Modified Sigma Filter Using Image Decomposition

Mi-Ran Gu, Dae-Seong Kang
Department of Electronic Engineering
Dong-A University
Hadan 2-dong, Saha-gu
Busan, KOREA
dskang@dau.ac.kr http://nmc.donga.ac.kr

Abstract: This paper is a study on image noise reduction of modified sigma filter using image decomposition. Conventional sigma filter has been shown to be a good solution both in terms of filtering accuracy and computational complexity. However, the sigma filter does not preserve well small edges especially for high level of additive noise. In this paper, we propose here a new method using a modified sigma filter. In our proposed method the input image is first decomposed in two components that have features of horizontal, vertical and diagonal direction. Then, two components are applied HPF and LPF. By applying a conventional sigma filter separately on each of them, the output image is reconstructed from the filtered components. Added noise is removed and our proposed method preserves the edges from the image. Comparative results from experiments show that the proposed algorithm achieves higher gains, on average, 2.6 dB PSNR than the sigma filter and 0.5 dB PSNR than the modified sigma filter. When relatively high levels of noise added, the proposed algorithm shows better performance than two conventional filters.

Key-Words: Image de-noising, Noise Reduction, Modified Sigma Filter, Noise Estimation, Image Decomposition.

1 Introduction
Noise reduction is a very important processing step in all digital imaging applications. Moreover, even for the latest manufacturing technologies of the camera sensors, the noise level is still high. As a consequence, image de-noising is and will always be an important research topic.

The methods of existing noise reduction can be roughly categorized into the spatial filters and frequency domain filters. The spatial filters are linear filter such as average filter, Gaussian filter and non-linear filters such as median filter, sigma filter. The frequency domain filters are band-pass filter, notch filter [1],[2],[3],[4].

Among many algorithms that exist in the open literature, the sigma filter [4] is probably one of the simplest de-noising methods. Due to its simplicity, this filter represents a good choice for implementation in mobile devices. However, the edge preservation performance of the sigma filter is not good, especially for small image details with variance close to the variance of the additive noise. In order to improve the detail preservation of the sigma filter, other more sophisticated approaches were proposed. For instance the fuzzy filter proposed in [5],[6] uses some fuzzy estimates of the local derivative to perform directional filtering of the image. Although it’s good filtering performances this approach have the disadvantage of relative high complexity and a large number of parameters that must be setup. In addition to noise reduction using wavelet was proposed [7]. Another alternative, called hybrid sigma filter, was proposed in [8] for speckle noise reduction and showed improved performances compared with the Lee’s sigma filter. The hybrid sigma filter, however, does not address the problem of additive noise reduction which is the focus of our work.

In this paper, we propose here a new method using a modified sigma filter [9]. The threshold of the sigma filter uses the estimated standard deviation of the noise by block-based noise estimation using the adaptive Gaussian filtering [10],[11]. In our proposed method the input image is first decomposed in two components that have features of horizontal, vertical and diagonal direction. Then, two components are applied HPF and LPF. By applying a conventional sigma filter separately on each of them, the output image is reconstructed from the filtered components. Added noise is removed and our proposed method preserves the edges from the image.

The rest of the paper is organized as follows. Section 2 describes the noise estimation algorithm. The modified sigma filter is shown in section 3. Based on the observations from section 3, in section 4 the new filtering scheme is introduced. In section 5 comparative results obtained with the sigma filter and the modified sigma filter and the new proposed method are presented and section 6 concludes the paper.

2 Noise Estimation
Generally, noise estimation algorithms in the spatial domain are classified into two approaches: block-based and filtering-based (smoothing-based) [10].
2.1 Block-Based Noise Estimation
In block-based methods, images are tessellated into a number of $M \times N$ blocks. Standard deviations of intensity (measures of intensity variation) are computed for all the blocks and sorted. The block with the smallest standard deviation has the least change of intensity. Therefore, the block with the smaller standard deviation is the smoother the block. The intensity variation of a smooth block may be due to noise, in which the standard deviation of the block is close to that of the Gaussian noise added.

2.2 Filtering-Based Noise Estimation
In filtering-based methods, a noisy input image is filtered by a low pass filter. The standard deviation of the difference image between the noisy input image and its filtered image is computed. Olsen’s noise estimation algorithm [10] yields good estimates for large noise cases.

2.3 Block-Based Noise Estimation Using Adaptive Gaussian Filtering
Block-based noise estimation using adaptive Gaussian filtering [11] is based on block-based noise estimation, in which an input image is assumed to be contaminated by the additive white Gaussian noise and a filtering process is performed by an adaptive Gaussian filter. Coefficients of a Gaussian filter are selected as functions of the standard deviation of the Gaussian noise that is estimated from an input noisy image. For estimation of the amount of noise (i.e., standard deviation of the Gaussian noise), we split an image into a number of blocks and select smooth blocks that are classified by the standard deviation of intensity of a block, where the standard deviation is computed from the difference of the selected block images between the noisy input image and its filtered image. Block-based noise estimation using adaptive Gaussian filtering can be efficiently applied to noise reduction in commercial image- or video-based applications such as digital cameras and digital television (DTV) for its performance and simplicity. Fig. 1 shows the flowchart of the block-based noise estimation method using adaptive Gaussian filtering.

