

Recursive Optical Flow Estimation—Adaptive Filtering Approach

M. Elad

Hewlett-Packard Labs—Israel, The Technion City, Haifa 32000, Israel

and

A. Feuer

Electrical Engineering Department, The Technion—Israel Institute of Technology, Haifa 32000, Israel

Received March 27, 1996; accepted April 13, 1998

This paper presents a new approach based on the differential framework proposed by Horn and Schunck, to the problem of recursive optical flow estimation from image sequences. The original method of Horn and Schunck is applicable only to the problem of estimating the optical flow between a pair of images from an image sequence. When we aim at estimating the optical flow for long image sequences recursively, the question is whether and how can we gain from previous estimates. In this paper we show that gain is achieved from both computational and accuracy points of view. Incorporation of the time axis into the estimation process is done by assuming temporal smoothness of the optical flow, resulting in simplified spatial–temporal models. The obtained models permit incorporation of the constrained weighted least squares (CWLS) estimator. This estimator is shown to yield RLS and LMS adaptive filter versions for recursive optical flow estimation in time. An interesting and desirable property of the proposed estimation algorithms is their flexibility with respect to performance versus computational requirements. By a simple choice of a parameter these algorithms can be modified to exploit the available time to improve their performance with respect to estimation error. The convergence properties of these estimation algorithms are analyzed. Simulations for various image sequences support the analysis and demonstrate the performance of the estimation algorithms. © 1998 Academic Press

1. INTRODUCTION

Optical flow is the displacement field related to each of the pixels in an image sequence. Such a displacement field results from the apparent motion of the image brightness in time. Estimating the optical flow is a fundamental problem in low-level vision and can undoubtedly serve many applications in image sequence processing. There are many different methods of estimating the optical flow [1–9, 14, 18–27]. Roughly speaking, these methods can be divided into correlation, energy, phase, and differential-based

methods [1]. This paper focuses on a generalization of the method proposed by Horn and Schunck for the estimation of optical flow, which is a differential-based method [2].

The differential framework methods start with a brightness constraint equation which forms a single linear equation per each pixel, constraining its motion vector [1, 2]. Such linear constraints are posed over all the pixels in the current image, forming an ill-posed estimation problem. The various differential-based methods thus vary in the way they add constraints in order to ensure a single and accurate solution to the estimation problem. For example, Lucas and Kanade [3] assume that the optical flow is locally constant, thus making possible the construction of a weighted combination of several constraints, assuming they have the same solution. The Horn and Schunck approach [2] was a regularization based on the assumption of spatial smoothness over the optical flow field.

In a comparison between different optical flow estimation algorithms made by Barron [4], it was found that the performance of the Horn and Schunck algorithm is inferior to other differential methods. However, for image sequences showing distant static objects filmed with camera motion, the spatial smoothness assumption is highly valid and the algorithm of Horn and Schunck performs quite well. Moreover, adopting the modifications in the way the local gradients are estimated proposed by Barron [4] can also improve the performance of the algorithm, resulting in an attractive method of optical flow estimation. Thus, for many applications the above assumptions about the motion field are quite reasonable [1, 2, 4–8].

Most algorithms for the estimation of optical flow concentrate on estimating the motion field between succeeding images in a sequence, disregarding the estimates obtained for the previous image pair. Among such procedures we can count both the Horn and Schunck [2] and the Lucas and Kanade [3] methods. However, several attempts have al-

ready been made to efficiently include the time axis in the optical flow estimation process. Both Singh [6] and Chin and co-workers [7, 8] proposed the application of a Kalman filter estimate the optical flow sequence in time. In [6], a correlation-based (block-matching) technique is generalized to include spatial and temporal smoothness. The measurements are fused in space and time using a Bayesian interpretation. However, the proposed method is computationally very complex since it includes (in addition to other things) block matching for each pixel at the measurements stage, and a full iterative Gauss–Siedel algorithm (with roughly the same computational complexity as the whole Horn and Schunck algorithm) for the spatial smoothness stage.

Chin and co-workers proposed [7, 8] a state-space model which combines temporal smoothness to the brightness constraint equation. This way they generalized the Horn and Schunck algorithm to adequately treat the time axis. In order to propagate the autocorrelation matrix in time efficiently, a square-root information (SRI) Kalman filter is proposed. The main drawback in their approach is the very high computational complexity of the resulting algorithm. The main problem in this regard is the order of the matrices involved, in spite of the fact that these matrices are sparse.

Fleet and Langley [9] proposed a different approach for the same task, based on the Lucas and Kanade [5] optical flow method, combined with recursive filters in time and space for the computation of the spatial and temporal gradients. Their method uses a recursive IIR filter as part of the temporal smoothing of the images before estimating the optical flow. The proposed smoothing is shown to yield recursive update equations for the estimation system in time. In this way, the temporal smoothness assumption is made implicit, rather than explicit.

Another recursive optical flow estimation approach is proposed by Black [18]. This work proposes the application of robust estimators, combined with “temporal continuity,” in order to obtain temporal coherence. The temporal continuity is obtained by penalizing motion vectors which deviate from their predicted values based on past data. Since robust estimation is applied, and since the overall optimization problem becomes nonconvex, sophisticated algorithms which enable convergence to the global minimum point are required. The proposed algorithm is an incremental version of the graduated nonconvexity (GNC) algorithm [19]. The main benefit of this algorithm is its ability to treat discontinuous optical flow fields because of the penalizing function—the Lorentzian [18, 19]. Its main drawbacks are its complexity and its relatively simplistic temporal smoothness model.

The purpose of this paper is to generalize the Horn and Schunck algorithm to inherently include the time axis, while preserving simplicity and low computation of the algorithms. We begin by combining the temporal smoothness assumption with the Horn and Schunck method, re-

sulting in a new model describing the propagation of the optical flow field in time and space. The new model lends itself easily to the application of the constrained weighted least squares (CWLS) estimator. Exact solution of the minimization problem yields the Pseudo-RLS algorithm [10], which roughly require the same amount of computations as the original Horn and Schunck algorithm, but applies temporal smoothness as well. Two simplified methods based on the LMS [10] algorithm are proposed, both with dramatic savings in the computational complexity. The resulting algorithms share simplicity, low computational cost, and accurate estimation performance. We prove that these algorithms converge to the true optical flow in time.

Before we turn to present our detailed approach, we acknowledge the vast existing literature on optical flow and motion estimation algorithms. However, in our list of references we have chosen just a sample of these results. Statistical methods such as the Markov random fields (MRF) and Bayesian based optical flow estimation algorithms [20, 21], multiresolution and multistage motion estimation techniques [14, 22–24], global motion estimation algorithms [25–27], and other approaches are emerging, combining estimation algorithms and applications. Our contribution in this paper is the suggestion of a generalized differential based method in a manner which we believe can be combined with the above trends.

We also note that in this paper we adopt the matrix notations proposed and used by Chin and co-workers [7, 8]. This representation, though “heavy” or complex, enables us to come up with more compact and clear equations. Another benefit of this representation is its relative ease in simulating the proposed algorithms using MATLAB. This, however, does not imply that the matrix approach is the one to adopt at the implementation stage and it should be clear that the complexity of the proposed algorithm is far smaller than that of the Horn and Schunck original algorithm.

This paper is organized as follows: In Section 2 we briefly present the differential framework and the Horn and Schunck algorithm. Section 3 presents the new spatial–temporal optical flow model and recursive algorithms to estimate the optical flow based on the RLS and the LMS adaptive filters. In Section 4 convergence properties of the proposed algorithms are discussed. Simulation results are presented in Section 5, and conclusion in Section 6.

2. HORN AND SCHUNCK FRAMEWORK FOR OPTICAL FLOW ESTIMATION

In this section we briefly present the differential framework [1, 2] and Horn and Schunck method for optical flow estimation [2]. The presentation will be done using matrix–vector notation, similarly to the way it is done in

[7, 8]. This notation will serve to simplify the analysis in the following sections. Further details on the presented optical flow estimation algorithm can be found in [1, 2, 7, 8].

The image sequence brightness is denoted by $I(x, y, t)$, where (x, y, t) represent the spatial and temporal location. The brightness constraint equation is thus

$$I(x, y, t) = I(x - dx(x, y, t), y - dy(x, y, t), t - 1) \quad (2.1)$$

where $[dx(x, y, t), dy(x, y, t)]$ is the motion vector which corresponds to the pixel positioned at (x, y, t) . Note that the temporal sampling rate is assumed to be 1 sample/s. Expanding the right term in the above equation using Taylor series and neglecting higher derivative terms, we get

$$\begin{aligned} I(x, y, t) &= I(x, y, t) - dx(x, y, t) \frac{\partial I(x, y, t)}{\partial x} \\ &\quad - dy(x, y, t) \frac{\partial I(x, y, t)}{\partial y} - \frac{\partial I(x, y, t)}{\partial t} \\ \Rightarrow 0 &= dx(x, y, t) \frac{\partial I(x, y, t)}{\partial x} \\ &\quad + dy(x, y, t) \frac{\partial I(x, y, t)}{\partial y} + \frac{\partial I(x, y, t)}{\partial t}. \end{aligned} \quad (2.2)$$

The above equation connects the local spatial and temporal gradients (assumed known) to the motion vector. Thus we have a single linear equation per pixel, posing a constraint over the motion field. Combining all those equations is possible by matrix notation. We define the following matrix and vectors:

$$\mathbf{Y}(t) = - \left[\frac{\partial I(1, 1, t)}{\partial t} \dots \frac{\partial I(x, y, t)}{\partial t} \dots \frac{\partial I(N, N, t)}{\partial t} \right]^T \quad (2.3)$$

$$H(t) = \left[\begin{array}{cccc} \frac{\partial I(1, 1, t)}{\partial x} & & & \\ & \dots & & \circ \\ & & \frac{\partial I(x, y, t)}{\partial x} & \\ \circ & & & \dots \\ & & & \frac{\partial I(N, N, t)}{\partial x} \end{array} \middle| \begin{array}{ccc} \frac{\partial I(1, 1, t)}{\partial y} & & \\ & \dots & \circ \\ & & \frac{\partial I(x, y, t)}{\partial y} \\ \circ & & \dots \\ & & & \frac{\partial I(N, N, t)}{\partial y} \end{array} \right] \quad (2.4)$$

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \end{bmatrix} \in \mathbb{R}^{2N^2} \quad \text{where} \quad \begin{aligned} \mathbf{D}_x^T &= [dx(1, 1, t) \dots dx(x, y, t) \dots dx(N, N, t)] \\ \mathbf{D}_y^T &= [dy(1, 1, t) \dots dy(x, y, t) \dots dy(N, N, t)] \end{aligned} \quad (2.5)$$

($[N \times N]$ is the number of pixels in each image) then we have the model equation

$$\mathbf{Y}(t) = H(t)\mathbf{X}(t) + \mathbf{E}(t) \quad E\{\mathbf{E}(t)\mathbf{E}^T(t)\} = \sigma_e^2(t)I, \quad (2.6)$$

where $\mathbf{X}(t)$ is the optical flow vector that should be estimated, and $\mathbf{E}(t)$ is the model error which comes from several sources: the fact that we neglect higher derivatives, the fact that we need to compute the local gradients from sampled signals, and the fact that the model is inaccurate for actual scenes for various reasons. The matrix $H(t)$ and the vector $\mathbf{Y}(t)$ can be both calculated from the image pair $I(x, y, t), I(x, y, t - 1)$ by discrete derivatives [1, 2, 7, 8].

Horn and Schunck proposed that additional spatial smoothness constraint should be combined in order to assure single solution and regularized problem. Letter, S denote a certain differentiation matrix (Horn and Schunck

proposed the Laplacian), the smoothness of $\mathbf{X}(t)$ can be measured by

$$\|\mathbf{S}\mathbf{D}_x\|_2^2 + \|\mathbf{S}\mathbf{D}_y\|_2^2 = \|\mathbf{S}\mathbf{X}(t)\|_2^2 \quad \text{where} \quad \mathbf{S} = \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix}. \quad (2.7)$$

Horn and Schunck proposed that the optical flow estimate should be the solution of the following quadratic minimization problem, which searches for the best matching optical flow vector while forcing smoothness of the solution,

$$\begin{aligned} \hat{\mathbf{X}}(t) &= \arg \min_{\mathbf{X}(t)} \{\varepsilon^2\} \\ &= \arg \min_{\mathbf{X}(t)} \{\|\mathbf{Y}(t) - H(t)\mathbf{X}(t)\|_2^2 + \beta\|\mathbf{S}\mathbf{X}(t)\|_2^2\}, \end{aligned} \quad (2.8)$$

where β is a parameter that controls the relative smooth-

ness required. The solution to the above minimization problem is simple to obtain because of the quadratic form of the overall error and is given by

$$\hat{\mathbf{X}}(t) = [H^T(t)H(t) + \beta \mathbf{S}^T \mathbf{S}]^{-1} H^T(t) \mathbf{Y}(t), \quad (2.9)$$

where instead of inverting the positive definite matrix shown above, the iterative Gauss–Siedel algorithm was suggested [2]. The above approach could be repeated at each time t in order to estimate the optical flow between pairs of images, disregarding the previous estimates.

