International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

Research Article

Improving Denoising Performance with Quality Enhancement Using Spatially Adaptive Iterative Filtering

Jainy Joseph†* and Anoop B.N†

†Department of Electronics and Communication, Mahatma Gandhi University, Kerala, India

Accepted 16 Nov 2015, Available online 26 Nov 2015, **Vol.5, No.6 (Dec 2015)**

Abstract

Digital image processing is a challenging domain of programming. There is always some noise present in all digital images. When we have to remove zero-mean white and homogeneous Gaussian additive noise from a given image, we address it as Image denoising problem. This is an important pre-processing task, for which spatial domain and transform domain image filters have achieved great success. However we cannot fine tune the denoising strength using spatial domain filters, but it can be efficiently done using shrinkage operators (in transform domain). In this work, we are proposing a novel approach for controlling the denoising strength using Spatially Adaptive Iterative Filtering (SAIF). The highlight of this technique is that we can automatically optimize the type of iteration and the iteration number w.r.t estimated risk using the plug-in risk estimator, after the adaptive iteration of filtering local image content with given base filter. Improved performance than often employed SURE estimator is the attracting characteristic of plug in estimator. Finally it is extended with guided filtering for quality improvement. Experimental results prove that SAIF with guided filtering improves denoising performance undoubtedly.

Keywords: *Spatial domain filters, denoising, SAIF, plug-in risk estimator, SURE, guided filtering*

1. Introduction

In our day to day lives, digital images play an exceptionally significant part being commonly used in handwriting recognition systems, validation of signatures, intelligent monitoring of traffic, satellite television and also in astronomy and geographical systems for the collection of geological data . Various noises and artifacts are introduced by the acquisition techniques and systems of digital imaging. Thus denoising plays a very vital role in the processing of images, its analysis and its various applications. By denoising, a lot of details of the image can be reserved and the random noise present is removed to a great extent. The presence of noise reduces the image's quality and also the visibility and perceptibility of images with lower contrast. Therefore by the process of denoising the images are enhanced and details hidden are recovered. In digital images, the noise found is additive in nature with power that is uniform in the entire bandwidth and has the normal (or Gaussian) distribution. This noise is known as Additive White Gaussian Noise (AWGN). The suppression of AWGN is a difficult process as it alters or manipulates all pixels of the image. In the process of removing noise a trade of has to be made between the suppression of noise and retaining the discontinuities in the image. A denoising technique thus has to be spatially adaptive by removing noise as well as not excessively smoothing important details of the image. Based on the noise model, a lot of different techniques are used. Sparsity, multi-resolution and edge detection are a few properties due to which the wavelets naturally cause spatially adaptive noise filtering.

Before further processing of images like segmentation, texture analysis etc, we always go for a preprocessing task called Image Denoising. This is due to the various noises that may arise in different inside and outside conditions which cannot be avoided. These noises must be removed at any cost since it causes degradation in visual quality of images. Here lies the importance of denoising algorithms. Ideally denoised images are noise free images. Since denoising depends on the images and noise model, we can say that it is problem specific technique. When the noise level of an image is too high, the denoising still remains as an open challenge. Hence to improve denoising performances researchers keep on consideration with it.

Normally, denoising algorithms are separated into two main classifications. They are named as Transform Domain Methods and Spatial Domain Methods. Transform domain methods are based on the assumption that the clean image can be well represented as a combination of few transform basis vectors. So the signal-to-noise-ratio (SNR) can be

*Corresponding author: **Jainy Joseph**

estimated and used to appropriately shrink the equivalent transform coefficients. Specifically speaking, if a basis element is detected as fitting to the true signal, its coefficient should be mostly preserved. Similarly, if an element is detected as a noise component, its coefficient should be shrunk more, or removed. By performing this, noise can be effectively crushed while most structures and finer details of the latent image are well-preserved.

