Robust Incremental Optical Flow

Michael Julian Black

YALEU/CSD/RR #923
September 1992

Abstract
Robust Incremental Optical Flow

Michael Julian Black
Yale University
1992

This thesis addresses the problem of recovering 2D image velocity, or optical flow, robustly over long image sequences. We develop a *robust estimation framework* for improving the reliability of motion estimates and an *incremental minimization framework* for recovering flow estimates over time.

Attempts to improve the robustness of optical flow have focused on detecting and accounting for motion discontinuities in the optical flow field. We show that motion discontinuities are one example of a more general class of model violations and that by formulating the optical flow problem as one of *robust estimation* the problems posed by motion discontinuities can be reduced, and the violations can be detected. Additionally, robust estimation provides a powerful framework for early vision problems that generalizes the popular "line process" approaches.

We formulate a *temporal continuity* constraint, which reflects the fact that the motion of a surface changes gradually over time. We exploit this constraint to develop a new incremental minimization framework and show how it is related to standard recursive estimation techniques. Within this framework we implement two incremental algorithms for minimizing non-convex objective functions over time; *Incremental Stochastic Minimization (ISM)* and *Incremental Graduated Non-Convexity (IGNC)*.

With this approach, motion estimates are always available, they are refined over time, the algorithm adapts to scene changes, and the amount of computation between frames is kept fixed. The psychophysical implications of temporal continuity are discussed and the power of the incremental minimization framework is demonstrated by extending image feature extraction over time.

Robust Incremental Optical Flow

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Michael Julian Black
December 1992

The evening mist
is the expressed patience
of morning haze.

James H. Black
(1947–1991)

Acknowledgements

I owe a great intellectual debt to Anandan; a man of wisdom and warmth, and to Drew McDermott who has an uncanny ability to find the weak link in a chain of reasoning. I am also greatful for the constant support and encouragement of David Heeger, my "West Coast advisor" and friend, and Greg Hager, for his insight and hallway conversations.

My collaboration with Michael Tarr has been both productive and fun. I thank Anand Rangarajan for his insights into line processes and robust estimation. This thesis has also benefited from discussions with my office mate, Sean Engelson, as well as from comments by Eric Mjolsness, and Ken Yip.

I would like to especially thank Beau Watson for providing a wonderful research environment, and my home away from home, at NASA Ames. I also want to thank the gang at NASA for lunch-time discussions and particularly Carlo Tiana for his support of all things technical. Also, Eero Simoncelli and Peet's coffee contributed greatly to the enjoyment of Ames.

I must also thank the Numerical Aerodynamic Simulation Program, at NASA Ames Research Center for providing access to a Connection Machine for a portion of my stay at NASA. And, I would especially like to thank Ted Adelson and Trevor Darrell for making the MIT Media Lab Connection Machine available to me, on a moment's notice, when things looked their darkest.

I would like to thank Joe Heel for kindly providing the "Pepsi can" image sequence and Bonivar Shridhar of the NASA Ames Research Center for providing the "Coke can" sequence.

My final thanks are reserved for the Teacher, the Writer, and the Muse.

This work was made possible, in part, by the generous support of the National Aeronautics and Space Administration (NASA Training Grant NGT–50749), the NASA Ames Research Center, Aerospace Human Factors Research Division (NASA RTOP 506–47), the Office of Naval Research (ONR Grant N00014–91–J–1577), the Whitaker Foundation, and the Defense Advanced Research Products Agency, contract number DAAA15–87–K–0001, administered by the Ballistic Research Laboratory.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

When we walk or drive or even move our heads, our view of the world changes. Even when we are at rest, the world around us may not be; objects fall, trees sway, and children run. Motion of this sort, and our understanding of it, seems so straightforward that we often take it for granted. In fact, this ability to understand a changing world is essential to survival; without it, there would be no continuity to our perceptions. If robots are to exist in the dynamic world of humans, as opposed to merely the factory floor, they too must possess this ability to understand motion.

What is required is a general and flexible representation of visual motion that can be used for many purposes and can be computed robustly and efficiently [Tarr and Black, 1991]. This thesis will consider *optical flow* as a representation of the apparent motion of the world projected on the image plane of a moving camera. Optical flow is the 2D velocity field, describing the apparent motion in the image, that results from independently moving objects in the scene or from observer motion. More specifically, consider the diagram in Figure 1.1 which illustrates how the translation and rotation of the camera cause the projected location p of a point P in the scene to move. Likewise, if point P is moving independently, its projection on the image plane will change, even when the camera is stationary. It is this vector field, $\mathbf{u}(x, y) = [u(x, y), v(x, y)]$, describing the horizontal and vertical image motion, that is to be recovered at every point in the image.

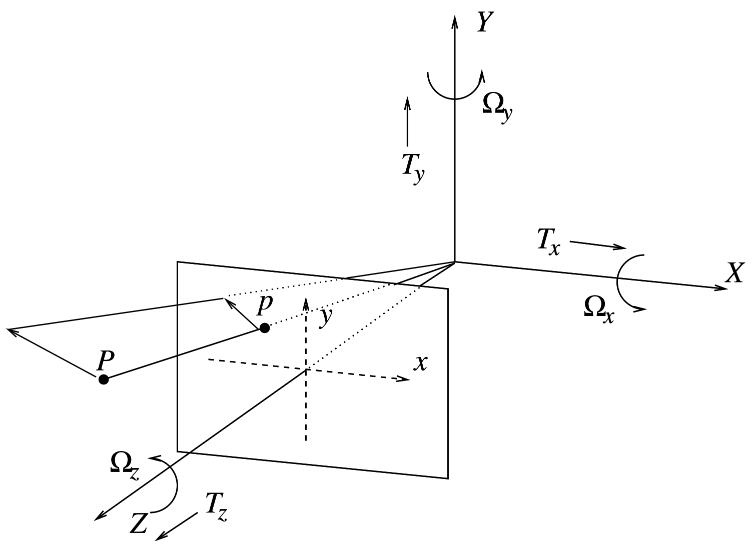


Figure 1.1: A point P in the scene projects to a point p in the $[x, y]$ coordinate system of the image plane of a camera centered at the origin of the camera coordinate system $[X, Y, Z]$, with its optical axis pointing in the Z direction. The motion of the camera is described by its translation $[T_X, T_Y, T_Z]$ and rotation $[\Omega_X, \Omega_Y, \Omega_Z]$.

One may ask, "What about the motion of a smooth surface like a smooth rotating sphere?" If the surface of the sphere is untextured then there will be no apparent motion on the image plane and hence no optical flow. This illustrates that the *motion field* [Horn, 1986], corresponding to the motion of points in the scene, is not always the same as the optical flow field. For most applications of optical flow, it is the motion field that is required and, typically, the world has enough structure that recovering optical flow provides a good approximation to the motion field. If this were not the case, then humans too, would not be able to exploit information about optical flow.

What motivates such a representation? Consider the example in Figure 1.2 which shows two images taken from a video camera mounted on a helicopter flying through a narrow ravine. The motion of the helicopter gives rise the optical flow field on the right. There is a wealth of information in this flow field and many uses have been proposed for robotics

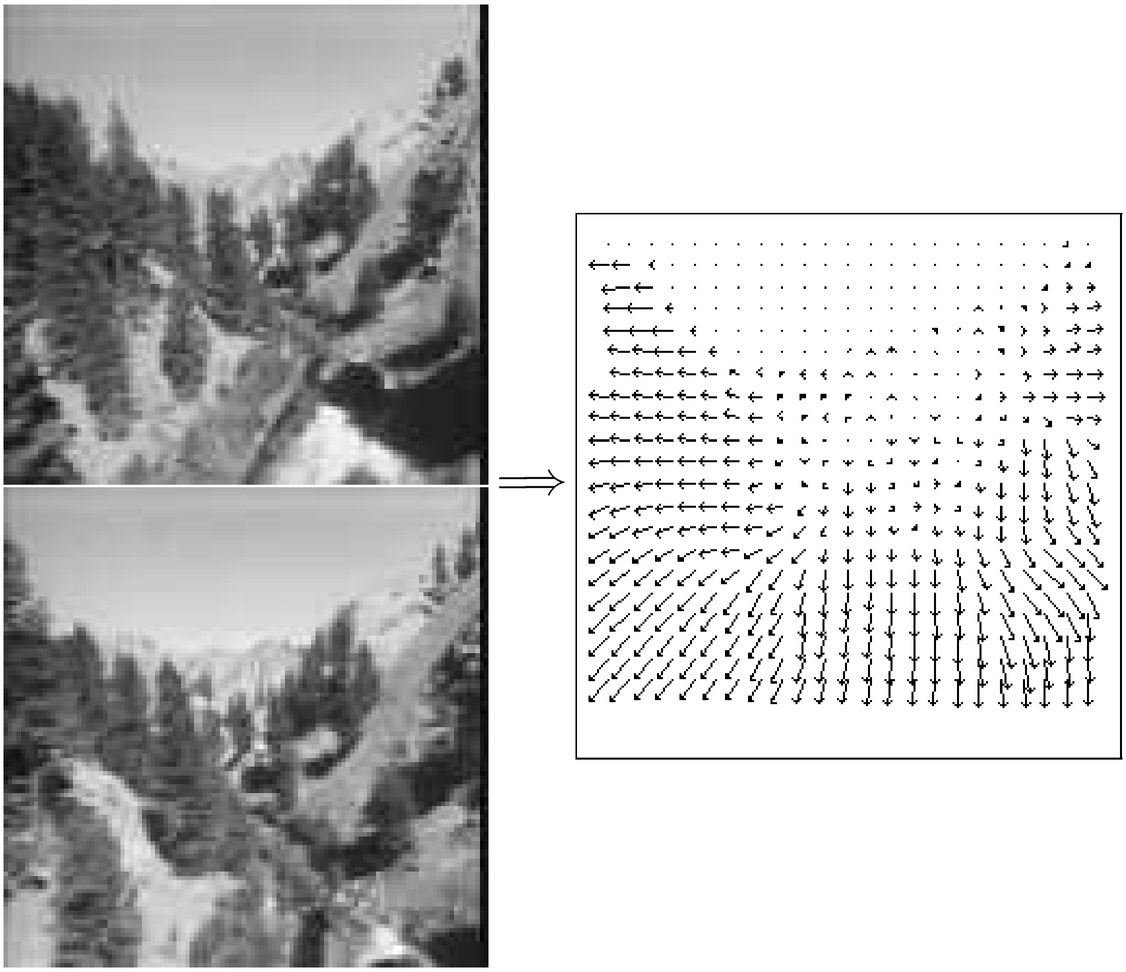


Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

and machine vision. For example, from this figure, we see that the flow vectors appear to emanate from a central point known as the *focus of expansion* [Gibson, 1979], and that points closer to the camera move more quickly across the image plane. Properties like this are thought to be important for biological vision systems [Gibson, 1979] and have been exploited in machine vision to recover observer motion [Lawton, 1983], detect obstacles [Ancona, 1992], avoid collisions [Nelson and Aloimonos, 1989], recover scene depth [Adiv, 1985], and track moving objects [Papanikolopoulos and Khosla, 1991]. There are other, non-robotic, applications of optical flow as well; particularly in the areas of medical imaging and image compression [Pratt, 1979].

There are still other properties of optical flow that can be exploited. Consider the image sequence in Figure 1.3 where a camera is translating parallel to the image plane. Notice that the flow field, in the top left of the figure, contains two distinct motions; the soda can is moving more rapidly than the background. This type of discontinuous flow field is the result of surfaces at different depths in the scene moving at different rates across the image plane, due to the effects of motion parallax or the independent motion of the objects. Since the location of these discontinuities in the flow field correspond to physically significant properties of the scene, they can be used to detect object boundaries [Black and Anandan, 1990a; Spoerri and Ullman, 1987; Thompson *et al.*, 1985; Thompson *et al.*, 1982] or segment the scene into distinct objects [Bouthemy and Rivero, 1987; Heitz and Bouthemy, 1990; Murray and Buxton, 1987; Peleg and Rom, 1990; Potter, 1980; Schunck, 1989a].

Motion can also be used to analyze the local relationship of surfaces at motion boundaries; in particular, whether surfaces are being *occluded* (covered) or *disoccluded* (revealed) [Black and Anandan, 1991b; Mutch and Thompson, 1988; Thompson *et al.*, 1985].[1] For example, the lower right of Figure 1.3 shows motion boundaries classified as occlusion boundaries (in white) or disocclusion boundaries (in black).

In computer vision, one is often interested in other properties of the scene that are unrelated to motion; for example, in the case of object recognition, it may be necessary to detect perceptually significant image properties like intensity edges (upper right of Figure 1.3). Motion and intensity information can be combined to improve the accuracy of motion segmentation [Black, 1992a; Gamble and Poggio, 1987; Heitz and Bouthemy, 1990; Thompson, 1980], and to distinguish between perceptual features that represent structural properties of the scene and those that are purely surface markings [Black, 1992a]. Additionally, if the optical flow is known, then traditionally static computation of image properties,

[1] Mutch and Thompson [1988] refer to these as regions of *accretion* or *deletion* respectively.

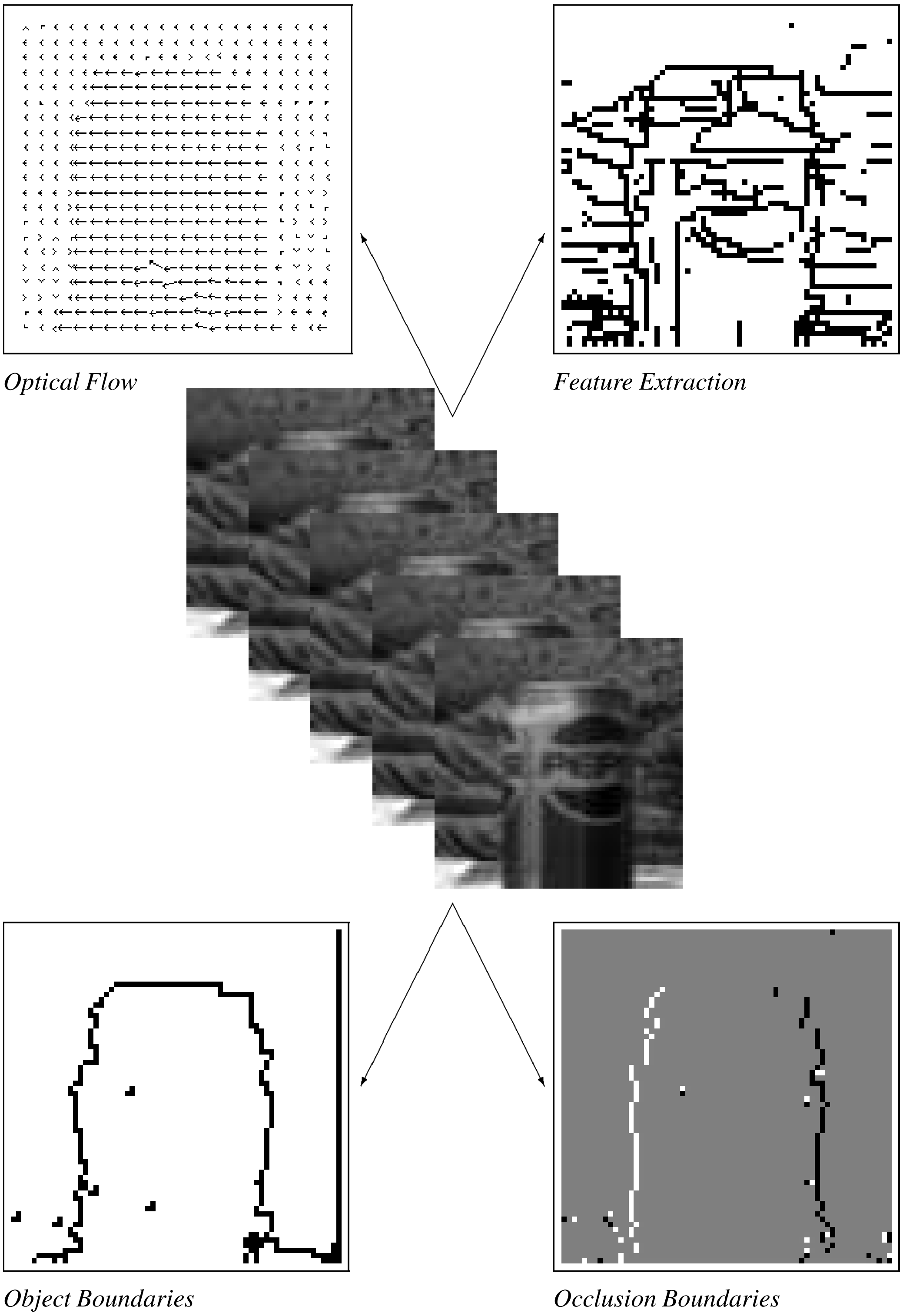


Figure 1.3: Optical flow can be used for a number of dynamic tasks; see text.

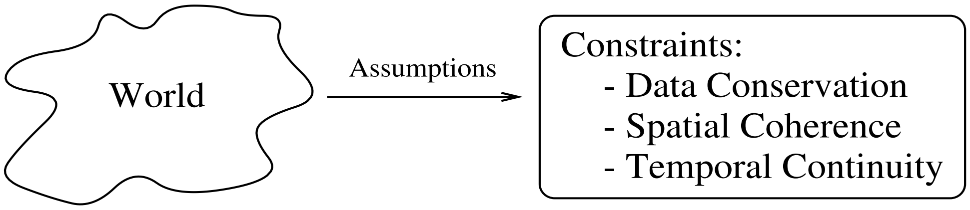


Figure 1.4: Constraints on image motion.

like intensity edges, can be made dynamic and extended over an image sequence.

The general problem of motion understanding, and in particular the computation of optical flow, has been one of the most intensely studied areas of computer vision. Despite rich mathematical foundations and steady progress, the results from years of computing and using optical flow have resulted in few practical applications. The failure of optical flow to live up to its promise may, in many cases, be attributed to a lack of robustness or to inefficiency. This thesis develops a framework for the *robust estimation* of optical flow to address the former, and exploits the constraint of *temporal continuity* to develop incremental algorithms designed to address the latter.

1.1 Constraints on Image Motion

This section considers the problem of recovering optical flow from image sequences. One begins by making some assumptions about the scene which by necessity are idealizations and will often be violated in practice. These assumptions are then embodied in a set of constraints on the interpretation of image motion as depicted in Figure 1.4. This thesis will explore the use of three such constraints: *data conservation*, *spatial coherence*, and *temporal continuity*.

Data Conservation

Algorithms for computing optic flow must somehow exploit the changes in image intensity over time. The most popular approaches include gradient-based techniques [Horn and

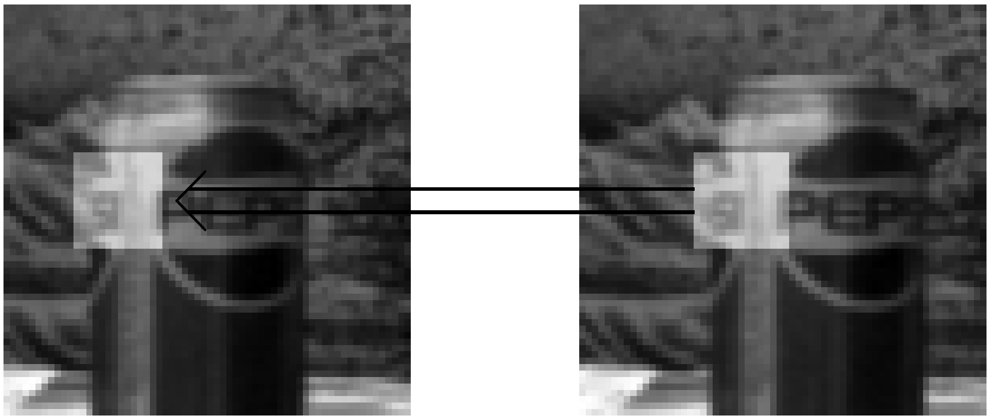


Figure 1.5: Data conservation assumption. The highlighted region in the right image looks roughly the same as the region in the left image, despite the fact that it has moved.

Schunck, 1981], correlation [Anandan, 1989], and spatio-temporal filtering [Heeger, 1987]. These approaches all exploit the assumption of data conservation[2]:

> *Image measurements (for example, image intensity) corresponding to a small image region remain the same, although the location of the region may change over time.*

That is, data is conserved as illustrated in Figure 1.5.[3]

Spatial Coherence

The data coherence constraint alone is not always sufficient to accurately recover optical flow. First, local motion estimates, based on data conservation, may only partially constrain the solution. Consider the motion of a line in Figure 1.6. Within a small region, the data conservation constraint cannot uniquely determine the motion of the line; an infinite number

[2] Also commonly referred to as the *intensity constancy assumption* [Horn, 1986].

[3] Notice that the highlighted region spans the boundary of the can and hence includes a portion of the background. This is a case for which the data conservation assumption does not hold. Constraint violations like this will be addressed in greater detail throughout the thesis.

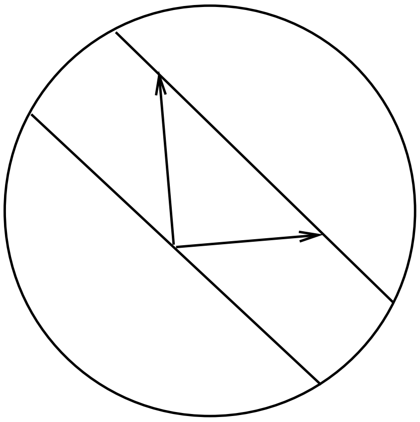


Figure 1.6: When viewed through a small aperture, the motion of a line is ambiguous.

of interpretations are consistent with the constraint. This is commonly referred to as the *aperture problem* [Horn, 1986].[4] Second, and more importantly, motion estimates based on the data conservation constraint are very sensitive to noise in the images, particularly in regions where there is very little spatial variation, or texture.

To overcome these problems, many approaches have exploited a *spatial coherence* assumption:

> *Neighboring points in the scene typically belong to the same surface and hence have similar velocities. Since neighboring points in the scene project to neighboring points in the image plane, we expect optical flow to vary smoothly.*

This is illustrated in Figure 1.7. This assumption is typically implemented as a *smoothness constraint* which is seen as *regularizing* the ill-posed problem [Poggio *et al.*, 1985; Terzopoulos, 1986].

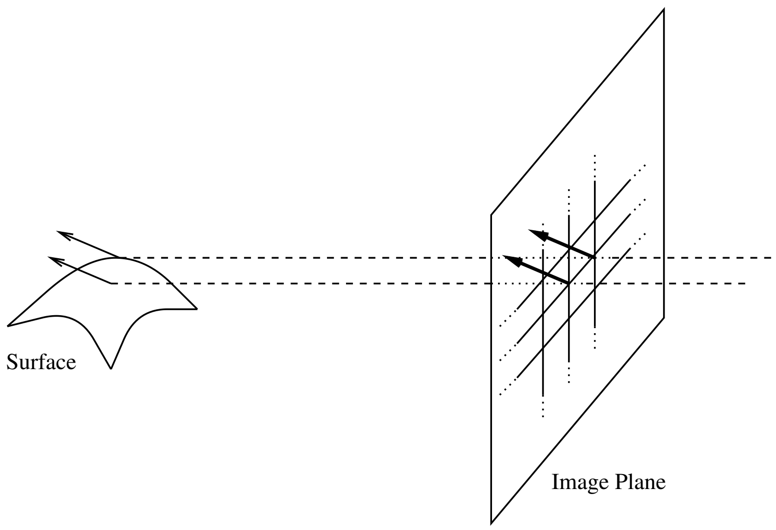


Figure 1.7: Spatial coherence assumption. Neighboring points in the image are assumed to belong to the same surface in the scene.

Temporal Persistence

The previous two constraints are commonly employed in the recovery of optical flow between two frames in an image sequence. A less commonly exploited assumption is that of temporal continuity:

> *The image motion of a surface patch changes gradually over time.*

This constraint, illustrated in Figure 1.8, can be formulated to account for various kinds of image motion; for example constant velocity or constant acceleration in the image plane.

Temporal continuity is a powerful constraint in as much as it reflects the stability and persistence of the scene. Most attempts to exploit the constraint have focused on batch processing of a spatio-temporal block of images; for example, spatio-temporal filtering [Heeger, 1987] and epipolar-plane image analysis [Bolles *et al.*, 1987]. Only recently has the power of the constraint been exploited to integrate motion information over time [Black and Anan-

[4] Because of this ambiguity, the problem of recovering optical flow is referred to as *ill-posed* [Bertero *et al.*, 1988; Marroquin *et al.*, 1987; Poggio *et al.*, 1985].

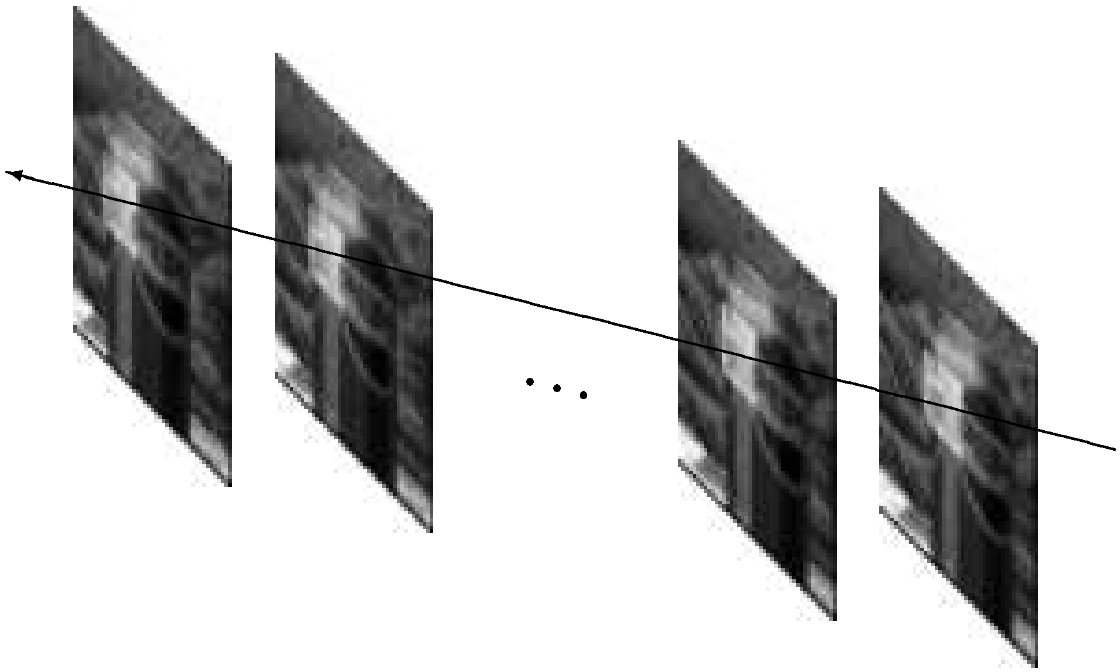


Figure 1.8: Temporal continuity assumption. A patch in the image is assumed to have the same motion (constant velocity, or acceleration) over time.

dan, 1991b; Black and Anandan, 1990b; Singh, 1991; Singh, 1992a]. This temporal integration both improves motion estimates by reducing noise over time and reduces the computation necessary for each new frame.

1.2 Robustness

There are a number of specific problems in optical flow that pertain directly to the robustness of current approaches and must be addressed. The most frequently examined issue is the recovery of piecewise smooth flow fields in the presence of motion boundaries. Related to this issue is the problem of actually localizing the motion discontinuities. The two issues are not the same. There are approaches to recovering piecewise smooth flow which do not explicitly recover motion boundaries [Nagel and Enkelmann, 1986; Singh, 1990] as well as approaches for recovering motion discontinuities which do not compute optical flow [Black and Anandan, 1990a; Mutch and Thompson, 1988; Spoerri and Ullman, 1987]. Since motion boundaries correspond to physically significant structures in the scene, their recovery is of great interest and, for some applications, may be of greater interest than the flow field.

The problems posed by motion discontinuities are one example of a broader issue in computing optical flow: algorithms for recovering optical flow embody a set of assumptions about the world which, by necessity, are simplifications and hence, will be violated. These common assumptions are frequently violated in the real world and this leads to algorithms that are not robust; for example, the simple spatial coherence constraint is violated at motion boundaries and its application results in inaccurate, or "over-smoothed", flow estimates at the boundary. To accurately recovery optical flow, one must either formulate more realistic constraints that model the violations or develop techniques that perform well even when violations are present. In reality, both of these approaches are necessary since any model of the world is an idealization and will be violated in practice. It is not enough however to simply ignore model violations; instead, the goal should be to detect and explain these violations because, as in the case of motion boundaries, they often correspond to interesting properties of the scene.

These issues lead to three goals to which optical flow algorithms should aspire:

1. Recover optical flow without smoothing across motion discontinuities.

2. Locate the motion boundaries so that they are available to other algorithms which require knowledge about the surface boundaries of objects.

3. Detect when the underlying assumptions of the model are violated.

A great deal of progress has been made on the first two goals and, indeed, there are numerous solutions for recovering piecewise smooth flow fields. Instead of focusing on the problem of recovering optical flow with discontinuities, this thesis will treat the the first two goals as special cases of the third, and focus on general issues of robustness in the presense of model violations. Common violations of the data, spatial, and temporal constraints are described below.

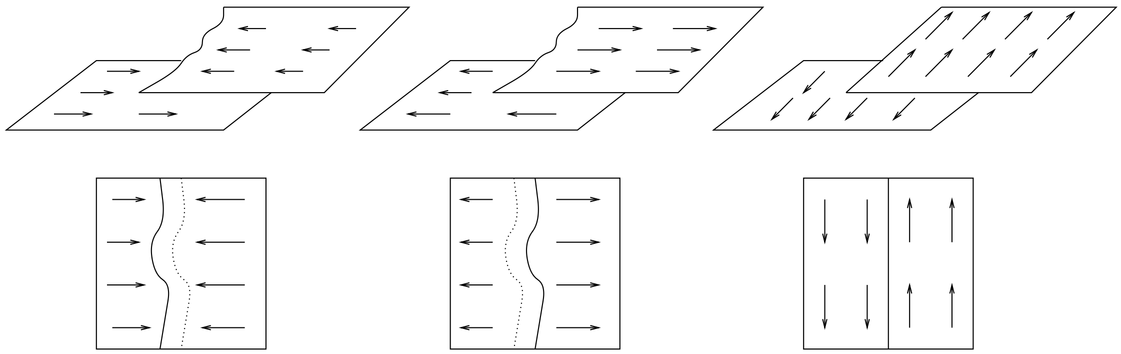
Data Conservation

The data conservation constraint is violated in numerous common lighting situations; for example: specular reflections do not necessarily "move with" the surface patch, shadows can change the appearance of a region, and the illumination may vary (particularly in outdoor scenes). Sensor noise can also be a source of violations, but is more easily modeled than unconstrained illumination changes.

The simple formulation of the data conservation constraint assumes that image regions undergo translation in the image plane and ignores the effect of deformations of local image patches due to the relative motion between the observer and the scene. In even simple scenes, the local image structure can undergo rotation, dilation, contraction, and shear [Koenderink and van Doorn, 1975]. Additionally, if the objects in the scene are not rigid, then the local image structure can change in complex ways.

Finally, when multiple motions exist within a region the data conservation constraint is violated. Recall the simple situation presented in Figure 1.5 where the highlighted region contains both the foreground and background. Since the soda can and the background are moving at different rates, no single motion can account for the intensity changes within the region. This violates the single motion assumption implicit in the data conservation constraint. Moreover, data is not conserved within the region since a portion of the background becomes occluded by the Pepsi can. More generally, in a cluttered scene even very small regions in the image are likely to contain multiple motions. Consider, for example, the case of *fragmented transparency* that arises when viewing a scene through swaying tree branches or while walking past a picket fence.

The problems become more profound still in cases of transparency where the intensity at a point in the image is not determined by a single surface, but may be altered by the interac-



Occlusion Disocclusion Shear

Figure 1.9: Motion Discontinuities

tion of some number of translucent or reflective surfaces. Examples include looking through a dirty window or watching one's reflection in a flowing stream. Humans cope with these situations routinely and can typically attend to one or more of the motions present.

Spatial Coherence

The spatial coherence constraint can take a number of forms, but the most common assumption is that the optical flow within a region is constant. While this assumption is not even valid for planar surfaces and arbitrary translational motion, the most obvious violation of the constraint is at motion discontinuities. Figure 1.9 illustrates some commonly occurring motion discontinuities; including occlusion, disocclusion, and shear.

Applying the spatial coherence constraint at motion discontinuities results in an "over-smoothed" flow field where the motion discontinuities have been obscured; for example, see Figure 1.10. The resulting flow field does not accurately describe the underlying motion. More significantly, the over-smoothing removes important information regarding the location of physically significant properties of the scene.

Figure 1.10: Over-smoothing of flow fields for the Pepsi can image sequence. On the left is a flow field computed with a standard smoothness constraint. The flow field on the right was computed with a modified smoothness constraint that took into account motion discontinuities. Careful inspection of the left image will reveal that the motion of the soda can blends smoothly into the background, while in the right image there is a sharp discontinuity between foreground and background.

Temporal Continuity

While temporal continuity provides a powerful constraint, it is typically only valid over short time intervals. The motion of objects in the world is not nearly so predictable; they stop, change direction, reverse course, etc. Even in a static world, the motion of the camera may not be smooth. If the camera is unstabilized, then vibration, bounce, and sway can all lead to violations of a simple temporal continuity assumption. Strict enforcement of the constraint would lead to mistakes in the predicted location of objects when their motion changes too rapidly.

These problems are related to the rate of image acquisition. Analogous to the Nyquist limit in sampling theory, if the sampling rate is significantly higher than the rate of change in the scene, then the constraint can be applied. This idea has been exploited in work on epipolar-plane image analysis [Bolles *et al.*, 1987] in which images are acquired frequently so that motion between images is kept small. This simplifies the problem of determining correspondence over time. In contrast, this thesis addresses the general problem of optical flow estimation in which dense temporal sampling cannot be guaranteed as is typically the case in applications such as autonomous robotics where images may be acquired at a relatively infrequent 30 frames/second. In practical applications, like mobile robotics, the constraint of temporal continuity is likely to be violated.

The constraint can also be violated at motion boundaries where occlusion and disocclusion each present problems. At occlusion boundaries, a surface which has persisted for some time suddenly disappears. A motion algorithm must be able to deal with this disappearance and either stop tracking the surface or, for some period of time, continue to track the surface despite the lack of visible evidence for its presence. In the latter case, the constraint could be maintained for occlusions of short duration.

Disocclusions present their own problem. All of a sudden a new surface, or portion of a surface, becomes visible and there is no previous information regarding its motion. Any algorithm exploiting temporal continuity must be able to adapt to these kinds of changes.

1.3 Incremental Motion Estimation

One of the goals of computer vision is to embody robots with the ability to understand and act in a dynamic environment. To cope with a continually changing retinal image, a robot must be able to detect and compensate for this image motion. Many approaches for estimating optical flow have focused on the analysis of motion between two frames in an image sequence [Anandan, 1989] while others have attempted to deal with spatiotemporal information by processing long sequences in a batch mode [Baker, 1988; Heeger, 1987]. More recently, there has been an interest in *incremental* approaches which are more suited to the dynamic nature of motion estimation [Black and Anandan, 1991b; Singh, 1991]. The goal of incremental motion estimation can be broadly defined as follows:

> *Incrementally integrate motion information from new images with previous optical flow estimates to obtain more accurate information about the motion in the scene over time.*

While this definition is broad enough to encompass many different techniques, there are some general properties that an incremental algorithm should have:

1. *Anytime Access:* Motion estimates are always available. This is important for robotic applications where motion information is being used for navigation or obstacle avoidance.

2. *Temporal Refinement*[5]: Flow estimates are refined over time as more data is acquired.

3. *Computation Reduction* [Heel, 1991]: By exploiting the information available over time, the amount of computation between any pair of frames is reduced. The goal is eventually to achieve real-time performance.

4. *Adaptation:* The nature of incremental approaches requires them to be adaptive. As the motion of the observer and scene changes over time, an incremental algorithm must adapt to the changes in motion and the changing retinal image.

While the definition rules out purely batch techniques for processing image sequences, it does not rule out the possibility of "local batch processing." Information from new images might be derived by examining some number of previous frames and performing a batch analysis. To be considered dynamic however, an algorithm must use historical information about the motion in the scene and combine it in some way with current information. While the definition also includes incremental feature-based motion algorithms [Faugeras *et al.*, 1987], this thesis will only address the estimation dense optical flow fields.[6]

[5]This idea has also been referred to as "quality improvement" [Heel, 1991].
[6]Where "dense" means that there is an estimate of the optical flow at each pixel in the image.

1.4 The Approach

This thesis addresses the issues of robust and dynamic optical flow estimation. The first half of the thesis exploits techniques from robust statistics [Hampel *et al.*, 1986; Huber, 1981; Rousseeuw and Leroy, 1987] to develop a framework for the robust estimation of optical flow. Such a formulation exacts a computational price and in response, the second half of the thesis develops an incremental minimization framework which is used to recover the optical flow robustly over a sequence of images.

Robust Estimation Framework

There is growing interest in the use of robust statistics and numerous researchers have applied the techniques to the standard problems of computer vision [Meer *et al.*, 1991; Schunck, 1990]. There are robust approaches for performing local image smoothing [Besl *et al.*, 1988; Blauer, 1991], classification [Chen and Schunck, 1990], surface reconstruction [Sinha and Schunck, 1992], segmentation [Meer *et al.*, 1990], pose estimation [Kumar and Hanson, 1990], edge detection [Lui *et al.*, 1990], and structure from motion or stereo [Tirumalai *et al.*, 1990; Weng and Cohen, 1990]. Robust statistical techniques have also been applied to the problem of image velocity estimation [Schunck, 1989a; Schunck, 1989b], but previous formulations lack a coherent, unified, framework for the robust recovery of optical flow fields.

This thesis shows that *robust estimation* [Hampel *et al.*, 1986; Huber, 1981] provides a framework for addressing many of the problems encountered in computing optical flow while generalizing previous approaches. The problems of robustness discussed here are not unique to optical flow, but appear again and again in early vision problems employing regularization techniques. The robust estimation framework that is developed in the context of optical flow can readily be applied to algorithms for problems like stereo, structure from motion, image reconstruction, surface reconstruction, and shape from shading.

The power of the robust estimation framework is illustrated by applying it to a number of common problems in motion estimation including correlation [Anandan, 1989], regression [Lucas and Kanade, 1981], and explicit smoothness (or regularization) [Horn and Schunck, 1981] approaches. In particular, the framework is exploited to develop a *robust gradient* method [Black, 1992b] which meets the three goals outlined above for robust optical flow: 1) it prevents over-smoothing, 2) it allows the recovery of motion discontinuities, and 3) violations of model assumptions are detected and their effects on the solution are reduced.

The robust gradient algorithm also illustrates how the framework provides a uniform way of treating both errors in the motion estimates derived from the input images and motion discontinuities in the flow field. The formulation is straightforward and experiments on real and synthetic image sequences show that it performs well in the presence of non-Gaussian noise and motion discontinuities. The flow field is recovered using a simple, deterministic, relaxation scheme.

The thesis also explores the relationship between the robust estimation approach and other formulations based on line processes [Geman and Geman, 1984] or weak continuity constraints [Blake and Zisserman, 1987]. In certain cases, one can show an equivalence between the robust estimation formulation and the line-process approaches. And, while the robust estimation framework is consistent with previous approaches, it offers some advantages. First, it provides a new statistical interpretation for these other approaches. Second, it provides new tools which can be brought to bear on the problem while not sacrificing the physical appeal of the line-process formulations. And, finally, this new framework shows how the line-process approaches can be generalized and applied to new problems in a way which improves the veracity of the results by reducing the sensitivity of the solution to erroneous measurements.

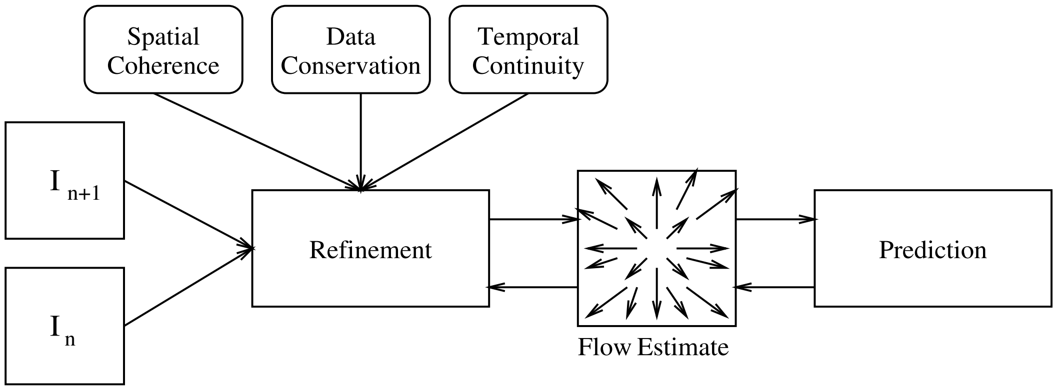


Figure 1.11: Incremental minimization strategy.

Incremental Minimization Framework

In this thesis, the problem of optical flow recovery is formulated as the minimization of an *objective function* composed of the data, spatial, and temporal constraints. The robust formulation of the constraints makes this objective function non-convex, and hence, expensive to minimize. To ameliorate this problem we exploit the temporal continuity constraint described above which, in addition to providing a powerful constraint for the interpretation of visual motion, allows us to predict the optical flow at the next instant in time. This property is used to develop a framework for incrementally minimizing the objective function over the length of an image sequence.

The basic framework is illustrated in Figure 1.11. At any instant in time, the algorithm has a current estimate of the flow field. When a new image is acquired, the constraints are applied and the estimate is refined. The important point is that this refinement stage takes some constant amount of time and only provides a revised estimate of the flow field. At this point, the temporal continuity assumption is exploited to predict what the flow field will be at the next instant in time. A new image is acquired, and the process is repeated. The result of this procedure is that the flow estimate is refined over time, while the goals of anytime access, temporal refinement, and computation reduction are met.

Numerous algorithms can be implemented using this general framework. This thesis describes two such implementations suitable for minimizing non-convex objective functions. The first, called *Incremental Stochastic Minimization (ISM)*, is an incremental version of *simulated annealing* [Kirkpatrick *et al.*, 1983]. The second, is an incremental version of Blake and Zisserman's [1987] Graduated Non-Convexity algorithm. In each of these algorithms, there is a parameter which controls the search for a global minimum. By dynamically controlling this parameter based on the image data, the goal of adaptation is also satisfied.

While the framework was primarily designed for computing optic flow, it has more general applicability. The ability to minimize an objective function over time by compensating for image motion allows other problems to be formulated and solved in this temporal minimization framework. For illustration, Chapter 8 will show how intensity-based feature extraction, which is commonly formulated as an optimization problem [Blake and Zisserman, 1987; Geman *et al.*, 1990], can be performed dynamically using this framework. Such an approach has a number of advantages over static feature extraction. For example, it amortizes the cost of extraction over the image sequence while automatically tracking image features. But, more important, the use of motion information allows image features to be classified as either structural or non-structural properties of the scene.

1.5 Overview of the Thesis

The first portion of the thesis is devoted to issues of robustness while the second half addresses the problem of incremental estimation. Given the diversity of techniques employed, previous work is described, and mathematical tools are introduced, as the need arises.

Chapter 2. The common formulations of the optical flow problem are reviewed. The chapter first introduces the standard formulation of the data conservation constraint and then reviews three common techniques for flow estimation: area regression, correlation, and regularization. In addition to describing the approaches, the chapter explores where they are violated and examines the current approaches for coping with motion discontinuities.

Chapter 3. The chapter begins by reviewing robust statistical techniques and then introduces the robust estimation framework and uses it to reformulate the regression, correlation, and regularization approaches. We then explore the relationships between the robust estimation approach and other current approaches by showing how line-process formulations can be generalized and converted to robust estimation problems. We also show that certain robust estimation problems can be converted to equivalent line-processes formulations.

Chapter 4. The chapter shows how the robust estimation formulations of optical flow compare to the least-squares formulations of regression and correlation. Using the robust estimation framework we develop a robust gradient-based algorithm and show how the framework leads to more accurate flow fields when noise and motion discontinuities are present. Detailed descriptions of the algorithm and experimental results on real and synthetic images are presented.

Chapter 5. The chapter introduces the temporal continuity constraint and reviews previous uses of the constraint. A general framework for incremental minimization is then developed and standard hierarchical approaches for coping with large motions are extended to this incremental framework. Finally, the relationship between the incremental minimization framework and recursive estimation techniques is discussed.

Chapter 6. In this chapter, the incremental minimization framework is applied to the problem of stochastic minimization. We introduce a version of simulated annealing which is suited to solving continuous minimization problems. We then present an incremental stochastic minimization algorithm which extends the annealing process over an image sequence. Experimental results on real and synthetic images are presented.

Chapter 7. A dynamic version of the Graduated Non-Convexity algorithm of Blake and Zisserman [1987] is developed using the incremental minimization framework. We then use this algorithm to demonstrate the psychophysical implications of the temporal continuity constraint.

Chapter 8. This chapter demonstrates how the framework of incremental minimization can be extended to allow other optimization problems to be solved over time. This is illustrated by implementing an incremental feature extraction algorithm which tracks image features over time and classifies them as object boundaries or surface markings.

Chapter 9. We conclude by examining what questions have been answered and what questions remain open. In doing so we point to a number of future directions for work in optical flow.

Chapter 2

Estimating Optical Flow: Approaches and Issues

Most current techniques for recovering optical flow exploit two constraints on image motion: *data conservation* and *spatial coherence*. The data conservation constraint is derived from the observation that surfaces generally persist in time and, hence, the intensity structure of a small region in one image remains constant over time, although its position may change. This assumption is often formulated as a first-order [Horn and Schunck, 1981] or second-order [Nagel, 1983b] constraint on image gradient. Alternatively, the correlation approach [Burt *et al.*, 1982] attempts to find the displacement that minimizes the disparity between an image region in one image and the displaced region in a future image under some match criterion; for example, minimization of the sum of squared differences between pixels in the region [Anandan, 1989]. In many commonly occurring situations, this assumption is violated for some subset of the points within the image region; for example, it is violated at motion boundaries and when specular reflections are present. In these cases, the data conservation constraint may still provide useful information if the violating points can be detected and removed from consideration. When global contrast changes are present, the simple formulation of the constraint may provide no useful information and data conservation measurements should be treated as suspect.

The spatial coherence constraint embodies the assumption that surfaces have spatial extent and hence neighboring pixels in an image are likely to belong to the same surface. Since the motion of neighboring points on a smooth rigid surface changes gradually, we can enforce a *smoothness constraint* [Horn and Schunck, 1981; Snyder, 1991] on the motion of neighboring points in the image plane. It has long been realized that such a constraint is violated at surface boundaries and much of the recent work in motion estimation has focused on where it is violated and how to reformulate it.

This chapter reviews the formulation of the data conservation constraint and the three principal approaches for exploiting the spatial coherence constraint: regression, correlation, and explicit smoothness techniques. We also point out the underlying assumptions of these approaches, indicate when they are violated, describe the problems that result, and review the major approaches for solving the problems.

There is an observation that can be made about many of the approaches described in this chapter. That is, the underlying models used in recovering optical flow assume that the models capture the motion of some finite region. To recover the flow accurately one is driven to applying the model to large regions. However, there is an important tradeoff. As the region size grows, so does the likelihood that the model no longer captures the motion of the region. It will be the focus of the next chapter to develop general tools for coping with this situation.

2.1 Data Conservation Constraint

This section reviews the assumptions underlying most optical flow algorithms. Let $I(x, y, t)$ be the image intensity[1] at a point (x, y) at time t. The data conservation constraint can be expressed in terms of the standard *intensity constancy assumption* as follows:

$$I(x, y, t) \quad = \quad I(x + \delta x, y + \delta y, t + \delta t),$$

[1] In fact, I may be a filtered version of the intensity image at time t.

$$= \quad I(x + u\delta t, y + v\delta t, t + \delta t), \tag{2.1}$$

where $\mathbf{u} = [u, v]^T$ is the horizontal and vertical image velocity at a point and δt is small. This simply states that the image value at time t, at a point (x, y), is the same as the value in a later image at a location offset by the optical flow.

Gradient-based approaches [Horn and Schunck, 1981] proceed by taking the Taylor series expansion of the right hand side of (2.1), yielding:

$$I(x, y, t) = I(x, y, t) + I_x u\delta t + I_y v\delta t + I_t\delta t + \epsilon, \tag{2.2}$$

where I_x, I_y, and I_t are the first partial derivatives of the brightness I with respect to x, y, and t respectively, and where ϵ contains the higher-order terms. Simplifying and dividing through by δt we obtain the standard optical flow constraint equation:

$$I_x u + I_y v + I_t = \nabla I^T \mathbf{u} + I_t = 0. \tag{2.3}$$

To recover an estimate of the optical flow at a point one could simply minimize the data-conservation term:

$$E_D(u, v) = \rho(I_x u + I_y v + I_t). \tag{2.4}$$

When $\rho(x) = x^2$ this corresponds to the standard least-squares estimate described by Horn and Schunck [1981]. As mentioned in Chapter 1, the data-conservation constraint alone is not sufficient for recovering optical flow due to the aperture problem and sensitivity to noise.

Given equation (2.3), we can now see the problem more clearly. Figure 2.1 illustrates that motions satisfying equation (2.3) are only constrained to lie along a line in (u, v) space. The equation only constrains the flow vector to lie in the direction of the image gradient; that is *normal* to the spatial image orientation [Horn, 1986]. Thus, the optical flow problem as stated, is ill-posed [Bertero *et al.*, 1988; Marroquin *et al.*, 1987] and requires additional constraints to recover a unique flow estimate.

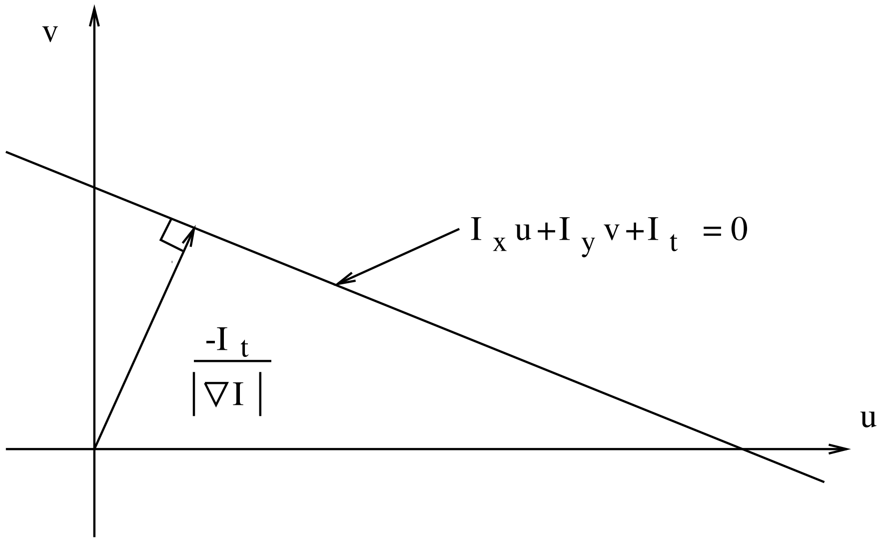


Figure 2.1: Intensity constancy constraint.

Assumptions and Violations

There are a number of implicit assumptions underlying this formulation that are often violated in practice. The simple statement of intensity constancy and the first-order Taylor series approximation assume locally constant translational motion and a planar image intensity function. While these approximations become valid in the limit as the size of the image region shrinks to zero, in practice some finite region size is required, and as it increases the validity of the assumptions are called into question. The intensity constancy assumption also implies that changes in intensity are due solely to motion and, hence, the constraint cannot account for changes in illumination, transparency, or specular reflections.

In practice, the constraint imposes a constant flow assumption over a neighborhood. This results from the fact that to estimate spatial derivatives from discrete images, one must necessarily examine a region of the image. The spatial and temporal derivatives can be estimated from the input using any number of schemes; for example image differences [Horn, 1986] or spatio-temporal filtering [Simoncelli and Adelson, 1991]. Beaudet [1978] defines a catalog of optimal local derivative filters; for example the simple 3×3 filters:

$$D_y^T = D_x = \frac{1}{6}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}.$$

The image derivatives are then computed by convolving the filters with the intensity image: $I_x = D_x * I$, and $I_y = D_y * I$.

Nagel [1983b] has pointed out that by considering second-order spatial derivatives, it is possible to uniquely recover the optical flow at corners in the grey-level image; that is, the aperture problem disappears at corners. The approach, however, still implies locally constant velocity.

Regardless of the approach, the estimates involve pooling information spatially. For small neighborhoods (for example, local differencing), these estimates of image derivatives are highly sensitive to noise, particularly in areas with little texture. A common approach then is to use derivative filters that span larger neighborhoods. The assumption of constant motion, however, is only a good approximation within a small region. For, as the region grows, its motion may be less well approximated by a constant model, and it is more likely to contain multiple motions. The important point to note is that when the neighborhood for estimating image derivatives spans a surface boundary, the resulting measurements may be meaningless. The best flow, **u**, derived by minimizing the intensity constraint equation (2.4) may be incorrect.

2.2 Regression Techniques

Assuming a model of constant flow within a region we can combine information from neighboring gradient constraint equations to determine the best flow $[u, v]$ satisfying all the equations by finding the $[u, v]$ that minimizes the sum of the constraints over the neighborhood:

$$E_D(u, v) = \sum_{(x,y)\in\mathcal{R}} \rho(I_x(x, y)u + I_y(x, y)v + I_t(x, y)), \tag{2.5}$$

where $\rho(x) = x^2$ and $\mathcal{R}$ is some image region.[2] More generally, one can assume a more complex flow model:

$$\mathbf{u}(x, y) = \mathbf{u}(x, y; \mathbf{a})$$

where **a** are the parameters of the model. Example models of image flow in a region $\mathcal{R}$ include constant, affine, and planar [Bergen *et al.*, 1992]. Our goal is to estimate the parameters, **a**, of the model within a region $\mathcal{R}$ by minimizing:

$$E_D(\mathbf{a}) = \sum_{\mathcal{R}} \rho(\nabla I^T \mathbf{u}(\mathbf{a}) + I_t). \tag{2.6}$$

In the constant case the model is simply the same as the equation above:

$$\mathbf{u}(\mathbf{a}) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}. \tag{2.7}$$

For an affine flow model we have:

$$\mathbf{u}(x, y; \mathbf{a}) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix}. \tag{2.8}$$

Notice that when $\rho(x) = x^2$ this is a standard least-squares regression. This regression, or area-based, approach has been applied to stereo matching [Lucas and Kanade, 1981], local motion estimation motion [Simoncelli *et al.*, 1991], and image registration [Bergen *et al.*, 1992].

Assumptions, Violations, and Previous Approaches

The approach assumes that the motion within a region can be described by a single parametric model. When a single surface is present, the affine flow model has been shown to be a reasonable approximation in many cases [Bergen *et al.*, 1992]. But as the complexity of the model increases (that is, more parameters must be estimated), larger image regions are

[2] In the future we will drop the indices (x, y) when it is clear that the equation applies at every point in a region.

required for accurate estimates. The larger the region under consideration, the more likely it is to contain multiple motions and, hence, not to be well approximated by the model.

It is sometimes possible to detect multiple motions by examining the residual of the least-squares solution. This idea has been exploited to produce iterative techniques that cope with and detect multiple motions within a region [Irani *et al.*, 1992]. The approach is to first compute the best least-squares estimate for the region, detect and remove regions that do not correspond to the main motion, and then recompute the quadratic estimate and repeat. Such an approach is a form of robust estimation and has been demonstrated to work well when the distracting motion occupies a small portion of the region [Irani *et al.*, 1992][3].

The case of multiple transparent motions is more complex. One approach [Bergen *et al.*, 1990a] uses an iterative algorithm to estimate one motion, perform a nulling operation to remove the intensity pattern giving rise to the motion, and then solve for the second motion. The process is repeated and the motion estimates are refined.

To use the regression approach for local motion estimation, region sizes must be kept small for efficiency and to reduce the likelihood of multiple motions. Intuitively, this renders the approach more sensitive to noise. One approach to take is to realize this problem and explicitly represent the uncertainty in the flow estimates [Simoncelli *et al.*, 1991]. This uncertainty estimate can be exploited by other algorithms that rely on accurate measurements of image motion.

Unlike the regression techniques above that try to find the best flow given the local intensity constraint equations, Schunck [1989a] proposes a method of *constraint line clustering* for computing flow estimates near motion discontinuities. The approach, which performs a cluster analysis on the intersection of constraint lines within a neighborhood, can be viewed as a robust statistical technique. While the approach does not formulate the optical flow

[3]This approach can work well when there is one dominant motion and the competing motion(s) (outliers) have little effect on the initial least-squares estimate, but the approach can be overwhelmed as will be shown in the following chapter.

problem in terms of robust estimation, Schunck suggests that "further experiments should be conducted to compare robust estimates with constraint line clustering," ([Schunck, 1989a], p. 1018). Schunck [1989b] has also used a least-median of squares regression technique to robustly determine the best constraint line intersection within a neighborhood.

2.3 Correlation Techniques

Correlation approaches are similar to the regression approaches, in that they begin with the intensity constancy assumption, but unlike the gradient formulation, adopt a matching strategy. Given a region in one image the goal is to find the displacement, $[u, v]$, of that region in the next image that minimizes the following error:

$$E_D(u, v) = \sum_{(x,y) \in \mathcal{R}} [I(x, y, t) - I(x + u\delta t, y + v\delta t, t + \delta t)]^2. \tag{2.9}$$

This is the standard Sum-of-Squared-Differences (SSD) measure [Anandan, 1987a]. As noted by Simoncelli and Adelson [1991], taking the Taylor series expansion of the correlation formulation, results in exactly the same formulation as the regression approach. Furthermore, Anandan [1987a; 1987b] has shown that, as δt tends to zero, the results obtained by minimizing his SSD formulation converge to those obtained by Nagel's second-order gradient-based approach [Nagel, 1983b]. Additionally, Anandan shows that as the size of $\mathcal{R}$ tends to zero, his SSD formulation converges to the first-order gradient-based formulation.

The approach is illustrated in Figure 2.2. Notice that, over a range of displacements, (u, v), the equation gives rise to a correlation surface in which the minimum corresponds to the best displacement given the match criterion. This correlation surface, E_D, is usually computed over discrete displacements. Determining the best match in this case is straightforward, but does not provide sub-pixel accuracy. One can interpolate the image and compute E_D at sub-pixel displacements, but this involves a more expensive search problem and

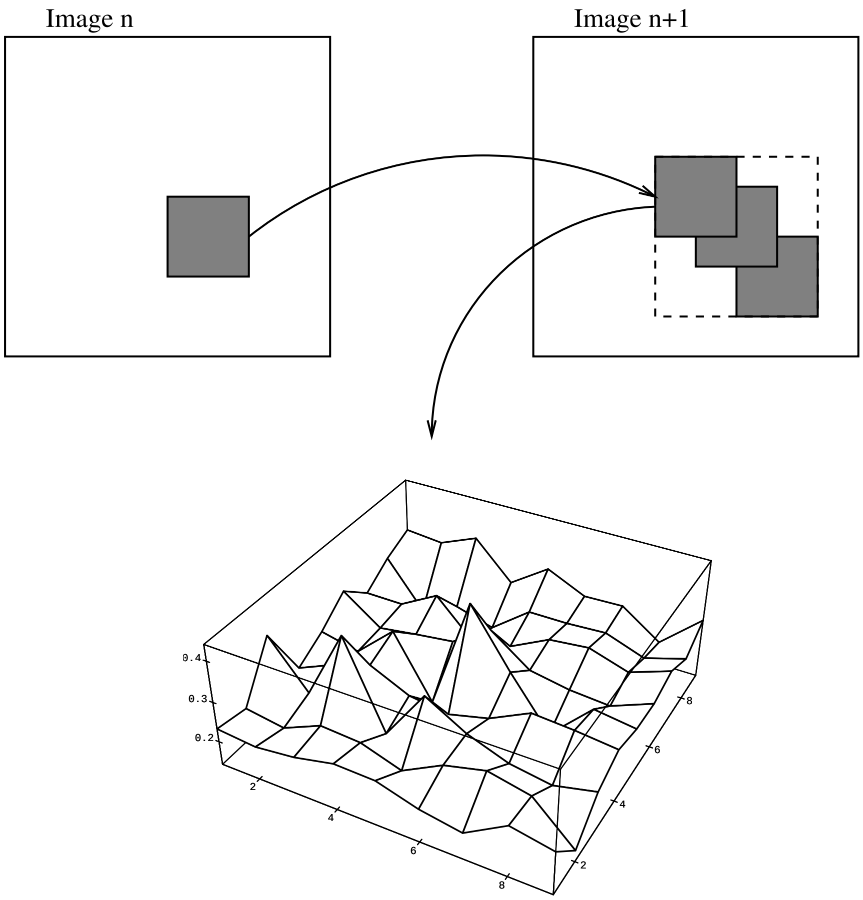


Figure 2.2: Correlation of a patch from one image into the next gives rise to a correlation surface (inverted here for display so that the minima appear as peaks).

the interpolation introduces assumptions about the underlying image structure. More commonly, one computes the discrete SSD surface, finds the best displacement, and computes sub-pixel estimates by fitting a quadratic to the minimum [Anandan, 1987a; Matthies *et al.*, 1989].

Correlation is a popular tool in computer vision and has formed the basic matching strategy in many motion and stereo algorithms. It has also been use for tracking [Papanikolopoulos and Khosla, 1991], and real-time correlation hardware [Burt *et al.*, 1989; Inoue *et al.*, 1992; Nishihara, 1984] makes it attractive for robotic applications.

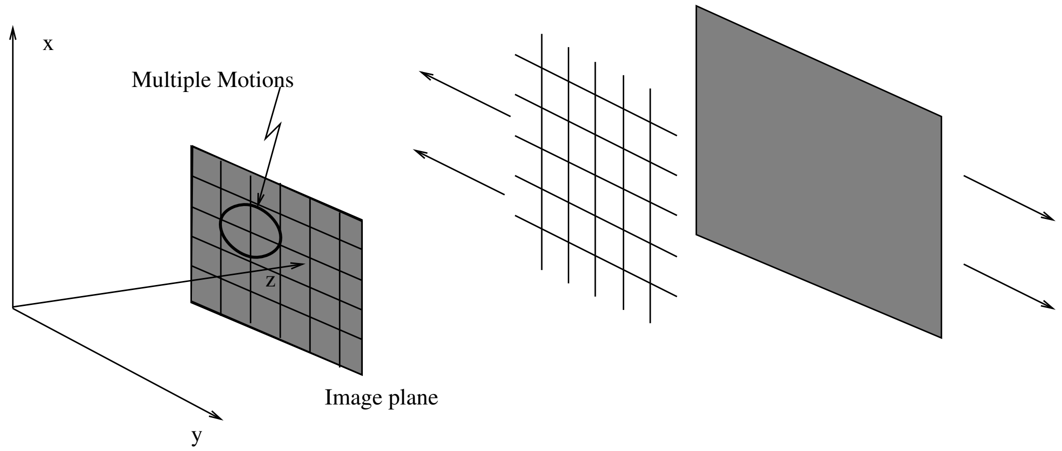


Figure 2.3: Multiple motions are particularly difficult to deal with in situations of *fragmented transparency.* Regardless of the window size chosen, multiple motions will be present.

Assumptions, Violations, and Previous Approaches

The correlation approach suffers from the same problems as the regression approach. In particular it assumes the flow field can be approximated as uniform translational motion within the region of interest. In practice, small amounts of rotation, divergence, and shear can be tolerated. While the approach also assumes constant illumination, some illumination changes can be accommodated by using a normalized form of correlation or by prefiltering the images with a Laplacian filter [Burt *et al.*, 1982].

