

DUAL POLYNOMIAL THRESHOLDING FOR TRANSFORM DENOISING IN APPLICATION TO LOCAL PIXEL GROUPING METHOD

Jafet Morales, David Akopian and SosAgaian

Department of Electrical and Computer Engineering, University of Texas at San Antonio,
San Antonio, Texas, USA

ABSTRACT

Thresholding operators have been used successfully for denoising signals, mostly in the wavelet domain. These operators transform a noisy coefficient into a denoised coefficient with a mapping that depends on signal statistics and the value of the noisy coefficient itself. This paper demonstrates that a polynomial threshold mapping can be used for enhanced denoising of Principal Component Analysis (PCA) transform coefficients. In particular, two polynomial threshold operators are used here to map the coefficients obtained with the popular local pixel grouping method (LPG-PCA), which eventually improves the denoising power of LPG-PCA. The method reduces the computational burden of LPG-PCA, by eliminating the need for a second iteration in most cases. Quality metrics and visual assessment show the improvement.

KEYWORDS

Principal Components, Denoising, Shrinkage, Threshold Operators

1. INTRODUCTION

Noise in an image can be due to a variety of reasons and can be introduced during the acquisition, transmission, or processing stages. But for data to be analyzed for a meaningful purpose either by the human eye or a computer, noise must be reduced or eliminated. The problem of noise removal has been studied extensively. These include, but are not limited to mean filters, the nonlinear median filter, simple adaptive filters, the Wiener filter, bandpass, and band-reject filters [1]. Modern algorithms may act on more complicated domains, such as dictionary domains [2] or the wavelet domain [3].

When performing a PCA transformation, the principal component will point in the direction for which the data has the highest variability (i.e. variance) and subsequent components will be ordered from highest to lowest variability, always having the maximum variability possible while being orthogonal to preceding components. In the context of image denoising, this means that information for distinguishing features of an image, such as edges, will be concentrated in the first components, whereas information about the noise will spread more evenly throughout the rest of components [5]. In the simplest scenario, denoising in the PCA domain can be performed by setting to zero some of the coefficients with the lowest variability.

Algorithms that denoise in a PCA domain can differ in many aspects when applied to image denoising. First, there are many ways to decompose the image into signals that will be denoised in their PCA domain. An obvious way to do this is to decompose the image into blocks of the same size. To denoise each one of these signals, several samples of each of those signals must be gathered in order to calculate the PCA transform for each one of them. Therefore, a grouping of signals similar to the target signal needs to be performed. There exists several ways to perform

this grouping. Second, once in the PCA domain, the signals can be denoised in different ways. The simplest way to do this is to set some of the last components in each PCA transformed signal to zero. Finally, the denoised signals must somehow be aggregated into a denoised image. For example, if there is an overlap between the windows to be denoised, then corresponding components in the original domain, or pixel locations, can be averaged. Some algorithms do not denoise signals directly in the PCA domain but rather use the domain to obtain better statistics that can be used in the denoising process. The principal neighborhood dictionaries for non-local means (PND-NLM) approach proposed in [4], for example, makes use of PCA to calculate more accurate weights for pixels to be weight-averaged into a denoised version of the target pixel. Such method is an improvement to the popular NLM denoising approach, which calculates the weights based on metrics obtained from the original domain. For PND-NLM, the results show that denoising accuracy peaks when calculating the weights using only a low number of dimensions compared to the number of dimensions in the original domain.

A very competitive denoising solution is proposed in [5] that uses local pixel grouping and principal component analysis (LPG-PCA). Although originally designed for grayscale and color images, the method has been also used successfully in the denoising of color filter array (CFA) images from single-sensor digital cameras [6]. One of the reasons LPG-PCA is attractive is because it is directly applicable to CFA images, whereas other state-of-the-art algorithms need the CFA images to be demosaicked or interpolated before application, as CFA images have a mosaic structure.

One disadvantage of (LPG-PCA) is its computational burden. The LPG-PCA algorithm, as described in [5], must be applied twice to obtain good denoising results. This is because the PCA transform and the grouping may be biased. The transform and the neighborhood grouping can be biased due to the noise and the lack of enough samples. Besides reducing the amount of noise, the first application is used to setup ground for better grouping and reduced bias of the PCA transform in the second stage. In this paper, an improvement to the first stage is proposed, which reduces the residual noise due to biasing of the transform and biasing of the grouping. The modification drastically improves the results obtained with a single application of LPG-PCA and it also reduces the artifacts obtained in a second application. The proposed improvement can be also extended to [6][7] and applied to other state-of-the-art methods such as the block-matching and 3-D filtering (BM3D) proposed in [8].

The authors of BM3D proposed a denoising method that stacks similar neighborhoods in a 3D image and performs a 3D transform. Their method has obtained good results when compared to other state-of-the-art methods, such as Portilla's Scale mixtures [9] and sparse and redundant representations over trained dictionaries (K-SVD) [10]. Their method is also composed of two stages. The purpose of the first stage is to setup the ground for improved grouping of blocks in the second stage.

Section 2 of this paper presents an overview of the LPG-PCA method for image denoising. In Section 3, polynomial thresholding is discussed. Section 4 describes how polynomial thresholding can be used to denoise the PCA coefficients obtained by LPG-PCA. Denoising experiments and results are presented in Section 5 and Section 6, respectively. Finally, remarks about performance and future outlook are discussed in Section 7.

2. LPG-PCA DENOISING

This section is an overview of the LPG-PCA method proposed by Zhang, Dong, Zhang, and Shi [5]. In [5], a *variable block* of pixels and a wider *training block* centered on the variable block are defined. In this paper the variable block is referred to as the *target block*, and the *training block* as the *training window*. The target block is the one to be denoised and it is selected with a sliding

window that slides across the entire image. Blocks that are similar to the target block are found in the training window in order to calculate the target block's PCA transform. Selection of similar neighborhoods inside the training window is performed based on block matching. Once similar blocks are found, they are used to define and apply PCA transform. In the PCA domain, the linear minimum mean square-error (LMMSE) technique is used to denoise the target block. The goal of LMMSE is to find the parameters for a linear function that can map a noisy coefficient into a clean coefficient. A different linear function is used for each component. After mapping, a "clean" version of the target block is obtained by transforming the target block back to image domain. Finally, redundant pixels from the sliding window are averaged.

