International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 1 (2012), pp. 23-33

http://www.irphouse.com

Estimation and Removal of Gaussian Noise in Digital Images

S. Suryanarayana, Dr. B.L. Deekshatulu[*], Dr. K. Lal Kishore [+] and Y. Rakesh Kumar++

Professor & Head, Dept of ECE, ACE Engg.College, Hyderabad, A.P., India
**Visiting Professor, Hyderabad Central University, Hyderabad, A.P., India*
+Director, R&D, JNTU, Hyderabad, A.P., India
++Asst. Prof., Dept of ECE, GNITS, Hyderabad, A.P., India

Abstract

In this paper a novel algorithm for Gaussian noise estimation and removal is proposed by using 3x3 sub windows in which the test pixel appears. The standard deviation(STD) for all sub-windows are used to define reference STD(σ_{ref})and minimum(σ_{min}) and maximum (σ_{max}) standard deviations. The average STD (σ_{avg}) is then calculated as the average of those STDs of all sub-windows whose STD falls with in the range of [$\sigma_{min,}$ σ_{max}]. This σ_{avg} is used for detecting and removing additive Gaussian noise. The performance is compared with that of the standard mean filter. The proposed scheme is outperforming than the standard mean filter.

Keywords: Additive Gaussian noise, standard deviation, sub-windows.

Introduction

Images can be contaminated with additive noise during acquisition and transmission. The two additive noises are Gaussian and impulse, the basic requirement in image denoising is to minimize this additive noise without affecting the features of the image. Normally non linear filters like median filters [1, 2] or rank order statistical filters [3, 4, 5] are more effective in eliminating impulse noise without effecting the edges information. Removing Gaussian noise involves smoothing the inside distinct region of an image. For this classical linear filters such as the Gaussian filter reduces noise efficiently but blur the edges significantly. For solving this problem nonlinear methods have been used by Perona and Malik[6], Bilateral filter were studied by Thomasi and Manducci[7]. In their methods authors used the local measure of an image to quantitatively detect edges and to smooth them less than the rest of the image. A universal noise removal filter presented in [8] based on simple statistics to detect impulse noise and is integrated to a filter designed to removal Gaussian noise. In [9] Total Least Square [TLS] is proposed by the authors for eliminating noise by modeling ideal image as a linear combination of image patches from the noisy image. The authors demonstrated the effectiveness of TLS algorithm even though it is computationally demanding. A directional adaptive central weighted median filter [DACWMF] is proposed in [10] works in wavelet domain with predefined thresholds applied separately to low frequency and high frequency bands. The authors mentioned that filtering is applied to only non edge regions of the image.

In this paper the authors' proposed new algorithm for estimating the presence of Gaussian noise and then effective minimization of the noise using a well defined static obtained from nine 3x3 sub windows in which the test pixel appears. Proposed method is outperforming than standard mean technique. The algorithm and simulation results are presented in following sections for different mean and variance combinations of additive Gaussian noise. To evaluate the performance of the proposed algorithm the parameter MAE, MSE and PSNR are used for varies standard images.

Proposed scheme

Estimation of Gaussian noise

Let 'X' is an original image, 'A' is observed image, and a general discrete time model for image degradation can be expressed as

$$A_{i,j} = X_{i,j} + \eta_{i,j}$$

For i,j = 1,2….N, where $X_{i,j}$ is original image pixels, $\eta_{i,j}$ is additive Gaussian noise and $A_{i,j}$ is the observed image. The objective of the restoration scheme is to recover the original image from the observed image. Here it is assumed the noise is normally distributed for a given mean and variance.

