Image Noise Level Estimation by Principal Component Analysis

Stanislav Pyatykh, Jürgen Hesser, and Lei Zheng

Abstract—**The problem of blind noise level estimation arises in many image processing applications, such as denoising, compression, and segmentation. In this paper, we propose a new noise level estimation method on the basis of principal component analysis of image blocks. We show that the noise variance can be estimated as the smallest eigenvalue of the image block covariance matrix. Compared with 13 existing methods, the proposed approach shows a good compromise between speed and accuracy. It is at least 15 times faster than methods with similar accuracy, and it is at least two times more accurate than other methods. Our method does not assume the existence of homogeneous areas in the input image and, hence, can successfully process images containing only textures.**

Index Terms—**Additive white noise, estimation, image processing, principal component analysis.**

I. INTRODUCTION

BLIND noise level estimation is an important image processing step, since the noise level is not always known beforehand, but many image denoising [1], [2], compression [3], and segmentation [4] algorithms take it as an input parameter; and their performance depends heavily on the accuracy of the noise level estimate.

The most widely used noise model, which is assumed in this work as well, is signal-independent additive white Gaussian noise. Noise variance estimation algorithms were being developed over the last two decades; and most of them include one or several common steps.

1) Separation of the signal from the noise.

   a) Preclassification of homogeneous areas [5]–[8]. These areas are the most suitable for noise variance estimation, because the noisy image variance equals the noise variance there.

   b) Image filtering [6], [9]–[15]. The processed image is convolved with a high-pass filter (e.g. Laplacian kernel); or the difference of the processed image and the response of a low-pass filter is computed. The filtering result contains the noise as well as object edges, which can be recognized by an edge detector and removed. The result of this procedure is assumed to contain only the noise, which allows direct estimation of the noise variance.

   c) Wavelet transform [16]–[19]. The simplest assumption that the wavelet coefficients at the finest decomposition level (subband HH_1) correspond only to the noise often leads to significant overestimates [19], because these wavelet coefficients are affected by image structures as well. In [16], it is assumed that only wavelet coefficients with the absolute value smaller than some threshold are caused by the noise, where the threshold is found by an iterative procedure.

2) Analysis of the local variance estimate distribution [10], [11], [20]–[23]. The result of the signal and noise separation is often not perfect, therefore the distribution of local (computed for image blocks) variance estimates contains outliers. Thereby, robust statistical methods insensitive to outliers are applied in order to compute the final noise variance estimate. Several approaches have been proposed here, such as the median of local estimates [10], the mode of local estimates [21], and the average of several smallest local estimates [22], [23].

Furthermore, there are some other original approaches: discrete cosine transform (DCT) of image blocks [24] concentrates image structures in low frequency transform coefficients, allowing noise variance estimation using high frequency coefficients; 3-D DCT of image block stacks [25] utilizes image self-similarity in order to separate the signal from the noise. In [26], the gray value distribution is analyzed. Gray values caused by image structures are considered as outliers; and a robust estimator is suggested. In [27], the distribution of a bandpass filter response is modeled as a two distribution Gaussian mixture, where the distributions are associated with the signal and the noise respectively. The model parameters are computed using the expectation-maximization algorithm. The digital representation of images is considered in work [28], where a measure of bit-plane randomness is used in order to compute the noise variance estimate. The authors of [29] suggest that the kurtosis of marginal bandpass filter response distributions should be constant for a noise-free image. That allows the construction of a kurtosis model for a noisy image; and the noise variance is assessed by finding the best parameters of this model. In [19], the processed image is denoised by the BayesShrink algorithm using several values of the noise variance; and the behavior of the residual

autocorrelation in a range of noise variance values is analyzed in order to find the true noise variance. In [30], a theoretical result for the noise standard deviation was established for the case when the observed image results from the random presence or absence of the signal in additive white Gaussian noise. Two training methods were proposed in [17]. In the first method, the noise standard deviation estimate is computed as a linear combination of normalized moments with learned coefficients. In the second method, the value of the cumulative distribution function (CDF) of local variances at a given point is computed for training images and stored in a lookup table against the noise variance. For a new image, the noise variance estimate is taken from the lookup table using the CDF value of this image. Last but not least, a Bayesian framework with a learned Markov random field prior (Fields of Experts [31]) for simultaneous deblurring and noise level estimation was proposed in [32].

Most of the methods listed above are based on the assumption, that the processed image contains a sufficient amount of homogeneous areas. However, this is not always the case, since there are images containing mostly textures. The problem of the noise variance estimation for such images has not been solved yet.

In this work, we propose a new noise level estimation method. It is based on principal component analysis (PCA) of image blocks, which has been already successfully utilized in various image processing tasks such as compression [33], denoising [34]–[36], and quality assessment [37]. The advantages of the proposed method are:

1) high computational efficiency;
2) ability to process images with textures, even if there are no homogeneous areas;
3) the same or better accuracy compared with the state of the art.

The rest of the article is organized as follows. We start with an example describing the idea in Subsection II-A. The method is explained in detail in Subsections II-B–II-E. The algorithm is given in Subsection II-F; and its efficient implementation is discussed in Subsection II-G. The results and the discussion are represented in Sections III and IV respectively. We conclude in Section V.

II. METHOD

A. Idea of the Method

Let us demonstrate the ability of image block PCA to estimate the noise variance on a simple 1D example. Consider noise-free signal $(x_k) = (2 + (-1)^k) = (1, 3, 1, 3, \ldots)$ and noisy signal $(y_k) = (x_k + n_k)$, where n_k are realizations of a random variable with normal distribution $\mathcal{N}(0; 0.5^2)$. The processing of these signals using a sliding window, with the width equal to 2, results in two point sets: $\{\mathbf{x}_k\} = \{(x_k; x_{k+1})\}$ for the noise-free signal and $\{\mathbf{y}_k\} = \{(y_k; y_{k+1})\} = \{(x_k; x_{k+1}) + (n_k; n_{k+1})\}$ for the noisy signal. By construction, points $\mathbf{x}_k$ can have only two values: $(1; 3)$ or $(3; 1)$. Points $\mathbf{y}_k$ are presented in Fig. 1.

Applying PCA to point set $\{\mathbf{y}_k\}$ gives new coordinate system $(u_1; u_2)$ (also shown in Fig. 1), in which u_1 is associated with
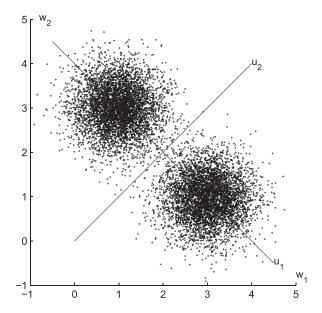

Fig. 1. Points $\mathbf{y}_k$ in original coordinate system $(w_1; w_2)$ and new coordinate system $(u_1; u_2)$ computed by PCA.

both the noise-free signal and the noise, and u_2 is associated only with the noise. This allows the computation of the noise variance estimate as the variance of the points along u_2.

This example shows some properties of the proposed method:

1) The method can be applied if the blocks computed from the noise-free signal can be represented by a number of dimensions smaller than the block size. In the example above, 2-dimensional points $\mathbf{x}_k$ with coordinates $(-\sqrt{2}; 0)$ or $(\sqrt{2}; 0)$ in the new coordinate system are purely situated on u_1. Therefore, they can be represented by 1-dimensional values in the new coordinate system.
2) If the blocks computed from the noise-free signal cannot be represented by a number of dimensions smaller than the block size, we cannot apply PCA directly in order to get the noise variance. In the example above, if the block set had three centroids, which did not lie on one line, then PCA would not provide a coordinate associated only with the noise.
3) No assumption about signal constancy is required. Indeed, in the example above, noise-free signal (x_k) contains no constant parts.

B. Image Block Model

Similar to the previous subsection, let $\mathbf{x}$ be a noise-free image of size $S_1 \times S_2$, where S_1 is the number of columns and S_2 is the number of rows, $\mathbf{y} = \mathbf{x} + \mathbf{n}$ be an image corrupted with signal-independent additive white Gaussian noise $\mathbf{n}$ with zero mean. Noise variance σ^2 is unknown and should be estimated.

Each of images $\mathbf{x}$, $\mathbf{n}$, $\mathbf{y}$ contains $N = (S_1 - M_1 + 1)(S_2 - M_2 + 1)$ blocks of size $M_1 \times M_2$, whose left-top corner positions are taken from set $\{1, \ldots, S_1 - M_1 + 1\} \times \{1, \ldots, S_2 - M_2 + 1\}$. These blocks can be rearranged into vectors with $M = M_1 M_2$ elements and considered as realizations $\mathbf{x}_i$, $\mathbf{n}_i$,

$\mathbf{y}_i$, $i = 1, \ldots, N$ of random vectors $\mathbf{X}$, $\mathbf{N}$, and $\mathbf{Y}$ respectively. As $\mathbf{n}$ is signal-independent additive white Gaussian noise, $\mathbf{N} \sim \mathcal{N}_M(0, \sigma^2 I)$ and $\mathrm{cov}(\mathbf{X}, \mathbf{N}) = 0$.

