A Fast Method For Image Noise Estimation Using Laplacian Operator and Adaptive Edge Detection

Shen-Chuan Tai[1] , Shih-Ming Yang[2]
Department of Electrical Engineering
National Cheng Kung University
No. 1, University Road, Tainan 701, Taiwan, R.O.C.
sctai@mail.ncku.edu.tw[1] , ysm93d@dcmc.ee.ncku.edu.tw[2]

***Abstract*—We present a simple and fast algorithm for image noise estimation. The input image is assumed to be corrupted by additive zero mean Gaussian noise. To exclude structures or details from contributing to the noise variance estimation, a simple edge detection algorithm using first-order gradients is applied first. Then a Laplacian operator followed by an averaging over the whole image will provide very accurate noise variance estimation. There is only one parameter which is self-determined and adaptive to the image contents. Simulation results show that the proposed algorithm performs well for different types of images over a large range of noise variances. Performance comparisons against other approaches are also provided.**

Index Terms — Gaussian noise, noise estimation, Laplacian operator, noise reduction, edge detection.

I. INTRODUCTION

Noise reduction is very important in digital image processing. It can improve the accuracy or performance of other processing techniques that follow, such as image segmentation or recognition. Many noise reduction algorithms assume that the noise level is known a priori; and the algorithms are adopted to the amount of noise instead of using fixed parameters [1][2]. To make the assumption come true, noise estimation becomes an important research topic.

In 1993, Olsen [3] gave a complete description and comparison of six earlier estimation algorithms. They are classified into two different approaches: filter-based (or smoothing-based) and block-based.

In filter-based methods, the noisy image is first filtered by a low-pass filter to suppress the image structures. Then the noise variance is computed from the difference between the noisy image and the filtered image. The main difficulty of filter-based methods is that the difference image is assumed to be the noise but this assumption is not true for images with structures or details.

In blocked-based methods, images are tessellated into a number of blocks. The noise variance is then computed from a set of homogeneous blocks. The main issue of block-based methods is how to identify the homogeneous blocks.

There were different approaches [4]-[7] proposed for noise estimation since 1993. Basically they are still filter-based, block-based, or the combination of both. Filter-based methods are found to perform well for high noise levels but they usually require a high computational load. Block-based methods are in general less complex, but they tend to overestimate when noise levels are low.

Immerkær [4] proposed a simple and fast noise estimation method. It only required convolutions and averaging. But it failed for certain types of images when the noise levels are low. Here, we propose a modified approach to improve the performance of his work.

The remaining of this paper is organized as follows: Section II gives the image model. Section III briefly describes the fast noise estimation method presented in [4]. Section IV illustrates our proposed method. Section V describes another related work for comparisons. Section VI shows the experimental results followed by conclusions.

II. IMAGE MODEL

We assume that the image is corrupted by additive, zero-mean white Gaussian noise with unknown deviation σ_n, and the model is given by:

$$I_n(x,y) = I(x,y) + n(x,y)\,, \qquad (1)$$

where x and y are the vertical and horizontal coordinates of a pixel, $I_n(x, y)$, $I(x, y)$ and $n(x, y)$ are the noisy image, the original image and the additive Gaussian noise respectively. Our goal is to estimate the standard deviation σ_n of the noise from the noisy image.

III. THE FAST ESTIMATION METHOD

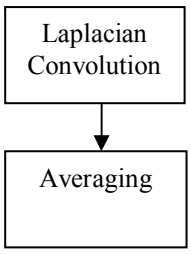

Figure 1. Block diagram of "Fast Estimation" [4].

978-1-4244-1688-2/08/$25.00 ©2008 IEEE

The first step of the "Fast Estimation" [4] method is to suppress the image structures by the following Laplacian operator:

$$N = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}. \qquad (2)$$

Then the standard deviation of the noise can be computed as

$$\sigma_n = \sqrt{\frac{\pi}{2}} \frac{1}{6(W-2)(H-2)} \sum_{\text{image } I} \left| I(x,y) * N \right|, \qquad (3)$$

where W and H are the width and height of the image respectively.