Spatial domain methods focus on a different noise suppression approach. Here there is direct manipulation of pixels in an image. In many cases spatial domain methods yield undesirable results for the reason that, usually it enhances the whole image in a uniform routine. It's not possible to effectively and selectively enhance the edges and other essential information. In this method, each pixel value is estimated as a weighted average of other pixels, where higher weights are assigned to "similar" pixels. Pixel similarities can be considered in several ways. For the bilateral filter, similarity is determined by both geometric and photometric distances between pixels. Takeda et al. suggested a locally adaptive regression kernel (LARK) denoising method, robustly calculating the pixel similarity based on geodesic distance. Another effective method called non-local means (NLM) covers the bilateral filter by replacing point-wise photometric distance with patch distances, which is more robust to noise.

As images contain some tuning parameters that affect the performance, determining denoising strength is practically problematic. A larger smoothing parameter results in over smoothed output which erase certain information. A fewer smoothing makes slight denoising which cause suppression of noise. So iterative filtering is an alternative approach for boosting the spatial domain filters. By means of this iterative approach, by applying the same filter several times we can make a well estimated output which is considered as bad with that filter. For this we should find out the best iteration number and the best iteration method using the SAIF strategy. Then we apply the guided image filter to the SAIF output.

2. Related Works

We know that spatial domain filters deals with direct manipulation of pixel values while, transform domain filters deal with frequency content of an image. In the class of spatial domain and transform domain filters there exists a number of algorithms. Linear and Non-Linear filters are the two further classifications of spatial domain filters. Similarly the two key classifications of transform domain filters are Adaptive and Non-Adaptive transforms.

Various algorithms in this transform domain methods differs in either the transform selection or the shrinkage strategy. First we can have a look on transform selection. Transform domain method have the capability to represent both low frequency components and the high frequency transients. Transforms such as wavelet, DCT etc are often employed and are easy to compute or analyze. However, they may not be effective in representing natural image content with sparse coefficient distributions. This would certainly increase the requirement on the shrinkage performance. Principle component analysis (PCA) is another transform which is generally used. This is a kind of Non-Adaptive Transform. PCA is more adaptive to local image content when compared with adaptive transforms. This can lead to a more sparse coefficient distribution. Though, such decompositions can be fairly sensitive to noise. K-SVD and K-LLD are other techniques which is more robust to noise. This is because they use over-complete dictionaries generated from training. But they are computationally expensive.

The shrinkage strategy is another important factor that needs to be completely considered. The best strategy that gets close to the optimal performance with respect to mean-squared-error (MSE) is the Wiener criterion. This determines the shrinking strength according to estimated SNR in each basis element.

A traditional way to remove noise from image data is to employ spatial filters. In Spatial Domain techniques, the relative positions and the values of a local neighborhood of pixels are significant. Spatial domain consists of different methods such as Bilateral filtering, NLM, LARK etc.

Bilateral filter (BLT), smoothen the images by means of a nonlinear combination of adjacent image values. The method combines pixel values based on both their geometric closeness and their photometric similarity. This kernel can be expressed in a separable fashion as follows:

$$K_{ij} = \exp\left\{\frac{-\|x_i - x_j\|}{h_x^2} + \frac{-\|y_i - y_j\|}{h_y^2}\right\} \tag{1}$$

in which h_x *and* h_y are smoothing (control) parameters.

The NLM is another popular data-dependent filter which closely look like the bilateral filter except that the photometric similarity is captured in a patch-wise manner:

$$K_{ij} = \exp\left\{\frac{-\|\mathrm{x}_i - \mathrm{x}_j\|}{h_x^2} + \frac{-\|\mathrm{y}_i - \mathrm{y}_j\|}{h_y^2}\right\} \tag{2}$$

where $\mathbf{y}_i$ and $\mathbf{y}_j$ are patches centered at y_i and y_j, respectively.

