A Modified Sigma Filter for Noise Reduction in Images

RADU CIPRIAN BILCU[1], MARKKU VEHVILAINEN[2]
[1,2]Multimedia Technologies Laboratory,
Nokia Research Center
Visiokatu 1, FIN-33720, Tampere
FINLAND
1 2

Abstract: - Noise reduction is a very important processing step in digital imaging and several different techniques were proposed in the open literature. Among them, 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 introduce a new sigma filter for noise reduction in images. We propose here a new method that first decomposes the input image in four components that are independently processed using a standard sigma filter. The output image is reconstructed from the filtered components. Comparative results between our approach and the sigma filter, on synthetic images and also on real images obtained with a camera phone are shown.

Key-Words: - Image de-noising, Image Restoration, Sigma filter, Noise estimation, Mean Squared Error.

1 Introduction

Due to the fast growing of the mobile devices market, there is an increased demand for high performance and robust image processing algorithms. More and more mobile imaging devices, such as camera phones and PDA's, incorporate several image processing algorithms and methods. Although, the manufacturing technologies of the CCD or CMOS sensors are very advanced, the snapped picture must always be digitally processed prior to storage or displaying. This is done in order to eliminate different distortions such as: noise, blur, geometrical and color distortions, to mention a few.

Digital images obtained by a camera phone are used for various purposes such as: sending multimedia messages (MMS), storage, printing, etc. As a consequence, there are several places where image processing algorithms can be incorporated. For instance when the images are used for MMS, image processing take place in the mobile device. Due to their limited processing power, the implemented algorithms must have a low computational complexity. On the other hand, when the snapped images are to be stored or printed, post processing can be done on a PC. In such case, usually the computational complexity and performances of the implemented algorithms can be much higher.

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. Among many algorithms, that exist in the open literature, the sigma filter [6] is probably one of the simplest de-noising method. 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] uses some fuzzy estimates of the local derivative to perform directional filtering of the image. Although its good filtering performances this approach have the disadvantage of relative high complexity and a large number of parameters that must be setup. Another alternative, called hybrid sigma filter, was proposed in [2] 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 a new modified sigma filter for additive noise reduction in images. In our proposed method the input image is first decomposed

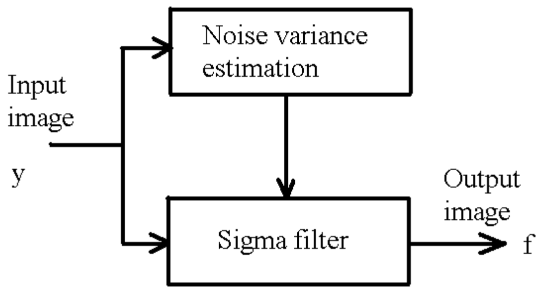

Figure 1: The block diagram of sigma filtering with noise variance estimation.

in four components and a sigma filter is applied separately on each of them. The output image is then reconstructed from the four filtered components.

The paper is organized as follows: in section 2 the standard sigma filter is described and its advantages and disadvantages are outlined. Based on the observations from section 2, in section 3 the new filtering scheme is introduced. In section 4 comparative results obtained with the sigma filter and the new proposed method are presented and section 5 concludes the paper.

2 Sigma Filter

In this section, we briefly review the standard sigma filter for additive noise [6] and outline its advantages and disadvantages. We assume the following model for the input image:

$$y(i,j) = x(i,j) + n(i,j). \tag{1}$$

where $y(i,j)$ is the observed image, $x(i,j)$ is the original clean image and $n(i,j)$ is a zero mean Gaussian distributed additive noise.

The main idea of the sigma filter is based on the fact that for a Gaussian distributed variable with mean μ and variance σ^2 a percentage of 95.5% of its samples lies inside the range $[\mu - 2\sigma, \mu + 2\sigma]$. Applying this observation to the model from (1), for every pixel $y(i,j)$ from the observed image, a local average is computed on those neighboring pixels that are inside the interval $[y(i,j) - 2\sigma, y(i,j) + 2\sigma]$. Corresponding pixel of the output image $f(i,j)$ is replaced with this local average.

This filtering scheme is based on the assumption that the pixel value $y(i,j)$ is a good estimate of the local mean and there are two issues that must be addressed. The first one is the selection of the neighboring pixels. Usually a rectangular $M \times M$ window centered at the current pixel is used for this (typically with M from 3 to 9). The second more important problem is the estimation of the additive noise variance σ_n^2. In practical applications the level of the additive noise is unknown therefore, some noise estimation method must be applied to the input image, prior filtering. For implementation in mobile devices, such methods must have low computational complexity and good estimation performances. In this paper we will use the approach in [3] that showed good estimation performance at a low computational complexity. As a consequence, a block diagram for de-noising based on sigma filter is depicted in Fig. 1. The input image is first passed through the noise estimation module and the estimated noise variance is then used in the sigma filter for de-noising.

