October 1991

LIDS-TH-2073

Research Supported By:

National Science Foundation
Grant NSF 9015281-MIP
and
Grant NSF INT-9002393

Office of Naval Research
Grant N00014-91-J-1004

U.S. Army Research Office
Contract DAAL03-86-K-0171

Dynamic Estimation in Computational Vision

Toshio Michael Chin

October 1991 LIDS-TH-2073

DYNAMIC ESTIMATION IN COMPUTATIONAL VISION

by

Toshio Michael Chin

This report is based on the unaltered thesis of Toshio Michael Chin submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Massachusetts Institute of Technology in February 1992. This research was carried out at the M.I.T. Laboratory for Information and Decision Systems and was supported in part by the National Science Foundation under grants 9015281-MIP and INT-9002393, the Office of Naval Research under grant N00014-91-J-1004, and the U.S. Army Research Office under contract DAAL03-86-K-0171.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

Dynamic Estimation in Computational Vision

by

Toshio Michael Chin

B.S., University of California at Berkeley (1982)
M.S., University of California at Berkeley (1983)
E.E., Massachusetts Institute of Technology (1990)

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

Doctor of Philosophy

at the

Massachusetts Institute of Technology

February 1992

Author .....
Department of Electrical Engineering and Computer Science
October 22, 1991

Certified by .....
Alan S. Willsky
Professor, Electrical Engineering
Thesis Supervisor

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

Dynamic Estimation in Computational Vision

by

Toshio Michael Chin

Submitted to the Department of Electrical Engineering and Computer Science
on October 22, 1991, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Spatial coherence constraints are commonly used to regularize the problems of reconstructing dense visual fields like depth, shape, and motion. Recent developments in theory and practice show that the local nature of spatial coherence constraints allows us to solve single-frame reconstruction problems efficiently with, for example, multiresolution approaches. While it is reasonable to impose *temporal* as well as spatial coherence on the unknown for a more robust estimation through data fusion over both space and time, such temporal, multi-frame extensions of the problems have not been as widely considered, perhaps due to the different and severe computational demands imposed by the sequential arrival of the image data. We present here an efficient filtering algorithm for sequential estimation of dense visual fields, using stochastic descriptor dynamic system models to capture temporal smoothness and dynamics of the fields.

Theoretically, standard Kalman filtering techniques (generalized for stochastic descriptor systems) are applicable to solving temporally-extended visual field reconstruction problems, but their implementation is practically impossible because of the high dimensionality and because the time-varying nature of such problems requires on-line propagation of large covariance matrices. By exploiting the inherent local spatial structure of the reconstruction problem, however, we have developed filtering techniques that effectively approximate the information form of the Kalman filter. This is achieved by replacing covariance propagation steps with what are essentially low-order spatial model identification steps, in which spatial models with a strictly local support are constructed based on covariance information. In effect, we are decomposing the multi-frame problem into a series of Bayesian single-frame problems, in which the spatial prior model used reflects knowledge from the previous image frames. The resulting filtering algorithm has memory and computational requirements of $O(N)$ each for a frame of data, where N is the number of pixels in a frame, and, additionally, the filter is implementable in parallel.

As low-level visual field reconstruction is often considered to be a front-end in a hierachical visual processing system and thus might be VLSI-implemented, we have also designed a square root version of the information Kalman filter as an alternative algorithm with a reduced numerical dynamic range. The square root information

filter features an efficient, iterative computational structure and is parallelizable as well.

Experiments have shown several beneficial effects of our multi-frame formulation applied to the sequential estimation of optical flow. For example, temporal assimilation of the data makes the reconstruction more robust to noise. Also, there are cases where the classic "aperture problem" of motion vision cannot be resolved satisfactorily by spatial regularization alone but is dramatically resolved by the additional temporal coherence constraint.

Thesis Supervisor: Alan S. Willsky
Title: Professor, Electrical Engineering

Acknowledgement

I wish to thank my thesis supervisor Professor Alan Willsky for his insights and ideas as well as the time he has spent with me on this work. I cannot over-emphasize that his enthusiasm and encouragement are crucial components of the thesis. I am also grateful to have Professors Berthold Horn and George Verghese as thesis committee members. Their expertise has been invaluable, and association with them has been a pleasure. Special thanks are also due to Professor Jae Lim, Dr. W. Clem Karl, Hiroshi Ando, and Eero Simoncelli for their advices and assistances that have directly influenced the outcome of the thesis.

The faculty, staff, and fellow students, past and present, of the Laboratory for Information and Decision Systems have had so much to offer me both academically and culturally. They have made my stay at M.I.T. stimulating, comfortable, and fun. The same applies to the friends in the Japanese Association of M.I.T. My mind escaped completely from the studies every time we organized "events" — usually large parties. My life in the Northeast so far cannot be described without orienteering, and I thank all orienteers (especially the field checkers, of course) for the good time in the woods. It has been a special experience to nurture our little orienteering club on this campus with Bent, Cecilie, Detlef, Grant, Mike, Sue, and others. May this so-called "thinking sport" continue to grow here!

Finally, I would like to express my deep gratitude towards my parents Fukuhatsu and Yumi Chin for their love, support, and continuing patience.

This work was supported in parts by the National Science Foundation under Grants 9015281-MIP and INT-9002393, the Office of Naval Research under Grant N00014-91-J-1004, and the US Army Research Office under Contract DAAL0386-K-0171.

Contents

Chapter 1

Introduction

Computational vision is a field of study that explores the design and analysis of systems that process images of natural scene to produce a simple, stable, and useful set of descriptions. The field is interdisciplinary in nature but is especially active in robotics where design of artificial systems able to make decisions autonomously based on visual inputs is of a fundamental importance and in psychophysics where computational theory guides researchers to formulate functional and structural models for the components of complex nervous systems. There is an enormous opportunity for theories of signal processing to contribute to computational vision. This thesis is concerned with applying stochastic system and estimation theories to a problem of general interest in computaional vision — reconstruction of so-called low-level visual fields. A major focus is placed on attaining implementational efficiency in processing sequences of images obtained as large, two-dimensional (2-D) arrays of brightness[1] values measured by an eye or a camera.

1.1 Low-level visual field reconstruction

A computational vision problem is often decomposed into subproblems. By far, the most popular approach is to characterize the computational task by a "bottom-up" hierarchical system of mostly independent processing modules [33, 52]. In the early

[1] i.e., light intensity

or low-level stages, structural and functional properties such as the depth, motion, and shape[2] of the imaged surfaces are inferred from the measurements while in later (higher-level) stages such rudimentary inferences are organized to enable the system to recognize and characterize various aspects of the objects to which the surfaces belong, leading to "understanding" of the visual environment. Low-level visual field reconstruction is an image processing problem in the very early stage of this hierarchical system. It aims to recreate a *dense* scalar or vector field of a particular feature of the visible surfaces.

Reconstruction of dense visual fields can be regarded as a problem in inverse optics, since physical properties of 3-D surfaces are to be recovered from the 2-D images that the surfaces generate [31, 73]. The inverse problems arising in visual field reconstruction are generally ill-posed [7], as the measurements from images do not constrain the desired solution sufficiently. An established approach to making such problems solvable is to impose an additional constraint on the solution by assuming that the field to be reconstructed varies "smoothly" over space [7, 33, 71]. A physical basis for such an assumption, commonly known as the *smoothness* or *spatial coherence* constraint [71], is that most surfaces in natural scenes exhibit some geometrical smoothness and structural integrity ("rigidity") on the whole. This approach has been successful in a wide variety of visual field reconstruction problems including depth interpolation [21, 22], shape from shading [32, 36, 37], and optical flow computation [34, 30]. Mathmatically, these reconstruction problems are formulated as least-squares problems in which the spatial coherence constraints are implemented as cost terms penalizing large spatial derivatives in the fields. Solution of such least-squares problems generally involves inversion of a partial differential operator over a spatial domain (i.e., the image frame), to which efficient numerical methods such as multigrid methods are applicable [72].

[2] which usually refers to a dense field of unit vectors perpendicular to the surface

1.2 Multi-frame visual field reconstruction

Much of the research effort on visual field reconstruction to this date has been focused mostly on obtaining a single frame of estimates; only recently have there been attempts to reconstruct multiple frames of estimates for time-varying visual fields. In multi-frame visual field reconstruction, the measurements are assimilated temporally as well as spatially to compute each frame of estimates. Thus, a multi-frame approach can be expected to provide us with more robost and accurate estimates than the corresponding single-frame approach applied repeatedly to the individual frames in an image sequence. The relatively slow progress in proliferation of multi-frame reconstruction methods can be attributed to their apparent computational complexity. Computational demands for multi-frame algorithms are much greater than their single-frame counterparts, as the measurements are distributed over a higher dimensional (space-time) domain.

Our main goal in this thesis is to develop a general, efficient computational framework for multi-frame visual reconstruction problems. We also demonstrate the efficiency and robostness of the resulting algorithms, as well as the improved accuracy in the estimates computed with them, by applying the framework to computation of several time-varying visual fields including optical flow.

1.3 Probabilistic framework

We formulate the multi-frame reconstruction problems by applying stochastic estimation theory. An estimation theoretic interpretation of a single-frame visual reconstruction problem is possible by treating the spatial coherence constraint as the prior model in a Bayesian estimation problem [17, 65, 71]. Similarly, a multi-frame problem can be considered as an estimation problem by modeling the time-varying visual field as a stochastic process, for which a Kalman filter can be used to compute the estimates sequentially in time. Although the sequential computational environment provided by Kalman filter is certainly desirable, the large dimension of the state vari-

able (typically on the order of the number of pixels in the image frame) makes the filter computationally unfeasible in its optimal form. Kalman filtering solutions for multi-frame visual reconstruction problem have been proposed in a variety of contexts [55, 71, 27, 10, 69]; however, in each of these cases some *ad hoc* simplifying assumptions are made in order to avoid the computational complexity issues. For example, in none of these previously proposed methods, is the uncertainty in the dynamic model for the time-varying visual field properly handled. Normally, in an exact implementation of a Kalman filter such uncertainty, which is expressed via the incorporation of "process noise", can serve the purpose of making the filter stable against modeling errors and responsive to new data [4, 16, 45]. Certainly, a filtering algorithm based on a more systematic approximation of Kalman filter is desirable. This thesis offers such algorithms. Specifically, we first formulate a general Kalman filtering solution for multi-frame visual reconstrucition problems and then design various approximation techniques that yield near-optimal and highly efficient solutions.

1.4 MRF-based approximation of Kalman filter

The approximate Kalman filtering techniques presented in this thesis are based on the conjecture that faithful statistical specifications of visual fields of interest are possible using strictly local spatial models. Specifically, to estimate a time-varying visual field sequentially, we propagate a Markov Random Field (MRF) model of the field over time and update the model using the measurements from the most recent image at each time step. MRF models are used extensively in single-frame visual reconstruction problems, e.g., [17, 11, 71, 29]; they are quite versatile modeling tools especially suitable for the important problem of estimating the locations of discontinuity in visual fields [17, 11]. Used in reconstruction of entirely smooth fields, an MRF-based formulation yields spatially local specifications for the estimated field, and explicit estimates can be computed from these specifications using efficient numerical techniques such as multi-grid methods [73, 71] as mentioned previously. That is, the locality of the specifications enables efficient computation of the estimates.

Essentially, the design of approximation techniques for the Kalman filter in multi-frame visual reconstruction amounts to the development of algorithms that efficiently manages an MRF model for the time-varying visual field by updating the model based on the temporal dynamics and image measurements. The algorithms are approximations of a Kalman filter because, as detailed in the thesis, the exact Kalman filtering algorithm in general yields a non-local description of the visual field. We will demonstrate that these approximate Kalman filters perform nearly the same as the optimal filter, supporting our view that the visual fields of interest are indeed faithfully described locally in space, thus enabling us to construct efficient space-time processing algorithms for multi-frame reconstruction.

1.5 Contributions

Contributions of this thesis are summarized as follows:

To visual field reconstruction

A probabilistic framework based on maximum likelihood estimation is devised for visual reconstruction problems, so that spatially and temporally distributed data can be fused efficiently for statistically optimal estimation of visual fields. Specifically, a general formulation for multi-frame visual reconstruction problems is obtained using a descriptor dynamic system model, and an information Kalman filter is derived based on this dynamic model, providing a sequential solution to multi-frame problems. Although the computational complexity of this Kalman filter is too high for a practical implementation, the filter can be approximated effectively by various numerical techniques based on the hypothesis that at every stage of estimation the (estimated) visual fields can be accurately modeled using locally-specified MRF's.

To filtering theory

Some aspects of the proposed filtering algorithms for visual reconstruction are related to several recent developments in signal processing. First, in [62], the use of the max-

imum likelihood estimation formalism is shown to be equally applicable to descriptor dynamic systems, yielding a generalization of standard Kalman filter algorithms. Using this formalism, an approach to derive an *information Kalman filter* for a descriptor system is developed in this thesis. Also, the Markovianity condition for a Gaussian random field, expressed in terms of the elements of the associated information matrix [44], is used to relate MRF modeling and the spatially local approximations for the information Kalman filter algorithms developed in this thesis.

The approximate Kalman filtering techniques should be applicable to general large-scale, multi-dimensional filtering problems. Specifically, the approximation techniques can be considered to be extensions of the techniques based on reduced-order modeling such as the "reduced update" and "strip filter" approximations for the 2-D Kalman filter [78, 57, 79, 76]. Also, a novel, iterative implementation of square root information Kalman filter is presented.

To optical flow computation

Optical flow is a dense field of perceptual motion vectors. Estimation of the motion field is particularly important in multi-frame visual reconstruction in developing a temporal dynamic model for any visual field, as the motion field establishes inter-frame correspondence of points in a dynamic field. Many visual processing schemes, including dynamic "structure-from-motion" (e.g., [27, 5]), use optical flow as their input. The proposed filter is used to estimate time-varying optical flow to demonstrate that a particular multi-frame formulation, essentially a simple temporal extension of the gradient-based approach due to Horn and Schunck [34], can lead to estimates more robust and accurate than traditional formulations, with essentially no increase in computational complexity.

1.6 Overview of the thesis

In Chapter 2, we review optical flow computation methods. The chapter serves as both an introduction to visual reconstruction by providing a case study of the

single-frame problems and a background for the multi-frame optical flow formulation presented later in the thesis. Then, a general, estimation theoretic formulation for the single-frame reconstruction problems is presented in Chapter 3. We introduce a maximum likelihood estimation framework in which the pieces of statistical data necessary for reconstruction can be combined efficiently. Extending the framework in time, a general formulation for the multi-frame reconstruction problems is developed in Chapter 4. This multi-frame formulation constitutes a main conceptual contribution of the thesis. The solutions for the multi-frame problems can be computed sequentially using the optimal Kalman filtering algorithms, including information and square root information filters, derived in the chapter. These Kalman filters carry state variables with large dimensions and hence raise computational complexity issues in implementation. These issues are addressed in Chapter 5, where various approximation techniques for the Kalman filter algorithms are proposed and verified. We approximate the information matrices and their square root versions as strictly local specification for MRF's to reduce computational complexity. Further, we propose methods to compute such MRF specifications efficiently by presenting a series expansion technique for matrix inversion in the information filter as well as an iterative technique for unitary factorization in the square root information filter. The numerical experiments presented in this chapter show that the approximated filters perform near-optimally, demonstrating the effectiveness of these efficient filtering algorithms for multi-frame visual reconstruction. Also, the numerical techniques featured in these algorithms offer some novel, iterative approaches to general large-scale filtering problems. In Chapter 6, we come back to the optical flow problem, as we examine performance of the filtering algorithms in multi-frame optical flow computation. We conclude the thesis in Chapter 7 with a summary and remarks on possible extensions of the presented work.

Chapter 2

Computing Motion

The relative motion between objects in a scene and a viewer, such as a camera with which to record images, gives rise to the apparent motion of objects in an image sequence. Computation of such perceptual motion from image sequences is of considerable interest in the study of vision — both for the development of artificial visual systems in robotics and for the modeling of biological visual systems. For example, the computed apparent motion in an image sequence can provide us with information necessary to detect object boundaries [56, 29] and to derive 3-D motion and structure of the objects in the image frame [48, 59]. Computing motion from image sequences is also important in applications in fields outside of robotics and brain sciences, such as in assessing motility of the heart [13, 64] and in interpretation and prediction of marine and atomospheric processes [51, 12]. Motion information is also useful for managing the image sequences themselves as it offers a basis for image sequence compression for efficient transmission and storage [60, 61].

There are two major approaches in computing apparent motion in image sequences. One of these is based on extracting a set of relatively sparse, yet highly discriminatory features from each image frame in the sequence. These features are matched from frame to frame, and the displacements of the locations of the features yield a set of sparse but relatively reliable 2-D motion vectors. Computational complexity involved with this method, especially for the inter-frame matching or the *correspondence problem*, is in general quite high [1, 2]. The other method is based

directly on the measured pixel values in the images, and it generally yields a dense vector field of apparent motion, or *optical flow*[1]. Efficient computation of optical flow serves as a major motivation for designing the algorithms for visual field reconstruction as described in the rest of this thesis. In this chapter, we review and discuss a well-known approach to the computation of a single frame of optical flow. A new method to sequentially compute time-varying optical flow will be described in Chapter 6.

2.1 Computing Optical Flow

Let us define $f(\mathbf{s})$ to be the 2-D optical flow vector where $\mathbf{s}$ is the indices for the space over which the optical flow is defined, i.e., $\mathbf{s} = [s_1, s_2]^T$ is the coordinate vector for the 2-D image frame. Let us also define $I(\mathbf{s}, t)$ to be the image brightness value or gray level at $\mathbf{s}$ where t is the temporal index specifying a frame in the image sequence. The problem of optical flow computation is to deduce $f(\mathbf{s})$ from $I(\mathbf{s}, t)$. We assume for the time being that $f(\mathbf{s})$ and $I(\mathbf{s}, t)$ are continuous functions of their respecitve indexing variables.

Based on a straightforward intuition, we can formulate an optical flow computation problem based on matching brightness values in successive frames of images. Specifically, let $I_0(\mathbf{s}) = I(\mathbf{s}, t)$ and $I_1(\mathbf{s}) = I(\mathbf{s}, t + \Delta t)$. Then, assuming that $f(\mathbf{s})$ is constant over a certain subregion in the image frame and that object brightness does not change over the Δt time interval, the following criterion allows us to compute an optical flow field [60, 3]:

$$\min_f |I_1(\mathbf{s} + (\Delta t) f) - I_0(\mathbf{s})| .$$

Obtaining this minimum often involves a non-linear search. In another method based on the same intuition, a flow field is computed from the difference in the phases of Fourier transforms of $I_0(\mathbf{s})$ and $I_1(\mathbf{s})$ [24]. The assumption that the optical flow is

[1] also called *image flow*

uniform over an image region is an obvious weakness of these approaches, as the flow vectors in general vary from one pixel location to anothor. By characterizing the relationship between the measured brightness values and the flow vectors at the resolution of a single pixel, however, this assumption can be relaxed. This can be achieved by characterizing the flow vectors using differential variations of the image brightness, as described in the pioneering work by Horn and Schunck [34]. For the rest of this chapter, we focus on optical flow computaion based on such an approach.

2.2 Brightness Change Constraint Equation

The *brightness change constraint equation* relates the optical flow vector $f(\mathbf{s})$ with the spatial and temporal gradients of the image brightness $I(\mathbf{s}, t)$. The principle behind this equation is that the brightness value associated with a given physical point (i.e., a point on the surface of an imaged object) in motion can be considered to be invariant for a sufficiently long duration, so that the position of the point can be tracked by referencing the same brightness value. Mathematically, such invariance of the brightness value can be expressed as

$$\frac{d}{dt} I(\mathbf{s}, t) = 0 \tag{2.1}$$

where $\mathbf{s}$ is the coordinates of the image of the moving point. Note that the coordinates associated with a physical point, $\mathbf{s}$, vary with time in general. By expanding the left hand side of the equation in terms of the partial derivatives, we have the *brightness change constraint equation* (BCCE):

$$\frac{\partial I}{\partial t} + \frac{\partial I}{\partial s_1}\frac{ds_1}{dt} + \frac{\partial I}{\partial s_2}\frac{ds_2}{dt} = 0. \tag{2.2}$$

2.2.1 Notation

In this thesis, we use matrix notation for the BCCE:

$$g(\mathbf{s}) = h(\mathbf{s})f(\mathbf{s}) \tag{2.3}$$

where we define:

- *the optical flow vector*

$$f(\mathbf{s}) \equiv \begin{bmatrix} f_1(\mathbf{s}) \\ f_2(\mathbf{s}) \end{bmatrix} \equiv \frac{d\mathbf{s}}{dt} = \begin{bmatrix} \frac{ds_1}{dt} \\ \frac{ds_2}{dt} \end{bmatrix} \tag{2.4}$$

- *the negative of the temporal gradient*

$$g(\mathbf{s}) \equiv -\frac{\partial I}{\partial t} \tag{2.5}$$

- *the spatial gradient*

$$h(\mathbf{s}) \equiv \frac{\partial I}{\partial \mathbf{s}} = \left[\frac{\partial I}{\partial s_1}, \frac{\partial I}{\partial s_2}\right] \tag{2.6}$$

2.2.2 The Aperture Problem

The brightness change constraint equation (2.3) gives only one constraint for the two components of the flow vector. This is the essence of a phenomenon called the *aperture problem* which states that unique motion cannot be detected by local measurements of the changes in the brightness values alone [34]. Specifically, consider representing the optical flow vector $f(\mathbf{s})$ in a local coordinate system where one of the two axes points along the spatial gradient vector $h(\mathbf{s})$. The component of $f(\mathbf{s})$ along this spatial gradient axis, denoted as $f^{\perp}(\mathbf{s})$, is given by

$$f^{\perp}(\mathbf{s}) \equiv \frac{h(\mathbf{s})}{\|h(\mathbf{s})\|} f(\mathbf{s}). \tag{2.7}$$

Then, by dividing both sides of (2.3) by $\|h(\mathbf{s})\|$ we obtain

$$f^{\perp}(\mathbf{s}) = \frac{g(\mathbf{s})}{\|h(\mathbf{s})\|}. \tag{2.8}$$

Thus, the brightness change constraint equation yields only the component of the flow vector along the spatial gradient vector.

Traditionally, the aperture problem is associated with perceptability of motion of a translating straight edge. Typically, $f^{\perp}(\mathbf{s})$ is the only *locally* observable quantity when determining movements of edges in image sequences [30], i.e., $f^{\perp}(\mathbf{s})$ is the apparent inter-frame displacement of a point on a straight edge.

2.2.3 Other implications of BCCE

1. Motion cannot be perceived from a sequence of completely feature-less images. This is reflected mathematically in the brightness change constraint equation (2.3) in such a way that when the spatial gradient $h(\mathbf{s})$ is zero the equation does not constrain the optical flow vector at all.

2. The fundamental assumption behind BCCE, (2.1), implying invariance of the brightness associated with a moving point, is not strictly correct for most natural image sequences. For example, illumination for a point on a moving surface can vary with time due to continuous change in the geometrical configuration between the surface and the lighting source. In practice, however, such variation in illumination occurs at a much slower time scale than movement of the point, so that the assumption (2.1) is reasonable.

3. Discrete implementation of BCCE requires several additional considerations. These will be discussed in Section 2.4.

2.3 Gradient-based Optical Flow Computation

2.3.1 Computing optical flow: a basic formulation

A well-known method for optical flow computation is the optimization approach introduced by Horn and Schunck [34]. In this formulation, the sum of two cost functionals of the unknown flow vectors is optimized. One of the cost terms penalizes flow vectors that deviate too much from satisfying the brightness change constraint equation (2.3) for a given set of brightness measurements. The other term, which is necessary because of the aperture problem associated with the brightness change constraint equation, forces the flow vectors to vary smoothly over space. Specifically, the "brightness constraint" cost term based on (2.3) is:

$$\|g(\mathbf{s}) - h(\mathbf{s})f(\mathbf{s})\|^2 \tag{2.9}$$

which underconstrains the flow vector because the vector has two components. As the necessary, additional constraint for the flow vector, Horn and Schunck have proposed to use the "smoothness constraint[2]" cost term

$$\left\| \frac{\partial}{\partial \mathbf{s}} f \right\|^2 \quad \left(\equiv \left\| \frac{\partial}{\partial s_1} f \right\|^2 + \left\| \frac{\partial}{\partial s_2} f \right\|^2 \right). \tag{2.10}$$

The optical flow computation problem is, then, to find $f(\mathbf{s})$ that minimize the sum of these two cost terms for a fixed t:

$$J = \int_{\mathcal{D}} \|g - hf\|^2 + \mu \left\| \frac{\partial}{\partial \mathbf{s}} f \right\|^2 d\mathbf{s} \tag{2.11}$$

where the domain of integration $\mathcal{D}$ is often the entire image frame and μ is a fixed optimization parameter.

Commonly, the optimization problem is solved by a two step process: (i)derivation of an implicit form of the solution via variational calculus and (ii)computing the

[2]Following terminolgy in computational vision, we use the word "constraint" to refer to a quadractic penality term in weighted least-squares problems.

explicit solution by an iterative numerical technique. In the next chapter, we present a maximum likelihood estimation approach to obtain an equivalent solution.

2.3.2 Spatial coherence constraint

The smoothness constraint (2.10) is a form of spatial coherence constraint, under which we essentially assume that the flow vectors within a local subregion of the image frame are *almost* identical. Such a constraint is reasonable for optical flow computation because in most practical cases motion in image sequences is generated by smooth surfaces undergoing rigid motion or deformation; thus, neighboring points on the surfaces tend to have similar velocities.

A spatial coherence constraint implies stiffness and smoothness of the imaged surfaces [30]. This is obviously violated along most object boundaries. More sophisticated versions of the constraint have been proposed to address the problems at object boundaries [58, 67] with limited success. Spatial coherence is essentially a local constraint and cannot capture global characteristics of boundaries in images.

Nevertheless, spatial coherence constraints are considered to be important mathematical components and hence are widely used in reconstruction of a variety of visual fields, including depth and shape as well as optical flow [7]. They tend to be expressed as first and/or second order differential cost terms, such as (2.10), in optimization formulations. An important mathematical characteristic of spatial coherence constraints is that they have spatially local support, i.e., derivatives are local properties of the unknown fields. This characteristic is a key feature of the low-level visual reconstruction problems in terms of their computational tractability, which will be discussed in detail in this thesis.

2.3.3 Optical flow along a contour

Many problems in vision require computing optical flow not over a 2-D region in the image frame but along a known contour line (edge). Hildreth [30], interested in quantifying motion of edges in image sequences, has formulated an optical flow computation

problem which is essentially a 1-D version of Horn and Schunck's formulation:

$$J_C = \int_C \left\| f^\perp - \frac{h}{\|h\|} f \right\|^2 + \mu \left\| \frac{\partial}{\partial s} f \right\|^2 ds \tag{2.12}$$

where C is a 1-D domain of integration representing a (possibly closed) contour and the 1-D spatial index s now denotes the location along this contour. The second term in the integral is the "observation constraint" relating the measurements made on the contour to the optical flow vectors. Thus, the apparent contour displacements $f^\perp$ over time and the unit normal vectors along the contour $\frac{h}{\|h\|}$ are the inputs to the optical flow computation algorithm instead of the brightness gradients[3]. We have discussed previously that the observation constraint in (2.12) is exactly the BCCE (2.3) normalized by $\|h(\mathbf{s})\|$. In the sequel we refer to the optical flow along a contour as the "1-D optical flow", while the optical flow over an image region as the "2-D optical flow".

2.4 Discretization Effects on Optical Flow Computation

In practice optical flow is computed over a discrete grid instead of a continuous domain. In this section we discuss the effects of discretizing the gradient-based flow computation method presented above. We also propose some practical techniques to reduce such discretization effects.

2.4.1 Signal bandwidths and optical flow

The brightness signal $I(\mathbf{s}, t)$ is typically obtained as a sequence of spatially discrete images. Let us first consider the effects of such signal sampling on motion perception.

By assuming that the optical flow is constant, f, over space and time, Martinez [54] has derived the following relationship between the spatial and temporal bandwidths

[3]although in practice brightness gradients are often used because $f^\perp$ can be difficult to measure

of the brightness signal and the magnitude of the optical flow:

$$\Omega_t = \Omega_{s_1}|f_1| + \Omega_{s_2}|f_2|, \tag{2.13}$$

where Ω_{s_1}, Ω_{s_2}, and Ω_t are the spatial and temporal bandwidths of the brightness $I(\mathbf{s},t)$ and f_1 and f_2 are the components of the flow vector f. An implication of this equation is as follows: Suppose that a given brightness pattern $I_0(\mathbf{s})$ is translated with a constant velocity f and that we are to sample the resulting brightness signal $I(\mathbf{s},t)$ and redetermine the velocity from the sampled images. To correctly perceive the motion from the sampled image sequence we need to avoid aliasing. The required minimum spatial sampling rates are determined by the spatial bandwidths Ω_{s_1} and Ω_{s_2} of the original brightness pattern $I_0(\mathbf{s})$. According to (2.13), however, the temporal sampling rate is a function of not only the bandwidths of the original image $I_0(\mathbf{s})$ but also the magnitudes of the velocity components. Thus, for a given scene $I_0(\mathbf{s})$ and optical flow f, there exists a minimum temporal sampling rate for undistorted motion perception. Martinez notes that temporal undersampling is often a problem in accurate motion estimation in typical video (e.g. TV) signals.

2.4.2 Temporal sampling and BCCE

Let us now consider the effects of temporal sampling on the BCCE. Let the sampling interval be Δt. Then a discrete version of the brightness invariance principle (2.1) can be written as

$$I(\mathbf{s}+\Delta\mathbf{s}, t+\Delta t) - I(\mathbf{s},t) = 0, \tag{2.14}$$

where

$$\Delta\mathbf{s} \equiv (\Delta t) f(\mathbf{s}). \tag{2.15}$$

By performing a Taylor expansion on (2.14) we obtain

$$\begin{aligned} 0 \;=\; & \frac{\partial I}{\partial \mathbf{s}}(\Delta \mathbf{s}) + \frac{\partial I}{\partial t}(\Delta t) \\ & + \frac{\partial^2 I}{\partial \mathbf{s} \partial t}(\Delta \mathbf{s})(\Delta t) + \frac{1}{2}(\Delta \mathbf{s})^T \frac{\partial^2 I}{\partial \mathbf{s}^2}(\Delta \mathbf{s}) + \frac{(\Delta t)^2}{2}\frac{\partial^2 I}{\partial t^2} \\ & + \cdots \end{aligned} \tag{2.16}$$

Then, by substituting (2.15) and dividing both sides by Δt, we can reduce Equation (2.16) to BCCE (2.3), if all the second order partial derivatives are zero, i.e.,

$$\frac{\partial^2 I}{\partial \mathbf{s} \partial t} = 0, \quad \frac{\partial^2 I}{\partial \mathbf{s}^2} = 0, \quad \text{and} \quad \frac{\partial^2 I}{\partial t^2} = 0. \tag{2.17}$$

These derivatives are all zero if and only if the brightness is linear with respect to its coordinates, i.e.,

$$I(\mathbf{s}, t) = [c_3,\ c_2]\,\mathbf{s} + c_1 t + c_0 \tag{2.18}$$

for some constants $c_0, \ldots, c_3$. Although we are not likely to encounter such an idealized brightness signal in practice, conditions like (2.17) and (2.18) tend to be approximatedly satisfied *regionally* in typical image sequences. In subregions of (sampled) images where (2.17) or (2.18) is badly violated, however, we cannot expect BCCE to provide us with correct information about the optical flow vectors.

2.4.3 Noise and brightness gradients

So far we have not discussed the effect of noise in obtaining the spatial and temporal gradients from samples of the brightness signal. Measuring gradients from images is a notoriously noise-sensitive process. Suppose the measurement $I'(\mathbf{s}, t)$ of the true brightness values $I(\mathbf{s}, t)$ is corrupted by an additive white noise $n(\mathbf{s}, t)$ with a variance σ^2:

$$I'(\mathbf{s}, t) = I(\mathbf{s}, t) + n(\mathbf{s}, t).$$

If the brightness gradients are computed by first order differencing, e.g.,

$$\begin{aligned}\frac{\partial I}{\partial s_1} &= \frac{1}{\Delta}\left(I'(s_1+\Delta, s_2, t) - I'(s_1, s_2, t)\right) \\ &= \frac{1}{\Delta}\left(I(s_1+\Delta, s_2, t) - I(s_1, s_2, t)\right) + \frac{1}{\Delta}\left(n(s_1+\Delta, s_2, t) - n(s_1, s_2, t)\right),\end{aligned}$$

then the variance for the corresponding additive noise increases to $\frac{4}{\Delta^2}\sigma^2$. Ideally, the sampling interval, Δ, should be small, but a smaller Δ leads to a larger effect of noise. Thus, gradient computation tends to amplify the effect of image noise, and for this reason the images are typically low-pass filtered to increase the signal-to-noise ratio before gradient compuatation. A simple and effective way to perform this is by some local spatial averaging of the image.

2.4.4 Techniques to reduce the discretization effects

Presmoothing the brightness

A way to reduce the effects of signal sampling[4] on BCCE is to intentionally blur the images before gradient computations. Kearney *et al.* [42] advocates such presmoothing because

- The second and highter order brightness gradients are diminished (so that (2.17) is approximated more closely over the sampled signal).
- The effects of noisy brightness measurement are reduced by averaging.

In another words, presmoothing leads to two desirable effects of decreasing the high frequency components in the signal (to facilitate application of the BCCE) and the noise (for more accurate computation of the image gradients). Additionally, a low-pass filtering effect of presmoothing decreases the spatial bandwidths Ω_{s_1} and Ω_{s_2} which in turn, via (2.13), reduces the temporal sampling rate required to avoid aliasing. Another way to look at this is

[4]more specifically, the effects of temporal undersampling. We assume that the spatial sampling rates satisfy the Nyquist criterion (high enough to avoid spatial aliasing). For typical images of natural scences, this assumption is reasonable.

- From an image sequence sampled with a particular temporal sampling rate, flow vectors with larger magnitudes can be estimated by decreasing the spatial bandwidths of the images by presmoothing.

In experiments to be presented in a later chapter, presmoothing is implemented by averaging over a small (e.g. 9×9), local neighborhood, and improvents in accuracy (over the cases in which no presmoothing has been applied) of the optical flow estimates are observed.

Checking the second order gradients

As discussed earlier, we cannot expect BCCE to provide us with useful information about motion in a sampled image sequence if the condition (2.17) is badly violated. Computing and checking the second order gradients $\frac{\partial^2 I}{\partial \mathbf{s} \partial t}$, $\frac{\partial^2 I}{\partial \mathbf{s}^2}$, and $\frac{\partial^2 I}{\partial t^2}$ at each pixel before optical flow computation are useful. Specifically, we can give a weight to BCCE associated with each pixel by how closely the measured second order gradients satisfy (2.17). We have found experimentally that, applied in conjunction with presmoothing, weighting BCCE at each pixel with a term

$$\exp\left(-k\left\|\frac{\partial^2 I}{\partial \mathbf{s} \partial t}\right\|^2\right),$$

with a constant parameter k, is particularly effective in increasing accuracy of the optical flow estimates (Section 6.4.2).

2.4.5 Summary

Undesirable effects of discretization on optical flow computation include:

1. For some fixed spatial and temporal sampling rates for the brightness signal, the magnitude of detectable motion is upperbounded.

2. Unless conditions like (2.17) and (2.18) are satisfied, the brightness change constraint equation loses its effectiveness as the temporal sampling interval increases.

3. Obtaining brightness gradients by finite differencing amplifies noise.

These effects can be reduced by

- presmoothing, i.e., blurring the images spatially before computing the gradients,

and

- weighting BCCE with a function of the magnitude(s) of the computed second order brightness gradient(s).

Chapter 3

An Estimation Theoretic Perspective of Single-frame Visual Reconstruction

In the previous chapter we have seen that optical flow can be computed from image gradients by solving a weighted least-squares problem in which the functional to be minimized has two types of constraint terms. One constraint is based on the brightness change constraint equation (BCCE) relating the measured image gradients to the optical flow vector at each pixel, and the other is the spatial coherence constraint which has been introduced to regularize the problem so that a meaningful unique solution exists. Many other low-level visual reconstruction problems are formulated similarly, i.e., they are least squares problems with two types of constraint terms — constraints imposed by the measurements made on the images and spatial coherence constraints. Since in low-level visual field reconstruction we try to recover some features of the objects in a 3-D domain (such as depth, shape, and motion of an object surface) from some *projected* information available in 2-D images, reconstructing these fields from measurements made on the images alone leads to underconstrained inverse problems [31]. The inclusion of spatial coherence constraints is by far the most common approach to make the problem mathematically well-posed, i.e., to ensure existence, uniqueness, and stability of the solution [7].

A typical mathematical expression for spatial coherence constraints takes the form of cost terms penalizing magnitudes of spatial gradients of the unknown fields. Physically, this corresponds to the assumption that the imaged quantities such as the

object surfaces have stiffness and smoothness [30]. Such an assumption is obviously violated at some special locations in the images such as at object boundaries, and detection of such locations of spatial discontinuities have been actively pursued by modifying and supplementing the spatial coherence constraints [11, 17, 58, 66, 67]. The issue of dealing with discontinuities in the visual fields is beyond the scope of this thesis.

This chapter is concerned with describing an estimation theoretic formulation of a general low-level visual field reconstruction problem. We *interpret* the traditional least-squares formulations as estimation problems, facilitating development of efficient multi-frame reconstruction algorithms to be presented in the next chapter. Bayesian estimation perspectives on visual reconstruction problems have been introduced before [71, 27]; however, casting the problems into a strictly Baysian framework is somewhat awkward because, as shown later, the prior densities tend not to exist mathematically. The maximum likelihood (ML) estimation framework described in this chapter, by viewing spatial coherence as a "noisy observation" and representing the mean-covariance pair implicitly as the "information pair", provides us with a more natural way to express the reconstruction problems in an estimation theoretic context. Moreover, as we will see in the next chapter, the ML framework blends in nicely with multi-frame reconstruction schemes based on *descriptor dynamic systems* which permit a wider range of temporal dynamic representations for the fields than the traditional Gauss-Markovian state-space systems.

3.1 Single-frame Visual Reconstruction

In this and the next two chapters we consider visual field reconstruction problems which are slightly more general than computation of optical flow. In this way, we can make explicit that the estimation algorithms presented are directly applicable to various low-level visual field reconstruction problems. Furthermore, examination of various forms of spatial coherence constraints not typically associated with optical flow computation is quite beneficial in the development of temporal dynamic models

for multi-frame reconstruction to be discussed in the next chapter.

Consider the problem of reconstructing the visual field $f(\mathbf{s})$ over a K-dimensional spatial domain $\mathcal{D}$ (i.e., $\mathbf{s} \in \mathcal{D} \subset \Re^K$) based on the measurements $g(\mathbf{s})$ and $h(\mathbf{s})$ from the images. In reconstruction of a *vector* visual field like optical flow, $f(\mathbf{s})$ is a vector function of $\mathbf{s}$, and, correspondingly, $g(\mathbf{s})$ and $h(\mathbf{s})$ are vector and matrix functions, respectively. As presented before, for example, in the case of optical flow reconstruction $f(\mathbf{s})$, $g(\mathbf{s})$, and $h(\mathbf{s})$ have respective dimensions of 2×1, 1×1, and 1×2. The general form of the single-frame visual field reconstruction problem considered in this thesis is

$$\min_f \int_{\mathcal{D}} \nu \left\| g - hf \right\|^2 + \sum_{\mathbf{i}} \mu_{\mathbf{i}} \left\| \frac{\partial^{j_1}}{\partial {s_1}^{j_1}} \frac{\partial^{j_2}}{\partial {s_2}^{j_2}} \cdots \frac{\partial^{j_K}}{\partial {s_K}^{j_K}} f \right\|^2 d\mathbf{s} \tag{3.1}$$

where $\nu(\mathbf{s}) \neq 0$ and $\mu_{\mathbf{i}}$ are strictly positive weighting parameters. We denote the components of the spatial index vector $\mathbf{s}$ by $s_k, k = 1, 2, \ldots, K$. The dimension K of the spatial domain in most visual reconstruction problems is at most 3. The orders of partial derivatives j_k's are non-negative integers, and $\frac{\partial^0}{\partial {s_k}^0} \equiv 1$. The index $\mathbf{i}$ is used to distinguish the K-tuples $(j_1, j_2, \ldots, j_K)$.

3.1.1 Measurement constraints

The first integrand term in (3.1) constrains the unknown $f(\mathbf{s})$ based on the measurements $g(\mathbf{s})$ and $h(\mathbf{s})$. The weighting function $\nu(\mathbf{s})$ represents the confidence we have in the measurements at a particular image point $\mathbf{s}$. The specific form of the constraint depends on the nature of the particular reconstruction problem. For example, as shown in the last chapter, when the unknown $f(\mathbf{s})$ is the optical flow, the measurement constraint equation $g - hf = 0$ corresponds to the *brightness change constraint equation* in which $g(\mathbf{s})$ and $h(\mathbf{s})$ are obtained from the temporal and spatial gradients of the image brightness. Another example is the generation of *sparse measurements of the depth field* by stereo-matching of feature points in a pair of images [21, 22]. In this case, the unknown $f(\mathbf{s})$ represents the depth field while $g(\mathbf{s})$ is the

sparse measurement of $f(\mathbf{s})$, and $h(\mathbf{s})$ is a binary function such that

$$h(\mathbf{s}) = \begin{cases} 1, & \text{if the depth is measured at } \mathbf{s} \\ 0, & \text{otherwise.} \end{cases}$$

Note that at the points $\mathbf{s}$ where $h(\mathbf{s}) = 0$, $f(\mathbf{s})$ is unconstrained by the measurement $g(\mathbf{s})$.

Linearity of the measurement constraints

The measurement constraint in (3.1) is linear with respect to $f(\mathbf{s})$. Although there are visual field reconstruction problems in which the measurement constraints are non-linear, the computational techniques designed for the linear measurement case can frequently be used in solving non-linear problems as well. For example, the non-linear measurement constraint equation for the *shape-from-shading* problem [33] has been approximated linearly based on Taylor expansion with good numerical results [27, 28]. Linearity in the measurement constraint also facilitates analytical and numerical tractability of the multi-frame reconstruction algorithms to be developed in the sequel. We, thus, concentrate our efforts in this thesis on the linear measurement case.

3.1.2 Spatial coherence constraints

In (3.1) the spatial coherence constraint is expressed as the sum of quadratic terms involving spatial derivatives of the unknown field $f(\mathbf{s})$. First and second order derivatives are most commonly used. While spatial coherence constraints make the reconstruction problems mathematically well-posed by supplementing the measurement constraints [7], they can also be considered to be our prior knowledge about the unknown field before the measurements are made [71]. Such prior models implied by spatial coherence constraints are successfully applied to a remarkable variety of visual reconstruction problems. Here are examples of some extensively used models:

- First order differential constraints and the Membrane model.

  Penalizing large magnitudes of the first order spatial derivatives tends to pro-

mote smoothly-varying behavior in the reconstruction of $f(\mathbf{s})$. On a 2-D spatial domain, the constriant is often called the *membrane* model, and it takes the form

$$\mu_1 \left\| \frac{\partial}{\partial s_1} f \right\|^2 + \mu_2 \left\| \frac{\partial}{\partial s_2} f \right\|^2 \tag{3.2}$$

where typically $\mu_1 = \mu_2$. The membrane model has been used in optical flow computation [34, 30] (as described in the previous chapter) and the shape-from-shading problem [33].

- Second order differntial constraints and the Thin-plate model.

  Constraining the second order derivatives promotes linearity, i.e., it tends to make $f(\mathbf{s})$ "straight" over a 1-D domain and "flat" over a 2-D domain. On a 2-D domain, this constraint is often called the *thin-plate* model and takes the form

$$\mu_1 \left\| \frac{\partial^2}{\partial {s_1}^2} f \right\|^2 + \mu_2 \left\| \frac{\partial^2}{\partial {s_2}^2} f \right\|^2 + \mu_3 \left\| \frac{\partial^2}{\partial s_1 \partial s_2} f \right\|^2 \tag{3.3}$$

  where typically $\frac{1}{2}\mu_3 = \mu_1 = \mu_2$. The thin-plate model is often used in surface interpolation [21, 22, 27, 28]. Note that if the membrane model were used as the surface model, it would favor a flat surface parallel with the image frame over a flat surface at an angle to the frame. The thin-plate model is, however, indifferent to the angle at which the surface is placed with respect to the frame.

- Hybrid constraints.

  In some cases, combining constraints based on the first and second order derivatives can be useful. A constraint of the form

$$\mu_1 \left\| \frac{\partial}{\partial s} f \right\|^2 + \mu_2 \left\| \frac{\partial^2}{\partial s^2} f \right\|^2 \tag{3.4}$$

  has been used to model the structure of object boundary contours [41], where $f(s)$ represents the coordinate vector (i.e., the location in the space) of a point

on the contour and s is the index along the contour. The first order derivative term keeps the structural integrity of the contour by encouraging neighboring points on the contour to be closely located in the image frame, while the second order term promotes smooth contours.

3.1.3 Solving the optimization problem

By far the most common approach to solving the optimization problem for single-frame visual reconstruction (3.1) is via variational calculus [33]. Namely, an Euler-Lagrange equation is obtained from the cost functional in (3.1) so that the solution is expressed implicitly as the solution of a partial differential equation. The partial differential equation is then solved by numerical techniques which usually involves iterative relaxation methods.

3.2 Maximum Likelihood Visual Reconstruction

In practice, the measurements are available over a *discrete* spatial domain. Rather than using the continuous formulation (3.1) and then discretizing the resulting Euler-Lagrange equation, we focus here on a purely discrete, vectorized formulation representing a discrete counterpart to (3.1). The single-frame visual reconstruction problem formulated in this way will be shown to be equivalent to a maximum likelihood estimation problem.

3.2.1 Discretization

Let us consider discretizating the spatial domain $\mathcal{D} \subset \Re^K$ by a uniform rectangular grid. Without loss of generality, we can set the grid points at integral values of the spatial coordinates, i.e., the components s_k of the spatial position vector $\mathbf{s}$ now take integer values such that $s_k = 1, 2, \ldots, m_k$ for given m_k's.

The vectorized visual field

We represent the unknown field over the entire discrete spatial domain by a single vector. Let us define the vector $\mathbf{f}$ to be a column vector whose components are $f(\mathbf{s})$ ordered lexicographically according to the spatial indices. Specifically, for $K = 3$, $\mathbf{f}$ is defined as

$$\mathbf{f} \equiv \begin{bmatrix} f_3(1) \\ f_3(2) \\ \vdots \\ f_3(m_3) \end{bmatrix}, \quad f_3(s_3) \equiv \begin{bmatrix} f_2(1, s_3) \\ f_2(2, s_3) \\ \vdots \\ f_2(m_2, s_3) \end{bmatrix}, \quad f_2(s_2, s_3) \equiv \begin{bmatrix} f(1, s_2, s_3) \\ f(2, s_2, s_3) \\ \vdots \\ f(m_1, s_2, s_3) \end{bmatrix},$$

e.g., if $f(\mathbf{s})$ is a *scalar* field, the unknown at a particular $\mathbf{s} = [s_1, s_2, s_3]^T$ is represented by the $[s_1 + (s_2 - 1)m_1 + (s_3 - 1)m_1 m_2]^{\text{th}}$ component of $\mathbf{f}$. Analogous definitions are used for the cases $K = 1$ and 2 and when $f(\mathbf{s})$ is a vector field. In general, if $f(\mathbf{s})$ is a d-dimensional vector field and the number of points in the discrete spatial domain is N, then the dimension of the vector $\mathbf{f}$ is Nd. The measurement vector $\mathbf{g}$ is similarly defined from $g(\mathbf{s})$. We define the matrix $\mathbf{H}$ to be a block diagonal matrix whose diagonal blocks are $h(\mathbf{s})$ sequenced according to the spatial indices by the same lexicographical order.

The spatial difference operators

Let us define $\mathbf{D}_k^{(j)}$ to be a matrix operator approximating the j^{th} order differentiation along the k^{th} spatial axis, i.e., $\mathbf{D}_k^{(j)}\mathbf{f} \approx \left[\frac{\partial^j}{\partial s_k{}^j} f(\mathbf{s})\right]$. In particular, the general forms for the first and second order difference operators, $\mathbf{D}_k^{(1)}$ and $\mathbf{D}_k^{(2)}$, used in this thesis are

$$\mathbf{D}_k^{(1)} = \begin{bmatrix} \Delta^{(1)} & & \\ & \ddots & \\ & & \Delta^{(1)} \end{bmatrix}, \quad \Delta^{(1)} \equiv \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \\ & \ddots & \ddots \\ & & -\mathbf{I} \quad \mathbf{I} \end{bmatrix},$$

$$\mathbf{D}_k^{(2)} = \begin{bmatrix} \Delta^{(2)} & & \\ & \ddots & \\ & & \Delta^{(2)} \end{bmatrix}, \quad \Delta^{(2)} \equiv \begin{bmatrix} \mathbf{I} & -2\mathbf{I} & \mathbf{I} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{I} & -2\mathbf{I} & \mathbf{I} \end{bmatrix},$$

where the appropriate dimensions for the blocks Δ's and $\mathbf{I}$'s are determined by k (i.e., by the particular spatial axis along which the difference is taken) as well as the dimension of the field components $f(\mathbf{s})$ (n.b., the unknown might be a vector field like optical flow). Using this notation, a discrete version of the membrane model can be written as

$$\mu_1 \left\| \mathbf{D}_1^{(1)} \mathbf{f} \right\|^2 + \mu_2 \left\| \mathbf{D}_2^{(1)} \mathbf{f} \right\|^2. \tag{3.5}$$

In order to represent the cross spatial derivatives in the spatial coherence constraint, we need to expand the notation for the matrix spatial difference operator. Let us define the matrix operator $\mathbf{S}^{(j_1, j_2, \ldots, j_K)}$ to be a finite difference approximation of the cross differentiation such that

$$\mathbf{S}^{(j_1, j_2, \ldots, j_K)} \mathbf{f} \approx \left[\frac{\partial^{j_1}}{\partial {s_1}^{j_1}} \frac{\partial^{j_2}}{\partial {s_2}^{j_2}} \cdots \frac{\partial^{j_K}}{\partial {s_K}^{j_K}} f(\mathbf{s}) \right].$$

For example, for a 2-D spatial domain ($K = 2$),

$$\mathbf{S}^{(1,1)} = \begin{bmatrix} -\Delta^{(1)} & \Delta^{(1)} & & \\ & \ddots & \ddots & \\ & & -\Delta^{(1)} & \Delta^{(1)} \end{bmatrix}$$

where $\Delta^{(1)}$ is defined before. A discrete version of the thin-plate model can then be written as

$$\mu_1 \left\| \mathbf{S}^{(2,0)} \mathbf{f} \right\|^2 + \mu_2 \left\| \mathbf{S}^{(0,2)} \mathbf{f} \right\|^2 + \mu_3 \left\| \mathbf{S}^{(1,1)} \mathbf{f} \right\|^2. \tag{3.6}$$

Note the following equivalent notations: $\mathbf{S}^{(1,0)} \equiv \mathbf{D}_1^{(1)}$ and $\mathbf{S}^{(0,1)} \equiv \mathbf{D}_2^{(1)}$. Thus, the

membrane model (3.5) can also be written as

$$\mu_1 \left\| \mathbf{S}^{(1,0)}\mathbf{f} \right\|^2 + \mu_2 \left\| \mathbf{S}^{(0,1)}\mathbf{f} \right\|^2 . \tag{3.7}$$

We adopt this latter notations for the membrane model throughout the thesis.

A discrete single-frame reconstruction problem

Using the vectors and matrices defined above, we can now write a discrete version of the single-frame low-level visual field reconstruction problem in (3.1) as

$$\min_{\mathbf{f}} \left(\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_{\mathbf{N}}^2 + \sum_i \mu_i \|\mathbf{S}_i\mathbf{f}\|^2 \right) \tag{3.8}$$

where

$$\mathbf{S}_i \equiv \mathbf{S}^{(j_1, j_2, \ldots, j_K)_i} \quad \text{for } i = 1, 2, \ldots,$$

and $(j_1, j_2, \ldots, j_K)_i$ denotes a particular value of the K-tuple $(j_1, j_2, \ldots, j_K)$ for each i while

$$\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_{\mathbf{N}}^2 \equiv (\mathbf{g} - \mathbf{H}\mathbf{f})^T \mathbf{N} (\mathbf{g} - \mathbf{H}\mathbf{f})$$

and $\mathbf{N}$ is a diagonal matrix whose diagonal components are $\nu(\mathbf{s})$ sequenced according to the spatial indices by the same lexicographical order as $f(\mathbf{s})$ in $\mathbf{f}$.

