Principal Neighborhood Dictionaries for Non-local Means Image Denoising

Tolga Tasdizen *Senior Member, IEEE*

***Abstract*—We present an in-depth analysis of a variation of the Non-local Means (NLM) image denoising algorithm that uses principal component analysis (PCA) to achieve a higher accuracy while reducing computational load. Image neighborhood vectors are first projected onto a lower-dimensional subspace using PCA. The dimensionality of this subspace is chosen automatically using parallel analysis. Consequently, neighborhood similarity weights for denoising are computed using distances in this subspace rather than the full space. The resulting algorithm is referred to as Principal Neighborhood Dictionary (PND) Non-local Means. We investigate PND's accuracy as a function of the dimensionality of the projection subspace and demonstrate that denoising accuracy peaks at a relatively low number of dimensions. The accuracy of NLM and PND are also examined with respect to the choice of image neighborhood and search window sizes. Finally, we present a quantitative and qualitative comparison of PND vs. NLM and another image neighborhood PCA-based state-of-the-art image denoising algorithm.**

***Index Terms*—Principal neighborhood, non-local means, principal component analysis, image denoising, parallel analysis.**

I. Introduction

As computational power increases, data-driven algorithms have begun to gain in popularity in many fields. In image processing, data-driven descriptions of structure are becoming increasingly important. Traditionally, many models used in applications such as denoising and segmentation have been based on the assumption of piecewise smoothness [1], [2], [3]. Unfortunately, this type of model is too simple to capture the textures present in a large percentage of real images. This drawback has limited the performance of such models, and motivated data-driven representations. One data-driven strategy is to use image neighborhoods or patches as a feature vector for representing local structure. Image neighborhoods are rich enough to capture the local structures of real images, but do not impose an explicit model. This representation has been used as a basis for image denoising [4], [5], [6], [7], [8], [9], [10], for texture synthesis [11], [12], and for texture segmentation [13]. For both denoising and segmentation, it has been demonstrated that the accuracy of this strategy is comparable to state-of-the-art methods in general and exceeds them in particular types of images such as those that have significant texture patterns. The drawback is the relatively high computational cost. The image neighborhood feature vector is typically high-dimensional. For instance, it is 49 dimensional if 7×7 neighborhoods are used. Hence, the computation of similarities between feature vectors incurs a large computational cost. One motivation of our work is to reduce the computational complexity of methods that rely on image neighborhood information.

The non-local means(NLM) image denoising algorithm averages pixel intensities using a weighting scheme based on the similarity of image neighborhoods [5]. The use of a lower-dimensional subspace of the space of image neighborhood vectors in conjunction with NLM was first proposed by Azzabou *et al.* [8]. A very similar approach that uses co-variance matrices instead of correlation matrices for subspace computation is given in [9]. In these methods, which we refer to as Principal Neighborhood Dictionary (PND) Non-local Means, the image neighborhood vectors are projected to a lower-dimensional subspace using principal component analysis (PCA). Then, the neighborhood similarity weights for denoising are computed from distances in this subspace resulting in significant computational savings. More importantly, it is also shown that this approach results in increased accuracy over using the full-dimensional ambient space [8], [9]. While it is clear that a global sample of image neighborhoods can not be represented in a reduced dimensionality linear subspace, the increased accuracy can be attributed to the robustness of the similarity criterion to noise. In other words, pairwise distances computed in the subspace defined by the significant eigenvectors of a principal component decomposition are more robust to additive noise than distances computed in the full-dimensional space. Another closely related paper uses singular value decomposition of the image neighborhood vectors for selecting the patches to be used in averaging [10].

One disadvantage of the approach in [9] is the introduction of a new free parameter to the algorithm – the dimensionality of the PCA subspace. Azzabou *et al.* propose to compare eigenvalues of the data correlation matrix to the noise variance to determine the subspace dimensionality [8]. In this paper, we extend our previous work [9] and propose an automatic dimensionality selection criteria using parallel analysis [14] that eliminates this free parameter. Compared to [8], our criteria does not require the estimation of noise variance and is shown to produce a more conservative estimate of dimensionality. We present a detailed analysis of the performance of the method with respect to subspace dimensionality and demonstrate that the dimensionality selection by parallel analysis provides good results. We also provide a detailed discussion of the effect of the smoothing kernel width parameter and search window size selection. Finally, we compare the PND approach to the original NLM algorithm [5] as well as another PCA-based state-of-the-art image denoising algorithm [4].

T. Tasdizen is with the Electrical and Computer Engineering Department and the Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, 84112 USA email: tolga@sci.utah.edu

II. Related Work

A. Image Restoration and Denoising

A comprehensive review of the literature on image restoration and denoising is beyond the scope of this paper. We only give a brief summary of the closest related work. One approach to image restoration arises from the variational formulation and the related partial differential equations (PDEs). The Mumford-Shah [1] and the Rudin-Osher-Fatemi total variation [3] models are the pioneering works in variational formulations in image processing. The PDE based approaches [2], [15], [16] are closely tied to the variational formulations. For instance, Nordstrom shows that the popular Perona and Malik anisotropic diffusion PDE [2] is the first variation of an energy [17]. Traditionally, variational formulations have modeled images as piecewise smooth or piecewise constant functions. While such models are reasonable for some types of images such as certain medical images and photographs of man-made objects, they are too restrictive for other types of images such as textures and natural scenes. To overcome this drawback, variational formulations related to the NLM algorithm that can preserve texture patterns have been proposed [18], [19].

Wavelet denoising methods [20], [21], [22], [23], [24] have also been proven to be very suitable for image restoration. In these approaches, the wavelet transform coefficients are modeled rather than the intensities of the image. By treating wavelet coefficients as random variables and modeling their probability density functions, image restoration can be set up as a problem of estimating the true wavelet coefficients. Patch based approaches can be seen as related to wavelet based approaches when patches are considered as dictionaries [25].

B. Image Neighborhood Based Filtering

Buades *et al.* introduced the NLM image denoising algorithm which averages pixel intensities weighted by the similarity of image neighborhoods [5]. Image neighborhoods are typically defined as 5×5, 7×7 or 9×9 square patches of pixels which can be seen as 25, 49 or 81 dimensional feature vectors, respectively. Then, the similarity of any two image neighborhoods is computed using an isotropic Gaussian kernel in this high-dimensional space. Finally, intensities of pixels in a search-window centered around each pixel in the image are averaged using these neighborhood similarities as the weighting function. More recently, Kervrann and Boulanger [6] have introduced an adaptive search-window approach which attempts to minimize the L2-risk with respect to the size of the search-window by analyzing the bias and variance of the estimator. Kervrann and Boulanger also show that their method is comparable in accuracy to state-of-the-art image denoising methods based on wavelets [24] and Markov Random Field models over neighborhoods [26]. Their method, as well as the standard NLM algorithm, is also shown to outperform classical methods such as total variation regularization [3], bilateral filtering [27] and Wiener filtering. Awate and Whitaker [7] introduced a statistical interpretation to the neighborhood-weighted averaging methods. Their approach is based on treating image neighborhoods as a random vector, computing the probability density function with non-parametric density estimation and formulating image denoising as an iterative entropy reduction. Dabov *et al.* use the block-matching technique, traditionally used in video processing, to stack similar two-dimensional image neighborhoods in to a three-dimensional array [28]. A decorrelating unitary transform is applied to the three-dimensional array to produce a sparse representation. Then, denoising is achieved by applying a threshold to these transform coefficients.

Mahmoudi and Sapiro have proposed a method to improve the computational efficiency of the NLM algorithm [29]. Their patch selection method removes unrelated neighborhoods from the search-window using responses to a small set of predetermined filters such as local averages of gray value and gradients. Unlike [29] the lower-dimensional vectors computed in [8], [9], [10] are data-driven. Additionally, in [8], [9] the lower-dimensional vectors are used for distance computation rather than patch selection.

Principal component analysis of neighborhoods have previously been used for various image processing tasks. Ke and Sukthankar [30] use principal components of image gradient neighborhoods as a descriptor in conjunction with SIFT feature points [31]. PCA of image neighborhoods was also used for denoising [4]. However, in that work, PCA is computed for local collections of image neighborhood samples and denoising is achieved by direct modification of the projection coefficients. In this paper and [8], [9], PCA is computed once, globally rather than locally. This results in a computationally more efficient algorithm. Furthermore, a non-local means averaging scheme is used rather than direct modification of projection coefficients. We present quantitative and qualitative comparisons to this method in Section IV-B. Finally, Elad and Aharon learn a sparse and redundant basis of image neighborhoods, i.e. the sparseland image patch model, for denoising images [25].

