Image denoising using the Lyapunov equation from non-uniform samples

João M. Sanches and Jorge S. Marques

Instituto Superior Técnico / Instituto de Sistemas e Robótica
jmrs@isr.ist.utl.pt

Abstract. This paper addresses two problems: an image denoising problem assuming dense observations and an image reconstruction problem from sparse data. It shows that both problems can be solved by the Sylvester/Lyapunov algebraic equation. The Sylvester/Lyapunov equation has been extensively studied in Control Theory and it can be efficiently solved by well known numeric algorithms. This paper proposes the use of these equations in image processing and describes simple and fast algorithms for image denoising and reconstruction.

1 Introduction

Image reconstruction aims to recover images from a partial set of observations, corrupted by noise. Several methods have been proposed to deal with this problem e.g., Bayesian methods [1], wavelets [2–4], anisotropic diffusion [5, 6], level sets [7, 8]. Bayesian methods based on Markov random fields are among the most popular. They require the specification of a prior model and a sensor model which are chosen by the user or learned from the data. The image estimate is them obtained by minimizing an energy function with a very high number of variables or by solving a huge set of linear equations. These operations are very time consuming and they are carried out by suboptimal approaches (e.g., block processing) or by iterative algorithms [1].

This paper proposes an alternative approach based on the Sylvester/Lyapunov (SL) algebraic equation. We consider a denoising problem (all the pixels are observed and corrupted by Gaussian noise) and an image reconstruction problem (only sparse observations are available) and show that the original image can be estimated in both cases using the SL equation, in a very simple way. In the first case (denoising) the estimation requires the solution of a single SL equation. In the second case (sparse data), this equation appears embedded in an iterative scheme which updates its parameters. Convergence is obtained after a small number of iterations. Both procedures are easy to implement since there are efficient algorithms to solve the SL equation (e.g., in Matlab). Experimental tests are presented in this paper showing that the proposed algorithms are efficient and fast and lead to good reconstruction results.

The paper is organized as follows. In section 2 we address the image denoising problem. Section 3 addresses the reconstruction problem from sparse and

non uniform samples. This problem arises for instance in the case of freehand 3D ultrasound or confocal microscopy imaging. Section 4 shows examples using synthetic data and section 5 concludes the paper.

2 Image reconstruction and denoising

This section addresses an image denoising problem with the SL equation. Let $\mathbf{X}$ be a $m \times n$ matrix, representing an unknown image and let

$$\mathbf{Y} = \mathbf{X} + \mathbf{\Gamma} \tag{1}$$

be a noisy observation of $\mathbf{X}$ where $\mathbf{\Gamma} \sim N(0, \sigma^2 I)$ is a random error with Gaussian distribution. We wish to estimate $\mathbf{X}$ from the noisy image $\mathbf{Y}$. The matrix $\mathbf{X}$ is assumed to be a Markov Random Field (MRF) which means that it has a Gibbs distribution, $p(\mathbf{X}) = \frac{1}{Z} e^{-\alpha U(\mathbf{X})}$ where Z is a normalization constant, α is a regularization parameter and $U(\mathbf{X})$ is the *internal energy*. This energy has the form $U(\mathbf{X}) = \sum_{(p,q)\in V} v(x_p - x_q)$ where V is the set of all pairs of neighbors and $v(\tau)$ is a *potential function*.

In this paper we adopt a quadratic potential function, $v(\tau) = \tau^2$ and a 4-neighborhood system. Therefore, the internal energy is

$$U(\mathbf{X}) = tr\left[(\theta_v \mathbf{X})^T(\theta_v \mathbf{X}) + (\theta_h \mathbf{X}^T)^T(\theta_h \mathbf{X}^T)\right] = tr\left[\mathbf{X}^T(\theta_v^T\theta_v)\mathbf{X} + \mathbf{X}(\theta_h^T\theta_h)\mathbf{X}^T\right]$$

where θ_v and θ_h are $n \times n$ and $m \times m$ difference operators. $\theta_v \mathbf{X}$ is a $n \times m$ matrix with all vertical differences of neighboring pixels and $\theta_h \mathbf{X}^T$ is a $m \times n$-dimensional matrix with all the horizontal differences between neighboring pixels. Both matrices, θ_v and θ_h, have the following structure

$$\theta = \begin{pmatrix} 1 & -1 & 0 & ... & 0 & 0 & 0 \\ -1 & 1 & 0 & ... & ... & ... & 0 \\ 0 & -1 & 1 & ... & ... & ... & 0 \\ ... & ... & ... & ... & ... & 1 & 0 \\ 0 & 0 & 0 & ... & ... & -1 & 1 \end{pmatrix}. \tag{2}$$

