Model-Based Motion Estimation and its Application to Restoration and Interpolation of Motion Pictures

Dennis Michael Martinez

S.B., S.M., Massachusetts Institute of Technology (1982)

E.E., Massachusetts Institute of Technology (1983)

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

Doctor of Philosophy

at the

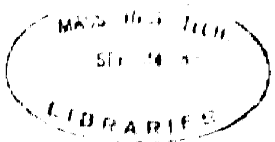
Massachusetts Institute of Technology

August 1986

Signature of Author _____
Department of Electrical Engineering and Computer Science
August 29, 1986

Certified by _____
Professor Jae S. Lim, Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Electrical Engineering Department Committee
on Graduate Students, Massachusetts Institute of Technology



Model-Based Motion Estimation and its Application to Restoration and Interpolation of Motion Pictures

by

Dennis Michael Martinez

Submitted to the Department of Electrical Engineering
and Computer Science on August 29, 1986 in partial fulfillment
of the requirements for the Degree of Doctor of Philosophy
in Electrical Engineering

Abstract

A motion picture can be manipulated in a variety of useful ways if object movement within the scene can be determined. Determining object movement is known as motion estimation. This thesis is concerned primarily with the problem of motion estimation from digitally sampled motion pictures.

Several models are developed that describe object motion with velocity fields. Given an image sequence, the velocity field is underconstrained and therefore cannot be determined uniquely. However, by imposing structural constraints on the velocity field in the form of a parametric model, it is possible to determine the model parameters uniquely.

The parametric models form the basis for two motion estimation algorithms which are described in this thesis. Experimental results are presented which demonstrate that these algorithms determine velocity fields more accurately than conventional region matching methods. One of the algorithms also has the desirable property of being computationally efficient. This algorithm is based on the least squares error criterion.

To demonstrate the performance of the least squares motion estimation algorithm, a motion-compensated noise reduction system was implemented. A number of experiments demonstrate that the motion-compensated noise reduction system can yield better results than conventional restoration methods.

A motion-compensated frame interpolation system was also implemented. This system permits frame rate conversion by arbitrary rates. Several experiments demonstrate that in a variety of situations, motion rendition obtained with the motion-compensated frame interpolation system is more natural than that which can be obtained with frame repetition strategies.

Thesis Supervisor: Jae S. Lim
Title: Associate Professor of Electrical Engineering

Acknowledgements

Throughout the five years of my graduate study at MIT I have had the privilege of working with and learning from many individuals. Space does not permit me to acknowledge all those who have contributed both to my graduate studies in general and to this thesis in particular. However, there are several individuals to whom I owe special gratitude.

I would first like to express my sincerest thanks to my thesis supervisor, Professor Jae S. Lim. His enthusiastic teaching and thoughtful supervision have had significant impact in many areas of my personal development, both technical and moral. Over the several years of working with Professor Lim, I have truly come to regard him as a leader and a friend.

I am grateful to my thesis readers Professor William Schreiber and Professor David Staelin for their valuable comments and discussions which improved the quality of this thesis.

My involvement with the Digital Signal Processing Group (DSPG) at MIT has played a significant role in shaping my technical interests. I profited greatly from numerous discussions with several members. In particular I wish to thank Webster Dove, Cory Meyers, Thrasyvoulos Pappas, David Izraelevitz, and Patrick Van-Hove for helping to make my stay in DSPG a very rewarding experience. I will always treasure our friendship.

I also wish to acknowledge the support provided by my sponsors. The National Science Foundation sponsored the theoretical investigation related to the development of motion estimation algorithms and some applications. The Advanced Television Research Program at MIT sponsored the application to noise reduction.

A special thanks goes to my parents, José and Elvira Martinez, who supported me in many ways throughout my career as a student at MIT. All those letters, phone calls, care packages, and dollars will never be forgotten.

One of my most significant experiences while I was a graduate student was to meet a lovely young lady by the name of Angela. Over the past several years she has cared for me and given me the love which enabled me to continue in the darkest moments when all seemed hopeless. I consider myself most fortunate that she is now my wife.

To God be the glory, both now and forever! Amen.

To my father and mother, José and Elvira Martinez, who gave me the freedom to pursue, the support to achieve, and the love to endure.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motion estimation

A motion picture is composed of a sequence of still frames which are displayed in rapid succession. The frame rate necessary to achieve proper motion rendition in typical visual scenes is sufficiently high that there is a great deal of temporal redundancy among adjacent frames. Most of the variation from one frame to the next is due to object motion. This motion may occur within the scene or relative to the camera which generates the sequence of still frames.

There are a wide variety of applications where one desires to manipulate a motion picture by exploiting the temporal redundancy [1]. In order to do this it is necessary to account for the presence of motion. The class of systems we are concerned with explicitly determine the movement of objects within the sequence of still frames. The process of determining the movement of objects within image sequences is known as motion estimation.

This thesis is concerned primarily with the problem of motion estimation. The motion estimation problem is phrased in a variety of contexts that depend on a particular representation for motion. The specific motion representation which we use is based on velocity fields.

[1] Some applications which have been proposed include (1) noise reduction, (2) spatio-temporal interpolation, and (3) motion picture coding.

1.1.1 Previous approaches to motion estimation

A number of methods for performing motion estimation have been proposed in the past. In general there have been three primary problems with previously used methods:

- motion estimation accuracy with noisy images
- estimating large velocities
- computational complexity

Many algorithms are explicitly formulated under the assumptions of high signal-to-noise level. As a consequence, if the algorithms are applied to noisy pictures, the motion estimation errors are typically large. Most motion-compensated systems require very accurate motion estimates in order to maintain adequate picture quality. Consequently the algorithms which are sensitive to noise are not generally useful.

In real-life motion pictures the velocity field is a complicated function of spatio-temporal position. Therefore most algorithms are based on local operations. One of the problems with this approach is that typically only small velocity fields can be estimated reliably.

Many applications of motion compensation require real-time operation. For real-time operation to be feasible it is necessary for the algorithms to be computationally efficient. Even in those applications where real-time operation is not required, computational complexity is an important characteristic which affects the cost of implementing a specific motion estimation algorithm.

1.1.2 A new approach to motion estimation

The purpose of this thesis is to present a new approach to motion estimation. This approach is based on parametric signal and velocity models. These models are general enough so they apply to a wide variety of signals derived from motion pictures. We present two new motion estimation algorithms which are based on

these models. The algorithms are capable of estimating the velocity field very accurately from noisy pictures. Furthermore, one of the algorithms has the important property that the model parameters can be determined by solving linear equations (least squares algorithm). Consequently the algorithm is computationally efficient.

These algorithms are based exclusively on local operations and consequently cannot estimate large velocities directly. However, because they typically generate velocity estimates with subpixel accuracy, they can be used on spatially downsampled images to generate accurate initial coarse velocity estimates. The coarse velocity estimates can be used at the original picture resolution to generate accurate estimates of large velocities. The resulting algorithm is referred to as a multigrid method.

1.2 Applications of motion estimation

The multigrid/least squares algorithm was used in several applications of motion-compensation. We developed a motion-compensated noise reduction system and a motion-compensated frame interpolation system.

1.2.1 Motion-compensated noise reduction

The basic structure of a motion-compensated noise reduction system is shown in Figure 1.1. At each point in the image sequence the velocity field is estimated and used to compute a motion trajectory. The signal intensity remains constant along motion trajectories. Therefore the samples along the trajectory are processed with a one-dimensional filter. We apply this technique to signals degraded with either additive noise or impulsive noise. For additive noise reduction the filter averages the samples and for impulsive noise the filter computes the median of the samples.

In these systems, motion estimation error introduces blur or other visible artifacts into the picture. Therefore they provide a subjective evaluation of the performance of the motion estimation algorithms.

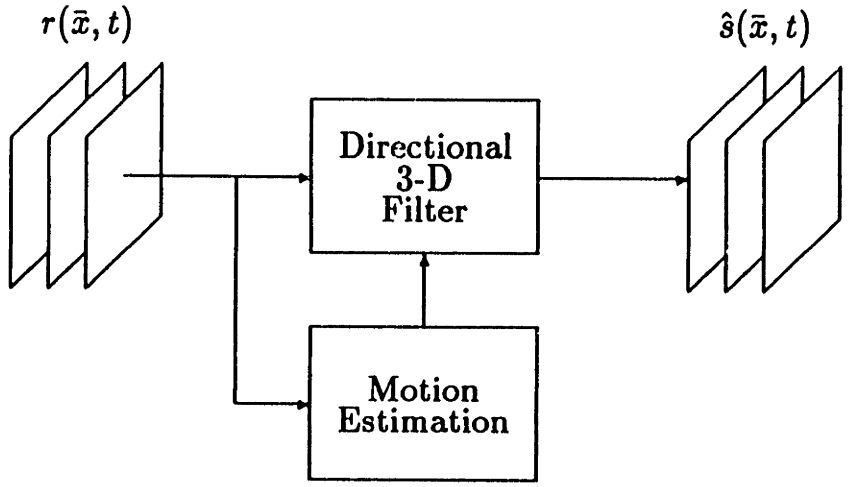


Figure 1.1: Motion-compensated noise reduction system

1.2.2 Motion-compensated frame interpolation

A motion-compensated frame interpolation system has the basic form shown in Figure 1.2. This system permits computing intermediate frames of the motion

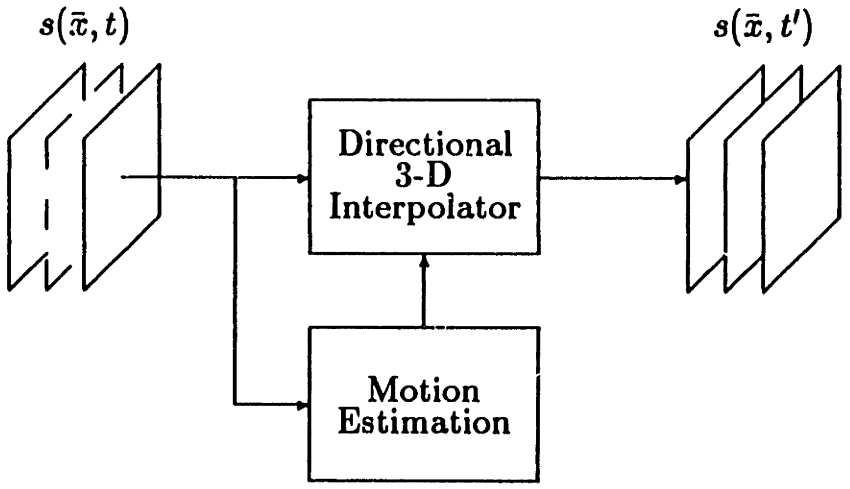


Figure 1.2: Motion-compensated frame interpolation system

picture. At each point where a sample is desired, the velocity field is estimated and projected onto the closest frame. The signal value at this position is used as the interpolated value.

1.3 Thesis overview

1.3.1 Survey of motion estimation algorithms

Motion estimation is a fundamental component of motion-compensated image processing systems. Consequently, a wide variety of motion estimation algorithms have been proposed in the literature. In Chapter 2 we review some of the more widely used methods.

1.3.2 Model-based motion estimation

In Appendix A we derive some very general motion models. The models form the mathematical basis for analyzing the motion estimation problem. A specific form of the model is used as the basis for two motion estimation algorithms which are described in Chapter 3. One algorithm is based entirely on linear models and uses the least squares error criterion. The second algorithm is based on a maximum likelihood parameter estimation method. For comparison purposes we also implemented a region matching algorithm which is described in Appendix B.

We summarize the computational requirements of these algorithms. The computational requirements for the region matching and maximum likelihood are very similar. However, the least squares algorithm requires substantially less computation than the region matching and maximum likelihood algorithms (almost two orders of magnitude).

For these algorithms we analyze the effect of additive random noise on motion estimation accuracy. The Cramer Rao bounds are derived for the case of additive white Gaussian noise and discrete observations of the signal.

In Chapter 3 we also present an algorithm for extending the effective search range of motion estimators. The algorithm is based on a multi-grid method and permits very large velocity fields to be estimated with high accuracy, in a computationally efficient manner.

1.3.3 Motion estimation experiments

In Chapter 4 a number of experiments are described which compare the motion estimation algorithms described in Chapter 3 and the region matching algorithm described in Appendix B. Two basic comparisons were made:

- motion estimation error as a function of signal-to-noise level
- picture quality obtained with motion-compensated temporal averaging

The first set of experiments measured the motion estimation error as a function of signal-to-noise level with synthetic test images. It is shown that the error performance of the least squares algorithm is very similar to the maximum likelihood algorithm, both of which are superior to the region matching algorithm for realistic signal-to-noise levels.

Next we processed some pictures by frame averaging along trajectories determined by the algorithms. For this operation, motion estimation error introduces artifacts and causes the resulting picture to be blurred. These experiments confirm the empirical results obtained with the synthetic test images. The pictures processed with the maximum likelihood and least squares algorithms were comparable, and better than the pictures processed with the region matching algorithm.

In addition we present some experiments in motion estimation of large velocities. These experiments demonstrate the effectiveness of the multigrid algorithm for estimating large velocities.

1.3.4 Motion picture restoration

In Chapter 5 we describe some motion picture restoration systems. The degradations that the restoration systems we developed can suppress include: (1) additive random noise, and (2) impulsive noise. We compared the pictures processed with the motion-compensated systems to those processed with adaptive single frame restoration and adaptive multiple frame restoration systems. On the basis of informal subjective viewing, the pictures processed with the motion-compensated sys-

tems were usually judged to be better than those processed with the two adaptive methods.

1.3.5 Motion picture frame interpolation

In Chapter 6 we describe two motion picture frame interpolation systems. We developed a system which performs frame rate conversion by motion-compensated frame interpolation. This system permits rate conversion by arbitrary frame rates (for example 10 %). We compared this system to an alternate method based on frame repetition. This system does frame rate conversion by repeating (or dropping) frames. Each "interpolated" frame is obtained by selecting the frame in the original sequence which is closest in time to the desired frame. A number of informal subjective tests revealed the motion-compensated system to yield comparable results to the repetition system for scenes with slight motion. However, when large moving areas are present, the motion-compensated interpolation method was preferred over the frame repetition method. When there are large moving areas the frame repetition method produces "jerky" motion, while the motion-compensated interpolation method yields more continuous motion rendition.

1.4 Notation and conventions

In this section we define the notation and conventions used throughout this thesis.

A great deal of our analysis involves systems of linear equations. We make extensive use of matrix and vector notation. Matrices are represented with upper case symbols (A, B, etc.) and vectors are represented with either upper or lower case symbols with a bar over the symbol ($\bar{a}$, $\bar{b}$, $\bar{S}$ etc.). For example, a set of linear equations is written as

$$A\bar{x} = \bar{b}.$$

The inverse of a matrix A is written as A^{-1}, and the transpose is written as A^T. Entries of a matrix are referred to with subscripted notation. Therefore, A_{ij} refers to the i^{th} row and j^{th} column of matrix A.

All vectors are column vectors. When written in line, the convention is $\bar{b} = (b_1, b_2, \ldots, b_N)^T$. Entries of a vector are referred to with subscripted notation. Therefore, b_i refers to the i^{th} element of vector $\bar{b}$.

We adopt a common notation used to distinguish continuous versus discrete signals. A signal whose independent variables are enclosed in parenthesis "$(\cdot)$" is a continuous signal, and a signal whose independent variables are enclosed in brackets "$[\cdot]$" is a discrete signal. Therefore the signal $s(\cdot)$ refers to a continuous signal, whereas the signal $s[\cdot]$ is a discrete signal.

The signals which we deal with are either single images or sequences of images which comprise a motion picture. The luminance of an image is a function of two variables, x and y. For the sake of notational convenience, the pair (x, y) will be written as $\bar{x}$ in many occasions. Therefore the image $s(x, y)$ is equivalent to the image $s(\bar{x})$. Continuous sequences of images are written as $s(x, y, t) = s(\bar{x}, t)$. Therefore $s(\bar{x}, t_0)$ refers to the frame at time instant t_0.

On several occasions we will use Fourier transforms. The Fourier transform of an image is

$$S(\omega_x, \omega_y) = \int_{-\infty}^{\infty} s(x, y) e^{-j(\omega_x x + \omega_y y)} dx\, dy \tag{1.1}$$

and the Fourier transform of a movie sequence is

$$S(\omega_x, \omega_y, \omega_t) = \int_{-\infty}^{\infty} s(x, y, t) e^{-j(\omega_x x + \omega_y y + \omega_t t)} dx\, dy\, dt. \tag{1.2}$$

Chapter 2

Survey of motion estimation algorithms

Motion-compensated image processing systems involve motion estimation in one form or another. The purpose of this chapter is to describe a variety of motion estimation algorithms which have been proposed in the past.

2.1 Motion estimation methodologies

Most image sequences are derived from natural scenes. A two-dimensional frame is obtained by projecting a three-dimensional illumination function onto the two-dimensional image plane of a camera and sampling in both space and time. Hence there is a strong relationship between the spatial and temporal properties of these signals. As motion occurs in the three-dimensional scene there are corresponding changes in the sequence of two-dimensional projections. A variety of methods have been proposed for extracting three-dimensional motion parameters from the sequence of two-dimensional projections [18,29,34,35,3]. These methods have focused primarily on the motion of rigid three-dimensional bodies. A common formulation of the problem involves determining the motion parameters, which include a translation component, rotation component, and center of rotation. It is clear that the motion characteristics which are found in typical real-life image se-

quences are vastly more complicated than this simple model can accommodate and a more general representation is required.

A more widely used representation of objects within image sequences involves segmenting the two-dimensional frames into different regions based on luminance properties and associating each region with a three-dimensional object. A dynamic scene is viewed as a set of two-dimensional regions that change dynamically in shape, texture, luminance, etc. as a function of time. With this representation motion estimation involves determining the movement of object boundaries and other features within the image sequence.

Within this framework there are basically three methodologies which have been used for motion estimation [15,11,25,28,33]:

- transform domain methods
- region matching methods
- spatio-temporal constraint methods

In the following sections we describe some algorithms based on these methodologies.

In addition to describing the algorithms, we discuss their limitations. To understand the limitations it is necessary to know the requirements of the algorithms. There are many factors which affect the requirements imposed on a particular algorithm. The most obvious factor is the intended application. Other factors are related to the specific properties of the signals which are being manipulated (frame rate, picture resolution, etc.). For reference purposes, we use the NTSC standard as a baseline system. This choice is motivated by the widespread use of this standard. Two application areas which we have investigated include noise reduction and frame interpolation. We use these systems to select the requirements of the algorithms. Therefore the requirements are stated for the problem of noise reduction or frame interpolation of NTSC signals.

The most important requirements can be itemized as follows:

- Accuracy/large velocities: In order to avoid introducing blur or other artifacts into the picture, velocities must be estimated typically with subpixel accuracy

(error < 1 pel/frame). This is an important requirement which most algorithms fail to meet. Furthermore, for NTSC pictures, velocity fields on the order of 10 pels/frame are present often.

- Resolution: As a moving object occludes the background, the velocity field is discontinuous. To avoid noticeable artifacts in these regions of the picture, the estimator must resolve velocity discontinuities over a spatial distance on the order of 2 to 3 pels.

- Signal-to-noise levels: For noise reduction applications, signal-to-noise levels as low as 20 dB are commonly present. For frame interpolation applications, signal-to-noise levels on the order of 30 - 40 dB are typical.

- Computational complexity: If real-time operation on NTSC signals is to be obtained, it is extremely important that the algorithm is computationally efficient.

2.1.1 Transform domain methods

One formulation of motion estimation in the transform domain is based on the relationship between Fourier transforms of shifted two-dimensional sequences [11]. If the Fourier transform of $s(x, y)$ is $S(\omega_x, \omega_y)$, then the transform of a shifted version of $s(x, y)$ is given by

$$s(x - d_x, y - d_y) \Longleftrightarrow S(\omega_x, \omega_y) \exp[-j2\pi(\omega_x d_x + \omega_y d_y)]. \tag{2.1}$$

Suppose we have two frames $s(x, y, t_0)$ and $s(x, y, t_1)$ corresponding to time instants t_0 and t_1, with two-dimensional Fourier transforms $S_0(\omega_x, \omega_y)$ and $S_1(\omega_x, \omega_y)$. If the frame at time instant t_1 is a shifted version of the frame at time instant t_0 with displacements d_x and d_y, then the unwrapped phase difference between the two Fourier transforms is

$$S_0(\omega_x, \omega_y) - S_1(\omega_x, \omega_y) = \delta\phi(\omega_x, \omega_y) = -2\pi(\omega_x d_x + \omega_y d_y). \tag{2.2}$$

Extending this basic principle to motion estimation is straightforward. The unwrapped phase difference between two frames is computed at a number of frequencies and a set of overdetermined linear equations is generated. Solving the set of equations leads to an estimate of the displacement field characterized by d_x and d_y

$$-2\pi \begin{bmatrix} \omega_{x1} & \omega_{y1} \\ \omega_{x2} & \omega_{y2} \\ \vdots & \vdots \\ \omega_{xN} & \omega_{yN} \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} \delta\phi(\omega_{x1}, \omega_{y1}) \\ \delta\phi(\omega_{x2}, \omega_{y2}) \\ \vdots \\ \delta\phi(\omega_{xN}, \omega_{yN}) \end{bmatrix}. \tag{2.3}$$

In practice this approach is very limited because it only applies to the case where all objects move in the same direction and by the same amount against a uniform background. Another difficulty with this method is that it requires computation of the unwrapped phase of the Fourier transform.

An alternate formulation was proposed by Stuller and Netravali [33]. It is based on a coefficient-recursive estimation procedure and can be summarized as follows. A given frame (say at time t_0) is partitioned into blocks. Consider one block centered about the point $\bar{x}_0$, where the samples are organized into a one-dimensional vector $\bar{S}(\bar{x}_0, t_0)$. Let $\bar{\phi}_n$ be the n^{th} basis vector of a unitary transform and let $c_n(\bar{x}_0, t_0)$ be the corresponding transform coefficient which is computed as follows

$$c_n(\bar{x}_0, t_0) = \bar{S}(\bar{x}, t_0)^T \bar{\phi}_n. \tag{2.4}$$

An error term for this coefficient is defined as the difference between the coefficient at time t_0 and the coefficient for a displaced frame at time $t_0 + \delta t$

$$e_n(\bar{d}, \bar{x}_0, t_0) = \bar{S}(\bar{x}_0, t_0)^T \bar{\phi}_n - \bar{S}(\bar{x}_0 - \bar{d}, t_0 + \delta t)^T \bar{\phi}_n \tag{2.5}$$

which can be simplified to

$$e_n(\bar{d}, \bar{x}_0, t_0) = \left(\bar{S}(\bar{x}_0, t_0) - \bar{S}(\bar{x}_0 - \bar{d}, t_0 + \delta t)\right)^T \bar{\phi}_n. \tag{2.6}$$

A coordinate descent algorithm is used to determine the displacement vector $\bar{d}$ which minimizes the ensemble $\{e_n^2\}$ over the set of basis vectors that comprise the

unitary transform. This iteration results in an estimate of the displacement field at the point $(\bar{x}_0, t_0)$. Based on some experiments with noisy images, this algorithm is reported to achieve slightly smaller estimation error than a pel recursive region matching algorithm described by Netravali and Robbins [25].

2.1.2 Region matching methods

A more general approach to the motion estimation problem is based on region matching methods. This approach involves segmenting a frame into small regions and searching for the displacement which produces a "best match" among possible regions in an adjacent frame. Most region matching methods can be described with the following formulation

$$\min_{\bar{d}} \left\{ C(\bar{d}, \bar{x}_0, t_0) = F[s(\bar{x}, t_0), s(\bar{x}_0 - \bar{d}, t_0 + \delta t)] \right\} \tag{2.7}$$

where $C(\cdot)$ is a cost function associated with a two-dimensional displacement vector $\bar{d}$ and $F[\cdot]$ is a function which measures the similarity between two frames which have been displaced relative to each other. The objective is to search over a two-dimensional space to determine the displacement $\bar{d}$ which minimizes the cost function at the spatio-temporal position $(\bar{x}_0, t_0)$.

A commonly used region matching method involves minimizing the sum of squares of two regions that have been displaced relative to each other. Specifically, an estimate of the displacement field is obtained by determining the vector $\bar{d}$ which minimizes the following expression

$$\min_{\bar{d}} \left\{ \sum_{i=1}^{N} [s(\bar{x}_i, t_0) - s(\bar{x}_i - \bar{d}, t_0 + \delta t)]^2 \right\} \tag{2.8}$$

where the set of points $\{\bar{x}_i\}$ are taken from a particular analysis window. The widely used pel recursive method of Netravali and Robbins [25] has this basic form. They use a steepest descent algorithm to minimize this function. This results in the iteration

$$\bar{d}_{k+1} = \bar{d}_k - \frac{\epsilon}{2} \nabla_d [DFD(\bar{d}_k)]^2 \tag{2.9}$$

where

$$DFD(\bar{d}) = s(\bar{x}_0, t_0) - s(\bar{x}_0 - \bar{d}, t_0 + \delta t) \tag{2.10}$$

is the displaced frame difference. In a later improvement of the algorithm [24], the squared displaced frame difference (DFD) is minimized over a region. The resulting algorithm resembles Equation (2.8). It should be noted that evaluation of Equation (2.9) requires that values of $s(\bar{x}, t)$ at arbitrary spatio-temporal positions are available. Therefore an interpolation procedure is required to compute values which are not on the sampling grid. The bilinear interpolator is often used. Numerous variations of this basic algorithm have been used in applications ranging from interframe coding [25] to noise reduction [8].

One of the primary problems with this approach is the computational requirement. Algorithms which address this problem have been proposed by several researchers. One straightforward modification involves using a nonlinear optimization procedure which converges at a faster rate than steepest descent with fixed line search parameter ϵ [32]. An alternative approach is to limit the search over $\bar{d}$ to a quantized space. This reduces the nonlinear optimization problem to a discrete search problem. Cafforio and Rocca [4] used a maximum likelihood search strategy in conjunction with dynamic programming techniques based on the Viterbi algorithm. Ninomiya and Ohtsuka [27] used an iterative binary tree search algorithm which refines an initial estimate through successive iterations over smaller search menus. A second problem with this approach is that it is sensitive to noise. In Chapter 4 we present some results that demonstrate this fact.

2.1.3 Spatio-temporal constraint methods

Uniform translation is one of the most common motion types encountered in image sequences. The following relationship models this situation

$$s(\bar{x}, t) = s(\bar{x} - \bar{v} \cdot (t - t_0), t_0) \tag{2.11}$$

where $\bar{v}$ is the velocity field in the region of interest. A direct consequence of this relationship is the spatio-temporal constraint equation

$$v_x \frac{\partial s}{\partial x} + v_y \frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} = 0. \tag{2.12}$$

In Appendix A we demonstrate that this is a special case of a much more general representation.

This constraint equation forms the basis for a variety of motion estimation algorithms which have been developed [15,19,25,28]. One method for estimating the velocity field from this equation is to evaluate the spatial and temporal gradients of the picture and generate a set of overdetermined linear equations

$$\begin{bmatrix} \left.\frac{\partial s}{\partial x}\right|_{P_1} & \left.\frac{\partial s}{\partial y}\right|_{P_1} \\ \vdots & \vdots \\ \left.\frac{\partial s}{\partial x}\right|_{P_N} & \left.\frac{\partial s}{\partial y}\right|_{P_N} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} \left.\frac{\partial s}{\partial t}\right|_{P_1} \\ \vdots \\ \left.\frac{\partial s}{\partial t}\right|_{P_N} \end{bmatrix}. \tag{2.13}$$

Commonly used methods involve estimating the gradients with finite differences. One problem with this approach is that obtaining accurate estimates of the spatio-temporal gradients from noisy images is difficult. This problem is further compounded when there is aliasing due to undersampling. In real-life motion pictures the frame rates are low enough that temporal undersampling is an important problem. Frame differences do not yield acceptable estimates of temporal gradients.

A more subtle problem is that this set of overdetermined equations does not always have a unique solution. Furthermore this problem is ill-conditioned whenever the samples used to form the estimate lie within an edge of the picture. One form of the least squares algorithm which we implemented minimizes the same expression, but deals with both problems of ill-conditioning and gradient estimation.

An alternative approach based on the constraint equation is to introduce an additional constraint. Horn and Schunck [13] introduced a smoothness constraint. They seek the solution which satisfies the constraint equation and simultaneously

minimizes the squared gradient of the velocity field. The first constraint generates the error function ε_m defined by

$$\varepsilon_m = \frac{\partial s}{\partial x} v_x + \frac{\partial s}{\partial y} v_y + \frac{\partial s}{\partial t} \tag{2.14}$$

and the second constraint generates the error function ε_l defined by

$$\varepsilon_l^2 = \left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial x}\right)^2. \tag{2.15}$$

The velocity field is determined by minimizing the function

$$\varepsilon^2 = \int\int (\alpha^2 \varepsilon_l^2 + \varepsilon_m^2) dx\, dy. \tag{2.16}$$

In this expression, α is a parameter which permits weighting the relative error due to each term in carrying out the minimization. The integral is taken over the entire region of support of the image.

They propose an iterative algorithm for determining the velocity field from this expression. The basic iteration can be written as

$$v_x^{i+1} = \tilde{v}_x^i - \frac{\frac{\partial s}{\partial x}\left[\frac{\partial s}{\partial x}\tilde{v}_x^i + \frac{\partial s}{\partial y}\tilde{v}_y^i\right] + \frac{\partial s}{\partial t}}{\alpha^2 + \frac{\partial s}{\partial x}^2 + \frac{\partial s}{\partial y}^2} \tag{2.17}$$

$$v_y^{i+1} = \tilde{v}_y^i - \frac{\frac{\partial s}{\partial y}\left[\frac{\partial s}{\partial x}\tilde{v}_x^i + \frac{\partial s}{\partial y}\tilde{v}_y^i\right] + \frac{\partial s}{\partial t}}{\alpha^2 + \frac{\partial s}{\partial x}^2 + \frac{\partial s}{\partial y}^2} \tag{2.18}$$

where $\tilde{v}_x$ and $\tilde{v}_y$ are local averages of the velocity field components. The gradients of the picture are computed with finite differences. Horn and Schunck report that this method yields large motion estimation errors if noise is present in the sequences. Furthermore this algorithm requires significant computation.

Chapter 3

Model-based motion estimation

In this chapter we describe two algorithms for estimating velocity fields from image sequences. The first algorithm is based on the spatio-temporal constraint equation described in Chapter 2 and is referred to as the least squares algorithm. The second algorithm is based on a maximum likelihood formulation. Both algorithms are used to estimate the translational components of a velocity field.

These two algorithms are based entirely on local operations. Consequently they can only determine relatively small velocity fields (on the order of 2 pels/frame). In Section 3.9 we describe a multigrid algorithm which uses these local algorithms at different picture resolutions. The multigrid algorithm permits estimation of large velocities accurately.

3.1 Motion models

Our motion estimation strategy is based on a very general motion model which is derived in Appendix A. In this section we summarize the important features of the model which form the basis for our motion estimation algorithms.

The models relate a sequence of frames to a single image with a motion description function $\bar{\alpha}(\bar{x}, t)$, which is defined by the expression

$$s(\bar{x}, t) = s_0(\bar{\alpha}(\bar{x}, t)). \tag{3.1}$$

A direct consequence of this relationship is that there exists a velocity field $\bar{v}(\bar{x}, t)$ which is related to the signal through the partial differential equation

$$v_x(\bar{x}, t)\frac{\partial s}{\partial x} + v_y(\bar{x}, t)\frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} = 0. \tag{3.2}$$

The velocity field components $v_x(\bar{x}, t)$ and $v_y(\bar{x}, t)$ can be determined uniquely from the motion description function by solving a set of linear equations. Conversely, the motion description function can be determined from the velocity field by solving a linear partial differential equation (provided the velocity field is a true velocity field which can be obtained from a motion description function).

In the context of this model, the problem of motion estimation is to determine either the motion description function or the velocity field from a given signal. The estimation problem has several components.

Signal model: The important aspects of the model are the relationship between the signal and a motion description function or equivalently between the signal and the velocity field.

Observation space: The observation space for this problem consists of discrete samples of the signal $r(\bar{x}, t)$ defined as

$$r(\bar{x}, t) = s(\bar{x}, t) + n(\bar{x}, t) \tag{3.3}$$

where $n(\bar{x}, t)$ is a random noise field.

