Optical Flow Computation via Multiscale Regularization[1]

Mark R. Luettgen
W. Clem Karl
Alan S. Willsky

June 30, 1992

Department of EECS, MIT
Room 35-439
Cambridge, MA 02139
e-mail: luettgen@athena.mit.edu
tel: 617-253-6172
fax: 617-258-8553

Abstract

A new approach to the problem of computing dense optical flow fields in an image sequence is presented. Standard formulations of this problem require the computationally intensive solution of an elliptic partial differential equation which arises from the often used "smoothness constraint" type regularization. We utilize the interpretation of the smoothness constraint as a "fractal prior" to motivate regularization based on a recently introduced class of multiscale stochastic models. The solution of the new problem formulation is computed with an efficient multiscale algorithm. Experiments on several image sequences demonstrate the substantial computational savings that can be achieved due to the fact that the algorithm is non-iterative and in fact has a per pixel computational complexity which is independent of image size. The new approach also has a number of other important advantages. Specifically, multiresolution flow field estimates are available, allowing great flexibility in dealing with the tradeoff between resolution and accuracy. Error covariance information is also available, which is of considerable use in assessing the accuracy of the estimates. Finally, the algorithm is an excellent "pre-conditioner" for the iterative algorithms associated with the smoothness constraint problem formulation. Indeed, its usefulness should extend to a wide variety of estimation problems including many involving Gauss Markov random fields and spatial processes defined as the solution of noise driven partial differential equations.

EDICS category 1.11.

[1] This work was performed in part while the first author visited the Institut de Recherche en Informatique et Systemes Aleatoires (IRISA), and was also supported by the Air Force Office of Scientific Research under Grant AFOSR-88-0032, by the Draper Laboratory IR&D Program under agreement DL-H-418524, by the Office of Naval Research under Grant N00014-91-J-1004 and by the Army Research Office under Grant DAAL03-36-K-0171.

Contents

1 Introduction

The apparent motion of brightness patterns in an image is referred to as the optical flow [14]. In computational vision, optical flow is an important input into higher level vision algorithms performing tasks such as segmentation, tracking, object detection, robot guidance and recovery of shape information [1, 19, 22, 25, 29]. In addition, methods for computing optical flow are an essential part of motion compensated coding schemes [3, 33].

In this paper, we present a new approach to the problem of computing optical flow. Standard formulations of this problem require the computationally intensive solution of an elliptic partial differential equation which arises from the often used "smoothness constraint" regularization term. We utilize the interpretation of the smoothness constraint as a "fractal prior" to motivate regularization based on a recently introduced class of multiscale stochastic models. These models are associated with efficient multiscale smoothing algorithms, and experiments on several image sequences demonstrate the substantial computational savings that can be achieved through their use.

Our approach is most easily understood by comparing it to the optical flow problem formulation originally proposed by Horn and Schunck [14]. As they discuss, information about the optical flow field can be obtained from the image sequence by making the assumption that changes in image brightness are due only to motion. This leads to the so called *brightness constraint equation* [14]:

$$0 = \frac{d}{dt}E(z_1, z_2, t) = \frac{\partial}{\partial t}E(z_1, z_2, t) + \nabla E(z_1, z_2, t) \cdot x(z_1, z_2, t) \tag{1}$$

where $E(z_1, z_2, t)$ is the image intensity as a function of time t and space (z_1, z_2), $x(z_1, z_2, t)$ is the optical flow vector field, and:

$$x = \begin{bmatrix} \frac{\partial z_1}{\partial t} & \frac{\partial z_2}{\partial t} \end{bmatrix}^T \tag{2}$$

$$\nabla E = \begin{bmatrix} \frac{\partial E}{\partial z_1} & \frac{\partial E}{\partial z_2} \end{bmatrix} \tag{3}$$

The brightness constraint equation (1) does not completely specify the flow field since it provides only one linear constraint for the two unknowns at each point. This is usually referred to as the aperture problem [14]. One way to obtain a unique solution is to *regularize* the problem by imposing an additional *smoothness constraint.* Specifically, one formulates the following optimization problem [14]:

$$\hat{x}_{SC} = \underset{x}{argmin} \iint R^{-1} \, (\frac{d}{dt}E)^2 + \|\nabla x\|^2 \, dz_1 dz_2 \tag{4}$$

The smoothness constraint is captured by the second term which penalizes large gradients in the optical flow. The constant R allows one to tradeoff between the relative importance in the cost function of the brightness and smoothness constraint terms. We refer to the optical flow estimate obtained from (4) as the smoothness constraint (SC) solution to the problem of computing optical flow.

One of the major problems associated with the formulation in (4) is that it leads to computationally intensive algorithms. Specifically, one can show that the solution of (4)

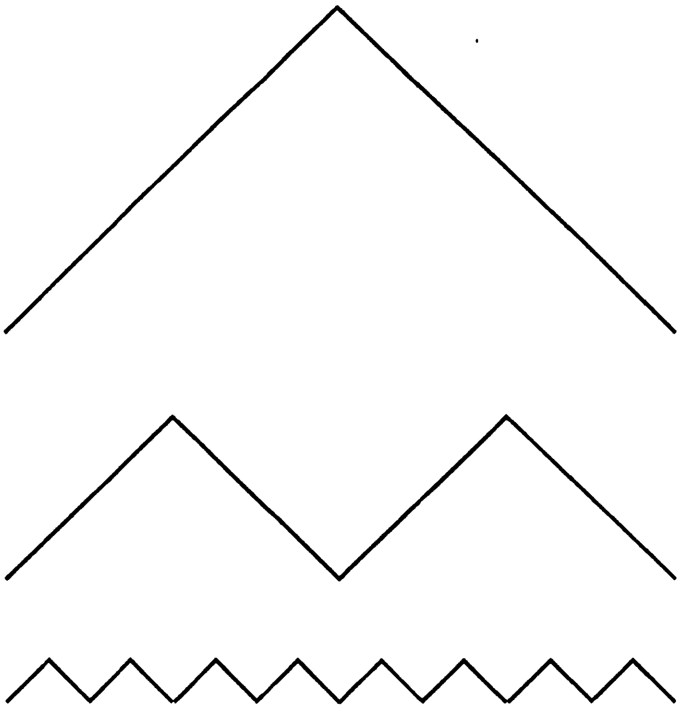

Figure 1: Depiction of three fields which are equally favored by the smoothness constraint, illustrating how this penalty provides a fractal prior model for the optical flow.

satisfies an elliptic partial differential equation (PDE) [14]. Discretization of this PDE leads to a sparse but extremely large set of linear equations. The equations are typically solved using iterative approaches, which require increasing numbers of iterations as the image size grows. One of the first iterative approaches used was the Gauss-Seidel relaxation algorithm [14, 27] which is extremely simple, but converges very slowly. Terzopoulos [31] proposed the use of *multigrid* approaches and reported a factor of 7 reduction in computation over the Gauss-Seidel approach. Successive over-relaxation (SOR) algorithms [16] also provide significant computational improvement over GS approaches and have been successfully used in [21, 23, 24].

The main purpose of this paper is to address the computational issue discussed above. To do this, we need to analyze the smoothness constraint in more detail. Note that the penalty associated with the smoothness constraint term in (4) is equal to the integral of the squared norm of the field gradient over the image plane. In a one-dimensional context, such a constraint would penalize each of the (one-dimensional) fields in Figure 1 equally. Intuitively, the smoothness constraint has a fractal nature, and in fact this can be demonstrated in a much more precise sense. As discussed in Rougee et. al. [23, 24], the optical flow problem formulation in (4) has an equivalent formulation in an estimation-theoretic context. Specifically, let us make the assumption that the two components of the optical flow field are independent, two-dimensional Brownian motions[2]. Next, suppose that

[2] More precisely, since we do not want to bias the optical flow estimates towards zero, we only assume that the *gradients* of the optical flow field components are equal to the *gradients* of the Brownian motion processes. This avoids placing a constraint on the DC (i.e. average) value of the optical flow and focuses

we interpret the brightness constraint as a noisy measurement of flow field:

$$-\frac{\partial}{\partial t}E(z_1,z_2,t) = \nabla E(z_1,z_2,t)\cdot x(z_1,z_2,t) + v(z_1,z_2,t) \tag{5}$$

where $v(z_1,z_2,t)$ is zero mean white Gaussian noise with intensity R. Then, the statistically optimal estimate of the flow field, given the measurements (5) and the Brownian motion prior model, is *the same as the optical flow estimate given by (4).* Thus, the smoothness constraint and the estimation-theoretic formulations are equivalent. The estimation-theoretic interpretation simply allows us to interpret the smoothness constraint as a Brownian motion model. In one-dimension, Brownian motion is a statistically self-similar, fractal process with a $1/f^2$ generalized spectrum [18], and for this reason the smoothness constraint is often referred to as a "fractal prior" [30].

Given that the smoothness constraint prior model has been introduced solely for regularization purposes, we are led to the idea of *replacing* it with another prior model which is similar in its fractal nature, but which leads to a computationally more attractive problem formulation. Replacing the prior model simply means that we want to change the regularization term in (4). The interpretation of the smoothness constraint as a fractal prior suggests that other prior models which also have self-similar characteristics would provide comparable flow estimates. In this paper, we demonstrate an approach which substitutes a fractal-like class of prior models recently introduced in [7, 8, 9, 11] for the smoothness constraint term. The structure of the new penalty term leads to an extremely efficient algorithm for the computation of optical flow estimates. The algorithm is not iterative and in fact requires a fixed number of floating point operations per pixel *independent of image size.* Thus, since all of the known methods for solving the smoothness constraint problem formulation have per pixel computational complexities that grow with image size, the computational savings associated with the new approach increases as the image size grows.

In addition to the computational savings, there are several other advantages that the new approach has. First, the models provide a representation of the optical flow field at multiple resolutions. For this reason, we say that they provide a multiscale regularization (MR) of the optical flow problem, and we refer below to the MR algorithm and solution. The multiscale representation of the flow field should provide great flexibility in dealing with the tradeoff between accuracy and resolution. Specifically, one can expect to obtain higher accuracy at lower resolutions, and one should be able to use the flow estimates at one resolution to decide if computational resources should be used to compute representations at finer resolutions.

Second, because we develop the approach in an estimation-theoretic context, the flow estimates have associated error covariance matrices which provide information about their quality. This information would be essential to addressing the tradeoff between resolution and accuracy as discussed above, and may also be useful to higher level vision algorithms which need to combine information in a rational way from a variety of sources [26]

A third advantage of the approach is that it is an excellent pre-conditioner for algorithms which compute the smoothness constraint solution. We stress that the optical flow estimates obtained with the new problem formulation will not exactly equal those given by (4), since

only on imposing a preference for smoothness in the flow.

the prior model is not exactly the same. Experimental evidence in Section 3 shows that the difference between the SC and MR flow estimates consists of mostly high spatial frequency components. These are precisely the components which can be quickly removed by the iterative algorithms computing a smoothness constraint solution. Thus, *if* one requires the estimate based on the smoothness constraint, it can be obtained by using, for instance, a "pre-conditioned SOR" algorithm, which utilizes the MR solution as an initial estimate of the optical flow, and we present results that demonstrate the efficiency of this approach.

Finally, our approach should be applicable in a substantially more general setting. In particular, it may provide a computationally attractive alternative to standard approaches to the broad class of estimation problems in which the underlying process to be estimated is modeled as a Gaussian Markov random field or is obtained as the solution of noise driven partial differential equations, or in which a "smoothness constraint" type regularization is employed.

This paper is organized as follows. In Section 2 we discuss in more detail an estimation-theoretic interpretation of the optical flow formulation in (4) and develop our new approach to the computation of optical flow. Section 3 presents experimental results on several real and synthetic image sequences. Section 4 provides further discussion and conclusions.

2 Multiscale Regularization

In the first part of this section we develop a discrete formulation of the optical flow problem, and discuss in more detail the estimation-theoretic interpretation of it. We then illustrate precisely how the smoothness constraint can be interpreted as a prior model for the flow field, and how it can be replaced by another, similar prior model which leads to a computationally more attractive problem formulation. The general class of prior models we use is then introduced along with an algorithm for finding the solution of the new optical flow problem formulation.

2.1 An Estimation-Theoretic Interpretation of the Optical Flow Problem

We start by introducing the following notation. Define:

$$y(z_1, z_2) \equiv -\frac{\partial}{\partial t}E(z_1, z_2, t) \tag{6}$$

$$C(z_1, z_2) \equiv \nabla E(z_1, z_2, t) \tag{7}$$

The brightness constraint equation (1) can then be written:

$$y(z_1, z_2) = C(z_1, z_2) \cdot x(z_1, z_2) \tag{8}$$

where the time dependence of the equations has been suppressed. In practice, brightness measurements are only available over a discrete set of points in space and time. Thus, the temporal and spatial derivative terms in the brightness constraint equation (8) must be approximated with finite differences, and the optical flow is only estimated on a discrete space-time grid. There are a number of important issues which arise due to the discretization; we refer the reader to [6] for a detailed discussion. We will assume here that the optical flow is to be estimated on the set $\{(z_1, z_2)|z_1 = ih, z_2 = jh; i, j \in \{1, \cdots, 2^M\}\}$ where h is the grid spacing and M is an integer. The assumption that the lattice is square and that the number of rows is equal to a power of two makes the subsequent development easier, but this is not essential as we discuss in Appendix A. Abusing notation, we write the brightness constraint (8) at the point (ih, jh) as:

$$y(i,j) = C(i,j) \cdot x(i,j) \tag{9}$$

where $y(i,j)$ is the measured temporal brightness derivative, $x(i,j)$ is the optical flow, and $C(i,j)$ is the spatial gradient of the image brightness at grid point (ih, jh).

