part, simulating physiological and psychophysical data merely demonstrates that the model is consistent with some of the experimental results on biological motion perception. The emphasis in future research will be to compare the predictions made by this model to those made by alternative image-flow models and to test those predictions with further experiments. Thus, the model may prove to be an interesting framework for future research in the psychophysics and neurophysiology of motion perception as well as in computer vision.

Section 4.1 uses the error analysis of image-flow presented in Section 3.2 and Appendix C to simulate the psychophysical data on velocity discrimination, and to compute limits on the accuracy of velocity estimation for the human visual system. Section 4.2 uses the ambiguity measure presented in Section 3.2 to simulate psychophysical data on the coherence of sine-grating plaids.

4.1 Velocity Discrimination

Discrimination is the ability to decide, in the presence of uncertainty, whether or not two things are the same. In the case of velocity discrimination, the observer must decide whether the

stimulus is moving with one velocity or with another very similar velocity.

McKee et al [98,99] have measured human ability to discriminate the velocities of moving patterns. They find that judgements of relative velocity depend on velocity alone and only incidentally on other cues (like contrast and temporal frequency). Practiced obsevers can discriminate a 5% difference in speed for a wide range of velocities and a variety of moving stimuli. In other words velocity discrimination follows Weber's Law; the just noticeable difference Δs between speed s and speed $s + \Delta s$ is proportional to s, i.e., $\Delta s/s = 0.05$ or $(s + \Delta s)/s = 1.05$.

There are two experimental methods one might use to measure the Weber fraction for velocity discrimination. In a two-alternative forced choice experiment the observer is shown two displays, one with speed $s = s_1$ and the other with speed $s_2 = s + \Delta s$, and he must choose which one moved faster. Threshold discrimination is the Δs for which the observer picks the right display 75% of the time.

Mckee et al [98,99] used the method of single stimuli. On each trial the observer is shown one of five velocities chosen from a narrow range and was forced to judge whether the single sample was faster or slower than the mean of the range. No specific standard was ever presented; instead the observer judged the velocity shown on each trial against an implicit mean established by the sequence of trials. The stimulus contrast or spatial frequency was varied randomly from trial to trial. Because no standard was presented, the observer was forced to abstract the velocity standard, and his judgement was not influenced by the particular contrast or spatial frequency chosen for a standard stimulus. Threshold discrimination is the Δs for which the observer correctly responds "faster" or "slower" 75% of the time.

These two experimental techniques are equivalent and may be modeled in a similar manner. In what follows, I consider the two-alternative forced-choice paradigm with moving random texture stimuli.

Figure 11 shows that for moving random textures both the actual velocity errors and the image-flow uncertainty estimates are proportional speed, i.e.,

$$\sigma = \hat{\sigma} = cs \tag{25}$$

where $c \approx 0.03$.

For two patterns moving with speeds $s_1 = s$ and $s_2 = s + \Delta s$, we assume that the error in

the estimates of each of the speeds is normally distributed,

$$\begin{aligned} \hat{s}_1 &\sim N(s_1, \sigma_1^2) \\ \hat{s}_2 &\sim N(s_2, \sigma_2^2) \end{aligned} \tag{26}$$

i.e.,

$$\begin{aligned} (\hat{s}_2 - \hat{s}_1) &\sim N(s_2 - s_1, \sigma_1^2 + \sigma_2^2) \\ &\sim N[\Delta s, c^2(2s^2 + 2s\Delta s + \Delta s^2)] \end{aligned} \tag{27}$$

where s_i are the actual values, $\hat{s}_i$ are the estimates, and σ_i^2 are the variances.

At threshold discrimination the observer correctly chooses the faster stimuli 75% of the time, i.e.,

$$\Pr\{(\hat{s}_2 - \hat{s}_1) > 0\} = 0.75 \tag{28}$$

For a normal distribution, $z \sim N(\mu, \sigma^2)$, the probability that $z > 0$ is given by

$$\begin{aligned} \Pr\{z > 0\} &= \frac{1}{\sqrt{2\pi}\sigma} \int_0^\infty \exp[-\frac{(z-\mu)^2}{2\sigma^2}]dx \\ &= (1/2)\left[\text{erf}\left(\frac{\sqrt{2}\mu}{2\sigma}\right) + 1\right] \end{aligned} \tag{29}$$

where $\text{erf}(x)$ is the standard error function,

$$\text{erf}(x) = (2/\sqrt{\pi}) \int_0^x \exp(-\xi^2)d\xi \tag{30}$$

Combining Equations (25), (27), (28), and (29) gives

$$\text{erf}(p) = 0.5 \tag{31}$$

where

$$p = \frac{\sqrt{2}\Delta s}{2c\sqrt{2s^2 + 2s\Delta s + \Delta s^2}} \tag{32}$$

The value of p that satisfies $\text{erf}(p) = 0.5$ may be found in a table of the erf function, $p = 0.4769361$.

Note that Equation (32) obeys Weber's Law since it is unchanged if we multiply both s and Δs by the same constant. Using c = 0.03 and $s = 1.0$, we get $\Delta s/s = 0.029$. This is comparable with the Weber fraction of $\Delta s/s = .05$ for human observers.

Equation (32) may also be used to quantify limits on the accuracy of velocity estimation for the human visual system. Using $\Delta s = .05$ and $s = 1.0$, and solving for c we get c = .051. This means that the standard deviation of the error in velocity estimation is 5.1% for human observers.

There are no cells in the human visual system that give a response proportional to speed. Similarly, none of the model's mechanisms give an output that is proportional to speed. Even so, error in the velocity estimates is proportional to speed for both the model and the human visual system. The model may thus provide some insight as to how the human visual system obeys Weber's Law for velocity discrmination.

4.2 Sine-grating Plaids

Figure 17(a) plots the psychometric function for coherence (probability of coherence) as the contrast of one of the component gratings is reduced. Figure 17(b) shows a plot of the ambiguity measure introduced in Section 3.3.3 as the relative contrast of the two component gratings is varied. In each case we may pick a threshold value (e.g., 50% probability, 0.001 ambiguity). Then we may vary the angle between the two component gratings or we may vary their relative spatial frequencies, and for each test case we measure the contrast that is needed to attain those threshold values.

In this way Adelson and Movshon [3] measured the threshold elevation of coherence for plaids made up of gratings with different spatial frequencies, plotted in Figure 18(a). As the frequencies of the two gratings were made different the tendency to cohere was reduced and the contrast needed for coherence was increased.

Figure 18(b) was generated by choosing a threshold value for the ambiguity measure; the plot shows the contrast elevation needed at each relative spatial frequency for the ambiguity measure to attain that value. Comparison of 18(a) and 18(b) indicates that the model's mechanisms are tuned to a somewhat narrower band of spatial frequencies than are the mechanisms of the human visual system. Figure 18(c) shows what happens if we increase the bandwidths of the model's filters by a factor of two. The plots in Figures 18(b) and 18(c) were generated using only one family (one spatial-frequency band) of filters ignoring interactions between spatial frequencies

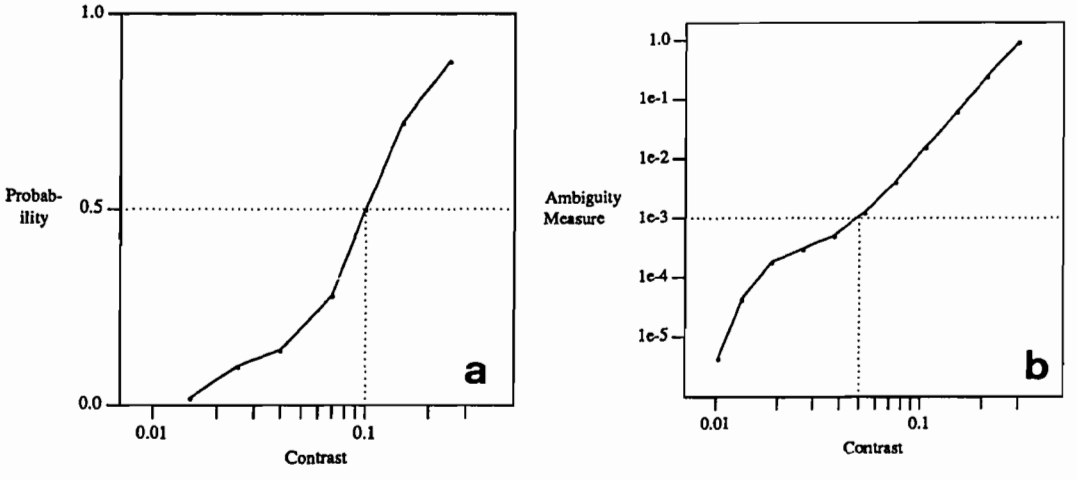


Figure 17: The influence of contrast on the coherence of sine-grating plaids. (a) (replotted from Adelson and Movshon [3]) One grating had a fixed contrast of 0.3 while the other was of variable contrast. The two gratings moved at an angle of 135°, both had a spatial frequency of 1.6 cycles deg^{-1}, and both moved at 3 deg s^{-1}. The plot shows the probablity that the observer judged the two gratings to be coherent. The dotted lines indicate the test-grating contrast needed to attain threshold (50% probability) coherence. Subject, EHA. (b) One grating had a fixed contrast of 0.3 while the other was of variable contrast. The two gratings moved at an angle of 120°, both had a spatial frequency of 0.25 cycles $pixel^{-1}$, and their speeds were chosen so that the coherent plaid moved at a speed of 2/3 pixels $frame^{-1}$. The plot shows the ambiguity measure as the contrast of the test grating was varied. The dotted lines indicate the test-grating contrast needed to attain threshold (0.001) ambiguity.

that almost certainly affect the psychophysical data.

Figure 19(a) shows the effect on coherence of varying the angular separation between the two gratings. As the angle was increased from 90° the tendency to cohere was reduced and the contrast needed for coherence was increased. The simulated data, plotted in Figure 19(b), is similar to that plotted in 19(a) up to an angle of 120°.

The plots in Figures 18 and 19 are promising. There are several parameters of the model that may be adjusted with the hope of matching the psychophysical data exactly: (1) The spatial bandwidths of the motion-energy filters — broader spatial bandwidth makes the plot in Figure 18(b) broader as shown in Figure 18(c); (2) The ratio of the temporal bandwidths to the spatial bandwidths — decreasing this ratio makes the plot in Figure 19(b) steeper; (3) The nature of the nonlinearities — for example, squaring accentuates the contrast difference more than absolute value and should tend to make the plot in Figure 18(b) narrower and the plot in Figure 19(b) steeper.

Different subjects were used to collect the data in Figures 18(a) and 19(a). Thus, the data in these two plots are inconsistent with one another requiring that different thresholds be used to generate Figures 18(b) and 19(b).

The eventual goal is to simulate all of the data for one subject with one choice parameters. We could measure the spatial and temporal bandwidths of the motion-energy channels psychophysically, leaving only the nonlinearities as free parameters.

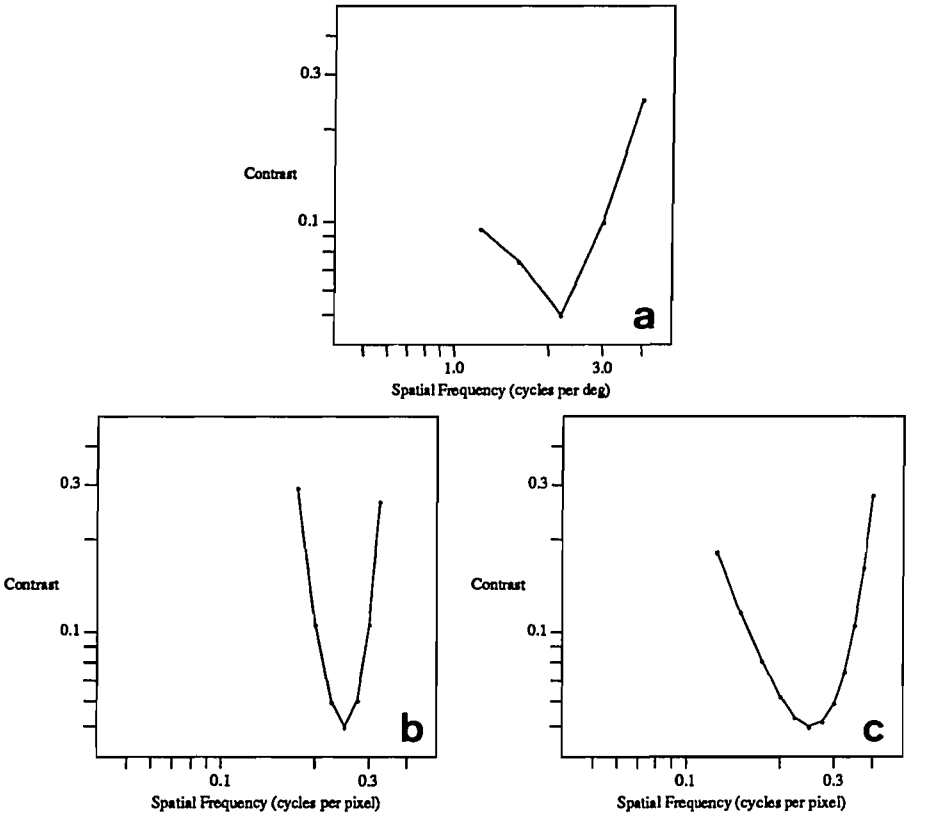


Figure 18: The influence of spatial frequency on the coherence of sine-grating plaids. (a) (replotted from Adelson and Movshon [3]) One grating had a fixed contrast of 0.3 while the other was of variable contrast. The two gratings moved at an angle of 135°, and both moved at 3 deg s^{-1}. The test grating was of variable contrast and variable spatial frequency. The plot shows the threshold contrast for coherence for a range of test spatial frequencies when the first grating was fixed at 2.2 cycles deg^{-1}. Subject, PA. (b) One grating had a fixed contrast of 0.3 and a fixed spatial frequency of 0.25 cycles $pixel^{-1}$ while the other was of variable contrast and spatial frequency. The two gratings moved at an angle of 90°, and their speeds were chosen so that the coherent plaid moved at a speed of 2/3 pixels $frame^{-1}$. The spread of the filter's Gaussian windows were $(\sigma_x, \sigma_y, \sigma_t) = (4.0, 4.0, 1.0)$. A fixed value was chosen as the threshold value for the ambiguity measure (this value was chosen in order to match the psychophysical data in (a) for the case when the fixed grating and test grating were of equal spatial frequency). For each test grating, the plot shows the contrast needed at that spatial frequency for the ambiguity measure to attain that value. (c) Same as (b) with the spread of the filter's Gaussian windows $(\sigma_x, \sigma_y, \sigma_t) = (2.0, 2.0, 0.5)$.

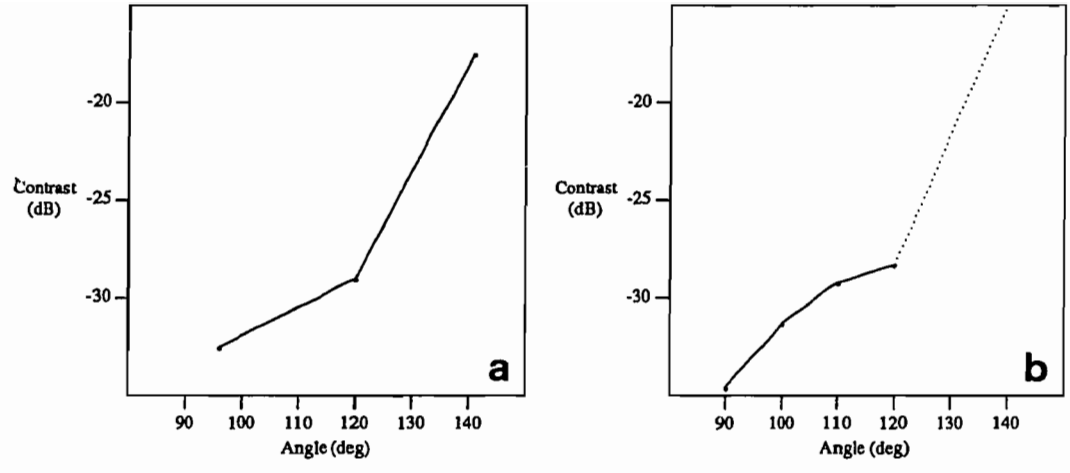


Figure 19: The influence of angle on the coherence of sine-grating plaids (a) One grating had a fixed contrast of 0.3 while the other was of variable contrast. The spatial frequency of one grating was fixed at 2.4 cycles deg^{-1} and that of the second grating was fixed at 1.2 cycles deg^{-1}. As the angle between the two gratings varied, their speeds were chosen so that the coherent plaid moved at a fixed speed of 7.5 deg s^{-1}. The plot shows the threshold contrast for coherence for a range of angles. Subject, EHA (b) One grating had a fixed contrast of 0.3 and both had a fixed spatial frequency of 0.25 cycles $pixel^{-1}$. The speed of the gratings was chosen so that the coherent plaid moved at a fixed speed of 2/3 pixels $frame^{-1}$. A fixed value was chosen as the threshold value for the ambiguity measure (this value was chosen in order to match the psychophysical data in (a) for 96°). For each angle, the plot shows the test-grating contrast needed for the ambiguity measure to attain that value. The dotted line indicates that no data was obtainable beyond 120°.

Chapter 5

Egomotion and the Stabilized World

This chapter describes research that was done in collaboration with Greg Hager.

Some of the goals of image motion interpretion are: (1) to estimate the observer's motion (egomotion); (2) to detect image regions that correspond to moving objects; (3) to recover the scene structure for the stationary background; (4) to estimate the 3-D motion of rigidly moving objects; (5) to recover the shape of rigidly moving objects; (6) to characterize the motion of nonrigidly moving objects. This chapter deals primarily with the second and third goals, detecting moving objects and recovering static scene structure. We also suggest a framework for using the resulting segmented flow field to update estimates of the egomotion parameters.

We approach these problems by assuming that we already have some information about the egomotion parameters. A number of authors have proposed methods for recovering the camera motion using visual information (see [18,135] for reviews of the literature). This has proven to be a difficult problem to solve in general, although the techniques show promise for recovering the direction of translation if the rotational component is already known. Information about camera motion may also be obtained from inertial sources, e.g., rate gyroscopes and accelerometers. The rotational component of motion is easily measured using a gyroscope, but the translational component is more difficult to estimate since it requires the integration of accelerations. There is a third source of information about the camera motion parameters for many robotics applications, as the motion of a camera mounted on a robot arm may be measured by differencing robot positions over time. We propose that reliable estimates of the camera motion will best be

obtained by combining information from visual and inertial/positional sensors.

Recovering scene structure from motion is greatly simplified if we have prior information about the camera motion. Bolles, Baker, and Marimont [24] present an effective technique for recovering scene structure when the observer is known to be moving through a static scene. They use a sequence of images sampled closely together in time and analyze slices through the space-time volume of luminance data. For straight-line camera motions, for example, moving feature points trace out lines in these slices, the slope of the lines being proportional to depth. The slices directly encode not only the three-dimensional positions of objects, but also such spatiotemporal events as the occlusion one object by another.

Heeger [61] also presents a technique for recovering depth using prior knowledge of the egomotion parameters. As discussed in Chapter 3, a distributed representation of image velocity can be computed by combining the outputs of a set of spatiotemporal motion-energy filters. For a fixed 3-D rigid-body motion depth values parameterize a line through image-velocity space (as discussed below in Section 5.1.1). Depth estimates are obtained by finding the peak in the distributed representation along this line. In this way, depth and image velocity are simultaneously extracted.

However, both of these techniques assume that the camera is moving through a stationary environment. In general we must first segment the images before we can estimate depth from egomotion. Segmentation of flow fields has previously been addressed by several authors [5,59,76,103,129,142,151,150]. A natural way to segment images based on motion information is to find image regions corresponding to entire objects that are moving differently from the stationary background.

