Open access • Proceedings Article • DOI:10.1109/ICIP.2006.312573

Robust Kernel Regression for Restoration and Reconstruction of Images from Sparse Noisy Data — Source link

Hiroyuki Takeda, Sina Farsiu, Peyman Milanfar

Institutions: University of California, Santa Cruz

Published on: 01 Oct 2006 - International Conference on Image Processing

Topics: Image restoration, Gaussian noise, Iterative reconstruction, Noise (video) and Noise reduction

ROBUST KERNEL REGRESSION FOR RESTORATION AND RECONSTRUCTION OF IMAGES FROM SPARSE NOISY DATA

Hiroyuki Takeda, Sina Farsiu, Peyman Milanfar

Department of Electrical Engineering, University of California at Santa Cruz
{htakeda,farsiu,milanfar}@soe.ucsc.edu

ABSTRACT

We introduce a class of robust non-parametric estimation methods which are ideally suited for the reconstruction of signals and images from noise-corrupted or sparsely collected samples. The filters derived from this class are locally adapted kernels which take into account both the local density of the available samples, and the actual values of these samples. As such, they are automatically steered and adapted to both the given sampling "geometry", and the samples' "radiometry". As the framework we proposed does not rely upon specific assumptions about noise or sampling distributions, it is applicable to a wide class of problems including efficient image upscaling, high quality reconstruction of an image from as little as 15% of its (irregularly sampled) pixels, super-resolution from noisy and under-determined data sets, state of the art denoising of images corrupted by Gaussian and other noise, effective removal of compression artifacts; and more.

Index Terms— Inverse problem, image reconstruction, piecewise polynomial approximation, nonlinear estimation

1. INTRODUCTION

Image processing methods have been exploited through the years to improve the quality of digital images. Many of the popular image processing tools have a limited scope of use; some can only be employed as denosing methods, while application of others are limited to upscaling regularly sampled data. Moreover, such methods estimate the underlying signal based on certain assumptions on data and noise models, a common example of which is modeling the noise as pure additive i.i.d. Gaussian. Although such limiting assumptions facilitate the design of optimal methods for a certain type of data, in real situations when the data and noise models do not faithfully describe the measured signal, the performance of such non-robust methods significantly degrades [1].

Classical parametric image processing methods rely on a specific model of the signal of interest, and seek to compute the parameters of this model in the presence of noise. In contrast to the parametric methods, non-parametric methods rely on the data itself to dictate the structure of the model, in which case this implicit model is referred to as *a regression function* [2]. We promote the use and improve upon a class of non-parametric methods called *kernel regression* [3], which generalizes some recently presented methods namely, *normalized convolution* [4], *bilateral filter* [5, 6], and *moving least-squares* [7].

The main advantage of the presented regression method is that it is a generic framework enabling direct use in a variety of applications, from single frame denoising to multi frame super-resolution [3]. Moreover, this method produces better and more stable results comparing to the state of the art methods in the literature, as it is robust to modeling errors and data outliers.

This paper is organized as follows. Section 2 is a brief introduction to the notion of adaptive kernel regression and the novel concept of using weighted l_1 norm penalty term in the kernel regression framework. Section 3 extends and generalizes the previous related methods to derive the details of the proposed robust regression method, focusing on appropriate choices for the kernel function. Simulation results are presented in Section 4, and Section 5 concludes this paper.

2. DATA-ADAPTED KERNEL REGRESSION

We treat the 2-D estimation problem where the measured data y_i at the position $\mathbf{x}_i = [x_{1i}, x_{2i}]^T$ is given by

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \cdots, P, \tag{1}$$

where $z(\cdot)$ is the (hitherto unspecified) regression function (i.e. an unknown image) to be estimated, P is the number of measured pixels, and ε_i's are the independent and identically distributed noise values (with otherwise no particular statistical distribution assumed).