There are only convolutions and averaging, no sophisticated statistics computation. That's why it is fast. But, how good is it? As we will see in Section VI, it performs well for a large range of noise levels. For highly textured images, though, it perceives thin lines as noise and over-estimates the noise. Therefore, we propose using an adaptive edge detector to remedy the problem.

IV. PROPOSED METHOD

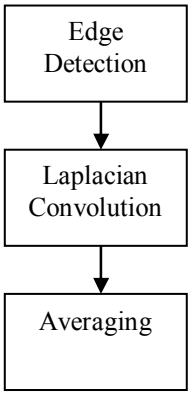

Figure 2. Block diagram of our proposed method.

We propose using the Sobel operator [8] for edge detection:

$$G_x = I(x,y) * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G_y = I(x,y) * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G = \left|G_x\right| + \left|G_y\right|. \qquad (4)$$

To decide the edge map, we select the threshold $\boldsymbol{G_{th}}$ for G based on the following suggestion quoted from Olsen [3]:

"In practice a percentage value p of the image area to be considered is much easier for the user to supply than a reasonable guess of the threshold value T."

First, we compute the histogram of G. Then, we select $\boldsymbol{G_{th}}$ to be the G value when the accumulated histogram reaches $\boldsymbol{p}$% of the whole image. In this way, even $\boldsymbol{p}$ is set to the same fixed value, the threshold values $\boldsymbol{G_{th}}$ will not be the same for different images. Therefore, we can say it is an "adaptive" edge detection.

After the edge map has been found, we simply follow the same approach as "the Fast Estimation" [4], but exclude the edge pixels.

V. RELATED WORK

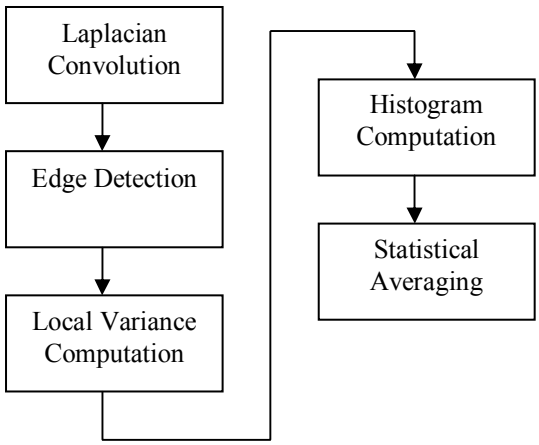

Figure 3. Block diagram of Bilcu and Vehvilainen's method [6].

Bilcu and Vehvilainen [6] proposed a method which was meant to improve the work by Rank, Lendl, and Unbehauen [5]. We found that Bilcu and Vehvilainen's method is very close to ours. They used the same Laplacian operator to suppress the image structure. They also used edge detection to exclude pixels belonging to edges. But we would like to emphasize the difference between these two methods.

First of all, the Laplacian convolution is computed pixel by pixel for both methods. In our method, the arithmetic average of the convolution results is computed to be the estimated noise deviation. But, in their method the local variances of 5*5 non-overlapping windows are computed first. Then the histogram computation is performed on the local variances. Finally, the statistical (weighted) average of the local variances is computed to be the estimated noise variance. It is obvious that their method will be more complicated than ours because of the local variance computation.

Secondly, we perform edge detection pixel by pixel. The threshold value is selected automatically and adaptive to the image contents. On the contrast, they perform edge detection on 5*5 windows. If the window contains edges, it is eliminated

from the estimation process. Since they didn't give the details on the edge detection method, we will implement in our own way in the following experiments.

VI. EXPERIMENTAL RESULTS

For performance comparisons, we define estimation ratio by the following equation to be the performance metric:

$$\text{estimation ratio} = \frac{\sigma_{estimated}}{\sigma_{added}}. \qquad (5)$$

The method proposed in [4] is referred to as FAST96. The method proposed in [6] is partially implemented without edge detection (BV05P) and then fully implemented with edge detection (BV05F). The edge detection algorithm is designed as the following:

First, the Sobel operation is performed pixel by pixel, and the same p% principle is applied to find the threshold G_{th}. For each 5*5 window, if there were more than N pixels are greater than G_{th}, the window is claimed to have edges.

