Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 605241, 14 pages
doi:10.1155/2010/605241

Research Article

Deconvolution of Defocused Image with Multivariate Local Polynomial Regression and Iterative Wiener Filtering in DWT Domain

Liyun Su[1] and Fenglan Li[2]

[1] *School of Mathematics and Statistics, Chongqing University of Technology, Chongqing 400054, China*
[2] *Library, Chongqing University of Technology, Chongqing 400054, China*

Correspondence should be addressed to Liyun Su, cloudhopping@163.com

Received 17 April 2010; Revised 23 June 2010; Accepted 10 August 2010

Academic Editor: Panos Liatsis

A novel semiblind defocused image deconvolution technique is proposed, which is based on multivariate local polynomial regression (MLPR) and iterative Wiener filtering (IWF). In this technique, firstly a multivariate local polynomial regression model is trained in wavelet domain to estimate defocus parameter. After obtaining the point spread function (PSF) parameter, iterative wiener filter is adopted to complete the restoration. We experimentally illustrate its performance on simulated data and real blurred image. Results show that the proposed PSF parameter estimation technique and the image restoration method are effective.

1. Introduction

Shooting a real world image with a camera through an optical device gives a 2D image where at least some parts are affected by a blur and noise. Images can be blurred by atmospheric turbulence, relative motion between sensors and objects, longer exposures, and so on, but the exact cause of blurring may be unknown. Restoration of blurred noisy images [1–3] is one of the main topics in many processing. The literatures [4–6] have given good methods to improve image qualities. The purpose of image restoration is to reconstruct an unobservable true image from a degraded observation. An observed image can be written, ignoring additive noise, as the two-dimensional (2D) convolution of the true image with a linear space-invariant (LSI) blur, known as the PSF. Restoration in the case of known blur, assuming the linear degradation model, is called linear image restoration and it has been presented extensively in the last three decades giving rise to a variety of solutions [7–10]. In many practical situations, however, the blur is unknown. Hence, both blur identification and image restoration must be performed from the degraded image. Restoration in the case of unknown blur is called blind image restoration [11–13]. Existing blind restoration methods can be categorized into two main groups: (i) those which estimate the PSF a priori independent of the true image so as to use it later with one of the linear image restoration methods, such as zero sheet separation, generalized cross validation, and maximum likelihood and expectation maximization based on the ARMA image model [14–16], and (ii) those which estimate the PSF and the true image simultaneously, such as nonnegative sand support constraints recursive inverse filtering, maximum likelihood and conjugate gradient minimization, and simulated annealing [17, 18]. Algorithms belonging to the first class are computationally simple, but they are limited to situations in which the PSF has a special form, and the true image has certain features. Algorithms belonging to the second class, which are computationally more complex, must be used for more general situations. In this paper, a kind of semi-blind image restoration algorithm is proposed in case of known the blur type (defocused blurring).

In general, discrete model for a linear degradation caused by blurring can be given by the following equation:

$$y(i,j) = h(i,j) * f(i,j) + n(i,j), \tag{1.1}$$

where $*$ indicates two-dimensional convolution, $f(i,j)$ represents in original image, $y(i,j)$ is the degraded image, $h(i,j)$ represents the two-dimensional PSF, and $n(i,j)$ is the additive noise. In this paper, we deal only with additive Gaussian noise, as it effectively models the noise in many different imaging scenarios. The difficulty in solving the restoration problem with a spatially varying blur commonly motivates the use of a stationary model for the blur. This leads to the following expression for the degradation system:

$$y(i,j) = h(i,j) * f(i,j) + n(i,j) = \sum_{k=1}^{M}\sum_{l=1}^{N} h(i-k, j-l) f(k,l) + n(i,j). \tag{1.2}$$

The use of linear techniques for solving the restoration problem is facilitated by using space-invariant model. Models that utilize space-variant degradations are also common, but lead to more complex solutions. As for defocus blur, PSF is modeled as a uniform intensity distribution within a circular disk,

$$h(i,j) = \begin{cases} \dfrac{1}{\pi R^2}, & \text{if } \sqrt{i^2 + j^2} \le R, \\ 0, & \text{otherwise,} \end{cases} \tag{1.3}$$

where disk radius R is the only unknown parameter for this type of blur.

Many existing image restoration algorithms assume that the PSF is known, but in practical it is not always the case. The restoration without knowing of the PSF is called blind image restoration. Fourier methods can be used to estimate the defocus parameter R through calculating a ratio of power of high frequencies portion to that of low frequencies portion. However, a main drawback of the method is its bad noise immunity. To solve this problem, [19] proposed a novel algorithm to overcome this shortcoming, the RBF neural network is applied to fit R. This scheme has good fitting, but bad prediction. To avoid the weak generalization ability, an more efficient method for estimating parameter R is proposed in this paper. The prediction ability of these two methods is compared with the trained five images. The steps of the presented algorithm in this paper is as follows: Firstly we construct feature vectors of several blurred images with known defocus radius R in wavelet domain, then a multivariate local polynomial estimation model [20–22] is trained using the vectors as inputs and defocus parameters as outputs. After the model is trained, the new defocused images are applied to the trained model for predicting the parameter R. For a semi-blind defocused image, R can be estimated through calculating the feature vectors and using it as input of the trained model. With known radius R, many traditional algorithm could be applied to restore the degraded image. In this paper, iterative Wiener filtering [23] is adopted to image restoration.

