Super-Resolution Reconstruction of Image Sequences

Michael Elad, *Member*, *IEEE*, and Arie Feuer, *Senior Member*, *IEEE*

Abstract—In an earlier work, we have introduced the problem of reconstructing a super-resolution image sequence from a given low resolution sequence. We proposed two iterative algorithms, the R-SD and the R-LMS, to generate the desired image sequence. These algorithms assume the knowledge of the blur, the down-sampling, the sequences motion, and the measurements noise characteristics, and apply a sequential reconstruction process. It has been shown that the computational complexity of these two algorithms makes both of them practically applicable. In this paper, we rederive these algorithms as approximations of the Kalman filter and then carry out a thorough analysis of their performance. For each algorithm, we calculate a bound on its deviation from the Kalman filter performance. We also show that the propagated information matrix within the R-SD algorithm remains sparse in time, thus ensuring the applicability of this algorithm. To support these analytical results we present some computer simulations on synthetic sequences, which also show the computational feasibility of these algorithms.

Index Terms—Image restoration, super resolution, dynamic estimation, kalman filter, adaptive filters, recursive least squares (RLS), least mean squares (LMS), steepest descent (SD).

1 Introduction

In an earlier work, [1], we have introduced the problem of reconstructing an image sequence of improved resolution from a blurred, under-sampled and noisy measured sequence. Our approach to this problem can be described as the combination of ideas from two related problems: Super-resolution reconstruction of a single image (see [2], [3], [4], [5], [6], and [7]) and the restoration of an image sequence from blurred and noisy data (see e.g., [8], [9]).

In [1], we have presented two types of algorithms, both computationally feasible (assuming that the information matrix in the R-SD algorithm is sparse). These algorithms have been extensively tested on computer generated data and a sample results of these experiments have been presented. Following the work in [1], we have carried out thorough performance analysis of these algorithms which we wish to present here. This analysis includes two parts. The first part corresponds to the convergence properties of the proposed algorithms and the second part relates to the computational complexity of the R-SD algorithm.

We start our presentation with a brief description of the problem and the model used. Consider a sequence of images $\{\underline{Y}(t)\}$, each image is of $M \times M$ pixels, as our measured data. We wish to generate a sequence $\{\underline{X}(t)\}$ of images of higher resolution, each image of $L \times L$ $(L > M)$ pixels and of improved quality. For convenience of notation, all images will be presented as vectors, ordered column-wise lexicographically. Namely, we have $\underline{Y}(t) \in \mathbb{R}^{M^2}$ and $\underline{X}(t) \in \mathbb{R}^{L^2}$. At each time instant t we assume that the two images are related via the following equation:

$$\underline{Y}(t) = DH(t)\underline{X}(t) + \underline{N}(t), \tag{1.1}$$

which means that $\underline{X}(t)$ is blurred, decimated (namely, down sampled) and contaminated by additive noise, giving $\underline{Y}(t)$. $H(t)$ is the blur matrix which may be space and time variant, D the decimation matrix assumed constant, and $\underline{N}(t)$ is a zero mean Gaussian noise with $W^{-1}(t) = E\{\underline{N}(t)\underline{N}^T(t)\}$. Furthermore, we assume that the sequence $\{\underline{X}(t)\}$ satisfies the following equation:

$$\underline{X}(t) = G(t)\underline{X}(t-1) + \underline{V}(t). \tag{1.2}$$

The matrix $G(t)$ stands for the geometric warp between the images $\underline{X}(t)$ and $\underline{X}(t-1)$, and $\underline{V}(t)$ is the system noise. Assuming the typical optical flow model, most pixels in the image $\underline{X}(t)$ originate from pixels in the image $\underline{X}(t-1)$. Therefore, for each such pixel the corresponding row in the matrix $G(t)$ contains only one nonzero element at a position which reflects the address of the source pixel in the previous image and this entry equals one (assuming no change in gray level in the ideal images). The above description of $G(t)$ corresponds to a nearest-neighbor interpolation, other methods of interpolating can be used as well, resulting with a specific structure row stochastic matrix $G(t)$. The vector $\underline{V}(t)$ contains all the new pixels which do not originate from the previous image, namely, it represents the innovation sequence. For the sake of our analysis here, we assume this vector to be a zero mean Gaussian process with $Q^{-1}(t) = E\{\underline{V}(t)\underline{V}^T(t)\}$.

With (1.1) and (1.2), the problem we posed can be viewed as a state estimation problem and the most natural tool to consider is the Kalman filter. For the linear model we have assumed, the Kalman filter will provide the optimal solution in the Mean Square Error sense ([10], [11], [12], [13], [14], [15]). However, because of the large dimensions

- *M. Elad is with HP Laboratories-Israel (HPL-I), The Technion City, Haifa 32000, Israel. E-mail: elad@hpli.hpl.hp.com.*
- *A. Feuer is with the Department of Electrical Engineering, The Technion-Israel Institute of Technology, Haifa 32000, Israel. E-mail: feuer@ee.technion.ac.il.*

Manuscript received 2 Feb. 1998; revised 17 Dec. 1998.
Recommended for acceptance by R. Chellappa.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107619.

0162-8828/99/$10.00 © 1999 IEEE

involved— $L^2 \times L^2$—the computation and storage required to use the Kalman filter makes its use in our case impractical. Hence, our goal is to develop algorithms which approximate the Kalman filter as far as performance, but are significantly less demanding computation wise.

It is important to note that throughout this paper we assume that the matrices D, $H(t)$, $W(t)$, $G(t)$, and $Q(t)$, which define the state-space system, are known. Determining D is directly dictated by the resolution increase rate. The matrix $W(t)$ is typically assumed to be $\sigma_N^{-2} I$, which stands for assuming that $\underline{N}(t)$ is a homogeneous white noise. Determining $H(t)$ and σ_N can be based on a knowledge of the camera characteristics, or some estimation prestage of them. As to the matrix $G(t)$, this matrix can be obtained by a motion estimation algorithm. However, this estimation algorithm must be reliable, giving subpixel accuracy, if we are to obtain improved resolution results [1], [7]. The matrix $Q(t)$ can be created as a subresult of the motion estimation stage, by detecting the spatio-temporal zones with innovation, and assigning entries in $Q(t)$ appropriately. However, as we shall see in the next section, our algorithms propose a way to totally avoid dealing with this matrix.

When needed, (1.1) may include a regularization expression which represents prior knowledge on $\underline{X}(t)$ embedded in it. Typically, for restoration and reconstruction applications (see [1], [2], [3], [4], [5], [6], [7], [8], and [9]) the prior knowledge corresponds to some type of smoothness property the image $\underline{X}(t)$ has. A commonly used format, using the Laplacian operator denoted by S is presented by the equation:

$$\underline{0} = S\underline{X}(t) + \underline{U}(t), \tag{1.3}$$

which simply means that applying the Laplacian on the image $\underline{X}(t)$ should give zeros up to some additive noise, $\underline{U}(t)$, which we assume again, to be zero mean Gaussian with $R^{-1}(t) = E\{\underline{U}(t)\underline{U}^T(t)\}$. Then, (1.1) is replaced by:

$$\begin{bmatrix} \underline{Y}(t) \\ \underline{0} \end{bmatrix} = \begin{bmatrix} DH(t) \\ S \end{bmatrix} \underline{X}(t) + \begin{bmatrix} \underline{N}(t) \\ \underline{U}(t) \end{bmatrix} \tag{1.4}$$

or more compactly by:

$$\underline{Y}_A(t) = H_A(t)\,\underline{X}(t) + \underline{N}_A(t)$$

with

$$\begin{aligned} W_A^{-1}(t) &= E\Big\{\underline{N}_A(t)\,\underline{N}_A(t)^T\Big\} = E\left\{ \begin{bmatrix} \underline{N}(t) \\ \underline{U}(t) \end{bmatrix} \begin{bmatrix} \underline{N}(t) \\ \underline{U}(t) \end{bmatrix}^T \right\} \\ &= \begin{bmatrix} W^{-1}(t) & 0 \\ 0 & R^{-1}(t) \end{bmatrix}. \end{aligned} \tag{1.5}$$

The matrix $R(t)$ is typically chosen such that $R(t) = \beta I$. This way, the parameter β controls the spatial smoothness of the resulting images. The higher the value of β is , the smoother the output is.

This paper is organized as follows: In the next section we present the Kalman filter reconstruction equations, and then derive the R-SD and the R-LMS algorithms from these equations, as approximations. Section 3 deals with the convergence analysis of the two proposed algorithms, adopting a theoretic point of view. Computational complexity issues are discussed in Section 4 with an in depth analysis in Appendix B. Section 5 presents simulation results and Section 6 concludes this paper.

2 Dynamic Estimation Algorithms

2.1 Approximated Kalman Filter

We first present the original Kalman filter equations, and then turn to present the approximated methods. It is well-known that the state-vector in the above model can be considered as a Gaussian random process and, hence, it is fully described by its mean vector and covariance matrix. Kalman filter proposes two steps of propagating these descriptors in time - prediction and update steps [10], [11], [12]. Assuming that the mean-covariance pair are known for time $(t-1)$ denoted by $\langle \hat{\underline{X}}(t-1); \hat{P}(t-1) \rangle$, the prediction step applies equation (1.2) to predict the mean-covariance pair for time t. The prediction equations which produce the pair $\langle \tilde{\underline{X}}(t); \tilde{P}(t) \rangle$ are:

$$\langle \tilde{\underline{X}}(t); \tilde{P}(t) \rangle = \langle G(t)\hat{\underline{X}}(t-1); G(t)\hat{P}(t-1)G^T(t) + Q^{-1}(t) \rangle. \tag{2.1}$$

The update step applies the measurement (1.5) in order to propagate the mean-covariance pair to get $\langle \hat{\underline{X}}(t); \hat{P}(t) \rangle$. The update step equations are:

$$\begin{aligned} \langle \hat{\underline{X}}(t); \hat{P}(t) \rangle = \Big\langle \hat{P}(t) \Big[\tilde{P}^{-1}(t)\tilde{\underline{X}}(t) + H_A^T(t) W_A(t) \underline{Y}_A(t) \Big]; \\ \Big[\tilde{P}^{-1}(t) + H_A^T(t) W_A(t) H_A(t) \Big]^{-1} \Big\rangle. \end{aligned} \tag{2.2}$$

A convenient alternative to the propagation of the mean-covariance pair is the use of the information pair [11], [12]. The information pair is defined as:

$$\begin{aligned} \text{Prediction}: \Big\langle \tilde{\underline{Z}}(t); \tilde{L}(t) \Big\rangle &\triangleq \Big\langle \tilde{P}^{-1}(t)\tilde{\underline{X}}(t); \tilde{P}^{-1}(t) \Big\rangle \\ \text{Update}: \Big\langle \hat{\underline{Z}}(t); \hat{L}(t) \Big\rangle &\triangleq \Big\langle \hat{P}^{-1}(t)\hat{\underline{X}}(t); \hat{P}^{-1}(t) \Big\rangle. \end{aligned} \tag{2.3}$$

Since there is a one-to-one correspondence between these pairs and the pairs in (2.1) and (2.2), the information pair can serve as an alternative for the propagation of the Kalman filter in time. When propagating the information pair in time, the prediction equations become:

$$\begin{aligned} \Big\langle \tilde{\underline{Z}}(t); \tilde{L}(t) \Big\rangle = &\Big\langle \tilde{L}(t) G(t) \hat{L}^{-1}(t-1) \hat{\underline{Z}}(t-1); \\ &\Big[G(t)\hat{L}^{-1}(t-1)G^T(t) + Q^{-1}(t) \Big]^{-1} \Big\rangle. \end{aligned} \tag{2.4}$$

and the update equations become:

$$\begin{aligned} \Big\langle \hat{\underline{Z}}(t); \hat{L}(t) \Big\rangle = &\Big\langle \tilde{\underline{Z}}(t) H_A^T(t) W_A(t) \underline{Y}_A(t); \tilde{L}(t) \\ &+ H_A^T(t) W_A H_A(t) \Big\rangle \end{aligned} \tag{2.5}$$

and as can be seen, there is a duality between the two representations. For the mean-covariance pair, the prediction step is simple, while the update step is complicated, whereas for the information pair the reverse is true.

In our attempt to simplify the Kalman filter we use the information pair approach. We shall replace the term $Q^{-1}(t)$

(which is nonsingular) with an approximation term of the form:

$$\alpha(t)G(t)\hat{P}(t-1)G^T(t) = \alpha(t)G(t)\hat{L}^{-1}(t-1)G^T(t),$$

where $\alpha(t)$ is some positive scalar. One possibility is to choose this scalar so that:

$$Q^{-1}(t) \le \alpha(t)G(t)\hat{P}(t-1)G^T(t), \tag{2.6}$$

which is surely possible only if we assume that the matrices $G(t)$ and $\hat{P}(t-1)$ are nonsingular for all t. Clearly, the above approach means that, by using a bigger auto-correlation matrix, we assume a stronger system's noise $\underline{V}(t)$. Such an approach is known as adding pseudo-noise to the system's equation ([11], [12]) and is typically proposed for the treatment of model inaccuracies [11], [12]. The additional noise causes the Kalman filter to rely more on the measurements, rather than on the internal behavior of the assumed model. Different methods for the choice of $\alpha(t)$ can be suggested, such as searching for $\alpha(t)$ which minimizes the estimation error. Such choice of $\alpha(t)$ is surely better from the estimation performance point of view.

