EURASIP Journal on Applied Signal Processing 2004:16, 2441–2450
© 2004 Hindawi Publishing Corporation

# Nonlinear Image Restoration Using a Radial Basis Function Network

**Keiji Icho**
*The Research & Development Department, Matsushita Electric Industrial Co., Ltd., Osaka 571-8501, Japan*
*Email: icho.k@jp.panasonic.com*

**Youji Iiguni**
*Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, Osaka 560-8531, Japan*
*Email: iiguni@sys.es.osaka-u.ac.jp*

**Hajime Maeda**
*Department of Communications Engineering, Graduale School of Engineering, Osaka University, Osaka 565-0871, Japan*
*Email: maeda@comm.eng.osaka-u.ac.jp*

*Received 4 August 2003; Revised 22 April 2004*

We propose a nonlinear image restoration method that uses the generalized radial basis function network (GRBFN) and a regularization method. The GRBFN is used to estimate the nonlinear blurring function. The regularization method is used to recover the original image from the nonlinearly degraded image. We alternately use the two estimation methods to restore the original image from the degraded image. Since the GRBFN approximates the nonlinear blurring function itself, the existence of the inverse of the blurring process does not need to be assured. A method of adjusting the regularization parameter according to image characteristics is also presented for improving restoration performance.

**Keywords and phrases:** radial basis function network, regularization, nonlinear image restoration, steepest descent technique, alternate estimation.

## 1. INTRODUCTION

In the recent years, a special class of artificial neural networks, called the radial basis function network (RBFN), has received considerable attention [1]. The RBFN has a universal approximation capability [2] and has successfully been applied to many signal and image processing problems due to its excellent approximation capability [3, 4]. The RBFN provides a smooth function that achieves a good tradeoff between fidelity to the data and smoothness. The regularization parameter controls the tradeoff between the two requirements. In the standard RBFN, the number of basis functions is set to equal to the number of data. Therefore, the computational requirement of constructing the RBFN gets larger as the number of data gets larger. For reducing the computational complexity, the generalized RBFN (GRBFN), which has a smaller number of basis functions than the number of data, has been proposed [1, 5].

Restoration of images degraded by blur and additive noise is very important in image processing. A number of image restoration methods have been proposed so far [6, 7, 8, 9]. In [8], a regularization approach has been developed, which provides a compromise between fidelity of the restored image to the degraded image and smoothness of the original image and the blurring function. Alternate estimation of original image and blurring function is employed for improving restoration performance, however the blurring function is assumed to be linear. In some applications, nonlinear blurring needs to be considered in the design of restoration process. The neglect results in an unacceptable restoration performance [10]. In [9], the GRBFN is used to restore the original image from the nonlinearly degraded image in the presence of observation noise, however the inverse of the blurring process is assumed to exist. Moreover, the GRBFN trained by a training image has been fixed during the processing of test images.

In this paper, a nonlinear image restoration method that uses the GRBFN and a regularization method proposed. The GRBFN is used to estimate the nonlinear blurring function under the assumption that the original image is known and smooth. The regularization method is used to recover the original image from the nonlinearly degraded image, under the assumption that the blurring function is known and smooth. A cost function is then minimized by using the steepest descent technique. In actual applications, neither the blurring function nor the original image is known. We thus alternately use the two estimation methods to restore the original image from the nonlinearly degraded image, under the single assumption of the smoothness of original image and blurring function. Since the GRBFN approximates the nonlinear blurring function itself, the existence of the inverse of the blurring process does not need to be assured. Moreover, in order to efficiently remove noise components, we propose a method of adjusting the regularization parameter according to image characteristics. It is shown through computer simulations that the adaptive selection method is useful in improving restoration performance.

## 2. FUNCTION APPROXIMATION BY USING THE GRBFN

### 2.1. RBFN

Suppose that the I/O relationship between the $p$-dimensional input vector $\mathbf{x}= [x_1, x_2, \dots, x_p]^T$ and the scalar output $y$ is described by

$$y = g(\mathbf{x}) + \varepsilon, \tag{1}$$

where $g$ is an unknown function, and $\varepsilon$ is a random noise with zero mean. Given a set of $N$ observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the regularization procedure determines a smooth function $f$ that minimizes the following cost function:

$$H[g] = \sum_{i=1}^{N} \left(y_i - g(\mathbf{x}_i)\right)^2 + \lambda S[g]. \tag{2}$$

The first term measures the distance between the data and the desired solution. The second term is the stabilizer that measures the cost associated with the deviation from the smoothness. The nonnegative parameter $\lambda$, called the regularization parameter, controls the tradeoff between the two terms. As $\lambda$ is larger, the resulting function becomes smoother, while the discrepancy between the data point and the network output becomes larger.

We choose the stabilizer as

$$S[g] = \sum_{n=0}^{\infty} (-1)^n \frac{\sigma^{2n}}{n!2^n} \nabla^{2n} g, \tag{3}$$

where $\nabla^{2n}$ is the $n$ iterated Laplacian operator in $n$ dimensions. Then, the minimization of $H[g]$ yields the following solution [1]:

$$g(\mathbf{x}) = \sum_{i=1}^{N} c_i h\left(|\mathbf{x} - \mathbf{x}_i|\right), \tag{4}$$

where $c_i$ and $\mathbf{x}_i$ are the network parameter and the RBF center, respectively, and $h$ is the Gaussian function defined by

$$h\left(|\mathbf{x} - \mathbf{x}_i|\right) = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_i|^2}{2\sigma^2}\right). \tag{5}$$

The parameter $\sigma$ is the width of the Gaussian function. The function $g$ is called the RBFN as it is expressed as a linear combination of RBFs, $h(|\mathbf{x} - \mathbf{x}_i|)$ $(i = 1, 2, \dots, N)$. The network parameter is determined by solving the following linear equations of size $(N \times N)$:

$$(\mathbf{H} + \lambda\mathbf{I})\mathbf{c} = \mathbf{y}. \tag{6}$$

Here, $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ is the output vector, $\mathbf{c} = [c_1, c_2, \dots, c_N]^T$ is the parameter vector, and $\mathbf{H}$ is the $(N \times N)$ matrix whose $ij$th element is given by $(\mathbf{H})_{ij} = h(|\mathbf{x}_i - \mathbf{x}_j|)$. The matrix $(\mathbf{H} + \lambda\mathbf{I})$ is always invertible for nonnegative $\lambda$ [11]. The RBFN has a universal approximation capability [2] and has successfully been used for approximating a large variety of nonlinear functions in signal and image processing problems [3, 4].

