Adaptive Regression Kernels for Image/Video Restoration and Recognition

Peyman Milanfar et al. *
Department of Electrical Engineering
University of California, Santa Cruz, CA 95064

Abstract

I will described a nonparametric framework for locally adaptive signal processing and analysis. This framework is based upon the notion of Kernel Regression which we generalize to adapt to local characteristics of the given data, resulting in descriptors which take into account both the spatial density of the samples ("the geometry"), and the actual values of those samples ("the radiometry"). These descriptors are exceedingly robust in capturing the underlying structure of the signals even in the presence of significant noise, missing data, and other disturbances. As the framework does not rely upon strong assumptions about noise or signal models, it is applicable to a wide variety of problems. On the processing side, I will illustrate examples in two and three dimensions including state of the art denoising, upscaling, and deblurring. On the analysis side, I will describe the application of the framework to training-free object detection in images, and action detection in video, from a single example.

OCIS Codes:100.2000 (Digital Image Processing); 100.3008 (Image recognition, algorithms and filters)

1 Steering Kernel Regression (SKR)

We first review the fundamental framework of *kernel regression* [3] and then describe its novel extension, the steering kernel regression (SKR), in 2-D. The extension to 3-D is straightforward. The KR framework defines its data model as

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

where y_i is a noisy sample at $\mathbf{x}_i$ (Note: x_{1i} and x_{2i} are spatial coordinates), $z(\cdot)$ 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 an arbitrary "window" around a position $\mathbf{x}$ 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 particular form of $z(\cdot)$ may remain unspecified, we can develop 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-th order Taylor series

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

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

$$\boldsymbol{\beta}_1 = \left[\frac{\partial z(\mathbf{x})}{\partial x_1}, \quad \frac{\partial z(\mathbf{x})}{\partial x_2}\right]^T, \qquad \boldsymbol{\beta}_2 = \frac{1}{2}\left[\frac{\partial^2 z(\mathbf{x})}{\partial x_1^2}, \quad 2\frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2}, \quad \frac{\partial^2 z(\mathbf{x})}{\partial x_2^2}\right]^T. \tag{3}$$

Since this approach is based on *local* signal representations, a logical step to take is to estimate the parameters $\{\boldsymbol{\beta}_n\}_{n=0}^N$ from all the neighboring samples $\{y_i\}_{i=1}^P$ while giving the nearby samples higher weights than samples farther away. A (weighted) least-square formulation of the fitting problem capturing this idea is to solve the following optimization problem,

$$\min_{\{\boldsymbol{\beta}_n\}_{n=0}^N} \sum_{i=1}^{P} \left[y_i - \beta_0 - \boldsymbol{\beta}_1^T (\mathbf{x}_i - \mathbf{x}) - \boldsymbol{\beta}_2^T \mathrm{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} - \cdots\right]^2 K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) \tag{4}$$

*This is joint work with Hiro Takeda and Hae Jong Seo; and was supported by US Air Force Grant F9550-07-1-0365. (e-mail contact: milanfar@ee.ucsc.edu)

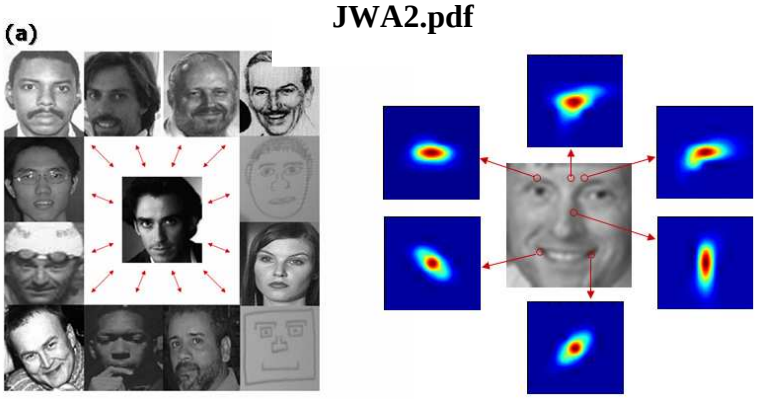

Figure 1: (a) Faces in various colors, lighting condition, occlusion, in-plane and out-of-plane rotation, and scale (b) Steering kernel weights at several locations in an image. The SKR weights can be used not only to process and enhance an image, but also as visual descriptors (features) to enable very effective recognition.

with

$$K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) = \frac{1}{\det(\mathbf{H}_i)} K\left(\mathbf{H}_i^{-1}(\mathbf{x}_i - \mathbf{x})\right), \tag{5}$$

where N is the regression order, $K(\cdot)$ is the kernel function (radially symmetric function such as a Gaussian), and $\mathbf{H}_i$ is the smoothing (2×2) matrix which dictates overall shape of the resulting weight function. The shape of the final weight kernels is perhaps the most important factor in determining the quality of estimated signals. Namely, it is desirable to have kernels that adapt themselves to the local structure of the measured signal, providing, for instance, strong filtering along an edge rather than across it. This last point is indeed the motivation behind the *steering* KR framework which we will describe in this paper.