Thompson and Pong [130] discuss the problem of detecting moving objects. If the observer is not moving and the illumination is constant, then motion detection is quite simple since there will be motion in the image if and only if an object is moving. However if the observer is also moving, then motion detection using visual information alone is quite difficult [130].

Motion detection, in general, requires that additional information about camera motion and/or scene structure be available. Prior information about the camera motion constrains the optical flow fields that can be generated by moving through an otherwise static environment. Prior information about scene structure places additional constraints. Violation of these constraints

are thus necessarily due to moving objects. The mechanism for segmentation described in this chapter may be classified as a *point-based* technique, that compares individual optical flow vectors against some standard to determine incompatibilities with the motion of the observer relative to the environment[1].

We can recover scene structure and we can often detect moving objects given prior information about the camera motion and about the image motion. However, information from perceptual sources (e.g., observations of image motion and camera motion) is inherently noisy and uncertain. Decisions and computations that rely on motion estimates will be more robust if we explicitly represent the uncertainty.

This chapter poses the detection of moving objects and the recovery of depth from motion as sensor fusion problems that necessitate combining information from different sensors in the presence of noise and uncertainty.

Section 5.1.1 reviews the geometry of rigid motion and then discusses how to distinguish between moving and stationary surfaces in a noiseless environment. Section 5.2 characterizes the uncertainty in the observations from the image flow and egomotion sensors. Section 5.3 presents the technique for integrating the information from these two sources to detect moving objects and recover scene structure, and suggests a framework for using the resulting segmented flow field to update estimates of the egomotion parameters. Finally, we show some example results.

5.1 Egomotion

This section reviews the geometry of rigid motion and then discusses how to distinguish between moving and stationary surfaces in a noiseless environment.

[1]Thompson and Pong [130] describe two other classes of methods for detecting moving objects. *Edge-based* techniques correspond to traditional edge detection looking for discontinuities in the flow field, and *region-based* methods examine a region of the flow field testing for whether or not the flow vectors are compatible with a rigid-body motion interpretation.

5.1.1 Geometry of Rigid Motion

First we review the equations relating rigid-body motion in 3-space to image motion under perspective projection. Then we show that for a fixed 3-D rigid-body motion, depth values at each image location parameterize a line through image-velocity space.

The equations relating rigid motion to image motion have been derived in several forms by a number authors (see [18,135] for reviews of the literature); the derivation presented below most closely follows that of Longuet-Higgins and Prazdny [88], Bruss and Horn [26], and Waxman and Ullman [145].

Each point on a patch of a rigid surface has an associated position vector relative to the viewer-centered coordinate frame depicted in Figure 20,

$$\vec{R} = [X, Y, Z(X,Y)]^T \tag{33}$$

Every point of a rigidly moving object shares the same six motion parameters relative to that coordinate frame,

$$\begin{aligned} \vec{\Omega} &= (\Omega_x, \Omega_y, \Omega_z)^T \\ \vec{T} &= (T_x, T_y, T_z)^T \end{aligned}$$

where $\vec{\Omega}$ is the rotational component (angular velocity) and $\vec{T}$ is the translational component (linear velocity) of the motion.

Equivalently, we may treat the object as stationary and let $\vec{D} = (T_x, T_y, T_z, \Omega_x, \Omega_y, \Omega_z)^T$ indicate the joint (linear and rotational) motion of the camera. Due to the motion of the camera, the relative motion of a point on a stationary surface is

$$\vec{V} = (V_x, V_y, V_z)^T = -\frac{d\vec{R}}{dt} \tag{34}$$

which is related to the motion parameters by

$$\vec{V} = -\left(\vec{\Omega} \times \vec{R} + \vec{T}\right) \tag{35}$$

Now we derive an equation for image velocity $\vec{\theta} = (u, v)^T$ as a function of the rigid-body motion parameters. Under perspective projection a point, $(X,Y,Z)^T$, in space projects to the

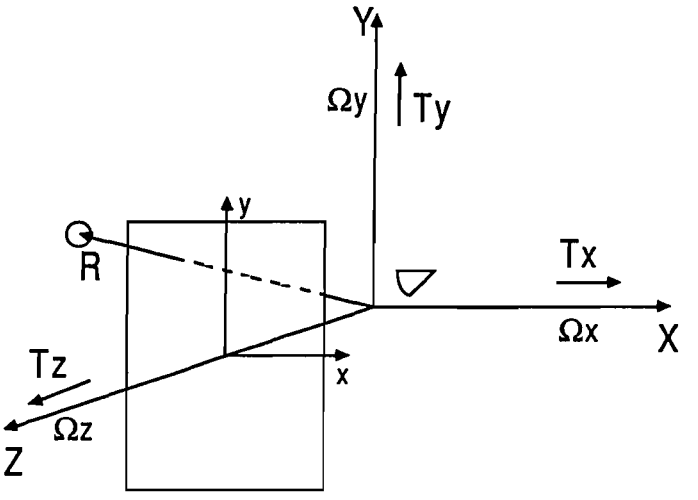


Figure 20: Viewer-centered coordinate frame, perspective projection, and rigid-body motion parameters.

image point, $(x, y)^T$,

$$\begin{aligned} x &= fX/Z \\ y &= fY/Z \end{aligned} \tag{36}$$

where f is the focal length. Taking derivatives with respect to time and substituting from Equation (35) to eliminate occurences of X and Y gives:

$$\theta = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} f\left(\frac{V_x}{Z} - \frac{xV_z}{fZ}\right) \\ f\left(\frac{V_y}{Z} - \frac{yV_z}{fZ}\right) \end{bmatrix} = \mathbf{A}(Z)\vec{V} \tag{37}$$

where

$$\mathbf{A}(Z) = \frac{1}{Z}\begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \tag{38}$$

From Equation (35) we know that

$$\vec{V} = \begin{bmatrix} -\Omega_y Z + \Omega_z Z(y/f) - T_x \\ -\Omega_z Z(x/f) + \Omega_x Z - T_y \\ -\Omega_x Z(y/f) + \Omega_y Z(x/f) - T_z \end{bmatrix} = \begin{bmatrix} -\Omega_y Z + \Omega_z Y - T_x \\ -\Omega_z X + \Omega_x Z - T_y \\ -\Omega_x Y + \Omega_y X - T_z \end{bmatrix} = \mathbf{B}\vec{D} \tag{39}$$

where

$$\mathbf{B} = \begin{bmatrix} 0 & -Z & Y & 1 & 0 & 0 \\ Z & 0 & -X & 0 & 1 & 0 \\ -Y & X & 0 & 0 & 0 & 1 \end{bmatrix} \tag{40}$$

Substituting Equation (39) into Equation (37) gives

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (xT_z - fT_x)/Z + (xy/f)\Omega_x - (f + x^2/f)\Omega_y + y\Omega_z \\ (yT_z - fT_y)/Z + (f + y^2/f)\Omega_x - (xy/f)\Omega_y + x\Omega_z \end{bmatrix} = \mathbf{A}(Z)\,\mathbf{B}\,\vec{D} \tag{41}$$

For fixed $\vec{\Omega}$, $\vec{T}$, x, and y Equation (41) is the parametric form of the equation of a line — changing Z corresponds to sliding along this line. This can be seen most easily by rewritting it as

$$\begin{aligned} u &= pa_1 + b_1 \\ v &= pa_2 + b_2 \end{aligned} \tag{42}$$

where

$$\begin{aligned} p &= 1/Z \\ a_1 &= xT_z - fT_x \\ a_2 &= yT_z - fT_y \\ b_1 &= (xy/f)\Omega_x - (f + x^2/f)\Omega_y + y\Omega_z \\ b_2 &= (f + y^2/f)\Omega_x - (xy/f)\Omega_y + x\Omega_z \end{aligned}$$

Thus, p parameterizes a line through image-velocity space, and each point along that line corresponds to a different depth. In the sequel, let $\mathbf{C}(p) = \mathbf{A}(1/p)\mathbf{B}$. For fixed p, $\mathbf{C}(p)$ represents the linear transformation from motion in three dimensions to image flow.

Actually, p parameterizes a line segment in image velocity space since surface points that are being viewed by the camera are always in front of the camera, i.e., $Z \in (f, \infty)$ and $p \in (0, 1/f)$. For fixed (x, y), f varies the length of the line segment, $\vec{T}$ varies the slope of line segment, and $\vec{\Omega}$ slides the segment around in image velocity space. Mulitplying each component of $\vec{T}$ by the same constant does not change the segment at all.

5.1.2 Egomotion in a Noiseless Environment

Prior information about the camera motion constrains the optical flow fields that can be generated by moving through an otherwise static environment. The image velocity at an image point is constrained to be along a line segment in image velocity space given by Equation (42) with $p \in (0, 1/f)$. In addition, prior information on depth (from time past and from other sources such as stereo) places additional constraints by specifying a smaller interval for the domain of p.

Image velocity for a given image location corresponds to a point in image-velocity space. If the point lies somewhere along the line segment, then the camera motion information and the image motion information are consistent with one another. This means that the image motion may be accounted for by the camera motion alone, i.e., that the image region may correspond to a surface patch that is stationary in the environment. In this case, we may obtain an estimate of the relative depth to that surface patch using Equation (42). On the other hand, if the point does not lie along the line segment, then the image motion must be due to a moving object.

Detecting moving objects is analogous to the fly-detector model presented in Chapter 1. Information about the relative depth of a surface point is *overconstrained* by observations from the two sensors. By random chance, it is unlikely (probability zero) that the two observations will be consistent for a given image region. In spite of this, we should oftentimes expect the two observations to be consistent because stationary surfaces are extremely prevalent in our environment. Thus, when the two sensor observations are consistent we may reliably infer that the image region corresponds to a stationary surface patch.

We will never mistakenly infer that a stationary surface is moving. There are, however, situations in which we might mistakenly infer that a moving surface is stationary. If the camera is translating to the right, for example, a distant surface patch moving to the left will have the same image motion as a nearby stationary surface patch. Prior information on Z (from time past and from other sensors like stereo) will help by adding more constraint, specifying a smaller interval for the constraint line segment.

The test for detecting moving objects does not depend on knowing the exact camera translation. If we multiply each component of $\vec{T}$ by the same constant, then the constraint line segment

is unchanged. Thus, we can still detect inconsistencies and recover a relative depth map given only the direction of the camera's translation.

5.2 Sensor Models

In the real world and using real sensors we must contend uncertainty in sensor data. In order to combine noisy information from different sources, each sensor must provide us both with observations and with some measure of the uncertainty in those observations. A sensor model [36] is a description of a sensor's ability to observe the environment. This is in general a function of the state of the environment, the state of the sensor itself, and the state of other sensors or cues in a multi-sensor system. A static sensor model or observation model describes the dependence of an observation on the state of the environment. In this section we adopt probabilistic sensor models to characterize the uncertainty in the egomotion and image flow observations.

5.2.1 Sensor Modeling

We will consider a measurement device subject to additive noise as a mathematical system of the following general form:

$$\tilde{\theta} = H(\vec{D}, p) + n \tag{43}$$

where H is a k-dimensional function of the true state of the environment, $(\vec{D}, p)$. Our observation, $\tilde{\theta}$, is a function of $(\vec{D}, p)$ and is contaminated by additive noise n. If the probability distribution of n is $n \sim f(\cdot)$, then the *likelihood* function based on the observed data is of the form $f(\tilde{\theta}|H(\vec{D}, p))$.

We assume that the observations of camera motion and of image velocity are contaminated by zero-mean additive Gaussian noise. This is an approximation that must be verified for particular sensors. However, if normal distributions were *not* valid approximations for the observation errors, then the procedures presented below would not be reliable. In spite of this, some insight can be gained by proceeding under the normality assumption. In future research, I hope to extend the analysis presented below to allow for more general assumptions about the error distributions.

Let $\tilde{\theta}$ be a random variable with associated probability density taking the general form a Gaussian. If $\vec{\theta}$ is the mean of that distribution and Λ_θ is its variance-covariance matrix, we write $\tilde{\theta} \sim N(\vec{\theta}, \Lambda_\theta)$, and we write the associated probability density as:

$$f(\tilde{\theta}) = \frac{1}{(2\pi)^{k/2}\,|\Lambda_\theta|^{1/2}} \exp\left(-\frac{1}{2}(\tilde{\theta}-\vec{\theta})^T \Lambda_\theta^{-1} (\tilde{\theta}-\vec{\theta})\right) \tag{44}$$

As discussed in Section 3.2, the noise in the observation, $\tilde{\theta}$, may also be represented either by the variance-covariance matrix, Λ_θ, or by the Fisher information matrix, Λ_θ^{-1}. We choose to represent uncertainty with the information matrix instead of using the variance-covariance matrix because the latter may be undefined when there is only partial information.

5.2.2 The Image Flow Sensor

Chapter 3 presents a model for the extraction of image flow. In addition to estimating image velocity, the model provides a measure of the uncertainty in the estimates. It is demonstrated that this uncertainty measure reflects the actual error in the velocity estimates, for translating image sequences of Gaussian white-noise random textures.

The model approximates image-flow uncertainty as jointly normal, and computes the variance-covariance matrix. Equation (42) expresses image velocity, $\vec{\theta} = (u, v)^T$ as a function of the six motion parameters, $\vec{\theta} = \mathbf{C}(p)\vec{D}$, where $\mathbf{C}(p)$ is the linear transform defined in Section 5.1.1. Since the conditional probability density of $\vec{\theta}$ is normal we write the observation of image velocity provided by the image flow sensor as:

$$\tilde{\theta} = \mathbf{C}(p)\vec{D} + n_\theta \quad n_\theta \sim N(0, \Lambda_\theta) \tag{45}$$

5.2.3 The Egomotion Sensor

We take the error in the estimates of the egomotion parameters to be modeled by spatially and temporally independent, zero-mean, Gaussian processes.

$$\tilde{D} = \vec{D} + n_D \quad n_D \sim N(0, \Lambda_D) \tag{46}$$

If the rate information is obtained, for instance, by differencing robot positions over time and if the robot postioning errors are constant variance Gaussian processes, then Λ_D is obtained

from the positioning accuracy of the robot. We assume that the robot positioning errors are independent so that Λ_D is be invertible.

5.3 Combining the Sensor Information

In this section, we focus on devising a statistical test for deciding whether or not the data from the two sensors are consistent with one another. For the present, we make no prior statistical assumptions about the parameters we are estimating, and use maximum likelihood techniques to combine information. Also, we suggest a framework for using the resulting segmented flow field to update estimates of the egomotion parameters. In future research, we hope to implement an incremental scheme for updating estimates of the motion parameters by combining prior information with new sensor observations.

Figure 21 illustrates how to detect moving objects in the presence of noise and uncertainty. If the two distributions provided by each of the sensors are consistent, then we may combine them to get an estimate of depth and improved estimate of image velocity.

5.3.1 Combining Information

We have two sensors, one providing noisy observations of D and the other noisy observations of $\vec{\theta}$. If the observations are independent, then their joint likelihood is simply the product of the individual likelihood functions. In this case, the maximum likelihood estimate (MLE), $\hat{D}$, is the value which maximizes the joint likelihood function. Equivalently, $\hat{D}$ minimizes the negated log-likelihood function:

$$l(D) = \log[f(\vec{D}, p)] = \left[\frac{1}{2}(\tilde{D} - \vec{D})^T \Lambda_D^{-1} (\tilde{D} - \vec{D}) + \frac{1}{2}(\tilde{\theta} - \mathbf{C}(p)\vec{D})^T \Lambda_\theta^{-1} (\tilde{\theta} - \mathbf{C}(p)\vec{D})\right] \tag{47}$$

For fixed p this minimum may be found either by differentiation or by standard algebraic techniques [125], and is given by

$$\hat{D}(p) = \left[\Lambda_D^{-1} + \mathbf{C}(p)^T \Lambda_\theta^{-1} \mathbf{C}(p)\right]^{-1} \left[\Lambda_D^{-1} \tilde{D} + \mathbf{C}(p)^T \Lambda_\theta^{-1} \tilde{\theta}\right] \tag{48}$$

provided the first quantity is invertible. In our case, Λ_D^{-1} is taken to be full rank so the inverse operation is well defined.

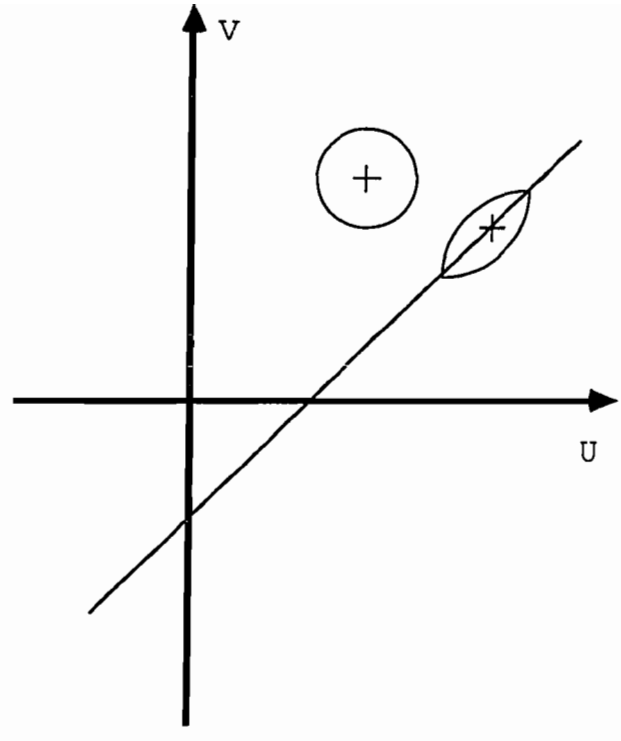


Figure 21: Detecting moving objects in the presence of uncertainty. The ellipse along the straight line in image velocity space represents the distribution of the observation provided by the egomotion sensor. The circle represents the distribution of the observation provided by the image flow sensor. If the two distributions are consistent, then we may combine them to get an estimate of depth and improved estimate of image velocity. If they are not consistent, then the image motion must be due to a moving object.

Substituting Equation (48) for $\vec{D}$ in Equation (47) and simplifying gives:

$$M(p;\tilde{D},\tilde{\theta},\Lambda_D^{-1},\Lambda_\theta^{-1}) = (1/2)\left[\tilde{D}^T\,\Lambda_D^{-1}\,\tilde{D} + \tilde{\theta}^T\,\Lambda_\theta^{-1}\,\tilde{\theta} - \hat{D}(p)^T\,\left(\Lambda_D^{-1}\tilde{D} + \mathbf{C}(p)^T\,\Lambda_\theta^{-1}\,\tilde{\theta}\right)\right] \tag{49}$$

The final MLE, $\hat{p}$, is obtained by numerically minimizing Equation (49) over $p \in (0, 1/f)$. Note that this operation is conservative in the sense that it chooses the depth which is most compatible with the two observations.

Several authors have proposed incremental schemes for estimating the three-dimensional motion parameters [25,38,56,122,134]. Given a prior distribution on the motion parameters, an updating scheme (e.g., an Extended Kalman filter) can be used to combine information across the flow field and over time [47]. An expression similar to Equation (48) tells us (for an image region corresponding to stationary surface) how to update estimates of the egomotion parameters in the presence of new image motion data.

Of course, the three-dimensional motion is underdetermined from a single flow observation. Geometrically, the combination rule given by Equation (48) can be viewed as reducing uncertainty in $\vec{D}$ in a two-dimensional subspace. Even using the information in the entire flow field may underdetermine the motion parameters. Several authors (for example, see [14,126]) discuss the inherent ambiguity in recovering motion parameters from optical flow.

5.3.2 Consistency of Information

Maximum likelihood is one way of combining information from two sensors. But what if one of the sensors is giving spurious data (perhaps it is broken) or is observing a process different from its counterpart. In such a case, we do not want to combine the information into a single estimate. We want to combine information from different sensors only if they *concur* with one another.

