Estimation of image noise variance

K.Rank, M.Lendl and R.Unbehauen

Abstract: A novel algorithm for estimating the noise variance of an image is presented. The image is assumed to be corrupted by Gaussian distributed noise. The algorithm estimates the noise variance in three steps. At first the noisy image is filtered by a horizontal and a vertical difference operator to suppress the influence of the (unknown) original image. In a second step a histogram of local signal variances is computed. Finally a statistical evaluation of the histogram provides the desired estimation value. For a comparison with several previously published estimation methods an ensemble of 128 natural and artificial test images is used. It is shown that with the novel algorithm more accurate results can be achieved.

1 Introduction

A common aim of image processing is to recover the original image x from the noisy image y. Many algorithms have been proposed in this field and usually the exact value of the noise variance σ_w^2 is required as a crucial filter parameter (see also [1]). For the estimation of the noise variance several algorithms have been developed [1–8]. An overview and discussion is given in [1, 9]. Generally the accuracy of the proposed algorithms is not always satisfactory.

2 Estimation algorithm

Following [10] let us define a digital, i.e. a sampled and quantised, grey level image z as a two-dimensional random variable $z(m, n)$ with a domain $D_0 \subset \mathbb{Z} \times \mathbb{Z}$ and the range $U \subset \mathbb{N}_0$. Since we only consider bounded images we may limit D_0 to $D_0 = \{(m, n)|m \in [1,M] \wedge n \in [1, N]; m, n, M, N \in \mathbb{N}\}$. So we describe an original image $x(m, n)$ by an array of $M\,N$ elements, where M and N are the numbers of rows and columns of the image x, respectively. In the same manner we define a noisy image $y(m, n)$ that is corrupted by additive zero-mean white Gaussian noise $w(m, n)$

$$y(m, n) = x(m, n) + w(m, n) \quad \forall\ (m, n) \in D_0 \qquad (1)$$

The noise signal $w(m, n)$ is assumed to be statistically independent of the original image x and the noise statistics to be constant within D_0.

Our novel estimation algorithm is organised in three steps. In a first step the image is preprocessed by a difference operator to minimise the influence of the original image. A second step computes a histogram of local standard deviations and in a third step the histogram is evaluated in order to receive the desired estimate $\hat{\sigma}_w^2$.

Step 1: Suppression of the original image x

The main difficulty of an accurate noise variance estimation is the intermixing of the statistics of the original image x and the noise w due to eqn. 1. The separation of the two signals is not an easy task. A large variance of the original image x usually causes an overestimation of the value for σ_w^2. To reduce the influence of the original image x we first filter the noisy image by a cascade of two 1-D difference operators. The first difference operation refers to the vertical direction and the second to the horizontal one, i.e.

$$y_1(m, n) = \frac{1}{\sqrt{2}}(y(m+1, n) - y(m, n)) \quad \forall\ (m, n) \in D_1 \quad (2)$$

with $D_1 = \{(m, n)\ |m \in [1, M-1] \wedge n \in [1, N]; m, n, M, N \in \mathbb{N}\}$ and

$$y_2(m, n) = \frac{1}{\sqrt{2}}(y_1(m, n+1) - y_1(m, n)) \quad \forall\ (m, n) \in D_2 \qquad (3)$$

with $D_2 = \{(m, n)|m \in [1, M-1] \wedge n \in [1, N-1]; m, n, M, N \in \mathbb{N}\}$. It is well known that the noise variance of the sum of two independent noise signals is the sum of the noise variances of the two components. Therefore it can easily be shown that the variance of the noise component is not affected by the operations in eqns. 2 and 3 since $w(m, n)$ is assumed to be a white noise signal, i.e. different pixels (m, n) are not correlated. Thus the noise variances of y and y_2 are the same.

