ISSN 2091-2730

Boosting The Performance Of Color Image Denoising

Sreelekshmi G, Dr.Shyama Das

Department of Computer Science & Engineering,

College of Engineering, Chengannur, Kerala, India

sreelekshmi1989@gmail.com

Abstract— Spatial domain and transform domain image filters have achieved great success in denoising. The transform domain image filters is the leading one when compared to spatial domain image filters. The reason is that fine tuning of denoising strength can be done efficiently using shrinkage operators but not easy to do in spatial domain filters. Here Spatially adaptive iterative filtering in short SAIF to control the denoising strength locally for both gray images and color images for any spatial domain filters is implemented. Using this approach we iteratively filter the local image content using the given base filter, and with respect to the estimated risk using plug in estimator, and the type of iteration and iteration number are automatically optimized. Finally it is extended with bilateral image filtering. Experimental results prove that the SAIF plus Iterative Bilateral filtering improves the denoising performance.

Keywords— Image Denoising, Spatial Domain filters, Risk estimators, Pixel aggregation, SURE, Plug-In-Risk Estimators , Iterative Bilateral Filtering.

INTRODUCTION

With the rapid growth of the Internet now a days, the number of image data is increasing exponentially. In all these digital images there carries noise of some degree. Image denoising is an important image processing task, both as a process itself, and as a component in other processes. Image denoising plays a crucial job in a wide range of applications such as image restoration, visual tracking, image registration, image segmentation, and image classification, where acquiring the original image content is important for robust performance. Image Denoising is an important pre-processing task before further processing of image like segmentation, feature extraction, texture analysis, etc. As a result, there is degradation in visual quality of an image. So for removing these noises image denoising algorithms are used.

In general, denoising algorithms are divided into two main categories namely Transform Domain Methods and Spatial Domain Methods. Transform Domain Methods or Frequency Domain techniques are based on modifying the Fourier transform of the image. This technique is mainly suited for processing the image based on frequency content. Spatial domain techniques are based on direct manipulation of pixels in an image. In many cases spatial domain methods produce undesirable results because usually it enhances the whole image in a uniform manner. It’s not possible to effectively and selectively enhance the edges and other required information. In this method, it first estimates the pixel value as weighted average of other pixels where the higher weights are assigned to similar pixels.

Practically,determining denoising strength is difficult as it contains some tuning parameters that affect the performance. A larger smoothing parameter causes over smoothed output which erase some information. A less smoothing makes little denoising which cause suppression of noise. So for boosting the spatial domain filters , an alternative approach is iterative filtering. Using 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 the iteration number and the best iteration method should be found out using the SAIF[1] strategy. Then the bilateral iterative filtering is applied to the SAIF output. This can be implemented for both gray images and color images.

RELATED WORK

There were a number of algorithms exists in the category of spatial domain filters and Transform domain filters . The spatial domain filters directly deal with the pixel values and transform domain filters deal with the frequency content of an image. Spatial domain filters are further divided into two linear and non linear filters. The main two classifications of transform domain filters are adaptive and non adaptive transforms. Transform domain method have the ability to represent both low frequency component such as image backgrounds and the high frequency transients such as image edges. Wavelet , DCT etc are different methods in adaptive transform domain criteria. They are easy to calculate but representing the natural image content using the sparse coefficient distribution may not be effective. Non adaptive transforms can also be applied. Principle component analysis (PCA) is an example. When compared with the adaptive transforms PCA effectively represents the natural image content using sparse coefficient distribution and are sensitive to noise. Another methods K-SVD[6] and K-LLD[11] which is computationally expensive but they are more robust to noise because they use over complete dictionaries that are generated from training methods.

