



Two-stage image denoising by principal component analysis with local pixel grouping

Lei Zhang^{a,*}, Weisheng Dong^{a,b}, David Zhang^a, Guangming Shi^b

^a Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

^b Key Laboratory of Intelligent Perception and Image Understanding (Chinese Ministry of Education), School of Electronic Engineering, Xidian University, China

ARTICLE INFO

Article history:

Received 5 November 2008

Received in revised form

18 September 2009

Accepted 22 September 2009

Keywords:

Denoising

Principal component analysis (PCA)

Edge preservation

ABSTRACT

This paper presents an efficient image denoising scheme by using principal component analysis (PCA) with local pixel grouping (LPG). For a better preservation of image local structures, a pixel and its nearest neighbors are modeled as a vector variable, whose training samples are selected from the local window by using block matching based LPG. Such an LPG procedure guarantees that only the sample blocks with similar contents are used in the local statistics calculation for PCA transform estimation, so that the image local features can be well preserved after coefficient shrinkage in the PCA domain to remove the noise. The LPG-PCA denoising procedure is iterated one more time to further improve the denoising performance, and the noise level is adaptively adjusted in the second stage. Experimental results on benchmark test images demonstrate that the LPG-PCA method achieves very competitive denoising performance, especially in image fine structure preservation, compared with state-of-the-art denoising algorithms.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Noise will be inevitably introduced in the image acquisition process and denoising is an essential step to improve the image quality. As a primary low-level image processing procedure, noise removal has been extensively studied and many denoising schemes have been proposed, from the earlier smoothing filters and frequency domain denoising methods [25] to the lately developed wavelet [1–10], curvelet [11] and ridgelet [12] based methods, sparse representation [13] and K-SVD [14] methods, shape-adaptive transform [15], bilateral filtering [16,17], non-local mean based methods [18,19] and non-local collaborative filtering [20]. With the rapid development of modern digital imaging devices and their increasingly wide applications in our daily life, there are increasing requirements of new denoising algorithms for higher image quality.

Wavelet transform (WT) [24] has proved to be effective in noise removal [1–10]. It decomposes the input signal into multiple scales, which represent different time-frequency components of the original signal. At each scale, some operations, such as thresholding [1,2] and statistical modeling [3–5], can be performed to suppress noise. Denoising is accomplished by transforming back the processed wavelet coefficients into spatial domain. Late development of WT denoising includes ridgelet

[12] and curvelet [11] methods for line structure preservation. Although WT has demonstrated its efficiency in denoising, it uses a fixed wavelet basis (with dilation and translation) to represent the image. For natural images, however, there is a rich amount of different local structural patterns, which cannot be well represented by using only one fixed wavelet basis. Therefore, WT-based methods can introduce many visual artifacts in the denoising output.

To overcome the problem of WT, in [21] Muresan and Parks proposed a spatially adaptive principal component analysis (PCA) based denoising scheme, which computes the locally fitted basis to transform the image. Elad and Aharon [13,14] proposed sparse redundant representation and K-SVD based denoising algorithm by training a highly over-complete dictionary. Foi et al. [15] applied a shape-adaptive discrete cosine transform (DCT) to the neighborhood, which can achieve very sparse representation of the image and hence lead to effective denoising. All these methods show better denoising performance than the conventional WT-based denoising algorithms.

The recently developed non-local means (NLM) approaches use a very different philosophy from the above methods in noise removal. The idea of NLM can be traced back to [23], where the similar image pixels are averaged according to their intensity distance. Similar ideas were used in the bilateral filtering methods [16,17], where both the spatial and intensity similarities are exploited for pixel averaging. In [18], the NLM denoising framework was well established. Each pixel is estimated as the weighted average of all the pixels in the image, and the weights

* Corresponding author.

E-mail address: cszhang@comp.polyu.edu.hk (L. Zhang).

are determined by the similarity between the pixels. This scheme was improved in [19], where the pair-wise hypothesis testing was used in the NLM estimation. Inspired by the success of NLM methods, recently Dabov et al. [20] proposed a collaborative image denoising scheme by patch matching and sparse 3D transform. They searched for similar blocks in the image by using block matching and grouped those blocks into a 3D cube. A sparse 3D transform was then applied to the cube and noise was suppressed by applying Wiener filtering in the transformed domain. The so-called **BM3D** algorithm achieves remarkable denoising results yet its implementation is a little complex.

In this paper we present an efficient **PCA-based denoising method with local pixel grouping (LPG)**. PCA is a classical de-correlation technique in statistical signal processing and it is pervasively used in pattern recognition and dimensionality reduction, etc. [26]. By transforming the original dataset into PCA domain and preserving only the several most significant principal components, the noise and trivial information can be removed. In [21], a PCA-based scheme was proposed for image denoising by using a moving window to calculate the local statistics, from which the local PCA transformation matrix was estimated. However, this scheme applies PCA directly to the noisy image without data selection and many noise residual and visual artifacts will appear in the denoised outputs.

In the proposed LPG-PCA, we model a pixel and its nearest neighbors as a vector variable. The training samples of this variable are selected by grouping the pixels with similar local spatial structures to the underlying one in the local window. With such an LPG procedure, the local statistics of the variables can be accurately computed so that the image edge structures can be well preserved after shrinkage in the PCA domain for noise removal.

As shown in Fig. 1, the proposed LPG-PCA algorithm has two stages. The first stage yields an initial estimation of the image by removing most of the noise and the second stage will further refine the output of the first stage. The two stages have the same procedures except for the parameter of noise level. Since the noise is significantly reduced in the first stage, the LPG accuracy will be much improved in the second stage so that the final denoising result is visually much better. Compared with WT that uses a fixed basis function to decompose the image, the proposed LPG-PCA method is a spatially adaptive image representation so that it can better characterize the image local structures. Compared with NLM and the BM3D methods, the proposed LPG-PCA method can use a relatively small local window to group the similar pixels for PCA training, yet it yields competitive results with state-of-the-art BM3D algorithm.

The rest of the paper is structured as follows. Section 2 briefly reviews the procedure of PCA. Section 3 presents the LPG-PCA denoising algorithm in detail. Section 4 presents the experimental results and Section 5 concludes the paper.

2. Principal component analysis (PCA)

Denote by $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_m]^T$ an m -component vector variable and denote by

$$\mathbf{X} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^n \\ x_2^1 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ x_m^1 & x_m^2 & \dots & x_m^n \end{bmatrix} \quad (2.1)$$

the sample matrix of \mathbf{x} , where x_i^j , $j=1,2,\dots,n$, are the discrete samples of variable x_i , $i=1,2,\dots,m$. The i th row of sample matrix \mathbf{X} , denoted by

$$X_i = [x_i^1 \ x_i^2 \ \dots \ x_i^n] \quad (2.2)$$

is called the sample vector of x_i . The mean value of X_i is calculated as

$$\mu_i = \frac{1}{n} \sum_{j=1}^n X_i(j) \quad (2.3)$$

and then the sample vector X_i is centralized as

$$\bar{X}_i = X_i - \mu_i = [\bar{x}_i^1 \ \bar{x}_i^2 \ \dots \ \bar{x}_i^n] \quad (2.4)$$

where $\bar{x}_i^j = x_i^j - \mu_i$. Accordingly, the centralized matrix of \mathbf{X} is

$$\bar{\mathbf{X}} = [\bar{X}_1^T \ \bar{X}_2^T \ \dots \ \bar{X}_m^T]^T \quad (2.5)$$

Finally, the co-variance matrix of the centralized dataset is calculated as

$$\mathbf{\Omega} = \frac{1}{n} \bar{\mathbf{X}} \bar{\mathbf{X}}^T \quad (2.6)$$

The goal of PCA is to find an orthonormal transformation matrix \mathbf{P} to de-correlate $\bar{\mathbf{X}}$, i.e. $\bar{\mathbf{Y}} = \mathbf{P} \bar{\mathbf{X}}$ so that the co-variance matrix of $\bar{\mathbf{Y}}$ is diagonal. Since the co-variance matrix $\mathbf{\Omega}$ is symmetrical, it can be written as:

$$\mathbf{\Omega} = \mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^T \quad (2.7)$$

where $\mathbf{\Phi} = [\phi_1 \ \phi_2 \ \dots \ \phi_m]$ is the $m \times m$ orthonormal eigenvector matrix and $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_m\}$ is the diagonal eigenvalue matrix with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$. The terms $\phi_1, \phi_2, \dots, \phi_m$ and $\lambda_1, \lambda_2, \dots, \lambda_m$ are the eigenvectors and eigenvalues of $\mathbf{\Omega}$. By setting

$$\mathbf{P} = \mathbf{\Phi}^T \quad (2.8)$$

$\bar{\mathbf{X}}$ can be decorrelated, i.e. $\bar{\mathbf{Y}} = \mathbf{P} \bar{\mathbf{X}}$ and $\mathbf{\Lambda} = (1/n) \bar{\mathbf{Y}} \bar{\mathbf{Y}}^T$.

An important property of PCA is that it fully de-correlates the original dataset $\bar{\mathbf{X}}$. Generally speaking, the energy of a signal will concentrate on a small subset of the PCA transformed dataset, while the energy of noise will evenly spread over the whole dataset. Therefore, the signal and noise can be better distinguished in the PCA domain.

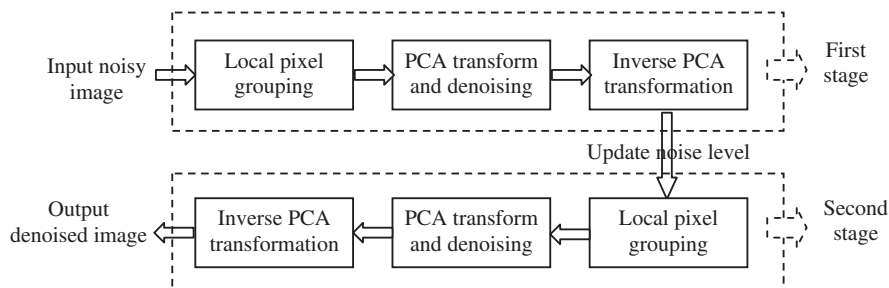


Fig. 1. Flowchart of the proposed two-stage LPG-PCA denoising scheme.