More recently, the LARK (also called *Steering Kernel*) was introduced which exploits the geodesic distance based on estimated gradients:

$$K_{ij} = \exp\left(-\left(\mathrm{x}_i - \mathrm{x}_j\right)^T C_{ij}\left(\mathrm{x}_i - \mathrm{x}_j\right)\right) \tag{3}$$

in which $\mathbf{C}_{ij}$ is a local covariance matrix of the pixel gradients computed from the given data. The gradient is computed from the noisy measurements y_j in a patch around $\mathbf{x}_i$. Important advantage of LARK is the robustness to noise and perturbations of the data.

In the proposed method a SAIF approach is used, which is able to control the denoising strength. This is achieved by choosing the best iteration method and iteration number with respect to the calculated MSE using plug-in risk estimator. Then it iteratively filters using a given filter up to the iteration number and using the least risk iteration number selected. In order to make the output more vibrant we then applies the guided image filtering to the SAIF output.

3. Methodology

The image denoising strategy which is used here is can boost its performance by utilizing its spatially adapted transform and by employing an optimized iteration method. We will now discuss how we use iteration method and spatially adaptive transform for denoising. We do this by considering each patch. We don't do this globally.

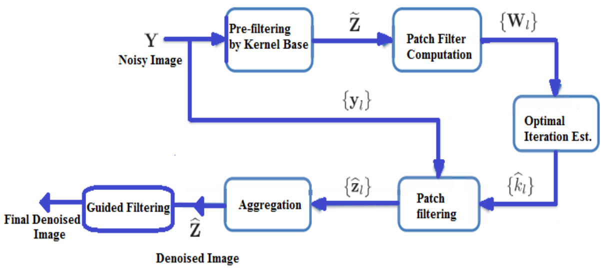


Fig.1 Block diagram of proposed method

For each patch, we apply our iterative filtering. We do this because we can calculate the signal to noise ratio locally patch wise. Depending on the iteration number, we get different qualities after the denoising is done. When our SAIF system gets the input noise images, it goes to two units before applying any modified filter on it. One of them is to calculate the optimal number of iterations. We preserve the same image and apply patch filtering after deciding the optimal number of iterations. We do all this by splitting the given noisy image into overlapping patches. We pass the image through a standard kernel baseline and use it to get the local filter. This is used in deciding the optimal iteration estimate. Iteration number is selected and filter patch are generated after this. There is one more final step called aggregation. This is done because the patches are overlapped. Fig.1 depicts the block diagram of the proposed approach.

3.1 Optimal iteration Estimation

This step is carried out to find the iteration number and to decide which iteration method is used. A method based on local signal to noise ratio is used to find the optimal number, called plug-in risk estimator.

Given a patch 'y' and its filter matrix 'W'. For each iteration method, the optimal stopping time $\hat{k}$ can be expressed as

$$\hat{k} = \arg\min_{k} MSE_k \tag{4}$$

Using the common technique SURE, an unbiased estimate of MSE can be computed. In this article an alternative method is proposed called plug-in risk estimator, which is biased and works based on an estimate of the local SNR. Practically, by using the base filter with some arbitrary parameter settings, eigen values and eigenvectors of the filter are estimated from a pre-filtered patch $\tilde{z}$. We have,

$$W(\tilde{z}) = VSV^T \tag{5}$$

Also, $W = VSV^T$ with $S^k = diag[\lambda_1^k \quad . \quad . \quad . \quad \lambda_n^k]$ where k is any non-negative real number. When implementing this, the filter can be applied with modified eigen values for any k > 0. This meaningfully improves the denoising performance as compared to when k is restricted to only positive integers. A real-valued k automatically and smoothly adjusts the local bandwidth of the filter.

The plug-in risk estimator and SURE estimator are explained below.

A. Plug-in risk estimator

Risk estimators for diffusion and boosting are computed based on the pre-filtered patch $\tilde{z}$. The signal coefficients can be estimated as

$$\tilde{b} = V^T \tilde{z} \tag{6}$$

The risk estimator should have some prior knowledge of the local SNR of the image. MSE_k in each patch can be estimated using the signal coefficients.