The brightness constraints (9) at all grid points can be grouped into one large set of linear equations to capture the optical flow information contained in the image sequence. Defining $\mathbf{x}$ as the vector of optical flow vectors at all grid points (using, say, a lexicographic ordering), $\mathbf{C}$ as the matrix containing the corresponding spatial gradient terms $C(i,j)$, and $\mathbf{y}$ as the vector of temporal gradients, we can write:

$$\mathbf{y} = \mathbf{Cx} \tag{10}$$

Then, the discrete counterpart of (4) is:

$$\begin{aligned} \hat{\mathbf{x}}_{SC} &\equiv \underset{\mathbf{x}}{argmin} \, ||\mathbf{y} - \mathbf{Cx}||^2_{\mathbf{R}^{-1}} + ||\mathbf{Lx}||^2_I \\ &= \underset{\mathbf{x}}{argmin} \, (\mathbf{y} - \mathbf{Cx})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{Cx}) + \mathbf{x}^T \mathbf{L}^T \mathbf{Lx} \end{aligned} \tag{11}$$

where the matrix $\mathbf{L}$ is a discrete approximation of the gradient operator in (4) and $\mathbf{R} = RI$. The regularization term $\mathbf{x}^T\mathbf{L}^T\mathbf{Lx}$ makes the optimization problem (11) well-posed. In particular, the solution of (15) satisfies the so called *normal equations* [28]:

$$(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} + \mathbf{L}^T\mathbf{L})\hat{\mathbf{x}}_{SC} = \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} \tag{12}$$

and the invertibility of $(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} + \mathbf{L}^T\mathbf{L})$ guarantees that $\hat{\mathbf{x}}_{SC}$ is unique. The normal equations (12) are the discrete counterpart of the partial differential equation that arises from (4).

An estimation-theoretic formulation of the optimization problem in (11) can now be developed, and we will use it to show that the *statistically optimal* estimate of the optical flow, given a particular set of measurements, is identical to the smoothness constraint solution given in (11). Specifically, given the measurements:

$$\mathbf{y} = \mathbf{Cx} + \mathbf{v} \tag{13}$$

$$\mathbf{0} = \mathbf{Lx} + \mathbf{w} \tag{14}$$

and the assumptions that[3] $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, I)$, the measurement vector $\bar{\mathbf{y}} \equiv [\mathbf{y}^T|\mathbf{0}]^T$ is conditionally Gaussian, and the maximum likelihood estimate [32] of $\mathbf{x}$ is:

$$\hat{\mathbf{x}}_{ML} \equiv \underset{\mathbf{x}}{argmax} \, p(\bar{\mathbf{y}}|\mathbf{x})$$

[3] The notation $\mathbf{z} \sim \mathcal{N}(\mathbf{m}, \mathbf{\Lambda})$ means that $\mathbf{z}$ has a Gaussian distribution, with mean $\mathbf{m}$ and variance $\mathbf{\Lambda}$.

$$
\begin{aligned}
&= \underset{\mathbf{x}}{argmin} - \log p(\bar{\mathbf{y}}|\mathbf{x}) \\
&= \underset{\mathbf{x}}{argmin}\, [\text{constant}] + \frac{1}{2}\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{C} \\ \mathbf{L} \end{bmatrix}\mathbf{x}\right)^T \begin{bmatrix} \mathbf{R} & \\ & I \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{C} \\ \mathbf{L} \end{bmatrix}\mathbf{x}\right) \\
&= \underset{\mathbf{x}}{argmin}\, (\mathbf{y} - \mathbf{C}\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{C}\mathbf{x}) + \mathbf{x}^T\mathbf{L}^T\mathbf{L}\mathbf{x} \qquad (15) \\
&= \hat{\mathbf{x}}_{SC}
\end{aligned}
$$

Thus, the maximum likelihood problem formulation results in the same solution as the smoothness constraint formulation when **L** is used to define an additional set of noisy measurements. The main point here is that by formulating the problem in this estimation-theoretic framework, we can use (14) to interpret the smoothness constraint as a prior probabilistic model for the flow field. Specifically, we can rewrite (14) as:

$$\mathbf{L}\mathbf{x} \;=\; -\mathbf{w} \qquad (16)$$

Recalling that **L** is an approximation to the gradient operator, we see that (16) is nothing more than a spatial difference equation model for **x** driven by the spatial white noise field **w**.

To some extent the precise form of this prior model is arbitrary, and thus we are led to the idea of introducing a new prior model which is similar in nature, but which leads to a computationally more attractive problem formulation. That is, we want to change the smoothness constraint term $\mathbf{x}^T\mathbf{L}^T\mathbf{L}\mathbf{x}$ in (15) to something similar, say, $\mathbf{x}^T\mathbf{S}\mathbf{x} \approx \mathbf{x}^T\mathbf{L}^T\mathbf{L}\mathbf{x}$ (where **S** is a symmetric positive semi-definite matrix) such that the resulting optimization problem is easy to solve. If we factor **S** as $\mathbf{S} = \bar{\mathbf{L}}^T\bar{\mathbf{L}}$ then we can interpret the new constraint as a prior probabilistic model just as we did with the smoothness constraint. In addition, there is a precise interpretation of what we have done as a Bayesian estimation problem. Specifically, if **S** is invertible and we let $\mathbf{\Lambda}^{-1} = \mathbf{S}$, then the use of this new constraint in place of the smoothness constraint is equivalent to modeling the flow field probabilistically as $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$, since in this case the Bayes' least squares estimate of the flow field **x**, given the measurements in (13) is given by:

$$\hat{\mathbf{x}}_{BLSE} \;=\; \underset{\mathbf{x}}{argmin}\, (\mathbf{y} - \mathbf{C}\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{C}\mathbf{x}) + \mathbf{x}^T\mathbf{\Lambda}^{-1}\mathbf{x} \qquad (17)$$

which corresponds to (15) with a different prior model term. The normal equations corresponding to (17) are given by:

$$(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} + \mathbf{\Lambda}^{-1})\hat{\mathbf{x}}_{BLSE} \;=\; \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} \qquad (18)$$

Comparison of the problem formulations (11) and (17), or of the normal equations (12) and (18), makes it apparent how the two problem formulations are related. Note that an analogous Bayesian interpretation can apparently be given to the smoothness constraint formulation (11), (12), with the corresponding prior model for optical flow given by $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, (\mathbf{L}^T\mathbf{L})^{-1})$. Recall, however, that **L** is an approximation to the spatial gradient operator and thus is *not* invertible since operating on constants with this operator yields zero. The probabilistic interpretation of this is that the model (16) places probabilistic constraints on the spatial *differences* of the optical flow, but not on its DC value. Indeed, it is not difficult to check that if we model optical flow instead as $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, (\mathbf{L}^T\mathbf{L} + \epsilon I)^{-1})$, where ϵ is any arbitrarily small positive number, then $\mathbf{L}^T\mathbf{L} + \epsilon I$ is indeed invertible and the DC value of $\mathbf{x}$ has a prior covariance P_0 on the order of $1/\epsilon$, so that $P_0 \to \infty$ as $\epsilon \to 0$. Thus, the original smoothness constraint formulation in essence assumes an infinite prior covariance on the DC value of optical flow. The alternate model developed in the next section has a similar parameter, P_0, representing the DC variance, which can similarly be set to ∞.

The choice of the new prior model is now clearly at the heart of the problem. Recalling that the smoothness constraint has the interpretation as a "fractal prior", we would like to choose a prior model which also has fractal-like characteristics. A natural way to specify such models is to explicitly represent the optical flow field at multiple scales. A stochastic modeling framework which allows us to do this, and which also leads to efficient algorithms for solving (17), (18), is described in the next section.

2.2 A Class of Multiscale Models

The models we utilize to replace the smoothness constraint prior model were recently introduced in [7, 8, 9, 11]. The models represent the flow field at multiple scales, i.e. for a set of scales $m = 0, \ldots, M$, with $m = 0$ being the coarsest scale and $m = M$ the finest scale, we define a set of optical flow fields indexed by scale and space, namely $x_m(i,j)$. At the m^{th} scale, the field consists of 4^m flow vectors, as illustrated in Figure 2, capturing features of the optical flow field discernible at that scale (i.e. finer resolution features of the field appear only in finer scale representations). Thus, the coarsest version of the flow field consists of just a single vector corresponding to the average value of the optical flow over the entire spatial domain of interest, and successively finer versions consist of a geometrically increasing number of vectors. At the finest level, the flow field is represented on a grid with the same resolution as the image brightness data. In particular, $x_M(i,j)$ corresponds to the optical flow vector $x(i,j)$ in (9).

Abstractly, we are representing the flow field on the *quadtree structure* illustrated in Figure 3. Pyramidal data structures such as the quadtree naturally arise in image processing algorithms which have a multiresolution component. For instance, successive filtering and decimation operations lead to images defined on such a hierarchy of grids in the Laplacian pyramid coding algorithm of Burt and Adelson [5] and in the closely related wavelet transform decomposition of images [17]. Also, the multigrid approaches to low level vision problems discussed by Terzopoulos [31] involve relaxation on a similar sequence of grids.

The model we introduce in this section describes in a probabilistic manner how the optical flow field $x(i,j) = x_M(i,j)$ is constructed by adding detail from one scale to the next. Just as the smoothness constraint prior model (16) describes probabilistic constraints among values of the optical flow at different spatial locations, our multiscale model describes such constraints among values at different *scales*. That is, our model describes the probabilistic evolution of $x_m(i,j)$ as the scale m evolves from coarse to fine. For notational convenience in describing such models, we denote nodes on the quadtree with a single abstract index s which is associated with the 3-tuple (m,i,j) where, again, m is the scale and (i,j) is a spatial location in the grid at the m^{th} scale. It is also useful to define an *upward shift operator* $\bar{\gamma}$. In particular, the *parent* of node s is denoted $s\bar{\gamma}$ (see Figure 3). We note that the operator $\bar{\gamma}$ is not one-to-one; it is in fact four-to-one since each node will have four

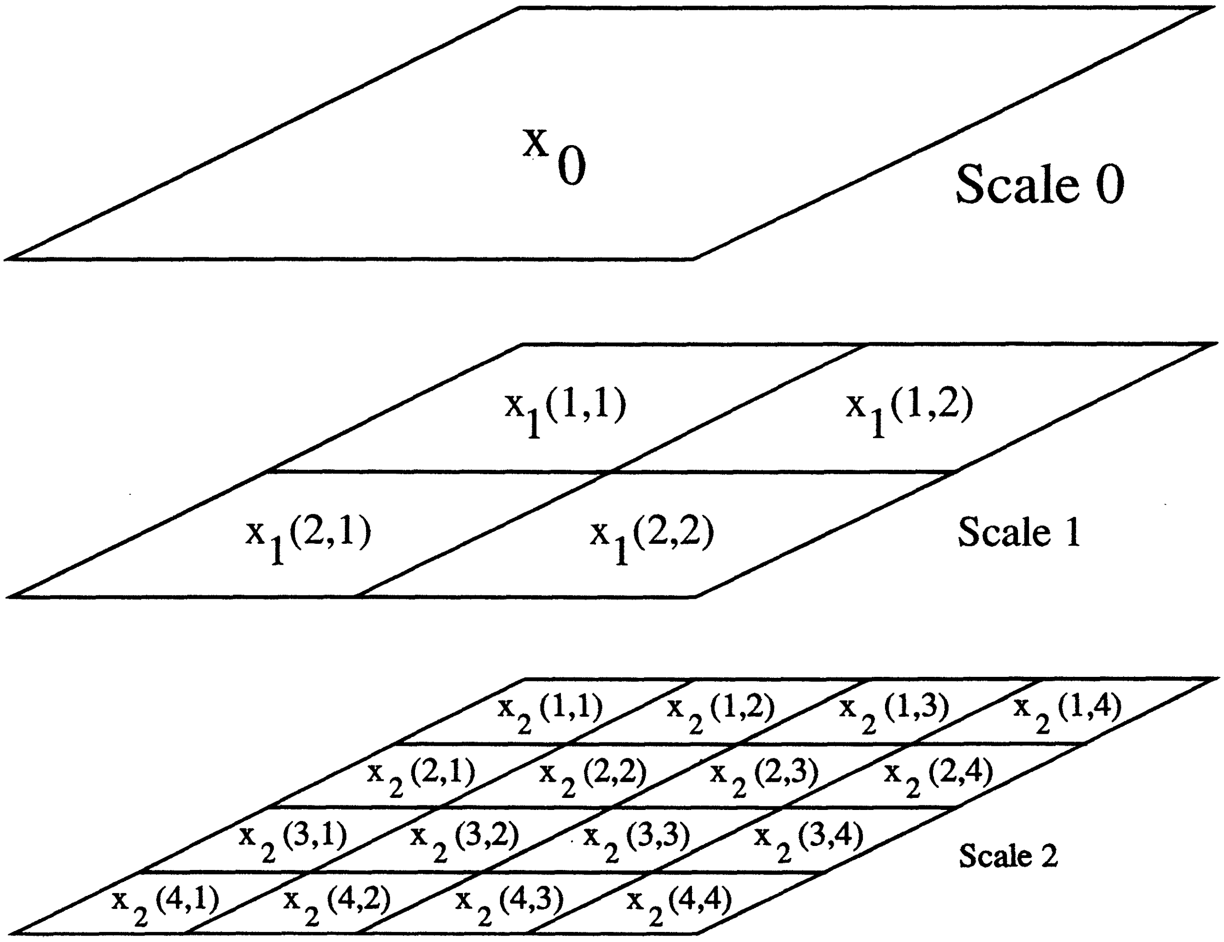

Figure 2: The structure of a multiscale optical flow field is depicted. The components of the field are denoted $x_m(i,j)$ where m refers to the scale and the pair (i,j) denotes a particular grid location at a given scale. At the coarsest scale, there is a single flow vector and, more generally, at the m^{th} scale there are 4^m vectors.

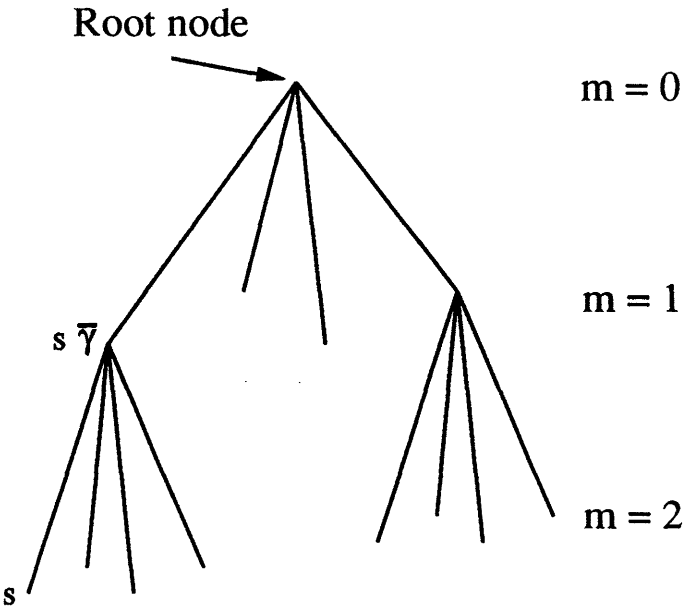

Figure 3: Quadtree structure on which the multiscale processes are defined. The abstract index s refers to a node in the quadtree; $s\bar{\gamma}$ refers to the parent of node s.