Let the noisy image is represented with A. the test pixel is located at (i,j), generally the 3 x 3 neighborhood is considered for normal filtering, whether corrupted or not. In our method we examined the 5 x 5 neighborhood of the test pixel in a different way. The 5 x 5 neighborhood is divided in to nine 3 x 3 sub-windows such that the test pixel appears in each of the sub-window. For each sub-window standard deviation (σ) [11, 12] is calculated. A reference standard deviation is decided as the median of the above sub-windows standard deviation (σ_i), i=1, 2 … 9, and two thresholds σ_{max} and σ_{min} are set and then average of the standard deviation (σ_{avg}) of nine sub windows whose standard deviation fall in the range [$\sigma_{min,}$ σ_{max}] is calculated. This average standard deviation is now used to estimate whether the pixel under test is corrupted or not. This is done based on the difference between mean of the 3x3 neighborhood of the test pixel (i, j) and the pixel value itself. The test pixel is decided corrupted based on the above difference lies within the range [a, b], otherwise treated as uncorrupted. Where the range limits 'a' and 'b' are experimentally obtained as 0.5 x σ_{avg} and 0.5, respectively. This is repeated for the entire noisy image. The detailed procedure is explained in the section 3.

Novel method to remove Gaussian noise

If the pixel is found corrupted then a filter is invoked. The corrupted pixel is replaced with a new value obtained from the following formula.

$$x_{new}(i, j) = [\mu - 0.5x\ \sigma avg\ ]$$

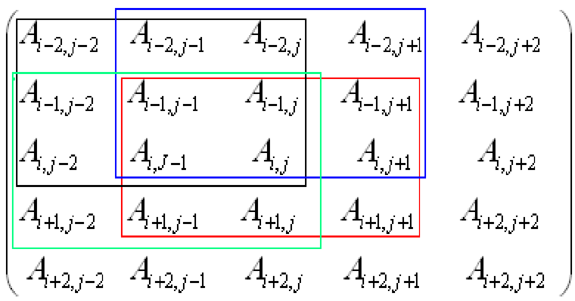
Where x_{new} is the new value for the pixel position represented by (i, j), μ is the mean of the 3x3 central subwindow, σavg is the average standard deviation defined in the previous section.

Algorithm

1. Consider a 5 x 5 test window A_T from the noisy image as:

$$A_T = \begin{pmatrix} A_{i-2,j-2} & A_{i-2,j-1} & A_{i-2,j} & A_{i-2,j+1} & A_{i-2,j+2} \\ A_{i-1,j-2} & A_{i-1,j-1} & A_{i-1,j} & A_{i-1,j+1} & A_{i-1,j+2} \\ A_{i,j-2} & A_{i,J-1} & A_{i,j} & A_{i,j+1} & A_{i,j+2} \\ A_{i+1,j-2} & A_{i+1,j-1} & A_{i+1,j} & A_{i+1,j+1} & A_{i+2,j+2} \\ A_{i+2,j-2} & A_{i+2,j-1} & A_{i+2,j} & A_{i+2,j+1} & A_{i+2,j+2} \end{pmatrix}$$

2. Divide this window into 3 x 3 sub-windows such that the test pixel should appear in each of the sub-windows. Nine such sub-windows are possible and four of them as shown below.



3. For each 3x3 sub-window calculate the standard deviation, $\sigma_{i,\ i=1,2\ \ldots\ldots N}$ where N is maximum number of the sub-windows, for this paper it is equal to 9.
4. Set reference standard deviation, (σ_{ref}), as median of $\sigma_{i,\ i=1,2\ \ldots\ldots N}$.
5. Set $\sigma_{min} = k_1 x\ \sigma_{ref}$.
6. Set $\sigma_{max} = k_2\ x\ \sigma_{ref}$.
7. Calculate average (σ_{avg}) of the standard deviations $\sigma_{i,\ i=1,2\ \ldots\ldots N}$ whose standard deviation lies in the range [$\sigma_{min}, \sigma_{max}$].
8. This σ_{avg} is used as a parameter to decide whether the test pixel is corrupted or not.

The above process is repeated by sliding 5x5 window one step forward row wise and then column wise to cover the entire image.

Results & Performance Evaluation

The algorithm developed is applied to detect and eliminate the Gaussian noise. The standard images like coins, peppers and Lenna for various means, i.e. 0, 0.01 and 0.05 and variances in the range of 0.01 to 0.06 for all means are used for performance evaluation.

The results for Lenna image are shown in Tables 1, 2 & 3, for means 0, 0.01 and 0.05 respectively. The comparative graphs are shown in figures 1, 2 & 3. The images are shown in fig: 4.The results obtained for Coins image for different combinations of mean and variances are shown in Table 4,5, & 6. The comparative graphs are shown in figures 5, 6 &7. The images are shown in fig: 8.