Diffusion Plug-in Risk Estimator:

$$Plug-in_k^{df} = \sum_{i=1}^{n} \left(1-\lambda_i^k\right)^2 \tilde{b}_i^2 + \sigma^2 \lambda_i^{2k} \tag{7}$$

Boosting Plug-in Risk Estimator:

$$Plug-in_k^{df} = \sum_{i=1}^{n} \left(1-\lambda_i\right)^{2k+2} \tilde{b}_i^2 + \sigma^2 \left(1-\left(1-\lambda_i\right)^{k+1}\right)^2$$

Algorithm:

Input: Noisy Patch: y, Pre-filtered Patch: $\tilde{z}$,Patch filter: W

Output: Denoised Patch: $\hat{z}$

1. Eigen-decomposition of the filter $W(\tilde{z}) = VSV^T$;
2. $\tilde{b} = v^T \tilde{z}$ - Compute the signal coefficients;
3. $Plug-in_k^{df}$, $Plug-in_k^{bs}$ - Compute the estimated risks;

4. If min{ $Plug-in_k^{df}$ } < min { $Plug-in_k^{bs}$ }
5. $\hat{k} = \arg\min_k Plug-in_k^{df}$ -Diffusion optimal iteration number;
6. $\hat{z} = VS^{\hat{k}}V^T y$ - Diffusion patch denoising;
7. else
8. $\hat{k} = \arg\min_k Plug-in_k^{bs}$ - Boosting optimal iteration number;
9. $\hat{z} = V\left(I-(I-S)^{\hat{k}+1}\right)V^T y$ -Boosting patch denoising;
10. end

In each patch, minimum values of $Plug-in_k^{df}$ and $Plug-in_k^{bs}$ as a function of k are computed and compared, and the iteration type with the least risk is chosen. Since the optimal iteration number $\hat{k}$ can be any real positive value, in the implementation of the diffusion filter, $W^{\hat{k}}$ is replaced by $VS^{\hat{k}}V^T y$ in which $S^k = diag\left[\lambda_1^k \quad . \quad . \quad . \quad \lambda_n^k\right]$

B. SURE estimator

The SURE estimator or MSE is defined for an estimate of the latent signal z, F(y) as:

$$SURE(y) = \|y - F(y)\|^2 + 2\sigma^2 div(F(y)) - n\sigma^2 \qquad (8)$$

If $F(y)$ is replaced by $W^k y$. It will act as a linear filtering framework. With this linear approximation we have: $divF(y) \approx tr(W^k)$. The SURE estimator for the diffusion process can be expressed as:

$$SURE_k^{df} = \left\|(I - W^k)y\right\|^2 + 2\sigma^2 tr(W^k) - n\sigma^2 \qquad (9)$$

After the eigen-decomposition of the filter, following equations are derived.

$$SURE_k^{df} = \sum_{i=1}^{n}\left(1-\lambda_i^k\right)^2 \tilde{b}_i^2 + 2\sigma^2\lambda_i^k - \sigma^2 \qquad (10)$$

$$SURE_k^{bs} = \sum_{i=1}^{n}(1-\lambda_i)^{2k+2}\tilde{b}_i^2 + 2\sigma^2\left(1-(1-\lambda_i)^{2k+2}\right) - \sigma^2$$

3.2 Aggregation

Since the patches are overlapped, for each pixel, there obtains multiple estimates. Therefore the final estimate of each pixel is obtained by aggregating all of these available estimates. Variance-based aggregation and Exponentially weighted aggregation are the two methods often employed for this aggregation tasks.

3.3 Guided Filtering

As a result of denoising, there are chances for blur near the edges of the denoised image. By considering the content of guidance image, we are computing the filter output in guided filtering. Thus, the edges of filter output are preserved and the output is not subjected to any gradient distortion.

4. Experimental Results

In this section various results of proposed method are analyzed. Firstly, a 256x256parrot image is given as input and its results with Non Local Means as given kernel is analyzed.