It is well known, that small details of the input image are not well preserved by the sigma filter. This is due to the fact that on the regions from the input image that have variance close to the noise variance almost all pixels from the local $M \times M$ window are used in the average process. This effect is influenced mainly by the method of noise estimation but it can also be noticed when the exact noise variance is known. For instance when the estimated noise variance is larger than the real noise level the blurring effect is evident. An immediate solution is to decrease the length of the selection range to $[y(i,j) - \Gamma\sigma, y(i,j) + \Gamma\sigma]$ with $\Gamma < 2$. This modification reduces also the filtering capabilities of the sigma filter in smooth areas of the input image. Moreover, the average estimator is not a good choice for monotonically increasing/decreasing regions of the input image. In such regions, a better solution would be to use a higher order polynomial. In order to avoid the use of such polynomials to model the local monotonicity another alternative is to perform some transformation to the input image prior de-noising.

Taking into account these observations, in the next section we will introduce a simple modification that improves the performances of the sigma filter for regions of the input image that contain small details.

3 The Proposed Approach

In this section we introduce our modified de-noising method based on the observations outlined in the pre-

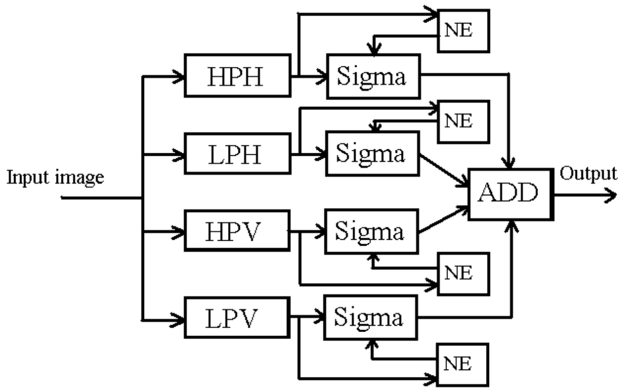

Figure 2: The block diagram of our proposed approach.

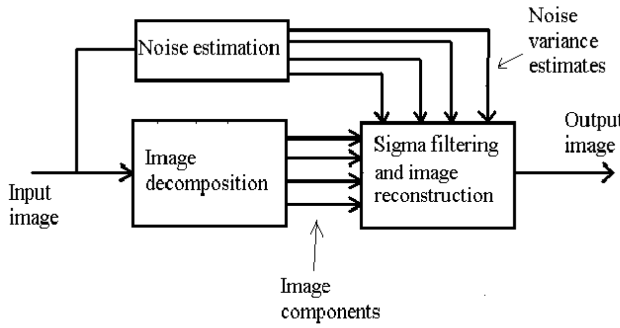

Figure 3: The block diagram of the fast implementation of our method.

vious section. The block diagram of our proposed method is depicted in Fig. 2 where the blocks denoted as HPV and HPH perform a high-pass filtering on the vertical and respectively horizontal directions. The blocks denoted as LPV and LPH perform a low-pass filtering on vertical and horizontal directions and the block denotes as ADD recombine the filtered components to obtain the restored image. Noise estimation is performed by the blocks denoted as NE and the sigma filters are represented by the blocks $Sigma$.

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

1. Compute the horizontal differences between adjacent pixels of the input image:

$$y_{HPH}(i,j) = \frac{1}{2}\left(y(i,j) - y(i,j-1)\right) \quad (2)$$

   where i and j are the corresponding vertical and horizontal coordinates of the pixels.

   This operation is performed by the block denoted as HPH in Fig. 2. Computation of these differences transforms the horizontal monotonically increasing/decreasing regions of the input image into constant regions (see [3] and the references therein). Moreover this operation also preserves the horizontal edges from the input image. Transformation of the monotonic regions into constant regions makes the simple averaging, performed by the sigma filter, a better model.

   The coefficient $\frac{1}{2}$ is introduced to preserve the dynamic range of $y_{HPH}(i,j)$. It does not influence the filtering accuracy and it can be discarded at this point. In such case, it must be included in the last step of the algorithm where the four components are combined to obtain the output image.

2. Compute the horizontal weighted sum of the adjacent pixels as follows:

$$y_{LPH}(i,j) = \frac{1}{2}\left(y(i,j) + y(i,j-1)\right) \quad (3)$$

   This operation is the complementary of the one in (2) and it is done primarily with the scope to reconstruct the original image. In addition to that, the edges of $y_{LPH}(i,j)$ are reduced and the filtering of this component can be made using a wider interval (larger Γ).