For peppers image the results are shown in Table 7, 8 &.9, for means 0, 0.01 and 0.05 respectively. The comparative graphs are shown in figures 9, 10 & 11. The images are shown in fig: 12.

The proposed algorithm of detecting and filtering Gaussian noise has proven to be excellent and far superior to standard mean filter. The improvement in PSNR for different variances and means are shown in the tables. It can be concluded from the tabular data that the proposed method is far better for a give mean and increasing variance than standard mean filtering method.

Table 1: Comparison of parameters (MAE, MSE, and PSNR) for Lenna with mean=0

Variance	Before filter			Standard filter			Detection and filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.039	0.005	71.18	0.045	0.003	73.22	0.027	0.002	76.09
0.02	0.055	0.01	68.22	0.059	0.005	71.09	0.033	0.002	75.022
0.03	0.068	0.015	66.39	0.072	0.007	69.59	0.037	0.0024	74.09
0.04	0.077	0.019	65.26	0.079	0.008	68.96	0.036	0.0024	74.33
0.05	0.087	0.024	64.28	0.087	0.01	67.98	0.039	0.0028	73.6
0.06	0.095	0.028	63.46	0.097	0.012	67.21	0.044	0.0034	72.84

Table 2: Comparison of parameters (MAE, MSE, and PSNR) for Lenna with mean=0.01.

Variance	Before filter			Standard filter			Detection and filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.035	0.004	68.22	0.04	0.003	71.09	0.026	0.0015	75.023
0.02	0.05	0.008	68.79	0.055	0.004	71.66	0.031	0.002	75.32
0.03	0.063	0.013	66.87	0.066	0.006	70.15	0.035	0.0023	74.46
0.04	0.073	0.018	65.59	0.074	0.007	69.42	0.034	0.002	74.83
0.05	0.083	0.023	64.49	0.084	0.009	68.39	0.038	0.003	73.95
0.06	0.089	0.027	63.82	0.091	0.011	67.71	0.041	0.003	73.32

Table 3: Comparison of parameters (MAE, MSE, and PSNR) for Lenna with mean=0.05.

Variance	Before filter			Standard filter			Detection and filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.034	0.0053	70.86	0.041	0.0028	73.63	0.027	0.0016	76.11
0.02	0.034	0.0053	70.87	0.041	0.0028	73.64	0.026	0.0016	76.13
0.03	0.045	0.009	68.64	0.054	0.004	72.07	0.029	0.0017	75.55
0.04	0.056	0.013	67.09	0.058	0.005	71.01	0.027	0.0018	75.1
0.05	0.065	0.017	65.84	0.066	0.0063	69.8	0.031	0.0019	74.38
0.06	0.073	0.021	64.93	0.075	0.008	69	0.034	0.0022	73.81

Table 4: Comparison of parameters (MAE, MSE, and PSNR) for coins with mean=0.

Variance	Before filter			Detection and filter			Standard filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.039	0.009	71.26	0.025	0.001	76.31	0.044	0.003	73.57
0.02	0.055	0.019	68.18	0.031	0.002	75.22	0.059	0.005	71.25
0.03	0.067	0.028	66.47	0.035	0.002	74.38	0.07	0.007	69.84
0.04	0.077	0.019	65.23	0.039	0.003	73.57	0.08	0.009	68.73
0.05	0.087	0.024	64.24	0.043	0.003	72.88	0.089	0.011	67.82
0.06	0.095	0.028	63.54	0.045	0.004	72.5	0.096	0.012	67.21

Table 5: Comparison of parameters (MAE, MSE, and PSNR) for coins with mean=0.01

Variance	Before filter			Detection and filter			Standard filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.035	0.004	71.92	0.023	0.001	76.54	0.039	0.003	74.17
0.02	0.05	0.009	68.8	0.028	0.002	75.58	0.054	0.004	71.9
0.03	0.064	0.014	66.79	0.033	0.002	74.57	0.067	0.006	70.22
0.04	0.073	0.018	65.64	0.036	0.003	73.94	0.075	0.008	69.13
0.05	0.082	0.023	64.58	0.04	0.003	73.33	0.085	0.009	68.26
0.06	0.091	0.027	63.76	0.043	0.004	72.69	0.093	0.012	67.49