If we adopt the MAP criterion, the estimate of $\mathbf{X}$ is

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} E(\mathbf{X}) \tag{3}$$

where the energy function $E(\mathbf{X}) = \log\left[p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})\right]$ is

$$E(\mathbf{X}) = tr\left[\frac{1}{2}(\mathbf{X}-\mathbf{Y})^T(\mathbf{X}-\mathbf{Y}) + \alpha\sigma^2\left(\mathbf{X}^T(\theta_v^T\theta_v)\mathbf{X} + \mathbf{X}(\theta_h^T\theta_h)\mathbf{X}^T\right)\right] \tag{4}$$

The solution of (3) is obtained by finding a stationary point of $E(\mathbf{X})$, which obeys the equation

$$\frac{\partial E(\mathbf{X})}{\partial \mathbf{x}_{ij}} = 0 \tag{5}$$

for $i = 1, ..., n$ and $j = 1, ..., m$. After straightforward manipulation (5) can be written as

$$\mathbf{X} - \mathbf{Y} + 2\alpha\sigma^2(A\mathbf{X} + \mathbf{X}B) = 0 \tag{6}$$

where $A = \theta_v^T\theta_v$ and $B = \theta_h^T\theta_h$ are $n \times n$ and $m \times m$ matrices respectively. This equation can be easily written in the form of a Sylvester equation

$$\mathcal{A}\mathbf{X} + \mathbf{X}\mathcal{B} + Q = 0 \tag{7}$$

where

$$\begin{aligned} \mathcal{A} &= I/2 + 2\alpha\sigma^2 A \\ \mathcal{B} &= I/2 + 2\alpha\sigma^2 B, \end{aligned} \tag{8}$$

$Q = -Y$ and I is the identity matrix. The reconstructed image can be obtained by solving the Sylvester equation. In the case of square images, $n = m$, $\mathcal{A} = \mathcal{B}$ are square and symmetric matrices and (7) becomes the well known Lyapunov equation.

The Lyapunov equation plays an important role in many branches of control theory namely, in stability analysis, optimal control and stochastic control [9]. There are efficient algorithm to solve the SL equation, some are include in mathematical packages (e.g., in Matlab) [10–12]. Therefore, we can easily use one of these algorithms to solve the image denoising problem.

It is important to stress that all the matrices used in (7) have low dimensions ($n \times m$, $n \times n$ or $m \times m$) while a direct implementation of the MAP denoising method involves the computation and inversion of a huge $nm \times nm$ matrix. This is not possible in practice and has to be solved by iterative algorithms [1].

3 Image Reconstruction from sparse data

This section considers image reconstruction from sparse observations. It is assumed that the observations are made at non-uniform positions in the image domain (see Fig. 1). We will address this problem assuming a continuous model for the image. Let $f : \Omega \to \mathbb{R}$, ($\Omega \subset \mathbb{R}^2$) be a continuous image to be estimated. We will assume that f of obtained by interpolating a discrete image $X = \{x_i\}$ using a set of interpolation functions. Therefore, $f(z)$ belongs to finite dimension linear space spanned by the set of interpolation functions: $\{\phi_1(z), \phi_2(z), ..., \phi_{nm}(z)\}$

$$f(z) = \sum_{i=1}^{nm} x_i\phi_i(z) \tag{9}$$

where x_i are the coefficients to be estimated, associated with each basis function. Herein, we assume that the basis functions, $\phi_i(z)$, are shifted versions of a known function $\phi(z)$ with finite supported, centered at the nodes of $n \times m$ 2D regular