3.2.2 Maximum likelihood estimation

The discrete, weighted least-squares formulation of the single-frame visual reconstruction problem (3.8) can be directly converted to a maximum likelihood estimation problem. As general theories of maximum likelihood estimation are of central importance to the development of algorithms in the subsequent chapters, we briefly present here some fundamental facts relevant to our visual reconstruction problems as well as establish some notation that will be used frequently in the sequel. For derivations

and other details, introductory texts on estimation theory (e.g. [16, 45, 74]) should be consulted.

The *maximum likelihood (ML)* estimation problem deals with estimation of an unknown vector $\mathbf{x}$ based on an observed sample of a random vector $\mathbf{y}$ whose distribution is parameterized by $\mathbf{x}$. Specifically, let $\mathcal{P}_{\mathbf{y}}(\cdot|\mathbf{x})$ be the probability density function for such a distribution. Given a sample value $\mathbf{y}_o$ of $\mathbf{y}$, the maximum likelihood estimate $\hat{\mathbf{x}}$ is defined to be the value of $\mathbf{x}$ such that $\mathcal{P}_{\mathbf{y}}(\mathbf{y}_o|\mathbf{x})$ is maximized. Much of development of estimation theory is based on problems that involves random processes with the *Gaussian probability density functions.*

Definition 1 We use the notation

$$\mathbf{r} \sim (\mathbf{r}_m, \mathbf{R}) .$$

to express that "$\mathbf{r}$ is a random vector with a joint Gaussian probability density function having a mean $\mathbf{r}_m$ and covariance matrix $\mathbf{R}$." We call $(\mathbf{r}_m, \mathbf{R})$ the *mean-covariance pair.*

One of the most extensively studied ML estimation problems involves a linear observation $\mathbf{y}$ of the unknown $\mathbf{x}$ corrupted by an additive zero-mean Gaussian random noise:

Find the ML estimate for $\mathbf{x}$ based on the observation

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{r}, \quad \mathbf{r} \sim (0, \mathbf{R}) \tag{3.9}$$

in which $\mathbf{C}$ is a given matrix and $\mathbf{r}$ is a zero-mean Gaussian random noise.

Note that the observation equation (3.9) implies that $\mathbf{y}$ has a Gaussian distribution parameterized by $\mathbf{x}$. In this thesis, this is the only type of ML estimation problems that we deal with. We refer to a linear observation corrupted by an additive zero-mean Gaussian noise, like that in (3.9), as a "Gaussian observation equation" or simply an "observation equation".

The ML estimate $\hat{\mathbf{x}}$ based on (3.9) given a value of $\mathbf{y}$ is

$$\hat{\mathbf{x}} = \left(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} \tag{3.10}$$

The *estimation error covariance* $\mathbf{P}_x$ is defined as $\mathbf{P}_x \equiv \mathcal{E}(\|\mathbf{x} - \hat{\mathbf{x}}\|^2)$ where $\mathcal{E}(\cdot)$ is the expected value operator, and for this ML estimation problem it is

$$\mathbf{P}_x = \left(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}. \tag{3.11}$$

Note that the effect of the measurement (3.9) is in essence to provide us with a probability density for $\mathbf{x}$:

Observation 3.1 *The vector* $\mathbf{x}$ *can be considered to have a* posterior *Gaussian distribution*

$$\mathbf{x} \sim (\ \hat{\mathbf{x}}, \mathbf{P}_x\ ), \tag{3.12}$$

after a particular value of the observation $\mathbf{y}$ *is obtained.*

An important fact regarding the ML estimation based on a "Gaussian observation equation" is its relationship with a weighted least-squares problem:

Fact 3.2 *The ML estimation problem for the unknown* $\mathbf{x}$ *based on the observation (3.9) is equivalent to the weighted least-squares problem*

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{C}\mathbf{x}\|^2_{\mathbf{R}^{-1}}. \tag{3.13}$$

This equivalence relation can be used to interpret deterministic least-squares problems statistically. Note: there is an implicit assumption that $\mathbf{R}$ is invertible and that $\mathbf{C}$ has a full column rank for the problem to be well-posed.

3.2.3 ML single-frame visual reconstruction

Using Fact 3.2 we can interpret the discrete single-frame visual reconstruction problem (3.8) as an ML estimation problem based on an observation equation. First note that (3.8) can be written as

$$\min_{\mathbf{f}} \|\mathbf{y} - \mathbf{C}\mathbf{f}\|^2_{\mathbf{R}^{-1}} \tag{3.14}$$

by defining

$$\mathbf{y} \equiv \begin{bmatrix} \mathbf{g} \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad \mathbf{C} \equiv \begin{bmatrix} \mathbf{H} \\ \mathbf{S}_1 \\ \mathbf{S}_2 \\ \vdots \end{bmatrix}, \quad \mathbf{R} \equiv \begin{bmatrix} \mathbf{N}^{-1} & & & \\ & \mu_1^{-1}\mathbf{I} & & \\ & & \mu_2^{-1}\mathbf{I} & \\ & & & \ddots \end{bmatrix}. \tag{3.15}$$

Then, estimation based on the Gaussian observation

$$\mathbf{y} = \mathbf{C}\mathbf{f} + \mathbf{r}, \quad \mathbf{r} \sim (\, 0, \mathbf{R} \,) \tag{3.16}$$

formulates the single-frame visual field reconstruction problem as an ML estimation problem.

The prior model as a set of observations

Let us look at this ML visual reconstruction problem closely. First note that the observation equation (3.16) consists of independent Gaussian observations

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{r}_0, \quad \mathbf{r}_0 \sim \left(0, \mathbf{N}^{-1} \right) \tag{3.17}$$

and

$$0 = \mathbf{S}_i\mathbf{f} + \mathbf{r}_i, \quad \mathbf{r}_i \sim \left(0, \mu_i^{-1}\mathbf{I} \right) \tag{3.18}$$

for $i = 1, 2, \ldots$. They are independent in the sense that the observation noise processes $\mathbf{r}_i$'s (for $i = 0, 1, 2, \ldots$) are probabilistically independent of each other. The ML estimation form of the single-frame visual field reconstruction involves finding an $\mathbf{f}$ that maximizes the *joint probability* of the particular set of observations $(\mathbf{g}, 0, 0, \cdots)$. This expanded view of the observations (as opposed to the composite view of (3.16)) is useful in gaining insights into how pieces of information about the unknown $\mathbf{f}$ are each represented and combined to yield the final estimate. For example, (3.17) represents the contribution from the measurements in the images, and (3.18) represents the prior knowledge (model) of the field, i.e., the spatial coherence constraints. In particular, the prior model implied by the spatial coherence constraints is expressed in the ML estimation framework as a set of observations (3.18), each indicating that the differential $\mathbf{S}_i\mathbf{f}$ of the unknown is "observed" to be zero, the ideal situation, with an uncertainty of $\mu_i^{-1}\mathbf{I}$.

3.2.4 Example: ML optical flow estimation

A discrete version of the optical flow computation problem can be formulated using a vectorized brightness constraint and the discrete membrane model of (3.7) as

$$\min_{\mathbf{f}} \left(\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_{\mathbf{N}}^2 + \mu_1 \|\mathbf{S}_1\mathbf{f}\|^2 + \mu_2 \|\mathbf{S}_2\mathbf{f}\|^2 \right) \tag{3.19}$$

where $\mathbf{g}$ and $\mathbf{H}$ are constructed from the image gradient measurements $\mathbf{g}(\mathbf{s}) = \frac{\partial I}{\partial \mathbf{s}}(\mathbf{s})$ and $\mathbf{h}(\mathbf{s}) = -\frac{\partial I}{\partial t}(\mathbf{s})$ while $\mathbf{S}_1$ and $\mathbf{S}_2$ are the first order difference operators, $\mathbf{S}_1 = \mathbf{S}^{(1,0)}$ and $\mathbf{S}_2 = \mathbf{S}^{(0,1)}$. This yields a maximum likelihood optical flow estimation problem based on the following "optical flow observation equation":

$$\begin{bmatrix} \mathbf{g} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \mathbf{f} + \mathbf{r}, \quad \mathbf{r} \sim \left(0, \begin{bmatrix} \mathbf{N}^{-1} & & \\ & \mu_1^{-1}\mathbf{I} & \\ & & \mu_2^{-1}\mathbf{I} \end{bmatrix} \right). \tag{3.20}$$

Interpreting the measurement noise

By expanding the observation equation as in the last subsection, we see that the brightness constraint (derived from BCCE) is represented in the ML estimation framework as

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{r}_0, \quad \mathbf{r}_0 \sim \left(0, \mathbf{N}^{-1}\right). \tag{3.21}$$

In this form, the observation noise $\mathbf{r}_0$ represents the deviation one allows $\mathbf{f}$ to make from the BCCE. In the last chapter we have seen how we might select components of $\mathbf{N}^{-1}$ to reflect the effect of undersampling and signal noise on gradient measurements. In some practical cases, more explicit statistical models for the quality of gradient measurements might exist. In such cases one might want to use an alternative observation model such as

$$\mathbf{g} + \mathbf{r}_g = (\mathbf{H} + \mathbf{r}_H)\mathbf{f}$$

where $\mathbf{r}_g$ and $\mathbf{r}_H$ are the noise processes representing the uncertainties in the gradient measurements. This leads to a *total least-squares* problem [18]. Other alternatives for measurement noise modeling are possible, but in this thesis we will not pursue this issue any further.

3.3 Implicit Solution for ML Estimation

Let us consider solving the single-frame ML visual field reconstruction problem. In solving an ML estimation problem based on a "Gaussian observation", one is ultimately interested in obtaining the posterior mean-covariance pair. We have presented a conceptual framework above such that the unknown visual field is reconstructed by fusing two sets of independent observations (3.17) and (3.18). The equations based on the spatial coherence constraints, (3.18), are considered to be "pre-observed" in the sense that the ideal values (zeros) have been observed before the actual mea-

surements on the images, (3.17), are made. Thus, one might consider first obtaining the ML solution based only on the spatial coherence constraints and then updating the resulting mean-covariance pair with the image measurements. This is exactly the *Bayesian* approach to the estimation problem where, essentially, all the pieces of the independent statistical knowledge assimilated so far for the purpose of estimating an unknown vector are summarized in the corrsponding mean-covariance pair. One can speculate that this approach is particularly convenient in a *multi-frame* visual reconstruction problem in which observations arriving sequentially over time must be accumulated in a statistically optimal way. The Bayesian approach is, however, not directly applicable to visual reconstruction because the covariance matrix does not always exist. For example, for a typical spatial coherence constraint the corresponding observation equations (3.18) do not constrain the unknown well enough to yield a posterior mean-covariance pair.

In this section we examine an implicit representation for the mean-covariance pair, called the *information pair*, and its uses in ML visual reconstruction. For our purposes, the information pairs are preferable to the mean-covariance pair because:

1. At all times in the visual reconstruction process, including the sequential multi-frame reconstruction problem to be presented later, the information pair for the unknown always exist.

2. As we will see, for our problems the information pair represents the same statistical data contained in the corresponding mean-covariance pair much more compactly than the mean-covariance pair itself. Specifically, the *information matrix* in single-frame visual reconstruction is very sparse whereas the corresponding covariance matrix is a full matrix.

3. Fusing statistical data contained in several independent observation equations can be achieved easily by a simple component-wise sum of the information pairs corresponding to each observation equation. The same operation with the mean-covariance pair involves matrix inversions.

Item 1 above has a theoretical implication; the information pair gives us a mathemati-

cal foundation on which we design computational algorithms for visual reconstruction. Item 2 has an implementational implication; fewer memory elements are required to store the statistical data implicitly than explicitly. Item 3 has a computational implication; the information pairs are convenient in statistical data fusion which can be performed with a simple component-wise operation. The fact that this computation can be performed independently for each component is important because it implies that the operation can be implemented in parallel, a desirable feature especially in image processing algorithms including visual field reconstruction.

Ultimately, the statistical data in the information pairs must be made explicit by translating the pairs into the mean-covariance pairs. As shown below, this translation process is actually embedded in the standard solution method for an ML estimation problem, and the sparseness of the information matrix (Item 2 above) plays a key role in enabling us to compute the mean-covariance pair efficiently.

3.3.1 Information form of ML estimate

Let us consider an implicit solution for a general ML estimation problem based on a Gaussian observation equation. We will specialize the discussions here to the single-frame ML visual reconstruction problem in the next subsection.

Definition 2 Consider a maximum likelihood estimation problem for the unknown $\mathbf{x}$ based on the observation equation (3.9), i.e., $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{r}$, $\mathbf{r} \sim (\ 0, \mathbf{R}\ )$. We call the quantities $\mathbf{z}_x$ and $\mathbf{L}_x$ defined as

$$\mathbf{z}_x \equiv \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} \tag{3.22}$$

$$\mathbf{L}_x \equiv \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} \tag{3.23}$$

the *information pair* associated with the unknown $\mathbf{x}$. We use double angular brackets as in

$$\mathbf{x} \sim \langle\langle\ \mathbf{z}_x, \mathbf{L}_x\ \rangle\rangle$$

to denote information pairs in order to distinguish them notationally from mean-covariance pairs. Also, the matrix $\mathbf{L}_x$ is called the *information matrix*[1].

By our assumption the observation noise covariance matrix $\mathbf{R}$ is always invertible; thus, the information pair exists for every observation equation. From (3.10) and (3.11) we see that the estimate and error covariance for the ML estimation problem can be obtained from the corresponding information pair as

$$\hat{\mathbf{x}} = \mathbf{L}_x^{-1}\mathbf{z}_x \tag{3.24}$$

$$\mathbf{P}_x = \mathbf{L}_x^{-1}. \tag{3.25}$$

Thus, the information pair $\langle\langle\ \mathbf{z}_x, \mathbf{L}_x\ \rangle\rangle$ contains the same statistical data as those in $(\ \hat{\mathbf{x}}, \mathbf{P}_x\ )$. Specifically, the information pair expresses the solution of the ML estimation problem *implicitly* in the sense that the estimate is given as the solution for an inverse problem

$$\mathbf{L}_x\hat{\mathbf{x}} = \mathbf{z}_x. \tag{3.26}$$

Clearly, invertibility of the information matrix $\mathbf{L}_x$ is cruicial for existence of $\hat{\mathbf{x}}$ and $\mathbf{P}_x$. In fact, we have the following:

Theorem 3.3 *For an ML estimation problem based on a Gaussian observation equation, the following three statements are equivalent:*

1. *A unique solution for the estimation problem exists.*
2. *The posterior estimation error covariance exists.*
3. *The information matrix is invertible.*

Proof: From (3.25) we have *2* $\leftrightarrow$ *3*, and from (3.24) we have *3* $\rightarrow$ *1*. It remains to show that either *1* $\rightarrow$ *2* or *1* $\rightarrow$ *3*. Suppose that *1* is true but *3* is not. By writing (3.24) as $\mathbf{L}_x\hat{\mathbf{x}} = \mathbf{z}_x$, we see that since $\mathbf{L}_x$ is not invertible $\hat{\mathbf{x}}$ cannot be unique. Thus,

[1]It is also called the *observation gramian.*

by contradiction, *1* $\rightarrow$ *3*. $\square$

If the information matrix is not invertible, then certain linear combinations of the elements of the mean of the corresponding random vector cannot be determined [39]. The information pair representation is more advantageous than the mean-covariance pair representation $\mathbf{x} \sim (\, \hat{\mathbf{x}}, \mathbf{P}_x \,)$ because the former always exists while the latter exists only if $\mathbf{L}_x$ is invertible. Moreover, as we will see later, the information matrices in single-frame visual field reconstruction problems are quite sparse, allowing us not only to store the second order statistics of the fields efficiently but also compute the estimates of the fields efficiently.

Fusing independent observations

Consider the ML estimation problem of maximizing the joint probability of the observations

$$\mathbf{y}_i = \mathbf{C}_i\mathbf{x} + \mathbf{r}_i, \quad \mathbf{r}_i \sim (\, 0, \mathbf{R}_i \,) \tag{3.27}$$

for $i = 1, 2, \ldots, m$, where $\mathbf{r}_i$'s are mutually independent. The information form of the ML solution is useful because of the following *additive property*:

Theorem 3.4 *Let the information pair associated with each of the observation equations in (3.27) be denoted by* $\langle\langle\, \mathbf{z}_i, \mathbf{L}_i \,\rangle\rangle$. *Then, the information pair associated with the problem of maximizing the joint probability of the observations for* $i = 1, 2, \ldots, m$ *is given by*

$$\left\langle\left\langle \sum_{i=1}^{m} \mathbf{z}_i \,,\, \sum_{i=1}^{m} \mathbf{L}_i \right\rangle\right\rangle,$$

assuming that the noise processes $\mathbf{r}_i$ *are mutually independent.*

Proof: The joint maximization problem can be formulated as the ML estimation problem based on the observation equation

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_m \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_m \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_m \end{bmatrix} \sim \left(0, \begin{bmatrix} \mathbf{R}_1 & & & \\ & \mathbf{R}_2 & & \\ & & \ddots & \\ & & & \mathbf{R}_m \end{bmatrix} \right),$$

and the information pair associated with this observation is

$$\left\langle\left\langle \sum_{i=1}^{m} \mathbf{C}_i^T \mathbf{R}_i^{-1} \mathbf{y}_i \, , \, \sum_{i=1}^{m} \mathbf{C}_i^T \mathbf{R}_i^{-1} \mathbf{C}_i \right\rangle\right\rangle$$

which proves the theorem. □

3.3.2 Implicit solution for visual reconstruction

Let us now focus on solution for the single-frame visual reconstruction problems. The information pair associated with (3.17) is

$$\left\langle\left\langle \mathbf{H}^T \mathbf{N} \mathbf{g} \, , \, \mathbf{H}^T \mathbf{N} \mathbf{H} \right\rangle\right\rangle$$

while those associated with (3.18) are

$$\left\langle\left\langle 0 \, , \, \sum_{i=1}^{m} \mu_i \mathbf{S}_i^T \mathbf{S}_i \right\rangle\right\rangle$$

for $i = 1, 2, \ldots, m$. Using Theorem 3.4 we have the implicit solution for the single-frame visual reconstruction problem as

$$\mathbf{f} \sim \left\langle\!\!\left\langle \mathbf{H}^T\mathbf{N}\mathbf{g} \, , \, \mathbf{H}^T\mathbf{N}\mathbf{H} + \sum_{i=1}^{m} \mu_i \mathbf{S}_i^T \mathbf{S}_i \right\rangle\!\!\right\rangle$$

which for brevity we write as

$$\mathbf{f} \sim \langle\langle \, \mathbf{z}, \mathbf{L} \, \rangle\rangle \, .$$

Note that operations involved in computing this information pair can all be performed *locally* in the sense that each of the necessary matrix operations such as multiplication is confined to elements corresponding to neighboring spatial locations, as the matrices $\mathbf{H}$ and $\mathbf{N}$ are diagonal and $\mathbf{S}_i$'s are sparsely banded. In particular, each of the components $z(\mathbf{s})$ of $\mathbf{z}$ is given by $h(\mathbf{s})\nu(\mathbf{s})g(\mathbf{s})$, a strictly local multiplication of the matrix elements. The information matrix $\mathbf{L}$ has a *sparsely banded* structure, reflecting the structures of $\mathbf{S}_i^T\mathbf{S}_i$ which are also locally computable. (Note: the matrix $\mathbf{H}^T\mathbf{N}\mathbf{H}$ is diagonal, and it does not normally influence the structure of $\mathbf{L}$.) The general structure of the information matrices in visual reconstruction is a principal focus of Chapter 5; an example of such matrix structures is presented later in this section. As mentioned before, the sparse structure of the information matrix has an important computational implication in solution of the visual reconstruction problem. Specifically, the estimated field $\widehat{\mathbf{f}}$ is computed as the solution for the inverse problem

$$\mathbf{L}\widehat{\mathbf{f}} = \mathbf{z} \tag{3.28}$$

where the information matrix $\mathbf{L}$ acts as an operator to be inverted. It can be shown, in fact, that this equation is a discrete version of the Euler-Lagrange equation obtained in variational solution of the continuous formulation of the corresponding reconstruction problem. The Euler-Lagrange equation is usually solved numerically by first discretizing it to a form similar to (3.28). Due to the usual large dimension, N, of the unknown field the inverse problem (3.28) must be solved iteratively; for a typical

value $10^4 \sim 10^6$ of N direct inversion of an $N \times N$ matrix is not practical. The sparse structure of the information matrix $\mathbf{L}$ not only facilitates standard iterative inversion schemes like the Jacobi and Gauss-Seidel iterations by simplifying computation in each iterative step but also allows us to use more sophisticated and efficient iterations like the multigrid methods. Terzopoulos [72] reports that for various single-frame problems it takes $O(N)$ iterations to obtain reasonable estimates, but with a multi-grid method the amount of computation can be reduced by an order of magnitude or more.

Besides computational efficiency, the sparseness of $\mathbf{L}$ provides us with a compact way in which to store the statistics of the unknown field at various stages of estimation. This is an important issue especially in multi-frame reconstruction problems in which such statistics must be propagated over time for temporal data fusion.

Example

For the optical flow problem in Section 3.2.4, the information pair associated with the *membrane model* is

$$\left\langle\left\langle\, 0\,,\, \mu_1 \mathbf{S}_1^T \mathbf{S}_1 + \mu_2 \mathbf{S}_2^T \mathbf{S}_2 \,\right\rangle\right\rangle.$$

The information matrix $\mathbf{L}_M \equiv \mu_1 \mathbf{S}_1^T \mathbf{S}_1 + \mu_2 \mathbf{S}_2^T \mathbf{S}_2$ has the following sparse structure

$$\mathbf{L}_M = \begin{bmatrix} \Delta_e & -\mathbf{I} & & & & \\ -\mathbf{I} & \Delta & -\mathbf{I} & & & \\ & -\mathbf{I} & \Delta & -\mathbf{I} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\mathbf{I} & \Delta & -\mathbf{I} \\ & & & & -\mathbf{I} & \Delta_e \end{bmatrix},$$

where

$$\Delta_e \equiv \begin{bmatrix} 2I & -I & & & & \\ -I & 3I & -I & & & \\ & -I & 3I & -I & & \\ & & \ddots & \ddots & \ddots & \\ & & & -I & 3I & -I \\ & & & & -I & 2I \end{bmatrix} \quad \Delta \equiv \begin{bmatrix} 3I & -I & & & & \\ -I & 4I & -I & & & \\ & -I & 4I & -I & & \\ & & \ddots & \ddots & \ddots & \\ & & & -I & 4I & -I \\ & & & & -I & 3I \end{bmatrix}$$

where I is a 2×2 matrix reflecting that optical flow is a 2-vector field. By close inspection, we can verify that this information matrix $\mathbf{L}_M$ can also be considered as a discrete implementaion of the 2-D Laplacian operator $\left(\frac{\partial^2}{\partial {s_1}^2} + \frac{\partial^2}{\partial {s_2}^2}\right)$. Each row of $\mathbf{L}_M\mathbf{f}$ is a weighted sum of $f(s_1, s_2), f(s_1+1, s_2), f(s_1, s_2+1), f(s_1-1, s_2)$, and $f(s_1, s_2-1)$. That is, treated as a matrix operator, $\mathbf{L}_M$ operates over a small, local neighborhood in the spatial domain. For this reason, the sparsely banded matrix structure of $\mathbf{L}_M$ is sometime called the *nearest neighbor* structure [43].

The posterior information pair for the optical flow field is given as

$$\mathbf{f} \sim \left\langle\left\langle\, \mathbf{H}^T\mathbf{N}\mathbf{g}\,,\, \mathbf{H}^T\mathbf{N}\mathbf{H} + \mathbf{L}_M \,\right\rangle\right\rangle.$$

Since $\mathbf{H}^T\mathbf{N}\mathbf{H}$ is a block diagonal matrix (with block size of 2×2), the matrix $\mathbf{H}^T\mathbf{N}\mathbf{H}+\mathbf{L}_M$ retains the nearest-neighbor structure. Sparseness of this matrix is important as the optical flow estimate $\widehat{\mathbf{f}}$ is obtained as the solution of the inverse problem

$$(\mathbf{H}^T\mathbf{N}\mathbf{H} + \mathbf{L}_M)\widehat{\mathbf{f}} = \mathbf{H}^T\mathbf{N}\mathbf{g}.$$

This equation is a discrete version of the Euler-Lagrange equation derived by Horn and Schunck [34] who used Gauss-Seidel iterations to solve the equation numerically. Each iteration essentially involves computation of only the previously mentioned, locally weighted sum for each component of the field. By organizing the computation into such weighted sum operations over mutually disjoint spatial neighborhoods, some iterative methods (e.g. the Jacobi iteration) can be implemented in parallel.

3.3.3 Wellposedness of visual reconstruction problem

The information matrix associated with a typical spatial coherence constraint,

$$\sum_{i=1}^{m} \mu_i \mathbf{S}_i^T \mathbf{S}_i,$$

is singular. This means that:

1. The explicit statistics, i.e., the mean-covariance pairs, corresponding to the spatial coherence constraint do not exist so that a Bayesian perspective on visual reconstruction is not well-founded in a strict sense.

2. The measurements must provide enough extra constraints to yield an unique estimate.

The second point indicates that although $\sum_{i=1}^{m} \mu_i \mathbf{S}_i^T \mathbf{S}_i$ is singular $\mathbf{H}^T\mathbf{N}\mathbf{H} + \sum_{i=1}^{m} \mu_i \mathbf{S}_i^T \mathbf{S}_i$ needs to be non-singular for a unique estimate to exist. This mathematical requirement is not so severe in practice. For example, the information matrix associated with the membrane model for a scalar field (i.e., the scalar-field version of $\mathbf{L}_M$ defined above) has only one null eigenvalue[2]; thus, a measurement at a single point in the spatial domain can provide enough constraint to satisfy the mathematical requirement for a wellposed problem. Practically, measurements are made over a good portion of the spatial domain (e.g., image frame) and often, as in the case for optical flow estimation, over the entire domain. In the sequel, therefore, we assume that the posterior information matrix

$$\mathbf{H}^T\mathbf{N}\mathbf{H} + \sum_{i=1}^{m} \mu_i \mathbf{S}_i^T \mathbf{S}_i$$

is always invertible.

[2]Since optical flow is a 2-vector field, $\mathbf{L}_M$ used for optical flow estimation has two null eigenvalues.

3.3.4 Spatial models implied by information pairs

Given the information pair of a visual field $\mathbf{f}$ as

$$\mathbf{f} \sim \langle\langle\, \mathbf{z}, \mathbf{L} \,\rangle\rangle\,,$$

let us consider a *spatially dynamic representation* of the field

$$\mathbf{L}\mathbf{f} = \zeta \tag{3.29}$$

where ζ is a random vector. Since the mean-covariance pair for $\mathbf{f}$ is given by

$$\mathbf{f} \sim \left(\, \mathbf{L}^{-1}\mathbf{z}\,,\, \mathbf{L}^{-1} \,\right),$$

the mean and covariance of ζ can be computed as $\mathbf{L}\mathbf{L}^{-1}\mathbf{z} = \mathbf{z}$ and $\mathbf{L}\mathbf{L}^{-1}\mathbf{L}^{T} = \mathbf{L}$, respectively, i.e.,

$$\zeta \sim (\, \mathbf{z}, \mathbf{L}\,). \tag{3.30}$$

We can interpret (3.29) as a spatial model for the field $\mathbf{f}$ in which the spatial random process ζ specifies the probabilistic behavior of the field $\mathbf{f}$ by controlling certain interactions, defined by the matrix operator $\mathbf{L}$, among components of $\mathbf{f}$ (cf. [38]).

In single-frame visual reconstruction problems, the matrix $\mathbf{L}$ has a certain sparsely banded structure, reflecting the local properties of the spatial coherence and measurement terms. Besides the previously mentioned benefits in storage and computation, another desirable feature of such locality and sparseness of $\mathbf{L}$ is that the dynamic representation (3.29) (3.30) forms a basis for approximation in multi-frame problems. Specifically, the visual field $\mathbf{f}$ is modeled through its internal interactions encoded by $\mathbf{L}$ and a spatial random process ζ which controls these interactions probabilistically, and the joint correlation of this controlling process is also given by $\mathbf{L}$. It makes a *physical* sense that a visual field is specifed by such *spatially local* interactions (among its components) and statistics (of ζ). As we will see, in multi-frame reconstruction

the corresponding information matrices are not strictly local; however, based on this physical intuition they will be approximated by local, sparsely banded matrices without losing much accuracy in estimation.

Markov Random Fields

A statistically equivalent dynamic representation to (3.29) (3.30) can be obtained using the estimation error process $\tilde{\mathbf{f}} \equiv \hat{\mathbf{f}} - \mathbf{f}$ as

$$\mathbf{L}\tilde{\mathbf{f}} = \tilde{\zeta}.$$

It can be shown that the mean-covariance pair of the *driving noise* $\tilde{\zeta}$ is given by

$$\tilde{\zeta} \sim (\, 0, \mathbf{L} \,).$$

This dynamic model is shown to be a *Markov Random Field (MRF)* [44], which has a multi-dimensional extension of the Markovian property for causal processes [77]. Interpreting the information matrix $\mathbf{L}$ as an MRF model for the estimation error $\tilde{\mathbf{f}}$ can be quite useful, as it connects our ML estimation formulations directly with other important formulations in visual field reconstruction such as detection of discontinuities [17, 29].

3.4 Conclusion

A wide range of single-frame visual field reconstruction problems can be formulated in terms of spatial differential constraints and linear measurement constraints. A vectorized version of such a problem can be considered as an ML estimation problem. An implicit representation for the statistical data provides us with a convenient framework to deal with this particular ML estimation problem, both theoretically and numerically. The statistical interpretation provided by the ML formulation motivates us to approach the multi-frame visual reconstruction problem as a stochastic process estimation problem.

Chapter 4

A Filtering Solution to Space-Time Reconstruction Problems

Reconstructing visual fields by processing multiple sets of measurements has an obvious advantage over reconstruction based on a single set, as accumulation of a larger quantity of data leads to a more reliable estimate due to reduction in measurement noise. An advantage not as obvious is that in some cases spatial coherence may not provide sufficient constraints to yield an adequate estimate, and hence temporal information must be utilized as well. For exmple, in optical flow estimation, if the measured spatial gradients have the identical directions over the entire image frame, the resulting aperture problem can be resolved only by incorporating more frames of measurements. In this chapter, we describe an extension of the single-frame reconstruction problem in which we consider fusing multiple frames of measurements obtained from images arriving sequentially over time.

We formulate multi-frame visual field reconstruction problems in an estimation theoretic framework. As described in the previous chapter, the single-frame problem can be formulated as a maximum likelihood estimation problem, so that the computed visual field can be considered as a jointly Gaussian random field. This allows us to treat a sequence of unknown fields indexed by time t, $\mathbf{f}(t)$, as a space-time stochastic process, by capturing the time evolution of the field probabilistically. We can then utilize well-developed sequential estimation algorithms, i.e., Kalman filter and its derivatives.

To use the Kalman filter algorithm, the space-time behavior of the unknown filed as well as the measurement processes must be expressed in a state-space dynamic system format. Gauss-Markov system models for $\mathbf{f}(t)$ and the corresponding Kalman filtering solutions have been proposed [55, 71, 26, 27, 69]. These methods, however, deal with computational complexity associated with Kalman filters by rather *ad hoc* approximate methods. We approach such a computational issue more systematically, leading to near-optimal methods. Moreover, we develop a more general formulation than the traditional Gauss-Markov models by temporal extensions of the spatial coherence principle. Specifically, we examine several potentially useful temporal coherence constraints and show that the most natural format to express the dynamics is the *descriptor system*[1] format [49, 50, 46].

We use the information form of Kalman filter, or *information filter*, in which the information pair is propagated over time instead of the mean-covariance pair. The information filter equations [4, 45] are algebraically equivalent to the standard, mean-covariance Kalman filter equations. The structure of the particular filtering problem governs the choice between the two sets of equations [40, 8]. For visual reconstruction, the information filter is a natural choice because of the possibilities that the information matrices can become singular at various stages of the estimation process. By viewing the descriptor dynamic equation as an ML observation equation, we obtain an associated information pair. From this perspective, we can treat the filtering problem as a sequential fusing of information pairs corresponding to the temporal/spatial coherence and measurement equations.

A more important reason to use the information pairs in our filtering algorithms is computational efficiency. The large dimensions of the unknown visual fields make conservation of memory space as well as reduction of the number of required algebraic operations essential. As we will see, information pairs allow us to express the statistics of the unknowns in a spatially concentrated format, which we exploit in the development of efficient approximate filtering algorithms. In this chapter, we present dynamic system models for multi-frame visual field reconstruction and derive, based

[1] also called *singular system*

on these models, optimal Kalman filtering algorithms, including the information filter and *square root information filter* which has well documented numerical stabilities. Development of the filtering algorithms continues into the next chapter where we discuss approximation issues.

4.1 Temporal Dynamic Equations

In this section we consider a general form of discrete dynamic equations that capture temporal behavior of visual fields. We seek a general dynamic expression for temporal stochastic processes suitable for visual fields. The dynamic equations are ideally based on some physical laws that govern the spatio-temporal behavior of the particular visual field of interest; however, most of the time we cannot assume such exogeneous, exact knowledge about the field to be available *a priori*. In fact, a major purpose of the "low-level" visual processing such as visual field reconstruction is to organize the raw visual data in a way useful to infer such hypotheses as what we are looking at and how exactly it is behaving over time and space. With this purpose in mind, the spatial coherence constraints have been utilized in single-frame visual field reconstruction problems. In this section, we examine their straightforward extensions over the time axis — *temporal coherence constraints* [23], consisting of cost terms of temporal derivatives. Specifically, consider the following temporal extension of the general single-frame visual reconstruction problem (3.1) from the last chapter:

$$\min_f \int_0^\tau \int_{\mathcal{D}} \nu \, \|g - hf\|^2 \;+\; \sum_{\mathbf{i}} \mu_{\mathbf{i}} \left\| \frac{\partial^{j_1}}{\partial {s_1}^{j_1}} \frac{\partial^{j_2}}{\partial {s_2}^{j_2}} \cdots \frac{\partial^{j_K}}{\partial {s_K}^{j_K}} f \right\|^2 + \sum_{\mathbf{i},n} \rho_{\mathbf{i},n} \left\| \frac{\partial^{j_1}}{\partial {s_1}^{j_1}} \frac{\partial^{j_2}}{\partial {s_2}^{j_2}} \cdots \frac{\partial^{j_K}}{\partial {s_K}^{j_K}} \frac{\partial^n}{\partial t^n} f \right\|^2 d\mathbf{s}\, dt \qquad (4.1)$$

where $f(\mathbf{s},t), g(\mathbf{s},t), h(\mathbf{s},t)$, and $\nu(\mathbf{s},t)$ are now space-time functions. We will examine some specific forms of the temporal coherence constraints and how they might be useful for multi-frame visual field reconstruction. At the same time, we present how each of these constraints can be represented as a discrete dynamic equation driven

by a white noise process. This leads to a general discrete dynamic representation for stochastic processes modeling various types of temporal behavior of the visual fields.

4.1.1 Gauss-Markov processes for $\mathbf{f}(t)$

In single-frame reconstruction problems, the *first order* differential constraint terms such as those used in the membrane model promote continuity of the field over space. An analogous constraint term over time is $\rho \left\| \frac{\partial}{\partial t} f(s,t) \right\|^2$ which promotes continuity of the field over time by penalizing rapid changes in time. An equivalent expression for this constraint is the dynamic model $\frac{\partial}{\partial t} f(s,t) = q(s,t)$ where $q(s,t)$ is a *zero-mean* space-time white noise with covariance $\rho^{-1} I$. Via discretization and vectorization (cf. Sec. 3.2.1), the first order temporal differential constraint thus leads to a discrete Gauss-Markov process model

$$\mathbf{f}(t) = \mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(0, \rho^{-1}\mathbf{I} \right).$$

A more general model can be formulated by incorporation of a time-varying *system matrix* $\mathbf{A}(t)$ as

$$\mathbf{f}(t) = \mathbf{A}(t)\mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(0, \rho^{-1}\mathbf{I} \right).$$

In some reconstruction problems the system matrix $\mathbf{A}(t)$ plays an important role of registering moving visual field onto the image frame — a fundamental issue in multi-frame visual reconstruction problems. That is, unlike many sequential estimation (and control) problems where there is no ambiguity in which estimates should be matched with which measurements, sequential multi-frame visual field reconstruction problem poses the *data-estimate registration* problem which calls for matching of estimates in the current frame with the measurements obtained from an image of the moving objects in the next frame. In essence, the matrix $\mathbf{A}(t)$ provides the system model with the information on how the estimated field from time $t-1$ should be warped in order to fit into the image frame at time t. For example, in multi-frame

estimation of dynamically evolving depth fields, Heel [25, 26, 27, 28] has proposed a method to adaptively construct $\mathbf{A}(t)$ based on the knowledge of relative motion between the imaged objects and the image frame. Also, in Chapter 6 we will show that for multi-frame optical flow estimation there is a differential approach to obtaining a system matrix $\mathbf{A}(t)$ to perform data-estimate registration. Here, it suffices to indicate that the matrix $\mathbf{A}(t)$ essentially performs some local position adjustments of the components of the field via shifting and averaging and that the matrix usually has a sparse structure in which the non-zero elements are concentrated around the main diagonal.

In summary, the following Gauss-Markov model allows us to capture first-order differential behavior of time-varying visual fields:

$$\mathbf{f}(t) = \mathbf{A}(t)\mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim (\ 0, \mathbf{Q}(t)\ ) \tag{4.2}$$

where $\mathbf{A}(t)$ is sparse and $\mathbf{Q}(t)$ is positive definite. This dynamic equation in conjunction with an observation equation for $\mathbf{f}(t)$ at each t forms a dynamic system whose state is $\mathbf{f}(t)$. As standard discrete Kalman filtering algorithms can be used directly to solve the reconstruction problems, first order differential constraints and Gauss-Markov modeling have been popular in multi-frame reconstruction problems [55, 71, 27, 10, 5, 69]. In a later section, however, we expand the notion of Gauss-Markov stochastic processes (specifically by allowing $\mathbf{Q}(t)$ to be "infinite") so that this modeling format and the corresponding estimation algorithms, i.e., the Kalman filter, are applicable to a wider range of temporal dynamics.

4.1.2 Autoregressive temporal models

Although direct applicability of the Kalman filtering algorithms makes Gauss-Markov modeling convenient, it does not mean that the first order temporal differential constraints form the only conceivable basis for temporal dynamic models. Neither does it mean that the other types of constraints do not lead to efficient sequential estimation algorithms like the Kalman filter.

In spatial coherence constraints, second derivatives like those in the thin-plate model are frequent. Such higher order derivatives also make sense for temporal coherence constraints. For example, consider a motion field generated by planar motion of a surface such that the motion vectors display a constant acceleration such as in planar rotation. In this case, a temporal coherence like $\frac{\partial^2}{\partial t^2} f(\mathbf{s},t) = q(\mathbf{s},t)$ (where $q(\mathbf{s},t)$ is a zero-mean white noise as defined above) makes sense, leading to a discrete vectorized dynamic model

$$\mathbf{f}(t+1) = 2\mathbf{f}(t) - \mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(0, \rho^{-1}\mathbf{I} \right).$$

This type of model has been used in the computation of optical flow [20]. In general for an n^{th} order derivative, the dynamic model can be written as

$$\mathbf{f}(t) = \sum_{j=1}^{n} a_j \mathbf{f}(t-j) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(0, \rho^{-1}\mathbf{I} \right)$$

with appropriate coefficients a_j's. With the data-estimates registration problem discussed above in mind, we can further generalize the model as

$$\mathbf{f}(t) = \sum_{j=1}^{n} \mathbf{A}_j(t)\mathbf{f}(t-j) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim (0, \mathbf{Q}(t)) \tag{4.3}$$

for $n \geq 1$ where $\mathbf{A}_j(t)$ and $\mathbf{Q}(t)$ are sparse matrices and $\mathbf{Q}(t)$ is positive-definite.

An autoregressive dynamic equation can be written as a Gauss-Markov process with an augmented state, i.e., using the state vector

$$\mathbf{x}(t) \equiv \left[\mathbf{f}^T(t), \mathbf{f}^T(t-1), \ldots, \mathbf{f}^T(t-n+1) \right]^T$$

for the model above. Thus, Kalman filtering algorithms are still applicable but with a substantial increase in computational requirements due to the increase in the state dimension.

4.1.3 Descriptor dynamic models

The driving noise process $\mathbf{q}(t)$ in the Gauss-Markov model(4.2) represents the variation in the field from one frame to the next. It is reasonable to impose spatial coherence on this variation, reflecting the fact that such variations have local correlations. Mathematically, such an idea is represented in the least-squares formulation as cross space-time partial differential constraints. In another case, we might have a situation where the temporal behavior of the field is best expressed by two or more temporal differential constraints imposed *simultaneously.* These two types of situations can be expressed as *descriptor dynamic equations.* Let us illustrate these for some specific cases.

- *motivation 1:*

  Consider the following temporal coherence constraint for the 1-D optical flow field:

$$\frac{\partial}{\partial s}\frac{\partial}{\partial t}f(s,t) = q(s,t)$$

  where $q(s,t)$ is a white noise process. This model requires the temporal variation in the optical flow field $\mathbf{f}(t)$ along the contour to be spatially smooth. This model makes sense because the contours are often associated with boundaries of object surfaces which are assumed (under various forms of spatial coherence constraint) to display stiffness and smoothness. A discrete, vector version of the model is

$$\mathbf{S}(\,\mathbf{f}(t) - \mathbf{f}(t-1)\,) = \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(\,0, \rho^{-1}\mathbf{I}\,\right)$$

  or

$$\mathbf{S}\mathbf{f}(t) = \mathbf{S}\mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(\,0, \rho^{-1}\mathbf{I}\,\right)$$

where $\mathbf{S}$ is a first-order spatial difference operator. The latter dynamic equation is in the standard descriptor form.

- *motivation 2:*

  For a similar constraint imposed on a 2-D optical flow, we have a simultaneous set of constraints:

$$\frac{\partial}{\partial s_1}\frac{\partial}{\partial t}f(\mathbf{s},t) = q_1(\mathbf{s},t)$$
$$\frac{\partial}{\partial s_2}\frac{\partial}{\partial t}f(\mathbf{s},t) = q_2(\mathbf{s},t)$$

  where $q_1(\mathbf{s},t)$ and $q_2(\mathbf{s},t)$ are white noise with covariance matrices $\rho_1^{-1}I$ and $\rho_2^{-1}I$, respectively. This corresponds to applying the membrane model to the first-order temporal variation of the field $\frac{\partial}{\partial t}f(\mathbf{s},t)$. In our vectorized form, this leads to a descriptor dynamic equation:

$$\begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \mathbf{f}(t) = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(0, \begin{bmatrix} \rho_1^{-1}\mathbf{I} & \\ & \rho_2^{-1}\mathbf{I} \end{bmatrix} \right).$$

- *motivation 3:*

  Another case where we might have multiple temporal coherence constraints is a temporal generalization of the deformable contour model used in [41], i.e., to use the first and second order temporal derivative constraints to promote continuity and linearity, respectively, in temporal variations of the field:

$$\frac{\partial}{\partial t}f(s,t) = q_3(s,t)$$
$$\frac{\partial^2}{\partial t^2}f(s,t) = q_4(s,t)$$

where the zero-mean white noise processes $q_3(s,t)$ and $q_4(s,t)$ have covariance $\rho_3^{-1}I$ and $\rho_4^{-1}I$, respectively. A vectorized form of this model is

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \mathbf{f}(t) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ 2\mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}(t-1) \\ \mathbf{f}(t-2) \end{bmatrix} + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(0, \begin{bmatrix} \rho_3^{-1}\mathbf{I} & \\ & \rho_4^{-1}\mathbf{I} \end{bmatrix} \right).$$

This equation is not in a standard descriptor form. We can write it in the standard form using a driving noise with a singular covariance matrix:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}(t) \\ \mathbf{f}(t-1) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ 2\mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{f}(t-1) \\ \mathbf{f}(t-2) \end{bmatrix} + \mathbf{q}(t),$$

$$\mathbf{q}(t) \sim \left(0, \begin{bmatrix} \rho_3^{-1}\mathbf{I} & & \\ & \rho_4^{-1}\mathbf{I} & \\ & & \mathbf{0} \end{bmatrix} \right).$$

- *generalization:*

  For general space-time coherence constraints, we must consider a set of simultaneous autoregressive models driven by white noise processes. Let $\mathbf{S}_i$ for $i = 1, 2, \ldots, m$ be various spatial difference operators[2]. Then, the following dynamic equations can express all the temporal models discussed so far:

$$\mathbf{S}_i\mathbf{f}(t) = \sum_{j=1}^{n_i} \mathbf{S}_i\mathbf{A}_{ij}(t)\mathbf{f}(t-j) + \mathbf{q}_i(t), \quad \mathbf{q}_i(t) \sim (\ 0, \mathbf{Q}_i(t)\ ) \tag{4.4}$$

  where $\mathbf{A}_{ij}(t)$ performs data-estimate registration and is a sparse matrix, and $\mathbf{Q}_i(t)$ is sparse and positive definite.

Dynamic equation for multi-frame reconstruction

The set of dynamic equations (4.4) can express all the dynamics discussed in this section, including (4.2) and (4.3). By defining $n \equiv \max_i n_i$, we can write them in a

[2]including the identity matrix which can be considered to be a zero-th order difference operator

single vectorized equation:

$$\begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_m \end{bmatrix} \mathbf{f}(t) = \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_m \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1n} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_{m1} & \mathbf{A}_{m2} & \cdots & \mathbf{A}_{mn} \end{bmatrix} \begin{bmatrix} \mathbf{f}(t-1) \\ \vdots \\ \mathbf{f}(t-n) \end{bmatrix} + \begin{bmatrix} \mathbf{q}_1(t) \\ \vdots \\ \mathbf{q}_m(t) \end{bmatrix} \tag{4.5}$$

where $\mathbf{A}_{ij} \equiv \mathbf{A}_{ij}(t)$, some of which are zero. This is the dynamic equation on which we base our development of a sequential estimation algorithm. For brevity we write the equation as

$$\mathbf{B}\mathbf{f}(t) = \mathbf{B}\mathbf{A}(t)\mathbf{x}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim (\ 0, \mathbf{Q}(t)\ ) \tag{4.6}$$

by introducing the correspondingly defined matrices $\mathbf{B}, \mathbf{A}(t), \mathbf{x}(t-1)$, and $\mathbf{q}(t)$ as well as $\mathbf{Q}(t)$ which is a block diagonal matrix whose blocks are $\mathbf{Q}_i(t)$'s.

Note that we can certainly write (4.6) in a standard descriptor form

$$\mathbf{E}'\mathbf{x}(t) = \mathbf{A}'(t)\mathbf{x}(t-1) + \mathbf{q}'(t), \quad \mathbf{q}'(t) \sim (\ 0, \mathbf{Q}'(t)\ ) \tag{4.7}$$

with $\mathbf{x}(t) \equiv \left[\mathbf{f}^T(t), \ldots, \mathbf{f}^T(t-n+1)\right]$ and appropriately defined matrices $\mathbf{E}'$, $\mathbf{A}'(t)$, $\mathbf{q}'(t)$ and $\mathbf{Q}'(t)$. Although sequential estimation algorithms have been developed for standard descriptor systems like (4.7) [62], the only truly dynamic part of (4.7) is (4.6) while the rest of (4.7) just involves transfering of the components $\mathbf{f}(t-1), \mathbf{f}(t-2), \ldots, \mathbf{f}(t-n)$ of $\mathbf{x}(t-1)$ to their appropriate positions in $\mathbf{x}(t)$. In our development of sequential information filter, we regard dynamic equations as observation equations for a set of unknowns at different times. Treating (4.7) as an observation equation involves in essence a set of perfect measurements corresponding to the trivial portion of the dynamics, which lead to a *singular* ML estimation problem which cannot be conveniently dealt with in information form. For these reasons we express the general temporal dynamics for the visual field process $\mathbf{f}(t)$ using (4.6), which captures the only non-trivial portion of the dynamics, thereby avoiding the unnecessary complication caused by the trivial shifting portion of the dynamics.

4.2 Information Filter for the General Dynamic System

Consider state estimation problem for the descriptor dynamic system

$$\mathbf{B}\mathbf{f}(t) = \mathbf{B}\mathbf{A}(t)\mathbf{x}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim (\, 0, \mathbf{Q}(t)\,) \tag{4.8}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{f}(t) + \mathbf{r}(t), \quad \mathbf{r}(t) \sim (\, 0, \mathbf{R}(t)\,) \tag{4.9}$$

where the dynamic equation is from the end of the last section and the measurement equation corresponds to the constraints in a single-frame reconstruction such as the "optical flow observation equation" in Section 3.2.4. The state of the system is

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{f}(t) \\ \vdots \\ \mathbf{f}(t-n+1) \end{bmatrix}. \tag{4.10}$$

In this section we present an information filter for the system, i.e., an algorithm to propagate the information pairs associated with $\mathbf{x}(t)$ and $\mathbf{f}(t)$ through the equations (4.8) and (4.9) over time.

4.2.1 Sequential ML estimation

As mentioned before, the dynamic system (4.8)(4.9) is a descriptor system. In this subsection we present how the explicit statistics, the mean-covariance pair, of the state can be obtained sequentially over time in a general descriptor system. Extending the approach presented here, we then design an information filter (which propagate implicit statistics) for our dynamic system in later subsections.

Propagation of mean-covariance pair

Nikoukah *et al.* [62] present a general mechanism for generating ML estimates recursively. In their work, previously estimated variables are replaced by "summary

measurements", and irrelevant variables are discarded. This procedure constitutes the basis for the descriptor Kalman filter. Suppose we have a descriptor system

$$\mathbf{E}'(t)\mathbf{x}(t) = \mathbf{A}'(t)\mathbf{x}(t-1) + \mathbf{q}'(t), \quad \mathbf{q}'(t) \sim (0, \mathbf{Q}'(t)) \tag{4.11}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{r}(t), \quad \mathbf{r}(t) \sim (0, \mathbf{R}(t)) . \tag{4.12}$$

At time t, given the state statistics from the *previous* frame

$$\mathbf{x}(t-1) \sim \left(\hat{\mathbf{x}}(t-1), \hat{\mathbf{P}}(t-1) \right) \tag{4.13}$$

we want to propagate it through the dynamic equation (4.11) to obtain the one-step *predicted* statistics $\mathbf{x}(t) = \left(\overline{\mathbf{x}}(t), \overline{\mathbf{P}}(t) \right)$ and then through the measurement equation (4.12) to obtain the *updated* statistics $\mathbf{x}(t) = \left(\hat{\mathbf{x}}(t), \hat{\mathbf{P}}(t) \right)$. The approach taken by Nikoukah *et al.* (and inherited in this thesis) is to view the prediction step as an ML estimation problem. Namely, the mean-covariance pair (4.13) implies the observation equation

$$\hat{\mathbf{x}}(t-1) = \mathbf{x}(t-1) + \mathbf{u}(t-1), \quad \mathbf{u}(t-1) \sim \left(0, \hat{\mathbf{P}}(t-1) \right) \tag{4.14}$$

and the dynamic equation (4.11) can be considered to be a *joint observation* of $\mathbf{x}(t-1)$ and $\mathbf{x}(t)$ as well, so that (4.11) and (4.14) can be written as a single observation equation

$$\begin{bmatrix} 0 \\ \hat{\mathbf{x}}(t-1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}'(t) & -\mathbf{E}'(t) \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t-1) \\ \mathbf{x}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{q}'(t) \\ \mathbf{u}(t-1) \end{bmatrix} .$$

Assuming that $\mathbf{E}'(t)$ has a full column rank, this observation equation yields the mean-covariance pair

$$\begin{bmatrix} \mathbf{x}(t-1) \\ \mathbf{x}(t) \end{bmatrix} \sim \left(\begin{bmatrix} \overline{\mathbf{x}}(t-1) \\ \overline{\mathbf{x}}(t) \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \right) \tag{4.15}$$

where $\dim(\overline{\mathbf{x}}(t-1)) = \dim(\mathbf{x}(t-1))$ and $\dim(\mathbf{P}_{11}) = \dim\left(\mathbf{x}(t)\mathbf{x}^T(t)\right)$. This mean-covariance pair specifies the joint Gaussian density for $\mathbf{x}(t-1)$ and $\mathbf{x}(t)$. The marginal densities are simply

$$\mathbf{x}(t-1) \sim (\overline{\mathbf{x}}(t-1), \mathbf{P}_{11}) \tag{4.16}$$

$$\mathbf{x}(t) \sim (\overline{\mathbf{x}}(t), \mathbf{P}_{22}). \tag{4.17}$$

The desired predicted statistics are obtained as (4.17).

In the next step, the update step, we want to update the statistics (4.17) using the measurement equation (4.12). The following facts are fairly intuitive but rigorously proven in [62] based on the ML estimation theory:

1. The estimate, i.e., the posterior mean-covariance pair, of $\mathbf{x}(t)$ based on (4.15) and (4.12) is the same as that based on (4.16), (4.17), and (4.12). This means that for the purpose of estimating $\mathbf{x}(t)$ the mean-covariance pairs in (4.16) and (4.17) summarize the information provided by the results from the previous time frames (4.13) and the dynamic equation (4.11).