While we struggled to find the best choice of p and N for BV05F, the p value for our method is set to 10 as suggested in [3] for all images.

The results presented here are obtained by averaging 20 Monte-Carlo simulations for each image (Fig. 4) and each noise deviation.

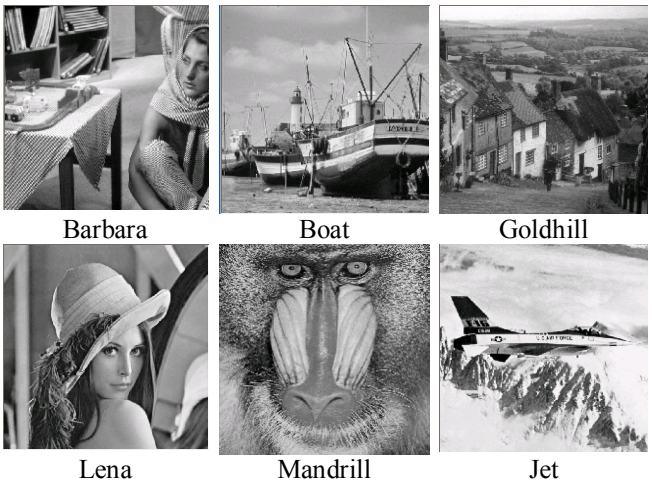

Figure 4. Six 512*512 images used for experiments.

Fig. 5 shows the performance of FAST96. While it performs well for high-level noise, it over-estimates when the noise level is low. Table I shows the exact data. For images with fine details (e.g. Mandrill) or structured patterns (e.g. Barbara), the estimation errors can be huge. The numbers in Table I printed in **bold** face indicate that the estimation error is greater than 50% for Lena. The worst case is more than 400% for Mandrill.

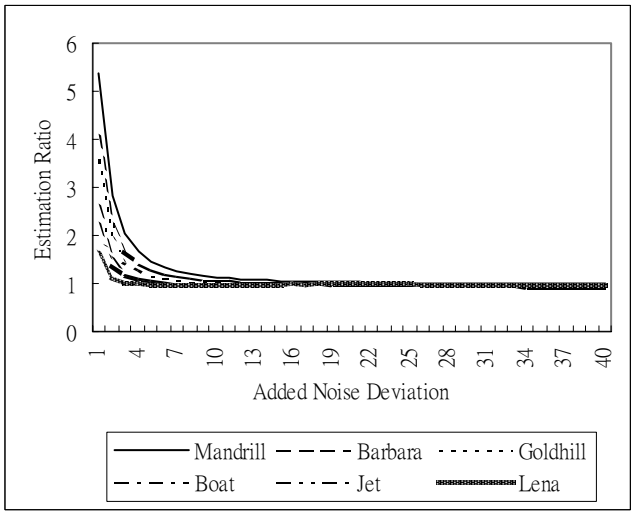

Figure 5. Estimation Ratio of FAST96 for noise deviation between 1 and 40.

TABLE I. ESTIMATION RATIO OF FAST96 FOR NOISE DEVIATION BETWEEN 1 AND 10.

σ_{added}	Mandrill	Barbara	Goldhill	Boat	Jet	Lena
1	**5.3877**	**4.0432**	**3.5299**	**2.5785**	**2.2256**	**1.5841**
2	**2.8344**	**2.2120**	**1.9350**	**1.5106**	1.3817	1.1078
3	**2.0395**	**1.6708**	1.4708	1.2286	1.1636	1.0115
4	**1.6721**	1.4301	1.2748	1.1185	1.0791	0.9828
5	1.4689	1.2993	1.1724	1.0649	1.0398	0.9724
6	1.3422	1.2185	1.1125	1.0356	1.0190	0.9700
7	1.2596	1.1668	1.0787	1.0196	1.0056	0.9687
8	1.2032	1.1292	1.0549	1.0096	0.9993	0.9704
9	1.1600	1.1038	1.0400	1.0017	0.9956	0.9710
10	1.1313	1.0831	1.0270	0.9984	0.9908	0.9731

Table II, Table III and Table IV show the comparisons among FAST96, BV05P and our method. The situations where our method makes significant improvements are printed in bold face. While we maintain the same good performance for high noise levels, we make significant improvements for low noise levels. For images with fine details or structured patterns (e.g. Mandrill and Barbara), the improvements are really amazing. All three methods tend to underestimate when the noise levels are high. But most of the estimation errors are less than 5%.