There is a tradeoff to be addressed with correlation-based approaches; as correlation window size is increased to improve the reliability of motion estimates, the likelihood that multiple motions will corrupt the solution also increases. To cope with multiple motions within a window, Okutomi and Kanade [1992] develop an “adaptive window” technique that adjusts the size of the correlation window to minimize the uncertainty in the estimate. Their implementation of the approach is limited by the use of a fixed shape (rectangular) window that cannot adapt to irregular surface boundaries. The approach also cannot cope with fragmented occlusion [Shizawa and Mase, 1991] (for example, trees or fences) where, regardless of window size or shape, multiple motions are present (see Figure 2.3). It would be interesting to extend the adaptive window approach to allow an arbitrary subset of the pixels within a region to be removed from consideration. Such an approach would have the same flavor as the approach described later in this thesis.

When multiple motions are present within a correlation region, the correlation surface may contain multiple minima corresponding to the different motions. Additionally, the motions will interfere with each other, making the minima less clearly defined, and hence, more sensitive to noise and less reliably detectable. In areas of low texture, noise can also produce multiple minima in the correlation surface. Confidence-based approaches attempt to deal with violations of the data term by assigning low confidence to these measurements. For example, Anandan [1989] computes a directionally selective confidence measure based on the curvature of the sum-of-squared-difference surface. Areas of low texture do not give rise to sharp peaks in the SSD surface and hence are assigned low confidence. Thus areas most sensitive to noise receive low confidence.

Singh [1992a] takes an estimation-theoretic approach to the problem. He first computes a *response distribution*, $\mathcal{D}$, given the SSD surface, E_D, defined over some range of discrete displacements, $-N \leq u, v \leq N$:

$$\mathcal{D}(u,v) = e^{-kE_D(u,v)}, \;\; -N \leq u, v \leq N.$$

He then computes a least-squares motion estimate:

$$\begin{aligned} u_c &= \frac{\sum_u \sum_v \mathcal{D}(u,v)u}{\sum_u \sum_v \mathcal{D}(u,v)}, \\ v_c &= \frac{\sum_u \sum_v \mathcal{D}(u,v)v}{\sum_u \sum_v \mathcal{D}(u,v)}, \end{aligned}$$

with the following covariance matrix:

$$S = \begin{bmatrix} \frac{\sum_u \sum_v \mathcal{D}(u,v)(u-u_c)^2}{\sum_u \sum_v \mathcal{D}(u,v)} & \frac{\sum_u \sum_v \mathcal{D}(u,v)(u-u_c)(v-v_c)}{\sum_u \sum_v \mathcal{D}(u,v)} \\ \frac{\sum_u \sum_v \mathcal{D}(u,v)(u-u_c)(v-v_c)}{\sum_u \sum_v \mathcal{D}(u,v)} & \frac{\sum_u \sum_v \mathcal{D}(u,v)(v-v_c)^2}{\sum_u \sum_v \mathcal{D}(u,v)} \end{bmatrix}.$$

Confidence measures can be defined as reciprocals of the eigenvalues of S. When multiple motions are present the motion estimate may be incorrect but the confidence in the motion estimate will be low.

While these multiple minima are a problem for optical flow algorithms, a number of authors [Anandan, 1987a; Black and Anandan, 1990a; Fennema and Thompson, 1979] point out they can be exploited for other purposes. The presence of multiple minima in the correlation surface indicates the possible presence of a motion discontinuity. With some additional constraints, it is possible to detect the presence of motion discontinuities before the computation of optical flow [Black and Anandan, 1990a].

2.4 Explicit Smoothness Techniques

The area-based techniques above employ an implicit spatial coherence constraint in that the flow within a region is assumed to conform to a single motion model. This section addresses regularization schemes which explicitly implement the spatial coherence constraint. The notion of a smoothness constraint is motivated by the fact that the local gradient information may only partially constrain the solution. Additionally, local gradient measurements are sensitive to image noise, particularly in areas containing little variation in contrast. The introduction of a spatial coherence constraint restricts the class of admissible solutions, making the problem well-posed. Such regularization techniques have received a great deal of attention (see [Poggio *et al.*, 1985] for a review).

The data conservation term, E_D, is now combined with an explicit smoothness term, E_S, to form an objective function, E, which is to be minimized:

$$E(\mathbf{u}) = \lambda E_D(\mathbf{u}) + E_S(\mathbf{u}), \tag{2.10}$$

where λ controls the relative importance of the two terms. The smoothness term is defined as a local constraint over a small spatial neighborhood. It is convenient to adopt a Markov

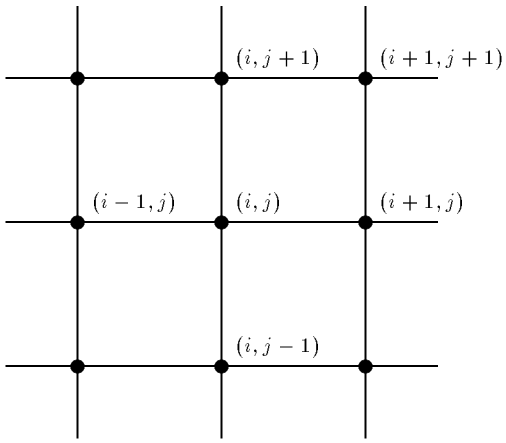


Figure 2.4: An image is treated as a grid of *sites*.

random field (MRF) formulation of the problem. In this chapter, this is formulation is purely for notational convenience and Markov random fields will not be formally introduced until Chapter 6.

For an image of size $n \times n$ pixels we define a grid of *sites* (Figure 2.4):

$$S = \{s_1, s_2, \ldots, s_{n^2} \mid \forall w \; 0 \leq i(s_w), j(s_w) \leq n-1\},$$

where $(i(s), j(s))$ denotes the pixel coordinates of site s. For different formulations of the constraint, we will define different *neighborhood systems*, $\mathcal{G}$, that determine the local interaction of sites. A *neighborhood system*, $\mathcal{G} = \{\mathcal{G}_s, s \in S\}$, satisfies the following conditions [Geman and Geman, 1984]:

1. $\mathcal{G}_s \subseteq S$,

2. $s \notin \mathcal{G}_s$, and

3. $s \in \mathcal{G}_t \Leftrightarrow t \in \mathcal{G}_s$.

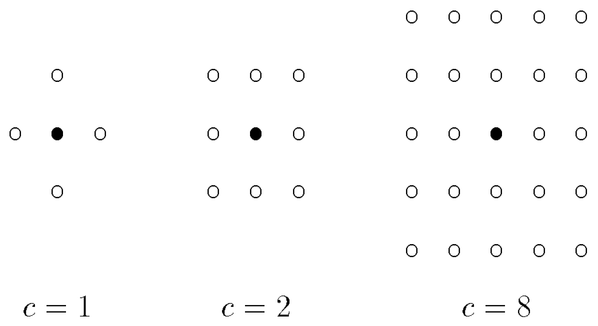


Figure 2.5: Various sized neighborhoods in a grid.

The pair $\{S, \mathcal{G}\}$ then defines a graph with $s \in S$ representing the vertices and pairs, $\{(s,t) \mid s \in \mathcal{G}_t\}$, being the edges. We define a *clique* to be a set of sites, $C \subseteq S$, such that if $s, t \in C$ and $s \neq t$, then $t \in \mathcal{G}_s$. Let $\mathcal{C}$ be a set of cliques.

In the case of images, we are interested in particular types of graphs, namely *grids*, hence we will consider local neighborhood systems of the form:

$$\mathcal{G}_s = \{t \mid 0 < (i(s) - i(t))^2 + (j(s) - j(t))^2 \leq c\}.$$

Figure 2.5 shows the neighborhood systems for various values of c. For first-order constraints $(c = 1)$[4], which are used extensively in the literature, we look only at the nearest neighbor relations (North, South, East, West) in the grid:

$$\mathcal{G}_s = \{t \mid (i,j) = (i(s), j(s)), \ \ (i(t), j(t)) \in \{(i+1, j), (i, j+1), (i-1, j), (i, j-1)\}\}.$$

Even this simple neighborhood system proves very useful and it has an added benefit of being easily realized on many parallel architectures.

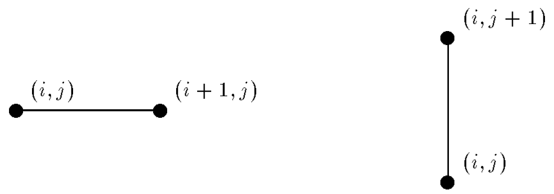
[4] We call these *first-order* because the neighborhoods allow us to formalize constraints based on the first-differences between a site and its neighbors.

2.4.1 Constant-Flow Model

The most common formulation of E_S is the *first-order*, or *membrane*, model. We take as a measure of smoothness the square of the velocity field gradient:

$$E_S(u, v) = u_x^2 + u_y^2 + v_x^2 + v_y^2, \tag{2.11}$$

where the subscripts indicate partial derivatives in the x or y direction. We can approximate this equation using a discrete first-order neighborhood system where we consider the following neighbors:



This leads to the approximation:

$$\begin{aligned} E_S(\mathbf{u}_{i,j}) &= (u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i,j+1})^2 + \\ &\quad (v_{i,j} - v_{i+1,j})^2 + (v_{i,j} - v_{i,j+1})^2. \end{aligned} \tag{2.12}$$

The minimum of this is simply the mean flow $\bar{\mathbf{u}}$ for the neighboring points to the North and East:

$$\frac{\partial E_S}{\partial u} = u - \frac{1}{2}(u_{i+1,j} + u_{i,j+1}) = u - \bar{u}, \tag{2.13}$$

$$\frac{\partial E_S}{\partial v} = v - \frac{1}{2}(v_{i+1,j} + v_{i,j+1}) = v - \bar{v}. \tag{2.14}$$

Notice that the mean flow is the best least-squares estimate of the flow for a constant-flow model. Thus, the simple first-order model implies a locally constant optical flow field. A more reliable estimate of the mean flow can be achieved by considering a larger region. For example, Horn [1986] suggests computing the average by convolving the components of the flow with the mask:

$$\frac{1}{20}\begin{bmatrix} 1 & 4 & 1 \\ 4 & 0 & 4 \\ 1 & 4 & 1 \end{bmatrix}.$$

Following the approach of Horn and Schunck [1981], we can take E_D to be the gradient constraint equation and E_S to be this simple constant-flow model. This gives the following least-squares formulation of optical flow:

$$E(\mathbf{u}) = \lambda(I_x u_{i,j} + I_y v_{i,j} + I_t)^2 + \frac{1}{2}[(u_{i,j} - \bar{u}_{i,j})^2 + (v_{i,j} - \bar{v}_{i,j})^2]. \tag{2.15}$$

This formulation admits a simple iterative relaxation scheme for determining the optical flow:

$$u_{i,j}^{(n+1)} = \bar{u}_{i,j}^n - \frac{I_x(I_x\bar{u}_{i,j}^n + I_y\bar{v}_{i,j}^n + I_t)}{1 + \lambda(I_x^2 + I_y^2)}, \tag{2.16}$$

$$v_{i,j}^{(n+1)} = \bar{v}_{i,j}^n - \frac{I_y(I_x\bar{u}_{i,j}^n + I_y\bar{v}_{i,j}^n + I_t)}{1 + \lambda(I_x^2 + I_y^2)}. \tag{2.17}$$

Intuitively, the problem with simple relaxation schemes like this is that to reduce the effects of noise one must oversmooth the flow field. Remaining faithful to the image measurements on the other hand results in a noisy flow field. What is needed is a way to ignore noisy measurements and at the same time prevent smoothing across discontinuities.

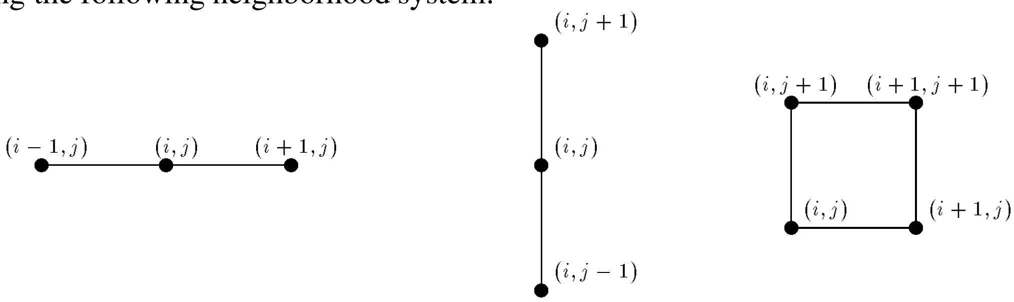
2.4.2 Affine-Flow Model

Now consider a second-order smoothness constraint which is referred to by Geman and Reynolds [1992] as the *planar* case and by Blake and Zisserman [1987] as the *plate* model. For notational simplicity we will consider just the horizontal component of the flow, u; the treatment is identical for the vertical component. The second-order constraint is:

$$E_S(u) = u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2, \tag{2.18}$$

where the subscripts indicate second partial derivatives of the flow.

Using the following neighborhood system:



a *b* *c*

the problem is discretized as follows:

$$u_{xx} = u_{i,j-1} + u_{i,j+1} - 2u_{i,j}, \tag{2.19}$$

$$u_{yy} = u_{i-1,j} + u_{i+1,j} - 2u_{i,j}, \tag{2.20}$$

$$u_{xy} = -u_{i,j} - u_{i+1,j+1} + u_{i,j+1} + u_{i+1,j}. \tag{2.21}$$

The smoothness constraint is minimized when the second partial derivatives u_{xx}, u_{yy}, and u_{xy} are zero. This is the the case when the flow field is locally *affine*; that is, linear in x and y. As mentioned earlier, these affine models of optical flow have become popular in regression-based formulations as an alternative to the constant-flow model.

Assumptions, Violations, and Previous Approaches

The two smoothness models described above both assume that a single model can describe the optical flow locally. Consider what happens if the flow field is discontinuous; that is, there are multiple motions present in the neighborhood. Figure 2.6 illustrates the situation. The constant-flow approximation forces $u_{i,j}$ to the average of its neighbors $u_{i+1,j}$, $u_{i,j+1}$, $u_{i-1,j}$, and $u_{i,j-1}$. The averaging of the smoothness constraint will result in a *blurring* across the motion boundary. Not only does this reduce the accuracy of the flow field, but it obscures important structural information about the presence of an object boundary. Instead of over-smoothing, what one would like to do is realize that the flow at $u_{i+1,j}$ is different from the rest and ignore it.

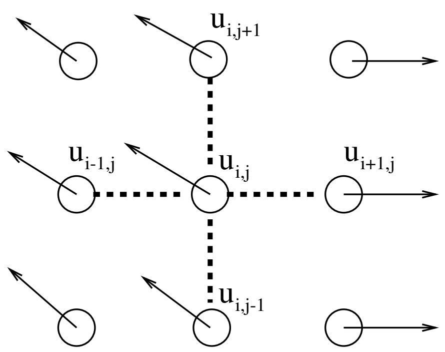
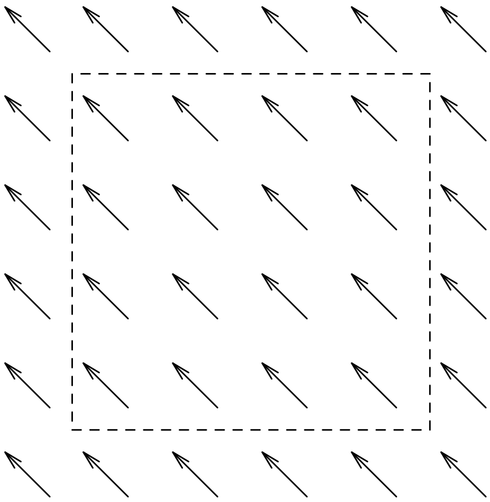


Figure 2.6: Smoothing across a flow discontinuity.

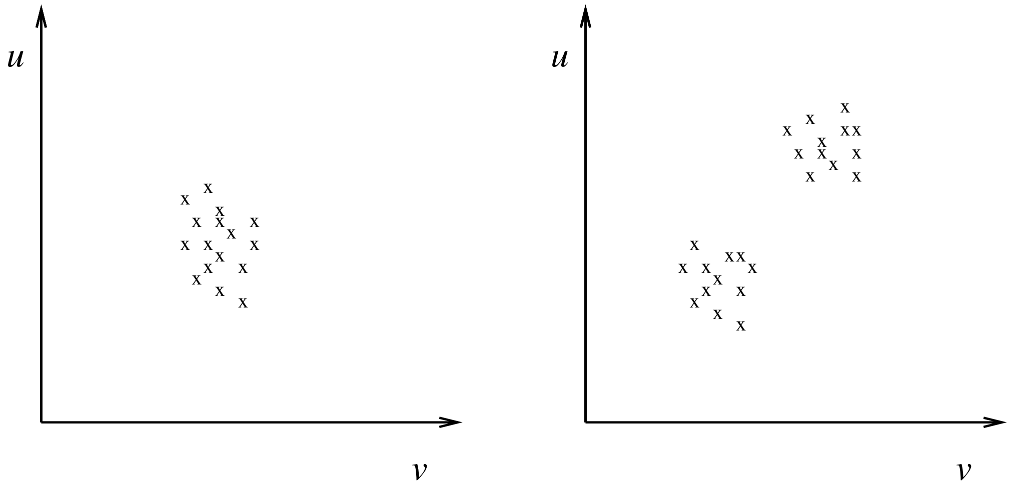
Another way to view this problem is by considering the distribution of flow vectors over a larger neighborhood. For example, the neighborhood in Figure 2.7*a* contains a set of flow vectors that are consistent with a constant flow assumption. In contrast, the neighborhood in Figure 2.7*b* spans a motion boundary; the flow vectors in the region fall into two distinct groups. This can be seen in Figure 2.8 where the flow vectors within a neighborhood are plotted in a u–v coordinate system. Figure 2.8*a* corresponds to constant flow within a region; in this situation, the vectors are clustered in u–v-space, and the mean flow provides a reasonable estimate of the motion. Figure 2.8*b* corresponds to the multiple-motion case where the flow vectors form multiple distinct clusters in u–v-space.

In the case of multiple clusters, the mean flow does not do a good job of characterizing the flow of either cluster. Instead, in cases like this, the goal should be to find the flow that best describes the majority of the data. There are numerous techniques that attempt this; the most important being the line-process approaches which provide general techniques for regularization with discontinuities. There are other techniques as well; including heuristic techniques that exploit information about the intensity image as well as algorithmic techniques that try to detect discontinuities in the flow field.



a *b*

Figure 2.7: Local neighborhoods of flow vectors; *a)* single motion within a neighborhood, *b)* multiple motions within a neighborhood.



a *b*

Figure 2.8: Local distributions of flow vectors; *a)* single motion, *b)* multiple motions.

Line-Process Approaches

A large number of researchers have focused on regularization problems involving discontinuities. For example, Terzopoulos [1986] proposes *controlled-continuity stabilizers* which can account for both first-order (step) and second-order (crease) discontinuities and can be extended to recover higher-order discontinuities. In related work, Szeliski [1988] describes a Bayesian framework for representing reconstruction problems involving discontinuities.

An important class of techniques for coping with spatial discontinuities are the Markov random field (MRF) formulations [Geman and Geman, 1984; Marroquin *et al.*, 1987] which a number of authors have applied to the optical flow problem [Black and Anandan, 1990b; Black and Anandan, 1991b; Konrad, 1989; Konrad and Dubois, 1988; Murray and Buxton, 1987; Tian and Shah, 1992]. These approaches represent discontinuities either explicitly with the use of a "line process" [Geman and Geman, 1984] or by using *weak continuity constraints* [Blake and Zisserman, 1987; Harris *et al.*, 1990; Koch *et al.*, 1988].

Consider the one dimensional example of Blake and Zisserman [1987]. Given some noisy discontinuous data d_i, $0 \leq i \leq n$, (Figure 2.9*a*) find a piecewise-smooth approximation u_i of the true function. With a standard least-squares approach the approximation will smooth the data too much and the discontinuity will not be recovered (Figure 2.9*b*). We desire a more robust fit to the data that provides a better piecewise-smooth interpretation and indicates the location of discontinuities. This can be expressed as the following minimization problem:

$$\min_{u,l} E(u,l) \text{ where } E(u,l) = E_D(u) + E_S(u,l) + E_P(l), \tag{2.22}$$

and where:

$$E_D(u) = \sum_{i=0}^{n} (u_i - d_i)^2 \tag{2.23}$$

$$E_S(u,l) = \beta \sum_{i=0}^{n} (u_i - u_{i-1})^2 (1 - l_i) \tag{2.24}$$

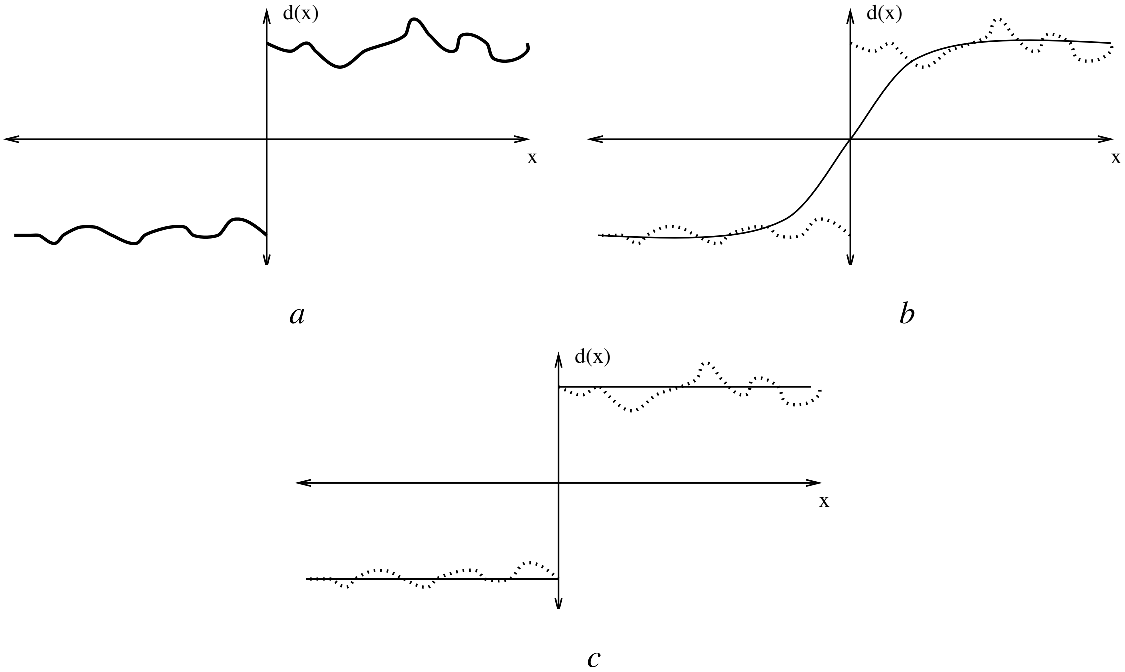


Figure 2.9: A 1D example of piecewise smoothness (redrawn from Blake and Zisserman [1987]). Figure *a* illustrates noisy data to which we would like to fit a 1D function. Figure *b* illustrates what the least-squares estimate found by minimizing $E(u) = \sum_i [(u_i - d_i)^2 + (u_i - u_{i-1})^2]$ might look like. Figure *c* shows the recovery of a discontinuous function.

$$E_P(l) \quad = \quad \alpha \sum_{i=0}^{n} l_i. \tag{2.25}$$

The first term, $E_D(u)$, in the objective function enforces fidelity to the data. The second term, $E_S(u, l)$, encodes a prior first-order smoothness assumption. The l_i are boolean valued *line variables* which indicate the presence $(l_i = 1)$ or absence $(l_i = 0)$ of a discontinuity between neighboring values. The final term, $E_P(l)$, is a *penalty term* which penalizes the introduction of a discontinuity. The idea here is that discontinuities are rare and should only be introduced when they contribute significantly to a better piecewise-smooth solution.

This notion of a line variable can be extended to two dimensions for recovering discontinuities in optical flow. We define a dual $n \times n$ lattice, $S^L = (s, t)$, of all nearest neighbor pairs (s, t) in S. Figure 2.10 shows the pixel sites $(\circ)$ in the original graph and the disconti-

$$\begin{array}{ccccc}
\circ & | & \circ & | & \circ \\
- & & - & & - \\
\circ & | & \circ & | & \circ \\
- & & - & & - \\
\circ & | & \circ & | & \circ
\end{array}$$

Figure 2.10: Arrangement of pixel sites $(\circ)$ and discontinuities $(|, -)$.

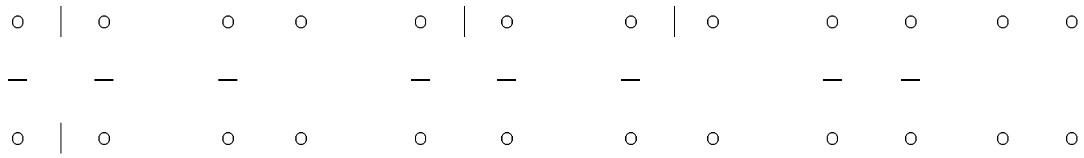
nuity processes $(|, -)$ between the sites. This lattice is coupled to the original in such a way that the best interpretation of the data will be one in which the data is piecewise smooth.

We define a line process $l = \{l_{s,t} \mid s, t \in S,\ t \in \mathcal{G}_s\}$, where $l_{s,t} \in \{0, 1\}$. If $l_{s,t} = 0$ then there is no discontinuity between the sites s and t. In the case where $l_{s,t} = 1$, the neighboring sites are disconnected and hence a discontinuity exists. To recover piecewise-constant optical flow the problem is reformulated to introduce line processes in the spatial term:

$$\begin{aligned}
E(\mathbf{u}, l) &= \sum_{s\in S}[E_D(\mathbf{u}_s) + \sum_{t\in\mathcal{G}_s}[(1 - l_{s,t})(u_s - u_t)^2 + (1 - l_{s,t})(v_s - v_t)^2 + \alpha l_{s,t}]] \\
&= \sum_{s\in S}[E_D(\mathbf{u}_s) + \sum_{t\in\mathcal{G}_s}[(1 - l_{s,t})((u_s - u_t)^2 + (v_s - v_t)^2) + \alpha l_{s,t}]] \\
&= \sum_{s\in S}[E_D(\mathbf{u}_s) + \sum_{t\in\mathcal{G}_s}[(1 - l_{s,t})\|\mathbf{u}_s - \mathbf{u}_t\|^2 + \alpha l_{s,t}]]. \qquad (2.26)
\end{aligned}$$

This first-order formulation with discontinuities corresponds to a *weak membrane* model, while the second-order formulation corresponds to a *thin plate*.

The computational task is now much greater as we must jointly estimate **u** and the discontinuities l. The original least-squares formulation of E was convex and hence easy to minimize. As we will see later, the introduction of the line processes results in a non-convex objective function that is more expensive to minimize. There are minimization procedures for non-convex problems that achieve good results but typically rely on expensive stochastic



ξ_1 ξ_2 ξ_3 ξ_4 ξ_5 ξ_6

Figure 2.11: Possible configurations of discontinuities at four neighboring edge sites (up to rotations of $\pi/2$).

minimization procedures [Geman and Geman, 1984; Kirkpatrick *et al.*, 1983] or continuation methods [Blake and Zisserman, 1987; Rangarajan and Chellapa]. A related approach implements analog, or binary, line processes in hardware using nonlinear resistive networks [Harris *et al.*, 1990; Koch *et al.*, 1988].

A common use of line process is to model the expected spatial properties of discontinuities in the image; for example the Gestalt notions of continuity or simplicity of form [Wertheimer, 1912; Koffka, 1935; Lowe, 1985]. This can be achieved by modifying the penalty term to take into account the local configurations of discontinuities:

$$E_P(l) = \alpha \sum_{c \in \mathcal{C}} W_c(l), \quad \alpha > 0, \tag{2.27}$$

where $c \in \mathcal{C}$ are cliques containing four neighboring edge sites, and W_c assigns a "weight" to the configuration of edges in the neighborhood. The possible configurations are shown in Figure 2.11. The penalties associated with each configuration can be assigned weights, $W_c = \xi_i$, to reflect a preference for certain configurations; for example $\xi_4 = \xi_5 = \xi_6 > \xi_3 > \xi_1 = \xi_2$.

Other Approaches

A number of approaches have explored the use of the grey-level intensity image to control the behavior of the smoothness constraint in optical flow [Cornelius and Kanade, 1983; Hildreth, 1983; Nagel and Enkelmann, 1986; Wu *et al.*, 1982]. These approaches exploit the heuristic that surface boundaries often appear as intensity discontinuities in images. One approach disables the smoothness term at intensity edges thus preventing smoothing across the boundary [Cornelius and Kanade, 1983; Gamble and Poggio, 1987; Hutchinson *et al.*, 1988]. Alternatively, the "oriented smoothness" constraint of Nagel and Enkelmann [1986] ties the effect of the smoothness constraint directly to the grey-level variation in the image, enforcing smoothness only along the directions for which the grey-level variation is not sufficient to determine the flow vector.

A related class of approaches use confidence measures computed from the data to propagate flow measurements from areas of high confidence to areas of low confidence [Anandan, 1989]. Singh, for example, [1990] uses covariance matrices computed from the SSD surfaces and the distribution of flow vectors in small neighborhoods to determine an optimal flow estimate.

Schunck [1989a] interleaves discontinuity detection and regularization. Given an estimate of the optical flow, motion discontinuities are detected in the flow field [Thompson *et al.*, 1985] and then a smoothness operator is applied that prevents smoothing across the boundary. This gives a new flow estimate and the process is repeated.

Darrell and Pentland [1991] have noted the limitations of edge-based schemes when recovering multiple motions in cases of fragmented occlusion. Instead, they propose a scheme in which multiple layers are used to represent the various motions. They use an Markov random field approach to assign pixels to the appropriate layer.

2.5 Large Motions

An implicit assumption of the gradient-based approaches is that the image motion is small; that is, less than a pixel. While the correlation approach can cope with larger motions, the computational burden, and possibility for false matches, rises rapidly as the search distance increases. To ameliorate these difficulties, multi-resolution schemes are often used.

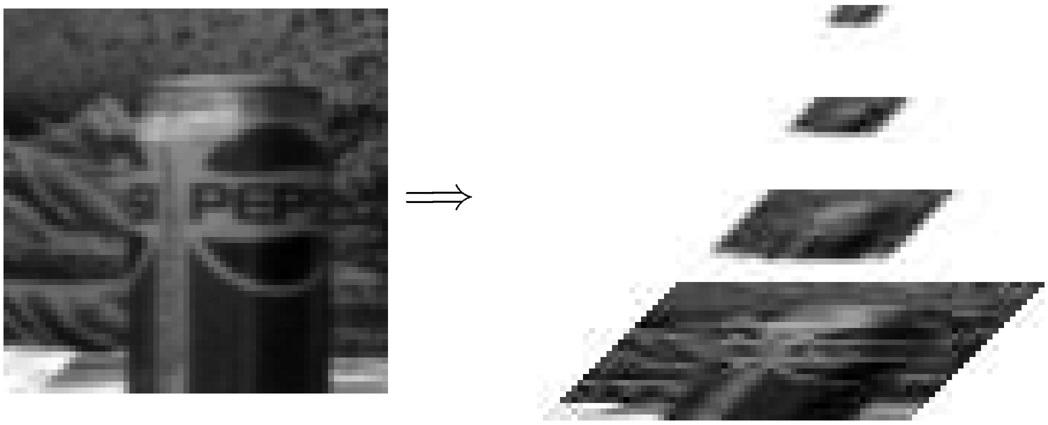


Figure 2.12: Gaussian pyramid. Each level in the pyramid is a subsampled version of the level below convolved with a Gaussian filter.

The basic idea is to construct a pyramidal representation of an image [Burt and Adelson, 1983] in which higher levels of the pyramid contain filtered and sub-sampled versions of the original image (Figure 2.12). This reduction operation can be implemented as convolution and sub-sampling:

```
if x ≡ 0(mod 2) and y ≡ 0(mod 2) then
    I^{p-1}(x/2, y/2) ← f * I^p(x, y)  ∀x, y at level p
end
```

$$\texttt{if}\ x \equiv 0 (\mathrm{mod}\ 2)\ \texttt{and}\ y \equiv 0 (\mathrm{mod}\ 2)\ \texttt{then}\quad I^{p-1}(\tfrac{x}{2}, \tfrac{y}{2}) \leftarrow f * I^p(x,y)\ \ \forall x, y \text{ at level } p\quad \texttt{end}$$

where f is some filter (for example a Gaussian), "$*$" represents convolution, and I^p is the image at level p in the pyramid.[5] While various other filters can be used, the general effect is to reduce the high frequency components of the image at the coarser scales. Multigrid relaxation algorithms [Terzopoulos, 1983] exploit this property to converge on a coarse solution at the high levels and successively refine the solution at finer levels of the pyramid.

[5] The image size at level p is $2^p \times 2^p$; so, for example, at the coarse level, $p = 0$.

Multi-resolution gradient-based schemes have been developed by Enkelmann [1986] and Glazer [1987]. Anandan [1989; 1987a] developed a coarse-to-fine correlation-based approach. Coarse-to-fine refinement strategies have also been applied to regression schemes [Bergen *et al.*, 1992]. Below, a simple coarse-to-fine scheme, that is consistent with these approaches, is described. Later in the thesis, we will explore a number of novel strategies.

The following is a sketch of a simple coarse-to-fine gradient-based approach:

```
for p from coarse-level to fine-level do
    u^p ← project(u^{p-1}, p)                              ; project with interpolation
    ΔI^p ← I^p(x, t) − I^p(x − u^p, t − 1)                 ; warp image by flow field
    min_δu(I^p_x δu + I^p_y δy + ΔI^p) + E_S(u^p + δu)     ; compute new δu
    u^p ← u^p + δu                                         ; update flow
end
```

The algorithm uses pyramids of images and flow fields. At a given level in the pyramid, the algorithm takes as an initial estimate, the *projection* of the optical flow computed at the next coarser level.[6] The flow estimate is used to *warp* the image at time $t-1$ "towards" the image at time t. This allows us to compute an estimate of the temporal derivative ΔI.[7] We then obtain a new estimate of the motion, $\delta\mathbf{u}$, between the partially registered images and update the motion estimate $\mathbf{u}$. This kind of hierarchical processing is illustrated in Figure 2.13.

The projection operation can take a number of forms, the simplest being "projection with duplication":

$$\mathbf{u}^p(x, y) \leftarrow 2\mathbf{u}^{p-1}(\lfloor \tfrac{x}{2} \rfloor, \lfloor \tfrac{y}{2} \rfloor), \quad \forall x, y \text{ at level } p.$$

Here grid points are projected from the level above and duplicated to "fill in" the missing grid points. A better scheme is "projection with interpolation":

[6] The initial flow at the coarse level is taken to be zero.

[7] Here we follow the notation of [Bergen *et al.*, 1992] in using ΔI instead of I_t to indicate that the temporal derivative is being estimated between a partially registered pair of images.

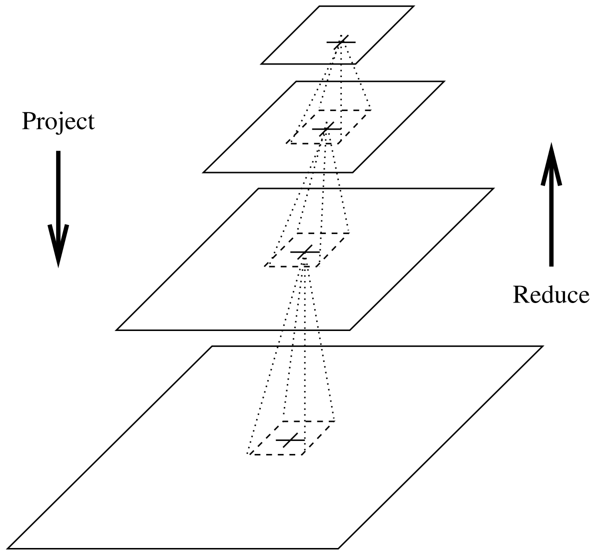


Figure 2.13: Hierarchical Processing.

$\mathbf{u}^p(x,y) \leftarrow 2\mathbf{u}^{p-1}(\text{round}(\frac{x}{2}), \text{round}(\frac{y}{2})), \quad \forall x, y$ at level p
`if` $x \equiv 1(\text{mod } 2)$ `and` $y \equiv 1(\text{mod } 2)$ `then`
$\quad \mathbf{u}^p(x,y) \leftarrow \frac{1}{4}[u^p(x-1,y-1) + u^p(x+1,y-1) +$
$\quad\quad u^p(x-1,y+1) + u^p(x+1,y+1)], \quad \forall x, y$ at level p
`else if` $x \equiv 1(\text{mod } 2)$ `then`
$\quad \mathbf{u}^p(x,y) \leftarrow \frac{1}{2}[u^p(x-1,y) + u^p(x+1,y)], \quad \forall x, y$ at level p
`else if` $y \equiv 1(\text{mod } 2)$ `then`
$\quad \mathbf{u}^p(x,y) \leftarrow \frac{1}{2}[u^p(x,y-1) + u^p(x,y+1)], \quad \forall x, y$ at level p
`end`

As in the case of projection, there are a number of different warping schemes that can be employed. First, let us consider the so called "backwards warp". For this kind of warping, we take the flow vector at each site and treat it as the incoming flow vector to that site. The tail of the vector will likely fall someplace in between sites, hence we perform a bi-linear interpolation to obtain the new estimate. This process is illustrated in Figure 2.14 and can

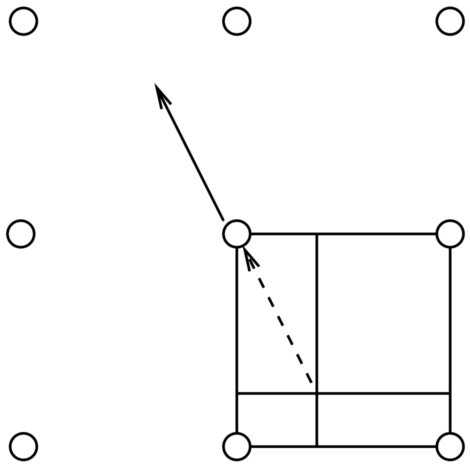


Figure 2.14: Backwards warping.

be expressed as:

$$I(x, y, t) \leftarrow I(x - u, y - v, t).$$

This scheme can easily be improved by the use of more sophisticated interpolation schemes; for example, bi-cubic spline interpolation.

In the case of a "forwards warp", the flow vector is used it to predict where the site will have moved. This kind of warping is more complex and computationally expensive, but more easily motivated than the backwards warp. We need to constrain the distance that any site can move by using a coarse-to-fine strategy. Then the scheme can be implemented by searching in a fixed neighborhood about a site to determine what flow vectors project to that neighborhood (see Figure 2.15). With such a scheme there may be collisions; that is, more than one flow vector may project to the same neighborhood. To determine the new value of a site, one needs a way to combine incoming flow vectors and resolve conflicts. Chapter 6 presents an implementation of such a warping scheme.

As Battiti *et al.,* [1991] have pointed out the simple coarse-to-fine approaches have a number of problems. In particular, if an error is made in estimating the motion at the coarse

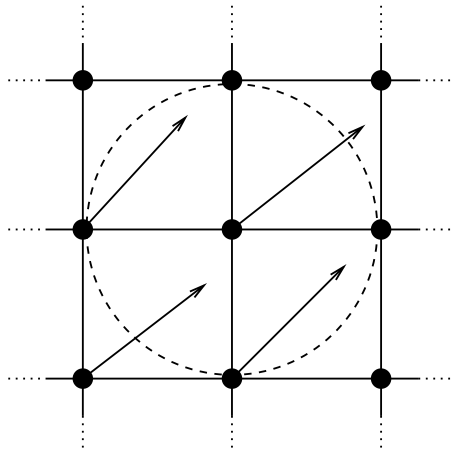


Figure 2.15: Forwards warping.

scale, it will be propagated down to the finer scales with no way to correct it resulting in *temporal aliasing*. More generally, the standard scheme does not provide criteria to decide what is the appropriate level in the pyramid at which to compute a particular velocity estimate. They suggest an adaptive scheme in which reliability estimates are used to determine the appropriate level at which to stop refining a particular flow estimate. Even such an adaptive scheme has limitations; for example, coarse-to-fine approaches cannot, in general, be used to estimate large motions of purely high frequency patterns.

The straightforward coarse-to-fine strategy is also not easily extended to incremental motion estimation. Chapter 5 develops two coarse-to-fine strategies that can be used in incremental estimation. The first approach is a *flow through* strategy which is a coarse-to-fine scheme without refinement. The motion is computed at each level of the pyramid independently and in parallel and then combined across levels using a strategy similar to that of Battiti *et al.* The second approach is a coarse-to-fine strategy, with refinement, that is appropriate for dynamic algorithms.

Chapter 3

Robust Estimation Framework

The previous chapter illustrated the generality of the problem posed by motion discontinuities; measurements are corrupted whenever information is pooled from a spatial neighborhood that spans a motion boundary. Both the data conservation and spatial coherence assumptions are affected by this problem.

We can view motion discontinuities as causing *violations* of the assumptions of data conservation and spatial coherence, and as such we would like to treat the violation of both constraints in a uniform manner. In particular, model violations such as these result in measurements that can be viewed in a statistical context as *outliers* [Hampel *et al.*, 1986; Huber, 1981]. The problem can be treated as one of recovering optical flow in the presense of these outliers and, hence, we appeal to the field of robust statistics which addresses the problem of estimation when the assumptions about the world are, by necessity, idealized and one expects that the assumptions will occasionally be violated.

This chapter formulates a framework for the robust estimation of optical flow. It begins by introducing robust estimation with an emphasis on on Hampel's [1986] approach based on influence functions. We draw on the ideas of robust estimation and influence functions in formulating problems in optical flow and illustrate the approach by reformulating regression, correlation, and explicit smoothness schemes in this framework.

This robust estimation approach is closely related to the traditional line-process approaches mentioned in the previous chapter. The notion of a line process, however, carries a spatial connotation and is applied only to the spatial smoothness term. Instead, we generalize the idea and define an *outlier process* that is applied to both the data and spatial terms. We will show how such a formulation in terms of binary or analog outlier processes can be converted into an equivalent robust estimation problem.

This relationship is explored more fully by showing how a wide class of robust estimation formulations admit physical interpretations in terms of binary or analog outlier processes. Following the work of Rangarajan and Chellapa [Rangarajan and Chellapa] we show how to construct equivalent outlier-process formulations for robust estimation problems.

3.1 Robust Statistics

The field of robust statistics [Hampel *et al.*, 1986; Huber, 1981] has developed to address the fact that the parametric models of classical statistics are often approximations of the phenomena being modeled. In particular, the field addresses how to handle *outliers*, or gross errors, which do not conform to the assumptions. While most of the work in computer vision has focused on developing optimal strategies for exact parametric models, there is a growing realization that we must be able to cope with situations for which our models were not designed.[1]

As identified by Hampel [1986, page 11] the main goals of robust statistics are:

(i) To describe the structure best fitting the bulk of the data.

(ii) To identify deviating data points (outliers) or deviating substructures for further treatment, if desired.

These goals mirror those we laid out in Chapter 1 and we will return to them after reviewing the fundamental ideas of robust estimation.

[1] As Einstein noted: "So far as mathematics is exact, it does not apply to nature; so far as it applies to nature, it is not exact."

To state the issue more concretely, robust statistics addresses the problem of finding the values for the parameters, $\mathbf{a} = [a_0, \ldots, a_n]$, that best fit a model, $\mathbf{u}(s; \mathbf{a})$, to a set of data measurements, $\mathbf{d} = \{d_0, d_1, \ldots, d_S\}$, in cases where the data differs statistically from the model assumptions.

In fitting a model, the goal is to find the values for the parameters, **a**, that minimize the size of the *residual* errors $(d_s - \mathbf{u}(s; \mathbf{a}))$:

$$\min_{\mathbf{a}} \sum_{s \in S} \rho(d_s - \mathbf{u}(s; \mathbf{a}), \sigma_s), \tag{3.1}$$

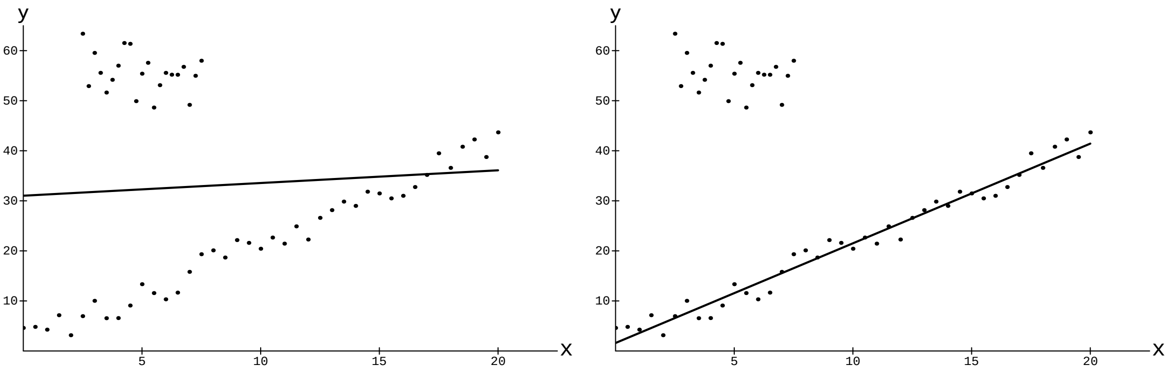
where σ_s is a scale parameter, which may or may not be present, and ρ is our *estimator*. When the errors in the measurements are normally distributed, the optimal estimator is the quadratic:

$$\rho(d_s - \mathbf{u}(s; \mathbf{a}), \sigma_s) = \frac{(d_s - \mathbf{u}(s; \mathbf{a}))^2}{2\sigma_s^2}, \tag{3.2}$$

which gives rise to the standard *least-squares* estimation problem. The function ρ is called an *M-estimator* since it corresponds to the *Maximum-likelihood* estimate. The *robustness* of a particular estimator refers to its insensitivity to outliers, or deviations, from the assumed statistical model.

An estimator is said to be robust if the solution to equation (3.1) is relatively insensitive to "small" deviations from the assumptions. The term "small" can have two meanings. The first refers to relatively small deviations for the bulk of the data and the second to large deviations for a few data points. The *breakdown point* of an estimator refers to the largest fraction of the data that can be arbitrarily bad and that will not cause the solution to be arbitrarily bad. For example, the least-squares approach has a breakdown point of 0% since introducing an arbitrarily bad outlier can produce arbitrarily bad estimates regardless of the sample size. A robust technique on the other hand may have a breakdown point of up to 50%; that is, the estimator can cope with up to 50% of the data being outliers.

Robust estimators are also characterized by their *efficiency*. The efficiency of a robust estimator refers to the ratio between the theoretically lowest variance achievable for a given



$y(x) = 0.25x + 31$ $\qquad$ $y(x) = 2x + 1.6$

a $\qquad$ *b*

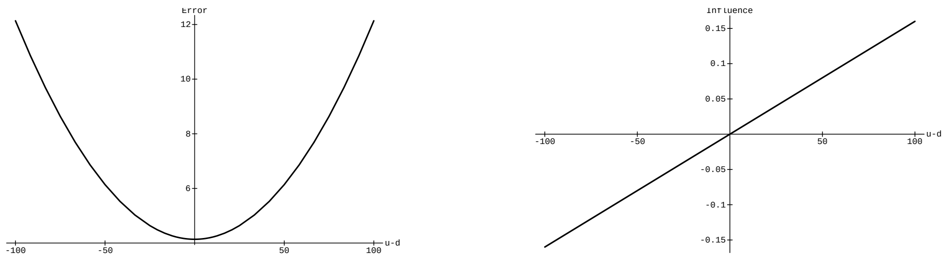
Figure 3.1: Fitting a straight line. The underlying model for the majority of the data is the line ($y(x) = 2.0x + 1.5$) to which uniform random noise over the range $[-3.25,\ 3.25]$ has been added. There are a number of outlying data points that do not conform to the model of a line. *a)* Least-squares fit. *b)* Robust fit (using Lorentzian estimator described in the text).

problem and the actual variance achieved using the robust estimator. For example, if the errors in the data are Gaussian, then the quadratic estimator is fully efficient, yielding the solution of theoretically minimum variance. One typically must trade off efficiency for robustness, but many robust estimators exist which are over 90% efficient.

3.1.1 Robust Estimators

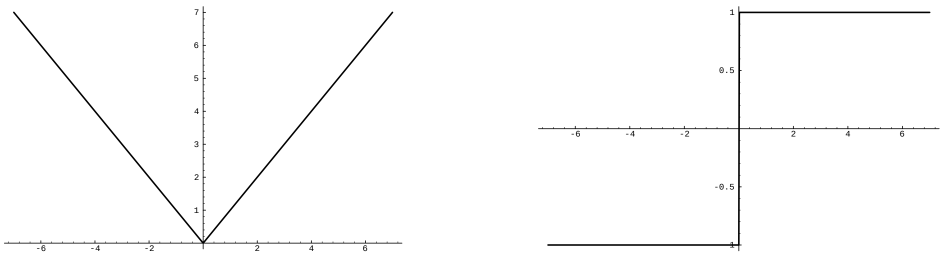
The least-squares approach is not without its problems. When the noise is not Gaussian, the estimate may be skewed from the "true" solution. Figure 3.1 shows an example of fitting a line to data in the presence of outliers. Figure 3.1a illustrates how the least-squares fit is skewed in the direction of the outliers. The fit recovered in figure 3.1b is robust in the sense that it rejects the outliers and recovers a "better" fit to the *majority* of the data.

The problem with the least-squares approach is that the outliers contribute "too much" to the overall solution. Outlying points are assigned a high weight by the quadratic estima-



a *b*

Figure 3.2: Quadratic estimator (a) and ψ-function (b).



a *b*

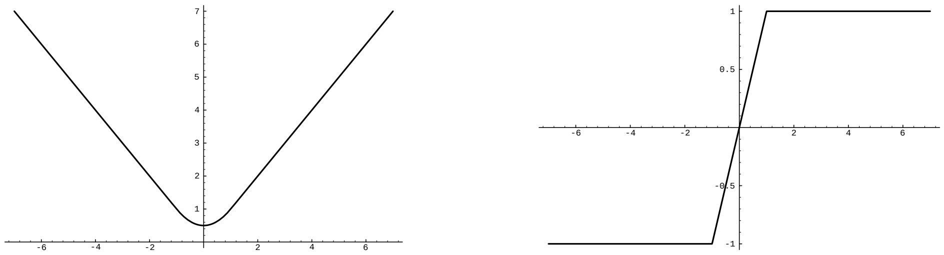
Figure 3.3: L1 norm. (a) Estimator, (b) ψ-function.

tor 3.2. To analyse the behavior of an estimator we take the approach of Hampel [1986] based on *influence functions*. The influence function characterizes the bias that a particular measurement has on the solution and is determined by the derivative, ψ, of the estimator [Hampel *et al.*, 1986]. Consider, for example, the quadratic estimator:

$$\rho(x) = x^2, \qquad \psi(x) = 2x. \tag{3.3}$$

For least-squares estimation, the influence of outliers increases linearly and without bound (Figure 3.2*b*).

To increase robustness, an estimator must be more forgiving about outlying measurements. The most obvious first step is to replace the quadratic (or L2 norm) with the absolute



a *b*

Figure 3.4: Huber's min-max estimator. (a) Estimator, (b) ψ-function.

value (or L1 norm):

$$\rho(x) = |x|, \qquad \psi(x) = \text{sign}(x). \tag{3.4}$$

In Figure 3.3 it is clear that outlying points are weighted less heavily by the L1 norm but the estimator is not robust in that it still has a breakdown point of 0%. Additionally, the L1 norm does not perform as well as the quadratic estimator when the errors are Gaussian. For this reason Huber [1981] proposed the *minimax* estimator (Figure 3.4):

$$\rho_\epsilon(x) = \begin{cases} x^2/2\epsilon + \epsilon/2 & |x| \leq \epsilon, \\ |x| & |x| > \epsilon, \end{cases} \qquad \psi_\epsilon(x) = \begin{cases} x/\epsilon, & |x| \leq \epsilon, \\ \text{sign}(x) & |x| > \epsilon. \end{cases} \tag{3.5}$$

Huber's minimax estimator combines the behavior of the L2 norm when the errors are small while maintaining the L1 norm's reduced sensitivity to outliers.[2]

To increase robustness further we will consider *redescending* estimators for which the influence of outliers tends to zero. There are two common examples, the first being *Andrews' sine:*

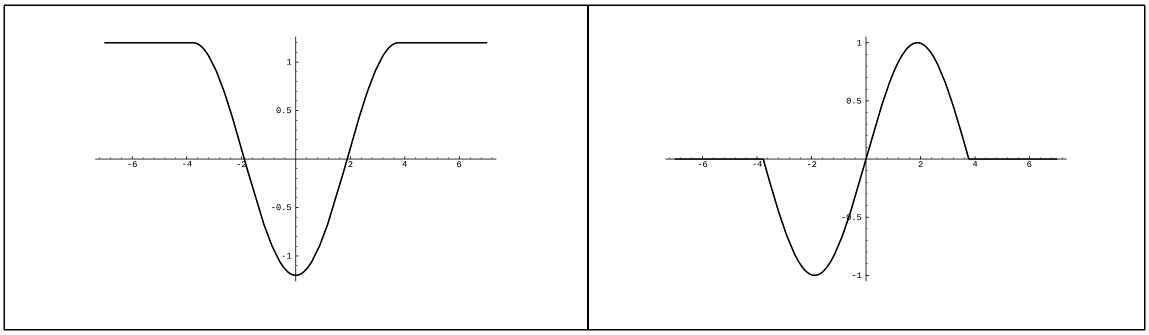
$$\rho(x,r) = \begin{cases} -r\cos(x/r) & |x| < \pi r, \\ r & \text{otherwise}, \end{cases} \qquad \psi(x,r) = \begin{cases} \sin(\frac{x}{r}) & |x| < \pi r, \\ 0 & \text{otherwise}. \end{cases} \tag{3.6}$$

The other commonly used estimator is *Tukey's biweight:*

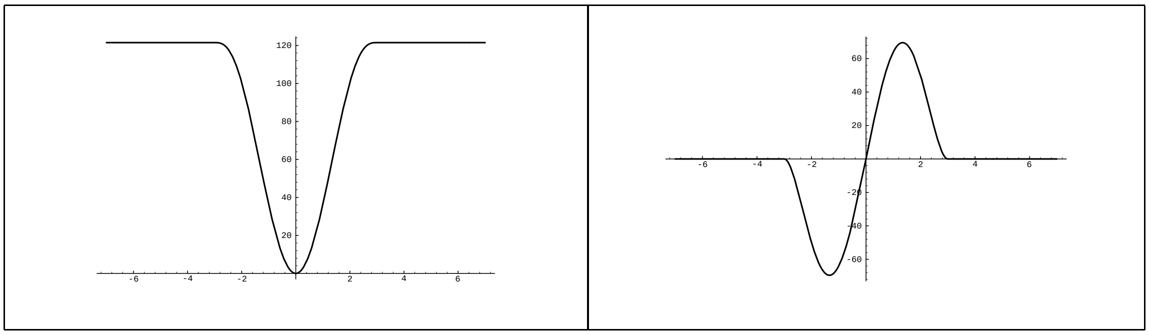
$$\rho(x,r) = \begin{cases} \frac{r^4x^2}{2} - \frac{r^2x^4}{2} + \frac{x^6}{6} & |x| < r, \\ \frac{r^6}{6} & \text{otherwise}, \end{cases} \qquad \psi(x,r) = \begin{cases} x(r^2 - x^2)^2 & |x| < r, \\ 0 & \text{otherwise}. \end{cases} \tag{3.7}$$

[2]The minimax ψ-function is often written as $\psi_\epsilon(x) = \min(\epsilon, \max(x, -\epsilon))$.

Andrews' Sine:



Tukey's Biweight:



Lorentzian:

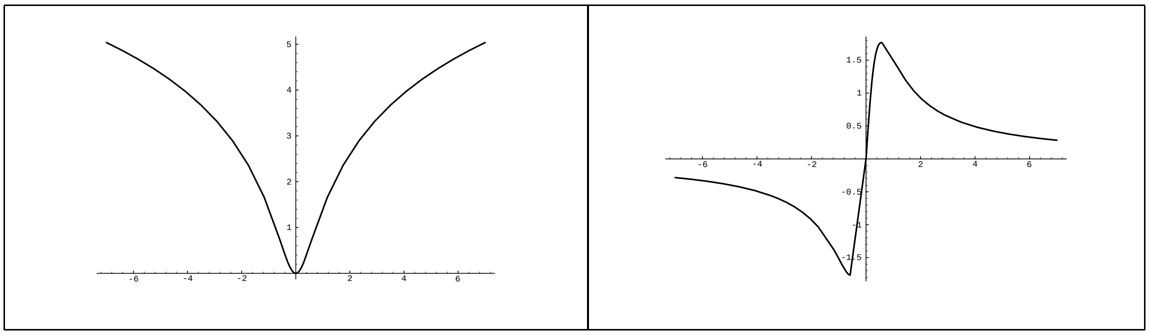


Figure 3.5: Redescending Estimators.

These estimators are plotted along with their ψ-functions in Figure 3.5. By examination of their ψ-functions we see that they both share a *saturating* property; that is, the influence of outliers tends to zero.

Another estimator with similar properties is the *Lorentzian:*

$$\rho_\sigma(x) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right), \qquad \psi_\sigma(x) = \frac{2x}{2\sigma^2 + x^2}. \tag{3.8}$$

Also illustrated in Figure 3.5, the Lorentzian is continuously differentiable, and its ψ-function has a very simple form. These properties make it attractive and we will come back to it in the following chapter where it will be used in developing a robust optical flow method.

3.2 Robust Estimation Framework

We now apply these tools of robust statistics to develop a framework for the robust estimation of optical flow. In Chapter 2 we introduced three standard approaches for estimating optical flow which were all posed in terms of least-squares estimation. We also indicated that the models used are idealizations that are frequently violated in practice and that the least-squares solutions are particularly sensitive to such violations. To improve the robustness, without sacrificing our simple models, we reformulate our minimization problems to account for outliers by using the robust estimators described above.

The regression approach is simply reformulated as:

$$\min_{u,v} E_D(u,v) \;\text{ where, }\; E_D(u,v) = \sum_{\mathcal{R}} \rho(I_x u + I_y v + I_t), \tag{3.9}$$

where ρ is a robust estimator. Similarly, we can reformulate correlation as the minimization of:

$$E_D(\mathbf{u}) = \sum_{(x,y)\in\mathcal{R}} \rho(I(x,y,t) - I(x + u\delta t, y + v\delta t, t + \delta t)). \tag{3.10}$$

The object function for the regularization approach becomes:

$$E(\mathbf{u}) = \sum_{s\in S}\left[\lambda\rho_1(I_x u + I_y v + I_t) + \sum_{t\in\mathcal{G}_s}\rho_2(\|\mathbf{u}_s - \mathbf{u}_t\|)\right], \tag{3.11}$$

where, in the regularization case, ρ_1 and ρ_2 may be different estimators. Such an approach can likewise be taken with many other early vision problems that are formulated in terms of least-squares optimization.

Notice what we have done. We have simply taken the standard formulations of optical flow and made the observation that these correspond to least-squares regression. Because each approach involves pooling information over a spatial neighborhood these least-squares formulations are inappropriate. By treating the problems in terms of robust estimation, we hope to alleviate the problems of oversmoothing and noise sensitivity typically associated with these approaches. In the case of the regression approach, the reformulation in terms of robust regression is not terribly surprising. Of greater interest is the robust formulation of the regularization approach. As pointed out in Chapter 2, the first-order regularization term corresponds to a locally constant model of optical flow, and minimizing the first-order formulation produces a least-squares estimate with respect to this model. This estimate is simply the mean flow in a neighborhood which is not robust and results in oversmoothing.

The relationship between regularization with discontinuities, regression, and outlier rejection has only recently become evident. Besl [1988] formulated neighborhood smoothing operations using robust techniques, but did not address regularization. Schunck later noted that standard first-order "regularization is a least-squares method and the algorithm produced by regularization averages data over local neighborhoods," [Schunck, 1990, page 6]. Schunck, however, did not formulate the regularization term using robust estimators. Independently, Black and Anandan [1990b] formulated the optical flow problem using robust estimators for both data and regularization terms. Black and Anandan [1991b] latter clarified the connection between outlier rejection and regularization with line processes. At about the same time Blauer and Levine [1991] were investigating regularization with the more robust L1 smoothness metric. And, more recently, Black [1992b] has shown that the standard Horn and Schunck formulation of optical flow corresponds to a least-squares regression problem, and by simply reformulating the problem as one of robust regression, the problems of oversmoothing and noise are drastically reduced.

This approach has a number of advantages. The formulation of minimization problems in the robust estimation framework is very similar to the familiar least-squares formulations and this gives the robust formulations an intuitive appeal. Robust estimators and their influence functions are also powerful formal and qualitative tools that are useful for analyzing robustness. In particular, examination of the influence functions provides a means of comparing estimators and their effects on outliers.

Within the robust estimation framework, it is natural to "robustify" both the data conservation and spatial coherence terms. With traditional line-process or weak-continuity approaches this has not been the case. These previous approaches have focused on the spatial term while maintaining a quadratic estimator for the data term. And, as we will see, the explicit use of robust estimators brings to light the relationships between the robust estimation framework and these previous approaches. Additionally, an analysis of a robust estimation problem can illuminate possible underlying physical interpretations in terms of binary or analog line processes.

Recall the three goals we set out for robust optical flow in Chapter 1:

1. Recover optical flow without smoothing across motion discontinuities.

2. Locate the actual motion boundaries so that they are available to other algorithms that require knowledge about the surface boundaries of objects.

3. Detect when the underlying assumptions of the model are violated.

These are very similar to the goals of robust statistics. The first corresponds to Hampel's first goal; that is, to recover the solution best fitting the "bulk of the data." When trying to recover the flow within a neighborhood that spans a motion discontinuity, there will be two conflicting sets of measurements corresponding to the two motions. By adopting the robust estimation framework, we can recover the dominant motion and ignore the other measurements as outliers.[3]

In satisfying the first goal we have identified the spatial outliers and, in so doing, have implicitly determined the existence of a motion boundary. By examining the outliers, the motion boundary can be recovered.

These first two goals are really special cases of the third goal. However, unlike most previous approaches, we are not simply interested in violations of the spatial coherence assumption. The robust estimation framework allows us to detect violations of the data coherence term as well which, as we will see, can significantly improve motion estimates. More generally, detecting where assumptions are violated, may, as Hampel suggests, allow algorithms to perform further processing in interesting regions of the image.

3.2.1 Minimization

The approach above decouples the problem of formulating an optimization problem robustly from the problem of recovering the solution. The minimization problem can be treated as a separate issue and a host of techniques can be brought to bear on the problem. In this thesis we consider two general schemes. In the case where our objective function is differentiable, local optimization is performed using Newton's method. In the case where the objective function is not differentiable, we will use a stochastic minimization scheme, but will defer describing the approach until Chapter 6.

For now, we will describe the general gradient-descent method, and, in the following chapter provide details of the method applied to various approaches. Given the following optimization problem,

$$\min_x f(x), \tag{3.12}$$

[3]This ignores the case where the discontinuity divides the neighborhood exactly in half so that 50% of the measurements are due to each motion. In this case, no technique will work well, and it is not even clear what the "correct" motion estimate should be.

we first approximate f by its Taylor series expansion,

$$f(x + \delta x) \approx f(x) + f'(x)\delta x + \frac{f''(x)}{2}\delta x^2 + \ldots . \tag{3.13}$$

To minimize f we drop the terms above first order, take the derivative, and set it equal to zero:

$$\frac{df}{dx} = 0 = f'(x) + f''(x)\delta x. \tag{3.14}$$

We then obtain the following iterative update equation,

$$x_{n+1} = x_n + \delta x \tag{3.15}$$

where,

$$\delta x = -\frac{f'(x)}{f''(x)} \tag{3.16}$$

Unlike the least-squares estimation formulation, the robust objective function is not guaranteed to be convex. This means that a local optimization procedure like the above may get "stuck" in local minima. To perform global optimization we can use a continuation method like Graduated Non-Convexity [Blake and Zisserman, 1987] as will be shown in the next chapter.