Returning to the optimization problem (4), regardless of the regression order and the dimensionality of the regression function, the estimate of the signal (i.e. pixel) value of interest β_0 is given by a weighted *linear* combination of the nearby samples:

$$\hat{z}(\mathbf{x}) = \hat{\beta}_0 = \sum_{i=1}^{P} W_i(K, \mathbf{H}_i, N, \mathbf{x}_i - \mathbf{x})\, y_i, \quad \sum_{i=1}^{P} W_i(\cdot) = 1, \tag{6}$$

where we call W_i the *equivalent kernel* function for y_i (q.v. [3] for the derivation).

What we described above is the "classic" kernel regression framework, which yields a point-wise estimator that is always a local *linear* (though not necessarily space-variant) combination of the neighboring samples. As such, it suffers from an inherent limitation. In the next section, we describe the framework of *steering* KR, in which the kernel weights themselves are computed from the local window, and therefore we arrive at filters with more complex (nonlinear and space-variant) action on the data.

The steering kernel framework is based on the idea of robustly obtaining local signal structures (e.g. edges in 2-D and planes in 3-D) by analyzing the radiometric (pixel value) differences locally, and feeding this structure information to the kernel function in order to affect its shape and size.

Consider the (2×2) smoothing matrix $\mathbf{H}_i$ in (5). In the generic "classical" case, this matrix is a scalar multiple of the identity. This results in kernel weights which have equal effect along all the x_1- and x_2-directions. However, if we properly choose this matrix, the kernel function can capture local structures. More precisely, we define the smoothing matrix as a symmetric matrix

$$\mathbf{H}_i = \mathbf{C}_i^{-\frac{1}{2}}, \tag{7}$$

which is called the *steering* matrix, and where the matrix $\mathbf{C}_i$ is estimated as the local covariance matrix of the neighborhood spatial gradient vectors as follows:

$$\mathbf{J}_i = \begin{bmatrix} \vdots & \vdots \\ z_{x_1}(\mathbf{x}_j) & z_{x_2}(\mathbf{x}_j) \\ \vdots & \vdots \end{bmatrix}, \quad \mathbf{x}_j \in w_i \quad \longrightarrow \quad \widehat{\mathbf{C}}_i = \mathbf{J}_i^T \mathbf{J}_i. \tag{8}$$

where $z_{x_1}(\cdot)$ and $z_{x_2}(\cdot)$ are the first derivatives along x_1- and x_2-axes, and w_i is a local analysis window around a sample position at $\mathbf{x}_i$. As illustrated in Fig. 1b, it is important to note that since $\mathbf{H}_i$ is different for each pixel i, the shape of the resulting weight function will not be a simple Gaussian with with elliptical contours.

With the above choice of the smoothing matrix, a Gaussian kernel, we now have the steering kernel function as

$$K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) = \sqrt{\det(\mathbf{C}_i)} \exp\left\{ -\left\| \mathbf{C}_i^{\frac{1}{2}} (\mathbf{x}_i - \mathbf{x}) \right\|_2^2 \right\}. \quad (9)$$

2 Applications to Restoration and Recognition

Restoration [2]: This is an example of simultaneous denoising and upscaling with the Foreman sequence. We enhanced and upscaled this sequence beyond its native spatial resolution by a factor of 2 (from QCIF (144×176) to CIF (288×352)). Two sample frames of the video are shown in Figure 2. As seen in the input frames, the video is compressed and carries some noise. We note that both the type of compression and the noise statistics are unknown to our method. The upscaled video by factor of 2 using Lanczos interpolation and (3-D) SKR methods are shown in Figures 2 in the next two columns.

Recognition [1]: The generic problem of interest here is this: We are given a single "query" or "example" image of an object of interest (for instance a picture of a face), and we are interested in detecting similar objects within other "target" images with which we are presented. The target images may contain such similar objects (say other faces) but these will generally appear in completely different context and under different imaging conditions. Examples of such differences can range from rather simple optical or geometric differences (such as occlusion, differing viewpoints, lighting, and scale changes); to more complex inherent structural differences such as for instance a hand-drawn picture of a face rather than a real face. (See Figure 1a.) We can essentially use (a normalized version of) the function $K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x})$ to represent an image's inherent local geometry; and from this function we extract features which will be used to compare the given patch against patches from another image. This approach has been successfully applied to detection of varied objects in both images and video.

Figure 2: Left: Restoration example. A video upscale example using a real video sequence: Column 1 shows 2 frames from the original video; next column shows the upscaled frames by Lanczos interpolation, and the third column shows the upscaled frames by 3-D SKR. Right: Recognition example. Note that the template example does not actually appear in any of the target images. Colored contours show the center of the detected matching regions.

3 References

[1] H. Seo and P. Milanfar. Training-free, generic object detection using locally adaptive regression kernels. *IEEE Trans. on Pattern Analysis and Machine Intell.*, 2009. In review.

[2] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Trans. on Image Proc.*, 16(2):349–366, Feb. 2007.

[3] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1995.