There is one interesting observation we would like to point out. If we compare the experimental results between FAST96 and BV05P, we would see that the two estimators almost have the same performance. Since we did not implement edge detection for BV05P, here the only difference between these

two methods is the way they compute the average deviation or variance. FAST96 simply computes the arithmetic average while BV05P computes the histogram first and then the statistical average. The results show that simple averaging is good enough and statistical averaging won't give us any benefits. The results in [3] reached the same conclusion. This observation justifies our approach. We compute histogram, too. But we use the results in finding the edge map and exclude image details from contributing to the noise variance. The histogram computation helps us detect edges adaptively. This is why we achieve improvements.

TABLE II. COMPARISONS OF ESTIMATION RATIO FOR BARBARA AND BOAT.

	Barbara			Boat		
σ added	FAST96	Ours	BV05P	FAST96	Ours	BV05P
1	4.0432	**2.2407**	3.2700	2.5785	**1.8875**	2.4540
2	2.2120	**1.3884**	1.8151	1.5106	**1.2096**	1.4493
3	1.6708	**1.1829**	1.4066	1.2286	**1.0600**	1.1967
4	1.4301	**1.1023**	1.2420	1.1185	**1.0056**	1.0998
5	1.2993	**1.0685**	1.1601	1.0649	**0.9896**	1.0544
6	1.2185	**1.0490**	1.1144	1.0356	0.9801	1.0266
7	1.1668	**1.0399**	1.0856	1.0196	0.9759	1.0125
8	1.1292	**1.0255**	1.0676	1.0096	0.9729	1.0021
9	1.1038	**1.0267**	1.0529	1.0017	0.9765	0.9960
10	1.0831	**1.0198**	1.0425	0.9984	0.9749	0.9923
15	1.0328	1.0065	1.0158	0.9895	0.9754	0.9822
20	1.0087	0.9938	0.9985	0.9844	0.9785	0.9778
25	0.9940	0.9836	0.9842	0.9802	0.9774	0.9740
30	0.9800	0.9727	0.9722	0.9750	0.9734	0.9679
35	0.9677	0.9598	0.9607	0.9689	0.9679	0.9594
40	0.9564	0.9486	0.9481	0.9584	0.9580	0.9507

TABLE III. COMPARISONS OF ESTIMATION RATIO FOR GOLDHILL AND JET.

	Goldhill			Jet		
σ added	FAST96	Ours	BV05P	FAST96	Ours	BV05P
1	3.5299	**2.3626**	3.5130	2.2256	**1.1223**	2.2286
2	1.9350	**1.4607**	1.9144	1.3817	**0.9637**	1.3575
3	1.4708	**1.2214**	1.4536	1.1636	**0.9418**	1.1493
4	1.2748	**1.1234**	1.2614	1.0791	**0.9451**	1.0689
5	1.1724	**1.0739**	1.1635	1.0398	0.9495	1.0324
6	1.1125	**1.0437**	1.1085	1.0190	0.9516	1.0138
7	1.0787	**1.0253**	1.0733	1.0056	0.9526	1.0010
8	1.0549	**1.0134**	1.0498	0.9993	0.9599	0.9936
9	1.0400	**1.0079**	1.0345	0.9956	0.9602	0.9887
10	1.0270	**1.0004**	1.0210	0.9908	0.9657	0.9853
15	1.0009	0.9919	0.9965	0.9829	0.9705	0.9780
20	0.9911	0.9864	0.9842	0.9767	0.9699	0.9699
25	0.9842	0.9775	0.9761	0.9652	0.9623	0.9570
30	0.9759	0.9690	0.9669	0.9482	0.9498	0.9397
35	0.9666	0.9591	0.9595	0.9311	0.9325	0.9221
40	0.9579	0.9546	0.9504	0.9114	0.9087	0.9028

TABLE IV. COMPARISONS OF ESTIMATION RATIO FOR LENA AND MANDRILL.