C. Principal Component Analysis

Let $S_\mathbf{X}$, $S_\mathbf{Y}$ be the sample covariance matrices of $\mathbf{X}$ and $\mathbf{Y}$ respectively, $\tilde{\lambda}_{\mathbf{X},1} \geq \tilde{\lambda}_{\mathbf{X},2} \geq \cdots \geq \tilde{\lambda}_{\mathbf{X},M}$ be the eigenvalues of $S_\mathbf{X}$ with the corresponding normalized eigenvectors $\tilde{\mathbf{v}}_{\mathbf{X},1}, \ldots, \tilde{\mathbf{v}}_{\mathbf{X},M}$, and $\tilde{\lambda}_{\mathbf{Y},1} \geq \tilde{\lambda}_{\mathbf{Y},2} \geq \cdots \geq \tilde{\lambda}_{\mathbf{Y},M}$ be the eigenvalues of $S_\mathbf{Y}$ with the corresponding normalized eigenvectors $\tilde{\mathbf{v}}_{\mathbf{Y},1}, \ldots, \tilde{\mathbf{v}}_{\mathbf{Y},M}$. Then, $\tilde{\mathbf{v}}_{\mathbf{Y},1}^\mathrm{T}\mathbf{Y}, \ldots, \tilde{\mathbf{v}}_{\mathbf{Y},M}^\mathrm{T}\mathbf{Y}$ represent the sample principal components of $\mathbf{Y}$ [38], which have the property

$$s^2(\tilde{\mathbf{v}}_{\mathbf{Y},k}^\mathrm{T}\mathbf{Y}) = \tilde{\lambda}_{\mathbf{Y},k}, \qquad k = 1, 2, \ldots, M$$

where s^2 denotes the sample variance.

In order to develop our method further, we define a class of noise-free images, for which PCA can be applied for noise variance estimation. Such noise-free images satisfy the following assumption:

Assumption 1: Let m be a predefined positive integer number. The information in noise-free image $\mathbf{x}$ is redundant in the sense that all $\mathbf{x}_i$ lie in subspace $V_{M-m} \subset \mathbb{R}^M$, whose dimension $M-m$ is smaller than the number of coordinates M.

When this assumption holds, we consider that random vector $\mathbf{X}$ takes its values only in subspace V_{M-m}. It means the existence of a linear dependence between components of $\mathbf{X}$, i.e. a linear dependence between pixels of $\mathbf{x}$ in the image blocks. This assumption also implies that $\mathbf{X}$ has zero variance along any direction orthogonal to V_{M-m}. The value of m is listed in Table II.

The following theorem provides a way to apply PCA for noise variance estimation:

Theorem 1: If Assumption 1 is satisfied then $\mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2|)$ is bounded above by $\sigma^2/\sqrt{N}$ asymptotically:

$$\mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2|) = O(\sigma^2/\sqrt{N}), \qquad N \to \infty \qquad (1)$$
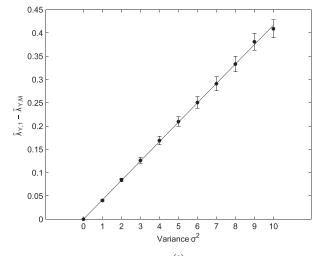
for all $i = M - m + 1, \ldots, M$.[1]

The formal proof is given in the Appendix.

The result of Theorem 1 can be intuitively explained as follows. When considering population principal components, $\mathrm{cov}(\mathbf{X}, \mathbf{N}) = 0$ implies that $\Sigma_\mathbf{Y} = \Sigma_\mathbf{X} + \Sigma_\mathbf{N}$, where $\Sigma_\mathbf{X}$, $\Sigma_\mathbf{N}$, and $\Sigma_\mathbf{Y}$ are the population covariance matrices of $\mathbf{X}$, $\mathbf{N}$, and $\mathbf{Y}$ respectively. Furthermore, $\Sigma_\mathbf{N} = \sigma^2 I$ and m smallest eigenvalues of $\Sigma_\mathbf{X}$ are zeros under Assumption 1. Hence m smallest eigenvalues of $\Sigma_\mathbf{Y}$ equal σ^2. For sample principal components, these equalities hold only approximately, i.e. $\tilde{\lambda}_{\mathbf{Y},i}$, $i = M - m + 1, \ldots, M$ are approximately equal σ^2, and the error $|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2|$ converges to zero as sample size N tends to infinity. Theorem 1 gives an asymptotic bound for the convergence speed. It states that the expected value of $|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2|$ converges to zero as $\sigma^2/\sqrt{N}$ or faster.

According to Theorem 1, when Assumption 1 is satisfied

$$\lim_{N \to \infty} \mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},M} - \sigma^2|) = 0 \qquad (2)$$

---

[1] Big O notation means that $\exists C$, $\exists N_0$ such that $\forall N \geq N_0$ $\mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2|) \leq C\sigma^2/\sqrt{N}$, where C does not depend on the distributions of $\mathbf{X}$ and $\mathbf{N}$.
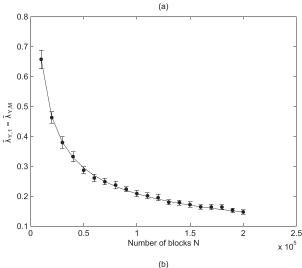

Fig. 2. Round markers and vertical bars are the mean values of $\tilde{\lambda}_{\mathbf{Y},1} - \tilde{\lambda}_{\mathbf{Y},M}$ and their 99% confidence intervals computed from 100 realizations of noise $\mathbf{n}$, respectively. The lines are the approximation by function $13.2\sigma^2/\sqrt{N}$. (a) $\tilde{\lambda}_{\mathbf{Y},1} - \tilde{\lambda}_{\mathbf{Y},M}$ as a function of variance σ^2 ($N = 10^5$, and $M = 4 \times 4$). (b) $\tilde{\lambda}_{\mathbf{Y},1} - \tilde{\lambda}_{\mathbf{Y},M}$ as a function of number of blocks N ($\sigma^2 = 5$, and $M = 4 \times 4$).

i.e. $\tilde{\lambda}_{\mathbf{Y},M}$ converges in mean to σ^2. Therefore, the noise variance can be estimated as $\tilde{\lambda}_{\mathbf{Y},M}$. Since convergence in mean implies convergence in probability, $\tilde{\lambda}_{\mathbf{Y},M}$ is a consistent estimator of the noise variance.

D. Check of Assumption 1

When Assumption 1 is right, we can compute the expected value of difference $\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M}$ by applying the triangle inequality and (1)

$$\mathbf{E}(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M}) = O(\sigma^2/\sqrt{N}). \qquad (3)$$

Hence, we have a necessary condition for the fulfillment of Assumption 1

$$\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} < T\sigma^2/\sqrt{N} \qquad (4)$$

where $T > 0$ is a fixed threshold, whose value is listed in Table II.

A question may arise whether condition (4) can be made stronger. Our experiments with image $\mathbf{x} = 0$ (see Fig. 2) show

TABLE I

VALUES OF ρ FOR THE REFERENCE IMAGES FROM THE TID2008 DATABASE (75 GRAYSCALE IMAGES). SECOND COLUMN IS THE SAMPLE MEAN, AND THE THIRD COLUMN IS THE SAMPLE STANDARD DEVIATION COMPUTED ACROSS ALL IMAGES. V_{M-m} WAS COMPUTED AS THE LINEAR SPAN OF $\tilde{\mathbf{v}}_{\mathbf{X},1}, \ldots, \tilde{\mathbf{v}}_{\mathbf{X},M-m}$. $M = 5 \times 5$

m	$\overline{\rho}$	$s(\rho)$
2	0.52	0.140
3	0.59	0.135
4	0.63	0.125
5	0.67	0.116
6	0.70	0.110
7	0.72	0.105
8	0.74	0.102
9	0.76	0.098
10	0.78	0.082
11	0.80	0.078
12	0.81	0.079
13	0.82	0.076
14	0.83	0.074
15	0.84	0.075

TABLE II

ALGORITHM PARAMETERS

M_1	5
M_2	5
C_0	3.1
p_0	0.0005
m	7
T	49
Δp	0.05
$p_{\min}$	0.06
$i_{\max}$	10

Fig. 3. Counterexample for selection of blocks with largest standard deviation.

that $\mathbf{E}(\tilde{\lambda}_{\mathbf{Y},1} - \tilde{\lambda}_{\mathbf{Y},M})$ fits to function $const \cdot \sigma^2 / \sqrt{N}$. Therefore, (3) is a tight upper bound, and (4) cannot be improved by changing the exponents of σ or N.

Then, if we have some estimate σ_{est}^2 of the noise variance, we can check (4) in order to check Assumption 1. If (4) holds, we take σ_{est}^2 as our final estimate. Otherwise, Assumption 1 is not satisfied, and we try to extract a subset of image blocks, for which Assumption 1 holds. This leads to an iterative procedure described in Subsection II-F.

E. Extraction of the Image Block Subset

As mentioned in the previous subsection, we need a strategy to extract a subset of image blocks, which satisfies Assumption 1. Let d_i be the distances of $\mathbf{x}_i$ to V_{M-m}, $i = 1, \ldots, N$. Assumption 1 holds, i.e. $\mathbf{x}_i \in V_{M-m}$, $i = 1, \ldots, N$, if and only if $d_i = 0$, $i = 1, \ldots, N$. Trying to satisfy this condition, it is reasonable to discard the blocks with the largest d_i from the total N image blocks.

Unfortunately, the values of d_i are not available in practice. Computation of the distances of $\mathbf{y}_i$ to V_{M-m} does not help, since a large distance of $\mathbf{y}_i$ to V_{M-m} can be caused by noise. Several heuristics may be applied therefore in order to select blocks with largest d_i, e.g. to pick blocks with largest standard deviation, largest range, or largest entropy. We use the first strategy, since it is fast to compute and the results are the most accurate in most cases. This strategy is examined below.

Let us consider the Spearman's rank correlation coefficient ρ between d_i and $s(\mathbf{x}_i)$, where $s(\mathbf{x}_i)$ is the sample standard deviation of elements of block $\mathbf{x}_i$, $i = 1, \ldots, N$. We have computed ρ for the reference images from the TID2008 database [39]. The results are shown in Table I. As one can see, there is a considerable positive correlation between d_i and $s(\mathbf{x}_i)$. That means large $s(\mathbf{x}_i)$ commonly corresponds to large d_i.

Since the noise is signal-independent, $s^2(\mathbf{x}_i)$ approximately equals $s^2(\mathbf{y}_i) - \sigma^2$. Hence large $s^2(\mathbf{y}_i)$ commonly corresponds to large d_i. As a result, we can discard blocks with the largest $s^2(\mathbf{y}_i)$ in order to discard blocks with the largest d_i.