3. LPG-PCA denoising algorithm

3.1. Modeling of spatially adaptive PCA denoising

As in previous literature, we assume that the noise v corrupted in the original image I is white additive with zero mean and standard deviation σ , i.e. $I_v = I + v$, where I_v is the observed noisy image. The image I and noise v are assumed to be uncorrelated. The goal of denoising is to obtain an estimation, denoted by \hat{I} , of I from the observation I_v . The denoised image \hat{I} is expected to be as close to I as possible.

An image pixel is described by two quantities, the spatial location and its intensity, while the image local structure is represented as a set of neighboring pixels at different intensity levels. Since most of the semantic information of an image is conveyed by its edge structures, edge preservation is highly desired in image denoising. To this end, in this paper we model a pixel and its nearest neighbors as a vector variable and perform noise reduction on the vector instead of the single pixel.

Referring to Fig. 2, for an underlying pixel to be denoised, we set a $K \times K$ window centered on it and denote by $\mathbf{x} = [x_1 \dots x_m]^T$, $m = K^2$, the vector containing all the components within the window. Since the observed image is noise corrupted, we denote by

$$\mathbf{x}_v = \mathbf{x} + \mathbf{v} \quad (3.1)$$

the noisy vector of \mathbf{x} , where $\mathbf{x}_v = [x_v^1 \dots x_v^m]^T$, $\mathbf{v} = [v_1 \dots v_m]^T$ and $x_k^v = x_k + v_k$, $k = 1, \dots, m$. To estimate \mathbf{x} from \mathbf{x}_v , we view them as (noiseless and noisy) vector variables so that the statistical methods such as PCA can be used.

In order to remove the noise from \mathbf{x}_v by using PCA, we need a set of training samples of \mathbf{x}_v so that the covariance matrix of \mathbf{x}_v and hence the PCA transformation matrix can be calculated. For this purpose, we use an $L \times L$ ($L > K$) training block centered on \mathbf{x}_v to find the training samples, as shown in Fig. 2. The simplest way is to take the pixels in each possible $K \times K$ block within the $L \times L$ training block as the samples of noisy variable \mathbf{x}_v . In this way, there are totally $(L-K+1)^2$ training samples for each component x_k^v of \mathbf{x}_v . However, there can be very different blocks from the given central $K \times K$ block in the $L \times L$ training window so that taking all the $K \times K$ blocks as the training samples of \mathbf{x}_v will lead to inaccurate estimation of the covariance matrix of \mathbf{x}_v , which subsequently leads to inaccurate estimation of the PCA transformation matrix and finally results in much noise residual (referring to Section 3.4 for an example). Therefore, selecting and grouping the training samples that similar to the central $K \times K$ block is necessary before applying the PCA transform for denoising.

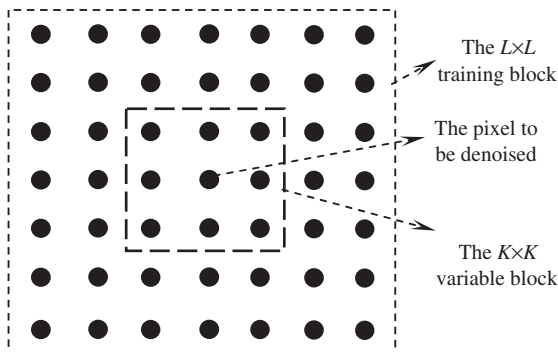


Fig. 2. Illustration of the modeling of LPG-PCA based denoising.

3.2. Local pixel grouping (LPG)

Grouping the training samples similar to the central $K \times K$ block in the $L \times L$ training window is indeed a classification problem and thus different grouping methods, such as block matching, correlation-based matching, K-means clustering, etc., can be employed based on different criteria. Among them, the block matching method may be the simplest yet very efficient one. In this paper, we employ it for LPG.

There are totally $(L-K+1)^2$ possible training blocks of \mathbf{x}_v in the $L \times L$ training window. We denote by \vec{x}_0^v the column sample vector containing the pixels in the central $K \times K$ block and denote by \vec{x}_i^v , $i=1, 2, \dots, (L-K+1)^2-1$, the sample vectors corresponding to the other blocks. Let \vec{x}_0 and \vec{x}_i be the associated noiseless sample vectors of \vec{x}_0^v and \vec{x}_i^v , respectively. It can be easily calculated that

$$e_i = \frac{1}{m} \sum_{k=1}^m \vec{x}_0^v(k) - \vec{x}_i^v(k)^2 \approx \frac{1}{m} \sum_{k=1}^m \vec{x}_0(k) - \vec{x}_i(k)^2 + 2\sigma^2 \quad (3.2)$$

In (3.2) we used the fact that noise v is white and uncorrelated with signal. With (3.2), if

$$e_i < T + 2\sigma^2 \quad (3.3)$$

where T is a preset threshold, then we select \vec{x}_i^v as a sample vector of \mathbf{x}_v .

Suppose we select n sample vectors of \mathbf{x}_v , including the central vector \vec{x}_0^v . For the convenience of expression, we denote these sample vectors as $\vec{x}_0^v, \vec{x}_1^v, \dots, \vec{x}_{n-1}^v$. The noiseless counterparts of these vectors are denoted as $\vec{x}_0, \vec{x}_1, \dots, \vec{x}_{n-1}$, accordingly. The training dataset for \mathbf{x}_v is then formed by

$$\mathbf{X}_v = [\vec{x}_0^v \ \vec{x}_1^v \ \dots \ \vec{x}_{n-1}^v] \quad (3.4)$$

The noiseless counterpart of \mathbf{X}_v is denoted as $\mathbf{X} = [\vec{x}_0 \ \vec{x}_1 \ \dots \ \vec{x}_{n-1}]$.

To guarantee there are enough samples in computing the PCA transformation matrix, n could not be too small. In practice, we will use at least $c \cdot m$ training samples of \mathbf{x}_v in denoising, where constant $c = 8 \sim 10$. That is to say, if $n < c \cdot m$, we will use the best $c \cdot m$ matched samples in PCA training. Usually, the best $c \cdot m$ matched samples are robust to estimate the image local statistics, and this operation makes the algorithm more stable to calculate the PCA transformation matrix.

Now the problem is how to estimate the noiseless dataset \mathbf{X} from the noisy measurement \mathbf{X}_v . Once \mathbf{X} is estimated, the central block and consequently the central underlying pixel can be extracted. Applying such procedures to each pixel and then the whole image I_v can be denoised.

3.3. LPG-PCA based denoising

In the $m \times n$ dataset matrix \mathbf{X}_v , each component x_k^v , $k=1, 2, \dots, m$, of the vector variable \mathbf{x}_v has n samples. Denote by X_k^v the row vector containing the n samples of x_k^v . Then the dataset \mathbf{X}_v can be represented as $\mathbf{X}_v = [X_1^v \dots X_m^v]^T$. Similarly, we have $\mathbf{X} = [X_1 \dots X_m]^T$, where X_k is the row vector containing the n samples of x_k , and $\mathbf{X}_v = \mathbf{X} + \mathbf{V}$, where $\mathbf{V} = [V_1^T \dots V_m^T]^T$ is the dataset of noise variable \mathbf{v} and V_k is the row sample vector of v_k .

Next we centralize dataset \mathbf{X}_v . The mean value of X_k^v is $\mu_k = (1/n) \sum_{i=1}^n X_k^v(i)$, and then X_k^v is centralized by $\bar{X}_k^v = X_k^v - \mu_k$. Since the noise v_k is zero-mean, X_k^v can also be centralized by $\bar{X}_k = X_k - \mu_k$. Then the centralized datasets of \mathbf{X}_v and \mathbf{X} are obtained as $\bar{\mathbf{X}}_v = [\bar{X}_1^v \dots \bar{X}_m^v]^T$ and $\bar{\mathbf{X}} = [\bar{X}_1 \dots \bar{X}_m]^T$, and we have

$$\bar{\mathbf{X}}_v = \bar{\mathbf{X}} + \mathbf{V} \quad (3.5)$$

Refer to Section 2, by computing the covariance matrix of $\bar{\mathbf{X}}$, denoted by $\Omega_{\bar{\mathbf{X}}}$, the PCA transformation matrix $\mathbf{P}_{\bar{\mathbf{X}}}$ can be obtained. However, the available dataset $\bar{\mathbf{X}}_v$ is noise corrupted so that $\Omega_{\bar{\mathbf{X}}}$ cannot be directly computed. With the linear model (3.5), we have

$$\Omega_{\bar{\mathbf{X}}_v} = \frac{1}{n} \bar{\mathbf{X}}_v \bar{\mathbf{X}}_v^T = \frac{1}{n} (\bar{\mathbf{X}} \bar{\mathbf{X}}^T + \bar{\mathbf{X}} \mathbf{V}^T + \mathbf{V} \bar{\mathbf{X}}^T + \mathbf{V} \mathbf{V}^T)$$

Since $\bar{\mathbf{X}}$ and \mathbf{V} are uncorrelated, items $\bar{\mathbf{X}} \mathbf{V}^T$ and $\mathbf{V} \bar{\mathbf{X}}^T$ will be nearly zero matrices and thus:

$$\Omega_{\bar{\mathbf{X}}_v} \approx \frac{1}{n} (\bar{\mathbf{X}} \bar{\mathbf{X}}^T + \mathbf{V} \mathbf{V}^T) = \Omega_{\bar{\mathbf{X}}} + \Omega_v \quad (3.6)$$

where $\Omega_{\bar{\mathbf{X}}} = (1/n) \bar{\mathbf{X}} \bar{\mathbf{X}}^T$ and $\Omega_v = (1/n) \mathbf{V} \mathbf{V}^T$.

The component $\Omega_v(i, j)$ is the correlation between v_i and v_j . Since v_i and v_j are uncorrelated for $i \neq j$, we know that Ω_v is a $m \times m$ diagonal matrix with all the diagonal components being σ^2 . In other words, Ω_v can be written as $\sigma^2 \mathbf{I}$, where \mathbf{I} is the identity matrix. Then it can be readily proved that the PCA transformation matrix $\mathbf{P}_{\bar{\mathbf{X}}}$ associated with $\Omega_{\bar{\mathbf{X}}}$ is the same as the PCA transformation matrix associated with $\Omega_{\bar{\mathbf{X}}_v}$.