2. Relationship between Wavelet Coefficients and R

The wavelet transform provides a powerful and versatile framework for image processing. It is widely used in the fields of image denoising, compression, fusion, image restoration [24–26], and so forth.

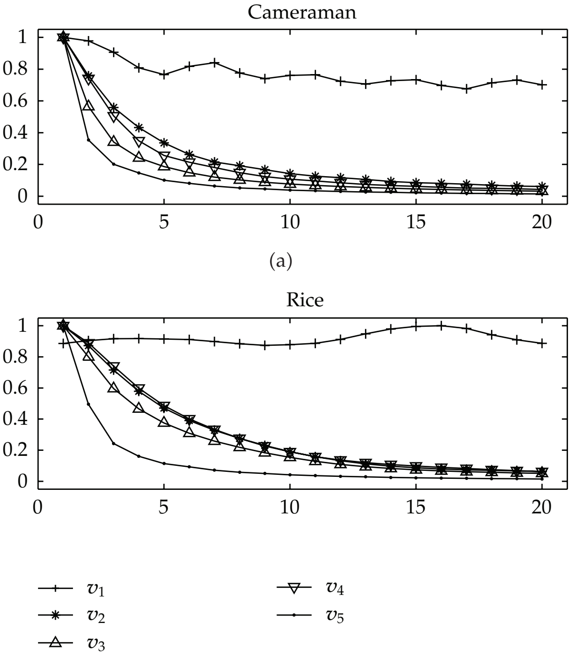
The two-dimensional discrete wavelet transform (DWT) [27, 28] hierarchically decompose an input image into a series of successively lower resolution images and their associated detail images. DWT is implemented by a set of filters, which are convolved with the image rows and columns. An image is convolved with low-pass and high-pass filters and the odd samples of the filtered outputs are discarded resulting in downsampling the image by a factor of 2. The l level wavelet decomposition of an image I results in an approximation image X_l and three detail images H_l, V_l, and D_l in horizontal, vertical, and diagonal directions respectively. Decomposition into l levels of an original image results in a down sampled image of resolution 2^l with respect to the image as well as detail images.

When an image is defocused, edged in it are smoothed and widened. The amount of high frequency band decreased, and that corresponding to low frequency band increases.

In order to denote the relationship between wavelet coefficients and defocused radius R, we define five variables named v_1, v_2, v_3, v_4, and v_5 as follows:

$$\begin{aligned} v_1 &= \frac{|V_2|_s}{|H_2|_s}, \\ v_2 &= \frac{|H_2|_s}{|X_2|_s}, \\ v_3 &= \frac{|H_1|_s}{\text{num}\{H_1\}}, \\ v_4 &= \frac{|H_2|_s}{\text{num}\{H_2\}}, \\ v_5 &= \frac{|D_1|_s}{\text{num}\{D_1\}}, \end{aligned} \tag{2.1}$$

where $|\cdot|_s$ represents the summation of all coefficients' absolute value, $\text{num}\{\cdot\}$ is total number of coefficients.



(a)

(b)

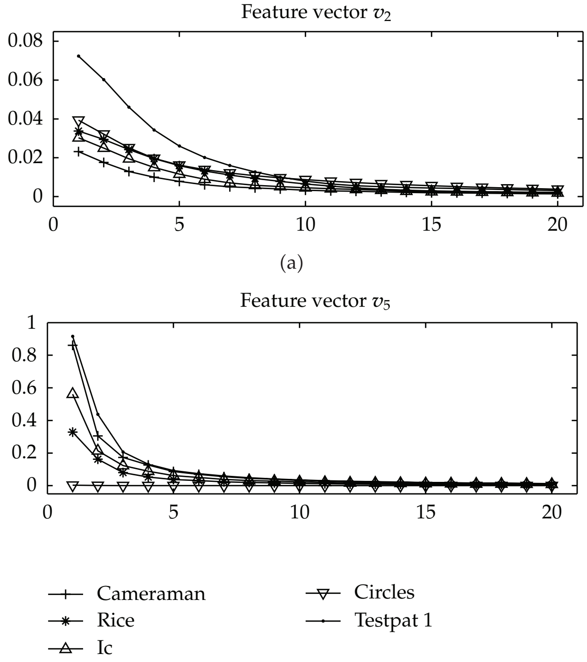
Figure 1: Relationship Between v_{1-5} and R.

An original image is blurred artificially by a uniform defocus PSF with R whose value rang from 1 to 20. The relationship between v_1, v_2, v_3, v_4, v_5 and R is shown in Figure 1, where the curves are normalized in $[0, 1]$ interval. When R increases, v_2, v_3, v_4 and v_5 decrease monotonously.