Using this approximation, the prediction equations of the information version Kalman filter become:

$$\begin{aligned}\left\langle \tilde{\underline{Z}}(t); \tilde{L}(t) \right\rangle &= \Big\langle \tilde{L}(t)G(t)\hat{L}^{-1}(t-1)\hat{\underline{Z}}(t-1); \\ &\qquad ; \Big[G(t)\hat{L}^{-1}(t-1)G^T(t) \\ &\qquad + \alpha(t)G(t)\hat{L}^{-1}(t-1)G^T(t)\Big]^{-1}\Big\rangle = \\ &= \Big\langle \frac{1}{1+\alpha(t)}G^{-T}(t)\hat{\underline{Z}}(t-1); \\ &\qquad \frac{1}{1+\alpha(t)}G^{-T}(t)\hat{L}(t-1)G^{-1}(t)\Big\rangle = \\ &= \Big\langle \frac{1}{1+\alpha(t)}F^T(t)\hat{\underline{Z}}(t-1); \\ &\qquad \frac{1}{1+\alpha(t)}F^T(t)\hat{L}(t-1)F(t)\Big\rangle,\end{aligned} \tag{2.7}$$

where we have denoted $F(t) = G^{-1}(t)$ (or its pseudo-inverse if $G(t)$ is singular). This is the backward motion matrix representing the motion operator from the current image $\underline{X}(t)$ to the previous one $\underline{X}(t-1)$. Therefore, this matrix has the same properties as the matrix $G(t)$—it is row stochastic matrix, it can represent any interpolation scheme, and so forth. We also denote $\lambda(t) = [1+\alpha(t)]^{-1}$. Using these notations, and the relations shown in (2.5), the entire Information Kalman filter propagation equations simplify to:

$$\begin{aligned}\hat{\underline{Z}}(t) &= \lambda(t)F^T(t)\hat{\underline{Z}}(t-1) + H_A^T(t)W_A(t)\underline{Y}_A(t) \\ \hat{L}(t) &= \lambda(t)F^T(t)\hat{L}(t-1)F(t) + H_A^T(t)W_A(t)H_A(t)\end{aligned} \tag{2.8}$$

and quite clearly, these recursive equations are much simpler to implement, compared to the previous ones.

An alternative approach would be to approximate the measurement noise auto-correlation matrix resulting in a similar simplification of the equations for the mean-covariance propagation. However, we feel that such approach is inferior to the one proposed here for two important reasons:

1. Typically, the measurements noise is known to be Gaussian additive white and homogeneous. On the other hand, the systems noise is not known and its auto-correlation matrix is a rough approximation to begin with. Thus, among the two, our approximation is bound to cause less damage.
2. As was said before, the alternative approach would have resulted in mean-covariance propagation equations. Our experiments indicate that, whereas the Kalman filter Information matrix is very sparse, the covariance matrix is generally more dense. Therefore, the alternative approach might still result in very heavy computational requirements.

Another more intuitive approach, which yields these two recursive equations, is to totally omit the system's noise by assuming $Q^{-1}(t) = 0$. In this case, by putting the system's equation into the measurement equation, the two state-space equations can be combined into an infinitely long sequence of equations of the form:

$$\begin{aligned}\underline{Y}_A(t-k) &= H_A(t-k)\underline{X}(t-k) + \underline{N}_A(t-k) \\ &= H_A(t-k)F(t-k+1)F(t-k+2) \\ &\quad \cdots F(t-1)F(t)\underline{X}(t) \\ &= H_A(t-k)\prod_{j=1}^{k}F(t-k+j)\underline{X}(t) + \underline{N}_A(t-k)\end{aligned} \tag{2.9}$$

for $k = 0, 1, 2, \cdots t$. We can now define a Weighted Least Squares (WLS) problem, where we search for the image $\underline{X}(t)$ which minimizes the function:

$$\begin{aligned}\varepsilon^2(t) = \sum_{k=0}^{\infty}\left[\prod_{j=0}^{k-1}\lambda(t-j)\right]\cdot \\ \left\|\underline{Y}_A(t-k) - H_A(t-k)\prod_{j=1}^{k}F(t-k+j)\underline{X}(t)\right\|^2_{W_A(t-k).}\end{aligned} \tag{2.10}$$

We give exponentially decaying weight to the error terms in the above squared error function in accordance with our wish to give smaller weight to distant measurements. Therefore, the exponential decaying weight operates as a forgetting mechanism, which controls the balance between allowing for changes in time while forcing smoothness. Taking the derivative of (2.10) and equating to zero, we get that the optimal estimated image should satisfy:

$$\begin{aligned}&\hat{L}(t)\hat{\underline{X}}(t) = \hat{\underline{Z}}(t) \\ &\text{where :}\quad \hat{\underline{Z}}(t) = \lambda(t)F^T(t)\hat{\underline{Z}}(t-1) + H_A^T(t)W_A(t)\underline{Y}_A(t) \\ &\qquad\qquad \hat{L}(t) = \lambda(t)F^T(t)\hat{L}(t-1)F(t) + H_A{}^T(t)W_A(t)H_A(t)\end{aligned} \tag{2.11}$$

and clearly, the above equations are identical to the ones obtained by the approximated Kalman filter.

To summarize this subsection, we have generated an approximation of the Kalman filter which basically involves solving a set of linear equations at each time instant. We name this algorithm, as presented in (2.11), the Pseudo-RLS algorithm. The reason for this name (see [1]) is the fact that its recursive equations resemble the RLS algorithm [15]. In the RLS algorithm, however, the inversion of $\hat{L}(t)$ is derived recursively using the matrix inversion lemma [15]. In our case, such approach is impossible because of the dimensions of the matrices involved and therefore, the name Pseudo-RLS.

2.2 The R-SD Algorithm

As we have seen above, the estimation problem is reduced to solving a very large set of linear equations (2.11). One immediate approach is, therefore, to apply some iterative algorithm, such as the Steepest-Decent, Conjugate-Gradient, Gauss-Seidel or others [13], [14], which solves these linear equations using only matrix-vector multiplications. Of course, to get the exact solution one would need to apply infinite number of these iterations. However, since the propagation of $\hat{L}(t)$ and $\hat{\underline{Z}}(t)$ is independent of the estimation $\hat{\underline{X}}(t)$, any error caused by limiting ourselves to a finite number of iterations at $(t-1)$ would not propagate to time t. Hence, we expect that, using a finite number of iterations would not cause divergence.

The first algorithm we propose consists of applying R iterations of the Steepest Descent (SD) algorithm at each time t. Hence, we name this algorithm R-SD. To complete the description of the algorithm we need to determine the initial value for the SD part at each t. A natural choice for the initialization is the vector $\hat{\underline{X}}_0(t) = G(t)\hat{\underline{X}}_R(t-1)$, where $\hat{\underline{X}}_R(t-1)$ is the result after the previous R iterations. This choice of initialization comes from the prediction step in the Kalman filter (2.1). The R-SD algorithm is then:

$$\begin{aligned}
\text{Initialization}: \quad & \hat{\underline{X}}_R(0) = E\{\underline{X}(0) \qquad \hat{L}(0) = \hat{P}^{-1}(0) \\
& \quad = \left[E\{\underline{X}(0)\underline{X}^T(0)\right]^{-1} \\
& \hat{\underline{Z}}(0) = \hat{L}(0)\hat{\underline{X}}_R(0) \\
\text{for } t \geq 1: \quad & \hat{\underline{Z}}(t) = \lambda(t)F^T(t)\hat{\underline{Z}}(t-1) + H_A^T(t)W_A(t)\underline{Y}_R(t) \\
& \hat{L}(t) = \lambda(t)F^T(t)\hat{L}(t-1)F(t) \\
& \qquad + H_A^T(t)W_A(t)H_A(t) \\
& \hat{\underline{X}}_0(t) = G(t)\hat{\underline{X}}_R(t-1) \\
& \hat{\underline{X}}_k(t) = \left[I - \mu\hat{L}(t)\right]\hat{\underline{X}}_{k-1}(t) \\
& \qquad + \mu\hat{\underline{Z}}(t) \qquad 1 \leq k \leq R.
\end{aligned} \tag{2.12}$$

Therefore, in order to apply the R-SD algorithm, we have to propagate first the approximated information pair $\langle\hat{\underline{Z}}(t); \hat{L}(t)\rangle$ in time, then use these terms in the recursive update equation of the estimated output vector $\hat{\underline{X}}_R(t)$ (the output of the above procedure after R iterations). Note that, compared to the Kalman filter estimator, the R-SD algorithm consist of two different layers of approximations. The first layer is the one presented in the previous subsection, namely, replacing the system's noise autocorrelation matrix. The other layer of approximation is the truncated application of the iterative algorithm, which surely would not give the exact solution per each temporal point.

The parameter μ in the above algorithm should be chosen so as to guarantee the Steepest Decent convergence ([13], [14]). Theoretically, μ should be determined by the largest eigenvalues of the matrix $\hat{L}(t)$. However, this value is complicated to obtain, and there are simpler ways to choose a reasonable μ. One safe approach is to use of the Normalized Steepest Decent (NSD) ([13], [14]), where the value of μ changes at each iteration as follows:

$$\begin{aligned}
& \mu_k(t) = \frac{\underline{E}_k^T(t)\underline{E}_k(t)}{\underline{E}_k^T(t)\hat{L}(t)\underline{E}_k(t)} \text{where } \underline{E}_k(t) = \hat{\underline{Z}}(t) - \hat{L}(t)\hat{\underline{X}}_{k-1}(t) \\
& \text{and } \hat{\underline{X}}_k(t) = \hat{\underline{X}}_{k-1}(t) + \mu_k(t)\underline{E}_k(t).
\end{aligned} \tag{2.13}$$

2.3 The R-LMS Algorithm

Taking the R-SD algorithm of the previous subsection and further approximating the information pair $\langle\hat{\underline{Z}}(t); \hat{L}(t)\rangle$ by the instantaneous values in (2.8), $H_A^T(t)W_A(t)\underline{Y}_A(t); H_A^T(t)W_A(t)H_A(t)$, respectively, we get an algorithm which resembles the LMS algorithm, and thus the name R-LMS algorithm. After some algebra we get:

$$\begin{aligned}
\text{Initialization}: \quad & \hat{\underline{X}}_R(0) = E\{\underline{X}(0)\} \\
\text{for } t \geq 1: \quad & \hat{\underline{X}}_0(t) = G(t)\hat{\underline{X}}_R(t-1) \\
& \hat{\underline{X}}_k(t) = \hat{\underline{X}}_{k-1}(t) + \mu H_A^T(t)W_A(t) \\
& \qquad \left[\underline{Y}(t) - H_A(t)\hat{\underline{X}}_{k-1}(t)\right] \quad 1 \leq k \leq R.
\end{aligned} \tag{2.14}$$

Clearly, the R-LMS algorithm is simpler than the R-SD algorithm, both in the computational and the memory requirements. We note that the R-LMS algorithm can also be obtained from the R-SD algorithm by assigning $\lambda(t) = 0$. Presumably, such value for $\lambda(t)$ means no temporal memory, and thus no temporal smoothness. Indeed, this is the case if infinitely many iterations are performed per each time point ($R \to \infty$). However, since R is finite and relatively small, temporal smoothness is not discarded, although it comes from different origin.

Here again, the step size μ should be chosen to guarantee the convergence of the LMS (see, for example, [13], [14], [15]). Here too, a possible choice is the normalized LMS resulting in:

$$\begin{aligned}
& \mu_k(t) = \frac{\underline{E}_k^T(t)\underline{E}_k(t)}{\underline{E}_k^T(t)H_A^T(t)W_A(t)H_A(t)\underline{E}_k(t)} \\
& \text{where } \underline{E}_k(t) = H_A^T(t)W_A(t)\left(\underline{Y}_A(t) - H_A(t)\hat{\underline{X}}_{k-1}(t)\right) \\
& \text{and } \hat{\underline{X}}_k(t) = \hat{\underline{X}}_{k-1}(t) + \mu_k(t)\underline{E}_k(t).
\end{aligned} \tag{2.15}$$

3 Convergence Properties of the Proposed Algorithms

In the previous section, we have presented again the two algorithms presented in [1], R-SD and R-LMS, for the recursive reconstruction of a super-resolution image sequence. In the discussion here, we emphasized the relationship of these algorithms, through the Pseudo-RLS, to the Kalman filter. This discussion provides the stepping stone for our analysis of these algorithms. In the sequel we use the Pseudo-RLS as an intermediate basis for performance comparison with the Kalman filter. Before we start our analysis let us recall the underlying model assumption for $\underline{X}(t)$, the required super-resolution image. Rewriting (1.2) and (1.5) we have:

$$\begin{aligned} &\underline{X}(t) = G(t)\underline{X}(t-1) + \underline{V}(t) \qquad && \underline{V}(t) \sim \mathbf{N}\{\underline{0},\, Q^{-1}(t)\} \\ &\underline{Y}_A(t) = H_A(t)\underline{X}(t) + \underline{N}_A(t) \qquad && \underline{N}_A(t) \sim \mathbf{N}\{\underline{0},\, W_A^{-1}(t)\}. \end{aligned} \tag{3.1}$$

3.1 The Pseudo-RLS Algorithm Analysis

We have introduced the Pseudo-RLS algorithm as a first approximation of the Kalman filter. Its estimate is the solution of the equation:

$$\hat{L}(t)\underline{\hat{X}}_{\mathrm{P-RLS}}(t) = \underline{\hat{Z}}(t) \tag{3.2}$$

where $\hat{L}(t)$ and $\underline{\hat{Z}}(t)$ satisfy the recursive updating formula:

$$\hat{L}(t) = \lambda(t)F^T(t)\hat{L}(t-1)F(t) + H_A^T(t)W_A(t)H_A(t) \tag{3.3}$$

$$\underline{\hat{Z}}(t) = \lambda(t)F^T(t)\underline{\hat{Z}}(t-1) + H_A^T(t)W_A(t)\underline{Y}_A(t) \tag{3.4}$$

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

Theorem 3.1.1. *The Pseudo-RLS algorithm for the model in (3.1) is in fact the Kalman filter when the model assumption is replaced by:*

$$\begin{aligned} &\underline{X}_1(t) = G(t)\underline{X}_1(t-1) + \underline{V}(t) \\ &\qquad\qquad \underline{V}_1(t) \sim \mathbf{N}\Big\{\underline{0}, Q^{-1}(t) = \alpha(t)G(t)\hat{L}^{-1}(t-1)G^T(t)\Big\} \\ &\underline{Y}_A(t) = H_A(t)\underline{X}_1(t) + \underline{N}_A(t) \qquad \underline{N}_A(t) \sim \mathbf{N}\Big\{\underline{0}, W_A^{-1}(t)\Big\} \end{aligned} \tag{3.5}$$

where:

$$\alpha(t) = \frac{1-\lambda(t)}{\lambda(t)} \tag{3.6}$$

Proof. The proof follows directly from our discussion in Section 2, where the Pseudo-RLS algorithm has been derived. There is a bootstrap behavior in the above model since the information matrix $\hat{L}(t-1)$, which is computed by the Kalman filter equations, serves for the definition of the model for time t. Thus, we actually have here a model which depends on the previous Kalman filter results. However, this bootstrap contains no noncasual operations, and therefore the above model contains no self contradictions. □

The above Theorem's significance can be seen in the following result:

Theorem 3.1.2. *The Pseudo-RLS algorithm guarantees an unbiased estimate of $\underline{X}(t)$ and if we choose $\lambda(t)$ such that:*

$$\lambda(t) \le \frac{1}{1 + \left\|Q^{-1}(t)\right\|_\infty \cdot \left\|\hat{L}(t-1)\right\|_\infty} \tag{3.7}$$

we are guaranteed to have:

$$\begin{aligned} \Sigma_{\mathrm{P-RLS}}(t) = E\Big\{&\Big[\underline{\hat{X}}_{\mathrm{P-RLS}}(t) - \underline{X}(t)\Big] \\ &\Big[\underline{\hat{X}}_{\mathrm{P-RLS}}(t) - \underline{X}(t)\Big]^T\Big\} \le \hat{L}^{-1}(t). \end{aligned} \tag{3.8}$$

Namely, the Pseudo-RLS estimation error covariance matrix is bounded by the matrix $\hat{L}^{-1}(t)$.