Each step of the block-based noise estimation method using adaptive Gaussian filtering is described as follows.

--First, this algorithm tessellates an input image into a number of $3 \times 16$ nonoverlapping image blocks ($b_{ij}$).

--Second, it computes the standard deviation ($\sigma_{ij}$) of intensity for each block $b_{ij}$ and finds the minimum standard deviation ($\sigma_{\text{min}}$). Because the block with the minimum standard deviation has less signal information than other blocks in the image, we select a number of blocks whose standard deviations of intensity are close to $\sigma_{\text{min}}$, in which the standard deviation of intensity is represented (quantized) by an integer. Quantization levels for the standard deviation can be arbitrarily chosen. In our study, the standard deviation is quantized to an integer for simplicity, where two different functions are used: floor and truncation functions.

\[
\text{Noisy image } I(i,j) \quad \downarrow \\
\text{Block tessellation (3x16, } b_{ij} \text{)} \quad \downarrow \\
\text{Compute } \sigma_{\text{min}} \quad \downarrow \\
\text{Select } B^* \quad \downarrow \\
\text{Gaussian filtering } H(B^*) \quad \downarrow \\
\text{Noise estimation } \hat{\sigma}_n
\]

Fig. 1. Block diagram of block-based noise estimation algorithm using adaptive Gaussian filtering

--Third, relatively homogeneous blocks are selected according to the quantized standard deviation of a block according to the following rules

\[
\text{if } \lfloor \sigma_{\text{min}} \rfloor = \langle \sigma_{\text{min}} \rangle, \quad B^* = \left \{ b_{ij} \mid \sigma_{ij} = \lfloor \sigma_{\text{min}} \rfloor \right \} \\
\text{else } B^* = \left \{ b_{ij} \mid \sigma_{ij} = \langle \sigma_{\text{min}} \rangle \right \}
\]

where $\lfloor \cdot \rfloor$ signifies the floor function, $\langle \cdot \rangle$ represents the round function, and $B^*$ denotes the selected set of blocks whose standard deviations of intensity are close to the standard deviation of the smoothest block, with the smallest standard deviation denoted by $\sigma_{\text{min}}$. Fig. 2 shows relatively homogeneous blocks $B^*$ selected by (1) and (2) when the standard deviation $\sigma_n$ of the Gaussian noise added is equal to five, in which black blocks are selected as homogeneous regions.
--In the fourth step, Gaussian filtering coefficients are computed. For Gaussian filtering, the mask size can be varied adaptively depending on the noise amount added in an input image: large (small) Gaussian mask is desirable for large (small) noise cases. In experiments, a $5 \times 5$ Gaussian filter $h(x, y)$ is used

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$ (3)

where $\sigma$ is set to $[\sigma_{\min}]$ or $\langle\sigma_{\min}\rangle$.

--In the fifth step, the blocks, with the standard deviation of intensity satisfying inequality in (1) or (2), are filtered by a Gaussian filter specified in (3). This step is similar to that in a filtering-based noise estimation method.

--Finally, we compute the standard deviation of the difference image between noisy and filtered images within selected blocks ($B^*$), which gives the estimated noise standard deviation ($\hat{\sigma}_n$).

3 Proposed Algorithm

In this section we introduce our modified de-noising method based on the modified sigma filter [9]. The input image is first decomposed in two components that have features of horizontal, vertical and diagonal direction. Then, two components are applied HPF and LPF. By applying a conventional sigma filter separately on each of them, the output image is reconstructed from the filtered components. Added noise is removed and our proposed method preserves the edges from the image. The block diagram of our proposed method is depicted in Fig. 3 where the blocks denoted as HPF_comp1 and LPF_comp2 perform a high-pass filtering on two components that have features of horizontal, vertical and diagonal direction. The blocks denoted as BNE Using AGF and the sigma filters are represented by the blocks Sigma Filter.

Our proposed filtering scheme can be described by the following steps:

--First, the input image is decomposed in two components that have features of horizontal, vertical and diagonal direction. Fig. 4 shows two decomposed components.

$$comp1 = y(i, j - 1) + y(i, j + 1) + y(i - 1, j) + y(i + 1, j)$$ (4)

$$comp2 = y(i - 1, j - 1) + y(i - 1, j + 1) + y(i + 1, j - 1) + y(i + 1, j + 1)$$ (5)
Computation of these differences transforms the diagonal monotonically increasing/decreasing regions of the input image into constant regions. Moreover this operation also preserves the diagonal edges from the input image.