Estimation procedure: Our objective is to formulate a procedure for estimating the

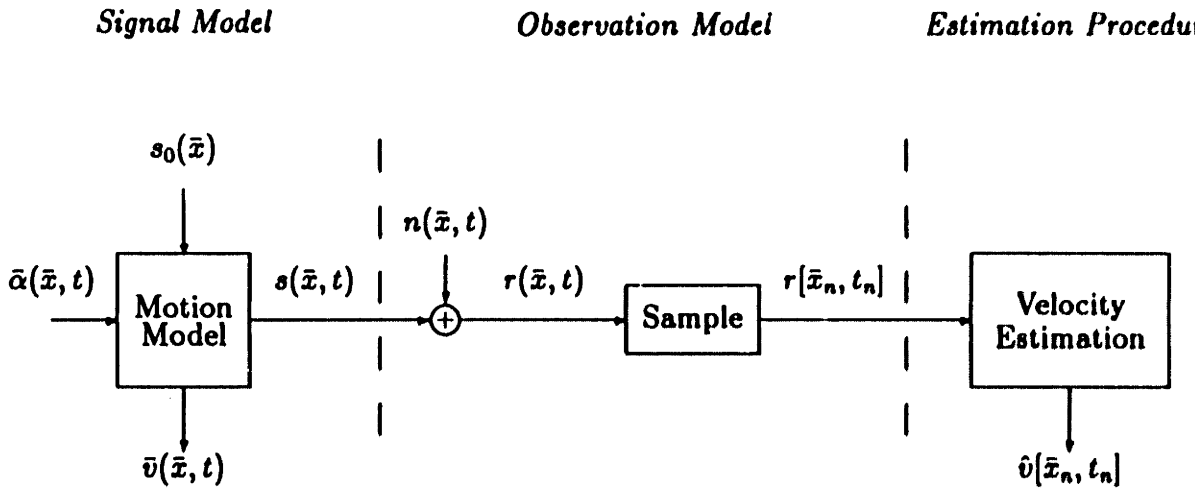


Figure 3.1: Motion estimation problem

parameters that define the velocity field from discrete observations of the signal. These components are illustrated in Figure 3.1.

3.2 Motion estimation based on local properties

The motion estimation problem is underconstrained in the sense that a unique solution does not exist. This is intuitively clear if we consider the fact that motion description functions and velocity fields are vector-valued functions of spatio-temporal position and the picture intensity is a scalar-valued function. Therefore in a sense there are more "unknowns" than "knowns".

To address this issue it is necessary to impose additional constraints into the problem. For real-life motion pictures, typically the velocity field varies much more slowly than the picture intensity. Therefore if the velocity at one point in the picture $(\bar{x}_0, t_0)$ has a velocity $\bar{v}(\bar{x}_0, t_0)$, then points in the neighborhood of $(\bar{x}_0, t_0)$ will usually have approximately the same velocity. This observation can be used to introduce another constraint.

We assume that over a small region of the picture the velocity field is constant and can be characterized by the two components of the velocity field. Therefore the method we use for determining arbitrary velocity fields is to use a translational model at each point in the picture. An estimate of the local velocity is obtained by using samples taken from a small region of the image sequence in the neighborhood of the point of interest. More specifically, suppose we want an estimate of the velocity field at the point $P_0 = (\bar{x}_0, t_0)$. The available samples in the vicinity of P_0 are used to form this estimate. Therefore the algorithms we describe in the next two sections deal specifically with the problem of estimating translational components of a velocity field from the samples in the neighborhood of the point where a velocity estimate is desired.

The model for local translation has two forms. The direct form is based on the motion description function $\bar{\alpha}(\bar{x}, t) = \bar{x} - \bar{v} \cdot (t - t_0)$

$$s(\bar{x}, t) = s_0(\bar{x} - \bar{v} \cdot (t - t_0)). \tag{3.4}$$

The differential form is based on the velocity field

$$v_x \frac{\partial s}{\partial x} + v_y \frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} = 0. \tag{3.5}$$

It should be noted that these two relations imply each other. The least squares algorithm is based on the differential form of the model and the maximum likelihood algorithm is based on the direct form. These algorithms are described in the next two sections.

3.3 Least squares motion estimation

The least squares motion estimation algorithm is based on the relationship specified by Equation (3.5) and assumes the constraint is satisfied approximately at all points in some region Ψ. Since a given signal will not always satisfy the constraint exactly, the right hand side (called the error) will be nonzero at some points within Ψ. The least squares estimator minimizes this squared error. There are two formulations which we consider. The first formulation minimizes the squared error at a set of N discrete points. For this case the velocity estimates are given by

$$\min_{v_x, v_y} \frac{1}{N} \left\{ \sum_{i=1}^{N} \left(v_x \frac{\partial s}{\partial x}\bigg|_{P_i} + v_y \frac{\partial s}{\partial y}\bigg|_{P_i} + \frac{\partial s}{\partial t}\bigg|_{P_i} \right)^2 \right\}. \tag{3.6}$$

The second formulation minimizes the squared error over the entire region Ψ, and results in the estimator

$$\min_{v_x, v_y} \left\{ \int\int\int_{\Psi} \left(v_x \frac{\partial s}{\partial x} + v_y \frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} \right)^2 dx\, dy\, dt \right\}. \tag{3.7}$$

These are quadratic functions of the parameter values, so the optimal velocity components are determined from a set of linear equations

$$W\bar{v} = \bar{\gamma}. \tag{3.8}$$

For the discrete point minimization, W and $\bar{\gamma}$ are given by

$$W = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\partial s}{\partial x}\Big|_{P_i} \right)^2 & \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\partial s}{\partial x}\Big|_{P_i} \right) \left(\frac{\partial s}{\partial y}\Big|_{P_i} \right) \\ \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\partial s}{\partial x}\Big|_{P_i} \right) \left(\frac{\partial s}{\partial y}\Big|_{P_i} \right) & \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\partial s}{\partial y}\Big|_{P_i} \right)^2 \end{bmatrix} \tag{3.9}$$

$$\bar{\gamma} = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\partial s}{\partial x}\Big|_{P_i} \right) \left(\frac{\partial s}{\partial t}\Big|_{P_i} \right) \\ \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\partial s}{\partial y}\Big|_{P_i} \right) \left(\frac{\partial s}{\partial t}\Big|_{P_i} \right) \end{bmatrix}. \tag{3.10}$$

For the continuous region minimization, W and $\bar{\gamma}$ are given by

$$W = \begin{bmatrix} \int\int\int_{\Psi} \left(\frac{\partial s}{\partial x}\right)^2 dx\,dy\,dt & \int\int\int_{\Psi} \left(\frac{\partial s}{\partial x}\right)\left(\frac{\partial s}{\partial y}\right) dx\,dy\,dt \\ \int\int\int_{\Psi} \left(\frac{\partial s}{\partial x}\right)\left(\frac{\partial s}{\partial y}\right) dx\,dy\,dt & \int\int\int_{\Psi} \left(\frac{\partial s}{\partial y}\right)^2 dx\,dy\,dt \end{bmatrix} \tag{3.11}$$

$$\bar{\gamma} = \begin{bmatrix} \int\int\int_{\Psi} \left(\frac{\partial s}{\partial x}\right)\left(\frac{\partial s}{\partial t}\right) dx\,dy\,dt \\ \int\int\int_{\Psi} \left(\frac{\partial s}{\partial y}\right)\left(\frac{\partial s}{\partial t}\right) dx\,dy\,dt \end{bmatrix}. \tag{3.12}$$

In Section 3.3.2 we describe a numerical procedure for computing W and $\bar{\gamma}$ from samples of the signal. Computing these quantities involves signal estimation. Consequently the overall least squares algorithm has the structure shown in Figure 3.2. In the remainder of this section we derive the conditions under which Equation (3.8) has a unique solution.

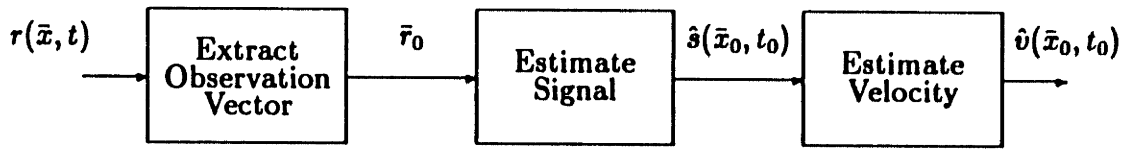


Figure 3.2: Least squares motion estimation algorithm

When W is nonsingular, the velocity components are obtained by solving the set of linear equations specified by Equation (3.8). However, when W is singular, a unique solution does not exist. It turns out in practice that over a large portion of typical pictures, W is ill-conditioned or nearly singular. In these regions, small errors in computing the entries of W or $\bar{\gamma}$ can lead to large errors in computing the velocity field. If all the samples used to compute W lie within a region of the picture where there is a perfect edge, then W will be singular. For the discrete

point minimization, this can be shown as follows [1].

We can determine a functional form which all signals $s(\bar{x},t)$ must satisfy such that W is singular. By direct evaluation, W is singular if and only if

$$W_{11}W_{22} = W_{12}W_{21}. \tag{3.13}$$

Therefore if and only if a signal satisfies the following equation, then W will be singular

$$\left[\sum_{i=1}^{N}\left(\frac{\partial s}{\partial x}\bigg|_{P_i}\right)^2\right]\left[\sum_{i=1}^{N}\left(\frac{\partial s}{\partial y}\bigg|_{P_i}\right)^2\right] = \left[\sum_{i=1}^{N}\left(\frac{\partial s}{\partial x}\frac{\partial s}{\partial y}\bigg|_{P_i}\right)^2\right]^2. \tag{3.14}$$

We can rewrite this equation in the form

$$(\bar{a}^T\bar{a})(\bar{b}^T\bar{b}) = (\bar{a}^T\bar{b})^2 \tag{3.15}$$

where

$$\bar{a} = \left(\frac{\partial s}{\partial x}\bigg|_{P_1}, \frac{\partial s}{\partial x}\bigg|_{P_2}, \cdots, \frac{\partial s}{\partial x}\bigg|_{P_N}\right)^T \tag{3.16}$$

and

$$\bar{b} = \left(\frac{\partial s}{\partial y}\bigg|_{P_1}, \frac{\partial s}{\partial y}\bigg|_{P_2}, \cdots, \frac{\partial s}{\partial y}\bigg|_{P_N}\right)^T. \tag{3.17}$$

From the Schwartz inequality it follows that $\bar{a} = \alpha\bar{b}$, for some constant α. Therefore

$$\frac{\partial s}{\partial x}\bigg|_{P_i} = \alpha\frac{\partial s}{\partial y}\bigg|_{P_i} \qquad \text{for } i = 1\cdots N. \tag{3.18}$$

Consider all signals such that

$$\frac{\partial s}{\partial x} = \alpha\frac{\partial s}{\partial y}. \tag{3.19}$$

The class of signals which has this property can be expressed in the form

$$s(\bar{x},t) = s_0(x - \alpha y, t). \tag{3.20}$$

This implies that $s(\bar{x},t)$ is constant along lines where $x = \alpha y$. Therefore the samples all lie along an edge which is parallel to the line $x = \alpha y$.

[1] The derivation of this condition for the continuous region minimization follows the same line of reasoning and leads to the same conclusion.

3.3.1 Motion estimation in the presence of edges

Edges are a very prominent feature in most images. Therefore it is necessary for us to guarantee that our algorithm is robust even though W is ill-conditioned. In this subsection we formulate a numerical solution to this problem and provide a physical interpretation of the result. The crucial result which we derive can be summarized as follows:

- When W is ill-conditioned, then there is one direction in which there is high contrast and another direction in which there is low contrast (there is a well defined edge).
- The eigenvectors of W point in the directions of minimum and maximum contrast.
- By applying a singular value decomposition (SVD) to the matrix W, we can generate an estimate of the velocity field in the direction orthogonal to the edge.

We derive these results for the discrete point minimization procedure, but the results also apply directly to the continuous region minimization.

Consider the problem of determining the stationary points of the following function (at a stationary point the function has either a local minimum or maximum)

$$\min_{\bar{\alpha}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\bar{\alpha}^T \nabla_x s(\bar{x}, t)|_{P_i} \right)^2 \right\} \tag{3.21}$$

subject to the constraint

$$\bar{\alpha}^T \bar{\alpha} = 1. \tag{3.22}$$

The quantity

$$\left(\bar{\alpha}^T \nabla_x s(\bar{x}, t)|_{P_i} \right)^2 \tag{3.23}$$

is the magnitude of the directional spatial derivative of the picture along direction $\bar{\alpha}$ at the point P_i. The summation represents the average magnitude of the directional derivative over the region of interest, so the extrema correspond to the directions

of minimum and maximum contrast. This is a quadratic function in the unknown vector $\bar{\alpha}$ and minimizing Equation (3.21) is equivalent to minimizing the function

$$\min_{\bar{\alpha}} \left\{ \bar{\alpha}^T W \bar{\alpha} \right\} \tag{3.24}$$

subject to the constraint given in Equation (3.22). In this expression, W is the same matrix as in the velocity estimator. This constrained optimization problem can be converted to an unconstrained problem by introducing the Lagrange multipier λ and minimizing the Lagrangian

$$\min_{\bar{\alpha}} \left\{ \bar{\alpha}^T W \bar{\alpha} + \lambda(1 - \bar{\alpha}^T \bar{\alpha}) \right\}. \tag{3.25}$$

The quantity to be minimized can be written in the form

$$\bar{\alpha}^T (W - \lambda I) \bar{\alpha} + \lambda. \tag{3.26}$$

Differentiating this equation with respect to $\bar{\alpha}$ and setting it equal to zero produces the result

$$(W - \lambda I)\bar{\alpha} = \bar{0}, \tag{3.27}$$

which is equivalent to

$$W\bar{\alpha} = \lambda \bar{\alpha}. \tag{3.28}$$

The results follow immediately from this equation.

- The Lagrange multipliers are the eigenvalues of W and the directions of minimum and maximum contrast are the eigenvectors of W.
- The larger eigenvalue (λ_{max}) is the average of the squared magnitude of the directional derivative along the direction of maximum contrast and the smaller eigenvalue (λ_{min}) is the average of the squared magnitude of the directional derivative along the direction of minimum contrast.

When W is ill-conditioned, then $\lambda_{max} >> \lambda_{min}$ and the average magnitude of the gradient of the picture is much larger along the direction of maximum contrast than along the direction of minimum contrast (there is an edge). The converse of this statement is also true.

In order to generate a robust velocity estimate when W is ill-conditioned, we make use of the SVD representation of W. The first step is to demonstrate that W is always positive semi-definite (by construction it is symmetric). The result follows directly by considering the quantity

$$\bar{x}^T W \bar{x} = \frac{1}{N} \sum_{i=1}^{N} \left(\bar{x}^T \nabla_x s(\bar{x}, t)|_{P_i} \right)^2 \geq 0. \tag{3.29}$$

Because W is a symmetric positive semidefinite matrix, the SVD representation is equivalent to the eigenvalue/eigenvector decomposition

$$W = \lambda_{min} \bar{\phi}_{min} \bar{\phi}_{min}^T + \lambda_{max} \bar{\phi}_{max} \bar{\phi}_{max}^T. \tag{3.30}$$

where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of W, and $\bar{\phi}_{min}$ and $\bar{\phi}_{max}$ are the corresponding orthonormal eigenvectors. When W is singular $\lambda_{min} = 0$, and

$$W = \lambda_{max} \bar{\phi}_{max} \bar{\phi}_{max}^T. \tag{3.31}$$

When $\lambda_{max} = 0$, then all the entries of W are zero and the velocity field is completely unconstrained. This occurs when all the samples lie in a region where the spatial gradient of $s(\bar{x}, t)$ is identically zero.

Since the eigenvectors of W are orthonormal, the velocity vector can be written in the form

$$\bar{v} = \alpha_{max} \bar{\phi}_{max} + \alpha_{min} \bar{\phi}_{min}. \tag{3.32}$$

By direct substitution, it follows that

$$\alpha_{max} = \frac{\bar{\phi}_{max}^T \bar{\gamma}}{\lambda_{max}} \tag{3.33}$$

and

$$\alpha_{min} = \frac{\bar{\phi}_{min}^T \bar{\gamma}}{\lambda_{min}}. \tag{3.34}$$

Therefore $\bar{v}$ is computed as follows

$$\bar{v} = \begin{cases} \alpha_{max} \bar{\phi}_{max} & \text{if } \lambda_{max} >> \lambda_{min} \\ W^{-1} \bar{\gamma} & \text{otherwise} \end{cases} \tag{3.35}$$

In actual computation, the condition $\lambda_{max} >> \lambda_{min}$ was implemented as $\lambda_{max} > 25\lambda_{min}$. The choice of 25 corresponds to an average gradient ratio of 5. This choice is rather arbitrary and the algorithm is not sensitive to variations of this parameter.

3.3.2 Computing spatio-temporal gradients

In order to compute the velocity estimates, it is necessary to compute the matrix W and the vector $\bar{\gamma}$. These quantities depend on the spatio-temporal gradients of the picture. In this subsection we describe a method of computing these gradients. We should first note that the commonly used approach to estimating the gradients is to use pel differences for computing the spatial gradients and frame differences to compute the temporal gradients. This approach does not yield acceptable estimates because of two problems; aliasing and the presence of noise. Therefore we present an alternative approach.

Recall that we do not observe the continuous signal $s(\bar{x}, t)$ directly, but only noisy observations of discrete samples. Therefore there are two distinct problems which we need to address; sampling and the presence of noise. We first discuss the problem of sampling and then discuss the problem of additive noise.

Suppose we have samples $s[n_1, n_2, n_3]$ of the continuous signal $s(x, y, t)$, defined by

$$s[n_1, n_2, n_3] = s(n_1 T_x, n_2 T_y, n_3 T_t) \tag{3.36}$$

where T_x, T_y, T_t are the sampling intervals along the x, y, and t axes respectively. The problem is to use the available samples to compute the spatio-temporal gradients of $s(\bar{x}, t)$. There are two cases of interest (1) bandlimited and (2) wideband signals.

It is well known that a bandlimited signal can be reconstructed from samples if several conditions are satisfied. Specifically, suppose $s(x, y, t)$ is a bandlimited signal with Fourier transform

$$s(x, y, t) \Longleftrightarrow S(\omega_x, \omega_y, \omega_t). \tag{3.37}$$

For simplicity, assume $S(\omega_x, \omega_y, \omega_t)$ has a region of support in the interior of a

parallelepiped. Therefore,

$$S(\omega_x, \omega_y, \omega_t) = 0 \quad \text{if} \quad |\omega_x| > \Omega_x \quad \text{or} \quad |\omega_y| > \Omega_y \quad \text{or} \quad |\omega_t| > \Omega_t. \tag{3.38}$$

If the sampling rates satisfy the inequalities

$$T_x < \frac{1}{2\Omega_x} \qquad T_y < \frac{1}{2\Omega_y} \qquad T_t < \frac{1}{2\Omega_t} \tag{3.39}$$

then $s(x, y, t)$ can be recovered from the samples by the interpolation formula

$$s(x, y, t) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_3=-\infty}^{\infty} s[n_1, n_2, n_3] \phi(x, n_1, T_x) \phi(y, n_2, T_y) \phi(t, n_3, T_t) \tag{3.40}$$

where the interpolation kernel $\phi(z, n, T_z)$ is

$$\phi(z, n, T_z) \frac{\sin\left[\left(\frac{\pi}{T_z}\right)(z - nT_z)\right]}{\left(\frac{\pi}{T_z}\right)(z - nT_t)}. \tag{3.41}$$

It is also possible to compute the derivatives at arbitrary points using the interpolation formula. The partial derivatives of $s(x, y, t)$ are given by the following expressions

$$\frac{\partial s}{\partial x} = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_3=-\infty}^{\infty} s[n_1, n_2, n_3] \phi'(x, n_1, T_x) \phi(y, n_2, T_y) \phi(t, n_3, T_t) \tag{3.42}$$

$$\frac{\partial s}{\partial y} = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_3=-\infty}^{\infty} s[n_1, n_2, n_3] \phi(x, n_1, T_x) \phi'(y, n_2, T_y) \phi(t, n_3, T_t) \tag{3.43}$$

$$\frac{\partial s}{\partial t} = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_3=-\infty}^{\infty} s[n_1, n_2, n_3] \phi(x, n_1, T_x) \phi(y, n_2, T_y) \phi'(t, n_3, T_t) \tag{3.44}$$

In actual implementation the summations are restricted to some finite interval. Therefore, in principle it is possible to compute W and $\bar{\gamma}$, and solve Equation (3.8) to compute an estimate of the velocity field. These formulas do not yield satisfactory results when dealing with signals obtained from typical motion pictures for the following two reasons:

- The actual signals are not bandlimited. In the following paragraphs we illustrate that this is especially a problem along the temporal direction. Furthermore, the formulas require that many frames be used for computing the temporal gradients. Since in practice only two or three frames are typically used to compute the velocity estimates, there is significant error in using these formulas for computing the temporal gradients.

- The signals are corrupted with additive noise. Computing derivatives from the interpolation formula enhances the noise and results in very poor estimates of the derivatives.

The signals which we encounter in television communication systems are broadband and there is usually aliasing due to the sampling process. For typical motion picture sequences there is substantial aliasing along the temporal direction. In some simple cases it is possible to illustrate the severity of this problem. Consider a two-dimensional scene which is translating with some velocity $\bar{v}$. The signal model is

$$s(\bar{x}, t) = s_0(\bar{x} - \bar{v} \cdot (t - t_0)). \tag{3.45}$$

For this model there is a direct relationship between the 2-D Fourier transform of $s_0(\bar{x})$ and the 3-D Fourier transform of $s(\bar{x}, t)$. Specifically, if

$$s_0(\bar{x}) \Longleftrightarrow S_0(\omega_x, \omega_y) \tag{3.46}$$

and

$$s(\bar{x}, t) \Longleftrightarrow S(\omega_x, \omega_y, \omega_t) \tag{3.47}$$

then

$$S(\omega_x, \omega_y, \omega_t) = S_0(\omega_x, \omega_y) \exp\left[-j(\omega_x v_x + \omega_y v_y)t_0\right] \delta(\omega_x v_x + \omega_y v_y + \omega_t). \tag{3.48}$$

Now suppose $s_0(\bar{x})$ is bandlimited to some interval $|\omega_x| < \Omega_x$ and $|\omega_y| < \Omega_y$. From Equation (3.48) it follows that $s(\bar{x}, t)$ is bandlimited to the interval $|\omega_x| < \Omega_x$, $|\omega_y| < \Omega_y$, and $|\omega_t| < \Omega_t$, where

$$\Omega_t = \Omega_x |v_x| + \Omega_y |v_y|. \tag{3.49}$$

This relationship has some important implications. First note that if $v_x = v_y = 0$, then the temporal bandwidth of the signal is zero as one would expect. More generally, it follows that the temporal bandwidth increases linearly with the magnitude of the velocity. As a consequence, if the magnitude of the velocity exceeds unity, then the signal must be sampled faster along the temporal direction than the

spatial directions in order to avoid temporal aliasing. In present television systems the temporal sampling rate is relatively slow and temporal aliasing is inevitably present. Typically these signals are aliased along the spatial coordinates as well.

An alternative approach to estimating the spatial and temporal gradients of the signal $s(\bar{x}, t)$ from noisy, undersampled data, is to use a parametric signal model. In anticipation of computational simplicity, we suggest a general class of signal models which possess a linear relationship between the model parameters and the signal. The class of models which we propose are of the form

$$s(\bar{x}, t) \approx \tilde{s}(\bar{x}, t) = \sum_{i=1}^{N} S_i \phi_i(\bar{x}, t). \tag{3.50}$$

A model is specified by selecting the set of functions $\{\phi_i(\bar{x}, t)\}$. With this approach, the available samples are used to estimate the model parameters $\{S_i\}$ and the entries of W and $\bar{\gamma}$ are computed from the signal

$$\tilde{s}(\bar{x}, t) = \sum_{i=1}^{N} S_i \phi_i(\bar{x}, t). \tag{3.51}$$

It is important to emphasize that this signal model is used only for the purpose of motion estimation. Any subsequent processing of the motion picture uses the samples directly instead of the signal approximation based on the model.

It should be noted that the interpolation formula given by Equation (3.40) is a special case of this modeling approach. A wide variety of interpolation schemes can be formulated this way. In general an interpolation scheme makes some assumptions about the underlying signal. The "ideal interpolator" is based on the assumption that the signal is bandlimited and sampled in excess of the Nyquist rate. When it is known that the underlying signal is aliased due to the sampling process and perhaps also degraded, other interpolation schemes can yield better results (for example bilinear interpolation). We can think of this process more formally as a signal estimation problem. Our observations are the degraded samples and the desired output is a continuous signal representation. During the signal estimation phase we can account for the presence of noise.

Since the signal $\tilde{s}(\bar{x}, t)$ depends linearly on the coefficients $\{S_i\}$, determination of the coefficients which minimize the mean squared error between the signal $r(\bar{x}, t)$ and

$\tilde{s}(\bar{x}, t)$ involves solving a set of linear equations. Specifically, given the observation vector $\bar{r}$ with $r_i = r(\bar{x}_i, t_i)$, this relationship is

$$\min_{\bar{S}} \left\{ |A_{sl}\bar{S} - \bar{r}|^2 \right\} \quad \Longrightarrow \quad \bar{S} = (A_{sl}^T A_{sl})^{-1} A_{sl}^T \bar{r} = Q_{sl}\bar{r} \tag{3.52}$$

where

$$A_{sl} = \begin{bmatrix} \phi_1(\bar{x}_1, t_1) & \phi_P(\bar{x}_1, t_1) \\ \vdots & \vdots \\ \phi_1(\bar{x}_N, t_N) & \cdots \quad \phi_P(\bar{x}_N, t_N) \end{bmatrix}. \tag{3.53}$$

To complete the algorithm specification we must select a set of basis functions, $\{\phi_i(\bar{x}, t)\}$. In the experiments with this algorithm we have used a set of three-dimensional algebraic polynomials as the basis functions.

$$\begin{aligned} &\phi_1(\bar{x}, t) = 1 \quad \phi_2(\bar{x}, t) = x \quad \phi_3(\bar{x}, t) = y \\ &\phi_4(\bar{x}, t) = t \quad \phi_5(\bar{x}, t) = x^2 \quad \phi_6(\bar{x}, t) = y^2 \\ &\phi_7(\bar{x}, t) = xy \quad \phi_8(\bar{x}, t) = xt \quad \phi_9(\bar{x}, t) = yt \end{aligned} \tag{3.54}$$

The factors involved in this selection process include:

- A very small region of the three-dimensional signal space is being modeled. The samples are taken from a window with a small spatial extent (typically 5 x 5), from two frames.
- The model is overdetermined and we are not seeking an exact representation. This is important when there is noise in the images.
- The model is used to estimate the spatio-temporal gradients of the picture. It is well known that given noisy samples, curve fitting methods yield better gradient estimates than finite difference methods. If the samples are restricted to small regions, then polynomials are the natural choice of functions to use for curve fitting.

Once the parameters $\{S_i\}$ have been estimated from Equation (3.52), then the matrix W and the vector $\bar{\gamma}$ for the discrete point minimization can be computed as

$$W_{11} = (G_x\bar{S})^T G_x\bar{S} \tag{3.55}$$

$$W_{12} = W_{21} = (G_x\bar{S})^T G_y\bar{S} \tag{3.56}$$

$$W_{22} = (G_y\bar{S})^T G_y\bar{S} \tag{3.57}$$

$$\gamma_1 = -(G_x\bar{S})^T G_t\bar{S} \tag{3.58}$$

$$\gamma_2 = -(G_y\bar{S})^T G_t\bar{S} \tag{3.59}$$

where

$$G_x = \begin{bmatrix} \left.\frac{\partial\phi_1}{\partial x}\right|_{P_1} & \cdots & \left.\frac{\partial\phi_P}{\partial x}\right|_{P_1} \\ \vdots & & \vdots \\ \left.\frac{\partial\phi_1}{\partial x}\right|_{P_N} & \cdots & \left.\frac{\partial\phi_P}{\partial x}\right|_{P_N} \end{bmatrix} \tag{3.60}$$

$$G_y = \begin{bmatrix} \left.\frac{\partial\phi_1}{\partial y}\right|_{P_1} & \cdots & \left.\frac{\partial\phi_P}{\partial y}\right|_{P_1} \\ \vdots & & \vdots \\ \left.\frac{\partial\phi_1}{\partial y}\right|_{P_N} & \cdots & \left.\frac{\partial\phi_P}{\partial y}\right|_{P_N} \end{bmatrix} \tag{3.61}$$

$$G_t = \begin{bmatrix} \left.\frac{\partial\phi_1}{\partial t}\right|_{P_1} & \cdots & \left.\frac{\partial\phi_P}{\partial t}\right|_{P_1} \\ \vdots & & \vdots \\ \left.\frac{\partial\phi_1}{\partial t}\right|_{P_N} & \cdots & \left.\frac{\partial\phi_P}{\partial t}\right|_{P_N} \end{bmatrix} \tag{3.62}$$

For the continuous region minimization, the integral is evaluated over a unit cube in the three-dimensional space. Therefore,

$$\int\int\int_{\Psi} dx\,dy\,dt \Longleftrightarrow \frac{1}{8}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} dx\,dy\,dt. \tag{3.63}$$

This range of integration was selected because it is compatible with the size of analysis windows used for obtaining the observation vector. This results in the following values for W and $\bar{\gamma}$

$$W_{11} = S_2^2 + \frac{1}{3}\left(4S_5^2 + S_7^2 + S_8^2\right) \tag{3.64}$$

$$W_{12} = W_{21} = S_2S_3 + \frac{1}{3}(2S_5S_7 + 2S_6S_7 + S_8S_9) \tag{3.65}$$

$$W_{22} = S_3^2 + \frac{1}{3}\left(4S_6^2 + S_7^2 + S_9^2\right) \tag{3.66}$$

$$\gamma_1 = -S_2S_4 - \frac{1}{3}(2S_5S_8 + S_7S_9) \tag{3.67}$$

$$\gamma_2 = -S_3S_4 - \frac{1}{3}(2S_6S_9 + S_7S_8) \tag{3.68}$$

Some preliminary experiments were performed in order to compare the discrete point and continuous region minimization procedures. It was found that there is no significant difference between these two methods. The motion estimation error as a function of signal-to-noise levels was measured to be essentially identical. The continuous region minimization is used exclusively in the remaining experiments.

3.4 Maximum likelihood motion estimation

The direct form of the motion model is the basis for a maximum likelihood motion estimation algorithm. This form is expressed by the relationship

$$s(\bar{x}, t) = s_0(\bar{x} - \bar{v}\ \ (t - t_0)). \tag{3.69}$$

The maximum likelihood method is a procedure for estimating unknown parameters from a set of observations. For the motion estimation problem, the parameters are the two components of the velocity field. The observations are samples of a given signal $r(\bar{x}, t)$.

This application is a special case of maximum likelihood because the available data consists of noisy observations of an unknown signal $s(\bar{x}, t)$, which is a function of the parameters v_x and v_y and the image $s_0(\bar{x})$. If the image $s_0(\bar{x})$ were known, then there is a straightforward formulation for the maximum likelihood velocity estimates. Since this is not the case, we will represent $s_0(\bar{x})$ in terms of a set of parameters and find the maximum likelihood estimates of the velocity parameters as well as the signal parameters that are used to model $s_0(\bar{x})$.

Note that the signal parameters are not desired and are referred to as "unwanted parameters" [36]. There are several methods for dealing with unwanted parameters. If the probability density of these parameters is known, then one can determine the true maximum likelihood estimates of the velocity components by integrating the marginal density governing the observations over the probability density of the signal parameters. In our application this is not possible, so we will obtain maximum likelihood estimates of both sets of parameters.

Suppose a given frame can be expressed in the following form

$$s_0(\bar{x}) = \sum_{i=1}^{P} S_i \phi(\bar{x}) \tag{3.70}$$

over a region in the neighborhood of some point $\bar{x}_0$. The set of functions $\phi_i(\bar{x})$ form the basis for the signal space. A signal is represented by the vector $\bar{S} = (S_1, S_2, \ldots, S_P)^T$. This signal representation reduces the two-dimensional signal

space into a finite dimensional vector space. The velocity constraint

$$s(\bar{x}, t) = s_0(\bar{x} - \bar{v} \cdot (t - t_0)) \tag{3.71}$$

together with the signal model given by Equation (3.70), leads to the parametric signal

$$s(\bar{x}, t) = \sum_{i=1}^{P} S_i \phi_i(\bar{x} - \bar{v} \cdot (t - t_0)). \tag{3.72}$$

Now that the signal has been represented parametrically, we want to obtain the maximum likelihood estimates for the vectors $\bar{S}$ and $\bar{v}$. The observation model for this estimator is

$$r(\bar{x}, t) = s(\bar{x}, t) + n(\bar{x}, t) \tag{3.73}$$

where $n(\bar{x}, t)$ is a zero-mean, white Gaussian noise process with variance σ_n^2.