Proof. The fact that the estimate is unbiased follows directly from Theorem 3.1.1 and the Kalman filter properties. Based on the Kalman filter equations [10], [11], [12] we have:

$$\begin{aligned} \underline{\hat{X}}_{\mathrm{P-RLS}}(t) = \underline{\hat{X}}_1(t) = &\Big[I - K(t)H_A(t)\Big] \\ &G(t)\underline{\hat{X}}_1(t-1) + K(t)\underline{Y}_A(t), \end{aligned} \tag{3.9}$$

where $K(t)$ is defined as the Kalman filter gain matrix [10], [11], [12], [13], [14], [15]. For $\Sigma_{\mathrm{P-RLS}}(t)$ we have from (3.1), Theorem (3.1.1) and (3.9):

$$\begin{aligned} \Sigma_{\mathrm{P-RLS}}(t) = &\Big(I - K(t)H_A(t)\Big) \\ &\Big(G(t)\Sigma_{\mathrm{P-RLS}}(t-1)G^T(t) + Q^{-1}(t)\Big) \cdot \Big(I - K(t)H_A(t)\Big)^T \\ &+ K(t)W_A(t)K^T(t). \end{aligned} \tag{3.10}$$

Applying the Kalman filter equations for the approximated model, (3.5), gives the following relation:

$$\begin{aligned} \hat{L}^{-1}(t) = &\Big(I - K(t)H_A(t)\Big)\Big(G(t)\hat{L}^{-1}(t-1)G^T(t) \\ &+ \alpha(t)G(t)\hat{L}^{-1}(t-1)G^T(t)\Big) \cdot \Big(I - K(t)H_A(t)\Big)^T \\ &+ K(t)W_A(t)K^T(t). \end{aligned} \tag{3.11}$$

By subtracting (3.10) from (3.11) we get:

$$\begin{aligned} \hat{L}^{-1}(t) - \Sigma_{\mathrm{P-RLS}}(t) = &\Big(I - K(t)H_A(t)\Big) \\ &\Big[G(t)\Big(\hat{L}^{-1}(t-1) - \Sigma_{\mathrm{P-RLS}}(t-1)\Big)G^T(t)\Big] \cdot \\ &\cdot \Big(I - K(t)H_A(t)\Big)^T + \Big(I - K(t)H_A(t)\Big) \cdot \\ &\cdot \Big[\alpha(t)G(t)\hat{L}^{-1}(t-1)G^T(t) - Q^{-1}(t)\Big]\Big(I - K(t)H_A(t)\Big)^T. \end{aligned} \tag{3.12}$$

Choosing $\lambda(t)$ according to (3.7) satisfies the following inequality:

$$\alpha(t)G(t)\hat{L}^{-1}(t-1)G^T(t) \ge Q^{-1}(t) \tag{3.13}$$

since we require:

$$\begin{aligned} \alpha(t) = \frac{1-\lambda(t)}{\lambda(t)} \ge &\left\|G^{-1}(t)\right\|_\infty^2 \cdot \left\|\hat{L}(t-1)\right\|_\infty \\ &\cdot \left\|Q^{-1}(t)\right\|_\infty \ge \left\|\hat{L}(t-1)\right\|_\infty \cdot \left\|Q^{-1}(t)\right\|_\infty. \end{aligned} \tag{3.14}$$

Clearly, from equations (3.12) and (3.13) we get that if $\hat{L}^{-1}(0) - \Sigma_{\mathrm{P-RLS}}(0) \geq 0$, then:

$$\hat{L}^{-1}(t) - \Sigma_{\mathrm{P-RLS}}(t) \geq 0 \qquad \forall t \geq 0. \tag{3.15}$$

But the above requirement, $\hat{L}^{-1}(0) - \Sigma(0) \geq 0$, is readily satisfied by the initialization applied to the Pseudo-RLS algorithm, and thus the theorem is proven. □

We recognize that the Pseudo-RLS dose not provide a practical solution if directly applied since it requires the inversion of $\hat{L}(t)$ which is an impractical task. The purpose of the R-SD and the R-LMS algorithms is to supply an estimate of $\underline{\hat{X}}_{\mathrm{P-RLS}}(t)$. The following lemma presents an important property of the sequence $\{\underline{\hat{X}}_{\mathrm{P-RLS}}(t)\}_{t>0}$–bounded variation of the sequence. This property will be used in the analysis of the R-SD and the R-LMS estimation algorithms.

Lemma 3.1.1. *The sequence* $\{\underline{\hat{X}}_{\mathrm{P-RLS}}(t)\}_{t>0}$ *satisfies the following property:*

$$\Delta_D \overset{\Delta}{=} \sup_{t>1} E\left\{\left\|\underline{\hat{X}}_{\mathrm{P-RLS}}(t) - G(t)\underline{\hat{X}}_{\mathrm{P-RLS}}(t-1)\right\|\right\} < \infty. \tag{3.16}$$

Proof. The proof is given in Appendix A. □

In this subsection, we have shown that the Pseudo-RLS algorithm can actually be viewed as a Kalman filter for a modified model, with bounded estimation error, and unbiased estimation. However, the Pseudo-RLS algorithm is still far too computationally complex to be implemented. The R-SD and the R-LMS algorithms are practical approximations of the Pseudo-RLS. In the following subsections we will investigate their properties.

3.2 The R-SD Algorithm Analysis

The R-SD equations updating the estimate in time are given by:

$$\begin{aligned}&\underline{\hat{X}}_0(t) = G(t)\underline{\hat{X}}_R(t-1)\\ &\underline{X}_j(t) = \underline{\hat{X}}_{j-1}(t) + \mu\left[\underline{\hat{Z}}(t) - \hat{L}(t)\underline{\hat{X}}_{j-1}(t)\right] \quad \text{for } 1 \leq j \leq R.\end{aligned} \tag{3.17}$$

Combining these equations we get a single equation which relates the R-th result for time $(t-1)$ to the R-th result for time t:

$$\begin{aligned}\underline{\hat{X}}_R(t) =& \left[I - \mu\hat{L}(t)\right]^M G(t)\underline{\hat{X}}_R(t-1) + \mu\sum_{k=0}^{R-1}\left[I - \mu\hat{L}(t)\right]^k \underline{\hat{Z}}(t)\\ =& \left[I - \mu\hat{L}(t)\right]^R \left[G(t)\underline{\hat{X}}_R(t-1) - \underline{\hat{X}}_{\mathrm{P-RLS}}(t)\right]\\ &+ \underline{\hat{X}}_{\mathrm{P-RLS}}(t),\end{aligned} \tag{3.18}$$

where we have used equation (3.2), and the formula for a sum of geometric sequences [10].

Theorem 3.2.1. *Consider the R-SD algorithm as given in equation (3.18) with arbitrary initial conditions* $\underline{\hat{X}}_R(0), \underline{\hat{Z}}(0)$ *and* $\hat{L}(0) \geq 0$. *Let:*

$$0 < \mu < \frac{1}{\lambda_{\max}} \text{where } \lambda_{\max} = \sup_{t>0}\left\{\max_{1\leq k\leq N^2}\left\{\lambda_k\{\hat{L}(t)\}\right\}\right\}. \tag{3.19}$$

Then, $\exists 0 < \varepsilon < 1$ *such that:*

$$\begin{aligned}&\sigma_D(t) \overset{\Delta}{=} E\left\{\left\|\underline{\hat{X}}_R(t) - \underline{\hat{X}}_{\mathrm{P-RLS}}(t)\right\|\right\}\\ &\leq \frac{\Delta_D \cdot g[1-\varepsilon]^R}{1 - g[1-\varepsilon]^R} + [1-\varepsilon]^{Rt} \cdot g^t \cdot \sigma_D(0) \quad \forall t > 0,\end{aligned} \tag{3.20}$$

where $g = \sup_{t>0}\|G(t)\|_2 < \infty$.

Proof. Since $\hat{L}(t)$ is positive definite (because of the regularization term), the spectral radius of the matrix $M(t) = \left[I - \mu\hat{L}(t)\right]^R$ is:

$$\begin{aligned}\|M(t)\|_2 &= \left\|\left[I - \mu\hat{L}(t)\right]^R\right\|_2\\ &\leq \left\|I - \mu\hat{L}(t)\right\|_2^R = \left\|I - \mu U(t)\Delta(t)U^T(t)\right\|_2^R\\ &= \left\|U(t)\left[I - \mu\Delta(t)\right]U^T(t)\right\|_2^R = \left\|I - \mu\Delta(t)\right\|_2^R.\end{aligned} \tag{3.21}$$

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 (3.19) guarantees that $|\lambda\{M(t)\}| < 1$, and thus there exists $0 < \varepsilon < 1$ such that:

$$\left\|M(t)\right\|_2 \leq [1-\varepsilon]^R \quad \forall t > 0. \tag{3.22}$$

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

$$\begin{aligned}\lambda_{\max} &= \sup_{t>0}\left\{\max_{1\leq k\leq N^2}\left\{\lambda_k\{\hat{L}(t)\}\right\}\right\} < \infty;\\ \lambda_{\min} &= \inf_{t>0}\left\{\min_{1\leq k\leq N^2}\left\{\lambda_k\{\hat{L}(t)\}\right\}\right\} > 0.\end{aligned} \tag{3.23}$$

The minimum eigenvalue is indeed greater than zero since $\hat{L}(t)$ is positive definite for all t. The maximum eigenvalue can be bounded by:

$$\begin{aligned}&\lambda_{\max} < \sup_{t\geq 0}\ \mathrm{tr}\{\hat{L}(t)\} =\\ &\sup_{t\geq 0}\sum_{k=0}^{\infty}\lambda^k \mathrm{tr}\left\{H^T(t-k)W(t-k)H(t-k) + \beta S^T R(t)S\right\}.\\ &\underset{\substack{\mathrm{tr}\{H^T(t-k)W(t-k)H(t-k)\}\leq T_1\\ \mathrm{tr}\{\beta S^T R(t)S\}\leq T_2}}{\overset{\leq}{\uparrow}} \sum_{k=0}^{\infty}\lambda^k[T_1 + T_2] = \frac{T_1 + T_2}{1 - \lambda} < \infty\end{aligned} \tag{3.24}$$

Using (3.18) we get:

$$\begin{aligned}\underline{\hat{X}}_R(t) - \underline{\hat{X}}_{\mathrm{P-RLS}}(t) = \left[I - \mu\hat{L}(t)\right]^R \cdot\\ G(t)\left[\underline{\hat{X}}_R(t-1) - \underline{\hat{X}}_{\mathrm{P-RLS}}(t-1)\right]\\ - \left[\underline{\hat{X}}_{\mathrm{P-RLS}}(t) - G(t)\underline{\hat{X}}_{\mathrm{P-RLS}}(t-1)\right]\Big\}.\end{aligned} \tag{3.25}$$

Applying Euclidean norm and expectation to the above equality, using the triangle inequality, and inserting the result of Lemma 3.1.1 we get:

$$\sigma_D(t) \leq [1-\varepsilon]^R \Big\{ g \cdot \sigma_D(t-1) + \Delta_D \Big\} \tag{3.26}$$

and by applying the above inequality recursively we get the inequality in (3.20), which completes this theorem's proof. □

From the above theorem it is evident that the slower $\underline{\hat{X}}_{\text{P-RLS}}(t)$ is, the better the R-SD tracking capabilities are. Increasing R clearly improves the R-SD tracking performance, any desired accuracy can be reached with large enough R.