"offspring" at the next scale. For instance, if s corresponds to any of the nodes in the upper left quadrant of the second level grid (see Figure 2), i.e. nodes $(2,1,1),(2,2,1),(2,1,2)$ or $(2,2,2)$, then $s\bar{\gamma}$ corresponds to their parent on the first level, namely node $(1,1,1)$.

We are now in a position to describe the class of multiscale models which describe the evolution of multiscale stochastic processes indexed by nodes on the quadtree. Specifically, a stochastic quadtree process $x(s)$ is described recursively by:

$$x(s) \quad = \quad A(s)x(s\bar{\gamma}) + B(s)w(s) \tag{19}$$

under the following assumptions:

$$x_0 \sim \mathcal{N}(0, P_0) \tag{20}$$

$$w(s) \sim \mathcal{N}(0, I) \tag{21}$$

The vectors x and w are referred to as the state and driving noise terms. The state variable x_0 at the root node of the tree provides an initial condition for the recursion. The driving noise is white in both space and scale, and is uncorrelated with the initial condition. Interpreting each level as a representation of a two-dimensional field, we see that (19) describes the evolution of the process from coarse to fine scales. The term $A(s)x(s\bar{\gamma})$ represents interpolation down to the next level, and $B(s)w(s)$ represents higher resolution detail added as the process evolves from one scale to the next. In the application of interest here, $x(s) = x_m(i,j)$, where $s = (m,i,j)$, and thus $A, B \in \Re^{2\times 2}$. Such a model corresponds in essence to a first-order recursion in scale for optical flow.[4]

Measurements of the finest level optical flow field are available from the brightness constraint. In particular, at a particular point (i,j) at the finest level M, we have a measurement equation corresponding to that in (9):

$$\begin{aligned} y(i,j) &= C(i,j)x_M(i,j) + v(i,j) \qquad (22)\\ v(i,j) &\sim \mathcal{N}(0,R) \qquad (23) \end{aligned}$$

where $C(i,j) \in \Re^{1\times 2}$ and the white Gaussian observation noise is assumed to be independent of the initial condition x_0 and the driving noise w in (19) – (21). Of course, we can group the state variables $x(s)$ at the finest level into a vector $\mathbf{x}_M$ as well as the corresponding measurements $y(s)$ and spatial gradient terms $C(s)$ in the same way as we did to get (10):

$$\begin{aligned} \mathbf{y} &= \mathbf{C}\mathbf{x}_M + \mathbf{v} \qquad (24)\\ \mathbf{v} &\sim \mathcal{N}(\mathbf{0},\mathbf{R}) \qquad (25) \end{aligned}$$

We now have exactly the framework which led to the statement of (17) as an alternative to the smoothness constraint formulation (15). In particular, the modeling equations (19) – (21) indicate that at the finest level of the quadtree, the flow field vectors will be a set of jointly Gaussian random variables $\mathbf{x}_M \sim \mathcal{N}(\mathbf{0},\mathbf{\Lambda})$, where $\mathbf{\Lambda}$ is implicitly given by the parameters in (19) – (21), and a set of noisy measurements given by (24). The Bayes' least squares estimate of $\mathbf{x}_M$ given the measurements in (24) and the prior model is:

$$\hat{\mathbf{x}}_M = \underset{\mathbf{x}_M}{argmin}\, (\mathbf{y} - \mathbf{C}\mathbf{x}_M)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{C}\mathbf{x}_M) + \mathbf{x}_M^T\mathbf{\Lambda}^{-1}\mathbf{x}_M \qquad (26)$$

The multiscale modeling framework thus provides an alternative to the smoothness constraint formulation of (11) or (15).

What remains to be done are (1) to specify a model within this class that has characteristics similar to those of the smoothness constraint prior model, and (2) to demonstrate why the use of this alternate multiresolution formulation is of any interest. We defer the latter of these to the next section and focus here on the former. In particular, for our multiscale model based on (19) – (21) to approximate the smoothness constraint prior we would like to choose our model parameters so that we have $\mathbf{L}^T\mathbf{L} \approx \mathbf{\Lambda}^{-1}$. The observation that the prior model implied by the operator $\mathbf{L}$ in (15) corresponds to a Brownian motion "fractal prior" suggests one approach to choosing the model parameters. In particular, the one-dimensional Brownian motion has a $1/f^2$ generalized spectrum [18]. It has been demonstrated that such processes are well approximated by multiscale models such as ours in one dimension if geometrically decreasing powers of noise are added at each level m of the process [9, 34]. In particular, this motivates the choice of $B(s) = b4^{-\frac{\mu m(s)}{2}}I$ in (19), where I is the 2×2 identity matrix, and where b and μ are scalar constants. The constant b directly controls the overall noise power in the process. Also, as discussed in [34], the choice of μ

[4] More generally, higher-order recursions in scale can be captured, just as in standard state space models, by increasing the order of the model, i.e. the dimension of $x(s)$. In this case the actual optical flow at node s would correspond to a subset of the components of $x(s)$, with the remainder of $x(s)$ devoted to capturing the memory in the multiscale recursion. In this paper, however, we restrict ourselves to the simple first order recursion.

controls the power law dependence of the generalized spectrum of the process at the finest resolution as well as the fractal dimension of its sample paths. Specifically, this spectrum has a $1/f^{\mu}$ dependence. Thus, the choice of $\mu = 2$ would correspond to a Brownian-like fractal process. To achieve greater flexibility in both the modeling and estimation, we allow μ to be a parameter that can be varied. In addition, recall that in the smoothness constraint formulation, $\mathbf{L}^T\mathbf{L}$ was not invertible because of the implicit assumption of infinite prior variance on the DC value of the optical flow field. In our multiscale regularization context, this would correspond to setting P_0 equal to infinity in (20). This can be done without difficulty in the estimation algorithms described next, but we have found that it is generally sufficient to simply choose P_0 to be a large multiple of the identity.

In closing this section let us illustrate some of the types of sample paths that result from scalar versions of multiresolution models of this type (i.e. where $x(s)$ is a scalar). Sample paths of the finest level of $x(s)$ from several scalar processes for different parameter choices of such a model are illustrated in Figures 4 to 6. The processes are shown as mesh plots to better illustrate the changes induced by choosing different model parameters. As noted above, the parameter $A(s)$ reflects interpolation from coarse to fine scales. As $A(s)$ goes to zero, the scale-to-scale "memory" of the process decreases, and in fact if $A(s) = 0$, the process will be white noise at each level. The effect of changing $A(s)$ is illustrated in Figures 4 and 5. The parameter μ in the driving noise term also effects the correlation structure of the process. As μ is increased, the driving noise variance is reduced from scale-to-scale at a faster rate. Thus, there is greater interscale correlation, and we expect the process to appear more structured. The effect of changing μ is illustrated in comparing Figures 4 and 6. As these figures indicate, the choice of $A(s) = 1$ leads to the fine scale process having obvious memory of coarse scale features, while the larger value of μ in Figure 6 leads to sample paths with smaller fine scale fluctuations than Figure 4.

2.3 The Multiscale Regularization Algorithm

We have now specified a class of models which will allow us to approximate the smoothness constraint prior model. The simple multiscale structure of these models leads to very efficient algorithms for computing the optimal estimate of the state given a set of measurements. One of these algorithms, which we refer to as the Multiscale Regularization (MR) algorithm, was developed in [7, 8, 9, 10] for one-dimensional signals, and its extension to images is described here.

The MR algorithm computes the Bayes least squares estimate of the state vectors (19) given the measurements (22) in two steps. The first step is an *upward* or *fine-to-coarse* sweep on the quadtree, which propagates the measurement information in parallel, level by level, from the fine scale nodes up to the root node. The second step is a *downward* or *coarse-to-fine* sweep which propagates the measurement information back down, and throughout the tree. The result is the least squares estimate of the state $x(s)$ at each node based on all of the data. The details of the upward and downward sweeps are given below and are discussed in much greater detail in [9, 10].

To begin, note first that the measurement model (22), allowing for the possibility of spatially varying noise intensity, can be written in the form:

$$y(s) \quad = \quad C(s)x(s) + v(s) \tag{27}$$

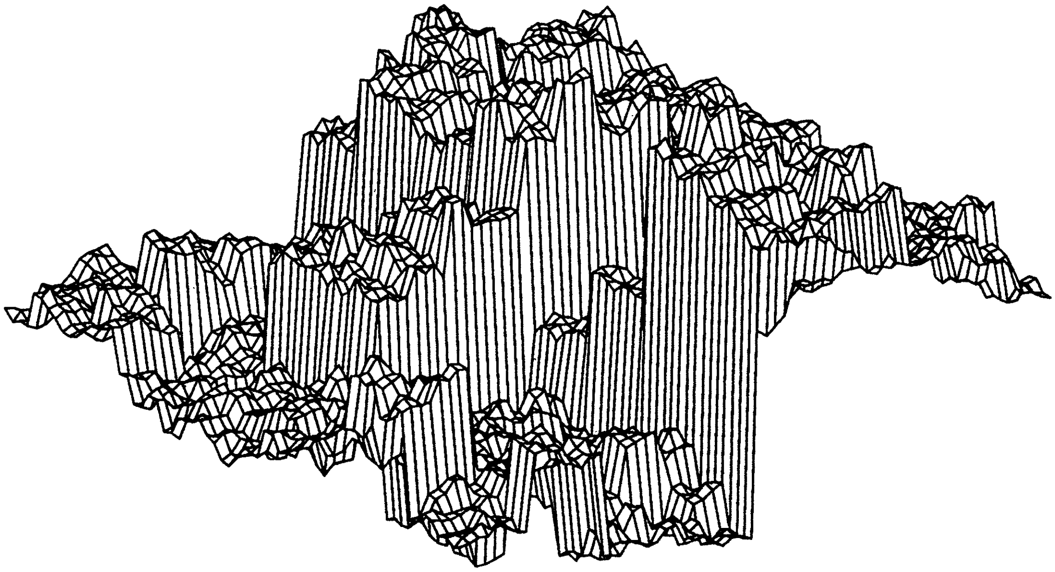

Figure 4: Finest scale of a scalar quadtree process sample path: $A(s) = 1, B(s) = 4^{-\frac{\mu m(s)}{2}}, \mu = 1$.

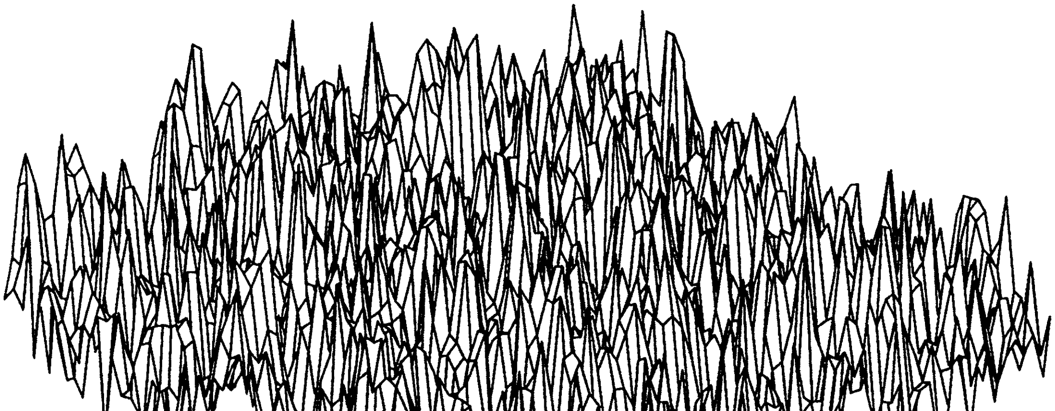

Figure 5: Finest scale of a scalar quadtree process sample path: $A(s) = 0.5, B(s) = 4^{-\frac{\mu m(s)}{2}}, \mu = 1$.

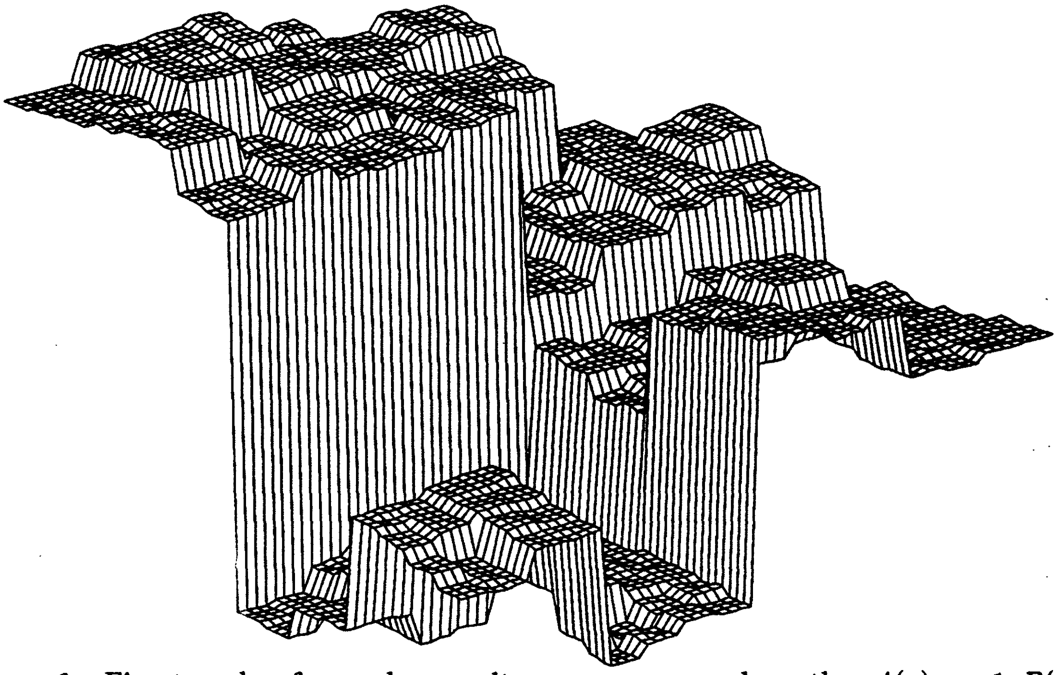

Figure 6: Finest scale of a scalar quadtree process sample path: $A(s) = 1, B(s) = 4^{-\frac{\mu m(s)}{2}}, \mu = 2$.

$$v(s) \sim \mathcal{N}(0, R(s)) \tag{28}$$

In the context of the optical flow estimation problem, measurements are taken only on the finest level, corresponding to $C(s) = 0$ unless s is a node at the finest level (i.e. unless $s = (M, i, j)$). However, in the more general modeling framework discussed in [9, 10], the measurements may be available at any node, and the noise variance may vary with node as in (28). We present here this more general algorithm in which, in addition, x, y and w may be of arbitrary dimension.