While the specific form of $z(\cdot)$ may remain unspecified, if we assume that it is locally smooth to some order N, then in order to estimate the value of the function at any given point $\mathbf{x}$, we can rely on a generic local expansion of the function about this point. Specifically, if $\mathbf{x}$ is near the sample at $\mathbf{x}_i$, we have the N-term Taylor series

$$\begin{aligned} z(\mathbf{x}_i) &\approx z(\mathbf{x}) + \{\boldsymbol{\nabla} z(\mathbf{x})\}^T(\mathbf{x}_i - \mathbf{x}) \\ &\quad + \frac{1}{2}(\mathbf{x}_i - \mathbf{x})^T\{\boldsymbol{\mathcal{H}} z(\mathbf{x})\}(\mathbf{x}_i - \mathbf{x}) + \cdots \qquad (2) \\ &= \beta_0 + \boldsymbol{\beta}_1^T(\mathbf{x}_i - \mathbf{x}) \\ &\quad + \boldsymbol{\beta}_2^T \operatorname{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} + \cdots, \qquad (3) \end{aligned}$$

where $\boldsymbol{\nabla}$ and $\boldsymbol{\mathcal{H}}$ are the gradient and Hessian operators respectively, and $\mathrm{vech}(\cdot)$ is the *half-vectorization operator* [2], which lexicographically orders the "lower-triangular" portion of a matrix into a column vector. Indeed the local approximation can be also built upon bases other than polynomials [8]

The above suggests that if we now think of the Taylor series as a local representation of the regression function, estimating the parameter β_0 can yield the desired (local) estimate of the regression function based on the data. Indeed, the parameters

This work was supported in part by DARPA/AFOSR Grant FA9550-06-1-0047; by AFOSR Grant F49620-03-1-0387, and by the National Science Foundation Science and Technology Center for Adaptive Optics, managed by the University of California at Santa Cruz under Cooperative Agreement No. AST-9876783.

$\{\boldsymbol{\beta}_n\}_{n=1}^N$ will provide localized information on the n-th *derivatives* of the regression function. Naturally, since this approach is based on *local* approximations, classical regression methods estimate the coefficients $\{\boldsymbol{\beta}_n\}_{n=0}^N$ from the data while giving the nearby samples higher weights than samples farther away ("geometric" weighting). However, it is also appropriate to weight samples based on their relative location with respect to a local edge ("radiometric" weighting), preforming the regression along and not across the edges, which is the basis of modern adaptive methods . A general formulation we propose, capturing this idea is to solve the following optimization problem:

$$\min_{\{\boldsymbol{\beta}_n\}_{n=0}^N} \sum_{i=1}^{P} \Big| y_i - \beta_0 - \boldsymbol{\beta}_1^T(\mathbf{x}_i - \mathbf{x}) - \boldsymbol{\beta}_2^T \operatorname{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} - \cdots \Big|^m \mathsf{K}(\mathbf{x}_i - \mathbf{x}, y_i - y) \quad (4)$$

where $\mathsf{K}(\cdot)$ is the *kernel function* which penalizes both geometric and radiometric distances and will be described in detail in Section 3, and m is the penalizing parameter. To the best of our knowledge, all kernel regression based methods in the literature choose the penalizing parameter as $m = 2$, and therefore pose (4) as a weighted least-squares problem. In Section 4, we show that robustness with respect to the outliers can be significantly improved by exploiting other values for the penalizing parameter such as $m = 1$, which in effect incorporates a robust l_1 norm estimator [1] in the kernel regression framework. Furthermore, we propose novel ways to adopt the kernel.

Using the matrix form, the optimization problem (4) can be posed as weighted l_m norm:

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}_{\mathbf{x}}\mathbf{b}\|_{\mathbf{W}_{\mathbf{x}}}^m, \quad (5)$$

where

$$\mathbf{y} = [y_1, y_2, \cdots, y_P]^T, \quad \mathbf{b} = \left[\beta_0, \boldsymbol{\beta}_1^T, \cdots, \boldsymbol{\beta}_N^T\right]^T, \quad (6)$$

$$\mathbf{W}_{\mathbf{x}} = \operatorname{diag}\left[\mathsf{K}(\mathbf{x}_1 - \mathbf{x}, y_1 - y), \cdots, \mathsf{K}(\mathbf{x}_P - \mathbf{x}, y_P - y)\right], \quad (7)$$