As in (2.7), we can decompose $\Omega_{\bar{\mathbf{X}}}$ as

$$\Omega_{\bar{\mathbf{X}}} = \Phi_{\bar{\mathbf{X}}} \Lambda_{\bar{\mathbf{X}}} \Phi_{\bar{\mathbf{X}}}^T \quad (3.7)$$

where $\Phi_{\bar{\mathbf{X}}}$ is the $m \times m$ orthonormal eigenvector matrix and $\Lambda_{\bar{\mathbf{X}}}$ is the diagonal eigenvalue matrix. Since $\Phi_{\bar{\mathbf{X}}}$ is an orthonormal matrix, we can write Ω_v as

$$\Omega_v = (\sigma^2 \mathbf{I}) \Phi_{\bar{\mathbf{X}}} \Phi_{\bar{\mathbf{X}}}^T = \Phi_{\bar{\mathbf{X}}} (\sigma^2 \mathbf{I}) \Phi_{\bar{\mathbf{X}}}^T = \Phi_{\bar{\mathbf{X}}} \Omega_v \Phi_{\bar{\mathbf{X}}}^T$$

Thus we have

$$\begin{aligned} \Omega_{\bar{\mathbf{X}}_v} &= \Omega_{\bar{\mathbf{X}}} + \Omega_v = \Phi_{\bar{\mathbf{X}}} \Lambda_{\bar{\mathbf{X}}} \Phi_{\bar{\mathbf{X}}}^T + \Phi_{\bar{\mathbf{X}}} (\sigma^2 \mathbf{I}) \Phi_{\bar{\mathbf{X}}}^T \\ &= \Phi_{\bar{\mathbf{X}}} (\Lambda_{\bar{\mathbf{X}}} + \sigma^2 \mathbf{I}) \Phi_{\bar{\mathbf{X}}}^T = \Phi_{\bar{\mathbf{X}}} \Lambda_{\bar{\mathbf{X}}_v} \Phi_{\bar{\mathbf{X}}}^T \end{aligned} \quad (3.8)$$

where $\Lambda_{\bar{\mathbf{X}}_v} = \Lambda_{\bar{\mathbf{X}}} + \sigma^2 \mathbf{I}$. Eq. (3.8) implies that $\Omega_{\bar{\mathbf{X}}_v}$ and $\Omega_{\bar{\mathbf{X}}}$ have the same eigenvector matrix $\Phi_{\bar{\mathbf{X}}}$. Thus, in practical implementation we can directly compute $\Phi_{\bar{\mathbf{X}}}$ by decomposing $\Omega_{\bar{\mathbf{X}}_v}$, instead of $\Omega_{\bar{\mathbf{X}}}$, and then the orthonormal PCA transformation matrix for $\bar{\mathbf{X}}$ is set as

$$\mathbf{P}_{\bar{\mathbf{X}}} = \Phi_{\bar{\mathbf{X}}}^T \quad (3.9)$$

Applying $\mathbf{P}_{\bar{\mathbf{X}}}$ to dataset $\bar{\mathbf{X}}_v$, we have

$$\bar{\mathbf{Y}}_v = \mathbf{P}_{\bar{\mathbf{X}}} \bar{\mathbf{X}}_v = \mathbf{P}_{\bar{\mathbf{X}}} \bar{\mathbf{X}} + \mathbf{P}_{\bar{\mathbf{X}}} \mathbf{V} = \bar{\mathbf{Y}} + \mathbf{V}_v$$

where $\bar{\mathbf{Y}} = \mathbf{P}_{\bar{\mathbf{X}}} \bar{\mathbf{X}}$ is the decorrelated dataset for $\bar{\mathbf{X}}$ and $\mathbf{V}_v = \mathbf{P}_{\bar{\mathbf{X}}} \mathbf{V}$ is the transformed noise dataset for \mathbf{V} . Since $\bar{\mathbf{Y}}$ and noise \mathbf{V}_v are uncorrelated, we can easily derive that the covariance matrix of $\bar{\mathbf{Y}}_v$ is

$$\Omega_{\bar{\mathbf{Y}}_v} = \frac{1}{n} \bar{\mathbf{Y}}_v \bar{\mathbf{Y}}_v^T = \Omega_{\bar{\mathbf{Y}}} + \Omega_{v_y} \quad (3.10)$$

where $\Omega_{\bar{\mathbf{Y}}} = \Lambda_{\bar{\mathbf{X}}}$ is the covariance matrix of decorrelated dataset $\bar{\mathbf{Y}}$ and $\Omega_{v_y} = \mathbf{P}_{\bar{\mathbf{X}}} \Omega_v \mathbf{P}_{\bar{\mathbf{X}}}^T$ is the covariance matrix of noise dataset \mathbf{V}_v .

In the PCA transformed domain $\bar{\mathbf{Y}}_v$, most energy of noiseless dataset $\bar{\mathbf{Y}}$ concentrates on the several most important components, while the energy of noise \mathbf{V}_v distributes much more evenly. The noise in $\bar{\mathbf{Y}}_v$ can be suppressed by using the linear minimum square-error estimation (LMMSE) technique. Since $\bar{\mathbf{Y}}_v$ is centralized, the LMMSE of \bar{Y}_k , i.e. the k th row of $\bar{\mathbf{Y}}_v$, is obtained as

$$\hat{\bar{Y}}_k = w_k \cdot \bar{Y}_v \quad (3.11)$$

where the shrinkage coefficient

$$w_k = \Omega_{\bar{\mathbf{Y}}}(k, k) / \Omega_{\bar{\mathbf{Y}}}(k, k) + \Omega_{v_y}(k, k) \quad (3.12)$$

and \bar{Y}_v is the k th row of $\bar{\mathbf{Y}}_v$. In flat zones, $\Omega_{\bar{\mathbf{Y}}}(k, k)$ is much smaller than $\Omega_{v_y}(k, k)$ so that w_k is close to 0. Hence most of the noise will

be suppressed in $\hat{\bar{Y}}_k$ by LMMSE operator $\hat{\bar{Y}}_k = w_k \cdot \bar{Y}_v$. In implementation we first calculate $\Omega_{\bar{\mathbf{Y}}_v}$ from the available noisy dataset $\bar{\mathbf{Y}}_v$ and then estimate $\Omega_{\bar{\mathbf{Y}}}(k, k)$ by $\Omega_{\bar{\mathbf{Y}}}(k, k) = \Omega_{\bar{\mathbf{Y}}_v}(k, k) - \Omega_{v_y}(k, k)$. In flat zones, it is often that $\Omega_{\bar{\mathbf{Y}}_v}(k, k) - \Omega_{v_y}(k, k) \leq 0$, and then we set $\Omega_{\bar{\mathbf{Y}}}(k, k) = 0$. In this case w_k will be exactly 0 and all the noise in \bar{Y}_v will be removed.

Denote by $\hat{\bar{\mathbf{Y}}}$ the matrix of all $\hat{\bar{Y}}_k$. By transforming $\hat{\bar{\mathbf{Y}}}$ back to the time domain, we obtain the denoised result of $\bar{\mathbf{X}}_v$ as

$$\hat{\bar{\mathbf{X}}} = \mathbf{P}_{\bar{\mathbf{X}}}^T \cdot \hat{\bar{\mathbf{Y}}} \quad (3.13)$$

In (3.13), we used the fact that $\mathbf{P}_{\bar{\mathbf{X}}}^{-1} = \mathbf{P}_{\bar{\mathbf{X}}}^T$. Adding the mean values μ_k back to $\hat{\bar{\mathbf{X}}}$ gives the denoised dataset $\hat{\bar{\mathbf{X}}}$. The estimation of the central block $\hat{\bar{x}}_0$, denoted as $\hat{\bar{x}}_0$, can then be extracted from $\hat{\bar{\mathbf{X}}}$ and finally the denoised result of the underlying central pixel can be extracted from $\hat{\bar{x}}_0$. Applying the above procedure to each pixel leads to the full denoised image of I_v .

3.4. Denoising refinement in the second stage

Most of the noise will be removed by using the denoising procedures described in Sections 3.1–3.3. However, there is still much visually unpleasant noise residual in the denoised image. Fig. 3 shows an example. Fig. 3a is the original image *Cameraman*; Fig. 3b is the noisy version of it ($\sigma=20$, PSNR=22.1 dB); Fig. 3c is the denoised image (PSNR=29.8 dB) by using the proposed LPG-PCA method in Sections 3.1–3.3. Although the PSNR is much improved, we can still see much noise residual in the denoising output.

There are mainly two reasons for the noise residual. First, because of the strong noise in the original dataset \mathbf{X}_v , the covariance matrix $\Omega_{\bar{\mathbf{X}}_v}$ is much noise corrupted, which leads to estimation bias of the PCA transformation matrix and hence deteriorates the denoising performance; second, the strong noise in the original dataset will also lead to LPG errors, which consequently results in estimation bias of the covariance matrix $\Omega_{\bar{\mathbf{X}}}$ (or $\Omega_{\bar{\mathbf{X}}_v}$). Therefore, it is necessary to further process the denoising output for a better noise reduction. Since the noise has been much removed in the first round of LPG-PCA denoising, the LPG accuracy and the estimation of $\Omega_{\bar{\mathbf{X}}}$ (or $\Omega_{\bar{\mathbf{X}}_v}$) can be much improved with the denoised image. Thus we can implement the LPG-PCA denoising procedure for the second round to enhance the denoising results.