Given N discrete observations

$$\begin{aligned} r_1 &= r(x_1, y_1, t_1) \\ r_2 &= r(x_2, y_2, t_2) \\ &\vdots \\ r_N &= r(x_N, y_N, t_N) \end{aligned} \tag{3.74}$$

define the observation vector

$$\bar{r} = (r_1, r_2, \ldots, r_N)^T. \tag{3.75}$$

In a similar fashion the signal and noise vectors are defined as

$$\bar{s} = (s_1, s_2, \ldots, s_N)^T \tag{3.76}$$

$$\bar{n} = (n_1, n_2, \ldots, n_N)^T \tag{3.77}$$

so that

$$\bar{r} = \bar{s} + \bar{n}. \tag{3.78}$$

The maximum likelihood estimator determines the parameter values which maximize the likelihood of the observations.

Since the noise field is white and Gaussian, it follows that $p(\bar{n})$, the probability density of $\bar{n}$ is

$$p(\bar{n}) = \left(\frac{1}{\sqrt{2\pi}\sigma_n}\right)^N \exp\left(-\frac{1}{2\sigma_n^2}\sum_{i=1}^{N} n_i^2\right) \tag{3.79}$$

Therefore the likelihood function for the observation vector $\bar{r}$ is

$$p(\bar{r}) = \left(\frac{1}{\sqrt{2\pi}\sigma_n}\right)^N \exp\left(-\frac{1}{2\sigma_n^2}\sum_{i=1}^{N}(r_i - s_i)^2\right). \tag{3.80}$$

Substituting the signal model given by Equation (3.72), into this expression yields the likelihood function governing the observation vector $\bar{r}$

$$p(\bar{r}) = \left(\frac{1}{\sqrt{2\pi}\sigma_n}\right)^N \exp\left(-\frac{1}{2\sigma_n^2}\sum_{i=1}^{N}\left(r_i - \sum_{j=1}^{P} S_j\phi_j(\bar{x}_i - \bar{v}\cdot(t_i - t_0))\right)^2\right). \tag{3.81}$$

Maximizing the likelihood function $p(\bar{r})$ is equivalent to minimizing the euclidean distance function $\lambda(\bar{S}, \bar{v})$ defined as

$$\lambda(\bar{S}, \bar{v}) = \sum_{i=1}^{N}\left(r_i - \sum_{j=1}^{P} S_j\phi_j(\bar{x}_i - \bar{v}\cdot(t_i - t_0))\right)^2. \tag{3.82}$$

The distance function is nonlinear in the unknowns $\bar{S}$ and $\bar{v}$, and a known closed-form solution does not exist for arbitrary basis functions. One method for minimizing the distance function is to apply a nonlinear optimization procedure and solve for all the parameters at once. This method does not perform well because often it converges to a local minimum of the objective function which is not a good estimate of the parameters. Furthermore, this method is very costly in terms of computational requirement.

We propose an alternative approach that is similar in style to the EM algorithm [6]. The EM algorithm is an efficient optimization procedure used to determine maximum likelihood parameter estimates from noisy or incomplete data. Iterative algorithms of this type have been analyzed extensively by Musicus [23]. This algorithm is motivated by recognizing that there is a natural division between the

signal and velocity parameters. Let

$$A_{ml}(\bar{v}) = \begin{bmatrix} \phi_1(\bar{x}_1 - \bar{v}\cdot(t_1 - t_0)) & \cdots & \phi_P(\bar{x}_1 - \bar{v}\cdot(t_1 - t_0)) \\ \vdots & & \vdots \\ \phi_1(\bar{x}_N - \bar{v}\cdot(t_N - t_0)) & \cdots & \phi_P(\bar{x}_N - \bar{v}\cdot(t_N - t_0)) \end{bmatrix} \tag{3.83}$$

so that

$$\lambda(\bar{S}, \bar{v}) = \left|\bar{r} - A_{ml}(\bar{v})\bar{S}\right|^2. \tag{3.84}$$

Suppose we hold the velocity vector $\bar{v}$ fixed and solve for the vector $\bar{S}$ which minimizes the distance function. Since the distance function is quadratic in $\bar{S}$, this problem reduces to solving an overdetermined set of linear equations. Next, suppose we hold $\bar{S}$ fixed and minimize the distance function over variations in $\bar{v}$. The distance function can be minimized with the same optimization procedure as applied in the region matching estimator described in Appendix B. We can summarize these two steps in the following manner:

Signal estimation:

$$\min_{\bar{S}}\{\lambda(\bar{S}, \bar{v})\} \quad \Longrightarrow \quad \bar{S} = [A_{ml}(\bar{v})^T A_{ml}(\bar{v})]^{-1} A_{ml}(\bar{v})^T \bar{r}. \tag{3.85}$$

Velocity estimation:

$$\min_{\bar{v}}\{\lambda(\bar{S}, \bar{v})\} \quad \Longrightarrow \quad \min_{\bar{v}} \left|\bar{r} - A_{ml}(\bar{v})\bar{S}\right|^2. \tag{3.86}$$

Figure 3.3 illustrates the operations performed by the algorithm at each point where a velocity estimate is required. Appendix D discusses the convergence properties of this algorithm.

It is important to distinguish the signal model used in the maximum likelihood estimator from the one used in the least squares estimator. The model used in the maximum likelihood estimator specified by Equation (3.70) is a two-dimensional representation that applies to a small region of a single frame. In contrast, the model specified by Equation (3.50) is a three-dimensional representation that applies to a small region of a set of frames.

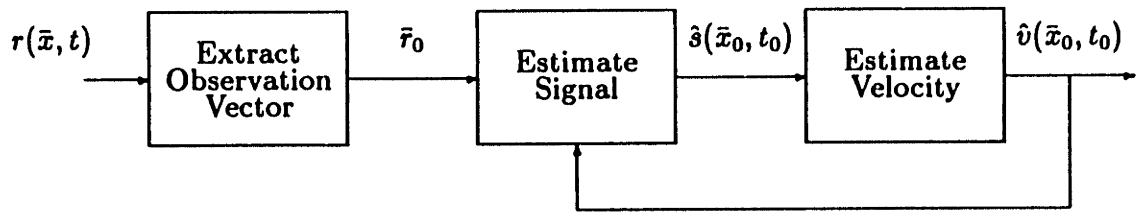


Figure 3.3: Maximum likelihood motion estimation algorithm

Both models are linear in the signal parameters, but there is an important difference between the computation involved in these two models. In the maximum likelihood estimator, the matrix $A_{ml}(\bar{v})$ in Equation (3.84) is a function of the velocity vector. Therefore as the iteration progresses, it is necessary to compute this matrix at arbitrary values of $\bar{v}$. In contrast, the matrix A_{ll} in Equation (3.52) is constant for a given sampling lattice. Therefore Q_{ll} can be computed off line once and for all, so that computing the signal vector $\bar{S}$ reduces to a matrix-vector multiply. This computation dominates the overall computational requirement of the algorithm.

3.4.1 Selection of model basis functions

In order to complete the description of the maximum likelihood estimator, it is necessary to specify the model basis functions $\phi_i(\bar{x})$. There are several considerations involved in selecting these functions.

- The first consideration is the spatial extent of the region which is being modeled. In a wide variety of situations it was found that a 5 x 5 window yields the best tradeoffs between resolution and accuracy. Windows smaller than this tend to yield large motion estimation errors, while windows larger than this yield unsatisfactory spatial resolution. This is a very small region relative to the size of the images being processed.

- The computational requirement of the algorithm is affected directly by the computational requirement of the basis functions needed in computing $A_{ml}(\bar{v})$.

- Finally, it is necessary to guarantee that the rank of $A_{ml}(\bar{v})$ is equal to P (the number of basis functions) for all $\bar{v}$.

Based on these requirements we have selected a set of two-dimensional algebraic polynomials as the basis functions. A second-order polynomial was used so that the basis functions can be written as

$$\begin{aligned} \phi_1(x,y) = 1 \quad \phi_2(x,y) = x \quad \phi_3(x,y) = y \\ \phi_4(x,y) = x^2 \quad \phi_5(x,y) = y^2 \quad \phi_6(x,y) = xy \end{aligned} \tag{3.87}$$

This set of basis functions models a small region of the image with a second-order Taylor series expansion. The computational requirement for these functions is minimal. In addition, shifted algebraic polynomials are always linearly independent over a rectangular lattice. Therefore $A_{ml}(\bar{v})$ will always have full rank for all finite $\bar{v}$.

It was experimentally determined that typically only 4 iterations are sufficient to achieve convergence with this choice of basis functions. Each iteration requires evaluating both the signal and velocity estimates. This quantity is used in obtaining the operation count for this estimator.

3.5 Motion modeling error

The motion models we have been using assume the velocity field is continuous and the signal is constant along the field lines. There are two important cases that occur in real pictures which violate these assumptions. As a moving object uncovers the background portion of a picture, the velocity field in these regions is not defined. More generally, when there is a scene change such that two temporally adjacent frames are entirely different, the velocity field in between the frames is not defined. In these cases there is large modeling error and this condition must be detected by the motion estimation algorithms.

Both of the algorithms which are described in the preceding sections have the same generic structure, involving the minimization of an objective function. At the end of the minimization phase, the value of the objective function at the optimal velocity estimate is a measure of the modeling error. We refer to this value as the residual. By comparing the residual to a threshold, it is possible to detect when the motion model is not appropriate for the given region of the picture.

More formally, suppose we have an objective function $f(\bar{v})$ and the "optimal" velocity vector $\bar{v}^*$ such that the residual $f(\bar{v}^*)$ is minimal in some sense. The objective functions we have developed are always nonnegative. Furthermore, if the residual is zero, there is no modeling error. Large residual values indicate large modeling error, which can arise from two possibilities; either the model does not apply to the region of the picture or the signal-to-noise level is very low. Therefore it is necessary to match the threshold to the noise level in the form of a likelihood ratio test (if $f(\bar{v}^*) < \gamma(\sigma_n^2)$ then the model applies, otherwise the two regions are incompatible).

A test of this form was used in the frame interpolation system we developed. When the regions were determined to be incompatible, a zero velocity field was assumed. This allowed uncovered regions to be projected properly onto the interpolated frame.

3.6 Computational complexity

For real-time applications, the computational complexity of the motion estimation algorithms is extremely important. Therefore the purpose of this section is to compare the computational complexity of the two algorithms described in the previous sections and the region matching algorithm described in Appendix B. The particular measure of computational complexity which we use is the number of primitive arithmetic operations (addition, subtraction, multiplication, and division). This analysis assumes that the spatial extent of the analysis window is 5 x 5 and the velocity estimate is computed at a time instant midway between two frames so that $|t_i - t_0| = 1$ for all samples $r(\bar{x}, t_i)$. These assumptions correspond to the parameters used in the experiments described in the next chapter. The iterative algorithms require a random number of iterations. In the same experiments we determined the average number of iterations required for convergence. These averages are used in computing the operation count for the iterative algorithms.

3.6.1 Computational complexity of least squares

There are two computational tasks associated with the least squares algorithm; signal estimation and velocity estimation.

The signal estimation phase involves solving a set of linear equations. Most of the computation can be done off line and the net computation is a matrix by vector multiply

$$\bar{S} = Q\bar{r} \tag{3.88}$$

where Q is a 9 x 50 matrix. Computing the vector $\bar{S}$ requires 891 operations.

The velocity estimation phase involves solving a 2 x 2 set of linear equations

$$W\bar{v} = \bar{\gamma}. \tag{3.89}$$

Refering to Equations (3.64), computing the entries of W requires 36 operations and computing the entries of $\bar{\gamma}$ requires 14 operations. Given W and $\bar{v}$, computing $\bar{v}$ requires 10 operations. Therefore the operation count for computing the velocity

estimate given the signal estimate is 60. The total operation count for the least squares algorithm is 951.

3.6.2 Computational complexity of maximum likelihood

There are two computational tasks associated with the maximum likelihood algorithm; signal estimation and velocity estimation.

Given a velocity estimate, computing a signal estimate involves solving an overdetermined set of linear equations

$$A_{ml}\bar{S} \approx \bar{r}. \tag{3.90}$$

The least squares solution is obtained from the normal equations

$$(A_{ml}^T A_{ml})^{-1} = A_{ml}^T \bar{r}. \tag{3.91}$$

The first step is to compute A_{ml}. Refering to Equation (3.83), each row of A_{ml} requires that we evaluate $\phi_i(\bar{x} - \bar{v}\delta t)$ for $i = 1, \ldots, 6$. Since $|t_i - t_0| = 1$, evaluating the argument for each row requires 2 operations and evaluating the $\phi_i(\cdot)$ requires 3 operations. Therefore, computing each row requires 5 operations. Since there are 50 rows, the total operation count for computing A_{ml} is 250.

Computing $A_{ml}^T A_{ml}$ requires 3861 operations and computing $A_{ml}^T \bar{r}$ requires 594 operations. Finally, solving the set of linear equations by Gaussian elimination (without partial pivoting) requires 206 operations. Therefore the total operation count for the signal estimation phase is 4911.

The computational requirement for the velocity estimation phase is dominated by the evaluation of the objective function, which is the residue associated with the least squares approximation

$$(A_{ml}\bar{S} - \bar{r})^2. \tag{3.92}$$

As before, the total operation count required to evaluate A_{ml} is 250. Given A_{ml}, the number of operations required to compute the residue is 699. Therefore the total operation count for evaluating the objective function is 949. On the average, the

objective function was evaluated 52 times. Therefore the total operation count for the velocity estimation phase is 49348.

The total operation count for each outer loop iteration of the estimator is 54259. Since the loop is executed 4 times, the total operation count for each velocity estimate with the maximum likelihood estimator is 217036.

3.6.3 Computational complexity of region matching

Virtually all the computation associated with the region matching algorithm occurs in computing the objective function. Therefore we begin with an assessment of the operation count for evaluating the objective function.

Values of the signal $r(\bar{x}, t)$ which are not on the sampling grid are computed with a bilinear interpolator. Refering to Equation (B.5) we can see that this requires 13 operations. Computing the interpolation position requires 2 operations. Therefore, computing each signal value requires 17 operations. For a 5 x 5 window there are 50 signal values which are needed. Therefore a total of 50 x 15 = 750 operations are required for computing all the signal values.

Given the signal values, computing the objective function requires 25 differences, 25 squares, and 24 additions. The total operation count for evaluating the objective function is 750 + 25 + 25 + 24 = 824.

In general, the objective function must be evaluated some random number of times before the iteration terminates. It was experimentally determined that on the average, the objective function was evaluated 62 times. Therefore the average operation count for each velocity estimate is 62 x 824 = 51088.

It should be noted that numerous simplifications of the algorithm can lower the operation count. For example, if we restrict the spatio-temporal positions at which velocity estimates are computed to lie on the sampling grid, then only 25 signal values need to be computed instead of 50. This almost reduces the total operation count by a factor of two. Other simplifications have been proposed by Netravali and Robbins [25,24], however these simplifications will in general increase the motion estimation error.

3.6.4 Summary of computational complexity

The computational requirement for the three motion estimation algorithms is summarized in the following table.

Simple Arithmetic Operation Count (add,sub,mul,div)		
	Total Count	Normalized Count
Least squares	951	1
Maximum likelihood	217036	228
Region matching	51088	54

This table shows that the least squares algorithm requires substantially less computation than the maximum likelihood and region matching algorithms.

3.7 Averaging velocity estimates

It is usually possible to decrease the motion estimation error generated with these algorithms by averaging the estimates obtained in the neighborhood of each pel. The algorithms obtain an estimate of the velocity field at each pel, independent from the estimates obtained at neighboring pels. Very often the velocity field is inconsistent, in the sense that it varies faster than the picture sampling rate would normally permit. These inconsistencies are minimized by averaging the estimates so that the overall velocity field varies smoothly.

The most simple averaging strategy is to form an unweighted average of the velocity estimates in the neighborhood of a point of interest. We have restricted the regions to rectangular windows, centered about the point of interest. Either a 3 x 3 or a 5 x 5 window is used. Therefore, an averaged estimate is obtained as

$$\bar{v}_{avg} = \frac{1}{N} \sum_{i=1}^{N} \bar{v}_i \tag{3.93}$$

where the set $\{\bar{v}_i\}$ are taken from the window centered about the pel of interest.

A potentially better averaging strategy is to include a weight for each term of the sum

$$\bar{v}_{avg} = \sum_{i=1}^{N} w_i \bar{v}_i \tag{3.94}$$

where the weights w_i sum to unity. The question then arises how to select the weights. There is a natural choice which is obtained by seeking the weights that minimize the estimation error. Suppose we have a set of zero-mean random variables, $(x_1, x_2, \ldots, x_N) = \bar{x}^T$ with covariance Λ_x where

$$\Lambda_x = E[\bar{x}\bar{x}^T]. \tag{3.95}$$

We seek a weighting vector $\bar{w}$, such that the quantity

$$E[(\bar{w}^T \bar{x})^2] = \bar{w}^T \Lambda_x \bar{w} \tag{3.96}$$

is minimized, subject to the constraint

$$\bar{w}^T \bar{1} = 1 \tag{3.97}$$

where $\bar{1} = (1, 1, \ldots, 1)^T$. The constraint requires that the weights sum to unity. The optimal weights can be found easily by introducing a Lagrange multiplier λ, and minimizing the Lagrangian with respect to $\bar{w}$

$$\min_{\bar{w}} \left\{ \bar{w}^T \Lambda_x \bar{w} + \lambda(1 - \bar{w}^T \bar{1}) \right\}. \tag{3.98}$$

The optimal weights are given by

$$\bar{w} = \lambda \Lambda_x^{-1} \bar{1}. \tag{3.99}$$

The Lagrange multiplier λ is chosen so as to satisfy Equation (3.97). A special case of interest is when the covariance matrix is diagonal

$$\Lambda_x = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_N^2 \end{bmatrix}, \tag{3.100}$$

so that the elements of $\bar{x}$ are uncorrelated. In this case, $\bar{w}$ is given by

$$\bar{w} = \left(\sum_{i=1}^{N} \sigma_i^2 \right) \begin{bmatrix} \frac{1}{\sigma_1^2} \\ \vdots \\ \frac{1}{\sigma_N^2} \end{bmatrix} \tag{3.101}$$

This method of covariance-weighted averaging can be used with the least squares estimator. One of the by-products of the computation for the least squares estimator is an estimate of the variance of the error. In Section 3.10 we show that the Fisher information matrix from which the Cramer Rao bounds are obtained is proportional to the matrix W. The eigenvectors of W point in the directions of minimum and maximum contrast, and the eigenvalues are the mean squared gradient over the region of interest. Therefore the eigenvalues are inversely proportional to the estimation error along the directions of the eigenvectors. Based on these facts and the result presented in Equation (3.101), we have used the sum of the eigenvalues as the weights. Each velocity estimate is weighted by $\lambda_{min} + \lambda_{max}$, which is

computed in the process of obtaining the estimates. This only applies to the least squares estimator.

It is important to contrast this weighted averaging strategy with unweighted averaging. Consider a region with high contrast which is adjacent to a region with low contrast. In the region of low contrast the estimation error is likely to be large and in the region of high contrast the estimation error is likely to be relatively small. A simple averaging strategy will corrupt the estimates obtained in the high contrast region. Conversely, with the weighted averaging strategy, the estimates obtained in the regions of high contrast dominate the resulting averaged estimate, resulting in improved performance.

3.8 Displaced analysis windows

The algorithms described in the previous sections operate on a small number of samples in the neighborhood of a point where a velocity estimate is desired. If the velocity of an object is sufficiently large, then the displacement field can exceed the size of the analysis window. The algorithms are unable to generate accurate estimates when this occurs.

In order to permit estimation of large velocity fields, these algorithms all operate on displaced frames. Suppose we have an a priori estimate of the velocity field at the point of interest [2]. Based on this estimate, the frames are displaced so that the analysis window is centered approximately around the initial displacement field estimate. If the initial velocity estimate is $\bar{v}$, then the displacement field is decomposed into two portions

$$\bar{D} = \begin{pmatrix} v_x \delta t \\ v_y \delta t \end{pmatrix} = \begin{pmatrix} D_x \\ D_y \end{pmatrix} = \begin{pmatrix} Int(D_x) + Frac(D_x) \\ Int(D_y) + Frac(D_y) \end{pmatrix}. \tag{3.102}$$

In this expression, $Int(\cdot)$ represents the integer closest to the real number $(\cdot)$ and $Frac(\cdot)$ is the difference between the real number $(\cdot)$ and $Int(\cdot)$. Therefore the window at time instant $t_0 + \delta t$ is displaced by $[Int(D_x), Int(D_y)]$ samples and the initial velocity estimate in the displaced window becomes

$$\bar{v}_{new} = \begin{pmatrix} \frac{Frac(D_x)}{\delta t} \\ \frac{Frac(D_y)}{\delta t} \end{pmatrix}. \tag{3.103}$$

[2] In the experiments where we measure motion estimation error, the initial estimates are randomly generated. The multigrid algorithm generates initial estimates based on an estimation procedure at a coarser grid.

3.9 Multigrid motion estimation

In all the algorithms described in the previous sections, the largest displacement field which can be determined reliably is some fraction of the analysis window size. The window sizes which yield suitable accuracy and resolution tradeoffs are smaller than typical displacement fields which are encountered in broadcast television systems. This has prompted the development of a multigrid motion estimation algorithm which we describe in this section.

The goal of this multigrid algorithm is to permit determination of large velocities, with both high resolution and high accuracy. This algorithm operates as follows. The velocity field is determined over the entire image on a coarse grid. The coarse grid is obtained from the original frames by down sampling the images. Down sampling the images has the property of contracting the velocity field. Large velocities in the original frame become small velocities in the down-sampled frames. If the original frames are $s_0(\bar{x}, t)$ and the velocity field is $\bar{v}_0(\bar{x}, t)$, then the velocity field for a down-sampled signal $s_d(\bar{x}, t)$ is

$$\bar{v}_d(\bar{x}, t) = \frac{\bar{v}_0(\bar{x}, t)}{d_s} \tag{3.104}$$

where

$$s_d(\bar{x}, t) = s(d_s \bar{x}, t) \tag{3.105}$$

and d_s is the down-sampling factor which is greater than one. The velocity field in the down-sampled frames is estimated with one of the algorithms described in the previous section. In the next stage, the coarse velocity field is interpolated to generate initial estimates of the velocity field at a finer grid. After the velocity field is estimated at one grid level, the velocity samples are averaged prior to interpolation. A bilinear interpolator is used to interpolate the velocity field to obtain the initial estimates at a finer grid. This process is repeated several times at successively finer grids. Figure 3.4 illustrates this procedure.

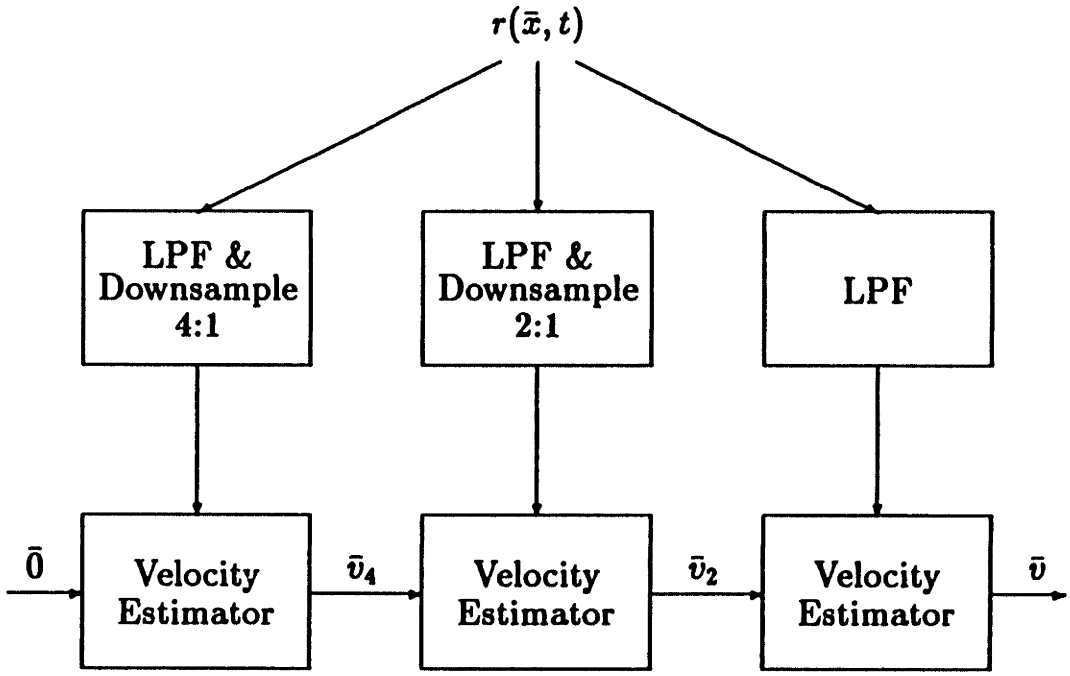


Figure 3.4: Multigrid motion estimation

It is necessary to apply a spatial filter prior to down sampling the images in order to avoid aliasing. For computational efficiency, we have used a separable filter, so that

$$h_d[n_1, n_2] = h[n_1]h[n_2] \tag{3.106}$$

where $h[n]$ is a one-dimensional low-pass filter. The impulse response of the one-dimensional filter is obtained by windowing an ideal low-pass filter with a hamming window. Each down-sampling filter requires specification of a cutoff frequency, ω_c. There are two quantities which specify the cutoff frequency; the down sampling factor (d_s) and the fraction of the bandwidth of the image which is to be retained (b_f). The latter parameter allows the down-sampled image to be low-pass filtered for noise suppression. With these parameters the impulse response of the ideal low-pass filter is

$$h_{ideal}[n] = \left(\frac{n}{\pi}\right) \sin\left(\frac{\pi b_f n}{d_s}\right). \tag{3.107}$$

The size of the hamming window for a given down-sampling factor was set equal to $12d_s + 1$. The choice of these parameters is not crucial. This particular choice yields good tradeoffs between computational requirement and accurate frequency response.

3.10 Bounds on motion estimation accuracy

When the signals are degraded with additive white Gaussian noise, it is possible to calculate a lower bound on the accuracy which any unbiased estimator can achieve. The bounds are related to the problem of determining a uniform velocity field characterized by the vector $\bar{v}$, from discrete observations of the signal $r(\bar{x}, t)$. These bounds are derived in Appendix C. In this section we describe the relationship between the bounds and various parameters related to signals and estimation algorithms.

There are no highly restrictive assumptions which are required in deriving the bounds. They apply directly to the least squares and maximum likelihood algorithms described in the previous sections and to the region matching algorithm described in Appendix B. The specific assumptions used in deriving the bounds and the consequences of these assumptions are itemized below:

- The signal $s(\bar{x}, t)$ is known at some time instant t_0:

  This assumption is never true, and the consequence is that the bounds are optimistic. Therefore we expect that the bounds can never be achieved by any algorithm. The maximum likelihood algorithm estimates the signal as well as the velocity. The experimentally determined motion estimation error for this algorithm is always larger than the bound. However, if we omit the signal estimation phase of the algorithm and substitute the exact signal, the experimentally determined estimation error is equal to the bound.

- The noise field is zero-mean, white Gaussian noise, and is uncorrelated with the signal:

  The consequences of violating any of these assumptions renders the bounds inapplicable. There are a wide variety of scenarios where the stated assumptions are satisfied approximately. The frequent occurrence of these scenarios provides the motivation for deriving the bounds.

- The first-order partial derivatives of the signal $s(\bar{x}, t)$ exist:

This assumption is justified in all realistic imaging systems.

The Cramer Rao bounds are expressed in terms of the Fisher information matrix, which is shown to be

$$J = \frac{1}{\sigma_n^2}\begin{bmatrix} \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)^2 (t_i - t_0)^2 & \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)\left(\frac{\partial s_i}{\partial y}\right)(t_i - t_0)^2 \\ \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)\left(\frac{\partial s_i}{\partial y}\right)(t_i - t_0)^2 & \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial y}\right)^2 (t_i - t_0)^2 \end{bmatrix} \tag{3.108}$$

The quantities $\{s_i\}$ correspond to sampling instants of the signal, so that

$$s_i = s(x_i, y_i, t_i) \qquad i = 1 \cdots N. \tag{3.109}$$

The bounds are expressed in terms of the Fisher information matrix and are given by

$$Var[\hat{v}_x - v_x] \geq \frac{J_{22}}{|J|} \tag{3.110}$$

and

$$Var[\hat{v}_y - v_y] \geq \frac{J_{11}}{|J|}. \tag{3.111}$$

There is a direct relationship between the Fisher information matrix and the matrix W found in the least squares motion estimation algorithm (discrete point minimization). Let the observation samples be taken at time $|t_i - t_0| = \delta t$. The Fisher information matrix becomes

$$J = \frac{\delta t^2}{\sigma_n^2}\begin{bmatrix} \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)^2 & \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)\left(\frac{\partial s_i}{\partial y}\right) \\ \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)\left(\frac{\partial s_i}{\partial y}\right) & \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial y}\right)^2 \end{bmatrix} = \frac{N\delta t^2}{\sigma_n^2} W. \tag{3.112}$$

Note that when W is singular, the Fisher information matrix is also singular. When this occurs, the bound for the component of the velocity in the direction of the edge becomes infinite, as expected. However, the bound for the component of the velocity field orthogonal to the direction of the edge remains finite.

Suppose there is an edge oriented along the y direction. The bound for the velocity along the x direction is

$$Var[\hat{v}_x - v_x] \geq \frac{\sigma_n^2}{\sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)^2 (t_i - t_0)^2}. \tag{3.113}$$

This is an important case and we will focus our attention on this result.

The first observation which we make is that the bound is proportional to the noise variance. Therefore the standard deviation of the motion estimation error is proportional to the standard deviation of the noise field.

If the signal gradient remains constant over the region of interest,

$$\frac{\partial s_i}{\partial x} = G_x \tag{3.114}$$

and the error standard deviation is proportional to $\frac{1}{G_x}$.

The algorithms described in the next chapter assume the samples of the signal are taken from two frames at times $t = t_0 + \delta t$ and $t = t_0 - \delta t$. For this case, $(t_i - t_0)^2 = \delta t^2$ and the error standard deviation is proportional to $\frac{1}{\delta t}$. This implies that the bound for the velocity field approaches zero as $\delta t \rightarrow \infty$. However, the displacement field is given by

$$\bar{d} = \bar{v}\delta t. \tag{3.115}$$

Therefore the error in the displacement field is independent of δt.

Finally, when the signal gradient remains constant and the sampling instants satisfy $(t_i - t_0)^2 = \delta t^2$, then the error standard deviation is proportional to $\frac{1}{N}$. Recall that N is the number of samples used to form the estimate. By way of contrast, suppose the samples lie on a rectangular grid of size $\sqrt{N}$ x $\sqrt{N}$ and there is an edge whose extent is less than $\sqrt{N}$ samples wide. For this case the error standard deviation is proportional to $\frac{1}{\sqrt{N}}$. This scenario occurs very frequently in actual practice and this result illustrates that typically only modest improvement in motion estimation accuracy can be obtained by increasing the window size. Alternatively, the number of samples in the window must increase by a large amount in order to yield a significant decrease in motion estimation error.

Chapter 4

Motion estimation experiments

In this chapter we present some experimental results which compare the motion estimation algorithms described in the previous chapter. These comparisons are based on both objective and subjective measures.

The first set of experiments were designed to measure the RMS motion estimation error as a function of signal-to-noise level. We present some data which relates the error to noise levels for these algorithms. In addition we present histograms which illustrate how these errors are distributed statistically. These measurements were made with several synthetic test images. The second set of experiments provide a subjective comparison of the algorithms. This is accomplished by frame averaging with temporal filters oriented along estimated motion trajectories. In all these experiments the spatial analysis window size was fixed to be 5 x 5.