3.2.2 Other Robust Approaches

There are other approaches to robust statistics that have been applied to computer vision problems. For example, Irani *et al.* [1992] formulated an area regression technique that performs iterative outlier rejection. The strategy is to first obtain a least-squares estimate of the motion which hopefully corresponds to the bulk of the data. Then outliers can be identified by examining the residuals. Some number of outliers are removed and a new least-squares estimate is found. This iterative scheme can work well when the outliers are not extremely bad or do not occupy much of the image.

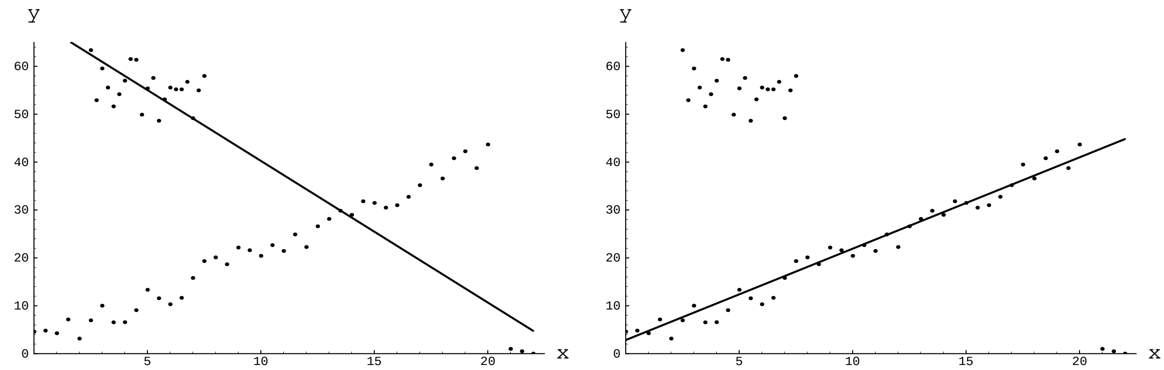


Figure 3.6: Problems with iterative outlier rejection (see text).

If, however, there are a large number of outliers, or the outliers fall far from the true solution, then the least squares estimate may be very poor. If the initial estimate is poor, outliers may have small residuals while the residuals of some true measurements look like outliers. In such a situation a simple iterative technique could lock onto an incorrect solution and reject the true measurements. This can be seen in Figure 3.6 where the data is the same as that in Figure 3.1 with the addition of a few outliers in the lower right. On the left we have used the iterative approach described for detecting and rejecting outliers. If the initial estimates are bad, the final result can be drastically poor. On the right is the result achieved using the robust estimator formulation with the Lorentzian estimator.

There are other avenues by which to approach the problem of robustness. One avenue that has become popular in computer vision involves the use of robust iterative procedures like least-median-of-squares (LMedS).[4] Model parameters are estimated by minimizing the median of the squares of the residuals:

$$\min_{\mathbf{a}} \mathrm{med}_s (d_s - \mathbf{u}(s; \mathbf{a}))^2. \tag{3.17}$$

The approach achieves a breakdown point of 50% due to the fact that the median can tolerate

[4]See [Rousseeuw and Leroy, 1987] for a complete description or [Meer *et al.*, 1991] for applications to computer vision.

up to half of the data being outliers.

Another approach to solving the robust estimation problem is to convert it into an equivalent iteratively reweighted least squares (IRLS) problem [Beaton and Tukey, 1974; Campbell, 1980]. The least-squares approach tries to find parameters, **a**, that produce small residuals, $(d_s - \mathbf{u}(s; \mathbf{a}))$. The idea of IRLS is to assign weights, w_s, to the residuals at each site, s, in a region, $\mathcal{R}$, where the weights control the influence of the residuals. High weights (approaching unity) are assigned to "good" data and lower weights to outlying data.

The M-estimation problem is first converted into an equivalent weighted least-squares problem:

$$\sum_{s \in \mathcal{R}} \rho(d_s - \mathbf{u}(s; \mathbf{a})) = \sum_{s \in \mathcal{R}} w_s (d_s - \mathbf{u}(s; \mathbf{a}))^2. \tag{3.18}$$

To minimize, we take the derivatives of both sides and set them equal to zero:

$$\sum_{s \in \mathcal{R}} \psi(d_s - \mathbf{u}(s; \mathbf{a})) = 2 \sum_{s \in R} w_s (d_s - \mathbf{u}(s; \mathbf{a})) = 0. \tag{3.19}$$

The weights are then given by,

$$w_s = \frac{\psi(d_s - \mathbf{u}(s; \mathbf{a}))}{d_s - \mathbf{u}(s; \mathbf{a})}. \tag{3.20}$$

The first step in the iterative solution of the weighted least-squares problem is to compute an initial estimate for the parameters, **a**. This can be done by setting the $w_s = 1$ and solving the unweighted least-squares problem. This initial estimate is then used to compute the weights w_s. These values are used to compute the weighted least-squares solution [Strang, 1976]. The weights are then updated and the process is repeated until some termination condition is met.

We prefer the straightforward formulation in terms of a minimization problem with robust estimators. Such an approach has two main advantages. First, standard global optimization techniques like continuation methods and stochastic minimization can be immediately applied to the problem. Second, as we will see in the next sections, our formulation allows us to clearly identify the relationships between the robust estimation framework and line-process approaches.

3.3 From Line Processes to Robust Estimation

Recall the line-process formulation of the optical flow problem where we take E_D to be the gradient formulation of the intensity constraint equation:

$$\begin{aligned} E(\mathbf{u}, l) &= \sum_{s \in S} [(I_x u_s + I_y v_s + I_t)^2 \\ &\quad + \sum_{t \in \mathcal{G}_s} [\alpha_S (1 - l_{s,t}) \|\mathbf{u}_s - \mathbf{u}_t\|^2 + \beta_S l_{s,t}]], \end{aligned} \tag{3.21}$$

where $l_{s,t}$ is a binary-valued line process, and where α_S and β_S are constant factors controlling the weighting of the smoothness term and the penalty term respectively. Such a formulation allows violations of the spatial smoothness term, but does not account for violations of the data term. As was shown in Chapter 2, the assumptions of the data term are also frequently violated in practice.

This prompts us to generalize the notion of a "line process" to that of an "outlier process" that can be applied to both data and spatial terms. The motivation behind such a generalization is to formulate a process that performs *outlier rejection* in the same spirit as the robust estimators do. The optical flow problem is then reformulated using outlier processes as follows:

$$\begin{aligned} E(\mathbf{u}, l, d) &= \sum_{s \in S} [\alpha_D (1 - d_s)(I_x u_s + I_y v_s + I_t)^2 + \beta_D d_s \\ &\quad + \sum_{t \in \mathcal{G}_s} [\alpha_S (1 - l_{s,t}) \|\mathbf{u}_s - \mathbf{u}_t\|^2 + \beta_S l_{s,t}]], \end{aligned} \tag{3.22}$$

where we have simply introduced a new process d_s and constant scaling factors α_D β_D. This process allows us to ignore erroneous information from the data term.

3.3.1 Eliminating the Outlier Process

The formulation above leads to an expensive joint estimation problem where one not only has to estimate **u** but also the binary outlier processes d and l. In the case of the simple binary line-process formulation, Blake and Zisserman [1987] showed how the line variables can be removed from the equation by first minimizing over them. They obtain a new objective function that is solely a function of **u**. Exactly the same treatment can be applied to the outlier-process version.

First, we rewrite equation (3.22) as:

$$\begin{aligned} E(\mathbf{u}, l, d) &= \sum_{s \in S} h((I_x u_s + I_y v_s + I_t), d_s, \alpha_D, \beta_D) + \\ & \qquad \sum_{s \in S} \sum_{t \in \mathcal{G}_s} h(\|\mathbf{u}_s - \mathbf{u}_t\|, l_{s,t}, \alpha_S, \beta_S), \end{aligned} \tag{3.23}$$

where,

$$h(x, p, \alpha, \beta) = \alpha(1-p)x^2 + \beta p, \tag{3.24}$$

and where p is an outlier process.

The optimization problem is then,

$$\begin{aligned} \min_{\mathbf{u}_s, l_{s,t}, d_s} & \sum_{s \in S} h((I_x u_s + I_y v_s + I_t), d_s, \alpha_D, \beta_D) \\ & \quad + \sum_{s \in S} \sum_{t \in \mathcal{G}_s} h(\|\mathbf{u}_s - \mathbf{u}_t\|, l_{s,t}, \alpha_S, \beta_S). \end{aligned} \tag{3.25}$$

Notice that the first term does not depend on l and the second term does not depend on d. Thus we can rewrite the equation as,

$$\begin{aligned} \min_{\mathbf{u}_s} \Bigg[\min_{d_s} \left(\sum_{s \in S} h((I_x u_s + I_y v_s + I_t), d_s, \alpha_D, \beta_D) \right) + \\ \min_{l_{s,t}} \left(\sum_{s \in S} \sum_{t \in \mathcal{G}_s} h(\|\mathbf{u}_s - \mathbf{u}_t\|, l_{s,t}, \alpha_S, \beta_S) \right) \Bigg]. \end{aligned} \tag{3.26}$$

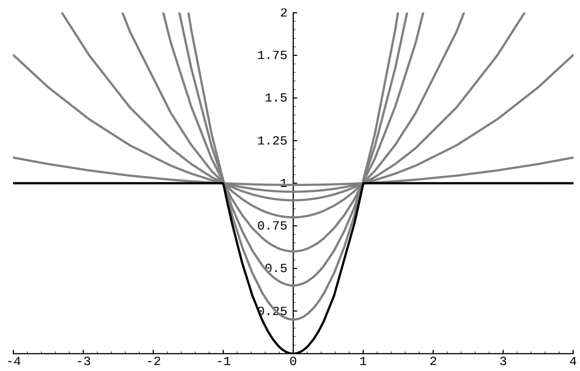


Figure 3.7: Family of quadratics with a common intersection. The infimum of this of this family is the *truncated quadratic* shown in bold.

Figure 3.7 shows the function $h(x, p, 1, 1)$ plotted for various values of the outlier process p, and the infimum of this family of quadratics plotted in bold.[5] This bold curve is the standard *truncated quadratic* used by Blake and Zisserman [1987]:

$$\rho(x, \alpha, \beta) = \begin{cases} \alpha x^2 & \text{if } |x| < \sqrt{\beta}/\sqrt{\alpha}, \\ \beta & \text{otherwise,} \end{cases} \tag{3.27}$$

$$\psi(x, \alpha, \beta) = \begin{cases} 2\alpha x & \text{if } |x| < \sqrt{\beta}/\sqrt{\alpha}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.28}$$

We can now eliminate the outlier processes from the equation and rewrite it in terms of the truncated quadratic:

$$\min_{\mathbf{u}} \sum_{s \in S} [\rho((I_x u_s + I_y v_s + I_t), \alpha_D, \beta_D) + \sum_{t \in \mathcal{G}_s} \rho(\|\mathbf{u}_s - \mathbf{u}_t\|, \alpha_S, \beta_S)]. \tag{3.29}$$

Notice that this is identical to the robust estimation formulation with the truncated quadratic as the estimator. This is one of the common redescending estimators used in robust statistics. Up to a fixed threshold, errors are weighted quadratically, but beyond that the estimator has a saturating property; errors receive a constant value. By examining the ψ-function (Figure 3.8) we see that the influence of outliers goes to zero beyond the threshold.

[5] While the figure shows a family of quadratics for $0 \leq p \leq 1$, the infimum is determined by the cases where $p = 0$ or $p = 1$. In this case, p is simply a binary valued line process.

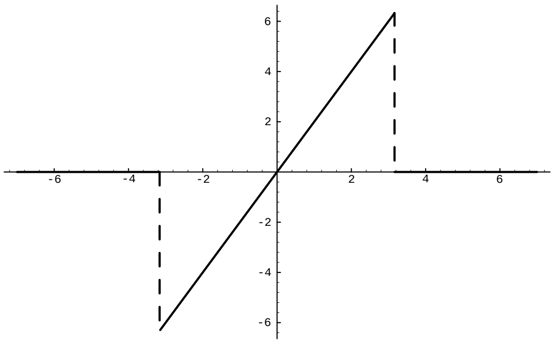


Figure 3.8: Truncated quadratic ψ-function.

Analog Outlier Processes

Geman and Reynolds [1992] showed that this approach can be generalized to *analog* line processes that assume continuous nonnegative values, $0 \leq p \leq 1$. Now if we modify the equation leading to the truncated quadratic by allowing the α and β above to be functions of the outlier process rather than constants, we have:

$$h(x, p) = \alpha(p)x^2 + \beta(p).$$

Geman and Reynolds [1992] show that choosing $\beta(p)$ such that:

$$\beta(0) = 0,$$
$$\beta(p) \ \text{ is strictly decreasing},$$
$$\beta(1) = -1,$$

and $\alpha(0) = 0$ with α increasing, results in a family of quadratics. The envelope of this family of quadratics is defined by taking the infimum:

$$\rho(x) = \inf_{0 \leq b \leq 1} (\alpha(p)x^2 + \beta(p)).$$

As in the case of binary line processes, performing the minimization over the line processes results in an estimator, ρ, where the line processes have been removed. For example,

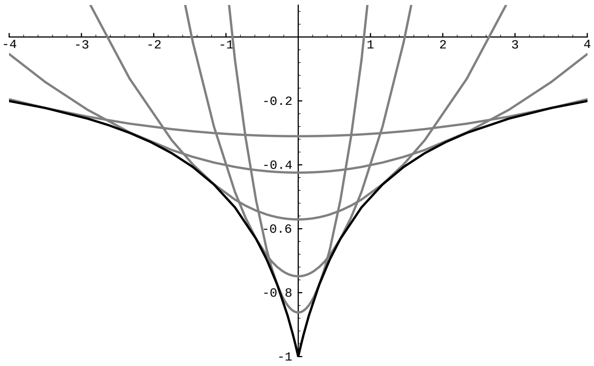


Figure 3.9: Geman and Reynolds estimator. Unlike the truncated quadratic, this family of quadratics has no common intersection.

Geman and Reynolds show that choosing:

$$\alpha(p) = \frac{p^{3/2}}{2(1 - p^{1/2})}, \quad \beta(p) = \frac{p - 3p^{1/2}}{2},$$

produces a family of quadratics whose infimum produces the following robust estimator:

$$\rho(x) = \frac{-1}{1 + |x|},$$

as shown in Figure 3.9.

This new ρ function has many of the same properties of the truncated quadratic; in particular, it has the saturating property of redescending estimators. This can be seen by examining its ψ-function (Figure 3.10). Notice that the strictly concave nature of ρ leads to an ψ-function where everything except a perfect match of model to data is treated as an outlier. Geman and Reynolds [1992] point out that this concavity on $(0, \infty)$ results in an estimator which, unlike the quadratic, does not interpolate across image transitions. Additionally, since the influence of outliers goes to zero, the estimator does not introduce a bias against large discontinuities.

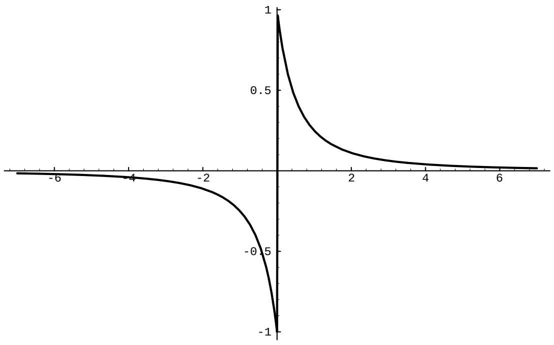


Figure 3.10: ψ-function for the Geman and Reynolds estimator

.

3.4 From Robust Estimators to Line Processes

The previous section showed that, given a formulation in terms of binary or analog outlier processes, we can derive an equivalent formulation in the robust estimation framework. This is only half the story. For certain choices of robust estimators, we can convert a robust estimation problem into an equivalent problem involving binary or analog outlier processes.

In this section we summarize the results of Rangarajan and Chellapa [Rangarajan and Chellapa] and apply them to the problem of recovering an equivalent formulation of the robust gradient equation with an analog outlier process. Recall, for example, the robust gradient formulation of optical flow:

$$E(u,v) = \sum_{m \in S} [\lambda \rho(I_x u_m + I_y v_m + I_t) + \sum_{n \in \mathcal{G}_m} \rho(\|\mathbf{u}_m - \mathbf{u}_n\|)], \tag{3.30}$$

where $\rho(x)$ is, for example, the Lorentzian estimator. We will first derive an equivalent cost function that contains a new process **s**. We then define the outlier processes in terms of this process **s**. This allows equation (3.30) to be rewritten in terms of outlier processes.

3.4.1 An equivalent objective function

We begin by introducing a new cost function, $\rho_s(t, s)$, which contains an unobservable process s. In our case we introduce two new processes, $\mathbf{s} = [s_d, s_{\mathbf{u}}]$, to account for violations of the data and spatial terms. The objective function is rewritten in terms of ρ_s as follows:

$$\begin{aligned} E(u, v, \mathbf{s}) \quad = \quad & \sum_{m \in S} [\lambda \rho_s(I_x u_m + I_y v_m + I_t, s_d) \\ & \qquad + \sum_{n \in \mathcal{G}_m} \rho_s(\|\mathbf{u}_m - \mathbf{u}_n\|, s_{\mathbf{u}})]. \end{aligned} \tag{3.31}$$

Our goal is for the minimum of (3.31) to be the same as the minimum of (3.30); that is, the introduction of the processes $s_d, s_{\mathbf{u}}$ should not change the solution. To achieve this, we must find the correct function ρ_s.

Both processes, s_d and $s_{\mathbf{u}}$, are treated identically, so for simplicity, we will drop the subscripts in the following discussion. Since s is going to be related to an outlier process we assume that $s \geq 0$. We also assume that the value, $\hat{s}$, that minimizes $\rho_s(t, s)$ is $\hat{s} = t^2$. Now let $\phi(s) = \rho(\sqrt{s})$ and $\phi'(s)$ be the derivative with respect to s. These assumptions mean that, like an outlier process, s is insensitive to the sign of the error.

We then minimize with respect to **s** and eliminate it from the cost function. If the result gives us our original cost function (3.30) then the introduction of **s** has not changed the solution to our minimization problem. For this to be the case, we need to find a function ρ_s such that the following conditions are met:

$$\min_s \rho_s(t, s) = \phi(t^2) = \rho(t) \;\text{ and }\; \hat{s} = \arg\min_s \rho_s(t, s) = t^2.$$

Rangarajan and Chellapa show that a function satisfying these requirements is:

$$\rho_s(t, s) = (t^2 - s)\phi'(s) + \phi(s), \tag{3.32}$$

if and only if:

$$\phi''(s) < 0, \quad s \geq 0. \tag{3.33}$$

As an example, we now consider the Lorentzian estimator; recall:

$$\rho(x) \quad = \quad \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right),$$

for which we derive the following functions:

$$\phi(s) \quad = \quad \log(1+s), \tag{3.34}$$

$$\phi'(s) \quad = \quad \frac{1}{1+s}, \tag{3.35}$$

$$\phi''(s) \quad = \quad -\frac{1}{(1+s)^2}. \tag{3.36}$$

We see that for the Lorentzian estimator $\phi''(s)$ satisfies condition (3.33) and hence if we take,

$$\rho_s(t,s) = \frac{t^2 - s}{1+s} + \log(1+s), \tag{3.37}$$

then the introduction of the unobservable **s** does not effect the optimal motion estimate.

3.4.2 Recovering the outlier process

Above, we introduced an unobservable process **s** to derive an equivalent objective function (3.31). We now show how to use s to construct an analog outlier process $z \in [0,1]$. There are many choices, but a reasonable set of properties for an analog outlier process is:

1. $z(0) = 0$,
2. $z(\infty) = 1$,
3. $z'(s) \geq 0$ with equality only at $s = 0$ and $s = \infty$.

The third property is required for the transformation to be monotonic and is guaranteed by the concavity of ϕ; $(\phi''(s) < 0,\ s \geq 0)$.

Rangarajan and Chellapa show that the function:

$$z(s) = 1 - \phi'(s), \tag{3.38}$$

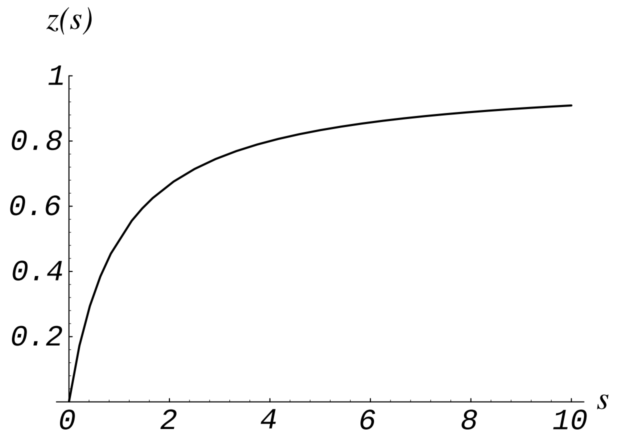


Figure 3.11: Analog Line process $z(s),\ s \geq 0$, for the Lorentzian estimator.

satisfies these properties when $\phi'(0) = 1$ and $\phi'(\infty) = 0$. In the case of the Lorentzian estimator we see that equation (3.36) satisfies these conditions as well as the concavity condition (3.33). Figure 3.11 illustrates the Lorentzian outlier process:

$$z(s) = 1 - \frac{1}{1+s}. \tag{3.39}$$

We now can construct an equivalent outlier-process formulation of our minimization problem. Recall that above we introduced a new cost function ρ_s that depended on a new process s. When,

$$\rho_s(t, s) = (t^2 - s)\phi'(s) + \phi(s),$$

the minimum of the objective function was unchanged by introducing the process s.

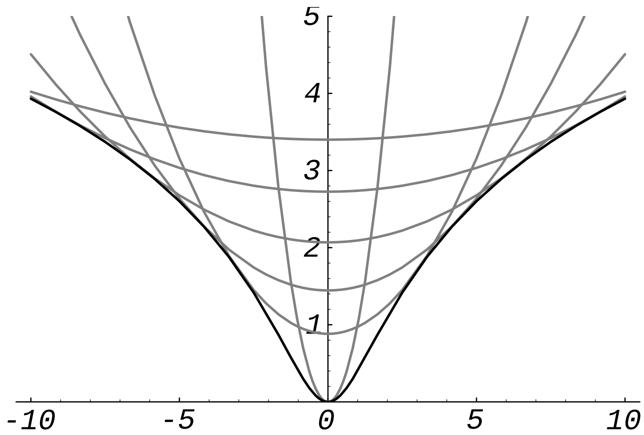


Figure 3.12: The function $\rho_s(t, s)$ is plotted for various values of s. The Lorentzian estimator ρ is the infimum of this family of functions.

Now, given the definition of the line process z in terms of s, we can rewrite the function ρ_s in terms of z:

$$\begin{aligned}
\rho_s(t, s) &= (t^2 - s)\phi'(s) + \phi(s) \\
&= t^2\phi'(s) - s\phi'(s) + \phi(s) \\
&= t^2(1 - z(s)) - s(1 - z(s)) + \phi(s) \\
&= t^2(1 - z(s)) + P(s), \qquad (3.40)
\end{aligned}$$

where $P(s) = \phi(s) - s(1 - z(s))$ can be thought of as the "penalty" for introducing a discontinuity. Figure 3.12 shows $\rho_s(t, s)$ plotted for various values of s. The figure also plots the Lorentzian estimator ρ which corresponds to the infimum of the $\rho_s(t, s)$ functions.[6]

Finally, we rewrite the robust optical flow equation (3.31) in terms of the outlier process z by simply substituting the outlier process $t^2(1 - z(s)) + P(s)$ for ρ_s in the objective

[6]For a similar treatment, the reader is refered to the work of Geman and Reynolds [1992].

function:

$$
\begin{aligned}
E(u, v, \mathbf{s}) &= \sum_{m \in S} [\lambda \rho_s(I_x u_m + I_y v_m + I_t, s_d) \\
&\quad + \sum_{n \in \mathcal{G}_m} \rho_s(\|\mathbf{u}_m - \mathbf{u}_n\|, s_{\mathbf{u}}), \\
&= \sum_{m \in S} [\lambda (I_x u_m + I_y v_m + I_t)^2 (1 - z(s_d)) + P(s_d) \\
&\quad + \sum_{n \in \mathcal{G}_m} (\|\mathbf{u}_m - \mathbf{u}_n\|^2 (1 - z(s_{\mathbf{u}})) + P(s_{\mathbf{u}}))].
\end{aligned}
\tag{3.41}
$$

Notice that substituting in the line process z does not change the minimum of the objective function.

3.5 Choosing an Estimator

One issue that arises with the robust estimation framework is how one chooses the appropriate estimator for a given problem. The answer will depend on a number of factors. First, one must consider the optimization scheme used to minimize the objective function. The scheme we use requires that the estimator be twice differentiable. For other implementations the issue may be the practicality of implementing the estimators, or their outlier-process formulations, in VLSI networks [Harris *et al.*, 1990; Hutchinson *et al.*, 1988; Koch *et al.*, 1988].

One may also require that the estimator meet the criteria necessary for there to be an equivalent outlier-process formulation. The robust formulation can then be extended by adding interactions between the spatial line processes in the standard way.

One may also have some incomplete knowledge about the distribution of measurement errors. In this case more statistically motivated estimators can be chosen. For example, a *contaminated Gaussian* model [Durrant-Whyte, 1987] can approximate the behavior of the truncated quadratic by varying the parameters:

$$
p(d_s | u_s) = \frac{1 - \epsilon}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(d_s - u_s)^2}{2\sigma_1^2}\right) + \frac{\epsilon}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(d_s - u_s)^2}{2\sigma_2^2}\right) \tag{3.42}
$$

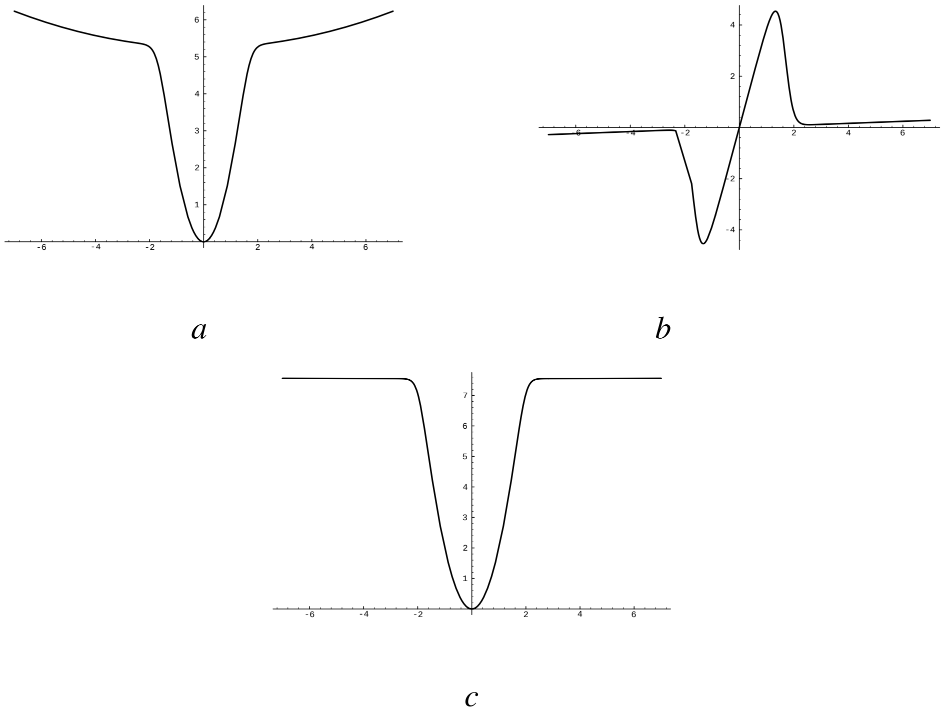


Figure 3.13: Contaminated Gaussian. *a)* shows the estimator for $\epsilon = 0.5$, $\sigma_1 = 0.5$, and $\sigma_2 = 5.0$. *b)* the ψ-function. *c)* with $\sigma_2 = 50.0$, the estimator becomes more like the truncated quadratic.

where $\sigma_1 << \sigma_2$ and $\epsilon << 1.0$. This model assumes that measurements are typically governed by a Gaussian distribution with small variance, but occasionally are characterized by a Gaussian with a large variance. The negative logarithm of this distribution gives the estimator in Figure 3.13a. By examining the ψ-function in Figure 3.13b we see that this estimator does not technically have the saturating property that we desire, but, by adjusting the parameters, we can achieve a reasonable approximation to the truncated quadratic 3.13c.

For global optimization with continuation methods, the estimator must have a control parameter that controls the shape of the function and the amount of outlier rejection performed. The truncated quadratic does not naturally have this property, and consequently Blake and Zisserman had to construct an approximation which could be controled.

Geiger and Girosi [1991] derive another approximation to the truncated quadratic based on a *mean field* formulation of the optimization problem. Instead of minimizing over the line processes, they integrate them out and derive the following estimator:

$$\rho_\beta(x) = \alpha - \frac{1}{\beta}\log(1 + e^{-\beta(\lambda x^2 - \alpha)}), \qquad \psi_\beta(x) = \frac{2e^{\beta(\alpha - \lambda x^2)}\lambda x}{1 + e^{\beta(\alpha-\lambda x^2)}}, \tag{3.43}$$

where β is a thermodynamically motivated control parameter which controls the shape of the function. This parameter, β, is the same as the inverse of the temperature parameter used in simulated annealing. Figure 3.14 shows the mean field estimator for various values of β. For small β the function behaves like the quadratic while, as β goes to infinity, the function approaches the shape of the truncated quadratic.

Leclerc [1989] derived another estimator by starting with a different formulation of the smoothness term. Consider the following objective function:

$$E(\mathbf{u}) = \sum_{s\in S} E_D(\mathbf{u}) + \sum_{t\in\mathcal{G}_s}(1 - \delta(\|\mathbf{u}_s - \mathbf{u}_t\|)). \tag{3.44}$$

where $\delta(x)$ is the Kronecker delta function,

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.45}$$

Leclerc developed a continuation strategy by approximating the delta function with the estimator (Figure 3.15),

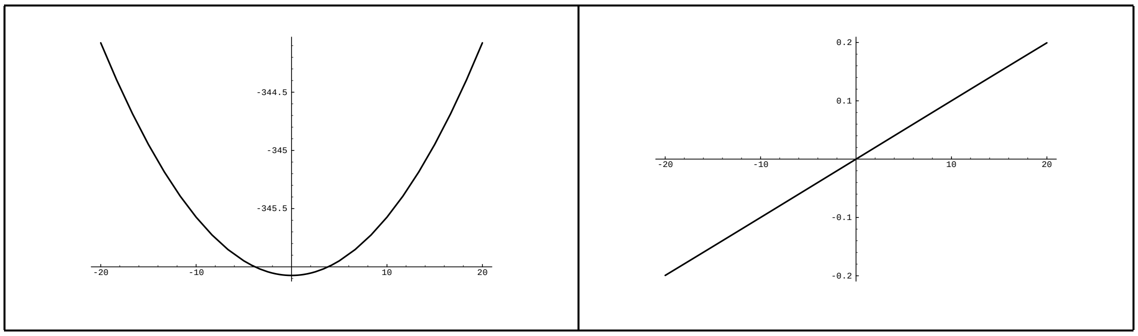
$$\rho_{\eta,\sigma}(x) = 1 - e^{-\frac{x^2}{(\eta\sigma)^2}}, \qquad \psi_{\eta,\sigma}(x) = \frac{2x}{(\eta\sigma)^2}e^{-\frac{x^2}{(\rho\sigma)^2}}. \tag{3.46}$$

As η goes to zero, the estimator approaches the delta function.

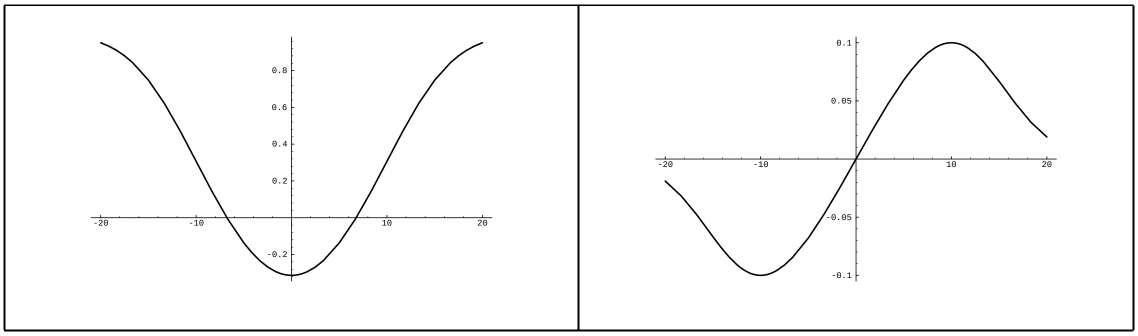
The Lorentzian estimator introduced above satisfies many of the criteria that we have mentioned. For example, it is twice differentiable and, as we saw, admits an analog line process. It also has a probabilistic interpretation in that it is an optimal estimator if the error distribution is Cauchy:

$$\text{Prob}(d_s - u_s, \sigma_s) \sim \frac{1}{1 + \frac{1}{2}\left(\frac{d_s - u_s}{\sigma_s}\right)^2}. \tag{3.47}$$

$\beta \to 0.0$



$\beta = 1.0$



$\beta \to \infty$

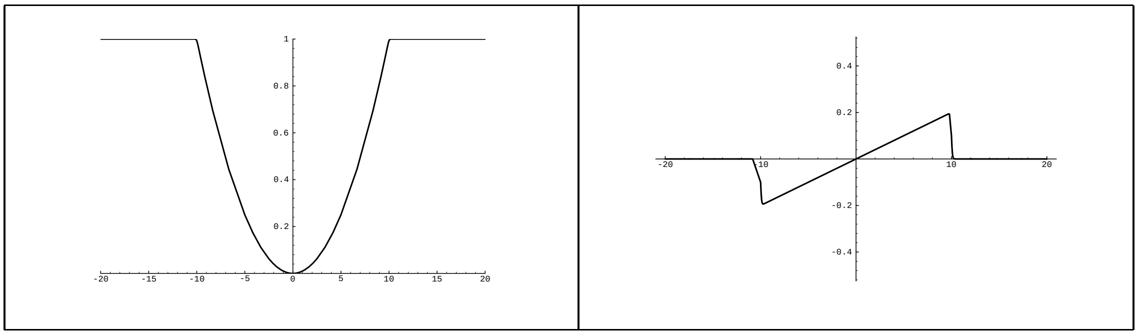


Figure 3.14: Mean Field Estimator

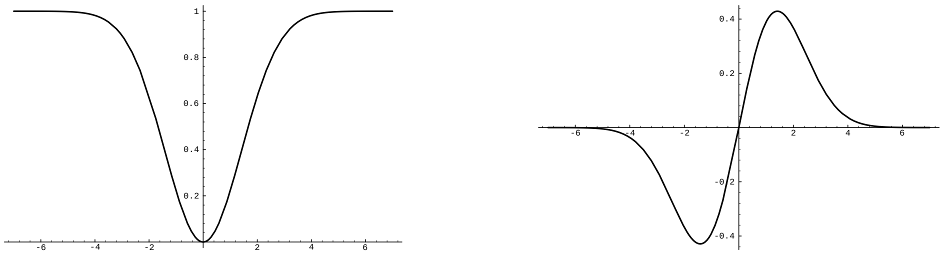


Figure 3.15: Leclerc estimator.

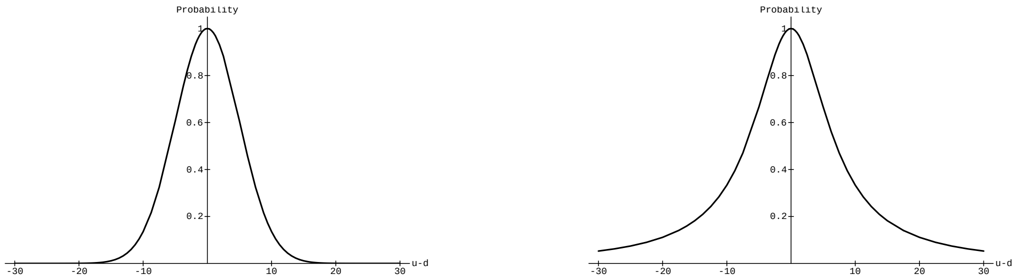


Figure 3.16: Gaussian and Cauchy distributions. The tails of the Gaussian, on the left, drop off quickly. The tails of the Cauchy distribution are more forgiving.

Both the Gaussian and Cauchy distributions are shown in Figure 3.16. Notice that the tails of the Cauchy distribution decrease more gradually than the Gaussian. This means that outliers are considered more likely and hence not penalized as heavily by the Lorentzian estimator when the do occur.[7]

Another important factor involves the scale parameters of some of the estimators. We will show in the following chapter how the scale parameter of the Lorentzian can be naturally set, and that it can be used to construct a sequence of estimators to be used in a continuation method.

[7] We do not know whether, and are not claiming that, the Cauchy distribution is the correct statistical model for the errors found in real scenes. It is merely presented here as a heuristic choice reflecting the kind of behavior we desire with respect to outliers.

Chapter 4

Robust Optical Flow: Experimental Results

This chapter demonstrates the benefits of the robust estimation framework. The framework from the previous chapter is applied to each of the approaches described in Chapter 2. This robust formulation of the regression, correlation, and explicit smoothness approaches improves the performance of the approaches, particularly when multiple motions are present. The main focus of the chapter will be the robust formulation of the explicit smoothness, or regularization, schemes, for these approachs provide a general purpose method for recovering dense optical flow fields. The method developed here is a simple reformulation of the standard first-order Horn and Schunck approach within the robust estimation framework. The resulting *robust gradient-based algorithm* is experimentally compared and contrasted with other optical flow techniques.

4.1 Regression Approaches

Since most statistical applications of robust statistics are to regression problems it is natural to formulate regression-based optical flow within the robust framework. Recall that the goal of the regression approach is to estimate the model parameters **a** that produce the best flow $\mathbf{u}(a)$ for a region $\mathcal{R}$. If the motion of the region is assumed to be constant, this reduces to finding the motion (u, v) that minimizes:

$$E(u, v) = \sum_{\mathcal{R}} \rho(I_x u + I_y v + I_t), \tag{4.1}$$

where the image derivatives, I_x, I_y, and I_t, are computed at each point in the region and a single value of (u, v) is estimated for the entire region.

The minimization problem is solved using Newton's method as described in the previous chapter. Below, the update equations for u are derived; exactly the same treatment can be applied to derive the update equations for v.

Taking the Taylor expansion of E gives:

$$\begin{aligned} E(u + \delta u, v) \;\; \approx \;\; & \sum_{\mathcal{R}} [\rho(I_x u + I_y v + I_t) + \psi(I_x u + I_y v + I_t) I_x \delta u + \\ & \frac{\partial}{\partial u} \psi(I_x u + I_y v + I_t) I_x^2 \delta u^2 + \ldots], \end{aligned} \tag{4.2}$$

where ψ is $\partial \rho(x)/\partial x$. To minimize, we take the derivative, drop the higher order terms, and set the equation to zero:

$$\frac{\partial E}{\partial u} = 0 = \sum_{\mathcal{R}} [\psi(I_x u + I_y v + I_T) I_x + \frac{\partial}{\partial u} \psi(I_x u + I_y v + I_t) I_x^2 \delta u]. \tag{4.3}$$

This gives us an update equation:

$$\delta u = -\frac{\sum_{\mathcal{R}} \psi(I_x u + I_y v + I_t) I_x}{\sum_{\mathcal{R}} \frac{\partial}{\partial u} \psi(I_x u + I_y v + I_t) I_x^2}, \tag{4.4}$$

and,

$$u_{n+1} = u_n + \delta u. \tag{4.5}$$

A slightly different minimization approach is adopted by Bergen *et al.* [1992]. They use a Gauss-Newton minimization procedure where, at the current iteration, i, there is an estimate, $\mathbf{u}_i$, and the change, $\delta \mathbf{u}$, in the estimate is obtained by minimizing:

$$E(\delta \mathbf{u}) = \sum_{\mathcal{R}} \rho(\nabla I \delta \mathbf{u} + \Delta I), \tag{4.6}$$

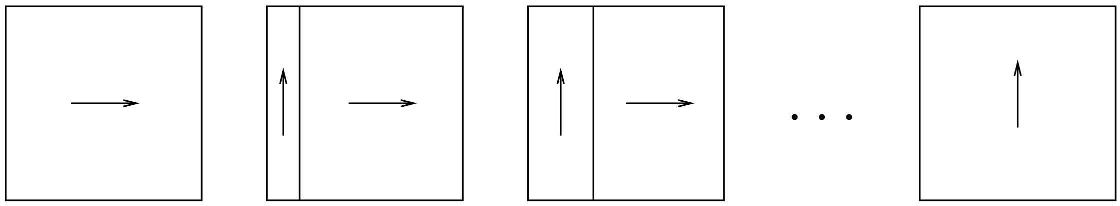


Figure 4.1: Constant Model Experiment.

where,

$$\Delta I(\mathbf{x}) = I(\mathbf{x}, t) - I(\mathbf{x} - \mathbf{u}_i, t - 1). \tag{4.7}$$

This has the effect of warping the previous image based on the current flow estimate. The temporal derivative is then computed with respect to this warped image and the estimate is refined by the new $\delta\mathbf{u}$. This Gauss-Newton approach may offer some advantages over the standard Newton's method in that the estimates of the temporal derivatives become more accurate as the images are registered by the warping process.

Bergen *et al.* [1992] take ρ to be the standard quadratic estimator and hence solve a least-squares regression problem. When, in addition to the dominant motion, there is another "distractor" motion within the region, the least squares estimate will be inaccurate. This can be contrasted with a robust formulation in which ρ is taken to be the Lorentzian estimator.

Consider Figure 4.1 which shows an experiment in which a simple constant flow model is used to estimate image velocity. A number of trials were performed, and in each case there were two random noise patterns present in the window; one moving to the right, the other moving up. Estimates of the horizontal motion component were computed as increasing amounts of the upward motion (the distractor) were added to the region. Figure 4.2 shows the results of the experiment. The dominant horizontal motion is shown as a solid line. The robust (Lorentzian) estimator does a good job of recovering the dominant horizontal motion and ignoring the distracting motion until approximately 40% of the region is occupied by the distractor. Not surprisingly, the least squares approach performs poorly, producing the

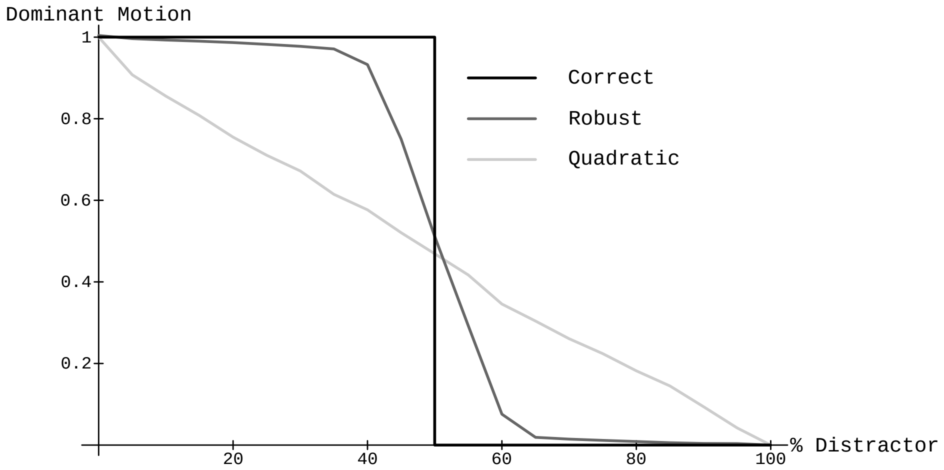


Figure 4.2: **Constant model experimental results.** Error for quadratic and robust formulations as a function of the amount of distractor.

mean horizontal motion rather than the dominant one.

4.2 Correlation-Based Approaches

The second set of optical flow techniques to be reformulated are the correlation-based approaches. Recall that the correlation approach is simply formulated as:

$$E_D(u, v) = \sum_{(x,y)\in\mathcal{R}} \rho(I(x, y, t) - I(x + u, y + v, t + 1)), \tag{4.8}$$

where $\rho(x) = x^2$ for the sum-of-squared-difference formulation. Also recall that computing the correlation over a range of displacements gives rise to a correlation surface (or SSD surface in the quadratic case) and that the minimum of this surface corresponds to the best motion estimate with respect to the data conservation assumption.

The standard quadratic error measure has the property that as data errors increase, the contribution of the error term increases without bound. As a result, when multiple motions are present within the neighborhood of a site, the correlation computed for one of the motions is corrupted by the data errors corresponding to the other motion [Black and Anandan, 1991b].

These problems can be reduced by adopting the robust estimation framework, in which the erroneous measurements are treated as outliers and rejected. Figure 4.3 compares the correlation surface generated using the quadratic estimator with one where ρ is a robust M-estimator; in this case:

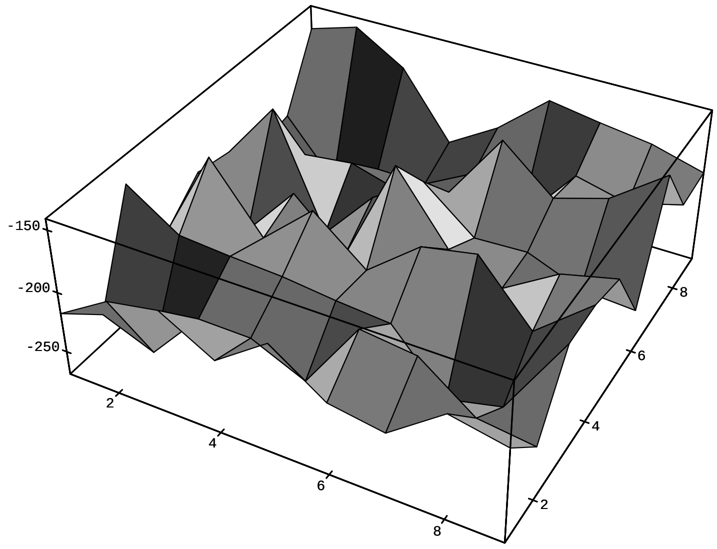
$$\rho(x, \sigma) = \frac{-1}{1 + (x/\sigma)^2}. \tag{4.9}$$

The surfaces are computed at the corner of a randomly textured square translating across a randomly textured background. The two peaks in Figure 4.3*b* correspond to the two motions present in a window centered at the corner. Notice that these peaks are not clearly defined when the quadratic estimator is used (Figure 4.3*a*).

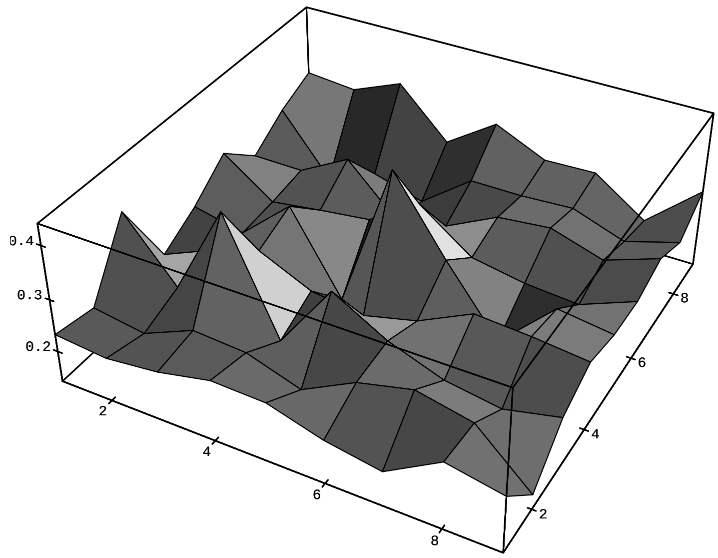
An important property of this robust correlation approach is that it does not suffer the same problems as the adaptive window technique of Okutomi and Kanade [1992]. Outliers can be scattered across the window with no spatial coherence as is the case with fragmented transparency. The approach, however, cannot deal with the more general case of multiple motions caused by reflection and translucency.

The robust correlation approach can be exploited in the early detection of motion discontinuities [Black and Anandan, 1990a]; that is, the estimation of motion boundaries before the computation of the optical flow field. In previous work, we formulated a number of constraints for detecting motion boundaries, one of which was the presence of multiple peaks in the correlation surface. The approach detected the two sharpest peaks in the correlation surface and then computed a heuristic measure indicating the confidence that multiple motions are present. If p_0 and p_1 are the values of E_D for the two best displacements then the confidence in the presense of a motion boundary can be heuristically defined as:

$$C_S = p_0/p_1.$$

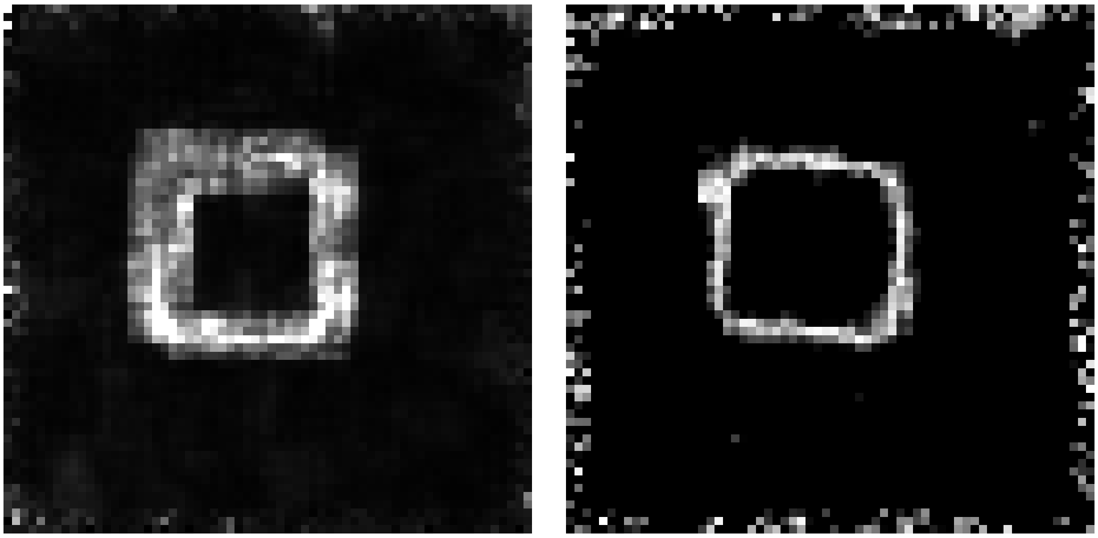


a



b

Figure 4.3: **SSD versus Robust Correlation.** Correlation surfaces computed at a translating corner (inverted for display); multiple motions are present within the correlation window. *a)* SSD surface. *b)* Robust correlation surface; noise is suppressed and peaks are more visible.



a *b*

Figure 4.4: Multiple motions in correlation surface (see text). *a)* Confidence C_S using SSD surface. *b)* Confidence using robust correlation (truncated quadratic).

This measure has a maximum of 1.0 at a motion boundary and falls off as distance from the boundary increases.

The effect of robust correlation on peak detection and boundary localization can be seen by considering the discontinuity confidence measure C_S displayed in Figure 4.4. The motion sequence consisted of a randomly textured square moving two pixels across a randomly textured background. Noise was added to the second image in the sequence, and the correlation was computed between the images. Figure 4.4*(a)* shows the results of peak detection when the SSD measure was used. When the quadratic was replaced by the truncated quadratic estimator, peak detection was noticeably improved (Figure 4.4*(b)*).

4.3 Explicit Smoothness Approaches

This section illustrates how robust estimation can be brought to bear on the explicit smoothness, or regularization, approaches to optical flow by considering the standard Horn and

Schunck formulation of the problem [Horn and Schunck, 1981]. This formulation represents a least-squares estimate of the flow field. Such estimates are commonly known to be sensitive to measurements that do not conform to the statistical assumptions of the model. A *robust gradient method* is formulated by recasting the least-squares formulation of optical flow in the robust estimation framework. The approach achieves the three goals stated earlier; it prevents smoothing across motion boundaries, it permits the recovery of motion boundaries, and it provides a general framework for treating, and detecting, violations of model assumptions.

We have shown how the data conservation and spatial coherence constraints can be made robust in the presence of erroneous image measurements and motion discontinuities. The least-squares form of the optical flow equation is reformulated as:[1]

$$\begin{aligned} E(u,v) &= \sum_{S} \lambda E_D(u,v) + E_S(u,v), \\ &= \sum_{s \in S} [\lambda \rho(I_x u_s + I_y v_s + I_t, \sigma_1) \\ &\qquad + \sum_{n \in \mathcal{G}_s} \rho(u_s - u_n, \sigma_2) + \sum_{n \in \mathcal{G}_s} \rho(v_s - v_n, \sigma_2)]. \end{aligned} \tag{4.10}$$

When ρ is the quadratic error measure, this is the least-squares optical flow equation. For the robust formulation, we simply replace the quadratic error measure by the more robust Lorentzian M-estimator. The implications of this reformulation are explored in the remainder of this section.

4.3.1 Discontinuities and Parameter Estimation

We would like to be able to set thresholds τ_1 and τ_2 that determine what data and smoothness errors are considered outliers. To do so we need to determine the appropriate values for σ_1 and σ_2. These values determine the point at which measurements are considered outliers,

[1] The smoothness term here is slightly different than the version proposed in the previous chapter where we minimized $\rho(||\mathbf{u}_s - \mathbf{u}_n||, \sigma)$. We treat u and v seperately in the current formulation to make clear the parallels to the Horn and Schunck approach.

which may be taken to be the point at which the influence of the measurements begins to decrease; that is where the derivative of the ψ-function:

$$\frac{\partial^2 \rho}{\partial x^2} = \frac{\partial \psi}{\partial x} = \frac{2(2\sigma^2 - x^2)}{(2\sigma^2 + x^2)^2}, \tag{4.11}$$

equals zero. This occurs when:

$$x = \pm\sqrt{2}\,\sigma. \tag{4.12}$$

So to define an outlier threshold τ, we set $\sigma = \tau/\sqrt{2}$.

For example, in the case of the smoothness constraint, if a difference of 0.05 pixels is considered a discontinuity then $\sigma_2 = 0.0353553$. This threshold could presumably be set on the basis of psychophysical evidence.[2] Motion discontinuities can be recovered from the computed flow field by examining where this threshold is exceeded:

$$l_{s,t} = \begin{cases} 1 & |u_s - u_t| \geq \tau_s \text{ or } |v_s - v_t| \geq \tau_s \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

For the data term we make a conservative estimate of the variance in the intensity error for the optimal flow field. We do this by computing the intensity error $I_x u + I_y v + I_t$ in the case where the flow is zero everywhere; that is, the error is simply I_t. We then compute the variance of this initial error and take that as the value of σ_1.

4.3.2 Convexity

The least-squares formulation of optic flow is relatively straightforward to solve since the objective function is convex. The robust formulation, however, may not be convex, because

[2]The spatial discontinuity threshold is currently set based on simple estimates of what constitutes a "noticeable" discontinuity. It is the term "noticeable" that needs to be studied more closely. For example what effect does contrast across the boundary, relative motion of the surfaces, and spatial frequency of the patterns on either side of the boundary have on our ability to perceive motion discontinuities. There has been some work in this area by [Baker and Braddick, 1982; van Doorn and Koenderink, 1983; Hildreth, 1984; Vaina and Grzywacz, 1992].

Since the correct threshold is not known, a threshold is chosen based on experience and is held constant throughout the experiments. This is important for judging the results of the approach. The discontinuity thresholds have not been adjusted to produce "nice looking" results on each of the chosen image sequences.

if the data and smoothness terms disagree, we can minimize for either one and treat the other as an outlier.

Formally, the objective function is convex when the *Hessian matrix*:

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial u^2} & \frac{\partial^2 E}{\partial u \partial v} \\ \\ \frac{\partial^2 E}{\partial v \partial u} & \frac{\partial^2 E}{\partial v^2} \end{bmatrix} \tag{4.14}$$

is positive definite [Rockafellar, 1970]. This condition is met if and only if both eigenvalues of the matrix H are positive. This gives us a simple test for convexity. It is easy to show that E is locally convex when:

$$\begin{aligned} \max_{s \in S} |(I_x u_s + I_y v_s + I_t)| &\leq \sqrt{2}\,\sigma_1 = \tau_1, \text{ and} && (4.15) \\ \max_{s \in S} \max_{n \in \mathcal{G}_s} |u_s - u_n| &\leq \sqrt{2}\,\sigma_2 = \tau_2, \text{ and} && (4.16) \\ \max_{s \in S} \max_{n \in \mathcal{G}_s} |v_s - v_n| &\leq \sqrt{2}\,\sigma_2 = \tau_2. && (4.17) \end{aligned}$$

These conditions correspond to the case where there are no data or spatial outliers. In this range, the ψ-function is roughly linear and the error function ρ is roughly quadratic.

Given that the objective function may be non-convex, there are a number of minimization techniques which can be brought to bear on the problem. First we will describe an optimization technique that rapidly converges to a local minimum. We then consider a global optimization strategy.

4.3.3 Simultaneous Over-Relaxation

Simultaneous Over-Relaxation (SOR) belongs to a family of relaxation techniques that include *Jacobi's* method and the *Gauss-Seidel* method [Press *et al.*, 1988; Strang, 1976; Varga, 1962]. We compute the first partial derivatives of the robust flow equation (4.10):

$$\begin{aligned} \frac{\partial E}{\partial u_s} &= \sum_{s \in S} [\lambda I_x \psi(I_x u_s + I_y v_s + I_t, \sigma_1) + \sum_{n \in \mathcal{G}_s} \psi(u_s - u_n, \sigma_2)], && (4.18) \\ \frac{\partial E}{\partial v_s} &= \sum_{s \in S} [\lambda I_y \psi(I_x u_s + I_y v_s + I_t, \sigma_1) + \sum_{n \in \mathcal{G}_s} \psi(v_s - v_n, \sigma_2)]. && (4.19) \end{aligned}$$

Then the iterative update equations for minimizing E at step $n+1$ are simply [Blake and Zisserman, 1987]:

$$u_s^{(n+1)} = u_s^{(n)} - \omega \frac{1}{T(u_s)} \frac{\partial E}{\partial u_s}, \tag{4.20}$$

$$v_s^{(n+1)} = v_s^{(n)} - \omega \frac{1}{T(v_s)} \frac{\partial E}{\partial v_s}, \tag{4.21}$$

where ω is an *overrelaxation parameter* that is used to *overcorrect* the estimate of $u^{(n+1)}$ at stage $n+1$.[3]

The terms $T(u_s)$ and $T(v_s)$ are upper bounds on the second partial derivatives of E:

$$T(u) \geq \frac{\partial^2 E}{\partial u_s^2}, \quad \forall s \in S, \tag{4.22}$$

$$T(v) \geq \frac{\partial^2 E}{\partial v_s^2}, \quad \forall s \in S. \tag{4.23}$$

The second derivative is maximized when both the data and smoothness errors are zero everywhere, which implies:

$$T(u) = \frac{\lambda I_x^2}{\sigma_1^2} + \frac{4}{\sigma_2^2}, \tag{4.24}$$

$$T(v) = \frac{\lambda I_y^2}{\sigma_1^2} + \frac{4}{\sigma_2^2}. \tag{4.25}$$

When $0 < \omega < 2$ the method can be shown to converge [Varga, 1962] but the rate of convergence is sensitive to the exact value of ω. While determining the optimal ω is difficult in the case of a non-linear problem, we can get a rough approximation by computing the optimal value for the linear Jacobi version of the problem. The optimal ω is then related to the largest eigenvalue ($\mu_{\max}$) of the Jacobi iteration matrix which can be shown to be:

$$\mu_{\max} = \cos \pi h, \tag{4.26}$$

$$h = \frac{1}{(n+1)}, \tag{4.27}$$

[3] This is simply Newton's method as formulated earlier when $\omega = 1$ and $T(u) = \partial^2 E / \partial u$. We adopt this slightly different formulation for faster convergence and consistency with Blake and Zisserman's notation. SOR can also stand for Successive Over-Relaxation, but here we prefer "Simultaneous" as the updating will be performed in parallel.

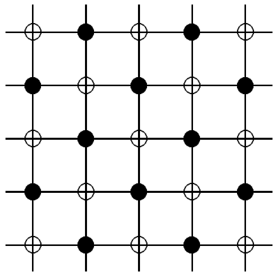


Figure 4.5: Checkerboard pattern for first-order smoothness exploits parallelism.

for an $n \times n$ problem [Strang, 1976]. The approximation to the optimal overcorrection is then:

$$\omega_{opt} = \frac{2(1 - \sqrt{1 - \mu^2_{\max}})}{\mu^2_{\max}}. \tag{4.28}$$

For example, a 128×128 image would have an overcorrection of $\omega_{opt} = 1.95209$, and a 64×64 image would have $\omega_{opt} = 1.90645$. In practice, this approximation works well and for an $n \times n$ problem acceptable convergence is reached within only n iterations.

The algorithm can be implemented sequentially, but is inherently parallel. Notice that, for first order constraints, each site is dependent on its nearest neighbors. While updating a site s, the estimates of its neighbors $t \in \mathcal{G}_s$ must be held fixed. By partitioning the sites using a checkerboard pattern, half the sites can be updated at once while the other half remains unchanged [Geman and Geman, 1984; Murray *et al.*, 1986]. In Figure 4.5 all the sites can be updated in two iterations by first updating the black sites and then updating the white sites in parallel. This parallelism can easily be exploited on a SIMD architecture like the Connection Machine [Hillis, 1985] with a physical processor for each site.

For higher order constraints more complex partitionings are required which increases the number of iterations required to update the entire field. But, since we expect the neighborhood size to be small with respect to the image, there is still a tremendous speedup gained by this data-level parallelism.

Faster convergence can be achieved using *Chebyshev acceleration* [Press *et al.*, 1988]. Here the value of ω is updated after each half-iteration (ie. after updating all the white, or all the black, sites) using the following scheme:

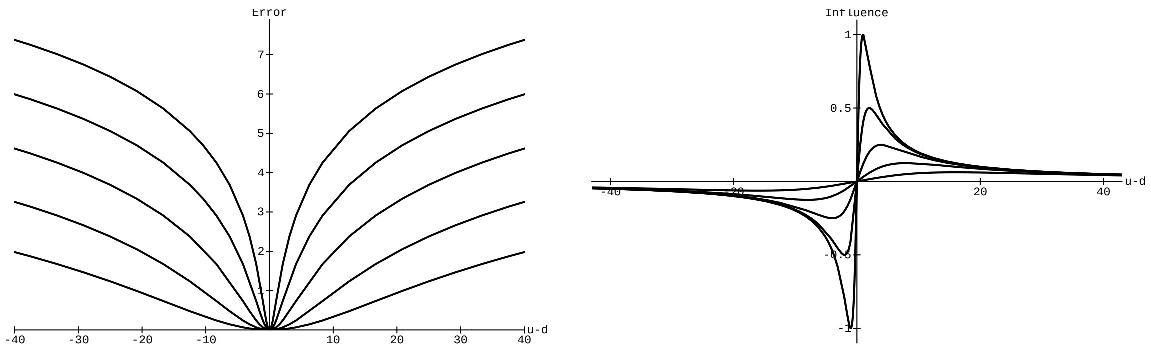
$$\begin{aligned} \omega^{(0)} &= 1, \\ \omega^{(1/2)} &= 1/(1-\mu_{\max}^2/2), \\ \omega^{(n+1/2)} &= 1/(1-\mu_{\max}^2\omega^{(n)}/4), \quad n = 1/2,\ 1, \ldots, \infty, \\ \omega^{(\infty)} &\rightarrow \omega_{opt}. \end{aligned}$$

4.3.4 Graduated Non-Convexity

We now turn to the problem of finding a globally optimal solution when the function is non-convex. Stochastic approaches like simulated annealing [Geman and Geman, 1984; Kirkpatrick *et al.*, 1983] have been used by a number of authors for recovering optical flow with non-convex objective functions [Black and Anandan, 1991b; Konrad and Dubois, 1988; Murray and Buxton, 1987]. We will explore this approach in detail later in the thesis but, for now, we can exploit the nature of the objective function and the choice of robust estimator to use a deterministic continuation method.