C. Parallel Analysis for Dimensionality Selection

There are various methods proposed in the literature for determining the number of components to retain in data analysis [32]. Parallel Analysis, originally proposed by Horn [14], is one of the most successful methods for determining the number of true principal components [33], [34]. Improvements to the original parallel analysis method have also been proposed. For instance, Glorfeld uses Monte-Carlo simulations which do not rely on the normal distribution assumption of the original method and is shown to generate more accurate estimates of the number of components [35]. Several researchers have noted the shortcomings of the parallel analysis method in data with oblique structure and proposed modifications [36], [37]. More recently, parallel analysis has been proposed as a way to determine the number of modes in shape analysis [38].

III. Methods

A. The non-local means algorithm

Starting from a discrete image u, a noisy observation of u at pixel i is defined as $v(i) = u(i) + n(i)$. Let $\mathcal{N}_i$ denote a $r \times r$ square neighborhood centered around pixel i. Also, let $\mathbf{y}(i)$

denote the vector whose elements are the gray level values of v at pixels in $\mathcal{N}_i$. Finally, $\mathcal{S}_i$ is a square search-window centered around pixel i. Then, the NLM algorithm [5] defines an estimator for $u(i)$ as

$$\hat{u}_{NL}(i) = \sum_{j\in\mathcal{S}_i} \frac{1}{Z(i)} e^{-\frac{\|\mathbf{y}(i)-\mathbf{y}(j)\|^2}{h^2}} v(j), \tag{1}$$

where $Z(i) = \sum_{j\in\mathcal{S}_i} e^{-\|\mathbf{y}(i)-\mathbf{y}(j)\|^2/h^2}$ is a normalizing term. The smoothing kernel width parameter h controls the extent of averaging. For true non-local means, the search window $\mathcal{S}_i$ needs to be the entire image for all i, which would give rise to global weighted averaging. However, for computational feasibility, $\mathcal{S}_i$ has traditionally been limited to a square window of modest size centered around pixel i. This is the *limited-range* implementation of the NLM algorithm as proposed in the pioneering work by [5]. For instance, a 21×21 window is used in [5] whereas a 7×7 window is used in [8]. We investigate the search window size's effect in Section IV-C.

The success of the NLM algorithm is attributed to the redundancy that is available in natural images. Constant intensity regions present no problem as there are a very large number of copies of similar neighborhoods in such areas of the image. Edges and other one-dimensional structures also have a relatively large number of copies of similar neighborhoods located along the structure of interest. The hardest case is that of intensity configurations that occur in textured regions. Buades *et al.* show that even in such cases, one can find similar neighborhoods if the search-window $\mathcal{S}$ is sufficiently large [5].

B. Principal Neighborhood Dictionary Non-local Means

In [8], [9], the distances $\| \ \mathbf{y}(i) - \mathbf{y}(j) \ \|^2$ in (1) are replaced by distances computed from projections of $\mathbf{y}$ onto a lower-dimensional subspace determined by PCA. In the rest of this paper we will refer to this method as the PND Non-local Means algorithm. Let Ω denote the entire set of pixels in the image. Also, let Ψ be a randomly chosen subset of Ω. Treating $\mathbf{y}(i)$ as observations drawn from a multivariate random process, we can estimate their covariance matrix as

$$\mathbf{C_y} = \frac{1}{|\Psi|} \sum_{i\in\Psi} \left(\mathbf{y}(i) - \bar{\mathbf{y}}\right)\left(\mathbf{y}(i) - \bar{\mathbf{y}}\right)^T, \tag{2}$$

where $\bar{\mathbf{y}} = \frac{1}{|\Psi|}\sum_{i\in\Psi}\mathbf{y}(i)$ is the sample mean and $|\Psi|$ is the number of elements in the set Ψ. A small subset $\Psi \subseteq \Omega$ is typically sufficient to accurately estimate the covariance matrix and results in computational savings. The dimensionality of a $r \times r$ neighborhood vector is r^2. For simplicity of notation, let $M = r^2$. Then $\mathbf{C_y}$ is a $M \times M$ matrix. Let $\{\mathbf{b}_p : p = 1 : M\}$ be the eigenvectors of $\mathbf{C_y}$, i.e. the principal neighborhoods, sorted in order of descending eigenvalues. Let the d-dimensional PCA subspace be the space spanned by $\{\mathbf{b}_p : p = 1 : d\}$. Then the projections of the image neighborhood vectors onto this subspace is given by

$$\mathbf{y}_d(i) = \sum_{p=1}^{d} \langle \mathbf{y}(i), \mathbf{b}_p \rangle \mathbf{b}_p, \tag{3}$$

where $\langle \mathbf{y}(i), \mathbf{b}_p \rangle$ denotes the inner product of the two vectors.

Let $\mathbf{f}_d(i) = [\langle \mathbf{y}(i), \mathbf{b}_1 \rangle \ldots \langle \mathbf{y}(i), \mathbf{b}_d \rangle]^T$ be the d-dimensional vector of projection coefficients. Then, due to the orthonormality of the basis functions

$$\|\mathbf{y}_d(i) - \mathbf{y}_d(j)\|^2 = \|\mathbf{f}_d(i) - \mathbf{f}_d(j)\|^2. \tag{4}$$

Finally, define a new family of estimators for $d \in [1, M]$

$$\hat{u}_d(i) = \sum_{j\in\mathcal{S}_i} \frac{1}{Z_d(i)} e^{-\frac{\|\mathbf{f}_d(i)-\mathbf{f}_d(j)\|^2}{h^2}} v(j), \tag{5}$$

where $Z_d(i) = \sum_{j\in\mathcal{S}_i} e^{-\|\mathbf{f}_d(i)-\mathbf{f}_d(j)\|^2/h^2}$ is the new normalizing term. Note that $\mathbf{y}_M(i) = \mathbf{y}(i)$; therefore, the proposed approach with $d = M$ is equivalent to the standard NLM, i.e. $\hat{u}_M(i) = \hat{u}_{NL}(i)$.

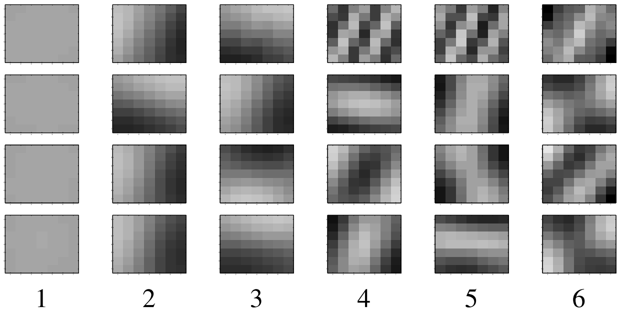


Fig. 1. Top six principal components for 7×7 image neighborhoods. Top to bottom rows: *Barbara*, *House*, *Lena* and *Peppers* principal components.

Figure 1 shows the top six principal neighborhoods, i.e. principal components, computed from 7×7 neighborhoods for the *Barbara*, *House*, *Lena* and *Peppers* images (see Figure 6). The first eigenvector (left column) corresponding to the largest eigenvalue of $\mathbf{C_y}$ is always approximately flat. This flat eigenvector represents the average intensity in the 7×7 neighborhood. The next two eigenvectors almost always represent two orthogonal gradient directions which are necessary for representing edges. The eigenvectors following these are more dependent on the specific image. Generally, the next few eigenvectors represent ridge patterns (rows 2-4 in Figure 1); however, in the case of strongly texture images, they can also represent the dominant texture patterns (*Barbara* - columns 4&5, row 1 in Figure 1). The *Barbara* image (Figure 6) is an example of the latter case due to the abundant stripe patterns. In [8], the correlation matrix is used in place of the covariance matrix. Differences in the principal neighborhoods of the covariance and correlation matrices are minor.

Significant principal neighborhoods are extremely robust to additive, independent and identically distributed noise. It is known that principal directions of a multivariate probability distribution function are not altered by addition of spherically symmetric noise. Therefore, for infinite sample sizes, the eigenvectors of the covariance matrix of image neighborhoods will not be altered by addition of independent and identically distributed noise to the image. Eigenvalues will be increased by the noise variance amount. The effects of noise on the eigenvalues of finite sample covariance matrices have also been previously investigated [39]. Experimental evidence suggests that principal neighborhoods (Figure 1) that correspond to the larger eigenvalues of the covariance matrix do not change in any noticeable way in the presence of noise.

C. Smoothing kernel width selection

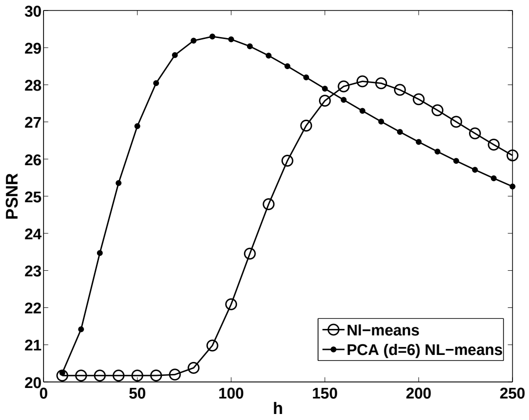


Fig. 2. PSNR (dB) as a function of the parameter h for the *peppers* image.