3. OPTICAL FLOW ESTIMATION ALONG THE TIME AXIS

As was said before, our aim is to propose a mechanism that will include the time axis in the optical flow estimation process. An implicit assumption in the following is that the properties of Horn and Schunck’s algorithm are acceptable for the case of two images and that all that remains is to attempt to generalize the algorithm for an infinitely long sequence of images. We seek a recursive approach which will enable to estimate the optical flow sequentially based on previous results, rather than in parallel form. We shall use the temporal smoothness assumption explicitly in order to include the time axis in the estimation process. We begin our presentation with the method presented in [7, 8].

3.1. Kalman Filter Approach

The Kalman filter is a very well-known estimator, aimed at giving the minimum mean squared error (MMSE) estimate of the state of a linear system, represented by state-space equations [11]. Thus, in order to use the Kalman filter for the optical flow estimation task, we must start with a model which represents the problem in a state-space form. The unknown optical flow at time t — $\mathbf{X}(t)$ —will serve as the state vector to be estimated. The *temporal smoothness* constraint can be represented by the equation

$$\mathbf{X}(t) = \mathbf{X}(t-1) + \mathbf{N}(t) \quad E\{\mathbf{N}(t)\mathbf{N}^T(t-v)\} = W_N(t)\delta(v) \quad (3.1)$$

which simply says that the change in time in the optical flow is white (in time) vector $\mathbf{N}(t)$. The above equation is the system equation in the state-space equations. Taking Eq. (2.6) and combining the regularization with it gives the measurements equation

$$\begin{aligned} \begin{bmatrix} \mathbf{Y}(t) \\ \mathbf{0} \end{bmatrix} &= \begin{bmatrix} H(t) \\ \mathbf{S} \end{bmatrix} \mathbf{X}(t) + \begin{bmatrix} \mathbf{E}(t) \\ \mathbf{F}(t) \end{bmatrix} \\ E \left\{ \begin{bmatrix} \mathbf{E}(t) \\ \mathbf{F}(t) \end{bmatrix} \begin{bmatrix} \mathbf{E}(t-v) \\ \mathbf{F}(t-v) \end{bmatrix}^T \right\} &= \begin{bmatrix} W_E(t) & 0 \\ 0 & W_F(t) \end{bmatrix} \delta(v), \end{aligned} \quad (3.2)$$

where the *spatial smoothness* serves here as an additional measurements vector of zeros. Having the above two equations permits the direct use of the Kalman filter. However, the dimensions of the matrices involved (though sparse) are very large, hence the direct application of the Kalman filter is impossible. In [7, 8], using this exact model, a square root information (SRI) Kalman filter [11] which propagates the square root of the inverse of the autocorrelation matrix in time is suggested. Yet the computational complexity of the final algorithm is far too high, and only parallel implementation can cope with it effectively.

3.2. Constrained Weighted Least-Squares Approach

Instead of the state-space model presented above, we can suggest the following model, which combines Horn and Schunck equations with the temporal smoothness constraint differently,

$$\begin{aligned} \forall k \geq 0 \quad \begin{bmatrix} \mathbf{Y}(t-k) \\ \mathbf{0} \end{bmatrix} &= \begin{bmatrix} H(t-k) \\ \mathbf{S} \end{bmatrix} \mathbf{X}(t) + \mathbf{E}(t, k) \\ E\{\mathbf{E}(t, k)\mathbf{E}^T(t, j)\} &= W_E(t, k)\delta(k-j) \\ &= \phi^{-1}(t, k) \begin{bmatrix} \sigma_e^2 I & 0 \\ 0 & \sigma_f^2 I \end{bmatrix} \delta(k-j), \end{aligned} \quad (3.3)$$

where

$$\phi(t, k) = \begin{cases} 1 & k = 0 \\ \prod_{i=0}^{k-1} \lambda(t-i) & k \geq 1. \end{cases} \quad (3.4)$$

Note that $\phi(t, k) = \lambda(t)\phi(t-1, k-1)$ for $k \geq 1$.

This model simply states that the optical flow vector $X(t)$ matches the model equations for all casual times $t-k \leq t$, and this way the temporal smoothness is applied. But, since we know that there are changes in the optical flow in time, we allow them by exponentially raising the variance of the model error for far-away model equations, and the parameters $0 \ll \lambda(t) < 1$ act as forgetting factors for this very purpose. Having the new model, we can define a quadratic error:

$$\begin{aligned} \varepsilon^2(t) &= \sum_{k=0}^t \left\| \begin{bmatrix} \mathbf{Y}(t-k) \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} H(t-k) \\ \mathbf{S} \end{bmatrix} \mathbf{X}(t) \right\|_{W_E^{-1}(t,k)}^2 \\ &= \frac{1}{\sigma_e^2} \sum_{k=0}^t \phi(t,k) \{ \|\mathbf{Y}(t-k) - H(t-k)\mathbf{X}(t)\|_2^2 + \beta \|\mathbf{S}\mathbf{X}(t)\|_2^2 \}. \end{aligned} \quad (3.5)$$

This squared error can be shown to emerge from the application of the Maximum Likelihood (ML) estimator [11] with regularization if we assume that all the random processes involved are Gaussian. The new parameter β is the ratio between the variance of the model error and the smoothness error— $\beta = \sigma_e^2/\sigma_f^2$. A different approach to the above error definition can be the use of the maximum a posteriori probability (MAP) estimator [11] on the model given in Eq. (3.3), where again the Gaussianity is assumed and the second equation (the smoothness term) stands for the prior or the auto-regressive (AR) model of the unknown random process $\mathbf{X}(t)$.

Differentiating with respect to the vector $\mathbf{X}(t)$ yields the equations

$$\frac{\partial \varepsilon^2(t)}{\partial \mathbf{X}(t)} = 0 = \frac{2}{\sigma_e^2} [R(t)\mathbf{X}(t) - \mathbf{P}(t)], \quad (3.6)$$

where

$$\begin{aligned} R(t) &= \sum_{k=0}^t \phi(t,k) \{ H^T(t-k)H(t-k) + \beta \mathbf{S}^T \mathbf{S} \} \\ &= \lambda(t)R(t-1) + H^T(t)H(t) + \beta \mathbf{S}^T \mathbf{S} \end{aligned} \quad (3.7)$$

$$\begin{aligned} \mathbf{P}(t) &= \sum_{k=0}^t \phi(t,k) H^T(t-k)\mathbf{Y}(t-k) \\ &= \lambda(t)\mathbf{P}(t-1) + H^T(t)\mathbf{Y}(t). \end{aligned} \quad (3.8)$$

Note that the matrix $R(t)$ is a symmetric positive definite (and nonsingular) matrix because of the regularization term $\beta \mathbf{S}^T \mathbf{S}$.

The obtained recursive equations for $R(t)$ and $\mathbf{P}(t)$ form the basis of the optical flow estimation algorithms that will be presented in the sequel. The choice of exponentially decaying weights in the temporal squared error is the reason behind the simple recursive formulation obtained.

Two interesting points with respect to the recursive formulation obtained should be mentioned here. First, assuming that the optical flow process is correctly modeled by the state equations (3.1), (3.2); obviously, the Kalman filter is the optimal linear estimator. The proposed formulation can be shown to emerge from this Kalman filter by simple approximation. In the next section this point is investigated, showing a connection between the forgetting factors $\lambda(t)$ and the model given in Eqs. (3.1)–(3.2).

Second, although Fleet and Langley [9] proposed a temporal generalization of the Lucas and Kanade [4] optical flow estimation method, their final result is also a one-pole filtering, similar to the one proposed here. Thus, the presented model and results are parallel to the method proposed by Fleet and Langley for the generalization of the Horn and Schunck [2] algorithm. However, both in [7 and 8] and in [9], the temporal smoothness is utilized only to improve the accuracy of the estimated optical flow. Here, as we show in the remainder of the paper, the new recursive formulation can be further simplified resulting in algorithms with reduced computation requirements, with assured convergence and acceptable performance. This way we gain both in accuracy and in computational efficiency.

3.3. The Pseudo-RLS Optical Flow Estimation Algorithm

One way to solve the minimization problem in Eq. (3.5) is direct solution of the linear system in Eq. (3.6). In order to do that, we have to first update $R(t)$ and $\mathbf{P}(t)$ in time according to Eqs. (3.7) and (3.8) and then solve the linear system in Eq. (3.6). The matrix $R(t)$ is sparse (see Appendix A for a proof and discussion of this property) and can easily be updated (in time) and stored. If the number of pixels in the image sequence is N^2 then the number of unknowns in the vector $\mathbf{X}(t)$ is $2N^2$ and the size of the matrix $R(t)$ is $[2N^2 \times 2N^2]$. Since N is typically large, this means that a direct inversion of $R(t)$ is impossible and indirect methods are required in order to solve (3.6). Many iterative algorithms such as the steepest decent (SD) [12], conjugate gradient (CG) [12], and error relaxation methods (such as Jacoby, Gauss–Siedel, successive over-relaxation) [13] and the multigrid method [14] can be suggested. The underlying principle common to all these iterative algorithms is their ability to solve Eq. (3.6) using only matrix–vector multiplications which are easily performed for sparse matrices.

The reason we call such procedures pseudo-RLS algorithms comes from the fact that we update the matrix $R(t)$ and the vector $\mathbf{P}(t)$ recursively, as can be done in the recursive least squares (RLS) algorithm [9]. However, in contrast to classic RLS, we do not propagate nor compute the matrix $Q(t) = R^{-1}(t)$ in time. In our case this matrix is not even sparse, which means, again, intolerable computational and storage demands.

Assuming that we applied one of the iterative algorithms successfully, with a sufficient number of iterations, the estimated optical flow will converge to

$$\hat{\mathbf{X}}(t) = R^{-1}(t)\mathbf{P}(t), \quad (3.9)$$

which we consider an optimal result since this vector minimizes the squared error defined in Eq. (3.5). The number

of computations required is similar to the number required by the original Horn and Schunck algorithm. This is because we have to compute the update terms for the matrix $R(t)$ and the vector $\mathbf{P}(t)$, which are exactly the terms computed for the Horn and Schunck algorithm; we have to add them to $R(t)$ and $\mathbf{P}(t)$, which requires only additions; and then we have to apply an iterative algorithm similar to what is done in the original Horn and Schunck algorithm. There can be a considerable saving in computations if we use the previous result as an initialization for the iterative procedure, but it is difficult to quantify the saving in computations if this initialization is used.

The pseudo-RLS algorithm requires the propagation of the matrix $R(t)$ in time as part of the estimation process. The practicality of the pseudo-RLS lays heavily on the assumption that $R(t)$ is highly sparse for all times $t > 0$, since $R(t)$ dimensions are huge. Appendix A discusses the structure of the matrix sequence $R(t)$ in time and shows that this sequence of matrices is indeed sparse for all $t > 0$.

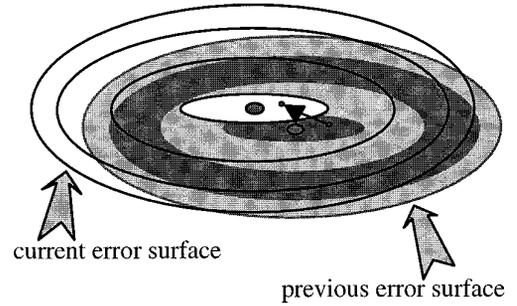
3.4. The LMS Optical Flow Estimation Algorithms

A different approach that can be taken in order to minimize the temporal squared error in Eq. (3.5) is suggested by the least mean squares (LMS) algorithm [10]. First, instead of a full minimization of this error at each time instant, we can simply take the previous result $\hat{\mathbf{X}}(t-1)$ and update it using the instantaneous gradient of the temporal squared error and get the following recursive equation:

$$\begin{aligned} \hat{\mathbf{X}}(t) &= \hat{\mathbf{X}}(t-1) - \frac{\mu}{2} \frac{\partial \varepsilon^2(t)}{\partial \mathbf{X}(t)} \Big|_{\hat{\mathbf{X}}(t-1)} \\ &= \hat{\mathbf{X}}(t-1) + \mu[\mathbf{P}(t) - R(t)\hat{\mathbf{X}}(t-1)]. \end{aligned} \quad (3.10)$$

The main idea behind this approach is the assumption that the quadratic surface representing the squared error $\varepsilon^2(t-1)$ changes only slightly in time, because of the temporal smoothness assumption. This can also be seen from Eq. (3.5), since for $\lambda(t) \rightarrow 1$ (which means high temporal smoothness) the update term is small compared to the accumulated error. Therefore, instead of minimizing $\varepsilon^2(t)$ from the start, all we have to do is to descend on the surface $\varepsilon^2(t)$ from the position $\hat{\mathbf{X}}(t-1)$ to a new position, governed by the local gradient of the surface to be minimized. Scheme 1 illustrates this idea for the 2-D case.

Another view of the obtained recursive equation is that Eq. (3.10) is simply one iteration of the steepest descent algorithm. Thus, instead of performing many iterations at each time instant as was proposed in the Pseudo-RLS algorithm, all we are proposing to do here is a single iteration, and continue to the next temporal point. Therefore, we can suggest also a midway algorithm, namely, at each temporal point, update the matrix and the vector $R(t)$ and



SCHEME 1. The M -LMS algorithm basic idea.

$\mathbf{P}(t)$ as usual using Eqs. (3.7) and (3.8), and then perform M steepest descent iterations. We already know that for $1 \ll M \rightarrow \infty$ we get the pseudo-RLS. We will refer to the above algorithm with M iterations per each time instant as the M -SD algorithm. One interesting property of the M -SD algorithm is its flexibility with regard to the computational requirements. For any optical flow estimation system which applies the M -SD algorithm we can adopt the algorithm to fully exploit the available time, by adding more iterations. The more iterations performed (which means that M is increased) the better is the quality of the estimated optical flow (see the next section).