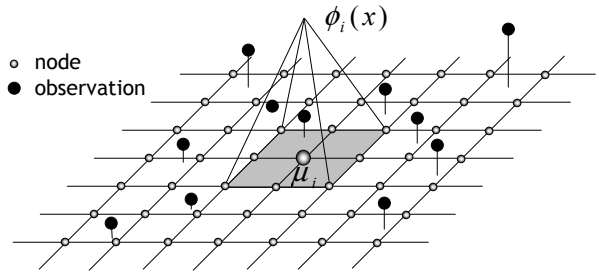

Fig. 1. 2D regular grid with a non-uniform sampling process.

grid, i.e., $\phi_i(z) = \phi(z - \mu_i)$, where μ_i is the location of the i-th node (see Fig. 1).

Consider now a L-dimensional column vector of noisy observations, $\mathbf{y} = \{y_i\}$ taken at non-uniform locations $\mathbf{z} = \{z_i\}$ given by $\mathbf{y} = F(\mathbf{z}) + \mathbf{n}$ where $F(\mathbf{z}) = \{f(z_1), f(z_2), ..., f(z_L)\}^T$ and $\mathbf{n} = \{n_1, n_2, ..., n_L\}^T$ are L-dimensional column vectors with $n_i \sim \mathcal{N}(0, \sigma^2)$ being a Gaussian random variable. As before, the MAP estimate of $\mathbf{X}$ is given by the minimization of the energy function (2). Assuming that the observations are conditionally independent, the data fidelity term is

$$E_Y = -\log(p(\mathbf{y}|\mathbf{X})) = \log\left[\prod_{i=1}^{L}(p(y_i|f(z_i))\right] = \frac{1}{2}\sum_{i=1}^{L}(f(z_i) - y_i)^2. \quad (10)$$

and the partial derivative of E_Y with respect to each unknown, x_p, is

$$\begin{aligned}\frac{\partial E_Y}{\partial x_p} &= \sum_{i=1}^{L}(f(z_i) - y_i)\phi_p(z_i) = \sum_{i=1}^{L}\left[\sum_{k=1}^{nm} x_k\phi_k(z_i) - y_i\right]\phi_p(z_i) \quad (11)\\ &= \sum_{k=1}^{nm}(x_k h_{pk}) - d_p,\end{aligned}$$

where $f(z)$ was replaced by(9), $h_{pk} = \sum_{i\in V(p,k)}\phi_k(z_i)\phi_p(z_i)$, $d_p = \sum_{i\in V(p)} y_i\phi_p(z_i)$, $V(p)$ is the neighborhood of x_p and $V(p,k)$ is the intersection of $V(p)$ with $V(k)$. This derivative can be efficiently computed since the sum has only 8 terms different from zero, corresponding to the 8 neighbors of pixel p. Furthermore, it can be expressed as a 2D convolution, of x_p with a space varying impulse response, $h_{p,k}$ minus a difference term d_p, $\partial E_Y/\partial x_p = h_{p,k} * x_p - d_p$. This can be written in a compact way as,

$$\frac{\partial E_Y}{\partial \mathbf{X}} = H * \mathbf{X} - D \quad (12)$$

where $*$ denotes the convolution of the discrete image $\mathbf{X}$ with the space varying impulse response H. In practice this amounts to convolving $\mathbf{X}$ with the following

space varying mask

$$H(i,j) = \begin{pmatrix} h_{i,j,i-1,j-1} & h_{i,j,i-1,j} & h_{i,j,i-1,j+1} \\ h_{i,j,i,j-1} & h_{i,j,i,j} & h_{i,j,i,j+1} \\ h_{i,j,i+1,j-1} & h_{i,j,i+1,j} & h_{i,j,i+1,j+1} \end{pmatrix} \tag{13}$$

which can be computed at the beginning once we know the sampling points. There are several equal coefficients among these masks due to symmetry. Approximately only half of the coefficients have to be computed. The derivative of the energy is therefore,

$$\frac{dE}{d\mathbf{X}} = 0 \Rightarrow H * \mathbf{X} - D + 2\alpha\sigma^2 \left[A\mathbf{X} + \mathbf{X}B\right] = 0 \tag{14}$$

This equation, as before, can be written as a SL equation,

$$\mathcal{A}\mathbf{X} + \mathbf{X}\mathcal{B} + Q(\mathbf{X}) = 0 \tag{15}$$

where $\mathcal{A}$ and $\mathcal{B}$ are obtained from (8) and $Q = H * \mathbf{X} - \mathbf{X} - D$.