Given a noisy image and a combination of $\mathcal{N}$ and d, there exists an optimal choice of the parameter h in Equation (5) that yields the best output in terms of signal-to-noise ratio. To illustrate this point, Figure 2 shows the *peak signal-to-noise ratio* (PSNR) of the estimator output $\hat{u}$ as a function of h for an image that was corrupted with Gaussian noise ($\sigma = 25$). A rule-of-thumb for choosing h was given in [5] for the NLM algorithm with 7×7 image neighborhoods. More specifically, Buades *et al.* suggest using $h = 10\sigma$. However, this choice of h may not be optimal. Furthermore, the optimal choice for h varies significantly with the image neighborhood size (applies to PND and the NLM algorithm) and choice of subspace dimensionality (applies to PND). For instance, it can be seen from Figure 2 that the peak PSNR is obtained at a lower h value for the proposed approach with $d = 6$ than for the standard NLM algorithm. This observation conforms to our expectations because distances computed in the subspace are necessarily smaller than distances computed in the full-dimensional ambient space.

We will now show how rules for choosing near optimal h parameters can be learned. We start by empirically finding the optimal h for each combination of d and $\mathcal{N}$ for the set of test images used in this paper. This is repeated at various noise standard deviations σ added to the images. To be more specific, given a noisy image and a combination of d and $\mathcal{N}$, golden section search [40] is used to find the h parameter value that maximizes the output PSNR. The optimal value of h behaves in a very predictable manner as a function of the noise level σ and PCA subspace dimensionality d. In Figure 3, optimal h values are shown as a function of σ for $d = 10$ and $d = 49$ of 7×7 image neighborhoods. For a fixed d, the relationship between optimal h and σ is linear. Therefore, for the d-dimensional subspace of $r \times r$ image neighborhoods, h can be chosen with the rule

$$h = m(r, d)\sigma + c(r, d). \tag{6}$$

Figure 3 also shows the best linear fit to optimal h as a function of σ. We use these linear fits as an automatic way of choosing h given an image neighborhood size and d. Table I shows the linear fit parameters for several choices of d of 7×7 image neighborhoods. Also shown are the error in the linear fit to the optimal h values and the resulting loss of PSNR in the denoised images. We note that as expected, the PSNR loss resulting from using the automated h selection instead of the optimal h is small. Parameters such as those shown in Table I can be pre-computed for all $\mathcal{N}$ and d of interest. Furthermore, as can be seen in Figure 2, the output PSNR curves have smooth, broad maxima. In other words, the peak PSNR is somewhat robust to small sub-optimalities in the selection of h. Therefore, we expect that h produced by these linear fit parameters will produce results for a much larger set of images than those from which they were learned. It is important to note that the parameters in (6) also depend on patch size. Figure 3 demonstrates the analysis for 7×7 patches. The same analysis could be repeated for other patch sizes as well. An alternative method for selecting h could be to analyze the bias and variance of the estimator. This type of analysis is used for selecting a search-window size in [6].

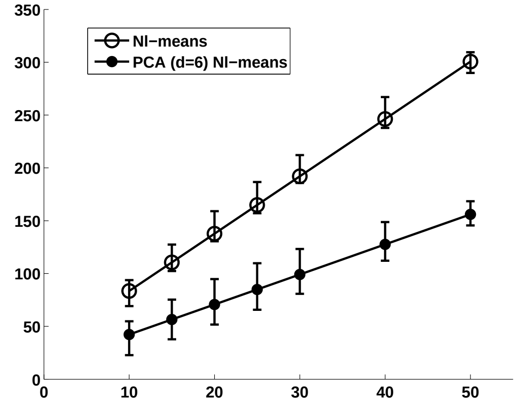


Fig. 3. Optimal h value as a function of Gaussian noise standard deviation σ. The data points correspond to the mean of the optimal h value for 8 test images while the bars demonstrate the minimum and maximum optimal h.

	$d = 6$	$d = 10$	$d = 20$	$d = 49$
Intercept (c)	13.81	22.55	29.31	29.17
Slope (m)	2.84	3.15	3.90	5.43
L_2 error for h fit	8.23	7.96	6.90	6.14
PSNR loss (dB)	0.14	0.12	0.08	0.06

TABLE I
Rows 1 & 2: Slope and intercepts used in determining h for various subspace dimensionality of 7×7 neighborhoods. Row 3& 4: Error in fit to optimal h and loss in output PSNR.

D. Automatic subspace dimensionality selection

The original parallel analysis method [14] compares the eigenvalues of the data covariance matrix to eigenvalues of the covariance matrix of an artificial data set. This artificial data set is generated by drawing samples from a multivariate normal distribution with the same dimensionality M, the same number of observations $|\Psi|$, and the same marginal standard deviations as the actual data. Let λ_p for $1 \leq p \leq M$ denote the eigenvalues of $\mathbf{C_y}$ sorted in descending order. Similarly, let α_p denote the sorted eigenvalues of the artificial data covariance matrix. Parallel analysis estimates data dimensionality as

$$d = max\left(\{1 \leq p \leq M | \lambda_p \geq \alpha_p\}\right). \qquad (7)$$

The intuition is that the α_p is a threshold for λ_p below which the *p'th* component is judged to have occurred due to chance.

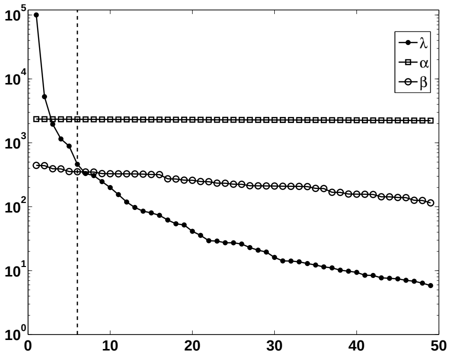


Fig. 4. Sorted eigenvalues of the *Lena* image (λ), eigenvalues from parallel analysis (α) and eigenvalues from modified parallel analysis (β).

An improvement to parallel analysis is to use Monte Carlo simulations to generate the artificial data [35] which removes the assumption of normal distribution. In our algorithm, we generate the artificial data by randomly permuting each element of the neighborhood vector across the sample Ψ. Let $y_{i,k}$ denote the *k'th* element of the neighborhood vector $\mathbf{y}(i)$. For each k generate a random permutation $j(i)$ of the sequence $i = 1 : |\Psi|$ and let $w_{i,k} = y_{j(i),k}$. Then, the random vectors $\mathbf{w}(i)$ are composed from the elements $w_{i,k}$. The artificial eigenvalues α are computed from the covariance matrix of $\mathbf{w}$. This method for computing the artificial covariance matrix keeps the marginal distributions intact while breaking any interdependencies between them. Figure 4 shows the λ and α computed in this manner from the *Lena* image. The number of significant components is under-estimated as two.

Several researchers have previously discussed that parallel analysis has a strong tendency to underestimate the number of components in data where the first component is much more significant than the rest of the components (oblique structure) [36], [37]. This is the case with image neighborhoods where the first component, which is always approximately the average intensity in the neighborhood (see Figure 1), has a much larger eigenvalue than the rest of the components. Therefore, we propose a modification to the parallel analysis algorithm in which we remove the effect of the first component. We compute the average intensity of the neighborhood $\mu_i = \frac{1}{M}\sum_{k \in \mathcal{N}_i} y_{i,k}$ and generate a new set of neighborhood vectors whose elements are $y'_{i,k} = y_{i,k} - \mu_i$. Finally, the artificial data are generated from the permutations $w_{i,k} = y'_{j(i),k}$. Figure 4 also shows artificial eigenvalues β computed in this modified manner. The number of significant components is found to be 6.

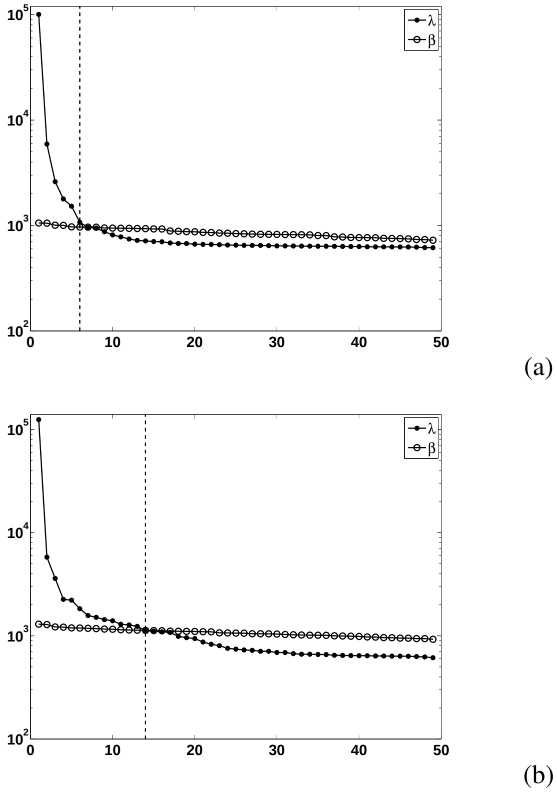


Fig. 5. Parallel Analysis: (a) *Lena* and (b) *Barbara* with noise $\sigma = 25$.