Table 6: Comparison of parameters (MAE, MSE, and PSNR) for coins with mean=0.05

Variance	Before filter			Detection and filter			Standard filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.019	0.002	75.02	0.018	0.001	76.93	0.027	0.001	76.35
0.02	0.034	0.005	70.86	0.023	0.001	76.37	0.039	0.002	73.99
0.03	0.045	0.009	68.64	0.027	0.002	75.72	0.05	0.003	72.29
0.04	0.055	0.013	67.09	0.03	0.002	75.1	0.059	0.005	71.01
0.05	0.065	0.017	65.84	0.034	0.002	74.38	0.068	0.007	69.8
0.06	0.073	0.021	64.93	0.036	0.003	73.81	0.076	0.008	69

Table 7: Comparison of parameters (MAE, MSE, and PSNR) for peppers with mean=0

Variance	Before filter			Detection and filter			Standard filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.039	0.005	71.25	0.022	0.001	78.05	0.041	0.0025	74.2
0.02	0.055	0.01	68.24	0.027	0.0015	76.46	0.057	0.004	71.66
0.03	0.067	0.015	66.48	0.032	0.002	75.25	0.069	0.006	70.07
0.04	0.078	0.019	65.21	0.041	0.003	73.4	0.081	0.009	68.59
0.05	0.087	0.024	64.26	0.044	0.003	72.83	0.089	0.011	67.79
0.06	0.096	0.029	63.46	0.047	0.004	72.24	0.098	0.013	67.02

Table 8: Comparison of parameters (MAE, MSE, and PSNR) for peppers with mean=0.01

Variance	Before filter			Detection and filter			Standard filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.034	0.004	71.93	0.019	0.001	78.38	0.037	0.002	74.9
0.02	0.051	0.002	68.7	0.026	0.0013	76.85	0.053	0.004	72.22
0.03	0.063	0.013	66.89	0.029	0.0017	75.8	0.064	0.006	70.63
0.04	0.07	0.018	65.64	0.038	0.002	73.87	0.075	0.008	69.13
0.05	0.08	0.022	64.63	0.042	0.003	73.09	0.084	0.009	68.18
0.06	0.09	0.027	63.76	0.045	0.004	72.63	0.093	0.012	67.43

Table 9: Comparison of parameters (MAE, MSE, and PSNR) for peppers with mean=0.05

Variance	Before filter			Detection and filter			Standard filter		
	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR
0.01	0.019	0.002	75.029	0.016	0.0008	79.23	0.024	0.001	77.67
0.02	0.034	0.005	70.89	0.019	0.001	78.29	0.037	0.002	74.76
0.03	0.045	0.008	68.65	0.023	0.0012	77.23	0.048	0.003	72.76
0.04	0.056	0.013	67.05	0.033	0.002	74.84	0.06	0.006	70.76
0.05	0.064	0.017	65.92	0.035	0.0024	74.29	0.068	0.007	69.81
0.06	0.074	0.021	64.89	0.038	0.0028	73.7	0.076	0.008	68.84

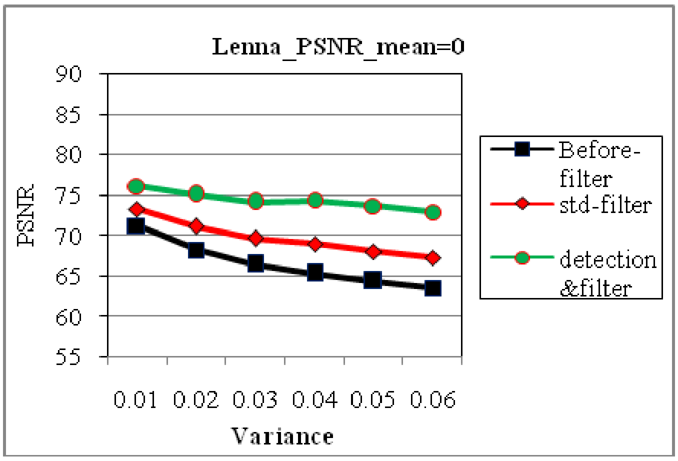


Figure 1: Graph of PSNR for Lenna image with mean=0.