The experiment presented in Table I shows that image structures, which are different from the general image texture, typically have a large local variance. However, this is not the case for all images. For example, the image shown in Fig. 3 consists of two parts: a stripe pattern on the left side, and a complicated texture on the right side. $s(\mathbf{x}_i) = 127.5$ and the mean of d_i is 0.1 for the blocks in the stripe pattern, but $s(\mathbf{x}_i) = 49.2$ and the mean of d_i is 9.1 for the blocks in the complicated texture even for $m = 1$. This synthetic example is unlikely for real-world images, but it shows that the heuristic we use cannot be proven.

F. Algorithm

Our noise variance estimation algorithm is presented in the main algorithm EstimateNoiseVariance, which calls GetUpperBound and GetNextEstimate. In these algorithms, $Q(p)$ is the p-quantile of $\{s^2(\mathbf{y}_i), i = 1, \ldots, N\}$ computed using Definition 3 from [40], and $B(p)$ is the subset of blocks of image $\mathbf{y}$, whose sample variance is not greater than $Q(p)$

$$B(p) = \{\mathbf{y}_i \mid s^2(\mathbf{y}_i) \le Q(p), \ i = 1, \ldots, N\}. \quad (5)$$

Algorithm EstimateNoiseVariance takes the result of algorithm GetUpperBound as the initial estimate and iteratively calls algorithm GetNextEstimate until convergence is reached. Parameter $i_{\max}$ is the maximum number of iterations. Its value is listed in Table II.

Algorithm GetUpperBound computes a noise variance upper bound. This algorithm is independent from image block

---

Algorithm 1 EstimateNoiseVariance

Input: Image $\mathbf{y}$
Output: Noise variance estimate σ_{est}^2

  1) $\sigma_{ub}^2 \leftarrow$ GetUpperBound($\mathbf{y}$)
  2) $\sigma_{est}^2 \leftarrow \sigma_{ub}^2$
  3) **for** $i = 1$ to $i_{\max}$ **do**
  4) $\sigma_{next}^2 \leftarrow$ GetNextEstimate($\mathbf{y}, \sigma_{est}^2, \sigma_{ub}^2$)
  5)   **if** $\sigma_{est}^2 = \sigma_{next}^2$ **then**
  6)   **return** σ_{est}^2
  7)   **end if**
  8) $\sigma_{est}^2 \leftarrow \sigma_{next}^2$
  9) **end for**
 10) **return** σ_{est}^2

Algorithm 2 GetNextEstimate

Input: Image $\mathbf{y}$, previous estimate σ_{est}^2, upper bound σ_{ub}^2
Output: Next estimate σ_{next}^2

  1) $p \leftarrow 1$
  2) $\sigma_{next}^2 \leftarrow 0$
  3) **while** $p \geq p_{\min}$
  4) $\tilde{\lambda}_{\mathbf{Y},1}, \ldots, \tilde{\lambda}_{\mathbf{Y},M} \leftarrow$ ApplyPCA($B(p)$)
  5) $\sigma_{next}^2 \leftarrow \tilde{\lambda}_{\mathbf{Y},M}$
  6)   **if** $\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} < T\sigma_{est}^2/\sqrt{|B(p)|}$ **and**
       $\sigma_{next}^2 \leq \sigma_{ub}^2$
  7)   **return** σ_{next}^2
  8)   **end if**
  9) $p \leftarrow p - \Delta p$
 10) **end while**
 11) **return** σ_{next}^2

PCA in order to increase the robustness of the whole procedure. Similar to many other noise estimation algorithms, it is based on the analysis of the image block variance distribution (see Section I). Namely, this algorithm returns $C_0 Q(p_0)$. The values of C_0 and p_0 are listed in Table II.

Algorithm GetNextEstimate extracts the subset of the image blocks, which satisfies Assumption 1. It implements the approaches described in Subsections II-D and II-E by taking p-quantiles of the block variance distribution. It starts from the largest possible p equal to 1, which corresponds to the whole set of image blocks. Then it discards blocks with the largest variance by reducing p to $1 - \Delta p$, $1 - 2\Delta p$, and so on, until p is smaller than $p_{\min}$. The values of $p_{\min}$, Δp, and T are listed in Table II. Upper bound σ_{ub}^2 is used as an additional check of the correctness of the computed estimate.

Algorithm ApplyPCA computes $\tilde{\lambda}_{\mathbf{Y},i}$, $i = 1, \ldots, M$.

G. Fast Implementation

When considering the execution time of the program, we have to concentrate on algorithm ApplyPCA, because it is called inside two loops: the first loop is in lines 3–9 of algorithm EstimateNoiseVariance; and the second loop is in lines 3–10 of algorithm GetNextEstimate. Algorithm ApplyPCA consists of two parts.

1) Computation of the sample covariance matrix

$$\frac{1}{|B(p)| - 1}$$
$$\times \left(\sum_{\mathbf{y}_i \in B(p)} \mathbf{y}_i \mathbf{y}_i^{\mathrm{T}} - \frac{1}{|B(p)|} \sum_{\mathbf{y}_i \in B(p)} \mathbf{y}_i \sum_{\mathbf{y}_i \in B(p)} \mathbf{y}_i^{\mathrm{T}} \right). \quad (6)$$

The number of operations is proportional to $|B(p)|M^2$.

2) Computation of the eigenvalues of the sample covariance matrix. The number of operations is proportional to M^3 [41].

Since $|B(p)| \gg M$, the computation of the sample covariance matrix is the most expensive part of algorithm ApplyPCA.

Let $C_X = \sum_{\mathbf{y}_i \in X} \mathbf{y}_i \mathbf{y}_i^{\mathrm{T}}$ and $\mathbf{c}_X = \sum_{\mathbf{y}_i \in X} \mathbf{y}_i$. Note that for disjoint sets X_1 and X_2, $C_{X_1 \cup X_2} = C_{X_1} + C_{X_2}$ and $\mathbf{c}_{X_1 \cup X_2} = \mathbf{c}_{X_1} + \mathbf{c}_{X_2}$. Then (6) can be represented as

$$\frac{1}{|B(p)| - 1} \left(C_{B(p)} - \frac{1}{|B(p)|} \mathbf{c}_{B(p)} \mathbf{c}_{B(p)}^{\mathrm{T}} \right). \quad (7)$$

Algorithm ApplyPCA is called only with arguments

$$B(1) \supset B(1 - \Delta p) \supset \cdots \supset B(1 - n\Delta p) \quad (8)$$

where $n = \lfloor (1 - p_{\min})/\Delta p \rfloor$ and $\lfloor x \rfloor$ is the largest integer not greater than x. For $j = 0, \ldots, n - 1$, let us consider sets

$$Y_j = \{\mathbf{y}_i \mid Q(1 - (j+1)\Delta p) < s^2(\mathbf{y}_i) \leq Q(1 - j\Delta p)\}. \quad (9)$$

Then, $B(1 - j\Delta p) = B(1 - (j+1)\Delta p) \cup Y_j$.

At the beginning of the program, we precompute matrices $C_{B(1-j\Delta p)}$ and vectors $\mathbf{c}_{B(1-j\Delta p)}$, $j = 0, \ldots, n$ in the following way. Matrices $C_{B(1-n\Delta p)}, C_{Y_0}, \ldots, C_{Y_{n-1}}$ and vectors $\mathbf{c}_{B(1-n\Delta p)}, \mathbf{c}_{Y_0}, \ldots, \mathbf{c}_{Y_{n-1}}$ are computed by definition and

$$C_{B(1-j\Delta p)} = C_{B(1-(j+1)\Delta p)} + C_{Y_j}$$
$$\mathbf{c}_{B(1-j\Delta p)} = \mathbf{c}_{B(1-(j+1)\Delta p)} + \mathbf{c}_{Y_j} \quad (10)$$

for $j = n - 1, \ldots, 0$. Then, these precomputed matrices and vectors are utilized in algorithm ApplyPCA when computing the sample covariance matrix using (7). When the precomputation is applied, the number of operations in (7) is proportional to M^2, which is $|B(p)|$ times smaller than in the direct implementation. Recursive procedure (10) ensures that the precomputation itself is optimal in the sense that expression $\mathbf{y}_i \mathbf{y}_i^{\mathrm{T}}$ is computed only once for each vector $\mathbf{y}_i$.

III. EXPERIMENTS

We have evaluated the accuracy and the speed of our method on two databases: TID2008 [39] and MeasTex [42].

The proposed algorithm has been compared with several recent methods:

1) methods which assume that the input image has a sufficient amount of homogeneous areas:

  a) [8], where Fisher's information is used in order to divide image blocks into two groups: homogeneous areas and textural areas

  b) [13], which applies a Sobel edge detection operator in order to exclude the noise-free image content

  c) [11], which applies Laplacian convolution and edge detection in order to find homogeneous areas

d) [18], which estimates the noise standard deviation as the median absolute deviation of the wavelet coefficients at the finest decomposition level. The Daubechies wavelet of length 8 has been used in the experiments

e) [21], where the noise variance is estimated as the mode of the distribution of local variances

f) [22], which divides the input image into blocks and computes the block standard deviations

g) [15], which subtracts low-frequency components detected by a Gaussian filter and edges detected by an edge detector from the input image. Since the method computes the noise variance as a function of the gray value, the estimates for all gray values have been averaged in order to compute the final estimate, as suggested by the authors during the personal discussion.

2) methods which use other assumptions about the input image:

a) [25], where nonlocal self-similarity of images is used in order to separate the noise from the signal

b) [24], where signal and noise separation is achieved with discrete cosine transform

c) [32], which treats the noise variance as a parameter of a Bayesian deblurring and denoising framework

d) [29], where the noise variance is estimated from a kurtosis model under the assumption that the kurtosis of marginal bandpass filter response distributions should be constant for a noise-free image

e) [28], which uses a measure of bit-plane randomness

f) [16], which utilizes multiresolution support data structure assuming that small wavelet transform coefficients correspond to the noise.

