Image Denoising using Locally Learned Dictionaries

Priyam Chatterjee and Peyman Milanfar

Department of Electrical Engineering,
University of California, Santa Cruz, CA 95064, USA.

ABSTRACT

In this paper we discuss a novel patch-based framework for image denoising through local geometric representations of an image. We learn local data adaptive bases that best capture the underlying geometric information from noisy image patches. To do so we first identify regions of similar structure in the given image and group them together. This is done by the use of meaningful features in the form of local kernels that capture similarities between pixels in a neighborhood. We then learn an informative basis (called a dictionary) for each cluster that best describes the patches in the cluster. Such a data representation can be achieved by performing a simple principal component analysis (PCA) on the member patches of each cluster. The number of principal components to consider in a particular cluster is dictated by the underlying geometry captured by the cluster and the strength of the corrupting noise. Once a dictionary is defined for a cluster, each patch in the cluster is denoised by expressing it as a linear combination of the dictionary elements. The coefficients of such a linear combination for any particular patch is determined in a regression framework using the local dictionary for the cluster. Each step of our method is well motivated and is shown to minimize some cost function. We then present an iterative extension of our algorithm that results in further performance gain. We validate our method through experiments with simulated as well as real noisy images. These indicate that our method is able to produce results that are quantitatively and qualitatively comparable to those obtained by some of the recently proposed state of the art denoising techniques.

1. INTRODUCTION

Modern image acquisition hardware is capable of taking images with extremely high resolution and at high shutter speeds. Both these factors lead to relatively fewer photons being available to each sensor as a result of which the captured images are increasingly more prone to corruption by noise. Manufacturers then have to depend on image denoising algorithms to suppress the effects of noise and consequently image denoising has been a field of active research in the image processing community. The problem of denoising can mathematically be shown as estimating the pixel intensity z_i from the observation model

$$y_i = z_i + \eta_i, \tag{1}$$

where y_i is the observed pixel intensity of the i-th pixel after being corrupted by zero mean independent identically distributed additive noise η_i. The noise can be sampled from any particular distribution that need not be known apriori. Since many recent image processing algorithms[1–4] work on image patches, it is useful to formulate a patch-based observation model as

$$\mathbf{y}_i = \mathbf{z}_i + \boldsymbol{\eta}_i, \tag{2}$$

where $\mathbf{z}_i$ is the original image patch with the i-th pixel at its center written in a vectorized format and $\mathbf{y}_i$ is the observed vectorized patch corrupted by a noise vector $\boldsymbol{\eta}_i$. Denoising in such a case is the inverse problem of estimating the patch intensities.

One of the first denoising algorithms to perform acceptable restoration even in the presence of considerable noise was proposed by Portilla *et al.*[5] This wavelet domain denoiser was considered to be the state of the art until recently. Buades *et al.*[1] proposed a non-local denoising algorithm that takes advantage of the repetitive

Further author information: (Send correspondence to P. Chatterjee)
P. Chatterjee : E-mail: priyam@soe.ucsc.edu, Web: http://www.soe.ucsc.edu/~priyam
P. Milanfar : E-mail: milanfar@soe.ucsc.edu, Web: http://www.soe.ucsc.edu/~milanfar

Computational Imaging VII, edited by Charles A. Bouman, Eric L. Miller, Ilya Pollak, Proc. of SPIE-IS&T Electronic Imaging,
SPIE Vol. 7246, 72460V · © 2009 SPIE-IS&T · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.810486

structures in an image. The authors showed that by performing a weighted averaging of pixels with similar neighborhoods, considerable denoising performance can be obtained. Kervrann *et al.*[2] considerably improved on a localized version of this algorithm by introducing an iterative scheme that takes advantage of a better weight calculation mechanism along with the flexibility of adapting the support of the smoothing kernel based on the variance of the intensity estimate at each pixel location. Another denoising algorithm named BM3D[4] was proposed by Dabov *et al.* that works on a related idea of locating similar patches in an image. However, unlike the previous two methods, this patch based method performs denoising in the transform domain. Another effective spatial domain method, K-SVD,[3] was proposed by Aharon *et al.* where the image patches are assumed to be sparse representable. They make use of this assumption to learn an effective overcomplete dictionary for the entire image such that each patch can be represented as a linear combination of only a few of the dictionary elements. Another spatial domain method that performs quite well was proposed by Takeda *et al.*[6] Their iterative method of Steering Kernel Regression (SKR) first computes a data adaptive steering kernel that takes into account the similarity of image pixels in a neighborhood. Once the kernels are computed for every pixel, kernel regression is performed to obtain an estimate of the noise-free pixel intensity. Many recent denoising methods[1,2,7] can be shown to have a similar kernel regression structure where the methods vary widely in their weight calculation mechanisms.