In the context of the M -SD presented above we proposed only the steepest descent algorithm. The reason is the relative simplicity of the required convergence analysis. However, the SD is known to be inferior to algorithms such as the CG and Gauss-Siedel. An interesting question is whether using other iterative algorithms can improve the convergence rate. This question will not be treated analytically in this paper, but simulation results for this issue will be presented.

Using Eq. (3.10) as our estimation process is an approximation to the pseudo-RLS estimator presented in Eq. (3.5). Using the recursive equations (3.7) and (3.8) in Eq. (3.10), we get

$$\begin{aligned} \hat{\mathbf{X}}(t) &= \hat{\mathbf{X}}(t-1) + \mu\lambda(t)[\mathbf{P}(t-1) - R(t-1)\hat{\mathbf{X}}(t-1)] \\ &\quad + \mu H^T(t)\mathbf{Y}(t) - \mu[H^T(t)H(t) + \beta\mathbf{S}^T\mathbf{S}]\hat{\mathbf{X}}(t-1). \end{aligned} \quad (3.11)$$

Thus, if we assume further that the previous solution $\hat{\mathbf{X}}(t-1)$ is close to the optimal one, then we can say that $\mathbf{P}(t-1) - R(t-1)\hat{\mathbf{X}}(t-1) = 0$, and this term can be omitted from the above equation. The new recursive equation is thus

$$\hat{\mathbf{X}}(t) = \hat{\mathbf{X}}(t-1) + \mu H^T(t)\mathbf{Y}(t) - \mu[H^T(t)H(t) + \beta\mathbf{S}^T\mathbf{S}]\hat{\mathbf{X}}(t-1), \quad (3.12)$$

which is a simpler algorithm with even more reduced computations, since this algorithm no longer require the propa-

gation of $R(t)$ and $\mathbf{P}(t)$ in time. The above equation is a single SD iteration of the original Horn and Schunck algorithm. Interesting as it may seem, Horn and Schunck in their original paper [2] suggested this very algorithm based on intuition only as an alternative to the application of their algorithm with many iterations per step. Following the same reasoning as in the M -SD algorithm, we can propose here that M iterations of Eq. (3.12) can be performed per time step, which might improve the overall performance of the algorithm, when compared to the single iteration algorithm. We choose to call such algorithm the M -LMS algorithm for obvious reasons.

The major questions with regard to the M -SD and the M -LMS algorithms are whether they converge and what their convergence properties are. An analysis of these topics is given in the next section. One more issue to discuss is the choice of μ for the M -SD and M -LMS algorithms. Theoretic bounds on the value of this parameter are derived in the next section. However, one easy and efficient approach to the choice of μ is the application of the following equation, which is taken from the normalized SD algorithm [12, 13]:

$$\mu_k(t) = \frac{\hat{\mathbf{E}}_k^T(t)\hat{\mathbf{E}}_k(t)}{\hat{\mathbf{E}}_k^T(t)R(t)\hat{\mathbf{E}}_k(t)}; \quad \hat{\mathbf{E}}_k(t) \triangleq \mathbf{P}(t) - R(t)\hat{\mathbf{X}}_k(t-1). \quad (3.13)$$

The NSD performs a line search which finds the best possible μ for each iteration separately. Since the NSD converges faster than the SD, such approach is better, but we will not supply a proof for this property in this framework. This approach ofcourse yields a time and iteration dependent value for μ . Similar equations can be written for the M -LMS algorithm with different definitions for $R(t)$ and $\mathbf{P}(t)$ (see Section 4.3 for further details).

3.5. Confidence Measures for the Above Algorithm

Many of the optical flow estimation algorithms supply as byproducts confidence measures coupled with motion vectors, determining the reliability of these motion vectors. It is expected that every motion estimation algorithm will supply estimates whose accuracy varies in space and time. Using these confidence measures to extract nonreliable estimates may serve various applications where there is no need for a motion vector for each pixel. In the optical flow estimation method of Lucas and Kanade [3] an efficient confidence measure for the estimation results is the condition number of a 2×2 matrix to be inverted, or the minimal eigenvalue of this matrix as proposed in [4]. We note that every pixel has a corresponding matrix like this, which then provides a simple confidence measure.

Following the same reasoning, the eigenvalues of the large matrix to be inverted in the Horn and Schunck algo-

rithm could serve as confidence measures somehow, but their computation is impossible because of the dimension on the matrix involved. In [4], it is proposed to threshold the spatial gradient local norm as a tool to measure the reliability of the estimates of the Horn and Schunck algorithm. The reliable estimates are those which satisfy

$$\|\nabla I(x, y, t)\|_2^2 = \left[\frac{\partial I(x, y, t)}{\partial x} \right]^2 + \left[\frac{\partial I(x, y, t)}{\partial y} \right]^2 \geq \text{Thr}, \quad (3.14)$$

where Thr is some threshold value. Naturally, this idea could serve for our algorithm too, since these gradients are available for the pseudo-RLS, M -SD, and M -LMS algorithms. However, Eq. (3.14) does not use knowledge accumulated over time. Instead, we suggest the use of the main diagonal elements of the matrix $R(t)$ as replacements for the eigenvalues. Thresholding these values can serve as confidence measures. Since there is some relation between the main diagonal values of a positive definite matrix and its eigenvalues [16], this choice of confidence measures seems worthwhile.

The proposed confidence measures can be applied in the Pseudo-RLS and the M -LMS algorithms. Since the M -LMS does not propagate the matrix $R(t)$ in time, we can use instead the confidence measures in Eq. (3.14) for this algorithm. Note that the above approach supplies each component of each motion vector with a confidence number. By adding the two confidence measures for the x and y components of each motion vector we get one confidence measure per each pixel. Adding the two components yields a confidence measure which is very similar to the one proposed in Eq. (3.14). If we omit the regularization, the sum of the x and y components of the matrix $H^T(t)H(t)$ is exactly the spatial gradient norm in (3.14). The regularization matrix $\beta\mathbf{S}^T\mathbf{S}$ adds a constant C to each of the gradients, yielding the value

$$f(x, y, t) = \left[\frac{\partial I(x, y, t)}{\partial x} + C \right]^2 + \left[\frac{\partial I(x, y, t)}{\partial y} + C \right]^2, \quad (3.15)$$

which can serve to detect low gradient norms by thresholding. The accumulation in time performed by the matrix $R(t)$ smooths the values of $f(x, y, t)$ over time. It is obvious that having a low local gradient norm at time t does not necessarily mean that the obtained estimate at this location is inaccurate, since we may get reliable information for this location from past history. Therefore, a better confidence measure which takes the time axis into account is the value

$$\begin{aligned} F(x, y, t) &= \sum_{k=0}^t \phi(t, k) f(x, y, t-k) \\ &= \lambda(t)F(x, y, t-1) + f(x, y, t) \end{aligned} \quad (3.16)$$

and this is exactly the value obtained using the main diagonal elements of the matrix $R(t)$, as proposed.

3.6. Higher Order Temporal Smoothness Model

Equation (3.1) represents the temporal smoothness assumption for all the above analysis. This equation is a first order AR model assumed on the optical flow sequence in time. Such model is adequate for sequences where the motion is almost constant, with relatively small variations in time. We can generalize the temporal smoothness model to be a higher order AR model, as proposed in [7, 8]. This way we could present a better temporal smoothness model for slowly varying optical flow sequences. The proposed higher order smoothness model is

$$\begin{aligned} \mathbf{X}(t) &= \sum_{k=1}^{K_0} a_k \mathbf{X}(t-k) + \mathbf{N}(t) \quad E\{\mathbf{N}(t)\mathbf{N}^T(t-v)\} \\ &= W_N(t)\delta(v), \end{aligned} \quad (3.17)$$

where $a_{K_0} \neq 0$ and the AR coefficients should be such that stability of the model is assured [10]. In order to represent this smoothness model as a state-space equation we define a new longer state vector, which gives the following state equations:

$$\begin{aligned} \mathbf{F}(t) &= \begin{bmatrix} \mathbf{X}(t) \\ \mathbf{X}(t-1) \\ \vdots \\ \mathbf{X}(t-K_0) \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{X}(t) \\ \mathbf{X}(t-1) \\ \vdots \\ \mathbf{X}(t-K_0) \end{bmatrix} \\ &= \begin{bmatrix} a_1 I & a_2 I & \cdots & a_{K_0} I \\ I & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(t-1) \\ \mathbf{X}(t-2) \\ \vdots \\ \mathbf{X}(t-K_0-1) \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{N}(t) \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ \Rightarrow \mathbf{F}(t) &= \mathbf{A}\mathbf{F}(t-1) + \tilde{\mathbf{N}}(t) \quad E\{\tilde{\mathbf{N}}(t)\tilde{\mathbf{N}}^T(t-v)\} \\ &= \tilde{W}_N(t)\delta(v) \end{aligned} \quad (3.18)$$

$$\begin{aligned} \begin{bmatrix} \mathbf{Y}(t) \\ \mathbf{0} \end{bmatrix} &= \begin{bmatrix} H(t) & 0 & \cdots & 0 \\ \mathbf{S} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(t) \\ \mathbf{X}(t-1) \\ \vdots \\ \mathbf{X}(t-K_0) \end{bmatrix} + \begin{bmatrix} \mathbf{E}(t) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \\ \Rightarrow \tilde{\mathbf{Y}}(t) &= \tilde{H}(t)\mathbf{F}(t) + \tilde{\mathbf{E}}(t) \quad E\{\tilde{\mathbf{E}}(t)\tilde{\mathbf{E}}^T(t-v)\} = \tilde{W}_E(t)\delta(v) \end{aligned} \quad (3.19)$$

Using the same reasoning as in Section 3.2, instead of the state-space model presented above we can suggest the following model, which combines Horn and Schunck equations with the temporal smoothness constraint differently,

$$\begin{aligned} \forall k \geq 0 \quad \tilde{\mathbf{Y}}(t-k) &= \tilde{H}(t-k)A^{-k}\mathbf{F}(t) + \tilde{\mathbf{E}}(t,k) \\ E\{\tilde{\mathbf{E}}(t,k)\tilde{\mathbf{E}}^T(t,j)\} &= W_{\tilde{\mathbf{E}}}(t,k)\delta(k-j) \\ &= \phi^{-1}(t,k) \begin{bmatrix} \sigma_m^2 I & 0 \\ 0 & \sigma_s^2 I \end{bmatrix} \delta(k-j), \end{aligned} \quad (3.20)$$

which is quite similar to the model in Eq. (3.3). The quadratic error term is then defined as

$$\varepsilon^2(t) = \sum_{k=0}^t \|\tilde{\mathbf{Y}}(t-k) - \tilde{H}(t-k)A^{-k}\mathbf{F}(t)\|_{W_{\tilde{\mathbf{E}}}(t,k)}^2. \quad (3.21)$$

Differentiation with respect to the vector $\mathbf{F}(t)$ yields the equations

$$\frac{\partial \varepsilon^2(t)}{\partial \mathbf{F}(t)} = 0 = \frac{2}{\sigma_m^2} [\tilde{R}(t)\mathbf{F}(t) - \tilde{\mathbf{P}}(t)], \quad (3.22)$$

where

$$\begin{aligned} \tilde{R}(t) &= \sum_{k=0}^t [A^T]^{-k} \tilde{H}^T(t-k) W_{\tilde{\mathbf{E}}}^{-1}(t,k) \tilde{H}(t-k) A^{-k} \sigma_m^2 \\ &= \lambda(t) A^{-T} \tilde{R}(t-1) A^{-1} \\ &\quad + \begin{bmatrix} H^T(t)H(t) + \beta \mathbf{S}^T \mathbf{S} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{2K_0 N^2 \times 2K_0 N^2} \end{aligned} \quad (3.23)$$

$$\begin{aligned} \tilde{\mathbf{P}}(t) &= \sum_{k=0}^t [A^T]^{-k} \tilde{H}^T(t-k) W_{\tilde{\mathbf{E}}}^{-1}(t,k) \tilde{\mathbf{Y}}(t-k) \sigma_m^2 \\ &= \lambda(t) A^{-T} \tilde{\mathbf{P}}(t-1) + \begin{bmatrix} H^T(t)\mathbf{Y}(t) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \end{aligned} \quad (3.24)$$

and again we got two recursive equations which could be used to estimate the state-vector $\mathbf{F}(t)$, or even only its upper part, which corresponds to the new optical flow to be estimated— $\mathbf{X}(t)$. The new defined problem is more complex than the one originated from the first-order AR

smoothness model since the matrix $\tilde{R}(t)$ is of size $[2K_0N^2 \times 2K_0N^2]$, compared to the size of $R(t)$, $[2N^2 \times 2N^2]$.