We have implemented our method both in C++ and Matlab in order to compare its execution time both with machine code and Matlab implementations of the others. The source code of both C++ and Matlab implementations is available at http://physics.medma.uni-heidelberg.de/cms/projects/132-pcanle.

A. Choice of the Parameters

We have tested our algorithm with different sets of the parameters; and we suggest the set presented in Table II. It has been used in all experiments in this section.

Regarding block size $M_1 \times M_2$, there is a trade-off between the ability to handle complicated textures and the statistical significance of the result. In order to satisfy Assumption 1, we need to find correlations between pixels of the image texture. Hence, the block size should be large enough, and, at least, be comparable to the size of the textural pattern. On the other hand, the block size cannot be arbitrary large. Since $\tilde{\lambda}_{\mathbf{Y},M}$ is the smallest order statistic of the sample eigenvalues representing the noise, its expected value for a finite N has a negative bias, which increases with $M = M_1 M_2$. Therefore, the block size should be as small as possible. 4×4, 5×5, and 6×6 blocks are good choices for real-world images of size

from 128×128 to 2048×2048; and we use 5×5 blocks in the experiments. When the horizontal and the vertical resolution of the input image are not equal, nonsquare blocks can be considered as well.

Parameters C_0 and p_0 have been chosen so that $\sigma_{ub}^2 = C_0 Q(p_0)$ is an upper bound of the true noise variance, i.e. this value always overestimates the noise level. Similar to [22] and [23], blocks with the smallest variances are used here, i.e. p_0 is close to 0. During the experiments with TID2008 and MeasTex, the output of our algorithm was never equal to σ_{ub}^2, hence it was always a PCA-based estimate.

Check (4) is robust when the difference between eigenvalue indices $M - m + 1$ and M is large, i.e. when m is large. However, if m is too large, $\tilde{\lambda}_{\mathbf{Y},M-m+1}$ is often influenced by the noise-free image data. $m = 7$ works properly for all images. The selection of T depends on the selection of m.

The results are not sensitive to parameter i_{max}, which can be selected from range $[3; +\infty)$. This parameter is needed only to guarantee that the algorithm always stops.

In 8-bit images with the gray value range $[0; 255]$, the noise is usually clipped. In order to prevent the influence of clipping to the noise level estimation process, we skip blocks, in which more than 10% of pixels have the value 0 or 255.

B. Noise Level Estimation Experiments With TID2008

The TID2008 database contains 25 RGB images. 24 of them are real-world scenes and one image is artificial. Each color component has been processed independently, i.e. the results for each noise level have been obtained using 75 grayscale images. Noisy images with the noise variance 65 and 130 are included in the database; and we have additionally tested our method with the noise variance 25 and 100. This database has been already applied for the evaluation of several other noise level estimation methods [8], [25], [43]. Some images from the database are shown in Fig. 4.

Though the reference images from TID2008 are considered as noise-free images, they still contain a small level of noise. This level should be estimated in order to compare all methods fairly. We have done this by using the following semi-automatic procedure:

1) Rectangular homogeneous area A has been selected manually in each reference image. This area contains almost no structure, i.e. it contains almost only noise with variance σ_{ref}^2. Therefore, the distribution of $\mathbf{x}(i, j)$, $(i, j) \in A$ can be approximated by $\mathcal{N}(\mu_A, \sigma_{ref}^2)$, where μ_A is the mean value of $\mathbf{x}(i, j)$ in A.

2) We have used a high-pass filter (see Section I) in order to remove possible image structures from A. Namely, we have considered the following differences between neighbor pixels:

a) $(\mathbf{x}(i+1, j) - \mathbf{x}(i, j))/\sqrt{2}$, if width$(A) >$ height(A),

b) $(\mathbf{x}(i, j+1) - \mathbf{x}(i, j))/\sqrt{2}$, if width$(A) \leq$ height(A),

where (i, j), $(i + 1, j)$, $(i, j + 1) \in A$.

3) Since $\mathbf{x}(i, j) \sim \mathcal{N}(\mu_A, \sigma_{ref}^2)$

$$(\mathbf{x}(i + 1, j) - \mathbf{x}(i, j))/\sqrt{2} \sim \mathcal{N}(0, \sigma_{ref}^2)$$

$$(\mathbf{x}(i, j + 1) - \mathbf{x}(i, j))/\sqrt{2} \sim \mathcal{N}(0, \sigma_{ref}^2) \quad (11)$$

Fig. 4. Images from the TID2008 database.

where $(i, j), (i + 1, j), (i, j + 1) \in A$. Therefore, the noise variance σ_{ref}^2 has been estimated as the variance of these differences.

The manually selected areas and the values of σ_{ref}^2 are available at http://physics.medma.uni-heidelberg.de/cms/projects/132-pcanle.

Then, all noise variance estimates have been corrected according to the following formula:

$$\sigma_{corr}^2 = \sigma_{est}^2 - \sigma_{ref}^2 \tag{12}$$

where σ_{est}^2 is the algorithm output.

The comparison results are presented in Tables III and IV. Since the execution time of [25] is not provided by the author, we have measured the execution time of the BM3D filter [2], which is a step of [25].

As one can see, the accuracy of the proposed method is the highest in most cases. The results of [25] and [8] are comparable to our results, but these approaches are much slower than the proposed method: [25] is more than 15 times slower; and [8] is about 50–180 times slower (assuming that the hardware speed difference is about 2 times).

The method [24] has 2 times larger $s(\sigma_{corr})$ and max $|\sigma_{corr} - \sigma|$ than the proposed method. The methods [11], [13], [15], [16], [18], [21], [22], [28], [29], [32] have more than 3 times larger $s(\sigma_{corr})$ and more than 4 times larger max $|\sigma_{corr} - \sigma|$ compared with the proposed method. The bias of these methods is much larger than that of our method in most cases.

C. Noise Level Estimation Experiments With MeasTex

All images in the TID2008 database contain small or large homogeneous areas. However, this is not the case for all images one can meet. For this reason, we have tested our method on images containing only textures. We have selected the MeasTex texture database, which has been already used in many works on texture analysis [42], [44], [45]. This database contains 236 real textures stored as 512×512 grayscale images. Several images from the database are shown in Fig. 5.

The comparison results are presented in Table V. Compared with other methods, the accuracy of the proposed method is significantly better in all cases: the standard deviation of the estimates is always more than 2.8 times smaller; and the maximum error is always more than 2.2 times

TABLE III

ACCURACY OF THE CONSIDERED METHODS FOR TID2008. $\overline{\sigma}_{corr} - \sigma$ IS THE BIAS OF CORRECTED ESTIMATES, $s(\sigma_{corr})$ IS THE STANDARD DEVIATION OF CORRECTED ESTIMATES, AND max $|\sigma_{corr} - \sigma|$ IS THE MAXIMUM DIFFERENCE BETWEEN A CORRECTED ESTIMATE AND THE TRUE VALUE. LAST COLUMN SHOWS PERCENTAGE OF IMAGES FOR WHICH A METHOD CANNOT ESTIMATE NOISE LEVEL. BEST RESULTS ARE IN BOLD. *VALUES OF σ_{est} HAVE BEEN PROVIDED BY THE AUTHORS

Method	$\overline{\sigma}_{corr} - \sigma$	$s(\sigma_{corr})$	max $|\sigma_{corr} - \sigma|$	% of Failures
$\sigma^2 = 25\ (\sigma = 5)$				
Proposed	**−0.027**	**0.147**	**0.500**	0
[8]	−	−	−	−
[13]	0.322	0.547	2.859	0
[11]	0.605	0.882	4.116	0
[18]	1.127	1.030	5.194	0
[21]	0.617	1.530	8.941	0
[22]	−1.499	1.822	5.000	57.3
[15]	4.954	3.408	21.037	0
[25]*	−0.039	0.158	0.525	0
[24]	−	−	−	−
[32]	−0.345	0.857	3.507	0
[29]	−0.487	3.323	24.719	1.3
[28]	3.227	2.266	9.158	0
[16]	2.144	2.224	8.903	0
$\sigma^2 = 65\ (\sigma \approx 8.062)$				
Proposed	−0.043	**0.103**	0.486	0
[8]*	−0.074	0.110	**0.401**	0
[13]	0.228	0.430	2.093	0
[11]	0.206	0.769	2.867	0
[18]	0.724	1.003	4.281	0
[21]	0.292	1.526	6.343	0
[22]	−1.467	2.044	8.062	45.3
[15]	4.049	3.290	19.557	0
[25]	−	−	−	−
[24]*	**0.001**	0.209	1.078	0
[32]	−0.858	0.971	4.211	0
[29]	−0.899	1.384	8.062	0
[28]	3.173	1.671	8.968	0
[16]	2.067	2.325	10.160	0
$\sigma^2 = 100\ (\sigma = 10)$				
Proposed	**0.009**	**0.125**	**0.307**	0
[8]	−	−	−	−
[13]	0.232	0.412	1.935	0
[11]	0.269	0.640	3.088	0
[18]	0.819	0.900	4.011	0
[21]	0.582	1.061	6.019	0
[22]	−1.517	2.145	10.000	42.7
[15]	3.553	3.111	20.238	0
[25]*	0.040	0.175	0.717	0
[24]	−	−	−	−
[32]	−0.746	0.750	2.400	0
[29]	−0.395	2.749	20.956	0
[28]	2.204	2.519	9.551	0
[16]	2.248	1.868	7.281	0
$\sigma^2 = 130\ (\sigma \approx 11.402)$				
Proposed	**0.014**	**0.110**	**0.386**	0
[8]*	−0.040	0.136	0.532	0
[13]	0.224	0.390	1.943	0
[11]	−0.025	0.777	3.297	0
[18]	0.477	0.989	3.711	0
[21]	0.250	1.464	5.665	0
[22]	−1.467	2.086	6.811	41.3
[15]	3.325	3.506	21.502	0
[25]	−	−	−	−
[24]*	0.094	0.228	1.170	0
[32]	−1.140	1.062	5.351	0
[29]	−0.634	2.634	19.321	0
[28]	2.700	2.510	9.604	0
[16]	2.132	2.279	10.197	0