In our present work, we propose a denoising framework through efficient representation of image patches by learned basis functions. We approach the problem of denoising by first explicitly segmenting the noisy image based on features that represent the underlying local image structure. For our work, we choose a normalized version of the local steering kernels of SKR[6] as the features. A suitable basis is then learned in each cluster such that all the patches in a particular cluster can be described effectively. The denoised estimate of pixel intensities are finally obtained in a local kernel regression formulation using the steering kernels and the locally learned dictionaries (hence the method name K-LLD). Our denoising framework is also motivated from unifying methods that perform denoising through dictionary learning[3] and those that can be formulated in a kernel regression framework.[1,2,7]

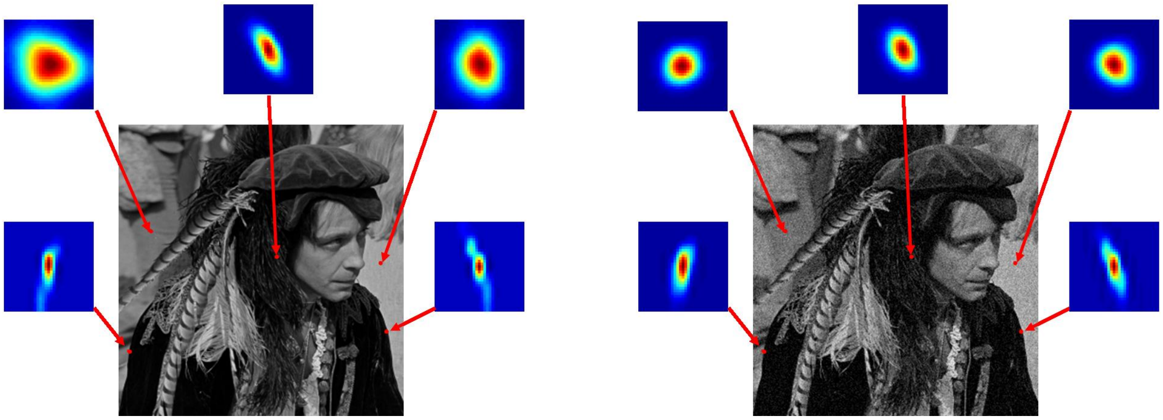
The remainder of the paper is outlined as follows: in Sec. 2 we briefly describe the SKR method since we choose the steering kernels as an indicator of local image geometry. We motivate and present our denoising framework in Sec. 3. Our method is then experimentally validated in Sec. 4. We finally summarize and conclude our work in Sec. 5.

2. STEERING KERNEL REGRESSION (SKR)

Kernel regression is a well studied method in statistics and signal processing. Recently, it was used to good effect to address image processing problems like denoising, interpolation and deblurring by Takeda *et al.*[6,8] While many image denoising methods[1,2,7] can be shown to have a corresponding kernel regression formulation,[9–12] the SKR method is distinguished by the way it forms the local regression weights. The weights used in each of the weighted averaging methods can be considered to be a measure of similarity of one pixel to another reference pixel. For localized methods, a local patch of such weights around each pixel (also known as *kernels*) needs to be estimated. In SKR, a robust estimate of the gradient is taken into consideration in analyzing the radiometric similarity of a reference pixel to other pixels in a neighborhood (patch). This then determines the shape and size of a canonical kernel (in particular, a Gaussian). The steering kernel for this particular case can be expressed as

$$w_{ij} = \frac{\sqrt{\det(\mathbf{C}_j)}}{2\pi h^2} \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{C}_j (\mathbf{x}_i - \mathbf{x}_j)}{2h^2}\right\} \tag{3}$$

where w_{ij} is the weight of the j-th pixel with respect to the i-th pixel, $\mathbf{C}_j$ denotes the symmetric gradient covariance matrix formed from the estimated vertical and horizontal gradients of the j-th pixel, $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^2$ denote the location of the i-th and the j-th pixels respectively and h is a global smoothing parameter also known as the bandwidth of the kernel. The weight w_{ij} is calculated for each pixel in a neighborhood $\mathcal{N}(i)$ with the i-th pixel at its center to form the weight matrix (or kernel). The vectorized version of this kernel we will denote by $\mathbf{w}_i = [\ldots w_{ij} \ldots]^T$ where $j \in \mathcal{N}(i)$. We refer the reader to the work of Takeda *et al.*[6] for a more in-depth explanation of the kernel formation process. A local kernel is obtained in this way for each pixel, providing a similarity measure of a pixel to its local neighborhood. A few such local kernels are shown in Fig. 1. It can be



(a) Man image

(b) Noisy image

Figure 1. Steering weights formed from (a) the original man image, (b) noisy image with AWG noise of std. dev 15. Note how the weights roughly represent the underlying image structure in the original image.

clearly seen how the kernels are representative of the underlying image structure. Furthermore, it can be seen that different locations in the image having different intensities but similar underlying structure still result in similarly shaped kernels.

For the SKR method of Takeda *et al.*,[6] the data is assumed to be smooth enough to be modeled locally as a polynomial of some low degree (usually 0, 1 or 2). We can then rewrite the data model of Eq. 2 as

$$\mathbf{y}_i = \mathbf{\Phi}\boldsymbol{\beta}_i + \boldsymbol{\eta}_i \tag{4}$$

where the dictionary $\mathbf{\Phi}$ is a matrix whose columns are formed from polynomial basis vectors and $\boldsymbol{\beta}$ is the vector of coefficients. Once the local weight matrices are formed using Eq. 3, the pixel intensity at each location is estimated in a local polynomial regression framework. This is done by solving the optimization problem

