Regularized Kernel Regression for Image Deblurring

Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar
Department of Electrical Engineering, University of California at Santa Cruz
{htakeda,farsiu,milanfar}@soe.ucsc.edu

***Abstract*— The framework of *kernel regression* [1], a non-parametric estimation method, has been widely used in different guises for solving a variety of image processing problems including denoising and interpolation [2]. In this paper, we extend the use of kernel regression for deblurring applications. Furthermore, we show that many of the popular image reconstruction techniques are special cases of the proposed framework. Simulation results confirm the effectiveness of our proposed methods.**

I. Introduction

In our earlier work [2], [3], we studied the kernel regression (KR) framework, and proposed the data-adaptive version of kernel regression for use in image and video processing. The applicability of data-adapted kernel regression is wide-ranging, for example, image denoising (including white Gaussian, Laplacian, Salt & Pepper, Compression artifacts, and Color artifacts), and image interpolation/reconstruction from regular and irregularly sampled data sets (e.g. image fusion and super-resolution). However, since these direct applications of KR neglected the atmosphere or camera's point spread function (blur) effects, the estimated signals (images) need further processing. Such a two-step filtering process (e.g. denoising + deblurring) is in general suboptimal [4]. In this paper, we develop a one-step procedure for denoising and deblurring, based on the kernel regression framework.

To date, the kernel regression framework has never been directly used for deblurring. As is well-known, deblurring is an ill-posed problem, and it requires an appropriate regularization which introduces prior information about the desired signals. For piecewise constant signals, the total variation (TV) regularization (L_1-norm), proposed in [5], is a better choice than the Tikhonov regularization (L_2-norm) for denoising [6], [7], [8] as well as deblurring [9], [10] applications. The implicit assumption of piecewise constancy in TV regularization prevents estimated signals from having fine texture and gradation. Hence, the question which we must consider is how to more effectively estimate signals with texture and gradation. In this paper, we incorporate the theory of kernel regression and propose a deblurring algorithm with a suitable regularization.

This work was supported in part 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.

This paper is organized as follows. In Section II, we briefly review the kernel regression framework and formulate a deblurring estimator. Section III illustrates two experiments on simulated data sets, and we conclude this paper in the last section.

II. Kernel Deblurring

A. Review

The kernel regression framework defines its data model in 2-D as

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \cdots, P, \quad \mathbf{x}_i = [x_{1i}, x_{2i}]^T, \quad (1)$$

where y_i is a noisy sample at $\mathbf{x}_i$, $z(\cdot)$ is *the* (hitherto unspecified) *regression function* to be estimated, ε_i is an i.i.d zero mean noise, and P is the total number of samples in a neighborhood (window) of interest. As such, the kernel regression framework provides a rich mechanism for computing point-wise estimates of the regression function with minimal assumptions about global signal or noise models.

While the specific form of $z(\cdot)$ may remain unspecified, we can rely on a generic local expansion of the function about a sampling point $\mathbf{x}_i$. Specifically, if $\mathbf{x}$ is near the sample at $\mathbf{x}_i$, we have the $(N+1)$-term Taylor series[1]

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

where $\boldsymbol{\nabla}$ and $\boldsymbol{\mathcal{H}}$ are the gradient (2×1) and Hessian (2×2) operators, respectively, and $\mathrm{vech}(\cdot)$ is the half-vectorization operator which lexicographically orders the lower triangular portion of a symmetric matrix. Furthermore, β_0 is $z(\mathbf{x})$, which is the pixel value of interest, and the vectors $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are

$$\boldsymbol{\beta}_1 = \left[\frac{\partial z(\mathbf{x})}{\partial x_1} \quad \frac{\partial z(\mathbf{x})}{\partial x_2} \right]^T, \quad (4)$$

$$\boldsymbol{\beta}_2 = \left[\frac{\partial^2 z(\mathbf{x})}{2\partial x_1^2} \quad \frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2} \quad \frac{\partial^2 z(\mathbf{x})}{2\partial x_2^2} \right]^T. \quad (5)$$