The quantity denoted by $M(p;\tilde{D},\tilde{\theta},\Lambda_D^{-1},\Lambda_\theta^{-1})$ in Equation (49) is referred to as a Mahalanobis distance, and is commonly used as a threshold rule for determining consistency between observations. The larger the disparity between two observations, the larger the resulting Mahalanobis distance. If the minimum Mahalanobis distance between the two observations is less than a given threshold then the two observations are taken to be consistent with one another and

we may combine the information from the two sensors to calculate a single best estimate for $\vec{\theta}$.

A crucial question is the choice of thresholding criteria. For the result in Section 5.3.3, a threshold was chosen interactively. In future research we hope to formulate a method for choosing the threshold automatically.

Mahalanobis distance, however, is essentially heuristic — in general it has no decision-theoretic basis. For example, if the noise distributions are non-symmetric, it will lead to biased results. In future research we will investigate alternative statistical hypothesis tests.

5.3.3 Example Results

Figure 22 shows the segmentation for the Yosemite fly-through sequence (Figure 9 in Chapter 3) achieved by thresholding Mahalanobis distance, Equation (49). The segmentation correctly classifies 91.0% of the sky points and 96.9% of the ground points. All of the classification errors are either at the horizon or near the edges of the image (due to the edge effects of the convolutions used to extract image flow).

A depth map was simultaneously recovered from the image sequence, and was converted to an altitude map by using the known camera position and orientation to transform the depth values to a viewer-independent coordinate frame. A histogram of the percent error between the actual and recovered altitude maps is shown in Figure 23(a). The average percent error is 3.6% and the standard deviation is 8.7%.

Figures 23(b) and 23(c) demonstrate that the recovered depth data is accurate enough to locate qualitative features of the landscape, e.g., the valley. For many perceptual tasks (for example, navigation), qualitative information (e.g., near versus far, high versus low, where is the valley, where is the moving object) is often sufficient. If necessary, more accurate depth information may be obtained utilizing image data over longer periods of time as demonstrated by Bolles, Baker, and Marimont [24].

5.4 Summary

Previous research [24,61,130] has demonstrated that we can recover scene structure and we can often detect moving objects given prior information about the camera motion and about

Figure 22: Segmentation into stationary and moving regions. Pixels near the edge of the image can not be classified due to the edge effects of the convolutions used to extract image flow. The segmentation correctly classifies 91.0% of the sky points and 96.9% of the ground points.

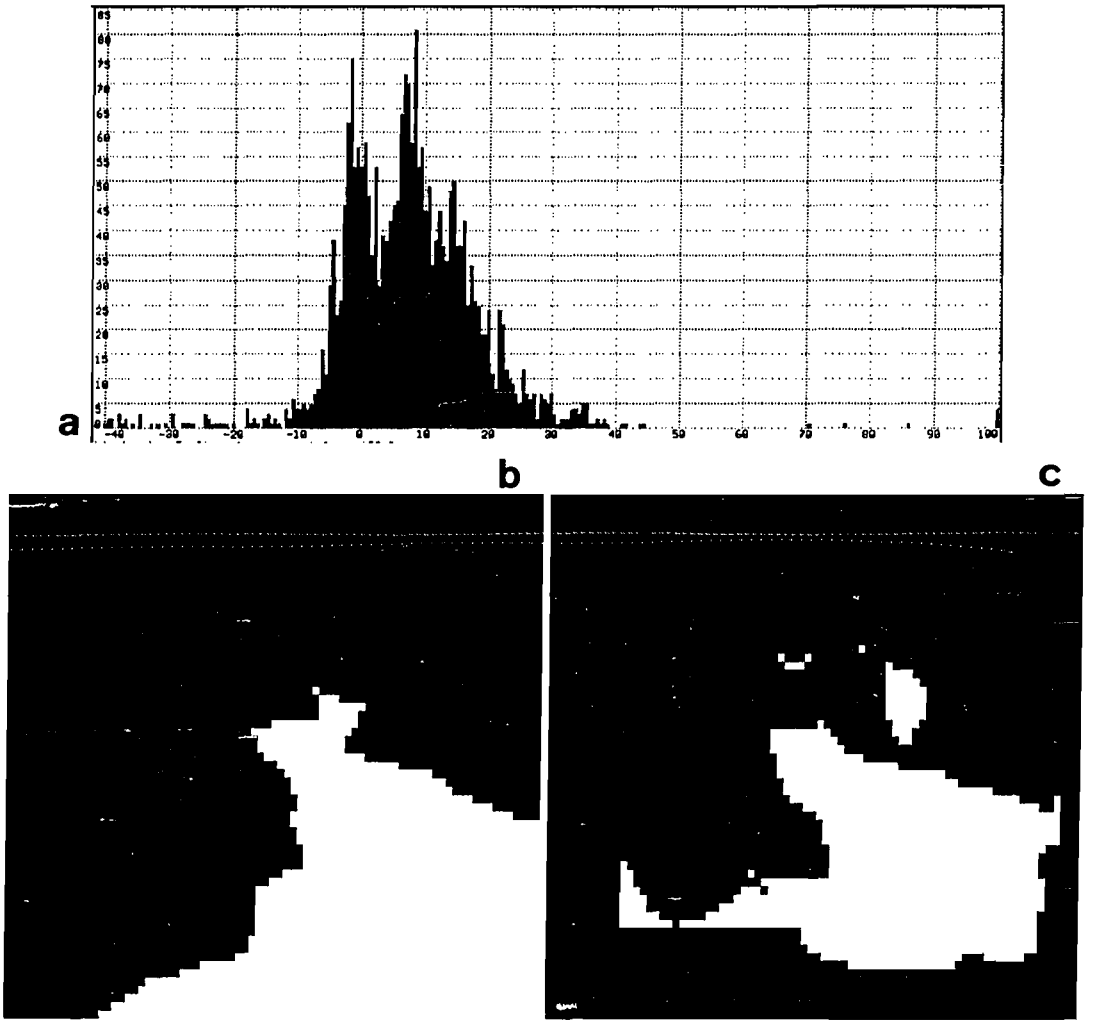


Figure 23: (a) Histogram of percent error between actual and recovered altitude maps. (b) The actual altitude map was thresholded at 1400 meters above sea level showing the valley in white. (c) Recovered valley. No altitude values may be computed near the edge of the image due to the edge effects of the convolutions used to extract image flow.

the image motion. However, information from perceptual sources (e.g., observations of image motion and camera motion) is inherently noisy and uncertain. This chapter poses the detection of moving objects and the recovery of depth from motion as sensor fusion problems that necessitate combining information from different sensors in the presence of noise and uncertainty.

Prior information about the camera motion constrains the optical flow fields that can be generated by moving through an otherwise static environment. Given the motion parameters, the image velocity at an image point is constrained to be along a line segment in image velocity space. In addition, prior information on depth (from time past and from other sources such as stereo) places additional constraints by specifying a smaller interval for the line segment. Image velocity for a given image location corresponds to a point in image-velocity space. We utilize Mahalanobis distance as a threshold rule for determining consistency between these two sensor observations.

If the two observations are consistent, then the image motion may be accounted for by the camera motion alone, i.e., the image region may correspond to a surface patch that is stationary in the environment. In this case, we may obtain an estimate of the relative depth to that surface. On the other hand, if the observations are not consistent, then the image motion must be due to a moving object.

If the observations are consistent, then an expression similar to Equation (48) tells us how to update estimates of the egomotion paraters in the presence of the new image motion data. This suggests an incremental scheme for recovering the motion parameters, perhaps utilizing the extended Kalman filter [47].

The procedure outlined in this chapter for the analysis of egomotion and stationary surfaces may be generalized to arbitrary motions of rigid objects. Given prior information about the rigid-body motion parameters for some small patch of the surface of a moving object, we may utilize image flow information to extract entire regions that are consistent with those motion parameters, recover depth, and update the estimates of the 3-D motion.

Chapter 6

Rigid Body Motion

The previous chapter discussed how to segment images and recover depth from motion information given prior knowledge of the three-dimensional rigid-body motion parameters. For the case of egomotion and the stationary background estimates of these motion parameters may come, in part, from inertial/postional sensors. We must, however, rely only on visual information to recover the motion parameters of moving objects.

Early perceptual studies suggested that the rigidity of objects may play a key role in the perception of motion [52,55,60,78,139]. A number of authors have proposed methods for recovering the three-dimensional rigid-body motion parameters, either from feature motions [16,25,30,38,66,75,86,113,117,118,122,123,131,132,134,136,147], from instantaneous flow fields [5,14,19,26,56,65,81,82,88,102,126,145,146,151], or directly from the time-varying image intensity [71,72,104]. A number of these papers are summarized and reviewed in [18,135]

However, no-one has yet produced a reliable method of recovering the motion parameters. In part, this is because the equations relating three-dimensional motion to image motion are nonlinear and the parameter space is six-dimensional.

In addition, relying only on the information from instantaneous flow fields (or displacement fields) confounds the problem. It is difficult to distinguish instantaneously between translation in X and rotation about Y (similarly translation in Y and rotation about X). The standard approach for separating T_x from Ω_y is to utilize the second spatial derivatives (or second differences across space) of the flow (or displacement) field. Derivative information is usually calculated directly

from velocity fields, for example, by fitting second-order flow fields to normal-flow estimates [146]. Such second derivatives are numerically unstable in the presence of noise.

Barron et al [19] perform an error analysis of a class of algorithms for estimating the motion parameters. They find that algorithms using individual velocity vectors require accuracy to within 1%. Equivalently, they find that algorithms using local image velocity information (first and second derivatives of velocity within a small neighborhood) require accuracy to within 10%. Current techniques for estimating image velocity cannot produce the required accuracy. As discussed in Section 4.1, even the human visual system does not seem to achieve the required accuracy.

The active vision/sensor fusion paradigm suggests that if a problem seems difficult to solve, then take more data. For example, some authors have suggested using prior information about depth (e.g., from stereo) to help recover the motion parameters [13,77,101,103,117,120,122,143,144] This Chapter proposes two additional sources of information: (1) measuring the spatio-temporal derivative information directly from the time-varying intensity data; (2) recording eye/camera position and eye/camera motion over time while tracking a surface point on a rigidly moving object.

6.1 Deformation Filters

Several authors suggest using deformation fields [81,82,88,142,145,150], the spatial derivatives of image velocity, in order to recover the motion parameters. In practice, these deformation fields have been computed by first estimating image velocity and then either: (1) taking differences between flow vectors; or (2) fitting second order flow fields to the flow vectors.

Some researchers suggest that we might rather estimate the deformations directly from the time-varying imagery [19,81,88]. In fact, there is psychophysical evidence that the human visual system has separate mechanisms for recovering image curl and divergence [114,115,116].

Chapter 3 discusses a technique for combining the outputs of a set of energy filters, each sensitive to image translations in a different direction, in order to estimate the image velocity for a local image region. Analogously, it may be possible to build linear filters sensitive to clockwise and counterclockwise rotations, and combine their outputs to estimate image curl.

We might similarly build mechanisms tuned for image divergence and image shear.

6.2 Tracking

Eye/camera movements have only two rotational degrees of freedom, pan and tilt, about axes that pass through the center of projection. Each camera orientation is associated with a different rotated coordinate frame. Rotation matrices specify the transform from one coordinate frame to another. In Appendix E I derive formulas for transforming position, velocity and angular velocity from one frame to another. I explain how to to fixate on a surface point, how to use image velocity information to track moving surface points over time, and how to warp an image to simulate the effect of an eye/camera movement.

Bandopadhay and Aloimonos [8,15] suggest that tracking the motion of a surface point simplifies the problem of recovering the motion parameters. This section proposes that both eye/camera position and eye/camera motion can be used as additional constraints to help recover the motion parameters.

First, records of eye/camera position while tracking a surface point allow us to transform the image motion observations (velocities and deformations) collected over time into a single coordinate frame where they may be combined.

Second, the angular velocity of a tracking eye movement over time is related to the fixed motion parameters of the rigidly moving object. For simplicity, I have so far considered only the two-dimensional case of rigid motion in a plane. We choose a coordinate system oriented with the camera's starting position as depicted in Figure (24). In this coordinate frame the motion of a point on the object's surface is given by

$$\begin{aligned} V_x(t) &= \Omega_y Z(t) + T_x \\ V_z(t) &= -\Omega_y X(t) + T_z \end{aligned} \tag{50}$$

where (T_x, T_y) is the translational component of the motion, Ω_y is the rotational component, $[V_x(t), V_z(t)]$ is the velocity of the point at each time, and $\vec{R}(t) = [X(t), Z(t)]$ is its position.

Equation (50) is a differential equation that relates position over time to velocity (the derivative of position with respect to time). It may be solved using Laplace transforms; the Laplace

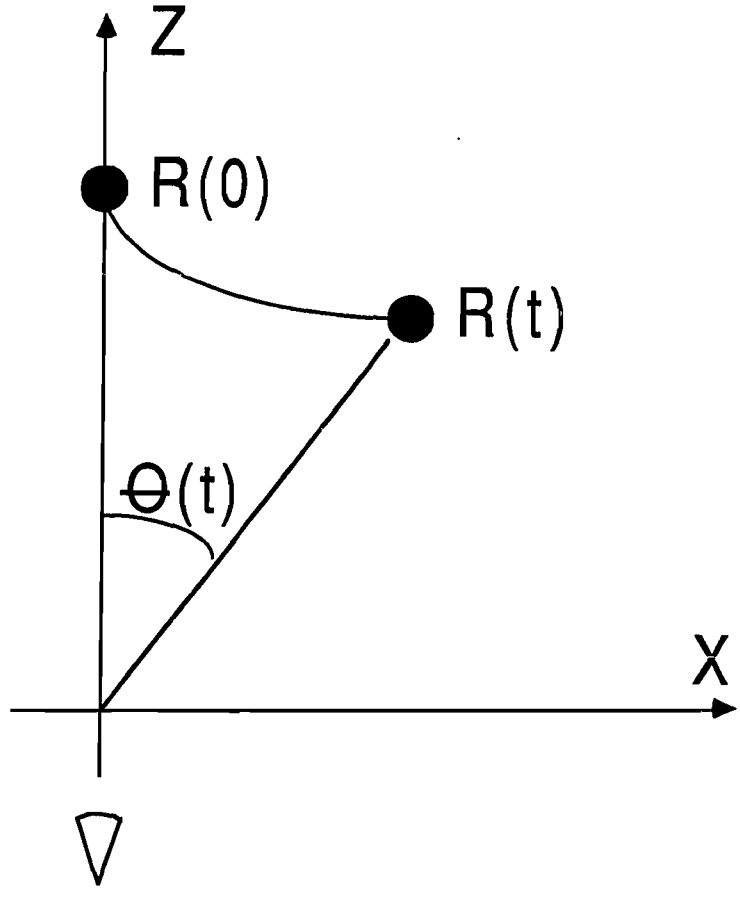


Figure 24: Tracking the motion of of a point on the surface of a rigidly moving object.

transform of Equation (50) is

$$\begin{aligned} sX(s) - X_0 &= \Omega_y Z(s) + T_x/s \\ sZ(s) - Z_0 &= -\Omega_y X(s) + T_x/s \end{aligned} \tag{51}$$

where X_0 and Z_0 are the intial conditions, the position of the point at $t = 0$. Solving this linear system of equations gives

$$\begin{aligned} Z(s) &= \frac{T_z - \Omega_y X_0 + sZ_0}{s^2 + \Omega_y^2} - \frac{\Omega_y T_x}{s(s^2 + \Omega^2)} \\ X(s) &= \frac{T_x}{s^2} + \frac{X_0}{s} + \frac{\Omega_y T_z - \Omega_y^2 X_0}{s(s^2 + \Omega_y^2)} - \frac{\Omega_y^2 T_x}{s^2(s^2 + \Omega_y^2)} + \frac{\Omega_y Z_0}{s^2 + \Omega_y^2} \end{aligned} \tag{52}$$

Taking inverse Laplace transforms and simplifying gives

$$\begin{aligned} X(t) &= \frac{T_x}{\Omega_y}\sin(\Omega_y t) + \frac{T_z}{\Omega_y}[1 - \cos(\Omega_y t)] + X_0\cos(\Omega_y t) + Z_0\sin(\Omega_y t) \\ Z(t) &= \frac{T_z}{\Omega_y}\sin(\Omega_y t) - \frac{T_x}{\Omega_y}[1 - \cos(\Omega_y t)] + Z_0\cos(\Omega_y t) - X_0\sin(\Omega_y t) \end{aligned} \tag{53}$$

As depicted in Figure (24) the camera position at each time, $\theta(t)$, is given by

$$\theta(t) = \tan^{-1}\left[\frac{X(t)}{Z(t)}\right] \tag{54}$$

Using Equations (50) and (54) the angular velocity of the camera at each time, $\Omega_c(t)$, is given by

$$\Omega_c(t) = \frac{d\theta(t)}{dt} = \frac{z^2(t)\Omega_y + X^2(t)\Omega_y + Z(t)T_x - X(t)T_z}{X^2(t) + Z^2(t)} \tag{55}$$

Taking $X_0 = 0$ (i.e., the camera is fixating on the point at $t = 0$) and substituting for $X(t)$ and $Z(t)$ from Equation (53) gives

$$\Omega_c(t;\Omega_y,T_x,T_z,Z_0) = \frac{N(t;\Omega_y,T_x,T_z,Z_0)}{D(t;\Omega_y,T_x,T_z,Z_0)} \tag{56}$$

$$\begin{aligned} N(t;\Omega_y,T_x,T_z,Z_0) &= \Omega_y^3 + \left(\frac{T_z^2}{Z_0^2} + \frac{T_x^2}{Z_0^2}\right)\Omega_y(1 - \cos(\Omega_y t)) \\ &\quad + \Omega_y^2\left(\frac{T_z}{Z_0}\sin(\Omega_y t) - \frac{T_x}{Z_0}\cos(\Omega_y t) + 2\frac{T_x}{Z_0}\right) \\ D(t;\Omega_y,T_x,T_z,Z_0) &= \Omega_y^2 + 2\left(\frac{T_z^2}{Z_0^2} + \frac{T_x^2}{Z_0^2}\right)(1 - \cos(\Omega_y t)) \\ &\quad + 2\Omega_y\left(\frac{T_z}{Z_0}\sin(\Omega_y t) - \frac{T_x}{Z_0}\cos(\Omega_y t) + \frac{T_x}{Z_0}\right) \end{aligned}$$

Equation (56) relates the angular velocity of the tracking camera movement to the motion parameters of a point on the surface of a rigidly moving object. Since Equation (56) is nonlinear, it is not apparent whether it is solvable in general, and if so whether the solution is unique. Some interesting special cases of this equation are listed below:

1. For $t = 0$,

$$\Omega_c(0; \Omega_y, T_x, T_z, Z_0) = \frac{T_x}{Z_0} + \Omega_y \tag{57}$$

2. For $T_x = T_z = 0$,

$$\Omega_c(t; \Omega_y, 0, 0, Z_0) = \Omega_y \tag{58}$$

3. For $\Omega_y = 0$,

$$\lim_{\Omega_y \to 0} \Omega_c(t; \Omega_y, T_x, T_z, Z_0) = \frac{T_x Z_0}{Z_0^2 + 2tT_z Z_0 + t^2 T_z^2 + t^2 T_x^2} \tag{59}$$

4. For $T_z = \Omega_y = 0$,

$$\lim_{\Omega_y \to 0} \Omega_c(t; \Omega_y, 0, T_z, Z_0) = \frac{T_x Z_0}{Z_0^2 + t^2 T_x^2} \tag{60}$$

5. For $T_x = \Omega_y = 0$,

$$\lim_{\Omega_y \to 0} \Omega_c(t; \Omega_y, T_x, 0, Z_0) = 0 \tag{61}$$

As discussed above, it is difficult to distinguish instantaneously between T_x and Ω_y. The eye/camera motion information should help solve this problem. Equation (58), for example, tells us that if both T_x and T_z are zero, then the camera motion is equal to Ω_y.