One important feature of the operation eqns. 2 and 3 is that areas within the processed image exhibiting a linear increase of the grey level value are transformed to areas with constant values. This transformation is essential for the noise variance estimation since otherwise the local variance estimates $\hat{\sigma}^2(m, n)$ in these areas are shifted to larger values (see eqn. 4). Furthermore the 1-D difference operators cause further suppression effects, e.g. a partial suppression of edges. These effects, however, are more difficult to describe and they depend largely on the complexity of the original image. Figs. 2*a*,*b* [Note 1]

IEE Proceedings online no. 19990238
DOI: 10.1049/vis:19990238

Paper first received 18th June 1998 and in revised form 5th February 1999

The authors are with Lehrstuhl für Allgemeine und Theoretische Elektrotechnik, Universität Erlangen-Nürnberg, Cauerstraße 7, 91058, Erlangen, Germany

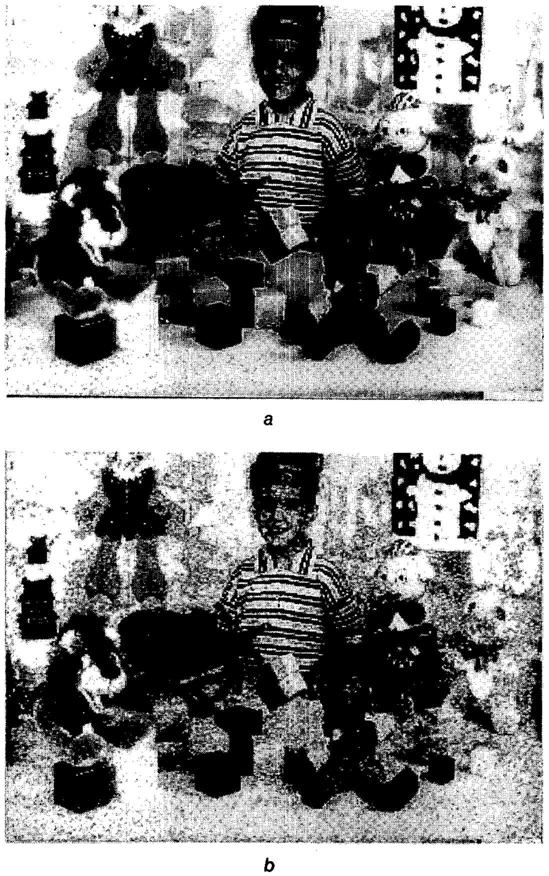

Fig. 1 *Playboy test image*
a Original
b Corrupted with Gaussian noise of $\sigma_w = 20$

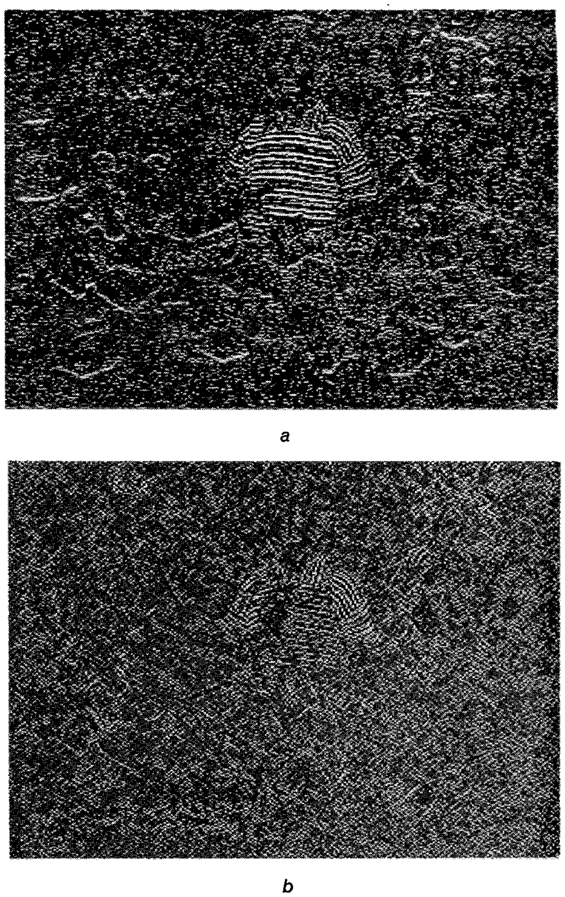

Fig. 2 *Difference operations performed on noisy test image of Fig. 1b*
a Vertical direction
b Vertical and horizontal direction

show the results of the operation eqns. 2 and 3 applied to the noisy image of Fig. 1*b*. The original image is shown in Fig. 1*a*. Although the noise variance is identical in Figs. 1*b* and 2*a,b* a slight residual of the original image can be seen in Fig. 2*b*.

