Support Vector Regression based Image Restoration*

Pankaj Kumar Sa and **Banshidhar Majhi**
Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India
{PankajKSa, BMajhi}@nitrkl.ac.in

Abstract

The point spread functions (PSF) responsible for degrading the observed images are very often not known. Hence, the image must be restored only from the available noisy blurred observation. This paper proposes two new image restoration algorithms, which are based on support vector regression (SVR). The first algorithm uses local variance and the second algorithm utilizes the concepts of fuzzy systems to counter blur in a given image. These algorithms significantly reduce the training time through their effective sample selection mechanisms. Experimental findings show that the proposed techniques deliver superior results for a variety of blurs and PSFs.

Image restoration, point spread function (PSF), support vector regression (SVR), fuzzy systems.

1 Introduction

Blurring in images may be introduced by relative object–camera motion, lens defocussing, atmospheric turbulence etc. Such degraded images are usually approximated as a two-dimensional convolution of original image with a linear shift invariant (LSI) blur, which is otherwise known as point spread function (PSF) in the presence of additive noise. If f is the original image, h is the PSF, and η is the additive noise then the degraded image g can be represented as —

$$g(x, y) = f(x, y) * h(x, y) + \eta(x, y) \tag{1}$$

where (x, y) represents the spatial coordinate of each pixel of the image [1, 2, 3].

Image restoration emphasizes on getting back the original image, as far as possible, from the degraded observation and the recovery process involves an inverse process termed as *deconvolution* [2]. The accuracy of the degradation model and the design of the filter decides the effectiveness of the deconvolution process. However, in many practical situations the degradation model (PSF) is often unknown, or little information is available about the original image. Therefore, the image f must be estimated from the convolved image g using either partial or no information about the blurring process and true image. Such an estimation problem, assuming the linear degradation model, is called blind deconvolution [3].

Blind image deconvolution has been widely studied by researchers. Ayers *et al.* [4] proposed an iterative blind deconvolution (IBD) scheme to estimate the original image and the PSF. Their scheme uses the spatial as well as frequency domain constraints. The iterative loop involves very simple calculations, and is repeated until the restored image is found to be satisfactory. However, the output is not definite;

*Presented in IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013), July 7–10, 2013, Hyderabad, India.

the algorithm may run into infinite loop without converging. In order to get the approximation of the true image, the initial estimate of the PSF should not vary much with the original PSF, which itself is a difficult task to realize in practical situation. The IBD algorithm can be made robust to noise by using the Lucy–Richardson (LR) algorithm based on maximum likelihood estimation (MLE) [5, 6]. This combined approach, known as adaptive LR maximum likelihood algorithm, has a very high computational complexity. In another work, Kundur *et al.* [7] modeled the image restoration as optimization problem in their work (NASRIF). The blurred image is input to a two-dimensional variable coefficient FIR filter whose output represents an estimate of the true image. This estimate is passed through a nonlinear constraint process that uses a non-expansive mapping to project the estimated image into the space representing the known characteristics of the true image. The difference between the projected image and the estimated image is used as an error signal to update the coefficients of the inverse filter. This algorithm is a good performer under high SNR and requires the background of the image to be of uniform intensity, which is unlikely in many situations. It also shows some noise amplification at low SNR. In a recent work, Li *et al.* [8] have suggested a principal component analysis based blind image deconvolution algorithm (BIDPCA). They have used the principle of variance maximization to deblur images degraded with atmospheric turbulence. Even though PCA is a multichannel blind image deconvolution method, the authors have applied it to a single channel by creating an ensemble of images, obtained by shifting the blurred image. Improper selection of PSF support produces artifacts in the restoration result. In another recent approach the same authors [9] have used support vector regression (SVR) to obtain an optimized mapping from predefined neighborhood in a degraded image to the central pixel in the original image. The SVR is first trained with sample observations to generate support vectors and later used to estimate the image on pixel by pixel basis. A small deviation in blur characteristics might cause a large error in the restoration result.

Blurring in images can be visualized as a low-pass filtering of original image. Also, not all blurs are same; they are very much different in terms of their parameters and point spread function (PSF) support. Images degraded by different blurs look significantly dissimilar from each other. However, in this dissimilarity there exists some commonality in the block levels. Small patches or blocks in one image bear some likeliness to patches in another image. This characteristic has been utilized in image compression based on vector quantization, where several source outputs are grouped together and encoded in a codebook as a single block [10, 11].

In this paper, features are extracted from diverse images in such a way that they bear some resemblance. Then a common model is regressed from these features through support vectors to restore images degraded with different blurs with different PSF. Two algorithms are suggested in this regard. The first algorithm is based on local variance of the image and is named as V-SVR. The second algorithm, F-SVR, is based on the concepts of fuzzy systems. Rest of the paper is organized as follows.