$$\mathbf{X}_{\mathbf{x}} = \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{x})^T & \operatorname{vech}^T\left\{(\mathbf{x}_1 - \mathbf{x})(\mathbf{x}_1 - \mathbf{x})^T\right\} & \cdots \\ 1 & (\mathbf{x}_2 - \mathbf{x})^T & \operatorname{vech}^T\left\{(\mathbf{x}_2 - \mathbf{x})(\mathbf{x}_2 - \mathbf{x})^T\right\} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (\mathbf{x}_P - \mathbf{x})^T & \operatorname{vech}^T\left\{(\mathbf{x}_P - \mathbf{x})(\mathbf{x}_P - \mathbf{x})^T\right\} & \cdots \end{bmatrix}, \quad (8)$$

with "diag" defining the diagonal elements of a diagonal matrix. We use steepest descent to find the solution to this minimization problem:

$$\hat{\mathbf{b}}^{(n+1)} = \hat{\mathbf{b}}^{(n)} + \alpha \mathbf{X}_{\mathbf{x}}^T \mathbf{W}_{\mathbf{x}} \operatorname{sign}\left(\mathbf{y} - \mathbf{X}_{\mathbf{x}}\hat{\mathbf{b}}^{(n)}\right) \odot \left|\mathbf{y} - \mathbf{X}_{\mathbf{x}}\hat{\mathbf{b}}^{(n)}\right|^{m-1}, \quad (9)$$

where α is a scalar defining the step size in the direction of the gradient, and $\odot$ is the element by element multiplication operator.

The order (N) of regression affects the complexity of the local approximation of the signal. In the non-parametric statistics literature, locally constant, linear and quadratic approximations (corresponding to $N = 0, 1, 2$ respectively) have been most widely considered [2]. In particular, choosing local constant estimation with $m = 2$, a locally linear adaptive filter is obtained, which is known as the *Nadaraya-Watson Estimator* (NWE) [3]. In general, lower order approximates, such as NWE, result in smoother images (large bias and small variance) as there are fewer degrees of freedom. On the other hand over-fitting happens in regressions using higher orders of approximation, resulting in small bias and larger estimation variance. Note that, in the experiments of Section 4 we used the second order ($N = 2$) approximation.

3. KERNEL FUNCTION SELECTION

The choice of kernel function greatly affects the quality of reconstruction. In this section, first we briefly review the classic "non-adaptive" kernel function, and then generalize it to derive two adaptive kernel functions with superior performance.

3.1. Classic Kernel Function

In classic kernel regression, samples are weighted based only on their spatial distances to a sample of interest, which simplifies the kernel $\mathsf{K}(\cdot)$ in (4) to

$$\mathsf{K}(\mathbf{x}_i - \mathbf{x}, y_i - y) \equiv K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}), \quad (10)$$

where $K_{\mathbf{H}_i}(\cdot)$ is defined as

$$K_{\mathbf{H}_i}(\mathbf{t}) = \frac{1}{\det(\mathbf{H}_i)} K\left(\mathbf{H}_i^{-1}\mathbf{t}\right), \quad (11)$$

which penalizes distance away from the local position where the approximation is centered. The 2×2 "smoothing" matrix $\mathbf{H}_i$ controls the strength of this penalty. The standard choice of the smoothing matrix is $\mathbf{H}_i = h\mu_i\mathbf{I}_2$, where μ_i is a scalar that captures the local density of data samples and h is the *global smoothing parameter*, extending the kernel to contain "enough" samples. As described in [3], in case of irregularly sampled data, it is reasonable to use smaller kernels in the areas with more available samples, whereas larger kernels are more suitable for the more sparsely sampled areas of the image. The choice of the particular form of the function $K(\cdot)$ is open, and may be selected as any symmetric function, which attains its maximum at zero such as Gaussian.

Since the shape of the classic kernels is independent of the radiometric (gray level) information, as described in [3], classic kernel based regression methods suffer from an inherent limitation due to the local linear action on the data. In what follows, we discuss extensions of the kernel regression method that enable this structure to have nonlinear, more effective, action on the data. The proposed adaptive kernel functions rely on not only the sample location and density, but also the radiometric properties of these samples. Therefore, the effective size and shape of the regression kernel are adapted locally to image features such as edges. This property is illustrated in Fig. 1, where the classical and adaptive kernel shapes in the presence of an edge are compared.