The target block is denoted by

$$\mathbf{x} = [x_1 x_2 \dots x_m]^T \quad (2.1)$$

where m is the total number of pixels in the block. The training window is used as the search space for blocks that are similar to the target block. Selected blocks, including the target block are aggregated into a matrix as follows

$$\mathbf{X} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^n \\ x_2^1 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ x_m^1 & x_m^2 & \dots & x_m^n \end{bmatrix} \quad (2.2)$$

Every column in \mathbf{X} represents a sample of the signal to be denoised, and every row represents a pixel position in the blocks. The average signal or block is obtained by averaging the matrix in (2.2) horizontally. The centralized matrix of \mathbf{X} is a matrix from which the average block has been subtracted from all the columns, and it is expressed as

$$\bar{\mathbf{X}} = \begin{bmatrix} \bar{x}_1^1 & \bar{x}_1^2 & \dots & \bar{x}_1^n \\ \bar{x}_2^1 & \bar{x}_2^2 & \dots & \bar{x}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \bar{x}_m^1 & \bar{x}_m^2 & \dots & \bar{x}_m^n \end{bmatrix} \quad (2.3)$$

The PCA transform is derived for the set of columns in (2.3), denoted further as orthonormal transform $\mathbf{P}_{\bar{\mathbf{X}}}$, which can decorrelate the dimensions of $\bar{\mathbf{X}}$. It is found by first computing the covariance matrix of $\bar{\mathbf{X}}$

$$\mathbf{\Omega}_{\bar{\mathbf{X}}} = \frac{1}{n} \bar{\mathbf{X}} \bar{\mathbf{X}}^T \quad (2.4)$$

where n is the number of the block samples for covariance matrix estimation. Since $\mathbf{\Omega}_{\bar{\mathbf{X}}}$ is symmetric, it can be diagonalized with

$$\mathbf{\Omega}_{\bar{\mathbf{X}}} = \mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^T \quad (2.5)$$

where Φ is the orthonormal eigenvector matrix of $\Omega_{\bar{X}}$ and Λ is the diagonal eigenvalue matrix. The PCA transform of \bar{X} is then defined as

$$\mathbf{P}_{\bar{X}} = \Phi^T \quad (2.6)$$

and the PCA linear transformation is applied as

$$\bar{Y} = \mathbf{P}_{\bar{X}} \bar{X} \quad (2.7)$$

In other words, $\mathbf{P}_{\bar{X}}$ is found by diagonalizing the covariance matrix of \bar{X} . While $\Omega_{\bar{X}}$ is not diagonal, the covariance matrix of \bar{Y} , denoted as $\Omega_{\bar{Y}}$ should be diagonal so that the components of \bar{Y} are decorrelated.

When clean samples are not available, the PCA transform can be calculated from a matrix of noisy samples

$$\bar{X}_v = \bar{X} + \mathbf{V} \quad (2.8)$$

It is justified in [5], that the PCA transform of \bar{X}_v approximates that of \bar{X} if the noise is assumed to be white additive with zero mean, and uncorrelated to the image. The noisy dataset in the PCA domain can be written as

$$\bar{Y}_v = \mathbf{P}_{\bar{X}} \bar{X}_v = \mathbf{P}_{\bar{X}} (\bar{X} + \mathbf{V}) = \mathbf{P}_{\bar{X}} \bar{X} + \mathbf{P}_{\bar{X}} \mathbf{V} = \bar{Y} + \mathbf{V}_y \quad (2.9)$$

where \mathbf{V}_y is the noise transformed with $\mathbf{P}_{\bar{X}}$.

There are several ways to remove noise from the target block \bar{X}_v in PCA transform domain. A simple way would be to set some of the last rows in \bar{Y}_v to zero. In the methods described in [6][7], the target block is denoised with the linear minimum mean square error (LMMSE) technique. With LMMSE, each row in \bar{Y} is estimated with

$$\hat{Y}_k = w_k \bar{Y}_{v_k} \quad (2.10)$$

where the subscript k has been used to denote the row number and w_k is the weight for row (i.e., PCA component) k . The optimum weights are calculated as

$$w_k = \frac{\Omega_{\bar{Y}}(k,k)}{\Omega_{\bar{Y}_v}(k,k)} \quad (2.11)$$

The covariance matrix of the clean dataset in the PCA domain $\Omega_{\bar{Y}}$ is not directly available. However, assuming \bar{Y} and \mathbf{V}_y are uncorrelated, we can write $\Omega_{\bar{Y}_v}$ as

$$\begin{aligned}\mathbf{\Omega}_{\bar{y}_v} &= \frac{1}{n} \bar{Y}_v \bar{Y}_v^T = \frac{1}{n} (\bar{Y} \bar{Y}^T + \bar{Y} \mathbf{V}_y^T + \mathbf{V}_y \bar{Y}^T + \mathbf{V}_y \mathbf{V}_y^T) \\ &\approx \frac{1}{n} (\bar{Y} \bar{Y}^T + \mathbf{V}_y \mathbf{V}_y^T) = \mathbf{\Omega}_{\bar{Y}} + \mathbf{\Omega}_{\mathbf{V}_y}\end{aligned}\quad (2.12)$$

Because the noise is uncorrelated, $\mathbf{\Omega}_{\mathbf{V}_y}$ is a diagonal matrix $\mathbf{\Omega}_{\mathbf{V}_y} = \sigma_v \mathbf{I}$, where σ_v is the variance of the noise. And

$$\mathbf{\Omega}_{\bar{Y}} = \mathbf{\Omega}_{\bar{Y}_v} - \mathbf{\Omega}_{\mathbf{V}_y} = \mathbf{\Omega}_{\bar{Y}_v} - \sigma_v \mathbf{I} \quad (2.13)$$

The estimate of the centralized target block is then obtained with

$$\hat{\mathbf{X}} = \mathbf{P}_{\bar{X}}^T \hat{\mathbf{Y}} \quad (2.14)$$

3. POLYNOMIAL THRESHOLD OPERATORS

Thresholding operators have been used extensively in the wavelet domain with the purpose of image denoising. Their effectiveness in the wavelet domain has been shown in [9][10][11][12][13]. A thresholding operator is a function that is used to map a noisy coefficient value into a denoised coefficient value. Two very popular thresholding operators are the hard and soft thresholding operators proposed in [14]. In the case of the hard thresholding operator, coefficients that have a value less than a threshold are set to zero. In the case of the soft threshold operator, coefficients that are lower than the threshold are set to zero and coefficients that exceed the threshold are diminished in magnitude by the value of the threshold. There are many approaches for finding the appropriate value for the threshold. The Universal threshold was proposed in [14]. This threshold value is the expected maximum value of N independent samples from a normal distribution $N(0, \sigma)$ [15], calculated as

$$\lambda_u = \sigma \sqrt{2 \log(N)} \quad (3.1)$$

A class of polynomial threshold operators (PTOs) has been proposed in [11]. These operators have a linear response for coefficients higher than the threshold and an odd polynomial response for coefficients lower than the threshold. Several degrees of freedom allow for the slope and intercept of the linear part to be adjusted as well as the coefficients of the polynomial section of the operator. In the literature, these type of operators have been optimized by using a linear least squares (LLS) optimization approach, which makes use of matrix inversions. An adaptive least mean squares (LMS) optimization technique for denoising filter adaptation was used in [16]. The same adaptive technique has been used in [12] in the context of wavelet thresholding.