### 2.2. GRBFN

The GRBFN, which has a smaller number of basis functions than the number of data, is represented by [1, 5]

$$g(\mathbf{x}) = \sum_{j=1}^{M} c_j \exp\left(-\frac{|\mathbf{x} - \mathbf{t}_j|^2}{2\sigma^2}\right), \quad (M < N), \tag{7}$$

where $M$ is the number of basis functions and the $p$-dimensional vector $\mathbf{t}_j$ $(j = 1, 2, \dots, M)$ is called the center of the GRBF. The network parameter $c_j$ $(j = 1, 2, \dots, M)$ is determined by solving the following linear equations of size $(M \times M)$,

$$(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{G}_0)\mathbf{c} = \mathbf{G}^T\mathbf{y}, \tag{8}$$

where we define the $(N \times M)$ matrix $\mathbf{G}$ and the $(M \times M)$ matrix $\mathbf{G}_0$ as

$$\begin{aligned} (\mathbf{G})_{ij} &= \exp\left(-\frac{|\mathbf{x}_i - \mathbf{t}_j|^2}{2\sigma^2}\right) \\ &\qquad (i = 1, 2, \dots, N,\ j = 1, 2, \dots, M), \\ (\mathbf{G}_0)_{pq} &= \exp\left(-\frac{|\mathbf{t}_p - \mathbf{t}_q|^2}{2\sigma^2}\right) \\ &\qquad (p, q = 1, 2, \dots, M), \end{aligned} \tag{9}$$

respectively, and put $\mathbf{c} = (c_1, c_2, \dots, c_M)^T$ and $\mathbf{y} = (y_1, y_2, \dots, y_N)^T$. We put $M = N$ to have $\mathbf{G} = \mathbf{G}_0 = \mathbf{H}$. The GRBFN then results in the standard RBFN.

The computational complexity of solving (8) is of order $M^3$, while that of solving (6) is of order $N^3$. Therefore, the GRBFN requires less computation than the standard RBFN, although the approximation accuracy of the GRBFN degrades as the number of basis functions $M$ decreases.

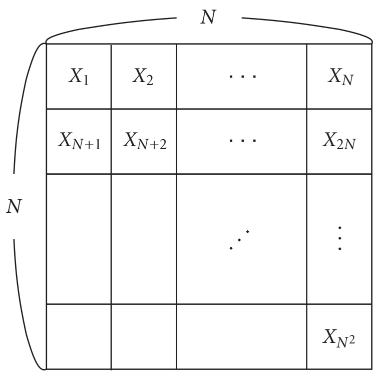

Figure 1: Pixel position $\mathbf{x}_i$.

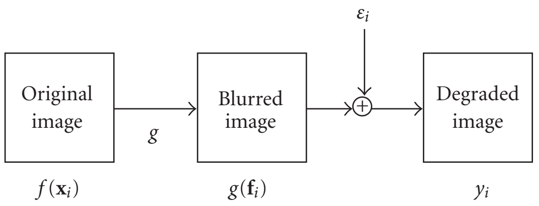

Figure 2: Degradation process.

| | | |
|---|---|---|
| $\mathbf{x}_{i-N-1}$ | $\mathbf{x}_{i-N}$ | $\mathbf{x}_{i-N+1}$ |
| $\mathbf{x}_{i-1}$ | $\mathbf{x}_{i}$ | $\mathbf{x}_{i+1}$ |
| $\mathbf{x}_{i+N-1}$ | $\mathbf{x}_{i+N}$ | $\mathbf{x}_{i+N+1}$ |

Figure 3: Pixel position considered in $\mathbf{f}_i$.

## 3. IMAGE RESTORATION BASED ON THE REGULARIZATION METHOD

We express the original image of size $(N \times N)$ as $f(\mathbf{x}_i)$ $(i = 1, 2, \ldots, N^2)$, where we choose the pixel position $\mathbf{x}_i$ as shown in Figure 1. Figure 2 illustrates the degradation process considered here. The degraded image $y_i$ at the position $\mathbf{x}_i$ is represented by

$$y_i = g(\mathbf{f}_i) + \varepsilon_i \quad (i = 1, 2, \ldots, N^2), \tag{10}$$

where $g$ is a space-invariant nonlinear blurring function, $\varepsilon_i$ is a random observation noise with zero mean. Also, $\mathbf{f}_i$ is the 9-dimensional vector (or the $(3 \times 3)$ image) consisting of pixel values of the original image at $\mathbf{x}_i$ and its 8-nearest neighbors, represented by

$$\begin{aligned}\mathbf{f}_i = \big(&f(\mathbf{x}_{i-N-1}), f(\mathbf{x}_{i-N}), f(\mathbf{x}_{i-N+1}), f(\mathbf{x}_{i-1}),\\ &f(\mathbf{x}_i), f(\mathbf{x}_{i+1}), f(\mathbf{x}_{i+N-1}), f(\mathbf{x}_{i+N}), f(\mathbf{x}_{i+N+1})\big)^T.\end{aligned} \tag{11}$$

Figure 3 illustrates the pixel position of the elements of $\mathbf{f}_i$. Although we here define $\mathbf{f}_i$ as in (11), other choices are possible.

We estimate the blurring function $g$ from the degraded image $y_i$ by using the GRBFN, under the assumption that the original image $f(\mathbf{x}_i)$ is known and $g$ is smooth. Next, we estimate $f(\mathbf{x}_i)$ from $y_i$ by using a regularization method, under the assumption that $g$ is known and $f(\mathbf{x}_i)$ is smooth. Finally, we alternately use the two methods to estimate $f(\mathbf{x}_i)$ from $y_i$, under the single assumption of the smoothness of $g$ and $f(\mathbf{x}_i)$.

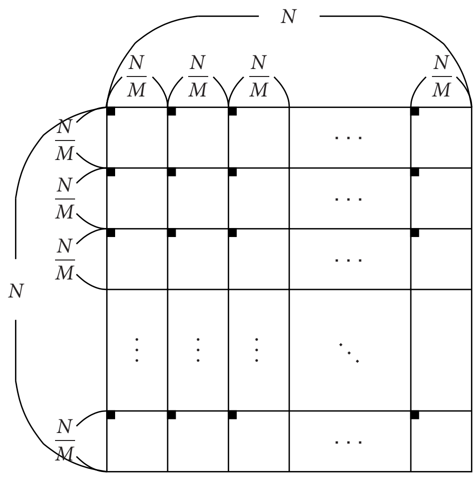

Figure 4: Pixel positions of the centers.

### 3.1. Estimation of blurring function

We approximate the smooth blurring function $g(\mathbf{f}_i)$ by using the GRBFN with $M^2$ centers. The output of the GRBFN is represented by

$$\hat{g}(\mathbf{f}_i) = \sum_{j=1}^{M^2} c_j \exp\left(-\frac{|\mathbf{f}_i - \mathbf{t}_j|^2}{2\sigma^2}\right), \tag{12}$$

where $\hat{g}$ denotes the estimate of $g$. We here put the 9-dimensional center $\mathbf{t}_j$ as

$$\mathbf{t}_{k+M(l-1)} = \mathbf{f}_{(N^2/M)(l-1)+(N/M)(k-1)+N+2} \quad (k, l = 1, 2, \ldots, M), \tag{13}$$

so that they are equally distributed over the whole image space. The black squares in Figure 4 denote the positions of the center. For example, when $N = 256$ and $M = 4$, the GRBFN has 16 centers such that $\mathbf{t}_1 = \mathbf{f}_{258}$, $\mathbf{t}_2 = \mathbf{f}_{322}, \ldots, \mathbf{t}_{16} = \mathbf{f}_{49602}$.