2. Further, estimating $\mathbf{x}(t)$ based on (4.17) and (4.12) yields the same result. That is, since (4.16) does not involve $\mathbf{x}(t)$ it can be discarded for the purpose of estimating $\mathbf{x}(t)$.

The marginal density (4.17) implies the observation equation

$$\overline{\mathbf{x}}(t) = \mathbf{x}(t) + \overline{\mathbf{u}}(t), \quad \overline{\mathbf{u}}(t) \sim (0, \mathbf{P}_{22}).$$

This equation and (4.12) can be written in a single observation equation

$$\begin{bmatrix} \overline{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{C}(t) \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \overline{\mathbf{u}}(t) \\ \mathbf{r}(t) \end{bmatrix}$$

from which updated mean-covariance pair can be computed.

Marginal information pair

Obtaining the marginal Gaussian densities (4.16)(4.17) from the joint density (4.15) can be considered to be a key step in the preceding sequential estimation algorithm in the sense that this step separates the "past" from the "present". Although marginal mean-covariance pairs are easily obtained from the joint mean-covariance pair by taking appropriate subvectors and submatrices out of the mean vector and the covariance matrix, an analogous process is a little more complicated for information pairs.

Suppose that you have a Gaussian random vector $\mathbf{x} \sim (\, \mathbf{m}, \mathbf{P} \,)$. Let us partition $\mathbf{x}$ into two subvectors $\mathbf{x}_1$ and $\mathbf{x}_2$ and consider obtaining marginal densities for them. We partition $\mathbf{m}$ and $\mathbf{P}$ as

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \left(\begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \right) \tag{4.18}$$

from which we can obtain the marginal densities for $\mathbf{x}_1$ and $\mathbf{x}_2$ as

$$\mathbf{x}_1 \sim (\, \mathbf{m}_1, \mathbf{P}_{11} \,), \quad \mathbf{x}_2 \sim (\, \mathbf{m}_2, \mathbf{P}_{22} \,). \tag{4.19}$$

We want to do the same for the information pair: given $\mathbf{x} \sim \langle\langle\, \mathbf{z}, \mathbf{L} \,\rangle\rangle$ we want to obtain the corresponding *marginal information pairs* $\mathbf{x}_1 \sim \langle\langle\, \mathbf{b}_1, \mathbf{K}_1 \,\rangle\rangle$ and $\mathbf{x}_2 \sim \langle\langle\, \mathbf{b}_2, \mathbf{K}_2 \,\rangle\rangle$. From the relationship between mean-covariance and information pairs, such marginal information pairs must satisfy

$$\mathbf{K}_1 = \mathbf{P}_{11}^{-1}, \quad \mathbf{K}_2 = \mathbf{P}_{22}^{-1} \tag{4.20}$$

and

$$\mathbf{b}_1 = \mathbf{K}_1 \mathbf{m}_1 \quad \mathbf{b}_2 = \mathbf{K}_2 \mathbf{m}_2. \tag{4.21}$$

Computationally, the marginal information pairs can be obtained as follows:

Lemma 4.1 *Given the joint information pair* $\mathbf{x} \sim \langle\langle\, \mathbf{z}, \mathbf{L} \,\rangle\rangle$, *consider the following*

partitioning:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \sim \left\langle\!\left\langle \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \right\rangle\!\right\rangle$$

where $\dim(\mathbf{z}_1) = \dim(\mathbf{x}_1)$ *and* $\dim(\mathbf{L}_{11}) = \dim(\mathbf{x}_1\mathbf{x}_1^T)$. *Then, if* $\mathbf{L}_{22}$ *is invertible, the information pair for* $\mathbf{x}_1$, $\langle\langle\ \mathbf{b}_1, \mathbf{K}_1\ \rangle\rangle$, *can be obtained as*

$$\mathbf{b}_1 = \mathbf{z}_1 - \mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{z}_2, \quad \mathbf{K}_1 = \mathbf{L}_{11} - \mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21}.$$

Also, if $\mathbf{L}_{11}$ *is invertible, the marginal information pair for* $\mathbf{x}_2$, $\langle\langle\ \mathbf{b}_2, \mathbf{K}_2\ \rangle\rangle$, *can be obtained as*

$$\mathbf{b}_2 = \mathbf{z}_2 - \mathbf{L}_{21}\mathbf{L}_{11}^{-1}\mathbf{z}_1, \quad \mathbf{K}_2 = \mathbf{L}_{22} - \mathbf{L}_{21}\mathbf{L}_{11}^{-1}\mathbf{L}_{12}.$$

Proof: We prove this result in the more restrictive case when the joint information matrix $\mathbf{L}$ is invertible. Then, by definition $\mathbf{P} = \mathbf{L}^{-1}$; thus, we have

$$\begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix}^{-1}$$

where $\dim(\mathbf{P}_{11}) = \dim(\mathbf{L}_{11})$ and from the block matrix inversion formulas in Appendix A we have

$$\mathbf{P}_{11}^{-1} = \mathbf{L}_{11} - \mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21} = \mathbf{K}_1$$
$$\mathbf{P}_{22}^{-1} = \mathbf{L}_{22} - \mathbf{L}_{21}\mathbf{L}_{11}^{-1}\mathbf{L}_{12} = \mathbf{K}_2.$$

Thus, (4.20) is satisfied. By definition, $\mathbf{m} = \mathbf{L}^{-1}\mathbf{z} = \mathbf{P}\mathbf{z}$; thus, we have

$$\mathbf{m}_1 = \mathbf{P}_{11}\mathbf{z}_1 + \mathbf{P}_{12}\mathbf{z}_2$$
$$\mathbf{m}_2 = \mathbf{P}_{21}\mathbf{z}_1 + \mathbf{P}_{22}\mathbf{z}_2.$$

But, again by the block matrix inversion formulas,

$$\mathbf{K}_1\mathbf{m}_1 = \mathbf{P}_{11}^{-1}\mathbf{m}_1 = \mathbf{z}_1 + \mathbf{P}_{11}^{-1}\mathbf{P}_{12}\mathbf{z}_2 = \mathbf{z}_1 - \mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{z}_2 = \mathbf{b}_1$$
$$\mathbf{K}_2\mathbf{m}_2 = \mathbf{P}_{22}^{-1}\mathbf{m}_2 = \mathbf{P}_{22}^{-1}\mathbf{P}_{21}\mathbf{z}_1 + \mathbf{z}_2 = \mathbf{z}_2 - \mathbf{L}_{21}\mathbf{L}_{11}^{-1}\mathbf{z}_1 = \mathbf{b}_2$$

Thus, (4.21) is satisfied.

Note that if only the information pair for $\mathbf{x}_1$ is desired we only have to require the submatrix $\mathbf{L}_{22}$ to be invertible, not the entire matrix $\mathbf{L}$. Similarly, to compute the information pair for $\mathbf{x}_2$, only $\mathbf{L}_{11}$ has to be inverted. We refer the readers to [62] for a more rigorous treatment of such matrix invertibility issues. □

4.2.2 Prediction step

Let us now begin derivation of an information filter for the visual reconstruction dynamic system (4.8) (4.9). Given the information pair from the previous frame

$$\mathbf{x}(t-1) \sim \left\langle\left\langle\, \hat{\mathbf{z}}(t-1), \hat{\mathbf{L}}(t-1) \,\right\rangle\right\rangle \tag{4.22}$$

we want find the *predicted information pair*

$$\mathbf{x}(t) \sim \left\langle\left\langle\, \bar{\mathbf{z}}(t), \bar{\mathbf{L}}(t) \,\right\rangle\right\rangle . \tag{4.23}$$

For the multi-frame reconstruction problem to be well-posed, the updated information matrix at each t must be non-singular. Let us assume such is the case, and, in particular, $\hat{\mathbf{L}}(t-1) > \mathbf{0}$. We will see, however, that the predicted information matrices may become singular even in a well-posed problem.

Let us define a vector ξ as

$$\xi \equiv \begin{bmatrix} \mathbf{f}(t) \\ \mathbf{f}(t-1) \\ \vdots \\ \mathbf{f}(t-n) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(t) \\ \mathbf{x}(t-1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{f}(t-n) \end{bmatrix}, \tag{4.24}$$

and let $\mathbf{u}(t-1)$ be the estimation error associated with $\hat{\mathbf{x}}(t-1)$, i.e.,

$$\mathbf{u}(t-1) \equiv \mathbf{x}(t-1) - \hat{\mathbf{x}}(t-1), \tag{4.25}$$

whose distribution is given by the mean-covariance pair $\left(0, \hat{\mathbf{L}}^{-1}(t-1)\right)$. Then, the prediction step can be considered to be the ML problem to esimate ξ given the following observation equation which is essentially a combination of (4.8) and (4.25):

$$\begin{bmatrix} 0 \\ \hat{\mathbf{x}}(t-1) \end{bmatrix} = \begin{bmatrix} -\mathbf{B} & \mathbf{B}\mathbf{A}(t) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \xi + \begin{bmatrix} \mathbf{q}(t) \\ \mathbf{u}(t-1) \end{bmatrix}, \tag{4.26}$$

$$\begin{bmatrix} \mathbf{q}(t) \\ \mathbf{u}(t-1) \end{bmatrix} \sim \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{Q}(t) & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{L}}^{-1}(t-1) \end{bmatrix} \right).$$

The implicit solution for this ML estimation problem, i.e., the information pair for ξ is given as $\xi \sim \langle\langle\, \eta, \Xi \,\rangle\rangle$ where

$$\eta \equiv \begin{bmatrix} 0 \\ \hat{\mathbf{z}}(t-1) \end{bmatrix}, \quad \Xi \equiv \begin{bmatrix} \mathbf{U} & -\mathbf{U}\mathbf{A} \\ -\mathbf{A}^T\mathbf{U} & \mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1) \end{bmatrix} \tag{4.27}$$

in which $\mathbf{U}$ and $\mathbf{A}$ are defined as

$$\mathbf{U} \equiv \mathbf{B}^T\mathbf{Q}^{-1}(t)\mathbf{B}, \quad \mathbf{A} \equiv \mathbf{A}(t). \tag{4.28}$$

Since $\mathbf{x}(t)$ is a subvector of ξ, we can use Lemma 4.1 to obtain its information pair. To obtain the marginal information pair, let us partition the matrix Ξ and vector η as

$$\Xi = \begin{bmatrix} \Xi_{11} & \Xi_{12} \\ \Xi_{21} & \Xi_{22} \end{bmatrix}, \quad \eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \tag{4.29}$$

so that $\dim(\Xi_{11}) = \dim(\mathbf{x}(t)\mathbf{x}^T(t))$, $\dim(\Xi_{22}) = \dim(\mathbf{f}(t)\mathbf{f}^T(t))$, and $\dim(\eta_1) =$

$\dim(\mathbf{x}(t))$. Then, using Lemma 4.1,

$$\overline{\mathbf{L}}(t) = \Xi_{11} - \Xi_{12}\Xi_{22}^{-1}\Xi_{21} \tag{4.30}$$

$$\overline{\mathbf{z}}(t) = \eta_1 - \Xi_{12}\Xi_{22}^{-1}\eta_2. \tag{4.31}$$

Note: to compute the desired information pair, the only matrix required to be invertible is Ξ_{22}. Thus, it is sufficient for us to check that $\hat{\mathbf{L}}(t-1)$ is non-singular to be able to compute the desired information pair, as Ξ_{22} is a block submatrix along the diagonal of the matrix $\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)$. The resulting predicted information matrix $\overline{\mathbf{L}}(t)$ is, however, not guaranteed to be non-singular. In fact, it is a singular matrix when $\mathbf{U}$ is singular. An important role of the update step, presented below, is to make certain that the resulting information matrix is non-singular, i.e., a possibly singular $\overline{\mathbf{L}}(t)$ must be converted to a non-singular $\hat{\mathbf{L}}(t)$.

4.2.3 Update step

In the update step, the predicted information pair obtained as above is fused with the information pair associated with the observation equation derived from the spatial coherence and measurement constraints applied to $\mathbf{f}(t)$, as expressed in (4.9) which by itself corresponds to a single-frame estimation problem for the field $\mathbf{f}(t)$. Then, the resulting updated information pair for $\mathbf{x}(t)$ is used to compute the estimate $\hat{\mathbf{f}}(t)$ of the field at time t. Rewriting (4.9) in terms of $\mathbf{x}(t)$ instead of $\mathbf{f}(t)$, the observation equation at time t has the form

$$\mathbf{y}(t) = [\mathbf{C}(t), \mathbf{0}, \ldots, \mathbf{0}]\,\mathbf{x}(t) + \mathbf{r}(t) \tag{4.32}$$

as only $\mathbf{f}(t)$ is observed. Based on the additive property (cf. Theorem 3.4, p.50), the information pair is updated as

$$\widehat{\mathbf{L}}(t) = \overline{\mathbf{L}}(t) + \begin{bmatrix} \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} & & & \\ & \mathbf{0} & & \\ & & \ddots & \\ & & & \mathbf{0} \end{bmatrix} \tag{4.33}$$

$$\widehat{\mathbf{z}}(t) = \overline{\mathbf{z}}(t) + \begin{bmatrix} \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y}(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.34}$$

As discussed in Chapter 3, we assume that the single-frame estimation problem associated with (4.9) well-posed, i.e., $\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}$ is non-singular. This assumption implies that $\widehat{\mathbf{L}}(t)$ as obtained above is a full-rank (invertible) matrix. This can be seen by noting that $\overline{\mathbf{L}}(t)$ might be singular due to rank-deficiency in the dynamic equation (4.8). Let $N \equiv \dim(\mathbf{f}(t))$. Then, since (4.8) is an N-dimensional vector equation, the rank-*deficiency* in $\overline{\mathbf{L}}(t)$ must be at most N. In fact, a careful inspection of (4.27), (4.29), and (4.30) shows that the rank-deficient part of $\overline{\mathbf{L}}(t)$ is the upper-left $N \times N$ block. The update equation for the information matrix (4.33), then, converts this block to be full rank. Thus, $\widehat{\mathbf{L}}(t)$ is non-singular if $\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}$ is non-singular.

To obtain the estimate $\widehat{\mathbf{f}}(t)$ we need to to compute the information pair

$$\mathbf{f}(t) \sim \left\langle\left\langle\, \hat{\mathbf{b}}(t), \widehat{\mathbf{K}}(t) \,\right\rangle\right\rangle.$$

Since $\mathbf{f}(t)$ is a subvector of $\mathbf{x}(t)$, we can again use Lemma 4.1 to obtain $\hat{\mathbf{b}}(t)$ and $\widehat{\mathbf{K}}(t)$. As in the solution for single-frame problems, the estimate is then obtained by solving the inverse problem

$$\widehat{\mathbf{K}}(t)\widehat{\mathbf{f}}(t) = \hat{\mathbf{b}}(t). \tag{4.35}$$

Recall that in the single-frame problems such an inverse problem can be solved efficiently by a variety of iterative methods due to the sparse structure of the information matrix to be inverted. For multi-frame problems, the updated information matrix $\widehat{\mathbf{K}}(t)$ does not have such a sparse structure in general. In the next chapter, however, we present techniques to approximate the information matrix by a sparse matrix having a structure similar to those found in the information matrices associated with the single-frame problems.

4.2.4 Summary of the information filter algorithm

- *Prediction step:*

  Given the information pair $\mathbf{x}(t-1) \sim \left\langle\left\langle\, \hat{\mathbf{z}}(t-1), \hat{\mathbf{L}}(t-1)\, \right\rangle\right\rangle$, form the following vector and matrix; then, partition them with η_i's and Ξ_{ij}'s as

$$\begin{bmatrix} 0 \\ \hat{\mathbf{z}}(t-1) \end{bmatrix} \equiv \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \tag{4.36}$$

$$\begin{bmatrix} \mathbf{U} & -\mathbf{UA} \\ -\mathbf{A}^T\mathbf{U} & \mathbf{A}^T\mathbf{UA} + \hat{\mathbf{L}}(t-1) \end{bmatrix} \equiv \begin{bmatrix} \Xi_{11} & \Xi_{12} \\ \Xi_{21} & \Xi_{22} \end{bmatrix} \tag{4.37}$$

  where $\mathbf{U} \equiv \mathbf{B}^T\mathbf{Q}^{-1}(t)\mathbf{B}$, $\mathbf{A} \equiv \mathbf{A}(t)$, $\dim(\eta_1) = \dim(\mathbf{x}(t))$, $\dim(\eta_2) = \dim(\mathbf{f}(t))$, and $\dim(\Xi_{22}) = \dim\left(\mathbf{f}(t)\mathbf{f}(t)^T\right)$. Then, the predicted information pair is computed as

$$\bar{\mathbf{z}}(t) = \eta_1 - \Xi_{12}\Xi_{22}^{-1}\eta_2 \tag{4.38}$$

$$\bar{\mathbf{L}}(t) = \Xi_{11} - \Xi_{12}\Xi_{22}^{-1}\Xi_{21}. \tag{4.39}$$

  Note that having a non-singular $\hat{\mathbf{L}}(t-1)$ is sufficient to ensure that Ξ_{22} is invertible.

- *Update step:*

Update the information pair as follows:

$$\hat{\mathbf{L}}(t) = \overline{\mathbf{L}}(t) + \begin{bmatrix} \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} & & & \\ & \mathbf{0} & & \\ & & \ddots & \\ & & & \mathbf{0} \end{bmatrix} \tag{4.40}$$

$$\hat{\mathbf{z}}(t) = \overline{\mathbf{z}}(t) + \begin{bmatrix} \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y}(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.41}$$

Partition the resulting information pair as

$$\hat{\mathbf{z}}(t) = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} \tag{4.42}$$

$$\hat{\mathbf{L}}(t) = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \tag{4.43}$$

so that $\dim(\mathbf{z}_1) = \dim(\mathbf{f}(t))$ and $\dim(\mathbf{L}_{11}) = \dim(\mathbf{K})$. The information pair associated with $\mathbf{f}(t)$ can be extracted using Lemma 4.1 as

$$\hat{\mathbf{b}}(t) = \mathbf{z}_1 - \mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{z}_2 \tag{4.44}$$

$$\widehat{\mathbf{K}}(t) = \mathbf{L}_{11} - \mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21} \tag{4.45}$$

from which the updated estimate can be computed as

$$\hat{\mathbf{f}}(t) = \widehat{\mathbf{K}}^{-1}(t)\hat{\mathbf{b}}(t), \quad \hat{\mathbf{P}}(t) = \widehat{\mathbf{K}}^{-1}(t). \tag{4.46}$$

The most computationally intensive step is inversion of the $(n-1)N \times (n-1)N$ matrix $\mathbf{L}_{22}$ in the update step (where N is the dimension of a frame of $\mathbf{f}(t)$ and n is the number of frames in the state vector $\mathbf{x}(t)$); thus, efficiency of this algorithm

heavily depends on the value of n as well as the structure of the submatrix $\mathbf{L}_{22}$.

4.2.5 Example 1

We illustrate the filtering process in a specific multi-frame reconstruction problem. Consider a discrete formulation of the following space-time reconstruction problem for $f(s,t)$:

$$\min_f \int\int \nu \|g - hf\|^2 + \mu_1 \left\| \frac{\partial}{\partial s} f \right\|^2 + \mu_2 \left\| \frac{\partial^2}{\partial s^2} f \right\|^2 + \rho_1 \left\| \frac{\partial}{\partial t} f \right\|^2 + \rho_2 \left\| \frac{\partial^2}{\partial t^2} f \right\|^2 ds\, dt.$$

By letting $f(s,t)$ parameterize the points on a contour in the image frame, we can apply this least-squares formulation to dynamic contour reconstruction problems such as estimation of heart chamber boundaries from a sequence of noisy images [13]. A discretized ML estimation version of this problem is the estimation based on the dynamic system

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \mathbf{f}(t) = \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ 2\mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}(t-1) \\ \mathbf{f}(t-2) \end{bmatrix} + \mathbf{q}(t) \tag{4.47}$$

$$\mathbf{q}(t) \sim \left(0, \begin{bmatrix} \rho_1^{-1}\mathbf{I} & \\ & \rho_2^{-1}\mathbf{I} \end{bmatrix} \right)$$

$$\begin{bmatrix} \mathbf{g}(t) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{H}(t) \\ \mathbf{S}^{(1)} \\ \mathbf{S}^{(2)} \end{bmatrix} \mathbf{f}(t) + \mathbf{r}(t) \tag{4.48}$$

$$\mathbf{r}(t) \sim \left(0, \begin{bmatrix} \mathbf{N}^{-1}(t) & & \\ & \mu_1^{-1}\mathbf{I} & \\ & & \mu_2^{-1}\mathbf{I} \end{bmatrix} \right).$$

- *Prediction:*

  Let the posterior statistics of $\mathbf{x}(t-1)$ from time $t-1$ be

$$\mathbf{x}(t-1) = \begin{bmatrix} \mathbf{f}(t-1) \\ \mathbf{f}(t-2) \end{bmatrix} \sim \left\langle\left\langle \begin{bmatrix} \hat{\mathbf{z}}_1 \\ \hat{\mathbf{z}}_2 \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{L}}_{11} & \hat{\mathbf{L}}_{12} \\ \hat{\mathbf{L}}_{21} & \hat{\mathbf{L}}_{22} \end{bmatrix} \right\rangle\right\rangle.$$

  Then, from (4.36) and (4.37),

$$\eta_1 = \begin{bmatrix} 0 \\ \hat{\mathbf{z}}_1 \end{bmatrix}, \qquad \eta_2 = \hat{\mathbf{z}}_2$$

$$\Xi_{11} = \begin{bmatrix} (\rho_1+\rho_2)\mathbf{I} & -(\rho_1+2\rho_2)\mathbf{I} \\ -(\rho_1+2\rho_2)\mathbf{I} & \hat{\mathbf{L}}_{11}+(\rho_1+4\rho_2)\mathbf{I} \end{bmatrix}, \quad \Xi_{12} = \begin{bmatrix} \rho_2\mathbf{I} \\ \hat{\mathbf{L}}_{12}-2\rho_2\mathbf{I} \end{bmatrix}$$

$$\Xi_{21} = \begin{bmatrix} \rho_2\mathbf{I} & \hat{\mathbf{L}}_{21}-2\rho_2\mathbf{I} \end{bmatrix}, \qquad \Xi_{22} = \hat{\mathbf{L}}_{22}+\rho_2\mathbf{I}$$

  and predicted information pair is given as

$$\begin{aligned} \overline{\mathbf{z}}(t) &= \eta_1 - \Xi_{21}\Xi_{22}^{-1}\eta_2 \\ \overline{\mathbf{L}}(t) &= \Xi_{11} - \Xi_{12}\Xi_{22}^{-1}\Xi_{21}. \end{aligned}$$

- *Update:*

  Let the predicted information pair be partitioned as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{f}(t) \\ \mathbf{f}(t-1) \end{bmatrix} \sim \left\langle\left\langle \begin{bmatrix} \overline{\mathbf{z}}_1 \\ \overline{\mathbf{z}}_2 \end{bmatrix}, \begin{bmatrix} \overline{\mathbf{L}}_{11} & \overline{\mathbf{L}}_{12} \\ \overline{\mathbf{L}}_{21} & \overline{\mathbf{L}}_{22} \end{bmatrix} \right\rangle\right\rangle.$$

  The updated information pair is given by

$$\begin{aligned} \hat{\mathbf{z}}(t) &= \begin{bmatrix} \overline{\mathbf{z}}_1 + \mathbf{H}^T\mathbf{N}\mathbf{g} \\ \overline{\mathbf{z}}_2 \end{bmatrix} \\ \hat{\mathbf{L}}(t) &= \begin{bmatrix} \overline{\mathbf{L}}_{11} + \mathbf{H}^T\mathbf{N}\mathbf{H} + \mu_1\mathbf{S}^{(1)^T}\mathbf{S}^{(1)} + \mu_2\mathbf{S}^{(2)^T}\mathbf{S}^{(2)} & \overline{\mathbf{L}}_{12} \\ \overline{\mathbf{L}}_{21} & \overline{\mathbf{L}}_{22} \end{bmatrix}. \end{aligned}$$

Using (4.42)–(4.45), we obtain the updated information pair for $\mathbf{f}$ as

$$\begin{aligned}\widehat{\mathbf{b}}(t) &= \overline{\mathbf{z}}_1 + \mathbf{H}^T\mathbf{N}\mathbf{g} - \overline{\mathbf{L}}_{12}\overline{\mathbf{L}}_{22}^{-1}\overline{\mathbf{z}}_2 \\ \widehat{\mathbf{K}}(t) &= \overline{\mathbf{L}}_{11} + \mathbf{H}^T\mathbf{N}\mathbf{H} + \mu_1\mathbf{S}^{(1)^T}\mathbf{S}^{(1)} + \mu_2\mathbf{S}^{(2)^T}\mathbf{S}^{(2)} - \overline{\mathbf{L}}_{12}\overline{\mathbf{L}}_{22}^{-1}\overline{\mathbf{L}}_{21}.\end{aligned}$$

4.3 The Gauss-Markov Case

In this section we consider a special case of the descriptor system model (4.8) (4.9) for the multi-frame visual field reconstruction problems. Specifically, we let $\mathbf{x}(t) = \mathbf{f}(t)$, i.e.,

$$\mathbf{B}\mathbf{f}(t) = \mathbf{B}\mathbf{A}(t)\mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim (\ 0, \mathbf{Q}(t)\ ) \tag{4.49}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{f}(t) + \mathbf{r}(t), \quad \mathbf{r}(t) \sim (\ 0, \mathbf{R}(t)\ ), \tag{4.50}$$

which implies a *first order* temporal dynamic model for $\mathbf{f}(t)$. For this particular dynamic system, the information filtering algorithm presented in the previous section can be considerably simplified. Note that the equation (4.49) is in a standard descriptor form and that it can be reformulated as a standard Gauss-Markov dynamic equation only if the matrix $\mathbf{B}$ has full column rank.

4.3.1 Filtering algorithm for the Gauss-Markov system

Let $\mathbf{U} \equiv \mathbf{B}^T\mathbf{Q}^{-1}(t)\mathbf{B}$ and $\mathbf{A} \equiv \mathbf{A}(t)$. Then, by applying the algorithm in Section 4.2.4 to the system (4.49) (4.50) the equations for the prediction and update steps can be obtained as follows:

- prediction:

$$\overline{\mathbf{L}}(t) = \mathbf{U} - \mathbf{U}\mathbf{A}\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \widehat{\mathbf{L}}(t-1)\right)^{-1}\mathbf{A}^T\mathbf{U} \tag{4.51}$$

$$\overline{\mathbf{f}}(t) = \mathbf{A}\widehat{\mathbf{f}}(t-1) \tag{4.52}$$

$$\overline{\mathbf{z}}(t) = \overline{\mathbf{L}}(t)\overline{\mathbf{f}}(t) \tag{4.53}$$

- update:

$$\hat{\mathbf{L}}(t) = \overline{\mathbf{L}}(t) + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} \tag{4.54}$$

$$\hat{\mathbf{z}}(t) = \overline{\mathbf{z}}(t) + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y}(t) \tag{4.55}$$

$$\hat{\mathbf{f}}(t) = \hat{\mathbf{L}}^{-1}(t)\hat{\mathbf{z}}(t). \tag{4.56}$$

An advantage over standard information Kalman filters

As detailed below, the descriptor dynamic equation (4.49) can be expressed in a Gauss-Markov form (4.57). Thus, standard information Kalman filtering equations (e.g., [4, 45]) can also be applied to the estimation problem. The filtering algorithm (4.51)-(4.56) is, however, more suitable for visual reconstruction mainly due to the fact that, unlike the traditional information Kalman filters, the inverse of $\mathbf{A}(t)$ is not needed. As previously mentioned, in visual reconstruction the system matrix $\mathbf{A}(t)$ often performs a local averaging and thus is sparse. Taking its inverse generally loses its sparseness and computational efficiency of the filter. Also, it is conceivable that $\mathbf{A}(t)$ may not even be invertible in some formulations.

Derivation

Deriving (4.51)-(4.56) by applying the general information filter algorithm in Section 4.2.4 to the system (4.49) (4.50) is straightforward except for (4.52), which is derived below. From the general filtering algorithm, the predicted information pair is computed as

$$\begin{aligned}
\overline{\mathbf{L}}(t) &= \mathbf{U} - \mathbf{U}\mathbf{A}\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\mathbf{A}^T\mathbf{U} \\
\overline{\mathbf{z}}(t) &= \mathbf{U}\mathbf{A}\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\hat{\mathbf{z}}(t-1).
\end{aligned}$$

But the latter equation can be written as

$$\overline{\mathbf{z}}(t) = \mathbf{U}\mathbf{A}\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\hat{\mathbf{L}}(t-1)\hat{\mathbf{f}}(t-1)$$

$$
\begin{aligned}
&= \mathbf{U}\mathbf{A}\left[\mathbf{I} - \left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\mathbf{A}^T\mathbf{U}\mathbf{A}\right]\hat{\mathbf{f}}(t-1) \\
&= \left[\mathbf{U} - \mathbf{U}\mathbf{A}\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\mathbf{A}^T\mathbf{U}\right]\mathbf{A}\hat{\mathbf{f}}(t-1) \\
&= \overline{\mathbf{L}}(t)\mathbf{A}\hat{\mathbf{f}}(t-1).
\end{aligned}
$$

Thus,

$$
\bar{\mathbf{f}}(t-1) = \overline{\mathbf{L}}^{-1}(t)\bar{\mathbf{z}}(t) = \mathbf{A}\hat{\mathbf{f}}(t-1),
$$

proving (4.52).

4.3.2 Generalized Gauss-Markov process

Consider the descriptor dynamic process (4.49), i.e.,

$$
\mathbf{B}\mathbf{f}(t) = \mathbf{B}\mathbf{A}(t)\mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim (\ 0, \mathbf{Q}(t)\ ).
$$

For the purpose of estimating $\mathbf{f}(t)$ we can consider this dynamic process as a *generalized Gauss-Markov process*

$$
\mathbf{f}(t) = \mathbf{A}(t)\mathbf{f}(t-1) + \mathbf{u}(t), \quad \mathbf{u}(t) \sim \langle\langle\ 0, \mathbf{U}(t)\ \rangle\rangle \tag{4.57}
$$

where

$$
\mathbf{U}(t) \equiv \mathbf{B}^T\mathbf{Q}^{-1}(t)\mathbf{B} \tag{4.58}
$$

in the sense that the information filtering equations applied to this dynamic equation is equivalent to those applied to (4.49) (using the same observation equation (4.50)). In general, the information matrix $\mathbf{U}(t)$ associated with the process noise $\mathbf{u}(t)$ is singular because the submatrix components of $\mathbf{B}$, the spatial difference operators $\mathbf{S}_i$, are rank-deficient. Since the Gaussian density function for $\mathbf{u}(t)$ does not exist in a strict sense, the dynamic process is not a standard Gauss-Markov process. The information filtering algorithm, however, does not require $\mathbf{U}(t)$ to be invertible, and it

can be applied to (4.57) just as effectively as to (4.49). The Gauss-Markov form (4.57) is in fact preferrable to its descriptor form (4.49) because the large static matrix $\mathbf{B}$ is replaced by a smaller matrix $\mathbf{U}(t)$, leading to a reduced representation for the system model. For the rest of the thesis, we use this generalized Gauss-Markov form for first order temporal dynamic equations of $\mathbf{f}(t)$, mainly due to its notational conciseness and implementational compactness.

4.3.3 Example 2

Consider multi-frame reconstruction of a 2-D visual field whose temporal dynamics are modeled by imposing the thin-plate constraint on its first order temporal difference as

$$\begin{bmatrix} \mathbf{S}^{(2,0)} \\ \mathbf{S}^{(0,2)} \\ 2\mathbf{S}^{(1,1)} \end{bmatrix} \mathbf{f}(t) = \begin{bmatrix} \mathbf{S}^{(2,0)} \\ \mathbf{S}^{(0,2)} \\ 2\mathbf{S}^{(1,1)} \end{bmatrix} \mathbf{f}(t-1) + \mathbf{q}(t) \tag{4.59}$$

$$\mathbf{q}(t) \sim \left(0, \begin{bmatrix} \rho_1^{-1}\mathbf{I} & & \\ & \rho_2^{-1}\mathbf{I} & \\ & & \rho_3^{-1}\mathbf{I} \end{bmatrix} \right),$$

while the spatial coherence is implemented by a thin-plate model as well, i.e.,

$$\begin{bmatrix} \mathbf{g}(t) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{H}(t) \\ \mathbf{S}^{(2,0)} \\ \mathbf{S}^{(0,2)} \\ 2\mathbf{S}^{(1,1)} \end{bmatrix} \mathbf{f}(t) + \mathbf{r}(t) \tag{4.60}$$

$$\mathbf{r}(t) \sim \left(0, \begin{bmatrix} \mathbf{N}^{-1}(t) & & & \\ & \mu_1^{-1}\mathbf{I} & & \\ & & \mu_2^{-1}\mathbf{I} & \\ & & & \mu_3^{-1}\mathbf{I} \end{bmatrix} \right).$$

As mentioned before, thin-plate models are suitable for surface interpolation, and we will present examples of multi-frame surface interpolation in the next chapter (using approximated filters). The generalized Gauss-Markov process corresponding to the descriptor dynamic equation (4.59) is

$$\mathbf{f}(t) = \mathbf{f}(t-1) + \mathbf{u}(t), \quad \mathbf{u}(t) \sim \langle\langle\ 0, \mathbf{U}(t)\ \rangle\rangle \tag{4.61}$$

where

$$\mathbf{U}(t) \equiv \rho_1 \mathbf{S}^{(2,0)^T}\mathbf{S}^{(2,0)} + \rho_2 \mathbf{S}^{(0,2)^T}\mathbf{S}^{(0,2)} + 4\rho_3 \mathbf{S}^{(1,1)^T}\mathbf{S}^{(1,1)}. \tag{4.62}$$

The equations (4.61) (4.60) form a generalized Gauss-Markov system, to which we can apply the information Kalman filtering algorithm (4.51)-(4.56) to estimate the visual field.

4.4 Square Root Information Filter (SRIF)

Invertibility of the updated information matrices are important in visual reconstruction as it implies wellposedness of the problem. In implementing the information filters presented so far, therefore, we need to be careful about whether the *computed* information matrix in each *update step* is non-singular, because numerical roundoff could introduce indefiniteness in information matrices in some reconstruction problems, especially in those problems that have near-rank deficient updated information matrices. For example, in optical flow estimation, the updated information matrix can become near-rank deficient when the directions of the spatial brightness gradients are nearly the same. In surface reconstruction, a similar situation can also arise when the depth measurements are very sparse over the image frame.

The so-called square root form of an information Kalman filter, or *a square root information filter (SRIF)*, improves numerical stability in such near-singular estimation problems by reducing numerical dynamic ranges of the variables [9, 47]. Besides providing an increased margin for numerical roundoff errors, such reduction in the dy-

namic ranges can also relieve memory requirements (i.e., the number of bits required to represent each variable) in filter implementation — a desirable feature as low-level visual reconstruction algorithms are likely to be VLSI-implemented as "front-ends" of a complex visual system. In this section, we present an SRIF algorithm for the generalized Gauss-Markov system discussed in the previous section.

4.4.1 Data fusion in square root form

The computational mechanisms for SRIF algorithms are considerably different from those for the information filter algorithms. Unitary transformation is a key operation in SRIF's. Here, we briefly review some rudiments of square root estimation theory [9].

Unitary transformation and unit white noise

Let us use δ to denote generically any zero-mean white noise process whose covariance matrix is an identity matrix, i.e.,

$$\delta \sim (0, \mathbf{I}) .$$

The dimension of δ is context-dependent. A *unitary transformation* on a (column) vector is a left multiplication by a matrix operator $\mathbf{T}$ such that $\mathbf{T}^T\mathbf{T} = \mathbf{I}$. An important fact is that the statistical properties of δ are unaltered by a unitary transformation, i.e., if $\delta' = \mathbf{T}\delta$ then $\delta' \sim (0, \mathbf{I})$. Thus, using our generic notation, we write $\delta = \mathbf{T}\delta$.

Matrix square root

If a square matrix M can be written as

$$M = S^T S$$

where S is a square matrix, we say that S is a *square root* of M and write

$$M^{1/2} \equiv S.$$

For notational convenience, we write the transpose of $M^{1/2}$ as $M^{T/2}$, and if $M^{1/2}$ is invertible we write its inverse as $M^{-1/2}$. Similarly, the transpose of the inverse is written as $M^{-T/2}$.

The usual definition of square root matrices does not require them to be square matrices; however, we will confine ourselves here to square roots that *are* square, as this will be important in developing sparse filter approximations in Chapter 5. For any square matrix, a square root matrix which is square always exists, but it is not unique. (For a symmetric matrix, however, a unique *symmetric* square root always exists. See Appendix C for more details on such symmetric square root matrices.)

SRI-pair

Let us define the *square root information pair* or *SRI-pair* to be a square root version of the information pair. Specifically, if $\mathbf{w}$ and $\mathbf{L}^{1/2}$ are the SRI-pair corresponding to the information pair $(\mathbf{z}, \mathbf{L})$ then

$$\mathbf{L}^{T/2}\mathbf{L}^{1/2} \equiv \mathbf{L}, \quad \mathbf{L}^{T/2}\mathbf{w} \equiv \mathbf{z}. \tag{4.63}$$

We denote a SRI-pair as

$$\left\langle\left\langle\, \mathbf{w}, \mathbf{L}^{1/2} \,\right\rangle\right\rangle^{1/2}$$

and call $\mathbf{L}^{1/2}$ the *square root information (SRI) matrix.* The corresponding mean-covariance pair $\left(\hat{\mathbf{f}}, \hat{\mathbf{P}}\right)$ can be computed from the SRI-pair as

$$\begin{aligned} \hat{\mathbf{P}} &= \mathbf{L}^{-1} = \left(\mathbf{L}^{T/2}\mathbf{L}^{1/2}\right)^{-1} \\ \hat{\mathbf{f}} &= \mathbf{L}^{-1}\mathbf{z} = \mathbf{L}^{-1/2}\mathbf{L}^{-T/2}\mathbf{z} = \mathbf{L}^{-1/2}\mathbf{w}. \end{aligned}$$

In particular, the mean $\hat{\mathbf{f}}$ can be obtained as the solution of the inverse problem set up by the SRI-pair:

$$\mathbf{L}^{1/2}\hat{\mathbf{f}} = \mathbf{w}.$$

Data fusion with the SRI-pair

We now present a fundamental algorithmic mechanism for SRIF's. Consider ML estimation based on the following two observation equations for an M-dimensional unknown vector $\mathbf{x}$:

$$\mathbf{y}_i = \mathbf{C}_i\mathbf{x} + \mathbf{r}_i, \quad \mathbf{r}_i \sim (\, 0, \mathbf{R}_i \,) \tag{4.64}$$

for $i = 1, 2$. Since $\mathbf{R}_i^{-1/2}\mathbf{r}_i = \delta$, these equations can be written as

$$\mathbf{R}_i^{-1/2}\mathbf{C}_i\mathbf{x} - \mathbf{R}_i^{-1/2}\mathbf{y}_i = \delta \tag{4.65}$$

from which the SRI-pair associated with each of these equations can be extracted as

$$\mathbf{x} \sim \left\langle\!\left\langle\, \mathbf{R}_i^{-1/2}\mathbf{y}_i \,,\, \mathbf{R}_i^{-1/2}\mathbf{C}_i \,\right\rangle\!\right\rangle^{1/2}$$

for $i = 1, 2$. By (4.63), these SRI-pairs can be verified to correspond to the information pairs $\left\langle\!\left\langle\, \mathbf{C}_i^T\mathbf{R}_i^{-1}\mathbf{y}_i \,,\, \mathbf{C}_i^T\mathbf{R}_i^{-1}\mathbf{C}_i \,\right\rangle\!\right\rangle$ for $i = 1, 2$.

To fuse the two SRI-pairs together, we set up a joint observation equation based on (4.65) as

$$\begin{bmatrix} \mathbf{R}_1^{-1/2}\mathbf{C}_1 & \mathbf{R}_1^{-1/2}\mathbf{y}_1 \\ \mathbf{R}_2^{-1/2}\mathbf{C}_2 & \mathbf{R}_2^{-1/2}\mathbf{y}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \delta. \tag{4.66}$$

For this estimation problem to be well-posed, the matrix operator on the left hand side of the equation must have a column rank of at least M. We assume such is the case. Then, there exists a unitary transform $\mathbf{T}$ that *nulls* the lower-left block of the

operator matrix above as

$$\mathbf{T}\begin{bmatrix} \mathbf{R}_1^{-1/2}\mathbf{C}_1 & \mathbf{R}_1^{-1/2}\mathbf{y}_1 \\ \mathbf{R}_2^{-1/2}\mathbf{C}_2 & \mathbf{R}_2^{-1/2}\mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} V & v_a \\ \mathbf{0} & v_b \end{bmatrix}$$

so that the resulting submatrix V is of dimension $M \times M$ and is non-singular. One such $\mathbf{T}$ can be obtained by *QR factorization* [18]. Since $\mathbf{T}$ does not alter the white noise process, we can left-multiply both sides of (4.66) by $\mathbf{T}$ and obtain

$$\begin{bmatrix} V & v_a \\ \mathbf{0} & v_b \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \delta. \tag{4.67}$$

The top M rows of this equation are

$$V\mathbf{x} - v_a = \delta$$

from which the fused SRI-pair can be obtained as

$$\mathbf{x} \sim \langle\langle\, v_a, V \,\rangle\rangle^{1/2}.$$

An important point of this procedure is that all computations are performed with square root matrices (i.e., avoiding explicit computation of the information matrices), which have smaller dynamic ranges, to ensure the previously mentioned numerical stability. The readers are referred to [9, 40] for more complete coverages of the square root estimation theory.

4.4.2 Prediction step for the SRIF

Consider the SRIF for the generalized Gauss-Markov system introduced in the last section:

$$\mathbf{f}(t) = \mathbf{A}(t)\mathbf{f}(t-1) + \mathbf{u}(t), \quad \mathbf{u}(t) \sim \langle\langle\, 0, \mathbf{U}(t) \,\rangle\rangle \tag{4.68}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{f}(t) + \mathbf{r}(t), \quad \mathbf{r}(t) \sim (\, 0, \mathbf{R}(t) \,). \tag{4.69}$$

Consider, in particular, propagating the updated SRI-pair from the previous frame, $\hat{\mathbf{w}}(t-1)$ and $\hat{\mathbf{L}}^{1/2}(t-1)$, through these two equations.

Let $\mathbf{U}^{1/2}$ be a square root of $\mathbf{U}(t)$. In the prediction step, we fuse information in the SRI-pair from the previous frame, i.e.,

$$\hat{\mathbf{w}}(t-1) = \hat{\mathbf{L}}^{1/2}(t-1)\mathbf{f}(t-1) + \delta,$$

with information in (4.68). The prediction step is then captured by the following observation equation:

$$\begin{bmatrix} \hat{\mathbf{w}}(t-1) \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{L}}^{1/2}(t-1) & \mathbf{0} \\ -\mathbf{U}^{1/2}\mathbf{A} & \mathbf{U}^{1/2} \end{bmatrix} \begin{bmatrix} \mathbf{f}(t-1) \\ \mathbf{f}(t) \end{bmatrix} + \delta$$

or

$$\begin{bmatrix} \hat{\mathbf{L}}^{1/2}(t-1) & \mathbf{0} & \hat{\mathbf{w}}(t-1) \\ -\mathbf{U}^{1/2}\mathbf{A} & \mathbf{U}^{1/2} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f}(t-1) \\ \mathbf{f}(t) \\ -1 \end{bmatrix} = \delta. \tag{4.70}$$

By applying an appropriate unitary transformation on both sides of this equation we can obtain

$$\begin{bmatrix} V_1 & V_2 & v_1 \\ \mathbf{0} & V_3 & v_2 \end{bmatrix} \begin{bmatrix} \mathbf{f}(t-1) \\ \mathbf{f}(t) \\ -1 \end{bmatrix} = \delta \tag{4.71}$$

(where $\dim(V_i) = \dim\left(\mathbf{f}^T(t)\mathbf{f}(t)\right)$ and $\dim(v_i) = \dim(\mathbf{f}(t))$ for $i = 1, 2, \ldots$) from which we obtain the predicted SRI-pair

$$\overline{\mathbf{L}}^{1/2}(t) = V_3, \quad \overline{\mathbf{w}}(t) = v_2. \tag{4.72}$$

4.4.3 Update step for the SRIF

Expanded observation equation

The observation equation in a visual reconstructing dynamic system $\mathbf{y}(t) = \mathbf{C}(t)\mathbf{f}(t) + \mathbf{r}(t)$ by itself corresponds to a single-frame reconstruction problem. We have discussed in the previous chapter that such an observation equation can be expanded into the mutually independent spatial coherence part and measurement part. That is,

$$\begin{bmatrix} \mathbf{g}(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{H}(t) \\ \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_m \end{bmatrix} \mathbf{f}(t) + \begin{bmatrix} \mathbf{r}_0(t) \\ \mathbf{r}_1(t) \\ \vdots \\ \mathbf{r}_m(t) \end{bmatrix}$$

can be written as

$$\mathbf{g}(t) = \mathbf{H}(t)\mathbf{f}(t) + \mathbf{r}_0(t), \quad \mathbf{r}_0(t) \sim \left(0, \mathbf{N}^{-1}(t) \right) \tag{4.73}$$

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_m \end{bmatrix} \mathbf{f}(t) + \begin{bmatrix} \mathbf{r}_1(t) \\ \vdots \\ \mathbf{r}_m(t) \end{bmatrix}, \quad \mathbf{r}_i(t) \sim \left(0, \mu_i^{-1}\mathbf{I} \right) \quad i = 1, 2, \ldots, \tag{4.74}$$

where the noise processes $\mathbf{r}_i(t), i = 0, 1, 2, \ldots$, are mutually independent. The SRI-pair associated with (4.74) is

$$\mathbf{f}(t) \sim \langle\langle\ 0, \mathbf{S}\ \rangle\rangle^{1/2}$$

where the square root matrix $\mathbf{S}$ is defined as

$$\mathbf{S} \equiv \left(\sum_{i=1}^{m} \mu_i \mathbf{S}_i^T \mathbf{S}_i \right)^{1/2}. \tag{4.75}$$

Thus, (4.74) is equivalent to

$$0 = \mathbf{S}\mathbf{f}(t) + \delta$$

in the sense that both of these equations lead to the same SRI-pair. Similarly, (4.73) is equivalent to

$$\mathbf{N}^{-1/2}(t)\mathbf{g}(t) = \mathbf{N}^{-1/2}(t)\mathbf{H}(t)\mathbf{f}(t) + \delta.$$

Consequently, the measurement equation, (4.69), of the system can also be expressed as a pair of equations

$$\mathbf{N}^{-1/2}(t)\mathbf{g}(t) = \mathbf{N}^{-1/2}(t)\mathbf{H}(t)\mathbf{f}(t) + \delta \tag{4.76}$$

$$0 = \mathbf{S}\mathbf{f}(t) + \delta. \tag{4.77}$$

Note that the latter equation is more compact than (4.74) as $\mathbf{S}$ has the dimension $\dim\left(\mathbf{f}(t)\mathbf{f}(t)^T\right)$.

- *Spatial coherence update:*

  Combining the predicted SRI-pair and the spatial coherence part of the observation equation, we have

$$\begin{bmatrix} \overline{\mathbf{w}}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{L}}^{1/2}(t) \\ \mathbf{S} \end{bmatrix} \mathbf{f}(t) + \delta \tag{4.78}$$

  which can be written as

$$\begin{bmatrix} \overline{\mathbf{L}}^{1/2}(t) & \overline{\mathbf{w}}(t) \\ \mathbf{S} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f}(t) \\ -1 \end{bmatrix} = \delta. \tag{4.79}$$

  By applying an appropriate unitary transform on both sides of this equation we can obtain

$$\begin{bmatrix} V_4 & v_3 \\ \mathbf{0} & v_4 \end{bmatrix} \begin{bmatrix} \mathbf{f}(t) \\ -1 \end{bmatrix} = \delta \tag{4.80}$$

which implies that the data assimilated so far for estimation of $\mathbf{f}(t)$ can be summarized as an observation equation

$$v_3 = V_4\mathbf{f}(t) + \delta, \tag{4.81}$$

whose SRI-pair representation is $\langle\langle\ v_3, V_4\ \rangle\rangle^{1/2}$.

- *Measurement update:*

  To complete updating the SRI-pair, the summary observation equation just derived must be combined with the measurement equation. These two equations can be written jointly as

$$\begin{bmatrix} v_3 \\ \mathbf{N}^{-1/2}\mathbf{g}(t) \end{bmatrix} = \begin{bmatrix} V_4 \\ \mathbf{N}^{-1/2}\mathbf{H} \end{bmatrix} \mathbf{f}(t) + \delta \tag{4.82}$$

  which can be written as

$$\begin{bmatrix} V_4 & v_3 \\ \mathbf{N}^{-1/2}\mathbf{H} & \mathbf{N}^{-1/2}\mathbf{g}(t) \end{bmatrix} \begin{bmatrix} \mathbf{f}(t) \\ -1 \end{bmatrix} = \delta. \tag{4.83}$$

  By applying an appropriate unitary transform on both sides of this equation we can obtain

$$\begin{bmatrix} V_5 & v_5 \\ \mathbf{0} & v_6 \end{bmatrix} \begin{bmatrix} \mathbf{f}(t) \\ -1 \end{bmatrix} = \delta \tag{4.84}$$

  from which the updated SRI-pair can be obtained as

$$\widehat{\mathbf{L}}^{1/2}(t) = V_5, \quad \widehat{\mathbf{w}}(t) = v_5. \tag{4.85}$$

- *Mean-covariance pair:*

  The estimate is computed from this SRI-pair as

$$\widehat{\mathbf{f}}(t) = \widehat{\mathbf{L}}^{-1/2}(t)\widehat{\mathbf{w}}(t), \quad \widehat{\mathbf{P}}(t) = \widehat{\mathbf{L}}^{-1/2}(t)\widehat{\mathbf{L}}^{-T/2}(t). \tag{4.86}$$

In particular, the estimate is obtained by solving the inverse problem

$$\hat{\mathbf{L}}^{1/2}(t)\hat{\mathbf{f}}(t) = \hat{\mathbf{w}}(t). \tag{4.87}$$

4.4.4 Summary of SRIF algorithm

The basic mechanism for propagation of the SRI-pair involves *nulling of certain submatrices by a unitary operation.* Let $M \equiv \dim(\mathbf{f}(t))$, i.e., if $f(\mathbf{s},t)$ is a d-vector defined over a discrete spatial domain with N points then $M = Nd$.

- *The prediction step.*

  Given the SRI-pair from the previous frame $\hat{\mathbf{w}}(t-1)$ and $\hat{\mathbf{L}}^{1/2}(t-1)$ we form the following $2M \times (2M+1)$ matrix using the SRI-pair as well as the $M \times M$ matrices $\mathbf{U}^{1/2}(t)\mathbf{A}(t)$ and $\mathbf{U}^{1/2}(t)$ as its block components; then, by an appropriate unitary transformation we null the lower left $M \times M$ block of the matrix, i.e.,

$$\begin{bmatrix} \hat{\mathbf{L}}^{1/2}(t-1) & \mathbf{0} & \hat{\mathbf{w}}(t-1) \\ -\mathbf{U}^{1/2}\mathbf{A} & \mathbf{U}^{1/2} & 0 \end{bmatrix} \rightarrow \begin{bmatrix} V_1 & V_2 & v_1 \\ \mathbf{0} & V_3 & v_2 \end{bmatrix} \tag{4.88}$$

  where V_k and v_k are $M \times M$ matrix and M-vector, respectively, for $k = 1, 2, \ldots$. Then, the predicted SRI-pair is given by $\overline{\mathbf{w}}(t) = v_2$ and $\overline{\mathbf{L}}(t) = V_3$.

- *The update step 1 – spatial coherence update.*

  We form the following $2M \times (M+1)$ matrix using the predicted SRI-pair and an $M \times M$ matrix $\mathbf{S} \equiv \left[\sum_i \mu_i \mathbf{S}_i^T \mathbf{S}_i\right]^{1/2}$; then, we null the lower-left $M \times M$ block of the matrix by a unitary transformation:

$$\begin{bmatrix} \overline{\mathbf{L}}^{1/2}(t) & \overline{\mathbf{w}}(t) \\ \mathbf{S} & 0 \end{bmatrix} \rightarrow \begin{bmatrix} V_4 & v_3 \\ \mathbf{0} & v_4 \end{bmatrix}. \tag{4.89}$$

- *The update step 2 – measurement update.*

  Using the $M \times M$ matrix V_4 and M-vector v_3 just computed as well as the

matrix $\mathbf{N}^{1/2}\mathbf{H}$ and vector $\mathbf{N}^{1/2}\mathbf{g}$ obtained from the measurements, we again form a matrix with $M+1$ columns and then null its lower-left block so that V_5 is $M \times M$:

$$\begin{bmatrix} V_4 & v_3 \\ \mathbf{N}^{1/2}\mathbf{H} & \mathbf{N}^{1/2}\mathbf{g} \end{bmatrix} \rightarrow \begin{bmatrix} V_5 & v_5 \\ \mathbf{0} & v_6 \end{bmatrix}. \tag{4.90}$$

The updated SRI-pair is given by $\hat{\mathbf{w}}(t) = v_5$ and $\hat{\mathbf{L}}^{1/2}(t) = V_5$. The estimated field is obtained by solving the inverse problem

$$\hat{\mathbf{L}}^{1/2}(t)\hat{\mathbf{f}}(t) = \hat{\mathbf{w}}(t). \tag{4.91}$$

In summary, propagation of the SRI-pair is achieved by nulling the appropriate lower-left blocks of the aggregate matrices shown on the left hand sides of (4.88) (4.89) (4.90).