The model given by (19) - (21), (27) - (28) is a downward model in the sense that the recursion starts from the root node and propagates down the quadtree from coarse-to-fine scales. In order to describe the upward sweep of the MR algorithm, we need a corresponding *upward model.* This upward model is equivalent to the downward model in the sense that the joint second order statistics of the states $x(s)$ and measurements $y(s)$ are the same. The upward model is given by[5] [8, 9]:

$$x(s\bar{\gamma}) = F(s)x(s) - A^{-1}(s)B(s)\tilde{w}(s) \tag{29}$$

$$y(s) = C(s)x(s) + v(s) \tag{30}$$

where:

$$F(s) = P_{s\bar{\gamma}}A^T(s)P_s^{-1} \tag{31}$$

[5] We use $\mathbf{E}[x]$ to denote the expected value of the random variable x and $\mathbf{E}[x|y]$ to denote the expected value of x given y.

$$\tilde{w}(s) = w(s) - \mathbf{E}[w(s)|x(s)] \tag{32}$$
$$\mathbf{E}[\tilde{w}(s)\tilde{w}^T(s)] = I - B^T P_s^{-1} B(s) \tag{33}$$
$$\equiv \tilde{Q}(s) \tag{34}$$

and where $P_s = \mathbf{E}[x(s)x^T(s)]$ is the variance of the state at node s and evolves according to the Lyapanov equation:

$$P_s = A(s)P_{s\bar{\gamma}}A^T(s) + B(s)B^T(s) \tag{35}$$

To proceed further we need to define some new notation.

$$Y_s = \{y(s')|s' = \text{s or } s' \text{ is a descendant of } s\} \tag{36}$$
$$Y_s^+ = Y_s \setminus \{s\} \tag{37}$$
$$\hat{x}(s'|s) = \mathbf{E}[x(s')|Y_s] \tag{38}$$
$$\hat{x}(s'|s+) = \mathbf{E}[x(s')|Y_s^+] \tag{39}$$
$$\hat{P}(s'|s) = \mathbf{E}[(x(s') - \hat{x}(s'|s))(x(s') - \hat{x}(s'|s))^T] \tag{40}$$
$$\hat{P}(s'|s+) = \mathbf{E}[(x(s') - \hat{x}(s'|s+))(x(s') - \hat{x}(s'|s+))^T] \tag{41}$$

where the notation $Y_s \setminus \{s\}$ means the node s is not included in the set Y_s^+. The upward sweep of the MR algorithm begins with the initialization of $\hat{x}(s|s+)$ and the corresponding error covariance $P(s|s+)$ at the finest level. Then, the estimate *update* is computed via:

$$\hat{x}(s|s) = \hat{x}(s|s+) + K(s)[y(s) - C(s)\hat{x}(s|s+)] \tag{42}$$
$$P(s|s) = [I - K(s)C(s)]P(s|s+) \tag{43}$$
$$K(s) = P(s|s+)C^T(s)V^{-1}(s) \tag{44}$$
$$V(s) = C(s)P(s|s+)C^T(s) + R(s) \tag{45}$$

Denote the offspring of $x(s)$ as $x(s\alpha_i), i = 1, \cdots, q$. For the quadtree model, $q = 4$. The updated estimates are *predicted* back up to the next level:

$$\hat{x}(s|s\alpha_i) = F(s\alpha_i)\hat{x}(s\alpha_i|s\alpha_i) \tag{46}$$
$$P(s|s\alpha_i) = F(s\alpha_i)P(s\alpha_i|s\alpha_i)F^T(s\alpha_i) + Q(s\alpha_i) \tag{47}$$
$$Q(s\alpha_i) = A^{-1}(s\alpha_i)B(s\alpha_i)\tilde{Q}(s\alpha_i)B^T(s\alpha_i)A^{-1}(s\alpha_i) \tag{48}$$

The predicted estimates are then *merged*:

$$\hat{x}(s|s+) = P(s|s+)\sum_{i=1}^{q} P^{-1}(s|s\alpha_i)\hat{x}(s|s\alpha_i) \tag{49}$$

$$P(s|s+) = [(1-q)P_s^{-1} + \sum_{i=1}^{q} P^{-1}(s|s\alpha_i)]^{-1} \tag{50}$$

The upward sweep given by the update, predict and merge equations proceeds recursively up the quadtree. At the top of the tree, one obtains the *smoothed* estimate of the root node, that is, an estimate based on all of the data. The estimate and its error covariance are given by:

$$\begin{aligned} \hat{x}^s(0) &= \hat{x}(0|0) && (51) \\ P^s(0) &= P(0|0) && (52) \end{aligned}$$

where the superscript s denotes the fact that these are *smoothed* estimates, that is, they are based on all of the available data. The smoothed estimate and associated error covariance at the root node provide initialization for the *downward sweep*, which is given by:

$$\begin{aligned} \hat{x}^s(s) &= \hat{x}(s|s) + J(s)[\hat{x}^s(s\bar{\gamma}) - \hat{x}(s\bar{\gamma}|s)] && (53) \\ P^s(s) &= P(s|s) + J(s)[P^s(s\bar{\gamma}) - P(s\bar{\gamma}|s)]J^T(s) && (54) \\ J(s) &= P(s|s)F^T(s)P^{-1}(s\bar{\gamma}|s) && (55) \end{aligned}$$

The estimates $\hat{x}^s(s)$ at the finest level of the quadtree provide the solution to (26). The form of the algorithm we have specified here, which generalizes standard Kalman filtering and smoothing algorithms to the multiscale context, obviously assumes that the state covariance P_s is well defined and finite, and it is not difficult to see from (35) that this will be the case if P_0 is finite. There is, however, an alternate form of this algorithm presented in [9, 10] which generalizes so-called *information form* algorithms for standard state space models and which propagates *inverses* of covariances. In this alternate form it is straightforward to accommodate the setting of P_0 to infinity (which corresponds to $P_0^{-1} = 0$), and we refer the reader to [9, 10] for details. As mentioned previously, we have found that setting P_0 to a large but finite multiple of the identity, and then using (42) - (50), (53) - (55), yields excellent results.

3 Experimental Results

3.1 Specification of the Multiscale Model

To specify the MR algorithm completely we need to choose the parameters in (19) - (21), (27) - (28). We utilize the following parameterization of the model:

$$\begin{aligned} x(s) &= x(s\bar{\gamma}) + (b4^{\frac{-\mu m(s)}{2}})w(s) && (56) \\ y(s) &= C(s)x(s) + v(s) && (57) \\ w(s) &\sim \mathcal{N}(0, I) && (58) \\ v(s) &\sim \mathcal{N}(0, R(s)) && (59) \\ x_0 &\sim \mathcal{N}(0, pI) && (60) \end{aligned}$$

where I is a 2×2 identity matrix. From (56) and (58) we see that the two components of the optical flow field are modeled as independent sets of random variables, and that each will have a fractal-like characteristic due to the choice of the driving noise gain $B(s)$ (as discussed in the previous section). We view μ and b as free model parameters which can be varied to control the degree and type of regularization in much the same way that the parameter R in the smoothness constraint formulation (4) is used to tradeoff between the data dependent and regularization terms in the optimization functional.

As discussed previously, the measurements $y(s)$ and measurement matrix $C(s)$ come directly from the image temporal and spatial gradients, which are available at the finest level of the quadtree. In the experiments described below, we use a simple two-image difference to approximate the temporal gradient. The spatial gradient is computed by smoothing the image with a 3 × 3 Gaussian kernel followed by a central difference approximation. The additive noise variance is given by $R(s)$. We have found empirically that the choice $R(s) = max(||C(s)||^2, 10)$ worked well. This choice effectively penalizes large spatial gradients, which are likely points of occlusion where the brightness constraint equation will not hold [26]. The parameter p in the prior covariance of the root node was set to $p = 100$. The distribution (60) on the root node effectively says that we are modeling the optical flow field components as zero mean random processes. The prior covariance reflects our confidence in this assumption. Since we do not believe that *any* prior assumption on the mean of optical flow field components can be justified, we set the parameter p such that the implied standard deviation is much larger than the sizes of the flow fields we expect to see.

We compare our approach computationally and visually to the the Gauss-Seidel (GS) and successive over-relaxation (SOR) algorithms, which can be used to compute the solution of the smoothness constraint formulation given by (11) or (15). The details of these algorithms can be found in Appendix B. Straightforward analysis shows that the GS and SOR algorithms require 14 and 18 floating point operations (flops) per pixel per iteration respectively. The number of iterations required for convergence of the iterative algorithms grows with image size [16]. For reasonable size images (say, 512 × 512), SOR may require on the order of hundreds of iterations to converge, so that the total computation per pixel can be on the order of $10^3 - 10^4$ flops. On the other hand, the MR algorithm requires 76 flops per pixel (see Appendix C). *Note that the MR algorithm is* not *iterative.* Thus, the computational gain associated with the MR algorithm can be on the order of one to two orders of magnitude.

3.2 Rotation Sequence

The first example is a sequence of Gaussian images modulated by a spatial sinewave. Specifically, the first frame intensity pattern is given by:

$$E(z_1, z_2, t_1) = \sin(\text{atan}(z_1 - 23, z_2 - 28)) \exp(-\frac{1}{2} z' Z^{-1} z) \tag{61}$$

$$z = \begin{bmatrix} z_1 - 23 \\ z_2 - 28 \end{bmatrix} \tag{62}$$

$$Z = \begin{bmatrix} 1000 & 0 \\ 0 & 500 \end{bmatrix} \tag{63}$$

where atan(z_1, z_2) is a 2π arctangent (atan(0,1) = 0, atan(1,0) = $-\pi$), $h = 1$ and $M = 6$ (i.e. the image lattice is 64 × 64, cf. the discussion about discretization at the beginning of Section 2.1). The second frame is equal to the first, rotated by 1 degree about pixel (23,28). The first frame and actual optical flow are illustrated in Figures 7 and 8.

Figure 9 illustrates the flow computed using the MR algorithm. The computed flow reflects the rotational nature of actual flow field, with error at the boundaries induced by the quadtree structure of the prior model. The degree of "blockiness" is determined by the parameters b and μ. Increasing the parameter b raises the level of uncertainty in the prior model, which implies that the MR algorithm will tend to provide less regularization (and hence less blockiness). A larger value of μ increases the prior model correlation between any two given states, thereby inducing more blockiness in the flow estimates.

Figure 10 illustrates the smoothness constraint (SC) flow estimates computed using the SOR algorithm (the SOR algorithm was initialized with identically zero flow estimates). The estimates required 50 SOR iterations to obtain, representing a factor of $50/4.2 = 11.9$ more computation than the MR algorithm. Figure 11 illustrates the root mean square[6] (rms) error in the flow estimates as a function of iteration for the SOR and GS algorithms. As expected, the SOR algorithm is significantly faster than the GS algorithm (they will converge to the same result since they are solving the same partial differential equation). The rms error in the MR flow estimates is depicted as a straight line, since the algorithm is not iterative.

The MR and SC flow estimates are not identical due to differences in the prior models. If there is particular interest in obtaining the SC solution, the question arises of using the MR solution as an initial guess for the iterative algorithms which compute the SC solution. Note that the difference between the SC and MR flow estimates is associated with the non-smooth, high frequency aspects of the MR flow at block edges. It is precisely these high frequency components that are quickly removed by SOR or GS algorithms computing the the smoothness constraint solution and suggests that the MR algorithm would provide an excellent *pre-conditioner* for the iterative algorithms. Figure 12 illustrates the optical flow estimates computed using a *pre-conditioned SOR* algorithm. The estimates correspond to 5 iterations of the SOR algorithm initialized with the MR flow estimates in Figure 9. In this example, the total computation required by the pre-conditioned SOR approach is a factor of $50/(4.2+5) = 5.4$ less than that required by the standard SOR algorithm (i.e. the algorithm initialized with identically zero flow estimates).

Figure 13 illustrates the rms difference between the smoothness constraint solution and the intermediate values of the GS, SOR and pre-conditioned SOR estimates as a function of iteration[7]. The error plot for the pre-conditioned SOR algorithm begins at 4.2 iterations to take into account the initial computation associated with the MR algorithm (which equals 4.2 SOR iterations). The figure demonstrates that the pre-conditioned SOR approach provides a substantially less computationally intensive approach to finding the SC flow estimates even for this small size problem.

3.3 Yosemite Sequence

The second example is a synthetic 256×256 image sequence which simulates the view from a small plane flying through the Yosemite Valley[8]. The first image in the sequence is shown in Figure 14 along with the actual flow field in Figure 15. The flow computed via the MR algorithm is shown in Figure 16 and the smoothness constraint solution is shown in Figure 17. The smoothness constraint flow estimates required 250 SOR iterations in this

[6] The rms error is only one measure of flow field quality and will not be appropriate for all applications (e.g. segmentation or coding).

[7] The smoothness constraint solution is approximated as the SOR algorithm optical flow estimates after 500 iterations.