The development of the Pseudo-RLS, M -SD, and M -LMS algorithms is very similar to the one presented earlier for the first order AR smoothness model. In this framework we shall present only the M -LMS approach for *second order* AR model. The M -LMS algorithm can be obtained from the pseudo-RLS direct approach by two major steps—fixing that $\lambda(t) = 0$, and performing several SD iterations per time point. The equation obtained from the first step is

$$\begin{aligned} \hat{\mathbf{F}}_1(t) &= A\hat{\mathbf{F}}_m(t-1) + \mu\sigma_m^2\{\tilde{H}^T(t)W_{\tilde{E}}^{-1}(t,0)\tilde{\mathbf{Y}}(t) \\ &\quad - \tilde{H}^T(t)W_{\tilde{E}}^{-1}(t,0)\tilde{H}(t)A\hat{\mathbf{F}}_m(t-1)\} \\ &= \begin{bmatrix} \alpha I & (1-\alpha)I \\ I & 0 \end{bmatrix} \hat{\mathbf{F}}_m(t-1) \\ &\quad + \mu \left\{ \begin{bmatrix} H^T(t)\mathbf{Y}(t) \\ \mathbf{0} \end{bmatrix} \right. \\ &\quad \left. - \begin{bmatrix} H^T(t)H(t) + \beta\mathbf{S}^T\mathbf{S} & 0 \\ 0 & 0 \end{bmatrix} \right. \\ &\quad \left. \begin{bmatrix} \alpha I & (1-\alpha)I \\ I & 0 \end{bmatrix} \hat{\mathbf{F}}_m(t-1) \right\} \end{aligned} \quad (3.25)$$

and the equation for the second step (M steepest descent steps) gives the equation

$$\begin{aligned} \hat{\mathbf{F}}_k(t) &= \hat{\mathbf{F}}_{k-1}(t) + \mu\{\tilde{H}^T(t)\tilde{\mathbf{Y}}(t) - \tilde{H}^T(t)\tilde{H}(t)\hat{\mathbf{F}}_{k-1}(t)\} = \hat{\mathbf{F}}_{k-1}(t) \\ &\quad + \mu \left\{ \begin{bmatrix} H^T(t)\mathbf{Y}(t) \\ \mathbf{0} \end{bmatrix} \right. \\ &\quad \left. - \begin{bmatrix} H^T(t)H(t) + \beta\mathbf{S}^T\mathbf{S} & 0 \\ 0 & 0 \end{bmatrix} \hat{\mathbf{F}}_{k-1}(t) \right\} \end{aligned} \quad (3.26)$$

for $k = 2$ to m . Bearing in mind that the state-vector $\mathbf{F}(t)$ contains two optical flow vectors, we can write the above vector–matrix equation as two equations,

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{X}}(t) \\ \hat{\mathbf{X}}(t-1) \end{bmatrix} &= \dots \\ &= \begin{bmatrix} [I - \mu[H^T(t)H(t) + \beta\mathbf{S}^T\mathbf{S}]][\alpha\hat{\mathbf{X}}(t-1) + \\ \quad (1-\alpha)\hat{\mathbf{X}}(t-2)] + \mu H^T(t)\mathbf{Y}(t) \\ \hat{\mathbf{X}}(t-1) \end{bmatrix}, \end{aligned} \quad (3.27)$$

but since we are interested in $\mathbf{X}(t)$ rather than $\mathbf{F}(t)$, and

beyond that, the second equation above is trivial with no contribution to the estimation goals, we can use the first equation alone as a recursive update for $\hat{\mathbf{X}}(t)$:

$$\begin{aligned} \hat{\mathbf{X}}(t) &= [I - \mu[H^T(t)H(t) + \beta\mathbf{S}^T\mathbf{S}]] \\ &\quad \times [\alpha\hat{\mathbf{X}}(t-1) + (1-\alpha)\hat{\mathbf{X}}(t-2)] + \mu H^T(t)\mathbf{Y}(t). \end{aligned} \quad (3.28)$$

As expected, the recursive equation $\hat{\mathbf{X}}(t)$ uses second order history, and performing M iterations of Eq. (3.28) gives the second order M -LMS algorithm. By assuming that $\alpha = 1$, Eq. (3.28) is reduced to the first order AR model M -LMS algorithm as given by Eq. (3.12).

4. CONVERGENCE PROPERTIES OF THE PROPOSED ALGORITHMS

In the previous section we proposed some relatively simple algorithms for the recursive estimation of optical flow for image sequences. In this section we analyze the properties of three of those algorithms. All three are first order AR smoothness model algorithms—the pseudo-RLS, the M -SD, and the M -LMS. The analysis of higher order smoothness models can be carried out using similar approach.

Before we start our analysis let us recall the underlying model assumption for $\mathbf{X}(t)$, the optical flow vector. Rewriting Eqs. (3.1) and (3.2), we have

$$\begin{aligned} \mathbf{X}(t) &= \mathbf{X}(t-1) + \mathbf{N}(t) \\ \mathbf{N}(t) &\Rightarrow \mathbf{G}\mathbf{0}, W_{\mathbf{N}}(t) = \sigma_{\mathbf{N}}^2 I \\ \tilde{\mathbf{Y}}(t) &= C(t)\mathbf{X}(t) + V(t) \\ V(t) &\Rightarrow \mathbf{G} \left\{ \mathbf{0}, W_{\mathbf{V}}(t) = \begin{bmatrix} \sigma_v^2 I & 0 \\ 0 & \sigma_f^2 I \end{bmatrix} \right\}, \end{aligned} \quad (4.1.1)$$

where

$$\tilde{\mathbf{Y}}(t) = \begin{bmatrix} \mathbf{Y}(t) \\ 0 \end{bmatrix}; \quad C(t) = \begin{bmatrix} H(t) \\ S \end{bmatrix}; \quad V(t) = \begin{bmatrix} E(t) \\ \mathbf{F}(t) \end{bmatrix}. \quad (4.1.2)$$

4.1. The Pseudo-RLS Algorithm Analysis

With the model assumed in (4.1.1), as we stated before, the first approach that comes to mind is the Kalman filter. This approach is, however, computationally prohibitive. Instead we proposed the pseudo-RLS algorithm, which as we have seen in Section 3.3 is defined by the equation:

$$R(t)\hat{\mathbf{X}}(t) = \mathbf{P}(t), \quad (4.1.3) \quad \lambda(t_1 + 1)P_1(t_1 + 1)R(t_1)$$

where $R(t)$ and $\mathbf{P}(t)$ satisfy the updating formula

$$R(t) = \lambda(t)R(t-1) + C^T(t)W_{\check{V}}^{-1}(t)C(t) \cdot \sigma_c^2 \quad (4.1.4)$$

$$\mathbf{P}(t) = \lambda(t)\mathbf{P}(t-1) + C^T(t)W_{\check{V}}^{-1}(t)\check{\mathbf{Y}}(t) \cdot \sigma_c^2 \quad (4.1.5)$$

and $0 < \lambda(t) < 1$. We first establish a relationship between the pseudo-RLS and the Kalman filter.

THEOREM 4.1-1. *The pseudo-RLS algorithm is in fact the Kalman filter when the model assumption in (4.1.1) is replaced by*

$$\mathbf{X}_1(t) = \mathbf{X}_1(t-1) + \mathbf{N}_1(t)$$

$$\mathbf{N}_1(t) \Rightarrow \mathbf{G}\{\mathbf{0}, W_{\mathbf{N}_1}(t) = \alpha(t)R^{-1}(t-1)\}$$

$$\check{\mathbf{Y}}(t) = C(t)\mathbf{X}_1(t) + V(t) \quad (4.1.6)$$

$$V(t) \Rightarrow \mathbf{G}\left\{\mathbf{0}, W_V(t) = \begin{bmatrix} \sigma_c^2 I & 0 \\ 0 & \sigma_v^2 I \end{bmatrix}\right\},$$

where

$$\alpha(t) = \frac{1 - \lambda(t)}{\lambda(t)}. \quad (4.1.7)$$

Proof. The Kalman filter equations for the model (4.1.6) are given by

$$\hat{\mathbf{X}}_1(t) = \hat{\mathbf{X}}_1(t-1) + K(t)[\check{\mathbf{Y}}(t) - C(t)\hat{\mathbf{X}}_1(t-1)] \quad (4.1.8)$$

$$K(t) = P_1(t)C^T(t)[C(t)P_1(t)C^T(t) + W_V(t)]^{-1} \quad (4.1.9)$$

and

$$\begin{aligned} P_1(t+1) &= P_1(t) + W_{\mathbf{N}_1}(t+1) \\ &\quad - P_1(t)C^T(t)[C(t)P_1(t)C^T(t) \\ &\quad + W_V(t)]^{-1}C(t)P_1(t). \end{aligned} \quad (4.1.10)$$

The matrix $P_1(t)$ plays the role of the prediction error covariance matrix in the Kalman filter. Then, assuming the initial conditions satisfy

$$\lambda(1)P_1(1)R(0) = I, \quad (4.1.11)$$

we can show that

$$\lambda(t)P_1(t)R(t-1) = I \quad \forall t > 0. \quad (4.1.12)$$

To show that let us assume (4.1.12) holds for some $t = t_1$. Then, using (4.1.10), (4.1.12), and (4.1.7) we get

$$\begin{aligned} &= \lambda(t_1 + 1)\alpha(t_1 + 1)R^{-1}(t_1)R(t_1) \\ &\quad + \lambda(t_1 + 1)(P_1^{-1}(t_1) + C^T(t_1)W_{\check{V}}^{-1}(t_1)C(t_1))^{-1}R(t_1) \\ &= [1 - \lambda(t_1 + 1)]I \\ &\quad + \lambda(t_1 + 1)[I + P_1(t_1)C^T(t_1)W_{\check{V}}^{-1}(t_1)C(t_1)]^{-1}P_1(t_1)R(t_1). \end{aligned}$$

But by (4.1.4) and (4.1.2), for $t = t_1$ we have

$$\begin{aligned} P_1(t_1)R(t_1) &= \lambda(t_1)P_1(t_1)R(t_1 - 1) + P_1(t_1)C^T(t_1)W_{\check{V}}^{-1}(t_1)C(t_1) \\ &= I + P_1(t_1)C^T(t_1)W_{\check{V}}^{-1}(t_1)C(t_1), \end{aligned}$$

so we get by substitution

$$\begin{aligned} \lambda(t_1 + 1)P_1(t_1 + 1)R(t_1) &= [1 - \lambda(t_1 + 1)]I \\ &\quad + \lambda(t_1 + 1)[P_1(t_1)R(t_1)]^{-1}P_1(t_1)R(t_1) = I \end{aligned}$$

and using simple induction (4.1.12) follows for all $t > 0$.

Now substituting (4.1.3) in (4.1.4) and using (4.1.12) to show that $C^T(t)W_{\check{V}}^{-1}(t) = R(t)K(t)$, we can readily show that $\hat{\mathbf{X}}(t)$ in (4.1.3) satisfies Eqs. (4.1.8), which completes the proof. ■

The significance of Theorem 4.1-1 can be seen in the following result, which relies heavily on Theorem 4.1-1.

THEOREM 4.1-2. *The pseudo-RLS algorithm guarantees an unbiased estimate of the optical flow $X(t)$, and if we choose $\lambda(t)$ such that*

$$\lambda(t) \leq \frac{1}{1 + \sigma_{\mathbf{N}}^2 \|R(t-1)\|}, \quad (4.1.13)$$

we are guaranteed to have

$$\Sigma(t) = E\{[\hat{\mathbf{X}}(t) - \mathbf{X}(t)][\hat{\mathbf{X}}(t) - \mathbf{X}(t)]^T\} \leq R^{-1}(t). \quad (4.1.14)$$

Namely, the pseudo-RLS estimation error covariance matrix is bounded by the matrix $R^{-1}(t)$.

Proof. The unbiasedness of the estimate follows directly from Theorem 4.1-1 and the Kalman filter properties. For $\Sigma(t)$ we have from (4.1.1), Theorem 4.1-1, and (4.1.8)

$$\begin{aligned} \Sigma(t) &= (I - K(t)C(t))(\Sigma(t-1) \\ &\quad + W_{\mathbf{N}}(t))(I - K(t))^T + K(t)W_V(t)K^T(t). \end{aligned} \quad (4.1.15)$$

While using (4.1.1), (4.1.9), (4.1.10), and Theorem 4.1-1 we can also show that

$$\begin{aligned} \lambda(t+1)P_1(t+1) &= (I - K(t)C(t))P_1(t)(I - K(t)C(t))^T \\ &\quad + K(t)W_V(t)K^T(t). \end{aligned}$$

Subtracting this from (4.1.15), we get

$$\begin{aligned} & \Sigma(t) - \lambda(t+1)P_1(t+1) \\ &= (I - K(t)C(t))(\Sigma(t-1) - \lambda(t)P_1(t))(I - K(t)C(t))^T \\ & \quad + (I - K(t)C(t))[W_N(t) - (1 - \lambda(t))P_1(t)](I - K(t)C(t))^T. \end{aligned}$$

Clearly, from (4.1.1) and (4.1.12) we have

$$\begin{aligned} & W_N(t) - (1 - \lambda(t))P_1(t) \\ & \leq \sigma_N^2 R^{-1}(t-1)[R(t-1) - \|R(t-1)\|I] \leq 0. \end{aligned} \quad (4.1.17)$$

On the other hand, using the definition of $C(t)$ in (4.1.2) and Eqs. (4.1.1), (4.1.4), (4.1.9), and (4.1.12) we can conclude that there exists $\varepsilon > 0$ such that

$$(I - K(t)C(t)) = \left[I + P_1(t)C(t)W_V^{-1}(t)C^T(t) \right]^{-1} \leq \frac{1}{1 + \varepsilon}. \quad (4.1.18)$$

Then, assuming that $\Sigma(0) \leq \lambda(1)P_1(0)$, (4.1.14) follows directly from (4.1.16), (4.1.17), and (4.1.8), and the theorem is proven. ■

The pseudo-RLS estimation result will be denoted in the following analysis as $\hat{\mathbf{X}}_{\text{opt}}(t) = R^{-1}(t)\mathbf{P}(t)$. These vectors are not obtainable by practical means since they require the inversion of $R(t)$ which is an impossible task. The purpose of the M -SD and the M -LMS algorithms is to supply an estimate which attempts to approximate $\hat{\mathbf{X}}_{\text{opt}}(t)$ —this is the reason we denote the pseudo-RLS result as an optimal sequence.