Now we want to discuss some other approaches concerning the topic of suppressing the original image x. The approach in [6] also uses 1-D difference operators in the vertical and horizontal directions (applied to a smoothed image). In contrast to our proposition, there the difference operators are not connected in cascade, but are parallel with an addition of the squared operator outputs. In our opinion, however, the main suppression result is obtained only by a cascade connection, which can easily be verified. In [4, 5] the suppression of the original image is achieved by a kind of surface fitting using a 3 × 3 window. This method needs a much larger computational effort than the simple cascade of 1-D difference operators, while the suppression results are slightly worse compared to our approach. In Immarkaer [2] the difference of two Laplacians in a 3 × 3 window is used, which yields very good results for some difficult images. From our experience the optimum operator depends significantly on the properties of the image. Therefore we think that crucial progress can be made by an adaptive prefiltering technique.

Step 2: Computing the histogram of local standard deviations

Neglecting the residual component of the original image in y_2 an estimate $\hat{\sigma}_w^2$ for the noise variance σ_w^2 can be achieved by computing the local variance in the neighbourhood region of a pixel (m, n). Let us describe this region by a square window of size $L \times L$, with $L = 2K + 1$. The number of pixels inside the window is $N_L = L^2$. We tested several different window sizes and found that for our algorithm the best results, as defined by eqn. 12, are achieved with the minimal possible window size, i.e. a size of 3 × 3. The local estimate $\hat{\sigma}^2(m, n)$ at pixel (m, n) is obtained (omitting the index y_2) as

$$\begin{aligned}\hat{\sigma}^2(m,n) &= \frac{1}{N_L - 1}\sum_{i=-K}^{K}\sum_{j=-K}^{K}[y_2(m+i, n+j) \\ &\quad - \hat{\mu}(m,n)]^2 \\ &= \frac{1}{N_L - 1}\left(\left[\sum_{i=-K}^{K}\sum_{j=-K}^{K} y_2^2(m+i, n+j)\right]\right. \\ &\quad \left. - N_L\hat{\mu}^2(m,n)\right) \quad \forall\,(m,n) \in D_3 \qquad (4)\end{aligned}$$

Note 1: A constant offset of 128 was added to the signals y_1 and y_2 in Fig. 2*a,b* in order to shift the signals to grey levels that provide a better visibility. Furthermore, Figs. 1*a,b* and 2*a,b* were horizontally expanded with a factor of 4/3.

with $D_3 = \{(m, n) | m \in [1+K, M-1-K] \wedge n \in [1+K, N-1-K]\}$. The local mean $\hat{\mu}(m, n)$ is computed as

$$\hat{\mu}(m, n) = \frac{1}{N_L} \sum_{i=-K}^{K} \sum_{j=-K}^{K} y_2(m+i, n+j) \quad \forall\, (m, n) \in D_3 \tag{5}$$

It is worth noting that the local mean $\hat{\mu}(m, n)$ has to be taken into account even if the global mean of the signal y_2 vanishes; the local mean is usually nonzero. With the prefactor $1/(N_L - 1)$ the expectation of the estimate $\hat{\sigma}^2$ in eqn. 4 becomes identical with the value of the noise variance σ_w^2 (if the residual component of the signal x is neglected),

$$E\{\hat{\sigma}^2(m, n)\} = \sigma_w^2 \tag{6}$$

Thus a global, and therefore more accurate, estimate can be obtained by averaging the local estimates $\hat{\sigma}^2$ over the complete domain D_3. For this purpose, and also for a more precise analysis in step 3, we collect the values $\hat{\sigma}(m, n)$ in a histogram $h(k)$. With the unsquared values $\hat{\sigma}$, instead of $\hat{\sigma}^2$, the figure domain of the histogram will be compressed and the resolution for small values of $\hat{\sigma}$ increases. The histogram $h(k)$ with the entries $\hat{\sigma}(m, n)$ is defined as

$$h(k) = \begin{cases} |\{(m, n) | k - \frac{1}{2} \leq \alpha\hat{\sigma}(m, n) < k + \frac{1}{2} \wedge (m, n) \in D_3\}| \\ \qquad \text{if } k = 1, \ldots, k_{\max} \\ 2|\{(m, n) | 0 \leq \alpha\hat{\sigma}(m, n) < \frac{1}{2} \wedge (m, n) \in D_3\}| \\ \qquad \text{if } k = 0 \end{cases} \tag{7}$$