Continuation methods [Rangarajan and Chellapa] involve constructing a sequence of approximations to the objective function by varying a control parameter. The initial approximation is constructed to be convex and, hence, is readily minimized using, for example, the SOR technique above. This minimum is then tracked as the control parameter is varied to produce successively better approximations of the true objective function. For a given objective function the challenge is to construct the sequence of approximations.

Specifically, we will consider the *Graduated Non-Convexity (GNC)*, algorithm which has been studied in detail by Blake and Zisserman [1987]. While Blake and Zisserman construct approximations to the truncated quadratic, we find that, by taking the Lorentzian as the robust estimator, there is a natural sequence of approximations. In the previous section,



a *b*

Figure 4.6: **Graduated Non-Convexity.** $\rho(x, \sigma)$ and $\psi(x, \sigma)$ plotted for thresholds $\tau \in \{16, 8, 4, 2, 1\}$. *a)* Error measure $\rho(x, \sigma)$. *b)* ψ-function $\psi(x, \sigma)$.

it was noted that E is convex if the outlier thresholds τ_1 and τ_2 are set to be greater than maximum data and smoothness errors. Assume that the motion in the scene is constrained to be less than some constant. This can be achieved by using a coarse-to-fine approach (see Chapter 2) in which, by refining the motion across scales, we ensure that the motion at any level of the pyramid is less than a pixel. Then we choose τ_2 to be twice the largest allowable motion. The maximum data error can be conservatively estimated from the images. First assume that the flow is zero everywhere, so $I_x u + I_y v + I_t = I_t$. Now we take as our estimate $\tau_1 = \max |I_t|$.

The minimization can begin with this convex approximation and the resulting coarse flow field approximation will contain no flow discontinuities. In this sense it will be very much like the least-squares flow estimate. Discontinuities can be gradually added by lowering the thresholds τ_1 and τ_2 and repeating the minimization. Figure 4.6 shows the error function (Figure 4.6*a*) and the ψ-function (Figure 4.6*b*) for various values of τ. In practice, we have found that a two-stage minimization works well. First the coarse convex approximation is used, followed by the original objective function.

To cope with motions larger than a single pixel we use the simple coarse-to-fine gradient-based strategy described in Chapter 2.

4.3.5 Experimental Results

This section presents a number of experiments using synthetic data that illustrate the behavior of the robust-gradient formulation in the presence of noise and motion discontinuities. In particular, we are interested in the effect of the robust data term on the final solution. We also show the performance of the algorithm on two real image sequences and compare the results with other approaches.

The robust-gradient technique is implemented using the GNC algorithm of the previous section, and the current Connection Machine [Hillis, 1985] implementation fully exploits the parallelism inherent in the formulation. There is a physical processor at each site in the image and only simple North-East-West-South (NEWS) communication is required between processors.

All experiments were performed using 200 iterations[4] of each algorithm even though 200 iterations are not typically necessary in the case of SOR. The only parameters that need to be empirically determined are τ_2 and λ. These were chosen to produce the best result for each algorithm, and remained unchanged for all the experiments: $\tau_2 = 0.05$, $\lambda = 10$ for the robust-gradient approach, and $\lambda = 50$ for the least-squares approach[5]. All other parameters were determined as specified in the previous sections. The spatial and temporal derivatives (I_x, I_y, I_t) were estimated using the simple technique described by Horn [1986].

Synthetic Sequence

The first experiment involves a synthetic sequence containing two textured surfaces, one which is stationary and one which is translating to the left (Figure 4.7a). The horizontal and vertical components of the flow are shown with the magnitude of the flow represented

[4] An iteration is taken to mean the updating of every site in the flow field.

[5] The different values of λ are due to the different ρ functions used; that is, the quadratic for the least-squares approach, and the Lorentzian for the robust-gradient method.


a b c

Figure 4.7: **Random Noise Example.** *a)* First random noise image in the sequence. *b)* True horizontal motion (black $= -1$ pixel, white $= 1$ pixel, gray $= 0$ pixels). *c)* True vertical motion.

by intensity, where black indicates motion to the left and up and, similarly, white indicates motion to the right and down. The true horizontal and vertical motions are shown in Figures 4.7*b* and 4.7*c* respectively.

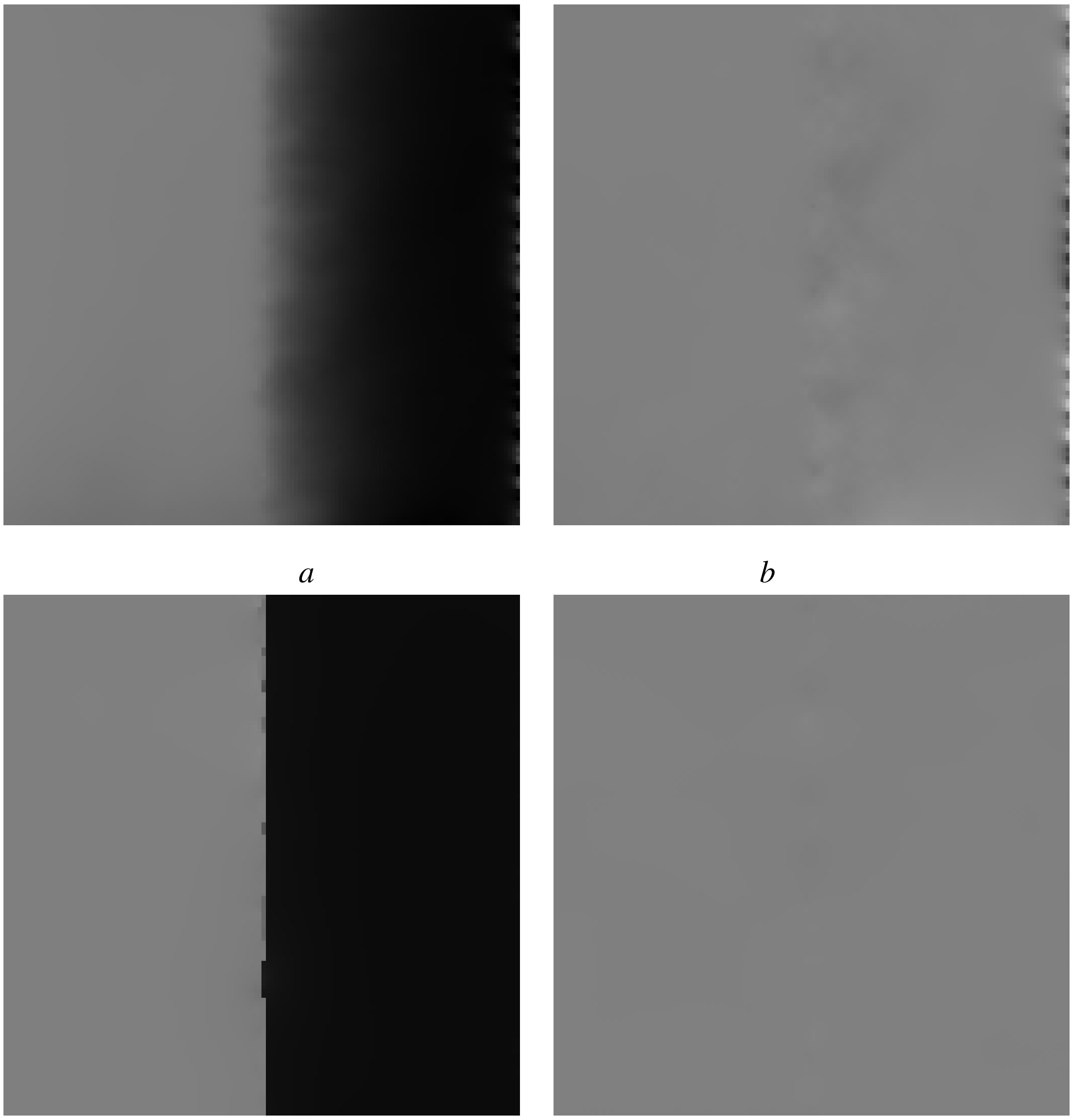
Figures 4.8*a* and 4.8*b* show the flow computed with the least squares formulation. Notice how the horizontal flow is smoothed across the motion boundary.[6] The robust-gradient technique does not suffer from this over-smoothing (Figures 4.8*c* and 4.8*d*).

	Percentage of flow vectors with error:	
Approach	$\leq 1\%$	$\leq 5\%$
Least Squares.	68%	83%
Robust Gradient.	79%	98%

Table 4.1: Error statistics for the noiseless case.

The accuracy of the flow vectors is shown in Table 4.1. For problems such as structure

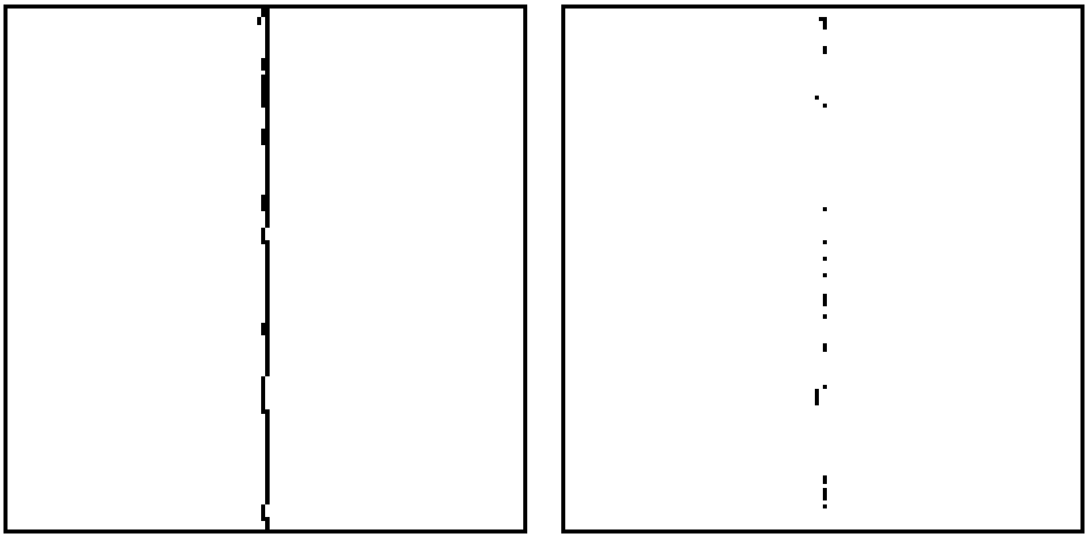
[6]Notice that the flow estimates are better in the left half of the image. This portion of the image did not move while the right portion was displaced by one pixel. The poorer motion estimates on the right are a result of the formulation of the gradient constraint which assumes small motion; the larger the motion the less accurate the estimates will be.



a *b*

c *d*

Figure 4.8: **Random Noise Sequence Results.** *a, b)* Least-squares solution; horizontal and vertical components of the flow. *c, d)* Robust-gradient solution; horizontal and vertical components of the flow.



a *b*

Figure 4.9: **Random Noise Sequence Outliers.** *a)* Motion discontinuities where the smoothness constraint is violated. *b)* Data outliers.

from motion, the accuracy of the flow estimates is crucial; errors of greater than 5% may render the results useless for this purpose. In the noiseless case, the robust-gradient approach finds 98% of the flow vectors to within 5% accuracy. This level of performance is achieved by recovering the flow more accurately within the vicinity of the motion boundary.

We can detect outliers where the final values of the data coherence and spatial smoothness terms are greater than the outlier thresholds τ_1 and τ_2. Motion discontinuities are simply outliers with respect to spatial smoothness. These are shown in Figure 4.9*a*.

There are outliers for the data term as well which, in this noiseless example, occur only at the motion boundary (Figure 4.9*b*). At this motion discontinuity the derivative estimates I_x, I_y, and I_t will be inaccurate as they pool information over a small neighborhood that spans the boundary. Due to occlusion occurring at the discontinuity, the error in the intensity constraint equation may be high. The robust data term allows these measurements to be treated as outliers.

This example illustrates how, even when no noise is present, the least-squares approach

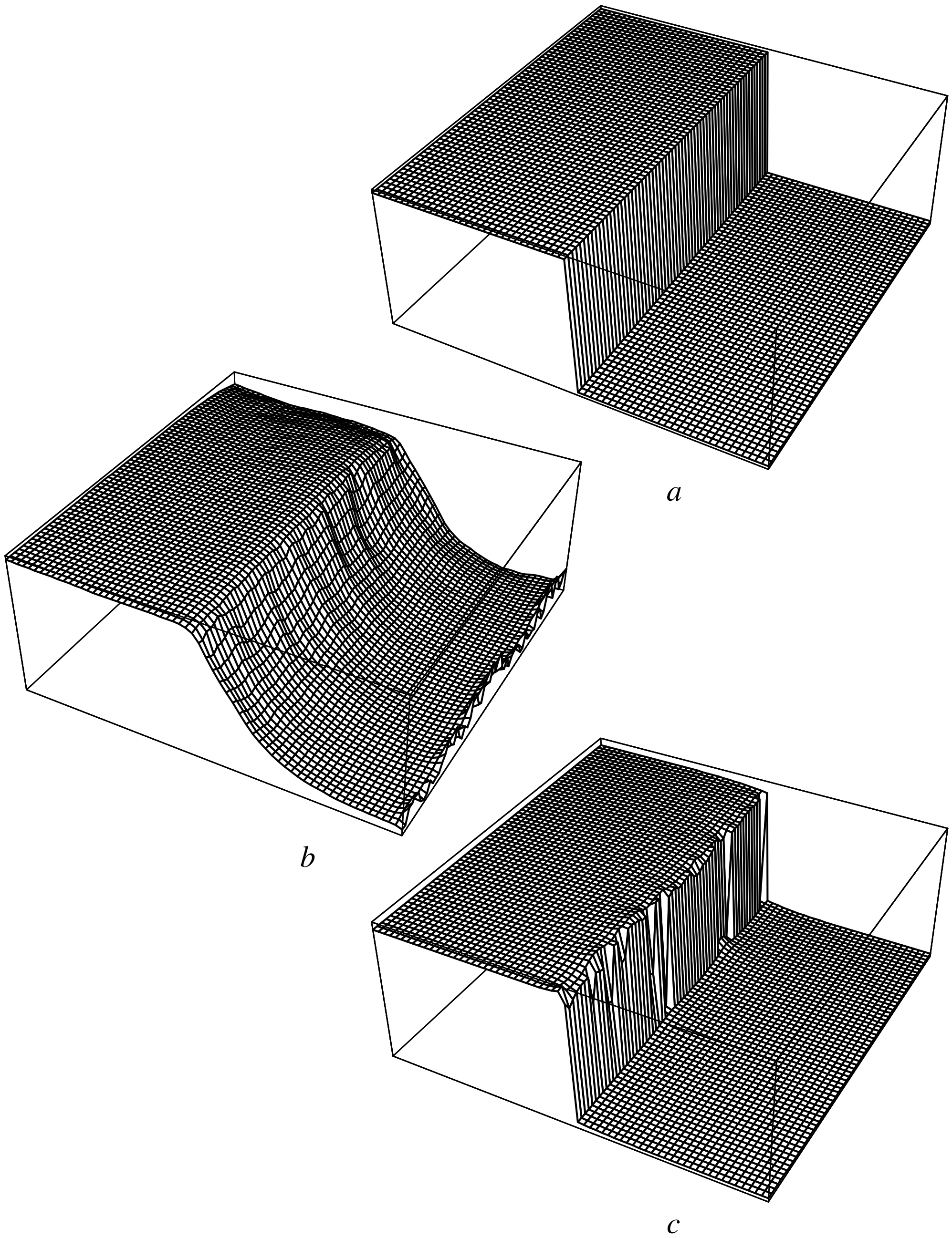


Figure 4.10: **Horizontal Displacement.** The horizontal component of motion is interprated as height and plotted. Figure *a* shows the plot for the true motion. Plotting the results illustrates the over-smoothing of the least-squares solution (*b*), and the sharp discontinuity which is preserved by the robust-gradient technique (*c*).

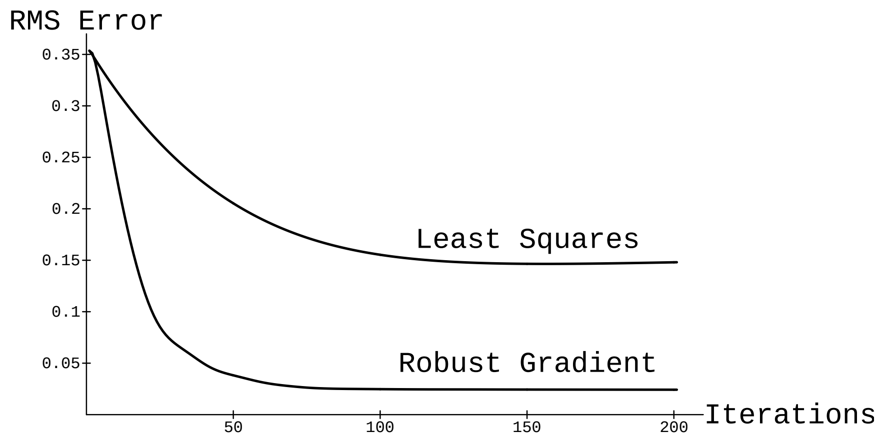
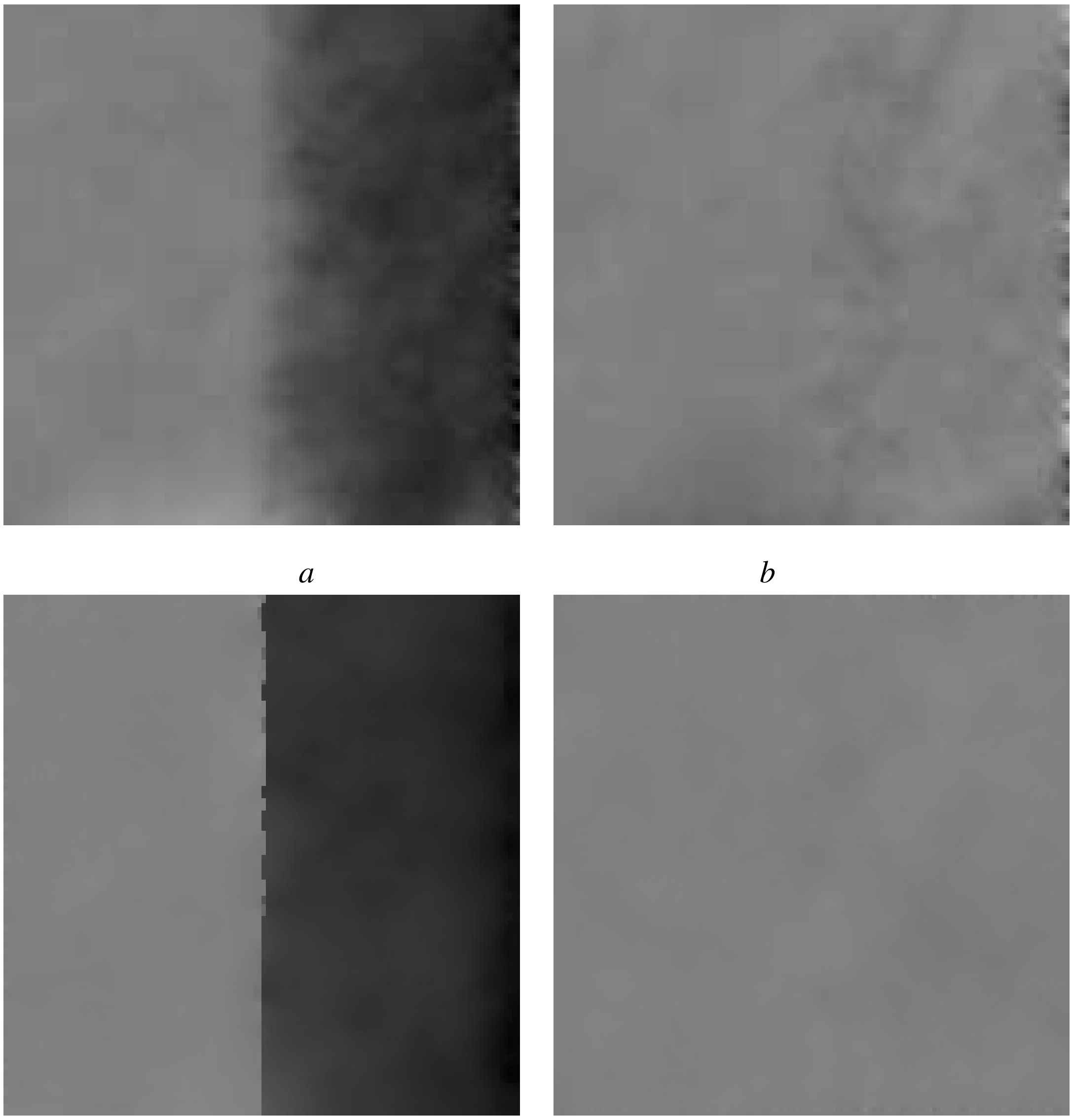


Figure 4.11: **Convergence.** Root mean squared error is plotted as a function of iterations for the standard Horn and Schunck scheme and the robust-gradient scheme using simultaneous over-relaxation on the random dot sequence shown in Figure 4.8. Notice that for an image of size 128×128 that SOR, with Chebyshev acceleration, converges in less than 128 iterations.

can perform badly by smoothing across discontinuities. The contrast between the approaches is vividly observable in Figure 4.10. Plotting the horizontal component of the flow field graphically shows the behavior of the two algorithms at motion boundaries. Figure 4.10*a* is the true horizontal motion while Figures 4.10*b* and 4.10*c* show the recovered motion using the least-squares and the robust techniques respectively.

Figure 4.11 shows the convergence behavior of the two algorithms. The faster convergence rate of the robust-gradient algorithm is due to the use of over-relaxation. This was not used in the least-squares case. The least-squares approach, however, does not approach the error of the robust-gradient technique. This is a result of errors due to over-smoothing at the motion boundary.

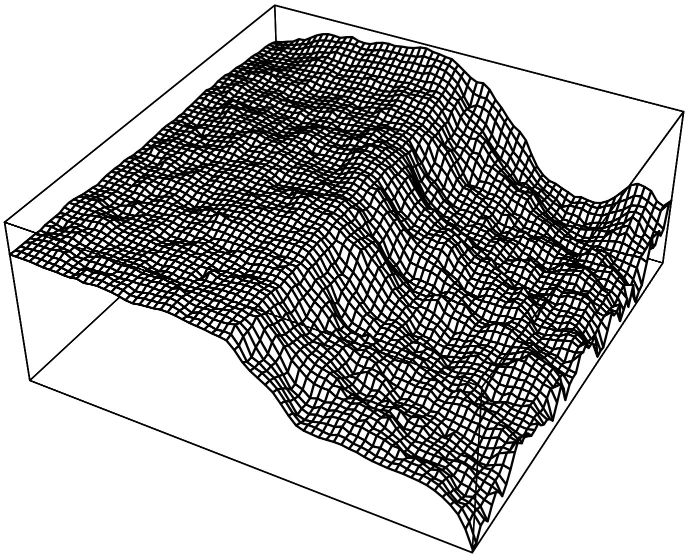
The effects of noise are explored in Figures 4.12 and 4.13. The figures show the effects of adding 5 percent, zero mean, uniform random noise to the second image in the sequence. The discontinuity is still preserved by the robust approach (Figure 4.12*c*). With the standard smoothness constraint there is a tradeoff between smoothing the noise and over-smoothing



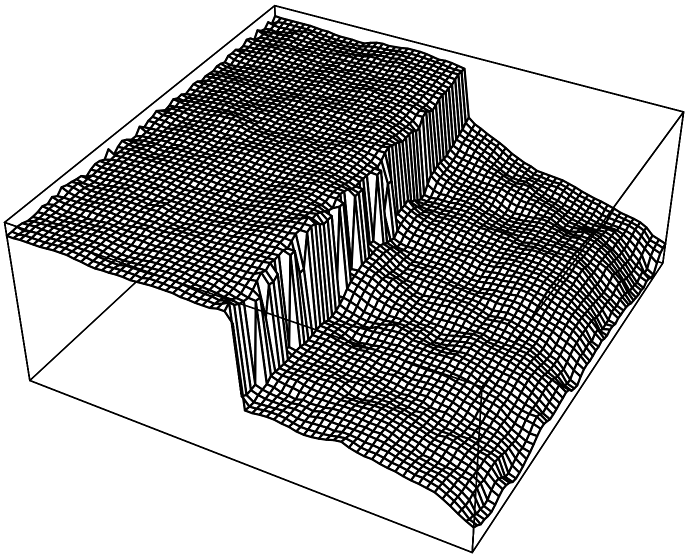
a *b*

c *d*

Figure 4.12: **Random Noise Sequence.** Computed flow in the case where 5 percent uniform noise is added to the second image. *a, b)* Horizontal and vertical least-squares flow. *c, d)* Horizontal and vertical robust flow.



a



b

Figure 4.13: **Horizontal Displacement (Noise added).** The horizontal component of motion is plotted for the case where 5 percent, zero mean, uniform random noise is added to the second image. *a)* The least-squares solution; smoothing to reduce noise over-smoothes the motion boundary. *b)* The robust-gradient approach smoothes the data while preserving the discontinuity.

the motion boundaries. The robust-gradient approach allows us to smooth the data and preserve the discontinuities (Figure 4.13*b*). The accuracy of the approaches is summarized in

	Percentage of flow vectors with error:	
Approach	$\leq 1\%$	$\leq 5\%$
Least Squares.	9%	38%
Robust Gradient.	30%	50%

Table 4.2: Error statistics for the 5% noise case.

Table 4.2. While both approaches achieve significantly less accuracy, the robust gradient approach still recovers half of the flow vectors to within 5% accuracy.

The Data Term

A significant difference between the robust approach described here and previous approaches to computing optical flow with discontinuities is that here the data component of the objective function is made robust. A more traditional formulation of the problem would include a line process or weak continuity constraint for the smoothness term [Blake and Zisserman, 1987; Geman and Geman, 1984; Harris *et al.*, 1990] and leave a quadratic data term. What advantage does the robust data term offer?

Consider the same random dot sequence as above but with the addition of 10% uniform noise. We compare the performance of three common approaches: a purely quadratic formulation (Horn and Schunck), a version with a quadratic data term and robust smoothness term (Blake and Zisserman), and the fully robust formulation described here.

The accuracy of the three approaches is summarized in Table 4.3. The results indicate that adding just a robust smoothness term increases the percentage of accurate flow vectors. Using both robust data and smoothness terms increases this percentage even more.

This is not the entire story. While the robust smoothness term by itself increases the percentage of accurate flow vectors, the mean error in the flow field increases. Table 4.4

	Percentage of flow vectors with error:	
Approach	$\leq 1\%$	$\leq 5\%$
Least Squares.	2%	20%
Robust Smoothness.	8%	30%
Robust Gradient.	16%	47%

Table 4.3: Error statistics for the 10% noise case.

Approach	*RMS Flow Error*	*RMS Intensity Error*
Both terms quadratic.	0.1814	2.600
Quadratic data, robust smoothness.	0.2208	1.889
Both terms robust.	0.0986	2.653

Table 4.4: **Behavior of data term.** The table shows the effects of the robust data and smoothness terms. The root mean squared errors in the flow estimate and the data term $(I_x u + I_y v + I_t)$ are shown for three common approaches.

further explores the effect of the robust terms. The purely quadratic solution attempts to be faithful to both the smoothness model and the noisy data; the result is moderately high errors in both the flow estimate and the intensity constraint.

Adding a robust smoothness term (for example by employing a line process) results in lower errors in the intensity constraint equation but with higher error in the flow estimate. With such a formulation, gross errors in the intensity data pull the solution away from the true flow while the robust term compounds matters by allowing discontinuities to be introduced. The result is that the accurate flow estimates are more accurate (as shown in Table 4.3) but that the the inaccurate flow estimates can be worse (as shown by the increase in the mean flow error in Table 4.4.

The fully robust version appears to provide the best balance. The robust data term allows the intensity constraint to be violated. Consequently, this version has the highest intensity error and the lowest error in the recovered flow field.

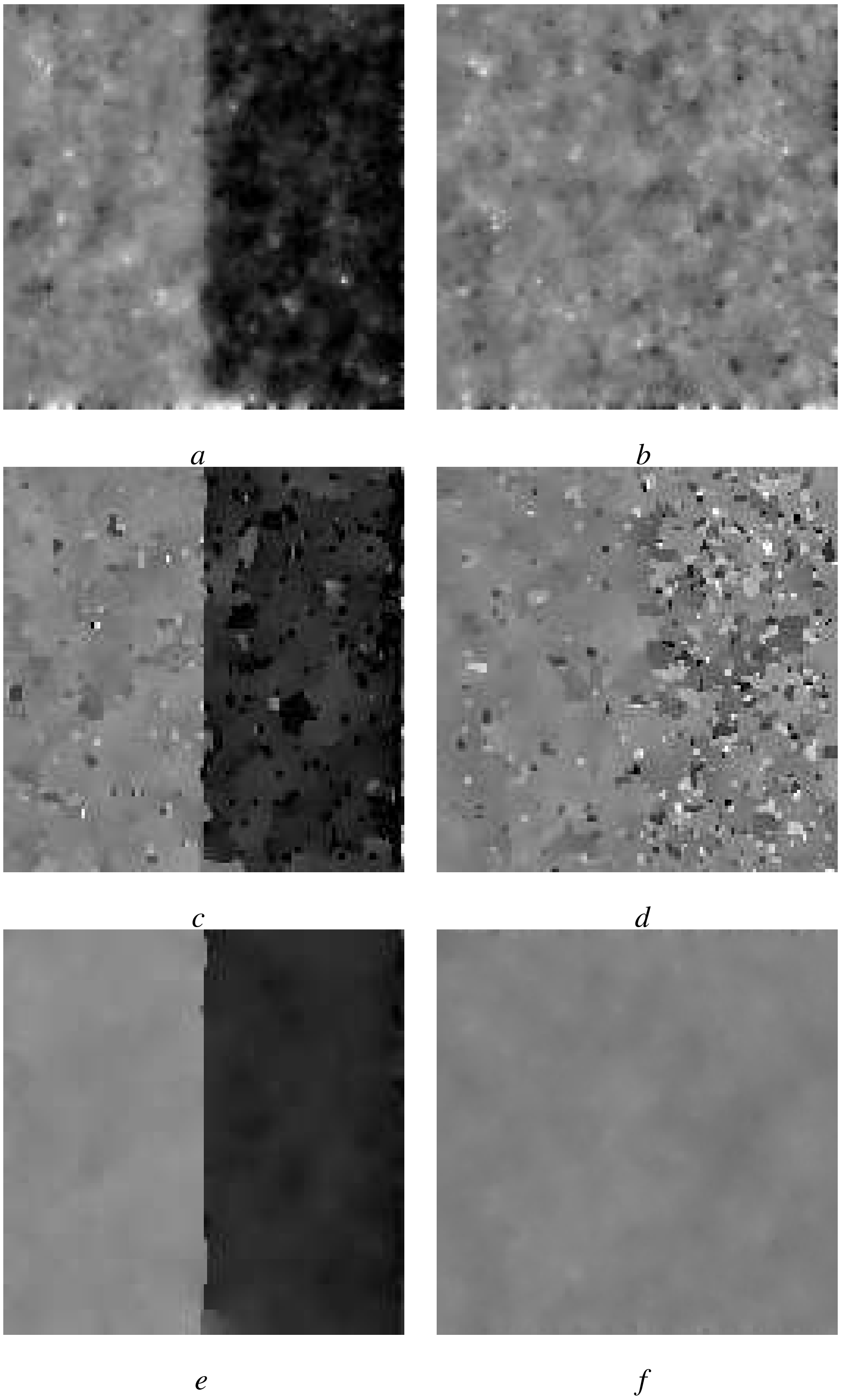
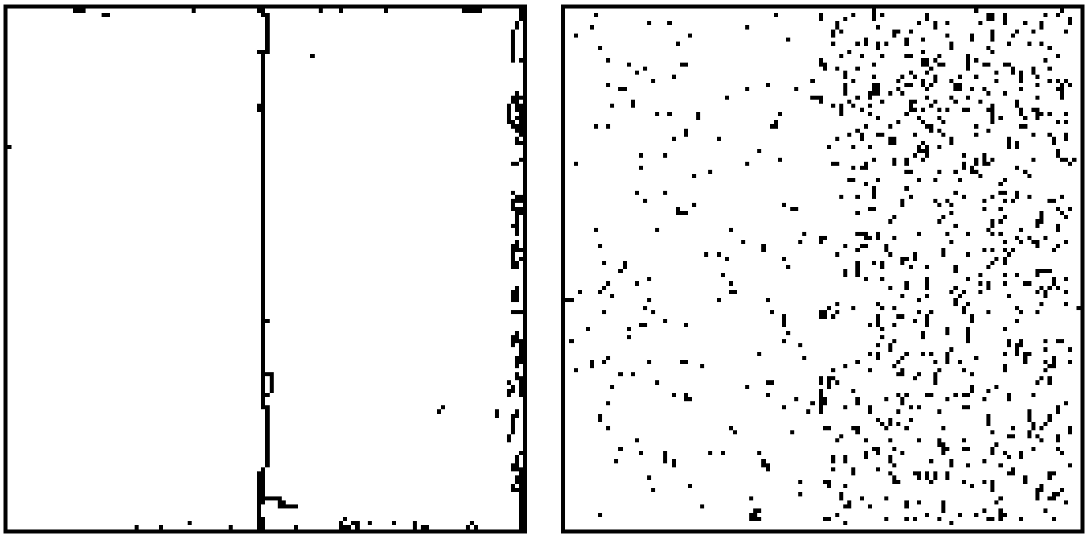


Figure 4.14: **Effect of robust data term,** (10% uniform noise). *a,b)* Least-squares (quadratic) solution. *c,d)* Quadratic data term and robust smoothness term. *e,f)* Fully robust formulation.



a *b*

Figure 4.15: **Outliers in the smoothness and data terms,** (10% uniform noise). *a)* Flow discontinuities. *b)* Data outliers.

These results are illustrated in Figure 4.14. Figures 4.14*a* and 4.14*b* show the noisy, but smooth, results obtained by least-squares. Figures 4.14*c* and 4.14*d* show the result of introducing a robust smoothness term alone. The recovered flow is piecewise smooth, but the gross errors in the data produce spurious motion discontinuities. Finally Figures 4.14*e* and 4.14*f* show the improvement realized when both the data and spatial terms are robust. Figure 4.15 shows where the spatial smoothness and data coherence terms are violated. Notice that a large number of data points are treated as outliers by the data term; especially when the motion is large.

The Pepsi Sequence

We next consider a real image sequence containing a Pepsi can in front of a textured background (Figure 4.16). The camera is translating to the right, resulting in the can being displaced approximately one pixel to the left in each frame and the background being displaced by approximately a third of a pixel between frames. Figures 4.17*a* and 4.17*b* show

Figure 4.16: **Pepsi Sequence.** First and last images in the 10 image sequence.

the results of applying the least-squares algorithm and illustrate how the flow is smoothed across the motion boundary.

The flow was also computed using Anandan's coarse-to-fine SSD algorithm [Anandan, 1989], and the results are shown in Figures 4.17*c* and 4.17*d*. Errors in the data term cannot be overridden by the smoothness term and consequently the results are noisy.

The results of the robust-gradient approach are shown in Figure 4.18. Figures 4.18*a* and 4.18*b* show the results achieved using the convex approximation in the first stage of the GNC algorithm. The results are similar to the smooth solution obtained with the least-squares approach. Figures 4.18*c* and 4.18*d* show the result of introducing spatial and data robustness. From the figure it is clear that the approach does an excellent job of preserving sharp motion discontinuities.

Figure 4.19*b* shows the locations where the smoothness constraint is violated (ie. the motion discontinuity is greater than τ_2); the boundaries correspond well to the physical boundaries of the can.

Finally, the least-squares and robust-gradient solutions can be compared by examining

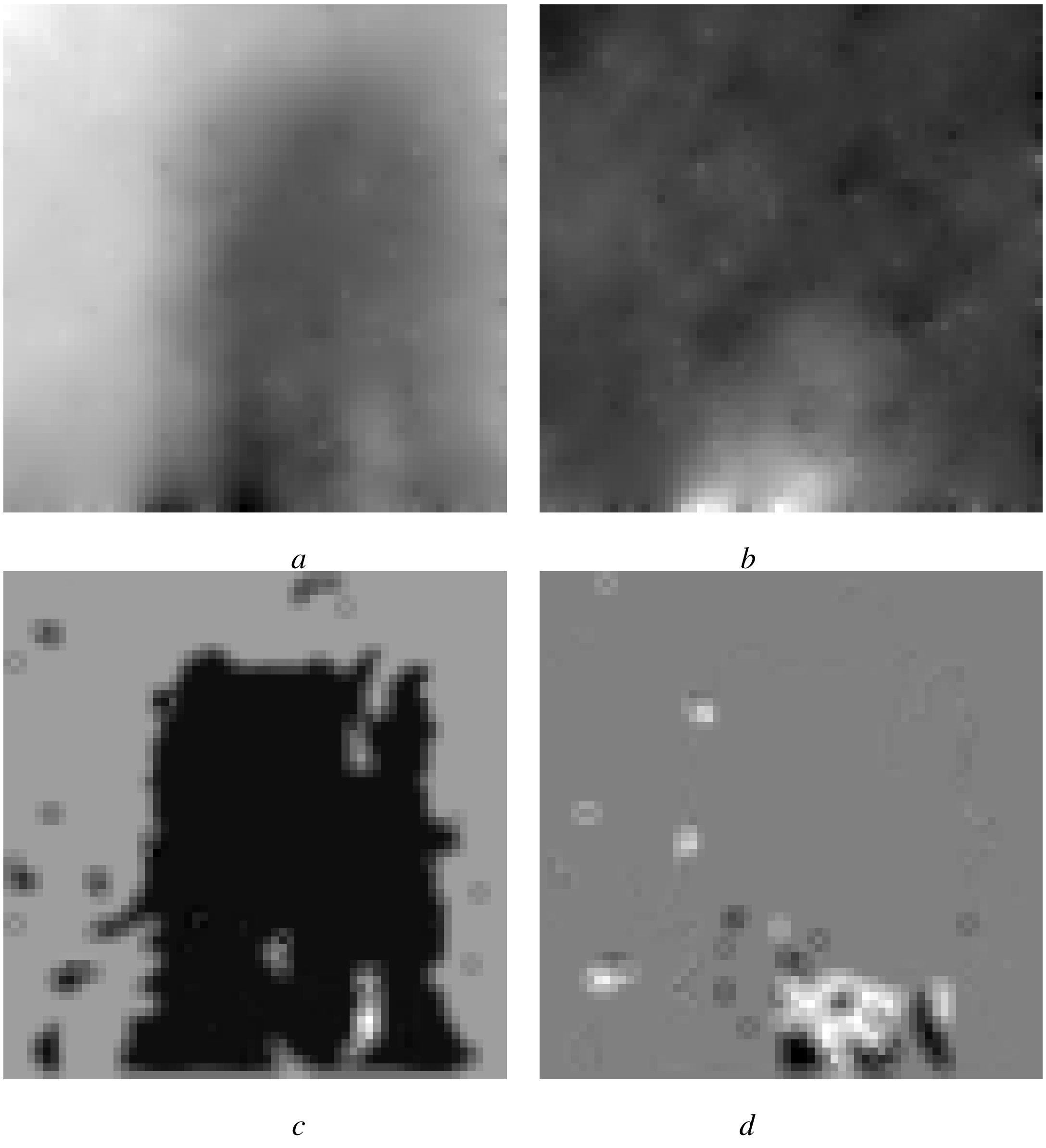
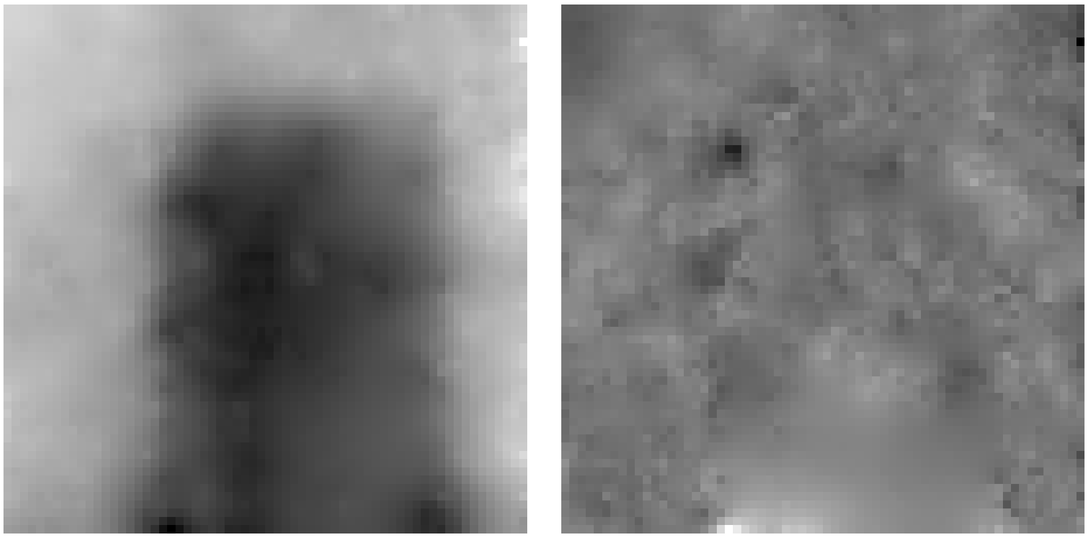
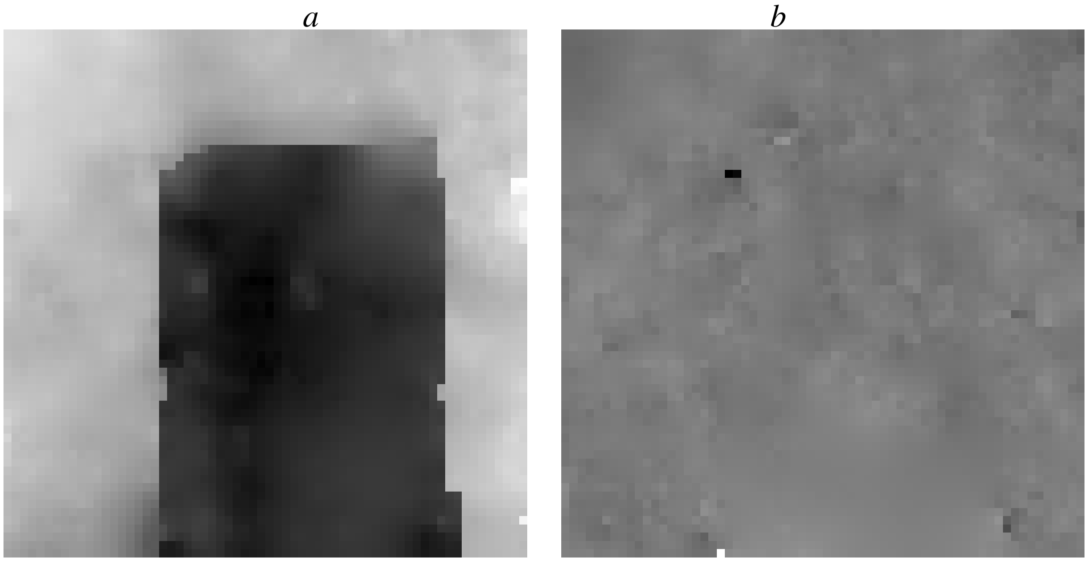


Figure 4.17: **The Pepsi Sequence.** *a, b)* Horn and Schunck optical flow. *c, d)* Anandan's coarse-to-fine SSD correlation.

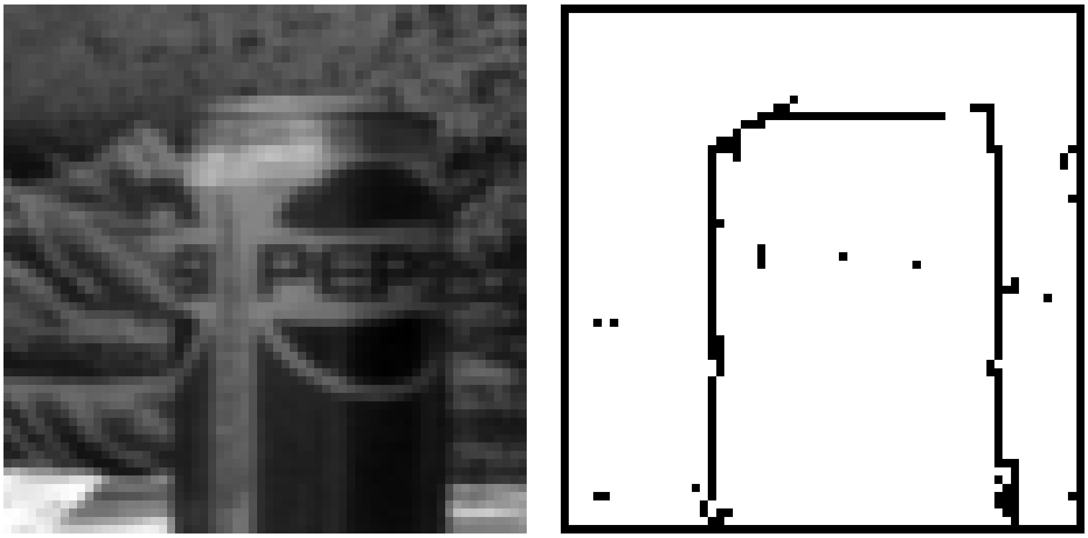


a b



c d

Figure 4.18: **The Pepsi Sequence.** *a, b)* Convex approximation (first stage of GNC algorithm). *c, d)* Robust gradient results: optical flow.



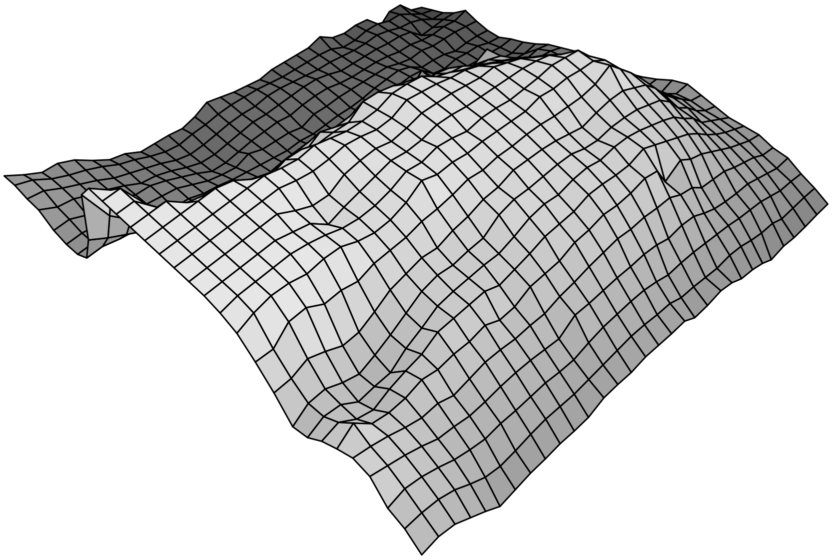
a *b*

Figure 4.19: **The Pepsi Sequence.** Motion discontinuities. *a)* First image in sequence. *b)* Motion boundaries detected using the robust-gradient approach. Boundaries correspond to sites where the spatial coherence constraint is violated.

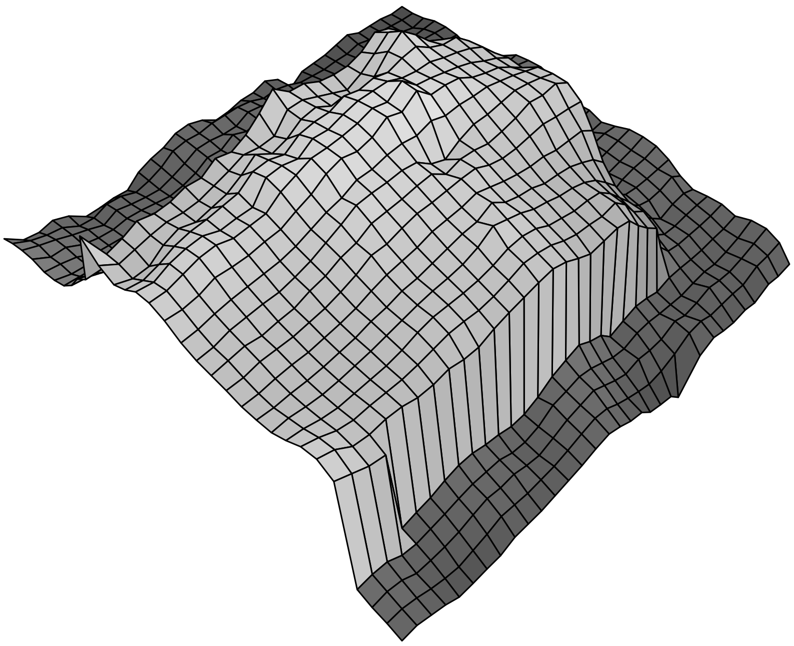
the plots in Figure 4.20. Here, the magnitude of the flow vectors is plotted. The sharp discontinuity present in the robust solution (Figure *b*) is lacking in the least-squares estimate (Figure *a*).

The Tree Sequence

Finally, we consider a more complex example with many discontinuities and motion greater than a pixel. The first two 233×256 images in the SRI tree sequence are seen in Figure 4.21. Figure 4.22 shows the motion discontinuities where the outlier threshold is exceeded for the smoothness constraint. As expected, the least-squares flow estimate (Figures 4.23 *a* and *b*) shows a good deal of over-smoothing. The robust flow, shown in figures *c* and *d* exhibits sharp motion boundaries, yet still recovers the smoothly varying flow of the ground plane.



a



b

Figure 4.20: **Pepsi flow magnitude.** The magnituded of the flow vectors is plotted for *a)* the least-squares solution, *b)* the robust-gradient solution.

a *b*

Figure 4.21: SRI tree image sequence; first two images.

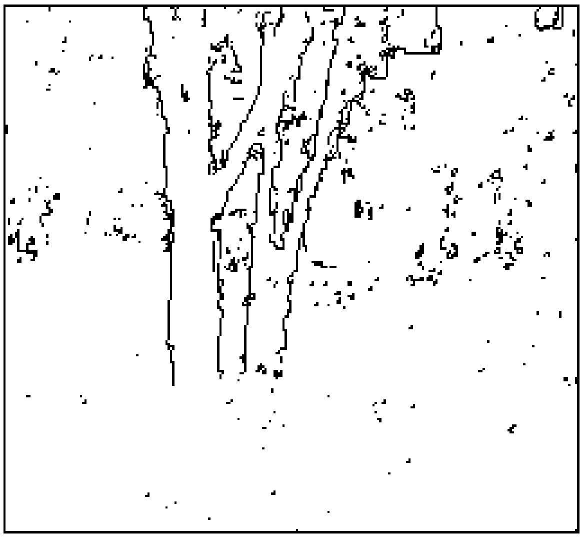


Figure 4.22: Tree Sequence discontinuities.

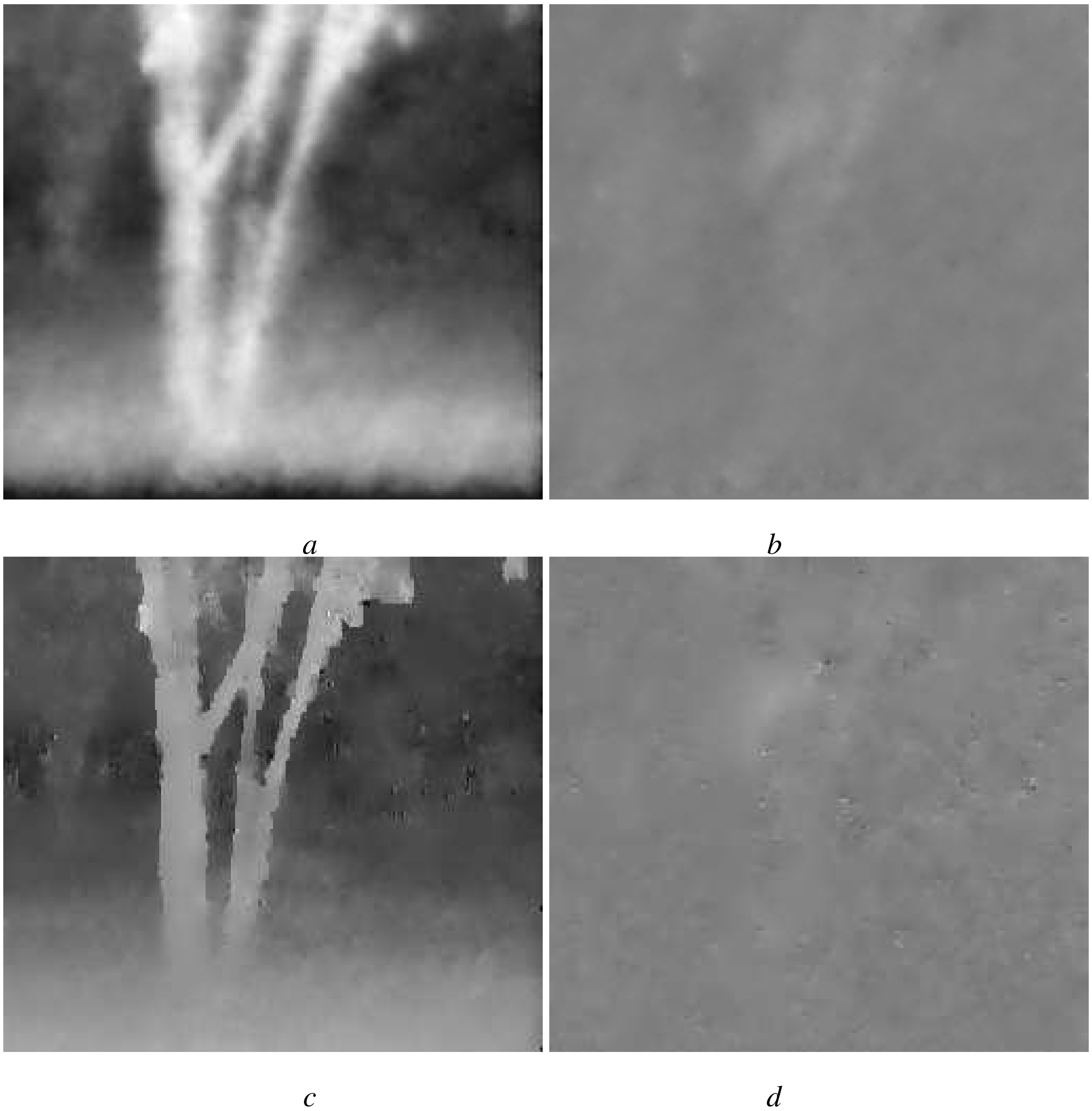


Figure 4.23: Tree sequence results; (a, b) least-squares estimates of horizontal and vertical flow, (c, d) robust-gradient estimate.

Chapter 5

Temporal Continuity

Until now, this thesis has addressed the problem of robustness in the estimation of optical flow between two frames. Motion sequences, however, typically contain more than two frames which means that motion algorithms should be capable of both processing long sequences in a reasonable way and exploiting the increased information available from multiple frames. While the remainder of this thesis will be devoted to the problems posed by image sequences, the issues of robustness related to motion discontinuities remain present and thus the robust estimation framework will be taken as a foundation on which to build a new framework for incremental motion estimation.

The robust estimation formulation of the previous chapters resulted in a computationally expensive non-convex minimization problem. Processing every pair of images in a sequence using such a technique is computationally infeasible given current hardware. Instead, we should be able propagate information from frame to frame in a principled way and exploit this information to reduce the cost of estimating the flow between any pair of frames.

Using long image sequences has another, possibly more significant, benefit; that is, it allows us to exploit information over time to improve the estimation of optical flow. By using information from a sequence of images, optical flow estimates can be refined as more information becomes available.

We start by formulating a new constraint which embodies our assumptions about the motion of objects over time. This *temporal continuity* constraint has the important effect of allowing us to predict the image motion at some future point in time based on the current flow field. As with the data conservation and spatial coherence constraints, the temporal continuity assumption is formulated in the image plane as a constraint that becomes a term in the objective function. Since the constraint is treated in exactly the same way as the other constraints, it fits naturally within the robust estimation framework. We next develop an *incremental minimization framework* for recovering flow estimates over time. Finally, we describe how our framework relates to incremental techniques based on recursive estimation.

5.1 A Temporal Continuity Constraint

When we consider more than two frames we have additional information that can be brought to bear on the estimation problem. In a stationary environment, the smooth motion of an observer causes surfaces to move in a predictable way. Even the motion of independently moving objects is often quite predictable due to the laws of physics.

This predictable motion of surfaces in the world gives rise to a predictable change in image velocity over time which we call *temporal continuity*. Consider Figure 5.1a in which a bar is moving to the right in the image plane. Since surfaces in the world tend to persist, and their motion is predictable, the bar sweeps out a volume in space and time as shown in Figure 5.1b. This property of images has been previously exploited, in a variety of ways, by other authors, for the estimation of image motion.

Figure 5.2 shows an x–t-slice through the spatiotemporal cube. As noted by Adelson and Bergen [1985] the orientation of the edges in the slice are determined by the horizontal motion of the bar. Optical flow can be recovered by recovering orientation in space-time using spatiotemporally oriented filters [Heeger, 1987].

Bolles, Baker, and Marimont [1987] use exactly the same information in their epipolar-

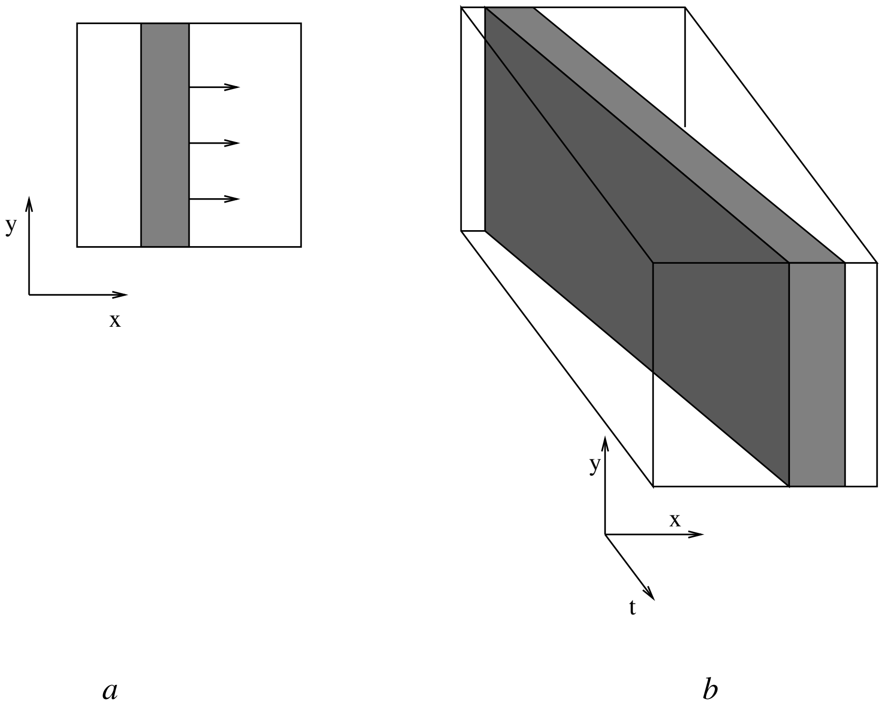


Figure 5.1: Continuity in space and time [Adelson and Bergen, 1985]. In figure *a* a bar is moving to the right in the image plane. In space and time, the bar produces a spatiotemporal block (Figure *b*).

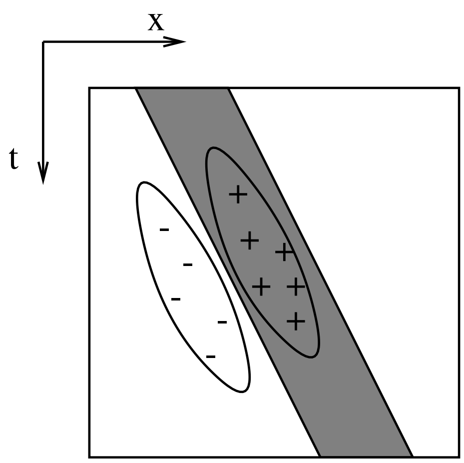


Figure 5.2: Horizontal motion can be determined by computing the orientation of edges in an x–t-slice of the space-time cube using oriented spatiotemporal filters.

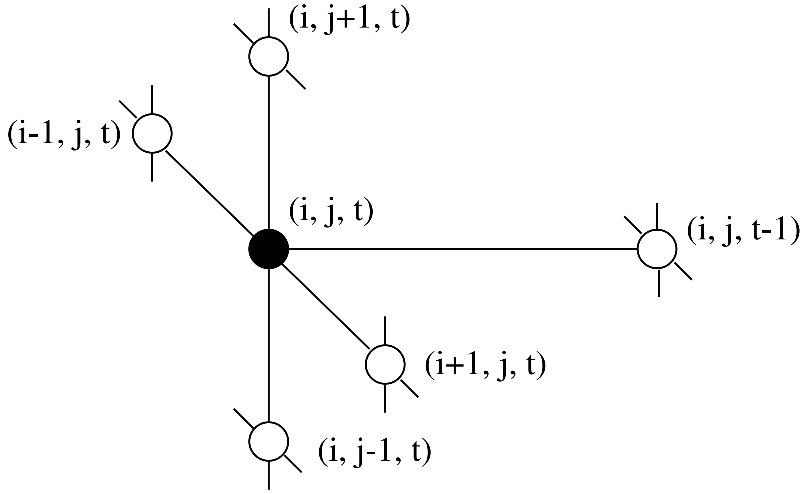


Figure 5.3: The standard neighborhood system can be extended to a spatiotemporal neighborhood by adding a neighbor in time [Murray and Buxton, 1987].

plane image analysis. In their case, they restrict camera motion to a horizontal plane so that all motion can be determined by examining an x–t-slice of the space-time cube. Given known camera motion, they recover the depth of scene features by examining the orientation of edges detected in the x–t-slices.

An alternative approach is pointed out by Murray and Buxton [1987], who extend the standard spatial neighborhood system to include neighbors in both space and time as illustrated in Figure 5.3. They then define a crude temporal continuity constraint, E_T, which assumes that the flow at a site remains constant over time:

$$E_T(\mathbf{u}(x,y,t), \mathbf{u}(x,y,t-1)) = \begin{cases} -w, & \mathbf{u}(x,y,t) = \mathbf{u}(x,y,t-1), \\ w, & \text{otherwise}, \end{cases} \tag{5.1}$$

for a positive weight w. Murray and Buxton introduced spatial discontinuities using a line-process formulation, but found that, given their approach, adding a temporal line process had an overall detrimental effect on the estimation.

We take a different approach which extends our robust formulation of the two frame estimation problem. We treat temporal continuity as a constraint on image velocity, formulate it to be robust, and incorporate it into the robust estimation problem. For example, consider the simple assumption that the image velocity of a surface patch is constant over time. While this is not a realistic assumption, it is illustrative. If we know the flow $\mathbf{u}(x, y, t)$ at a particular instant in time, then we can predict what the flow will be at the next instant, $t + \delta t$, as follows:

$$\mathbf{u}^-(x, y, t) = \mathbf{u}(x - u\delta t, y - v\delta t, t - \delta t), \tag{5.2}$$

where $\mathbf{u}^-$ is the "predicted" flow field. This equation corresponds to a backwards warp of the flow field *by our current estimate of the flow*.[1]

A more realistic assumption would be one of *constant acceleration* in the image plane, which can be formulated as:

$$\begin{aligned} \mathbf{u}^-(x, y, t) \;\; = \;\; & \mathbf{u}(x - u\delta t, y - v\delta t, t - \delta t) + \\ & \frac{\partial}{\partial t}\mathbf{u}(x - u\delta t, y - v\delta t, t - \delta t)\delta t, \end{aligned} \tag{5.3}$$

where the acceleration is approximated by:

$$\frac{\partial}{\partial t}\mathbf{u}(x, y, t) \approx (\mathbf{u}(x, y, t) - \mathbf{u}^-(x, y, t)). \tag{5.4}$$

This is still an idealization of the temporal evolution of image motion even for simple scenes and observer motions. For example, at an occlusion boundary the occluded surface abruptly disappears and, hence, temporal continuity is lost. There is, however, psychophysical evidence to suggest that humans do represent constant acceleration, as opposed to constant velocity, under appropriate conditions [Freyd, 1983].