Suppose we have a corrupted signal \mathbf{s}_v and we wish to calculate $\hat{\mathbf{S}}$, which is an estimate of the clean signal \mathbf{s} . The polynomial threshold operator is defined as

$$\hat{s}_1 = T(s_{v_i}) = \begin{cases} a_{N-1} s_{v_i} - a_N \text{sign}(s_{v_i}) \lambda |s_{v_i}| & |s_{v_i}| > \lambda \\ \sum_{r=1}^{N-2} a_r s_{v_i}^{2r+1} & |s_{v_i}| \leq \lambda \end{cases} \quad (3.2)$$

where \hat{s}_1 is the estimate of the i^{th} sample of the clean signal, s_{v_i} is the i^{th} sample of the noisy signal, \mathbf{a} is a vector with coefficients for the polynomial and linear parts of the operator, and λ is the threshold.

For denoising of an entire signal, we group the samples in a vector and rewrite equation (3.2) in vector form

$$\hat{\mathbf{s}} = \mathbf{T}(\mathbf{s}_v) = \mathbf{F}(\mathbf{s}_v)\mathbf{a} \quad (3.3)$$

where $\mathbf{F}(\mathbf{s}_v)$ is a matrix for which the i^{th} row is

$$\mathbf{f}_i(\mathbf{s}_v) = \begin{cases} [0 & 0 & \dots & 0 & s_{v_i} & -\text{sgn}(s_{v_i})\lambda] & |s_{v_i}| > \lambda \\ [s_{v_i}s_{v_i}^3 & \dots & s_{v_i}^{2N-3} & 0 & 0] & |s_{v_i}| \leq \lambda \end{cases} \quad (3.4)$$

The optimal coefficients of \mathbf{a} , which can be used to minimize the mean-square-error (MSE) between the clean signal and the noisy signal can be found by solving

$$\mathbf{a}_{\text{opt}} = \arg \min_{\mathbf{a}} \|\mathbf{s} - \mathbf{F}(\mathbf{s}_v)\mathbf{a}\|^2 \quad (3.5)$$

Even though a linear system of equations $\mathbf{Bc} = \mathbf{d}$ where \mathbf{B} is a matrix and \mathbf{c} and \mathbf{d} are vectors, may have no solution, the solution $\hat{\mathbf{c}}$ to $\mathbf{B}^T\mathbf{B}\hat{\mathbf{c}} = \mathbf{B}^T\mathbf{d}$ is the solution that minimizes $\|\mathbf{Bc} - \mathbf{d}\|^2$. Therefore, the linear least squares solution in (3.5) can be obtained with

$$\mathbf{a}_{\text{opt}} = \left(\mathbf{F}^T(\mathbf{s}_v)\mathbf{F}(\mathbf{s}_v) \right)^{-1} \mathbf{F}^T(\mathbf{s}_v)\mathbf{s} \quad (3.6)$$

The coefficients of \mathbf{a}_{opt} can then be used to denoise the signal \mathbf{s}_v . This section is an overview of the LPG-PCA method proposed by Zhang, Dong, Zhang, and Shi [5]. In [5], a *variable block* of pixels and a wider *training block* centered on the variable block are defined. In this paper the variable block is referred to as the *target block*, and the *training block* as the *training window*.

4. DUAL POLYNOMIAL THRESHOLDING OF PCA COEFFICIENTS

In this section we explain how the denoised coefficients obtained with (2.10) can be further denoised by using PTOs. The coefficients to be further denoised are the LMMSE denoised coefficients of the target block in its PCA domain and not the coefficients from other blocks in the training window, since those blocks are only used to calculate the PCA transform and the weights in (2.11). The PTOs are applied to the column of $\hat{\mathbf{Y}}$ that corresponds to the target block (the block to be denoised), before transforming the column back to image domain and assembling it into the denoised version of the image. We will refer to the column of each $\hat{\mathbf{Y}}$ that corresponds to the target block as $\hat{\mathbf{y}}$.

When denoising each sample of $\hat{\mathbf{y}}$, one PTO is used from two that are available for each PCA component. In other words, a total of $m \times 2$ PTOs are used to denoise the entire image, where m is the number of pixels in the target block. Selection of the right PTO from the two that are

available for each PCA component is done with a simple binary classification scheme. If the sample \hat{y}_k of \hat{y} is estimated to be composed mainly of noise, a highly suppressive PTO should be used to denoise it. On the other hand, if the sample is estimated to be mostly clean, a less suppressive PTO is to be used.

The weights calculated in (2.11) can be used to estimate the level of noise on a given sample \hat{y}_k . The diagonals of the covariance matrices from which the weights are calculated can also be written as

$$\mathbf{\Omega}_{\bar{Y}} = \begin{bmatrix} \sigma_{\bar{Y}_1} & c_{21} & \cdots & c_{m1} \\ c_{12} & \sigma_{\bar{Y}_2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ c_{1m} & \cdots & \cdots & \sigma_{\bar{Y}_m} \end{bmatrix} \quad (4.1)$$

$$\mathbf{\Omega}_{\bar{Y}_v} = \begin{bmatrix} \sigma_{\bar{Y}_{v1}} & c_{21} & \cdots & c_{m1} \\ c_{12} & \sigma_{\bar{Y}_{v2}} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ c_{1m} & \cdots & \cdots & \sigma_{\bar{Y}_{vm}} \end{bmatrix} \quad (4.2)$$

where $\sigma_{\bar{Y}_k}$ and $\sigma_{\bar{Y}_{vk}}$ are the variances of the k th row of \bar{Y} and \bar{Y}_v , respectively. These are the variances of each PCA component for the given training window. The weight for the k^{th} PCA component in (2.11) can therefore be rewritten as

$$w_k = \frac{\sigma_{\bar{Y}_k}^2}{\sigma_{\bar{Y}_{vk}}^2} \quad (4.3)$$

A dual polynomial threshold operator (BPTO) for further denoising of \hat{y} can then be defined as

$$T_B(\hat{y}_k) = \begin{cases} T_{k1}(\hat{y}_k)w_k \geq \delta_k \\ T_{k2}(\hat{y}_k)w_k < \delta_k \end{cases} \quad (4.4)$$

Where T_{k1} corresponds to PTO 1 of the k^{th} PCA component of the block and T_{k2} corresponds to PTO 2 of the k^{th} PCA component of the block, and δ_k is a threshold for the k^{th} weight, which can be heuristically adjusted for different type of images.