[8] This sequence was synthesized by Lyn Quam of SRI International.

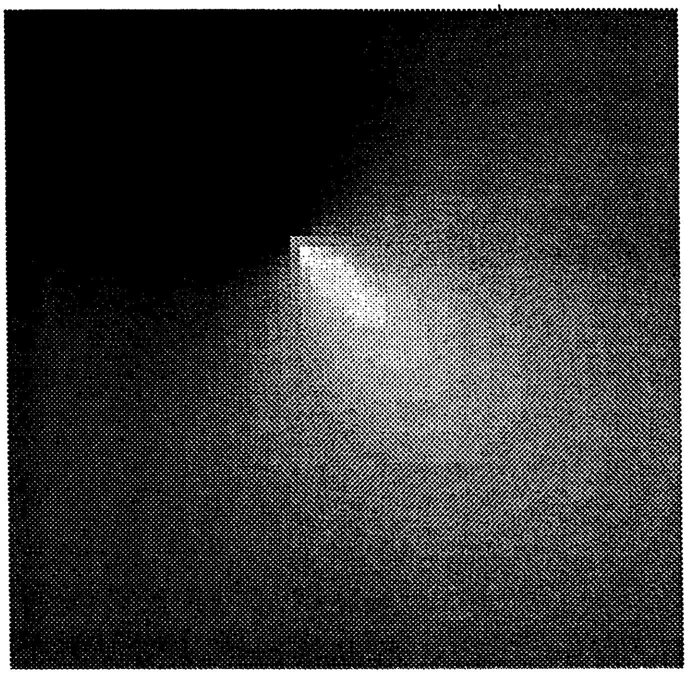

Figure 7: First frame of the "rotation" sequence.

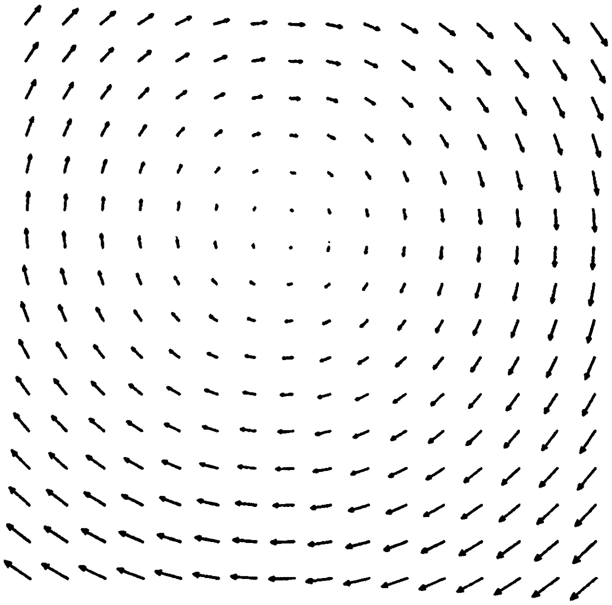

Figure 8: Rotation sequence true optical flow.

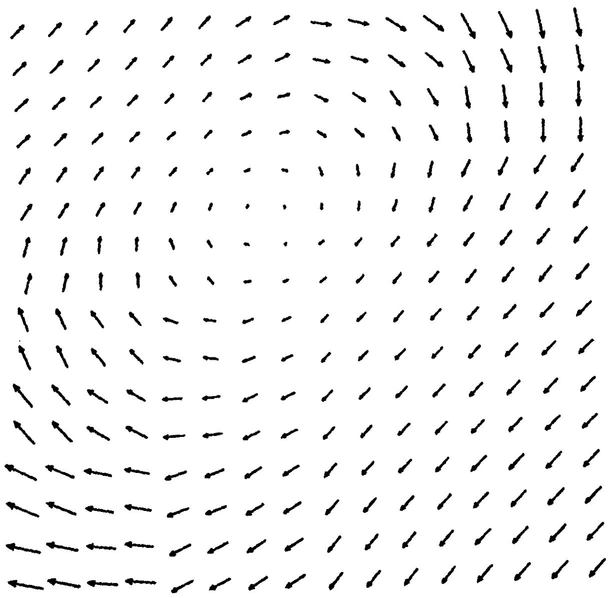

Figure 9: Multiscale Regularization (MR) algorithm flow estimates: $b = 10, \mu = 0.7$.

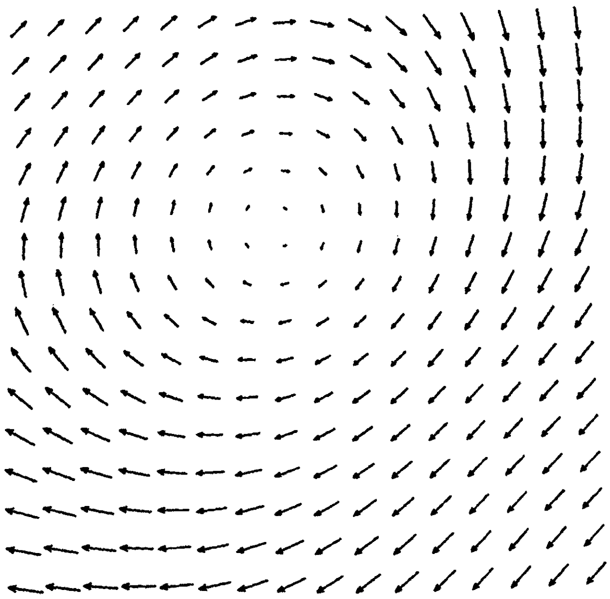

Figure 10: Successive over relaxation (SOR) algorithm flow estimates: $R = 10^2$, 50 iterations. The SOR algorithm required a factor of 11.9 more computation than the MR algorithm in this example.

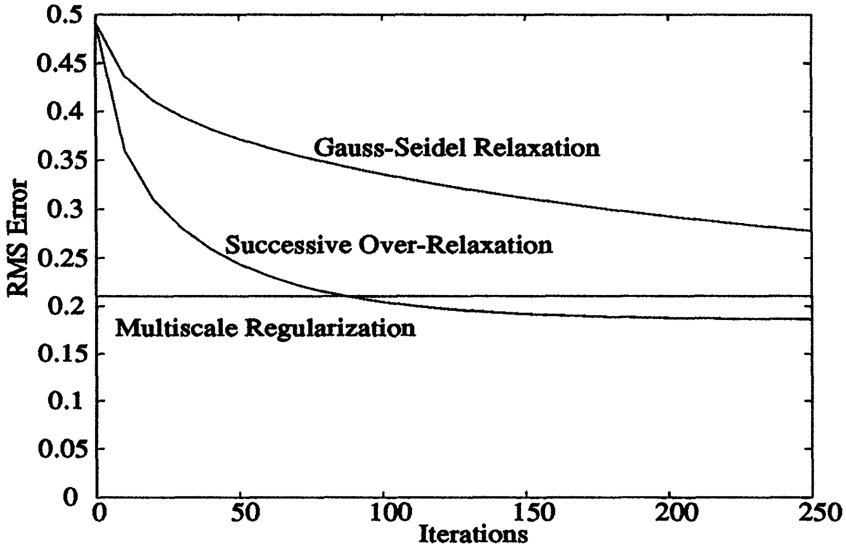

Figure 11: Rms Error Comparison of MR, SOR and Gauss-Seidel (GS) algorithm flow estimates for the rotation sequence.

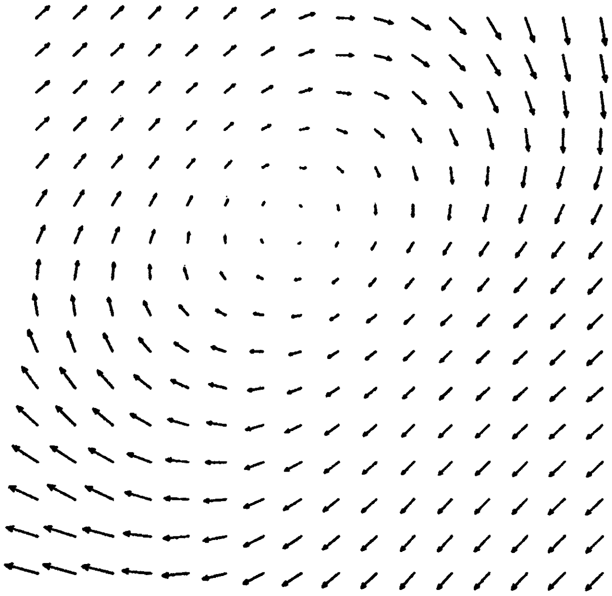

Figure 12: Pre-conditioned SOR algorithm flow estimates: $R = 10^2$, 5 iterations. The pre-conditioned SOR algorithm is initialized with the MR flow estimates.

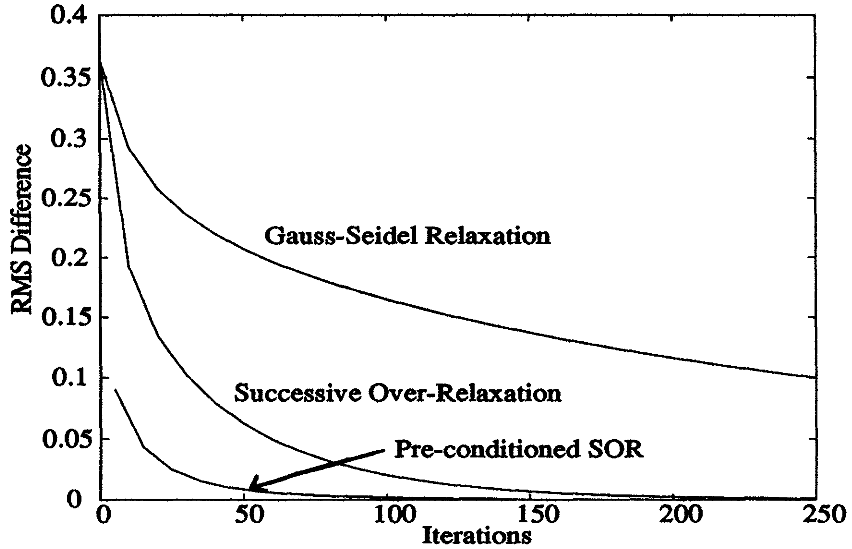

Figure 13: Rms Difference Comparison illustrates how the pre-conditioned SOR, SOR and GS algorithms converge to the smoothness constraint solution. The plots show the rms difference between the smoothness constraint solution and the estimates as a function of iteration. All will eventually converge, but the pre-conditioned SOR algorithm converges much faster than SOR or GS.

example, representing a factor of 60 more computation than the MR estimates. Note the substantial increase over the previous example in the number of iterations required for the SOR algorithm to converge. The number of iterations required for convergence depends on several things, including the parameter R, the image gradient characteristics and the image size. Theoretical analysis in [16] shows that the SOR algorithm requires on the order of N iterations for an $N \times N$ image. Thus, we expect substantially more computational savings as the image size increases.

An rms error comparison of the algorithms is shown in Figure 18. Neither of the estimates coincides with the actual optical flow, but they do have comparable rms error as in the previous example. In addition, the figure illustrates the computational advantage of the MR algorithm. In particular, the SOR algorithm is still reducing the rms error in its flow estimates after 300 iterations, at which point the MR algorithm requires a factor of $300/4.2 = 71.4$ less computation.

Again, there may be interest in obtaining the solution to the smoothness constraint problem formulation. Figure 19 depicts the pre-conditioned SOR results (based on initialization with the MR flow estimates in Figure 16) after 20 iterations. Visually, there is almost no difference between the pre-conditioned SOR estimates and the estimates shown in Figure 17. The computational gain associated with the pre-conditioned algorithm is 10.3. Figure 20 illustrates how the GS, SOR and pre-conditioned SOR algorithms converge to the smoothness constraint solution. For any given number of iterations, the pre-conditioned SOR estimates are substantially closer to the final solution than the GS or SOR estimates.

This image sequence contains a problem often encountered in real images: regions of

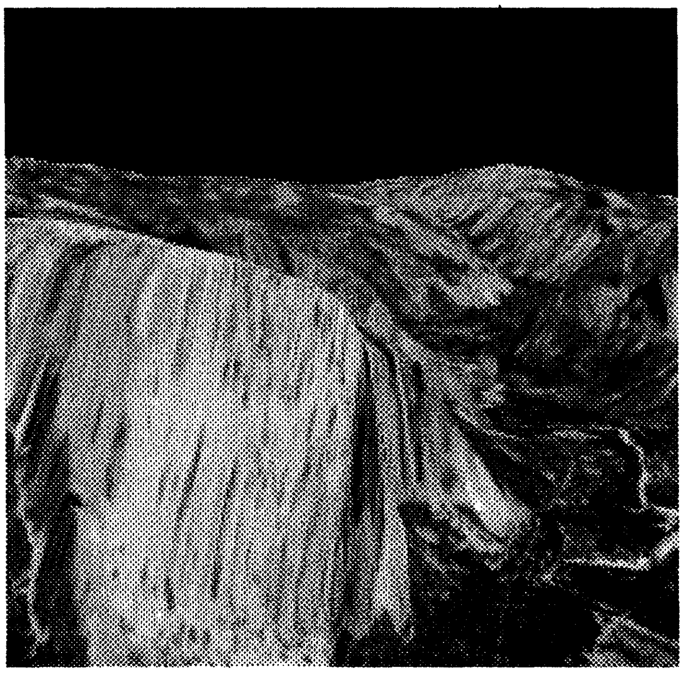

Figure 14: First frame of "Yosemite" sequence.

constant intensity. The problem is that the lack of gradient information in that region implies that the optical flow is not well defined. The smoothness constraint and multiscale prior models provide a means of interpolating out into these regions. The result of this is apparent in the top portion of Figures 16 and 17. An advantage of the MR formulation is that it accomplishes this extrapolation at an appropriately coarser, and hence computationally simpler, scale.

3.4 Moving Vehicle Sequence

The third example is a real[9] 512 × 512 image sequence and depicts the view from a car driving down a road. The first image in the sequence is illustrated in Figure 21. The MR and SOR flow estimates are shown in Figures 22 – 23. In this example, the MR solution required a factor of $800/4.2 = 190$ less computation.