The last set of experiments deal with the problem of estimating large velocity fields. For these experiments we used real images with synthetic velocity fields and controlled noise levels. These experiments demonstrate the effectiveness of the multigrid algorithm for estimating large velocities.

4.1 Measuring motion estimation error

Several synthetic test sequences were used to measure motion estimation error as function of signal-to-noise level. The first sequence consists of two frames with a set of uniform gradient edges. This sequence was selected because it fits the signal models exactly and the bound for the estimation error can be computed directly. Figure 4.1 contains one frame of this sequence. This frame contains a set of regions

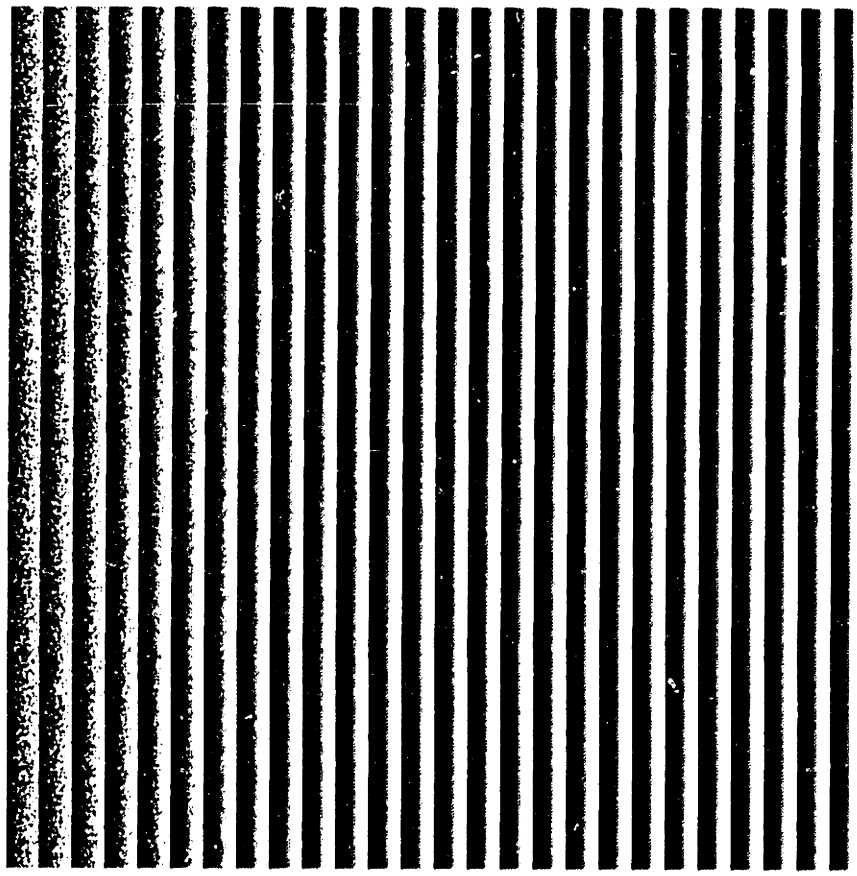


Figure 4.1: Uniform gradient edges

where the intensity increases linearly along the x direction. Therefore the gradient is uniform in these regions. Figure 4.2 illustrates the horizontal cross section of one of these edges. The left edge of the picture contains the highest noise level and the noise level decreases as one progresses from left to right. Both the noise and gradient were controlled to permit experimental measurement of motion estimation

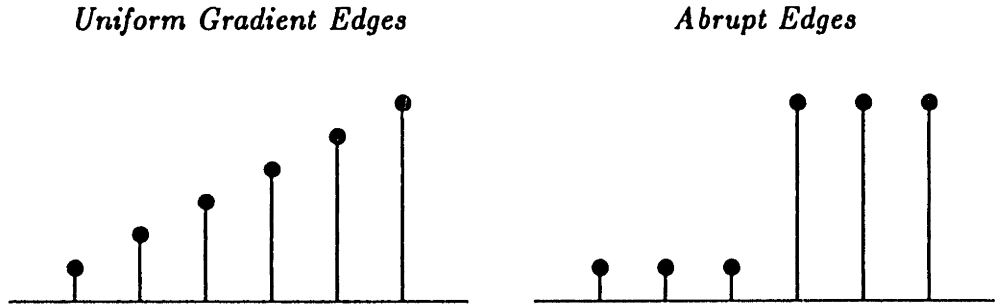


Figure 4.2: Edge cross sections

error as a function of signal-to-noise level.

The second test sequence consists of two frames with a set of abrupt edges. Figure 4.3 illustrates one frame of this sequence. This frame contains a set of edges where the intensity change is abrupt from one pel to another. Figure 4.2 illustrates the horizontal cross section of one of these edges. This sequence was selected for two reasons. First, it represents a very common feature that is present in real-life pictures. Second, these edges do not exactly fit the signal models used in the maximum likelihood and least squares motion estimators. Therefore this sequence illustrates the effect of modeling error in addition to the effects of additive noise.

In these experiments we do not perform velocity averaging because we are interested primarily in comparing the relative errors of the basic estimators. For these sequences we are interested in the RMS estimation error only for the component of the velocity which is orthogonal to the edges. The estimators actually determined both components of the velocity field, but since the edges are uniform in the vertical direction, the component of the velocity in this direction is not constrained by the signal. Therefore the RMS estimation error was defined as

$$RMS\ Error = \sqrt{(v_x - \hat{v}_x)^2} \tag{4.1}$$

where the x direction is orthogonal to the edges.

It is straightforward to compute the Cramer Rao bounds for the uniform gradient

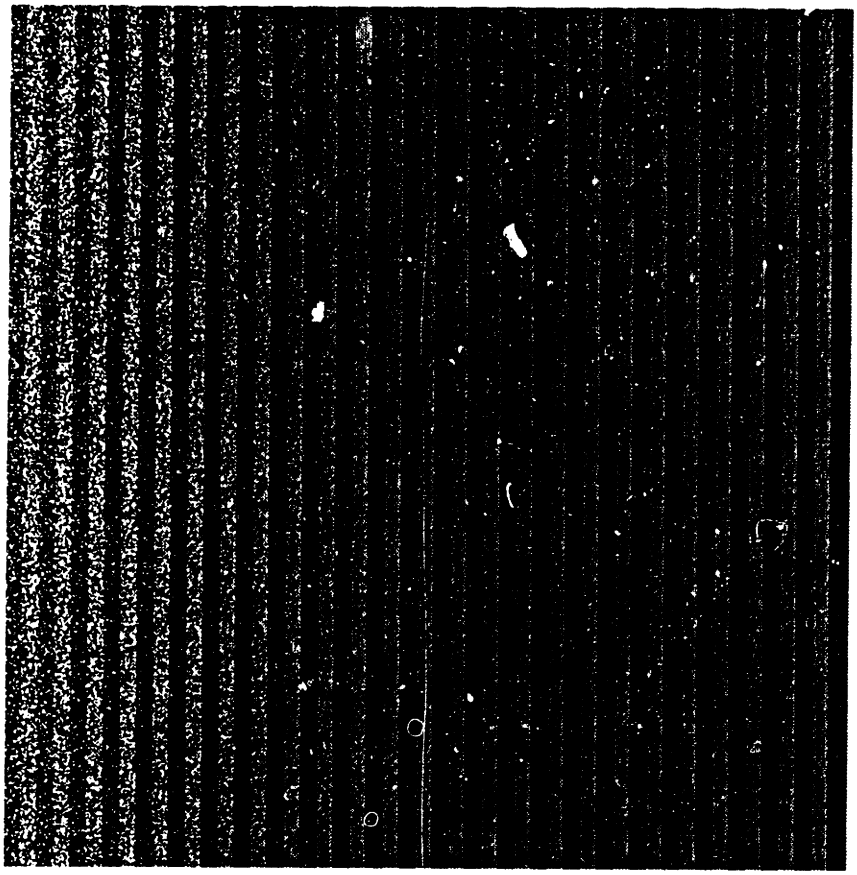


Figure 4.3: Abrupt edges

edge sequence. Referring to Equation (C.22) in Appendix C, we have the result

$$Var[\hat{v}_x - v_x] \geq \frac{\sigma_n^2}{\sum_{i=1}^{N} \left(\frac{\partial s_i}{\partial x}\right)^2 (t_i - t_0)^2}. \tag{4.2}$$

The following conditions hold for this test sequence:

- Since all the observation samples lie in a region where the gradient is constant,

$$\left(\frac{\partial s_i}{\partial x}\right)^2 = G_x^2. \tag{4.3}$$

- Two frames are used in forming the estimates. For all samples, $|t_i - t_0| = 1$.

Therefore the Cramer Rao bounds reduce to

$$Var[\hat{v}_x - v_x] \geq \frac{\sigma_n^2}{NG_x^2}. \tag{4.4}$$

For a 5 x 5 analysis window, $N = 50$ because there are 25 observation points in each frame. Based on this bound, we define the signal-to-noise level (SNR) as

$$SNR = 10 \log_{10} \left(\frac{G_x^2}{\sigma_n^2} \right) \tag{4.5}$$

Computing the Cramer Rao bound for the abrupt edge sequence is slightly more co.:plicated. The problem is that there is not a unique definition for the gradient of the picture at the observation samples which lie on both sides of the intensity discontinuity. However, referring to Figure 4.2, we can define a signal which has a piecewise continuous first derivative, by linear interpolation between the given sample values. The gradient at each sample point is taken to be the average of the gradient values on both sides of the sample point. Therefore, at the points which straddle a discontinuity, the gradient is equal to $\delta I/2$, where the intensity change step size is δt. Therefore the Cramer Rao bound for samples taken from a M x M window in two frames is

$$Var[\hat{v}_x - v_x] \geq \frac{2\sigma_n^2}{M(\delta I)^2}. \tag{4.6}$$

In these experiments a random initial velocity is used as the starting point (the true velocity field is zero everyw here). The initial estimate is a uniformly distributed random variable in the range $(-1.5, 1.5)$. This range was selected because it is compatible with the size of the analysis window that was used in forming the estimates. For each signal-to-noise level, 400 estimates were obtained with each algorithm and the resulting experimental error is the average over these estimates. Therefore the experimental RMS error was computed as follows

$$RMS\ Error = \left(\frac{1}{400} \sum_{i=1}^{400} \left(\hat{v}_{x(i)} \right)^2 \right)^{\frac{1}{2}}. \tag{4.7}$$

The range of SNR values which we made measurements of estimation error was selected in the following manner. We are primarily interested in the range of SNR values where the estimation error in the displacement field is less than the spatial sampling interval (subpel accuracy). There are two reasons for this. Firstly, if the pictures are to be filtered along the estim .ted motion trajectories, displacement field estimation errors on the order of a pel introduce spatial blur that is comparable

to spatial filters. If this occurs then temporal filters do not offer any advantage over spatial filters. Secondly, the multigrid algorithm for estimating large velocities requires subpel accuracy for proper operation. Therefore a range of SNR values was selected so that the dynamic range of the displacement field estimation errors approximately spanned one pel.

4.1.1 Uniform gradient edges

Figure 4.4 presents the results for the uniform gradient edge test sequence.

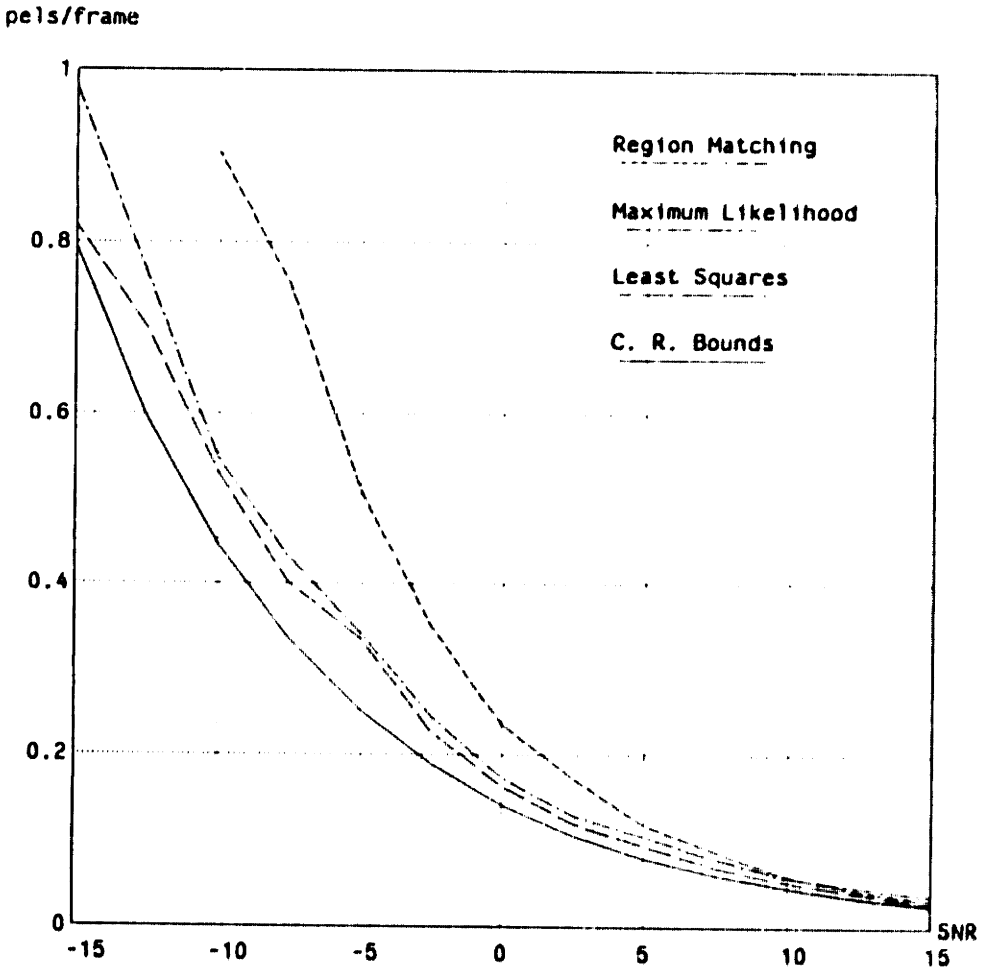


Figure 4.4: Motion estimation error: Uniform gradient edges

Several observations should be noted in these plots:

- The motion estimation error is always greater than the Cramer Rao bound.

- The least squares and maximum likelihood estimators yield almost identical motion estimation errors, which are uniformly smaller than the region matching method (for very high SNR values the region matching method yields a slightly smaller estimation error than the maximum likelihood method).

We can gain more insight into the properties of these estimators by examining the error histograms shown in Figure 4.5. These histograms correspond to a SNR

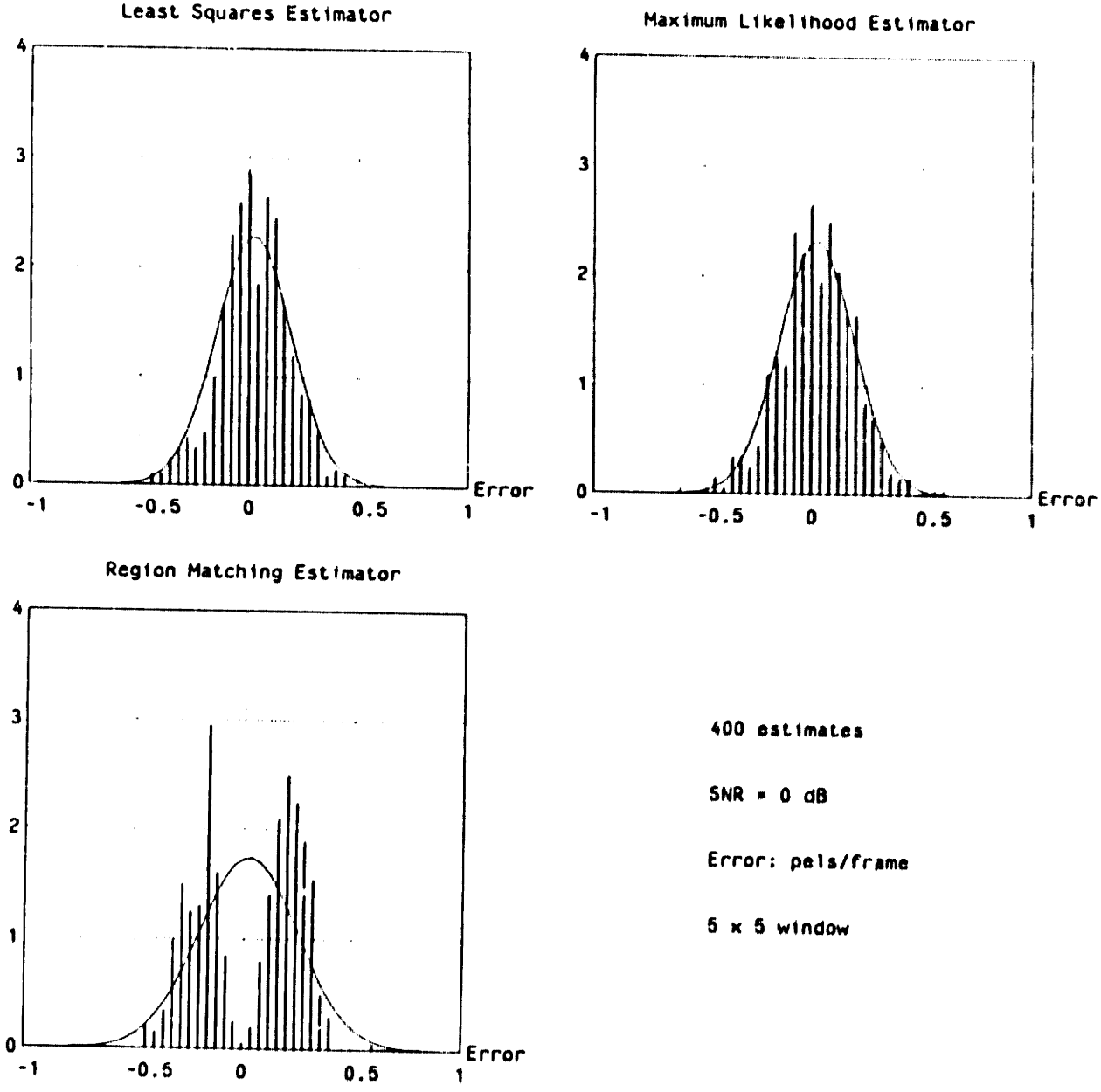


Figure 4.5: Error histograms: Uniform gradient edges

of 0 dB, and are generated from the same estimates used to compute the RMS error at this SNR. For each histogram we also plot a Gaussian distribution that has the same variance as the experimental measurements. We can note several features in these histograms:

- The estimation errors for the least squares and maximum likelihood methods are essentially Gaussian distributed.
- The estimation errors for the region matching method tend to cluster at two values on both sides of the origin.

The clustering phenomena of the region matching estimator is an artifact of the bilinear interpolator used to compute signal values which are not on the sampling grid. If a signal value is desired at a point on the sampling grid, the resulting value is the true signal value plus a random noise quantity which has the same noise variance the noise field. On the other hand, if we desire a signal value which is not on the sampling grid, the bilinear interpolator computes an interpolation value with an additive noise term which has a smaller variance than the noise field (because of averaging). This results in an objective function that is smaller for velocity values which induce a displacement field that is not on a sampling grid point. We can illustrate this by plotting the objective function for the region matching estimator for a direction which is orthogonal to the edge. Figure 4.6 contains the objective functions at a particular location in the picture, for the region matching and maximum likelihood algorithms. These objective functions are typical of those generated by the estimators.

Note that the objective function for the maximum likelihood estimator is well behaved, containing only a single stationary point, while the region matching estimator objective function contains several local minima. The descent algorithm will converge to the global minimum in the maximum likelihood estimator, but will converge to a local minimum in the region matching estimator.

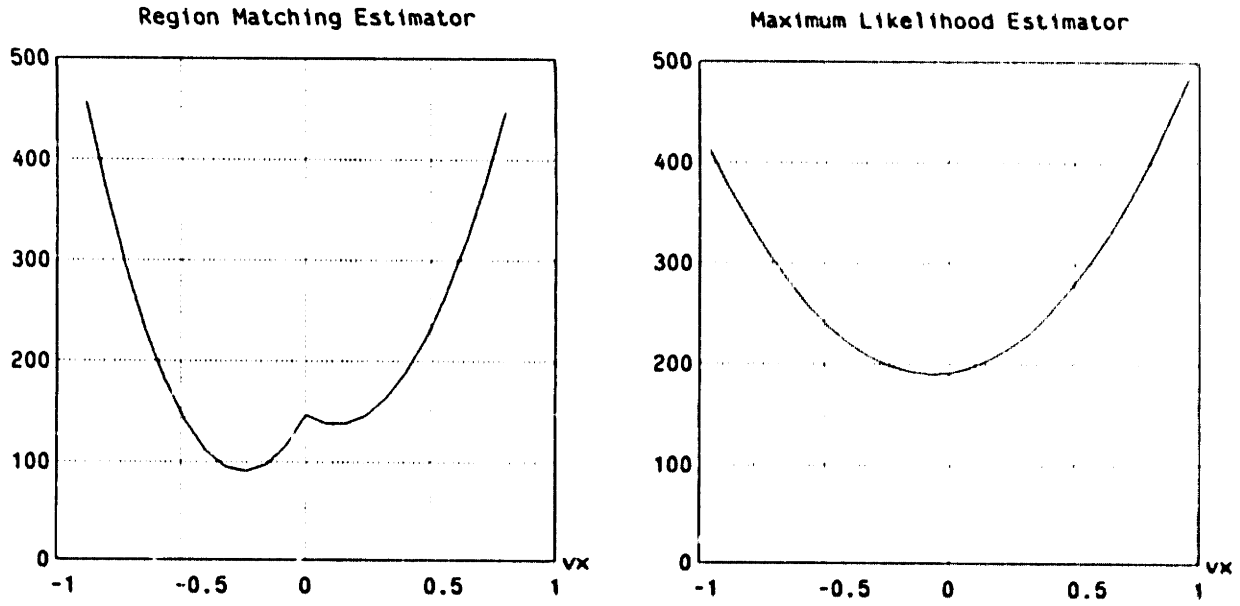


Figure 4.6: Typical objective functions: Uniform gradient edges

4.1.2 Abrupt edges

Figure 4.7 presents the results for the abrupt edge test sequence. Several features should be noted from these plots:

- The least squares and maximum likelihood estimators yield almost identical estimation errors.
- The region matching estimator yields considerably larger estimation errors for low and moderate SNR values, and smaller estimation errors for very high SNR values.
- The error associated with the least squares method approaches an asymptotic value of 0.173 as the SNR tends to infinity. This is due to modeling error.
- The error associated with the maximum likelihood method approaches an asymptotic value of 0.169 as the SNR tends to infinity, and is also due to modeling error.

It is straightforward to compute the asymptotic values associated with the least squares and maximum likelihood estimators. The initial velocity estimate for v_x is

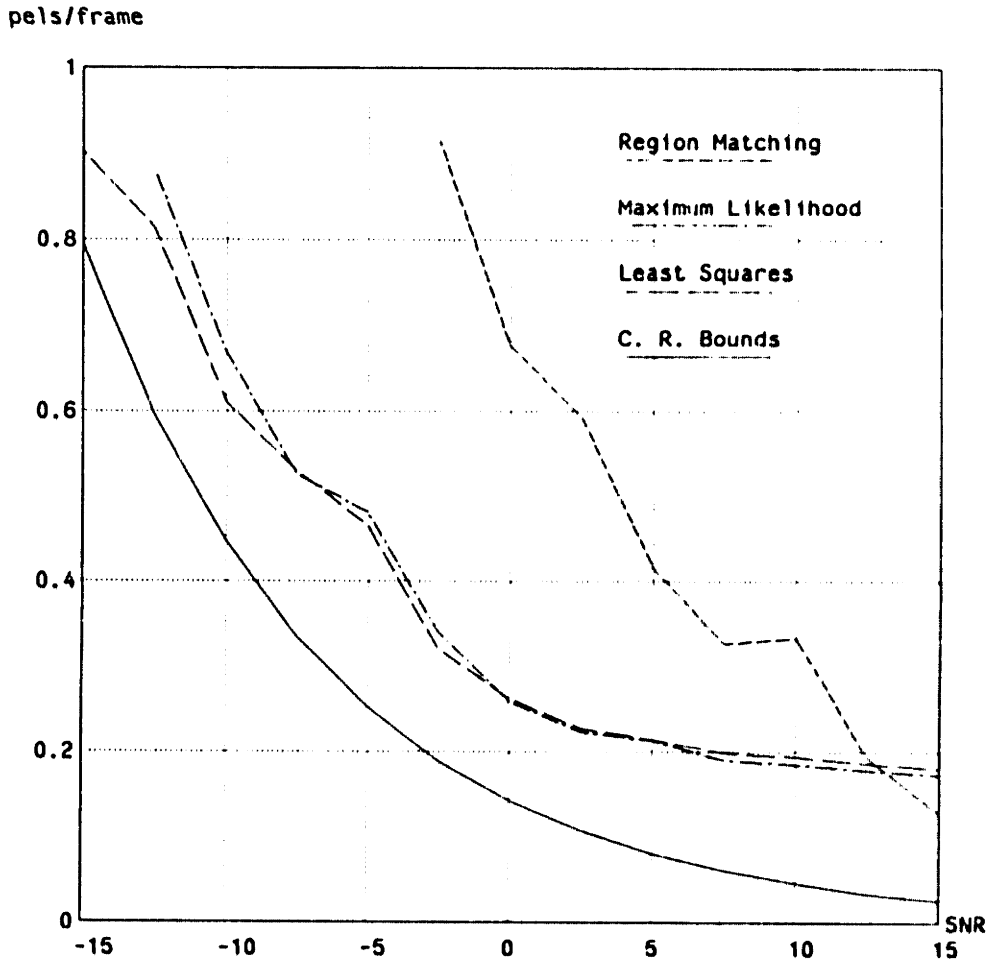


Figure 4.7: Motion estimation error: Abrupt edges

a uniform random variable in the range $(-1.5, 1.5)$. The analysis window selection process partitions this range into three regions illustrated in Figure 4.8. The velocity estimate which is generated by the estimators for each region is shown in the following table.

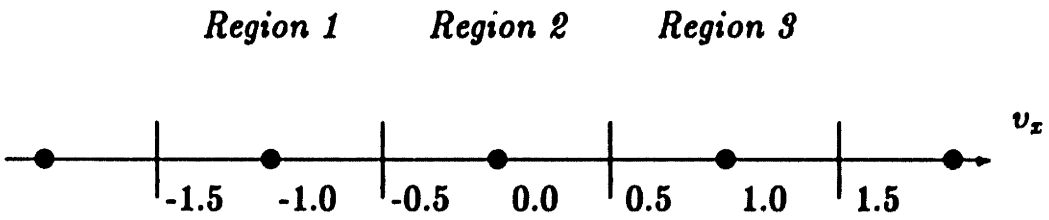


Figure 4.8: Analysis region partitions

Estimation asymptotic limits			
	Region 1	Region 2	Region 3
Least squares	-0.212	0	0.212
Maximum likelihood	-0.208	0	0.208

The probability that the initial estimate is in either of the regions is equally likely. Therefore the standard deviation for the least squares estimate is $\sqrt{\frac{2}{3}}$ x 0.212 = 0.173, and the standard deviation for the maximum likelihood estimate is $\sqrt{\frac{2}{3}}$ x 0.208 = 0.169. When there is noise, these asymptotes become the expected value of the estimate in each region and there are statistical deviations about the means.

Figure 4.9 contains the error histograms and Figure 4.10 presents typical objective functions for this test sequence. In these plots we can see that the region matching estimator exhibits the same problems with this test sequence as in the uniform gradient edge test sequence.

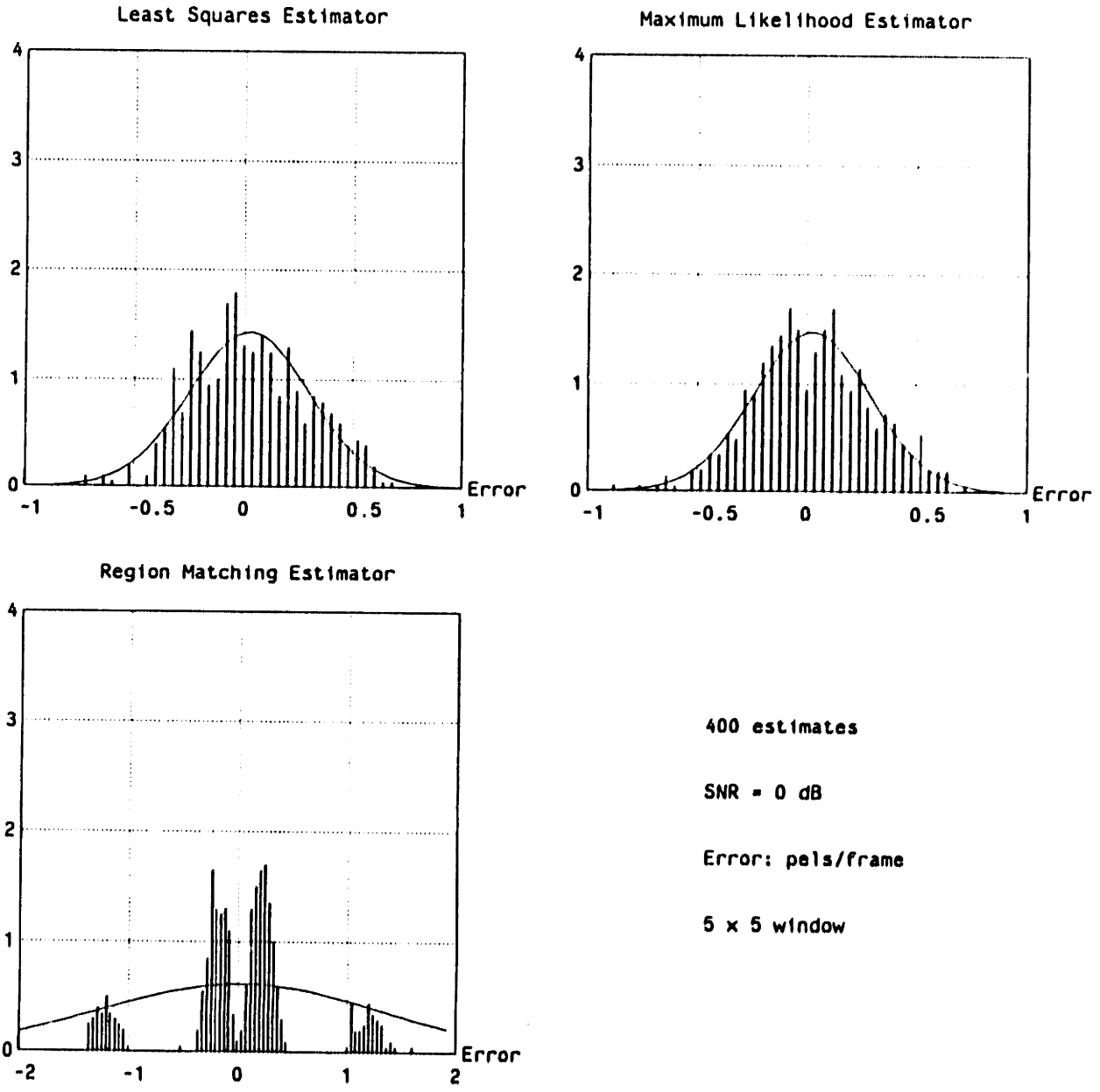


Figure 4.9: Motion estimation error histograms: Abrupt edges

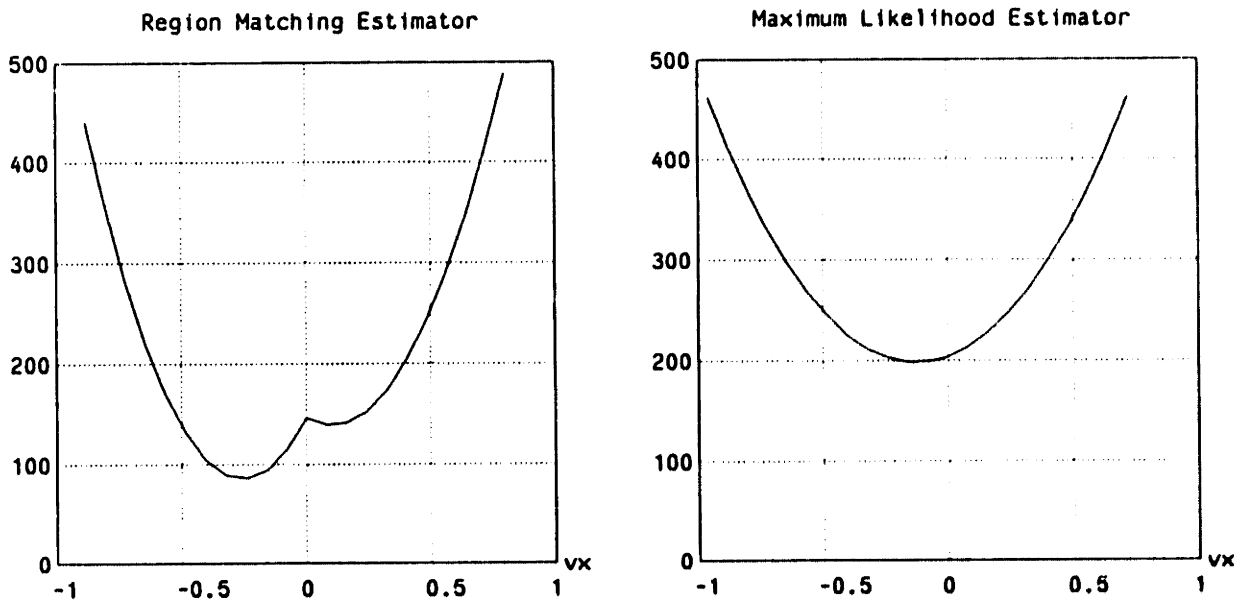


Figure 4.10: Typical objective functions: Abrupt edges

4.1.3 Prefiltered edges

It is possible to reduce the estimation error by applying a spatial filter to the images prior to motion estimation. To demonstrate this we filtered the abrupt edge test sequence and repeated the same experiment. The spatial filter was an unweighted average of the samples on a 3 x 3 grid. The Cramer Rao bounds have been modified to account for the fact that the images were prefiltered. With this filter the effective analysis window size increases from 5 x 5 to 7 x 7. Figure 4.11 presents the results for the filtered edge test sequence and Figure 4.12 presents typical error histograms.