3.2. Bilateral Kernel Function

A simple and intuitive choice of the adaptive kernel $\mathsf{K}(\cdot)$ is to use separate terms for penalizing the spatial and radiometric distances. Indeed this is precisely the thinking behind the *bilateral* filter, introduced in [5, 6]. The bilateral kernel choice is then

$$\mathsf{K}(\mathbf{x}_i - \mathbf{x}, y_i - y) \equiv K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) K_{h_r}(y_i - y), \quad (12)$$

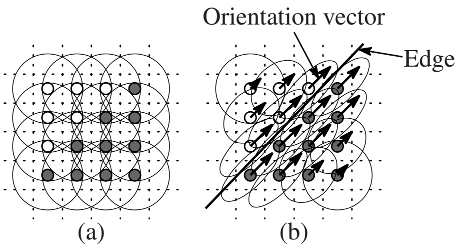


Fig. 1. Kernel spread in a uniformly sampled data set. (a) Kernels in the classic method depend only on the sample density. (b) Adaptive kernels elongate with respect to the edge.

where h_r is the radiometric smoothing scalar that controls the rate of decay, and $K_{\mathbf{H}_i}(\cdot)$ and $K_{h_r}(\cdot)$ are the spatial and radiometric kernel functions, respectively. In general, the application of bilateral kernel is limited to denoising problem, since the pixel value (y) at an arbitral position ($\mathbf{x}$) might not be available from data. This limitation, however, can be overcome by using an initial estimate of y by an appropriate interpolation technique [3]. Also, breaking $\mathsf{K}(\cdot)$ into spatial and radiometric terms as utilized in the bilateral case weakens the estimator performance since it limits the degrees of freedom and ignores correlations between positions of the pixels and their values. The following section provides a solution to overcome this drawback.

3.3. Steering Kernel Function

Based upon the earlier non-parametric framework, the filtering procedure we propose next takes the above ideas one step further. In particular, we observe that the effect of computing $K_{h_r}(y_i - y)$ in (12) is to implicitly measure a function of the local gradient estimated between neighboring values, and to use this estimate to weight the respective measurements. As an example, if a pixel is located near an edge, then pixels on the same side of the edge will have much stronger influence in the filtering. With this intuition in mind, we propose a two-step approach where first an initial estimate of the image gradients is made using some kind of gradient estimator (say the second order *classic* kernel regression method). Next, this estimate is used to measure the dominant orientation of the local gradients in the image. In a second filtering stage, this orientation information is used to adaptively "steer" the local kernel, resulting in elongated, elliptical contours spread along the directions of the local edge structure. With these locally adapted kernels, the denoising is effected most strongly along the edges, rather than across them, resulting in strong preservation of details in the final output. To be more specific, the steering kernel takes the form

$$\mathsf{K}(\mathbf{x}_i - \mathbf{x}, y_i - y) \equiv K_{\mathbf{H}_i^{\mathrm{s}}}(\mathbf{x}_i - \mathbf{x}), \tag{13}$$

where $\mathbf{H}_i^{\mathrm{s}}$'s are the data-dependent full matrices which we call *steering* matrices. They are defined as

$$\mathbf{H}_i^{\mathrm{s}} = h\mu_i \mathbf{C}_i^{-\frac{1}{2}}, \tag{14}$$

where $\mathbf{C}_i$'s are (symmetric) covariance matrices based on the local gray-values. A good choice for $\mathbf{C}_i$ will effectively spread the kernel function along the local edges as shown in Fig. 1. It is worth noting that even if we choose a large h in order to have a strong denoising effect, the undesirable blurring effect which would otherwise have resulted, is tempered around edges with appropriate choice of $\mathbf{C}_i$'s. With such steering matrices, for example, if we choose a Gaussian kernel, the steering kernel is mathematically represented as

$$K_{\mathbf{H}_i^{\mathrm{s}}}(\mathbf{x}_i - \mathbf{x}) = \frac{\sqrt{\det(\mathbf{C}_i)}}{2\pi h^2} \exp\left\{ -\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{C}_i (\mathbf{x}_i - \mathbf{x})}{2h^2} \right\}. \tag{15}$$