As shown in Fig. 1, the noise level should be updated in the second stage of LPG-PCA denoising algorithm. Denote by \hat{I} the denoised version of noisy image I_v in the first stage. We can write \hat{I} as $\hat{I} = I + v_s$, where v_s is the residual in the denoised image. We need to estimate the level of v_s , denoted by $\sigma_s = \sqrt{E[v_s^2]}$, and input it to the second stage of LPG-PCA denoising. Here we estimate σ_s based on the difference between \hat{I} and I_v . Let

$$\tilde{I} = I_v - \hat{I} = v - v_s \quad (3.14)$$

We have:

$$\begin{aligned} E[\tilde{I}^2] &= E[v^2] + E[v_s^2] - 2E[v \cdot v_s] \\ &= \sigma^2 + \sigma_s^2 - 2E[v \cdot v_s] \end{aligned} \quad (3.15)$$

We approximately view v_s as the smoothed version of noise v , and it contains mainly the low frequency component of v . Let $\tilde{v} = v - v_s$ be their difference and \tilde{v} contains mainly the high frequency component of v . There is $E[v \cdot v_s] = E[\tilde{v} \cdot v_s] + E[v_s^2]$. In



Fig. 3. (a) Original image *Cameraman*; (b) noisy image (PSNR=22.1 dB); (c) denoised image after the first stage of the proposed method (PSNR=29.8 dB) and (d) denoised image after the second stage of the proposed method (PSNR=30.1 dB). We see that the visual quality is much improved after the second stage refinement.

general, $E[\tilde{v} \cdot v_s]$ is much smaller compared with $E[v_s^2]$. For example, after the first stage denoising of noisy image *Cameraman* ($\sigma=20$), we have $E[v_s^2]=72$ and $E[\tilde{v} \cdot v_s]=17$. For the convenience of development, we remove $E[\tilde{v} \cdot v_s]$ from $E[v \cdot v_s]$, and let

$$E[v \cdot v_s] = E[\tilde{v} \cdot v_s] + E[v_s^2] \approx E[v_s^2] = \sigma_s^2$$

Thus from (3.15) we have

$$\sigma_s^2 \approx \sigma^2 - E[\tilde{I}^2] \quad (3.16)$$

In practice, v_s will include not only the noise residual but also the estimation error of noiseless image I . Therefore, in implementation we let

$$\sigma_s = c_s \cdot \sqrt{\sigma^2 - E[\tilde{I}^2]} \quad (3.17)$$

where $c_s < 1$ is a constant. We experimentally found that setting c_s around 0.35 can lead to satisfying denoising results for most of the testing images. Fig. 3d shows the denoising results (PSNR=30.1 dB) of *Cameraman* after the second stage. Although the PSNR is not improved too much on this image, we can clearly see that the visual quality is much improved by effectively removing the noise residual in the first stage.

3.5. Denoising of color images

There are two approaches to extending the proposed LPG-PCA algorithm to color images. The first approach is to apply separately LPG-PCA to each of the red, green and blue channels. This approach is simple to implement but it ignores the spectral correlation in the color image. The second approach is to form a $K \times K \times 3$ color variable cube with each $K \times K$ variable block

corresponding to the red, green or blue channel. Like in the denoising of grey level image, the color variable cube is stretched to a color variable vector of dimension $3K^2$. Then the training samples of the color variable vector are selected in the local $L \times L \times 3$ window using the LPG procedure. All the other steps are the same as those in the LPG-PCA denoising of grey level images.

Compared with the first approach, the second approach can exploit both the spatial correlation and the spectral correlation in denoising color images. However, there are two main problems. First, the dimensionality of the color variable vector is three times that of the gray level image, and this will increase significantly the computational cost in the PCA denoising process. Second, the high dimensionality of the color variable vector requires much more training samples to be found in the LPG processing. Nonetheless, we may not be able to find enough training samples in the local neighborhood so that the covariance matrix of the color variable vector may not be accurately estimated, and hence the denoising performance can be reduced. With the above consideration, in this paper we choose the first approach for LPG-PCA based color image denoising due to its simplicity and robustness.

4. Experimental results

In the proposed LPG-PCA denoising algorithm, most of the computational cost spends on LPG grouping and PCA transformation, and thus the complexity mainly depends on two parameters: the size K of the variable block and the size L of training block. In LPG grouping, it requires $(2K^2-1) \cdot (L-K+1)^2$ additions, $K^2 \cdot (L-K+1)^2$ multiplications and $(L-K+1)^2$ “less than” logic operations. Suppose in average S training samples are selected, i.e.



Fig. 4. The test images *Lena*, *Cameraman*, *Barbara*, *Peppers*, *House*, *Bloodcell*, *Paint*, *Monarch*, *Tower* (color) and *Parrot* (color).

Table 1

The PSNR (dB) and SSIM results of the denoised images in the two stages by the proposed LPG-PCA method.

Images	<i>Lena</i>	<i>Cameraman</i>	<i>House</i>	<i>Paint</i>	<i>Monarch</i>
First stage					
$\sigma=10$	33.6(0.9218)	33.9(0.9261)	35.4(0.9003)	33.5(0.9280)	34.0(0.9522)
$\sigma=20$	29.5(0.8346)	29.8(0.8320)	31.8(0.8084)	29.3(0.8440)	29.6(0.8859)
$\sigma=30$	27.1(0.7441)	27.3(0.7395)	29.3(0.7225)	26.8(0.7467)	27.0(0.8071)
$\sigma=40$	25.4(0.6597)	25.5(0.6393)	27.3(0.6243)	25.0(0.6590)	25.2(0.7267)
Second stage					
$\sigma=10$	33.7(0.9243)	34.1(0.9356)	35.6(0.9012)	33.6(0.9311)	34.2(0.9594)
$\sigma=20$	29.7(0.8605)	30.1(0.8902)	32.5(0.8471)	29.5(0.8683)	30.0(0.9202)
$\sigma=30$	27.6(0.8066)	27.8(0.8558)	30.4(0.8185)	27.2(0.8088)	27.4(0.8769)
$\sigma=40$	26.0(0.7578)	26.2(0.8211)	28.9(0.7902)	25.6(0.7569)	25.9(0.8378)
Images	<i>Barbara</i>	<i>Peppers</i>	<i>Bloodcell</i>	<i>Tower (color)</i>	<i>Parrot (color)</i>
First stage					
$\sigma=10$	32.5(0.9357)	33.4(0.8909)	34.6(0.9137)	34.7(0.9047)	34.5(0.9198)
$\sigma=20$	28.3(0.8530)	29.9(0.8177)	31.3(0.8587)	30.6(0.7922)	30.6(0.8337)
$\sigma=30$	26.0(0.7663)	27.5(0.7332)	28.6(0.7864)	28.3(0.6772)	28.2(0.7434)
$\sigma=40$	24.2(0.6741)	25.9(0.6447)	26.7(0.7076)	26.6(0.5718)	26.3(0.6564)
Second stage					
$\sigma=10$	32.5(0.9378)	33.3(0.8943)	34.8(0.9173)	34.8(0.9123)	34.6(0.9255)
$\sigma=20$	28.5(0.8716)	30.1(0.8413)	32.0(0.8836)	31.1(0.8522)	31.1(0.8776)
$\sigma=30$	26.2(0.8028)	27.9(0.7973)	29.6(0.8538)	29.1(0.8069)	29.0(0.8415)
$\sigma=40$	24.5(0.7378)	26.7(0.7648)	28.0(0.8239)	27.8(0.7695)	27.5(0.8097)

The value in the parenthesis is the SSIM measure.

the dataset \mathbf{X}_0 is of dimension $K^2 \times S$. Then in the PCA transformation, it requires $K^2 \cdot S + (S^2 - 1) \cdot K^4 + (K^2 - 1) \cdot K^2 \cdot S$ additions, $K^4 \cdot (S + S^2)$ multiplications, and an SVD decomposition of a $K^2 \times K^2$ definite covariance matrix. In this paper, we set $K=5$ and $L=41$ in all the experiments to test the denoising performance. The threshold T in the LPG grouping is set to 25.

In the implementation of LPG-PCA denoising, actually the complete $K \times K$ block centered on the given pixel will be denoised. Therefore, the finally restored value at a pixel can be set as the average of all the estimates obtained by all windows containing the pixel. This strategy was also used in [21]. By our experiments, this can increase about 0.3 dB the noise reduction for most of the test images.

The proposed LPG-PCA algorithm can be viewed as a completion and extension of the PCA-based denoising algorithm in [21]. We compare LPG-PCA with four representative and state-of-the-art denoising algorithms: the wavelet-based denoising methods [8,10]; the sparse representation based K-SVD denoising method [14]; and the recently developed BM3D denoising method [20].¹

The BM3D method is one of the best denoising methods and it has been viewed as a benchmark for denoising algorithm evaluation. The ten test images (size: 256×256) used in the experiments, including eight grey level images and two color images, are shown in Fig. 4. We added Gaussian white noise of different levels ($\sigma=10, 20, 30$ and 40 , respectively,) to the original image and use the five denoising algorithms for noise removal. Due to the limitation of space, in this paper we can only show partial denoising results. The Matlab codes of our algorithm and all the experimental results can be downloaded in the webpage <http://www.comp.polyu.edu.hk/~cslzhang/LPG-PCA-denoising.htm>.

We evaluate and compare the different methods by using two measures: PSNR and SSIM [22]. Although PSNR can measure the intensity difference between two images, it is well-known that it may fail to describe the visual perception quality of the image. On the other hand, how to evaluate the visual quality of an image is a very difficult problem and it is currently an active research topic. The SSIM index proposed in [22] is one of the most commonly used measures for image visual quality assessment. Compared with PSNR, SSIM can better reflect the structure similarity between the target image and the reference image.

¹ We thank the authors of [8,10,14,20] for sharing their programs.

We first verify the improvement of the noise removal in the second stage of the PLG-PCA method. Table 1 lists the PSNR and SSIM measures of the first stage and second stage denoising outputs on the test image set. We can see that the second stage can improve 0.1–1.5 dB the PSNR values for different images under different noise level (σ is from 10 to 40). Although for some images the second stage will not improve much the PSNR measures, the SSIM measures, which can better reflect the image visual quality, can be much improved. For instance, for image *Lena* with noise level $\sigma=30$, the SSIM measure is much increased from 0.7441 to 0.8066 after the second stage denoising, while the PSNR is increased by only 0.5 dB.

We then compare the different methods on denoising. Table 2 list the PSNR and SSIM results by different methods on the 10 test images. Let's first see the PSNR measures by different methods. From Table 2 we see that the algorithm BM3D has the highest PSNR measures. This is because it sufficiently exploits the non-local redundancies in the image. The K-SVD algorithm uses a pre-trained over-complete dictionary in the denoising process and it achieves almost the same PSNR results as those by the proposed LPG-PCA algorithm. The PSNR result of LPG-PCA is higher than the wavelet-based methods [8,10], and the wavelet-based method [10] has the lowest PSNR value.

Table 2

The PSNR (dB) and SSIM results of the denoised images at different noise levels and by different schemes.