Since this approach is based on *local* approximations, a logical step to take is to estimate the parameters $\{\boldsymbol{\beta}_n\}_{n=0}^N$

[1]Other localized representations are also possible and may be advantageous.

from all the samples $\{y_i\}_{i=1}^{P}$ while giving the nearby samples higher weights than samples farther away. A formulation of the fitting problem capturing this idea is to solve the following optimization problem,

$$\min_{\{\boldsymbol{\beta}_n\}_{n=0}^{N}} \sum_{i=1}^{P} \Big| y_i - \beta_0 - \boldsymbol{\beta}_1^T(\mathbf{x}_i - \mathbf{x}) - \boldsymbol{\beta}_2^T \operatorname{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T - \cdots\right\} \Big|^m K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) \quad (6)$$

with

$$K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) = \frac{1}{\det(\mathbf{H}_i)} K(\mathbf{H}_i^{-1}(\mathbf{x}_i - \mathbf{x})), \quad (7)$$

where N is the regression order, m is the error norm parameter typically set to 2, $K(\cdot)$ is the kernel function (a radially symmetric function), and $\mathbf{H}_i$ is the smoothing (2×2) matrix which dictates the "footprint" of the kernel function. Although the choice of the particular form of the kernel is open, it has a relatively small effect on the accuracy of estimation [11]. More details about the optimization problem above can be found in [2], [3].

Following our previous work [2], we also briefly review three types of the kernel functions.

1) Classic kernel:

$$K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}), \text{ with } \mathbf{H}_i = h\mathbf{I}, \quad (8)$$

where h is the global (spatial) smoothing parameter. This is the standard choice of the kernel function and it is a function of only spatial information (spatial distances). However the classic kernel suffers from a limitation due to the local linear action on the given data. Thus, in [2], we proposed two data-adapted kernel functions; bilateral kernel and steering kernel, which depend not only on spatial information but also on radiometric information.

2) Bilateral kernel:

$$K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) K_{h_r}(y_i - y), \text{ with } \mathbf{H}_i = h\mathbf{I}, \quad (9)$$

where h_r is the global radiometric smoothing parameter. In [2], we showed that the popular bilateral filter [12], [13] is a special case of the general KR denoising approach using the above bilateral kernel. The bilateral kernel takes radiometric distances explicitly into account, which limits its performance, particularly when the measurements are very noisy.

3) Steering kernel:

$$K_{\mathbf{H}_i^{\text{steer}}}(\mathbf{x}_i - \mathbf{x}), \text{ with } \mathbf{H}_i^{\text{steer}} = h\mathbf{C}_i^{\frac{1}{2}}, \quad (10)$$

is a more robust choice for the data-adaptive kernel functions [14], [2]. We call $\mathbf{H}_i^{\text{steer}}$ the steering matrix, and a naive estimate of the covariance matrix $\mathbf{C}_i$ may be obtained as follows:

$$\widehat{\mathbf{C}}_i \approx \begin{bmatrix} \sum_{\mathbf{x}_j \in w_i} z_{x_1}(\mathbf{x}_j) z_{x_1}(\mathbf{x}_j) & \sum_{\mathbf{x}_j \in w_i} z_{x_1}(\mathbf{x}_j) z_{x_2}(\mathbf{x}_j) \\ \sum_{\mathbf{x}_j \in w_i} z_{x_1}(\mathbf{x}_j) z_{x_2}(\mathbf{x}_j) & \sum_{\mathbf{x}_j \in w_i} z_{x_2}(\mathbf{x}_j) z_{x_2}(\mathbf{x}_j) \end{bmatrix}, \quad (11)$$

where $z_{x_i}(\cdot)$ and $z_{x_2}(\cdot)$ are the first derivatives along x_1 and x_2 directions and w_i is a local analysis window around the position of interest. The intuitive idea behind the superior performance of the steering kernel is to rely on the image structure information (given by the local dominant orientation [15]), instead of using radiometric distances directly to make the data-adaptive kernels. Since the regularized local orientation estimate as formulated in [2] has strong tolerance to noise, the steering kernel is more stable than the bilateral kernel.