Training of the operator T_B , which can be used to denoise the entire image consists of finding the optimum polynomial coefficient sets \mathbf{a}_{k1} and \mathbf{a}_{k2} for each pair of PTOs. The total number of coefficients will be $N \times m \times 2$, given that the total number of coefficients per polynomial is N and the number of pixels in each target block is m . The thresholds λ_{k1} and λ_{k2} in (3.2), can be calculated with (3.1), unless a different type of threshold is chosen. To estimate \mathbf{a}_{k1} , \mathbf{a}_{k2} , and the optimum weight threshold δ_k for each component k , we can calculate \mathbf{a}_{k1} and \mathbf{a}_{k2} using (3.5), for discrete values of δ_k , selecting the δ_k , \mathbf{a}_{k1} and \mathbf{a}_{k2} that minimize MSE between the training image and its noisy counterpart.

5. CASE STUDY

Figure 1 shows a noisy brain MRI image, and the image that results after denoising with LPG-PCA. The original image was artificially corrupted with Gaussian noise ($\sigma = 20$).

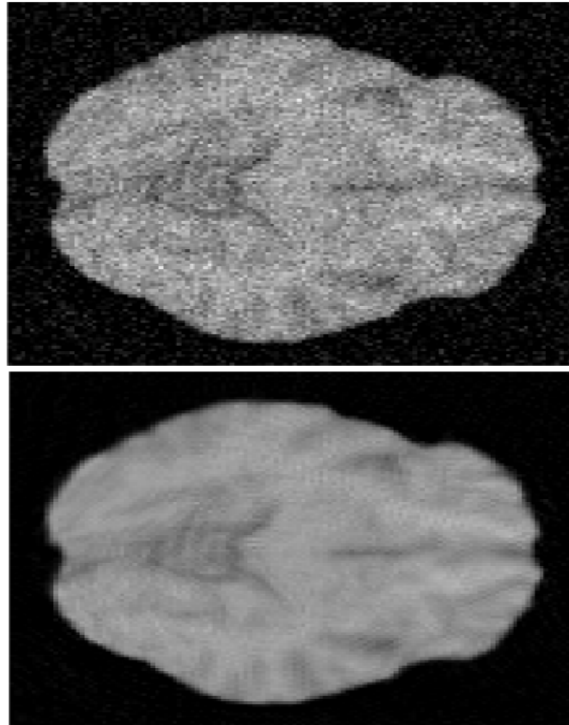


Figure 1. Brain MRI image corrupted with Gaussian noise ($\sigma = 20$) (left); denoised with LPG-PCA (right)

Fig. 2 shows a plot of the clean vs. noisy LPG-PCA coefficients of the corrupted MRI image. Only the second PCA component is shown in Fig. 2. The original LPG-PCA method uses LMMSE to clean the noisy coefficients. The LMMSE denoised coefficients vs. noisy coefficients are plotted in Fig. 3. Ideally, the plots in Fig. 2 and Fig. 3 would be very similar.

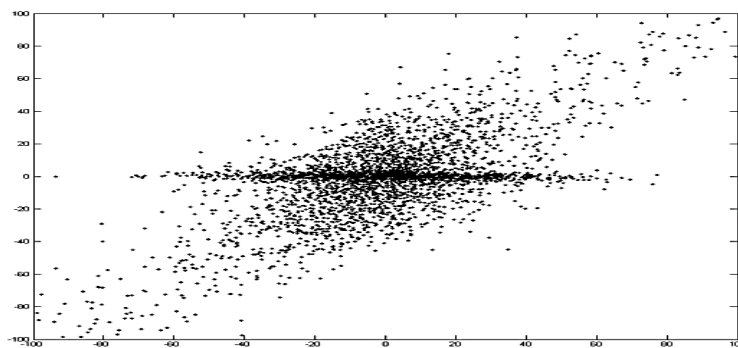


Figure 2. Clean (y-axis) vs. noisy (x-axis) LPG-PCA coefficients for the 2nd component of corrupted brain MRI image

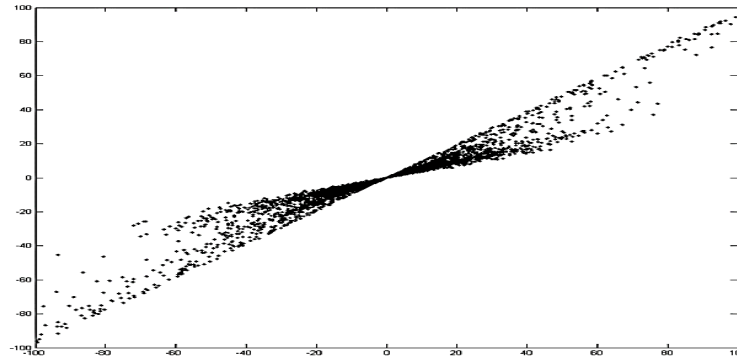


Figure 3. LMMSE denoised (y-axis) vs. noisy (x-axis) LPG-PCA coefficients for the 2nd component of corrupted brain MRI image

It is clear from the plots in Fig. 2-3 that the correction obtained by applying only LMMSE to the LPG-PCA coefficients can be further improved. In particular, the coefficients obtained with LMMSE denoising do not reflect the sparsity of the original signal, shown in Fig. 2, since there are not many denoised coefficients clustered around zero. The proposed adaptive polynomial technique can be used to map the LMMSE denoised coefficients to better denoised coefficients by training the polynomial with similar images that have been corrupted with a similar noise distribution. Two polynomials have been used per component instead of one in order to reduce biasing of nonzero coefficients by the coefficients that are supposed to be zero. Thus, the coefficients for a particular component are classified into two subgroups. Each group is denoised with a separate polynomial. Classification into each group is done by thresholding the weights calculated from the LMMSE denoising procedure as in (4.4). The correct threshold is found by minimizing the MSE of the reconstructed coefficients separately for every component during training.

Fig. 4 shows the clean and noisy images that were used to train the PTOs used to denoise the image in Fig. 1. The original training image is the concatenation of 4 brain MRI images and its noisy counterpart has been corrupted with Gaussian noise ($\sigma = 20$). The training images are twice the size of the target image in every dimension.

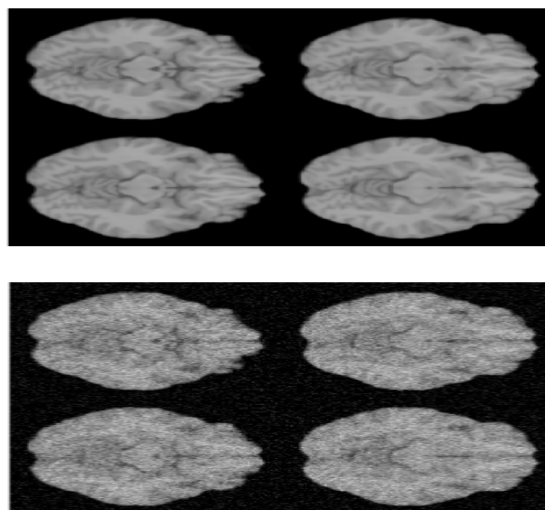


Figure 4. Original training image (left); corrupted with Gaussian noise ($\sigma = 20$) (right)

After training, the PTOs can be used to further denoise the LMMSE denoised LPG-PCA coefficients of a target image. In the case of the image in Fig. 1, the denoised coefficients vs. noisy coefficients for the 2nd component are shown in Fig. 5. The plot shows many more coefficients closer to zero than those in Fig. 3, which were calculated by applying only LMMSE. These are the coefficients for which the weight in (4.3) is much lower, which means they were mostly made of noise. Coefficients with higher weights are almost left untouched.