Figure 5(a) shows the parallel analysis applied to the noisy version ($\sigma = 25$) of the *Lena* image. The number of significant components is still computed as 6 which shows the robustness of the method. Figure 5(b) shows the parallel analysis results for the noisy *Barbara* image. In this case, the number of significant components is 14. This larger number can be attributed to the textured nature of the image which generates additional salient neighborhood components.

Notice that the moving from left to right in Figure 4, the β values decrease. Hence, the parallel analysis method is not equivalent to a fixed threshold applied to the data eigenvalues. This is different from [8] where data eigenvalues are directly compared to an estimate of noise variance. Another difference of the proposed dimensionality selection from the method used in [8] is that parallel analysis does not require a previous estimate of the noise variance. Finally, it can be seen from Figure 5 that a direct comparison to noise variance ($\sigma^2 = 25^2 = 625$ in this case) would result in a significantly larger dimensionality estimate than the parallel analysis method. This can be problematic because as will be discussed in Section IV-A, the dimensionality selected by parallel analysis correlates very well with the dimensionality that yields the best denoising results in experiments.

Typically, the artificial eigenvalues are simulated multiple times. However, we have found that if $|\Psi|$ is sufficiently large, i.e. 10% of the entire set of pixels Ω, a single simulation almost always gives the same result as multiple simulations. This is desirable from a computational complexity point of view.

It is well known that the least significant eigenvalue of a sample covariance matrix can be used as an estimator for the noise variance. More specifically, Muresan and Parks [4] have suggested using the smallest eigenvalue of the covariance of sample image neighborhoods as an estimator for noise variance; in other words, $\hat{\sigma} = \sqrt{\lambda_M}$. This estimator is biased to slightly underestimate the noise standard deviation due to the finite sample size; however, we find this bias to be quite small. Finally, this noise estimate can be used together with Equation (6) to select a h parameter. The pseudo-code for the PND algorithm is given below along with the subroutines for the modified parallel analysis and PCA.

Algorithm 1 PND (v,r)

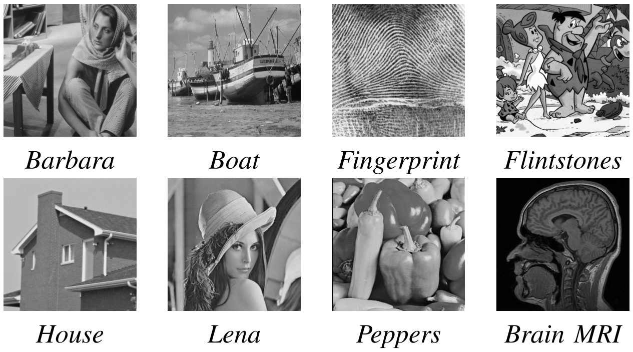
```
Generate all $r \times r$ image neighborhood vectors $\mathbf{y}(i)$
$M \leftarrow r^2$
Pick a random subsample $\Psi \subset \Omega$ with $\frac{|\Omega|}{10}$ elements
$\{\lambda_p, \mathbf{b}\}_{p=1}^M \leftarrow \mathrm{PCA}(\{\mathbf{y}(i) : i \in \Psi\})$
$\hat{\sigma} \leftarrow \sqrt{\lambda_M}$
$d \leftarrow \mathrm{ParallelAnalysis}\ \big(\{\mathbf{y}(i) : i \in \Psi\}, \{\lambda_p\}_{p=1}^M\big)$
**for** $i = 1$ to $|\Omega|$ **do**
    $\mathbf{f}_d(i) \leftarrow [\langle \mathbf{y}(i), \mathbf{b}_1 \rangle \ldots \langle \mathbf{y}(i), \mathbf{b}_d \rangle]^T$
**end for**
$h \leftarrow m(r,d)\hat{\sigma} + c(r,d)$
**for** $i = 1$ to $|\Omega|$ **do**
    $Z_d(i) = \sum_{j \in \mathcal{S}_i} e^{-\|\mathbf{f}_d(i) - \mathbf{f}_d(j)\|^2/h^2}$
    $\hat{u}_d(i) \leftarrow \sum_{j \in \mathcal{S}_i} \frac{1}{Z_d(i)} e^{-\frac{\|\mathbf{f}_d(i) - \mathbf{f}_d(j)\|^2}{h^2}} v(j)$
**end for**
**return** $\hat{u}$
```

Algorithm 2 ParallelAnalysis $\big(\{\mathbf{y}_i\}_{i=1}^N, \{\lambda_p\}_{p=1}^M\big)$

```
**for** $i = 1$ to $N$ **do**
    $\mu_i \leftarrow \frac{1}{M} \sum_{k \in \mathcal{N}_i} y_{i,k}$
    **for** $k = 1$ to $M$ **do**
        $y'_{i,k} \leftarrow y_{i,k} - \mu_i$
    **end for**
**end for**
**for** $k = 1$ to $M$ **do**
    Generate $j(i)$, a random permutation of numbers 1 to $N$
    Let $w_{i,k} \leftarrow y'_{j(i),k}$ for $i = 1 : N$
**end for**
$\{\beta_p\}_{p=1}^M \leftarrow \mathrm{PCA}\big(\{\mathbf{w}(i)\}_{i=1}^N\big)$
**return** $\underset{1 \geq p \geq M}{\arg\max}\ \lambda_p \geq \beta_p$.
```

Algorithm 3 PCA $\big(\{\mathbf{x}(i)\}_{i=1}^N\big)$

```
$\bar{\mathbf{y}} \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbf{y}(i)$
$\mathbf{C} \leftarrow \frac{1}{N} \sum_{i=1}^N (\mathbf{y}(i) - \bar{\mathbf{y}})\,(\mathbf{y}(i) - \bar{\mathbf{y}})^T$
**return** Sorted eigenvalues and eigenvectors $\{\lambda_p, \mathbf{b}_p\}_{p=1}^M$ of the $M \times M$ covariance matrix $\mathbf{C}$
```



Barbara *Boat* *Fingerprint* *Flintstones*

House *Lena* *Peppers* *Brain MRI*

Fig. 6. Images used in the experiments.

IV. Experiments

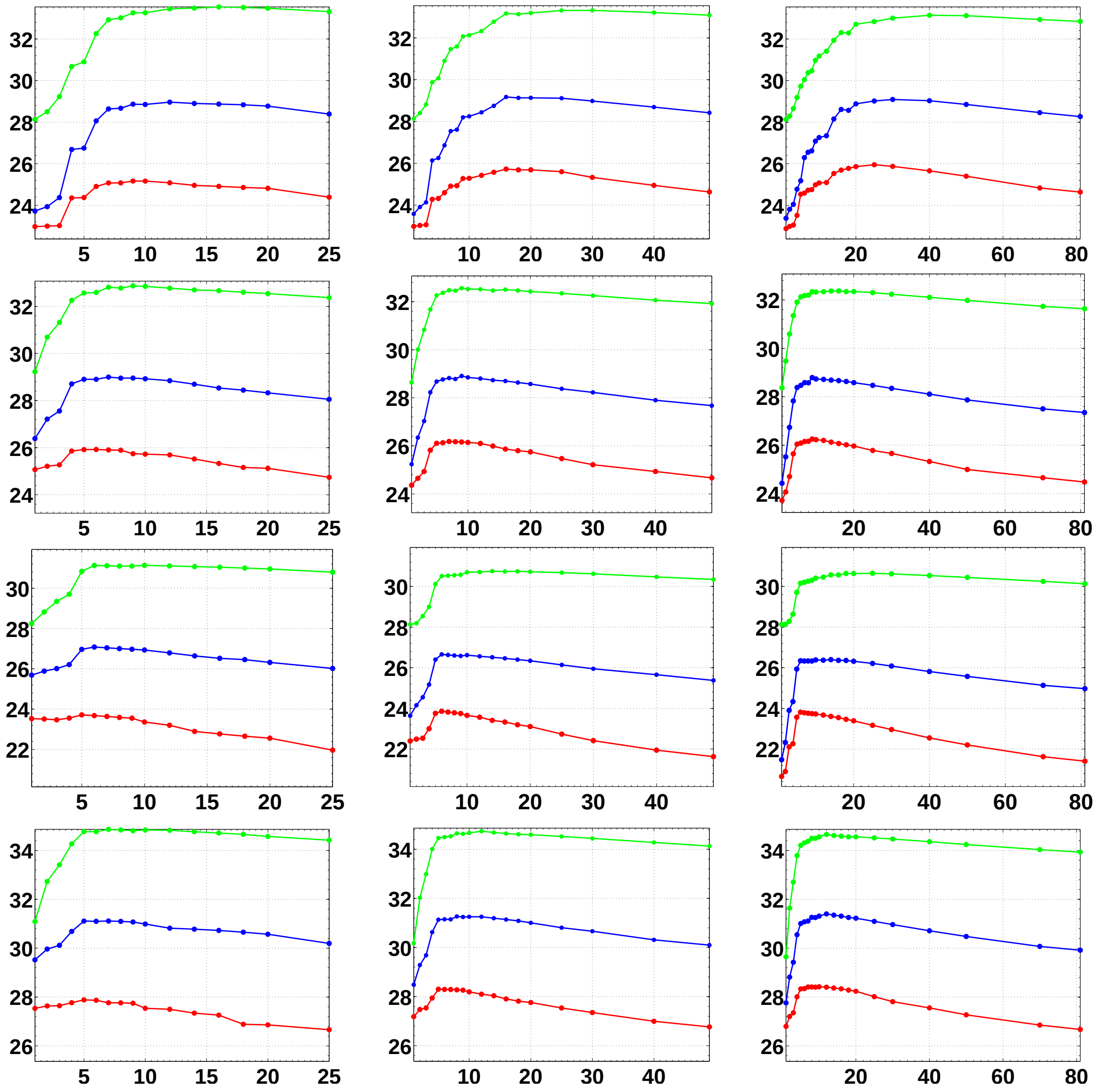
In this section we present detailed experimental results studying the behavior of the PND algorithm with respect to subspace dimensionality (Section IV-A) and search-window size selections (Section IV-C). We also present quantitative and qualitative comparisons with the original NLM algorithm [5] as well as the Adaptive PCA (APCA) algorithm [4] (Section IV-B). All of the experiments were performed on a set of eight images (shown in Figure 6) including those used by Portilla etal [24] and several additional images. We study the performance of the proposed approach using images corrupted with additive, independent Gaussian noise with standard deviation σ $10, 25$ and 50.