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_i &= \arg\ \min_{\boldsymbol{\beta}_i} \left(\mathbf{y}_i - \mathbf{\Phi}\boldsymbol{\beta}_i\right)^T \mathbf{W}_i \left(\mathbf{y}_i - \mathbf{\Phi}\boldsymbol{\beta}_i\right) \\ &= \arg\ \min_{\boldsymbol{\beta}_i} \|\mathbf{y}_i - \mathbf{\Phi}\boldsymbol{\beta}_i\|^2_{\mathbf{W}_i},\end{aligned} \tag{5}$$

where $\mathbf{W}_i = \text{diag}[\mathbf{w}_i]$. The weighted least squares problem of Eq. 5 has a solution in

$$\widehat{\boldsymbol{\beta}}_i = \left(\mathbf{\Phi}^T\mathbf{W}_i\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{W}_i\mathbf{y}_i. \tag{6}$$

The denoised estimate is then given by $\hat{z}_i = \mathbf{e}_1^T\widehat{\boldsymbol{\beta}}_i$ where pre-multiplication by the first column of the identity matrix ($\mathbf{e}_1$) results in retaining only the first elements of the $\widehat{\boldsymbol{\beta}}_i$ vector as the estimated intensity. Further details of that algorithm are explained at length in the original work by Takeda *et al.*[6]

3. DENOISING WITH LOCALLY LEARNED DICTIONARIES (K-LLD)

Although the SKR method is very effective, it suffers from two main limitations : a) the basis function is forced to be polynomial across the entire image; b) the order of regression is kept unchanged for the entire image. Our framework aims at removing such restrictions by allowing the dictionary as well as the regression order to be dictated by the image structure.

In order to generalize the SKR framework, we need to first learn dictionaries from the given image data such that they form useful descriptors of the image patches. Our algorithm works on the principle of first identifying patches with similar geometric structure and grouping them together to learn a dictionary that best describes the group. Once such a dictionary is learned for a particular group (or cluster), the coefficients (β) for the

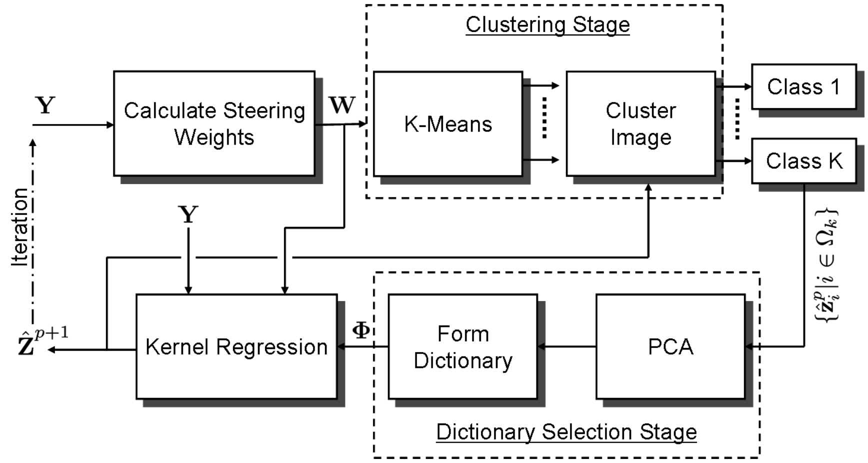


Figure 2. Block diagram of our iterative denoising framework.

linear combination is learned separately for each patch in a kernel regression framework. Our algorithm consists of three steps : the *clustering* step where patches with similar geometric structure are grouped together, the *dictionary learning* step where a dictionary is learned for each cluster, and the *coefficient calculation* step where the coefficients of the linear combination are calculated on a patch by patch basis. Now, we describe each step in detail.