The determination of the number and position of centers is very important in the application of the GRBFN. The orthogonal least square training method for determining the centers has been proposed in [12], and the limitation is suggested in [13]. Dynamic configuration methods that adaptively add a new basis function to the network according to a prescribed performance index have been proposed in [14, 15]. However, we simply choose the centers as in (13), because the iterative determination of the number and position of centers is computationally expensive, and (13) gives a satisfactory estimation accuracy as shown below if the user appropriately set the values of $\lambda$ and $\sigma^2$.

TABLE 1: MSE for different values of $\lambda$ and $\sigma^2$, ($E[\varepsilon_i^2] = 200$).

| $\sigma^2$ | $\lambda$ | | | | |
|---|---|---|---|---|---|
| | 0.0001 | 0.001 | 0.01 | 0.1 | 1.0 |
| 100 | 180.96 | 180.96 | 180.96 | 180.96 | 180.96 |
| 200 | 14.10 | 14.10 | 14.10 | 14.10 | 14.11 |
| 500 | 2.45 | 2.45 | 2.45 | 2.47 | 2.74 |
| 1000 | 2.31 | 2.30 | 2.33 | 2.61 | 3.08 |
| 2000 | 2.57 | 2.58 | 2.76 | 2.81 | 2.88 |
| 10000 | 2.82 | 2.82 | 2.80 | 3.02 | 16.97 |

TABLE 2: MSE for different values of $\lambda$ and $\sigma^2$, ($E[\varepsilon_i^2] = 400$).

| $\sigma^2$ | $\lambda$ | | | | |
|---|---|---|---|---|---|
| | 0.0001 | 0.001 | 0.01 | 0.1 | 1.0 |
| 100 | 180.99 | 180.99 | 180.99 | 180.99 | 180.98 |
| 200 | 14.15 | 14.15 | 14.15 | 14.15 | 14.16 |
| 500 | 2.51 | 2.51 | 2.51 | 2.52 | 2.76 |
| 1000 | 2.36 | 2.36 | 2.38 | 2.64 | 3.10 |
| 2000 | 2.62 | 2.63 | 2.81 | 2.84 | 2.89 |
| 10000 | 2.84 | 2.84 | 2.82 | 3.03 | 16.92 |

The regularization parameter $c_i$ ($i = 1, 2, \ldots, M^2$) is determined by solving the following linear equations of size ($M^2 \times M^2$):

$$(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{G}_0)\mathbf{c} = \mathbf{G}^T\mathbf{y}, \tag{14}$$

where we define the ($N^2 \times M^2$) matrix $\mathbf{G}$ and the ($M^2 \times M^2$) matrix $\mathbf{G}_0$ as

$$\begin{aligned}
(\mathbf{G})_{ij} &= \exp\left(-\frac{|\mathbf{f}_i - \mathbf{t}_j|^2}{2\sigma^2}\right), \quad (i = 1, 2, \ldots, N^2,\ j = 1, 2, \ldots, M^2),\\
(\mathbf{G}_0)_{pq} &= \exp\left(-\frac{|\mathbf{t}_p - \mathbf{t}_q|^2}{2\sigma^2}\right), \quad (p, q = 1, 2, \ldots, M^2),
\end{aligned} \tag{15}$$

respectively, and put $\mathbf{c} = (c_1, c_2, \ldots, c_{M^2})^T$ and $\mathbf{y} = (y_1, y_2, \ldots, y_{N^2})^T$. Although we may no longer find physical properties of the blurring process from (12), the GRBFN can approximate an arbitrary nonlinear function with high accuracy [11].

## 3.2. Example of blur estimation

We here consider the following nonlinear blurring function:

$$\begin{aligned}
g(\mathbf{f}_i) = \frac{1}{3}\big(&f^2(\mathbf{x}_{i-N-1}) + f^2(\mathbf{x}_{i-N}) + f^2(\mathbf{x}_{i-N+1})\\
&+ f^2(\mathbf{x}_{i-1}) + f^2(\mathbf{x}_i) + f^2(\mathbf{x}_{i+1})\\
&+ f^2(\mathbf{x}_{i+N-1}) + f^2(\mathbf{x}_{i+N}) + f^2(\mathbf{x}_{i+N+1})\big)^{1/2}.
\end{aligned} \tag{16}$$

We degraded the Girl image of size (256 × 256) with 8 bit grayscale by using (16). Using the degraded image, we measured the computation time and the mean square error (MSE) between the true and approximated images, computed by

$$\text{MSE} = \frac{1}{N^2}\sum_{i=1}^{N^2}\left(\hat{g}(\mathbf{f}_i) - g(\mathbf{f}_i)\right)^2. \tag{17}$$

As the number of basis functions $M$ increases, the MSE becomes smaller, while the computation time becomes larger. Therefore, we have to make a tradeoff between the computation time and the MSE in determining the parameter $M$.

We measured the computation time for different values of $M$, where we used an IBM PC/AT compatible computer with an Intel Pentium II 750 MHz and 256 MB DRAM. The computation times for $M = 2, 4, 8, 16$ are 0.95, 6.15, 78.4, and 1226.6 (s), respectively. We next measured the MSE for different values of $\lambda$ and $\sigma^2$, where we put $M = 4$. Tables 1 and 2 summarize the results for $E[\varepsilon_i^2] = 200$ and $E[\varepsilon_i^2] = 400$, respectively. We see that choosing $\lambda = 0.001$ and $\sigma^2 = 1000$ provides a good tradeoff between the computation time and the MSE, and that the MSE is fairly robust against the choice. From these results, we put $M = 4$, $\lambda = 0.001$, and $\sigma^2 = 1000$ throughout the simulations.

## 3.3. Estimation of original image

We assume that the blurring function $g$ is known and use a regularization method to estimate $f(\mathbf{x}_i)$ from $y_i$. We choose a cost function to be minimized as

$$\begin{aligned}
J[\mathbf{F}] = &\sum_{i=1}^{N^2}\left(y_i - g(\mathbf{f}_i)\right)^2\\
&+ \gamma\sum_{i=1}^{N^2}\Big(f(\mathbf{x}_{i-N}) + f(\mathbf{x}_{i-1}) - 4f(\mathbf{x}_i)\\
&\qquad + f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i+N})\Big)^2,
\end{aligned} \tag{18}$$

where $\mathbf{F} \overset{\text{def}}{=} [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_{N^2})]^T$ denotes the vector representation of the original image. The first term measures the distance between the degraded and blurred images, and the second term is the Laplacian operator that measures the smoothness of the original image. The regularization parameter $\gamma$ controls the tradeoff between the two terms. We here use the following steepest descent method for the minimization:

$$\widehat{f^{k+1}}(\mathbf{x}_i) = \widehat{f^k}(\mathbf{x}_i) - \frac{\mu}{N^2}\frac{\partial J}{\partial f(\mathbf{x}_i)}\bigg|_{\mathbf{F}=\widehat{\mathbf{F}^k}} \quad (i = 1, 2, \ldots, N^2), \tag{19}$$

where $\mu$ is a small positive constant called the step size, and $\widehat{\mathbf{F}^k} = [\widehat{f^k}(\mathbf{x}_1), \widehat{f^k}(\mathbf{x}_2), \ldots, \widehat{f^k}(\mathbf{x}_{N^2})]^T$ denotes the vector representation of the restored image at the $k$th iteration. The detailed computation of $\partial J/\partial f(\mathbf{x}_i)$ is shown in Appendix A.