Two blur models are discussed in Section 2. The support vector regression (SVR) is outlined in Section 3. The proposed algorithms are described in Section 4. Simulation results are presented in the Section 5. Finally, Section 6 provides the concluding remarks.

2 Blur Models

Blurring process smoothens the sharp transitions of gray levels in an image. It also decreases the contrast by averaging the pixels. The characteristics of out-of-focus blur and motion blur are discussed below.

2.0.1 *Out-of-Focus Blur*

Out-of-focus blur can be conveyed by the image's circle of confusion. While imaging a point source, if the rays of light from a lens do not converge to focus, the optical spot thus created is called circle of confusion or blur circle. Generally, the out-of-focus area takes the shape of a round disk having radius equal to that of the circular aperture. If R is the radius, the PSF can be described as —

$$h(x,y) = \begin{cases} \frac{1}{\pi R^2} & \text{if } \sqrt{x^2+y^2} \leq R, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The corresponding frequency zeros are concentric circles about the origin that are nearly periodic in R.

2.0.2 *Motion Blur*

Suppose that a scene to be recorded undergoes a planar motion relative to the sensor. Assume the relative motion to be of uniform velocity v at an angle θ with the horizontal axis. If T is the duration of exposure then the blur length is $L = vT$ and the motion blur PSF may be expressed as —

$$h(x,y) = \begin{cases} \frac{1}{L} & \text{if } 0 \leq |x| \leq L\cos\theta; y = L\sin\theta, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

For horizontal motion ($\theta = 0$), the above expression reduces to —

$$h(x,y) = \begin{cases} \frac{1}{L} & \text{if } 0 \leq |x| \leq L; y = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

3 Support Vector Regression

The maximal margin technique of the support vector method [12] can also be used for regression. A nonlinear function is mapped into a high-dimensional feature space via a nonlinear mapping. A linear learning machine is then used to learn the nonlinear function in the new space by controlling the system parameter, which is independent of dimensionality of the space [13, 14]. A brief review on support vector regression (SVR) theory is presented in this section.

Consider a training set —

$$(X_1; y_1), (X_2; y_2), \ldots, (X_N; y_N)$$

where, $X_i \in \mathbb{R}^n$ are the input vectors and $y_i \in \mathbb{R}$ is the target output of X_i, $i = 1, 2, \ldots, N$.

In ε-SVR [12], a linear function

$$f(X) = \langle w, X \rangle + b \tag{5}$$

is to be determined, with $w \in \mathbb{R}^n$ as high-dimensional weight vector and $b \in \mathbb{R}$ as the bias, such that there shall be no penalty for error less than ε, and also $f(X)$ should be flat. The meaning of flatness is that the value of w should be as small as possible. This can be achieved by minimizing the norm of w. This formulation, thus, can be expressed as a convex optimization problem as —

$$\text{minimize } \frac{1}{2} \parallel w \parallel^2 \tag{6}$$

subject to —

$$\begin{aligned} (\langle w, X_i \rangle + b) - y_i &\leq \varepsilon \\ y_i - (\langle w, X_i \rangle + b) &\leq \varepsilon \end{aligned}$$

The function f should be approximated in such a way that all pairs (X_i, y_i) will fall with ε precision. However, sometimes some errors need to be allowed. By introducing slack variables ζ_i and $\hat{\zeta}_i$, the optimization problem (6) boils down to —

$$\text{minimize } \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{N} \left(\zeta_i + \hat{\zeta}_i \right) \tag{7}$$

subject to —

$$\begin{aligned} (\langle w, X_i \rangle + b) - y_i &\leq \varepsilon + \zeta_i \\ y_i - (\langle w, X_i \rangle + b) &\leq \varepsilon + \hat{\zeta}_i \\ \zeta_i,\ \hat{\zeta}_i &\geq 0 \\ i = 1, 2, \ldots, N & \end{aligned}$$

One of the slack variables is for exceeding the target by more than ε, where as the other one is for falling below the target by more than ε. There exists a trade-off between the flatness of f and tolerance ε, *i.e.*, margin and error. This is being determined by the constant $C > 0$. This is termed as ε-insensitive loss function, which is expressed as —

$$|\zeta|_\varepsilon = \begin{cases} 0 & \text{if } |\zeta| \leq \varepsilon \\ |\zeta| - \varepsilon & \text{otherwise} \end{cases} \tag{8}$$

By simplifying the above optimization problem (7) through primal and dual expression the support vector expansion can be formulated as —

$$f(X) = \sum_{i=1}^{N} (\alpha_i - \hat{\alpha}_i) \langle X_i, X \rangle + b \tag{9}$$

where α_i and $\hat{\alpha}_i$ are the Lagrangian multipliers and the training samples for which $(\alpha_i - \hat{\alpha}_i) \neq 0$ are called support vectors (SVs). The complexity of the function's representation depends only on the number of SVs and is independent of the dimensionality of the input space.