TABLE IV

EXECUTION TIME OF THE CONSIDERED METHODS FOR TID2008. min t IS THE MINIMUM EXECUTION TIME, $\overline{t}$ IS THE AVERAGE EXECUTION TIME, max t IS THE MAXIMUM EXECUTION TIME. ALL IMPLEMENTATIONS ARE SINGLE-THREADED. *VALUES HAVE BEEN PROVIDED BY THE AUTHORS

Method	Machine Code (C++, Object Pascal)			Matlab			CPU
	min t	$\overline{t}$	max t	min t	$\overline{t}$	max t	
Proposed	102 ms	159 ms	169 ms	911 ms	1491 ms	1583 ms	Intel i7 920 2.67 GHz
[8]*	–	–	–	3 min	–	10 min	Intel Core 2 Duo 1.66 GHz
[13]	1.9 ms	3.1 ms	3.5 ms	–	–	–	Intel i7 920 2.67 GHz
[11]	–	–	–	635 ms	682 ms	1608 ms	Intel i7 920 2.67 GHz
[18]	–	–	–	36 ms	38 ms	43 ms	Intel i7 920 2.67 GHz
[21]	–	–	–	62 ms	73 ms	402 ms	Intel i7 920 2.67 GHz
[22]	1.0 ms	1.1 ms	18.8 ms	–	–	–	Intel i7 920 2.67 GHz
[15]	–	–	–	532 ms	681 ms	1621 ms	Intel i7 920 2.67 GHz
[25] ([2])	2628 ms	–	–	–	–	–	Intel i7 920 2.67 GHz
[24]*	∼ 250 ms			–	–	–	Intel Celeron 1.4 GHz
[32]	–	–	–	3.7 min	4.8 min	10 min	Intel i7 920 2.67 GHz
[29]	–	–	–	2968 ms	2968 ms	3373 ms	Intel i7 920 2.67 GHz
[28]	80 ms	123 ms	140 ms	–	–	–	Intel i7 920 2.67 GHz
[16]	520 ms	805 ms	980 ms	–	–	–	Intel i7 920 2.67 GHz

Fig. 5.　Images from the MeasTex database.

(a)

(b)

Fig. 6.　Images for which the proposed method has an error larger than $3s(\sigma_{\text{est}})$. (a) Fabric textures. (b) Metal textures.
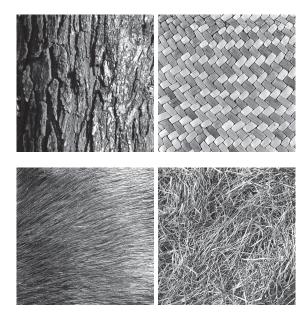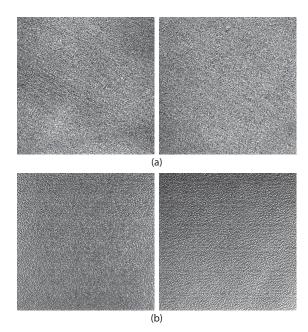
smaller. The bias of our method is much smaller in most cases as well. The results of [8], [25], and [24] are not available.

This experiment also shows the limitations of the proposed method. Among 236 images in the database, there are 4 images, for which the error is larger than $3s(\sigma_{\text{est}})$. These images are textures of fabric and metal (see Fig. 6). In order to examine the properties of these images, we have computed the following measure for each image $\mathbf{x}$ in the database:

$$\overline{R}_{\mathbf{x}} = \frac{1}{24}\left(\sum_{\Delta i=0}^{4}\sum_{\Delta j=0}^{4}|R_{\mathbf{x}}(\Delta i, \Delta j)| - |R_{\mathbf{x}}(0,0)|\right) \quad (13)$$

where $R_{\mathbf{x}}$ is the autocorrelation function:

$$R_{\mathbf{x}}(\Delta i, \Delta j) = \frac{1}{\sigma_{\mathbf{x}}^2 N_{\mathbf{x}}} \sum_{i,j}(\mathbf{x}(i, j) - \mu_{\mathbf{x}})$$
$$\times (\mathbf{x}(i + \Delta i, j + \Delta j) - \mu_{\mathbf{x}}). \quad (14)$$

Above, the sum is computed over all i and j such that (i, j) and $(i + \Delta i, j + \Delta j)$ are inside the image domain, $N_{\mathbf{x}}$ is the number of items in this sum, $\sigma_{\mathbf{x}}^2$ is the variance of image $\mathbf{x}$, and $\mu_{\mathbf{x}}$ is the mean of image $\mathbf{x}$. $\overline{R}_{\mathbf{x}}$ is always in $[0; 1]$. It reflects the correlation between neighbor image pixels: $\overline{R}_{\mathbf{x}} = 0$ when $\mathbf{x}$ is white noise; and $\overline{R}_{\mathbf{x}} = 1$ when the neighbor pixels in $\mathbf{x}$

TABLE V

ACCURACY OF THE CONSIDERED METHODS FOR MEASTEX. $\overline{\sigma}_{\text{est}} - \sigma$ IS THE BIAS OF ESTIMATES, $s(\sigma_{\text{est}})$ IS THE STANDARD DEVIATION OF ESTIMATES, AND $\max |\sigma_{\text{est}} - \sigma|$ IS THE MAXIMUM DIFFERENCE BETWEEN AN ESTIMATE AND THE TRUE VALUE. LAST COLUMN SHOWS PERCENTAGE OF IMAGES FOR WHICH A METHOD CANNOT ESTIMATE NOISE LEVEL. BEST RESULTS ARE IN BOLD

Method	$\overline{\sigma}_{\text{est}} - \sigma$	$s(\sigma_{\text{est}})$	$\max |\sigma_{\text{est}} - \sigma|$	% of Failures
$\sigma = 10$
Proposed	**0.283**	**0.845**	**7.235**	0
[8]	–	–	–	–
[13]	1.075	2.418	16.520	0
[11]	0.997	2.746	17.689	0
[18]	1.662	3.486	21.557	0
[21]	6.426	10.334	56.555	0
[22]	1.947	8.118	38.551	68.2
[15]	5.129	5.241	38.756	0.4
[25]	–	–	–	–
[24]	–	–	–	–
[32]	0.489	5.033	37.958	0
[29]	−0.648	2.799	23.646	0.4
[28]	6.917	10.215	75.213	0
[16]	6.014	9.328	47.079	0
$\sigma = 15$
Proposed	0.170	**0.592**	**4.868**	0
[8]	–	–	–	–
[13]	0.836	2.007	13.554	0
[11]	0.571	2.266	14.814	0
[18]	1.291	2.969	18.605	0
[21]	5.709	9.294	48.667	0
[22]	1.858	7.238	34.703	57.6
[15]	3.748	4.698	34.986	1.3
[25]	–	–	–	–
[24]	–	–	–	–
[32]	**−0.053**	4.803	36.573	0
[29]	−0.753	2.772	20.942	0
[28]	8.433	10.797	74.195	0
[16]	5.539	8.388	43.381	0
$\sigma = 20$
Proposed	**0.080**	**0.461**	**3.084**	0
[8]	–	–	–	–
[13]	0.625	1.727	11.382	0
[11]	0.239	1.932	12.540	0
[18]	1.015	2.573	16.240	0
[21]	5.054	8.594	46.388	0
[22]	1.238	7.240	31.465	47.0
[15]	2.801	4.641	40.440	3.0
[25]	–	–	–	–
[24]	–	–	–	–
[32]	−0.295	4.656	34.860	0
[29]	−0.870	2.398	20.000	1.3
[28]	10.607	12.944	73.093	0
[16]	5.199	7.615	40.076	0

are in an exact linear dependence. For MeasTex, the average value of $\overline{R}_{\mathbf{x}}$ is 0.6 and the standard deviation is 0.2. For the images shown in Fig. 6, $\overline{R}_{\mathbf{x}}$ takes its smallest values: 0.08,

0.07, 0.06, and 0.09. Therefore, these images are the closest to white noise in the database, which explains the largest error of our algorithm for these images.

D. Denoising Experiments

Finally, we tested the noise level estimation algorithms in a denoising application. We utilized the denoising method [2], which outperforms the methods [46]–[49] and can be considered as the state of the art in image denoising. The results are presented in Table VI. Due to the article length limitation, only the minimal value of the peak signal-to-noise ratio (PSNR) over each image database is presented. However, this value represents the behavior of a noise level estimator in the worst case and, therefore, the applicability of the estimator. Note that the minimal value is undefined for the methods which fail on some images.

Regarding the TID2008 database, the methods [8], [25] and the proposed method result in roughly the same denoising quality as the true noise levels. The results of the method [29] are good for high noise levels, but become significantly worse as the noise level decreases. The minimal PSNR for the other methods is more than 0.4 dB lower than that for the proposed method. For the noise variances 65, 100, and 130, the results with our algorithm or the method [8] are slightly better than the results with the true noise levels, because the denoising algorithm [2] achieves the best PSNR at some noise level, which is close to the true value but not exactly equal to it.

Speaking of the MeasTex database, the minimal PSNR with our method is smaller than the noisy image PSNR when σ is low. For $\sigma = 20$, the minimal PSNR with our algorithm is better than that for the noisy images but worse than that with true σ. This reflects the large error of the proposed method on stochastic textures discussed in the previous subsection. For all noise levels, our algorithm results in at least 2 dB higher minimal PSNR than the other methods.