The temporal continuity constraint is formulated here in terms of image motion and not scene motion. As with the other constraints, choosing the correct model is important for both accuracy and robustness. Constant image-plane acceleration is only an initial approximation. There needs to be more study of what constitutes a good temporal model, and how

[1]As we saw in Chapter 2, backwards warping of the intensity image is commonly used in coarse-to-fine approaches. The use here is slightly different. The flow estimate is being used to warp itself, and in doing so predict what the motion will be in the future.

the appropriate temporal model can be adaptively chosen, from among a continuum of models, to best capture the temporal evolution of the sequence.

Given a prediction of image motion we can formulate a temporal continuity constraint:

$$\mathbf{u}(x, y, t) = \mathbf{u}^{-}(x, y, t), \tag{5.5}$$

$$E_T(\mathbf{u}, \mathbf{u}^{-}) = \rho(\mathbf{u}(x, y, t) - \mathbf{u}^{-}(x, y, t)), \tag{5.6}$$

which states that the current estimate $\mathbf{u}$ should not differ from the predicted flow $\mathbf{u}^{-}$. Since we expect the constraint to be violated, and we have no statistical model for how these violations will occur, we choose ρ to be a robust estimator. This allows our estimate to differ from the prediction in cases where the motion is not predicted by the model.

We now formulate our objective function as a combination of the data, spatial, and temporal constraints:

$$E(\mathbf{u}, \mathbf{u}^{-}) = \beta_D E_D(\mathbf{u}) + \beta_S E_S(\mathbf{u}) + \beta_T E_T(\mathbf{u}, \mathbf{u}^{-}), \tag{5.7}$$

where the β_i control the relative importance of the terms, and where $\mathbf{u}^{-}$ is determined by the constant acceleration assumption. Once again, this function may be non-convex and in the next section we examine how it can be minimized more efficiently.

The objective function is composed of three separate terms, each of which is formulated to account for violations. This approach assumes that violations of the constraints are independent; for example, image noise might cause the data term to be violated, but the best flow field is achieved when the spatial and temporal constraints are still enforced. This independence is crucial for robust recovery, since it allows any one term to be violated without removing the effect of the other constraints.

5.2 Incremental Estimation

Most previous approaches to exploiting long image sequences, including spatiotemporal filtering and epipolar-plane analysis, involve local batch computations; a number of images are collected and then processed. In contrast, we are interested in incrementally processing a sequence of images to achieve the goals of anytime access, temporal refinement, computation reduction, and adaptation. It is entirely possible that the locally batch approaches can be made incremental as is demonstrated by Baker's extension of epipolar-plane analysis over time using an incremental process called the "Weaving Wall" [Baker, 1989; Baker, 1988]. The focus of this chapter, however, is on how temporal continuity can be exploited to extend a minimization scheme over time.

First, consider the two frame motion estimation problem. Between every pair of frames we compute the optical flow using the data conservation and spatial coherence terms. But, since the robust estimation problem is non-convex such an approach involves a computationally expensive, iterative minimization procedure between each frame and does not meet the goals of incremental estimation.

Now consider a simple incremental strategy based on the assumption of Murray and Buxton [1987] that the flow at a site does not change over time. This leads to the following algorithm:

Algorithm 1:
  $\mathbf{u}, \mathbf{u}^- \leftarrow [0, 0]$
  `for each` image
    $\mathbf{u} \leftarrow \min_{\mathbf{u}} E(\mathbf{u}, \mathbf{u}^-)$; minimize E beginning at $\mathbf{u}^-$
    $\mathbf{u}^- \leftarrow \mathbf{u}$; Murray and Buxton assumption
  `end`,

where the objective function E contains the data, spatial, and temporal constraints. For each new image we begin minimizing the objective function starting with the flow estimate $\mathbf{u}^-$ from the previous time instant. The idea is that $\mathbf{u}^-$ provides a good initial estimate of the flow and, hence, whatever minimization strategy is used should converge quickly to a global minimum. Faster convergence, however, is not guaranteed and, hence, the goal of computation reduction is not met. The estimate $\mathbf{u}^-$ also provides a prediction of the flow which is used in the temporal continuity constraint.

The most obvious problem with this approach is that the assumption that the flow at a site does not change over time is rarely satisfied. Such a simple assumption does not provide a useful prediction of the flow at the next time instant. The approach can be improved by employing a more realistic model of temporal continuity. For example, consider the following algorithm which assumes constant acceleration:

Algorithm 2:

$\mathbf{u}, \mathbf{u}^- \leftarrow [0, 0]$

`for each` image

$\mathbf{u} \leftarrow \min_{\mathbf{u}} E(\mathbf{u}, \mathbf{u}^-)$; minimize E beginning at $\mathbf{u}^-$

$\mathbf{u} \leftarrow \mathbf{u} + (\mathbf{u} - \mathbf{u}^-)$; constant acceleration

$\mathbf{u}^-(x, y) \leftarrow \mathbf{u}(x - u, y - v)$; warp flow by current estimate

`end`.

The constant acceleration assumption allows us to warp the flow field given the current estimate. This has two advantages: first, it provides a better initial starting position for the minimization process and, second, it provides a better estimate for use in the temporal constraint.

Even though we have an initial estimate, minimizing E may still be a computationally expensive proposition. Assume that a continuation method like GNC or a stochastic technique like simulated annealing is employed. Either choice requires a control parameter, T, which is updated as the objective function is minimized. In the case of Algorithm 1 and Algorithm 2, a complete minimization is performed for each new image. This is infeasible for real-time applications because the amount of computation is not known *a priori*. In order to maintain a fixed amount of computation between frames Algorithm 2 is modified as follows:

Algorithm 3:

```
u, u⁻ ← [0, 0]
T ← initial value
n ← fixed, small number of iterations
for each image
    for n iterations
        u ← minimize(E, u, u⁻, T)      ; one-step minimization
        T ← f(T)                       ; update the continuation parameter
    end
    u ← u + (u − u⁻)                   ; constant acceleration
    u⁻(x, y) ← u(x − u, y − v)         ; warp flow by current flow
end,
```

where we have now made the control parameter explicit. The new algorithm restricts the number of iterations of the minimization routine performed for each new image. Additionally, the control parameter is now updated over time, resulting in a minimization that proceeds over the image sequence.

This new algorithm meets all but one of our requirements for incremental estimation. First, a motion estimate is always available. Second, the estimate improves over time as more iterations of the minimization strategy are performed and as the continuation parameter is adjusted. Third, we have limited the amount of computation between frames to a constant amount.

This leaves the requirement of adaptability unaddressed; that is, the algorithm assumes that the flow estimate is monotonically improving over time and is unable to respond to sudden changes in the scene which are reflected in sudden changes to the objective function. As a result, it is possible for the algorithm to get "stuck" in a local minimum. In locations where a violation occurs, the algorithm should ignore its previous estimates and, essentially, start over. We modify Algorithm 3 to be adaptive to the most common violation; motion discontinuities:

```
Algorithm 4:
    u, u⁻ ← [0, 0]
    T ← initial value at every site
    n ← fixed, small number of iterations
    for each image
        for n iterations
            u ← minimize (E, u, u⁻, T)       ; peform n iterations
            T(x, y) ← f(T(x, y))             ; update the continuation parameter
        end
        u ← u + (u − u⁻)                     ; constant acceleration
        u⁻(x, y) ← u(x − u, y − v)           ; warp flow by current flow
        T(x, y) ← T(x − u, y − v)            ; warp control parameter
        if location (x, y) is occluded or disoccluded then
            T(x, y) ← initial value
            u, u⁻ ← [0, 0]
        end if
    end.
```

When a motion boundary is detected, the continuation parameter is reset and the flow estimate is reinitialized. Thus, the value of the continuation parameter, **T**, at each site in the image varies independently, whereas before, there was a single value of T, for all of the sites. Since the new control parameters are associated with particular sites, it is necessary to warp these values along with the flow field.

We have implemented two different algorithms within this framework. The first is an incremental version of simulated annealing, and the second is an incremental version of GNC. We will explore the details of these implementations in the following chapters.

What can be said about the global convergence of the incremental minimization algorithm? First, the optimization schemes required for minimizing a non-convex objective function are typically not guaranteed to converge to the global minimum even in the static case.[2] Not surprisingly, the incremental version cannot hope to do better.

[2]This includes simulated annealing, since, despite theoretical convergence in the limit with a logarithmic cooling schedule, practical applications typically rely on faster, linear, cooling schedules and do not have the luxury of infinite computation.

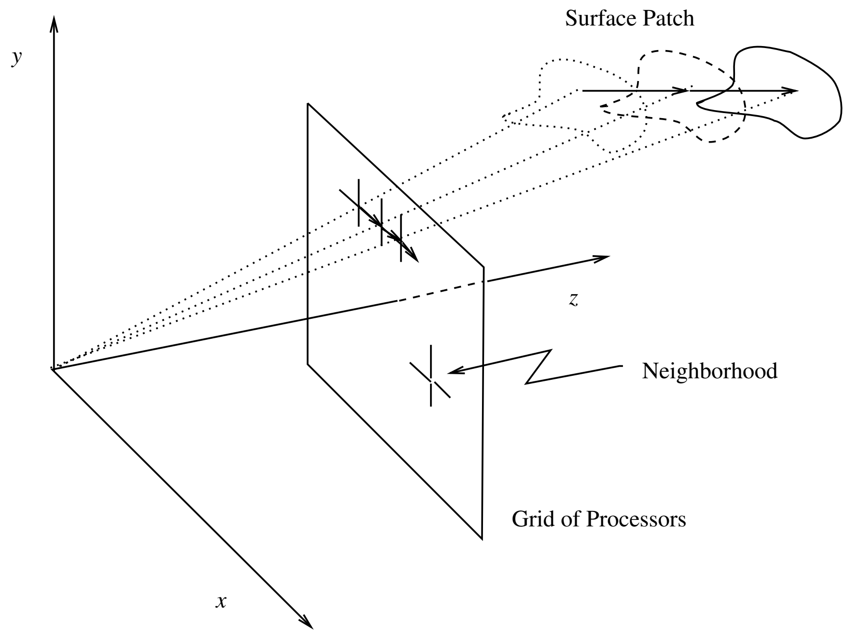


Figure 5.4: Incremental Model.

While no theoretical convergence results exist, numerous experiments have been performed with two different incremental algorithms. When presented with synthetic sequences containing noise, the algorithms converge rapidly to the true motion. In real scenes, the approaches converge to results that are qualitatively equivalent, or superior, to the published results of standard two-frame algorithms.

5.2.1 The Computational Model

One way to view the incremental minimization scheme is illustrated in Figure 5.4. Imagine a grid of processors where there is a processor at each site responsible for tracking the projected motion of a small surface patch in the image. As the patch moves, the processor estimates its motion using the three constraints we have described. When all the processors have decided where their patch is moving, they communicate with their neighbors to pass along all the information about the patches to the appropriate processor; that is they warp and resample the flow field.

A number of algorithms have been implemented within the incremental minimization framework. The implementation was performed on a Connection Machine (CM-2) [Hillis, 1985] which is a massively parallel SIMD machine. By configuring the machine's processors as an array, it is possible to have a physical processor at every site in the image. The simple neighborhood communications required by the model are extremely efficient when implemented on an array of processors. For the first-order neighborhood system, minimizing the objective function only requires nearest neighbor communication. Image warping may involve slightly more distant communication, but is only performed once per image.

In fact, the Connection Machine is a much more flexible architecture than is required. A simple fixed array of processors would suffice, and given the simple regular structure, special purpose hardware could be developed. For example, the robust minimization problem may be mapped onto analog resistive networks [Harris *et al.*, 1990; Koch *et al.*, 1988], where the minimum of the objective function is determined by the stationary voltage distribution of the network.

As we will see in Chapter 8, the incremental minimization model is quite general, and each processor can maintain more information about the scene than simply motion vectors. For example, they may maintain information about the depth, orientation, intensity, or curvature of the surface patch. The effective tracking of surface patches by the model allows many common vision algorithms to be formulated and solved in an incremental fashion.

5.2.2 Large Motions

To account for large motions within this framework we need to extend the coarse-to-fine strategies developed in Chapter 2. The most obvious approach is to take the current flow estimate and use it to construct a pyramid of flow fields. The coarse-level flow field can then be used to initialize the coarse-to-fine strategy at the next time instant.

This is a particularly poor approach. A great deal of computational effort goes into re-

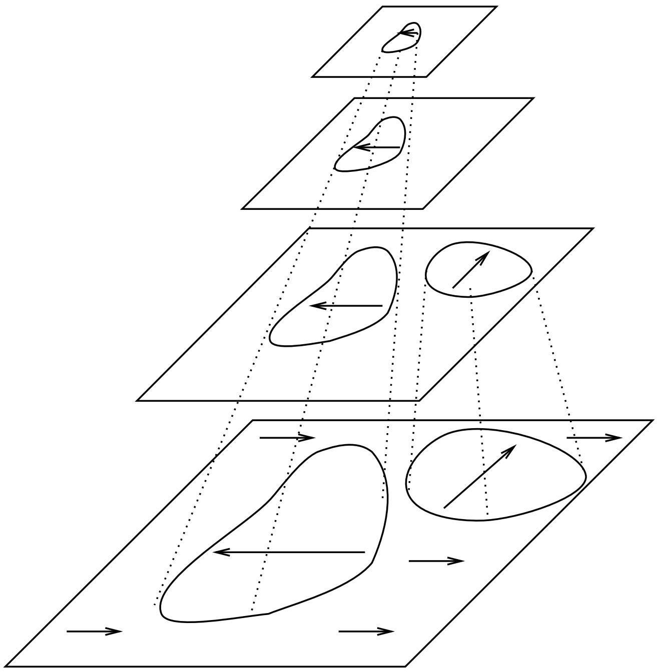


Figure 5.5: Coarse-to-fine "flow through" strategy. The motion at the finest level of the pyramid is determined by the coarsest level at which the motion is greater than half a pixel. This coarse motion then "flows through" to the fine levels, being appropriately scaled, but not refined.

fining a flow estimate in the coarse-to-fine scheme. In the simple approach above, most of that work is abandoned by using a coarse version of the flow estimate when a new image is acquired. Instead what is needed is a way to combine information across levels in the pyramid without undoing work that has been performed at the fine levels. Experiments have been performed with two hierarchical schemes: a *flow-through* strategy and a *coarse-to-fine-when-changed* approach.

In the flow-through scheme, each level in the pyramid is processed in parallel. One way to view this is as a pyramid of spatiotemporally tuned motion detectors, where at coarse levels of the pyramid large motions are detected and at fine levels small motions are detected. Unlike the coarse-to-fine approach, there is no refinement of motions detected at the coarse level. The final motion estimate at a site is determined by the level best suited to computing the motion. With this scheme, small motions are known with greater absolute accuracy, but relative accuracy is the same across levels. The approach is illustrated in Figure 5.5.

The flow-through algorithm can be summarized in three steps:

1. Given a current flow estimate, create a pyramid of flow estimates,

2. At each level in the pyramid perform, in parallel, incremental minimization to improve the flow estimate at those sites for which the horizontal and vertical components of the flow are less than one pixel. Hold the estimate at all other sites fixed.

3. Combine estimates across levels using a flow through strategy:

Coarse-to-Fine-Flow-Through:
    $p \leftarrow$ coarse-level
    $\mathbf{u} \leftarrow \mathbf{u}^p$
    `for` p `from` coarse-level $+1$ `to` fine-level `do`
        $\mathbf{u} \leftarrow$ project($\mathbf{u}, p$) ; project multiplying by 2
        `if` $|u(x,y)| < 1$ `and` $|v(x,y)| < 1$ `then`
                $\mathbf{u}(x,y) \leftarrow \mathbf{u}^p(x,y)$
    `end`

The parallel nature of flow-through approach is appealing, but it may not be appropriate for applications like structure from motion, in which the accuracy of the flow estimates is critical. In cases like this, we want to have the benefits of the refinement approach while maintaining the benefits of incremental estimation. We propose a limited coarse-to-fine approach in which information flows from coarse levels to finer levels only when the levels disagree significantly:[3]

[3] Heeger has dubbed this the "coarse-to-fine-sanity-check" method.

```
Coarse-to-Fine-When-Changed:
  u^{-1} ← [0,0]
  for p from coarse-level to fine-level do
      if |u^p − project(u^{p−1}, p)| > 0.5 then          ; check if changed
          u^p ← project(u^{p−1}, p)                      ; reset from above
      end
      u^p ← refine(E(u^p))
  end
```

If the motion is changing rapidly in an area, the algorithm will use coarse information to detect the motion and then refine it. In areas of the image that are moving predictably, no information flows from above, but, rather, the estimates are continually improved at the appropriate level in the pyramid.

5.3 Relationship to Recursive Estimation

To better understand the incremental estimation framework this section examines its relationship to more traditional incremental techniques. In particular, we examine the Kalman filter (see [Gelb, 1974] for an overview) which is an optimal filtering technique for estimating the state of a *linear* system. While the Kalman filter has been used extensively in motion and structure estimation in feature based approaches [Faugeras *et al.*, 1987], we will focus here on computing dense flow estimates. Such a Kalman filter based flow algorithm has recently been implemented by Singh [1992a] and based on work in incremental depth estimation [Heel, 1991; Matthies *et al.*, 1989; Szeliski, 1988].

5.3.1 The Kalman Filter

We begin by reviewing the basic discrete Kalman filter. The filter exploits three explicit probablistic models: the *system* model, the *measurement* model, and the *prior* model. The algorithm works in two phases. The *prediction* phase extrapolates the current state to the next time instant, while the *update* phase integrates predicted estimates with new measurements.

Given a dynamical system with state $\mathbf{u}_k$, at time k, we describe the changes in state over time by a *system model:*

$$\mathbf{u}_k = \Phi_k \mathbf{u}_{k-1} + \mathbf{q}_k, \qquad \mathbf{q}_k \sim N(0, \mathbf{Q}_k), \tag{5.8}$$

where Φ_k is a known transition matrix which maps state variables at one time instant to state variables at the next time instant with the addition of Gaussian noise having covariance, $\mathbf{Q}_k$. In the case of motion estimation, these state variables are the flow vectors.

The *measurement matrix*, $\mathbf{H}_k$, relates noisy measurements, $\mathbf{d}_k$, to the current state with the addition of Gaussian noise having covariance, $\mathbf{R}_k$:

$$\mathbf{d}_k = \mathbf{H}_k \mathbf{u}_k + \mathbf{r}_k, \qquad \mathbf{r}_k \sim N(0, \mathbf{R}_k). \tag{5.9}$$

This defines the *measurement model.*

A *prior model* describes the system state and covariance before any measurements have been made:

$$\mathbf{u}_0 \sim N(\hat{\mathbf{u}}_0, \mathbf{P}_0), \tag{5.10}$$

where $\mathbf{P}_0$ is the prior covariance matrix. Additionally, we assume that the measurement noise $\mathbf{r}_k$ and the system noise $\mathbf{q}_k$ are uncorrelated.

Our goal is to estimate $\mathbf{u}_k$ given our knowledge of the system's behavior over time as defined by the above models and given some measurement $\mathbf{d}_k$. Given a current estimate $\hat{\mathbf{u}}^+_{k-1}$ we obtain a predicted estimate $\hat{\mathbf{u}}^-_k$ and a new prior covariance matrix $\mathbf{P}^-_k$ at the next time instant using the *prediction* equations:

$$\hat{\mathbf{u}}^-_k = \Phi_{k-1} \hat{\mathbf{u}}^+_{k-1} \tag{5.11}$$

$$\mathbf{P}^-_k = \Phi_{k-1} \mathbf{P}^+_{k-1} \Phi^T_{k-1} + \mathbf{Q}_{k-1}. \tag{5.12}$$

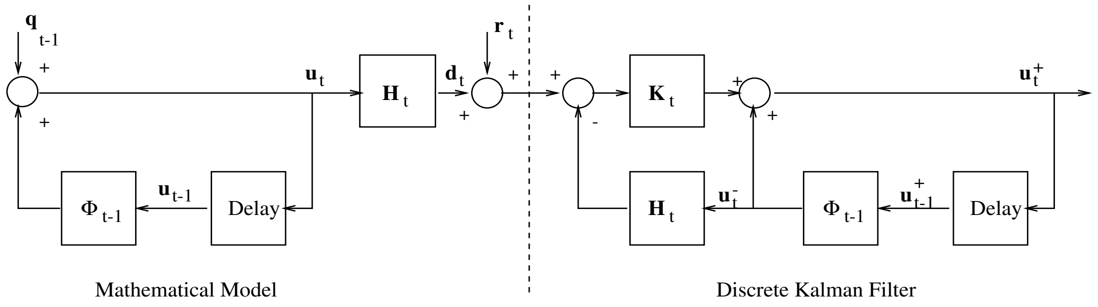


Figure 5.6: System Model and Discrete Kalman [Gelb, 1974].

The *update* phase integrates a new measurement into the current state while accounting for the measurement noise:

$$\hat{\mathbf{u}}_k^+ \quad = \quad \hat{\mathbf{u}}_k^- + \mathbf{K}_k(\mathbf{d}_k - \mathbf{H}_k\hat{\mathbf{u}}_k^+). \tag{5.13}$$

This updates the estimate by adding to the prediction, the difference between the prediction and the measurement scaled by the Kalman filter *gain matrix* which is defined as:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \tag{5.14}$$

Finally, the covariance of the new estimate is updated:

$$\mathbf{P}_k^+ = (\mathrm{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-. \tag{5.15}$$

These two equations can be simplified to [Gelb, 1974]:

$$(\mathbf{P}_k^+)^{-1} \quad = \quad (\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k, \tag{5.16}$$

$$\mathbf{K}_k \quad = \quad \mathbf{P}_k^+ \mathbf{H}_k^T \mathbf{R}_k^{-1}. \tag{5.17}$$

The filter can now be summarized by the block diagram in 5.6. The left portion of the figure describes the system model and is purely a mathematical abstraction of the true system and measurement processes. The portion on the right is essentially a flow chart describing the implemented portion of the discrete Kalman filter.

5.3.2 Measurements

For motion estimation, measurements will be estimates of the image flow between a pair of images in an image sequence. This estimate can be computed using any number of techniques. All that is required is that for each new frame, we compute a new flow measurement and that this measurement have an associated covariance matrix. This covariance matrix $\mathbf{R}_k$ is estimated from the data which corresponds to using an *adaptive Kalman filter* [Gelb, 1974].

For example, in Chapter 2 we showed the correlation-based approach of Singh [1992a] in which he computes a least-squares estimate of the motion and its covariance matrix from the SSD surface. Likewise, Anandan's confidence measures [Anandan, 1989], or Simoncelli *et al.*'s [1991] probability distributions might be used to estimate the covariance. Heel [1989] proposes a related method based on the SSD surface. Recall that the SSD surface is defined as:

$$E_D(\mathbf{u}) = \sum_{(x,y)\in\mathcal{R}} [I(x,y,t) - I(x + u\delta t, y + v\delta t, t + \delta t)]^2.$$

If $\mathbf{u}^*$ is the best displacement, then Heel defines the variance in u and v as:

$$\sigma_u^2 = \frac{E_D(\mathbf{u}^*)}{E_D(\mathbf{u}^*)_{xx}} \text{ and } \sigma_v^2 = \frac{E_D(\mathbf{u}^*)}{E_D(\mathbf{u}^*)_{yy}},$$

where the subscripts indicate the second partial derivatives of the error surface with respect to x and y respectively. With these measures, the variance is high when the error $E_D(\mathbf{u}^*)$ is high or the curvature at the minimum is low. Low curvature occurs when there is little texture and hence the measurements are unreliable.

5.3.3 Prior Model

The prior model should express the assumption of spatial coherence. This can be done by modeling spatial smoothness in terms of the prior covariance matrix $\mathbf{P}$. In the first-order, or membrane, case the smoothness energy is defined as:

$$E_p(\mathbf{u}) = \frac{1}{2}\sum_{(i,j)}[(u_{i+1,j} - u_{i,j})^2 + (v_{i,j+1} - v_{i,j})^2]. \tag{5.18}$$

Collecting all the $u_{i,j}$ into a vector **u** we can rewrite the smoothness energy as [Szeliski, 1988]:

$$E_p(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{A}_p\mathbf{u}. \tag{5.19}$$

The prior covariance is defined as: $\mathbf{P}_0 = \mathbf{A}_p^{-1}$.

With no other knowledge, it may be reasonable to begin with a prior model which states that the world is smooth. But as information about discontinuities becomes available, the prior model should change to take them into account. This corresponds to using an adaptive Kalman filter [Gelb, 1974].

5.3.4 Prediction

The constraint of temporal continuity is embodied in the state transition matrix Φ_k. The current flow estimate is used to predict the location that a site has moved to and, hence, the flow at that new location at the next instant in time. Since it is unlikely that the flow components are integers, the new location of a site will fall somewhere between the regular pixel-grid locations. To keep a fixed grid of sites, the flow field must be interpolated and resampled after the state update is performed. This means that the state transition matrix Φ_k is not a simple linear operation which is known in advance. The process can be implemented as a warping operation that takes a flow estimate and produces a predicted flow field. We must also specify how the state covariance matrix is updated. This algorithmic approach is the one taken in all current implementations [Heel, 1991; Matthies *et al.*, 1989; Singh, 1991; Szeliski, 1988] which implement a warping procedure similar to those presented in Chapter 2.

5.3.5 The Kalman Filter and Estimation

The above discussion has presented one way of viewing the Kalman filter. An alternative view is to begin with the underlying objective function and derive the optimal closed-form estimate. In particular, the update stage of the Kalman filter finds the $\mathbf{u}^+$ that minimizes the following objective function [Gelb, 1974]:

$$E(\mathbf{u}^+) = \frac{1}{2}(\mathbf{u}^+ - \mathbf{u}^-)^T(\mathbf{P}^-)^{-1}(\mathbf{u}^+ - \mathbf{u}^-) + \frac{1}{2}(\mathbf{d} - \mathbf{H}\mathbf{u}^+)^T\mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\mathbf{u}^+). \tag{5.20}$$

This can be seen by examining where this function is minimized. We differentiate with respect to $\mathbf{u}^+$ and set the result to zero:

$$\frac{\partial E}{\partial \mathbf{u}^+} = ((\mathbf{P}^-)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\mathbf{u}^+ - (\mathbf{P}^-)^{-1}\mathbf{u}^- - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d} = 0. \tag{5.21}$$

If we take $(\mathbf{P}^+)^{-1} = (\mathbf{P}^-)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$, then substituting into equation (5.21), we have:

$$\begin{aligned} 0 \;=\; & ((\mathbf{P}^+)^{-1} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\mathbf{u}^+ \\ & - ((\mathbf{P}^+)^{-1} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\mathbf{u}^- - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}. \end{aligned}$$

Simplifying and multiplying through by $\mathbf{P}^+$ we get:

$$\begin{aligned} 0 &= (\mathbf{P}^+)^{-1}\mathbf{u}^+ - (\mathbf{P}^+)^{-1}\mathbf{u}^- + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{u}^- - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}, \\ &= \mathbf{u}^+ - \mathbf{u}^- + (\mathbf{P}^+)\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{u}^- - \mathbf{d}). \end{aligned}$$

This is simply:

$$\begin{aligned} \mathbf{u}^+ &= \mathbf{u}^- + \mathbf{P}^+\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\mathbf{u}^-), \\ &= \mathbf{u}^- + \mathbf{K}(\mathbf{d} - \mathbf{H}\mathbf{u}^-), \end{aligned} \tag{5.22}$$

since from equation (5.17), we have $\mathbf{K} = (\mathbf{P}^+)\mathbf{H}^T\mathbf{R}^{-1}$. Thus we have derived the Kalman filter update equation (5.13) from the objective function. Viewed in this way, the Kalman filter update is simply minimizing the objective function (5.20).

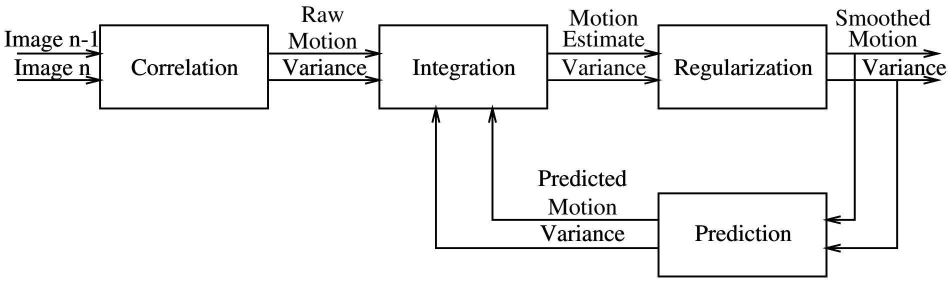


Figure 5.7: Kalman Filter Implementation [Matthies *et al.*, 1989; Szeliski, 1988].

The first term of this objective function specifies the temporal continuity constraint, while the second expresses data conservation. The minimum $\mathbf{u}^+$ is the estimate that minimizes the combined data error $(\mathbf{d} - \mathbf{H}\mathbf{u}^+)$ and the temporal continuity error $(\mathbf{u}^+ - \mathbf{u}^-)$ weighted by the appropriate covariance matrix. But the covariance matrix $\mathbf{P}^{-1}$ is simply the prior smoothness term $\mathbf{A}_p$. Hence, the minimum $\mathbf{u}^+$ is found subject to the first-order smoothness constraint.

5.3.6 Implementations

In the Kalman filter formulation of optical flow, the measurement covariance depends on the data, the prior smoothness model must be adaptive to cope with discontinuities, and the prediction phase requires a warping of the flow field. With these constraints, devising an optimal filter is difficult and computationally expensive, if possible at all.

Therefore current implementations typically rely on approximations which result in algorithms that fall outside the strict Kalman filtering framework. One approach that has been widely used [Heel, 1991; Matthies *et al.*, 1989; Singh, 1991; Szeliski, 1988] is to treat spatial smoothness as a separate regularization process applied to the current estimates. While outside the traditional Kalman filter framework, this approach works well in practice as it is simple and efficient and can be made sensitive to motion discontinuities [Singh, 1991].

Consider, for example, the approach of Matthies, Szeliski, and Kanade [1989] for incrementally estimating dense depth. Figure 5.7 is an adaptation of their incremental depth algorithm to the flow estimation problem. When a new image is acquired, the best estimate and its variance are found by an SSD approach. This measurement is then integrated with the previous estimate using the Kalman filter update equation and the variance estimate is revised. The current estimate is then smoothed using a standard regularization technique which could take into account motion discontinuities. The smoothed estimate is then used to warp the flow field to derive a predicted motion estimate.

Singh's [1992a] approach is based on this model, but uses a unique smoothing phase that takes into account the distribution of flow vectors in a local neighborhood. By computing a weighted least-squares estimate of the image velocity and a covariance matrix for the estimate, Singh is able to optimally combine the data conservation and spatial coherence information.[4] While the approach performs better at motion boundaries than standard regularization techniques, it still suffers from over-smoothing.

5.3.7 Comparison and Discussion

At first glance the Kalman filter and ISM approaches seem radically different. While they share the same goal of incremental and adaptive computation, they rest on different assumptions and make different implementation choices. This section highlights these differences to bring to light the relative advantages and disadvantages of each approach. Despite these differences, however, there are fundamental similarities in the approaches and each represents an instantiation of a general incremental paradigm.

The Kalman filter paradigm is computationally simple and is based on well developed estimation theory. We have also seen that a strict implementation of the filter has computational problems and, hence, practical approximations are made. First, the implementation of the spatial coherence assumption in the Kalman filter framework can result in expensive matrix inversions. Typically a separate regularization stage is used instead. Second, it is not

[4] Optimality depends on the assumptions of Gaussian noise and smooth motion being met.

clear how to incorporate motion discontinuities into the framework. If they are dealt with in the smoothing process in the standard way by adding line processes, the simplicity of the Kalman filter is lost. The compromise approach of Singh exploits the covariance estimate in the smoothing process and achieves better results at motion boundaries that straightforward regularization. Third, the prediction stage does not involve a predetermined linear operation, therefore, current approaches rely on a separate phase that warps the flow field based on the current flow estimates. Hence implementations of the approach can differ significantly from theory. Also, in interesting scenes, we have seen that the Gaussian noise assumption of the filter is violated at motion discontinuities. Yet, despite these simplifications and violations of the assumptions, the Kalman filter approach performs well in practice and achieves the goals of incremental estimation.

The incremental minimization approach was developed to specifically cope with the non-convex optimization problems resulting from the robust estimation framework. A key difference between the approaches is the treatment of the various constraints. In the Kalman filter framework, the data measurement takes place as a separate process which does not receive information about the state of the system. It therefore cannot be influenced by the current motion estimate or by the other constraints. This can be a serious problem in the case of multiple motions where there may be multiple peaks in the correlation surface and where each peak may have fairly high confidence (or low variance). It may not be possible to determine the "best" peak from the data measurements alone. The reason being, that the best interpretation is dependent on the other constraints.

In contrast, the incremental minimization approach takes the view that all the constraints are *jointly* minimized, hence the "best" motion is the one which, not only is a minimum in the SSD surface, but also minimizes the other constraints. In this way, all the constraints receive a unified treatment that allows the application of the robust estimation framework.

Despite their different motivations, the approaches have a great deal in common. Both

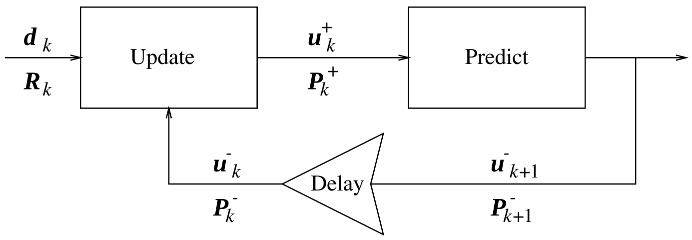


Figure 5.8: Kalman filter block diagram, (from [Heel, 1990]).

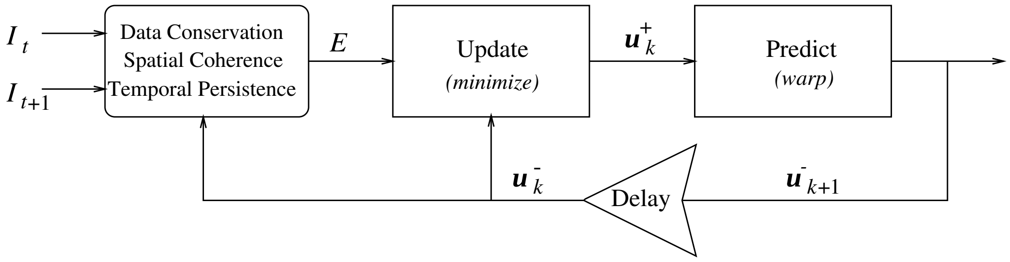


Figure 5.9: Incremental Minimization block diagram.

the incremental minimization framework and the recursive estimation framework share the same components: data conservation, spatial coherence, and temporal continuity. They differ in how these components fit together. Consider, for example, the simplified block diagram of the Kalman filter in figure 5.8. This is very similar to the diagram in figure 5.9 that captures the basic flow of the incremental minimization framework.

More significantly, the update stage of both approaches can be seen as minimizing an objective function. In the case of the Kalman filter, the least-squares formulation of the objective function results in a closed form solution. As we have seen, a least-squares approach can have problems in practice. If we replace that objective function with our robust objective function, and adopt an appropriate minimization strategy, we can essentially convert the Kalman filter framework into the incremental minimization framework. In the process, we loose the simplicity and efficiency of the simple discrete Kalman filter, while gaining the benefits of the robust minimization scheme.

Comparing the objective functions of the two approaches reveals another difference. In the Kalman filter framework, the spatial coherence constraint is applied to the difference between the estimated flow and the predicted flow:

$$(\mathbf{u}^+ - \mathbf{u}^-)^T (\mathbf{P}^-)^{-1} (\mathbf{u}^+ - \mathbf{u}^-).$$

This is a different notion of spatial coherence than that employed in this thesis where the constraint is appled to the flow estimate $\mathbf{u}^+$. This difference may be significant if the detection and preservation of motion boundaries is important.

Chapter 6

Incremental Stochastic Minimization

To make concrete the framework of the previous chapter, this chapter considers the incremental recovery of optical flow given the robust formulation proposed by Black and Anandan [1990b; 1991b]. We begin by reviewing this robust, correlation-based, formulation of the problem. Given this formulation, it is not possible to directly apply the GNC technique that was used for the robust gradient-based formulation. Instead we pose the problem in the context of Markov random fields and develop a stochastic minimization technique that is capable of recovering sub-pixel motion estimates. We then recast the stochastic minimization problem in the incremental minimization framework and demonstrate its performance on real and synthetic image sequences. We call this new algorithm *Incremental Stochastic Minimization (ISM)* [Black and Anandan, 1990b; Black and Anandan, 1991b].

6.1 Robust Formulation

As before, we pose the following minimization problem:

$$E(\mathbf{u}, \mathbf{u}^-) = \beta_D E_D(\mathbf{u}) + \beta_S E_S(\mathbf{u}) + \beta_T E_T(\mathbf{u}, \mathbf{u}^-). \tag{6.1}$$

where the data conservation term is formulated as robust correlation,

$$E_D(\mathbf{u}) = \sum_{(x,y)\in\mathcal{R}} \rho_D(I(x, y, t) - I(x + u, y + v, t + 1), \Delta_D), \tag{6.2}$$

with the following robust estimator:

$$\rho_D(x, \Delta_D) = \frac{-1}{1 + \left(\frac{x}{\Delta_D}\right)^2}. \tag{6.3}$$

We adopt a first order smoothness term:

$$E_S(\mathbf{u}_s) = \sum_{t \in \mathcal{G}_s} \rho_S(u_s - u_t, \Delta_S) + \rho_S(v_s - v_t, \Delta_S), \tag{6.4}$$

where ρ_S is the Geman and Reynolds robust estimator:

$$\rho_S(x, \Delta_S) = \frac{-1}{1 + \frac{|x|}{\Delta_S}}. \tag{6.5}$$

The temporal persistence term assumes constant acceleration and is formulated as:

$$E_T(\mathbf{u}, \mathbf{u}^-) \quad = \quad \rho_T(u - u^-, \Delta_T) + \rho_T(v - v^-, \Delta_T), \tag{6.6}$$

where $\rho_T = \rho_S$, with a possibly different value for Δ_T.

To efficiently minimize this function, the correlation surface is first computed over a range of displacements, thus avoiding the cost of recomputing the correlation during the minimization process. The range of displacements is kept small by use of a coarse-to-fine flow-through approach, so that at each level in the pyramid, we need only estimate motions of one pixel or less. Sub-pixel motion estimates are computed by interpolating the error surface with *bi-cubic splines* [Press *et al.*, 1988].

To estimate motions of one pixel or less using bi–cubic spline interpolation requires computing the correlation for displacements within a 5×5 pixel search area centered about zero displacement; this produces a 5×5 correlation surface. First a spline is fit to each row in the correlation surface, which requires that the first and second derivatives of the surface, along the row, be computed. These values can be stored. Then computing the value of the surface at any sub-pixel displacement involves first computing the interpolated value for each row, then fitting a spline to the new sub-pixel column.

6.2 Markov Random Fields

To minimize the above objective function, it is convenient to pose the problem as a Markov random field (MRF). MRF's have been used extensively in computer vision particularly for modeling texture [Derin and Elliott, 1987] and in the restoration of images [Geman and Reynolds, 1992; Geman and Geman, 1984]. More recently they have been used in recovering dense disparity maps from stereo images [Barnard, 1989] and optical flow from image sequences [Black and Anandan, 1991b; Black and Anandan, 1990b; Konrad, 1989; Konrad and Dubois, 1988; Murray and Buxton, 1987; Tian and Shah, 1992].

The MRF representation derives its popularity from its ability to model the expected spatial properties of image data in a Bayesian framework with a prior distribution. This ability to model expected spatial dependencies allows MRF's to be used in segmentation [Chou and Brown, 1990; Cohen and Nguyen, 1988; Derin and Elliott, 1987; Dubes *et al.*, 1990; Geman *et al.*, 1990], classification [Szeliski, 1988], and surface reconstruction [Geiger and Girosi, 1991; Marroquin *et al.*, 1987; Szeliski, 1988].

We will briefly review the foundations of MRF's and then recast the robust optical flow problem in this framework. Let $X = \{X_s \mid s \in S\}$ be a set of random variables indexed by the sites in the graph. The *state space*, Λ_s, of a variable X_s defines the possible values that X_s can take on and is denoted as $\Lambda_s \subset \mathbb{R}$, where $X_s \in \Lambda_s$.[1] The *configuration space*, Ω, defines the set of all possible configurations of the variables in the graph:

$$\Omega = \{\omega = (x_{s_1}, \ldots, x_{s_{n^2}}) \mid x_{s_i} \in \Lambda\}.$$

We say that X is a MRF with respect to a neighborhood system $\mathcal{G}$, as defined in Chapter 2, if:

$$P(X = \omega) > 0, \quad \text{for all } \omega \in \Omega, \tag{6.7}$$

$$P(X_s = x_s \mid X_t = x_t, s \neq t) = P(X_s = x_s \mid X_t = x_t, t \in \mathcal{G}_s) \tag{6.8}$$

[1] In the case of optical flow the X_s are random *vectors* and $\Lambda_s \subset \mathbb{R} \times \mathbb{R}$. This extension is straightforward.

for all $s \in S$ and $(x_{s_1}, \ldots, x_{s_{n^2}}) \in \Omega$. The power of this concept is that any joint probability $P(X = \omega)$ satisfying (6.7) is *uniquely determined* by the *local characteristics* $P(X_s = x_s \mid X_t = x_t, s \neq t)$ [Geman and Geman, 1984]. For vision applications, like surface reconstruction, the prior models used involve small neighborhoods. Thus the probability of a site having a particular value depends only on its neighbors' values. For small neighborhoods, efficient computational methods can be devised.

6.2.1 Gibbs Distributions

It is convenient to specify prior expectations about the data as an energy function E defined over small neighborhoods:

$$E(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega). \tag{6.9}$$

V_C depends only on those sites $s \in C$ and represents the *potentials* contributed to E by the sites in the cliques $C \in \mathcal{C}$. In the case of our optical flow problem, E_D and E_T have trivial neighborhoods which are just a single site, and the prior smoothness term E_S has the first-order neighborhood defined in Chapter 2.

The energy function, E, can be converted into a probability measure Π on Ω using the *Gibbs distribution*:

$$\Pi(\omega) = Z^{-1} e^{-E(\omega)/T}, \tag{6.10}$$

where Z, called the *partition function*, is the normalizing term:

$$Z = \sum_{\omega} e^{-E(\omega)/T}, \tag{6.11}$$

and where T is a *temperature* constant that serves to sharpen (or flatten) the distribution. When T is made large, Π becomes flat, but low values of T sharpen the mode(s) of Π.

This Gibbs distribution is a powerful modeling tool, but before it becomes useful to us, we need the following theorem due to Hammersley and Clifford [Geman and Geman, 1984]:

Theorem 1 *A random variable X is a Markov random field with respect to a neighborhood system $\mathcal{G}$ if and only if $\Pi(\omega) = P(X = \omega)$ is a Gibbs distribution with respect to $\mathcal{G}$.*

This theorem, stating the equivalence of MRF's and Gibbs distributions, means that if a problem can be defined in terms of local potentials then there is a simple way of formulating the problem in terms of MRF's. There is still a problem however. Notice that the partition function Z is defined by summing over *all* possible configurations $\omega \in \Omega$. This renders Z intractable for any problem of interesting size.

This is easily fixed by exploiting the equivalence theorem. If $\Pi(\omega) = P(X = \omega)$ is a Gibbs distribution then:

$$P(X_s = x_s \mid X_t = x_t, s \neq t) = Z_s^{-1} \exp -\frac{1}{T} \sum_{c \in C} V_c(\omega), \tag{6.12}$$

$$Z_s = \sum_{x \in \Lambda} \exp -\frac{1}{T} \sum_{c \in C} V_c(\omega^x). \tag{6.13}$$

where ω^x denotes the configuration which agrees with x everywhere except possibly at site s. Since these equations depend only on the local neighbors of a site, they are a practical way of specifying MRF's.

6.2.2 Optical Flow

Until now we have only looked at formalizing the prior model. Vision problems, however, typically involve *inverse problems*; for example, finding the "best" flow estimate **u** given the image data I, temporal information $\mathbf{u}^-$, and prior smoothness model.

In the Bayesian approach this corresponds to maximizing the *posterior distribution* $P(\mathbf{u} \mid I, \mathbf{u}^-)$. This is referred to as *maximum a posteriori*, or *MAP*, estimation. Applying Bayes' rule, and making the appropriate independence assumptions, gives:

$$P(\mathbf{u} \mid I, \mathbf{u}^-) = \frac{P(I \mid \mathbf{u}) P(\mathbf{u}^- \mid \mathbf{u}) P(\mathbf{u})}{P(I) P(\mathbf{u}^-)}. \tag{6.14}$$

Notice that $P(I)$ and $P(\mathbf{u}^-)$ are constant and do not involve **u** and, hence, are not of interest. Also, $P(\mathbf{u})$ is the prior smoothness model defined earlier:

$$P(\mathbf{u}) = Z_S^{-1} e^{-\beta_S E_S(\mathbf{u})/T}. \tag{6.15}$$

The other two terms are simply:

$$\begin{aligned} P(I \mid \mathbf{u}) &= Z_D^{-1} e^{-\beta_D E_D(\mathbf{u})/T}, && (6.16) \\ P(\mathbf{u}^- \mid \mathbf{u}) &= Z_T^{-1} e^{-\beta_T E_T(\mathbf{u},\mathbf{u}^-)/T}. && (6.17) \end{aligned}$$

Rewriting (6.14) in terms of the new posterior energy function combining the data, temporal, and smoothness terms gives:

$$P(\mathbf{u} \mid I, \mathbf{u}^-) = \frac{1}{Z} e^{-E(\mathbf{u},\mathbf{u}^-)}, \tag{6.18}$$

where E is just the objective function from above:

$$E(\mathbf{u}, \mathbf{u}^-) = \beta_D E_D(\mathbf{u}) + \beta_S E_S(\mathbf{u}) + \beta_T E_T(\mathbf{u}, \mathbf{u}^-),$$

which, as we have mentioned, typically has many local minima making the task of minimizing it difficult.

Our goal is to recover the MAP estimate, or equivalently, the minimum of E. The following section describes stochastic sampling techniques for actually computing this estimate.

6.3 Stochastic Minimization

When a problem is specified in terms of an energy function $E(\omega)$ as described above, the goal is to find the configuration ω for which E has the minimum energy. In analogy to physics, we are looking for the *ground state* of the system. For interesting problems, E will have many locally minimum energy states. This makes finding the ground state impossible

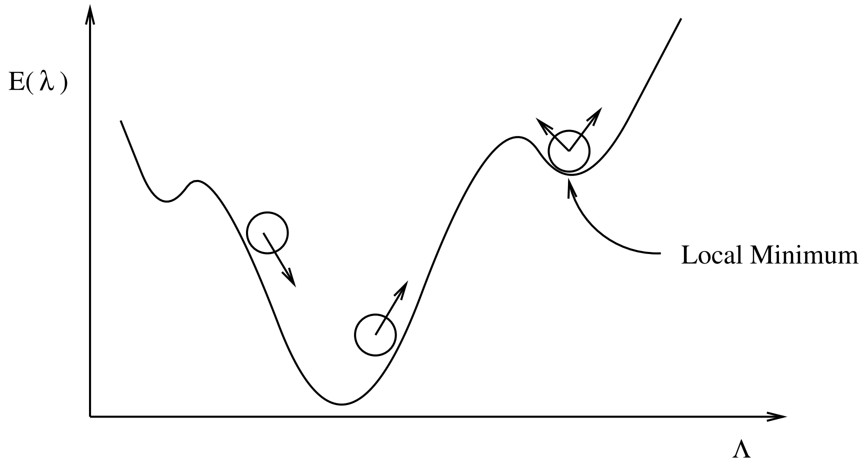


Figure 6.1: Minimizing a non-convex objective function. Gradient-descent techniques may become trapped in local minima.

with simple gradient based search techniques. Such techniques will typically get *stuck* in a local minima (Figure 6.1).

In real physical systems, the ground state can be reached through a slow, controlled, cooling of the system. By starting at high temperatures, where atoms are moving freely, and by slowly cooling, or *annealing*, the system gradually reaches a stable global minimum energy state without getting trapped in locally stable, but non-optimal, states.

6.3.1 Metropolis Algorithm

The Metropolis algorithm [Metropolis *et al.*, 1953] is a standard stochastic technique for finding a configuration, $\mathbf{x} = (x_1, \ldots, x_n)$, that minimizes an energy function $E(\mathbf{x})$. Starting with a random configuration $\mathbf{x}_0$, at each step of the algorithm we choose a small perturbation $\Delta\mathbf{x}$ of $\mathbf{x}$. This is chosen randomly, subject to constraints on acceptable state transitions.

We now compute the change in the system energy caused by the perturbation:

$$\Delta E = E(\mathbf{x} + \Delta\mathbf{x}) - E(\mathbf{x}). \tag{6.19}$$

If $\Delta E \leq 0$ then the perturbation is beneficial and should be *accepted*. If this is the case, then the state is updated and the process begins again.

On the other hand, if $\Delta E > 0$ then the move causes the energy to *increase*. In this case, the new configuration is accepted with probability given by the Boltzman distribution:

$$P(\Delta E) = e^{-(\Delta E/k_B T)}, \tag{6.20}$$

where k_B is the Boltzman constant and T is a temperature constant. This can be implemented by generating a random number in the interval $[0, 1]$ and testing it against $P(\Delta E)$. If it is less than $P(\Delta E)$ then accept the configuration, else reject it and reuse the old configuration. The process is then repeated.

In the case of simulated annealing, the Metropolis algorithm is used with an initial high temperature and then increasingly cooler temperatures. At high temperatures $P(\Delta E)$ approaches unity and hence nearly all configurations are accepted. The temperature is then gradually lowered in stages and the system is allowed to reach equilibrium. The procedure stops when a ground state is reached and no more changes occur.

6.3.2 Gibbs Sampler

Geman and Geman [1984] describe a related stochastic minimization technique called the *Gibbs sampler* which is more suited to our optimization problem. Suppose there is a processor at each site $s \in S$ with the communication between processors determined by the neighborhoods $\mathcal{G}_s$. Also, at each discrete time step t, the state of the processor is the random variable $X_s(t) \in \Lambda_s$. The value of $X_s(t)$ is to be updated while the values of the neighboring processors $r \in \mathcal{G}_s$ are kept fixed.

Given the energy function E construct the Gibbs distribution Π in the usual way. The processor then chooses a new state value of $x \in \Lambda_s$ by sampling from the local characteristics of Π. In other words:

> *Choose a state value $x \in \Lambda$ according to the distribution Π given the values of the neighbors.*

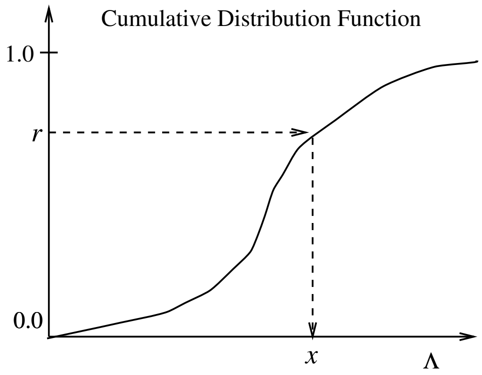


Figure 6.2: Gibbs Sampler; Monte Carlo sampling.

This sampling is performed using a Monte Carlo technique. Compute the cumulative density function of $X_s(t)$ using the local characteristics of the probability density function Π. Then generate a random number, r, in the interval $[0, 1]$. Using the cumulative density function, find the value of $x \in \Lambda_s$ that corresponds to the random number (see Figure 6.2 for an illustration). This process can be repeated for each site using a fixed or random *site visitation strategy*.

The annealing process is best illustrated by an example. Imagine that, at a site s, the energy can be characterized by the surface shown in Figure 6.3. First we construct the local characteristic of Π with a high temperature; say $T = 0.1$ (see Figure 6.4). Notice that the density function is nearly flat; this means that the cumulative distribution will be nearly linear which will mean that the sampler will choose $x \in \Lambda$ randomly. As the temperature is lowered, the modes of the distribution become more pronounced. At low temperature, $T = 0.005$, a single peak stands out with probability approaching unity. As the temperature is lowered this peak will be chosen by the sampler with increasing probability. At intermediate temperatures, there are multiple peaks. To avoid getting stuck in one of these false minima, the Gibbs Sampler must follow a particular *annealing schedule*.

An annealing schedule specifies the rate at which the temperature is lowered. To guarantee convergence the temperature must be lowered very slowly, particularly near the "freez-

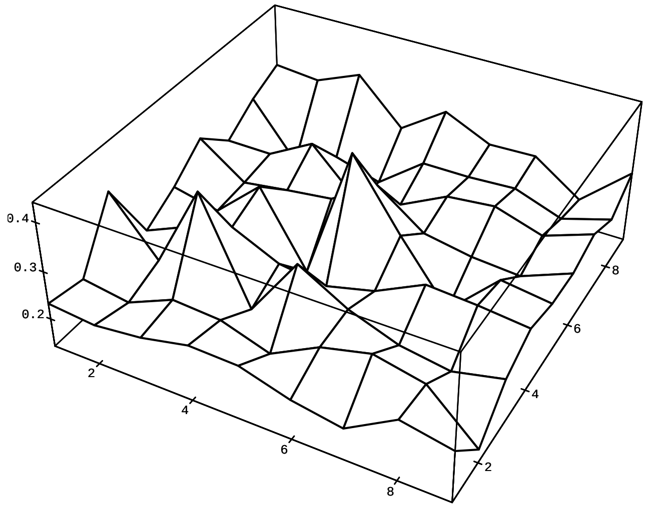


Figure 6.3: Initial error surface (inverted for display).

ing point" of the system. For convergence as the time $t \to \infty$, regardless of the starting configuration of the system, a number of conditions must be met:

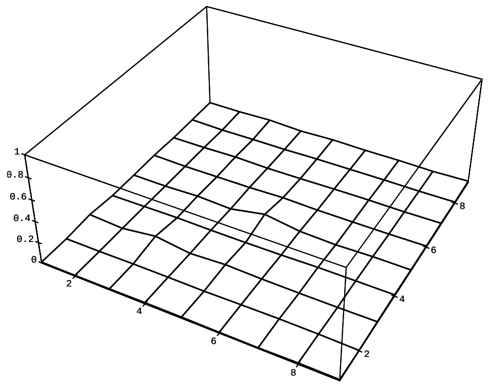
$$T(t) \to 0 \ \text{ as } \ t \to \infty, \tag{6.21}$$

$$T(t) \geq \frac{N\Delta}{\log t} \ \text{ for all } \ t \geq t_0 \ \ (t \geq 2). \tag{6.22}$$

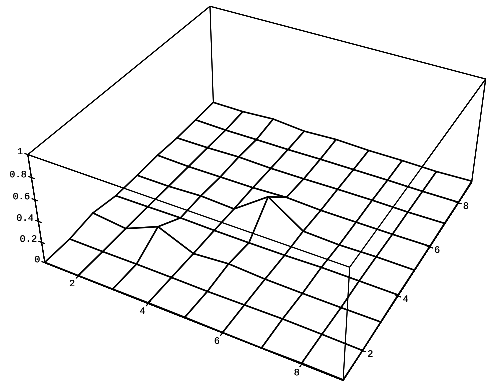
Additionally, each site must be visited infinitely often. Here, N is the number of sites in the MRF and Δ is the difference between maximum and minimum energy states.

This logarithmic schedule is impractically slow given current hardware. In practice, the initial temperature is chosen empirically and a more rapid, exponential or linear, cooling schedule is chosen. For this increased efficiency, one gives up the global convergence results obtained with the logarithmic schedule.

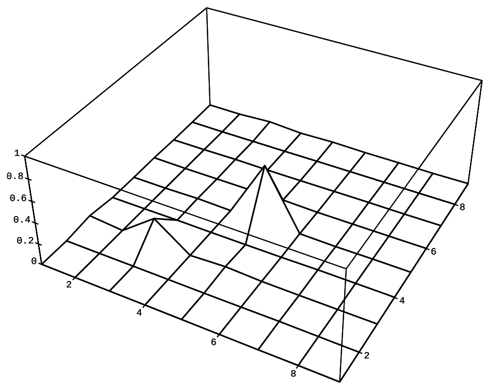
The single-site replacement algorithm above can easily be parallelized in the same way that the robust gradient method was parallelized. By dividing the sites into a checkerboard of black and white sites, half the sites can be updated simultaneously while the other half are held fixed.



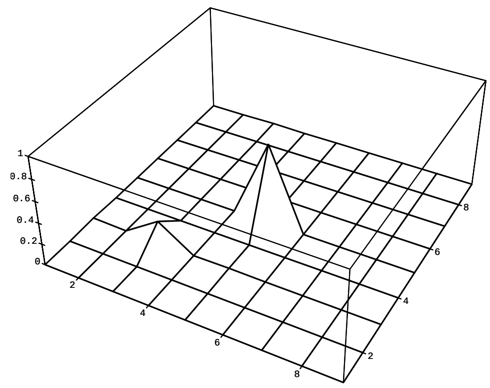
$T = 0.1$



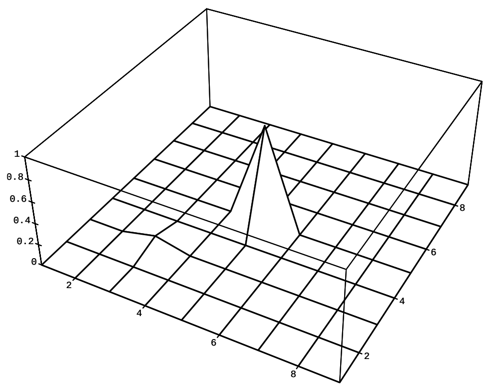
$T = 0.05$



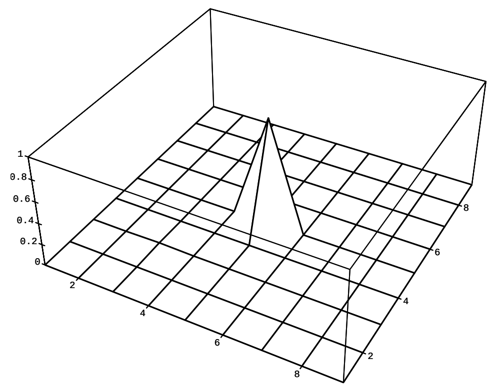
$T = 0.03$



$T = 0.02$



$T = 0.01$



$T = 0.005$

Figure 6.4: Example of annealing.

6.3.3 Continuous Annealing

In the case of optical flow, the state space Λ of all possible flow vectors is infinite, but, as defined, the Gibbs sampler requires a finite state space at any given time in order to effectively compute the partition function. The idea that allows us to solve continuous problems is that the state space can vary over time depending on the local properties of the function being minimized. At a given time t, we have an estimate of the motion $\mathbf{u}_t$, and consider making small changes $\Delta\mathbf{u}_t$ to the estimate. Vanderbilt and Louie [1984] define an extension to the standard Metropolis Monte Carlo technique which is *adaptive* in that the state space (defined by the step size, $\Delta\mathbf{u}_t$) automatically adapts to the local shape of the function being minimized. Here we extend their results to the Gibbs sampler.

The basic idea is to use the covariance matrix of a random walk to characterize the shape of the function. We set the state space so that it best explores the function by making the covariance matrix of the state space proportional to the covariance matrix of the random walk. Intuitively, if the variance along a particular search direction is large, then we want to increase the step size in that direction to get a coarse view of the function. When the true minimum has been chosen at a coarse level, the variance will shrink. To explore the minimum more finely, the area covered by the state space should shrink resulting in smaller step sizes. The basic idea is illustrated in Figure 6.5.

At a given site and at a given time, the state space Λ is always a discrete 3×3 neighborhood of the current estimate, but the area covered by the neighborhood varies based on the current step size $\Delta\mathbf{u}_t = [\Delta u_t, \Delta v_t]$. Given a current estimate $\mathbf{u}_t = [u_t, v_t]$, at time t the state space Λ is defined as:

$$\Lambda = \{\mathbf{u} + \Delta\mathbf{u} \mid \Delta\mathbf{u} = \mathbf{Q} \cdot \mathbf{l},\ \mathbf{l} = [l_1, l_2]^T,\ l_1, l_2 \in \{-(3/2)^{\frac{1}{2}}, 0, (3/2)^{\frac{1}{2}}\}\}, \tag{6.23}$$

where $\mathbf{Q}$ is a 2×2 matrix which controls the step size. Elements of the state space are all examined with equal probability, so the choice of trial steps is governed by a uniform probability distribution $\mathbf{g}(\mathbf{l})$ which, over $\{-(3/2)^{\frac{1}{2}}, 0, (3/2)^{\frac{1}{2}}\}$, has zero mean and unit variance.

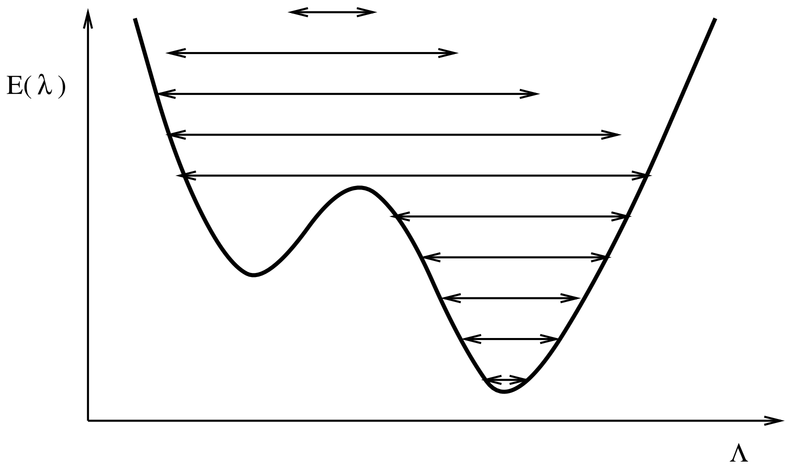


Figure 6.5: This figure [Vanderbilt and Louie, 1984] illustrates the adaptive nature of the continuous annealing process. The state space varies over time to best explore the function being minimized.

Since the mean of Λ is **u**, the covariance matrix **s**, of the state space is simply:

$$s_{ij} = \sum_{\Delta \mathbf{u} \in \Lambda} \Delta \mathbf{u}_i \Delta \mathbf{u}_j \mathbf{g}(\mathbf{l}). \tag{6.24}$$

Vanderbilt and Louie [1984] note that this can be expressed as:

$$\mathbf{s} = \mathbf{Q} \cdot \mathbf{Q}^T. \tag{6.25}$$

Hence we can generate a state space with any desired covariance matrix **s** by solving for **Q** using Cholesky decomposition [Strang, 1976] and then using **Q** to generate the state space in equation 6.23.