Since the true optical flow is not available (as it was in the previous simulated examples), an alternate performance metric is needed. In particular, we will use a reconstruction error metric, which is often used in contexts in which one is interested in using optical flow for motion compensated coding. This metric measures the mean square difference between the current image in a sequence and an estimate of it based on the computed optical flow, the previous image, and a bilinear interpolation scheme [20]. The optical flow used is that associated with the current image. Essentially, one estimates the brightness at any given point by using the optical flow to project that point back to the previous image. In general, that point will not be on the image plane, and the bilinear interpolation is required.

[9] The sequence was provided by Saab-Scania.

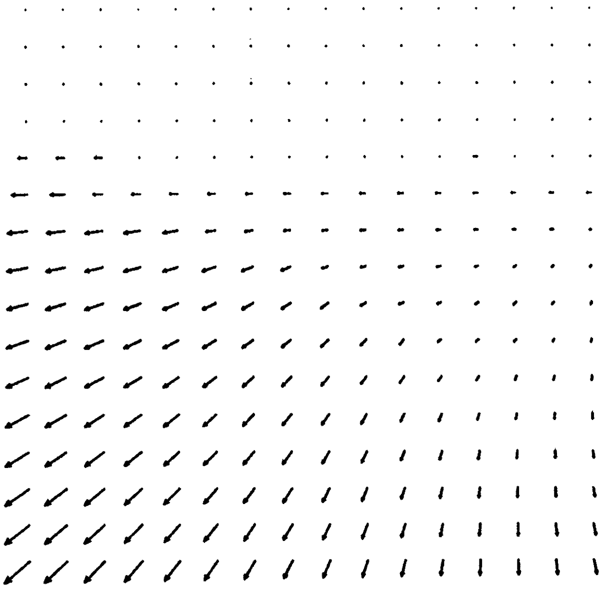

Figure 15: Yosemite sequence true optical flow.

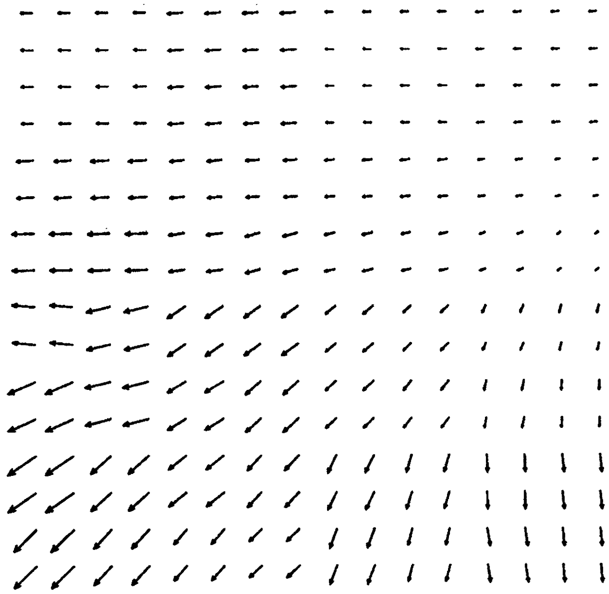

Figure 16: MR algorithm flow estimates: $b = 10, \mu = 2.5$.

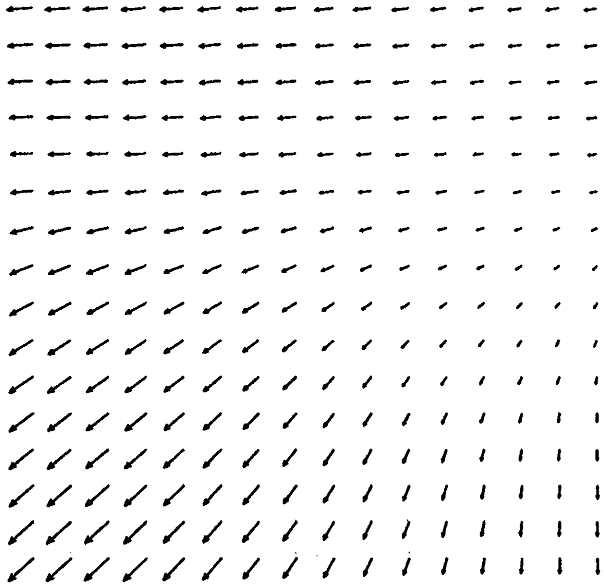

Figure 17: SOR algorithm flow estimates: $R = 500^2$, 250 iterations. The SOR algorithm required a factor of 60 more computation in this example.

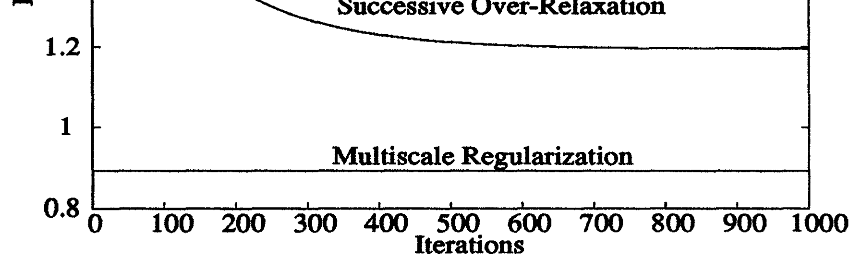

Figure 18: Rms Error Comparison of MR, SOR and Gauss-Seidel (GS) algorithm flow estimates for the Yosemite sequence.

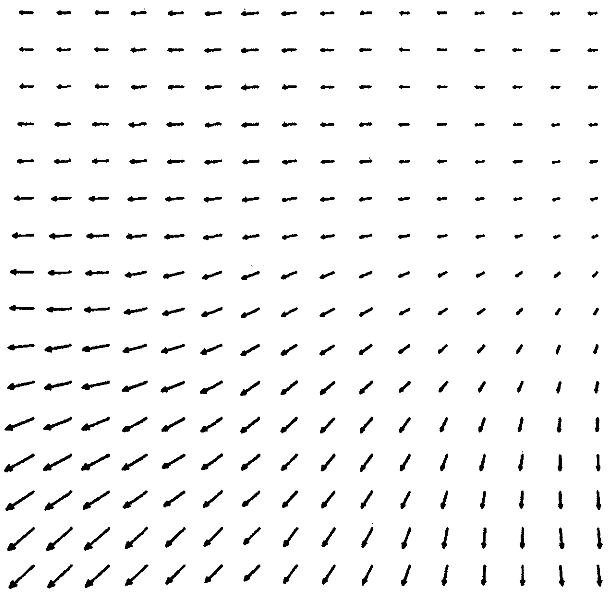

Figure 19: Pre-conditioned SOR estimates: $R = 500^2$, 20 iterations.

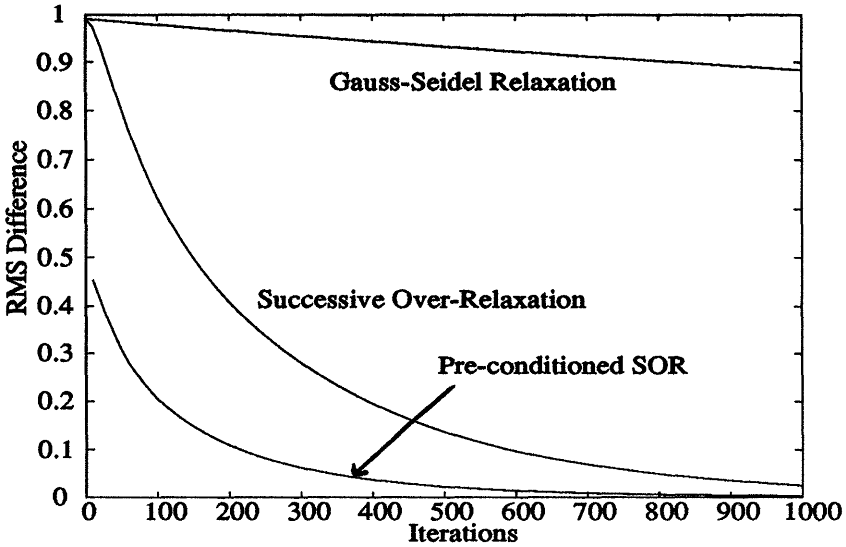

Figure 20: Rms Difference Comparison illustrates how the pre-conditioned SOR, SOR and GS algorithms converge to the smoothness constraint solution. The plots show the rms difference between the smoothness constraint solution and the estimates as a function of iteration. The pre-conditioned SOR algorithm is substantially closer to the final smoothness constraint result after any given number of iterations.

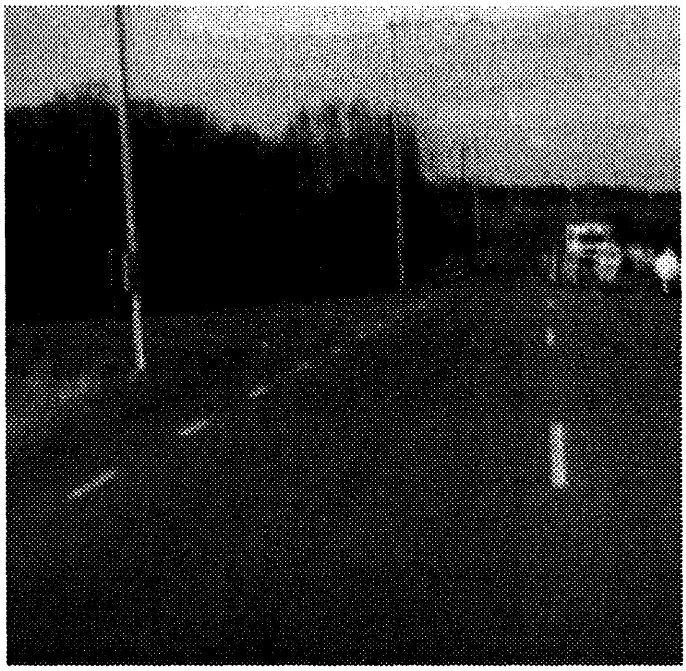

Figure 21: First frame of "Moving Vehicle" sequence.

Figure 24 depicts the rms reconstruction error based on the MR, SOR and GS approaches as a function of iteration and illustrates that two approaches provide comparable reconstruction error results, just as they provided comparable rms error results in the previous examples. In addition, it illustrates the computational advantage of the MR algorithm. Indeed, the SOR algorithm is still improving its optical flow estimates (with respect to this metric) after 200 iterations, at which point the MR algorithm requires a factor of 200/4.2 = 47.6 less computation.

Again, we illustrate how the MR algorithm can be used as a pre-conditioner. Figure 25 depicts pre-conditioned SOR optical flow estimates after 30 iterations. The pre-conditioned SOR results are nearly indistinguishable from the standard SOR estimates in Figure 23 and represent a factor of 23 reduction in computational cost. Figure 26 provides a view of how the GS, SOR and pre-conditioned SOR algorithms converge to the smoothness constraint estimates and illustrates that the pre-conditioned SOR algorithm converges significantly faster than the other algorithms.

Note finally that the number of iterations required for convergence in this example is greater than that in the previous two, again supporting the theoretical results in [16]. Thus, one can expect to achieve greater computational savings through the MR algorithm as the image size grows.

4 Conclusions

We have presented a new approach to the computation of optical flow. A new problem formulation is developed which utilizes the "fractal prior" interpretation of the smoothness

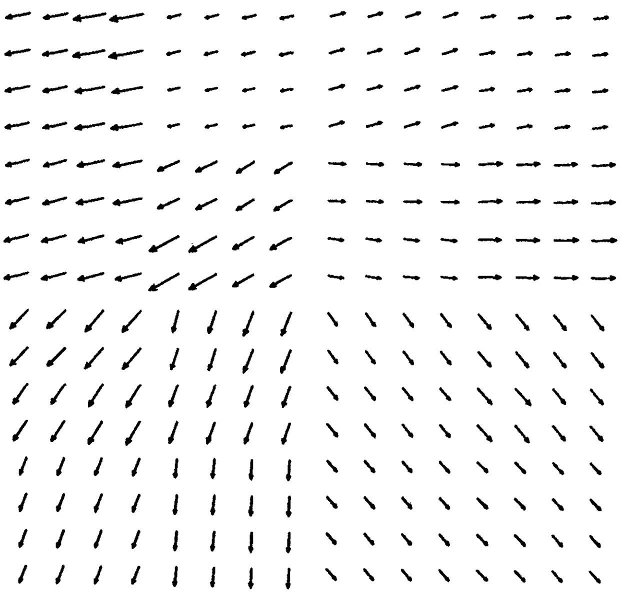

Figure 22: MR algorithm flow estimates: $b = 0.025, \mu = 0.5$.

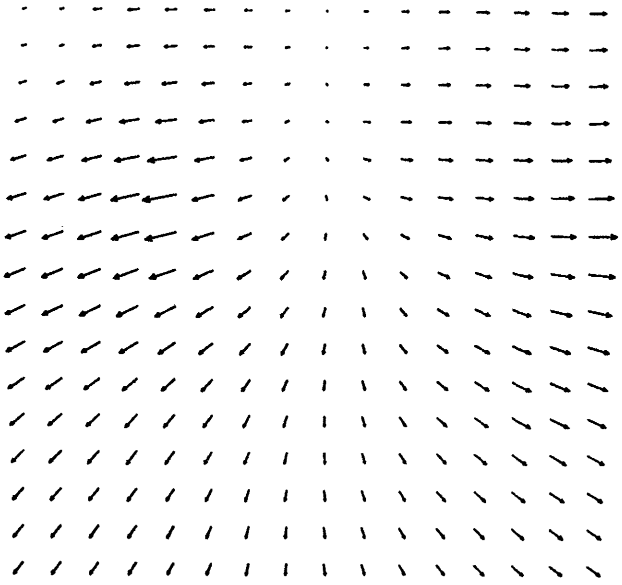

Figure 23: SOR algorithm flow estimates: $R = 500^2$, 800 iterations. The SOR algorithm required a factor of 190 more computation in this example.