The local edge structure is related to the gradient covariance (or equivalently, the locally dominant orientation). In [3] we have shown that a convenient form of representing the covariance matrix, is to decompose it into three components as follows:

$$\mathbf{C}_i = \gamma_i \mathbf{U}_{\theta_i} \mathbf{\Lambda}_i \mathbf{U}_{\theta_i}^T, \tag{16}$$

$$\mathbf{U}_{\theta_i} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}, \quad \mathbf{\Lambda}_i = \begin{bmatrix} \sigma_i & 0 \\ 0 & \sigma_i^{-1} \end{bmatrix}. \tag{17}$$

where $\mathbf{U}_{\theta_i}$ is the rotation matrix and $\mathbf{\Lambda}_i$ is the elongation matrix. Now the covariance matrix is given by the three parameters γ_i, θ_i and σ_i, which are the scaling, rotation, and elongation parameters, respectively and the effect of which are as follows. First, the initial circular kernel is elongated by the elongation matrix $\mathbf{\Lambda}_i$ with semi-minor and major axes given by σ_i and σ_i^{-1}, respectively. Second, the elongated kernel is rotated by the matrix $\mathbf{U}_{\theta_i}$. Finally, the kernel is scaled by the scaling parameter γ_i. We refer the reader to [3] for the details of estimating these parameters in an iterative fashion. We note that the presented formulation is close to the apparently independently derived normalized convolution formulation of [4].

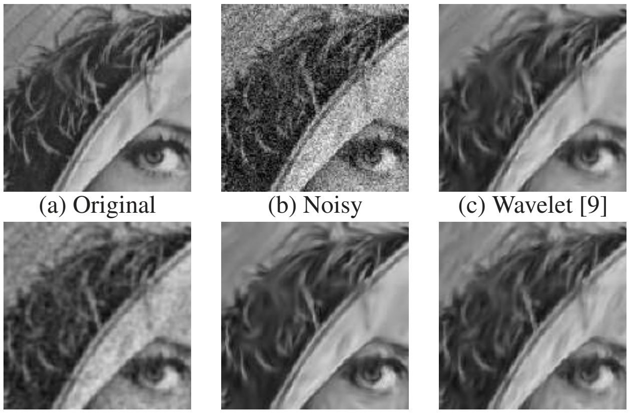
4. EXPERIMENTS

In this section we compare the performance of the proposed algorithm to other methods. We show that in presence of white Gaussian noise the proposed robust kernel regression method works as well if not better than the state of the art recent wavelet based denoising method of [9], and other popular methods. We also note that the wavelet method in general is computationally more efficient than the steering kernel method. However, in presence of other noise models (such as salt and pepper noise) while the performance of non-robust methods dramatically degrades, the proposed l_1 based robust method effectively removes the noise. The criterion for parameter selections in all the examples was to choose parameters which gave the best RMSE values.

In the first experiment, we added white Gaussian noise with standard deviation (STD) of 25 to the original image of Fig. 2(a) resulting in the noisy image of Fig. 2(b). Denoised images using the wavelet[1] method of [9]; classic kernel regression method ($m = 2$, $h = 1.33$), steering kernel regression method ($m = 2$, $h = 1.33$, 7 iterations initialized with l_2 classic), steering kernel regression method ($m = 1$, $h = 3$, 2 iterations initialized with l_1 classic) and corresponding Root Mean Square Error (RMSE) values are shown in Fig. 2(c)-(f), respectively.

In the second experiment we added 20% salt and pepper noise to the original image of Fig. 2(a) resulting in the noisy image of Fig. 3(a). Denoised images using a 3 × 3 median filter, wavelet method of [9], classic kernel regression method ($m = 2$,

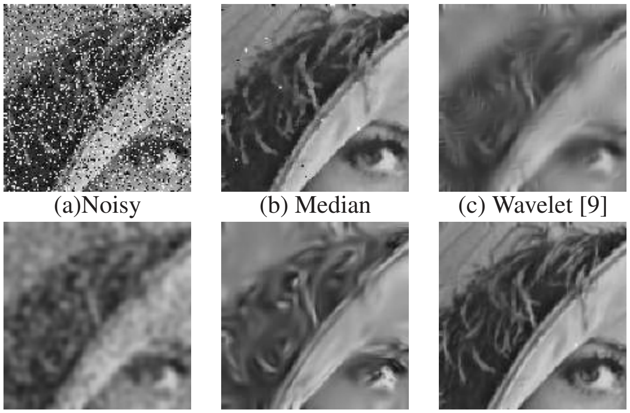
[1] This result is produced by the software, available on http://decsai.ugr.es/∼javier/denoise/index.html.