6.3 Summary

The previous chapter proposes a procedure for using image motion information to segment images, recover depth, and update estimates of the motion parameters. This chapter proposes two additional sources of information for solving these problems; deformation fields extracted directly from the time-varying imagery, and records of both eye/camera motion and eye/camera position while tracking a surface point on a rigid object.

It seems to me that a reliable method for recovering the motion parameters should obey the following three general principles: (1) the motion parameters should be computed using a small

image region that corresponds to a single moving surface; (2) the method should utilize as much data as possible in order to have sufficient overconstraint for reliable estimates; (3) the method should provide an explicit test for rigidity by checking for consistency amoungst the data. In order to simultaneously satisfy the first two principles the method should collect data over time while tracking the motion of a small patch of surface. This suggests an incremental scheme (perhaps using the extended Kalman filter [47]) that tracks the motion of a small surface patch, utilizing the incoming image motion and eye/camera movement data to update estimates of the motion parameters.

Chapter 7

Turbulent Flow

This chapter describes research that was done in collaboration with Alex Pentland.

Turbulent motion is quite prevalent in the natural outdoors world. Some examples are clouds, waterfalls, waves, boiling water, the leaves of trees or bushes rustling in the wind, grass or wheat fields blowing in the wind, flags fluttering in the wind, fire, and smoke. Despite the conventional attitude that such motion is purely random, people are able to gather considerable information from it — e.g., average velocity, viscosity, or quantity of flow — just by visual observation.

Imagine, for instance, that you are standing on a bridge on a quiet, windless day. Upstream, the water looks like a flat motionless surface. But downstream, there are turbulent wakes behind the bridge supports. The surface of the water is rough in the turbulent region, i.e., the orientation of the water's surface normal varies wildly over space and time. A human observer does not see any motion upstream of the bridge because the surface is perfectly smooth. But he interprets the turbulent region downstream of the bridge as motion and he sees, immediately and unconsciously, the direction and speed in which the flow moves. Similarly, a waterfall or white-water rapids is interpreted as motion and the observer has a sense for the parameters of the flow.

As another type of example, consider a tree blowing in the wind. Even though individual leaves and branches are moving differently from one another, the overall motion of the tree seems natural and self-consistent. A human observer can distinguish the tree from the background and

from other objects that surround it: somehow the coherence of the motion of the leaves helps the observer to separate the tree from its background.

How are these perceptual inferences possible? To fully understand these phenomena we first need to model turbulent motion and the processes that cause it. Section 7.1 discusses the fractal nature of turbulent flow. Section 7.2 discusses how this fractal space-time behavior might allow us to recognize instances of turbulent flows, and to differentiate them from other phenomena. Section 7.2.2 develops a physiologically-plausible technique, using the outputs of motion-energy filters, for estimating the fractal dimension for each region of an image sequence. However, experimentation indicates that this algorithm is *not* reliable for obtaining local estimates of the fractal scaling parameter. In spite of this, we are able to show some preliminary results in Section 7.2.4 discriminating image regions based on fractal scaling.

7.1 A Model of Turbulence

The predominant fact that determines the physical properties of turbulent flow is its lack of *coherence*. The relationship between coherence of motion and the physical properties of moving fluid is well understood, and is summarized by the empirical relationship known as the *Reynolds number*[1]:

$$\Re = \frac{VSD}{v} \tag{62}$$

where V is velocity, S is obstacle size, D is density, and v is viscosity. Several examples should clarify the relationship. Turbulence is proportional to velocity — a flag flutters more in a stronger wind. It is proportional to obstacle size — a freighter makes more wake than a windsurfer. Turbulence is proportional to density since greater density means that there are more particles which can, and will, interact with one another. Viscosity is inversely proportional to $\Re$ — it is easier to make air or water turbulent than it is to make oil or molasses turbulent.

One approach to understanding turbulence is to ask how it arises. For very low speeds (low Reynolds number), the flow around an object is regular and time-independent (laminar flow). As $\Re$ is increased, the motion gains swirls but remains time-independent. As $\Re$ is increased still further, the swirls may break away and start moving downstream. This induces a time-dependent

[1] Osborne Reynolds (19th century), a good introduction to fluid mechanics is given in Aris [9].

Figure 25: Drawing of turbulent flow by Leonardo da Vinci. Notice the similarity across scales (swirls within swirls within swirls) which is characteristic of fractals.

flow pattern since the velocity measured at a point downstream is periodic. A further increase in $\Re$ results in partially periodic and partially irregular velocity as the swirls begin to induce irregular internal swirls. Finally, a very complex velocity field is induced, and the flow becomes completely chaotic. This is called fully developed turbulence (Figure 25). There is, however, an underlying regularity in this motion that can be analyzed. In Figure 25, for instance, we see a series of swirls, and within those swirls are smaller swirls, etc.

Perhaps the most useful current models for the structure of turbulent flow are based on Mandelbrot's [95] notions of *fractal functions.* Recent work in physics pioneered by Mitchell Feigenbaum [39,54] has demonstrated that many deterministic nonlinear systems can result in chaotic behvior which is statistically invariant over a wide range of scales. This work came as a revelation to modern physics. For centuries, probabilistic descriptions of systems were regarded as no more than conveniences to be invoked when the deterministic equations were difficult or impossible to solve for one reason or another. The demonstration that a deterministic system may result in chaotic behavior, however, showed that a probabilistic model of such a system could, in some cases, be *more* valid than a deterministic one. Solving the Navier-Stokes equations for fluid flow provide a perfect example. Turbulent flow is so complicated and chaotic

that attempting to solve the Navier-Stokes equations analytically no longer makes sense; it is more valid to model the flow as a statistical system. The model of turbulent flow discussed in this chapter, therefore, describes the statistical geometry (the fractal characteristics) of turbulent flow rather than studying turbulence analytically in the manner of the Navier-Stokes equations.

7.1.1 The Mathematics of Fractal Brownian Functions

The path of a particle exhibiting Brownian motion is the canonical example of most naturally occuring fractals; the discussion that follows, therefore, will be devoted exclusively to fractal Brownian functions, which are a mathematical generalization of Brownian motion.

Definition: A random function $I(x)$ is a **fractal Brownian function** if for all x and Δx:

$$Pr\left(\frac{I(x+\Delta x)-I(x)}{\|\Delta x\|^H} < y\right) = F(y) \tag{63}$$

where $F(y)$ is a cumulative distribution function, and the variable H is the **fractal scaling parameter.** If $H = 1/2$ and $F(y)$ comes from a zero-mean Gaussian with unit variance, then $I(x)$ is the classical Brownian function.

Note that x can be interpreted as a vector quantity, thus providing an extension to two or more topological dimensions. If the topological dimension is T, the fractal dimension D of $I(x)$ is:

$$D = T + (1 - H) \tag{64}$$

The power spectrum, $P(\omega)$, of $I(x)$ is (see Mandlebrot [95] for discussion of the proof of this proportionality):

$$p(\omega) = c\omega^{-2H-1} \tag{65}$$

for some constant c. A Brownian fractal function as defined by Equation (63), can be generated in the Fourier domain to give a power spectrum as in Equation (65) [138].

Definition: A **fractal Brownian surface** is a continuous function that obeys the statistical description given by Equation (63) for $\delta_{min} < \Delta\vec{x} < \delta_{max}$, where $\vec{x}$ is a two-dimensional vector.

Although true mathematical fractals obey Equation (63) for all $\Delta\vec{x}$, real surfaces have a constant fractal scaling parameter over only a range of scales; δ_{min} and δ_{max} specify the bounds on this range. Although true mathematical fractals are everywhere nondifferentiable, the surface normal of a fractal Brownian surface is defined with respect to δ_{min}.

7.1.2 The Fractal Characteristics of Turbulence

A large number of papers, both theoretical and experimental, have been written during the past few years relating fractals to turbulent or chaotic systems; for instance, see Mandlebrot [95], Lovejoy and Schertzer [90], or recent issues of *Physical Review Letters.* As a result, considerable experimental evidence now supports fractal models of turbulence. For example, Lovejoy (see Lovejoy [90] for references) has verified the fractal scaling of cloud and rain areas and perimeters for scales ranging from .16 to 1000 km.

Turbulence exhibits fractal characteristics in several ways. First, the shape of turbulent regions is fractal. Consider turbulence which is restricted to a portion of an otherwise laminar fluid, e.g., a boat's wake. If we examine the boundary of such a turbulent area (for instance, the oil spill shown in Figure 26), we will discover a hierarchy of vortex-like indentations occuring at all scales. Just like in the coastline example, shape is repeated over a large range of scales. The presence of detail at all scales causes the shape to be fractal in nature. For fluids such as water the magnitude of the indentations increases with Reynolds number.

A second way in which turbulence is fractal concerns its *intermittency.* Turbulence eventually ends in dissipation: due to the fluid's viscosity, the energy of visible motion transforms into heat. Some regions in space are marked by very high dissipation, while other regions seem by contrast nearly free of dissipation. For example, we all know that wind comes in gusts, and within those gusts are smaller scale gusts. This is well illustrated by the "turbulence" one feels during an airplane trip. Every so often, a large airplane is shaken about, which shows that certain regions of the atmosphere are strongly dissipative. A smaller airplane acts as a more sensitive probe: it "feels" turbulent gusts that leave the large airplane undisturbed, and it experiences each shock received by the large airplane as a burst of weaker shocks. Theoretical models and computer models which exhibit the fractal nature of the intermittency of turbulence have been developed by a number of authors including Frisch et al [43], Chorin [29], and Lovejoy and

Figure 26: (from [Van Dyke, 1982]) Turbulent wake showing geometric similarity at different scales.

Schertzer [90].

Perhaps the most convincing argument for fractal models of turbulence is that fractal models can be used to generate images and image sequences that look like turbulent flow. For example, Mandlebrot and Voss have developed models of cloud formations using Brownian fractals.

7.1.3 Generative Models of Turbulence

A turbulent medium can be represented by a fractal function of three variables, i.e., $\vec{x} = (x, y, z)$ and $I(\vec{x})$ represents either the energy (velocity) or density at every point in space[2]. For example, the surface of a cloud is an iso-value surface within the fractal volume, those points in space which that equal energy.

A realistic looking display of a fractal cloud can be generated by letting the local light scattering vary as $I(x, y, z)$ [138], i.e., the light scatters fractally with the same fractal scaling parameter as that of the surface itself. The example cloud shown in Figure 27 was rendered

[2]Note that modeling energy and density are equivalent since dynamic systems tend to concentrate particles in regions of low velocity, i.e., minimize energy.

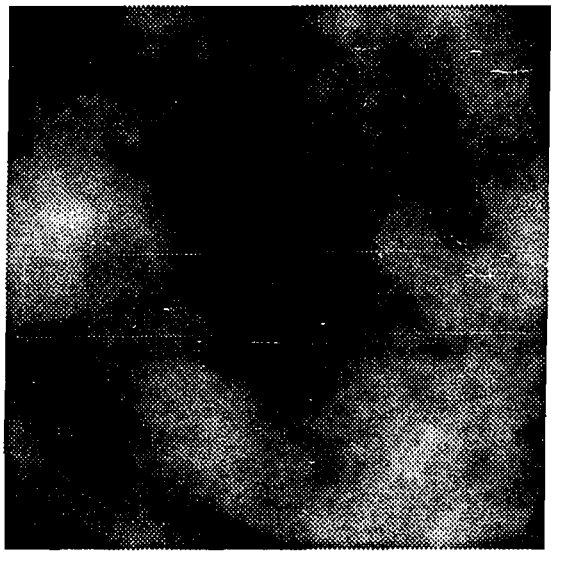


Figure 27: Computer generated image of clouds.

assuming that the light was being transmitted through the fractal volume rather than reflecting from its surface (e.g., looking up at the sky during mid-day).

A fractal function with $\vec{x} = (x, y, z, t)$ models a cloud which is changing its shape over time. If we add a bias flow field to the zero-mean fractal, then the cloud will move as it changes in shape,

$$D(\vec{x}) = I(\vec{x}) + \vec{V}(\vec{x}) \tag{66}$$

where $D(\vec{x})$ is the cloud's density as a function of space and time, $I(\vec{x})$ is a four dimensional zero-mean fractal, and $\vec{V}(\vec{x})$ is potential flow.

Other turbulent systems can be modeled in similar ways. For example, white water rapids are just like clouds with another term in the fractal energy function due to gravity which sticks most of the fractal volume to the ground. Fire can be modeled by adding another term which adds energy due to heat causing the smaller particles to rise.

Note that in modeling turbulent flow in this manner we are assuming that the fractal is isotropic in space-time, i.e., any submanifold or slice through the function in space-time will have the same fractal scaling parameter. This assumption is classically known as *Taylor's assumption of frozen turbulence* that has been empirically verified for atmospheric turbulence by Brown and Robinson [31] for distances up to 1000 km. The consequence of this assumption

is that for each instant in time, each component of the surface normal obeys the statistical description of Equation (63). Moreover, if we look at one position in space for a series of time intervals, the surface normal obeys the same statistical description. The orientation of the surface normal at a point $\vec{x}_1$ in space is changing over time according to Equation (63). Similarly, the orientation at a different point $\vec{x}_2$ varies according to Equation (63). Statistically, $\vec{x}_1$ and $\vec{x}_2$ are changing the same way, but the two points are out of "phase." For example, the surface normal at one point may be leaning in one direction while it is leaning in the opposite direction at the other. A snapshot of the entire signal has the same statistical behavior as that of a single point over a period of time.

Although this is an extremely simple model of turbulent systems, it is both first-order correct in terms of the underlying physics, and produces correct-looking images of turbulent phenomena. For purposes of perception, therefore, this model may be sufficient. If we were trying to model the detailed dynamics of this system, however, a more complicated fractal-based model (e.g., Mandlebrot and Lovejoy [89]) would be more useful.

7.2 Recognizing Turbulence

In the first part of this section we discuss how this fractal space-time behavior might allow us to recognize instances of turbulent flows, and to differentiate them from other phenomena. Second, we present a technique for measuring the fractal scaling parameter of an image sequence using the outputs of motion-energy filters. Finally, we show some preliminary results.

7.2.1 Recognizing Instances of Turbulent Flow

How do we know if we are really looking at a turbulent system? As an analogy, let us again consider the case of rigid motion. A solid object moves rigidly in 3-space, and such motion projects into an image sequence with a specific type of regularity. We use the rigid-motion model to detect that regularity, and then make two inferences: (1) the regularity is due to rigid motion in 3-space; (2) the rigid motion is a result of a single, solid object moving through space.

A similar chain of relationships holds for turbulent flow. If a motion is turbulent in 3-space, then according to our model of turbulent flow it will obey a scaling law as in Equation (63).

Pentland and Kube [85,109] have shown that under a variety of imaging situations fractal surfaces project to fractal images. We have used this result to generate realistic-looking (synthetic) sequences of moving clouds and tree leaves. The ability to generate realistic cloud sequences is partial confirmation of the validity of the model. The problem remaining, then, is to invert the process. That is, we want to detect instances of fractal scaling in an image sequence, that will then allow us to make the following inferences: (1) that this regularity results from a motion in 3-space described by the fractal model; (2) that this fractal 3-space motion is, in fact, a turbulent flow.

For both rigid and turbulent motion, we know that our inferences will generally be reliable, because we can normally preclude both the ways in which our inferences can go wrong.

In the case of a moving solid body, the first type of potential error is that we think we have a rigid motion when in fact we do not. We can preclude this type of error because the equations are overconstrained; i.e., we can estimate the motion parameters using part of our data, and then check our answers using the remaining data. The second type of error is that we think we do not have an instance of rigid motion when in fact we do. We will never make this error since we will always infer rigidity when the the data is self-consistent.

In the case of turbulent flow, similarly, the first type of potential error is that we think we have an instance of turbulent flow when in fact we do not. As in the rigid motion case we can preclude this type of error because the equations are overconstrained: the estimated fractal scaling parameter H must have the *same* value (to within the uncertainty of image noise) for a range of scales in both space and time. Thus we can estimate the motion parameters using part of our data, and then check our answers using the remaining data. The second type of error is that we think we do not have an instance of turbulent flow when in fact we do. We will never make this error since we will always infer turbulence when the the data is self-consistent.

In future research, we hope to develop a reliable statistical test for recoginizing fractal processes based on consistency of fractal behavior across a range of scales.

7.2.2 Measurement of Fractal Scaling Parameter

The fractal scaling parameter H can be measured either directly from the second-order statistics (dipole statistics) of $I(x)$ by use of Equation (63), or from $I(x)$'s Fourier power spectrum $P(\omega)$

by use of Equation (65). From Equation (65) we get

$$\log[P(\omega)] = -(2H + 1)\log(\omega) + \log(c) \tag{67}$$

Pentland [109] computed the fast Fourier transform (FFT) for each 8x8 window of an image and used a linear regression on log-power versus log-frequency to determine the fractal scaling parameter H.

We have developed an efficient method of measuring the fractal scaling parameter using physiologically-plausible linear filters. As discussed in Chapters 2 and 3, an energy filter is the sum of the squared outputs of a quadrature pair (odd- and even-phase) of linear bandpass filters. Parseval's theorem states that the integral of the squared values over the space-time domain is proportional to the integral of the squared Fourier components over the frequency domain. Convolution with a bandpass filter results in a signal that is restricted to a limited range of frequencies. Therefore, the integral of the square of the convolved signal is proportional to the integral of the power of the original signal over this range of frequencies. The average output of an energy filter is thus proportional to the amount of power (energy) in the Fourier spectrum of the signal that lies within the filter's sensitive range.

In previous research [63], we proposed that the logarithim of the ratio of the outputs of two energy filters is linearly related to the fractal scaling parameter. Mallat [93] has since proven that this is the case. Let $\mathcal{R}(\omega_0)$ be the average output of an energy filter tuned to frequency ω_0, and let $\mathcal{R}(2\omega_0)$ be the average output of a filter with twice the bandwidth tuned to twice the frequency. Mallat [93] shows that

$$\frac{\mathcal{R}(2\omega_0)}{\mathcal{R}(\omega_0)} = 2^{-2H} \tag{68}$$

This formula may be generalized by considering pairs of filters that are not necessarily an octave apart.

$$\frac{\mathcal{R}(d\omega_0)}{\mathcal{R}(\omega_0)} = d^{-2H} \tag{69}$$

or

$$\frac{\log[\mathcal{R}(2\omega_0)] - \log[\mathcal{R}(\omega_0)]}{\log(d)} = -2H \tag{70}$$

An algorithm for estimating the fractal scaling parameter for each region of an image sequence is as follows:

1. Convolve the image sequence with three-dimensional sine- and cosine-phase Gabor filters with their peak response at ω_0.

2. Convolve with sine- and cosine- phase Gabor filters with their peak response at $d\omega_0$.

3. Compute the Gabor energy for each of steps (1) and (2) by summing the squares of the responses of the sine- and cosine-phase filters.

4. Average the results of step (3) by convolving with a Gaussian.

5. Estimate H using Equation (70).

6. Do steps (1) through (5) for a variety of orientations and ω_0's.

However, experimentation with static images indicates that this algorithm is *not* reliable for obtaining local estimates of the fractal scaling parameter. In order to get accurate estimates of H we must average over large image regions in Step (4). On real textured images, the algorithm produces markedly inferior results compared to Pentland's original regression technique.

In future research, it will be interesting to investigate why these two similar techniques give such different results, and to develop a robust and efficient mechanism for estimating fractal scaling parameter.

7.2.3 Turbulent Flow and Bias Flow

Many natural turbulent processes have a mean bias flow, e.g., clouds move while they change in shape, and the turbulent wake of a boat appears to follow the boat. Mathematically, we address this as in Equation (66), by adding a bias flow velocity to the stationary spatiotemporal fractal image sequence.