The following lemma presents an important property of the sequence $\{\hat{\mathbf{X}}_{\text{opt}}(t)\}_{t>0}$ —bounded variation of the sequence. This property will be used in the analysis of the M -SD and the M -LMS estimation algorithms.

LEMMA 4.1–1. *The sequence $\{\hat{\mathbf{X}}_{\text{opt}}(t)\}_{t>0}$ satisfies the following property:*

$$\Delta_D \triangleq \sup_{t>1} E\{\|\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)\|\} < \infty. \quad (4.1.18)$$

Proof. The proof is given in Appendix B. ■

In this sub-section we have shown that the pseudo-RLS algorithm is actually an approximated Kalman filter with bounded estimation error and unbiased estimation. However, despite the proposed approximation, the pseudo-RLS algorithm is yet far too complex to be implemented. The M -SD and the M -LMS algorithms are practical approximations of the pseudo-RLS. In the following subsections we shall analyze their properties.

4.2. The M -SD Algorithm Analysis

The M -SD algorithm equations updating the estimate of the optical flow in time are given by

$$\hat{\mathbf{X}}_0(t) = \hat{\mathbf{X}}_M(t-1)$$

$$\mathbf{X}_j(t) = \hat{\mathbf{X}}_{j-1}(t) + \mu[\mathbf{P}(t) - R(t)\hat{\mathbf{X}}_{j-1}(t)] \quad \text{for } 1 \leq j \leq M. \quad (4.2.1)$$

Chaining these equations together we get a single recursive equation which presents the relation between the M th result for time $(t-1)$ and the M th result for time (t) ,

$$\begin{aligned} \hat{\mathbf{X}}_M(t) &= [I - \mu R(t)]^M \hat{\mathbf{X}}_M(t-1) + \mu \sum_{k=0}^{M-1} [I - \mu R(t)]^k \mathbf{P}(t) \\ &= [I - \mu R(t)]^M [\hat{\mathbf{X}}_M(t-1) - \hat{\mathbf{X}}_{\text{opt}}(t)] + \hat{\mathbf{X}}_{\text{opt}}(t), \end{aligned} \quad (4.2.2)$$

where we have used the fact that $R(t)\hat{\mathbf{X}}_{\text{opt}}(t) = \mathbf{P}(t)$, and the formula for a sum of geometric sequences [16]. The relationship between the M -SD algorithm and the pseudo-RLS solution is given by the following theorem.

THEOREM 4.2–1. *Consider the M -SD algorithm as given in Eq. (4.2.2) with arbitrary initial conditions $\hat{\mathbf{X}}_M(0)$, $\mathbf{P}(0)$, and $R(0) \geq 0$. Let*

$$0 < \mu < \frac{2}{\lambda_{\max}}, \quad \text{where} \quad \lambda_{\max} = \sup_{t>0} \left\{ \max_{1 \leq k \leq N^2} \{\lambda_k\{R(t)\}\} \right\}. \quad (4.2.3)$$

Then $\exists 0 < \varepsilon < 1$ such that

$$\begin{aligned} \forall t > 0 \quad \sigma_D(t) &\triangleq E\{\|\hat{\mathbf{X}}_M(t) - \hat{\mathbf{X}}_{\text{opt}}(t)\|\} \\ &\leq \frac{\Delta_D \cdot [1 - \varepsilon]^M}{1 - [1 - \varepsilon]^M} + [1 - \varepsilon]^{Mt} \cdot \sigma_D(0). \end{aligned} \quad (4.2.4)$$

Proof. Since $R(t)$ is positive definite, the spectral radius of the matrix $M(t) = [I - \mu R(t)]^M$ is

$$\begin{aligned} \|M(t)\|_2 &= \|[I - \mu R(t)]^M\|_2 \leq \|I - \mu R(t)\|_2^M \\ &= \|I - \mu U(t)\Delta(t)U^T(t)\|_2^M \\ &= \|U(t)[I - \mu\Delta(t)]U^T(t)\|_2^M = \|I - \mu\Delta(t)\|_2^M. \end{aligned} \quad (4.2.5)$$

The obtained matrix $I - \mu\Delta(t)$ is diagonal with $1 - \mu\lambda_k(t)$ on the main diagonal. The choice of μ as given in (4.2.3) guarantees that $|\lambda\{M(t)\}| < 1$, and thus

$$\exists 0 < \varepsilon < 1 \quad \forall t > 0 \quad \|M(t)\|_2 \leq [1 - \varepsilon]^M. \quad (4.2.6)$$

In order to guarantee that such a choice of μ always exists we must show the following two things:

$$\begin{aligned}\lambda_{\max} &= \sup_{t>0} \left\{ \max_{1 \leq k \leq N^2} \{\lambda_k\{R(t)\}\} \right\} < \infty; \\ \lambda_{\min} &= \inf_{t>0} \left\{ \min_{1 \leq k \leq N^2} \{\lambda_k\{R(t)\}\} \right\} > 0.\end{aligned}\quad (4.2.7)$$

The minimum eigenvalue is indeed higher than zero since $R(t)$ is positive definite for all t . The maximum eigenvalue can be bounded by

$$\begin{aligned}\lambda_{\max} &< \sup_{t \geq 0} \text{tr}\{R(t)\} \\ &= \sup_{t \geq 0} \sum_{k=0}^{\infty} \lambda^k \text{tr}\{H^T(t-k)H(t-k) + \beta \mathbf{S}^T \mathbf{S}\} \\ &\leq \sum_{\substack{\text{tr}\{H^T(t-k)H(t-k)\} \leq T_1 \\ \text{tr}\{\beta \mathbf{S}^T \mathbf{S}\} \leq T_2}} \sum_{k=0}^{\infty} \lambda^k [T_1 + T_2] = \frac{T_1 + T_2}{1 - \lambda} < \infty.\end{aligned}\quad (4.2.8)$$

Using (4.2.2) we get

$$\begin{aligned}\hat{\mathbf{X}}_M(t) - \hat{\mathbf{X}}_{\text{opt}}(t) &= [I - \mu R(t)]^M \{[\hat{\mathbf{X}}_M(t-1) - \hat{\mathbf{X}}_{\text{opt}}(t-1)] \\ &\quad - [\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)]\}.\end{aligned}\quad (4.2.9)$$

Applying Euclidean norm and expectation to the above equality, using the triangle inequality, and inserting the result of Lemma 4.1-1 we get

$$\sigma_{\text{D}}(t) \leq [1 - \varepsilon]^M \{\sigma_{\text{D}}(t-1) + \Delta_{\text{D}}\}. \quad (4.2.10)$$

By applying the above inequality recursively we get the inequality in (4.2.4), which completes this theorem's proof. ■

From the above theorem it is evident that the smaller the changes in $\hat{\mathbf{X}}_{\text{opt}}(t)$ with time, the better the M -SD tracking capabilities are. Increasing M clearly improves the M -SD tracking performance—any desired accuracy can be reached with large enough M .

4.3. The M -LMS Algorithm Analysis

The M -LMS algorithm updating equations are

$$\begin{aligned}\hat{\mathbf{x}}_0(t) &= \hat{\mathbf{x}}_m(t-1) \\ \hat{\mathbf{x}}_j(t) &= \hat{\mathbf{x}}_{j-1}(t) + \mu [H^T(t)\mathbf{Y}(t) \\ &\quad - [H^T(t)H(t) + \beta \mathbf{S}^T \mathbf{S}]\hat{\mathbf{x}}_{j-1}(t)] \quad \text{for } 1 \leq j \leq m.\end{aligned}\quad (4.3.1)$$

Let us define the sequence $\hat{\chi}_{\text{opt}}(t)$ through $\mathbf{p}(t) = r(t)\hat{\chi}_{\text{opt}}(t)$, where

$$r(t) = H^T(t)H(t) + \beta \mathbf{S}^T \mathbf{S}; \mathbf{p}(t) = H^T(t)\mathbf{Y}(t). \quad (4.3.2)$$

Then, from (4.3.1), we get

$$\begin{aligned}\hat{\mathbf{x}}_m(t) &= [I - \mu r(t)]^M \hat{\mathbf{x}}_m(t-1) + \mu \sum_{k=0}^{m-1} [I - \mu r(t)]^k \mathbf{p}(t) \\ &= [I - \mu r(t)]^m [\hat{\mathbf{x}}_m(t-1) - \hat{\chi}_{\text{opt}}(t)] + \hat{\chi}_{\text{opt}}(t).\end{aligned}\quad (4.3.3)$$

This equation is similar to the one given for the M -SD algorithm with matrix $r(t)$ replacing $R(t)$, vector $\mathbf{p}(t)$ replacing $\mathbf{P}(t)$, and $\hat{\chi}_{\text{opt}}(t)$ replacing $\hat{\mathbf{X}}_{\text{opt}}(t)$. Note that the sequence $\{\hat{\chi}_{\text{opt}}(t)\}_{t \geq 0}$ is actually the Horn and Schunck optical flow as described in Eq. (2.9).

From Eq. (4.3.3) clearly we get $\hat{\mathbf{x}}_m(t) \rightarrow \hat{\chi}_{\text{opt}}(t)$ as $m \rightarrow \infty$. However, as argued earlier, this is not a desired result. In contrary to the M -SD algorithm, increasing the number of iterations per time point does not necessarily improve the estimation performance. Actually, since the M -LMS can be obtained from the M -SD algorithm using $\lambda(t) = 0$ (thus removing the temporal smoothness factor), using small m values is the only mechanism by which to reach the temporal memory.

THEOREM 4.3-1. Consider the M -LMS algorithm as given in Eq. (4.3.1) with arbitrary initial condition $\hat{\mathbf{x}}_m(0)$. Let

$$0 < \mu < \frac{2}{\lambda_{\max}}, \quad \text{where} \quad \lambda_{\max} = \sup_{t>0} \left\{ \max_{1 \leq k \leq N^2} \{\lambda_k\{r(t)\}\} \right\}.\quad (4.3.4)$$

Then $\exists 0 < \varepsilon < 1$ and $0 < C_1 < \infty$ such that

$$\begin{aligned}\forall t > 0 \quad \sigma_{\text{f}}(t) &\triangleq E\{\|\hat{\mathbf{x}}_m(t) - \hat{\mathbf{X}}_{\text{opt}}(t)\|\} \\ &\leq [1 - \varepsilon]^m \sigma_{\text{f}}(0) + \frac{[[1 - \varepsilon]^{m-1}(C_1 + 1) + C_1]}{1 - [1 - \varepsilon]^m} \Delta_{\text{D}}.\end{aligned}\quad (4.3.5)$$

Proof. Using Eqs. (4.1.3–5) and (4.3.1) we get that for the first iteration

$$\begin{aligned}\hat{\mathbf{f}}_1(t) &\triangleq \hat{\mathbf{x}}_1(t) - \hat{\mathbf{X}}_{\text{opt}}(t) \\ &= \hat{\mathbf{x}}_m(t-1) - \hat{\mathbf{X}}_{\text{opt}}(t) + \mu [\mathbf{P}(t) - R(t)\hat{\mathbf{x}}_m(t-1)] \\ &\quad - \lambda \mu [\mathbf{P}(t-1) - R(t-1)\hat{\mathbf{x}}_m(t-1)] \\ &= [I - \mu R(t)][\hat{\mathbf{f}}_m(t-1) - \hat{\Delta}_{\text{opt}}(t)] \\ &\quad + \lambda \mu R(t-1)\hat{\mathbf{f}}_m(t-1) \\ &= [I - \mu r(t)]\hat{\mathbf{f}}_m(t-1) - [I - \mu R(t)]\hat{\Delta}_{\text{opt}}(t),\end{aligned}\quad (4.3.6)$$

where we have defined $\hat{\Delta}_{\text{opt}}(t) \triangleq \hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)$. Similarly for the $2 \leq k \leq m$ other iterations we get

$$\begin{aligned}
 \hat{\mathbf{f}}_k(t) &\triangleq \hat{\mathbf{x}}_k(t) - \hat{\mathbf{X}}_{\text{opt}}(t) \\
 &= \hat{\mathbf{x}}_{k-1}(t) - \hat{\mathbf{X}}_{\text{opt}}(t) + \mu[\mathbf{P}(t) - R(t)\hat{\mathbf{x}}_{k-1}(t)] \\
 &\quad - \lambda\mu[\mathbf{P}(t-1) - R(t-1)\hat{\mathbf{x}}_{k-1}(t)] \\
 &= \hat{\mathbf{f}}_{k-1}(t) - \mu R(t)\hat{\mathbf{f}}_{k-1}(t) \\
 &\quad - \lambda\mu R(t-1)[\hat{\mathbf{X}}_{\text{opt}}(t-1) - \hat{\mathbf{x}}_{k-1}(t)] \\
 &= [I - \mu r(t)]\hat{\mathbf{f}}_{k-1}(t) + \lambda\mu R(t-1)\hat{\mathbf{D}}_{\text{opt}}(t).
 \end{aligned} \tag{4.3.7}$$