In order to estimate defocus parameter R, only known the roughly similar relationship is not enough. As shown in Figure 2, every image has monotonous curve between v_2, v_5 and R, but they are not superposition. For a degraded unknown PSF image, R cannot be calculated because the curve of the given image is not known. For example, if v_2 of image "rice" has been calculated, and then we estimate R according curve if "ic" in Figure 2, wrong results are obtained obviously. To solve this problem, one of the methods is to choose neural networks. Computational artificial neural networks are known to have the capability for performing complex mappings between input and output data, but neural network method has bad generalization ability. Here we propose a multivariate local polynomial regression model to estimate R. The variables v_{1-5} are chosen to train the multivariate local polynomial estimation model. Prediction Comparison is made to verify the advantages of multivariate local polynomial fitting.

3. Training Multivariate Local Polynomial Estimation Model

Multivariate local polynomial fitting is an attractive method both from theoretical and practical point of view. Multivariate local polynomial method has a small mean squared error compared with the Nadaraya-Watson estimator which leads to an undesirable form of the bias and the Gasser-Muller estimator which has to pay a price in variance when dealing



(a)

(b)

Figure 2: Curves v_2 and v_5 of Different Images.

with a random design model. Multivariate local polynomial fitting also has other advantages. The method adapts to various types of designs such as random and fixed designs, highly clustered and nearly uniform designs. Furthermore, there is an absence of boundary effects: the bias at the boundary stays automatically of the same order as the interior, without use of specific boundary kernels. The local polynomial approximation approach is appealing on general scientific grounds: the least squares principle to be applied opens the way to a wealth of statistical knowledge and thus easy generalizations. In this Section, we briefly outline and review the idea of the extension of multivariate local polynomial fitting [20–22] to the parameter R of defocused PSF.

3.1. Multivariate Kernel Function

To localize data in the m-dimension, we need a multikernel function. Generally speaking, a multivariate kernel function refers to an m-variate function satisfying

$$\int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} K(\underline{x})\, d\underline{x} = 1. \tag{3.1}$$

Here and hereafter, we use $\int$ to indicate multivariate integration over the m-dimensional Euclidean space.

There are two common methods for constructing multivariate kernel functions. For a univariate kernel $k(x)$, the product kernel is given by

$$K(\underline{x}) = \prod_{i=1}^{m} k(x_i), \tag{3.2}$$

and the spherically symmetric kernel is defined as

$$K(\underline{x}) = c_{K,m} K(\|\underline{x}\|), \tag{3.3}$$

where $c_{K,m} = \{\int K(\|\underline{x}\|)d\underline{x}\}^{-1}$ is a normalization constant and $\|\underline{x}\| = (x_1^2 + x_2^2 + \cdots + x_m^2)^{-1/2}$. Popular choices of $K(\underline{x})$ include the standard d-variate normal density

$$K(\underline{x}) = (2\pi)^{-m/2} \exp\left(-\frac{\|\underline{x}\|^2}{2}\right), \tag{3.4}$$

and the spherical *Epanechnikov* kernel

$$K(\underline{x}) = \left\{\frac{d(d+2)\Gamma(m/2)}{4\pi^{m/2}}\right\}\left(1 - \|\underline{x}\|^2\right)_+. \tag{3.5}$$

The latter is the optimal kernel, according to Fan and Gijbels [21] and Su [22].

The localization in multivariate nonparametric regression is frequently carried out by the kernel weighting. Let $\underline{H}$ be a symmetric positive-definite matrix called a bandwidth matrix. The localization scheme at a point $\underline{x}$ assigns the weight

$$K_{\underline{H}}\left(\underline{X_i} - \underline{x}\right), \quad \text{with } K_{\underline{H}}(\underline{x}) = \left|\underline{H}\right|^{-1} K\left(\underline{H}^{-1}\underline{x}\right), \tag{3.6}$$

where $|\underline{H}|$ is the determinant of the matrix $\underline{H}$. The bandwidth matrix is introduced to accommodate the dependent structure in the independent variables. For practical problems, the bandwidth matrix $\underline{H}$ is taken to be a diagonal matrix. The different independent variables will be accommodated into different scales. For simplification, the bandwidth matrix is designed into $\underline{H} = h\underline{I_m}$ ($\underline{I_m}$ denoting the identity matrix of order m).