In order to regress nonlinear problems, the above linear SVR (9) need to be extended to nonlinear SVR by mapping the input space into high dimensional feature space through a nonlinear operator ϕ. The SV expansion may then be deduced as —

$$\begin{aligned} f(X) &= \sum_{i=1}^{N} (\alpha_i - \hat{\alpha}_i) \langle \phi(X_i), \phi(X) \rangle + b \\ &= \sum_{i=1}^{N} (\alpha_i - \hat{\alpha}_i) \mathcal{K} \langle X_i, X \rangle + b \end{aligned} \tag{10}$$

where, $\mathcal{K}\langle X_i, X \rangle = \langle \phi(X_i), \phi(X) \rangle$ is the kernel function like radial basis, polynomial etc. Radial basis kernel is used in the proposed algorithms.

4 Proposed Algorithms

In this paper, two schemes on image deblurring based on SVR are presented. The schemes look for similar features in small blocks, which are not apparent in degraded images wholly. A function is then regressed that can treat different blurs. In the existing literature, however, training time is the major bottleneck [9]. To overcome this limitation, both the proposed schemes reduce the training sample size by clustering them. The clustering mechanism is based on the premise that edges in an image convey more information than smooth area. Statistically, variance of edge regions is more than that of smooth

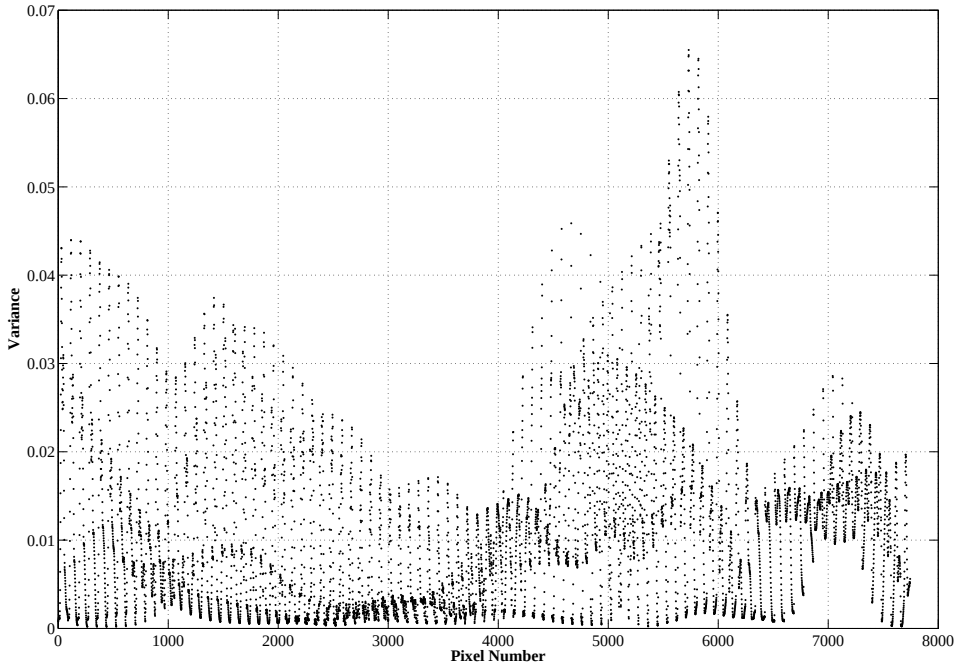


Figure 1: Variance plot of *Lena* image of size 100×100 degraded with out-of-focus blur of $R = 3$.

regions. This characteristic of image has been exploited while clustering the training image. In the first proposition the training image is clustered on the basis of local variance.

Integration of fuzzy systems and SVR is becoming very popular nowadays [15, 16, 17, 18, 19, 20, 21, 22, 23]. In the second proposition, fuzzy systems is introduced into SVR, where every cluster has some amount of contribution (membership) on the output. Both the proposed schemes help in reducing the number of parameters to be trained and thereby decreasing the training time substantially. Each scheme is discussed below in detail.

4.1 Image Restoration through V-SVR

The algorithm consists of three principal steps; Clustering, Training and Restoration. Each of them are explained in sequel.

4.1.1 V-SVR Clustering

A variance matrix of a training image is constructed in this step. To understand the clustering mechanism, consider the *Lena* image (down sampled to 100×100) degraded with out-of-focus blur with radius $R = 3$. The local variance of each of the $(2m + 1) \times (2m + 1)$ neighborhood of the blurred training image pixel $g(x, y)$ is computed. Considering $m = 3$, variance of each of the 7×7 window is computed and the plot of center pixel versus variance is shown in Figure 1. For convenience, the boundary pixels are ignored in the plot. It is observed that maximum numbers of pixels are having smaller variance. They correspond to the smooth regions and represent redundant information, whereas larger variance corresponds to edge region. The training set is constructed with few samples from smaller variance and more samples from larger variance. Randomly 5% of the pixels with local variance in the $(0, 0.01]$ range, 10% from $(0.01, 0.02]$ range, 20% from $(0.02, 0.03]$ range, and 50% from $(0.03, 0.04]$ range are selected. All the pixels above 0.04 are also selected. Even though the percentages are different, the number of pixels selected in each range is almost same as the pixel count decreases with the increase in variance.