4.4.5 Example 3

In Example 2 (Sec. 4.3.3), a generalized Gauss-Markov system based on thin-plate space-time constraints (4.61) (4.60) is presented. To apply the SRIF equations just presented we need to first compute the square roots $\mathbf{U}^{1/2}$ and $\mathbf{S}$ based on

$$\mathbf{U} = \rho_1 \mathbf{S}^{(2,0)^T}\mathbf{S}^{(2,0)} + \rho_2 \mathbf{S}^{(0,2)^T}\mathbf{S}^{(0,2)} + 4\rho_3 \mathbf{S}^{(1,1)^T}\mathbf{S}^{(1,1)} \tag{4.92}$$

$$\mathbf{S}^T\mathbf{S} = \mu_1 \mathbf{S}^{(2,0)^T}\mathbf{S}^{(2,0)} + \mu_2 \mathbf{S}^{(0,2)^T}\mathbf{S}^{(0,2)} + 4\mu_3 \mathbf{S}^{(1,1)^T}\mathbf{S}^{(1,1)}. \tag{4.93}$$

We also need the square root matrix $\mathbf{N}^{1/2}$, a square root of a (block) diagonal matrix. Noting that $\mathbf{A} = \mathbf{I}$ in this problem, we have all the parameter matrices necessary to propagate the SRI-pair as (4.88)-(4.90). The estimate at each t is computed from the updated SRI-pair as in (4.91).

4.4.6 Spatial model implied by SRI-pairs

We have seen previously that an information pair

$$\mathbf{f} \sim \langle\langle\, \mathbf{z}, \mathbf{L} \,\rangle\rangle$$

implies a spatial model for the visual field $\mathbf{f}$

$$\mathbf{L}\mathbf{f} = \zeta, \quad \zeta \sim (\, \mathbf{z}, \mathbf{L} \,).$$

By left-multiplying both sides by $\mathbf{L}^{-T/2}$ we have

$$\mathbf{L}^{1/2}\mathbf{f} = \lambda, \quad \lambda \sim (\, \mathbf{w}, \mathbf{I} \,) \tag{4.94}$$

where $\lambda \equiv \mathbf{L}^{-T/2}\zeta$ whose mean and covariance are given by $\mathbf{L}^{-T/2}\mathbf{z} = \mathbf{w}$ and $\mathbf{L}^{-T/2}\mathbf{L}\mathbf{L}^{-1/2} = \mathbf{I}$, respectively. This shows that an SRI-pair also implies a spatial model for $\mathbf{f}$.

4.5 Concluding Remarks

In this chapter we presented various temporal coherence constraints for multi-frame visual field reconstruction as direct extensions of the spatial coherence constraints used in the single-frame reconstruction problems. The resulting formulation of the multi-frame reconstruction is a state estimation problem for the descriptor dynamic system (4.8) (4.9) for which we derived an information filtering algorithm in Section 4.2.4. A useful special (i.e., first order) case of the problem can be formulated in a generalized Gauss-Markov system format (4.57) (4.50), and the corresponding filtering algorithm (4.51)-(4.56) is considerably simplified. A square root version of this filter is also presented in Section 4.4.4.

4.5.1 Recursion over space

The fitering algorithms presented in this chapter essentially perform recursive data assimilation over the time axis. As the temporal coherence constraints in visual reconstruction plays the same mathematical role as the spatial coherence, a similar recursive processing can be performed over one of the spatial axes. Such spatially recursive computation is particularly useful for efficient reconstruction of visual fields defined over a 1-D space, as demonstrated in [65] for estimation of optical flow along a contour (2.12). An analogous recursion over a higher dimensional space faces computational complexity issues because of large state dimensions [57, 79, 78]. One notable difference between temporal and spatial recursion is that the unknown processes in general lose causality over space; thus, for spatially recursive estimation *smoothing*, rather than filtering, algorithms are used.

4.5.2 Locality of spatial models implied by information pairs

We have seen that throughout stages of single-frame estimation the information matrix preserves a sparse structure, providing us with a local spatial model interpretation of the corresponding information pairs. The same is not true in multi-frame problems. Let us, for example, consider the information filtering algorithms, (4.51)-(4.56), for the generalized Gauss-Markov process. The information matrix loses its sparseness because of the matrix inversion operation in (4.51), as an inverse of a non-diagonal, sparse matrix is in general a full matrix. From the spatial model perspective, it is conceivable that such information matrix still retains strong local components because physical relations among the components of the visual fields, as well as the measurements with which we reconstruct them, are usually spatially local. A similar statement can be also made about the SRIF algorithm, as the SRI-pair also implies a spatial model for the unknown field. These provide us with the motivation for effective and efficient approximate implementations of the filtering algorithms described in this chapter. Such implemental issues are discussed next.

Chapter 5

Suboptimal Filtering Techniques

The number of pixels, N, in a typical image frame is in the range of 10^4 to 10^6. The information filter for a generalized Gauss-Markov process presented in the last chapter requires storage of $O(N^2)$ elements of the information or the square-root information (SRI) matrices — a substantial memory requirement. Moreover, an optimal propagation of such large matrices over time requires a practically impossible amount of computation. Computational requirements are particularly severe for matrix inversions in the information filter and unitary transformations in the SRI filter. Clearly, practical implementation of these filters must be achieved through some approximations. In this chapter we discuss such implementational issues.

5.1 Approximating the Information Filter

5.1.1 The filtering problem

Let us consider the information filter for a generalized Gauss-Markov system

$$\mathbf{f}(t) = \mathbf{A}(t)\mathbf{f}(t-1) + \mathbf{u}(t), \quad \mathbf{u}(t) \sim \langle\langle\ 0, \mathbf{U}(t)\ \rangle\rangle \tag{5.1}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{f}(t) + \mathbf{r}(t), \quad \mathbf{r}(t) \sim (\ 0, \mathbf{R}(t)\ ) \tag{5.2}$$

which, as discussed in the last chapter, is a descriptor system in disguise. The information filter corresponding to this system is

$$\begin{aligned}
\bar{\mathbf{L}}(t) &= \mathbf{U} - \mathbf{U}\mathbf{A}\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\mathbf{A}^T\mathbf{U} && (5.3)\\
\bar{\mathbf{f}}(t) &= \mathbf{A}\hat{\mathbf{f}}(t-1) && (5.4)\\
\bar{\mathbf{z}}(t) &= \bar{\mathbf{L}}(t)\bar{\mathbf{f}}(t) && (5.5)\\
\hat{\mathbf{L}}(t) &= \bar{\mathbf{L}}(t) + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} && (5.6)\\
\hat{\mathbf{z}}(t) &= \bar{\mathbf{z}}(t) + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y}(t) && (5.7)\\
\hat{\mathbf{f}}(t) &= \hat{\mathbf{L}}^{-1}(t)\hat{\mathbf{z}}(t). && (5.8)
\end{aligned}$$

We first consider approximating this information filter. Approximation for the corresponding square-root filter will be discussed later in the chapter.

2-D scalar fields

For explicitness in presentation, we assume that the field is defined over a 2-D space (e.g. an image frame) and that the field is scalar. The filter approximation techniques can be straightforwardly extended to the other cases where we need to deal with vector fields and fields defined over a 1-D or 3-D space. We will comment on such extensions at the end of the chapter.

5.1.2 Approximation strategy

Through approximations performed on the information filter we want to achieve:

- Reduction in storage requirements for the information matrix.
- Reduction in computational complexities, especially those imposed by matrix inverses, and enhanced parallelizability.

Computational structure of the single-frame problems

Our approximation methods are motivated by the structure of the single-frame visual reconstruction problems whose solutions can be computed efficiently. Recall from

Chapter 3 that in the single-frame visual reconstruction problems the *information matrices* have certain sparsely banded structures. Such matrix structures allow us to

- store only the $O(N)$ non-zero elements of the information matrix, and
- invert the information matrix efficiently using an iterative method.

Inversion of the information matrix is needed when the estimates are computed from the information pairs obtained from the measurements and the spatial coherence constraints, and it is a key computational step in the single-frame problems. As mentioned previously, for various single-frame problems it takes $O(N)$ iterations to obtain reasonable estimates, but with a multigrid method the amount of computation can be reduced by an order of magnitude or more [72].

The sparse structure of the information matrices that leads to such efficient implementations of the single-frame problems reflects the fact that the spatial coherence constraints are localized in the sense that each constraint equation essentially operates over a confined area on the spatial domain to take a local weighted average among some neighboring field elements. (Recall that such local averages are used as measures of how smooth, i.e., coherent, the field is.) The relation between the local nature of the problem formulation and the sparse, banded structures of the information matrices is elaborated later.

Computational structure of the multi-frame problems

The multi-frame version of the reconstruction problem has an inherently local structure as well, as the extents of the temporal coherence constraints are also localized. Although in its optimal form the information filter algorithm still requires some global operations such as matrix inversion over the entire image frame, the algorithm is "almost there" in terms of computational efficiency, and we will describe in the sequel how to make an effective approximation of the multi-frame algorithm so that the resulting suboptimal filter is truly local (thus, is parallelizable and requires much reduced memory for information matrix storage) while demanding no more computational complexity per frame than the corresponding single-frame problem. Specifi-

cally, in the filtering equations, all except (5.3) preserve sparseness of the information matrix. Also, (5.3) is the only step that requires an explicit inverse matrix. Hence, an efficient implementation of the information filter seems to be possible by approximating this equation in a way to enhance computational efficiency and preserve the sparse matrix structure. Note: (5.8) also requires inversion of an updated information matrix, but if all the information matrices are approximated to be sparse, then this step would have the same computational structure as the corresponding inversion step in the single-frame problem.

5.1.3 The approximation problem

We intend to approximate (5.3) so that the approximated predicted information matrix $\overline{\mathbf{L}}_a(t)$ is confined to have the sparse banded structure analogous to those of the information matrices in the single-frame problems. Development of such an approximate filtering algorithm can be regarded as a two-step process. First of all, we need to make certain that *truncating* the optimal predicted information matrix $\overline{\mathbf{L}}$ to be a sparse matrix, $\overline{\mathbf{L}}_a(t)$, will actually lead to an effective visual reconstrution filter. Then, we need to find a way to compute $\overline{\mathbf{L}}_a(t)$ efficiently or to approximate $\overline{\mathbf{L}}_a(t)$ itself so that we can avoid the explicit matrix inversion in (5.3). These issues are addressed in Sections 5.2 and 5.3.

5.1.4 Spatial modeling perspective of the approximation

As discussed previously, an information pair implies a spatial model for the corresponding unknown field. The information matrix, in particular, can be considered to encode interactions among the components of the field in such a spatial model. It makes physical sense that the information matrices are local, i.e., have structures corresponding to the locality of the spatial relationships among the components of the field, as the natural forces and energies governing structural characteristics of the field usually have local extent.

Let us consider the spatial model for the predicted information pair

$$\mathbf{f}(t) \sim \left\langle\left\langle\, \bar{\mathbf{z}}(t), \overline{\mathbf{L}}(t) \,\right\rangle\right\rangle,$$

i.e.,

$$\overline{\mathbf{L}}(t)\mathbf{f}(t) = \zeta, \quad \zeta \sim \left(\, \bar{\mathbf{z}}(t), \overline{\mathbf{L}}(t) \,\right). \tag{5.9}$$

Because of the matrix inversion process in (5.3), $\overline{\mathbf{L}}(t)$ is a full matrix even though $\hat{\mathbf{L}}(t-1)$ might be a sparsely banded matrix such as the updated information matrix in a single-frame problem. Approximating $\overline{\mathbf{L}}(t)$ by a sparse matrix $\overline{\mathbf{L}}_a(t)$ having a "local" structure described above corresponds to characterizing the field $\mathbf{f}(t)$ by a reduced-order version of the spatial model (5.9). This leads to the following physical intuition for our approximation strategy for the information filter:

- The prediction step of the information filter can be considered as a model realization process (for the error in the one-step predicted field). Such a model associated with the *optimal* information filter, which tends to yield full information matrices, characterizes the spatially discrete field by specifying every conceivable interaction among its components.

- Since a visual field in natural scenes can normally be specified by spatially local interactions among its components, a reduced order model obtained by truncating the predicted information matrix should be effective.

5.2 Truncated Information Filter

Consider the information filtering algorithm (5.3)-(5.8) in which the information matrices are truncated to have a certain sparse, diagonally banded structure that we call a *neighborhood structure*. If the number of bands is $O(1)$, the information matrix with such a neighborhood structure has only $O(N)$ non-zero elements; thus, this information matrix can store the approximated second order statistical data with

only that many storage elements as opposed to the usual number of $O(N^2)$. In this section, we examine such a truncation of information matrices and its effect on filter performance. Computational complexity issues will be discussed in Section 5.3.

5.2.1 Structural constraints on the information matrix

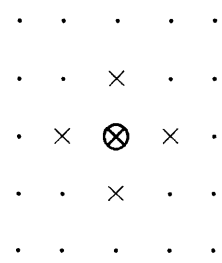
Let us first make explicit the neighborhood structures to be imposed on the information matrices. The motivation for such structural constraints is provided by spatially local (i.e., neighborhood) interactions among the components of the unknown field. As detailed previously in our notation N components of the field $f(\mathbf{s},t)$ are organized by a lexicographical order to form the unknown vector $\mathbf{f}(t)$. Because of this lexicographical ordering, the spatial relationships among the components are often not apparent in the vector notation. As we will see, the diagonal bands in a matrix with the neighborhood structure may not always be adjacent to each other even though the matrix structure reflects the local spatial relationships among the components of the field.

The neighbor sets

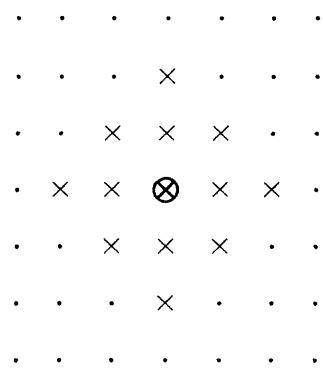
The components, $f(\mathbf{s},t)$, of the unknown field at a given time t are defined over a rectangular, discrete spatial domain. Let us define a *neighbor set* to be a set of spatial locations within a certain distance from a given point in the domain. The size of a neighbor set is determined by the number of *layers*, specifying the maximum rectangular distance[1] between the given point and each point in the set. Let us pictorally present the smallest neighbor sets over a 2-D spatial domain. In the figures below, $\times$'s represent the locations of the neighbors of the point denoted by $\otimes$. The locations of these $\times$'s and $\otimes$ form a neighbor set.

[1] also called "city block" or "Manhattan" distance

- 1-layer neighbor set.



- 2-layer neighbor set.



Note that the number of elements in a neighbor set with a certain number of layers, ℓ, is given by $2\ell(\ell+1)+1$. Obviously, a neighbor set centered around a pixel near an edge of the image frame has fewer elements because of truncation of the set by the edge.

The neighborhood structures

An information matrix can be considered to be a matrix operator operating on the field, as the predicted field at time t can be specified by a spatial model (5.9), i.e.,

$$\overline{\mathbf{L}}(t)\mathbf{f}(t) = \zeta, \quad \zeta \sim \left(\overline{\mathbf{z}}(t), \overline{\mathbf{L}}(t) \right).$$

That is, each row of the information matrix forms an inner product with the field vector $\mathbf{f}(t)$ to yield a weighted average of field elements, modeling the interaction

among certain components of the field. As discussed before, we intend to constrain the spatial extent of such an interaction to be local around a given point, and the spatial extent is, in fact, given by a neighbor set. Specifically, we constrain each row of the information matrix to have a structure corresponding to a lexicographically ordered neighbor set. (The lexicographical order is, of course, the same order by which $f(\mathbf{s}, t)$ are organized into $\mathbf{f}(t)$.)

Motivated by this, a matrix with a *neighborhood structure* is obtained from a neighbor set as follows: For each row, say the k^{th} row, of the matrix consider the (given) neighbor set centered around the corresponding, i.e., the k^{th}, element in the field vector. For the matrix to be a neighborhood structured matrix, all the elements in the row vector outside this neighbor set must be zero. Note that the diagonal elements of the matrix always belong to a neighbor set. Let us show some examples of neighborhood structured matrices.

- A neighborhood structured matrix corresponding to the 1-layer neighbor set is block tri-diagonal where the diagonal blocks are themselves tri-diagonal while the off-diagonal blocks are diagonal. Specifically, consider a rectangular 2-D spatial domain with the dimension $n \times m$ (i.e., $N = nm$); then a 1-layer neighborhood matrix has a structre

$$\begin{bmatrix} A_1 & C_1 & & \\ B_1 & A_2 & \ddots & \\ & \ddots & \ddots & C_{m-1} \\ & & B_{m-1} & A_m \end{bmatrix}$$

  where A's are $n \times n$ tri-diagonal submatrices while B's and C's are $n \times n$ diagonal submatrices. Clearly, the matrix has 5 diagonal bands, each corresponding to one of the 5 elements in the 1-layer neighbor set. This is also called the *nearest neighbor structure* [43]. The information matrix corresponding to a membrane model has this structure.

- Similarly, a 2-layer neighborhood matrix has a block penta-diagonal structure

$$\begin{bmatrix} A_1 & C_1 & E_1 & & & & \\ B_1 & A_2 & C_2 & E_2 & & & \\ D_1 & B_2 & A_3 & C_3 & E_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & E_{m-2} \\ & & & \ddots & \ddots & \ddots & C_{m-1} \\ & & & & D_{m-2} & B_{m-1} & A_m \end{bmatrix}$$

  so that the diagonal blocks A's are penta-diagonal, the first off-diagonal blocks B's and C's are tri-diagonal, and the second off-diagonal blocks D's and E's are diagonal. The matrix can be verified to have 13 diagonal bands, the same number as the elements in the 2-layer neighbor set. The information matrix corresponding to a thin-plate model has this structure.

The fact that the neighborhood structures reflect the structures of information matrices corresponding to various spatial coherence constraint serves as a motivation for constraining the information matrix in the filter by these structures. As demonstrated above, a neighborhood structured matrix has as many bands as the elements in the corresponding neighbor set. A neighbor set with a larger number of layers than those depicted above may be desirable for filter approximation purposes, e.g., when the spatial coherence constraints use higher order derivaties and when the system matrix $\mathbf{A}(t)$ has significant non-zero elements far from their main diagonals.

5.2.2 Filtering with truncated information matrices

We consider here an approximated information filter obtained by forcing $\overline{\mathbf{L}}(t)$ in (5.3) to have a neighborhood structure by truncation.

Definition 3 Let $\mathcal{W}$ be an $N \times N$ neighborhood structured matrix whose elements are either 0 or 1; an element is 1 if it belongs to the neighbor set, 0 if not. Thus, for

a given information matrix $\mathbf{L}$, the operation

$$\mathcal{W} \odot \mathbf{L},$$

where $\odot$ is an element-by-element multiplication operator, results in an information matrix truncated to be a matrix with the desired neighborhood structure. Thus, $\mathcal{W}$ can "mask" information matrices into neighborhood matrices. Since a certain neighborhood structure can be specified by the number of layers, we use the notation $\mathcal{W}_\ell$ to denote a *masking matrix* corresponding to a specific neighbor set with ℓ layers. Also, for convenience in terminology, a matrix $\mathbf{A}$ is said to be $\mathcal{W}$-*structured* or to be *a* $\mathcal{W}$ *matrix* if $\mathbf{A}$ has the same neighborhood structure as $\mathcal{W}$.

The truncated information filter algorithm

The optimal information filter has been given in (5.3)-(5.8). The truncated (or masked) information filter can be obtained by replacing (5.3) with

$$\overline{\mathbf{L}}(t) = \mathcal{W}_\ell \odot \left[\mathbf{U} - \mathbf{U}\mathbf{A}\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\mathbf{A}^T\mathbf{U}\right] \tag{5.10}$$

where the masking matrix $\mathcal{W}_\ell$ has been specified by the number of layers, ℓ, in the corresponding neighbor set. If ℓ has been chosen large enough so that the neighbor set contains the spatial extent of the spatial coherence constraint, the rest of the filtering algorithm does not alter the structure of the information matrix. Because of the matrix inverse on the right hand side of (5.10), this approximation does not improve on the computational complexity in the optimal information filter implementation. As we defer the discussion of this issue to the next section, however, let us point out that the truncation step greatly improves the storage situation, as the number of elements in the (approximated) information matrix is now $O(N)$, and the inversion operation in the update step, (5.8), can now be implemented efficiently by the afore-mentioned iterative methods.

Comments

- This truncation process corresponds to minimizing the Frobenius norm of the approximation error. For a given neighborhood structure, $\mathcal{W}$, let $\mathbf{L}_a$ be a $\mathcal{W}$-structured matrix. it can be easily shown that $\|\mathbf{L} - \mathbf{L}_a\|_F$ is minimized when $\mathbf{L}_a = \mathcal{W} \odot \mathbf{L}$.

- Besides simple truncation, there are possibilities for defining other optimization criteria for choosing $\mathbf{L}_a$. For example, we can choose $\mathbf{L}_a$ to be the matrix with the particular neighborhood structure that minimizes the Bhattacharyya and other distance measures [68] for probability density functions. Recall that an information pair implicitly defines Gaussian density function, so that we can choose $\mathbf{L}_a$ to be one that minimizes the "distance" between the Gaussian densities associated with $\mathbf{L}_a$ and $\mathbf{L}$. Specifically, if we use the *Bhattacharyya distance* we have

$$\min \ln \det \left[(\mathbf{L} + \mathbf{L}_a) \left(\mathbf{L}^{-1} + \mathbf{L}_a^{-1} \right) \right]$$

  while if we use the *divergence* we have

$$\min \operatorname{trace} \left[(\mathbf{L} - \mathbf{L}_a) \left(\mathbf{L}_a^{-1} - \mathbf{L}^{-1} \right) \right].$$

  Although these approximation criteria are attractive in the sense that they have an information theoretic foundation, there is no apparent way to compute $\mathbf{L}_a$ efficiently based on them. We, thus, stay with the simple truncation approach.

5.2.3 Numerical results

Consider applying the truncated information filter to the following dynamic system:

$$\mathbf{f}(t) = \mathbf{f}(t-1) + \mathbf{u}(t), \quad \mathbf{u}(t) \sim \langle\langle\ 0, \rho \mathbf{I}\ \rangle\rangle \tag{5.11}$$

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{I} \\ \mathbf{S}^{(1,0)} \\ \mathbf{S}^{(0,1)} \end{bmatrix} \mathbf{f}(t) + \mathbf{r}(t), \quad \mathbf{r}(t) \sim \left(0, \begin{bmatrix} \nu^{-1}\mathbf{I} & & \\ & \mathbf{I} & \\ & & \mathbf{I} \end{bmatrix} \right) \tag{5.12}$$

where $\mathbf{f}(t)$ is a scalar field defined over a 10×10 spatial domain. Estimation of $\mathbf{f}(t)$ corresponds to solving a discrete counterpart of the continuous multi-frame reconstruction problem

$$\int_0^\tau \int_{\mathcal{D}} \nu \|g - hf\|^2 + \left\| \frac{\partial}{\partial s_1} f \right\|^2 + \left\| \frac{\partial}{\partial s_2} f \right\|^2 + \rho \left\| \frac{\partial}{\partial t} f \right\|^2 \, d\mathbf{s} \, dt.$$

Let $\alpha \equiv \rho^{-1}$ and $\beta \equiv \nu^{-1}$. Then, α and β represent the variances of the process and measurement noise processes, respectively. We want to determine how well the truncated information matrices are approximated in the filter by examining the *percent approximation error*

$$\frac{\|\mathbf{L}_{apx} - \mathbf{L}_{opt}\|}{\|\mathbf{L}_{opt}\|} \times 100, \tag{5.13}$$

where $\mathbf{L}_{apx}$ is the approximated (i.e., truncated) information matrix and $\mathbf{L}_{opt}$ is the optimal information matrix. The 2-norm [18] is used to compute the matrix norms. Using this measure of performance, we consider the cases where the information matrices are truncated by the masking matrices $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$ as well as the cases where the process noise parameter α and the measurement noise parameter β are varied in the ranges of $[10^{-5}, 10^2]$ and $[10^{-1}, 10^2]$, respectively.

Figure 5-1 shows the approximation errors for the case $\alpha = \beta = 1$ when the filter has processed 10 frames ($t = 1, \ldots, 10$) of data with each of the three different masks. Referring to the figure we can make the following comments:

- As expected, the matrices are better approximated when fewer elements are truncated. The solid lines are associated with the filter using $\mathcal{W}_1$ for truncation, while the dashed and dotted lines are associated with those using $\mathcal{W}_2$ and $\mathcal{W}_3$, respectively.

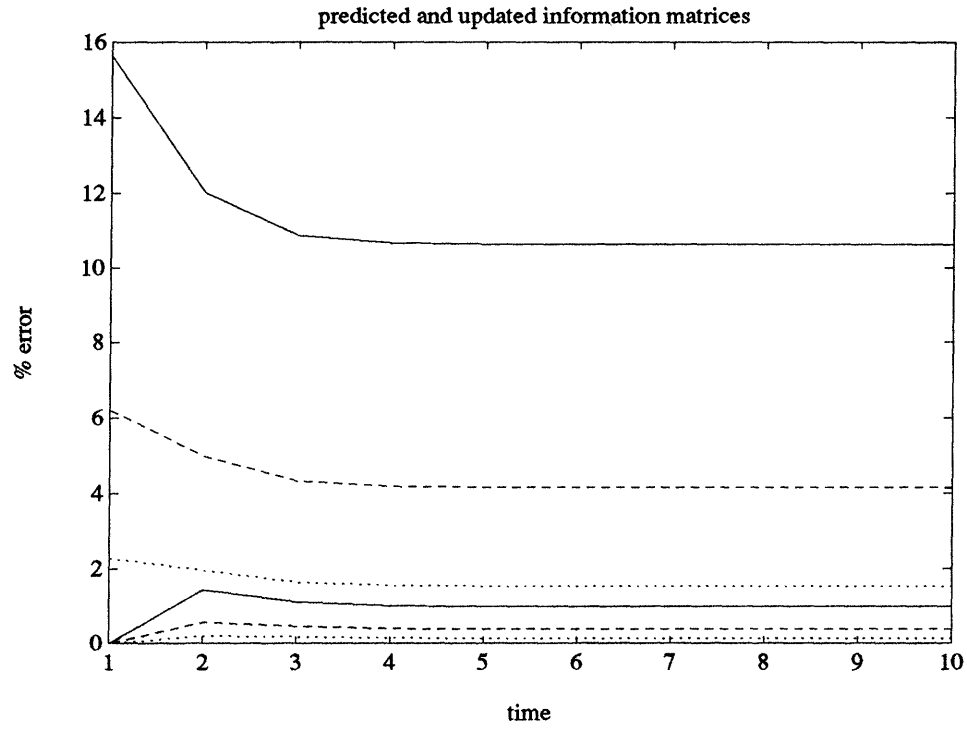


Figure 5-1: The approximation errors for the predicted and updated information matrices using different structural constraints in truncation — $\mathcal{W}_1$ (solid-line), $\mathcal{W}_2$ (dashed-line), and $\mathcal{W}_3$ (dotted-line). The top three lines are associated with the predicted informaton matrices, while the bottom three lines are associated with the updated information matrices.

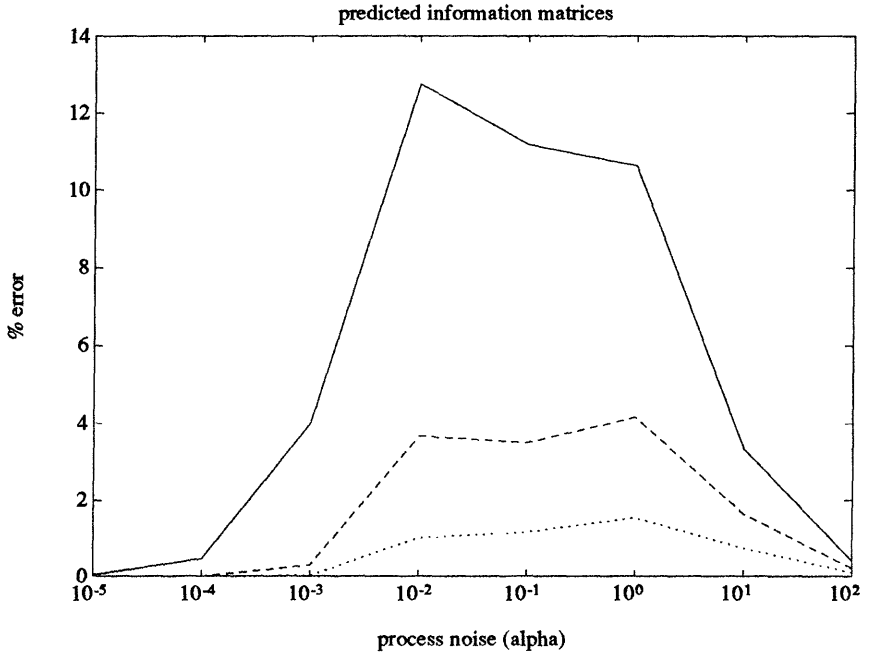


Figure 5-2: The effect of the process noise parameter on the approximation errors for the predicted information matrices using different structural constraints in truncation — $\mathcal{W}_1$ (solid-line), $\mathcal{W}_2$ (dashed-line), and $\mathcal{W}_3$ (dotted-line).

- The approximation errors for the predicted information matrices (top three lines) are consistently higher than those for the updated information matrices (bottom three lines). As we will see, this is a prevaling characteristic of a filter with structurally constrained information matrices.

- As the dynamic system (5.11) (5.12) is time-invariant, it is expected that the predicted and updated information matrices for the optimal filter converge to their steady-state values. We can observe from the figure that the approximation errors reach steady-state conditions by $t = 5$ as well, implying that the approximated information matrices also converge in time.

Effects of filter parameters

Let us consider how the modeling parameters α and β affect the performance of the approximated filters. Figures 5-2 and 5-3 show the approximation errors for the predicted and updated information matrices, respectively, at $t = 10$ when α is varied

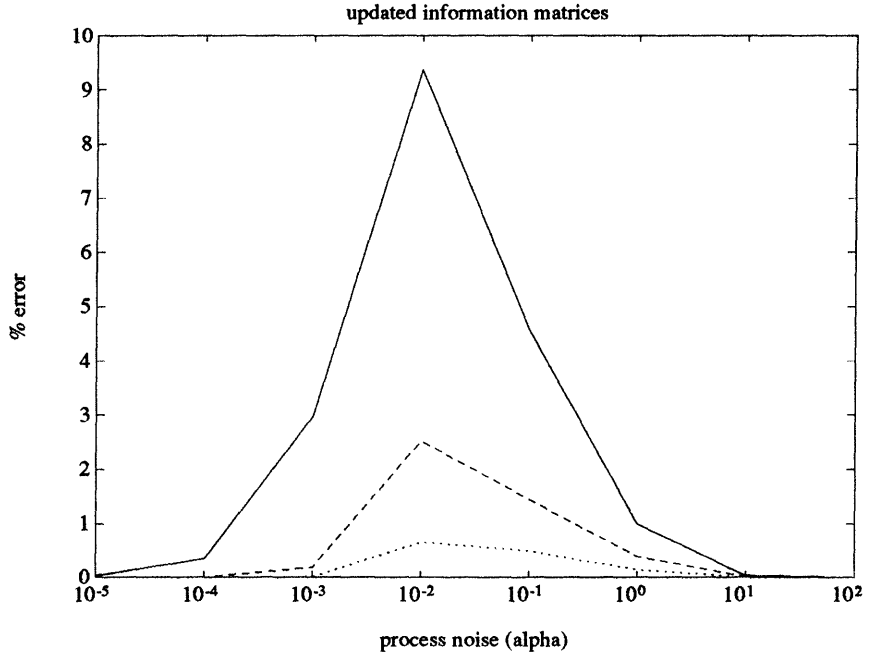


Figure 5-3: The effect of the process noise parameter on the approximation errors for the updated information matrices using different structural constraints in truncation — $\mathcal{W}_1$ (solid-line), $\mathcal{W}_2$ (dashed-line), and $\mathcal{W}_3$ (dotted-line).

from 10^{-5} to 10^2, while keeping $\beta = 1$. Again, the solid lines are associated with $\mathcal{W}_1$-structure approximation, the dashed lines are with $\mathcal{W}_2$, and the dotted lines are with $\mathcal{W}_3$. The near-unimodal shapes of the curves in the figures can be explained by the effect of the strength of the process noise on the structure of the optimal predicted information matrix. When the process noise is progressively decreased, the prediction based on (5.11) becomes closer to being perfect, and, in particular, the predicted information matrix approaches the updated information matrix from the previous time frame. Thus, the optimal predicted information matrix in this case almost has the same structure as the updated information matrix, i.e., the nearest neighbor structure, and truncation has small effect on it. When the process noise is very high, on the other hand, the prediction is close to providing *no* information about the unknown, i.e., the optimal predicted information matrix approaches zero. Thus, the structural constraints on the predicted information matrix again has small effect. The process noise significantly affects the performance of the truncated filter only in the cases $\alpha \in [10^{-3}, 10^1]$, and the maximum approximation errors are still

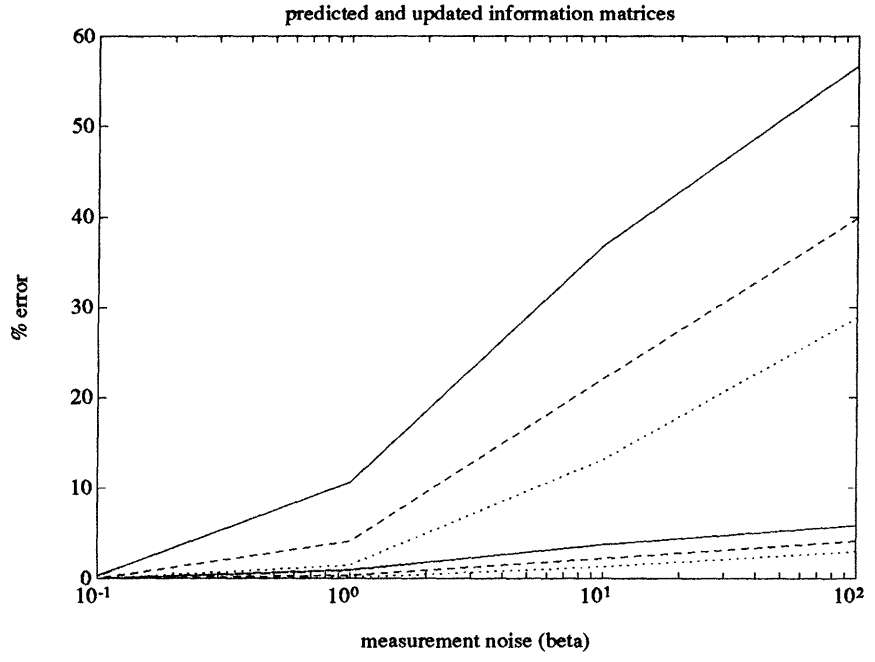


Figure 5-4: The effect of the measurement noise parameter on the approximation errors for the predicted and updated information matrices using different structural constraints in truncation — $\mathcal{W}_1$ (solid-line), $\mathcal{W}_2$ (dashed-line), and $\mathcal{W}_3$ (dotted-line).

low. For example, with the $\mathcal{W}_2$ structural constriant, the maximum errors for the predicted and updated information matrices are 4 and 2.5%, respectively, for this particular experiment.

The performance of the truncated filters is affected strongly by the strength of the measurement noise. Figure 5-4 shows the approximation errors for the predicted and updated information matrices when β is varied from 10^{-1} to 10^2 while keeping $\alpha = 1$. The approximation errors for the predicted information matrices (represented by the top three curves) are extremely high when the measurement noise covariance β is high, although the maximum error for the corresponding updated information matrices (bottom three curves) is still less than 7%. Recall that the diagonal information matrix associated with a measurement equation strengthens the diagonal part of the filter information matrix, thereby increasing the relative size of the norm of the elements within the $\mathcal{W}$-structure against the norm of the elements to be truncated off. A low value of ν (corresponding to a high level of measurement noise, β), therefore, makes the effect of truncation on the matrix greater.

Summary

Based on the criterion (5.13), the information matrices produced by the truncated information filter approximate those produced by the optimal filter well, if the measurement noise covariance is sufficiently small. Certainly, it is of interest to know how well the truncated information filter approximates the optimal *estimates* of the unknown visual fields. Experimentation performed for this purpose is presented in Section 5.5 in which estimates computed using various approximate and optimal filters are compared.

5.3 Efficient Sparse Matrix Inversion

Continuing the discussion of the information filter approximated by truncated information matrices, we examine in this section approaches to propagating the truncated information matrix, or its approximation, efficiently through the filtering equations.

5.3.1 The problem

The computational bottleneck in the approximate filtering scheme presented in the last section is (5.10) because of the inversion of the matrix

$$\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1), \tag{5.14}$$

which has a neighborhood structure. Note that (5.10) can also be written as

$$\overline{\mathbf{L}}(t) = \mathcal{W}_{\ell_1} \odot \left[\mathbf{U} - \mathbf{U}\mathbf{A}\left\{\mathcal{W}_{\ell_2} \odot \left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}\right\}\mathbf{A}^T\mathbf{U}\right] \tag{5.15}$$

for an $\ell_2 \geq \ell_1$. That is, if we intend to constrain the information matrices to have a $\mathcal{W}_{\ell_1}$-structure, we could first constrain the intermediate result $\left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1}$ to be of another neighborhood structure, $\mathcal{W}_{\ell_2}$, associated with a sufficiently larger neighbor set than that accompanying $\mathcal{W}_{\ell_1}$. This two-step structural constraint makes the matrix multiplications in (5.15) simple and *localized*, as every matrix involved in

the multiplications would be sparse with $O(N)$ elements as the result of the extra matrix truncation step.

The issue that we address in this section, then, is how to compute or approximate

$$\mathcal{W}_{\ell_2} \odot \left(\mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)\right)^{-1} \tag{5.16}$$

efficiently. The rest of the computations in (5.15), i.e., matrix multiplications, subtraction, and truncation, are all spatially confined so that they are parallelizable. Our hope is that the $O(N)$ elements in the matrix (5.16) can be computed or approximated just as efficiently.

5.3.2 Recursive inversion

As mentioned above, the computational bottleneck is the inversion of a $\mathcal{W}$-structured matrix. We have seen that a neighborhood structured matrix has a block banded structure, e.g., $\mathcal{W}_1$ is block tri-diagonal, $\mathcal{W}_2$ is block penta-diagonal, etc. We show here that such matrices can be inverted by processing their blocks sequentially.

Outer product representations for inverse elements

Asplund [6] has described various properties of inverses of diagonally banded matrices of a certain kind. Of these properties, that of particular interest to us is that the elements of the *inverses* of a *symmetric* matrix which is centrally banded (e.g., tri-diagonal, penta-diagonal, septa-diagonal, etc.) can be obtained as the elements of the sum of rank-1 outer products. For example, given a symmetric tri-diagonal matrix $\mathbf{T}$, the elements of its inverse $\mathbf{T}^{-1}$ can be obtained by first computing the elements of an outer product between a pair of vectors. Specifically, there exist $\{u_i\}$ and $\{v_j\}$ such that $(\mathbf{T}^{-1})_{i,j} = (\mathbf{T}^{-1})_{j,i} = u_i v_j$ for $i \leq j$. Similarly, for a symmetric penta-diagonal matrix $\mathbf{P}$, there exist $\{s_i\}, \{t_j\}, \{u_i\}$, and $\{v_j\}$ such that $(\mathbf{P}^{-1})_{i,j} = (\mathbf{P}^{-1})_{j,i} = s_i t_j + u_i v_j$ for $i \leq j$. An important implication of this result is that while inversion of an $N \times N$ matrix requires computing $O(N^2)$ distinct elements of the inverse, if the matrix has a symmetric, banded structure described above, these elements can

be deduced from $O(N)$ intermediate values such as $\{u_i\}$ and $\{v_j\}$.

Recursive inversion of symmetric block tri-diagonal matrices

Concus *et al.* [14] have presented a recursive algorithm to obtain the sequences $\{u_i\}$ and $\{v_j\}$ introduced above for a symmetric tri-diagonal matrix $\mathbf{T}$. Their result can be generalized to a *block* tri-diagonal, symmetric matrix $\mathbf{T}$ as follows:
Let

$$\mathbf{T} = \begin{bmatrix} A_1 & B_1^T & & & \\ B_1 & A_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & B_{m-1}^T \\ & & & B_{m-1} & A_m \end{bmatrix}. \tag{5.17}$$

Compute $\{U_i\}$ and $\{V_j\}$ recursively as

$$U_1 = I \tag{5.18}$$

$$U_2 = -B_1^{-T} A_1 \tag{5.19}$$

$$U_k = -B_{k-1}^{-T} \left(A_{k-1} U_{k-1} + B_{k-2} U_{k-2} \right), \quad k = 3, 4, \ldots, m \tag{5.20}$$

$$V_m = \left(A_m U_m + B_{m-1} U_{m-1} \right)^{-1} \tag{5.21}$$

$$V_k = \left(A_k U_k + B_{k-1} U_{k-1} \right)^{-1} \left[I - (U_k V_{k+1} B_k)^T \right], \quad k = m-1,\ m-2,\ \ldots,\ 2 \tag{5.22}$$

$$V_1 = \left(A_1 U_1 \right)^{-1} \left[I - (U_1 V_2 B_1)^T \right]. \tag{5.23}$$

Then, the $(i,j)^{\text{th}}$ block of $\mathbf{T}^{-1}$ is given by

$$\left(\mathbf{T}^{-1} \right)_{i,j} = \left(\mathbf{T}^{-1} \right)_{j,i} = U_i V_j, \quad i \leq j. \tag{5.24}$$

Proof: by substition into $\mathbf{T}^{-1}\mathbf{T} = \mathbf{T}\mathbf{T}^{-1} = \mathbf{I}$.

Enforcing the nearest-neighbor ($\mathcal{W}_1$) structure

Let us consider the problem of approximating the inverse of $\mathbf{T}$ as

$$\mathcal{W}_1 \odot \mathbf{T}^{-1}.$$

for the case where $\mathbf{T}$ has the "nearest-neighbor" structure (so that A_k's are tri-diagonal and B_k's are diagonal). We can specialize the above algorithm to obtain only the tri-diagonal blocks of $\mathbf{T}^{-1}$ as follows:

- Compute $\{U_k\}$ as above: Eq. (5.18)–(5.20).
- Initialize the backward recursion as:

$$\begin{aligned} V &= (A_m U_m + B_{m-1} U_{m-1})^{-1} && (5.25)\\ \left(\mathbf{T}^{-1}\right)_{m,m} &= U_m V && (5.26)\\ W &= U_{m-1} V && (5.27)\\ \left(\mathbf{T}^{-1}\right)_{m-1,m} &= W && (5.28)\\ \left(\mathbf{T}^{-1}\right)_{m,m-1} &= W^T. && (5.29) \end{aligned}$$

- For $k = m-1,\ m-2,\ \ldots,\ 2,$

$$\begin{aligned} V &= (A_k U_k + B_{k-1} U_{k-1})^{-1} \left[I - (W B_k)^T\right] && (5.30)\\ \left(\mathbf{T}^{-1}\right)_{k,k} &= U_k V && (5.31)\\ W &= U_{k-1} V && (5.32)\\ \left(\mathbf{T}^{-1}\right)_{k-1,k} &= W && (5.33)\\ \left(\mathbf{T}^{-1}\right)_{k,k-1} &= W^T. && (5.34) \end{aligned}$$

- Finally,

$$\left(\mathbf{T}^{-1}\right)_{1,1} = A_1^{-1}\left[I - (W B_1)^T\right]. \qquad (5.35)$$

Let the dimension of the 2-D spatial domain be $m \times n$ (i.e., $N = mn$), so that the block size is $n \times n$. Then, the computational complexity of this algorithm can be assessed from the m inversions of $n \times n$ matrices and $O(m)$ multiplications between $n \times n$ matrices of various degrees of sparseness, resulting in a total of $O(mn^3)$ flops. (Note that since $\mathbf{T}$ has a $\mathcal{W}_1$ structure, B_k's are diagonal.)

To complete the approximate inversion, we need to truncate the result from the above algorithm to a $\mathcal{W}_1$ matrix: the diagonal blocks, $(\mathbf{T}^{-1})_{k,k}$, are truncated to be tri-diagonal and the off-diagonal blocks, $(\mathbf{T}^{-1})_{k-1,k}$ and $(\mathbf{T}^{-1})_{k,k-1}$, are truncated to be diagonal.

Applying the algorithm to the information filter

An advantage of the algorithm just presented is that an exact truncated inverse matrix can be obtained in a finite number of steps, so that the information filter may be implemented without an additional approximation over that caused by the truncation of the information matrices. A serious disadvantage is, however, that the computational requirement is still substantial for most applications (except for a special case where the spatial domain is an elongated rectangle such that $n \ll m$).

5.3.3 Inversion by polynomial approximation

Let us consider taking an alternative approach to computing (5.16) on page 115 efficiently. Specifically, we express the matrix inverse in (5.16) as an infinite series of easily computable terms. We show that truncating the infinite series leads us to an efficient implementation of the information filter.

Diagonal dominance

We introduce here a matrix property important for convergence of the series representation of the inverse matrix in (5.16).

Definition 4 (cf. [35, 75]) A matrix A is said to be *diagonally dominant* if its elements a_{ij}, where i and j are the row and column indices, satisfy

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \quad \forall i.$$

If the inequality is a strict inequality, then A is said to be *strictly diagonally dominant.*

Polynomial approximation for the matrix inverse

Suppose we are interested in computing the inverse of a symmetric matrix $\mathbf{K}$. Let Δ be a diagonal matrix formed by the main diagonal of $\mathbf{K}$ and define Ω to be $\Omega \equiv \mathbf{K} - \Delta$, i.e., the off-diagonal part of $\mathbf{K}$. Then, $\mathbf{K}^{-1}$ can be obtained by the following infinite series

$$\begin{aligned} \mathbf{K}^{-1} = \; & \Delta^{-1} - \Delta^{-1}\Omega\Delta^{-1} \\ & + \Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1} - \Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1} \\ & + \Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1} - \Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1}\Omega\Delta^{-1} \\ & + \cdots \end{aligned} \tag{5.36}$$

which converges if all eigenvalues of $\Delta^{-1}\Omega$ reside within the unit disk, or equivalently, if $\mathbf{K}$ is strictly diagonally dominant [14, 75]. Convergence is especially fast when the eigenvalues of $\Delta^{-1}\Omega$ are close to zero, and taking the first few terms of the series makes a good approximation of the inverse. To compute the matrix inverse in (5.16) we set $\mathbf{K} = \mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)$. Then, since this $\mathbf{K}$ has a neighborhood structure the matrix Ω will be sparse, leading to a situation where the first several terms in the series are very sparse (in fact, they have certain $\mathcal{W}$-structures). The operations involved in computing the finite approximation of the series are, thus, all locally confined so that they are parallelizable.

The diagonal dominance characteristic of the matrix $\mathbf{K} = \mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)$ is not easy to verify analytically. We can speculate, however, that it is diagonally dominant because of the strong tendency for the matrix $\hat{\mathbf{L}}(t-1)$ to be diagonally

dominant. That is, in each time frame the predicted information matrix is updated by an addition of the measurement information matrix which is generally a diagonal and positive semidefinite matrix.

The finite series approximation already has a $\mathcal{W}$-structure; thus, it can be regarded as an approximate computation of (5.16), which can then be used in (5.15) to approximate the predicted information matrix. The computational complexity of such an approximation process is $O(N)$ because of the sparseness of the matrix Ω. Moreover, as mentioned before, the matrix structure makes every step of the computation parallelizable, providing us with a possibility for quite efficient implementation of (5.15).

5.3.4 A Suboptimal Information Filter

We propose a suboptimal information filter for the generalized Gauss-Markov system (5.1) (5.2) based on the polynomial inverse approximation described above. The effictiveness of the approximated filter is examined through some numerical experimentation.

The filtering algorithm

The optimal filter is given as (5.3)-(5.8). Given a masking matrix $\mathcal{W}$ to structurally constrain the information matrices, we obtain a suboptimal filter by replacing (5.3) of the optimal filter with:

1. Compute the $\mathcal{W}$-structured matrix $\mathbf{K} = \mathbf{A}^T\mathbf{U}\mathbf{A} + \hat{\mathbf{L}}(t-1)$.

2. Let Δ be a diagonal matrix whose diagonal is the same as the main diagonal of $\mathbf{K}$. Let $\Omega = \mathbf{K} - \Delta$.

3. Use the first several (a fixed number of) terms in the infinite series (5.36) to approximate $\mathbf{K}^{-1}$ as $(\mathbf{K}^{-1})_a$.

4. $\overline{\mathbf{L}}(t) = \mathcal{W} \odot \left[\mathbf{U} - \mathbf{U}\mathbf{A}^T (\mathbf{K}^{-1})_a \mathbf{A}\mathbf{U}\right]$.

As discussed before, this modification on the optimal filter reduces both the storage and computational requirements to $O(N)$.

Comment

The approximation method presented so far can be considered as a 3-D (space-time domain), temporal extension of the "reduced update filter" and "strip filter" approximations for the 2-D Kalman filters developed mainly for image restoration [78, 57, 79, 76]. In essence, these approximation techniques also aim to gain computational efficiency by constraining certain matrices in Kalman filter, such as the gain matrix, to be "local".

5.3.5 Numerical results

We apply the suboptimal information filter to the dynamic system used in the experiments in Section 5.2.3, (5.11) (5.12), to examine its performance. Again, we use the quantity (5.13) to measure the closeness of approximation of the information matrices. The two charts in Figure 5-5 show the approximation errors for the predicted and updated information matrices when different numbers of terms are used to approximate the infinite series (5.36). The filter parameters are $\alpha = \beta = 1$, and the structural constraint is $\mathcal{W}_2$. The six solid lines, from top to botton, shown in each chart represent the errors when the first one to six terms, respectively, in the series are used. The dashed line in each chart is the error for the "truncated information filter" of Section 5.2.2 (i.e., corresponding to using the complete infinite series and then truncating to the $\mathcal{W}_2$ neighborhood structure). As can be observed, as the number of terms increases the error approaches that corresponding to optimal truncation.

Number of terms and structural constraints

It appears that accuracy gained per addition of a term in the series dimishes as the number of terms in the series increases. Here, we determine such an effect for given structural constraints on the information matrices. The two charts in Figures 5-6

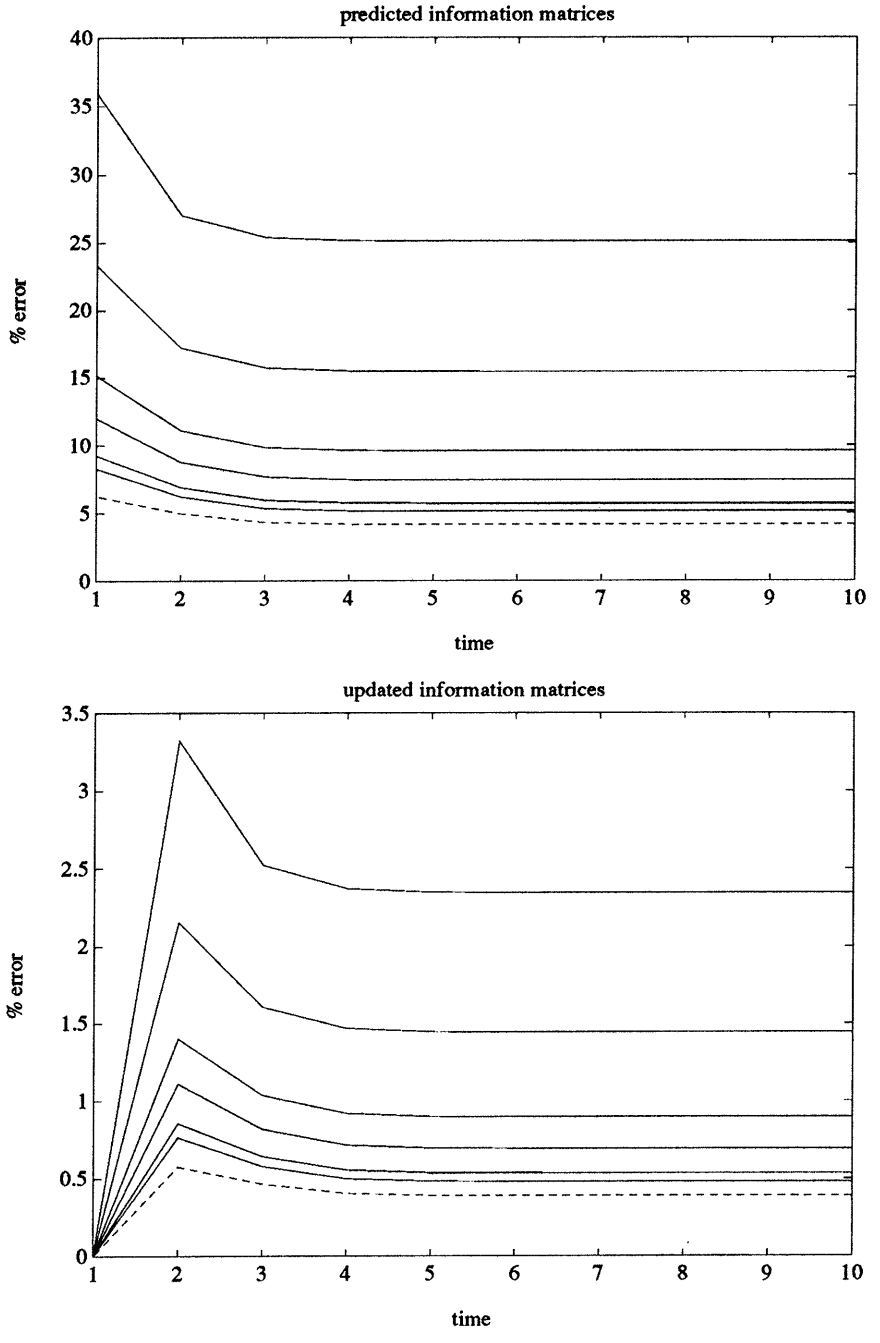


Figure 5-5: Performance of the suboptimal information filters using various number of terms in series approximation.