where here the symbol $|\{\cdot\}|$ denotes the number of elements of the set $\{\cdot\}$. Please note that the factor 2 for the case $k = 0$ is necessary since the interval [0, 1/2] has only half the length of the other intervals $[k - 1/2, k + 1/2]$. The value $k_{\max}$ has to be large enough to cover the whole figure domain of the values $\alpha\hat{\sigma}(m, n)$, $(m, n) \in D_3$. By means of the parameter α the domain of the histogram $h(k)$ can be compressed or expanded. Depending on the dynamic range of the image the value of α (with a suitable $k_{\max}$) should be large enough, so the domain of the histogram will contain about 100 to 200 samples for a k (The exact value is not crucial). For our test images (x, $y \in [0, 255]$) a value of $\alpha = 1$ was sufficient. Fig. 3 shows three histograms $h(k)$: one of the noisy image of Fig. 1*b*, of a noisy image with the same noise variance ($\sigma_w = 20$) but with a uniform original ($x(m, n) = 128$), and one of the noise-free playboy image.

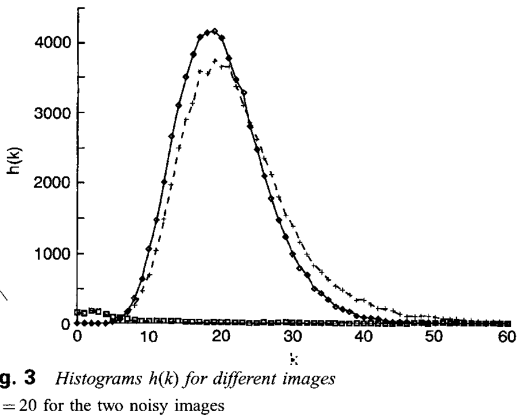

Fig. 3 *Histograms h(k) for different images*

$\sigma_w = 20$ for the two noisy images
◇ noisy uniform image
+ noisy Playboy image
□ noise-free Playboy image

Step 3: Evaluation of the histogram

An averaging [Note 2] of the values $\hat{\sigma}^2(m, n)$ for $(m, n) \in D_3$, as mentioned in step 2, can now be performed by computing the mean square value of the histogram [Note 3],

$$s_1^2 = \frac{\sum_{k=0}^{k_{\max}} k^2 h(k)}{\sum_{k=0}^{k_{\max}} h(k)} \tag{8}$$

Then the value s_1^2/α^2 is an initial global estimate for the noise variance σ_w^2. However, from eqn. 8 usually an overestimate is obtained. This effect can easily be explained by comparing the histograms for the noisy uniform image and the noisy playboy image of Fig. 3. In eqn. 4 the residual component of the original image has been neglected. The variance of this component shifts the histogram slightly to the right and causes a long tail of large values. Therefore the average of eqn. 8 yields a larger value compared to a histogram for a uniform image.

A further improvement of the estimation is based on the additional knowledge that the histogram $h(k)$ for a uniform image must have a certain shape (it can be expressed by means of a χ^2-function). A characteristic feature of this histogram is its rapid descent for large values of k. In contrast to this, the histogram $h(k)$ for a natural image usually exhibits a significantly slower descent since in essence it results from a convolution of the histogram for a noisy uniform image with the histogram for the noise-free original image (an example is shown in Fig. 3). This *a priori* knowledge can be used for a further reduction of the influence of the original image. A simple, easy to implement and very fast method, for solving this deconvolution problem is the fade-out of the histogram by a descending weighting function $g(k)$. Here also fitted with a χ^2-function is a possible [9]; however, this would imply a greater effort with respect to the implementation and higher computational costs, while the results cannot be improved significantly. The best results were achieved by a soft fade-out employing a cosine function

$$g_l(k) = \begin{cases} 1 & \text{if } k \leq s_l \\ \frac{1}{2}\left(1 - \cos\left(\frac{\beta - k/s_l}{1 - \beta}\pi\right)\right) & \text{if } s_l < k < \beta s_l \\ 0 & \text{if } k \geq \beta s_l \end{cases} \tag{9}$$

The fade-out procedure is performed several times (indicated by index l) as explained later. The initial value s_1 is taken from eqn. 8. The shape of the fade-out function g_l is depicted in Fig. 4. The adjustment of β can be performed independently of a certain ensemble of test images. It is obtained by using exclusively the uniform image and its noisy versions for an optimisation process (cf. Section 3) and yields a value of $\beta = 2.15$. An optimisation by means of the 128 test images yields a value of $\beta = 2.12$. This indicates that a lower value of β magnifies the effect of the

Note 2: Since $h(k)$ is Gaussian-like, here the mean operation is better than the median. Only if the residual component of the original image is extremely high the median operator or a combination of median and mean turns out to be advantageous. Further investigations are necessary in this case.