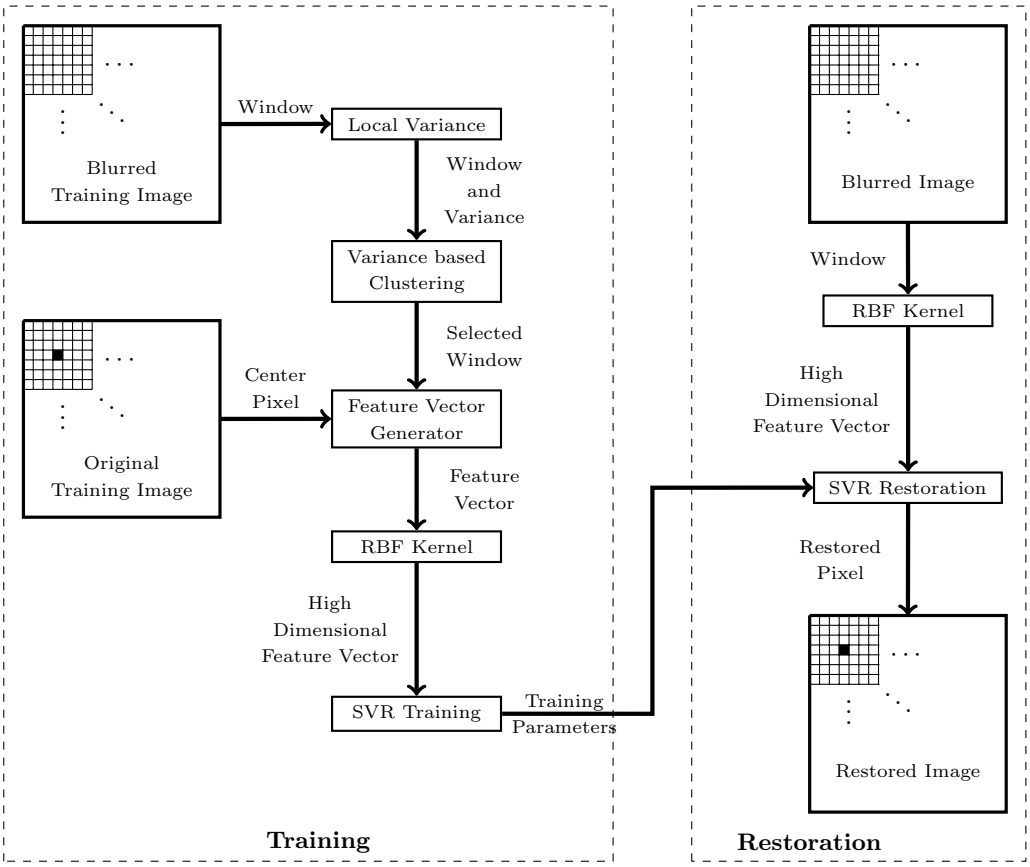


Figure 2: Block diagram of V-SVR restoration.

4.1.2 V-SVR Training

The 7×7 neighborhood of the clustered training image pixel $g(x, y)$ is converted to a row vector of size 1×49. This vector serves as the input vector X_i for the SVR and corresponding true pixel of $f(x, y)$ is the target pixel y_i. A training set is generated by shifting this window from left-top to right-bottom on each of the ensemble of blurred images.

The ε-insensitive loss function is used for the soft margin and the radial basis function is used as the nonlinear kernel for mapping the input data to a higher dimension. The training for the input data is performed using the LibSVM tool and the support vectors are recorded for subsequent testing.

4.1.3 V-SVR Restoration

Restoration of a degraded image is performed on a pixel-by-pixel basis. The same 7×7 neighborhood is used to construct the input vector from the given blurred image. The trained SVR is then used to approximate each pixel value. Performance of the scheme is analysed in Section 5. The sequence of operations followed for training as well as restoration is depicted in the block diagram (Figure 2).

4.2 Image Restoration through F-SVR

This algorithm also consists of the same three principal steps; Clustering, Training and Restoration. Unlike V-SVR, the clustering algorithm in F-SVR uses fuzzy systems.