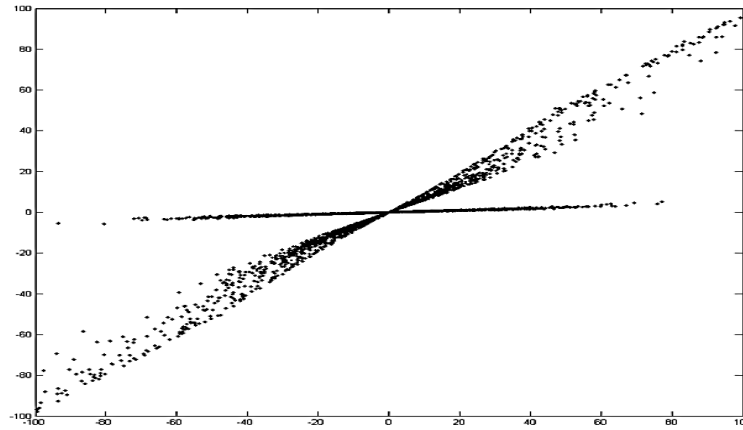


Figure 5. LMMSE Polynomial denoised coefficients (y-axis) vs. noisy coefficients (x-axis) for 2nd component of the MRI image in Fig. 1

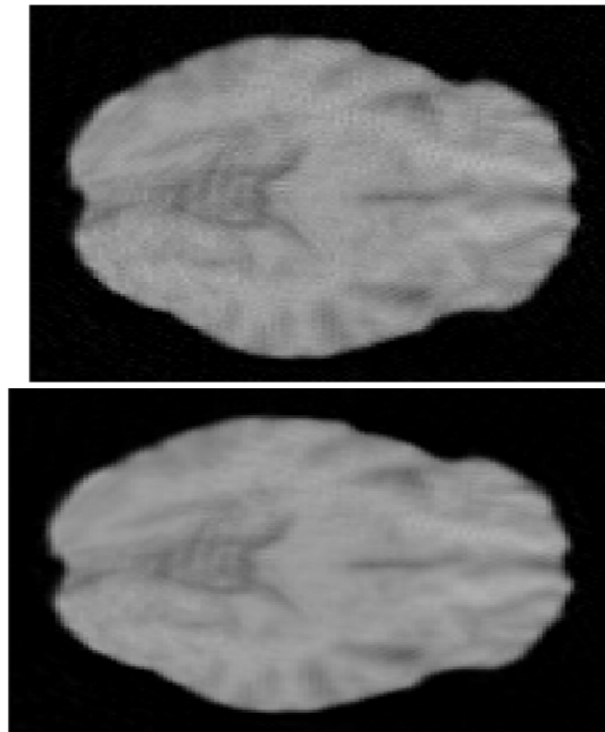


Figure 6. Brain MRI image denoised with LMMSE without polynomial thresholding (left); denoised with LMMSE and polynomial thresholding (right)

The image that results by using LMMSE and the image that results when post-processing with the two polynomials per component are shown together for comparison in Fig. 6.

6. PERFORMANCE ASSESSMENT

The proposed modification to LPG-PCA was tested on the denoising of three datasets of five 128X128 8-bit grayscale images each. The first dataset contains slices from a 3D brain MRI image; the second dataset contains slices from a microscopic 3D femur bone image; and the third dataset contains slices from a 3D neuron cells image. Slices from 3D images were used because of the similarity between the images, meaning that a single trained set of polynomials could be used to denoise each type of image. The datasets were corrupted by Gaussian noise with $\sigma = 10, 20, 30$ and 40 .

The fast version of LPG-PCA was used in the experiments. This version of the algorithm performs a dimensionality reduction so that only 40 percent of the PCA coefficients are used in the grouping phase of the algorithm. All of the coefficients are then used in the PCA decomposition and denoising phases of the algorithm.

In the implementation of polynomial thresholding a total of $N = 4$ coefficients were used for each PTO. The threshold for the polynomial operator was set to be the universal threshold. While training the PTOs for each component, the threshold for the LMMSE weights was varied between 0 and 3 at intervals of .02. At each one of these thresholds the optimum polynomials were calculated. The polynomial threshold operator, the LMMSE weight threshold and their corresponding two polynomials that minimized MSE for each component were chosen as the optimum parameters for denoising each component in a particular set of images. Polynomial thresholding was not used, however, for PCA components in which the number of coefficients above the threshold was less than 20. In these cases the LMMSE denoised coefficients were used as the final denoised coefficients. This was done in order to avoid overfitting of the polynomial to a very few coefficients.

The performance of the modified LPG-PCA algorithm was compared to that of the original LPG-PCA algorithm for one and two iterations. The amount of noise to be reduced by the second iteration is considerably less so that a different set of polynomials were trained to be used for the second iteration. This means that two sets of polynomials were generated during training.

The peak signal to noise ratio (PSNR) results in Table 1 show that a single iteration of the modified LPG-PCA method performs much better than a single iteration of the original LPG-PCA. In fact, a single iteration of the modified LPG-PCA performs better than two iterations of the original method for most of the experiments when using PSNR as a measure of performance. When using the structural similarity index (SSIM) as the performance metric, the results of a single iteration of the modified LPG-PCA are very close to those obtained with two iterations of the original LPG-PCA. In the second iteration of the modified LPG-PCA there is an additional improvement to both metrics in most cases. In both iterations the improvement to the original method is better for high levels of noise. The visual results from Fig. 7-9 show that the improvement by the modification is seen in a reduction of the residual noise, but also in the artifacts created by the original method.

7. CONCLUSION

An enhanced LPG-PCA method has been proposed in this paper. The linear mapping of PCA coefficients by LMMSE in the original implementation of LPG-PCA has been improved by adding an optimal nonlinear mapping stage. The additional mapping of coefficients is done by separating coefficients for every PCA component into two subgroups and using a separate

polynomial for each group. The polynomials are trained to denoise specific types of images. Image degradation metrics and visual appearance show that the proposed stage is an improvement to the original LPG-PCA method. Possible future directions would be to explore whether the method described can improve denoising in other domains and to assess its performance for other types of noise.