3.3 The R-LMS Algorithm Analysis

The R-LMS equations are given in (2.14). Using the notation:

$$\hat{l}(t) = H_A^T(t) W_A(t) H_A(t) \qquad \text{and} \qquad \underline{\hat{z}}(t) = H_A^T W_A(t) \underline{Y}_A(t) \tag{3.27}$$

and, in order to distinguish from the estimates gererated by the R-SD, we use $\underline{\hat{x}}$ instead of $\underline{\hat{X}}$ in (2.14). Then we have for the R-LMS:

$$\begin{array}{rcl} \underline{\hat{x}}_0(t) & = & G(t)\underline{\hat{x}}_R(t-1) \\ \text{and for} & & 1 \leq j \leq R \\ \underline{\hat{x}}_j(t) & = & \Big(I - \mu\hat{l}(t)\Big)\underline{\hat{x}}_{j-1}(t) + \mu\underline{\hat{z}}(t) \end{array} \tag{3.28}$$

In order to analyze the properties of the R-LMS we define the sequence χ_{Intra} through $\underline{\hat{z}}(t) = \hat{l}(t)\chi_{Intra}(t)$. Then, from (3.28) we get:

$$\begin{aligned} \underline{\hat{x}}_R(t) &= \Big[I - \mu\hat{l}(t)\Big]^R G(t)\underline{\hat{x}}_R(t-1) + \mu\sum_{k=0}^{R-1}\Big[I - \mu\hat{l}(t)\Big]^k \underline{\hat{z}}(t) \\ &= \Big[I - \mu\hat{l}(t)\Big]^R \Big[G(t)\underline{\hat{x}}_R(t-1) - \underline{\hat{\chi}}_{\text{Intra}}(t)\Big] + \underline{\hat{\chi}}_{\text{Intra}}(t). \end{aligned} \tag{3.29}$$

This equation is similar to the one given for the R-SD algorithm with $\hat{l}(t)$ replacing $\hat{L}(t), \underline{\hat{z}}(t),$ replacing $\underline{\hat{Z}}(t)$, and $\underline{\hat{\chi}}_{\text{Intra}}$ replacing $\underline{\hat{X}}_{\text{P-RLS}}(t)$. Note that the sequence $\{\underline{\hat{\chi}}_{\text{Intra}}(t)\}_{t\geq 1}$ is actually the result of an intra-frame approach for the reconstruction task, totally disregarding the temporal smoothness property (which explains the name chosen for these vectors). As such, showing that the R-LMS algorithm converges to this sequence is not a satisfactory result. Furthermore, we can see from (3.29) that for $R \to \infty$ we get $\underline{\hat{x}}_R(t) \to \underline{\hat{\chi}}_{\text{Intra}}(t)$. Thus, in contrast to the R-SD algorithm, increasing the number of iterations per time point does not necessarily improve the estimation performance. Actually, since the R-LMS can be obtained from the R-SD algorithm using $\lambda(t) = 0$ (this way removing the temporal smoothness factor), using small R values is the only mechanism to re-achieve the temporal memory.

Theorem 3.3.1. *Consider R-LMS algorithm as given in equation (3.27) with arbitrary initial condition $\underline{\hat{x}}_R(0)$. Let:*

$$0 < \mu < \frac{2}{\lambda_{\max}} \text{where } \lambda_{\max} = \sup_{t>0}\left\{ \max_{1\leq l\leq N^2}\left\{\lambda_k\{\hat{l}(t)\}\right\}\right\}. \tag{3.30}$$

Then, there exists $o < \varepsilon < 1$ and sufficiently large R such that $[1-\varepsilon]^R g < 1$ and then:

$$\begin{aligned} \sigma_f(t) &\overset{\triangle}{=} E\Big\{\Big\|\underline{\hat{x}}_R(t) - \underline{\hat{X}}_{\text{P-RLS}}(t)\Big\|\Big\} \\ &\leq [1-\varepsilon]^{\text{Rt}} \cdot g^t \cdot \sigma_f(0) + \frac{\Big[[1-\varepsilon]^{R-1}C_1 + C_2\Big]}{1 - [1-\varepsilon]^R g}\Delta_D \quad \forall t > 0 \end{aligned}, \tag{3.31}$$

where $g = \sup_{t>0}\|G(t)\|_2 < \infty$, and C_1, C_2 are two finite positive constants.

Proof. Using equations (3.2)–(3.4) and (3.27) we get that for the first iteration:

$$\begin{aligned} \underline{\hat{f}}_1(t) &\overset{\triangle}{=} \underline{\hat{x}}_1(t) - \underline{\hat{X}}_{\text{P-RLS}}(t) = G(t)\underline{\hat{x}}_R(t-1) - \underline{\hat{X}}_{\text{P-RLS}}(t) \\ &\quad + \mu\Big[\underline{\hat{Z}}(t) - \hat{L}(t)G(t)\underline{\hat{x}}_R(t-1)\Big] \\ &\quad - \lambda(t)\mu F^T(t)\Big[\underline{\hat{Z}}(t-1) - \hat{L}(t-1)F(t)G(t)\underline{\hat{x}}_R(t-1)\Big] \\ &= \Big[I - \mu\hat{L}(t)\Big]\Big[G(t)\underline{\hat{f}}_R(t-1) - \underline{\hat{\triangle}}_{\text{P-RLS}}(t)\Big] \\ &\quad + \lambda(t)\mu F^T(t)\hat{L}(t-1)F(t)G(t)\underline{\hat{f}}_R(t-1) \\ &= \Big[I - \mu\hat{l}(t)\Big]G(t)\underline{\hat{f}}_R(t-1) - \Big[I - \mu\hat{L}(t)\Big]\underline{\hat{\triangle}}_{\text{P-RLS}}(t), \end{aligned} \tag{3.32}$$

where we have used the relation $G(t) = F^{-1}(t)$ (assuming that $G(t)$ is nonsingular) and the definition $\underline{\hat{\triangle}}_{\text{P-RLS}}(t) \overset{\triangle}{=} \underline{\hat{X}}_{\text{P-RLS}}(t) - G(t)\underline{\hat{X}}_{\text{P-RLS}}(t-1)$. For the $2 \leq k \leq R$ other iterations:

$$\begin{aligned} &\underline{\hat{f}}_k(t) \overset{\triangle}{=} \underline{\hat{x}}_k(t) - \underline{\hat{X}}_{\text{P-RLS}}(t) = \underline{\hat{x}}_{k-1}(t) - \underline{\hat{X}}_{\text{P-RLS}}(t) \\ &+ \mu\Big[\underline{\hat{Z}}(t) - \hat{L}(t)\underline{\hat{x}}_{k-1}(t)\Big] \\ &- \lambda(t)\mu F^T(t)\Big[\underline{\hat{Z}}(t-1) - \hat{L}(t-1)F(t)\underline{\hat{x}}_{k-1}(t)\Big] \\ &\underline{\hat{f}}_{k-1}(t) - \mu\hat{L}(t)\underline{\hat{f}}_{k-1}(t) - \lambda(t)\mu F^T(t)\hat{L}(t-1)F(t) \\ &\Big[-\underline{\hat{\triangle}}_{\text{P-RLS}}(t) - \underline{\hat{f}}_{k-1}(t)\Big] = \Big[I - \mu\hat{l}(t)\Big]\underline{\hat{f}}_{k-1}(t) \\ &+ \mu\Big[\hat{L}(t) - \hat{l}(t)\Big]\underline{\hat{\triangle}}_{\text{P-RLS}}(t). \end{aligned} \tag{3.33}$$

Chaining these R equations together we get:

$$\begin{aligned} &\underline{\hat{f}}_R(t) = \Big[I - \mu\hat{l}(t)\Big]^R G(t)\underline{\hat{f}}_R(t-1) - \Big[I - \mu\hat{l}(t)\Big]^{R-1} \\ &\Big[I - \mu\hat{L}(t)\Big]\underline{\hat{\triangle}}_{\text{P-RLS}}(t) + \Big[I - \Big[I - \mu\hat{l}(t)\Big]^{R-1}\Big] \\ &\Big[\hat{l}^{-1}(t)\hat{L}(t) - I\Big]\underline{\hat{\triangle}}_{\text{P-RLS}}(t) = \Big[I - \mu\hat{l}(t)\Big]^R \\ &G(t)\underline{\hat{f}}_R(t-1) - \Big[I - \mu\hat{l}(t)\Big]^{R-1} \\ &\Big[\hat{l}^{-1}(t)\hat{L}(t) - \mu\hat{L}(t)\Big]\underline{\hat{\triangle}}_{\text{P-RLS}}(t) + \Big[\hat{l}^{-1}(t)\hat{L}(t) - I\Big]\underline{\hat{\triangle}}_{\text{P-RLS}}(t). \end{aligned} \tag{3.34}$$

Choosing μ according to (3.30) guaranties that there exists $0 < \varepsilon < 1$ such that $\|I - \mu\hat{l}(t)\| < 1 - \varepsilon\ \forall t$. Applying Euclidean norm and expectation to the above equality we get:

$$\sigma_f(t) \triangleq E\left\{\left\|\hat{\underline{f}}_R(t)\right\|\right\} \leq [1-\varepsilon]^R \cdot g \cdot \sigma_f(t-1) + \left[[1-\varepsilon]^{R-1}C_1 + C_2\right]\Delta_D, \tag{3.35}$$

where we have used the bounds:

$$\begin{aligned}\left\|\left(\hat{l}^{-1}(t) - \mu I\right)\hat{L}(t)\right\| &\leq \left\|\hat{l}^{-1}(t) - \mu I\right\| \cdot \left\|\hat{L}(t)\right\| \\ &\leq C_1\left\|I - \hat{l}^{-1}(t)\hat{L}(t)\right\| \\ &\leq 1 + \left\|\hat{l}^{-1}\right\| \cdot \left\|\hat{L}(t)\right\| \leq C_2.\end{aligned}$$

Assume that R is sufficiently large so that $[1-\varepsilon]^R g < 1$. Applying the above inequality recursively we get:

$$\begin{aligned}\sigma_f(t) \leq &[1-\varepsilon]^{\mathrm{Rt}} \cdot g^t \cdot \sigma_f(0) + \sum_{k=0}^{t-1}[1-\varepsilon]^{Rk} \\ &\cdot g^k\left[[1-\varepsilon]^{R-1}C_1 + C_2\right]\Delta_D \\ \leq &[1-\varepsilon]^{\mathrm{Rt}} \cdot g^t \cdot \sigma_f(0) \\ &+ \frac{1-[1-\varepsilon]^{Rt} \cdot g^t}{1-[1-\varepsilon]^R g}\left[[1-\varepsilon]^{R-1}C_1 + C_2\right]\Delta_D \\ \leq &[1-\varepsilon]^{\mathrm{Rt}} \cdot g^t \cdot \sigma_f(0) + \frac{\left[[1-\varepsilon]^{R-1}C_1 + C_2\right]}{1-[1-\varepsilon]^R g}\Delta_D,\end{aligned}$$

which is exactly the statement of the theorem. □

There is a major difference between the R-LMS result and the one obtained for the R-SD algorithm. In the bound obtained for the R-LMS algorithm, increasing R does not result in the estimation error converging to zero. Rather, we have shown in the above theorem that the R-LMS algorithm yields bounded estimation error, which resembles the results of the theoretic LMS [15]. A drawback in the obtained bound is that it does not reflect the already known fact that it is recommended not to perform many iterations in the R-LMS, since this way we get the intra-frame result without temporal smoothness.

Remark. Note that the $\mu_k(t)$ suggested for the R-SD in (2.13) (and similarly, for the R-LMS in (2.15) satisfies the conditions in Theorem 3.2.1 (Theorem 3.3.1).

4 Computational Complexity of the Proposed Algorithms

The computational complexity of the R-SD and the R-LMS algorithms is already described in details in [1]. One important problem in this context, however, that was not addressed in [1] will be presented here. The R-SD algorithm propagates the information pair in time. In order to assure that the required computations and memory of this algorithm are practical, we need to show that the information matrix remain sparse in time. Otherwise, we get that a very large matrix (of size $L^2 \times L^2$) becomes dense, thus requiring intolerable amount of memory and computations. The entire analysis which proves the sparseness property of the matrix $\hat{L}(t)$ is given in Appendix B. Here, we give a short intuitive discussion on the obtained result. As shown in (3.3), the information matrix is propagated through the following recursive equation:

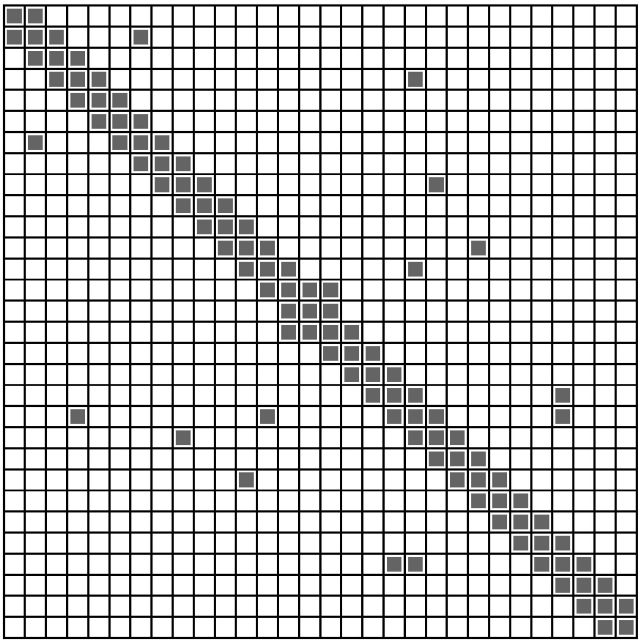


Fig. 1. The sparseness structure of the information matrix.

$$\hat{L}(t) = \lambda(t)F^T(t)\hat{L}(t-1)F(t) + H_A^T(t)W_A(t)H_A(t).$$

Let us assume that $\hat{L}(t-1)$ is sparse with a specific form: it is filled with nonzero entries in a central band around its main diagonal and additional nonzero entries are very sparsely spread outside this band. Fig. 1 presents such a structure.

Multiplying by $F(t)$ from both sides, $F^T(t)\hat{L}(t-1)F(t)$, it is shown that the overall number of nonzero entries in the resulting matrix reduces (see Appendix B). The central band now contains holes (zero entries), and the remaining matrix region remains sparse. The multiplication by $\lambda(t)$ does not affect the number of nonzero entries. The addition of the matrix $H_A^T(t)W_AH_A(t)$ refills the central band. This way we see that the first stage (multiplying $F^T(t)\hat{L}(t-1)F(t)$) removes nonzero entries, and the second stage (adding $H_A^T(t)W_A(t)H_A(t)$) regains them. At the steady state, we get that the density of the matrix $\hat{L}(t)$ fluctuate around a bounded very low value, which assures the sparseness of this matrix.