### 3.4. Alternate estimation of blurring function and original image

In the blur estimation described in Section 3.1, the original image $f(\mathbf{x}_i)$ is assumed to be known. In the original image estimation described in Section 3.3, the blur function $g(\mathbf{f}_i)$ is assumed to be known. However, in actual applications, neither the original image nor the blurring function is known. Therefore, we alternately use the blur and image estimation methods.

We prepare a clean original image, and generate a training image by blurring the original image and adding observation noise to the blurred image. We apply the blur estimation method to the training image under the assumption that the original image is known. We denote the initial estimate of the blurring function by $\hat{g}$. The training image is used only for obtaining $\hat{g}$. We call the degraded image to be restored the test image. Unlike the training image, we assume that the original image of the test image is unknown. We replace the blurring function $g$ in (18) by the initial estimate $\hat{g}$ to have

$$J[\mathbf{F}] = \sum_{i=1}^{N^2} \left(y_i - \hat{g}(\mathbf{f}_i)\right)^2 + \gamma \sum_{i=1}^{N^2} \Big(f(\mathbf{x}_{i-N}) + f(\mathbf{x}_{i-1}) - 4f(\mathbf{x}_i) + f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i+N})\Big)^2. \tag{20}$$

We minimize the cost function (20) with respect to $\mathbf{F}$ by using the steepest descent method (19), and obtain the restored image $\widehat{\mathbf{F}^{\mathbf{k}}}$. In general, the initial estimate $\hat{g}$ is not accurate. We thus replace $f(\mathbf{x}_i)$ by $\widehat{f^k}(\mathbf{x}_i)$ in (14) and (15), and again estimate the blurring function by solving the linear equations (14). We alternately repeat the estimations of the blurring function and the original image until the iteration (19) converges. The blur function is uniquely estimated by solving (14), while the original image is iteratively estimated by using the steepest descent method (19) so that the cost function $J[\mathbf{F}]$ is minimized. Since $J[\mathbf{F}]$ is nonlinear with respect to $\mathbf{F}$, the steepest descent method may converge to local minima of $J$.

## 4. SIMULATION RESULTS

Using Girl, Lena, and Baboon images of size $256 \times 256$ with 8-bit gray levels, we evaluated the restoration performance of the proposed method. We generated the training image by using the following blurring function:

$$g(\mathbf{f}_i) = \frac{1}{3}\Big(f^2(\mathbf{x}_{i-N-1}) + 1.1f^2(\mathbf{x}_{i-N}) + 0.9f^2(\mathbf{x}_{i-N+1}) + 0.8f^2(\mathbf{x}_{i-1}) + f^2(\mathbf{x}_i) + 1.2f^2(\mathbf{x}_{i+1}) + 0.9f^2(\mathbf{x}_{i+N-1}) + 1.1f^2(\mathbf{x}_{i+N}) + f^2(\mathbf{x}_{i+N+1})\Big)^{1/2}, \tag{21}$$

while we generated the test images by using (16). It is here noted that the blurring functions are different in the training and test images. Throughout the simulations, we put $E[\varepsilon_i^2] = 200$.

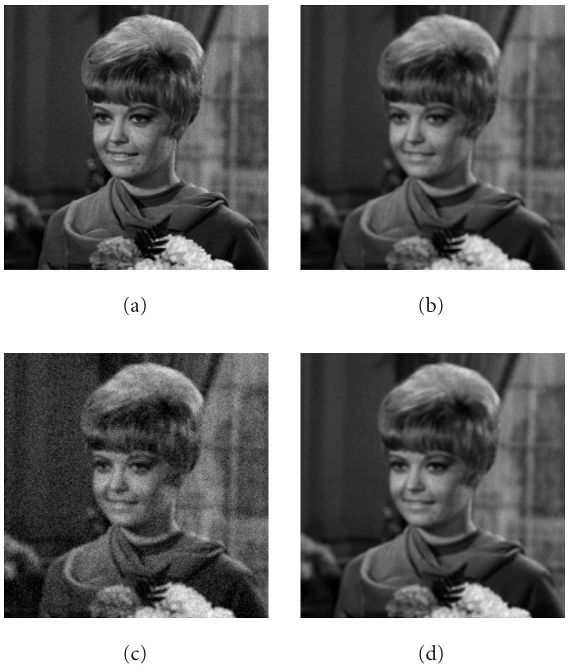
(a) (b) (c) (d)

Figure 5: Blur estimation result: (a) original image $f(\mathbf{x}_i)$, (b) blurred image $g(\mathbf{f}_i)$, (c) degraded image $y_i$, and (d) output of the GRBFN $\hat{g}(\mathbf{f}_i)$.

We used the degraded Girl image as the training image, the degraded Lena and Baboon images as the test images, and compared the restoration performances of the following two methods.

*Method A*

This method comprises two steps.

Step (1) Using the training image, we initially estimate the blurring function by the GRBFN.
Step (2) Using the test image, we repeat the update procedure (19) to estimate the original image until no significant improvement is obtained.

*Method B*

This method comprises four steps.

Step (1) Using the training image, we initially estimate the blurring function by the GRBFN.
Step (2) Using the test image, we repeat the update procedure (19) 10 times to obtain a restored image at an intermediate stage.
Step (3) We regard the restored image as the original image, and again estimate the blurring function.
Step (4) Repeat Step (2) and Step (3) until no significant improvement is obtained in Step (2).

In Method A, the initial estimate of the blurring function has been used without any change during the processing of

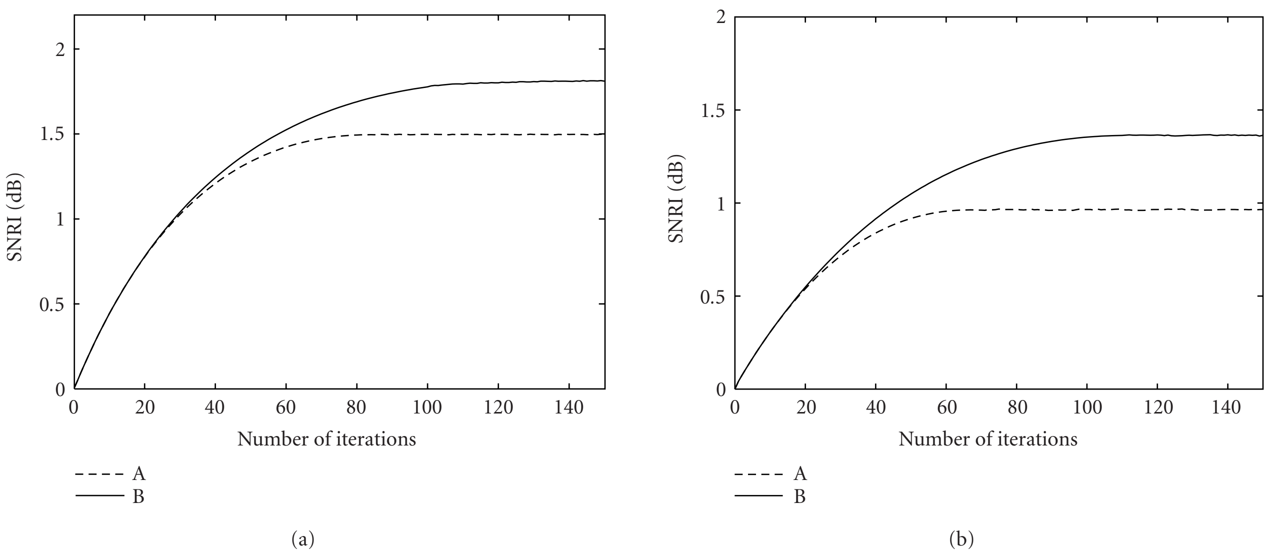

FIGURE 6: Convergence curves of SNRIs made by Methods A and B: (a) Lena and (b) Baboon.

test images, while in Method B, both of the blurring function and the original image are iteratively estimated. We can verify the usefulness of the alternate estimation by comparing the restoration performance of Methods A and B.