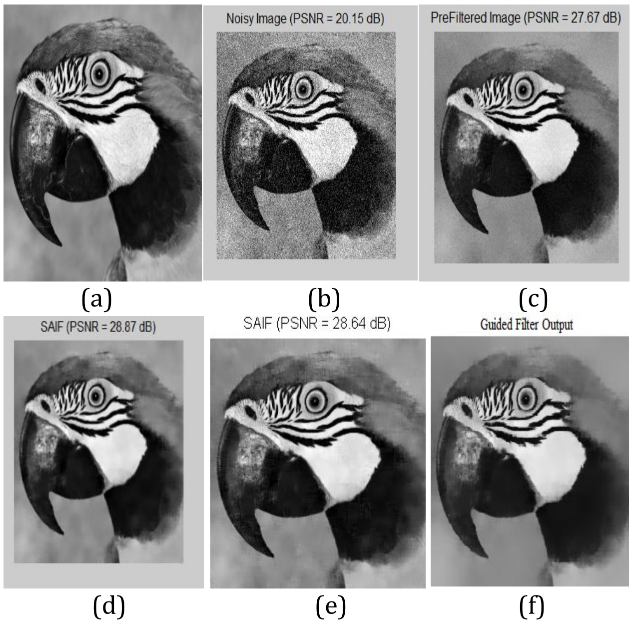


Fig.2 (a)Input Image (b)Noisy image (c)Pre Filtered image (d)SAIF output with Plug-in Estimator (e)SAIF output with SURE Estimator (f)Guided Filter Output

Here we have added zero mean white noise with variance 25 to the input parrot image (Fig.(a)) to obtain a noisy image with PSNR of 20.15db (Fig.(b)). With the given kernel (NLM as mentioned earlier) pre-filtering was done and we have got an output with PSNR of 27.67db (Fig.(c)). As we see pre-filtered output still have some noisy components with it. We performed our iterative SAIF method (Plug-in) for improved performance and this results in an output image with PSNR 28.87 db (Fig.(d)). For comparison purpose SURE output is as shown in Fig.(e). Plug-in thus shows an improved denoised result. The quality of our denoised result is again improved by following a guided filtering (shown in Fig.(f)).The quality was measured as 3.9275 for plug-in output image and is 4.6893 for guided filtered output.

Table 1 PSNR values for parrot image

σ	PreFiltering	SURE	Plug-in
25	27.67	28.64	28.87
15	30.21	31.21	31.33

Table 2 PSNR values for House image

σ	PreFiltering	SURE	Plug-in
25	28.24	30.76	30.81
15	32.66	33.77	33.85

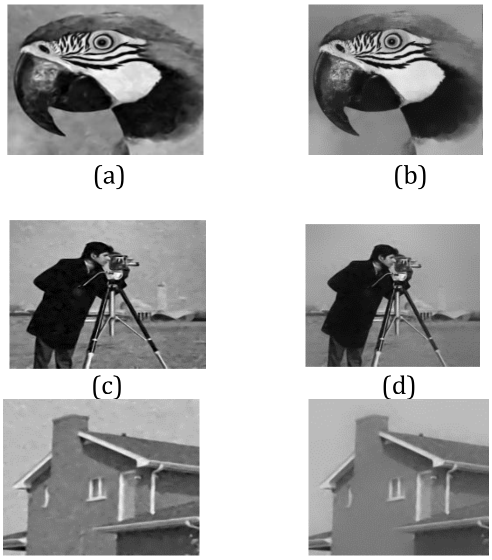
Now we can analyze the performance of two different images (ie. parrot and house) fortwo different variances with given kernel NLM(Table 1and Table 2). At a glance, we can say that the PSNR value of the plug-in method is higher than pre-filtered and SURE outputs. As the variance is reduced, the PSNR value of each stages of pre-filtering, SURE and Plug-in increases. But still plug-in gives the better denoised result.