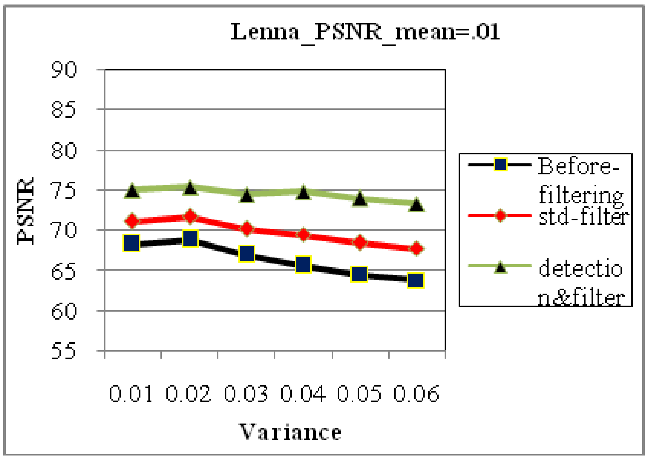


Figure 2: Graph of PSNR for Lenna image with mean=0.01.

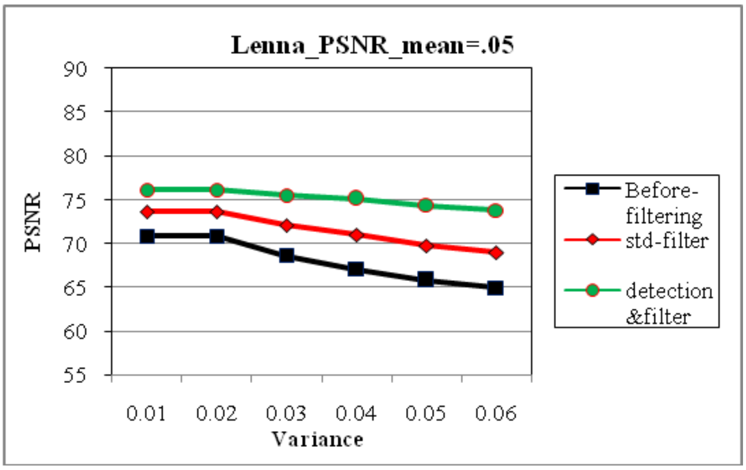


Figure 3: Graph of PSNR for Lenna image with mean=0.05.

Figure 4: (a) Original, (b) '0' mean '0.03' variance Gaussian noise, (c) Proposed Detection & filtered, (d) Standard filtered of Coins image.

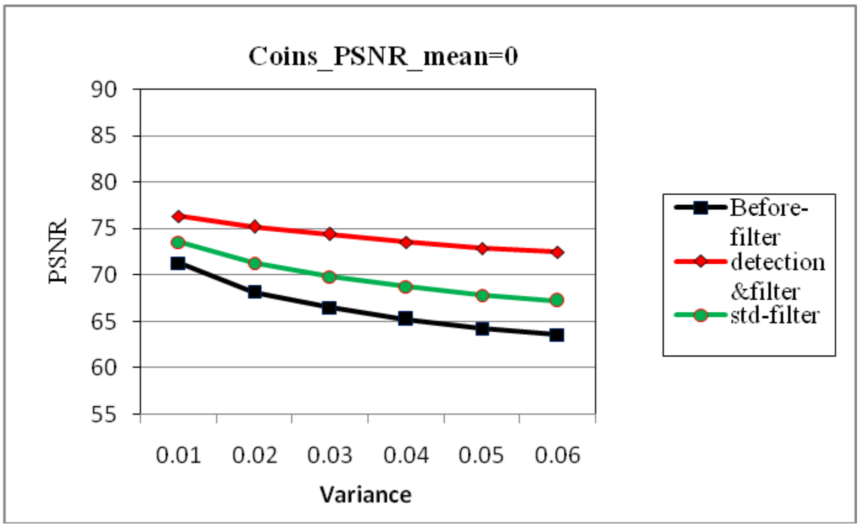


Figure 5: Graph of PSNR for coins image with mean=0.