Chaining these m equations together we get

$$\begin{aligned}
 \hat{\mathbf{f}}_m(t) &= [I - \mu r(t)]^m \hat{\mathbf{f}}_m(t-1) \\
 &\quad - [I - \mu r(t)]^m r^{-1}(t) R(t) \hat{\mathbf{D}}_{\text{opt}}(t) \\
 &\quad + [r^{-1}(t) R(t) - I] \hat{\mathbf{D}}_{\text{opt}}(t).
 \end{aligned} \tag{4.3.8}$$

Choosing μ according to (4.3.4) guarantees that $\exists \varepsilon \in (0, 1) \mid \forall t \parallel I - \mu r(t) \parallel < 1 - \varepsilon$. Applying Euclidean norm and expectation to the above equality and inserting the result of Lemma 4.1–1, we get

$$\begin{aligned}
 \sigma_{\mathbf{f}}(t) &\triangleq E\{\|\hat{\mathbf{f}}_m(t)\|\} \leq [1 - \varepsilon]^m \sigma_{\mathbf{f}}(t-1) \\
 &\quad + \frac{1}{[1 - \varepsilon]^{m-1} C_1 + [C_1 + 1]} \Delta_{\mathbf{D}},
 \end{aligned} \tag{4.3.9}$$

where we have used the bounds

$$\begin{aligned}
 \parallel r^{-1}(t) R(t) \parallel &\leq \parallel r^{-1}(t) \parallel \cdot \parallel R(t) \parallel \leq C_1 \\
 \parallel I - r^{-1}(t) R(t) \parallel &\leq 1 + \parallel r^{-1}(t) \parallel \cdot \parallel R(t) \parallel \leq C_1 + 1.
 \end{aligned} \tag{4.3.10}$$

Applying the above inequality, recursively we get

$$\begin{aligned}
 \sigma_{\mathbf{f}}(t) &\leq [1 - \varepsilon]^m \sigma_{\mathbf{f}}(0) + \sum_{k=0}^{t-1} [1 - \varepsilon]^{mk} \\
 &\quad \frac{1}{[1 - \varepsilon]^{m-1} C_1 + (C_1 + 1)} \Delta_{\mathbf{D}} \\
 &\leq [1 - \varepsilon]^m \sigma_{\mathbf{f}}(0) + \frac{1 - [1 - \varepsilon]^{mt}}{1 - [1 - \varepsilon]^m} \\
 &\quad \frac{1}{[1 - \varepsilon]^{m-1} C_1 + (C_1 + 1)} \Delta_{\mathbf{D}} \\
 &\leq [1 - \varepsilon]^m \sigma_{\mathbf{f}}(0) + \frac{[1 - \varepsilon]^{m-1} C_1 + (C_1 + 1)}{1 - [1 - \varepsilon]^m} \Delta_{\mathbf{D}},
 \end{aligned} \tag{4.3.11}$$

which is exactly the statement of the theorem. ■

We have derived performance bounds for the M -SD and M -LMS algorithms. While these bounds are quite similar, our experience indicates a difference in the performance of these two algorithms. In particular, we refer to the dependence of the performance of these algorithms on the value of M —the repetitious between-sample times. In the

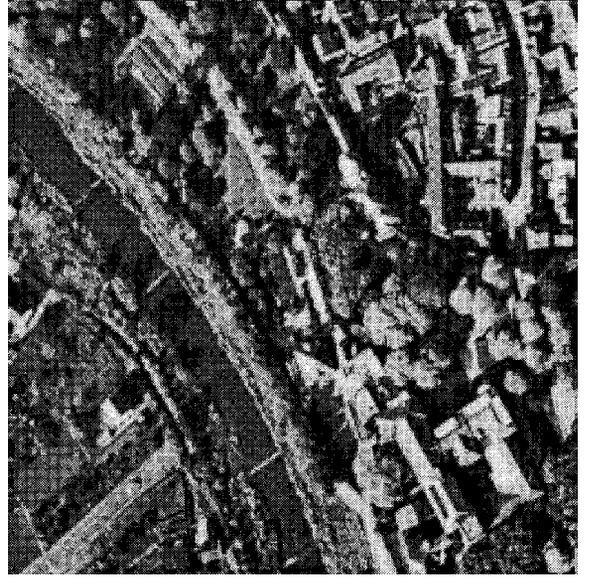


FIG. 1. The source image.

M -SD both the bound derived and our experiments show monotonicity—the larger M is, the better the performance. However, in the M -LMS, as we commented after equation (4.3.3), the performance for large M is not satisfactory. Hence, there seems to be a value of M at which the performance is optimal and increasing M beyond this value causes deterioration in performance. We did not succeed in capturing this phenomenon in our derivation of the bounds in Eq. (4.3.5) and Eq. (4.3.8).

5. SIMULATIONS AND RESULTS

In this section we shall present various demonstrations of the proposed optical flow estimation algorithms and discuss their properties. All the tests were performed on semisynthetic image sequences with a priori known optical flow, in order to be able to quantify the results. There were four different image sequences, all based on a single image shown in Fig. 1.

In the first and second image sequences the optical flow is constant in time—one with constant translations ($[-0.7, 0.5]$ per image) and the second with constant rotation (1.2° per image). In the third image sequence there is a constant rotation as above with additional zoom in and out in the form of half a cycle of a sine function in the range $[0.85-1.15]$. The fourth sequence was constructed by constant translation ($[-0.6, 0]$ per image) with the same zoom in and out as in the third sequence. The optical flow sequences in the third and fourth sequences thus change in time. Figures 2, 3, 4, and 5 each present six equally spaced images and optical flow maps from these sequences. Each of these four sequences contained 101 images of size $[50 \times 50]$

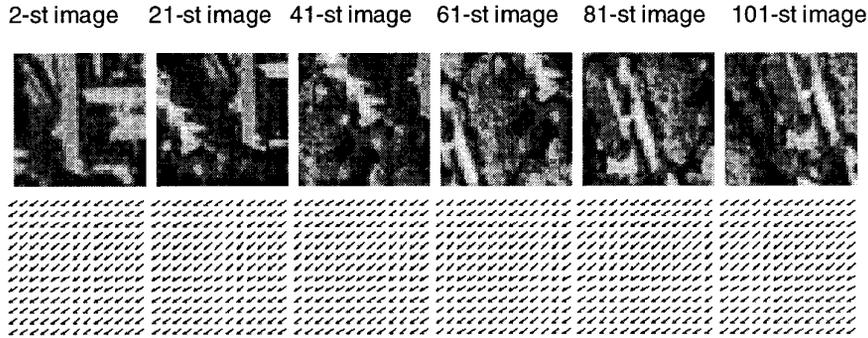


FIG. 2. The first image sequence.

pixels. Each image in these movies was contaminated by additive Gaussian white random noise with variance $\sigma_n = 4$, where the dynamic range of the images is $[0, 255]$.

Because there are too many result graphs, we have chosen to include only part of the graphical results and only for the *third sequence only*, and to refer to all the other results only if important contradictions exists.

In all the demonstrations we have compared the estimated optical flow to the true optical flow in order to determine quantitative results. Two formulations for the computation of this estimation errors were used—the direct (normalized) mean square error (DMSE) (as used in [8]) and the weighted (normalized) mean square error (WMSE). These two errors were computed according to

$$\begin{aligned}
 E_{\text{DMSE}}(t) &= \frac{\|\hat{\mathbf{X}}(t) - \mathbf{X}(t)\|_2}{\|\mathbf{X}(t)\|_2} \\
 &= \sqrt{\frac{\sum_{k=1}^{N^2} [\hat{\mathbf{X}}_k(t) - \mathbf{X}_k(t)]^2}{\sum_{k=1}^{N^2} [\mathbf{X}_k(t)]^2}} \quad (5.1)
 \end{aligned}$$

$$\begin{aligned}
 E_{\text{WMSE}}(t) &= \frac{\|\hat{\mathbf{X}}(t) - \mathbf{X}(t)\|_{W(t)}}{\|\mathbf{X}(t)\|_2} \\
 &= \sqrt{\frac{\sum_{k=1}^{N^2} W_k(t) [\hat{\mathbf{X}}_k(t) - \mathbf{X}_k(t)]^2}{\sum_{k=1}^{N^2} W_k(t) [\mathbf{X}_k(t)]^2}}. \quad (5.2)
 \end{aligned}$$

The weights that were used in the WMSE were based on the confidence measures presented in Section 3.5. The confidence measures were scaled so that the least confident estimates are given a zero weight, and after this scaling each motion vector received the square of its confidence measure as its weight (this choice was found empirically to be good).

The image sequences were spatially blurred using a 5×5 uniform smoothing kernel before entering the optical flow estimation process. This step is well recommended [2, 4, 7, 8] for additive noise suppression and better match to the differential framework. The boundaries of the images were found to cause estimation errors because of the pre-smoothing which is a neighborhood operation. In order to

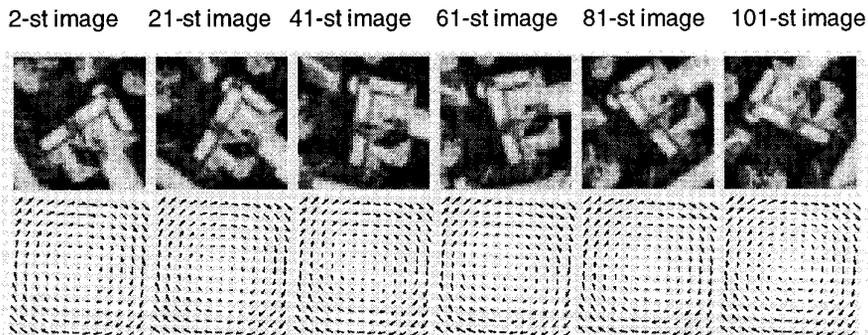


FIG. 3. The second image sequence.

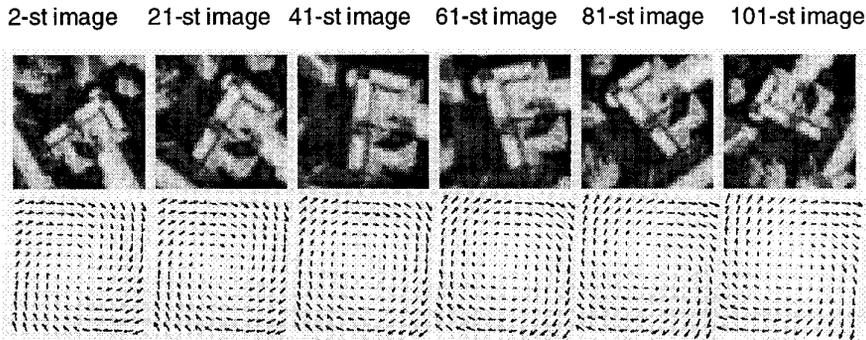


FIG. 4. The third image sequence.

remove the influence of the spatial and temporal erroneous gradients on the estimation results, we used a diagonal weight matrix V in the quadratic term in Eq. (3.4). The alternative squared error is therefore

$$\varepsilon^2(t) = \sum_{k=0}^{\infty} \lambda^k \{ \|\mathbf{Y}(t-k) - H(t-k)\mathbf{X}(t)\|_V^2 + \beta \|\mathbf{S}\mathbf{X}(t)\|_2^2 \}, \quad (5.3)$$

where the diagonal of the matrix V is zero for boundary pixels (three pixels from each border), and ones otherwise. This way we also got as a byproduct that the confidence measures of the estimate results in the boundaries were the lowest.

The Laplacian which was used is the one that was recommended by Horn and Schunch [2]. In lines which correspond to boundaries of the optical flow, the main diagonal element were modified to give that the sum of the elements in the line is zero. The spatial derivatives were computed with the simple kernel $[-1, 0, 1]/2$. The temporal gradients were computed by simple subtraction of the current and the previous images.

We first present the various results which corresponds to the M -SD algorithm. In this algorithm we are propagat-

ing the matrix $R(t)$ and the vector $\mathbf{P}(t)$ in time. Both of them were initialized to zero for $t = 0$. Unless stated otherwise, the iterative scheme was that used in normalized steepest descent, so the parameter μ is determined internally. In all the simulations we tested the performance of the algorithm twice—once with initialization $\hat{\mathbf{X}}(0) = \mathbf{0}$ to learn the convergence properties, and ones with true optical flow $\hat{\mathbf{X}}(0) = \mathbf{X}_{\text{opt}}(1)$ to learn the steady-state and tracking performances. However, we shall present the $\hat{\mathbf{X}}(0) = \mathbf{0}$ initialization only.

Graph 1 presents the influence of λ on the algorithm performance for the third sequence. Since the third sequence has a temporally varying optical flow, a less than one forgetting factor is required in order to allow for changes in the estimation process. Choosing $\lambda = 0.8-0.85$ ($\beta = 1000$, and $M = 10$ iterations) gave similar results—the WMSE converged to less than 10% error for both zero and true initializations. Graph 2 presents the performance for varying β . It was found that $\beta = 300$ gives the best results, leading to less than 10% weighted estimation error ($\lambda = 0.85$ and $M = 10$). Graph 3 tested the influence of the number of iterations M . As expected, the higher the value of M , the faster the convergence, and to a lower steady state error. In this case ($\lambda = 0.85$ and $\beta = 1000$), for $M = 30$ we obtained 8% weighted estimation error.