In general, the denoising results show that a higher noise level estimation accuracy leads to a higher denoising quality in most cases.

IV. DISCUSSION

The proposed method does not assume the existence of homogeneous areas in the input image. Instead, it belongs to the class of algorithms, which are based on a sparse image representation. One of the first methods from this class is [18], whereas [24] and [25] are more recent.

The method [18] utilizes a wavelet transform. This approach does not work well for images with textures, because textures usually contain high frequencies and affect the finest decomposition level, from which the noise variance is estimated [19]. It was outperformed by other techniques, e.g. [11], [24], [25], [19], and [13].

The method [24] applies 2-D DCT of image blocks. It assumes that the image structures occupy only low frequencies and high frequencies contain only noise. Compared with this method, the proposed algorithm uses PCA instead of DCT. The transform computed by PCA depends on the data in contrast

TABLE VI

MINIMAL PSNR FOR DIFFERENT NOISE LEVEL ESTIMATORS. FOR TID2008, CORRECTED ESTIMATES σ_{corr} WERE USED. THE ROW
"NOISY IMAGE PSNR" SHOWS THE PSNR BEFORE DENOSING. THE ROW "TRUE σ" SHOWS THE PSNR WHEN THE
TRUE NOISE LEVEL IS PASSED TO THE DENOISER. THE METHODS WHICH HAVE FAILURES ARE MARKED
WITH "F." BEST RESULTS ARE MARKED IN BOLD

Method	TID2008				MeasTex		
	$\sigma^2 = 25$ ($\sigma = 5$)	$\sigma^2 = 65$ ($\sigma \approx 8.062$)	$\sigma^2 = 100$ ($\sigma = 10$)	$\sigma^2 = 130$ ($\sigma \approx 11.402$)	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$
Noisy image PSNR	*34.15*	*30.00*	*28.13*	*26.99*	*28.13*	*24.61*	*22.11*
True σ	*35.08*	*31.53*	*30.05*	*29.08*	*28.42*	*25.20*	*22.98*
Proposed	**35.03**	31.54	**30.06**	29.12	**26.99**	**24.53**	**22.67**
[8]	–	**31.57**	–	**29.14**	–	–	–
[13]	33.21	30.43	29.24	28.36	21.87	21.08	20.25
[11]	32.21	30.09	29.03	28.34	22.30	21.35	20.41
[18]	30.75	29.10	28.26	27.66	20.71	20.12	19.46
[21]	28.78	28.26	27.62	27.08	15.47	15.50	15.38
[22]	F	F	F	F	F	F	F
[15]	29.53	28.38	27.72	27.36	F	F	F
[25]	34.97	–	29.78	–	–	–	–
[24]	–	31.05	–	28.67	–	–	–
[32]	32.36	30.11	28.99	28.31	16.50	16.19	15.86
[29]	F	30.68	30.03	29.11	F	19.63	F
[28]	28.50	27.11	26.28	25.77	14.80	14.69	14.58
[16]	28.64	27.65	27.06	26.60	15.88	15.77	15.64

to DCT, which is predefined. Therefore, PCA can efficiently process a larger class of images, including those which contain structures with high frequencies. The evidence can be found in Table III: the maximum error of [24] is more than two times larger than that of the proposed method, i.e. PCA can handle some images in the TID2008 database much more efficiently than DCT.

Compared with [25], which assumes the existence of similar blocks in the image, our approach assumes the correlation between pixels in image blocks. These two assumptions cannot be compared directly, because there are images which satisfy the first and does not satisfy the second and vice versa. Indeed, the experiments with TID2008 demonstrate only a small improvement of the results compared with [25]. A more significant difference is in the execution time: [25] has expensive steps of block matching and image prefiltering, which make it more than 15 times slower than our algorithm.

The proposed method can be generalized to 3-D images, which are acquired e.g. by magnetic resonance scanners. In this case, we have 3-D blocks of size $M_1 \times M_2 \times M_3$, which are rearranged into vectors of size $M = M_1 M_2 M_3$. Then, these vectors can be processed in the same way as for 2-D images.

Although the proposed method is developed for signal-independent additive white Gaussian noise, it can be applied to other noise types by utilizing a variance-stabilizing transformation (VST). The VST transforms the input image into an image, whose noise is approximately additive white Gaussian [50]. For example, the VST derived in [51] is used for Rician noise, and the VST from [52] is used for speckle noise. Then, the noise level in the transformed image can be estimated by

the proposed method, and the level of the original noise can be computed.

Our algorithm has some optimization potential. For example, the following parts of the program can be efficiently parallelized:

1) the block variance computation, since it is done for each block independently;
2) the computation of matrices $C_{B(1-n\Delta p)}, C_{Y_0}, \ldots, C_{Y_{n-1}}$ and vectors $\mathbf{c}_{B(1-n\Delta p)}, \mathbf{c}_{Y_0}, \ldots, \mathbf{c}_{Y_{n-1}}$, because they are independent from each other.

V. CONCLUSION

In this work, we have presented a new noise level estimation algorithm. The comparison with the several best state of the art methods shows that the accuracy of the proposed approach is the highest in most cases. Among the methods with similar accuracy, our algorithm is always more than 15 times faster.

Since the proposed method does not require the existence of homogeneous areas in the input image, it can also be applied to textures. Our experiments show that only stochastic textures, whose correlation properties are very close to those of white noise, cannot be successfully processed.

During our denoising experiments, we observed that a higher noise level estimation accuracy leads to a higher denoising quality in most cases. It shows the importance of a careful selection of the noise estimator in a denoising application. We also observed that the denoising quality with our algorithm was approximately the same as that with the true noise level if the image was not a stochastic texture; hence the proposed method can be successfully applied in image denoising. Our

approach can also be utilized in image compression and segmentation applications which require noise level estimation.

APPENDIX
PROOF OF THEOREM 1

Further, let $\|A\|$ be the spectral norm of matrix A, and $\mathbf{1}_X$ be the indicator function of set X.

Results on the probability distributions for sample principal components were derived in [53] under the assumption that the original random vector has a multivariate normal distribution. However, random vector $\mathbf{X}$ can have an arbitrary distribution, therefore these results cannot be applied here.

Given population covariance matrix $\Sigma_{\mathbf{Y}}$, sample covariance matrix $S_{\mathbf{Y}}$ can be seen as the sum of $\Sigma_{\mathbf{Y}}$ and small perturbation $S_{\mathbf{Y}} - \Sigma_{\mathbf{Y}}$. Therefore, our proof is based on matrix perturbation theory. Specifically, we need to estimate the perturbation of the eigenvalues. This value can be bounded by $\|S_{\mathbf{Y}} - \Sigma_{\mathbf{Y}}\|$ (Lemma 1 below), which is not a tight bound in our case. Theorem 2.3 in [54, p. 183] gives an estimate with accuracy $\|S_{\mathbf{Y}} - \Sigma_{\mathbf{Y}}\|^2$, but it can be applied only for eigenvalues with multiplicity 1, which is not the case when Assumption 1 holds. In [55, p. 76], this result was extended to the case of eigenvalues with an arbitrary multiplicity. However, the formulation in [55] has some restrictions and cannot be applied here directly. For this reason, we use our own formulation (Lemma 2 below), whose proof has the same steps as those in [54] and [55].

Lemma 1: Let $A \in \mathbb{C}^{M \times M}$ and $B \in \mathbb{C}^{M \times M}$ be Hermitian matrices, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ be the eigenvalues of A, and $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_M$ be the eigenvalues of perturbed matrix $\tilde{A} = A + B$. Then $\forall i = 1, \ldots, M \; |\tilde{\lambda}_i - \lambda_i| \leq \|B\|$.

Proof: See [54, p. 203]. ∎

Lemma 2: Let $A \in \mathbb{C}^{M \times M}$ and $B \in \mathbb{C}^{M \times M}$ be Hermitian matrices, λ be an eigenvalue of A with multiplicity $\overline{m}$ and normalized eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_{\overline{m}}$, $\delta > 0$ be the minimum of the distances between λ and the other eigenvalues of A, and $\tilde{A} = A + B$ be a perturbed matrix. If $\|B\| < \frac{\delta}{4M}$ then there are exactly $\overline{m}$ eigenvalues $\tilde{\lambda}_k$ of $\tilde{A}$, which satisfy

$$|\tilde{\lambda}_k - \lambda| \leq \max_{i=1,\ldots,\overline{m}} \sum_{j=1}^{\overline{m}} |\mathbf{v}_i^{\mathrm{T}} B \mathbf{v}_j| + \frac{4M^2}{\delta} \|B\|^2. \quad (15)$$

Proof: See [54, p. 183] and [55, p. 76]. ∎

Further, let

$$B = S_{\mathbf{Y}} - \Sigma_{\mathbf{Y}}$$
$$d = \frac{\delta}{4M}$$
$$\mu_{\alpha\beta}^{(ij)} = \mathbf{E}\big((y_i - \mathbf{E}(y_i))^\alpha (y_j - \mathbf{E}(y_j))^\beta\big)$$
$$K = \max_{i,j=1,\ldots,M} \mu_{22}^{(ij)} \quad (16)$$

where y_i are the entries of $\mathbf{Y}$. Note that K depends only on the distribution of $\mathbf{Y}$ and is independent of N.