Table 1. PSNR and SSIM results for denoising brain MRI, neuron cells, and femur bone images with LPG-PCA and LPG-PCA with polynomial thresholding

Dataset	Noise Level	PSNR				SSIM			
		LPG-PCA One iteration	LPG-PCA Modified One iteration	LPG-PCA Two iterations	LPG-PCA Modified Two iterations	LPG-PCA One iteration	LPG-PCA Modified One iteration	LPG-PCA Two iterations	LPG-PCA Modified Two iterations
Brain MRI	$\sigma = 10$	36.4	37.3	37.4	36.0	0.92	0.94	0.95	0.95
	$\sigma = 20$	31.7	33.2	33.1	33.8	0.78	0.86	0.87	0.89
	$\sigma = 30$	29.0	31.0	30.8	31.6	0.66	0.79	0.80	0.84
	$\sigma = 40$	27.0	29.4	29.3	29.7	0.55	0.73	0.73	0.79
Neurons	$\sigma = 10$	33.5	33.9	33.9	34.0	0.92	0.95	0.95	0.95
	$\sigma = 20$	28.9	29.6	29.5	29.8	0.78	0.88	0.88	0.91
	$\sigma = 30$	26.3	27.2	27.1	27.5	0.65	0.82	0.82	0.86
	$\sigma = 40$	24.4	25.5	25.5	25.9	0.54	0.75	0.74	0.83
Femur	$\sigma = 10$	34.2	34.8	34.6	35.2	0.88	0.90	0.91	0.92
	$\sigma = 20$	29.3	30.0	29.7	30.5	0.68	0.75	0.75	0.80
	$\sigma = 30$	26.6	27.4	27.2	27.9	0.50	0.61	0.61	0.67
	$\sigma = 40$	24.9	26.0	25.8	26.3	0.38	0.51	0.51	0.57

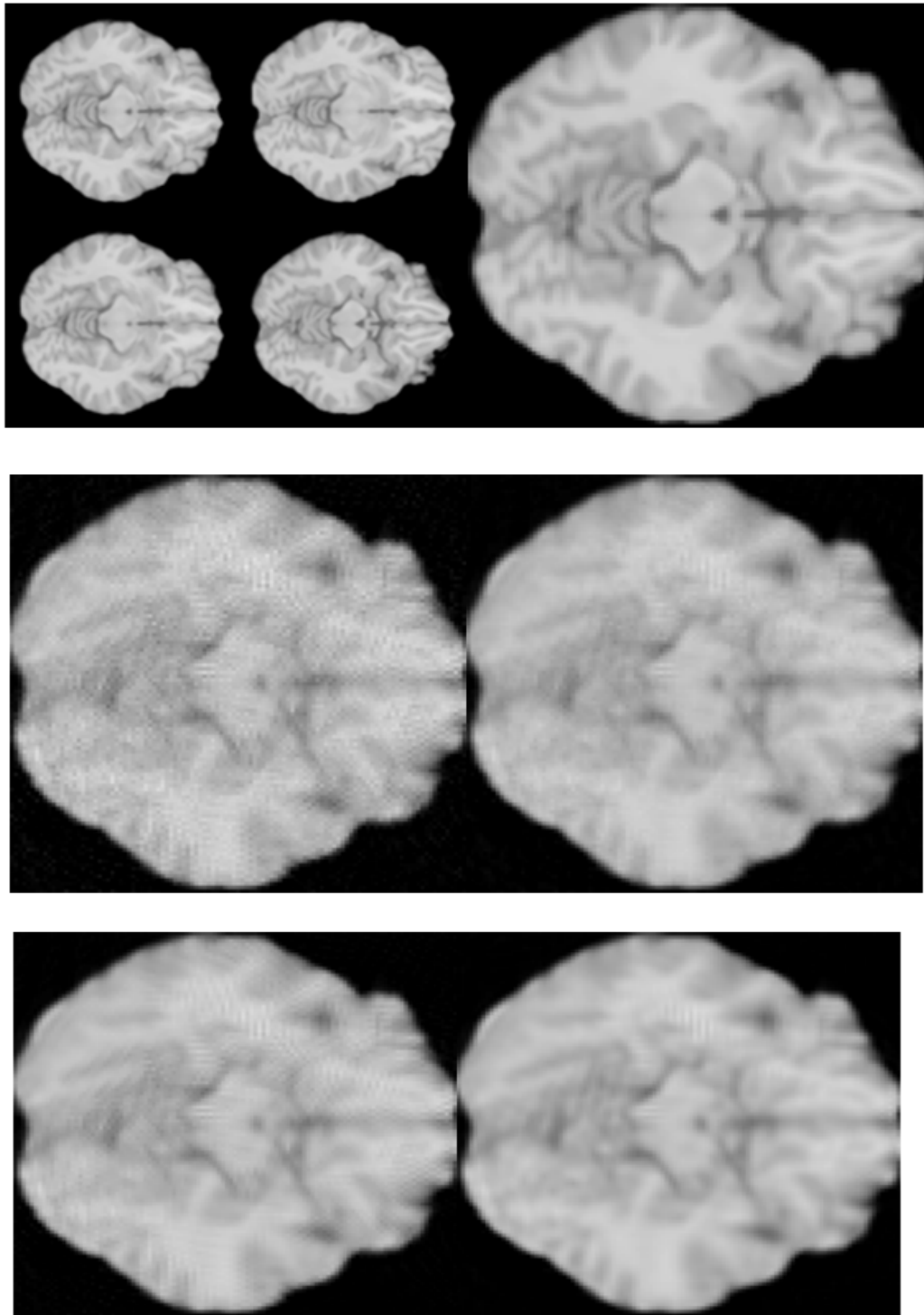


Figure 7. Brain MRI training image (top left); original image for testing (top right); first iteration denoising with LPG-PCA (middle left); first iteration denoising with LPG-PCA and polynomial thresholding (middle right); second iteration denoising with LPG-PCA (bottom left); second iteration denoising with LPG-PCA and polynomial thresholding (bottom right)