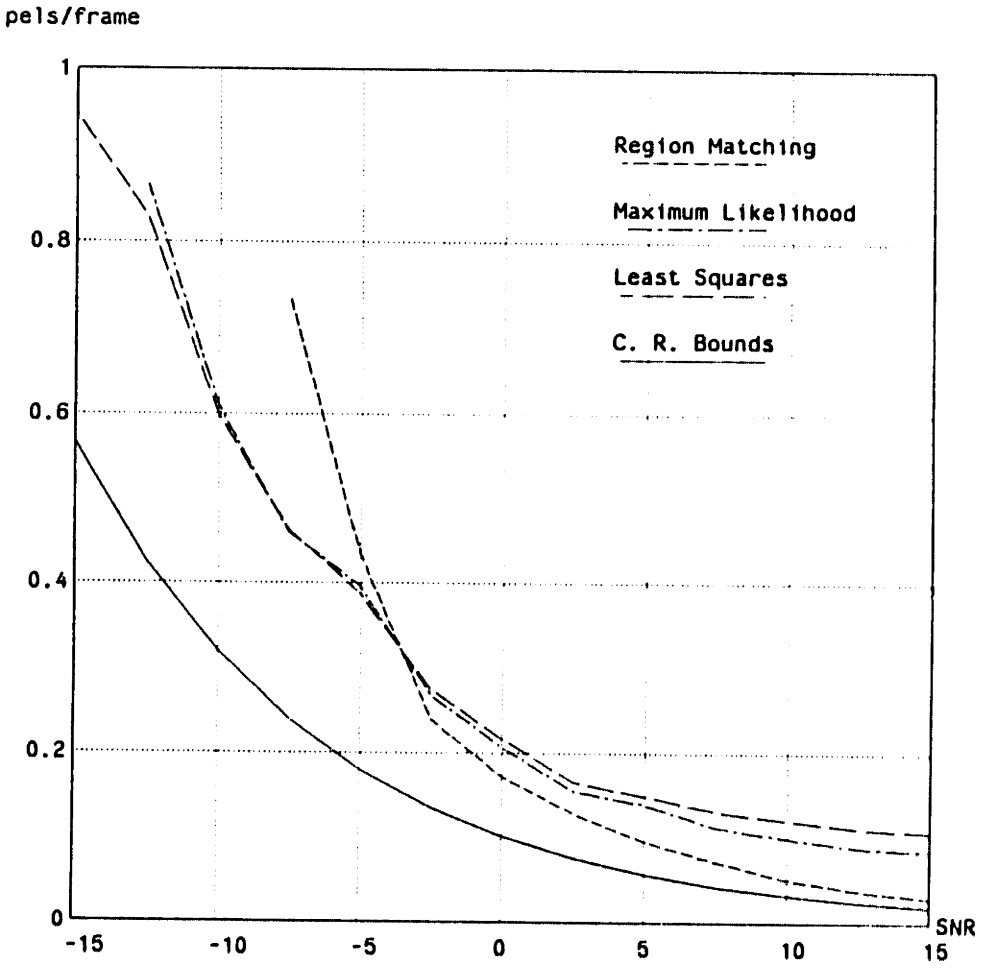


Figure 4.11: Motion estimation error: Filtered edges

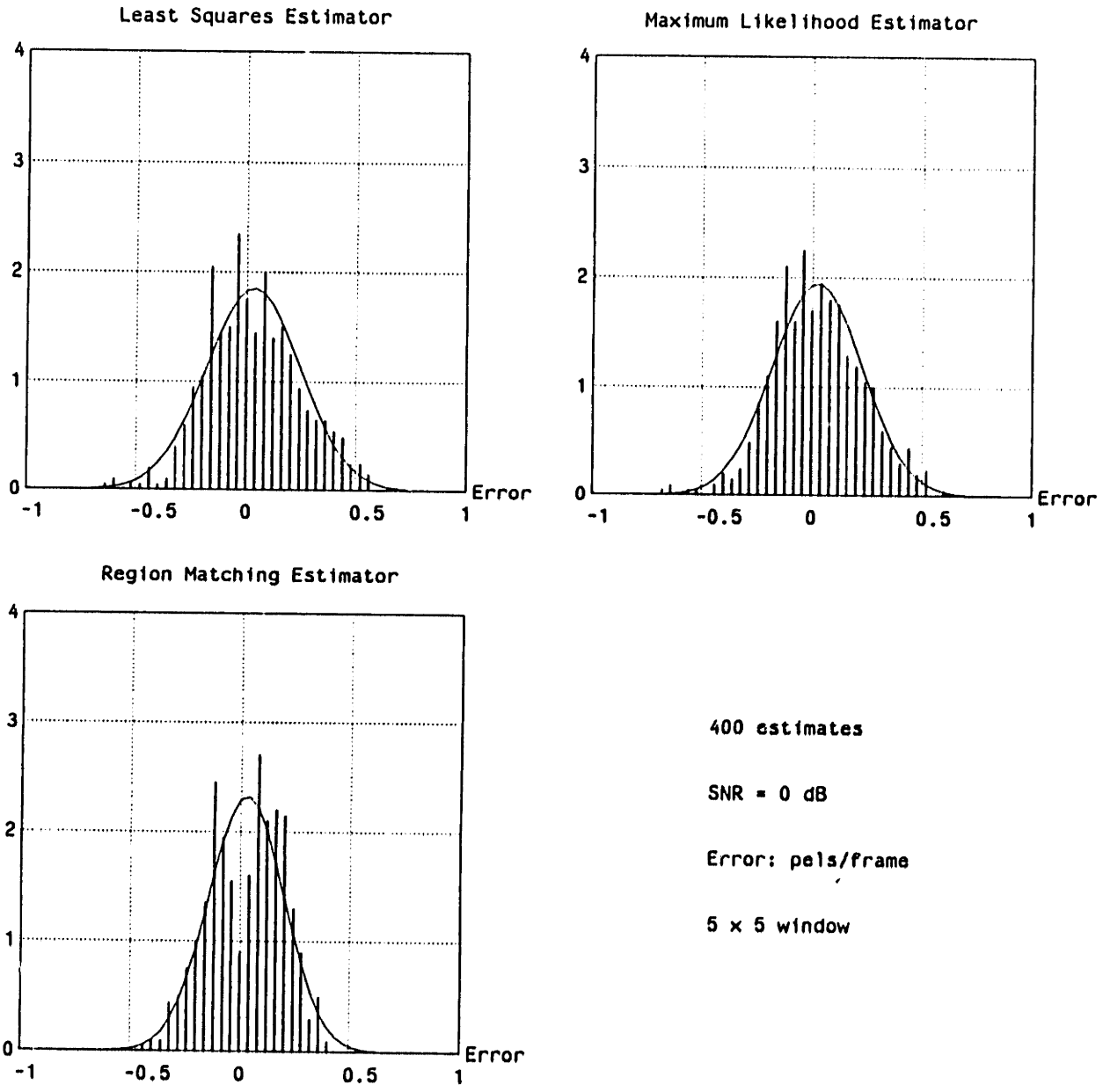


Figure 4.12: Motion estimation error histograms: Filtered edges

In these figures we can note several features:

- The estimation error for the least squares and maximum likelihood estimators only decreases slightly.
- There is a large improvement in the estimation error for the region matching estimator.
- The asymptotic values for the least squares and maximum likelihood estimators are 0.098 and 0.076 pels/frame respectively.

The results of this experiment are not surprising. Both the least squares and maximum likelihood algorithms implicitly smooth the data by the process of signal estimation. Therefore we expect only slight improvement in performance by spatial-prefiltering. Conversely, we expect that prefiltering should make a significant improvement in estimation accuracy for the region matching algorithm because the effective signal-to-noise ratio is increased.

4.1.4 Discussion of empirical measurements

On the basis of these experiments several observations should be made.

- The estimation error obtained with the least squares and maximum likelihood algorithms are almost identical. Strictly on the basis of error performance, these two algorithms are basically equivalent. Because they implicitly smooth the pictures by virtue of the signal models, very little improvement results by prefiltering the images.

- The least squares and maximum likelihood algorithms are more accurate than the region matching algorithm if no prefiltering is performed (perhaps except at very high signal-to-noise levels).

- The performance of the region matching algorithm is significantly improved by prefiltering the frames prior to motion estimation. If prefiltering is performed, the region matching algorithm is more accurate than the least squares and maximum likelihood algorithms at moderate to high signal-to-noise levels. This fact is perhaps of little significance because at these signal-to-noise levels the error is on the order of 0.2 pels/frame.

Therefore on the basis of these experiments the least squares and maximum likelihood algorithms are judged to be superior to the region matching algorithm.

4.2 Subjective evaluation

In this section we present some results obtained by motion-compensated frame averaging. The purpose is to provide a subjective evaluation of the motion estimation algorithms. Figure 4.13 illustrates the experimental system that was used in these experiments. Two frames $r(\bar{x}, t_0)$ and $r(\bar{x}, t_1)$ are obtained by taking a still

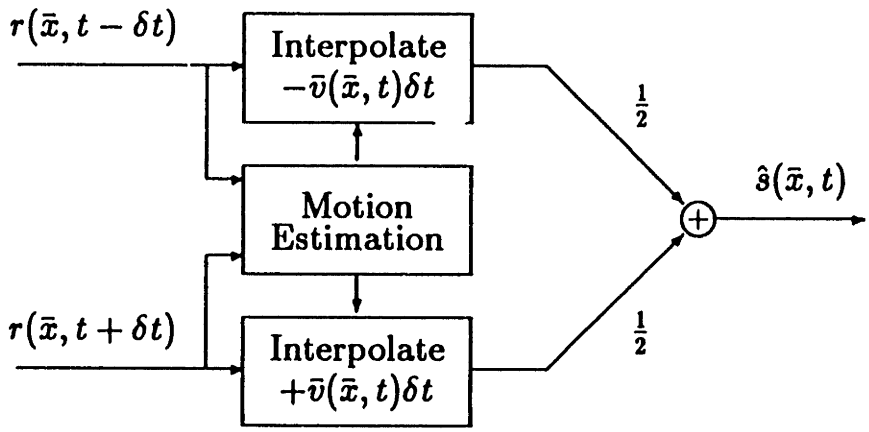


Figure 4.13: Motion-compensated temporal averaging

frame and adding two different noise fields to it. Therefore the true velocity field is zero everywhere. A random initial velocity estimate is used as the starting point for the estimators at each point in the picture. Experiments were conducted at both moderate and very low signal-to-noise levels. The moderate signal-to-noise level pictures contained additive noise with a standard deviation of 10 and the low signal-to-noise level pictures contained additive noise with a standard deviation of 20. Figures 4.14 and 4.15 contain the original and degraded frames. In addition these figures illustrate the effect of averaging with the random initial velocity field and the exact velocity field.

The pictures were processed with velocity estimates obtained in the following manners:

- Direct velocity estimates: The velocity estimates computed by the algorithms

Original

Degraded

Exact velocity

Random initial velocity

Figure 4.14: Test images ($\sigma_n = 10$)

Original

Degraded

Exact velocity

Random initial velocity

Figure 4.15: Test images ($\sigma_n = 20$)

are used directly without any modifications.

- Spatial prefiltering: The frames were filtered prior to motion estimation.
- Velocity averaging: The velocity estimates were averaged. Each estimate was averaged with the nearest 8 estimates on a 3 x 3 grid (unweighted averaging). In this experiment spatial prefiltering is not performed.

Figures 4.16, 4.17, and 4.18 present the results for the pictures with noise standard deviation equal to 10 and Figures 4.19, 4.20, and 4.21 present the results for the pictures with noise standard deviation equal to 20.

From these pictures we can make several observations:

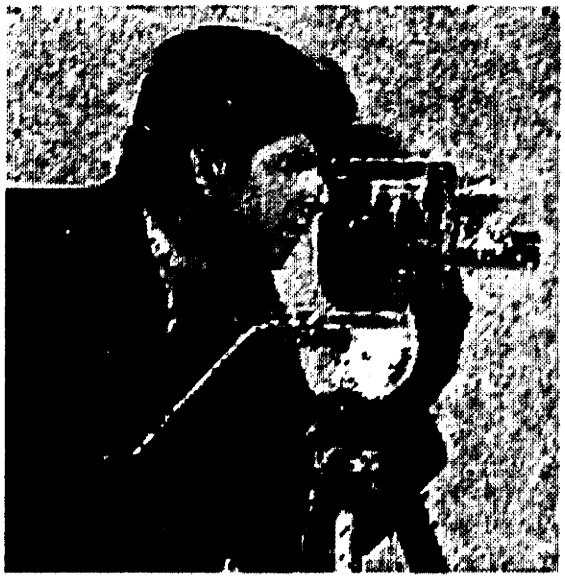
- If the estimates are used directly, artifacts are introduced into the picture with all three algorithms. The least squares algorithm introduces the fewest artifacts and the region matching algorithm introduces the most artifacts.
- If the pictures are spatially filtered prior to motion estimation, there is slight improvement in the least squares and maximum likelihood examples and there is significant improvement in the region matching example. The pictures processed with the least squares algorithm contain only minimal artifacts and the artifacts introduced with the maximum likelihood and region matching methods are comparable.
- If the velocity estimates are averaged, there are essentially no visible artifacts with the least squares and maximum likelihood algorithms, but some visible artifacts remain with the region matching method.

These experiments agree well with the results of the empirically determined motion estimation error curves. In addition these experiments illustrate that for the least squares and maximum likelihood algorithms it is better to increase the effective window size by velocity averaging than by spatial prefiltering in order to improve the motion estimation error. In part this indicates that although the analysis windows overlap considerably, the errors tend to remain highly uncorrelated. Consequently, averaging yields a significant reduction in estimation error.

In experiments with a variety of other test images it was determined that the weighted averaging strategy described in Chapter 3 yields slightly better results than unweighted averaging (this only applies to the least squares algorithm). Based on the results described in this section and the computational requirements of these algorithms, the least squares algorithm with weighted averaging is used exclusively in the remaining experiments.

Degraded

Region matching



Maximum likelihood

Least squares

Figure 4.16: Direct velocity estimates ($\sigma_n = 10$)

Degraded

Region matching

Maximum likelihood

Least squares

Figure 4.17: Spatial prefiltering ($\sigma_n = 10$)

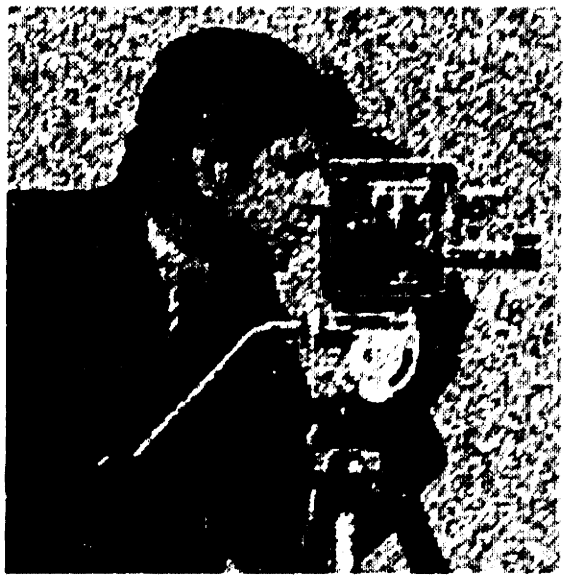
Degraded

Region matching

Maximum likelihood

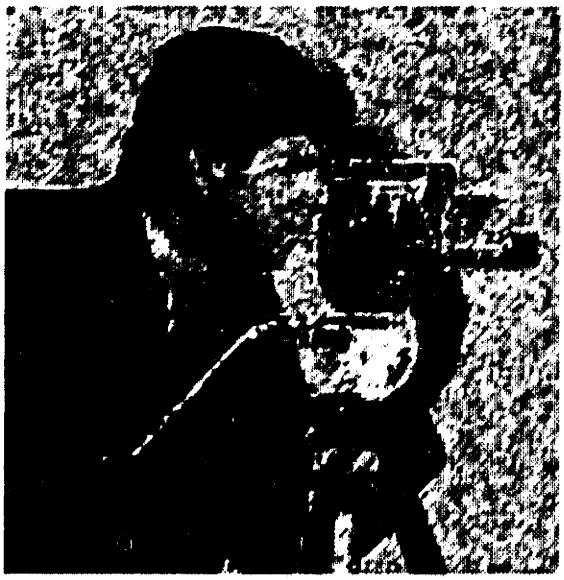
Least squares

Figure 4.18: Velocity averaging ($\sigma_n = 10$)



Degraded

Region matching



Maximum likelihood

Least squares

Figure 4.19: Direct velocity estimates ($\sigma_n = 20$)

Degraded

Region matching

Maximum likelihood

Least squares

Figure 4.20: Spatial prefiltering ($\sigma_n = 20$)

Degraded

Region matching

Maximum likelihood

Least squares

Figure 4.21: Velocity averaging ($\sigma_n = 20$)

4.3 Multigrid experimental results

In this subsection we present some results obtained with the multigrid algorithm. The test sequences are the same as those used in the previous section, except that the frames were displaced by different amounts to generate large velocity fields. Specifically, we generated velocity fields of 1, 2, 4, and 6 pels/frame. In all these experiments we used the least squares algorithm with weighted averaging.

Figures 4.22 and 4.23 present the results for the sequence where the noise standard deviation was 10. Figures 4.24 and 4.25 present the results for the sequences where the noise standard deviation was 20. In these pictures we show the frames processed both with and without the multigrid algorithm. The multigrid algorithm used a three level grid (4, 2, then 1).

These pictures illustrate the necessity of a strategy for dealing with large velocity fields. They also illustrate that the multigrid algorithm is effective at dealing with large velocities. More examples are included in the next chapter where actual motion pictures were processed with this technique.

$v = 1$, no multigrid

$v = 1$, multigrid

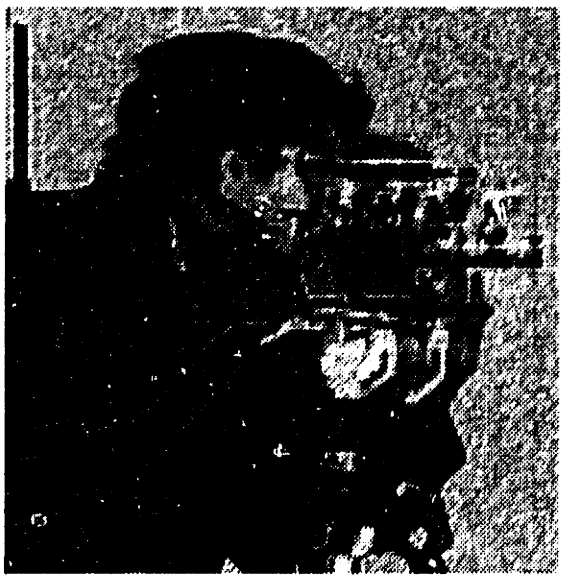
$v = 2$, no multigrid

$v = 2$, multigrid

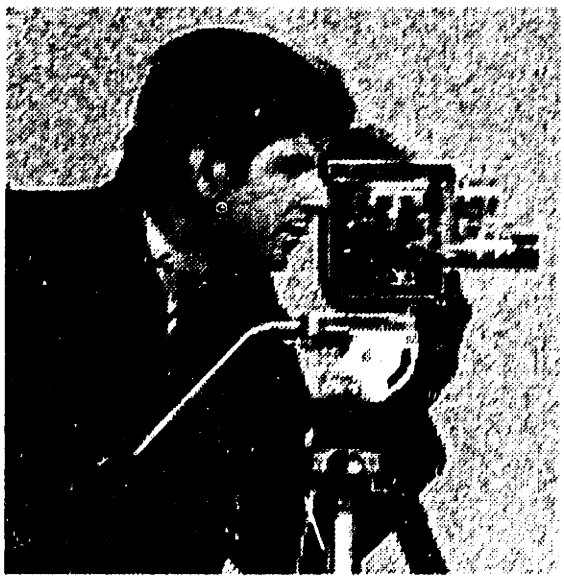
Figure 4.22: Multigrid results ($v = 1,2$ $\sigma_n = 10$)

$v = 4$, no multigrid

$v = 4$, multigrid



$v = 6$, no multigrid



$v = 6$, multigrid

Figure 4.23: Multigrid results ($v = 4{,}6$ $\sigma_n = 10$)

$v = 1$, no multigrid

$v = 1$, multigrid

$v = 2$, no multigrid

$v = 2$, multigrid

Figure 4.24: Multigrid results ($v = 1,2$ $\sigma_n = 20$)

$v = 4$, no multigrid

$v = 4$, multigrid

$v = 6$, no multigrid

$v = 6$, multigrid

Figure 4.25: Multigrid results ($v = 4{,}6$ $\sigma_n = 20$)

Chapter 5

Motion picture restoration

In the previous chapters we focused on the problem of motion estimation from noisy samples of an image sequence. The basic problem was to extract a velocity field which describes the motion of objects in an image sequence. This effort culminated in the formulation of an algorithm which can estimate both large and small velocities very accurately. In this chapter we apply this algorithm to the problem of motion picture restoration. The canonical system for representing the restoration process is shown in Figure 5.1. Several degradations that occur in practice which we consider in this work include additive noise and impulsive noise.

A wide variety of algorithms for image restoration have been proposed in the past. There are numerous contexts in which this problem has been phrased and studied. Our attention is restricted to the specific case of noise removal. A more general formulation includes deconvolution methods for removing blur in addition to noise. There are three distinct methodologies into which noise reduction systems for motion pictures can be classified

- single frame restoration
- multiple frame restoration (without motion compensation)
- motion-compensated restoration

In the following subsections we briefly summarize these methods and describe an implementation of each method which we will compare with the other methods.

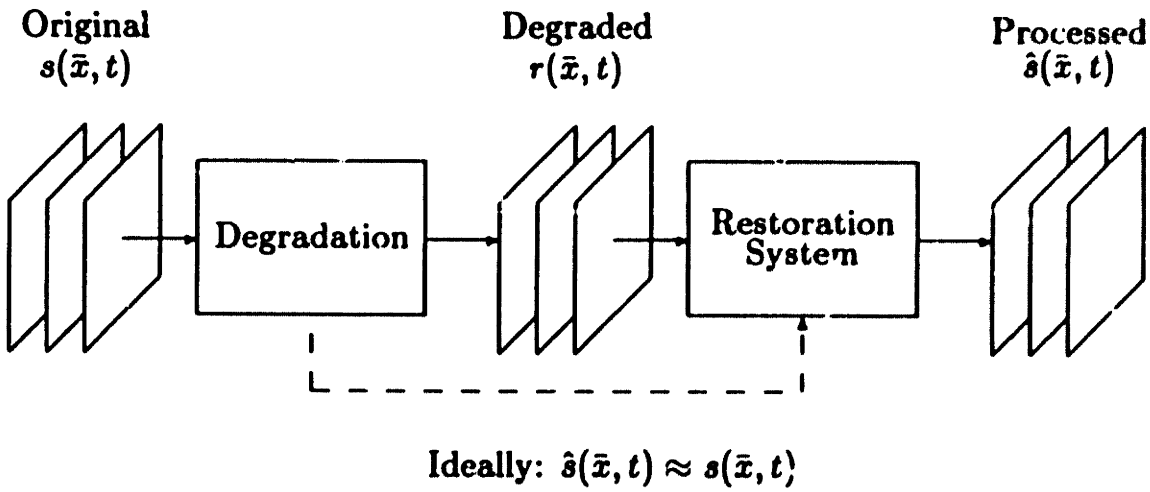


Figure 5.1: Canonical restoration system

5.1 Single frame restoration systems

Many methods for restoring single frame images have been proposed in the literature. This field of study is fairly well advanced and comprehensive surveys of these methods are given in [2,9,10,30]. The most widely used restoration model includes a point spread function (PSF) which is spatially invariant and observations which have been corrupted with additive noise. Given a signal $s(x,y)$, PSF $h(x,y)$, and noise field $n(x,y)$, the observation $r(x,y)$ is given by

$$r(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(u,v)s(x-u,y-v)du\,dv + n(x,y). \tag{5.1}$$

The restoration problem is to estimate $s(x,y)$ from the observations $r(x,y)$. Several methods which have been applied to this problem include

- inverse filtering
- Wiener filtering
- homomorphic restoration
- iterative restoration methods (with and without constraints).

So far as subjective tests are concerned, these methods have only achieved limited success. Much of the difficulty is that they minimize a global function of the error which often does not reflect important perceptual characteristics of the human visual system. For example, the method of Wiener filtering results in pictures which are severely blurred, although in a mean-squared sense the method is "optimal".

A number of alternate methods which address this issue have been proposed in the literature [1,17,5]. The general approach involves the use of adaptive filters. Many of these methods can be generalized in the following manner. If a signal estimate at a point (x_0, y_0) is desired, an adaptive filter is applied to the signal using observations in the neighborhood of (x_0, y_0). The parameters of the filter are adapted according to the local image characteristics. This procedure is illustrated in Figure 5.2. For example, the method of Anderson and Netravali [1] uses a subjective criterion to adapt the parameters of an FIR filter and leads to good tradeoffs between blur introduced by the filter and noise removal.

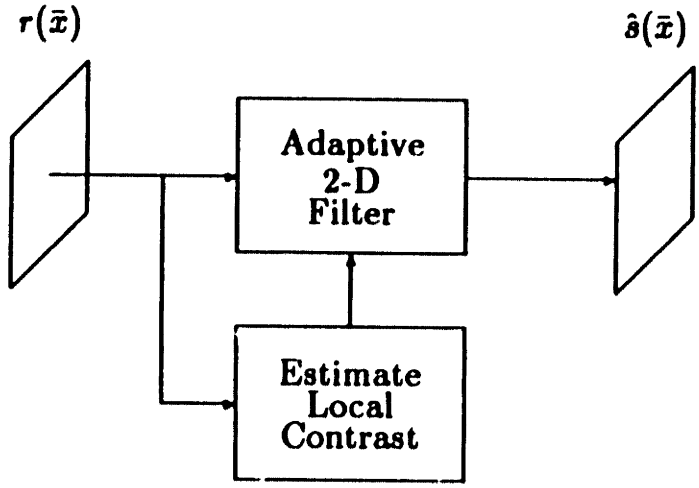


Figure 5.2: Adaptive image restoration

For comparison purposes we have implemented an algorithm proposed by Chan and Lim [5]. It involves filtering each frame with a set of adaptive one-dimensional filters oriented along the major correlation directions of the image (0, 45, 90, and 135 degrees). The adaptive filters have the same structure as the algorithm described in the next section.

5.2 Multiple frame restoration systems

For motion picture noise reduction one can use the same strategies as in the single frame restoration systems. Adaptive filters can account for the presence of motion. These methods are attractive because they do not require motion estimation. Rather than explicitly trying to determine motion trajectories along which temporal filters are applied, these algorithms combine both operations into a single estimator/filter structure. Martinez and Lim [21] proposed an algorithm which is an extension of a method developed by Chan and Lim [5] for processing single frames. Samy [31] proposed several algorithms which are similar to the algorithm of Martinez and Lim.

The algorithm proposed by Martinez and Lim assumes that the signal has five primary correlation directions corresponding to: (1) no motion, (2) translation in the +x direction, (3) translation in the -x direction, (4) translation in the +y direction, and (5) translation in the -y direction.

An adaptive one-dimensional filter is applied to the three-dimensional sequence along these 5 directions, producing 4 intermediate frame sequences and the final output sequence. This is illustrated in Figure 5.3. The structure of the filters was

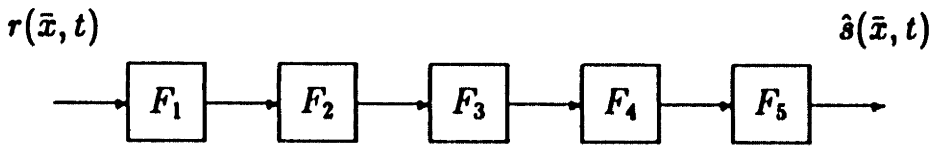


Figure 5.3: Multidirectional adaptive noise reduction system

chosen to satisfy the following heuristics:

- If the image is highly correlated along the direction, then the filter should have a low cutoff frequency. Conversely if the image is not highly correlated, then the filter should have no effect.

- In order to avoid introducing artifacts and other further deteriorations, the

statistical mean of the output sequence was made equal to the statistical mean of the input sequence.

A filter structure which possesses the desired characteristics is the linear least squares point estimator which has the following form [36]

$$\hat{s}(\bar{x},t) = \frac{\sigma_s^2(\bar{x},t)}{\sigma_s^2(\bar{x},t)+\sigma_n^2}[r(\bar{x},t) - m_s(\bar{x},t)] + m_s(\bar{x},t). \tag{5.2}$$

In this expression, $m_s(\bar{x},t)$ and $\sigma_s^2(\bar{x},t)$ are the estimated mean and variance of the desired sequence $s(\bar{x},t)$, and σ_n^2 is the variance of the noise field. For a particular direction and spatio-temporal position $(\bar{x}_0,t_0)$, $\sigma_s^2(\bar{x}_0,t_0)$ and $m_s(\bar{x}_0,t_0)$ are evaluated according to the following set of equations

$$m_s(\bar{x}_0,t_0) = \frac{1}{N}\sum_{i=1}^{N} s(\bar{x}_i,t_i) \tag{5.3}$$

$$\sigma_r^2(\bar{x}_0,t_0) = \frac{1}{N}\sum_{i=1}^{N}[s(\bar{x}_i,t_i) - m_s(\bar{x}_0,t_0)]^2 \tag{5.4}$$

$$\sigma_s^2(\bar{x}_0,t_0) = \begin{cases} 0 & \text{if } \sigma_n^2 > \sigma_r^2(\bar{x}_0,t_0) \\ \sigma_r^2(\bar{x}_0,t_0) - \sigma_n^2 & \text{otherwise} \end{cases} \tag{5.5}$$

The set of points $\{(\bar{x}_i,t_i)\}$ are taken from samples along one of the five correlation directions and are centered about the point $(\bar{x}_0,t_0)$.

Several important properties of this filter structure should be noted. When the signal variance estimate $\sigma_s^2(\bar{x},t)$, is much larger than the noise variance σ_n^2 (high SNR), then the output reduces to $\hat{s}(\bar{x},t) = r(\bar{x},t)$, and the filter does nothing to the sequence. When the signal variance estimate is much smaller than the noise variance, the output reduces to $\hat{s}(\bar{x},t) = m_s(\bar{x},t)$. This corresponds to the maximum amount of noise reduction possible with a 1-D FIR filter structure. The first case corresponds to the situation where the contour is not oriented along a motion trajectory, while the latter case corresponds to the situation where the contour coincides with a motion trajectory.