Consider a Takagi-Sugeno fuzzy (TS) model with two fuzzy sets (A), two input variables (X), and a

consequent (f'). Following is the structure of the rule —

$$\begin{aligned} \text{Rule } j: \quad & \text{IF } x_1 \text{ is } A_{j1} \text{ AND } x_2 \text{ is } A_{j2} \\ & \text{THEN } f' = \sum_{k=1}^{2} p_{jk} x_k + b \end{aligned} \tag{11}$$

where b is a bias term introduced into the TS fuzzy rule to make it congruent with the SVR. The polynomial coefficients are represented by p_{jk} (or with a vector notation $\vec{p}_j$). The fuzzy sets A_{jk} are governed by the following Gaussian membership function —

$$M_{jk}(x_k) = \exp\left[-\frac{(x_k - c_{jk})^2}{\sigma_{jk}^2}\right] \tag{12}$$

where c_{jk} (or $\vec{c}_j$) denotes the center and σ_{jk} (or $\vec{\sigma}_j$) denotes the spread. The firing strength of j^{th} rule is computed as —

$$\mu_j(X) = \prod_{k=1}^{2} M_{jk}(x_k) \tag{13}$$

For r number of rules, the output is expressed with weighted sum method as —

$$\begin{aligned} f'(X) &= \sum_{j=1}^{r} \left[\mu_j(X) \sum_{k=1}^{2} p_{jk} x_k\right] + b \\ &= \sum_{j=1}^{r} \left[(X \cdot \vec{p}_j) \prod_{k=1}^{2} M_{jk}(x_k)\right] + b \\ &= \sum_{j=1}^{r} \left[(X \cdot \vec{p}_j) \prod_{k=1}^{2} \exp\left\{-\frac{(x_k - c_{jk})^2}{\sigma_{jk}^2}\right\}\right] + b \\ &= \sum_{j=1}^{r} \left[(X \cdot \vec{p}_j) \exp\left\{-\sum_{k=1}^{2} \frac{(x_k - c_{jk})^2}{\sigma_{jk}^2}\right\}\right] + b \end{aligned} \tag{14}$$

The consequent (14) has been expressed in terms of SVR. Apropos of the expression (10), the nonlinear support vector expansion is given as —

$$\begin{aligned} f(X) &= \sum_{i=1}^{N} (\alpha_i - \hat{\alpha}_i) \langle \phi(X_i), \phi(X) \rangle + b \\ &= \sum_{i=1}^{N} (\alpha_i - \hat{\alpha}_i) \sum_{j=1}^{r} \phi_j(X_i) \phi_j(X) + b \\ &= \sum_{j=1}^{r} \left(\sum_{i=1}^{N} (\alpha_i - \hat{\alpha}_i)\, \phi_j(X_i)\right) \phi_j(X) + b \\ &= \sum_{j=1}^{r} \kappa_j \phi_j(X) + b \end{aligned} \tag{15}$$

$$\text{where, } \kappa_j = \sum_{i=1}^{N} (\alpha_i - \hat{\alpha}_i)\, \phi_j(X_i) \tag{16}$$

and $\phi(\cdot)$ is the nonlinear kernel defined as the product of a linear and a Gaussian kernel. For an input variable x, the expression for the kernel with c as the center and σ as the spread is given as —