As mentioned, the covariance matrix of the actual steps that would be taken in a random walk can be used as a measure of the local shape of the function. We want the covariance matrix of the state space to be proportional to this covariance matrix of a random walk. The actual step taken at a time t is determined by the probability distribution $\Pi(\mathbf{u}_t + \Delta \mathbf{u}_t)$ defined over the space of displacements. Using Π we can compute the mean μ at time t (note we drop t when it is constant across all terms):

$$\mu_i = \sum_{\mathbf{u} \in \Lambda} \Pi(\mathbf{u})\mathbf{u}_i. \tag{6.26}$$

The covariance matrix **S** of Π given the current step size is:

$$S_{ij} = \sum_{\mathbf{u} \in \Lambda} (\mathbf{u}_i - \mu_i)(\mathbf{u}_j - \mu_j)\Pi(\mathbf{u}). \tag{6.27}$$

We make the covariance matrix of the state space at time $t+1$ proportional to $\mathbf{S}^{(t)}$:

$$\mathbf{s}^{(t+1)} = \chi \mathbf{S}^{(t)}, \tag{6.28}$$

where χ is a scaling factor. Now solving $s^{(t+1)} = \mathbf{Q} \cdot \mathbf{Q}^T$ for **Q** gives the **Q** for determining the state space at the next time instant. Performing Cholesky decomposition involves using Gaussian elimination to factor **S** into $LDL^T = L\sqrt{D}\sqrt{D}L^T$, where L is lower triangular and D is diagonal. For the simple 2×2 case of optical flow we have:

$$\mathbf{S} = \begin{bmatrix} S_{00} & S_{01} \\ S_{01} & S_{11} \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \begin{bmatrix} \sqrt{S_{00}} & 0 \\ S_{10}/\sqrt{S_{00}} & \sqrt{S_{11} - S_{10}^2/S_{00}} \end{bmatrix}. \tag{6.29}$$

We now need to choose the scale factor for χ. Assume a step size $\Delta\mathbf{u}$ and imagine the case in which Π is uniform so $\mathbf{s}^{(t)} = \mathbf{S}^{(t)}$. No information is being gained with the current step size so we should increase it. If $\chi > 1$ then the step size will be increased by a factor of $\sqrt{\chi}$ on the next iteration. Over time, as the the algorithm settles into the true minimum, the variance will decrease. The result will be decreasing step sizes which allow the minimum to be explored more precisely. In all our experiments we take $\chi = 3$ as suggested by Vanderbilt and Louie.

To prevent the state space from growing or shrinking too rapidly, we control the rate at which new information from **S** overwrites the previous information:

$$\mathbf{s}^{(t+1)} = \alpha\chi\mathbf{S}^{(t)} + (1-\alpha)\mathbf{s}^{(t)},$$

where α can be viewed as a damping factor. In our experiments $\alpha = 0.5$. Additionally, to prevent the state space from going to zero, or growing larger than the maximum expected motion, we set a lower and upper bound on the state space.

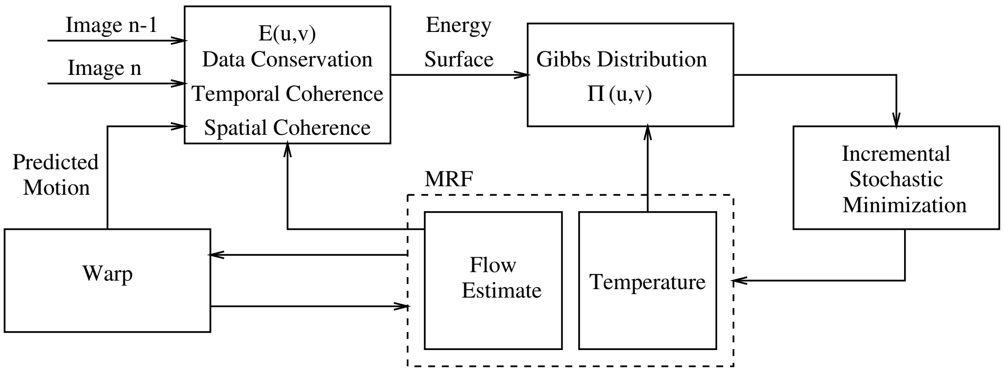


Figure 6.6: Incremental Stochastic Minimization (see text).

6.4 Incremental Stochastic Minimization

The obvious disadvantage of using simulated annealing is that its computational expense is prohibitive. So, we pose the stochastic minimization problem in the incremental framework with the continuous Gibbs sampler as the minimization procedure and the temperature as the control parameter. The algorithm is summarized in the block diagram in Figure 6.6.

When a new image is acquired, the current motion estimate at a given site (representing a particular surface patch) is used as the starting point for the continuous annealing algorithm and to refine the predicted motion used in the temporal coherence constraint. The current temperature at each site is used as the initial temperature, which is then lowered according to the annealing schedule.

After a fixed (usually small) number of iterations of the annealing process, each site has a new motion estimate and temperature. The various properties of the associated surface are then propagated to the new site where the patch has moved. The flow, temperature, and adaptive state space information are warped along with the sites of the MRF by the flow estimate. The propagation algorithm described below also detects occlusion and disocclusion boundaries.

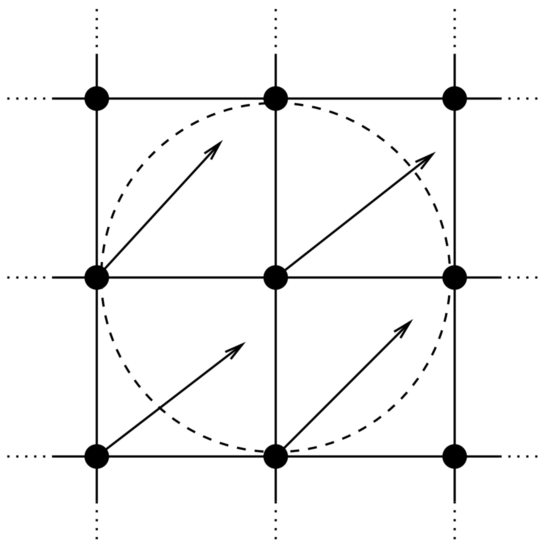


Figure 6.7: Forward Warping

6.4.1 Prediction (Warping)

We use a forwards warping strategy as illustrated in Figure 6.7 in which each level of the pyramid is warped simultaneously. Using a coarse-to-fine flow-through approach we can assume that, at any level, all motions are less than a pixel, since larger motions will be dealt with by combining information across levels. Each site s determines which of its neighbors are moving towards it. This is done by examining its own motion and the motion of its eight immediate neighbors to identify those sites whose new location is estimated to be within a pixel of the site s. Let this set of neighbors be denoted as $\eta(s)$. New estimates of image properties at each site are obtained by a weighted sum of the properties stored at the sites belonging to this refined neighborhood. Examples of properties belonging to a site are its motion, temperature, and state space. Additional properties like image intensity or higher level information about surface membership may also be present.

This propagation can be viewed as a forward warping of the sites according to the motion estimate [Black and Anandan, 1991b]. Since the motion is not discrete, the field is resampled using a weighted interpolation. When doing this resampling, it is necessary to resolve conflicts that arise in the forward warp. This is done by weighting the motion estimates by Π.

Let ρ be a property of s. Then the new estimate of $\rho(s)$ is given by:

$$\rho(s) = \frac{1}{w(s)} \sum_{t \in \eta(s)} \Pi(\mathbf{u}(t))(1 - d(s,t))\rho(t) \tag{6.30}$$

$$w(s) = \sum_{t \in \eta(s)} \Pi(\mathbf{u}(t))(1 - d(s,t)), \tag{6.31}$$

where w is a normalizing term, and $d(s,t)$ is the distance between the projection of site t and the location of site s.

6.4.2 Occlusion and Disocclusion

The propagation algorithm outlined above can be made sensitive to the presence of occlusion and disocclusion around each site. To understand how this is done, observe that the normalizing factor w roughly measures the total flow into a site. In the absence of motion discontinuities this should be approximately unity. However, if occlusions are present within the neighborhood of a site, we may expect multiple sites to move towards it, thereby increasing the total in-flow. Similarly, if there is a disocclusion, we may expect the total flow to be less than unity.

The current version of our algorithm includes a simple implementation of the idea described above for occlusion/disocclusion detection. The net flow, which is measured by the quantity w, is estimated and compared against two thresholds, one above and one below unity, in order to categorize a site as occlusion, disocclusion, or single motion. This is obviously too simple to handle complex situations and may fail even in simple situations. For example, if there is significant divergence (or convergence) present within the neighborhood of a site, net flow will differ from unity, even if there are no motion discontinuities. Despite these shortcomings, the simple approach performs well in our experiments.

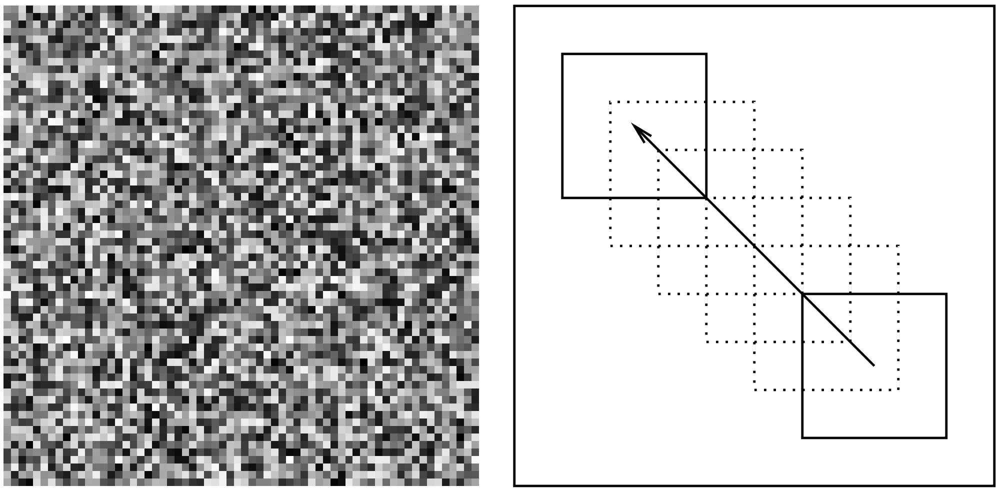
In the current algorithm no special processing is done at occlusion sites, other than to simply indicate them as such. A more sophisticated approach would involve modifying the propagation scheme to take contributions from processors which correspond to the occluding surface. If this information were available from higher level processes as a property of the site, it could easily be incorporated.

A disoccluded site indicates a new patch of the environment which was previously hidden from view. For this new patch, there is no prior motion estimate, hence the annealing process should be initially uncommitted about the motion. This is achieved by initializing the site to have a high temperature and initializing the motion estimate and state space. Note that even if false disocclusions are detected in areas which do not correspond to our motion model, increasing the temperatures may still be useful to extend the search space at that site. In many cases it is also desirable to reinitialize occluded sites since there is ambiguity as to the correct motion estimate near the occlusion boundary.

It should be clear that unlike standard annealing, our algorithm uses different temperatures for the different sites and dynamically modifies the temperature according to the information available at a site. As a patch is tracked, its temperature will decrease over time. Hence, the temperatures of patches that have been tracked over many frames, and whose motion is precisely known, tend to be lower than those of more recently disoccluded (i.e., new) patches.

Convergence

Unlike simulated annealing, our new algorithm, which uses the continuous annealing scheme, incremental minimization, and a linear cooling schedule, is not guaranteed to converge. Empirical results, however, indicate that the approach does, in practice, converge to near-optimal sub-pixel motion estimates. Obviously, the degree to which the constraints accurately reflect the physics of the world will affect both the convergence and the accuracy of the algorithm.



a *b*

Figure 6.8: Moving square sequence (30 images). *a)* Intensity image. *b)* 20×20 pixel patch translates one pixel up and to the left per frame. Notice that the stationary patch is not visible in the intensity image. When the patch moves its boundaries are distinctly visible.

6.5 Experiments

The incremental algorithm has been tested on real and synthetic image sequences. Experiments with controlled synthetic data illustrate the performance of the algorithm, while real image sequences, demonstrate the algorithm's ability to achieve qualitatively good motion estimates in the presence of noise. Without ground truth, no quantitative analysis of the real motion sequences is possible.

6.5.1 Synthetic Moving Square

The first example involves a synthetic image sequence with a 20×20 pixel textured square moving across a 64×64 pixel stationary textured background. The random noise texture of the foreground and background patches is uniformly distributed between 0 and 255 (Figure 6.8*a*). The sequence consists of thirty frames; in each frame the foreground patch moves one pixel up and to the left (Figure 6.8*b*). Uniform random noise over the range $[-\eta, \eta]$ was added to each image in the sequence. For the example in Figure 6.8, η is taken to be five percent of the intensity range; so $\eta = 12.75$.

The results of the motion algorithm applied to the sequence are shown in Figure 6.9. Only a single iteration of the temporal annealing algorithm was performed for each pair of images in the sequence. The following weights were used:

Discontinuities			Weights		
$\Delta_D = 5.0$	$\Delta_S = 0.5$	$\Delta_T = 0.5$	$\beta_D = 2.0$	$\beta_S = 1.0$	$\beta_T = 1.0$

A rapid, nearly linear, cooling schedule was employed with a starting temperature for each site s of $T_s = 0.4$. For this simple example, the state space was kept fixed, resulting in a discrete estimation problem.

Each row of images in the figure represents a snapshot of the algorithm at an instant in time. The first and second images in each row show the current estimate of the horizontal and vertical motion respectively. Dark areas correspond to a negative motion (left or up) and bright areas correspond to positive motion (right or down); stationary areas appear gray. The third image in each row shows the location of motion discontinuities. Occlusion boundaries are displayed as white, and disocclusion boundaries appear black.

Since the algorithm works on a sequence of images, results are always available, and the quality of the information increases with time. Row *a* shows the current motion and discontinuity estimates after only two frames. By the sixth frame (row *b*), the errors in the initial estimate are being corrected and the system is beginning to settle into a global interpretation. The results at the end of the thirty frame sequence are shown in row *c*. The motion estimates are accurate everywhere, with the exception of recently disoccluded areas, and the discontinuity estimates correspond well to the actual boundary of the moving patch. Recently disoccluded areas with high temperatures have not yet settled into a stable interpretation and are hence less accurate. Increasing the number of iterations per frame would permit these areas to settle more quickly to the correct interpretation, but would diminish the dynamic nature of the processing.

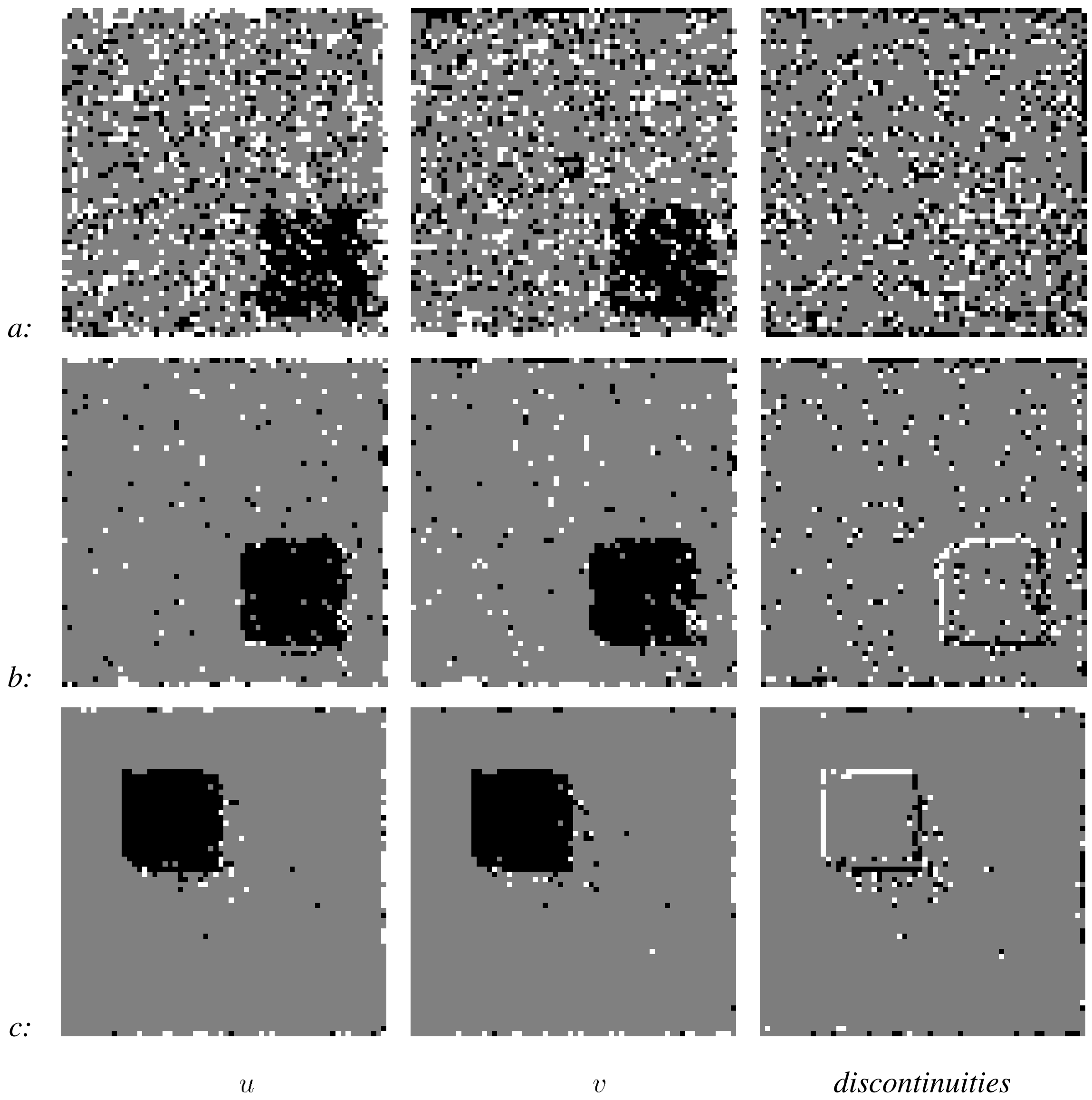


Figure 6.9: Random Dot Image Sequence. Each row shows the state of the model at an instant in time. *a)* state after two frames; *b)* after six frames; *c)* after 30 frames.

Figure 6.10: Temperature at each site at the end of the sequence.

Figure 6.10 shows the temperature at each site at the end of the image sequence. Lighter areas correspond to higher temperatures. The stationary background and the patch itself are dark, indicating that by the end of the sequence, the motion of these areas is known accurately. The brightest areas correspond to recently disoccluded portions of the background. As time progresses, the motion of these areas becomes known with more accuracy and the temperature decreases. This phenomenon can be observed in the figure as a fading "vapor trail" left by the moving square.

6.5.2 Convergence Experiments

While no theoretical proof of convergence exists, in practice the ISM algorithm converges to the correct solution even in the presence of noise. For this and all future experiments we will use the continuous annealing strategy. To illustrate the convergence properties of the algorithm a synthetic image sequence was generated. The sequence consists of a 64×64 pixel uniform random signal over the range $[0, 255]$ which is undergoing a uniform translation of one half pixel to the right and down per frame.

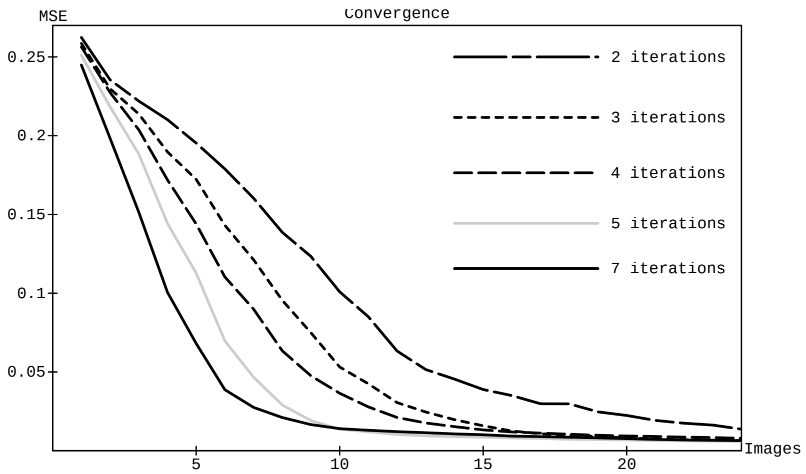


Figure 6.11: ISM Convergence Experiments. Mean Squared Error (MSE) (in pixels) as a function of the number of frames in a 25 image sequence. The results are plotted for tests involving varying numbers of iterations of the annealing algorithm per frame (from 2 to 7).

The initial experiments consider a noiseless signal and examine the convergence of the algorithm over time. Error is computed as the mean squared error (MSE) of the motion estimate in an $n \times n$ region $\mathcal{R}$ of the image which is visible for the entire sequence:

$$MSE = \frac{1}{n^2} \sum_{s \in \mathcal{R}} (\mathbf{u}_{true} - \mathbf{u}_s)^T (\mathbf{u}_{true} - \mathbf{u}_s).$$

Error in recently disoccluded regions, corresponding to the left and top edges of the image, will be higher than those regions which have been tracked over the length of the sequence. Figure 6.11 plots mean squared error of the motion estimate as a function of the number of images examined in the sequence. The error is plotted for trials using 2, 3, 4, 5 and 7 iterations per frame. Increasing the number of iterations per frame increases the rate of convergence but, even with only three iterations per frame, the algorithm converges to the correct solution within approximately 25 frames.

The next experiment addresses the effect of noise on the convergence of the algorithm. Uniform random noise over the range $[-\gamma/2, \gamma/2]$ was added to each image in the sequence, where γ is a percentage of the total intensity range. Figure 6.12 shows the effect of zero to

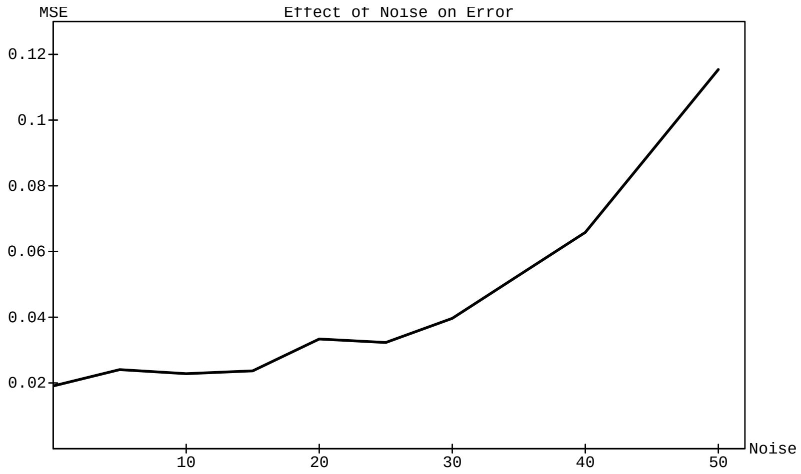


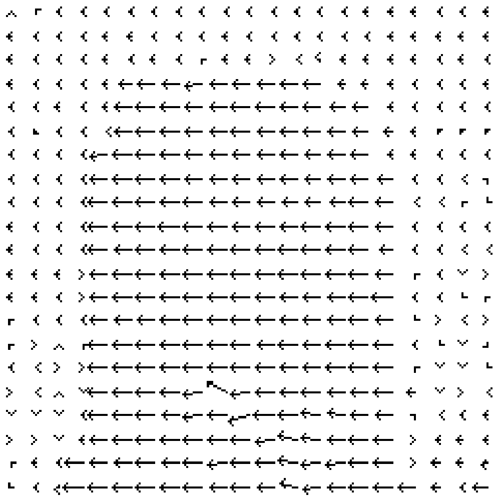
Figure 6.12: Noise Experiments. Mean squared error as a function of noise (from 0% to 50%) is plotted plotted for a 10 frame sequence with 5 iterations per frame.

50 percent noise on the error of the motion estimate. The experiment is performed with only a 10 frame image sequence with five iterations per frame. The results indicate the the algorithm is tolerant to fairly large amounts of noise (up to about 30%). Above that, longer sequences or more iterations per frame would be required to reach acceptable levels of error.

6.5.3 Sub-Pixel Motion and Discontinuities

The following experiment involves an image sequence consisting of eight 64×64 square images; the last image in the sequence is shown in Figure 6.13*a*. The images contain a soda can in the foreground; the motion of which is slightly less than one pixel to the left between each frame. The can is moving in front of a textured background that is also undergoing a slight motion to the left; there is no vertical motion. Since all the motion is less than a pixel, this sequence tests the sub-pixel accuracy of the algorithm independently of the multi-resolution strategy. The flow field, computed to sub-pixel accuracy, is shown in Figure 6.13*b*.

These results can be compared with those obtained using the hierarchical SSD algorithm of Anandan [1989] shown in Figures 6.14*a* and 6.14*b*. The horizontal and vertical compo-



a *b*

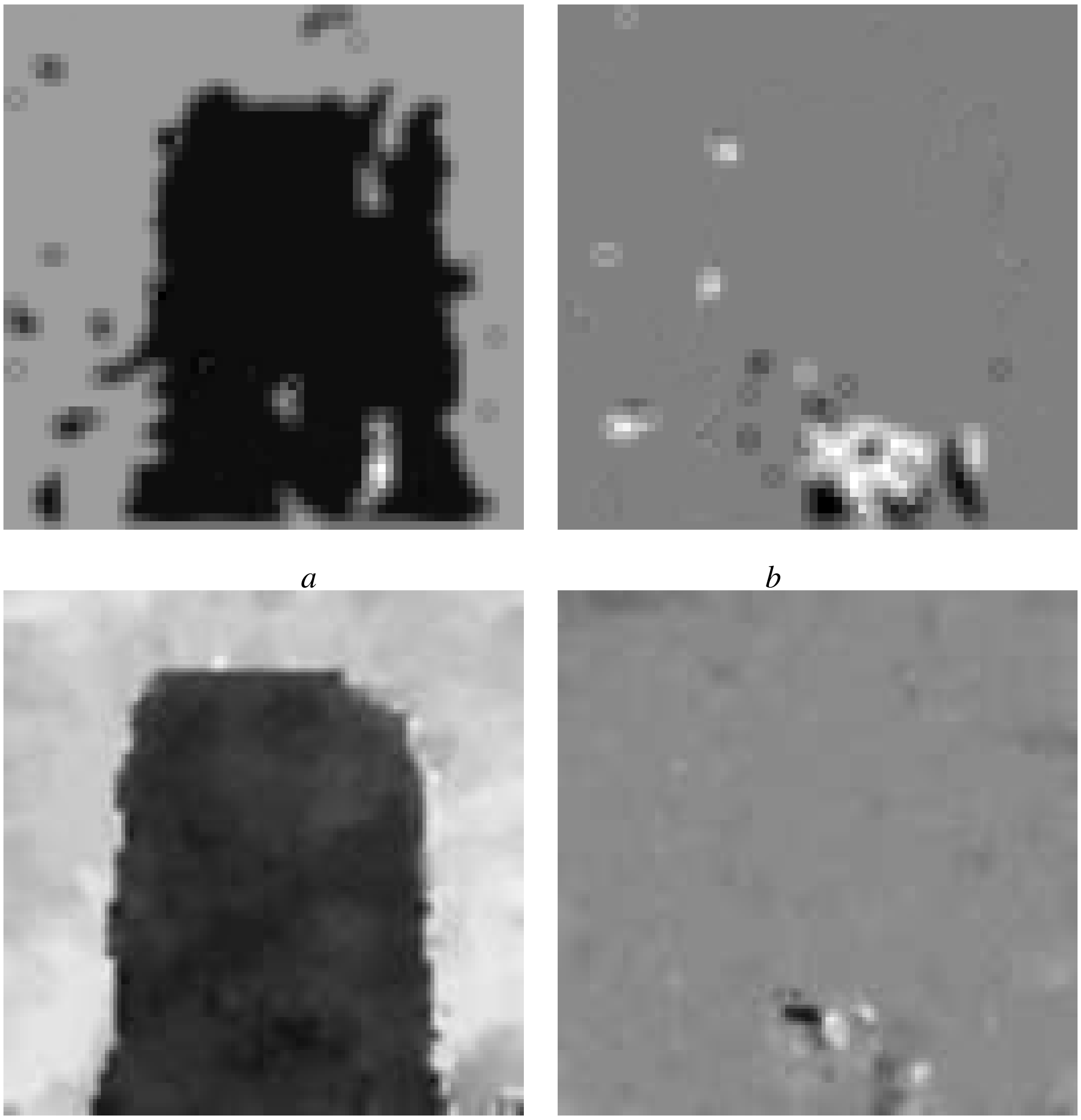
Figure 6.13: Pepsi can image sequence (results after eight frames): *a)* Intensity image; *b)* Flow field.

nents of the flow field computed by the ISM approach are shown in Figures 6.14*c* and 6.14*d* respectively. Notice that over-smoothing does not take place and flow discontinuities are maintained. Also notice that the errors in the vertical motion estimate correspond to areas of low image contrast. A longer image sequence or more iterations per frame would likely reduce the errors further. Overall, the performance is significantly improved over the two frame algorithm.

Occlusion and disocclusion boundary estimates are shown in Figure 6.15. The brighter the area, the more likely it is to be an occlusion boundary. Similarly, dark areas indicate disocclusion. It is important to remember that, while these results show only the final frames in the image sequence, both flow and discontinuity estimates are available at all times.

Figure 6.16 illustrates the adaptive state space used by the continuous annealing algorithm at the end of the image sequence. Recall, from equation (6.23), that the state space, at a site, s, is determined by:

$$\Lambda = \{\mathbf{u} + \Delta\mathbf{u} \mid \Delta\mathbf{u} = \mathbf{Q} \cdot \mathbf{l},\ \mathbf{l} = [l_1, l_2]^T,\ l_1, l_2 \in \{-(3/2)^{\frac{1}{2}}, 0, (3/2)^{\frac{1}{2}}\}\},$$



a b

c d

Figure 6.14: Pepsi can image sequence. *a, b)* Anandan's SSD algorithm, horizontal and vertical flow. *c, d)* ISM algorithm after eight frames, horizontal and vertical flow.

Figure 6.15: Pepsi can image sequence: Discontinuities

where,

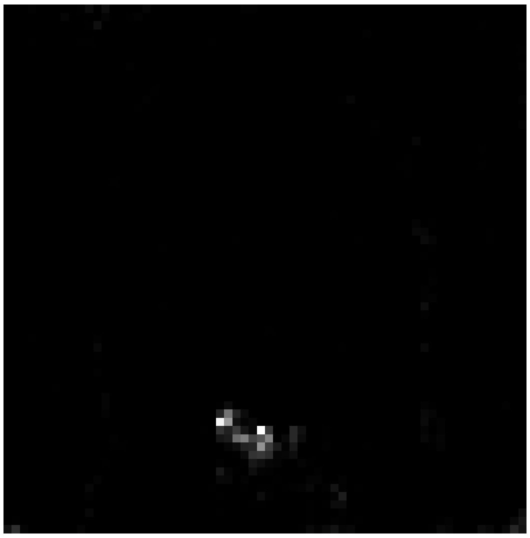
$$\mathbf{Q}_s = \begin{bmatrix} Q_{00,s} & 0 \\ Q_{01,s} & Q_{11,s} \end{bmatrix},$$

and where $Q_{00,s}$, $Q_{11,s}$, and $Q_{01,s}$ refer to the image pixel values in the respective images in Figure 6.16. Bright areas correspond to large values, dark to small. Figure 6.16*a* shows that the horizontal component of the state space. The area covered by the state space is largest at the can boundaries due to the local uncertainty in the motion estimate.

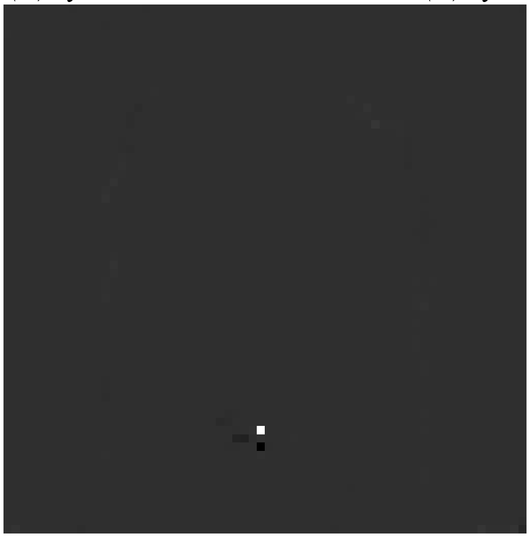
The values represented in the figure range from 0.05 to 0.82 which means that the state space for u ranges from 0.06 to 1.0 pixels. The values in Figure 6.16*b* show the vertical component of the state space and range from 0.05 to 0.74. The largest vertical uncertainty is in the homogeneous region of the can. The values for the off diagonal component shown in Figure 6.16c range from -0.02 to 0.09, where grey is zero, black is negative, and white is positive. Although it is difficult to see in images (b) and (c) due to a lack of contrast, the search area is also somewhat larger near the can boundary.

These figures illustrate a number of properties of the adaptive state space. First, across the motion boundary of the can, there is difference in the horizontal flow. For points within

(a) Q_{00}



(b) Q_{11}



(c) Q_{01}

Figure 6.16: Pepsi can image sequence: State space (see text).

the vicinity of the boundary, the shape of the objective function reflects the presense of multiple motions and contains multiple minima. The uncertainty in the horizontal motion at the boundary is then reflected in the increased size of the horizontal search direction. Since there is little uncertainty in the vertical direction, the search in that direction remains relatively small.

Second, the lower center portion of the can contains a nearly homogeneous region. The lack of texture in this region means that the objective function contains no clearly defined minimum, but rather is a relatively flat surface. This results in a high variance for a random walk and, hence, the state space grows to explore this region more effectively.

6.5.4 SRI Tree Sequence

We now consider a more complex sequence containing 63 images (numbered 32–95) of size 128×128 pixels. Figure 6.17 shows six images in the sequence. The translational motion of the camera is parallel to the image plane and the maximum pixel motion between frames is less than one pixel. This sequence offers more challenges than the Pepsi sequence in that it is much longer, there is significant image noise, and there are many closely spaced discontinuities (fragmented occlusion). Additionally, the image motion of the ground plane violates the assumption of piecewise constant flow underlying our formulation of spatial coherence. In the Pepsi sequence, on the other hand, the flow is essentially piecewise constant, and hence, a first order smoothness constraint is appropriate.

For this experiment, we used ten iterations of the annealing algorithm per frame with an initial temperature of $T = 0.5$ and a linear cooling schedule. The other parameters were as follows:

Discontinuities			Weights		
$\Delta_D = 5.0$	$\Delta_S = 0.25$	$\Delta_T = 0.1$	$\beta_D = 1.0$	$\beta_S = 3.0$	$\beta_T = 1.0$

A small, 3×3, region was used for computing the robust correlation.

33

43

53

63

73

83

Figure 6.17: SRI tree sequence (images); (numbers 33, 43, 53, 63, 73, 83).

Running on an 8K processor Connection Machine, the algorithm took three seconds to update every site using the continuous Gibbs Sampler. That meant that it took 30 seconds to process each new image, and approximately 30 minutes to process the entire sequence. Of that computation, approximately 55% was actually performed on the Connection Machine.

The results are summarized in Figure 6.18 which shows the horizontal component of the flow after every ten frames in the sequence. The estimate starts out coarse and noisy and becomes smoother and less noisy over time. The vertical motion is nearly zero everywhere and is shown in Figure 6.19.

For this experiment 128×128 smoothed and subsampled versions of the original SRI images were used due to Connection Machine memory limits which precluded using the full sized images. This had two effects on the results. First, all the motions were less than a pixel so a multi-resolution scheme was not required. Second, the branches of the tree in the subsampled sequence are very narrow, making precise recovery of the branch boundaries difficult. One can compare these results with those of the robust gradient algorithm in Chapter 4 which used the full sized images and achieved better resolution with respect to the motion of the branches and their boundaries.

The motion discontinuities are shown in Figure 6.20 and the temperature at an intermediate stage is shown in Figure 6.21. Notice that areas of high temperature correspond to motion discontinuities.

We mentioned above that the assumption of piecewise constant image motion is violated by the ground plane. In the results presented, only motions greater than $\Delta_S = 0.25$ pixels were considered violations of the spatial smoothness term. The effect of reducing this to $\Delta_S = 0.1$ pixels is demonstrated in Figure 6.22. In this case, the more strict smoothness term causes the ground plane to be split into two piecewise constant regions. What this example illustrates is the need for higher order models for recovering general motion. In particular, we might first recover the piecewise constant flow in Figure 6.22*b* and then apply

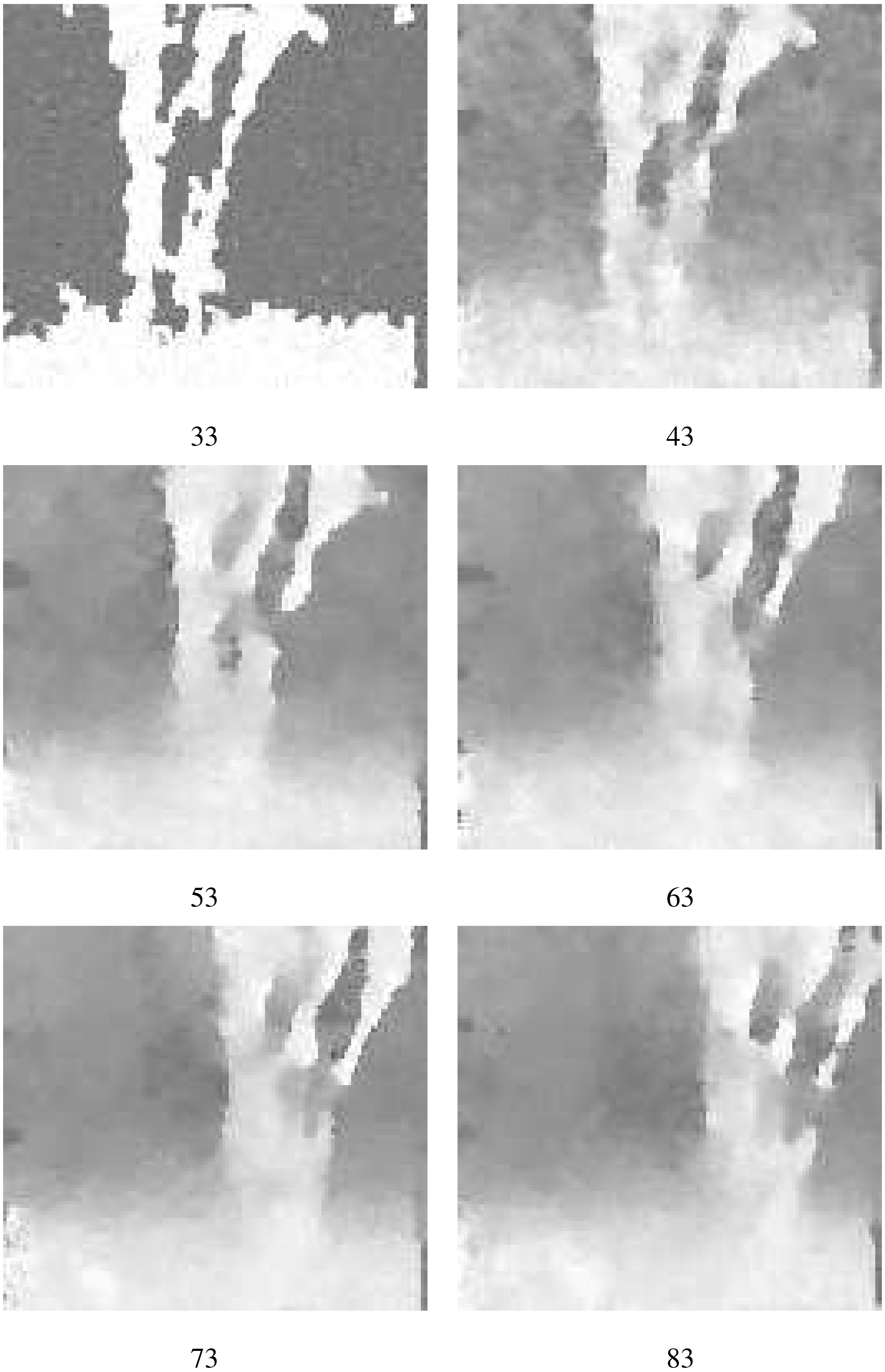


Figure 6.18: SRI tree sequence (horizontal flow); (numbers 33, 43, 53, 63, 73, 83).

Figure 6.19: SRI tree sequence (vertical flow.)

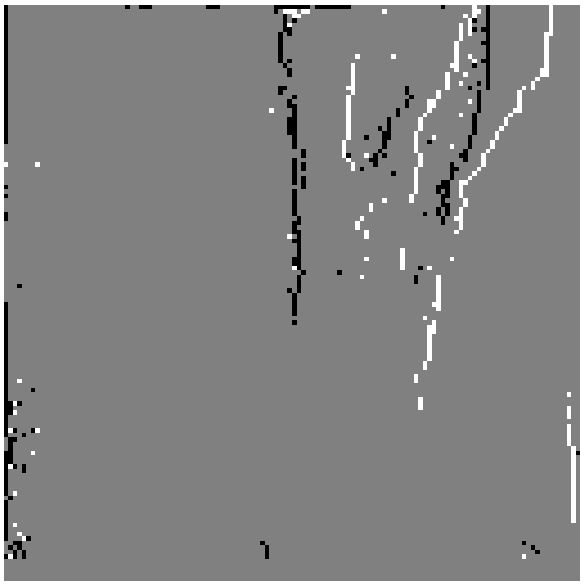


Figure 6.20: SRI tree sequence (thresholded discontinuities at image 73).

Figure 6.21: SRI tree sequence (temperature at image 53).

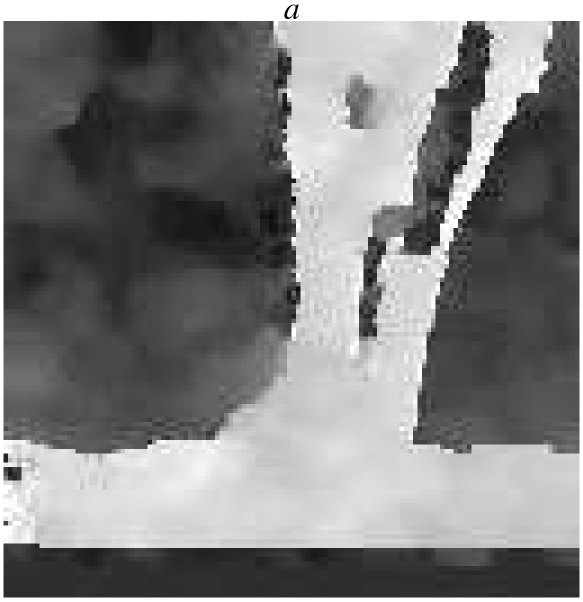
a second order smoothness model [Geman and Reynolds, 1992].

6.5.5 Nap-Of-the-Earth Experiment

The final experiment tests the full algorithm, including the multi-resolution flow-through strategy. The test sequence consists of 100 images of size 128×128 pixels. The images were acquired from a camera mounted on a helicopter in *Nap-Of-the-Earth* (NOE) flight. The sequence is challenging in many respects. First the range of motion in the images is wide; from 0 to approximately 4 pixels. To cope with motions of up to 4 pixels, a three level pyramid was used. Second, there are areas in the images of low contrast where good data estimates are not available. Finally, the motion is complex and changing; there is pitch, yaw and rotation in addition to translation. The actual motion is corrupted by jitter introduced by the camera mounting and turbulence.

Unfortunately, it is impossible to convey the dynamic behavior of the algorithm over the 100 image sequence in a static format for presentation here. Figures 6.23, 6.24, and 6.25 shows six snapshots of the processing after 15, 30, 45, 60, 75 and 90 frames. The data

a



b

Figure 6.22: Tree sequence: smoothness assumption violated. Figure *(a)* shows the horizontal flow at image 70 using the same parameter settings as in Figure 6.18 (in particular $\Delta_S = 0.25$). The horizontal flow in figure *(b)* was obtained with the same parameters except with $\Delta_S = 0.1$ (the same as was used in the Pepsi sequence). See text for an analysis.

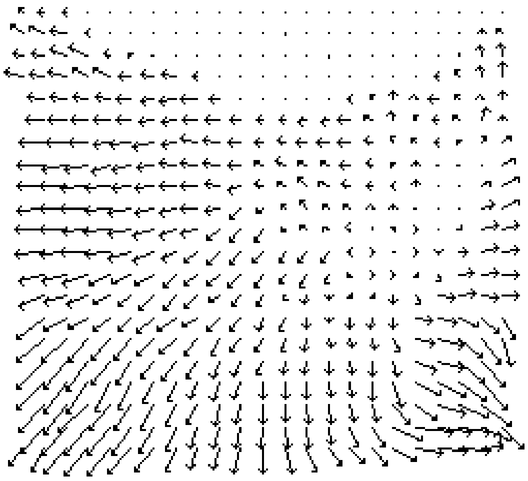


Image: 11185 *Flow: 11185*

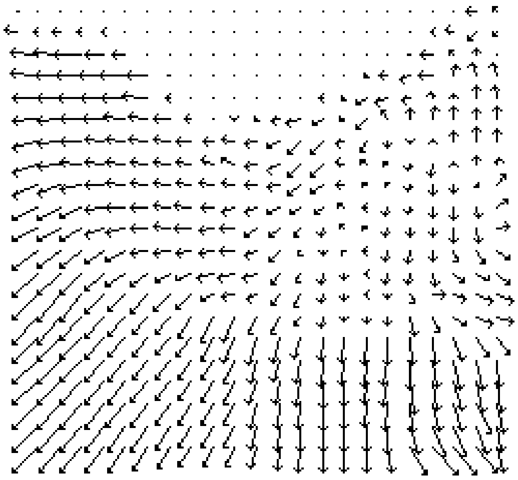
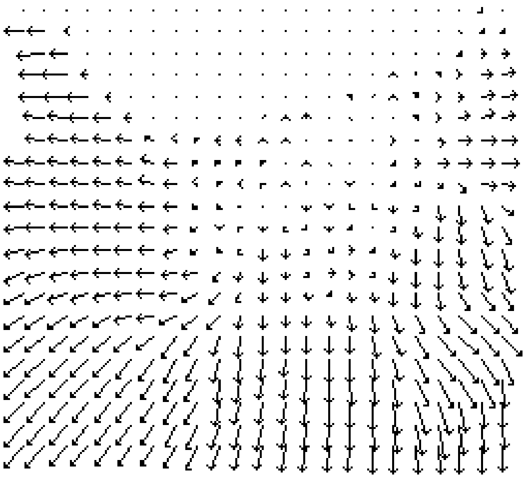


Image: 11200 *Flow: 11200*

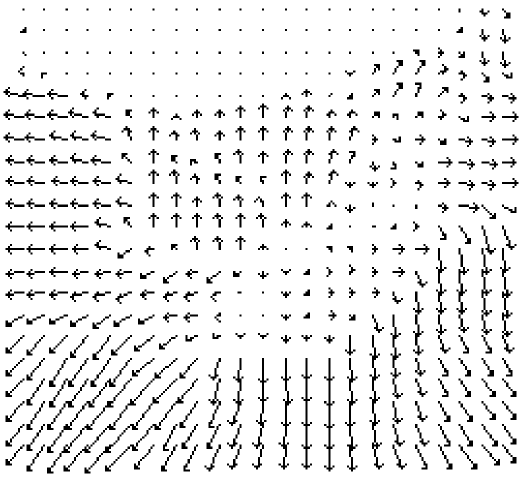
Figure 6.23: Nap-Of-the-Earth Helicopter Sequence. Snapshots of images and associated flow fields in a 100 image sequence.

Image: 11215



Flow: 11215

Image: 11230



Flow: 11230

Figure 6.24: Nap-Of-the-Earth Helicopter Sequence. Snapshots of images and associated flow fields in a 100 image sequence.

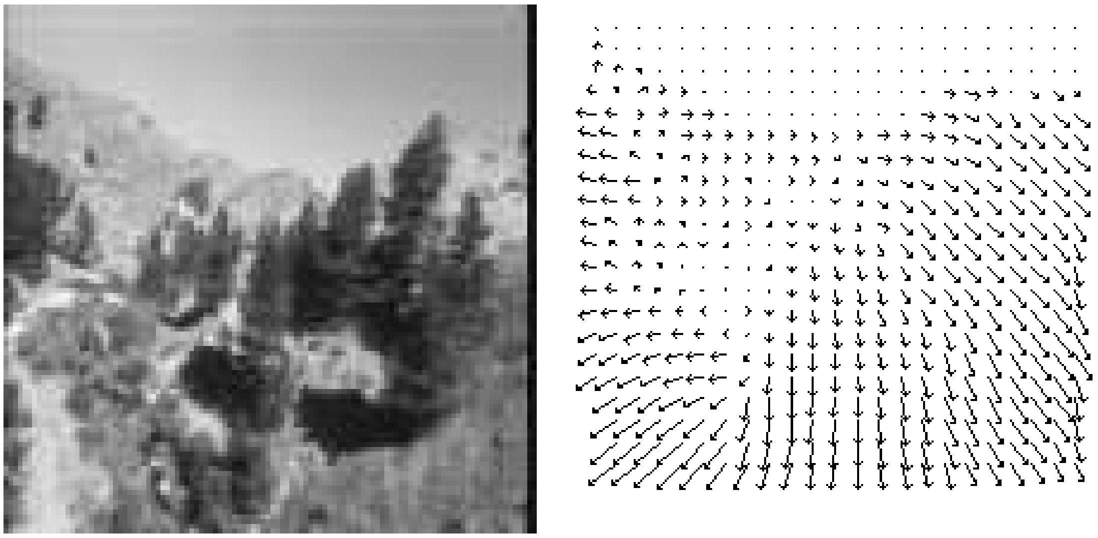


Image: 11245 *Flow: 11245*

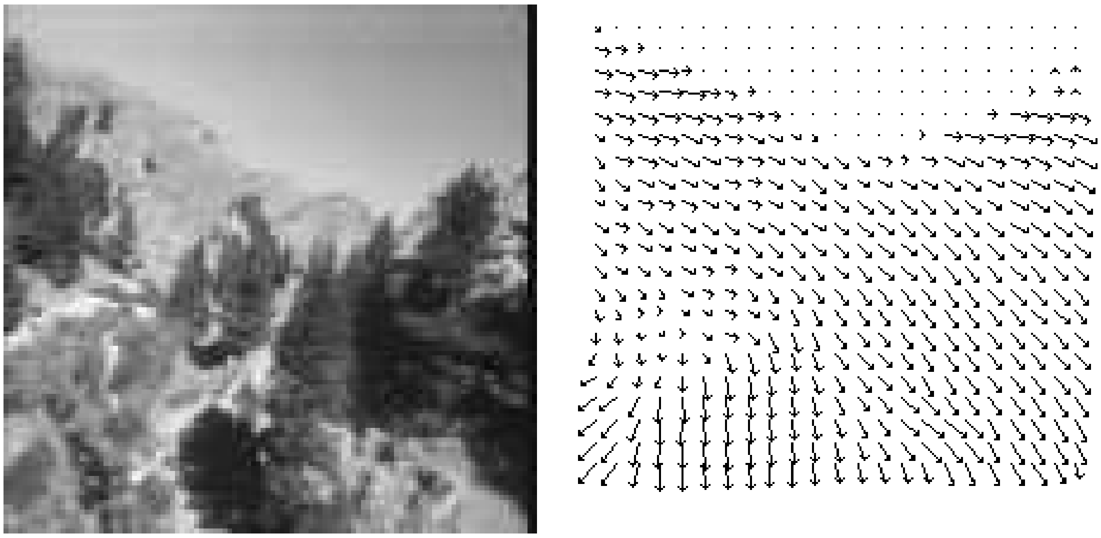


Image: 11260 *Flow: 11260*

Figure 6.25: Nap-Of-the-Earth Helicopter Sequence. Snapshots of images and associated flow fields in a 100 image sequence.

conservation constraint used a 9×9 window with band-pass filtered images. Seven iterations of the annealing algorithm were used per frame with a linear cooling schedule. The various parameters mentioned previously were set as follows:

Discontinuities			Weights		
$\Delta_D = 15.0$	$\Delta_S = 0.25$	$\Delta_T = 0.5$	$\beta_D = 2.0$	$\beta_S = 2.5$	$\beta_T = 1.0$

Even after only 15 frames, noise in the motion estimate is small. In Figure 6.23 a rotation to the right, in addition to the translation, can be seen. Figure 6.24 spans a largely translational sequence. Throughout this portion of the sequence however, the aircraft is undergoing significant pitching fore and aft which causes the temporal coherence constraint to be violated between frames. In the final portion of the image sequence (Figure 6.25) the helicopter is banking while rotating to the left.

Chapter 7

Incremental GNC: Algorithm and Implications

The previous chapter developed an incremental stochastic algorithm for minimizing a correlation-based objective function. Given the formulation of the problem, this stochastic approach was appropriate, but if the objective function is differentiable, then more efficient minimization procedures may be used. As we saw in Chapter 4, the GNC algorithm provides a deterministic minimization scheme that can be applied to the robust gradient-based formulation of optical flow. In this Chapter, the incremental minimization framework is applied to the robust gradient-based problem and an *Incremental Graduated Non-Convexity (IGNC)* algorithm is developed. Details of the algorithm are provided and it is used to illustrate the psychophysical implications of temporal continuity.

7.1 Incremental GNC

To illustrate the IGNC approach we adopt the robust gradient-based formulation of optical flow presented in Chapter 4, but here we add a temporal continuity term:

$$E(\mathbf{u}, \mathbf{u}^-) = \beta_D E_D(\mathbf{u}) + \beta_S E_S(\mathbf{u}) + \beta_T E_T(\mathbf{u}, \mathbf{u}^-). \tag{7.1}$$

Recall that the data conservation, spatial coherence, and temporal continuity constraints are formulated as follows:

$$E_D(\mathbf{u}) = \rho_D(I_x u + I_y v + I_t, \sigma_D), \tag{7.2}$$

$$E_S(\mathbf{u}_s) = \sum_{n \in \mathcal{G}_s} \rho_S(u_s - u_n, \sigma_S) + \sum_{n \in \mathcal{G}_s} \rho_S(v_s - v_n, \sigma_S), \tag{7.3}$$

$$E_T(\mathbf{u}, \mathbf{u}^-) = \rho_T(u - u^-, \sigma_T) + \rho_T(v - v^-, \sigma_T), \tag{7.4}$$

where ρ is taken to be the Lorentzian estimator.

The ISM algorithm used the Gibbs sampler as the minimization procedure and the temperature parameter was annealed over the length of the image sequence. For the IGNC algorithm we replace the stochastic minimization procedure with the Simultaneous Over-Relaxation (SOR) algorithm from Chapter 5. For the robust-gradient algorithm there is no temperature parameter, but rather there are three control parameters (σ_D, σ_S, and σ_T) which controlled the convexity of the objective function. It is these values that will be decreased over time in the same way that the temperature parameter was annealed.

The overall structure of the incremental algorithm remains the same as the ISM case and is sketched in Figure 7.1. When a new image is acquired, n iterations of a hierarchical SOR algorithm are performed, beginning with the current estimate. When the estimate has been refined, the constant acceleration assumption is applied to predict the flow at the next time instant. Since the continuation parameters are associated with particular sites, the flow estimate is used to predict their values at the next time instant. Unlike the ISM algorithm, here we choose to implement prediction using the computationally simpler backwards warp. After warping, the flow and continuation parameters are reset at motion discontinuities.

One significant difference between ISM and IGNC is in the type of hierarchical scheme employed. For ISM, we used the flow-through approach, but with IGNC we adopt the coarse-to-fine-when-changed method that is implemented by the recursive function "Pyramid-SOR" in Figure 7.2. In the ISM approach the flow-through strategy was used to efficiently implement the forwards warp by restricting the maximum motion at any level of the pyra-

```
;;;
;;; Incremental Minimization using:
;;;        1. Robust gradient-based formulation,
;;;        2. Coarse to fine when changed,
;;;        3. Simultaneous Over-Relaxation.
;;;
u, u⁻ ← [0, 0]
σ_D, σ_S, σ_T ← initial value at every site
n ← fixed, small number of iterations
for each image
    ;; perform n iterations of SOR
    u ← Pyramid-SOR(I_{t-1}, I_t, max-level, min-level, n, u⁻, u⁻)
    ;; Constant acceleration assumption
    u ← u + (u − u⁻)
    ;; prediction: warp flow and continuation parameters
    u⁻(x, y) ← u(x − u, y − v)
    σ_i(x, y) ← σ_i(x − u, y − v),  i ∈ {D, S, T}
    ;; reset flow and control parameters at motion boundaries
    if location (x, y) is occluded or disoccluded then
        σ_D, σ_S, σ_T ← initial value
        u, u⁻ ← [0, 0]
    else
        σ_i(x, y) ← f(σ_i(x, y)),  i ∈ {D, S, T}
    end if
end.
```

Figure 7.1: Incremental Graduated Non-Convexity Algorithm.

```
;;;
;;; "Coarse-to-Fine-When-Changed" Method.
;;;
Pyramid-SOR (I_{t-1}, I_t, max-level, min-level, iters, u, u−)
    if max-level ≤ min-level then
        δu ← u
        u ← [0, 0]
        u ← SOR(I_{t-1}, I_t, δu, u, u^-, iters)
    else
        ;;
        ;; recursively call Pyramid-SOR
        ;;
        u^{p-1} ← Pyramid-SOR(reduce(I_{t-1}), reduce(I_t), max-level − 1,
                              min-level, iters, (reduce(u)/2), (reduce(u^-)/2))
        u^p ← project(u^{p-1}, max-level − 1)
        ;;
        ;; Coarse to fine when changed, u,v are updated by coarse level
        ;;
        when |u − u^p| > 0.5 or |v − v^p| > 0.5
            u ← u^p
        ;;
        ;; warp image I_{t-1} by (u, v)
        ;;
        I_{t-1}(x, y) ← I_{t-1}(x − u, y − v)
        δu ← [0, 0]
        u ← SOR(I_{t-1}, I_t, δu, u, u^-, iters)
    end
    return u
end
```

Figure 7.2: Coarse-to-Fine-When-Changed Algorithm.

mid. The flow-through approach also permitted increased parallelism which helped offset the computational expense of the stochastic algorithm.

In the case of IGNC, the coarse-to-find-when-changed strategy is made feasible by replacing the forwards warping scheme with a backwards warp. Additionally, the deterministic relaxation strategy employed in the IGNC approach results in faster convergence than stochastic algorithm. As a result of this increased efficiency, we are willing the pay the price of sequentially processing each level in the pyramid to gain the benefits of refinement across scales.

The SOR procedure for the robust gradient-based estimation problem is sketched in Figure 7.3, where:

$$\begin{aligned}
\frac{\partial E}{\partial u_s} &= \beta_D I_x \psi((I_x \delta u_s + I_y \delta v_s + I_t), \sigma_D) + \\
&\qquad \beta_S \sum_{n \in \mathcal{G}_s} \psi(u_s - u_n, \sigma_S) + \beta_T \psi(u_s - u_s^-, \sigma_T) \\
\frac{\partial E}{\partial v_s} &= \beta_D I_y \psi((I_x \delta u_s + I_y \delta v_s + I_t), \sigma_D) + \\
&\qquad \beta_S \sum_{n \in \mathcal{G}_s} \psi(v_s - v_n, \sigma_S) + \beta_T \psi(v_s - v_s^-, \sigma_T),
\end{aligned}$$

and

$$\psi(x, \sigma) = \frac{2x}{2\sigma^2 + x^2}.$$

Given an image I_{t-1} which has already been warped by the current flow estimate $\mathbf{u}$, the algorithm computes the $\delta\mathbf{u}$ which refines the estimate. Sites are divided in a checkerboard pattern into black and white sites and each group is updated in parallel.

The IGNC algorithm has a number of advantages over the ISM approach. First, IGNC is computationally much simpler than ISM. Second, sub-pixel estimates are naturally recovered given the gradient-based formulation and they can be refined across levels. Finally, since the approach is deterministic, it starts with a coarse solution and refines it across scale and time. With the stochastic approach, the results at high temperatures are, by design, random and hence do not provide useful coarse descriptions at early stages of the processing.

```
;;;
;;; Simultaneous Over-Relaxation
;;;
SOR(I_{t-1}, I_t, δu, u^p, u^-, iters)
    ;; compute derivatives by convolving with masks G_x and G_y
    I_x ← G_x * I_t
    I_y ← G_y * I_t
    I_t ← I_t − I_{t-1}
    ;; compute bounds on second derivatives of E
    S_u ← β_D I_x^2 / σ_D^2 + 4β_S / σ_D^2 + β_T / σ_T^2
    S_v ← β_D I_y^2 / σ_D^2 + 4β_S / σ_S^2 + β_T / σ_T^2
    for iters iterations do
        for sites ∈ {black, white} do
            u ← u^p + δu
            δu ← δu − ω (1/S_u) ∂E/∂u
            u ← u^p + δu
            δv ← δv − ω (1/S_v) ∂E/∂v
        end
    end
    u ← u + δu
    return u
end
```

Figure 7.3: Simultaneous Over-Relaxation (SOR) Algorithm.

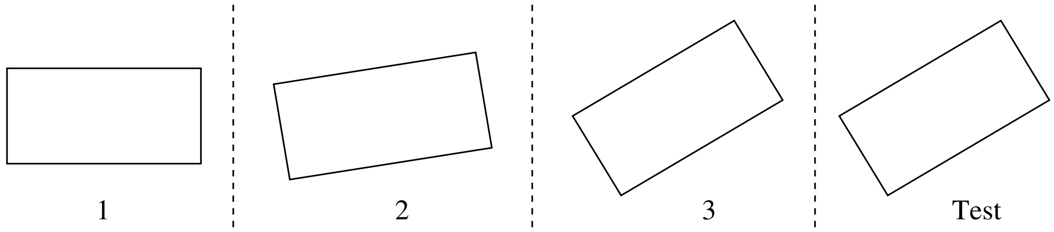


Figure 7.4: **Rotation Experiment,** (Freyd and Finke [1985])

Two experiments with the IGNC algorithm are presented in the following section and are used to explore the implications of the temporal continuity assumption.

7.2 Psychophysical Implications

This thesis has demonstrated the computational advantages that a temporal continuity assumption can provide. The development of the constraint has been based on two goals: 1) to more accurately model the motion of surfaces over time, and 2) to extend motion estimation over time for improved efficiency. From an engineering standpoint, these are sensible goals, particularly given limited computational resources. Given the computational benefits of temporal continuity, it is natural to ask whether humans, likewise, exploit the temporal continuity of moving bodies.

This issue has been explored in the work of Freyd and her collaborators in their studies of *representational momentum*. Freyd [1987] observed that a subject's memory of the final position of a moving object was distorted in the direction of motion. Consider the experiment in Figure 7.4. Subjects were shown three images of a rectangle rotating 17 degrees per frame and told to remember the final image. They were then presented with a number of test images that were either identical to the last frame or in which the rectangle was rotated a few degrees in either direction.

Subjects were required to indicate whether the test image was the same as, or different from, the final image. Their responses are plotted in Figure 7.5. Positive values indicate

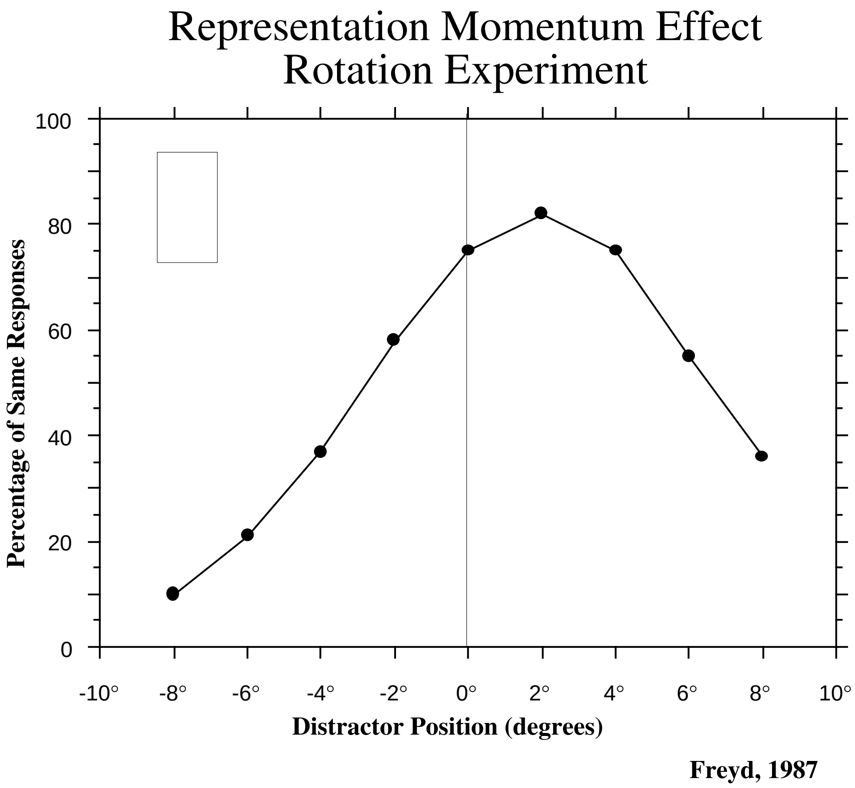


Figure 7.5: Representational Momentum Effect.

degrees of rotation in the same direction as the inducing motion. The experiment indicates that subjects experienced a slight shift in the remembered location of the rectangle in the direction of motion.