--Fifth step, Apply a sigma filter separately on the four computed components \( y_{\text{HPF comp}}(i, j) \), \( y_{\text{LPF comp}}(i, j) \), \( y_{\text{HPF comp}}(i, j) \) and \( y_{\text{LPF comp}}(i, j) \) to obtain \( f_{\text{HPF comp}} \), \( f_{\text{LPF comp}} \), \( f_{\text{HPF comp}} \) and \( f_{\text{LPF comp}} \) respectively. This is done by the blocks Sigma Filter in Fig. 3.

\[
\begin{align*}
\delta_{k,l} &= \begin{cases}
1; & |y(k,l) - y(i,j)| \leq \Delta \\
0; & |y(k,l) - y(i,j)| > \Delta
\end{cases} \\
f(i,j) &= \frac{1}{2} \left( \sum_{k=i-m}^{i+m} \sum_{l=j-n}^{j+n} \delta_{k,l} y(k,l) + \sum_{k=i-m}^{i+m} \sum_{l=j-n}^{j+n} \delta_{k,l} \right)
\end{align*}
\]

where \((i, j)\) is the corresponding coordinates of the pixels and \((k, l)\) is the corresponding coordinates of the window. \(\Delta\) is the estimated standard deviation of the noise by block-based noise estimation using the adaptive Gaussian filtering.

The four sigma filters, in Fig. 3, necessitate estimation of the noise variance from the corresponding image component. This is done, in the implementation from Fig. 3, separately for the four components by the blocks denoted as BNE Using AGF. A simpler and faster implementation in which the noise variance is estimated just once and most of the processing steps are done in parallel is discussed in the sequel.

--Finally, reconstruct the output image from the filtered components as follows

\[
f(i,j) = \frac{1}{2} \left( f_{\text{HPF comp}} + f_{\text{LPF comp}} \right)
\]

### 4 Simulation Results and Discussions

In this section, we show the comparative performances of our proposed method and the standard sigma filter and the modified sigma filter. To this end, we selected three images ‘Lenna’, ‘Pepper’ and ‘Airplane’ and we added zero mean Gaussian noise with different variances to them. The original images were represented on 8 bits (values in the range [0, 255]). In experiments, a \(5 \times 5\) sigma filter is used.

To visually compare the performances of the three methods, in the left images of Fig. 5, we show parts of the processed images. Fig. 5 shows the three methods result on a noisy grayscale image ‘Lenna’ (512 \(\times\) 512). Fig. 5 (a) is the original image and Fig. 5 (b) is the pre-processed image which was applied the Gaussian noise of 25 standard deviations. Fig. 5 (c), 5 (d) and 5 (e) show the obtained image using the standard sigma filter, the modified sigma filter and the proposed method,
respectively. The effect of image smoothing and edge preservation can be seen in the left images of Fig. 5. The added noise and fine texture in Fig. 5 (e) have been mostly removed.

To graphically illustrate the smoothing effect, the right graphs of Fig. 5 display the comparison of five pixel-intensity profiles taken from the left images of Fig. 5, respectively. The five intensity profiles were drawn horizontally across Lenna’s forehead (marked with black lines in images). The smoothing effect on pixel intensity fluctuations is clearly shown in the right graphs of Fig. 5. Meanwhile, the sharp edges in the original image are largely preserved. The obtained graph using the proposed method is the most similar original graph. We note that our proposed method better preserve small details.
Evaluation factors for comparing the performance are MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio).

\[
MSE = \frac{1}{MN} \sum_{i=0}^{M} \sum_{j=0}^{N} [x(i, j) - \hat{x}(i, j)]^2
\]  

(12)

\[
PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{[dB]}
\]  

(13)

where \(x(i, j)\) is the original image and \(\hat{x}(i, j)\) is the filtered image. \(M, N\) are Horizontal and vertical size of the image.

PSNR between the filtered image and the original clean image, obtained with three algorithms are shown in Fig. 6. From these numerical values we clearly see that our proposed algorithm provide the higher PSNR especially for high levels of the additive noise.

<table>
<thead>
<tr>
<th>Standard deviation of noise</th>
<th>Sigma filter</th>
<th>Modified sigma filter</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>36.747</td>
<td>37.479</td>
<td>37.290</td>
</tr>
<tr>
<td>15</td>
<td>29.349</td>
<td>31.718</td>
<td>32.353</td>
</tr>
<tr>
<td>25</td>
<td>25.579</td>
<td>28.618</td>
<td>29.739</td>
</tr>
<tr>
<td>Average value</td>
<td>30.558</td>
<td>32.605</td>
<td>33.127</td>
</tr>
</tbody>
</table>

Table I shows the PSNR comparison of each algorithm according to the standard deviation of added Gaussian noise. Comparative results from experiments show that the proposed algorithm achieves higher gains, on average, 2.6 dB PSNR than the sigma filter and 0.5 dB PSNR than the modified sigma filter. When relatively high levels of noise added, the proposed algorithm shows better performance than two conventional filters.

5 Conclusion
This paper presents a new method using a modified sigma filter for image de-noising. The new algorithm has improved performances in terms of MSE and also preserves better the fine details of the processed image as opposed with the standard sigma filter and the modified sigma filter. Comparative results from experiments show that the proposed algorithm achieves higher gains, on average, 2.6 dB PSNR than the sigma filter and 0.5 dB PSNR than the modified sigma filter. When relatively high levels of noise added, the proposed algorithm shows better performance than two conventional filters.