Methods	[10]	[8]	[14]	[20]	Proposed
<i>Lena</i>					
$\sigma=10$	33.1(0.9154)	33.2(0.9160)	33.5(0.9203)	33.9(0.9272)	33.7(0.9243)
$\sigma=20$	29.2(0.8455)	29.4(0.8514)	29.7(0.8571)	30.2(0.8699)	29.7(0.8605)
$\sigma=30$	27.2(0.7878)	27.5(0.7964)	27.8(0.8055)	28.3(0.8231)	27.6(0.8066)
$\sigma=40$	25.7(0.7315)	26.0(0.7466)	26.2(0.7504)	27.3(0.7727)	26.0(0.7578)
<i>Cameraman</i>					
$\sigma=10$	33.2(0.9170)	33.7(0.9307)	33.9(0.9334)	34.4(0.9399)	34.1(0.9356)
$\sigma=20$	29.1(0.8449)	29.6(0.8744)	29.9(0.8810)	30.6(0.8962)	30.1(0.8902)
$\sigma=30$	26.8(0.7945)	27.5(0.8307)	27.9(0.8426)	28.5(0.8655)	27.8(0.8558)
$\sigma=40$	25.3(0.7310)	26.0(0.7806)	26.5(0.8048)	27.1(0.8303)	26.2(0.8211)
<i>House</i>					
$\sigma=10$	34.4(0.8791)	34.8(0.8809)	35.5(0.8960)	36.2(0.9143)	35.6(0.9012)
$\sigma=20$	31.3(0.8199)	32.1(0.8374)	32.7(0.8458)	33.3(0.8553)	32.5(0.8471)
$\sigma=30$	29.4(0.7829)	30.2(0.8066)	30.7(0.8137)	31.6(0.8319)	30.4(0.8185)
$\sigma=40$	28.1(0.7409)	28.9(0.7708)	29.1(0.7771)	30.7(0.8065)	28.9(0.7891)
<i>Paint</i>					
$\sigma=10$	33.0(0.9227)	33.5(0.9319)	33.5(0.9293)	33.7(0.9329)	33.6(0.9311)
$\sigma=20$	29.0(0.8513)	29.6(0.8687)	29.6(0.8655)	29.9(0.8731)	29.5(0.8683)
$\sigma=30$	26.9(0.7897)	27.5(0.8110)	27.5(0.8091)	27.7(0.8196)	27.2(0.8088)
$\sigma=40$	25.6(0.7408)	26.0(0.7616)	26.0(0.7599)	26.6(0.7711)	25.6(0.7569)
<i>Monarch</i>					
$\sigma=10$	33.1(0.9442)	33.6(0.9527)	33.5(0.9501)	33.9(0.9577)	34.2(0.9594)
$\sigma=20$	28.8(0.8912)	29.5(0.9076)	29.6(0.9077)	30.1(0.9222)	30.0(0.9202)
$\sigma=30$	26.5(0.8370)	27.1(0.8583)	27.4(0.8663)	28.0(0.8850)	27.4(0.8769)
$\sigma=40$	25.0(0.7916)	25.7(0.8179)	25.9(0.8260)	26.6(0.8462)	25.9(0.8378)
<i>Barbara</i>					
$\sigma=10$	31.6(0.9241)	31.6(0.9246)	32.3(0.9349)	32.7(0.9420)	32.5(0.9378)
$\sigma=20$	27.4(0.8314)	27.2(0.8316)	28.4(0.8646)	28.9(0.8819)	28.5(0.8716)
$\sigma=30$	25.1(0.7472)	25.0(0.7475)	26.3(0.7919)	26.8(0.8165)	26.2(0.8028)
$\sigma=40$	23.5(0.6696)	23.5(0.6718)	24.7(0.7262)	25.0(0.7444)	24.5(0.7378)
<i>Peppers</i>					
$\sigma=10$	33.1(0.8853)	33.3(0.8901)	33.4(0.8920)	33.6(0.8939)	33.3(0.8909)
$\sigma=20$	29.8(0.8272)	30.1(0.8381)	30.3(0.8400)	30.6(0.8496)	30.1(0.8413)
$\sigma=30$	27.8(0.7781)	28.3(0.7968)	28.4(0.7983)	28.8(0.8108)	27.9(0.7973)
$\sigma=40$	26.4(0.7339)	26.9(0.7552)	27.1(0.7657)	27.2(0.7729)	26.7(0.7648)
<i>Blood cell</i>					
$\sigma=10$	34.6(0.9125)	34.5(0.9136)	35.0(0.9183)	35.0(0.9190)	34.8(0.9137)
$\sigma=20$	31.5(0.8706)	31.5(0.8790)	32.3(0.8859)	32.3(0.8874)	32.0(0.8836)
$\sigma=30$	29.2(0.8338)	29.4(0.8473)	29.9(0.8525)	30.2(0.8622)	29.6(0.8538)
$\sigma=40$	27.4(0.7899)	27.8(0.8129)	28.4(0.8227)	28.0(0.8264)	28.0(0.8239)
<i>Tower (color)</i>					
$\sigma=10$	34.2(0.9017)	34.6(0.9099)	34.7(0.9115)	35.0(0.9144)	34.8(0.9123)
$\sigma=20$	30.5(0.8270)	31.1(0.8444)	31.4(0.8533)	31.6(0.8576)	31.1(0.8522)
$\sigma=30$	28.5(0.7711)	29.2(0.7919)	29.3(0.8018)	29.7(0.8135)	29.1(0.8069)
$\sigma=40$	27.3(0.7277)	27.9(0.7505)	27.9(0.7583)	28.3(0.7760)	27.8(0.7695)
<i>Parrot (color)</i>					
$\sigma=10$	34.0(0.9158)	34.1(0.9190)	34.3(0.9215)	34.6(0.9274)	34.6(0.9255)
$\sigma=20$	30.3(0.8523)	30.6(0.8665)	30.8(0.8684)	31.2(0.8832)	31.1(0.8776)
$\sigma=30$	28.2(0.8048)	28.6(0.8269)	28.8(0.8308)	29.3(0.8505)	29.0(0.8415)
$\sigma=40$	26.7(0.7642)	27.2(0.7925)	27.4(0.7994)	27.5(0.8179)	27.5(0.8097)

The value in the parenthesis is the SSIM measurement.



Fig. 5. The denoising results of *Lena* by different schemes. (a) Noiseless *Lena*; denoised images by methods (b) [10]; (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

Let's then focus on the SSIM measure and the visual quality evaluation of these denoising algorithms. From Table 2 we see that BM3D again achieves the highest SSIM measures. Although

the proposed LPG-PCA has almost the same PSNR results as K-SVD, it has higher SSIM measures than K-SVD. Again, the two wavelet-based denoising methods have the lowest SSIM



Fig. 6. The denoising results of *Cameraman* by different schemes. (a) Noiseless *Cameraman*; denoised images by methods (b) [10]; (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

measures. Figs. 5–14 show the cropped and zoom-in denoising results of the ten noisy images (with noise level $\sigma=20$) by different methods. The sub-figure (a) is the original image; sub-

figures (b–f) are the denoised images by the methods in [8,10,14,20] and the proposed LPG-PCA methods, respectively. We see that although BM3D has higher SSIM measures than

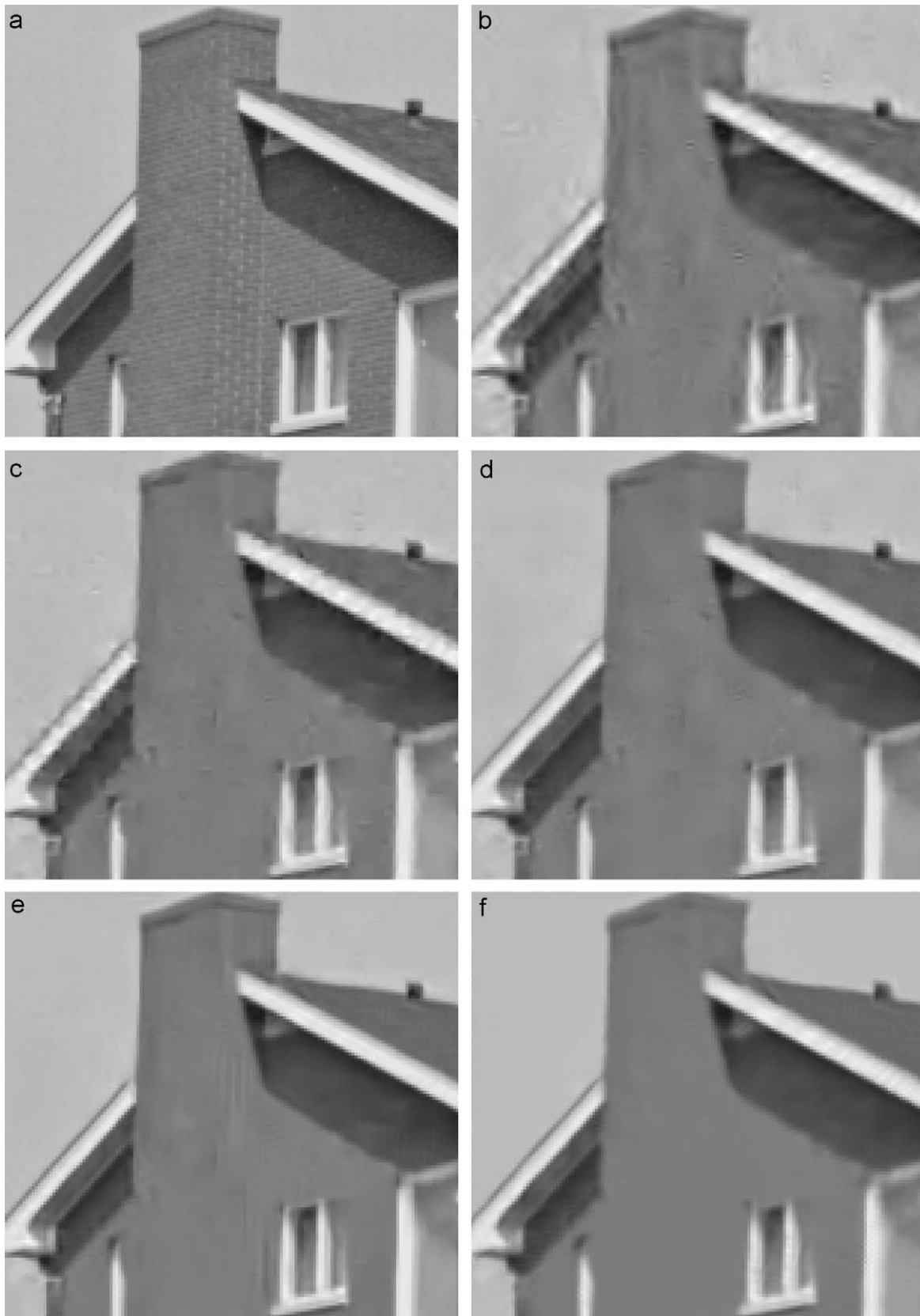


Fig. 7. The denoising results of *House* by different schemes. (a) Noiseless *House*; denoised images by methods (b) [10]; (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

LPG-PCA, their denoised images are very similar in real visual perception, and they have much better visual quality than all the other methods. The K-SVD method generates many visually

disturbing artifacts in the denoised image. The two wavelet-based denoising methods [8,10] have the worst visual quality. This is because in WT, the same wavelet basis function (with dilation



Fig. 8. The denoising results of *Monarch* by different schemes. (a) Noiseless *Monarch*; denoised images by methods (b) [10]; (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

and translation) is used to de-correlate the many different image structures. Often this is not efficient enough to represent the image content so that many denoising errors appear.