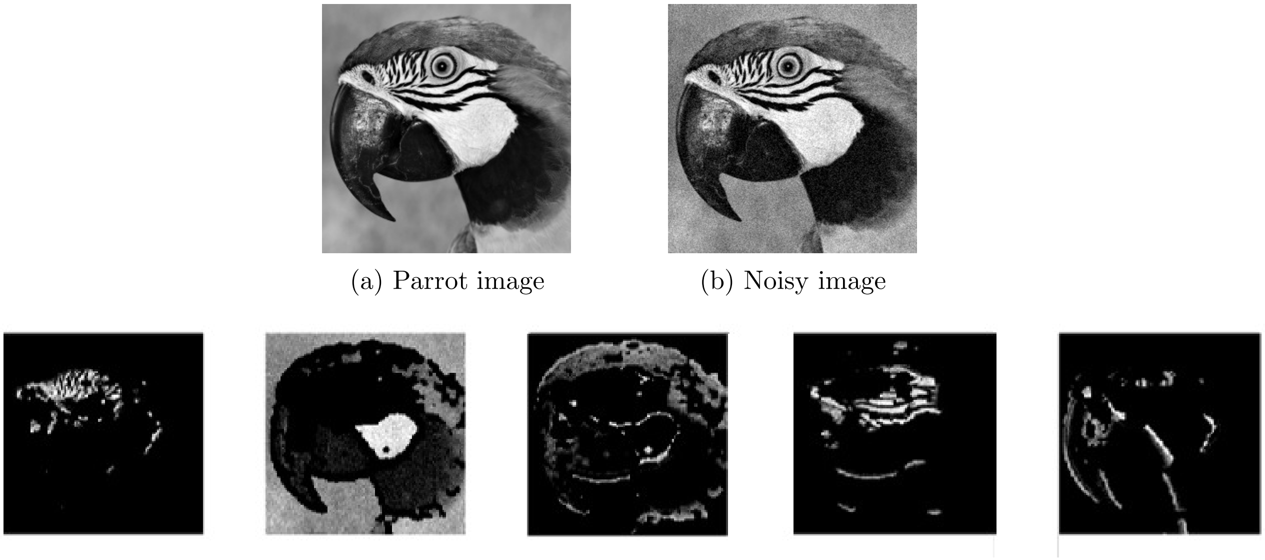
3.1 Clustering

The first step of our algorithm requires the image to be clustered into regions of similar geometric structure. In order to do this we need to first identify features that capture the underlying geometry of the image patches. As can be seen from Fig. 1, the steering kernels of SKR[6] align themselves to the underlying edge orientation and strength and can thus form good indicators of the underlying geometry. Moreover, the steering kernels exhibit a certain robustness in the presence of considerable noise, a property which is quite desirable. In order to make sure that the features are scaled equally in the feature space, we normalize the steering kernels to sum up to unity. These normalized vectors we will denote as $\check{\mathbf{w}}_i = \mathbf{w}_i / \sum_j w_{ij}$. An added advantage of these normalized vectors is that they are largely invariant to the mean patch intensities. We thus use these normalized steering kernels, in a vectorized form, as our feature vectors. For a patch size of $\sqrt{N} \times \sqrt{N}$, we obtain a feature vector of size $N \times 1$ for each of the pixels in the image.

Once the feature vectors are formed, we need a clustering method that can differentiate between the geometric descriptors and hence cluster the image. The K-Means algorithm[13] proves to be quite an effective method that suites our needs. Fig. 3 illustrates the performance of clustering a parrot image containing widely varying texture and smooth regions with our choice of features in conjunction with K-Means. It can be easily seen that patterns in a particular orientation are clustered together, even though they may have very different absolute intensities. This can be seen where the white cheek region is clustered together with other gray areas.

3.2 Dictionary Learning

Once the image is clustered into regions of similar geometric structure, we proceed to learn dictionaries for each of the clusters individually. An effective dictionary should be able to describe every patch in its corresponding cluster. However, the patches belonging to a particular cluster can have widely different patch intensities. Since our aim at learning a dictionary is to capture the geometric structure, we first *center* the patches in a particular cluster by subtracting from them the mean patch for that cluster. The dictionary is then trained to best describe these mean-subtracted patches in a mean squared error (MSE) sense. Mathematically, we solve the optimization



(a) Parrot image

(b) Noisy image

(c) Clustering of noisy image

Figure 3. Clustering using the steering weights as features. Notice how the clustering takes into account the geometric structure of the underlying image even in the presence of additive white Gaussian noise of standard deviation 15.

problem

$$\begin{aligned} \hat{\mathbf{\Phi}}^{(k)} &= \arg\min_{\mathbf{\Phi}^{(k)}} \sum_{i\in\Omega_k} \left\| \left(\mathbf{y}_i - \bar{\mathbf{y}}^{(k)}\right) - \hat{\mathbf{y}}_i \right\|^2 \\ &= \arg\min_{\mathbf{\Phi}^{(k)}} \sum_{i\in\Omega_k} \left\| \left(\mathbf{y}_i - \bar{\mathbf{y}}^{(k)}\right) - \mathbf{\Phi}^{(k)}\boldsymbol{\beta}_i \right\|^2 \end{aligned} \tag{7}$$

where $\bar{\mathbf{y}}^{(k)}$ is the mean vector of the set of patches in the k-th cluster, $\mathbf{\Phi}^{(k)}$ represents the dictionary for cluster Ω_k and $\boldsymbol{\beta}_i$ denotes a vector of coefficients for the linear combination of dictionary atoms to compute $\hat{\mathbf{y}}_i$. Calculating the optimal dictionary in this framework is non-trivial since the $\boldsymbol{\beta}_i$ vectors are also unknown. In order to circumvent this difficulty, we use the method of variable projections[14] where we first solve the optimization problem of Eq. 7 for the unknown vectors $\boldsymbol{\beta}_i$. The optimal $\boldsymbol{\beta}_i$ vector has a closed form solution in

$$\boldsymbol{\beta}_i = \left(\mathbf{\Phi}^{(k)T}\mathbf{\Phi}^{(k)}\right)^{-1} \mathbf{\Phi}^{(k)T}\tilde{\mathbf{y}}_i^{(k)}. \tag{8}$$

where $\tilde{\mathbf{y}}_i^{(k)} = \mathbf{y}_i - \bar{\mathbf{y}}^{(k)}$ is a mean subtracted patch in the k-th cluster. We further enforce our desired dictionary to be orthogonal in nature. This further reduces the expression of Eq. 8 to $\boldsymbol{\beta}_i = \mathbf{\Phi}^{(k)T}\tilde{\mathbf{y}}_i^{(k)}$. This is then plugged back into Eq. 7 to obtain an expression in a single unknown as