5.3 Motion-compensated restoration systems

One of the problems with the systems described in the preceding sections is that they involve spatial filters in one form or another. These filters have the property that typically noise can only be removed at the expense of picture sharpness. By way of contrast, motion-compensated systems can many times operate without any loss in picture sharpness.

A comparative study of two methods for motion-compensated noise reduction was conducted by Huang and Hsu [14] in two experiments. In the first experiment an FIR temporal filter was applied along a suitably chosen direction in the three-dimensional signal space. The direction was chosen by searching over a small number of directions for the one with the smallest variance. This motion estimator is essentially an M-ary detector. In the second experiment an explicit motion trajectory was estimated. The motion estimation algorithm was based on the method of spatio-temporal constraints. The estimation procedure involved solving a set of overdetermined linear equations for a motion estimate. An FIR filter was applied along the estimated motion trajectory. Significant improvements in subjective image quality were reported in both experiments.

Several other methods for motion-compensated noise reduction have been proposed by other researchers. McMann et al. [22] and Dennis [7] developed motion-compensated noise reduction systems incorporating IIR filter structures. The filters are only applied along the temporal direction, but are adapted according to a motion detector. The motion detector is essentially a first-order linear predictor. If the prediction error is small, it is assumed that there is no motion, and first-order recursive filtering is performed. If the prediction error is large, it is assumed that there is motion in the vicinity, and no filtering is performed.

Dubois and Sabri [8] have improved this method by performing explicit motion estimation. Their motion estimation algorithm is based on the method proposed by Netravali and Robbins [25]. In this system a motion trajectory is estimated and the signal is filtered along the trajectory with a first-order recursive filter. The

coefficient of the filter is adapted according to the first-order linear prediction error along the motion trajectory. The filter has a very low cutoff frequency if the error is small. As the prediction error increases, the filter tends towards an all-pass frequency response and the signal is not filtered.

5.4 Experiments in motion-compensated noise reduction

In this section we describe a noise reduction system which uses the multigrid/least squares motion estimation algorithm. The system we developed applies a one-dimensional directional filter to the image sequence at each point in the picture. The samples which make up the filter are obtained from a three point motion trajectory which is determined from the estimated velocity field at each point.

A three point filter was used in all the experiments which were conducted. Therefore the filter traversed three frames, centered on the frame for which output is desired (hereafter referred to as the current frame). The three points which are used to estimate the signal at each sample location in the current frame are obtained as follows. Let us denote the time instant corresponding to the current frame as t_0, so that $t_0 - \delta t$ and $t_0 + \delta t$ are the time instants corresponding to the past and future frame. Note that since we make use of the "future frame", there is a one frame delay in processing time.

Let $(\bar{x}_0)$ be the spatial position within the current frame where a signal estimate is desired. The velocity field that was computed between the current and past frame is evaluated at the spatio-temporal instant $(\bar{x}_0, t_0)$, and projected backwards in time to obtain a displacement field in the past frame. Similarly, the velocity field that was computed between the current and future frame is evaluated at the spatio-temporal instant $(\bar{x}_0, t_0)$, and projected forwards in time to obtain a displacement field in the future frame. This procedure is illustrated in Figure 5.4. Therefore the three samples of the signal which we require are

- $r_{past} = r(\bar{x}_0 - \bar{v}_{past}(\bar{x}_0, t_0), t_0 - \delta t)) \quad \Longrightarrow \quad$ past frame
- $r_{current} = r(\bar{x}_0, t_0) \quad \Longrightarrow \quad$ current frame
- $r_{future} = r(\bar{x}_0 + \bar{v}_{future}(\bar{x}_0, t_0), t_0 + \delta t)) \quad \Longrightarrow \quad$ future frame

The sample in the current frame is on the sampling grid. However, in general the samples in the past and future frames are not. Therefore it is necessary to compute

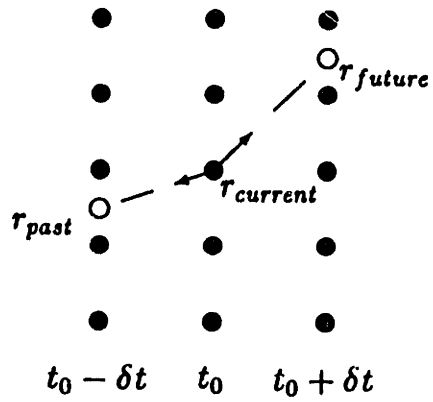


Figure 5.4: Three point sample along motion trajectory

these points with a spatial interpolator. Experiments were conducted with both bilinear and truncated ideal interpolators. In many cases there was noticeable blur in the pictures that were processed with the bilinear interpolator. The pictures processed with the truncated ideal interpolator were noticeably sharper, therefore this interpolator was used exclusively in the remaining experiments. This interpolator can be written as

$$r(x,y) = \alpha \sum_{n_1=-N}^{N} \sum_{n_2=-N}^{N} r[n_1,n_2]\phi(x,n_1,T_x)\phi(y,n_2,T_y) \tag{5.6}$$

where

$$r[n_1,n_2] = r(n_1T_x, n_2T_y) \tag{5.7}$$

$$\phi(z,n,T_z) = \frac{\sin\left[\left(\frac{\pi}{T_z}\right)(z-nT_z)\right]}{\left(\frac{\pi}{T_z}\right)(z-nT_z)} \tag{5.8}$$

and α was chosen so that the interpolation coefficients sum to unity. In the experiments which were conducted, N was equal to 3, so that the resulting interpolation filter has a 7 x 7 region of support.

5.4.1 Additive random noise

For pictures which have been degraded with additive random noise, motion-compensated restoration is accomplished with frame averaging. Therefore the signal estimate $\hat{s}(\bar{x}_0, t_0)$ is given by

$$\hat{s}(\bar{x}_0, t_0) = \frac{1}{3}(r_{past} + r_{current} + r_{future}). \tag{5.9}$$

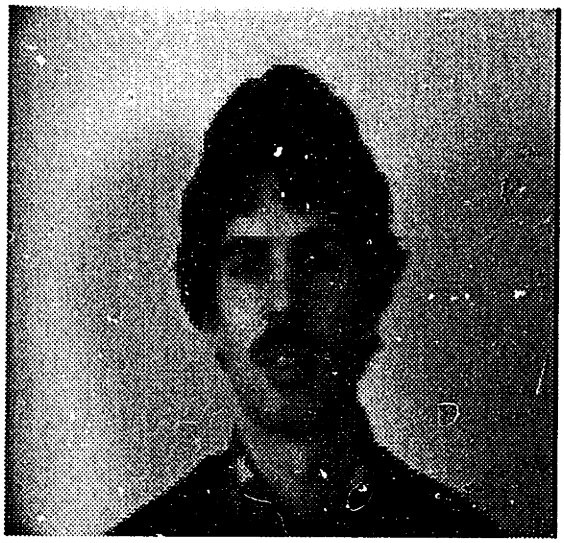
A number of experiments were performed in order to compare the results obtained with motion-compensated frame averaging to the single frame and multiple frame restoration methods described in the previous sections. In particular, we compared the following systems:

- motion-compensated frame averaging
- adaptive single frame restoration
- adaptive multiple frame restoration

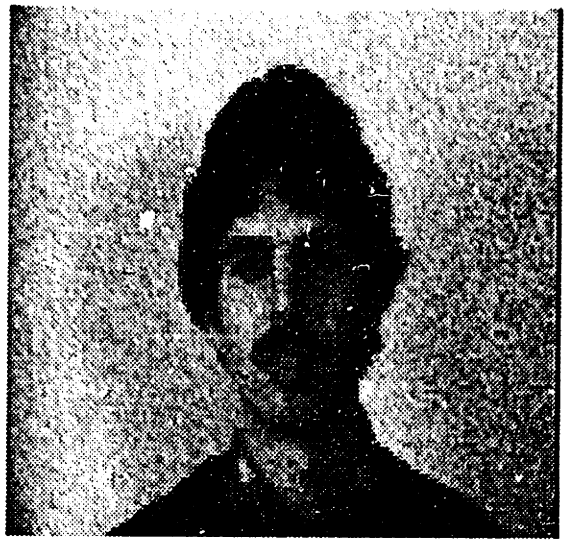
On a variety of test sequences the results were generally consistent. We can illustrate the results with an example. Figure 5.5 contains one frame from the original sequence and the corresponding degraded frame. The original frames were degraded with additive white Gaussian noise (standard deviation = 10). The resulting SNR was 16.5 dB. The spatial resolution of these pictures is 128 x 120 pels/frame, and the temporal sampling rate is 15 frames/second. Figure 5.6 contains the degraded frame and the processed frames using the three methods described previously.

Informal subjective evaluation of the sequences when viewed as a motion picture reveal the following observations:

- The sequences processed with only motion-compensated temporal averaging are the sharpest, with little or no visible blur. The noise is still visible, but remains spatially uncorrelated. The improvement in SNR is 4.7 dB.
- The sequences processed with the adaptive single frame restoration algorithm are very blurred. Although most of the visible noise is removed, there are visible artifacts in the pictures.



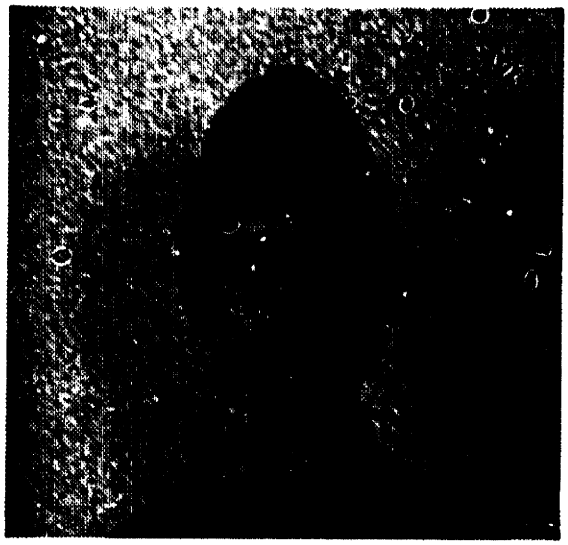
Original



Degraded

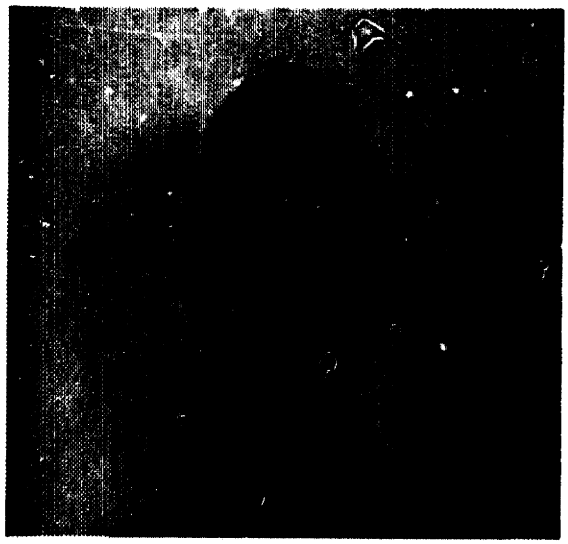
Figure 5.5: Additive noise test images

- The sequences processed with the adaptive multiple frame restoration system are better than the sequences processed with the single frame system. Most of the visible noise is removed in the background and slowly moving regions of the picture. The amount of blur is very minimal. However, in regions with moderate or large velocities, little noise is removed. This produces a noise field which is correlated with moving objects. A few visible artifacts are present in the pictures.



Degraded

Motion compensated FIR



Adaptive single frame

Adaptive multiple frame

Figure 5.6: Comparison of additive noise restoration systems

5.4.2 Impulsive noise

In order to demonstrate that the proposed restoration approach is applicable to other degradations as well, we also experimented with pictures that were degraded with impulsive noise due to random bit errors. The images were sampled and quantized with eight bits per sample. Consider the hypothetical problem of transmitting this sequence over a noisy channel.

We can model a wide variety of communication channels as a binary symmetric channel. A memoryless binary symmetric channel is characterized by P, the probability that an arbitrary bit is received in error at the receiver. If a pulse code modulated image is transmitted over a memoryless binary symmetric channel, then the intensity of random pels is modified. If the low-order bits of the pel are modified, there will be little or no visible difference. Conversely, if the high-order bits are altered, a dark pel may become a bright pel and vice versa. The net effect is to produce impulsive noise.

For pictures which have been degraded with random bit errors, restoration is accomplished with temporal median filters. Therefore the signal estimate $\hat{s}(\bar{x}_0, t_0)$ is given by

$$\hat{s}(\bar{x}_0, t_0) = MEDIAN(r_{past} + r_{current} + r_{future}). \tag{5.10}$$

We have experimented with several median filter topologies

- motion-compensated temporal median filter (3 point)
- spatial median filter (3 point vertical orientation)
- spatial median filter (5 point cross)

Figure 5.7 illustrates the effect of random bit errors. The bit error rate for this test sequence was $P = 0.02$. Figure 5.8 contains the frames processed with the methods described previously. Informal subjective evaluation of the sequences when viewed as a motion picture reveal the following observations:

- The amount of visible noise removed by the 3-point temporal and spatial median filters is essentially identical. Little or no blur is visible in the sequences.

Original Degraded

Figure 5.7: Random bit error test images

However, the spatial median filter introduces artifacts on the boundaries of object edges.

- The 5-point spatial filter removes essentially all the visible noise. However, it introduces significant blur into the picture. Occasionally there are some visible artifacts along the edges of moving objects.

Degraded

Motion compensated median

3 point spatial median

5 point spatial median

Figure 5.8: Comparison of impulsive noise restoration systems

5.5 Summary of noise reduction results

We implemented and compared several systems for noise reduction within motion pictures. For the case of additive noise we implemented a single frame restoration system based on a cascade of one-dimensional adaptive filters, an extension of this system to multiple frames, and a motion-compensated frame averaging system. The pictures processed with the motion-compensated system were generally preferred over the other two approaches.

For the case of impulsive noise we compared a 3-point motion-compensated median filter to a 3-point spatial median filter and a 5-point spatial median filter. Although the residual noise remaining with the two 3-point filters was essentially identical, the spatial filter introduces artifacts into the picture along the edges of objects. These artifacts are not noticed in the individual frames, but become apparent in the motion pictures. The pictures processed with the 5-point spatial median filter look very blurred relative to the pictures processed with the other filters, but the residual noise which remained was lower.

Chapter 6

Motion picture frame interpolation

Most motion picture sequences are obtained by sampling at a uniform temporal rate. However, for a variety of reasons it may be desirable (or even necessary) to display the sequence at a different rate. A common example of this problem occurs when motion picture films are shown on a conventional NTSC television system. The motion picture industry uses a standard frame rate of 24 frames per second. However, the NTSC standard uses a 2-to-1 interlaced format, scanned at a rate of 60 fields per second, or 30 frames per second. In order to show a motion picture film on an NTSC television system, temporal interpolation is necessary. The technique used in this case is known as the 3:2 pull down, in which a frame from the film is shown for 3 successive fields, followed by the next frame shown for 2 successive fields, and so on.

This technique can be generalized by using a temporal sample-and-hold interpolation scheme which operates as follows. If we are given a sequence $s(\bar{x}, t)$, at time instants $t_n = nT$ and we desire the frame corresponding to an arbitrary time $t = \tau$, then the frame at time instant $t = t_m$ is used, where t_m satisfies the inequality

$$|t_m - \tau| \leq |t_n - \tau| \quad \forall \quad n, m. \tag{6.1}$$

In other words, the interpolated frame is equal to the frame from the original

sequence which is closest in time to the desired frame. One of the primary problems with this approach is that the resulting sequence often exhibits jerky motion.

An alternative approach which has been suggested is to use motion-compensated interpolation [16,26]. This involves determining motion trajectories of objects in the scene and extrapolating their positions to the time instant where an interpolated frame is desired. Therefore two separate operations are performed; motion estimation and interpolation.

This method is based on the motion model

$$s(\bar{x}, t) = s(\bar{x} - \bar{v}(\bar{x}, t)(t - t_0), t_0). \tag{6.2}$$

Therefore the frame at an arbitrary time t can be computed from the frame at time t_0 by projecting the velocity field from the desired frame onto the given frame. In order to compute the interpolated pel value at spatio-temporal position $(\bar{x}_0, t)$, we first evaluate the velocity field at this position. The velocity field is projected onto the frame at time t_0, which is the frame closest in time to the desired frame. This procedure is illustrated in Figure 6.1.

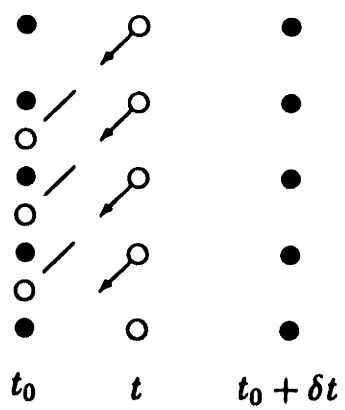


Figure 6.1: Velocity field projection

6.1 Interpolation experimental results

It is not possible to illustrate the motion rendition characteristics of these methods with only still pictures. However, we can illustrate the quality of the still frames generated with the motion-compensated interpolator. Figure 6.2 contains a set of three frames (two key original frames and an interpolated frame). These frames have a spatial resolution of 384 horizontal and 256 vertical pels, with a temporal sampling rate of 30 frames per second. They were obtained directly from an NTSC signal. The interpolated frame corresponds to the time instant midway between the two key original frames. This test sequence is actually in color. The velocity field is computed from the luminance component and is used to interpolate the RGB components individually. Most of the variation in the frames occurs around the person's lips.

These frame interpolation algorithms have been applied to the problem of frame rate modification. A number of experiments were conducted to compare these two approaches. Specifically, we have conducted the following experiments:

- Speed up by 10 and 20 percent.
- Slow down by 10 and 20 percent.
- Frame rate conversion from 24 frames per second (motion picture standard frame rate) to 60 fields per second (NTSC rates).

In general, the motion rendition in the sequences computed with the motion-compensated interpolation method is superior to the same sequences computed with the frame repetition method. The differences are most striking when the scene contains large moving objects.

Original frame 1

Interpolated frame

Original frame 2

Figure 6.2: Motion-compensated interpolated frame

Chapter 7

Conclusions

7.1 Contributions

This thesis was concerned primarily with the problem of motion estimation. This work was motivated in part by limitations of previously used approaches to motion estimation. The most significant contribution of this work was the development of a least squares motion estimation algorithm which has three important characteristics:

- motion estimation accuracy
- capability of estimating large velocities
- computational efficiency.

The primary limitation of previously used motion estimation algorithms is the failure to possess these three important characteristics.

There are two fundamental components to the least squares algorithm, a velocity field model and a signal model. The velocity field model is based on the conceptual principle of mapping single images into sequences of images with analytic mappings. A direct consequence of this mapping is that the velocity field is related linearly to the signal. The linear relationship motivates the use of the least squares error criterion so that a velocity field estimate can be obtained by solving linear equations.

To implement digital processors for performing motion estimation, it is necessary to sample the motion picture in space and time. Therefore, a motion estimation algorithm only has samples of the signal available. Furthermore, these samples are often corrupted with noise. To bridge the gap between the available sampled images and the continuous model which relates the signal to the velocity field, we introduced a general class of linear signal models. A least squares method is used to estimate the signal model parameters from the available samples. This process yields a continuous signal representation. This continuous signal representation is used to compute the least squares velocity estimate.

To demonstrate the usefulness of the least squares motion estimation algorithm, we explored two applications, noise reduction and frame interpolation. In a variety of experiments we demonstrated that a motion-compensated noise reduction system based on the least squares algorithm can yield better results than alternate noise reduction methods. This judgement was based on considerations of picture sharpness, residual noise, and visible artifacts introduced by the noise reduction system.

We also developed a system for frame rate modification of motion pictures. We demonstrated that a motion-compensated frame interpolation system based on the least squares algorithm yields better motion rendition than conventional frame repetition/dropping methods of frame rate modification.

7.2 Directions for future work

There are many areas where this thesis can be extended. We can partition these extensions into two categories, modeling and application.

7.2.1 Alternate velocity field models

A very general model for describing motion was presented in Appendix A. The model relates the velocity field to the signal with a linear partial differential equation. A zero-order velocity field approximation to the model forms the basis for

the least squares motion estimation algorithm. One of the potentially interesting extensions of this thesis is to explore the use of higher order velocity field models.

At each point where a velocity field estimate is desired, the least squares algorithm assumes the velocity field is constant over a small region in the neighborhood of the point of interest. Instead, one might explore the use of a first or second-order model which includes higher order terms of a Taylor series approximation to the velocity field. This approach has the important property that determining the coefficients with the least squares algorithm involves solving only linear equations. Therefore the resulting algorithm will still be computationally efficient.

7.2.2 Alternate signal models

The least squares algorithm uses a linear signal model to interpolate the available samples of the motion picture. Our implementation of the least squares algorithm uses a three-dimensional polynomial signal model. There are many other three dimensional functional forms which also can be used (for example trigonometric or exponential forms). To maintain computational efficiency, the only requirement is that the model remains linear in the parameters which characterize the signal.

7.2.3 Additional applications

This thesis explored on only two potential applications of motion estimation, noise reduction and frame interpolation. In addition to these applications, there are other applications which can benefit from the use of the least squares motion estimation algorithm. Two interesting applications include (1) conversion of interlaced fields into progressively scanned frames and (2) motion-compensated picture coding.

Appendix A

Models for describing motion

A.1 Introduction

In this appendix we develop some models for describing motion within image sequences. The models serve two primary purposes:

- The models provide a mathematical definition for motion in terms of velocity fields and several properties of the motion estimation problem can be deduced from these models.

- Several parametric forms of the models are the basis for computationally efficient motion estimation algorithms. The translational form of the model is analyzed in great detail. In addition we analyze the case of zooming and rotation.

These models are based on a continuous space-time representation of the signals. Although the models are formulated specifically for monochromatic pictures, they can be applied directly to color pictures. If an R-G-B representation is used, then the model can be applied directly to each color component individually, or to the luminance component which is obtained from the tricolor components.

A motion picture sequence is composed of a set of two-dimensional projections of a three-dimensional visual field. Each projection corresponds to the visual field at a particular instant of time. As objects within the field of view move, there

are corresponding changes in the projections. Each point within a two-dimensional projection is generated by superimposing the contributions due to all reflective surfaces within the scene, in response to all the light sources. This superposition occurs optically in photographic recording. Three quantities are required in order to construct this projection mathematically: (1) a complete specification of all the light sources involved in forming the picture, (2) a description of the geometry of all visible surfaces, and (3) specification of the reflectivity of all visible surfaces.

Spatio-temporal intensity variations within a sequence of projections are caused by many phenomena, which may occur individually or in combination. Some examples include:

- The objects in the field of view move relative to the light sources and observation point.
- The observer moves relative to the light sources and field of view.
- The light sources vary in time.

Strictly speaking, a complete motion description of a visual scene requires knowledge of the motion trajectories of all visible surfaces with respect to the light sources and observation point. Based on a complete three-dimensional description of the scene (including motion information), in principle one could determine the spatio-temporal intensity variations within the sequence of projections.

In the applications which we are interested in, only the sequence of projections is available. We do not have direct information about object reflectivity, surface geometry, object motion, or light source temporal variation. Therefore we can deduce motion information only from the picture sequence itself. The models which we develop specifically attempt to relate one frame to the next in terms of a motion description.

A.2 Parametric methods for modeling motion

It should be noted that there are many ways of characterizing motion. Therefore we are faced with the problem of selecting a suitable representation in which we can pose the motion estimation problem. Every motion representation is based on a model of some physical situation. For a motion model to be useful, it should apply to a wide variety of scenarios encountered in motion pictures.

We propose a parametric approach to modeling motion. There are several reasons for advocating a parametric motion representation. Our primary objective is not to model motion picture sequences, but to manipulate them. It is necessary to develop models which are useful from a mathematical perspective and are also computationally tractable. Parametric modeling procedures can often possess both of these characteristics and we will emphasize only those models which do.

It is recognized that there are signals which are not represented well by a given model. The consequences of this depend largely on how the signals are manipulated based on the model parameters and how the results are evaluated. In the context of picture processing, the ultimate criterion for evaluation is subjective examination of the processed sequences.

The outline of our development of parametric motion models is as follows:

- We begin with a very general representation for modeling motion. For this purpose we introduce the concept of motion description functions which define a mapping from a single image into a three-dimensional image sequence.

- It is shown that the motion description functions are related to the underlying signal through a partial differential equation. In principle, solving the partial differential equation permits determination of the motion description functions from the signal.

- There is a direct correspondence between motion description functions and velocity fields. The velocity field can be determined from a motion description function by solving a linear equation. Furthermore, a motion description

function can be determined from a velocity field by solving a linear partial differential equation.

- By restricting the functional form of the motion description function, in many cases the partial differential equation can be solved in a straightforward manner to determine both the motion description function and the velocity field from a given signal.

- For the cases of translation, zooming, and rotation, it is shown that very simple parametric representations exist. For these cases, determination of the motion parameters reduces to the problem of solving a set of linear equations.

A.3 Motion models in a cartesian space

In a cartesian coordinate system, motion is modeled as a mapping from a single two-dimensional luminance function $s_0(x, y)$, into a three-dimensional luminance function $s(x, y, t)$. This mapping is generated by the motion description functions $\alpha_x(x, y, t)$ and $\alpha_y(x, y, t)$

$$s(x, y, t) = s_0(\alpha_x(x, y, t), \alpha_y(x, y, t)). \tag{A.1}$$

By convention we impose the constraints

$$\alpha_x(x, y, t_0) = x \tag{A.2}$$

$$\alpha_y(x, y, t_0) = y \tag{A.3}$$

so that

$$s(x, y, t_0) = s_0(x, y). \tag{A.4}$$

This formulation is capable of describing a very broad class of motion types. For example, we can describe translation, zooming, rotation, and deformation. Figure A.1 illustrates the velocity fields associated with these motion types. One phenomenon which cannot be described directly with this formulation is the occlusion and uncovering of a background object by a moving object. In this case the velocity field is undefined in the newly uncovered regions. The same problem is encountered when there is a scene change where successive frames are completely different. These phenomena require special attention and are briefly discussed in Chapter 3 in the context of motion estimation algorithms.

Given this formulation, motion determination reduces to the problem of computing $\alpha_x(x, y, t)$ and $\alpha_y(x, y, t)$ from $s(x, y, t)$. The first step in our analysis of this model is to relate the motion description functions $\alpha_x(x, y, t)$ and $\alpha_y(x, y, t)$ to the signal $s(x, y, t)$. This relationship is stated in terms of a partial differential equation. In succeeding sections it is shown that by restricting the functional form of $\alpha_x(x, y, t)$ and $\alpha_y(x, y, t)$, we can derive closed form solutions to this differential equation given a signal $s(x, y, t)$.

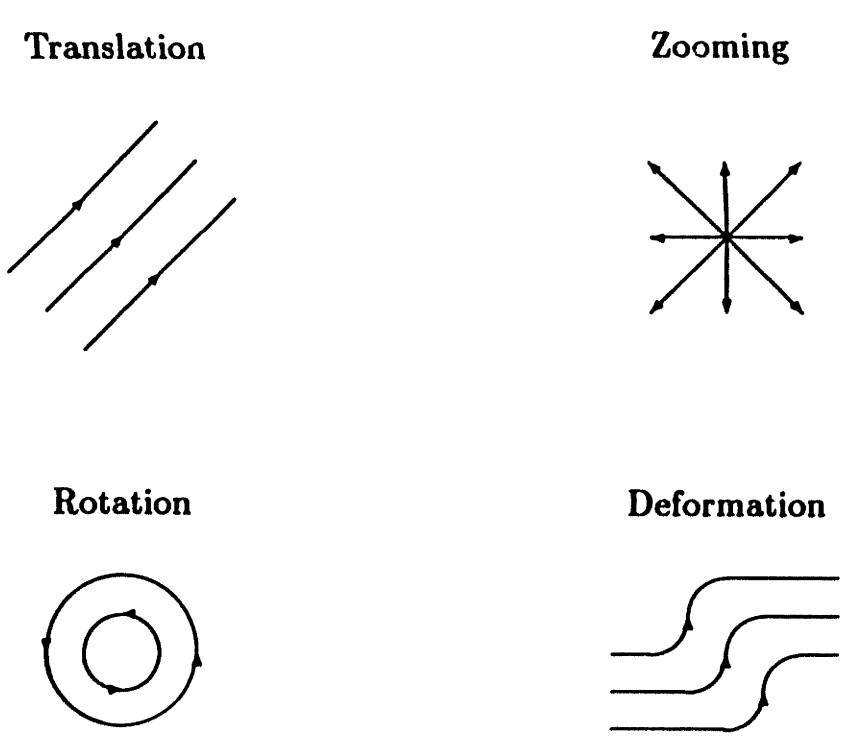


Figure A.1: Velocity fields

We begin with the set of partial derivatives of $s(x, y, t)$ with respect to the independent variables x, y, and t

$$\frac{\partial s}{\partial x} = \frac{\partial s_0}{\partial \alpha_x}\frac{\partial \alpha_x}{\partial x} + \frac{\partial s_0}{\partial \alpha_y}\frac{\partial \alpha_y}{\partial x} \tag{A.5}$$

$$\frac{\partial s}{\partial y} = \frac{\partial s_0}{\partial \alpha_x}\frac{\partial \alpha_x}{\partial y} + \frac{\partial s_0}{\partial \alpha_y}\frac{\partial \alpha_y}{\partial y} \tag{A.6}$$

$$\frac{\partial s}{\partial t} = \frac{\partial s_0}{\partial \alpha_x}\frac{\partial \alpha_x}{\partial t} + \frac{\partial s_0}{\partial \alpha_y}\frac{\partial \alpha_y}{\partial t}. \tag{A.7}$$

The partial derivatives with respect to x and y can be written in matrix notation as follows

$$\begin{bmatrix} \frac{\partial \alpha_x}{\partial x} & \frac{\partial \alpha_y}{\partial x} \\ \frac{\partial \alpha_x}{\partial y} & \frac{\partial \alpha_y}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial s_0}{\partial \alpha_x} \\ \frac{\partial s_0}{\partial \alpha_y} \end{bmatrix} = \begin{bmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial y} \end{bmatrix}. \tag{A.8}$$

From this system of equations we can solve for the partial derivatives of $s_0(\cdot)$ with respect to $\alpha_x(\cdot)$ and $\alpha_y(\cdot)$, provided the determinant of the matrix does not vanish. By substituting these quantities into Equation (A.7) we arrive at the desired partial

differential equation

$$\left(\frac{\partial \alpha_x}{\partial x}\frac{\partial \alpha_y}{\partial y} - \frac{\partial \alpha_x}{\partial y}\frac{\partial \alpha_y}{\partial x}\right)\frac{\partial s}{\partial t} = \left(\frac{\partial \alpha_y}{\partial y}\frac{\partial \alpha_x}{\partial t} - \frac{\partial \alpha_x}{\partial y}\frac{\partial \alpha_y}{\partial t}\right)\frac{\partial s}{\partial x} + \left(\frac{\partial \alpha_x}{\partial x}\frac{\partial \alpha_y}{\partial t} - \frac{\partial \alpha_y}{\partial x}\frac{\partial \alpha_x}{\partial t}\right)\frac{\partial s}{\partial y}. \tag{A.9}$$

The requirement that the determinant of the matrix is nonzero is equivalent to the condition

$$\frac{\partial \alpha_x}{\partial x}\frac{\partial \alpha_y}{\partial y} \neq \frac{\partial \alpha_x}{\partial y}\frac{\partial \alpha_y}{\partial x}. \tag{A.10}$$

Recall that Equation (A.9) is valid for all x, y, and t where the model applies. In the following subsections we present the solution to this differential equation for several specific motion types of interest.