The optimization problem (6) eventually provides a point-wise estimator of the regression function. However, as stated in the previous section, the data model (1) ignores other distortion effects. In the following section, we use a blurred and noise-ridden data model and derive a KR based deblurring/denoising estimator.

B. Deblurring Estimator and Related Methods

Defining the shift-invariant point spread function (PSF) as $b(\mathbf{x})$ and the desired true function as $u(\mathbf{x})$, we consider the blurred and noise-ridden data model

$$y \;=\; z(\mathbf{x}) + \varepsilon = b(\mathbf{x}) * u(\mathbf{x}) + \varepsilon, \quad (12)$$

where $*$ is the convolution operator. For convenience, we write the point-wise model in vector form as:

$$\underline{\mathbf{Y}} = \underline{\mathbf{Z}} + \underline{\boldsymbol{\varepsilon}} = \mathbf{B}\underline{\mathbf{U}} + \underline{\boldsymbol{\varepsilon}}, \quad (13)$$

where $\mathbf{Y}$ is the blurry and noisy measured image, $\mathbf{Z}$ is the blurry image, $\mathbf{B}$ is the blur operator, and $\mathbf{U}$ is the image of interest. The underline notation denotes matrices that are lexicographically ordered into column-stack vectors:

$$\underline{\mathbf{Y}} = \begin{bmatrix} y_1 \\ \vdots \\ y_P \end{bmatrix}, \underline{\mathbf{Z}} = \begin{bmatrix} z(\mathbf{x}_1) \\ \vdots \\ z(\mathbf{x}_P) \end{bmatrix}, \underline{\mathbf{U}} = \begin{bmatrix} u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_P) \end{bmatrix}, \underline{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_P \end{bmatrix}. \quad (14)$$

Using the above notations, we also rewrite the point-wise Taylor expansion (2) in vector form as (see Appendix for the derivation)

$$\begin{aligned} \underline{\mathbf{Z}} \;&\approx\; \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\Big(\underline{\mathbf{Z}} + \underline{\mathbf{Z}}_{x_1}v_1 + \underline{\mathbf{Z}}_{x_2}v_2 \\ &\qquad + \underline{\mathbf{Z}}_{x_1^2}v_1^2 + \underline{\mathbf{Z}}_{x_1x_2}v_1v_2 + \underline{\mathbf{Z}}_{x_2^2}v_2^2 + \cdots\Big) \\ &=\; \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbf{B}\Big(\underline{\mathbf{U}} + \underline{\mathbf{U}}_{x_1}v_1 + \underline{\mathbf{U}}_{x_2}v_2 \\ &\qquad + \underline{\mathbf{U}}_{x_1^2}v_1^2 + \underline{\mathbf{U}}_{x_1x_2}v_1v_2 + \underline{\mathbf{U}}_{x_2^2}v_2^2 + \cdots\Big) \\ &=\; \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbb{B}\mathbb{U} \end{aligned} \quad (15)$$

with

$$\begin{aligned} \mathbb{B} \;&=\; \Big[\mathbf{B} \;\; \mathbf{B}\mathbf{I}_{v_1} \;\; \mathbf{B}\mathbf{I}_{v_2} \;\; \mathbf{B}\mathbf{I}_{v_1^2} \;\; \mathbf{B}\mathbf{I}_{v_1v_2} \;\; \mathbf{B}\mathbf{I}_{v_2^2}\cdots\Big], \\ \mathbb{U} \;&=\; \Big[\underline{\mathbf{U}}^T \;\; \underline{\mathbf{U}}_{x_1}^T \;\; \underline{\mathbf{U}}_{x_2}^T \;\; \underline{\mathbf{U}}_{x_1^2}^T \;\; \underline{\mathbf{U}}_{x_1x_2}^T \;\; \underline{\mathbf{U}}_{x_2^2}^T\cdots\Big]^T, \end{aligned} \quad (16)$$

where $\mathbf{S}_{x_1}^{v_1}$ is the v_1-pixel shift operator along the x_1 direction, $\mathbf{Z}_{x_1}$ and $\mathbf{Z}_{x_1^2}$ are the first and the second derivative in the

x_1 direction respectively, and $\mathbf{I}_{v_1} = \text{diag}\{v_1, \cdots, v_1\}$. In the absence of other modeling errors and noise, the approximation suggests the following constraint:

$$\underline{\mathbf{Z}} - \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbb{B}\mathbb{U} = \underline{\mathbf{0}}. \tag{17}$$