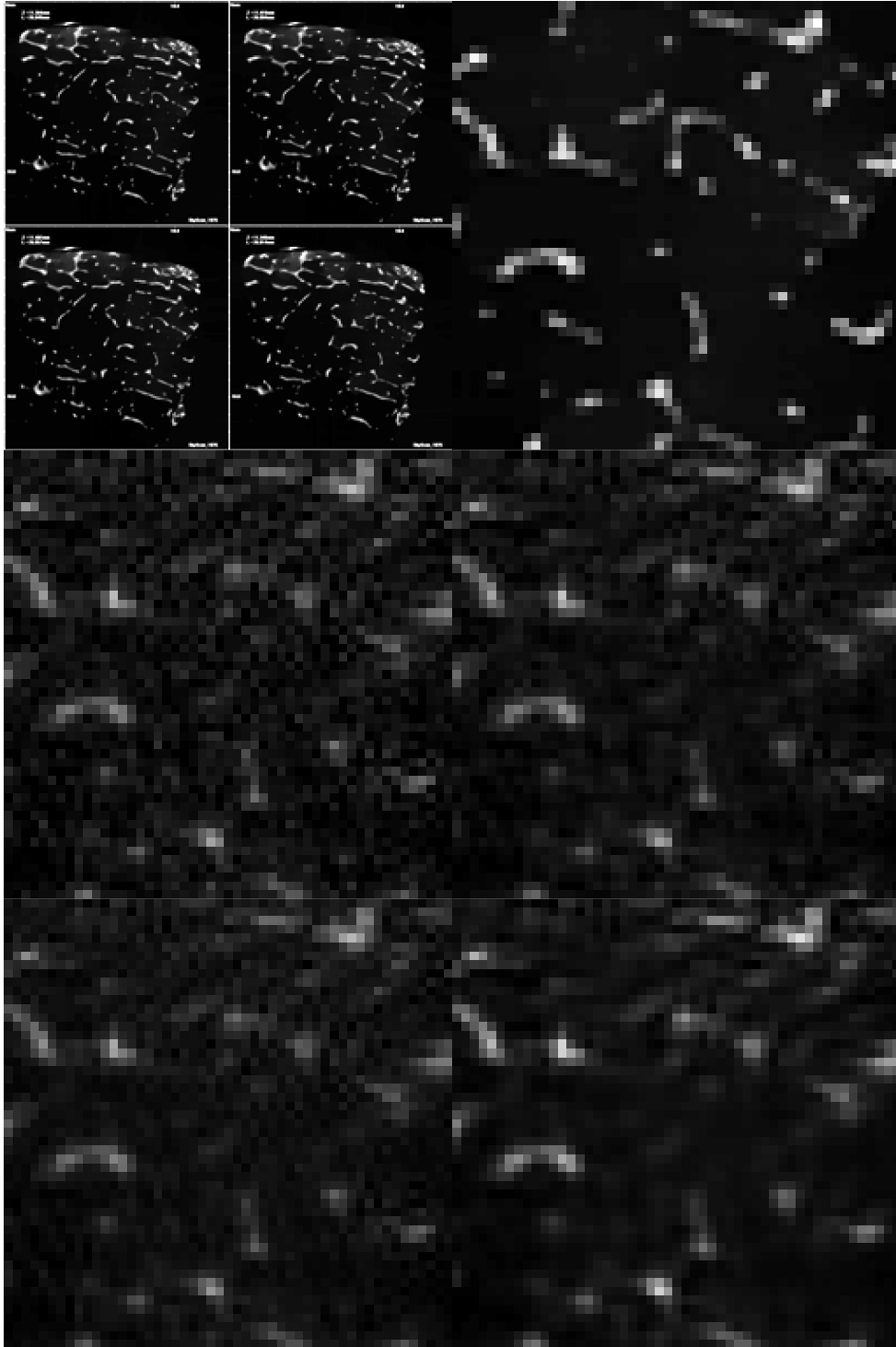


Figure 8. Femur bone training image (top left); original image for testing (top right); first iteration] denoising with LPG-PCA (middle left); first iteration denoising with LPG-PCA and polynomial thresholding (middle right); second iteration denoising with LPG-PCA (bottom left); second iteration denoising with LPG-PCA and polynomial thresholding (bottom right)

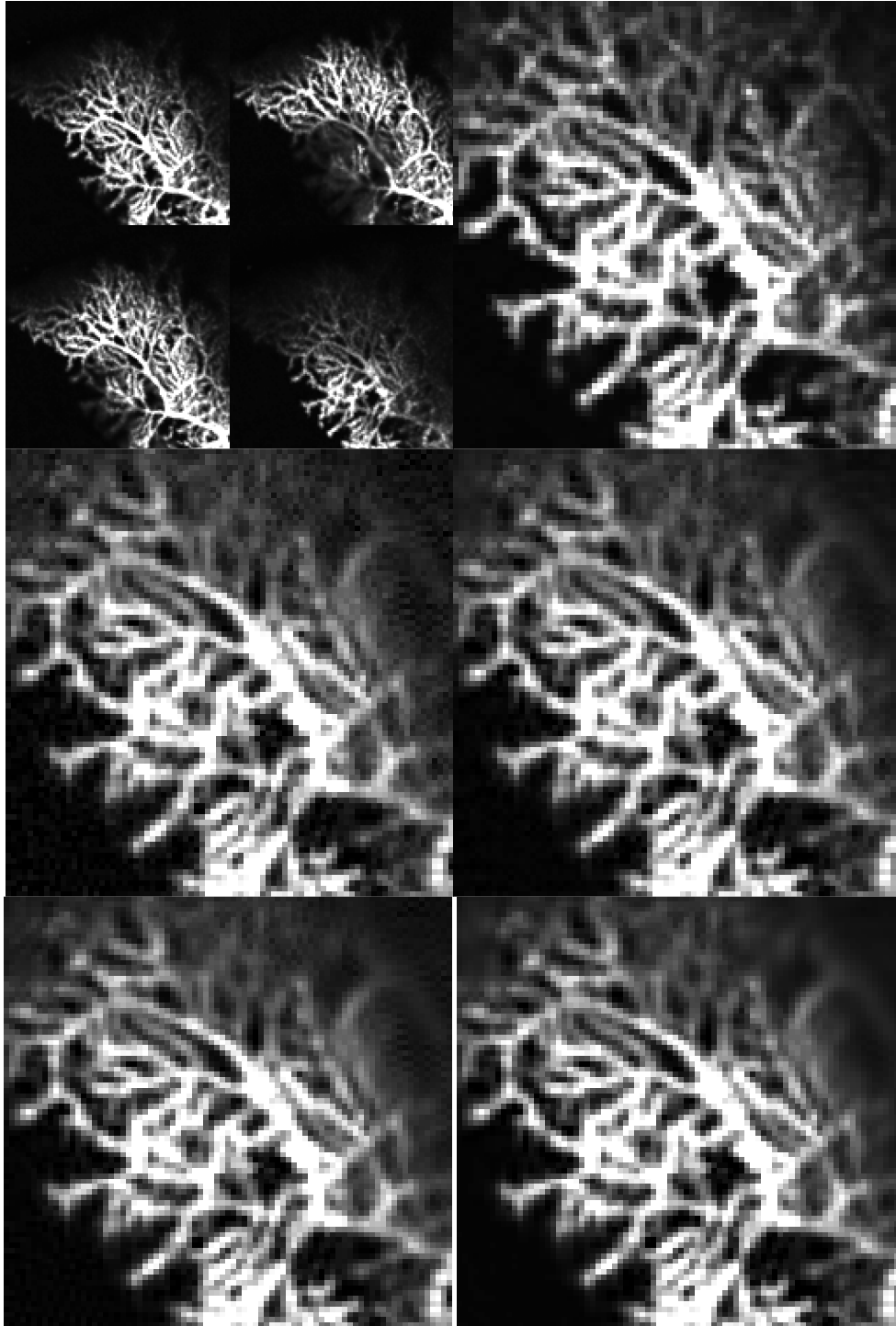


Figure 9. Neuron cells training image (top left); original image for testing (top right); first iteration

denoising with LPG-PCA (middle left); first iteration denoising with LPG-PCA and polynomial thresholding (middle right); second iteration denoising with LPG-PCA (bottom left); second iteration denoising with LPG-PCA and polynomial thresholding (bottom right)