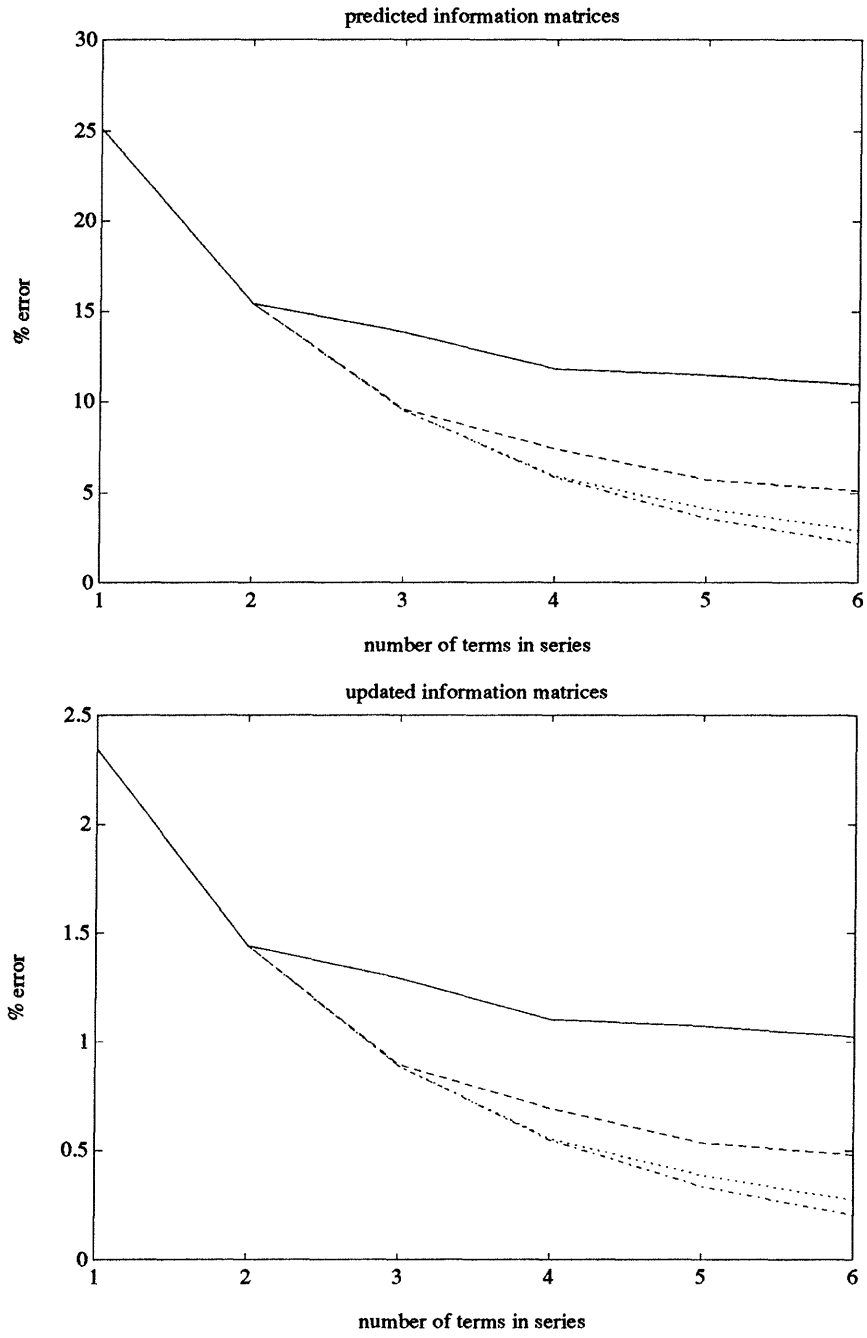


Figure 5-6: The performance of the series-approximated suboptimal information filters as a function of the number of terms in the series. The solid, dashed, and dotted lines correspond to the filters with $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$ structural constraints, respectively. The dash-dot lines correspond to the filter with no sturcutral constraint.

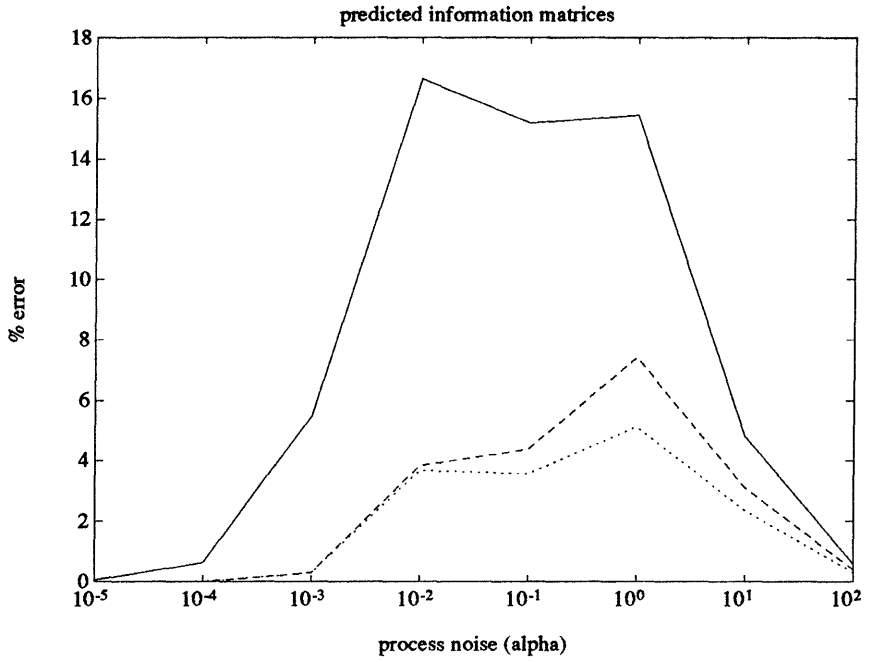


Figure 5-7: The effect of the process noise parameter on the approximation errors for the predicted information matrices using different numbers of terms in the series approximation — 2 (solid-line), 4 (dashed-line), and 6 (dotted-line).

show the approximation errors for the predicted and updated information matrices at $t = 10$ as a function of the number of terms in the series. ($\alpha = \beta = 1.$) The solid lines are the errors associated with a $\mathcal{W}_1$-structural constraint, while the dashed and dotted lines are those associated with $\mathcal{W}_2$ and $\mathcal{W}_3$-structural constraints, respectively. The dash-dot lines represent the errors when no structural constraint is applied. As can be observed, for a tighter structural constraint the gain in accuracy obtained by including more terms in the series levels off at an earlier point.

Effects of filter parameters

As in the numerical experiment in Section 5.2.3, the effects of the process and measurement noise parameters α and β on the suboptimal information filter are determined. Figures 5-7 and 5-8 show the errors at $t = 10$ when the number of terms in the series is 2 (solid lines), 4 (dash lines), and 6 (dotted lines). The structural constraint for the information matrices is $\mathcal{W}_2$. Generally, the results shown are similar to those presented in Section 5.2.3: the error curves as a function of α (Fig.'s 5-7 and 5-8)

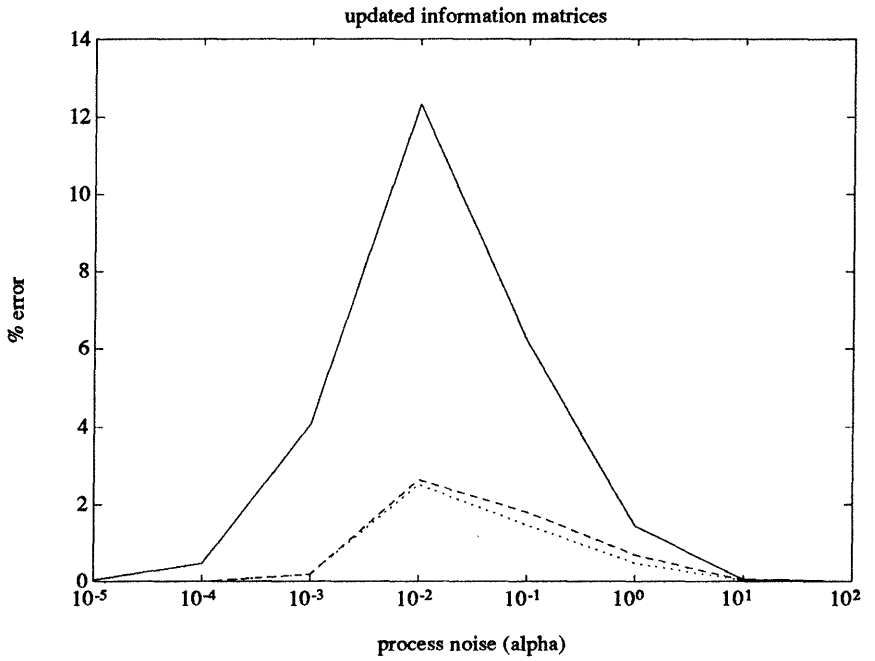


Figure 5-8: The effect of the process noise parameter on the approximation errors for the updated information matrices using different numbers of terms in the series approximation — 2 (solid-line), 4 (dashed-line), and 6 (dotted-line).

show unimodal patterns, and the error curves are monotonically increasing with β (Fig. 5-9).

Summary

A relatively small number of terms in the series (5.36) is sufficient for an effective approximation of the truncated information filter in Section 5.2.2. In particular, a tighter structural constraint $\mathcal{W}$ for the information matrix allows approximation by a smaller number of terms.

The qualitative effects of the model parameters, i.e., α and β, on the series approximated filter are similar to those on the optimally truncated filter. A particularly important characteristic of these filters is the sensitivity to the measurement noise — the measurement $\mathbf{g}(t)$ of the unknown field $\mathbf{f}(t)$ must be modeled to have a sufficiently high fidelity for the approximation techniques to work.

In this section we have shown that the series-approximated suboptimal information filter can be used to approximate the optimal *information matrix* efficiently. In

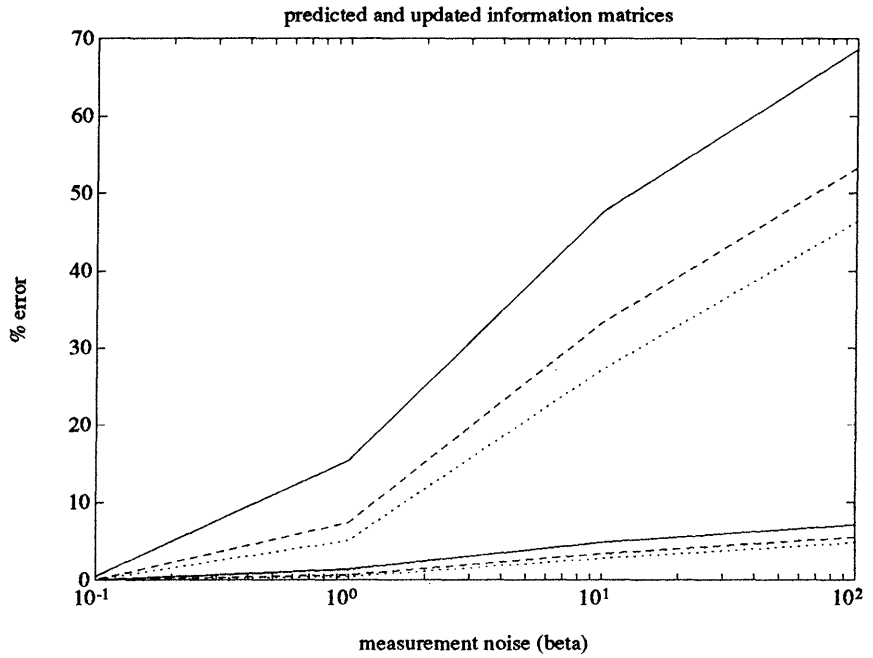


Figure 5-9: The effect of the measurement noise parameter on the approximation errors for the predicted and updated information matrices using different numbers of terms in the series approximation — 2 (solid-line), 4 (dashed-line), and 6 (dotted-line). The top three lines are associated with the predicted information matrices, while the bottom three lines with the updated information matrices.

Section 5.5 we will present numerical results on how well this and other suboptimal filters produce *estimates* of the visual field $\mathbf{f}(t)$.

5.4 Approximating the SRIF

In this section we present an efficient, iterative implementation of the square root information filter (SRIF) for the extended Gauss-Markov system (5.1) (5.2). Recall that for explicitness of presentation we have assumed that $\mathbf{f}(t)$ is a scalar field defined over a 2-D space and that $N \equiv \dim(\mathbf{f}(t))$. Although the general optimal filtering algorithm has been presented in Section 4.4.4, let us highlight its key operations for our specific case:

- *Unitary transformations for propagation of the SRI-pair.*
  To update the SRI-pair from the previous frame, $\left\langle\!\left\langle\, \hat{\mathbf{w}}(t-1), \hat{\mathbf{L}}^{1/2}(t-1) \,\right\rangle\!\right\rangle^{1/2}$, by propagating it through the system equations (5.1) (5.2), we have formulated three stages of unitary transformation problems in which the lower-left $N \times N$ block of each operand matrix is to be nulled, i.e.,

$$\begin{bmatrix} \hat{\mathbf{L}}^{1/2}(t-1) & \mathbf{0} & \hat{\mathbf{w}}(t-1) \\ -\mathbf{U}^{1/2}\mathbf{A} & \mathbf{U}^{1/2} & 0 \end{bmatrix} \rightarrow \begin{bmatrix} V_1 & V_2 & v_1 \\ \mathbf{0} & V_3 & v_2 \end{bmatrix} \tag{5.37}$$

$$\begin{bmatrix} V_3 & v_2 \\ \mathbf{S} & 0 \end{bmatrix} \rightarrow \begin{bmatrix} V_4 & v_3 \\ \mathbf{0} & v_4 \end{bmatrix} \tag{5.38}$$

$$\begin{bmatrix} V_4 & v_3 \\ \mathbf{N}^{1/2}\mathbf{H} & \mathbf{N}^{1/2}\mathbf{g} \end{bmatrix} \rightarrow \begin{bmatrix} V_5 & v_5 \\ \mathbf{0} & v_6 \end{bmatrix} \tag{5.39}$$

  where V_k and v_k are an $N \times N$ matrix and an N-vector, respectively, for $k = 1, 2, \ldots$, and $\mathbf{U}, \mathbf{A}, \mathbf{S}, \mathbf{N}$, and $\mathbf{H}$ are $N \times N$ matrices obtained from the components of the system equations (5.1) (5.2) while $\mathbf{g}$ is the N-dimensional measurement vector. The updated SRI-pair is given by $\hat{\mathbf{w}}(t) = v_5$ and $\hat{\mathbf{L}}^{1/2}(t) = V_5$.

- *Inverse problem for updating the estimate.*
  The estimate for the visual field $\hat{\mathbf{f}}(t)$ is obtained from the updated SRI-pair as

the solution of the inverse problem

$$\widehat{\mathbf{L}}^{1/2}(t)\widehat{\mathbf{f}}(t) = \widehat{\mathbf{w}}(t). \tag{5.40}$$

5.4.1 Approximation strategy

The computational and storage demands for the optimal SRIF are as follows:

1. The N^2 elements of the square root information (SRI) matrix must be stored.
2. Propagation of the SRI-pair costs $O(N^3)$ flops per time frame.
3. The updated SRI matrix must be inverted to obtain the estimate at each time frame.

The approximation strategy for SRIF is fundamentally the same as that used for the information filter discussed previously. That is:

- Restrict the SRI matrix to have a neighborhood structure so that it is represented by $O(N)$ elements. This relieves the storage problem (Item 1 above) tremendously. It also facilitates iterative inversion of the updated SRI matrix (Item 3), especially with a multigrid method.
- Find an efficient way to compute such a masked (approximated) SRI matrix to reduce the computational complexity associated with propagation of the SRI-pair (Item 2). Specifically, we are aiming to develop a computational scheme that is *iterative* and *parallelizable* — two features that make the scheme suitable for a modular hardware (e.g., VLSI) implementation.

In this section, we first address the efficiency issue in propagation of the SRI-pair by presenting an iterative scheme which converges to the optimal square root filter in the limit. This scheme is parallelizable, and if implemented in parallel each iteration costs $O(N)$ flops. We then consider imposing the neighborhood structural constraints on the SRI matrix. Representing the SRI matrix by a sparse, $\mathcal{W}$-structured matrix

allows us to approximate the iterative SRI-pair propagation scheme so that its cost per iteration is reduced to $O(1)$ flops.

The square-root of a square matrix is not uniquely defined. In a traditional SRIF algorithm [9], the SRI matrices are the upper triangular square-roots of the information matrices. As we will see, a notable feature of the iterative SRIF presented in this section is that the SRI matrices are not triangular.

5.4.2 Iterative propagation of the SRI-pair

The basic mechanism for propagation of the SRI-pair involves nulling of certain submatrices by a unitary operation. In particular, propagation of the SRI-pair is achieved by nulling the lower-left $N \times N$ blocks of the aggregate matrices shown on the left hand sides of (5.37) (5.38) (5.39).

QR Factorization

The QR factorization [18] is a systematic algorithm to perform the nulling required in a traditional formulation of SRIF's [9]. It nulls the columns of the target submatrix sequentially from left to right. Although it is a standard technique used in SRIF's, its computational complexity, $O(N^3)$ flops, is too high for filtering of large dimensional visual fields and it is not implementable in parallel.

Givens Rotation

The Given rotation is a rudimentary unitary operation that nulls a specific matrix element, and QR factorization can be considered as a precisely ordered sequence of Givens rotations [18, 70]. A single Givens rotation operates on only a pair of rows in a matrix at a time; thus, Givens rotations operating on disjoint pairs of rows can be performed simultaneously. A clever scheduling of rotations can, therefore, lead to a parallelizable nulling scheme.

The principle behind the Givens rotation is that for any pair of real numbers, p

and q, there exists a rotation θ such that

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} p' \\ 0 \end{bmatrix},$$

i.e., an element q in a vector can be nulled *against* another element p by a left-multiplication by a unitary rotation matrix.

Now, suppose we have partitioned an arbitrary $2N \times N$ matrix into an upper $N \times N$ block P and a lower $N \times N$ block Q as

$$\begin{bmatrix} P \\ Q \end{bmatrix},$$

and suppose we want to null an element q_{ij} in the lower block against an element in P. Any matrix element in the j^{th} column can be used to null q_{ij} with a Givens rotation. Let us denote such an element in block P by p_{kj}, and let the rotation that nulls q_{ij} against p_{kj} be θ. Then a left-multiplication on the matrix by the following unitary operator, which is an identity matrix with modifications on rows and columns k and i, will null the element q_{ij}:

$$\begin{bmatrix}
1 & & & & & & & & & \\
& \ddots & & & & & & & & \\
& & 1 & & & & & & & \\
& & & C & & & & S & & \\
& & & & 1 & & & & & \\
& & & & & \ddots & & & & \\
& & & & & & 1 & & & \\
& & & -S & & & & C & & \\
& & & & & & & & 1 & \\
& & & & & & & & & \ddots \\
& & & & & & & & & & 1
\end{bmatrix}$$

where $C \equiv \cos\theta$ and $S \equiv \sin\theta$. Note that all the elements in rows k and i of the

operand matrix are modified as the result of this multiplication, while the remaining rows are left unmodified.

Submatrix nulling by parallel rotations

Let us now consider the problem of nulling an entire submatrix by a unitary transformation, i.e.,

$$\begin{bmatrix} P \\ Q \end{bmatrix} \rightarrow \begin{bmatrix} P' \\ \mathbf{0} \end{bmatrix} \tag{5.41}$$

where P and Q are previously defined $N \times N$ blocks of an arbitrary $2N \times N$ matrix. In essence, this is an abstraction of the main computational problem involved in propagation of the SRI-pair. The solution, i.e., the unitary transformation, for the nulling problem (5.41) is directly applicable to the stages of the SRI-pair propagation operations (5.37) (5.38) (5.39) when P and Q are appropriately selected.

We present here an iterative algorithm to perform this nulling operation. Motivated by computational efficiency, our algorithm makes use of selective nulling by Givens rotations, similar to that used in the Jacobi method for eigenvalue computations[2] [15, 18, 70], which can lead to parallel implementation.

Let us call the $N \times N$ lower block the *eliminatee* block and the $N \times N$ upper block the *elminator* block. A *diagonal band* of the eliminatee block Q is formally defined as the set of elements $q_{j+b,j}$ for $j \in \mathcal{J}_b$ given a fixed integer $b \in [-N+1, N-1]$, where $\mathcal{J}_b \equiv [1, N] \cap [1-b, N-b]$. The value of b determines the location of the band in Q. For example, $b = 0$ specifies the main diagonal, while $b = 1$ and -1 specify the upper and lower first off-diagonal bands, respectively.

Our proposed algorithm calls for nulling of elements in a diagonal band of Q against the elements in the *main diagonal* of the eliminator block P. Note that such band elements can be nulled simultaneously, i.e., in parallel. That is, the set of Givens rotations associated with nulling the band element $q_{j+b,j}$ (for a fixed b) against p_{jj}

[2] as well as other iterative methods to reduce matrix structure by selective nulling of elements (see [70] for details)

for every $j \in \mathcal{J}_b$ operates on disjoint pairs of rows — row j of P and row $j+b$ of Q for $j \in \mathcal{J}_b$. This simultaneous nulling of a band in the eliminatee block constitutes a fundamental operating unit on which our iterative SRIF algorithm is based.

Let us further define a *sweep*[3] to be a sequential set of the nulling operations on bands so that *every* diagonal band (i.e., $\forall b \in [-N+1, N-1]$) in the eliminatee block Q is nulled once. This process essentially nulls every element in Q once in a systematic fashion. We do not, at this point, worry about the order in which the diagonal bands are nulled. Note, however, that a band which has been nulled previously can become non-zero again while another band is being nulled. Before describing the process of sweep in any more detail, let us present a key result:

Theorem 5.1 *The elminatee block Q is completely nulled in the limit by iterations of sweeps.*

A sketch of the proof of this theorem is provided in Appendix B. The number of iterations for a satisfactory convergence (e.g. satisfying $\|Q\| < \epsilon$ for an $\epsilon > 0$) varies with the contents of the blocks P and Q in general. As we will see, for visual reconstruction problems, reasonably good estimates can be obtained by iterating less than N times. Note that when implemented in parallel, the computational cost of a sweep (i.e., an iteration) is $O(N)$ flops.

An illustration of sweep

Let us illustrate the proposed nulling process using a simple example. Suppose that $N = 5$ and both P and Q are tri-diagonal. In this example, the diagonal bands of Q are nulled by the following specific order:

1. The main diagonal, i.e., $b = 0$.
2. The upper (right) first off-diagonal, $b = 1$.
3. The lower (left) first off-diagonal, $b = -1$.

[3] named after a similar operation in the iterative Jacobi method [15, 18]

4. The upper second off-diagonal, $b = 2$.

5. The lower second off-diagonal, $b = -2$.

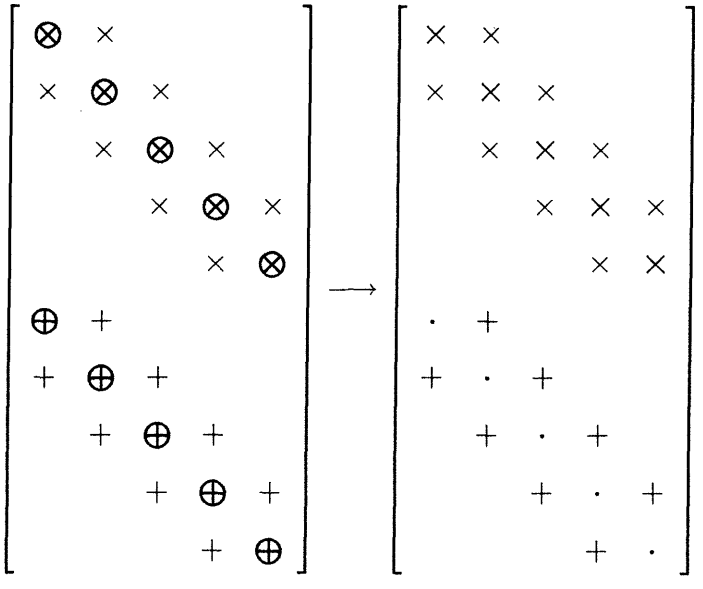
6. The upper third off-diagonal, $b = 3$.

7. The lower third off-diagonal, $b = -3$.

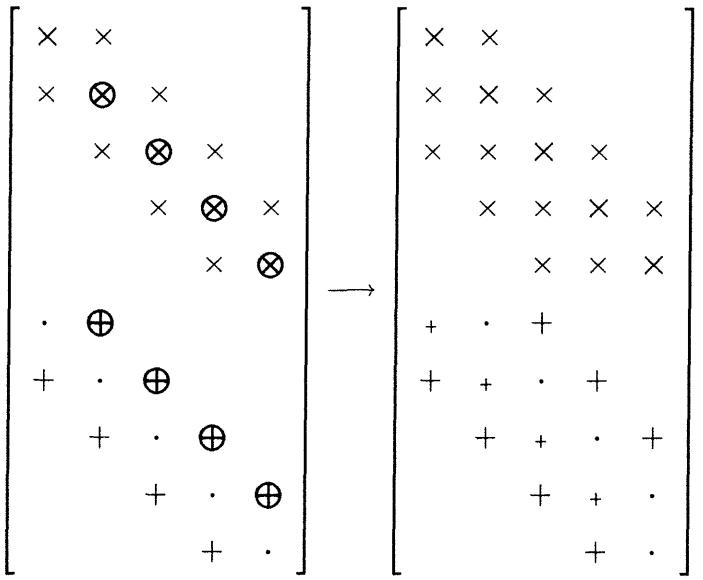
8. The upper fourth off-diagonal, $b = 4$.

9. The lower fourth off-diagonal, $b = -4$.

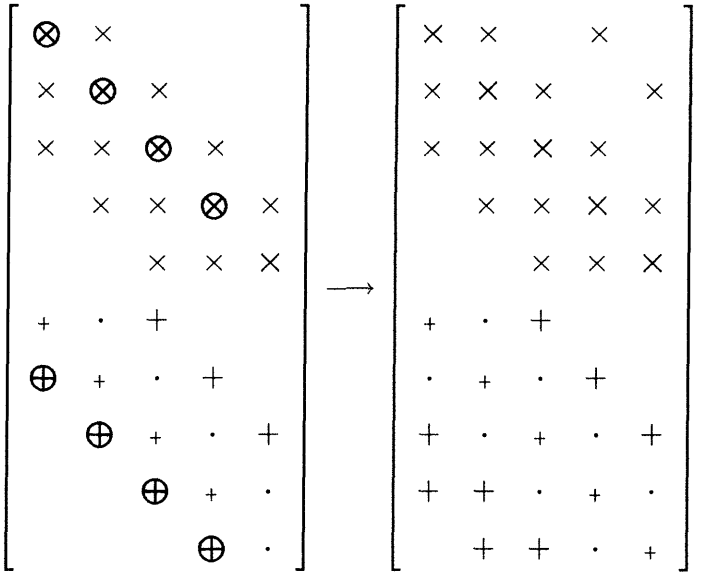
Step 1 above is pictorally presented below. Here we show non-zero elements of the 5×5 eliminator and eliminatee blocks. The $\times$'s and $+$'s denote the locations of the non-zero elements in the eliminator and eliminatee blocks, respectively. An element $\oplus$ is nulled against $\otimes$ in the same column. The nulled elements are denoted by small dots, $\cdot$'s.



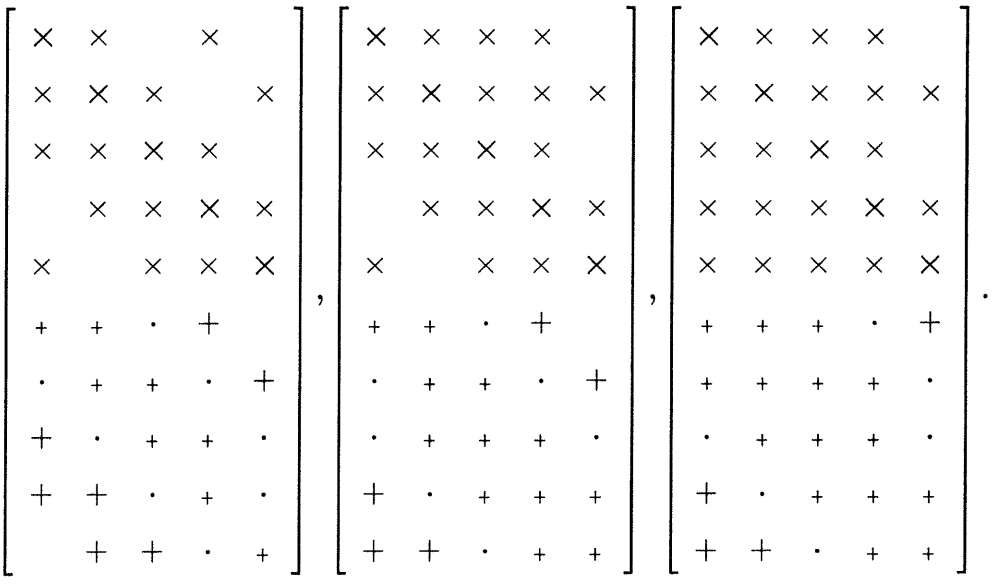
Using the same notations, Step 2 is illustrated below.



Note that after Step 2, the eliminatee main diagonal, which has been nulled previously, becomes non-zero again (depicted by smaller +'s). Also note the appearence of an additional non-zero band in each of eliminator and eliminatee blocks. Such addition of non-zero bands can also be seen in Step 3:



Continuing the process, we illustrate the structures of the matrix after Steps 4, 5, and 6:



A sweep for this particular matrix is complete after Step 9, by which time every element in the eliminatee block Q has been nulled exactly once. The eliminator block P, which has started as a tri-diagonal matrix, is a full matrix after a sweep. Since every element in Q is nulled against a main diagonal element in P, a sweep in essence is a systematic attempt to transfer the energy (i.e., the square-sum of the elements) in Q to the main diagonal of P. In fact, as shown in Appendix B, the norm of the main diagonal of P is non-decreasing throughout a sweep, and despite losing the initial sparse structure the eliminator block typically ends up storing larger portion of its energy in the main diagonal after several sweeps.

Incomplete sweeps

Let us now consider applying iterations of sweeps to a matrix. When performing iterative nulling on a matrix with sparsely banded blocks such as the matrix above before Step 1, it is often advantageous computationally to null only a subset of the diagonal bands in the eliminatee block. We call such an iteration an *incomplete sweep.* In the illustration above, an iteration that terminates before Step 9 is an example of incomplete sweep.

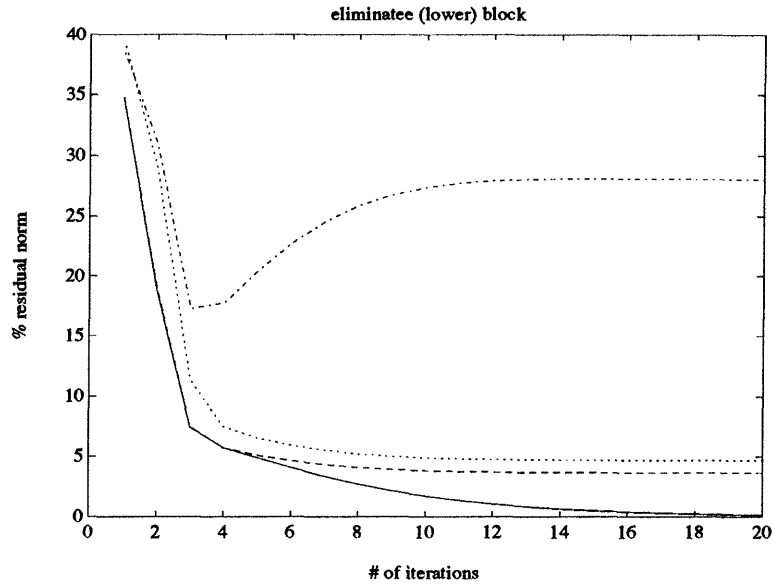


Figure 5-10: The reduction of the Frobenius norm of Q.

Let us examine convergence of the iterative nulling procedure numerically. In particular, we iterate the sweep illustrated above on the following 10×5 matrix:

$$\begin{bmatrix} P \\ Q \end{bmatrix} = \begin{bmatrix} 1 & 2 & & & \\ 3 & 4 & 5 & & \\ & 6 & 7 & 8 & \\ & & 9 & 1 & 2 \\ & & & 3 & 4 \\ 5 & 6 & & & \\ 7 & 8 & 9 & & \\ & 1 & 2 & 3 & \\ & & 4 & 5 & 6 \\ & & & 7 & 8 \end{bmatrix}.$$

In addition, we consider iterating the incomplete sweeps that perform nulling only up to Step 3, 5, and 7. Note that these incomplete sweeps correspond to constraining the *extent of sweep* by a 1-D versions of $\mathcal{W}_1$ (i.e., tri-diagonal), $\mathcal{W}_2$ (penta-diagonal), $\mathcal{W}_3$ (septa-diagonal) structures, respectively. Figure 5-10 shows the ratios of the Frobe-

nius norms of Q before and after a certain number of iterations of these incomplete and complete sweeps. Note that all except the $\mathcal{W}_1$-constrained incomplete sweep (dash-dot line in the figure) have performed well in the sense that less than 10% of the initial energy in Q remains after the fourth iteration. Nulling with the complete sweep (solid line) reduces the energy quickly down to about 7% in the first three iterations and then gradually down to near-zero by the 20^{th} iteration. The reduction of energy for incomplete sweeps where the extents of sweep are $\mathcal{W}_2$ (dotted line) and $\mathcal{W}_3$ (dashed line) seem to level off at around 5% and 4%, respectively, by the 10^{th} iteration.

An iterative SRIF

The iterative nulling procedure (Theorem 5.1) is directly applicable to the SRIF algorithm. That is, for the matrix in the left hand side of each of (5.37) (5.38) (5.39), we can let the $N \times N$ upper-left submatrix to be the eliminator block and the $N \times N$ lower-left submatrix to be the eliminatee block. The nulling algorithm offers a parallelizable scheme to propagate the SRI-pair optimally by iterations; however, we still need to address the issues of reducing the storage requirement for the SRI matrix and solving the inverse problem (5.40) (to obtain the estimate $\widehat{\mathbf{f}}(t)$) efficiently. We examine these issues in the next subsection.

Let us comment that the iterative nulling procedure or a modified version of it can be applicable to a general SRIF problem, and it offers an alternative to the traditional filtering scheme based on QR factorization. The iterative procedure, therefore, might merit further study in the context of general square root filtering theory although such study is out of context of this thesis.

5.4.3 Approximating the iterative SRIF

Truncating the SRI matrix

As mentioned in the previous chapter, an SRI-pair implies a spatial model for the corresponding visual field. Specifically, for $\mathbf{f} \sim \left\langle\!\left\langle\, \mathbf{w}, \mathbf{L}^{1/2} \,\right\rangle\!\right\rangle^{1/2}$ we have

$$\mathbf{L}^{1/2}\mathbf{f} = \lambda, \quad \lambda \sim (\, \mathbf{w}, \mathbf{I} \,). \tag{5.42}$$

As in the design of the approximate information filter, this motivates us to approximate the SRI matrix by truncation — a spatially local model *should* be able to approximate (5.42) effectively. Truncating the SRI matrix to a $\mathcal{W}$-structured matrix reduces the storage requirement from $O(N^2)$ to $O(N)$ and allows us to solve the inverse problem (5.40) efficiently (e.g., by a multigrid method).

An SRI matrix corresponding to an information matrix is not unique. As we intend to truncate the SRI matrix in the iterative SRIF algorithm, it is desirable that an SRI matrix computed using the iterative nulling procedure described above has most of its energy concentrated within a given $\mathcal{W}$-structure so that truncation has little effect on the quality of the corresponding estimate. A seemingly effective approach (judging from the results of the numerical examples to be shown) is to impose a condition on the equations (5.37) (5.38) (5.39) such that all their $N \times N$ submatrices, including $-\mathbf{U}^{1/2}\mathbf{A}, \mathbf{U}^{1/2}, \mathbf{S}, \mathbf{N}^{1/2}\mathbf{H}$, and the intermediate SRI matrices for the field, have the $\mathcal{W}$-structure prior to nulling. In another words, this condition enforces the matrices in the left hand sides of (5.37) (5.38) (5.39) to be sparsely banded except for their respective right-most columns. The idea behind this approach is that by confining energies of the $N \times N$ submatrices in the desired $\mathcal{W}$-structure and focusing the sweep procedure around the bands that belong to this $\mathcal{W}$-structure (hence impeding spread of energy to undesired bands) we should be able to preserve most of the energy within the $\mathcal{W}$-structures of the submatrices throughout the nulling procedure.

Symmetric square roots

Let us consider, then, how well the matrices $-\mathbf{U}^{1/2}\mathbf{A}, \mathbf{U}^{1/2}, \mathbf{S}$, and $\mathbf{N}^{1/2}\mathbf{H}$ can be represented by $\mathcal{W}$-structured matrices. Since $\mathbf{N}$ and $\mathbf{H}$ are diagonal matrices and $\mathbf{A}$ is already a $\mathcal{W}$ matrix, we focus our attention on $\mathbf{U}^{1/2}$ and $\mathbf{S}$. Recall from the previous chapter that these two matrices are *computed* as square roots of the information matrices $\mathbf{U}(t) \equiv \mathbf{B}^T\mathbf{Q}^{-1}(t)\mathbf{B}$ and $\sum_i \mu_i \mathbf{S}_i^T\mathbf{S}_i$ which are $\mathcal{W}$ matrices. As any of the square roots of an arbitrary sparse matrix is not sparse in general, we expect $\mathbf{U}^{1/2}$ and $\mathbf{S}$ to be full matrices. Since these square root matrices are to be truncated to a given $\mathcal{W}$-structure, ideally we select a form of square root that tends to concentrate energy in the $\mathcal{W}$-structure. Since a $\mathcal{W}$ matrix is structurally symmetric, we have chosen to use symmetric square roots for $\mathbf{U}^{1/2}$ and $\mathbf{S}$ and have obtained good numerical results. A symmetric square root is uniquely defined for a symmetric matrix [35], and the symmetric square root of a large positive semi-definite matrix can be approximated efficiently as detailed in Appendix C.

An approximate iterative SRIF

Reflecting the considerations indicated so far, we have implemented the following approximate SRIF algorithm which, as we will see, has performed effectively in numerical experimentation. The algorithm propagates the SRI-pair as in (5.37), (5.38), and (5.39) using an approximation of the iterative nulling procedure with the following features:

- *Truncation of the blocks.*

  The $N \times N$ submatrices in the left hand sides of the equations are truncated by $\mathcal{W}_{\ell_1}$ for a given ℓ_1 before each nulling procedure.

- *Incomplete sweep.*

  In each iteration, rather than performing a complete sweep, only the bands in $\mathcal{W}_{\ell_2}$ are nulled. We choose the "extent of sweep" ℓ_2 to be at least as large as ℓ_1.

- *Fixed number of iterations.*

  In each nulling session, a fixed number of iterations of incomplete sweep is

performed.

Symmetric square roots are used to compute the parameter matrices $\mathbf{U}^{1/2}$ and $\mathbf{S}$. Then, for each t the (approximate) updated SRI-pair from (5.39) is used to compute the estimate as in (5.40). As $\widehat{\mathbf{L}}^{1/2}(t)$ is truncated to be a $\mathcal{W}_{\ell_1}$ matrix, this is an inverse problem involving a sparse operator and can be performed efficiently.

5.4.4 Numerical results

We apply the approximate iterative SRIF to the dynamic system used in the experiments in Section 5.2.3, (5.11) (5.12), to examine its performance. We measure the closeness of approximation by squaring the computed SRI matrices and comparing them to the optimal information matrices using the criterion (5.13). Recall that for this particular problem $N = 100$.

The two graphs in Figure 5-11 show the approximation errors for the predicted and updated information matrices, respectively, when various incomplete sweeps are used for nulling. Three iterations of these sweeps have been used in each nulling session. The filter parameters are $\alpha = \beta = 1$, and no structural constraint has been applied to the blocks, i.e., $\ell_1 = \infty$ or more precisely $\ell_1 \geq 2N - 2$, leading to a masking matrix $\mathcal{W}_{\ell_1}$ whose elements are all ones. The five solid lines, from top to botton, shown in each graph represent the errors corresponding to $\ell_2 = 1, 2, 3, 4$, and 5, respectively. As can be observed, as the extent of sweep increases the error decreases.

Extent of a sweep, number of iterations, and structural constraints

The two graphs in Figure 5-12 show the approximation errors for the predicted and updated information matrices, respectively, at $t = 10$ as functions of ℓ_2 with $\alpha = \beta = 1$ and $\ell_1 = \infty$. The solid lines are the errors when one iteration per nulling session is applied, while the dashed, dotted, and dash-dot lines correspond with the cases for two, three, and four iterations per session, respectively. The curves for the three and four sweeps almost overlap each other in each graph. The graphs indicate that two or three iterations per nulling session is sufficient to achieve the near-best performance

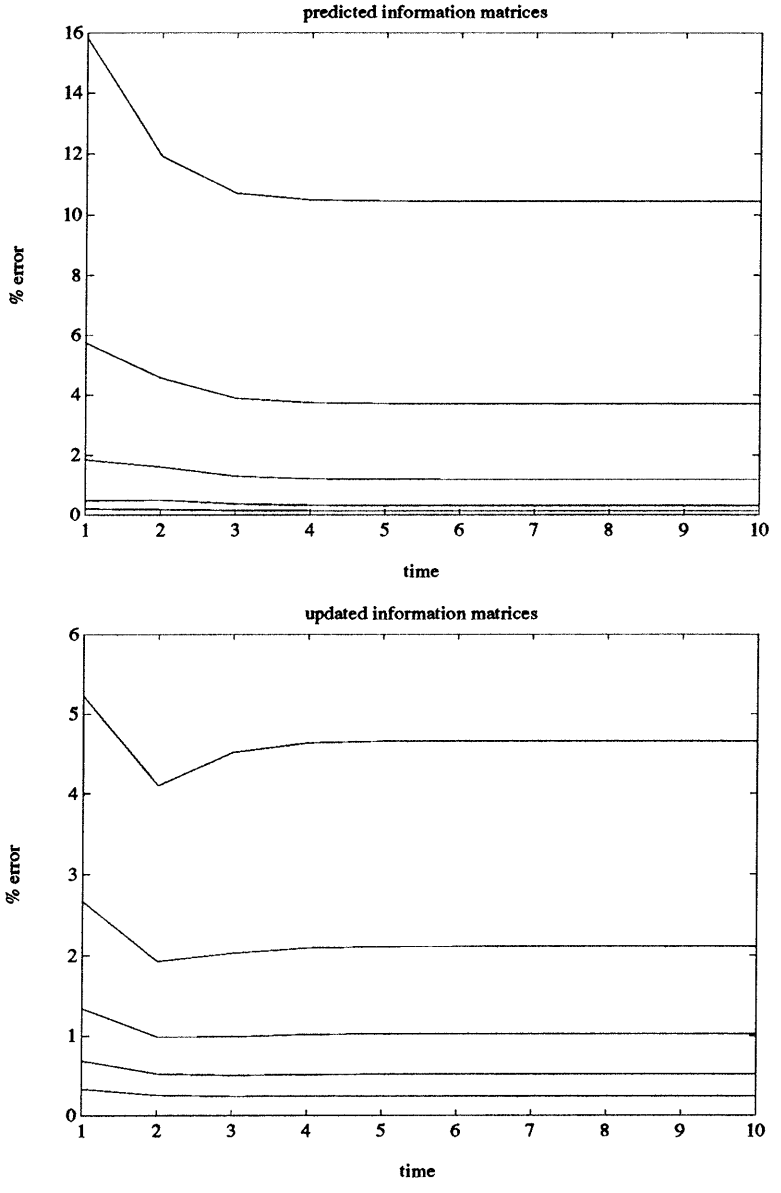


Figure 5-11: Performance of the approximate iterative SRIF using various incomplete sweeps, i.e., $\ell_2 = 1, 2, 3, 4, 5$.

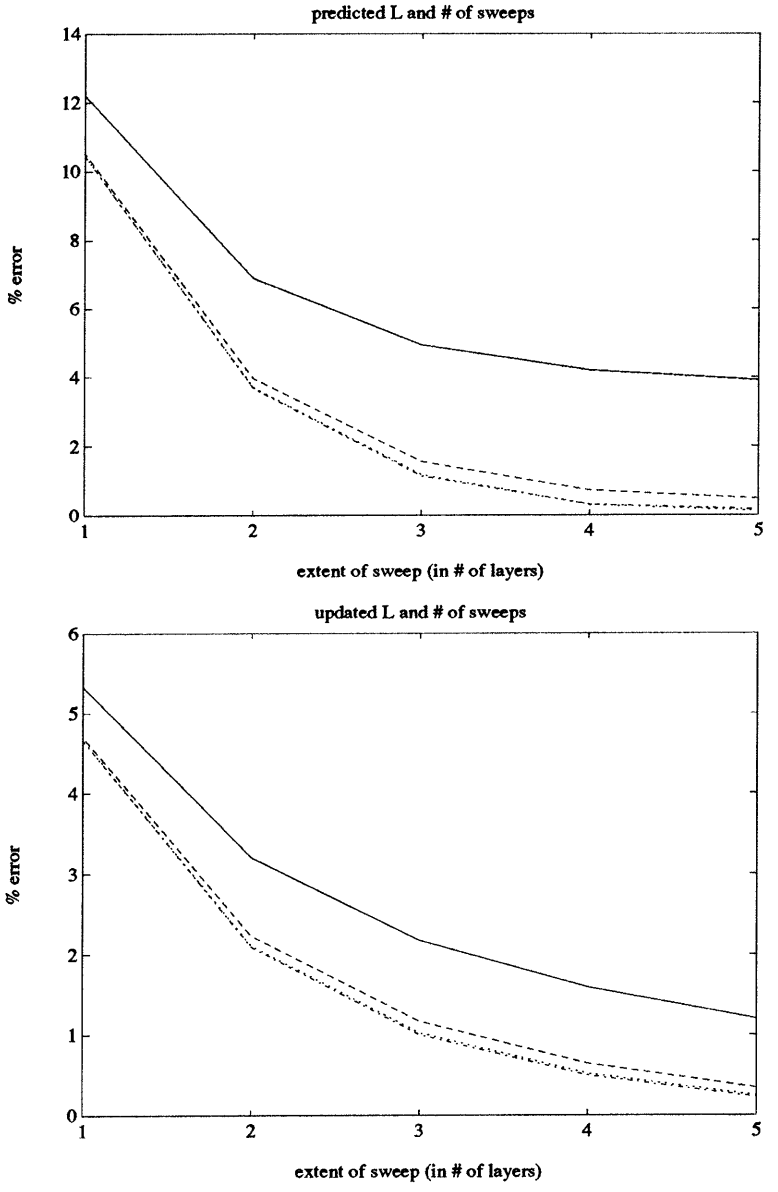


Figure 5-12: Performance of the approximate iterative SRIF when the number of iteration per nulling session is 1 (solid lines), 2 (dashed lines), 3 (dotted lines), and 4 (dash-dot lines) as funcations of ℓ_2.

in the given situation.

Let us consider the effects of structural constraints, i.e., ℓ_1, on the blocks. The two graphs in Figure 5-13 show the approximation errors for the predicted and updated information matrices, respectively, at $t = 10$ as functions of ℓ_2. ($\alpha = \beta = 1$.) Three iterations are used for each nulling session. The solid lines are the errors associated with an $\mathcal{W}_1$-structural constraint ($\ell_1 = 1$), while the dashed and dotted lines are those associated with $\mathcal{W}_2$ and $\mathcal{W}_3$-structural constraints ($\ell_1 = 2$ and $\ell_1 = 3$), respectively. The dash-dot lines represent the errors when no structural constraint ($\ell_1 = \infty$) is applied. These graphs show that for a given structural constraint $\mathcal{W}_{\ell_1}$, extending the incomplete sweep beyond the ℓ_1^{th} layer (i.e., using a value of ℓ_2 strictly larger than ℓ_1) does not lead to a large gain in accuracy.

Effects of filter parameters

As in Section 5.2.3, the effects of the process and measurement noise parameters α and β on the approximate iterative SRIF are examined. Figures 5-14, 5-15, and 5-16 show the errors when the number of sweeps in each nulling session is 1 (solid lines), 2 (dash lines), and 3 (dotted lines). The structural constraint is $\mathcal{W}_2$ for each case. As in the case with the approximated information filters, the performance of the approximate SRIF degrades as the magnitude of the measurement noise β is increased (Fig. 5-9). The effects of the process noise α (Fig.'s 5-14 and 5-15), however, are relatively modest.

Summary

Incomplete sweeps have been shown to be effective in iterative SRIF. In particular, for a given structural constraint $\mathcal{W}_{\ell_1}$ for the SRI matrix, we can choose the extent of sweep ℓ_2 to be $\ell_2 = \ell_1$. A further increase in the value of ℓ_2 does not lead to gain in accuracy.

Only a small number of iterations per nulling session is sufficient to propagate the SRI matrix (or its truncated version) with high accuracy. In particular, a near-best performance has been obtained using only 3 iterations per sessions.

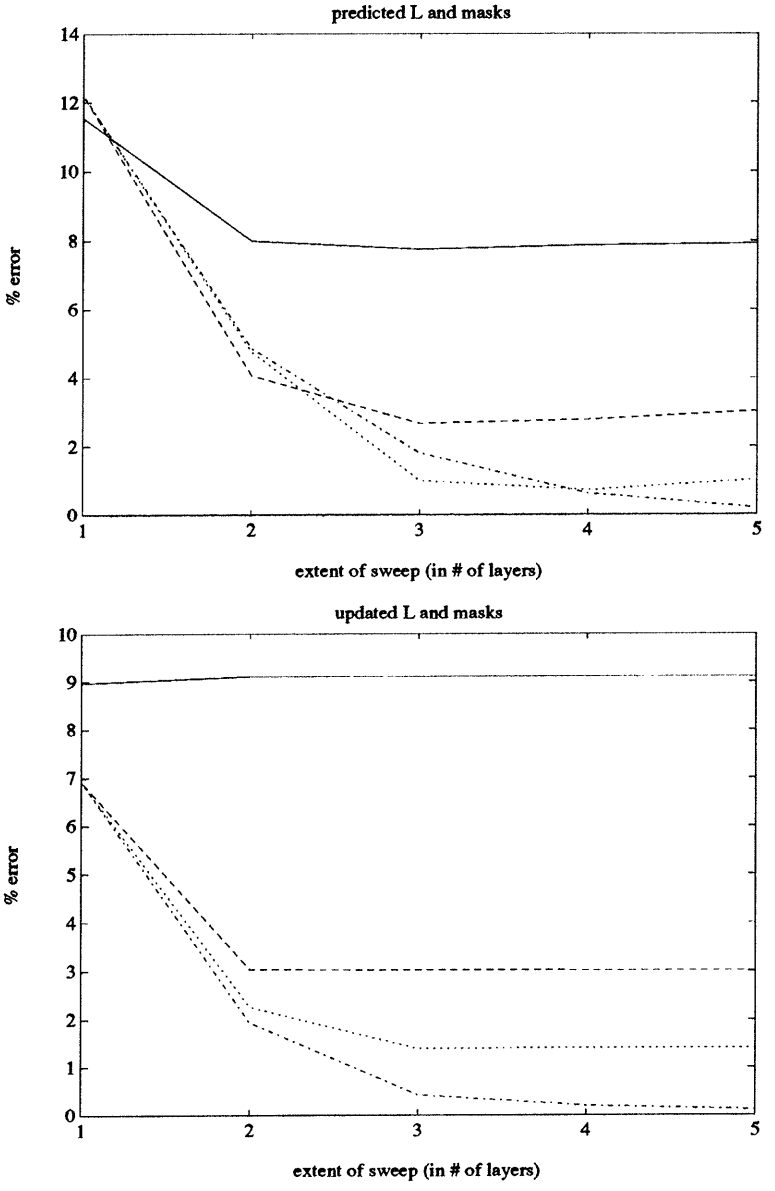


Figure 5-13: Performance of the approximate iterative SRIF for $\ell_1 = 1$ (solid lines), 2 (dashed lines), and 3 (dotted lines) as well as when there is no structural constraint (dash-dot lines) as functions of ℓ_2.