$$\begin{aligned} \hat{\mathbf{\Phi}}^{(k)} &= \arg\min_{\mathbf{\Phi}^{(k)}} \sum_{i\in\Omega_k} \|\tilde{\mathbf{y}}_i^{(k)} - \mathbf{\Phi}^{(k)}\mathbf{\Phi}^{(k)T}\tilde{\mathbf{y}}_i^{(k)}\|^2 \\ &= \arg\min_{\mathbf{\Phi}^{(k)}} \|\widetilde{\mathbf{Y}}^{(k)} - \mathbf{\Phi}^{(k)}\mathbf{\Phi}^{(k)T}\widetilde{\mathbf{Y}}^{(k)}\|_F^2 \qquad \text{such that} \quad \mathbf{\Phi}^{(k)T}\mathbf{\Phi}^{(k)} = \mathbf{I} \end{aligned} \tag{9}$$

where $\widetilde{\mathbf{Y}}^{(k)}$ denotes the data matrix whose columns consist of all mean subtracted vectorized patches ($\tilde{\mathbf{y}}_i^{(k)}$) from the k-th cluster and $\|.\|_F$ denotes the Frobenius norm. The optimization framework of Eq. 9 can be easily solved by a principal component analysis (PCA) on the centered data matrix $\widetilde{\mathbf{Y}}^{(k)}$. The optimal dictionary is thus formed by using the principal components as its columns. However, since the image patches contain noise, care must be taken to ensure that the dictionary does not overfit the data so as to retain the corrupting noise. To enforce such a constraint, we truncate the dictionary to contain only the most descriptive m_k principal components which satisfies

$$m_k = \max(m) \;\; \text{such that} \;\; \sum_{j=m+1}^{M} s_j^2 \geqslant rN\sigma^2 \tag{10}$$

where $s_1 \geqslant s_2 \geqslant \ldots \geqslant s_M \geqslant 0$ are the singular values of the data matrix $\widetilde{\mathbf{Y}}^{(k)}$, N is the number of pixels in each patch, r is a constant and σ is the standard deviation of the corrupting noise which is assumed to be known or can be estimated.[2,15] Once such an optimal dictionary is learned for each of the K clusters, each noise-suppressed patch in the image can be described as a linear combination of the columns of its corresponding dictionary as

$$\begin{aligned}\widehat{\mathbf{z}}_i &= \bar{\mathbf{y}}^{(k)} + \sum_{j=1}^{m_k} \hat{\boldsymbol{\phi}}_{(j)}^{(k)} \beta_{(j)i} \quad \forall \;\; i \in \Omega_k \\ &= \bar{\mathbf{y}}^{(k)} + \widehat{\boldsymbol{\Phi}}^{(k)} \boldsymbol{\beta}_i. \end{aligned} \tag{11}$$

3.3 Coefficient Calculation

At this point the coefficients of the linear combination need to be estimated. While it is possible to solve Eq. 11 for the optimal $\boldsymbol{\beta}_i$ vectors by directly using Eq. 9, we refrain from doing this. This is because while the clustering process is robust, it can by no means be assumed to be error-free, especially in the presence of strong noise. This can lead to at least some patches being wrongly clustered. The dictionary then may not be able to describe such patches effectively. This is especially true for patches that lie near the cluster boundaries. In order to deal with such errors, we take a more local approach to coefficient calculation by incorporating the (already available) steering kernels in the process as weights. Mathematically, we find the optimal $\boldsymbol{\beta}_i$ vectors such that

$$\widehat{\boldsymbol{\beta}}_i = \arg\min_{\boldsymbol{\beta}_i} \|\widetilde{\mathbf{y}}_i^{(k)} - \widehat{\boldsymbol{\Phi}}^{(k)} \boldsymbol{\beta}_i\|_{\mathbf{W}_i}^2 \quad \forall \; i \in \Omega_k \tag{12}$$

The optimal $\boldsymbol{\beta}_i$ vector that solves this weighted least squares problem is

$$\widehat{\boldsymbol{\beta}}_i = \left(\widehat{\boldsymbol{\Phi}}^{(k)T} \mathbf{W}_i \widehat{\boldsymbol{\Phi}}^{(k)}\right)^{-1} \widehat{\boldsymbol{\Phi}}^{(k)T} \mathbf{W}_i \widetilde{\mathbf{y}}_i^{(k)}. \tag{13}$$

The patches can now be reconstructed with the estimated $\widehat{\boldsymbol{\beta}}_i$ vectors using Eq. 11. However, one must keep in mind that the intensity estimate of each pixel in a reconstructed patch is true only in locations where the local steering kernel has a high value. This is to say that when the center pixel of a patch lies on one side of an edge, the patch estimate will not be able to predict with high confidence the denoised intensities for the pixels in the said patch that lie on the other side of the edge under consideration. To address this issue and considering that a patch estimate is obtained corresponding to each pixel in the image, we simply retain the pixel estimates for the center pixels of the reconstructed patches (where the local steering kernels have the highest weights). Thus, the denoised estimate at each pixel location is given by

$$\hat{z}_i = \mathbf{c}^T \widehat{\mathbf{z}}_i = \mathbf{c}^T \left(\bar{\mathbf{y}}^{(k)} + \widehat{\boldsymbol{\Phi}}^{(k)} \widehat{\boldsymbol{\beta}}_i\right). \tag{14}$$

where $\mathbf{c} = [0 \; \ldots \; 0 \; 1 \; 0 \; \ldots \; 0]^T$ is a vector consisting of all zeros except a 1 as its center element*. This allows us to get a denoised estimate for each pixel in the image. We next show that further increase in denoising performance can be achieved through an iterative scheme.