REFERENCES

- [1] R.C. Gonzalez, R.E. Woods, (2002), *Digital Image Processing*, Second Ed., Prentice-Hall, Englewood cliffs, NJ
- [2] M. Elad, M. Aharon, (2006), "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Trans. Image Processing*, Vol. 15, No. 12, pp. 3736-3745
- [3] J.S. Walker, (2008), *A Primer on Wavelets and Their Scientific Applications*, Second Ed., Chapman and Hall/CRC
- [4] T. Tasdizen, (2009), "Principal Neighborhood Dictionaries for Nonlocal Means Image Denoising," *IEEE Transactions on Image Processing*, Vol.18, No.12, pp.2649-2660
- [5] L. Zhang, W. Dong, D. Zhang, G. Shi, (2010), "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, Vol. 43, No. 4, 1531-1549
- [6] L. Zhang, R. Lukac, X. Wu, D. Zhang, (2009), "PCA-Based Spatially Adaptive Denoising of CFA Images for Single-Sensor Digital Cameras," *IEEE Transactions on Image Processing*, Vol.18, No.4, pp.797-812
- [7] D.D. Muresan, T.W. Parks, (2003), "Adaptive principal components and image denoising," *Proceedings of the 2003 International Conference on Image Processing*, Vol.1, pp. I- 101-4
- [8] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, (2007), "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Transactions on Image Processing*, Vol.16, No.8, pp.2080-2095
- [9] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, (2003), "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, Vol.12, No.11, pp. 1338- 1351
- [8] M. Elad and M. Aharon, (2006), "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, Vol. 15, No. 12, pp. 3736–3745
- [9] I.K. Fodor and C. Kamath, (2003), "Denoising through wavelet shrinkage: An empirical study," *J. Electron. Imag.*, Vol. 12, No. 1, pp. 151–160
- [10] E. Hostalkova, O. Vysata, A. Prochazka, (2007), "Multi-Dimensional Biomedical Image De-Noising using Haar Transform," *15th International Conference on Digital Signal Processing, 2007*, pp.175-178
- [11] C.B. Smith, S. Agaian, D. Akopian, (2008), "A Wavelet-Denoising Approach Using Polynomial Threshold Operators," *IEEE Signal Processing Letters*, Vol.15, No., pp.906-909
- [12] S. Sathyanarayana, D. Akopian, S. Agaian (2011), "An adaptive LMS technique for wavelet polynomial threshold denoising," *SPIE Mobile Multimedia/Image Processing, Security, and Applications*, Vol. 8063, pp. 806308:1-10
- [13] M. Chan, S. Sathyanarayana, D. Akopian, S. Agaian, "Application of wavelet polynomial threshold for interpolation and denoising in bioimaging," *SPIE Mobile Multimedia/Image Processing, Security, and Applications*, Vol. 8063, pp. 80630Z1-12
- [14] D.L. Donoho, I.M. Johnstone, (1994), "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, Vol. 81, pp. 425–455
- [15] D.G. Goring, V.I. Nikora, (2003), "Discussion of "Despiking Acoustic Doppler VelocimeterData"", *J. Hydraul. Eng.* Vol. 129, No. 6, pp. 484–487
- [16] D. Akopian and J. Astola, "An optimal nonlinear extension of linear filters based on distributed arithmetic," *IEEE Transactions on Image Processing*, Vol. 14, No.5, May 2005, pp. 616-623.

AUTHORS

Jafet Morales is an electrical engineering Ph.D. candidate at the University of Texas at San Antonio (UTSA) in the area of digital signal processing. He received his M.S. degree in computer engineering from St. Mary's University at San Antonio, Texas, where he wrote a thesis on fundus image quality assessment. He has seven publications in signal processing, computer science, and physics. His current interests include WLAN-based indoor positioning, digital image noise removal techniques, human activity recognition from motiontracking device signals, and automated messaging systems.



David Akopian is a professor at the University of Texas at San Antonio (UTSA). Prior to joining UTSA, he was a Senior Research Engineer and Specialist with Nokia Corporation from 1999 to 2003. From 1993 to 1999 he was a member of the teaching and research staff of Tampere University of Technology, Finland, where he also received a Ph.D. degree in electrical engineering. Dr. Akopian's current research interests include digital signal processing algorithms for communication, positioning and navigation algorithms, and dedicated hardware architectures and platforms for healthcare applications. He authored and co-authored more than 35 patents and 140 publications. He served in organizing and program committees of many IEEE conferences and co-chairs SPIE's annual Multimedia on Mobile Devices conference. His research has been supported by National Science Foundation, National Institutes of Health, U.S. Air Force, U.S. Navy, and Texas Higher Education Coordinating Board.



SosAgaian is the Peter T. Flawn Professor of Electrical and Computer Engineering at the University of Texas, San Antonio and Professor at the University of Texas Health Science Center, San Antonio. Dr. Agaian received the M.S. degree (summa cum laude) in mathematics and mechanics from Yerevan State University, Armenia, the Ph.D. degree in math and physics from the Steklov Institute of Mathematics, Russian Academy of Sciences, and the Doctor of Engineering Sciences degree from the Institute of Control Systems, Russian Academy of Sciences. He has authored more than 500 scientific papers and 7 books, and holds 14 patents. He is a Fellow of SPIE, the American Association for the Advancement of Science (AAAS), and Imaging Sciences and Technology (IS&T). He also serves as a foreign member of the Armenian National Academy. He is the recipient of the MAESTRO Educator of the Year award, sponsored by the Society of Mexican American Engineers. Various technologies he has invented have been adopted by the U.S. government and across multiple disciplines, and have been commercialized by industry. He is an Editorial Board Member of the Journal of Pattern Recognition and Image Analysis and an Associate Editor for several journals, including the Journal of Electronic Imaging (SPIE, IS&T) and IEEE Systems Journal. His research interests are in multimedia processing, imaging systems, information security, artificial intelligence, computer vision, 3D imaging sensors, image fusion, and biomedical and health Informatics.