Figure 5 shows the blur estimation result obtained by using the training image. Figures 5a, 5b, and 5c show the original, blurred, and degraded Girl images, respectively, and Figure 5d shows the output of the GRBFN. We see that $g(\mathbf{f}_i)$ and $\hat{g}(\mathbf{f}_i)$ are very close to each other. We measured the signal-to-noise ratio improvement (SNRI) for each iteration, defined by

$$\begin{aligned} \text{SNRI} &= \left(\text{SNR of } \widehat{f^k}(\mathbf{x}_i)\right) - \left(\text{SNR of } y_i\right) \\ &= 10\log_{10}\frac{\sum_{i=1}^{N^2} f^2(\mathbf{x}_i)}{\sum_{i=1}^{N^2}\left(\widehat{f^k}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2} \\ &\quad - 10\log_{10}\frac{\sum_{i=1}^{N^2} f^2(\mathbf{x}_i)}{\sum_{i=1}^{N^2}\left(y_i - f(\mathbf{x}_i)\right)^2} \\ &= 10\log_{10}\frac{\sum_{i=1}^{N^2}\left(y_i - f(\mathbf{x}_i)\right)^2}{\sum_{i=1}^{N^2}\left(\widehat{f^k}(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2}. \end{aligned} \tag{22}$$

Figure 6 shows the convergence behaviors of SNRI for Methods A and B, where Figures 6a and 6b are the results of Lena and Baboon images, respectively. The horizontal axis denotes the total number of iterations in the steepest descent method. We put $\mu = 0.02$ so that the SNRI is higher and the recursion (19) converges in 100 iterations. This means that the number of turns from Step (4) to Step (2) is $9 = 100/10 - 1$. We see that Method B gives a better restoration performance than Method A in both test images. We also see that the SNRI of Lena image is higher than that of Baboon image, because Baboon image contains a large amount of high-frequency components, such as hairs, and the important high-frequency components as well as noise components are removed by minimizing the cost function (20).

Figures 7a, 7b, and 7c show the original, blurred, and degraded Lena images, respectively. Figure 7d shows the finally restored image by using Method B. Results for Baboon image are also shown in Figure 8. We can confirm the effectiveness of the proposed method.

## 5. IMAGE RESTORATION WITH VARIABLE $\gamma$

### 5.1. *Determination of* $\gamma$

Natural images are composed of various components, such as the flat part and the edge. In order to efficiently remove noise components without degrading the image quality, we adjust the parameter $\gamma$ according to image characteristics in the cost function (20).

The new cost function with the variable parameter $\gamma(\mathbf{x}_i)$ is represented by

$$\begin{aligned} J[\mathbf{F}] &= \sum_{i=1}^{N^2}\left(y_i - g(\mathbf{f}_i)\right)^2 \\ &\quad + \sum_{i=1}^{N^2}\gamma(\mathbf{x}_i)\left(f(\mathbf{x}_{i-N}) + f(\mathbf{x}_{i-1})\right. \\ &\qquad \left. - 4f(\mathbf{x}_i) + f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i+N})\right)^2. \end{aligned} \tag{23}$$

We should make the value of $\gamma(\mathbf{x}_i)$ larger at the flat parts to efficiently remove noise components, and should make it smaller at the edges to preserve the edge information. We minimize $J$ by using the steepest descent technique in the same way as (19), however the computation of $\partial J/\partial f(\mathbf{x}_i)$ is slightly different from the previous one. The detailed computation is shown in Appendix B.

In actual applications, we can obtain only a test image degraded by blur and observation noise, and the original image is unknown. Even if we pass the test image through a high pass filter, we can not accurately estimate the edge position

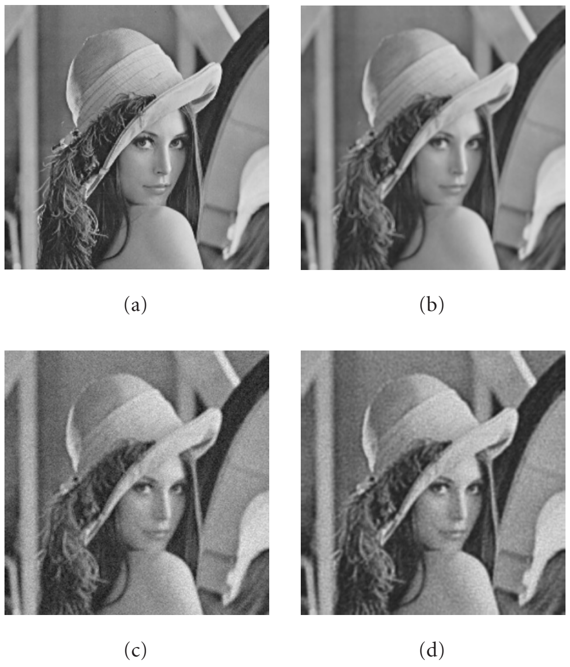

FIGURE 7: Results for Lena image: (a) original image $f(\mathbf{x}_i)$, (b) blurred image $g(\mathbf{f}_i)$, (c) degraded image $y_i$, and (d) restored image $\hat{f}(\mathbf{x}_i)$, (SNRI = 1.82).