3.4 Iterative Denoising

Although the steering kernel calculation mechanism is fairly robust to the presence of noise, a strong noise can influence the process. Especially in the case where the underlying image contains very fine texture, the steering weights (and hence the features for the clustering stage) fail to truly represent the underlying geometric structure. This can be easily seen in Fig. 1(b) where, under the presence of considerable noise, the kernel from the texture

*For the concept of center pixel of a patch to make sense, we choose the patch dimensions $\sqrt{N} \times \sqrt{N}$ to be odd.

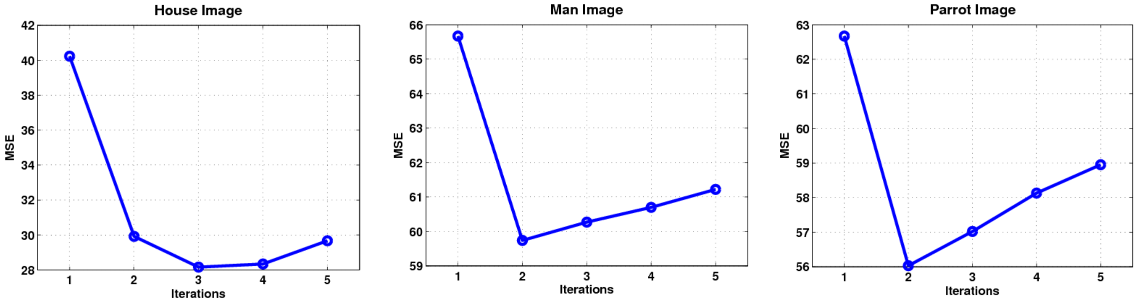


Figure 4. Illustration of how the MSE varies with iterations for AWG noise of standard deviation 15. The MSE of the input image was approximately 225 in each case.

region of the hair is largely indistinguishable from that of the smooth background. This may result in errors in clustering. The effect of noise present in the image patches and the clustering errors can then be propagated to the dictionary learning step, which in turn influences the final denoising estimate even though a local regression framework largely restricts the effects of clustering errors. An iterative framework for denoising is thus employed to better the performance of our algorithm. A block diagram of our iterative framework is illustrated in Fig. 2.

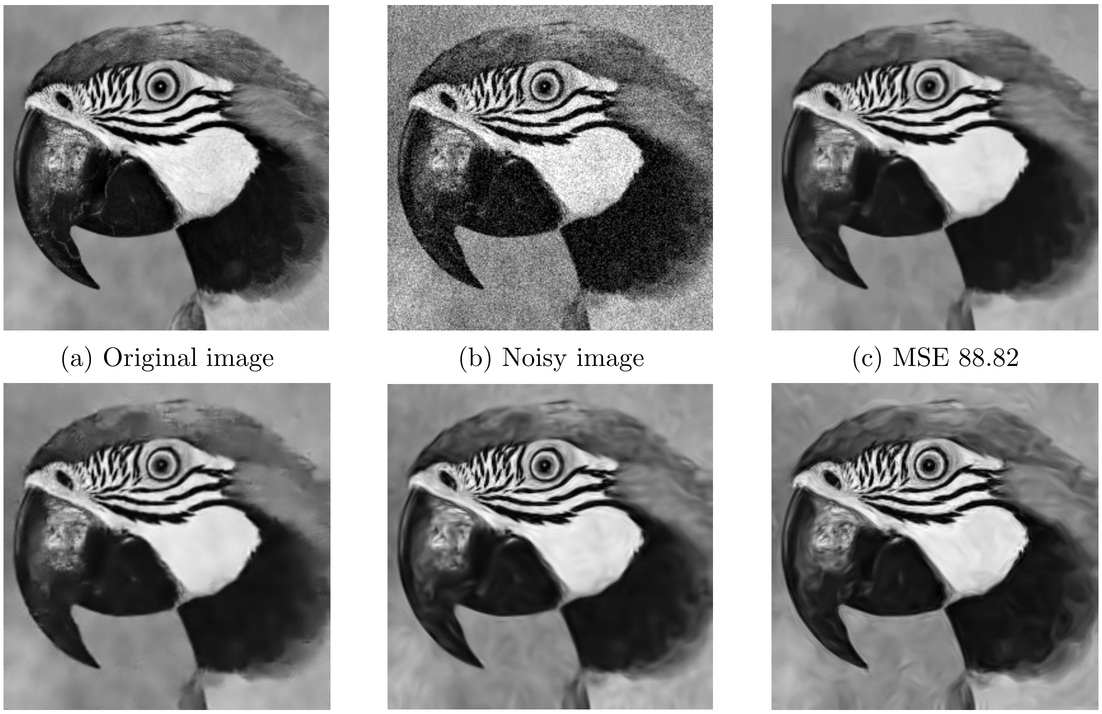
The merits of the iterating mechanism are quantitatively shown in Fig. 4 where a sharp decrease in MSE of the denoised output can be observed as we iterate. However, on iterating further, a reduction in performance can be seen due to the blurring effects on the output. The number of iterations needed to converge on the best result depends largely on the strength of the corrupting noise, and somewhat on the structural composition of the underlying image. The dependence on the latter can be seen in Fig. 4 where the somewhat smooth house image requires one more iteration to achieve the best denoised estimate, in terms of MSE, as compared to the more textured parrot image.