The proposed LPG-PCA denoising algorithm uses PCA to adaptively compute the local image decomposition transform so that it can better represent the image local structure. In addition, the

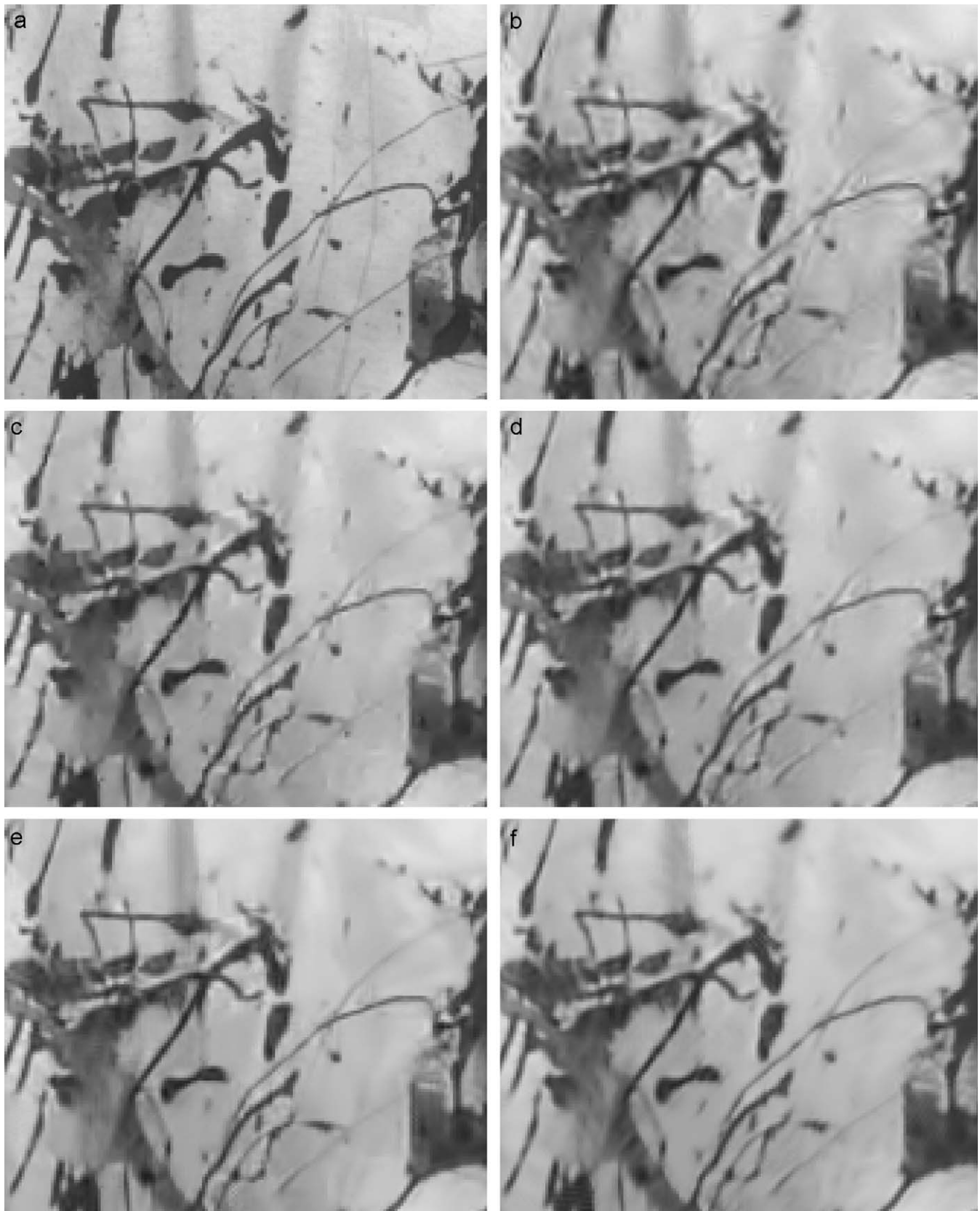


Fig. 9. The denoising results of *Paint* by different schemes. (a) Noiseless *Paint*; denoised images by methods (b) [10], (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

LPG operation is employed to ensure that only the right samples are involved in PCA training. The denoised images by BM3D and LPG-PCA are very similar in terms of visual perception. Both of them can

well preserve the image edges and remove the noise without introducing too many artifacts. Although the PSNR and SSIM measures of LPG-PCA are lower than that of BM3D, LPG-PCA has

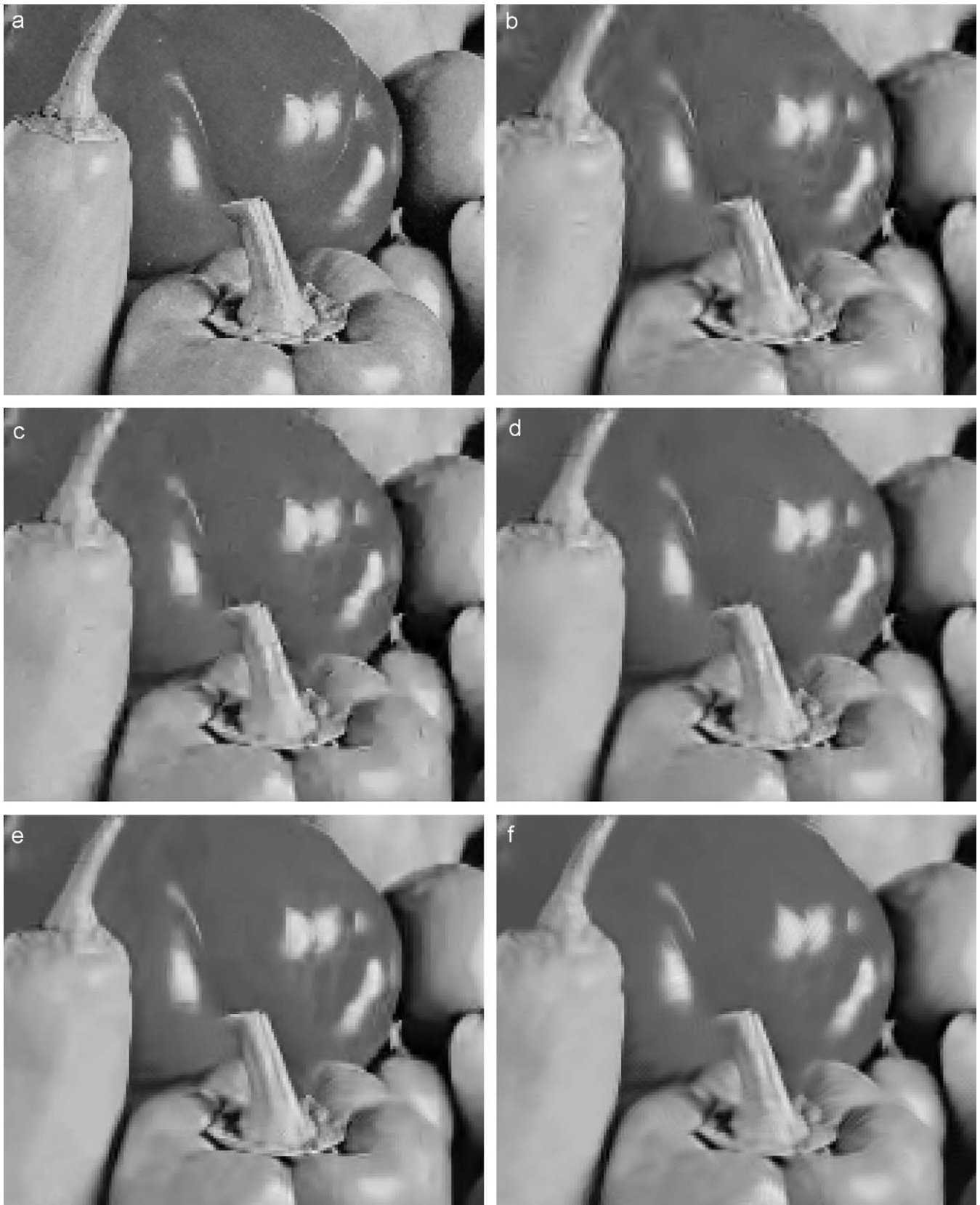


Fig. 10. The denoising results of *Peppers* by different schemes. (a) Noiseless *Peppers*; denoised images by methods (b) [10]; (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

competitive results in **edge preservation**. BM3D works better in preserving large-grain edges and denoising smoothing areas (e.g. the image *House*), where there are a rich amount of

non-local redundancies that could be exploited, while LPG-PCA works better in preserving image fine structures (e.g. the eye area of image *Lena* and the camera boundary in image



Fig. 11. The denoising results of *Barbara* by different schemes. (a) Noiseless *Barbara*; denoised images by methods (b) [10]; (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

Cameraman), where BM3D may generate some artifacts because there are not so many non-local redundancies around those structures.