	Lena			Mandrill		
σ added	FAST96	Ours	BV05P	FAST96	Ours	BV05P
1	1.5841	**1.1777**	1.5063	5.3877	**2.6836**	5.4631
2	1.1078	**0.9555**	1.0826	2.8344	**1.5776**	2.8364
3	1.0115	0.9345	1.0031	2.0395	**1.2888**	2.0197
4	0.9828	0.9356	0.9784	1.6721	**1.1696**	1.6477
5	0.9724	0.9397	0.9691	1.4689	**1.1123**	1.4434
6	0.9700	0.9434	0.9660	1.3422	**1.0796**	1.3235
7	0.9687	0.9470	0.9655	1.2596	**1.0590**	1.2450
8	0.9704	0.9533	0.9639	1.2032	**1.0451**	1.1914
9	0.9710	0.9548	0.9661	1.1600	**1.0373**	1.1522
10	0.9731	0.9610	0.9667	1.1313	**1.0280**	1.1249
15	0.9781	0.9689	0.9722	1.0533	1.0083	1.0488
20	0.9804	0.9773	0.9731	1.0251	1.0025	1.0211
25	0.9793	0.9800	0.9715	1.0123	0.9989	1.0060
30	0.9744	0.9760	0.9674	1.0017	0.9937	0.9954
35	0.9705	0.9726	0.9606	0.9919	0.9915	0.9857
40	0.9619	0.9633	0.9524	0.9845	0.9859	0.9771

Table V, Table VI and Table VII show the comparisons between BV05F and our method. For BV05F we focus on low noise levels. The ***P*** and ***N*** values shown in the tables are the best choice for noise deviation between 1 and 10. Although the performance of our method and BV05F are close to each other, BV05F has two problems. First, its performance goes down when the noise levels go high. For some images, for example Jet, Mandrill and Goldhill, the performance decay very fast. Secondly, we only need one parameter (***P***) and it can be set to a fixed value of 10 for all images. While for BV05F there are two parameters (***P*** and ***N***) and we were not able to find an unified approach for determining the best choice of ***P*** and ***N***.

As far as the complexity being concerned, we implemented all these methods on a personal computer running Windows XP Professional with 2.2GHz Intel Pentium 4 CPU and 752MB main memory. For a 512*512 image, the execution time of FAST96, our method BV05P and BV05F were 0.04, 0.06, 0.06 and 0.14 seconds respectively. Therefore, we can conclude that our method outperforms BV05 with less complexity. Compared to FAST96, our method makes significant improvements. While the complexity is increased by 50%, the execution time is still acceptable.

VII. CONCLUSIONS

We propose an image noise estimation method based on [4]. An edge detector is introduced to exclude image details from contributing to the noise variance. The threshold value of the edge detector is self-determined and adaptive to the image contents. Simulation results have been presented and shown that we achieve significant improvements without too much complexity. However, if the complexity is not an issue, we can further improve the performance by tuning the parameter ***p*** according to a priori knowledge of the image or noise variance value [3].

TABLE V. COMPARISONS BETWEEN OUR METHOD AND BV05F FOR BARBARA AND BOAT.

	Barbara		Boat	
σ added	Ours	BV05F P=20,N=5	Ours	BV05F P=30,N=5
1	2.2407	**1.8771**	1.8875	**1.8550**
2	**1.3884**	1.9648	**1.2096**	1.9369
3	1.1829	**1.1609**	**1.0600**	1.3127
4	1.1023	**1.0412**	**1.0056**	1.1372
5	1.0685	**0.9968**	**0.9896**	1.0618
6	1.0490	**0.9780**	0.9801	**1.0229**
7	1.0399	**0.9704**	0.9759	**0.9981**
8	1.0255	**0.9970**	0.9729	**1.0154**
9	1.0267	**0.9905**	0.9765	**1.0007**
10	1.0198	**0.9805**	0.9749	**0.9907**
15	**1.0065**	0.9563	0.9754	**0.9933**
20	**0.9938**	0.9435	0.9785	**0.9853**
25	**0.9836**	0.9358	0.9774	**0.9839**
30	**0.9727**	0.9240	0.9734	**0.9770**
35	**0.9598**	0.9127	0.9679	**0.9725**
40	**0.9486**	0.9035	0.9580	**0.9621**

TABLE VI. COMPARISONS BETWEEN OUR METHOD AND BV05F FOR GOLDHILL AND JET.