Consider a fractal image that merely translates in the image plane from frame to frame, and a space-time fractal image sequence in the presence of a bias flow field that changes its spatial form (bubbles) while it translates. Both of these image sequences are fractal in space and in time. For each instant in time, each obeys the fractal scaling law across space, and if we look at one spatial location for a series of time intervals each obeys the fractal scaling law over time. For example, the rigidly-moving landscape region and the turbulent cloud region in the yosemite

fly-through sequence (Figure 9) are both fractal in space and in time. The technique described so far in this chapter will be unable to distinguish between such regions.

The difference between these two motions is that if you take away the bias flow, one will be fractal in space but uniform in time while the other will still be fractal in space and in time. Chapter 3 presents a model for the extraction of local image velocity. The extracted flow field may be used to compensate for the bias flow. This procedure was used for the preliminary results discussed below.

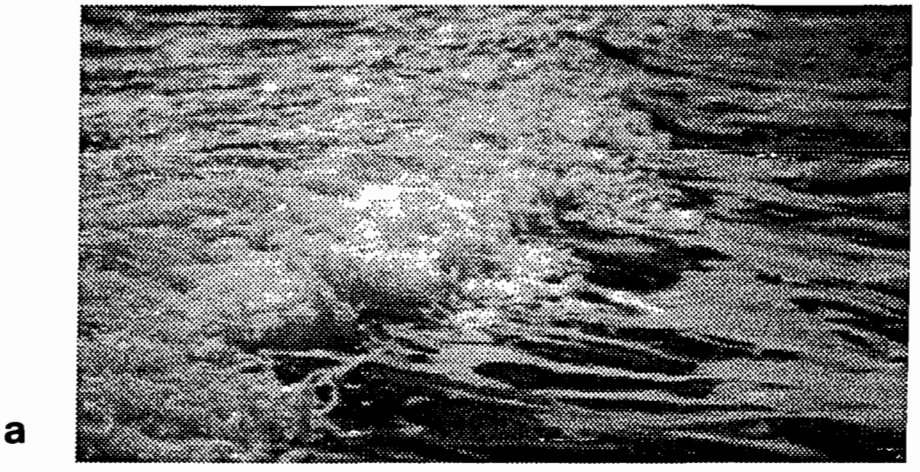
Alternatively, the extracted image flow may be used to drive eye/camera movements as discussed in Appendix E. This is an active solution to compensating for bias flow over a local image region.

7.2.4 Results

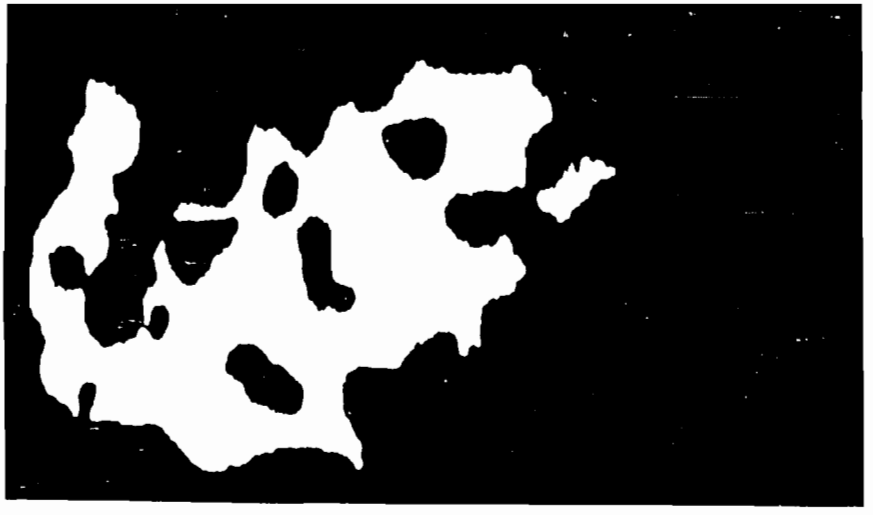
As yet, we have been unable to obtain real image sequences of turbulent motion. The results below are for a static real image and for a computer-generated image sequence.

Motor-Boat Wake Figure 28(a) is a picture of a turbulent wake behind a motor boat. We estimated the fractal scaling parameter of this image at each pixel for four different orientations over two octaves of spatial frequency. If the estimates for a given region of the image are not consistent, then we know that the region is not fractal, i.e., not due to a turbulent process. The result for a certain choice of thresholds is shown in Figure 28(b). Thresholding was used in this example merely to demonstrate that fractal scaling parameter distinguishes between the two regions — other segmentation procedures may be used instead. The turbulent regions near the edge of the picture are missed due to the edge effects of the convolution.

Yosemite Fly-Through For the Yosemite fly-through image sequence (Figure 9 in Chapter 3), we may compensate for the average image motion by shifting each local region of each image in the sequence opposite to the extracted flow. This results in a new image sequence in which the landscape region is motionless. The clouds, on the other hand, were generated as stationary, spatiotemporal fractals (they change their shape over time) in the presence of a bias flow field that moves them rightward. Compensating for the extracted bias flow yields stationary clouds



a



b

Figure 28: (a) Turbulent wake behind a motor boat. Fractal dimension was estimated for four orientations over two octaves of spatial frequency. (b) Region for which the estimates were consistent with one another within a given threshold.

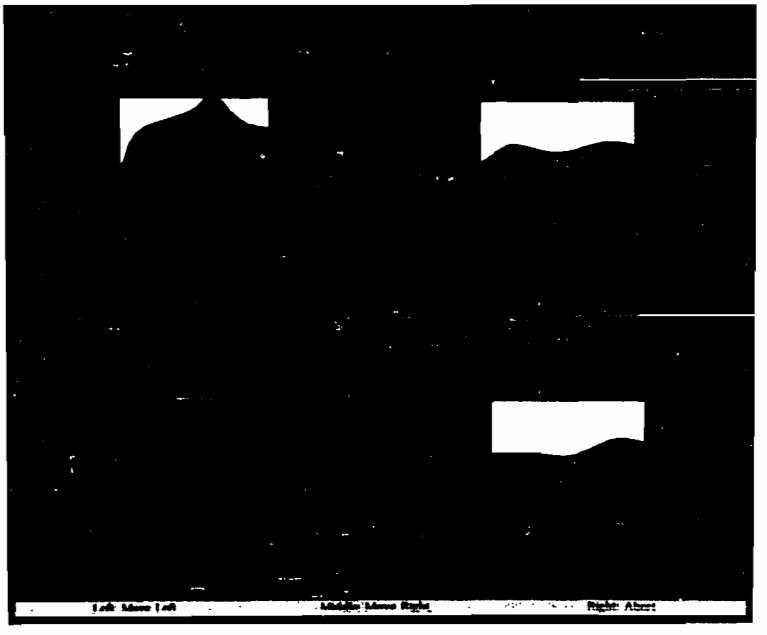


Figure 29: Segmentation of the Yosemite fly-through image sequence based on fractal scaling parameter using a threshold. (top-left) Segmentation using filters oriented along the t-axis. (top-right) oriented along the x-axis. (bottom-right) oriented along the y-axis. (bottom-left) Histogram of fractal scaling parameter used to pick the threshold.

that still change their shape over time. Figure 29 shows the segmentation based on fractal scaling parameter using a threshold. Again, thresholding was used in this example merely to demonstrate that the fractal measure distinguishes between the two regions.

7.3 Summary

This chapter discusses a fractal model for turbulent flow. The recognition of turbulent flow is analogous to the recogition of rigid motion; it indicates whether or not a particular model of motion is applicable to a given region of an image sequence. If we observe that the fractal scaling parameter within a given region changes over space, time, or scale, then we are certain that the region is not an image of a single turbulent flow. On the other hand, if the observations are constant over space, time, and scale, then we may infer that we are observing a single turbulent flow.

A technique is presented for estimating the fractal scaling parameter of fractal image sequences using linear filters. However, experimentation indicates that this algorithm is *not* reliable for obtaining local estimates of the fractal scaling parameter. In future research, we hope to develop a reliable algorithm for estimating fractal scaling parameter and a reliable statistical test for recoginizing fractal processes.

Chapter 8

Conclusion

As observers move through the environment or shift their direction of gaze, the world moves past them. In addition, there may be objects that are moving differently from the static background, either rigid-body motions or nonrigid (e.g., turbulent) ones. This dissertation discusses several models for motion perception that: (1) extract image flow; (2) detect moving objects and recognize the relative motion of the stationary environment due to an observer's own movement; (3) recognize rigid-body motion of moving objects; (4) recognize turbulent flow. The computations for all of these models are based on measuring motion energy, a multiresolution representation of motion information extracted from image sequences.

This dissertation asserts that the basic function of preattentive/peripheral/immediate visual perception is perceptual organization, the detection of regularities in images that correspond to regularities in the environment. The models discussed in this proposal allow us to test the hypothesis that regions of an image sequence are the result of certain processes in the three-dimensional physical world (e.g., rigid motion or turbulent motion), and then recover the parameters (e.g., 3-D shape and 3-D motion) of those processes. The ultimate goal of this research is to determine which model is most appropriate for a given region.

Combining data from different sensors and using active vision (e.g., head and eye movements) together provide extra constraints on a number of vision problems. Motion analysis plays a key role for active observers who are moving their head and eyes in order to better perceive the environment. Conversely, active vision and sensor fusion are key ingredients for motion

analysis, particularly since sensor information is subject to noise and uncertainty.

8.1 Image Flow

This dissertation presents a model for computing local image velocity consonant with current views regarding the neurophysiology and psychophysics of motion perception. The power spectrum of a moving texture occupies a tilted plane in the spatiotemporal-frequency domain. The model uses 3-D (space-time) Gabor filters to sample this power spectrum and by combining the outputs of several such filters the model estimates the slope of the plane (i.e., the velocity of the moving texture). The model gives accurate estimates of two-dimensional velocity for a wide variety of test cases including realistic images, sequences generated from images of natural textures, and some sine-grating plaid patterns.

The error in the velocity estimates for translating image sequences is from two sources. First, image textures are stochastic — thus Equation (8) is correct only on average. I posit an additive Gaussian model for the variability in the motion energy measurements. Second, the maximum-likelihood estimate is equal to the least-squares estimate only for the case of additive Gaussian process variability — thus the optimality of using least-squares depends on the validity of the Gaussian approximation in Equations (22) and (21)

The primary source of error for realistic image sequences is that the model assumes image translation, ignoring motion boundaries, accelerations, deformations (rotation, divergence, shear), and motion transparency. Rather, the model computes the average image velocity within a Gaussian-shaped window.

A parallel implementation of the model results in a distributed representation of image velocity. The computations leading to this distributed representation are simply a series of linear steps (convolutions, weighted sums) alternating with point nonlinearities (squaring, exponentiation). The model is therefore encompassed by the general framework for parallel distributed processing put forth by Rummelhart and McClelland [119].

Sensor error can be characterized by a sensor model that is a statistical model of a sensor's ability to observe the environment. A sensor model is formulated for the extraction of image flow, and it is demonstrated that the sensor model accurately reflects the actual error in the

velocity estimates for translating image sequences of Gaussian white-noise textures. However, the sensor model significantly underestimates the actual errors for realistic image sequences because these errors are mainly due to deformations, accelerations, and motion boundaries in the flow. They are not simply due to the motion-energy measurement noise.

Ambiguity due to the aperture problem is a special case of image flow uncertainty. Preliminary results indicate that ambiguity due to the aperture problem might be recognized using the sensor model.

The model appears to solve the aperture problem as well as the human visual system since it extracts the correct velocity for patterns having large differences in contrast at different spatial orientations ($> 32 : 1$ contrast ratio for some patterns). The model's capability for velocity discrimination (Weber fraction of 0.029) is also comparable to that of the human visual system.

This dissertation thus demonstrates the promise of computing optical flow using motion energy filters. There are any number of related techniques (e.g., [74,141] using different filters, or using different rules for combining the filter outputs). I suggest using psychophysical and electrophysiological experimentation to distinguish between them.

The model may be used to simulate psychophysical data on velocity discrimination and on the coherence of sine-grating plaids. In [62] I compare the computations performed by the model to the stages of the visual motion pathway of the primate brain, and I suggest how the model might be used to simulate electrophysiological data.

For the most part, simulating physiological and psychophysical data merely demonstrates that the model is consistent with some of the experimental results on biological motion perception. The emphasis in future research will be to compare the predictions made by this model to those made by alternative image-flow models and to test those predictions with further experiments. Thus, the model may prove to be an interesting framework for future research in the psychophysics and neurophysiology of motion perception as well as in computer vision.

8.2 Egomotion and Rigid-Body Motion

Previous research [24,61,130] has demonstrated that we can recover scene structure and we can often detect moving objects given information about both the camera motion and the image

motion. However, information from perceptual sources (e.g., observations of image motion and camera motion) is inherently noisy and uncertain. This dissertation poses the detection of moving objects and the recovery of depth from motion as sensor fusion problems that necessitate combining information from different sensors in the presence of noise and uncertainty.

Prior information about the camera motion constrains the optical flow fields that can be generated by moving through an otherwise static environment. Given the motion parameters, the image velocity at an image point is constrained to be along a line segment in image velocity space. In addition, prior information on depth (from time past and from other sources such as stereo) places additional constraints by specifying a smaller interval for the line segment. Image velocity for a given image location corresponds to a point in image-velocity space. We utilize Mahalanobis distance as a threshold rule for determining consistency between these two sensor observations.

If the two observations are consistent, then the image motion may be accounted for by the camera motion alone, i.e., the image region may correspond to a surface patch that is stationary in the environment. In this case, we may obtain an estimate of the relative depth to that surface. On the other hand, if the observations are not consistent, then the image motion must be due to a moving object.

For observations that are consistent, we have derived an equation that tells us how to update estimates of the egomotion paraters in the presence of the new image motion data. This suggests an incremental scheme for recovering the motion parameters, perhaps utilizing the extended Kalman filter [47].

The procedure outlined in this dissertation for the analysis of egomotion and stationary surfaces may be generalized to arbitrary motions of rigid objects. Given prior information about the the rigid-body motion parameters for some small patch of the surface of a moving object, we may utilize image flow information to extract entire regions that are consistent with those motion parameters, recover depth, and update estimates of the motion parameters.

This dissertation proposes two additional sources of information for recognizing rigid-body motion; deformation fields extracted directly from the time-varying imagery, and records of both eye/camera motion and eye/camera position while tracking a surface point on a rigid object.

This dissertation proposes that a reliable method for recovering the motion parameters should

obey the following three general principles: (1) the motion parameters should be computed using a small image region that corresponds to a single moving surface; (2) the recovery method should utilize as much data as possible in order to have sufficient overconstraint for reliable estimates; (3) the method should provide an explicit test for rigidity by checking for consistency amoungst the data. In order to simultaneously satisfy the first two principles, the method must collect data over time while tracking the motion of a small patch of surface. This suggests an incremental scheme (perhaps using the extended Kalman filter [47]) that tracks the motion of a small surface patch, utilizing the incoming image motion and eye/camera movement data to update estimates of the motion parameters.

8.3 Turbulent Flow

This dissertation discusses a fractal model for turbulent flow. The recognition of turbulent flow is analogous to the recogition of rigid motion; it indicates whether or not a particular model of motion is applicable to a given region of an image sequence. If we observe that the fractal scaling parameter within a given region changes over space, time, or scale, then we are certain that the region is not an image of a single turbulent flow. On the other hand, if the observations are constant over space, time, and scale, then we may infer that we are observing a single turbulent flow.

A technique is presented for estimating the fractal scaling parameter of fractal image sequences using linear filters. Unfortunately, experimentation indicates that this algorithm is not reliable for obtaining local estimates of the fractal scaling parameter. In future research, I hope to develop a reliable algorithm for estimating fractal scaling parameter and a reliable statistical test for recoginizing fractal processes.

8.4 Contributions

This dissertation presents a model of image flow that accurately and robustly estimates image velocity for translating textured image sequences. The model appears to extract image velocity with accuracy comparable to that of the human vision system. It is robust with respect to image

noise and with respect to the aperture problem. For translating textured image sequences, I do not believe that you can do much better.

The image flow model also demonstrates the usefulness of the motion energy multiresolution representation of image sequences.

Furthermore, the image flow model is an exciting framework for motivating future research in the psychophysics and neurophysiology of biological motion perception.

The image flow research presented in this dissertation also provides a good example of sensor modeling. A general goal of sensor design is to make it possible to model the sensor analytically, rather than just empirically. The image flow model is based on signal processing mathematics (probability, stochastic processes, and linear systems) [106]. Thus, it was possible to formulate a sensor model for the error in the velocity estimates.

The model for detecting moving objects and recovering static scene structure provides an example of a formulation of a sensor fusion problem based on the general framework for active vision/sensor fusion put forth by Bajcsy et al [6,7,36,84].

The research presented in this dissertation emphasizes perceptual organization. For many perceptual tasks (e.g., navigation) we do not need exact quantitative information like precise localization of boundaries and precise depth estimates. Rather we rely on qualitative information like moving versus stationary and near versus far.

This dissertation presents two examples of recovering qualitative information from a realistic image sequence: (1) separating the clouds from the landscape in the Yosemite fly-through image sequence (Figures 22 and 29); (2) locating the valley in the Yosemite sequence (Figure 23).

8.5 Future Research

The experience I have had working with motion energy motivates a variety of research into low-level vision and image representation. Multiresolution energy representations should prove to be useful for solving many low-level vision problems including motion analysis, texture discrimination, orientation selection, and boundary detection [1,21,48,94,133]. An important issue for energy representations is the choice of filters. Adelson and Simoncelli [4] and Mallat [93], for example, suggest using quadrature-mirror filters that form an orthogonal and complete

basis for image representation.

In section 3.1.2 I allude to two issues that are fundamental for image representation with energy measures: (1) measuring average local energy without spatial blurring; (2) automatic gain control (adaptation). Several researchers [1,92,133] are looking into addressing these issues using cascades of energy filters. First, we convolve the image with sine- and cosine-phase filters, square and add to get energy, and quantize for efficiency and for automatic gain control. Then, we repeat the entire process on the resulting image decomposition. We end up with a multiresolution decomposition of each level of a multiresolution encoding of the original image.

The limitation of the image flow model presented in this dissertation is that it assumes image translation, ignoring motion boundaries, accelerations, deformations (rotation, divergence, shear), and motion transparency. Rather, the model computes the average image velocity within a Gaussian-shaped window.

Motion transparency is an important future direction for image flow research. Motion transparency is abundant in the real world, for example, specularities and shadows move differently from the surfaces upon which they rest. In principle, the distributed representation of image flow discussed in Chapter 3 can encode multiple velocity estimates corresponding to different motions in the same spatial location. The frequency-domain analysis used to develop the model is also extendable to transparent motions, e.g., the power spectrum of two translating image sequences superimposed on top of one another occupies two planes in the spatiotemporal frequency domain.

Further extensions to the image flow model would help it deal with motion boundaries, deformations, and accelerations. One way of detecting motion boundaries might be to use a cascade of energy filters [1] as described above. One way of estimating deformation fields might be to use deformation filters (e.g., energy filters that are sensitive to clockwise and counterclockwise image rotations), as discussed in Chapter 6. Zucker et al [107,153] propose using relaxation labeling to deal simultaneously with deformations, accelerations, motion boundaries, and transparency.

Another important unsolved issue is how to combine image motion information extracted from the different levels in a multiresolution motion analysis. This problem is complicated by temporal aliasing in temporaliy sampled image sequences. A simple coarse to fine strategy is

not sufficient since the fine resolution (high spatial frequency) motion estimates may be subject to temporal aliasing. One possible solution is to avoid temporal aliasing by analog low-pass temporal filtering in the CCD array before sampling.

Energy models of low-level vision, image representation, texture discrimination, and image motion analysis are motivated in large part by recent progress in psychophysics and electrophysiology. An exciting prospect for future research is the interaction of computational (theoretical) research and experimental (psychophysical and electrophysiological) research to better understand both biological and machine vision. The goal is to compare the predictions made by various alternative models in order to motivate further psychophysical and electrophysiological experiments.