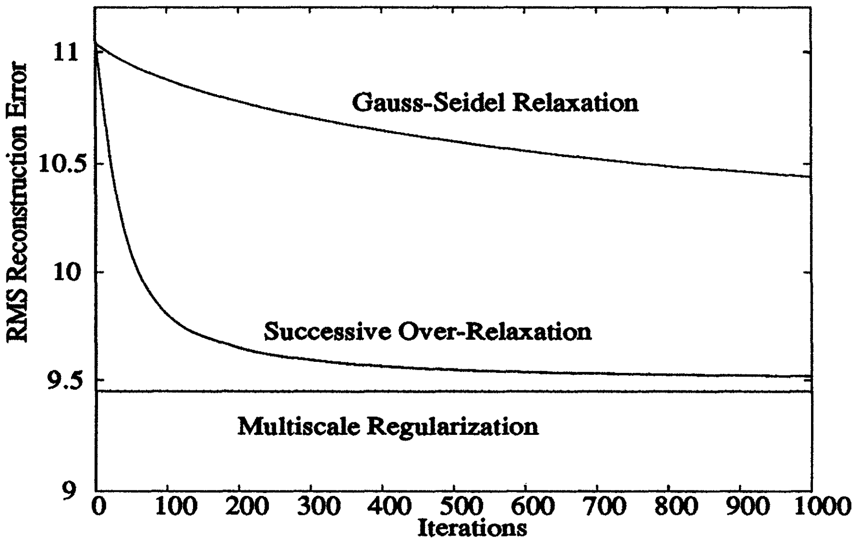

Figure 24: Reconstruction Error Comparison of MR, SOR and Gauss-Seidel (GS) algorithm flow estimates for the Saab sequence.

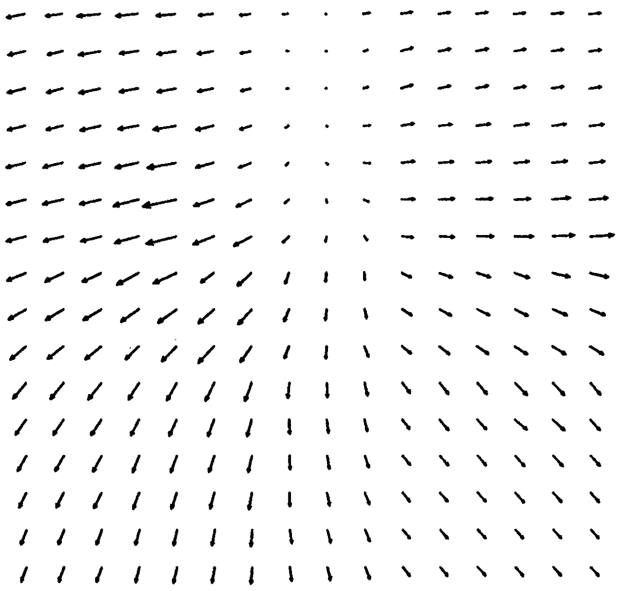

Figure 25: Result of post-processing MR estimates with 30 iterations of the SOR algorithm. Total computational effort is equivalent to 34.2 iterations of the SOR algorithm.

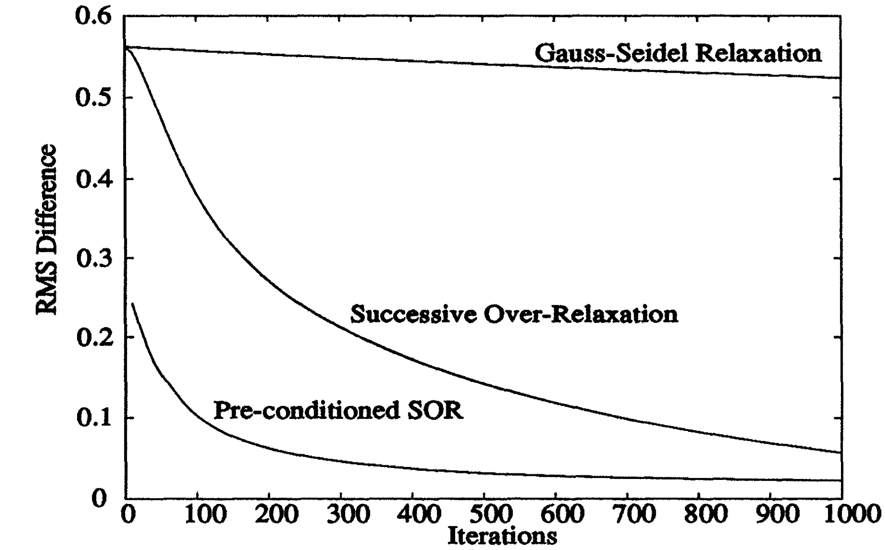

Figure 26: Rms Difference Comparison demonstrates how the MR solution may be used as an initial condition for the iterative algorithms computing a smoothness constraint solution.

constraint to motivate regularization based on a recently introduced class of multiscale stochastic models. The approach has the advantage that the solution of the new problem formulation can be computed very fast, in comparison to the solution of formulations based on a smoothness constraint. In fact, the computational savings realized are on the order of a factor of 10 to 100. Larger gains are associated with larger images since the iterative approaches associated with the smoothness constraint solution take longer to converge as the image grows, whereas the per pixel computation associated with the MR algorithm is independent of image size.

The approach has a number of advantages in addition to the reduction in computational cost. First, multiresolution representations of the flow field are available and, although we have not taken advantage of it in this paper, the MR algorithm also allows for *multiresolution measurements* of the optical flow. Second, error covariance information is available, allowing one to assess the quality of the estimated optical flow. Finally, the MR algorithm is an excellent pre-conditioner for algorithms computing a solution based on a smoothness constraint formulation. In addition, while we have not pursued it here, the multiresolution philosophy introduced here may offer a promising approach to motion-compensated image sequence coding. In particular, although we used the coding metric of reconstruction error as the basis for the comparison of the SC and MR approaches in Section 3.4, the methods presented here would not be the method of choice for that problem. In particular, motion-compensated coding algorithms designed specifically to minimize this criterion [3, 20, 33] will generally outperform the SC and MR approaches (which are not). However, the computationally efficient MR algorithm can be used as an initial preconditioning step for such coding algorithms. In addition, one can also imagine a second way in which MR ideas could be used in this context. In particular, one of the problems with the SC and MR based methods is the differential form of the brightness constraint which, given the discrete nature of spatial and temporal sampling, is only valid for relatively small interframe displacements. In contrast, methods such as [3, 20, 33] use a direct displaced frame matching metric, which is nothing but the integrated version of the brightness constraint. A common approach to dealing with larger displacements with the differential brightness constraint is to spatially blur the image sequences – i.e. to consider lower resolution versions of the image to estimate larger displacements [12, 13]. What this suggests is an MR approach in which we not only have a multiresolution model for optical flow *but also multiresolution measurements* – i.e. measurements as in (27) but for triples $s = (m, i, j)$ at *several* scales. The development of such an approach remains for the future.

Finally, in this paper we have focused on a particular image processing problem, the computation of optical flow. However, we believe that the multiscale stochastic modeling approach can be more generally useful. In particular, it may provide a computationally attractive alternative to standard approaches to the broad class of estimation problems in which the underlying field to be estimated is modeled as a Gaussian Markov random field or as the solution of noise driven partial differential equations, or in which a "smoothness constraint" type regularization is employed. Viewing the multiscale models as an alternative underlying model should lead to significant computational savings for such problems.

Acknowledgment*: The authors gratefully acknowledge the contributions of Dr. Albert Benveniste who, among other things, first suggested that the problem of computing optical flow would be well suited to this approach.*

A Non-homogeneous Tree Structures

We made the assumption at the beginning of Section 2.1 that the image lattice is square, and that the number of rows is equal to a power of two. The reason we have done this is because of the fact that the multiscale model described above is defined on a quadtree structure. There are at least two ways to relax the assumption. First, we could simply zero pad the image lattice to make it fit the quadtree structure. In some sense, we are changing the image lattice to fit the modeling structure. A second, slightly more elegant approach, would be to change the modeling structure to accommodate the lattice. In particular, we would like to have a structure which gives us the proper number of nodes on the finest level. The quadtree structure is homogeneous in the sense that each parent has four offspring; what we are proposing are non-homogeneous tree structures in which different parents may have different numbers of offspring. For example, suppose one had a 6 × 9 lattice. Figure 27 illustrates a sequence of grids that one might use to model accommodate this lattice. In the first level, the root node has six offspring, two in the row direction and three in the column direction. At the second level, each node has nine offspring, three in the row direction and three in the column direction. Thus, at the finest level there is a 6 × 9 lattice. This example illustrates only one simple suggestion. More complicated tree structures could be derived, and certainly the idea could be combined with zero padding.

B The Gauss-Seidel and Successive Over-Relaxation Algorithms

One can show using the calculus of variations that the solution to (4) satisfies the following set of coupled partial differential equations [14]:

$$\begin{aligned} \nabla^2 x_1 &= RE_{z_1}(E_t + \nabla E \cdot x) && (64) \\ \nabla^2 x_2 &= RE_{z_2}(E_t + \nabla E \cdot x) && (65) \end{aligned}$$

where:

$$\begin{aligned} E_{z_1} &= \frac{\partial}{\partial z_1} E(z_1, z_2, t) && (66) \\ E_{z_2} &= \frac{\partial}{\partial z_2} E(z_1, z_2, t) && (67) \\ E_t &= \frac{\partial}{\partial t} E(z_1, z_2, t) && (68) \\ x &= \begin{bmatrix} \frac{\partial z_1}{\partial t} & \frac{\partial z_2}{\partial t} \end{bmatrix}^T && (69) \end{aligned}$$

and where ∇^2 is the Laplacian operator, x_1 and x_2 are the first and second components of the vector x, and R is the parameter controlling the tradeoff between the smoothness and data dependent terms in (4). Denote $x(i,j) \equiv x(ih, jh, t)$ where h is the grid spacing. Discretizing (64), (65) on a uniform grid $\{(i,j) | i \in \{1, ..., Z_1\}, j \in \{1, ..., Z_2\}\}$ leads to the following equations at each point:

$$\bar{x}_1 - 4x_{1,i,j} = h^2 RE_{z_1}(E_{z_1} x_{1,i,j} + E_{z_2} x_{2,i,j} + E_t) \qquad (70)$$

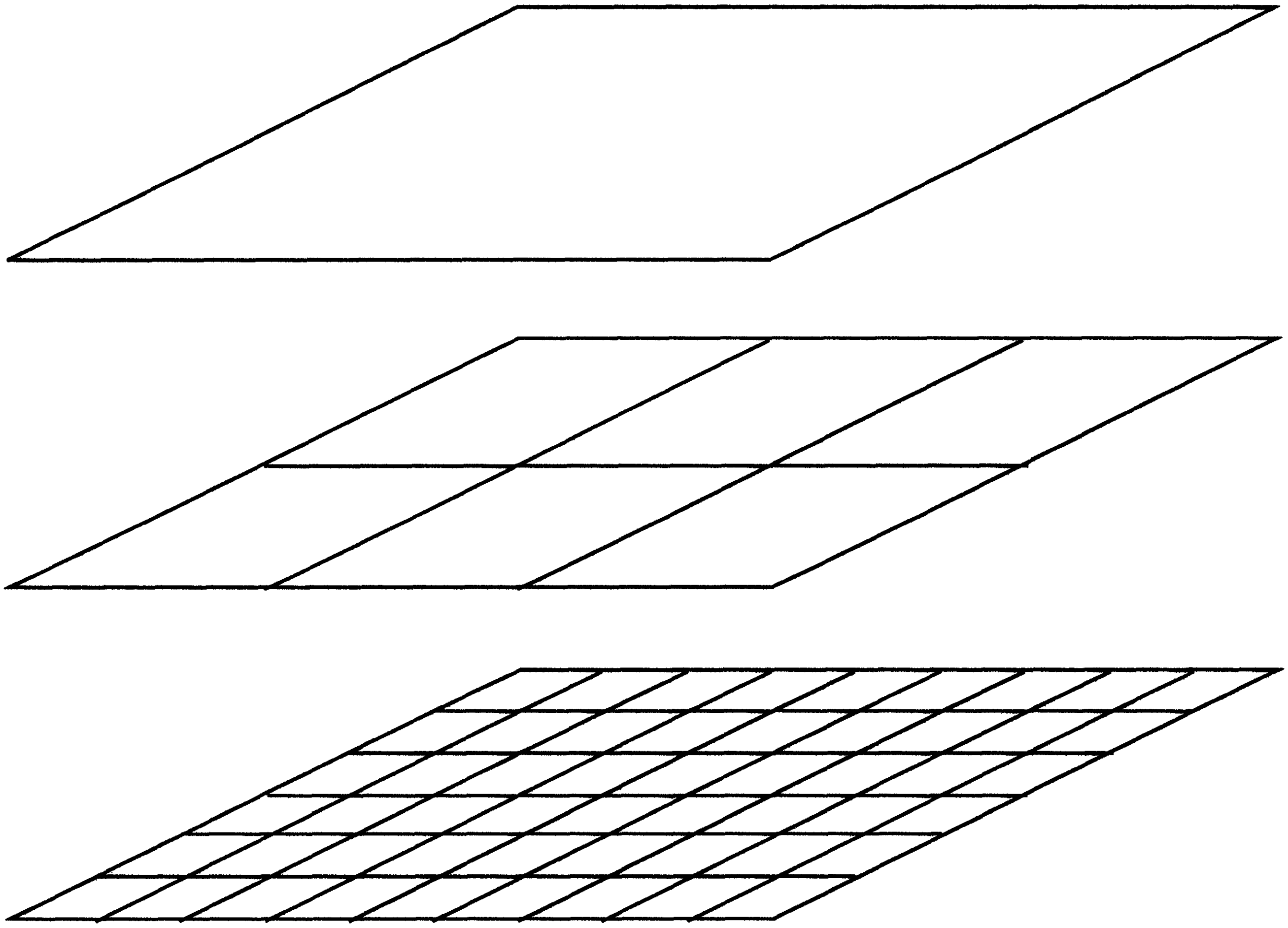

Figure 27: Non-homogeneous tree structure for lattices which are not square. The grid structure is a simple extension of the quadtree structure in that it allows for varying numbers of "offspring" from each parent. The figure illustrates a hierarchy of grids for a 6×9 lattice.