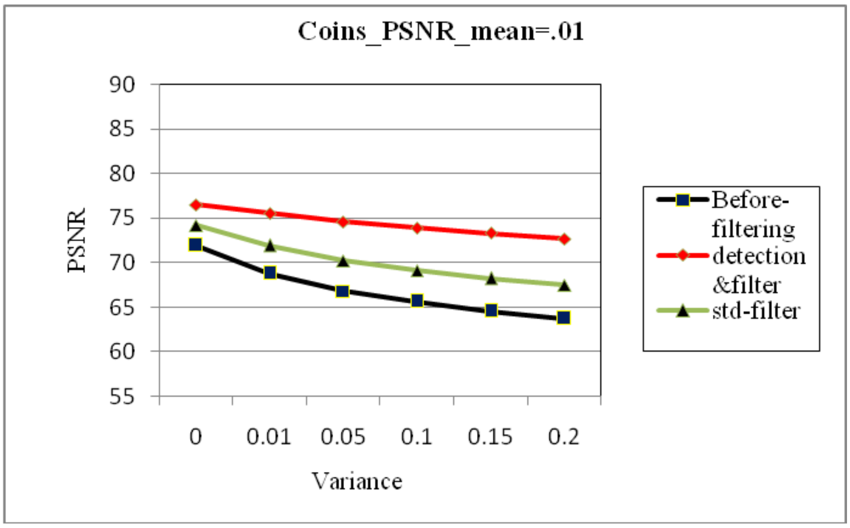


Figure 6: Graph of PSNR for coins image with mean=0.01.

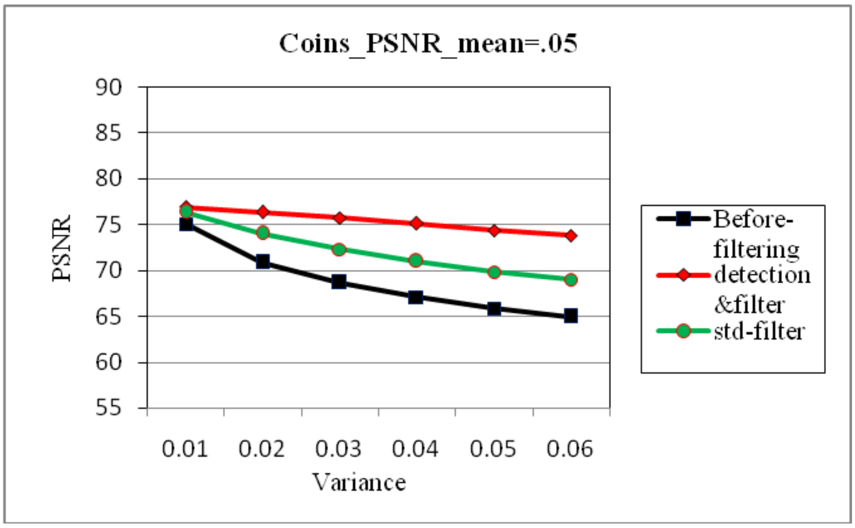


Figure 7: Graph of PSNR for coins image with mean=0.05.

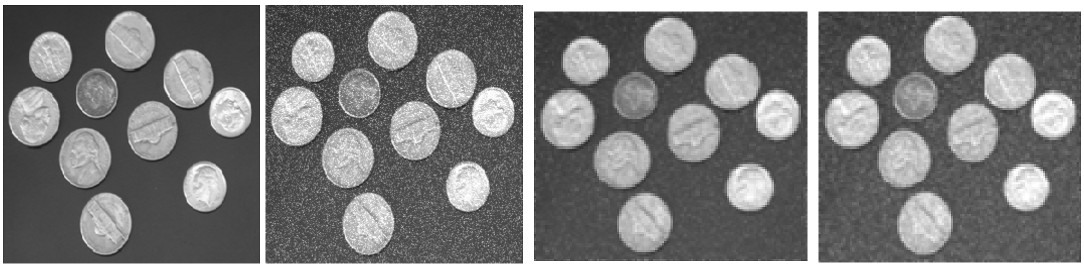


Figure 8: (a) Original, (b) '0' mean '0.03' variance Gaussian noise, (c) Proposed Detection & filtered, (d) Standard filtered of Coins image