Two parameters govern the steady-state density bound of the matrix $\hat{L}(t)$. The first parameter is the width of the central nonzero band of the matrix $H_A^T(t)W_AH_A(t)$. The wider this band is, the denser the matrix $\hat{L}(t)$ becomes. The second parameter is the maximal displacement by the warp matrix $F(t)$. Clearly, if $F(t) = I$, the density of $\hat{L}(t)$ is the same as the one of the matrix $H_A^T(t)W_AH_A(t)$, and this case stands for the minimal possible density. The farther $F(t)$ spreads pixels from their original location (i.e., having large displacements), the denser the matrix $\hat{L}(t)$ becomes. More details about this behavior is presented in Appendix B.

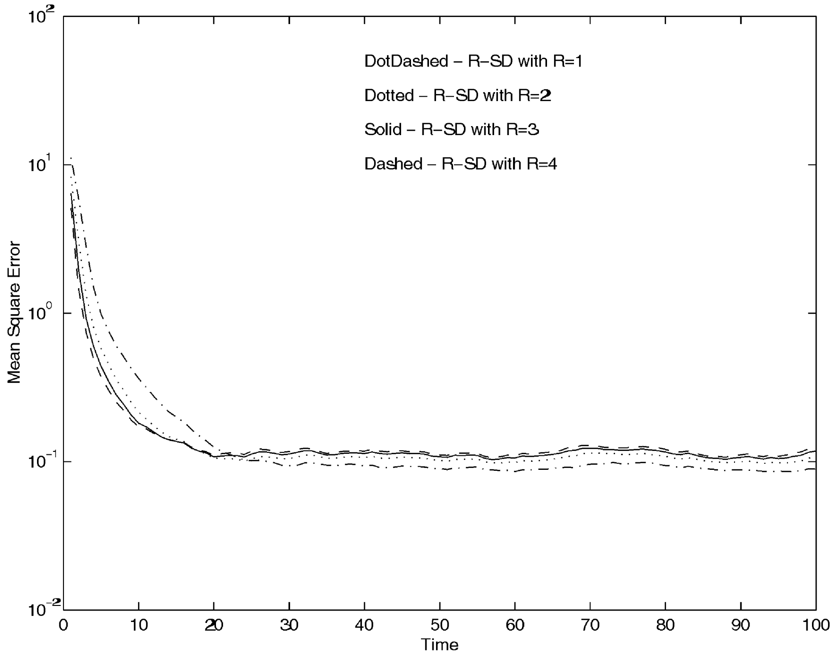


Fig. 2. The R-SD results for R=1,2,3, and 4.

5 Simulations and Analysis

In this section, we present two simulation experiments, Part I presents a comparison between the exact Kalman filter and it's various proposed approximations. Part II presents two image based examples, through which we demonstrate the feasibility of the proposed approach for super-resolution image sequence reconstruction.

5.1 Part I—The R-SD and the R-LMS vs. the Kalman Filter

In order to be able to compare the proposed algorithms with the exact Kalman filter, we have to use relatively low dimensional reconstruction problem. The test we present was done on a state vector containing 100 elements. State space equations were applied, with the following matrices:

- $G(t)$—One dimensional global shift by 1 sample to the left.

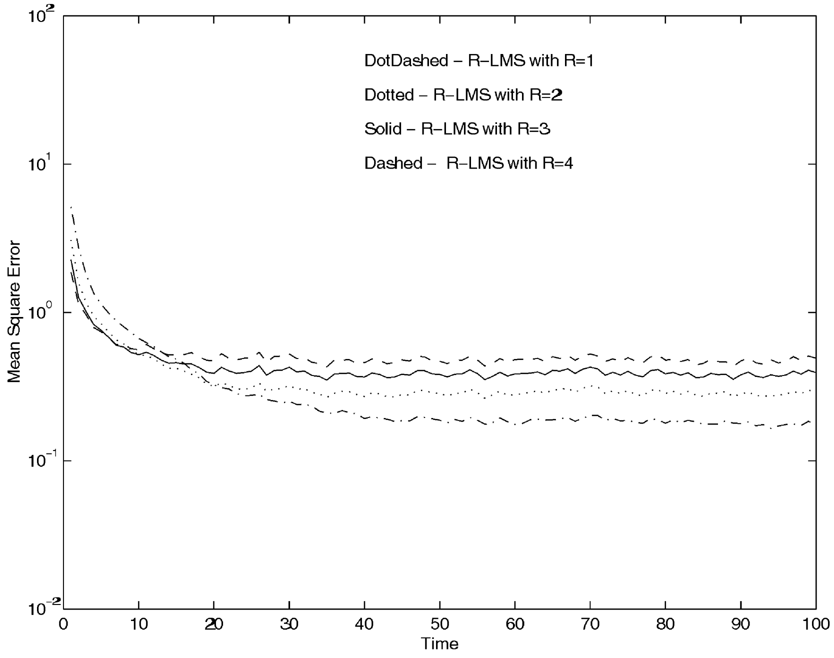


Fig. 3. The R-LMS results for R=1,2,3, and 4.

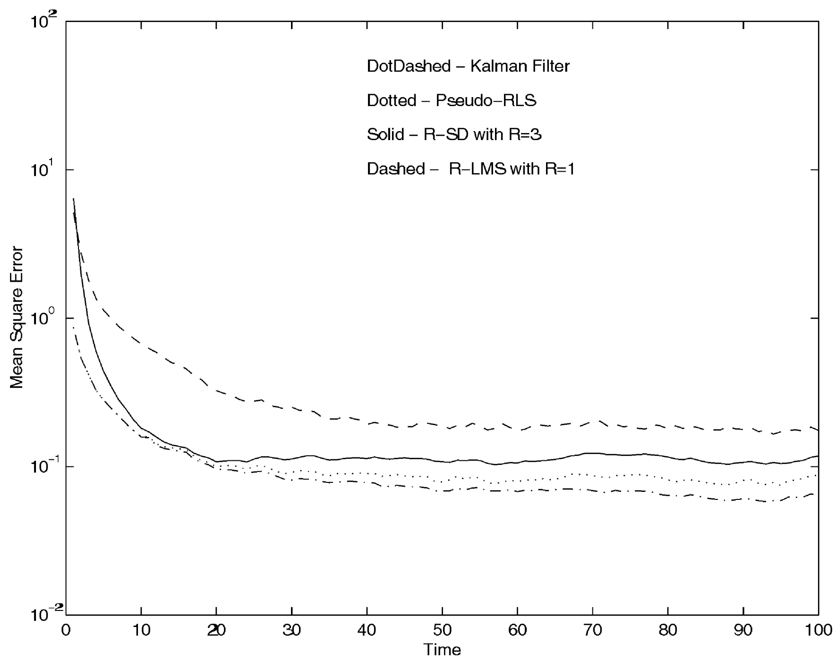


Fig. 4. Comparison between the various estimation schemes.

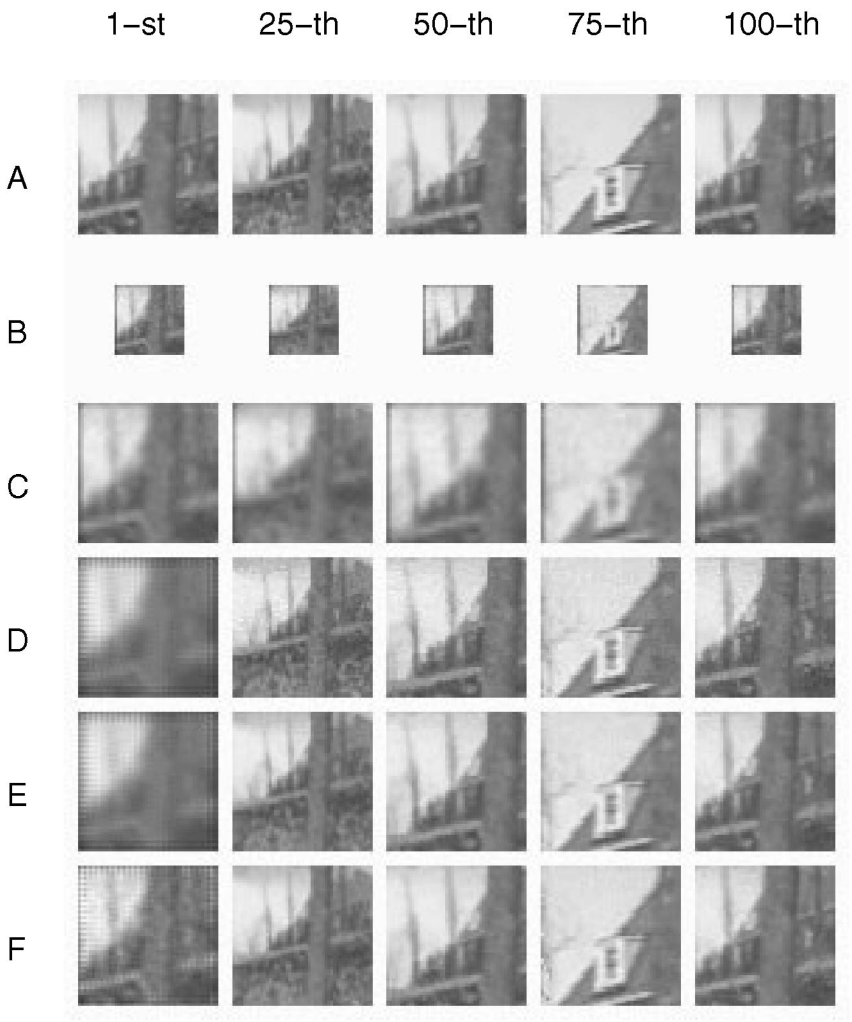


Fig. 5. The first sequence results.

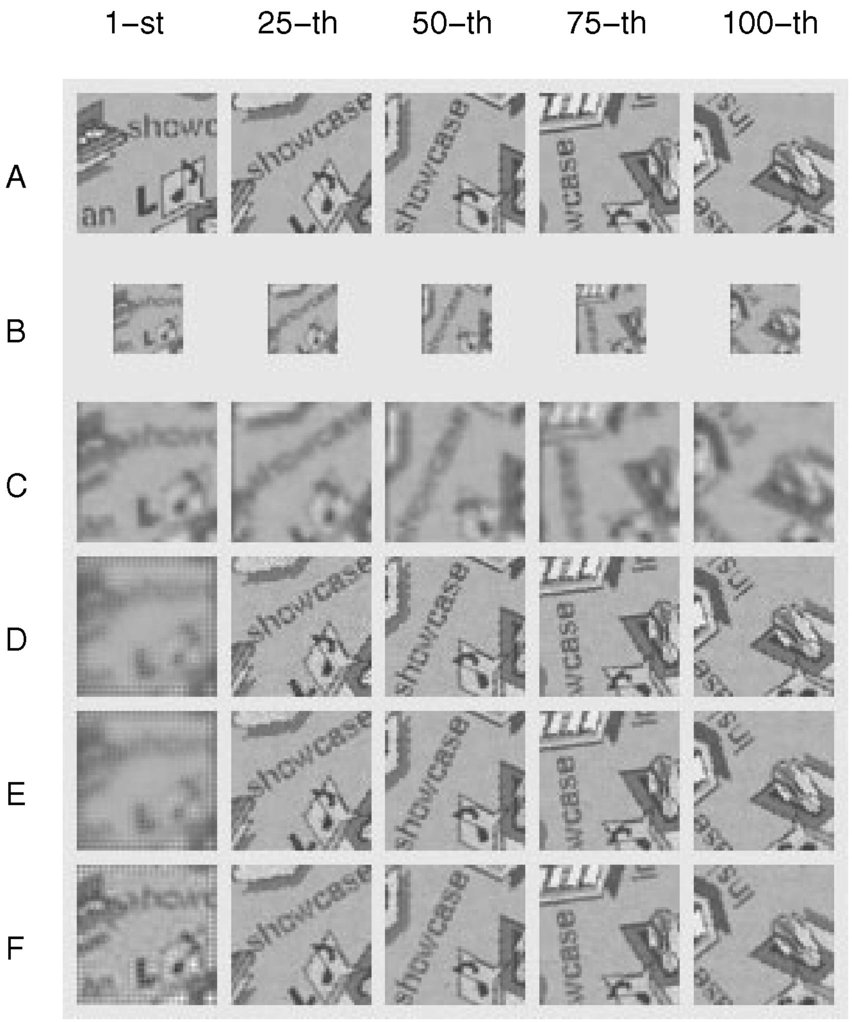


Fig. 6. The second sequence results.

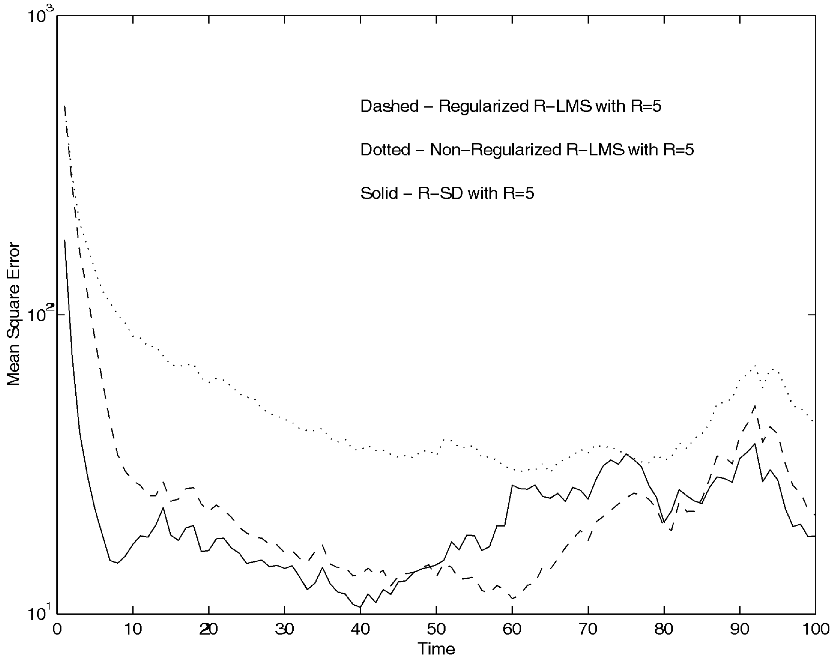


Fig. 7. The MSE for the R-SD and the R-LMS—results of the first sequence.