$$\bar{x}_2 - 4x_{2,i,j} = h^2 R E_{z_2}(E_{z_1} x_{1,i,j} + E_{z_2} x_{2,i,j} + E_t) \tag{71}$$

where:

$$\begin{bmatrix} x_{1,i,j} \\ x_{2,i,j} \end{bmatrix} \equiv x(i,j) \tag{72}$$

$$\bar{x}_1 = x_{1,i-1,j} + x_{1,i+1,j} + x_{1,i,j-1} + x_{1,i,j+1} \tag{73}$$

$$\bar{x}_2 = x_{2,i-1,j} + x_{2,i+1,j} + x_{2,i,j-1} + x_{2,i,j+1} \tag{74}$$

The GS and SOR algorithms separate the image grid into two sets of points. These are generally referred to as the *Red points* ($i + j$ is even) and the *Black points* ($i + j$ is odd). The Gauss-Seidel iterations can be derived by solving (70) and (71) for $x_{1,i,j}$ and $x_{2,i,j}$:

GS Red Points

$$x^{n+1}_{1,i,j} = (\bar{x}^n_1 - h^2 R E_{z_1}(E_{z_2} x^n_{2,i,j} + E_t))/d_{1,i,j} \tag{75}$$

$$x^{n+1}_{2,i,j} = (\bar{x}^n_2 - h^2 R E_{z_2}(E_{z_1} x^{n+1}_{1,i,j} + E_t))/d_{2,i,j} \tag{76}$$

GS Black Points

$$x^{n+1}_{1,i,j} = (\bar{x}^{n+1}_1 - h^2 R E_{z_1}(E_{z_2} x^{n+1}_{2,i,j} + E_t))/d_{1,i,j} \tag{77}$$

$$x^{n+1}_{2,i,j} = (\bar{x}^{n+1}_2 - h^2 R E_{z_2}(E_{z_1} x^{n+1}_{1,i,j} + E_t))/d_{2,i,j} \tag{78}$$

where:

$$d_{1,i,j} = 4 + h^2 R E_{z_1} \tag{79}$$

$$d_{2,i,j} = 4 + h^2 R E_{z_2} \tag{80}$$

The SOR algorithm is very similar to the GS algorithm, except that certain *relaxation parameters* are introduced to increase the convergence rate. The SOR iterations are given by:

SOR Red Points

$$x^{n+1}_{1,i,j} = (1 - w_{1,i,j})x^n_{1,i,j} + w_{1,i,j}(\bar{x}^n_1 - h^2 R E_{z_1}(E_{z_2} x^n_{2,i,j} + E_t))/d_{1,i,j} \tag{81}$$

$$x^{n+1}_{2,i,j} = (1 - w_{2,i,j})x^n_{2,i,j} + w_{2,i,j}(\bar{x}^n_2 - h^2 R E_{z_2}(E_{z_1} x^{n+1}_{1,i,j} + E_t))/d_{2,i,j} \tag{82}$$

SOR Black Points

$$x^{n+1}_{1,i,j} = (1 - w_{1,i,j})x^n_{1,i,j} + w_{1,i,j}(\bar{x}^{n+1}_1 - h^2 R E_{z_1}(E_{z_2} x^{n+1}_{2,i,j} + E_t))/d_{1,i,j} \tag{83}$$

$$x^{n+1}_{2,i,j} = (1 - w_{2,i,j})x^n_{2,i,j} + w_{2,i,j}(\bar{x}^{n+1}_2 - h^2 R E_{z_2}(E_{z_1} x^{n+1}_{1,i,j} + E_t))/d_{2,i,j} \tag{84}$$

where:

$$
\begin{aligned}
w_{1,i,j} &= \frac{2}{1+(1-\rho_{1,i,j}^2)^{\frac{1}{2}}} && (85)\\
w_{2,i,j} &= \frac{2}{1+(1-\rho_{2,i,j}^2)^{\frac{1}{2}}} && (86)\\
\rho_{1,i,j} &= \frac{2}{d_{1,i,j}}(\cos\frac{\pi}{Z_1+1}+\cos\frac{\pi}{Z_2+1}) && (87)\\
\rho_{2,i,j} &= \frac{2}{d_{2,i,j}}(\cos\frac{\pi}{Z_1+1}+\cos\frac{\pi}{Z_2+1}) && (88)
\end{aligned}
$$

From (75), we see that each GS iteration requires 10 adds and 4 multiplies per pixel per iteration. From (81), we see that each SOR iteration requires 12 adds and 6 multiplies per pixel per iteration[10]. Thus, GS and SOR require 14 and 18 flops per pixel per iteration respectively.

C MR Algorithm Complexity Analysis

In this section we analyze the computational complexity of the MR algorithm. The analysis applies to the specific model given by (56) — (60). The model is repeated here for convenience:

$$
\begin{aligned}
x(s) &= x(s\bar{\gamma})+(b4^{\frac{-\mu m(s)}{2}})w(s) && (89)\\
y(s) &= C(s)x(s)+v(s) && (90)\\
w(s) &\sim \mathcal{N}(0,I) && (91)\\
v(s) &\sim \mathcal{N}(0,R(s)) && (92)\\
x_0 &\sim \mathcal{N}(0,pI) && (93)
\end{aligned}
$$

where $R(s) = \max(||C(s)||, 10)$. The analysis below takes into account all floating point adds, multiplies and divides.

Consider first the update step given by (42) - (45). $P(s|s+)$ is initialized with pI. Computation of $V^{-1}(s)$ requires 6 floating point operations (the inverse requires 1 divide since $V(s)$ is a scalar and the comparison required to compute $R(s)$ is not counted). Computation of $K(s)$ requires 3 flops. Computation of $P(s|s)$ requires 7 flops (Perform the $C(s)P(s|s+)$ first, and use the fact that $P(s|s)$ must be symmetric). Initialize $\hat{x}(s|s+)$ with zero. Computation of $\hat{x}(s|s)$ then requires 2 flops. The update step is required only at the finest level, since this is the only place we have data for in the optical flow problem. Thus, the total computation associated with this step is 18×4^l flops (l is defined to be the number of levels in the quadtree. There are 4^l points at the finest level.)

Next, consider the prediction step, (46) - (48). Computation of $Q(s\alpha_i)$ is negligible because this parameter varies only as a function of scale (level). Computation of $P(s|s\alpha_i)$

[10] All additions and multiplications which are redundant from iteration to iteration have been ignored. For instance, $1 - w_{1,i,j}$ does not count as an add in (81) since one could compute this once at the start instead of every iteration.

requires 5 flops (note that $F(s)$ and $Q(s\alpha_i)$ are diagonal multiples of the identity). Computation of the predicted estimate $\hat{x}(s|s\alpha_i)$ requires 2 flops. These computations must be done at levels 1 through l. Thus, the total computation associated with this step is approximately $7 \times 4/3 \times 4^l$ flops.

Next, consider the merge step, (49) – (50). Computation of $P(s|s+)$ requires 44 flops (there are five 2×2 inverses requiring 6 flops apiece, and the computation of $(1-q)P_{x(s)}^{-1}$ is negligible since it only varies with scale. The inverses require only 6 flops because the matrices involved are 2×2 and symmetric.) Computation of $\hat{x}(s|s+)$ requires 36 flops. The merge step must be done at levels 0 through $l-1$. Thus, the total computation associated with this step is $80 \times 1/3 \times 4^l$ flops.

Finally, consider the steps in the downward sweep, (53) — (55). Computation of $J(s)$ requires 12 flops (the matrix $P(s\bar{\gamma}|s)$ has already been inverted in (50), $F(s)$ is a multiple of the identity and $J(s)$ is symmetric.) Computation of $P_s(s)$ is not required, unless one is explicitly interested in the error covariance of the smoothed estimate. Computation of $\hat{x}_s(s)$ requires 10 flops. The smoothing step must be done at levels 1 through l. Thus, the total computation associated with this step is 22×4^l flops.

We can now add up all of the computations associated with the MR algorithm. There are 4^l pixels in the problem domain, and thus the algorithm requires $18 + 28/3 + 80/3 + 22 = 76$ flops per pixel. We note that this is a lower bound on the number of flops per pixel in any implementation of the algorithm and that the implementation with the lowest number of flops per pixel may not be the best. The reason is simply that there may not be enough memory available to keep all intermediate calculations around (such as the inverses computed in (50) and reused in (55)). We compute the complexity of the GS and SOR algorithms in the same way (i.e. all intermediate results are assumed to be available), and thus the computational comparison we make between these algorithms is based on optimal (in terms of the number of flops) implementations. Suboptimal implementation of the MR algorithm will lower its computational advantage, but any reasonable implementation (for instance one which saves just $\hat{x}(s|s)$, $P(s|s)$ and the measurement data) will still provide a significant savings over the SOR and GS algorithms.

References

[1] J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images – a review." Proc. IEEE, 76:917-935, 1988.

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

[3] N. Baaziz and C. Labit, "Multigrid motion estimation on pyramidal representations for image sequence coding," IRISA internal publication No. 572, February 1991.

[4] M. Bertero, T. Poggio and V. Torre, "Ill-posed problems in early vision," Proc. IEEE, 76:869-889, 1988.

[5] P. Burt and E. Adelson, "The Laplacian Pyramid as a compact image code," *IEEE Trans. Comm.*, **31**:482-540, 1983.

[6] T. Chin, "Dynamic Estimation in Computational Vision," MIT Dept. of EECS Ph.D. Thesis, Oct. 1991.

[7] K. C. Chou, A. S. Willsky, A. Benveniste and M. Basseville, "Recursive and Iterative estimation algorithms for multiresolution stochastic processes," Proc. of the IEEE CDC, Dec. 1989.

[8] K. C. Chou, *A stochastic modeling approach to multiscale signal processing*, MIT Dept. of EECS Ph.D. Thesis, May 1991.

[9] K. C. Chou, A. S. Willsky and A. Benveniste, "Multiscale Recursive Estimation, Data Fusion and Regularization," MIT LIDS Report # 2085, submitted for publication to IEEE Trans. on Automatic Control.

[10] K. C. Chou, A. S. Willsky and R. Nikoukhah, "Multiscale Systems, Kalman Filters and Riccati Equations," submitted for publication to IEEE Trans. on Automatic Control.

[11] S. C. Clippingdale and R. G. Wilson, "Least Squares Image Estimation on a Multiresolution Pyramid," Proceedings of the 1989 International Conference on Acoustics, Speech and Signal Processing.

[12] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," Computer Vision, Graphics and Image Proc., 43:150-177, 1988.

[13] Heitz, F. and Bouthemy, P., *Multimodal estimation of discontinuous optical flow using Markov random fields.* INRIA Report 1367, Jan. 1991, also submitted to IEEE Trans. on Pattern Analysis and Machine Intelligence.

[14] B. K. P. Horn and B. Schunck, "Determining Optical Flow," Artificial Intelligence, 17:185-203, 1981.

[15] J. KEARNEY, W. THOMPSON AND D. BOLEY, "Optical Flow Estimation: An Error Analysis of Gradient Based Methods with Local Optimization," *IEEE Trans. PAMI*, **2**:229-244, 1987.

[16] C.-C J. KUO, B. C. LEVY AND B. R. MUSICUS, "A local relaxation method for solving elliptic PDE's on mesh connected arrays," SIAM J. Sci. Stat. Comput., 8:550-573, 1987.

[17] S. MALLAT, "Multi-frequency Channel Decomposition of Images and Wavelet Models," *IEEE Trans. ASSP*, **37**:2091-2110, 1989.

[18] B. MANDELBROT AND H. VAN NESS, "Fractional Brownian Motions, Fractional Noises and Applications," SIAM Review, 10:422-436, Oct. 1968.

[19] D. MURRAY AND B. BUXTON, "Scene Segmentation From Visual Motion Using Global Optimization," IEEE PAMI 2:220-228, 1987.

[20] A. NETRAVALI AND J. ROBBINS, "Motion-Compensated Television Coding: Part 1," Bell System Technical Journal 58:631 - 670, 1979.

[21] J. PRINCE AND E. MCVEIGH, "Motion estimation from tagged MR image sequences." Johns Hopkins University Dept. of Elec. and Computer Engineering Technical Report JHU/ECE 91-05, also submitted to IEEE Transactions on Medical Imaging, 1991.

[22] D. RAVIV, "A Quantitative Approach to Camera Fixation," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Maui, Hawaii, 1991.

[23] A. ROUGEE, B. LEVY AND A. S. WILLSKY, "An estimation-based approach to the reconstruction of optical flow," Laboratory for Information and Decision Systems Technical Report, No. 1663, MIT, April 1987.

[24] A. ROUGEE, B. LEVY AND A. S. WILLSKY, "Reconstruction of two dimensional velocity fields as a linear estimation problem," in Proceedings 1st Int. Conf. Computer Vision (London, England) pp. 646-650, June 1987.

[25] A. SHIO AND J. SKLANSKY, "Segmentation of People in Motion," in Proc. IEEE Workshop on Visual Motion, Princeton NJ, Oct. 7-9, 1991, pp. 325 - 332.

[26] E. SIMONCELLI AND E. ADELSON AND D. HEEGER, "Probability Distributions of Optical Flow," IEEE Conference on Computer Vision and Pattern Recognition, Maui Hawaii, June 1991.

[27] G. STRANG, *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley MA, 1986.

[28] G. STRANG, *Linear Algebra and its Applications.* Harcourt-Brace-Jovanovich, 1988.

[29] V. SUNDARESWAREN, "Egomotion from Global Flow Field Data," in Proc. IEEE Workshop on Visual Motion, Princeton NJ, Oct. 7-9, 1991, pp. 140-5.

[30] R. Szeliski, *Bayesian Modeling of Uncertainty in Low-level Vision.* Kluwer Academic, 1989.

[31] D. Terzopoulos, "Image analysis using multigrid relaxation methods," IEEE Trans. on PAMI, 8:129-139, 1986.

[32] H. L. VanTrees, *Detection, Estimation and Modulation Theory: Part 1.* Wiley, New York, 1968.

[33] D. Walker and K. Rao, "Improved pel-recursive motion compensation," IEEE Trans. on Communications, 32:1128-1134, 1984.

[34] G. Wornell, "A Karhunen-Loeve like expansion for 1/f processes," IEEE Trans. on Inf. Theory, 36:859-861, 1990.