In summary, as a non-local collaborative denoising technique, BM3D can effectively exploit the non-local redundancy in the image to suppress noise. Therefore, it could have very high

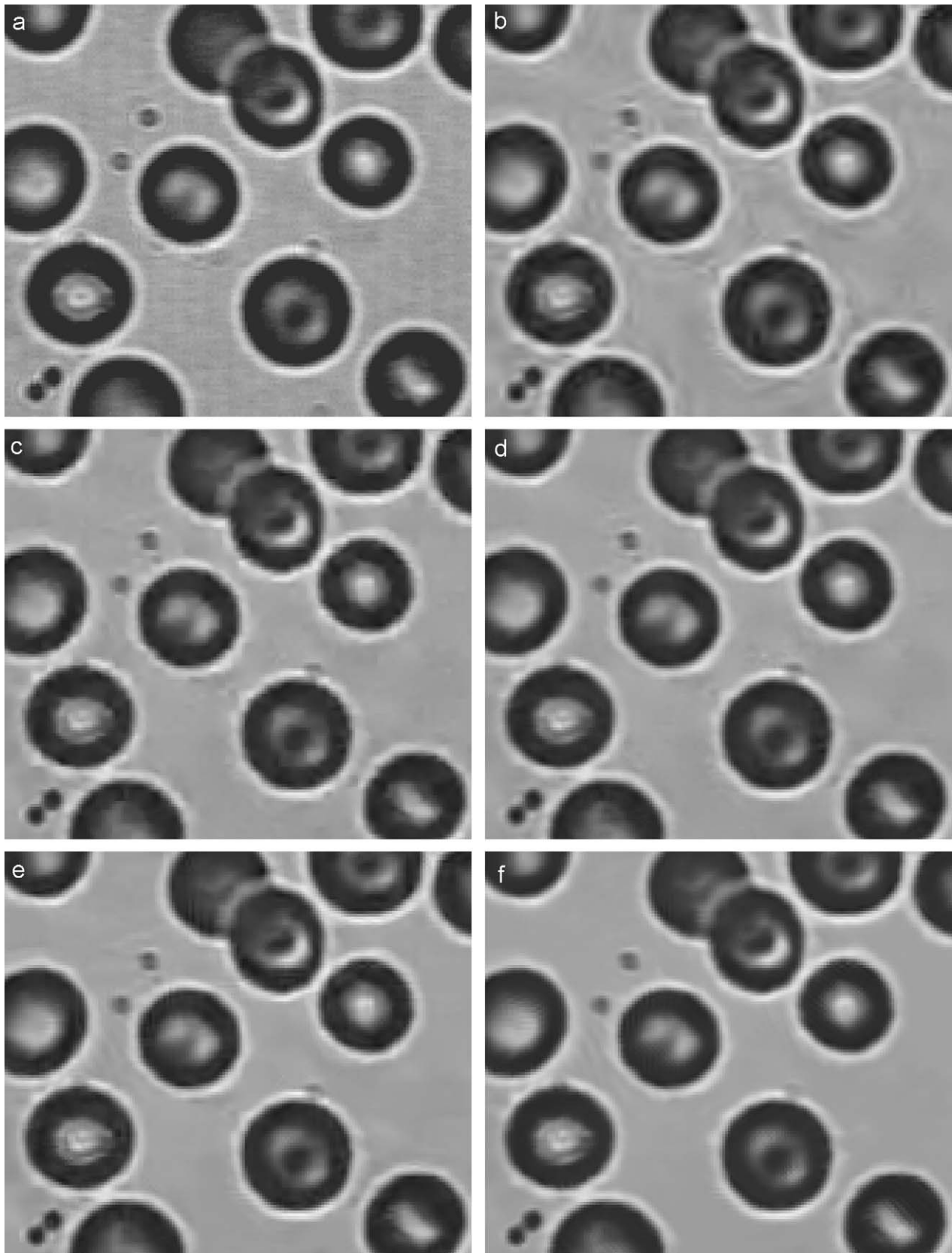


Fig. 12. The denoising results of *Bloodcell* by different schemes. (a) Noiseless *Bloodcell*; denoised images by methods (b) [10]; (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

PSNR and SSIM measures. The large-grain structures and smooth areas could be well reconstructed. However, for fine-grain structures, incorrect non-local information may be introduced

by BM3D for image restoration so that some visible artifacts can be generated in those areas. The proposed LPG-PCA method can be viewed as a semi-non-local scheme. It uses a local window to

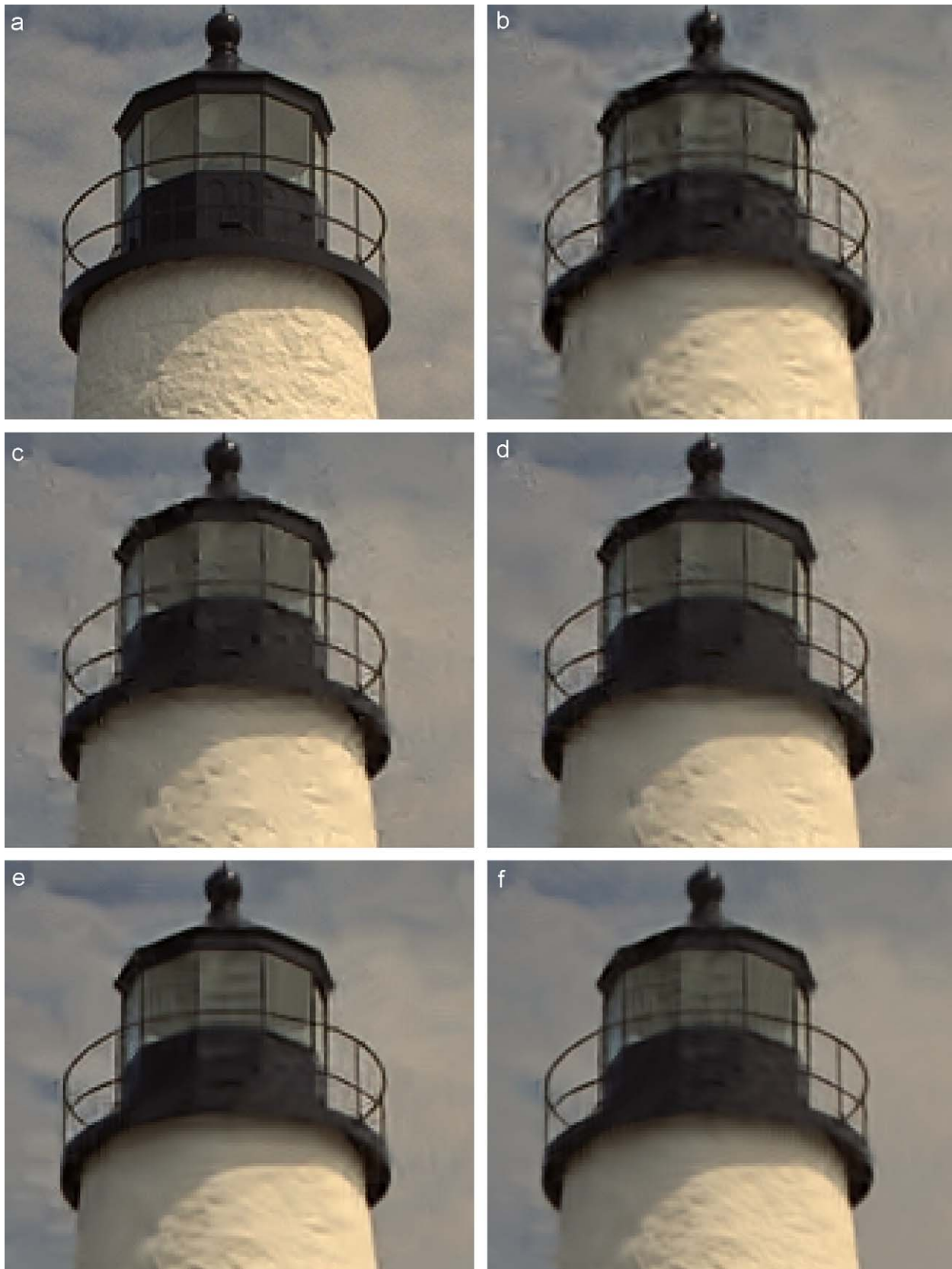


Fig. 13. The denoising results of color *Tower* by different schemes. (a) Noiseless *Tower*; denoised images by methods (b) [10], (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