Lemma 3: In the conditions of Theorem 1

$$\mathbf{E}(\|B\|^2) = O\left(\frac{K}{N}\right). \quad (17)$$

Proof: Let b_{ij} be the entries of B. Since $\mathbf{E}(S_{\mathbf{Y}}) = \Sigma_{\mathbf{Y}}$, $\mathbf{E}(b_{ij}) = 0$. From [56], [57]

$$
\begin{aligned}
\mathrm{var}(b_{ij}) &= \frac{\mu_{22}^{(ij)}}{N} + \frac{\mu_{20}^{(ij)} \mu_{02}^{(ij)}}{N^2 - N} - \frac{N-2}{N^2 - N}\big(\mu_{11}^{(ij)}\big)^2 \\
&\leq \frac{\mu_{22}^{(ij)}}{N} + \frac{\mu_{20}^{(ij)} \mu_{02}^{(ij)}}{N^2 - N} \\
&= O\left(\frac{K}{N}\right)
\end{aligned}
\quad (18)
$$

because $1/(N^2 - N)$ is infinitesimal compared with $1/N$. Therefore

$$
\begin{aligned}
\mathbf{E}(\|B\|^2) &\leq \mathbf{E}\left(\sum_{i,j=1}^{M} b_{ij}^2\right) = \sum_{i,j=1}^{M} \mathrm{var}(b_{ij}) \\
&= O\left(\frac{K}{N}\right).
\end{aligned}
\quad (19)
$$
∎

Lemma 4: In the conditions of Theorem 1

$$\mathbf{E}(\|B\| \mathbf{1}_{\|B\| \geq d}) = O\left(\frac{K}{N}\right) \quad (20)$$

Proof: Let $\overline{F}_{\|B\|}(x)$ be the complementary cumulative distribution function of $\|B\|$. Since $\overline{F}_{\|B\|}(x) \leq \mathbf{E}(\|B\|^2)/x^2$ from Chebyshev's inequality, then (see [58])

$$
\begin{aligned}
\mathbf{E}(\|B\| \mathbf{1}_{\|B\| \geq d}) &= d\overline{F}_{\|B\|}(d) + \int_d^{+\infty} \overline{F}_{\|B\|}(x) dx \\
&\leq \frac{\mathbf{E}(\|B\|^2)}{d} + \mathbf{E}(\|B\|^2) \int_d^{+\infty} \frac{dx}{x^2} \\
&= \frac{2\mathbf{E}(\|B\|^2)}{d}
\end{aligned}
$$

Then (20) follows from (17). ∎

Proof: [Proof of Theorem 1] Let $\lambda_{\mathbf{X},1} \geq \lambda_{\mathbf{X},2} \geq \ldots \geq \lambda_{\mathbf{X},M}$ be the eigenvalues of $\Sigma_{\mathbf{X}}$ with the corresponding normalized eigenvectors $\mathbf{v}_{\mathbf{X},1}, \ldots, \mathbf{v}_{\mathbf{X},M}$. Since

$$\Sigma_{\mathbf{Y}} \mathbf{v}_{\mathbf{X},i} = \Sigma_{\mathbf{X}} \mathbf{v}_{\mathbf{X},i} + \sigma^2 \mathbf{v}_{\mathbf{X},i} = (\lambda_{\mathbf{X},i} + \sigma^2) \mathbf{v}_{\mathbf{X},i}, \quad (21)$$

each eigenvalue of $\Sigma_{\mathbf{Y}}$ equals the sum of an eigenvalue of $\Sigma_{\mathbf{X}}$ and σ^2, and the eigenvectors of $\Sigma_{\mathbf{X}}$ and $\Sigma_{\mathbf{Y}}$ are the same. Under Assumption 1, the last $\overline{m} \geq m$ eigenvalues of $\Sigma_{\mathbf{X}}$ are zeros, therefore, the last $\overline{m}$ eigenvalues of $\Sigma_{\mathbf{Y}}$ equal σ^2.

Let $J = \{M - \overline{m} + 1, \ldots, M\}$ be the set of indices of zero eigenvalues of $\Sigma_{\mathbf{X}}$. Using Lemma 2 with matrices $\Sigma_{\mathbf{Y}}$ and B, for $k \in J$

$$
\begin{aligned}
\mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2| \mathbf{1}_{\|B\| < d}) &\leq \mathbf{E}\left(\max_{i \in J} \sum_{j \in J} |\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} B \mathbf{v}_{\mathbf{X},j}|\right) \\
&\quad + \frac{4M^2}{\delta} \mathbf{E}(\|B\|^2).
\end{aligned}
\quad (22)
$$

Consider the first summand on the right side of (22). Denoting the sample covariance by q, for $i, j \in J$ we

have

$$\begin{aligned}
\mathbf{E}\big((\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} B \mathbf{v}_{\mathbf{X},j})^2\big) &= \mathrm{var}(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} B \mathbf{v}_{\mathbf{X},j}) \\
&= \mathrm{var}(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} S_{\mathbf{Y}} \mathbf{v}_{\mathbf{X},j}) \\
&= \mathrm{var}(q(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} \mathbf{Y}, \mathbf{v}_{\mathbf{X},j}^{\mathrm{T}} \mathbf{Y})) \\
&= \mathrm{var}(q(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} \mathbf{N}, \mathbf{v}_{\mathbf{X},j}^{\mathrm{T}} \mathbf{N}))
\end{aligned} \tag{23}$$

because $\mathbf{E}(B)=0$ and $\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}}\mathbf{X}=\mathbf{v}_{\mathbf{X},j}^{\mathrm{T}}\mathbf{X}=0$. From [56], [57]

$$\begin{aligned}
\mathrm{var}(q(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} \mathbf{N}, \mathbf{v}_{\mathbf{X},j}^{\mathrm{T}} \mathbf{N})) &= \frac{\mathbf{E}\big((\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}}\mathbf{N})^2(\mathbf{v}_{\mathbf{X},j}^{\mathrm{T}}\mathbf{N})^2\big)}{N} \\
&+ \frac{\mathrm{var}(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}}\mathbf{N})\mathrm{var}(\mathbf{v}_{\mathbf{X},j}^{\mathrm{T}}\mathbf{N})}{N^2-N} \\
&- \frac{N-2}{N^2-N}\mathbf{E}(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}}\mathbf{N}\cdot\mathbf{v}_{\mathbf{X},j}^{\mathrm{T}}\mathbf{N})^2.
\end{aligned} \tag{24}$$

The entries of $\mathbf{N}$ are distributed as $\mathcal{N}(0,\sigma^2)$, so that their variance equals σ^2 and the forth moment equals $3\sigma^4$. Hence

$$\mathrm{var}(q(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}}\mathbf{N}, \mathbf{v}_{\mathbf{X},j}^{\mathrm{T}}\mathbf{N})) = O\Big(\frac{\sigma^4}{N}\Big). \tag{25}$$

Then, using the matrix norm inequalities and the fact that $\mathbf{E}(\sqrt{X}) \le \sqrt{\mathbf{E}(X)}$

$$\begin{aligned}
\mathbf{E}\Big(\max_{i\in J}\sum_{j\in J}|\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} B \mathbf{v}_{\mathbf{X},j}|\Big) &\le \overline{m}\mathbf{E}\Big(\sqrt{\sum_{i,j\in J}(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} B \mathbf{v}_{\mathbf{X},j})^2}\Big) \\
&\le \overline{m}\sqrt{\sum_{i,j\in J}\mathbf{E}\big((\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}} B \mathbf{v}_{\mathbf{X},j})^2\big)} \\
&= O\Big(\frac{\sigma^2}{\sqrt{N}}\Big)
\end{aligned} \tag{26}$$

from (23) and (25).

Utilizing Lemmas 1 and 4

$$\begin{aligned}
\mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\mathbf{1}_{\|B\|\ge d}) &\le \mathbf{E}(\|B\|\mathbf{1}_{\|B\|\ge d}) \\
&= O\Big(\frac{K}{N}\Big).
\end{aligned} \tag{27}$$

In order to get the final result, we should combine bounds (17), (22), (26), and (27). Since $N\to\infty$, $1/N$ is infinitesimal compared with $1/\sqrt{N}$, and

$$\begin{aligned}
\mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|) &= \mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\mathbf{1}_{\|B\|<d}) \\
&+ \mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\mathbf{1}_{\|B\|\ge d}) \\
&= O\Big(\frac{\sigma^2}{\sqrt{N}}\Big) + O\Big(\frac{K}{N}\Big) + O\Big(\frac{K}{N}\Big) \\
&= O\Big(\frac{\sigma^2}{\sqrt{N}}\Big).
\end{aligned} \tag{28}$$