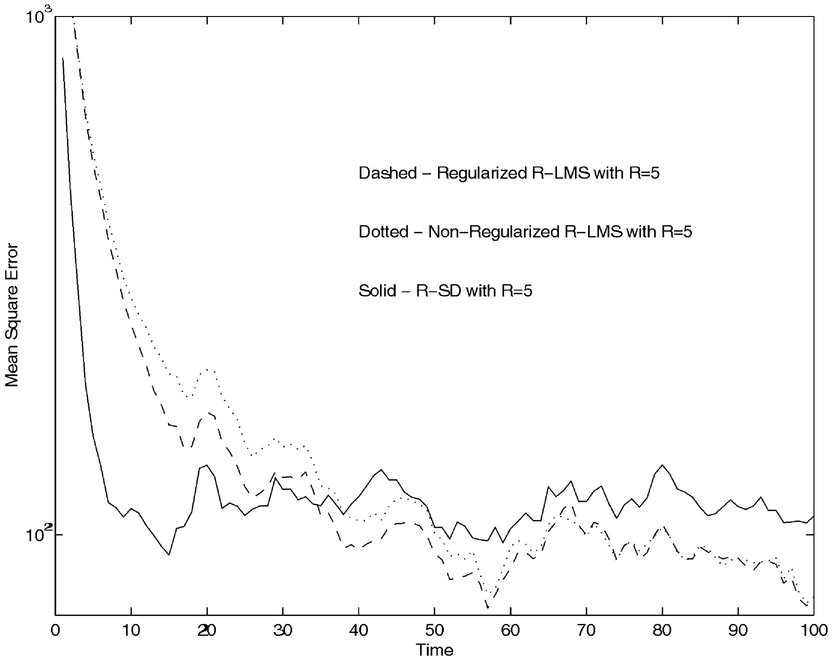


Fig. 8. The MSE for the R-SD and the R-LMS—results of the second sequence.

- $H_A(t)$—1D Linear space invariant blur with the kernel [0.2, 0.6, 0.2].
- $Q^{-1}(t)$—Diagonal matrix with random diagonal entries in the range [0.5-5].
- $W_A^{-1}(t)$—Diagonal matrix with random diagonal entries in the range [0.1-1].
- $\hat{P}(0)$—A random positive definite matrix with values in the range [0-5].
- $\hat{\underline{X}}(t)$—A random Gaussian vector with the above covariance matrix (+127).

With the matrices as defined above, 20 Monte-Carlo applications of the linear systems were used. For each such application, the Kalman filter, the Pseudo-RLS algorithm (with constant forgetting factor $= 0.95$), the R-SD algorithm (with $R = 1, 2, 3, 4$), and the R-LMS algorithm (for $R = 1, 2, 3, 4$) were tested. The following graphs present averages of the Mean-Squared Errors between the obtained estimations and the true state-vectors for 100 temporal points. Fig. 2 presents the R-SD results, Fig. 3 presents the R-LMS results, and Fig. 4 shows the comparison between all the discussed algorithms.

Several properties can be observed in the obtained results:

1. Raising the value of R in the R-SD and the R-LMS seems to improve the convergence in the initial part, but, on the other hand, causes a degradation in the steady state performance. Bearing in mind that these tests were applied on time-invariant linear system (all the above matrices are fixed in time), we can say that raising the value of R improves tracking performance.
2. As expected, the Kalman filter gives the best possible performance. The Pseudo-RLS is the next best method. The worst method is the R-LMS (R=1). However, for all the approximated methods (P-RLS, R-SD & R-LMS) the performance is quite close to the ideal Kalman filter results.

5.2 Part II—The Image Sequence Reconstruction Problem

The two tests presented in this part are based on two synthetic sequences, each containing 100 images of size $[50 \times 50]$ pixels. These sequences serve as the ideal images. The two measured image sequences were generated from these ideal sequences, by blurring each image using $[3 \times 3]$ uniform kernel, decimation using 2:1 decimation ration on each axis, and adding zero mean Gaussian white noise with $\sigma = 5$ (the dynamic range of the gray level in the images is 0-255). Thus the measured sequences contain 100 images of size $[25 \times 25]$ pixels each. These dimensions were chosen in order to shorten the simulations run-time and to overcome the memory limitations posed by MATLAB.

The R-SD and the R-LMS algorithms were applied using five iterations per each time point. In all cases, the initialization image at $t = 0$ was chosen to be a bilinear interpolated version of the first measured image. The applied regularization in all the tests was the Laplacian operator, using relative weight $\beta = 0.02$. In the reconstruction process, the true motion, blur and decimation operators were assumed known.

In Fig. 5, the results of the first test are given. In order to illustrate the temporal axis, the first, the 25th, the 50th, the 75th, and the 100th images of each sequence are given. The motion in this sequence consist of global zoom in and out and global translation motion. The given sequences are A: The ideal sequence; B: The measured sequence; C: Bilinear interpolation of the measurement sequence; D: The R-LMS results without regularization; E: The R-LMS results with regularization; and F: The R-SD results with regularization.

TABLE 1
Mean MSE and PSNR for the Two Sequences

	Sequence 1		Sequence 2	
	MSE	PSNR	MSE	PSNR
R-LMS	40.29	32.21 dB	92.58	28.48 dB
R-LMS with Reg.	22.49	34.97 dB	90.92	28.56 dB
R-SD	25.05	34.25 dB	112.43	27.64 dB

Similar to this figure, Fig. 6 presents the results for the second sequence. The motion in this sequence consist of global constant rotation while zooming in and out. Fig. 7 and Fig. 8 gives the MSE for these sequences, obtained from the R-SD and the R-LMS (with and without regularization) algorithms.

Peak-SNR values can be directly evaluated from the last two graphs by the relation $PSNR = 10\log_{10}\frac{255^2}{MSE}$. For completeness, the mean MSE, together with the mean PSNR values for $t = 51 - 100$ is given in Table 1 for the two sequences.

From these results we can conclude the following:

1. In both examples, using any of the two proposed reconstruction methods, there is a clear improvement both in the resolution quality and in the suppression of noise and blur degradation effects.
2. Regularization improves the performance in both examples.

6 Summary and Conclusions

In [1] we have presented two novel algorithms for image sequence reconstruction from down-sampled, blurred, and noisy measurements—the R-SD and the R-LMS. In this paper we analyze the performance of these algorithms. The presented analysis includes theoretic and simulation based results, demonstrating the strength of these dynamic estimation algorithms. A connection between the exact Kalman filter and the forgetting factor in the RLS algorithm is found and used. This way we relate the Kalman filter and the proposed algorithm's performances. A special and unique part of this paper is devoted to the question of the R-SD complexity (memory and computations). We have shown that the propagated (in time) information matrix remains sparse, with very low population rate. This result, which was obtained using probabilistic considerations, ensures that the R-SD is indeed a practical algorithm.

We should point out that the application of the proposed algorithms assumes exact knowledge of the *motion* between images and the *blur operators*. This is also the assumption underlying the analysis we perform. In practice these have to be estimated from the given data before or during the application of our algorithms. It is quite clear that these, the motion estimates in particular, strongly affect the performance of our algorithms and the quality of the resulting images. Further work, in an attempt to quantify these effects, combining analytical efforts with experiments in applying these algorithms to real data, is in progress.

As last point we mention that the obtained algorithms require the determination of several parameters such as λ—the forgetting factor, μ—the SD step size, and β—the smoothness strength. Currently, the method to choose these parameters' values is by trial and error. Further work is currently underway in an attempt to develop more precise recommendations for these parameters.

Appendix A

The Temporal Change of $\{\hat{\underline{X}}_{\mathrm{P-RLS}}(t)\}_{t>0}$

Lemma A.1. *The sequence* $\{\hat{\underline{X}}_{\mathrm{P-RLS}}(t)\}_{t>0}$ *satisfies the following property for all* $t > 1$*:*

$$\Delta_D \triangleq \sup_{t>1} E\left\{\left\|\hat{\underline{X}}_{\mathrm{opt}}(t) - G(t)\hat{\underline{X}}_{\mathrm{opt}}(t-1)\right\|\right\} < \infty. \qquad (A.1)$$

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

$$\Sigma_\Delta(t) \triangleq E\left\{\left[\hat{\underline{X}}_{\mathrm{opt}}(t) - G(t)\hat{\underline{X}}_{\mathrm{opt}}(t-1)\right]\left[\hat{\underline{X}}_{\mathrm{opt}}(t) - G(t)\hat{\underline{X}}_{\mathrm{opt}}(t-1)\right]^T\right\}, \qquad (A.2)$$

using the relations (3.9) and (3.1):

$$\hat{\underline{X}}_{P-RLS}(t) = \left[I - K(t)H_A(t)\right]G(t)\hat{\underline{X}}_{P-RLS}(t-1) + K(t)\underline{Y}_A(t) \qquad (A.3)$$

$$\begin{aligned}\underline{Y}_A(t) &= H_A(t)\underline{X}(t) + \underline{N}_A(t)\\ &= H_A(t)G(t)\underline{X}(t-1) + H_A(t)\underline{V}(t) + \underline{N}_A(t).\end{aligned} \qquad (A.4)$$

Combining these relations we get:

$$\begin{aligned}&\hat{\underline{X}}_{P-RLS}(t) - G(t)\hat{\underline{X}}_{P-RLS}(t-1)\\ &= K(t)\Big[H_A(t)G(t)\left[\underline{X}(t-1) - \hat{\underline{X}}_{P-RLS}(t-1)\right] + \underline{N}_A(t) + H_A(t)\underline{V}(t)\Big].\end{aligned} \qquad (A.5)$$

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

$$\begin{aligned}\Sigma_\Delta(t) =& K(t)H_A(t)\left[G(t)\Sigma(t-1)G^T(t) + Q^{-1}(t)\right]\\ &H_A^T(t)K^T(t) + K(t)W_A^{-1}(t)K^T(t),\end{aligned} \qquad (A.6)$$

where we have defined

$$\Sigma(t-1) \overset{\triangle}{=} E\left\{ \left[\underline{\hat{X}}_{P-RLS}(t-1) - \underline{X}(t-1)\right]\left[\underline{\hat{X}}_{P-RLS}(t-1) - \underline{X}(t-1)\right]^T \right\}.$$

Taking spectral norm on both sides of the (A.6) we get:

$$\begin{aligned}\left\|\Sigma_\Delta(t)\right\| \leq \left\|K(t)H_A(t)\right\|^2\left[\left\|G(t)\right\|^2 \cdot \left\|\Sigma(t)\right\| + \left\|Q^{-1}(t)\right\|\right] \\ + \left\|K(t)W_A^{-1}(t)K^T(t)\right\|.\end{aligned} \tag{A.7}$$

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

$$\begin{aligned}0 \leq I - K(t)H_A(t) \leq I \Rightarrow \left\|K(t)H_A(t)\right\| \\ = \left\|I - \hat{P}(t)\tilde{P}^{-1}(t)\right\| \leq 1 + \left\|\hat{P}(t)\tilde{P}^{-1}(t)\right\| \leq 2.\end{aligned} \tag{A.8}$$

The matrix $\Sigma(t-1)$ is the Pseudo-RLS estimation error covariance matrix. Based on the result of Theorem 3.1.2 we have that $\Sigma(t) \leq \hat{L}^{-1}(t) < \infty$. The norm $\|K(t)W_A(t)K^T(t)\|$ is bounded since:

$$\begin{aligned}\left\|K(t)W_A^{-1}(t)K^T(t)\right\| &\leq \mathrm{tr}\left\{K(t)W_A^{-1}(t)K^T(t)\right\} \\ &= \mathrm{tr}\left\{K^T(t)K(t)W_A^{-1}(t)\right\} \\ &\leq N\left\|K^T(t)K(t)W_A^{-1}(t)\right\| \\ &\leq N\left\|K^T(t)K(t)\right\| \cdot \left\|W_A^{-1}(t)\right\|.\end{aligned} \tag{A.9}$$

The norm of $K^T(t)K(t)$ is bounded by:

$$\begin{aligned}&\left\|K^T(t)K(t)\right\| = \left\|\Big(1+\alpha(t)\Big)^2\hat{P}(t-1)H_A^T(t)\right. \\ &\left[\Big(1+\alpha(t)\Big)H_A(t)\hat{P}(t-1)H_A^T(t) + W_A^{-1}(t)\right]^{-2} \\ &\cdot\left[\Big(1+\alpha(t)\Big)H_A(t)\hat{P}(t-1)H_A^T(t) + W_A^T(t)\right]^{-1} H_A(t)\hat{P}(t-1)\| \\ &\leq \Big(1+\alpha(t)\Big)^2 N^2\left\|H_A^T(t)H_A(t)\right\| \\ &\cdot \left\|\hat{P}(t-1)\right\|^2 \cdot \left\|W_A^{-1}(t)\right\| < \infty.\end{aligned} \tag{A.10}$$

Inserting all the above results into (A.7) and using the facts $\|G(t)\|, \|H_A(t)\|, \|W_A^{-1}(t)\|, \|Q^{-1}(t)\| < \infty$ we get that $\|\Sigma_\Delta(t)\| \leq C < \infty$. To complete the proof we observe that:

$$\begin{aligned}&E\left\{\left\|\underline{\hat{X}}_{\mathrm{opt}}(t) - \underline{\hat{X}}_{\mathrm{opt}}(t-1)\right\|\right\} \\ &= E\left\{\sqrt{\left[\underline{\hat{X}}_{\mathrm{opt}}(t) - \underline{\hat{X}}_{\mathrm{opt}}(t-1)\right]^T\left[\underline{\hat{X}}_{\mathrm{opt}}(t) - \underline{\hat{X}}_{\mathrm{opt}}(t-1)\right]}\right\} \\ &\leq \sqrt{E\left\{\left[\underline{\hat{X}}_{\mathrm{opt}}(t) - \underline{\hat{X}}_{\mathrm{opt}}(t-1)\right]^T\left[\underline{\hat{X}}_{\mathrm{opt}}(t) - \underline{\hat{X}}_{\mathrm{opt}}(t-1)\right]\right\}} \\ &= \sqrt{\mathrm{tr}\{\Sigma_\Delta(t)\}} \leq \sqrt{\mathbf{N}\|\sigma_\Delta(t)\|} < \infty.\end{aligned} \tag{A.11}$$

□

Appendix B—The Sparseness of the Information Matrix $\hat{L}(t)$

Definition B.1. *A Nearest Neighbor Warp (NNW) Central Band Limited Sparse (CBLS) matrix of width* p *is defined as an* $[N \times N]$ *matrix F which satisfies:*

1. $\forall\, 1 \leq i,j \leq N$, *and* $|i-j| > p$, $F_{[i,j]} = 0$.
2. $\forall 1 \leq i \leq N$ *there exists a unique value* $i - p \leq j(i) \leq i + p$ *such that* $F_{[i,j(i)]} = 1$.
3. $\forall 1 \leq i \leq N$ $\mathrm{Prob}\{j(i)\} = [1+2p]^{-1}$ *for*

   $$i - p \leq j(i) \leq i + p,$$

   and zero otherwise (uniform PDF), ignoring boundaries effects.
4. $\forall 1 \leq i_1 < i_2 \leq N$

   $$\mathrm{Prob}\{j(i_1), j(i_2)\} = \{j(i_1)\}\mathrm{Prob}\{j(i_2)\}$$

   (independence).