3. Compute the vertical differences and sums between adjacent pixels similar to (2) and (3) respectively.

$$y_{HPV}(i,j) = \frac{1}{2}\left(y(i,j) - y(i-1,j)\right) \quad (4)$$

$$y_{LPV}(i,j) = \frac{1}{2}\left(y(i,j) + y(i-1,j)\right) \quad (5)$$

4. Apply a sigma filter separately on the four computed components y_{HPH}, y_{LPH}, y_{HPV} and y_{LPV} to obtain f_{HPH}, f_{LPH}, f_{HPV} and f_{LPV} respectively. This is done by the blocks $Sigma$ in Fig. 2.

   The four sigma filters, in Fig. 2, necessitate estimation of the noise variance from the corresponding image component. This is done, in the implementation from Fig. 2, separately for

Figure 4: Parts of the: original image (up left), input noisy image ($\sigma_n^2 = 100$, up right), filtered image using the proposed algorithm (bottom left) and the result of sigma filter (bottom right).

Figure 5: Parts of the: original image (up left), input noisy image ($\sigma_n^2 = 300$, up right), filtered image using the proposed algorithm (bottom left) and the result of sigma filter (bottom right).

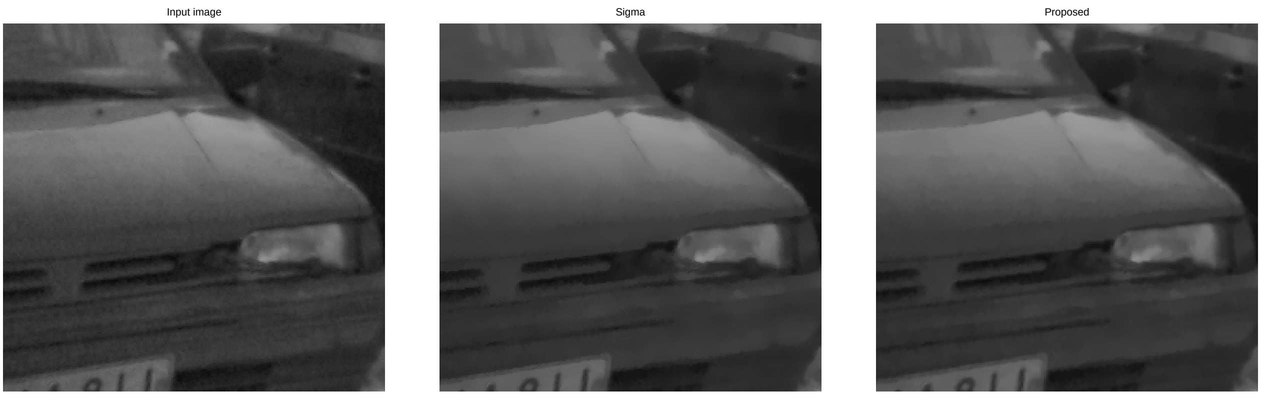

Figure 6: Parts of the: input image (left), filtered image using sigma filter (middle) and the result of our proposed method (right).

the four components by the blocks denoted as NE. 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.

5. Reconstruct the output image from the filtered components as follows:

$$f(i,j) = \frac{1}{2}\left(f_{HPH}(i,j) + f_{LPH}(i,j) + \right. \\ \left. + f_{HPV}(i,j) + f_{LPV}(i,j)\right) \quad (6)$$

Again the factor $\frac{1}{2}$ is included to preserve the dynamic range. We must emphasize here, that in the case the four components are not scaled in (2)-(5), this coefficient must be set to $\frac{1}{4}$ in (6).

3.1 Fast implementation

The block diagram from Fig. 2 might be too complicated for implementation into mobile devices with low processing power. From our experience, most of the processing time is spend on estimation of the noise variances from the four image components separately. Moreover, computing f_{HPH}, f_{LPH}, f_{HPV} and f_{LPV} independently, necessitate four scans of the complete input image.

A solution to highly decrease the running time is

Lena			Cameraman			Boats		
Input	Sigma	Proposed	Input	Sigma	Proposed	Input	Sigma	Proposed
10.06	7.66	6.77	9.98	6.0126	5.53	9.95	10.15	9.73
24.85	14.99	12.63	25.04	12.57	11.23	24.96	18.27	16.25
50.08	25.28	20.84	50.49	22.35	20.06	50.03	29.90	25.58
74.94	34.61	28.23	75.76	31.06	28.08	74.90	40.40	34.01
100.27	42.94	35.17	100.70	39.33	35.51	99.94	49.88	41.78
125.17	50.47	41.42	125.18	47.89	42.92	124.85	59.25	48.97
150.10	57.89	47.47	150.02	55.65	49.94	150.09	68.11	56.37
175.37	65.55	53.73	174.21	61.93	55.68	174.16	76.21	62.70
198.52	71.98	58.92	200.05	70.68	63.89	199.18	83.89	69.44
224.12	78.85	64.82	225.41	78.18	69.76	226.37	92.64	76.39
250.87	86.01	70.59	249.65	85.71	76.69	248.86	99.08	81.67
274.06	91.13	75.12	276.04	92.34	82.82	274.88	107.39	88.67
300.41	98.13	81.59	301.45	99.50	88.98	298.93	114.01	94.22

Table 1: MSE of the compared algorithms for several input images and noise variances.

first to implement image decomposition in a single block. Doing this parallel processing it is possible to compute the four image components in just one scan of the input image.