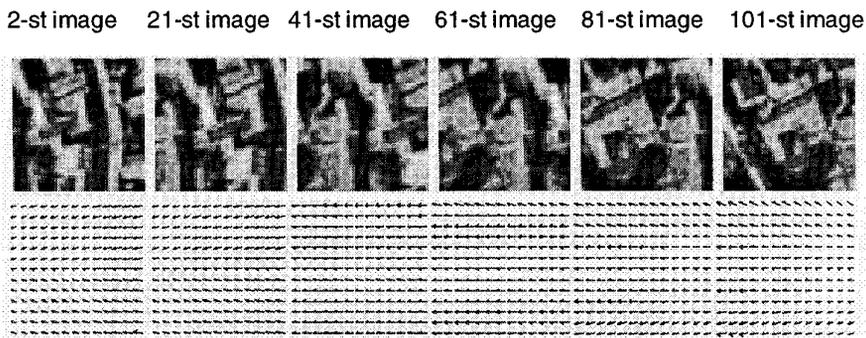
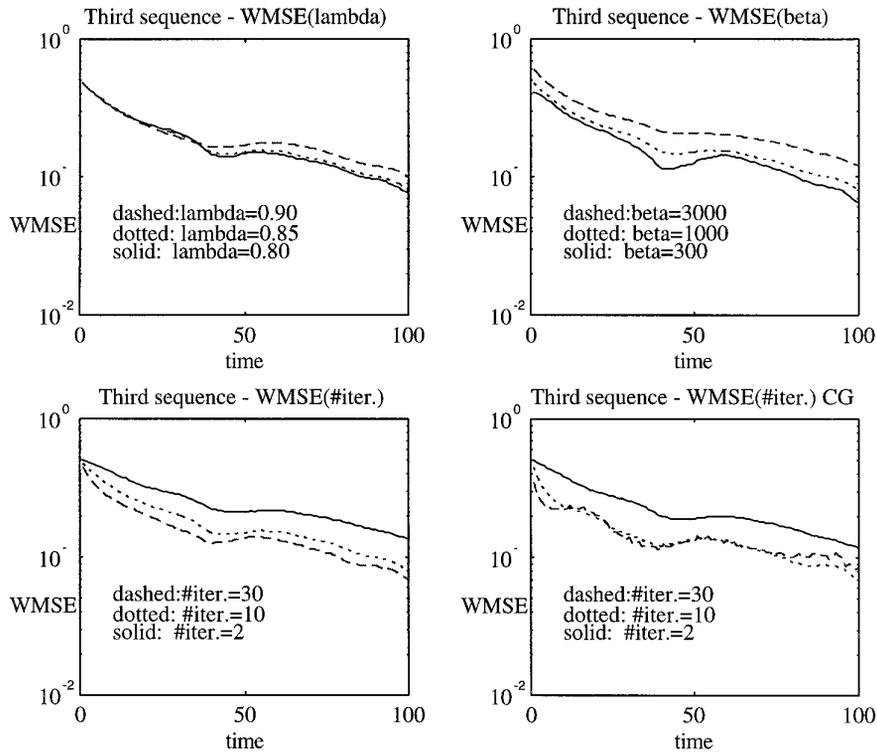


FIG. 5. The fourth image sequence.



GRAPHS 1–4. The influence of λ , β , M , and the replacement with the CG on the M -SD performance.

The final simulated comparison with regard to the M -SD is the replacement of the NSD with the conjugate gradient (CG). For a small number of iterations ($M = 2$) this change almost was not perceived. This is because two iterations of the NSD are almost equivalent to two CG iterations. For higher values of M ($M = 10$ and 30) faster convergence was obtained, but with the same steady state error. Graph 4 present these results.

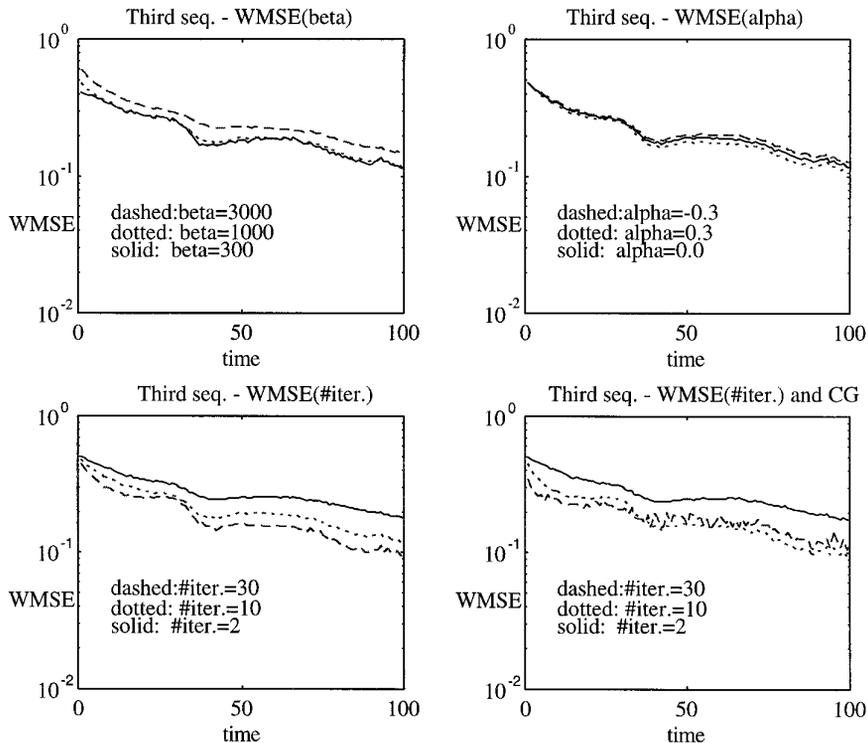
Referring to the other image sequences, the results were very similar. The best λ for the first and second sequences was $\lambda = 0.95$, reflecting the fact that the optical flow sequence is constant in time. The best value of β was 5000 if fast convergence was sought, but higher values gave lower steady state error. The weighted errors for these sequences was in the range 2–5%. The parameters that best matched the fourth sequence were $\lambda = 0.8$ and $\beta = 300$. The steady state weighted error in this case was 12%.

We now turn to present the various results which correspond to the M -LMS algorithm. We have tested this algorithm simply by using the M -SD algorithm with $\lambda = 0$, the iterative scheme that was used in normalized steepest descent. As before, we have made each test twice, for $\hat{\mathbf{X}}(0) = \mathbf{0}$ and for $\hat{\mathbf{X}}(0) = \mathbf{X}_{\text{opt}}(1)$, but only the zero initialization will be presented here. The best value of β for the third sequence was in the range $\beta = 300$ – 1000 . The steady state weighted error was 10% (for $m = 10$ iterations)

(Graph 5). We have tried second order AR model by using $\alpha = \pm 0.3$. As can be shown from Graph 6, it was found that the smoothing choice $\alpha = 0.3$ gave slightly better results. Increasing the number of iterations m in the low range of values ($m < 50$) gave better results both for the convergence and the steady state results (Graph 7). Changing to the CG algorithm almost did not influence the performance results (Graph 8).

Figure 6 presents the true optical flow (A) versus the performance of the M -SD (B), the M -LMS (C), and the Horn and Schunck (D) algorithms. The parameters for the various parameters were chosen to be the ones with the best results from the previous graphs. As can be seen, the M -SD and the M -LMS outperform the Horn and Schunck algorithm, and converge to high quality optical flow.

Generally speaking, we can make the following important observations: First, the obtained DMSE for all these cases was significantly higher than the WMSE, revealing the effectiveness of the proposed confidence measure. Second, all the simulations revealed a robustness to the various parameters involved—the performance does not change significantly with the parameters. Third, in order to get better insight into the estimation error results we have simulated the Horn and Schunck algorithm on the same sequences. Graph 9 shows the performance of the Horn



GRAPHS 5-8. The influence of β , α , M , and the replacement with the CG on the M -LMS performance.

and Schunck algorithm [2] on the third sequence for various values of β . As can be seen, the error is almost constant and is relatively very high (WMSE = 35%). We should note that these results were obtained by performing 200 NSD iterations per each temporal point, thus using much

more computations to yield quite poor results. This comparison confirms our original claim that improvement is possible both for the accuracy and complexity points of view if the time axis is used properly.

One final remark that should be mentioned is this: we

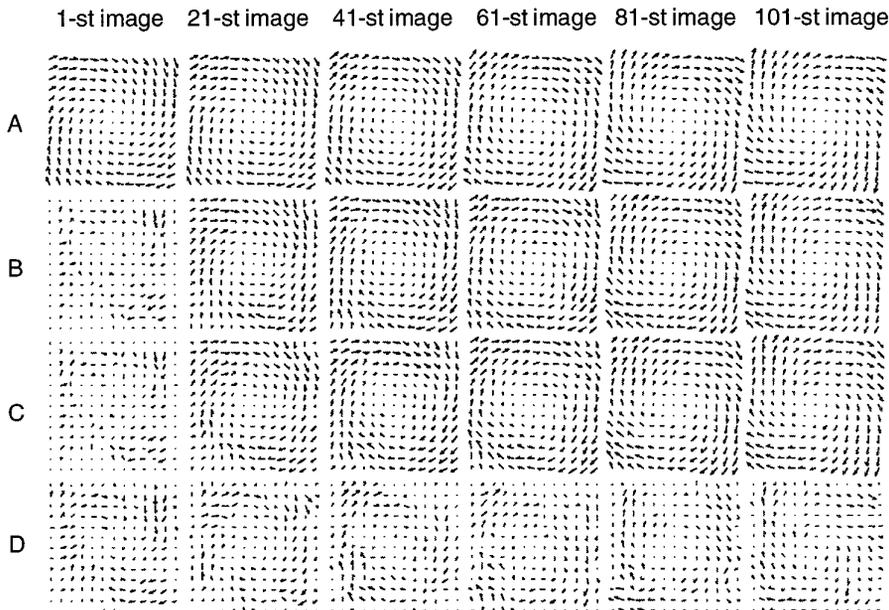
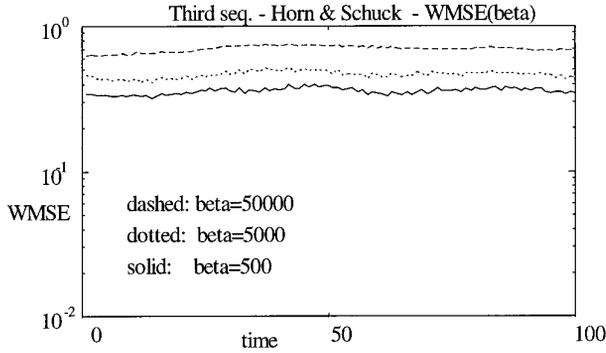


FIG. 6. The optical flow, (A) true, (B) M -SD, (C) M -LMS, (D) Horn and Schunck algorithms.



GRAPH 9. Horn and Schunck algorithm performance.

have simulated only image sequences with small image size— $[50 \times 50]$ pixels—because of MATLAB's memory and CPU limitations. We believe that for bigger images, the proposed algorithms can perform better, yielding lower weighted estimation error.

6. CONCLUSION

In this paper we have presented new algorithms for the estimation of optical flow for image sequences. These new algorithms were based on the Horn and Schunck algorithm [2], generalized to include temporal smoothness. The undertaken approach started from state-space equations modeling the estimation problem, but instead of applying the Kalman filter, which seems natural at this point, we chose to further simplify the model, leading to adaptive filtering formulations resembling the RLS and the LMS algorithms [5, 10, 11].

The new estimation methods were shown to converge to the required optical flow sequence both by analytical analysis and simulations. Their main advantage is shown to be the low complexity requirement, while they give more accurate results than the Horn and Schunck algorithm. Among the interesting features of these new algorithms are their simplicity, modularity, and robustness. By simple parameter choice we can control the steady state estimation error and convergence rate, at the expense of linearly growing computational complexity.

APPENDIX A

The Structured Properties of the Matrix $R(t)$

The proposed pseudo-RLS and M -SD algorithms construct a matrix $R(t)$ as part of the estimation process. This appendix is devoted to the discussion on the sparseness of this matrix which is a very important property that can be used to save computations and memory.

DEFINITION A-1. A density number of a matrix A , de-

noted by $\#d\{A\}$, is the number of its nonzero elements relative to its size.

DEFINITION A-2. A matrix A is considered sparse if $\#d\{A\} \ll 1$.

DEFINITION A-3. A sparse matrix A is structured if its nonzero elements appear in a certain logical ordering.

THEOREM A-1. The sequence of matrices $R(t)$ (for $t > 0$) in the pseudo-RLS algorithm are structured sparse matrices for all $t > 0$ with fixed density $\#d\{R(t)\} \cong 19/N^2$ and fixed structure, where N^2 is the number of pixels in the images.

Proof. The matrix $R(t)$ is given by the equation

$$R(t) = \sum_{k=0}^{\infty} \lambda^k \{H^T(t-k)H(t-k) + \beta S^T S\} \quad (\text{A.1})$$

$$= \lambda R(t-1) + H^T(t)H(t) + \beta S^T S$$

We start by analyzing the structure of the term $H^T(t)H(t)$. The matrix $H(t)$, as given in equation (2.4), is a row combination of two diagonal matrices, where each of them is of size $[N^2 \times N^2]$. Thus, the term $H^T(t)H(t)$ has the structure described in Fig. 2, and it is easy to see that there are $4N^2$ nonzero elements in this term.