The findings of Freyd and Finke [1985] indicate that the effect is dependent on coherent implied motion and is not a result of the subjects guessing the next logical orientation of the block in the image sequence. This effect occurs very rapidly (10-100 ms) and is viewed as a small, continuous, shift in memory of the remembered position of the object. The amount of shift increases with the implied velocity of the object and as the retention interval between the last image and the test pattern increases.

Finke and Freyd [1985] attribute this shift to a cognitive internalization of the physical properties of time and momentum in the world. In particular, they argue that the effect cannot be accounted for by low-level, sensory processes, but is rather due to an inherent dynamic property of mental representations. The justification for this assertion is derived from the fact that strong momentum effects are observed even for long retention intervals (up to two seconds between the third image and the test image). Finke and Freyd suggest that, because of the persistence of the effect over long retention intervals, the effect is due to cognitive processing and not "triggered motion detectors." Freyd [1987] further argues that the results suggest that mental representations are intrinsically dynamic.

As noted by Tarr and Black [1992], the properties of representational momentum bear a strong resemblance to the properties exhibited by our incremental flow algorithms. Our algorithms, however, do not contain high-level cognitive representations of objects or the physical properties of momentum. Rather, they contain a simple temporal continuity constraint that is formulated in terms of motion in the image plane. We ask whether this sort of early-vision constraint can account for the effects of representational momentum without appealing to dynamic cognitive representations. To explore the relationship between the temporal continuity constraint and the results in representational momentum, we repeat the psychophysical experiments but with the IGNC algorithm as the "subject".

7.2.1 Methodology

Our goal is to explore whether our temporal continuity constraint can account for the kinds of systematic distortions observed in humans. We begin by constructing image sequences similar to those used by Freyd and Finke in their psychophysical studies. Given our formulation of the optical flow problem, accurate flow estimates are only possible in textured areas, so our stimuli are constructed of textured regions moving over a textured background as in Figure 7.6. The background and foreground textures are constructed by convolving uniform random noise over the ranges $[0, 127]$ and $[128, 255]$ respectively with a Gaussian filter. For each trial a new randomly textured image sequence was generated to avoid pos-

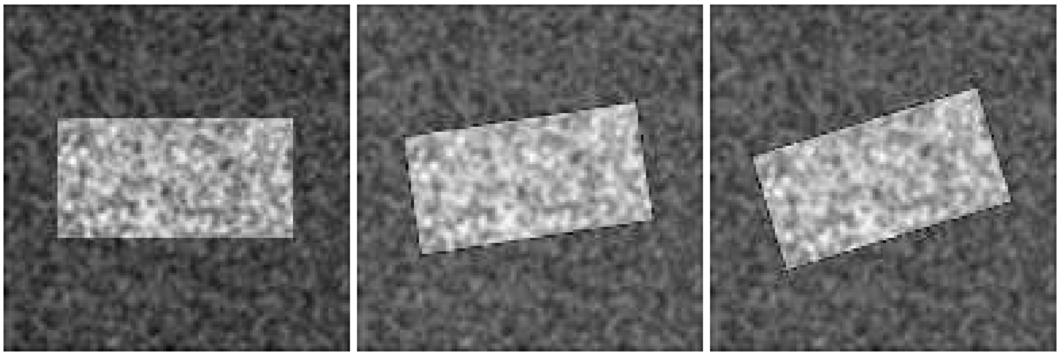


Figure 7.6: Rotation Experiment, test images.

sible systematic errors due to accidental properties of a particular random sequence.[1]

The images were 128×128 pixels in size and there were 8 degrees of rotation between frames which is less rotation than the 17 degrees used by Freyd and Finke. The first order optical flow model does not account for rotations, and a rotation of more than about 8 degrees per frame proved too much for reliable flow estimation.[2] The test images were rotated by +4, +3, +2, +1, 0, -1, -2, -3, -4 degrees from the final position.

The IGNC algorithm was applied to each test sequence of four images (three inducing images and one test image). Due to the large rotation, a three level pyramid was used where the coarse level images were 32×32 pixels. The experiments were run with between 5 and 10 iterations of the SOR algorithm at each level and the combination across levels was achieved using the coarse-to-fine-when-changed method. As in Chapter 4, the Lorentzian was chosen as the robust estimator. The only parameters that were varied were the weight, β_t, associated with the temporal term and the amount of temporal disparity, σ_t, that constituted a violation of the constraint. The parameters for the other terms were as follows for all experiments:

[1] The texture requirement is simply a byproduct of our formulation of the data and smoothness terms and is unrelated to the temporal continuity constraint. We might, for example, compute optical flow using only image features like lines [Faugeras *et al.*, 1987]. The constraint of temporal continuity is equally applicable to such a formulation.

[2] Once again this issue is separate from the issue of temporal continuity and we could reformulate the data conservation term to account for rotations.

Discontinuities		Weights	
$\Delta_D = 5.0$	$\Delta_S = 0.2$	$\beta_D = 10.0$	$\beta_S = 1.0$

The IGNC algorithm computes optical flow, but cannot answer the question, "Is this image the same as the previous image?" To perform experiments analogous to the psychophysical trials we require a way of answering this question. It is reasonable to assume that the test image that minimizes the error in the intensity constraint equation is most likely to be perceived as the "same". Thus, after estimating the optical flow over the image sequence, we compute a measure of "sameness" by taking the inverse of the data conservation term summed over the entire image:

$$\text{“same”} \sim \frac{1}{\sum (I_x u + I_y v + I_t)^2}. \tag{7.5}$$

Experiments were performed with four different "subjects". The parameters controlling the relative importance of the temporal continuity constraint and the outlier rejection threshold for temporal constraint violations were varied for each subject. The results were then averaged across subjects.

In the psychophysical trials each of the first three images was displayed for 250 ms while the final image stayed visible until the subject responded. The interstimulus interval (ISI) between images was varied from 100 to 900 ms in increments of 100 ms. The smaller the ISI, the greater the implied velocity. Time is not explicitly represented in the IGNC algorithm, but velocity is, so instead of varying the ISI, we vary the implied velocity which is determined by:

$$\nu = \frac{250\text{ms}}{\text{ISI}},$$

which means the that:

$$\nu \in \{2.5,\ 1.25,\ 0.833,\ 0.625,\ 0.5,\ 0.417,\ 0.357,\ 0.3125,\ 0.278\}.$$

After processing three images, we have a predicted flow (u_s, v_s) at every site in the image which is computed using the constant acceleration assumption. We take the predicted esti-

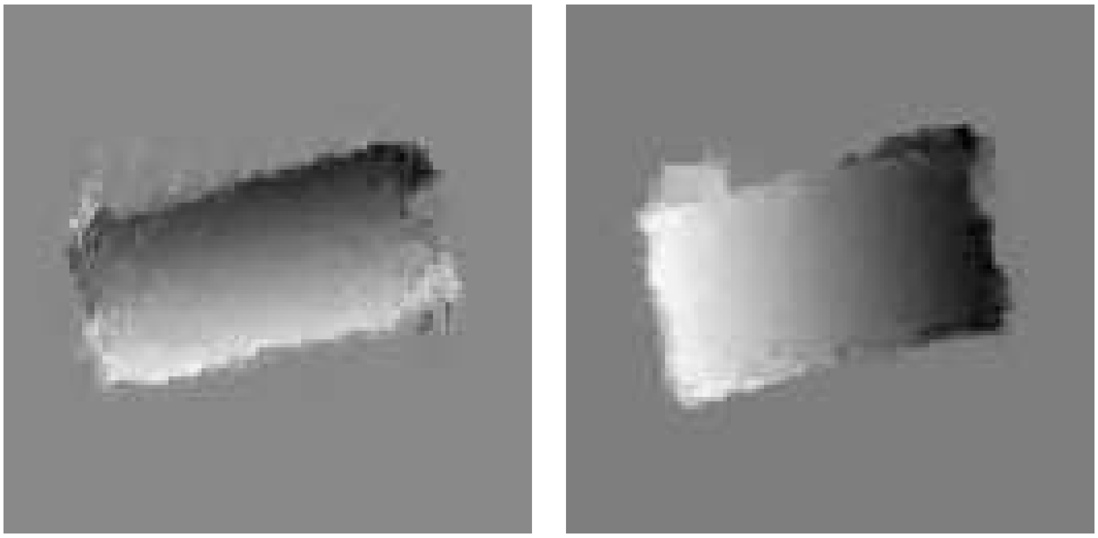


Figure 7.7: Rotation Experiment, optical flow.

mate and multiply it by the implied velocity:

$$(u_s^\nu, v_s^\nu) = (\nu u_s, \nu v_s),$$

which gives the predicted velocity between the third and test images. This estimate is then used in the warping process to predict the flow at the next time instant.

Each "subject" was run with each ISI and the results were averaged. In the psychophysical trials reaction times above 2000 ms were removed from the data. In our case reactions times were, in a sense, held fixed by performing a fixed number of iterations of the algorithm. An alternative would have been to set a threshold on the maximum error and report sameness as the inverse of the number of iterations required to reach the threshold.

7.2.2 Rotation Experiment: Results

In the case of the rotation experiment, the parameter settings for the four subjects were:

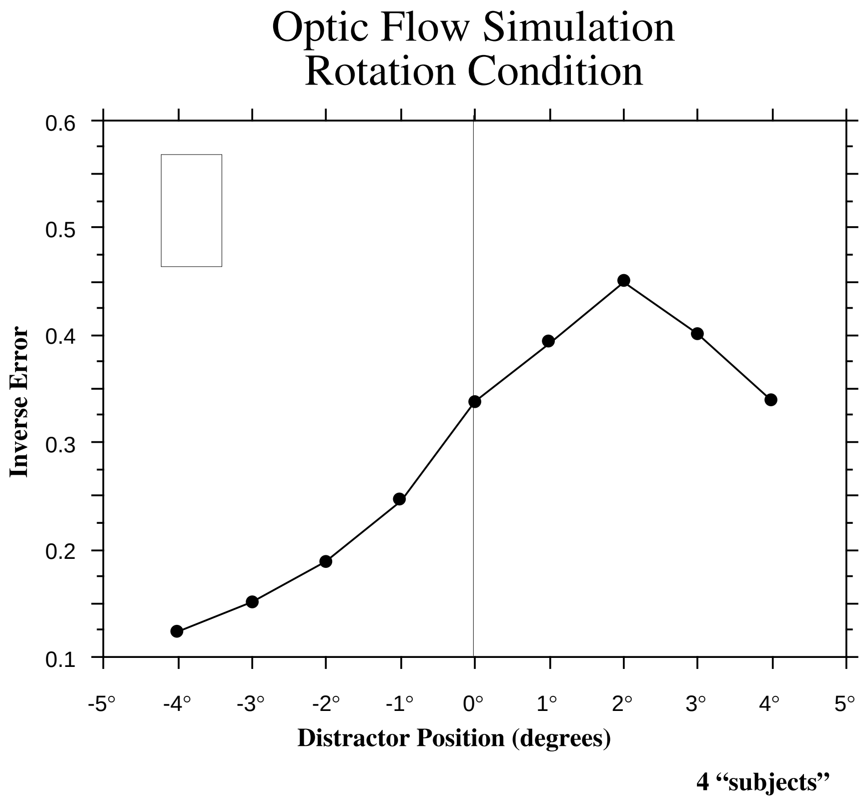


Figure 7.8: Optic Flow Simulation: Rotation condition.

Subject	β_T	σ_T	Iterations
1	3.0	0.5	10
2	4.0	0.5	10
3	4.0	0.6	10
4	4.0	0.4	10

where all other parameters were held fixed. The optical flow estimated after the three inducing images is shown in Figure 7.7.[3]

This flow is scaled by ν before warping the estimates and processing the test image. After 10 iterations of the incremental algorithm are performed with the test image, the "sameness" measure is computed using the current flow estimate. The average response across

[3] For simplicity, the version of the IGNC algorithm used did not detect motion boundaries and reset the control parameter in disoccluded regions. For this region, the flow estimates at the boundaries of the object are recovered less accurately than they might have been.

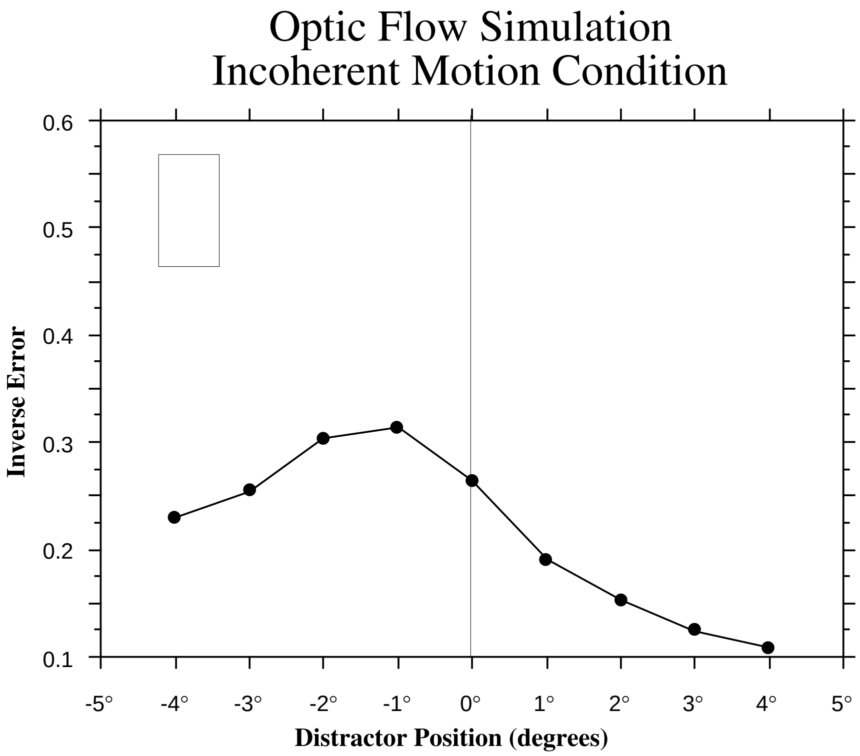


Figure 7.9: Optic Flow Simulation: Incoherent motion condition.

subjects and ISI's is shown in Figure 7.8. The graph shows a systematic distortion in the direction of the rotation that is consistent with the psychophysical data.

For the rotation experiment, we observe the maximum shift at approximately 2 degrees past the actual final position. Relative to the amount of rotation in the sequence, this is roughly twice the shift observed by Kelly and Freyd [1987]. The amount of shift, however, can be controlled by changing the settings of β_t and σ_T. We have not attempted to choose parameter settings to match our results to the psychophysical data, but merely illustrate how temporal continuity can produce the same kinds of systematic distortions.

7.2.3 Incoherent Motion Condition

Freyd and Finke [1985] demonstrated that the representational momentum effect depends on coherent motion. They performed an experiment in which the first and second induc-

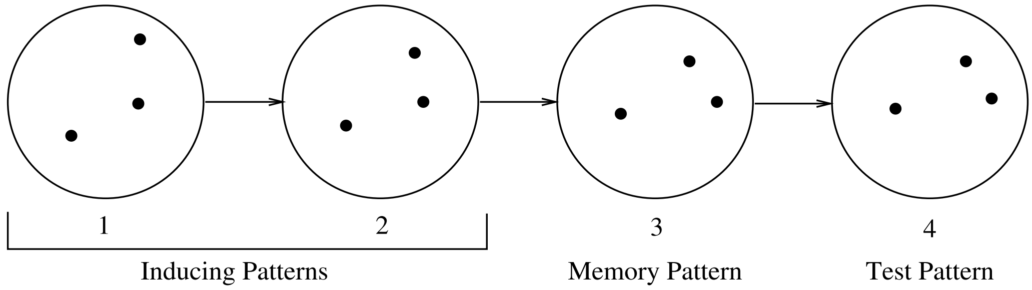


Figure 7.10: Translation Experiment, [Finke and Freyd, 1985]

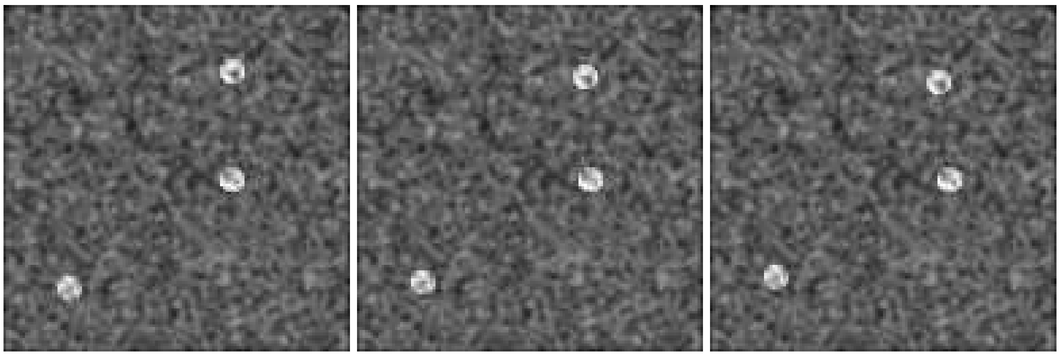


Figure 7.11: Translation Experiment, test images.

ing images were switched. With such a sequence, no significant momentum effect was observed.

We performed an analogous experiment with our "subjects" and the results are plotted in Figure 7.9. The forward distortion has vanished, but there is a slight distortion in the backwards direction. Since the first two images were reversed, the motion is initially in the backwards direction. Given the subjects' weighting of the temporal continuity constraint above, there was still a slight shift in the direction of this initial motion, however, the overall effect was markedly reduced both in the amount of shift and the magnitude of the response.

7.2.4 Translation Experiment

Finke and Freyd [1985] also performed a series of experiments using translating dots as seen in Figure 7.10. We constructed an analogous experiment using the image sequence in Figure

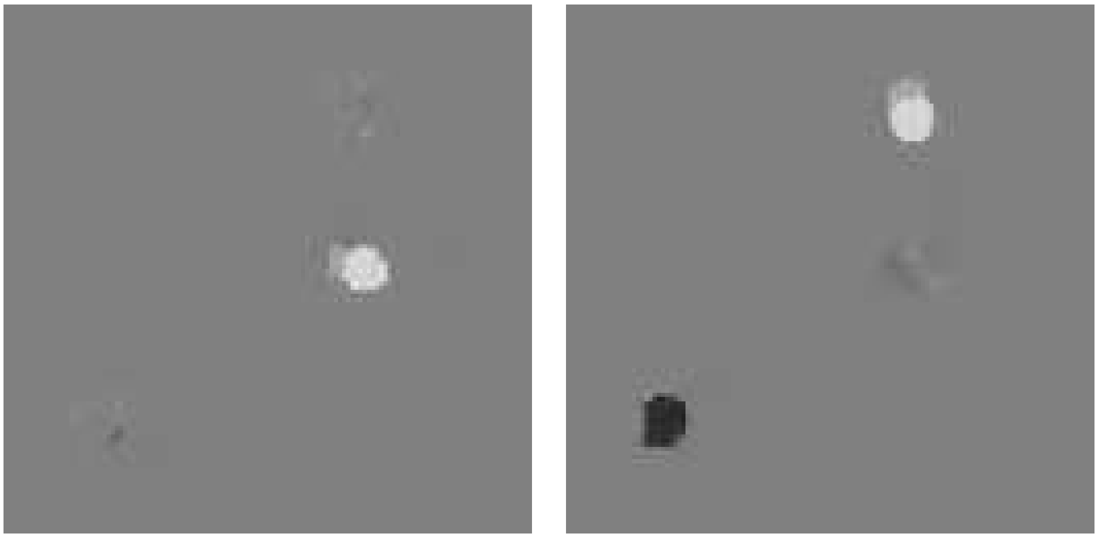


Figure 7.12: Translation Experiment, optical flow.

7.11 in which the foreground and background textures were created in the same way as in the rotation experiment. In our sequence of 128×128 images, the highest dot moved down two pixels per frame, the dot below it translated to the right by two pixels per frame, and the dot in the lower left moved up by two pixels per frame.

The experiment was carried out using the same methodology as the rotation experiment with the following parameter settings for the three "subjects":

Subject	β_T	σ_T	Iterations
5	4.0	0.25	5
6	4.0	0.4	10
7	4.0	0.4	5

For this experiment, fewer iterations of the minimization strategy were required to achieve acceptable flow estimates due to the fact that the translational motion of the dots corresponds to our model of optical flow. And, since the motion between frames was at most two pixels, a two level pyramid was used so that motion at the coarse level would be no more than one pixel. The optical flow estimate after the inducing frames is shown in Figure 7.12.

The test images contained dots that were offset from the final positions by +3, +2, +1, 0, -1, -2, -3 pixels in the direction of their original motion. The the average response across

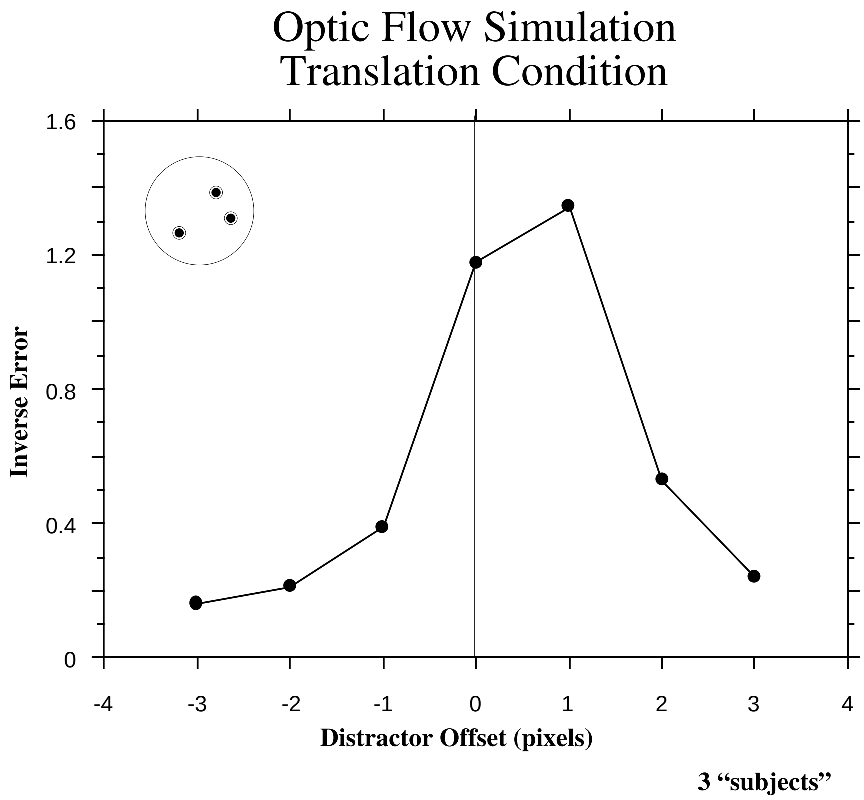


Figure 7.13: Optic Flow Simulation: Translation condition.

subjects and ISI's, after five iterations of the minimization procedure, is shown in Figure 7.13. Once again we find a systematic distortion in the direction of motion that, in this case, is less than a pixel. Note that this distortion is not the same as jumping to the next "logical" position. If that were the case, a two pixel displacement in the positive direction would be optimal.

7.2.5 Analysis

The IGNC algorithm was developed to meet our goals of incremental estimation: anytime access, temporal refinement, computation reduction, and adaptation. While our goal was to compute optical flow over time, our experiments indicate that the introduction of a temporal continuity constraint can produce momentum effects very similar to those exhibited by humans. In particular, we have observed similar shifts in the "remembered" position of objects which depend on coherent motion. This shift is purely the result of local processes in the image plane with no higher level knowledge about objects or their motion.

As Freyd and her collaborators suggest, temporal continuity is an important constraint. As such, it would not be surprising to find the constraint exploited at multiple levels of visual processing. At cognitive levels, dynamic representations, with a temporal dimension, may help explain effects such as mental rotation [Cooper, 1976]. Unlike mental rotation however, the representational momentum effect is cognitively impenetrable [Freyd, 1987]; that is, it is not affected by beliefs, expectations, or intentions.

Our results suggest is that the explanation for the phenomenon of representational momentum may exist at early stages in the processing of motion, without appealing to cognitive internalizations of physical momentum. That is not to say that the representational momentum effect in humans is due to an optical flow-like computation, merely that, from a computational viewpoint, it may be accounted for by sensory processes. Additionally, our results underscore the importance of exploiting temporal continuity in early vision.

Chapter 8

Incremental Feature Extraction

The incremental minimization framework was developed to meet the demands of optical flow estimation and, in the case of motion, there was a clear need, and advantage, to performing the recovery over time. Motion estimation, however, is just one aspect of computer vision. There are many other problems such as object recognition, segmentation, perceptual grouping, shape recovery, and stereo analysis. Most of the work on these problems has focused on the analysis of a single image or multiple static views.

For real-time dynamic systems requiring visual scene analysis, especially for navigation related applications, the view of the world is constantly changing. In such an environment it is unlikely that the system has the luxury of processing each image to recover a complete description of the scene. To do so would further imply that the system effectively has no visual memory, but rather "sees" each image as a new scene which must be interpreted. In addition to being computationally infeasible, such an approach sacrifices robustness that can be gained by integrating information over time.

In general, it would be desirable to recover scene descriptions over time in the same way that we recover motion over time. In this chapter, we show that our approach to incremental motion analysis can be generalized to other vision problems as well. The key idea is to formulate the problem in terms of objective function minimization, with suitable spatial and temporal constraints [Black, 1992a].

A similar idea is described by Heel for surface reconstruction using the Kalman filter framework, which he calls "Dynamic Motion Vision" [Heel, 1989]. Singh [1992b] also uses such a framework to estimate both motion and image intensity over an image sequence in order to enhance noisy intensity image sequences. Such an approach has medical applications for problems involving low-dosage X-ray sequences in which the cumulative amount of radiation the patient receives over the sequence must be kept low. These low dosages, however, result in a low signal to noise ratio. Singh has shown that, by incrementally estimating intensity along with motion, that the quality of the images can be enhanced over time.

In this chapter, we show that our framework allows static objective function minimization problems to converted to incremental estimation problems. To illustrate the generality of our approach we consider the problem of recovering image features that are both physically significant and perceptually salient. A more complete description scene description might include a representation of the surfaces present, surface boundaries, surface properties relevant to the task, and relationships between surfaces, but we limit the scope of our present goals in order to show that the applicability of the incremental estimation framework is not limited to motion estimation.

The goal of this process is to simultaneously *extract* and *track* image features over time. The features we consider are intensity discontinuities; that is, edges. Simultaneous extraction and tracking allows the incremental improvement and refinement of the features. Additionally, by combining motion and intensity information, discontinuities can be classified as surface markings or actual surface boundaries.

For illustration, we formulate discontinuity extraction via a simple model of image regions using local constraints on intensity and motion. These regions correspond to surface patches of constant intensity. The constraints model patch boundaries as discontinuities in intensity, motion, or both intensity and motion. The recovery problem is then modeled as a

Markov random field in which patch boundaries are represented as line processes. Feature extraction is performed dynamically over a sequence of images by exploiting the techniques of *incremental stochastic minimization (ISM)* described in Chapter 6.

8.1 Previous Work

Previous approaches to feature extraction, using energy minimization formulations, have focused on either static boundary detection, image segmentation, or motion segmentation. Static approaches that attempt to recover surface boundaries from the 2D properties of a single image are usually not sufficient for a structural description of the scene. These techniques include the recovery of perceptually significant image properties; for example segmentation based on intensity [Blake and Zisserman, 1987; Chou and Brown, 1990] or texture [Derin and Elliott, 1987; Geman *et al.*, 1990], location of intensity discontinuities, and perceptual grouping of regions or edges. While there are serious limitations to using these techniques alone to recover structure, their results can be used heuristically as cues to possible surface boundaries due the fact that different surfaces often have different material properties and, hence, may have different texture or intensity.

Structural information about image features can be gained by analyzing their behavior over time. Attempts to deal with image features in a dynamic environment have focused on the tracking of features over time [Navab *et al.*, 1990; Viéville and Faugeras, 1990]. A notable exception to the tracking approach detects moving intensity edges over time by observing the space-time behavior of the edge moving across a fixed detector array [Kahn, 1988; Kahn, 1985].

Motion segmentation, on the other hand, attempts to segment the scene into structurally significant regions using image motion. Early approaches focused on the segmentation and analysis of the computed flow field [Thompson *et al.*, 1985]. Other approaches have attempted to incorporate discontinuities into the flow field computation [Black and Anandan, 1991b; Murray and Buxton, 1987; Tian and Shah, 1992], thus computing flow and segmenting simultaneously. There has been recent emphasis on segmenting and tracking image regions using motion, but without computing the flow field [Bouthemy and Lalande, 1990; Bouthemy and Rivero, 1987; François and Bouthemy, 1991; Peleg and Rom, 1990]. While these approaches are promising since they provide structural information, they typically provide only a coarse segmentation of the scene.

In attempt to improve motion segmentation a number of researchers have combined intensity and motion information. Thompson [1980] describes a region merging technique that uses similarity constraints on brightness and motion for segmentation. Heitz and Bouthemy [1990] combine gradient based and edge based motion estimation and realize improved motion estimates and the localization of motion discontinuities. In the context of stereo reconstruction, Luo and Maître [1990] use a segmented intensity image to correct and improve disparity estimates.

8.2 Joint Modeling of Intensity and Motion with Discontinuities

To model our assumptions about the intensity structure and motion in the scene we adopt a Markov random field approach [Geman and Geman, 1984] in which we formalize our prior model in terms of constraints, defined as energy functions over local neighborhoods our grid of sites S. As we did in Chapter 2, we add a dual lattice, $l(s,t)$, of line variables between all sites s and their neighbors $t \in \mathcal{G}_s$.

This line process defines the boundaries of the image patches. If $l(s,t) = 1$ then the sites s and t are said to belong to the same image patch. In the case where $l(s,t) = 0$, the neighboring sites are disconnected and hence a discontinuity exists.

Associated with each site s is a random vector $X(t) = [\mathbf{u}, i, l]$ that represents the horizontal and vertical image motion $\mathbf{u} = (u, v)$, the intensity i, and the discontinuity estimates

l at time t. A discrete state space $\Lambda_s(t)$ defines the possible values that the random vector can take on at time t.

To model surface patches we formulate three energy terms, $E_{\mathcal{M}}$, $E_{\mathcal{I}}$, and $E_{\mathcal{L}}$ that express our prior beliefs about the motion field, the intensity structure, and the organization of discontinuities respectively. The energy terms are combined into an objective function that is to be minimized:

$$E(\mathbf{u}, \mathbf{u}^-, i, i^-, l, l^-) = E_{\mathcal{M}}(\mathbf{u}, \mathbf{u}^-, l) + E_{\mathcal{I}}(i, i^-, l) + E_{\mathcal{L}}(l, l^-). \tag{8.1}$$

The terms $\mathbf{u}^-$, i^-, and l^- are predicted values obtained by the incremental minimization process.

We convert the energy function, E, into a probability measure Π by exploiting the equivalence between Gibbs distributions and MRF's:

$$\Pi(X(t)) = Z^{-1} e^{-E(X(t))/T(t)}, \tag{8.2}$$

where Z is the normalizing constant:

$$Z = \sum_{X(t) \in \Lambda(t)} e^{-E(X(t))/T(t)}. \tag{8.3}$$

Minimizing the objective function is equivalent to finding the maximum of Π.

8.2.1 The Intensity Model

We adopt a piecewise constant, or *weak membrane*, model of intensity [Blake and Zisserman, 1987]. This first order approximation to image intensity can easily be extended to higher order approximations [Blake and Zisserman, 1987; Geman and Reynolds, 1992] or to more complex texture models [Geman *et al.*, 1990]. The current formulation differs from previous formulations in that we add a temporal continuity term to express the expected change in the image over time.

The prior model of image intensity is formulated as the energy term:

$$E_{\mathcal{I}}(I, i, i^-, l, s) = \omega_{D_{\mathcal{I}}} D_{\mathcal{I}}(I, i, s) + \omega_{T_{\mathcal{I}}} T_{\mathcal{I}}(i, i^-, s) + \omega_{S_{\mathcal{I}}} S_{\mathcal{I}}(i, l, s), \tag{8.4}$$

where the ω_* are constant weights that control the relative importance of the constraints, and where the data conservation term is defined as:

$$D_{\mathcal{I}}(I, i, s) \quad = \quad (I(s) - i(s))^2. \tag{8.5}$$

This expresses the constraint that the current estimate i should be close to the current intensity image I.

The temporal continuity term expresses the notion that the current estimate should not differ from the predicted value i^-:

$$T_{\mathcal{I}}(i, i^-, s) \quad = \quad (i(s) - i^-(s))^2. \tag{8.6}$$

Finally, the spatial coherence term expresses an expectation of piecewise constant image patches with discontinuities:

$$S_{\mathcal{I}}(i, l, s) \quad = \quad \sum_{n \in \mathcal{G}_s} l(s, n)(i(s) - i(n))^2. \tag{8.7}$$

When no discontinuity is present between sites s and n ($l(s, n) = 1$) we expect the differences in neighboring intensity values to be similar. If, however, a discontinuity is present ($l(s, n) = 0$) the difference between neighbors does not contribute to the energy term.

8.2.2 The Boundary Model

We want to constrain the use of discontinuities based on our expectations of how they occur in images. For example, we expect discontinuities to be rare and particular combinations to be more likely than others. Hence, discontinuities that do not conform to expectations are penalized. The boundary model can then be expressed as the sum of a temporal continuity

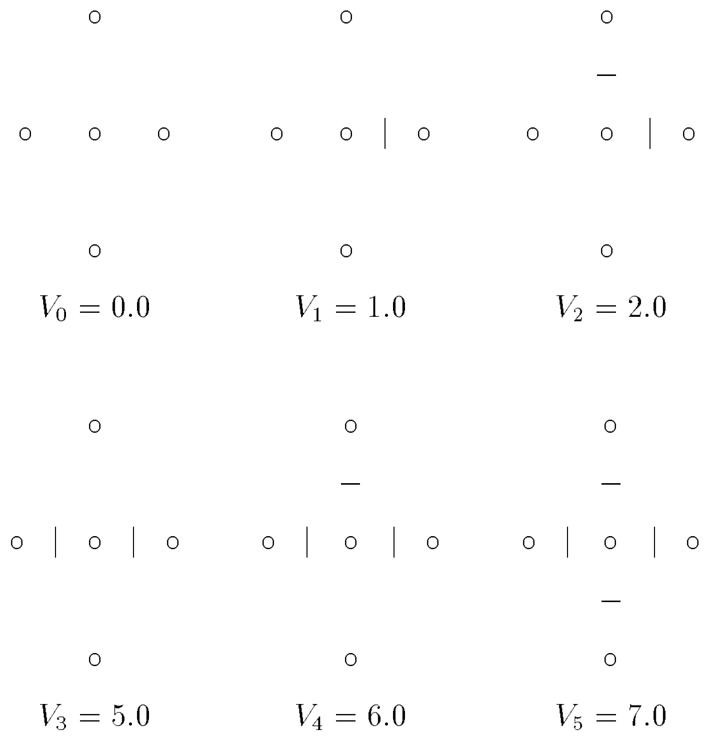


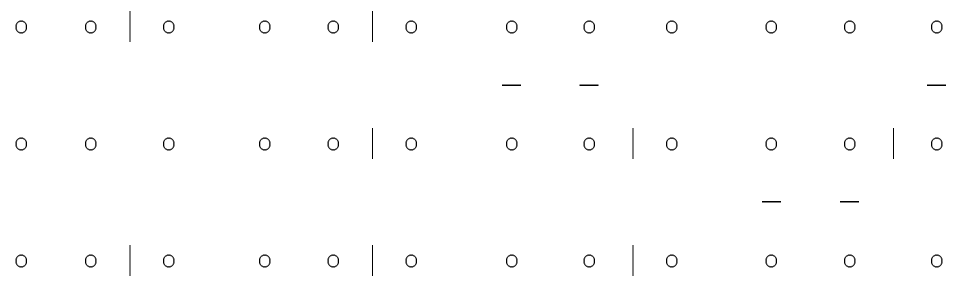
Figure 8.1: Examples of local surface patch discontinuities. Configuration V_0 (no discontinuities) is preferred to the situation, V_1, where a discontinuity is introduced. A corner, V_2, is deemed less likely than a single discontinuity. Cliques V_3, V_4, and V_5, are highly penalized as they do not admit plausible physical interpretations.

term and a penalty term defined as the sum of clique potentials V_C over a set of cliques $\mathcal{C}$:

$$E_{\mathcal{L}}(l, l^{-}, s) = \omega_{T_{\mathcal{L}}} \sum_{n \in \mathcal{G}_s} (l(s,n) - l^{-}(s,n))^2 + \omega_{P_{\mathcal{L}}} \sum_{C \in \mathcal{C}} V_C(l), \tag{8.8}$$

where $\omega_{T_{\mathcal{L}}}$ and $\omega_{P_{\mathcal{L}}}$ are constant weights.

One component of the penalty term expresses our expectation about the local configuration of discontinuities about a site. Figure 8.1 shows the possible local configurations up to rotation. We also express expectations about the local organization of boundaries; for example we express notions like "good continuation" and "closure" that correspond to assumptions about surface boundaries (Figure 8.2). The values for these clique potentials



$V_6 = 1.0$ $V_7 = -2.0$ $V_8 = -1.0$ $V_9 = 0.0$

Figure 8.2: Examples of local organization of discontinuities based on continuity with neighboring patches. The lack of continuation in V_6 is penalized, while good continuation, V_7 is rewarded. Corners, V_8, and steps, V_9, are also rewarded.

were determined experimentally and are similar to those of previous approaches [Chou and Brown, 1990; Murray and Buxton, 1987].

8.2.3 The Motion Model

The motion model we adopt is similar to that already considered in the ISM framework in that we express our prior assumptions about the motion in terms of three constraints: data consistency, temporal continuity, and spatial coherence. This prior motion model is formulated as an objective function:

$$E_{\mathcal{M}}(I_n, I_{n+1}, \mathbf{u}, \mathbf{u}^-, l, s) = \\ \omega_{D_{\mathcal{M}}} D_{\mathcal{M}}(I_n, I_{n+1}, \mathbf{u}, s) + \omega_{T_{\mathcal{M}}} T_{\mathcal{M}}(\mathbf{u}, \mathbf{u}^-, s) + \omega_{S_{\mathcal{M}}} S_{\mathcal{M}}(\mathbf{u}, l, s), \tag{8.9}$$

where the ω_* are constant weights, and where the spatial term is analogous to that of the intensity model:

$$S_{\mathcal{M}}(\mathbf{u}, l, s) = \sum_{t \in \mathcal{G}_s} l(s,t) \|\mathbf{u}(s) - \mathbf{u}(t)\|^2. \tag{8.10}$$

We could formulate separate line processes for intensity and motion discontinuities, but prefer to have a single line process that represents a discontinuity either due to intensity or motion. If a motion discontinuity is present, this prevents smoothing the intensity values across the surface boundary. Intensity discontinuities can be viewed as providing evidence for a possible surface boundary and hence we do not want to smooth the flow field across them.

For the temporal term we assume constant acceleration, and formulate the term without discontinuities as:

$$T_{\mathcal{M}}(\mathbf{u}, \mathbf{u}^-, s) \quad = \quad \|\mathbf{u}(s) - \mathbf{u}^-(s)\|^2. \tag{8.11}$$

We have not incorporated temporal discontinuities into the model for simplicity. They could be added either explicitly by introducing a temporal outlier process or implicitly by using a robust estimator.

For the data conservation constraint we adopt the robust correlation approach from Chapter 6:

$$D_{\mathcal{M}}(\mathbf{u}) = \sum_{(x,y)\in\mathcal{R}} \rho_D(I(x,y,t) - I(x+u, y+v, t+1), \Delta_D), \tag{8.12}$$

with the following robust estimator:

$$\rho_D(x, \Delta_D) = \frac{-1}{1 + \left(\frac{x}{\Delta_D}\right)^2}, \tag{8.13}$$

where Δ_D is a constant scale factor ($\Delta_D = 5.0$ in our experiments). Sub-pixel motions are computed as before.

8.3 The Computational Problem

We have seen how to use the Gibbs sampler to minimize non-convex objective functions like the one presented here. As mentioned earlier, each site contains a random vector $X(t) = [\mathbf{u}, i, l]$ that represents the motion, intensity, and discontinuity estimates at time t. The discontinuity component of this state space is taken to be binary, so that $l \in \{0, 1\}$. While

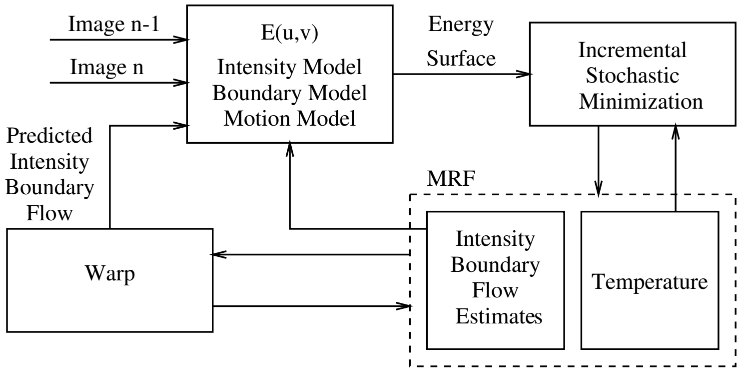


Figure 8.3: Incremental feature extraction within the ISM framework.

this works well in practice, it does not allow sub-pixel localization of the discontinuities. It may be possible to represent and recover sub-pixel discontinuity estimates by allowing real valued connections between sites [Ballard, 1987; Szeliski, 1988].

The intensity component i can take on any intensity value in the range $[0, 255]$. For efficiency, we can restrict i to take on only integer values in that range. This, however, still results in a large state space. We make the further approximation that the value of i at site s is taken from the union of intervals of intensity values about $i(s)$, the neighbors $i(t)$ of s, and the current data value $I_n(s)$. Small intervals result in a smaller state space without any apparent degradation in performance.[1]

The motion component $\mathbf{u} = (u, v)$ is defined over a continuous range of displacements u and v and hence we use the adaptive state space presented in Chapter 6 for the continuous annealing problem.

Incremental Minimization

To minimize our new objective function over time we pose the problem as one of incremental minimization as shown in Figure 8.3. The minimization stage involves updating each of the current motion, intensity, and line-process estimates while holding the neighboring

[1] A similar approach is taken by Geman and Reynolds [1992] in the context of image restoration.

estimates fixed. After a new estimate, $[\mathbf{u}, i, l]$, has been computed, the properties of the associated surface are then propagated to the new site where the patch has moved. Instead of simply warping the properties associated with the flow estimates we now also warp the intensity and discontinuity information to get new estimated values $[\mathbf{u}^-, i^-, l^-]$. This forwards warping is performed using the same prediction scheme as the ISM approach.[2]

Motion Discontinuities

Given a current estimate, $[\mathbf{u}, i, l]$, the line-process values l can be analyzed to determine whether they are motion boundaries or intensity boundaries. We set a threshold, τ_m, that determines the maximum allowable disparity in local motion estimates. A pixel is classified as a motion boundary if a discontinuity is present and the disparity in the motion estimates between neighbors is greater than τ_m:

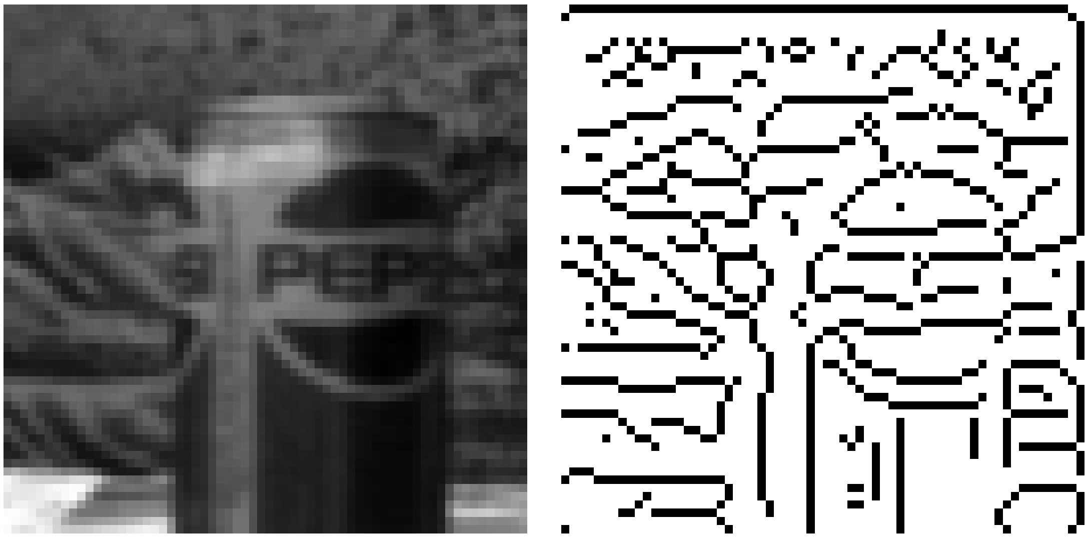
$$\tau_m < \max_{t \in \mathcal{G}_s}(l(s,t) \cdot \max(|u_s - u_t|, |v_s - v_t|)).$$

All other discontinuities are classified as surface markings. The motion discontinuities are further classified as occluding or disoccluding using the technique that measures the total flow into a site.

8.4 Experimental Results

A number of experiments have been performed using real image sequences. For these experiments, the parameters of the model were determined empirically and then used for all experiments. The intensity model parameters were: $\omega_{D_\mathcal{I}} = \omega_{T_\mathcal{I}} = 1/40^2$ and $\omega_{S_\mathcal{I}} = 1/20^2$. For the boundary model, we set the weights as follows: $\omega_{T_\mathcal{L}} = 0.5$ and $\omega_{P_\mathcal{L}} = 1.0$. Finally, for the motion model, we have: $\omega_{D_\mathcal{M}} = 0.5$, $\omega_{T_\mathcal{M}} = 0.1$, and $\omega_{S_\mathcal{M}} = 1.5$, with a 3×3 correlation window. An initial temperature of $T(0) = 0.3$ was chosen with a annealing schedule of $T(t+1) = T(t) - 0.0025$.

[2] This warping may result in non-integer estimates for i^- and non-binary values for l^-.



a b

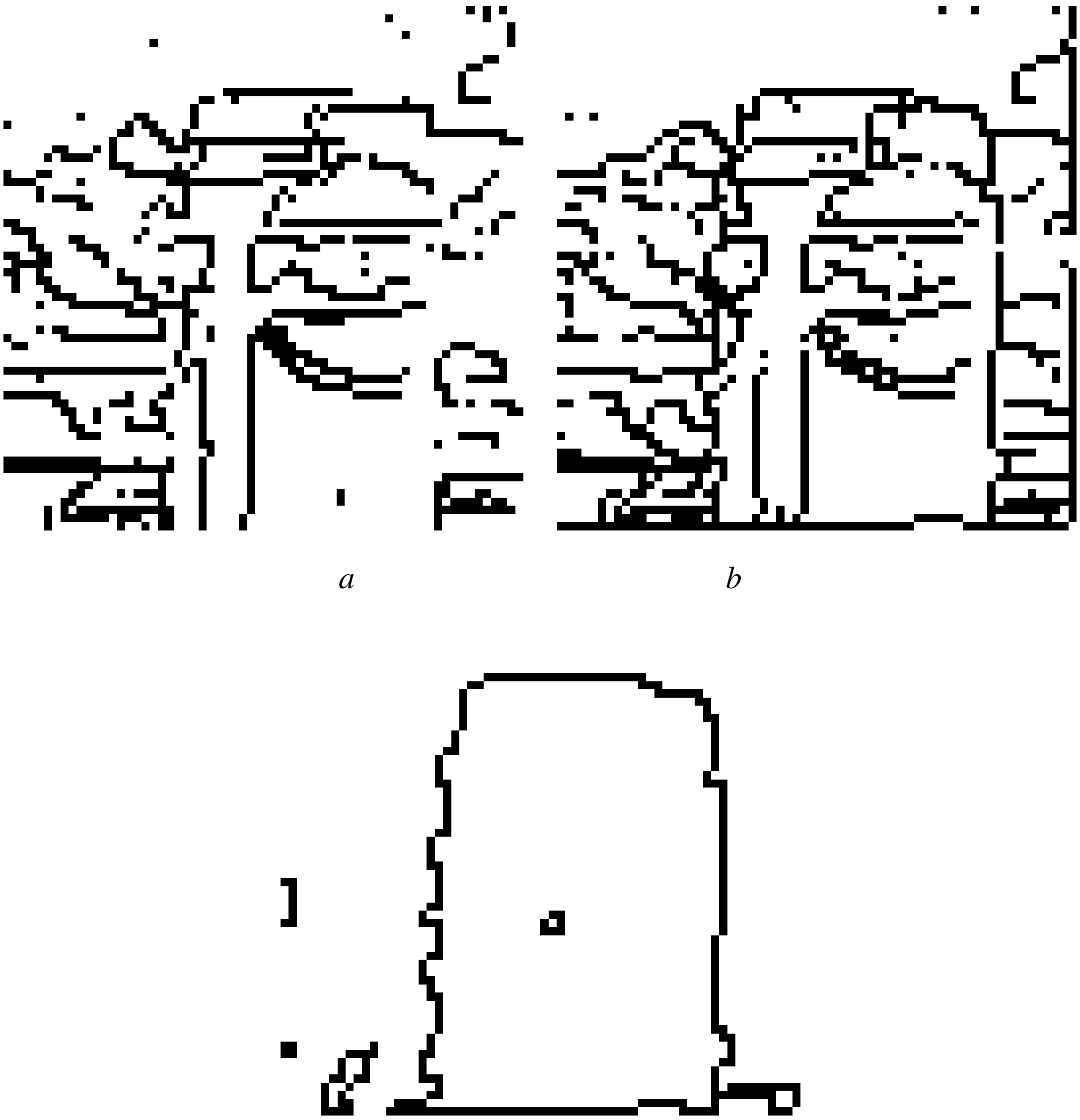
Figure 8.4: **Can and Canny:** *a)* First image in the soda can sequence. *b)* Edges in the image extracted with the Canny edge operator.

8.4.1 The Pepsi Sequence

The first experiment uses the Pepsi-can image sequence consisting of ten 64×64 square images. As an example of traditional, intensity-based, feature extraction techniques, the Canny edge operator was applied to the image. [3] The edges are shown in Figure 8.4*b*. For comparison, Figure 8.5*a* shows intensity-based feature extraction using a piecewise constant intensity model with no motion, or temporal, information. The figure shows the estimate for a single static image after 25 iterations of the Gibbs sampler. As with the Canny edges, the results correspond to intensity markings.

Figure 8.5*b* shows the results for the same image when a joint intensity and motion model is used. The results are from a two image sequence after 25 iterations. Compare the boundaries corresponding to the right and left edges of the can. In Figure 8.5*a* the similarity of intensity between the can and the background results in smoothing across the object

[3]The quality of the features is not very good, but the image is only 64×64 pixels in size.



a *b*

c

Figure 8.5: **Feature extraction.** *a)* Intensity-based feature extraction without motion. *b)* Features extracted using a joint intensity and motion model. *c)* Structural features in the scene.

boundary. When motion information is added in Figure 8.5*b* the object boundary is detected and smoothing does not occur across it.

Not only does the joint intensity and motion model improve the extraction process, it provides additional information about the scene. In particular, it allows us to classify discontinuities as structural properties of the scene or purely surface markings. Figure 8.5*c* shows the motion boundary detected with the joint model.

The power of the approach, however, does not lie in the ability to recover features using one or two frames, but rather in the ability to perform the recovery incrementally over an image sequence. Figure 8.6 shows the results of processing the full ten image sequence. For this experiment, no sub-pixel motion estimation was used and five iterations of the annealing algorithm were performed between frames.

Figure 8.6*a* shows the last image in the sequence. Figure 8.6*b* shows the reconstructed intensity image that reflects the piecewise constant intensity estimates in the image patches. The horizontal and vertical motion is shown in Figures 8.6*c* and *d* respectively. Dark areas indicate leftward or upward motion and similarly, bright areas indicate motion to the right and down. Notice that motion estimates are available in homogeneous areas where motion estimates are typically poor. Also, the modeling of discontinuities allows sharp motion boundaries and prevents over-smoothing.

Figure 8.7*a* shows the values of $l(s,t)$ that were classified as motion boundaries, while Figure 8.7*b* shows the classification of these motion boundaries as occluding (bright areas) or disoccluding (dark areas).

Figure 8.8 illustrates the evolution of the features over the ten image sequence. The estimates start out noisy and are refined over time. Only five iterations of the annealing algorithm were used between each pair of frames. By carrying out the minimization over the sequence, the amount of computation between frames is kept constant without sacrificing the quality of the recovered features. The processing time for each frame was approximately

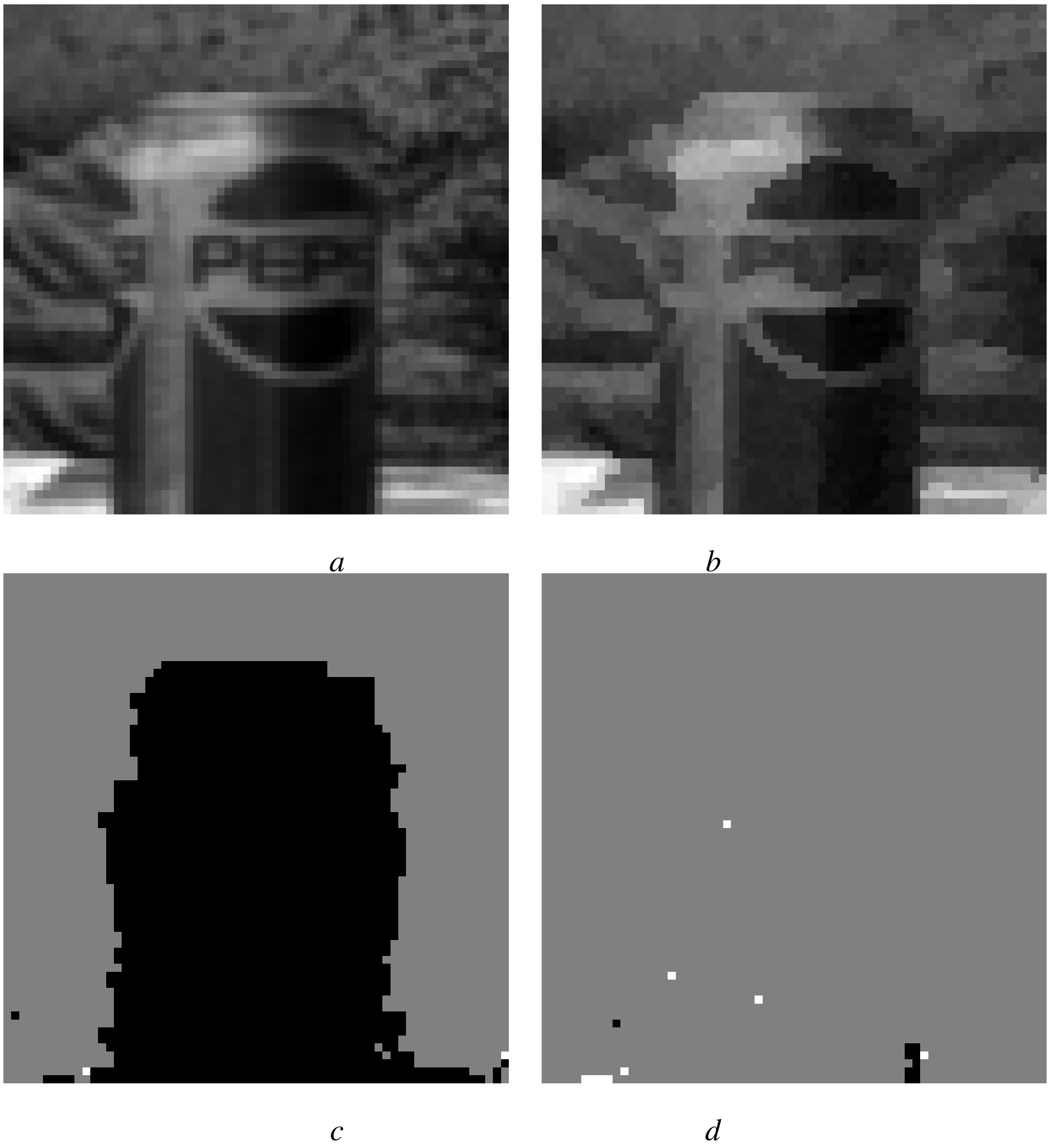


Figure 8.6: Incremental Feature Extraction. Results for a ten image sequence. *a)* Last image in the sequence. *b)* Reconstructed intensity image. *c)* Horizontal component of image motion. *d)* Vertical component of image motion.

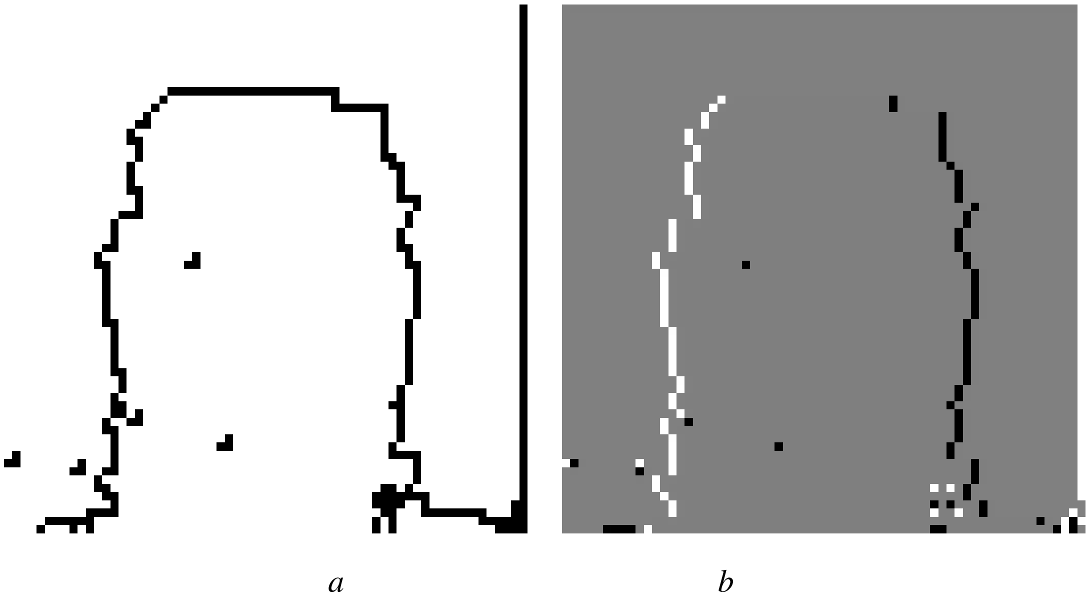


Figure 8.7: Incremental Feature Extraction. Results for a ten image sequence. *a)* Motion boundaries. *b)* Occlusion and disocclusion boundaries.

30 seconds with approximately 44% of the computation performed on the Connection Machine.

The knowledge of motion boundaries along with the first-order flow estimates may provide enough information for many purposive vision tasks. One could also compute other properties of the image patches including the depth, orientation, and curvature of the patches. If such a description were available, the scene could be reconstructed from the patch data. We use the disparity estimates to construct a pseudo depth map that is then used to illustrate such a reconstruction. In Figure 8.9*a* the disparity data and patch boundaries are used to reconstruct a piecewise smooth version of the $2\frac{1}{2}$ dimensional scene.[4] Motion discontinuities correspond to depth discontinuities, while intensity discontinuities appear as surface markings. In Figure 8.9*b* the intensity estimates were used to construct a realistic rendering of the original scene.

[4]For this experiment, sub-pixel motion estimates were not computed. For this reason the surface of the can appears flat when it is, in fact, curved.

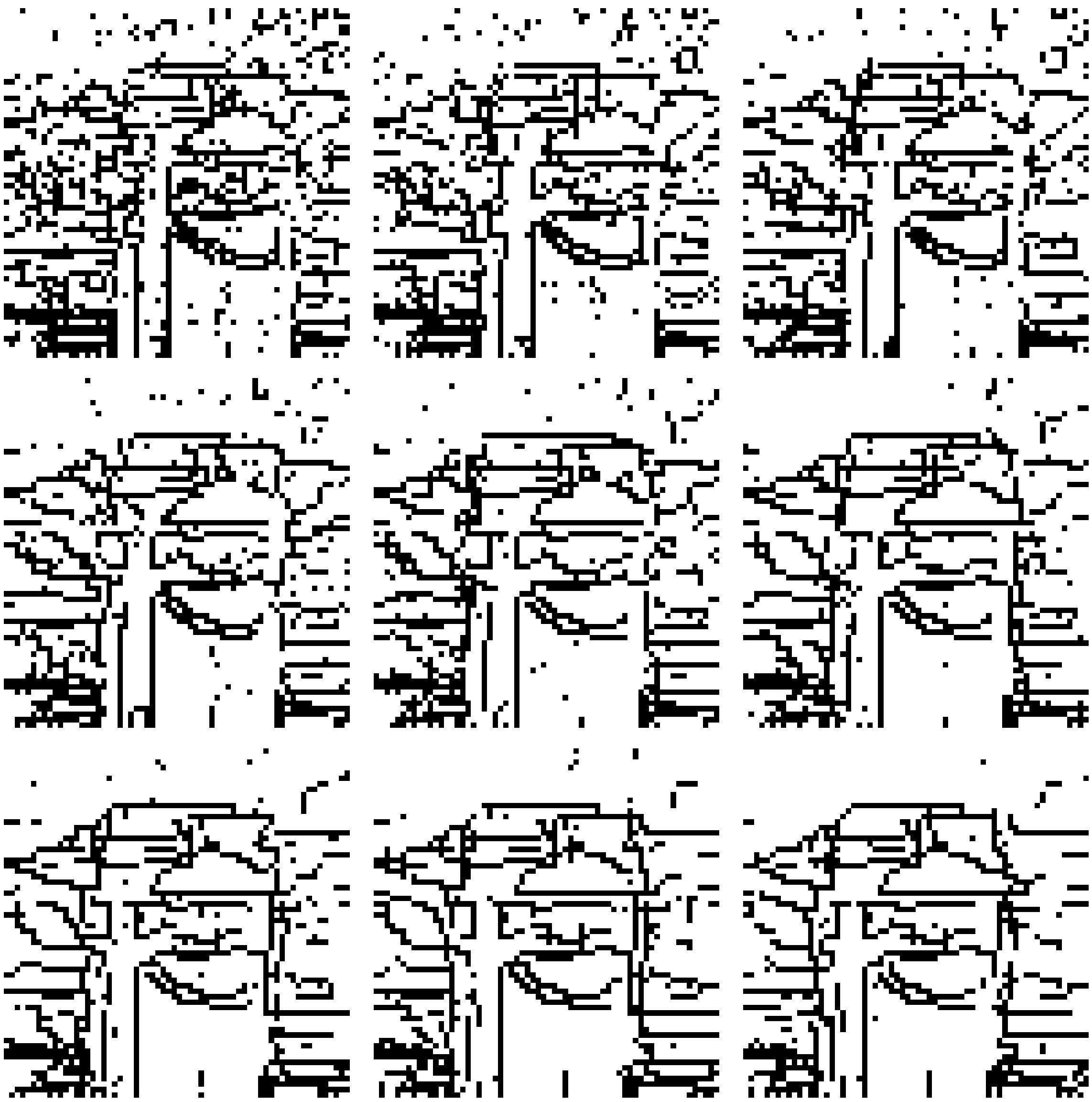
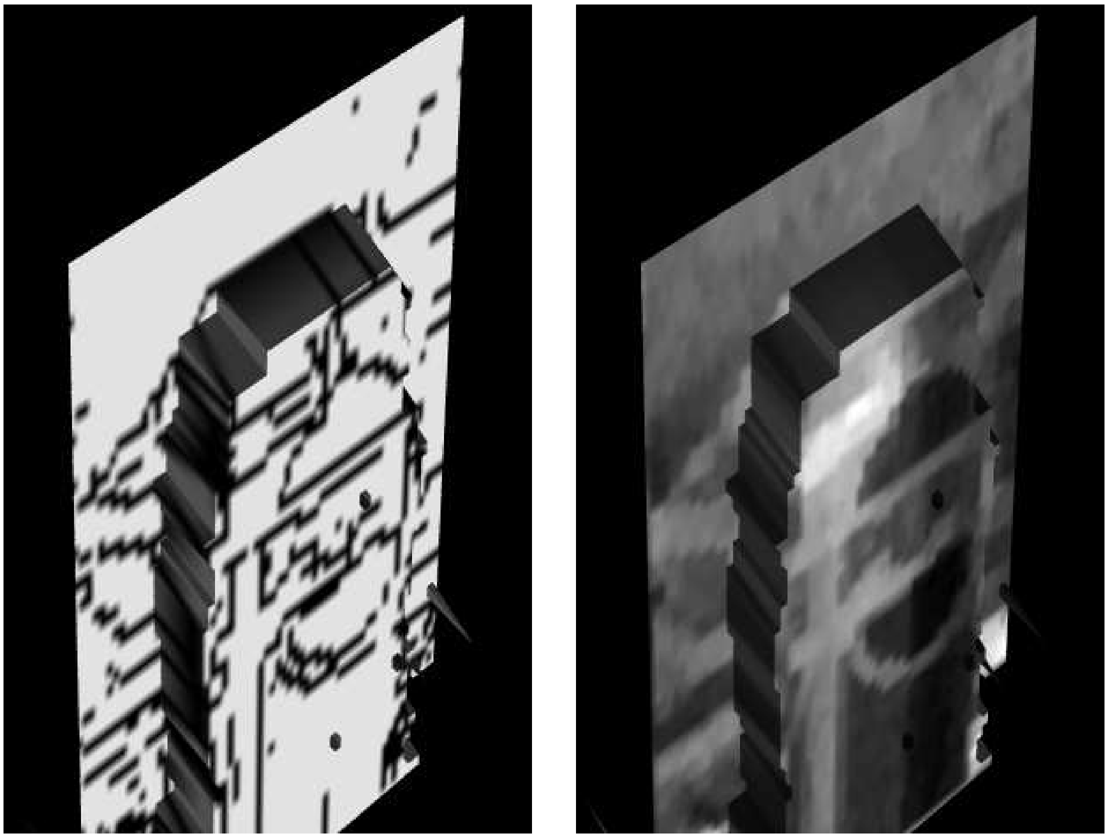


Figure 8.8: **Incremental Feature Extraction.** The images show the evolution (left to right, top to bottom) of features over a ten image sequence.



a *b*

Figure 8.9: Reconstructed views of the scene: *a)* intensity discontinuities, *b)* estimated intensity.