A. Subspace Dimensionality and Image Neighborhood Size

We first present results that systematically study the behavior of the estimator given by Equation (5) with respect to the PCA subspace dimensionality. In Section IV-B, we will compare the performance of the full PND algorithm, including automatic dimensionality selection, to the NLM [5] and APCA [4] algorithms.

Figure 7 illustrates the best PSNR at the optimal h parameter values. The rows correspond to four of the eight test images; the other four images behave similarly and are omitted for brevity. From left to right, the columns correspond to 5×5, 7×7 and 9×9 image neighborhoods. In each graph, the PSNR of the denoised image is plotted against the PCA subspace dimensionality. Recall that when the PCA subspace dimensionality is equal to the number of pixels in $\mathcal{N}$ the proposed algorithm is equivalent to the NLM algorithm. Therefore, the original NLM algorithm corresponds to the rightmost data point of each graph. Finally, each graph shows three curves corresponding to the three input noise levels.

In all cases, the best results are obtained at a relatively low PCA subspace dimensionality d. The curves shown in Figure 7 (except for *Barbara*) have a very characteristic shape: steeply increasing PSNR for $d < 6$, a knee around $d = 6$ and flat or gradually declining PSNR for $d > 6$. For higher noise levels, the PSNR declines significantly beyond the knee whereas for lower noise levels it is flatter. In other words, the advantage of the proposed approach over the standard NLM algorithm increases with higher input noise levels. The increased accuracy at lower d values can be attributed to the



5×5 neighborhood 7×7 neighborhood 9×9 neighborhood

Fig. 7. PSNR (dB) vs. PCA subspace dimensionality for various neighborhood sizes and noise levels. *Low noise* ($\sigma = 10$) green curve, *medium noise* ($\sigma = 25$) blue curve, *high noise* ($\sigma = 50$) red curve. Top to bottom rows: *Barbara, Boat, Fingerprint* and *Lena* images.

observation that distances computed in the lower-dimensional space are likely to be more accurate than distance computed from the full-dimensional space because PCA discards the most irrelevant dimensions. This explanation based on the accuracy of distances is also supported by the observation that the difference in PSNR between PND and NLM increases with increasing input noise level.

For the *Barbara* image the best d ranges from 7 to 20 depending on the image neighborhood size and input noise level. The higher subspace dimensions necessary for obtaining the best PSNR with this image can be attributed to its complex textured nature. Recall from Figure 1 that unlike the other images, the top principal components of *Barbara* image include texture components (stripe patterns which are common in that image). This results in a larger number of salient principal components compared to the other images.

While Figure 7 clearly illustrates the effects of PCA subspace dimensionality on the quality of denoising results, Table II offers an easier comparison across various image neighborhood sizes. For each test image, Table II includes three rows, one for each input noise level. Each row gives the best PSNR values at the optimal choice of d for different neighborhood sizes. In other words, the best PSNR value corresponding to the maxima of the curves in Figure 7 is included in Table II. Results for image neighborhoods ranging from 3×3 to 9×9 are provided. Also, results by using only the center pixel intensity (1×1 image neighborhood) are given for comparison. Finally, the overall best PSNR across the various neighborhood sizes for a particular image and noise level is shown in boldface.

As other researchers have previously shown, the NLM algorithm outperforms algorithms which only use the center pixel intensity such as bilateral filtering. Also notice that the best image neighborhood size increases with input noise

	Input	1×1	3×3	5×5	7×7	9×9
Barbara	28.13	30.39	33.41	**33.54**	33.32	33.14
	20.17	24.49	28.14	28.96	**29.17**	29.09
	14.15	21.53	24.08	25.17	25.72	**25.95**
Boat	28.13	30.64	**33.07**	32.87	32.57	32.37
	20.17	25.11	28.59	**29.00**	28.91	28.79
	14.15	22.19	25.26	25.92	26.18	**26.25**
Fingerprint	28.13	28.75	**31.92**	31.13	30.75	30.65
	20.17	22.00	**27.24**	27.08	26.66	26.40
	14.15	18.23	23.18	23.72	**23.86**	23.81
Flinstones	28.13	30.42	**31.92**	31.54	31.22	30.92
	20.17	23.40	27.49	**27.89**	27.51	27.03
	14.15	18.63	23.61	24.29	**24.40**	24.08
House	28.13	32.41	35.44	**35.45**	35.09	34.89
	20.17	25.84	30.60	31.61	31.73	**31.75**
	14.15	22.38	27.00	27.88	28.45	**28.69**
Lena	28.13	31.57	34.77	**34.86**	34.74	34.66
	20.17	26.27	30.40	31.11	31.27	**31.40**
	14.15	23.39	27.12	27.88	28.30	**28.42**
Peppers	28.13	30.66	33.73	**33.81**	33.53	33.30
	20.17	24.42	28.48	**29.38**	29.30	29.06
	14.15	20.40	24.70	25.70	**26.11**	26.07
Brain MRI	28.13	31.08	**33.39**	33.26	33.12	33.08
	20.17	24.98	28.75	**29.31**	29.24	29.27
	14.15	22.10	25.29	26.04	26.40	**26.41**

TABLE II

PSNR VALUES AT THE OPTIMAL SUBSPACE DIMENSIONALITY. INPUT PSNR FOR THREE NOISE LEVELS ($\sigma = 10, 25, 50$) SHOWN IN COLUMN 2. COLUMNS 3-7 SHOW THE RESULTS AT THE BEST DIMENSIONALITY FOR NEIGHBORHOOD SIZES FROM 1×1 TO 9×9. THE OVERALL BEST PSNR ACROSS THE NEIGHBORHOOD SIZES FOR A PARTICULAR IMAGE AND NOISE LEVEL IS SHOWN IN BOLDFACE.

	Noise σ	5×5	7×7	9×9
		$25D$	$49D$	$81D$
Barbara	10	9	13	18
	25	9	13	18
	50	10	17	19
Boat	10	6	9	13
	25	7	9	13
	50	9	9	13
Fingerprint	10	5	7	9
	25	5	7	9
	50	5	7	9
Flinstones	10	6	9	13
	25	6	9	13
	50	6	9	14
House	10	5	7	9
	25	6	7	9
	50	7	7	11
Lena	10	5	6	10
	25	5	6	10
	50	18	7	10
Peppers	10	5	6	9
	25	6	6	9
	50	6	6	10
Brain MRI	10	6	10	13
	25	7	10	13
	50	7	11	13

TABLE III

PCA SUBSPACE DIMENSIONALITY SELECTED BY PARALLEL ANALYSIS.