Future research on detecting moving objects should emphasize robustness. In particular, how robust is the detection of moving objects in the face of poor sensor models (inaccuracies in the variance-covariance matrices, violotions of the Gaussianity assumption)? As discussed above, we have a reasonably good understanding of the limitations of the image flow model. We need to understand how these limitations affect the qualitative judgement of moving versus stationary.

Another direction for future research on motion interpretation is to study the incremental scheme for rigid-body motion perception proposed in Chapter 6; tracking the motion of a small surface patch, utilizing the incoming image motion and eye/camera movement data to update estimates of the motion parameters.

In order to investigate the robustness of motion interpretation schemes using real image data it will be necessary to extract image motion information rapidly. Singh [121] has begun an implementation of my image flow model on a PIPE machine — he believes that the flow field for a 512 X 512 X 7 image sequence can be computed in about two seconds.

Fractal models of physical systems are now being used in a variety of scientific disciplines including physics, chemistry, astronomy, and meteorology. Reliable algorithms both for estimating fractal scaling parameter and for recognizing instances of fractal processes are therefore of broad interest.

A final direction for future research is to develop overconstrained models for other types

of motion (e.g., elastic motion), as well as for other types of visual information (e.g., overconstrained shape models [111,12] and overconstrained texture models [63,109]).

Appendix A

Gabor Filters From Separable Components

To convolve a two-dimensional image by a horizontally oriented sine-phase Gabor filter, we may convolve each image row by a one-dimensional sine-phase Gabor filter, then convolve each column of the resulting image by a one-dimensional Gaussian. This appendix outlines a new technique for building three-dimensional Gabor filters of any orientation and with elliptical Gaussian windows of any aspect ratio from linear combinations of separable filters by making use of the following trigonometric identities:

$$\begin{aligned}\sin(\omega_{t_0} + t\omega_{x_0} x + \omega_{y_0} y) &= \sin(\omega_{t_0} t)\cos(\omega_{x_0} x)\cos(\omega_{y_0} y) \\ &- \sin(\omega_{t_0} t)\sin(\omega_{x_0} x)\sin(\omega_{y_0} y) \\ &+ \cos(\omega_{t_0} t)\sin(\omega_{x_0} x)\cos(\omega_{y_0} y) \\ &+ \cos(\omega_{t_0} t)\cos(\omega_{x_0} x)\sin(\omega_{y_0} y)\end{aligned} \tag{71}$$

$$\begin{aligned}\cos(\omega_{t_0} + \omega_{x_0} x + \omega_{y_0} y) &= \cos(\omega_{t_0} t)\cos(\omega_{x_0} x)\cos(\omega_{y_0} y) \\ &- \cos(\omega_{t_0} t)\sin(\omega_{x_0} x)\sin(\omega_{y_0} y) \\ &- \sin(\omega_{t_0} t)\sin(\omega_{x_0} x)\cos(\omega_{y_0} y) \\ &- \sin(\omega_{t_0} t)\cos(\omega_{x_0} x)\sin(\omega_{y_0} y)\end{aligned} \tag{72}$$

Let $G_s(t, \sigma_t, \omega_{t_0})$ be a one-dimensional sine-phase Gabor function as given by Equation (3), and let $G_c(t, \sigma_t, \omega_{t_0})$ be the corresponding cosine-phase filter. Using Equation (71), the output of an arbitrarily-oriented three-dimensional (space-time) sine-phase Gabor filter may be computed by doing the following separable convolutions:

1. Convolve the image sequence in time by $G_s(t, \sigma_t, \omega_{t_0})$, next each image row by $G_c(x, \sigma_x, \omega_{x_0})$, and then each column by $G_c(y, \sigma_y, \omega_{y_0})$.

2. Convolve the image sequence in time by $G_s(t, \sigma_t, \omega_{t_0})$, next each image row by $G_s(x, \sigma_x, \omega_{x_0})$, and then each column by $G_s(y, \sigma_y, \omega_{y_0})$.

3. Convolve the image sequence in time by $G_c(t, \sigma_t, \omega_{t_0})$, next each image row by $G_s(x, \sigma_x, \omega_{x_0})$, and then each column by $G_c(y, \sigma_y, \omega_{y_0})$.

4. Convolve the image sequence in time by $G_c(t, \sigma_t, \omega_{t_0})$, next each image row by $G_c(x, \sigma_x, \omega_{x_0})$, and then each column by $G_s(y, \sigma_y, \omega_{y_0})$.

5. Subtract the result of Step (2) from the sum of the results of Steps (1), (3), and (4). Note that if σ_x, σ_y, and σ_t are not equal, the Gaussian window will be elliptical, but the axes of the ellipsoid will always be parallel to the x, y, and t axes.

Appendix B

Gabor Energy

In this appendix I derive an equation for the Gabor energy of a one-dimensional sine wave and for the Gabor energy of a one-dimensional Gaussian white-noise process. I also derive an equation for the covariance of the outputs of two Gabor-energy filters, each convolved with a Gaussian white noise signal.

B.1 Gabor Energy for a Sine Wave

The Fourier transforms of a Gaussian function, a sine wave, and a cosine wave are:

$$\mathcal{F}\left\{\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right\} = \exp(-2\pi^2\sigma^2\omega^2) \tag{73}$$

$$\mathcal{F}\{\sin(2\pi\omega_0 x)\} = i/2[\delta(\omega+\omega_0) - \delta(\omega-\omega_0)] \tag{74}$$

$$\mathcal{F}\{\cos(2\pi\omega_0 x)\} = 1/2[\delta(\omega+\omega_0) + \delta(\omega-\omega_0)] \tag{75}$$

One-dimensional Gabor functions are:

$$G_s(\omega_0,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\sin(2\pi\omega_0 x) \tag{76}$$

$$G_c(\omega_0,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\cos(2\pi\omega_0 x) \tag{77}$$

The Fourier transform of a Gabor function is the convolution of Equation (73) with either Equation (74) or (75),

$$\mathcal{F}\{G_s(\omega_0,\sigma)\} = i/2\left\{\exp[-2\pi^2\sigma^2(\omega+\omega_0)^2] - \exp[-2\pi^2\sigma^2(\omega-\omega_0)^2]\right\} \tag{78}$$

$$\mathcal{F}\{G_c(\omega_0,\sigma)\} = 1/2\left\{\exp[-2\pi^2\sigma^2(\omega+\omega_0)^2] + \exp[-2\pi^2\sigma^2(\omega-\omega_0)^2]\right\} \quad (79)$$

The Shift theorem states:

$$\mathcal{F}\{f(x-a)\} = \exp[-2\pi i\omega a]\mathcal{F}\{f(x)\} \quad (80)$$

Using Equations (80) and (74) gives the Fourier transform of a sine wave at any phase:

$$\begin{aligned}\mathcal{F}\{\sin(2\pi\bar{\omega}x+\phi)\} &= \mathcal{F}\left\{\sin\left[2\pi\bar{\omega}\left(x-\frac{-\phi}{2\pi\bar{\omega}}\right)\right]\right\} \quad (81)\\ &= (i/2)\exp(i\phi\omega/\bar{\omega})[\delta(\omega+\bar{\omega})-\delta(\omega-\bar{\omega})]\end{aligned}$$

Note that if $\phi = 0$ then Equation (81) equals Equation (74) and if $\phi = \pi/2$ then Equation (81) equals Equation (75).

The squared-output of a sine-phase Gabor filter convolved with a sine-wave of arbitrary phase is equal to (by Parseval's theorem) the squared-power of the product of their Fourier transforms:

$$\begin{aligned}\int_{-\infty}^{\infty}|G_s(\omega_0,\sigma)*\sin(2\pi\bar{\omega}x+\phi)|^2\,dx &= \int_{-\infty}^{\infty}|\mathcal{F}\{G_s(\omega_0,\sigma)\}\mathcal{F}\{\sin(2\pi\bar{\omega}x+\phi)\}|^2\,d\omega \quad (82)\\ &= |(1/4)[f(\omega)-g(\omega)]\exp(i\phi\omega/\bar{\omega})[\delta(\omega+\bar{\omega})-\delta(\omega-\bar{\omega})]|^2\\ &= |(1/4)[f(\omega)-g(\omega)][\cos(\phi\omega/\bar{\omega})+i\sin(\phi\omega/\bar{\omega})][\delta(\omega+\bar{\omega})-\delta(\omega-\bar{\omega})]|^2\\ &= (1/16)[f(\omega)-g(\omega)]^2[\cos^2(\phi\omega/\bar{\omega})+\sin^2(\phi\omega/\bar{\omega})][\delta(\omega+\bar{\omega})-\delta(\omega-\bar{\omega})]^2\\ &= (1/16)[f(\omega)-g(\omega)]^2[\delta(\omega+\bar{\omega})+\delta(\omega-\bar{\omega})-2\delta(\omega+\bar{\omega})\delta(\omega-\bar{\omega})]\\ &= (1/16)[f(\bar{\omega})-g(\bar{\omega})]^2+(1/16)[f(-\bar{\omega})-g(-\bar{\omega})]^2\\ &= (1/8)[f(\bar{\omega})-g(\bar{\omega})]^2\end{aligned}$$

i.e.,

$$\int_{-\infty}^{\infty}|\mathcal{F}\{G_s(\omega_0,\sigma)\}\mathcal{F}\{\sin(2\pi\bar{\omega}x+\phi)\}|^2\,d\omega = (1/8)[f(\bar{\omega})-g(\bar{\omega})]^2 \quad (83)$$

where

$$\begin{aligned}f(\omega) &= \exp[-2\pi^2\sigma^2(\omega-\omega_0)^2] \quad (84)\\ g(\omega) &= \exp[-2\pi^2\sigma^2(\omega+\omega_0)^2]\end{aligned}$$

Similarly, the squared-output of a cosine-phase Gabor filter convolved with a sine-wave of arbitrary phase is given by

$$\int_{-\infty}^{\infty} |G_c(\omega_0,\sigma) * \sin(2\pi\bar{\omega}x+\phi)|^2\,dx = \int_{-\infty}^{\infty} |\mathcal{F}\{G_c(\omega_0,\sigma)\}\mathcal{F}\{\sin(2\pi\bar{\omega}x+\phi)\}|^2\,d\omega \quad (85)$$

$$\begin{aligned}
&= |(i/4)[f(\omega)+g(\omega)]\exp(i\phi\omega/\bar{\omega})[\delta(\omega+\bar{\omega})-\delta(\omega-\bar{\omega})]|^2 \\
&= |(1/4)[f(\omega)+g(\omega)][i\cos(\phi\omega/\bar{\omega})-\sin(\phi\omega/\bar{\omega})][\delta(\omega+\bar{\omega})-\delta(\omega-\bar{\omega})]|^2 \\
&= (1/16)[f(\omega)+g(\omega)]^2[\cos^2(\phi\omega/\bar{\omega})+\sin^2(\phi\omega/\bar{\omega})][\delta(\omega+\bar{\omega})-\delta(\omega-\bar{\omega})]^2 \\
&= (1/16)[f(\omega)+g(\omega)]^2[\delta(\omega+\bar{\omega})+\delta(\omega-\bar{\omega})-2\delta(\omega+\bar{\omega})\delta(\omega-\bar{\omega})] \\
&= (1/16)[f(\bar{\omega})+g(\bar{\omega})]^2+(1/16)[f(-\bar{\omega})+g(-\bar{\omega})]^2 \\
&= (1/8)[f(\bar{\omega})+g(\bar{\omega})]^2
\end{aligned}$$

i.e.,

$$\int_{-\infty}^{\infty} |\mathcal{F}\{G_c(\omega_0,\sigma)\}\mathcal{F}\{\sin(2\pi\bar{\omega}x+\phi)\}|^2\,d\omega = (1/8)[f(\bar{\omega})+g(\bar{\omega})]^2 \quad (86)$$

Combining Equations (83) and (86), gives the phase-independent gabor energy:

$$\begin{aligned}
(1/8)[f(\bar{\omega})-g(\bar{\omega})]^2+(1/8)[f(\bar{\omega})+g(\bar{\omega})]^2 &= (1/4)[f^2(\bar{\omega})+g^2(\bar{\omega})] \quad (87)\\
&= (1/4)\exp[-4\pi^2\sigma^2(\bar{\omega}-\omega_0)^2] \\
&\quad +(1/4)\exp[-4\pi^2\sigma^2(\bar{\omega}+\omega_0)^2]
\end{aligned}$$

B.2 Gabor Energy for White Noise

Let $x(t)$ be a zero-mean Gaussian white-noise random process with average intensity k. The Fourier transform of $x(t)$ is

$$X(\omega) = A(\omega) + iB(\omega) \quad (88)$$

Since the Fourier transform is a linear operation both $A(\omega)$ and $B(\omega)$ are zero-mean Gaussian white-noise random processes.

The squared-output of a sine-phase Gabor filter convolved with $x(t)$ is given by

$$\int_{-\infty}^{\infty} |G_s(\omega_0,\sigma) * x(t)|^2\,dx = \int_{-\infty}^{\infty} |\mathcal{F}\{G_s(\omega_0,\sigma)\}\mathcal{F}\{x(t)\}|^2\,d\omega \quad (89)$$

$$\begin{aligned}
&= \int_{-\infty}^{\infty} |(i/2)A(\omega)g(\omega)-(i/2)A(\omega)f(\omega)-(1/2)B(\omega)g(\omega)+(1/2)B(\omega)f(\omega)|^2\,d\omega \\
&= (1/4)\int_{-\infty}^{\infty} [f(\omega)-g(\omega)]^2[A^2(\omega)+B^2(\omega)]d\omega
\end{aligned}$$

where $f(\omega)$ and $g(\omega)$ are defined above in Equation (84).

Similarly, the squared-output of a cosine-phase Gabor filter convolved with $x(t)$ is given by

$$\begin{aligned}\int_{-\infty}^{\infty} |G_c(\omega_0,\sigma) * x(t)|^2\, dx &= \int_{-\infty}^{\infty} |\mathcal{F}\{G_c(\omega_0,\sigma)\}\mathcal{F}\{x(t)\}|^2\, d\omega \qquad (90)\\ &= \int_{-\infty}^{\infty} |(1/2)A(\omega)g(\omega) + (1/2)A(\omega)f(\omega) + (i/2)B(\omega)g(\omega) + (i/2)B(\omega)f(\omega)|^2\, d\omega\\ &= (1/4)\int_{-\infty}^{\infty} [f(\omega)+g(\omega)]^2[A^2(\omega)+B^2(\omega)]d\omega\end{aligned}$$

The sum of the squared outputs of sine- and cosine-phases is

$$\mathcal{R}(\omega) = (1/2)\int_{-\infty}^{\infty} [f^2(\omega)+g^2(\omega)][A^2(\omega)+B^2(\omega)]d\omega \qquad (91)$$

The expected value of $\mathcal{R}(\omega)$ in Equation (91) is

$$\begin{aligned}&\mathrm{E}\left\{(1/2)\int_{-\infty}^{\infty} [f^2(\omega)+g^2(\omega)][A^2(\omega)+B^2(\omega)]d\omega\right\} \qquad (92)\\ &\qquad = (1/2)\int_{-\infty}^{\infty} [f^2(\omega)+g^2(\omega)]\mathrm{E}\left\{[A^2(\omega)+B^2(\omega)]\right\} d\omega\\ &\qquad = (k^2/2)\int_{-\infty}^{\infty} [f^2(\omega)+g^2(\omega)]d\omega\\ &\qquad = k^2\sqrt{2\pi}\sigma\end{aligned}$$

where k^2 is the average intensity of the white noise and σ is the Gaussian window size of the Gabor filter.

The covariance of two Gabor energy filter outputs is

$$\mathrm{cov}\,[\mathcal{R}_1(v),\mathcal{R}_2(\nu)] = \mathrm{E}\,\{\mathcal{R}_1(v)\mathcal{R}_2(\nu)\} - \mathrm{E}\,\{\mathcal{R}_1(v)\}\,\mathrm{E}\,\{\mathcal{R}_2(\nu)\} \qquad (93)$$

where

$$\begin{aligned}\mathcal{R}_1(v) &= (1/2)\int_{-\infty}^{\infty} [f_1^2(v)+g_1^2(v)][A^2(v)+B^2(v)]dv\\ \mathcal{R}_2(\nu) &= (1/2)\int_{-\infty}^{\infty} [f_2^2(\nu)+g_2^2(\nu)][A^2(\nu)+B^2(\nu)]d\nu\\ f_1(v) &= \exp[-2\pi^2\sigma^2(v-\omega_1)^2]\\ g_1(v) &= \exp[-2\pi^2\sigma^2(v+\omega_1)^2]\\ f_2(\nu) &= \exp[-2\pi^2\sigma^2(\nu-\omega_2)^2]\\ g_2(\nu) &= \exp[-2\pi^2\sigma^2(\nu+\omega_2)^2]\end{aligned}$$

where σ is the Gaussian window size of the Gabor filters and ω_1 and ω_2 are their center frequencies.

Simplifying gives

$$\begin{aligned} \mathrm{cov}\,[\mathcal{R}_1(v), \mathcal{R}_2(\nu)] &= (1/4)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[f_1^2(v)+g_1^2(v)][f_2^2(\nu)+g_2^2(\nu)]\Phi(v,\nu)dvd\nu \quad (94) \\ \Phi(v,\nu) &= \mathrm{E}\left\{[A^2(v)+B^2(v)][A^2(\nu)+B^2(\nu)]\right\} \\ &\quad -\mathrm{E}\left\{[A^2(v)+B^2(v)]\right\}\mathrm{E}\left\{[A^2(\nu)+B^2(\nu)]\right\} \end{aligned}$$

From [106, page 307] we know that

$$\Phi(v,\nu) = k^2[\delta(v+\nu)+\delta(v-\nu)] \quad (95)$$

i.e.,

$$\begin{aligned} \mathrm{cov}\,[\mathcal{R}_1(v), \mathcal{R}_2(v)] &= (k^2/4)\int_{-\infty}^{\infty}[f_1^2(v)+g_1^2(v)][f_2^2(-v)+g_2^2(-v)]dv \quad (96) \\ &\quad +(k^2/4)\int_{-\infty}^{\infty}[f_1^2(v)+g_1^2(v)][f_2^2(v)+g_2^2(v)]dv \\ &= (k^2/2)\int_{-\infty}^{\infty}[f_1^2(v)+g_1^2(v)][f_2^2(v)+g_2^2(v)]dv \end{aligned}$$

Appendix C

Motion-Energy Sensor Model

This appendix formultes a sensor model to characterize the variability in the motion energy measurements for a translating random texture. As discussed in Section 3.2, I posit an additive Gausian model for the variability in the motion energy measurements.

The image flow model estimates velocity utilizing the motion energy measurements by minimizing

$$\begin{aligned} l(u,v) &= \sum_{i=1}^{12} \left[m_i - \hat{K}_i \mathcal{R}_i(u,v) \right]^2 \\ \hat{K}_i &= \frac{\overline{m}_i}{\overline{\mathcal{R}}_i(u,v)} \end{aligned} \tag{97}$$

where

$$\begin{aligned} \overline{m}_i &= m_i + m_1 + m_2 \\ \overline{\mathcal{R}}_i(u,v) &= \mathcal{R}_i(u,v) + \mathcal{R}_1(u,v) + \mathcal{R}_2(u,v) \end{aligned} \tag{98}$$

where m_i is the output of the ith filter, m_1 and m_2 are the outputs of the two filters that share the same orientation the ith filter, and $\mathcal{R}_i(u,v)$, $\mathcal{R}_1(u,v)$ and $\mathcal{R}_2(u,v)$ are the corresponding predicted motion energies given by Equation (9).