$$\phi(x) = (x \cdot c) \exp\left\{\frac{(x - c)^2}{\sigma^2}\right\} \tag{17}$$

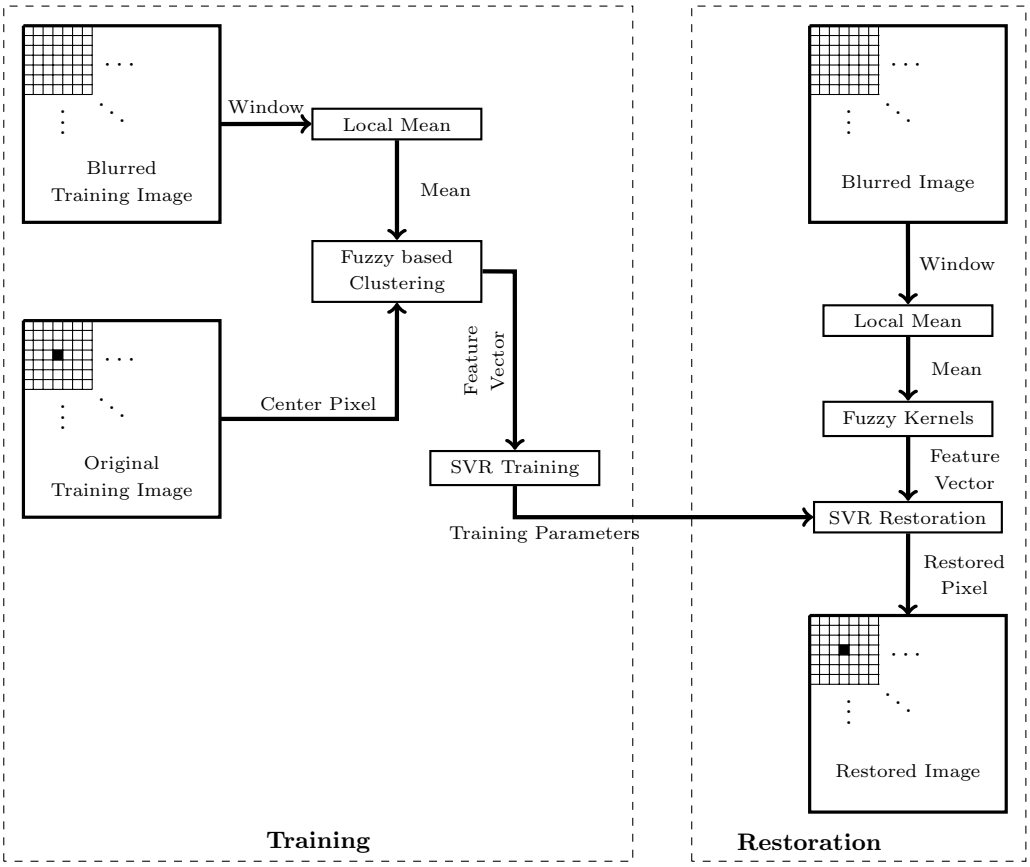


Figure 3: Block diagram of F-SVR restoration.

Under each of the r rules, there are k number of kernels having different center and spread. The $\vec{c}_j$ and $\vec{\sigma}_j$ denote the center and spread vector for kernels in j^{th} rule. The above equation can now be expressed for j^{th} rule as —

$$\phi_j(X) = (X \cdot \vec{c}_j) \exp\left\{ -\sum_{k=1}^{2} \frac{(x_k - c_{jk})^2}{\sigma_{jk}^2} \right\} \tag{18}$$

By substituting (18) in (15) we have —

$$f(X) = \sum_{j=1}^{r} \left[\kappa_j \, (X \cdot \vec{c}_j) \exp\left\{ -\sum_{k=1}^{2} \frac{(x_k - c_{jk})^2}{\sigma_{jk}^2} \right\} \right] + b \tag{19}$$

By assigning the polynomial coefficients $\vec{p}_j = \kappa_j \vec{c}_j$, the consequent $f'(X)$ in equation (14) is, thus, expressed in terms of support vector expansion $f(X)$ as represented in (19).

4.2.1 F-SVR Clustering

One of the two input variables of X to F-SVR Clustering is the mean of the $(2m+1) \times (2m+1)$ neighborhood of the blurred image pixel $g(x, y)$. The second input is the corresponding gray level of the original image $f(x, y)$. *Lena* image (down sampled to 100×100) degraded with out-of-focus blur with radius $R = 3$ is clustered for training. Here $m = 3$ is considered and thereby created windows of size 7×7 each. In F-SVR, the size of the problem mostly depends on the number of rules r. By clustering the input vectors, the number of rules is reduced. The first input vector is initialized with center of the cluster along with a pre-specified spread. To determine the belongingness of subsequent input vectors X_i, the firing strength $\mu(X_i)$ of each existing cluster is computed. The input vector belongs to that

cluster for which it has generated maximum strength. If the strength $\mu(X_i)$ falls below a threshold level, then a new cluster is created with X_i as the new cluster center. This process is repeated for the entire set of pixels by sliding the neighborhood window from top-left to bottom-right. Once the number of clusters is determined, the training pairs for SVR are generated as —

$$\left(\vec{\phi}(X_1); y_1\right), \left(\vec{\phi}(X_2); y_2\right), \ldots, \left(\vec{\phi}(X_N); y_N\right) \tag{20}$$

where,

$$\vec{\phi}(X_i) = [\phi_1(X_i), \phi_2(X_i), \ldots, \phi_r(X_i)] \tag{21}$$

4.2.2 F-SVR Training

Training samples for *Lena* image are generated in congruence with (20). In this training, the ε-insensitive as the loss function and the radial basis function as the nonlinear kernel are used. Here again input data is trained using LibSVM tool to generate support vectors. They are used for subsequent testing.

4.2.3 F-SVR Restoration

Restoration of a degraded image is performed on a pixel-by-pixel basis. The 7×7 neighborhood from the given blurred image is used to construct the input vector. The same kernel that has been used in training is used here to raise the dimension of the input vector. The trained SVR is then used to approximate each pixel value. Performance analysis of the scheme is presented in Section 5. The block diagram in Figure 3 depicts the sequence of operations followed in F-SVR.