It is instructive to rewrite Equation (A.9) in the following form

$$v_x(x,y,t)\frac{\partial s}{\partial x} + v_y(x,y,t)\frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} = 0, \tag{A.11}$$

where

$$v_x(x,y,t) = -\frac{\left(\frac{\partial \alpha_y}{\partial y}\frac{\partial \alpha_x}{\partial t} - \frac{\partial \alpha_x}{\partial y}\frac{\partial \alpha_y}{\partial t}\right)}{\left(\frac{\partial \alpha_x}{\partial x}\frac{\partial \alpha_y}{\partial y} - \frac{\partial \alpha_x}{\partial y}\frac{\partial \alpha_y}{\partial x}\right)} \tag{A.12}$$

and

$$v_y(x,y,t) = -\frac{\left(\frac{\partial \alpha_x}{\partial x}\frac{\partial \alpha_y}{\partial t} - \frac{\partial \alpha_y}{\partial x}\frac{\partial \alpha_x}{\partial t}\right)}{\left(\frac{\partial \alpha_x}{\partial x}\frac{\partial \alpha_y}{\partial y} - \frac{\partial \alpha_x}{\partial y}\frac{\partial \alpha_y}{\partial x}\right)}. \tag{A.13}$$

This representation is motivated by recognizing that the quantities $v_x(x,y,t)$ and $v_y(x,y,t)$ have the units of spatial distance over time. In fact, these quantities are the velocity field components, which form the basis for analyzing several motion types that can be described easily with this model. We can express this relationship with matrix notation as follows

$$\begin{bmatrix} \frac{\partial \alpha_x}{\partial x} & \frac{\partial \alpha_x}{\partial y} \\ \frac{\partial \alpha_y}{\partial x} & \frac{\partial \alpha_y}{\partial y} \end{bmatrix} \begin{bmatrix} v_x(x,y,t) \\ v_y(x,y,t) \end{bmatrix} = - \begin{bmatrix} \frac{\partial \alpha_x}{\partial t} \\ \frac{\partial \alpha_y}{\partial t} \end{bmatrix} \tag{A.14}$$

This demonstrates that the velocity field can be determined from the motion description functions by solving a set of linear equations. This set of linear solutions has a unique solution because the matrix is nonsingular as required by Equation (A.10). This set of linear equations also specifies two linear partial differential equations that can be solved to obtain $\alpha_x(x,y,t)$ and $\alpha_y(x,y,t)$ from the velocity field. Both equations have the form

$$\frac{\partial \alpha}{\partial x} v_x(x,y,t) + \frac{\partial \alpha}{\partial y} v_y(x,y,t) + \frac{\partial \alpha}{\partial t} = 0. \tag{A.15}$$

If $v_x(x,y,t)$ and $v_y(x,y,t)$ are valid velocity fields then this partial differential equation can be solved to determine functional forms for $\alpha_x(x,y,t)$ and $\alpha_y(x,y,t)$. These forms are reduced to a specific function by introducing the boundary conditions $\alpha_x(x,y,t_0) = x$ and $\alpha_y(x,y,t_0) = y$. The following subsections illustrate this procedure for some simple cases.

One should contrast the result of Equation (A.11) with the spatio-temporal constraint equation described in Chapter 2.

Simple translation:

$$\begin{gathered} s(x,y,t) = s_0(x - v_x \cdot (t - t_0), y - v_y \ \ (t - t_0)) \\ \Updownarrow \\ v_x \frac{\partial s}{\partial x} + v_y \frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} = 0 \end{gathered} \tag{A.16}$$

General motion:

$$\begin{gathered} s(x,y,t) = s_0(\alpha_x(x,y,t), \alpha_y(x,y,t)) \\ \Updownarrow \\ v_x(x,y,t) \frac{\partial s}{\partial x} + v_y(x,y,t) \frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} = 0 \end{gathered} \tag{A.17}$$

Note that the case of simple translation is a special case of analytic mapping in which the velocity field is not a function of spatio-temporal position.

We have shown that if the motion picture sequence is obtained by mapping a single frame onto a sequence of frames with a motion description function, then there

exists a unique velocity field associated with this sequence. It is important to note that the converse is also true. If a motion picture satisfies Equation (A.11), then there is an associated motion description function. Therefore the motion description function representation and the velocity field representation imply each other. This follows because Equation (A.11) is a linear first-order partial differential equation and the most general solution to this equation is the motion description function representation.

The importance of the velocity field is that the signal remains constant along directions parallel to the velocity field at each point. To show this, consider the total differential of the signal $s(x, y, t)$

$$ds(x, y, t) = \frac{\partial s}{\partial x}dx + \frac{\partial s}{\partial y}dy + \frac{\partial s}{\partial t}dt. \tag{A.18}$$

The total differential relates the change in luminance to differential changes in spatio-temporal position along the direction $(dx, dy, dt)^T$. If we define the direction $d\bar{z}$ as follows

$$d\bar{z} = (v_x(\bar{x}, t), v_y(\bar{x}, t), 1)^T dz \tag{A.19}$$

then Equation (A.11) states that along direction $d\bar{z}$, the differential change in luminance is zero and therefore the signal is constant along this direction. The field lines determined by the vector field $(v_x(\bar{x}, t), v_y(\bar{x}, t), 1)^T$ are referred to as "optical flow lines".

The difficulty with this representation is that the velocity field cannot be determined uniquely from a signal $s(x, y, t)$. To demonstrate this, consider any vector $(\nu_x(x, y, t), \nu_y(x, y, t), 0)^T$, such that

$$\nu_x(x, y, t)\frac{\partial s}{\partial x} + \nu_y(x, y, t)\frac{\partial s}{\partial y} = 0. \tag{A.20}$$

The two-dimensional vector $(\nu_x(x, y, t), \nu_y(x, y, t))^T$ is orthogonal to the spatial gradient of the signal $s(x, y, t)$ at every point. By direct substitution it follows that the velocity field defined by

$$\bar{v}_{new}(x, y, t) = \bar{v}(x, y, t) + \bar{\nu}(x, y, t) \tag{A.21}$$

still satisfies Equation (A.11). Consequently, the optical flow lines are not defined uniquely along contours where $s(x, y, t)$ is constant.

Our primary interest is to extract a unique velocity field from a given signal. An additional constraint is necessary in order to accomplish this goal. The additional constraint which we impose is structural. This means we force the velocity field to have a specific spatio-temporal structure. There are two cases which we consider. The first case treats velocity fields which have a specific spatial structure, but an arbitrary temporal structure. Next we specialize this result so that the velocity field is constant along the temporal direction. We restrict the spatial structure to relatively simple forms that can be defined in terms of a set of parameters. This is how we formulate parametric velocity field models.

A.3.1 Translation

A model for translation is based on the assumption that there is a region within the signal space which is translating along a fixed direction. We will use the symbol Ψ to denote this region. For the case of translation, the motion description functions $\alpha_x(x, y, t)$ and $\alpha_y(x, y, t)$ can be written as

$$\alpha_x(x, y, t) = x - D_x(t); \quad \text{with} \quad D_x(t_0) = 0 \tag{A.22}$$

$$\alpha_y(x, y, t) = y - D_y(t); \quad \text{with} \quad D_y(t_0) = 0 \tag{A.23}$$

where $D_x(t)$ and $D_y(t)$ are the components of the displacement field. The velocity field is

$$v_x = \frac{dD_x(t)}{dt} \qquad v_y = \frac{dD_y(t)}{dt}. \tag{A.24}$$

The partial derivatives of $\alpha_x(\cdot)$ and $\alpha_y(\cdot)$ with respect to x, y, and t are

$$\frac{\partial \alpha_x}{\partial x} = 1; \; \frac{\partial \alpha_x}{\partial y} = 0; \; \frac{\partial \alpha_x}{\partial t} = -\frac{dD_x}{dt} \tag{A.25}$$

$$\frac{\partial \alpha_y}{\partial x} = 0; \; \frac{\partial \alpha_y}{\partial y} = 1; \; \frac{\partial \alpha_y}{\partial t} = -\frac{dD_y}{dt}. \tag{A.26}$$

It follows by inspection that Equation (A.10) is satisfied. Substituting these derivatives into Equation (A.9) we obtain the result

$$\frac{dD_x}{dt}\frac{\partial s}{\partial x} + \frac{dD_y}{dt}\frac{\partial s}{\partial y} = -\frac{\partial s}{\partial t} \tag{A.27}$$

which is assumed to be valid at all points within Ψ. In particular, for any two spatial positions

$$(x_0, y_0, t) = P_0 \tag{A.28}$$

$$(x_1, y_1, t) = P_1 \tag{A.29}$$

within Ψ, we can generate the following set of linear equations

$$\frac{d\alpha_x}{dt}\frac{\partial s}{\partial x}\bigg|_{P_0} + \frac{d\alpha_y}{dt}\frac{\partial s}{\partial y}\bigg|_{P_0} = -\frac{\partial s}{\partial t}\bigg|_{P_0} \tag{A.30}$$

$$\frac{d\alpha_x}{dt}\frac{\partial s}{\partial x}\bigg|_{P_1} + \frac{d\alpha_y}{dt}\frac{\partial s}{\partial y}\bigg|_{P_1} = -\frac{\partial s}{\partial t}\bigg|_{P_1}. \tag{A.31}$$

In order to simplify the notation, define the matrix

$$A_{trans}(t) = \begin{bmatrix} \dfrac{\partial s}{\partial x}\Big|_{P_0} & \dfrac{\partial s}{\partial y}\Big|_{P_0} \\ \dfrac{\partial s}{\partial x}\Big|_{P_1} & \dfrac{\partial s}{\partial y}\Big|_{P_1} \end{bmatrix} \tag{A.32}$$

and the column vectors

$$\bar{v}(t) = \begin{bmatrix} \dfrac{da_x}{dt} \\ \dfrac{da_y}{dt} \end{bmatrix} \qquad \bar{b}_{trans}(t) = -\begin{bmatrix} \dfrac{\partial s}{\partial t}\Big|_{P_0} \\ \dfrac{\partial s}{\partial t}\Big|_{P_1} \end{bmatrix}. \tag{A.33}$$

This allows Equations (A.30) and (A.31) to be written as

$$A_{trans}(t)\bar{v}(t) = \bar{b}_{trans}(t) \tag{A.34}$$

which possesses a unique solution for all time if and only if $DET\,[A_{trans}(t)] \neq 0$. The solution is given by

$$\bar{v}(t) = \frac{d}{dt}\bar{D}(t) = A^{-1}_{trans}(t)\bar{b}_{trans}(t). \tag{A.35}$$

Integrating the velocity field between two time instants yields the displacement field

$$\bar{D}(t, t_0) = \int_{t_0}^{t} A^{-1}_{trans}(\gamma)\bar{b}(\gamma)d\gamma. \tag{A.36}$$

A special case of this result is to consider only constant velocity fields which model the situation where there is no acceleration of the objects within the scene. In this case the motion description functions become

$$\alpha_x(t) = v_x \cdot (t - t_0) \tag{A.37}$$

$$\alpha_y(t) = v_y \cdot (t - t_0). \tag{A.38}$$

Substituting these relations into Equation (A.27) yields the result

$$v_x \frac{\partial s}{\partial x} + v_y \frac{\partial s}{\partial y} + \frac{\partial s}{\partial t} = 0. \tag{A.39}$$

Equation (A.39) is the spatio-temporal constraint equation described in Chapter 2.

Evaluating the partial derivatives of the signal at a particular point (x_0, y_0, t_0) generates a linear constraint on the values of v_x and v_y. This constraint can be illustrated graphically as shown in Figure A.2. The constraint equation requires the

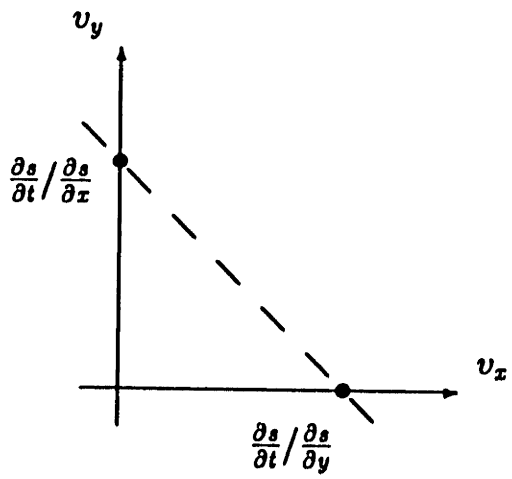


Figure A.2: Translational velocity constraint

values of v_x and v_y to lie on the dashed line. The set of linear equations specified by Equation (A.34) corresponds to locating the intersection of two constraint lines.

Since the velocity field is uniform in both space and time, we can drop the time dependence of the quantities in Equation (A.34), resulting in

$$A_{trans}\bar{v} = \bar{b}_{trans}. \tag{A.40}$$

In order to compute $\bar{v}$, there are three cases of interest:

- $A_{trans} = 0$: The velocity field is completely unconstrained. This occurs when the spatial gradients are identically zero.
- $Det(A_{trans}) \neq 0$: The two constraint lines intersect at a single point and there is a unique solution for the velocity field.
- $Det(A_{trans}) = 0$: The two constraint lines are collinear and only a linear constraint on the components of the velocity field is specified. This occurs when the region Ψ only contains edges oriented along some direction. Only the component of the velocity field which is orthogonal to the edge is defined uniquely.

A.3.2 Zooming

In this subsection we present a model for zooming and derive a closed-form solution that is similar to the case of translation. A model for zooming about the origin of a cartesian coordinate system is based on the following motion description functions [1]

$$\alpha_x(x, y, t) = xa_x(t); \quad \text{with} \quad a_x(t_0) = 1 \tag{A.41}$$

$$\alpha_y(x, y, t) = ya_y(t); \quad \text{with} \quad a_x(t_0) = 1. \tag{A.42}$$

Evaluating the partial derivatives of $a_x(t)$ and $a_y(t)$ with respect to x, y, and t, then substituting into Equation (A.9) yields the result

$$\frac{x}{a_x}\frac{da_x}{dt}\frac{\partial s}{\partial x} + \frac{y}{a_y}\frac{da_y}{dt}\frac{\partial s}{\partial y} = \frac{\partial s}{\partial t} \tag{A.43}$$

[1] The more general case of zooming about an arbitrary point can be treated in a similar manner by redefining the origin.

which can be simplified to

$$x\frac{d\log(a_x)}{dt}\frac{\partial s}{\partial x}+y\frac{d\log(a_y)}{dt}\frac{\partial s}{\partial y}=\frac{\partial s}{\partial t}. \tag{A.44}$$

The velocity field for this case is

$$v_x(x,y,t)=-x\frac{d\log\left(a_x(t)\right)}{dt} \tag{A.45}$$

$$v_y(x,y,t)=-y\frac{d\log\left(a_y(t)\right)}{dt}. \tag{A.46}$$

Equation (A.10) is satisfied provided $a_x(t)\neq 0$ and $a_y(t)\neq 0$. Assuming Equation (A.44) is valid for all points in some region Ψ and selecting two spatial positions

$$(x_0,y_0,t)=P_0 \tag{A.47}$$

$$(x_1,y_1,t)=P_1 \tag{A.48}$$

within Ψ, we can generate the following set of equations

$$\frac{d\log(a_x)}{dt}x\frac{\partial s}{\partial x}\bigg|_{P_0}+\frac{d\log(a_y)}{dt}y\frac{\partial s}{\partial y}\bigg|_{P_0}=\frac{\partial s}{\partial t}\bigg|_{P_0} \tag{A.49}$$

$$\frac{d\log(a_x)}{dt}x\frac{\partial s}{\partial x}\bigg|_{P_1}+\frac{d\log(a_y)}{dt}y\frac{\partial s}{\partial y}\bigg|_{P_1}=\frac{\partial s}{\partial t}\bigg|_{P_1} \tag{A.50}$$

To simplify the notation, define the matrix

$$A_{zoom}(t)=\begin{bmatrix} x\dfrac{\partial s}{\partial x}\Big|_{P_0} & y\dfrac{\partial s}{\partial y}\Big|_{P_0} \\ x\dfrac{\partial s}{\partial x}\Big|_{P_1} & y\dfrac{\partial s}{\partial y}\Big|_{P_1} \end{bmatrix} \tag{A.51}$$

and the column vectors

$$\bar{f}(t)=\begin{bmatrix} \dfrac{d\log(a_x)}{dt} \\ \dfrac{d\log(a_y)}{dt} \end{bmatrix} \qquad \bar{b}_{zoom}(t)=-\begin{bmatrix} \dfrac{\partial s}{\partial t}\Big|_{P_0} \\ \dfrac{\partial s}{\partial t}\Big|_{P_1} \end{bmatrix}. \tag{A.52}$$

We can rewrite Equations (A.49) and (A.50) in the following form

$$A_{zoom}(t)\bar{f}(t)=\bar{b}_{zoom}(t). \tag{A.53}$$

A unique solution for $\bar{f}(t)$ exists if and only if $DET\,[A_{zoom}(t)] \neq 0$. The solution is given by

$$\bar{f}(t) = C^{-1}_{zoom}(t)\bar{b}_{zoom}(t) \tag{A.54}$$

and the velocity field is

$$v_x(t) = -xf_{zoom(x)}(t) \qquad v_y(t) = -yf_{zoom(y)}(t). \tag{A.55}$$

A uniform velocity field is specified by the motion description functions

$$a_x(t) = \exp\left[z_x(t - t_0)\right] \quad \Longrightarrow \quad f_{zoom(x)} = z_x \tag{A.56}$$

$$a_y(t) = \exp\left[z_y(t - t_0)\right] \quad \Longrightarrow \quad f_{zoom(y)} = z_y \tag{A.57}$$

where z_x and z_y are the zooming factors along the x and y directions respectively. The velocity field is

$$v_x = -z_x x \qquad v_y = -z_y y \tag{A.58}$$

and the zooming parameters z_x and z_y are obtained from

$$C_{zoom}\bar{z} = \bar{b}_{zoom}. \tag{A.59}$$

Therefore the zooming parameters can be determined uniquely if and only if C_{zoom} is nonsingular. It is important to note the strong connection between this result and the result for the case of uniform translation shown in Equation (A.34)

A.4 Motion models in a rotational space

Following a procedure analogous to that in Section A.3, we can formulate similar results within polar coordinates. This permits us to treat the case of rotation with a straightforward mathematical development. In polar coordinates we are concerned with motion descriptions of the form

$$s(r,\theta,t) = s_0(\alpha_r(r,\theta,t), \alpha_\theta(r,\theta,t)) \tag{A.60}$$

together with the constraints

$$\alpha_r(r,\theta,t_0) = r \tag{A.61}$$

$$\alpha_\theta(r,\theta,t_0) = \theta \tag{A.62}$$

so that

$$s(r,\theta,t_0) = s_0(r,\theta). \tag{A.63}$$

The partial differential equation which relates $\alpha_r(r,\theta,t)$, and $\alpha_\theta(r,\theta,t)$ to $s(r,\theta,t)$ can be found from Equation (A.9) by performing variable substitution. Associating x with r, and y with θ we arrive at

$$\left(\frac{\partial\alpha_r}{\partial r}\frac{\partial\alpha_\theta}{\partial\theta} - \frac{\partial\alpha_r}{\partial\theta}\frac{\partial\alpha_\theta}{\partial r}\right)\frac{\partial s}{\partial t} = \left(\frac{\partial\alpha_\theta}{\partial\theta}\frac{\partial\alpha_r}{\partial t} - \frac{\partial\alpha_r}{\partial\theta}\frac{\partial\alpha_\theta}{\partial t}\right)\frac{\partial s}{\partial r} + \left(\frac{\partial\alpha_r}{\partial r}\frac{\partial\alpha_\theta}{\partial t} - \frac{\partial\alpha_\theta}{\partial r}\frac{\partial\alpha_r}{\partial t}\right)\frac{\partial s}{\partial\theta} \tag{A.64}$$

provided

$$\frac{\partial\alpha_r}{\partial r}\frac{\partial\alpha_\theta}{\partial\theta} \neq \frac{\partial\alpha_r}{\partial\theta}\frac{\partial\alpha_\theta}{\partial r}. \tag{A.65}$$

The velocity field is

$$v_r(r,\theta,t) = -\frac{\left(\frac{\partial\alpha_\theta}{\partial\theta}\frac{\partial\alpha_r}{\partial t} - \frac{\partial\alpha_r}{\partial\theta}\frac{\partial\alpha_\theta}{\partial t}\right)}{\left(\frac{\partial\alpha_r}{\partial r}\frac{\partial\alpha_\theta}{\partial\theta} - \frac{\partial\alpha_r}{\partial\theta}\frac{\partial\alpha_\theta}{\partial r}\right)} \tag{A.66}$$

$$v_\theta(r,\theta,t) = -\frac{\left(\frac{\partial\alpha_r}{\partial r}\frac{\partial\alpha_\theta}{\partial t} - \frac{\partial\alpha_\theta}{\partial r}\frac{\partial\alpha_r}{\partial t}\right)}{\left(\frac{\partial\alpha_r}{\partial r}\frac{\partial\alpha_\theta}{\partial\theta} - \frac{\partial\alpha_r}{\partial\theta}\frac{\partial\alpha_\theta}{\partial r}\right)} \tag{A.67}$$

A.4.1 Rotation

For the case of rotation about the origin, $\alpha_r(\cdot)$ and $\alpha_\theta(\cdot)$ can be written as

$$\alpha_r(r,\theta,t) = r - \rho(t); \quad \text{with} \quad \rho(t_0) = 0 \tag{A.68}$$

$$\alpha_\theta(r,\theta,t) = \theta - \phi(t); \quad \text{with} \quad \phi(t_0) = 0 \tag{A.69}$$

and the corresponding velocity field is

$$v_r(t) = \frac{d\rho(t)}{dt} \qquad v_\theta(t) = \frac{d\phi(t)}{dt}. \tag{A.70}$$

Evaluating the partial derivatives with respect to r, θ, and t, then substituting into Equation (A.64) yields the result

$$\frac{d\rho(t)}{dt}\frac{\partial s}{\partial r} + \frac{d\phi(t)}{dt}\frac{\partial s}{\partial \theta} = -\frac{\partial s}{\partial t}. \tag{A.71}$$

In a manner identical to the cases of uniform translation and zooming, we select two spatial positions

$$(r_0,\theta_0,t) = P_0 \tag{A.72}$$

$$(r_1,\theta_1,t) = P_1 \tag{A.73}$$

and generate the linear equations

$$A_{rot}(t)\bar{v}(t) = \bar{b}_{rot}(t), \tag{A.74}$$

where

$$A_{rot}(t) = \begin{bmatrix} \left.\frac{\partial s}{\partial r}\right|_{P_0} & \left.\frac{\partial s}{\partial \theta}\right|_{P_0} \\ \left.\frac{\partial s}{\partial r}\right|_{P_1} & \left.\frac{\partial s}{\partial \theta}\right|_{P_1} \end{bmatrix} \qquad \bar{b}_{rot}(t) = -\begin{bmatrix} \left.\frac{\partial s}{\partial t}\right|_{P_0} \\ \left.\frac{\partial s}{\partial t}\right|_{P_1} \end{bmatrix}. \tag{A.75}$$

A unique solution for the velocity field exists if and only if $Det\,[A_{rot}(t)] \neq 0$ for all t. For uniform rotational fields, we drop the time dependence of the parameters and assume v_r and v_θ are constant, resulting in the equations

$$A_{rot}\bar{v} = \bar{b}_{rot}. \tag{A.76}$$

A.5 Discussion of models

In summary, we have developed several parametric models for describing motion that occurs within image sequences. The models are based on the use of motion description functions. There is a direct relationship between the motion description functions and the velocity field. A motion description function defines a unique velocity field and a motion description function can be obtained from a velocity field.

The velocity field is related to the signal through a partial differential equation. However, the differential equation does not completely constrain the velocity field. Conceptually, this is because an image is a scalar-valued function, while the velocity field is a vector function. Each point in the image corresponds to one "equation", but there are two components to the velocity field.

In order to extract a unique velocity field from a given signal, it is necessary to impose an additional constraint on the velocity field. We have demonstrated that parametric structural constraints for modeling translation, zooming, and rotation greatly reduce the ambiguity in the velocity field. These models also have the important property that they are computationally efficient. In particular, determining translational, zooming, and rotational parameters involves solving linear equations with two unknowns. The linear properties of these procedures can be extended directly to the problem of parameter estimation. If a least squares error criterion is used, the parameter estimates are also obtained by solving a set of linear equations. These algorithms can be implemented in a computationally efficient manner.

Appendix B

Motion estimation by region matching

The most widely used methods for motion estimation are based on region matching or correlation algorithms. Because of the widespread use of this approach, we have implemented a region matching algorithm which serves as a baseline system for comparing the performance of several other algorithms. In this appendix we describe the implementation details of this algorithm. A similar algorithm was extensively studied by Hinman [12].

Suppose we want an estimate of the velocity field at an arbitrary spatio-temporal position $(\bar{x}_0, t)$, where t may not coincide with a temporal sampling instant. Let t_0 and t_1 be two temporal sampling instants, such that $t_0 \leq t < t_1$. The velocity estimate is obtained from the frames $r(\bar{x}, t_0)$ and $r(\bar{x}, t_1)$ by minimizing the following expression

$$\min_{\bar{v}} \left\{ f(\bar{v}) = \sum_{i=1}^{N} [r(\bar{x}_i - \bar{v} \cdot (t - t_0), t_0) - r(\bar{x}_i - \bar{v} \cdot (t - t_1), t_1)]^2 \right\} \qquad \text{(B.1)}$$

A more general formulation of this algorithm includes a weighting coefficient for each term of the sum [24].

Note that in this expression the objective function $f(\bar{v})$ is a nonlinear function of the velocity vector and there is not a closed-form solution to this equation. Therefore the velocity vector which minimizes the objective function is determined

numerically with a nonlinear programming algorithm. This is an unconstrained optimization problem in the vector $\bar{v}$.

Since we only have samples of $r(\bar{x}, t)$ available, solving Equation (B.1) requires computing values of $r(\bar{x}, t)$ which are not on the sampling grid. This is accomplished with a spatial interpolator. There are two primary considerations involved in selecting an interpolator; interpolation accuracy and computational complexity. We have compared two different interpolation methods. The first method is based on a truncated interpolation kernel that approximates an ideal interpolator. This interpolator can be written as

$$s(x, y) = \sum_{n_1=-N}^{N} \sum_{n_2=-N}^{N} s[n_1, n_2]\phi(x, n_1, T_x)\phi(y, n_2, T_y) \tag{B.2}$$

where

$$s[n_1, n_2] = s(n_1 T_x, n_2 T_y) \tag{B.3}$$

and

$$\phi(z, n, T_z) = \frac{\sin\left[\left(\frac{\pi}{T_z}\right)(z - nT_z)\right]}{\left(\frac{\pi}{T_z}\right)(z - nT_z)}. \tag{B.4}$$

The second method uses a bilinear interplator. If x, y are integers, and d_x, d_y are displacements limited to the interval (-1,1), then the interpolated value is given by

$$\begin{aligned} s(x + d_x, y + d_y) \;=\; & (1 - d_x)(1 - d_y)s(x, y) + \\ & (d_x)(d_y)s(x + 1, y + 1) + \\ & (1 - d_x)(d_y)s(x, y + 1) + \\ & (d_x)(1 - d_y)s(x + 1, y). \end{aligned} \tag{B.5}$$

Several experiments were performed to compare the motion estimation errors which result when each of these interpolators it used. It was found that the bilinear interpolator produced uniformly smaller motion estimation errors than the truncated ideal interpolator for all signal-to-noise levels. Furthermore, the bilinear interpolator requires significantly less computation. Therefore in all the remaining experiments the bilinear interpolator was used exclusively.

We experimented with two different optimization procedures; a steepest decent method and a quasi-Newton method. Both procedures are descent algorithms which can be written in the form [1]

$$\bar{v}_{k+1} = \bar{v}_k - \alpha_k H_k \nabla f(\bar{v}_k) \tag{B.6}$$

where H_k is a positive definite steering matrix, and α_k is chosen to minimize the function

$$f(\bar{v}_k - \alpha_k H_k \nabla f(\bar{v}_k)). \tag{B.7}$$

The steering matrix operates on the gradient to produce a descent direction.

A steepest decent algorithm uses the identity matrix as the steering matrix, so the descent iteration becomes

$$\bar{v}_{k+1} = \bar{v}_k - \alpha_k \nabla f(\bar{v}_k). \tag{B.8}$$

One of the difficulties with the steepest descent algorithm is that the convergence rate is very slow if the objective function is highly elliptical. Quasi-Newton methods are often used in order to improve the convergence rate. Quasi-Newton methods exploit the property that if the steering matrix is approximately equal to the inverse hessian of the objective function in the vicinity of the optimal solution, then the convergence rate is considerably faster than steepest descent. The specific quasi-Newton method which we have used is a member of the Broyden family of algorithms, of which the Davidon-Fletcher-Powell method (DFP), is a special case. This family of methods construct an approximation to the inverse hessian during the descent process by using a sequence of rank 1 corrections. The implementation which we used applies the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update formula

$$H_{k+1} = H_k + \left(\frac{1 + \bar{q}_k^T H_k \bar{q}_k}{\bar{q}_k^T \bar{p}_k}\right) - \frac{\bar{p}_k \bar{q}_k^T H_k + H_k \bar{q}_k \bar{p}_k^T}{\bar{q}_k^T \bar{p}_k} \tag{B.9}$$

where H_k is the approximate inverse hessian for the k^{th} iteration. This algorithm operates as follows.

[1] These methods are commonly used for minimizing a nonlinear function of a vector. A complete discussion and analysis of these algorithms is presented by Luenberger [20].

Let $H_0 = I$ and $\bar{g}_k = \nabla f(\bar{v}_k)$ at each iteration.

Step 1: Compute the descent direction $\bar{d}_k = -H_k \bar{g}_k$.

Step 2: Minimize $f(\bar{v}_k + \alpha_k \bar{d}_k)$ with respect to α_k, to obtain $\bar{v}_{k+1} = \bar{v}_k + \alpha_k \bar{d}_k$.

Step 4: Compute $\bar{p}_k = \alpha_k \bar{d}_k$ and $\bar{q}_k = \bar{g}_{k+1} - \bar{g}_k$

Step 5: Update the inverse hessian according to Equation (B.9).

Step 6: Return to step 1.

Both the steepest descent and BFGS algorithms involve a line search, where the function

$$g(\alpha) = f(\bar{v} + \alpha \bar{d}) \tag{B.10}$$

is minimized. By virtue of its construction, only positive values of α are involved in the minimization and $g(\alpha)$ is guaranteed to possess a negative derivative at $\alpha = 0$.

The line search procedure which we used is based on an iterative quadratic curve fit and involves two steps. First, a value of α is found such that $g(\alpha)$ possesses a positive derivative. Let this point be denoted as α_{max}, and let $\alpha_{min} = 0$. If the function is unimodal, there is a unique point where $g(\alpha)$ has zero slope and hence is a local minimum of the function. This local minimum lies between α_{min} and α_{max}. A quadratic function $q(\alpha)$ is determined that has the same derivatives as $g(\alpha)$ at the two endpoints α_{min} and α_{max}. The position of the stationary point of $q(\alpha)$ is taken as an approximation to the value of α which minimizes $g(\alpha)$. Let this value be denoted as $\hat{\alpha}$. The derivative of $g(\alpha)$ at $\hat{\alpha}$ is evaluated and if it is positive, then α_{max} is set to $\hat{\alpha}$, otherwise α_{min} is set to $\hat{\alpha}$. This process is repeated until the difference between α_{max} and α_{min} is less than some threshold. The value of $\hat{\alpha}$ at the end of the iteration is taken as the value of α which minimizes Equation (B.10). One iteration of this procedure is illustrated in Figure B.1. In Appendix D we prove that this algorithm converges to the stationary points of $g(\alpha)$.

Initial experiments with both the steepest descent and BFGS algorithms revealed the following properties. If all the samples used in forming the estimate lie in a region with a perfect edge, the objective function is constant along lines that are parallel to the edge and the hessian becomes singular. This poses no problem for the steepest descent algorithm, but the BFGS algorithm becomes numerically

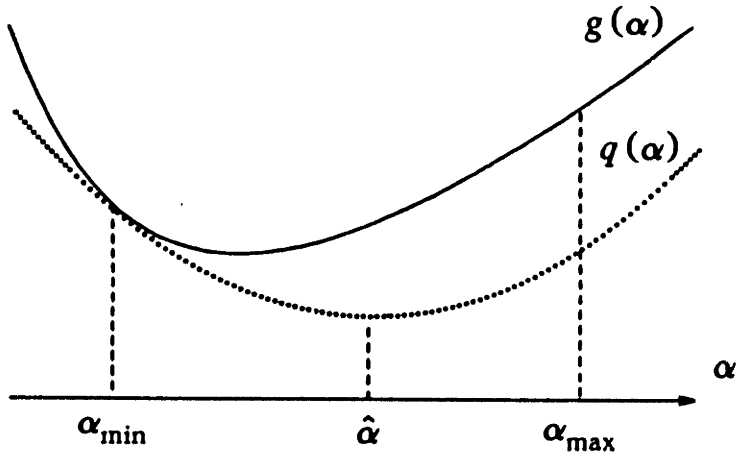


Figure B.1: Iterative line search procedure

unstable because the inverse hessian does not exist. Furthermore, the objective function is rarely approximated well by a quadratic function. Therefore the descent direction chosen by the BFGS algorithm is not always substantially better that the direction of steepest descent. This was confirmed by noting that the convergence rate of the BFGS algorithm was only slightly faster than the convergence rate of the steepest descent method, but involves more computation. For these reasons the remainder of the experiments used the steepest descent algorithm exclusively.