Spatial domain consists of different methods such as Bilateral filtering[7], NLM[12], LARK [5]etc. In bilateral filtering the pixel similarity is calculate by using the photometric and geometric distance between the pixels. Another method is Locally Adaptive Regression Kernel in short LARK, in which pixel similarity is calculated based on the geodesic distances and the approach is very simple. A further and one of the most successful method is Non Local Means (NLM). Actually NLM is an improvement of Bilateral filtering. NLM is implemented in our proposed system's prefiltering phase. NLM is more robust to noise. The difference of NLM with bilateral filtering is that it replaces the point wise photometric distance of bilateral filtering with patch distance. Practically, determining denoising strength is difficult as it contains some tuning parameters that affect the performance. A larger smoothing parameter causes over smoothed output which erase some information. A less smoothing makes little denoising which cause suppression of noise. So for boosting the spatial domain filters, an alternative denoising strength of spatial domain methods according to the calculated local SNR. The proposed method consists of SAIF strategy which is able to control the denoising strength of spatial domain methods 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 NLM upto the iteration number and using the least risk iteration number chosen. To make the output more vivid then applies the iterative bilateral filtering to the SAIF output.

PROPOSED METHOD

This method is proposed for boosting the performance of any spatial method. In this method a denoising strategy is described by employing an optimized iteration method. The proposed method is implemented for both gray images and color images. For a gray image, starting from the original image and add noise to the corresponding grayscale image in a controlled fashion. Then the corrupted image was obtained and the corrupted image can be subjected to all available filters in spatial domain methods. The corrupted image is then splitted into different overlapping patches and each patch is denoised separately. In order to calculate the local filter, first estimates the image using a standard kernel baseline. Here NLM is used . Boosting and Diffusion are the two iteration approaches used here. We then calculate the MSE for two different iteration approaches for each patch. By comparing the values method with least risk is chosen and consequently the iteration number is selected and the filtered patch is generated. The last step of the SAIF strategy is to aggregate these overlapping patches. Then a iterative bilateral filtering is also employed to make the output more vivid. The system architecture as follows. Each step is described briefly in below sections.

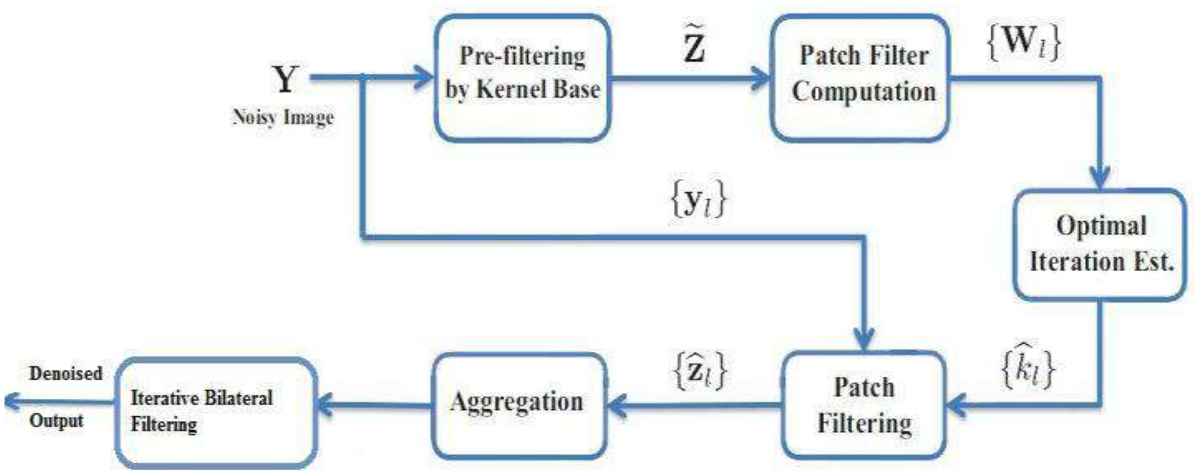

Figure A: Block Diagram of the Proposed System

From the figure itself, the proposed system consists of six steps.

1. Prefiltering
2. Patch Filter Computation
3. Optimal iteration estimation
4. Patch Filtering
5. Aggregation
6. Iterative Bilateral Filtering

1) Pre filtering

Before prefiltering noise is added to the original image and make it corrupted. Noise is added by changing the value of the variable sigma. For prefiltering we propose kernel methods. The kernel methods used in this works are Non-Local Means (NLM), Bilateral filter, LARK. The NLM is a very popular data-dependent filter which closely resembles the bilateral filter except that the photometric similarity is captured in a patch-wise manner that is average of all the pixels in the image. Non local means filter not only compares the grey level in a single point but also the geometrical configuration in a whole neighborhood and it is more robust than neighborhood filter. The general equation is:

$$K_{ij} = \exp\left\{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{h_x^2} + \frac{-\|\mathbf{y}_i - \mathbf{y}_j\|^2}{h_y^2}\right\} \quad (1)$$