The variance of $\left[m_i - \hat{K}_i \mathcal{R}_i(u,v) \right]$ is given by

$$\begin{aligned} \sigma_i^2(u,v) &= \text{var}\left(m_i - \overline{m}_i \frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)} \right) \\ &= \left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}}_i(u,v)} - 1 \right)^2 \text{var}(m_i) \end{aligned} \tag{99}$$

$$+\left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}_i}(u,v)}\right)^2 [\mathrm{var}(m_2)+\mathrm{var}(m_3)]$$
$$+2\left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}_i}(u,v)}-1\right)\left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}_i}(u,v)}\right)[\mathrm{cov}(m_i,m_2)+\mathrm{cov}(m_i,m_3)]$$
$$+2\left(\frac{\mathcal{R}_i(u,v)}{\overline{\mathcal{R}_i}(u,v)}\right)^2 [\mathrm{cov}(m_2,m_3)]$$

where $\mathrm{cov}(m_i,m_j)$ is the covariance of two the motion energy measurements and $\mathrm{var}(m_i) = \mathrm{cov}(m_i,m_i)$.

Equation (96) in Appendix B expresses the covariance of the outputs of two one-dimensional Gabor-energy filters, each convolved with a one-dimensional Gaussian white-noise signal. Analogously, for a translating two-dimensional Gaussian white-noise random field we get

$$\mathrm{cov}(m_1,m_2) = (k^2/4)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[f_1^2(\omega_x,\omega_y,u,v)+g_1^2(\omega_x,\omega_y,u,v)] \qquad (100)$$
$$[f_2^2(\omega_x,\omega_y,u,v)+g_2^2(\omega_x,\omega_y,u,v)]d\omega_x d\omega_y$$
$$f_1(\omega_x,\omega_y,u,v) = (1/4)\exp\{-2\pi^2[\sigma_x^2(\omega_x-\omega_{x_1})^2+\sigma_y^2(\omega_y-\omega_{y_1})^2$$
$$+\sigma_t^2(u\omega_x+v\omega_y-\omega_{t_1})^2]\}$$
$$g_1(\omega_x,\omega_y,u,v) = (1/4)\exp\{-2\pi^2[\sigma_x^2(\omega_x+\omega_{x_1})^2+\sigma_y^2(\omega_y+\omega_{y_1})^2$$
$$+\sigma_t^2(u\omega_x+v\omega_y+\omega_{t_1})^2]\}$$
$$f_2(\omega_x,\omega_y,u,v) = (1/4)\exp\{-2\pi^2[\sigma_x^2(\omega_x-\omega_{x_2})^2+\sigma_y^2(\omega_y-\omega_{y_2})^2$$
$$+\sigma_t^2(u\omega_x+v\omega_y-\omega_{t_2})^2]\}$$
$$g_2(\omega_x,\omega_y,u,v) = (1/4)\exp\{-2\pi^2[\sigma_x^2(\omega_x+\omega_{x_2})^2+\sigma_y^2(\omega_y+\omega_{y_2})^2$$
$$+\sigma_t^2(u\omega_x+v\omega_y+\omega_{t_2})^2]\}$$

where k is proportional to image contrast, $(\omega_{x_i},\omega_{y_i},\omega_{t_i})$ is the center frequency of each of the filters, $(\sigma_x,\sigma_y,\sigma_t)$ is the spread of the filters' spatiotemporal Gaussian windows, and (u,v) is the velocity of the translating pattern.

The integral of Equation (100) is easily evaluated. Together, equations (99) and (100) are a sensor model for the motion energy measurements. To estimate $\sigma_i^2(u,v)$ we use $(\hat{u},\hat{v})$ provided by the image flow model as estimates of (u,v) and we use the average $\hat{K}_i$ as an estimate for k,

$$k \approx (1/12)\sum_{i=1}^{12}\hat{K}_i \qquad (101)$$

	$\vec{\theta} = (0.25, 0.25)$	$\vec{\theta} = (0.5, 0.5)$	$\vec{\theta} = (1.0, 1.0)$
rt	1.03	1.25	1.15
lt	1.05	1.36	1.43
st1	1.03	1.25	1.15
up	1.09	1.05	1.21
dn	1.10	1.18	1.58
st2	1.10	1.04	1.21
ur	1.03	0.95	1.17
dl	1.05	1.07	-
st3	1.03	0.95	1.31
ul	1.00	1.11	1.06
dr	1.01	1.01	1.02
st4	1.13	0.99	0.98

Table 1: Motion-energy measurement data were obtained from Gaussian white-noise random texture motion sequences moving upward and rightward with three different speeds. The actual variances of $[m_i - \hat{K}_i \mathcal{R}_i(u,v)]$ were computed from the data. The sensor model was used to simulate these variances. The table gives the ratio of the actual to the simulated values for each of the twelve motion energies. A ratio less than one indicates that the simulated values overestimate the variance. There is no table entry for the filter most sensitive to down-left motion at velocity $(1.0, 1.0)$ since the output of that filter is essentially zero for that velocity. The average percent error in the variance estimates is 17.7%.

Table 1 shows empirical tests of the accuracy of the sensor model. Motion-energy measurement data were obtained from Gaussian white-noise random-texture motion sequences at three different velocities. The actual variances were computed from the data. Equation (100) was used to simulate $\text{var}(m_i)$ and $\text{cov}(m_i, m_j)$. Equation (99) was then used to estimate $\sigma_i^2(u,v) = \text{var}[m_i - \hat{K}_i \mathcal{R}_i(u,v)]$. The table gives the ratio of the actual to the estimated values for each of the twelve motion energies. The average percent error in the variance estimates is 17.7%. So there is reasonably good agreement in the table between the actual and simulated measurement variability for translating Gaussian white-noise textures.

Appendix D

Mahalanobis Distance

The Appendix reviews maximum-likelihood estimation and derives Mahalanobis distance for a one-dimensional parameter space.

Consider an example in which we have two sensors, each providing noisy observations of some parameter θ. If the noise in each sensor is additive zero-mean Gaussian, then each observation $\tilde{\theta}_i$ $(i = 1 - 2)$, is given by

$$\begin{aligned} \tilde{\theta}_i &= \theta + n_i \\ n_i &\sim N(0, \sigma_i^2) \end{aligned} \tag{102}$$

We may therefore write the probablity density for the estimate from the first sensor as

$$f_1(\tilde{\theta}_1|\theta) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left\{\frac{-(\tilde{\theta}_1 - \theta)^2}{2\sigma_1^2}\right\} \tag{103}$$

and that of the estimate from the second sensor as

$$f_2(\tilde{\theta}_2|\theta) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left\{\frac{-(\tilde{\theta}_2 - \theta)^2}{2\sigma_2^2}\right\} \tag{104}$$

The maximum likelihood estimate (MLE), $\hat{\theta}$, is the one that simultaneously maximizes both of these probability densities, i.e., it maximizes

$$f(\theta) = f_1(\tilde{\theta}_1|\theta) f_2(\tilde{\theta}_2|\theta) \tag{105}$$

Equivalently, it maximizes the log-likelihood function:

$$l(\theta) = \log[f(\theta)] = -\left[\frac{(\tilde{\theta}_1 - \theta)^2}{2\sigma_1^2} + \frac{(\tilde{\theta}_2 - \theta)^2}{2\sigma_2^2}\right] \tag{106}$$

This maximum is found by taking the derivative and setting it equal to zero

$$\sigma_2^2(\tilde{\theta}_1 - \hat{\theta}) + \sigma_1^2(\tilde{\theta}_2 - \hat{\theta}) = 0 \tag{107}$$

giving

$$\hat{\theta} = \frac{\tilde{\theta}_1\sigma_2^2 + \tilde{\theta}_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{108}$$

For a two-dimensional parameter, $\vec{\theta} = (u, v)^T$, each sensor provides estimates $\tilde{\theta}_i = (u_i, v_i)^T$ and the additive Gaussian noise is characterized by variance-covariance matrices of the form

$$\Lambda_i = \begin{pmatrix} \sigma_{u_i}^2 & \sigma_{u_i v_i} \\ \sigma_{u_i v_i} & \sigma_{v_i}^2 \end{pmatrix} \tag{109}$$

The MLE obtained by combining the two estimates is

$$\hat{\theta} = \left(\Lambda_1^{-1} + \Lambda_2^{-1}\right)^{-1}\left(\Lambda_1^{-1}\tilde{\theta}_2 + \Lambda_2^{-1}\tilde{\theta}_1\right) \tag{110}$$

Maximum likelihood is one way of combining information from two sensors. But, we want to combine information from different sensors only if they *concur* with one another. Mahalanobis distance is a test for consistency between sensor observations. Let

$$g(\theta) = f_1(\tilde{\theta}_1|\theta) + f_2(\tilde{\theta}_2|\theta) \tag{111}$$

where $f_1(\tilde{\theta}_1|\theta)$ and $f_2(\tilde{\theta}_2|\theta)$ are normal densities as above. Hager and Durrant-Whyte [57] argue that the two observations form a consensus only if the superposition of the two sensor observations is unimodal, i.e., only if there exists a θ such that $\frac{\partial^2 f_i(\tilde{\theta}_1|\theta)}{\partial\theta^2}|_\theta \leq 0$ for each i,

$$\frac{\partial^2 f_i(\tilde{\theta}_1, \theta)}{\partial\theta^2} = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left\{\frac{-(\tilde{\theta}_i - \theta)^2}{2\sigma_i^2}\right\}\frac{1}{\sigma_i^2}\left[\frac{(\tilde{\theta}_i - \theta)^2}{\sigma_i^2} - 1\right] \leq 0 \tag{112}$$

i.e.,

$$\frac{(\tilde{\theta}_i - \theta)^2}{\sigma_i^2} \leq 1 \tag{113}$$

Since the left hand side of this Equation is always positive we must find a θ that satisfies

$$(1/2)\frac{(\tilde{\theta}_1 - \theta)^2}{\sigma_1^2} + (1/2)\frac{(\tilde{\theta}_2 - \theta)^2}{\sigma_2^2} \leq 1 \tag{114}$$

The value of θ that makes the left hand side of this equation a minimum is the maximum likelihood estimate given in Equation (108) above. Substituting Equation (108) for θ in Equation (114) gives the Mahalanobis distance,

$$M(\tilde{\theta}_1, \tilde{\theta}_2, \sigma_1^2, \sigma_2^2) = (1/2)\frac{(\tilde{\theta}_1 - \tilde{\theta}_2)^2}{(\sigma_1^2 + \sigma_2^2)} \tag{115}$$

Mahalanobis distance is the distance between the two observations from relative to (weighted by) the noise in each of the observations. If the Mahalanobis distance is less than some fixed threshold (say, 1), then we will say that the two sensors form a consensus, and we may combine the information from the two sensors to calculate a single best estimate for θ.

For a two-dimensional parameter, $\vec{\theta} = (u, v)^T$, each sensor provides observations $\tilde{\theta}_i = (u_i, v_i)^T$ and the additive Gaussian noise is characterized by variance-covariance matrices given by Equation (109). The Mahalanobis distance is

$$M(\tilde{\theta}_1, \tilde{\theta}_2, \Lambda_1, \Lambda_2) = (1/2)\left(\tilde{\theta}_1 - \tilde{\theta}_2\right)^T \Lambda_1^{-1}\left(\Lambda_1^{-1} + \Lambda_2^{-1}\right)^{-1}\Lambda_2^{-1}\left(\tilde{\theta}_1 - \tilde{\theta}_2\right) \tag{116}$$

Appendix E

Eye Movements

Eye/camera movements have only rotational two degrees of freedom, pan and tilt, about axes that pass through the center of projection. Each camera orientation is associated with a different rotated coordinate frame. Rotation matrices specify the transform from one coordinate frame to another. In this Appendix, I derive formulas for transforming position, velocity and angular velocity from one frame to another. I explain how to to fixate on a surface point, how to use image velocity information to track moving surface points over time, and how to warp an image to simulate the effect of an eye/camera movement.

E.1 Camera Orientation and Fixation

The orientation of a camera may be expressed as a coordinate transformation with respect to a base coordinate frame in either of two ways. First, the orientation may be specified as a rotation θ about an axis $\vec{k} = (k_x, k_y, k_z)$, giving the rotation matrix [108]:

$$\begin{aligned} \mathbf{A} &= \mathrm{Rot}(\vec{k}, \theta) \qquad (117) \\ &= \begin{pmatrix} k_x k_x \mathrm{vers}(\theta) + \cos(\theta) & k_y k_x \mathrm{vers}(\theta) - k_z \sin(\theta) & k_z k_x \mathrm{vers}(\theta) + k_y \sin(\theta) \\ k_x k_y \mathrm{vers}(\theta) + k_z \sin(\theta) & k_y k_y \mathrm{vers}(\theta) + \cos(\theta) & k_z k_y \mathrm{vers}(\theta) - k_x \sin(\theta) \\ k_x k_z \mathrm{vers}(\theta) - k_y \sin(\theta) & k_y k_z \mathrm{vers}(\theta) + k_x \sin(\theta) & k_z k_z \mathrm{vers}(\theta) + \cos(\theta) \end{pmatrix} \end{aligned}$$

in which $\|\vec{k}\| = 1$ and $\mathrm{vers}(\theta) = [1 - \cos(\theta)]$. Since camera orientation has only two rotational degrees of freedom $k_z = 0$.

The second way to specify camera orientation is as a rotation ϕ_y about y-axis followed by rotation ϕ_x about x-axis, giving the rotation matrix:

$$\begin{aligned}
\mathbf{A} &= \mathrm{Rot}_y(\phi_y)\mathrm{Rot}_x(\phi_x) \qquad (118)\\
&= \begin{pmatrix} \cos(\phi_y) & 0 & \sin(\phi_y) \\ 0 & 1 & 0 \\ -\sin(\phi_y) & 0 & \cos(\phi_y) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_x) & -\sin(\phi_x) \\ 0 & k_x \sin(\phi_x) & \cos(\phi_x) \end{pmatrix} \\
&= \begin{pmatrix} \cos(\phi_y) & 0 & \sin(\phi_y) \\ \sin(\phi_x)\sin(\phi_y) & \cos(\phi_x) & -\sin(\phi_x)\cos(\phi_y) \\ -\cos(\phi_x)\sin(\phi_y) & \sin(\phi_x) & \cos(\phi_x)\cos(\phi_y) \end{pmatrix}
\end{aligned}$$

In order to fixate upon image position (x, y) we may rotate the camera, shifting (x, y) to $(0, 0)$. Let us express the angles ϕ_x and ϕ_y in terms of image location (x, y), and focal length f. From the geometry of perspective projection it is clear that:

$$\begin{aligned}
\phi_x &= \tan^{-1}(x/f) \qquad (119)\\
\phi_y &= \tan^{-1}(y/f)
\end{aligned}$$

where f is the focal length. It is important to note that we do *not* need depth information in order to fixate on a surface point.

We may also express the axis (k_x, k_y) and the angle θ in terms of image location (x, y), and focal length f,

$$\begin{aligned}
\cos(\theta) &= \frac{f}{\sqrt{x^2 + y^2 + f^2}} \qquad (120)\\
\sin(\theta) &= \frac{\sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2 + f^2}} \\
k_x &= \frac{-y}{\sqrt{x^2 + y^2}} \\
k_y &= \frac{x}{\sqrt{x^2 + y^2}}
\end{aligned}$$

E.2 Transforming Between Camera Orientations

To transform the position of an image point in the new coordinate system into base coordinates (unrotated coordinate frame), we simply multiply on the left by the rotation matrix:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{A} \begin{pmatrix} x' \\ y' \end{pmatrix} \tag{121}$$

where $(x, y)^T$ is the position in base coordinates and $(x', y')^T$ is the position in the rotated coordinate frame.

Since a rotation matrix, **A**, is orthonormal, its inverse is equal to its transpose

$$\mathbf{A}^{-1} = \mathbf{A}^T \tag{122}$$

Thus, to transform a point in base coordinates into the rotated coordinate frame, we simply multiply on the left by $\mathbf{A}^T$.

Paul [108] derives equations for transforming differential relations (e.g., differential rotations and translations) from one coordinate frame to another. Let the differential rotation in base coordinates be $\vec{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$, and the differential translation in base coordinates be $\vec{T} = (T_x, T_y, T_z)$, and let us represent the elements of the rotation matrix as

$$\mathbf{A} = \begin{pmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{pmatrix} \tag{123}$$

The differential rotations and translations in the rotated coordinate frame are computed by

$$\begin{pmatrix} t_x^A \\ t_y^A \\ t_z^A \\ \Omega_x^A \\ \Omega_y^A \\ \Omega_z^A \end{pmatrix} = \begin{pmatrix} n_x & n_y & n_z & 0 & 0 & 0 \\ o_x & o_y & o_z & 0 & 0 & 0 \\ a_x & a_y & a_z & 0 & 0 & 0 \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{pmatrix} \begin{pmatrix} T_x \\ T_y \\ T_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} \tag{124}$$

Since $\mathbf{A}^{-1} = \mathbf{A}^T$, transforming differential relationships from the rotated frame back to base coordinates is achieved by using A^T in the above equations.

E.3 Camera Movements and Tracking

A camera movement can be specified in either of two ways: (1) as a rotation with angular velocity Ω_θ about an arbitrary axis $\vec{k} = (k_x, k_y, 0)$; (2) as a rotation about the x-axis with angular velocity Ω_x coupled with a rotation about the y-axis with angular velocity Ω_y. In the former case, the new camera orientation at time t is given by $\mathrm{Rot}(\vec{k}, \Omega_\theta t)$ in Equation (117). In the latter case, the new camera orientation at time t is given by $\mathrm{Rot}_y(\Omega_y t)\mathrm{Rot}_x(\Omega_x t)$ in Equation (118).

In either case we have a rotation matrix at time each t to transform an image into each new coordinate frame. Applying each of these coordinate transforms results in a warping of the image over time simulating the effect of a camera movement. It is important to note that we do *not* need depth information in order to simulate camera movements.

Consider that we are fixating on a moving surface point at a particular time t, and that we know the image velocity at the fixation point for that time, $[u(t), v(t)]$. A camera movement to track the moving surface point is given by

$$\begin{aligned} \Omega_x &= u(t)/f \\ \Omega_y &= v(t)/f \end{aligned} \tag{125}$$

where f is focal length. As the surface point moves, we must continually remeasure the image velocity at the point of fixation and update the angular velocities.

Bibliography

[1] E H Adelson. Media-Technology Laboratory, MIT, personal communication.

[2] E H Adelson and J R Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299, 1985.

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

[4] E H Adelson and E Simoncelli. Orthogonal pyramid transforms for image coding. 1987. to appear in Proceedings of SPIE — Visual Communication and Image Processing, Cambridge, MA.

[5] G Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.

[6] P Allen. *Object Recognition Using Vision and Touch.* PhD thesis, Computer and Information Science Department, University of Pennsylvania, 1985.

[7] P Allen and R Bajcsy. *Two sensors are better than one: Example of integration of vision and touch.* Technical Report MS-CIS-85-29, University of Pennsylvania, Computer and Information Science Department, 1985.

[8] J Aloimonos, I Weiss, and A Bandyopadhyay. Active vision. In *Proceedings of the first International Conference on Computer Vision*, pages 35–54, London, June 1987. to appear in Internation Journal of Computer Vision.

[9] R Aris. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1962.

[10] N Badler. *Temporal scene analysis: Conceptual descriptions of object movements.* Technical Report 80, Department of Computer Science, University of Toronto, 1975.

[11] R Bajcsy. *Active perception vs. passive perception.* Technical Report MS-CIS-85-54, University of Pennsylvania, Computer and Information Science Department, 1985.

[12] R Bajcsy and F Solina. Three dimensional shape representation revisited. In *Proceedings of the first International Conference on Computer Vision*, pages 231–240, London, June 1987. also available as University of Pennsylvania Computer and Information Science Department technical report MS-CIS-87-19.

[13] D H Ballard and O A Kimball. Rigid body motion from depth and optical flow. *Computer Vision, Graphic and Image Processing*, 95:95–115, 1983.

[14] A Bandopadhay. *Constraints on the computation of rigid motion parameters from retinal displacements.* Technical Report CAR-TR-134, University of Rochester, Department of Computer Science, 1985.