In this case, $Q(\mathbf{X})$ depends on $\mathbf{X}$ which means that the solution can not be obtained in one iteration. Therefore, an iterative algorithm is proposed. In each iteration a new estimate of $\mathbf{X}$, $\mathbf{X}_t$ is obtained from the previous estimate of $\mathbf{X}$, $\mathbf{X}_{t-1}$ by solving

$$\mathcal{A}\mathbf{X} + \mathbf{X}\mathcal{B} + Q(\mathbf{X}_{t-1}) = 0 \tag{16}$$

where $Q = [H(p) - \delta(p)] * \mathbf{X}_{t-1} - D$. Equation (16) is iteratively solved until convergence is achieved.

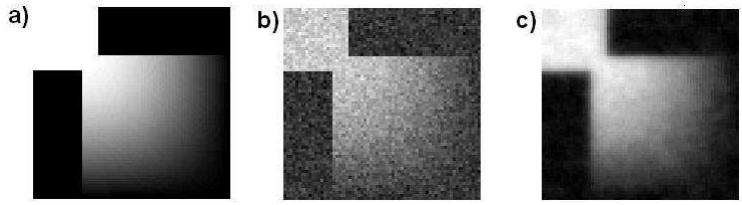

Fig. 2. Image denoising using a synthetic 64×64 pixel image. a)Original, b)noisy with zero mean additive Gaussian noise ($\sigma = 0.25$), c)denoised image. Processing times 530.12 and 5.31 seconds using the VECT and the LYAP methods respectively.

When there is no overlap of basis functions, which is usually the case in image processing (square pixels), all coefficients $h_{kp} = 0$ for $k \neq p$. In this case, the computation of the term $H(p) * \mathbf{X}$ is replaced by a simpler computation of $H.\mathbf{X}$ where "." denotes the Hadamard product and $H_p = \sum_{i \in V(p)} \phi_p^2(z_i)$. Therefore, in this case,

$$[Q]_p = \sum_{i \in V(p)} \left[\phi_p^2(z_i)\right] - 1 - d_p \tag{17}$$

where p is a bi-dimensional index. In image denoising problems with regular grids and without interpolation, there is only one observation per unknown. In this case, $H(p) = \delta_p$, that is, $Q = -Y$. In this particular case the denoising operation can be performed by solving only once the equation $A\mathbf{X}+\mathbf{X}B-Y = 0$. Finally, in the general case, it is possible to ignore the cross terms by making $\phi_p(x_i)\phi_k(x_i) \approx 0$. In this case, the computation is strongly simplified with a minor degradation of the results, where Q is computed using (17). The degradation resulting from this simplification depends on the amount and spatial density of the data.

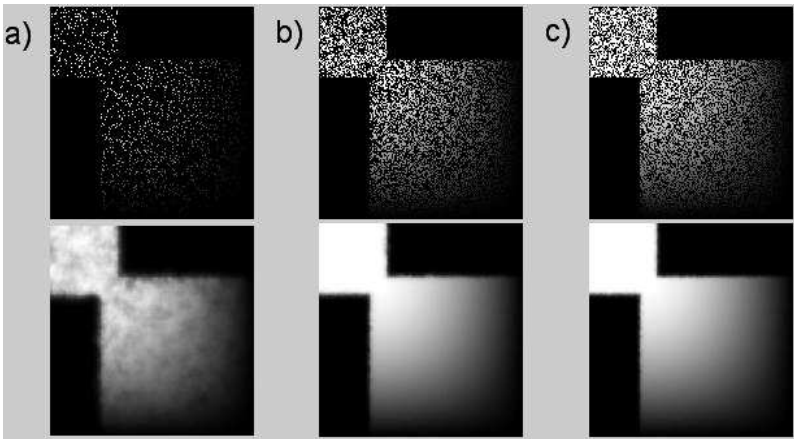

Fig. 3. Image reconstruction from sparse observations. a)$0.1L$ samples (28.34dB), b)$0.5L$ samples (46.36dB) and c)$0.75L$ samples where $L = 256^2$ is the number of pixels of the original 256×256 underlying image. (48.73dB).