Note 3: The entries of $h(k)$ are the unsquared values.

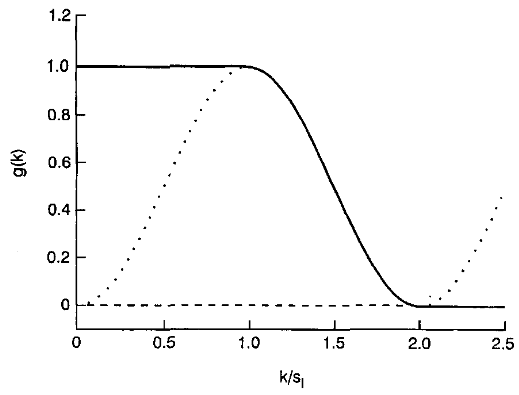

Fig. 4 *Weighting function of cosine fade-out, $\beta = 2$*

fade-out procedure. For certain applications or specific images this effect could be used for a better suppression of the residual component of the original image.

The exact shape of the fade-out function is not crucial. For example, we tested a fade-out with a function descending linearly from $k = s_l$, ($g_l = 1$) to $k = \beta_{s_l}$, ($g_l = 0$) with a value for β of about 2.0 or a simple cutoff of the histogram at β_{s_l} ($g_l = 1$ for $k < \beta_{s_l}$, $g_l = 0$ otherwise) with $\beta = 1.5$. In most cases the results compare with the cosine fade-out.

Now, using the weighting function of eqn. 9 an improved value of the square mean s^2 is computed iteratively by

$$s_{l+1}^2 = \frac{\sum_{k=0}^{k_{\max}} k^2 g_l(k)h(k)}{\sum_{k=0}^{k_{\max}} g_l(k)h(k)} \qquad (10)$$

where the initial value s_l is taken from eqn. 8. Thus the mean square value s^2 can be refined successively by evaluating alternately eqns. 9 and 10. After three to five iterations a convergence to a nearly constant value $s_{l_{\max}}^2$ is achieved. With a fixed value of $l_{\max} = 4$ we obtained good results; all the results given in this paper are related to $l_{\max} = 4$. The final estimate $\hat{\sigma}_\omega^2$ for the noise variance σ_w^2 then reads

$$\hat{\sigma}_w^2 = s_{l_{\max}}^2 / \alpha^2 \qquad (11)$$

The convergence of the iterative fade-out process of eqns. 9 and 10 can easily be proved by induction: for a lower s_{l+1} the descent of the weighting function g_{l+1} starts at lower values of k, which results in a lower or equal value s_{l+2}, etc. The initial condition of the proof, i.e. $s_2 < s_l$, can be verified by comparing eqns. 8 and 10 for $l = 1$.

3 Comparison with other methods

In this Section we present the results of a rough comparison with several other estimation methods. A more detailed comparison is left for future work. Investigations have to be made including a much larger number of different kinds of images and different prefiltering methods for a larger sample of different values of σ_w. This will provide a better insight of the performance of this novel approach.

For a comparison with the methods described in [1, 4–6] we used an ensemble of 128 images with a size of 256 × 256 pixels for each image. These images were generated by adding Gaussian distributed noise of different noise variances to 16 original images. It should be noted that due to the saturation effect, caused by a limited range $x, y \in [0, 255]$ – in practice often an 8 bit coding is used – the resulting noise is no longer exactly zero-mean and exactly Gaussian distributed, especially for large noise variances. Furthermore, the resulting noise variance is slightly lower than the noise variance of the added signal. With respect to these effects we added noise in a way that the resulting images had standard deviations of 0, 2, 5, 10, 20, 30, 50, 80. These values are denoted as σ_a (cf. eqns. 14 and 15).