	5×5	5×5	7×7	7×7	9×9	9×9
	PND	NL	PND	NL	PND	NL
Barbara	33.15	**33.29**	32.41	**33.05**	31.73	**32.81**
	28.83	28.36	**28.67**	28.41	**28.50**	28.27
	25.16	24.39	**25.68**	24.62	**25.72**	24.64
Boat	**32.43**	32.13	**32.38**	31.55	**32.03**	31.22
	28.95	28.01	**28.90**	27.66	**28.66**	27.34
	25.74	24.74	**26.16**	24.66	**26.14**	24.48
Fingerprint	30.48	**30.69**	30.06	**30.24**	29.85	**30.00**
	26.95	26.02	**26.45**	25.37	**25.98**	24.94
	23.68	21.90	**23.82**	21.59	**23.74**	21.38
Flinstones	31.36	**31.62**	31.01	**31.22**	30.79	**30.90**
	27.74	26.88	**26.96**	26.08	**26.42**	25.44
	24.29	22.74	**24.08**	22.24	**23.66**	21.71
House	**35.26**	34.94	**34.81**	34.61	**34.56**	34.41
	31.59	30.52	**31.59**	30.40	**31.48**	30.18
	27.83	26.27	**28.40**	26.33	**28.63**	26.18
Lena	**34.77**	34.32	**34.43**	33.98	**34.50**	33.79
	31.11	30.20	**31.14**	30.09	**31.31**	29.91
	26.71	26.56	**28.27**	26.72	**28.42**	26.65
Peppers	**33.76**	33.24	**33.46**	32.81	**33.14**	32.49
	29.37	28.39	**29.26**	28.01	**28.95**	27.61
	25.68	24.49	**26.10**	24.31	**25.93**	23.91
Brain MRI	**33.07**	32.61	**32.57**	32.15	**32.49**	31.88
	29.24	28.37	**29.03**	28.08	**29.09**	27.76
	26.04	24.81	**26.33**	24.87	**26.38**	24.72

TABLE IV

PSNR FOR IMAGES DENOISED WITH PND AND NLM [5]. FOR EACH IMAGE, THE 3 ROWS CORRESPOND TO NOISE LEVELS $\sigma = 10, 25$ AND 50.

level. It can be argued that this is a trade-off between the reliability of weights vs. curse of dimensionality. Larger image neighborhoods result in a higher-dimensional feature space. This results in a sparser samples (curse of dimensionality) and less reliable weighted averages in equations (1) and (5) due to a lack of nearby sample points. On the other hand, larger image neighborhoods also provide a more pronounced averaging effect in equations (1) and (5) due to the larger spatial extent of the principal neighborhoods. This can result in weights less susceptible to noise. As the noise level increases, weight reliability becomes increasingly important; hence, larger image neighborhoods are preferred.

B. Comparison with NLM and APCA algorithms

Table III shows the d values selected by parallel analysis as described in Section III-D for various neighborhood sizes and noise levels. Table IV compares the results of the proposed algorithm with these automatically chosen d and h values to the results of the NLM algorithm. The h parameter for the NLM algorithm is also selected with the same rules for a fair comparison. Note that this results in better PSNR outcomes for the NLM algorithm compared to choosing $h = 10\sigma$ as suggested in [5]. When the noise is moderate or high, PND significantly (greater than 1 dB difference) outperforms NLM denoising. The advantages of the proposed approach increase with increasing noise level. For low noise ($\sigma = 10$), PND performs slightly better for five of the eight images while NLM performs slightly better for three of the eight. As the noise level decreases, the length of the projection of image neighborhoods in the complementary space that is orthogonal to the PCA subspace also decreases. Consequently, the distances in the PCA subspace become better approximations to the distances in the full-dimensional space. In other words, the difference between the two distance computations become minimal, which in turn results in very similar performance of the two approaches. However, note that image neighborhoods can not be perfectly represented in reduced dimensionality linear subspaces; hence, in the absence of noise, distances computed in lower-dimensional subspaces are sub-optimal.

	5×5 PND	5×5 APCA	7×7 PND	7×7 APCA	9×9 PND	9×9 APCA
Barbara	33.15	**34.53**	32.41	**34.78**	31.73	**34.82**
	28.83	**29.67**	28.67	**30.05**	28.50	**30.03**
	25.16	**25.90**	25.68	**26.40**	25.72	**26.28**
Boat	32.43	**33.40**	32.38	**33.46**	32.03	**33.43**
	28.95	**29.04**	28.90	**29.12**	28.66	**29.01**
	25.74	**25.76**	**26.16**	25.89	**26.14**	25.66
Fingerprint	30.48	**32.47**	30.06	**32.42**	29.85	**32.34**
	26.95	**27.48**	26.45	**27.42**	25.98	**27.30**
	23.68	**24.04**	23.82	**24.02**	23.74	**23.84**
Flinstones	31.36	**31.25**	31.01	**31.23**	30.79	**31.22**
	27.74	26.99	26.96	**27.09**	26.42	**27.06**
	24.29	23.40	**24.08**	23.51	**23.66**	23.39
House	35.26	**35.55**	34.81	**35.80**	34.56	**35.82**
	31.59	30.78	**31.59**	30.99	**31.48**	30.80
	27.83	27.12	**28.40**	27.34	**28.63**	26.92
Lena	34.77	**35.53**	34.43	**35.62**	34.50	**35.57**
	31.11	**31.22**	31.14	**31.40**	**31.31**	31.23
	26.71	**27.51**	**28.27**	27.76	**28.42**	27.39
Peppers	**33.76**	33.40	**33.46**	33.45	33.14	**33.41**
	29.37	28.76	**29.26**	28.83	**28.95**	28.68
	25.68	25.32	**26.10**	25.38	**25.93**	25.02
Brain MRI	33.07	**33.32**	32.57	**33.37**	32.49	**33.34**
	29.24	29.19	29.03	**29.19**	**29.09**	29.00
	26.04	**26.23**	**26.33**	26.19	**26.38**	25.77

TABLE V
PSNR FOR IMAGES DENOISED WITH PND AND APCA [4]. FOR EACH IMAGE, THE 3 ROWS CORRESPOND TO NOISE LEVELS $\sigma = 10, 25$ AND 50.

	Barbara	*Boat*	*Finger-print*	*Flints-tones*
5×5 PND	64	58	44	55
5×5 NLM	107	106	105	104
5×5 APCA	64	64	63	64
7×7 PND	78	64	56	64
7×7 NLM	171	171	171	171
7×7 APCA	117	117	118	116
9×9 PND	85	76	65	78
9×9 NLM	249	250	250	251
9×9 APCA	239	239	231	230

	House	*Lena*	*Peppers*	*Brain MRI*
5×5 PND	14	67	13	15
5×5 NLM	26	105	26	26
5×5 APCA	16	64	16	16
7×7 PND	15	55	14	17
7×7 NLM	42	172	42	42
7×7 APCA	29	118	30	31
9×9 PND	16	67	16	19
9×9 NLM	61	250	62	62
9×9 APCA	58	234	58	58

TABLE VI
COMPUTATION TIMES (IN SECONDS) FOR THE PND, NLM APCA ALGORITHMS. ALL ALGORITHMS WERE CODED IN *MATLAB*. THE *House*, *Peppers* AND *Brain MRI* ARE 256×256, ALL OTHERS ARE 512×512.

We also compare the PND method to the Adaptive PCA (APCA) method of Muresan and Parks [4]. In APCA local PCAs are used to project the image neighborhoods on to a local basis. The maximum likelihood estimator for a given projection coefficient is computed using estimates of noise variance and that coefficient's variance. Similar to PND and NLM, the APCA algorithm entails a choice of image neighborhood size. We tested the APCA algorithm with 5×5, 7×7 and 9×9 neighborhood sizes. For the rest of the parameters in our implementation of APCA, we use the suggestions given in [4]: The size of the window used for local PCA is referred to as the training region in [4] and is fixed as a 21×21 square window. The maximum likelihood estimators apply to the 7×7 central denoising region of each training region. Finally, adjacent denoising regions overlap by three pixels as suggested in [4] to avoid blocking artifacts. Table V compares the PND and APCA methods for the various image neighborhood sizes and noise variances. In general, APCA outperforms PND; however, for the *House* and *Peppers* images PND performs better. The performances are roughly even for the *Flinstones* and *Brain MRI* images. However, despite the overall better quantitative performance of the APCA algorithm over PND, there are visual artifacts that are associated with APCA denoising which do not occur with the PND or NLM algorithms. Figure 8 illustrates detailed denoising results for the APCA, PND and NLM algorithms for portions of some test images. Local, directional oscillatory artifacts associated with the APCA algorithm can be seen in the *Barbara* and *House* images. These artifacts are most noticeable in the tablecloth region of the *Barbara* and the background in the *House* image. Furthermore, the artifacts are more significant in the cases with higher noise. Artifacts are not observed in the *Fingerprint* image, possibly because the local PCA model is a very good fit for the homogeneous texture pattern found in this image. Finally, notice that the PND algorithm outperforms NLM visually as well as quantitatively. This can be seen in the tablecloth, pants and other striped patterned regions of the *Barbara* image, especially for the case with higher noise.