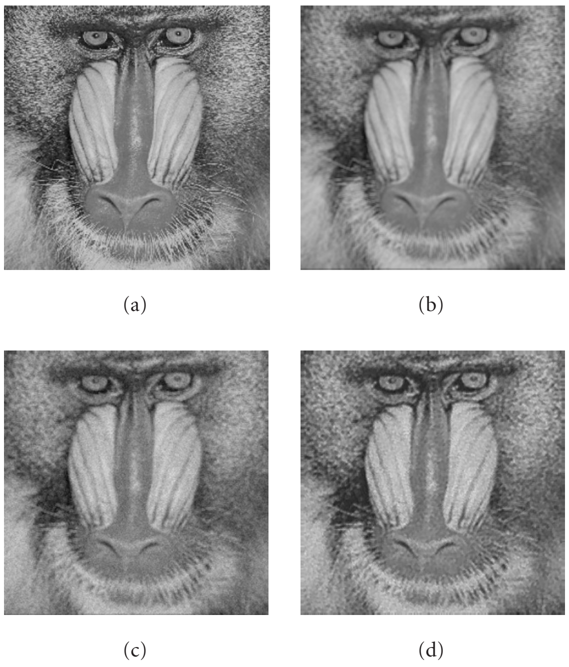

FIGURE 8: Results for Baboon image: (a) original image $f(\mathbf{x}_i)$, (b) blurred image $g(\mathbf{f}_i)$, (c) degraded image $y_i$, and (d) restored image $\hat{f}(\mathbf{x}_i)$, (SNRI = 0.97).

due to the increased effect of noise components. Therefore, we smoothed the test image by using the following mean filter of size $(3 \times 3)$:

$$s(\mathbf{x}_i) = \frac{1}{9}\big(y_{i-N-1} + y_{i-N} + y_{i-N+1} + y_{i-1} + y_i + y_{i+1} + y_{i+N-1} + y_{i+N} + y_{i+N+1}\big), \tag{24}$$

and then estimated the edge position by the size of the derivative of $s(\mathbf{x}_i)$, computed by

$$d(\mathbf{x}_i) = \sqrt{\{s(\mathbf{x}_{i-N}) - s(\mathbf{x}_{i+N})\}^2 + \{s(\mathbf{x}_{i-1}) - s(\mathbf{x}_{i+1})\}^2}. \tag{25}$$

The value of $d(\mathbf{x}_i)$ becomes small near the edges. We thus put

$$\gamma(\mathbf{x}_i) = Ae^{-ad(\mathbf{x}_i)} \tag{26}$$

so that the value of $\gamma(\mathbf{x}_i)$ becomes small near the edges, where $A$ and $a$ are constants. Equation (26) achieves a good tradeoff between restoration performance and computational simplicity, although other choices exist. We may estimate the edge position more accurately by using an order statistics filter instead of the mean filter (24). But the use of the order statistics filter increases the computational complexity and the restoration performance is fairly robust against the value of the regularization parameter $\gamma(\mathbf{x}_i)$ [16].

The restoration method with variable regularization parameter is summarized as follows.

Step (1) We smooth the test image by the mean filter (24) to generate the edge image $d(\mathbf{x}_i)$, and then compute the value of $\gamma(\mathbf{x}_i)$ by using (26).

Step (2) Using the training image, we initially estimate the blurring function by the GRBFN.

Step (3) Using the test image, we repeat the update procedure (19) 10 times to obtain a restored image at an intermediate stage.

Step (4) We regard the restored image as the original image, and again estimate the blurring function by using the test image.

Step (5) Repeat Step (3) and Step (4) until no significant improvement is attained in Step (3).

### 5.2. Restoration results

Figure 9 shows the restoration results for Lena image, where we put $A = 10^5$ and $a = 0.05$. These parameters were determined in a heuristic manner so that the SNRI becomes higher. Figures 9a, 9b, and 9c show the smoothed images $s(\mathbf{x}_i)$, the edge image $d(\mathbf{x}_i)$, and the finally restored image $\hat{f}(\mathbf{x}_i)$. The value of SNRI is also shown. Results for Baboon image are shown in Figure 10. These results show that changing $\gamma$ according to image characteristics improves the restoration performance by 0.25 (dB) and 0.39 (dB) for Lena and Baboon images, respectively.

We also apply the proposed method to a JPEG image taken by a digital camera. Figure 11 shows the result, where Figures 11a, 11b, 11c and 11d are the original, blurred, degraded, and restored images, respectively. The SNRI is 2.18 (dB), which is slightly higher than those of Lena and Baboon images. Because the JPEG image inherently contains a small amount of high-frequency components as high-frequency coefficients have been discarded in compression process.

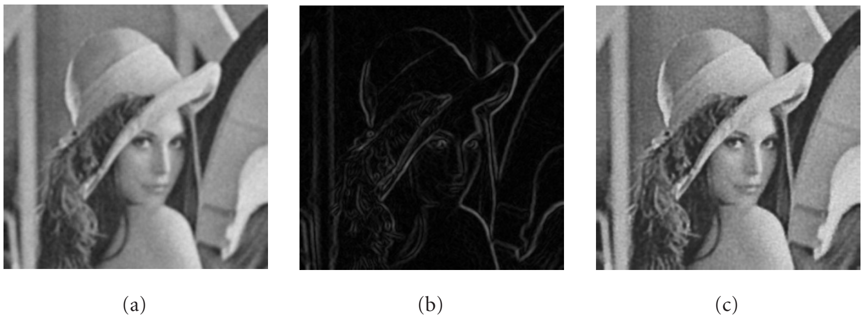

FIGURE 9: Results for Lena image with variable $\gamma$: (a) smoothed image $s(\mathbf{x}_i)$, (b) edge image $d(\mathbf{x}_i)$, and (c) restored image $\hat{f}(\mathbf{x}_i)$, (SNRI = 2.03).

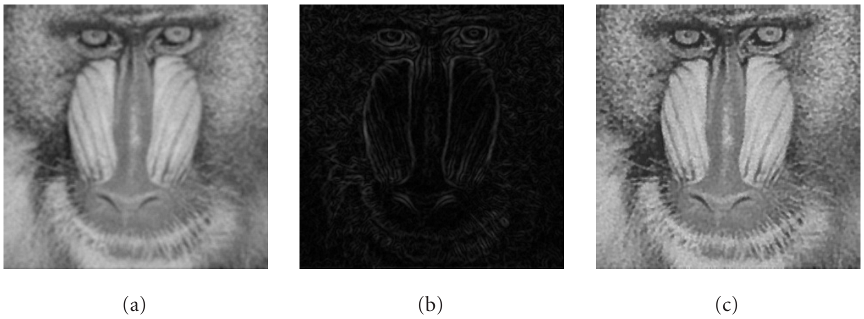

FIGURE 10: Results for Baboon with variable $\gamma$: (a) smoothed image $s(\mathbf{x}_i)$, (b) edge image $d(\mathbf{x}_i)$, and (c) restored image $\hat{f}(\mathbf{x}_i)$, (SNRI = 1.36).

## 6. CONCLUSION

We presented the nonlinear image restoration method by using the GRBFN and the regularization method. We used the GRBFN to estimate the nonlinear blurring function, and used the regularization method to restore the original image. We alternately used the two estimation methods for restoring the original image from the nonlinearly degraded image, under the single assumption of the smoothness of original image and blurring function. The salient feature is that the proposed estimation method is applicable even when the inverse of the blurring process does not exist. We also presented the adaptive regularization parameter selection method for further improvement of restoration performance.