3.2. Multivariate Predictor with Local Polynomial Fitting

Suppose that the input vector is $\underline{V} = (v_1, v_2, v_3, v_4, v_5)$, the model is fitted by the function

$$R = f(\underline{V}). \tag{3.7}$$

Our purpose is to obtain the estimation $\widehat{R}=\widehat{f}(\underline{V})$ of function f, this paper, we use the dth order multivariate local polynomial $f(\underline{V})$ to predict the defocused parameter R_T value based on the point $\underline{V_T}$ of the test image. The polynomial function can be described as

$$f(\underline{V}) \approx \sum_{0\le|\underline{j}|\le d} \frac{1}{\underline{j}!} D^{(\underline{j})} f_i\left(\underline{V_T}\right)\left(\underline{V}-\underline{V_T}\right)^{\underline{j}} = \sum_{0\le|\underline{j}|\le d} \underline{b}_{\underline{j}}\left(\underline{V_T}\right)\left(\underline{V}-\underline{V_T}\right)^{\underline{j}}, \tag{3.8}$$

where

$$\begin{gathered} m=5, \quad \underline{j}=(j_1,j_2,\dots,j_m), \quad \underline{j}!=j_1!j_2!\cdots j_m!, \quad \left|\underline{j}\right|=\sum_{l=1}^{m} j_l, \\ \sum_{0\le|\underline{j}|\le d} = \sum_{|\underline{j}|=0}^{d}\left(\underset{|\underline{j}|=j_1+j_2+\cdots+j_m}{\sum_{j_1=0}^{|\underline{j}|}\sum_{j_2=0}^{|\underline{j}|}\cdots\sum_{j_m=0}^{|\underline{j}|}}\right), \qquad \underline{V}^{\underline{j}}=v_1^{j_1}v_2^{j_2}\cdots v_m^{j_m}, \\ D^{(\underline{j})} f_i\left(\underline{V_T}\right)=\left.\frac{\partial^{|\underline{j}|} f_i(\underline{V})}{\partial v_1^{j_1}\partial v_2^{j_2}\cdots\partial v_m^{j_m}}\right|_{\underline{V}=\underline{V_T}}, \qquad \underline{b}_{\underline{j}}\left(\underline{V_T}\right)=\frac{1}{\underline{j}!}D^{(\underline{j})} f_i\left(\underline{V_T}\right). \end{gathered} \tag{3.9}$$

In the multivariate prediction method, $\underline{V_{T_a}}(a=1,2,\dots,A)$ denotes the trained image feature vectors. Using A pairs of $(\underline{V_{T_a}},R_a)$, for which the values are already known, the coefficients of f_i is determined by minimizing

$$\sum_{a=1}^{A}\left[R_a-\sum_{0\le\left|\underline{j}\right|\le d}\underline{b}_{\underline{j}}\left(\underline{V_T}\right)\left(\underline{V_{T_a}}-\underline{V_T}\right)^{\underline{j}}\right]^2 \cdot K_{\underline{H}}\left(\underline{V_{T_a}}-\underline{V_T}\right). \tag{3.10}$$

For the weighted least squared problem, a matrix form can be described by

$$\underline{W}^{1/2}\cdot\underline{Y}=\underline{W}^{1/2}\cdot\underline{X}\cdot\underline{B}+\underline{\varepsilon}, \tag{3.11}$$

where

$$\begin{gathered} \underline{Y}=(y_1,y_2,\dots,y_A)^T, \qquad y_a=R_a, \\ \underline{B}=\left(\underline{b}_0\left(\underline{V_T}\right),\underline{b}_1\left(\underline{V_T}\right),\dots,\underline{b}_d\left(\underline{V_T}\right)\right)^T, \\ \underline{W}=\operatorname{diag}\left\{K_{\underline{H}}\left(\underline{V_{T_1}}-\underline{V_T}\right),K_{\underline{H}}\left(\underline{V_{T_2}}-\underline{V_T}\right),\dots,K_{\underline{H}}\left(\underline{V_{T_A}}-\underline{V_T}\right)\right\}, \end{gathered} \tag{3.12}$$

and $\underline{X}$ is the $A \times S(S = \sum_{0\le|\underline{j}|\le d} |\underline{j}|/\underline{j}!)$,

$$\underline{X} = \begin{pmatrix} 1 & \left(\underline{V_{T_1}} - \underline{V_T}\right)^1 & \cdots & \left(\underline{V_{T_1}} - \underline{V_T}\right)^d \\ 1 & \left(\underline{V_{T_2}} - \underline{V_T}\right)^1 & \cdots & \left(\underline{V_{T_2}} - \underline{V_T}\right)^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \left(\underline{V_{T_A}} - \underline{V_T}\right)^1 & \cdots & \left(\underline{V_{T_A}} - \underline{V_T}\right)^d \end{pmatrix}. \tag{3.13}$$

We then have the least squared solution with multivariate local polynomial fitting as follows:

$$\underline{\widehat{B}} = \left(\underline{W}^{1/2}\underline{X}\right)^{\dagger}\underline{Y} \tag{3.14}$$

or, when $\underline{X}^T\underline{WX}$ is inverse, the estimation can be written by

$$\underline{\widehat{B}} = \left(\underline{X}^T\underline{WX}\right)^{-1}\underline{X}^T\underline{WY}, \tag{3.15}$$

then we can get the estimation $\widehat{R}_T = \widehat{f}(\underline{V_T})$

$$\widehat{R}_T = \widehat{f}\left(\underline{V_T}\right) = \underline{E_1}\left(\underline{X}^T\underline{WX}\right)^{-1}\underline{X}^T\underline{WY}, \tag{3.16}$$

where $\underline{E_1} = (1, 0, 0, \ldots, 0)_{1\times S}$.