	Goldhill		Jet	
σ added	Ours	BV05F P=20,N=5	Ours	BV05F P=20,N=5
1	2.3626	**2.3165**	**1.1223**	1.1018
2	**1.4607**	2.3965	**0.9637**	1.2314
3	**1.2214**	1.5259	0.9418	**1.0297**
4	**1.1234**	1.2730	0.9451	**0.9870**
5	**1.0739**	1.1553	0.9495	**0.9649**
6	**1.0437**	1.0237	0.9516	**0.9529**
7	**1.0253**	0.9975	**0.9526**	0.9416
8	**1.0134**	0.9766	**0.9599**	0.9355
9	**1.0079**	0.9658	**0.9602**	0.9313
10	**1.0004**	0.9558	**0.9657**	0.9264
15	**0.9919**	0.9327	**0.9705**	0.9197
20	**0.9864**	0.9240	**0.9699**	0.9129
25	**0.9775**	0.9234	**0.9623**	0.9041
30	**0.9690**	0.9149	**0.9498**	0.8932
35	**0.9591**	0.9067	**0.9325**	0.8737
40	**0.9546**	0.8981	**0.9087**	0.8546

TABLE VII. COMPARISONS BETWEEN OUR METHOD AND BV05F FOR LENA AND MANDRILL.

	Lena		Mandrill	
σ added	Ours	BV05F P=20,N=10	Ours	BV05F P=10,N=5
1	**1.1777**	1.3385	**2.6836**	2.6718
2	**0.9555**	1.5244	**1.5776**	2.7354
3	**0.9345**	1.0815	**1.2888**	1.5219
4	0.9356	**1.0073**	**1.1696**	1.3093
5	0.9397	**0.9798**	**1.1123**	1.1165
6	0.9434	**0.9689**	**1.0796**	1.0785
7	0.9470	**0.9655**	1.0590	**1.0523**
8	0.9533	**0.9639**	1.0451	**0.9962**
9	0.9548	**0.9654**	1.0373	**0.9838**
10	0.9610	**0.9660**	1.0280	**0.9749**
15	0.9689	**0.9724**	**1.0083**	0.9145
20	0.9773	**0.9733**	**1.0025**	0.8924
25	**0.9800**	0.9713	**0.9989**	0.8752
30	**0.9760**	0.9685	**0.9937**	0.8651
35	**0.9726**	0.9626	**0.9915**	0.8558
40	**0.9633**	0.9554	**0.9859**	0.8459

Acknowledgements

We thank the anonymous reviewers for giving us valuable suggestions, which help us improve the quality of this paper.

REFERENCES

[1] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," Sixth International Conference on Compute, pp. 839-46, New Dclhi, India, 1998.

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

[3] S. I. Olsen, "Estimation of noise in images: An evaluation", Comput. Vision Graphics Image Process. Graphic Models and Image Process, Vol. 55, No. 4, pp. 319-323, 1993.

[4] J. Immerkær, "Fast Noise Variance Estimation", Computer Vision and Image Understanding, Vol. 64, No. 2, pp. 300-302, Sep. 1996.

[5] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," Proc. IEE Vis. Image Signal Process., Vol. 146, No. 2, pp. 80-84, Apr. 1999.

[6] R. C. Bilcu and M. Vehvilainen, "A New Method for Noise Estimation in Images," Proc. IEEE EURASIP International Workshop on Nonlinear Signal and Image Processing, Sapporo, Japan, May 18-20, 2005.

[7] A. Amer and E. Dubois, "Fast and Reliable Structure-Oriented Video Noise Estimation," IEEE Trans. On Circuits and Systems for Video Technology, Vol. 15, No. 1, pp. 113-118, Jan. 2005.

[8] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," Prentice-Hall, Inc., New Jersey, USA, 2002.