## APPENDICES

## A. COMPUTATION OF $\partial J/\partial f(\mathbf{x}_i)$

The derivative of $J$ with respect to $f(\mathbf{x}_i)$ is represented by

$$\begin{aligned}\frac{\partial J}{\partial f(\mathbf{x}_i)} &= \frac{\partial}{\partial f(\mathbf{x}_i)}\sum_{j=1}^{N^2}\left(y_j - g(\mathbf{f}_j)\right)^2 \\ &\quad + \gamma\frac{\partial}{\partial f(\mathbf{x}_i)}\sum_{j=1}^{N^2}\big(f(\mathbf{x}_{j-N}) + f(\mathbf{x}_{j-1}) - 4f(\mathbf{x}_j) \\ &\qquad + f(\mathbf{x}_{j+1}) + f(\mathbf{x}_{j+N})\big)^2.\end{aligned} \tag{A.1}$$

The first term in the right-hand side of (A.1) is expressed as

$$\begin{aligned}&\frac{\partial}{\partial f(\mathbf{x}_i)}\sum_{j=1}^{N^2}\left(y_j - g(\mathbf{f}_j)\right)^2 \\ &\quad = 2\sum_{j=1}^{N^2}\left(y_j - g(\mathbf{f}_j)\right)\frac{\partial}{\partial f(\mathbf{x}_i)}\left(y_i - g(\mathbf{f}_j)\right) \\ &\quad = -2\left(y_{i-N-1} - g(\mathbf{f}_{i-N-1})\right)\frac{\partial g(\mathbf{f}_{i-N-1})}{\partial f(\mathbf{x}_i)} \\ &\qquad - 2\left(y_{i-N} - g(\mathbf{f}_{i-N})\right)\frac{\partial g(\mathbf{f}_{i-N})}{\partial f(\mathbf{x}_i)} \\ &\qquad - 2\left(y_{i-N+1} - g(\mathbf{f}_{i-N+1})\right)\frac{\partial g(\mathbf{f}_{i-N+1})}{\partial f(\mathbf{x}_i)} \\ &\qquad - 2\left(y_{i-1} - g(\mathbf{f}_{i-1})\right)\frac{\partial g(\mathbf{f}_{i-1})}{\partial f(\mathbf{x}_i)} \\ &\qquad - 2\left(y_i - g(\mathbf{f}_i)\right)\frac{\partial g(\mathbf{f}_i)}{\partial f(\mathbf{x}_i)} - 2\left(y_{i+1} - g(\mathbf{f}_{i+1})\right)\frac{\partial g(\mathbf{f}_{i+1})}{\partial f(\mathbf{x}_i)} \\ &\qquad - 2\left(y_{i+N-1} - g(\mathbf{f}_{i+N-1})\right)\frac{\partial g(\mathbf{f}_{i+N-1})}{\partial f(\mathbf{x}_i)} \\ &\qquad - 2\left(y_{i+N} - g(\mathbf{f}_{i+N})\right)\frac{\partial g(\mathbf{f}_{i+N})}{\partial f(\mathbf{x}_i)} \\ &\qquad - 2\left(y_{i+N+1} - g(\mathbf{f}_{i+N+1})\right)\frac{\partial g(\mathbf{f}_{i+N+1})}{\partial f(\mathbf{x}_i)}\end{aligned} \tag{A.2}$$

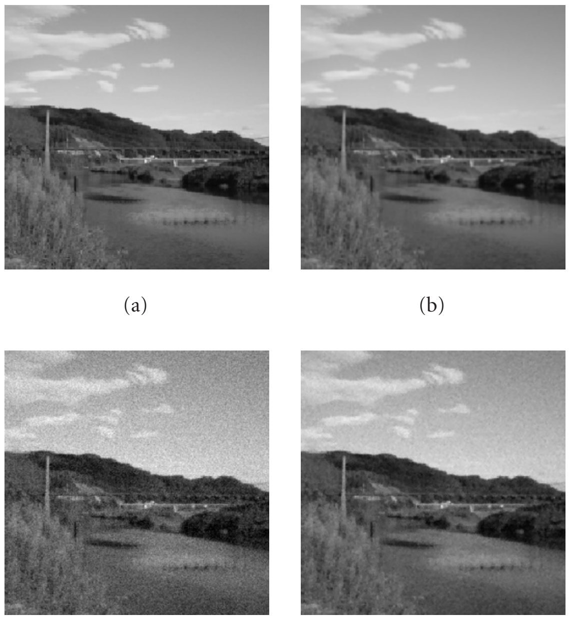

FIGURE 11: Results for a natural image: (a) original image $f(\mathbf{x}_i)$, (b) blurred image $g(\mathbf{f}_i)$, (c) degraded image $y_i$, and (d) restored image $\hat{f}(\mathbf{x}_i)$, (SNRI = 2.18).

with

$$\begin{aligned}
&\frac{\partial g(\mathbf{f}_{i-N-1})}{\partial f(\mathbf{x}_i)} \\
&\quad= \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_9}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i-N-1}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i-N})}{\partial f(\mathbf{x}_i)} \\
&\quad= \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_8}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i-N}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i-N+1})}{\partial f(\mathbf{x}_i)} \\
&\quad= \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_7}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i-N+1}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i-1})}{\partial f(\mathbf{x}_i)} \\
&\quad= \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_6}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i-1}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i})}{\partial f(\mathbf{x}_i)} = \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_5}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i+1})}{\partial f(\mathbf{x}_i)} = \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_4}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i+1}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i+N-1})}{\partial f(\mathbf{x}_i)} \\
&\quad= \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_3}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i+N-1}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i+N})}{\partial f(\mathbf{x}_i)} \\
&\quad= \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_2}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i+N}-\mathbf{t}_j|^2}{2\sigma^2}\right), \\
&\frac{\partial g(\mathbf{f}_{i+N+1})}{\partial f(\mathbf{x}_i)} \\
&\quad= \sum_{j=1}^{M^2} c_j\left(-\frac{f(\mathbf{x}_i)-(\mathbf{t}_j)_1}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{f}_{i+N+1}-\mathbf{t}_j|^2}{2\sigma^2}\right).
\end{aligned} \tag{A.3}$$

Here, $(\mathbf{t}_j)_l$ denotes the $l$th element of the 9-dimensional vector $\mathbf{t}_j$. The second term in the right-hand side of (A.1) is computed by

$$\begin{aligned}
&\gamma\frac{\partial}{\partial f(\mathbf{x}_i)}\sum_{j=1}^{N^2}\big(f(\mathbf{x}_{j-N})+f(\mathbf{x}_{j-1})-4f(\mathbf{x}_j) \\
&\qquad\qquad +f(\mathbf{x}_{j+1})+f(\mathbf{x}_{j+N})\big)^2 \\
&\quad=2\gamma\big\{f(\mathbf{x}_{i-2N})+f(\mathbf{x}_{i-N-1})-4f(\mathbf{x}_{i-N})+f(\mathbf{x}_{i-N+1})+f(\mathbf{x}_i) \\
&\qquad +f(\mathbf{x}_{i-N-1})+f(\mathbf{x}_{i-2})-4f(\mathbf{x}_{i-1})+f(\mathbf{x}_i)+f(\mathbf{x}_{i+N-1}) \\
&\qquad -4\big(f(\mathbf{x}_{i-N})+f(\mathbf{x}_{i-1})-4f(\mathbf{x}_i)+f(\mathbf{x}_{i+1})+f(\mathbf{x}_{i+N})\big) \\
&\qquad +f(\mathbf{x}_{i-N+1})+f(\mathbf{x}_i)-4f(\mathbf{x}_{i+1})+f(\mathbf{x}_{i+2}) \\
&\qquad +f(\mathbf{x}_{i+N+1})+f(\mathbf{x}_i)+f(\mathbf{x}_{i+N-1})-4f(\mathbf{x}_{i+N}) \\
&\qquad +f(\mathbf{x}_{i+N+1})+f(\mathbf{x}_{i+2N})\big\} \\
&\quad=2\gamma\big\{f(\mathbf{x}_{i-2N})+2f(\mathbf{x}_{i-N-1})-8f(\mathbf{x}_{i-N}) \\
&\qquad +2f(\mathbf{x}_{i-N+1})+f(\mathbf{x}_{i-2})-8f(\mathbf{x}_{i-1})+20f(\mathbf{x}_i) \\
&\qquad -8f(\mathbf{x}_{i+1})+f(\mathbf{x}_{i+2})+2f(\mathbf{x}_{i+N-1})-8f(\mathbf{x}_{i+N}) \\
&\qquad +2f(\mathbf{x}_{i+N+1})+f(\mathbf{x}_{i+2N})\big\}.
\end{aligned} \tag{A.4}$$