Computing the $\underline{\widehat{B}}$ will suffer from large computational cost. we can use the recursive least squared method to reduce the computation complexity, and it is very powerful especially in the real time prediction problems. There are several important issues about the bandwidth, the order of multivariate local polynomial function and the kernel function which have to be discussed. The three problems will be presented in Section 3.3.

3.3. Parameters Selections

For the multivariate local polynomial predictor, there are three important problems which have significant influence to the prediction accuracy and computational complexity. First of all, there is the choice of the bandwidth matrix, which plays a rather crucial role. The bandwidth matrix $\underline{H}$ is taken to be a diagonal matrix. For simplification, the bandwidth matrix is designed into $\underline{H} = h\underline{I_m}$. So the most important thing is to find the bandwidth h. A too big bandwidth underparameterizes the regression function, causing a large modeling bias, while a too small bandwidth overparameterizes the unknown function and results in noisy estimates. In theory, there exists an optimal bandwidth h_{opt} in the meaning of mean squared error, such that

$$h_{\text{opt}} = \arg\min_h \int \left(f(\underline{x}) - \widehat{f}(\underline{x})\right)^2 d\underline{x}. \tag{3.17}$$

But the optimal bandwidth cannot be solved directly. So we discuss how to get the asymptotically optimal bandwidth. There are quite a few important techniques for selecting the bandwidth. such as cross-validation and plug-in bandwidth selectors. a conceptually simple technique, with theoretical justification and good empirical performance, is the plug-in technique.

Another issue in multivariate local polynomial fitting is the choice of the order of the polynomial. Since the modeling bias is primarily controlled by the bandwidth, this issue is less crucial however. For a given bandwidth h, a large value of d would expectedly reduce the modeling bias, but would cause a large variance and a considerable computational cost. Since the bandwidth is used to control the modeling complexity, and due to the sparsity of local data in multidimensional space, a higher-order polynomial is rarely used. We use the local quadratic regression to indicate the flavor of the multivariate local polynomial fitting, that is to say, $d = 2$.

The third issue is the selection of the kernel function. In this paper, of course, we choose the optimal spherical *Epanechnikov* kernel function, which minimizes the asymptotic MSE of the resulting multivariate local polynomial estimators, as our kernel function.

3.4. Estimating the Defocused Parameter

Twenty original images are chosen to train the model. The images are defocused artificially with R whose value ranging from 2 to 7. So the total number of training samples are 120. Then feature vectors are constructed using variables v_{1-5} of each image:

$$\underline{V} = (v_1, v_2, v_3, v_4, v_5). \tag{3.18}$$

The defocused parameter R is the model output.

When training samples $\{\underline{V}_{T_a}, R_a\}_{a=1}^{120}$ are given, obtaining weights matrix B, according to the relationship between the V and R, then the defocused parameter R can be calculated using the trained model.

4. Iterative Wiener Filter

Wiener filtering (minimizing mean square error) is commonly used to restore linearly-degraded images. To obtain optimal results, there must be accurate knowledge of the covariance of the ideal image. In this section, the so-called iterative Wiener filter [19, 23] is used to restore the original image.

The imaging system H is assumed to be linear shift invariant with additive, independent, white noise processes of known variance. the model for the observed image y is given in matrix notation by

$$y = Hf + n, \tag{4.1}$$

where f is the ideal image. The optimal linear minimum mean-squared error, or Wiener restoration filter given by

$$\widehat{f} = By, \tag{4.2}$$

where $B = R_{ff}H^T[HR_{ff}H^T + R_{nn}]^{-1}$, requires accurate knowledge of R_{ff}, the autocorrelation of ideal image f. However, in practical situations f is usually not available and only a single copy of the blurred image to be restored, y, is provided. In the absence of a more accurate knowledge of the ideal image f, the blurred image y is often used in its place simply because there is no other information about f readily available. The signal y is subsequently used to compute an estimate of R_{ff} and this estimate is used in place of R_{ff} in (4.2).

The following summarizes the iterative Wiener filtering procedure.

Step 1 (Initialization). Use y to compute an initial ($i = 0$) estimate of R_{ff} by

$$R_{ff}(0) = R_{yy} = E\left\{yy^T\right\}. \tag{4.3}$$

Step 2 (Filter Construction). Use $R_{ff}(i)$, the ith estimate of R_{ff} to construct the $(i+1)$th restoration filter $B(i+1)$ given by

$$B_{i+1} = R_{ff}H^T\left[HR_{ff}H^T + R_{nn}\right]^{-1}. \tag{4.4}$$

Step 3 (Restoration). Restore y by the $B(i+1)$ filter to obtain $\hat{f}(i+1)$, the $(i+1)$th estimate of f

$$\hat{f}(i+1) = B(i+1)y. \tag{4.5}$$

Step 4 (Update). Use $\hat{f}(i+1)$ to compute an improved estimate of R_{ff}, given by

$$R_{ff}(i+1) = E\left\{\hat{f}(i+1)\hat{f}^T(i+1)\right\}. \tag{4.6}$$

Step 5 (Iteration). Increment i and repeat Steps 2, 3, 4, and 5.