Since Taylor approximation with a finite number of terms is only valid when v_1 and v_2 are small, it suggests a likelihood term with larger weights for smaller v_1, v_2:

$$C_L(\mathbb{U}) = \sum_{v_1}\sum_{v_2}\left\|\underline{\mathbf{Y}} - \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbb{B}\mathbb{U}\right\|_{\mathbf{W}(\mathbf{v})}^{m}, \tag{18}$$

where $\mathbf{v} = [v_1, v_2]^T$, m is the error norm parameter and

$$\mathbf{W}(\mathbf{v}) = \text{diag}\left\{K_{\mathbf{H}_1}(\mathbf{v}), \cdots, K_{\mathbf{H}_P}(\mathbf{v})\right\}. \tag{19}$$

However, with this cost function alone, the optimization problem might be still ill-posed; in particular, when the width of the PSF is large. Therefore, we introduce a regularization term which further restricts the solution space of the signal of interest. To this end, since the Taylor expansion locally represents the desired signals, we have another approximation for the true image $\mathbf{U}$ directly:

$$\begin{aligned}\underline{\mathbf{U}} \;&\approx\; \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\Big(\underline{\mathbf{U}} + \underline{\mathbf{U}}_{x_1}v_1 + \underline{\mathbf{U}}_{x_2}v_2 \\ &\qquad + \underline{\mathbf{U}}_{x_1^2}v_1^2 + \underline{\mathbf{U}}_{x_1x_2}v_1v_2 + \underline{\mathbf{U}}_{x_2^2}v_2^2 + \cdots\Big) \\ &=\; \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbb{I}\mathbb{U}\end{aligned} \tag{20}$$

with

$$\mathbb{I} = \Big[\mathbf{I}\;\; \mathbf{I}_{v_1}\;\; \mathbf{I}_{v_2}\;\; \mathbf{I}_{v_1^2}\;\; \mathbf{I}_{v_1v_2}\;\; \mathbf{I}_{v_2^2}\cdots\Big]. \tag{21}$$

The above approximation gives the regularization term with weights,

$$C_R(\mathbb{U}) = \sum_{v_1}\sum_{v_2}\left\|\underline{\mathbf{U}} - \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbb{I}\mathbb{U}\right\|_{\mathbf{W}_r(\mathbf{v})}^{m_r}, \tag{22}$$

where m_r is the error norm parameter and $\mathbf{W}_r(\mathbf{v})$ is the weight matrix for this term.

In summary, the overall optimization problem is formulated as

$$\begin{aligned}\min_{\mathbb{U}} C(\mathbb{U}) \;&=\; \min_{\mathbb{U}}\left\{C_L(\mathbb{U}) + \lambda C_R(\mathbb{U})\right\} \\ &=\; \min_{\mathbb{U}}\sum_{v_1}\sum_{v_2}\Bigg\{\left\|\underline{\mathbf{Y}} - \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbb{B}\mathbb{U}\right\|_{\mathbf{W}(\mathbf{v})}^{m} \\ &\qquad + \lambda\left\|\underline{\mathbf{U}} - \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\mathbb{I}\mathbb{U}\right\|_{\mathbf{W}_r(\mathbf{v})}^{m_r}\Bigg\},\end{aligned} \tag{23}$$

where λ is the regularization parameter. We solve the optimization problem using the steepest descent method:

$$\widehat{\mathbb{U}}^{(\ell+1)} = \widehat{\mathbb{U}}^{(\ell)} + \nu\left.\frac{\partial C(\mathbb{U})}{\partial \mathbb{U}}\right|_{\mathbb{U}=\widehat{\mathbb{U}}^{(\ell)}}, \tag{24}$$

where ν is the step size. When using the data-adapted kernel functions, we need to calculate the weight matrices $\mathbf{W}$ and $\mathbf{W}_r$ beforehand. In [2], we proposed an iterative way to refine the weights. However, in the case of using the steepest descent method here (or other iterative methods to minimize the cost function $C(\mathbb{U})$), after the initialization, we will update $\mathbb{U}$ and the weight matrices alternately in each iteration[2].