B.1 Summary of region matching method

In summary, the region matching algorithm which we have implemented generates continuous estimates of the velocity field at arbitrary spatio-temporal positions and involves three primary components:

- The velocity field is determined by minimizing the sum of squared differences between two displaced frames.
- A steepest descent algorithm is used to minimize the objective function. The line search is accomplished with an iterative quadratic curve fit procedure.
- A bilinear interpolation procedure is used to compute the values of the signal at points which are not on the sampling grid.

One of the problems with this algorithm is that the objective function does not have continuous first-order partial derivatives at all points (the derivatives are discontinuous at the sampling points). Therefore it is not possible to guarantee convergence of the algorithm (refer to Appendix D). In practice it was found that this is only a problem at low signal-to-noise levels.

Appendix C

Cramer Rao Bounds

In this appendix we derive the Cramer Rao bounds which apply to the motion estimation algorithms described in Chapter 3. This derivation is based on the presentation developed by Van Trees [36]. The bounds are derived for the case when a known signal is degraded with additive white Gaussian noise [1]. Therefore, the observed signal $r(x, y, t)$ is defined as

$$r(x, y, t) = s(x, y, t) + n(x, y, t). \tag{C.1}$$

The bounds are derived for a translation motion model in which the signal satisfies the relation

$$s(x, y, t) = s_0(x - v_x \cdot (t - t_0), y - v_y \cdot (t - t_0)). \tag{C.2}$$

We are interested in deriving the bounds when our observations consist of samples of $r(x, y, t)$ on an arbitrary sampling grid. Therefore, assume we are given N discrete observations

$$\begin{aligned} r_1 &= r(x_1, y_1, t_1) \\ r_2 &= r(x_2, y_2, t_2) \\ &\vdots \\ r_N &= r(x_N, y_N, t_N) \end{aligned} \tag{C.3}$$

[1] Because we assume a known signal in deriving the bounds, the bounds are actually lower than necessary. In practice we are dealing with unknown signals; therefore the bounds are optimististic and no estimator should achieve the bound.

where the points (x_i, y_i, t_i) are spatio-temporal sampling instances of $r(x, y, t)$. In order to simplify the notation we define the observation vector

$$\bar{r} = (r_1, r_2, \ldots, r_N)^T \tag{C.4}$$

along with the signal and noise vectors

$$\bar{s} = (s_1, s_2, \ldots, s_N)^T \tag{C.5}$$

and

$$\bar{n} = (n_1, n_2, \ldots, n_N)^T \tag{C.6}$$

so that

$$\bar{r} = \bar{s} + \bar{n}. \tag{C.7}$$

The noise field is assumed to be zero-mean, white Gaussian noise with variance σ_n^2, so the probability density of the noise vector $\bar{n}$ is given by

$$p(\bar{n}) = \left(\frac{1}{\sqrt{2\pi}\sigma_n}\right)^N \exp\left(-\frac{1}{2\sigma_n^2}\sum_{i=1}^{N} n_i^2\right). \tag{C.8}$$

Substituting the observation Equation (C.7) into Equation (C.8) we arrive at an expression for the probability density function of the observation vector

$$p(\bar{r}) = \left(\frac{1}{\sqrt{2\pi}\sigma_n}\right)^N \exp\left(-\frac{1}{2\sigma_n^2}\sum_{i=1}^{N}(r_i - s_i)^2\right). \tag{C.9}$$

Define the log-likelihood function λ to be

$$\lambda = \log(p(\bar{r})), \tag{C.10}$$

and the Fisher information matrix to be

$$J = -E\begin{bmatrix} \dfrac{\partial^2\lambda}{\partial v_x^2} & \dfrac{\partial^2\lambda}{\partial v_x \partial v_y} \\ \dfrac{\partial^2\lambda}{\partial v_y \partial v_x} & \dfrac{\partial^2\lambda}{\partial v_y^2} \end{bmatrix} \tag{C.11}$$

where $E[\cdot]$ is the expectation operator. The bounds are expressed in terms of the Fisher information matrix are given by

$$Var[\hat{v}_x - v_x] \geq J_{11}^{-1} = \frac{J_{22}}{|J|} \tag{C.12}$$

and

$$Var[\hat{v}_y - v_y] \geq J_{22}^{-1} = \frac{J_{11}}{|J|}. \tag{C.13}$$

The elements of the Fisher information matrix are evaluated to be

$$J_{11} = E\left[\frac{\partial^2 \lambda}{\partial v_x^2}\right] = -\frac{1}{\sigma_n^2}\sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial v_x}\right)^2 \tag{C.14}$$

$$J_{12} = J_{21} = E\left[\frac{\partial^2 \lambda}{\partial v_x \partial v_y}\right] = -\frac{1}{\sigma_n^2}\sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial v_x}\right)\left(\frac{\partial s_i}{\partial v_y}\right) \tag{C.15}$$

$$J_{22} = E\left[\frac{\partial^2 \lambda}{\partial v_y^2}\right] = -\frac{1}{\sigma_n^2}\sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial v_y}\right)^2 \tag{C.16}$$

From the motion model given by Equation (C.2) we can evaluate the partial derivatives with respect to v_x and v_y to be

$$\frac{\partial s_i}{\partial v_x} = -\frac{\partial s_i}{\partial x_i}(t_i - t_0) \tag{C.17}$$

$$\frac{\partial s_i}{\partial v_y} = -\frac{\partial s_i}{\partial y_i}(t_i - t_0) \tag{C.18}$$

so the Fisher information matrix can be written as

$$J = \frac{1}{\sigma_n^2}\begin{bmatrix} \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)^2 (t_i - t_0)^2 & \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)\left(\frac{\partial s_i}{\partial y}\right)(t_i - t_0)^2 \\ \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)\left(\frac{\partial s_i}{\partial y}\right)(t_i - t_0)^2 & \sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial y}\right)^2 (t_i - t_0)^2 \end{bmatrix}. \tag{C.19}$$

Therefore the bounds are given by

$$Var[\hat{v}_x - v_x] \geq \frac{\sigma_n^2}{|J|}\sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial y}\right)^2 (t_i - t_0)^2 \tag{C.20}$$

and

$$Var[\hat{v}_y - v_y] \geq \frac{\sigma_n^2}{|J|}\sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)^2 (t_i - t_0)^2. \tag{C.21}$$

A special case of the bounds occurs when the signal is independent of either x or y. This occurs when there are well defined vertical or horizontal edges in a picture. In these cases the Fisher information matrix becomes singular, but the bound for the velocity component orthogonal to the edge remains finite. In these cases the bounds become

$$\frac{\partial s_i}{\partial y} = 0 \Longrightarrow Var[\hat{v}_x - v_x] \geq \frac{\sigma_n^2}{\sum_{i=1}^{N}\left(\frac{\partial s_i}{\partial x}\right)^2 (t_i - t_0)^2} \tag{C.22}$$

and

$$\frac{\partial s_i}{\partial x} = 0 \Longrightarrow Var[\hat{v}_y - v_y] \geq \frac{\sigma_n^2}{\sum_{i=1}^{N} \left(\frac{\partial s_i}{\partial y}\right)^2 (t_i - t_0)^2}. \tag{C.23}$$

We refer to this case as the one-dimensional motion estimation problem. More generally we can have edges oriented along arbitrary directions rather than along either the x or y axis. By rotating the coordinate system to obtain a new coordinate system (x', y'), the same result follows.

Appendix D

Convergence analysis of descent methods

Both the region matching and maximum likelihood estimators use descent algorithms to minimize an objective function. The descent algorithms attempt to minimize a nonlinear scalar-valued vector function. In formal terms we want to solve the following problem

$$\min_{\bar{x}} \{f(\bar{x})\} \tag{D.1}$$

where $f(\cdot)$ is a given nonlinear function of the vector $\bar{x}$. The approach which is often used to solve problems of this type is to use iterative descent methods that begin with an initial estimate $\bar{x}_0$ and generate a sequence $\{\bar{x}_k\}$ such that

$$f(\bar{x}_i) < f(\bar{x}_j) \quad \forall \quad i > j. \tag{D.2}$$

Successive vectors in the sequence $\{\bar{x}_k\}$ strictly decrease the objective function $f(\cdot)$, unless the sequence has converged to an element of the solution set. An iterative algorithm is said to converge if the sequence $\{\bar{x}_k\}$ approaches a limit point that is a local minimum of the objective function. In this appendix we discuss the conditions under which these iterative algorithms are guaranteed to converge.

Luenberger [20] discusses a very general convergence theory which is specifically related to this problem. Musicus [23] discusses similar results in the context of

parameter estimation problems. Our discussion is based on the presentation of Luenberger.

D.1 Global convergence theorem

Let us denote an arbitrary descent algorithm by $A(\cdot)$, so that successive members of the sequence $\{\bar{x}_k\}$ are generated as follows

$$\bar{x}_{k+1} \in A(\bar{x}_k). \tag{D.3}$$

In general the algorithm $A(\cdot)$ is a point-to-set mapping and $\bar{x}_{k+1}$ is a point in the set. The points $\bar{x}$ are members of a set X and there is a subset $\Gamma \subset X$ which is the solution set. For continuous objective functions the solution set is comprised of all the points which are a local minimum of the objective function. The central result of the global convergence theorem is as follows. If these three conditions are satisfied:

i) all points $\bar{x}_k$ are contained in a compact set $S \subset X$

ii) there exists a continuous descent function $z(\cdot)$ such that:
if $\bar{x} \in \Gamma$ then $z(\bar{y}) \leq z(\bar{x})$ for all $\bar{y} \in A(\bar{x})$
otherwise $z(\bar{y}) < z(\bar{x})$ for all $\bar{y} \in A(\bar{x})$

iii) the mapping $A(\cdot)$ is closed at points outside Γ

then the limit of any convergent subsequence of $\bar{x}_k$ is a solution.

A corollary to this theorem states that if the set Γ consists of a single point $\bar{x}^*$, then the sequence $\{\bar{x}_k\}$ converges to that point (this is the global minimum of the function). The proof of this theorem is given by Luenberger [20] on page 188.

The first condition requires that the sequence $\{\bar{x}_k\}$ lies within a compact set $S \subset X$. This implies that S is both closed and bounded. In many cases this condition is not a restriction on any particular algorithm, rather it is a condition under which a particular objective function $f(\cdot)$ will contain a bounded solution set.

The second condition requires that each step of the algorithm strictly decreases the descent function at all points which are not in the solution set.

The last condition restricts the algorithm $A(\ )$ to be closed at all points outside of Γ. An algorithm $A(\cdot)$ is closed at a point $\bar{x}$ if the conditions:

i) $\bar{x}_k \rightarrow \bar{x}, \bar{x}_k \in X$

ii) $\bar{y}_k \rightarrow \bar{y}, \bar{y}_k \in A(\bar{x}_k)$

imply $\bar{y} \in A(\bar{x})$. An equivalent condition for point-to-point mappings is that $A(\cdot)$ is continuous. If $A(\bar{x})$ is closed at each point $x \in X$, then it is said to be closed on X. In many algorithms this is the restrictive assumption which must be satisfied in order to guarantee convergence.

Several algorithms which we will analyze are composite mappings of the form $C = BA$, where $A : X \rightarrow Y$ and $B : Y \rightarrow Z$, so that $C : X \rightarrow Z$. Luenberger [20] (page 187) proves the following:

Composite mapping theorem. Let $A : X \rightarrow Y$ and $B : Y \rightarrow Z$ be point-to-set mappings. If A is closed at $\bar{x} \in X$, B is closed on $A(\bar{x})$, and Y is compact, then the composite map $C = BA$ is closed at $\bar{x}$. An important corollary states that if $A : X \rightarrow Y$ is a point-to-point mapping that is continuous at $\bar{x}$ and B is closed at $A(\bar{x})$, then the composite map $C = BA$ is closed at $\bar{x}$.

D.2 Convergence of iterative line search

Virtually all descent methods incorporate a line search procedure. We use an iterative quadratic curve fit procedure to locate the approximate value of α which minimizes the function $f(\bar{x} + \alpha\bar{d})$. It is straightforward to derive the necessary conditions for the algorithm to converge by invoking the global convergence theorem.

Let $g(\alpha) = f'(\alpha)$. The line search algorithm determines $\hat{\alpha}$ such that $g(\hat{\alpha}) \approx 0$ The iteration begins with a given initial interval $[\alpha_{min}, \alpha_{max}]$, where

$$0 \leq \alpha_{min} \leq \alpha_{max} \tag{D.4}$$

and

$$g(\alpha_{min}) \leq 0 \quad \text{and} \quad g(\alpha_{max}) \geq 0. \tag{D.5}$$

Define the vector $\bar{\alpha} = (\alpha_1, \alpha_2)^T$, where $\alpha_1 \leq \alpha_2$. From a given interval $[\alpha_1, \alpha_2]$, a new point $\hat{\alpha}$ is determined with a quadratic curve fit

$$\hat{\alpha} = \alpha_1 - g(\alpha_1)\left[\frac{(\alpha_1 - \alpha_2)}{g(\alpha_1) - g(\alpha_2)}\right]. \tag{D.6}$$

The point-to-point mapping $(\tilde{\alpha}_1, \tilde{\alpha}_2)^T = A(\bar{\alpha})$ is defined as:

i) if $g(\hat{\alpha}) = 0$ then $\tilde{\alpha}_1 = \tilde{\alpha}_2 = \hat{\alpha}$

ii) if $g(\hat{\alpha}) < 0$ then $\tilde{\alpha}_1 = \hat{\alpha}$ and $\tilde{\alpha}_2 = \alpha_2$

iii) if $g(\hat{\alpha}) > 0$ then $\tilde{\alpha}_1 = \alpha_1$ and $\tilde{\alpha}_2 = \hat{\alpha}$

By construction, $\hat{\alpha}$ is contained in the interval $[\alpha_1, \alpha_2]$. Consequently the sequence $\bar{\alpha}_k$ defined by this algorithm is guaranteed to lie in the compact set $\alpha_{min} \leq \alpha_1 \leq \alpha_{max}$ and $\alpha_{min} \leq \alpha_2 \leq \alpha_{max}$. Therefore condition i) of the global convergence theorem is satisfied.

A suitable descent function for this algorithm is

$$z(\bar{\alpha}) = |\bar{\alpha}|^2. \tag{D.7}$$

By construction, each iteration strictly decreases $z(\bar{\alpha})$ unless $\alpha_1 = \alpha_2 = \alpha^*$ and $g(\alpha^*) = 0$ (which is a point in the solution set Γ). Therefore condition ii) of the global convergence theorem is satisfied.

Finally, the algorithm defined above is continuous at all points, except when $\alpha_1 = \alpha_2 = \alpha^*$, $g(\alpha^*) = 0$, and $g(\alpha^*)/g'(\alpha^*)$ is unbounded. Therefore condition iii) of the global convergence theorem is satisfied.

Since all the conditions of the global convergence theorem are satisfied, the line search procedure is guaranteed to converge. However, we have only guaranteed that the algorithm converges to a stationary point of $f(\cdot)$. We still need to guarantee that it is a local minimum and not a local maximum. If $f(\cdot)$ is unimodal, there is only a single local minimum to which the algorithm is guaranteed to converge.

In general, if the initial interval $[\alpha_{min}, \alpha_{max}]$ contains at least one local maximum, under pathological conditions the algorithm can converge to such a point. To guard against this the higher order derivatives of $f(\cdot)$ are tested. If the point is a local maximum the entire procedure is repeated with a different initial condition.

D.3 Closure of line search

In actual practice the line search is terminated after some criterion is satisfied. Therefore in most cases the iteration does not reach a true limit point. The iteration is terminated when $\alpha_2 - \alpha_2 \leq \epsilon$ and $f(\bar{x} + \alpha\bar{d}) < f(\bar{x})$. In other words, the uncertainty as to the true value of α is less than ϵ and the line search decreases the objective function. One of the conditions required in order to guarantee convergence of the steepest descent algorithm is that the line search procedure is closed. Therefore in this section we prove that our line search procedure is closed.

The line search procedure is a point-to-set mapping $S : E^{2n+2} \rightarrow E^n$ defined as follows:

$$S(\bar{x}, \bar{d}, \alpha_{min}, \alpha_{max}) = \{\bar{y} : \bar{y} = \bar{x} + \alpha\bar{d}\} \tag{D.8}$$

where α satisfies the conditions

$$\alpha_{min} \leq \alpha \leq \alpha_{max}. \tag{D.9}$$

Theorem D.1 *The mapping defined by Equations D.8 and D.9 is closed at all* $(\bar{x}, \bar{d}, \alpha_{min}, \alpha_{max})$ *if* $\bar{d} \neq \bar{0}$.

Proof: Suppose $\{\bar{x}_k\}$, $\{\bar{d}_k\}$, $\{\alpha_{min}(k)\}$, $\{\alpha_{max}(k)\}$, and $\{\bar{y}_k\}$ are sequences such that $\bar{x}_k \rightarrow \bar{x}$, $\bar{d}_k \rightarrow \bar{d}$, $\alpha_{min}(k) \rightarrow \alpha_{min}$, $\alpha_{max}(k) \rightarrow \alpha_{max}$, $\bar{y}_k \rightarrow \bar{y}$, and $\bar{y}_k \in S(\bar{x}_k, \bar{d}_k, \alpha_{min}(k), \alpha_{max}(k))$. We want to show that $\bar{y} \in S(\bar{x}, \bar{d}, \alpha_{min}, \alpha_{max})$.

For each k we have $\bar{y}_k = \bar{x}_k + \alpha_k\bar{d}_k$ for some α_k. Therefore

$$\alpha_k = \frac{|\bar{y}_k - \bar{x}_k|}{|\bar{d}_k|}. \tag{D.10}$$

Taking the limit results in

$$\alpha_k \rightarrow \alpha^* \equiv \frac{|\bar{y} - \bar{x}|}{|\bar{d}|}. \tag{D.11}$$

Therefore $\bar{y} = \bar{x} + \alpha^* \bar{d}$. Now we must show that $\alpha_{min} \leq \alpha^* \leq \alpha_{max}$. By construction of the algorithm, α_k is obtained from a continuous function of $\alpha_{min}(k)$ and $\alpha_{max}(k)$ such that

$$\alpha_{min}(k) \leq \alpha_k \leq \alpha_{max}(k). \tag{D.12}$$

Taking the limits as $k \rightarrow \infty$ we get

$$\alpha_{min} \leq \alpha^* \leq \alpha_{max}. \qquad \square \tag{D.13}$$

D.4 Convergence of steepest descent

The steepest descent algorithm is defined by the iteration

$$\bar{x}_{k+1} = \bar{x}_k - \alpha_k \nabla f(\bar{x}_k), \tag{D.14}$$

where α_k is a nonnegative scalar that is determined with the line search procedure. To insure that the line search procedure is well-defined for all descent directions, we will assume that $f(\bar{x}) \longrightarrow \infty$ as $|\bar{x}| \longrightarrow \infty$ and that there exists a radius R_{max} such that $f(\bar{x}) \leq f(\bar{x}^*)$ if $|\bar{x}| < R_{max}$ for some $\bar{x}^*$. This condition insures that if $|\bar{x}| \leq R_{max}$ then there exists an $\alpha = \alpha_{limit}$ such that $f'_\alpha(\bar{x} - \alpha_{limit} \nabla f(\bar{x})) > 0$. This value of α is given by the positive root of

$$\alpha_{limit} = \frac{\bar{x}^T \nabla f(\bar{x})}{|\nabla f(\bar{x})|^2} \left(1 \pm \sqrt{1 - \frac{|(\bar{x}|^2 - R_{max})|\nabla f(\bar{x})|^2}{\bar{x}^T \nabla f(\bar{x})}} \right) \tag{D.15}$$

Each step of the steepest descent algorithm is a composite mapping $A = SG$. At each point $x \in X$, G is a continuous point-to-point mapping $G : E^n \rightarrow E^{2n+2}$ defined as follows

$$G(\bar{x}) = \{\bar{y} : \bar{y} = (\bar{x}, -\nabla f(\bar{x}), 0, \alpha_{limit})^T\}. \tag{D.16}$$

The mapping S is the line search procedure defined in the previous section. Since G is a continuous point-to-point mapping and S is closed, it follows from the corollary to the composite mapping theorem that the mapping A is closed. Therefore condition iii) of the global convergence theorem is satisfied.

By construction, each iteration strictly decreases the objective function $f(\cdot)$, unless $\nabla f(\bar{x}) = 0$. Therefore condition ii) of the global convergence theorem is satisfied.

Finally, for any point $\bar{x}_0$ such that $|\bar{x}_0| \leq R_{max}$ it follows that the sequence $\{\bar{x}_k\}$ lies within the compact set $X = \{\bar{x} : |\bar{x}| \leq R_{max}\}$. Therefore condition i) of the global convergence theorem is satisfied and the steepest descent algorithm is guaranteed to converge.

D.5 Convergence of region matching

In the region matching algorithm we minimize the objective function

$$\min_{\bar{v}} \left\{ f(\bar{v}) = \sum_{i=1}^{N} [r(\bar{x}_i - \bar{v}(t - t_0), t_0) - r(\bar{x}_i - \bar{v}(t - t_1), t_1)]^2 \right\} \tag{D.17}$$

In evaluating the objective function at arbitrary $\bar{v}$ it is necessary to compute the values of $r(\bar{x}, t)$ at points that are not on the sampling grid. These values are computed with a spatial interpolator. We have used a bilinear interpolator which has the property that $f(\bar{v})$ is continuous but the first-order partial derivatives are not. In fact, any interpolator which uses a finite support window will have the property that the first-order partial derivatives of the objective function with respect to $\bar{v}$ are not continuous.

The steepest descent algorithm uses the negative gradient as the line search direction. Therefore the algorithm is not defined when the gradient is evaluated at a point of discontinuity. Based on this fact we cannot guarantee convergence of the region matching algorithm. Nevertheless, we found that in practice the algorithm converges properly when the signal-to-noise levels are high, but often diverges at low signal-to-noise levels.

D.6 Convergence of maximum likelihood

The maximum likelihood algorithm involves determining the parameter values $(\bar{S}, \bar{v})$ that minimize the distance function $\lambda(\bar{S}, \bar{v})$, where $\bar{S}$ is an element of E^p and

$\bar{v}$ is an element of E^2. Because of the special structure of the problem, the minimization is carried out in two steps. Given an estimate $(\bar{S}_k, \bar{v}_k)$, another estimate $(\bar{S}_{k+1}, \bar{v}_{k+1})$ is generated as follows:

Step 1:

$$\min_{\bar{S}} \left\{\lambda(\bar{S}, \bar{v}_k)\right\} \Longrightarrow \bar{S}_{k+1} \tag{D.18}$$

Step 2:

Let $\Phi(\bar{S}, \bar{v})$ be the set of points such that $\bar{v}^* \in \Phi(\bar{S}, \bar{v})$ iff $\lambda(\bar{S}, \bar{v}^*) \leq \lambda(\bar{S}, \bar{v})$. More specifically, if the gradient of $\lambda(\bar{S}, \bar{v})$ with respect to $\bar{v}$ is a nonzero vector, then we require this to be a strict inequality. The vector $\bar{v}_{k+1}$ is taken to be a point in Φ as determined by the steepest descent algorithm.

The overall algorithm is a composite mapping $C : E^{p+2} \rightarrow E^{p+2} = AB$, where A is a point-to-point mapping and B is a point-to-set mapping. These mappings can be defined formally as follows.

$A : E^{p+2} \rightarrow E^{p+2}$ where

$$A(\bar{S}, \bar{v}) = \{(\bar{S}^*, \bar{v}) : \lambda(\bar{S}^*, \bar{v}) \leq \lambda(\bar{S}, \bar{v}) \ \ \forall \ \ \bar{S}\} \tag{D.19}$$

$B : E^{p+2} \rightarrow E^{p+2}$ where

$$B(\bar{S}, \bar{v}) = \{(\bar{S}, \bar{v}^*) : \bar{v}^* \in \Phi(\bar{S}, \bar{v})\} \tag{D.20}$$

By construction, the overall mapping C is guaranteed to decrease the descent function $\lambda(\cdot)$ unless a local minimum has been reached. Therefore condition ii) of the global convergence theorem is satisfied.

The mapping A is a continuous point-to-point mapping because it is obtained by solving a set of linear equations. We now prove:

Theorem D.2 *If $\lambda(\bar{S}, \bar{v})$ is continuous, then the mapping B defined by Equation D.20 is closed on all $A(\bar{S}, \bar{v})$.*

Proof: Let $\bar{x}, \bar{y} \in E^{p+2}$ such that $\bar{y} = B(\bar{x})$. Suppose $\{\bar{x}_k\}$ and $\{\bar{y}_k\}$ are sequences such that $\bar{x}_k \rightarrow \bar{x}$, $\bar{y}_k \in B(\bar{x})$, and $\bar{y}_k \rightarrow \bar{y}$. We want to show that $\bar{y} \in B(\bar{x})$.

For each k we have

$$\lambda(\bar{y}_k) \leq \lambda(\bar{x}_k). \tag{D.21}$$

Because $\lambda(\bar{S}, \bar{v})$ is continuous for all $\bar{x}$, taking the limit as $k \dashrightarrow \infty$ gives

$$\lambda(\bar{y}) \leq \lambda(\bar{x}) \quad \Longrightarrow \quad \bar{y} \in B(\bar{x}). \quad \square \tag{D.22}$$

According to the corollary to the composite mapping theorem, it follows that A is closed at all $(\bar{S}, \bar{v})$ and therefore condition iii) of the global convergence theorem is satisfied.

From the global convergence theorem it follows that if the sequence $(\bar{S}_k, \bar{v}_k)$ lies within a compact set, then the algorithm converges to a limit point which is a local minimum of the distance function. For the choice of model basis functions used in the estimator, it turns out that the distance function is a multivariate polynomial in $(\bar{S}, \bar{v})$. Furthermore, it is constrained to be nonnegative. Therefore as $(\bar{S}, \bar{v})$ tends toward infinity along any direction, one of two things must happen; either $\lambda(\bar{S}, \bar{v})$ remains constant or it also tends towards infinity. In the former case the extrema are multidimensional surfaces, while in the latter case the extrema are discrete points. In both of these cases we can construct a compact set composed of all points where $|\bar{S}, \bar{v}| < R_{max}$ such that all members of the sequence $(\bar{S}_k, \bar{v}_k)$ remain within the compact set. Therefore convergence can be guaranteed.

Bibliography

[1] G. Leigh Anderson and Arun N. Netravali. Image restoration based on a subjective criterion. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-6:pp. 845–853, December 1976.

[2] H. C. Andrews and B. R. Hunt. *Digital Image Restoration.* Prentice Hall, Englewood Cliffs, New Jersey, 1977.

[3] Ted J. Broida and Rama Chellapa. Estimation of object motion parameters from noisy images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8(No. 1):pp. 90–99, January 1986.

[4] Ciro Cafforio and Fabio Rocca. Methods for measuring small displacements of television images. *IEEE Trans. on Information Theory*, Vol. IT-22:pp. 573–579, September 1976.

[5] P. Chan and J. S. Lim. One-dimensional processing for adaptive image restoration. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33:pp. 117–125, February 1985.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, Vol. 39(No. 1):pp. 1–22, 1977. Series B (Methodological).

[7] T. J. Dennis. Nonlinear temporal filter for television picture noise reduction. *IEE Proc.*, Vol. 127, Pt. G(No. 2):pp. 52–56, April 1980.

[8] E. Dubois and S. Sabri. Noise reduction in image sequences using motion-compensated temporal filtering. *IEEE Trans. on Communications*, Vol. COM-32(No. 7):826–831, July 1983.

[9] B. R. Frieden. Image enhancement and restoration. In *Picture Processing and Digital Filtering*, pages pp. 177–248, 1979.

[10] Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing.* Addison Wesley, Reading, Massachusetts, 1977.

[11] B. G. Haskell. Frame-to-frame coding of television pictures using two-dimensional fourier transforms. *IEEE Trans. on Information Theory*, IT-20(No. 1):pp. 119–120, January 1974.

[12] Brian Lee Hinman. *Theory and Applications of Image Motion Estimation.* Master's thesis, Massachusetts Institute of Technology, 1984.

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

[14] T. S. Huang and Y. P. Hsu. *Image Sequence Enhancement*, chapter 4, pages pp. 289–309. Springer-Verlag, 1981.

[15] T. S. Huang and R. Y. Tsai. *Image Sequence Analysis: Motion Estimation*, chapter 1, pages pp. 1–18. Springer-Verlag, 1981.

[16] Edward A. Krause. *Motion Estimation and Interpolation in Time-Varying Imagery.* Master's thesis, Massachusetts Institute of Technology, 1984.

[17] J. S. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2(No. 2):pp. 165-168, March 1980.

[18] George R. Legters Jr. and Tzay Y. Young. A mathematical model for computer image tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-4(No. 6):pp. 583–594, November 1982.

[19] J. O. Limb and J. A. Murphy. Measuring the speed of moving objects from television signals. *IEEE Trans. on Communications*, Vol. COM-23(No. 4):pp. 474–478, April 1975.

[20] David G. Luenberger. *Linear and Nonlinear Programming.* Addison Wesley, Reading, Massachusetts, 1984.

[21] Dennis Martinez and Jae S. Lim. Implicit motion-compensated noise reduction of motion video scenes. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages pp. 375–378, March 1985.

[22] R. H. McMann, S. Kreinik, J. K. Moore, A. Kaiser, and J. Rossi. A digital noise reducer for encoded ntsc signals. *SMPTE Journal*, March:134–140, March 1978.

[23] Bruce R. Musicus. *Iterative Algorithms for Optimal Signal Reconstruction and Parameter Identification Given Noisy and Incomplete Data.* PhD thesis, Massachusetts Institute of Technology, 1982.

[24] A. N. Netravali and J. D. Robbins. Motion-compensated coding: some new results. *The Bell System Technical Journal*, Vol. 59:pp. 1735–1745, November 1980.

[25] A. N. Netravali and J. D. Robbins. Motion-compensated television coding: part 1. *The Bell System Technical Journal*, Vol. 58:pp. 631–670, March 1979.

[26] A. N. Netravali and J. D. Robbins. Video signal interpolation using motion estimation. U. S. Patent No. 4,383,272, May 10 1983.

[27] Y. Ninomiya and Y. Ohtsuka. A motion-compensated interframe coding scheme for television pictures. *IEEE Trans. on Communications*, Vol. COM-30(No. 1):pp. 201–211, January 1982.

[28] R. Paquin and E. Dubois. A spatio-temporal gradient method for estimating the displacement field in time-varying imagery. *Computer Vision, Graphics, and Image Processing*, Vol. 21:pp. 205–221, 1983.

[29] John W. Roach and J. K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2(No. 6):pp. 554–562, November 1980.

[30] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing.* Academic Press, New York, New York, 1976.

[31] Roger Samy. An adaptive image sequence filtering scheme based on motion detection. In *Second International Technical Symposium on Optical and Electro-Optical Applied Science and Engineering*, December 1985.

[32] R. Srinivasan and K. R. Rao. Predictive coding based on efficient motion estimation. *IEEE Trans. on Communications*, Vol. COM-33(No. 8):888–896, August 1985.

[33] J. A. Stuller and A. N. Netravali. Transform domain motion estimation. *The Bell System Technical Journal*, Vol 58:1673–1702, September 1979.

[34] R. Y. Tsai and T. S. Huang. Estimating three dimensional motion parameters of a rigid planar patch. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29(No. 6):pp. 1147–1152, December 1981.

[35] R. Y. Tsai, T. S. Huang, and W. L. Zhu. Estimating three dimensional motion parameters of a rigid planar patch, ii: singular value decomposition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30(No. 4):pp. 525–534, August 1982.

[36] H. L. Van Trees. *Detection, Estimation, and Modulation Theory.* John Wiley and Sons, 1968.