4 Experimental results

In this section we present experimental results obtained with synthetic data in a Pentium 4 PC running at $2.8GHz$. Two methods are used and compared in both methods. The first one, denoted VECT, vectorizes the images involved and reconstructs the images accurately using direct or pseudo matrix inversion. In the second method, denoted LYAP, we use the SL solver, without requiring the matrix inversion.

Fig.2.a) shows the 64×64 noiseless image used in this experiment. It consists of three constant regions and a smooth region where the intensity varies according to a cosine function. Fig.2.b) shows the noisy image, with additive Gaussian white noise with $\sigma = 0.25$ and Fig.2.c) shows the denoised image. The results using both methods are equal but the processing time was 530.12 seconds for the VECT algorithm and 5.31 seconds for the LYAP method.

The noiseless image was used to test the performance of both algorithms with several image sizes and noise energy. Table 1 shows the simulation results for several image dimensions, $N \times N$, and for three values of noise: small ($\sigma = 1e^{-4}$), medium ($\sigma = 0.15$) and severe ($\sigma = 0.5$). Several figures of merit are shown to assess the algorithm performance: the signal to noise ratio (SNR), the minimized energy values, E, and the processing time for each experiment and for both methods. Both algorithms produce the same result, same SNR and E (energy), but the LYAP method clearly outperforms the VECT in terms of processing time when $N > 16$. We note an unexpected jump in the processing time when the image dimensions increase from $N = 48$ to $N = 56$. This results from the fact that, for large enough images, the PC starts to use the HD virtual memory. In addition the algorithm was also tested with non-uniform sampling with the

	$\sigma = 1e^{-4}$				$\sigma = 0.15$				$\sigma = 0.5$			
N	SNR	E	time(s)		SNR	E	time(s)		SNR	E	time(s)	
	(dB)		VECT	LYAP	(dB)		VECT	LYAP			VECT	LYAP
8	12.5	2.1	**0.00**	**0.04**	12.4	2.2	**0.04**	**0.49**	5.9	4.0	**0.03**	**0.78**
16	22.5	5.5	**0.08**	**0.06**	22.3	5.8	**0.09**	**0.32**	17.9	10.4	**0.09**	**0.17**
24	28.1	8.5	**0.68**	**0.06**	27.1	9.6	**0.72**	**0.76**	23.6	20.2	**0.72**	**0.54**
32	30.4	11.9	**3.91**	**0.07**	29.1	13.8	**3.94**	**0.06**	23.2	33.2	**3.93**	**0.08**
40	33.4	14.7	**14.46**	**0.07**	31.5	17.7	**14.36**	**0.10**	23.5	48.7	**14.35**	**0.07**
48	35.7	17.4	**43.85**	**0.08**	33.9	21.6	**43.63**	**0.08**	24.8	65.9	**43.65**	**0.43**
56	36.8	20.8	**133.27**	**4.50**	34.9	26.5	**120.26**	**2.65**	25.5	82.8	**120.31**	**2.13**
64	38.4	23.5	**516.88**	**7.85**	36.3	31.1	**381.87**	**4.89**	25.1	110.9	**432.32**	**5.80**

Table 1. Simulation results for several image dimensions ($N \times N$) and three different noise energy values, $\sigma = \{10^{-4}, 1.5, 0.5\}$.

LYAP method. Fig.3 shows the reconstruction results obtained with the LYAP method for three different numbers of samples, a)$0.1L$, b)$0.5L$ and c)$0.75L$, where $L = 256^2$ is the number of pixels of the original image. The SNR obtained in these experiments was (28.34dB), (46.36dB) and (48.73dB) for $0.1L$, $0.5L$ and $0.75L$ samples respectively. The iterative LYAP method (see equation (16)) spent 24.3 seconds to generate the 256×256 reconstructed images. The method was also applied to real images with good results as well.