This definition implies that F has a specific row-stochastic form.

Definition B.2. *A Central Band Limited Sparse (CBLS) positive definite* $[N \times N]$ *matrix M which satisfies:*

1. $\forall 1 \leq i,j \leq N$, *and* $|i-j| > q$, $M_{[i,j]} = 0$
2. $\forall 1 \leq i,j \leq N$, *and* $|i-j| \leq q$, $M_{[i,j]} \neq 0$

is defined as a Full Positive Definite (FPD) CBLS matrix of width q.

Definition B.3. *For a given matrix* $A \in \mathbb{R}^{N \times M}$, $NNZ(A)$ *denotes the number of its nonzero entries. The density of A,* $d(A)$, *is defined as:*

$$d(A) = \frac{NNZ(A)}{NM}.$$

Clearly, $d\{A\} = \{A^T\}$ *and* $d\{A+B\} \leq d\{A\} + d\{B\}$.

The matrix sequence $\{\hat{L}(t)\}$ we are concerned with is generated via the following equation:

$$\hat{L}(t) = \lambda(t)F^T(t)\hat{L}(t-1)F(t) + M(t), \tag{B.1}$$

where $\{F(t)\}$ and $\{M(t)\}$ are given sequences of NNW CBLS and FPD CBLS matrices, respectively. We intend to show that, with a sparse initial matrix, the whole sequence remains sparse. It is clear that every $\hat{L}(t)$ has

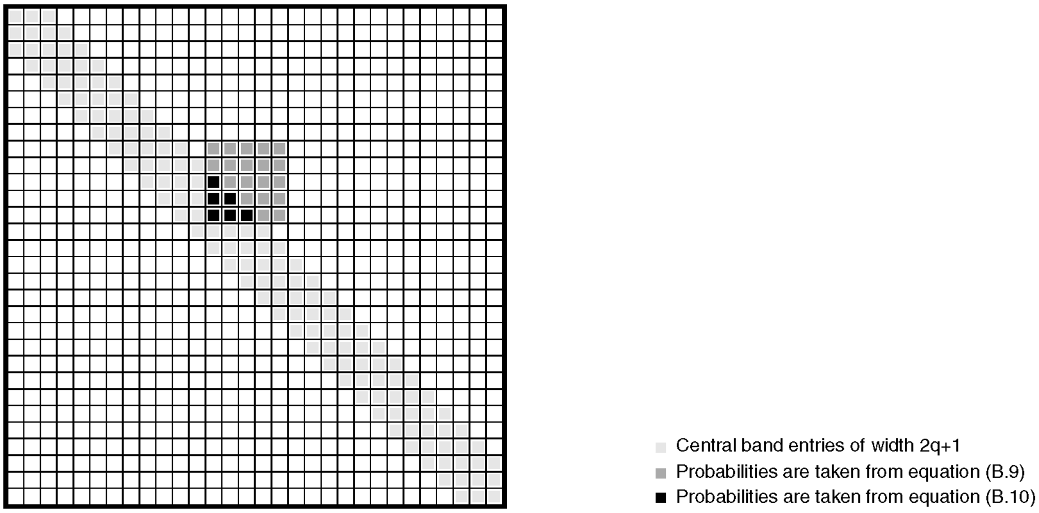

□ Central band entries of width 2q+1
▨ Probabilities are taken from equation (B.9)
■ Probabilities are taken from equation (B.10)

Fig. 9. The non-"1" multiplied probabilities in equation (B.8).

a center band of width $2q+1$ of nonzero entries, so we will concentrate on the remaining part of these matrices. To simplify the analysis, we will assume, in the transition from time $(t-1)$ to t according to (B.1), that the nonzero entries of $\hat{L}(t-1)$ *outside* the center band are uniformly distributed. It should be noted that this assumption results in a more conservative bound on the density at time t and, hence, at the steady state stage as well.

Theorem B.1. *Consider two matrix sequences: $\{F(t)\}$NNW CBLS matrices of width p and $\{M(t)\}$ FPD CBLS matrices of width q. Let the sequence $\{\hat{L}(t)\}$ be generated via (B.1) with some initial matrix $\hat{L}(0)$. Then, the density of the matrices $\hat{L}(t)$ as $t \to \infty$ is bounded by:*

$$d\Big\{\hat{L}(t)\Big\}\Big|_{t\to\infty} \cong \frac{2q+1}{N} + \frac{4p}{N}. \tag{B.2}$$

Proof. We investigate first how the density of $\hat{L}(t)$ is related to that of $\hat{L}(t-1)$ in (B.1). For that purpose, we consider the relationship $L_1 = F^T L F + M$, where M is an FPSD CBLS matrix of width q, and F is NNW CBLS matrix of width p. For L, we have $L(k,\ell) \neq 0$ for $|k-\ell| \leq q$. Then, with $d\{L\}$, the density of L, we assume:

$$\begin{aligned}&\text{Prob}\Big\{L(k,\ell)0, |k-\ell|>q\Big\}\\ &= \frac{N^2 \cdot d\{L\} - \Big[N(2q+1) - q(q+1)\Big]}{N^2 - \Big[N(2q+1) - q(q+1)\Big]}\\ &= P_0.\end{aligned} \tag{B.3}$$

Or, with $N \gg q$ we can write:

$$d\{L\} \cong P_0 + (1-P_0)\frac{2q+1}{N}. \tag{B.4}$$

By Definition B.1, the matrix F can be written as:

$$F = \sum_{i=1}^{N} E_i E_{j(i)}^T, \tag{B.5}$$

where E_i is the ith column of the Nth order identity matrix, and $j(i)$ is an integer random variable with uniform probability in the range $[i-p, i+p]$. Throughout the analysis in the sequel we assume *that a sum of values is zero only if all the values are zero*. With this, clearly:

$$L_1(k,\ell) \neq 0 \qquad \forall |k-\ell| \leq q. \tag{B.6}$$

Hence, we concentrate on $L_1(k,\ell)$ for $\forall |k-\ell| > q$. Let:

$$\begin{aligned}A &= \Big\{(k,\ell), 1 \leq k,\ell \leq N \big| |k-\ell| \leq q\Big\}\\ \bar{A} &= \Big\{(k,\ell), 1 \leq k,\ell \leq N \big| |k-\ell| > q\Big\}.\end{aligned} \tag{B.7}$$

Then, with $N \gg q$ we can write:

$$E\Big[d\{L_1\}\Big] \cong \frac{N^2 - \sum_{(k,\ell)\in\bar{A}} Q(k,\ell)}{N^2}, \tag{B.8}$$

where $Q(k,l) = \{L_1(k,\ell) = 0\}$. For $(k,\ell) \in \bar{A}$:

$$\begin{aligned}L_1(k,\ell) &= E_k^T L_1 E_\ell\\ &= E_k^t\left(\sum_{i=1}^{N} E_{j(i)} E_i^T\right)\left(\sum_{m=1}^{N}\sum_{r=1}^{N} L(m,r) E_m E_r^T\right)\\ &\quad \left(\sum_{s=1}^{N} E_s E_{j(s)}^T\right) E_\ell\\ &= \sum_{i=1}^{N}\sum_{m=1}^{N}\sum_{r=1}^{N}\sum_{s=1}^{N} L(m,r)\Big(E_k^T E_{j(i)}\Big)\Big(E_i^T E_m\Big)\\ &\quad \Big(E_r^T E_s\Big)\Big(E_{j(s)}^T E_\ell\Big).\end{aligned}$$

Since $E_i^T E_m$ only if $i = m$ and $E_r^T E_s = 1$ only if $r = s$, we get:

$$L_1(k,\ell) = \sum_{i=1}^{N}\sum_{s=1}^{N} L(i,s)\Big(E_k^T E_{j(i)}\Big)\Big(E_{j(s)}^T E_\ell\Big).$$

Now, clearly, $L_1(k,\ell) = 0$ if $L(i,s)(E_k^T E_{j(i)})(E_{j(s)}^T E_\ell) = 0$ for all $1 \leq i,s \leq N$. Defining:

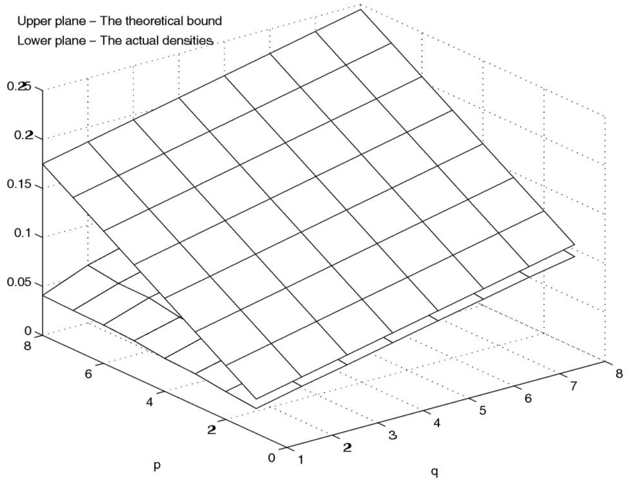


Fig. 10. The actual and the bound steady-state densities of 200×200 matrices $\hat{L}(t)$ for various values of p and q.

$$R_{[k,\ell]}(i,s) = \mathrm{Prob}\Big\{L(i,s)\Big(E_k^T E_{j(i)}\Big)\Big(E_{j(s)}^T E_\ell\Big) = 0\Big\},$$

we have $Q(k,\ell) = \prod_{i=1}^N \prod_{s=1}^N R_{[k,\ell]}(i,s)$. From (B.5):

$$\begin{aligned} R_{[k,\ell]}(i,s) &= \mathrm{Prob}\Big\{L(i,s) = 0 \text{ or } k \neq j(i) \text{ or } \ell(s)\Big\} \\ &= \mathrm{Prob}\Big\{L(i,s) = 0\Big\} \\ &\quad + \mathrm{Prob}\Big\{k \neq j(i)\Big\} + \mathrm{Prob}\Big\{\ell \neq j(s)\Big\} \\ &\quad - \mathrm{Prob}\Big\{L(i,s) = 0 \text{ and } k \neq j(i)\Big\} \\ &\quad - \mathrm{Prob}\Big\{L(i,s) = 0 \text{ and } \ell \neq j(s)\Big\} \\ &\quad - \mathrm{Prob}\Big\{k \neq j(i) \text{ and } \ell \neq j(s)\Big\} \\ &\quad + \mathrm{Prob}\Big\{L(i,s) = 0 \text{ and } k(i) \text{ and } \ell \neq j(s)\Big\}. \end{aligned}$$

Ignoring boundaries effects, we have:

$$\mathrm{Prob}\Big\{k \neq j(i)\Big\} = \begin{cases} 1 & |k-i| > p \\ \frac{2p}{2p+1} & |k-i| \leq p. \end{cases}$$

Since, for $(i,s) \in A, L(i,s) \neq 0,$ for $(i,s) \in A$ we have:

$$R_{[k,\ell]}(i,s) = \begin{cases} 1 - \frac{1}{(2p+1)^2} & |k-i| \leq p \text{ and } |\ell - s| \leq p \\ 1 & |k-i| > p \text{ or } |\ell - s| > p. \end{cases} \tag{B.9}$$

Similarly, for $(i,s) \in \bar{A}$:

$$R_{[\ell,\ell]}(i,s) = \begin{cases} 1 - \frac{P_0}{(2p+1)^2} & |k-i| \leq p \text{ and } |\ell - s| \leq p \\ 1 & |k-i| > p \text{ or } |\ell - s| > p. \end{cases} \tag{B.10}$$

Then, from all the above we get:

$$Q(k,\ell) = \prod_{i=1}^N \prod_{s=1}^N R_{[\ell,\ell]}(i,s) =$$
$$\begin{cases} \Big(1 - \frac{P_0}{(2p+1)^2}\Big)^{(2p+1)^2} & |k-\ell| > p+q \\ \Big(1 - \frac{P_0}{(2p+1)^2}\Big)^{(2p+1)^2 - f} \cdot \Big(1 - \frac{1}{(2p+1)^2}\Big)^f & p+q \geq |k-\ell| > q, \end{cases} \tag{B.11}$$

where $f = (2p + q - |k-\ell| + 1)(2p + q - |k-\ell| + 2)/2$. Observing (B.9) and (B.10), we note that most of these terms are "1" and there are exactly $(2p+1)^2$ exceptions. For $|k-\ell| > p+q$, all these exceptions fall in region $(i,s) \in \bar{A}$, which explains the upper term in (B.11). If $q < |k-\ell| \leq p+q$, some different probabilities are taken from region $(i,s) \in A$, as shown in Fig. 9.