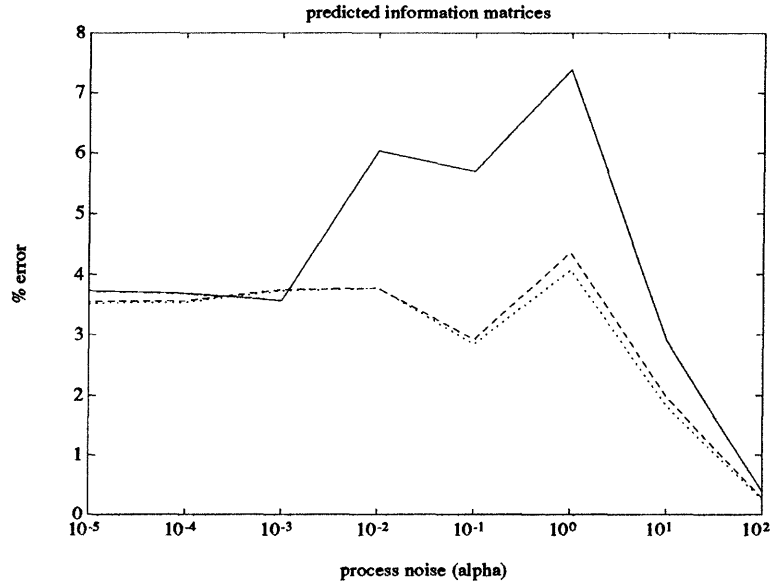


Figure 5-14: The effect of the process noise on the approximation errors for the predicted information matrices for different numbers of iterations per nulling session — 1 (solid line), 2 (dashed line), 3 (dotted line).

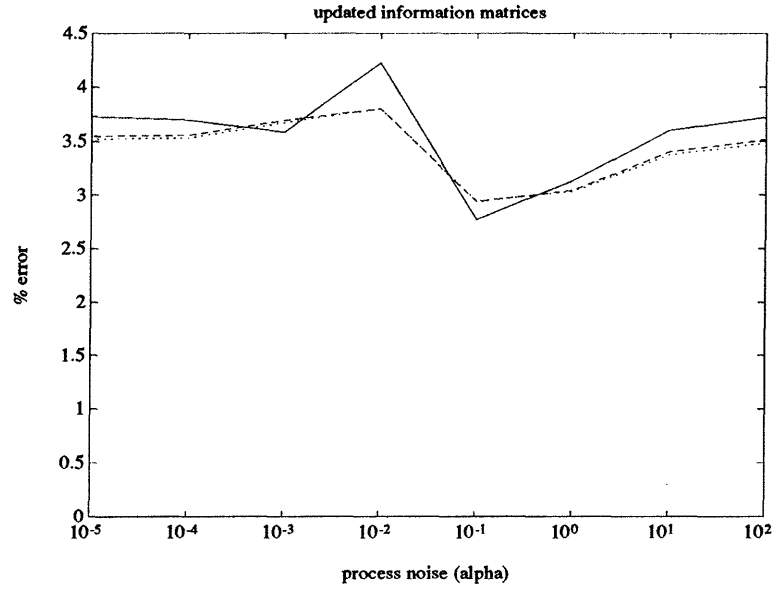


Figure 5-15: The effect of the process noise on the approximation errors for the updated information matrices for different numbers of iterations per nulling session — 1 (solid line), 2 (dashed line), 3 (dotted line).

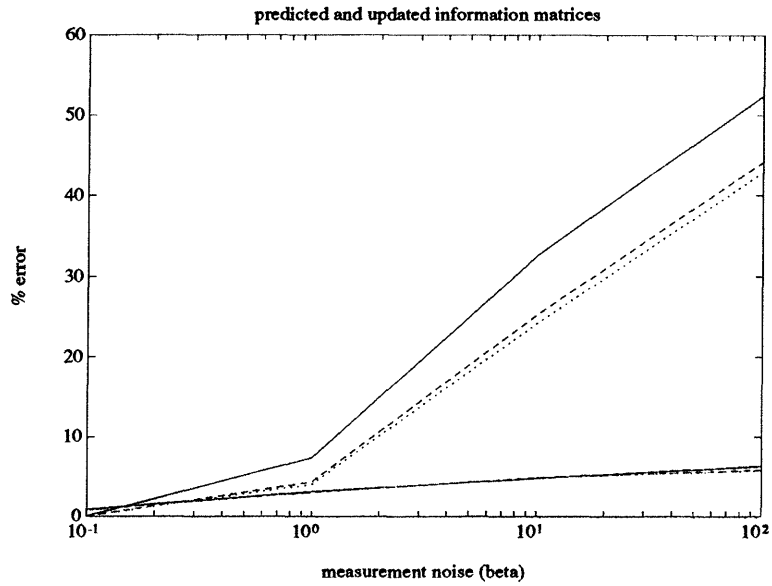


Figure 5-16: The effect of the measurement noise on the approximation errors for the predicted and updated information matrices for different numbers of iterations per nulling session — 1 (solid line), 2 (dashed line), 3 (dotted line).

As in the case for the approximated information filter, the approximate iterative SRIF is sensitive to the level of noise in the measurement. The measurement noise variance should be sufficiently low for a reliable performance. The effect of the process noise, on the other hand, has been observed to be small.

5.5 Simulations: Surface Interpolation

In Sections 5.2.3, 5.3.5, and 5.4.4 numerical examples are presented to illustrate how closely the computational techniques described in this chapter can approximate an optimal set of information matrices. In this section we examine how closely these techniques, particularly the series approximated information filter of Section 5.3.4 and the approximate iterative SRIF of Section 5.4.3, can *estimate* artificially generated scalar fields, $\mathbf{f}(t)$. We add white Gaussian random noise to $\mathbf{f}(t)$ to simulate noisy observations, $\mathbf{g}(t)$, which enter these suboptimal filters as the inputs. We measure

the performance of the suboptimal Kalman filters by *% estimation error*

$$\frac{\left\| \mathcal{E}(\,\widehat{\mathbf{f}}(t)\,) - \mathbf{f}(t) \right\|}{\|\mathbf{f}(t)\|} \times 100, \tag{5.43}$$

where $\widehat{\mathbf{f}}(t)$ is the estimate generated by the filters. Each of the suboptimal filters performs estimation on the same sample path of the observation process $\mathbf{g}(t)$. The estimates based on several such samples are averaged to obtain an average estimate, $\mathcal{E}(\,\widehat{\mathbf{f}}(t)\,)$, for each filter.

Our primary concern in this section is to examine how closely the suboptimal filters can approximate the optimal estimates by comparing the estimation errors (5.43) associated with the suboptimal and optimal filters.

5.5.1 Static surface

The dynamic system model (5.11) (5.12) introduced in Section 5.2.3 (and used also in Sections 5.3.5 and 5.4.4) is used to reconstruct the 10×10 static scalar field (surface) depicted in Figure 5-17. Note that the model (5.11) (5.12) is by no means perfectly matched to the estimation problem for this particular visual field. Specifically, the process noise in (5.11) should not be used for estimation of a static quantity, and the membrane model contained in (5.12) is not as suitable as the thin-plate model for surface reconstruction (as elaborated below). This estimation problem, however, allows us to quantify *relative* performance among the optimal and suboptimal filters in terms of the actual estimates and complements the results in Sections 5.2.3, 5.3.5, and 5.4.4. The surface depicted in Figure 5-17 is corrupted by an additive space-time white Gaussian noise to achieve a signal-to-noise ratio (SNR) of about 2.27. The parameters used for the dynamic system model is $\alpha = 10^{-2}$, corresponding to the variance of the process noise, and $\beta = 10^{-1}$, corresponding to the variance of the measurement noise.

Figure 5-18 depicts the estimate at $t = 10$ obtained by the optimal Kalman filter. It shows expected flattening of the central knoll in the field caused by the over-smoothing caused by the use of the membrane model. Nevertheless, the estimation

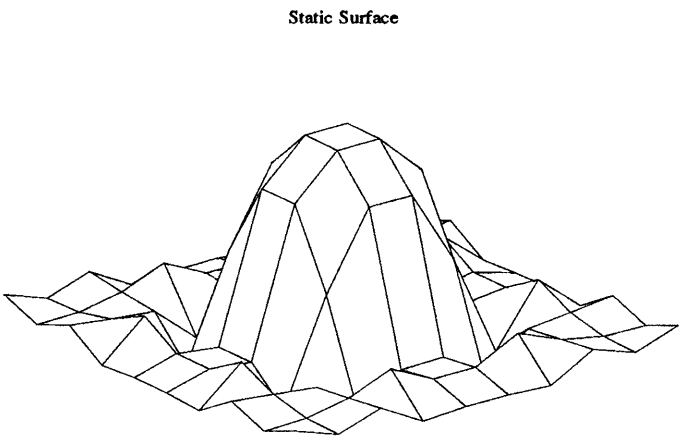


Figure 5-17: The surface to be reconstructed.

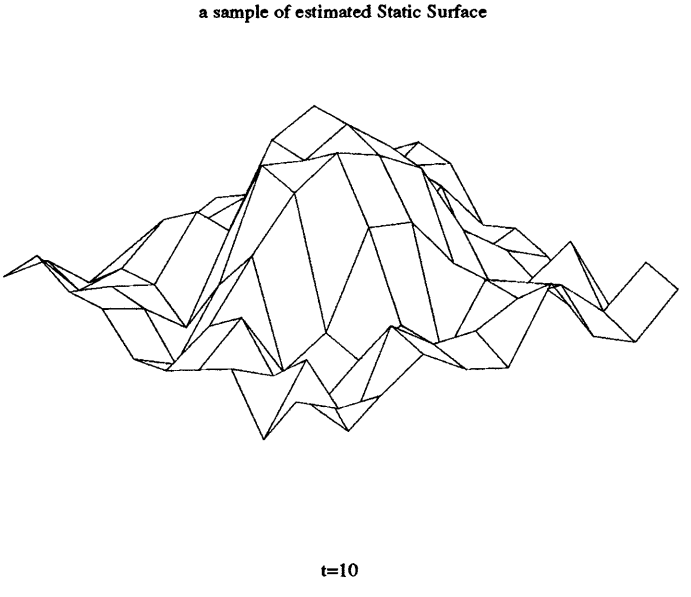


Figure 5-18: A reconstructed surface after 10 frames of measurements have been incorporated using an optical Kalman filter.

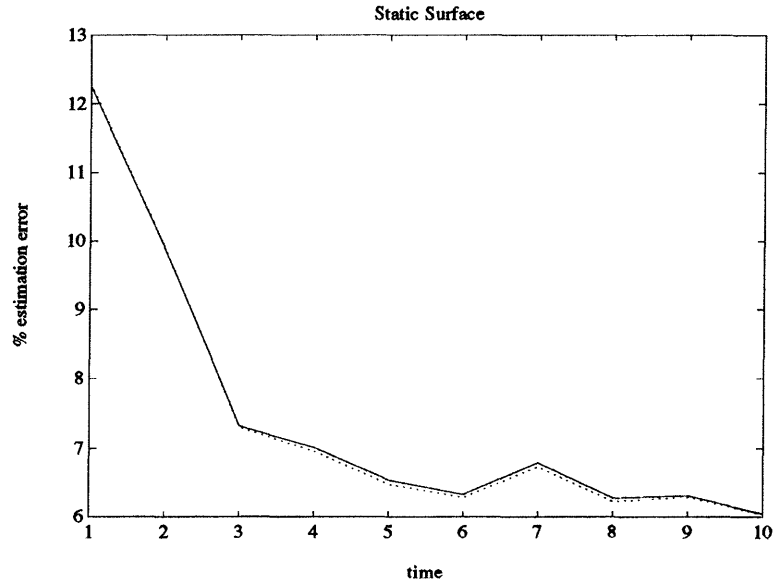


Figure 5-19: Performance of various implementations of Kalman filter measured by "% estimation errors". The solid line corresponds to both an optimal Kalman filter and series approximated information filter, as the estimation errors for these two implementations are close enough to be indistinguishable on this figure. The dotted line corresponds to an approximate iterative SRIF.

errors associated with the optimal Kalman filter as well as suboptimal filters have decreased from about 12% at $t = 1$ to about 6% at $t = 10$, as presented in Figure 5-19 which shows three nearly identical error curves. The three estimation error curves, almost indistinguishable from each other, belong to the optimal Kalman filter, series-approximated information filter, and approximate iterative SRIF, and they illustrate closeness of these particular suboptimal estimates to the optimal estimates. Ten (10) samples of estimates are averaged to obtain the curves. For the series-approximated information filter, only 2 terms in the series (cf. (5.36)) are used in conjunction with an $\mathcal{W}_1$-structural constraint for the information matrices. Increasing the number of terms in the series and relaxing the structural constraint for the information matrices have not been observed to alter the corresponding estimation errors noticably. For the approximate iterative SRIF, on the other hand, a $\mathcal{W}_4$-structural constraint is imposed on the SRI matrices and in each nulling iteration an incomplete sweep corresponding to the same structural constraint is used. Three iterations of such an incomplete sweep are used in each nulling session. Changing the structural constriant

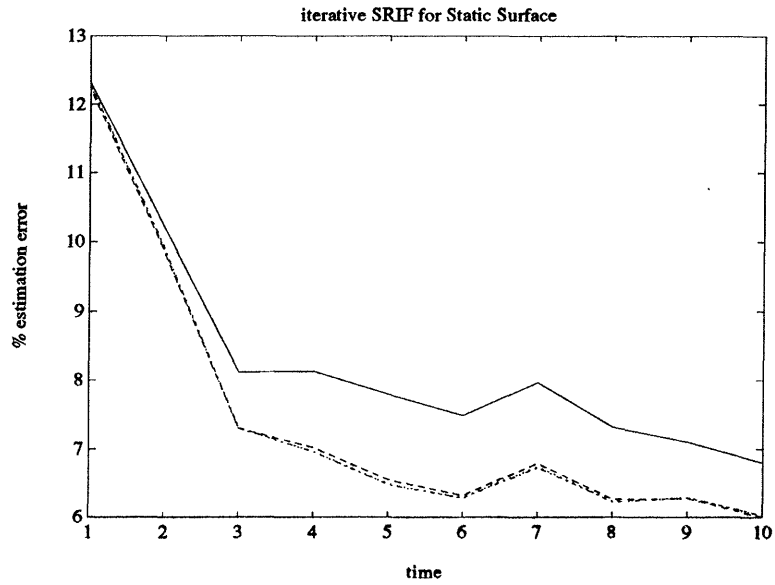


Figure 5-20: Performance of various approximate iterative SRIF's. The solid line corresponds to $\mathcal{W}_1$-truncated, the dashed line to $\mathcal{W}_2$-truncated, the dotted line to $\mathcal{W}_3$-truncated, and the dash-dot line to $\mathcal{W}_4$-truncated. The last two of these lines overlap each other and are indistinguishable from each other.

for the SRI matrices has been observed to alter performance only slightly. Figure 5-20 shows estimation error curves for SRIF's with the structural constraints $\mathcal{W}_1$, $\mathcal{W}_2$, $\mathcal{W}_3$, and $\mathcal{W}_4$. The filter performance depicted in Figure 5-20 improves as the structural constraint is relaxed and the extent of sweep is widened; however, the improvement saturates quickly as the error curves for $\mathcal{W}_3$ and $\mathcal{W}_4$ are almost identical.

5.5.2 Moving surface

A sequence of 16×16 images of a moving tip of quadratic cone has been synthesized, and the moving surface has been reconstructed based on noisy observation of the image sequence using an optimal Kalman filter, series approximated information filter, and approximate iterative SRIF. The surface $\mathbf{f}(t)$ translates across the image frame with a constant velocity whose components along the two frame axes are both 0.2 pixels/frame. That is,

$$f(s_1, s_2, t) = f(s_1 + 0.2, s_2 + 0.2, t - 1).$$

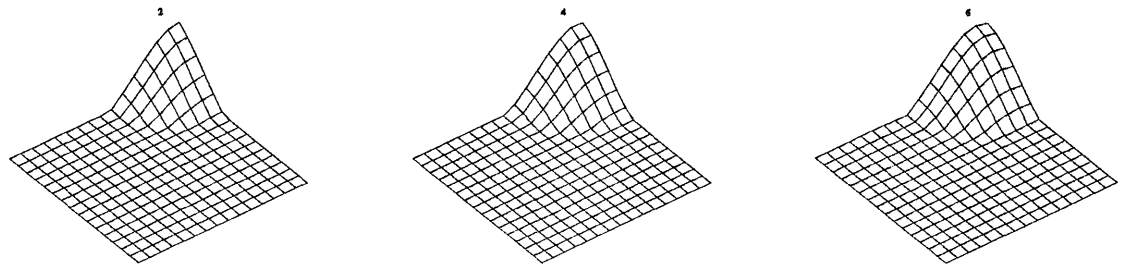


Figure 5-21: The moving surface to be reconstructed at $t = 2, 4$, and 6.

Figure 5-21 shows $\mathbf{f}(t)$ at $t = 2, 4$, and 6. Since the spatial coordinates s_1 and s_2 take only integer values in the discrete dynamic model on which the filters are based on, we use the following approximate model

$$\begin{aligned} f(s_1, s_2, t) &= (1-0.2)^2 f(s_1, s_2, t-1) \\ &\quad + (0.2)(1-0.2) f(s_1+1, s_2, t-1) \\ &\quad + (0.2)(1-0.2) f(s_1, s_2+1, t-1) \\ &\quad + (0.2)^2 f(s_1+1, s_2+1, t-1), \end{aligned}$$

which we express as a matrix dynamic equation

$$\mathbf{f}(t) = \mathbf{A}\mathbf{f}(t-1).$$

In essence, the matrix $\mathbf{A}$ approximately performs spatial shifting of the elements of $\mathbf{f}(t-1)$ by a subpixel (in this case 0.2) amount (see Heel's work such as [27] for more details).

A zero-mean white Gaussian process has been added to $\mathbf{f}(t)$ to simulate a noisy measurement $\mathbf{g}(t)$ with SNR of about 2. Moreover, we have created a situation where at each t only randomly chosen 50% of the points on the surface can be observed. That is, the measurement model is

$$\mathbf{g}(t) = \mathbf{H}(t)\mathbf{f}(t) + \mathbf{r}_0(t) \tag{5.44}$$

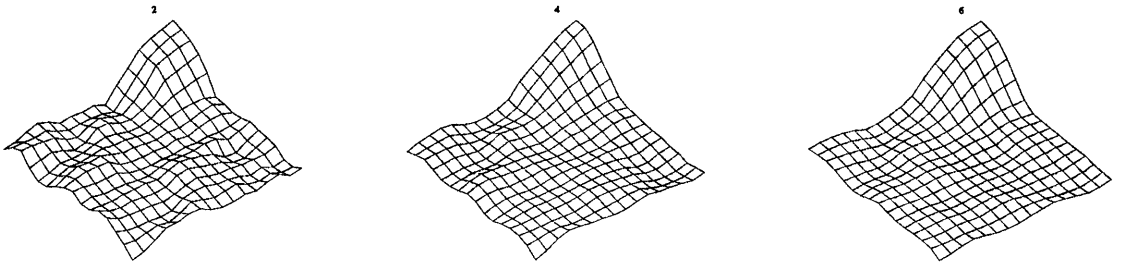


Figure 5-22: Reconstructed moving surface by optimal Kalman filter at $t = 2, 4$, and 6.

where randomly selected 50% (on the average) of the components of $\mathbf{H}(t)$ are 1 and the rest of the components are 0. This type of partial observation is common in surface interpolation using depth data obtained from stereo matching [21, 22], as matching can be performed only on selected features in the images.

The dynamic system model on which the filters are based on is

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{S}^{(1,0)} \\ \mathbf{S}^{(0,1)} \end{bmatrix} \mathbf{f}(t) = \begin{bmatrix} \mathbf{I} \\ \mathbf{S}^{(1,0)} \\ \mathbf{S}^{(0,1)} \end{bmatrix} \mathbf{A}\mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim (\, 0, \alpha\mathbf{I}\,) \tag{5.45}$$

$$\begin{bmatrix} \mathbf{g}(t) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{H}(t) \\ \mathbf{S}^{(2,0)} \\ \mathbf{S}^{(0,2)} \\ 2\mathbf{S}^{(1,1)} \end{bmatrix} \mathbf{f}(t) + \mathbf{r}(t), \; \mathbf{r}(t) \sim \left(0, \begin{bmatrix} \beta\mathbf{I} & & & \\ & \mathbf{I} & & \\ & & \mathbf{I} & \\ & & & \mathbf{I} \end{bmatrix} \right). \tag{5.46}$$

An extended Gauss-Markov form of this descriptor dynamic system can be obtained as presented in Section 4.3.3. Note that a thin-plate model is used as the spatial coherence constraint. Thin-plate models are considered suitable for surface interpolation [21]. The dynamic equation reflects the temporal coherence constraint that penalizes large deviation from the dynamic model $\mathbf{f}(t) = \mathbf{A}\mathbf{f}(t-1)$ *and* imposes smoothness on the deviation $\mathbf{f}(t) - \mathbf{A}\mathbf{f}(t-1)$ using a membrane model. The application of the membrane model makes the process noise spatially smooth, which is reasonable from the perspective that the noise reflects (at least partially) the effect of surface motion which should exhibit some spatial coherence. As in the static surface case, we let

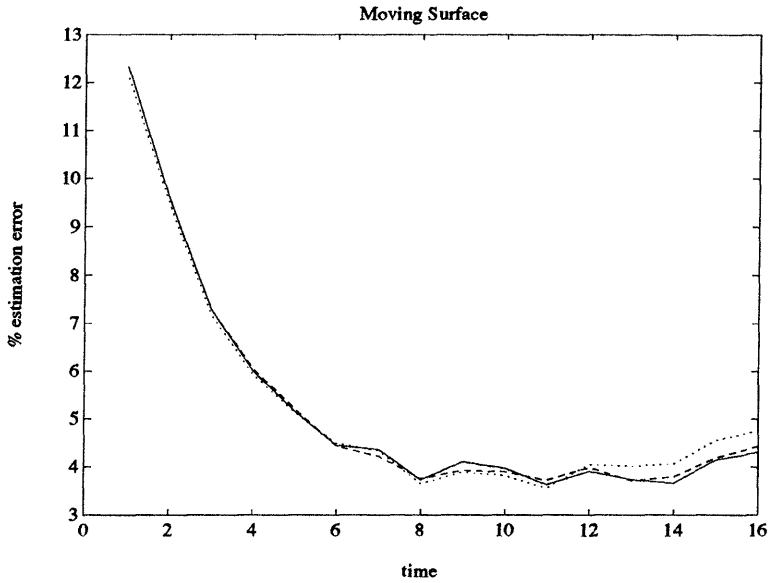


Figure 5-23: Performance of various implementation of Kalman filter measured by "% estimation errors". The errors associated with the optimal Kalman filter (solid line), series-approximated information filter (dashed line), and approximate iterative SRIF (dotted line) are shown.

$\alpha = 10^{-2}$ and $\beta = 10^{-1}$.

Figure 5-22 shows the surfaces reconstructed by the optimal Kalman filter based on the dynamic system above. Observe that the qualitative appearence of the estimated surface improves as more frames of data are incorporated into the estimate. The earlier estimates are expected to be especially noisy because as indicated above the surface is only partially observable in each image frame.

Figure 5-23 shows the estimation errors for the optimal Kalman filter (solid line), series-approximated information filter (dashed line), and approximate iterative SRIF (dotted line) for the first 16 frames. Four samples are averaged to obtain each curve in the figure. The error curves indicate that the two suboptimal filters perform just as well as the optimal Kalman filter. The estimation error associated with each filter decreases steadily from about 12% at $t = 1$ to about 4% at $t = 8$. In the series-approximated information filter, the information matrix is constrained to be $\mathcal{W}_6$-structured, and the first 8 terms are used to approximate the infinite series (5.36) in the prediction step. For the approximate iterative SRIF, on the other hand, a $\mathcal{W}_7$

matrix is used to truncate the SRI matrix and to define the extent of sweep (i.e., $\ell_1 = \ell_2 = 7$). The number of iteration per nulling session is set to be 3. For the two approximate filters, looser structural constraints than the previous examples are appropriate because of the larger spatial extent of the thin-plate model (as opposed to the membrane model used as the spatial coherence constraint in the previous examples) and non-zero off-diagonal elements in the system matrix $\mathbf{A}$.

5.5.3 Summary of simulations

In the two surface reconstruction simulations, the approximate filters have performed almost identically to the corresponding optimal Kalman filter. The discrepancy between the optimal filter and an approximated filter appears smaller when the computed estimates are used as the criterion, i.e., (5.43), as in the simulations just presented than when the information matrices are used, i.e., (5.13), as in the examples in Sections 5.2.3, 5.3.5, and 5.4.4. In a sense this is desirable, because to be able to produce an *estimate* of the same quality as those from the optimal filter is the primary concern in the design of approximate filters.

5.6 Extension to Higher Dimensions

The presentation so far in this chapter is concerned with filter approximation techniques for estimation of scalar fields defined over 2-D spatial domains. The techniques are extendable for estimation of other kinds of fields.

5.6.1 Fields defined over higher dimensional spaces

For fields defined over spaces with dimension other than 2, we need to consider how the neighborhood structures can be defined in such dimensions. This is straightforward. In 2-D, a $\mathcal{W}$-structured matrix has a *nested* centrally banded structure, i.e., it is a block banded matrix whose blocks are themselves banded. For a neighborhood structured matrix in a higher dimensional space, then, the degree of nesting is just

that much higher.

In 1-D space, on the other hand, there is no nesting, i.e., the 1-D counterpart of the $\mathcal{W}$-structure is simply a centrally banded structure such as tri-diagonal, penta-diagonal, etc.

5.6.2 Vector fields

Each element of a vector field, $f(\mathbf{s},t)$, has more than one component. For example, in optical flow $f(\mathbf{s},t)$ represents the two components of a 2-D motion vector. For simplicity, let us consider the case of a 2-D vector field. To extend the approximation techniques in this chapter to estimation of such fields, we need to reconsider what we have been treating as scalars as vectors. Specifically, each "element" in the $N \times N$ information and SRI matrices is now a 2×2 submatrix. For the most part, such treatment of the matrix "elements" is all that is required to extend the approximation techniques. For example, in the polynomial approximation of the matrix inverse (5.36), the "diagonal matrix" Δ is now more exactly a block diagonal matrix whose blocks are 2×2. Also, the recursive matrix inversion technique (Sec. 5.3.2) is not affected by this extension as it already works on the basis of matrix blocks, e.g., if $f(\mathbf{s},t)$ is a 2-vector the blocks in (5.17) simply double in dimension.

A block Givens rotation

As a single exception, however, the extension discussed above is a little more complicated for the Givens rotation used in the iterative SRIF algorithm: How do you null a 2×2 "element" against another? Specifically, let

$$\mathbf{p} \equiv \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix} \quad \mathbf{q} \equiv \begin{bmatrix} q_1 & q_2 \\ q_3 & q_4 \end{bmatrix}$$

and consider nulling $\mathbf{q}$ against $\mathbf{p}$ in a matrix

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} \equiv \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \\ q_1 & q_2 \\ q_3 & q_4 \end{bmatrix}$$

with a unitary operation. This can be accomplished by a two-step process: First use $\mathbf{p}$ to null the first row $[q_1\ q_2]$ of $\mathbf{q}$; then, use $\mathbf{p}$ again to null the second row $[q_3\ q_4]$. Each of these steps can be performed by essentially a sequece of two Givens rotations. Specifically, let

$$\begin{bmatrix} p_1' & p_2' \\ p_3' & p_4' \\ q_1' & q_2' \end{bmatrix} = \begin{bmatrix} c_1 & & s_1 \\ & 1 & \\ -s_1 & & c_1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & c_2 & s_2 \\ & -s_2 & c_2 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \\ q_1 & q_2 \end{bmatrix}$$

where $c_k \equiv \cos\theta_k$ and $s_k \equiv \sin\theta_k$ for $k = 1, 2$. The desired sequence of two rotations θ_1, θ_2 can be obtained from the two equations $q_1' = 0$ and $q_2' = 0$. In practice, it tends to be easier to solve for c_k's and s_k's directly, aided by the trigonometric identity $c_k^2 + s_k^2 = 1$. (In fact, efficient implementation of a Givens rotation is of a major importance as rotations tend to be used repeatedly. References on this topic include [19, 18].)

5.7 Conclusion

Guided by the locality principle, we have developed efficient approximate implementation techniques for the information Kalman filter and square root information filter for multi-frame visual reconstruction problems. The computational cost to propagate the information or SRI pair has been reduced to $O(N)$ flops or less per frame. Thus, the computational complexity for processing a frame in the approximate multi-frame reconstruction filters is on the same order as that for a single-frame reconstruction problem. Through various numerical experimentations, we have shown that the ap-

proximate filters perform just as well as the corresponding optimal filter, as long as the measurement noise is sufficiently low.

Chapter 6

Multi-frame Optical Flow Estimation

In single-frame optical flow computation, spatial coherence constraints are vital not only for mathematical well-posedness of the problem formulation but also for visual acceptability of the results. In this chapter, we investigate the role of an analogous constraint imposed over time for a multi-frame version of the flow computation problem. Specifically, we consider applying a *first order* temporal coherence constraint on time-varying optical flow fields.

6.1 Temporal Dynamics of Optical Flow

Constructing plausible temporal dynamic models for the optical flow field is a key step in our algorithm development. In general, dynamics of the optical flow and other visual fields cannot be captured explicitly without some knowledge of the physical properties of the objects being imaged. For sequential estimation of the depth field, for example, Heel [26, 27] proposes a physically explicit dynamic model based on a geometrical relationship (due to Longuet-Higgins and Prazdny [48]) among the motion parameters of a rigid object, the depth field associated with the object surface, and the optical flow vectors. Conceivably, the same (or a similar) relationship can be used to derive a temporal dynamic model for the optical flow field. This approach, however, leads to a non-linear estimation problem involving simultaneous computation of the optical flow and depth fields as well as the motion parameters.

Alternatively, temporal behavior of the optical flow field can be described in terms

of the *temporal coherence* principle which imposes an inertia condition on motion, essentially allowing the optical flow vectors to change only gradually in time. Temporal coherence models of optical flow are applicable to a wide range of motions in natural scenes, as most movements display inertia of some type. It has also been suggested that temporal coherence might play an important role in human motion vision [23].

In this chapter, we consider multi-frame estimation of optical flow based on a discrete formulation of the following space-time reconstruction problem:

$$\min_f \int_0^\tau \int_{\mathcal{D}} \nu \|g - hf\|^2 + \mu_1 \left\| \frac{\partial}{\partial s_1} f \right\|^2 + \mu_2 \left\| \frac{\partial}{\partial s_2} f \right\|^2 + \rho \left\| \frac{\partial}{\partial t} f \right\|^2 ds\, dt \qquad (6.1)$$

which is essentially a simple temporal extension of the optical flow formulation (2.11) by Horn and Schunck [34] using a first order temporal coherence constraint.

6.1.1 Eulerian v.s. Lagrangian perspective

We have mentioned in Chapter 4 that designing temporal dynamic models for visual fields requires dealing with the data-estimate correspondence problem which arises due to the motion of the imaged surfaces. Here, we briefly discuss how the correspondence problem is handled in our first-order temporal modeling for the optical flow.

Let $q(\mathbf{s}, t)$ be a zero-mean space-time white noise with covariance ρ^{-1}. Then (6.1) implies the temporal dynamic model

$$\frac{\partial}{\partial t} f(\mathbf{s}, t) = q(\mathbf{s}, t), \qquad (6.2)$$

which imposes temporal coherence on the flow vector associated with each *pixel location.* Alternatively, consider the dynamic model using a first-order total derivative instead of the partial derivative,

$$\frac{d}{dt} f(\mathbf{s}, t) = q(\mathbf{s}, t), \qquad (6.3)$$

which implies that the motion of each *surface element* changes gradually over time.

That is, the temporal coherence model (6.2) is viewer-based, i.e., Eulerian, while (6.3) is object-based, i.e., Lagrangian.

The Lagrangian model (6.3) certainly makes more physical sense as it achieves data-estimate correspondence based on the motion of the imaged object; however, there are reasons that the Eulerian model (6.2) might be at least as appropriate as (6.3):

1. The total derivative on the left hand side of (6.3) can be expanded in terms of the partial derivatives as

$$\frac{d}{dt}f = \frac{\partial}{\partial t}f + \left[\frac{\partial}{\partial \mathbf{s}}f\right] f. \tag{6.4}$$

   A spatial coherence constraint will keep the spatial variation in the estimated flow field small, and only flow vectors with relatively small magnitudes can be computed accurately based on the brightness gradients (Section 2.4). Thus, the term $\left[\frac{\partial}{\partial \mathbf{s}}f\right] f$ tends to have a small magnitude in practice, so that approximating (6.3) with (6.2) is reasonable.

2. There are some motion patterns, such as static planar rotation, in which the flow vectors are invariant with respect to the pixel locations but not with respect to the surface elements. Thus, for such motion patterns (6.2) is more appropriate than (6.3).

The Eulerian model (6.2) is simple to implement. Specifically, as described in Chapter 4, (6.2) leads to a discrete temporal dynamic model for the optical flow $\mathbf{f}(t)$:

$$\mathbf{f}(t) = \mathbf{f}(t-1) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(0, \rho^{-1}\mathbf{I}\right), \tag{6.5}$$

where the noise $\mathbf{q}(t)$ is uncorrelated over time. This Gauss-Markov dynamic model indicates that the optical flow evolves in time as the accumulation of a random perturbation at each time frame. Implementation of the Lagrangian model (6.3), on the other hand, requires maintaining a correspondence between the image frame coordinates and moving surface elements, e.g., using the estimated flow vectors to track

the images of the surface elements in the frame [10]. That is, using (6.4) as an intermediate step, (6.3) can be discretized as

$$f(\mathbf{s},t) - f(\mathbf{s},t-1) + \left[\frac{\partial}{\partial \mathbf{s}} f\right] f(\mathbf{s},t-1) = q(\mathbf{s},t)$$

leading to a temporal dynamic model for $\mathbf{f}(t)$:

$$\mathbf{f}(t) = \left(\mathbf{I} - \left[\frac{\partial}{\partial \mathbf{s}}\mathbf{f}\right]\right) \mathbf{f}(t-1) + \mathbf{q}(t), \tag{6.6}$$

where $\left[\frac{\partial}{\partial \mathbf{s}}\mathbf{f}\right]$ is a block diagonal matrix whose 2×2 blocks are $\left[\frac{\partial}{\partial \mathbf{s}} f\right]$. The matrix $\left(\mathbf{I} - \left[\frac{\partial}{\partial \mathbf{s}}\mathbf{f}\right]\right)$ is a specific form of the matrix $\mathbf{A}(t)$ in (4.2), introduced to perform the inter-frame correspondence operation described in Chapter 4. Usually, such a correspondence matrix is computed "on-line" adaptively [26, 27]. In our discrete Lagrangian model (6.6), the spatial derivative $\left[\frac{\partial}{\partial \mathbf{s}}\mathbf{f}\right]$ must be *estimated* by, for example, differentiating the estimated flow $\widehat{\mathbf{f}}(t-1)$. Noisy estimates of the optical flow can, therefore, make the implementation of the Lagrangian model (6.6) unstable.

In summary, although the Eulerian model essentially ignores the data-estimate correspondence issue, it is a good approximation of the more sophisticated Lagrangian model. This is because the brightness change constraint equation requires a sufficiently high temporal sampling rate with respect to the magnitudes of the optical flow vectors (cf. Sec. 2.4) so that the inter-frame motion is not large enough for the correspondence issue to have significant effects on estimation. Our main objective in this chapter is to compare optical flow estimates *with* and *without* temporal coherence, and the simplicity of the Eulerian model (6.5) makes it attractive for us to use for this purpose. Certainly, a more physically oriented model such as (6.6) can be more appropriate for specific motion estimation problems; however, modeling at such specific details is beyond the scope of this thesis.

6.1.2 The Simple Temporal Coherence (STC) Model

As described above, the temporal coherence constraint in (6.1) leads to the discrete Eulerian temporal dynamic model (6.5) for the optical flow $\mathbf{f}(t)$. We refer to this discrete dynamic model as the *simple temporal coherence (STC) model* or the *STC constraint.* The complete formulation of the multi-frame optical flow estimation that corresponds to a discrete version of (6.1) is given by the dynamic system consisting of the STC dynamic equation (6.5) and the measurment equation

$$\begin{bmatrix} \mathbf{g}(t) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{H}(t) \\ \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \mathbf{f}(t) + \mathbf{r}(t), \quad \mathbf{r}(t) \sim \left(0, \begin{bmatrix} \mathbf{N}^{-1}(t) & & \\ & \mu_1^{-1}\mathbf{I} & \\ & & \mu_2^{-1}\mathbf{I} \end{bmatrix} \right) \tag{6.7}$$

which has been described in Section 3.2.4. This last equation is a discrete version of the single-frame optical flow formulation by Horn and Schunck [34]. Indeed, note that (6.1) is exactly the Horn and Schunck formulation when $\rho = 0$.

6.2 Computing Optical Flow with STC

In this section we demonstrate the beneficial effects of the simple temporal coherence (STC) constraint applied to optical flow computations. Synthetic image sequences of moving brightness patterns are processed by a STC-based optical flow estimation method as well as by other more traditional flow computation methods, and the improvements gained by using the STC constraint are presented. In this section, small image frames are used so that the optical flow estimates can be computed exactly by direct matrix inversion instead of the usual iterative computation, in order to facilitate comparison of the *optimal* estimates by different methods. The following three methods are compared:

- *The* "single-frame" *method.*

  Each frame of optical flow is computed independently, i.e., without any provision for temporal integration of data. A discrete version of Horn and Schunck's

single-frame optical flow computation algorithm [34] is used. This approach corresponds to solving the maximum likelihood (ML) estimation problem based on the observation equation (6.7) at each t. In this chapter, unless otherwise indicated, we give equal weights to the brightness constraints at all pixels, so that $\mathbf{N}(t) = \mathbf{I}$. The ML estimate for the optical flow, $\widehat{\mathbf{f}}(t)$, is obtained by solving the inversion problem

$$\left(\mathbf{H}^T(t)\mathbf{N}(t)\mathbf{H}(t) + \mu_1\mathbf{S}_1^T\mathbf{S}_1 + \mu_2\mathbf{S}_2^T\mathbf{S}_2\right)\widehat{\mathbf{f}}(t) = \mathbf{H}^T(t)\mathbf{g}(t). \tag{6.8}$$

Typically, Gauss-Seidel iterations are used to solve this equation. In the experiments presented in this chapter, however, the optimal estimates are obtained non-iteratively by matrix inversion unless otherwise indicated. (We have chosen to work with small image frames so that this is practically possible.)

- *The* "once-GS" *method.*

  In [34], Horn and Schunck suggest an approach to multi-frame optical flow estimation not based on a statistical optimality criterion. Let us refer to this algorithm as the "once-GS" method. The method performs only one Gauss-Seidel iteration for the inverse problem (6.8) at each t but uses the estimate from the previous frame, $\widehat{\mathbf{f}}(t-1)$, to initialize the iteration. Unlike the "single-frame" method, therefore, this method *does* have some provision for propagating the estimates temporally. Note, however, that if the iterations are allowed to converge for each frame of data, the resulting flow estimates would exactly be the same as the "single-frame" estimates. Although the once-GS method is *ad hoc* in terms of its temporal integration of data, its ease in implementation is appealing from a practical point of view. It is an interesting method to compare against the STC-based method.

  Because the "once-GS" method iterates only once per frame of data, it must process many frames before a reasonable estimate starts to emerge. We do not want such an initial transient effect to be a factor in our experimental study; therefore, the first frame of data is processed non-iteratively by direct matrix

inversion, i.e., using the "single-frame" method above. (Thus, the method is receiving a much more favorable initial estimate in this experiment than in practice.)

- *The* "STC" *method.*

  The *simple temporal coherence* constraint is imposed on the flow vectors as well as the standard spatial coherence constraint. Optical flow under such constraints is computed sequentially using the information Kalman filter based on the dynamic system (6.5) (6.7):

  – prediction step

$$\overline{\mathbf{L}}(t) = \rho\mathbf{I} - \rho^2 \left(\hat{\mathbf{L}}(t-1) + \rho\mathbf{I}\right)^{-1} \tag{6.9}$$

$$\overline{\mathbf{f}}(t) = \hat{\mathbf{f}}(t-1) \tag{6.10}$$

$$\overline{\mathbf{z}}(t) = \overline{\mathbf{L}}(t)\overline{\mathbf{f}}(t) \tag{6.11}$$

  – update step

$$\hat{\mathbf{L}}(t) = \overline{\mathbf{L}}(t) + \mathbf{H}^T(t)\mathbf{N}(t)\mathbf{H}(t) + \mu_1\mathbf{S}_1^T\mathbf{S}_1 + \mu_2\mathbf{S}_2^T\mathbf{S}_2 \tag{6.12}$$

$$\hat{\mathbf{z}}(t) = \overline{\mathbf{z}}(t) + \mathbf{H}^T(t)\mathbf{N}(t)\mathbf{g}(t) \tag{6.13}$$

$$\hat{\mathbf{f}}(t) = \hat{\mathbf{L}}^{-1}(t)\hat{\mathbf{z}}(t) \tag{6.14}$$

  where $\overline{\mathbf{L}}(t)$ and $\hat{\mathbf{L}}(t)$ denote the predicted and updated information matrices, respectively. The small image frame size allows us to perform the exact computation necessary for optimal filtering. Suboptimal filtering techniques are, however, necessary for a larger, more typical image frame size. Examples using suboptimal Kalman filters are presented in later sections.

Note that all three methods presented above process the initial frame of data (spatial and temporal gradients of the brightness) identically; they all compute the initial estimates from (6.8) by direct matrix inversion.

6.2.1 Measurement integration by STC

One of the advantages of using a temporal coherence constraint in optical flow estimation is improvement in estimates due to temporal accumulation of information regarding the flow vectors. Because of the aperture problem, the measurements of the brightness gradients by themselves are not sufficient to deduce the optical flow vectors uniquely. That is, the brightness measurement at each pixel at a certain time can only provide us with partial information about the optical flow vector at that pixel. The spatial coherence ("smoothness") constraint, typically used as the extra constraint to the brightness constraint in single-frame flow computation, effectively supplements the gradient measurements at each pixel with the measurements at surrounding pixels, constraining the problem well enough so that a unique optimal flow vector can be computed at that pixel. In multi-frame flow computation, such integration of measurements can be expanded over time as well as space.

Reconstruction of optical flow using only spatial data integration (i.e., by using only spatial coherence constraints as in the case with the "single-frame" method described above) cannot be performed correctly when a complete set of the information necessary to estimate the flow vectors is not contained in *each* frame of images. Specifically, since variation in the orientations of the measured spatial gradients is necessary to resolve the aperture problem, optical flow computation method employing only a spatial coherence constraint will have difficulties dealing with images such that all spatial gradients are oriented in almost the same direction (including the case where the number of spatial gradient vectors with significant magnitudes are very small). Addition of a temporal coherence constraint can often relieve such difficulties. We give a demonstration below that the STC constraint is, in fact, instrumental in correctly estimating the flow.

Experiment: Rotating Ramp

1. *The motion.*

   A brightness pattern is rotated constantly over time. The amount of rotation between adjacent image frames is 0.1 radians (slightly less than 6°).

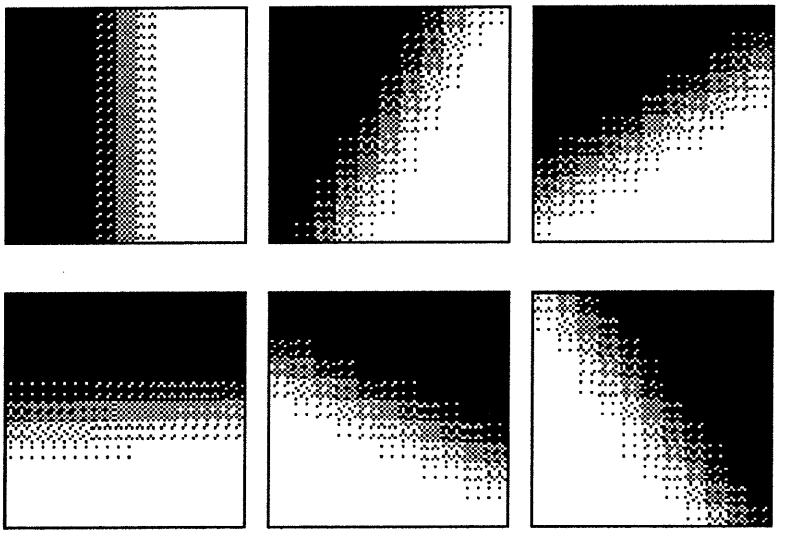


Figure 6-1: Rotating Ramp. Frames 0, 5, 10, 15, 20, and 25 are shown.

2. *The image sequence.*

   The brightness pattern to be rotated is a 10×10 images of a sloping edge. Figure 6-1 shows frames 0, 5, 10, 15, 20, and 25 of the image sequence. The sloping edge, or ramp, is a quarter-wave of a sinusoid changing from -1 to 1 over a band approximately 5-pixel wide. The pixel values are measured with floating-point accuracy without noise. The ramp is the only region in the image with non-zero spatial gradients; the rest of the image frame is feature-less (constant brightness) so that motion is undetectable there. Since the ramp forms a straight edge, all the spatial gradient vectors in each image frame are oriented in the identical direction.

3. *The flow estimates.*

   Figure 6-2 shows the estimated flow vectors using the three methods mentioned above: "single-frame", "once-GS", and "STC". As described before, all three methods begin with the same initial estimates, reflected in the results for frame 0 in the figure. The STC method produces a fairly accurate estimate at frame 25. The estimate by the once-GS method at frame 25 appears to be fairly good, also. The single-frame method, however, fails completely. This shows that some sort of temporal integration of measurements is necessary for correct estimation.

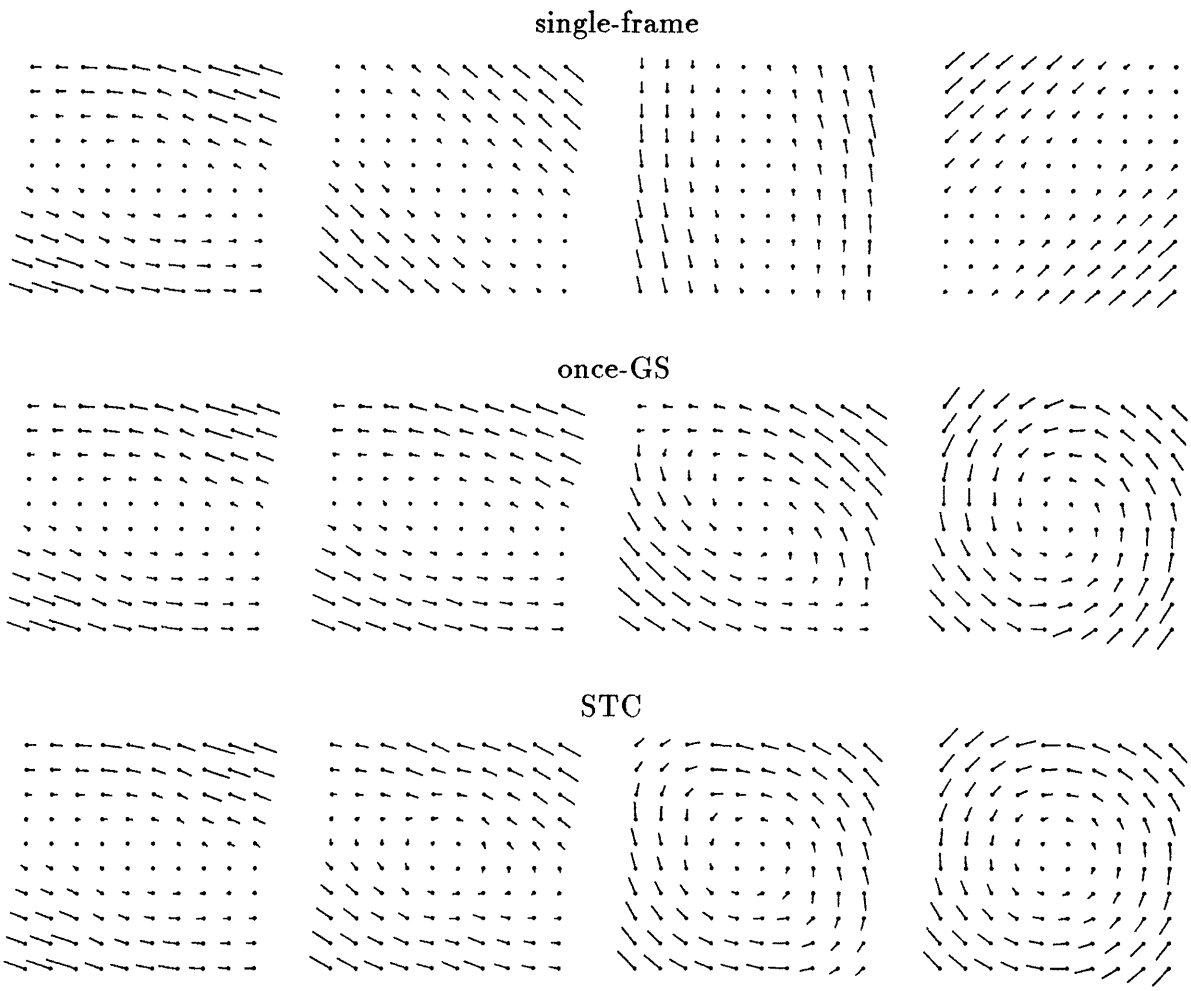


Figure 6-2: Optical flow estimates for the rotation ramp example. The flow patterns at frames (from left to right) 0, 5, 15, and 25 are shown. The flow vectors are magnified by 1.5 for clarity.

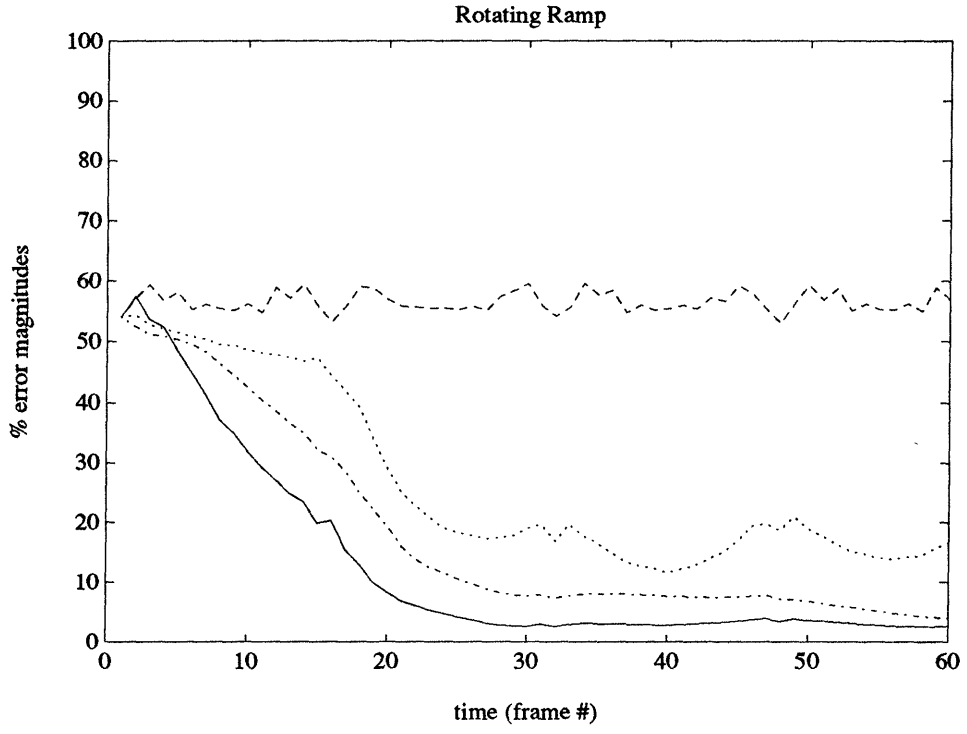


Figure 6-3: The normalized average estimation errors in the rotating ramp example for the three methods: single-frame(dashed line), once-GS(dotted line), and STC(solid line). The dash-dot line shows the errors for the hybrid of the once-GS and STC methods.