The second filter used is Bilateral (BL) filter, its a non-linear filtering technique concept of Gaussian smoothing by weighting the filter coefficients with their corresponding relative pixel intensities. Pixels that are very different in intensity from the central pixel are weighted less even though they may be in close proximity to the central pixel. BL filter smoothes images by means of a nonlinear combination of nearby image values. This kernel can be expressed in a separable fashion as follows:

$$K_{ij} = \exp\left\{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{h_x^2} + \frac{-(y_i - y_j)^2}{h_y^2}\right\} \quad (2)$$

in which hx and hy are the smoothing parameters. The next method used here is LARK.. The LARK, also called Steering Kernel exploits the geodesic distance based on estimated gradients.

$$K_{ij} = \exp\left\{-(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{C}_{ij}(\mathbf{x}_i - \mathbf{x}_j)\right\} \quad (3)$$

2) Patch filter Computation

As per SAIF strategy to calculate the local filter, use an estimated image which is filtered by any of the standard kernel baseline. The pixels are represented through a matrix-vector multiplication form, by stacking the weight vectors together. Here the image is split into different overlapping patches, in which each patch is of same size. And the next step is make two different patches such as prefiltered projected patches and the noisy projected patches.

3) Optimal Iteration Method

Diffusion and Boosting are the two different possible iteration methods.

3.1) Diffusion

The idea of diffusion in image filtering was originally motivated by the physical principles of heat propagation and described using a partial differential equation:

$$\widehat{\mathbf{z}}_k = \mathbf{W}\widehat{\mathbf{z}}_{k-1} = \mathbf{W}^k\mathbf{y}. \quad (4)$$

Each application of **W** can be interpreted as one step of anisotropic diffusion with the filter **W**. Choosing a small iteration number *k* preserves the underlying structure, but also does little denoising. Minimization of MSE determines when is the best time to stop filtering. The overall MSE can be expressed by this equation:

$$\mathrm{MSE}_k = \|\mathrm{bias}_k\|^2 + \mathrm{var}(\widehat{\mathbf{z}}_k) = \sum_{i=1}^{n}(1-\lambda_i^k)^2 b_i^2 + \sigma^2\lambda_i^{2k} \quad (5)$$

3.2) Boosting

Although the classic diffusion filtering has been used widely, this method often fails in denoising image regions with low SNR. This is due to the fact that each diffusion iteration is essentially one step of low-pass filtering. In other words, diffusion always removes some components of the noise and signal, concurrently. The overall MSE can be calculated by this equation:

$$\mathrm{MSE}_k = \sum_{i=1}^{n}(1-\lambda_i)^{2k+2} b_i^2 + \sigma^2\left(1-(1-\lambda_i)^{k+1}\right)^2 \quad (6)$$

4) Patch filtering

In this method, risk estimators for diffusion and boosting are computed based on the prefiltered patch z, computed using the base filter with arbitrary parameters. More explicitly, the signal coefficients can be estimated. The optimized per-patch filtering can be expressed as choosing between one of these two iterations: Either Diffusion Plug-in Risk Estimator:

$$\text{Plug-in}_k^{df} = \sum_{i=1}^{n} (1 - \lambda_i^k)^2 \tilde{b}_i^2 + \sigma^2 \lambda_i^{2k}. \tag{7}$$

Or Boosting Plug-in Risk Estimator :

$$\text{Plug-in}_k^{bs} = \sum_{i=1}^{n} (1 - \lambda_i)^{2k+2} \tilde{b}_i^2 + \sigma^2 \left(1 - (1 - \lambda_i)^{k+1}\right)^2 \tag{8}$$

After calculating this, which will be implemented in Plug-in Risk estimator algorithm.
Based on this algorithm if the minimum value of plug-in risk estimator for diffusion is less than that of the minimum value of Plug in risk estimator of boosting and diffusion will be selected and vice versa. The algorithm is discussed below:

Algorithm 1: Plug-in Risk Estimator

Input: Noisy Patch: $\mathbf{y}$, Pre-filtered Patch: $\tilde{\mathbf{z}}$, Patch Filter: $\mathbf{W}$
Output: Denoised Patch: $\hat{\mathbf{z}}$

Eigen-decomposition of the filter $\mathbf{W}(\tilde{\mathbf{z}}) = \mathbf{V}\mathbf{S}\mathbf{V}^T$;
$\tilde{\mathbf{b}} = \mathbf{V}^T \tilde{\mathbf{z}} \Leftarrow$ Compute the signal coefficients;
Plug-in_k^{df}, $\text{Plug-in}_k^{bs} \Leftarrow$ Compute the estimated risks;
if $\min\{\text{Plug-in}_k^{df}\} < \min\{\text{Plug-in}_k^{bs}\}$
    $\hat{k} = \underset{k}{\text{argmin}}\ \text{Plug-in}_k^{df} \Leftarrow$ Diffusion optimal iteration number;
    $\hat{\mathbf{z}} = \mathbf{V}\mathbf{S}^{\hat{k}}\mathbf{V}^T\mathbf{y} \Leftarrow$ Diffusion patch denoising;
else
    $\hat{k} = \underset{k}{\text{argmin}}\ \text{Plug-in}_k^{bs} \Leftarrow$ Boosting optimal iteration number;
    $\hat{\mathbf{z}} = \mathbf{V}\left(\mathbf{I} - (\mathbf{I} - \mathbf{S})^{\hat{k}+1}\right)\mathbf{V}^T\mathbf{y} \Leftarrow$ Boosing patch denoising;
end

Figure 2: Plug in risk algorithm

5) Aggregation

As a result of the overlapped patches, multiple estimates are obtained for each pixel. It need to aggregate all of these estimates to compute the final estimate for each pixel. An exponentially weighted averaging is used for the plug-in estimator. The weighted averaging can improve the aggregation especially when the weights are estimated on the risk estimated with each estimate. In this framework a variance-based frame. aggregation is employed in SURE and an exponentially weighted averaging is used for the plug-in estimator.

6) Iterative Bilateral Filtering

Bilateral filtering smoothes images while preserving edges, by means of a nonlinear combination of nearby image values. The method is noniterative, local, and simple. It combines gray levels or colors based on both their geometric closeness and their photometric similarity, and prefers near values to distant values in both domain and range.

RESULTS

Figure 3: Implementation using SAIF and Bilateral Filtering.

ACKNOWLEDGMENT

I would like to thank to the HOD of CS department, Project Coordinator and all friends for their valuable support and suggestions for the fulfillment of this work.

CONCLUSION

Denoising is considered as one of the fundamental challenges in the field of image processing. For engineers and scientists, it has been permanent research topic. Here an improved denoising by data-dependent kernels is presented. Patch wise iterative filtering is carried out here. 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 iterative bilateral 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:

[1] Hossein Talebi, Xiang Zhu, Peyman Milanfar, “How to SAIF-ly Boost Denoising Performance”, IEEE transactions on image processing, vol. 22, no. 4, april 2013.

[2] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain,” IEEE Trans. Image Process., vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[4] D. D. Muresan and T. W. Parks, “Adaptive principal components and image denoising,” in Proc. IEEE Int. Conf. Image Process., Sep. 2003, pp. 101–104

[5] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Trans. Image Process.*, vol. 16,no. 2, pp. 349–366, Feb. 2007

[6] P. Chatterjee and P. Milanfar, “Clustering-based denoising with locally learned dictionaries,” *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, Jul. 2009.

[7] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Jan. 1998, pp. 836–846.

[8] C. Kervrann and J. Boulanger, “Optimal spatial adaptation for patch based image denoising,” *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[9] J. Boulanger, C. Kervrann, and P. Bouthemy, “Space-time adaptation for patch-based image sequence restoration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1096–1102, Jun. 2007

[10] L. Zhang, W. Dong, D. Zhang, and G. Shi, “Two-stage image denoising by principal component analysis with local pixel grouping,” *Pattern Recognit.*, vol. 43, pp. 1531–1549, Apr. 2010

[11] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[12] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Model. Simulat., Int. J.*, vol. 4, no. 2, pp. 490–530, 2005.