We have implemented the PND, NLM and APCA algorithms in MATLAB (Copyright The Mathworks, Inc.). Table VI provides the run times of the two algorithms for the eight test images. The run times for the NLM and APCA algorithms are approximately the same for images of the same size, whereas the run times for the PND algorithm depend on the subspace dimension d selected by the parallel analysis subroutine (See Table III). PND is always faster than NLM and APCA. The computational complexity of NLM is $O(|\Omega| \cdot |\mathcal{S}| \cdot M)$ where $|\Omega|$, $|\mathcal{S}|$ and M are the number of pixels in the image, in the search window $\mathcal{S}$ and in the neighborhood vector $\mathcal{N}$, respectively. In comparison, the complexity when using a d-dimensional subspace is $O(|\Omega| \cdot |\mathcal{S}| \cdot d)$. The additional costs in building the covariance matrices for PCA and parallel analysis is $O(|\Psi| \cdot M^2)$. The cost for computing the projection coefficients $\{f_p : p = 1 : d\}$ are and $O(|\Omega| \cdot M \cdot d)$. Eigenvectors are computed once globally for a small matrix ($M \times M$); this cost is negligible. As pointed out in Algorithm 1, we choose Ψ as a random subset of Ω with one tenth the size; hence, $|\Psi| = 0.1|\Omega|$. Therefore, the total complexity for PND is $O\left(|\Omega| \cdot \left(|\mathcal{S}|d + M^2 + 0.1Md\right)\right)$. This is significantly smaller than the NLM cost because typically $|\mathcal{S}| >> M$. For typical window sizes used in the literature $|\mathcal{S}| = 441$ and $M = 49$. If larger search windows $\mathcal{S}$ are used, which can be desirable from a PSNR point of view, the computational savings over NLM and APCA increase.

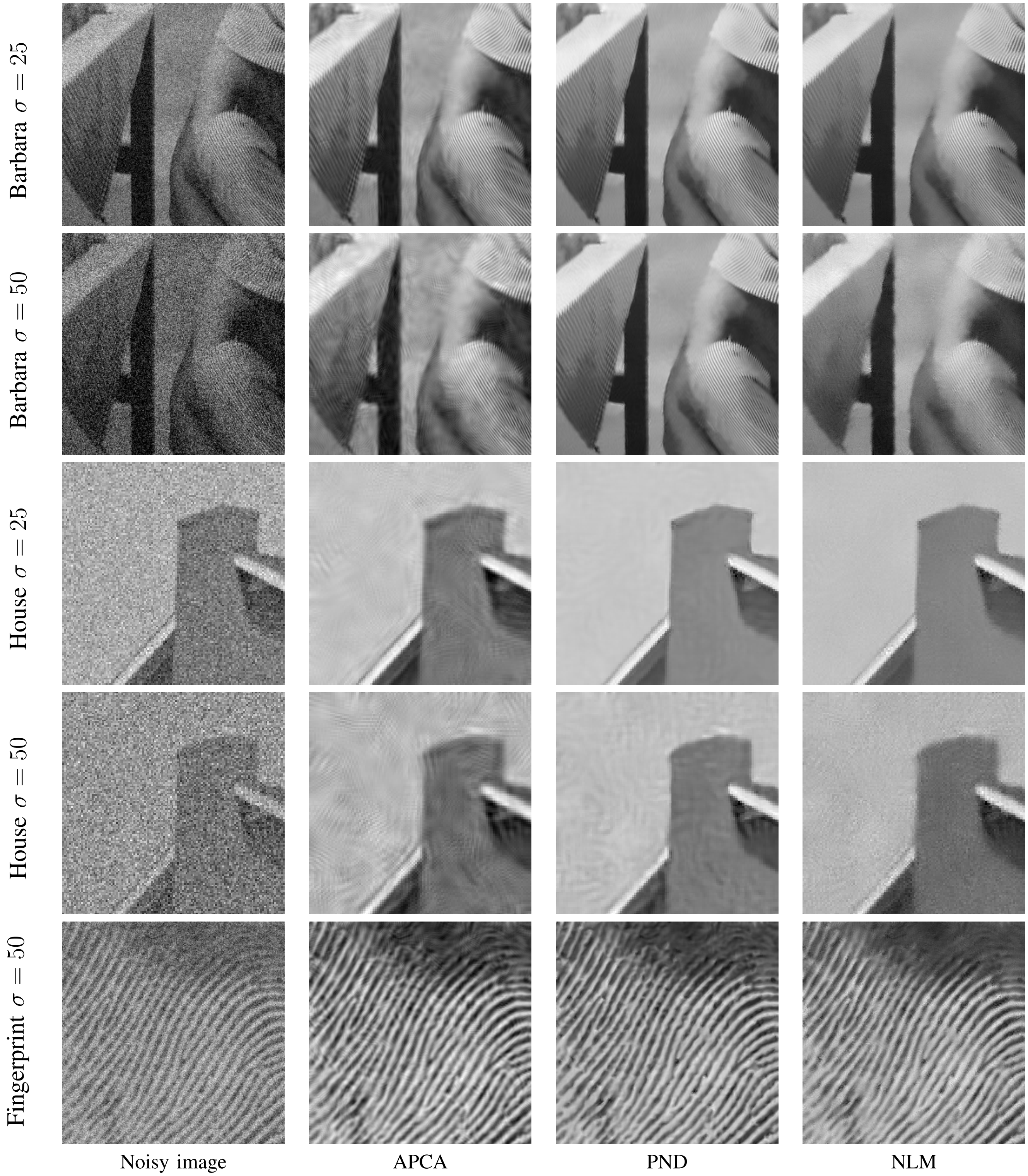


Fig. 8. Visual comparison of APCA [4], PND and NLM denoising [5] using 7 × 7 image neighborhoods.

C. Effect of search window size

As explained in Section III-A, this paper and other non-local means image filtering approaches in the literature use the limited-range implementation for computational feasibility. In the limited-range implementation, the search window $\mathcal{S}_i$ in (1) is defined to be a square window of limited size centered at pixel i rather than the entire set of pixels in the image. This brings into question whether the limited-range implementation effects the performance of NLM. In other words, how does denoising performance change with the size of $\mathcal{S}_i$? Most non-local means based papers in the literature such as [5], [29], [8], [9], [10] have chosen a fixed search window size. In contrast, Kervrann and Boulanger [6] propose an adaptive search-window approach based on an analysis of estimator's the bias and variance. In this section, we investigate the effects of search-window size in fixed size approaches.

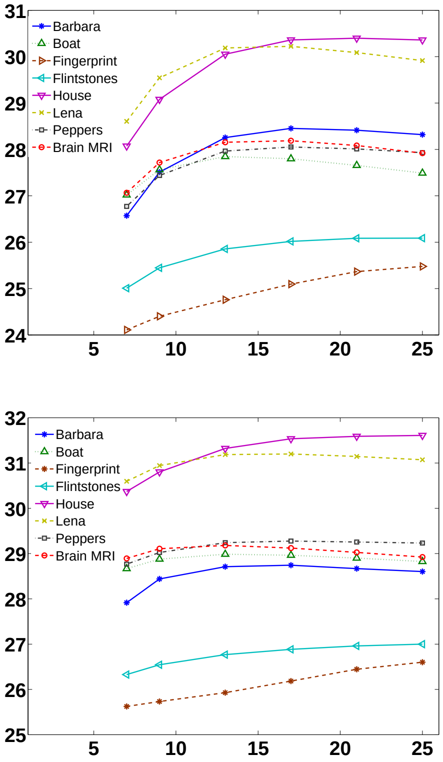


Fig. 9. PSNR vs. length of search window side. Top: NLM. Bottom: PND.

Figure 9 plots the PSNR performance of the NLM and PND algorithms vs search window size. More specifically, the x-axis is the length of the sides of the square window $\mathcal{S}$. From Figure 9, we can see that denoising performance for both algorithms first increases with search window size and then saturates beyond a size of approximately 17 × 17. In fact, for certain images such as *Lena*, *Brain MRI* and *Boat* there is a slight degradation in performance beyond this size. This performance degradation is more noticeable with NLM than PND. The only exception appears to be the *Fingerprint* image for which denoising accuracy monotonically increases for all search window sizes we tested. These observations suggest that, for most images. the success of the NLM algorithm could be attributed more to its use of image neighborhoods than its non-local nature. In general, it is likely that using larger search-windows do not provide additional neighborhood examples that are close to the one being denoised. In the case of the *Fingerprint* image there is a single dominant texture, modulo rotation, that is present everywhere. Hence, larger search-windows can provide additional useful examples; however, due to the presence of rotation, we still expect that PSNR accuracy would saturate beyond a certain point. The decreasing performance phenomenon needs to be investigated further. One possible cause is related to the choice of the smoothing kernel width parameter. It also suggests that both NLM and PND can benefit from an adaptive search-window size such as the one proposed in [6].