The matrix S is built from two Laplacian matrices as described in Eq. (2.7). Denoting the Laplacian matrix as S , we have that S is symmetric and thus

$$S^T S = \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} \cdot \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} = \begin{bmatrix} S \cdot S & 0 \\ 0 & S \cdot S \end{bmatrix}. \quad (\text{A.2})$$

Using the Laplacian kernel as defined by Horn and Schunck,

$$K\{\text{Laplacian}\} = \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & -1 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}. \quad (\text{A.3})$$

We have that each line in S contains the most nine nonzero

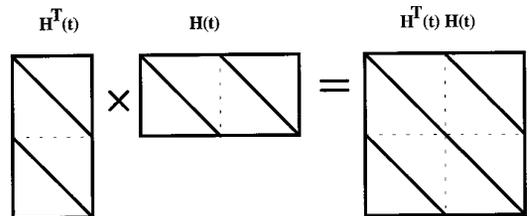


FIG. A-1. The structure of $H^T(t)H(t)$.

elements. The only lines which do not satisfy this property are the ones representing image boundaries. The multiplication of S by S is simply the performance of the Laplacian twice, and the overall kernel size of this combination contains 37 nonzero elements. Thus, the term $\mathbf{S}^T\mathbf{S}$ contains a little less than $2 \cdot 37N^2$ nonzero elements. However, the main diagonal in $\mathbf{S}^T\mathbf{S}$ is full, and so is the main diagonal of $H^T(t)H(t)$. Thus, the overall number of nonzero elements is $74N^2 + 4N^2 - 2N^2 = 76N^2$, and when divided by the matrix size $4N^4$ we get that $\#d\{H^T(t)H(t) + \beta\mathbf{S}^T\mathbf{S}\} \cong 19/N^2$. The nonzero elements of the overall matrix $H^T(H(t) + \beta\mathbf{S}^T\mathbf{S})$ are very well structured, populating 38 specific diagonals.

Since the position of the nonzero elements in $H(t)$ does not depend on the time t , we have that any combination of such matrices has the same density number, and according to Eq. (A.1) the above density also applied to $R(t)$. Thus, the density is very small, which means that $R(t)$ is very sparse, the density number is indeed constant in time, and these matrices are structured with constant ordering in time. ■

APPENDIX B

Bounding the Temporal Change of the Sequence $\{\hat{\mathbf{X}}_{\text{opt}}(t)\}_{t>0}$

LEMMA B-1. *The sequence $\{\hat{\mathbf{X}}_{\text{opt}}(t)\}_{t>0}$ satisfies the following property for all $t > 1$:*

$$\Delta_D \triangleq \sup_{t>1} E\{\|\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)\|\} < \infty. \quad (\text{B.1})$$

Proof. Let us first bound the spectral norm of the auto-correlation matrix

$$\Sigma_\Delta(t) \triangleq E\{[\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)][\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)]^T\} \quad (\text{B.2})$$

using the relations (4.1.8), (4.1.1), and (4.1.2):

$$\hat{\mathbf{X}}_{\text{opt}}(t) = [I - K(t)C(t)]\hat{\mathbf{X}}_{\text{opt}}(t-1) + K(t)\tilde{\mathbf{Y}}(t) \quad (\text{B.3})$$

$$\begin{aligned} \tilde{\mathbf{Y}}(t) &= C(t)\mathbf{X}(t) + \mathbf{V}(t) = C(t)\mathbf{X}(t-1) \\ &\quad + C(t)\mathbf{N}(t) + \mathbf{V}(t). \end{aligned} \quad (\text{B.4})$$

Combining these relations we get

$$\begin{aligned} \hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1) &= K(t)[C(t)[\mathbf{X}(t-1) \\ &\quad - \hat{\mathbf{X}}_{\text{opt}}(t-1)] + \mathbf{V}(t) + C(t)\mathbf{N}(t). \end{aligned} \quad (\text{B.5})$$

Thus, the matrix $\Sigma_\Delta(t)$ can be represented as

$$\begin{aligned} \Sigma_\Delta(t) &= K(t)C(t)[\Sigma(t-1) + W_N(t)]H^T(t)K^T(t) \\ &\quad + K(t)W_V(t)K^T(t), \end{aligned} \quad (\text{B.6})$$

where we define

$$\Sigma(t-1) \triangleq E\{[\hat{\mathbf{X}}_{\text{opt}}(t-1) - \mathbf{X}(t-1)][\hat{\mathbf{X}}_{\text{opt}}(t-1) - \mathbf{X}(t-1)]^T\}.$$

Taking the spectral norm on both sides of Eq. (B.6) we get

$$\|\Sigma_\Delta(t)\| \leq \|K(t)C(t)\|^2\|\Sigma_E(t)\| + \|W_N(t)\| + \|K(t)W_V(t)K^T(t)\|. \quad (\text{B.8})$$

From the Kalman filter equations we have that $I - K(t)C(t) = \hat{P}(t)\hat{P}^{-1}(t)$, where $\hat{P}(t)$ is the Kalman estimation error covariance matrix and $\hat{P}(t)$ is the Kalman prediction error covariance matrix. Since $\hat{P}(t) \leq \hat{P}(t)$, we get that

$$0 \leq I - K(t)C(t) \leq I \Rightarrow \|K(t)C(t)\| \leq 2. \quad (\text{B.9})$$

The matrix $\Sigma(t-1)$ is the pseudo-RLS estimation error covariance matrix. Based on the result of Theorem 4.1-2 we have

$$\begin{aligned} \Sigma(t) &\leq R^{-1}(t) = \left[\sum_{k=0}^t \lambda^k C^T(t-k)W_V^{-1}C(t-k) \right]^{-1} \\ &\leq \frac{1-\lambda}{1-\lambda^{t+1}} [S^T W_F^{-1} S]^{-1} \leq [S^T W_F^{-1} S]^{-1} \quad (\text{B.10}) \\ &\Rightarrow \|\Sigma(t)\| \leq \|[S^T W_F^{-1} S]^{-1}\| < \infty. \end{aligned}$$

The norm $\|K(t)W_V(t)K^T(t)\|$ is bounded since

$$\begin{aligned} \|K(t)W_V(t)K^T(t)\| &\leq \text{tr}\{K(t)W_V(t)K^T(t)\} \\ &= \text{tr}\{K^T(t)K(t)W_V(t)\} \\ &\leq N\|K^T(t)K(t)W_V(t)\| \\ &\leq N\|K^T(t)K(t)\| \cdot \max\{\sigma_e^2, \sigma_f^2\} \end{aligned} \quad (\text{B.11})$$

and the norm of $K^T(t)K(t)$ is bounded by

$$\begin{aligned} \|K^T(t)K(t)\| &= \|(1+\alpha)^2\hat{P}(t-1)C^T(t)[(1+\alpha)C(t)\hat{P}(t-1)C^T(t) \\ &\quad + W_V(t)]^{-2} \cdot [(1+\alpha)C(t)\hat{P}(t-1)C^T(t) \\ &\quad + W_V(t)]^{-1}C(t)\hat{P}(t-1)\| \\ &\leq (1+\alpha)^2 N\|C(t)\hat{P}(t-1)C^T(t)\| \cdot \max\{\sigma_e^{-2}, \sigma_f^{-2}\} \\ &\leq (1+\alpha)^2 N^2\|C^T(t)C(t)\| \cdot \|\hat{P}(t-1)\|^2 \cdot \max\{\sigma_e^{-2}, \sigma_f^{-2}\} \\ &\leq \infty. \end{aligned} \quad (\text{B.12})$$

Using all the above results and inserting them into (B.8) we get that $\|\Sigma_\Delta(t)\| \leq C < \infty$. In order to prove the inequality (B.1) we observe that

$$\begin{aligned} E\{\|\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)\|\} &= E\{\sqrt{[\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)]^T [\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)]}\} \\ &\leq \sqrt{E\{[\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)]^T [\hat{\mathbf{X}}_{\text{opt}}(t) - \hat{\mathbf{X}}_{\text{opt}}(t-1)]\}} \\ &= \sqrt{\text{tr}\{\Sigma_\Delta(t)\}} \leq \sqrt{N\|\Sigma_\Delta(t)\|} < \infty \end{aligned} \quad (\text{B.13})$$

and the lemma is proved. ■

ACKNOWLEDGMENT

We thank Dr. Rami Atar for a fruitful conversation regarding the convergence analysis presented in this paper. We also thank the anonymous reviewers for their suggestions, which resulted in an improved paper.

This research was supported by the Israel Science Foundation, founded by the Israeli Academy of Sciences and Humanities, and by the Technion V.P.R. fund, N. Haar and R. Zinn Research Fund.

REFERENCES

1. A. Singh, *Optic Flow Computation: A Unified Perspective*, IEEE Computer Society Press, 1992.
2. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artif. Intell.* **17**, 1981, 185–204.
3. B. Lucas and T. Kanade, An iterative image registration technique with application to stereo vision, in *Proceedings, DARPA Image Understanding Workshop, 1981*, pp. 121–130.
4. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, Performance of optical flow techniques, *Int. J. Comput. Vision* **12**, 1994, 43–77.
5. M. Elad and A. Feuer, Super-resolution restoration of continuous image sequence using the LMS algorithm, in *Proceedings, IEEE 18th Conference, Tel-Aviv, Israel, 1995*.
6. A. Singh, Incremental estimation of image flow using the Kalman filter, *J. Vis. Commun. Image Represent.* **3**, 1992, 39–57.
7. T. M. Chin, W. C. Karl, A. M. Mariano, and A. S. Willsky, Square root filtering in time-sequential estimation of random fields, *Proc. SPIE* **1903**, 1993, 51–58.
8. T. M. Chin, *Dynamic Estimation in Computational Vision*, Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, 1992.
9. D. J. Fleet and K. Langley, Recursive filters for optical flow, *IEEE Trans. Pattern Anal. Mach. Intelligence* **17**, 1995, 61–67.
10. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, New York, 1986.
11. C. K. Chui and G. Chen, *Kalman Filtering*, Springer-Verlag, Berlin, New York, 1990.
12. D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1984.
13. L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic Press, San Diego, CA, 1981.
14. D. Terzopoulos, Image analysis using multigrid relaxation methods, *IEEE Trans. Pattern Anal. Mach. Intelligence* **8**, 1986, 129–139.
15. A. Feuer and E. Weinstein, Convergence analysis of LMS filters with uncorrelated Gaussian data, *IEEE Trans. Acouih. Speech Signal Process* **33**, 1985, 222–230.
16. R. H. Horn and C. J. Johnson, *Matrix Analysis*, Cambridge Univ. Press, Cambridge, UK, 1985.
17. A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, San Diego, 1970.
18. M. J. Black, Recursive non-linear estimation of discontinuous flow fields, in *The European Conference on Computer Vision, 1994*, pp. 139–145.
19. A. Blake and A. Zisserman, *Visual Reconstruction*, The MIT Press, Cambridge, MA, 1987.
20. S. Geman and D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images, *IEEE Trans. Pattern Analy. Mach. Intell.* **6**, 1984, 721–741.
21. J. Konard and E. Dubois, Bayesian estimation of motion vector fields, *IEEE Trans. Pattern Analy. Mach. Intell.* **14**, 1992, 910–927.
22. C. Cafforio and F. Rocca, Methods for measuring small displacements of television images, *IEEE Trans. Inform. Theory* **32**, 1976, 573–579.
23. G. R. Bergen, P. J. Burt, and S. Peleg, A three-frame algorithm for estimating two component image motion, *IEEE Trans. Pattern Analy. Mach. Intell.* **14**, 1992, 886–895.
24. J. Y. A. Wang and E. H. Adelson, Representing moving images with layers, *IEEE Trans. Image Process.* **3**, 1994, 625–638.
25. P. Migliorati and S. Tubaro, Multistage motion estimation for image interpolation, *Signal Process. Image Commun.* **7**, 1995, 187–199.
26. C. S. Fush and P. Maragos, Affine models for motion and shape recovery, in *SPIE Vis. Commun. Image Process. Boston, 1992*, pp. 120–134.
27. A. Amitay and D. Malah, Global motion estimation in image sequences of 3-D scenes for coding application, *Signal Process. Image Commun.* **6**, 1995, 507–520.



MICHAEL ELAD was born in Haifa, Israel, on December 1963. He received his B.Sc., M.Sc., and Ph.D. degrees from the electrical engineering department in the Technion, Haifa, Israel, in 1986, 1988, and 1997, respectively. Since January 1997 he has been employed by Hewlett Packard Laboratories-Israel, located in the Technion city in Haifa, as a research and development engineer. His current research interests include image reconstruction problems, adaptive filtering theory applied to image processing, and optimization theory in image processing, computer vision, and pattern recognition.



ARIE FEUER received his B.Sc. and M.Sc. from the Technion-Israel Institute of Technology, located in Haifa, Israel, in 1967 and 1973, respectively, and the Ph.D. degree from Yale University, New Haven, CT, in 1978. From 1967 to 1970 he was employed by Technomatic Israel, working on factory automation. From 1978 through 1983 he was with Bell Labs, Holmdel, NJ, studying telephone networks performance. Since 1983 he has been with the electrical engineering department in the Technion, Haifa, Israel. His research interests are in adaptive systems and sampled data systems both in control and in signal and image processing.