4. RESULTS

We validated our method of denoising by performing various experiments, with images corrupted with simulated noise as well as those where noise appeared in the actual image capturing process. First we explain our experimental setup which requires a few parameters to be tuned. For the simulated noise experiments, the parameters for our method were tuned to obtain the best possible results in terms of MSE. One of the important parameters to be tuned is the smoothing parameter for the kernel formation process. The optimal value was chosen heuristically. Another input needed by our algorithm is the number of clusters (K) for the clustering stage. The optimal number of clusters varies from image to image as it depends on the geometric composition of the image. However, our experiments have indicated that in most cases, choosing this parameter value to lie between 5 and 10 results in denoising output close to that obtained when K is tuned for the best results. Apart from this, we fixed the value of r that controls the number of elements in the dictionary in Eq. 10 to be 2.5 for all our experiments. Another choice that needs to be made by the end user is the size of the image patches. For our experiments, the patch size was chosen to be 13×13.

We first show results obtained by our method in the simulated noise case. For this, we created noisy images from original noise-free images. The noise was modeled to be zero mean additive white Gaussian. We compare the results obtained by our method to those obtained by some recently proposed methods like the K-SVD, SKR and BM3D in Fig. 5. Our result was obtained using $K = 10$ clusters and it needed 4 iterations to converge to the best result. It can be seen that our method improves on the performance of the SKR method and even more so when compared to the K-SVD technique. However, the state of the art transform domain BM3D method is seen to outperform our method in terms of MSE.

We further tested our method on images where the noise distributions were unknown and no ground truth was available. Moreover, the images are in color and the noise across the different color channels is likely correlated. We show in Fig. 6 that our method is able to perform effective denoising even for such a real world example. To deal with color images, we first transform the color image into the YCbCr color space. Then a rough estimate of



(a) Original image (b) Noisy image (c) MSE 88.82

(d) MSE 101.54 (e) MSE 99.96 (f) MSE 96.95

Figure 5. Comparison of denoising results on parrot image corrupted by additive white Gaussian noise of standard deviation 25. (a) Original image, (b) noisy image, (c) BM3D,[4] (d) K-SVD,[3] (e) iterative SKR,[6] (f) proposed method (K-LLD).

the noise variance is made heuristically for the Y (intensity) channel and the smoothing parameter is tuned for the best visual result. These parameters are then used in the other channels as well. The results obtained by our method are visually compared to those obtained by the BM3D and SKR methods. We also show the residual images which show the noise eliminated by the denoising process where presence of lesser structure indicates that the method gets rids of the noise without affecting the finer details of the image. It can be seen that our method compares well to these recently proposed methods in such cases as well.

5. CONCLUSION

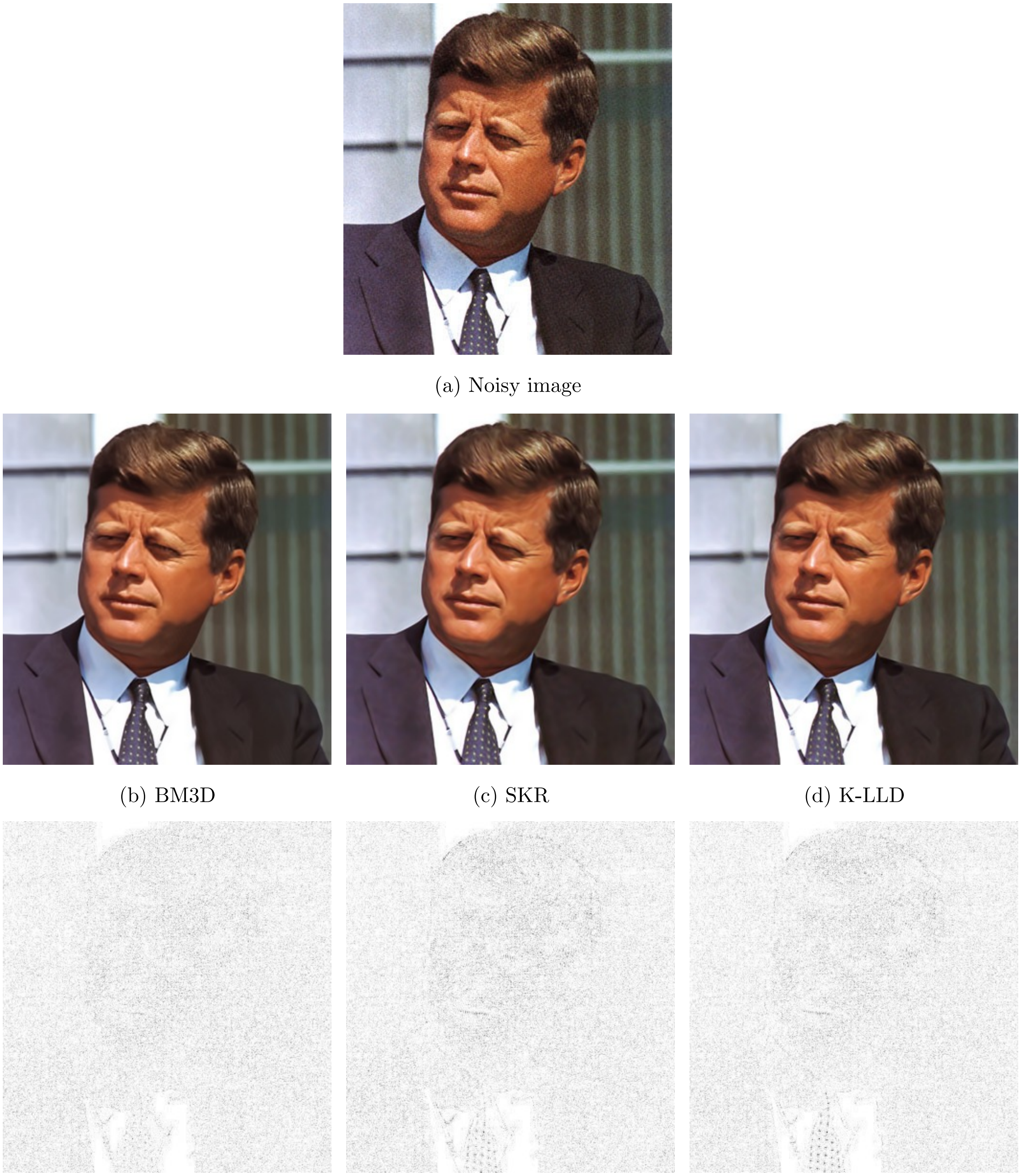
In this paper we have presented a patch based framework that performs denoising through geometric representation of images. In our work we first cluster the image into regions of similar geometric structure. To perform such a clustering, we make use of meaningful features in the form of normalized steering kernels to capture the geometric structure of patches. A dictionary is then learned for each cluster such that the dictionary of a particular cluster is able to effectively describe all the member patches. A denoised estimate for each pixel is then obtained in a kernel regression framework using the learned dictionary and the corresponding local steering kernel. The method can be easily iterated to obtain a gain in denoising performance. We have validated our work with experimental results that prove that our method is capable of performing denoising comparable to the state of the art methods, both in the case of simulated as well as real noise.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. Air Force under grant FA9550-07-1-0365.