5 Simulation Results

Both V-SVR and F-SVR schemes are simulated to compare their efficiency with respect to training and subsequent restoration. Both the schemes are trained with *Lena* image, which is blurred with out-of-focus blur of radius $R = 3$. The training samples are derived from a 7×7 window for every pixel of the blurred image. The Blind Image Deconvolution through SVR (BIDSVR) scheme reported in [9] is also simulated under similar conditions. The training performance parameters like number of iterations, number of support vectors (SVs), and number of bounded support vectors are listed in Table 1. It is observed that both V-SVR and F-SVR show tremendous improvement in terms of SVs and bounded SVs, with almost one-fifth numbers of iterations as compared to that of BIDSVR.

Once the training is completed, V-SVR, F-SVR, and BIDSVR are used to restore the images blurred with different PSFs. For this purpose following three experiments are conducted and their subjective and objective performances are analysed.

Experiment-I Two synthetic binary images, namely *NITRKL* and *Hello* as well as a grayscale image *Cameraman* are blurred with out-of-focus blur of radius 5, 7, and 5 respectively. These images are then subjected to deblurring with all the three schemes and the restored results are shown in Figures 4, 5, and 6 respectively. The results obtained from V-SVR and F-SVR are better than that obtained from BIDSVR in all the three cases. The peak signal to noise ratio (PSNR) (22) shown in Table 2 also reflects a similar trend.

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\frac{1}{MN} \sum_{x,y=1}^{M,N} \left(f(x,y) - \hat{f}(x,y) \right)^2} \right) dB \tag{22}$$

where MN is the size of the image, and $f(x,y)$ and $\hat{f}(x,y)$ represent the pixel values at $(x,y)^{th}$ location of original and restored image respectively.

Experiment-II The same three images *NITRKL*, *Hello*, and *Cameraman* are subjected to a horizontal blur of length 7, 7, and 5 respectively. These images are deblurred through all the three schemes, and restored results are shown in Figures 7, 8, and 9 in sequel. Here also a comparable improved performance of V-SVR and F-SVR over BIDSVR is observed, both visually as well as in terms of PSNR, is observed. However, the results are slightly inferior as compared to the results of out-of-focus blur. This is due to the fact that training is performed in out-of-focus blur.

Experiment-III In this experiment, the restoration schemes are tested with photographic blurred images. A blurred image is captured by deliberately defocussing the lens system of the camera. This image (*Fevi Stik*) is then restored with all the three schemes and their performance are shown in Figure 10, which demonstrates the superiority of the proposed schemes.

From the three experiments conducted and observations made out of them reveal the followings —

(i) The proposed schemes V-SVR and F-SVR outperforms BIDSVR scheme with respect to objective as well as subjective performance measures.

(ii) The training performed with out-focus-blur blur of radius $R = 3$ on *Lena* image, is used to deblur different images blurred with radius 5 and 7. In addition, the schemes are also used to deblur motion blurred images with different length and observed marked improved performances. This signifies the generalized behavior of the proposed schemes.

(iii) Since the proposed schemes show a remarkable performance for binary images, they are best suited for document imaging applications.

(iv) The proposed methods work well for photographic images too and can be applied to any real time applications.

6 Conclusion

Two algorithms (V-SVR and F-SVR) based on SVR for restoring degraded images are proposed in this paper. Extensive simulations on synthetic binary images, grayscale images and photographic images have been carried out and both subjective and objective performance measures are analyzed. The quick learning capabilities of the proposed schemes are also demonstrated. In general, it is observed that the proposed schemes have bettered the performance of other reported schemes.

Table 1: Training comparison of BIDSVR, V-SVR, and F-SVR for *Lena* image degraded with out-of-focus blur ($R = 3$).

Attribute	BIDSVR	V-SVR	F-SVR
Number of iterations	36351	7987	6868
Number of SVs	54561	9753	8239
Number of bounded SVs	54321	9649	8142

References

[1] Anil K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[2] Jae S. Lim. *Two-Dimensional Signal and Image Processing.* Prentice Hall, 1990.

Table 2: PSNR (dB) comparison of BIDSVR, V-SVR, and F-SVR for different images with varying PSF support.

Image	Blur	PSF	PSNR (dB) of		
			BIDSVR	V-SVR	F-SVR
NITRKL	Out-of-Focus	$R = 5$	26.053	26.503	26.554
Hello		$R = 7$	20.036	22.613	24.394
Cameraman		$R = 5$	20.593	21.069	23.270
NITRKL	Motion	$L = 7$	22.050	22.351	23.743
Hello		$L = 7$	19.236	22.105	22.605
Cameraman		$L = 5$	19.944	22.208	22.400

[3] Rafael Gonzalez and Richard Woods. *Digital Image Processing*. Addison Wesley, 1992.

[4] G. R. Ayers and J. C. Dainty. Iterative blind deconvolution methods and its applications. *Optics Letter*, 13(7):547 – 549, July 1988.

[5] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):754 – 754, 1974.