5 Conclusions

This paper shows that image reconstruction and denoising with Gaussian noise can be performed using the SL equation. This equation can be efficiently solved by numeric methods leading to fast reconstruction and denoising algorithm. This method avoids the use of huge $nm \times nm$ matrices and their vectorization and inversion.

Experimental results comparing the proposed algorithm with a direct implementation of image denoising and reconstruction show that important com-

putational time savings are achieved. A comparizon with other state of the art reconstruction algorithms in terms of computational efficientcy will be presented in a future work.

Appendix - Basic algorithm

An exact and non-iterative optimization algorithm may be derived by vectorizing the matrices involved, that is, making $\mathbf{x} = vect(\mathbf{X})$. The corresponding energy function is $E(\mathbf{x}) = \frac{1}{2}(F(\mathbf{z}) - \mathbf{y})^T(F(\mathbf{z}) - \mathbf{y}) + \alpha\sigma^2((\Delta_V\mathbf{x})^T(\Delta_V\mathbf{x}) + (\Delta_H\mathbf{x})^T(\Delta_H\mathbf{x}))$ where Δ_V and Δ_H are $NM \times NM$ difference matrices, $F(\mathbf{z})$ and $\mathbf{y}$ are L-dimensional column vectors, $\mathbf{x}$ is a NM-dimensional column vector and $F(\mathbf{z}) = \Psi(\mathbf{z})\mathbf{x}$ where Ψ is the following $L \times NM$ matrix

$$\Psi = \begin{pmatrix} \phi_1(z_1) & ... & \phi_{nm}(z_1) \\ ... & ... & ... \\ \phi_1(z_L) & ... & \phi_{nm}(z_L) \end{pmatrix} \tag{18}$$

The minimizer of $E(\mathbf{x})$ is $\hat{\mathbf{x}} = (\Psi^T\Psi + 2\alpha\sigma^2(\Delta_V^T\Delta_V + \Delta_H^T\Delta_H))^{-1}\Psi^T\mathbf{y}$. This computation of $\hat{\mathbf{x}}$ is difficult in practice because of the huge dimensions of the matrices involved.

References

1. M. R. Banham, A. K. Katsaggelos, Digital Image Restoration, IEEE Signal Processing Magazine, Vol. 14, n 2, March 1997.
2. D. Wei, A.C. Bovik, Wavelet denoising for image enhancement, in The Handbook of Image and Video Processing, A.C. Bovik, (Ed.), Academic Press, 2000.
3. M. Figueiredo, R. Novak, An EM Algorithm for Wavelet-Based Image Restoration, IEEE Trans. Image Processing, Vol. 12, 906-916, 2003.
4. Chang, G., Yu, B., and Vetterli, M. Adaptive wavelet thresholding for image denoising and compression, IEEE Trans. Image Processing, 9 (2000), 1532-1546.
5. M. J. Black, G. Sapiro, D. H. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Trans. Image Processing, Vol. 7, 421-432, 1998.
6. C. Riddell, H. Benali, I. Buvat, Diffusion Regularization for Iterative Reconstruction in Emission Tomography, IEEE Trans. Nuclear Science, VOL. 52, 669-675, 2005
7. J. C. Ye, Y. Bresler, P. Moulin, A Self-Referencing Level-Set Method for Image Reconstruction from Sparse Fourier Samples, Int. Journal of Computer Vision, Vol. 50, 253-270, 2002.
8. T. Deschamps, R. Malladi, I. Ravve, Fast Evolution of Image Manifolds and Application to Filtering and Segmentation in 3D Medical Images, IEEE Trans. Visualization and Computer Graphics, Vol. 10, 525-535, 2004
9. Brogan, William L. Modern Control Theory, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
10. Bartels, R.H. and Stewart, G.W., Solution of the matrix equation A X + XB = C, Comm. A.C.M., 15, pp. 820-826, 1972.
11. Barraud, A.Y., A numerical algorithm to solve A XA - X = Q., IEEE Trans. Auto. Contr., AC-22, pp. 883-885, 1977.
12. D. Calvetti and L. Reichel, Application of ADI Iterative Methods to the Restoration of Noisy Images, SIAM J. Matrix Anal. Appl., vol.17,no. 1, 1996.