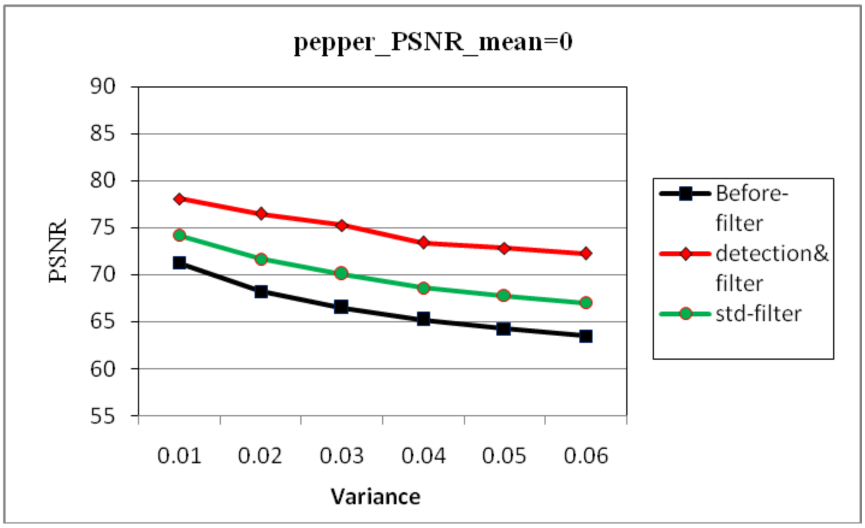


Figure 9: Graph of PSNR for peppers image with mean=0.

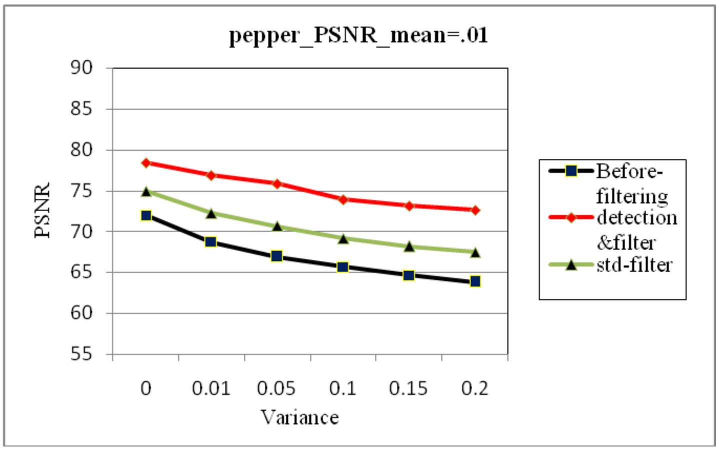


Figure 10: Graph of PSNR for peppers image with mean=0.01.

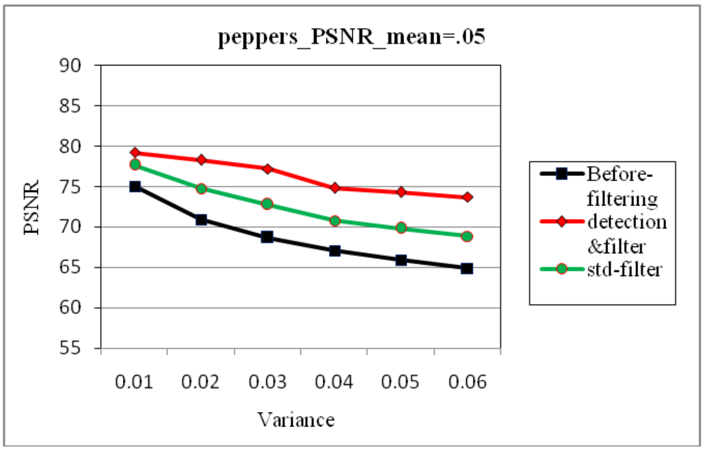


Figure 11: Graph of PSNR for peppers image with mean=0.05.