8.4.2 The Coke Sequence

The second image sequence contains 38 images of size 128×128 pixels. The camera is translating along the camera axis with the focus of expansion centered on the Coke can. Figures 8.10*a* and *b* show the first and last images in the sequence. Figure 8.10*c* shows the image features recovered at the end of the image sequence. Unlike standard boundary detection, these features have been tracked over the length of the sequence. Figure 8.10*d* shows only features that are classified as motion discontinuities and are, hence, likely to correspond to surface boundaries. The pencils and metal bracket are correctly interpreted

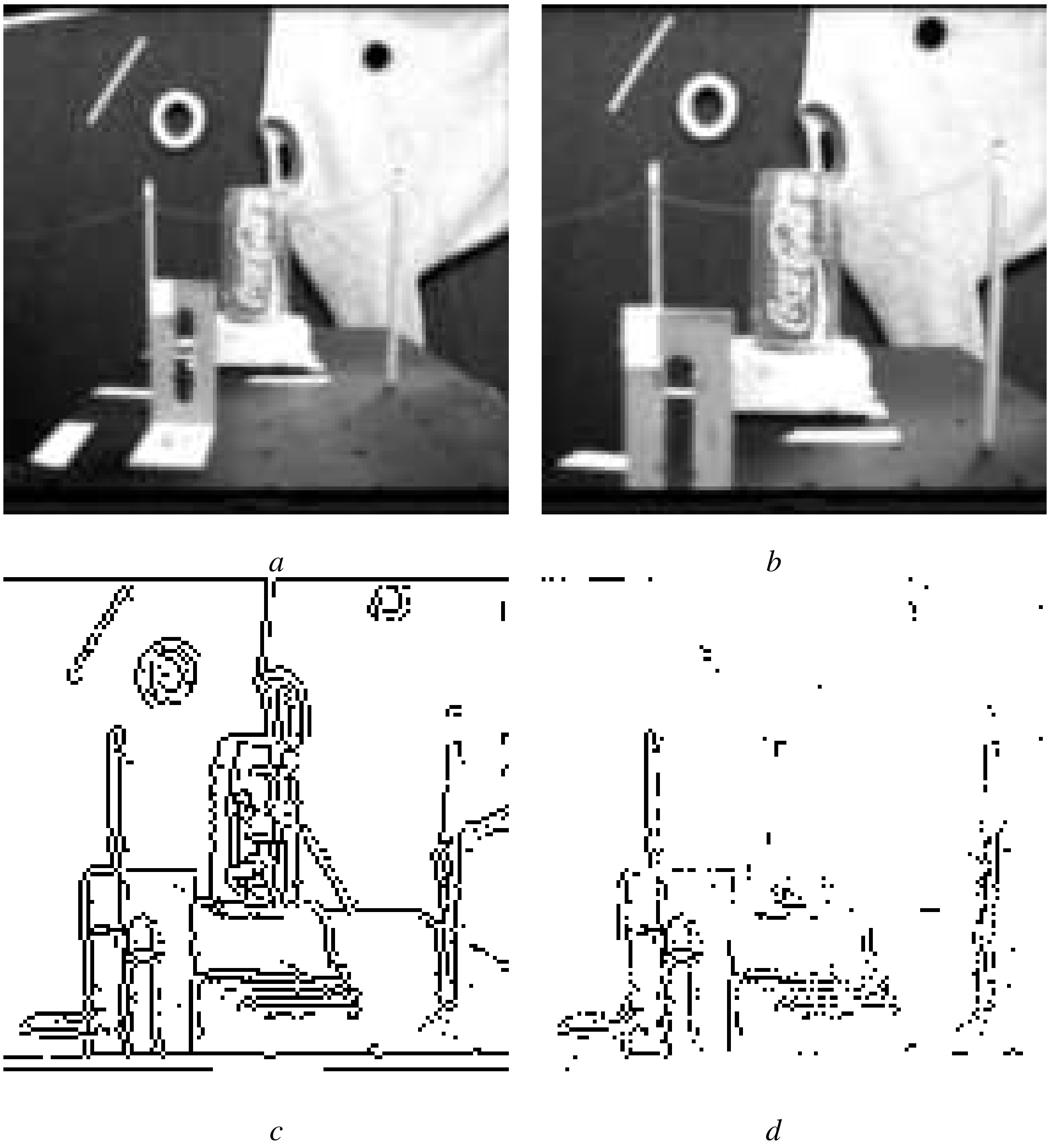


Figure 8.10: **The Coke Sequence.** Figures *a* and *b* are the first and last images in the sequence respectively. Figure *c* shows the image features recovered at the end of the sequence. Figure *d* shows only those features that are likely to have a physical interpretation.

as physically significant while the sweater is interpreted as purely surface marking.

The wire strung between the pencils has not been detected at all. The 128×128 pixel image sequence used here is a smoothed and subsampled version of the original images due to the memory limitations of the Connection Machine used to run the experiments. At the reduced resolution, the wire is not a salient perceptual feature.

Notice that the Coke-can boundary is incorrectly interpreted as surface marking. This is a result of small interframe displacements and the location of the can at the focus of expansion; the motion of the can boundary is not significant enough to classify it as structural with the current scheme. This suggests the need for a different classification scheme that takes into account the behavior of features over a longer time span.

Figure 8.11 shows the evolution of the image features over time. Five iterations of the annealing algorithm were used between frames with a processing time of approximately one minute per frame; approximately 58% of the computation was performed on the Connection Machine. The estimates improve as the features are tracked over the image sequence. Due to the relatively large homogeneous regions in the image, the dense motion estimates are poor. Accurate dense flow however is not required for incremental feature extraction. All that is required is that the motion estimates at the discontinuities be accurate.

8.5 Issues and Future Work

There are a number of issues to be addressed regarding the approach described. First, the current implementation employs only simple first order models of intensity and motion. While such a model may produce useful qualitative results in many situations, it is clearly not sufficient. In particular, to cope with textured surfaces more complicated image formation models will be required. There are a number of texture models formulated in terms of Markov random fields [Derin and Elliott, 1987; Geman *et al.*, 1990]. Our incremental minimization approach could readily be applied to these MRF texture models.

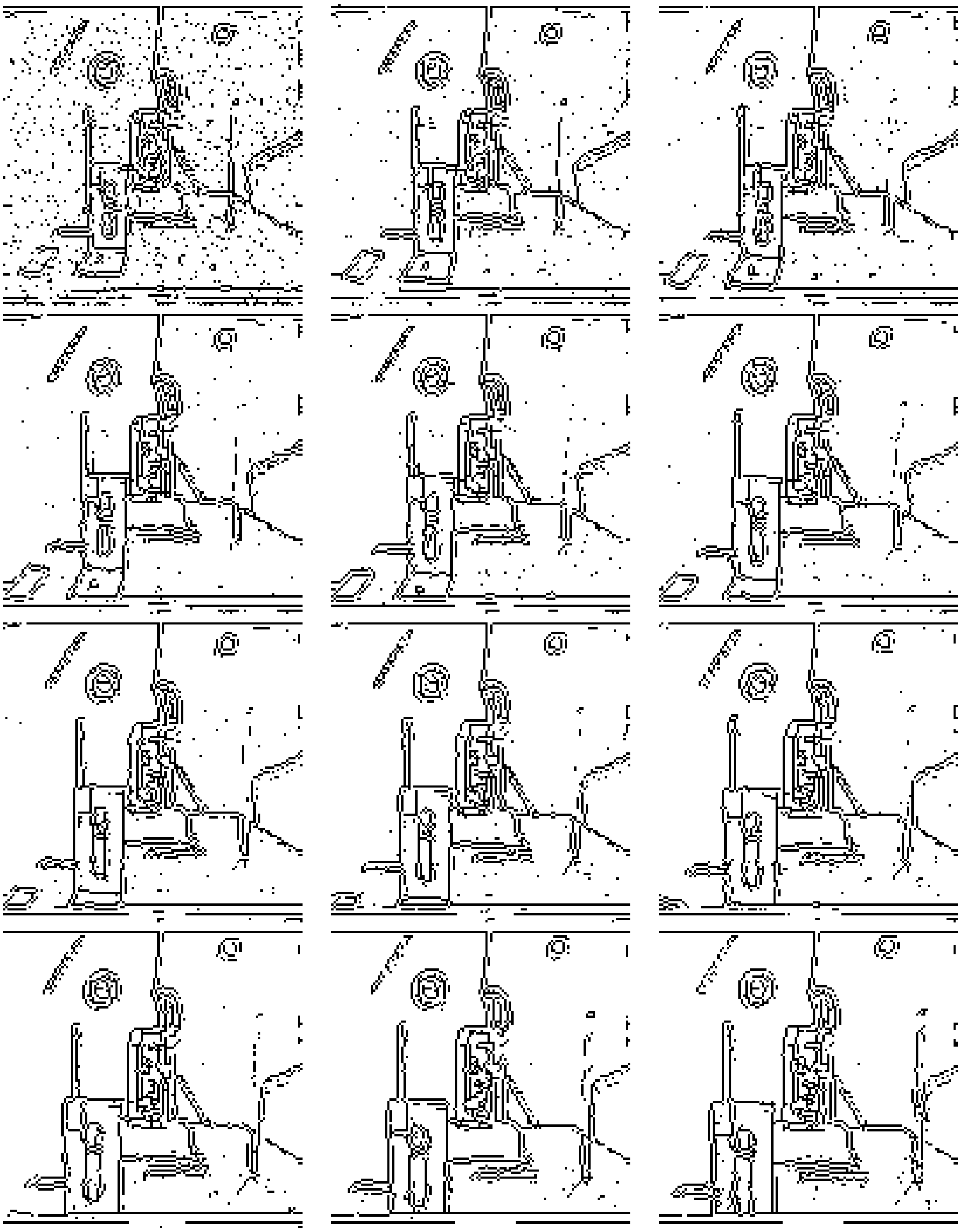


Figure 8.11: **Incremental Feature Extraction.** The sequence shows the evolution (left to right, top to bottom) of features at every third image in the 38 image sequence.

A second issue that must be addressed is one shared by many minimization approaches; that is the parameter estimation problem. The construction of an objective function with weights controlling the importance of the various terms is often based on intuition or empirical studies. The problem becomes more pronounced as the complexity of the model increases. In the model proposed here there are eight weights, ten clique energies, one scaling factor, an initial temperature, and a cooling rate that must be determined. Experiments with the current model indicate that it is relatively insensitive to changes in the parameters. The general problem, however, remains open.

Finally, the local optimization approach to recovering surface patches is only the first step in recovering the structure of the scene. If our goal is to recover environmental structure, or the *layout* of a scene [Gibson, 1979], then we must recover the *surfaces* present, their properties, and their relationships to each other. For this, more powerful surface models will be required; for example 3D parameterized surface models [Hung *et al.*, 1991; Hung *et al.*, 1988] or oriented particles [Szeliski and Tonnesen, 1991]. Non-local properties of the patches will need to be computed and additional perceptual organization processes will likely be needed to group patches that are consistent with the surface models.

The surface patch segmentation can be viewed as a coarse surface approximation. Given patches, we might formulate higher order models of surface properties; for example, second order smoothness of motion. Additionally, we might compute the depth, orientation, and curvature within each patch. These, and other, properties of patches might be used to organize them into surfaces.

Once surfaces have been extracted from the scene and their boundaries analyzed, inferences may be made about the relationships between the surfaces. In particular the relative depth, occlusion, or attachment relationships of surfaces may be determined. This in conjunction with the recovery of persistent properties (such as material, texture, shape, and reflectance) would constitute the layout of the scene.

8.6 Summary

In this chapter we have demonstrated the usefulness of the incremental minimization framework by applying it to the problem of feature extraction. We have presented an incremental approach for extracting stable perceptual features over time. The approach formulates a model of image regions in terms of constraints on intensity and motion while accounting for discontinuities. The incremental stochastic minimization scheme is used to recover boundaries over a sequence of images.

The approach has advantages over traditional feature extraction and motion estimation techniques. In particular, it is incremental and dynamic. This allows feature extraction and motion estimation to be performed over time, while reducing the amount of computation between frames and increasing veracity.

Additionally, the approach provides information about the structural properties of the scene. While intensity based segmentation alone provides information about the spatial structure of the image, motion provides information about object boundaries. Motion segmentation alone, however, provides fairly coarse information. Combining the two types of information provides a richer description of the scene.

Chapter 9

Conclusion

There have been two main themes running through this thesis. The first is the problem of robustness: “How can we remain insensitive to violations of our assumptions, particularly at motion discontinuities?” And the second is the issue of incremental processing: “How can we exploit and integrate information over time to improve the quality of optical flow estimates and reduce the computational overhead associated with computing them?”

These themes have led us to explore the use of robust statistical techniques and to develop a new framework for incremental estimation. This chapter will summarize the main contributions of the thesis and examine some open questions and future directions.

9.1 Contributions

9.1.1 Robust Estimation

1. We pointed out how motion discontinuities result in violations of the data coherence, spatial conservation, and temporal continuity assumptions and have shown how robust estimation techniques can be used to make the recovery of optical flow relatively insensitive to these violations.

2. A robust estimation framework for optical flow was developed. The approach provides a direct way to improve the performance of standard least-squares estimation.

3. The framework reduced the problem of smoothing across motion boundaries.

4. By detecting outliers, the location of motion discontinuities and violations of the data conservation constraint were recovered. These outliers provide useful information and may be used in further processing.

5. We introduced the notion of an outlier process as a generalization of the traditional line process and showed how the outlier-process formulation of the optical flow problem can be converted into an equivalent robust estimation problem. For certain robust estimators, we showed how the robust estimation problem can be expressed in terms of outlier processes.

6. The robust estimation framework was used to reformulate three common problems in optical flow. First, robust estimation was applied to area regression approaches, making them less sensitive to multiple motions. Next, peak detection in a correlation surface was enhanced by using the robust formulation. Finally, a robust gradient-based algorithm was developed for estimating dense flow fields.

7. Blake and Zisserman's Graduated Non-Convexity algorithm was generalized to efficiently minimize our non-convex robust estimation problem. A hierarchical version was also developed to estimate large motions.

8. An important aspect of our framework is that all the constraints are handled uniformly. This leads to the robust formulation of the data conservation term which results in improved flow estimates in the presence of noise.

9.1.2 Incremental Estimation

1. We introduced an explicit temporal continuity constraint, which embodies the assumption of constant image-plane acceleration.

2. An incremental estimation framework for minimizing non-convex objective functions over time was developed. The approach uses the constraint of temporal continuity to compensate for the effects image motion on the objective function. Motion estimates are always available and are refined over time. The approach is adaptive and amortizes the cost of minimization over the length of an image sequence.

3. To recover large motions over time, two hierarchical strategies were developed. The flow-through strategy can be viewed as a pyramid of spatiotemporally tuned motion detectors in which motion is combined across levels without refinement. An efficient coarse-to-fine approach to incremental estimation was also developed in which coarse estimates are propagated to the finer levels only when they disagree with the refined estimates.

4. The relationship between the incremental minimization framework and Kalman filtering was explored.

5. An incremental stochastic minimization algorithm was developed and its performance was illustrated on real and synthetic image sequences. The correlation-based approach is formulated as a Markov random field and the minimization is performed using a Gibbs sampler which has been extended to adaptively recover sub-pixel motion estimates.

6. An incremental version of the Graduated Non-Convexity algorithm was also implemented.

7. The algorithm was used to illustrate the effect of the temporal continuity constraint and to show how the incremental flow algorithm produces results similar to those observed in psychophysical studies of representational momentum.

8. The generality of our approach to incremental minimization was illustrated by for-

mulating feature extraction as objective function minimization and by extending the recovery over time. Features are extracted and tracked over the image sequence and are classified as structural properties or surface markings.

9.2 Open Questions

The work presented in this thesis advances the state of the art in optical flow estimation by providing new tools for approaching the problems of optical flow. This thesis has only begun to explore the issues of robustness and incremental estimation, and our endevour opens a number of avenues for further exploration.

Choice of Estimator

Although we have identified some criteria for choosing one robust estimator over another, the choice generally remains heuristic and specialized to the particular problem at hand. In some situations, the choice is dictated by external factors like the minimization framework (for example, the mean field function). In still other situations we may have a model for the kinds of outliers that can be expected and, when statistical models of outliers are available, they should be used.

Other Robust Techniques

The comparison of our formulation of robust estimation to other formulations of the problem remains an open issue. For instance, the least-median-of-squares approach has recently gained popularity in computer vision. The relative advantages of these other approaches have to be weighed against the simplicity of the robust estimation approach. An interesting approach, which we have not pursued, is the iteratively reweighted least-squares formulation. Such an approach might fit naturally within the recursive estimation framework.

Parameter Estimation

The estimation of model parameters is an ever-present problem which remains to be addressed in a systematic fashion. In some cases, we can automatically estimate parameters from the data. Where this is not possible, they have been determined empirically.

An interesting possibility of using "training" to address this problem was introduced in Chapter 7. There we had numerous "subjects", each of which had a slightly different parameter setting. By varying the settings it may be possible to match the performance of the algorithm to the performance of humans in a related task. It may also be possible to use neural-network algorithms to "learn" the appropriate parameter settings. The general problem, however, is shared by most minimization approaches and remains open.

Higher-Order Constraints

Our experiments have focused on first-order data and spatial constraints. There are a number of unanswered questions about how to best employ higher-order constraints, particularly in the context of incremental estimation. In the same way that the coarse-to-fine approach is inappropriate for incremental processing so is the most common approach to dealing with higher-level models in which a coarse, first-order, estimate is found first and then refined by applying higher-order models (see for example, [Geman and Reynolds, 1992]). In the case of incremental processing we want to refine our models over time and adaptively choose the appropriate constraints at a given instant in time.

With higher-order models, the number of parameters that need to be estimated increases. Unfortunately, the breakdown point of a robust estimator decreases as the number of model parameters increases [Li, 1985]. It is unclear what impact this fact will have in practice.

Translucency and Reflections

The robust formulation of the data conservation term deals with multiple motions which occur at surface boundaries, be they extended or fragmented. The extension to handle multiple motions resulting from translucency or reflection remains open.

Real-Time Flow

We have addressed the issue of efficiency in the context of computation reduction between frames but have not attempted to deal with the issues of real-time optical flow estimation. Hardware has been built which implements smoothness constraints with spatial discontinuities using functions similar to our robust estimators [Harris *et al.*, 1990; Hutchinson *et al.*, 1988; Koch *et al.*, 1988]. The implementation of robust data and temporal constraints in hardware remains an open issue.

Dynamic Environments

One of our goals for an incremental algorithm is that it be dynamic; that is, it should adapt to changes in the scene. The incremental minimization framework is adaptive in that patches of the world are tracked over time and, when the temporal constraint is violated at motion boundaries, the algorithm adapts by reseting the flow and control parameters.

This simple approach to controlling the algorithm may be improved. In general, the algorithm can be made more adaptive by relaxing the strict "annealing" schedule applied to the control parameters. For example, consider the ISM approach. In a loose sense, the temperature at each site reflects the uncertainty in the flow estimate at that site. Our current approach uses the heuristic that portions of the scene that are changing quickly, or have been recently disoccluded, have a higher temperature, while those portions of the scene that have been stably tracked over many frames have a lower temperature.

This idea can be further extended. For instance, in highly textured areas of the image where the motion corresponds to our assumptions, a faster annealing schedule might be appropriate. In other areas we may wish a slower schedule and, over time, if the motion does not correspond well to the temporal predictions, we may want to *increase* the temperature to reflect an increased uncertainty in the flow estimates.

Such an adaptive regime might be realized by using the *specific heat* of the system (see Kirkpatrick *et al.* [1983] for details). The result would be a self-regulatory system that is appropriate for dynamic applications.

We can define the local expected cost of the system at a temperature T:

$$\langle E(T)\rangle = \sum_{s \in \mathcal{G}_s} E(s)\Pi(s,T), \tag{9.1}$$

and the expected square cost as:

$$\langle E^2(T)\rangle = \sum_{s \in \mathcal{G}_s} E^2(s)\Pi(s,T). \tag{9.2}$$

The variance is then defined as the difference between the expected square cost and the square of the expected cost:

$$\sigma^2 = \langle E^2(T)\rangle - \langle E(T)\rangle^2. \tag{9.3}$$

The specific heat of the system, $C(T)$ is given by [Huang *et al.*, 1986]:

$$\begin{aligned} C(T) &= \frac{\partial}{\partial T}\langle E(T)\rangle \\ &= \frac{1}{T^2}\sigma^2(T) \\ &= \frac{1}{T^2}\langle E^2(T)\rangle - \langle E(T)\rangle^2. \end{aligned} \tag{9.4}$$

When $C(T)$ is large the system is at a critical point or *phase transition* [Laarhoven and Aarts, 1988]. In analogy to physical systems, a maximum in the specific heat of a fluid indicates the transition to a solid. In the case of simulated annealing, it indicates that the system is becoming frozen in a minimum and in order to avoid becoming trapped in a local minimum the system should be cooled more slowly at this point.

A number of authors have developed adaptive cooling schemes based on this notion [Huang *et al.*, 1986; Kirkpatrick *et al.*, 1983; Laarhoven and Aarts, 1988]. While these annealing schemes, based on specific heat, may be more efficient, they are still monotonic. Such schemes are not adaptive to unexpected scene changes; this is critical for dynamic estimation.

Confidence Measures

We have not specifically addressed the issue of assigning confidence measures to our flow vectors. In situations where the estimates are poor it may be useful to report the confidence in the flow estimate so that the information can be appropriately weighed by processes that rely on optical flow as input.

As mentioned above, the inverse temperature can be thought of as one measure of confidence. More interestingly, consider the adaptive state space scheme developed for continuous annealing. Given a flow estimate, we computed a covariance matrix, **S**, which represented the local shape of the error surface.

We can take the inverse of the minimum eigenvalue of **S** as a measure of confidence. Figure 9.1 shows this confidence measure for the SRI tree sequence at frame 42. Areas of low confidence appear as dark regions and correspond to the occluding and disoccluding tree branches and the lower left portion of the image which contains new image regions due to the rightward motion of the ground plane.

While this measure need further study, it shows promise. The confidence measures of Anandan [1989] only take into account the shape of the SSD surface. The approach here takes into account all three constraints by computing the shape of the objective function at the current estimate. This is more similar to the approach of Singh [1992a] who uses the covariance matrix of the Kalman filter to compute the confidence.

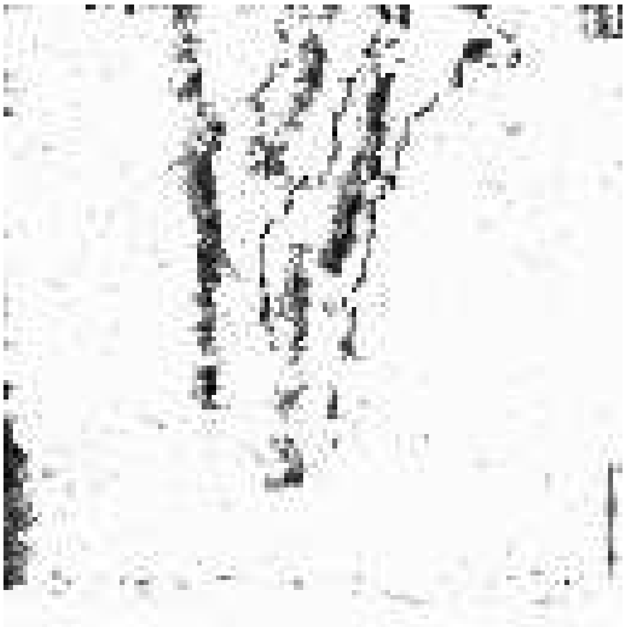


Figure 9.1: SRI Tree sequence, confidence measure.

Minimization Schemes

In this thesis we have explored two schemes for minimizing non-convex objective functions; one stochastic and the other deterministic. The stochastic approach provides a good way of exploring the energy landscape at a coarse scale without becoming trapped in local minima. It is not, however, an efficient scheme for exploring local minima which may be better explored with a gradient-based scheme.

The Graduated Non-Convexity algorithm overcomes this problem and allows the use of gradient-based schemes like simultaneous over-relaxation. The approach requires that a sequence of approximations to the objective function be defined. Blake and Zisserman define such a sequence when the truncated quadratic estimator is used for the spatial term. We defined a sequence of approximations using the Lorentzian estimator for the data and spatial constraints. For other estimators, developing such a sequence may not be as easy.

When neither of the above approaches is appropriate, we may be able to combine the two to derive a new algorithm which performs global changes stochastically and local changes deterministically. Such an approach is in the same spirit as work on *large-step Markov chains* [Martin *et al.*, 1991] for the Traveling Salesman Problem in which deterministic local search techniques are combined with stochastic sampling methods.

Accuracy

One of the main uses for dense optical flow has been the recovery of 3D structure in the form of a depth map.[1] Such a processes requires accurate flow measurements and the failure of structure from motion algorithms to robustly recover depth has been used to argue that the endevour is seriously flawed. Aloimonos, for example, has argued that flow errors of even 1% can destroy the computation of 3D structure and has further argued that "only multiple frame algorithms have the potential of leading to robust structure from motion modules" [Aloimonos, 1990, page 351].

The simple models presented here have been used to illustrate the issues surrounding, and our solutions to, the problems of robustness and temporal integration in the estimation of optical flow. For increased accuracy, more powerful models can readily be implemented in our framework. With higher-order models, the use of confidence measures, and integration over multiple frames, there is the real possibility for useful measurements about 3D structure.

9.3 Future Directions

In addition to the open questions above, there are a number of related problems that deserve future attention.

[1] The phrase "structure from motion" is unfortunate for it typically refers only to the recovery of dense depth from motion. Such a definition is too narrow, since motion discontinuities are also structural properties of the scene. And, there are numerous other structural properties like surface orientation and curvature that one might recover from motion information.

Local Intensity Models

There has been recent interest in using regression-based techniques with affine and quadratic flow models [Bergen *et al.*, 1992]. These approaches are typically applied to a large area of an image (or the entire image). Such an approach is valid in only a very restricted set of situations. In particular, the approaches are not valid when multiple motions are present. In such situations, a small number of parameters cannot describe the complex motion within the region. Hence, applying these region-based techniques alone produces flow fields that do not capture the structural information available at motion boundaries.

A local application of the affine model would be more appropriate. Unfortunately, small image regions may not contain enough information to reliably estimate the increased number of parameters. This means that a smoothness constraint will be needed.

In the robust correlation-based formulation of the optical flow problem we assumed a constant flow model which led to the use of correlation for the data term and to a first-order smoothness constraint. If we assume affine flow then we are led to a locally affine regression model for the data term and a second order smoothness constraint.

The problem of reliably recovering the affine parameters in a small region still remains. One possible approach is to use the flow estimates in a neighborhood to estimate the affine parameters. Recall the affine flow model:

$$\begin{aligned} u &= a_1 + a_2 dx + a_3 dy, \\ v &= a_4 + a_5 dx + a_5 dy. \end{aligned}$$

For a local motion estimate centered at a point we are only interested in estimating a_1 and a_4 since dx and dy at that point are zero. To do this accurately, we could compute the estimates for a_2, a_3, a_5, and a_6 from the neighborhood flow, and use them in the data conservation term. Holding the estimated parameters fixed, we can minimize at each site:

$$\min_{a_1, a_4} \sum_{\mathcal{R}} \rho(\nabla I^T \mathbf{u}(\mathbf{a}) + I_t) + \rho(\mathbf{u}(\mathbf{a})_{xx}) + 2\rho(\mathbf{u}(\mathbf{a})_{xy}) + \rho(\mathbf{u}(\mathbf{a})_{yy}),$$

where the subscripts xx, yy, and xy indicate second partial derivatives of the flow field. If the local image flow provides a good estimate of the affine parameters, this approach should result in more accurate, and reliable, sub-pixel motion estimates from the data conservation term.

Early Discontinuity Detection

We mentioned briefly how multiple peaks in the correlation surface could be used to detect motion boundaries before the estimation of a dense flow field. In [Black and Anandan, 1990a] we combined this idea with a spatial coherence constraint implemented by controlled continuity splines (or *snakes*) [Kass *et al.*, 1987]. We also described, but did not implement, a temporal continuity constraint.

This approach should be reconsidered in the light of the incremental minimization and Kalman filter frameworks. If discontinuities can be reliably detected at early stages of motion processing it could greatly simplify the later stages. For example, if a motion-based segmentation of the scene is available, then area regression techniques could be applied to each region independently. Least-squares techniques could then be used to rapidly, and reliably, recover the motion within these regions.

Recursive Estimation

We pointed out that there is a relationship between the incremental minimization framework and the Kalman filter, and that the Kalman filter can be viewed in the context of energy minimization. This points to a number of different research directions.

First, a quantitative comparison of the techniques needs to be performed. Such a comparison between Kalman filtering and incremental minimization should include their computational costs, sensitivity to model violations, accuracy, convergence rates, and performance at motion boundaries.

Second, the theoretical relationship between the approaches needs to be examined more fully. In particular, the iteratively reweighted least-squares formulation of the robust estimation problem may provide a link between the approaches. In the case where the statistics of the noise are unknown, the minimum variance estimation criterion of the Kalman filter can be dropped in favor of a robust weighted least-squares criterion. The filter derived for this robust objective function would likely be very similar to our incremental minimization approach.

The differences in the implementation of the approaches will likely prove less significant to performance than the temporal model employed. The exact form of this constraint is as crucial as how it is used, and we have just begun to study the problem.

It is also important to understand the relationships between these two approaches and other optimization schemes. There is a strong similarity between the two approaches we have described and techniques in physically-based modeling [Metaxas and Terzopoulos, 1991]; for example, deformable spline models [Kass *et al.*, 1987] attempt to minimize an objective function composed of an external data term, an internal smoothness term, and a temporal term related to physical models of momentum. As opposed to seeing these techniques as isolated algorithms it is important to understand the relationships between them and, hence, place them in the appropriate context.

Dynamic Vision

Vision is becoming more dynamic; the term *dynamic* has been used in many ways to describe current vision systems. For instance, Heel talks of “Dynamic Motion Vision” [Heel, 1989] to describe the incremental recovery of structure from motion. Burt *et al.* [1989] talk of *dynamic analysis* techniques for focusing processing resources to achieve real-time motion estimation and tracking. The field of active vision is dedicated to dynamic vision tasks and the interaction between “seeing” and “doing”.

These ideas are all important parts of a dynamic vision system. The goal of “dynamic vision”, however, can be stated more generally:

> Recover information about the world from a non-stationary sequence of images.

The active vision paradigm specializes this general goal to problems in which the camera is intentionally moved to gather more information. The purposive approach [Aloimonos, 1990] restricts the the goal further in recovering qualitative information that can be determined quickly and robustly.

The general statement of dynamic vision includes the problems of static computer vision extended over time. The previous chapter took a small step in this direction by extending feature extraction over an image sequence. More generally, the techniques for incremental motion estimation may provide the tools to begin extending other algorithms over time as well. This should be a promising area for research as the demand for real-time image processing on mobile platforms increases.

9.4 Discussion

Our two themes of robustness and temporal integration are driven by a more basic goal. That is the desire to make the recovery of optical flow practical, fast, and accurate. This thesis has developed new tools which we believe will bring us closer to this goal by addressing two of the crucial issues in flow estimation. Are we on the threshold of a new era in motion estimation? And if so, what does that future promise?

A number of recent advances, in addition to our own, hold out promise for the future of motion estimation. In particular, advances in real-time flow computation are making applications of optical flow realizable. The possibilities for practical real-time motion estimation have been improved by research on visual attention and selective processing. The result of this trend is that applications of visual motion are being made in the contexts of robotics, communications, and home entertainment. The approaches to robustness and temporal integration presented in this thesis are complementary with work on real-time estimation and will improve the reliability and accuracy of these systems.

The problem of optical flow, however, is not solved and there are still many important issues to be addressed, some of which have been described above. The open questions and future directions above are primarily concerned with early stages of visual processing, for this is where the majority of the emphasis, and the success, has been.

There is a largely unexplored world that exists between our current models of motion and the cognitive domain of representation, reasoning, and awareness. The current active vision systems are analogous to humans who experience “blindsight” as the result of damage to their visual cortex. These patients have no conscious sight, but they can locate and point to visual targets and can even track moving objects with their eyes. To move beyond this point will require the integration of motion, and vision in general, with the traditional domains of artificial intelligence. Given our currently impoverished theories of knowledge, action, and reasoning, researchers may still be working on the recovery and interpretation of visual motion at the Greek calends.

In this gap between our human awareness of motion, and our current theories of optical flow, lies the essential conundrum of motion estimation. Detecting and representing motion is fundamental to human perception for it allows us to understand a changing world and discern its structure. It is a sea of motion in which we are immersed, and we wish to provide our robots with an awareness of this predictably protean world.

Bibliography

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

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

[Aggarwal and Nandhakumar, 1988] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images – a review. *Proceedings of the IEEE*, 76(8):917–935, August 1988.

[Aloimonos and Duriç, 1992] Y. Aloimonos and Z. Duriç. Active egomotion estimation: A qualitative approach. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 497–510. Springer-Verlag, May 1992.

[Aloimonos and Rosenfeld, 1991] Y. Aloimonos and A. Rosenfeld. A response to "Ignorance, myopia, and naivete in computer vision systems" by R. C. Jain and T. O. Binford. *CVGIP: Image Understanding*, 53(1):120–124, 1991.

[Aloimonos *et al.*, 1987] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. In *Proceedings of the First International Conference on Computer Vision*, pages 35–54, London, England, June 1987. IEEE Computer Society Press, Los Alamitos, California.

[Aloimonos, 1990] J. Aloimonos. Purposive and qualitative active vision. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 346–360, Atlantic City, NJ, June 1990.

[Anandan, 1984] P. Anandan. Computing dense displacement fields with confidence measures in scenes containing occlusion. *SPIE Intelligent Robots and Computer Vision*, 521, 1984.

[Anandan, 1987a] P. Anandan. *Measuring visual motion from image sequences*. PhD thesis, University of Massachusetts, Amherst, 1987. COINS TR 87-21.

[Anandan, 1987b] P. Anandan. A unified perspective on computational techniques for the measurement of visual motion. In *Proc. First Int. Conf. on Computer Vision, ICCV-87*, pages 219–230, London, England, June 1987.

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

[Ancona, 1992] N. Ancona. A fast obstacle detection method based on optical flow. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 267–271. Springer-Verlag, May 1992.

[Baker and Braddick, 1982] C. L. Baker and O. J. Braddick. Does segregation of differently moving areas depend on relative or absolute displacement? *Vision Research*, 7:851–856, 1982.

[Baker, 1988] H. H. Baker. Surface reconstruction from image sequences. In *Proc. Second Int. Conf. on Computer Vision*, pages 334–343, Tampa, Florida, December 1988.

[Baker, 1989] H. H. Baker. Building surfaces of evolution: The weaving wall. *International Journal of Computer Vision*, 3:51–71, 1989.

[Ballard, 1987] D. H. Ballard. Interpolation coding: A representation for numbers in neural models. *Biol. Cybern.*, 57:389–402, 1987.

[Ballard, 1989] D. H. Ballard. Reference frames for animate vision. In *Proc. IJCAI-89*, pages 1635–1641, Detroit, Michigan, 1989.

[Barnard, 1989] S. T. Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3:17–32, 1989.

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

[Battiti *et al.*, 1991] R. Battiti, E. Amaldi, and C. Koch. Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy. *International Journal of Computer Vision*, 6(2):133–145, 1991.

[Beaton and Tukey, 1974] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:147–185, 1974.

[Beaudet, 1978] P. R. Beaudet. Rotationally invariant image operators. In *Proc. IEEE Int. Conf. on Pattern Recognition*, pages 377–384, 1978.

[Bergen *et al.*, 1990a] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proc. Int. Conf. on Comp. Vision, ICCV-90*, pages 27–30, Osaka, Japan, December 1990.

[Bergen *et al.*, 1990b] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. Multiple component image motion: Motion estimation. Technical report, David Sarnoff Research Center, January 1990.

[Bergen *et al.*, 1992] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 237–252. Springer-Verlag, May 1992.

[Bertero *et al.*, 1988] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, August 1988.

[Besl *et al.*, 1988] P. J. Besl, J. B. Birch, and L. T. Watson. Robust window operators. In *Proc. Int. Conf. on Comp. Vision, ICCV-88*, pages 591–600, 1988.

[Bilbro and Snyder, 1988] G. Bilbro and W. Snyder. Image restoration by mean field annealing. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 594–601. Morgan Kaufman, San Mateo, CA, 1988.

[Bilbro *et al.*, 1988] G. Bilbro, W. Snyder, R. Mann, D. Van den Bout, T. Miller, and M. White. Optimization by mean field annealing. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 91–98. Morgan Kaufman, San Mateo, CA, 1988.

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

[Black and Anandan, 1990b] M. J. Black and P. Anandan. A model for the detection of motion over time. In *Proc. Int. Conf. on Computer Vision, ICCV-90*, pages 33–37, Osaka, Japan, December 1990.

[Black and Anandan, 1991a] M. J. Black and P. Anandan. Dynamic motion estimation and feature extraction over long image sequences. In *Proc. IJCAI Workshop on Dynamic Scene Understanding*, Sydney, Australia, August 1991.

[Black and Anandan, 1991b] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. Computer Vision and Pattern Recognition, CVPR-91*, pages 296–302, Maui, Hawaii, June 1991.

[Black, 1992a] M. J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 485–493. Springer-Verlag, May 1992.

[Black, 1992b] M. J. Black. A robust gradient method for determining optical flow. Technical Report YALEU/DCS/RR-891, Yale University, February 1992.

[Blake and Zisserman, 1987] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, Massachusetts, 1987.

[Blauer, 1991] M. Blauer. Image smoothing with shape invariance and L_1 curvature constraints. In *Proceedings SPIE*, Boston, Mass, November 1991.

[Bolles *et al.*, 1987] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *Internation Journal of Computer Vision*, 1(1):7–57, 1987.

[Bouthemy and Lalande, 1990] P. Bouthemy and P. Lalande. Detection and tracking of moving objects based on a statistical regularization method in space and time. In *Proc. First European Conf. on Computer Vision, ECCV-90*, pages 307–311, Antibes, France, April 1990.

[Bouthemy and Rivero, 1987] P. Bouthemy and J. S. Rivero. A hierarchical likelihood approach for region segmentation according to motion-based criteria. In *Proc. First Int. Conf. on Computer Vision, ICCV-87*, pages 463–467, London, England, June 1987.

[Burt and Adelson, 1983] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-34(4):532–540, 1983.

[Burt *et al.*, 1982] P. J. Burt, C. Yen, and X. Xu. Local correlation measures for motion analysis: A comparative study. *IEEE Proc. PRIP*, pages 269–274, 1982.

[Burt *et al.*, 1989] P. J. Burt, J. R. Bergen, R. Hingorani, R. Kolczynski, W. A. Lee, A. Leung, J. Lubin, and H. Shvaytser. Object tracking with a moving camera: An application of dynamic motion analysis. In *Proceedings of the Workshop on Visual Motion*, pages 2–12, Irvine, CA, March 1989.

[Campbell, 1980] N. E. Campbell. Robust procedures in multivariate analysis I: Robust covariance estimation. *Appl. Statist.*, 29(3):231–237, 1980.

[Chen and Schunck, 1990] D. S. Chen and B. G. Schunck. Robust statistical methods for building classification procedures. In *Proc. Int. Workshop on Robust Computer Vision*, pages 72–85, Seattle, WA, October 1990.

[Chou and Brown, 1990] P. B. Chou and C. M. Brown. The theory and practice of Bayesian image labeling. *Int. Journal of Computer Vision*, 4(3):185–210, 1990.

[Cipolla and Blake, 1992] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 187–202. Springer-Verlag, May 1992.

[Cohen and Nguyen, 1988] P. Cohen and H. H. Nguyen. Unsupervised Bayesian estimation for segmenting textured images. In *Proc. Second Int. Conf. on Computer Vision*, pages 303–309, Tampa, Florida, December 1988.

[Cooper, 1976] L. A. Cooper. Demonstration of a mental analog of an external rotation. *Perception & Psychophysics*, 19:296–302, 1976.

[Cornelius and Kanade, 1983] N. Cornelius and T. Kanade. Adapting optical flow to measure object motion in reflectance and X-ray image sequences. In *Proc. ACM Siggraph/Sigart Interdisciplinary Workshop on Motion: Representation and Perception*, pages 50–58, Toronto, Ont., Canada, April 1983.

[Darrell and Pentland, 1991] T. Darrell and A. Pentland. Discontinuity models and multilayer description networks. Technical Report 162, M.I.T. Media Lab Vision and Modeling Group, May 1991.

[Derin and Elliott, 1987] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):39–55, January 1987.

[Dubes *et al.*, 1990] R. C. Dubes, A. K. Jain, S. G. Nadabar, and C. C. Chen. MRF model-based algorithms for image segmentation. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 808–814, Atlantic City, NJ, June 1990.

[Durrant-Whyte, 1987] H. F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *International Journal of Robotics Research*, 6(3):3–24, 1987.

[Eero and David, 1992] S. Eero and H. David. Vision and dynamic control for automated milk frothing: An analysis of what went wrong. *Cappuccino Quarterly*, pages 384–375, April 1 1992.

[Enkelmann, 1986] W. Enkelmann. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. In *Workshop on Motion: Representation and Analysis*, pages 81–87, Charlston, SC, September 1986.

[Faugeras *et al.*, 1987] O. D. Faugeras, F. Lustman, and G. Tocani. Motion and structure form point and line matches. In *Proc. IEEE First Int. Conf. on Computer Vision, ICCV-87*, pages 25–34, London, England, June 1987.

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

[Finke and Freyd, 1985] R. A. Finke and J. J. Freyd. Transformations of visual memory induced by implied motions of pattern elements. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(4):780–794, 1985.

[François and Bouthemy, 1991] E. François and P. Bouthemy. Multiframe-based identification of mobile components of a scene with a moving camera. In *Proc. Comp. Vision and Pattern Recognition, CVPR-91*, pages 166–172, Maui, Hawaii, June 1991.

[Freyd and Finke, 1985] J. J. Freyd and R. A. Finke. A velocity effect for representational momentum. *Bulletin of the Psychonomic Society*, 23(6):443–446, 1985.

[Freyd, 1983] J. J. Freyd. *Apparent accelerated motion and dynamic mental representations*. PhD thesis, Stanford University, Stanford, CA, 1983.

[Freyd, 1987] J. J. Freyd. Dynamic mental representations. *Psychological Review*, 94(4):427–438, 1987.

[Gamble and Poggio, 1987] E. Gamble and T. Poggio. Integration of intensity edges with stereo and motion. Technical Report Artificial Intelligence Lab Memo No. 970, MIT, 1987.

[Geiger and Girosi, 1991] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRF's: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5), May 1991.

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

[Geman and Geman, 1984] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

[Geman and McClure, 1987] S. Geman and D. E. McClure. Statistical methods for tomographic image reconstruction. In *Proceedings of the 46th Session of the ISI, Bulletin of the ISI*, 1987.

[Geman and Reynolds, 1992] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):376–383, March 1992.

[Geman *et al.*, 1990] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):609–628, July 1990.

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

[Glazer, 1987] F. Glazer. *Hierarchical motion detection*. PhD thesis, University of Massachusetts, Amherst, MA, 1987. COINS TR 87–02.

[Hampel *et al.*, 1986] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, New York, NY, 1986.

[Harris *et al.*, 1990] J. G. Harris, C. Koch, E. Staats, and J. Luo. Analog hardware for detecting discontinuities in early vision. *Int. Journal of Comp. Vision*, 4(3):211–223, June 1990.

[Heeger and Jepson, 1990] D. J. Heeger and A. Jepson. Simple method for computing 3D motion and depth. In *Proc. Int. Conf. on Comp. Vision, ICCV-90*, pages 96–100, Osaka, Japan, December 1990.

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

[Heel, 1989] J. Heel. Dynamic motion vision. In *Proc. Image Understanding Workshop*, pages 701–713, Palo Alto, CA, May 1989.

[Heel, 1990] J. Heel. Temporally integrated surface reconstruction. In *Proc. IEEE Int. Conf. on Comp. Vision, ICCV-90*, pages 292–295, Osaka, Japan, December 1990.

[Heel, 1991] J. Heel. Temporal surface reconstruction. In *Proc. IEEE Comp. Vision and Pattern Recognition, CVPR-91*, pages 607–612, Maui, Hawaii, June 1991.

[Heitz and Bouthemy, 1990] F. Heitz and P. Bouthemy. Multimodal motion estimation and segmentation using Markov random fields. In *Proc. IEEE Int. Conf. on Pattern Recognition*, pages 378–383, June 1990.

[Hildreth, 1983] E. C. Hildreth. Computing velocity fields along contours. In *Proc. ACM Siggraph/Sigart Interdisciplinary Workshop on Motion: Representation and Perception*, pages 26–32, Toronto, Ont., Canada, April 1983.

[Hildreth, 1984] E. C. Hildreth. *The Measurement of Visual Motion*. MIT Press, Cambridge, Mass., 1984.

[Hillis, 1985] W. D. Hillis. *The Connection Machine*. The MIT Press, Cambridge, Massachusetts, 1985.

[Horn and Schunck, 1981] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, August 1981.

[Horn and Weldon, 1988] B. K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *Int. Journal of Computer Vision*, 2(1):51–76, June 1988.

[Horn, 1986] B. K. P. Horn. *Robot Vision*. The MIT Press, Cambridge, Massachusetts, 1986.

[Huang *et al.*, 1986] M. D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli. An efficient general cooling schedule for simulated annealing. In *Proc. IEEE Int. Conf. on Computer-Aided Design*, pages 381–384, November 1986.

[Huber, 1981] P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, NY, 1981.

[Hung *et al.*, 1988] Y. Hung, D. B. Cooper, and B. Cernuschi-Frias. Bayesian estimation of 3-D surfaces from a sequence of images. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 906–911, April 1988.

[Hung *et al.*, 1991] Y. Hung, D. B. Cooper, and B. Cernuschi-Frias. Asymptotic Bayesian surface estimation using an image sequence. *International Journal of Computer Vision*, 2(6):105–132, 1991.

[Hutchinson *et al.*, 1988] J. Hutchinson, C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computer*, pages 52–63, March 1988.

[Inoue *et al.*, 1992] H. Inoue, T. Tachikawa, and M. Inaba. Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1621–1626, May 1992.

[Irani *et al.*, 1992] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 282–287. Springer-Verlag, May 1992.

[Kahn, 1985] P. Kahn. Local determination of a moving contrast edge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(4):402–409, July 1985.

[Kahn, 1988] P. Kahn. Integrating moving edge information along a 2D trajectory in densely sampled imagery. In *IEEE Proc. Comp. Vision and Pattern Recognition*, pages 702–709, June 1988.

[Kanade and Okutomi, 1990] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. Technical Report CMU-CS-90-120, Carnegie Mellon University, April 1990.

[Kass *et al.*, 1987] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. First International Conference on Computer Vision*, pages 259–268, June 1987.

[Kelly and Freyd, 1987] M. H. Kelly and J. J. Freyd. Explorations of representational momentum. *Cognitive Psychology*, 19:369–401, 1987.

[Kirkpatrick *et al.*, 1983] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[Koch *et al.*, 1988] C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computer*, pages 52–63, March 1988.

[Koenderink and van Doorn, 1975] J. J. Koenderink and A. J. van Doorn. 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.

[Koffka, 1935] K. Koffka. *Pinciples of Gestalt Psychology*. Harcourt, Brace and World, New York, 1935.

[Konrad and Dubois, 1988] J. Konrad and E. Dubois. Miltigrid Bayesian estimation of image motion fields using stochastic relaxation. In *Int. Conf. on Computer Vision*, pages 354–362, 1988.

[Konrad, 1989] J. Konrad. *Bayesian estimation of motion fields from image sequences*. PhD thesis, McGill University, Montreal, Canada, June 1989.

[Kumar and Hanson, 1990] R. Kumar and A. R. Hanson. Analysis of different robust methods for pose refinement. In *Proc. Int. Workshop on Robust Computer Vision*, pages 167–182, Seattle, WA, October 1990.

[Laarhoven and Aarts, 1988] P. J. M. Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel Pub. Co., Dordrecht, Holland, 1988.

[Lawton, 1983] D. T. Lawton. Processing translational motion sequences. *Computer Vision Graphics and Image Processing*, 22:116–144, 1983.

[Leclerc, 1989] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102, 1989.

[Li, 1985] G. Li. Robust regression. In F. Mosteller and J. W. Tukey, editors, *Exploring Data, Tables, Trends and Shapes*. John Wiley & Sons, New York, 1985.

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

[Lowe, 1985] D. Lowe. *Perceptual organization and visual recognition.* Kluwer Academic Pub., Boston, MA, 1985.

[Lucas and Kanade, 1981] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th IJCAI*, pages 674–679, Vancouver, B. C., Canada, 1981.

[Lui *et al.*, 1990] L. Lui, B. G. Schunck, and C. C. Meyer. On robust edge detection. In *Proc. Int. Workshop on Robust Computer Vision*, pages 261–286, Seattle, WA, October 1990.

[Luo and Maître, 1990] W. Luo and H. Maître. Using surface model to correct and fit disparity data in stereo vision. In *Proc. IEEE Int. Conf. on Pattern Recognition*, pages 60–64, June 1990.

[Lynch, 1960] K. Lynch. *The Image of the City.* The MIT Press, Cambridge, Massachusetts, 1960.

[Marr, 1982] D. Marr. *Vision.* W. H. Freeman and Company, New York, NY, 1982.

[Marroquin *et al.*, 1987] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. of the American Statistical Assoc.*, 82(397):76–89, March 1987.

[Marroquin, 1987] J. L. Marroquin. Deterministic Bayesian estimation of Markovian random fields with applications to computational vision. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 597–601, London, England, June 1987.

[Martin *et al.*, 1991] O. Martin, S. W. Otto, and E. W. Felten. Large-step Markov chains for the Traveling Salesman Problem. *Complex Systems*, 5(3):299–326, 1991.

[Matthies *et al.*, 1989] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–236, September 1989.

[Meer *et al.*, 1990] P. Meer, D. Mintz, and A. Rosenfeld. Robust recovery of piecewise polynomial image structure. In *Proc. Int. Workshop on Robust Computer Vision*, pages 109–126, Seattle, WA, October 1990.

[Meer *et al.*, 1991] P. Meer, D. Mintz, and A. Rosenfeld. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.

[Metaxas and Terzopoulos, 1991] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. In *Proc. Computer Vision and Pattern Recognition, CVPR-91*, pages 337–343, Maui, Hawaii, June 1991.

[Metropolis *et al.*, 1953] N. Metropolis, A. Rosenbluth, A. Teller M. Rosenbluth, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

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

[Murray and Buxton, 1990] D. W. Murray and B. F. Buxton. *Experiments in the Machine Interpretation of Visual Motion*. The MIT Press, Cambridge, Massachusetts, 1990.

[Murray *et al.*, 1986] D. W. Murray, A. Kashko, and H. Buxton. A parallel approach to the picture restoration algorithm of Geman and Geman on a SIMD machine. *Image and Vision Computing*, 4(3):133–142, August 1986.

[Mutch and Thompson, 1988] K. M. Mutch and W. B. Thompson. Analysis of accretion and deletion at boundaries in dynamic scenes. In W. Richards, editor, *Natural Computation*, pages 44–54. The MIT Press, Cambridge, Mass., 1988.

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

[Nagel, 1983a] H. H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *IJCAI*, pages 945–951, Karlsruhe, West Germany, August 1983.

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

[Nagel, 1987] H. H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33(3):299–324, November 1987.

[Navab *et al.*, 1990] N. Navab, R. Deriche, and O. D. Faugeras. Recovering 3D motion and structure from stereo and 2D token tracking cooperation. In *Proc. Int. Conf. on Comp. Vision, ICCV-90*, pages 513–516, Osaka, Japan, December 1990.

[Nelson and Aloimonos, 1989] R. C. Nelson and J. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(10):1102–1106, 1989.

[Nishihara, 1984] H. K. Nishihara. Practical real-time imaging stereo matcher. *Optical Engineering*, 23(5):536–545, 1984.

[Okutomi and Kanade, 1992] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7(2):143–162, January 1992.

[Papanikolopoulos and Khosla, 1991] N. P. Papanikolopoulos and P. K. Khosla. Feature based robotic visual tracking of 3-D translational motion. In *Proc. IEEE Conf. on Decision and Control*, December 1991.

[Peleg and Rom, 1990] S. Peleg and H. Rom. Motion based segmentation. In *Proc. IEEE Int. Conf. on Pattern Recognition*, pages 109–113, June 1990.

[Perona and Malik, 1990] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[Poggio *et al.*, 1985] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(26):314–319, September 1985.

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

[Prager and Arbib, 1983] J. M. Prager and M. A. Arbib. Computing the optic flow: The MATCH algorithm and prediction. *Computer Vision, Graphics and Image Processing*, 24:271–304, 1983.

[Pratt, 1979] W. K. Pratt. *Image Transmission Techniques*. Academic Press, Inc., New York, 1979.

[Press *et al.*, 1988] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1988.

[Rangarajan and Chellapa] A. Rangarajan and R. Chellapa. A continuation method for image estimation and segmentation. submitted, IEEE PAMI.

[Rangarajan and Chellappa, 1990] A. Rangarajan and R. Chellappa. Generalized graduated non-convexity algorithm for maximum a posteriori image estimation. In *Proc. 10th Int. Conf. on Pattern Recognition*, Atlantic City, NJ, June 1990.

[Rangarajan and Chellappa, 1991] A. Rangarajan and R. Chellappa. Image estimation and segmentation using a continuation method. In *Proc. ICASSP '91, IEEE Conf. on Acoust., Speech, and Signal Processing*, 1991.

[Rockafellar, 1970] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[Rousseeuw and Leroy, 1987] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, 1987.

[Schunck, 1983] B. G. Schunck. *Motion segmentation and estimation*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 1983.

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

[Schunck, 1989b] B. G. Schunck. Robust estimation of image flow. In *Proceedings SPIE*, November 1989.

[Schunck, 1990] B. G. Schunck. Robust computational vision. In *Proc. Int. Workshop on Robust Computer Vision*, pages 1–18, Seattle, WA, October 1990.

[Shizawa and Mase, 1991] M. Shizawa and K. Mase. Principle of superposition: A common computational framework for analysis of multiple motion. In *Proc. IEEE Workshop on Visual Motion*, pages 164–172, Princeton, NJ, October 1991.

[Simoncelli and Adelson, 1991] E. P. Simoncelli and E. H. Adelson. Computation of optical flow: Relationship between several standard techniques. Vision and Modeling Technical Report 165, MIT Media Laboratory, March 1991.

[Simoncelli and Heeger, 1991] E. P. Simoncelli and D. J. Heeger. Relationship between gradient, spatio-temporal energy, and regression models for motion perception. In *Investigative Opthamology and Visual Science Supplement*, volume 32, March 1991.

[Simoncelli *et al.*, 1991] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In *Proc. Computer Vision and Pattern Recognition, CVPR-91*, pages 310–315, Maui, Hawaii, June 1991.

[Singh, 1990] A. Singh. An estimation-theoretic framework for image-flow computation. In *Proc. Int. Conf. on Comp. Vision, ICCV-90*, pages 168–177, Osaka, Japan, 1990.

[Singh, 1991] A. Singh. Incremental estimation of image-flow using a Kalman filter. In *Proc. IEEE Workshop on Visual Motion*, pages 36–43, Princeton, NJ, October 1991.

[Singh, 1992a] A. Singh. Incremental estimation of image flow using a Kalman filter. *J. of Visual Communication and Image Representation*, 3(1):39–57, March 1992.

[Singh, 1992b] A. Singh. *Optic Flow Computation: A Unified Perspective*. IEEE Computer Society Press, 1992.

[Sinha and Schunck, 1992] S. S. Sinha and B. G. Schunck. A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):36–55, January 1992.

[Snyder, 1991] M. A. Snyder. On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1105–1114, November 1991.

[Spoerri and Ullman, 1987] A. Spoerri and S. Ullman. The early detection of motion boundaries. In *Proc. 1st ICCV*, pages 209–218, London, UK, June 1987.

[Strang, 1976] G. Strang. *Linear Algebra and its Applications*. Academic Press, New York, 1976.

[Swain *et al.*, 1990] M. J. Swain, L. E. Wixson, and P. B. Chou. Efficient parallel estimation for Markov random fields. In M. Henrion, R. D. Shachter, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 407–419. Elsevier Science Publishers B. V., North-Holland, 1990.

[Szeliski and Tonnesen, 1991] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. Technical Report CRL 91/14, Digital Equipment Corporation, Cambridge Research Lab, December 1991.

[Szeliski, 1988] R. S. Szeliski. *Bayesian modeling of uncertainty in low-level vision*. PhD thesis, Carnegie Mellon University, 1988.

[Szeliski, 1991] R. Szeliski. Shape from rotation. In *Proc. IEEE Comp. Vision and Pattern Recognition, CVPR-91*, pages 625–630, Maui, Hawaii, June 1991.

[Tarr and Black, 1991] M. J. Tarr and M. J. Black. A computational and evolutionary perspective on the role of representation in computer vision. Technical Report YALEU/DCS/RR-899, Yale University, October 1991.

[Tarr and Black, 1992] M. J. Tarr and M. J. Black. Psychophysical implications of temporal persistence in early vision: A computational account of representational momentum. In *Investigative Opthamology and Visual Science Supplement*, volume 33, page 1050, May 1992.

[Terzopoulos *et al.*, 1987] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models for 3D object recognition. In *Proc. IEEE Int. Conf. on Computer Vision, ICCV-87*, pages 269–276, London, England, June 1987.

[Terzopoulos, 1983] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision Graphics and Image Processing*, 24:52–96, 1983.

[Terzopoulos, 1986] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):413–424, July 1986.

[Thompson *et al.*, 1982] W. B. Thompson, K. M. Mutch, and V. Berzins. Edge detection in optical flow fields. In *Proc. of the Second National Conference on Artificial Intelligence*, pages 26–29, August 1982.

[Thompson *et al.*, 1985] W. B. Thompson, K. M. Mutch, and V. A. Berzins. Dynamic occlusion analysis in optical flow fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(4):374–383, July 1985.

[Thompson, 1980] W. B. Thompson. Combining motion and contrast for segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2:543–549, 1980.

[Tian and Shah, 1992] Y. Tian and M. Shah. MRF-Based motion estimation and segmentation. Technical Report CS-TR-92-13, University of Central Florida, Orlando, FL, July 1992.

[Tirumalai *et al.*, 1990] A. P. Tirumalai, B. G. Schunck, and R. C. Jain. Robust dynamic stereo for incremental disparity map refinement. In *Proc. Int. Workshop on Robust Computer Vision*, pages 412–434, Seattle, WA, October 1990.

[Tomasi and Kanade, 1990] C. Tomasi and T. Kanade. Shape and motion without depth. In *Proc. Int. Conf. on Comp. Vision, ICCV-90*, pages 91–95, Osaka, Japan, December 1990.

[Vaina and Grzywacz, 1992] L. M. Vaina and N. M. Grzywacz. Testing computational theories of motion discontinuities: A psychophysical study. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 212–216. Springer-Verlag, May 1992.

[van Doorn and Koenderink, 1983] A. J. van Doorn and J. J. Koenderink. Detectability of velocity gradients in moving random-dot patterns. *Vision Research*, 23:799–804, 1983.

[Vanderbilt and Louie, 1984] D. Vanderbilt and S. G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *J. of Comp. Physics*, 56:259–271, 1984.

[Varga, 1962] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Inc, New Jersey, 1962.

[Viéville and Faugeras, 1990] T. Viéville and O. Faugeras. Feed-forward recovery of motion and structure from a sequence of 2D-lines matches. In *Proc. Int. Conf. on Comp. Vision, ICCV-90*, pages 517–520, Osaka, Japan, December 1990.

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

[Weng and Cohen, 1990] J. Weng and P. Cohen. Robust motion and structure estimation using stereo vision. In *Proc. Int. Workshop on Robust Computer Vision*, pages 367–388, Seattle, WA, October 1990.

[Wertheimer, 1912] M. Wertheimer. Experimentelle Stuidien uber das Sehen von Beuegung. *Zeitsshrift fuer Psychologie*, 61:161–265, 1912.

[Woodfill and Zabih, 1991a] J. Woodfill and R. Zabih. An algorithm for real-time tracking of non-rigid objects. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-91)*, July 1991.

[Woodfill and Zabih, 1991b] J. Woodfill and R. Zabih. Using motion vision for a simple robotic task. In *Proceedings of the AAAI Fall Symposium on Sensory Aspects of Robotic Intelligence*, November 1991.

[Wu *et al.*, 1982] Z. Wu, H. Sun, and L. S. Davis. Determining velocities by propagation. In *Proc. Int. Conf. Pattern Recognition*, pages 1147–1149, Munich, West Germany, October 1982.

[Zerubia and Chellappa, 1990] J. Zerubia and R. Chellappa. Mean field approximation using compound Gauss-Markov random fields for edge detection and image estimation. In *Proc. ICASSP '90, IEEE Conf. on Acoust., Speech, and Signal Processing*, pages 21293–2196, April 1990.

[Zhuang and Haralick, 1990] X. Zhuang and R. M. Haralick. A highly robust estimator for computer vision. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 545–550, Atlantic City, NJ, June 1990.

[Zhuang *et al.*, 1992] X. Zhuang, T. Wang, and P. Zhang. A highly robust estimator through partially likelihood function modeling and its application to computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):19–35, January 1992.