(a) Original (b) The blurred image

Fig. 1. The original image and the image blurred with a 5×5 Gaussian PSF with $\sigma = 1.5$. The corresponding RMSE is 11.48.

There are two points that we must not ignore about the regularization cost function (22). First, the regularization term is general enough to subsume several other popular regularization terms existing in the literature. In particular, when we choose the regression order $N = 0$, (22) becomes

$$\sum_{v_1}\sum_{v_2}\left\|\underline{\mathbf{U}} - \mathbf{S}_{x_1}^{-v_1}\mathbf{S}_{x_2}^{-v_2}\underline{\mathbf{U}}\right\|_{\mathbf{W}_r(\mathbf{v})}^{m_r}. \tag{25}$$

For $m_r = 2$ and $m_r = 1$, (25) can be regarded as Tikhonov and digital TV, respectively, with $\mathbf{W}_r(\mathbf{v}) = \mathbf{I}$, $|v_1| \leq 1$, and $|v_2| \leq 1$. Furthermore, for $m_r = 1$, the formulation (25) is well known as Bilateral Total Variation (BTV) [16] with $\mathbf{W}_r(\mathbf{v}) = \alpha^{|v_1|+|v_2|}\mathbf{I}$, where $0 \leq \alpha \leq 1$. That is to say, the kernel function for BTV is

$$K_\alpha(\mathbf{x}_i - \mathbf{x}) = \alpha^{|x_{1i}-x_1|+|x_{2i}-x_2|}, \tag{26}$$

where α is the global smoothing parameter in this case. Of course, other choices of the kernel function are also possible. In the literature, the data-adaptive versions of the kernel function have recently become popular for image restoration. The kernel (or weight) functions of Bilateral filter [12], [13], Mean-Shift [17], [18], and Non-Local Mean [19] are typical examples, and those kernel functions are usable for (23) as well.

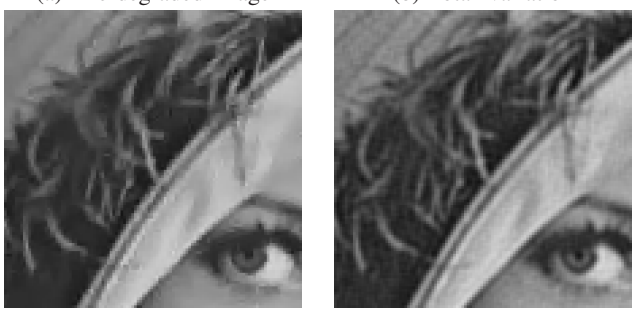
Second, all the related regularization terms and methods discussed above are zeroth order Taylor approximations (i.e. they neglect the higher order derivatives). Hence, the estimated images often tend to appear piecewise constant. On the other hand, the regularization term in (22) can naturally take higher order derivatives into account, resulting in estimated images with more detailed texture.

III. Experiments

Using the eye section of the Lena image, which is shown in Fig.1(a), we create a blurred image, shown in Fig.1(b),

[2]Alternatively, one can consider updating the weight matrices every few iterations.