## B. COMPUTATION OF $\partial J/\partial f(\mathbf{x}_i)$ WITH VARIABLE $\gamma$

The derivative of $J$ with respect to $f(\mathbf{x}_i)$ is represented by

$$\begin{aligned}
\frac{\partial J[\mathbf{F}]}{\partial f(\mathbf{x}_i)} &= \frac{\partial}{\partial f(\mathbf{x}_i)}\sum_{j=1}^{N^2}\left(y_j-g(\mathbf{f}_j)\right)^2 \\
&\quad+\frac{\partial}{\partial f(\mathbf{x}_i)}\sum_{j=1}^{N^2}\gamma(\mathbf{x}_i)\big(f(\mathbf{x}_{j-N})+f(\mathbf{x}_{j-1})-4f(\mathbf{x}_j) \\
&\qquad\qquad +f(\mathbf{x}_{j+1})+f(\mathbf{x}_{j+N})\big)^2.
\end{aligned} \tag{B.1}$$

The first term in the right-hand side of (B.1) is computed by (A.2). The second term in the right-hand side of (B.1) can be computed by

$$\begin{aligned}
&\frac{\partial}{\partial f(\mathbf{x}_i)} \sum_{j=1}^{N^2} \gamma(\mathbf{x}_j)\Big\{ f(\mathbf{x}_{j-N}) + f(\mathbf{x}_{j-1}) - 4f(\mathbf{x}_j) \\
&\qquad + f(\mathbf{x}_{j+1}) + f(\mathbf{x}_{j+N})\Big\}^2 \\
&= 2\Big\{\gamma(\mathbf{x}_{i-N})\Big(f(\mathbf{x}_{i-2N}) + f(\mathbf{x}_{i-N-1}) \\
&\qquad - 4f(\mathbf{x}_{i-N}) + f(\mathbf{x}_{i-N+1}) + f(\mathbf{x}_i)\Big) \\
&\quad + \gamma(\mathbf{x}_{i-1})\Big(f(\mathbf{x}_{i-N-1}) + f(\mathbf{x}_{i-2}) - 4f(\mathbf{x}_{i-1}) \\
&\qquad + f(\mathbf{x}_i) + f(\mathbf{x}_{i+N-1})\Big) \\
&\quad - 4\gamma(\mathbf{x}_i)\Big(f(\mathbf{x}_{i-N}) + f(\mathbf{x}_{i-1}) - 4f(\mathbf{x}_i) \\
&\qquad + f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i+N})\Big) \\
&\quad + \gamma(\mathbf{x}_{i+1})\Big(f(\mathbf{x}_{i-N+1}) + f(\mathbf{x}_i) - 4f(\mathbf{x}_{i+1}) \\
&\qquad + f(\mathbf{x}_{i+2}) + f(\mathbf{x}_{i+N+1})\Big) \\
&\quad + \gamma(\mathbf{x}_{i+N})\Big(f(\mathbf{x}_i) + f(\mathbf{x}_{i+N-1}) - 4f(\mathbf{x}_{i+N}) \\
&\qquad + f(\mathbf{x}_{i+N+1}) + f(\mathbf{x}_{i+2N})\Big)\Big\}.
\end{aligned} \tag{B.2}$$

## REFERENCES

[1] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[2] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.

[3] F. M. A. Acosta, "Radial basis function and related models: an overview," *Signal Processing*, vol. 45, no. 1, pp. 37–58, 1995.

[4] B. Mulgrew, "Applying radial basis functions," *IEEE Signal Processing Magazine*, vol. 13, no. 2, pp. 50–65, 1996.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, USA, 1999.

[6] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Processing Magazine*, vol. 13, no. 3, pp. 43–64, 1996.

[7] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Magazine*, vol. 14, no. 2, pp. 24–41, 1997.

[8] Y.-L. You and M. Kaveh, "A regularization approach to joint blur identification and image restoration," *IEEE Trans. Image Processing*, vol. 5, no. 3, pp. 416–428, 1996.

[9] I. Cha and A. Kassam, "RBFN restoration of nonlinearly degraded images," *IEEE Trans. Image Processing*, vol. 5, no. 6, pp. 964–975, 1996.

[10] A. K. Katsaggelos, *Digital Image Restoration*, Springer-Verlag, New York, NY, USA, 1991.

[11] C. A. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions," *Constr. Approx.*, vol. 2, no. 1, pp. 11–22, 1986.

[12] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[13] A. Sherstinsky and R. W. Picard, "On the efficiency of the orthogonal least squares training method for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 195–200, 1996.

[14] S. Lee and R. M. Kil, "A Gaussian potential function network with hierarchically self-organizing learning," *Neural Networks*, vol. 4, no. 2, pp. 207–224, 1991.

[15] A. G. Bors and M. Gabbouj, "Minimal topology for a radial basis functions neural network for pattern classification," *Digital Signal Processing*, vol. 4, no. 3, pp. 173–188, 1994.

[16] N. P. Galatsanos and A. K. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," *IEEE Trans. Image Processing*, vol. 1, no. 3, pp. 322–336, 1992.

**Keiji Icho** received the B.E. and M.E. degrees in communications engineering from Osaka University, Osaka, Japan, in 1999 and 2001, respectively. Since 2001, he has been working on software development for communication systems at Matsushita Electric Industrial Co., Ltd., Osaka, Japan.

**Youji Iiguni** received the B.E. and M.E. degrees in applied mathematics and physics from Kyoto University, Kyoto, Japan, in 1982 and 1984, respectively, and the D.E. degree from Kyoto University in 1990. He was an Assistant Professor at Kyoto University from 1984 to 1995, and an Associate Professor at Osaka University from 1995 to 2003. Since 2003, he has been a Professor at Osaka University. His research interests include signal and image processing.

**Hajime Maeda** received the B.E., M.E., and D.E. degrees in communications engineering from Osaka University, Osaka, Japan, in 1966, 1968, and 1971, respectively. He was an Assistant Professor at Osaka University from 1971 to 1975, an Associate Professor from 1975 to 1993, and a Professor from 1993 to 2001. He has passed away on September 8, 2001. His research interests included control theory, nonlinear analysis, and signal processing.