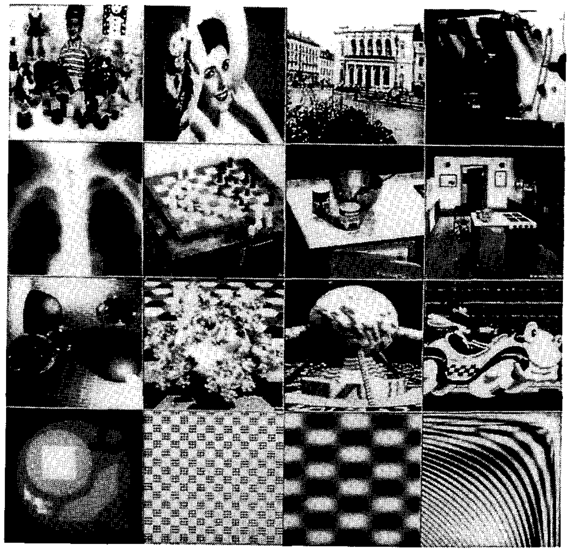

Fig. 5 *16 original images used for comparison*

The 16 original images are depicted in Fig. 5. The first five images (counting row-wise) are natural images. The next six images were generated artificially by a ray tracing program (e.g. see [11]). This method offers the advantage that natural-like original images can be generated absolutely noise-free – in contrast to real natural images. The last five images were generated otherwise. In our opinion this is a good mixture between natural, natural-like and artificial images.

In the following we use the standard deviation instead of the noise variance since it is more illustrative than the noise variance itself.

For our comparison the variable parameters of each estimation algorithm were optimised with respect to the averaged squared absolute estimation error [Note 4] of the noise standard deviation defined as

$$\overline{\varepsilon_a^2} = \frac{1}{N_I} \sum_{i=1}^{N_I} \varepsilon_a^2(i) \qquad (12)$$

where N_I is the number of tested images and $\varepsilon_a(i)$ is the absolute estimation error of a single image with number i,

$$\varepsilon_a(i) = \hat{\sigma}_w(i) - \sigma_w(i) \qquad (13)$$

The noise variance

$$\sigma_w^2(i) = \sigma_a^2(i) + \sigma_0^2(j) \qquad (14)$$

Note 4: We also worked with the averaged squared relative error that can sometimes be found in the literature as an optimisation criterion. However, in our experience and in agreement with Press *et al.* ([12], Section 9.1) the 'absolute error' eqn. 12 is better suited as an optimisation criterion than the 'relative error' since the relative error can take excessively large values for small noise variances σ_w.

Table 1: Estimation errors for different noise variance estimation algorithms, optimised with respect to averaged squared absolute error $\overline{\varepsilon_a^2}$ and averaged over 128 noisy test images with eight different σ_w

	Other algorithms			Our algorithm		
	[1]	[4, 5]	[6]	No interleave	Interleave-rate = 3	Interleave-rate = 7
$\overline{\varepsilon_a^2}$	37.70	62.57	13.10	1.99	2.04	2.50
$\bar{\varepsilon}_a$	3.58	5.30	1.61	0.90	0.91	0.99
T_c, s	0.41	1.56	0.26	1.19	0.14	0.03

T_c indicates the computational time for image

depends on the variance $\sigma_a^2(i)$ of the noise actually added to the images and a term $\sigma_0^2(j)$ that expresses the possibly existing noise variance of the original image itself. The index $j = j(i)$ denotes the number of the original image (say x_j) from which the noisy image numbered with i (say y_i) was generated by adding noise. The term $\sigma_0^2(j)$ takes care of the assumption that the five natural images are already disturbed by a certain degree of noise. For the 11 artificially generated images we have $\sigma_0^2(j) = 0$. Since $\sigma_0^2(j)$ is unknown for the five natural images it has to be taken into account by means of five additional free parameters within the optimisation process. The actually added noise variance $\sigma_a^2(i)$ of test image y_i is computed as

$$\sigma_a^2(i) = \frac{1}{|D_0|} \sum_{(m,n)\in D_0} (y_i(m, n) - x_j(m, n) - \mu_{ij})^2 \qquad (15)$$

where

$$\mu_{ij} = \frac{1}{|D_0|} \sum_{(m,n)\in D_0} (y_i(m, n) - x_j(m, n)) \qquad (16)$$

We eliminate the mean μ_{ij} – resulting from saturation effects – from eqn. 15 since it is irrelevant.