A great reduction in the processing time is then obtained if the noise variance is not estimated separately for the four image components but is done just once. This can be implemented at the beginning of the algorithm by estimating the noise variance σ_n^2 from the input image $y(i,j)$. The noise variances of the four components can be then obtained from σ_n^2 taking into account the linear operations performed by (2)-(5) as follows (see [3] and the references therein):

$$\sigma_{HPH}^2 = \sigma_{LPH}^2 = \sigma_{HPV}^2 = \sigma_{LPV}^2 = \frac{\sigma_n^2}{2} \quad (7)$$

More than that, sigma filtering and reconstruction of the output image from the filtered components can be done in just one scan of the image to further reduce the processing time. The block diagram of the fast implementation described above is depicted in Fig. 3.

4 Simulations and Results

In this section we show the comparative performances of our proposed approach and the standard sigma filter. To this end, we selected three images (lena, boats and cameraman) and we added zero mean Gaussian noise with different variances to them. The original images were represented on 8 bits (values in the range $[0, 255]$).

The Mean Squared Error (MSE), between the filtered image and the original clean image, obtained with both algorithms are shown in Table 1. From these numerical values we clearly see that our proposed algorithm provide the lower MSE especially for high levels of the additive noise.

To visually compare the performances of the two approaches, in Fig. 4 and Fig. 5, we show parts of the processed images. We note that our proposed method better preserve small details (for instance the fine details on Lena's hat in Fig. 4 and the fine details at bottom of the boats image).

Comparative results obtained with both algorithms on one image obtained with a Nokia cameraphone are shown in Fig. 6. Also here we can see better preservation of the fine details when the proposed de-noising method is applied.

5 Conclusions

In this paper we have introduced a new 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. A fast implementation with low computational complexity was also presented. The processing time of the new method is not much higher that that of the sigma filter and this makes it suitable for practical implementations on mobile devices. Further developments, of the new algorithm, with improved performances for signal dependent noise, are under consideration.

References:

[1] J. Astola, P. Kuosmanen, *Fundamentals of Non-*

linear Digital Filtering, New York: CRC Press, 1997.

[2] L. Alparone, S. Baronti, A. Garzelli, A Hybrid Sigma Filter for Unbiased and Edge-Preserving Speckle Reduction, in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium, IGARS '95*, Vol. 2, 1995, pp. 1409-1411.

[3] R. C. Bilcu, M. Vehvilainen, A New Method for Noise Estimation in Images, in *Proceedings of IEEE-EURASIP International Workshop on Nonlinear Signal and Image Processing, NSIP 2005*, May 18-20, 2005, Sapporo, Japan, pp: 290-293.

[4] A. Bosco, K. Findlater, S. Battiato, A. Castorina, A Noise Reduction Filter for Full-Frame Data Imaging Devices, in *IEEE Transactions on Consumer Electronics*, Vol. 49, No. 3, August, 2003, pp. 676-682.

[5] D. Van De Ville, M. Nachtagael, D. Van der Weken, E. E. Kerre, W. Philips, I. Lemahieu, Noise Reduction by Fuzzy Image Filtering, in *IEEE Transactions on Fuzzy Systems*, Vol. 11, No. 4, August, 2003, pp. 429-436.

[6] J. S. Lee, Digital Image Smoothing and the Sigma Filter, in *Computer Graphics Image Processing*, Vol. 24, 1983, pp. 255-269.

[7] K. Deergha Rao, M.N.S. Swamy, E.I. Plotkin, Adaptive Filtering Approaches for Color Image and Video Restoration, in *IEE Proceedings of Visual Image Signal Processing*, Vol. 150, No. 3, June, 2003, pp. 168-177.

[8] D.-H. Shin, R.-H Park, S. Yang, J.-H. Jung, Block-Based Noise Estimation Using Adaptive Gaussian Filtering, in *IEEE Transactions on Consumer Electronics*, Vol. 51, No. 1, February, 2005, pp. 218-226.

[9] Z. Ma, H. R. Wu, B. Qiu, A Window Adaptive Hybrid Vector Filter for Color Image Restoration, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004*, May 17-21, 2004, Vol. 3, pp. 205-208.