[15] A Bandopadhay, B Chandra, and D H Ballard. Active navigation: tracking an environmental point considered beneficial. In *Proceedings of Workshop on Motion: Representation and analysis*, pages 23–29, IEEE, Charleston, South Carolina, May 1986.

[16] A Bandopadhay and R Dutta. Measuring image motion in dynamic images. In *Proceedings of Workshop on Motion: Representation and analysis*, pages 67–72, IEEE, Charleston, South Carolina, May 1986.

[17] S T Barnard and W B Thomson. Disparity analysis of images. *IEEE Pattern Analysis and Machine Intelligence*, 2(4):333–340, 1980.

[18] J Barron. *A survey of approaches for determining optic flow, environmental layout and egomotion.* Technical Report RBCV-TR-84-5, Department of Computer Science, University of Toronto, 1984.

[19] J L Barron. The sensitivity of motion and structure computations. In *AAAI*, pages 700–705, Seattle, July 1987.

[20] H G Barrow and J M Tenenbaum. Recovering intrinsic scene characteristics from images. In A Hanson and E Riseman, editors, *Computer Vision Systems*, Academic Press, New York, 1978.

[21] J R Bergen. SRI/David Sarnoff Research Laboratory, personal communication.

[22] I Biederman. Human image understanding: recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 31:29–73, 1985.

[23] A Bobick and W Richards. *Classifying objects from visual information.* Technical Report 879, MIT AI memo, 1985.

[24] R C Bolles, H H Baker, and D H Marimont. Epipolar-plane image analysis: an approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.

[25] T J Broida and R Chellappa. Kinematics and structure of a rigid object from a sequence of noisy images. In *Proceedings of Workshop on Motion: Representation and analysis*, pages 95–100, IEEE, Charleston, South Carolina, May 1986.

[26] A R Bruss and B K P Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21:3–20, 1983.

[27] D C Burr and J Ross. Contrast sensitivity at high velocities. *Vision Research*, 22:479–484, 1982.

[28] P Burt. Fast algorithms for estimating local image properties. *Computer Vision, Graphics, and Image Processing*, 21:368–382, 1983.

[29] A Chorin. Estimates of intermittency, spectra, and blow-up in developed turbulence. *Communications on Pure and Applied Mathematics*, 34:853–866, 1981.

[30] W F Clocksin. Perepetion of surface slant and edge labels from optical flow: a computational approach. *Perception*, 9(3):253–269, 1980.

[31] P S Brown G D and Robinson. The variance spectrum of tropospheric winds over Eastern Europe. *Journal Atmospheric Science*, 36:270–286, 1979.

[32] J G Daugman. Two-dimensional analysis of cortical receptive field profiles. *Vision Research*, 20:846–856, 1980.

[33] J G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.

[34] M H DeGroot. *Probability and Statistics.* Addison-Wesley Publishing Co., Menlo Park, California, 1975.

[35] M P doCarmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1976.

[36] H F Durrant-Whyte. *Integration, coordination and control of multi-sensor robot systems.* PhD thesis, Computer and Information Science Department, University of Pennsylvania, 1987. available as technical report MS-CIS-86-67.

[37] M Fahle and T Poggio. Visual hyperacuity: spatiotemporal interpolation in human vision. *Proc. R. Soc. Lond.*, 213:451–477, 1981.

[38] O D Faugeras, F Lustman, and G Toscani. Motion and structure from motion from point and line matches. In *Proceedings of the first International Conference on Computer Vision*, pages 25–34, London, June 1987.

[39] M J Feigenbaum. Universal behavior in nonlinear systems. *Los Alamos Science*, 4–27, summer 1980.

[40] D J Fleet. *The early processing of spatio-temporal visual information.* Master's thesis, Department of Computer Science, University of Toronto, 1984. available as Technical Report RBCV-TR-84-7.

[41] 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.

[42] D J Fleet and A D Jepson. Velocity extraction without form interpretation. In *Proceedings of the Third Workshop on Computer Vision: Representation and Control*, pages 179–185, IEEE, Bellaire, Michigan, 1985.

[43] U Frisch, P Sulem, and M Nelkin. A simple dynamical model of intermittent fully developed turbulence. *Journal of Fluid Mechanics*, 87:719–736, 1978.

[44] D Gabor. Theory of communication. *J.IEE London*, 93:429–457, 1946.

[45] H Gafni and Y Zeevi. A model for processing of movement in the visual system. *Biological Cybernetics*, 32:165–173, 1979.

[46] H Gafni and Y Zeevi. A model for separation of spatial and temporal information in the visual system. *Biological Cybernetics*, 28:73–82, 1977.

[47] A Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, 1974.

[48] G L Gerstein and M R Turner. Neural assemblies as building blocks of cortical computation. to appear in Computational Neuroscience, E Schwartz, ed., MIT Press.

[49] E J Gibson and E S Spelke. The development of perception. In P H Mussen, editor, *Handbook of Child Psychology, Fourth Edition*, John Wiley and Sons, New York, 1984.

[50] J J Gibson. *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston, 1979.

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

[52] J J Gibson and E J Gibson. Continuous perspective transformations and the perception of rigid motions. *Journal of Experimental Psychology*, 54(2):129–138, 1957.

[53] P E Gill, W Murray, and M H Wright. *Practical Optimization.* Academic Press, New York, 1981.

[54] J Gleick. Solving the mathematical riddle of chaos. *The New York Times Magazine*, 31–, June 10 1984.

[55] B F Green. Figure coherence in the kinetic depth effect. *Journal of Experimental Psychology*, 62:272–282, 1961.

[56] N M Grzywacz and E C Hildreth. *The incremental rigidity scheme for recovering structure from motion: Position vs. velocity based formulations.* Technical Report 845, MIT AI Lab, 1895.

[57] G Hager and H F Durrant-Whyte. *Information and multi-sensor coordination.* Technical Report MS-CIS-86-68, University of Pennsylvania, Computer and Information Science Department, 1986. to appear in Uncertainty in Artificial Intelligence, edited by J Lemmer and T Kanal.

[58] Greg Hager. Searching for information. In *Proceedings of Workshop on Spatial Reasoning and Multi-sensor Fusion*, St Charles, IL, October 1987. available as University of Pennsylvania, Computer and Information Science Department technical report MS-CIS-87-71.

[59] R Hartley. Segmentation of optical flow fileds by pyramid linking. *Pattern Recognition Letters*, 3:253–262, 1985.

[60] J C Hay. Optical motions and space perception: an extension of Gibson's analysis. *Psychological Review*, 73(6):550–565, 1966.

[61] David J Heeger. Depth and flow from motion energy. In *AAAI 86*, pages 657–663, American Association of Artificial Intelligence, Philadelphia, August 1986.

[62] David J Heeger. Model for the extraction of image flow. *Journal of the Optical Society of America A*, 4(8):1455–1471, 1987. available as University of Pennsylvania Computer and Information Science Department Technical Report MS-CIS-87-04.

[63] David J Heeger and Alex P Pentland. Measurement of fractal dimension using gabor filters. in preparation.

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

[65] D D Hoffman. Inferring local surface orientation from motion fields. *Journal of the Optical Society of America A*, 72(7):888–892, 1982.

[66] D D Hoffman and B M Bennett. Inferring the relative three-dimensional positions for two moving points. *Journal of the Optical Society of America A*, 2(2):350–353, 1985.

[67] D D Hoffman and B M Bennett. Visual representations: meaning and truth conditions. 1985. in press.

[68] D D Hoffman and B E Flinchbaugh. *The interpretation of biological motion.* Technical Report 608, MIT AI Lab, 1981.

[69] D D Hoffman and W A Richards. Parts of recognition. *Cognition*, 18:65–96, 1985.

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

[71] B K P Horn and E J Weldon. Computationally-efficient methods for recovering translational motion. In *Proceedings of the first International Conference on Computer Vision*, pages 2–11, London, June 1987.

[72] B K P Horn and E J Weldon. Robust direct methods for recovering motion. 1987. to appear in International Journal of Computer Vision.

[73] R A Hummel and S W Zucker. On the foundations of relaxing labelling processes. *IEEE Pattern Analysis and Machine Intelligence*, 5(3):267–287, 1983.

[74] L Jacobson and H Wechsler. Derivation of optical flow using a spatiotemporal-frequency approach. *Computer Vision, Graphics, and Image Processing*, 38:29–65, 1987.

[75] R Jain. Direct computation of the focus of expansion. *IEEE Pattern Analysis and Machine Intelligence*, 5(1):58–63, 1983.

[76] R Jain, W N Martin, and J K Aggarwal. Segmentation through the detection of changes due to motion. *Computer Graphics and Image Processing*, 11:13–34, 1979.

[77] M R M Jenkin. *The stereopsis of time-varying images.* Technical Report RBCV-TR-84-3, University of Toronto, Department of Computer Science, 1984.

[78] G Johansson. Visual motion perception. *Scientific American*, 232:76–88, 1975.

[79] J K Kearney and W B Thompson. An error analysis of gradient-based methods for optical flow estimation. *IEEE Pattern Analysis and Machine Intelligence*, 19(2):229–244, 1987.

[80] D H Kelly. Motion and vision II. Stabilized spatio-temporal threshold surface. *Journal of the Optical Society of America A*, 69(10):1340–1349, 1979.

[81] J J Koenderink and A J van Dorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791, 1975.

[82] J J Koenderink and A J van Dorn. Local structure of movement parallax of the plane. *Journal of the Optical Society of America A*, 66(7):717–723, 1975.

[83] K Koffka. *Priciples of Gestalt Psychology.* Harcourt-Brace, New York, 1935.

[84] E P Krotkov. *Exploratory visual sensing for determining spatial layout with an agile stereo camera system.* PhD thesis, Computer and Information Science Department, University of Pennsylvania, 1987. available as technical report MS-CIS-87-29.

[85] P Kube and A P Pentland. On the imaging of fractal surfaces. 1987. to appear in IEEE Pattern Anylysis and Machine Intelligence.

[86] D T Lawton. Processing translational motion sequences. *Computer Graphics and Image Processing*, 116–144, 1983.

[87] M Leyton. Generative systems of analyzers. *Computer Vision, Graphics, and Image Processing*, 31:201–241, 1985.

[88] H C Longuet-Higgins and K Prazdny. The interpretation of a moving retinal image. *Proc. Roy. Soc. Lond. B*, 208:385–397, 1980.

[89] S Lovejoy and B B Mandlebrot. Fractal properties of rain, and a fractal model. *TELLUS*, 37A:209–232, 1985.

[90] S Lovejoy and D Schertzer. Scale invariance, symmetries, fractals, and stochastic simulations of atomospheric phenomena. *Bulletin of the American Meteorological Society*, 67(1):21–32, 1986.

[91] D G Lowe. *Three-dimensional object recognition from single two-dimensional images.* Technical Report 202, Courant Institute of Mathematical Sciences, NYU, 1986.

[92] S G Mallat. *A compact multiresolution representation: the wavelet model.* Technical Report MS-CIS-87-113, University of Pennsylvania Computer and Information Science Department, 1987.

[93] S G Mallat. Scale change versus scale space representation. In *Proceedings of the First International Conference on Computer Vision*, pages 592–596, IEEE, London, 1987. also available as University of Pennsylvania, Computer and Information Science Department technical report MS-CIS-87-22, and submitted to IEEE Pattern Analysis and Machine Intelligence.

[94] S W Mallat. Computer and Information Science Department, University of Pennsylvania, personal communication.

[95] B B Mandlebrot. *The Fractal Geometry of Nature.* W.H. Freeman and Co., New York, N.Y., 1983.

[96] D Marr. *Vision.* W. H. Freeman and Co., San Francisco, 1982.

[97] D Marr and E Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London*, B207:187–217, 1980.

[98] S P McKee. A local mechanism for differential velocity detection. *Vision Research*, 21:491–500, 1981.

[99] S P McKee, G H Silverman, and K Nakayama. Precise velocity discrimination despite random variations in temporal frequency and contrast. *Vision Research*, 26(4):609–619, 1986.

[100] J Melsa and D Cohn. *Decision and Estimation Theory.* McGraw-Hill Book Co., New York, 1978.

[101] A Mitiche. On combining stereopsis and kineopsis for space perception. In *Proceedings of the First Conference on Artificial Intelligence Applications*, pages 156–160, IEEE, Denver, December 1984.

[102] A Mitiche. On kineopsis and compuatation of structure and motion. *IEEE Pattern Analysis and Machine Intelligence*, 8(1):109–112, 1986.

[103] D W Murray and B F Buxton. Scene segmentation from visual motion using global optimization. *IEEE Pattern Analysis and Machine Intelligence*, 9:220–228, 1987.

[104] S Negahdaripour and B K P Horn. Direct passive navigation. 1986. to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence.

[105] J O'Rourke and N I Badler. Model-based image analysis of human motion using constraint propogation. *IEEE Pattern Analysis and Machine Intelligence*, 3(4):522–537, 1980.

[106] A Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Book Co., New York, 1986.

[107] P Parent and S W Zucker. *Trace inference, curvature consistency, and curve detection.* Technical Report CIM-86-3, Computer Vision and Robotics Laboratory, McGill University, 1986.

[108] R P Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, Mass, 1981.

[109] A P Pentland. Fractal-based description of natural scenes. *IEEE Pattern Analysis and Machine Intelligence*, 6(6):661–674, 1984.

[110] A P Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3), 1986. also available as SRI International AI center technical report 357.

[111] A P Pentland. Recognition by parts. In *Proceedings of the first International Conference on Computer Vision*, pages 612–620, London, June 1987.

[112] T Poggio, V Torre, and C Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 1985.

[113] K Prazdny. Egomotion and relative depth from optical flow. *Biological Cybernetics*, 102:87–102, 1980.

[114] D Regan and K I Beverley. Looming detectors in the human visual pathway. *Vision Research*, 18:415–421, 1978.

[115] D Regan and K I Beverley. Visual responses to vorticity and the neural analysis of optic flow. *Journal of the Optical Society of America A*, 2(2):280–283, 1985.

[116] D Regan and K I Beverley. Visually guided locomotion: psychophysical evidence for a neural mechanism sensitive to flow patterns. *Science*, 205:311–313, 1979.

[117] J H Reiger and D T Lawton. Processing differential image motion. *Journal of the Optical Society of America A*, 2(2):354–359, 1985.

[118] J W Roach and J K Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Pattern Analysis and Machine Intelligence*, 2(6):554–562, 1980.

[119] D E Rummelhart and J L McClelland, editors. *Parallel Distributed Processing: explorations in the microstructure of cognition.* MIT Press, Cambridge, Mass., 1986.

[120] G Sandini and M Tistarelli. Recovery of depth information: camera motion as an integration to stereo. In *Proceedings of Workshop on Motion: Representation and analysis*, pages 39–43, IEEE, Charleston, South Carolina, May 1986.

[121] A Singh. Computer Science Department, Columbia University, personal communication.

[122] M A Snyder. The accuracy of 3d parmeters in correspondence-based techniques: startup and updating. In *Proceedings of Workshop on Motion: Representation and analysis*, pages 53–59, IEEE, Charleston, South Carolina, May 1986.

[123] M E Spetsakis and J Aloimonos. Closed form solution to the structure from motion problem from line correspondences. In *AAAI*, pages 738–743, Seattle, July 1987.

[124] P S Stevens. *Patterns in Nature*. Atlantic Little, Brown Books, Boston, 1974.

[125] G Strang, editor. *Linear Algebra and its Applications*. Academic Press, New York, 1980.

[126] M Subbarao. *Interpretation of visual motion: A computational study*. Technical Report, Computer Vision Laboratory, Center for Automation Research, University of Maryland, 1986.

[127] D Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.

[128] D Thompson. *On Growth and Form*. The University Press, Cambridge, England, 1942.

[129] W B Thompson, K M Mutch, and V A Berzins. Dynamic occlusion analysis in optical flow fields. *IEEE Pattern Analysis and Machine Intelligence*, 4:374–383, 1985.

[130] W B Thompson and T C Pong. Motion and structure from motion from point and line matches. In *Proceedings of the first International Conference on Computer Vision*, pages 201–208, London, June 1987.

[131] R Y Tsai and T S Huang. *Estimating 3-D motion parameters of a rigid planar patch*. Technical Report R-922, University of Illinois, Coordinate Science Lab, 1981.

[132] R Y Tsai and T S Huang. *Uniqueness and estimation of 3-D motion parameters of rigid objects with curved surfaces*. Technical Report R-921, University of Illinois, Coordinate Science Lab, 1981.

[133] M R Turner. Texture discrimination by Gabor functions. *Biological Cybernetics*, 55:71–82, 1986.

[134] S Ullman. Maximizing rigidity: the incremental recovery of 3-D structure from rigid and rubbery motion. *Perception*, 13:255–274, 1984.

[135] S Ullman. Recent computational studies in the interpretation of structure and motion. In J Beck, B Hope, and A Rosenfeld, editors, *Human and Machine Vision*, Academic Press, New York, 1983.

[136] Shimon Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, Massachusetts, 1979.

[137] J P H van Santen and G Sperling. Elaborated Reichardt detectors. *Journal of the Optical Society of America A*, 2(2):300–321, 1985.

[138] R Voss. Random fractal forgeries. notes from a tutorial entitled 'Fractals: basic concepts, computation and rendering' given at SIGGRAPH (San Francisco), 1985.

[139] H Wallach and D N O'Connell. The kinetic depth effect. *Journal of Experimental Psychology*, 45(4):205–217, 1953.

[140] A B Watson and A J Ahumada. *A look at motion in the frequency domain.* Technical Report 84352, NASA-Ames Research Center, 1983.

[141] A B Watson and A J Ahumada. Model of human visual-motion sensing. *Journal of the Optical Society of America A*, 2(2):322–342, 1985.

[142] A M Waxman. An image flow paradigm. In *Proceedings of the Second IEEE Workshop on Computer Vision: Representation and Control*, pages 49–57, Anapolis, April 1984.

[143] A M Waxman and J H Duncan. Binocular image flows: steps toward stereo-motion fusion. *IEEE Pattern Analysis and Machine Intelligence*, 8:715–729, 1986.

[144] A M Waxman and S S Sinha. *Dynamic stereo: Passive ranging to moving objects from relative image flows.* Technical Report, Computer Vision Laboratory, Center for Automation Research, University of Maryland, 1984.

[145] A M Waxman and S Ullman. Surface structure and three-dimensional motion from image flow kinematics. *International Journal of Robotics Research*, 4(3):72–94, 1985.

[146] A M Waxman and K Wohn. Contour evolution, neighborhood deformation, and global image flow: planar surfaces in motion. *International Journal of Robotics Research*, 4(3):95–108, 1985.

[147] J A Webb and J K Aggarwal. Visually interpreting the motion of objects in space. *IEEE Computer*, 40–46, August 1981.

[148] M Wertheimer. Laws of organization in perceptual forms. In W D Ellis, editor, *A Source Book of Gestalt Psychology*, Harcourt Brace, New York, 1923.

[149] A P Witkin and J M Tenenbaum. On the role of structure in vision. In A P Pentland, editor, *From Pixels to Predicates*, pages 481–543, Ablex Publishing Co., Norwood, NJ, 1985.

[150] K Wohn and A M Waxman. *The analytic structure of image flows: Deformation and segmentation.* Technical Report, Computer Vision Laboratory, Center for Automation Research, University of Maryland, 1987. submitted to Computer Vision, Graphics, and Image Processing.

[151] K Wohn and A M Waxman. *Contour evolution, neighborhood deformation, and local image flow: curved surfaces in motion.* Technical Report CAR-TR-134, Computer Vision Laboratory, Center for Automation Research, University of Maryland, 1985. submitted to International Journal of Robotics Research.

[152] S W Zucker. *The diversity of perceptual grouping.* Technical Report TR-85-1R, Computer Vision and Robotics Laboratory, McGill University, 1985.

[153] S W Zucker and L Iverson. *From orientation selection to optical flow.* Technical Report CIM-86-2, Computer Vision and Robotics Laboratory, McGill University, 1986.