(a) The degraded image (b) Total Variation



(c) ForWaRD [20] (d) Classic kernel deblurring

(e) Bilateral kernel deblurring (f) Steering kernel deblurring

Fig. 2. The deblurring simulation with small amount of noise: (a) the blurred noisy image (BSNR=40[dB]), (b) the image deblurred with total variation ($\lambda = 0.0001$), (c) ForWaRD [20], (d) the image deblurred with classic kernel ($N = 2, m = 2, m_r = 2, h = 0.25, \lambda = 0.5$), (e) the image deblurred with bilateral kernel ($N = 2, m = 2, m_r = 2, h = 0.25, h_r = 40.0, \lambda = 0.5$), and (f) the image deblurred with steering kernel ($N = 2, m = 2, m_r = 2, h = 0.15, \lambda = 0.5$, one iteration). The corresponding RMSE's are (a) 11.49, (b) 6.38, (c) 6.95 (d) 6.01, (e) 5.96, and (f) 5.99.

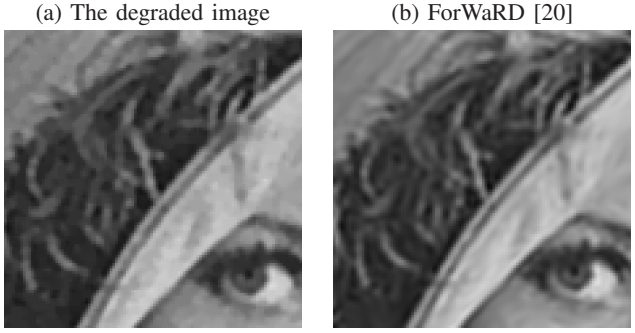
by convolving with a 5×5 Gaussian PSF with a standard deviation of 1.5. The resulting RMSE for the blurred image is 11.48. With the blurry Lena image, we did simulations with two different noise levels.

First, we added white Gaussian noise with $\mathrm{BSNR} = 40[\mathrm{dB}]$[3], which gives us the image shown in Fig.2(a). The estimated images by the TV method ($\lambda = 0.0001$), ForWaRD[4] [20], the classic kernel deblurring ($N = 2, m = 2, m_r = 2, h = 0.25, \lambda = 0.5$), the bilateral kernel deblurring ($N = 2, m = 2, m_r = 2, h = 0.25, h_r = 40.0, \lambda = 0.5$), and the steering kernel deblurring ($N = 2, m = 2, m_r = 2, h = 0.15, \lambda = 0.5$) are illustrated in Figs.2(b)-(f), respectively. The corresponding RMSE's are (a) 11.49, (b) 6.38, (c) 6.95 (d) 6.01, (e) 5.96, and (f) 5.99. In this case, the estimated images are (at least visually) very similar.

[3]Blurred Signal to Noise Ratio = $10 \log \left(\frac{\mathrm{var(blurred\ signal)}}{\mathrm{var(noise)}} \right)$ [dB]

[4]The software is available at http://www-dsp.rice.edu/software/ward.shtml. We used the default parameters which are suggested by the authors.

(a) The degraded image (b) ForWaRD [20]



(c) BTV [16] (d) Steering kernel deblurring

Fig. 3. The deblurring simulation with large amount of noise: (a) the blurred noisy image (BSNR=20[dB]), (b) ForWaRD [20], (c) Bilateral Total Variation [16], and (d) steering kernel deblurring ($N = 2, m = 2, m_r = 1, h = 1.4, \lambda = 0.5$). The corresponding RMSE's are (a) 12.36, (b) 9.55, (c) 8.91, and (d) 8.59.

Second, we set the amount of noise to $\mathrm{BSNR} = 20[\mathrm{dB}]$ to create the blurry and noisy image shown in Fig.3(a). The estimated image by ForWaRD [20], BTV [16], and the steering kernel deblurring ($N = 2, m = 2, m_r = 1, h = 1.4, \lambda = 0.5$) are illustrated in Fig.3(b), Fig.3(c), and Fig.3(d), respectively. The corresponding RMSE's are (a) 12.36, (b) 9.55, (c) 8.91, and (d) 8.59. The steering kernel is the best in this simulation.

IV. Conclusion and Future Work

This paper presented a simultaneous denoising and deblurring method based on the kernel regression framework, explained how it is related to other methods, and demonstrated its performance using simulated data sets. The simulations indicated the superiority of our methods in the most challenging scenarios; namely, for very noisy deblurring problems.

