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, based on 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 $x_i$. Specifically, if $x$ is near the sample at $x_i$, we have the $(N+1)$-term Taylor series
\begin{equation}
z(x) \approx z(x_i) + \{\nabla z(x_i)\}^T (x_i - x) + \frac{1}{2} (x_i - x)^T \{\mathcal{H} z(x_i)\} (x_i - x) + \cdots
\end{equation}
\begin{equation}
= \beta_0 + \beta_1^T (x_i - x) + \beta_2^T \text{vech} \{ (x_i - x)(x_i - x)^T \} + \cdots
\end{equation}
where $\nabla$ and $\mathcal{H}$ are the gradient $(2 \times 1)$ and Hessian $(2 \times 2)$ operators, respectively, and $\text{vech} (\cdot)$ is the half-vectorization operator which lexicographically orders the lower triangular portion of a symmetric matrix. Furthermore, $\beta_0$ is $z(x)$, which is the pixel value of interest, and the vectors $\beta_1$ and $\beta_2$ are
\begin{equation}
\beta_1 = \left[ \frac{\partial z(x)}{\partial x_1}, \frac{\partial z(x)}{\partial x_2} \right]^T,
\end{equation}
\begin{equation}
\beta_2 = \left[ \frac{\partial^2 z(x)}{2 \partial x_1^2}, \frac{\partial^2 z(x)}{\partial x_1 \partial x_2}, \frac{\partial^2 z(x)}{2 \partial x_2^2} \right]^T.
\end{equation}
Since this approach is based on local approximations, a logical step to take is to estimate the parameters $\{\beta_n\}_{n=0}^N$.

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
\begin{equation}
y_i = z(x_i) + \varepsilon_i, \quad i = 1, \cdots, P, \quad x_i = [x_{1i}, x_{2i}]^T,
\end{equation}
where $y_i$ is a noisy sample at $x_i$, $z(\cdot)$ is the (hitherto unspecified) regression function to be estimated, $\varepsilon_i$ 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.

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.

1Other localized representations are also possible and may be advantageous.
<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td></td>
<td>00-00-2006 to 00-00-2006</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regularized Kernel Regression for Image Deblurring</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5a. CONTRACT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5b. GRANT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5d. PROJECT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5e. TASK NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5f. WORK UNIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of California Santa Cruz, Electrical Engineering Department, 1156 High Street, Santa Cruz, CA, 95064</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release; distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT unclassified</td>
</tr>
<tr>
<td>b. ABSTRACT unclassified</td>
</tr>
<tr>
<td>c. THIS PAGE unclassified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
<th>19. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
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,

\[
\min_{\{\beta_i\}_{i=0}^N} \sum_{i=1}^P |y_i - \beta_0 - \beta_1^T (x_i - x) \]

\[-\beta_2^T \text{vec} \{ (x_i - x)(x_i - x)^T - \cdots \} \]

\[\approx K_{H_i}(x_i - x)(6)\]

with

\[K_{H_i}(x_i - x) = \frac{1}{\det(H_i)} K(H_i^{-1}(x_i - x)), \quad (7)\]

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 \( H_i \) is the smoothing \((2 \times 2)\) 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:**

\[K_{H_i}(x_i - x), \quad \text{with} \quad H_i = hI, \quad (8)\]

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:**

\[K_{H_i}(x_i - x)K_{h_r}(y_i - y), \quad \text{with} \quad H_i = hI, \quad (9)\]

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:**

\[K_{H_i}^{\text{steer}}(x_i - x), \quad \text{with} \quad H_i^{\text{steer}} = hC_i^1, \quad (10)\]

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

\[C_i \approx \left[ \sum_{x_j \in \Omega_1} z_{x_1}(x_j)z_{x_1}(x_j) \sum_{x_j \in \Omega_2} z_{x_2}(x_j)z_{x_2}(x_j) \sum_{x_j \in \Omega_1} z_{x_1}(x_j)z_{x_2}(x_j) \sum_{x_j \in \Omega_2} z_{x_1}(x_j)z_{x_2}(x_j) \right], \quad (11)\]

where \( z_{x_1}(\cdot) \) 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 \( b(x) \) and the desired true function as \( u(x) \), we consider the blurred and noise-ridden data model

\[y = z(x) + \varepsilon = b(x) \ast u(x) + \varepsilon, \quad (12)\]

where \( \ast \) is the convolution operator. For convenience, we write the point-wise model in vector form as:

\[Y = Z + \varepsilon = BU + \varepsilon.\]