(a) Original (b) Noisy (c) Wavelet [9]

(d) l_2 Classic (e) l_2 Steering (f) l_1 Steering

Fig. 2. Gaussian noise removal experiment. Corresponding RMSE values for (b)-(f) are 25.0, 9.71, 11.36, 10.11, and 10.71, respectively.



(a)Noisy (b) Median (c) Wavelet [9]

(d) l_2 Classic (e) l_2 Steering (f) l_1 Steering

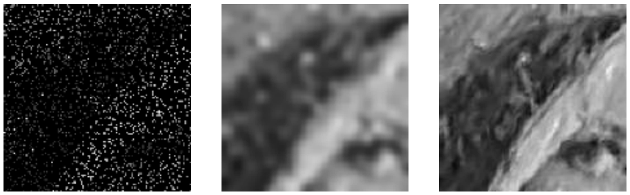
Fig. 3. Salt & pepper noise removal experiment. Corresponding RMSE values for Figures(a)-(f) are 63.84, 11.05, 21.54, 21.81, 21.06, and 7.14, respectively.

$h = 2.46$), steering kernel regression method ($m = 2$, $h = 2.25$, 20 iterations initialized with l_2 classic), steering kernel regression method ($m = 1$,$h = 2.25$, zero iteration initialized with l_1 classic) and corresponding RMSE values are shown in Fig. 3(b)-(f), respectively.

In our final experiment, we added white Gaussian noise with STD of 10 along with 5% salt and pepper noise to the original image of Fig. 2(a). Then, we randomly eliminated 85% of these noisy pixels, creating the sparse image of Fig. 4(a). Interpolated and denoised images using the Delaunay-spline smoother (refer to [3] for details), and the iterative steering kernel regression method ($m = 1$, $h = 3$, 0 iterations) and corresponding RMSE values are shown in Fig. 4(b)-(c), respectively.

5. CONCLUSIONS

In this paper we promoted, extended, and demonstrated kernel regression as a general framework for studying several efficient denoising and interpolation algorithms. To overcome the inherent limitations dictated by the linear filtering properties of the classic kernel regression methods, we introduced the non-linear data-adapted class of kernel regressors with superior performance. Furthermore, we achieved robustness with respect to outliers in data and noise model by incorporating the l_1 norm penalty in the kernel regression framework. Image deblurring is also an important issue in image reconstruction, and it is a part of our ongoing work within this framework.



(a) Subsampled (b) Del. Spline (c) l_1 Steering

Fig. 4. Sparse-noisy image interpolation experiment. (a) is the input image with 85% of pixels removed, and further corrupted by adding Gaussian and salt and pepper noise. Reconstructed images using the Delaunay-spline smoother (RMSE=22.5), and the l_1 steering kernel regression (RMSE=17.5) methods, are shown in (b)-(c), respectively.

6. REFERENCES

[1] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super-resolution," *IEEE Trans. on Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[2] D. Ruppert and M. P. Wand, "Multivariate locally weighted least squares regression," *The annals of statistics*, vol. 22, no. 3, pp. 1346–1370, Sept. 1994.

[3] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," submitted to IEEE Trans. on Image Proc., 2005, available at *http://www.soe.ucsc.edu/∼milanfar*.

[4] T. Q. Pham, L. J. van Vliet, and K. Schutte, "Robust fusion of irregularly sampled data using adaptive normalized convolution," *EURASIP Journal on Applied Signal Processing*, 2006.

[5] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *in Proc. IEEE Int. Conf. Computer Vision, New Delhi, India*, pp. 836–846, Jan. 1998.

[6] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. on Image Processing*, vol. 11, no. 10, pp. 1141–1150, Oct. 2002.

[7] N. K. Bose and N. Ahuja, "Superresolution and noise filtering using moving least squares," submitted to IEEE Trans. on Image Proc., 2005.

[8] W. Hardle, M. Muller, S. Sperlich, and A. Werwatz, *Nonparametric and Semiparametric Models*, Springer Series in Statistics. Springer, Berlin ; New York, 2004.

[9] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. on Image Proc.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.