(The computation parameters were $\rho = 1$ and $\mu_1 = \mu_2 = 0.00025$.)

4. *The estimation errors.*

Let us define the "percent normalized average estimation error" at time (frame) t as

$$\frac{\sum_{\mathbf{s}} \left\| \widehat{f}(\mathbf{s},t) - f(\mathbf{s},t) \right\|}{\sum_{\mathbf{s}} \left\| f(\mathbf{s},t) \right\|} \times 100, \tag{6.15}$$

where f is the true flow and $\widehat{f}$ is the estimated flow. This quantity is computed for the three methods, and the results are plotted on Figure 6-3. The dashed, dotted, and solid curves correspond to the errors associated with the single-frame, once-GS, and STC methods, respectively. The percent errors for the once-GS method are at least 10% higher than those for the STC method.

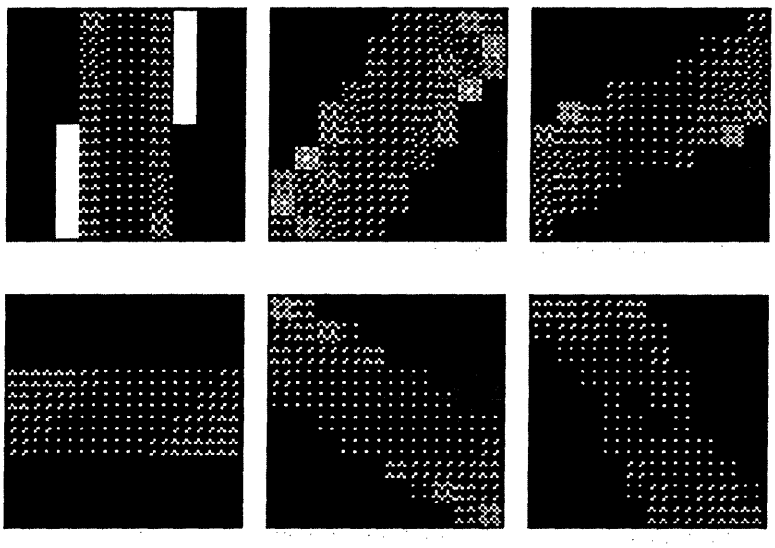


Figure 6-4: Kalman gain magnitudes for STC in the rotating ramp case. Frames 0, 5, 10, 15, 20, and 25 are shown.

5. *The Kalman gains and variances for the STC method.*

   The STC method is implemented as a Kalman filter. One can visualize the temporal integration process of the STC method by observing the "images" of the magnitudes of the Kalman gains and variances. Figure 6-4 shows the magnitudes of the Kalman gains at frames 0, 5, 10, 15, 20, and 25. Lighter pixels have higher values than darker pixels. (Note that the frame size is only 10×10, resulting in jagged appearence of the images.) The magnitude of Kalman gain is an indication of how much the filter values the new data in updating the estimate. By comparing Figures 6-4 and 6-1 one can observe that the Kalman gain is high where the image contrast is high. Figure 6-5 shows the magnitudes of the error variances. Pixels with low (dark) variances have high confidence in their associated flow vector estimates. Notice that the area of high confidence grows with time, indicating that the filter produces good estimates of flow vectors over a wider region in the image frame as more measurements are integrated over time.

6. *A "once-GS" implementation of the STC Kalman filter.*

   The once-GS method performs only an *ad hoc* propagation of the flow estimate between frames. It is difficult to determine how much the previous estimate

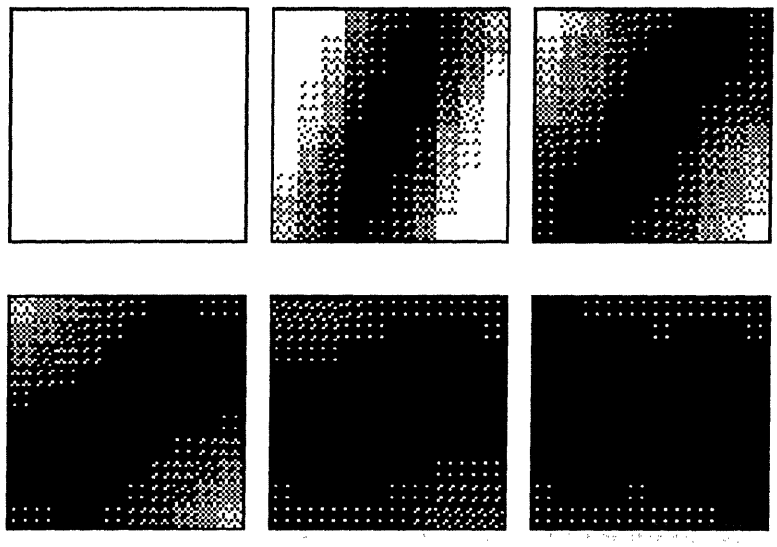


Figure 6-5: Error variance magnitudes for STC in the rotating ramp case. Frames 0, 5, 10, 15, 20, and 25 are shown.

(thus, the previous measurements) influences the present estimate, as the propagated estimate from the previous frame appears only as the initial condition for the Gauss-Seidel iteration. Nevertheless, the method is attractive for its computational efficiency. Let us consider a hybrid method which retains the computational feature of the once-GS method while properly imposing a temporal coherence constraint as in the STC method. The resulting algorithm is essentially a STC-based Kalman filter where the update step is performed by only a single Gauss-Seidel iteration (initialized by the predicted estimate). The average normalized error magnitude for such an algorithm is plotted as the dash-dot line in Figure 6-3. The improvement over the original once-GS method (dotted line) is evident, as the error decreases almost monotonically down to below 5% (at frame 60).

6.2.2 Noise reduction by STC

A single pair of frames in image sequences of many natural scenes often contains enough information (i.e., variation in spatial gradients of the brightness) to reconstruct the motion field. Even in these image sequences, however, a temporal coherence constraint can improve the quality of optical flow estimates by reducting the

effect of measurement noise. Measurement of gradients is a noise-sensitive process. As described in Section 2.4, presmoothing, i.e., to spatially blur the images before obtaining the brightness gradients, can often be important for maintaining validity of the brightness constraint and hence reducing the sensitivity of the optical flow estimates to noise in the measurements. The effect of noise can be decreased further by applying a temporal coherence constraint on the flow. The constraint serves to average the noisy data in time. Let us examine how the STC constraint performs for such a purpose.

Experiment: Collision

1. *The motion.*

   A viewer approaches a wall at a constant velocity until collision. We define the time of collision to be $t = 0$. Because of the effect of perspective projection, the viewer, looking at some fixed optical features on the wall, perceives motion. The optical flow that the viewer sees has a radially expanding pattern, in which the flow vector magnitudes increase with time. Appendix D mathematically details the dynamics of the time-varying optical flow vectors $f(\mathbf{s}, t)$. Figure 6-6 shows the spatial average of $\|f(\mathbf{s}, t)\|$ and the corresponding standard deviation at each frame t. Note that in the beginning (e.g., $t \leq -50$), the flow is almost static in time, making application of the STC constraint suitable in this region. Close to the collision, however, the flow vectors have great variations in time and space, so that the temporal and spatial sampling rates necessary for the brightness change constraint equation to be valid (cf. Sec. 2.4) might become too high for this particular image sequence to satisfy. In fact, as we will see, it is not possible to estimate the flow vectors with reasonable quality for $t \geq -10$ from the image sequence.

2. *The image sequence.*

   There are 101 images in the sequence. The frame size is 10×10. The brightness pattern on the wall is a sinusoidal checkerboard pattern, and in the initial image

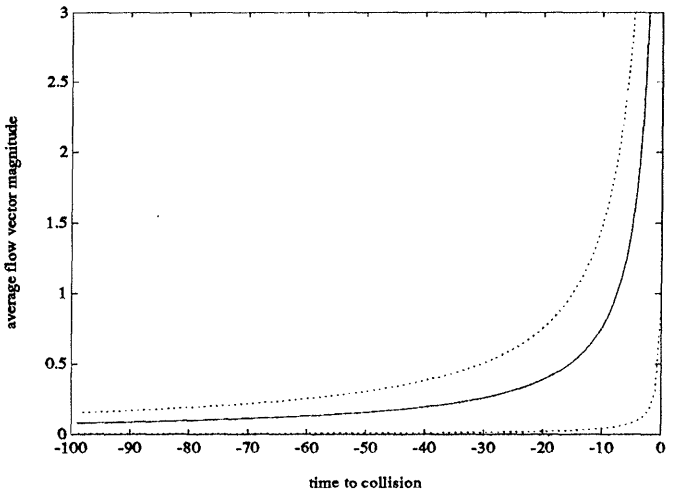


Figure 6-6: Collision motion pattern: the spatial average $a(t)$ (solid line) of the flow vector magnitudes for each image frame t and $a(t) + \sigma(t)$ where $\sigma(t)$ is the corresponding standard deviation (dotted lines) are shown.

frame, at $t = -100$, approximately one period of the sinusoid along each axis can be viewed. The focus of expansion[1] in the optical flow pattern is chosen to be located outside of the image frame, at about 5 pixels below and to the right of the lower right corner of the frame.

3. *The noise*

   The pixel values are floating-point values between -1 and 1. A temporally and spatially white Gaussian noise process with variance 0.01 has been added to each pixel to achieve a signal-to-noise ratio of about 20.

4. *Presmoothing.*

   Each image is convolved with the 3×3 unit uniform stencil (a 3×3 matrix of ones) to smooth the brightness values spatially before the brightness gradients are computed by finite differencing.

5. *The flow estimates.*

   Figure 6-7 shows the true optical flow and the estimated flow at $t = -90$,

[1] For the synthesized motion (pure translation), each frame of optical flow consists of vectors of varying length that all pass through a single point when extended. This point where the optical flow vector is zero is called the *focus of expansion* [33].

-80, -70, and -60 using the three methods: single-frame, once-GS, and STC. Qualitatively, the STC method clearly produces the best flow estimates. (The computation parameters were $\rho = 1$ and $\mu_1 = \mu_2 = 0.025$.)

6. *The estimation errors.*

   The normalized average estimation errors (6.15) for the three methods are plotted on Figure 6-8. The estimation errors for the STC method are significantly lower than those for the single-frame method. Note that the largest decrease in the errors for this method occurs within the first 10 frames. The once-GS method has noise-reduction capability as well, but its response time is longer than for the STC method. The errors of the STC and once-GS methods increase steadily after $t = -50$, because of the gradual increase in spatial and temporal variations in the flow vectors and hence corresponding increase in the required sampling rates. (Note: the sampling rates are fixed in this particular image sequence.)

6.2.3 Summary

The experiments in this section have demonstrated the importance of temporal integration of measurements in optical flow computation and the effectiveness of the STC constraint in performing this task. Specifically, the STC constraint can help resolve the aperture problem and reduce noise sensitivity by assimilating more data (over time) than traditional optical flow computation methods.

6.3 An Evaluation of the STC Modeling

As described before, an Eulerian temporal dynamic model such as the STC model should be nearly as effective in multi-frame optical flow estimation as a Lagrangian model. We demonstrate this in this section. For this purpose, let us compare the performance of the STC-based method in reconstructing the motion vectors in the Collision image sequence in the previous section with the performance of a method

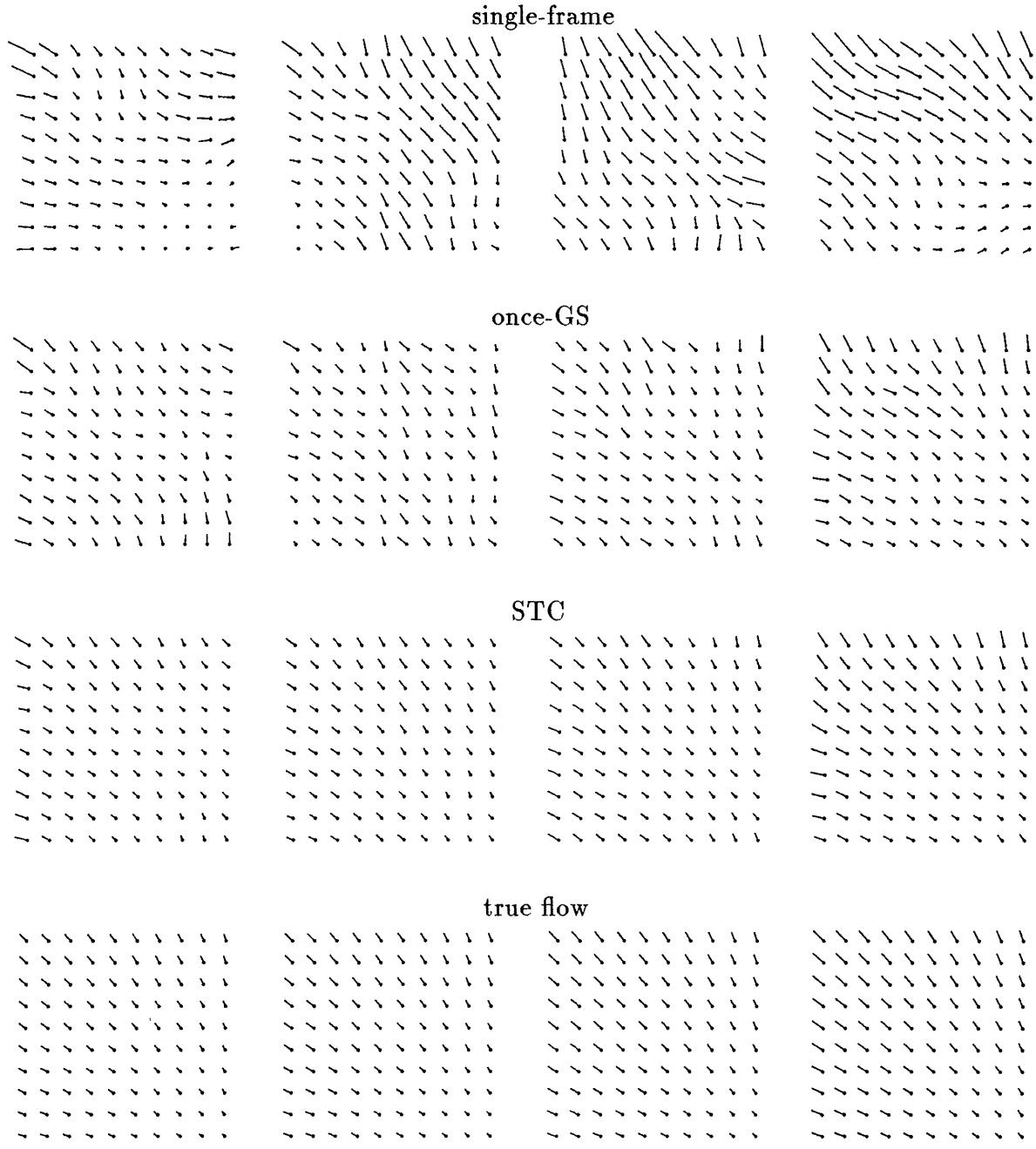


Figure 6-7: Optical flow estimates at times $-90, -80, -70$, and -60. The flows are magnified by 2 for clarity.

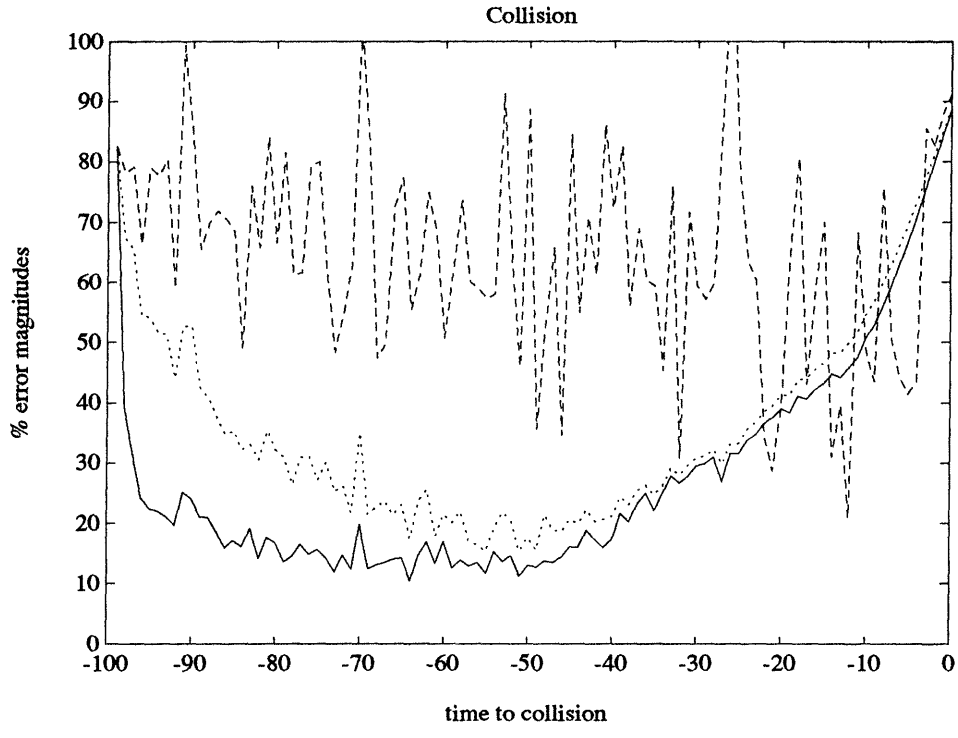


Figure 6-8: The estimation errors in the collision example for the methods: single-frame(dashed-line), once-GS(dotted-line), and STC(solid-line).

based on an accurate dynamic representation of the flow vectors for this particular image sequence. As noted before, the flow vector in the Collision sequence displays a range of temporal dynamics, from near-static to rapidly changing.

As derived in Appendix D, the true flow vectors in the Collision image sequence obey the following Lagrangian temporal dynamics:

$$\frac{d}{dt}\begin{bmatrix} f_1(\mathbf{s},t) \\ f_2(\mathbf{s},t) \end{bmatrix} = 2\begin{bmatrix} \frac{(f_1(\mathbf{s},t))^2}{\Delta s_1} \\ \frac{(f_2(\mathbf{s},t))^2}{\Delta s_2} \end{bmatrix} \tag{6.16}$$

for

$$\begin{bmatrix} f_1(\mathbf{s},t) \\ f_2(\mathbf{s},t) \end{bmatrix} \equiv f(\mathbf{s},t) \quad \text{and} \quad \begin{bmatrix} \Delta s_1 \\ \Delta s_2 \end{bmatrix} \equiv \mathbf{s} - \mathbf{s}_{\text{FOE}} \tag{6.17}$$

where $\mathbf{s}_{\text{FOE}}$ is the location of the focus of expansion. (Recall that the optical flow

displays expanding patterns.) A discrete version of (6.16) is also derived as

$$\begin{bmatrix} f_1(\mathbf{s},t) \\ f_2(\mathbf{s},t) \end{bmatrix} = \begin{bmatrix} 1-\frac{\partial f_1}{\partial s_1} & -\frac{\partial f_1}{\partial s_2} \\ -\frac{\partial f_2}{\partial s_1} & 1-\frac{\partial f_2}{\partial s_2} \end{bmatrix} \begin{bmatrix} f_1(\mathbf{s},t-1) \\ f_2(\mathbf{s},t-1) \end{bmatrix} + 2 \begin{bmatrix} \frac{(f_1(\mathbf{s},t-1))^2}{\Delta s_1} \\ \frac{(f_2(\mathbf{s},t-1))^2}{\Delta s_2} \end{bmatrix} \tag{6.18}$$

where the spatial partial derivatives are evaluated at $(\mathbf{s}, t-1)$. Let us call this the *nonlinear model* of the temporal dynamics of the flow vectors in the Collision sequence. Obviously, the model is accurate only for this particular image sequence. (In practice, we usually cannot predict the temporal dynamics of the flow well enough to construct a model with such accuracy. We synthesize such a model here for the comparison purpose only.)

An extended Kalman filter is implemented (see Appendix D) based on the "nonlinear model" (6.18) to estimate the optical flow using the noisy Collision image sequence. It is assumed that $\mathbf{s}_{\text{FOE}}$ is known. Figure 6-9 shows the percent normalized estimation errors (6.15) for the filter based on the nonlinear model as well as for the filter based on the STC model. As expected, the nonlinear model yields more accurate estimate than the STC model when the flow vectors are changing rapidly, i.e., for $t \geq -35$. When the flow vectors are not changing as rapidly, however, the estimates based on the STC model has slightly smaller errors than those based on the nonlinear model. The right hand side of the nonlinear model (6.18) consists of second order terms with respect to $f(\mathbf{s},t)$ as well as spatial derivatives which must be computed from the estimate of $f(\mathbf{s},t)$; thus, it can be speculated that the prediction step of the extended Kalman filter based on (6.18) amplifies estimation errors.

The flow vectors estimated with both the nonlinear and STC models are shown on Figure 6-10. The estimated flows depicted in the figure are from the time interval during which the flow vectors are changing quite rapidly. In the frames near the collision (i.e., $t \geq -30$) the average errors (in Fig. 6-9) for the nonlinear-estimates are lower than those for the STC-estimates; however, the flow patterns produced by the nonlinear model-based filter are not necessarily better in these frames. For example, the nonlinear-estimate at frame $t = -10$ depicted in Figure 6-10 is qualitatively quite different from the corresponding true flow pattern.

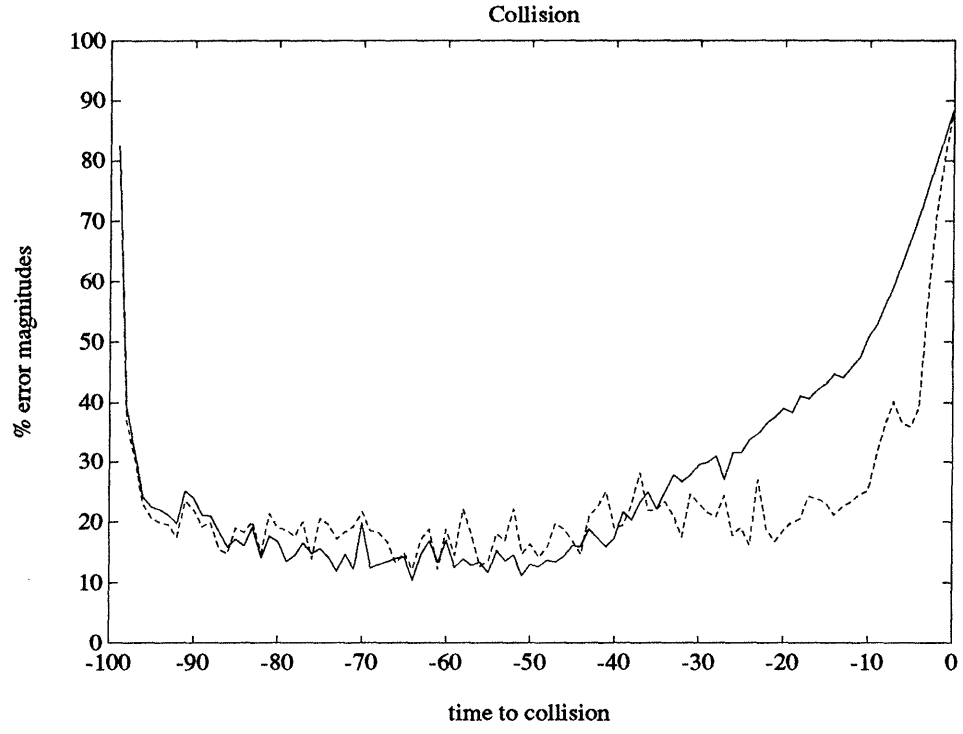


Figure 6-9: Normalized average errors for the collision experiment. The errors for the estimates based on the STC(solid line) and nonlinear(dashed line) models are shown.

To conclude, the experiment in this section has demonstrated that the mathematical simplicity of the STC model can be advantageous in terms of the robustness of filter performance as well as ease in implementation, compared with a more complex model that attempts to capture the flow dynamics in greater details. Specifically, the "non-linear" filter is sensitive to spatial variations in the estimated flow vectors because of the spatial gradient terms in (6.18). The detailed modeling approach has another drawback in that the temporal dynamics of optical flow cannot be predetermined in most applications.

Practically speaking, to track motions with large temporal variations using a STC-based method, one needs to sample the image frequently enough over time. It is in accord with a characteristic of a gradient-based optical flow computation, which also requires high enough temporal sampling rate for accurate measurements of the temporal gradients.

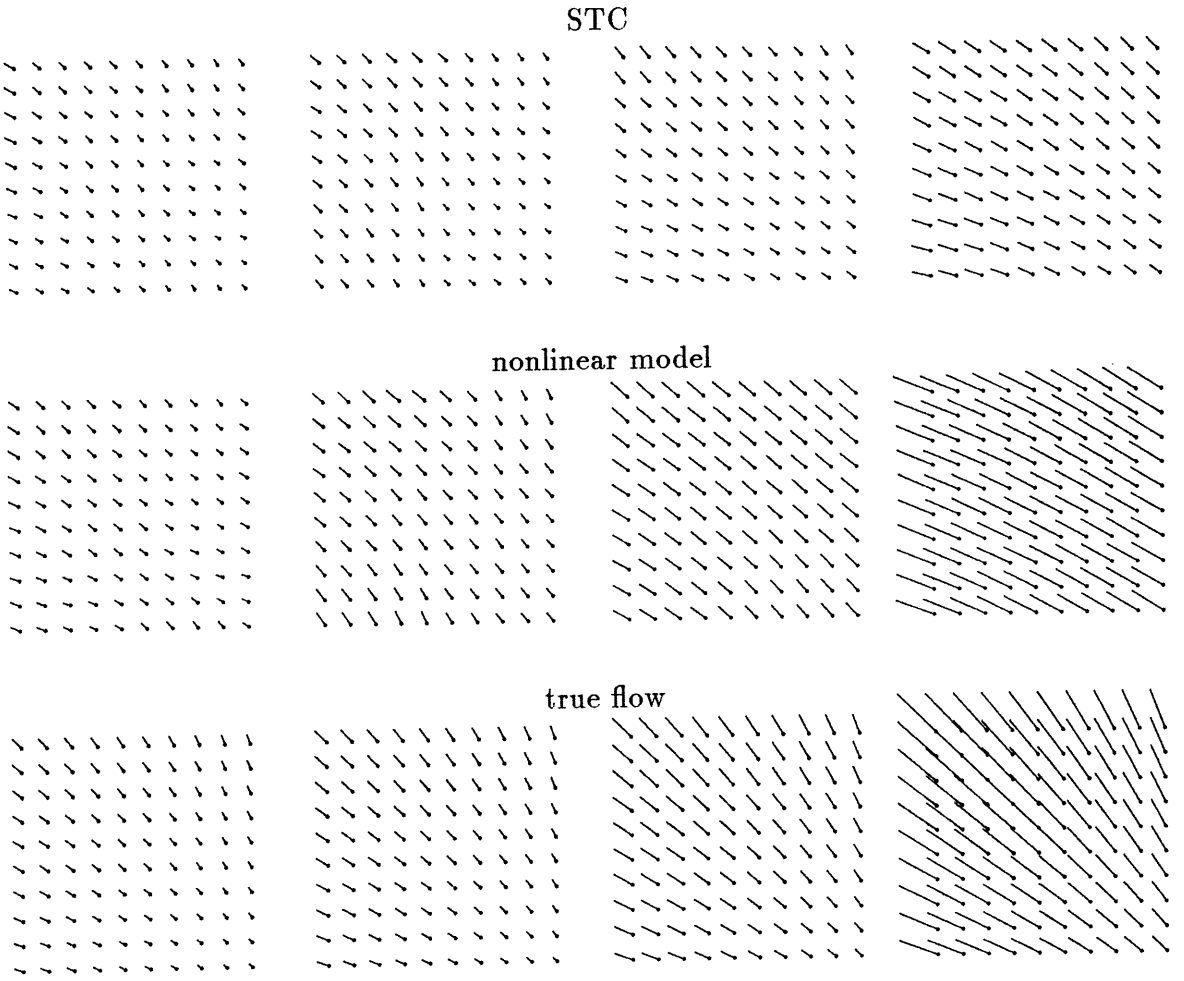


Figure 6-10: Estimated flow for the collision experiment. Frames $t = -40, -30, -20,$ and -10 are shown.

6.4 Suboptimal Filtering based on STC

The number of pixels in a frame of a typical image sequence is on the order of 10^4 to 10^6. Such a large data size makes optimal implementation of the STC-based information Kalman filter (6.9)-(6.14) impractical. In this section, we investigate effectiveness of filter approximation techniques discussed in Chapter 5, by applying the series approximation presented in Section 5.3.4 to the STC-based information filter. Only the first two terms in the infinite series (5.36) are used, i.e., the prediction step (6.9) is replaced by

$$\overline{\mathbf{L}}(t) = \rho\mathbf{I} - \rho^2(\Delta^{-1} - \Delta^{-1}\Omega\Delta^{-1}) \tag{6.19}$$

where Δ is a block diagonal matrix whose diagonal blocks are 2×2 and are identical to the corresponding block diagonal part of the matrix $\widehat{\mathbf{L}}(t-1) + \rho\mathbf{I}$ while Ω is given by $\Omega = \widehat{\mathbf{L}}(t-1) + \rho\mathbf{I} - \Delta$.

Recall from Section 3.3.2 that the matrix $\mathbf{H}^T(t)\mathbf{N}(t)\mathbf{H}(t) + \mu_1\mathbf{S}_1^T\mathbf{S}_1 + \mu_2\mathbf{S}_2^T\mathbf{S}_2$ has a *nearest neighbor* structure. The suboptimal prediction step (6.19) preserves the nearest neighbor structure of the (approximated) information matrix, i.e., if $\widehat{\mathbf{L}}(t-1)$ has a nearest neighbor structure, so does $\overline{\mathbf{L}}(t)$. The nearest neighbor structure is also preserved in the optimal update step (6.12), which means that the suboptimal filter preserves such a matrix structure for all t. It can be verified straightforwardly that propagating the information matrix in the suboptimal filter as in (6.12) and (6.19) costs only $O(N)$ flops per frame and has a local, modular computational structure suitable for parallel implementation.

6.4.1 Experiment: approximation error

The image sequences in the Rotating Ramp (Section 6.2.1) and Collision (Section 6.2.2) experiments are processed with the series-approximated STC-filter. The resulting estimates are compared with the estimates obtained with the corresponding optimal Kalman filter. Figure 6-11 shows the average errors for the optimal and suboptimal

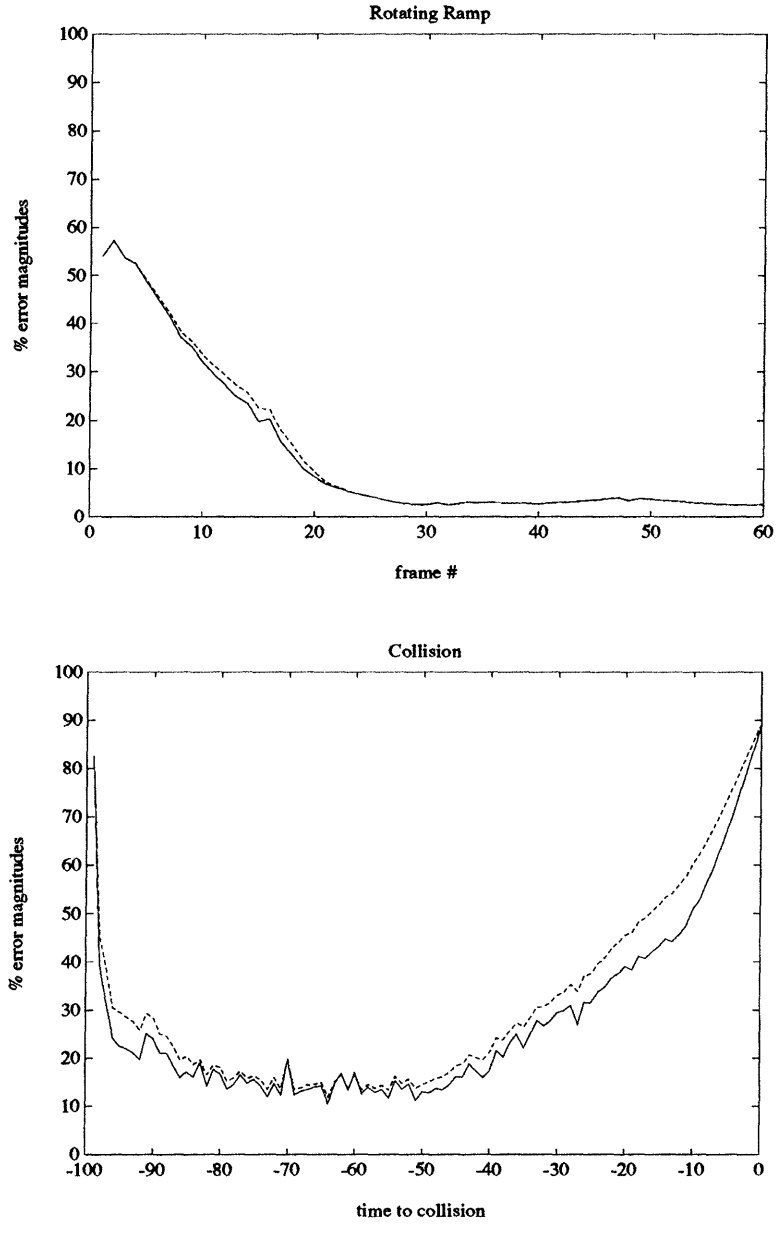


Figure 6-11: The estimation errors for the suboptimal STC-filter (dashed-line) and the optimal filter (solid-line).

estimates; the difference between the suboptimal and optimal estimates appear to be small. To quantify the difference between the two estimates, we have computed for each t a percent normalized average approximation error

$$\frac{\sum_{\mathbf{s}} \left(\left\| \Delta \widehat{f}_s(\mathbf{s},t) \right\| - \left\| \Delta \widehat{f}_o(\mathbf{s},t) \right\| \right)}{\sum_{\mathbf{s}} \left\| f(\mathbf{s},t) \right\|} \tag{6.20}$$

where $\Delta \widehat{f}_s$ and $\Delta \widehat{f}_o$ are the estimation errors for the suboptimal and optimal filters, respectively, and f is the true flow. Figure 6-12 shows the approximation errors associated with the flow estimates for the two image sequences. In the Rotating Ramp case, in which the images are noise-free, the approximation error is at most 3% and is negligible for most t. The errors are often negative, meaning that the supoptimal filter has estimated more accurately than the optimal filter in some frames. For the noisy Collision image sequence, the approximation error ranges between 0% and 10%. Note that large approximation errors occur only in the time range $t \geq -50$ where even the optimal filter does not perform well due to undersampling (cf. Fig. 6-8).

6.4.2 Computing optical flow in a realistic image sequence

The improvements in implementational efficiency (in terms of both computational costs and stroage requirements) achieved by the filter approximations described previously allows us to impose the STC constraint on realistic multi-frame optical flow estimation problems, which usually require processing image sequences with a much larger frame size than those in the experiments presented in this chapter so far.

Experiment: Yosemite

We compare the estimates obtained by using the suboptimal STC-filter and the "single-frame" method of Section 6.2.

1. *The image sequence.*

   A motion image sequence is synthetically generated based on a 3-D topographic

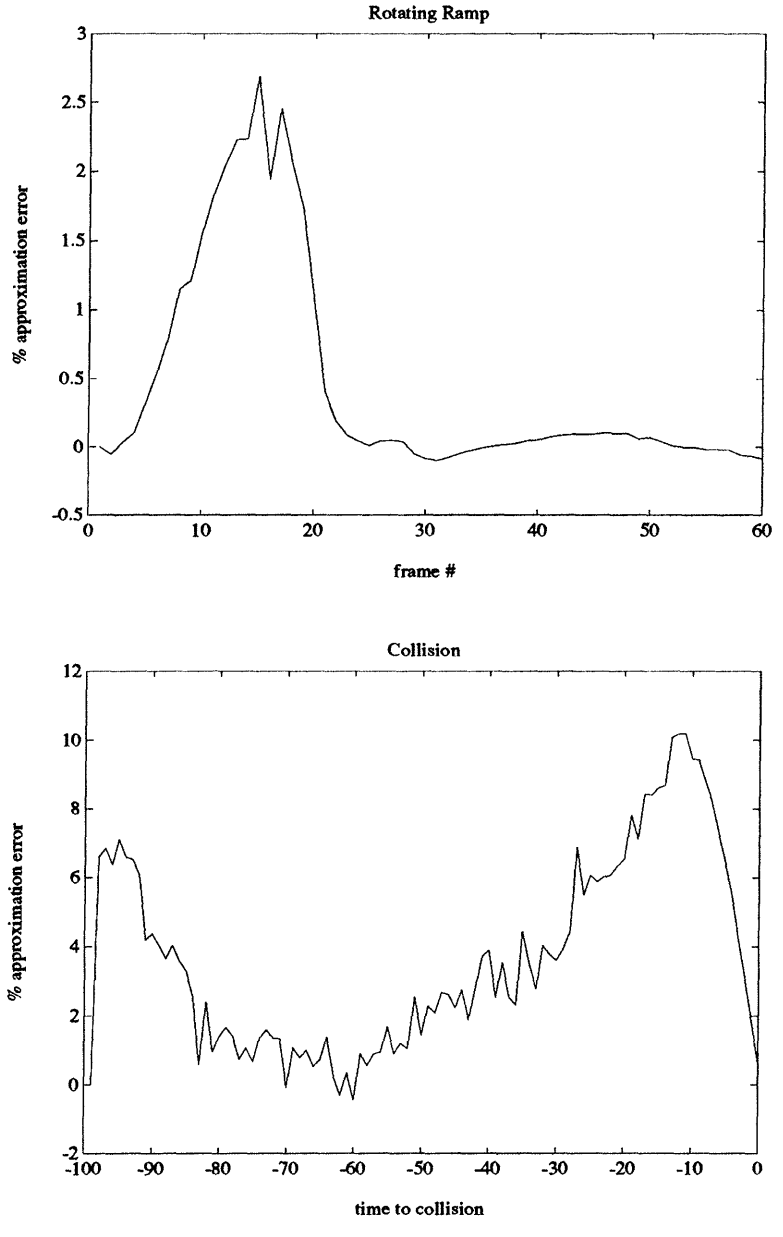


Figure 6-12: The errors introduced by the approximation of the optimal STC-filter (as % of the magnitudes of the true flow).

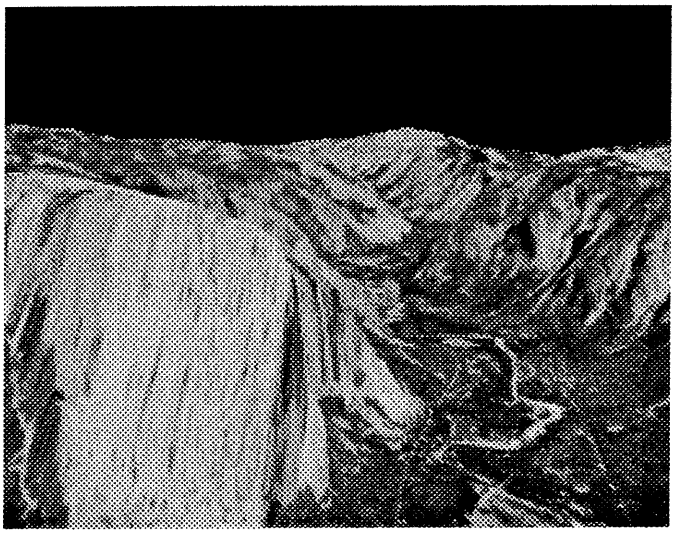


Figure 6-13: A frame from the Yosemite image sequence

model of the Yosemite Valley[2]. Figure 6-13 shows a typical frame from the sequence. The frame size is 316×252, and the pixel values are quantized to 256 grey levels. We have corrupted the images by adding an independent Gaussian noise with a variance 9 to each pixel to achieve a signal-to-noise ratio of about 85 and then requantizing the resulting pixel values to 256 grey levels.

2. *The true optical flow.*

   An advantage of using a synthetic motion image sequence is that the true motion is known. Figure 6-14 shows the tenth frame of the true optical flow sequence.

3. *Presmoothing.*

   The 9×9 unit uniform stencil is used to spatially smooth the images before brightness gradients are computed.

4. *Weighting the gradient measurements.*

   In Section 2.4 we have determined that the second order brightness gradients $\frac{\partial^2 I}{\partial s \partial t}$, $\frac{\partial^2 I}{\partial^2 s}$, and $\frac{\partial^2 I}{\partial^2 t}$ should be small for the brightness change constraint equation to be accurate for a given set of spatial and temporal sampling rates. In this experiment we weight the gradient measurement with a function of $\frac{\partial^2 I}{\partial s \partial t}$ to quan-

[2]Lyn Quam of SRI International has produced the image sequence.

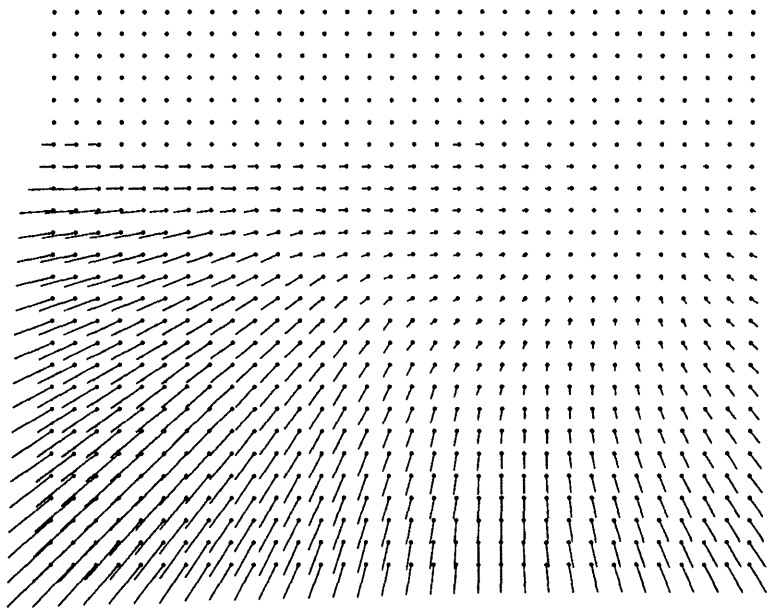


Figure 6-14: The true flow for the tenth frame in the Yosemite image sequence. The flow vectors are reduced by a factor of 2 for display purposes.

tify the emphasis placed on the brightness constraint at each pixel. Specifically, for the observation equation

$$g(\mathbf{s},t) = h(\mathbf{s},t)f(\mathbf{s},t) + e(\mathbf{s},t) \quad e(\mathbf{s},t) \sim \left(0, \nu^{-1}(\mathbf{s},t) \right) \tag{6.21}$$

we set

$$\nu(\mathbf{s},t) = \exp\left(-k \left\| \frac{\partial^2}{\partial \mathbf{s} \partial t} I(\mathbf{s},t) \right\|^2 \right) \tag{6.22}$$

where k is a constant that we have chosen to be $\frac{1}{2}$ for the Yosemite image sequence. Since $\nu(\mathbf{s},t)$ are the elements of the diagonal matrix $\mathbf{N}(t)$, using (6.22) we no longer have $\mathbf{N}(t) = \mathbf{I}$. The value of $\frac{\partial^2}{\partial \mathbf{s} \partial t} I$'s are obtained by forward temporal differencing of the computed first order spatial gradients. Replacing the usual $\nu(\mathbf{s},t) = 1$ with (6.22), we have been able to decrease the average estimation error by over 4% per frame. We have attempted similar weighting schemes of the observating equation (6.21) using other second order gradients but have not found them as beneficial as (6.22).

5. *Iterative solution for the estimates.*

Both the suboptimal STC-filter and single-frame method require solving the inverse problems (6.14) and (6.8), respectively, to obtain the flow estimates at each t. We have used 500 Gauss-Seidel iterations to solve the inverse problem for each frame of flow estimate, except for the estimates in the first frame. The estimate from the previous frame is used to initialize each iterative procedure. To obtain the estimates in the first frame, due to lack of a favorable initialization values, 3500 iterations have been used. Note that the suboptimal STC-filter (but not its optimal version) allows us to use such an iterative method by setting up each update step as an inversion of a nearest neighbor operator.

6. *The flow estimates and estimation errors.*

Figure 6-15 shows the tenth frame of the estimated flow vectors computed by the single-frame method and suboptimal STC-filter. The filter parameters were $\rho = 10$ and $\mu_1 = \mu_2 = 250$. Noise-rejection effect due to the STC constraint can be observed in the upper part of the frame. Figure 6-16 shows the estimation error vectors associated with the estimates in Figure 6-15. Some systematic estimation errors can be observed in the lower left corner of the frame. Possible sources of such errors are:

- The lower left corner of the frame is the region where the magnitudes of the flow vectors are the largest. As discussed in Section 2.4, the brightness change constraint equation is less effective for large flow vectors.
- In this region of the image sequence is a large cliff (*El Capitan*) which displays vertical striations of brightness. In fact, the brightness images in this region lacks contrast in the *horizontal* direction, making the vertical motion components difficult to be estimated. The error vectors in Figure 6-16 confirm this; the flows tend to be underestimated in the vertical direction.

Finally, Figure 6-17 shows the estimation errors in the first ten frames for the two flow computation methods. The suboptimal STC-filter consistently yields

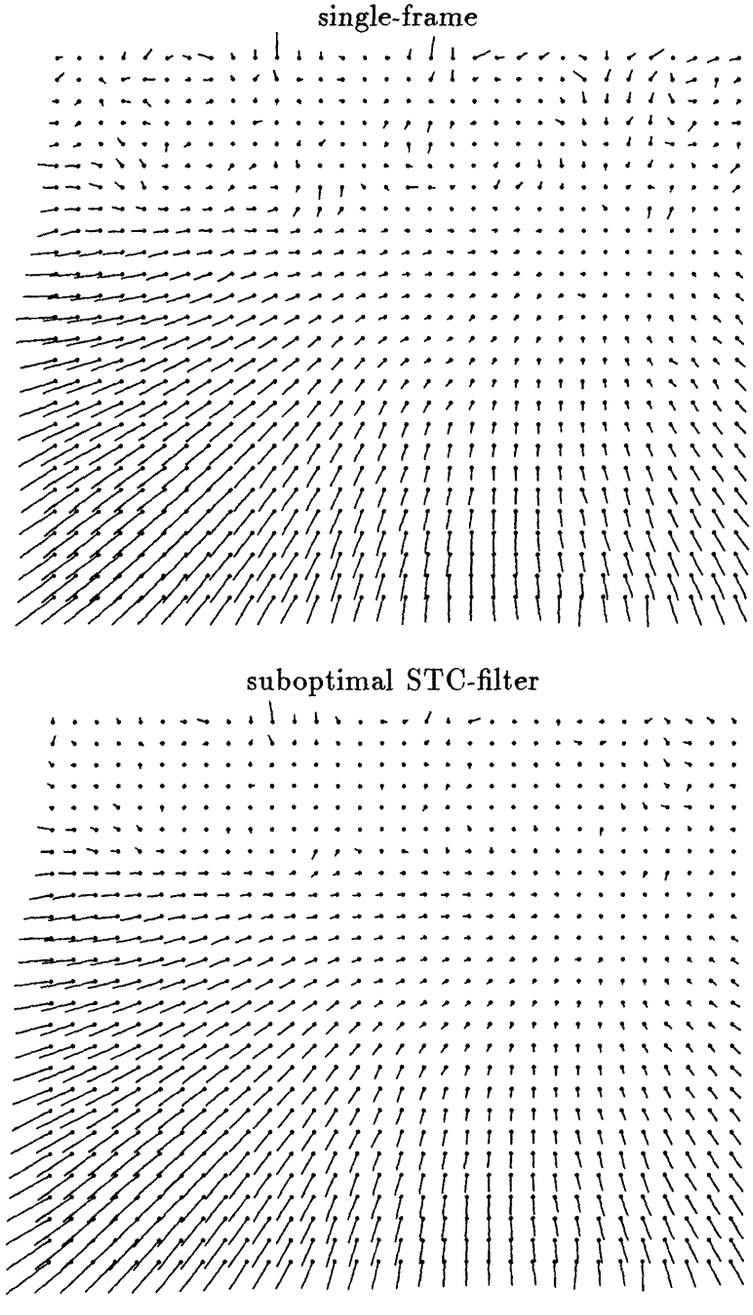


Figure 6-15: The optical flow estimates for the tenth image frame by the single-frame method and the suboptimal STC-filter. The flow vectors are reduced by a factor of 2 for display purposes.

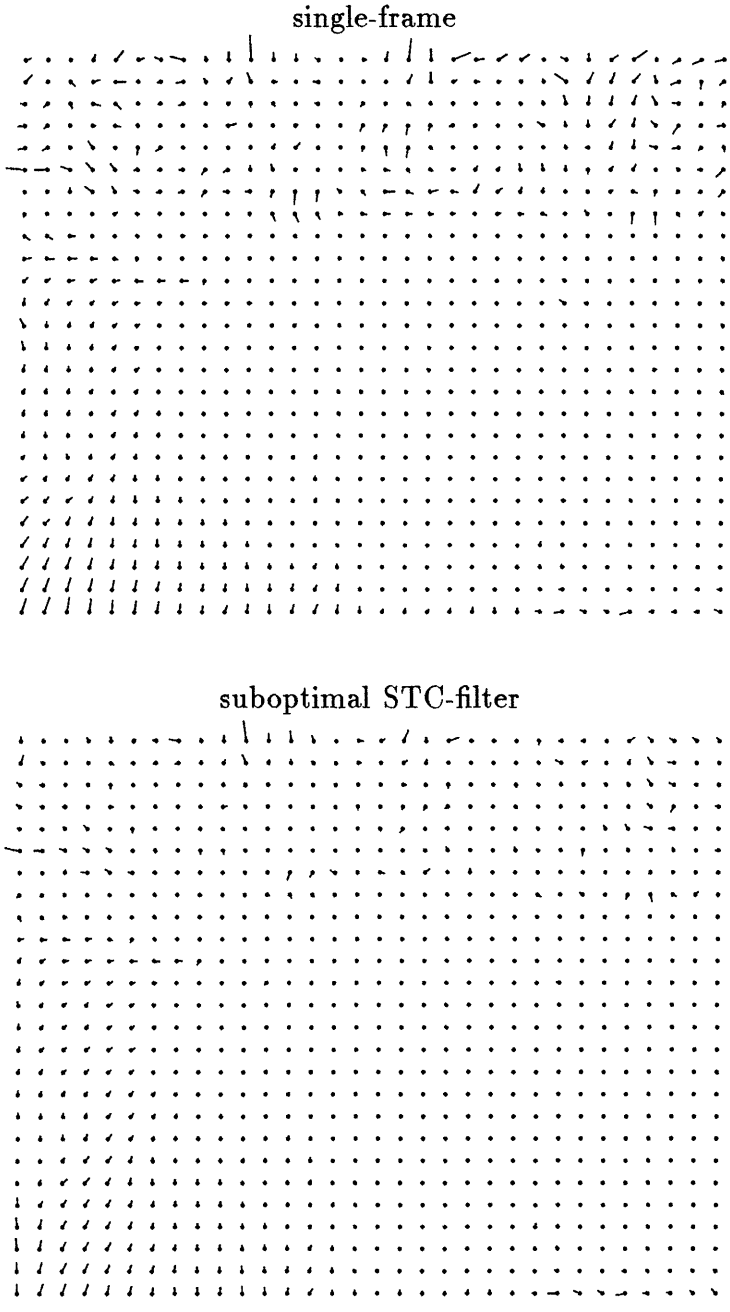


Figure 6-16: The estimation errors corresponding to the estimated flow in Figure 6-15.