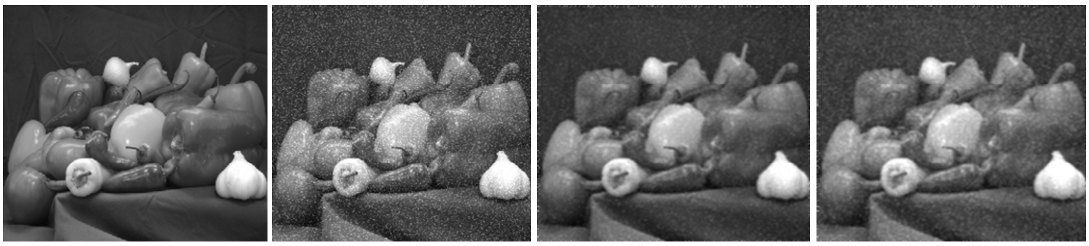


Figure 12: (a) Original, (b) ‘0’ mean ‘0.03’ variance Gaussian noise, (c) Proposed Detection & filtered, (d) Standard filtered of Peppers image.

Conclusion

In this paper a novel detection and filtering schemes for Gaussian noise were developed and compared with standard filtering method. Here a 5x5 window is considered for detection scheme. This 5x5 window is divided into nine 3x3 sub-windows in which the test pixel appears. The standard deviations of all sub-windows are calculated in which the test pixel appears and is providing relative information about the amount of the noise, if present, of the test pixel. The maximum standard deviation and minimum standard deviation are calculated using constants k_1 and k_2. The reasonable values are obtained as 0.5 and 2 respectively. It is concluded that the pixel is corrupted if the difference magnitude, $|\mu\text{-}x|$ lies in the range [a, b] where μ is mean of 3x3 neighborhood of the test pixel in which test pixel lies at the center of the window and x is test pixel intensity. Our proposed method shows better performance in detecting and filtering Gaussian noise than the standard mean filter for various combinations of mean and variances of additive Gaussian noise.

References

[1] H Lin and A.N. Willson , Jr., "Median filters with adaptive length," IEEE Trans. Circuits Syst., vol.35, pp.675-690, June.1998
[2] T. Sun and Y.Neuvo, "Detail-Preserving median based filters in image processing," Patt. Recogn. Lett., vol.15, pp.341-347, Apr.1994.
[3] A.C. Bovik, T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," IEEE Trans. ASSP, vol.31, pp.1342-1350, Dec.1983.
[4] R. C. Hardie and K. E. Barner, "Rank conditioned rank selection filters for signal restoration," IEEE Trans. Image processing, vol.3, pp.92-206, Mar.1994..
[5] G. Pok, J.Lin, and A.S. Nair, "Selective removal of impulse noise based on homogeneity level information," IEEE Trans. Image processing, vol.12, pp.85-92, Jan.2003.
[6] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," IEEE Trans. Pattern Anal. Machine Intell., vol.12, pp.629-639, 1990.
[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Proc. 1998 IEEE Int. Conf. Computer Vision, pp.839-846.
[8] Roman Garnett et.al, "A Universal Noise Removal Algorithm with an Impulse Detector," IEEE Trans. On Image processing , vol.14, pp.1747-1754, Nov.2005.
[9] Keigo Hirakawa and Thomas W. Parks, "Image Denoising Using Total Least Squares," IEEE Trans. On Image processing, vol.15, no.9, Sept.2006.
[10] Yeqiu Li et.al, "Removal of Gaussian Noise from Degraded Images in Wavelet Domain," Translated from Denki Ronbunshi, vol.126-c, No.11, Nov.2006, pp.1351-1358.
[11] William W. Hines et.al, "Probability and Statistics in Engineering," 4th Edition, John Wiley & Sons, Inc. 2003.
[12] Murray R. Spiegel et.al, "Probability and Statistics" Schanum's outlines", 2nd Edition, Tata McGraw-Hill 2007.