∎

REFERENCES

[1] A. Buades, B. Coll, and J. Morel, "On image denoising methods," *SIAM Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
[3] J. Walker, "Combined image compressor and denoiser based on tree-adapted wavelet shrinkage," *Opt. Eng.*, vol. 41, no. 7, pp. 1520–1527, 2002.
[4] P. Rosin, "Thresholding for change detection," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 274–279.
[5] M. Salmeri, A. Mencattini, E. Ricci, and A. Salsano, "Noise estimation in digital images using fuzzy processing," in *Proc. Int. Conf. Image Process.*, vol. 1. 2001, pp. 517–520.
[6] D. Shin, R. Park, S. Yang, and J. Jung, "Block-based noise estimation using adaptive Gaussian filtering," *IEEE Trans. Consum. Electron.*, vol. 51, no. 1, pp. 218–226, Feb. 2005.
[7] S. Abramov, V. Lukin, B. Vozel, K. Chehdi, and J. Astola, "Segmentation-based method for blind evaluation of noise variance in images," *J. Appl. Remote Sensing*, vol. 2, p. 023533, Jan. 2008.
[8] M. Uss, B. Vozel, V. Lukin, S. Abramov, I. Baryshev, and K. Chehdi, "Image informative maps for estimating noise standard deviation and texture parameters," *EURASIP J. Adv. Signal Process.*, vol. 2011, pp. 806516-1–806516-12, Mar. 2011.
[9] M. Liévin, F. Luthon, and E. Keeve, "Entropic estimation of noise for medical volume restoration," in *Proc. 16th Int. Conf. Pattern Recognit.*, vol. 3. 2002, pp. 871–874.
[10] B. Corner, R. Narayanan, and S. Reichenbach, "Noise estimation in remote sensing imagery using data masking," *Int. J. Remote Sensing*, vol. 24, no. 4, pp. 689–702, 2003.
[11] R. Bilcu and M. Vehvilainen, "New method for noise estimation in images," in *Proc. IEEE-Eurasip Nonlinear Signal Image Process.*, May 2005, p. 25.
[12] F. Russo, "Gaussian noise estimation in digital images using nonlinear sharpening and genetic optimization," in *Proc. Instrum. Meas. Technol. Conf.*, 2007, pp. 1–5.
[13] S. Yang and S. Tai, "Fast and reliable image-noise estimation using a hybrid approach," *J. Electron. Imag.*, vol. 19, no. 3, pp. 033007-1–033007-15, 2010.
[14] A. Liu, "A fast method of estimating Gaussian noise," in *Proc. 1st Int. Conf. Inf. Sci. Eng.*, 2009, pp. 441–444.
[15] M. Salmeri, A. Mencattini, G. Rabottino, and R. Lojacono, "Signal-dependent noise characterization for mammographic images denoising," presented at the IMEKO TC4 Symp., Florence, Italy, 2008.
[16] J. Starck and F. Murtagh, "Automatic noise estimation from the multiresolution support," *Publ. Astron. Soc. Pacific*, vol. 110, no. 744, pp. 193–199, 1998.
[17] A. De Stefano, P. White, and W. Collis, "Training methods for image noise level estimation on wavelet components," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 2400–2407, Jan. 2004.
[18] D. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.
[19] M. Hashemi and S. Beheshti, "Adaptive noise variance estimation in bayesshrink," *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 12–15, Jan. 2010.
[20] L. Alparone, M. Selva, L. Capobianco, S. Moretti, L. Chiarantini, and F. Butera, "Quality assessment of data products from a new generation airborne imaging spectrometer," in *Proc. IEEE Int. Geosci. Remote Sensing Symp.*, vol. 4. Jul. 2009, pp. 422–425.
[21] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, and C. Alberola-López, "Automatic noise estimation in images using local statistics. Additive and multiplicative cases," *Image Vis. Comput.*, vol. 27, no. 6, pp. 756–770, 2009.
[22] I. van Zyl Marais and W. Steyn, "Noise estimation algorithms for onboard image quality assessment," presented at the Int. Conf. Space Technol., Athens, Greece, 2009.
[23] M. Ghazal and A. Amer, "Homogeneity localization using particle filters with application to noise estimation," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1786–1796, Jul. 2011.

[24] N. Ponomarenko, V. Lukin, M. Zriakhov, A. Kaarna, and J. Astola, "An automatic approach to lossy compression of aviris images," in *Proc. IEEE Int. Geosci. Remote Sensing Symp.*, Jun. 2007, pp. 472–475.

[25] A. Danielyan and A. Foi, "Noise variance estimation in nonlocal transform domain," in *Proc. Int. Workshop Local Nonlocal Approx. Image Process.*, 2009, pp. 41–45.

[26] D. Makovoz, "Noise variance estimation in signal processing," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Jul. 2006, pp. 364–369.

[27] P. Wyatt and H. Nakai, "Developing nonstationary noise estimation for application in edge and corner detection," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1840–1853, Jul. 2007.

[28] A. Barducci, D. Guzzi, P. Marcoionni, and I. Pippi, "Assessing noise amplitude in remotely sensed images using bit-plane and scatterplot approaches," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2665–2675, Aug. 2007.

[29] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," in *Proc. 12th Int. Conf. Comput. Vis.*, 2009, pp. 2209–2216.

[30] D. Pastor, "A theoretical result for processing signals that have unknown distributions and priors in white Gaussian noise," *Comput. Stat. Data Anal.*, vol. 52, no. 6, pp. 3167–3186, 2008.

[31] S. Roth and M. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, 2009.

[32] U. Schmidt, K. Schelten, and S. Roth, "Bayesian deblurring with integrated noise estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Mar. 2011, pp. 2625–2632.

[33] K. Konstantinides, B. Natarajan, and G. Yovanof, "Noise estimation and filtering using block-based singular value decomposition," *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 479–483, Mar. 1997.

[34] D. Muresan and T. Parks, "Adaptive principal components and image denoising," in *Proc. Int. Conf. Image Process.*, vol. 1. 2003, pp. 101–104.

[35] T. Tasdizen, "Principal components for non-local means image denoising," in *Proc. IEEE Int. Conf. Image Process.*, Feb. 2008, pp. 1728–1731.

[36] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D image denoising with shape-adaptive principal component analysis," in *Proc. Workshop Signal Process. Adapt. Sparse Struct. Represent.*, vol. 49. 2009, pp. 1–6.

[37] H. Tang, N. Joshi, and A. Kapoor, "Learning a blind measure of perceptual image quality," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2011, pp. 305–312.

[38] I. Jolliffe, "Principal component analysis," in *Encyclopedia of Statistics in Behavioral Science*. New York: Springer-Verlag, 2002.

[39] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008 – A database for evaluation of full-reference visual quality assessment metrics," *Adv. Modern Radioelectron.*, vol. 10, no. 4, pp. 30–45, 2009.

[40] R. Hyndman and Y. Fan, "Sample quantiles in statistical packages," *Amer. Stat.*, vol. 50, no. 4, pp. 361–365, 1996.

[41] W. Prass, *Numerical Recipes: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[42] G. Smith and I. Burns, "Measuring texture classification algorithms," *Pattern Recognit. Lett.*, vol. 18, no. 14, pp. 1495–1501, 1997.

[43] V. Lukin, S. Abramov, M. Uss, I. Marusiy, N. Ponomarenko, A. Zelensky, B. Vozel, and K. Chehdi, "Testing of methods for blind estimation of noise variance on large image database," in *Theoretical and Practical Aspects of Digital Signal Processing in Information-Telecommunication Systems*. Shakhty, Russia: SRSUES, 2009, pp. 43–70.

[44] P. Koltsov, "Comparative study of texture detection and classification algorithms," *Comput. Math. Math. Phys.*, vol. 51, no. 8, pp. 1460–1466, 2011.

[45] M. Sharma and S. Singh, "Evaluation of texture methods for image analysis," in *Proc. 7th Australian Intell. Inf. Syst. Conf.*, 2001, pp. 117–121.

[46] M. Miller and N. Kingsbury, "Image denoising using derotated complex wavelet coefficients," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1500–1511, Sep. 2008.

[47] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[48] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[49] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, May 2007.

[50] P. Prucnal and B. Saleh, "Transformation of image-signal-dependent noise into image-signal-independent noise," *Opt. Lett.*, vol. 6, no. 7, pp. 316–318, 1981.

[51] A. Foi, "Noise estimation and removal in MR imaging: The variance-stabilization approach," in *Proc. Int. Symp. Biomed. Imag., Nano Macro*, 2011, pp. 1809–1814.

[52] R. Oktem, K. Egiazarian, V. Lukin, N. Ponomarenko, and O. Tsymbal, "Locally adaptive DCT filtering for signal-dependent noise removal," *EURASIP J. Adv. Signal Process.*, vol. 2007, p. 10, Jun. 2007.

[53] T. Anderson, "Asymptotic theory for principal component analysis," *Ann. Math. Stat.*, vol. 34, no. 1, pp. 122–148, 1963.

[54] G. Stewart and J. Sun, *Matrix Perturbation Theory*. San Diego, CA: Academic, 1990.

[55] J. Wilkinson, *The Algebraic Eigenvalue Problem*. New York: Oxford Univ. Press, 1988.

[56] R. Unnikrishnan, J. Lalonde, N. Vandapel, and M. Hebert, "Scale selection for the analysis of point-sampled curves," in *Proc. 3rd Int. Symp. 3-D Data Process., Visualizat., Transmiss.*, 2006, pp. 1026–1033.

[57] R. Unnikrishnan, J. Lalonde, N. Vandapel, and M. Hebert, "Scale selection for the analysis of point-sampled curves: Extended report," Dept. Electr. Eng., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-06-25, 2006, p. 335.

[58] M. Bryson, "Heavy-tailed distributions: Properties and tests," *Technometrics*, vol. 16, no. 1, pp. 61–68, 1974.

Stanislav Pyatykh was born in Sverdlovsk, Russia, in 1987. He received the Diploma degree in computer science from Belarusian State University, Belarus, in 2009. He is currently pursuing the Ph.D. degree with Heidelberg University, Mannheim, Germany.

He joined the Medical Faculty Mannheim, Heidelberg University in 2009. His current research interests include noise level estimation, image denoising, and image segmentation.

Jürgen Hesser received the Diplom-Physiker degree and the Dr. rer. nat. degree in physics, with a focus on global optimization with genetic algorithms, from the University of Heidelberg, Heidelberg, Germany, in 1988 and 1992, respectively, and the Habilitation degree in computer science from the University of Mannheim, Mannheim, Germany, in 1999.

He has been a Post-Doctoral Fellow with the University of Mannheim for computer graphics, bioinformatics, and medical technology since 1992. He became a Deputy Professor of medical technology with the University of Mannheim in 1998. He joined the Medical Faculty Mannheim, University of Heidelberg, as a Professor of experimental radiation oncology in 2007. He was a Co-Founder of three spin-off companies.

Prof. Hesser was the recipient of several Innovation Awards for an intelligent plaster in 2011, the Innovation Award and Roboters Prize for kV-MV imaging in 2010, the DEGRO Poster Awards in 2009, and several national and local Innovation Awards for excellent research.

Lei Zheng received the M.Eng. degree in biomedical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2004. He is currently pursuing the Ph.D. degree with Heidelberg University, Mannheim, Germany.

He visited University Medical Center Hamburg-Eppendorf in 2002 for the VOXEL-MAN project. He has been a Research Fellow with the University of Mannheim since 2004. He joined the Medical Faculty Mannheim, Heidelberg University, in 2008. His current research interests include vessel segmentation, brain segmentation, denoising, and statistical image analysis.