As the optimisation method we used the well-known simplex algorithm [12, 13], to avoid the evaluation of the gradient of the error function. For our simulation we ran several optimisations with different initial values for the parameters $\sigma_0(j)$, $j - 1, \ldots, 5$ and β. Table 1 shows the results of the comparison. Besides the averaged squared errors we also entered the more illustrative average deviation

$$\bar{\varepsilon}_a = \frac{1}{N_I} \sum_{i=1}^{N_I} |\varepsilon_a(i)| \qquad (17)$$

where $\bar{\varepsilon}_a$ is a measure for the average absolute estimation error of the noise standard deviation. Please note that Table 1 can give only a rough overview of the performance of the tested algorithms since some results depend on the ensemble of test images used. Poor values of $\bar{\varepsilon}_a$ result mainly from one or two 'outlierimages' which are difficult to estimate. The poor results of the algorithm [4, 5] are due to a fixed threshold and an underestimation by a certain percentage. After an improvement of these deficits the error $\bar{\varepsilon}_a$ could be halved. Nevertheless, from Table 1 and from our experience with many other test images we consider that the accuracy of our algorithm compares favourably with the other tested algorithms.

The computation time of our algorithm for one test image is 1.19 s on a 23 MFLOPS/76 MIPS machine, where 99.3% of the computation time was consumed by steps 1 and 2, in particular for the computation of the variances. The computation time can significantly be reduced if eqns. 4, 5 and 7 are evaluated with an undersampled version of the set D_3 without great loss of accuracy. Table 1 shows two examples with interleave rates of 3 and 7 for each direction. For Gaussian-distributed noise a further reduction of the computation time can be achieved by using an approximation when computing the variances [2].

4 Conclusion

In this paper a novel algorithm for the estimation of the image noise variance has been proposed. The algorithm is based on an iterative histogram analysis with the aid of a cosine fade-out function. The entries of the histogram are the local noise standard deviations, whereas the image has been prefiltered by difference operators in the vertical and horizontal directions. A rough comparison with several previously published estimation methods indicates an improved accuracy of the new algorithm with respect to the other methods.

A field for further investigations could be an adaptive prefiltering of the noisy image for a better suppression of the influence of the original image since the degree of this influence is decisive for the accuracy of the estimation.

5 References

1 MEER, P., JOLION, J.-M., and ROSENFELD, A.: 'A fast parallel algorithm for blind estimation of noise variance', *IEEE Trans. Pattern Anal. Machine Intell.*, 1990, **12**, pp. 216–223

2 IMMERKAER, J.: 'Fast noise variance estimation', *Comput. Vis. Image Underst.*, 1996, **64**, pp. 300–302

3 LEE, J.S., and HOPPEL, K.: 'Noise modeling and estimation of remotely-sensed images'. Proceedings of 1989 International Geoscience and Remote Sensing, Vancouver, 1989, **2**, pp. 1005–1008

4 BESL, P.J., and JAIN, R.C.: 'Segmentation through variable-order surface fitting', *IEEE Trans. Pattern Anal. Machine Intell.*, 1988, **10**, pp. 167–192

5 BESL, P. J.: 'Surfaces in range image understanding' (Springer, 1988)

6 VOORHEES, H., and POGGIO, T.: 'Detecting blobs as textons in natural images'. Proceedings of Image Understanding Workshop, Los Angelos, Los Angelos, 1987, pp. 892–899

7 MASTIN, G.A.: 'Adaptive filters for digital image noise smoothing, an evaluation', *Comput. Vis. Graphics Image Process.*, 1985, **31**, pp. 103–121

8 BRACHO, R., and SANDERSON, A.C.: 'Segmentation of images based on intensity gradient information'. Proceedings of CVPR-85 Conference on Computer Vision and Pattern Recognition, San Francisco, 1985, pp. 341–347

9 OLSEN, S.I: 'Estimation of noise in images: An evaluation' *CVGIP, Graph. Models Image Process.*, 1993, **55**, pp. 319–323

10 GONZALEZ, R.C., and WINTZ, P.: 'Digital image processing' (Addison-Wesley, 1987, 2nd edn.)

11 GLASSNER, A.S.: 'An introduction to ray tracing' (Academic Press, London, 1989)

12 PRESS, W.H., TEUKOLSKY, S.A., VETTERING, W.T., and FLANNERY, B.P.: 'Numerical recipes in C' (Cambridge University Press, Cambridge, 1994, 2nd edn.)

13 NELDER, J.A., and MEAD, R.: 'A simplex method for function minimization', *Comput. J.*, **7**, (4), 1965, pp. 308–313