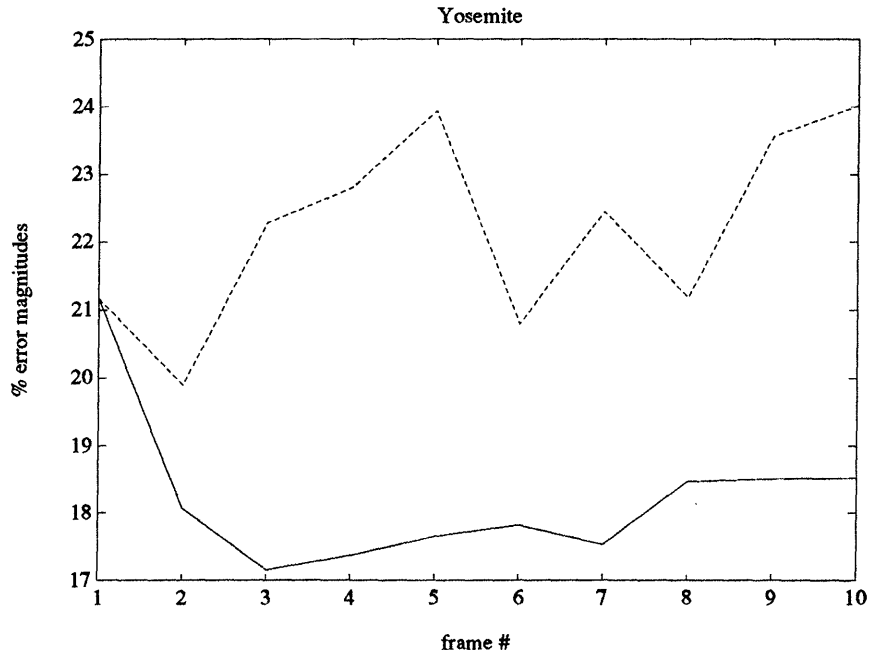


Figure 6-17: The estimation errors by the suboptimal STC-filter(solid-line) and the single-frame method(dotted-line) during processing of the noisy Yosemite image sequence.

more accurate estimates than the single-frame method.

Experiment: Stagnation Flow

The optical flow sequences examined so far in this chapter are all projections of the movements of solid objects. In this experiment we consider estimation of the motion of a non-rigid body, in particular fluid flow, using the suboptimal STC-filter and the "single-frame" method.

1. *The flow.*

   Figure 6-18 shows a flow pattern whose velocity vector at point (x, y) is given by $(Ax, -Ay)$ for $A = 0.1$, where the coordinate origin is at the midpoint of the bottom edge of the figure. This type of flow (for an arbitrary constant A)[3] is useful for a local characterization of *stagnation flow* [63], i.e., the flow of fluid obstructed perpendicularly by a solid object.

[3]As described in [63], this is a planar potential flow given by a *stream function* Axy, i.e., a function which is constant along each streamline.

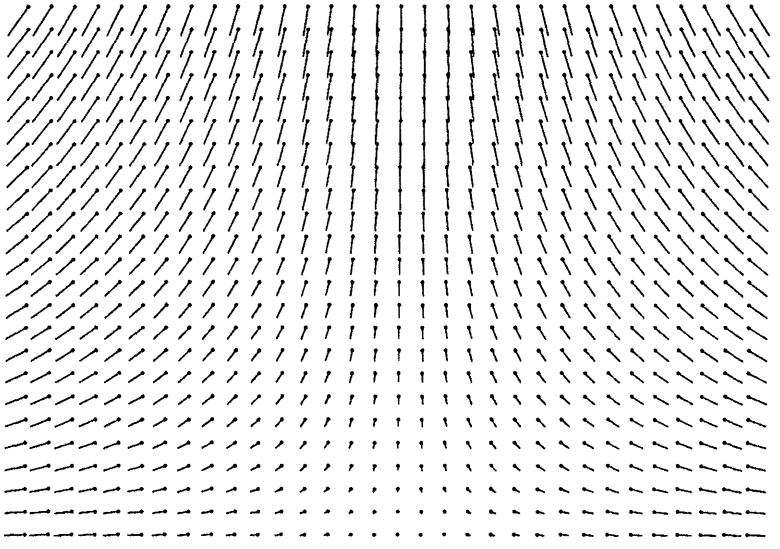


Figure 6-18: The true flow in the Stagnation Flow experiment. Every other flow vector along each axes is shown with a magnification factor of 4 for clarity.

2. *The image sequence.*

   A sequence of 64×48 images are synthesized based on the velocity field described above. Figure 6-19 presents four images from the sequence. Note that the direction of the predominant contrasts in each image changes from mostly horizontal in the early frames to mostly vertical in later frames, implying that some type of temporal coherence constraint is necessary for correct estimation of the flow from this image sequence. As in the case with the Yosemite experiment, we have corrupted the images by adding an independent Gaussian noise with a variance 9 to each pixel to achieve a signal-to-noise ratio of about 85.

3. *The flow estimates and estimation errors.*

   We have used the same single-frame method and suboptimal STC-filter as those used in the Yosemite experiment, except for the parameter change so that $\mu_1 = \mu_2 = 0.025$ (while still using $\rho = 10$), to process the image sequence. Figure 6-20 shows frame 18 of the estimated flow vectors computed by the two algorithms. Clearly, the single-frame method has completely failed to reconstruct the flow field, while the suboptimal STC-filter has performed a

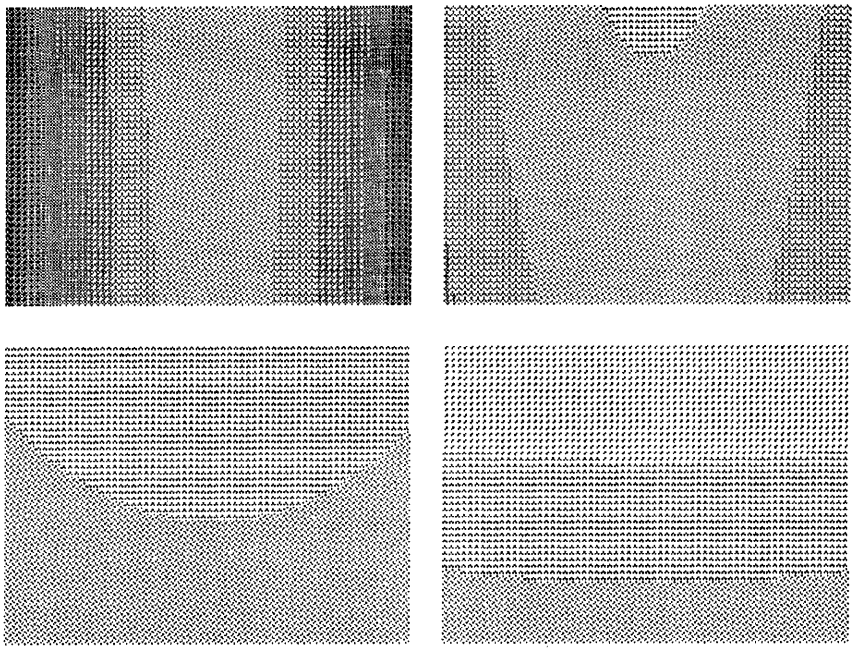


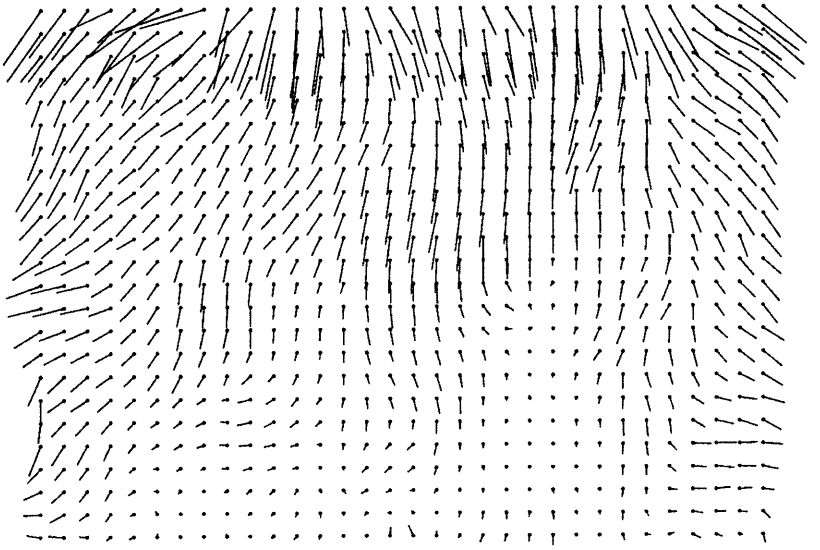
Figure 6-19: The Stagnation Flow image sequence. Frames 0 and 7 (top row) as well as 14 and 21 (bottom row) are shown.

reasonable reproduction of the flow in Figure 6-18. The estimation errors for the two algorithms, shown in Figure 6-21, are consistent with this observation. The error curve associated with the suboptimal STC-filter indicates that the filter has produced increasingly accurate estimates as more measurements are assimilated.

4. *The number of iterations required.*

   Both the single-frame method and suboptimal STC-filter have been allowed to use a maximum of 500 Gauss-Seidel iterations to compute the estimates at each t; however, the actual numbers of iterations required for convergence of the solution (to within 10^{-7}) have typically been lower, as shown in Figure 6-22. Note that, as in the case with the Yosemite experiment, both algorithms initialize each iterative session (except in the first frame) using the respective estimates from the the previous frame. Figure 6-22 indicates that the suboptimal STC-filter requires progressively fewer iterations to compute the estimates (e.g., down to 16 iterations for the estimates in frame 18) and has a much superior convergence property than the single-frame method which requires 100

single-frame



suboptimal STC-filter

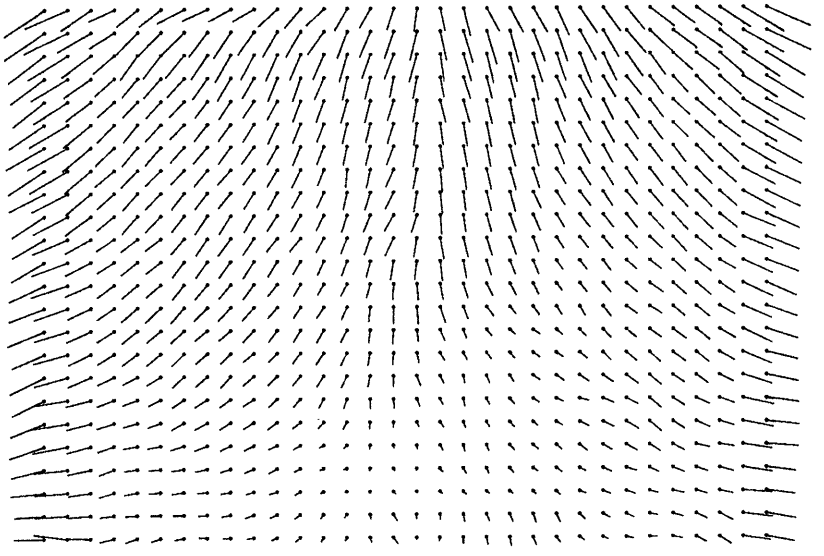


Figure 6-20: The optical flow estimates for frame 18 of the Stagnation Flow sequence by the single-frame method and the suboptimal STC-filter.

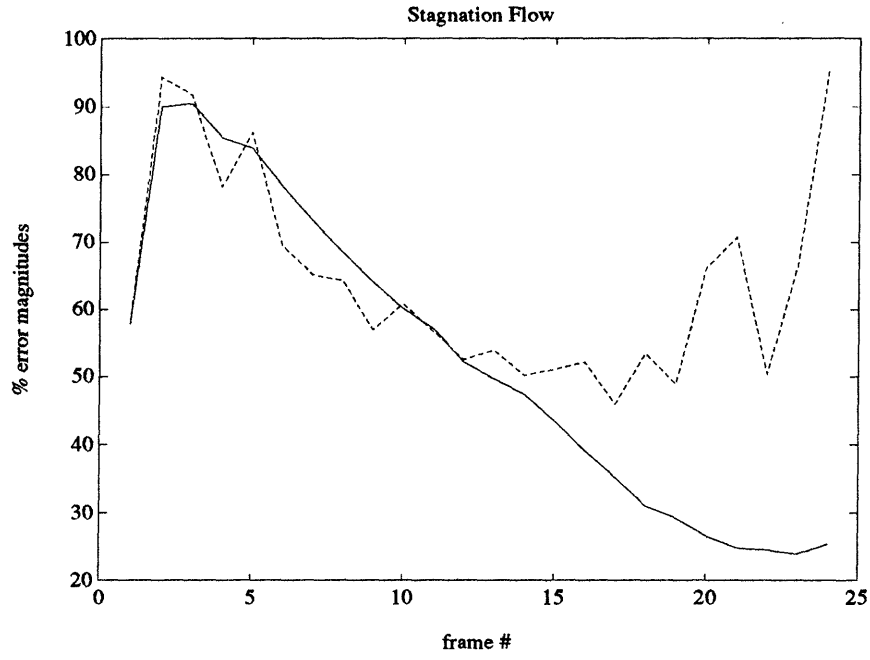


Figure 6-21: The estimation errors by the suboptimal STC-filter (solid-line) and the single-frame method (dotted-line) for the Stagnation Flow experiment.

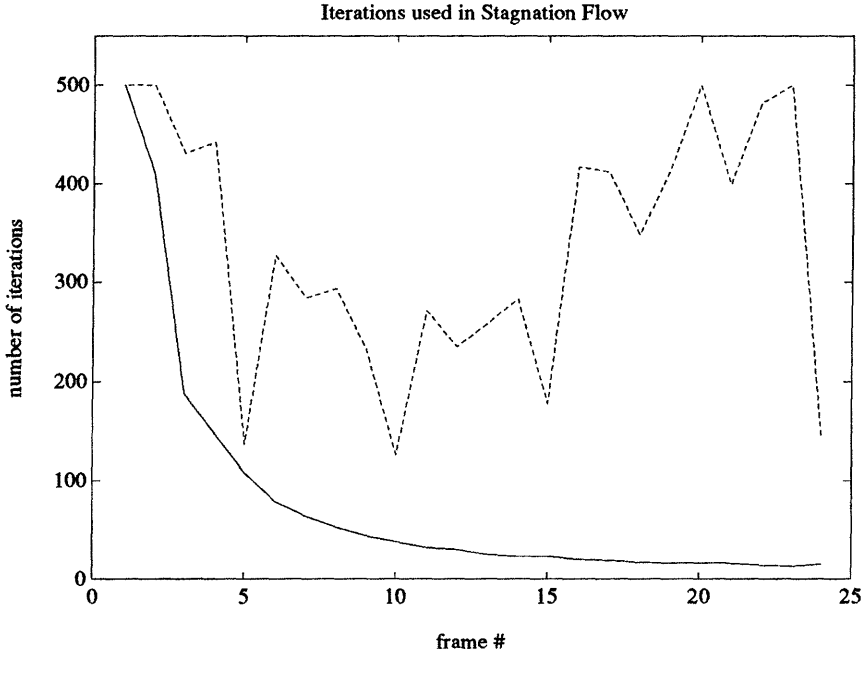


Figure 6-22: The number of iterations used by the suboptimal STC-filter (solid-line) and the single-frame method (dotted-line) for convergence of the estimates in the Stagnation Flow experiment.

to 500 iterations for the estimates in any frame.

6.5 Conclusion

Using primarily the STC model, we have demonstrated that a temporal coherence constraint can improve the quality of the optical flow estimates by performing temporal measurement integration and noise reduction. We have also observed the versatility of the STC model which has been derived from an Eulerian interpretation of a simple, non-parameteric, first order temporal differential model for the time-varying optical flow. The STC-based information Kalman filter is shown to be approximable by using only two series terms in the series-approximation method, leading to a computationally efficient formulation of an effective, general-purpose procedure for multi-frame optical flow estimation.

Chapter 7

Conclusion

7.1 Thesis Summary

A general and efficient algorithmic framework for sequential estimation of time-varying visual fields has been developed in this thesis. A prominent feature of the algorithmic framework is that each of the independent pieces of knowledge about the visual field (i.e., the coherence constraints and measurements) is statistically represented in an implicit format called an information pair, which can also be interpreted as a specification of a *spatial model* for the visual field. In the information pair representation (and its square root version) the statistical data can be efficiently combined over space and time, leading to the development of several useful multi-frame visual reconstruction algorithms that are sequential in time and parallelizable in space. The main contributions of the thesis are:

1. *Multi-frame visual reconstruction with temporal coherence.*

   We have demonstrated that multi-frame visual reconstruction based on *temporal coherence*, which is essentially a temporal extension of well-established spatial coherence constraints (e.g., the membrane and thin-plate models), can yield more robust estimates of smoothly time-varying visual fields than reconstruction by independent, serial applications of a single-frame algorithm. In particular, in multi-frame optical flow estimation we have illustrated that a simple form of the temporal coherence constraint can reduce the effect of noise in measurements

and can in some cases resolve aperture problems not resolvable by imposing only spatial coherence constraints.

2. *Estimation theoretic framework for multi-frame visual reconstruction.*
   A discrete formulation of the traditional, variational approach in single-frame visual reconstruction has been interpreted as a *maximum likelihood* estimation problem, allowing formulation of visual reconstruction in a statistical framework. For multi-frame reconstruction problems, a descriptor dynamic system has been formulated. The system is specified by the dynamic equation obtained from the temporal coherence constraint and the observation equation obtained from the spatial coherence constraint and the measurements (so that the observation equation by itself specifies a single-frame reconstruction problem). Kalman filtering algorithms are formulated based on the descriptor system such that the implicit form of the statistics of the visual field is propagated over time for sequential estimation.

3. *Efficient and effective filter approximation based on spatially local modeling.*
   The information pair representation of the statistical data has been shown to be computationally advantageous in visual field reconstruction because of its compactness and its ease in fusing data (by simple component-wise addition). The information pair (as well as the square root information pair) can be interpreted to specify a spatial model for the unknown field at various stages of estimation. In a single-frame reconstruction problem, such models are spatially local and can be shown to be equivalent to Markov Random Field modeling of the visual field [17, 71]. In a multi-frame problem, on the other hand, the models are not strictly local, and as a result the corresponding Kalman filter faces an enormous computational complexity when implemented optimally. Based on the conjecture that visual fields should be specifiable by spatially local models, however, we have developed computationally efficient, approximate filtering techniques that have been demonstrated to yield near-optimal estimates. A key result in this filter approximation is that the information (as well as the square

root information) matrix in multi-frame visual reconstruction can effectively be approximated by sparse matrices corresponding to spatially local specifications of visual fields.

7.2 Extensions and Future Work

7.2.1 Statistical framework for visual computation

We speculate that the statistical estimation theoretic framework presented in this thesis for a class of low-level visual processing tasks can be extended for formulation of some higher-level tasks. Specifically, the statistical representation of the estimates offers a measure of uncertainty which should be useful in assessing the quality in the estimates produced by low-level processing modules as well as in combining these estimates. Also, the storage and computational efficiency provided by the information pair representation for the statistics can make implementation of a complex, multi-level visual processing task practical.

Data fusion

Enhanced, robust estimation of a visual field may be possible by combining several simultaneous estimates of the field based on different visual cues. For example, we can consider computing optical flow not only using the brightness gradient measurements but also relying on some sparse yet accurate motion vectors obtained by feature matching. Such a merger of primary estimates (e.g., two optical flow fields respectively based on brightness gradients and feature matching) can be easily accomplished using the information pair representations of the estimates. The information pair representation provides us with a unified framework to perform computation necessary for such a data fusion task.

Hypothesis testing for higher-level vision

High-level visual information processing often requires decision-making based on a set of results provided by low-level modules. The second order statistics (i.e., a measure of uncertainty in estimates) contained in the information pair allows us to formulate this task as a statistical hypothesis testing problem. For example, a statistical approach is shown to be useful in a formulation for reconstructing a smooth surface partially occluded by other surfaces [53], which essentially involves first estimating segmented patches of surfaces from the measurements and then deciding which patches belong to the same surface. Thus, in a statistical framework, a low-level task of visual field estimation can be interfaced naturally with a high-level processing task formulated using statistical decision theory.

High-level to low-level feedback through spatial modeling

As a sequence of images is processed, the visual system gradually "understands" its environment, i.e., the physical properties of the objects in the field of view. We can consider using this high-level understanding of the imaged environment, such as the locations of the object boundaries along which imposing spatial smoothness is not appropriate, to provide the low-level visual field reconstruction modules with improved prior spatial models. Such a "top-down" feedback of visual information can be formulated in the presented framework through refinement of the spatial models implied by the information pair. That is, the spatial model interpretation of the information pair should be useful in capturing inter-level communication of visual information.

7.2.2 Techniques for general large-scale space-time filtering

The filters and approximation techniques developed for multi-frame visual field reconstruction can be studied in terms of general large-scale space-time filtering problems. In particular, extensions and detailed numerical analyses of the approximation techniques developed in Chapter 5 may contribute to general signal processing theory.

Nested series-approximated information filter

The series approximation for matrix inversion used in the information filter (Section 5.3.4) converges only if $\Delta^{-1}\Omega$ has all its eigenvalues within the unit circle in the complex plane. Although we have presented an approximation method such that Δ is a diagonal matrix, we can alternatively *choose* the matrix Δ that satisfies this eigenvalue condition so that the approximation technique is applicable to a wider range of filtering problems. That is, the only reason to choose Δ to be the diagonal part of the matrix to be inverted is that Δ^{-1} is easily computable. However, an alternatively chosen Δ, which is a sparse matrix so that $\Delta^{-1}\Omega$ satisfies the eigenvalue condition, can be inverted by another series approximation, leading to a nesting of the series approximation algorithm. In this way, filtering problems to which the original form of the approximation algorithm are not applicable can now be approximated and made efficient.

Analysis of the iterative SRIF

The iterative SRIF (Section 5.4.2), unlike the series-approximated information filter, theoretically converges for all cases. For this algorithm, a more detailed study of its general convergence properties seems appropriate, so that such implementational parameters for the approximate iterative SRIF (Section 5.4.3) as the number of iterations per nulling session and depth of nulling in each incomplete sweeep can be chosen systematically.

Appendix A

Block Matrix Inversion

Let $\mathbf{A}$ and $\mathbf{B}$ be matrices which are inverses of each other, i.e., $\mathbf{A}^{-1} = \mathbf{B}$. Partition these matrices as

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \tag{A.1}$$

so that $\dim A_{ij} = \dim B_{ij}$ and A_{ii}'s are square. Then, A_{ij}'s and B_{ij}'s are related as follows:

$$\begin{aligned} A_{11} &= \left(B_{11} - B_{12}B_{22}^{-1}B_{21}\right)^{-1} && \text{(A.2)} \\ &= B_{11}^{-1} + B_{11}^{-1}B_{12}A_{22}B_{21}B_{11}^{-1} && \text{(A.3)} \\ A_{22} &= \left(B_{22} - B_{21}B_{11}^{-1}B_{12}\right)^{-1} && \text{(A.4)} \\ &= B_{22}^{-1} + B_{22}^{-1}B_{21}A_{11}B_{12}B_{22}^{-1} && \text{(A.5)} \\ A_{12} &= -B_{11}^{-1}B_{12}A_{22} && \text{(A.6)} \\ &= -A_{11}B_{12}B_{22}^{-1} && \text{(A.7)} \\ A_{21} &= -B_{22}^{-1}B_{21}A_{11} && \text{(A.8)} \\ &= -A_{22}B_{21}B_{11}^{-1} && \text{(A.9)} \end{aligned}$$

proof

By substitution into $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$.

Appendix B

Convergence of Iterative SRIF

Let P and Q be $N \times N$ matrices whose elements are denoted as p_{ij} and q_{ij}, respectively. Consider the matrix

$$\begin{bmatrix} P \\ Q \end{bmatrix} \tag{B.1}$$

and the problem of nulling its lower square block Q by application of a series of Givens rotations. Specifically, let us first examine the following procedure:

Elementary Iterative Nulling Procedure

In each iteration, perform:

- Find the element q_{kl} such that $|q_{kl}| = \max_{i,j} |q_{ij}|$.
- Null the element q_{kl} by Givens rotation against the *diagonal* element of the submatrix P in the same column, i.e., p_{ll}.

Let $P^{(n)}$ and $Q^{(n)}$ be the eliminator and eliminatee blocks, respectively, after n^{th} iteration. To show that the iterative procedure nulls the eliminatee block in the limit, let us define $\Delta^{(n)}$ to be

$$\Delta^{(n)} \equiv \sum_{j=1}^{N} p_{jj}^2, \tag{B.2}$$

i.e., the total energy in the main diagonal of the eliminator block P. Then, we have

$$\Delta^{(n+1)} = \Delta^{(n)} + \left(q_{kl}^{(n)}\right)^2 \tag{B.3}$$

where $\left|q_{kl}^{(n)}\right| \equiv \max_{i,j} \left|q_{ij}^{(n)}\right|$. This can be easily shown by first noting that in the $(n+1)^{\text{th}}$ iteration a Givens rotation is applied to the l^{th} row of $P^{(n)}$ and k^{th} row of $Q^{(n)}$ leaving the other rows unchanged. Since unitary transform like Givens rotation preserves the 2-norms of the operand vectors,

$$\left(p_{ll}^{(n+1)}\right)^2 + \left(q_{kl}^{(n+1)}\right)^2 = \left(p_{ll}^{(n)}\right)^2 + \left(q_{kl}^{(n)}\right)^2.$$

But since $\left(q_{kl}^{(n+1)}\right)^2 = 0$ according to the Elementary Iterative Nulling Procedure defined above, we have

$$\left(p_{ll}^{(n+1)}\right)^2 = \left(p_{ll}^{(n)}\right)^2 + \left(q_{kl}^{(n)}\right)^2$$

leading to (B.3). From (B.3) we observe that $\left\{\Delta^{(n)}\right\}$ is a non-decreasing sequence

$$\Delta^{(n+1)} \geq \Delta^{(n)}, \quad \forall n. \tag{B.4}$$

The sequence $\left\{\Delta^{(n)}\right\}$ is also upper-bounded. To see this, note that unitary transforms preserve the Frobenius norm[1] of the operand matrix [35]. In particular, we have

$$\left\| \begin{bmatrix} P^{(n)} \\ Q^{(n)} \end{bmatrix} \right\|_F^2 = \left\|P^{(n)}\right\|_F^2 + \left\|Q^{(n)}\right\|_F^2 = K, \quad \forall n,$$

where K is a constant. Thus,

$$\Delta^{(n)} \leq \left\|P^{(n)}\right\|_F^2 \leq K. \tag{B.5}$$

[1]The Frobenius norm [18] of a matrix A is defined as $\|A\|_F \equiv \left[\sum_i \sum_j a_{ij}^2\right]^{1/2}$.

The conditions (B.4) and (B.5) imply that the sequence $\left\{\Delta^{(n)}\right\}$ converges, i.e.,

$$\lim_{n\to\infty} \Delta^{(n)} = \Delta_\infty \tag{B.6}$$

for a constant $\Delta_\infty \leq K$. Now if

$$\lim_{n\to\infty} q_{kl}^{(n)} \neq 0,$$

then (B.3) implies that

$$\Delta^{(n+1)} > \Delta^{(n)}.$$

Thus, we must have

$$\lim_{n\to\infty} q_{kl}^{(n)} = 0,$$

and since $\left(q_{kl}^{(n)}\right)^2 = \max_{i,j}\left(q_{ij}^{(n)}\right)^2$,

$$\lim_{n\to\infty} \left\|Q^{(n)}\right\| = 0.$$

The Elementary Iterative Nulling Procedure, therefore, can completely null the eliminatee block Q in the limit.

Proving Theorem 5.1

Theorem 5.1 is concerned with the convergence of the nulling procedure using the set of rotations that we have defined as a "sweep" in Chapter 5. A sweep is different from the Elementary Iterative Nulling Procedure in that each rotation is not selectively nulling the eliminatee element with the largest magnitude. The convergence property (B.6) of the sequence $\left\{\Delta^{(n)}\right\}$, however, holds true regardless of the choice of the eliminatee element to be nulled in each Givens rotation, and this indeed does imply

$$\lim_{n\to\infty} \left\|Q^{(n)}\right\| = 0 \tag{B.7}$$

when iterations of sweeps are used.

Appendix C

Computing Symmetric Square Roots

To use the square root information filter presented in this thesis, we need to specify the square root information matrices $\mathbf{U}^{1/2}(t)$ and $\mathbf{S} \equiv \left[\sum_i \mu_i \mathbf{S}_i^T \mathbf{S}_i\right]^{1/2}$. Note that these matrices are square roots of symmetric, positive semidefinite matrices.

Let us consider computing a square root of a symmetric, positive semidefinite matrix $\mathbf{M}$. For any symmetric and positive semidefinite matrix there is a unique symmetric square root matrix which is positive semidefinite [35]. A direct way to compute such a symmetric square root matrix is via similarity transformation [70]. Specifically, let $\mathbf{M} = \mathbf{T}^T\mathbf{D}\mathbf{T}$, where $\mathbf{D}$ is a diagonal matrix of eigenvalues of $\mathbf{M}$ and $\mathbf{T}$ is a unitary matrix whose columns are eignevectors of $\mathbf{M}$; then, $\mathbf{M}^{1/2} = \mathbf{T}^T\mathbf{D}^{1/2}\mathbf{T}$ is the symmetric square root of $\mathbf{M}$. Eigenvalue problems are computationally expensive for a (typically) large matrix like $\mathbf{U}(t)$ and $\left[\sum_i \mu_i \mathbf{S}_i^T \mathbf{S}_i\right]$ even if we can perform the computation "off-line" from the filter. Alternatively, symmetric square root matrices can be computed iteratively [18]:

Let $\mathbf{X}_0 = \mathbf{I}$ and compute the sequence

$$\mathbf{X}_{k+1} = \frac{1}{2}\left(\mathbf{X}_k + \mathbf{M}\mathbf{X}_k^{-1}\right)$$

for $k = 0, 1, 2, \ldots$

Then $\mathbf{X}_k \rightarrow \mathbf{M}^{1/2}$ as $k \rightarrow \infty$, if $\mathbf{M} > \mathbf{0}$.

It is possible to show [18] that this algorithm converges if $\mathbf{M}$ is strictly positive

definite. In performing the numerical experiments presented in Chapter 5, however, we have found that the following modified scheme works quite well for a positive *semidefinite* $\mathbf{M}$:

Let $\mathbf{X}_0 = \mathbf{I}$ and compute the sequence

$$\mathbf{X}_{k+1} = \frac{1}{2}\left(\mathbf{X}_k + \mathbf{M}\mathbf{X}_k^{-1}\right)$$

for $k = 0, 1, 2, \ldots$ until $\left\|\mathbf{X}_{k+1}^2 - \mathbf{M}\right\| > \|\mathbf{X}_k^2 - \mathbf{M}\|$.
Let $\mathbf{M}^{1/2} = \mathbf{X}_k$.

Appendix D

Optical Flow in Collision

This appendix details dynamics of the optical flow vectors in the Collision image sequence (Section 6.2.2). It also presents an extended Kalman filter for estimation of such optical flow.

A viewer moves along a straight line perpendicular to a "wall" (a plane). On the wall is a fixed brightness pattern. As the viewer, looking at the brightness pattern, moves towards or away from the wall he perceives motion due to the effect of perspective projection. The optical flow patterns that he perceives are expanding flow vector patterns.

Imaging geometry

Let Z be the distance of the viewer from the wall and κ be the focal distance associated with the viewer (thus, the image plane is located $Z + \kappa$ from the wall.). Let (R, θ) be the polar coordinates of a point on the wall. Then, by perspective projection the image of the point appears at the location $(r, \theta + \pi)$, in polar coordinates, on the image plane, where R and r are related by

$$\frac{Z}{R} = \frac{\kappa}{r}. \tag{D.1}$$

This relation can be deduced easily from the geometrical relations among R, Z, κ, and r depicted on Figure D-1. Motion is perceived by tracking a point on the wall

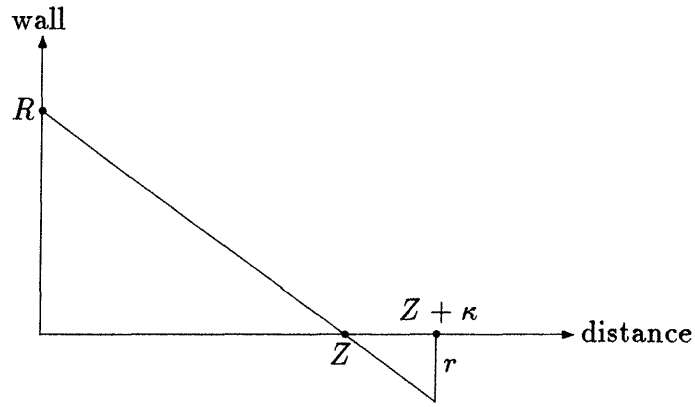


Figure D-1: The point R on the wall projecting its image on the image plane located at $Z + \kappa$.

(distinguished from other points by its associated brightness value) over time; so we fix R (and θ). Since κ is also a given, fixed quantity, we have:

$$Zr = \text{constant}. \tag{D.2}$$

Since the optical flow in this case has an expanding pattern which is circularly symmetric, the flow pattern is independent of θ. Thus, to characterize the optical flow vectors, it suffices to know the changes in r. So, we take the time derivative of (D.2):

$$\frac{d}{dt}(Zr) = Z\dot{r} + \dot{Z}r = 0. \tag{D.3}$$

Thus, we have an expression for the magnitude of the optical flow $\dot{r}$:

$$\dot{r} = -\left(\frac{\dot{Z}}{Z}\right) r. \tag{D.4}$$

The direction of the flow vectors is radially outward with respect to the focus of expansion.

Constant viewer velocity

For the rest of this appendix, we consider the case that the viewer is approaching the wall with a constant velocity, i.e., let

$$\dot{Z} = -v_0 \tag{D.5}$$

$$Z = -v_0 t + Z_0 \tag{D.6}$$

where the constants v_0 and Z_0 represent the speed and initial location of the viewer, respectively. Substituting these into (D.13) we obtain

$$\dot{r} = -\left(\frac{1}{t-T}\right) r \tag{D.7}$$

where $T \equiv \frac{Z_0}{v_0}$ is the "time to collision" when the viewer is at the initial location.

Temporal change in $\dot{r}$

The rate of change in the optical flow magnitude, $\ddot{r}$, can be expressed without a term involving Z. By twice differentiating (D.2) we have

$$\frac{d^2}{dt^2}(Zr) = Z\ddot{r} + 2\dot{Z}\dot{r} + \ddot{Z}r = 0. \tag{D.8}$$

Since $\ddot{Z} = 0$,

$$\ddot{r} = -2\left(\frac{\dot{Z}}{Z}\right)\dot{r}, \tag{D.9}$$

in which we can substitue (D.4) to replace Z's with r's as

$$\ddot{r} = 2\frac{\dot{r}^2}{r}. \tag{D.10}$$

Expressions in the Cartesian system

To convert the polar coordinates (r, θ) to Cartesian coordintes $\mathbf{s}$ we have

$$\mathbf{s} - \mathbf{s}_{\text{FOE}} = r \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{D.11}$$

where $\mathbf{s}_{\text{FOE}}$ is the location of the focus of expansion, i.e., the Cartesian coordinates of the intersection between the Z-axis and the image plane. Let us define $\Delta\mathbf{s} \equiv \mathbf{s} - \mathbf{s}_{\text{FOE}}$ so that its components Δs_1 and Δs_2 satisfy

$$\begin{bmatrix} \Delta s_1 \\ \Delta s_2 \end{bmatrix} = r \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \tag{D.12}$$

Since θ and $\mathbf{s}_{\text{FOE}}$ are time-independent, we can differentiate both sides of this equation to obtain the optical flow vector $f(\mathbf{s}, t)$ as

$$f(\mathbf{s}, t) = \begin{bmatrix} f_1(\mathbf{s}, t) \\ f_2(\mathbf{s}, t) \end{bmatrix} = \dot{r} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{D.13}$$

for a given θ. Moreover, we have

$$\frac{d}{dt} f(\mathbf{s}, t) = \frac{d}{dt} \begin{bmatrix} f_1(\mathbf{s}, t) \\ f_2(\mathbf{s}, t) \end{bmatrix} = \ddot{r} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \tag{D.14}$$

We can use (D.12), (D.13), and (D.14) to eliminate $r, \dot{r}$, and $\ddot{r}$ from (D.10) so that we finally obtain the temporal dynamic model for $f(\mathbf{s}, t)$:

$$\frac{d}{dt} \begin{bmatrix} f_1(\mathbf{s}, t) \\ f_2(\mathbf{s}, t) \end{bmatrix} = 2 \begin{bmatrix} \frac{(f_1(\mathbf{s},t))^2}{\Delta s_1} \\ \frac{(f_2(\mathbf{s},t))^2}{\Delta s_2} \end{bmatrix}. \tag{D.15}$$

Discretized dynamic model

We can write the left hand side of (D.15) in terms of the partial derivatives as

$$\begin{aligned}\frac{d}{dt}f &= \frac{\partial}{\partial t}f + \left(\frac{\partial f}{\partial \mathbf{s}}\right)\frac{d\mathbf{s}}{dt} \\ &= \frac{\partial}{\partial t}f + \left(\frac{\partial f}{\partial \mathbf{s}}\right) f\end{aligned}$$

where the arguments $\mathbf{s}$ and t are omitted for conciseness. By the finite difference approximation $\frac{\partial}{\partial t}f(\mathbf{s},t) \approx f(\mathbf{s},t) - f(\mathbf{s},t-1)$, we have

$$\frac{d}{dt}f(\mathbf{s},t) \approx f(\mathbf{s},t) - \left[I - \left(\frac{\partial f}{\partial \mathbf{s}}\right)\right] f(\mathbf{s},t-1).$$

By substituting this to the left hand side of (D.15) we have a discrete temporal dynamic model for $f(\mathbf{s},t)$:

$$\begin{bmatrix} f_1(\mathbf{s},t) \\ f_2(\mathbf{s},t) \end{bmatrix} = \begin{bmatrix} 1 - \frac{\partial f_1}{\partial s_1} & -\frac{\partial f_1}{\partial s_2} \\ -\frac{\partial f_2}{\partial s_1} & 1 - \frac{\partial f_2}{\partial s_2} \end{bmatrix} \begin{bmatrix} f_1(\mathbf{s},t-1) \\ f_2(\mathbf{s},t-1) \end{bmatrix} + 2 \begin{bmatrix} \frac{(f_1(\mathbf{s},t-1))^2}{\Delta s_1} \\ \frac{(f_2(\mathbf{s},t-1))^2}{\Delta s_2} \end{bmatrix} \qquad \text{(D.16)}$$

where the spatial partial derivatives are evaluated at the pixel $(\mathbf{s},t-1)$.

Extended Kalman filter

1. *Notation*

   Equation (D.16) specifies a mapping from $f(\mathbf{s},t-1)$ to $f(\mathbf{s},t)$. Let us describe the nonlinear function of $f(\mathbf{s},t-1)$ on the right hand side of (D.16) as $\gamma(\cdot)$ so that (D.16) can be expressed as

   $$f(\mathbf{s},t) = \gamma(\, f(\mathbf{s},t-1)\,). \qquad \text{(D.17)}$$

   Note that $\mathbf{s}_{\text{FOE}}$ is a parameter of the function $\gamma(\cdot)$.

Let $\mathbf{f}(t)$ be the optical flow field at time t, i.e., formally, it is a column vector constructed by a lexicographic ordering of the optical flow vectors $f(\mathbf{s}, t)$. We define the non-linear operator $\mathbf{\Gamma}(\cdot)$ to express the temporal dynamics of $\mathbf{f}(t)$

$$\mathbf{f}(t) = \mathbf{\Gamma}(\, \mathbf{f}(t-1)\,) \tag{D.18}$$

so that each component of $\mathbf{f}(t)$ is obtained from the corresponding component in $\mathbf{f}(t-1)$ according to (D.17). Note that $\mathbf{\Gamma}(\cdot)$ operates locally in space and that $\mathbf{s}_{\text{FOE}}$ is a parameter of the function.

2. *Temporal dynamic model.*

   We use the model

$$\mathbf{f}(t) = \mathbf{\Gamma}(\, \mathbf{f}(t-1)\,) + \mathbf{q}(t), \quad \mathbf{q}(t) \sim \left(\, 0, \rho^{-1}\mathbf{I}\,\right) \tag{D.19}$$

   where $\mathbf{q}(t)$ is uncorrelated over time. The addition of the "fictious process noise" $\mathbf{q}(t)$ is a standard technique in sequential estimation theorey to account for modeling errors [16]. In our case, the quantization errors essentially account for the entire modeling error. The errors are still substantial especially due to the second order terms and differentiation in the mapping $\mathbf{\Gamma}(\cdot)$.

3. *Extended Kalman filter thoeory.*

   The mean and covariance of the estimates can be propagated through a nonlinear equation for sequential estimation purposes using a technique known as the extended Kalman filter [39, 16]. Essentially, it is a technique based on linear expansion.

   Through the nonlinear mapping $\mathbf{\Gamma}$ in (D.19), the estimate and its covariance are propagated as follows based on the extended Kalman filtering theory:

$$\bar{\mathbf{f}}(t) = \mathbf{\Gamma}(\, \hat{\mathbf{f}}(t-1)\,) \tag{D.20}$$

$$\bar{\mathbf{P}}(t) = \left[\frac{\partial}{\partial \mathbf{f}}\mathbf{\Gamma}\right]^T \hat{\mathbf{P}}(t-1)\left[\frac{\partial}{\partial \mathbf{f}}\mathbf{\Gamma}\right] + \rho^{-1}\mathbf{I} \tag{D.21}$$

where $\widehat{\mathbf{f}}(t-1)$ is used to evaluate $\frac{\partial}{\partial \mathbf{f}}\boldsymbol{\Gamma}$ (thus, it is a matrix of known values).

4. *Extended Kalman filter implementation.*

The extended Kalman filter algorithm corresponding to the system with the process (D.19) and observation (6.7) is as follows:

- predication step

$$\overline{\mathbf{L}}(t) = \rho\mathbf{I} - \rho^2\left(\boldsymbol{\Phi}^T(t)\widehat{\mathbf{L}}(t-1)\boldsymbol{\Phi}(t) + \rho\mathbf{I}\right)^{-1} \quad \text{(D.22)}$$

$$\overline{\mathbf{f}}(t) = \boldsymbol{\Gamma}(\,\widehat{\mathbf{f}}(t-1)\,) \quad \text{(D.23)}$$

$$\overline{\mathbf{z}}(t) = \overline{\mathbf{L}}(t)\overline{\mathbf{f}}(t) \quad \text{(D.24)}$$

- update step

$$\widehat{\mathbf{L}}(t) = \overline{\mathbf{L}}(t) + \mathbf{H}^T(t)\mathbf{N}(t)\mathbf{H}(t) + \mu_1\mathbf{S}_1^T\mathbf{S}_1 + \mu_2\mathbf{S}_2^T\mathbf{S}_2 \quad \text{(D.25)}$$

$$\widehat{\mathbf{z}}(t) = \overline{\mathbf{z}}(t) + \mathbf{H}^T\mathbf{N}(t)\mathbf{g}(t) \quad \text{(D.26)}$$

$$\widehat{\mathbf{f}}(t) = \widehat{\mathbf{L}}^{-1}(t)\widehat{\mathbf{z}}(t) \quad \text{(D.27)}$$

where $\overline{\mathbf{L}}(t)$ and $\widehat{\mathbf{L}}(t)$ denote the predicted and updated information matrices, respectively.

The matrix $\boldsymbol{\Phi}(t)$ is defined as the block diagonal matrix whose blocks are the 2×2 inverse, $A^{-1}(\mathbf{s},t)$, of a matrix defined as

$$A(\mathbf{s},t) \equiv \begin{bmatrix} 1 - \frac{\partial \widehat{f}_1}{\partial s_1} + 4\frac{\widehat{f}_1}{\Delta s_1} & -\frac{\partial \widehat{f}_1}{\partial s_2} \\ -\frac{\partial \widehat{f}_2}{\partial s_1} & 1 - \frac{\partial \widehat{f}_2}{\partial s_2} + 4\frac{\widehat{f}_2}{\Delta s_2} \end{bmatrix}.$$

The order in which $A^{-1}(\mathbf{s},t)$'s are placed as the diagonal blocks of $\boldsymbol{\Phi}(t)$ is identical to the order of $f(\mathbf{s},t)$'s in the vector $\mathbf{f}(t)$.

Bibliography

[1] J. K. Aggarwal, L. S. Davis, and W. N. Martin. Correspondence processes in dynamic scene analysis. *Proceedings of the IEEE*, 69:562–572, 1981.

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

[3] P. Anandan. *Measuring Visual Motion from Image Sequences.* PhD thesis, University of Massachusetts, Amherst, 1987.

[4] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, N.J., 1979.

[5] H. Ando. Dynamic reconstruction of 3-D structure and 3-D motion. In *Proceedings of Workshop on Visual Motion*, pages 101–110. IEEE Computer Society Press, 1991. Princeton, NJ.

[6] E. Asplund. Inverses of matrices $\{a_{ij}\}$ which satisfy $a_{ij} = 0$ for $j > i + p$. *Mathematica Scandinavica*, 7:57–60, 1959.

[7] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76:869–889, 1988.

[8] G. J. Bierman. A comparison of discrete linear filtering algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, AES-9:28–37, 1973.

[9] G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation.* Academic Press, New York, 1977.

[10] M. J. Black and P. Anandan. A model for the detection of motion over time. In *Third International Conference on Computer Vision*, pages 33–37. IEEE Computer Society Press, 1990. Osaka, Japan.

[11] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, Massachesetts, 1987.

[12] E. P. Chassignet, D. B. Olson, and D. B. Boudra. Motion and evolution of oceanic rings in a numerical model and in observations. *Journal of Geophysical Research*, 95:22,121–22,140, 1990.

[13] L. D. Cohen and I. Cohen. A finite element method applied to new active contour models and 3D reconstruction from cross sections. In *Third International Conference on Computer Vision*, pages 587–591. IEEE Computer Society Press, 1990. Osaka, Japan.

[14] P. Concus, G. H. Golub, and G. Meurant. Block preconditioning for the conjugate gradient method. *SIAM J. Sci. Stat. Comput.*, 6:220–252, 1985.

[15] G. E. Forsythe and P. Henrici. The cyclic Jacobi method for computing the principal values of a complex matrix. *Trans. Amer. Math. Soc.*, 94:1–23, 1960.

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

[17] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, 1984.

[18] G. H. Golub and C. F. van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, Maryland, 1989.

[19] J. Gotze and U. Schwiegelshohn. A square root and division free Givens rotation for solving least squares problems on systolic arrays. *SIAM J. Sci. Stat. Comput.*, 12:800–807, 1991.

[20] P. Greenway. Temporal regularisation of optical flow. In *Proceedings of SPIE Vol.848 Intelligent Robots and Computer Vision: Sixth in a Series*, pages 478–485, 1987.

[21] W. E. L. Grimson. A computational theory of visual surface interpolation. *Proceedings of the Royal Society of London B*, 298:395–427, 1982.

[22] W. E. L. Grimson. An implementation of a computational theory of visual surface interpolation. *Computer Vision, Graphics, and Image Processing*, 22:39–69, 1983.

[23] N. M. Grzywacz, J. A. Smith, and A. L. Yuille. A common theoretical framework for visual motion's spatial and temporal coherence. In *Proceedings of Workshop on Visual Motion*, pages 148–155. IEEE Computer Society Press, 1989. Irvine, CA.

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

[25] J. Heel. Dynamic motion vision. In *Proceedings of the DARPA Image Understanding Workshop*, 1989. Palo Alto, CA.

[26] J. Heel. Direct estimation of structure and motion from multiple frames. A.I.Memo No. 1190, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1990.

[27] J. Heel. *Temporal Surface Reconstruction.* PhD thesis, Massachusetts Institute of Technology, 1991.

[28] J. Heel and S. Rao. Temporal integration of visual surface reconstruction. In *Proceedings of the DARPA Image Understanding Workshop*, 1990. Pittsburgh, PA.

[29] F. Heitz, P. Perez, E. Memin, and P. Bouthemy. Parallel visual motion analysis using multiscale Markov random fields. In *Proceedings of Workshop on Visual Motion.* IEEE Computer Society Press, 1991. Princeton, NJ.

[30] E. C. Hildreth. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23:309–354, 1984.

[31] B. K. P. Horn. Image intensity understanding. *Artificial Intelligence*, 8:201–231, 1977.

[32] B. K. P. Horn. Hill-shading and reflectance map. *Proceedings of the IEEE*, 69:14–47, 1981.

[33] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, Massachesetts, 1986.

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

[35] R. A. Horn and C. A. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[36] K. Ikeuchi. Determination of surface orientations of specular surfaces by using the photometric stereo method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3:661–669, 1981.

[37] K. Ikeuchi and B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.

[38] A. K. Jain. Advances in mathematical models for image processing. *Proceedings of the IEEE*, 69:502–528, 1981.

[39] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[40] P. G. Kaminski, A. E. Bryson, and S. F. Schmidt. Discrete square root filtering: A survey of current techniques. *IEEE Transactions on Automatic Control*, AC-16:727–736, 1971.

[41] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[42] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9:229–244, 1987.

[43] B. C. Levy, M. B. Adams, and A. S. Willsky. Solution and linear estimation of 2-D nearest-neighbor models. *Proceedings of the IEEE*, 78:627–641, 1990.

[44] B. C. Levy, R. Frezza, and A. J. Krener. Modeling and estimation of discrete-time Gaussian reciprocal processes. to appear in IEEE Transactions on Automatic Control, 1990.

[45] F. L. Lewis. *Optimal Estimation.* John Wiley & Sons, New York, 1986.

[46] F. L. Lewis. A survey of linear singular systems. *Circuits, Systems, and Signal Processing*, 5:3–36, 1986.

[47] F. Ling. Givens rotation based least squares lattice and related algorithms. *IEEE Transactions on Signal Processing*, 39:1541–1551, 1991.

[48] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London B*, 208:385–397, 1980.

[49] D. G. Luenberger. Dynamic equations in descriptor form. *IEEE Transactions on Automatic Control*, AC-22:312–321, 1977.

[50] D. G. Luenberger. Time-invariant descriptor systems. *Automatica*, 14:473–480, 1978.

[51] A. J. Mariano. Contour analysis: A new approach for melding geophysical fields. *Journal of Atmospheric and Oceanic Technology*, 7:285–295, 1990.

[52] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Freeman, San Francisco, 1982.

[53] J. L. Marroquin. Random measure fields and the integration of visual information. Centro de Investigacion en Matematicas, Guanajuato, Mexico, 1991.

[54] D. M. Martinez. *Model-Based Motion Estimation and its Application to Restoration and Interpolation of Motion Pictures.* PhD thesis, Massachusetts Institute of Technology, 1986.

[55] L. H. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3, 1989.

[56] A. Meygret and M. Thonnat. Segmentation of optical flow and 3D data for the interpretation of mobile objects. In *Third International Conference on Computer Vision*, pages 238–245. IEEE Computer Society Press, 1990. Osaka, Japan.

[57] M. S. Murphy and L. M. Silverman. Image model representation and line-by-line recursive restoration. *IEEE Transactions on Automatic Control*, AC-23:809–816, 1978.

[58] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:565–593, 1986.

[59] S. Negahdaripour and B. K. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, 1987.

[60] A. N. Netravali and J. D. Robbins. Motion-compensated television coding: Part I. *Bell System Technical Journal*, 58:631–670, 1979.

[61] A. N. Netravali and J. D. Robbins. Motion-compensated coding: Some new results. *Bell System Technical Journal*, 59:1735–1745, 1980.

[62] R. Nikoukhah, A. S. Willsky, and B. C. Levy. Kalman filtering and Riccati equations for descriptor systems. submitted to IEEE Transactions on Automatic Control, 1991.

[63] M. C. Potter and J. F. Foss. *Fluid Mechanics.* Great Lakes Press, Okemos, Michigan, 1982.

[64] J. L. Prince and E. R. McVeigh. Motion estimation from tagged MR image sequences. submitted to IEEE Transactions on Medical Imaging, 1991.

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

[66] P. Saint-Marc, J.-S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13:514–529, 1991.

[67] B. G. Schunck. The image flow constraint equation. *Computer Vision, Graphics, and Image Processing*, 35:20–46, 1986.

[68] F. C. Schweppe. *Uncertain Dynamic Systems.* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[69] A. Singh. Incremental estimation of image-flow using a Kalman filter. In *Proceedings of Workshop on Visual Motion*, pages 36–43. IEEE Computer Society Press, 1991. Princeton, NJ.

[70] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, New York, 1980.

[71] R. Szeliski. *Baysian Modeling of Uncertainty in Low-level Vision.* Kluwer Academic Publishers, Norwell, Massachuesetts, 1989.

[72] D. Terzopoulos. Image analysis using multigrid relaxation models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:129–139, 1986.

[73] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:413–424, 1986.

[74] H. L. van Trees. *Detection, Estimation, and Modulation Theory (Part I).* John Wiley & Sons, New York, 1968.

[75] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1962.

[76] A. S. Willsky. Relationship between digital signal processing and control and estimation theory. *Proceedings of the IEEE*, 66:996–1017, 1978.

[77] J. W. Woods. Two-dimensional discrete Markovian fields. *IEEE Transactions on Information Theorey*, IT-18:232–240, 1972.

[78] J. W. Woods. Two-dimensional Kalman filtering. In T. S. Huang, editor, *Two-Dimensional Digital Signal Processing I*, pages 155–205. Springer-Verlag, 1981.

[79] J. W. Woods and C. H. Radewan. Kalman filtering in two dimensions. *IEEE Transactions on Information Theorey*, IT-23:473–482, 1977. (also see "Correction" in IT-25: 628-629).