Table 3 PSNR value with LARK kernel

	Image	PreFiltered	SURE	Plug-in
σ(25)	Parrot	27.60	28.15	28.26
	House	29.88	31.34	31.26

Above Table 3shows the performance of two different images (ie. parrot and house)with variance 25 for LARK kernel. The plug-in estimator shows consistent improvement over both the standard estimate using the kernel, and the optimally iterated kernel from SURE for `parrot' image. But for rich textured `house' image, SURE method outperforms the plug-in estimator.

Guided filtered images along with SAIF output for given LARK kernel (with Plug-in estimator) is shown below.



(a) (b) (c) (d) (e) (f)

Fig.3(a). SAIF output of parrot image (b). Guided output of Parrot image (c). SAIF output of Cameraman image (d). Guided output of Cameraman image (e). SAIF outputs of House image (f) Guided outputs of House image

The problem we observed in the plug-in estimator was that it shows less quality in the output images. Thus to solve this problem a guided filter was introduced to the system. The table below shows the quality analysis of various images for different kernels (BLT,NLM,LARK).

Table 4 Image quality analysis

Image	Kernel	SURE	Plug-in	Guided
Parrot	BLT	7.8234	6.4244	7.8873
	NLM	3.67	3.9275	4.6893
	LARK	3.5056	4.6636	5.3894
House	BLT	7.8123	8.3782	11.9847
	NLM	5.8465	8.5659	11.3408
	LARK	6.0674	6.1328	9.5143
Cameraman	BLT	7.1790	7.2707	8.1842
	NLM	5.1161	7.6213	9.5990
	LARK	4.3975	4.9872	7.6788

From above table, it is clear that by employing guided filtering the quality of denoised image have improved, as we expected. For every kernel, ie. BLT,NLM and LARK, a significant improvement in the quality have happened.

Conclusion

Denoising is considered as one of the fundamental challenges in the field of image processing. For engineers and scientists, it has been a permanent research topic. Here an improved denoising by data-dependent kernels is presented. SAIF strategy and guided image filtering performed this task. Patch wise iterative filtering is carried out here. Diffusion and Boosting are the two complementary iteration techniques that are used here. We select the best iteration method and iteration number according to the MSE value (risk value). The plug-in risk estimator used estimated local SNR as empirical prior knowledge of latent signal. Plug-in estimator outperforms the already existing SURE method in most of the cases. Better estimate of local SNR is the added feature of this method. To make the output more robust guided image filtering is employed. The performance of the output is determined based on Quality parameter of the image. It is a good and promising method. This can be effectively applied in the field of image processing since a promising improved result is guaranteed.

References

Hossein Talebi, Xiang Zhu, Peyman Milanfar (2013), How to SAIF-ly Boost Denoising Performance, *IEEE Trans. Imag. Process.,* Vol. 22, No. 4.

Kaiming, gian Sun, (2013), Guided Image Filtering, *IEEE Trans. Imag. Process,* Vol.22 No.4.

J. Portilla, V. Strela, M. Wainwright, and E. P.Simoncelli (2003), Image denoising using scale mixtures of Gaussians in the wavelet domain, *IEEE Trans. Imag. Process.*,Vol.12No.11.

H. Talebi and P. Milanfar (2012), Improving denoising filters by optimal diffusion, *Proc. ICIP,* pp. 1–4.

P. Chatterjee and P. Milanfar (2010), Is denoising dead? *IEEE Trans. Image Process.,* Vol. 19, No. 4, pp. 895–911.

J. Chen, S. Paris, and F. Durand (2007), Real-Time Edge-Aware Image Processing with the Bilateral Grid, *ACM Trans. Graphics* Vol. 26, No. 3

A. Buades, B. Coll, and J. M. Morel (2005), A review of image denoising algorithms, with a new one, *Multiscale Model. Simulat., Int. J.,* Vol. 4, No. 2, pp. 490–530.