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

\[Z \approx S_{x_1}^{-v_1}S_{x_2}^{-v_2} \left( Z + Z_{x_1}v_1 + Z_{x_2}v_2 \right. \]

\[+ Z_{x_1}v_1^2 + Z_{x_1x_2}v_1v_2 + Z_{x_2}v_2^2 + \cdots \]

\[= S_{x_1}^{-v_1}S_{x_2}^{-v_2}B \left( U + U_{x_1}v_1 + U_{x_2}v_2 \right. \]

\[+ U_{x_1}v_1^2 + U_{x_1x_2}v_1v_2 + U_{x_2}v_2^2 + \cdots \]

\[= S_{x_1}^{-v_1}S_{x_2}^{-v_2}BU \quad (15)\]

with

\[B = \left[ \begin{array}{cccc} B & BI_{v_1} & BI_{v_2} & BI_{v_1v_2} & BI_{v_1} & BI_{v_1} \end{array} \right], \]

\[U = \left[ \begin{array}{cccc} U & U_T & U_{x_1} & U_{x_1} & U_{x_1} & U_{x_1} \end{array} \right]^T, \quad (16)\]

where \( S_{x_1}^{-v_1} \) is the \( v_1 \)-pixel shift operator along the \( x_1 \) direction, \( Z_{x_1} \) and \( Z_{x_2} \) are the first and the second derivative in the
\( x_1 \) direction respectively, and \( I_{v_1} = \text{diag}\{v_1, \cdots, v_1\} \). In the absence of other modeling errors and noise, the approximation suggests the following constraint:

\[
Z - S^{-v_1} x_1 S^{-v_2} x_2 B U = 0. \tag{17}
\]

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 \( v_1, v_2 \):

\[
C_L(U) = \sum_{v_1} \sum_{v_2} \left\| Y - S^{-v_1} x_1 S^{-v_2} x_2 B U \right\|^m_{W(v)}, \tag{18}
\]

where \( v = [v_1, v_2]^T \), \( m \) is the error norm parameter and

\[
W(v) = \text{diag} \{K_{H_1}(v), \cdots, K_{H_r}(v)\}. \tag{19}
\]

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 \( U \) directly:

\[
U \approx S^{-v_1} x_1 S^{-v_2} x_2 \left( U + U_{v_1} + U_{v_2} \right)
+ U_{v_1 v_1} + \cdots \tag{20}
\]

The above approximation gives the regularization term with weights,

\[
C_R(U) = \sum_{v_1} \sum_{v_2} \left\| U - S^{-v_1} x_1 S^{-v_2} x_2 U \right\|^{m_r}_{W_r(v)}, \tag{22}
\]

where \( m_r \) is the error norm parameter and \( W_r(v) \) is the weight matrix for this term.

In summary, the overall optimization problem is formulated as

\[
\min_U C(U) = \min_U \{C_L(U) + \lambda C_R(U)\}
= \min_U \sum_{v_1} \sum_{v_2} \left\{ \left\| Y - S^{-v_1} x_1 S^{-v_2} x_2 B U \right\|^{m_r}_{W(v)} \right. \\
+ \left. \lambda \left\| U - S^{-v_1} x_1 S^{-v_2} x_2 U \right\|^{m_r}_{W_r(v)} \right\}, \tag{23}
\]

where \( \lambda \) is the regularization parameter. We solve the optimization problem using the steepest descent method:

\[
\hat{U}^{(t+1)} = \hat{U}^{(t)} + \nu \frac{\partial C(U)}{\partial U} \bigg|_{U = \hat{U}^{(t)}}, \tag{24}
\]

where \( \nu \) is the step size. When using the data-adapted kernel functions, we need to calculate the weight matrices \( W \) and \( W_r \) 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 \( C(U) \)), after the initialization, we will update \( U \) and the weight matrices alternately in each iteration\(^2\).

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

\[
\sum_{v_1} \sum_{v_2} \left\| U - S^{-v_1} x_1 S^{-v_2} x_2 U \right\|^{m_r}_{W_r(v)}. \tag{25}
\]

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

\[
K_\alpha(x_1 - x) = \alpha |x_1 - x_1| + |x_2 - x_2|, \tag{26}
\]

where \( \alpha \) 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.
by convolving with a $5 \times 5$ 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 BSNR = 40[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 ($N = 2, m = 2, m_r = 1, h = 1.4, \lambda = 0.5$), 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.

$^3$Blurred Signal to Noise Ratio $= 10 \log \left(\frac{\text{var(blurred signal)}}{\text{var(noise)}}\right)$ [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.

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. In this case, the estimated images are (at least visually) very similar.

Second, we set the amount of noise to BSNR = 20[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 Figs.3(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.

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

When we have \( P \) positions of interest, \( Z = [z(x_1), \ldots, z(x_P)]^T \) 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

\[
S^vZ \approx Z + Z'v + Z''\frac{v^2}{2} + \cdots, \quad (29)
\]

where \( S^v \) is the warping operator. Finally, we have the Taylor expansion in matrix form

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