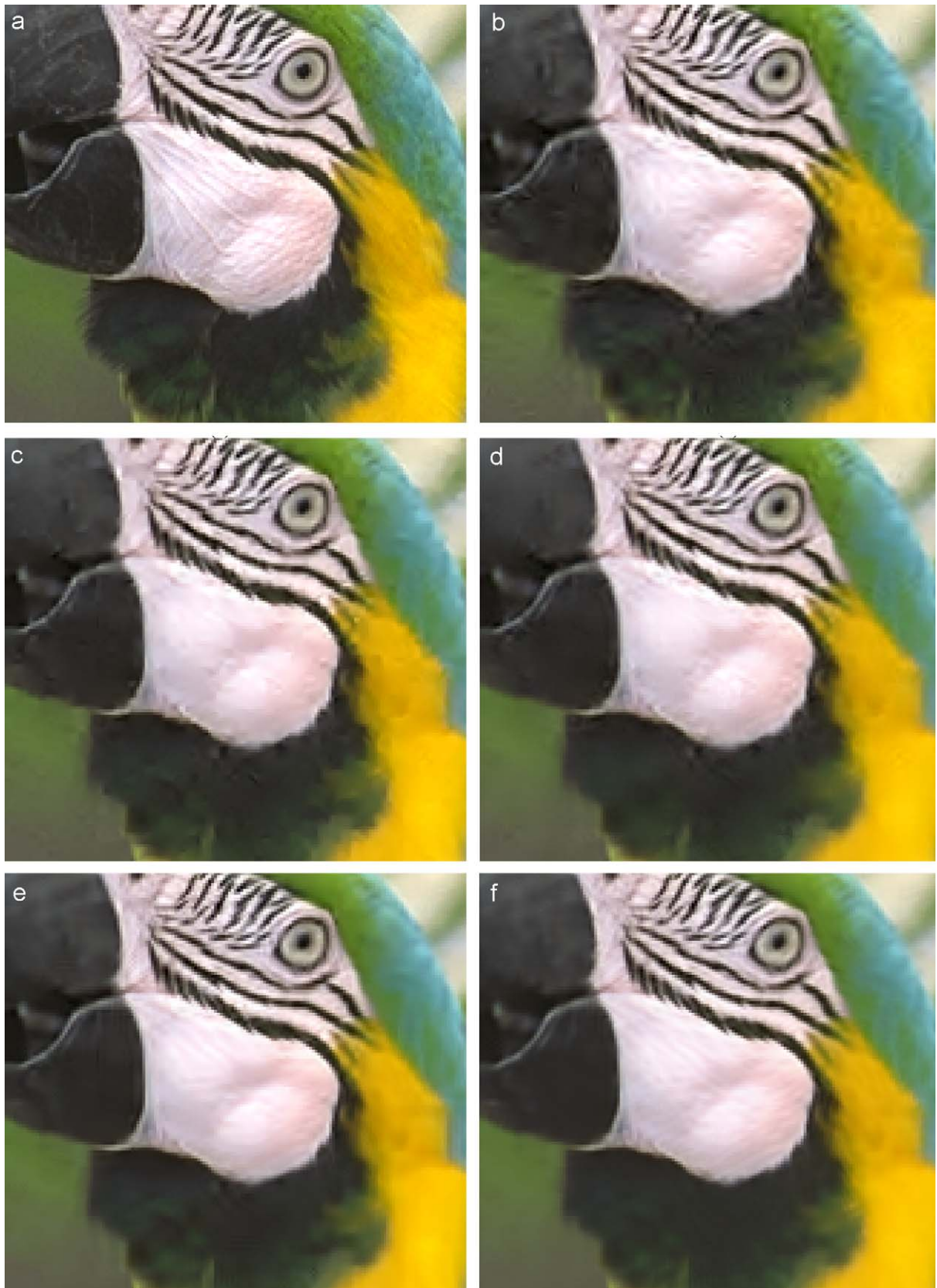


Fig. 14. The denoising results of color *Parrot* by different schemes. (a) Noiseless *Parrot*; denoised images by methods (b) [10], (c) [8]; (d) [14]; (e) [20]; and (f) the proposed LPG-PCA method.

adaptively train the local transform. The vector variable for denoising is defined on a small local block so that LPG-PCA works well in fine-grain edge preservation.

5. Conclusion

This paper proposed a spatially adaptive image denoising scheme by using principal component analysis (PCA). To preserve the local image structures when denoising, we modeled a pixel and its nearest neighbors as a vector variable, and the denoising of the pixel was converted into the estimation of the variable from its noisy observations. The PCA technique was used for such estimation and the PCA transformation matrix was adaptively trained from the local window of the image. However, in a local window there can have very different structures from the underlying one; therefore, a training sample selection procedure is necessary. The block matching based local pixel grouping (LPG) was used for such a purpose and it guarantees that only the similar sample blocks to the given one are used in the PCA transform matrix estimation. The PCA transformation coefficients were then shrunk to remove noise. The above LPG-PCA denoising procedure was iterated one more time to improve the denoising performance. Our experimental results demonstrated that LPG-PCA can effectively preserve the image fine structures while smoothing noise. It presents a competitive denoising solution compared with state-of-the-art denoising algorithms, such as BM3D.

Acknowledgments

This research is supported by the Hong Kong RGC General Research Fund (PolyU 5330/07E), the Hong Kong Innovation and Technology Fund (ITS/081/08), and the National Science Foundation Council (NSFC) of China Key Grant under no. 60634030.

References

- [1] D.L. Donoho, De-noising by soft thresholding, *IEEE Transactions on Information Theory* 41 (1995) 613–627.
- [2] R.R. Coifman, D.L. Donoho, Translation-invariant de-noising, in: A. Antoniadis, G. Oppenheim (Eds.), *Wavelet and Statistics*, Springer, Berlin, Germany, 1995.
- [3] M.K. Mihçak, I. Kozintsev, K. Ramchandran, P. Moulin, Low-complexity image denoising based on statistical modeling of wavelet coefficients, *IEEE Signal Processing Letters* 6 (12) (1999) 300–303.
- [4] S.G. Chang, B. Yu, M. Vetterli, Spatially adaptive wavelet thresholding with context modeling for image denoising, *IEEE Transaction on Image Processing* 9 (9) (2000) 1522–1531.

- [5] A. Pizurica, W. Philips, I. Lamachieu, M. Achery, A joint inter- and intrascale statistical model for Bayesian wavelet based image denoising, *IEEE Transaction on Image Processing* 11 (5) (2002) 545–557.
- [6] L. Zhang, B. Paul, X. Wu, Hybrid inter- and intra wavelet scale image restoration, *Pattern Recognition* 36 (8) (2003) 1737–1746.
- [7] Z. Hou, Adaptive singular value decomposition in wavelet domain for image denoising, *Pattern Recognition* 36 (8) (2003) 1747–1763.
- [8] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, *IEEE Transaction on Image Processing* 12 (11) (2003) 1338–1351.
- [9] L. Zhang, P. Bao, X. Wu, Multiscale LMMSE-based image denoising with optimal wavelet selection, *IEEE Transaction on Circuits and Systems for Video Technology* 15 (4) (2005) 469–481.
- [10] A. Pizurica, W. Philips, Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising, *IEEE Transaction on Image Processing* 15 (3) (2006) 654–665.
- [11] J.L. Starck, E.J. Candes, D.L. Donoho, The curvelet transform for image denoising, *IEEE Transaction on Image Processing* 11 (6) (2002) 670–684.
- [12] G.Y. Chen, B. Kégl, Image denoising with complex ridgelets, *Pattern Recognition* 40 (2) (2007) 578–585.
- [13] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Transaction on Image Processing* 15 (12) (2006) 3736–3745.
- [14] M. Aharon, M. Elad, A.M. Bruckstein, The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation, *IEEE Transaction on Signal Processing* 54 (11) (2006) 4311–4322.
- [15] A. Foi, V. Katkovnik, K. Egiazarian, Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images, *IEEE Transaction on Image Processing* 16 (5) (2007).
- [16] C. Tomasi, R. Manduchi, Bilateral filtering for gray and colour images, in: *Proceedings of the 1998 IEEE International Conference on Computer Vision*, Bombay, India, 1998, pp. 839–846.
- [17] D. Barash, A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation, *IEEE Transaction on Pattern Analysis and Machine Intelligence* 24 (6) (2002) 844–847.
- [18] A. Buades, B. Coll, J.M. Morel, A review of image denoising algorithms, with a new one, *Multiscale Modeling Simulation* 4 (2) (2005) 490–530.
- [19] C. Kervrann, J. Boulanger, Optimal spatial adaptation for patch based image denoising, *IEEE Transaction on Image Processing* 15 (10) (2006) 2866–2878.
- [20] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, *IEEE Transaction on Image Processing* 16 (8) (2007) 2080–2095.
- [21] D.D. Muresan, T.W. Parks, Adaptive principal components and image denoising, in: *Proceedings of the 2003 International Conference on Image Processing*, 14–17 September, vol. 1, 2003, pp. 1101–1104.
- [22] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Transaction on Image Processing* 13 (4) (2004).
- [23] L.P. Yaroslavsky, *Digital Signal Processing—An Introduction*, Springer, Berlin, 1985.
- [24] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, New York, 1998.
- [25] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, second ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.
- [26] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed., Academic Press, New York, 1991.

About the Author—LEI ZHANG received the B.S. degree in 1995 from Shenyang Institute of Aeronautical Engineering, Shenyang, PR China, the M.S. and Ph.D. degrees in Automatic Control Theory and Engineering from Northwestern Polytechnical University, Xi'an, PR China, respectively, in 1998 and 2001. From 2001 to 2002, he was a research associate in the Department of Computing, The Hong Kong Polytechnic University. From January 2003 to January 2006 he worked as a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, McMaster University, Canada. Since January 2006, he has been an Assistant Professor in the Department of Computing, The Hong Kong Polytechnic University. His research interests include image and video processing, biometrics, pattern recognition, multisensor data fusion, optimal estimation theory, etc. Dr. Zhang is an associate editor of *IEEE Transactions on SMC-C*.

About the Author—WEISHENG DONG received the B.S. degree in Electronic Engineering from the Hua Zhong University of Science and Technology, Wu Han, China, in 2004. He is currently pursuing the Ph.D. degree in Circuits and System in Xidian University, Xi'an, China. From September to December 2006, he was a visiting student at Microsoft Research Asia, Beijing, China. He is now a Research Assistant in the Department of Computing, The Hong Kong Polytechnic University. His research interests include image compression, denoising, interpolation, and inverse problems.

About the Author—DAVID ZHANG graduated in computer science from Peking University in 1974 and received his M.Sc. and Ph.D. degrees in Computer Science and Engineering from the Harbin Institute of Technology (HIT), Harbin, PR China, in 1983 and 1985, respectively. He received the second Ph.D. degree in Electrical and Computer Engineering at the University of Waterloo, Waterloo, Canada, in 1994. From 1986 to 1988, he was a Postdoctoral Fellow at Tsinghua University, Beijing, China, and became an Associate Professor at Academia Sinica, Beijing, China. Currently, he is a Professor with the Hong Kong Polytechnic University, Hong Kong. He is Founder and Director of Biometrics Research Centers supported by the Government of the Hong Kong SAR (UGC/CRC). He is also Founder and Editor-in-Chief of the *International Journal of Image and Graphics* (IJIG), Book Editor, The Kluwer International Series on Biometrics, and an Associate Editor of several international journals. His research interests include automated biometrics-based authentication, pattern recognition, biometric technology and systems. As a principal investigator, he has finished many biometrics projects since 1980. So far, he has published over 200 papers and 10 books.

About the Author—GUANGMING SHI received his B.S. degree in Automatic Control, the M.S. degree in Computer Control and the Ph.D. degree in Electronic Information Technology, all from Xidian University in 1985, 1988 and 2002, respectively. He joined the School of Electronic Engineering, Xidian University, in 1988. From 1994 to 1996, as a Research Assistant, he cooperated with the Department of Electronic Engineering at the University of Hong Kong. Since 2003, he has been a Professor in the School of Electronic Engineering at Xidian University, and in 2004 the head of National Instruction Base of Electrician and Electronic (NIBEE). From June to December in 2004, he had studied in the Department of Electronic Engineering at University of Illinois at Urbana-Champaign (UIUC). Presently, he is the Deputy Director of the School of Electronic Engineering, Xidian University, and the academic leader in the subject of Circuits and Systems. His research interests include compressed sensing, theory and design of multivariate filter banks, image denoising, low-bit-rate image/video coding and implementation of algorithms for intelligent signal processing (using DSP&FPGA). Professor Shi has authored or co-authored over 60 research papers.