We applied the method for 2-dimensional data. However, it is also possible to generalize this method to deal with higher-dimensional data. Moreover, aside from single frame deblurring and denoising, the proposed method is also applicable to irregularly sampled data sets and to multi-frame deconvolution problems as well.

APPENDIX I
TAYLOR APPROXIMATION IN VECTOR FORM

The Taylor expansion in the 1-D case is

$$z(x_i) \approx z(x) + z'(x)(x_i - x) + \frac{1}{2}z''(x)(x_i - x) + \cdots . \quad (27)$$

When we have P positions of interest, $\underline{\mathbf{Z}} = [z(x_1), \cdots, z(x_P)]^T$ is a vector composed of the data set of interest. By defining $v = x_i - x$, the nearby values for each position are approximately expressed by using the Taylor expansion as

$$\begin{bmatrix} z(x_1+v) \\ \vdots \\ z(x_P+v) \end{bmatrix} \approx \begin{bmatrix} z(x_1) \\ \vdots \\ z(x_P) \end{bmatrix} + \begin{bmatrix} z'(x_1) \\ \vdots \\ z'(x_P) \end{bmatrix} v + \begin{bmatrix} z''(x_1) \\ \vdots \\ z''(x_P) \end{bmatrix} \frac{v^2}{2} + \cdots \quad (28)$$

which in vector form is

$$\mathbf{S}^v \underline{\mathbf{Z}} \approx \underline{\mathbf{Z}} + \underline{\mathbf{Z}}' v + \underline{\mathbf{Z}}'' \frac{v^2}{2} + \cdots , \quad (29)$$

where $\mathbf{S}^v$ is the warping operator. Finally, we have the Taylor expansion in matrix form

$$\underline{\mathbf{Z}} \approx \mathbf{S}^{-v} \left(\underline{\mathbf{Z}} + \underline{\mathbf{Z}}' v + \underline{\mathbf{Z}}'' \frac{v^2}{2} + \cdots \right). \quad (30)$$

REFERENCES

[1] M. P. Wand and M. C. Jones, *Kernel Smoothing*, ser. Monographs on Statistics and Applied Probability. London; New York: Chapman and Hall, 1995.

[2] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," August 2006, accepted to IEEE Transactions on Image Processing.

[3] ——, "Robust kernel regression for restoration and reconstruction of images from sparse, noisy data," *Proceedings of the International Conference on Image Processing, Atlanta, GA*, pp. 1257–1260, October 2006.

[4] V. Katkovnic, K. Egiazarian, and J. Astola, "A spatially adaptive nonparametric regression image deblurring," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1469–1478, October 2005.

[5] S. Osher and L. I. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM Journal on Numerical Analysis*, vol. 27, no. 4, pp. 919–940, August 1990.

[6] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268.

[7] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM Journal Scientific Computing*, no. 1, pp. 227–238, January 1996.

[8] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Transactions of Image Processing*, vol. 10, no. 2, pp. 231–241, February 2001.

[9] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *SIAM Multiscale Model. and Simu*, vol. 4, pp. 460–489, 2005.

[10] D. C. Dobson and F. Santosa, "Recovery of blocky images from noisy and blurred data," *SIAM Journal on Applied Mathematics*, vol. 56, no. 4, pp. 1181–1198, August 1996.

[11] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, ser. Monographs on Statistics and Applied Probability. London; New York: Chapman and Hall, 1986.

[12] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceeding of the 1998 IEEE International Conference of Compute Vision, Bombay, India*, pp. 836–846, January 1998.

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

[14] 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, Article ID 83268*, 2006.

[15] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA*, November 2002.

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

[17] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions of Information Theory*, pp. 32–40, 1975.

[18] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 603–619, May 2002.

[19] A. Buades, B. Coll, and J. M. Morel, "A review of image denosing algorithms, with a new one," *Multiscale Modeling and Simulation (SIAM interdisciplinary journal)*, vol. 4, no. 2, pp. 490–530, 2005.

[20] R. Neelamani, H. Choi, and R. Baraniuk, "ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 418–433, February 2004.