The above result conforms with our expectation that, near the central band, the probability to get zero entry in L_1 is smaller. Returning to (B.8), we get:

$$\begin{aligned} E\Big[d\{L_1\}\Big] &= \frac{N^2 - \sum_{(k,\ell)\in\bar{A}} Q(k,\ell)}{N^2} \\ &\cong \frac{N^2 - \Big[N^2 - N(2q+4p+1)\Big]\Big(\frac{1-P_0}{(2p_1)^2}\Big)^{(2p+1)^2}}{N^2} \\ &\quad - \frac{2N\Big[1 - \frac{P_0}{(2p+1)^2}\Big]^{(2p+1)^2} \cdot \sum_{k=1}^{2p}\Big[\frac{(2p+1)^2 - 1}{(2p+1)^2 - P_0}\Big]^{\frac{k(k+1)}{2}}}{N^2}. \end{aligned}$$

Denoting:

$$\beta(p,P_0) = \sum_{k=1}^{2p}\left[\frac{(2p+1)^2 - 1}{(2p+1)^2 - P_0}\right]^{\frac{k(k+1)}{2}} \text{ and}$$

$$\varepsilon(p,P_0) = \left[1 - \frac{P_0}{(2p+1)^2}\right]^{(2p+1)^2} \geq 1 - P_0,$$

we get:

$$\begin{aligned} E\Big[d\{L_1\}\Big] &= \frac{N - \Big[N - (2q+1)\Big]\varepsilon(p,P_0)}{N} \\ &\quad + \frac{\Big[4p - 2\beta(p,P_0)\Big]\varepsilon(p,P_0)}{N} \\ &= E\Big[d\{L\}\Big] + \frac{\Big[N - (2q+1)\Big]\Big[1 - P_0 - \varepsilon(p,P_0)\Big]}{N} \\ &\quad + \frac{\Big[4p - 2\beta(p,P_0)\Big]\epsilon(p,P_0)}{N}. \end{aligned}$$

In steady state, we get that the density is constant which means that P_0 satisfies the following equation:

$$\Big[N - (2q+1)\Big]\Big[\varepsilon(p,P_0) - 1 + P_0\Big] = 2\Big[2p - \beta(p,P_0)\Big]\varepsilon(p,P_0).$$

By using (B.4):

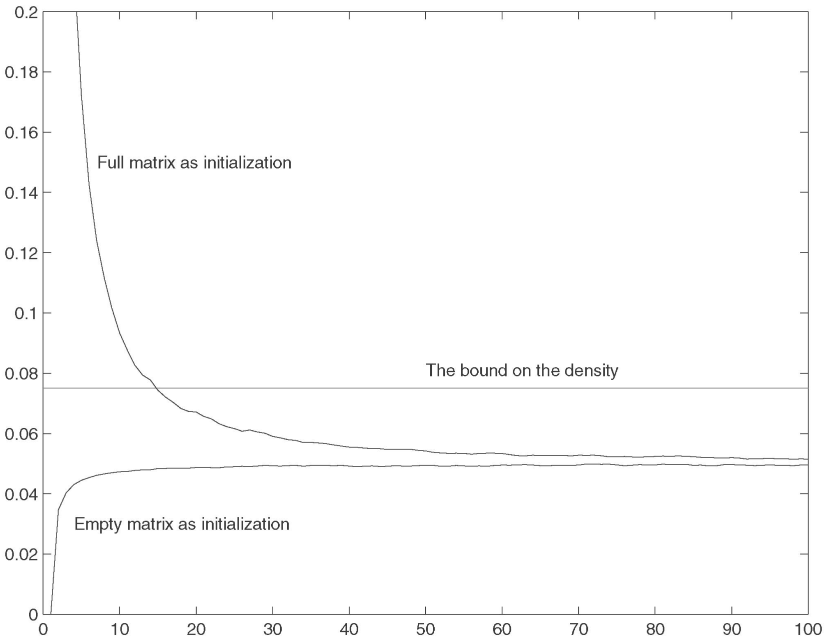
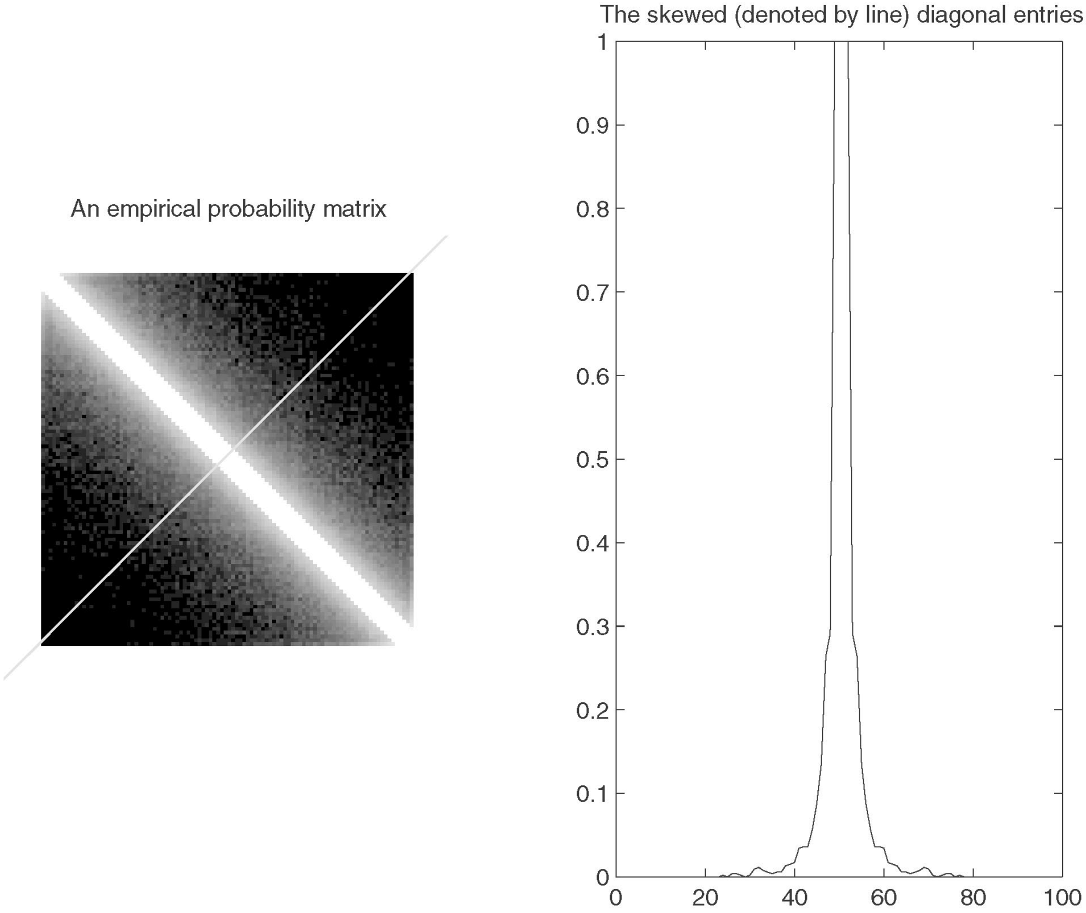


Fig. 11. The density as a function of time for full and empty initialization matrices.


An empirical probability matrix

The skewed (denoted by line) diagonal entries

Fig. 12. Empirical results for the probabilities to get a nonzero entry as a function of the position in the matrix $\hat{L}(t)$.

$$\Big[N-(2Q+1)\Big](1-P_0)=$$
$$N\Big[1-d\{L\}\Big]\Rightarrow\Big[N-(2q+1)\Big]\varepsilon(p,P_0)-N\Big[1-d\{L_1\}\Big]=$$
$$2\Big[2p-\beta(p,P_0)\Big]\varepsilon(p,P_0),$$

which in turn gives:

$$\begin{aligned} d\{L_1\} &= \frac{N-\varepsilon(p,P_0)\Big[N-(2q+1)\Big]+2\varepsilon(p,P_0)\Big[2p-\beta(p,P_0)\Big]}{N} \\ &\underset{\uparrow_{\beta(p,P_0)>0}}{\leq} \frac{N-\varepsilon(p,P_0)\Big[N-(2q+1)\Big]+4\epsilon(p,P_0)p}{N} \\ &\underset{\uparrow_{\varepsilon(p,P_0\cong 1-P_0}}{\cong} \frac{N-[1-P_0]\Big[N-(2q+1)\Big]+4[1-P_0]p}{N} \\ &\underset{\uparrow_{P_0\ll 1}}{\leq} \frac{N-\Big[N-(2q+1)\Big]+4p}{N}=\frac{2q+1}{N}+\frac{4p}{N}, \end{aligned} \tag{B.12}$$

which is the statement of this theorem. □

Simulations carried out show that indeed the obtained expression is a tight upper-bound on the actual resulting density of $\hat{L}(t)$. The actual values are very low, close to the density of the matrix M, thus ensuring that $\hat{L}(t)$ is indeed very sparse. Fig. 10 shows the values of the actual steady state densities for 200×200 matrices $\hat{L}(t)$ for various values of p and q (in the range 1 to 8). On the same axes, the value of the bound are given. As can be shown, the approximation of $\{\hat{L}(t)\}$ as a linear function of both q and p is very good and the approximation gives a very good guess of the actual values.

Fig. 11 shows $d\{\hat{L}(t)\}$ as a function of time for two initializations, empty and full matrices $\hat{L}(0)$. This 200×200 matrix corresponds to $p=2$ and $q=3$. The value of the bound is also given on the same axes—$(7+8)/200=0.075$.

Fig. 12 presents an empirical map of the probabilities of getting a nonzero value at each position for a 100×100 matrix. This probabilities map was obtained for $p=3$ and $q=4$ by creating 1,000 random generations of $\hat{L}(50)$ and averaging the probabilities. As can be seen, our assumption that the density is exponentially decaying as a function of the diagonal index is valid. Hence, our use of the uniform distribution leads indeed to an upper bound in steady state.

ACKNOWLEDGMENTS

The authors are grateful to Dr. Nir Cohen for the helpful discussions during the research. The authors wish to thank the anonymous reviewer for his/her thoroughness and suggestions, which helped in improving the paper. This work was supported in part by the Ollendorff Center Research Fund, by the The Israel Science Foundation founded by The Israel Academy of Sciences and Humanities, and by the Technion V.P.R. fund, N. Haar and R. Zinn Research Fund.

REFERENCES

[1] M. Elad and A. Feuer, “Super-Resolution Restoration of Continuous Image Sequence—Adaptive Filtering Approach,” *IEEE Trans. Image Processing,* vol. 8, no. 3, pp. 387-395, Marc. 1999.

[2] M. Irani and S. Peleg, “Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency,” *J. Visualization, Computers, and Image Recognition,* vol. 4, pp. 324-335, Dec. 1993.

[3] A.J. Patti, M.I. Sezan, and A.M. Tekalp, “High-Resolution Image Reconstruction from a Low-Resolution Image Sequence in the Presence of Time-Varying Motion Blur,” *Proc. Int'l Conf. Image Processing,* pp. 343-347, Austin, Tex., Nov. 1994.

[4] R.S. Schultz and R.L. Stevenson, “Extraction of High-Resolution Frames from Video Sequences,” *IEEE Trans. Image Processing,* vol. 5, pp. 996-1,011, June 1996.

[5] H. Shekarforoush, M. Berthod, J. Zerubia, and M. Werman, “Sub-Pixel Bayesian Estimation of Albedo and Height,” *Int'l J. Computer Vision,* vol. 19, pp. 289-300, 1996.

[6] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Henson, “Super-Resolved Surface Reconstruction from Multiple Images,” NASA, Ames Research Center, Technical Report FIA-94-12, 1994.

[7] M. Elad and A. Feuer, “Restoration of Single Super-Resolution Image from Several Blurred, Noisy and Down-Sampled Measured Images,” *IEEE Trans. Image Processing,* vol. 6, pp. 1,646-1,658, Dec. 1997.

[8] T. Patti, A.M. Tekalp, and A.M Sezan, “Image Sequence Restoration and De-Interlacing by Motion-Compensated Kalman Filter,” *Proc. SPIE—The Int'l Soc. Optical Eng.,* vol. 1,903, pp. 59-70, 1991.

[9] A.K. Katsagellos, J.N. Driessen, S.N. Efstratiadis, and L.J. Lagendijk, “Spatio-Temporal Motion Compensated Noise Filtering of Image Sequences,” *Proc. SPIE—Int'l Soc. Optical Eng.,* vol. 1,199, pp. 61-70, 1989.

[10] C.K. Chui and G. Chen, *Klaman Filtering.* Springer-Verlag, 1990.

[11] A.H. Jazwinski, *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

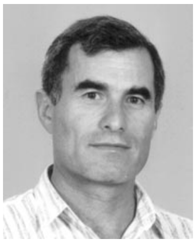
[12] P.S. Maybeck, *Stochastic Models, Estimation and Control,* vols. I-II. New York: Academic Press, 1979.

[13] L.A. Hageman and D.M. Young, *Applied Iterative Methods.* Academic Press, 1981.

[14] D.P. Bertsekas, *Nonlinear Programming.* Belmont, Mass.: Athena Scientific, 1995.

[15] S. Haykin, *Adaptive Filter Theory.* Prentice Hall, 1986.

Michael Elad received his BSc, MSc, and DSc degrees from the Electrical Engineering Department of the Technion-Israel Institute of Technology, Haifa, Israel, in 1986, 1988, and 1997, respectively. Since January 1997, he has been with Hewlett-Packard Laboratories-Israel, in Haifa, as a reserach and development engineer. His current research interests include image reconstruction problems, adaptive filtering theory applied to image processing, and numerical optimization theory in image processing, computer vision, pattern recognition, and machine learning. He is a member of the IEEE.



Arie Feuer received his BSc and MSc from the Technion-Israel Institute of Technology, Haifa, in 1967 and 1973, respectively, and the PhD degree from Yale University, New Haven, Connecticut, in 1978. From 1967 to 1970, he was with Technomatic Israel, working on factory automation. From 1978 through 1983, he was with Bell Labs, Holmdel, New Jersey, studying telephone network performance. Since 1983, he has been with the Electrical Engineering Department at the Technion. His research interests are in adaptive systems and sampled data systems, both in control and in signal and image processing. He is a senior member of the IEEE.