5. Experimental Results and Analysis

The experiments are carried out by using the Matlab image processing toolbox. The performance of the proposed image restoration algorithm has been evaluated using the classical gray-scale Moon image, Coins image, Saturn image, and Tire image in Matlab toolbox. To verify the good ability of restoration of the proposed algorithm, one real blurred image is used for the deconvolution procedure. The results show our method is very successful for this kind of blurred image.

In image restoration studies, the degradation modelled by blurring and additive noise is referred to in terms of the metric blurred signal-to-noise ratio (BSNR). This metric for a zero-mean $M \times N$ image is given by

$$\text{BSNR} = 10\log_{10}\left\{\frac{(1/\text{MN})\sum_{m=1}^{M}\sum_{n=1}^{N} z^2(m,n)}{\sigma_v^2}\right\}, \tag{5.1}$$

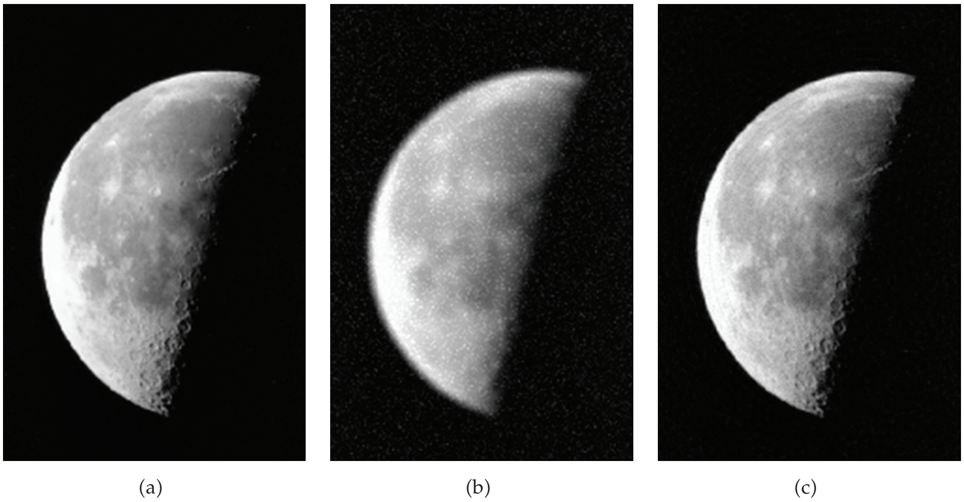


Figure 3: Result of Moon. True image (a), blurred image (b), estimated image(c), BSNR = 12.35, ISNR = 22.56.

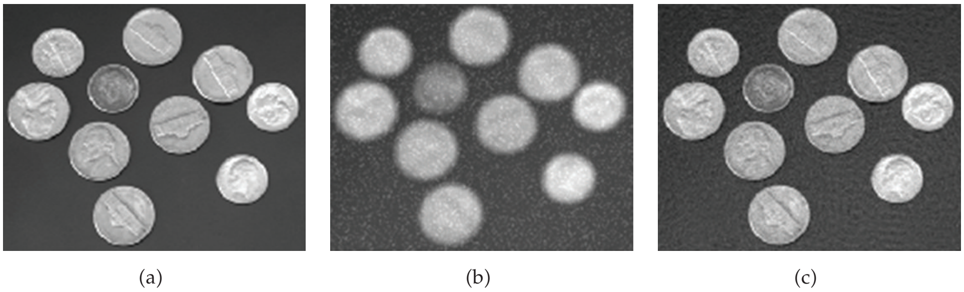


Figure 4: Result for Coins. True image (a), blurred image (b), restored image (c), BSNR = 11.22, ISNR = 23.14.

where $z(m,n)$ is the noise-free blurred image and σ_v^2 is the additive noise variance.

For the purpose of objectively testing the performance of linear image restoration algorithms, the improvement in signal-to-noise ratio (ISNR) is often used. ISNR is defined as

$$\mathrm{ISNR} = 10\log_{10}\left\{\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[f(m,n)-y(m,n)\right]^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[f(m,n)-\widehat{f}(m,n)\right]^2}\right\}, \tag{5.2}$$

where $f(m,n)$ and $y(m,n)$ are the original and degraded image pixel intensity values and $\widehat{f}(m,n)$ is the restored true image pixel intensity value. ISNR cannot be used when the true image is unknown, but it can be used to compare different methods in simulations when the true image is known.