[6] William Hadley Richardson. Bayesian-based iterative method of image restoration. *Journal of Optical Society of America*, 62(1):55 – 59, January 1972.

[7] Deepa Kundur and Dimitrios Hatzinakos. A Novel Blind Deconvolution Scheme for Image Restoration using Recursive Filtering. *IEEE Transaction on Signal Processing*, 46(2):375 – 390, February 1998.

[8] Dalong Li, Russell M. Mersereau, and Steven Simske. Atmospheric Turbulence-Degraded Image Restoration Using Principal Components Analysis. *IEEE Geoscience and Remote Sensing Letter*, 4(3):340 – 344, July 2007.

[9] Dalong Li, Russell M. Mersereau, and Steven Simske. Blind Image Deconvolution Through Support Vector Regression. *IEEE Transactions on Neural Networks*, 18(3):931 – 935, May 2007.

[10] Khalid Sayood. *Introduction to Data Compression*. Academic Press, 1996.

[11] David Salomon. *Data Compression: The Complete Reference*. Springer-Verlag, London, 1998.

[12] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[13] Nello Cristianini and John Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[14] Alex J. Smola and Bernhard Scholkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199 – 222, August 2004. Kluwer Academic Publisher.

[15] Chun-Fu Lin and Sheng-De Wang. Fuzzy Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):464 – 471, March 2002.

[16] Jung-Hsien Chiang and Pei-Yi Hao. Support Vector Learning Mechanism for Fuzzy Rule-Based Modeling: A New Approach. *IEEE Transactions on Fuzzy Systems*, 12(1):1 – 12, February 2004.

[17] Alistair Shilton and Daniel T. H. Lai. Iterative Fuzzy Support Vector Machine Classification. In *IEEE International Conference on Fuzzy Systems*, FUZZ-IEEE - 2007, pages 1 – 6, London, UK, July 2007. IEEE.

[18] Kui Wu and Kim-Hui Yap. Fuzzy SVM for Content-Based Image Retrival. *IEEE Computational Intelligence Magazine*, 1(2):10 – 16, May 2006.

[19] Jacek M. LeRski. On support vector regression machines with linguistic interpretation of the kernel matrix. *Fuzzy Sets and Systems*, 157(8):1098 – 1113, April 2006.

[20] Chia-Feng Juang and Chin-Teng Lin. An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications. *IEEE Transactions on Fuzzy Systems*, 6(1):12 – 32, February 1998.

[21] Chia-Feng Juang and Cheng-Da Hsieh. TS-fuzzy system-based support vector regression. *Fuzzy Sets and Systems*, 160(17):2486 – 2504, September 2009.

[22] Dug Hun Hong and Changha Hwang. Support vector fuzzy regression machines. *Fuzzy Sets and Systems*, 138(2):271 – 281, September 2003.

[23] Pei-Yi Hao. Fuzzy one-class support vector machines. *Fuzzy Sets and Systems*, 159(18):2317 – 2336, September 2008.

(a) True Image (b) Blurred (c) BIDSVR (d) V-SVR (e) F-SVR

Figure 4: Restoration results of *NITRKL* image degraded with out-of-focus blur ($R = 5$) by various schemes.

(a) True Image (b) Blurred (c) BIDSVR (d) V-SVR (e) F-SVR

Figure 5: Restoration results of *Hello* image degraded with out-of-focus blur ($R = 7$) by various schemes.

(a) True Image (b) Blurred (c) BIDSVR (d) V-SVR (e) F-SVR

Figure 6: Restoration results of *Cameraman* image degraded with out-of-focus blur ($R = 5$) by various schemes.

(a) True Image (b) Blurred (c) BIDSVR (d) V-SVR (e) F-SVR

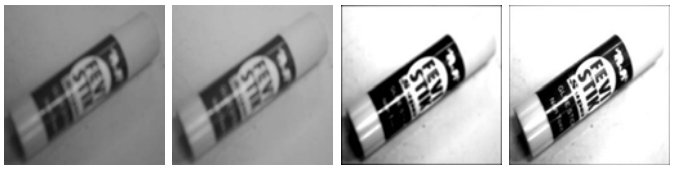
Figure 7: Restoration results of *NITRKL* image degraded with horizontal motion blur ($L = 7$) by various schemes.

(a) True Image (b) Blurred (c) BIDSVR (d) V-SVR (e) F-SVR

Figure 8: Restoration results of *Hello* image degraded with horizontal motion blur ($L = 7$) by various schemes.

(a) True Image (b) Blurred (c) BIDSVR (d) V-SVR (e) F-SVR

Figure 9: Restoration results of *Cameraman* image degraded with horizontal motion blur ($L = 5$) by various schemes.



(a) Blurred (b) BIDSVR (c) V-SVR (d) F-SVR

Figure 10: Restoration results of a photographic out-of-focus blurred image *Fevi Stik* by various schemes.