REFERENCES

1. Buades, A., Coll, B., and Morel, J. M., “A Review of Image Denoising Methods, with a New One,” *Multiscale Modeling and Simulation* **4**(2), 490–530 (2005).



(a) Noisy image

(b) BM3D

(c) SKR

(d) K-LLD

(e) BM3D residual

(f) SKR residual

(g) K-LLD residual

Figure 6. Comparison of denoising results for a real noise case - JFK image.

2. Kervrann, C. and Boulanger, J., “Optimal Spatial Adaptation for Patch-Based Image Denoising,” *IEEE Transactions on Image Processing* **15**, 2866–2878 (October 2006).

3. Elad, M. and Aharon, M., "Image Denoising via Sparse and Redundant Representations over Learned Dictionaries," *IEEE Transactions on Image Processing* **15**, 3736–3745 (December 2006).
4. Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. O., "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Transactions on Image Processing* **16**, 2080–2095 (August 2007).
5. Portilla, J., Strela, V., Wainwright, M. J., and Simoncelli, E. P., "Image Denoising using a Scale Mixture of Gaussians in the Wavelet Domain," *IEEE Transactions on Image Processing* **12**, 1338–1351 (November 2003).
6. Takeda, H., Farsiu, S., and Milanfar, P., "Kernel Regression for Image Processing and Reconstruction," *IEEE Transactions on Image Processing* **16**, 349–66 (February 2007).
7. Tomasi, C. and Manduchi, R., "Bilateral Filtering for Gray and Color Images," in [*Proceedings of the Sixth International Conference on Computer Vision (ICCV)*], 839–846 (January 1998).
8. Takeda, H., Farsiu, S., and Milanfar, P., "Deblurring using Regularized Locally-Adaptive Kernel Regression," *IEEE Transactions on Image Processing* **17**, 550–563 (April 2008).
9. Takeda, H., Farsiu, S., and Milanfar, P., "Higher Order Bilateral Filters and their Properties," in [*Proceedings of the SPIE Conference on Computational Imaging V*], **6498**, 64980S (February 2007).
10. Buades, A., Coll, B., and Morel, J.-M., "The Staircasing Effect in Neighborhood Filters and its Solution," *IEEE Transactions on Image Processing* **15**, 1499–1505 (July 2006).
11. Chatterjee, P. and Milanfar, P., "A Generalization of Non-Local Means via Kernel Regression," in [*Proceedings of IS&T/SPIE Conference on Computational Imaging VI*], **6814**, 68140P (January 2008).
12. Seo, H. J. and Milanfar, P., "Video Denoising Using Higher Order Optimal Space-Time Adaptation," in [*Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*], 1249–1252 (April 2008).
13. Lloyd, S., "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory* **28**, 129–137 (March 1982).
14. Golub, G. H. and Pereyra, V., "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate," *SIAM Journal on Numerical Analysis* **10**(2), 413–432 (1973).
15. Liu, C., Szeliski, R., Kang, S. B., Zitnick, C. L., and Freeman, W. T., "Automatic Estimation and Removal of Noise from a Single Image," *IEEE Trans. on Pattern Analysis and Machince Intelligence* **30**, 299–314 (February 2008).