In order to find the good performance of the proposed multivariate local polynomial Regression method (MLPR) compared with the RBF neural network algorithm (RBFNN) which is investigated in the literature [19], the same defocused blurred images are used for

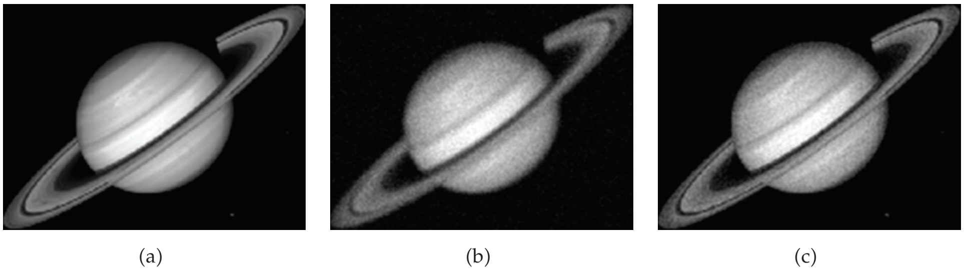


Figure 5: Result for Saturn. True image (a), blurred image (b), restored image (c), BSNR = 13.17, ISNR = 24.31.

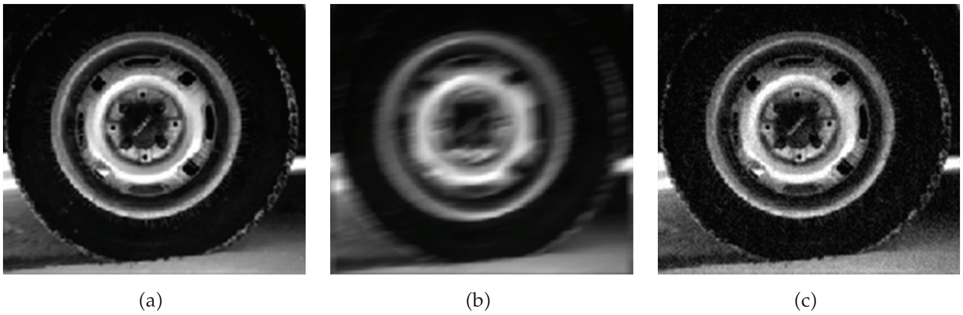


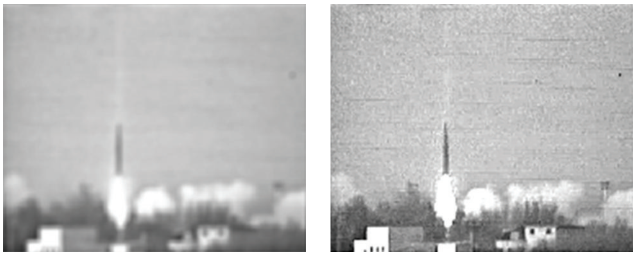
Figure 6: Result for *Tire*. True image (a), blurred image (b), restored image (c), BSNR = 11.56, ISNR = 22.09.

the experiments. Mean squared prediction errors are shown in Table 1. From Table 1, we can conclude that the prediction results of MLPR predictor are significantly better than the RBF neural network method in the same simulated data.

Figures 3, 4, 5 and 6, in which the true images, blurred images and estimated true images are depicted in the left, middle and right column, respectively, illustrate how the method behaves in Moon, Coins, Saturn and Tire images. It is clear from Figures 3–6 that performance of the new method is effective in different images. Figure 7 also shows that the presented algorithm is good for real blurred image.

6. Conclusions

A new method that is based on multivariate local polynomial regression model and iterative Wiener filtering for semi-blind restoration of blurred noisy images was proposed in this paper. Defocused parameter was estimated by a multivariate local polynomial regression model trained in wavelet domain. The main advantages of the proposed technique are that it is not only robust to noise because wavelet transform has an excellent denoising ability, but also effective to artificially and practically defocused blurred image. Restoration is successfully realized by the iterative Wiener filter, resulting in improved the image quality. The algorithm was justified via simulation and real image. Defocused image parameter can be successfully estimated by using trained model. Experimental results show the proposed algorithm is reliable and robust for defocused blurred image restoration.



(a) (b)

Figure 7: Result for real blurred image. blurred image (a) restored image (b).

Table 1: MSE using both methods.

Training image	Different methods	e_{MSE}
Moon	RBFNN	4.81×10^{-6}
Moon	MLPR	4.13×10^{-8}
Coins	RBFNN	5.06×10^{-6}
Coins	MLPR	3.97×10^{-8}
Saturn	RBFNN	6.62×10^{-6}
Saturn	MLPR	5.65×10^{-9}
Tire	RBFNN	8.04×10^{-6}
Tire	MLPR	7.19×10^{-8}

Acknowledgments

This work was supported by Chongqing CSTC foundation of China (CSTC2009BB2420), Chongqing CMEC foundation of China (KJ100810, KJ100818), and CQUT foundation of China (2007ZD16).

References

[1] S. Chountasis, V. N. Katsikis, and D. Pappas, “Applications of the Moore-Penrose inverse in digital image restoration,” *Mathematical Problems in Engineering*, vol. 2009, Article ID 170724, 12 pages, 2009.

[2] S. Chountasis, V. N. Katsikis, and D. Pappas, “Digital image reconstruction in the spectral domain utilizing the Moore-Penrose inverse,” *Mathematical Problems in Engineering*, vol. 2010, Article ID 750352, 14 pages, 2010.