V. Conclusion

The accuracy and computational cost of the NLM image denoising algorithm [5] is improved by computing neighborhood similarities, i.e. averaging weights, after a PCA projection to a lower-dimensional subspace. Unlike the predetermined filters introduced in [29] for reducing the NLM computational cost, PND and the methods described in [8], [10] are data-driven and can adapt to the statistics of a given image. Also, in [29], the weights are still computed from the original high-dimensional vectors after the selection of neighborhoods to include in the weighted average. In this work, the lower-dimensional projections are not only used as a search criteria but also for computing similarity weights resulting in better accuracy in addition to reduced computational cost. It is clear that image neighborhoods can not be modeled in a global, linear subspace. Nevertheless, better accuracy is explained in terms of the increased reliability in the similarity weights when they are computed in a subspace that captures most of the true variability in neighborhoods and limits the effects of noise. For images with very regular textures such as a fingerprint image, it was observed that APCA performs better than PND both visually and quantitatively. This results suggests that i) a semi-local PCA instead of global PCA could also benefit the proposed method, and ii) it might be necessary to adaptively select a different subspace dimensionality in different image parts to better capture texture patterns. Finally, we showed that parallel analysis can be used to automatically determine a subspace dimensionality that yields good results.

The NLM algorithm has been previously applied to color images by measuring L_2 distance in the RGB image neighborhood space [29]. Similarly, the proposed approach can be extended to color images by performing in the RGB image neighborhood space which is formed by concatenating image neighborhoods in the three channels into a single vector. Another interesting direction for future research is to use a separate bandwidth for each element of the projected vector $\mathbf{f}_d$ in equation (5). While the elements of $\mathbf{y}$ have equal marginal distributions, the same is not true for $\mathbf{f}_d$, and a possible room for improvement is to use the PCA eigenvalues to determine different bandwidth parameters for the different projection coefficients. Finally, the Principal Neighborhoods approach can also be easily applied to other denoising and segmentation algorithms that use similarity measures based on image neighborhood vectors [7], [6], [13].

ACKNOWLEDGEMENT

This work was supported by NIH R01 EB005832 and NSF CCF0732227. The author would also like to acknowledge the support of the Utah Science Technology and Research Initiative. We thank Joshua Cates, Thomas Fletcher and Ross Whitaker for helpful discussions on parallel analysis. We also thank the anonymous reviewers whose comments helped greatly improve the paper.

REFERENCES

[1] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 4, pp. 577–685, 1989.

[2] P. Perona and J. Malik, "Scale-space edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[3] L. I. Rudin, S. J. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[4] D. D. Muresan and T. W. Parks, "Adaptive principal components and image denoising," in *Proc. Int. Conf. on Image Proc.*, 2003, vol. 1, pp. 101–104.

[5] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Conf. on Comp. Vis. and Pattern Recog.*, 2005, pp. 60–65.

[6] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. on Image Proc.*, vol. 15, no. 10, pp. 2866–2878, 2006.

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

[8] N. Azzabou, N. Paragios, and F. Guichard, "Image denoising based on adapted dictionary computation," in *Proc. Int. Conf. on Image Proc.*, 2007, vol. III, pp. 109–112.

[9] T. Tasdizen, "Principal components for non-local means image denoising," in *Proc. Int. Conf. on Image Proc.*, 2008.

[10] J. Orchard, M. Ebrahim, and A. Wang, "Efficient nonlocal-means denoising using the SVD," in *Proc. Int. Conf. on Image Proc.*, 2008.

[11] A. A. Efros amd T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. on Comp. Vis.*, 1999, pp. 1033–1038.

[12] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proc. SIGGRAPH*, 2000, pp. 479–488.

[13] S. P. Awate, T. Tasdizen, and R. T. Whitaker, "Unsupervised Texture Segmentation with Nonparametric Neighborhood Statistics," in *Proc. European Conference on Comp. Vis.*, 2006, pp. 494–507.

[14] J. L. Horn, "A rationale and test for the number of factors in factor analysis," *Psychometrica*, vol. 30, no. 2, pp. 179–185, 1965.

[15] J. Weickert, *Anisotropic Diffusion in Image Proc.*, Teubner-Verlag, 1998.

[16] T. Chan, J. Shen, and L. Vese, "Variational pde models in image proc.," *Notice of Am. Math. Soc.*, vol. 50, pp. 14–26, 2003.

[17] K. N. Nordstrom, "Biased anisotropic diffusion: A unified regularization and diffusion approach to edge detection," *Image Vision Computing*, vol. 8, no. 4, pp. 318–327, 1990.

[18] S. Kindermann, S. Osher, and P. W. Jones, "Deblurring and denoising of images by nonlocal functionals," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1091–1115, 2005.

[19] T. Brox, O. Kleinschimdt, and D. Cremers, "Efficient nonlocal means for denoising of textural patterns," *IEEE Trans. on Image Proc.*, vol. 17, no. 7, pp. 1083–1092, 2008.

[20] A. Hyvarinen, P. Hoyer, and E. Oja, "Sparse code shrinkage: Denosing of nongaussian data by maximum likelihood estimation," *Neural Computation*, vol. 11, no. 7, pp. 1739–1768, 1999.

[21] J. Starck, E. Candes, and D. Donoho, "The curvelet transform for image denoising," *IEEE Trans. on Image Proc.*, vol. 11, no. 6, 2000.

[22] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy, "A joint inter and intrascale statistical model for bayesian wavelet based image denoising," *IEEE Trans. on Image Proc.*, vol. 11, pp. 545–557, 2002.

[23] L. Sendur and I. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *IEEE Trans. on Signal Proc.*, vol. 50, pp. 2744–2756, 2002.

[24] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Trans. on Image Proc.*, vol. 12, pp. 1338–1351, 2003.

[25] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionary," *IEEE Trans. on Image Proc.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[26] S. Roth and M. J. Black, "Fields of experts: a framework for learning image priors with applications," in *Proc. IEEE Conf. on Comp. Vis. and Pattern Recog.*, 2005, vol. 2, pp. 860–867.

[27] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. on Comp. Vis.*, 1998, pp. 839–846.

[28] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," in *Proc. SPIE Electronic Imaging: Alg. and Sys., N. Networks, and Machine Learning*, 2006, vol. 6064.

[29] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Proc. Letters*, vol. 12, no. 12, pp. 839–842, 2005.

[30] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Proc. IEEE Conf. on Comp. Vis. and Pattern Recog.*, 2004, vol. 2, pp. 506–513.

[31] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. on Comp. Vis.*, 1999, pp. 1150–1157.

[32] J. L. Horn and R. Engstrom, "Cattell's scree test in relation to bartlett's chi-square test and other observations on the number of factors problem," *Multivariate Behavioral Research*, vol. 14, pp. 283–300, 1979.

[33] W. F. Velicer, "Determining the number of components from the matrix of partial correlations," *Psychometrika*, vol. 41, pp. 321–327, 1976.

[34] J. C. Hayton, D. G. Allen, and V. Scarpello, "Factor retention decisions in exploratory factor analysis: A tutorial on parallel analysis," *Organizational Research Methods*, vol. 7, no. 2, pp. 191–205, 2004.

[35] L. W. Glorfeld, "An improvement on horn's parallel analysis methodology for selecting the correct number of factor's to retain," *Educ. and Psych. Measurement*, vol. 55, no. 3, pp. 377–393, 1995.

[36] N. E. Turner, "The effect of common variance ans structure on random data eigenvalues: implications for the accuracy of parallel analysis," *Educ. and Psych. Measurement*, vol. 58, no. 4, pp. 541–568, 1998.

[37] A. Beauducel, "Problems with parallel analysis in data sets with oblique simple structure," *Methods of Physchological Research Online*, vol. 6, no. 2, pp. 141–157, 2001.

[38] J. Cates, P. T. Fletcher, M. Styner, H. Hazlett, and R. T. Whitaker, "Particle-based shape analysis of multi-object complexes," in *LNCS 5241: Proc. Int. Conf. on Medical Image Computing and Comp. Assisted Intervention*, 2008, pp. 477–485.

[39] R. Everson and S. Roberts, "Infering the eigenvalues of covariance matrices from limited, noisy data," *IEEE Trans. on Signal Proc.*, vol. 48, no. 7, pp. 2083–2091, 2000.

[40] J. Kiefer, "Sequential minimax search for a maximum," in *Proc. American Mathematical Society*, 1953, pp. 502–506.

Tolga Tasdizen Dr. Tolga Tasdizen received the B.S. degree in electrical and electronics engineering from Bogazici University in 1995. He received his M.S. and Ph.D. degrees in engineering from Brown University in 1997 and 2001, respectively. After a postdoctoral researcher position in the Scientific Computing and Imaging (SCI) Institute at the University of Utah, he was a Research Assistant Professor in the School of Computing at the same institution. Since 2008, he has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Utah. Dr. Tasdizen is also a Utah Science Technology and Research Initiative (USTAR) faculty member in the SCI Institute. His research interests are in image processing and pattern recognition with a focus on applications in biological and medical image analysis. He is an associate member of Bio Imaging and Signal Processing Technical Committee (BISP TC) of the IEEE Signal Processing Society. He is also an organizer for the International Workshop on Microscopic Image Analysis with Applications in Biology (MIAAB).