[3] L. Su, H. Ma, Z. Li, and S. Ju, “Blind image restoration based on constant modulus with averaging and ANFIS,” in *Proceedings of the 4th International Conference on Image and Graphics (ICIG ’07)*, pp. 143–148, Chengdu, China, 2007.

[4] M. Alonso, Jr., A. Barreto, and M. Adjouadi, “Digital image inverse filtering for improving visual acuity for computer users with visual aberrations,” *Inverse Problems in Science and Engineering*, vol. 16, no. 8, pp. 957–966, 2008.

[5] M. Alonso Jr., A. Barreto, J. G. Cremades, J. A. Jacko, and M. Adjouadi, “Image pre-compensation to facilitate computer access for users with refractive errors,” *Behaviour & Information Technology*, vol. 24, no. 3, pp. 161–173, 2005.

[6] L. Bar, N. Sochen, and N. Kiryati, “Semi-blind image restoration via Mumford-Shah regularization,” *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 483–493, 2006.

[7] W. Chen, M. Chen, and J. Zhou, “Adaptively regularized constrained total least-squares image restoration,” *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 588–596, 2000.

[8] S. P. Awate and R. T. Whitaker, “Unsupervised, information-theoretic, adaptive image filtering for image restoration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 364–376, 2006.

[9] X. Gu and L. Gao, “A new method for parameter estimation of edge-preserving regularization in image restoration,” *Journal of Computational and Applied Mathematics*, vol. 225, no. 2, pp. 478–486, 2009.

[10] L.-Z. Lu, M. K. Ng, and F.-R. Lin, “Approximation BFGS methods for nonlinear image restoration,” *Journal of Computational and Applied Mathematics*, vol. 226, no. 1, pp. 84–91, 2009.

[11] F. Šroubek and J. Flusser, “Multichannel blind iterative image restoration,” *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1094–1106, 2003.

[12] M. A. T. Figueiredo and R. D. Nowak, “An EM algorithm for wavelet-based image restoration,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[13] Y. Liao and X. Lin, “Blind image restoration with eigen-face subspace,” *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1766–1772, 2005.

[14] M. M. Chang, A. M. Tekalp, and A. T. Erdem, “Blur identification using the bispectrum,” *IEEE Transactions on Signal Processing*, vol. 39, no. 10, pp. 2323–2325, 1991.

[15] S. J. Reeves and R. M. Mersereau, “Blur identification by the method of generalized cross-validation,” *IEEE Transactions on Image Processing*, vol. 1, no. 3, pp. 301–311, 1992.

[16] R. L. Lagendijk, J. Biemond, and D. E. Boekee, “Identification and restoration of noisy blurred images using the expectation-maximization algorithm,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1180–1191, 1990.

[17] D. Kundur and D. Hatzinakos, “On the use of Lyapunov criteria to analyze the convergence of blind deconvolution algorithms,” *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp. 2918–2925, 1998.

[18] A. K. Katsaggelos and K. T. Lay, “Maximum likelihood blur identification and image restoration using the EM algorithm,” *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 729–733, 1991.

[19] L. Su, F. Li, F. Xu, and Y. Liu, “Defocused image restoration using RBF network and iterative Wiener filter in wavelet domain,” in *Proceedings of International Congress on Image and Signal Processing (CISP '08)*, pp. 311–315, Sanya, China, May 2008.

[20] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, vol. 7 of *Cambridge Nonlinear Science Series*, Cambridge University Press, Cambridge, UK, 1997.

[21] J. Fan and I. Gijbels, *Local Polynomial Modelling and Its Applications*, vol. 66 of *Monographs on Statistics and Applied Probability*, Chapman & Hall, London, UK, 1996.

[22] L. Su, “Prediction of multivariate chaotic time series with local polynomial fitting,” *Computers & Mathematics with Applications*, vol. 59, no. 2, pp. 737–744, 2010.

[23] A. D. Hillery and R. T. Chin, “Iterative Wiener filters for image restoration,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '90)*, vol. 4, pp. 1901–1904, Albuquerque, NM, USA, April 1990.

[24] P. L. Combettes and J.-C. Pesquet, “Wavelet-constrained image restoration,” *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 2, no. 4, pp. 371–389, 2004.

[25] H. Zhou, T. Liu, F. Lin, Y. Pang, and J. Wu, “Image restoration and detail preservation by Bayesian estimation,” *International Journal of Image and Graphics*, vol. 7, no. 2, pp. 497–514, 2007.

[26] P. Guo, H. Li, and M. R. Lyu, “Blind image restoration by combining wavelet transform and RBF neural network,” *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 5, no. 1, pp. 15–26, 2007.

[27] M. Li, Y. Wu, and Q. Zhang, “SAR image segmentation based on mixture context and wavelet hidden-class-label Markov random field,” *Computers & Mathematics with Applications*, vol. 57, no. 6, pp. 961–969, 2009.

[28] P. Li and Y. Fang, “A wavelet interpolation Galerkin method for the simulation of MEMS devices under the effect of squeeze film damping,” *Mathematical Problems in Engineering*, vol. 2010, Article ID 586718, 25 pages, 2010.